



Using Data Science to Optimize Waste Collection

By

Guilherme Lopes Londres

Master Thesis in Modeling, Data Analysis
and Decision Support Systems

Supervised by:
Professor Doutor João Manuel Portela da Gama

Faculdade de Economia
Universidade do Porto

2020

Biographical note

Born in the marvelous city of Rio de Janeiro, Brazil, Guilherme Lopes Londres, has started has graduated as Metallurgical Engineer at Federal University of Rio de Janeiro (UFRJ) in 2017. Spending more than 3 years as part of Composites Laboratory (LaCom/COPPE), under supervision of Prof. Dr. Fernando Bastian, where he had the opportunity to apply his Python coding skills in real world data, while doing research.

Guilherme also completed a Materials and Metallurgy Engineering M.Sc. degree in the same University, in 2018. Developed an intelligent classifier to identify the a subtle flaw in a polymeric material used in pipes for deep sea oil extraction. It was achieved coding a neural network classifier using Keras in Python.

In September 2018, he moves from Rio to Porto, to start the M.Sc. in Data Analytics at Faculty of Economics of University of Porto (FEP/U.Porto). In late 2018, Guilherme had the chance to take part into a project in LIAAD/INESCTED, under supervision of Prof. Dr. João Gama. The study was published and gave the basis for the current thesis.

On year later, in September 2019, Guilherme is hired as AI Software Developer Trainee at Nokia, then moving to the capital of Portugal, Lisbon. Working alongside with teams spread in Europe (and in the world sometimes), have become the responsible for the conversations of Nokia Digital Assitant, a chatbot for the telecom industry, also automatizing tests with Python and fine tuning ELK, you know, for search.

Acknowledgements

In first place, I would like to acknowledge my parents, Lúcia and Manoel, for supporting me all these years in my trajectory from Brazil to the World. Thank you! Without you it would be not possible. At all.

To my long-lasting girlfriend Virgínia Bertolo, my partner, living together for more than 6 months (if needed, we can prove it, Mr. Immigration Officer); supporting me, advising (academically) and sharing the most incredible moments. Thanks, honey!

I would like to acknowledge Prof. João Gama, for the support and giving me the chance of being part of LIAAD, Laboratory of Artificial Intelligence at INESCITEC and developing the work which resulted in a published paper.

Also, I would like to acknowledge Nuno Filipe, which embraced the idea and made available the data for the study which resulted in the paper and now was extended to this master thesis.

To my colleagues at Nokia: Rui, Pedro, Thiago and Benoit (le boss!). The company gave me opportunity to work in the industry, learn and expand my abilities, and you guys turned it into a pleasant journey.

To the friends I have made in Portugal. Actually, we have built a sort of a family at Rua de Paranhos, 58 [by the way, that still my official residence in Portugal, if IRS send something, please let me know]. Andrei, Cássio, Diego, Dorgas, Eduardo and Pablo, thanks for the unforgettable moments.

To D. Vera and Eduardo Bastian, who gave me support when applying for this master's program and still always remembered, even if we take 2 months to reply a message.

To my friend Rafael Cidade, who now plays electronic keychains in London. Always available for academical, personal or any other stuff.

Finally, to my lifelong friends. Specially, the greatest Gustavo Cabral, always available and ready for some advice and, why not, a keynote. Thanks, Dr. Keynote.

For all of you, my deepest and sincere thank you!

Abstract

One of the main goals of a smart city is to use intelligently connected services to support decision making in city administration, delivering better living conditions and more sustainable urban spaces. Among the principal problems, modern cities faces are waste collection. In northern Portugal, a company is responsible for collection and recycling the waste of several cities. The data to be used was gathered in the last three years during the non-residential collection circuits; this works intends to be able to classify when there are waste bins to be collected or not. And if available, the quantity. With more than 500 customers using this service, the aim is to build an individual forecasting system for each one of the customers. Therefore, it will be possible to use this data to optimize the route of the trucks, which makes this collection. Around 70% of the costs for picking-up the waste is due to fuel. Also, making the route optimized it is possible not only to offer a better service but to decrease the greenhouse gases emissions. The problem will be solved using Python programming language and its foremost machine learning and data exploring libraries such as Keras, XGBoost and scikit-learn. This project is based in the other two published works but bringing more complexity and completeness to the previous ones. The main goal of this work was to improve the scores obtained in the two previous publications, however it was concluded that those results were good enough. Even though some interesting comparisons could be done between the results obtained with XGBoost, neural networks and decision trees.

Keywords: Data science for social good, smart cities, waste collection

Resumo

Um dos principais objetivos de uma cidade inteligente é usar serviços conectados de maneira inteligente para apoiar a tomada de decisões na administração da cidade, proporcionando melhores condições de vida e espaços urbanos mais sustentáveis. Entre os principais problemas nas cidades modernas está a coleta de lixo. No norte de Portugal, uma empresa é responsável pela coleta e reciclagem dos resíduos de várias cidades. Os dados a serem utilizados foram coletados nos últimos três anos durante os circuitos de coleta não-residenciais; este trabalho pretende ser capaz de prever lixeiras a serem coletadas ou não. E caso haja, a quantidade. Com mais de 100 clientes usando esse serviço, o objetivo é criar um sistema de previsão individual para cada um dos clientes. Portanto, será possível usar estes dados para otimizar a rota dos caminhões, responsáveis pela coleta. Cerca de 70% dos custos de coleta de lixo são causados por gastos com combustível. Além disso, otimizando a rota, é possível não apenas oferecer um serviço melhor, mas também diminuir as emissões de gases de efeito estufa. O problema será resolvido usando a linguagem de programação Python e seu principal aprendizado de máquina e bibliotecas de exploração de dados como Keras, XGBoost e scikit-learn. Este projeto é baseado nos outros dois trabalhos publicados, mas trazendo mais complexidade e completude aos anteriores. O objetivo principal deste trabalho foi melhorar os escores obtidos nas duas publicações anteriores, porém concluiu-se que esses resultados foram bons o suficiente. Ainda assim, algumas comparações interessantes podem ser feitas entre os resultados obtidos com o XGBoost, redes neurais e árvores de decisão.

Palavras-chave: *Data science* para o bem-estar social, cidades inteligentes, coleta de rejeitos

Contents

Biographical note	ii
Acknowledgements	iii
Abstract	iv
Resumo	v
1 Introduction	1
1.1 Objective	2
1.2 Organization of the Thesis	3
2 Literature Review	4
2.1 Data Science for Social Good	4
2.2 Smart Cities	5
2.3 Smart Waste Management	5
2.4 Learning Algorithms	6
2.4.1 Boosting Algorithms	6
2.4.2 AdaBoost	7
2.4.3 Gradient Boosted Decision Trees	7
2.4.4 XGBoost	8
2.4.5 Neural Networks	8
2.5 Sliding Windows	10
2.6 State Of The Art	12
3 Methodology	14
3.1 Dataset	14
3.2 Data Exploration	15
3.3 Data Preprocessing	16
3.3.1 Dealing with missing information (NA's)	16

3.3.2	Dealing with outliers	17
3.3.3	Removing inactive customers	17
3.3.4	Dataset characteristics	18
3.4	Models setup	21
3.4.1	XGBoost	21
3.4.2	Neural Networks	21
3.4.3	Sliding Windows	22
4	Results and Discussion	23
4.1	Results obtained with XGBoost	23
4.1.1	Predicting the presence of containers	24
4.1.2	Predicting the quantity of containers	25
4.2	Results obtained with Neural Networks	27
4.2.1	Predicting the presence of containers	27
4.2.2	Predicting the quantity of containers	30
4.3	Results Discussion	32
4.3.1	Predicting the presence of waste	32
4.3.2	Predicting the quantity of containers	35
4.3.3	Results of Sliding Windows	37
4.4	The Best Model	39
5	Conclusions	42
5.1	Future Works	43
	Bibliography	44

List of Figures

2.1	Percentage of recycled plastic in Portugal. Adapted from Lusa (2019)	6
2.2	Example of the structure of a neural network, with m inputs, two hidden layers and one exit.	9
2.3	Shape of ReLU function. Extracted from Sarkar (2018)	10
2.4	Example of a sliding window, with time step of 1 month.	11
3.1	Example of missing data.	17
3.2	Box-plot of 5 randomly picked customers. (a) 2% outliers, (b) 5% outliers, (c) 1% outliers, (d) 5% outliers and (e) with 6% outliers.	17
3.3	Restaurant A - it is possible to see a increasing trend during the summer months.	18
3.4	Restaurant B - There is no clear oscillation.	19
3.5	Restaurant B - Collection of (a) paper and (b) plastic containers compared with the total sum of paper, plastic and glass line.	19
3.6	Restaurant C - After a spike in May, the trend is stable.	20
3.7	Diverse business - no clear trace of seasonality can be seen.	20
4.1	Histogram of results obtained by XGB for the presence of containers.	25
4.2	Histogram of results obtained by XGB for the quantity of containers.	26
4.3	Histogram of results obtained by NN with 7 layers and 0.30 split for the presence of containers.	29
4.4	Histogram of results obtained by NN with 7 layers and 0.30 split to predict the quantity of containers.	31
4.5	Critical Difference Diagram with models for the binary prediction.	33
4.6	Comparison of histograms from XGBoost, Neural Network with 7 layers and Decision Tree.	35
4.8	Comparison of histograms from XGBoost, Neural Network with 7 layers and Decision Tree.	37

4.7	Critical Difference Diagram with models for the quantity of containers prediction.	38
4.9	Critical Difference Diagram for all models, considering both cases.	41

List of Tables

4.1	Average and rank of the results for each model, for each material, trained with XGBoost, for binary classification.	24
4.2	Training with XGB to predict the presence of containers - accuracy obtained. . .	24
4.3	The average results for the 1p1 training of XGBoost.	25
4.4	Average and rank of the results for each model, for each material, trained with XGBoost.	26
4.5	Training with XGB to predict the quantity of containers - accuracy obtained. . .	26
4.6	The average results for the 1p1 training of XGBoost.	27
4.7	Average and rank of the results for each model, for each material, trained with Neural Networks	28
4.8	Training with NN with 7 layers and 0.30 split to predict the quantity of containers - accuracy obtained.	29
4.9	Average results for training with neural networks 1p1.	29
4.10	Average and rank of the results for each model, for each material, trained with Neural Networks	30
4.11	Training with NN with 7 layers and 0.30 split to predict the quantity of containers - accuracy obtained.	31
4.12	Average results for 1p1 training of neural networks.	32
4.13	Ranked models based on average of results for the prediction of presence of waste to be collected, including the reference Londres, Filipe, and Gama (2020).	34
4.14	List with average results obtained with all models and compared to Londres et al. (2020).	36
4.15	Average results for binary prediction, 1p1.	39
4.16	Average results for quantity prediction, 1p1.	39
4.17	Final rank with all models, considering both scenarios.	40

Chapter 1

Introduction

As the world population grows every day, and cities get more and more crowded, massive amounts of food and goods are consumed. After this chain effect, one thing is inevitable: the increase in waste generation. A fast and straightforward example may be taken while drinking pure coffee. For every marvelous cup of premium coffee brewed under the pressure of 12bar in a home machine, there is a capsule to be discarded. Depending on the coffee brand, the capsule is made of aluminum, and there is a return and reprocessing program. Alternatively, maybe it is made of polymer (plastic in ordinary language), turning to recycle quite challenging. About 39,000 coffee capsules are made in the world, which up to 29,000 ends in landfills (Moskvitch (2019)).

This elementary summary is only one of many which can be taken from a person's routine. On the other side, there are complex systems to collect and adequately destinate all this waste. Indeed, regarding Northern Portugal, September 2019, 43,050 tons of waste was collected, which about 24% was recovered¹ (LIPOR (2019)). In this region, the organization which is responsible for the collection and treatment of the waste offers a service named door-to-door non-residential collection. It consists of delivering three 120-liter colored bags for waste deposit. Each one of these green, blue and yellow containers is intended for a particular type of material, respectively glass, carton and paper and, plastic and metal. Each signee, which usually is hospitals, restaurants, and shops, has its bags collected a few days during the week.

The main target of this project is to design an intelligent system based on machine learning algorithms to forecast when those bag containers will be full enough to be collected. With this approach, it will be possible to optimize the collection routes of the trucks. The costs with fuel are responsible not only for more greenhouse gas emissions but for almost 70% of the

¹Recovering means being properly destined to recycle or revaluation.

collection costs (Teixeira, Antunes, and de Sousa (2004)). The idea of optimizing such routes is not new and widely reported in the literature as in (Calabrò and Komilis (2019), Martinho et al. (2017), Navghane, Killedar, and Rohokale (2016)). However, these works usually tackle it as an optimization problem regarding the routes, or, more recently, suggesting ways to implement smart sensors to trash containers.

The idea of using smart sensors in bins is exciting, but it can be costly and complex to implement. On the other hand, Ferrer and Alba (2019) develops a software concatenating the optimization and machine learning algorithms. His objective is to build a route optimizer, which takes into account how full the trash bins are. To perform this estimation, the authors use machine learning algorithms, using a source of the data collected by the trucks during the circuits.

The focus of this work is using artificial intelligence through machine learning algorithms to build a classifier. It primarily will predict if there is a container to be collected and, secondly, how many containers will be full enough to be taken. It will be possible using the data of the last three years (2017-2019) of waste collection routes by the company, which is responsible for that in northern Portugal. It is pathed in the study made by de Freitas Filipe (2018) in his Master's thesis. His project could successfully assess the feasibility of the idea. Now, it will be expanded, exploring newly available data and other machine learning algorithms, expecting to reach higher accuracy in the forecasting.

1.1 Objective

During his master's thesis, de Freitas Filipe (2018) has used the data available by that time. Today, after two years, as the data collection at LIPOR, the company responsible for waste collection and treatment in northern Portugal, was kept, there are more data available. Also, a shift in the methodology is proposed in this work. There is the intention to build a reusable pipeline and, therefore, Python programming language (Python Software Foundation (n.d.)) will be used. Furthermore, the previous work had as a target the ability to explain the decisions. In other words, the possibility of the paths taken by the software to be human-understandable. This is not a constraint anymore.

The main target of this project is achieving the best possible accuracy. Therefore, machine learning algorithms that usually achieve better results (Gama (2017)) can be used, such as random forests, neural networks, and even deep learning.

1.2 Organization of the Thesis

The main objective of this work is to build a intelligent classifier to be used by LIPOR, to optimize its waste collection routes. Being able to predict if there will be waste containers to be collected, or even by predicting how full the collecting truck will be, by previously knowing the quantity of containers which should be made available per each customer. In chapter 2 a literature review is done to give a basic knowledge concerning smart cities, waste collection and a brief theoretical concept of the algorithms which were used. In the same chapter a review of the state-of-the-art is presented, to check what have been already published regarding the use of intelligent systems to optimize waste collection.

In chapter 3, the methodology used during the development of this work is presented. The dataset is described and what were the decisions taken regarding missing information and outliers. Also, the setups of the algorithms and the sliding windows are given. Chapter 4 contains the results of the models and the pertinent discussions to better understand each outcome. Finally, in chapter 5 the conclusions are presented followed by some suggestions to further improving this work.

Chapter 2

Literature Review

2.1 Data Science for Social Good

Data science is, in a certain way, a new field of science. It involves statistics, machine learning, and data. Lots of data. Social good comprehends many different concepts and, more recently, is the term used for the usage of technology, science, and, mainly, data, aiming to solve or diminish social problems.

It is widespread nowadays to have events like hackathons and summits, sponsored by big companies to gather people with knowledge, companies, or government institutions with the data and a problem to tackle. As examples, it is possible to cite the Eurekathon (NOS (2019)). An event where a telecommunications company provided the data, the government the problem – how to better take care of isolated older adults in Portugal, and students and young professionals the solution in 24 hours.

Today, in 2020, there is a sort of hacker philosophy among the ones who work with data. This way of thinking drives the developer community and big companies such as Google, Facebook, and Microsoft to open to the world what they have been doing. The simplest way of doing it is publishing their researches on open websites, so anyone with an internet connection may read it without having to pay. Also, how many courses and tutorials are open on YouTube and internet blogs? These people tend to help other people.

Another great example is SoGood, a workshop (Alzate et al. (2019)), which is part of ECML-PKDD (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases). It is a dedicated part of this prominent conference to publish and spread projects of data science aimed for social good. It is prevalent to see social good projects to involve another relatively new concept: the smart cities.

2.2 Smart Cities

The concept of a smart city comes from the idea of using intelligent systems to help the decision-making process in the administration of a city. One example is the use of big data for better understanding and identifying the sources of old problems such as floods. Maybe it is possible to find a correlation with the incorrect waste disposal in specific regions and the occurrence of such a problem. Knowing the origin of the issue allows authorities to tackle it more appropriately and efficiently. Smart cities are popping everywhere in the world, increasing people's life quality and, also permit the better usage of monetary resources (Eremia, Toma, and Sanduleac (2017)).

Therefore, a subject that maybe was unnoticed by people now comes to attention as it affects their lives. One of the most critical topics in smart cities is waste management.

2.3 Smart Waste Management

Almost as a standard, the traditional way to collect waste is by sending trash trucks with big containers to a pre-defined route. At specific points, it stops, gathers the trash bins repeatedly until the truck gets fully loaded or the route ends. It is being replicated in many cities around the world, for decades. A paradigm faced by waste companies is how to improve those route, or, how to reduce its expenses (Arribas, Blazquez, and Lamas (2010)). Only the fuel consumption of those trucks is responsible for 70% of the whole operation (Teixeira et al. (2004)).

One of the ways of doing this reformulation is changing the way this collection happens. There are a few different models (Dat, Truc Linh, Chou, and Yu (2012), Tung and Pinnoi (2000), Sniezek and Bodin (2006)), such as the one similar to which is used in the city of Porto in Portugal, where big waste containers are disposed in pre-defined points in a neighborhood. The so-called "eco-pontos," or, eco-points in a free translation. In each of those places have three big colored containers: green (glass), blue (paper and carton) and yellow (metal and plastic), and a grey bigger container for non-recyclable waste. A few days a week, a specific truck collects one of the specific containers.

More recently, there is the adoption of smart sensors in the trash bins. With this approach, the companies can know when such a container is full, or almost full, ready to be collected. This information can later be used as an input for intelligent routing systems. Classically, what can be seen in the literature, is that this routing subject is tackled with different optimization algorithms (Calabrò and Komilis (2019), Martinho et al. (2017), Navghane et al. (2016), Teixeira et al. (2004)). An innovation was the mixture of two different approaches by Ferrer and Alba (2019) when the

authors have used not only an optimization side but an intelligent system to predict how full the trash bins would be.

Smart waste management is searching for new and more efficient ways to tackle it, using new connected sensors to help route management or other technological approaches. Every day the society becomes more active and aware of how vital the appropriate waste management is. Therefore, more investment and development are rising in this field not only as government politics but from popular demand (de Freitas Filipe (2018), ISWA (2017)). In Portugal, the volume of recycled plastic was increasing until 2015, as seen in Figure 2.1. The reason decrease in the last years is due to the change in the population behavior, buying less plastic (Lusa (2019)).

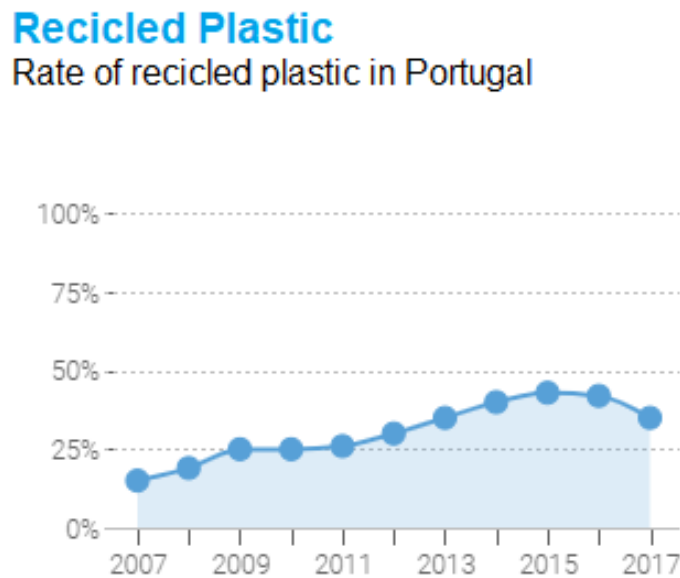


Figure 2.1: Percentage of recycled plastic in Portugal. Adapted from Lusa (2019)

2.4 Learning Algorithms

Among the vast variety of learning algorithms available to be used, two mainstream techniques were chosen: XGBoost and neural networks.

2.4.1 Boosting Algorithms

The main idea behind the boosting technique is to reinforce the scores obtained by a simple, or weak algorithm, such as decision tree. It will concentrate in the points where misclassifications

have occurred, retraining those points in order to increase the score. It happens through several iterations, which can be called estimators (Pedregosa et al. (2011)). As an example of boosting algorithms it is possible to cite a few such as AdaBoost, XGBoost, and LightGBM.

In this work it was used the XGBoost algorithm, to better understand this method, it is easier to understand AdaBoost.

2.4.2 AdaBoost

Adaptive boosting, or AdaBoost, works combining the results of weak classifier and giving weights to each result. It starts with all the training observations receiving the same weights. Then, misclassifications receive higher weights, which should therefore be minimized. It is done by retraining those sets of points, increasing or decreasing its boundaries in order to correctly classify every example. From this characteristic derives its weakness. AdaBoost tends to be easily misled by noise in the dataset, what can result in overfitting (Desarda (2019)).

2.4.3 Gradient Boosted Decision Trees

Gradient Boosted Decision Trees is an ensemble machine learning algorithm. When combining more than one technique, the name ensemble is given for the merge which generally yields better results. Also known as Gradient Tree Boosting, it is the combination of decision trees and gradient boosting. Despite being a boosting algorithm, gradient boosting differs from AdaBoost as it is a gradient optimizer, while the first optimizes weights (Rahman et al. (2020)). Being a gradient optimizer, means it is a gradient descent optimizer. In other words, there is a loss function which should be minimized, if it was in the same direction of the gradient, it would be growing forever. Yet, as it is a gradient descent optimizer, it will try to diminish this loss function, moving it in the opposite direction of the gradient, trying to reach zero loss (Haykin (2009), Mason, Baxter, Bartlett, and Frean (1999)).

Gradient Tree Boosting combines decision trees and the gradient boosting, as previously said. Decision trees is known as a weak algorithm, therefore a good fit for an ensemble. The procedure in its essence is similar to AdaBoost, a few trees will be generated (or stumps), and combining the result of each one will be taken into account. The stumps with misclassifications will then be retrained in the next iteration and the weighted sum will be done again. This process is repeated until the iterations (or estimators) is achieved or a given threshold is reached.

2.4.4 XGBoost

XGBoost, acronym for eXtreme Gradient Boosting (Chen and Guestrin (2016)), is a gradient boosting algorithm combined with - mainly - decision trees. Developed on top of the ideas of Friedman (2001), it has become a mainstream algorithm after its recognized performance in many Kaggle competitions (Reinstein (2017)). The principal characteristics of this algorithm is the speed of training, the possibility of parallelization, and the high accuracy it usually achieves. On the downside, there is the facility to overfit.

This algorithm differs from the gradient boosted trees as it uses both L1 and L2 regularization and second order derivative for the loss function. The latter will give better information regarding the direction of the gradient, therefore facilitating the path until the minimum value (Abou Omar (2018)). The regularization will reduce the tendency for overfitting without increasing the bias in the same order. It works adding penalization to weights which become too high in comparison to the other on the weights matrix (Ng (2004)).

Similar to XGBoost there is LightGBM (Ke et al. (2017)), developed by a research group from Microsoft, it differs from the first, as it uses an histogram-based algorithm to find the best scores to use, as the second use a scheme called GOSS - Gradient-based one-side sampling (Rahman et al. (2020)). A deeper and comprehensive comparison of these and other models can be seen at Demšar (2006).

2.4.5 Neural Networks

Artificial neural networks (ANNs), or simply neural networks (NNs), are a set of computational techniques inspired by the functioning of the human brain, mainly in its learning capacity. The human brain structure is basically composed of neurons interconnected by what is called synapses. Thus, when a stimulus occurs in the dendrite, with a voltage greater than that necessary to polarize the axon, the electrical impulse is then transmitted from one neuron to another.

The first artificial neuron model was introduced by McCulloch and Pitts in 1943 (McCulloch and Pitts (1943)). In the beginning a lot was developed on the subject until most of the studies stagnated due to the limitation of that structure in separating nonlinear separable classes. This type of class, very common in electronics applications, is called the or-exclusive function (XOR). In 1986, with the development of the backpropagation training algorithm, it was possible to separate these types of classes, making it possible to use networks with more than one hidden layer, which today is called multilayer perceptron (MLP) (Moura Junior (2014)). NNs are usually used both as classifiers, separating different classes, as well as approximators, where a function is defined to describe a data set.

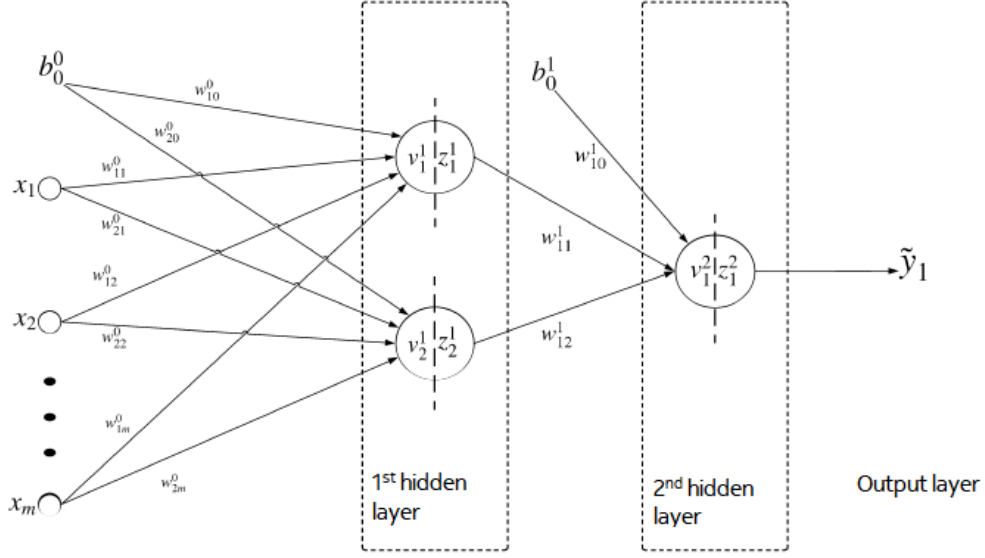


Figure 2.2: Example of the structure of a neural network, with m inputs, two hidden layers and one exit.

Activation Functions

Activation functions work as limiters or compactors, restricting the output values to a limit, usually between -1 and 1 or between 0 and 1. These functions can be divided into two types: linear and non-linear (Haykin (2009)). In the case of using only linear activation functions, the network becomes a linear combiner. However, it is more usual to use nonlinear ones, such as the hyperbolic tangent (*Tanh*) (eq. 2.1), a type of sigmoid function (eq. 2.2).

$$z_n^k = \varphi(v_n^k) = \tanh(v_n^k) = \frac{1 - \exp(-2v_n^k)}{1 + \exp(-2v_n^k)} \quad (2.1)$$

$$z_n^k = \varphi(v_n^k) = \frac{1}{1 + \exp(-v_n^k)} \quad (2.2)$$

The sigmoid function (eq. 2.2) is also known as the logistic function. With the advancement of training techniques, new activation functions have emerged and, currently, as reported in the literature and in the manuals of the machine learning libraries, the most used functions are ReLU and Softmax (Goodfellow, Bengio, and Courville (2016)). ReLU (eq. 2.3) is the acronym for Rectified Linear Unit, the shape of its curve is shown in Figure 2.3.

$$f(x) = x^+ = \max(0, x) \quad (2.3)$$

The main difference between the hyperbolic tangent and ReLU functions is the image range of the two. The first ranges from -1 to 1, while ReLU ranges from 0 to $+\infty$. This characteristic when using *Tanh* means that in deep networks negative gradients continue to be updated, which slows down the training process and turns not viable its usage for deep learning (the case of NNs with many layers).

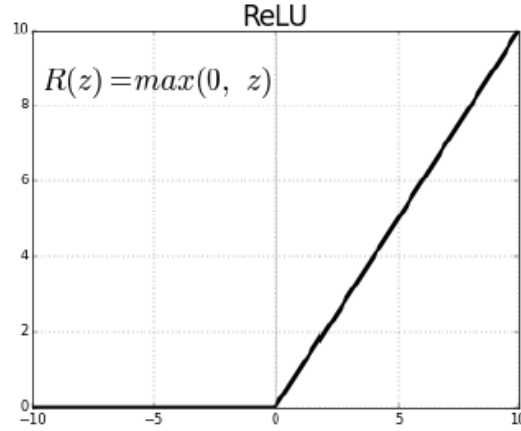


Figure 2.3: Shape of ReLU function. Extracted from Sarkar (2018)

The Softmax function (eq. 2.3), is a generalization of the logistics function, being very useful especially when inserted in the network's output layer. This function has its greatest advantage over the sigmoid due to the fact that the latter tends to quickly saturate the values at its limits, 0 and 1. Softmax, however, makes a smoother distribution, resulting in a probability of belonging to certain categorical class.

$$\text{softmax}_i(a) = \frac{e^{a_i}}{\sum_{i=1}^K e^{a_i}} \quad (2.4)$$

Different activation functions can be combined in the same NN in order to obtain different results with the same architecture.

2.5 Sliding Windows

Regarding the type of the problem presented, where it is desired to forecast the data for the next circuit of waste collecting, it was thought that would be interesting to have an additional concept, the sliding windows. The main idea here is to train the model with data which is temporally closer to the prediction which will be done. Using this approach it is expected to reduce

or even eliminate the traces of seasonality which may be present.

Imagine the case that a new customer has just arrived, how would be possible to train a model for him if there is any of a very limited amount of data? With the sliding window the idea is to use newer data to predict the next scenario, like in the case of the new customer. But also to forget past data of old customers. Think on a customer, a restaurant, which just opened last year and, therefore, had a certain demand and now became successful and its activity has increased 100%. If the model still relying in this data from one year ago, probably it will decrease its accuracy, and will probably fail.

In this work, it will be used with the implementation of the neural networks, using one month to predict the next one, what has been called 1p1. However, there are no restrictions regarding the time step chose. For instance, it could be used a step of 1 week, 1 day, or, depending on the application, minutes, seconds and so on. As the collection circuits are run 2 or 3 times a week, choosing a weekly step would imply in a very reduced dataset to be used for the training of the models. Therefore, 1 month of increase was chosen, as it would be sufficient to gather the a minimum amount of data to be used as input of the model. An example of the strucure which is used in this work is shown in Figure 2.4.

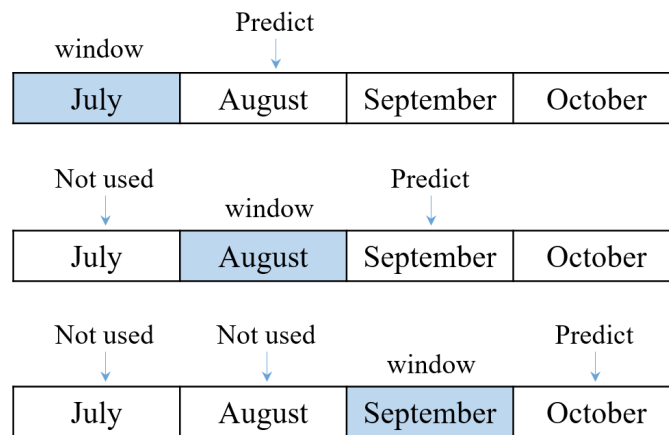


Figure 2.4: Example of a sliding window, with time step of 1 month.

2.6 State Of The Art

In this section, a brief review of the most recent studies regarding the areas concerned by this work are presented.

de Freitas Filipe (2018), in his master's thesis, has proposed the use of machine learning algorithms to predict the existence of waste available for collection and also the quantity of waste. The data of one year (2017) of waste collection in northern Portugal was the source for the project. There were available more than 26.000 single observations, regarding 141 customers. However, it was only validated against five customers. One of his goals, or constraints, was to find an intelligent method to solve the problem with an elevated level of explicability, mainly decision trees and its variants. The author used Weka and Rapid Miner in work, and inside those environments, the algorithms were tested. Mean accuracy of 97.45% was the best result achieved.

Londres et al. (2020) developed an extension of de Freitas Filipe (2018) work. In fact, the same dataset was used, the main goal was to reproduce the idea developed at the thesis but to make a prediction system for each one of the 141 customers. It was achieved using Python and its machine learning libraries such as Scikit-learn. The decision trees algorithm was used, to preserve the system explainable. The system was trained in two different ways. The first the whole dataset with data for all months was introduced. The other approach tried to simulate the real world of data, i.e., training with one month and prediction the next one. In other words, the concept of sliding windows was introduced. The main idea for using this technique was the get rid of possible biased results provided by different months. This procedure was divided into three ways so-called: 1p1, 2p1, and 3p1.

- **1p1** - one previous month used for training and then predicting the next one;
- **2p1** - two previous months used for training and then predicting the next one;
- **3p1** - three previous months used for training and then predicting the next one.

Also, the authors predicted the availability of trash bins to be collected and the number of bins available. To be able to report the result for each one of the 141 customers, the results were plotted as histograms, were more with the best scoring approach being the 1p1 where more than 85% of the customers reached at least 96.2% of F1 score.

Ferrer and Alba (2019) brings innovation to the field when mixing two different approaches. The traditional way to tackle the routing problem, i.e., trough optimization algorithms, with the usage of artificial intelligence to increase the quality of their system. The authors explain that

the differential of their system is tackling and trying to predict how full the trash bins will be. In other systems, it is usually referred to as a fine-grain feature and therefore considered constant.

Bringing this variable to the problem, they not only increase the complexity of the problem but also open a new way to deal with it. Using the data gathered by the truck drivers, reporting for a few months how the containers were full or not, it was possible to predict with the desired accuracy if a bin would be ready for pick-up or not. They have considered 80% of the bin capacity as the threshold for being inserted in the collection route.

To classify, or, to predict, the completeness of the trash containers, the authors have used the Weka's time series analysis tool. Inside this package, they have specifically tried three different algorithms: Linear regression, Gaussian processes, and Support Vector Machine for regression. The best result was achieved with the Gaussian algorithm, reaching 3.78% or mean absolute error. To validate the model, they have compared the predictions with real data of new collections.

Chapter 3

Methodology

The Cross Industry Standard Process for Data Mining (CRISP-DM) was used as the main methodology while developing the project. Also, this work was implemented using Python (Python Software Foundation (n.d.)) and its most commonly used libraries for data science, such as scipy (Virtanen et al. (2019)), scikit-learn (Pedregosa et al. (2011)), pandas (McKinney (2010)), numpy (Van Der Walt, Colbert, and Varoquaux (2011)), XGBoost (Chen and Guehrouh (2016)) and Keras (Chollet et al. (2015)) all used on jupyter notebooks (Pérez and Granger (2007)) in a Google Colab instance.

The first candidates for training the data were neural networks and the XGBoost algorithms (Gama (2017)). Those algorithms are widely used and reported in the literature. Also, the concept of the sliding windows will be used, as it was already seen to improve the accuracy of this particular problem (Londres et al. (2020)). When reporting, probably the most appropriate way is using histograms, as there are more than one hundred single systems to be developed and reported.

3.1 Dataset

The dataset available for this work was provided by the company which is responsible for the waste collection in northern Portugal, LIPOR. It comprehends two distinct cities, hereafter city A and city B, summing 5 collection circuits. The oldest record dates to March of 2017, while the newest dates to January of 2020. For city A, there are annotations relative to a total of 388 circuits which have been executed, representing 198 collection points and 24,795 observations. For city B, the biggest, the circuits were run 880 times, involving 256 customers and 48,170 single observations. Therefore, there is available almost 73,000 observations, representing 534 different customers. Those numbers are almost the triple of the data available for the previous works (de Freitas Filipe (2018), Londres et al. (2020)).

The main objective in this work is to create a intelligent predictor for each single customer. As there are more than 500 different customers, there will be more than 500 unique predictors. In order to do that, the dataset will be divided by each customer id, therefore creating up to 534 subsets. This number is not equal the total amount of customers due to the fact that not even all customer have data registered, of there is not enough data which can be used. It is decided and filtered along the Python script.

In the beginning, data was gathered by the employees responsible for the waste collection, registering it manually. Nowadays, this information is acquired electronically. So, fewer errors, mainly the missing information, should appear in the dataset.

3.2 Data Exploration

Data exploration is the first step of any data science project, knowing what kind of data and its characteristics are fundamental to have the first insights and decide how to tackle it.

The features originally available in the dataset were:

- Type of waste collection
- Municipality
- Circuit number
- Circuit designation
- Date
- Weekday
- Month
- Work shift - morning/afternoon
- Customer's ID
- Sequential number
- GPS coordinates
- Type of Business
- Quantity of bags by type - paper, plastic and, glass.

Some of those variables were discarded due to the approach which was taken. As aforementioned, the main idea is to build one predictor for each of the customers. Therefore, some data will be repeated, or will not vary, as they are sometimes correlated to the customer, such as Municipality and Customer's ID, for instance.

3.3 Data Preprocessing

Some of the variables were not used. Despite having two different cities, A and B, and some variables have difference when comparing the two cities, they do not vary when compared in the customer level. As was decided to create one predictor per customer, there will be no difference among those variables: Type of waste collection, Municipality, Circuit number, Circuit designation, Customer's ID, Sequential number, GPS coordinates and Type of business. Therefore, they were removed.

Another three categorical features were introduced to the model:

- Climate - cold, warm or hot;
- Week number - 1, 2, 3 or 4;
- Proximity to a holiday or extended weekend - yes or no.

3.3.1 Dealing with missing information (NA's)

In the beginning of this dataset collection, in 2017, it was manually done by the persons responsible for the waste gathering, namely the truck driver and his assistant. As their job involves other demanding tasks, sometimes the focus of these professionals in assuring the quality of the data, which could lead to some data inconsistency, or missing information. Nowadays, this process was semi-automated, there is a tablet available for the data collection, which still relying in the truck driver and his assistant to complete, but comes with pre-filled data (such as Customer ID, address, date), and turned the job simpler for those workers. First, it was observed two different behaviors: sometimes there was some kind of waste to be collected, which was properly annotated while other types were left in blank. For instance: there was 2 containers of paper, 1 of plastic and the no information for glass. Or there is no information at all. Both examples can be seen in Fig 3.1

For the first situation, where only one of the fields was left in blank, it was adopted the following strategy: the missing value was set as 0 (zero). For the second case, where no information was registered, the whole row was eliminated.

date	weekday	month	shift	latitude	longitude	activity	Paper	Plastic	Glass
18.01.03	wednesday	Janeiro	Manha	41.34428	-8.7390069	OUTROS	2	1	
18.01.04	thursday	Janeiro	Manha	41.34428	-8.7390069	OUTROS			

Figure 3.1: Example of missing data.

3.3.2 Dealing with outliers

One of the most important aspects of the data preparation is to investigate the presence of outliers in the dataset. And, if they are present, decide what to do with them. Analyzing 5 randomly picked customers, it was traced a box-plot for each of them, presented in the Figure 3.2.

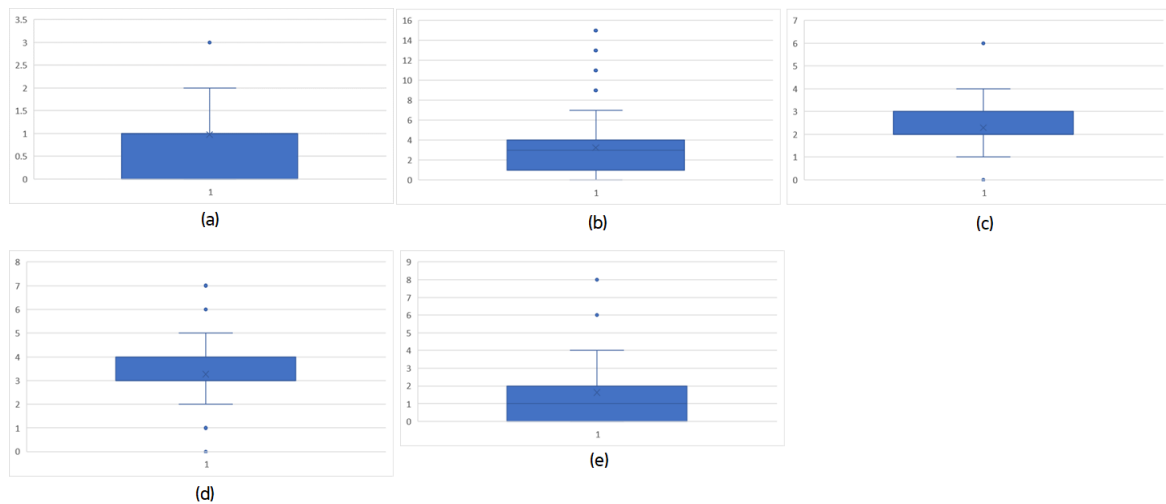


Figure 3.2: Box-plot of 5 randomly picked customers. (a) 2% outliers, (b) 5% outliers, (c) 1% outliers, (d) 5% outliers and (e) with 6% outliers.

After the analysis, it was seen that there were about 4% were outliers, with more than 1.5 standard deviation above the 75% quartile or 1.5 standard deviation below 25% quartile. Therefore, it was decided to remove the outliers, as they were representing a small fraction of the whole dataset.

3.3.3 Removing inactive customers

Among the 534 customers, some registers were too small to be used for training a dedicated model. In other words, there were not enough data about some customers, as an example, customers with only 2 records. So, it was adopted the criteria of 4 records per month as the minimum necessary for the customer to be considered active.

It can be explained due to the fact that some customers may start being part of the program, and then changing their minds or use the service only for a short period of testing. Regardless of the reason, those customers were considered inactive and removed from the dataset.

3.3.4 Dataset characteristics

When taking in mind the source of this dataset, it is interesting to have into consideration some traces of seasonality. And, if present, how does those traces may influence the behavior of the dataset. This subject was already explored in Londres et al. (2020), and there were not found any clear evidence of seasonality but the influence of summer months.

In a deeper investigation, it could be seen that this trend does exist sometimes, however, it does not seems to be representative among the customers. First, restaurants and bars were separated from other types of commerce, as they would be more influenced by the summer months. Then, it could be find one example where it occurs, reported in Figure 3.3, with a clear spike during the summer months.

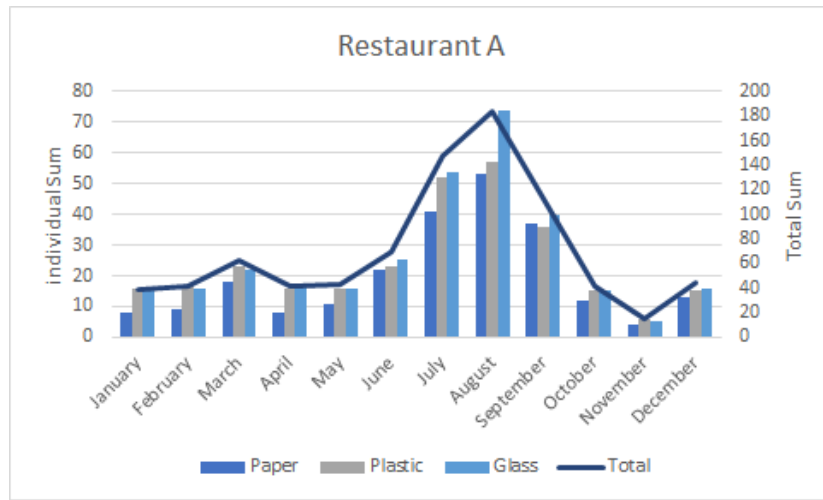


Figure 3.3: Restaurant A - it is possible to see a increasing trend during the summer months.

Continuing the analysis, when looking for other randomly picked restaurants, the trend was not followed. It can be seen in Figure 3.4, that the overall curve seems to remain stable the whole year. Moreover, when zooming in, and looking for the trends of paper and plastic (Figure 3.5) separately, despite having a sharper variations, it cannot be concluded that there is any trace of seasonality.

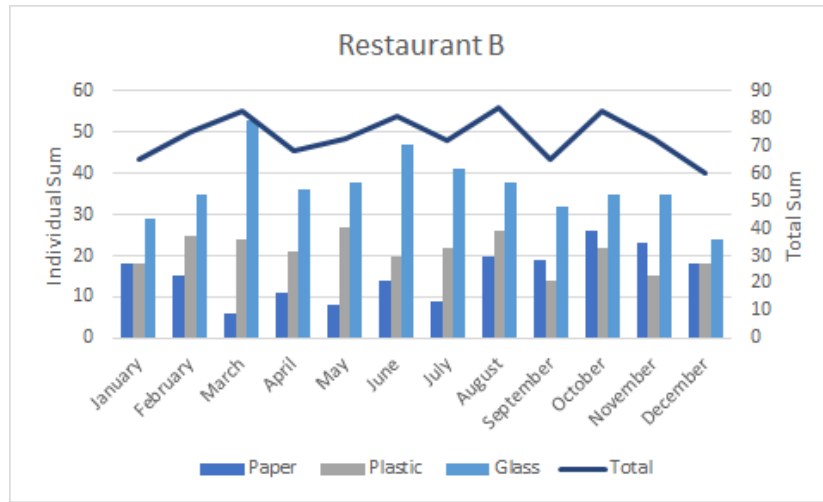


Figure 3.4: Restaurant B - There is no clear oscillation.

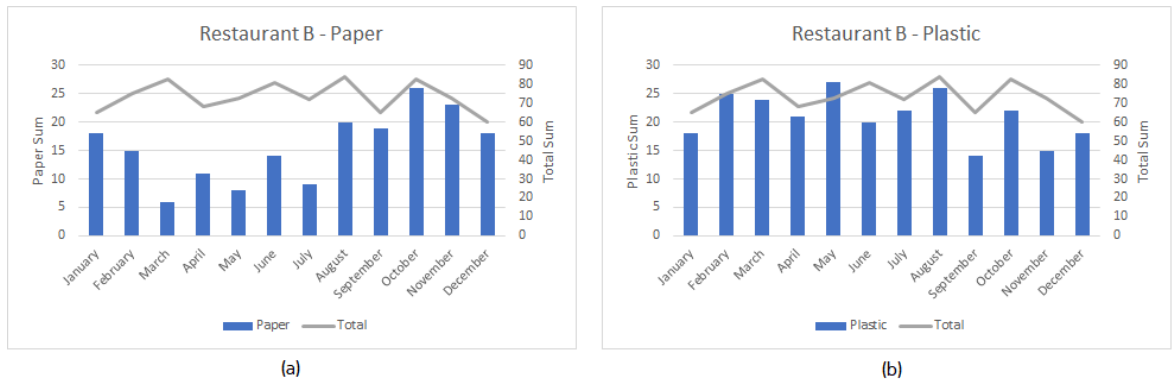


Figure 3.5: Restaurant B - Collection of (a) paper and (b) plastic containers compared with the total sum of paper, plastic and glass line.

Finally, analyzing Restaurant C (Figure 3.6), it can be seen that there was a lowering tendency from January until April, with a sudden spike in May, followed by stability. This jump in May cannot be explained just with this analysis, and it goes further the scope of this work. Nevertheless, it confirms the absence of seasonality in the dataset, regarding restaurants and bars.

There are two main reasons to analyze restaurants and bars separately from other commerce. First, the possible tendency of those establishments being more busy during summer months. Secondly, due to the nature of this business, they are more prone to produce glass waste, mainly from wine and water bottles. Therefore, there shall be no glass or it should be very occasional to other customers.

Randomly selecting 3 clients which are not restaurants/bars, it were selected a retail store, a

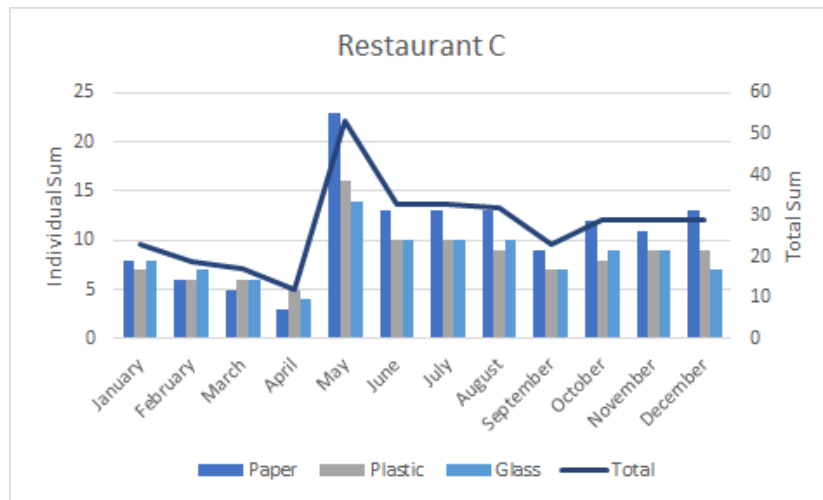


Figure 3.6: Restaurant C - After a spike in May, the trend is stable.

hotel and a mini-market. The hotel, depending on its type, such as a hotel for executives or a family vacation resort, would probably see different trends in the data, probably business hotels would show spikes in the months were a family resort would present depressions. However, there is no clear trend in the analysis of this hotel, neither on the other two establishments, as can be seen in Figure 3.7.



Figure 3.7: Diverse business - no clear trace of seasonality can be seen.

3.4 Models setup

In this section it will be described all the setup used for the development of this work. There were trained a few different combinations of setups for the algorithms, the best are shown in the results section of chapter 4.

3.4.1 XGBoost

For the XGBoost the following setup was used:

- Number of estimators: 20 and 100;
- Max tree depth: 5;
- Learning rate: 0.1;
- Training/Test split: 20%, 25%, 30% and 35%;
- Number of classes: 2 [for binnary classification] and 5 [for non-binnary];
- Objective function: softmax.

3.4.2 Neural Networks

The neural networks developed in this work were generated using Keras (Chollet et al. (2015)) library in Python (Python Software Foundation (n.d.)). The Following combination of parameters were used in many combinations:

- Number of layers: 3, 5, 7, and 10;
- Number of neurons:
 - 3 layers: 21/12/5
 - 5 layers: 21/15/12/8/5
 - 7 layers: 21/18/15/12/10/8/5
 - 10 layers: 21/18/15/12/10/9/8/7/6/5
- Activation functions: ReLU and Softmax;
- Training/Test split: 20%, 25%, 30% and 35%;

- Optimizer: RMSprop
- Loss function: Categorical Cross Entropy;
- Metric: Accuracy
- Epochs: 300;
- Early stopping epochs: 6

3.4.3 Sliding Windows

The sliding windows were used in a single configuration, with 1 month as time step. It was called 1p1, in reference of the usage of 1 month of training for predicting the next one.

Chapter 4

Results and Discussion

In this chapter there will be presented the results and the discussion obtained during this work. It is divided in three different sections, the first showing the results obtained with XGBoost, the second related to neural networks and finally a discussion and comparison with previous works.

4.1 Results obtained with XGBoost

The results here presented were obtained with the use of the packet XGBoost (Chen and Guestrin (2016)) for Python, with the support of pandas dataframes (McKinney (2010)) and sklearn (Pedregosa et al. (2011)) to manipulate the dataset and make the necessary transformations. To feed the dataset for the training with XGBoost, it was necessary to convert some variables. Weekdays, month and shift were transformed into dummy variables. Dummy variables are a sort of categorical representation of a value. It is done by collecting all the possible values for that variable and combining it in a matrix of zeros and ones. To represent one variable, the matrix is then marked with one, on that column and zero on the others. It is important to avoid biases while training the algorithm, and it is one safe way to represent categorical data.

The three columns of interest were Paper, Plastic and Glass. These columns contains the number of bags, or containers, of each material left by the customers to be collected by the waste truck. These variables were not altered, however they might be normalized inside the training process.

Two ways of classification were proposed in this work, the first was the prediction of the presence or absence of containers to be collected. The second, the quantity of containers to be collected. In both cases the split between train and test sets was 30% of testing data and the adopted metrics is accuracy. It was decided that this measure would be effective and well fitted

in the subject in discussion.

4.1.1 Predicting the presence of containers

The presence of containers predicted by XGBoost algorithm is presented in this section. Here there is no difference if there were 2, 3 or more containers to be collected; All of them were reduced to 1, *i.e.*, converted to symbolize the presence of a single container.

The average of the results can be seen in the Table 4.1. Regarding the nomenclature of the models, the first digits are in reference for the number of estimators, or boosting rounds, the last digits are in reference of the cut for the training/testing split. For instance, "XGB_20_030" was trained with 20 boosting rounds and 30% of the dataset was used for testing, therefore, 70% was used for training.

Table 4.1: Average and rank of the results for each model, for each material, trained with XGBoost, for binary classification.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
XGB_20_030	0.916	1	0.915	1.5	0.959	1.5	1.33
XGB_20_025	0.915	2	0.915	1.5	0.955	6	3.17
XGB_100_030	0.914	3	0.914	3	0.957	4	3.33
XGB_20_020	0.913	4	0.912	5	0.959	1.5	3.50
XGB_100_035	0.912	5	0.913	4	0.957	4	4.33
XGB_20_035	0.911	6	0.909	6	0.957	4	5.33

As the intent of this work is to build a classifier for each customer, there is a histogram of model XGB_20_030 with predictions relative to paper containers in Figure 4.1, which cuts may be seen in the Table 4.2. The idea here is to understand how is the percentage of clients having elevated scores.

Table 4.2: Training with XGB to predict the presence of containers - accuracy obtained.

	Accuracy
>95%	46.6
>90%	64.8
>85%	71.2
>80%	78.5

It can be seen that 78.5% of all customer are able to reach more than 80% of accuracy when predicting the presence or absence of waste to be collected. In fact, observing the histogram it

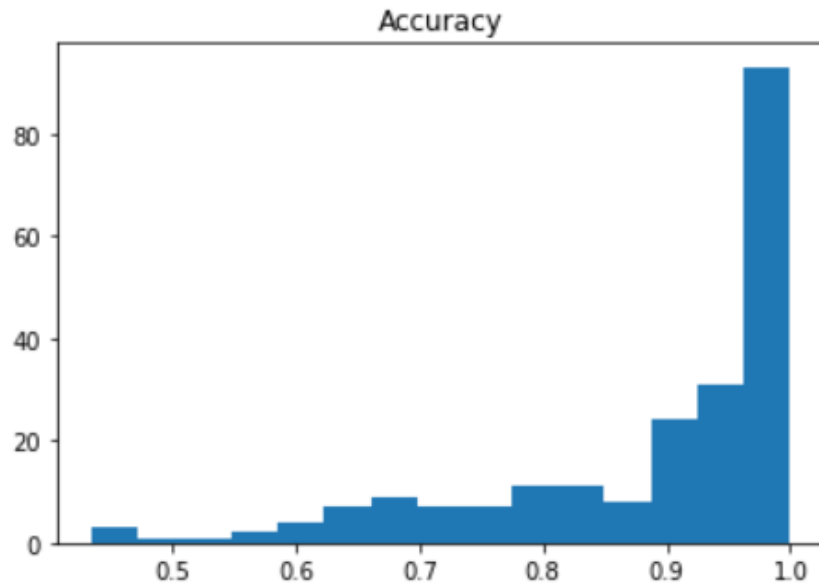


Figure 4.1: Histogram of results obtained by XGB for the presence of containers.

can be concluded that a significant amount of customers are around 90% of accurate predictions. However, the models have reached elevated values of accuracy, resulting in a mean of at least 90% accuracy.

Now, considering the sliding window technique, here are presented the results after training the best ranked model. The time step is 1 month, hereinafter called 1p1. The nomenclature of the models remains the same. The results are shown in Table 4.3.

Table 4.3: The average results for the 1p1 training of XGBoost.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
XGB_20_030	0.825	1	0.836	1	0.935	1	1.00

It is possible to see a decrease in the average results, however they still above 80% of accuracy.

4.1.2 Predicting the quantity of containers

This is the trickier part of the problem. Predict not only if there will be waste to be collected, but how many containers of each type: Paper, Plastic or Glass. The average of the results are shown bellow in Table 4.4. The nomenclature of the models, are the same as for the binary prediction, *i.e.*, the first group of digits are in reference to the number of boosting rounds and the second group in reference of the test/train split.

Table 4.4: Average and rank of the results for each model, for each material, trained with XG-Boost.

	Paper	Rank	Plastic	Rank	Glass	Rank	Avg. Rank
XGB_20_030	0.799	1	0.760	1.5	0.925	2	1.500
XGB_20_025	0.798	2	0.760	1.5	0.922	5.5	3.000
XGB_20_020	0.795	5	0.756	5	0.927	1	3.667
XGB_100_030	0.796	3.5	0.759	3	0.922	5.5	4.000
XGB_20_035	0.796	3.5	0.754	6	0.924	3.5	4.333
XGB_100_035	0.790	6	0.758	4	0.924	3.5	4.500

Also, it is possible to see a histogram with the distribution of XGB_20_030 in predictions relative to paper containers, in Figure 4.2 and in the Table 4.5

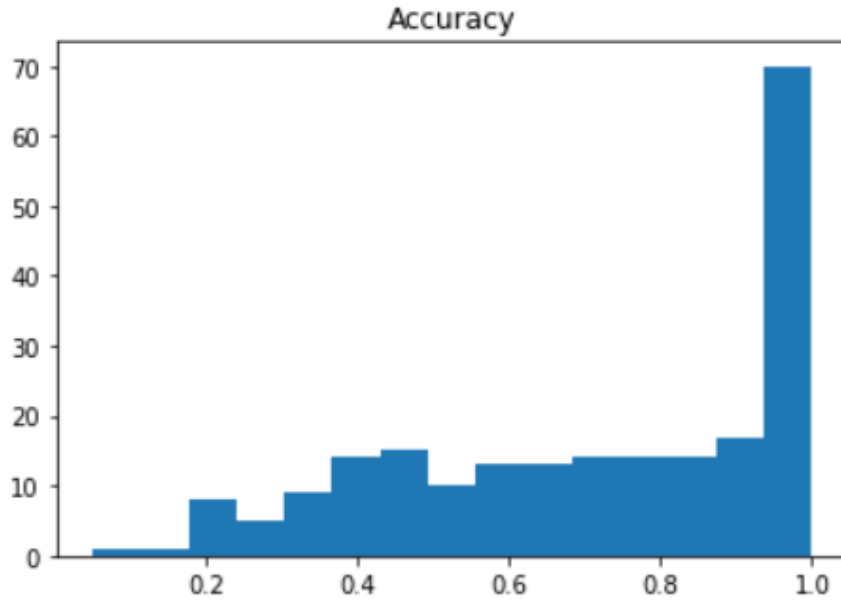


Figure 4.2: Histogram of results obtained by XGB for the quantity of containers.

Table 4.5: Training with XGB to predict the quantity of containers - accuracy obtained.

	Accuracy
>95%	31.2
>90%	36.2
>85%	41.3
>80%	48.6

Analyzing the results obtained via XGBoost algorithm, it is possible to see that scores above 80% are obtained by 48.6% of the customers. Also, based in the average of the results are very close to 80% of accuracy. These result will be properly discussed in the section 4.3.

The results for the sliding windows model is shown in Table 4.6. The time step is 1 month.

Table 4.6: The average results for the 1p1 training of XGBoost.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
XGB_20_030	0.673	1	0.643	1	0.894	1	1.00

Similar to what happened to the binary prediction, there was a decrease in the average of the results. It should be properly addresses in section 4.3.

4.2 Results obtained with Neural Networks

The neural networks used in this work were implemented using the Keras (Chollet et al. (2015)) library, with the support of pandas (McKinney (2010)) for dataframes and sklearn (Pedregosa et al. (2011)) for data preprocessing.

Similarly to what was used for the preparation of XGBoost, some variables were transformed into categorical values and also into dummy variables, which can also be known by one hot encoding. The aforementioned variables are month, weekday and shift.

A neural network can be shallow, with just 2 or 3 layers, or deep, with more than that, with no limits. However, the deeper the network heavier the training will be. In other words, the deeper the network, more computationally expensive it is. So, it was decided to start with 3 layers and increasing it up to 10 layers. Also, it was varied the percentage of data used for training and testing; from 20% up to 35%. There is no percentage defined for this division, therefore being an empirical decision.

Several tests were done, with different numbers of layers and percentage division between train and test groups. The most interesting results will be presented in the following sections. Also, there was done the division for predicting the presence of absence of waste containers to be collected and the quantity of containers of each type.

4.2.1 Predicting the presence of containers

In this section there will be presented the results for the prediction for the presence or absence of waste containers, regardless of the quantity of each one. As mentioned before, several setups were tested and the most interesting will be presented here.

The average of the results and the ranks are shown in Table 4.7. The nomenclature follows a pattern, where the first digit group is relate do the number of layers, and the second group to the train/test split. For instance, "NN_7_030" has 7 layers and was trained with 30% of the dataset for testing and, therefore, 70% of dataset for testing.

Table 4.7: Average and rank of the results for each model, for each material, trained with Neural Networks

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
NN_7_030	0.952	3.5	0.952	3.5	0.954	5	4.00
NN_10_030	0.954	1	0.954	1	0.952	13	5.00
NN_7_035	0.952	3.5	0.952	3.5	0.953	9.5	5.50
NN_10_035	0.953	2	0.953	2	0.952	13	5.67
NN_7_025	0.903	7.5	0.901	8	0.954	5	6.83
NN_3_030	0.904	5	0.903	5.5	0.952	13	7.83
NN_3_020	0.899	14.5	0.901	8	0.955	1.5	8.00
NN_10_025	0.903	7.5	0.899	13.5	0.954	5	8.67
NN_5_020	0.902	10.5	0.896	16	0.955	1.5	9.33
NN_3_025	0.903	7.5	0.903	5.5	0.951	15.5	9.50
NN_10_020	0.9	13	0.9	11	0.954	5	9.67
NN_3_035	0.899	14.5	0.901	8	0.953	9.5	10.67
NN_5_025	0.903	7.5	0.897	15	0.953	9.5	10.67
NN_5_030	0.901	12	0.9	11	0.953	9.5	10.83
NN_7_020	0.898	16	0.899	13.5	0.954	5	11.50
NN_5_035	0.902	10.5	0.9	11	0.951	15.5	12.33

As the main objective of this work is to develop a predictive model for each one of the customers, a interesting way to observe its performance is plotting a histogram, which can be seen in Figure 4.3, and its cuts in Table 4.8. The results are in reference to model "NN_7_030", prediction the presence of absence of paper containers.

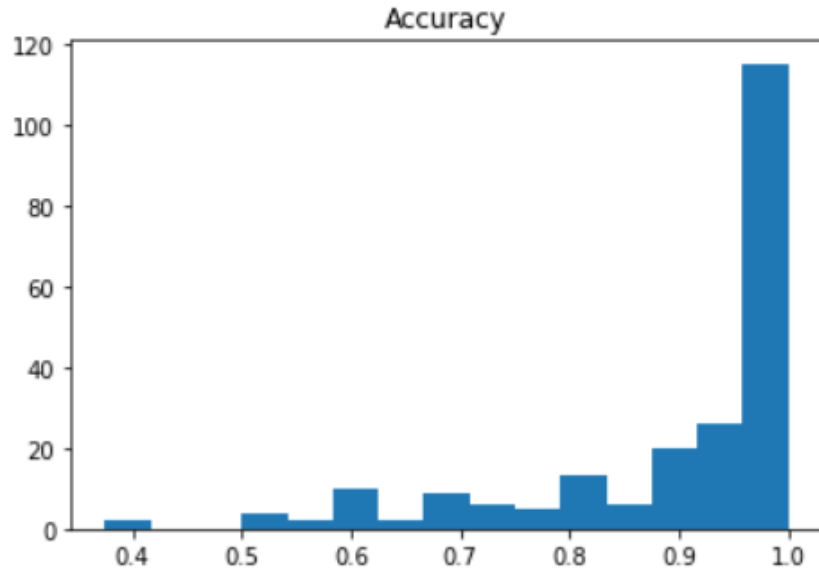


Figure 4.3: Histogram of results obtained by NN with 7 layers and 0.30 split for the presence of containers.

Table 4.8: Training with NN with 7 layers and 0.30 split to predict the quantity of containers - accuracy obtained.

	Accuracy
>95%	45.1
>90%	56.5
>85%	64.3
>80%	70.6

Now, considering the sliding window technique, here are presented the results after training the three best ranked models. The time step is 1 month, again called 1p1. The nomenclature of the models remains the same.

The results with the usage of sliding window is presented in the Table 4.9.

Table 4.9: Average results for training with neural networks 1p1.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
NN_10_030	0.628	2	0.626	1	0.221	1	1.33
NN_7_030	0.633	1	0.628	2	0.222	2	1.67

When compared to the other neural networks it can be seen that the 1p1, or sliding window with 1 month of train to next month of prediction, was considerably worst. It will be properly discussed in section 4.3.

4.2.2 Predicting the quantity of containers

In this sections are presented the results for the case where it is tried to predict the number of each containers available to be collected by the trucks. Several neural networks structures were experimented, the three best results are shown here.

The averages of the results are shown in Table 4.10. The naming of the models follows the same pattern as before, the first numbers group are the number of layers of the neural network, and the second group makes reference to the train/test split.

Table 4.10: Average and rank of the results for each model, for each material, trained with Neural Networks

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
NN_7_030	0.902	2.5	0.903	2	0.954	1	1.83
NN_7_035	0.903	1	0.901	4	0.953	2	2.33
NN_10_035	0.898	4	0.904	1	0.952	3.5	2.83
NN_10_030	0.902	2.5	0.902	3	0.952	3.5	3.00
NN_3_030	0.786	5	0.735	6.5	0.920	8	6.50
NN_5_030	0.778	8	0.738	5	0.920	8	7.00
NN_3_035	0.779	6.5	0.733	9	0.920	8	7.83
NN_7_020	0.779	6.5	0.729	12	0.921	5	7.83
NN_3_025	0.776	9.5	0.735	6.5	0.919	12	9.33
NN_3_020	0.776	9.5	0.728	14	0.920	8	10.50
NN_5_035	0.774	12	0.731	10.5	0.919	12	11.50
NN_5_025	0.773	13.5	0.734	8	0.917	15	12.17
NN_7_025	0.773	13.5	0.731	10.5	0.918	14	12.67
NN_10_020	0.769	16	0.72	16	0.920	8	13.33
NN_10_025	0.775	11	0.728	14	0.916	16	13.67
NN_5_020	0.770	15	0.728	14	0.919	12	13.67

Following, the results are presented in a form of histogram which can be seen in Figure 4.4 and Table 4.11. It is referent to model "NN_7_030" trained to predict the quantity of paper containers.

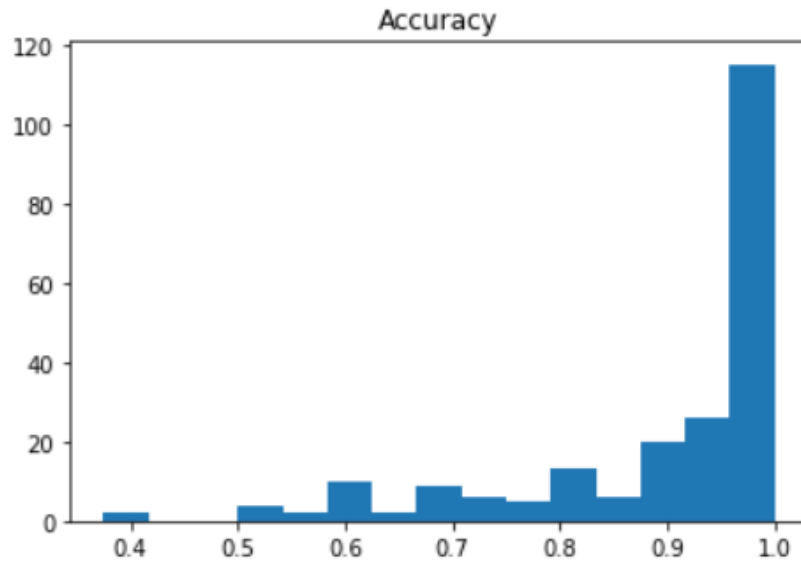


Figure 4.4: Histogram of results obtained by NN with 7 layers and 0.30 split to predict the quantity of containers.

Table 4.11: Training with NN with 7 layers and 0.30 split to predict the quantity of containers - accuracy obtained.

	Accuracy
>95%	52.3
>90%	65.5
>85%	74.6
>80%	81.8

It can be seen that almost 82% of customers have accuracy for their predictions above 80%. Also, the best model presents average results over 90%.

Regarding the sliding windows, it was set over the same configuration for the 7 layers neural network. The training split is 30%. The results are shown in Table

Again the result obtained when using the sliding window technique combined with neural networks present a result considerably smaller than the other setups. It will be properly discussed in session 4.3.

Table 4.12: Average results for 1p1 training of neural networks.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
NN_10_030	0.482	1	0.444	2	0.186	1	1.33
NN_7_030	0.479	2	0.447	1	0.184	2	1.67

4.3 Results Discussion

The results shown in the previous sections should be discussed here. Additionally, it will be compared with the results obtained in Londres et al. (2020), where this work is based on. Aside from the simply deciding if one result was good or not, it is important to try to understand the reasons which have lead to that numbers. As in the results, the two cases should be divided. the prediction of presence of waste to be collected and then the prediction of the quantity of containers.

4.3.1 Predicting the presence of waste

The first and simplest case is to predict the presence, or absence, of waste to be collected. It does not matter the quantity of it, if there are 2 or more containers of any type, it will be considered as 1. On the other hand, if there are no containers to be collected it is considered as absence of waste to be gathered.

The results obtained in this work will be presented next with the one obtained in Londres et al. (2020) in the Table 4.13.

It is possible to notice that the results from the reference Londres et al. (2020) is ranked in 10th place, when compared to the numbers obtained by the models developed in this work. Even though, the scores are very close when compared individually taking into account each material to be collected. To better understand if this difference is significant, the critical difference (CD) will be used. In this diagram, the critical difference is calculated and, based on the ranking of each model, it can be measured the distance between one particular model to another. And, if the distance is higher then the CD, then it means that the difference between those two models is significant. Otherwise, it cannot be affirmed. The confidence level used for this calculation was $\alpha = 0.05$, which means a 95% of confidence. Due to limitations in the algorithm used to trace the diagram, only the top 20 results are shown.

The critical difference diagram is shown in Figure 4.5, and the calculated critical difference is $CD = 6.182$.

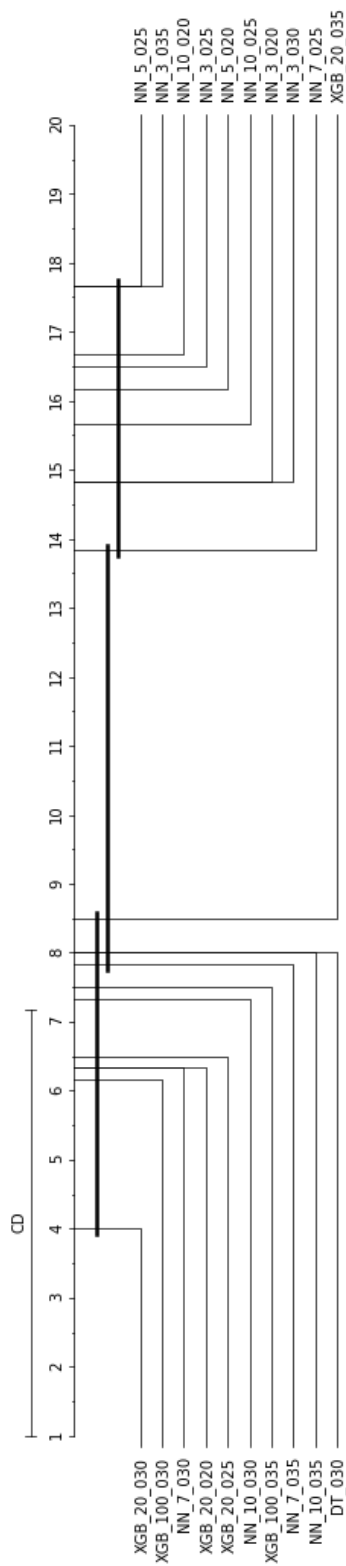


Figure 4.5: Critical Difference Diagram with models for the binary prediction.

Table 4.13: Ranked models based on average of results for the prediction of presence of waste to be collected, including the reference Londres et al. (2020).

		Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
1	XGB_20_030	0.916	5	0.915	5.5	0.959	1.5	4.00
2	XGB_100_030	0.914	7	0.914	7	0.957	4.5	6.17
3	NN_7_030	0.952	3.5	0.952	3.5	0.954	12	6.33
4	XGB_20_020	0.913	8.5	0.912	9	0.959	1.5	6.33
5	XGB_20_025	0.915	6	0.915	5.5	0.955	8	6.50
6	NN_10_030	0.954	1	0.954	1	0.952	20	7.33
7	XGB_100_035	0.912	10	0.913	8	0.957	4.5	7.50
8	NN_7_035	0.952	3.5	0.952	3.5	0.953	16.5	7.83
9	NN_10_035	0.953	2	0.953	2	0.952	20	8.00
10	DT_030	0.913	8.5	0.907	11	0.957	4.5	8.00
11	XGB_20_035	0.911	11	0.909	10	0.957	4.5	8.50
12	NN_7_025	0.903	14.5	0.901	15	0.954	12	13.83
13	NN_3_030	0.904	12	0.903	12.5	0.952	20	14.83
14	NN_3_020	0.899	21.5	0.901	15	0.955	8	14.83
15	NN_10_025	0.903	14.5	0.899	20.5	0.954	12	15.67
16	NN_5_020	0.902	17.5	0.896	23	0.955	8	16.17
17	NN_3_025	0.903	14.5	0.903	12.5	0.951	22.5	16.50
18	NN_10_020	0.900	20	0.900	18	0.954	12	16.67
19	NN_3_035	0.899	21.5	0.901	15	0.953	16.5	17.67
20	NN_5_025	0.903	14.5	0.897	22	0.953	16.5	17.67
21	NN_5_030	0.901	19	0.9	18	0.953	16.5	17.83
22	NN_7_020	0.898	23	0.899	20.5	0.954	12	18.50
23	NN_5_035	0.902	17.5	0.9	18	0.951	22.5	19.33

It can be seen that from the first until the 11th ranked model, it is not possible to affirm that there are significant difference between those models. However, it is possible to see a separation to the neural network models, which dominates the lower tier. Maybe it can be explained by the fact that this classification, or prediction, being based in simple structured data and few variables. As stated before, this first task should predict if there will be or not waste to be collected, therefore, predict if the result will be 1 or 0. Also, the data seems to be imbalanced and decision tree is an example of cost-sensitive algorithm (Ling and Sheng (2010)). As the XGBoost tends to operate on these misclassifications, it tends to occupy the top positions of the rank.

Regardless of the position of the models, it can be seen that the average accuracy achieved by all the 23 models are above 90% accuracy, which is, in fact, a very good result for the presented

problem.

When comparing the histograms for three models discussed above, it is possible to see that the two first: XGBoost and neural network with 7 layers, present very similar shapes. However, it is possible to affirm that the first one depicts a more concentrated population after 0.8 (or 80% of accuracy). At the same time, it is possible to see that the results from the decision tree has also a considerable aggregation around 80% of precision and then a spike in the vicinities of 100%. This peak is also noticed in the other two models.

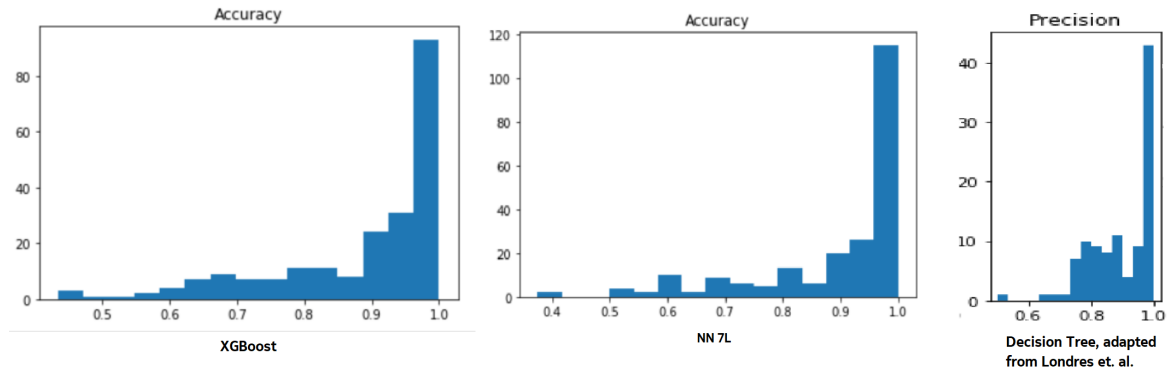


Figure 4.6: Comparison of histograms from XGBoost, Neural Network with 7 layers and Decision Tree.

4.3.2 Predicting the quantity of containers

When comparing the second topic, the number of containers which should be available for collection, the scenario is a little bit different. If in the first and simplest type of prediction the results from neural networks were the more modest, in the second case it is not observed. At least, not at the top 5 positions. Please, refer to the Table 4.14 to compare the results for the second case of classification.

Table 4.14: List with average results obtained with all models and compared to Londres et al. (2020).

		Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
1	NN_7_030	0.902	2.5	0.903	2	0.954	1	1.83
2	NN_7_035	0.903	1	0.901	4	0.953	2	2.33
3	NN_10_035	0.898	4	0.904	1	0.952	3.5	2.83
4	NN_10_030	0.902	2.5	0.902	3	0.952	3.5	3.00
5	XGB_20_030	0.799	5	0.760	5.5	0.925	6	5.50
6	XGB_20_025	0.798	6	0.760	5.5	0.922	10.5	7.33
7	XGB_20_020	0.795	9	0.756	9	0.927	5	7.67
8	XGB_100_030	0.796	7.5	0.759	7	0.922	10.5	8.33
9	XGB_20_035	0.796	7.5	0.754	10	0.924	7.5	8.33
10	XGB_100_035	0.790	11	0.758	8	0.924	7.5	8.83
11	DT_030	0.793	10	0.746	11	0.923	9	10.00
12	NN_3_030	0.786	12	0.735	13.5	0.920	15	13.50
13	NN_5_030	0.778	15	0.738	12	0.920	15	14.00
14	NN_3_035	0.779	13.5	0.733	16	0.920	15	14.83
15	NN_7_020	0.779	13.5	0.729	19	0.921	12	14.83
16	NN_3_025	0.776	16.5	0.735	13.5	0.919	19	16.33
17	NN_3_020	0.776	16.5	0.728	21	0.920	15	17.50
18	NN_5_035	0.774	19	0.731	17.5	0.919	19	18.50
19	NN_5_025	0.773	20.5	0.734	15	0.917	22	19.17
20	NN_7_025	0.773	20.5	0.731	17.5	0.918	21	19.67
21	NN_10_020	0.769	23	0.720	23	0.920	15	20.33
22	NN_10_025	0.775	18	0.728	21	0.916	23	20.67
23	NN_5_020	0.770	22	0.728	21	0.919	19	20.67

For the prediction of the quantity of containers which should be available for collection, it can be seen a slight difference between the results of the top 4 models, which are all neural networks and the following results. This difference is more pronounced when observing the scores regarding the prediction of plastic and paper. When considering glass, the results are closer.

What can be inferred here is that, for a more complex decision the neural networks performed better. As it is known, neural networks are a sort of black box (Buhrmester, Münch, and Arens (2019)), so it is not possible yet to affirm what has influenced the model to take the correct decisions. However, it was expected for it to perform better as one of its characteristics is this high sensitiveness (Novak, Bahri, Abolafia, Pennington, and Sohl-Dickstein (2018)).

The results from the model developed in Londres et al. (2020) is ranked 11th in comparison to the ones from this work. For a better view of the performance of the models, they will be displaced in a critical difference diagram. Again, there is a limitation for the top 20 models to be displayed in the Figure 4.7.

Finally, looking for the histograms from Figure 4.8, it is possible to see that the distribution of the population for the XGBoost model is considerably more sparse then for the neural networks results. In the latter, the data is concentrated around 90% of accuracy. A almost similar trend occurs with the histogram from the decision tree, it is possible to note that the values are more concentrated after the 70% value.

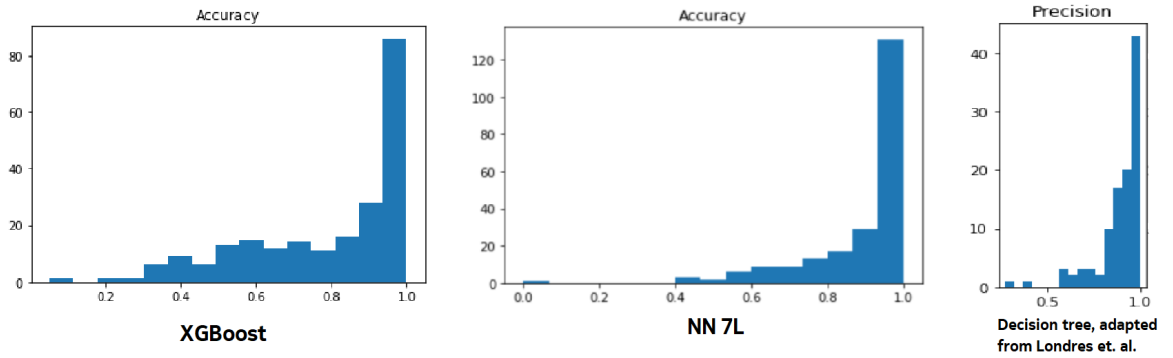


Figure 4.8: Comparison of histograms from XGBoost, Neural Network with 7 layers and Decision Tree.

4.3.3 Results of Sliding Windows

The idea of sliding windows are promising for the case like the one being studied here. Specially for the situation where there is not too much data available of, the total opposite, when there are models trained in data collected a few years ago. As the customers are all business as well, it can depict a very dynamic scenario.

When analyzing the results obtained in for the sliding windows in this work, which were applied on top of the structure developed for the neural networks, with 7 and 10 layers, the numbers are considerably low. Although, the results obtained for the 1p1 on top of XGBoost model bring lower than the "non-1p1", it does still with average above 80% of accuracy. The results can be seen side by side in the Table 4.15 below.

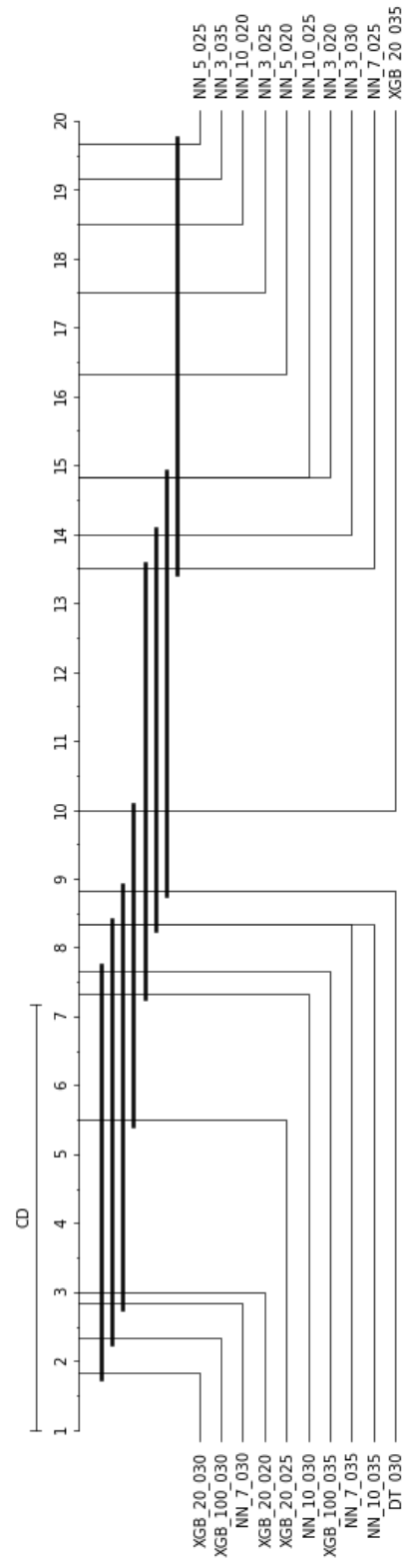


Figure 4.7: Critical Difference Diagram with models for the quantity of containers prediction.

Table 4.15: Average results for binary prediction, 1p1.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
XGB_20_030	0.825	1	0.836	1	0.935	1	1.00
NN_7_030	0.633	2	0.628	3	0.222	2	2.00
NN_10_030	0.628	3	0.626	2	0.221	3	2.33

Considering the second problem, the prediction of the quantity of containers to be collected, all models perform even worst. Again, what is more pronounced are the results from the neural networks. Those scores can be seen in Table 4.16.

Table 4.16: Average results for quantity prediction, 1p1.

	Plastic	Rank	Paper	Rank	Glass	Rank	Avg. Rank
XGB_20_030	0.673	1	0.643	1	0.894	1	1.00
NN_7_030	0.482	2	0.444	3	0.186	2	2.00
NN_10_030	0.479	3	0.447	2	0.184	3	2.33

The lower scores in the 1p1 models are fairly smaller considering XGBoost model for the binary prediction problem, and considerably poor for the other scenarios. The reason for that is probably the volume of data available for the training. As it is only considering the time step of 1 month, and the majority of waste collections happens 3 times per week, it would in average result in 12 records per month. What is, in fact, a very low amount of data.

4.4 The Best Model

If there is a way to elect the best model would be ranking the mean average of all results, considering both cases: the binary prediction, which should preview if there will be waste to be collected or not, and the second scenario, where it should forecast the quantity of containers that will be available for collection for each one of the materials: paper, plastic and glass. Also, to analyze if they can be considered different, it should be used measured the critical difference.

Here are presented the rank with all models, considering both cases. Please, refer to Table 4.17.

Table 4.17: Final rank with all models, considering both scenarios.

Rank	Model	Avg. Rank	Rank	Model	Avg. Rank
1	NN_7_030	4.08	13	NN_5_030	15.92
2	XGB_20_030	4.75	14	NN_3_020	16.17
3	NN_7_035	5.08	15	NN_3_035	16.25
4	NN_10_030	5.17	16	NN_3_025	16.42
5	NN_10_035	5.42	17	NN_7_020	16.67
6	XGB_20_025	6.92	18	NN_7_025	16.75
7	XGB_20_020	7.00	19	NN_10_025	18.17
8	XGB_100_030	7.25	20	NN_5_020	18.42
9	XGB_100_035	8.17	21	NN_5_025	18.42
10	XGB_20_035	8.42	22	NN_10_020	18.50
11	DT_030	9.00	23	NN_5_035	18.92
12	NN_3_030	14.17			

The critical difference diagram for the models, can be found in Figure 4.9. Combining the information from the ranking and the diagram, it cannot be assured that there is difference between the first and the 11th models. Therefore, it is not possible to select a best mode with 100% of certainty.

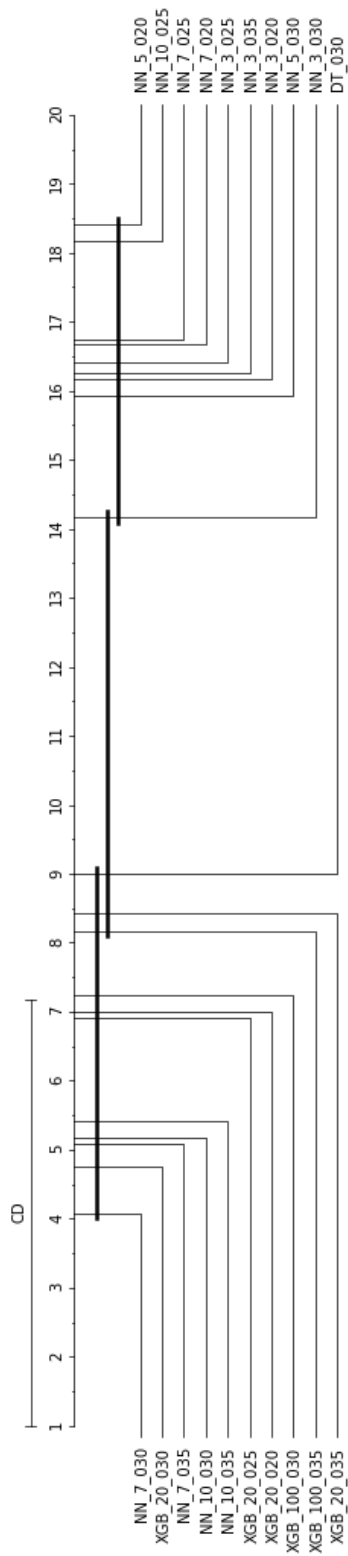


Figure 4.9: Critical Difference Diagram for all models, considering both cases.

Chapter 5

Conclusions

Analyzing the results obtained by the XGBoost models and the neural networks with 7 layers and 30% split, for the two cases: predicting the presence of waste to be collected and predicting the number of containers, shows that it is possible to have a unique predictor for each customer. Where at least 80% of the uses will have more than 80% of accuracy. Despite proposing an enhancement of the scores obtained by the 1p1 model cited in Londres et al. (2020), it can be concluded that those results were substantially high. As per using more sophisticated algorithms it cannot be affirmed that any of the model developed in the scope of this work performs considerably better than the one developed at Londres et al. (2020).

The results with the sliding windows do not match what was expected, *i.e.*, to have increase the scores when compared with its base models (as in Londres et al. (2020)); and there is no clear explanation for the cause of this disparity. Therefore it should be more investigated.

However, a valuable comparison could be made. Three different algorithms, from the simple decision trees to complex XGBoost were used to achieve better results, scoring higher accuracy, on a particular problem. Clearly, the results obtained by the proposed models are able to answer the questions if there will be waste and how many. In conclusion, the results obtained by Londres et al. (2020) with the 1p1 model were good enough. Nevertheless, the application and investigation of other models show but these models are solid and have reliable performance.

Even though higher results are desired, the trade-off may not be worthily. Use this forecasting system with a error margin for the capacity of the trucks probably will suffice. Therefore the proposed task of predicting the presence of waste to be collected and predicting the number of containers to be collected was accomplished and it was fruitful. It can be said that the routes are possible to be optimized, the trucks may save fuel and diminish its environmental footprint at the same time as they turn the peer-to-peer non-residential waste collection more lucrative.

5.1 Future Works

Aiming to improve this work, the following topics are therefore suggested:

- Have a closer look to the customers which are performing below 60%. Despite having a particular amount of customers with little data available, there may exist others with a regular collection of data but are scoring poorly;
- Study carefully the possible reasons for the difference in the results with the sliding windows.

Bibliography

- Abou Omar, K. B. (2018). Xgboost and lgbm for porto seguro's kaggle challenge: A comparison. *Preprint Semester Project*.
- Alzate, C., et al. (Eds.). (2019). *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings* (Vol. 11329). Cham: Springer International Publishing. Retrieved 2020-01-15, from <http://link.springer.com/10.1007/978-3-030-13453-2> doi: 10.1007/978-3-030-13453-2
- Arribas, C. A., Blazquez, C. A., & Lamas, A. (2010, April). Urban solid waste collection system using mathematical modelling and tools of geographic information systems. *Waste Management & Research*, 28(4), 355–363. Retrieved 2020-01-15, from <http://journals.sagepub.com/doi/10.1177/0734242X09353435> doi: 10.1177/0734242X09353435
- Buhrmester, V., Münch, D., & Arens, M. (2019, November). Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. *arXiv e-prints*, arXiv:1911.12116.
- Calabrò, P. S., & Komilis, D. (2019). A standardized inspection methodology to evaluate municipal solid waste collection performance. *Journal of Environmental Management*, 246, 184 - 191. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0301479719307704> doi: <https://doi.org/10.1016/j.jenvman.2019.05.142>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/2939672.2939785> doi: 10.1145/2939672.2939785
- Chollet, F., et al. (2015). *Keras*. GitHub. Retrieved from <https://github.com/fchollet/keras>
- Dat, L. Q., Truc Linh, D. T., Chou, S.-Y., & Yu, V. F. (2012, June). Optimizing reverse logistic costs for recycling end-of-life electrical and electronic products. *Expert Systems with Applications*, 39(7), 6380–6387. Retrieved 2020-01-15, from <https://linkinghub.elsevier.com/retrieve/pii/S0957417411017027> doi: 10.1016/j.eswa.2011.12.031

- de Freitas Filipe, N. M. (2018). *Smart cities - agendamento dinamico de recolhas de residuos porta-a-porta*. Retrieved from <https://hdl.handle.net/10216/116449>
- Demšar, J. (2006, December). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7, 1–30.
- Desarda, A. (2019, January). *Understanding AdaBoost*. Retrieved 2020-09-24, from <https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe>
- Eremia, M., Toma, L., & Sanduleac, M. (2017). The smart city concept in the 21st century. *Procedia Engineering*, 181, 12–19. Retrieved 2020-01-24, from <https://linkinghub.elsevier.com/retrieve/pii/S1877705817309402> doi: 10.1016/j.proeng.2017.02.357
- Ferrer, J., & Alba, E. (2019). Bin-ct: Urban waste collection based on predicting the container fill level. *Biosystems*. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0303264718301333> doi: <https://doi.org/10.1016/j.biosystems.2019.04.006>
- Friedman, J. H. (2001, 10). Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5), 1189–1232. Retrieved from <https://doi.org/10.1214/aos/1013203451> doi: 10.1214/aos/1013203451
- Gama, J. (2017). *Extração de Conhecimento de Dados: Data Mining* (3rd ed.). Lisboa: Edições Sílabo. (OCLC: 1026289946)
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (<http://www.deeplearningbook.org>)
- Haykin, S. (2009). *Neural networks and learning machines* (No. v. 10). Prentice Hall. Retrieved from https://books.google.com.br/books?id=K7P36lKzI_QC
- ISWA. (2017). *Iswa: Press release: The future of waste management: Results of iswa's global survey* (Tech. Rep.). LIPOR. Retrieved 2020-01-24, from <https://www.iswa.org/home/news/news-detail/browse/12/article/press-release-waste-management-will-be-transformed-by-new-technologies/109/>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon et al. (Eds.), *Advances in neural information processing systems 30* (pp. 3146–3154). Curran Associates, Inc. Retrieved from <http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf>
- Ling, C., & Sheng, V. (2010, 01). Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*.
- LIPOR. (2019, September). *Relatórios Mensais - Setembro 2019* (Tech. Rep.). LIPOR. Retrieved 2020-01-22, from <https://portal.lipor.pt/pls/apex/observatorio/v1/imagem/7>
- Londres, G., Filipe, N., & Gama, J. (2020). Optimizing waste collection: A data mining approach.

- In P. Cellier & K. Driessens (Eds.), *Machine learning and knowledge discovery in databases* (pp. 570–578). Cham: Springer International Publishing.
- Lusa, A. (2019, July 19). *Reciclagem cresceu 11% em Portugal no primeiro semestre do ano*. Retrieved 2020-01-10, from <https://observador.pt/2019/07/19/reciclagem-cresceu-11-em-portugal-no-primeiro-semester-do-ano/>
- Martinho, G., Gomes, A., Santos, P., Ramos, M., Cardoso, J., Silveira, A., & Pires, A. (2017, March). A case study of packaging waste collection systems in Portugal – Part I: Performance and operation analysis. *Waste Management*, 61, 96–107. Retrieved 2020-01-30, from <https://linkinghub.elsevier.com/retrieve/pii/S0956053X17300326> doi: 10.1016/j.wasman.2017.01.030
- Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). Boosting algorithms as gradient descent. In *Proceedings of the 12th international conference on neural information processing systems* (p. 512–518). Cambridge, MA, USA: MIT Press.
- McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. , 5(4), 115–133.
- McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (p. 51 - 56).
- Moskvitch, K. (2019, May). Turns out coffee pods are actually pretty good for the environment. *Wired UK*. Retrieved 2020-01-24, from <https://www.wired.co.uk/article/coffee-pods-nespresso-recycling>
- Moura Junior, N. N. (2014). *Modelo neural para correção da estimação de energia para um detector de partículas de propósito geral* (Dissertação de M.Sc.).
- Navghane, S. S., Killedar, M. S., & Rohokale, V. M. (2016). IoT based smart garbage and waste collection bin. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, 5(5), 1576–1578. Retrieved from <http://www.gaingon.net/pdf2016/1451532530065381.pdf>
- Ng, A. Y. (2004). Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on machine learning* (p. 78). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/1015330.1015435> doi: 10.1145/1015330.1015435
- NOS. (2019). *Vencedores eurekaathon propõem soluções para combater isolamento e exclusão social*. Retrieved 2020-01-25, from </noticias/vencedores-eurekaathon-propoe-solucoes-para-combater-isolamento-e-exclusao-social/>
- Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., & Sohl-Dickstein, J. (2018, June). Sensitivity and Generalization in Neural Networks: an Empirical Study. *arXiv:1802.08760 [cs, stat]*.

- Retrieved 2020-09-21, from <http://arxiv.org/abs/1802.08760> (arXiv: 1802.08760)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pérez, F., & Granger, B. E. (2007, May). IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3), 21–29. Retrieved from <https://ipython.org> doi: 10.1109/MCSE.2007.53
- Python Software Foundation. (n.d.). Python 3.6.9rc1 documentation [Computer software manual]. Retrieved from <https://docs.python.org/3.6/>
- Rahman, S., Irfan, M., Raza, M., Ghorri, K., Yaqoob, S., & Awais, M. (2020, 02). Performance analysis of boosting classifiers in recognizing activities of daily living. *International Journal of Environmental Research and Public Health*, 17, 1082. doi: 10.3390/ijerph17031082
- Reinstein, I. (2017). *XGBoost, a Top Machine Learning Method on Kaggle, Explained*. Retrieved 2020-09-22, from <https://www.kdnuggets.com/xgboost-a-top-machine-learning-method-on-kaggle-explained.html/>
- Sarkar, K. (2018, May). *ReLU : Not a Differentiable Function: Why used in Gradient Based Optimization?* Retrieved 2020-09-22, from <https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec>
- Sniecek, J., & Bodin, L. (2006, April). Using mixed integer programming for solving the capacitated arc routing problem with vehicle/site dependencies with an application to the routing of residential sanitation collection vehicles. *Annals of Operations Research*, 144(1), 33–58. Retrieved 2020-01-15, from <http://link.springer.com/10.1007/s10479-006-0006-y> doi: 10.1007/s10479-006-0006-y
- Teixeira, J., Antunes, A. P., & de Sousa, J. P. (2004, November). Recyclable waste collection planning—a case study. *European Journal of Operational Research*, 158(3), 543–554. Retrieved 2020-01-22, from <https://linkinghub.elsevier.com/retrieve/pii/S0377221703003795> doi: 10.1016/S0377-2217(03)00379-5
- Tung, D. V., & Pinnoli, A. (2000, September). Vehicle routing–scheduling for waste collection in Hanoi. *European Journal of Operational Research*, 125(3), 449–468. Retrieved 2020-01-15, from <https://linkinghub.elsevier.com/retrieve/pii/S0377221799004087> doi: 10.1016/S0377-2217(99)00408-7
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2), 22.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... Contributors, S. . . (2019, July). SciPy 1.0—Fundamental Algorithms for Scientific Computing

in Python. *arXiv:1907.10121 [physics]*. Retrieved 2020-01-10, from <http://arxiv.org/abs/1907.10121> (arXiv: 1907.10121)