

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Intelligent Tourist Routes

Luís Fernando Frutuoso Fernandes Mouta

WORKING VERSION



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Alexandra Oliveira

Co-Supervisor: Luís Paulo Reis

Co-Supervisor: Carlos Rebelo

October 28, 2021

Intelligent Tourist Routes

Luís Fernando Frutuoso Fernandes Mouta

Mestrado Integrado em Engenharia Informática e Computação

October 28, 2021

Abstract

In a world that's increasingly social and open, to go to a not yet explored place without any help or sort of guidance is not ideal. It is always recommended to have a plan to enjoy sightseeing at its finest.

Nowadays in tourism everyone wants a fast and easy answer, nobody wishes all the hard work that happens behind the scenes on organizing a journey.

With millions of people traveling to unfamiliar cities to spend holidays, travel recommendation becomes necessary to assist tourists in planning their trips more efficiently

Since we live in a world extremely technological there is a need to optimize tourism to maximize the satisfaction of the tourists. Therefore, the creation of the concept Smart Tourism.

Smart Tourism is a new catch line applied to describe the increasing reliance of tourism destinations that allows for massive amounts of data to be transformed into value propositions.

There are new ways of describing technological, economic, and social features regarding smart tourism that rely heavily on sensors, big data, open data and new ways of connectivity and exchange of information.

By other words smart tourism can be defined as a way to support tourism as an integrated effort at a destination in combination with the use of advanced technologies to transform it into on-site experiences and business value-proposition with a clear focus on efficiency, sustainability and experience enrichment.

The purpose of this dissertation is to develop knowledge, through the use of Machine learning models, in order to simplify the whole planning journey operation, thus reducing planning time and increasing the pleasure of the user.

Serving as a prerequisite understanding tourist behavior patterns is therefore of great importance. Recently, geo-tagged photos on social media platforms like Flickr have provided a rich data source that captures location histories of tourists and reflects their preferences.

The intelligent system developed intends to help travellers to plan their visit according to their general preferences, obtained from , selections of photos of points of interest shown to them, and the restrictions of the traveller in terms of time to spend.

In this dissertation, information obtained from Flickr.com will be used to provide additional business knowledge. Using a data mining approach, diverse methodologies like clustering will be explored using several features.

To better explore the characteristics of the data in the problem above the results obtained from the machine learning model are compared and evaluated to obtain better conclusions regarding user's characteristics and preferences.

Four different machine learning algorithms are implemented and it is served as input the data regarding Tourist's information and as output, was obtained information regarding how well the data was distributed and general approaches that can be made with the result plots.

Lastly, it was possible to also allow, with the help of Machine learning methods, to become more accurate at predicting outcomes without being explicitly programmed to do so.

In a short form,with the help of a measuring technique the best algorithm turned out to be a mix between DBSCAN and BIRCH. On an overall sense the best one was DBSCAN but regarding the best value obtained was BIRCH.

Key-words(Theme): Smart Tourism, Smart Technology, Smart Business Ecosystems, Business Models, Open innovation, Big Data, Internet of Things, Smart Destinations, Smart Tourist, Smart City, Sustainable environment, E-tourism, Tourism experience, Digital Footprint.

Resumo

Em um mundo cada vez mais social e aberto, ir para um lugar ainda não explorado sem qualquer ajuda ou tipo de orientação não é o ideal. É sempre recomendável ter um plano para aproveitar o melhor dos passeios turísticos.

Hoje em dia, no turismo, todos querem uma resposta rápida e fácil, ninguém deseja todo o trabalho árduo que acontece nos bastidores para organizar uma viagem.

Com milhões de pessoas viajando para cidades desconhecidas para passar férias, a recomendação de viagens torna-se necessária para ajudar os turistas a planejarem suas viagens com mais eficiência

Por vivermos em um mundo extremamente tecnológico, é necessário otimizar o turismo para maximizar a satisfação dos turistas. Daí a criação do conceito Smart Tourism.

Smart Tourism é uma nova linha de captura aplicada para descrever a crescente dependência dos destinos turísticos que permite que grandes quantidades de dados sejam transformadas em propostas de valor.

Existem novas maneiras de descrever características tecnológicas, econômicas e sociais em relação ao turismo inteligente que dependem fortemente de sensores, big data, dados abertos e novas formas de conectividade e troca de informações.

Em outras palavras, o turismo inteligente pode ser definido como uma forma de apoiar o turismo como um esforço integrado em um destino em combinação com o uso de tecnologias avançadas para transformá-lo em experiências no local e proposição de valor de negócios com um foco claro na eficiência, sustentabilidade e enriquecimento da experiência.

O objetivo desta dissertação é desenvolver conhecimento, por meio da utilização de modelos de aprendizado de máquina, de forma a simplificar toda a operação de jornada de planejamento, reduzindo assim o tempo de planejamento e aumentando o prazer do usuário.

Servir como um pré-requisito para a compreensão dos padrões de comportamento do turista é, portanto, de grande importância. Recentemente, fotos com geo-tag em plataformas de mídia social como o Flickr forneceram uma rica fonte de dados que captura históricos de localização de turistas e reflete suas preferências.

O sistema inteligente desenvolvido pretende ajudar o viajante a planejar a sua visita de acordo com as suas preferências gerais, obtidas a partir das seleções de fotos dos pontos de interesse que lhes são apresentados e as restrições do viajante quanto ao tempo de permanência.

Nesta dissertação, as informações obtidas no Flickr.com serão usadas para fornecer conhecimento comercial adicional. Usando uma abordagem de mineração de dados, diversas metodologias como clustering serão exploradas usando vários recursos.

Para explorar melhor as características dos dados do problema acima, os resultados obtidos no modelo de aprendizado de máquina são comparados e avaliados para obter melhores conclusões quanto às características e preferências do usuário.

Quatro diferentes algoritmos de Machine Learning foram implementados e são servidos como dados de input referentes às informações do turista e como saída, foram obtidas informações sobre

como os dados foram bem distribuídos e abordagens gerais que podem ser feitas com os gráficos de resultados.

Por último, também foi possível permitir, com a ajuda de métodos de Machine Learning, a tornar-se mais preciso na previsão de resultados sem ser explicitamente programado para isso.

Resumindo, com a ajuda de uma técnica de medição, o melhor algoritmo acabou sendo uma mistura entre DBSCAN e BIRCH. De um modo geral, o melhor foi DBSCAN, mas em relação ao melhor valor obtido foi BIRCH.

Palavras-chave (tema): Turismo inteligente, Tecnologia inteligente, Ecossistemas de negócios inteligentes, Modelos de negócios, Inovação aberta, Big Data, Internet das coisas, Destinos inteligentes, Turismo inteligente, Cidade inteligente, Ambiente sustentável, E-turismo, Experiência turística, Digital Pegada.

Abbreviations

ML	Machine Learning
SSMS	SQL Server Management Studio
API	Application Programming Interface
Client	Person or Organization using the services of a professional person or company
CSS	Cascading Style Sheets – simple mechanism for adding style to Web documents
Framework	Layered structure of prewritten code to which you add your own code to solve a problem in specific domain
Front-End	A "front-end" application is an application in which the users interact with directly
HTTP	Hypertext Transfer Protocol – application protocol for distributed hypermedia information systems
HTTPS	Hypertext Transfer Protocol Secure – protocol for secure communication over a computer network
IDE	Integrated Development Environment – software application used by computer programmers for software development
JSON	JavaScript Object Notation - syntax for storing and exchanging data
REST	Representational State Transfer - Software architectural style of the World Wide Web
SQL	Structured Query Language - standard interactive and programming language designed for managing data held in relational database management system
UML	Unified Modeling Language - modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system
DBSCAN	Density-based spatial clustering of applications with noise
MS	Mean Shift
BIRCH	Balanced iterative reducing and clustering using hierarchies
AP	Affinity Propagation
E-Tourism	Electronic Tourism
CM	Computing Methodologies
B2B	Business 2 Business
B2C	Business 2 Client
C2C	Customer 2 Customer
FP-growth	Frequent Pattern Growth
eps	Epsilon Distance
SCRUM	Agile Methodology
URL	Uniform Resource Locator

Acknowledgements

I want to express my sincere thanks to everyone who helped and supported me in the realization of this project.

To the Informatics Engineering Department, I thank the opportunity that I had to develop technical and personal skills during the last years on an academic environment.

I would like to thank FEUP for proposing an ambitious project that proved to be a valuable experience.

This project would not have been possible without the help of my supervisors, Professor Luis Paulo Reis and Professora Alexandra Oliveira. I am grateful for their revising work and advices given to guide this project.

Special thanks to my family and my girlfriend ,for never giving up on me, and for helping me achieve what I never thought I could.

Finally, I want to direct my sincere thanks to my friends for the continuous support given, not only during this work but also during my academic years.

Luis Fernando Mouta

*“All things are difficult,
before they are easy”*

Dr. Thomas Fuller

Contents

1	Introduction	1
1.1	BackGround	1
1.2	Project Contributions	1
1.3	Main Objectives	2
1.4	Dissertation Structure	2
2	State of the Art	5
2.1	Addressed Problem	5
2.2	Smart Tourism Definition	7
2.2.1	Smart Tourism vs E-Tourism	9
2.2.2	Why Smart?	9
2.2.3	Smart vs Intelligence	11
2.3	Business Areas	11
2.3.1	Smart Tourism Technologies	12
2.3.2	The Concept of Tourist	12
2.3.3	The Concept of Smart Cities	13
2.3.4	The Concept of Smart Destination	15
2.3.5	The Concept of Smart Business	15
2.4	Market Research	16
2.4.1	Museu Digital da Universidade do Porto	16
2.4.2	Trip Advisor	17
2.4.3	Smart Tourism	17
2.4.4	Overview of reviewed Platforms and Applications	18
2.5	Machine Learning model	18
2.5.1	Supervised Learning	19
2.5.2	Unsupervised Learning	20
2.5.3	Reinforcement Learning	25
3	Work Environment	27
3.1	Proposed Model	27
3.2	Work Methodology	27
3.3	Work Planning	29
3.4	Estimated Timeline	30
3.5	Employed Technologies	30
3.5.1	Postman	30
3.5.2	Visual Paradigm	30
3.5.3	SQL Server Management Studio	30
3.5.4	PyCharm	31

3.5.5	Python	31
4	Technical Description	33
4.1	Technical Description	33
4.2	Analysis and Design	33
4.2.1	Business Logic	33
4.2.2	Domain Model	34
4.2.3	Requirement Analysis	36
5	Solution Development	41
5.1	Data Processing	49
5.2	Data Storage	50
5.3	Data Preparation for Machine Learning	53
5.4	Machine Learning	61
5.4.1	Mean Shift	62
5.4.2	Affinity Propagation	67
5.4.3	DBSCAN	73
5.4.4	BIRCH	80
6	Conclusions	87
6.1	Dissertation Summary	87
6.2	Accomplished Goals	87
6.3	Limitations and future work	88
6.4	Final Appreciation	88
6.4.1	Planning	88
6.4.2	Result Evaluation	89
6.5	Conclusion	90
A		91
A.1	Overall Definition of Smart Tourism	91
A.2	Tourist characterization based on the information collected from Flickr. Source: Own elaboration	91
A.3	Query from Flickr API to populate Photos Table	99
A.4	Query from Flickr API necessary to obtain a User's Information	99
A.5	Tourist Classification according to Chadwick	99
A.6	Sprint Tasks	99
	References	101

List of Figures

2.1	Data Mining Picture Diagram	6
2.2	Components and Layers of Smart Tourism	8
2.3	Smart Tourism vs E-tourism	10
2.4	Factors of a Smart City	15
2.5	Comparison between different platforms	18
2.6	Purity Function Score	22
2.7	DBSCAN main concepts	24
4.1	Activity Diagram - Overview of the Project	35
4.2	Domain Model of the Smart Tourism Project	36
4.3	Domain Mode Close Up on tourist Table	37
4.4	Domain Mode Close Up on Photo Table	37
5.1	Overview of API Flickr Methods for "Photo"	43
5.2	Result from Select * from Photos, intermediary results	49
5.3	Result from Select * from Tourists, initial results	50
5.4	Database Overview	51
5.5	Tourists being stored on SQL Server Management Studio	53
5.6	Gender Bar Chart	58
5.7	Location Bar Chart	59
5.8	First photo Published Data Chart	59
5.9	ListGroupUsers According to Topic Bar Chart	60
5.10	Preference into Topics Bar Chart	60
5.11	A comparison of different clustering algorithms	61
5.12	Pseudo-Code Mean Shift Algorithm for mode estimation and clustering	63
5.13	First Mean Shift Plot	65
5.14	Description of the Clusters of the First Mean Shift Algorithm	65
5.15	Mean Shift Plot of the Second Mean Shift Algorithm	66
5.16	Description of the Clusters of the Second Mean Shift Algorithm	67
5.17	Mean Shift Advantages and Disadvantages Table	67
5.18	Equations Affinity Propagation	68
5.19	Affinity Propagation Plot	70
5.20	Description of the Clusters First Affinity Propagation Plot	70
5.21	Second Affinity Propagation Plot	71
5.22	Description of the Clusters Second Affinity Propagation Plot	72
5.23	Affinity Propagation Advantages and Disadvantages Table	72
5.24	DBSCAN Pseudo Code - setOfPoints	73
5.25	DBSCAN Pseudo Code - expandCluster	74

5.26	DBSCAN First Implementation Plot	77
5.27	Description of the Clusters First DBSCAN Plot	77
5.28	Second DBSCAN Plot	79
5.29	Description of the Clusters Second DBSCAN Plot	79
5.30	DBSCAN Advantages and Disadvantages Table	80
5.31	BIRCH overview	80
5.32	Pseudo-code BIRCH	81
5.33	BIRCH Plot	83
5.34	Description of the Clusters First BIRCH Plot	84
5.35	Second BIRCH Plot	85
5.36	Description of the Clusters Second BIRCH Plot	85
5.37	BIRCH Advantages and Disadvantages Table	86
A.1	Tourist Classification	99

List of Tables

2.1	Definitions of Smart Cities	14
3.1	Scrum - Major Roles	28
3.2	Scrum – Sprint Specific Rules	29
3.3	Scrum Artifacts	29
4.1	Functional Requirements	39
4.2	Non-Functional Requirements	39
5.1	Rest of the Implemented methods to retrieve information from Flickr.com	46
5.2	Purity Measure Mean Shift	67
5.3	Purity Measure Affinity Propagation	73
5.4	Purity Measure DBSCAN	79
5.5	Purity Measure BIRCH	85
6.1	Result Evaluation	89
A.1	Overall Definitions of Smart Tourism. Source: own elaboration	91
A.2	Sprint Tasks	100

Code Snippet

5.1	API Request photos.search	43
5.2	API Request people.getInfo	44
5.3	API Request people.getPublicGroups	45
5.4	Gender Detector Implementation	47
5.5	Method cleanString	48
5.6	Method responsible for String Tokenize and Stemming and also removing Stop Words	48
5.7	Stop Words	49
5.8	Database Connection	51
5.9	Procedure to insert a Photo into Photos Table	51
5.10	Procedure to insert a Tourist into Tourists Table	52
5.11	Most Common Words Code Snippet	53
5.12	Result of Most Common Words Code Snippet	54
5.13	Sequence Matcher Code Snippet	54
5.14	DataFrame of Sequence Matcher	55
5.15	Prepare Corpus Auxiliary Method	56
5.16	Create Gensim LSA Model	57
5.17	Gensim LSA Model Results	57
5.18	First Mean Shift Implementation	63
5.19	Second Mean Shift Implementation	65
5.20	First Affinity Propagation Implementation	69
5.21	Second Affinity Propagation Implementation	71
5.22	First DBSCAN clustering Algorithm Implementation	75
5.23	Second DBSCAN clustering Algorithm Implementation	77
5.24	First BIRCH clustering Algorithm Implementation	82
5.25	Second BIRCH clustering Algorithm Implementation	83

Chapter 1

Introduction

In this chapter it is explained the background of this project as well as the motivation in accepting this challenge. It will also contain a brief presentation of this project to give an overview about the desired objectives.

1.1 BackGround

Most people like to travel and Porto was elected the most interesting city in Europe to visit in 2019. With great potential for attractiveness, Porto has endless options for tourist routes.

In order to be able to conceive a good trip some things have to be taken into consideration such as taking into account the needs and constraints of the environment and the user, but also allow some degree of free exploration of the city, adapting the offer according to the user's preferences.

This dissertation intends to develop an intelligent system capable of maximizing visitor satisfaction according to users' preferences and interests. The scope of this project is only focused entirely towards the city of Porto.

Regarding the user's preference it is easier if we draw a profile that is gauged directly through modern segmentation and profile discovery techniques and indirectly through the uploads given by users to sets of photographs of the places of interest.

In a general approach the desired intention and the main goal of this project is to create a virtual system, with module for segmentation and discovery of user profiles based on modern techniques where travelers can optimize their journey trips.

1.2 Project Contributions

One of the main advantages of the project is that it will allow, with the use of advanced technologies, to provide valuable and more efficient experiences to everyone.

Although the project is a great contribution to everyone in Porto, this project could be useful to all around the world.

In addition to the advantages mentioned before, the project is going to require a lot of knowledge to be acquired during the whole iterative process regarding skills in tools to be able to perform the desired solution.

With the help of a dataset provenient from Flickr.com and with the help of different Machine Learning Methods it is possible to make personalised recommendations and to draw conclusions regarding tourists information's and their own preferences when travelling to places.

The main contribution of this dissertation was to present a new model for smart tourism destinations using the steps of machine learning with the help of a clustering evaluation measure to possibly improve future tourism experience. The research present in this Dissertation intends to provide a theoretical contribution for future operationalization of the Smart Tourism concept to make personalised recommendations

1.3 Main Objectives

The main objectives of this work are also considered the relevant characteristics pointed in the list below:

- Data retrieval and Data Mining from a Geo-tagged website
- Creation of a Database that will serve as a base to store all the information needed to fulfill the main purpose of this project. (this information can be for example, meteorology information, news, cultural information, transportation schedules, museums information and so on);
- Application and Comparison of Artificial Intelligence Models adequate to the project regarding tourist's information
- Predict Preference places of a Tourist user
- Integrating the Database and the Intelligent System into an overall functioning solution.

1.4 Dissertation Structure

This Dissertation consists of four main chapters divided by specific sections that describe different matters.

The State of the Art chapter is initialized by a clear description of what is done in the areas related to this project scope, followed by the identification of the business area associated to the problem. The areas that are important to be mentioned are the proper definition of "Smart Tourism", the Business Areas, State of Art and the last area, market Research, which is focused on existing solutions that provide an insight of how other platforms provide ideas for features for the project in question.

In the Work Environment chapter is described the proposal model and is properly identified by describing the study that contributed to the selection of the best approaches to define a viable solution. It is also described the work methodology and the development process adopted. It is also defined in this chapter the technologies that were selected.

Finally, the last chapter, Conclusion, presents the conclusion of the project developed, highlighting the strengths and shortcomings of the solution. Besides that, it is referred eventual setbacks and future improvements.

To conclude, the bibliography and complementary information is attached. The appendices present additional information that is invoked during the Dissertation for a more detailed examination.

Chapter 2

State of the Art

This chapter explains in detail the problem addressed under the specified scope asked in the beginning of the project. It will also be addressed the essence of the Project, in which definitions such as Smart tourism, E-Tourism, Smart City, Smart Destination and Smart Tourist will come above and their meaning will need to be explained.

In the section of business areas it will be described the areas of business that this project affects.

And in the State of the Art section all information collected during the research phase on similar solutions, including details, advantages, disadvantages, and comparisons between elements will be documented. Similar market-based applications will also be examined in order to provide a better understanding on the envisioned solution.

2.1 Addressed Problem

In order to be an efficient travel operator it can't only take into account the user's needs and constraints, but also allow some degree of free exploration of the city, adapting the offer according to the user's preferences. The overall picture of the context is a good starting point in order to provide the user a memorable trip. The user also has the advantage of giving feedback on the suggested places of interest in order to enhance the learning of the system.

If the customer wants to plan a trip by recurring to this created solution, the first thing to do is to attack the user's preference and interests. These will be gauged directly through modern segmentation and profile discovery techniques and indirectly through the score given by users to sets of photographs (normal and 360) of the places of interest. The first way of engaging the user's preference and interests is to segment and discover the user profile based on modern questionnaire / wizard techniques and the second way is by using a module for segmenting and discovering user profiles based on indirect photo scoring techniques (normal and 360 °).

Before proceeding it is important to enhance that a parallel module, after the profile discovering techniques are applied, is also happening that collects complementary information relevant to the automatic generation of routes. This module is responsible for obtaining decisive instruments

that will help in creating the most perfect schedule for someone such as weather for natural factors, review platforms for peer rankings, score review by other users, schedule of attractions, local history, transports, resources and a bunch more.

Now that it is possible to have a better understanding of the scenario description the main modules comes in, which is the development of an intelligent tourist path generator that combines these various layers of information, maximizing visitor satisfaction that must be measured along the various points of interest generated for the route.

In order for the generated course to be completely optimized and also enhance the learning of the system it will be required along the way for the user to give feedback on the suggested placed of interest. Following this methodology it will be possible to guarantee the best touristic route for a specific customer and adapt in order to provide the max satisfaction.

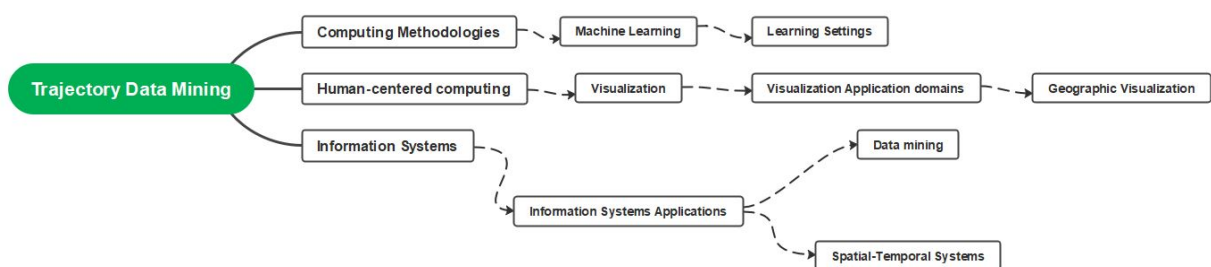
Also since most of the information won't be able to retrieve through the use of a hand-made questionnaire, instead, it will be required to fetch from Flickr.com API information regarding public Tourist's in order to have a dataset on which to work on. This information will be regarding the city of Porto and will provide insights on how past tourists came to Porto on holidays and shown what they visit and will serve as a good dataset.

Since this dataset will be obtained from public information, most of the data that will be retrieved will not be clean and there will have to be a need to recur to natural language techniques in order to make this data more readable and cleaner.

This problem basically consists on the appliance of different concepts, such as computing methodologies, Human-centered computing and Information Systems. By Computing Methodologies we try to reinforce Machine Learning and for the system to be able to learn for itself in order to provide, with each iteration, a even better trip to the user of the platform. Regarding Human-centered computing it is possible to state Visualization issues such as GeoGraphic visualization. By Last and most important, the Information Systems refers to the process of Data Mining and Spatial-Temporal Systems in order to obtain the most data possible.

A description of this concepts can be seen on (Figure 2.1):

Figure 2.1: Data Mining Picture Diagram



2.2 Smart Tourism Definition

The purpose of this section is to define Smart Tourism [1], clarify current smart tourism trends and then lists its technical and business foundations, it also is important to distinguish "Smart Tourism" from "E-Tourism", understand why it is called "Smart" Tourism and not Intelligent Tourism.

Smart Tourism is another popular expression used to portray tourism destinations, their industries and also their sightseers' expanding dependence on emerging forms of Smart Tourism which allows large amounts of data to be transformed into value propositions. However, taking into consideration that isn't a well defined concept and yet to be defined in a more specific way it basically prevents from knowing its true theoretical development. [2]

Essentially, Smart Tourism is described by its online services and it addresses the user's needs of obtaining inclusive information about tourism services rapidly, advantageously and conveniently by gathering, communicating and preparing the tourism information. [3]

It sees Smart Tourism that can deal with various issues and inconveniences looked by the Tourism information services. These issues and troubles are pointed towards seeking the max value of current tourism resources to achieve qualitative changes in the ways, channels and means of tourism information services.

By other words Smart Tourism can be seen as overall, transparent, precise, easy [4] and prompt application of tourism information with two types of techniques:

- smart interest and the utilization of management techniques that can capable of managing request and access;
- smart promoting innovation that can be utilized to target the proper customer segments to deliver fitting messages.

Lopez de Avila [5] defined "Smart Tourism" as the following:

"an innovative tourist destination, built on an infrastructure of state-of-the-art technology guaranteeing the sustainable development of tourist areas, accessible to everyone, which facilitates the visitor's interaction with and integration into his or hers surroundings, increases the quality of the experience at the destination, and improves residents' quality of life" [6].

According to this definition of Lopez de Avila, the focus is directed towards the traveler as he is the user of all these Smart Tourism innovations aiming to provide support for travelers in the following ways [7]:

- predict user needs based on a variety of factors, and making recommendations in the selection of context-specific consumption activities such as points of interest;
- enhance travelers location experience by providing rich information, location-based and customized interactive services;

- enable travelers to share their movement experience with the goal that they can help different travelers in their decision making process, revitalizing and fortifying their travelling experiences.

In a certain way, for Smart Tourism development, the most central concern is bridging the Physical world with the Digital world, thus allowing for a better overall experience regarding this social phenomenon (Smart Tourism) which is arising from the convergence of the tourism's experiences. [8]

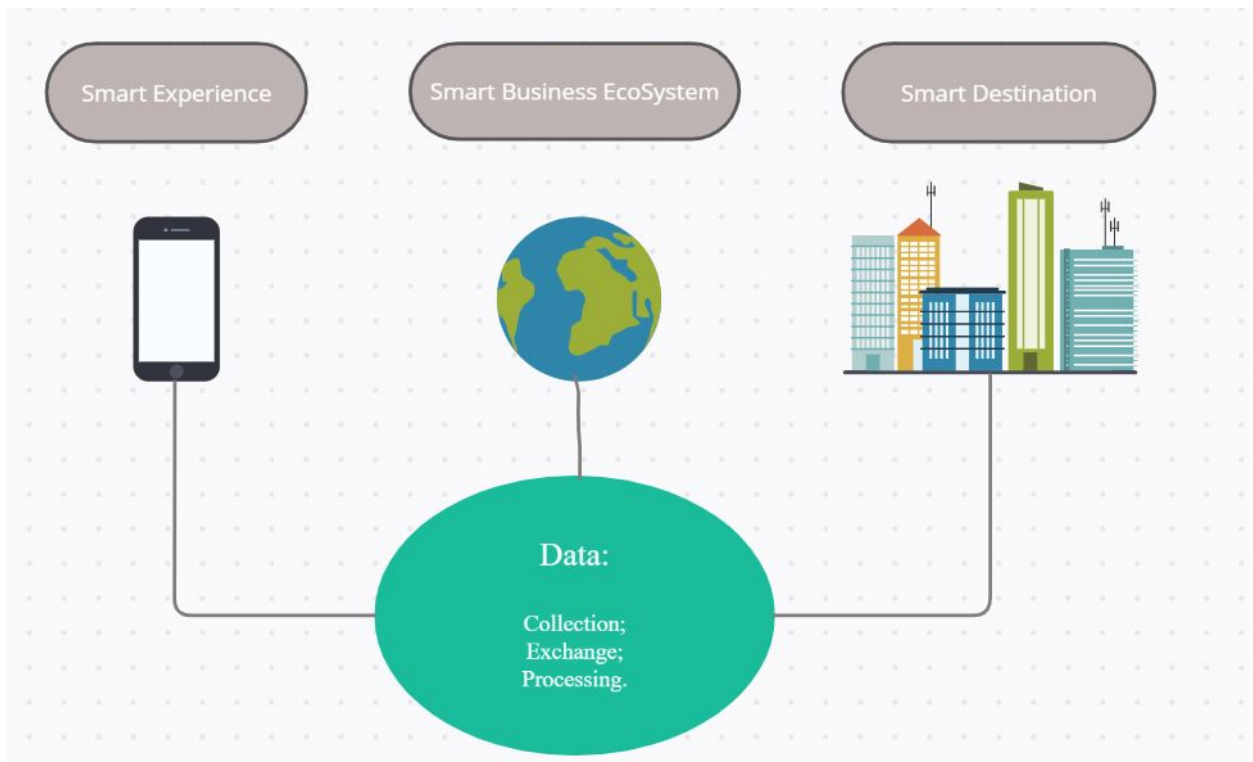
Smart Tourism is viewed as the last phase of the development of information and communication technologies including a physical ecosystem incorporating objections, organizations, people and public encounters. [9]

Data conglomeration, information aggregation, universal network and ongoing synchronization are the major drivers and the principal thrusters of such smart tourism experiences. The smart tourism experience is efficient and proficient and rich in meaning. Tourists became an active participant in its creation. Not exclusively will they consume but will likewise create, explain or otherwise enhance that data that constitutes the premise of the experience. [10].

What it means is that Smart Tourism spans three layers across three components: a smart data layer intended to gather information; a smart exchange layer supporting interconnection and by last a clever processing layer reliable for the information examination, analysis, perception, visualization, integration and intelligent use of data. [11]

This layers can be seen on (Figure 2.2):

Figure 2.2: Components and Layers of Smart Tourism



Smart Tourism likewise plainly lays on the capacity to gather tremendous measures of information as well as to cleverly store, measure, consolidate, combine and utilize enormous information to inform business innovation, operations and service.

Within a Smart Tourism setting, this innovation in the technology is a critical part of the data information system which is relied upon to furnish tourism consumers and service providers with more important data, better decision support, greater mobility, and, ultimately, more enjoyable tourism experience.

For sure, the very own concept of Smart Tourism is generally founded on the presumption that the consumers will willingly share information/data. Smart Tourism business relies heavily on free data and access to innovative platforms to make use of the information able and transform it into value propositions. Simultaneously, Smart Tourism infrastructures may prompt new data that is not balanced, which can be commercially exploited. [12]

Research in the field of Smart Tourism is still extremely limited and the majority of them give contextual analyses of existing projects. It additionally centers around consumer-perspective which raises a problem, customer privacy, which is an obvious issue in Smart Tourism.

Despite these concerns, Smart Tourism is still an incredible prospect which will bring more helpful, protected, manageable and sustainable living spaces for both residents and tourists, more customized travel encounters and therefore more relevant tourism experiences. Also this promising scenario will provide even greater opportunities for new services, business models and market openings. [13]

Summing up, Smart Tourism ultimately aims at revolutionizing tourist experience creation.

2.2.1 Smart Tourism vs E-Tourism

These two concepts are at the same time very similar and very different, E-tourism is about digital associations and Smart Tourism refers mostly to the connection established between the physical and the technological worlds.

Smart Tourism can be mistaken for E-Tourism because several concepts can have identical meaning for both types of Tourism, concepts as, information and communication advances, information systems, and online media ideas. [14]

With the improvement of data and correspondence advances, E-Tourism has arisen because of combination of worldwide dispersion and focal reservation system in the tourism industry with web-based technologies.

Smart Tourism is mostly distinguished from E-Tourism due to the combination of data and correspondence advancements with the actual physical infrastructure (Figure 2.3).

2.2.2 Why Smart?

The word “Smart” is the most popular expression when describing any development in technology, economic and social studies. These developments use sensors, big data, open data and new forms

Figure 2.3: Smart Tourism vs E-tourism

	E-Tourism	Smart Tourism
Sphere	Digital	Bridging digital & physical
Core Technology	Websites	Sensors & smartphones
Travel phase	Pre & post-travel	During trip
Lifeblood	Information	Big data
Paradigm	Interactivity	Technology-mediated co-creation
Structure	Value chain/intermediaries	Ecosystem
Exchange	B2B, B2C, C2C	Public-private-consumer collaboration

of connectivity and communication in order to be able to exchange information as well as the ability to infer and reason. [15]

The concept of “Smart” is used in an innovative way of actionable, near real time and real-time data, integrate and share data, use complex analysis, modeling, optimization, and visualization with the purpose of making better operational decisions. For instance, this term has been added to cities to describe the innovation in the use of technologies with the aim to achieve optimization, fair governance, sustainability, and a new level in the quality of life.

The meaning of "Smart" is the capacity to rapidly, deftly, and precisely comprehend and tackle issues. Being Smart is having wisdom, having experience and knowledge, having all kinds of information, and having decision-making ability, having new types of cooperation and worth creation that can bring development, business venture and seriousness. This concept has become an inexorably well-known term to portray innovative, financial, and social improvements powered by smart advancements that depend on sensors, large information, open information and open API, better approaches for availability among people and machines and multi-gadget, arranged trade of data. [16]

However, "Smart" has gradually become a frothy idea often used to promote clear political plans and arrangements for selling technological solutions. This is especially true because of "smart tourism". In addition, there is a lack of definition clarity: suddenly everything becomes “smart”. The tourism industry, it is often used for public information activities or trivial things, for example, to promote the development of free wireless Internet or multi-function applications. [17] [18]

Since tourism cannot be simply regarded as a combination of wisdom (people) and travel (industry), the term proposed is “smart tourism” instead of “wisdom tourism”. [19]

It is not unexpected to see the idea of "Smart" being applied to events that envelop the travel

industry. From numerous points of view, smart tourism can be viewed as a sensible movement from customary the travel industry and more as of late e-tourism.

A great deal of the data that makes the travel industry "smart" comes from associations upheld by online media and exploits distributed computing that arose with Web.

2.2.3 Smart vs Intelligence

Smart Tourism may be very much like the expression, "Intelligent Tourism". Intelligence means having the option to change the state or activity considering fluctuating circumstances, varying necessities, and past encounters, which implies that knowledge can produce suitable outcomes dependent on various requirements, various states, and distinctive notable encounters.

Nevertheless, "smart" signifies to make the best choice in different and complicated conditions, so entitling a program as "smart" is not quite the same as calling it intelligent". The substance of "smart" is broader and requires large data inputs. [20]

In addition, "intelligent" falls into the ambit of innovation, while "smart" puts more accentuation on the mechanical results for individuals. [21]

Smartness emphasizes the straightforwardness with which individuals can naturally get appropriate and exact services (being "Smart" can secretly see the individual's necessities and offer precise assistance data) by information aggregation with technological methods (devices).

2.3 Business Areas

In Europe a considerable lot of the Smart Tourism initiatives were a result or born off Smart City projects and, as an outcome, Smart Tourism destinations are progressively showing up in the European tourism landscape and travel industry scene. The focus is on Europe, however, is more on innovation and competitiveness and creating smart end-client applications that help enhance the tourism experiences utilizing previously existing information consolidated and prepared recently. [22]

Also if analyzed from another point of view, numerous new business types can arise and develop from Smart Tourism (for example, upscale tourism services), which has totally changed manners by which to organize and communicate tourism information and moreover has changed tourists' behaviour.

In Smart Tourism, technology is viewed as an infrastructure, instead of an individual data system, and incorporates an assortment of Smart computing technologies that coordinate equipment, programming, hardware, software and network technologies to give ongoing attention to this present reality and progressed investigation to help people settle on more astute choices about other alternatives. [23]

Numerous technological developments are thus instrumental to facilitating smart tourism goals.

A very important Area to be aware regarding Smart Tourism is also IoT, Internet of Things, which relates to everything that is connected to each other via the Internet without time, space and entity limitation.

The advancements driven by the IoT have significant ramifications and implications for the Smart Tourism development since travel involves development through reality and this "smart" environment will develop to know about, and have the option to address, the travelers' contextual needs in a pervasive yet non-intrusive way. [24]

However, eventually, understanding the Internet of Things(IoT) will be critical for creating the desired pervasive, "smart" technological environment that encompasses physical and digital infrastructures.

2.3.1 Smart Tourism Technologies

Smart Technology is a synopsis term for explicit technologies and technology-driven phenomena that gives information and availability in manners that were unrealistic previously.

In the context of tourism, Smart technologies are changing consumer experiences and are producing creative tourism business models. "Cloud computing, big data, mobile apps, location-based services, Geo-tag services, beacon technology, virtual reality, augmented reality, distributed computing, enlarged reality and social networking services are for the most part front line instances of Smart technologies enhancing the tourism experiences and services." [25]

When it is spoken off Smart Tourism technologies we are referring to a wide range of online tourism applications and information sources, for example, online travel agencies, personal blogs, social media, smartphone applications, government, and business web sites, and so on. In this regard, smart tourism technologies have a fundamental impact on the overall travel experience.

Regarding all the previous information boarded was possible to recognize six possible perspectives or levels of smartness for technology:

- Adapting: modifying behavior to fit the environment
- Sensing: carrying attention to regular things
- Inferring: making inferences from rules and perceptions
- Learning: utilizing experience to improve performance
- Anticipating: thinking and reasoning about what to do next.
- Self-organizing: be able to promote self-learning

By last it is also possible to see that "Smart Tourism" aims to employ mobile digital connectivity to make it even wiser, significant and sustainable among everyone but mainly between the tourists and the destination.

2.3.2 The Concept of Tourist

The current assumption is that all the information is incredibly significant to organizations and will be openly given by the Smart tourists who look for enriched tourism experiences.

Smart Tourism changes tourist information search behaviours. The tour information gets adaptable and different at the phase of collecting information. Tourists can acquire a wide range of touring data thorough web sites dependent on previous tour behaviour, clicking activity on web, spending records and other information sources. Tourists can likewise appreciate an experience of the tourist destinations by applying three-dimensional virtual reality software. By doing so they become acquainted about various information about tour destinations and can also receive electronic coupons.

Touring arrangements have become extremely flexible and accessible to tourists during their holiday. Tourists won't have an agenda fully restricted by the with complete booked hours of action and will be able to plan their trips before their departures due to the flexibility provided and they will also be able to change the arrangement at any time. The techniques for sharing any tour experience take different structures. For instance, tourists can record their own travel route via photographs taken at the destination. [26]

The Smart tourists use cell phones to take advantage of information infrastructures provided at the destination to practically enhance their experiences and add value to them.

Nowadays, the widespread use of mobile devices, particularly of the smartphones and its various applications, implies a period of exceptional availability and opportunity to the Tourist.

2.3.3 The Concept of Smart Cities

Ideas such as “digital cities”, “virtual cities”, “information cities”, and “cyber cities” have come to the forefront together with information societies.

Smart City concept has been created to make urban areas, that consume over 75% of the world's energy and produce 80% of the greenhouse gas emissions, more innovative, interconnected, maintainable, sustainable, comfortable, appealing and reliable. [27]

A general definition of "Smart City" is to be characterized as an urban environment which is supported by intelligent systems and can offer progressed and creative services to residents in order to improve the general nature and quality of their life.

To have a better overview of what is a Smart City a table (Table 2.1) is presented with several definitions.

Smart Tourism is by all means used to communicate smart destinations, which are an extraordinary and special piece of smart cities.

As indicated by another methodology, the purpose behind the development of a Smart City is to upgrade individuals' personal satisfaction by offering progressed and creative innovative types of services.

Proficiency and maintainability are basic drivers of the Smart City movement. Large information and open data, sensors inserted in city foundation like public transport and utilities, mobile connectivity, free Wi-Fi and versatile network are fundamental to creating innovative and technological application within Smart City frameworks. [28]

Table 2.1: Definitions of Smart Cities

Definition	Source
"A city [is] smart when investments in human and social capital and traditional (transport) and modern communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance"	Caragliu et al(2011, p.70)
"Creative or smart city experiments aimed at nurturing a creative economy through investment in quality of life which in turn attracts knowledge workers to live and work in smart cities. The nexus of competitive advantage had shifted to those regions that can generate, retain, and attract the best talent"	Thite (2011 pp. 623-4)
"A smart city is a well defined geographical area, in which high technologies, logistic, energy production and so on cooperate to create benefits for citizens in terms of well-being, inclusion and participation, environment quality, intelligent development: it is governed by a well defined pool of subjects, able to state the rules and policy for the city government and development"	Dameri(2013, p.2549)
"A smart city is an urban environment which, supported by pervasive systems, is able to offer advanced and innovative services to citizens in order to improve the overall quality of their life"	Piro et al (2014, p. 169)
"The most common characteristics of smart cities are a city's networked infrastructure that enables political efficiency and social and cultural development , an emphasis on business-led urban development and creative activities for the promotion of urban growth, social inclusion of various urban residents and social capital in urban development, the natural environment as a strategic component for the future"	Albino et al (2015, p.11)
"The smartness of a city refers to its ability to attract human capital and to mobilise this human capital in collaborations between the various (organised and individual) actors through the use of information and communication technologies"	Meijer and Bolivar(2015, p.7)

The idea has been unmistakably applied to urban areas and summed up under the term "smart cities". A Smart City at this point is considered a city that uses advanced and communication technology to enhance asset creation and utilization, optimize resource production and consumption.

To have a better understanding of the factors of a Smart city a figure is presented below (Figure 2.4):

A city is considered "smart" when interests in human and social capital and traditional transport and modern communication infrastructure fuel supportable financial development and a high quality of life, with an astute administration of regular assets, through the use of participatory government. [29]

Subsequently, Smart Tourism upholds city improvement and services in various manners. Steady innovation in applications of software, hardware and network developments implies that the Smart Tourism city can react quickly, productively and adequately to the tourism needs and necessities and will actually want to beat contenders and keep up long haul prosperity.

Smart Tourism permits tourists to better communicate and interact within urban areas to set up nearer associations and establish relationships with occupants as well as nearby organizations

Figure 2.4: Factors of a Smart City

<i>Factors of a Smart City</i>	
Smart economy (competitiveness)	Smart people (social and human capital)
<ul style="list-style-type: none"> • Innovative spirit • Entrepreneurship • Economic image and trademarks • Productivity • Flexibility of labour market • International embeddedness • Ability to transform 	<ul style="list-style-type: none"> • Level of qualification • Affinity to life long learning • Social and ethnic plurality • Flexibility • Creativity • Cosmopolitanism/Openmindedness • Participation in public life
Smart governance (participation)	Smart mobility (Transport and ICT)
<ul style="list-style-type: none"> • Participation in decision-making • Public and social services • Transparent governance • Political strategies and perspectives 	<ul style="list-style-type: none"> • Local accessibility • (Inter-)national accessibility • Availability of ICT-infrastructure • Sustainable, innovative, and safe transport systems
Smart environment (natural resources)	Smart living (quality of life)
<ul style="list-style-type: none"> • Attractivity of natural conditions • Pollution • Environmental protection • Sustainable resource management 	<ul style="list-style-type: none"> • Cultural facilities • Health conditions • Individual safety • Housing quality • Education facilities • Touristic attractivity • Social cohesion

in order to promote information disclosure.

The ultimate goal of smart cities is to expand seriousness and upgrade personal satisfaction, everything being equal, including inhabitants and tourists.

Another important concept that has emerged is Smart Destinations, which are special cases of Smart Cities: they apply Smart City principles and standards to infrastructure and think about occupants as well as tourists in their endeavors to help mobility, asset accessibility and designation, maintainability, personal satisfaction and also promote the quality of life's.

2.3.4 The Concept of Smart Destination

The Smart destination has been defined as intelligent and sustainable and is an augmentation of the Smart City in that it likewise incorporates the touristic foundation, for example, attractions, visit transports and so on. [30]

Lopez de Avila defined "Smart Destination" as the following: "an innovative tourist destination, built on an infrastructure of state-of-the-art technology guaranteeing the sustainable development of tourist areas, accessible to everyone, which facilitates the visitor's interaction with and integration into his or her surroundings, increases the quality of the experience at the destination, and improves residents' quality of life" [31].

This definition tries to portray the smart tourism destination as expecting stakeholders to be dynamically through technological platforms to collect, make, exchange information that can be utilized to enrich tourism experiences continuously.

2.3.5 The Concept of Smart Business

Smart Business refers to the complex business ecosystem that makes and supports the exchange of touristic assets and the co-making of the travel industry experience.

Furthermore it is also possible to describe smart business by powerfully interconnected stakeholders, the digitization of core business processes and measures and organizational agility.

Also smart business networks form an integral part of the smart tourism system. Smart Business along with the smart technology infrastructure and smart destination they form a smart tourism ecosystem. Smart Tourists that utilize their own innovations to take advantage of the assets of this smart tourism ecosystem also actively contribute information through their travels, movements, questions, query's and content uploads are additionally included as key species in the environment, among different players, for example, government, inhabitants, residents and media.

Collecting, processing, handling, trading and exchanging tourism-relevant data is an important information and considered to be a core function within the Smart Tourism ecosystem.

"A smart tourism ecosystem (STE) consequently can be defined as a tourism system that takes advantage of smart technology in creating, managing and delivering intelligent touristic services/-experiences and is characterized by intensive information sharing and value co-creation". [32]

The term Smart Tourism ecosystem infers first and foremost that its attention is on a shared objective or reason identified with the creation and utilization of touristic value, culminating in meaningful touristic experiences. [33]

2.4 Market Research

The Market Research was focused on existing solutions that would provide an insight of how other applications incorporate the various aspects of our solution, ideas for new features, as well as knowledge of what is actually useful for the user in the current state of the market.

In this section a research phase began, where the market was analyzed in search for the right tools and technologies for the realization of this project. This information gathered identified the potential and advantages of each technology, and possible limitations that could arise in the long term. When making the comparison between these similar technologies, it was possible to define which would give more advantages to the project. The choice of the most suitable tools and frameworks is crucial for achieving the project objectives to make the process easier and faster. Since this project was developed from scratch, there was greater freedom for selecting these frameworks.

2.4.1 Museu Digital da Universidade do Porto

The Digital Museum application of the University of Porto is a project of the Vice-Rectorate for Culture, with the technological support of Weblevel - Information Technologies, which aims to preserve and disseminate the material and immaterial heritage of a University that grows with the city Porto, contributing to the creation of a live, wallless digital locus, where the stories of artifacts, people and the construction of science are dynamically (co) created, (re) used and enriched. [34]

A native APP - Digital Museum of the University of Porto - was developed so that for all those who live in the city of Porto, have access to a vast historical and cultural collection existing in

the various organic units of the University of Porto, fully digitized. Tourists will also have the opportunity to have a more direct contact with the history, culture and characters that have marked the City, through the routes they can carry out.

It is a very simple and intuitive Application that allows the customer to be more aware of the vast historical empire of the city of Porto.

This application contains as well Geo-map utilities to support travelers in their sightseeing, containing information regarding historical and cultural points on Porto and also shows the most nearby touristic points that a tourist can access.

This application is similar to the solution we want to implement. This application provides a Geo-map utility and shows nearby touristic points along a course. This feature is identical to the platform intended to build.

The proposed solution and this application both try to offer the traveler utilities in order to maximize its satisfaction.

2.4.2 Trip Advisor

Trip Advisor is considered the world's largest travel platform, helping million of travelers every month in order to maximize each trip. Travelers around the world use the Trip Advisor application and website to search over reviews and opinions on numerous sources, restaurants, experiences, airlines and cruises. Whether you are planning to travel or are already traveling, travelers use Trip Advisor to compare low prices on hotels, flights and cruises, book and gather popular, as well as to book excellent restaurants. [35]

"Trip Advisor is an online travel research company, empowering users to plan and enjoy the ideal trip. Trip Advisor's travel research platform aggregates reviews and opinions of members about destinations, accommodations (including hotels, B&Bs, specialty lodging and vacation rentals), restaurants and activities throughout the world through its flagship Trip Advisor brand."

Trip Advisor Mission is to help people around the world so they can plan the perfect trip.

This application allows the user to do the following:

- Discover great tips and advice from travelers
- Save travel ideas and see them on a map
- Book must-do tours and activities

This application is also similar taking into consideration it allows for travelers to share ideas and tips among other travelers, resulting in a community with shared knowledge.

2.4.3 Smart Tourism

Smart Tourism offers a quick access to activities, news and important places of your region allowing for tourists to view comments of other tourists, share their experiences and recommend their own experience.

This application allows the user to do the following:

- search and Geolocation of places, activities and rides in your area;
- news and agenda;
- pictures published by users (#Hashtags and Social network);
- user's reviews and comments.

This application is very similar to the project scope intended to build since it basically states and defends the same, although it doesn't contain a smart tour generator.

2.4.4 Overview of reviewed Platforms and Applications

There still exists a wide range of platforms that contribute towards the same goal as the topic of this dissertation.

In the Figure below (Figure 2.5), a definition of more existing platforms will be shown that provide an insight of how they incorporate various aspects of their own solution and ideas for new features.

Figure 2.5: Comparison between different platforms

Reference	Focus	Variables	Issues and Limitations	Similarities to the Proposed Model
Musement	Musement helps travellers get the best from destinations by providing a great choice of local tours and attractions bookable.	Museum, Art, Tours, Attractions	Limitation of local planning Needs to implement a smart track generator	Helps travellers and tourists to get the best from destinations
Yelp	The Yelp app provides key information about businesses from user-generated content.	Photos, Tips, Users, Reviews	Slightly confusing Interface. As with any user-generated content app, all subjective information must be taken with a big grain of salt. Tips are a little buried.	Useful amount of information to be displayed for all kind of users. Easy to use and navigate with a clean UI. Relevant business details provided. Availability to post reviews and photos.
Moovit	Moovit is the leader in Mobility as a service solution and the most popular urban mobility app in the world.	Mobility, Travel	Only serves one purpose that is mobility, should abroad a wide range.	Allows for users to move around in urban areas via the best routes
Rail Planner	Rail Planner is essentially a trip planner and a timetable in your pocket.	Mobility, Travel, Planning, Trip, Timetable	Doesn't segmentize profile users. Should try to implement user-based scoring techniques using photos.	Plan a whole trip. Look up timers and help moving around

2.5 Machine Learning model

Machine Learning algorithms are programs (mathematics and logic), and they adjust themselves to make them perform better when they encounter more data. The "learning" part of Machine Learning means that these programs change the way they process data over time, just as humans change the way they process data through learning. Therefore, a Machine Learning algorithm is a

program that has a specific way to adjust its own parameters and to be able to give feedback on its previous performance in making predictions about a dataset.

Simply put, a Machine Learning algorithm is the engine of Machine Learning, which means an algorithm that converts a data set into a model.

There are several kinds of Machine Learning Algorithms such as supervised, unsupervised, classification, regression and so on, and to find which kind of algorithm works best for a specific problem depends on the kind of situation it is required to solve, the computing resources available and the nature of the data.

Machine Learning is a cycle where a system is prepared with the capacity to learn from experience over time. Machine Learning algorithms construct a model dependent on a dataset with the intent of anticipating some snippet of data or settling on potential results. These algorithms can be classified under three main categories:

2.5.1 Supervised Learning

In Supervised Learning, the Algorithms build a mathematical model from training data, in which the labels in the data set act as a teacher in order to train the model.

Supervised Learning, is considered a subcategory of Machine Learning. It is characterized by its use of labeled datasets to prepare algorithms to classify information or anticipate results precisely. As information is fed into the model, it changes its weights until the model has been fitted accordingly, which happens as a component of the cross validation process.

This category of Machine Learning utilizes a training set to help models to yield the ideal result. This preparation dataset incorporates inputs and correct outputs, which permit the model to learn over the long run. The algorithm estimates its exactness through the loss function, changing until the mistake has been adequately minimized.

Supervised Learning can be isolated into two kinds of issues: Classification and Regression

Classification utilizes an algorithm to precisely allocate test information into explicit categories. It perceives explicit elements inside the dataset and endeavors to reach a few determinations on how those elements should be labeled or characterized. The main classification algorithms can be seen on the following list:

- **Linear Classifiers** classify data into labels based on a linear combination of input features.
- **Support Vector Machines(SVM)** which helps to solve many practical problems by creating a line or a hyperplane which separates the data into classes.
- **Decision Trees** are a very specific type of probability that enables a person to make a decision regarding a specific process.

- **K-nearest Neighbor** is a simple algorithm that stores all available classes and classifies new cases based on a similarity measure, which can be, for example distance functions.
- **Random Forest** consists of many decision trees imbued with a technique that combines many classifiers to provide solutions to complex problems.

Regression is utilized to comprehend the connection that exists between dependent and independent variables. It is regularly used to make projections, for example, for deals income for a given business. The main Regression algorithms can be seen on the following list:

- **Linear Regression** is used to predict the value of a variable based on the value of another variable, according to the general sense, the value that is intended to predict is called the dependent variable and the other one independent variable.
- **Logistical Regression** allows to obtain odds ratio in the presence of more than one explanatory variable.
- **Polynomial Regression** fits a nonlinear relationship between the value of the corresponding conditional mean with its y-axis.

A lot more algorithms are used in supervised learning and only a few, the most common, were approached with brief explanations regarding them.

2.5.2 Unsupervised Learning

In unsupervised learning, the Algorithm builds a model on data that only has input functions but no output labels. Then the model is trained to find some structure in the data, by other words training is done without guidance which represents that the information that is sent to the model during the training phase is not labelled, categorized, or classified.

The model attempts to distinguish similarities in the data collection and make a design that is available in the information that is received. The accuracy of the clustering system is hard to assess in light of the fact that there is no target proportion of what segment of the information into separate clusters is the most valuable.

The main unsupervised learning issues can be isolated into clustering and association issues. The objective behind the first type of unsupervised learning issues is to find the normal groupings between elements in the dataset with the end goal that those components are comparable among themselves as per at least one significant qualities or properties, while in the last mentioned, the objective is to discover successive examples or create rules which are legitimate for huge parts of the data set collection.

Clustering is a technique that allows to discover patterns in data or natural grouping in data. Unlike Supervised learning, clustering Algorithms only interpret the input data and find natural

groups or clusters in feature space. The main Clustering algorithms can be seen on the following list:

- **Affinity Propagation** "consists in each data point sends a message to all other points informing its targets of each target's relative attractiveness to the sender". [36]
- **Spectral Clustering** is a technique that results in identifying agglomerates of nodes in a graph by connecting their respective edges. [37] [38] [39]
- **DBSCAN** is known as a density based clustering non parametric algorithm that given a set of points groups them together closely packaged marking as outliers points that fall into low-density regions. [40]
- **OPTICS** finds density-based clusters in spatial data by ordering the points of database according to their neighbors. [41]
- **Gaussian Mixture** represents a normally distributed sub population within an overall population. [42]

Association stands as a rule-based machine learning in order to help discover interesting relationships between variables. It also allows for the algorithm to try and learn taking into consideration that the data isn't labeled, therefore helping to discover some measures of interestingness. The main Association algorithms can be seen on the following list:

- **Apriori Algorithm** is used to gain insight into the structured relationships between different items involved. [43]
- **FP-growth** is widely used for frequent pattern mining. [44]

Regarding the problem of Smart Tourism the type of Machine Learning algorithms that are going to be used are the unsupervised ones, there are a lot of popular algorithms taking into consideration Clustering in unsupervised learning. Before the Technical implementation of the algorithms, all the implemented ones will be described accordingly.

A very important measure to be aware of the possibilities that can be done with the algorithm is to evaluate their Purity which is a simple and transparent measure method.

Purity is the measurement of the quantity of a prevalent component of a substance when only that component is present. In classification, purity measures the extent to which a group of records share the same class. It is also termed class purity or homogeneity, and sometimes impurity is measured instead. [45]

Purity is an external evaluation criterion of cluster quality. It is the percent of the total number of objects(data points) that were classified correctly, in the unit range [0..1].

Its calculation can be thought of as follows: For each cluster, count the number of data points from the most common class in said cluster. Now take the sum over all clusters and divide by the total number of data points. Formally, given some set of clusters M and some set of classes D , both partitioning N data points. [46]

Technically a "cluster well assigned" implies that each cluster has identified a group of objects as the same class that the ground truth has indicated. The ground truth classification of those objects as the measure of assignment correctness, however to do so it is required to know which cluster maps to which ground truth classification. If it were 100% accurate then each would map to exactly one, but in reality it contains some points whose ground truth classified them as several other classifications. Naturally then we can see that the highest clustering quality will be obtained by mapping which has the most number of correct classifications. [47]

To compute Purity, each cluster is assigned to a class which is most frequent in the cluster. To calculate it is needed to count the number of correctly assigned documents and divide it by N , which stands for the, total amount of documents. Formally it is known by (Figure 2.6):

Figure 2.6: Purity Function Score

$$\text{purity}(\Omega, \mathbf{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

In a general sense bad clustering have a purity value close to 0 and good clusterings have a purity value close to 1, if all the correct documents match the total amount it means that it has a perfect purity rate.

High purity is easy to achieve when the number of clusters is large, since there are a large amount of samples the data can predict better its own cluster.

Regarding the Algorithms that are going to be implemented, will be the following, Mean Shift, Affinity Propagation, DBSCAN and BIRCH.

2.5.2.1 Mean Shift

The unsupervised learning Algorithm in Machine Learning, mean shift or mode-seeking algorithm is based on the centroid-based in which the centroid finds the higher density center in dense smooth data points. [48]

Mean Shift is a very different learning Algorithm than the known flat clustering methodology of the k-means clustering, this new algorithm is known as a hierarchical clustering algorithm that unlike k-means that is explicit as a parameter the number of clusters to be passed as input, Mean Shift does not require that as an input parameter. The machine figures out how many clusters there ought to be and where those clusters are not requiring to be explicit said on the parameters. [49]

This Algorithm assigns the data points to the clusters in an iterative way by shifting points towards the mode, which is represented by the highest density of data points in the region, it works around the concept of Kernel Density Estimation also known as KDE. The Kernel is associated with mathematical computation related to the weight that is attributed to the data points.

In a more general approach Mean Shift is non-parametric, iterative mode-seeking algorithm widely used in pattern recognition and computer vision, it was originally used for density gradient estimation and is now used in areas such as classification.

This unsupervised algorithm aims to discover blobs in a smooth density of samples, it is also centroid based that works by updating points of the data set to centroids to be the mean of the points within a given region (which has also the name of bandwidth) [50]

In a general sense, Mean-shift clustering: Given a set of data points, the algorithm iteratively assigns each data point towards the closest cluster centroid and direction to the closest cluster centroid is determined by where most of the points nearby are at. So, in each iteration each data point will move closer to where the most points are at, which is or will lead to the cluster center. When the algorithm stops, each point is assigned to a cluster.

2.5.2.2 Affinity Propagation

Affinity Propagation is a graph theoretic clustering method that does not require to specify the number of clusters in advance. This algorithm takes as input the similarities between the data points and identifies the exemplars based on specific criteria's. [51]

Other techniques for clustering, like k-means clustering, are very sensitive to data sets and need to be rerun many times to obtain an optimal solution, in that sense a new approach exists which is the Affinity Propagation that tries to resolve these kinds of problems.

This algorithm is known in computer science as a message-passing algorithm, suggesting that it can be understood by taking an anthropomorphic viewpoint. Imagining that each item being clustered sends messages to all the other items informing it's respective targets of the target's relative attractiveness to the sender. Each target responds to all senders with a reply that contains its availability to associate with the sender. The message-passing procedure applies and is ran until a consensus is reached on the best linkage for each item. In an optimal view, the best associate for each item is that item's exemplar. Also, the same items that share the same exemplar are in the same cluster. [52]

Affinity Propagation is based on similarities between pairs of data points, and it simultaneously considers all data points as potential exemplars, or the so-called cluster centers. This technique searches for clusters in a recursive way throughout an iterative process.

The core idea of the Affinity Propagation is to absorb all the data points as if they were all potential clusters centers and the negative value of the Euclidean distance between two data points as the affinity. Taking this into consideration, the sum of the affinity of one data point for other data point is bigger and so is also the probability of this data point be a cluster center. By other words Affinity Propagation has a greedy strategy that maximizes the value of the clustering network during every iteration. [53]

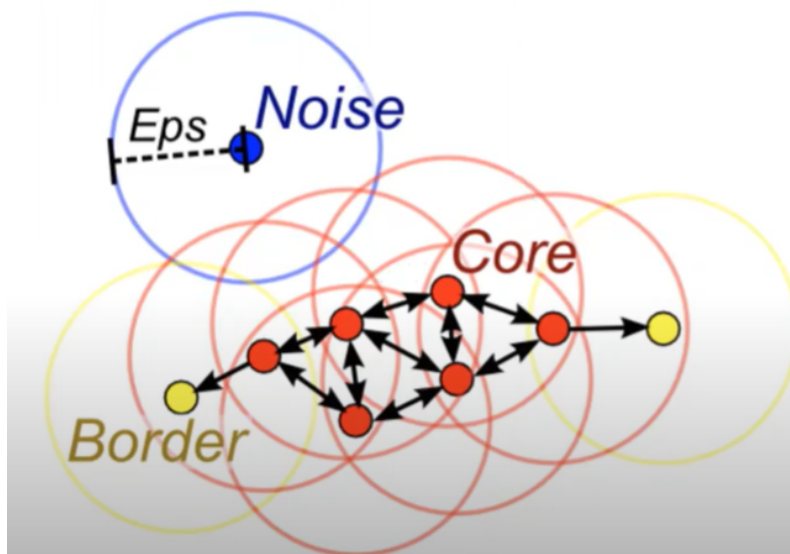
2.5.2.3 DBSCAN

Regarding this unsupervised algorithm, DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise, there are some basics required to know which the input parameters and the variables are used, to understand why they are required and what is their meaning. [54]

This algorithm views clusters as areas of high density and separates them from the areas that have low density, and due to this generic view of the algorithm, the clusters can actually be of any shape opposed with other types of unsupervised algorithms in which the clusters have a determined shape. The main component of this algorithm is the concept of “core points” which are the group of samples in areas of high density and is formally defined by two parameters, which are “*min_samples* and *eps*”, which define formally what the algorithm needs to work. [55]

The first parameter states the importance given to minimum number of samples that defines how many samples are required to form a cluster point, in a general way, if “*min_samples*” has the number of four it means that a point is a Core point if it has four samples in the same dense region. The second parameter which is *eps*, epsilon, is the distance measure introduced to form a dense region. So, to speak from a sample, we draw a circle with the radius of epsilon and all the points inside that circle belong to that core sample as it can be seen on (Figure 2.7).

Figure 2.7: DBSCAN main concepts



It is possible to deduce then that higher the “*min_samples*” value or lower the “*eps*” value indicates higher density necessary to form a cluster.

In a general way, a core point in the dataset exists if it meets the following conditions that there must have “*min_samples*” or above of other samples within a distance of “*eps*”, which are defined as neighbors of the core sample. If these requisites are met it means that the core sample belongs on a dense area of the vector space. A cluster is a set of core points that can be built by recursively doing what was expressed previously, taking a core sample, and finding all its neighbors that are core samples and so on. [56]

There are some exceptions to these points expressed before, which are the Border Points, these are still part of the cluster because it's within epsilon of a core point but do not meet the *min_samples* criteria stated above. These points are considered an outlier by the algorithm due to the fact of not meeting the criteria but are at least eps in distance from any core sample. [57]

By last, there can also be the case of a point not assigned to a cluster, which is a Noise Point, this point exists if from a sample we draw a circle with the distance of "eps" and there are not any data points inside this circle. [58]

The parameter "*min_samples*", in a way, tries to control the algorithm towards noise, since the higher this parameter becomes the most susceptible it becomes to large data sets, and the other parameter "eps" is crucial because it decides the distance function and it cannot be left at its own default value since it controls the local neighborhood of the points. By other words if the value of eps is too small all the data will be clustered, and if it is chosen too large it will cause close clusters to be merged into one cluster and eventually all the data will become one big single cluster. [59]

2.5.2.4 BIRCH

BIRCH demonstrates that is especially suitable for very large databases and it makes a large clustering problem tractable by concentrating on densely occupied portions and creating a compact summary. It utilizes measurements that capture the natural closeness of data and can be stored and updated incrementally in a height-balanced tree. [60]

Its inventors claim it as BIRCH, but it can also be named as Balanced Iterative Reducing and Clustering using hierarchies, allowing to perform hierarchical clustering, and having an ability that allows to cluster incrementally and dynamically incoming to produce the best quality clustering for a given data set.

BIRCH architecture additionally offers openings for parallelism, and for interactive or dynamic execution tuning dependent on knowledge about the dataset, acquired throughout the execution. This algorithm is local meaning that in each clustering decision that is made there is not the need to scan all the data points or all the currently existing clusters. It is often used to complement other clustering techniques by creating a condensed summary of the dataset. [61]

Taking into consideration that BIRCH can't represent categorical attributes all the information previously had to be converted into metric attributes for the algorithm to be able to compute its information. [62]

2.5.3 Reinforcement Learning

In reinforcement learning, the model learns to perform tasks by performing a set of actions and decisions completed by itself, and then learns from the feedback of these actions and decisions.

This category of Machine Learning tries to operate as a type of dynamic learning using combinations of punishment and reward. There is no apparent arrangement in these algorithms, however the reinforcement specialist attempts to expand the output through the criticism got from the environment.

Chapter 3

Work Environment

This chapter explains how the work development is going to be organized and what procedures were followed inside this project. At the beginning of this chapter it will also be presented a solution proposal.

3.1 Proposed Model

With the use of different Machine Learning models to predict a Tourist behaviour an attempt to solve the proposed problem will be made capable of maximizing user's preferences and interests.

It is also important to enhance, one more time, that this project will be specific to Porto only. The main functionalities of the intelligent system are going to be:

- Insert specific Porto touristic routes, schedules, transportation information.
- Be able to also discover user profiles based on photo commentaries and descriptions
- Possibility of inserting different types of user (a user can go on holidays for only 1 weekend or a full week or even a full month, and all these are different use cases that have to be approached differently).
- Makes use of different Machine Learning algorithms to predict a tourist behaviour

With this proposed model it is expected that the solution offered improves drastically the experience of customer journey. This list of requisites will be modified and/or refined throughout the project development whenever it is necessary.

3.2 Work Methodology

In this section it is presented the processes and techniques used during the working process. The project development is going to follow Scrum methodology and also some others techniques and tools related to version control.

Scrum is one of the best ways of implementing agile. It is a lightweight framework designed for small team management with the goal of developing complex products. [63]

A key principle of Scrum is the recognition that customers will eventually change about what they want or need from the product that is being developed, therefore there will be unpredictable changes. [64] As such, Scrum methodology states that a problem cannot be fully understood or defined up front, instead it is focused on responding to requirements and adapting to the market conditions. [65]

A Scrum Team is a group of individuals working together to deliver the requested product increments. In Scrum [66] [67] there are three major roles, Product Owner, [68], Scrum Master [69] and Team [70] identified, in (Table 3.1):

Table 3.1: Scrum - Major Roles

Product Owner	The Product Owner should be a person with vision. The Product Owner is responsible for continuously communicating the priorities to the development team, by other words he is a representation of the client. He is responsible for managing the Backlog, which includes expressing each item and ordering them to achieve the objective and making sure the Development Team understands the requirements.
Scrum Master	The Scrum Master is responsible for guiding the Development Team, ensuring the Scrum theory, practices and rules are being followed correctly and maximizing the value created by the team. The Scrum Master works to remove any impediments that are obstructing the team from achieving its sprint goals. This helps the team to remain creating and productive during the sprints and making sure its successes are visible to the Product Owner.
Team	The development team is responsible for self-organizing to complete work. A scrum development team must organize itself and have a high rate of collaboration between the elements in order to achieve the objectives of each sprint.

Since there is only one author on this project, and following the Scrum Methodology, this author will be considered as the Scrum Master every week and at the same time as a Team.

There is a technical person guiding the developing of the project, an advisor, Carlos Rebelo, that defines the primary activities to be realized in the project's scope.

Although the project isn't going to be made within a team environment of developers, the Scrum methodology and Agile Methods are still applied. The Sprint is the foundation of the Scrum. It consists in a period of time during which specific work has to be completed and made ready for review.

Each sprint contains the Sprint Planning, Weekly Scrums and Sprint Review. [71]

Therefore, it was build the Scrum table shown in (Table 3.2):

In order to provide key information to help the Scrum Team to understand and be aware of the product that is being developed there are some Scrum Artifacts. The two major artifacts used were the Product Backlog and the Sprint Backlog. [72] The Scrum Artifacts are identified on (Table 3.3):

Table 3.2: Scrum – Sprint Specific Rules

Sprint Planning	The sprint planning marks the beginning of the sprint and it's where the sprint goal and objectives are defined.
Weekly Scrums	This consists in a thirty-minute event, mostly made in the beginning of each week, where it was shared the completed tasks and what still needs to be completed and if any obstacle appeared.
Sprint Review	The Sprint Review marks the end of Sprint where a meeting takes place. In Vodafone Portugal the Sprint Review took place at the end of every week to keep track of the evolution of the work and what was accomplished. In this meeting feedback and ideas were discussed that brought more value to the product.

Table 3.3: Scrum Artifacts

Product Backlog	This artifact is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The product owner is responsible for the Product Backlog, including its content, availability, and ordering.
Sprint Backlog	The Sprint Backlog is a set of Product Backlog items selected for the sprint, where changes can be easily tracked so the Development team is aware of the progress done.

Although the Scrum is the main methodology used in this project, some auxiliary methodologies and tools related to version control are going to be used too.

For the version control it will be used the Bitbucket [73], which is a web-based solution used to host team projects allowing to safely store code and having good control over it. It contains features like pull requests, inline comments, and integration with tools such as JIRA [74].

3.3 Work Planning

The project is going to use Agile methodology SCRUM, having been divided into fourteen iterations. In order to provide a better overview of the work planning a Gant diagram was prepared with the temporal sequence of iterations.

The plan contemplates the existence of the following stages:

- study phase to relevant methodologies and technologies for the development of the project;
- the development of different modules;
- period for further improvements.

In addition to the steps described above, it is also planned that the preparation of the dissertation would be made during the period of the project.

According to the methodology, created a table, represented in (Annex 7), which records the estimate period of each sprint and the modules that were worked during each one.

3.4 Estimated Timeline

The assessed course of events is shown in the (Annex 7). The implementation of the Intelligent System and the development of the platform were be the majority of the work so they require more time and more effort.

3.5 Employed Technologies

Several technologies were used in order to develop the current project. The choice of technologies was not imposed and were selected according to their suitability to achieve the desired goal.

3.5.1 Postman

Postman [75] represents an API development tool which vision is to help build a super-fast and smooth workflow for API development by providing some powerful features (API Documentation Postman, 2017).

“To test an API a request is made to the required resource (usually a URL) using one the verbs (or methods). This request can also have headers defined or other additional parameters. The result of the request is an HTTP response, most commonly encoded in the JSON format and the respective status of the request” [76].

Postman was used to test the services created in the application to determine security flaws, bad formatting in the responses or any bug that may occur.

3.5.2 Visual Paradigm

“Visual Paradigm is a software tool designed for software development teams to model business information system and manage development processes.” [77]

In addition to modeling support, it provides dissertation generation and code engineering capabilities including code generation. It can also reverse engineer diagrams from code and provide round-trip engineering for various programming languages.

3.5.3 SQL Server Management Studio

SQL Server Management Studio is a technology first launched with Microsoft SQL Server that is used for configuring components on the SQL Servers.

SQL Server Management Studio [78] is an integrated environment for managing any SQL infrastructure. This software application is used to access, configure, manage, administer and develop all components of SQL Server. It also provides a single comprehensive utility that combines a broad group of graphical tools which work with objects and features of the server.

This technology was used in order to develop the database where all the information that is going to be fetched from the API will be stored.

3.5.4 PyCharm

PyCharm is a devoted Python Integrated Development Environment (IDE) giving a wide scope of fundamental tools for Python developers, firmly coordinated to establish an advantageous environment for productive Python, web, and data science development.

This IDE is used in computer programming, specifically for the Python language and was developed by the company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems and it also support web Development.

Pycharm has a smart code editor implemented that provides first-class support for all the languages it supports, such as Python, JavaScript, TypeScript, CSS and more. It also allows a regular user to take advantage of code completion, code inspections, on-the-fly error highlighting and quick-fixes.

3.5.5 Python

Python is currently, perhaps, the most famous and generally utilized programming language on the planet. Besides web and software development, Python is also used for purposes such as data analytics, Machine Learning and even design.

This language is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its significant level inherent information structures, joined with dynamic composing and dynamic binding, make it extremely appealing for Rapid Application Development. Python's straightforward, simple to learn grammar stresses meaningfulness and hence diminishes the expense of program maintenance. Python upholds modules and bundles, which energizes program measured quality and code reuse.

Regularly, developers experience passionate feelings for Python considering the expanded usefulness it gives. Since there is no aggregation step, the alter test-troubleshoot cycle is inconceivably quick. Troubleshooting Python programs is simple: a bug or terrible info won't ever cause a division issue. All things considered, when the translator finds a blunder, it raises a special case. At the point when the program doesn't get the exemption, the translator prints a stack follow. A source level debugger permits examination of nearby and worldwide factors, assessment of self-assertive articulations, setting breakpoints, venturing through the code a line at an at once, on. The debugger is written in Python itself, vouching for Python's thoughtful force.

Lastly Python Libraries play a vital role in Machine Learning and data science through the direct use of data manipulation and more. In this sense, Python Libraries are a set of useful functions that eliminate the need for writing codes from scratch.

Chapter 4

Technical Description

4.1 Technical Description

The Technical Description chapter describes the work done during the project. In the first section, **Analysis and Design**, it will be identified all the Architecture and Design that define the application, and the next chapter, **Solution Development**, will serve the purpose of explaining the implantation process.

Technical decisions were also made during the implementation of the solution which changed the design as a result, because of requirement changes or limitations on the technologies.

In this chapter decisions, used technologies and the followed patterns are explained. It is organized according to modules in which the application is structured.

4.2 Analysis and Design

4.2.1 Business Logic

In order to have a better overview of the project this section was elaborated to clarify all questions regarding this topic.

There is only one major actor involved, a regular tourist or by other words User.

All this business logic will be based on already existing users of Flickr.com that will provide insight and information regarding touristic places in order to build a recommendation system towards final users, which will be the new users that will have an enhance touristic travel journey.

This regular tourist has some requirements: it must have an account in Flickr and uploaded photos regarding Porto in order to be considered part of the Business Logic. Therefore he is then able to upload personal or public photos to the Flickr API which will be later retrieved to fill a dataset in order to promote Machine Learning Algorithms to build a recommendation system for future tourists.

A big overview of this project is that it starts with API Requests, GET, in order to be able to fetch all photos regarding Porto. This requests have a lot of restrictions regarding it, such as "Minimum Upload Date, Maximum Upload Date, Accuracy, Content Type, extras,etc" in order

to make the request the most accurate possible so the data obtained comes even more clean and appropriated towards the objective of this project.

After all the photos are retrieved regarding our main topic "Porto" it is still required to perform actions on top of this information, such as Data Cleaning, Stemming, Tokenize, among others in order for this data to become fully readable. When this data becomes clean, is then stored and inserted into a Database Table.

These requests towards the topic of "Porto" gave a lot of results and they allowed also to identify profiles of users that uploaded the photos we were able to fetch. Therefore all these users/tourists were then saved in a Tourist Table on the Database. It was also required to apply some Machine Learning Algorithms such as Gender Detector to be able to make some profiles readable and also to eliminate Duplicated Profiles.

Now, it is possible with all the information obtained to start preparing towards the main objective which is building a Machine Learning Algorithm on top of a certain dataset in order to produce a recommendation system for future tourists. In order to be able to promote all of these, some particular information had to be retrieved from the tables created earlier that would give a hint regarding tourist's preferences, places of interest or restrictions.

Flickr suffered a major privacy politic update and most of the information that was possible to retrieve from users was therefore denied leaving only two main attributes. The descriptions of the photos a user posts and the groups a user joins. With these two fields it was then possible to trace which are the preferences and points of interest of each specific tourist. It was then required to fetch all this information, the public group information regarding a tourist profile and also the descriptions of the photos a user posts and to submit to to all the operations to be able to retrieve clean Data at the end.

Lastly all the information was finally obtained and ready to start implementing Machine Learning algorithms to build the recommendation system.

In order to understand better this overview of the Project and all the big actions that are involved an Activity Diagram of the whole Project was elaborated and shown on (Figure 4.1).

4.2.2 Domain Model

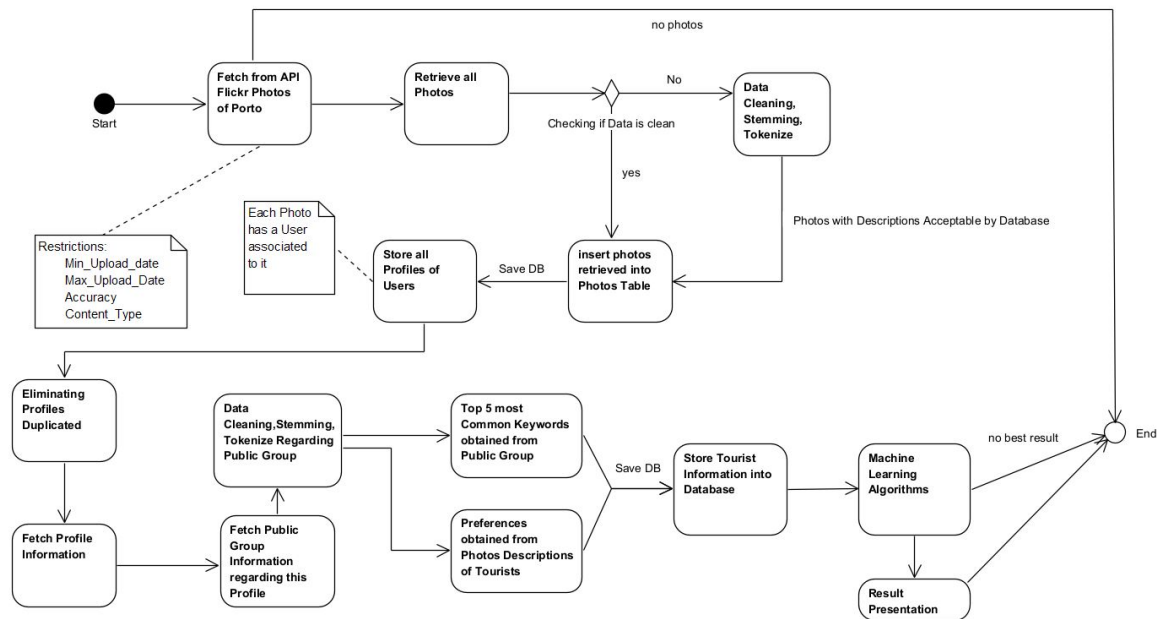
Domain Model [79] is a way to describe and model real world entities and the relationships between them, which collectively describe the problem domain space. Identifying domain entities and their relationships provides an effective basis for understanding the problem and helps practitioners design system for maintainability, testability, and incremental development therefore Domain Model consist on the primary modeling area in Agile development.

Using the Unified Modelling Language, it was built the model shown in (Figure 4.2).

Each domain concept has a specific definition that should be clarified early. In this way, the future use of this concepts will be more perceptible.

There is only one type of User in this project, also known as Tourist. A normal user/Tourist is going to be represented by the conceptual class Tourist and is able to perform the following actions:

Figure 4.1: Activity Diagram - Overview of the Project



check his own Gallery, review his Photo Set, have a place of Interest, have a route associated to him and answer a Wizard. A better overview of the Tourist table can be seen on (Figure 4.3).

The difference between Gallery and PhotoSet, relies on the fact that a Gallery contains the photos taken by the user and the PhotoSet may contain photos that weren't taken by the user but the user was able to fetch these photos from somewhere else and included them in his own PhotoSet.

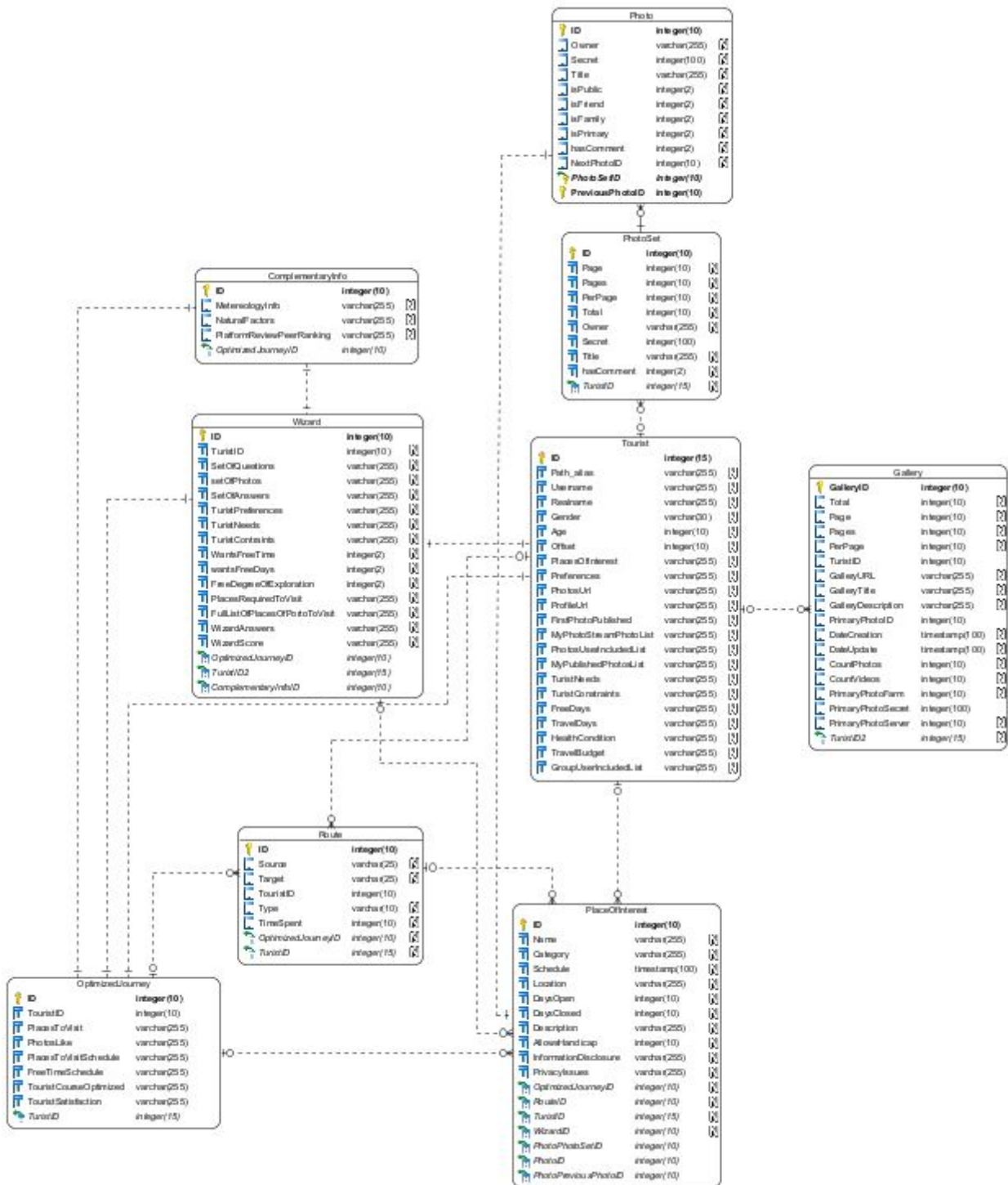
It is also necessary to emphasize that all the photos are stored in the Database because they contain information needed to guess the places of Interest of a specific Tourist. Each photo contains a description that is an indicator of a user like's. A better overview of the Photo table can be seen on (Figure 4.4).

A tourist will be mostly addressed due to his points of interest therefore having this link to the conceptual class PlaceOfInterest. Machine Learning Algorithms implemented further in this dissertation will require to know the tourist preferences and that's where the Tourists' Place of Interests tags are used.

The Route associated to the Tourist represents a sort of Optimized Journey which will be the output of the Recommendation System and it will store the final results, on the Database, on the Optimized Journey Table.

In a general sense, after all the requests have been made to the Fickr API and all the information has been retrieved and saved to the Database, Machine Learning Algorithms will be used and applied on top of this Datasets to be able to trace a behaviour on the Data. At the end of this procedure the recommendation system is built from the output of the Machine Learning Algorithm.

Figure 4.2: Domain Model of the Smart Tourism Project



4.2.3 Requirement Analysis

Analysis and the definition of the requirements is the first stage of the project. At this stage, client requirements are gathered and an initial analysis of the problem is made. FURPS+ [80] has been

Figure 4.3: Domain Mode Close Up on tourist Table















































Tourist		
 ID	integer(15)	
 Path_alias	varchar(255)	
 Username	varchar(255)	
 Realname	varchar(255)	
 Gender	varchar(30)	
 Age	integer(10)	
 Offset	integer(10)	
 PlacesOfInterest	varchar(255)	
 Preferences	varchar(255)	
 PhotosUrl	varchar(255)	
 ProfileUrl	varchar(255)	
 FirstPhotoPublished	varchar(255)	
 MyPhotoStreamPhotoList	varchar(255)	
 PhotosUserIncludedList	varchar(255)	
 MyPublishedPhotosList	varchar(255)	
 TuristNeeds	varchar(255)	
 TuristConstraints	varchar(255)	
 FreeDays	varchar(255)	
 TravelDays	varchar(255)	
 HealthCondition	varchar(255)	
 TravelBudget	varchar(255)	
 GroupUserIncludedList	varchar(255)	

Figure 4.4: Domain Mode Close Up on Photo Table

Photo		
 ID	integer(10)	
 Owner	varchar(255)	
 Secret	integer(100)	
 Title	varchar(255)	
 isPublic	integer(2)	
 isFriend	integer(2)	
 isFamily	integer(2)	
 isPrimary	integer(2)	
 hasComment	integer(2)	
 NextPhotoID	integer(10)	
 PhotoSetID	integer(10)	
 PreviousPhotoID	integer(10)	

adopted as a way to group these different requirements into more specific groups, respectively:

- Functionality
- Usability

- Reliability
- Performance
- Supportability
- Design Requirements
- Implementation Requirements
- Interface Requirements
- Physical Requirements

This is done on the basis of the information provided by the client in the form of meetings and documentation. In order to be able to design and implement this project's solution, the system's functional requirements and non-functional requirements had to be collected as part of the analysis phase of the project.

A functional requirement specifies something the system should do, like business rules, transactions, adjustments, administrative functions. In the other side we have a non-functional requirement that describes how the system works, by other words, it essentially specifies how the system should behave and that it represents the "quality attributes" of a system. [81]

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements, like performance, scalability, capacity, availability, reliability, maintainability, security, usability and response time.

FURPS+ was chosen to capture functional and non-functional requirements. The FURPS+ classification addresses both functional and non-functional requirements. FURPS is an acronym for Functionality, Usability, Reliability, Performance and Supportability [82]. The "+" in FURPS+ acronym is generally used to represent additional design constraints. In this case we will be addressing these additional constraints by separating them in design constraints, implementation, interface and physical categories [83].

The next sub-chapter will identify the modules of the Project and also the functional and non-functional requirements.

4.2.3.1 Functional Requirements

In this section, it is presented the functional requirements collected during the analysis period. They were collected from a tourist point of view trying to identify the affected modules [84].

Since a Use Case shows how a user interacts with a System in order to achieve a desired result that's exactly the same purpose of a functional requirement, to describe the functions and behaviors that a system is or should be capable of. Therefore the functional requirements are represented by the Use Cases.

The functional Requirements are presented on (Table 4.2):

Table 4.1: Functional Requirements

Functional Requirement	Module/Actions
Discovering Tourist Profile	API Requests
Discovering user profiles Based on modern questionnaire	API Requests + Machine Learning Algorithm
Discovering user profiles based on indirect photo scoring techniques	API Requests + Machine Learning Algorithm
Development of a module for collecting complementary information relevant to the automatic generation of routes	API Requests + Machine Learning Algorithm
Development of a Smart Tour generator that maximizes tourist satisfaction	Recommendation System
Adding a Photo to the Gallery	API Requests
Check Tourist Gallery	XX API Requests
Check Tourist Information/Profile	API Requests

4.2.3.2 Non-Functional Requirements

In this section, it is presented the Non-Functional Requirements collected during the analysis period.

In order to be able to develop this project there was the need to choose a framework in which the user could develop. The framework PyCharm was chosen between all the available Programming Tools due to the fact that this framework allows the user to easily adapt to Machine Learning Algorithms due to its high presence of Machine Learning Libraries.

The Non-Functional Requirements were collected from a user and a Database point of view identifying the affected modules.

Table 4.2: Non-Functional Requirements

Non-Functional Requirement	Module/Actions
Performance	API Requests
Scalability	API Requests + Machine Learning Algorithm
Capacity	API Requests + Machine Learning Algorithm
Availability	API Requests + Machine Learning Algorithm
Reliability	Recommendation System
Maintainability	API Requests
Security	XX API Requests
Data Integrity	API Requests

As said above, non-functional requirements indicate the system's 'quality attributes' or 'quality credits'.

The Project implemented contains high importance on Non-Functional Requirements such as Performance, it must thrive to have a good performance in order to improve a user's experience and so forth. All the non-functional requirements explicit have a high importance on the project.

Chapter 5

Solution Development

For the solution development process, it was established periodic sprints with goals to accomplish, involving one or more modules for each of them. In this section it will be described the development process with the support of Code Snippets, solution images to offer a visual perspective and the main decisions taken during the development phase.

Considering that during sprints there was parallel development between modules, in addition to describing each module, there will be a reference part to the interconnection between the affected modules.

Firstly a description of the desired dataset preparation will be mentioned and described, secondly there's going to be an approach to the techniques used to clarify the Dataset, and after this steps, Machine Learning algorithms will be introduced.

The main purpose of this section is to see if the initial hypothesis of this dissertation can be tested or not. This hypothesis consists in predicting places to be visited based on some profile data of tourists.

Data Collection

As referenced previously, the information will be retrieved and gathered from the photographs accessible on Flickr.com, a well known photograph sharing stage where individuals can share photographs. Data collection Users shared their travel path through footprints left in geo-tagged photos that they have taken and uploaded on Flickr. Destinations refer to popular places, such as attractions and/or landmarks within a city.

What separates Flickr from other famous photograph sharing applications like Facebook and Instagram is that it's really a photograph driven stage worked for proficient photographic artists and photography fans to show off their work while enjoying the work by others. [85].

It's more centered around the art of photography than some other significant informal community out there. Consider it Instagram for proficient photographic artists.

Flickr also promotes a very nice community, working on the odds of getting more openness for everybody's profile photographs. Other than favoriting other users' photos, creating galleries,

joining groups, and following people, everybody's client experience can be upgraded as well by doing the following: including descriptions for the photographs and labeling your photographs with keywords tags.

All these characteristics and all the information flickr provides make it an awesome starting point for this project allowing for the developer to have a majority of the information available at his disposal.

At the beginning of the project after some market research it was expected to have a lot more information regarding Flickr user's profiles. Fields like Profession, Age, Social status, Places of Interest, user's needs, user's constraints were expected to be retrieved from Flickr's but due to privacy policy updates, all these fields were restricted allowing for only a handful of fields to be retrieved.

So therefore to advance in this project the only fields possible to actually retrieve from Flickr that would be useful towards the goal of this project were the following: User ID, real name, Location, User's Photos and User's Groups, it is also important to refer that a large part of the fields are empty or with incomprehensible characters.

Still after this disadvantages Flickr is the most promising social platform regarding online photo management and sharing and allows for a better overview of all user's profile and photos.

In 2018 there were more than 10 billion photographs uploaded to Flickr and more than 175 million geotagged pictures stored. Regarding the city of Porto, since the start of 2018 until July 2020, there are more than five million photographs with the Tag "Porto", however just around 8% contained geotagged data. Of that 8%, just 20% are really identified with the city of Porto, Portugal. After information cleaning, it was possible to retrieve over 1 million photos for 318 tourists (See Annex 2). Given the extense list, the number of photos and the information regarding them are not available in (Annex 2) but the information is available for consulting if necessary.

Information was extracted from Flickr.com through the Flickr API. The API permits any people to approach these photographs alongside their text based metadata.

The Flickr API consists of a set of callable methods, and some API endpoints. This software intermediary allows the developer to access Flickr's information and retrieve Data from it. There are a lot of accessible fields in the Flickr API regarding all kinds of situations, such as Activity, Authentication, Profile, User, Group, Collection, Album and so on, but the most interesting field is regarding the "Photo" service as can be seen on on (Figure 5.1).

Several Querys were used in order to retrieve all the relevant information towards the development of the project. One example of a query used in this API contains some parameters necessary to fulfill this study aims (See Annex 4). It is important to also notice that all the requests have to be authenticated with an API_Key that had to be requested before starting this project.

The query returns the first 100 geotagged photos taken and uploaded since 1st January of 2018 to 1st of July of 2020 within the tag "Porto", given in the parameter tag. There are some limitations to what you can do with Flickr's API and the the maximum amount of photos possible to retrieve in one response is 100 and you can't increase this value, so you have to iterate through the pages of the response obtained.

Figure 5.1: Overview of API Flickr Methods for "Photo"

photos

- flickr.photos.addTags
- flickr.photos.delete
- flickr.photos.getAllContexts
- flickr.photos.getContactsPhotos
- flickr.photos.getContactsPublicPhotos
- flickr.photos.getContext
- flickr.photos.getCounts
- flickr.photos.getExif
- flickr.photos.getFavorites
- flickr.photos.getInfo
- flickr.photos.getNotInSet
- flickr.photos.getPerms
- flickr.photos.getPopular
- flickr.photos.getRecent
- flickr.photos.getSizes
- flickr.photos.getUntagged
- flickr.photos.getWithGeoData
- flickr.photos.getWithoutGeoData
- flickr.photos.recentlyUpdated
- flickr.photos.removeTag
- flickr.photos.search
- flickr.photos.setContentType
- flickr.photos.setDates
- flickr.photos.setMeta
- flickr.photos.setPerms
- flickr.photos.setSafetyLevel
- flickr.photos.setTags

Given this limitation, the algorithm had to be ran several times with loops, in order to overcome such limitation as it can be seen in Code Snippet 5.1

```
1 for page in range(0, int(pageTotalCount)) :
2     response = requests.get(
3         "https://api.flickr.com/services/rest/?method=flickr.photos.search&
4         api_key=c729641e6b00e16409c529d9dc22f89e&tags"
5         "=" + str(
6             apiKeyWord) + "&min_upload_date=2019-01-01&max_upload_date
7             =2021-05-01&accuracy=10&content_type=4"
8             "&per_page=100&page=" + str(page + 1) + "&format=json&nojsoncallback
9             =1")
```

Code Snippet 5.1: API Request photos.search

A lot more restrictions, limitations and adversities happened on this request. Beforehand it was required to make a blank request in order to retrieve the number of pages required to iterate over this cycle in order to obtain all the information that the Flick API could provide.

After this blank request was made then the real request could happen in order to fetch all the photos with the pre-defined criteria in the request. Most of the responses didn't come in the same way and there was sometimes a lot of extra fields to be aware of, and therefore the responses didn't had a pre-defined size so it wasn't easy to do a global method and the exceptions and boundaries of this method had to be tested out and included in the method to fetch the information from the API. Also it is important to refer that sometimes the delimiters of the strings were also changed and didn't appear in the response sometimes so an extra caution had to be placed regarding this situation as well.

For some attributes there wasn't the need to explicit them on the request because they already came in the response such as the information about when the photo was taken, the value "data-taken", and some other relevant fields as well.

After all these restrictions the request was possible to be done and all the photos regarding the case of Porto were retrieved with the extra arguments such as Geo Location.

The attribute extra was used with the value "geo" in order to simply download photos with geo information attached.

The property accuracy is related to the precision level of the location information and allows to choose the accuracy level of the photos (regarding this scale of information, the lower the scale the wider the range is, the lowest possible value is 1 that relates to World level and the highest recorded accuracy level is 16 which relates to Street level).

In this study, a very precise level -between 8 and 12- will be used, which implies that the data obtained will be at the level of Region or a City. In this case the only photos that truly matter are the ones related to a city in order to avoid miss-leading pictures.

After this request has been successfully achieve it was also of the developer's interest to acquire additional and more personal information about the users and a second query was also used (See Annex 4).

Regarding this API call it is only necessary to provide the user's ID in order to retrieve insights about an individual's information.

The results from this query provided insights about individual's username, real name, country origin, user photos url, profile user url, and also the first date when a photo was taken. Before the privacy policy update it was possible to retrieve information such as gender, social state, profession and also places of interest which adds a remarkable increment of information towards making a user's profile.

An example of a request to fetch user's information can be seen in Code Snippet 5.2

```

1   for x in range(0, len(listOfProfiles)):
2       responsePeopleGetInfo = requests.get(
3           "https://www.flickr.com/services/rest/?method=flickr.people.getInfo&
4           api_key=429fdbf7f9096180b7c964c869652482"
5           "&user_id=" + listOfProfiles[x] + "&format=json&nojsoncallback=1")
6   personInfo = responsePeopleGetInfo.content.decode().split('"person":{') [1]

```

Code Snippet 5.2: API Request people.getInfo

In order to make this request possible also a lot of exceptions and boundaries of this method had to be tested out and included in the method to fetch the information from the API. It is important to refer that sometimes user's had a profile with little to none information whatsoever which made the information retrieval process slightly harder due to making this limit test cases possible.

Also simultaneously to this process of Requesting user's profile it was also used another method in order to retrieved the public Groups that a user belongs to as it can be seen in Code Snippet 5.3

```

1     responseGetPublicGroups = requests.get (
2         "https://www.flickr.com/services/rest/?method=flickr.people.
   getPublicGroups&api_key"
3         "=429fdbf7f9096180b7c964c869652482&user_id=" + listOfProfiles[x] + "&
   format=json&nojsoncallback=1")
4
5     publicGroupsInfo = responseGetPublicGroups.content.decode().split('"
   groups":{') [1]

```

Code Snippet 5.3: API Request people.getPublicGroups

This API call only requests as well that the only field to be introduced is the user's ID in order to retrieve insights about individual's public Groups.

The response obtained from this request is as simple as a list of the groups that a user belongs to.

Some more requests have been made in order to retrieve more relevant information such as:

- people.getPhotos which returns photos from the given user's photostream. Only photos visible to the calling user will be returned.
- people.getPhotosOf which returns a list of photos containing a particular Flickr user.
- people.getPublicPhotos which returns the list of public groups a user is a member of.
- people.getGroups which returns the full complete list of groups a user is a member of which requires additional permissions and gives more information that the people.getPublicGroups request used in this project.
- galleries.getList returns the list of galleries created by a user. Sorted from newest to oldest.
- galleries.getContext returns next and previous favorites for a photo in a user's favorites.
- galleries.getListForPhotos returns the list of galleries to which a photo has been added. Galleries are returned sorted by date which the photo was added to the gallery.

In order to give a better overview of all the Requests used but that weren't described in detail a table regarding their information was created as it can be seen on (Table 5.1):

Table 5.1: Rest of the Implemented methods to retrieve information from Flickr.com

API Methods	Required Fields	Description
people.getPhotos	API_key Gallery_ID	Return photos from the given user's photostream. Only photos visible to the calling user will be returned.
people.getPhotosOf	API_key User_ID	Returns a list of photos containing a particular Flickr member
people.getPublicPhotos	API_key User_ID	Get a list of public photos for the given user
people.getGroups	API_key User_ID	Returns the list of groups a user is a member of
galleries.getList	API_key User_ID	Return the list of galleries created by a user. Sorted from newest to oldest.
galleries.getContext	API_key Photo_ID User_ID	Returns next and previous favorites for a photo in a user's favorites.
galleries.getListForPhotos	API_key Photo_ID	Returns the list of galleries to which a photo has been added. Galleries are returned sorted by date which the photo was added to the gallery.

Data cleaning and filtering

Regarding this section there are some assumptions that have to be made beforehand regarding for example the definition of Tourist and the information disclosure and privacy issues.

Within this study it is important to enhance that there are some differences between tourists, travelers and visitors. This differences were not taken into account so it was not necessary to distinguish these concepts as can be seen on Annex 5.

The second point it was needed to consider the amount of information disclosed. Sharing information on Flickr isn't mandatory and most part of the users didn't provide information about it. From all the users collected (318) only a slight part of them provided complete information about themselves. There was a lot of grammatical errors, spelling errors and wrong names regarding all the fields.

In order to prepare the Dataset it was required to actually clean and filter it properly. A lot of operations were submitted to the Data such as tokenize, removing stop words, data stemming and some other complementary operations.

The first operations that were used in this project regarding Data cleaning and filtering were the most simple ones when retrieving information directly from the API. A lot of the information

came with punctuation appended to the string and it was required to do a lot of splits in order to actually obtain the desired strings.

Although some operations are important to mention, the first problem encountered was actually finding the gender of the user's profile because due, once again, to the privacy policy update. This update didn't allow any longer to retrieve information regarding gender.

As such it was necessary to import the library "gender_guesser.detector" and implement code to actually try to find out if a person is actually from the male or female sex as it can be seen in Code Snippet 5.4

```
1     d = gender.Detector()
2
3     for x in range(0, len(listOfProfiles)):
4         if d.get_gender(listRealNames[x]) != "unknown":
5             listGenders.append(d.get_gender(listRealNames[x]))
6         else:
7             if d.get_gender(listUsernames[x]) != "unknown":
8                 listGenders.append(d.get_gender(listUsernames[x]))
9             else:
10                if d.get_gender(listPathAlias[x]) != "unknown":
11                    listGenders.append(d.get_gender(listUsernames[x]))
12                else:
13                    if d.get_gender(listRealNames[x].split(sep=" ", maxsplit=2)
14                    [0]) != "unknown":
15                        listGenders.append(d.get_gender(listRealNames[x].split(
16                        sep=" ", maxsplit=2)[0]))
17                    else:
18                        if d.get_gender(listUsernames[x].split(sep=" ", maxsplit
19                        =2)[0]) != "unknown":
20                            listGenders.append(d.get_gender(listUsernames[x].
21                            split(sep=" ", maxsplit=2)[0]))
22                        else:
23                            listGenders.append("Null")
```

Code Snippet 5.4: Gender Detector Implementation

This python package uses the underlying data from the program "gender" with the objective of obtaining one of the following results:

- unknown - name not found
- andy - androgynous
- mostly_male
- mostly_female
- male
- female

The difference between "andy" and "unknown" is that the former is found to have the same probability to be male than to be female, while the later means that the name wasn't found in the database.

The second problem was actually cleaning the Data because it came with a lot of special characters and regarding this aspect it was easy to overcome with a simple auxiliary function that eliminates special characters as it can be seen in Code Snippet 5.5

```
1 def cleanString(string):
2     return re.sub('[^a-zA-Z.\d\s]', '', string)
```

Code Snippet 5.5: Method cleanString

The third problem was actually preparing the document lists, this part is related to pre-processing the text and an auxiliary function was as well created as it can be seen in Code Snippet 5.6

```
1 def preprocess_data(doc_set):
2     """
3     #Input : document list
4     #Purpose: preprocess text (tokenize, removing stopwords, and stemming)
5     #Output : preprocessed text
6     """
7     tokenizer = RegexpTokenizer(r'\w+') # initialize regex tokenizer
8     p_stemmer = PorterStemmer() # Create p_stemmer of class PorterStemmer
9     texts = [] # list for tokenized documents in loop
10    for i in doc_set:
11        tokens = tokenizer.tokenize(str(i)) # clean and tokenize document string
12        stopped_tokens = [i for i in tokens if not i in stop_words] # remove
13        stop words from tokens
14        stemmed_tokens = [p_stemmer.stem(i) for i in stopped_tokens] # stem
15        tokens
16        non_digit_tokens = [i for i in stemmed_tokens if not i.isdigit()] #
17        remove digits
18        texts.append(non_digit_tokens)
19    return texts
```

Code Snippet 5.6: Method responsible for String Tokenize and Stemming and also removing Stop Words

Word tokenization is the process involved with splitting an enormous sample of text into words. This is a prerequisite in natural language processing tasks where each word should be caught and exposed to additional investigation like characterizing and counting them for a specific sentiment and so forth. The Natural Language Tool kit(NLTK) is a library used to accomplish this.

Regarding Stop Words, a stop word is a commonly used word (such as "the","a","an","in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. Removing this words allows for bigger words to take more importance and also we increment the space in the database by eliminating words with small interest. NLTK, once again, in python has a list of stopwords stored in 16 different

languages that allows for an easy implementation to promote Datasets more clean. (Code Snippet 5.7)

```
1 stop_words = set(stopwords.words('english'))
2 stop_words.add('user')
3 stop_words.add('belong')
4 stop_words.add('group')
5 stop_words.add('flickr')
```

Code Snippet 5.7: Stop Words

Lastly, another operation is Stemming which is the process of reducing a word to its word stem, this is a very important part in natural language understanding and is also part of the Natural Language Tool Kit.

There is also an operation which is important to refer which is the last one, which removes all the digits from the document set. After the operations such as tokenize, removing stop words and stemming document list it is also required to remove the digit words to fully complete the `pre_process_data` operation and have fully complete clean data.

5.1 Data Processing

Some additional fields had to be created and inserted so the data could be used for the purposes of this study. For example, an additional field was implemented on Tourist table that with the help of a library we were able to detect gender on the tourist.

Regarding photos table there wasn't the need to implement any extra fields, besides one, because all the information that was retrieved from the API was all necessary and it wasn't possible to detect anything else, the most relevant information is the description of the photo which was possible to retrieve as it can be seen on (Figure 5.2).

Figure 5.2: Result from Select * from Photos, intermediary results

ID	PerPage	Total	Owner	Secret	Title	isPublic	isFriend	isFamily		
89738	50936269613	1	100	16	191522024@N08	02376217f7	Bom Sucesso	1	0	0
89739	48156688826	1	100	16	144992359@N06	56d4ef8526	Bom Sucesso	1	0	0
89740	51044528168	1	100	16	144992359@N06	d1e58af2aa	Sketches from Algarve	1	0	0
89741	50730189631	1	100	16	191522024@N08	b6aa9fe1d5	PVu00f4r do sol	1	0	0
89742	50165650788	1	100	16	21182099@N05	c868105c06	Morpho telemachus iphiclus, larvae	1	0	0
89743	49516065826	1	100	16	26577438@N06	774e76d14f	A.R.S. Arquitectos	1	0	0
89744	49516289907	1	100	16	26577438@N06	834d7b58a5	A.R.S. Arquitectos	1	0	0
89745	48166202981	1	100	16	144992359@N06	8d70627287	Bom Sucesso - ready for Start but, ...	1	0	0
89746	48081042892	1	100	16	8047395@N04	181e3abcc5	6\u00aa Festival Junino	1	0	0
89747	33939846738	1	100	16	26577438@N06	13b4c21b68	ARS - Arquitectos. Porto, Portugal	1	0	0
89748	47764858282	1	100	16	26577438@N06	7826a8cf6e	ARS - Arquitectos. Porto, Portugal	1	0	0
89749	47264692242	1	100	16	60694653@N08	396bb446c7	STCP 3210	1	0	0
89750	46315487214	1	100	16	60083277@N00	86a1aec245	Farol e torre	1	0	0
89751	40874629513	1	100	16	60083277@N00	ba8d93d342	Farol decorativo	1	0	0
89752	32097760647	1	100	16	60083277@N00	68c12b1964	Farol nunca usado	1	0	0
89753	46736910982	1	100	16	60083277@N00	b0ec7a2089	Three sakuras	1	0	0
89754	49464017183	1	100	34	8751400@N06	793d82ac6f	101 (381)	1	0	0
89755	49464731722	1	100	34	8751400@N06	d11fc1955	101 (382)	1	0	0
89756	49464016478	1	100	34	8751400@N06	fe5765c0c3	101 (383)	1	0	0
89757	49464016348	1	100	34	8751400@N06	b7f823e244	101 (384)	1	0	0
89758	49464731152	1	100	34	8751400@N06	4b3eee99f3	101 (385)	1	0	0
89759	49464731002	1	100	34	8751400@N06	123052be3c	101 (386)	1	0	0
89760	49464730827	1	100	34	8751400@N06	b78d7042b7	101 (387)	1	0	0

Regarding the principal tables, Photos and Tourists, it was required to create that additional field for Networking Analysis purposes.

Regarding the owner attribute a extra field was created which will be assigned a number $N = \{1, \dots, n\}$ which will be used for linking the tables one to another. A similar approach was used in the other table. (Figure 5.3).

Figure 5.3: Result from Select * from Tourists, initial results

ID	Path_alias	Username	Realname	Gender	Location	Offset	Preferences
1	jb_1984	JB_1984	null	Null	null	null	Ribeira,Perola do Bolhao
2	alvaro pi	alvaro pi	alvaro pi	Null	Espa\u00f1a - Spain	+01:00	Palace da Aljazeera, Palac
3	photographer695	photographer695	null	Null	London, England	+00:00	Hotel Palace Nicolelis da
4	carloscoutinho	Carlos Afonso Coutinho	Carlos Afonso Pereira Coutinho	male	Porto - Portugal	+00:00	Porto, River Douro
5	Jos\u00e9 Santar\u00e9m	Jos\u00e9 Santar\u00e9m	Jos\u00e9 Santar\u00e9m	Null	null	null	River Douro, Boat, Fisherm
6	tiago_miranda94	Giacomo Giugiaro	Tiago Miranda	male	null	null	Trains, Marco Canaveses, P
7	heli3planes	heli3planes	heli3 planes	Null	null	null	Red Bull Air Race Porto, R
8	nagystvan88	nagystvan88	Nagy Istv\u00e1n	male	null	+01:00	Catmania,Cat
9	_tonidelong	_tonidelong	null	Null	null	null	Madrid
10	raeinforma	Real Academia Espa\u00f1ola	RAE	Null	Espa\u00f1a	+01:00	Family photo, Santiago Mun
11	itza	mnovela2293	maria luisa novela	Null	null	-03:00	Palace Linderhof,Baviera
12	tomasinrin	tomasinrin	Tom\u00e1s Ch\u00e1vez Hurtado	Null	null	-06:00	IMG_3908,IMG_3955
13	Luis Santiago Calle Quiros	Luis Santiago Calle Quiros	Luis Santiago Calle Quiros	male	null	null	Perspective Arquitectonica
14	Jacs fotojornalismo	Jacs fotojornalismo	Jacs	Null	Portugal	+00:00	jacs_photo_desp_mot_10441,
15	Cyro Henrique de Barros Lopes	Cyro Henrique de Barros Lopes	Cyro Henrique de Barros Lopes	Null	null	null	Botanic Garden, Jardim Bot
16	Biblioteca de Arte-Funda\u00e7\u00e3o	Biblioteca de Arte-Funda\u00e7\u00e3o	Biblioteca de Arte / Art Libra...	Null	null	+00:00	Se Porto,Portugal, Vista A
17	Ag\u00eancia Bras\u00edlia	Ag\u00eancia Bras\u00edlia	null	Null	Bras\u00edlia, Bra...	-03:00	Descarte irregular de oleo
18	Alexandre Lambertini	Alexandre Lambertini	Alexandre Lambertini	male	Bauru , Brasil	-03:00	Crazy Train
19	Fernando Stankuns	Fernando Stankuns	Fernando Stankuns	male	Jundia\u00ed, Braz...	-03:00	Jesus Christ, aleljadinho,
20	Moacir de Sa Pereira	Moacir de Sa Pereira	Moacir de Sa Pereira	Null	Tala\u00ed, Casc...	+00:00	Praia,Beach

This extra fields allows to link the whole chain.

Even so, it starts to become a bit tricky to see all this information, after all, there exists over 1 million photos and over 300 different users. In order to be able to get a better overview of the information and to be able to provide a better insight regarding all the tables an export to Excel was made and the complete information was transformed into different .CSV files.

Exporting to CSV allowed for a better understanding of all the concepts involved and another point of view of the data collected, such as:

- Number of tourists and photos considered;
- Number of photos taken by region/place/attraction;
- Number of tourists that photographed a certain place/region/attraction;
- General tourists' characterization, considering origin, username, and public groups;
- General photo description and places that tourists have visited;

5.2 Data Storage

Since we are dealing with a big dataset of information, it will be needed to resort to the help of a Database which can store very large numbers of records efficiently. A database also provides the tools to ease on the search for a specific record in a given set of data. It also allows to add new data with ease and in case of need, to edit or delete old data.

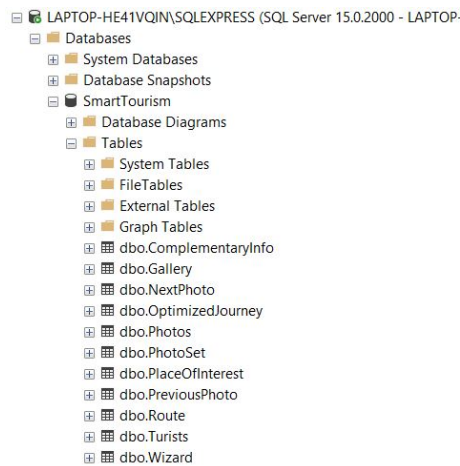
Databases also can do several operations with just a click of the button, such as sorting elements, ordering by category, hide or delete columns and even import into another application and also allows for more than one person to be able to access the same database at the same time.

At last it also allows to build query's on top of the database which allow to retrieve information regarding really specific data.

Such a database was built on SQL Server Management Studio (SSMS), which is an integrated environment for managing any SQL infrastructure, proving tools to configure, monitor, administer and query instances of SQL Server.

An overview of the project explorer of the created Database can be seen on (Figure 5.4).

Figure 5.4: Database Overview



After the creation of the Database and the respective tables it was necessary to link the code being developed in PyCharm to the Database created in SSMS.

A connection was made between both modules in a way that interacts between as a bridge and allows to make possible requests as well as store information.

The connection can be seen in (Code Snippet 5.8).

```

1 conn = pyodbc.connect ('Driver={SQL Server};'
2                       'Server=LAPTOP-HE41VQIN\SQLEXPRESS;'
3                       'Database=SmartTourism;'
4                       'Trusted_Connection=yes;')
```

Code Snippet 5.8: Database Connection

After these two modules are interconnected it is possible to store information on the Database creating procedures for that same purpose.

All the created procedures store information in order to populate all the tables in the Database, although for the purpose of simplicity only two procedures will be shown here.

In order to store information regarding a photo all the fields have to be passed with the restriction imposed by the Database as it can be seen in (Code Snippet 5.9).

```

1 def insert_variables_into_photos_table(ID, Page, Pages, PerPage, Total, Owner,
2   Secret, Title, isPublic, isFriend, isFamily, isPrimary, hasComment):
3   cursor = conn.cursor()
4   sql_insert_query = """INSERT INTO Photos (ID, Page, Pages, PerPage, Total,
5   Owner, Secret, Title, isPublic, isFriend, isFamily, isPrimary, hasComment)
```

```

4             VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?) """
5 # VALUES (%d,%d,%d,%d,%s,%s,%s,%d,%d,%d,%d,%d)
6 record = (
7     int(ID), int(Page), int(Pages), int(PerPage), int(Total), Owner, int(
8     Secret), Title, int(isPublic), int(isFriend), int(isFamily), int(isPrimary),
9     int(hasComment)
10
11 cursor.execute(sql_insert_query, record)
12 conn.commit()

```

Code Snippet 5.9: Procedure to insert a Photo into Photos Table

The same applies if you want to store a tourist as it can be seen in (Code Snippet 5.10)

```

1 def insert_variables_into_tourist_table(ID, Path_alias, Username, Realname,
2   Gender, Location, Age, Offset, PlacesOfInterest, Preferences, PhotosUrl,
3   ProfileUrl, FirstPhotoPublished, MyPhotoStreamPhotoList,
4   PhotosUserIncludedList, GroupUserIncludedList,
5   MyPublishedPhotosList, TouristNeeds, TouristConstraints, FreeDays, TravelDays,
6   HealthCondition, TravelBudget):
7   cursor = conn.cursor()
8   sql_insert_query = """INSERT INTO Tourists (ID, Path_alias, Username,
9   Realname, Gender, Location, Age, Offset, PlacesOfInterest, Preferences,
10  PhotosUrl, ProfileUrl, FirstPhotoPublished, MyPhotoStreamPhotoList,
11  PhotosUserIncludedList, GroupUserIncludedList, MyPublishedPhotosList,
12  TouristNeeds, TouristConstraints, FreeDays, TravelDays, HealthCondition,
13  TravelBudget)
14
15             VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?) """
16 record = (
17     int(ID), Path_alias, Username, Realname, Gender, Location, int(Age),
18     Offset,
19     PlacesOfInterest, Preferences, PhotosUrl, ProfileUrl, FirstPhotoPublished
20     ,
21     MyPhotoStreamPhotoList, PhotosUserIncludedList, GroupUserIncludedList,
22     MyPublishedPhotosList, TouristNeeds, TouristConstraints, int(FreeDays),
23     int(TravelDays), HealthCondition, int(TravelBudget)
24
25 cursor.execute(sql_insert_query, record)
26 conn.commit()

```

Code Snippet 5.10: Procedure to insert a Tourist into Tourists Table

Some complementary help had to be applied regarding Tourists' table due to the appearance of duplicates. In order to prevent this from happening all the duplicated profiles were eliminated from the Tourists' table. An overview, in order to see that it is being correctly saved on the other endpoint, of the data being saved on the SSMS database can be seen on (Figure 5.5).

Figure 5.5: Tourists being stored on SQL Server Management Studio

ID	Path_alias	Username	Realname	Gender	Location	Offset	Preferences
1	64453831@N08	jb_1984	JB_1984	Null	null	null	Ribeira,Perola do Bolhao
2	132825659@N02	null	alvaro pi	Null	Espal00f1a - Spain	+01:00	Palace da Aljafferia, Palace de los luna
3	41087279@N00	null	photographer695	Null	London, England	+00:00	Hotel Palace Nicoletis da Orlando las damas, Santo Domi...
4	50357226@N00	carloscoutinho	Carlos Afonso Coutinho	male	Porto - Portugal	+00:00	Porto, River Douro
5	159915226@N07	null	Josiu00e9 Santariu00e9m	Null	null	null	River Douro, Boat, Fisherman
6	48083463@N08	tiago_miranda94	Giacomo Giugiaro	male	null	null	Trains, Marco Canaveses, Porto, Sao Bento, Station, CP
7	158534870@N07	heli3planes	heli3 planes	Null	null	null	Red Bull Air Race Porto, Race, Air, 2017
8	186269826@N02	null	ngyistvan88	male	null	+01:00	Catmania,Cat
9	40685767@N00	_tonidelong	_tonidelong	Null	null	null	Madrid
10	118450654@N07	raeinforma	Real Academia Espalu00f1ola	Null	Espalu00f1a	+01:00	Family photo, Santiago Munoz Machado,Javier Nadal
11	131897066@N07	itza	mnovela2293	Null	null	-03:00	Palace Linderhof,Baviera
12	21195678@N04	tomasinrin	tomasinrin	Null	null	-06:00	IMG_3908,IMG_3955
13	192654410@N04	null	Luis Santiago Calle Quiros	male	null	null	Perspective Arquitectonica
14	142960189@N08	jacs-sport	Jacs fotojornalismo	Null	Portugal	+00:00	jacs_photo_desp_mot_10441,jacs_photo_desp_mot_13...
15	192000349@N07	cyrohenrique	Cyro Henrique de Barros Lopes	Null	null	null	Botanic Garden, Jardim Botânico, Rock Structure
16	26577438@N06	bibliarte	Biblioteca de Arte-Fundalu00e7u00e3o Calouste G...	Null	null	+00:00	Se Porto,Portugal, Vista Areca, Aproveitamentos Hidralic...
17	64586261@N02	agenciabrasilia	Agju00eancia Braslu00edia	Null	Braslu00edia, Brasil	-03:00	Descarte irregular de oleo causa prejuizos a rede de esgo...
18	25710556@N06	alexandreilambertini	Alexandre Lambertini	male	Bauru., Brasil	-03:00	Crazy Train
19	82289802@N00	stankuns	Fernando Stankuns	male	Jundiaiu00ed, Brazil	-03:00	Jesus Christ, alejajadinho.carregamento da cruz, thom cro...
20	69574134@N00	moacirdsp	Moacir de Sa Pereira	Null	Talalu00edde, Cascais, Portugal	+00:00	Praia,Beach

5.3 Data Preparation for Machine Learning

After all the information has been fetched from the Flickr API and stored in the Smart Tourism Database that was created for that purpose it is now necessary to prepare the Data for the Machine Learning Algorithms.

There are mainly two big types of information that are going to be dealt with in order to predict the places of interests of a Tourist, which are the description of the photos and the public groups a Tourist belongs to. There are only these two fields that a developer can work with in order to obtain further information regarding the tourist, once again, due to policy privacy update from Flickr part.

Taking into consideration that only these two fields are going to be useful it is necessary to promote some procedures to do on top of them in order to make this data actually readable and in order to retrieve some pertinent information from it.

Regarding the information of the public groups a tourist belongs, it is possible to know what are the user interests analyzing the most common words of all the groups of the user. With the help of the NLTK library in python, a function called "FreqDist" allows to give the frequency of the words within a text.

Before doing this all this data was submitted to the auxiliary functions created and intended for the purpose of cleaning the data in order to obtain a big document with all the public groups and with the clean data. After these operations of Data Cleaning it is then possible with the help of this library to obtain the frequency of the words as it can be seen in (Code Snippet 5.11).

```

1 listOfMostCommonWords = []
2
3 for z in range(0, len(listOfGroupDescriptions)):
4     listOfMostCommonWords.append(nltk.FreqDist(listOfGroupDescriptions[z]).
5     most_common(5))
6
7 for x in range(len(listOfMostCommonWords)):
8     print(listOfMostCommonWords[x])

```

Code Snippet 5.11: Most Common Words Code Snippet

The FreqDist class is used to encode “frequency distributions”, which count the number of times that each outcome of an experiment occurs.

For the purposes of this dissertation all the tourists were submitted to this process and only the five most common words were retrieved, although it will only be shown a handful of them in (Code Snippet 5.12).

```

1 [ ('magia', 1), ('de', 1), ('foto', 1), ('ipernity', 1), ('survivorstaller', 1)]
2 [ ('award', 18), ('post', 15), ('invitation', 13), ('admin', 10), ('world', 6)]
3 [ ('trams', 5), ('trains', 2), ('test', 2), ('beta', 2), ('alpha', 2)]
4 [ ('world', 15), ('photos', 11), ('travel', 7), ('best', 6), ('photography', 5)]
5 [ ('travel', 5), ('world', 5), ('photography', 3), ('mexico', 3), ('please', 2)]
6 [ ('flowers', 4), ('nature', 4), ('cats', 3), ('beauty', 3), ('flower', 3)]
7 [ ('art', 5), ('images', 4), ('historic', 3), ('church', 2), ('please', 2)]
8 [ ('aircraft', 14), ('street', 11), ('art', 9), ('british', 8), ('pub', 7)]
9 [ ('post', 7), ('favourite', 6), ('fav', 5), ('photography', 5), ('comment', 3)]
10 [ ('world', 6), ('post', 6), ('comment', 5), ('rio', 4), ('kids', 3)]
11 [ ('post', 72), ('award', 49), ('comment', 35), ('invite', 24), ('photos', 17)]
12 [ ('level', 44), ('post', 20), ('favourite', 18), ('comment', 15), ('photography',
    14)]
13 [ ('life', 72), ('museum', 62), ('castle', 53), ('palace', 51), ('fashion', 27)]
14 [ ('post', 9), ('portugal', 7), ('art', 5), ('artist', 4), ('photos', 4)]
15 [ ('futebol', 3), ('football', 2), ('soccer', 2), ('brazil', 1), ('messi', 1)]
16 [ ('art', 4), ('award', 3), ('photo', 3), ('painting', 2), ('world', 2)]
17 [ ('world', 4), ('best', 3), ('galaxy', 3), ('gallery', 2), ('night', 2)]
18 [ ('images', 3), ('photography', 3), ('around', 2), ('80s', 1), ('style', 1)]
19 []
20 [ ('post', 17), ('portugal', 13), ('award', 12), ('world', 7), ('comment', 7)]
21 []
22 [ ('post', 66), ('award', 37), ('comment', 35), ('world', 20), ('photos', 13)]
23 [ ('boats', 15), ('france', 11), ('photos', 10), ('world', 10), ('cars', 7)]
24 [ ('post', 9), ('portugal', 7), ('art', 5), ('quot', 4), ('photos', 4)]
25 [ ('best', 3), ('shot', 3), ('post', 2), ('comment', 2), ('animal', 2)]

```

Code Snippet 5.12: Result of Most Common Words Code Snippet

With the use of this it is possible to see a certain similarity between some words and start to understand what are the points of interest for a specific tourist, but even so it is not enough information to predict what are the places of interest.

So, with the help of the library "difflib" for text pre-processing purposes and using the list of MostCommonWords an attempt to attribute a similarity between words is introduced as it can be seen in (Code Snippet 5.13).

```

1 counter: int = 0
2 fullListWithAllSimilarWords = []
3
4 for z in range(len(listWithAllTheWordsInOneList)):
5     counter = counter + 1
6     tmpList = [listWithAllTheWordsInOneList[z]]
7     for counter in range(len(listWithAllTheWordsInOneList)):

```

```

8     score = difflib.SequenceMatcher(None, listWithAllTheWordsInOneList[z],
9                                     listWithAllTheWordsInOneList[counter]).
    ratio() * 100
10     if 75 < score < 99.9:
11         tmpList.append(listWithAllTheWordsInOneList[counter])
12
13     if counter == (len(listWithAllTheWordsInOneList) - 1):
14         if len(tmpList) > 2:
15             tmpList = list(dict.fromkeys(tmpList))
16             fullListWithAllSimilarWords.append(tmpList)
17
18 df = pd.DataFrame(fullListWithAllSimilarWords)
19 df = df.drop_duplicates()
20
21 print(df)

```

Code Snippet 5.13: Sequence Matcher Code Snippet

Regarding this (Code Snippet ??) a score between (75 and 99.9) is the range expectable to define if a word is similar to another. A word with 100% similarity is equal to another so it won't be considered. The 75% was defined with base on several dissertations that also define that if two words have over 75% similarity are considered of the same family. Obviously that the value 75% will be an arbitrary decision for this case.

The values expressed before are multiplied by 100 because initially the similarity score is a float comprehended in [0, 1] between two strings.

Basically this algorithm sums the sizes of all matched sequences returned that consist of triples describing matching subsequences and calculates their ratio.

At the end all this information is stored on a Dataframe which allows for a better visualization of the data retrieved.

With the use of this technique from this library it is possible to also see some reassemble between some words but it isn't still enough to truly obtain a person's place of interests and preferences as it can be seen in (Code Snippet 5.14).

	Col1	Col2	...	Col15	Col16
0	light	night	...	None	None
1	travel	traveling	...	None	None
2	limit	unlimit	...	None	None
3	least	pleas	...	None	None
4	portugal	portugues	...	None	None
5	photo	porto	...	None	None
6	moment	comment	...	None	None
7	porto	post	...	None	None
8	castle	castel	...	None	None
9	fav	fave	...	None	None
10	photographypaisaj	photographi	...	None	None
11	santiago	antiguo	...	None	None
12	unlimit	limit	...	None	None
13	vistaart	vista	...	None	None

```

16 14          fot          foto ...          None  None
17 15          ampwb        ampw  ...          None  None
18 16    beautiful    beauti ...          None  None

```

Code Snippet 5.14: DataFrame of Sequence Matcher

This Code Snippet will only show a part of the Code regarding this Dataframe, because it combines a lot of words so it also obtains a lot of results, most precisely over 200.

Even after all of this it isn't still possible to obtain the places of interest and preferences of a specific Tourists but it's getting close.

A final approach will be regarding the Latent Semantic Analysis, also known as LSA, which helps to discover hidden topics from given documents.

Discovering topics are beneficial for different purposes such as clustering documents, organizing online available content for information retrieval and recommendations.

Topic modeling is a text mining technique which gives techniques for identifying co-occurring keywords to summarize large collections of textual information. It helps in discovering hidden topics in the document, annotate the documents with these topics, and organize a large amount of unstructured data.

Topic Modeling consequently finds the concealed hidden themes from given records. It is an unsupervised text analytics algorithm that is utilized for discovering the group of words from the given document. These group of words addresses a topic. There is a possibility that, a single document can connect with various hidden topics.

There is a big difference between Text Classification and Topic Modeling since text classification is a supervised machine learning problem, in which a text document is classified into a set of classes, while, Topic Modeling is the process of discovering groups of co-occurring words in text documents. By other words, Topic Modeling can be used to solve the text classification problem. Topic Modeling will identify the topics present in a document while text classification classifies the text into a single class.

LSA learns latent topics by performing a matrix decomposition on the document-term matrix using Singular value decomposition. LSA is typically used as a dimension reduction or noise reducing technique. LSA will be implemented with the help of Gensim libraries.

Regarding the implementation of this algorithm it is also required for the data to be submitted to data cleaning operations in order to be able to process it.

In order to promote the LSA algorithm, an auxiliary function had to be created that converts the clean document into a list of Document term matrix that is then accepted by the algorithm as it can be seen in (Code Snippet 5.15).

```

1 def prepare_corpus(doc_clean):
2     """
3     #Input : clean document
4     #Purpose: create term dictionary of our corpus and Converting list of
5     documents (corpus) into Document Term Matrix
6     #Output : term dictionary and Document Term Matrix
7     """

```

```

7     # Creating the term dictionary of our corpus, where every unique term is
    assigned an index. dictionary = corpora.Dictionary(doc_clean)
8     dictionary = corpora.Dictionary(doc_clean)
9     # Converting list of documents (corpus) into Document Term Matrix using
    dictionary prepared above.
10    doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
11    return dictionary, doc_term_matrix

```

Code Snippet 5.15: Prepare Corpus Auxiliary Method

After the creation of this document term matrix and the dictionary of terms it is then possible to generate a model using the Gensim LSA Model as it can be seen in (Code Snippet 5.16).

```

1 def create_gensim_lsa_model(doc_clean, number_of_topics, words):
2     """
3     #Input  : clean document, number of topics and number of words associated
    with each topic
4     #Purpose: create LSA model using gensim
5     #Output : return LSA model
6     """
7     dictionary, doc_term_matrix = prepare_corpus(doc_clean) # generate LSA model
8     lsamodel = LsiModel(doc_term_matrix, num_topics=number_of_topics, id2word=
    dictionary) # train model
9     print(lsamodel.print_topics(num_topics=number_of_topics, num_words=words))
10
11    return lsamodel

```

Code Snippet 5.16: Create Gensim LSA Model

After all the above functions have been ran with some others that aren't describe but are helpful to either optimize the results by identifying an optimum amount of topics or to generate coherence scores or even to just simply plot, it is possible to obtain the number of topics and respective coherence values as it can be seen briefly in (Code Snippet 5.17).

```

1 [(0, '0.531*"post" + 0.426*"award" + 0.243*"comment" + 0.239*"photo" + 0.201*"
    world"'),
2 (1, '0.471*"train" + 0.316*"railway" + 0.270*"fav" + 0.223*"class" + 0.216*"
    level"'),
3 (2, '0.568*"level" + 0.348*"fav" + -0.195*"art" + -0.177*"world" + -0.139*"
    franchise"'),
4 (3, '-0.318*"post" + 0.317*"day" + 0.293*"view" + -0.264*"train" + 0.190*"window
    "'),
5 (4, '0.623*"day" + 0.457*"garden" + 0.215*"doll" + -0.178*"view" + -0.133*"
    flower"'),
6 (5, '-0.329*"favourite" + -0.294*"view" + 0.278*"invite" + -0.259*"post" +
    0.181*"photography"'),
7 (6, '0.505*"view" + -0.293*"museum" + -0.263*"art" + -0.205*"palace" + -0.204*"
    castle"'),
8 (7, '0.474*"life" + 0.430*"optic" + 0.403*"view" + 0.340*"second" + 0.185*"
    fashion"'),
9 (8, '-0.399*"city" + 0.272*"view" + 0.267*"street" + -0.236*"best" + -0.201*"
    award"'),

```

```

10 (9, '0.334*"level" + -0.323*"believe" + 0.238*"lmf" + -0.220*"quot" + 0.207*"art
    '''),
11 (10, '0.382*"level" + -0.357*"invite" + -0.231*"admin" + -0.211*"fave" + 0.198*"
    world''') and so on ... ]
12

```

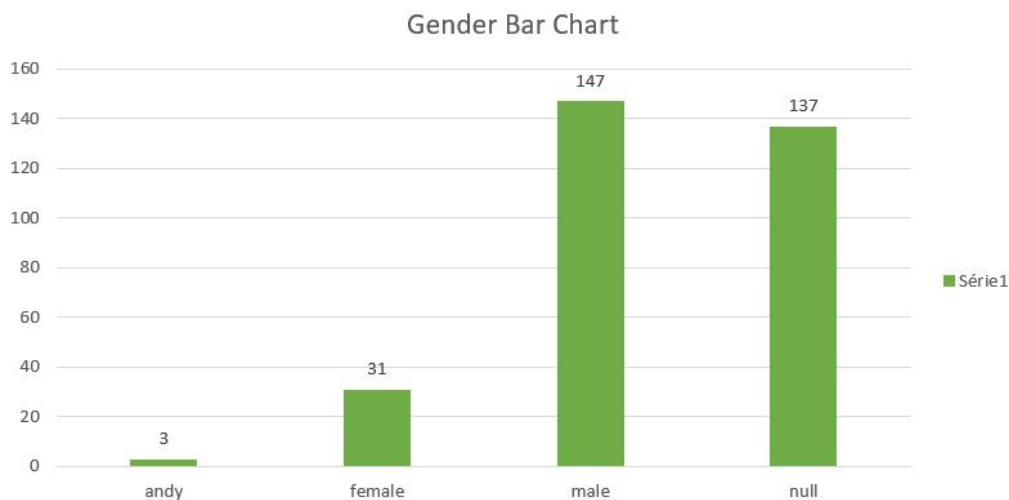
Code Snippet 5.17: Gensim LSA Model Results

After all these operations have been done it is also necessary to create some visual graphs that will allow to understand and give a better overview of the information that is possible to retrieve.

With the help of the data previously exported to CSV and also with the use of the database some graphs were made in order to provide a better insight of the information.

The first graph developed is regarding the Gender, this will be a bar chart and will demonstrate the distribution of gender between all the tourists that are present from the Tourist Database as it can be seen on (Figure 5.6).

Figure 5.6: Gender Bar Chart



As it is possible to see from the (Figure 5.6) the predominant sex is male, and it is also important to notice that a lot of tourists don't have comprehensible names in order for the gender detector to determine their gender. An important reference needs to be made to the word "andy" which relates to androgynous, as neither distinguishably masculine neither feminine.

Another bar chart developed is regarding Tourists' Location, which will give an insight about the origin location of the users and their predominant countries as it can be seen on (Figure 5.7).

A very important graph was made regarding the date where the first photo was taken, which can provide an insight of how long a user is on Flickr.com allowing to distinguish the experience of the users as it can be seen on (Figure 5.8).

The next two graphs are really important because they offer a different point of view about the big topics that are involved in order to distinguish the Tourists interest from places of Interest.

The first graph elaborated was the List of Groups that a user is involved as it can be seen on (Figure 5.9).

Figure 5.7: Location Bar Chart

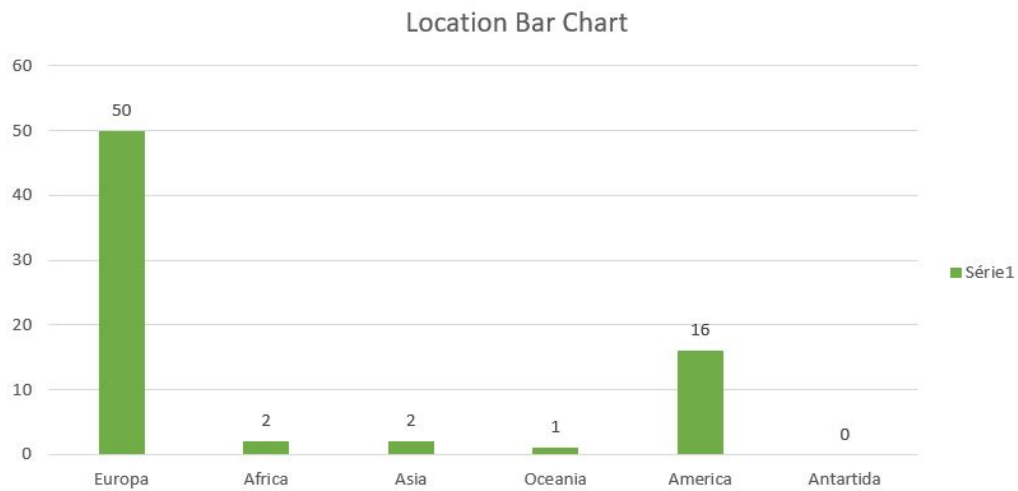
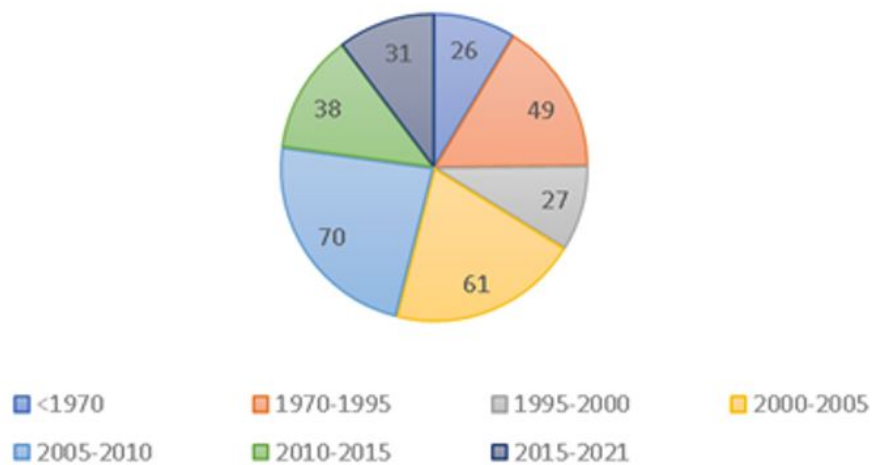


Figure 5.8: First photo Published Data Chart

firstPhotoPublished Date Chart



After analyzing the bar chart it is possible to see that there is not an even distribution among all the words that are presented, which relates to the importance of some complementary Topics.

Lastly there is a graph that is really important which relates to all the previous algorithms and is somehow a conclusion which allows to insert the Tourists according to their Preferences as it can be seen on (Figure 5.10).

This last graph was elaborated taking into consideration data refinement using photo descriptions and the previous algorithms.

Taking into consideration all these graphs and the Algorithms used is therefore possible to prepare the Data in order to provide the Machine Learning Algorithms with the best input possible

Figure 5.9: ListGroupUsers According to Topic Bar Chart

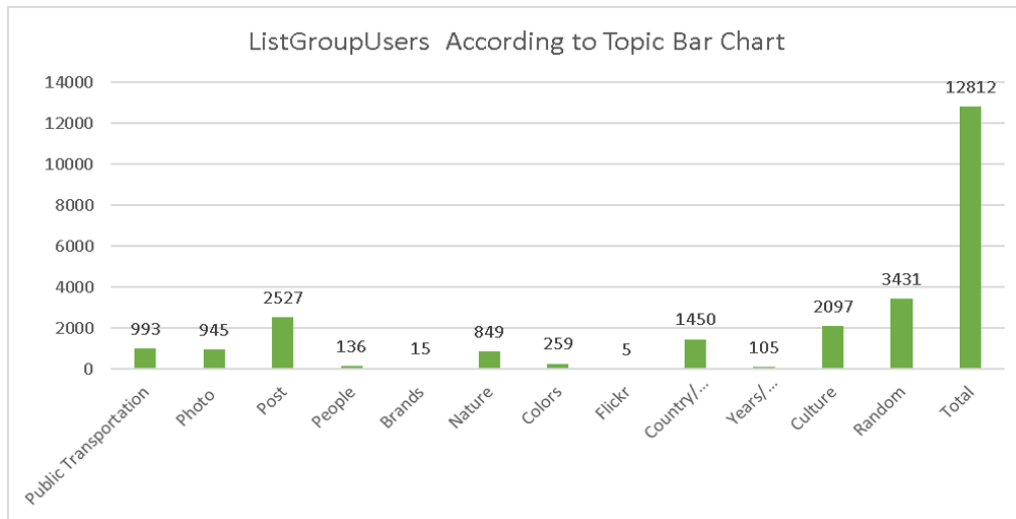
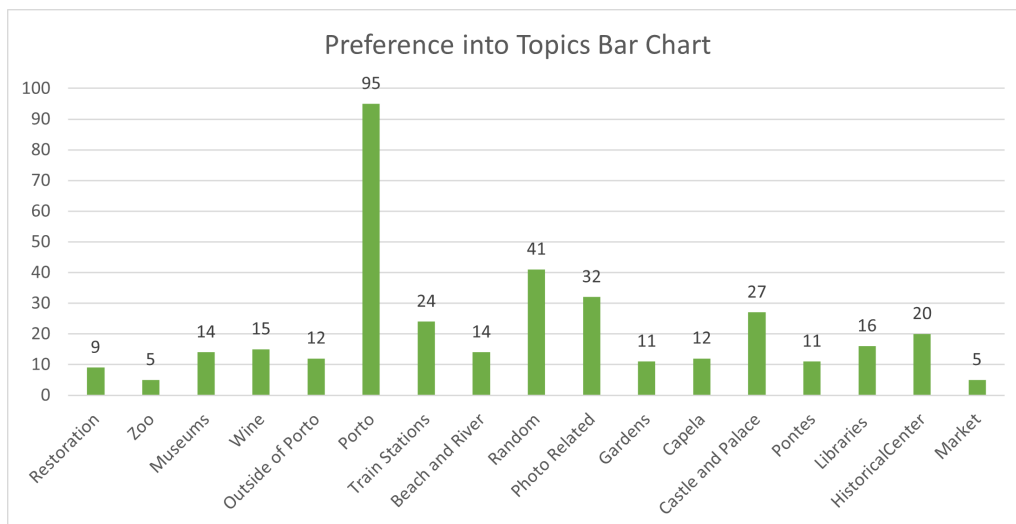


Figure 5.10: Preference into Topics Bar Chart



to obtain an even better outcome.

To sum up all this section, it is possible to differentiate certain Topics/Features and determine specific User's Interest or Preferences.

In short, there are close to twelve different types of Preferences which are: Restoration, Zoo, Museums, Wine, Porto, Beach, Garden, Chapel, Castle, Bridges, Library and Market.

Also two very important lists were formed from all this data preparation. The first list contains the user's preferences or, by other name, the user's interest and the other list contains the most common data regarding the user's public groups after it has been passed for all the previous processes.

5.4 Machine Learning

Unsupervised Learning is a sub-section of Machine Learning whose goal relies upon, trying to discover patterns in data sets or classifying the information into categories without being trained explicitly. This type of Machine Learning does not require the supervision of models by users and helps to discover unknown patterns in unlabeled datasets. This technique, unsupervised learning, helps data analysts to save time by providing algorithms that enhance the grouping and the investigation of data.

This leads to the process of clustering which divides categorized data into similar group or clusters ensuring that similar data points are identified and therefore grouped.

Though the use of clusters makes it easier to profile data according to each respective attributes allowing eventually to users sort the data into specific groups.

It also helps diminishing the size of the dataset taking into consideration that if there are too many clusters it means some of them are irrelevant due to the dataset being comprised of too many variables.

There are several types of clustering techniques in unsupervised machine learning but only a few will be brought up in this Article, which are, *Mean Shift*, *Affinity Propagation*, *DBSCAN* and *BIRCH*.

All these techniques are unsupervised clustering techniques and all the differences regarding input parameters, scalability, use cases and geometry will be explained on (Figure 5.11).

Figure 5.11: A comparison of different clustering algorithms

Method Name	Parameters	Scalability	Use Case	Geometry
Mean Shift	Bandwidth	Not scalable with n_samples	Many clusters; Uneven cluster size; Non-flat geometry; Inductive.	General purpose even cluster size
Affinity Propagation	Damping; Sample Preference.	Not scalable with n_samples	Many clusters; Uneven cluster size; Non-flat geometry; Inductive.	Graph distance (e.g. nearest-neighbor graph)
DBSCAN	Neighborhood Size	Very large n_samples; Medium n_clusters	Non-flat geometry; Uneven cluster size; Transductive.	Distances between nearest points
BIRCH	Branching factor; Threshold; Optional Global Clusterer	Large n_clusters and n_samples	Large dataset; Outlier removal; Data reduction; Inductive.	Distances between points

In a way as the name clustering implies it is anticipated that a suitable algorithm is capable of discovering structures on its own by exploring similarities or differences between singular data.

Regarding the next sub-sections, in each one of them a new Machine Learning Algorithm will be presented and two approaches will be shown with the respective description of the clusters

shown in the plots. The first approach will be using the list of Preferences and the other one will be with the list of Group Descriptions.

5.4.1 Mean Shift

5.4.1.1 Mean Shift Algorithm

Mean Shift procedure consists of two steps: the first one is the construction of a probability density model which reflects the underlying distribution of the data points allowing for a better representation of the clusters zones and points, the second step, is regarding the mapping of each point to the model of the density which is closest to the point. [86]

In a general sense, Mean-shift clustering: Given a set of data points, the algorithm iteratively assigns each data point towards the closest cluster centroid and the direction to the closest cluster centroid is determined by where most of the points nearby are at.

So, in each iteration each data point will move closer to where the most points are at, which is or will lead to the cluster center. When the algorithm stops, each point is assigned to a cluster.

Mean Shift algorithm assumed that n data points in a d -dimensional Euclidean space are independent, identically distributed samples drawn from a population with an unknown density function.

As it can be seen on (Figure 5.12). the algorithm Mean Shift for mode estimation and clustering starts with initializing a mode estimate to one of the observed data points, It then evaluates its mean shift vector merging the estimated clusters centers that are closed to a pre-defined distance. [87]

After all these steps are done in a repetitive way till all the data set has been achieved all the clusters that are small number of data points are eliminated, merging all the others having at the end an estimated number of clusters.

On a high level, Mean shift works as follows:

- Create a cluster for each data-point
- Each of the cluster is shifted towards a higher density region by shifting their corresponding centroid. This step is repeated until no shift yields a higher density.
- Assigning the data points to the cluster window in which they reside.

5.4.1.2 Determining the Parameters

The only important parameter to choose correctly is the bandwidth that determines the resulting clusters, which can look very different depending on the bandwidth introduced.

If the bandwidth chosen is too small, it will result in each point having its own cluster and in case the bandwidth is too big it will converge into one single big cluster.

Choosing the correct bandwidth is crucial and it is incredible hard to decide which one is the best and it is also possible, and this is the case, to estimate the bandwidth by the data introduced.

Figure 5.12: Pseudo-Code Mean Shift Algorithm for mode estimation and clustering

Algorithm MS algorithm for mode estimation and clustering

Input : Bandwidth h , profile function $g(x)$, threshold ϵ , and data set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $n \geq 2$.

Output: Estimated modes of the pdf, $\mathbf{y}_1^*, \dots, \mathbf{y}_k^*$, where k is the number of modes. The estimated modes are zero-dimensional principal curves that can be interpreted as centers of clusters.

```

begin
  for (i=1 to n) do
    Initialization:  $j \leftarrow 1$ ;  $\mathbf{y}_j \leftarrow \mathbf{x}_i$ ;
    /* Initialize the mode estimate sequence  $\mathbf{y}_1$  to be
       one of the observed data points. */
    repeat
      /* Evaluate the mean shift vector using (??) */
       $\mathbf{m}(\mathbf{y}_j) = \frac{\sum_{i=1}^n \mathbf{x}_i g(\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\|^2)}{\sum_{i=1}^n g(\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\|^2)} - \mathbf{y}_j$ ;
      /* Update the mode estimate using (5) */
       $\mathbf{y}_{j+1} = \mathbf{m}(\mathbf{y}_j) + \mathbf{y}_j$ ;
    until Until  $\|\mathbf{y}_{j+1} - \mathbf{y}_j\| < \epsilon$  for some predefined threshold  $\epsilon$ ;
     $\mathbf{y}_i^* \leftarrow \mathbf{y}_j$ ;
  Merge the estimated cluster centers that are closer than  $h$ ;
  Eliminate clusters that attract small number of data points;

```

There is also an important case to highlight which is the case of the bandwidth not being introduced in that case it assumes that it was not given, and the bandwidth will be estimated using "`sklearn.cluster.estimate_bandwidth`" regarding the dataset introduced. This function is only to use with the Mean Shift Algorithm.

5.4.1.3 Implementation of Mean Shift clustering Algorithm

Mean Shift will be implemented with the help of the library `scikit-learn` in order to discover clusters in data provided that is not separated without configuring the number of clusters.

Before the creation of the Mean Shift Algorithm it is necessary to clean the List with the Group Descriptions and convert it in a single string in order to be accepted by the algorithm. This string represents basically a document.

It is also necessary to create a transform function as it expressed in line 4 "`vectorizer = CountVectorizer()`" which will be responsible to tokenize and build vocabulary. This function will encode the document in a vector so that the Algorithm Mean shift can process it.

Finally after this the Mean Shift object is created and is fitted with the vector of Preferences.

The first mean shift approach is regarding the list of preferences of a Tourist as it can be seen in (Code Snippet 5.18).

```

1 stringListPreferences = cleanString((' '.join([str(elem) for elem in
      listOfPreferences])).replace(',',' '))

```

```

2 stringListPreferences = [stringListPreferences]
3
4 vectorizer = CountVectorizer()
5 vectorizer.fit(stringListPreferences)
6 vectorPreferences = vectorizer.transform(stringListPreferences)
7
8 ms = MeanShift()
9 ms.fit(vectorPreferences)
10 cluster_centers = ms.cluster_centers_
11
12 fig = plt.figure()
13 ax = fig.add_subplot(111, projection='3d')
14 ax.scatter(vectorPreferences[:, 0], vectorPreferences[:, 1], vectorPreferences[:,
15           2], marker='o')
16 # draw and each value represents a dot
17 ax.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
18           cluster_centers[:, 2], marker='o', color='blue', linewidth=5,
19           zorder=10)
20 # Zorder how close the points are to observer
21 # linewidth -> line thickness
22 plt.title('Estimated number of clusters List Preferences: %d' % len(
23           cluster_centers))
24 plt.show()

```

Code Snippet 5.18: First Mean Shift Implementation

After the creation of the Mean Shift object with the appropriate vector it is then possible to plot the corresponding figure.

A figure object is created with the help of the *matplotlib* library to plot the data points and centroids in a 3D Graph. An auxiliary part of defining the axis is made and the plot will be drawn in a 1x1 grid in the first subplot with a 3d projection.

The points will be drawn and will be represented by dots. Everything will be done in an iterative way slicing all the rows from the columns basically, using a methodology like the following *[all rows, column[0,1,2]]*.

This approach results in a very good result as it can be seen on (Figure 5.13).

In this plot it is possible to actually see thirteen clusters which is really close to twelve different points of interest.

To obtain a better overview it is possible to describe the coordinates of those clusters and to actually identify them and place them accordingly. It is also possible to identify what was the prevailing gender according to the identification of the cluster and also what other points of interest are connected to that Cluster as it can be seen on (Figure 5.14).

The most prevailing cluster of all presented previously is Porto obtaining 87 data points close to that cluster.

After this list has been thoroughly reviewed and a full description of it has been made, it is possible to move on to the second vector to be analyzed.

Figure 5.13: First Mean Shift Plot

Estimated number of clusters List Preferences: 13

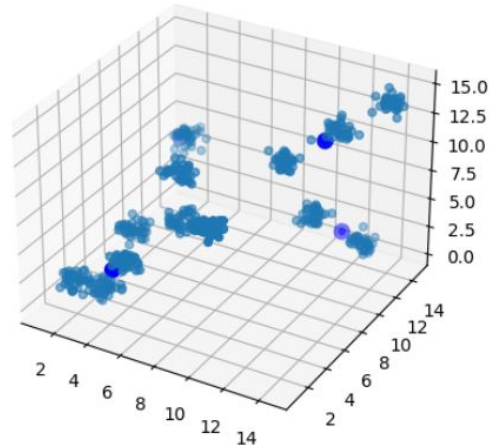


Figure 5.14: Description of the Clusters of the First Mean Shift Algorithm

Cluster Coordinate	Identification	Gender	Other Points of Interest
[1.15342;4.39992;1.28204]	Castle	Male = 45% Female = 15% Andy = 5% Null = 35%	Palace = 30% Chapel = 15%
[1.93213;4.50134;1.78342]	Porto	Male = 30% Female = 30% Andy = 5% Null = 35%	Beach = 45% Monastery = 25%
[1.74662;7.77231;1.54261]	Palace	Male = 30% Female = 25% Andy = 0% Null = 45%	Porto = 45% Castle = 15%
[0.12025;11.12435;1.99234]	Beach	Male = 35% Female = 20% Andy = 0% Null = 45%	Wine = 25% Monastery = 10%
[2.01775;13.00242;2.38312]	Chapel	Male = 40% Female = 15% Andy = 0% Null = 45%	Castle = 45% Monastery = 15%
[3.82853;12.99921;3.42121]	Monastery	Male = 30% Female = 25% Andy = 0% Null = 45%	Porto = 35% Chapel = 30%
[1.03254;15.22156;3.82934]	Library	Male = 25% Female = 50% Andy = 0% Null = 25%	Porto = 25% Monastery = 20%
[0.94245;14.93482;5.23132]	Market	Male = 35% Female = 15% Andy = 0% Null = 50%	Porto = 30% Wine = 25%
[7.74924;13.87212;4.87932]	Restoration	Male = 15% Female = 20% Andy = 0% Null = 65%	Porto = 15% Wine = 25%
[8.48713;14.25677;1.99897]	Museums	Male = 35% Female = 15% Andy = 0% Null = 50%	Porto = 20% Restoration = 10%
[11.87735;15.73569;1.11112]	Bridge	Male = 25% Female = 25% Andy = 0% Null = 50%	Porto = 25% Museum = 10%
[10.00026;15.26161;8.95334]	Wine	Male = 55% Female = 15% Andy = 0% Null = 30%	Porto = 25% Bridge = 15%
[13.55555;14.68666;11.563454]	Garden	Male = 25% Female = 25% Andy = 0% Null = 50%	Porto = 35% Chapel = 15%

Also another implementation of Mean Shift was introduced regarding the vector List Group Descriptions as it can be seen in (Code Snippet 5.19).

```

1 stringListGroupDescriptions = cleanString((' '.join([str(elem) for elem in
    fullListWithAllSimilarWords])).replace(',', ' '))
2 stringListGroupDescriptions = [stringListGroupDescriptions]
3
4 vectorizer = CountVectorizer()
5 vectorizer.fit(stringListGroupDescriptions)
6 vectorGroupDescriptions = vectorizer.transform(stringListGroupDescriptions)
7
8 ms = MeanShift()
9 ms.fit(vectorGroupDescriptions)
10 cluster_centers = ms.cluster_centers_

```

```

11
12 fig = plt.figure()
13 ax = fig.add_subplot(111, projection='3d')
14 ax.scatter(vectorGroupDescriptions[:, 0], vectorGroupDescriptions[:, 1],
15           vectorGroupDescriptions[:, 2], marker='o')
16 ax.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
17           cluster_centers[:, 2], marker='o', color='blue', linewidth=5,
18           zorder=10)
19 plt.title('Estimated number of clusters List Group Descriptions: %d' % len(
20           cluster_centers))
20 plt.show()

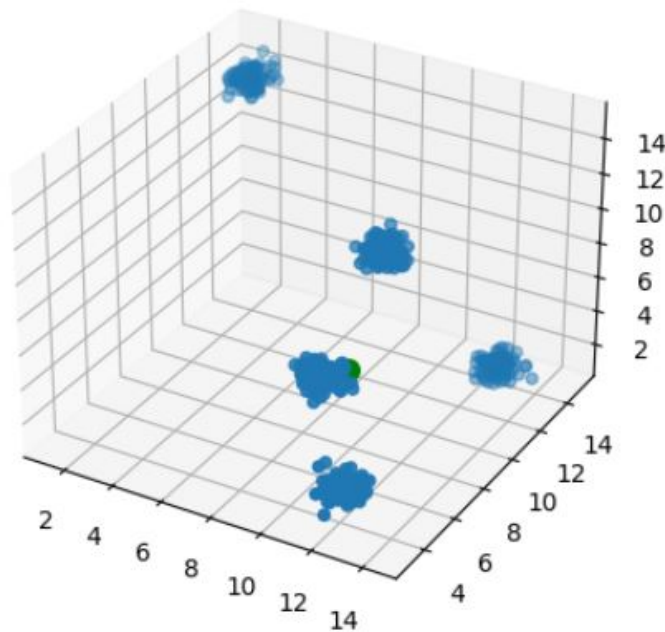
```

Code Snippet 5.19: Second Mean Shift Implementation

This approach results in different result as it can be seen on (Figure 5.15).

Figure 5.15: Mean Shift Plot of the Second Mean Shift Algorithm

Estimated number of clusters Vector List Group Descriptions : 5



In order to distinguish this clusters from the other ones expressed previously it was also possible to illustrate them regarding their cluster Coordinates and identifications as it can be seen on (Figure 5.16).

5.4.1.4 Performance Evaluation

Mean shift is a simple cluster method that works very well on spherical-shaped data, because it automatically selects the number of clusters contrary to other algorithms. Although this algorithm

Figure 5.16: Description of the Clusters of the Second Mean Shift Algorithm

Cluster Coordinate	Identification	Gender	Other Points of Interest
[11.65785;5.86471;1.23874]	Post	Male = 40% Female = 30 % Andy = 0% Null = 30 %	Porto = 45% View = 25%
[7.024158;11.56317;0.52378]	Train	Male = 45% Female = 20% Andy = 2,5% Null = 32,5%	Porto = 15% Garden = 20%
[12.57482;15.89532;0.21486]	Porto	Male = 30% Female = 25% Andy = 0% Null = 45%	Porto = 45% Castle = 15%
[7.15327;14.99786;5.45239]	Garden	Male = 55% Female = 15% Andy = 0% Null = 30%	View = 50% Train = 25%
[1.52479;14.85234;13.86324]	View	Male = 15% Female = 10% Andy = 0% Null = 45%	Post = 35% Garden = 10%

is computationally expensive and often converges too slowly to be practical on large scale applications it is not dependent on the initialization since at the start each point can be a cluster and it decides what is the best towards the data set introduced.

Since this is an unsupervised clustering algorithm there are no common quantitative measure of the classification accuracy, therefore the accuracy is evaluated by visual inspection.

It is also important to enhance its advantages and disadvantages as it can be seen on (Figure 5.17).

Figure 5.17: Mean Shift Advantages and Disadvantages Table

Advantages of Mean Shift	Disadvantages of Mean Shift
<ul style="list-style-type: none"> Automatically chooses the cluster numbers. 	<ul style="list-style-type: none"> The performance is good in one dimension and scalability issue in higher dimension.
<ul style="list-style-type: none"> Automatically finds the shape of the clusters. 	<ul style="list-style-type: none"> The computation of the algorithm is high.
<ul style="list-style-type: none"> Good performance with outliers 	

A very important is also the Purity ratio obtained when using Mean Shift in both approaches. It was possible to calculate using the ratio obtained from dividing the correct number of documents by the total amount of documents, which gave as a result the following values as it can be seen in (Table 5.2).

Table 5.2: Purity Measure Mean Shift

Score Purity Mean Shift ListGroupDescriptions	0.611
Score Purity Mean Shift List Preferences	0.724

5.4.2 Affinity Propagation

5.4.2.1 Affinity Propagation Algorithm

Regarding the algorithm it operates on three different matrices: A similarity matrix(s), a responsibility matrix(r) and an availability matrix(a). All the results are stored in a criterion matrix(c) it

is also important to notice that all these matrices are updated iteratively and can be represented by four equations as it can be seen on (Figure 5.18).

Figure 5.18: Equations Affinity Propagation

$$r(i, k) \leftarrow s(i, k) - \max_{k' \text{ such that } k' \neq k} \{a(i, k') + s(i, k')\}, \quad (1)$$

$$a(k, k) \leftarrow \sum_{i' \text{ such that } i' \neq k} \max\{0, r(i', k)\}, \quad (2)$$

$$a(i, k) \leftarrow \min \left\{ 0, r(k, k) + \sum_{i' \text{ such that } i' \in \{i, k\}} \max\{0, r(i', k)\} \right\} \quad (3)$$

$$c(i, k) \leftarrow r(i, k) + a(i, k). \quad (4)$$

Similarity Matrix(s) provides information regarding the similarity between any instance, and this is obtained by summing the squares of the differences between variables that make up the items. Therefore, the greater the distance between the two points, smaller the similarity becomes between them.

The next step of the algorithm is to construct an availability matrix that sets all the elements to Zero. After this is done, it is applied the equation (1) to compute the responsibility matrix. This matrix refers and quantifies how well-suited elements will be to an exemplar. The next two equations are used to update iteratively the diagonal and off-diagonal elements of the availability matrix. [88]

The final step is to apply the equation (4) and give rise to the criterion matrix. The columns with the highest criterion value for each row identify the exemplar for the item of that row. Rows that share the same exemplar are in the same cluster.

There are also some important steps to be analyzed which are the identification of outliers, and to do this part all the signs of the items in the similarity matrix are reserved for them to become positive.

5.4.2.2 Determining the Parameters

Affinity Propagation takes as input a collection of real valued similarities between data points, but it mainly base are two important parameters which are the preference and the damping factor.

The first one controls how many exemplars or prototypes are used and the last one damps the responsibility and availability of messages to avoid numerical oscillations when these messages are updates.

The value of damping factor can also not be introduced and is assumed by default as 0.5 to which represents a value that is compared to the incoming values, weight wise).

5.4.2.3 Implementation of Affinity Propagation clustering Algorithm

Affinity Propagation at a more detailed level works on messages propagated between sample data with each other having scores based on their computation level. The messages score are updated after each iteration, and that's how the true clusters start to be formed.

This Algorithm works "gossiping" around updating itself through small and smooth updates in order to evaluate better individual samples across time.

Affinity Propagation will be implemented with the help of the library *scikit-learn* in order to discover similarities in the data provided.

First the object Affinity Propagation has to be created following with the data fitting in order to be able to populate the Algorithm. The next lines are based on the derivation of the characteristics such as the exemplars and labels and by consequence the number of clusters as it can be seen in (Code Snippet 5.20).

```

1 af = AffinityPropagation(preference=-50)
2 af.fit(vectorGroupDescriptions)
3
4 cluster_centers_indices = af.cluster_centers_indices_
5 labels = af.labels_
6
7 n_clusters_ = len(cluster_centers_indices)
8
9 colors = cycle('bgrcmkykbgrcmkykbgrcmkykbgrcmkyk')
10 for k, col in zip(range(n_clusters_), colors):
11     class_members = labels == k
12     cluster_center = vectorGroupDescriptions[cluster_centers_indices[k]]
13     plt.plot(vectorGroupDescriptions[class_members, 0], vectorGroupDescriptions[
14         class_members, 1], col + '.')
15     plt.plot(cluster_center[0], cluster_center[1], 'o', markerfacecolor=col,
16         markeredgecolor='k', markersize=14)
17     for x in vectorGroupDescriptions[class_members]:
18         plt.plot([cluster_center[0], x[0]], [cluster_center[1], x[1]], col)
19 plt.title('Estimated number clusters Affinity Propagation ListPreferences : %d' %
20     len(n_clusters_))
21 plt.show()

```

Code Snippet 5.20: First Affinity Propagation Implementation

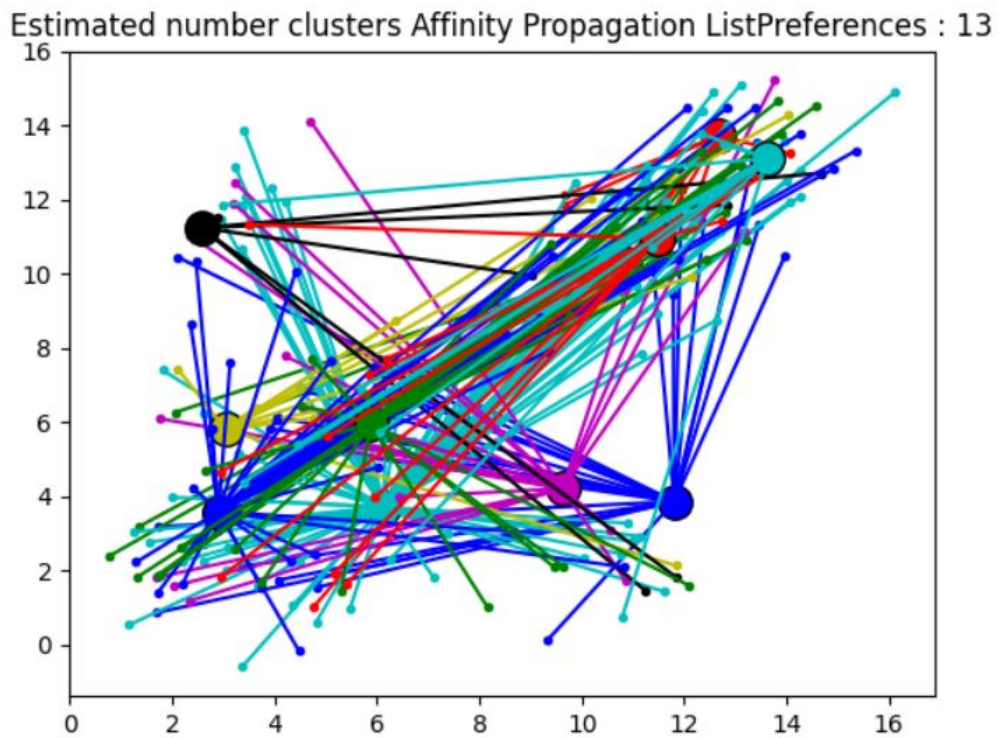
Also a cycle of colors is made to represent each affinity in a different color so things do not get mixed up.

The rest of the code is regarding the Plot Result and is mostly directed towards drawing the plot.

A visual representation of this code snippet can be seen on (Figure 5.19).

To obtain a better overview it is possible to describe the coordinates of those clusters and to actually identify them and place them accordingly. It is also possible to identify what was the

Figure 5.19: Affinity Propagation Plot



prevailing gender according to the identification of the cluster and also what other points of interest are connected to that Cluster as it can be seen on (Figure 5.20).

Figure 5.20: Description of the Clusters First Affinity Propagation Plot

Cluster Coordinate	Identification	Gender	Other Points of Interest
[3.12586;3.98526]	Castle	Male = 30% Female = 25% Andy = 5% Null = 40%	Porto = 10% Bridge = 15%
[6.00074;3.89741]	Porto	Male = 35% Female = 20% Andy = 0% Null = 45%	Beach = 30% Wine = 20%
[3.31478;5.74589]	Palace	Male = 30% Female = 15% Andy = 5% Null = 45%	Porto = 45% Castle = 15%
[6.14782;5.87425]	Beach	Male = 30% Female = 10% Andy = 0% Null = 60%	Porto = 25% Restoration = 10%
[9.87462;5.42533]	Chapel	Male = 30% Female = 20% Andy = 0% Null = 50%	Porto = 45% Monastery = 15%
[11.82216;4.22374]	Monastery	Male = 25% Female = 25% Andy = 0% Null = 50%	Porto = 35% Chapel = 10%
[11.56283;11.68742]	Library	Male = 20% Female = 30% Andy = 5% Null = 45%	Porto = 15% Museums = 5%
[13.62218;13.22476]	Market	Male = 25% Female = 20% Andy = 0% Null = 55%	Porto = 30% Wine = 25%
[2.23485;11.65891]	Restoration	Male = 15% Female = 10% Andy = 0% Null = 75%	Porto = 15% Market = 25%
[12.18543;13.98543]	Museums	Male = 40% Female = 10% Andy = 0% Null = 50%	Porto = 20% Palace = 10%
[7.25492;5.67482]	Bridge	Male = 35% Female = 30% Andy = 0% Null = 35%	Porto = 25% Museum = 20%
[10.54128;11.37412]	Wine	Male = 45% Female = 25% Andy = 0% Null = 30%	Porto = 30% Restoration = 15%
[8.33417;8.11249]	Garden	Male = 10% Female = 10% Andy = 0% Null = 80%	Porto = 25% Palace = 10%

After this list has been thoroughly reviewed and a full description of it has been made, it is possible to move on to the second vector to be analyzed.

Also another implementation of Mean Shift was introduced regarding the vector List GroupDescriptions as it can be seen in (Code Snippet 5.20).

```

1 af2 = AffinityPropagation(preference=-50)
2 af2.fit(vectorListPreferences)
3 cluster_centers_indices2 = af2.cluster_centers_indices_
4 labels2 = af2.labels_
5
6 n_clusters_2 = len(cluster_centers_indices2)
7
8 colors = cycle('bgrcmykbgrcmykbgrcmykbgrcmyk')
9 for k, col in zip(range(n_clusters_2), colors):
10     class_members2 = labels2 == k
11     cluster_center2 = vectorListPreferences[cluster_centers_indices2[k]]
12     plt.plot(vectorListPreferences[class_members2, 0], vectorListPreferences[
13         class_members2, 1], col + '.')
14     plt.plot(cluster_center2[0], cluster_center2[1], 'o', markerfacecolor=col,
15             markeredgecolor='k', markersize=14)
16     for x in vectorListPreferences[class_members2]:
17         plt.plot([cluster_center2[0], x[0]], [cluster_center2[1], x[1]], col)
18 plt.title('Estimated number clusters Affinity Propagation ListGroupDescriptions:
19           %d' % len(cluster_centers_indices2))
20 plt.show()

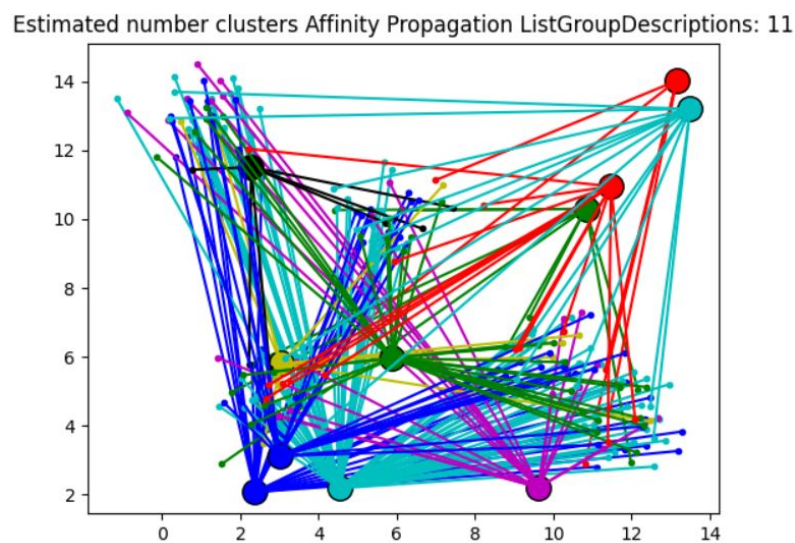
```

Code Snippet 5.21: Second Affinity Propagation Implementation

Since it was previously explained all the methodology behind it, most of the operations are repetitive from the first implementation of the Algorithm Affinity Propagation.

This second implementation generates a different plot than the first Affinity Propagation plot as it can be seen on (Figure 5.21).

Figure 5.21: Second Affinity Propagation Plot



By its instance it generates also a different description of the clusters since the results are very different than the first Plot as it can be seen on (Figure 5.22).

Figure 5.22: Description of the Clusters Second Affinity Propagation Plot

Cluster Coordinate	Identification	Gender	Other Points of Interest
[2.24856;11.68423]	Post	Male = 25% Female = 15% Andy = 5% Null = 55%	Porto = 20% Train = 15%
[2.21487;1.99875]	Train	Male = 30% Female = 20% Andy = 0% Null = 50%	Beach = 35% Porto = 5%
[4.31482;2.23478]	Porto	Male = 40% Female = 15% Andy = 0% Null = 45%	View = 35% Photo = 15%
[3.21783;3.87846]	Garden	Male = 15% Female = 15% Andy = 0% Null = 70%	View = 20% Porto = 10%
[9.88951;2.23589]	View	Male = 60% Female = 10% Andy = 0% Null = 30%	Porto = 25% Garden = 15%
[5.99978;6.00015]	Photo	Male = 50% Female = 20% Andy = 5% Null = 25%	Porto = 30% Chapel = 5%
[10.98742;10.87523]	Museum	Male = 45% Female = 20% Andy = 0% Null = 35%	Porto = 15% Art = 10%
[13.78526;13.55621]	Palace	Male = 35% Female = 15% Andy = 0% Null = 50%	Porto = 20% Museum = 15%
[5.21259;6.17986]	World	Male = 20% Female = 15% Andy = 0% Null = 65%	Porto = 30% Art = 10%
[11.42358;10.58742]	Favourite	Male = 35% Female = 25% Andy = 0% Null = 40%	Porto = 10% View = 10%
[13.45271;13.95712]	Art	Male = 35% Female = 25% Andy = 5% Null = 35%	Porto = 25% Museum = 10%

5.4.2.4 Performance Evaluation

Affinity propagation is a high speed, flexible and low error clustering algorithm that identifies clusters and outliers respectively. Since this is an unsupervised clustering algorithm there are no common quantitative measure of the classification accuracy, therefore the accuracy is evaluated by visual inspection.

It is also important to enhance its advantages and disadvantages as it can be seen on (Figure 5.23).

Figure 5.23: Affinity Propagation Advantages and Disadvantages Table

Advantages of Affinity Propagation:	Disadvantages of Affinity Propagation:
<ul style="list-style-type: none"> • Low time complexity 	<ul style="list-style-type: none"> • Not suitable for non-convex data
<ul style="list-style-type: none"> • High computing efficiency 	<ul style="list-style-type: none"> • Relatively sensitive to the outliers
<ul style="list-style-type: none"> • Easy to code 	<ul style="list-style-type: none"> • Easily drawn into local optimal
	<ul style="list-style-type: none"> • The number of clusters needed to be preset
	<ul style="list-style-type: none"> • The clustering result sensitive to the number of clusters

A very important is also the Purity ratio obtained when using Affinity Propagation in both approaches. It was possible to calculate using the ratio obtained from dividing the correct number of documents by the total amount of documents, which gave as a result the following values as it can be seen in (Table 5.3).

Table 5.3: Purity Measure Affinity Propagation

Score Purity Affinity Propagation ListGroupDescriptions	0.459
Score Purity Affinity Propagation List Preferences	0.684

5.4.3 DBSCAN

5.4.3.1 DBSCAN Algorithm

The main purpose of this deterministic algorithm is to find clusters, so it starts with an arbitrary point and retrieves all the points reached within the same density in a distance of “*eps*”. This arbitrary point is a core point if this procedure yields a cluster with the *minPoints* or more. In case that the arbitrary point chosen is a border point, no points are density reachable and DBSCAN algorithm must visit the next point in the dataset.

The (Figure 5.24) presents a basic version of the algorithm DBSCAN omitting details of data types and also omitting information about the clusters. [89]

Figure 5.24: DBSCAN Pseudo Code - setOfPoints

```

DBSCAN (SetOfPoints, Eps, MinPts)
// SetOfPoints is UNCLASSIFIED
ClusterId := nextId(NOISE);
FOR i FROM 1 TO SetOfPoints.size DO
  Point := SetOfPoints.get(i);
  IF Point.ClId = UNCLASSIFIED THEN
    IF ExpandCluster(SetOfPoints, Point,
      ClusterId, Eps, MinPts) THEN
      ClusterId := nextId(ClusterId)
    END IF
  END IF
END FOR
END; // DBSCAN

```

Regarding some variables expressed, “*SetOfPoints*” means either the whole database or a discovered cluster from a previous run. The other two variables are the global parameters, respectively, “*eps*” meaning Epsilon, the measure distance, and “*minPts*” the minimum number of points to form a core point. However, the most important function used by DBSCAN is “*expandCluster*” which is presented on Figure 5.25:

A call to *setOfPoints.regionQuery* is made which returns the points in *Eps-neighborhood* distance as a list of points and is saved under the name of seeds. After this step, it is mandatory to compare to the initial parameter “*MinPts*” and if it is above the value, it means that there exists a core Point but if it falls below, it means no core Point.

Figure 5.25: DBSCAN Pseudo Code - expandCluster

```

ExpandCluster(SetOfPoints, Point, ClId, Eps,
              MinPts) : Boolean;
seeds:=SetOfPoints.regionQuery(Point,Eps);
IF seeds.size<MinPts THEN // no core point
  SetOfPoint.changeClId(Point,NOISE);
  RETURN False;
ELSE // all points in seeds are density-
  // reachable from Point
  SetOfPoints.changeClIds(seeds,ClId);
  seeds.delete(Point);
  WHILE seeds <> Empty DO
    currentP := seeds.first();
    result := SetOfPoints.regionQuery(currentP,
                                      Eps);

    IF result.size >= MinPts THEN
      FOR i FROM 1 TO result.size DO
        resultP := result.get(i);
        IF resultP.ClId
          IN {UNCLASSIFIED, NOISE} THEN
          IF resultP.ClId = UNCLASSIFIED THEN
            seeds.append(resultP);
          END IF;
          SetOfPoints.changeClId(resultP,ClId);
        END IF; // UNCLASSIFIED or NOISE
      END FOR;
    END IF; // result.size >= MinPts
    seeds.delete(currentP);
  END WHILE; // seeds <> Empty
  RETURN True;
END IF
END; // ExpandCluster

```

If it falls below, they are initially marked as Noise Points through the use of the function “*changeCLIds*”, although they can be changed later to border points of a cluster if they contain at least one density-reachable point.

If *seeds.size* actually meets the desired value and the if statement returns True, then all the points in the seeds vector are density reachable from the initial point leading to the inner part of this procedure. This next part fetches one element at a time from the list and evaluates it and checks its value, a recursive call is made to “*setOfPoints.regionQuery*” because it is always required to see the neighborhood points at the “*eps*” distance from a specific point.

It is important to keep in mind that this is a very detailed explanation of how DBSCAN works. Regarding the actual code to this machine learning project an easier approach will be implemented with the help of scikit-learn library.

5.4.3.2 Determining the Parameters

DBSCAN in order to work needs two parameters, “*Eps*” and “*minPts*”. According to experiments and research done it indicates that *k-dist* graphs for $k > 4$ are not different from the 4-dist graphs and so they need considerably more computation, therefore the parameter *MinPts* in most cases is set to four for all databases in case of two-dimensional data.

Regarding the other parameter “*Eps*” of DBSCAN, it is necessary to compute the system and display the 4-dist graph for the dataset and if possible to estimate the percentage of noise, this percentage is entered, and the system should try to reach a “*Eps*” value regarding this initial noise value.

5.4.3.3 Implementation of DBSCAN clustering Algorithm

DBSCAN instead of being a partition or hierarchical clustering-based Algorithm is instead a Density based Algorithm having more efficient techniques when it comes to arbitrary shaped cluster or detecting outliers.

All the data points in a plot can easily be grouped in random shapes and without caring about outliers, although DBSCAN fights this, identifying clusters in high-density regions and outliers.

DBSCAN is known for standing out in density-based spatial clustering of applications with noise defending a principal idea that a point belongs to a cluster if it is close to many points from that cluster.

As it was identified previously there are two key parameters of the Algorithm DBSCAN, “*eps*” that represents the distance of the neighborhoods of a cluster and “*minPts*”, the minimum number of data points to define a cluster. Also, it is important to enhance that two points are neighbors if their distance is less or equal to “*eps*”.

Regarding these two parameters, points can also be classified as **core points**, **border points** or **outliers**, each own with its own respective meaning.

The value chosen for “*minPts*” is going to be 4 meaning that for a core point to exist there has to be at least 4 points within the surrounding area with radius of “*eps*”. Border points, in this case, will be reachable (being in the surrounding area) from a core point and must have less than 4 points within their neighborhood. Outliers are not core points and cannot be reached from a core point.

To determine the optimal value for “*eps*”, it was necessary to compute the distance for every point in the dataset of its 4th nearest neighbor, because “*minPts*” = 4. After all the distances have been computed they are sorted in increasing order and a plot is made regarding them. The optimal distance value for *eps* will be at the elbow of the plot drawn, which has the value of “0.3”. So, in this case, *eps* = 0.3.

A starting point is selected at random, and its corresponding neighborhood is determined using radius “*eps*”. If there are at least “*minPts*” in the surrounding area the point is then market as a Core Point and it starts to form a cluster else, it forms a noise point. Once a cluster starts to form all the points inside the cluster become a part of it. After this point has been visited the next step is to choose, again randomly, another point and fully visit it. This step is repeated till there are no more points to visit.

The library used to implement the DBSCAN algorithm will be *Scikit-learn* that contains a lot of tools to help with Machine Learning Algorithms.

A DBSCAN object is created with the corresponding two parameters shown above. After the creation of this object the data is fitted within it as it can be seen in (Code Snippet 5.22).

```
1 vectorPreferences = StandardScaler().fit_transform(vectorPreferences)
2
3 db = DBSCAN(eps=0.3, min_samples=4)
4 db.fit(vectorPreferences)
5
```

```

6 core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
7 core_samples_mask[db.core_sample_indices_] = True
8 labels = db.labels_
9
10 n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
11 n_noise_ = list(labels).count(-1)
12
13 unique_labels = set(labels)
14 colors = [plt.cm.Spectral(each)
15           for each in np.linspace(0, 1, len(unique_labels))]
16 for k, col in zip(unique_labels, colors):
17     if k == -1:
18         # Black used for noise.
19         col = [0, 0, 0, 1]
20
21     class_member_mask = (labels == k)
22
23     xy = vectorPreferences[class_member_mask & core_samples_mask]
24     plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
25            markeredgecolor='k', markersize=14)
26
27     xy = vectorPreferences[class_member_mask & ~core_samples_mask]
28     plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
29            markeredgecolor='k', markersize=6)
30
31
32 plt.title('Estimated number of clusters DBSCAN ListPreferences: %d' % n_clusters_
33         )
34 plt.show()

```

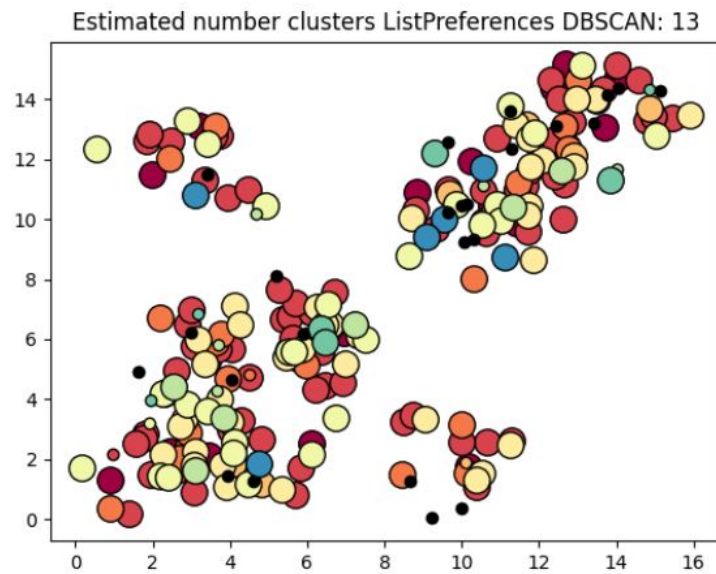
Code Snippet 5.22: First DBSCAN clustering Algorithm Implementation

After DBSCAN has been computed some operations are important to enhance on this code snippet. The operation on line six serves as a purpose to return an array of zeros with the same shape and type as a given array, dtype will override the data type of the result. The next line, seven, "core_sample_indices" is regarding the attributes and it is index of core samples.

Line ten is also very important because it counts the number of clusters in labels ignoring noise if it is present. When defining the set of colors, the color black will be removed because it will be used for another purpose which will be to identify noise. The rest of the code is to plot the following graph to decide which format string will be chosen as an abbreviation for quickly setting basic line properties. Some attributes within the plot function of "matplotlib.pyplot.plot" are used with the intent of defining properties, as for example. "MarketFaceColor" which is responsible for changing the plotted lines to fill in the markers with the same color as the default marker edge color.

This algorithm implemented allows to generate the following Plot that is represented on (Figure 5.26).

Figure 5.26: DBSCAN First Implementation Plot



In this plot drawn it's a bit tricky to see all the thirteen cluster so in order to help visualize the data being expressed a table is created in order to describe the coordinates of the clusters and the information attached to them.(Figure 5.27).

Figure 5.27: Description of the Clusters First DBSCAN Plot

Cluster Coordinate	Identification	Gender	Other Points of Interest
[1.51423;2.41235]	Castle	Male = 40% Female = 15% Andy = 5% Null = 45%	Palace = 15% Chapel = 15%
[2.43244;4.42344]	Porto	Male = 25% Female = 30% Andy = 5% Null = 45%	Beach = 40% Monastery = 25%
[3.74662;7.77231]	Palace	Male = 20% Female = 20% Andy = 0% Null = 60%	Porto = 40% Castle = 15%
[3.12025;11.12435]	Beach	Male = 35% Female = 15% Andy = 0% Null = 55%	Wine = 20% Monastery = 10%
[4.51775;2.51354]	Chapel	Male = 45% Female = 10% Andy = 0% Null = 45%	Castle = 40% Monastery = 15%
[3.82853;4.99921]	Monastery	Male = 30% Female = 15% Andy = 0% Null = 55%	Porto = 30% Chapel = 30%
[11.03254;3.22156]	Library	Male = 25% Female = 50% Andy = 0% Null = 25%	Porto = 20% Monastery = 20%
[6.94245;6.93482]	Market	Male = 55% Female = 15% Andy = 0% Null = 30%	Porto = 25% Wine = 25%
[7.74924;6.87212]	Restoration	Male = 35% Female = 20% Andy = 0% Null = 45%	Porto = 15% Wine = 15%
[12.48713;14.25677]	Museums	Male = 35% Female = 15% Andy = 0% Null = 50%	Porto = 10% Restoration = 10%
[11.87735;15.73569]	Bridge	Male = 45% Female = 25% Andy = 0% Null = 30%	Porto = 55% Museum = 10%
[10.00026;15.26161]	Wine	Male = 75% Female = 15% Andy = 0% Null = 10%	Porto = 15% Bridge = 15%
[13.55555;14.68666]	Garden	Male = 25% Female = 25% Andy = 0% Null = 50%	Porto = 25% Chapel = 15%

After this list has been thoroughly reviewed and a full description of it has been made, it is possible to move on to the second vector to be analyzed.

Also another implementation of DBSCAN was introduced regarding the vector List Group Descriptions as it can be seen in (Code Snippet 5.23).

```

1 vectorListGroupDescriptions = StandardScaler().fit_transform(
    vectorListGroupDescriptions)
2
3 db = DBSCAN(eps=0.3, min_samples=4)
4 db.fit(vectorListGroupDescriptions)
5
6 core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
7 core_samples_mask[db.core_sample_indices_] = True
8 labels = db.labels_
9
10 n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
11 n_noise_ = list(labels).count(-1)
12
13 unique_labels = set(labels)
14 colors = [plt.cm.Spectral(each)
15           for each in np.linspace(0, 1, len(unique_labels))]
16 for k, col in zip(unique_labels, colors):
17     if k == -1:
18         # Black used for noise.
19         col = [0, 0, 0, 1]
20
21     class_member_mask = (labels == k)
22
23     xy = vectorListGroupDescriptions[class_member_mask & core_samples_mask]
24     plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
25             markeredgecolor='k', markersize=14)
26
27     xy = vectorListGroupDescriptions[class_member_mask & ~core_samples_mask]
28     plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
29             markeredgecolor='k', markersize=6)
30
31
32 plt.title('Estimated number of clusters DBSCAN ListGroupDescriptions: %d' %
33         n_clusters_)
34 plt.show()

```

Code Snippet 5.23: Second DBSCAN clustering Algorithm Implementation

Since it was previously explained all the methodology behind it, most of the operations are repetitive from the first implementation of the Algorithm DBSCAN.

This second implementation generates a different plot than the first DBSCAN plot as it can be seen on (Figure 5.28).

By its instance it generates also a different description of the clusters since the results are very different than the first Plot as it can be seen on (Figure 5.29).

5.4.3.4 Performance Evaluation

In this subsection the performance of DBSCAN is evaluated and is compared with the previous clustering algorithms in terms of effectivity (accuracy). Since this is an unsupervised clustering

Figure 5.28: Second DBSCAN Plot

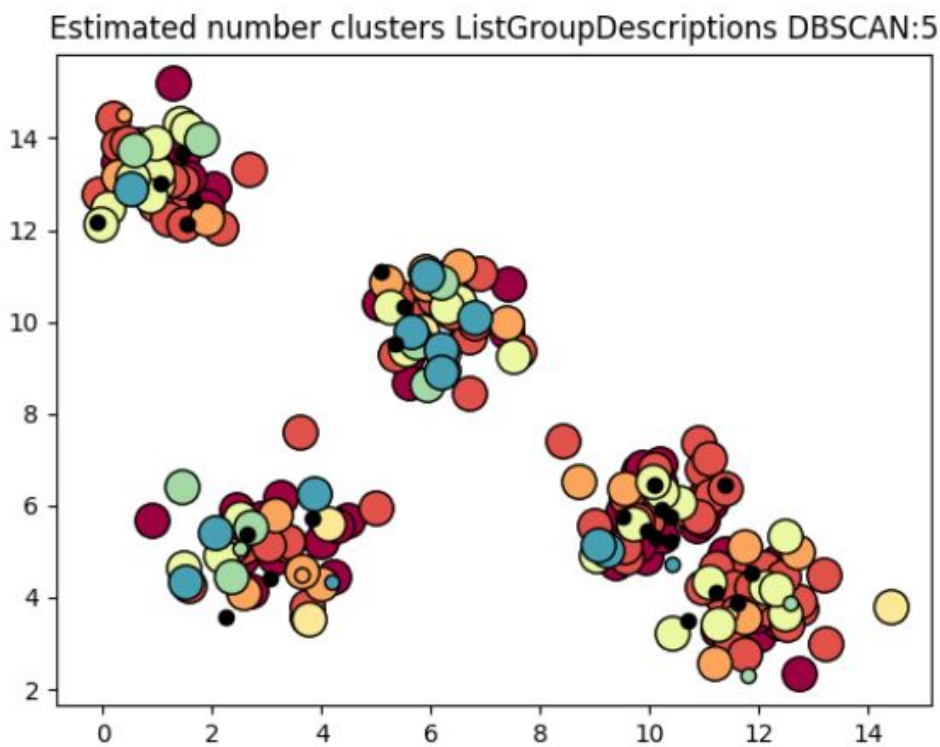


Figure 5.29: Description of the Clusters Second DBSCAN Plot

Cluster Coordinate	Identification	Gender	Other Points of Interest
[2.85342;5.55592]	Post	Male = 45% Female = 2% Andy = 5% Null = 35%	Palace = 15% Chapel = 5%
[1.93213;13.50134]	Train	Male = 35% Female = 30% Andy = 0% Null = 35%	Beach = 25% Monastery = 5%
[6.34662;10.27231]	Porto	Male = 35% Female = 25% Andy = 0% Null = 40%	Porto = 25% Castle = 20%
[10.12025;6.12435]	Garden	Male = 45% Female = 10% Andy = 0% Null = 45%	Wine = 15% Monastery = 5%
[11.61775;3.50242]	View	Male = 25% Female = 15% Andy = 0% Null = 60%	Castle = 35% Monastery = 5%

algorithm there are no common quantitative measure of the classification accuracy, therefore the accuracy is evaluated by visual inspection.

It is also important to enhance its advantages and disadvantages as it can be seen on (Figure 5.30).

A very important is also the Purity ratio obtained when using DBSCAN in both approaches. It was possible to calculate using the ratio obtained from dividing the correct number of documents by the total amount of documents, which gave as a result the following values as it can be seen in (Table 5.4).

Table 5.4: Purity Measure DBSCAN

Score Purity DBSCAN ListGroupDescriptions	0.798
Score Purity DBSCAN List Preferences	0.821

Figure 5.30: DBSCAN Advantages and Disadvantages Table

Advantages of DBSCAN:	Disadvantages of DBSCAN:
<ul style="list-style-type: none"> Is great at separating clusters of high density versus clusters of low density within a given dataset. 	<ul style="list-style-type: none"> Does not work well with dealing with clusters of varying densities. While DBSCAN is great at separating high density clusters from low density clusters, DBSCAN struggles with clusters of similar density.
<ul style="list-style-type: none"> Is great with handling outlier within the dataset. 	<ul style="list-style-type: none"> Struggles with high dimensionality data, if given data with too many dimensions DBSCAN suffers.

5.4.4 BIRCH

5.4.4.1 BIRCH Algorithm

Birch core functions are all about two main important terms: Clustering Feature and Clustering Feature Tree.

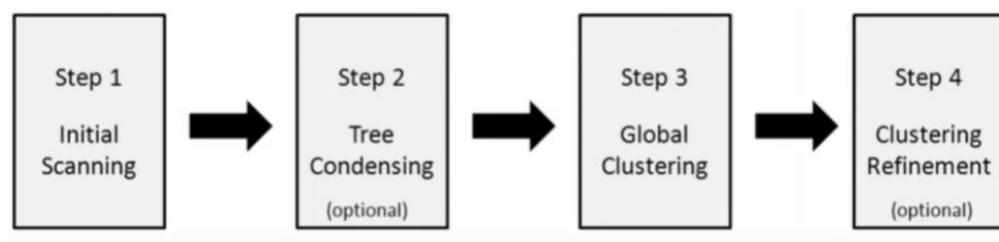
The first one, Clustering Feature, also known as CF, refers to BIRCH algorithm trying to minimize the memory requirement of large dataset by summarizing the information contained in dense regions as clustering feature entries.

BIRCH summarizes large datasets into smaller ones forming then the so-called Clustering Feature Regions. A Clustering Feature entry is defined as an ordered Triple (N, LS, SS). The first element refers to the number of Data points in the cluster, the second one is the linear sum of those data points and by last, the last parameter, refers to the squared sum of the data points in that cluster.

The second one, Clustering Feature Tree, is a tree representation where each leaf node contains a sub-cluster. Each entry in a Clustering Feature Tree contains a pointer to a child node and a Cluster Feature section made up of the sum of Cluster Feature entries in the child nodes. There are a maximum number of data points a sub-cluster in the leaf node of the Cluster Feature Tree can hold, which is referred as the name of threshold value. [90]

The Algorithm behind BIRCH can be seen in a big overview on (Figure 5.31):

Figure 5.31: BIRCH overview



The Cluster Feature Tree is a very compact representation of the original data set because not every entry in a leaf node is a single data point.

Looking at the Algorithm more closely it is possible to see that it is divided into four different phases/steps.

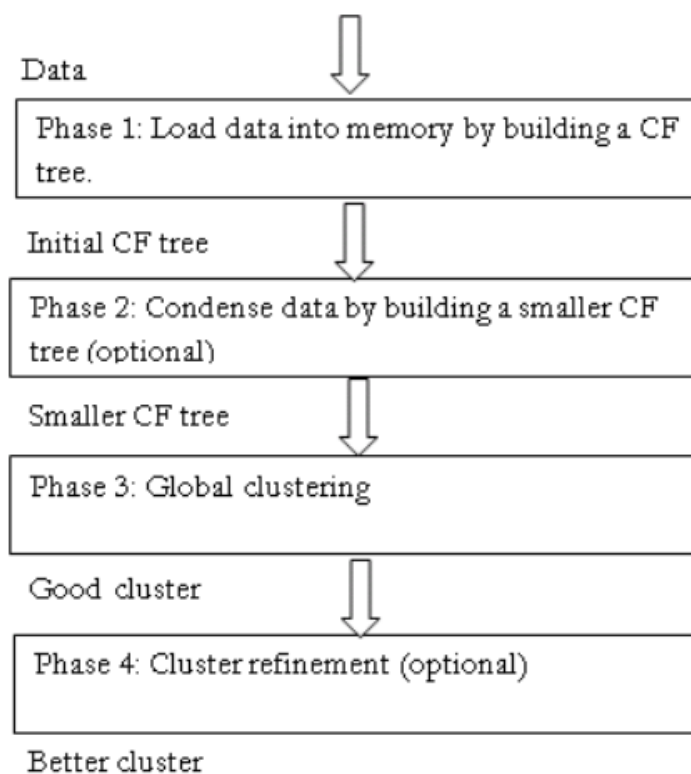
The first phase refers to Loading Data into memory by building an initial Cluster Feature Tree, it can be seen also as an initial scanning of the data set to load the data into memory and as such build the Cluster Feature Tree.

The second phase, Tree Condensing, resizes the data set by building a smaller Cluster Feature Tree trying to remove more outliers.

After these two phases are concluded, the third step refers to Global Clustering in which are used Cluster Feature Entries and lastly the Clustering Refinement in which fixes the problem with Cluster Feature trees where same valued data points may be assigned to different leaf entries.

A more detailed representation of the BIRCH algorithm can be seen on (Figure 5.32):

Figure 5.32: Pseudo-code BIRCH



5.4.4.2 Determining the Parameters

BIRCH takes as input a collection of values, but from all of them three are the most important ones which are “*threshold*”, “*branching_factor*” and “*n_clusters*”.

The first one refers, as expressed previously, to the maximum number of data points a sub-cluster in the leaf node of the Cluster Feature tree can hold.

The second parameter is referent to the maximum number of Cluster Feature sub-clusters in each node. If it happens for a new sample to enter exceeding the *branching_factor*, then that node is split in two nodes redistributing the sub clusters in each node.

The last parameter is referring to the final clustering step, which treats the sub clusters from the leaves as new samples. It is always set to None so the final clustering step is not performed and the sub clusters may be returned as they are.

5.4.4.3 Implementation of BIRCH clustering Algorithm

BIRCH is a scalable clustering method that only requires one time scan of the data set making it perfect to work with large datasets. This Algorithm, more formally, starts by clustering the dataset first in small summaries after clustering the whole dataset.

In context to the Clustering Feature Tree the algorithm compresses the data into the sets of Clustering Feature nodes.

The Algorithm starts after defining some initial factors. The *branching_factor* is related to the maximum number of Cluster Feature sub clusters in each node and is set by default at 50.

Another important parameter to highlight is the threshold which is defined as the radius of the sub cluster obtained by merging a new sample and the closest sub cluster should be lesser than the threshold. Otherwise a new sub cluster is started.

First the object BIRCH has to be created following with the data fitting in order to be able to populate the Algorithm. The next lines are based on the derivation of the characteristics such as the exemplars and labels and by consequence the number of clusters as it can be seen in (Code Snippet 5.24).

```

1 model = Birch(branching_factor=50, n_clusters=None, threshold=1.5)
2 model.fit(vectorListPreferences)
3
4 labels = model.labels_
5 centroids = model.subcluster_centers_
6 pred = model.predict(vectorListPreferences)
7
8 plt.scatter(vectorListPreferences[:, 0], vectorListPreferences[:, 1], c = pred)
9 plt.scatter(vectorListPreferences[:, 0], vectorListPreferences[:, 1], c=labels,
10             cmap='rainbow', alpha=0.7, edgecolors='b')
11 plt.title('Estimated number of clusters Birch vectorListPreferences: %d' % len(
12           centroids))
13 plt.show()

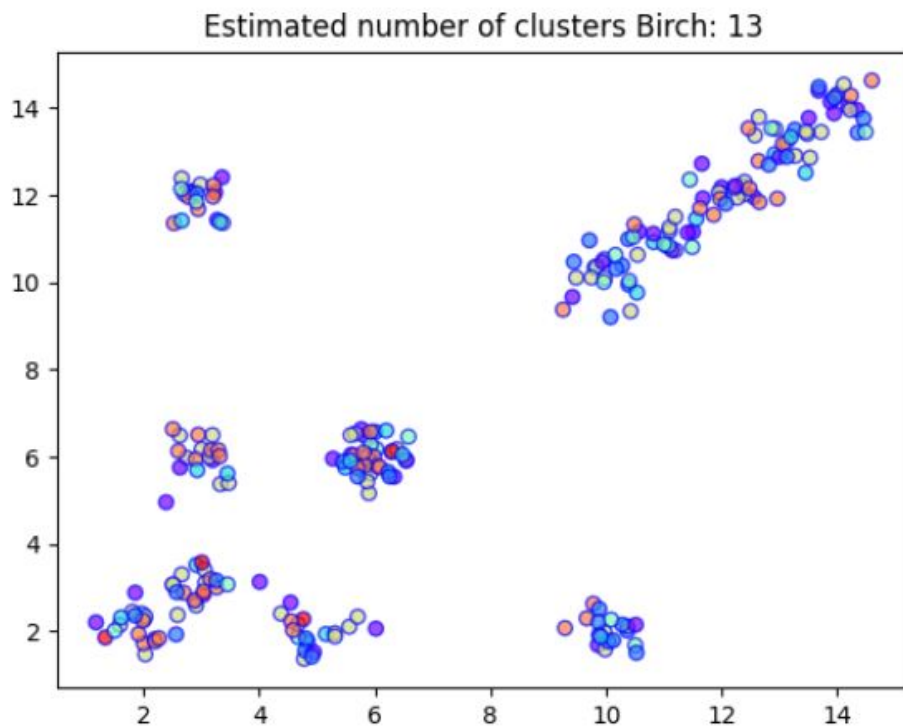
```

Code Snippet 5.24: First BIRCH clustering Algorithm Implementation

The rest of the code is regarding the Plot Result and is mostly directed towards drawing the plot.

A visual representation of this code snippet can be seen on (Figure 5.33).

Figure 5.33: BIRCH Plot



To obtain a better overview it is possible to describe the coordinates of those clusters and to actually identify them and place them accordingly. It is also possible to identify what was the prevailing gender according to the identification of the cluster and also what other points of interest are connected to that Cluster as it can be seen on (Figure 5.34).

After this list has been thoroughly reviewed and a full description of it has been made, it is possible to move on to the second vector to be analyzed.

Also another implementation of BIRCH was introduced regarding the vector List GroupDescriptions as it can be seen in (Code Snippet 5.25).

```

1 model = Birch(branching_factor=50, n_clusters=None, threshold=1.5)
2 model.fit(vectorGroupDescriptions)
3
4 labels = model.labels_
5 centroids = model.subcluster_centers_
6 pred = model.predict(vectorGroupDescriptions)
7
8 plt.scatter(vectorGroupDescriptions[:, 0], vectorGroupDescriptions[:, 1], c =
    pred)
9 plt.scatter(vectorGroupDescriptions[:, 0], vectorGroupDescriptions[:, 1], c=
    labels, cmap='rainbow', alpha=0.7, edgecolors='b')
10 plt.title('Estimated number of clusters Birch GroupDescriptions: %d' % len(
    clusters))

```

Figure 5.34: Description of the Clusters First BIRCH Plot

Cluster Coordinate	Identification	Gender	Other Points of Interest
[1.72762;2.62346]	Castle	Male = 40% Female = 15% Andy = 5% Null = 45%	Palace = 15% Chapel = 15%
[2.85468;4.73457]	Porto	Male = 25% Female = 30% Andy = 5% Null = 45%	Beach = 20% Monastery = 15%
[3.82345;7.73456]	Palace	Male = 20% Female = 20% Andy = 0% Null = 60%	Porto = 35% Castle = 15%
[3.62347;11.72348]	Beach	Male = 35% Female = 15% Andy = 0% Null = 55%	Wine = 15% Monastery = 5%
[4.72346;2.72457]	Chapel	Male = 45% Female = 10% Andy = 0% Null = 45%	Castle = 25% Monastery = 10%
[3.77457;4.75472]	Monastery	Male = 30% Female = 15% Andy = 0% Null = 55%	Porto = 25% Chapel = 20%
[11.72883;3.78245]	Library	Male = 25% Female = 50% Andy = 0% Null = 25%	Porto = 15% Monastery = 20%
[6.38345;6.83456]	Market	Male = 55% Female = 15% Andy = 0% Null = 30%	Porto = 15% Wine = 15%
[7.72345;6.73457]	Restoration	Male = 35% Female = 20% Andy = 0% Null = 45%	Porto = 15% Wine = 10%
[12.76234;14.77234]	Museums	Male = 35% Female = 15% Andy = 0% Null = 50%	Porto = 25% Restoration = 10%
[11.27882;15.27245]	Bridge	Male = 45% Female = 25% Andy = 0% Null = 30%	Porto = 15% Museum = 15%
[10.26234;15.23564]	Wine	Male = 75% Female = 15% Andy = 0% Null = 10%	Porto = 15% Bridge = 25%
[13.73457;14.82443]	Garden	Male = 25% Female = 25% Andy = 0% Null = 50%	Porto = 25% Chapel = 15%

```
11 plt.show()
```

Code Snippet 5.25: Second BIRCH clustering Algorithm Implementation

Since it was previously explained all the methodology behind it, most of the operations are repetitive from the first implementation of the Algorithm BIRCH.

This second implementation generates a different plot than the first BIRCH plot as it can be seen on (Figure 5.35).

By its instance it generates also a different description of the clusters since the results are very different than the first Plot as it can be seen (Figure 5.36).

5.4.4.4 Performance Evaluation

Birch provides a clustering method for very large datasets, making it plausible by concentrating on densely occupied regions by creating compact summaries.

Since this is an unsupervised clustering algorithm there are no common quantitative measure of the classification accuracy, therefore the accuracy is evaluated by visual inspection.

It is also important to enhance its advantages and disadvantages as it can be seen on (Figure 5.37).

A very important one is also the Purity ratio obtained when using BIRCH in both approaches. It was possible to calculate using the ratio obtained from dividing the correct number of documents

Figure 5.35: Second BIRCH Plot

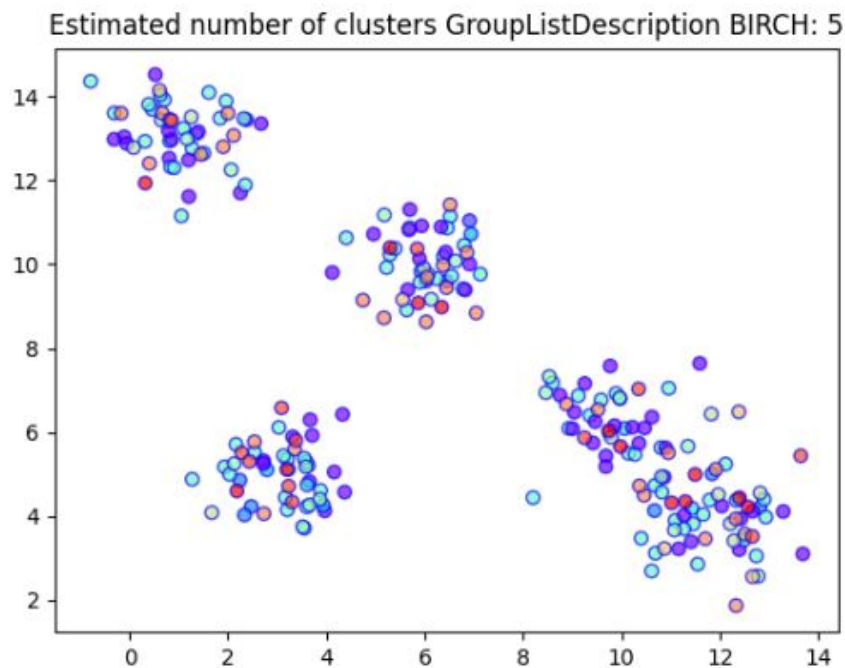


Figure 5.36: Description of the Clusters Second BIRCH Plot

Cluster Coordinate	Identification	Gender	Other Points of Interest
[2.742552;5.64564]	Post	Male = 40% Female = 2% Andy = 5% Null = 30%	Palace = 15% Chapel = 5%
[1.56282;13.78356]	Train	Male = 40% Female = 20% Andy = 0% Null = 40%	Beach = 5% Monastery = 5%
[6.77245;10.66727]	Porto	Male = 35% Female = 15% Andy = 0% Null = 50%	Porto = 5% Castle = 0%
[10.64889;6.38825]	Garden	Male = 50% Female = 0% Andy = 0% Null = 50%	Wine = 15% Monastery = 5%
[11.63478;3.72457]	View	Male = 25% Female = 15% Andy = 0% Null = 60%	Castle = 35% Monastery = 5%

by the total amount of documents, which gave as a result the following values as it can be seen in (Table 5.5).

Table 5.5: Purity Measure BIRCH

Score Purity BIRCH ListGroupDescriptions	0.519
Score Purity BIRCH List Preferences	0.853

Figure 5.37: BIRCH Advantages and Disadvantages Table

Advantages of BIRCH	Disadvantages of BIRCH
<ul style="list-style-type: none"> • One of its main advantages is its capacity to incrementally and dynamically cluster incoming, multi-dimensional metric data points to generate the best quality clustering for a given set of resources (memory and time constraint) 	<ul style="list-style-type: none"> • Birch has one major drawback – it can only process metric attributes. A metric attribute is any attribute whose value can be represented in Euclidean space
<ul style="list-style-type: none"> • It is scalable clustering method 	<ul style="list-style-type: none"> • Relatively sensitive to the outliers
<ul style="list-style-type: none"> • Designed for very large data sets 	<ul style="list-style-type: none"> • Easily drawn into local optimal
<ul style="list-style-type: none"> • Only one scan of data is necessary 	<ul style="list-style-type: none"> • The number of clusters needed to be preset

Chapter 6

Conclusions

This chapter consists in a succinct description of the project's main problem and the most relevant characteristic of the developed solution.

6.1 Dissertation Summary

The premise of this project was the creation of a web platform bounded to an intelligent system. The search of the most appropriate technologies for the project were also a very important step during this dissertation. The most relevant characteristics of the overall solution were:

- Choosing a framework for the Machine Learning developing
- Choosing a Database operator
- Choosing what Machine Learning modules to implement
- Connecting the Database to the Intelligent system

6.2 Accomplished Goals

The accomplished goals can be considered to be the relevant characteristics depicted in the previous section. However, we can summarize, at a higher level, the main fulfilled objectives of the solution in the following list:

- Developing a module responsible for extracting information from the API
- Developing the back-end part of the Project
- Integrating the Application with the back-end part.
- Use of Machine Learning Algorithms to improve data quality
- Building intelligent systems to work with the clean Data obtained

It is considered that the stated objectives have all been successfully completed. However, it is emphasized that currently the development of an application is typically an iterative and incremental process. In this sense, during the period of the dissertation, essentially, the main pillars have been developed that enhance the continuous development of the project in a later period.

6.3 Limitations and future work

Although there are no current limitations that impact the functional use of the solution, there is always room for improvements.

From a simple analysis it was possible to notice a few future improvements that could be done:

- The requests to the API could be optimized by taking better advantage of asynchronous calls;
- Multi-Language support;
- Due to Flickr Privacy Policy update there was a lot of restriction on the information that could be retrieved, perhaps investigate more ways to obtain information regarding Tourist and their corresponding photos;
- Further implementation of more cluster Machine Learning Algorithms.
- Requests to the API more optimized in order to obtain better data to ease the process of data cleaning.

6.4 Final Appreciation

This section is dedicated to a final appreciation regarding the project planning and a summary of the obtained results.

6.4.1 Planning

The initial planning that was developed (Table A.2) was changed many times during the course of the project until reaching a final form. This allowed for a certain flexibility in terms of keeping up with the defined goals.

In the planning made, one month was allocated for the investigation and study of open-source technologies. During this month many sample projects were created in order to test future aspects and functionalities of the Machine Learning Algorithms. These projects were used to see if the open-source frameworks would cover and accomplish the project requirements.

After this month, the development of the whole modules started and it lasted two months according to the initial planning. Although the development of the modules lasted more due to the integration that it had to be made with the Back-end.

After this month, the development didn't went as expected, many tasks took more time than it was expected and many adversities were appearing. The development took a lot more time to do due to Privacy policy restrictions from Flickr part and also because of the complexity of the Machine Learning algorithms.

Outside of the exception mentioned in the previous paragraph, the rest of the development went according to what had been initially planned.

6.4.2 Result Evaluation

After the implementation of the four Algorithms it is possible to see some contrasts that are evident among them as it can be seen in (Table 6.1). The Algorithm that returned the best score regarding it's implementation was, in an overall sense, was DBSCAN, since it was capable of obtaining both high scores regarding the List of Group Descriptions and List Preferences. Nonetheless it is still important to highlight that the highest score obtain was not from the Algorithm DBSCAN, instead it was from BIRCH regarding the List of Preferences, although it obtained a normal score regarding the List of Group Descriptions.

Table 6.1: Result Evaluation

Score Purity Mean Shift ListGroupDescriptions	0.611
Score Purity Mean Shift List Preferences	0.724
Score Purity Affinity Propagation ListGroupDescriptions	0.459
Score Purity Affinity Propagation List Preferences	0.684
Score Purity DBSCAN ListGroupDescriptions	0.798
Score Purity DBSCAN List Preferences	0.821
Score Purity BIRCH ListGroupDescriptions	0.519
Score Purity BIRCH List Preferences	0.853

The higher the purity score means that the objects are well assigned to the cluster they belong to.

It is important to also keep in mind that a good clustering solution results in having compactness measures, separation measures and connectivity measures. The first one evaluates how close the objects are from the same cluster. If a cluster has low variation of a compact measure it means the points are not separated from each other which mean it's compact, it's a good cluster, validating it. Regarding separation measures it indicates how well-separated a cluster is from the other clusters and lastly, connectivity, represents the extent items that are placed in the same cluster as their nearest neighbors in the same data space.

In all the implemented Algorithms it is possible to see all these measures being allocated. The most promising Algorithm of the four that were implemented, which was Mean Shift, after all returned to be almost the worst of them. And, a surprise emerged, having as the most promising Algorithm, DBSCAN.

6.5 Conclusion

In this Dissertation, various definitions and contents of the concept of smart tourism were studied in detail, and the general expression in the definition was emphasized. In the definition and description of the literature, the concept of smart destination and smart tourist have been also analyzed.

The main topic of the dissertation was to endeavor to give a reasonable definition and outline the basic assumptions of the smart tourism concept. It distinguishes smart destinations, smart business ecosystems and smart experiences as three essential segments upheld by the data creation, processing and exchange layers. Also some differences were identified mainly Smart vs Intelligence and Smart Tourism vs e-Tourism. The first difference was regarding which term better suits for describing this new way of providing Tourism, whether it is Smart or Intelligence and, after studying both words, we reached the conclusion that the best and most suitable word is smart due to its contents, because it symbolizes a more extensive and large data input approach. The other main difference, Smart Tourism vs E-Tourism, proved that smart tourism is different from e-tourism, not only in the core technology it uses, but also in the method of creating a better destination experience.

After this literature review the main problem was presented which was the development of a platform with an intelligent system that could be intended of maximizing visitor satisfaction, creating dynamic and personalized routes according to users' preferences and interests.

As such it was developed an Intelligent System capable of collecting complementary information relevant to the automatic generation of routes and be of use to the customer.

We expect that the developed system be effective and outperform the classical approaches on how to define and customize a journey to a customer.

Appendix A

A.1 Overall Definition of Smart Tourism

Table A.1: Overall Definitions of Smart Tourism. Source: own elaboration

(Buhalis 2003; Werthner and Ricci 2004).	“Smart tourism can be seen as a logical progression from traditional tourism and more recently e-tourism in that the groundwork for the innovations and the technological orientation of the industry and the consumers were laid early with the extensive adoption of information and communication technologies (ICT) in tourism, for instance in the form of global distribution and central reservation systems, the integration of Web-based technologies that led to the emergence of e-Tourism”
United Nations World Tourism Organization (UNWTO, 2017)	“Smart tourism includes smart tourism experiences that enable tourists to communicate and interact more closely with local residents, local businesses, local government, and tourist attractions in cities”
Wang (2014)	“Smart tourism functions are described in the cycle of smart service, smart guide, smart shopping guide, payment settings, service line, and smart destination management. Within the specified cycle, smart tourism functions include transactions, such as online or credit card payment, traffic flow, weather information, and registration of tourist movements”
Gretzel (2018)	“Smart tourism should be regarded as a tourism development and management mindset or philosophy with larger implications for tourism governance and for the strategic orientation of the destination”

A.2 Tourist characterization based on the information collected from Flickr. Source: Own elaboration

	Identification	Path_alias	Gender	Location	PlacesOfInterest
1	64453831@N08	jb_1984	Null	Null	Porto
2	132825659@N02	Null	Null	Spain	Castles Palaces
3	41087279@N00	Null	Null	UK	Hotel Holidays
4	50357226@N00	carloscoutinho	Male	Portugal	Porto River
5	159915226@N07	Null	Null	Null	Porto River
6	48083463@N08	tiago_miranda94	Male	Null	Porto Train
7	158534870@N07	heli3planes	Null	Null	Race
8	186269826@N02	Null	Male	Null	Cat
9	40685767@N00	_tonidelong	Null	Null	Null
10	118450654@N07	raeinforma	Null	Spain	Null
11	13189706@N07	itza	Null	Null	Palaces
12	21195678@N02	tomasinrin	Null	Null	Null
13	192654410@N04	Null	Male	Null	Null
14	142960189@N08	jaes-sport	Null	Portugal	Null
15	192000348@N07	cyrohenrique	Null	Null	Garden
16	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
17	64686261@N02	agenciabrasilia	Null	Brazil	Null
18	25710856@N06	alexandre Lambertini	Male	Brazil	Train
19	82289802@N00	stankuns	Male	Brazil	Chapel
20	69574134@N00	moacirdsp	Null	Portugal	Beach
21	134943669@N05	Null	Null	Null	Chapel
22	30564870@N05	Null	Male	Spain	Null
23	51839301@N00	Null	Male	Null	Null
24	125725157@N04	Null	Null	Null	Porto
25	154547444@N06	Null	Male	Null	Beach Garden Porto
26	43830008@N05	larrymyhre	Male	USA	Porto
27	190787816@N08	Null	Male	Null	Porto
28	163158844@N05	Null	Null	Null	Porto
29	189527910@N08	Null	Null	Null	Train
30	147013804@N03	Null	Male	Belarus	Porto
31	69696865@N04	ma_collection	Male	Null	Null
32	73849556@N08	johanpape	Male	Null	Porto
33	79078037@N03	wgaspar	Null	Null	Chapel
34	167399054@N07	Null	Female	Null	Null
35	187359146@N07	Null	Male	Spain	River
36	157554803@N02	Null	Null	Null	Palace Chapel
37	105117706@N07	togisi	Male	Null	Train
38	191839120@N02	habitacaosp	Null	Null	Chapel
39	135489431@N06	jj-d	Null	France	Null
40	37043255@N04	Null	Male	Italy	Porto
41	133200397@N03	sergeigussev	Male	Null	Porto
42	26082117@N07	sybarite48	Male	Null	Null
43	22710957@N05	punxutawneyphil	Male	Null	Null
44	23974506@N05	louisjacob	Null	Canada	Wine Garden
45	147285568@N03	Null	Null	Portugal	Null
46	124304205@N07	carlosperulan	Male	Spain	Palace
47	46191841@N00	franganillo	Male	Null	Porto Barcelona
48	23863971@N08	javiolano	Null	Null	Null
49	34288348@N07	dameboudicca	Female	Sweden	Palace
50	50879678@N03	Null	Male	France	Porto
51	101545735@N08	Null	Null	Spain	Null
52	147380472@N04	Null	Male	Null	Palace
53	124156478@N05	frajamara	Mostly_Male	Spain	Palace

	Identification	Path_alias	Gender	Location	PlacesOfInterest
54	154377259@N04	Null	Male	Null	Null
55	96738681@N02	mflinera	Male	Spain	Palace
56	62841817@N05	luajr	Male	Mexico City	Palace
57	157554803@N02	Null	Null	Null	Palace Chapel Monastery
58	74304499@N06	d_polo	Male	Null	Null
59	112365209@N03	Null	Male	Spain	Null
60	48597791@N04	ernstkers	Male	Null	Bridge Chapel
61	190944552@N05	Null	Male	Null	Porto
62	91124353@N05	sky_hlv	Male	Null	Palace Spain
63	105368886@N07	Null	Male	USA	Palace
64	190771852@N05	Null	Null	Germany	Null
65	94858257@N00	deepfriedkudzu	Female	USA	Palace
66	45898619@N08	Null	Null	UK	Train
67	97612991@N05	ajtomaz	Male	Null	Bridge Chapel
68	15631662@N00	rixa	Female	Brazil	Brazil
69	60403423@N00	fer-ribeiro	Male	Portugal	Null
70	181859947@N04	romaabrantas	Null	Portugal	Null
71	138171302@N03	noxikitty	Null	Null	Null
72	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
73	45904055@N06	diegosoaresproducoes	Null	Null	Wedding
74	181719773@N05	Null	Female	Brazil	Garden
75	30032799@N07	eli_medeiros	Null	Null	Null
76	82289802@N00	stankuns	Male	Brazil	Chapel
77	154377259@N04	Null	Male	Null	Castles

	Identification	Path_alias	Gender	Location	PlacesOfInterest
78	7791788@N04	Null	Null	Portugal	Church Castle
79	152329057@N06	Null	Null	Null	Null
80	69574134@N00	moacirdsp	Null	Portugal	Beach
81	46017193@N00	guyfoggiwill	Mostly_Female	UK	Church Castle
82	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
83	109657983@N02	niloresende	Male	Brazil	Chapel
84	26082117@N07	sybarite48	Male	Null	Null
85	134237361@N08	Null	Female	Null	Null
86	27250052@N07	marcusbranda	Male	Brazil	Null
87	126940304@N02	redesmare	Null	Brazil	Null
88	189677531@N08	dannielliberlonj	Null	Null	Null
89	134943669@N05	Null	Null	Null	Chapel
90	39641028@N04	fernandodelgado	Male	Portugal	Null
91	192786779@N03	Null	Male	Null	Null
92	142960189@N08	jacs-sport	Null	Portugal	Null
93	192688110@N03	Null	Male	Null	Null
94	60591747@N06	joe_bloggs_railway_photos	Male	Null	Porto
95	151483630@N02	Null	Female	Brazil	Train Cars
96	191938216@N08	peruredondo	Female	Spain	Palace Porto
97	142960189@N08	jacs-sport	Null	Portugal	Null
98	136440562@N02	lorestars	Mostly_Female	Spain	Porto Garden Museums
99	144598141@N07	Null	Null	Null	Porto
100	64453831@N08	jb_1984	Null	Null	Porto

	Identification	Path_alias	Gender	Location	PlacesOfInterest
101	37563893@N00	endogamia	Male	Null	Porto
102	187999844@N06	Null	Null	Null	Beach
103	154547444@N06	Null	Male	Null	Beach Garden Porto
104	129490826@N04	governodematogrossoimagem	Null	Null	College
105	186269826@N02	Null	Male	Null	Cat
106	126478940@N02	Null	Male	Null	Garden
107	54929734@N00	stoupaduck	Male	UK	Porto
108	168196594@N04	Null	Male	Null	Porto Bridge
109	94380433@N05	omgdolls	Null	Germany	Null
110	151555943@N06	olobao	Null	Null	Null
111	55608858@N05	jrobertoblancorojo	Male	Null	Porto
112	143003901@N03	malvene	Null	Null	Null
113	21109902@N00	bibber	Null	Null	Garden
114	77062625@N08	ruidanielferreira	Male	Null	Null
115	191522024@N08	republicafotografica	Null	Null	Porto
116	43830008@N05	larrymyhre	Male	USA	Porto
117	181719773@N05	Null	Female	Brazil	Garden
118	50879678@N03	Null	Male	France	Porto
119	137290456@N04	psyched	Null	Null	Null
120	94835768@N04	Null	Female	Germany	Null
121	153321046@N07	Null	Null	Null	Null
122	157554803@N02	Null	Null	Null	Palace Chapel Monastery Garden
123	23489706@N05	jakza	Null	Brazil	Null
124	21841909@N08	flores_paisagens	Null	Brazil	Null
125	192866118@N07	nivaldomenezesfotografia	Null	Brazil	Null
126	93964712@N03	huof	Andy	Null	Null
127	142523289@N07	oculbeforefootbal	Null	Null	Garden
128	125106052@N04	marcos_jerlich	Male	Null	Null

	Identification	Path_alias	Gender	Location	PlacesOfInterest
129	26577438@N06	biarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
130	189898733@N06	Null	Null	Null	Porto
131	7209760@N05	letsplaythat	Male	Brazil	Null
132	42938039@N02	chacajaba	Mostly_Female	Null	Null
133	69574134@N00	moacirdsp	Null	Portugal	Beach
134	44377899@N08	ciqueira	Male	Brazil	Null
135	131747856@N08	jg-instants_of_light	Null	Null	Monastery Church
136	12349333@N07	jfmvaz	Male	Portugal	Garden Porto
137	15631662@N00	rixa	Female	Brazil	Null
138	80848542@N00	zwigmar	Null	Portugal	Null
139	13817902@N00	duanemoore	Male	USA	Porto
140	7426825@N07	peters_view	Null	Europe	Null
141	181719773@N05	Null	Female	Brazil	Wine Restoration Garden
142	158652191@N04	felnob	Male	Portugal	Null
143	26577438@N06	biarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
144	106167288@N03	Null	Null	France	Null
145	24696776@N06	bellaphon	Null	UK	Restoration
146	32374483@N00	lifes_too_short_to_drink_cheap_wine	Null	USA	Wine
147	42345437@N05	Null	Male	Null	Wine

	Identification	Path_alias	Gender	Location	PlacesOfInterest
148	192605652@N05	juniorboo	Null	Brazil	Chapel Church
149	192778893@N04	guenter_becker	Null	Null	Wine Restoration
150	158334061@N05	teatroguaira	Null	Null	Restoration
151	184771757@N05	assessoriarfb	Null	Null	Wine Restoration
152	191662510@N04	Null	Andy	UK	Null
153	63889287@N02	gpekin	Male	Null	Null
154	20944194@N06	wamwick_carter	Null	Australia	Library
155	192878725@N02	Null	Male	Null	Null
156	16086041@N00	acider	Male	Null	Null
157	191871390@N03	darioperacchi	Male	Italy	Monastery Church
158	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
159	31526369@N00	cinglesdeberti	Mostly_Female	Null	Null
160	146405858@N06	stuartepulvermuller	Null	Null	Africa
161	32113849@N02	y_xs	Mostly_Male	Null	Null
162	146649643@N08	prefeituradeitapevi	Null	Brazil	Porto
163	182015205@N06	Null	Female	Null	Null
164	69574134@N00	moacirdsp	Null	Portugal	Beach
165	83193914@N00	mcmvjr1971	Male	Brazil	Null
166	142992283@N07	cbhriodasvelhas	Null	Null	Rain
167	191857692@N03	Null	Female	Null	Null
168	191675752@N05	ulicoucaok	Male	Null	Null
169	156407248@N08	daniielduran0006	Male	Null	Null
170	104208685@N05	muniarica	Null	Null	Null
171	167887978@N03	Null	Male	Null	Null
172	9236683@N02	profjoao	Null	Brazil	Null

	Identification	Path_alias	Gender	Location	PlacesOfInterest
173	184648015@N02	wesleyxavi	Male	Null	Null
174	49733424@N02	gildatonello	Female	Brazil	Garden
175	141594777@N02	Null	Female	Null	Porto Zoo
176	13723429@N03	claudioarriens	Male	Null	Garden
177	17998004@N00	scoordeiro	Male	Portugal	Portugal
178	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
179	191945373@N05	fotosricardocampos	Male	Null	Null
180	15631662@N00	rixa	Female	Brazil	Null
181	43830008@N05	larrymyhre	Male	USA	Porto
182	154547444@N06	Null	Male	Null	Porto Beach Garden
183	94356082@N03	Null	Null	Null	Null
184	49532757@N05	ferraz_gustavo	Male	Brazil	Garden
185	192025051@N07	globalsportsnet	Null	Null	Garden
186	15631662@N00	rixa	Female	Brazil	Null
187	9771767@N04	tqtran	Null	Null	Null
188	87901948@N08	paco_san_juan_nofrio	Null	Spain	Library
189	49143546@N06	agenciasenado	Null	Brazil	Garden
190	99897397@N02	mandapropndf	Null	Brazil	Null
191	51127152@N04	fabianocaetano	Male	Null	Garden
192	145846856@N08	Null	Male	Brazil	Garden
193	116337886@N07	a_little_brighter	Male	USA	Null
194	155983864@N06	Null	Male	Null	Monastery Church
195	7138083@N04	xiquinho	Null	Null	Porto
196	191705209@N06	Null	Null	Null	Beach
197	39641028@N04	fernandodelgado	Male	Portugal	Null

	Identification	Path_alias	Gender	Location	PlacesOfInterest
198	158652191@N04	felnob	Male	Portugal	Car Wine
199	69574134@N00	moacirdsp	Null	Portugal	Beach
200	28402310@N06	Null	Male	Sweden	Null
201	190694387@N03	Null	Male	Null	Null
202	127342087@N06	Null	Null	Portugal	Null
203	37772404@N04	peddroneto	Male	Null	Null
204	33000909@N08	carlospontalti	Male	Null	Null
205	192605652@N05	juniorboo	Null	Brazil	Null
206	24696776@N06	bellaphon	Null	UK	Restoration
207	192778893@N04	guenter_becker	Null	Null	Wine
208	32374483@N00	lifes_too_short_to_drink_cheap_wine	Null	USA	Wine
209	158334061@N05	teatroguaira	Null	Null	Restoration
210	42345437@N05	Null	Male	Null	Wine
211	184771757@N05	assessoriarfb	Null	Null	Wine
212	12363891@N03	Null	Male	Portugal	Null
213	77516652@N02	Null	Null	Portugal	Null
214	152221251@N07	emilien50	Male	Null	Porto
215	145157152@N06	Null	Male	Null	Null
216	183855681@N02	Null	Female	Null	Porto
217	192314097@N04	Null	Female	Null	Porto
218	154547444@N06	Null	Male	Null	Beach Garden Porto
219	133200397@N03	sergeigussev	Male	Null	Porto
220	27461366@N07	ernest_descals	Male	Null	Null
221	158652191@N04	felnob	Male	Portugal	Car Wine

	Identification	Path_alias	Gender	Location	PlacesOfInterest
222	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
223	106167288@N03	Null	Null	France	Null
224	21188987@N04	jviegas	Male	Null	Null
225	80848542@N00	zwingmar	Null	Portugal	Null
226	121307261@N02	Null	Null	Null	Null
227	50357226@N00	carloscoutinho	Male	Portugal	Porto
228	24490268@N04	oneterny	Mostly_Male	UK	Null
229	98519674@N00	t3mujin	Null	Portugal	Girls
230	144992359@N06	werner-schoen	Mostly_Female	Null	Porto
231	8047395@N04	pindavale	Null	Brazil	Null
232	8751400@N06	gesongerloff	Male	Brazil	Null
233	44626001@N07	jokaguerra	Null	Brazil	Beach
234	41404125@N07	jmpascual	Male	Spain	Train
235	33938770@N05	dannyfoster	Male	Canada	Null
236	61343200@N02	anacarvalhais	Female	Portugal	Null
237	144992359@N06	werner-schoen	Mostly_Female	Null	Porto
238	128655002@N02	gicristovaophotography	Andy	Null	Castles Palaces
239	78492139@N04	visitportoandnorth	Null	Portugal	Castles Palaces
240	63281384@N00	vribeiro	Null	Portugal	Monastery Church
241	41980486@N07	sharonhahndarlin	Female	Null	Porto
242	60694653@N08	elad283	Null	Null	Train

	Identification	Path_alias	Gender	Location	PlacesOfInterest
243	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
244	41980486@N07	sharonhahndarlin	Female	Null	Porto
245	73422480@N00	tompagenet	Male	UK	Null
246	88810541@N08	richardboyd484	Male	Null	Porto,Train
247	157509912@N06	Null	Male	France	Porto
248	10579681@N04	antoniocorreia	Male	Portugal	Null
249	154547444@N06	Null	Male	Null	Beach Garden Porto
250	187665811@N03	Null	Male	Null	Garden Porto
251	60694653@N08	elad283	Null	Null	Train
252	157509912@N06	Null	Male	France	Porto
253	15631662@N00	rixa	Female	Brazil	Null
254	186269826@N02	Null	Male	Null	Cat
255	148317604@N06	Null	Null	Null	Null
256	33838140@N05	joco08	Male	Luxembourg	Null
257	13817902@N00	duanemoore	Male	USA	Porto
258	65228321@N00	giubr	Male	Null	Null
259	168196594@N04	Null	Male	Null	Port Garden Bridge
260	97612991@N05	ajtomaz	Male	Null	Null
261	21182099@N05	Null	Male	Brazil	Null

	Identification	Path_alias	Gender	Location	PlacesOfInterest
262	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
263	190259538@N06	joabizarrofotografia	Null	Null	Porto
264	147858930@N06	luxurypete	Male	Null	Monastery Church
265	23199003@N06	polispoliuiu	Null	Cyprus	Porto
266	61574452@N00	maradentro	Null	Spain	Null
267	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
268	60694653@N08	elad283	Null	Null	Train
269	60083277@N00	Null	Null	Portugal	Null
270	14818554@N05	atramos	Null	Null	Null
271	64453831@N08	j_b_1984	Null	Null	Porto
272	50879678@N03	Null	Male	France	Null
273	133428854@N08	Null	Male	Null	Null
274	59140308@N04	torredelporticiolo	Null	Italy	Null
275	46855129@N03	Null	Male	France	Null
276	185308961@N02	Null	Male	Portugal	Null
277	69574134@N00	moacirdsp	Null	Portugal	Beach
278	94380433@N05	omgdolls	Null	Germany	Null
279	12306079@N02	aheroy	Null	Netherlands	Null
280	48597791@N04	ernstkers	Male	Null	Bridge Chapel
281	48597791@N04	ernstkers	Male	Null	Bridge Chapel

	Identification	Path_alias	Gender	Location	PlacesOfInterest
282	186269826@N02	Null	Male	Null	Cat
283	181914731@N02	Null	Male	Null	Null
284	77597938@N00	nhowells	Male	UK	Null
285	131522476@N05	carloscosta77	Male	Null	Porto
286	40025409@N03	juanbdj	Male	Spain	Bridge
287	128719564@N08	dariodelia	Male	Null	Null
288	10037165@N02	maxlindquist	Male	Cyprus	Null
289	192439180@N04	tombarth	Male	Null	Porto
290	87433225@N00	maucantara	Null	Brazil	Porto
291	152221251@N07	emilien50	Male	Null	Porto
292	160294373@N04	schaikoski_spotter	Male	Brazil	Null
293	65211201@N00	flaneur	Male	USA	Porto
294	60591747@N06	joe_bloggs_railway_photos	Male	Null	Porto
295	149245688@N02	Null	Null	Null	Porto
296	152125454@N03	marzocchi	Female	Italy	Porto
297	46595448@N05	gelfiser	Null	Italy	Null
298	121307261@N02	Null	Null	Null	Null
299	137245032@N04	philkingsbury	Male	Australia	Bridge
300	100787040@N07	dustintrain	Male	China	Porto
301	87901948@N08	paco_san_juan_riofrio	Null	Spain	Library
302	56975064@N08	dinoignani	Male	Italy	Library
303	60403423@N00	fer-ribeiro	Male	Portugal	Null
304	140450721@N08	appenwill	Male	UK	Null
305	90993272@N04	Null	Null	Null	Porto
306	90134546@N00	loose_grip_99	Null	UK	Porto
307	147218853@N08	cpo_D1	Null	Null	Bridge
308	46191841@N00	franganillo	Male	Null	Porto Palace Castle Barcelona
309	43830008@N05	larrymyhre	Male	USA	Porto
310	131454435@N05	Null	Male	Null	Null

	Identification	Path_alias	Gender	Location	PlacesOfInterest
311	26577438@N06	biblarte	Null	Null	Porto Monastery Chapel Museum Garden Church Beach
312	35081255@N08	sintragate	Male	Null	Null
313	57712732@N04	nucleus7	Male	Netherlands	Null
314	50879678@N03	Null	Male	France	Porto
315	26250980@N02	thetapanofhongkong	Mostly_Male	Null	Null
316	69574134@N00	moacirdsp	Null	Portugal	Beach
317	157554803@N02	Null	Null	Null	Palace Chapel Monastery Garden
318	63281384@N00	vribeiro	Null	Portugal	Monastery Church

A.3 Query from Flickr API to populate Photos Table

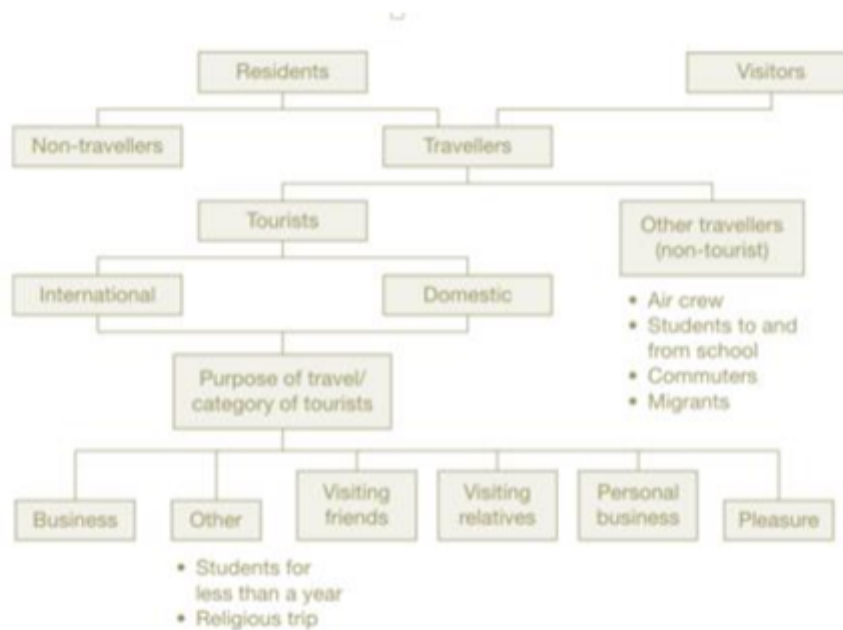
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=c729641e6b00e16409c529d9dc22f89e&tags=Porto&min_upload_date=2018-01-01&max_upload_date=2020-07-01&accuracy=10&content_type=4&has_geo=1&extras=geo%2&per_page=100&format=json&nojsoncallback=1

A.4 Query from Flickr API necessary to obtain a User's Information

[https://www.flickr.com/services/rest/?method=flickr.people.getInfo&api_key=429fdbf7f9096180b7c964c869652482&user_id=listOfProfiles\[x\]&format=json&nojsoncallback=1](https://www.flickr.com/services/rest/?method=flickr.people.getInfo&api_key=429fdbf7f9096180b7c964c869652482&user_id=listOfProfiles[x]&format=json&nojsoncallback=1)

A.5 Tourist Classification according to Chadwick

Figure A.1: Tourist Classification



A.6 Sprint Tasks

Table A.2: Sprint Tasks

Sprint	Sprint Start	Sprint End	Modules and Deliveries
Sprint 1	12/02/2021	26/02/2021	Study of Development Tools
Sprint 2	26/02/2021	12/03/2021	Design
Sprint 3	12/03/2021	19/03/2021	Design
Sprint 4	19/03/2021	26/03/2021	Data Mining
Sprint 5	26/03/2021	02/04/2021	Data Mining, Data Cleaning
Sprint 6	02/04/2021	16/04/2021	Data Mining, Data Cleaning and Dissertation
Sprint 7	16/04/2021	30/04/2021	Data Mining, Data Cleaning Development of Intelligent System
Sprint 8	30/04/2021	14/05/2021	Data Mining, Data Cleaning Development of Intelligent System and Dissertation
Sprint 9	14/05/2021	28/05/2021	Data Mining, Data Cleaning and Development of Intelligent System and Dissertation
Sprint 10	14/05/2021	28/05/2021	Development of Intelligent System and Dissertation
Sprint 11	28/05/2021	11/06/2021	Development of Intelligent System and Dissertation
Sprint 12	11/06/2021	18/06/2021	Dissertation and Refinement of Project
Sprint 13	18/06/2021	25/06/202	Dissertation and Refinement of Project
Sprint 14	25/06/2021	02/07/2021	Development of Intelligent System and Dissertation
Sprint 15	02/07/2021	09/07/2021	Dissertation
Sprint 16	09/07/2021	16/07/2021	Dissertation
Sprint 17	16/07/2021	23/07/2021	Dissertation

References

- [1] Yunpeng Li, Clark Hu, Chao Huang, and Liqiong Duan. The concept of smart tourism in the context of tourism information services. *Tourism Management*, 58:293–300, 2017.
- [2] Dimitrios Buhalis and Aditya Amarangana. Smart tourism destinations. In *Information and communication technologies in tourism 2014*, pages 553–564. Springer, 2013.
- [3] Ulrike Gretzel, Marianna Sigala, Zheng Xiang, and Chulmo Koo. Smart tourism: foundations and developments. *Electronic Markets*, 25(3):179–188, 2015.
- [4] M Sajid Khan, Mina Woo, Kichan Nam, and Prakash K Chathoth. Smart city and smart tourism: A case of dubai. *Sustainability*, 9(12):2279, 2017.
- [5] Antonio López de Avila Muñoz and Susana García Sánchez. Destinos turísticos inteligentes. *Economía industrial*, (395):61–69, 2015.
- [6] Rua-Huan Tsaih and Chih Chun Hsu. Artificial intelligence in smart tourism: A conceptual framework. *Artificial Intelligence*, 2018.
- [7] Meena Kumari Pradhan, Jungjoo Oh, and Hwansoo Lee. Understanding travelers’ behavior for sustainable smart tourism: A technology readiness perspective. *Sustainability*, 10(11):4259, 2018.
- [8] William Cannon Hunter, Namho Chung, Ulrike Gretzel, and Chulmo Koo. Constructivist research in smart tourism. *Asia Pacific Journal of Information Systems*, 25(1):103–118, 2015.
- [9] Yeongbae Choe and Daniel R Fesenmaier. The quantified traveler: Implications for smart tourism development. In *Analytics in smart tourism design*, pages 65–77. Springer, 2017.
- [10] Zheng Xiang and Daniel R Fesenmaier. Big data analytics, tourism design and smart tourism. In *Analytics in smart tourism design*, pages 299–307. Springer, 2017.
- [11] Pasquale Del Vecchio, Gioconda Mele, Valentina Ndou, and Giustina Secundo. Creating value from social big data: Implications for smart tourism destinations. *Information Processing & Management*, 54(5):847–860, 2018.
- [12] Dobrica Z Jovicic. From the traditional understanding of tourism destination to the smart tourism destination. *Current Issues in Tourism*, 22(3):276–282, 2019.
- [13] Chulmo Koo, Seunghun Shin, Keehun Kim, Chulwon Kim, and Namho Chung. Smart tourism of the korea: A case study. In *PACIS*, page 138, 2013.

- [14] Rula A Hamid, AS Albahri, Jwan K Alwan, ZT Al-qaysi, OS Albahri, AA Zaidan, Alhamzah Alnoor, AH Alamoodi, and BB Zaidan. How smart is e-tourism? a systematic review of smart tourism recommendation system applying data management. *Computer Science Review*, 39:100337, 2021.
- [15] Dimitrios Buhalis. Technology in tourism—from information communication technologies to etourism and smart tourism towards ambient intelligence tourism: a perspective article. *Tourism Review*, 2019.
- [16] Ji Hoon Park, Cheolhan Lee, Changsok Yoo, and Yoonjae Nam. An analysis of the utilization of facebook by local korean governments for tourism development and the network of smart tourism ecosystem. *International Journal of Information Management*, 36(6):1320–1327, 2016.
- [17] Xia Wang, Xiang Robert Li, Feng Zhen, and JinHe Zhang. How smart is your tourist attraction?: Measuring tourist preferences of smart tourism attractions via a fcem-ahp and ipa approach. *Tourism management*, 54:309–320, 2016.
- [18] Chulmo Koo, Seunghun Shin, Ulrike Gretzel, William Cannon Hunter, and Namho Chung. Conceptualization of smart tourism destination competitiveness. *Asia Pacific Journal of Information Systems*, 26(4):561–576, 2016.
- [19] Umesh Bodkhe, Pronaya Bhattacharya, Sudeep Tanwar, Sudhanshu Tyagi, Neeraj Kumar, and Mohammad S Obaidat. Blohost: Blockchain enabled smart tourism and hospitality management. In *2019 international conference on computer, information and telecommunication systems (CITS)*, pages 1–5. IEEE, 2019.
- [20] Qionghua Tu and Aili Liu. Framework of smart tourism research and related progress in china. In *International conference on management and engineering (CME 2014)*, pages 140–146. DEStech Publications, Inc, 2014.
- [21] Ning Wang. Research on construction of smart tourism perception system and management platform. In *Applied Mechanics and Materials*, volume 687, pages 1745–1748. Trans Tech Publ, 2014.
- [22] Tomáš Gajdošík. Smart tourism: concepts and insights from central europe. *Czech Journal of Tourism*, 7(1):25–44, 2018.
- [23] Joao Romao and Bart Neuts. Territorial capital, smart tourism specialization and sustainable regional development: Experiences from europe. *Habitat International*, 68:64–74, 2017.
- [24] Aruditya JASROTIA and Amit GANGOTIA. Smart cities to smart tourism destinations: A review paper. *Journal of tourism intelligence and smartness*, 1(1):47–56, 2018.
- [25] Chul Woo Yoo, Jahyun Goo, C Derrick Huang, Kichan Nam, and Mina Woo. Improving travel decision support satisfaction with smart tourism technologies: A framework of tourist elaboration likelihood and self-efficacy. *Technological Forecasting and Social Change*, 123:330–341, 2017.
- [26] Shiwei Shen, Marios Sotiriadis, and Qing Zhou. Could smart tourists be sustainable and responsible as well? the contribution of social networking sites to improving their sustainable and responsible behavior. *Sustainability*, 12(4):1470, 2020.

- [27] Kim Boes, Dimitrios Buhalis, and Alessandro Inversini. Conceptualising smart tourism destination dimensions. In *Information and communication technologies in tourism 2015*, pages 391–403. Springer, 2015.
- [28] Ulrike Gretzel, Lina Zhong, and Chulmo Koo. Application of smart tourism to cities. *International Journal of Tourism Cities*, 2016.
- [29] Ayşen Civelek. Smart cities and smart tourism: Smart city projects and applications in turkey. In *Proceedings of MAC 2018*, pages 323–332. MAC Prague Consulting Ltd Prague, 2018.
- [30] Ulrike Gretzel and Michelle Scarpino-Johns. Destination resilience and smart tourism destinations. *Tourism Review International*, 22(3-4):263–276, 2018.
- [31] Giuseppe D’Aniello, Matteo Gaeta, and Marek Z Reformat. Collective perception in smart tourism destinations with rough sets. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, pages 1–6. IEEE, 2017.
- [32] Kim Boes, Dimitrios Buhalis, and Alessandro Inversini. Smart tourism destinations: ecosystems for tourism destination competitiveness. *International Journal of Tourism Cities*, 2016.
- [33] Maurvi Vasavada and Yash J Padhiyar. Smart tourism»: Growth for tomorrow. *Journal for Researchl Volume*, 1(12), 2016.
- [34] Susana Medina. O museu da faculdade de engenharia da universidade do porto e suas coleções. *Coleções Científicas Luso-Brasileiras: patrimônio a ser descoberto*, 2010.
- [35] Kyung-Hyan Yoo, Marianna Sigala, and Ulrike Gretzel. Exploring tripadvisor. In *Open tourism*, pages 239–255. Springer, 2016.
- [36] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Community detection by affinity propagation. *Work*, page 12, 2008.
- [37] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
- [38] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 274–285. SIAM, 2005.
- [39] Hongjie Jia, Shifei Ding, Xinzheng Xu, and Ru Nie. The latest research progress on spectral clustering. *Neural Computing and Applications*, 24(7):1477–1486, 2014.
- [40] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.
- [41] KP Agrawal, Sanjay Garg, Shashikant Sharma, and Pinkal Patel. Development and validation of optics based spatio-temporal clustering technique. *Information Sciences*, 369:388–401, 2016.
- [42] Xinhua Zhuang, Yan Huang, Kannappan Palaniappan, and Yunxin Zhao. Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing*, 5(9):1293–1302, 1996.

- [43] Mohammed Al-Maolegi and Bassam Arkok. An improved apriori algorithm for association rules. *arXiv preprint arXiv:1403.3948*, 2014.
- [44] Christian Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 1–5, 2005.
- [45] Paul Held and BioTek Instruments. Measure your purity. *GIT Laboratory Journal*, 5:9–11, 2008.
- [46] Richard S Rogers, Michael Abernathy, Douglas D Richardson, Jason C Rouse, Justin B Sperry, Patrick Swann, Jette Wypych, Christopher Yu, Li Zang, and Rohini Deshpande. A view on the importance of “multi-attribute method” for measuring purity of biopharmaceuticals and improving overall control strategy. *The AAPS journal*, 20(1):1–8, 2018.
- [47] Isaac Nape, Valeria Rodríguez-Fajardo, Feng Zhu, Hsiao-Chih Huang, Jonathan Leach, and Andrew Forbes. Measuring dimensionality and purity of high-dimensional entangled states. *Nature communications*, 12(1):1–8, 2021.
- [48] Miguel A Carreira-Perpinan. Acceleration strategies for gaussian mean-shift image segmentation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 1, pages 1160–1167. IEEE, 2006.
- [49] Jong-Hyun Ryu, G Wan, and Sujin Kim. Optimal design of a cusum chart for a mean shift of unknown size. *Journal of Quality Technology*, 42(3):311–326, 2010.
- [50] Lin Yang, Peter Meer, and David J Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [51] Zhiwen Yu, Le Li, Jiming Liu, Jun Zhang, and Guoqiang Han. Adaptive noise immune cluster ensemble using affinity propagation. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3176–3189, 2015.
- [52] Limin Wang, Zhiyuan Hao, Xuming Han, and Ruihong Zhou. Gravity theory-based affinity propagation clustering algorithm and its applications. *Tehnički vjesnik*, 25(4):1125–1135, 2018.
- [53] BI Anqi and Wang Shitong. Transfer affinity propagation clustering algorithm based on kullback-leiber distance. , 38(8):2076–2084, 2016.
- [54] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Db-scan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.
- [55] Domenica Arlia and Massimo Coppola. Experiments in parallel clustering with dbscan. In *European Conference on Parallel Processing*, pages 326–331. Springer, 2001.
- [56] Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with r. *Journal of Statistical Software*, 91(1):1–30, 2019.
- [57] Xu Shou-kun, Wang Chao, Zhuang Li-hua, and Gao Xin-hua. Dbscan clustering algorithm for the detection of nearby open clusters based on gaia-dr2two. *Chinese Astronomy and Astrophysics*, 43(2):225–236, 2019.

- [58] Godwin Ogbuabor and FN Ugwoke. Clustering algorithm for a healthcare dataset using silhouette score value. *International Journal of Computer Science & Information Technology (IJCSIT)*, 10(2):27–37, 2018.
- [59] Hua Jiang, Jing Li, Shenghe Yi, Xiangyang Wang, and Xin Hu. A new hybrid method based on partitioning-based dbscan and ant clustering. *Expert Systems with Applications*, 38(8):9373–9381, 2011.
- [60] Jacob Kogan, Charles Nicholas, and Mike Wiacek. Hybrid clustering with divergences. In *Survey of Text Mining II*, pages 65–85. Springer, 2008.
- [61] Fernando Berzal. «clustering jerarquico. línea». Available: <http://elvex.ugr.es/idbis/dm/slides/42%20Clustering>, 2017.
- [62] Gurpreet Singh, Jaskaranjit Kaur, and Yusuf Mulge. Performance evaluation of enhanced hierarchical and partitioning based clustering algorithm (epbca) in data mining. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 805–810. IEEE, 2015.
- [63] Mona Singh. U-scrum: An agile methodology for promoting usability. In *Agile 2008 Conference*, pages 555–560. IEEE, 2008.
- [64] Jakub Miler and Paulina Gaida. On the agile mindset of an effective team—an industrial opinion survey. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 841–849. IEEE, 2019.
- [65] M Mahalakshmi and Mukund Sundararajan. Traditional sdlc vs scrum methodology—a comparative study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6):192–196, 2013.
- [66] Ashish Mundra, Sanjay Misra, and Chitra A Dhawale. Practical scrum-scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications*, pages 119–123. IEEE, 2013.
- [67] Ken Schwaber and Jeff Sutherland. The scrum guide. *Scrum Alliance*, 21:19, 2011.
- [68] AGILE Alliance. Agile practice guide this book. Project Management Institute, 2017.
- [69] Julian M Bass. Scrum master activities: process tailoring in large enterprise projects. In *2014 IEEE 9th International Conference on Global Software Engineering*, pages 6–15. IEEE, 2014.
- [70] Bob Schatz and Ibrahim Abdelshafi. Primavera gets agile: a successful transition to agile development. *IEEE software*, 22(3):36–42, 2005.
- [71] Nicholas H Gist, Michael V Fedewa, Rod K Dishman, and Kirk J Cureton. Sprint interval training effects on aerobic capacity: a systematic review and meta-analysis. *Sports medicine*, 44(2):269–279, 2014.
- [72] Steve Berczuk. Back to basics: The role of agile principles in success with an distributed scrum team. In *Agile 2007 (AGILE 2007)*, pages 382–388. IEEE, 2007.
- [73] Chris Mutel. Brightway: an open source framework for life cycle assessment. *Journal of Open Source Software*, 2(12):236, 2017.

- [74] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. The emotional side of software developers in jira. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, pages 480–483. IEEE, 2016.
- [75] Neil Postman. Tecnopólio. *A rendição da cultura à tecnologia*. São Paulo: Nobel, pages 13–14, 1994.
- [76] Meigu Guan. On the windy postman problem. *Discrete Applied Mathematics*, 9(1):41–46, 1984.
- [77] Carsten Mueller and Anton Pohl. Plugin for visual paradigm for generating and intelligent optimization of a component diagram based on attributes in the class diagram. *J. Softw.*, 10(3):355–365, 2015.
- [78] Jérôme Gabillaud. *SQL Server 2014: Administración de una base de datos transaccional con SQL Server Management Studio*. Ediciones ENI, 2015.
- [79] Christiaan Lemmen, Peter Van Oosterom, and Rohan Bennett. The land administration domain model. *Land use policy*, 49:535–545, 2015.
- [80] Alex Fabian Yungan Gualli, Cristian Hugo Morales Alarcón, Jorge Edwin Delgado Altamirano, and Lady Marieliza Espinoza Tinoco. Modelo furps para el análisis del rendimiento de frameworks jsf. *3C TIC. Cuadernos de desarrollo aplicados a las TIC*, pages 65–83, 2019.
- [81] Ulf Eriksson. Functional requirements vs non functional requirements, 2017.
- [82] Speed-Learning Agile. Agile in a flash.
- [83] David Huff and Bradley M McCALLUM. Calibrating the huff model using arcgis business analyst. *ESRI White Paper*, pages 1–33, 2008.
- [84] Martin Glinz. On non-functional requirements. In *15th IEEE international requirements engineering conference (RE 2007)*, pages 21–26. IEEE, 2007.
- [85] Lei Wu, Xian-Sheng Hua, Nenghai Yu, Wei-Ying Ma, and Shipeng Li. Flickr distance. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 31–40, 2008.
- [86] LI Xiang-Ru, WU Fu-Chao, and HU Zhan-Yi. Convergence of a mean shift algorithm. 2005.
- [87] Brian Fulkerson and Stefano Soatto. Really quick shift: Image segmentation on a gpu. In *European Conference on Computer Vision*, pages 350–358. Springer, 2010.
- [88] B Jia, B Yu, Qi Wu, Chuanfeng Wei, and Rob Law. Adaptive affinity propagation method based on improved cuckoo search. *Knowledge-Based Systems*, 111:27–35, 2016.
- [89] Mete Çelik, Filiz Dadaşer-Çelik, and Ahmet Şakir Dokuz. Anomaly detection in temperature data using dbSCAN algorithm. In *2011 international symposium on innovations in intelligent systems and applications*, pages 91–95. IEEE, 2011.
- [90] Dongwei Guo, Jingwen Chen, Yingjie Chen, and Zhiyu Li. Lbirc: an improved birch algorithm based on link. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 74–78, 2018.