

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data Mining for Studying Drug Interactions and Adverse Effects Prediction

Diogo Rafael Amorim Mendes



Master in Informatics and Computing Engineering

Supervisor: Rui Camacho

February 27, 2023

Data Mining for Studying Drug Interactions and Adverse Effects Prediction

Diogo Rafael Amorim Mendes

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

President: Prof. Henrique Daniel de Avelar Lopes Cardoso

Referee: Prof. Rui Carlos Camacho de Sousa Ferreira da Silva

Referee: Prof. Carlos Manuel Abreu Gomes Ferreira

February 27, 2023

Abstract

It is common, especially in elderly people, for a patient to be prescribed multiple drugs to treat one or more health conditions. In these situations, it is possible that the molecules that constitute the active ingredient of each medicine will interact with molecules from other ingested drugs, occasionally causing unwanted adverse reactions. The study of these reactions involves a deep understanding of its intervenients at a molecular level, and requires large amounts of medical and molecular data handling.

The main goal of this project was to develop a platform that enables the user to access and upload Adverse Drug Events data, and simultaneously use different functionalities capable of parsing, filtering and extending this data with new chemical concepts. Using this events, the user can access the platform's ability to predict the seriousness of drug interactions, or even its concrete outcome. By conducting this experiments, another objective is to identify if the origin of certain adverse events is exclusively associated with the interaction of active principles, or if certain human characteristics can also facilitate the occurrence of unpropitious reactions.

Our approach is to get access to this data and find useful ways to complement it by adding a panoply of functionalities to the platform. At a later stage, Data Mining algorithms can be applied to these enhanced datasets that include information from real reported incidents. Initially, it is important to assess the relevant information included in the data, which means collecting values that one expects to be essential in predicting the consequences of drug interactions that lead to these adverse effects. Further on, the goal will be transforming some of this content into new data that might provide further insight and knowledge to what is unfolding behind the curtains of these events, but also provide the user with tools that can be used independently for other medical or even chemical purposes.

Despite the existence of knowledge about a large number of interactions, scientific fields are constantly evolving, which means that new breakthroughs are always happening, that lead to the development of original drugs. This, consequently reinforces that an exhaustive study of these drug interactions would be an immense task, which supports the reason why predicting unknown interactions is vital. A platform capable of comprising various useful tools in one place will certainly be a significant contribution to the medical or pharmaceutical fields, and even chemistry.

Resumo

É comum, especialmente em idosos, que um paciente seja prescrito com múltiplos medicamentos para tratar uma ou mais condições de saúde. Nestas situações, é possível que as moléculas que constituem o ingrediente ativo de cada medicamento interajam com moléculas de outros medicamentos ingeridos, ocasionalmente causando reações adversas indesejadas. O estudo dessas reações envolve uma compreensão profunda dos intervenientes a nível molecular e requer uma grande quantidade de manipulação de dados médicos e moleculares.

O principal objetivo deste projeto foi desenvolver uma plataforma que permita ao utilizador comunicar e enviar dados de Eventos Adversos de Medicamentos e, simultaneamente, usar diferentes funcionalidades capazes de analisar, filtrar e estender esses dados com novos conceitos químicos. Utilizando estes eventos, o utilizador pode usufruir da capacidade da plataforma de prever a gravidade das interações medicamentosas ou até mesmo o desenlace em concreto das mesmas. Ao realizar essas experiências, outro objetivo é identificar se a origem de certos eventos adversos está exclusivamente associada à interação de princípios ativos ou se certas características humanas também podem facilitar a ocorrência de reações desfavoráveis.

A nossa abordagem passa por obter acesso a esses dados e encontrar maneiras úteis de complementá-los, adicionando uma variedade de funcionalidades à plataforma. Em uma fase posterior, algoritmos de Data Mining podem ser aplicados a esses conjuntos de dados aprimorados que incluem informações de incidentes reais relatados. Inicialmente, é importante avaliar as informações relevantes incluídas nos dados, o que significa colecionar valores que se espera serem essenciais na previsão das consequências das interações medicamentosas que levam a esses efeitos adversos. Mais adiante, o objetivo será transformar parte desse conteúdo em novos dados que possam fornecer mais perspectivas e conhecimento sobre o que está a acontecer atrás dos bastidores desses eventos, mas também fornecer ao utilizador ferramentas que possam ser usadas independentemente para outros fins médicos ou mesmo químicos.

Apesar da existência de conhecimento sobre um grande número de interações, os campos científicos estão em constante evolução, o que significa que novas descobertas estão sempre a acontecer, o que leva ao desenvolvimento de novos medicamentos. Isso, consequentemente, reforça que um estudo exaustivo dessas interações medicamentosas seria uma tarefa hercúlia, o que apoia a razão pela qual prever interações desconhecidas é vital. Uma plataforma capaz de reunir várias ferramentas úteis em um só lugar certamente será uma contribuição significativa para os campos médico ou farmacêutico, e até mesmo químico.

Acknowledgments

First and foremost, this dissertation would not have been possible without the guidance and support of many individuals, and I want that to echo. Reflecting on the past semester, I would like to express my deepest gratitude to my advisor, Professor Rui Camacho, who helped me to the maximum. Yielding, Professor Camacho was always supportive throughout this project, with his expertise in the field of data mining being instrumental in the development of the Tamingo platform, including his skills in leadership. Researching and building this project became a lot easier with his availability and willingness to aid at any time.

The Faculty of Engineering of the University of Porto (FEUP) also played an enormous role, and I would also like to extend my appreciation to it, for providing me with the skills and knowledge necessary to undertake such an ample project. Yet, despite the faced challenges, I was able to persevere thanks to the education and training I received throughout my years at FEUP, an institution I am sure will always be a great pillar in my professional career, and whose help was essential.

Additionally, I would like to express my deepest gratitude for my parents Mário and Salette, without them I would not be the man I am today and every achievement of mine is without a doubt also theirs, for they always have been so dedicated. Your son will continue to try and follow the good human values you raised me to uphold and I hope to keep making you proud, striving to always be better.

Another big part of my life also deserves praise, so last but definitely not least, a word to my friends for their unwavering support, and a special heartfelt shout-out to my other half for the unconditional encouragement and love throughout this journey, which I am sure will only grow.

A sincere thank you to all, for helping me up until the end, in a way no one else can!

Diogo Mendes

*“The future belongs to those
who believe in the beauty of their dreams.”*

Eleanor Roosevelt

Contents

1	Introduction	1
1.1	Related Work	1
1.2	Context	2
1.3	Goals	2
1.4	Structure	3
2	Data Mining and Adverse Drugs Effects	4
2.1	Health Informatics	4
2.2	Drug Interactions	5
2.3	Relevant Web Repositories	8
2.4	Data Mining	10
2.5	Machine Learning	12
2.6	Graph Mining	15
2.7	Evaluation Metrics	16
3	A platform for ADE studies - Tamingo	21
3.1	Technologies	21
3.2	Layout and Design	22
3.3	Capabilities	23
3.4	User Controlled Data Manipulation	24
3.5	Automated Data Operations	27
4	Case Study - DM applied to ADE	33
4.1	Phase 1 - Managing OpenFDA Data	33
4.2	Phase 2 - Personalized Dataset Creation	35
4.3	Phase 3 - Data Analysis	35
5	Conclusions	43
5.1	Future Work	43
5.2	Reflection and Future Directions	44
	References	45
A	OpenFDA Data Parser	49
B	Funtional Groups SMARTS	52
C	Snippet of SDF Compiled by Tamingo	53

D	Function Converting SMILES data to SDF Format	55
E	OpenFDA Data Snippet (JSON)	56

List of Figures

3.1	Tamingo Use Case Diagram	23
4.1	Access Pre-Loaded OpenFDA Data	34
4.2	Upload OpenFDA Data	34
4.3	Filtering System	35

List of Tables

2.1	Confusion matrix example (rows: predicted values; columns: actual values) . . .	18
3.1	SMILES Syntax Representation	27
3.2	Drug Categories Examples	28
4.1	Step 1 Results	38
4.2	Step 2 Results	38
4.3	Step 3 Results	39
4.4	Step 4 Results	41

Abreviaturas e Símbolos

ADE	Adverse Drug Event
ADR/ADRs	Adverse Drug Reaction(s)
ADReCS	Adverse Drug Reaction Classification System
API	Application Programming Interface
AUC-ROC	Area Under the Curve of the Receiver Operating Characteristic
CID	PubChem Compound IDentifier
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DM	Data Mining
FAERS	FDA Adverse Event Reporting System
FDA	United States Food and Drug Administration
HI	Health Informatics
JSON	JavaScript Object Notation
K-NN	K-Nearest Neighbors
MedDRA	Medical Dictionary for Regulatory Activities
ML	Machine Learning
PV	Pharmacovigilance
SDF	Standard Data File
SIDER	SIDEffect Resource
SMARTS	Simplified Molecular Input Line Entry System Arbitrary Target Specification
SMILES	Simplified Molecular Input Line Entry System
SVM	Support Vector Machine
UMLS	Unified Medical Language System

Chapter 1

Introduction

The work reported in this dissertation demonstrates the usefulness of building a platform capable of applying Data Mining (DM) techniques to predict the outcomes of Adverse Drug Reactions (ADRs), ultimately building a less expensive and more accurate way of predicting these events from the constituents of the drugs involved, when compared to the methods of today, that rely on small sample tests, which frequently end up not correctly depicting the intended consumer, making it problematic to obtain accurate and reliable results.

Verified and trustworthy web repositories were used, namely to collect data containing real reports of drug administration, as well as the subsequently ramified ADRs. Through DM techniques, it was possible to predict outcomes based on these cases and on the structure of the molecules that identify the active ingredient of each drug.

The experiments that were done include a quantitative assessment of the generated models, in order to evaluate the quality of the forecasted adverse reactions.

1.1 Related Work

The identification and prediction of adverse drug events (ADEs) are crucial for ensuring drug safety and reducing the risk of harm to patients. With the increasing use of computational methods in pharmacology, the article "Improving the Prediction of Adverse Drug Events Using Feature Fusion-Based Predictive Network Models" [29] presents a new approach for predicting drug-ADE associations using feature fusion-based predictive network models (FFPNMs) with different machine learning (ML) methods. The authors used the Jaccard and Adamic-Adar indices to build FFPNMs with logistic regression, random forest, and support vector machine ML models. The study was based on a dataset of 152 drugs and 633 ADEs obtained from the FDA Adverse Event Reporting System (FAERS) 2010 dataset. The FFPNMs' performance was assessed using the area under the receiver operating characteristic curve (AUROC) value, with the FFPNM with RF achieving the highest predictive performance.

In the healthcare industry, data mining techniques are being increasingly applied to improve patient care, as illustrated by the article "Healthcare data mining: Predicting hospital length of

stay of dengue patients" [45]. This article proposes a data mining approach to predict inpatient length of stay of dengue patients at the time of admission, which can be used to enhance clinical decision-making and resource management in hospitals. The authors utilized the C4.5 algorithm, a decision tree classifier, to analyze a dataset of dengue patients, achieving an accuracy of 71.57% and developing a prototype prediction system to support clinicians.

Another research article, "Application of data mining data in pharmacovigilance" [44], highlights the importance of detecting potential ADEs using data mining techniques. The article points out that traditional ADE detection systems are not efficient enough, and data mining techniques such as cluster analysis, link analysis, deviation detection, and disproportionality assessment can improve the accuracy and efficiency of ADE detection.

The literature review indicates that computational approaches, including network-based predictive models and data mining techniques, have shown significant potential in predicting and identifying potential ADEs. These methods can enhance the accuracy and efficiency of pharmacovigilance and lead to the detection and prevention of ADEs. Data mining techniques can also find applications in other areas beyond pharmacovigilance, indicating their broad applicability to various fields. This dissertation aims to further explore the potential of data mining techniques in pharmacovigilance and identify how they can be used alongside other methods for ADE detection and prevention.

1.2 Context

When a drug is prescribed, it is done so with the intent of curing or treating, in most cases, a single "target"/disease. As people get older, the probability of being on any type of medication rises. Statistics show that on average an adult ingests 4 or more prescription drugs a day, with an astounding 39% of elders taking 5 or more everyday [10]. ADRs are, in its essence, unwanted reactions that differ from the expected result of a treatment. It is estimated that approximately 5.3% of hospital admissions are due to ADRs, with this number changing to 10.7% in senior patients, making it one of the most common cause of hospital admissions [23]. ADRs are also listed as being between the 4th and 6th leading causes of death, as around 200.000 deaths in the European Union are attributed to these undesirable effects [34].

Before commercialization, a pharmaceutical goes through expensive clinical trials, known for only using a small pool of people, when it is usually intended to be distributed through a much larger number of individuals [24].

1.3 Goals

For these reasons, it is important to consistently discover new ADRs after a new drug is marketed, and it is clear that there is a demand for finding alternatives that use more effective methods, capable of providing better results with the goal of diminishing risk and potentially saving human lives.

Through Health Informatics (HI) and DM, it may be possible to use the chemical potential and characteristics of drugs and their adverse effects to create predictive models for the outcome of interactions by taking into consideration the active ingredients in action, as well as some patient features, which consequently can lead to finding hints as to why and how a particular reaction may occur.

Additionally, this platform will also contain other useful features related with the manipulation of molecular characteristics of drugs, and adverse events data, with the intention of making it an all-around app for various chemical and medical purposes.

This work aims to be a step forward in completely abolishing expensive clinical trials, and transition to a world where after careful analysis of drugs at a molecular level, it is possible to accurately predict ADRs with greater efficiency than clinical tests.

1.4 Structure

In addition to the introduction, that summarily describes the motivation and objectives of this project, along with a review of existing literature, this dissertation contains 4 additional chapters.

In chapter 2 an exhaustive study of the current state-of-the-art is elaborated, and it is based on this knowledge that the following chapters are built.

Chapter 3 presents all of the characteristics of the platform that was built to achieve every proposed goal, while chapter 4 focuses on how a user is able to take advantage of said features in order to construct useful predictive models.

Lastly, chapter 5 presents a brainstorm of possible future improvements to the platform, culminating with a review of the project.

Chapter 2

Data Mining and Adverse Drugs Effects

2.1 Health Informatics

Health informatics is a multidisciplinary field of study related to the wide range of resources that can be applied in the management and use of biomedical information, including computing and the study of the nature of medical information itself. In its essence HI represents the study of information in the provision of health care. It is the analysis of how medical knowledge is created, shaped, shared and applied [18].

A variety of activities are related with HI, namely biological signals processing, medical images processing, informatics security in health, and also the area of this dissertation, data mining of medical information.

Innovations in the field of health are already a reality and are advancing more and more each day with a positive impact. These novelties bring with them a series of benefits, both for patients and professionals/organizations in the industry. Advantages can range from more accurate diagnoses to the improvement of information integration, where it is possible to collect all the important information about a patient in an electronic medical record, optimizing the work performed by the medical team, since everyone involved can access this information. At a later stage this type of medical information can be shared and provided to researchers, so that new breakthroughs may arise from it.

With the explosion of big data, there was a need and also a thirst to look for alternatives and solutions that could analyze this vast amount of data. Researchers may use machine learning techniques, along with computers and algorithms to analyze this immense amount of data, in order to find, in an easier way, solutions to the most diverse problems in this area [39].

Using this technology, the capture and collection of patient data, for example, through electronic medical records or clinical exams, occurs faster and also, after the data analysis stage, solutions are returned in the form of diagnoses for possible diseases or possible reactions to medication and their probabilities of occurrence.

Pharmacovigilance

The pharmaceutical science related to the collection, detection, evaluation, monitoring, and prevention of ADRs is called pharmacovigilance (PV). PV focuses heavily on ADRs, that is, any prejudicial and unexpected response to a drug (or sets of drugs). Medication errors such as overdose or incorrect use of a drug, and drug exposure during pregnancy are also relevant and of interest even when an adverse event is not present, since it may result in a future ADR [31].

For PV to happen, data has to exist, so an emphasis exists on how information is collected from patients and professionals in the area of health through PV agreements, and also from medical literature. Whenever a new drug is to be marketed or tested, in the great majority of countries, all the gathered adverse event data must be delivered to the corresponding drug regulatory authority.

Ultimately, PV's preoccupations are on assessing potential menaces linked with drugs and diminishing the risk of potential damage that patients might endure. Every company must conduct a thorough drug safety and PV audit to confirm that laws and regulations were followed [1]. The work done in this dissertation could aid companies in being more efficient and lighten the effort that is needed to comply with every guideline.

Adverse events reports may emerge from various sources, such as, voluntary reports from healthcare professionals and patients, patient support programs reports, clinical study reports, literature information collection, media reports, and finally transcripts reported to the drug regulatory authorities.

The collected information is coded with standard terminology from medical coding dictionaries, such as MedDRA (most commonly used), with the goal of converting adverse events statements and information into quickly identified codes.

2.2 Drug Interactions

Adverse Drug Reactions

An ADR is a prejudicial and unintended outcome caused by the use of medication. ADRs can occur after a single dose, prolonged administration of a drug or result from the combination of two or more drugs [36]. It is important to differentiate between an ADR and the term *side effect*, since the latter can be both beneficial and harmful. An adverse drug event (ADE) consists on any not expected or inappropriate occurrence, at the time a drug is taken, whether or not it is associated with the administration of a specific drug. An ADR is a specific type of ADE, since ADRs are just one type of harm related with medications, as harm can also be caused by other factors, such as failing to take prescribed pharmaceuticals [40].

When it comes to the classification of ADRs, it can be done by cause and severity.

Classification by cause can be divided into different types. For example, Type A corresponds to enhanced pharmacological effects, more specifically ones that are dose-dependent and predictable. Type A reactions constitute close to 80% of ADRs and are often a consequence of the

drug's primary pharmacological effect or a low therapeutic index (ratio comparing the blood concentration at which a drug causes a therapeutic effect to the amount that causes toxicity), making it predictable. Usually, they are mild, even though they can also turn out to be severe or fatal. More acute reactions are mostly due to inadequate dosage, particularly when drug elimination from the patient is impaired, but other possibilities should never be discarded and be properly investigated.

Regarding seriousness, the FDA (U.S. Food and Drug Administration) classifies an event as serious when the outcome of a patient corresponds to one of the following: death, life-threatening, hospitalization, some form of disability, congenital abnormality or intervention is required to block any permanent harm.

Pharmacodynamic Interactions

Drug interactions occur when the mechanism of action of a drug is disturbed by concomitant substances, for example, other medications. The cause is often the inhibition of receptors for the drug, forcing drug molecules to bind to other unintended targets, which consequently may lead to the occurrence of ADRs [35].

In depth, when free receptors are occupied by agonists (drugs that bind to, and activate receptors) and antagonists (drugs that impede activation), the probability of drugs reaching the intended receptor diminishes, since most receptors are already occupied. Therefore, when the number of free receptors lowers, drugs begin to attach to other secondary receptors, causing unintended reactions.

The risk of a drug interaction increases with the number of drugs used, which means that the 39% of elders who take 5 or more medications per day, as mentioned in the previous chapter, have a higher risk of experiencing undesired effects.

There are three types of drug interactions: additive (expected result, when adding up the effects of each drug), synergistic (greater effect than expected after drug combination, which can be beneficial at times, although it also may increase the risk of overdose), and antagonistic (lesser-than-expected effect when drugs combine). Since the individual effects of each medication may vary according to patient characteristics, it is sometimes hard to differentiate between additive and synergistic interactions [11].

Synergy and antagonism can occur during distinct phases of drug interaction with an organism. Different responses of a drug receptor have resulted in various classifications that have fundamental applications in the pharmacodynamics of these interactions. There is still little knowledge about drug mechanisms, which makes it arduous to offer a clear classification for these responses.

Direct drug interactions are also possible and can occur when two drugs are mixed prior to being administered.

A key factor in pharmacodynamic interactions is how an organism responds after administration of a drug. As mentioned, these changes are extremely difficult to classify, due to the existence of a great variety of modes of action, and that a number of drugs can produce their effects through many distinct mechanisms.

This richness in diversity, and a well-founded and widely agreed on suspicion that there are more unknown interactions than those of knowledge, calls for a need to investigate and understand these mechanisms in all cases, except the trivial ones.

Pharmacodynamic interactions can occur in three areas: drug receptors, making it the most easily defined and common manifestation; signal transduction mechanisms, which are molecular processes that start after a drug interacts with the receptor, that may lead to the surfacing of ADRs; antagonistic physiological systems, where simultaneously taken medications cause ADRs, due to the effect of a particular substance being indirectly elevated in the presence of a second [32].

Drug interactions can also have a positive impact, however negative interactions naturally generate more interest, since they are of great pathological significance and most of the time not anticipated with many not even being diagnosed.

Studying the underlying factors that may catalyze the appearance of such interactions, may lead to progress in prevention of ADRs or faster diagnosis. Advanced age (drug interactions can be affected by age related physiologic alterations), poly-pharmacy (use of multiple drugs by one person increases the probability of reaction), genetics (enzymes are synthesized by genes to metabolize drugs; some people are more or less prone to drug interactions, since the activity of these enzymes may vary due to genetic variations), and renal or hepatic diseases (quantity of drugs in the blood can be altered if the liver or kidneys don't function correctly) are some aspects that could potentiate the occurrence of ADRs [8].

Active Ingredient

The active ingredient is the biologically active ingredient in a pharmaceutical. It is of note that some drugs can possess more than one active ingredient.

The dosage form (specific configuration in which a pharmaceutical is marketed) of a drug contains the active ingredient, which is the drug itself, and excipients (inactive ingredients), which can be the tablet composition or the liquid where the active ingredient is present. Medications are fundamentally chosen for their active ingredients, with excipients being carefully chosen during development, with the intent of making the active ingredient reach the desired target in the organism in the appropriate time and with the planned proportions [5].

Generally, patients are not aware or struggle to identify the active ingredients in the medications they take, while also not being knowledgeable of what an active ingredient actually is, which might be a possible danger, particularly when taking multiple medications, as previously explained. Nowadays, online services can be accessed by anyone who wants to know the active ingredients of a vast spectrum of drugs.

Molecular Descriptors

Molecular descriptors are numerical representations of the structural features of a molecule. They are used in computational chemistry or HI to characterize and compare chemical compounds.

There are several types of molecular descriptors, including topological, constitutional, geometric, and electronic descriptors.

Topological descriptors, such as the Wiener index, describe the connectivity of atoms in a molecule. Constitutional descriptors, such as the molecular weight and number of rotatable bonds, describe the composition of a molecule. Geometric descriptors, like the molecular electrostatic potential and the spatial distribution of electron density, describe the spatial arrangement of atoms in a molecule. Electronic descriptors, for example the molecular orbital energies and electron density, describe the electronic properties of a molecule [43].

The selection of molecular descriptors for a given project depends on the specific application and the type of information required. For example, topological descriptors may be more relevant for studying the pharmacokinetics of a compound, while electronic descriptors may be more relevant for studying its pharmacodynamics.

In the field of ADE prediction, molecular descriptors play an important role in understanding the interactions between drugs and how they can lead to certain reactions. A recent study found that a combination of constitutional, geometric, and electronic descriptors were able to correctly predict toxicity with a high degree of accuracy [19].

Another study found that using a combination of topological and constitutional descriptors improved the prediction of ADEs compared to using only one type of descriptor. This highlights the importance of using a diverse set of descriptors to fully capture the complexity of drug-drug interactions [25].

Overall, molecular descriptors are a valuable tool in understanding the properties of chemical compounds and their interactions. It is important to carefully select the appropriate set of descriptors for a specific project in order to extract the most relevant information.

2.3 Relevant Web Repositories

There are several repositories of interest when it comes to finding information for known drug interactions, terminologies, ADR reports or even molecules.

OpenFDA is a repository with the main goal of creating easy access to publicly available data, to give everyone the ability of being informed. Programmers have their work facilitated, since all the data is documented, indexed and formatted in a way that makes this high-value and scalable data easier to be handled and transformed. In the creation of any application, developers can enter a public-access portal and quickly acquire and utilize this data on their projects [21].

A variety of API endpoints are available for public use, from data related with ADEs reports, labeling of drug products, to animal and veterinary data, or even food information.

With more than 427 million API calls since it was launched, openFDA is certainly a trustworthy and widely renowned repository for ADE reporting and drug products labeling and safety alerts.

PubChem is a public and open repository containing information about chemical substances and their biological activities. After its exponential growth, it has now become a point of reference

when it comes to chemical data, with their services being used by scientific communities in a variety of areas, ranging from health informatics and drug research, to chemical medicine and biology [22].

Universities, pharmaceutical companies and even government agencies contribute to PubChem, alongside more than 850 other sources.

This repository contains more than 110 million descriptions of unique chemical structures, 270 million information regarding chemical entities, over 1 million biological experiments, as well as almost 300 million biological activity data points.

Programmers are once again taken into consideration, since PubChem provides them with several ways of accessing data, including PUG-View, PUG-SOAP or PUG-REST. It is of note that the latter provides access to information on PubChem in a way that is not possible with none of the other PUG services, with It being capable of handling short, synchronous requests, with no need for an intermediate step to confirm whether that request was completed or not.

SIDER (Side Effect Resource) is a database that contains data on various drugs and their respective recorded ADRs. This information is obtained from public documents and package inserts, including details on ADR frequency, classifications for each drug and ADR, as well as links to outside sources that provide even more knowledge on, for example, drug-target relations.

Currently, it stores information on almost 6000 ADRs that are a consequence of approximately 1400 different drugs, culminating in a total of close to 140 thousand drug-ADR pairs [26].

MedDRA (Medical Dictionary for Regulatory Activities) is a standardized and clinically validated international medical terminology, used in the pharmaceutical industry by various regulatory organizations, with the objective of being used in the registration, documentation and safety monitoring of medicinal products, throughout their development, thanks to their dedicated single standardized terminology, which in the end improves the effectiveness and transparency of medical product regulation [12].

DrugBank is a free comprehensive online database that possesses information related to drugs and drug targets. It is used as a resource in health informatics, more specifically in bio and chem informatics, with detailed chemical, pharmacological and pharmaceutical data, as well as descriptions on sequence, structure and pathway of drug targets [46].

An user can search by drug, gene, reaction, pathway, target protein, category or class, and it will find detailed information on more than 14 thousand drugs.

ADReCS (Adverse Drug Reaction Classification System) is a XML database that integrates ADR and drug information from various databases of medical repositories, including FAERS, DailyMed, SIDER4.1, MedDRA, DrugBank and PubChem.

This vast amount of data originates from a range of sources, such as laboratory results, medical and pharmaceutical records or even consumer records. It is precisely because of this divergence in points of origin that ADReCS was created, with the aim of standardizing each drug and ADE description using MedDRA and UMLS as reference points. A unique ID with four digital combinations is assigned to each ADR, in a four level tree-like architecture [9].

It possesses more than 9000 ADR terms and over 2500 drugs, with a total of approximately 800 thousand drug-ADR pairs.

2.4 Data Mining

Data mining is the activity of sorting through sizable datasets, with the goal of identifying patterns and connections that may lead to solving a particular problem. Techniques and tools used in DM enable the prediction of possible events [7].

DM is a key part of data analysis and one of the main disciplines of data science, using advanced analysis techniques to acquire relevant information from datasets. Furthermore, DM is a relevant phase in the knowledge discovery in databases (KDD) process, which is a methodology in data science design to collect, process, and analyze data.

A variety of methods of statistical analysis and machine learning techniques are employed in DM, with the goal of transforming large amounts of data into purposeful insights.

Although it is more commonly applied in business environments, DM can be pertinent in any ambience where a great quantity of data is available, such as the Life Sciences (Biology, Genomics, Proteomics, Metabolomics, Medicine, among others). Nowadays, the healthcare industry produces about 30% of global data and it is forecasted that by 2025 this number will rise to 36%. Comprehending certain patterns in this segmented data with a well planned medical data mining implementation can bring great benefits, namely in the avoidance of harmful drug interactions [13].

The FDA recommends patients to contact a doctor or pharmacist before taking a new drug, but in reality the lack of time in a clinical setting makes this approach impractical. This is where DM steps in, with the power of mitigating the risk of undesired reactions. It is true that the most lethal and threatening ADRs are already known, but with novel drugs constantly being developed at a steady rhythm, the chance of human error is always lurking. Doctors and nurses, in addition to patients, would benefit from a system that could successfully analyze the chemical composition of medications and analyze research and clinical records, predicting possible harmful reactions.

The core components of DM include machine learning and statistical analysis, without disregarding the data management tasks that need to be done to assemble the data to be analyzed. Machine learning algorithms and artificial intelligence tools have undoubtedly been crucial for the automatization of the process, and simplified the mining of huge datasets [2].

When every element is in place, the process of DM goes through four different stages:

- **Selection** - All of the data that is of relevance to a particular analytical application is identified and collected. The sources of this data can greatly vary from data warehouses to web repositories or even external data sources. No matter where the data originates from, it is usually transferred to a data lake for the following stages of this process to occur.
- **Preparation** - Data proceeds to be formatted and prepared for mining, starting with data exploration, profiling and pre-processing, ending with data cleaning, to correct possible

errors and refine the quality of information. Unless the objective is analyzing raw data, data transformation will also be applied for the purpose of consistency.

- **Mining** - Choice of the most suitable data mining technique and application of at least one mining algorithm. When it comes to machine learning, algorithms should first be run on sample datasets to detect the information that is being searched for, and only then on the complete dataset. It is in this stage that patterns are recognized and knowledge obtained.
- **Interpretation** - Results are used to create analytical models to extract insights from the mined patterns. The findings are often shared through data visualization or using data storytelling techniques.

Multiple techniques can be put to use when mining data for distinct projects [20]. A common use case of DM is the recognition of patterns, and that can be empowered by various methods, namely anomaly detection, which is capable of identifying outliers in datasets. Additional DM types encompass the following:

- **Association Rule Mining** - Association rules consist of if-then statements designed to pinpoint connections between data elements. Criteria of support and confidence are applied to evaluate relationships, where support is responsible for measuring how often related elements emerge in a dataset, while confidence calculates how many times an association rule is accurate.
- **Classification** - Dataset elements are assigned to different categories, previously defined as segments of the DM process. Some classification techniques include Naive Bayes classifiers, decision trees, k-nearest neighbor and logistic regression.
- **Clustering** - Data elements sharing specific characteristics are grouped in clusters as part of DM applications. Methods incorporate hierarchical clustering, k-means clustering, and Gaussian mixture models.
- **Regression** - Finds relationships in datasets, calculating forecasted data values based on a list of variables. Some examples are linear regression and multivariate regression. It is of note that classification techniques, such as decision trees, can also be used to do regressions.
- **Sequence and Path Analysis** - Process of mining to observe patterns where a particular set of events and values will lead to later consequences.

Software and Tools

Generally, DM tools are available as part of software platforms that also possess additional data science and advanced analytical tools. DM software is capable of performing data preparation, providing predictive modeling support, contains built-in algorithms and a GUI-based development environment, and instruments for deploying models and evaluating their work.

Tools for DM are offered by a large quantity of companies, including Google, Microsoft, IBM, Oracle, RapidMiner, SAP, AWS, Tibco Software, Alteryx, Dataiku, Databricks, H2O.ai, SAS Institute, DataRobot and Knime. Open-source technologies are also available, with some examples being Orange, Weka, DataMelt, Rattle, Elki and scikit-learn. Open-source options are also distributed by a number of software companies, where Knime, for instance, merges an open source analytical platform with commercial software, in order to manage data science applications, at the same time as companies such as H2O.ai and Dataiku provide free versions of their tools.

2.5 Machine Learning

Machine Learning is a class of artificial intelligence that permits software applications to become more accurate in predicting events, without actually being directly programmed to do so. ML algorithms utilize historical data as input to predict new values [6].

ML is important for various reasons, including the fact that it brings additional insight into possible trends and patterns. Standard ML is generally classified by how a specific algorithm learns to become more meticulous in its predictions. Four basic types are of knowledge, namely supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. Depending on the data in question, software developers may opt for different approaches:

- **Supervised learning** - Algorithms with labeled training data are provided and the variables the algorithm will evaluate for correlations are defined. The input and output of the algorithm is designated.

Supervised learning algorithms are useful in tasks such as binary classification(split data in two categories), multi-class classification(select from more than two types of answers), regression modeling(forecast continuous values), or ensemble methods(merge predictions from various ML models to obtain an accurate prediction).

- **Unsupervised learning** - Algorithms train on unlabeled data, scanning datasets to look for any significant connections that can be utilized assemble data points into subsets. Predetermined elements come into play, in particular, the data where the algorithms train and the predictions or recommendations they produce.

Neural networks and most types of deep learning are unsupervised algorithms, and of use in tasks like clustering(divide the dataset in groups contingent on similarity), association mining(pinpoint batches of items in a dataset that usually appear together), anomaly detection(identify unexpected data points in a dataset), or even dimension reduction(reduce the amount of variables in a dataset).

- **Semi-supervised learning** - Combination of the previously mentioned types of ML, where an algorithm can be fed with a small quantity of mostly labeled training data, however the model has permission to explore the data as it wishes and acquire its own understanding of the dataset, such as its dimensions, which can be applied after to new and unlabeled data.

Algorithms' performance generally improves by training on labeled datasets, however the process of labeling data can be expensive and time-consuming.

Areas where semi-supervised learning thrives include machine translation(teach algorithms to translate languages without a complete dictionary of words), fraud detection(identify fraud cases with just a sample of positive examples), and labelling data(learn to apply labels to data in big sets automatically, when the algorithm is trained on small datasets).

- **Reinforcement learning** - Used to train a machine to complete a multiple step process, in which there exists explicitly defined rules. An algorithm is programmed to complete a specific task and given positive or negative signs as it figures out how to finish its duty. The directions and decisions to make along the way are, for the most part, decided by the algorithm.

Normally, reinforcement learning is used in robotics(robots learn to execute tasks in the real world), video game-play(teach bots how to play various video games), or resource management(with a defined objective, help companies plan how to allocate finite resources).

Classification Algorithms

Classification is the process of understanding and organizing ideas and objects into predefined categories. Machine learning programs use various algorithms, utilizing pre-categorized training datasets, to classify subsequent datasets into categories.

In ML, classification algorithms analyze input training data to forecast the probability that following data will fit into one of the predetermined categories. In sum, classification is a type of pattern recognition, with algorithms being applied to training data to encounter the same pattern in future datasets [41].

- **Decision Tree** - Supervised learning algorithm very suitable for classification problems, since it is capable of sorting classes with great precision. It functions similarly to a flowchart, by dividing data points into two similar categories each time, starting from the 'tree trunk', to 'branches', and then 'leaves', with categories becoming increasingly more alike. The end result is categories within categories, which allows for natural classification with slender human oversight.
- **Random Forest** - An *extension* of the decision tree, where the first step is building an abundance of decision trees with training data, and then adjust the new data in one of the trees as a 'random forest'. In short, it averages the data to attach it to the closest tree on the data scale. These types of models are capable of fixing the decision tree problem of occasionally unnecessarily pushing data points into a category.
- **Support Vector Machines** - SVMs use algorithms to train and classify data into degrees of polarity. They assign a hyperplane that best separates each tag/category. Working in two dimensions this would simply be a line, where each category is separated by this line.

The best hyperplane is the one further away from each category, however, in more complex datasets, a unique line may not be sufficient to sort the data into two areas, so other shapes such as circles might be needed. When working with SVMs, the higher the data complexity, the higher the accuracy of the predictor. SVMs are multidimensional, and this is one of the reasons why it allows for more accurate machine learning.

- **Logistic Regression** - Calculation applied to predict a binary aftermath. It can be displayed as Yes/No, Green/Red, etc, where independent variables(categorical or numeric) are put under analysis to determine the binary aftermath with results fitting into one of the two categories. The dependent variable is always categorical: $P(Y = 1|X)$ or $P(Y = 0|X)$, is the probability of Y(dependent variable), given X (independent variable).
- **Naive Bayes** - Measures the possibility of a data point belonging to a particular category or not. To come to a decision of whether or not a particular piece of data should be attributed to a specific category, it is needed to calculate the following:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (2.1)$$

- **K-Nearest Neighbors** - Algorithm of pattern recognition that utilizes training datasets to encounter the k nearest relatives in upcoming examples. When k-nearest neighbor is applied in classification, calculations are done to put data into the category of its closest neighbor. When $k = 1$, it is placed in the category closest to 1. K is defined as a plurality search of its neighbors.
- **Neural Networks** - Algorithm modeled after the structure and function of the human brain. It is made up of layers of interconnected "neurons," which process and transmit information. The network is trained on a dataset, adjusting the strengths of the connections between neurons, known as weights, in order to make accurate predictions or classifications on new data. Neural networks have been successful in a variety of applications, such as image and speech recognition, natural language processing, and drug discovery. They are particularly useful in tasks where the relationships between inputs and outputs are complex and not easily defined by a set of rules.

Clustering Algorithms

Clustering is a ML technique that consists of grouping data points together. A clustering algorithm can be used, given a set of data points, to classify each one into a particular group. Data points in the same group are supposed to share the similar properties, while data points in distinct groups should share very different features. It is an unsupervised learning method, used to gain valuable insight from data, by observing in which group the data points fall into [37].

- **K-Means Clustering** - Probably the most common clustering algorithm, as well as easy to comprehend and apply in code.

First step is selecting a number of categories to use and arbitrarily initializing their center points (vectors with the same length as each data point vector). A good practice is identifying seemingly distinct groups in the data to come up with the number of categories to use.

Every data point is allocated by calculating the distance between that point and every group center, and then allocating the point to be in the group whose center is closest to it. The group center is then recalculated by taking the average of every vector in the group.

This process is then repeated for a number of iterations or until the cluster centers stop having drastic changes between iterations. Randomly initializing the group centers for a few times can also be done, to verify which run produces the best results.

It is quite fast, as it requires few calculations, having a linear complexity of $O(n)$. Some down-points on the other hand are, that the number of categories are manually chosen, and this can be an arduous task at times. Since K-Means starts with randomly chosen cluster centers, different and not repeatable results can be obtained, which is a sign of poor consistency. There are clustering methods more consistent.

- **DBSCAN** - Density-Based Spatial Clustering of Applications with Noise is a clustering algorithm similar to Mean-Shift Clustering, but with considerable advantages, as it does not require a pre-set number of clusters and is capable of identifying outliers as noise.

It starts with a random data point that has not been visited. The neighborhood of the respective point is stipulated by an epsilon distance ϵ (all points within this distance are neighborhood points).

With enough points inside this neighborhood, the clustering process starts and the current data point becomes the first point in the newly obtained cluster. If this does not happen the point will be classified as noise, but can later become part of the cluster, however in both cases it is marked as visited.

The process of making all points in the neighborhood ϵ belong to the same cluster is repeated for every novel point that is added to the cluster group, until every point in the cluster has been visited and labeled.

After being finished with the cluster in question, another point is analyzed, which will lead to another cluster or noise. This procedure is repeated until every point is visited, making each point either part of a cluster or just noise.

Some disadvantages are that it does not perform as well as other algorithms when clusters possess varying densities, and when the data is highly dimensional.

2.6 Graph Mining

Graph mining is an approach where mining techniques are applied to find patterns or relationships in the given real-world collection of graphs. By mining the graph, common substructures and

relationships can be recognized, which aids in clustering the graph sets, finding a relationship between graph sets, or even discriminating or characterizing graphs. It can be used for example in the pre-processing of molecules substructures. Predicting these trends can help build models for the enhancement of applications utilized in real-time [38].

- **gSpan Algorithm** - Commonly used algorithm for discovering frequent subgraphs in a graph database. Its input is a set of labeled connected graphs (comprised of vertices and edges, with a few labels providing information on both vertices and edges) and a threshold named *minsup* (a value between 0 and 100 percent).

The objective of frequent subgraph mining is discovering relevant subgraphs appearing in a set of graphs. A subgraph is considered relevant if it appears various times in a set of graphs. In other words, the goal is to discover subgraphs that are common to multiple graphs. This can be of use in finding associations between chemical elements common to many chemical molecules.

The algorithm will enumerate as output all frequent subgraphs and their support values. A frequent subgraph is a subgraph that shows up in at least *minsup* percents of the graphs from the graph database.

2.7 Evaluation Metrics

A crucial step in every project is assessing the obtained results, in this case that translates to evaluating the used machine learning algorithms. On the one hand, a model may present acceptable results when it is evaluated using a specific metric, but on the other hand those results can turn out to be substandard when other metrics are involved [30].

Machine learning tasks can be divided into Regression or Classification, and the same applies to the performance metrics. A great variety of metrics exist for both, however we will focus only on the most widely used, and analyze the information they declare regarding the performance of a model.

Regression models have constant output, and for that reason metrics that are capable of measuring how far apart predictions and reality are, apply:

- **Mean Squared Error** - Most used metric for regression problems, responsible for finding the average of the squared difference between the target value and the value that was predicted by the regression model. Due to its differentiable nature, it can be optimized. It can lead to an overestimation of how poor the model is, since it penalizes small errors by squaring them. Other key point is that it is a metric more prone to outliers than others, as a consequence of the squaring factor.

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_j - \check{y}_j)^2 \quad (2.2)$$

Where:

- N: datums number
- y_j : target value
- $\check{y}_j$: predicted value

- **Mean Absolute Error** - Average of the difference between the real value and the predicted one. Measures how far the predictions were from the correct output. It is non-differentiable and more resistant against outliers than MSE, because it does not exaggerate errors.

$$MAE = \frac{1}{N} \sum_{j=1}^N |y_j - \check{y}_j| \quad (2.3)$$

Where:

- N: datums number
- y_j : target value
- $\check{y}_j$: predicted value

- **Root Mean Squared Error** - Square root of the average of the squared difference between the ground-truth value and the predicted value. More simply, it is $\sqrt{MSE}$. The differentiable property of MSE is retained, but it does not penalize smaller errors, by square rooting the MSE formula. The interpretation of errors can be performed fluidly and it is less prone to struggle in the presence of outliers.

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (y_j - \check{y}_j)^2} \quad (2.4)$$

Where:

- N: datums number
- y_j : target value
- $\check{y}_j$: predicted value

- **R²** - It consists of a metric calculated based on other metrics, using the sum of squared errors. This coefficient of determination measures how adequate the fit of the regression line is.

$$coeff(R^2) = 1 - \frac{SE(line)}{SE(\bar{Y})} \quad (2.5)$$

Where:

- Y: target
- SE: squared errors
- $SE(\bar{Y})$ (variance of Y) = $\sum_{i=1}^N (y_i - \bar{y})^2$

When it comes to Classification problems, one of the most researched areas in the world, they possess discrete output, which in this case calls for the need of a metric with the capacity to compare separate classes. Classification metrics measure the performance of a model and express if the classification is acceptable or substandard, with each metric doing it in a distinct way:

- **Accuracy** - Metric that is simple to apply, and corresponds to the percentage of correct predictions.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (2.6)$$

- **Confusion Matrix** - Tabular visualization of the model predictions against the real labels (see Table 2.1). The rows in confusion matrices correspond to the instances in predicted classes, while the columns represent the instances in real classes. It is needed to set a null hypothesis as an assumption in order to understand a confusion matrix. Every cell in a confusion matrix represents an evaluation factor:

- **True Positive(TP)**: positive class samples predicted correctly by the model.
- **True Negative(TN)**: negative class samples predicted correctly by the model.
- **False Positive(FP)**: positive class samples predicted incorrectly by the model. Qualifies as a Type-I error in statistical terminology.
- **False Negative(FN)**: negative class samples predicted incorrectly by the model. Qualifies as a Type-II error in statistical terminology.

Table 2.1: Confusion matrix example (rows: predicted values; columns: actual values)

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

- **Precision** - Ratio of true positives and total predicted positives. This metric focuses on Type-I errors. A precision value close to 1 means that a given model kept track of every true positive, and is also capable of classifying well between correct and incorrect premises. The existence of a Type-II error can't be measured by this metric. Scores below 0.5 means that the number of false positives is high, and that can be attributed to imbalanced classes or unadjusted hyperparameters. This metric alone will not reveal a model's performance on various grounds.

$$precision = \frac{TP}{TP + FP} \quad (2.7)$$

Where:

- $0 < P < 1$

- **Recall** - Ratio of true positives for every ground-truth positive. This metric focuses on Type-II errors. A recall value close to 1 means that a given model kept track of every true

positive, and is also capable of classifying well between correct and incorrect premises. The existence of a Type-I error can't be measured by this metric. Scores below 0.5 means that the number of false negatives is high, and that can be attributed to imbalanced classes or unadjusted hyperparameters. Regarding this and the previous metric, it is of importance to highlight that both are only used in particular scenarios, since each one can only identify the existence of a specific type of error.

$$recall = \frac{TP}{TP + FN} \quad (2.8)$$

Where:

- $0 < R < 1$
- **Precision-Recall trade-off** - For model enhancement, one can either improve precision or recall, but never both. This trade-off is of great impact in real-world scenarios, so the exclusive use of precision or recall is not an advised approach. Corporate reports and online competitions compels that both precision and recall be used in combination as a submission metric.
- **F1-score** - It is a merger between precision and recall, and more specifically the harmonic mean of both. An elevated F1-score represents a high precision and recall, as well as a good balance between them, which translates into great results on imbalanced classification problems. On the other hand, a low F1-score only evaluates performance at a threshold, and is not very informative. It is not clear if the problem is low precision or low recall, when a low F1-score is obtained, and also whether the model contains Type-I or Type-II errors. F1 is one of the most popular metrics in judging performance of models.

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \quad (2.9)$$

- **AUC-ROC Curves** - Area under receiver operating characteristics curves makes use of true positive rates(TPR) and false positive rates(FPR). TPR represents the proportion of positive values that are assertively considered as positive, when compared to all positive values. A high TPR means fewer positive values missed. FPR represents the proportion of negative values that are incorrectly considered as positive, when compared to all negative values. A high FPR mean more negative values misclassified.

$$TPR = \frac{TP}{TP + FN} \quad \text{and} \quad FPR = \frac{FP}{FP + TN} \quad (2.10)$$

In order to combine TPR and FPR into a unique metric, they need to be computed with various contrasting thresholds for the logistic regression, and plotted on a graph. The emerging curve is designated the ROC curve, and the AUC-ROC metric corresponds to the area below this curve.

This area is equivalent to the probability of a arbitrarily chosen positive example ranking above (more likely of being positive than negative) a randomly chosen negative example. A high AUC-ROC signifies that the used algorithm ranks testing data in a good manner, with most positive cases at one end and negative ones at the other.

It is important to underline that AUC-ROC curves are applied in academic researches, as well as when comparing distinct classifiers, but are not a favorable choice when a considerable class imbalance is present.

Chapter 3

A platform for ADE studies - Tamingo

As mentioned before, adverse drug events (ADEs) are a significant public health concern, and studying ADEs requires the collection and analysis of large amounts of data. To facilitate this process, a web-based platform called Tamingo for ADE studies was developed. Tamingo aims to provide an easy-to-use interface for both researchers and participants, and can help streamline the data collection and analysis process.

3.1 Technologies

In this chapter, we will discuss the technologies used to build Tamingo, specifically Python, Django, and SQLite. We will also explain why these technologies are well-suited to building a web-based ADE study platform, and why one should choose Python and Django over other alternatives.

Python

Python is a popular, high-level programming language that is well-suited for scientific computing and data analysis. Python is an interpreted language, which makes it easy to write and test code, and it has a large and active community that has developed a wide variety of libraries and frameworks for scientific computing and data analysis. The libraries numpy, scipy, pandas, matplotlib, seaborn, sklearn, and many more, are widely used for data science related projects, including ones dealing with ADEs. Python has a simple and readable syntax which makes it more appealing, and this is particularly useful for a project such as this one, where the goal is to make data collection and analysis as user-friendly as possible [27].

Django

Django is a high-level web framework written in Python. It is designed to make it easy to build web applications, and it follows the model-view-controller (MVC) architectural pattern. One of the most significant advantages of using Django is that it is designed to be highly secure, and it takes

care of many security issues automatically. This is essential in building a web-based ADE study platform, as data security is a critical concern when dealing with sensitive health information. Additionally, Django offers a built-in admin interface, which enables easy management of the data by the researchers. This can save significant time and resources in the long run [28].

SQLite

SQLite is a lightweight, file-based relational database management system. It is often used as the database for small-to-medium sized web applications. Because it is file-based, it is simple to set up and manage, and it does not require a separate server. This makes it an ideal choice for smaller projects, or projects that are not expected to handle a large amount of traffic. Additionally, SQLite is known for its robustness and reliability, it can handle large datasets and concurrent requests well [4].

Together, these technologies helped create Tamingo, an easy-to-use, secure, and reliable platform for data collection and analysis in ADE studies.

3.2 Layout and Design

The Tamingo platform for ADE studies is designed to be user-friendly and intuitive for researchers, healthcare professionals and others. As seen in the previous chapter, the platform is built using Django, a powerful web framework that allows for the creation of complex and customizable web applications.

The homepage of the Tamingo platform provides a brief slogan of what the platform aims to achieve through the collection and analysis of ADEs: "Empowering Drug Development with ADE Knowledge: Shaping the Future of Medicine through Research". The homepage features a clean and modern design, with clear and concise information presented in an easy-to-read format.

One of the key features of the Tamingo platform is the sidebar, which is always present and easily accessible to the user. The sidebar allows the user to easily switch between tasks and navigate the platform. As detailed in the subsequent chapter, the platform's sidebar also enables users to access various functionalities such as data collection, data visualization, and data analysis, through the following options: "Home", "OpenFDA", "Filter", "SMILES->SDF", "Descriptors", "Predict", and "Admin".

The layout and design of the Tamingo platform has been carefully considered to ensure that users have a seamless and efficient experience. The platform has been optimized for use on desktop.

Overall, Tamingo provides a powerful and user-friendly solution for conducting ADE studies, making it easier for anyone to collect and analyze data, in order to make substantiated claims about drugs characteristics.

3.3 Capabilities

This powerful and versatile tool offers a wide range of capabilities for users from different backgrounds who are interested in ADE studies. This chapter will delve into the platform's features, including data cleaning, data transformation, data collection, data visualization, and data analysis. Additionally, we will explore the platform's user-friendly interface and intuitive navigation, which make it easy for users to access and utilize the platform's full range of capabilities. All in all, the Tamingo platform is a comprehensive solution that offers a complete set of tools for conducting ADE studies, and this chapter will provide a detailed examination of the platform's capabilities and functionality.

The use case diagram presented below illustrates the functionalities offered by Tamingo and the interactions between the user and the system. The diagram shows the main use cases of the platform and the relationships between the actors and the system:

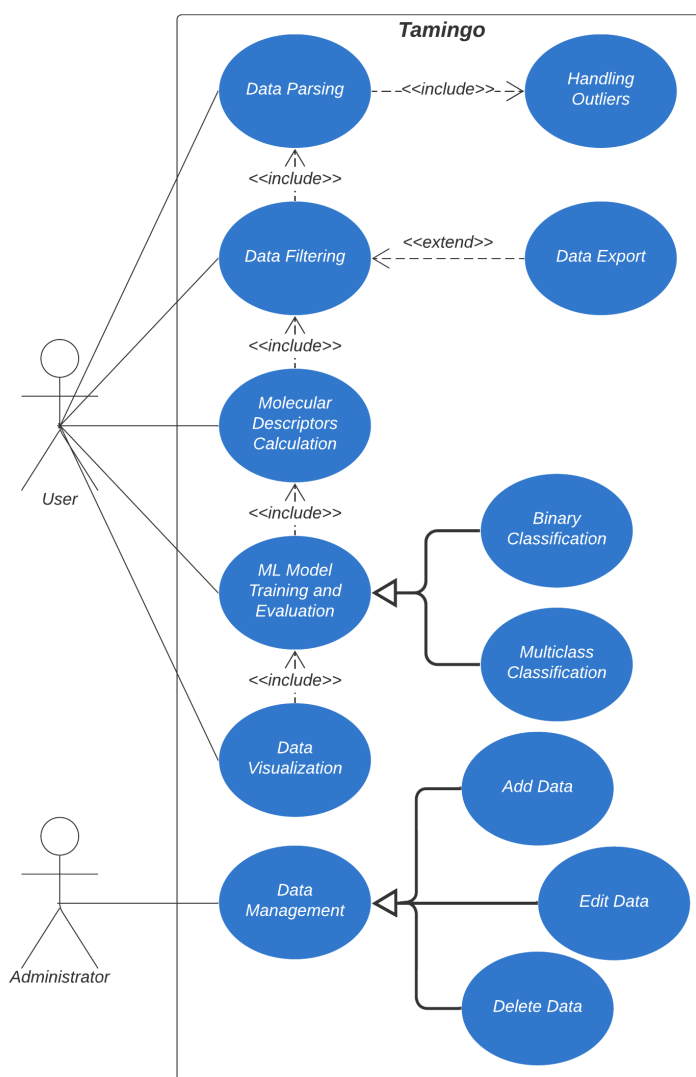


Figure 3.1: Tamingo Use Case Diagram

3.4 User Controlled Data Manipulation

OpenFDA ADE Data Collection

In order to better understand and mitigate the risks associated with ADEs, it is important to have access to accurate and detailed data on these events. The OpenFDA API provides a rich source of information on ADEs, that has been applied in a series of known researches,[33] including data on patient demographics, the seriousness of the event, and the drugs involved.

To acquire ADE data from OpenFDA, the data was directly downloaded in the .zip compressed format from the OpenFDA website and is stored in the JSON format. It is important to note that the data obtained from OpenFDA is based on voluntarily reported adverse events and may not represent the entire population of adverse events. Therefore, the data should be used with caution and should not be used to make causal inferences.

Once the data was appropriately cleaned, it was important to extract the most relevant elements of the data. Some key elements of interest include the patient's gender, weight, the seriousness of the event (represented in a binary manner, where an event is considered to be serious when the ADE resulted in death, a life threatening condition, hospitalization, disability, congenital anomaly, or other serious condition and not serious otherwise), the patient's reaction as a MedDRA term, the reaction outcome at the time of last observation (represented by one of six different values: Recovered/resolved; Recovering/resolving; Not recovered/not resolved; Recovered/resolved with sequelae; Fatal; Unknown), and the active ingredient of each drug involved.

To extract these elements, data parsing and data cleaning techniques such as regular expressions, string manipulation, and filtering were applied. The Python json library was used to perform a lot of these processes, since it is capable of decoding JSON strings into Python objects: The `json.load()` function was used to read JSON-formatted data from a file into various Python objects, as exemplified in the created parser in Appendix A.

Once the relevant elements are extracted by the platform, the data can be analyzed to identify patterns and trends in ADEs. For example, the data can be grouped by drug or by patient characteristics to arrive to conclusions on if these influence the risk for ADEs.

Overall, by extracting the most relevant elements of the data, it is possible to gain insights into patterns and trends in ADEs, and to identify areas where further research is needed.

Data Filtering

Data filtering is a process that enables users to access and analyze specific subsets of data based on their preferences. It plays a crucial role in the Tamingo Django-based platform, as it allows users to easily access and analyze specific subsets of the data according to their preferences. The platform is designed to take into account a variety of user-specified criteria, such as the existence of particular reactions, drugs, or functional groups, and filter the data accordingly [16].

The implementation of data filtering in the Tamingo platform begins with the creation of filters for each column in the data. These filters can be applied by the user according to their specific

needs and preferences. For example, a user may choose to filter the data based on the presence of a specific functional group in a drug, or based on the outcome of a specific reaction. The user may also be able to filter by other criteria like pick only cases for a specific gender or only drugs belonging to a specific category.

To facilitate the filtering process, the platform utilizes the Django-filter library, which allows for the creation of dynamic filters based on the data model. The Django-filter library utilizes the built-in queryset filtering capabilities of Django to generate a filtered queryset based on the user-specified criteria. The queryset is then converted into a pandas dataframe, which can be further modified or exported as a .csv file for further analysis or experimentation.

Data filtering plays an essential role in data analysis, as it helps users to focus on specific subsets of data that are most relevant to their research or experimentation. In the Tamingo platform, the data filtering functionality is flexible and user-friendly, allowing for easy access to specific subsets of the data based on their preferences [17]. Additionally, the ability to export the filtered data as a .csv file provides users with the opportunity to perform additional analysis or experimentation inside or outside of the platform.

The following list aims to facilitate comprehension of the filtering process described in this section:

- **The user can filter the data in 8 different ways:** by seriousness, sex, weight, the type of reaction, drugs, functional groups, categories, or outcome.
- **The user can apply one or multiple filters at a time**
- **The selected filters are applied to the data in the table**
- **The filtered data is automatically exported to a CSV file**

Machine Learning

The Tamingo platform allows for the application of machine learning algorithms to predict the seriousness of a reaction or the outcome of a reaction. The use of machine learning in Tamingo enables the platform to analyze large amounts of data and make predictions based on patterns and trends in the data.

For the prediction of the seriousness of a reaction, various binary classification models can be used, such as the Multinomial Naive Bayes, Logistic Regression, k-nearest Neighbors, Linear Support Vector, Decision Tree, Bagging, AdaBoost, Random Forest, Soft Voting/Majority Rule, and Multi-layer Perceptron classifiers. These models are all well-established algorithms for binary classification and have been proven to be effective for this task. The models are trained using a dataset of previous reactions involving two drugs, with the outcome being labeled as either serious or not serious. The parameters used to create these models include patient characteristics such as sex and weight, as well as the functional groups and molecular descriptors of the drugs involved in the reaction.

To predict the outcome of a reaction, multiclass classification models are available, such as the k-nearest Neighbors, Decision Tree, Bernoulli Naive Bayes, Random Forest, Ridge Regression, Label Propagation, and One-vs-the-Rest classifiers. These models are widely used algorithms for multi-class classification and have been verified to be successful in accomplishing this endeavor. The models can be trained using a dataset of previous reactions, with the outcome being labeled into 5 different possibilities: recovered/resolved, recovering/resolving, not recovered/not resolved, recovered/resolved with sequelae (consequent health issues), or fatal. Similarly to the previous prediction, these models also take into account patient characteristics such as gender and body weight, as well as the functional groups and molecular descriptors of the drugs that are involved in the reaction.

The chosen library to achieve this in Tamingo was scikit-learn, a powerful machine learning library for python. Scikit-learn provides a wide range of algorithms for classification, regression, clustering, and dimensionality reduction, which makes it a suitable choice for implementing machine learning in the Tamingo platform [3].

In order to present the results of the machine learning models to the user, Tamingo utilizes a table format. The table includes the name of the model used and the accuracy of each one. The user can easily compare the performance of different models and select the one that best fits their needs.

In summary, Tamingo is a powerful platform that uses machine learning algorithms to predict the seriousness of a reaction, as well as the outcome of a reaction. The platform utilizes a variety of parameters such as patient characteristics, functional groups, and molecular descriptors to train these models. The use of well-established algorithms from the scikit-learn library enables the platform to effectively analyze large amounts of data and make accurate predictions.

Data Management

The web app includes a built-in data management feature that allows users to manually edit, remove, or add new data to all the database models. The user can access this feature through the Tamingo's sidebar and make changes to the data as needed. The data management feature is designed to be user-friendly and easy to use, allowing users to quickly and easily make changes to the data without the need for extensive technical knowledge. This feature is particularly useful for maintaining the accuracy and completeness of the data stored in the platform's databases, and can be used to keep the data up-to-date and relevant to the user's needs. Additionally, this feature is also a way for the user to customize and personalize the data according to their personal preference.

3.5 Automated Data Operations

SMILES Conversion

The conversion of active ingredients from pharmaceutical drugs into their respective SMILES codes is an important step in the process of drug discovery and development. SMILES codes, or Simplified Molecular Input Line Entry System codes, are a standardized format for representing the chemical structure of molecules. These codes are used in various applications, including computer-aided drug design, or virtual screening of chemical libraries [14]. In this section, we will discuss the process of converting active ingredients from pharmaceutical drugs obtained from OpenFDA into their respective SMILES codes, using Python API calls to PubChem. Additionally, we will discuss how every time a SMILES is retrieved, it is saved in a Django model in order to save time in the future by not having to do an API call since we already possess the SMILES code.

A few example of what SMILES codes look like compared to some known compounds and their molecular formulas can be seen in Table 3.1.

Table 3.1: SMILES Syntax Representation

Compound	SMILES
Sodium Chloride (NaCl)	[Na+].[Cl-]
Carbonic Acid (CH ₂ O ₃)	C(=O)(O)O
Morphine (C ₁₇ H ₁₉ NO ₃)	CN1CCC23C4C1CC5=C2C(=C(C=C5)O)OC3C(C=C4)O
Cellulose (C ₁₂ H ₂₂ O ₁₁)	C(C1C(C(C(C(O1)OC2C(OC(C(C2O)O)O)CO)O)O)O)O

In order to convert active ingredients from pharmaceutical drugs into their respective SMILES codes, we made API calls to PubChem, which is a public, online database of chemical information, as soon as each active ingredient was retrieved from OpenFDA. The API calls were made using Python, and the returned information included the canonical SMILES code for each active ingredient.

PubChem provides a number of different APIs, each with its own set of capabilities and parameters. The chosen API and the most commonly used API for searching for chemical compounds is the PUG REST (PubChem Useful Gadgets RESTful) API. This API allows for a variety of different search types, including searching by compound name, chemical identifier, and molecular structure.

Once the SMILES codes were obtained, they were saved in a Django model. The model was designed to store the active ingredient name, its PubChem ID, and the SMILES code. This way, the next time the same active ingredient emerges, the SMILES code can be retrieved from the model instead of making an API call to PubChem, saving time and resources.

The process of converting active ingredients from pharmaceutical drugs into their respective SMILES codes using Python API calls to PubChem was successful. Thousands of active ingredients were converted and their respective SMILES codes saved.

By saving the SMILES codes in a Django model, it was possible to significantly reduce the time required to retrieve the SMILES codes in the future. For example, if we needed to retrieve the SMILES code for an active ingredient that had already been converted, it could be retrieved in less than a second, as opposed to the 0.5 seconds, which is the average time taken for an API call.

Overall, this process is a valuable tool for drug discovery and development, as it allows for the efficient and accurate representation of chemical structures.

Drug Categorization

Similarly to the previous section, we will once again explore the use of Python to make API calls to PubChem in order to retrieve the category of each drug based on its active ingredient, and use the PUG VIEW (PubChem Useful Gadgets View) API instead of the PUG REST API.

The PUG VIEW API provides a user-friendly interface for searching and retrieving data from the PubChem database. The first step in using the PUG VIEW API to retrieve the category of a drug based on its active ingredient is to search for the compound using its name or identifier. This can be done using the "compound/name" or "compound/cid" endpoint, respectively. For example, to search for the compound "aspirin," the following API call could be used:

https://pubchem.ncbi.nlm.nih.gov/pug_view/data/compound/name/aspirin/cids/JSON

This call will return a JSON object containing the PubChem Compound Identifier (CID) for aspirin, which can then be used to retrieve additional information about the compound.

Once the CID has been obtained, the next step is to retrieve the category of the drug using the "compound/cid" endpoint with the Mesh Pharmacological Classification header to facilitate the JSON parsing, like so:

https://pubchem.ncbi.nlm.nih.gov/pug_view/data/compound/cid/JSON?heading=Mesh+Pharmacological+Classification

By using the PUG VIEW API and Python to make API calls to PubChem, we are able to retrieve the category of a drug based on its active ingredient. This information can then be saved in a Django model for future use, so that the same API calls do not have to be made repeatedly.

Some examples of drug categories are showcased in Table 3.2.

Table 3.2: Drug Categories Examples

Drug	Category
Atenolol	Anti-Arrhythmia Agents
Duloxetine	Analgesics
Clozaril	Antipsychotic Agents
Taxotere	Tubulin Modulators

Furthermore, the method used in this dissertation can be applied to other endpoints of the PubChem API to retrieve other information about the compound, such as its chemical structure or toxicity data. Additionally, it can be used to retrieve information about multiple compounds at once, by using the batch search functionality provided by the API.

Functional Groups Identification

Tamingo utilizes a combination of SMARTS (Simplified Molecular Input Line Entry System Arbitrary Target Specification) codes and the python RDKit library to identify the functional groups present in drugs. SMARTS is a language used to describe molecular substructures, which allows for the identification of specific functional groups within a molecule.

Functional groups are specific atoms or groups of atoms within a molecule that are responsible for its chemical properties and reactivity [42]. Some examples of common functional groups are amines (NH₂), carboxylic acids (COOH), alcohols (OH) and ethers (R–O–R).

The process begins by collecting a file of the most relevant SMARTS codes from Daylight, which is a company that provides software tools for chemistry and biology research. These SMARTS codes represent the functional groups that are of interest for ADE studies. Using the python RDKit library, a molecule object is created from the SMILES code of a given drug using the Chem.MolFromSmiles() function. This molecule object is then compared to query molecule objects created from the stored SMARTS codes using the Chem.MolFromSmarts() function. The GetSubstructMatches() function is used to identify any instances of the functional groups represented by the SMARTS codes in the molecule. This process can be repeated for each functional group in the database, providing a comprehensive analysis of the functional groups present in the given drug.

The functional groups and their respective SMARTS codes chosen to be included in Tamingo, as the most common and relevant in ADE studies are available in Appendix B.

Such information can aid in understanding the potential adverse effects of a drug and aid in the development of safer and more effective pharmaceuticals by providing insights into the chemical structure of the drug and the role of functional groups in its properties and reactivity.

Molecular Descriptors Compilation

Obtaining molecular descriptors from SMILES is an important step in the analysis of chemical compounds. Molecular descriptors are numerical values or arrays that describe the structural and physicochemical properties of a molecule. These descriptors can be used to predict the biological activity, toxicity, or other properties of a compound, and can also be used as input for machine learning models [15]. Some examples of common molecular descriptors include:

- **Molecular weight** - The sum of the atomic weights of all atoms in a molecule
- **LogP** - A measure of the hydrophobicity and lipophilicity of a compound
- **TPSA** (Topological polar surface area) - A measure of the polar surface area of a molecule
- **HBA** (Hydrogen bond acceptors) - The number of atoms in a molecule that can act as hydrogen bond acceptors
- **HBD** (Hydrogen bond donors) - The number of atoms in a molecule that can act as hydrogen bond donors

One way to obtain molecular descriptors from SMILES is through the use of Mordred, a molecular descriptor calculator. Mordred is a python library that can calculate more than 2000 molecular descriptors, including constitutional, topological, geometrical, and electronic descriptors. The library is compatible with the Open Babel and RDKit libraries, which are commonly used for SMILES manipulation, and in the case of this dissertation fits perfectly with the use of the latter library.

Mordred can be used to calculate molecular descriptors for a single compound or for a large dataset of compounds. The library also includes pre-processing steps such as cleaning and standardizing SMILES, which is important for accurate descriptor calculation.

It is recommended to perform these calculations after data filtering, where the user is sure about what characteristics should be present in their data. Data filtering allows the user to focus on a specific subset of compounds and increases the relevance of the calculated descriptors. Additionally, data filtering can also help to identify and remove duplicate or invalid compounds, which can negatively impact descriptor calculation.

In conclusion, obtaining molecular descriptors from SMILES is an important step in the analysis of chemical compounds. Mordred is a powerful tool that can be used to calculate a wide range of molecular descriptors, such as molecular weight, logP, TPSA, HBA, and HBD, and it is recommended to perform these calculations after data filtering to increase the relevance of the calculated descriptors. The calculated descriptors can then be used in various scenarios, as detailed in the following chapter, and also other applications, such as drug discovery, toxicity prediction, and more.

SMILES to SDF Conversion

Chemical compounds can be represented in various formats, including SMILES and Standard Data File (SDF) format. SDF is a widely used format in the field of chem-informatics and drug discovery, as it can store not only the chemical structure of a compound, but also additional information such as properties and experimental data. The conversion of SMILES to SDF format is a common task in chem-informatics, as it allows for the integration of various data sources, and the ability to manipulate and analyze the data using various software programs. A snippet of a file compiled by Tamingo in the SDF format is available at Appendix C.

There are a variety of ways to convert SMILES to SDF format, including using online tools and software programs. One common method is to use the Open Babel toolkit, an open-source chemical toolbox that supports conversions between various chemical file formats, including SMILES and SDF. Open Babel uses a force-field-based approach to generate the 3D coordinates for the atoms in the molecule, and it can also assign partial atomic charges using a variety of methods such as Gasteiger or MMFF94. It also allows for the generation of 2D diagrams, useful for visualizing the chemical structure.

Another popular method is to use the RDKit, a collection of chem-informatics and machine learning tools that can be used to convert SMILES to SDF format. The RDKit uses a graph-based

representation of the chemical structure, and it can also generate 2D and 3D coordinates, generate conformers, and perform molecular docking.

Some web services such as PubChem, ChemSpider, and ChEMBL provide web-based conversion services, allowing users to upload SMILES files and receive SDF files in return. These web services also provide additional functionalities such as compound searching, property prediction, and bio-activity data retrieval.

Another approach is to use software packages such as Python and its libraries such as Pybel, ChemPy or PandasTools. These libraries provide easy-to-use functions for converting SMILES strings to SDF format. Pybel is built on top of the Open Babel toolkit, allowing the user to access the functionalities of Open Babel, and it also provides a pythonic interface for the manipulation of chemical structures. ChemPy is a Python library for chem-informatics and computational chemistry, and it can be used to generate 2D and 3D coordinates, and perform molecular dynamics simulations. PandasTools is a python library that extends the functionality of the popular data manipulation library pandas, to work with chemical data. It can be used to convert SMILES to SDF format by utilizing the functionality of the open-source toolkit RDKit.

When converting SMILES to SDF format, there were a few considerations taken into account. One important consideration is the preservation of structural information and additional data. The chosen method should be able to correctly convert the SMILES representation to the SDF format, while preserving the structural information and additional data, such as properties and experimental data. Another consideration is the handling of stereo-chemistry. Stereo-chemistry is the three-dimensional arrangement of atoms in a molecule, and it plays an important role in the biological activity of a compound. The chosen method should be able to correctly handle the stereo-chemistry information present in the SMILES representation, and correctly convey it in the SDF format.

Taken all of this into account, it was decided to implement this feature into Tamingo with the aid of PandasTools and its connection with RDKit.

Each substance name and respective SMILES code was included in a CSV file instantaneously read into a Pandas dataframe, which was modified using the AddMoleculeColumnToFrame function available in PandasTools, that is responsible for providing a set of tools for working with molecular data. The function allows for the addition of a new column to a dataframe that contains molecular information, such as SMILES strings or molecular descriptors.

The function takes two arguments:

- **frame** - This is the dataframe to which the new column will be added.
- **smiles_col** - This is the name of the column in the dataframe that contains the SMILES strings of the molecules.

It returns a new dataframe with the added column which contains the molecular information, which is immediately modified by applying the WriteSDF function, which allows the writing of the contents of a Pandas dataframe to a file in the SDF format.

The function takes two arguments:

- **frame** - This is the dataframe that contains the molecular information to be written to the SDF file.
- **file** - This is the path and filename of the SDF file to be created.

The exact code created to achieve this is available in [Appendix D](#)

Chapter 4

Case Study - DM applied to ADE

ADEs are a major public health concern, as they can lead to serious complications and even death. One of the main challenges in understanding and preventing ADEs is the vast amount of data that needs to be analyzed in order to identify patterns and trends that can help to predict and prevent these events. Data mining is a powerful tool that can be used to analyze large amounts of data and extract useful information that can help to improve the safety and efficacy of drugs.

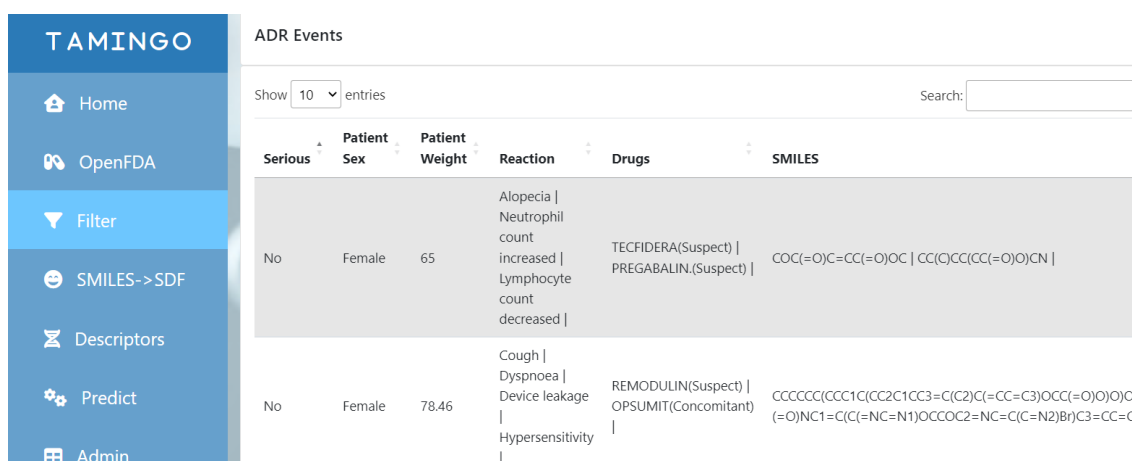
The field of data mining has grown significantly in recent years, and there are now a wide range of techniques and algorithms that can be used to analyze data. These can be utilized to identify patterns and trends in data that can help to predict and prevent ADEs.

In this chapter, we will discuss the importance of data preprocessing and data filtering, which are key steps in the data mining process that ensure that the data is clean and ready for analysis, showcasing how a user can obtain a personalized dataset. In addition, the various machine learning techniques that can be applied to the ADE data will be mentioned, and how these techniques can be used to improve our understanding of ADEs and help to prevent these events. Further on, the challenges and limitations of data mining in this context, and how these can be overcome, will also be pointed out.

The goal of this chapter is to provide an overview of the data mining techniques that can be applied to ADE data and the potential benefits of these techniques for improving the safety and efficacy of drugs.

4.1 Phase 1 - Managing OpenFDA Data

As briefly discussed in Chapter 3, the Tamingo platform takes advantage of the wealth of data on ADEs available through OpenFDA. The data provided by OpenFDA includes a wide range of information on ADEs, including patient characteristics, drug information, and the outcomes of the reactions. However, not all of this information is relevant to the goals of the Tamingo platform, which is capable of predicting the seriousness of ADEs and the outcomes of reactions, as we can observe in Appendix E.



The screenshot shows the TAMINGO ADR Events interface. On the left is a sidebar with navigation options: Home, OpenFDA, Filter (highlighted), SMILES->SDF, Descriptors, Predict, and Admin. The main area displays a table of ADR events. At the top, there's a search bar and a dropdown to show 10 entries. The table has columns for Serious, Patient Sex, Patient Weight, Reaction, Drugs, and SMILES. Two entries are visible:

Serious	Patient Sex	Patient Weight	Reaction	Drugs	SMILES
No	Female	65	Alopecia Neutrophil count increased Lymphocyte count decreased	TECFIDERA(Suspect) PREGABALIN.(Suspect)	<chem>COC(=O)C=CC(=O)OC CC(C)CC(CC(=O)O)CN </chem>
No	Female	78.46	Cough Dyspnoea Device leakage Hypersensitivity	REMODULIN(Suspect) OPSUMIT(Concomitant)	<chem>CCCCC(CCC1C(CC2C1CC3=C(C2)C(=CC(=C3)OCC(=O)O)O)O(=O)NC1=C(C(C(=NC=N1)OCCOC2=NC=C(C=N2)Br)C3=CC=C</chem>

Figure 4.1: Access Pre-Loaded OpenFDA Data

The platform already contains all of the OpenFDA ADE data corresponding to the years between 2019 and 2021, which the user can access directly by choosing the "Filter" option and observing all of the available data (see Figure 4.1).

An alternative way of loading OpenFDA data is by downloading it at the Drug Adverse Events section at the [OpenFDA website](#), and then uploading it through the "OpenFDA" option, where multiple data files can be added simultaneously (see Figure 4.2)

The data provided by OpenFDA is available in .json files, which are a widely used data format for storing and exchanging data. In order to parse these files, the Tamingo platform makes use of the JSON module in python. This module provides a range of functions for working with JSON data, including functions for loading JSON data from files and strings, and functions for accessing and manipulating the data.

To ensure that the data used in the Tamingo platform is relevant and useful, a process of preprocessing the data is undertaken. This process included parsing the json files provided by OpenFDA, and ignoring various undesirable fields that are not relevant to the goals of the project. Examples of some undesirable fields that can be ignored include information on the reporter of the ADE, and the date referring to when the ADE was reported.

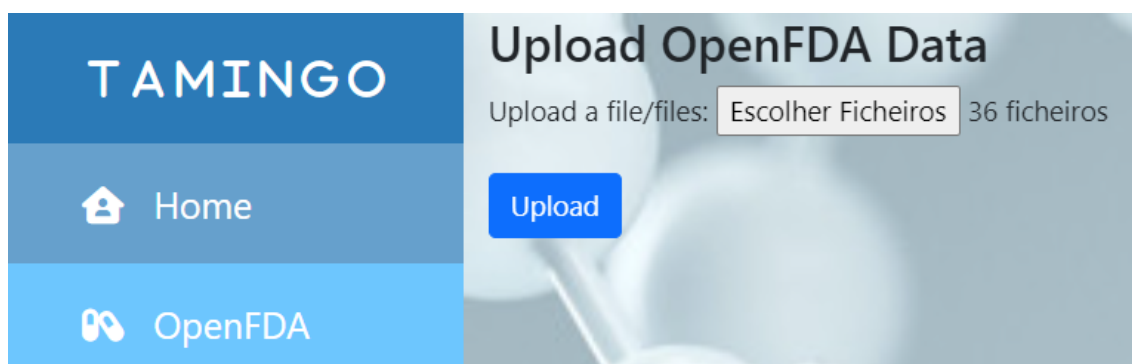
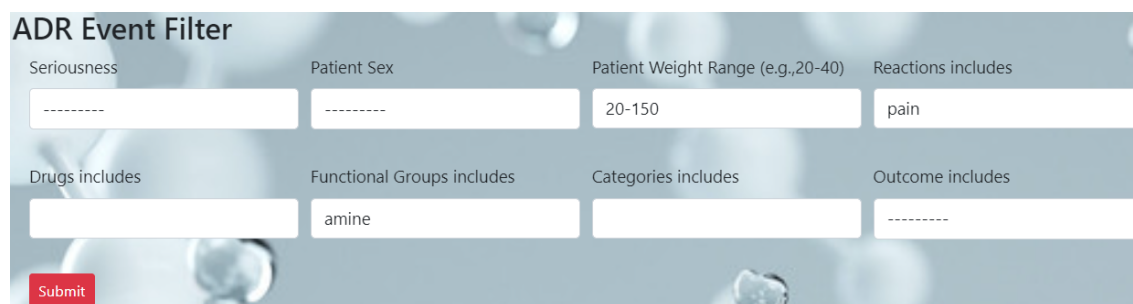


Figure 4.2: Upload OpenFDA Data



The screenshot shows a web interface titled "ADR Event Filter". It contains eight input fields arranged in two rows of four. The first row includes "Seriousness" (empty), "Patient Sex" (empty), "Patient Weight Range (e.g.,20-40)" (filled with "20-150"), and "Reactions includes" (filled with "pain"). The second row includes "Drugs includes" (empty), "Functional Groups includes" (filled with "amine"), "Categories includes" (empty), and "Outcome includes" (empty). A red "Submit" button is located at the bottom left of the form area.

Figure 4.3: Filtering System

The selection of relevant fields was based on a thorough analysis of the available data, and the goals of the Tamingo platform. The fields that were ultimately selected for use in the platform include: **seriousness**, **patient sex**, **patient weight**, **reaction description**, **reaction outcome**, as well as the **drug pair** involved in the reaction. These fields were chosen as they are considered to be the most relevant for predicting the seriousness of ADEs and the outcomes of reactions, and to reiterate, an overview of the parser can be observed in Appendix A. This allows other researchers and developers to replicate the process of preprocessing the data and get to know how to potentially access these fields in their own work.

4.2 Phase 2 - Personalized Dataset Creation

Preprocessing ADE data is an essential step in the data mining process, as it ensures that the data is clean and ready for analysis. That is when the filtering process takes place, and the user can choose from the following filter options how to create a personalized dataset (see Figure 4.3)

As displayed, Tamingo offers 8 distinct filtering options, so that personalization of a desired dataset can be maximized. For this particular case study, it was prioritized the gathering of fairly large amounts of data, predicting that this practice would lead to obtaining better results by correlation.

Three distinct filtering options were applied, starting by looking for ADEs where the patient's weight was between the range of 20-150kg. Furthermore, the fairly common reaction of "pain" had to be part of the description of the patient's response, in addition to the Amine functional group belonging to at least one of the drug pair intervenients.

4.3 Phase 3 - Data Analysis

The Tamingo platform offers a powerful tool for analyzing ADEs and predicting the seriousness of reactions. In order to demonstrate the capabilities of the platform, this case study was decided to be conducted using the pre-loaded data from OpenFDA, which covers the years from 2019 to 2021, with the showcased filters in the previous section being applied. This data is available

on the platform for users to experiment with and gain a deeper understanding of the platform's capabilities.

The main goal of this case study was to build a binary classification prediction model capable of predicting the seriousness of a reaction. ADEs are considered serious if they result in death, a life-threatening condition, hospitalization, disability, congenital anomaly, or other serious condition. To achieve this, the model utilized a range of attributes including patient's sex and weight, as well as thousands of molecular descriptors that correspond to the two interacting drugs. The training data for this study had an equal number of positive and negative examples, since Tamingo is capable of automatically handling this 50-50 split, and the scikit-learn library was chosen to perform the classification. Accuracy, which is known for being a harsh metric, was the chosen criteria to evaluate the models.

In total, the data included 3464 different cases, with 1680 attributes, mostly representing the hundreds of molecular descriptors of each drug.

Data was randomly split into training, testing and validation data, with 70% of the data being used for training, and 15% for both testing and validation. This procedure is essential to ensure that the possible over-fitting that could occur in the training dataset does not translate to misleading accuracy scores. For the sake of being as harsh as the metric itself the accuracy score of the training data was not taken into account in the results that are presented in the following tables of this chapter. Instead, the average of the accuracy percentages obtained in both the testing and validation data was considered as the correct score of each model.

Step 1

The first step consisted on choosing which algorithms would be best suited to solve this type of problem and get an idea of how good was our data as a starting point. After careful analysis, 10 different algorithms known to be suitable for this types of binary classification problems were applied in order to construct these models, including:

- **Naive-Bayes** - Probabilistic algorithm that utilizes Bayes' theorem to make predictions. This algorithm is based on the assumption that the features in the data are independent of each other, which makes it computationally efficient. The algorithm was used in this project for its ability to handle large amounts of data and make predictions quickly.
- **Logistic Regression** - statistical method that is used to model a binary dependent variable. This algorithm is useful for predicting a binary outcome, such as serious or not serious, based on one or more independent variables.
- **K-NN** - non-parametric method that can be used in classification problems. This algorithm is based on the idea that similar observations are likely to have similar outcomes.
- **SVM** - supervised learning algorithm that can also be utilized for classification. This algorithm is particularly useful for high-dimensional data and non-linear decision boundaries.

- **Decision Tree** - supervised learning algorithm that can be applied in classification. This algorithm is based on the idea of breaking down a dataset into smaller subsets based on certain conditions.
- **Bagging Tree** - ensemble algorithm that is used to improve the performance of a single classifier by combining the predictions of multiple classifiers. It works by training multiple instances of the base classifier on different subsets of the data and then averaging the predictions. In this case study, it was used to try and improve the Decision Tree algorithm performance.
- **AdaBoosted Tree** - another ensemble algorithm that is used to improve the performance of a single classifier (again we applied to the Decision Tree) by combining the predictions of multiple classifiers. It works by training multiple instances of the base classifier on different subsets of the data and then weighting the predictions based on the performance of the previous classifiers.
- **Random Forest** - ensemble algorithm that is used non-exclusively for classification tasks. It works by training multiple decision trees on different subsets of the data and then averaging the predictions.
- **Neural Network** - algorithm that is inspired by the structure and function of the human brain. It consists of layers of interconnected nodes, called artificial neurons, that process and transmit information.
- **Voting Classifier** - ensemble method that combines the predictions of multiple base classifiers in order to improve the overall performance of the model. We used the models created thanks to the previous 9 methods as a base for this algorithm.

All of the algorithms were run on the personalized dataset, but without the functional groups and categories being included at this stage, giving us an idea of how the models would perform right away. It is of importance to remind that since the data contains equal amounts of positive and negative examples, a model can only be considered "useful" if its accuracy is at least superior to 50%, since even a broken clock is correct half of the time.

The obtained results were very unbalanced, and this could only be related with the lack of data scaling. To solve this issue, some preprocessing techniques were applied, such as:

- **Standard Scaler** - Technique that standardizes a feature by subtracting the mean and then scaling to unit variance. This is done by calculating the mean and standard deviation of the data, and then subtracting the mean from each data point and dividing by the standard deviation. This results in a dataset with a mean of 0 and a standard deviation of 1, and is useful for normalizing the data for use in machine learning algorithms.
- **Min-Max Scaler** - Method that scales the data to a specific range, in this case between 0 and 1. This is done by subtracting the minimum value from each data point and then dividing by

the range (maximum value minus minimum value). This results in a dataset where all the values are between 0 and 1 and is useful for algorithms that are sensitive to the scale of the input features.

As shown in Table 4.1, results varied for better or for worse depending on the type of scaling used, and so after careful comparison each algorithm was given the data with which it performs better, leading to the results of the final column "Calibrated Data".

Table 4.1: Step 1 Results

Algorithms	Original Data	SS Data	MMS Data	Calibrated Data
Naive-Bayes	65.236%	60.354%	66.811%	67.48%
Logistic Reg.	52.638%	70.945%	67.126%	73.504%
K-NN	68.583%	69.488%	71.457%	71.024%
SVM	54.331%	65.433%	59.016%	66.063%
Decision Tree	67.874%	63.11%	51.22%	69.409%
Bagging Tree	72.165%	70.709%	53.031%	73.15%
AdaBoosted Tree	71.614%	71.417%	53.898%	73.11%
Random Forest	73.346%	74.528%	57.874%	73.386%
Neural Network	59.724%	72.126%	71.142%	73.583%
Voting Classifier	73.11%	74.409%	73.427%	75.512%

Step 2

In the second step, new attributes were included in the model, namely drug categories, as well as the functional groups of each pharmaceutical. This step aimed to improve the accuracy of the model by providing additional information about the drugs involved in the reaction. Table 4.2 shows the slight improvement that each model experienced.

Table 4.2: Step 2 Results

Algorithms	Categories + FG Included
Naive-Bayes	68.339%
Logistic Reg.	74.307%
K-NN	72.874%
SVM	66.472%
Decision Tree	70.228%
Bagging Tree	73.323%
AdaBoosted Tree	73.969%
Random Forest	73.976%
Neural Network	74.15%
Voting Classifier	76.118%

Step 3

Step three consisted of applying feature selection techniques to identify the most relevant attributes for predicting the seriousness of a reaction:

- **Variance Threshold** - Feature selection method that removes all features whose variance doesn't meet a threshold. In this study, it was responsible for removing all features with zero variance. This method is useful for removing features that don't provide much information, such as features that are always constant.
- **Tree-based Selection** - Method that, in this study used the Random Forest model to determine the importance of each feature and only keep the most important features. It can be used with any estimator that has a 'coef_' or 'feature_importances_' attribute. This method is useful for removing features that don't provide much information, such as features that are highly correlated with others.

This step was important in order to reduce the dimensionality of the data and improve model performances. Results are shown in Table 4.3.

Table 4.3: Step 3 Results

Algorithms	With Feature Selection
Naive-Bayes	69.63%
Logistic Reg.	75.197%
K-NN	73.276%
SVM	70.409%
Decision Tree	71.472%
Bagging Tree	75.882%
AdaBoosted Tree	74.071%
Random Forest	75.213%
Neural Network	75.496%
Voting Classifier	77.37%

Step 4

The final step involved editing the parameters of each used algorithm to try and extract the best possible results. This step was crucial in order to optimize the performance of the models and ensure that they were able to make somewhat accurate predictions.

It was by far the most lengthy step, since it took a high number of different parameter modifications and comparisons, in order to try and obtain the most out of each algorithm.

The following list, enumerates what parameters were altered in each algorithm:

- **Naive-Bayes** - Additive (Laplace/Lidstone) smoothing parameter was set to 0.1.

- **Logistic Regression** - The inverse of regularization strength was equal to 0.01, with the maximum number of iterations taken for the solvers to converge being set at 1000. The algorithm chosen to be used in the optimization problem was the "sag".
- **K-NN** - Number of neighbors was equal to 10. The weight function used in prediction was "distance", which means that neighbors that are closer to a query point will have more influence than further away neighbors. Algorithm used to compute the nearest neighbor was "kd_tree".
- **SVM** - Regularization parameter was set to 0.0006, keeping in mind that the strength of the regularization is inversely proportional to this value. The 'dual' setting had the False value, since the number of samples is superior to the number of features in the data.
- **Decision Tree** - The maximum depth of the tree was equal to 10. Also the number of features to consider when looking for the best split was equal to the square root of the total number of features, by assigning "sqrt" to the max_features parameter.
- **Bagging Tree** - 100 was the number of base estimators in the ensemble. The max_samples and max_features settings were both equal to 0.7.
- **AdaBoosted Tree** - The maximum number of estimators at which boosting is terminated was equivalent to 100. Additionally, the weight applied to each classifier at each boosting iteration was equal to 0.1.
- **Random Forest** - The maximum depth was set to 9.
- **Neural Network** - Parameter hidden_layer_sizes was equal to (500, 250), with the maximum number of iteration until convergence being 1000. When it comes to the activation function for the hidden layer, 'tanh' was the chosen one, which is a hyperbolic tan function, returning $f(x) = \tanh(x)$. The learning rate schedule for weight updates was set to 'adaptive', meaning that it keeps the learning rate constant to 'learning_rate_init' as long as training loss keeps decreasing. Finally, the solver for weight optimization was 'sgd', also known as stochastic gradient descent.
- **Voting Classifier** - Every previous model was included in the list of tuples referring to the 'estimators' parameter. Also 'n-jobs' was given the -1 value to ensure that all processors were being used to train the model.

Results of this step are shown in Table 4.4

Table 4.4: Step 4 Results

Algorithms	Tuned Parameters
Naive-Bayes	73.811%
Logistic Reg.	78.961%
K-NN	76.126%
SVM	78.803%
Decision Tree	75.669%
Bagging Tree	80.496%
AdaBoosted Tree	78.811%
Random Forest	81.197%
Neural Network	79.48%
Voting Classifier	81.496%

Summary and Discussion

After presenting this case study of a binary classification problem to predict whether or not an ADE involving two drugs was serious, some conclusions and insights can now be taken. The main goal of this study was to showcase the capability of Tamingo to evaluate the performance of different machine learning algorithms, as well as the impact of preprocessing and feature engineering on the final results.

The first step of the study consisted of running the algorithms with not much preprocessing done, just by applying a pair of data scaling methods. This step aimed to establish a baseline for the performance of the algorithms. The results of this step showed that the algorithms performed lower than expected, with the overall accuracy just managing to stay above the 70% mark. In the second step, extra features were added to the data, namely 'categories' and 'functional groups'. These features were created by analyzing the chemical structure of the compounds involved in the reactions. The results of this step showed that the performance of the algorithms improved ever so slightly. This could be due to the fact that since the data had hundreds of features already, and these extra ones comprised only a few dozen more, the impact turned out not to be so great.

The third step consisted of applying feature selection techniques to the data, such as variance thresholds. This step aimed to identify the most relevant features for the classification problem and to remove irrelevant or redundant features. The results of this step showed that the performance of the algorithms improved even more, with a notable increase in accuracy. It is worth mentioning that every model except for the Naive-Bayes was performing at a rate superior to 70%.

The final step of the study was about changing the parameters of each algorithm. This step aimed to optimize the performance of the algorithms by tuning their parameters. The results of this step showed that the performance of the algorithms improved significantly, with a substantial increase in accuracy. By analysing the final table, it is possible to observe that both the Random Forest and the Voting Classifier models had accuracy scores superior to 81%!

In conclusion, this case study highlights the importance of considering preprocessing and feature engineering in the development of machine learning models for binary classification problems. Using Tamingo as a basis, managing to build a model that predicts 8 out of 10 ADE cases seriousness is very promising and a good step towards achieving new breakthroughs in the pharmaceutical industry. The results of this study can be used as a guide for future research in the field and can inform the development of more accurate and reliable machine learning models for predicting serious reactions, with molecular descriptors, drug categorization and functional groups potentially on the forefront of these breakthroughs.

Chapter 5

Conclusions

In this dissertation, we have discussed the various components of the Tamingo platform, including the data preprocessing and filtering, SMILES and SMARTS compilation and manipulation, drugs categorization and functional groups, the relevance of the SDF format, the molecular descriptors calculations, and the machine learning algorithms used to predict the seriousness of reactions and the outcomes of reactions. We have also discussed the use of a platform like Tamingo in the context of ADEs, and the potential benefits of using machine learning and molecular descriptors to improve our understanding of ADEs and help prevent these events.

Tamingo is a valuable and all-around health informatics tool. The platform uses machine learning algorithms and molecular descriptors to analyze large amounts of data and make accurate predictions. With further development and refinement, it has the potential to significantly improve our understanding of ADEs and help prevent these events.

5.1 Future Work

The Tamingo platform is a powerful and useful tool that can be applied to many areas in health informatics. However, there is still much room for improvement and further development that can be done to enhance the capabilities of the platform.

One possible enhancement would be to enrich the quantity of ADE data included in the platform's database, since there is no such thing as too much data, as long as it has quality content. This expansion of the database might allow the platform to make more accurate predictions and provide more information to users.

Another area of future work is to improve the accuracy of the platform's predictions. The platform currently uses a range of attributes and molecular descriptors to make predictions. However, there may be other attributes or even more molecular descriptors that could be used to improve the accuracy of the predictions.

A third area of future work is to incorporate more advanced machine learning techniques. The platform currently uses a range of machine learning algorithms, such as logistic regression and decision trees. However, there are other machine learning techniques that could be used to improve

the platform's performance. For example, deep learning techniques such as convolutional neural networks and recurrent neural networks could be used to analyze the data and make predictions.

Another area of future work is to develop more refined feature selection techniques. The platform currently uses basic feature selection techniques, such as variance threshold and correlation-based feature selection. However, more advanced feature selection techniques, such as genetic algorithms or particle swarm optimization, could be used to identify the most relevant attributes and molecular descriptors for making predictions.

Finally, it would be useful to explore the possibility of applying the Tamingo platform to other medical/pharmaceutical areas. The platform currently focuses on ADEs, but it could be applied to other areas such as predicting the effectiveness of a treatment or the risk of a disease.

In conclusion, there are many areas of future work that can be done to further develop and improve Tamingo, with the goal of providing even more valuable insights to users.

5.2 Reflection and Future Directions

Throughout this dissertation project, and the consequent development of Tamingo, a data mining platform for ADEs, a lot was accomplished.

Firstly, it is important to highlight the technical achievements of the project. Tamingo was built using a variety of technologies, including Django, and is capable of handling large amounts of data from the OpenFDA database. The platform is able to parse, filter, and preprocess ADE data in a structured and efficient manner, allowing users to easily access and analyze this information. Additionally, Tamingo includes the option to create machine learning models based on binary classification or multiclass classification, with feature selection, and algorithm optimization, which enables users to extract valuable insights from the data.

Furthermore, this project has also been a personal growth experience. Throughout the project, I have developed a deep understanding of data mining techniques and their application in the field of ADEs. I also increased my understanding of different technologies, such as Django or SQLite, and how to use them to create a functional web application. Furthermore, I have also learnt how to work independently and how to manage my time effectively, which are skills that will be valuable throughout my professional career.

Overall, this dissertation project has been a challenging but extremely rewarding experience. Tamingo is a functional and useful platform that has the potential to contribute to the field of ADEs, and the knowledge and skills that I have acquired during its development will remain and accompany me in my future challenges.

References

- [1] Ahmed Al-Temimi. Scope of pharmacovigilance: Comprehensive review. *Chemical Science International Journal*, 31:29–39, 12 2022.
- [2] Salem Alaisawi and Salem Khalifa. Artificial intelligence in data mining and big data. *Data Mining*, 01 2020.
- [3] Amir Ali and Muhammad Amin. *Hands-On Machine Learning with Scikit-Learn*. Amazon Kindle Direct Publishing, 03 2019.
- [4] Grant Allen and Mike Owens. The definitive guide to sqlite. *The Expert’s Voice in Open Source*, 01 2010.
- [5] Mohammed Alvi and Parnali Chatterjee. *Excipients and Active Pharmaceutical Ingredients*, chapter 24, pages 347–361. Mohammed Alvi, 01 2014.
- [6] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. Machine learning from theory to algorithms: An overview. *Journal of Physics: Conference Series*, 1142:012012, 11 2018.
- [7] M. Bharati and Bharati Ramageri. Data mining techniques and applications. *Indian Journal of Computer Science and Engineering*, 1, 12 2010.
- [8] Lars Bjerrum, Beatriz López-Valcárcel, and Gert Petersen. Risk factors for potential drug interactions in general practice. *The European journal of general practice*, 14:23–9, 02 2008.
- [9] Meichun Cai, Quan Xu, Yan-Jing Pan, Wen Pan, Nan Ji, Yin-Bo Li, Haijing Jin, Ke Liu, and Zhi-Liang Ji. Adreco: An ontology database for aiding standardization and hierarchical classification of adverse drug reaction terms. *Nucleic acids research*, 43, 10 2014.
- [10] Christina Charlesworth, Ellen Smit, David Lee, Fatimah Alramadhan, and Michelle Odden. Polypharmacy among adults aged 65 years and older in the united states: 1988-2010. *The journals of gerontology. Series A, Biological sciences and medical sciences*, 70, 03 2015.
- [11] Murat Cokol. Drugs and their interactions. *Current drug discovery technologies*, 10, 12 2012.
- [12] Khushal Dalvi, Prof Kasar, Parikshit Pise, Sandesh More, and Mangesh Hadawale. Coding in pharmacovigilance using meddra: A review. *International Journal of Advanced Research in Science, Communication and Technology*, pages 236–240, 06 2022.
- [13] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. Big data in healthcare: management, analysis and future prospects. *J. Big Data*, 6(1), December 2019.

- [14] Dian Eka Ratnawati, Marjono Marjono, and Syaiful Anam. Prediction of active compounds from smiles codes using backpropagation algorithm. In *Coverage of Basic Sciences toward the World's Sustainability Challenges*, volume 2021, page 060009, 10 2018.
- [15] Nikolas Fechner, Georg Hinselmann, and Joerg Wegner. *Molecular Descriptors*, pages 89–145. Taylor & Francis Group, 01 2010.
- [16] Thomas Haslwanter. *Data Filtering*, pages 71–104. Springer, 06 2021.
- [17] Thomas Haslwanter. *Data Filtering*, pages 71–104. Springer, 06 2021.
- [18] Michael Imhoff, Andrew Webb, and Andreas Goldschmidt. Health informatics. *Intensive care medicine*, 27:179–86, 02 2001.
- [19] Keerthana Jaganathan, Hilal Tayara, and Kil Chong. An explainable supervised machine learning model for predicting respiratory toxicity of chemicals using optimal molecular descriptors. *Pharmaceutics*, 14:832, 04 2022.
- [20] Guoqian Jiang, Hongfang Liu, Harold Solbrig, and Christopher Chute. Mining severe drug-drug interaction adverse events using semantic web technologies: A case study. *BioData mining*, 8:12, 06 2015.
- [21] Taha Kass-Hout, Zhiheng Xu, Matthew Mohebbi, Hans Nelsen, Adam Baker, Jonathan Levine, Elaine Johanson, and Roselie Bright. Openfda: An innovative platform providing access to a wealth of fda's publicly available data. *Journal of the American Medical Informatics Association : JAMIA*, 23, 12 2015.
- [22] Sunghwan Kim. Exploring chemical information in pubchem. *Current Protocols*, 1, 08 2021.
- [23] Chuenjid Kongkaew, Peter Noyce, and Darren Ashcroft. Hospital admissions associated with adverse drug reactions: A systematic review of prospective observational studies. *The Annals of pharmacotherapy*, 42:1017–25, 08 2008.
- [24] Panos Kouvelis, Joseph Milner, and Zhili Tian. Clinical trials for new drug development: Optimal investment and application. *Manufacturing and Service Operations Management*, 19, 06 2017.
- [25] Sophia Krix, Lauren DeLong, Sumit Madan, Daniel Domingo-Fernández, Ashar Ahmad, Sheraz Gul, Andrea Zaliani, and Holger Fröhlich. Multigml: Multimodal graph machine learning for prediction of adverse drug events. *Research Gate*, 12 2022.
- [26] Michael Kuhn, Ivica Letunic, Lars Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44, 10 2015.
- [27] Rakesh Kumar. Future for scientific computing using python. *International Journal of Engineering Technologies and Management Research*, 2:30–41, 01 2020.
- [28] Hui-chao Li and Shun-fa Shen. Construction of college students' physical health data sharing system based on django framework. *Journal of Sensors*, 2021:1–7, 11 2021.
- [29] Jin Li, Xiangmin Ji, and Liyan Hua. Improving the prediction of adverse drug events using feature fusion-based predictive network models. *IEEE Access*, PP:1–1, 03 2020.

- [30] Yangguang Liu, Yangming Zhou, Shiting Wen, and Chaogang Tang. A strategy on selecting performance metrics for classifier evaluation. *International Journal of Mobile Computing and Multimedia Communications*, 6:20–35, 10 2014.
- [31] Mudasir Maqbool, Mohmad Dar, Shafiq Rasool, Ahsan Ullaha Bhat, and Mohammad Geer. Drug safety and pharmacovigilance: An overview. *PV*, 9:543–548, 04 2019.
- [32] Janja Marc. 7. pharmacogenetics of drug receptors. *EJIFCC*, 19, 04 2008.
- [33] L. Meng, X.-W Tang, H.-H Ji, L. Song, X.-D Niu, and Too Jia. Mining and evaluation of statin-associated adverse events signals: data mining of the public version of the openfda adverse event reporting system. *Chinese Journal of New Drugs*, 28:244–248, 01 2019.
- [34] Jean-Louis Montastruc, Margaux Lafaurie, Claire Canecaude, Genevieve Durrieu, Agnès Sommet, François Montastruc, and Haleh Bagheri. Fatal adverse drug reactions: a world-wide perspective in the who pharmacovigilance database. *British Journal of Clinical Pharmacology*, 87, 04 2021.
- [35] Madhav Mutalik and Dhara Sanghavi. Review of drug interactions: A comprehensive update. *British Journal of Pharmaceutical Research*, 4:954–980, 03 2014.
- [36] Diogo Pinto, Pedro Costa, Rui Camacho, and Vítor Santos Costa. Predicting drugs adverse side-effects using a recommender-system. In Nathalie Japkowicz and Stan Matwin, editors, *Discovery Science*, pages 201–208, Cham, 2015. Springer International Publishing.
- [37] Jinan Redha. A review of clustering algorithms. *International Journal of Computer Science and Mobile Applications*, 10 2022.
- [38] Saif ur Rehman, Khan Ullah, and Simon Fong. Graph mining: A survey of graph mining techniques. In *Conference: 2012 Seventh International Conference on Digital Information Management*, 08 2012.
- [39] Blagoj Ristevski and Ming Chen. Big data analytics in medicine and healthcare. *Journal of Integrative Bioinformatics*, 15, 05 2018.
- [40] Ankur Rohilla and Singhraj Yadav. Adverse drug reactions: An overview. *International Journal of Pharmacological Research*, 3, 04 2013.
- [41] Soraya Sedkaoui and Mounia Khelfaoui. *Classification Algorithms*, pages 171–194. ISTE, 01 2020.
- [42] Hrvoj Vancik. *Functional Groups*, pages 19–22. Springer, 01 2022.
- [43] Stephan Wagner, Hua Wang, and Gang Yu. Molecular graphs and the inverse wiener index problem. *Discrete Applied Mathematics*, 157:1544–1554, 04 2009.
- [44] Andrew Wilson, Lehana Thabane, and Anne Holbrook. Application of data mining data in pharmacovigilance. *British journal of clinical pharmacology*, 57:127–34, 03 2004.
- [45] Iwan Wiratmadja, Siti Salamah, and Rajesri Govindaraju. Healthcare data mining: Predicting hospital length of stay of dengue patients. *Journal of Engineering and Technological Sciences*, 50:110–126, 03 2018.

- [46] David Wishart, Yannick Djoumbou, An Chi Guo, Elvis Lo, Ana Marcu, Jason Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, and Michael Wilson. Drugbank 5.0: A major update to the drugbank database for 2018. *Nucleic acids research*, 46, 11 2017.

Appendix A

OpenFDA Data Parser

```
data = json.load(f)

for i in data['results']:

    skip = False

    serious = i.get('serious')

    if serious == "1":
        serious = "Yes"
    elif serious == "2":
        serious = "No"
    else:
        skip = True

    if i.get('patient').get('patientsex') != None:
        patient_sex = i.get('patient').get('patientsex')

        if patient_sex == "1":
            patient_sex = "Male"
        elif patient_sex == "2":
            patient_sex = "Female"
        else:
            patient_sex = "Unknown"
            skip = True
    else:
        patient_sex = "Unknown"
        skip = True
```

```
if i.get('patient').get('patientweight') != None:
    patient_weight = i.get('patient').get('patientweight')
else:
    patient_weight = "Unknown"
    skip = True

reaction = ""
drugs = ""
categories = ""
outcome = ""

print(n)
n+=1

for j in i.get('patient').get('reaction'):
    if skip == True:
        break

    reaction_description = j['reactionmeddrapt']

    try:
        reaction_outcome = j['reactionoutcome']

        if reaction_outcome == "1":
            reaction_outcome = "Recovered"
        elif reaction_outcome == "2":
            reaction_outcome = "Recovering"
        elif reaction_outcome == "3":
            reaction_outcome = "Not recovered"
        elif reaction_outcome == "4":
            reaction_outcome = "Recovered with sequelae"
        elif reaction_outcome == "5":
            reaction_outcome = "Fatal"
        elif reaction_outcome == "6":
            reaction_outcome = "Unknown"
    except KeyError:
        reaction_outcome = "Unknown"

    reaction += reaction_description + " | "
```

```
outcome += reaction_outcome + " | "

#remove duplicate entries
reaction = ' | '.join(dict.fromkeys(reaction.split(" | ")))
#remove duplicate entries
outcome = ' | '.join(dict.fromkeys(outcome.split(" | ")))

if skip == False:
    if "Unknown" in outcome:
        outcome = outcome.replace("Unknown | ", "")
    if len(outcome.split(" | ")) != 2:
        skip = True

for j in i.get('patient').get('drug'):
    if skip == True:
        break

    if j['drugcharacterization'] == "1":
        drug_characterization = "Suspect"
    elif j['drugcharacterization'] == "2":
        drug_characterization = "Concomitant"
    elif j['drugcharacterization'] == "3":
        drug_characterization = "Interacting"

    drugs += j['medicinalproduct'] + "(" + drug_characterization + ") | "

#remove duplicate entries
drugs = ' | '.join(dict.fromkeys(drugs.split(" | ")))
```

Appendix B

Functional Groups SMARTS

Acid_Chloride: C(=O)Cl
Carboxylic_Acid: C(=O)[O;H,-]
Sulfonyl_Chloride: [\$(S-!@[#6])] (=O) (=O) (Cl)
Amine: [N;\$ (N-[#6]); !\$(N-[!#6;!#1]); !\$(N-C=[O,N,S])]
Amide: [NX3][CX3] (= [OX1]) [#6]
Boronic_Acid: [\$(B-!@[#6])] (O) (O)
Isocyanate: [\$(N-!@[#6])] (= !@C= !@O)
Alcohol: [O;H1;\$ (O-!@[#6;!\$(C= !@[O,N,S]))]
Aldehyde: [CH;D2;!\$(C-[!#6;!#1])]=O
Halogen: [\$([F,Cl,Br,I]-!@[#6]); !\$([F,Cl,Br,I]-!@C-!@[F,Cl,Br,I]); !\$([F,Cl,Br,I]-[C,S] (= [O,S,N]))]
Azide: [N;H0;\$ (N-[#6]); D2]=[N;D2]=[N;D1]
Nitro: [N;H0;\$ (N-[#6]); D3] (= [O;D1]) ~ [O;D1]
Terminal_Alkyne: [C;\$ (C#[CH])]
Imine: [CX3;\$ ([C] ([#6]) [#6]), \$ ([CH] [#6])] = [NX2] [#6]
Imide: [CX3] (= [OX1]) [NX3H] [CX3] (= [OX1])
Thiol: [#16X2H]
Benzene_Ring: [\$([cR1]1[cR1][cR1][cR1][cR1][cR1]1), \$(c1cccc1cccc2)]
Ether: [OD2] ([#6]) [#6]
Ester: [#6][CX3] (=O) [OX2H0] [#6]
Ketone: [#6][CX3] (=O) [#6]
Enol: [OX2H] [#6X3] = [#6]
Phenol: [OX2H] [cX3]:[c]
Hydroxyl: [OX2H]

Appendix C

Snippet of SDF Compiled by Tamingo

```
RDKit          2D

15 15  0  0  0  0  0  0  0  0  0999 V2000
    5.2500   -1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.7500   -1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.0000   -2.5981    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.5000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7500   -1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.7500   -1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.5000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.7500    1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7500    1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -3.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -3.7500    1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -3.7500   -1.2990    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -3.0000   -2.5981    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -5.2500   -1.2990    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0

 1  2  1  0
 2  3  1  0
 2  4  1  0
 4  5  1  0
 5  6  2  0
 6  7  1  0
 7  8  2  0
 8  9  1  0
 9 10  2  0
```

```
      8 11  1  0
     11 12  1  0
     11 13  1  0
     13 14  2  0
     13 15  1  0
     10  5  1  0
M  END
>  <SMILES>  (1)
CC(C)CC1=CC=C(C=C1)C(C)C(=O)O

>  <Drug>  (1)
ibuprofen

$$$$
```

Appendix D

Function Converting SMILES data to SDF Format

```
with open('smiles.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    # write multiple rows
    writer.writerows(smiles_data)

pp = pd.read_csv('smiles.csv', names=['SMILES', 'Drug'])
PandasTools.AddMoleculeColumnToFrame(pp, 'SMILES', 'Molecule')
PandasTools.WriteSDF(pp, 'smiles.sdf', molColName='Molecule',
properties=list(pp.columns))
```

Appendix E

OpenFDA Data Snippet (JSON)

```
"safetyreportversion": "5",
  "safetyreportid": "14830242",
  "primarysourcecountry": "US",
  "occurcountry": "US",
  "transmissiondateformat": "102",
  "transmissiondate": "20220303",
  "reporttype": "2",
  "serious": "1",
  "seriousnessdeath": "2",
  "seriousnesslifethreatening": "2",
  "seriousnesshospitalization": "2",
  "seriousnessdisabling": "2",
  "seriousnesscongenitalanomaly": "2",
  "seriousnessother": "1",
  "receivedateformat": "102",
  "receivedate": "20180430",
  "receiptdateformat": "102",
  "receiptdate": "20211129",
  "fulfillexpeditecriteria": "1",
  "companynumb": "US-JAZZ-2018-US-006718",
  "primarysource": {
    "reportercountry": "US",
    "qualification": "1"
  },
  "sender": {
    "sendertype": "2",
    "senderorganization": "FDA-Public Use"
  },
```

```
"receiver": {
  "receivertype": "6",
  "receiverorganization": "FDA"
},
"patient": {
  "patientsex": "1",
  "reaction": [
    {
      "reactionmeddraversionpt": "24.1",
      "reactionmeddrapt": "Aortic stenosis",
      "reactionoutcome": "3"
    }
  ],
"drug": [
  {
    "drugcharacterization": "1",
    "medicinalproduct": "XYREM",
    "drugauthorizationnumb": "021196",
    "drugstructuredosagenumb": "2.25",
    "drugstructuredosageunit": "002",
    "drugseparatedosagenumb": "1",
    "drugintervallosageunitnumb": "12",
    "drugintervaldosagedefinition": "805",
    "drugdosagetext": "2.25 G, BID",
    "drugdosageform": "Oral solution",
    "drugadministrationroute": "048",
    "drugindication": "Narcolepsy",
    "drugstartdateformat": "610",
    "drugstartdate": "200306",
    "drugenddateformat": "610",
    "drugenddate": "200309",
    "actiondrug": "5",
    "drugadditional": "3",
    "activesubstance": {
      "activesubstancename": "SODIUM OXYBATE"
    }
  }
]
```