**Universidade de Aveiro** Departamento de Eletrónica,
Telecomunicações e Informática
**2020**

**Luís Miguel
Marques Fonseca**

# ***Machine Learning*** **para deteção de padrões e previsão de ocorrências criminais**

# **Machine Learning for pattern detection and prediction of criminal occurrences**

**Universidade de Aveiro** Departamento de Eletrónica,
Telecomunicações e Informática
**2020**

**Luís Miguel
Marques Fonseca**

# ***Machine Learning*** **para deteção de padrões e previsão de ocorrências criminais**

# Machine Learning for pattern detection and prediction of criminal occurrences

"*If things are not failing, you are not innovating enough*"

— Elon Musk

**Universidade de Aveiro** Departamento de Eletrónica,
**2020** Telecomunicações e Informática

**Luís Miguel
Marques Fonseca**

# *Machine Learning* para deteção de padrões e previsão de ocorrências criminais

# Machine Learning for pattern detection and prediction of criminal occurrences

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica da Doutora Susana Isabel Barreto de Miranda Sargento, Professora Catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Filipe Cabral Pinto, Consultor Sénior da Altice Labs.

**o júri / the jury**

presidente / president

Professor Doutor Luís Filipe de Seabra Lopes
professor associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Professora Doutora Ana Cristina Wanzeller Guedes de Lacerda
professora adjunta da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viseu

Professora Doutora Susana Isabel Barreto de Miranda Sargento
professora catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (orientadora)

**agradecimentos / acknowledgements**

Com a finalização desta dissertação não posso deixar de agradecer a algumas pessoas que, direta ou indiretamente, contribuiram nesta etapa tão importante da minha vida pessoal e profissional.
Em primeiro lugar, agradeço à Professora Doutora Susana Sargento e do Doutor Filipe Cabral Pinto pela orientação prestada na elaboração e desenvolvimento do trabalho apresentado.
À Altice Labs, por me ter dado a oportunidade de realizar a dissertação integrada num ambiente de trabalho profissional, especialmente aos meus colegas de equipa pelo apoio e debate de temas que contribuiram para a tomada de algumas decisões.
Ao Instituto de Telecomunicações, pelas condições proporcionadas.
Aos meus pais e ao meu irmão, por todo o apoio e incentivo que me deram ao longo desta etapa da minha formação.

**Palavras Chave** *Machine Learning*; Previsão de Crimes; Cidades inteligentes; *Big data*.

## Resumo

O aumento da população mundial, especialmente nos grandes centros urbanos, tem resultado em novos desafios tais como a gestão de recursos naturais, gestão de infraestruturas, bem como a otimização dos serviços para promover a qualidade de vida dos cidadãos.

Um dos maiores e mais importantes desafios é a gestão da segurança pública. Para além de ser um fator de interesse quer da população em geral quer das autoridades, também é um domínio que influencia outros indicadores essenciais numa cidade como o turismo e o emprego. A segurança pública reflete-se no crescimento económico e no desenvolvimento social de uma comunidade.

Nesta dissertação é proposta uma solução para previsão de ocorrências criminais numa cidade baseada em dados de histórico de incidentes e dados demográficos. Será apresentado todo o ciclo de vida do processo de aprendizagem do modelo para dotar uma organização da capacidade preditiva: desde a recolha dos dados da sua fonte de origem, o tratamento e transformações aplicadas aos mesmos, escolha, avaliação e implementação do modelo de *Machine Learning* até à camada de aplicação.

Serão implementados modelos de classificação para previsão do risco criminal para um dado intervalo temporal e localização, e modelos de regressão para previsão do número de crimes. Irão ser utilizados algoritmos de *Machine Learning* como *Random Forest*, Redes Neuronais, *K-Nearest Neighbors* e Regressão Logística para a aprendizagem do modelo de previsão de ocorrências onde serão comparados os seus desempenhos de acordo com o tratamento e transformação dos dados utilizados. Os resultados do modelo escolhido evidenciam que a utilização de técnicas de *Machine Learning* auxiliam a antecipação de ocorrências criminais, o que contribuiu para o reforço da segurança pública.

Por fim, irá ser procedida a implementação dos modelos numa plataforma que fornece uma API para que entidades externas possam solicitar previsões em tempo real. Será também apresentada a aplicação onde é possível mostrar visualmente as previsões de ocorrências criminais.

**Keywords** Machine Learning; Crime Prediction; Smart Cities; Big data.

## Abstract

The increase of the world population, especially in large urban centers, has resulted in new challenges such as the management of natural resources and infrastructures as well as the optimization of services to promote the quality of citizens' life.

One of the biggest and most important challenges is the management of public safety, since, in addition to being a factor of interest to both the general population and the authorities, it is also an area that influences other essential indicators in a city such as tourism and employment. Public Safety has impact on the economic growth and social development of a community.

This dissertation proposes a solution for the prediction of criminal occurrences in a city based on historical data of incidents and demographic data. The entire life cycle of the model's learning process will be presented to provide an organization with predictive capability: start with the data collection from its original source, the treatment and transformations applied to them, the choice and the evaluation and implementation of the Machine Learning model up to the application layer.

Classification models will be implemented to predict criminal risk for a given time interval and location, as well as regression models to predict the number of crimes. Machine Learning algorithms, such as Random Forest, Neural Networks, K-Nearest Neighbors and Logistic Regression will be used to predict occurrences, and their performance will be compared according to the data processing and transformation used. The results of the chosen model show that the use of Machine Learning techniques helps to anticipate criminal occurrences, which contributed to the reinforcement of public security.

Finally, the models will be implemented on a platform that provides an API to enable other entities to request for predictions in real-time. An application will also be presented where it is possible to show criminal occurrences predictions visually.

# Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| **IoT** | Internet of Things |
| **ICT** | Information and Communications Technologies |
| **RDBMS** | Relational Database Management System |
| **XML** | Extensible Markup Language |
| **JSON** | JavaScript Object Notation |
| **CSV** | Comma-separated values |
| **HTML** | HyperText Markup Language |
| **HDFS** | Hadoop Distributed File System |
| **NoSQL** | Not Only SQL |
| **ML** | Machine Learning |
| **AI** | Artificial Intelligence |
| **API** | Application Programming Interface |
| **RFID** | Radio-Frequency Identification |
| **IT** | Information Technology |
| **SFTP** | SSH File Transfer Protocol |
| **HTTP** | Hypertext Transfer Protocol |
| **API** | Application Programming Interface |
| **SSH** | Secure Shell |
| **MAE** | Mean Absolute Error |
| **MSE** | Mean Squared Error |
| **RF** | Random forest |
| **DT** | Decision tree |
| **KNN** | K-Nearest Neighbors |
| **NN** | Neural Network |
| **LR** | Logistic Regression |
| **EU** | European Union |
| **UI** | User Interface |
| **REST** | Representational State Transfer |

CHAPTER 1

# Introduction

This chapter describes the overall motivation for addressing the topics of this dissertation, which aims to be a contribution to the response to urban population growth and the side effects inherent to this phenomenon. This chapter also covers the main contributions of this document and a summary of the structure of the document.

## 1.1 Motivation

Currently, the world is going through an accelerated process of urbanization. It has been notorious in recent years the movement of people to cities; and according to projections, by the United Nations, in 2050, almost 70% of the population will live in urban areas [1]. Figure 1.1 shows a comparison of rural and urban population growth, relative to the total world population. These projections are based on the UN World Urbanization Prospects [2]. As evidenced, the ratio of the population living in urban areas compared to those living in rural areas has been increasing and, by 2050 forecasts, this dissimilarity will continue to expand.

As a result of this large influx of people into urban centers, cities have embraced technologies such as Big Data and the Internet of Things (IoT) to obtain data for many applications such as smart street lighting, smart parking or waste management, among others. Internet of Things is one of the big data sources, but all city management processes, supported by different systems, generate true big data. These are called the cities of the future [1].

However, cities have been unable to take full advantage of the data collected, because much of this data is siled to serve specific needs without any links between activity domains. Moreover, data comes in different formats, without a common data element organization model, rather than being open and used to contribute to a common need, such as, improve the quality of the life of the citizens by providing intelligent services in a large variety of aspects such as transportation, healthcare, environment, energy and public safety. In Information

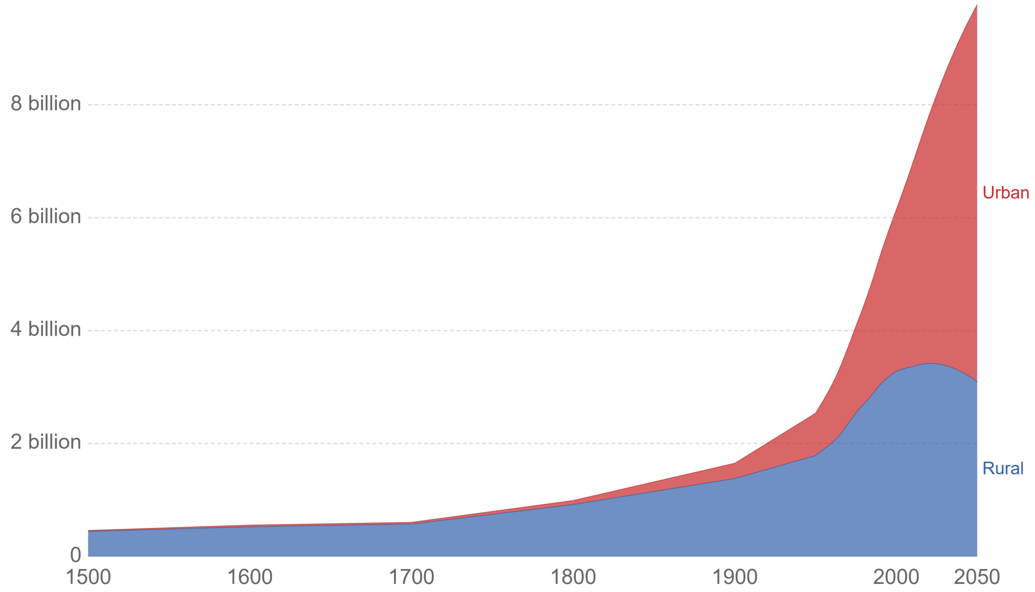


**Figure 1.1:** Urban and rural population projected to 2050 [3]

Technology (IT), a silo describes any management system that is unable to operate with any other system.

Therefore, one of the biggest issues for smart cities is how to handle immeasurable amounts of information generated by organizations, systems, and people every day. However, useful information can be extracted with data analysis that can assist the development of a smart city. In this way, it is a challenge to take full advantage of data produced by cities and their ecosystems.

Combining strategic policies promotes sustainable development, economic growth, and better living conditions for citizens. In this sense, data mining and machine learning techniques can be applied to smart cities applications in order to get insights from the urban space, easing the sustainable development [4].

Over time, there has been a growing concern among the population regarding safety. There are several causes for this, among them the high number of terrorist attacks that occurred in the decade of 2010, as can be seen in the Figure 1.2. Although in recent years the number of attacks has been decreasing, that number remains much higher compared to years before 2010[1]. Another reason, as also stated, is the increase of urban population that has as result a large agglomerate of people, that is, the increase of population density, which consequently generates more disorder and stress.

Indeed, this is a "hot topic" for a Smart City considering the common interest of citizens and authorities, since a safer city is reflected in the comfort and confidence of the general population.

[1] The source, Global Terrorism Database, defines terrorism as “acts of violence by non-state actors, perpetrated against civilian populations, intended to cause fear, in order to achieve a political objective.” Its definition excludes violence initiated by governments (state terrorism) and open combat between opposing armed forces.

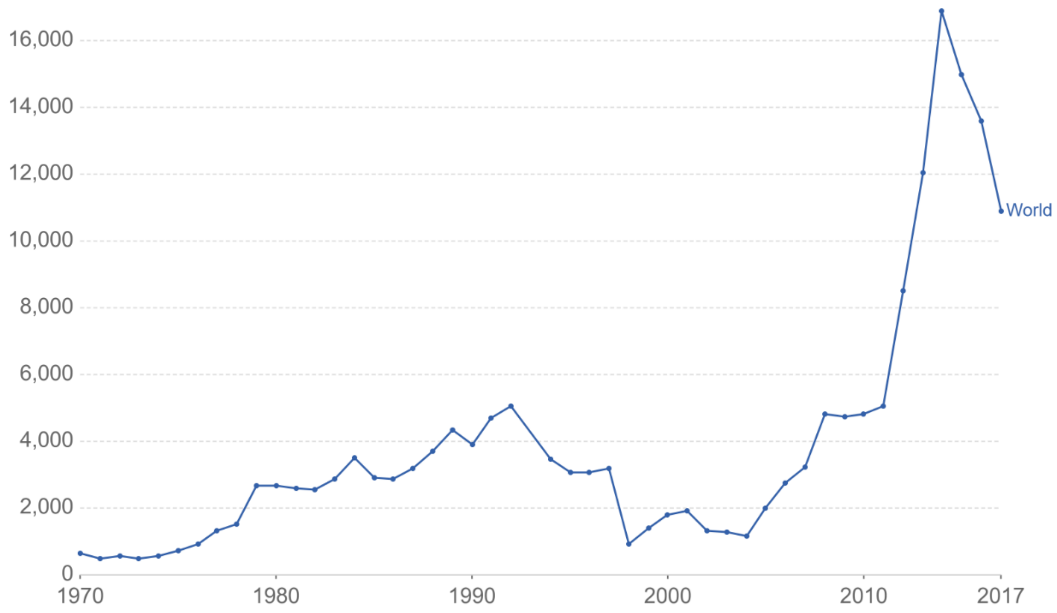


**Figure 1.2:** Number of terrorist attacks [5]

## 1.2 Objectives

Overall, the objective of this dissertation is to provide a solution to monitor and predict criminal incidences in order to provide citizens and city authorities knowledge about the most dangerous areas, thus, adding value to the city for improving public safety.

In this sense, this type of prediction can be useful in several ways, from more optimized and effective design of patrol routes, to tourists who are unaware of the most dangerous areas of cities.

The major challenges are the extraction, ingestion and analysis of information from the cities' technological ecosystem data. Data disparity, heterogeneity and volume make it difficult to correlate and extract value. Thus, the aim is to use Machine Learning techniques for pattern detection and prediction of occurrences in the city context.

In this line of reasoning, for the development of this dissertation there are some specific objectives, such as:

1. Definition of a use case focused on public safety issue, supported by the global information available, in order to explore the application of machine learning techniques for pattern detection and prediction of occurrences.
2. Study and analysis of data provided by the cities - characterization of available sources, identification of relevant flows and complementary information needed to enrich the data at work.
3. Explore the classification and regression algorithms to provide a qualitative and quantitative prediction of crimes, and also analyze and compare their performance.
4. Presentation of criminal predictions in a graphical interface to enable the user to explore and take benefit of the information provided.

## 1.3 Contributions

The work in this dissertation aims to contribute to the improvement of public safety based on the knowledge extracted from data using IT tools and technologies. The work is divided into the following steps:

- Analysis and processing of data made available by security authorities, and verify whether information and knowledge can be extracted from them;
- Implementation of a classification model where the goal is to predict the criminal risk for a given location and time interval;
- Implementation of a regression model where the goal is to predict the exact number of crimes that will occur for a given location and time interval;
- Development of an architecture that provides an API for other entities to consult criminal prediction.

A conference paper entitled "An Application for Risk of Crime Prediction Using Machine Learning" has been submitted and accepted in International Conference on Machine Learning and Applications 2020. This paper focuses on predicting criminal occurrences using a classification model. A new paper that extends with the overall work is being prepared.

## 1.4 Document Structure

This document contains 8 chapters. At the beginning of each chapter, there is an introduction to the subjects that will be addressed, and at the end, there is a summary to highlight the most important points. The document is structured as follows:

- **Chapter 2 - State of the Art**: This chapter starts by covering the terms of big data, smart cities and data platforms including the analysis of some architectures. A conceptual approach to machine learning will be done, where some related works about the exploitation of criminal data and crime prediction using machine learning techniques will be presented.
- **Chapter 3 - Study Scenario**: This chapter describes the scenario chosen and the dataset used in this work, which fits in the public safety of the city, more properly in the crime prediction.
- **Chapter 4 - Data Preparation**: Taking into account the scenario, a set of operations were carried out in order to optimize the quality of the data. In this chapter, all of these operations will be described.
- **Chapter 5 - Classification Model**: This chapter describes the entire process in the development of the crime risk classification model, where some approaches to model will be presented and tested with four machine learning algorithms. Finally, the results obtained will be presented and discussed.
- **Chapter 6 - Regression Model**: This chapter describes the entire process in the development of a regression model, that is, aiming at predicting the number of crimes. The results obtained from the algorithms will be presented and discussed.

- **Chapter 7 - Model Deploy and Application**: This chapter describes the last step in a machine learning project, the model deployment, where the model is applied in a system where several entities can interact with it. Also, it will be shown an example of a client application that makes requests to model and present the results.
- **Chapter 8 - Conclusions and Future Work**: This chapter concludes this document, systematizing the work done and presenting proposals for future work.

CHAPTER 2

# State of the Art

This chapter describes the concepts that will be addressed in this dissertation. Firstly, it will be framed on the themes of big data and smart cities. Subsequently, it will be explored machine learning topics, where a conceptual approach and the algorithms used in this work will be explained. Finally, it will be presented related works about the exploitation of criminal data and crime prediction using machine learning techniques.

The contents presented here have as main source the literature from the scientific community, together with publications of a technical nature related to this theme, in order to obtain reliable and quality information.

## 2.1 Big Data and Platforms

A huge amount of data is generated every day and it comes from everywhere. Small quotidian actions generate data such as sending an email or an online search. According to Forbes, in 2018, 2.5 quintillion bytes of data were created each day, and moreover, it is also stated that 90% of the data was generated in the last two years [6].

Figure 2.1 shows a prediction of digital data growth until 2025 [7]. As can be seen, it is notorious the growth of generated data. Notice that currently, the digital data is around 50 zettabytes (ZB), but according to the forecast, the size of this data will almost quadruplicate in the next five years. There are a lot of factors that influence this phenomenon, for example, the increasing number of people and cities connected to the internet, that is, the digital transformation.

For a better understanding of the magnitude of these numbers that measures the quantity of data, Table 2.1 shows a comparison of the units of measurement of information, from a single bit to yottabyte.

Due to its complexity, it is important to note that the concept of big data is an abstraction, since it is not just a huge volume of data, but a whole set of characteristics that defines

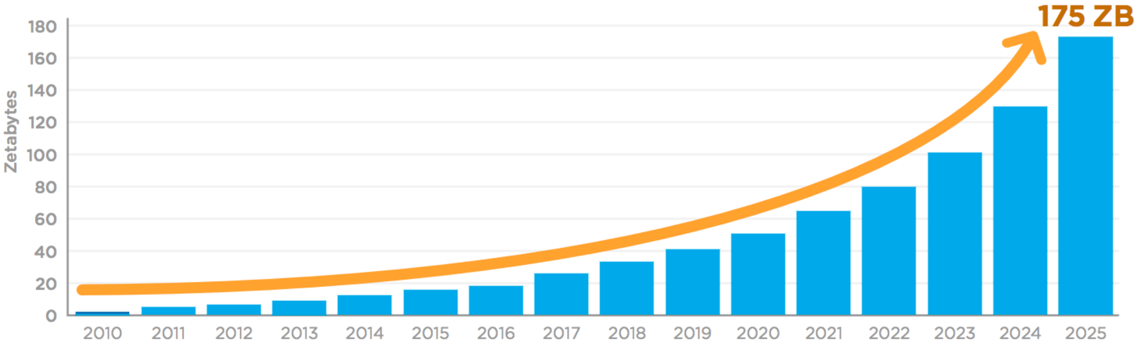


**Figure 2.1:** Digital data growth over time [7]

**Table 2.1:** Standard units of measurement used for data storage

| Unit | Equal to: | Size in Bytes |
|---|---|---|
| bit (b) | 0 or 1 | 1/8 |
| byte (B) | 8 bits | 1 |
| kilobyte (KB) | 1,024 bytes | 1,024 |
| megabyte (MB) | 1,024 kilobytes | 1,048,576 |
| gigabyte (GB) | 1,024 megabytes | 1,073,741,824 |
| terabyte (TB) | 1,024 gigabytes | 1,099,511,627,776 |
| petabyte (PB) | 1,024 terabytes | 1,125,899,906,842,624 |
| exabyte (EB) | 1,024 petabytes | 1,152,921,504,606,846,976 |
| zettabyte (ZB) | 1,024 exabytes | 1,180,591,620,717,411,303,424 |
| yottabyte (YB) | 1,024 zettabytes | 1,208,925,819,614,629,174,706,176 |

it, however, based on this significant data increase, the term of big data is mainly used to describe enormous datasets [8]. Big data refers to the phenomenon of managing enormous amounts of data that can be structured, semi-structured or unstructured. It is often difficult to analyze and visualize data of this magnitude [9].

There are many sources of data such as sensors of IoT applications, online transactions, emails, videos, searches queries, health records, social networking interactions, among others [10]. Thus, all data must be analyzed to gain and enhance insight through advanced processing, enabling the extraction of information and knowledge.

In Big Data, data is commonly characterized by V's, in which they stand out, Volume, Velocity and Variety [11]:

- **Volume**: the amount of data, normally, big data leads with Petabytes or more of data;
- **Velocity**: how fast the data is generated, streamed and aggregated;
- **Variety**: data is generated in several formats, being it numbers, text documents, audio, etc.

Table 2.2 shows the main differences between traditional data and big data.

**Table 2.2:** Comparison between tradition data and big data [12].

| | **Traditional Data** | **Big Data** |
|---|---|---|
| **Volume** | GB | constantly updated (PB currently) |
| **Generated Data** | per hour, day, ... | faster |
| **Structure** | structured | semi-structured or un-structured |
| **Data Source** | centralized | fully distributed |
| **Data Integration** | easy | difficult |
| **Data Storage** | RDBMS | HDFS, NoSQL |
| **Access** | interactive | batch or near real-time |

Structured data is usually stored in Relational Database Management Systems (RDBMSs) being typically presented in tables. This type of data contains a structure to be retrieved, such as a tag or a column. On the other hand, the unstructured data is not structured via pre-defined data models or schema, for example, text files, e-mail and social media like Facebook or YouTube. The semi-structured data has internal tags that identify separate data elements such as Extensible Markup Language (XML), JavaScript Object Notation (JSON) and HyperText Markup Language (HTML) [12]. This type of data are normally stored in Not Only SQL (NoSQL) databases and Hadoop Distributed File System (HDFS).

### Comparison between streaming processing and batch processing

There are two big data paradigms according to processing time requirements, these paradigms are present in 2.2:

- **Streaming Processing**: In this paradigm, the data must be processed as soon as possible, that means, data needs to be processed in real-time, as it comes and quickly detect conditions within a small time period from the point of receiving the data to deliver results. As the name implies, data arrives in a stream and, commonly, the intervals of data arrival are at the level of second or millisecond, for example, in an online application [12]. Stream processing is appropriate for tasks like fraud detection: while transaction data is processed, it is possible to detect anomalies that signal fraud in real-time and then stop fraudulent transactions before they are finalized [13].
- **Batch Processing**: In this paradigm, the processing happens in blocks of data that were already stored before. Data arrives over longer periods of time, such as daily or weekly, and can contains millions of records [12].

Big data platforms can use these paradigms, but each one has differences that will cause architectural distinctions [12].

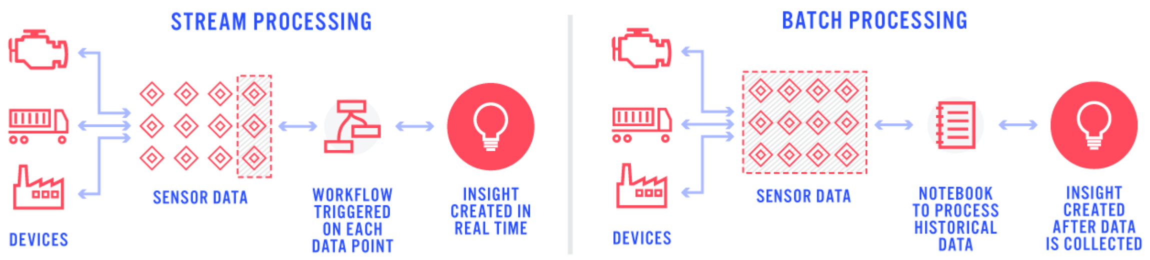

STREAM PROCESSING
DEVICES
SENSOR DATA
WORKFLOW TRIGGERED ON EACH DATA POINT
INSIGHT CREATED IN REAL TIME
BATCH PROCESSING
DEVICES
SENSOR DATA
NOTEBOOK TO PROCESS HISTORICAL DATA
INSIGHT CREATED AFTER DATA IS COLLECTED

**Figure 2.2:** Comparison between streaming processing and batch processing [14]

### 2.1.1 Internet of Things and Smart Cities Data

The concept of the Internet of Things refers to the interconnection of quotidian objects that can be a fitness tracker, a thermostat, a lock, among others, and can see, hear, think, make decisions and also exchange information with each other [15]. This paradigm benefits greatly from the evolution of wireless telecommunication networks [16], considering that IoT is the combination of wired and wireless communications technologies, or in other words, they are physical objects connected to the internet. The term Internet of Things was first mentioned in 1999 by Kevin Ashton [17] and, according to the Cisco Internet Business Solutions Group (IBSG), the IoT arose when the number of interconnected devices exceeds the number of people on our planet, i.e., between 2008 and 2009 [18].

The Smart City term emerged in 1997 due to the Kyoto Protocol and aimed to solve urban problems such as congestion, lack of school places, pollution and poor public services, and has been attracting some attention regarding urban development initiatives. According to Caragliu *et al.*, a city can be designated as smart when investing in human and social capital, as well as in transport and Information and Communications Technologies, fosters sustainable economic growth and a comfortable quality of life by intelligently managing natural resources through a participatory government [19].

The emergence of Smart Cities, Industry 4.0 and other areas of IoT applications has resulted in the generation of a large volume of data [15]. The concept of smart cities has played an important role in academia and in the industry, since it can improve services such as traffic management, water management, and energy consumption, as well as improving the quality of life for the citizens in a general form [20].

According to Giffinger *et al.* a smart city can be classified into six aspects: environment, economy, governance, living, mobility, and people [21]. These are the main points for solutions to urban development and management of these topics that will lead to a smarter city. By implanting sensors across city infrastructures and get data from existing systems, including mobile devices of citizens, it may be applied big data analysis supported by Machine Learning (ML) techniques to monitor and anticipate urban phenomena, discover patterns over time, changes in behaviors, extract insights, and event prediction in heterogeneous environments.

Debajyoti Pal *et al.* proposed a conceptual big data framework for smart cities in [22].

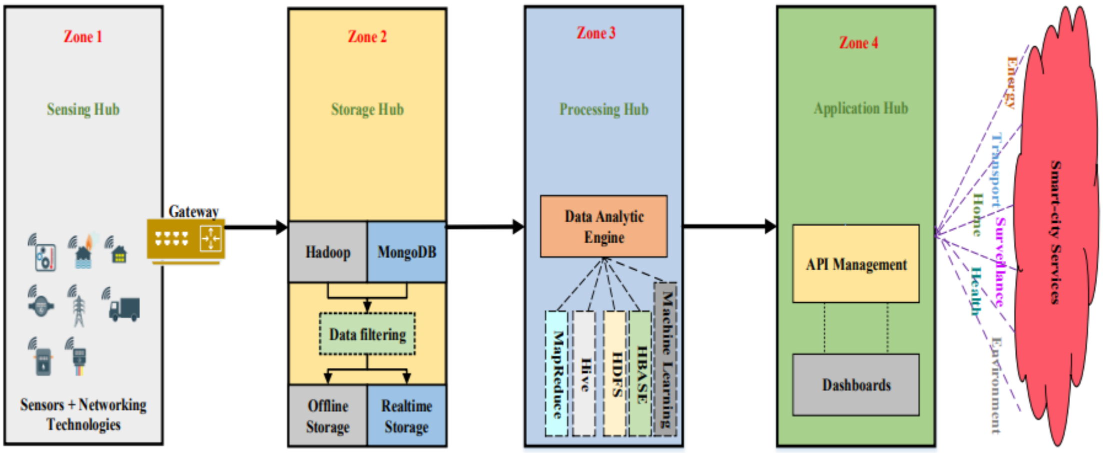


**Figure 2.3:** Big data framework for smart cities [22]

The entire framework has been divided into four distinct zones as can be seen in Figure 2.3. They are: Zone 1 (Sensing Hub), Zone 2 (Storage Hub), Zone 3 (Processing Hub), and Zone 4 (Application Hub) and all of them are interlinked, that means, the the output from one serves as an input to the next.

### Zone 1 (Sensing Hub)

This zone is a physical layer composed of sensors and objects interconnected by a variety of network technologies. These sensors generate data of interest and communication that can take place either via wired or wireless (Radio-Frequency Identification (RFID), WiFi, Zigbee, Bluetooth, etc.) network.

### Zone 2 (Storage Hub)

This zone is responsible for storing the raw naïve data that is generated by Zone 1. The main concern in this area is not to lose information, since it means loss of value that the data may bring. The authors stated that it is desirable to use some cheap massive data storage platform like the Hadoop based systems. However, a smart city has varying requirements, some data collected by Zone 1 may be time-critical and require real-time processing. Therefore, platforms like MongoDB that have real-time processing capabilities may be a good option and they include it in Zone 2. Filtering techniques should also be used, as raw data contain a lot of noise.

### Zone 3 (Processing Hub)

This zone is responsible for processing and analyzing the data. When using a Hadoop based system, the storage requirements are fulfilled by the HDFS, whereas processing is done by the MapReduce algorithm, and this ensures data scalability. If real time processing is required, HBASE [1], an open-source, distributed, versioned, non-relational database, can be used which speeds up the data look-up rate. For querying, and managing the overall functionality, Hive [2] can be used. Regardless of the platform used, the main function of this zone is to provide the required decisions.

### Zone 4 (Application Hub)

This zone is an interface between the processing hub and the current users of the various smart-city services. The main goal is the API management and providing suitable dashboards to the users depending upon the application context. The decisions that are generated in the previous zone (Processing Hub) are extremely diverse in nature, and hence, categorized into suitable themes in this phase, and finally, they are transferred to the appropriate channel.

Another approach about big data framework is described in Figure 2.4. This framework is generic, that means, it can be applied for any smart city scenario, because it was slipped in two different zones that allows this generality: City Learning Model and City Runtime [23].

[1] https://hbase.apache.org/

[2] https://hive.apache.org/

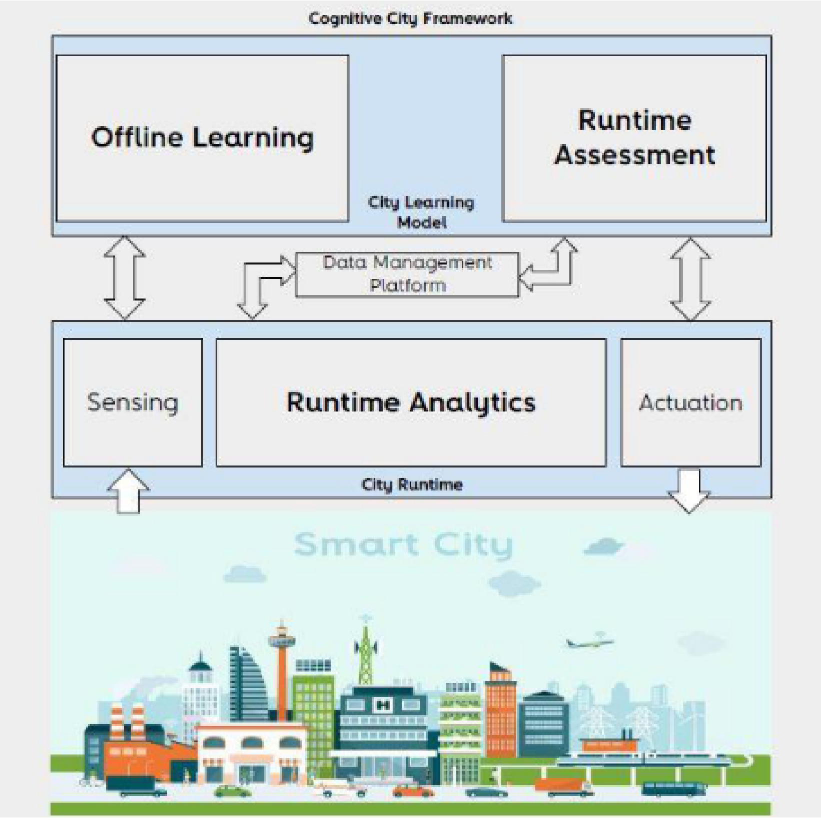


**Figure 2.4:** Generic Cognitive City Framework Architecture [23]

### City Learning Model

In this zone it is built the model to be applied in the cities in an offline learning entity, which encompasses the pre-processing data, algorithm training, evaluation and selection of the best model. It is also made the evaluation of the model, taking into account new samples that are received among the time in runtime assessment entity. It is compared the forecast made by with real result, in other words, verifies whether forecast predicted the correct result or not. In case the predictions results are worse, it is essential to rebuild the model to get better results.

### City Runtime

The aim of this zone is to act and manage the infrastructure. The model developed in offline mode is used to manage occurrences in the city. This City Runtime zone encompasses the following components:

- **Sensing**: this entity collects raw data by sensors distributed throughout the city. It can be collected from several assets such as noise, traffic, temperature among others;
- **Runtime analytics**: according to authors, this entity is the "brain" of the run time system since it runs the machine learning model deployed in the city system, allowing to get the results;
- **Actuation**: according to the results obtained from the runtime analytics entity a specific recommendation is set in order to update the city infrastructure status. This entity sends commands towards the infrastructure in order to adapt it to the instantaneous city needs.

Once this platform is constantly receiving data from the city, it is required to be included a data management platform to store and share the city data. Beyond that, this platform makes data mediation between different system entities.

### 2.1.2 Big data Opportunities for Society and Companies

A study about the sense and mean of data should be done by a company or organization to understand how can it take advantage and benefit from it.

According to McKinsey & Company, big data can create value, as concluded in a survey of the following topics [24] [8]:

- **Healthcare**: they have researched the U.S. healthcare, and it was concluded that big data has great potential for creating a clinical decision support systems, analyze disease patterns and improve public health. They affirm that if big data could be creatively and effectively utilized, it is possible to reduce the expenditure for the U.S. healthcare by over 8%;
- **Public sector**: based on the European Union public sector administration, they stated that big data can bring value and create transparency by accessible related data, discover needs, improve performance, customize actions for suitable products and

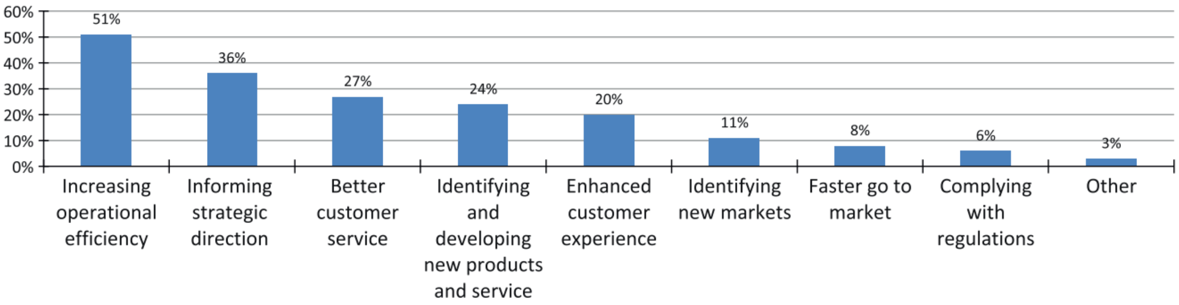


**Figure 2.5:** Big Data Opportunities for Enterprises [25]

services, decision making with automated systems to decrease risks, innovating new products and services. Big data can be used also to improve the efficiency of government operations, since it can detect fraud and errors;

- **Retail**: based on U.S. retail, they analyse the variety of price, product placement in stores, distribution and logistics optimization, web based markets and, with these indicators, retailers can improve their profit by more than 60%;
- **Manufacturing**: through big data analysis it can be improved demand forecasting, supply chain planning, sales support and developed production operations;
- **Personal location data**: advertising can be improved according to the user's geographic location, as well as geo-location data can be used for urban planning.

In fact, the most successful companies are those that support their decisions in knowledge generated by data. For example, Netflix analyzes the viewing habits of its users to recommend content that will appeal to the user. In addition, it also uses this information to acquire the rights to new films and series which it anticipates will have a good public acceptance, and thus avoids bad investments in content acquisition[3]. Another interesting example is Starbucks, which has an app that is used by customers to place their orders. This application aimed to improve the service provided; it also collects information about their customers' buying habits and preferences. Thus, when customers arrive at the counter baristas already know what are their preferences. Moreover, the company uses this information to create more relevant marketing campaigns, as well as for deciding locations to open new stores, and also deciding future menu updates[4].

In [25], once again, the idea has been reinforced that taking advantage of valuable knowledge beyond big data will become the basic competition for today's companies, and create new competitors capable of attracting employees with critical big data skills. In Figure 2.5 it is illustrated the business sectors that can take advantage through harnessing Big Data, as can be seen, above 50% of 560 enterprises think Big Data will help them in increasing operational efficient.

[3] https://www.icas.com/thought-leadership/technology/10-companies-using-big-data/

[4] https://www.kolabtree.com/blog/5-companies-using-big-data-and-ai-to-improve-performance/

## 2.2 Related Work on Machine Learning

Machine Learning is a field of Artificial Intelligence (AI) giving the ability for the machine to learn without explicit programming and aimed to solving more complex problems. Pattern recognition and Computational Learning Theory was the basis for the emergence of machine learning [20].

The term Machine Learning was first used in 1959 by Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence [26]. Later, in 1997, Tom M. Mitchell, a researcher who contributed to the advancement of machine learning, artificial intelligence and cognitive neuroscience, gave a more formal and widely cited definition of the algorithms studied in the machine learning field: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."* [27].

Processing the data generated by a smart city is a major challenge from the perspective of analytics and machine learning. Only a small fraction of collected data is used to improve citizens' lives. The causes for this come from security and privacy issues to legacy systems that do not have Application Programming Interfaces (APIs), costs associated with information transfer, and other problems. Hence, there is the need to use machine learning to explore unlabeled and labeled data in the context of smart cities [28].

With the increase of data collected for processing and analysis, there comes the need for data mining tools. As already mentioned, the data is not only traditional discrete data, but also data streams generated by sensors from industrial equipment, automobiles, among others. These data streams can be data about location, motion, temperature and even chemical changes in the air [29].

### Data Science vs. Data Mining vs. Machine Learning

With the growth of the generated data, new terms associated with processing and handling data are coming up, such as, data science, data mining and machine learning. There is no standard definition for each one of these terms. Even in the scientific community, there is no exact definition, and the definitions given by many authors are somewhat subjective. Basically, the main objective of these fields is to get knowledge from data that can be useful for eventual use. However, there are certain aspects that relate more to one field than another.

Data science is a field that includes everything that is associated with the collection, cleaning, preparation, modeling and final analysis of data. Data science combines programming, logical reasoning, mathematics and statistics. An analogy can be made with an umbrella that contains several techniques that are used for extracting the information and the insights of data [30].

Data mining is the process of garnering incomprehensible and unknown data, and then using that data to make relevant business decisions, that is, it is the process of knowledge discovery for distinguishing the relationships and patterns that were previously unknown [30].

Machine learning follows the relatively same process of data mining. It follows the method of data analysis which is responsible for the creation of the model. Moreover, to recognize patterns in the data, it uses algorithms that iteratively gain knowledge from data and then forecasts trends [30].

### 2.2.1 Key Machine Learning Terminology and Concepts

Before going into more complex machine learning topics, it is particularly important to understand the terminology used in this area. Otherwise, it will be more difficult to have a full understanding of the subjects that will be mentioned throughout this work.

#### Labels

A label is the thing that wants to be predicted — the $y$ variable in simple linear regression. The label could be the future price of a car, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.

#### Features

A feature is an input variable — the $x$ variable in simple linear regression. A machine learning project can use just one feature or can use millions of them, specified as:

$$x_1, x_2, \ldots x_N$$

To evaluate the price of a house, the features can be, for example, its location, whether or not it has a swimming pool, the number of rooms, among other typical characteristics of a house.

#### Models

A model defines the relationship between features and label. For this, the algorithms find patterns in data, then a model recognizes those patterns to make predictions on new data, as shown in the Figure 2.6. For example, a house price prediction model might associate certain features strongly than others. A model has two phases in its life-cycle:

- **Training**: In this phase, the model is created and learnt with training data. In the case of supervised learning, it is given to the model labeled examples, which enables the model to gradually learn the relationships between features and label [28].
- **Inference**: In this phase, it is given the test (unlabeled data) to the model and it is used the trained model to make the predictions ($y'$) [28].

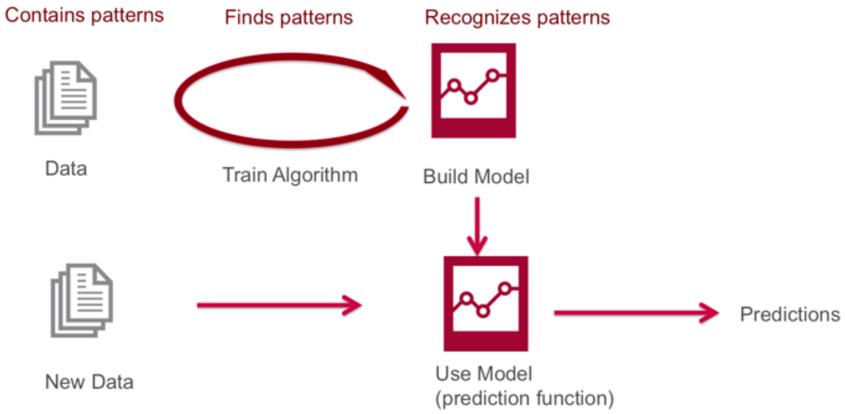


**Figure 2.6:** Building the model by finding patterns in data [31]

### Train/Test Split

When splitting data into training and testing, it is necessary to take into account that test set is large enough to yield statistically meaningful results and is representative of the data set as a whole, that means, the test set must have the same characteristics as the training set. It is recommended to never train with test data, since this data must be unknown to the model not to influence the results of the inference.

### Cross Validation

In this case, the data is also split in train and test, but it is applied to more subsets. It aims to estimate how good this model is in practice, that is, its performance for a new set of data. So, the data is split into $k$ subsets, $k-1$ of these subsets are used for the train, and the last subset is to hold for the test. In Figure 2.7 is present a visual representation of cross validation with 5 folds.

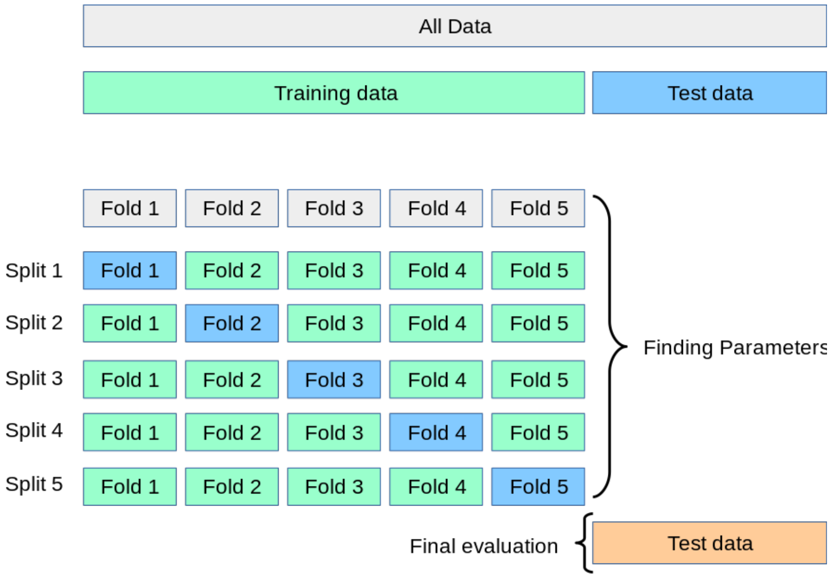


**Figure 2.7:** Visual representation of cross validation with 5 folds [32]

### Overfitting and Underfitting

Overfitting means that the model has trained "too well" and is now fit too closely to the training dataset, which means, the model fits very well with the previously observed dataset, but it is ineffective to predict new results. This usually happens when the model is too complex. For example, the model prefabs too many features/variables compared to the number of observations. When this happens, the model learns or describes the "noise" in the training data instead of the actual relationships between variables in the data [32].

On the other hand, when a model is underfitted, it means that the model does not fit the training data, and therefore misses the trends in the data. Usually, this is the result of a very simple model that does not have enough predictors/independent variables. Therefore, this model will have poor predictive ability [32].

Figure 2.8 shows graphically these concepts.

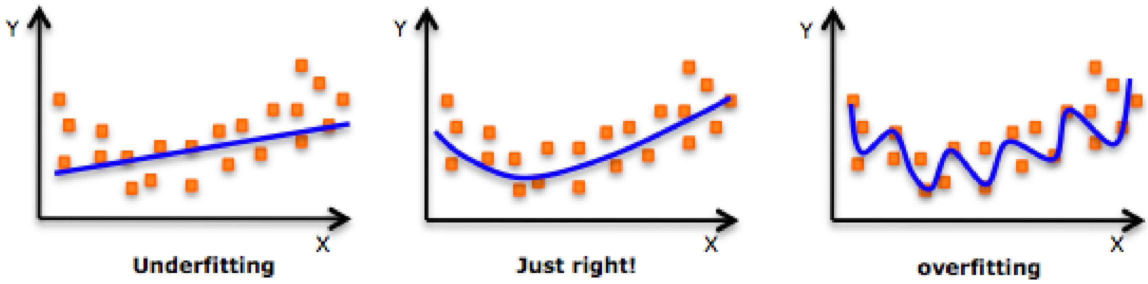


**Figure 2.8:** An example of overfitting, underfitting and a model that's "just right!" [32]

### Confusion Matrix

Confusion Matrix allows to measure performance of machine learning classification problem where output can be two or more classes. It is a table with different combinations of predicted and actual values, as can be seen in Table 2.3 [33].

**Table 2.3:** Confusion Matrix

| | **Actually Positive (1)** | **Actually Negative (0)** |
|---|---|---|
| **Predicted Positive (1)** | *True Positives (TPs)* | *Falses Positives (FPs)* |
| **Predicted Negative (0)** | *True Negatives (TNs)* | *Falses Negatives (FNs)* |

- **True Positive (TP)**: it was predicted positive and it is true;
- **True Negative (TN)**: it was predicted negative and it is true;
- **False Positive (FP)**: it was predicted positive and it is false;
- **False Negative (FN)**: it was predicted negative and it is false.

There are several metrics that can be deduced from the confusion matrix such as precision, recall, accuracy and F1 score [34].

### Correlation coefficient

The correlation coefficient measures the strength of association between two variables. The value of a correlation coefficient will range between -1 and 1. How much larger is the absolute value of the correlation coefficient, stronger is the relationship between the variables. The strongest relationships are indicated by coefficient values of -1 or 1; otherwise, the weaker relationships are indicated by a value of 0. If the correlation is positive, it means that, as one variable becomes larger, the other variable tends to become larger too. On the other hand, a negative correlation means that, if one of the variables grows larger, the other usually gets smaller. Strong correlations on the scatter plots are indicated by the data points plotted just as a straight line whether positive or negative, as can be seen in Figure 2.9. The more random the data points, the weaker the correlations between the variables [35].

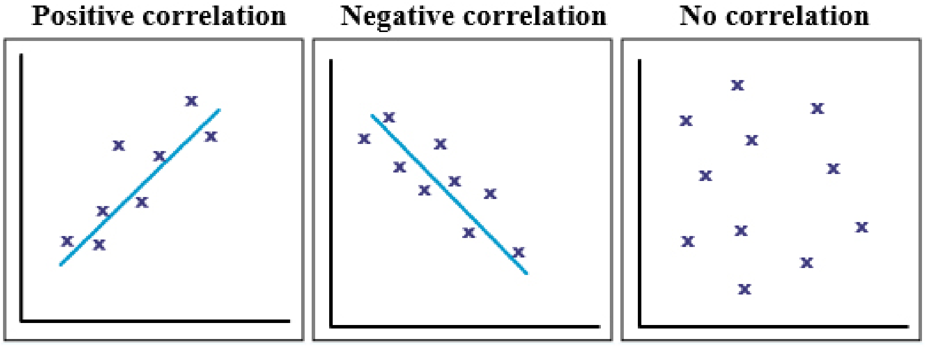


**Figure 2.9:** Correlation coefficient

## 2.2.2 Categories of Machine Learning algorithms

Selecting the right algorithm or combination of algorithms for the job is a constant challenge for who works in this field, and it depends of the situations and type of data that is dealing, which means, there is no perfect algorithm that is good for all cases.

Before examining specific algorithms it is important to understand the four categories where they are split.

### Supervised Learning

It is particularly useful when the labels for a given dataset are known. All inputs and outputs of historic are known, but need to be predicted for other new instances. Therefore, the purpose is to find a general algorithm that maps the inputs to the outputs. In this process, the goal is to identify patterns [28]. A common application is the security sector, with the use of tools to identify suspicious behavior on a network and ensure protection against attacks. This branch of machine learning works by feeding the machine sample data with various features (represented as $X$), and the correct value output of the data, the target, (represented as $y$) [36].

For example, in a prediction of a car price, the algorithm can formulate predictions by analyzing the relationship between car attributes (year of make, car brand, mileage, etc.) and the selling price of other cars sold before based on historical data. Thus, seeing that the algorithm knows the final price of the other cars sold, it can found relationships between the cars' characteristics and its price.

Subsequently, after the machine knows the rules and patterns of the data, it creates a model and, once prepared, it can be applied to new data and tested. After the model has passed both the training and test data stages, it is ready to be applied and used in the real world. Figure 2.10 shows the process in a supervised learning prediction.

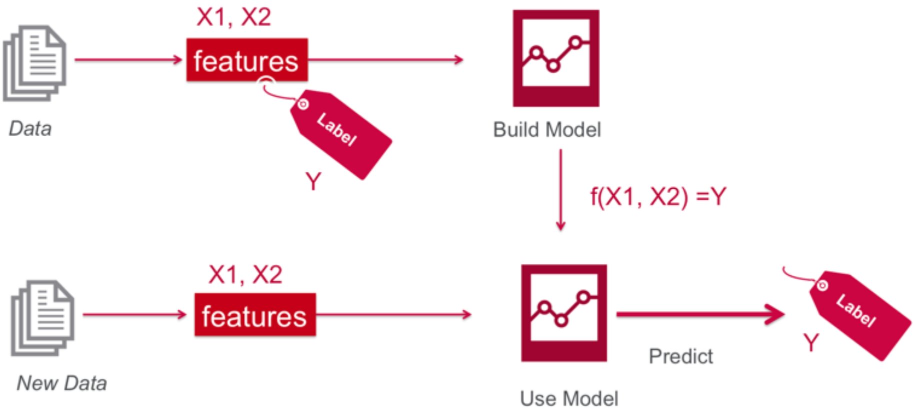


**Figure 2.10:** Supervised learning prediction [31]

Within this category, the algorithms can be classification or regression. The main difference between them is that the output variable in regression is numerical (or continuous), whereas for classification it is categorical (or discrete).

Regression prediction problems are usually quantities or sizes. For example, when it is provided a dataset about real estate market, and to predict its prices, it is used a regression task because the price will be a continuous output [37].

On the other hand, classification algorithms attempt to estimate discrete or categorical output, for example, when it is provided a dataset about real estate market, a classification algorithm can try to predict whether the prices for the houses "sell more or less than the recommended retail price" [37].

### Unsupervised Learning

In this case, the challenge is to find implicit relationships in an unlabeled data set because there are no target variables [28]. That is, unsupervised learning requires the system to develop its own conclusions from a given dataset. For this, the machine must uncover hidden patterns and create labels through the use of unsupervised learning algorithms. It is particularly useful for problems with no expected results that should appear. As can be seen in Figure 2.11, an example of unsupervised learning is grouping similar customers based on purchase data.

In this case, the algorithm will try to find relations between customers' characteristics and, based on these relations, groups similar customers.

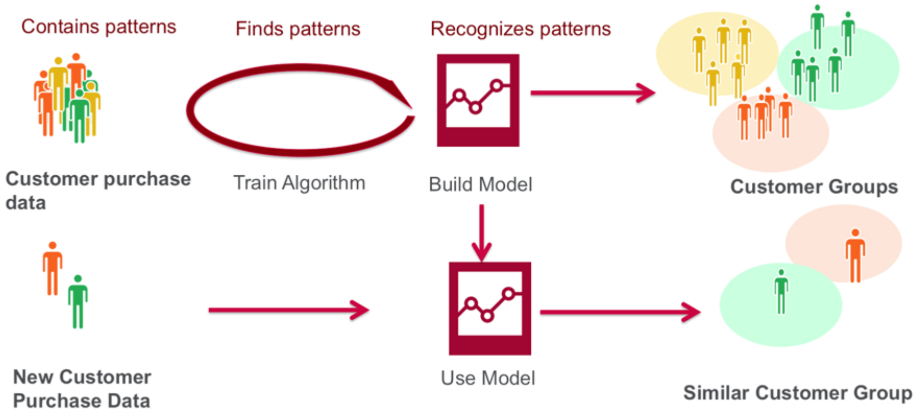


**Figure 2.11:** Unsupervised learning prediction [31]

This type of algorithms can be applied to recommendation systems where, based on collaborative or specific information, software can filter out optimal content.

As in supervised learning, the algorithms can be classification or regression; in unsupervised learning algorithms can also be clustering or dimensionality reduction.

The clustering algorithms attempt to estimate a discrete or categorical output. In this sense, the process of these algorithms is grouping similar entities together. Clustering is important to find intrinsic groups among the unlabeled data present.

Dimensionality reduction algorithms have a different role. They are particularly useful when the number of features is high which makes difficult the visualization of the training set. Moreover, some of these features are correlated, and hence are redundant. So, dimensionality reduction is the process of reducing the number of random features under consideration, by obtaining a set of principal features.

### Semi-Supervised Learning

Supervised Learning algorithms have as a mainly disadvantaged the need of hand-labeled either by a Machine Learning Engineer or a Data Scientist. When the dataset is very large, the cost of this process is too expensive; nevertheless, the application spectrum of unsupervised learning is limited. Regarding to these constraints, semi-supervised learning is a combination of supervised and unsupervised learning: the algorithms are trained upon a combination of labeled and unlabeled data, and are very useful when the acquisition of unlabeled data is relatively cheap whereas labeling data is very expensive. Most of the time, it is employed few labeled data and many unlabelled data as part of the training set. These algorithms try to explore the structural information contained in the unlabeled data, in order to generate predictive models that work better than models that only use labeled data [38].

Figure 2.12 is a outline of the different machine learning approaches demonstrated before related to the data type used (labeled/unlabeled).

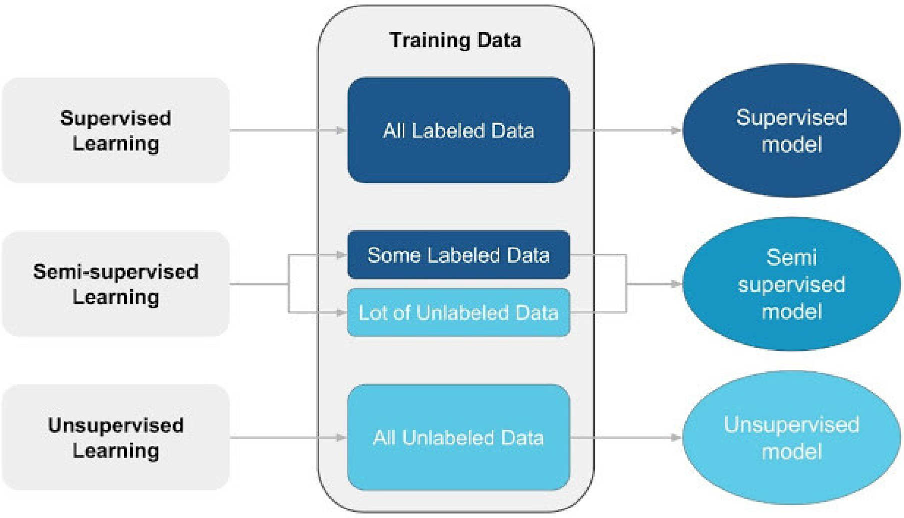


**Figure 2.12:** Outline of the different machine learning approaches [38]

### Reinforcement Learning

In this category, the system is encouraged to learn from trial and error, optimizing the process in direct practice [28]. Application examples are autonomous vehicles, game AI and robot navigation that learns according to input received from the environment where it is inserted, as shown in Figure 2.13.

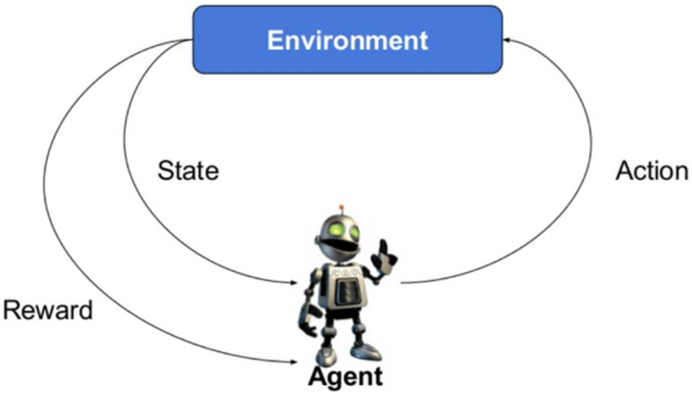


**Figure 2.13:** Typical Reinforcement Learning Scenario

## 2.2.3 Machine Learning Algorithms

There are many machine learning algorithms, however, some are variations of others, and the choice of an algorithm is never linear, and it is very dependent on the application scenario.

A diagram of machine learning algorithms is illustrated in Figure 2.14, where it can be seen the hierarchy of different machine learning algorithms, including supervised, unsupervised and reinforcement learning techniques. The two major categories of supervised learning are classification and regression, which lead to discrete/qualitative and continuous/quantitative targets, respectively [39].

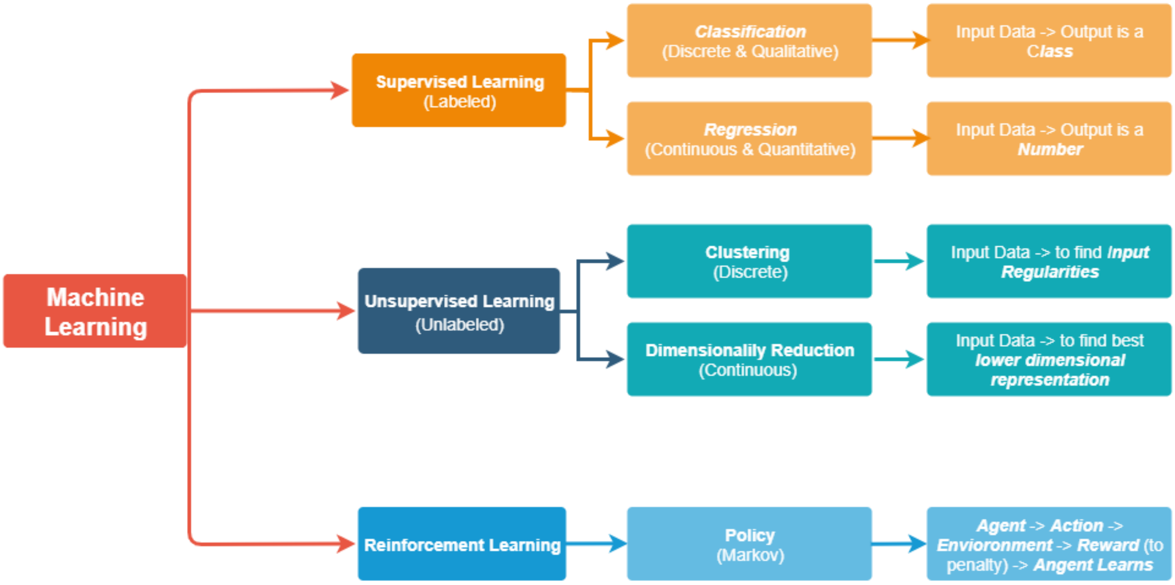

Machine Learning
Supervised Learning (Labeled)
Classification (Discrete & Qualitative)
Input Data -> Output is a Class
Regression (Continuous & Quantitative)
Input Data -> Output is a Number
Unsupervised Learning (Unlabeled)
Clustering (Discrete)
Input Data -> to find Input Regularities
Dimensionalily Reduction (Continuous)
Input Data -> to find best lower dimensional representation
Reinforcement Learning
Policy (Markov)
Agent -> Action -> Envioronment -> Reward (to penalty) -> Angent Learns

**Figure 2.14:** Overview diagram of machine learning algorithms [39]

Next, a general overview of the algorithms that will be used in this work will be given, where it will focus on the main characteristics of each one. Note that the following algorithms belong to supervisor learning, since this is the approach adopted in the present work.

### Random Forest

Random forest (RF) is a fine supervised classification method consisting of several Decision trees (DTs), which are constructed during the training process. The final prediction is an aggregation of the decisions made by trees in the forest [40]. This is an approach to solve classification and regression problems by continuously splitting data based on a certain parameter. The decisions are in the leaves, and the data is split in the nodes of trees [41].

The randomization of RF is induced by bootstrap and random feature subspace. Bootstrap is used to choose records randomly from the original datasets to build trees. If this parameter is assigned to false, the whole dataset is used to build each tree. However, some samples may appear more than once after bootstrapping. To solve this issue, random feature subspace is used to control both the randomness of the bootstrapping of the samples used when building trees, and the sampling of the features to consider when looking for the best split at each node [42].

In case of classification task, features are split with regard to their target variables purity. The entire algorithm is designed to optimize each split on maximizing purity, that is, how homogenized the groupings are. To measure the quality of a split, there are *gini* for the Gini impurity and *entropy* for the information gain. Gini impurity measures the probability of a particular variable being wrongly classified when it is randomly chosen. If all the elements belong to a single class, then it can be called pure. The degree of Gini impurity varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes, so, a

Gini Index of 0.5 denotes equally distributed elements into some classes. Information Gain is used to determine which feature gives the maximum information about a class. It is based on the concept of entropy, which is the degree of uncertainty [43].

Figure 2.15 presents a scheme of random forest with three trees. Each decision tree in the forest considers a random subset of features, and only has access to a random set of the training data. This enhances diversity in the forest, leading to more robust overall predictions. To make a prediction, the random forest will take a majority vote for the predicted class if it is leading with a classification task; for a regression task the random forest takes an average of all the individual decision tree estimates [44].

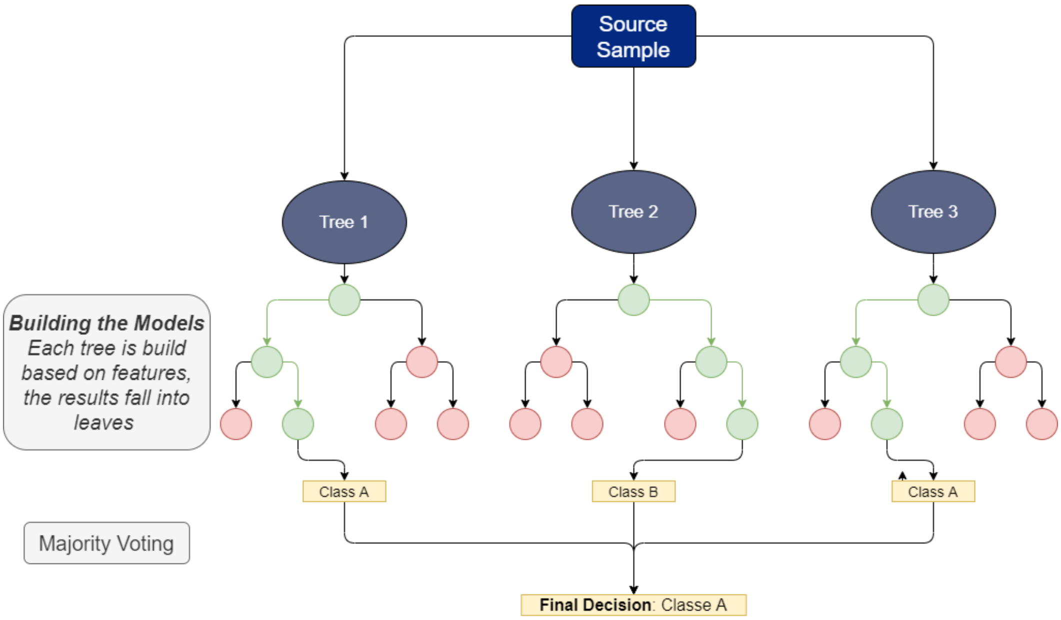


**Figure 2.15:** Scheme of random forest operation

### Neural Networks

Artificial Neural Networks are computational models inspired by the human brain that are capable of machine learning and can be used for both classification and regression problems[5].

A Neural Network (NN) is composed by layers of neurons. In essence, the neuron receives inputs, multiplies them by some weights, and then passes them into an activation function such as *identity*, *logistic*, *tanh* or *relu*.

There are three layers of a neural network: input, hidden, and output layers. The input layer receives the data, whereas the output layer returns the prediction. The layers in between are known as hidden layers, where the intermediate computation takes place. When a neural network has many hidden layers it is also know as deep learning. A multi-layer neuron is sensitive to feature scaling, then it is highly recommended the scaling of data.

[5] https://www.pluralsight.com/guides/machine-learning-neural-networks-scikit-learn

Figure 2.16 presents a neural network highlighting a neuron that receives three inputs and will compute the sum of products of inputs by their respective weight, then, it uses a nonlinear activation function and it will fire an output according to function result.

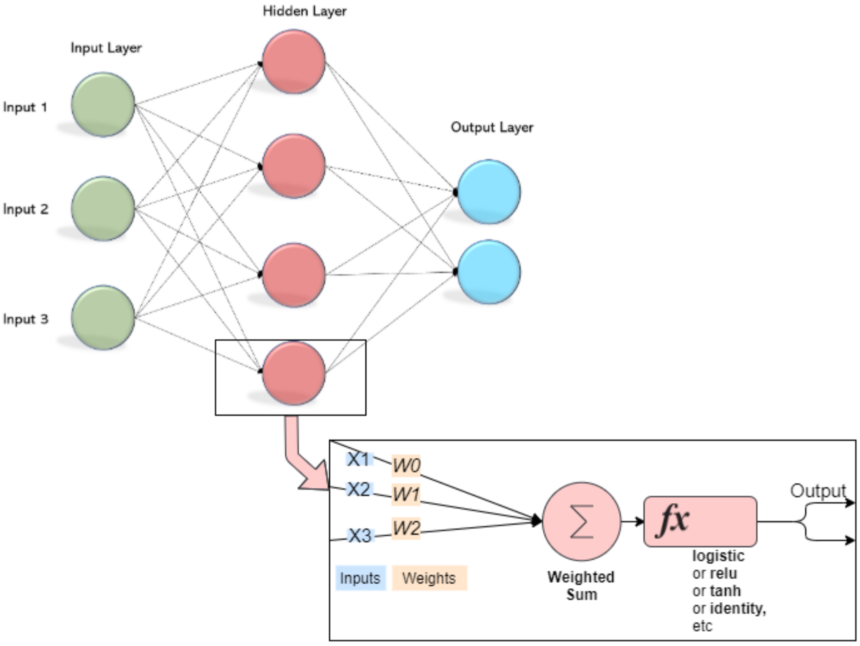


**Figure 2.16:** Artificial neural network adapted from [45]

### K-Nearest Neighbors

K-Nearest Neighbors (KNN) algorithm is a supervised machine learning algorithm that assumes that similar things exist in close proximity and can be used to solve both classification and regression problems. KNN finds the distances between a sample and all the examples in the data, selecting the specified number examples ($K$) closest to the sample. Then, it is voted for the most frequent label, in a classification task, or averages the labels, in a regression task, as in a random forest. Usually, the distance function used is the Euclidean distance. Like a neural network is sensitive to feature scaling, so it is also highly recommended the scaling of data [46].

This algorithm stores all the available cases and classifies a new instance based on a similarity measure. As can be seen in Figure 2.17, in this case in a classification task, when the algorithm receives a new instance (green circle), it will search for the $K$ nearest neighbors based on the distance. If $K$ is equal to 3, the algorithm will predict that the new instance belongs to class B (red triangle); if $K$ is equal to 5 then, it will be predicted that the new instance belongs to class A (blue square). In this simple example it can be noticed that the value of $K$ can influence the results.

There is no structured method to find the best value for $K$; however, some good practices should be taken into account such as choose an odd in order to avoid confusion between two classes of data. Smaller values for $K$ can be noisy, and larger values of $K$ can increase bias and are computationally expensive. A good starting point can be $K = sqrt(n)$, where $n$ stands for the number of samples in the training dataset [47].

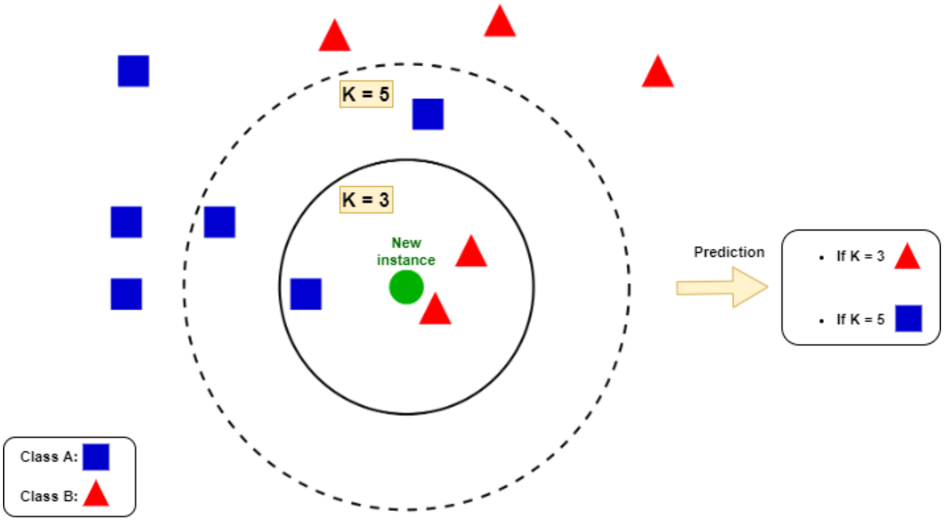


**Figure 2.17:** An example of KNN classification task with $K$=3 and $K$=5

### Logistic Regression

Logistic Regression (LR) is a generalized linear regression analysis model, which is often used in machine learning [48]. Logistic regression does not directly fit a straight line as it happens in linear regression. Instead, it is fitted a $S$ shaped curve, called Sigmoid, as can be seen in Figure 2.18. By computing the sigmoid function, it is given a probability of an observation belonging to one of the two categories. It also can noticed that Y-axis goes from 0 to 1 because the sigmoid function always takes as values between 0 and 1 (see Equation 2.1), and this fits very well for classification in two different categories (binary logistic regression). However, there is multinomial logistic regression where the target variable has three or more nominal categories, and also ordinal logistic regression where the target variable has three or more ordinal categories [49].

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

By computing the sigmoid function, it is given a probability of an observation belonging to one of the two categories. Usually, to train a Logistic Regression model (fit the $S$ shaped line to data), it is used an iterative optimization algorithm like Gradient Descent to calculate the parameters of the model, where the objective is to find a local minimum (can be the global minimum or not) of a function, where in each step the (negative) direction of the gradient is taken as is represented in Figure 2.19.

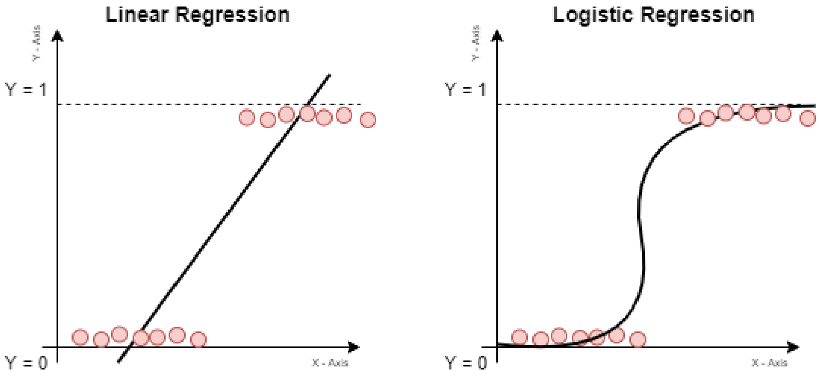


**Figure 2.18:** Linear Regression Vs. Logistic Regression

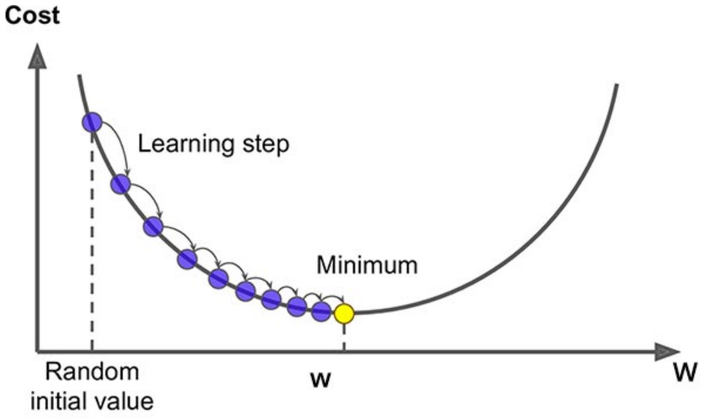


**Figure 2.19:** Gradient Descent steps to find local minimum

### 2.2.4 Main Steps of Machine Learning Workflow

As already mentioned, the main objective of a machine learning application is to get knowledge from data. In this sense, it is necessary to follow a set of steps, from the gathering of raw data to the final application. Figure 2.20 shows a complete overview of the phases in a machine learning workflow.

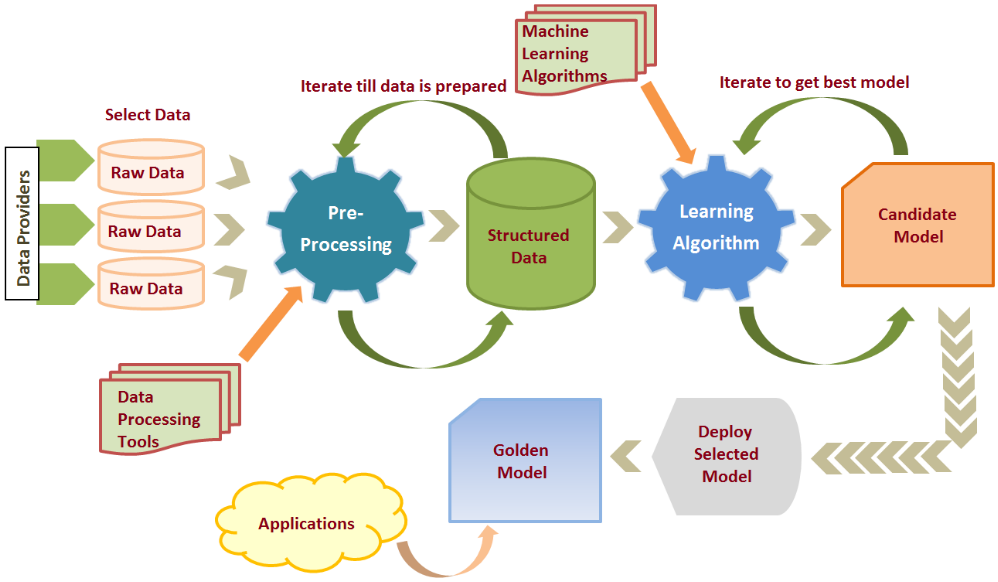


**Figure 2.20:** Machine Learning Workflow Steps [50]

### Data Acquisition

The first step is data acquisition, data can be from heterogeneous sources and is collected in a variety of ways, such as by querying a database, Hypertext Transfer Protocol (HTTP) Request to an external Application Programming Interface (API), or through the SSH File Transfer Protocol (SFTP) to extract files from partner entity servers. SFTP is a file transfer protocol that uses Secure Shell (SSH) technology to authenticate contact and establish a secure connection between machines [6]. Moreover, the data can be collected in different formats, such as Excel sheets, Comma-separated values (CSV), JSON, text files, images and so on.

At this stage, it must also be assessed whether the data collected meets the needs of the problem to be solved.´

### Pre-processing

Real-world data are often noisy, with missing values, or have a lot of other discrepancies. Data pre-processing is a technique that is used to convert the raw data into a clean dataset. The quality of the model results starts with the quality of the data and, most of the time, raw data are incomplete and can not be sent through a model since that would cause certain errors. This is one of the most time consuming parts of a machine learning project.

At this stage there are several actions that can be taken to improve the quality of the data. Removing missing values, depending on the case, it may be useful to eliminate a column or record that contains missing data or fill in the missing data with the mean / median of the attribute values. However, it is necessary to bear in mind that, when deleting data, it can have collateral consequences, which can be reflected in the loss of information [51]. The outliers in the data and their effects should also be analyzed.

[6] https://www.ssh.com/ssh/sftp

The presence of categorical variables in datates is another common issue. In these cases it is necessary to transform the nominal values into numerical ones [52].

Most of the times, the dataset contains features highly varying in magnitudes, units and range. For example, the values in the column representing the age of employees of a given firm vary between 20 and 60 years, and the values in the salary column vary between 1000 and 80000 dollars. This can cause a problem in the training of the model once the salary has a scale much greater than the age which will consequently have a much greater influence in the result, a neural network is very sensitive to not scaling data. In these cases, it is necessary to bring all features to the same level of magnitudes. This can be achieved by scaling [53].

For this, there are methods to perform feature scaling, such as standardization, which replaces the values by their Z-Scores[7], that means, redistributes the features with their mean equals 0 and standard deviation equals 1 (see Equation 2.2) [53].

$$x' = \frac{x - \mu}{\sigma} \tag{2.2}$$

Normalization can also be applied. In this case, values are re-scaled into a range of [0,1]. This might be useful in some cases where all parameters need to have the same positive scale (see Equation 2.3). Normalization helps prevent attributes with initially large ranges from outweighing attributes with initially smaller ranges [54].

$$x' = \frac{x - xMin}{xMax - xMin} \tag{2.3}$$

When the quality of the cleaned data is good enough, the next step is to get a better understanding of the patterns that are inherent in the data. This data analysis helps the choice and development of an appropriate predictive model for the target.

### Modeling

In this phase, the main goal is to identify the optimal data features for the machine learning model and create an informative machine learning model that predicts the target most accurately, able to be suitable for production [55].

As seen in section 2.2.3 there are many algorithms. Depending on the case, some may perform better than others, which implies applying several algorithms to obtain the model that has the best evaluation metrics.

### Deploy selected model

The last step of machine learning workflow is the deployment of the model to apply it in a real-world system.

If the model is producing good results with acceptable speed, then the model is able to be deployed in the real system. The model can be exposed through an open API interface,

[7] `https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/z-score/`

which allows being consumed from various applications, such as online websites, spreadsheets, dashboards, line-of-business applications and back-end applications [56].

It is also a good practice to periodically evaluate the production model to check if it continues to perform as intended.

### 2.2.5 Application Areas

Machine learning has been changing modern life. It is possible to gain knowledge of many phenomena from various fields and thereby make smarter decisions. There is a very wide range of machine learning application areas, such as video games, finances, network security, autonomous cars among many others.

In fact, machine learning is capable to detect unseen behaviors and helps the decision making with the knowledge taken by data, which can be very powerful. An example of that are the elections that made Donald Trump president of the USA. In this case, Cambridge Analytica, a British company, obtained data from about 87 million Facebook user profiles in the United States. The data would have been used to feed a system capable of drawing a psychographic profile of the American population to use in Donald Trump's campaign for the presidency. The mechanism would have made it possible to understand voters' behavioral traits in order to offer them political propaganda with a better chance of success. Advertising was distributed on Facebook in the form of sponsored ads in the feed [57].

Even though all of the machine learning applications are directly or indirectly related to citizens' life, there are some areas which have greater influence in the quotidian of citizens. Some of these areas are involved with the objectives of a smart city as can be seen in Figure 2.21.

Smart Traffic is an essential aspect in a Smart City. For example, in Nanjing City, China, sensors were installed on 1 million private cars, 7000 buses and 10,000 taxis. The data generated by these sensors were transferred daily to the Nanjing Information Center, where it was analyzed and then sent to commuters' smartphones. With this information, the government officials created new traffic routes to avoid congestion and, as a result, it was not necessary to spend money on new routes [59].

Smart Environment is another important aspect in a Smart City, as it can bring major improvements in agriculture through more reliable weather forecasts, and also by monitoring soil parameters such as moisture and minerals. People can be warned of hazardous conditions and can also provide better management of energy expenditure [60].

To save energy with street lighting, sensors can also be used next to the lamp posts that will adjust the brightness according to the presence of pedestrians, cyclists or cars [61]. This is a simple example of smart energy, and it is one of the major research areas of IoT considering that it is essential to reduce overall power consumption [62].

Another interesting example that demonstrates the potential of machine learning applied to health is a model developed and refined by DeepMind[8], a British company specializing in

[8] `https://deepmind.com/`

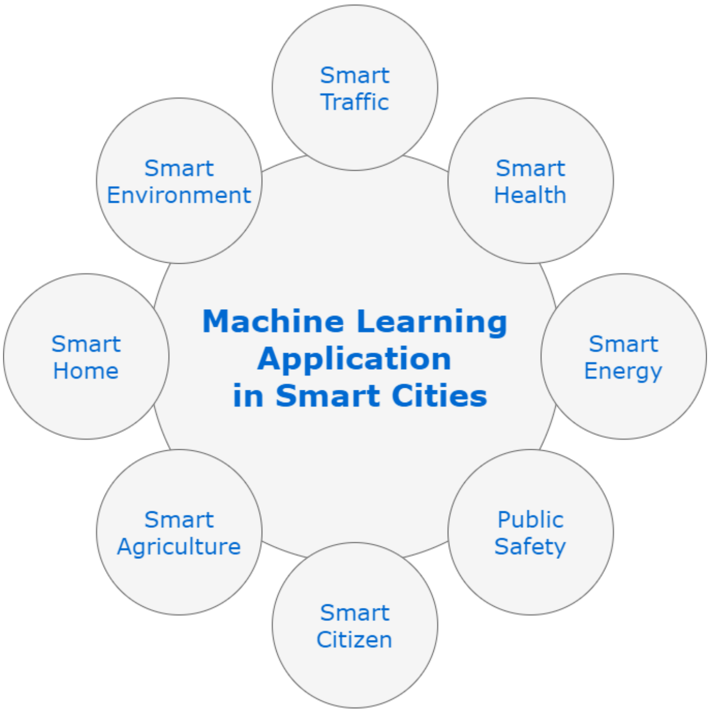


**Figure 2.21:** Some applications of machine learning in the context of a smart city [58]

AI, achieved more accurate results than human breast cancer experts. The algorithm achieved less false positives (when the mammogram looks abnormal, even without cancer, which will scare the least cancer patients who may have) and better false negative results (when cancer exists but is not detected). Early detection is considered essential to achieve better success rates in combating this disease[9]. The algorithm did not analyze previous exams or patient history, focusing only on the most recent mammogram and yet achieved more accurate results. Despite the good results, DeepMind notes that more studies and greater cooperation with health institutions are needed to be able to generalize the use of this system [63].

## 2.3 Machine Learning Applied to Crime Prevention

With the increase of the urban population, the concern about public safety is growing up. In this sense, one of the smart cities topics that governments are very focused to improve is public safety, which results in the creation of a national crime prevention program for combat of criminality and terrorism. For example, Organization for Security and Co-operation in Europe (OSCE) has guiding principles in countering terrorism where it is emphasized that terrorism is one of the most significant threats to peace, security and stability, as well as to the enjoyment of human rights and social and economic development[10].

[9] `https://www.wsj.com/articles/google-ai-beats-doctors-at-breast-cancer-detectionsometimes-11577901600`

[10] `https://www.osce.org/countering-terrorism`

### 2.3.1 Using Historical Data

One way that can help the combat of crime is to use data from previous crimes with the aim to predict and prevent future incidences. An example of this was demonstrated by McClendon *et al.* in [35], where their research showed how effective and accurate machine learning algorithms can be at predicting violent crimes.

A comparative study was conducted between the violent crime patterns from two datasets with crime historical data. The focus of the research is towards analyzing the crime patterns of the four violent crime categories, which are murders, rapes, robberies and assaults.

The authors used the following algorithms: Linear Regression, Additive Regression, and Decision Stump algorithms using the same finite set of features. In this research, they used WEKA for model development, an open-source data mining software[11].

After the implementation of the algorithms, the result outputs five metrics that evaluate the effectiveness and efficiency of the algorithms: correlation coefficient, mean absolute error, root mean squared error, relative absolute error, and the root relative squared error. They observe the linear regression algorithm to be very effective and accurate in predicting the crime data based on the training set input for the three algorithms. They also stated that the relatively poor performance of the Decision Stump algorithm could be attributed to a certain factor of randomness in the various crimes and the associated features (exhibits a low correlation coefficient among the three algorithms). The branches of the decision trees are more rigid and give accurate results only if the test set follows the pattern modeled. On the other hand, the linear regression algorithm could handle randomness in the test samples to a certain extent.

Overall, the linear regression algorithm performed the best among the three selected algorithms; then, they concluded that machine learning has become a vital part of crime detection and prevention. In addition, they highlighted other applications such as determining criminal "hot spots", creating criminal profiles, and learning crime trends.

### 2.3.2 Spatial-temporal Analysis of Crimes

Lin *et al.* in [64] proposes a data-driven method based on “broken windows” theory and spatial analysis to analyze crime data using machine learning algorithms, and thus predict emerging crime hotspots for additional police attention. Based on this theory, they design a model that predicts the incidence of drug-related crime in the following month based on the incidence of drug-related crime, fraud, assault, intimidation, auto theft, and burglary in the current month. In this way, it is possible to extend the model with its spatial-temporal characteristics. Each grid which split from the map is regarded as a sample, and accumulates samples in the same time scale to construct matrices. Each matrix represents different spatial-temporal status; then, the matrix can be used to train and test by machine learning algorithms.

[11] `https://www.cs.waikato.ac.nz/ml/weka/`

Figure 2.22 shows how the map was split into grids. They followed the next steps to build the method:

- Step 1: Split the map $\mathbf{M}$ by grids $g_n$, $\mathbf{M} = \{g_1, g_2, ..., g_n\}$
- Step 2: Each grid g has the features $f_n$, $\mathbf{g} = \{f_1, f_2, ..., f_n\}$ and calculate the value of each feature in the grid
- Step 3: Drug-related crimes which will happen in the following month are denoted as $Y$. If no drug-related crime occurs in the grid, the grid is a coldspot, and $Y = 0$. Otherwise, the grid is a hotspot, and $Y = 1$.
- Step 4: If the sum of the grid is zero, it's an empty grid that should be removed.

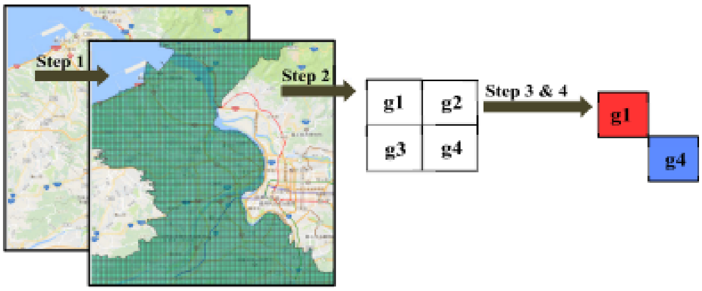


**Figure 2.22:** Spatial-temporal analysis workflow [64]

They combined 56 features with eight different types of crime and seven different spatial-temporal patterns including the current month, following month, last year, grids surrounded eight-directions in the current month, grids surrounded eight-direction last year, tendency, proportion. After calculating the value for each feature in the grids, it was set drug crime for the following month as the dependent variable. The goal is to predict crime hotspots for the following month, which are not in the same temporal environment. Empty grids are removed to prevent model performance degradation. For the different time scales, they design seven sets of accumulated data over 1, 3, 6, 9, 12, 15, and 18 months. Then, they prepared the data frames to train the models. Experiments were run using different algorithms (Deep Learning, Random Forest, and Naïve Bayes) to compare prediction results against the proposed method.

Thereby, they demonstrate a machine learning method designed to provide improved prediction of future crime hotspots, with results are validated by actual crime data. It was concluded that the model tuned using Deep Learning provides the best performance, and it is also stated that visualizations of predicted hotspots can assist patrol planning and improve crime prevention.

### 2.3.3 Using Dynamic Features

Rumi *et al.* explored how dynamic features can significantly improve crime prediction in [65]. Their motivation is based on the fact that many studies are only based on demographic data as regional characteristics and not exploring human mobility through social media.

The main challenge of their research is that dynamic information is very sparse compared to the relatively static information. To address this issue, it was developed a matrix factorization based approach to estimate the missing dynamic features across the city. The authors stated that with dynamic features, in addition to crime prediction, it is feasible to make a prediction of the category of crime, such as, Theft, Unlawful Entry, Drug Offence, Traffic Related Offence, Fraud and Assault.

In addition to historical, demographic and geographic features, authors extract the dynamic features from check-ins of Foursquare users. Foursquare is a geosocial and micro-blogging network that allows the user to indicate where they are and search for their contacts who are close to that location[12]. A location with visitors from diverse backgrounds in a time interval is highly correlated with some types of crime event such as Theft; then, the authors concluded that monitoring the fluctuation of visitor diversity at locations provides useful information to the crime event prediction.

It was used real datasets in Brisbane and New York City, and it was performed the following algorithms SVM, Random Forest, Neural Network, and Logistic Regression integrated with an ensemble based learning framework for crime event prediction. The structure of the ensemble method is demonstrated in Figure 2.23.

The first step of the framework is to divide the training space based upon the types of the features. In this case, it was divided the training space into four subsets (historical, geographic, demographic and dynamic) to provide different views in the prediction model.

The second step builds classifiers using different learning algorithms for each non-overlapping training subset.

The third step ensembles the component classifiers for each training subset separately. It was combined the output of the component classifiers for each subset. Among different types of combination technique, authors used the sum rule.

The fourth step aggregates the outcomes of each feature subsets for the final outcome. It was trained an SVM learning algorithm to aggregate the predictions made by each training subset.

The prediction performance before and after adding the proposed dynamic features have been compared. The test results demonstrate that the improvement of prediction performance after adding dynamic features is considerable and statistically significant. With this approach, the authors claim to have performance improvements in terms of precision and recall between 2% to 16%, depending on the category.

[12] https://foursquare.com/

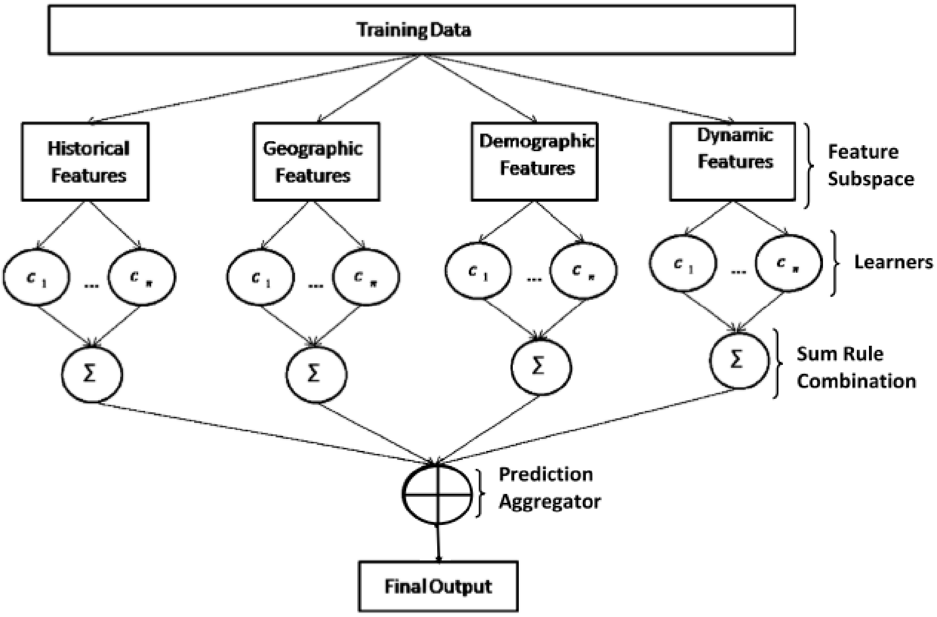


**Figure 2.23:** Ensemble model for crime event prediction model [65]

## 2.4 Summary

This chapter aimed to focus on several topics that are essential for the work that will be described in the following chapters. In this order of ideas, first, it was made an overview of big data. It can be concluded that it is very important for corporations and even to countries to retrieve most value possible from generated data, in order to increase profits, improve processes, or even people's lives.

With good data both in terms of quantity and quality, more can be taken through the application of machine learning techniques. In addition to the knowledge taken from the data, with machine learning, it is possible to make predictions for the future, since algorithms can learn and recognize patterns from data, and then it can be created a model with the capability to make forecasts.

Of the various application areas of machine learning, more emphasis was placed on public safety. A literature review was made to realize which works have already been done in this area. It was described a work where it was used data from crimes that occurred in the past and the aim is to predict future occurrences. Other works focus on spatial-temporal analysis where the aim is to split geographical data in grids and then predict hotspots for a given month. Finally, it was presented a work where it was used dynamic features. In this case, features from human mobility to make a prediction of categories of crimes.

All these models and examples will be important to develop prediction mechanisms in a safety scenario.

CHAPTER 3

# Study Scenario

This section describes the scenario chosen and the dataset used in this work. The scenario fits in the public safety of the city, more properly in the crime prediction. For that, it was used a dataset of incidents that occurred in San Francisco, which is one of the most know cities of the United States of America. In addition to being a densely populated city, San Francisco is a large financial center and popular tourist destination. Thus, this city faces many challenges in the most diverse verticals, one of which is public safety. Data of police incident reports about this city were used to build the scenario.

## 3.1 Scenario description

The scenario of the work presented in this dissertation aims to make a prediction of the crime risk given a location and a period of time. As can be seen in Figure 3.1, the prediction will be categorical and numerical.

For a categorical prediction the output will be a category. In case of binary classification, the categories are **Non-Crime** and **Crime-Crime**, whereas in multiclass classification, the outputs are **Reduced**, **Moderated** and **High**. For numerical prediction, the output will be the **number of crimes** predicted.

One of the objectives of this scenario is to compare the performance of the model in the binary and multiclass classification.

Some examples of user stories that can be applied in this scenario are: as a **tourist**, I want to know the likelihood of a crime occurring tomorrow night in Tenderloin, to find out if it is safe to go there; as a **citizen**, I want to know which are the safest neighborhoods to go walking with the family; as a **police officer**, I want to know which neighborhoods I should patrol at night because they are more prone to crimes.

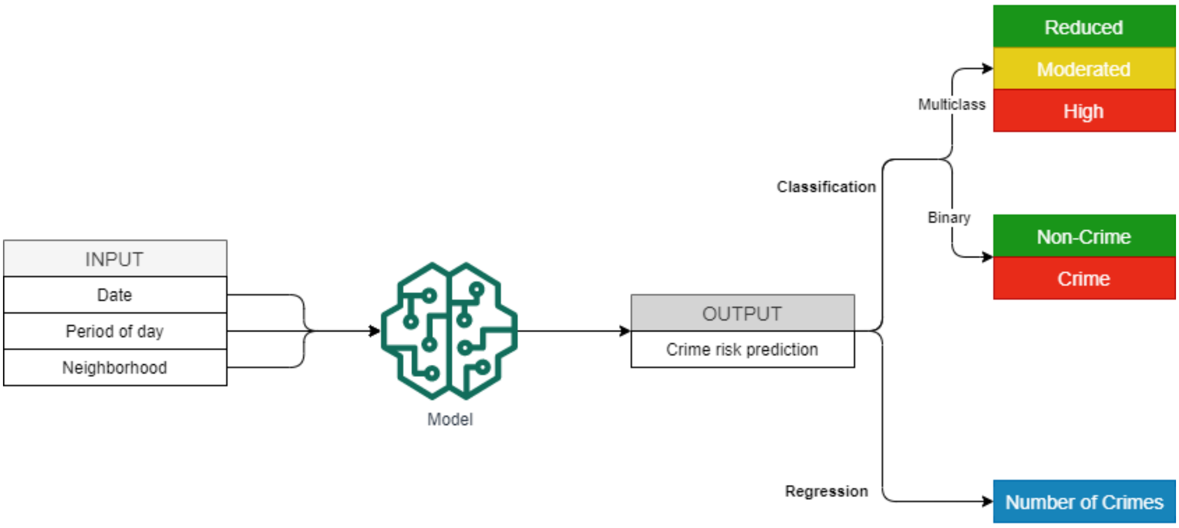


**Figure 3.1:** Crime prediction scenario

## 3.2 Dataset overview and its characteristics

The dataset used in this dissertation is from the San Francisco open data platform that contains hundreds of datasets from the city and county of San Francisco[1], The dataset contains about 317 thousand rows and 29 columns where each row is a police incident report from January 2018 to the present. These incidents can be filed by officers and by individuals through self-service online reporting for non-emergency cases.

There are some considerations to take into account on the current dataset. An incident reported must be approved by a supervising officer. Once approved and electronically signed by a Sergeant or Lieutenant, no further information can be added to the initial report. A supplemental report will be generated if necessary for additional information or clarification. This means that an individual status will not change on an initial report, but may be updated later through a supplemental report, which aims to provide additional incident information or to clarify a mistake in the initial report.

Tables 3.1, 3.2, 3.3 and 3.4 presents all columns of the dataset, as well as its description and type.

**Table 3.1:** Dataset columns description (part 1)

| Column Name | Description | Type |
|---|---|---|
| *Incident Datetime* | The date and time when the incident occurred | Datetime |
| *Incident Date* | The date when the incident occurred | Datetime |
| *Incident Time* | The time when the incident occurred | Plain text |
| *Incident Year* | The year when the incident occurred, provided as a convenience for filtering | Plain text |
| *Incident Day of Week* | The day of the week that incident occurred | Plain text |

[1] https://datasf.org/

**Table 3.2:** Dataset columns description (part 2)

| | | |
|---|---|---|
| *Report Datetime* | Distinct from Incident Datetime, Report Datetime is when the report was filed. | Datetime |
| *Row ID* | An identifier unique to the dataset | Plain text |
| *Incident ID* | This is the system generated identifier for incident reports. An incident report can have multiple incident codes associated. Thus, this identifier, while unique to the report, will be duplicated within this dataset to represent those 1 to many relationships when they exist. Incident IDs and Incident Numbers both uniquely identify reports, but Incident Numbers are what are used and referenced in the cases and report documents | Plain text |
| *Incident Number* | The number issued on the report, sometimes interchangeably referred to as the Case Number | Plain text |
| *CAD Number* | The Computer Aided Dispatch is the system used by the Department of Emergency Management (DEM) to dispatch officers and other public safety personnel. CAD Numbers are assigned by the DEM system and linked to relevant incident reports (Incident Number). Not all Incidents will have a CAD Number. Reports filed online via Coplogic ( see field: Filed Online) will not have a CAD Number and certain other reports not filed through the DEM system will also not have these numbers | Plain text |
| *Report Type Code* | A system code for report types, these have corresponding descriptions within the dataset | Plain text |
| *Report Type Description* | The description of the report type, can be one of: Initial; Initial Supplement; Vehicle Initial; Vehicle Supplement; Coplogic Initial; Coplogic Supplement | Plain text |
| *Filed Online* | Police reports can be filed online for non-emergency cases. These reports are entered via a self-service system called Coplogic [2]. This field is a boolean indicating the record was filed this way. These are also indicated in the Report Type Code and Report Type Description fields | Checkbox |
| *Incident Code* | Incident Codes are the system codes to describe a type of incident. A single incident report can have one or many incident types associated. In those cases, there are multiple rows representing a unique combination of the Incident ID and Incident Code | Plain text |
| *Incident Category* | A category mapped on to the Incident Code used in statistics and reporting. Mappings provided by the Crime Analysis Unit of the Police Department | Plain text |

**Table 3.3:** Dataset columns description (part 3)

| | | |
|---|---|---|
| *Incident Subcategory* | A subcategory mapped on to the Incident Code used in statistics and reporting. These nest inside the Category field. Mappings provided by the Crime Analysis Unit of the Police Department | Plain text |
| *Incident Description* | The description of the incident that corresponds with the Incident Code | Plain text |
| *Resolution* | The resolution of the incident at the time of the report. Can be one of: - Cite or Arrest Adult - Cite or Arrest Juvenile - Exceptional Adult - Exceptional Juvenile - Open or Active - Unfounded Note: once a report is filed, the resolution does not change on the filed report later. Updates to a case will be issued later as Supplemental reports if there is a status change | Plain text |
| *Intersection* | The 2 or more street names that intersect closest to the original incident separated by a forward slash. Note, the possible intersections will only include those that satisfy the privacy controls | Plain text |
| *CNN* | The unique identifier of the intersection for reference back to other related basemap datasets[3] | Plain text |
| *Police District* | The Police District reflecting current boundaries (boundaries changed in 2015)[4] | Plain text |
| *Analysis Neighborhood* | The Department of Public Health and the Mayor's Office of Housing and Community Development, with support from the Planning Department, created 41 neighborhoods by grouping 2010 Census tracts, using common real estate and resident definitions for the purpose of providing consistency in the analysis and reporting of socio-economic, demographic, and environmental data, and data on City-funded programs and services. They are not codified in Planning Code nor Administrative Code. This boundary is produced by assigning Census tracts to neighborhoods based on existing neighborhood definitions used by Planning and MOHCD. A qualitative assessment was made to identify the appropriate neighborhood for a given tract based on understanding of population distribution and significant landmarks. Once all tracts have been assigned a neighborhood, the tracts were dissolved to produce these boundaries[5] | Plain text |
| *Supervisor District* | There are 11 members elected to the Board of Supervisors in San Francisco, each representing a geographic district. The Board of Supervisors is the legislative body for San Francisco. The districts are numbered 1 through 11[6] | Plain text |

**Table 3.4:** Dataset columns description (part 4)

| | | |
|---|---|---|
| *Latitude* | The latitude coordinate in WGS84, spatial reference is EPSG:4326. Note, will be blank where geocoding was not possible | Number |
| *Longitude* | The longitude coordinate in WGS84, spatial reference is EPSG:4326. Note, will be blank where geocoding was not possible | Number |
| *Point* | The point geometry used for mapping features in the open data portal platform. Latitude and Longitude are provided separately as well as a convenience. Note, will be blank where geocoding was not possible | Point |
| *Current Police Districts* | ID of Police District where the incident occurred | Number |
| *Current Supervisor Districts* | ID of the District's Supervisor where the incident occurred | Number |
| *Analysis Neighborhoods* | Corresponds to Analysis Neighborhood (categorical) | Number |

## 3.3 Data understanding

Data understanding is an important step in a machine learning work, since it is here where it must be evaluated the available data and its alignment with the specific objective. Data are the fuel of machine learning, and if it is weak, then, it is often difficult to achieve the goals initially proposed. Thus, each dataset column was explored in order to understand its meaning and some trends. In this exploratory analysis charts were made to get insights that could be useful for the next steps.

The first six columns (*Incident Datetime*, *Incident Date*, *Incident Time*, *Incident Year*, *Incident Day of Week* and *Report Datetime*) are about Date and Time of incident. *Incident Datetime* is the moment when the incident occurred, while *Report Datetime* is the moment when the incident was report to the officers. Taking into account the study scenario, it was used *Incident Datetime.*

To check the distribution of incidents by months, a bar chart was made showing the number of incidents per month. As can be seen in Figure 3.2, there is no significant variance between the different months; however, it can be noted that in the winter months there is a slight decrease in incidents compared to the summer months, which may make sense, since in the summer, the city receives more tourists and the population itself spends more time away from home, which may influence these numbers.

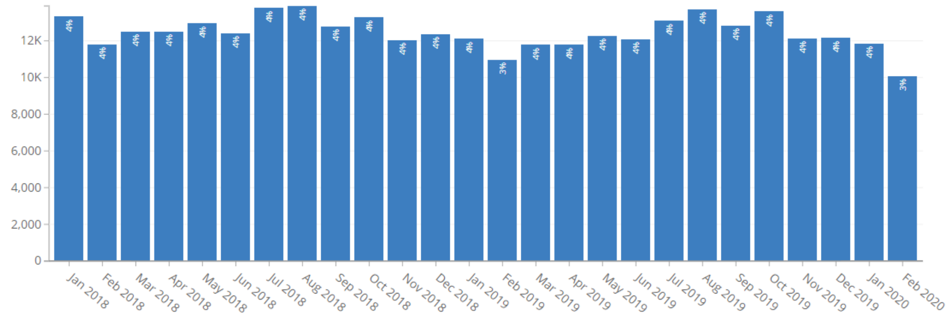


**Figure 3.2:** Number of incidents per month

In Figure 3.3, it can also be seen that there is no significant variance between days of the week, although on Friday and Wednesday there is a slightly higher frequency of incidents in comparison with the other days.

On the other hand, there is a large variance of incidents with regard to the time of its occurrence. As can be seen in Figure 3.4, there is a larger number of incidents during the solar day (from 8:00 until 21:00) compared to night (from 21:00 until 8:00).

The following columns (*Row ID*, *Incident ID*, *Incident Number*, *CAD Number* and *Incident Code*), as stated in Table 3.2, are identifiers assigned to incidents. Each incident can have one or many associated *Incident Codes*, that means, the officer can record an incident code for the warrant as well as the discovery of narcotics, for example. For this reason, *Row ID* is unique across rows, but *Incident ID* and *Incident Number* are unique to an incident and can be duplicated within the dataset.

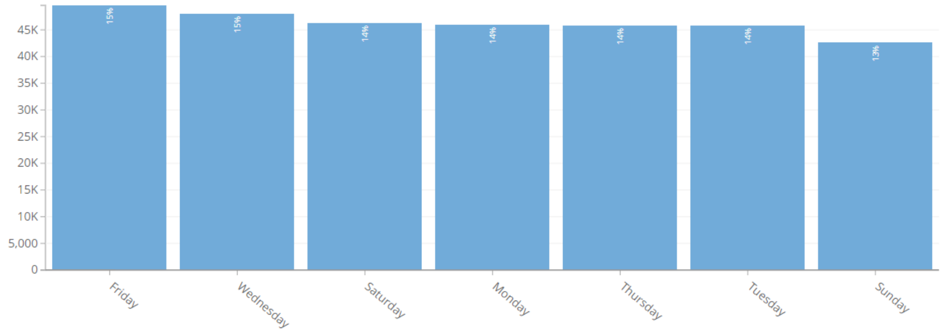


**Figure 3.3:** Number of incidents per day of the week

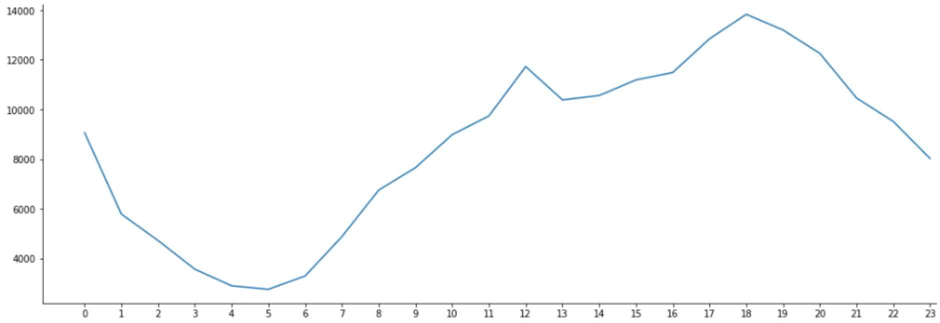


**Figure 3.4:** Number of incidents per hour

The columns *Report Type Code* and *Report Type Description* are directly related to each other. These columns indicate the report source that can come from one of three sources:

1. **Initial**: the first report filed for the incident;
2. **Vehicle**: a special incident report related to stolen and/or recovered vehicles;
3. **Coplogic**: filed online by an individual.

Each report can either be an initial one or a supplement (adding some more information to an already reported incident). Then, there are six possible values for the column *Report Type Description*:

1. **Initial**
2. **Initial Supplement**
3. **Vehicle Initial**
4. **Vehicle Supplement**
5. **Coplogic Initial**
6. **Coplogic Supplement**

These report types can be discerned through the *Report Type Description* and *Report Type Code* fields. The values of these fields are mapped between them as shown in Figure 3.5.

According to the *Filed Online* column, it can also be concluded that 78.3% of incidents were reported by officers, and only 21.7% by citizens through the online platform.

Regarding the *Incident Category*, *Incident Subcategory* and *Incident Description*, all these columns are related between them. Insofar, a category can have multiple subcategories and a subcategory can have multiple descriptions. There are 50 distinct categories, 75 distinct

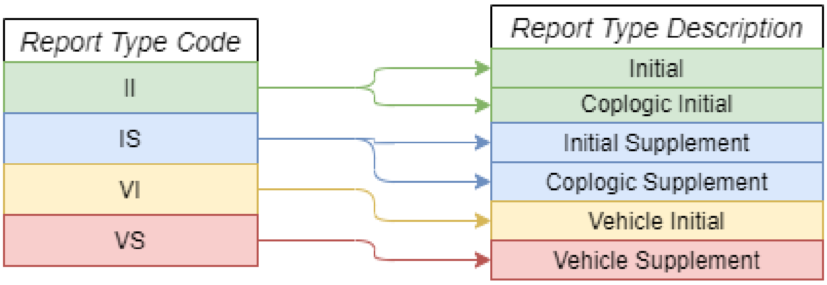


**Figure 3.5:** Mapping between *Report Type Description* and *Report Type Code* fields

subcategories and 770 distinct descriptions. As can be seen in Figure 3.6 that shows the distribution of incidents per category, the category most frequent is Larceny Theft.

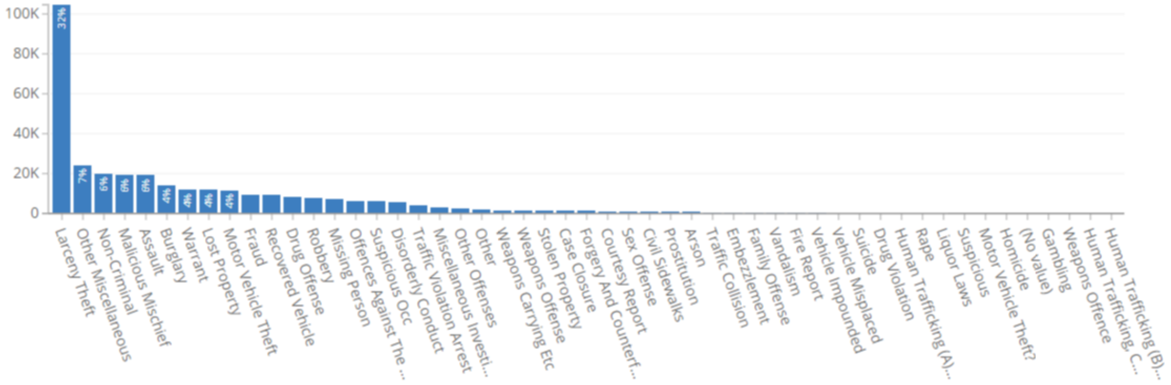


**Figure 3.6:** Number of incidents per category

The distribution of values of column *Resolution* is the following: Open or Active - 75.4%, Cite or Arrest Adult - 23.6%, Unfounded - 0.7% and Exceptional Adult - 0.3%.

There are several columns related to the location of an incident: *Intersection*, *CNN*, *Analysis Neighborhood*, *Latitude*, *Longitude*, *point*, *Analysis Neighborhoods*.

All incident locations are shown at the intersection level only. Records are masked to intersection to minimize the risk of re-identification to an individual. Centerline Node Network (CNN) field is a unique identifier for each intersection. An intersection can be between three streets (Stockton, Green and Columbus) at a point of intersection. A street can be split into segments, i.e., a segment ends where it intersects with another segment or at the physical end of a street. So, each segment sits between two nodes and both (segments and nodes) have a CNN identifier.

Since each *Intersection* has a *CNN*, and each *CNN* is represented by a *point* that groups *Latitude* and *Longitude* values, then, all these columns are directly related between them. These columns have 6379 distinct values.

San Francisco Neighborhoods are represented in Figure 3.7[7]. Both columns *Analysis Neighborhood* and *Analysis Neighborhoods* represent these columns, but *Analysis Neighborhood*

[7] https://californiatravelmedia.com/san-francisco-neighborhood-guide/

contains the name of neighborhoods and the *Analysis Neighborhoods* contains the ID. There are 41 distinct neighborhoods.

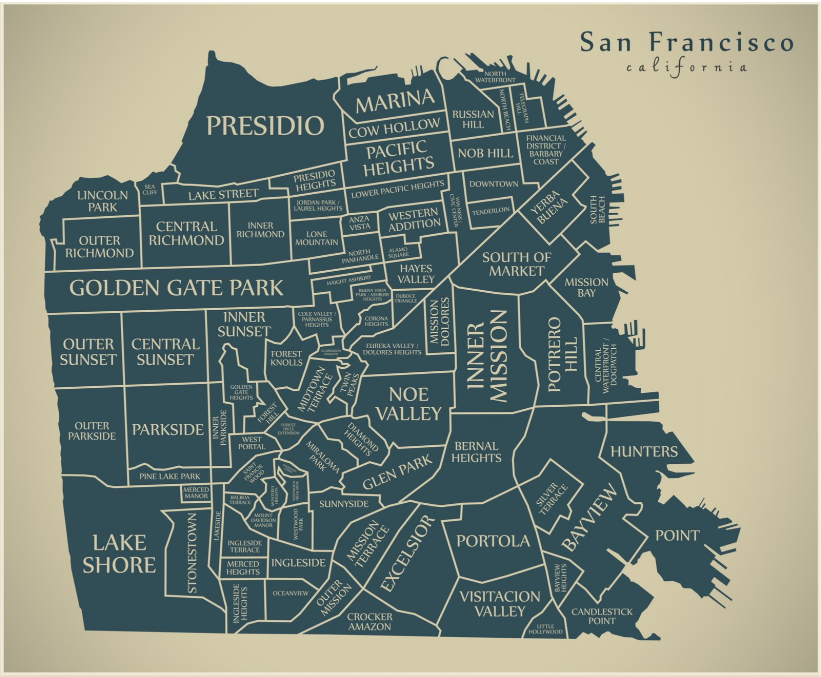


**Figure 3.7:** San Francisco Neighborhoods

Figure 3.8 shows the distribution of incidents per neighborhood. It is clear that a large part of the incidents occurred in four neighborhoods, they are: Mission, Tenderloin, Financial District/South Beach and South of Market.

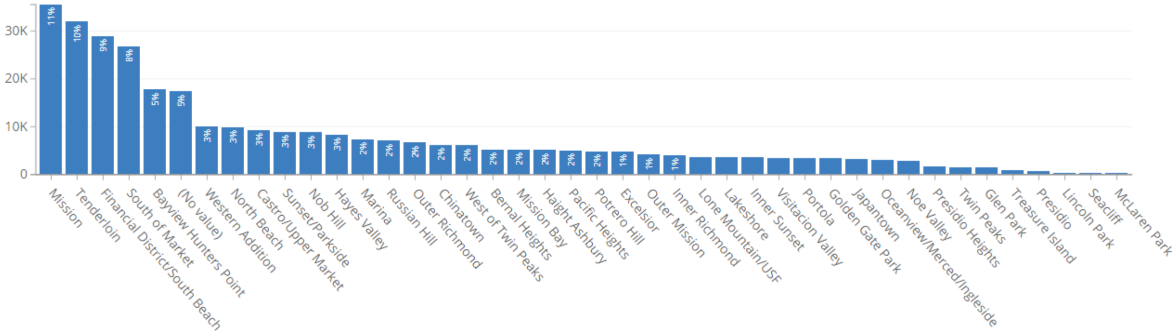


**Figure 3.8:** Number of incidents per neighborhood

*Police District*, *Current Police Districts* and *Current Supervisor Districts* have 11 distinct values. They are depicted in Figure 3.9.

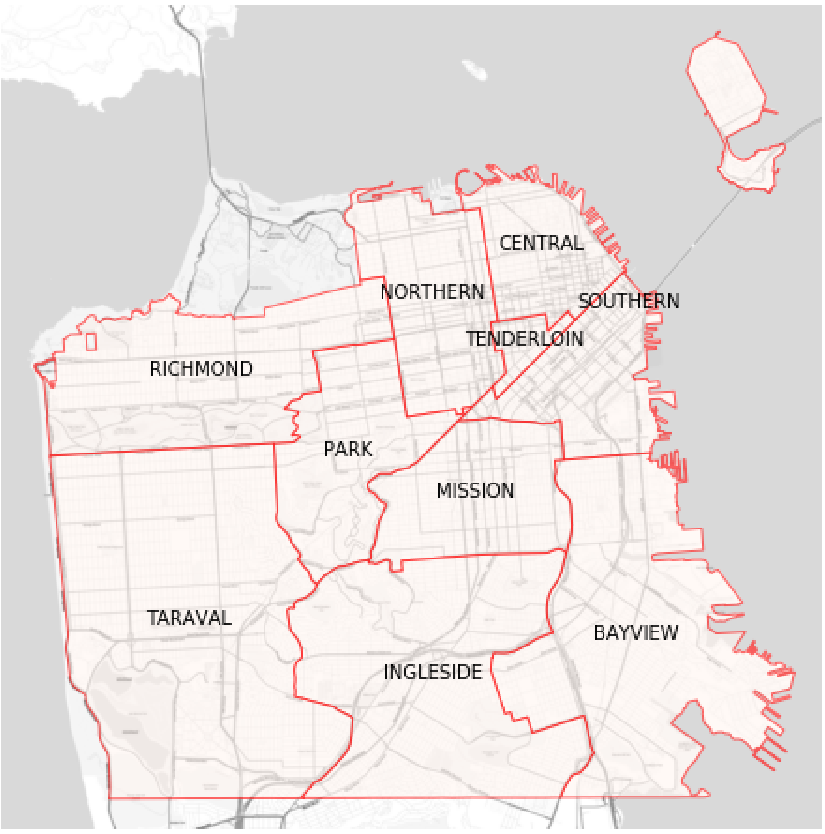


**Figure 3.9:** San Francisco Police Districts

## 3.4 Summary

In this chapter, the study scenario for this dissertation was presented. Briefly, the objective is to make a crime risk prediction for the city of San Francisco. For this, a dataset from the same city will be used. Data from this dataset were also explored through graphs in order to understand the data that will be used.

This work is a fusion of the approach of McClendon *et al.* given that historical data is used to predict future criminal events, as well as that of Lin *et al.*, since the spatial-temporal approach is also employed, considering it is aimed to make a prediction for each neighborhood for a given time interval.

CHAPTER 4

# Data Preparation

In the vast majority of times, raw data can contain errors, which impairs the data quality and in turn will influence the results. Data preparation usually takes a considerable part of the machine learning process, but it is crucial for removing faulty data and filling in gaps. Data preparation has as main goal to catch errors and inconsistencies with the aim to enhance data quality that can be processed and analyzed more quickly and efficiently. Taking into account the described scenario, a set of operations were carried out in order to optimize the quality of the data.

The performance of the machine learning algorithms depends significantly on the treatment that is given to the data previously; for that reason, the pre-processing is an iterative process with the aim to improve constantly data quality. It is important to note that many of the decisions taken in this section were the result of many attempts to increase the quality of the data.

## 4.1 Handling Missing Values

Before handling missing values, it was checked for duplicate values in the dataset, but none was found.

$NULL$ or $NaN$ values can greatly impair the performance of the model or even make it impossible to build it, then it is necessary to make some interventions [66]. Table 4.1 shows the number (absolute and relative) of missing values of each dataset column.

As can be seen in Table 4.1 there are fourteen columns with missing values. Then it is necessary to analyse values contained in these columns to make a decision in the way to treat missing values. As stated in 2.2.4 in the process of handling missing values, there are several ways to handle this issue, such as delete columns/records or fill in the missing data with the mean/median of the attribute value, or even try to predict what is the missing value.

**Table 4.1:** Missing values of each dataset column

| Column Name | Description | Type |
|---|---|---|
| *Incident Datetime* | 0 | 0% |
| *Incident Date* | 0 | 0% |
| *Incident Time* | 0 | 0% |
| *Incident Year* | 0 | 0% |
| *Incident Day of Week* | 0 | 0% |
| *Report Datetime* | 0 | 0% |
| *Row ID* | 0 | 0% |
| *Incident ID* | 0 | 0% |
| *Incident Number* | 0 | 0% |
| *CAD Number* | 73960 | 23.286% |
| *Report Type Code* | 0 | 0% |
| *Report Type Description* | 0 | 0% |
| *Filed Online* | 248627 | 78.279% |
| *Incident Code* | 0 | 0% |
| *Incident Category* | 19 | 0.006% |
| *Incident Subcategory* | 19 | 0.006% |
| *Incident Description* | 0 | 0% |
| *Resolution* | 0 | 0% |
| *Intersection* | 17037 | 5.364% |
| *CNN* | 17037 | 5.364% |
| *Police District* | 0 | 0% |
| *Analysis Neighborhood* | 17099 | 5.383% |
| *Supervisor District* | 17037 | 5.364% |
| *Latitude* | 17037 | 5.364% |
| *Longitude* | 17037 | 5.364% |
| *Point* | 17037 | 5.364% |
| *Current Police Districts* | 17492 | 5.507% |
| *Current Supervisor Districts* | 17095 | 5.382% |
| *Analysis Neighborhoods* | 17157 | 5.402% |

Now, it will be explained the actions taken to handle all missing values of the dataset (up to down in order of Table 4.1).

*CAD Number* is used to dispatch officers and other public safety personnel. Therefore, given that the field is not significant for final purposed and has a high number of missing values, then this column was deleted.

*Filled Online* is a boolean field that indicates if the record was filed online by citizens or not. As can be seen, this field has about 78% of missing fields; however, after analysing this column, it is was noticed that just *True* values are filled, which means that all records filed by officers were represented as *NaN*. For this reason, all *NaN* records were replaced by *False*.

The records where *Incident Category* is *NaN* are the same where *Incident Subcategory* is *NaN* too. Hence, after a deep review, it was not possible to infer what kind of category the incidents belonged to. Moreover, seeing that the records just represent 0.006% of the total, then the records were deleted.

The same case happened for the following columns: *Intersection*, *CNN*, *Supervisor District*, *Latitude*, *Longitude* and *point*. Once all these columns represent location, they are related between them, then it is not possible to be fill them. One possibility would be fill them with mode, i.e., with the value that occurs most often, but the values of these fields are quite varied, so a lot of wrong data would be entered.

*Analysis Neighborhood* and *Analysis Neighborhoods* have a relation 1-1 between them. *Analysis Neighborhood* contains the name of a neighborhood, for example, Chinatown whereas *Analysis Neighborhoods* represents the ID of a certain neighborhood, which in the case of Chinatown is ID 6. In this case, there are more missing values in *Analysis Neighborhoods* (IDs) than in Analysis Neighborhood (names) which means that it is possible to infer some IDs with the names and not the other way around because there are no records where *Analysis Neighborhoods* is filled and *Analysis Neighborhoods* do not. In this way, records, where Analysis Neighborhood is *NaN* was deleted, and the remaining *Analysis Neighborhoods*was filled with the ID of *Analysis Neighborhood*. For this, it was necessary to create a dictionary that maps all *Analysis Neighborhoods* and *Analysis Neighborhood*.

After processing the missing values stated before, remain 393 record of *Police Districts* that are *NaN*. It was verified if the values in this column were related with some columns of location but, as opposed to *Analysis Neighborhood* and *Analysis Neighborhoods*, there are no relation. Records where *Current Police Districts* was *NaN*, *Current Police Districts* and *Current Supervisor Districts* were also *NaN*. In this sense, the missing values of this column were deleted.

After all this processing to handle missing values, the dataset went from 317613 to 300102 rows, which represents a decrease of 17511, about 5.5%.

## 4.2 Removing non-crime categories

As shown in Figure 3.6, there are 50 different categories of incidents, however not all of these categories are crimes. Hence, all records whose category is not considered as a crime have been removed. Tables 4.2 and 4.3 shows all incidents categories and their frequencies.

**Table 4.2:** Incidents Categories and its frequency (part 1)

| Category | Count |
|---|---|
| Larceny Theft | 92582 |
| Other Miscellaneous | 23541 |
| Non-Criminal | 19384 |
| Assault | 18689 |
| Malicious Mischief | 17994 |
| Burglary | 13635 |
| Warrant | 11518 |
| Motor Vehicle Theft | 11072 |

**Table 4.3:** Incidents Categories and its frequency (part 2)

| | |
|---|---|
| Fraud | 9319 |
| Lost Property | 9281 |
| Drug Offense | 8102 |
| Robbery | 7425 |
| Missing Person | 7243 |
| Recovered Vehicle | 6717 |
| Lost Property | 6717 |
| Offences Against The Family And Children | 6064 |
| Suspicious Occ | 5872 |
| Disorderly Conduct | 5573 |
| Traffic Violation Arrest | 4116 |
| Miscellaneous Investigation | 2758 |
| Other Offenses | 2240 |
| Other | 2047 |
| Weapons Carrying Etc | 1653 |
| Weapons Offense | 1629 |
| Stolen Property | 1586 |
| Forgery And Counterfeiting | 1358 |
| Case Closure | 1337 |
| Courtesy Report | 916 |
| Sex Offense | 900 |
| Civil Sidewalks | 846 |
| Prostitution | 794 |
| Arson | 709 |
| Traffic Collision | 599 |
| Embezzlement | 428 |
| Family Offense | 400 |
| Vandalism | 399 |
| Fire Report | 317 |
| Vehicle Impounded | 230 |
| Suicide | 145 |
| Vehicle Misplaced | 137 |
| Drug Violation | 104 |
| Human Trafficking (A), Commercial Sex Acts | 101 |
| Rape | 83 |
| Liquor Laws | 72 |
| Suspicious | 56 |
| Motor Vehicle Theft? | 45 |
| Homicide | 38 |
| Gambling | 20 |
| Weapons Offence | 14 |
| Human Trafficking, Commercial Sex Acts | 13 |
| Human Trafficking (B), Involuntary Servitude | 1 |

The following categories have been removed:

- Non-Criminal
- Lost Property
- Case Closure
- Courtesy Report
- Traffic Collision
- Fire Report
- Vehicle Misplaced
- Missing Person
- Recovered Vehicle

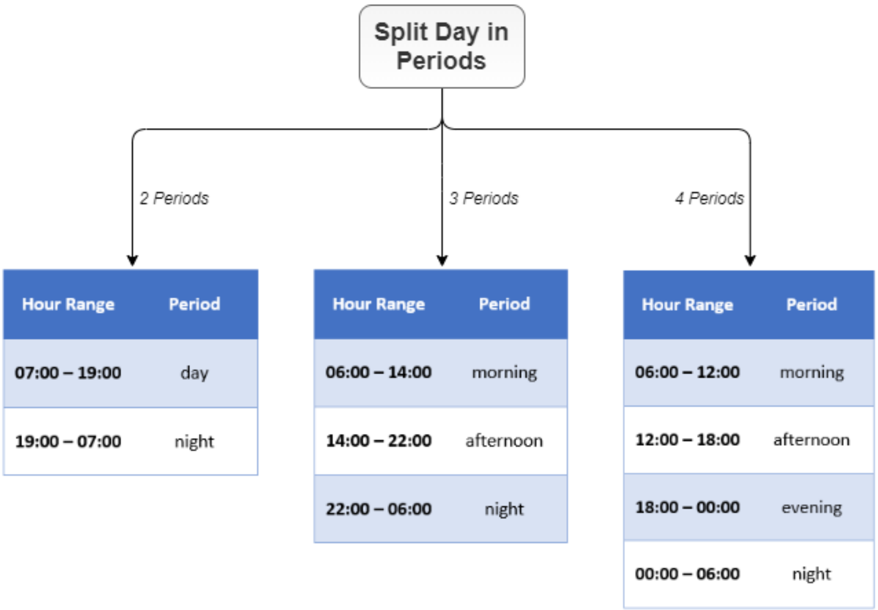
After removing these categories, the dataset was left with 254171 rows, that is, there were 45931 rows that were not crimes. Also, the charts shown in Section 3.3 were again built; however, there were no significant differences in the distribution of crimes either by months, days of the week, time of the day or neighborhoods compared to the distribution of all incidents.

## 4.3 Handling Times

In the scenario described, the idea is to divide the day into periods. Then, according to this requirement, a new column called *Period_of_day* was created.

Data pre-processing is not a straightforward step; data pre-processing is an iterative process, since sometimes small changes can have a significant impact on the model's performance.

In this sense, three different approaches were followed to fill this column according to the population lifestyle and the daylight hours. In one approach, the day was split into two equal periods, another into three equal periods and finally into four equal periods. Figure 4.1 shows the time interval in which each period fits.



| Hour Range | Period |
|---|---|
| 07:00 – 19:00 | day |
| 19:00 – 07:00 | night |

| Hour Range | Period |
|---|---|
| 06:00 – 14:00 | morning |
| 14:00 – 22:00 | afternoon |
| 22:00 – 06:00 | night |

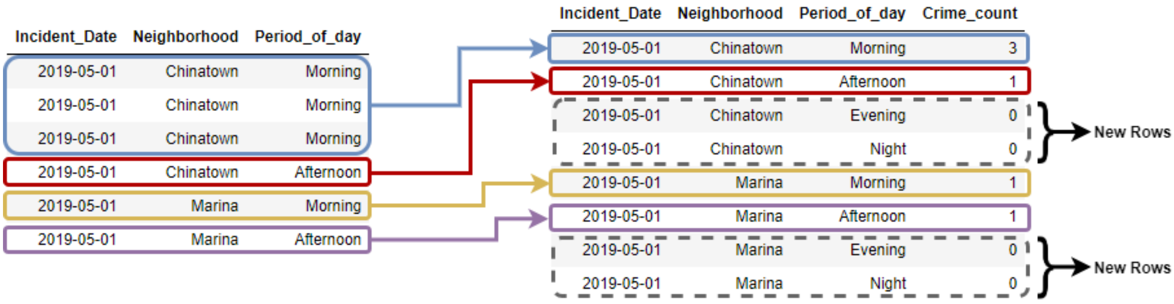
| Hour Range | Period |
|---|---|
| 06:00 – 12:00 | morning |
| 12:00 – 18:00 | afternoon |
| 18:00 – 00:00 | evening |
| 00:00 – 06:00 | night |

**Figure 4.1:** Splitting of day approaches

## 4.4 Group Crimes by neighborhood and period of day

As shown in Figure 3.1, the neighborhood and the period of the day were used to make a prediction. In this sense, a grouping of crimes was made taking into account the values in columns *Incident_Date*, *Neighborhood* and *Period_of_day*.

Figure 4.2 illustrates the grouping that was done. In the chunk of dataset on the left is a part of the original dataset, and the other dataset on the right is the resulting dataset after grouping. The objective is to obtain a count of crimes that occurred on a certain date, in a certain neighborhood, at a certain period of the day. As can be seen, on the 1st May 2019 (2019-05-01) in Chinatown, three crimes were committed in the morning. In this case, these 3 lines will be grouped in only one and a new column is created called *Crime_count* which will be filled with a 3. This action is represented by a blue box in Fig. 4.2. When there is no crime on a certain date, the neighborhood and the period of day, then, it will be added a new line with value of *Crime_count* equal to 0. This operation is represented in Figure 4.2 with a dashed line.



| Incident_Date | Neighborhood | Period_of_day |
|---|---|---|
| 2019-05-01 | Chinatown | Morning |
| 2019-05-01 | Chinatown | Morning |
| 2019-05-01 | Chinatown | Morning |
| 2019-05-01 | Chinatown | Afternoon |
| 2019-05-01 | Marina | Morning |
| 2019-05-01 | Marina | Afternoon |

| Incident_Date | Neighborhood | Period_of_day | Crime_count |
|---|---|---|---|
| 2019-05-01 | Chinatown | Morning | 3 |
| 2019-05-01 | Chinatown | Afternoon | 1 |
| 2019-05-01 | Chinatown | Evening | 0 |
| 2019-05-01 | Chinatown | Night | 0 |
| 2019-05-01 | Marina | Morning | 1 |
| 2019-05-01 | Marina | Afternoon | 1 |
| 2019-05-01 | Marina | Evening | 0 |
| 2019-05-01 | Marina | Night | 0 |

**Figure 4.2:** Grouping crimes by neighborhood and period of day

*Crime_count* will be the target when using regression algorithms, as this column represents a quantity.

## 4.5 Risk categorization

To apply classification algorithms, it is necessary to categorize the crime risk. Such as the splitting of the day in periods, several approaches have been followed, now there will also be 2 approaches to categorizing risk.

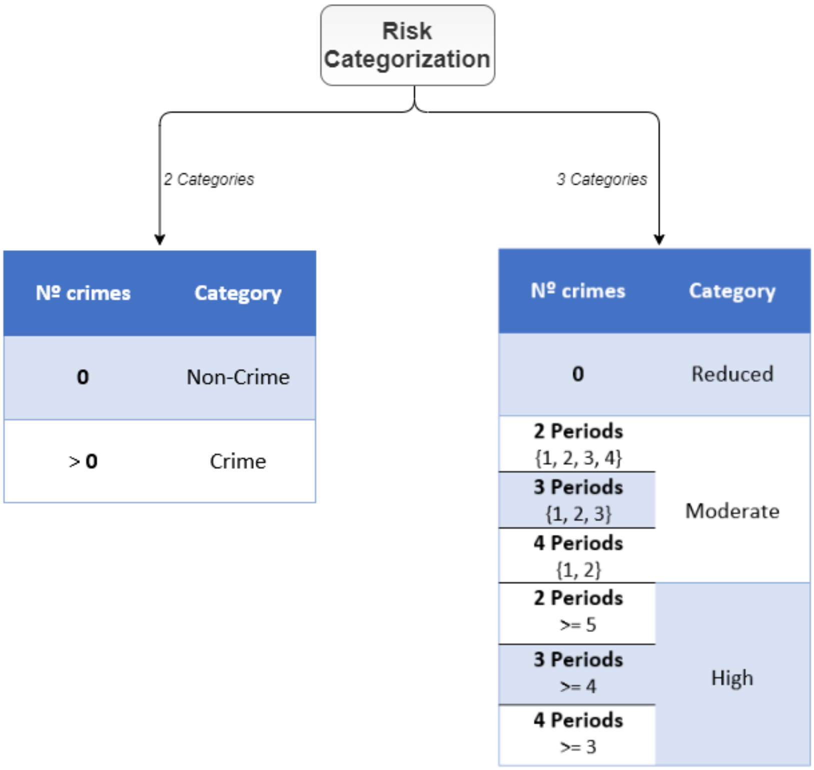
In the first approach, a binary classification will be made, that is, there will be two possible values for the prediction, **Non-Crime** or **Crime**. The criteria for this classification is the following: if the number of crimes is equals zero, then the category will be Non-Crime; otherwise, if number of crimes is greater than zero, then the category will be Crime.

In the second approach, a multiclass classification will be made, thus, in this case there will be three possible classifications for the risk: **Reduced**, **Moderated** and **High**. Unlike binary classification, the criteria for this classification depends of the number of periods of the day. Considering that the day is split into two periods (day and night) each period will have a duration of 12 hours; then, it is normal to have more crimes than in a division of the day into four periods (morning, afternoon, evening and night) where each period has a duration of 6 hours.

For this reason, it was examined the mean of crimes per period for all approaches in the split. If the number of crimes is equal to zero, the category will be Reduced for all approaches. For the allocation of the remaining categories, the mean number of crimes per period in each approach was taken into account and has the following values:

- **2 periods** (day and night): mean is 4
- **3 periods** (morning, afternoon and night): mean is 3
- **4 periods** (morning, afternoon, evening and night): mean is 2

Thus, the risk will be considered moderate if the number of crimes is less or equal to the mean of crimes. Note that these values can be configured. Figure 4.3 demonstrates the two approaches to risk categorization explained above.



| Nº crimes | Category |
|---|---|
| 0 | Non-Crime |
| > 0 | Crime |

| Nº crimes | Category |
|---|---|
| 0 | Reduced |
| 2 Periods<br>{1, 2, 3, 4} | Moderate |
| 3 Periods<br>{1, 2, 3} | |
| 4 Periods<br>{1, 2} | |
| 2 Periods<br>>= 5 | High |
| 3 Periods<br>>= 4 | |
| 4 Periods<br>>= 3 | |

**Figure 4.3:** Risk categorization according to the number of crimes

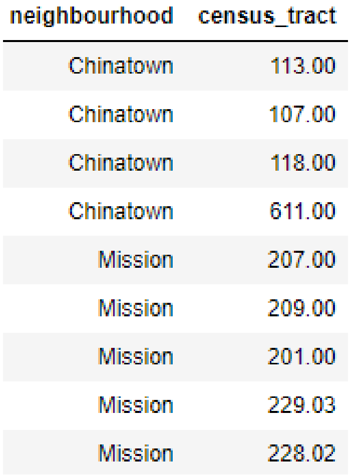
## 4.6 Merging other data sources

In order to enrich the current dataset, a search was made for data that could be useful considering the scenario. However, it is a real challenge to find good data both in terms of quantity and quality. Since the location sent as an input will be the neighborhood, it was searched data that could add value to this attribute.

No information was found directly related to the neighborhood, but about census tracts. According to the description about Analysis Neighborhood in Table 3.3, it was created

41 neighborhoods by grouping 2010 Census tracts, using common real estate and resident definitions for the purpose of providing consistency in the analysis and reporting of socio-economic, demographic, and environmental data. So, through census tracts data, it is possible to infer data for a neighborhood by aggregating data from census tracts that belong to a particular neighborhood.

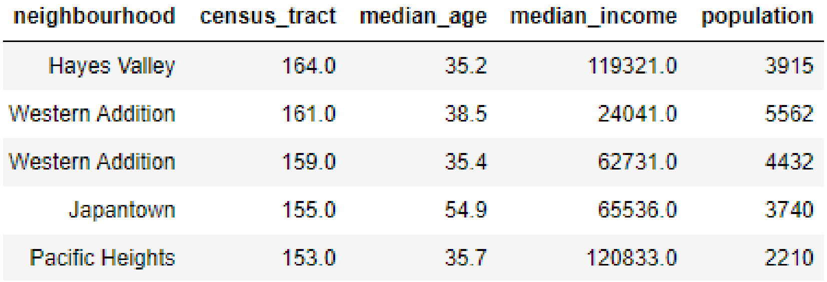
First of all, it is necessary to get the dataset that maps neighborhoods and census tracts. This dataset[1] contains all the census tracts and which neighborhood they belong to. A chunk of this dataset can be seen in Figure 4.4, where it presents census tracts of Chinatown and Mission.



| neighbourhood | census_tract |
|---|---|
| Chinatown | 113.00 |
| Chinatown | 107.00 |
| Chinatown | 118.00 |
| Chinatown | 611.00 |
| Mission | 207.00 |
| Mission | 209.00 |
| Mission | 201.00 |
| Mission | 229.03 |
| Mission | 228.02 |

**Figure 4.4:** Census tracts of Chinatown and Mission

Then, it was sought for the median income[2], median age[3] and the total population[4] per census tract. All of these data have different sources, consequently there are three different datasets. Thus, it was necessary to merge these three datasets to data with the mapping between neighborhood and census tract, resulting in a new dataset, where a chunk is shown in Figure 4.5.



| neighbourhood | census_tract | median_age | median_income | population |
|---|---|---|---|---|
| Hayes Valley | 164.0 | 35.2 | 119321.0 | 3915 |
| Western Addition | 161.0 | 38.5 | 24041.0 | 5562 |
| Western Addition | 159.0 | 35.4 | 62731.0 | 4432 |
| Japantown | 155.0 | 54.9 | 65536.0 | 3740 |
| Pacific Heights | 153.0 | 35.7 | 120833.0 | 2210 |

**Figure 4.5:** Median income, age and total population per census tract

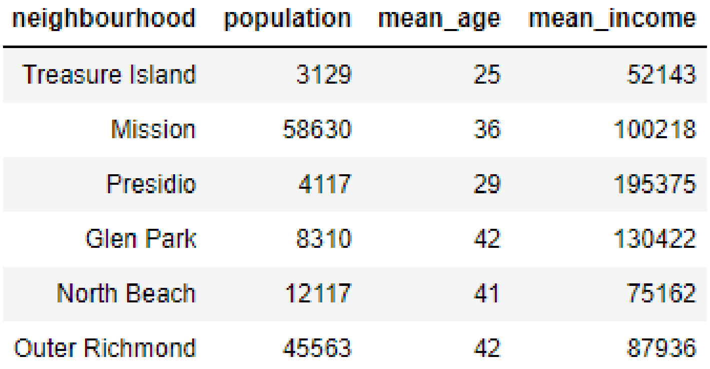
[1] https://data.sfgov.org/Geographic-Locations-and-Boundaries/Analysis-Neighborhoods-2010-census-tracts-assigned/bwbp-wk3r/

[2] https://data.census.gov/cedsci/table?q=san%20francisco%20income

[3] https://data.census.gov/cedsci/table?q=san%20francisco%20age

[4] https://data.census.gov/cedsci/table?q=san%20francisco%20population

Nevertheless, the values of income, age and population are not grouped by neighborhood, but by census tract. For this reason, it was necessary to group these values. In this way, the income and age were aggregated by their mean per neighborhood, whereas in relation to the population, a sum was applied as can be seen in Figure 4.6.



| neighbourhood | population | mean_age | mean_income |
|---:|---:|---:|---:|
| Treasure Island | 3129 | 25 | 52143 |
| Mission | 58630 | 36 | 100218 |
| Presidio | 4117 | 29 | 195375 |
| Glen Park | 8310 | 42 | 130422 |
| North Beach | 12117 | 41 | 75162 |
| Outer Richmond | 45563 | 42 | 87936 |

**Figure 4.6:** Total population, mean age and mean income per neighborhood

To get some insights, it was built three bar charts. Returning to the graph of Figure 3.8, where the distribution of incidents by neighborhood is shown, it is verified that there are four neighborhoods that stand out with their high number of occurrences, they are Mission, Tenderloin, Financial District/South Beach and South of Market.

As can be noted in Figure 4.7, of the ten most populous neighborhoods, two of them are the ones with the highest criminal incidence (Mission and Tenderloin).

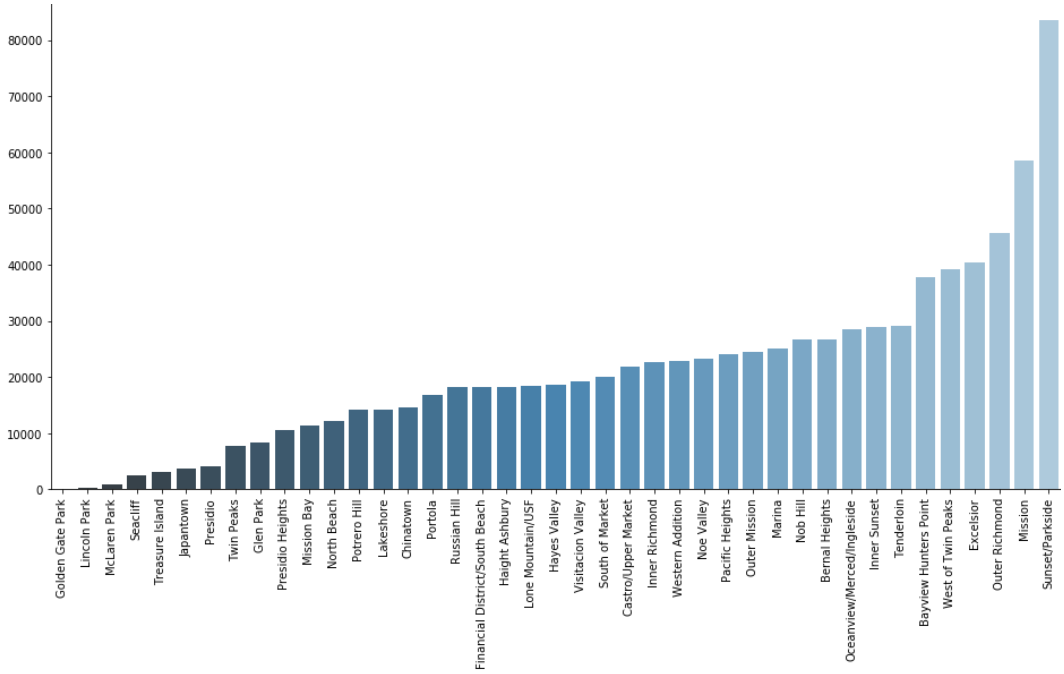


**Figure 4.7:** Population per neighborhood

In Figure 4.8, it can be seen that none of the neighborhoods with the highest criminal incidence are present in the ten with the highest income.

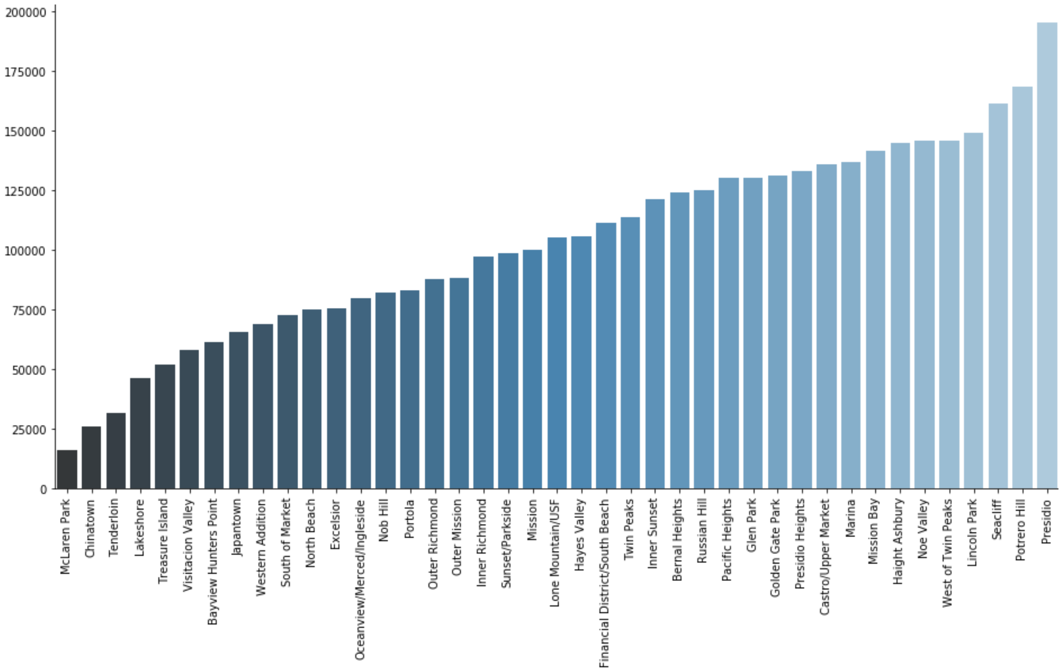


**Figure 4.8:** Average income by neighborhood

Of the highest criminal incidence neighborhoods, only South of Market is in the ten oldest, as shown in the graph of Figure 4.9.

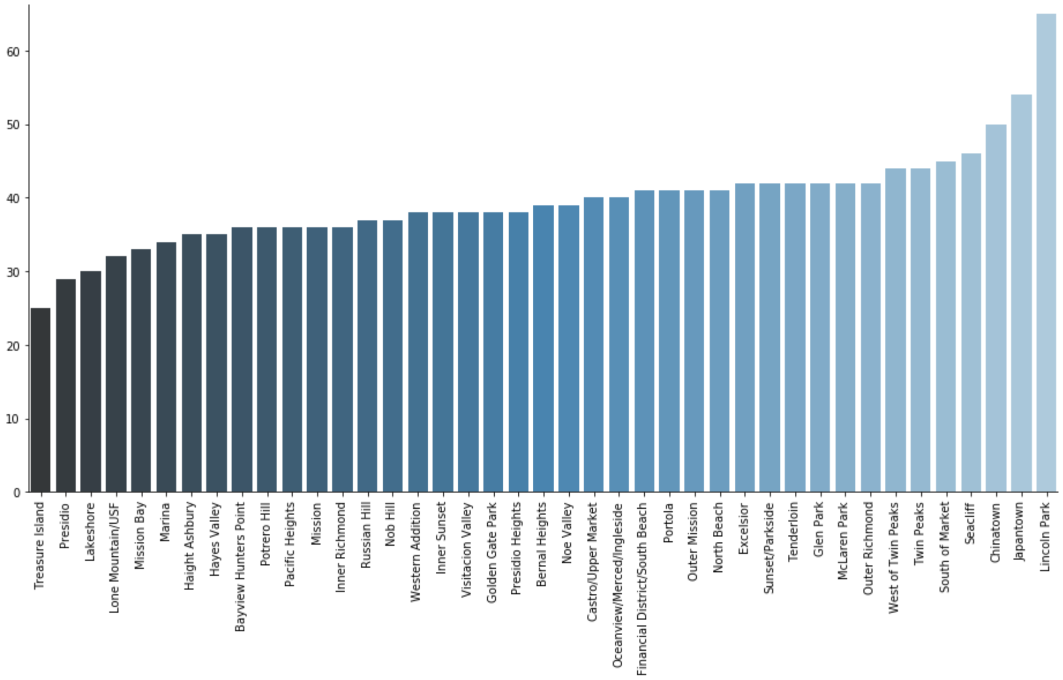


**Figure 4.9:** Average age by neighborhood

The next step is to merge this dataset which already aggregates four datasets (locations, income, age and populations) on the census tract, to the dataset that contains the crimes on the neighborhood column. Figure 4.10 illustrates this merge process.

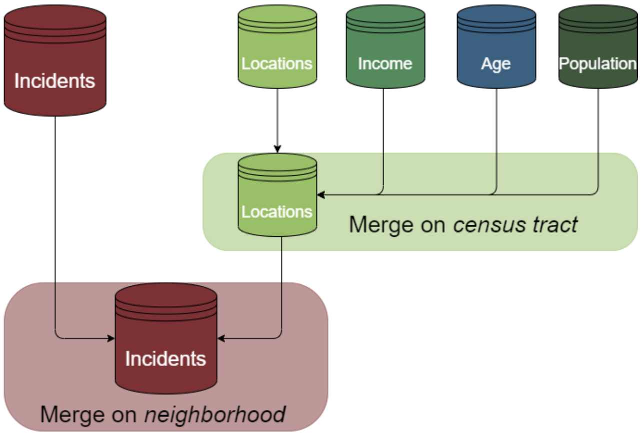


**Figure 4.10:** Merge of dataframes

To verify the correlation of new data with the number of crimes, it was built a correlation matrix in a heatmap form. Briefly, a correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data. The aim of the heatmap is to provide a colored visual summary of information, and it is great to see trends and correlations between data. Then, it can be noted in Figure 4.11 that the amount of population has some correlation with the number of crimes; on the other hand, the median age has a weak correlation with the number of crimes[5]. In reality, these values make sense, since a place where there are many people is more prone to confusion and misunderstandings.

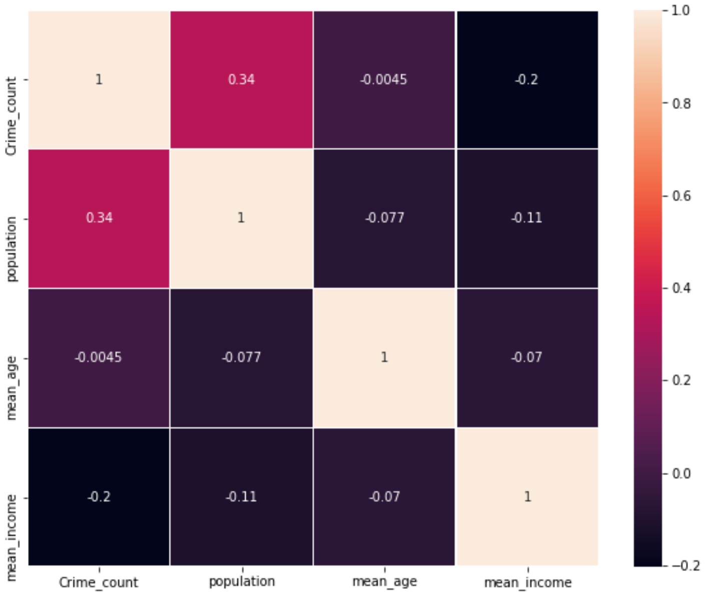


**Figure 4.11:** Correlation between population, mean income, mean age and number of crimes

[5] This correlation matrix was built for dividing the day into 2 parts (day and night) and for a binary classification (Crime vs. Non-Crime). However, for the remaining approaches, the correlation coefficients are quite similar.

## 4.7 Relationship of periods of day and location with crimes frequency

After the data transformation, it is relevant to build some more visualizations. Bearing in mind that the period of the day and location will be the key attributes to make the prediction, it is important to study in more detail the relationship between these and the crimes frequency.

Since a day is divided into several parts, some differences are noted in the frequencies of the number of crimes per period. To have a better understanding of these differences, some visualizations were built for each type of division.

### Day and Night

In this approach, the day is divided into two parts, each representing **12 hours**. In Figure 4.12, each point represents one crime, and it is clear that there is a higher number of crimes recorded during the day time period compared to night. Also, there is a higher dispersion in the count of crimes during the day. These scattered points are outliers, that is, they represent records where there was an exceptional high number of crimes. While at night the number of crimes is usually lower, during the day this number can grow to higher values.

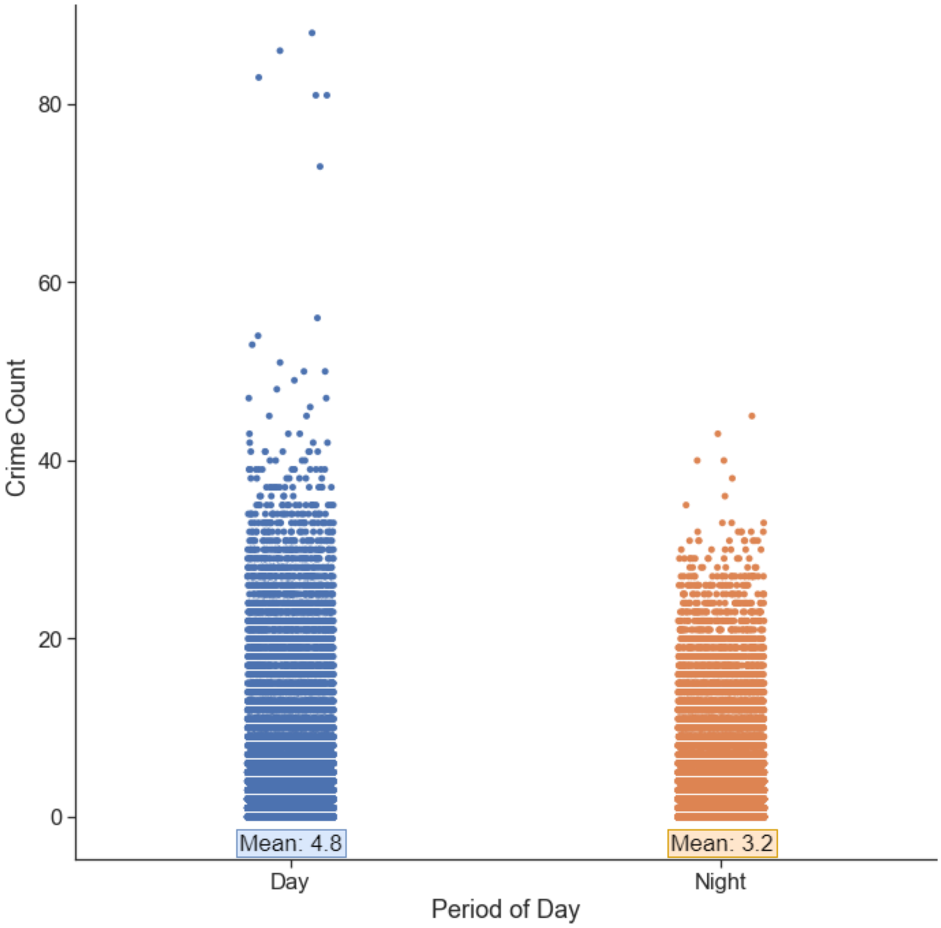


**Figure 4.12:** Comparison of crime counts between day and night

### Morning, Afternoon and Night

In this approach, the day is divided into three parts, morning, afternoon and night, each consisting of **8 hours**. Figure 4.13 shows that the night period has the lowest mean crime rate, while the morning and afternoon have mean crime rates of 2.6 and 3.6, respectively. It can be seen that there is a greater distribution of crimes during the morning or afternoon, while during the night the number of crimes is usually reduced.

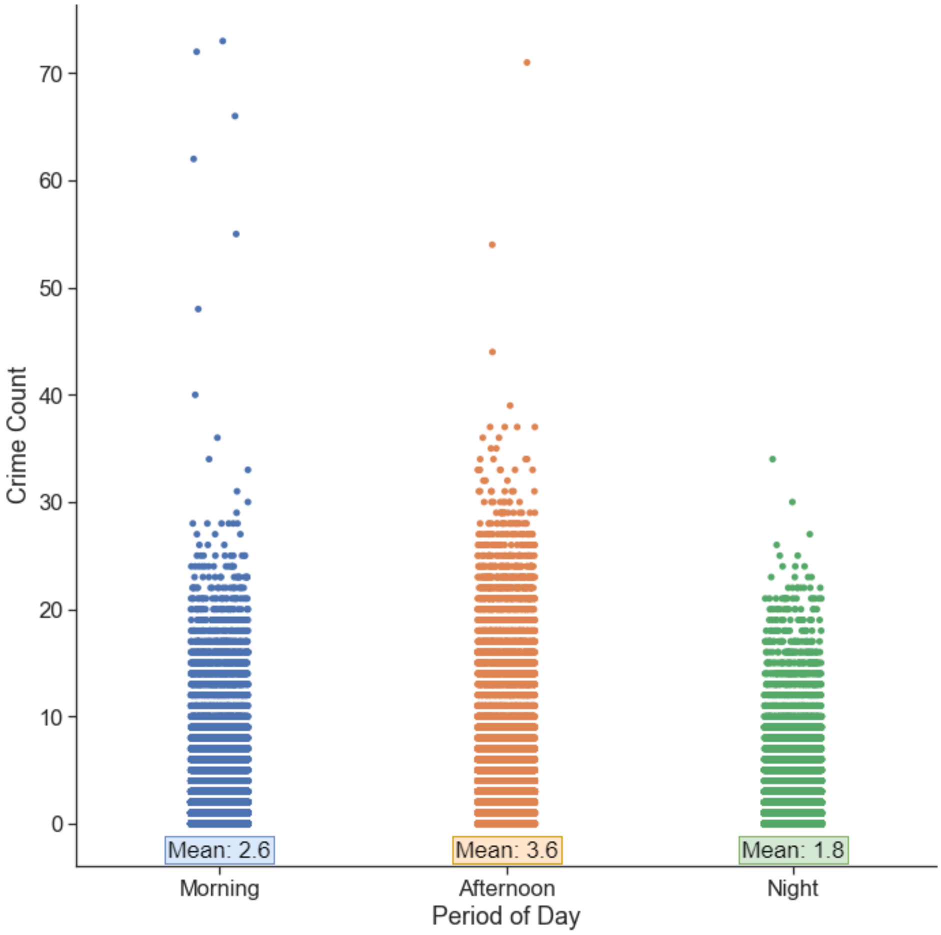


**Figure 4.13:** Comparison of crime counts between morning, afternoon and night

### Morning, Afternoon, Evening and Night

In this approach, the day is divided into four parts, morning, afternoon, evening and night, with each part consisting of **6 hours**. In Figure 4.14, it can be noted that, once again, night and morning have the lowest crime rates, that is, less of 2 in both cases, whereas during the evening and afternoon the mean is equal to or greater than 2.5. It is interesting to note that, although the average crime rate is lower in the morning compared to evening, there are more outliers in the morning.

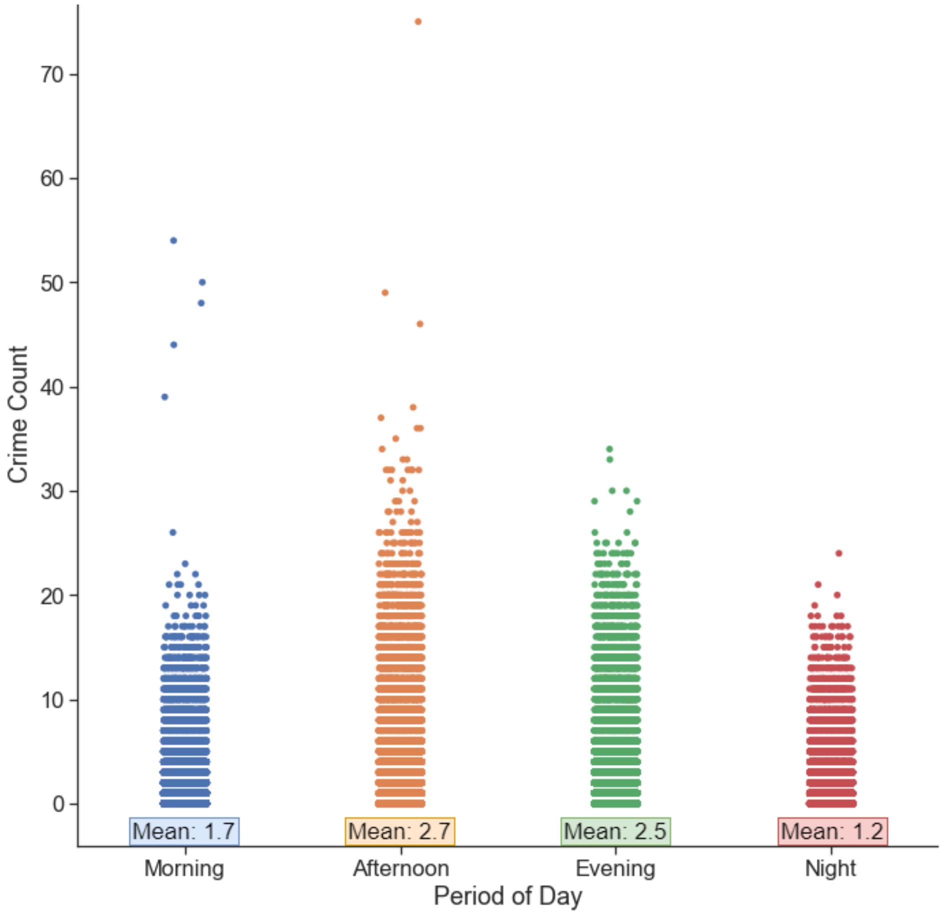


**Figure 4.14:** Comparison of the average crime count between morning, afternoon, evening and night

### Location

In order to obtain a sense of the geographical areas that are most likely to occur crimes, a visualization was developed to understand which areas have the highest criminal density. For this, it was calculated the average of the incidents per day for every police district. As can be noted in Figure 4.15, the northeastern zone is the most critical zone, as it has a higher criminal density as opposed to the west and south zones which appear to be more peaceful.

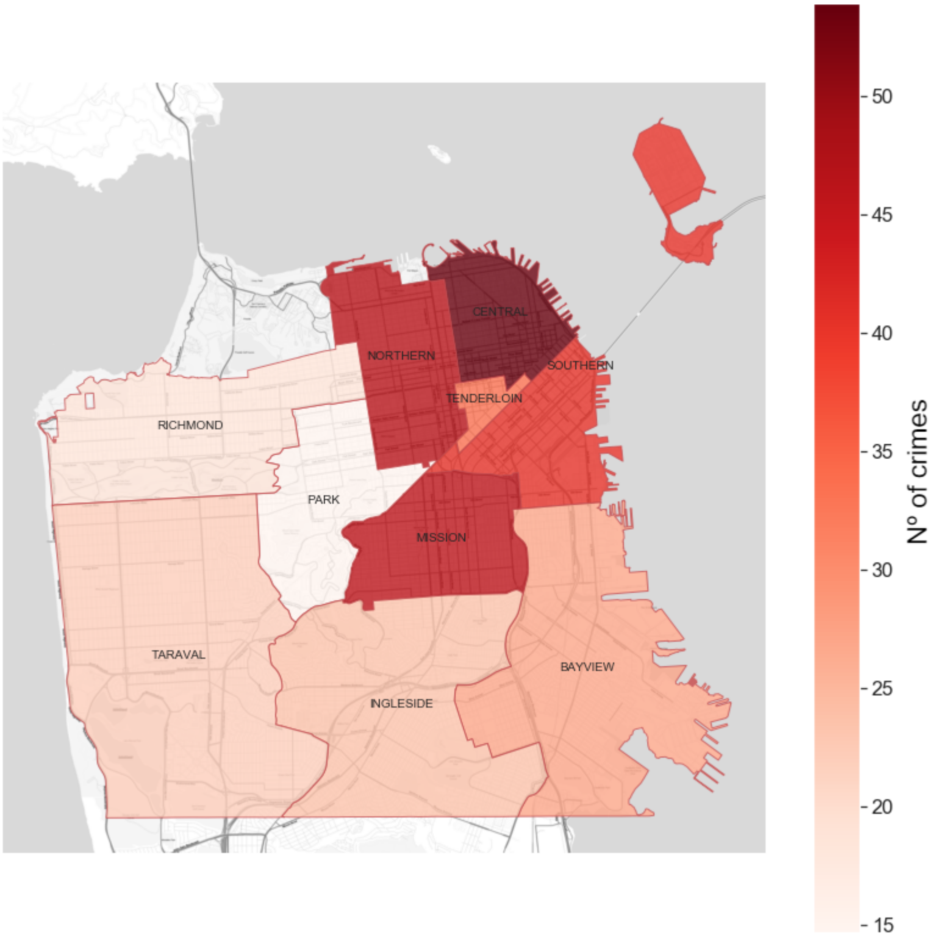


**Figure 4.15:** Criminal density by district

## 4.8 Summary

As already noted, data preparation is one of the most important and time-consuming stages in a machine learning project. In this chapter, the strategies for cleaning, preparing and transforming the data were explained. The decisions taken must have in view the final application that was demonstrated in this chapter.

Given the nature of the data, there was an effort to make several transformations on the data, such as dividing the day into several periods; in this case, three approaches were followed that will be tested and discussed in the next chapters.

It is also necessary to categorize the risk of crime based on the number of crimes that have occurred. Bearing in mind that, in each approach in the division of the day, the number of hours in each period varies, so the risk of crime should accompany this variation. For this purpose, the mean of crimes in each approach was used to distinguish between moderate and high crime risk.

To enrich the data, there was also a concern to look for other sources that could add value and thus improve the learning model to be developed.

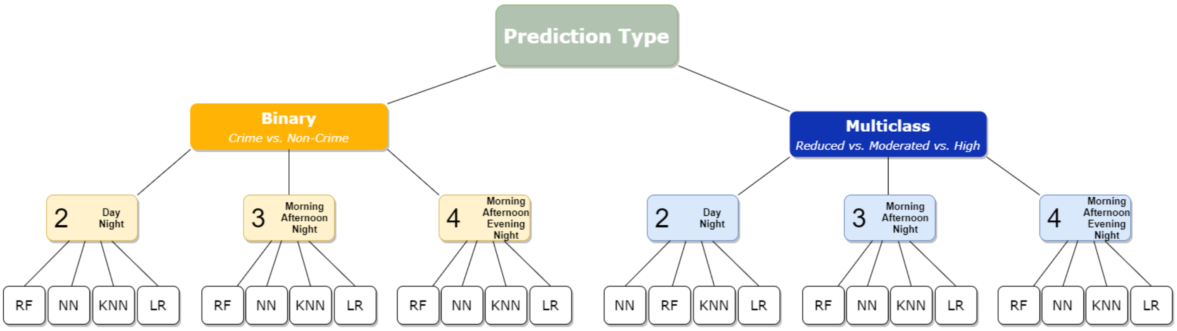
CHAPTER 5

# Classification Model

This chapter describes the entire process in the development of the crime risk classification model. First, the various approaches to learn the model will be presented. Next, the evaluation metrics of the models will be described. The four algorithms, presented in section 2.2.3, will be used: Random Forest, Neural Network, K-Nearest Neighbors and Logistic Regression. Finally, the results obtained for the various tested approaches will be presented and discussed.

## 5.1 Approaches to modeling

In this section, the distinct approaches presented in the previous section will be tested with the different algorithms. Figure 5.1 shows all combinations between different approaches and different algorithms. The main objective is to compare the results of the different algorithms for the different approaches.

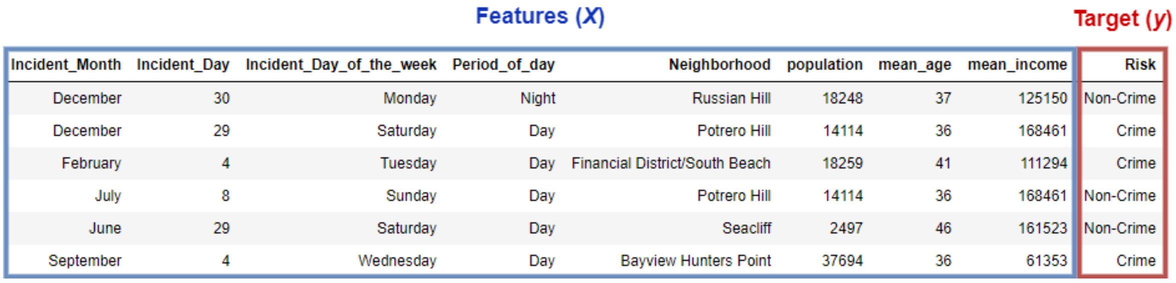


Prediction Type
Binary
Crime vs. Non-Crime
Multiclass
Reduced vs. Moderated vs. High
2 Day Night
3 Morning Afternoon Night
4 Morning Afternoon Evening Night
2 Day Night
3 Morning Afternoon Night
4 Morning Afternoon Evening Night
RF NN KNN LR
RF NN KNN LR
RF NN KNN LR
NN RF KNN LR
RF NN KNN LR
RF NN KNN LR

**Figure 5.1:** Combining the different approaches with the various algorithms

Since only supervised algorithms will be used, it must be indicated which columns represent the descriptive resources and which column represents the label, that is, the target. Figure 5.2 presents six samples of the dataset. As can be noted, the first height columns will be used as descriptive columns and the last one, the risk of crime, will be used as the target.

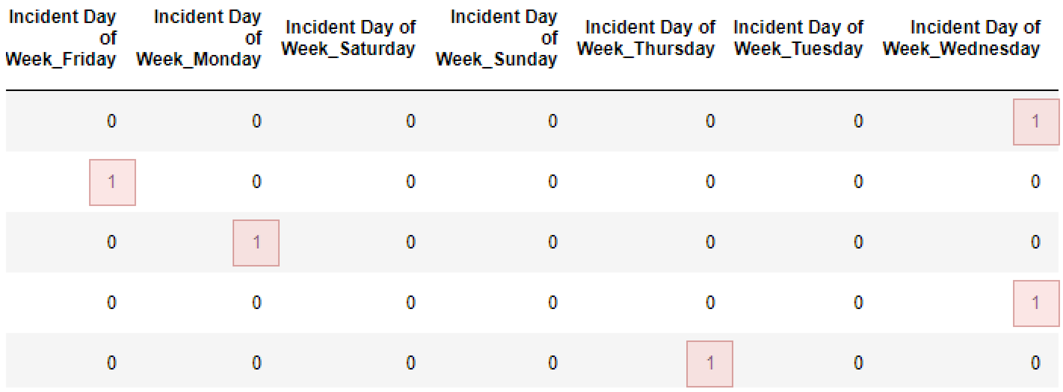
Later, the feature selection was performed, and as a result, *Incident_Month* and *Incident_Day* were not used, because it was found that these features would not add value and they would decrease the performance of the model.


Features (*X*)
Target (*y*)

| Incident_Month | Incident_Day | Incident_Day_of_the_week | Period_of_day | Neighborhood | population | mean_age | mean_income | Risk |
|---|---|---|---|---|---|---|---|---|
| December | 30 | Monday | Night | Russian Hill | 18248 | 37 | 125150 | Non-Crime |
| December | 29 | Saturday | Day | Potrero Hill | 14114 | 36 | 168461 | Crime |
| February | 4 | Tuesday | Day | Financial District/South Beach | 18259 | 41 | 111294 | Crime |
| July | 8 | Sunday | Day | Potrero Hill | 14114 | 36 | 168461 | Non-Crime |
| June | 29 | Saturday | Day | Seacliff | 2497 | 46 | 161523 | Non-Crime |
| September | 4 | Wednesday | Day | Bayview Hunters Point | 37694 | 36 | 61353 | Crime |

**Figure 5.2:** Split descriptive columns from the categorical target column

It can also can be seen in Figure 5.2 that there are several categorical variables, that need to be converted them to numeric form. For that, it was used *get_dummies*[1] function from *pandas*. This function creates binary columns (dummy columns) for each categorical value and assigns 1 to the column if the feature belongs or 0 otherwise. Figure 5.3 shows the results after the transformation for column *Incident_Day_of_the_week*.



| Incident Day of Week_Friday | Incident Day of Week_Monday | Incident Day of Week_Saturday | Incident Day of Week_Sunday | Incident Day of Week_Thursday | Incident Day of Week_Tuesday | Incident Day of Week_Wednesday |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |

**Figure 5.3:** Conversion of column *Incident_Day_of_the_week* from categorical to numeric

In relation to the numerical columns, tests were also performed with and without the scaling of the features. For example, the min and max values of the *mean_age* column are 25 and 65 years, respectively, which makes a range of 40 years; on the other hand, the min and max values of *mean_income* are 16016 and 195375, respectively, which makes a range of 179359, so, the values of these two columns are on completely different scales. To perform the scaling it was used the $MinMaxScaler$[2] from $scikit-learn$ This estimator scales and translates each feature individually between zero and one. For each value in a feature, it will subtract the minimum value in the feature and then divides it by the range (the formula is presented in Equation 2.3). $MinMaxScaler$ maintains the shape of the original distribution. It does not meaningfully change the information embedded in the original data and outliers are preserved.

[1] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

In some tested approaches, the data were relatively unbalanced. To address this issue there are two techniques: **undersampling** - this technique randomly removes samples from the majority class, with or without replacement; and **oversampling** - based on the samples already presented in the dataset, this technique creates new samples from the minority class. The use of oversampling presented better results with a technique called SMOTE (Synthetic Minority Over-sampling TEchnique). This function is provided by $imbalanced - learn$[3].

## 5.2 Defining Model Evaluation Rules

Model evaluation metrics are required to quantify model performance; to evaluate the model, it is necessary to split the data into training and testing.

Firstly, it was used the classic approach to do a simple splitting of data, where 75% is used for training and 25% for testing. The function *train_test_split*[4] of $scikit - learn$ allows making a proportional division of the dataset taking into account the target For this, it is necessary to define the $stratify$ parameter which is the target. This issue is very important in the validation of the model, to ensure that all classes of the target are proportionally tested.

However, in order to provide ample data for training the model and to leave ample data for validation, it was used Cross validation. With this method, the data is divided into $k$ subsets (folds). The holdout method is repeated $k$ times, such that each time, one of the $k$ folds is used as the test data and $k - 1$ of the folds as training data. This significantly reduces bias and also significantly reduces variance, as most of the data is also being used in the validation set. Interchanging the training and test sets also adds to the effectiveness of this method. It was used four folds, which means that each fold has 25% of data. The performance measure reported by k-fold cross-validation is then the average of the values computed in the loop, in this case, with four iterations.

Thus, it was used *cross_validate*[5] method from $scikit-learn$. This method allows specify multiple metrics for evaluation, and it also returns a dict containing fit-times, score-times in addition to the test score.

The following evaluation metrics were used.

### Precision

Out of all the positive classes that have been predicted correctly, the precision gives how many are actually positive. Precision is a good measure to determine when the costs of False Positive is high. For instance, in email spam detection, a false positive means that an email that is non-spam (actual negative) has been identified as spam (predicted spam). The email

[3] `https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html`

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`

[5] `https://scikit-learn.org/stable/modules/cross_validation.html`

user might lose important emails if the precision is not high for the spam detection model. The formula of precision is presented in Equation 5.1.

$$\text{precision} = \frac{TP}{TP + FP} \tag{5.1}$$

### Recall

Out of all the positive classes, the recall gives how much has been predicted correctly. Recall calculates how many of the actual positives the model captures through labeling it as Positive (True Positive). For instance, a sick patient (Actual Positive) goes through the test and predicted as not sick (Predicted Negative). The cost associated with False Negative will be extremely high if the sickness is contagious. The formula of recall is presented in Equation 5.2.

$$\text{recall} = \frac{TP}{TP + FN} \tag{5.2}$$

### F1 Score

F1 Score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more. F1 Score might be a better measure to use if is needed to seek a balance between Precision and Recall and there is an uneven class distribution (large number of Actual Negatives). The formula for the F1 Score is presented in Equation 5.3.

$$\text{F1 Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5.3}$$

The results of each metric of each fold will be the average weighted by support (the number of true instances for each label). The final results will be the mean of the results of the folds.

## 5.3 Hyperparameter Tuning

Before executing the algorithms for the different approaches, it was verified which hyperparameters increased performance. Briefly, a hyperparameter is a parameter of the own algorithm whose value is used to control the learning process, and its optimization and adjustment aims to take full advantage of the algorithm, that is, improve the performance. In *sklearn*, hyperparameters are passed in as arguments to the constructor of the model classes and must be set before training.

The best way to choose the right parameters and to guarantee the optimal combination of them is by exhaustive searching all combinations through all parameters of the algorithm. Nevertheless, this process takes a large time and it is computationally expensive since there are many combinations to be tested. Subsequently, only the parameters that most influence the performance are examined.

In this sense, the hyperparameter tuning was tested for binary classification approach (Crime vs. Non-Crime) and division of days into two parts (day and night). It was noted that in most cases the best results are achieved with the parameters already assigned by default. Nevertheless, some improvements have been achieved by changing some parameters as will be shown below.

### Random Forest

It was used $RandomForestClassifier$[6] estimator from *sklearn* to perform the model with Random Forest. Figure 5.4 shows the variations of evaluation metrics according to the number of trees in the Random Forest, and it can be noted that from 10 to 300 decision trees, the results do not vary much. The best result found is with 30 decision trees and this is the number of decision trees used. The number of trees in the forest by default is 100; hence with fewer trees it were found best results.

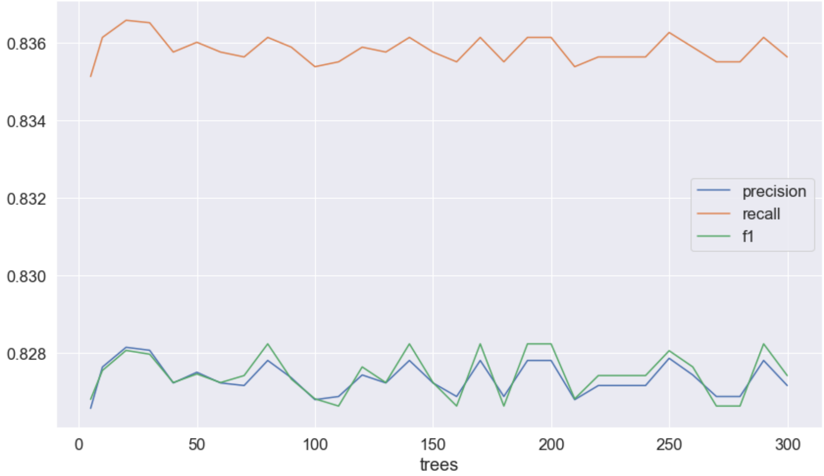


**Figure 5.4:** Variations of precision, recall and f1-score according to decision trees in the Random Forest

Also, tests were made regarding the function to measure the quality of a split, *entropy* and *gini*. However, there were no differences in the results between the two functions. It was used *entropy*.

### Neural Network

It was used $MLPClassifier$[7] estimator from *sklearn* to perform model with Neural Network. Figure 5.5 shows the variations of evaluation metrics according to activation function (*identity*, *logistic*, *tanh* or *relu*) used in the Neural Network, and it can be noted that

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`

the best results were achieved with *logistic* function. So, this is the activation function used. By default *relu* is the activation function used.

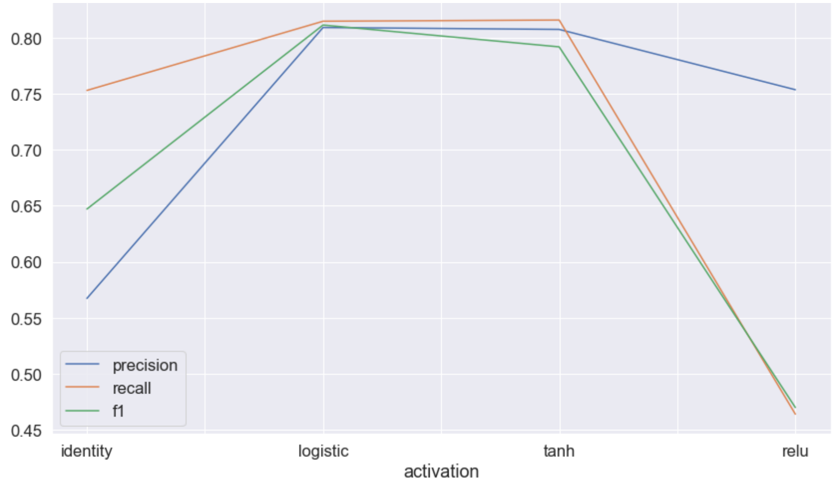


**Figure 5.5:** Variations of precision, recall and f1-score according to activation function used in Neural Network

In Figure 5.6 are presented the equation and the plot of each activation functions.

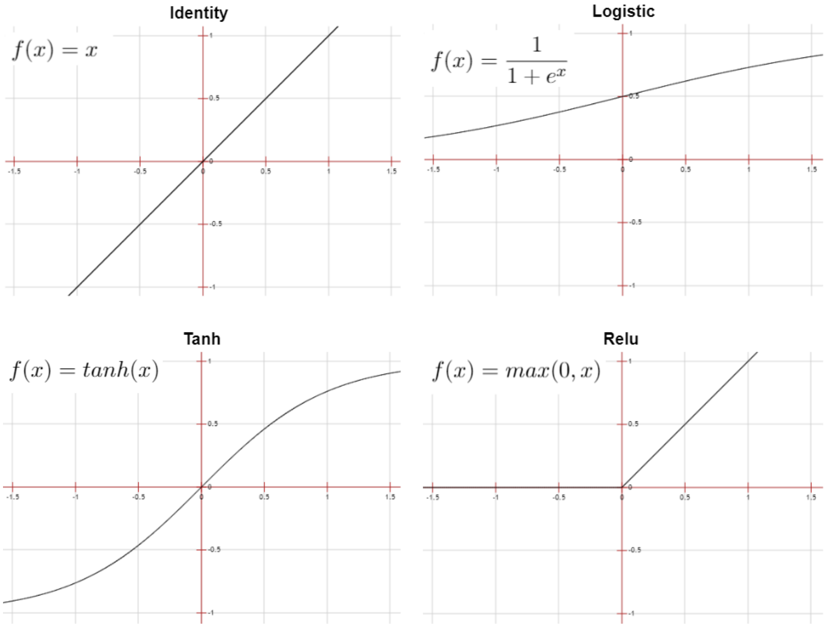


**Figure 5.6:** Activation function for the hidden layer

These functions have advantages and disadvantages and should be considered in order to understand what type of problem they are most suitable for:

- **Identity**: it generates a series of activation values, but as it is a linear function, it has a fixed derivative. It is useful to implement linear bottleneck [67];
- **Logistic**: as it uses a sigmoid function, it performs better for no linear problems. An advantage of this function is that it produces a value in the range of (0,1) when encountered with (- infinite, + infinite) as in the linear function [67];

- **Tanh**: it has a structure very similar to Sigmoid function, but this function is defined as (-1, + 1) [67];
- **Relu**: the rectified linear unit function has the same characteristics as the linear function on the positive axis. Having a value of 0 on the negative axis means that the network will run faster. The fact that the calculation load is less than the sigmoid and hyperbolic tangent functions has led to a higher preference for multi-layer networks [67].

Figure 5.6 shows that *logistic* and *tanh* have a better performance compared to *identity* and *relu*. One of the possible reasons for this can be the fact that these are not linear functions, which facilitates the learning process for this specific problem.

Tests were done with several hidden layers, from 1 layer to 5, and with different neurons. Either all layers with 50 neurons or all layers with 100 neurons. However, the best result was found with 3 hidden layers with 100 neurons each, which is the default value.

### K Neighbors Classifier

It was used $KNeighborsClassifier$[8] estimator from *sklearn* to perform the model with KNN. Figure 5.7 shows the variations of evaluation metrics according to neighbors (10 to 300) in the KNN, and it can be noted that from 50 neighbors the results do not have significant variation. The best result found was with 190 neighbors, then, it was chosen 191 neighbors to avoid confusion between two classes of data. The number of neighbors used by default is 5 and, in this case, this number is insufficient.

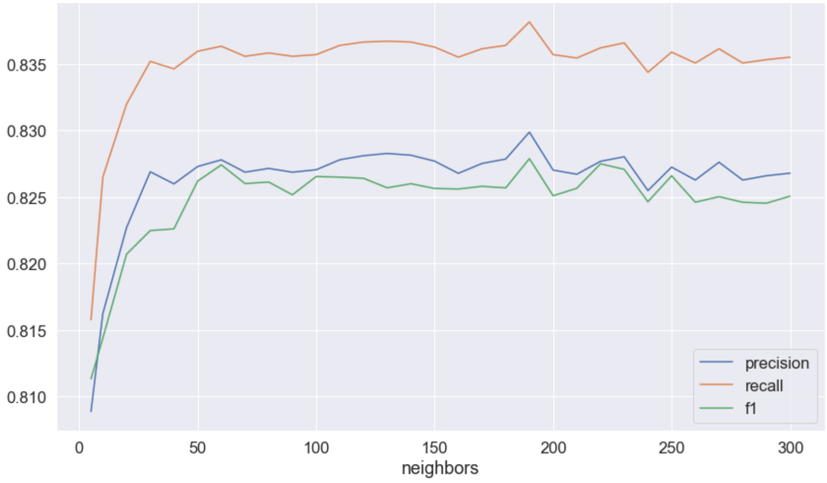


**Figure 5.7:** Variations of precision, recall and f1-score according to neighbors in the KNN

### Logistic Regression

It was used *LogisticRegression*[9] estimator from *sklearn* to perform the model with Logistic Regression. Figure 5.8 shows the variations of evaluation metrics according to solver

[8] `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

(*newton-cg*, *lbfgs*, *liblinea*, *sag* or *saga*) used in the Logistic Regression, and it can be noted that the best results are achieved with *newton-cg* algorithm, so, this the solver used. By default *lbfgs* is the algorithm used.

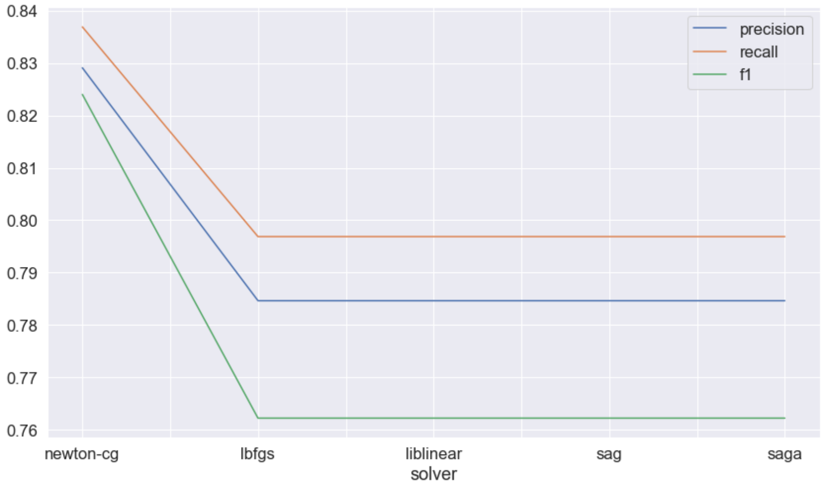


**Figure 5.8:** Variations of precision, recall and f1-score according to solver used in Logistic Regression

Regarding the library documentation, for small datasets, *liblinea* is a good choice, whereas *sag* and *saga* are faster for large ones. For multiclass problems, only *newton-cg*, *sag*, *saga* and *lbfgs* handle multinomial loss. Then, another advantage of *newton-cg* algorithm is its ability for both binary and multiclass classification.

## 5.4 Binary Classification - Crime vs. Non-Crime

Binary classification is the task of classifying the elements of a given dataset into two groups. As already stated, these groups are **Crime** and **Non-Crime**, that is, the aim is to predict if it will happen a crime or not.

After all pre-processing and massive observations on the data to increase understanding about it, now, it is time to start applying machine learning algorithms and observe the results.

### 5.4.1 Day and Night

Splitting the day in periods of day and night, the dataset has more samples of crimes than non-crimes, more precisely, 47800 of rows (75%) are crime and 15668 (25%) are non-crime, as shown in Figure 5.9. The disparity in the count of each class shows that it may be beneficial to proceed to the balancing of the data.

Table 5.1 presents the results of four algorithms executed for the *day and night* approach. After observing the results, some conclusions can be drawn. All algorithms have similar results, although Random Forest and KNN indicated to have a slightly higher performance. The results of these algorithms did not change significantly with the scaling as well as Logistic Regression, whereas, the results of Neural Network already had a more noticeable improvement. It can also be observed that the results got a little worse when the data is balanced. The

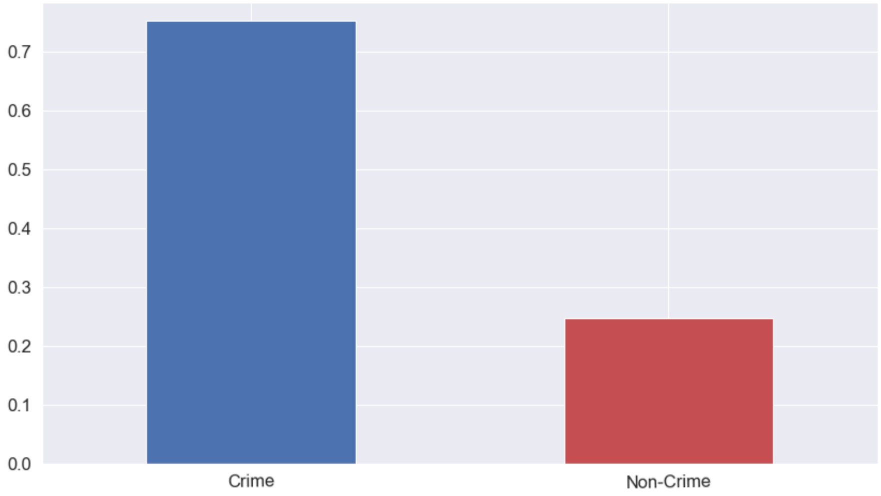


**Figure 5.9:** Normalized count of Crimes and Non-Crimes for *day and night* approach

justification is based on the number of crime samples, about 75%, that is, the vast majority. Then, it may have occurred overfitting and the model adjusted very well to the samples that were a crime, hence the precision and recall for this metric are almost 100% and influence the overall accuracy and recall. Consequently, after balancing the dataset, the results become more realistic. In terms of training time, Random Forest proved to be the fastest.

**Table 5.1:** Results of binary classification for *day and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **Precision** | 83% | 81% | 83% | 83% | 78% | 75% | 78% | 78% | 83% | 83% | 83% | 83% | 79% | 78% | 78% | 78% |
| **Recall** | 84% | 81% | 84% | 84% | 78% | 73% | 78% | 78% | 84% | 84% | 84% | 84% | 78% | 78% | 78% | 78% |
| **F1-Score** | 84% | 81% | 84% | 82% | 78% | 73% | 78% | 78% | 83% | 83% | 83% | 83% | 78% | 78% | 78% | 78% |
| **Time (s)** | 0.33 | 4.08 | 0.34 | 6.33 | 2.53 | 5.69 | 0.88 | 11.80 | 1.43 | 2.97 | 1.77 | 1.67 | 2.40 | 12.47 | 5.62 | 2.56 |

### 5.4.2 Morning, Afternoon and Night

Splitting the day in periods of morning, afternoon and night, the dataset has more samples of crimes than non-crimes, more precisely, 62021 of rows (65%) are crime and 33181 (35%) are non-crime, as shown in Figure 5.10. Compared to the *day and night* approach, now, the unbalanced is not so substantial.

Table 5.2 presents the results of four algorithms executed for the *morning, afternoon and night* approach. Again, the algorithms performed similarly. When the data was scaled, the algorithms that showed the best results at the recall level were Neural Network and Logistic Regression, and it can also be seen that with scaling and balanced data, the results are the same for all algorithms. Again, the results also got slightly worse when the data was balanced. It can also be noted that, in general, the results of all the algorithms fall slightly (about 3% to 4%). The reason for this is that, now, the day is divided into more parts, which adds more complexity.

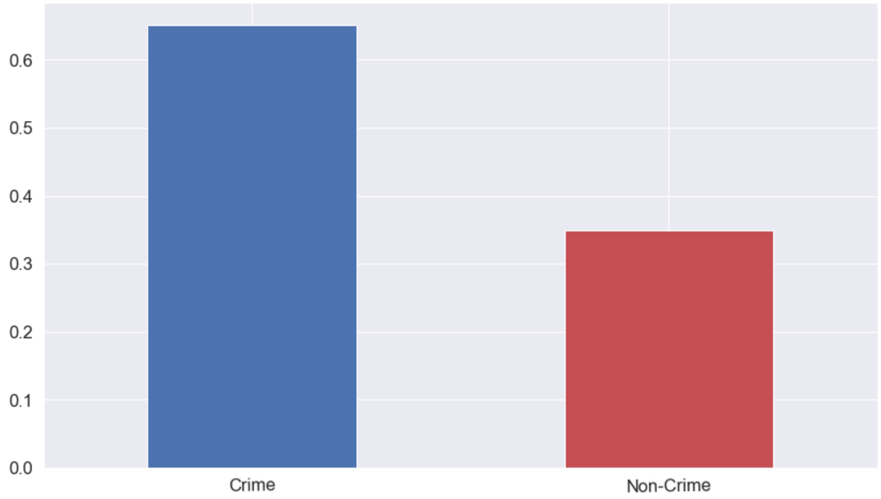


**Figure 5.10:** Normalized count of Crimes and Non-Crimes for *morning, afternoon and night* approach

**Table 5.2:** Results of binary classification for *morning, afternoon and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **Precision** | 77% | 73% | 77% | 77% | 75% | 71% | 75% | 75% | 77% | 77% | 77% | 77% | 75% | 75% | 75% | 75% |
| **Recall** | 78% | 74% | 78% | 78% | 75% | 69% | 75% | 75% | 76% | 78% | 77% | 78% | 75% | 75% | 75% | 75% |
| **F1-Score** | 77% | 73% | 77% | 77% | 75% | 69% | 75% | 75% | 77% | 77% | 76% | 77% | 75% | 75% | 75% | 75% |
| **Time (s)** | 2.43 | 6.23 | 11.24 | 0.83 | 4.05 | 7.45 | 1.22 | 16.02 | 3.36 | 16.80 | 5.38 | 2.77 | 5.18 | 21.62 | 8.65 | 3.84 |

### 5.4.3 Morning, Afternoon, Evening and Night

Splitting the day in periods of morning, afternoon, evening and night, the dataset has more samples of crimes than non-crimes, more precisely, 73167 of rows (58%) are crime and 53769 (42%) are non-crime, as shown in Figure 5.11. Compared to the *day and night* and *morning, afternoon and night* approaches, this is the approach where the data is more balanced.

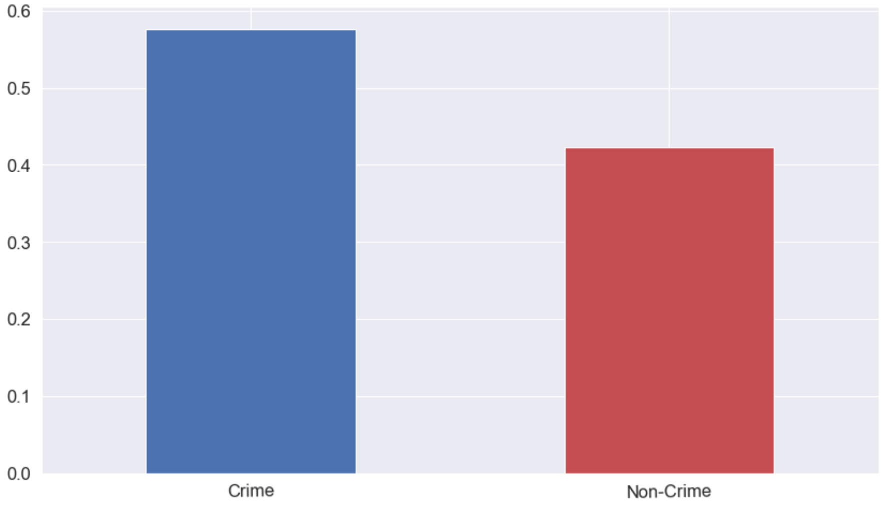


**Figure 5.11:** Normalized count of Crimes and Non-Crimes for *morning, afternoon, evening and night*

Table 5.3 presents the results of four algorithms executed for the *morning, afternoon,* *evening and night* approach. As in the two previous approaches, the algorithms performed similarly. In this case, the balancing of the data had little effect on the results, since, as already shown in Figure 4.14, the number of samples per class was already balanced. The improvement in the performance of the neural network with scaling was quite noticeable. On the other hand, KNN has the same result for all data transformations. It can also be noted that, in general, the results of all algorithms fall less in relation to the previous approach (about 1% to 2%).

**Table 5.3:** Results of binary classification for *morning, afternoon, evening and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **Precision** | 74% | 70% | 74% | 74% | 74% | 69% | 74% | 74% | 74% | 75% | 74% | 74% | 74% | 74% | 74% | 74% |
| **Recall** | 75% | 70% | 74% | 76% | 74% | 67% | 74% | 74% | 75% | 75% | 74% | 75% | 74% | 74% | 74% | 74% |
| **F1-Score** | 74% | 69% | 74% | 74% | 74% | 67% | 74% | 74% | 74% | 75% | 74% | 74% | 74% | 74% | 74% | 74% |
| **Time (s)** | 3.63 | 7.33 | 1.15 | 15.06 | 5.09 | 8.51 | 1.40 | 17.83 | 3.81 | 19.16 | 8.77 | 3.48 | 5.23 | 25.10 | 15.56 | 3.67 |

## 5.5 Multiclass Classification - Reduced vs. Moderated vs. High

Multiclass classification is the task of classifying the elements of a given dataset into three or more groups. As already stated, these groups are **Reduced**, **Moderated** and **High**. As in the binary classification, all algorithms will be executed for the different approaches in splitting the day. The results will be presented in tables to analyze.

### 5.5.1 Day and Night

Splitting the day in periods of day and night, the most frequent class is Moderated with 30036 rows (47%), followed by class High with 17764 rows (28%), and finally Reduced with 15668 rows (25%), as shown in Figure 5.12.

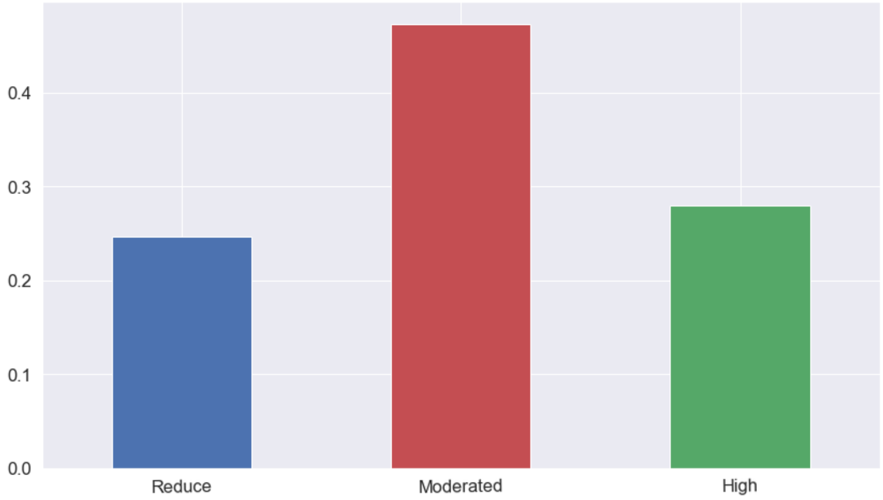


**Figure 5.12:** Normalized count of Reduced, Moderated and High of *day and night* approach

Table 5.4 presents the results of the four algorithms executed for the *day and night* approach. The results are quite similar across all algorithms. The increase of performance is notorious when data has been scaled on the Neuronal Network, about 11% of improvement. In the remaining algorithms, there were no significant differences.

**Table 5.4:** Results of multiclass classification for *day and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **Precision** | 74% | 67% | 74% | 74% | 73% | 65% | 73% | 73% | 74% | 74% | 74% | 74% | 73% | 73% | 73% | 73% |
| **Recall** | 72% | 63% | 72% | 72% | 72% | 63% | 71% | 71% | 72% | 73% | 72% | 72% | 72% | 75% | 71% | 71% |
| **F1-Score** | 72% | 64% | 72% | 71% | 72% | 64% | 72% | 72% | 72% | 72% | 72% | 71% | 72% | 72% | 72% | 72% |
| **Time (s)** | 1.42 | 4.25 | 0.32 | 19.85 | 2.90 | 4.25 | 0.62 | 31.45 | 1.50 | 7.20 | 1.83 | 4.06 | 2.50 | 11.80 | 11.80 | 6.09 |

### 5.5.2 Morning, Afternoon and Night

Splitting the day in periods of morning, afternoon and night, the most frequent class is also Moderated with 38709 rows (41%), but this time, followed by class Reduced with 33181 rows (35%), and finally High with 23312 rows (24%), as shown in Figure 5.13.

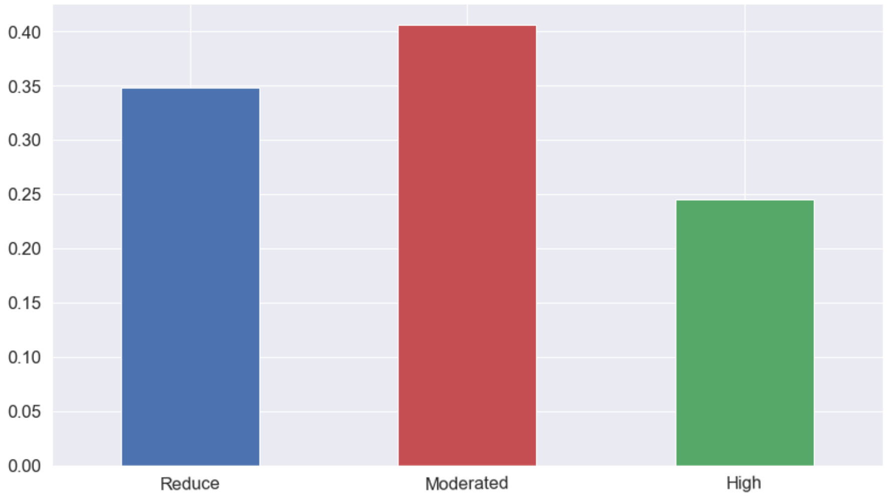


**Figure 5.13:** Normalized count of Reduced, Moderated and High of *morning, afternoon and night* approach

Table 5.5 presents the results of four algorithms executed for the *morning, afternoon and night* approach. Once again, the results are quite similar across all algorithms and the performance of Neuronal Network had about 15% of improvement with scaling. In the remaining algorithms, there were no significant differences.

**Table 5.5:** Results of multiclass classification for *morning, afternoon and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **Precision** | 71% | 63% | 72% | 72% | 70% | 60% | 71% | 70% | 71% | 71% | 72% | 72% | 70% | 71% | 71% | 70% |
| **Recall** | 69% | 60% | 71% | 69% | 69% | 58% | 71% | 68% | 69% | 69% | 71% | 69% | 69% | 69% | 68% | 69% |
| **F1-Score** | 69% | 60% | 69% | 69% | 70% | 60% | 69% | 69% | 69% | 69% | 69% | 69% | 70% | 69% | 69% | 69% |
| **Time (s)** | 2.44 | 4.61 | 0.63 | 31.00 | 3.34 | 4.81 | 0.84 | 36.88 | 2.59 | 12.27 | 3.74 | 7.01 | 3.56 | 16.10 | 6.74 | 8.49 |

### 5.5.3 Morning, Afternoon, Evening and Night

Splitting the day in periods of morning, afternoon, evening and night, the most frequent class is also Reduced with 53769 rows (42%), followed by class Moderated with 40054 rows (32%), and finally High with 33113 rows (26%), as shown in Figure 5.14.

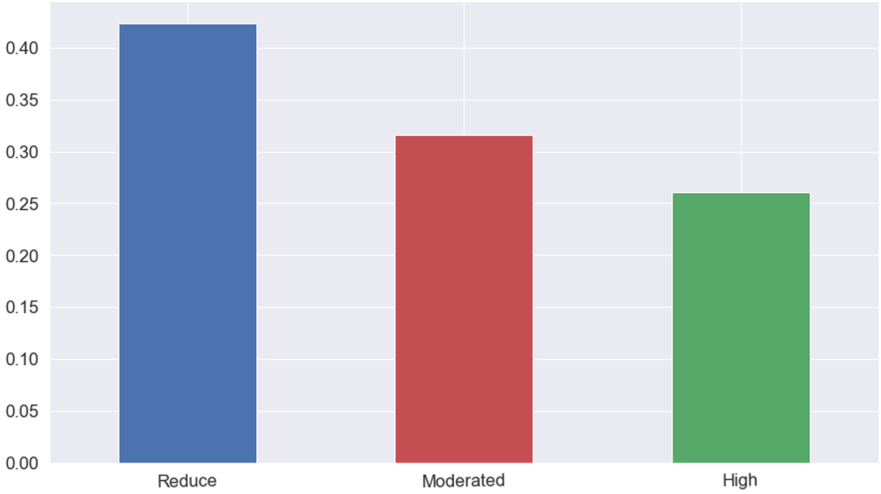


**Figure 5.14:** Normalized count of Reduced, Moderated and High of *morning, afternoon, evening and night* approach

Table 5.6 presents the results of the four algorithms executed for the *morning, afternoon, evening and night* approach. As with the two previous approaches, the neural network improves the results with scaling, and in the remaining algorithms, there were no significant differences.

**Table 5.6:** Results of multiclass classification for *morning, afternoon, evening and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **Precision** | 67% | 56% | 66% | 66% | 67% | 53% | 66% | 66% | 66% | 66% | 65% | 66% | 66% | 66% | 66% | 66% |
| **Recall** | 67% | 55% | 66% | 67% | 66% | 53% | 65% | 66% | 66% | 66% | 65% | 66% | 66% | 66% | 65% | 66% |
| **F1-Score** | 67% | 54% | 66% | 66% | 66% | 52% | 66% | 66% | 66% | 66% | 65% | 65% | 66% | 65% | 66% | 66% |
| **Time (s)** | 3.60 | 6.26 | 1.08 | 45.10 | 5.05 | 6.34 | 1.51 | 57.10 | 3.55 | 20.09 | 8.20 | 9.41 | 5.76 | 28.50 | 14.00 | 12.87 |

## 5.6 Chosen Model

Regarding the classification task, the split of the day chosen is *morning, afternoon, evening and night*. Although the results for this split have shown some decrease in performance comparing to the other approaches, it has a great advantage, which is the increase of detail for the period of time chosen, which in turn will be more useful for the final user. As stated by Rumi *et al.* in [65], crime prediction in finer temporal grain will help the police to design their patrol strategy dynamically, and it will increase the probability to reduce crime rate more effectively.

The multiclass classification was chosen, that is, the risk will be measured as Reduced, Moderated or High.

Although all algorithms have demonstrated similar performances, random forest algorithm was chosen, because it performed slightly better. Overall, in the selected model, data will not be scaled since the random forest is not sensitive to feature scaling. Also, the data will not be balanced since in the chosen *morning, afternoon, evening and night* approach, all classes are reasonably balanced.

Considering the frequency of each crime risk as a baseline, a comparison can be made with the results obtained by the chosen model and, thus, verify the improvement achieved. For example, 42% of records belong to Reduced class, so, if the model always predicts Reduced, the precision will be 42%, since out of all the predicted classes as Reduced just 42% are actually positive. The model has a precision of 69% regarding the Reduced class, consequently, it was achieved an improvement of 65% over the baseline. As can be seen in Table 5.7, considering all categories with the ML model, an improvement of 110% was obtained.

It is also important to note that the High category is the one with the most marked improvement at about 178%. From the user's point of view, this can be very appropriate as the main concern is to anticipate locations with a high criminal risk.

**Table 5.7:** Improvements achieved with the ML model compared to the baseline

| Risk | Baseline | Precision | ML Model Improvement |
|---|---|---|---|
| *Reduced* | 42% | 69% | 65% |
| *Moderated* | 32% | 60% | 88% |
| *High* | 26% | 72% | 178% |
| Overall | 33% | 67% | 110% |

Figure 5.15 shows a representation of the beginning of the Random Forest used in the model. In each node, it can be seen which feature is used for the decision, the entropy value, the number of samples and the respective class.

It was also examined the importance of each feature in the construction of random forest and it was noted that the population and income were among the most important features as well as the period of the day. On the other hand, the features related to the day of the week proved to be less important. In fact, these values make sense as was said throughout this document, for example, in Figure 3.3, it was seen that there was no great variation in crimes compared to the day of the week, yet, it was noted that there were significant variations in the frequency of crimes according to the period of the day, as seen in Figure 4.14.

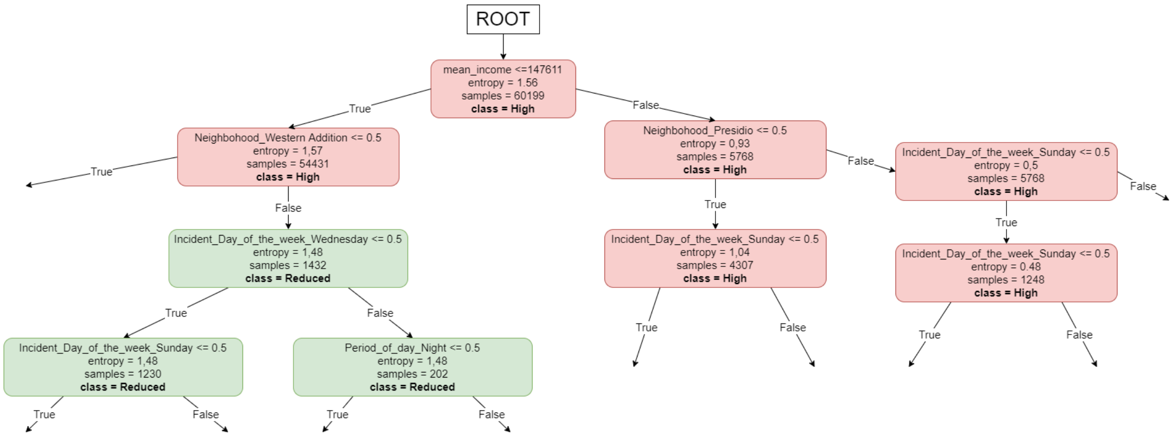


**Figure 5.15:** Begin of Random Forest in Classification Model

## 5.7 Summary

In this chapter, the various approaches were tested with the several algorithms to find the best classification model. Thus, models with scaled and non-scaled, and balanced and unbalanced data were tested.

Before applying each algorithm to learn the model, there was a concern to find the best value for hyperparameters that add performance.

The best results were obtained in the binary classification, which goes according to the initial expectations, since the multiclass classification is a more complex problem to solve. However, with this type of classification, it can be obtained a more detailed prediction in relation to binary classification.

The algorithm with the best performance is Random Forest with 110% of improvement over the baseline.

CHAPTER 6

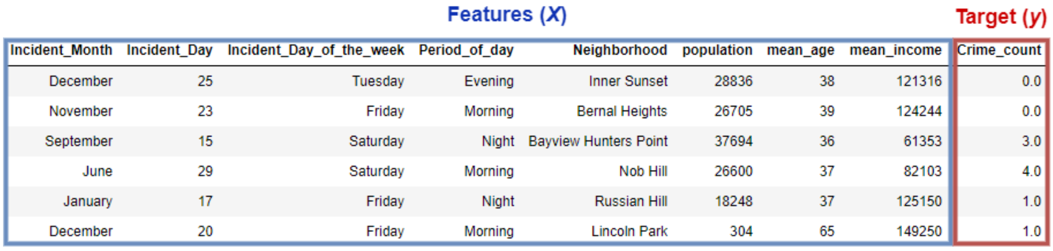
# Regression Model

This chapter describes the entire process in the development of a regression model. The aim is to predict the number of crimes, it means, predict a numerical variable and not a class.

The evaluation metrics of the models will be described; in this case, they will be the mean absolute error and the mean squared error. The same algorithms will be used as in the classification model, but this time adapted for a regression task. In order to make the most of these hyperparameters tuning will be done. Finally, the results obtained for the various tested approaches will be presented and discussed.

## 6.1 Defining Model Evaluation Rules

Data preparation is mainly similar to the classification model, although there are some minor differences. One of them is the definition of a numeric target, and as can be seen in Figure 6.1, in this time the count of crimes is the target of the regression model. Later, the *Incident_Month* and *Incident_Day* were not used because it was found that these features would not add value, and would even decrease the performance of the model.



Features (X)
Target (y)

| Incident_Month | Incident_Day | Incident_Day_of_the_week | Period_of_day | Neighborhood | population | mean_age | mean_income | Crime_count |
|---|---|---|---|---|---|---|---|---|
| December | 25 | Tuesday | Evening | Inner Sunset | 28836 | 38 | 121316 | 0.0 |
| November | 23 | Friday | Morning | Bernal Heights | 26705 | 39 | 124244 | 0.0 |
| September | 15 | Saturday | Night | Bayview Hunters Point | 37694 | 36 | 61353 | 3.0 |
| June | 29 | Saturday | Morning | Nob Hill | 26600 | 37 | 82103 | 4.0 |
| January | 17 | Friday | Night | Russian Hill | 18248 | 37 | 125150 | 1.0 |
| December | 20 | Friday | Morning | Lincoln Park | 304 | 65 | 149250 | 1.0 |

**Figure 6.1:** Split descriptive columns from the numeric target column

The evaluation metrics for a regression model are different from the evaluation metric of a classification model. In this case, it is aimed to measure the error rate between the real values and the model prediction values, as can be seen in Figure 6.2, where predictions are represented by a red line and observed values by blue points.

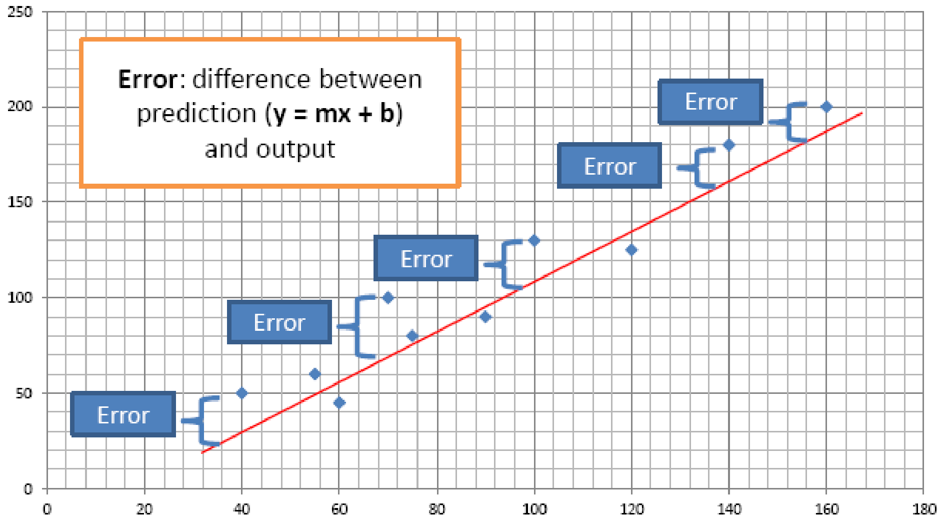


**Figure 6.2:** Absolute difference between the predicted values and observed values

The following evaluation metrics are used.

### Mean Absolute Error

The Mean Absolute Error (MAE) measures the absolute difference between the predicted values and observed values of the target feature. The absolute value of the difference from the predicted values and known values are taken, and then divided by the number of observations in the dataset. It measures the average magnitude of the error. The values of the mean absolute error can range from 0 to infinite: the lower is the value the more accurate will be the algorithm. MAE is evaluated by the Equation 6.1 which represents the average squared difference between the observed values and what is predicted.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_{pred} - x_{obs}| \tag{6.1}$$

### Mean Squared Error

The Mean Squared Error (MSE) measures the average squared difference between the observed values and what is predicted. The squaring is necessary to remove any negative signs, and it also gives more weight to larger differences. The smaller the means squared error, the closer the model is to find the line of best fit. It is defined by Equation 6.2:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_{pred} - x_{obs})^2 \tag{6.2}$$

As in the evaluation of the classification model, it was used *cross_validate* method from *scikit* with four folds, and the final results comprise the mean of results of folds.

## 6.2 Hyperparameter Tunning

Since the algorithms used in the classification model can also be used for regression, it was decided to use them for this task, but, at this time the algorithms are facing a regression task. It was decided to check which parameters would be used to optimize performance again.

### Random Forest

It was used $RandomForestRegressor$[1] estimator from *sklearn* to perform the model with Random Forest. Figure 6.3 shows the variations of evaluation metrics according to the number of trees in the Random Forest (10 to 300), and it can be noted that, from 10 to 300 decision trees, the variation of results are non-significant. Thus, 30 decision trees will be used as in the regression algorithm.

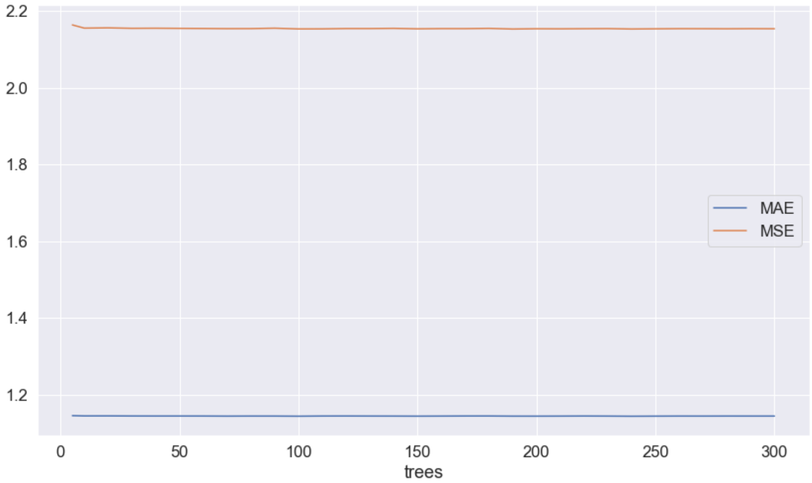


**Figure 6.3:** Variations of MAE and MSE according to decision trees in the Random Forest

### Neural Network

It was used $MLPRegressor$[2] estimator from *sklearn* to perform the model with Neural Network. Figure 6.4 shows the variations of evaluation metrics according to activation function (*identity*, *logistic*, *tanh* or *relu*) used in the Neural Network, and it can be noted that the best results were achieved with *tanh* and *logistic* functions. However, *tanh* has shown to slightly outperform; therefore, this activation function used and not *logistic* as in regression model. By default *relu* is the activation function used.

Also, tests were made regarding the function to measure the quality of a split, *MSE* and *MAE*. However, there were no differences between the two functions. so it was chosen *MSE*.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html`

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html`

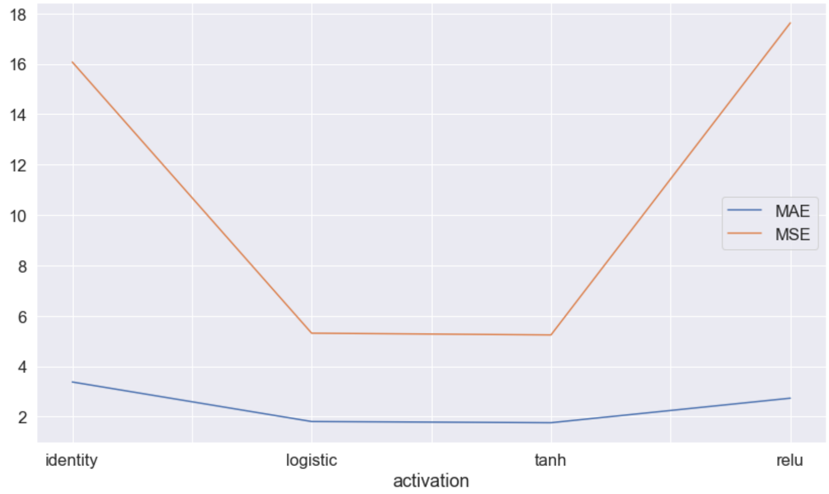


**Figure 6.4:** Variations of MAE and MSE according to activation function used in Neural Network

### K Neighbors Regressor

It was used $KNeighborsRegressor$[3] estimator from *sklearn* to learn the model with KNN. Figure 6.5 shows the variations of evaluation metrics according to neighbors (10 to 300) in the KNN, and it can be noted that, from 50 neighbors, the results do not have significant variation. The best result found is with 80 neighbors, then, it was chosen this number of neighbors. There was a change from 191 to 81 neighbors compared to the classification model.

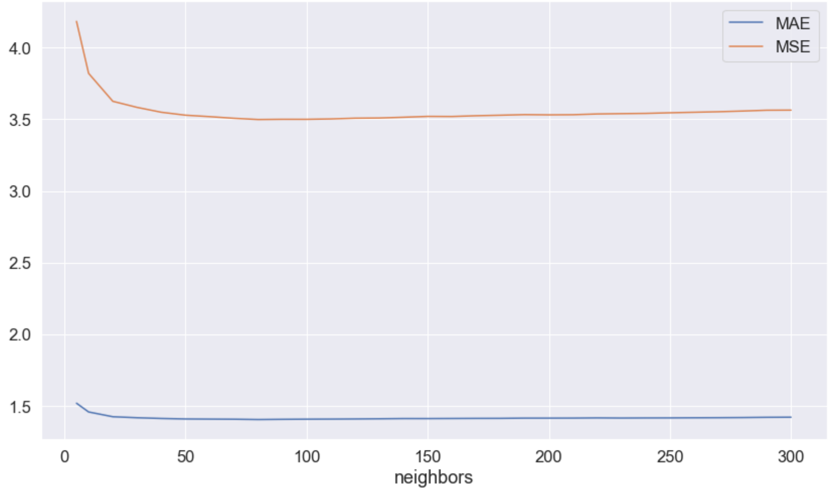


**Figure 6.5:** Variations of MAE and MSE according to neighbors in the KNN

### Logistic Regression

For Logistic Regression, the same estimator was used as in the classification task. Figure 6.6 shows the variations of evaluation metrics according to solver (*newton-cg*, *lbfgs*, *liblinea*, *sag* or *saga*) used in the Logistic Regression, and it can be noted that the best results are

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html`

achieved with *newton-cg* algorithm (equal to the regression model), therefore, this is the solver used. By default *lbfgs* is the algorithm used.

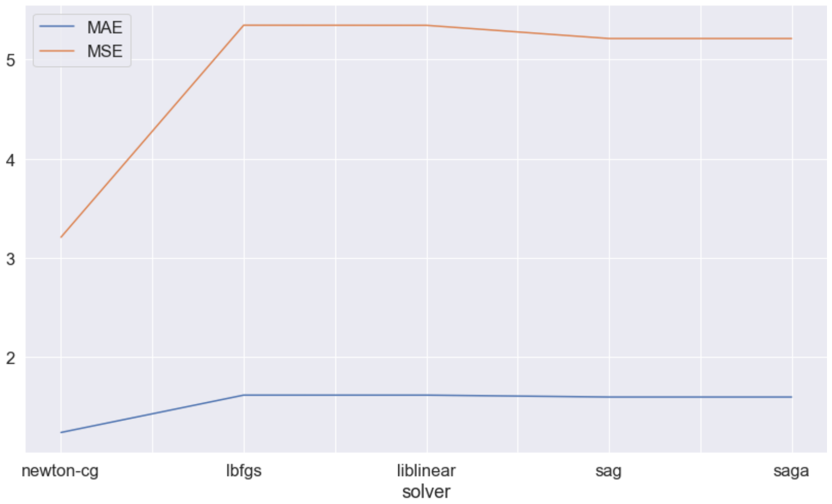


**Figure 6.6:** Variations of MAE and MSE according to solver used in Logistic Regression

## 6.3 Results

It was considered 3 approaches to divide the day. Next, the results of the algorithms for each one will be presented.

### 6.3.1 Day and Night

Figure 6.7 shows the distribution of the number of crimes in the *day and night* approach; it also presents some annotations regarding the percentiles. It can be noted that 90% of the samples have a crime count equal to or less than 10. Thus, it can be considered that the higher values are outliers, that is, they are values that are unusual and that can cause anomalies in the results obtained by algorithms. In this sense, samples with a crime count greater than 10 were not considered.

Table 6.1 presents the results of the four algorithms executed for the *day and night* approach. In general, all algorithms had similar performances, which decreases when data balancing is made. Neural Network has a notorious improvement with data scaling, and Logistic Regression is the algorithm that takes the longest to train.

**Table 6.1:** Results of regression for *day and night*

| | **Without Scaling** | | | | | | | | **With Scaling** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Unbalanced** | | | | **Banlaced** | | | | **Unbalanced** | | | | **Banlaced** | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **MAE** | 1.41 | 1.78 | 1.41 | 1.53 | 1.72 | 2.80 | 1.95 | 2.00 | 1.41 | 1.41 | 1.19 | 1.53 | 1.72 | 1.72 | 1.97 | 2.00 |
| **MSE** | 3.54 | 5.10 | 3.54 | 5.09 | 4.50 | 10.72 | 5.76 | 7.31 | 3.54 | 3.54 | 1.41 | 5.10 | 4.50 | 4.52 | 5.88 | 7.32 |
| **Time (s)** | 2.61 | 4.17 | 0.22 | 66.10 | 14.22 | 8.58 | 2.12 | 208.73 | 2.78 | 5.99 | 3.55 | 13.44 | 14.45 | 63.54 | 14.56 | 54.72 |

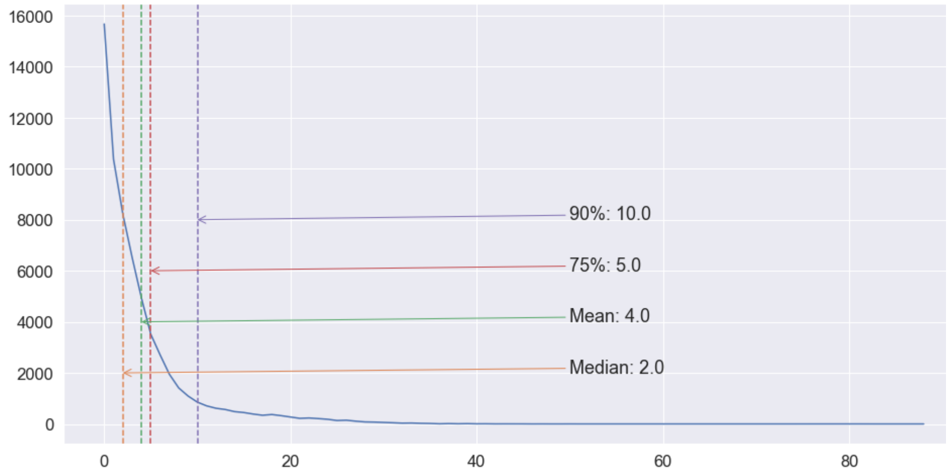


**Figure 6.7:** Distribution of crime count in the *day and night* approach

### 6.3.2 Morning, Afternoon and Night

Figure 6.8 shows the distribution of the number of crimes in the *morning, afternoon and night* approach. It can be noted that 90% of the samples have a crime count equal to or less than 7; for the same reason presented in approach *day and night*, samples with a crime count greater than 7 were not considered.

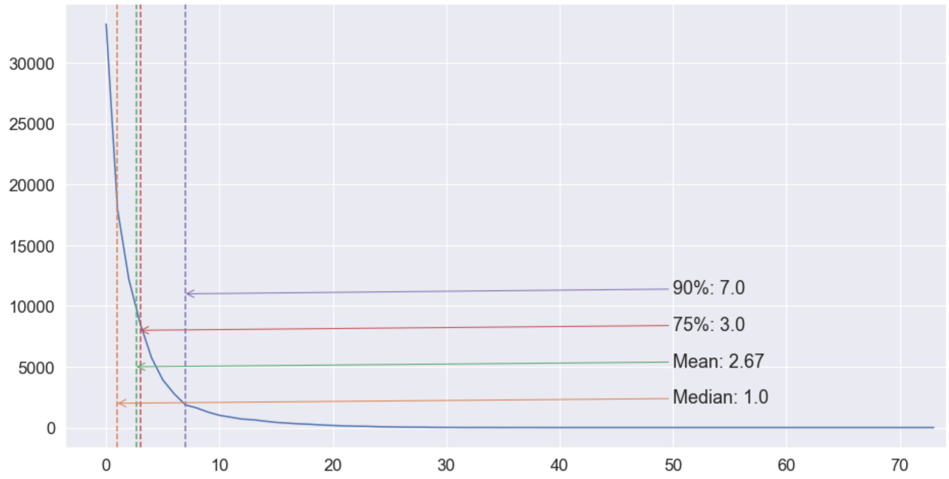


**Figure 6.8:** Distribution of crime count in the *morning, afternoon and night* approach

Table 6.2 presents the results of the four algorithms executed for the *morning, afternoon and night* approach. Once again, all algorithms had similar performances; which decreases when data balancing was made; Neural Network had a notorious improvement with data scaling. It can be noted that the performance of the algorithms has improved slightly compared to the previous approach.

**Table 6.2:** Results of regression for *morning, afternoon and night*

| | Without Scaling | | | | | | | | With Scaling | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unbalanced | | | | Banlaced | | | | Unbalanced | | | | Banlaced | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **MAE** | 1.09 | 1.34 | 1.09 | 1.19 | 1.41 | 1.97 | 1.59 | 1.59 | 1.09 | 1.09 | 1.10 | 1.19 | 1.41 | 1.41 | 1.63 | 1.59 |
| **MSE** | 2.12 | 2.95 | 2.12 | 3.36 | 2.99 | 5.58 | 3.76 | 4.66 | 2.12 | 2.12 | 2.11 | 3.38 | 2.99 | 3.00 | 4.01 | 4.66 |
| **Time (s)** | 4.81 | 4.38 | 0.48 | 86.63 | 26.02 | 11.16 | 5.29 | 294.70 | 5.11 | 12.50 | 12.50 | 16.53 | 25.78 | 73.38 | 41.44 | 77.09 |

### 6.3.3 Morning, Afternoon, Evening and Night

Figure 6.9 shows the distribution of the number of crimes in the *morning, afternoon, evening and night* approach. It can be noted that 90% of the samples have a crime count equal to or less than 6; for the same reasons, presented in the previous approaches, samples with a crime count greater than 6 were not considered.

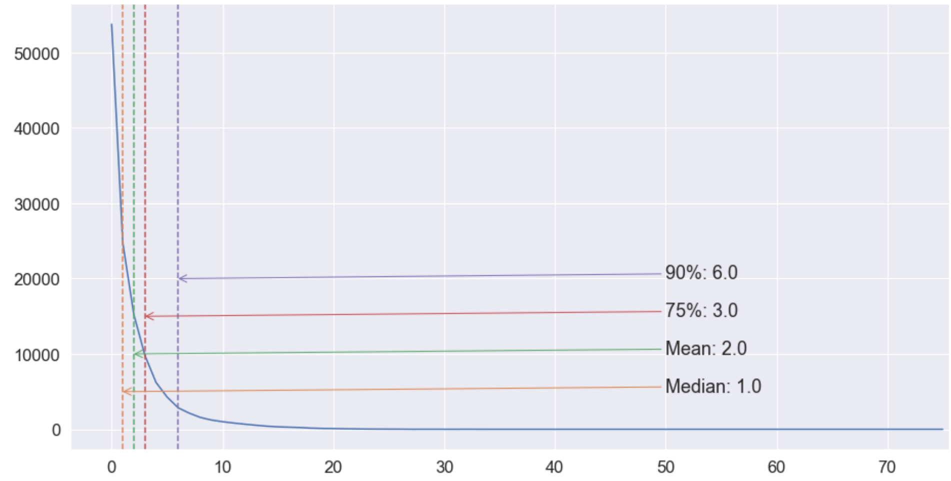


**Figure 6.9:** Distribution of crime count in the *morning, afternoon, evening and night* approach

Table 6.3 presents the results of the four algorithms executed for the *morning, afternoon, evening and night* approach. As in previous approaches, performance is identical which decreases when data balancing was made, and Neural Network had a notorious improvement with data scaling.

**Table 6.3:** Results of regression for *morning, afternoon, evening and night*

| | Without Scaling | | | | | | | | With Scaling | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unbalanced | | | | Banlaced | | | | Unbalanced | | | | Banlaced | | | |
| | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR | RF | NN | KNN | LR |
| **MAE** | 0.94 | 1.18 | 0.94 | 0.98 | 1.30 | 1.82 | 1.45 | 1.37 | 0.94 | 0.95 | 0.94 | 0.97 | 1.30 | 1.28 | 1.47 | 1.41 |
| **MSE** | 1.61 | 2.20 | 1.61 | 2.51 | 2.49 | 4.51 | 3.06 | 3.57 | 1.61 | 1.60 | 1.61 | 2.49 | 2.49 | 2.43 | 3.00 | 3.74 |
| **Time (s)** | 7.42 | 5.23 | 0.66 | 94.34 | 45.81 | 8.61 | 10.27 | 10.24 | 7.86 | 15.37 | 5.57 | 21.65 | 43.20 | 116.41 | 107.03 | 85.99 |

## 6.4 Chosen Model

Regarding the regression task, the day split chosen was *morning, afternoon, evening and night* as well as in classification, the reasons are the same, and the algorithm chosen is again the Random Forest. In this case, the function to measure the quality of a split is Mean Squared Error and not entropy as in the classification model.

Figure 6.10 shows a representation of the beginning of the Random Forest used in the model. In each node it can be seen which feature is used in the decision, the MSE value, the number of samples and the respective value (number of crimes).

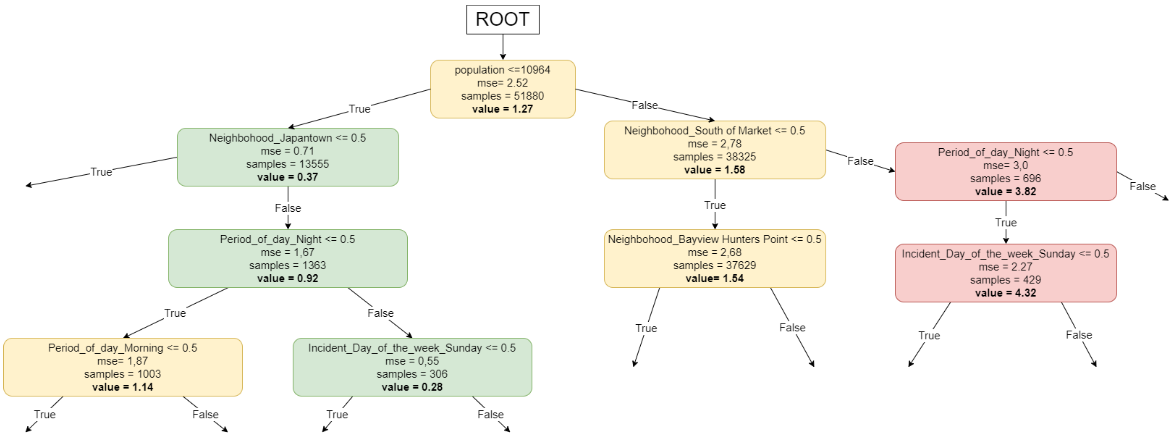


**Figure 6.10:** Begin of random forest in regression model

The most important features for the construction of the regression Random Forest are the same ones used in the random forest classification, which would be expected.

In both models, RF was chosen since it was the one that presented the best results in comparison with the other algorithms. This can be justified by several innate characteristics in RF, such as methods for balancing error in class population unbalanced datasets and it can handle non-linear parameters efficiently, thus, non-linear parameters do not affect the performance of a RF, contrasting with others algorithms[4].

It is also important to note that it is a faster algorithm because it works only on a subset of features, so it can easily work with hundreds of features. Prediction speed is significantly faster than training speed because it can save generated forests for future uses.

## 6.5 Summary

In this chapter, the various approaches were tested with several algorithms to find the best regression model.

As was evident in the results of these approaches, the shorter the period, the better are the results. This can be explained based on the count of crimes in shorter periods which is less dispersed; for example, in the *day and night* approach, the count goes from 0 to 10 while in the *morning, afternoon, evening and night* approach, it goes from 0 to 6. So, it is easier for the algorithm to predict with more exactness.

The algorithm with the best performance is Random Forest with a mean absolute error less than 1.

[4] `https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706`

CHAPTER 7

# Model Deploy and Application

This chapter describes the deployment of the classification and regression models in the system, where several entities can interact with it. In this stage the model is incorporated into the organization processes. Also, it will be shown an example of a client application that makes requests to the model and through an API, presents the results in a map of San Francisco.

## 7.1 Model Deployment and API Creation

After finding the best model that meets the initial requirements, it is necessary to deploy it. For this approach it is used *joblib*[1] that saves the model as a binary object to disk, enabling its usage in an application.

It was developed a back-end application in *Django*, a high-level Python Web framework[2], that contains the models of classification and regression. This application also retrieves daily incidents of the San Francisco Open Data, through the Representational State Transfer (REST) API provided, and stores it in big data warehouse. To keep the predictions as reliable as possible, it is appropriate to learn the model weekly with the most recent data. In this case, it will be done a batch processing, since the data will be stored on the platform daily with the incident records of this day, and only after accumulating a week of data; then, it will these be analyzed.

This application also provides an API that allows other entities sending a request and obtaining a prediction. The high-level architecture of the service logic explained is shown in Figure 7.1.

The API provides the following endpoint:

[1] https://joblib.readthedocs.io/en/latest/

[2] https://www.djangoproject.com/

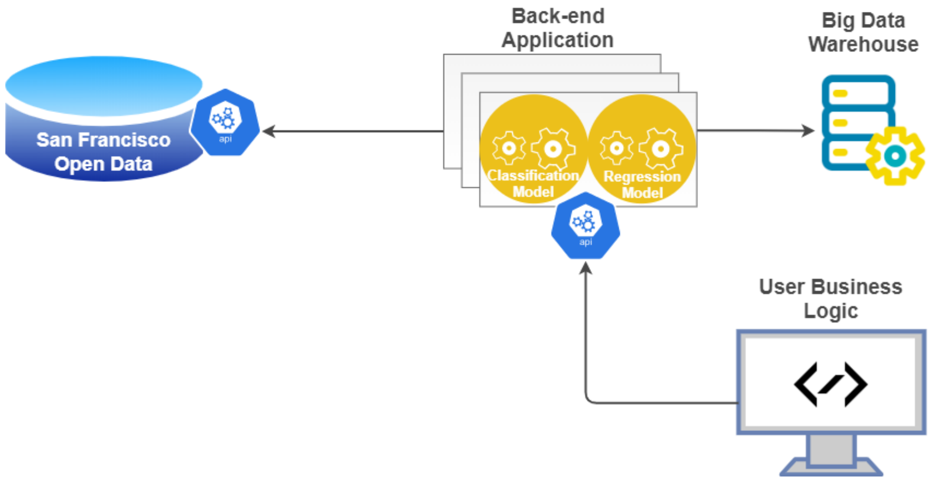


**Figure 7.1:** High-level architecture of service logic

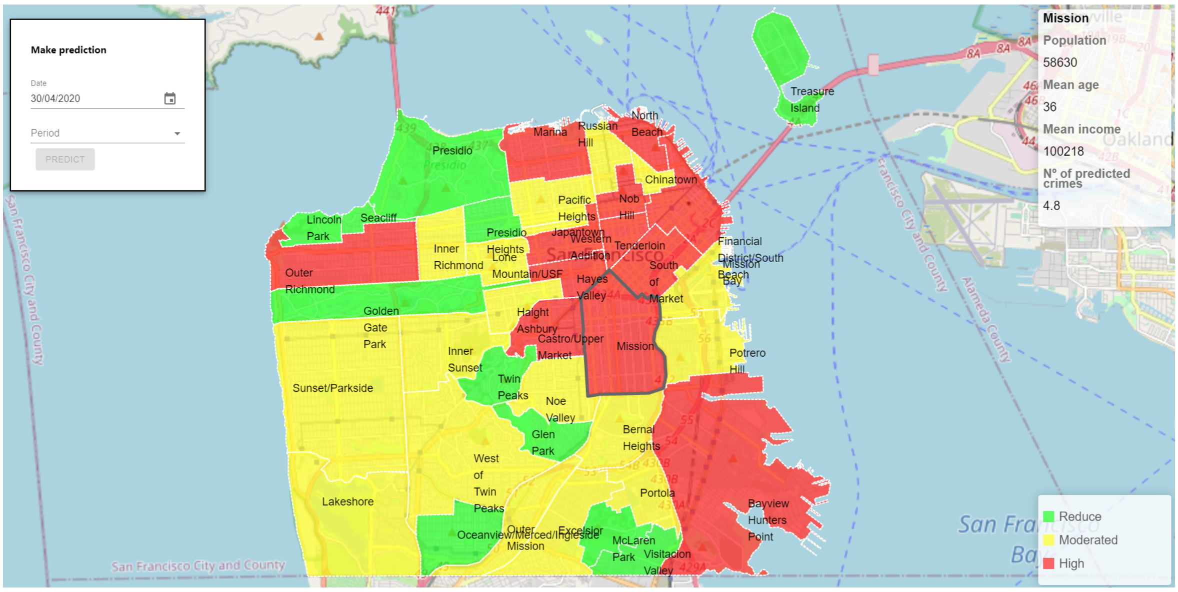
- **GET $url/api/predictions**: when this endpoint is called, a JSON response is returned with predictions for the current date and time for all neighborhoods. In this case, the period of the day and the day of the week are considered in the server-side according to the current date and time
- **GET $url/api/predictions/date/yyyy-mm-dd/period/$period_of_the_day**: this endpoint should be used with the aim to specify the day and period of the day of the prediction, for example, for tomorrow night. The JSON response will return the prediction for all neighborhoods.
- **GET $url/api/predictions/neighborhood/$neighborhood/date/yyyy-mm-dd/period/$period_of_the_day**: this endpoint, in addition to specifying the day and period of the day, also allows to indicate the neighborhood, consequently the JSON response will only return the forecast for that neighborhood.

In the example below, there is a JSON response with the prediction for all neighborhoods. As can be seen, for each neighborhood, a qualitative prediction of the risk of crime and a quantitative prediction of the number of crimes are made.

```
{
   "Bayview Hunters Point": {
      "classification": "High",
      "regression": 3.8
   },
   "Bernal Heights": {
      "classification": "Moderated",
      "regression": 1.4
   },
   ...
   "Glen Park": {
      "classification": "Reduce",
      "regression": 0.4
   }
}
```

## 7.2 Client Application

In order to have a graphical interface in which any user could easily consult the crime risk prediction for San Francisco, it was integrated a front-end application developed in React, a JavaScript library for building User Interfaces (UIs)[3]. This application makes requests to API provided by the back-end application and shows the results iteratively on a map using Leaflet, an open-source JavaScript library[4].



Make prediction

Date: 30/04/2020

Period

PREDICT

Mission

Population: 58630

Mean age: 36

Mean income: 100218

Nº of predicted crimes: 4.8

🟩 Reduce

🟨 Moderated

🟥 High

**Figure 7.2:** Front-end application that shows the prediction for San Francisco on the map

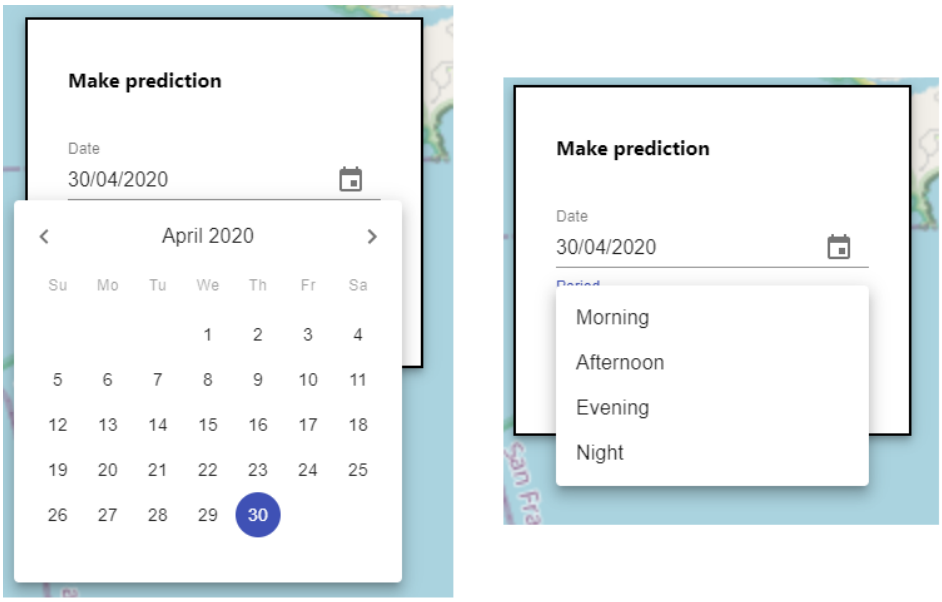
When loading the page shown in Figure 7.2, the Client application fetches the current prediction from the back-end and shows it over the Map. To give an overview of the panorama, each neighborhood is given a color according to the risk of crime for the current moment. As can be seen on the label in the lower right corner, if the crime risk is Reduced, the assigned color is **green**; if the crime risk is Moderate, the assigned color is **yellow**; and if it is High, the assigned color is **red**.

In the upper right corner, more information about the selected neighborhood is shown. To give more detail about a neighborhood's criminal prediction, in addition to demographics, the estimated number of crimes is shown.

To select another date to make a prediction, the user can use the widget in the upper left corner that allows the choose the data and period of the day, as shown in Figure 7.3. The user cannot choose a date greater than a week, since it is necessary to retrain the model weekly with the most recent data. If the user chooses a date before today, it will return the real result instead of prediction.

[3] https://reactjs.org/

[4] https://leafletjs.com/



Make prediction

Date

30/04/2020

April 2020

| Su | Mo | Tu | We | Th | Fr | Sa |
|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | | |

Make prediction

Date

30/04/2020

Period

Morning

Afternoon

Evening

Night

**Figure 7.3:** Interface to choose date and period of the day to make a prediction

## 7.3 Summary

In this chapter, it was described the model deployed in a back-end application, which in turn provides a REST API which enables other entities to communicate with.

Finally, a client application was integrated, which in this case shows an iterative map of San Francisco with criminal predictions in a "user-friendly" way. Other types of client applications could also have been integrated, such as a mobile application or another back-end application.

CHAPTER 8

# Conclusions and Future Work

Throughout the dissertation, several conclusions were drawn according to the results obtained. However, this chapter aims to make a general reflection on all the work developed with a focus on the challenges that were faced and how they were solved.

## 8.1 Conclusions

The work presented in this dissertation aimed to develop a solution that, through Machine Learning techniques, could be a contribution to the improvement of public safety in a city, since it is a major concern in cities around the world.

Thus, this work involved several stages since the data collection, where the best sources of data were searched and identified; data understanding where raw data has been converted into a clean dataset; modeling where several algorithms were performed to find the best model; and finally the model deploy where the model was deployed in a real system.

Like any project of this nature, there is an essential component of research on work already carried out under the same theme. Some authors stated that crime event prediction is a challenging task and it is extremely important to have the right data in order to obtain satisfactory results. This was the first big challenge, finding a dataset that really had the necessary quality to be able to work on it. As stated by Bryan Landerman, Amazon Web Services Enterprise Strategist, machines do not make decisions by instinct, all decisions are based on the training performed and the data provided. This justifies why it is necessary to provide the right data for learning. Otherwise, like humans without proper training, the model will be ill-prepared to make the right decisions.

Indeed, it is notorious the influence of the preparation and transformation of data in the results of the model. Namely, in the splitting of the day in several periods and in the two types of classification carried out, binary and multiclass. In spite of the performance measurements of the models, such as accuracy or recall, are extremely important to assess whether a model is fit for production or not, it is also very important to have a critical view of the results. In the present work it was verified that, with longer time interval periods of the day and with binary classification, the results obtained were naturally better. However, from the point of view of the business, or in other words, the point of view of the user who will use this application, this approach was not efficient due to having less temporal detail. So, there is a trade-off between having more detail in the multiclass prediction and losing some exactness in the results, or having a binary prediction, which is more high-level but with more exactness. There is no rule that says the perfect solution, each case is different and must be analyzed taking into account the final objective.

In order to enrich the developed solution, both classification and regression models were trained. Despite using the same data for both models, there were aspects to be taken into account. In addition to changing the target, in data pre-processing it was necessary to handle the outliers in the regression model, to prevent that they influence negatively the results. On the other hand, in the classification model it is not necessary to do this treatment, since even if a sample has a very high number of crimes, the label will still be high.

In the classification model, there was an improvement in the precision of prediction of the criminal category by around 110% compared to baseline, which is a good result since with the developed ML model it is possible to make a prediction of the risk of crime with more than double the precision.

In the regression model, the mean absolute error is less than 1, which is considered a low error, so a good result was also achieved.

Then, taking into account the results, it can be concluded that the general objective of providing a solution based on machine learning techniques to improve public safety through the prediction of criminal events has been achieved.

This solution is developed for the city of San Francisco, since the data used belonged to it. However, with the required training, the solution can be adapted to other cities if similar data is made available.

## 8.2 Future Work

In a machine learning project, there is always a margin for progression, since small changes in some steps, such as pre-processing, could reflect in better results in the end.

A scenario that could be adopted is the prediction of the crime category, since this information is present in the dataset. It was developed a solution for this scenario, however, the final results were not satisfactory, given that for the same location and time interval, crimes from different categories occurred. To improve the results, the approach of Rumi *et al.* was followed in order to try to integrate dynamic data in the dataset, such as human mobility data, road traffic, sports and cultural events between other sources of data that could add value and quality, but this type of data is not available to the general public, so it was not possible to obtain it. In short, this would imply refining the model through new data sources and evolving the rationale of the desired target. Thus, this is one of the main topics for future work, adding more data heterogeneity.

Other points that should be implemented or improved are:

- Implement models with other algorithms. The algorithms used are classic algorithms, and it would be interesting to try another type of algorithms to see if there is a performance improvement.
- Give greater importance to the most recent crime events, that is, events that occurred in the last month will have a more significant weight in training the model. This approach was not followed because the dataset of incidents only had data from the last two years. However, with the accumulation of data over time, it would be interesting to adopt this approach.
- Extend this solution to other cities for further comparison. It would be relevant to see if criminal trends change depending on the city. For example, checking whether crimes that occur in one city have any influence on crimes that occur in another, or if cities that have different characteristics follow the same criminal standards, for example comparing a coastal city with an inland city, or a city with high population density with a less populated city.

# References

[1] U. Nations, *68% of the world population projected to live in urban areas by 2050, says un | un desa | united nations department of economic and social affairs*, https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html, (Accessed on 26/10/2019), May 2018.

[2] K. Klein Goldewijk, A. Beusen, and P. Janssen, "Long-term dynamic modeling of global population and built-up area in a spatially explicit way: HYDE 3.1", *Holocene*, vol. 20, no. 4, pp. 565–573, 2010, ISSN: 09596836. DOI: 10.1177/0959683609356587.

[3] H. Ritchie and M. Roser, *Urbanization - our world in data*, https://ourworldindata.org/urbanization, (Accessed on 01/28/2020), Nov. 2019.

[4] Y. Wu, W. Zhang, J. Shen, Z. Mo, and Y. Peng, "Smart city with Chinese characteristics against the background of big data: Idea, action and risk", *Journal of Cleaner Production*, vol. 173, pp. 60–66, 2018, ISSN: 09596526. DOI: 10.1016/j.jclepro.2017.01.047. [Online]. Available: https://doi.org/10.1016/j.jclepro.2017.01.047.

[5] G. T. Database, *Incidents over time*, https://www.start.umd.edu/gtd/, (Accessed on 04/05/2020), Dec. 2018.

[6] B. Marr, *How much data do we create every day? the mind-blowing stats everyone should read*, https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#4486a8e660ba, (Accessed on 12/30/2019), May 2018.

[7] J. R. David Reinsel John Gantz, *The digitization of the world from edge to core*, https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf, (Accessed on 12/30/2019), Nov. 2018.

[8] M. Chen, S. Mao, and Y. Liu, "Big data: A survey", *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014, ISSN: 1383469X. DOI: 10.1007/s11036-013-0489-0.

[9] S. Sagiroglu and D. Sinanc, "Big Data : A Review", *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 42–47, 2013. DOI: 10.1109/CTS.2013.6567202.

[10] IBM, P. Zikopoulos, and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, 1st. McGraw-Hill Osborne Media, 2011, ISBN: 9780071790536.

[11] M. S. Hadi, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Big data analytics for wireless and wired network design: A survey", *Computer Networks*, vol. 132, pp. 180–199, 2018, ISSN: 13891286. DOI: 10.1016/j.comnet.2018.01.016. [Online]. Available: https://doi.org/10.1016/j.comnet.2018.01.016.

[12] H. Hu, Y. Wen, T. S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial", *IEEE Access*, vol. 2, pp. 652–687, 2014, ISSN: 21693536. DOI: 10.1109/ACCESS.2014.2332453.

[13] G. Vaseekaran, *Big data battle : Batch processing vs stream processing*, https://medium.com/@gowthamy/big-data-battle-batch-processing-vs-stream-processing-5d94600d8103, (Accessed on 12/30/2019), Oct. 2017.

[14] B. Cannadat, *Introducing losant notebooks*, https://www.losant.com/blog/platform-update-2019-04-25, (Accessed on 01/28/2020), Apr. 2019.

[15] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, ISSN: 1553877X. DOI: `10.1109/COMST.2015.2444095`.

[16] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey", *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, ISSN: 13891286. DOI: `10.1016/j.comnet.2010.05.010`.

[17] A. K, "That 'Internet of Things' Thing", *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[18] D. Evans, "IoT by Cisco 2011", *Cisco Internet Business Solutions Group (IBSG)*, no. April, 2011, ISSN: 09598138. DOI: `10.1109/IEEESTD.2007.373646`. arXiv: `arXiv:1011.1669v3`.

[19] A. Caragliu, C. D. Bo, and P. Nijkamp, "Smart Cities in Europe Smart Cities in Europe", *Proceedings of the 3rd Central European Conference in Regional Science*, vol. 0732, no. November, pp. 45–59, 2009. DOI: `10.1080/10630732.2011.601117`. [Online]. Available: `http://degree.ubvu.vu.nl/repec/vua/wpaper/pdf/20090048.pdf`.

[20] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey", *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018, ISSN: 23528648. DOI: `10.1016/j.dcan.2017.10.002`. arXiv: `1802.06305`.

[21] R. Giffinger, C. Fertner, H. Kramar, and E. Meijers, "City-ranking of European medium-sized cities", *Centre of Regional Science, Vienna UT*, no. October, 2007. [Online]. Available: `http://www.smart-cities.eu/download/city_ranking_final.pdf`.

[22] D. Pal, T. Triyason, and P. Padungweang, "Big data in smart-cities: Current research and challenges", *Indonesian Journal of Electrical Engineering and Informatics*, vol. 6, no. 4, pp. 351–360, 2018, ISSN: 20893272. DOI: `10.11591/ijeei.v6i4.543`.

[23] C. Carvalho, F. Pinto, I. Borges, G. Machado, and I. Oliveira, "Cognitive Cities: an Architectural Framework for the Cities of the Future", *Computer Science & Information Technology (CS & IT)*, vol. 9, no. 13, pp. 173–182, 2019. DOI: `10.5121/csit.2019.91314`.

[24] M. G. Institute, *Mgi_big_data_exec_summary.ashx*, `https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI_big_data_exec_summary.ashx`, (Accessed on 12/30/2019), May 2011.

[25] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data", *Information Sciences*, vol. 275, pp. 314–347, 2014, ISSN: 0020-0255. DOI: `https://doi.org/10.1016/j.ins.2014.01.015`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0020025514000346`.

[26] A. L. Samuel, "Some studies in machine learning using the game of checkers", *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, Jul. 1959, ISSN: 0018-8646. DOI: `10.1147/rd.33.0210`.

[27] C. Grosan and A. Abraham, *Machine Learning*. 1997, vol. 17, pp. 261–268, ISBN: 9783642210037. DOI: `10.1007/978-3-642-21004-4_10`.

[28] M. Mohammadi and A. Al-Fuqaha, "Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges", *IEEE Communications Magazine*, vol. 56, no. 2, pp. 94–101, 2018, ISSN: 01636804. DOI: `10.1109/MCOM.2018.1700298`. arXiv: `1810.04107`.

[29] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises", *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015, ISSN: 00076813. DOI: `10.1016/j.bushor.2015.03.008`. [Online]. Available: `http://dx.doi.org/10.1016/j.bushor.2015.03.008`.

[30] L. Heiler, *Difference of data science, machine learning and data mining - data science central*, `https://www.datasciencecentral.com/profiles/blogs/difference-of-data-science-machine-learning-and-data-mining`, (Accessed on 04/27/2020), Mar. 2017.

[31] C. McDonald, *Demystifying ai, machine learning, and deep learning - dzone ai*, `https://dzone.com/articles/demystifying-ai-machine-learning-and-deep-learning`, (Accessed on 01/18/2020), Aug. 2017.

[32] A. Bronshtein, *Train/test split and cross validation in python - towards data science*, https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6, (Accessed on 01/18/2020), May 2017.

[33] S. Narkhede, *Understanding confusion matrix - towards data science*, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, (Accessed on 01/19/2020), Sep. 2018.

[34] H. Awad, H. Ibrahim, S. M. Nor, A. Mohammed, and A. B. Mohammed, "Taxonomy of Machine Learning Algorithms to classify real- time Interactive applications", *IRACST – International Journal of Computer Networks and Wireless Communications*, vol. 2, no. 1, pp. 2250–3501, 2012. [Online]. Available: https://pdfs.semanticscholar.org/449a/34e3c27ac8398ac3a6f744c3d9e7f7deca08.pdf.

[35] L. McClendon and N. Meghanathan, "Using Machine Learning Algorithms to Analyze Crime Data", *Machine Learning and Applications: An International Journal*, vol. 2, no. 1, pp. 1–12, 2015. DOI: 10.5121/mlaij.2015.2101.

[36] O. Theobald, *Machine Learning For Absolute Beginners.* 2017, p. 128, ISBN: 9781549617218.

[37] M. J. Garbade, *Regression versus classification machine learning: What's the difference?*, https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7, (Accessed on 01/18/2020), Aug. 2018.

[38] A. Ibañez, *Semi-supervised learning... the great unknown - think big*, https://business.blogthinkbig.com/semi-supervised-learning-the-great-unknown/, (Accessed on 01/19/2020), May 2019.

[39] J. C. Kabugo, S.-L. Jämsä-Jounela, R. Schiemann, and C. Binder, "Industry 4.0 based process data analytics platform: A waste-to-energy plant case study", *International Journal of Electrical Power & Energy Systems*, vol. 115, p. 105 508, Feb. 2020, ISSN: 01420615. DOI: 10.1016/j.ijepes.2019.105508. [Online]. Available: google.pt.

[40] H. Yi, Q. Xiong, Q. Zou, R. Xu, K. Wang, and M. Gao, "A Novel Random Forest and its Application on Classification of Air Quality", *Proceedings - 2019 8th International Congress on Advanced Applied Informatics, IIAI-AAI 2019*, pp. 35–38, 2019. DOI: 10.1109/IIAI-AAI.2019.00018.

[41] S. Ray, "A Quick Review of Machine Learning Algorithms", *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, pp. 35–39, 2019. DOI: 10.1109/COMITCon.2019.8862451.

[42] L. Breiman, "Bagging predictions", *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996, ISSN: 08856125.

[43] A. Hershy, *Gini index vs information entropy - towards data science*, https://towardsdatascience.com/gini-index-vs-information-entropy-7a7e4fed3fcb, (Accessed on 04/25/2020), Jul. 2019.

[44] W. Koehrsen, *Random forest simple explanation - will koehrsen - medium*, https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d, (Accessed on 04/09/2020), Dec. 2017.

[45] A. Mohanty, *Multi layer perceptron (mlp) models on real world banking data*, https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f, (Accessed on 04/09/2020), May 2019.

[46] O. Harrison, *Machine learning basics with the k-nearest neighbors algorithm*, https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761, (Accessed on 04/10/2020), Oct. 2018.

[47] D. Subramanian, *A simple introduction to k-nearest neighbors algorithm*, https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e, (Accessed on 04/10/2020), Jun. 2019.

[48] Y. Wang, Y. Ou, X. Deng, L. Zhao, and C. Zhang, "The ship collision accidents based on logistic regression and big data", pp. 4438–4440, Jun. 2019, ISSN: 1948-9447. DOI: 10.1109/CCDC.2019.8832686.

[49] N. S. Chauhan, *Real world implementation of logistic regression - towards data science*, https://towardsdatascience.com/real-world-implementation-of-logistic-regression-5136cefb8125, (Accessed on 04/11/2020), Mar. 2019.

[50] Imarticus, *What is machine learning and does it matter?*, https://blog.imarticus.org/what-is-machine-learning-and-does-it-matter/, (Accessed on 01/26/2020), Feb. 2019.

[51] P. Pandey, *Data preprocessing : Concepts - towards data science*, https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825, (Accessed on 01/26/2020), Nov. 2019.

[52] D. ( Sarkar, *Categorical data - towards data science*, https://towardsdatascience.com/understanding-feature-engineering-part-2-categorical-data-f54324193e63, (Accessed on 01/26/2020), Jan. 2018.

[53] S. Asaithambi, *Why, how and when to scale your features - greyatom - medium*, https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e, (Accessed on 01/26/2020), Dec. 2017.

[54] L. A. Shalabi, R. Mahmod, A. Azim, A. Ghani, and Y. M. Saman, "A New Model for Extracting a Classifactory Knowledge from Large Datasets Using Rough Set Approach A New Model For Extracting A Classifactory Knowledge From Large Datasets Using Rough Set Approach", no. January 1999, 1999.

[55] Microsoft, *Modeling stage of the team data science process lifecycle | microsoft docs*, https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle-modeling, (Accessed on 01/26/2020), Jan. 2020.

[56] ——, *Deployment stage of the team data science process lifecycle | microsoft docs*, https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle-deployment, (Accessed on 01/26/2020), Jan. 2020.

[57] J. Isaak and M. J. Hanna, "User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection", *Computer*, vol. 51, no. 8, pp. 56–59, 2018, ISSN: 15580814. DOI: 10.1109/MC.2018.3191268.

[58] B. N. Mohapatra and P. P. Panda, "Machine learning applications to smart city", *ACCENTS Transactions on Image Processing and Computer Vision*, vol. 5, no. 14, pp. 1–6, 2019. DOI: 10.19101/tipcv.2018.412004.

[59] A. Gonfalonieri, *Big Data & Smart Cities: How can we prepare for them?*, Dec. 2018. [Online]. Available: https://medium.com/dataseries/big-data-and-smart-cities-why-we-need-them-now-a194b2498fb1 (visited on 10/30/2019).

[60] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities", *Journal of Internet Services and Applications*, vol. 6, no. 1, pp. 1–15, 2015, ISSN: 18690238. DOI: 10.1186/s13174-015-0041-5. [Online]. Available: http://dx.doi.org/10.1186/s13174-015-0041-5.

[61] P. Publishing, *Artificial Intelligence for Smart Cities - Becoming Human: Artificial Intelligence Magazine.* [Online]. Available: https://becominghuman.ai/artificial-intelligence-for-smart-cities-64e6774808f8 (visited on 10/30/2019).

[62] J. Pan, R. Jain, S. Paul, T. Vu, A. Saifullah, and M. Sha, "An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments", *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 527–537, 2015, ISSN: 23274662. DOI: 10.1109/JIOT.2015.2413397.

[63] G. Torbet, *Google's ai can detect breast cancer more accurately than experts | engadget*, https://www.engadget.com/2020/01/01/googles-ai-can-detect-breast-cancer-more-accurately-than-expert/, (Accessed on 01/02/2020), Jan. 2020.

[64] Y. L. Lin, T. Y. Chen, and L. C. Yu, "Using Machine Learning to Assist Crime Prevention", *Proceedings - 2017 6th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2017*, pp. 1029–1030, 2017. DOI: 10.1109/IIAI-AAI.2017.46.

[65] S. K. Rumi, K. Deng, and F. D. Salim, "Crime event prediction with dynamic features", *EPJ Data Science*, vol. 7, no. 1, 2018, ISSN: 21931127. DOI: 10.1140/epjds/s13688-018-0171-7. [Online]. Available: http://dx.doi.org/10.1140/epjds/s13688-018-0171-7.

[66] D. Kumar, *Introduction to data preprocessing in machine learning*, https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d, (Accessed on 02/24/2020), Dec. 2018.

[67] A. Kızrak, *Comparison of activation functions for deep neural networks*, `https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a`, (Accessed on 07/08/2020), May 2019.