eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Master's Dissertation

Master's Degree In Computational Engineering and Intelligent Systems

# Development of artificial intelligence models for the enrichment and exploitation of geospatial data in the built environment

*Baterdene Batmunkh*

**Advisors**
José David Núñez González (UPV/EHU)
José Antonio Chica Páez (Tecnalia)
June 2022

# Abstract

Geospatial data treatment is an important task since it is a big part of big data. Nowadays, geospatial data exploitation is lacking in terms of artificial intelligence. In this work, we focus on the usage of a machine learning models to exploit geospatial data. We will follow a complete workflow from the collection and first descriptive analysis of the data to the development and evaluation of the different machine learning algorithms. From download dataset we will predict if the download will lead to civil work, in other words, it is a classification problem. We conclude that combining machine learning and geospatial data we can get a lot out of it.

# Contents

# List of Figures

# List of Tables

# CHAPTER 1

# Introduction

In recent years, the amount of geospatial data has grown and will grow exponentially according to the U.S. National Geospatial Intelligence Agency. That is why the traditional treatment of this information has become completely obsolete, both in terms of computing capacity and exploitation of knowledge. Thus, we must move towards an information analysis methodology that delegates tasks to computational intelligence.

The main objective of this project is the development of an artificial intelligence model to enrich and exploit this geospatial data. Therefore, a complete workflow will be followed from the collection and first descriptive analysis of the data to the development and evaluation of the machine learning models.

To be precise, we have two datasets of all the downloads carried out in two cities. The data from the first city is collected since 2002, while the second city is collected since 2010. First we are going to clean this two datasets and apply some methods such as outlier detection, feature selection, etc. to get a certain dataset adapted to our project, machine learning algorithms.

Furthermore, we will have another dataset of construction works done in those two cities. However, due to some problems there is still no data to validate our proposal, so we used different ways to replace that dataset. On the one hand, we built some models to generate a synthetic dataset with some degree of arbitrary complexity, on the other hand we followed some reasoning, so that if our approach succeeds on this dataset, it can be useful when we acquire the dataset.

Then, we will associate the construction work dataset to the download dataset and select possible downloads that led to construction work. Therefore, it is a classification problem which identifies the downloads that led to construction work.

Finally, to evaluate the different classification models we used some evaluation metrics, such as precision, accuracy, recall, etc. Subsequently compared the models statistically.

## 1.1 State of the art

Over recent years, the exploitation of geospatial data has been of great importance, since a significant part of big data is actually geospatial data, and the size of such data is rapidly growing by at least 20% every year as it says in [1]. This exploitation benefits in fuel and time savings, increased income, urban planning, medical care, etc. On the other hand, geospatial data is important for Earth observation, geographic information system/building information modeling (GIS/BIM) integration and 3D/4D urban planning [2]. The general concept to analyze GIS and BIM data structures and spatial relationship will be of great importance in emerging applications such as smart cities and digital twins [3].

In last two decades there have been several projects related to geospatial data and artificial intelligence.

In 2008 [4] as geographic information systems (GIS) are widely used in urban police agencies to crime pattern analysis and as many of the underlying processes that give rise to crime patterns are not visible, they combined criminology, computer simulation and geographic information systems to examine crime patterns form and what can be done to prevent crime. To address this problem, a virtual cityscapes to model artificial crime patterns within a computing environment was created. In 2015 [5] support vector machine (SVM) and coactive neuro-fuzzy inference system (CANFIS) algorithms were tested to predict crash severity in a regional highway corridor and discover spatial and non-spatial factors that are systematically related to crash severity. Also, a sensitivity analysis is carried out to determine the relative influence of the crash. In 2017 [6] a GIS based flood modeling for Damansara river basin in Malaysia. The frequency ratio method was combined with SVM to estimate the probability of flooding. The flood hazard map was produced by combining the flood probability map with flood triggers such as daily rainfall and flood depth. The approach of this project would be effective for flood risk management in the study. Furthermore, [7] presents an advanced methodology developed by using Information and Communication Technologies (ICT) and artificial intelligence to support decision making in public and judicial administration. A prototype Management Information System for public administration (MISPA) was developed to provide a computerized way of managing geospatial urban, environmental and crime data of an urban area. The proposed system was developed aiming at the systematization and modernization of public, judicial and police authorities that have to be dealt by studying urban data regarding crime and environmental data and supports decision making based on crime forecasting. In 2018 [8] used machine learning to address the challenge of layers in geospatial data. The fundamental hurdle in geospatial data is identifying what number of feature levels is necessary to represent user's multidimensional preferences by considering semantics, such as spatial similarity and metadata attributes of static data sets. In addition, [9] as geospatial artificial intelligence (GeoAI) is an emerging scientific discipline and GeoAI provides important advantages for exposure modeling in environmental epidemiology, an overview of key concepts surrounding the evolving and interdisciplinary field of GeoAI including spatial data science, machine learning, deep learning, and data mining; recent GeoAI applications in research; and potential future direction for GeoAI in environmental epidemiology is provided. In 2019 [10] a modeling combining the LogitBoost classifier and decision tree and geospatial data from multiple sources were used for the spatial prediction of susceptibility to tropical forest fires. This project is necessary for disaster management and a primary reference source in territorial planning. SVM, random forest

(FR) and kernel logistic regression (KLR) were used as benchmarks. In 2020 [11] a project to facilitate planning efforts using multitude of tightly interlocked component measured by new sensors, data collection and spatio-temporal analysis methods. With geospatial data and urban analysis understand urban dynamics and human behavior for planning to improve livability. Moreover, [12] in India random forest model to produce exposure maps of the areas and populations potentially exposed to high arsenic concentrations in groundwater. On top of that, [13] using AI-based techniques for 3D point clouds and geospatial digital twins as generic component of geospatial AI. 3D point clouds can be seen as a corpus with similar properties as natural language corpora and formulate a 'naturalness hypothesis' for 3D points clouds. Finally [14] summarizes the historical origins of GeoAI development, introduces spatially explicit and implicit AI models, reviews recent GeoAI research and applications (including spatial representation learning, spatiotemporal prediction and spatial interpolation, monitoring of geographic resources and environment, cartography, and geo-text data semantic analysis), and identifies several potential research challenges and directions for the future development of GeoAI. In 2021 [15] as advancement of connected sensors, cloud technologies, big data analytics, machine learning algorithms, and ubiquitous sensing systems have enabled cognitive Internet of Things, it carries out a quantitative literature review of ProQuest, Scopus, and the Web of Science throughout March and April 2021, with search terms including "cognitive Internet of Things," "cognitive computing technologies," and "cognitive sensor networks". By analyzing and eliminating controversial or unclear findings in researches published between 2015 and 2021, only 142 papers met the eligibility criteria.

## 1.2 Objectives

Firstly, considering that we are going to have the download dataset and the construction work dataset, our main objective of this project is to classify the download dataset if they will lead into a construction work using these two datasets. To achieve this objective, the process has been broken down into more specific tasks.

Firstly, database comprehension, understanding the meaning of the each feature of the database and spotting the values of the features is key part of the project, since they communicate all information related to the whole project. Here, we analyzed every and each feature of the database and found the meaning of it. Problems related to this task can be several, such as the wrong meaning of the feature, not perceiving two or more features meaning the same, etc.

Then, dataset cleaning, clean data will increase overall productivity and allow for the highest quality information in the decision-making. To carry out this task, we did different changes in the dataset, such as feature selection, correcting the wrong spellings, putting the same structure and removing the atypical observations. Implementing this task can bring several problems too. For example, applying wrong techniques that will affect the models, removing wrong features or selecting unnecessary features, removing or correcting wrongly the observations, etc.

Afterwards, data analysis, once deciding that in our dataset remains the highest quality information's, we drew conclusions by analyzing the dataset. This conclusions could help to proceed with the project. For instance, as we had different types of data such as quantitative and qualitative, we analyzed the features two by two. When we carried out this task we avoided some problems such as not analyzing two uncorrelated features, not accounting for the time of the year that can lead to misleading, etc.

Finally, testing with different machine learning algorithms. To fulfill this task, first we feed the dataset to different machine learning algorithms, then inspected the performance of the each algorithms by different evaluation metrics. Moreover compared the models by practical and statistical significance. At last, for more experiment we tested the performance by unbalancing the predictor class.

## 1.3 Research proposal

The proposal of this project is to spread the use of geospatial data and intelligence artificial and get the most out of this data. We will clean the raw data and use various techniques to have certain data for the machine learning algorithms. The main focus of this project is to classify the download dataset if they will lead to construction work. We will use this prediction to make decision on some topics. For example, notify the town hall that there can be construction work, thus identify the similar construction works that have to be executed in the same area, so that the several construction works can be planned together in time to reduce times of inconvenience to neighbors. Furthermore, crossing the data of the construction works that are going to be executed, with the data of the use of credit cards, see the impact that the construction works have on local commerce. To that end, compare the previous period of work and subsequent scenarios. Also, analyze the general economic impact of the construction works on commerce, in this way help them activate local financial aid programs. Finally, create a business plan to sell materials to the downloading company.

CHAPTER 2

# Metodology

In this section we will describe the different databases, comment all characteristics of the machines used and the steps taken to develop this project.

2.1 is the pipeline which summarizes the process of the experiment.

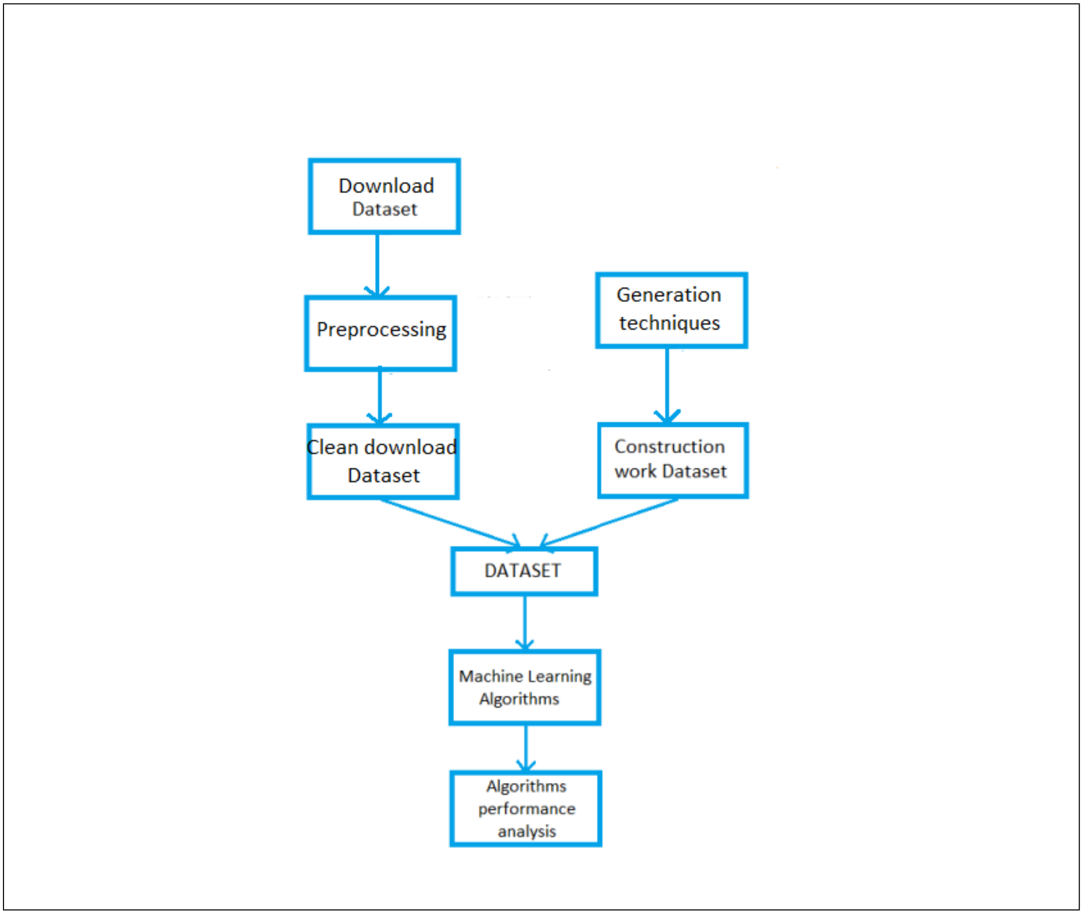

Download Dataset
Preprocessing
Generation techniques
Clean download Dataset
Construction work Dataset
DATASET
Machine Learning Algorithms
Algorithms performance analysis

**Figure 2.1:** Data map after the 'WORK' variable generation. 0 are the downloads that did not lead to construction work and 1 are the downloads that led to construction work.

## 2.1 Materials

The materials used in this work were a laptop of a processor AMD Ryzen 5 4600H with Radeon Graphics 3 GHz, 8 GB RAM and 64-bit operating system, windows 2010. he coding was done using R version 4.1.3 .

## 2.2 Database description

The first database of downloads was collected since 2002. This database has a total of 7497 downloads and each download has 13 different information associated, in other words, 13 features.

| | |
|---|---|
| 1: REQUEST_ID | download identification, unique integer values. |
| 2: NAME | Name of the enterprise that downloads the materials. |
| 3: DATE | Date when the download is done: YY-MM-DD HH:MM:SS.MMM |
| 4: CENTROID | Geographic centroid of the download: 123456,1234567. |
| 5: MUNICIPALITY | Municipality where the download is done. |
| 6: TYPE | download type: 'cultive' (3); 'work' (1932); 'minor work' (18); 'project' (906). |
| 7: PROMOTER | Name of the company that has requested the download. |
| 8: CIF | CIF of the enterprise that downloads. |
| 9: CLIENT_ID | Identification in the database of the enterprise that downloads the materials. |
| 10: ACTIVITY | The activity carried out by the enterprise that downloads the materials. |
| 11: AREA | The area it covers from the centroid in hectares. |
| 12: COST | download cost in euros: two decimal places for cents. |
| 13: SERVICE_CLASS | The purpose of the download: 'canalization' (1946); 'edification' (146); 'civil work' (423); 'urbanistic planning' (5); 'urbanization' (160); 'others' (175). |

**Table 2.1:** Features: on the left feature names and on the right feature description.

The second database of downloads was collected since 2010. This database has a total of 2859 downloads, however has 14 features, one more than the other one.

| | |
|---|---|
| USER | User that records the information of the downloads on a database. |

In addition to the two databases, there is another database of construction work licenses. This dataset is used to associate with the download dataset and identify the downloads that led to a construction work. However, we have a problem as there is still no data. Thus, we have had to generate a synthetic dataset with some degree of arbitrary complexity, so if our approach succeeds on this dataset, it can be successful in real life experiments.

In order to generate this dataset, we surely know that the construction work dataset will have features as the date and the centroid. So, our synthetic dataset will revolve around those features. As we said previously, we used two different ways to replace this dataset. In the first way, we generated 1200 (0.44% of total downloads) values for the 'YEAR' feature following the yearly download distribution as we can see in Figure 2.2. Here we considered that the quantity of the downloads and the construction works are proportional, that is to say, if there are more downloads, there will be more construction works.

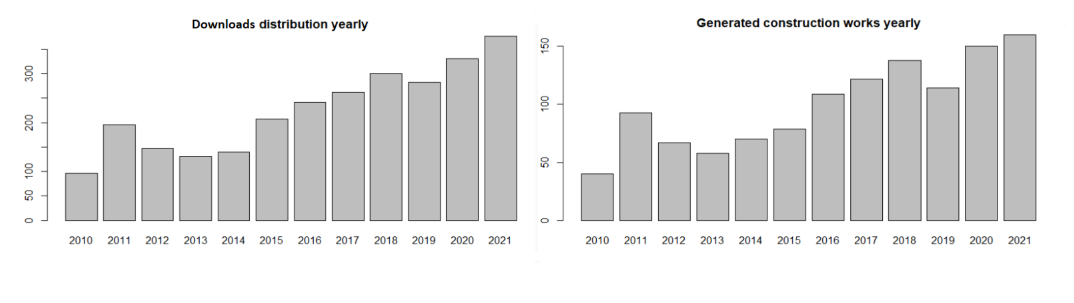


**Figure 2.2:** On the left downloads distribution yearly and on the right generated 'YEAR' feature distribution.

Then, we generated 'MONTH' feature following arbitrary maps for each year. We used arbitrary maps since monthly construction works distribution is quite aleatory. These arbitrary maps were created by two utility functions (Figure 2.3). We used two utility functions to compare that in machine learning algorithms will not have much difference. We followed next steps to generate the months for each year.

1. Using the utility function, create a arbitrary map.
2. Following that arbitrary map generate the months taking into account that we set maximum construction works when we generated the years.
3. Review the generated months for construction works with download quantity for that month, since there can not be more construction works than the downloads.

Lastly, we generated the points for the centroid. For this task we used different distributions (Figure 2.4):

- **Normal distribution** [16]: Normal distribution also known as Gaussian Gauss is a type of continuous probability distribution for a real valued random feature. We used this distribution considering that there will be more construction works in the city center than in the outskirts of the city.
- **Bimodal distribution** [17]: A bimodal distribution is a probability distribution with two modes. This distribution has two most commonly occurring value in a dataset as we can see in Figure 2.4. In this distribution we considered that the construction works will be more scattered.

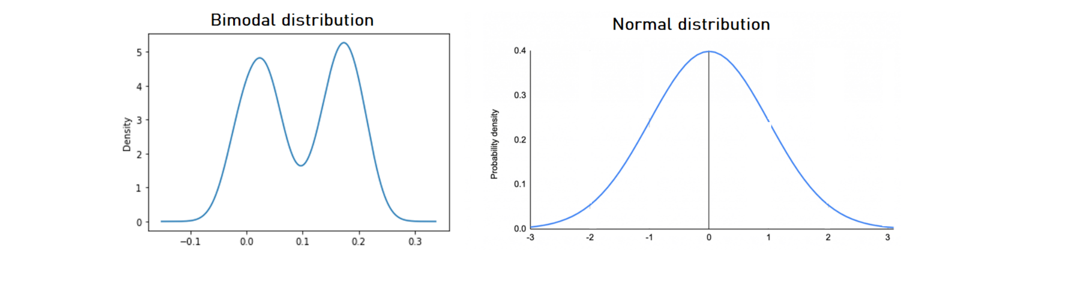


**Figure 2.4:** On the left, bimodal distribution, where we can see two main peaks. On the right, normal distribution where there is only one peak

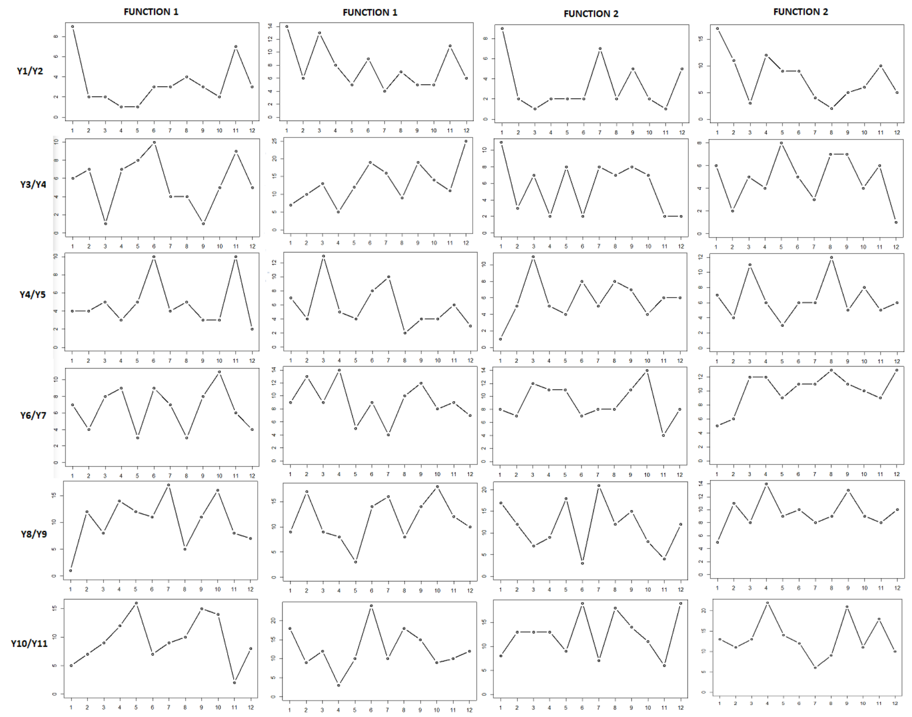


**Figure 2.3:** First two columns are arbitrary map generated using first function and the other two columns are arbitrary maps generated using the second function. Each arbitrary map is for one year.

To carry out this task we followed the next steps.

1. For each month take all construction work downloads done in that month.

2. Following the two different distributions generate X points of the centroid taking as parameter downloads point X‘s max, min, mean, deviation standard, etc.

3. For each generated X points select downloads that their X point of centroid are close to.

4. Again using the distributions generate Y points of the centroid taking as parameter downloads that are close to the generated X.

Our construction works synthetic dataset have 11 features: feature ’YEAR’; two ’MONTH’ feature for each utility function; for each ’MONTH’ feature, point X and point Y using normal distribution and another point X and point Y using bimodal distribution.

## 2.3 Development

In this section we will specify the steps taken to develop this project. After, once generated the synthetic datasets, we have 3 different datasets: the downloads dataset, function following construction works dataset and reasoning following construction works dataset.

Then, we have to associate the downloads dataset with the other two. However, as the downloads dataset is incomplete, it needs a cleansing.

After analyzing the downloads dataset, we applied several preprocessing to clean it. First of all, feature selection, some features of the download dataset are meaningless. This way, we reduce the number of input variables to minimize the computational cost of modeling.

| | |
|---|---|
| REQUEST_ID | We can identify the downloads by integer sequence beginning by 1. |
| NAME | We can identify the enterprises by their CIF, how they name it is meaningless for machine learning algorithms. |
| USER | Users that records the information do not affect the result. |
| MUNICIPALITY | Municipality where the download is done do not affect the result. Moreover, it has static value, therefore the standar deviation is 0. |
| CLIENT_ID | We will use CIF to identify the enterprises that downloads the materials. |

**Table 2.2:** Meaningless features: on the left feature names and on the right the reason why it is meaningless.

On the other hand, some features are better represented by other ways. Thus, some tasks are effectively carried out, such as the analysis of the dataset.

| | |
|---|---|
| DATE | We represented the date yearly and monthly. In future we may represent daily too. Thus the dataset has two more features: YEAR; MONTH. |
| CENTROID | Geographic points are better represented by X and Y. |

**Table 2.3:** Features that can be represented better.

After previous changes, the new dataset has 11 features: 'TYPE'; 'PROMOTER'; 'CIF'; 'ACTIVITY'; 'AREA'; 'COST'; 'SERVICE_CLASS'; 'X'; 'Y'; 'YEAR'; 'MONTH'.

Over this new dataset, we changed the values of the feature 'PROMOTER', since there were values that meant the same but it was written differently. For example, 'city hall' and 'city hal'. After changing the values for 'PROMOTER' feature, it had x unique values, which could affect the performance of the models due to high cardinality. As a consequence we selected the unique values that has appeared more than x times and others sampled as 'other', by this way we had x unique values for 'PROMOTER' feature.

Furthermore, detection of atypical or abnormal observations is important, since it can potentially affect the estimation of parameters. There are different ways to deal with atypical observations, but we decided to remove them. For this task, after analyzing the dataset we perceived that some points were out of the range of the city as we can see in the Figure 2.5, thus this points would be classified as atypical observations, then to be removed.

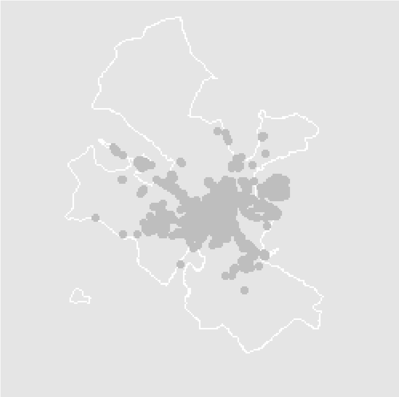


**Figure 2.5:** Map of the city and at the right the atypical observations, such as the points out of the limit.

Once the cleaning was done, we analyzed the download dataset associating the features between them and drew some conclusions.

To begin with, we did temporal analysis using the features 'YEAR' and 'MONTH' to see the temporal download behavior. Analyzing it yearly we realized that one city had uptrend since 2010 (100 downloads) until 2021 (400 downloads). Hence, it can be expected that in 2022 there will also be quite a few downloads. Likewise, analyzing it monthly we can notice the deterioration in August and December, since these months are holidays, and between these two months there is a rise and a fall (Figure 2.6).

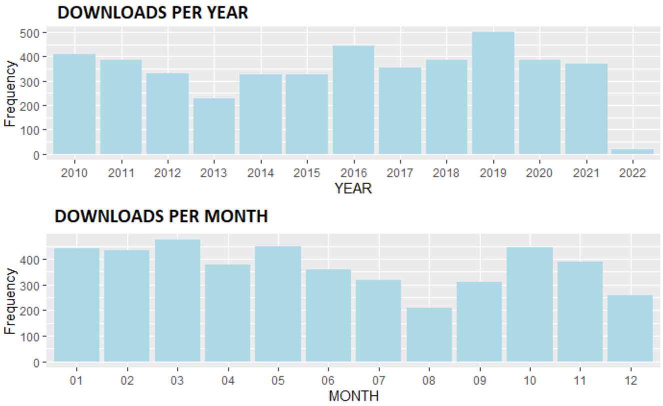


**Figure 2.6:** Data analysis: downloads per year and downloads per month.

Then, we did temporal analysis of the feature 'TYPE' and realized that the uptrend is due to the download type *project* (Figure 2.7), since it has increased vastly last years. Doing it monthly follows the trend that we have said before, deterioration in August and December, and between them a rise and a fall.

Also, doing the temporal analysis of the promoter, the city hall promoted lot more in years of covid compared to the other years, since in 2020 and 2021 they promoted 120 downloads each year, while the maximum of the other years do not surpass 60.

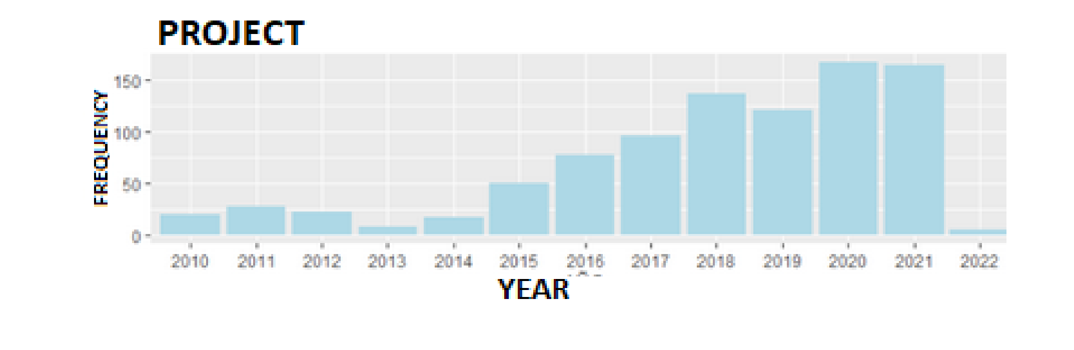


**Figure 2.7:** Data analysis: 'project' type frequency yearly.

Finally, downloads that leads to urbanization and civil work has uptrend temporarily, while canalization and the others had different distributions.

Furthermore, we did associate different features and drew some conclusions as the city hall focus more on canalization and civil work, the city hall contacts more with enterprises that are contractor, the area it covers depending on download type, etc (Figure 2.8 and Figure 2.9).

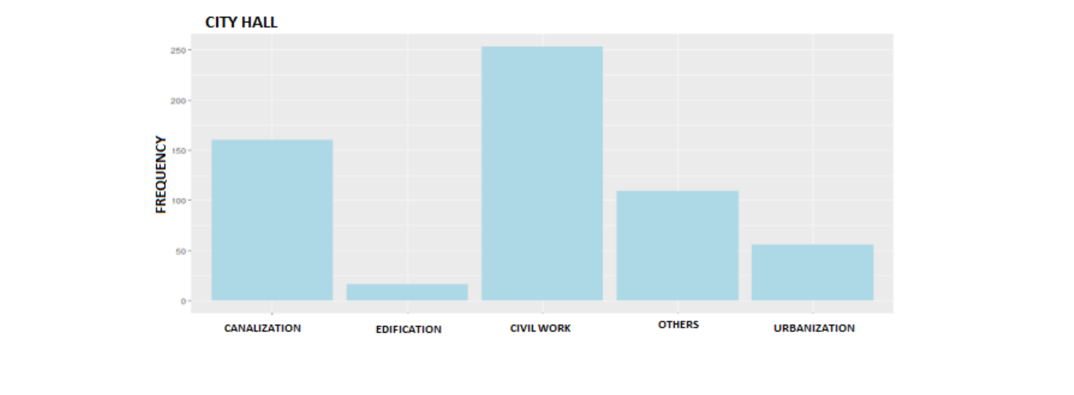


**Figure 2.8:** Data analysis: City hall distribution on the purpose of the download .

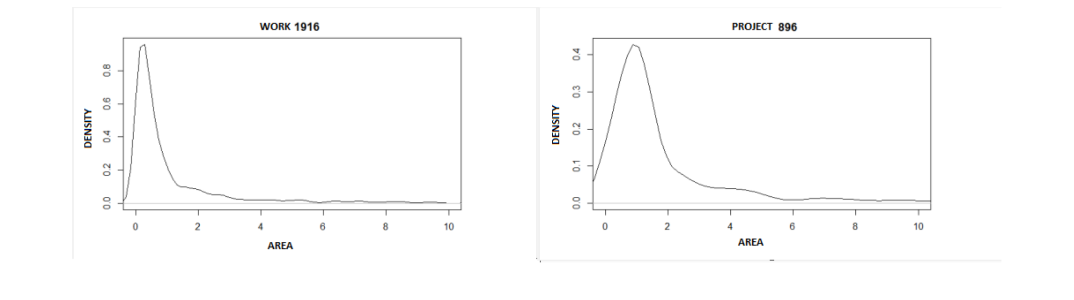


**Figure 2.9:** Data analysis: Area density depending on the two most appearing type of the download.

After analyzing the dataset we decided to cluster by geographical points. This way we could analyze the changes depending on clusters and see if there is different performances.

We used two different methods for the clustering, therefore, we have two more features, one for each cluster (Figure 2.10):

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN)** [18]: Unsupervised learning algorithm for clustering. This algorithm uses density to cluster the data points. Initially this algorithm classifies the points into three categories: Core points, Border points and Noise points. Core points must have equal or greater than minimum neighbors. Border points has less than minimum neighbors and the point should be in the neighborhood of a core point. Lastly, noise points are points that are neither a core point nor a boundary point. Once classified the points, if two core points are neighbors they are linked by a density edge and are called density connected points. Finally, it discards noise, assign cluster to a core point, color all the density connected points of a core point and color boundary points according to the nearest core point.

- **Model-Based Clustering** [19]: Statistical approach to data clustering. The fit between the given data and some mathematical model, and is based on the assumption that data are created by combination of a basic probability distribution. Initially assigns k cluster centers randomly and iteratively refines the cluster based on two steps: Expectation step and maximization step.

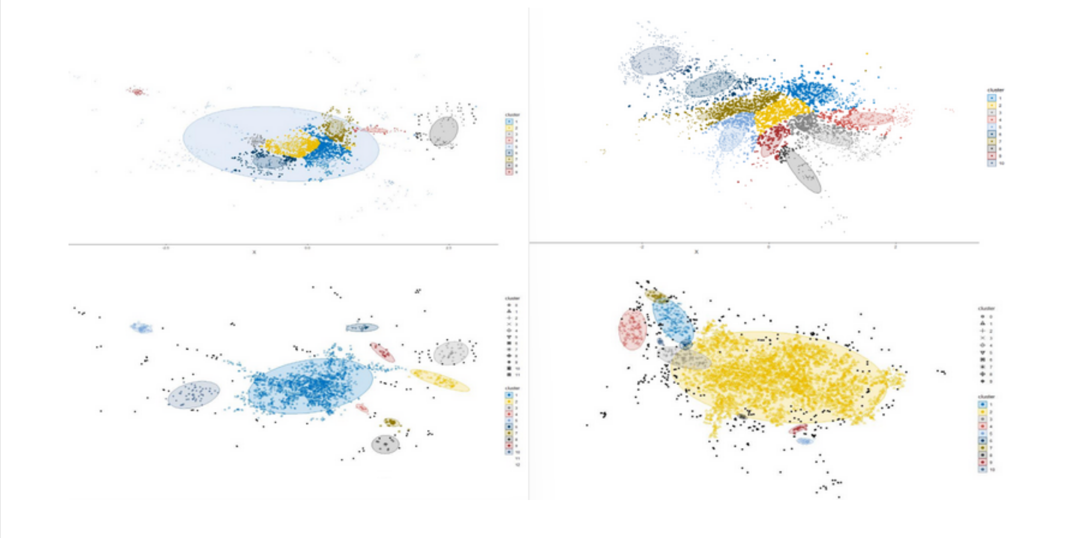


**Figure 2.10:** On the left two different clusters of one city and on the right the other two different clusters of the other city. The top is the cluster using Model-Based Clustering, while the bottom is the cluster using DBSCAN.

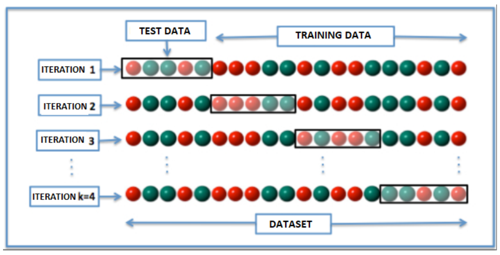
Then we analyzed the cost depending on clusters and saw that the cost do not change depending on where it downloads.

Once the download dataset is cleaned, we have to associate with the construction works dataset to add another feature to the download dataset that represents the download has led to a construction work. As we generated the construction works dataset previously, we have to associate it with the download dataset.

First, we added a new binary feature 'WORK' to the download dataset which represents that the download has led to construction work or not. The succeeding phase, for each instance in construction works dataset we took the closest download done in that year and month and assigned that download a value of 1 in the feature that represents that the download lead to a construction work. After this phase, our download dataset has 4 more features, since our construction works dataset had 4 different geographical point: two points from the first utility function using normal and bimodal distribution; another two points from the second utility function using normal and bimodal distribution.

### 2.3.1 Supervised classification

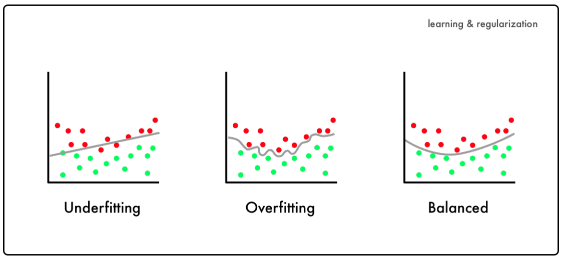
After cleaning the dataset and prepared it for the machine learning models, we will fed the download dataset to classification models using generated binary feature as class variable. We used 100 different seeds for every model, which was trained using 10 fold cross-validation [20]. This method, 10 fold cross-validation, partition the training dataset depending on a variable 'k' which in this case is 10. It partition the training dataset into 10 subset, then uses each subset as test data and the rest as training data (Figure 2.11). Consequently, avoids overfitting [21], overfitting occurs when a statistical model fits exactly against its training data. So the model cannot perform accurately against unseen data (Figure 2.12). Overfitting is a common problem of generalization, which causes a poor performance of machine learning algorithms. This problem can be identified by checking the evaluation metrics of the model on the training dataset and the test dataset, for example, when the accuracy on training dataset is much higher than the test dataset.



1

**Figure 2.11:** 4-fold cross validation. Dataset partition into 4 subsets and every iteration uses each subset as test data and the rest as training data[1].

[1] Source: https://en.wikipedia.org/wiki/File:K-fold_cross_validation_EN.svg

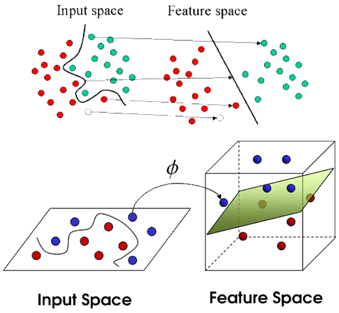


2

**Figure 2.12:** Graphical visualization of overfitting. On the first column, we can see underfitting since the line that separates two classes is too lazy; On the second column, overfitting, where the line is more specified; Lastly, balanced where the line is more smooth comparing to first two columns[2].

Then we will fed to different classification models to see the performance.

- **Support Vector Machine (SVM)** [22]: Supervised learning model that is able to generalize between two different classes. SVM checks for a hyperplane that is able to distinguish best between two classes among many hyperplanes. When the data is non linearly separable, SVM makes use of kernel tricks [23] to make it linearly separable. This kernel tricks help in projecting data points to the higher dimensional space which they can be linearly separated (Figure 2.13).



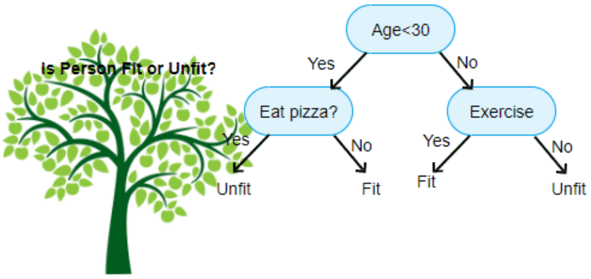
3

**Figure 2.13:** SVM classifies the classes by a line. The input space is changed to be able to separate the classes by a line[3].

- **Decision Tree** [24]: Algorithm that uses a set of rules to make decisions, type of flowchart. This model uses the dataset features to create yes/no questions and continually split the dataset until it isolates all data points belonging to each class. Every question is a node of the tree and the first node is called the root node. (Not well-suited to continuous variables) (Figure 2.14).

[2] Source: https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d

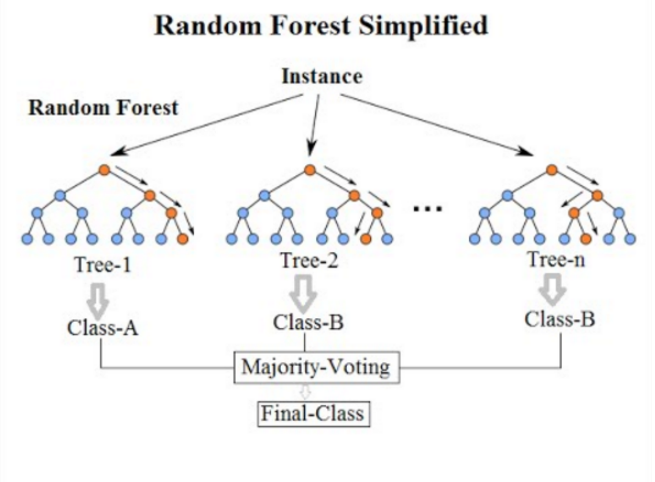
[3] Source: https://analisisdedatos.net/mineria/tecnicas/SVM/ejSVM.png



4

**Figure 2.14:** Decision tree to classify if the person is fit or not fit. On the first row is dealing with a continuous feature 'Age', which uses greater than; While on the second row is dealing with binary features of answer yes or no[4].

- **Random Forest** [25]: Model that combines the output of multiple decision trees to reach a single result. Random forest feature randomness, also known as the random subspace method [26], generates a random subset of features, which ensures low correlation among decision trees. Random forest algorithms predict more accurate results, particularly when the individual trees are uncorrelated with each other (Figure 2.15).



5

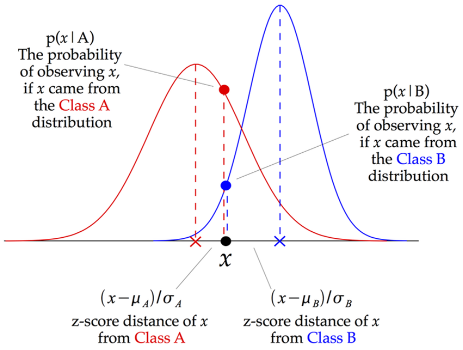
**Figure 2.15:** Random forest made up by several decision trees, which combines the output of the decision trees to reach a single result[5].

- **Gaussian Naive Bayes** [27]: Naive Bayes is a probabilistic algorithm based on Bayes theorem. Gaussian Naive Bayes is the extension of naive Bayes that follows Gaussian

[4] Source: https://rubenjromo.com/wp-content/uploads/2019/10/decisiontree.png.webp

[5] Source: https://upload.wikimedia.org/wikipedia/commons/thumb/7/76/Random_forest_diagram_complete.png/330px-Random_forest_diagram_complete.png

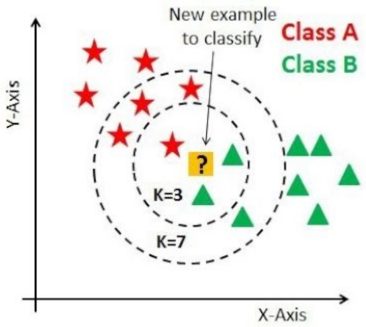
normal distribution and supports continuous data (Figure 2.16).


p($x$ | A)
The probability of observing $x$, if $x$ came from the Class A distribution
p($x$ | B)
The probability of observing $x$, if $x$ came from the Class B distribution
$x$
$(x-\mu_A)/\sigma_A$
z-score distance of $x$ from Class A
$(x-\mu_B)/\sigma_B$
z-score distance of $x$ from Class B

[6]

**Figure 2.16:** For each data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class[6].

- **K-Nearest Neighbor (KNN)** [28]: Algorithm that uses feature similarity to predict. It takes 'k' most similar object (neighbors) using distances and depending those objects it decides the outcome. It is recommendable to use odd k numbers, since there is a risk of a tie in even numbers. This algorithm is sensitive to the local structure of the data, since it is only approximated locally (Figure 2.17).



[7]

**Figure 2.17:** The point of the center is classified differently depending on the k. If the k is 3, then the point will be classified as Class B since there are more class B points in the area. However, if the k is 7, then the point will be classified as Class A since there are more[7].

After evaluating the models, we perceived that the results were not that good as we can see in the chapter 3 section results. This problem arises since our data has too much

[6] Source: https://www.researchgate.net/figure/Illustration-of-how-a-Gaussian-Naive%2DBayes-GNB-classifier-works-For-each-data-point_fig8_255695722

[7] Source: https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn%2D35d936e554eb

randomness and the models has the problem to classify them properly. Thus, we have decided to take another path when it comes to generating the predictor class.

We decided to follow some reasoning to generate the predictor class and generate a data map as we can see in Figure 2.18.

- TYPE: Set 1 on 90% of the minority classes, that is, those that appear less than 40 times.
- PROMOTER: From the 'city hall' (553) and 'company name' (544) to 60% we have set 1, since these values have risen in recent years. Therefore, we have reasoned that their licenses were being accepted. Moreover, we have given more importance to 'city hall', since we thought that 'city hall' projects has more possibility to be accepted.
- ACTIVITY: From the 'city hall' and 'engineering' to 60% we have set 1. Here we have followed same reasoning: uptrend and city hall.
- SERVICE_CLASS: From 'urbanization' and 'civil work' to 70% we have set 1.
- From other features we could not get reasonable things, so from the rest of the 0 values to the 15% we have set 1 randomly.

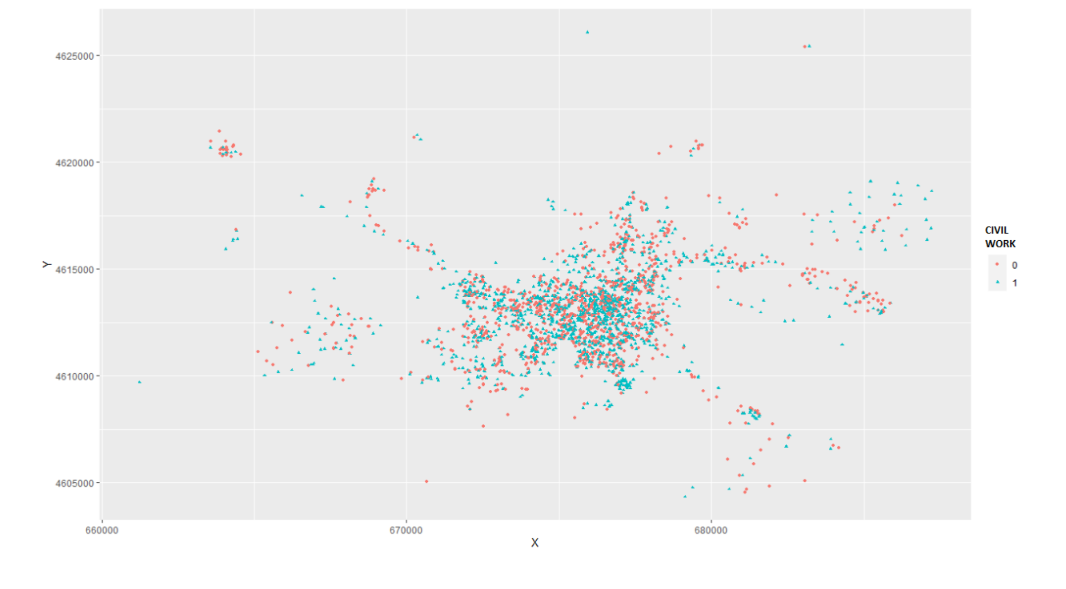


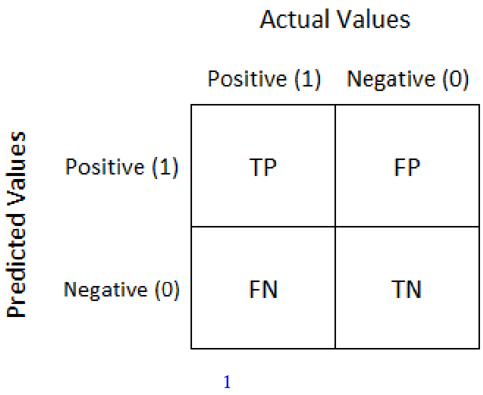
**Figure 2.18:** Data map after the 'WORK' variable generation. 0 are the downloads that did not lead to construction work and 1 are the downloads that led to construction work.

The download dataset needed a general review, since we just added one variable value, that is, for one construction work there can be more than one download, so if one of those downloads has 1 other downloads have to have 1. This review was carried out by checking if some features are equals and the centroid point is in a certain radius of the download centroid that led to construction work. After this review we got 1245 instances for class 0 and 1487 instances for class 1.

CHAPTER 3

# Aplication

Once the models are trained and tested it has to be evaluated to see the performance. For this task we used different evaluation metrics. This evaluation metrics are calculated using a confusion matrix. Confusion matrix is an N x N matrix, where N is the number of target classes. This matrix compares the actual target values with those predicted by the machine learning model. In our case, since we have 2 different classes, the matrix is 2x2 and it has 4 different values: TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative).



| | | Actual Values | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

[1]

**Figure 3.1:** 2x2 confusion matrix. TP (True Positive); FP (False Positive); FN (False Negative); TN (True Negative)[1].

- **Accuracy**: Measures how often the classifier makes the correct prediction. It is the ratio between the number of correct predictions and the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

[1] Source: https://bookdown.org/f_izco/BDC-POC/metricas.html

- **Precision**: Measures the correctness that is achieved in true prediction, in other words, the actual positive predictions out of all the total positive prediction.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**: Measures what percentage of the positives have been classified as positive.

$$Recall = \frac{TP}{TP + FN}$$

- **Specificity**: Measures the proportion of true negatives that are correctly identified by the model.

$$Specificity = \frac{TN}{TN + FP}$$

- **f-score**: The harmonic mean of precision and recall. The harmonic mean is not sensitive to extremely large values, unlike simple averages. It maintains balance between the precision and recall.

$$fscore = 2 * \frac{Recall * Precision}{Recall + Precision}$$

- **Matthews Correlation Coefficient (MCC)**: Measures the correlation of the true classes with the predicted labels. It ranges in the interval [-1, +1], with -1 meaning perfect misclassification and +1 perfect classification, while 0 means random guessing. MCC produces high score only if the prediction obtained good results in all of the confusion matrix categories (TP, FP, TN, FN).

$$MCC = \frac{TN * TP - FN + FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- **Multiclass Confusion Entropy (MCEN)**: Measures generated entropy from misclassified cases considering not only how the cases of each fixed class have been misclassified into other classes, but also how the cases of the other classes have been misclassified as belonging to this class, as it says in article [29].

After testing the models and measuring the performance practically, we have compared the models statistically. First we applied Kolmogorov Smirnov [30] to see if there is normality in our evaluation metrics. We compared the p-value if it is less than 0.05 or not. When p-value is less than 0.05, it rejects the null hypothesis, which we have sufficient evidence to say that the data does not follow a normal distribution. Otherwise, it would follow a normal distribution.

Depending if the data comes from a normal distribution or not, we will use ANOVA or Kruskal Wallis [31] to compare between models accuracy's . Applying ANOVA or Kruskal Wallis we got to focus on the p-value, if it is greater than 0.05, we can conclude that there are not significant differences between machine learning models.

## 3.1 Results

In this section we will show the results achieved by the machine learning algorithms. First we will show the results achieved using the utility function and the distributions as we can see in the tables 3.1, 3.2, 3.3 and 3.4. The first two tables are from normal distribution in construction works, namely, considering that that there will be more construction works in the city center than in the outskirts of the city. While, the consequent two are from bimodal distribution, which considers that the construction works will be more scattered. These results were accomplished using a distribution of 1204 for class 0 and 1536 for class 1, that is to say, the 44% of the downloads do not lead to construction work meanwhile the 56% do lead to construction work. As we can see the first results were done on a balanced predictor class.

However, the results achieved using the utility functions are lacking, since they do not outdo 60%. In this experiment, we can see that all models performs similarly. These results are the cause of a large randomness. In the Figure 2.2 we can see the randomness of the generated database, and the models have difficulty to classify, since when there is nothing to learn and the models produce predictions uncorrelated to the label. Moreover, we can see the slight difference between normal and bimodal distribution. When the construction work is more scattered the models performs slightly better. Then when we compared the models statistically. First we have proved that the data do not come from normal distribution since we got the p-value less than 0.05 after applying kolgomorov, thus we reject the null hypothesis.

As our data do not come from a normal distribution we applied Kruskal Wallis comparing the models. We have not identified significant differences between the models, since the p-value is less than the significance level 0.05.

| **Mean** / **Standard deviation** | **SVM** | **Random Forest** | **Decision Tree** | **Gaussian Naive Bayes** | **k Nearest Neighbor** |
|---|---|---|---|---|---|
| **Accuracy** | 0.526 / ±0.002 | **0.597** / ±0.026 | 0.550 / ±0.026 | 0.528 / ±0.027 | 0.528 / ±0.027 |
| **Precision** | 0.570 / ±0.009 | 0.572 / ±0.036 | 0.568 / ±0.021 | **0.586** / ±0.041 | 0.499 / ±0.032 |
| **Recall** | 0.398 / ±0.023 | **0.516** / ±0.044 | 0.369 / ±0.163 | 0.360 / ±0.050 | 0.465 / ±0.042 |
| **Specificity** | 0.604 / ±0.031 | 0.581 / ±0.023 | 0.641 / ±0.041 | **0.646** / ±0.039 | 0.582 / ±0.026 |
| **f-Score** | 0.458 / ±0.018 | **0.542** / ±0.035 | 0.490 / ±0.025 | 0.444 / ±0.045 | 0.481 / ±0.034 |
| **MCC** | 0.052 / ±0.024 | **0.171** / ±0.060 | 0.070 / ±0.058 | 0.082 / ±0.057 | 0.046 / ±0.057 |
| **MCEN** | 0.722 / ±0.026 | **0.847** / ±0.022 | 0.782 / ±0.156 | 0.808 / ±0.032 | 0.887 / ±0.015 |

**Table 3.1:** Results achieved by generating the construction work dataset using the utility function and the points generated by normal distribution, in concrete, mean and standard deviation of the evaluation metrics for different machine learning algorithms.

| Confidence Interval | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.525–0.526 | **0.590–0.594** | 0.548–0.551 | 0.515–0.518 | 0.527–0.530 |
| **Precision** | 0.546–0.548 | 0.567–0.571 | 0.567–0.569 | **0.587–0.593** | 0.497–0.500 |
| **Recall** | 0.395–0.401 | **0.513–0.518** | 0.359–0.379 | 0.356–0.363 | 0.462–0.467 |
| **Specificity** | 0.598–0.609 | 0.579–0.582 | 0.636–0.642 | **0.641–0.648** | 0.578–0.583 |
| **f-Score** | 0.456–0.459 | **0.539–0.544** | 0.489–0.491 | 0.441–0.447 | 0.479–0.483 |
| **MCC** | 0.050–0.056 | **0.167–0.175** | 0.066–0.073 | 0.078–0.085 | 0.043–0.050 |
| **MCEN** | 0.712–0.734 | 0.846–0.849 | 0.773–0.792 | 0.806–0.810 | **0.886–0.888** |

**Table 3.2:** Results achieved by generating the construction work dataset using utility function and the points generated by normal distribution, in concrete, confidence interval of the evaluation metrics for different machine learning algorithms.

| Mean / Standard deviation | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.541 / ±0.019 | **0.621** / ±0.027 | 0.602 / ±0.028 | 0.575 / ±0.024 | 0.571 / ±0.027 |
| **Precision** | 0.562 / ±0.102 | **0.609** / ±0.033 | 0.576 / ±0.041 | 0.566 / ±0.016 | 0.548 / ±0.032 |
| **Recall** | 0.119 / ±0.067 | 0.549 / ±0.044 | 0.331 / ±0.184 | **0.823** / ±0.033 | 0.504 / ±0.042 |
| **Specificity** | **0.919** / ±0.041 | 0.682 / ±0.038 | 0.784 / ±0.115 | 0.298 / ±0.040 | 0.626 / ±0.039 |
| **f-Score** | 0.189 / ±0.093 | 0.577 / ±0.034 | 0.453 / ±0.105 | **0.671** / ±0.019 | 0.524 / ± 0.033 |
| **MCC** | 0.061 / ± 0.064 | **0.235**/ ±0.056 | 0.124 / ±0.084 | 0.143 / ±0.057 | 0.132 / ±0.055 |
| **MCEN** | 0.543 / ±0.093 | 0.827 / ±0.021 | 0.711 / ±0.184 | 0.727 / ±0.037 | **0.862** / ±0.017 |

**Table 3.3:** Results achieved by generating the construction work dataset using utility function and the points generated by bimodal distribution, in concrete, mean and standard deviation of the evaluation metrics for different machine learning algorithms.

| Confidence Interval | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.539–0.542 | **0.619–0.623** | 0.601–0.604 | 0.573–0.576 | 0.569–0.573 |
| **Precision** | 0.556–0.568 | **0.606–0.611** | 0.573–0.579 | 0.565–0.567 | 0.546–0.550 |
| **Recall** | 0.115–0.123 | 0.547–0.552 | 0.320–0.343 | **0.821–0.825** | 0.502–0.507 |
| **Specificity** | **0.917–0.922** | 0.680–0.684 | 0.777–0.791 | 0.296–0.301 | 0.623–0.628 |
| **f-Score** | 0.183–0.195 | 0.575–0.579 | 0.447–0.460 | **0.669–0.672** | 0.522–0.527 |
| **MCC** | 0.057–0.065 | **0.231–0.238** | 0.119–0.129 | 0.140–0.147 | 0.128–0.135 |
| **MCEN** | 0.537–0.549 | 0.825–0.828 | 0.699–0.722 | 0.725–0.730 | **0.861–0.864** |

**Table 3.4:** Results achieved by generating the construction work dataset using utility function and the points generated by bimodal distribution, in concrete, confidence interval of the evaluation metrics for different machine learning algorithms.

Then, as our models did not performed that well with such randomness, we tried following some reasoning as we said previously. The results achieved using this method went quite well as we can see the following tables of the mean values and standard deviation for different evaluation metrics in each classification model 3.5. We achieved the best results for our dataset using random forest as we can see the numbers in boldface. On the other hand, the second table 3.6 shows the confidence interval for previous evaluation metrics in each classification.

Analyzing these tables, random forest and decision tree give the best results, since our dataset is a mixture of categorical variables and continuous variables. Thus these models natively handle these predictors without having to transform them. SVM is slightly worse than the random forest and decision tree. However, there is a noticeable deterioration in precision using Gaussian Naive Bayes. Finally K Nearest Neighbor is the worst model, since we used euclidean distance. Here we realized that we were using euclidean distance instead of trying hamming distance or Manhattan distance.

Hamming distance is the number of bit positions in which they differ, while comparing two vectors of equal length. This metric is generally used when comparing texts or binary vectors. While in Manhattan distance, the distance between two points is the sum of the absolute differences of their Cartesian coordinates.

Cartesian coordinates are the numbers that indicate the location of a point relative to a fixed reference point, the origin. Thus, considering our dataset, in future we will try using Manhattan distance.

Testing statistically and applying Kolgomorov Smirnov we got the p-value less than 0.05, thus we reject the null hypothesis, consequently, our data does not follow normal distribution. In consequence, we applied Kruskal Wallis, yet we did not get significant difference between models.

| Mean / Standard deviation | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.679 / ±0.026 | **0.781** / ±0.022 | 0.744 / ±0.024 | 0.684 / ±0.026 | 0.570 / ±0.027 |
| **Precision** | 0.712 / ±0.024 | **0.786** / ±0.024 | 0.708 / ±0.068 | 0.629 / ±0.030 | 0.604 / ±0.021 |
| **Recall** | 0.721 / ±0.035 | **0.832** / ±0.029 | 0.818 / ±0.109 | 0.686 / ±0.041 | 0.661 / ±0.040 |
| **Specificity** | 0.627 / ±0.041 | **0.711** / ±0.041 | 0.537 / ±0.243 | 0.682 / ±0.037 | 0.447 / ±0.041 |
| **f-Score** | 0.716 / ±0.024 | **0.808** / ±0.020 | 0.750 / ±0.039 | 0.656 / ±0.029 | 0.630 / ±0.026 |
| **MCC** | 0.349 / ±0.052 | **0.550** / ±0.048 | 0.364 / ±0.172 | 0.367 / ±0.053 | 0.110 / ±0.054 |
| **MCEN** | 0.782 / ±0.024 | 0.663 / ±0.033 | 0.652 / ±0.162 | 0.775 / ±0.025 | **0.857** / ±0.019 |

**Table 3.5:** Results achieved by generating the predictor class by following some reasoning, in concrete, mean and standard deviation of the evaluation metrics for different machine learning algorithms.

| Confidence Interval | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.678–0.681 | **0.779–0.782** | 0.742–0.745 | 0.682–0.685 | 0.569–0.572 |
| **Precision** | 0.710–0.713 | **0.785–0.788** | 0.704–0.712 | 0.627–0.631 | 0.602–0.605 |
| **Recall** | 0.719–0.723 | **0.831–0.834** | 0.811–0.825 | 0.683–0.688 | 0.658–0.663 |
| **Specificity** | 0.624–0.629 | **0.708–0.713** | 0.522–0.552 | 0.680–0.685 | 0.444–0.449 |
| **f-Score** | 0.714–0.717 | **0.807–0.810** | 0.748–0.753 | 0.654–0.657 | 0.629–0.632 |
| **MCC** | 0.346–0.352 | **0.547–0.553** | 0.353–0.375 | 0.363–0.370 | 0.107–0.114 |
| **MCEN** | 0.780–0.783 | 0.661–0.665 | 0.642–0.662 | 0.773–0.776 | **0.855–0.858** |

**Table 3.6:** Results achieved by generating the predictor class by following some reasoning, in concrete, confidence interval of evaluation metrics for different machine learning algorithms.

For more experiments, we tried unbalancing the predictor class. Initially we had 1204 instances for class 0 and 1536 instances for class 1. First we tried decreasing instances for class 1 randomly, getting 1204 instances for class 0 and 401 instances for class 1. Namely, 75% for class 0 and 25% for class 1. Executing like this, we achieved the following results in the table 3.7 and table 3.8.

When we unbalance reducing the class 1, SVM recall falls over. SVM is very limited when the dataset is imbalanced and how our positive examples are less, it can not predict well positive instances. Also, random forest and decision tree models performs worse when the predictor class is unbalanced, yet recall is what falls a lot when the unbalance is more towards class 1. Nevertheless, Gaussian Naive Bayes performs notably well when the predictor class is unbalanced.

Statistically the data do not follow normal distribution and when we compared the models using Kruskal Wallis, we acquired differences between random forest and decision tree.

| Mean / Standard deviation | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.743 / ±0.021 | **0.787** / ±0.024 | 0.777 / ±0.022 | 0.736 / ±0.028 | 0.722 / ±0.017 |
| **Precision** | 0.458 / ±0.145 | 0.601 / ±0.079 | 0.628 / ±0.122 | **0.802** / ±0.018 | 0.254 / ±0.123 |
| **Recall** | 0.120 / ±0.063 | 0.430 / ±0.082 | 0.227 / ±0.115 | **0.861** / ±0.032 | 0.077 / ±0.045 |
| **Specificity** | 0.950 / ±0.030 | 0.902 / ±0.032 | **0.951**/ ±0.034 | 0.360 / ±0.073 | 0.923 / ±0.032 |
| **f-Score** | 0.207 / ±0.067 | 0.496 / ±0.068 | 0.361 / ± 0.084 | **0.830** / ±0.019 | 0.122 / ± 0.056 |
| **MCC** | 0.117 / ± 0.093 | **0.375** / ±0.077 | 0.252 / ±0.126 | 0.244 / ±0.082 | 0.002 / ±0.001 |
| **MCEN** | 0.499 / ±0.090 | 0.591 / ±0.052 | 0.491 / ±0.098 | **0.663** / ±0.041 | 0.568 / ±0.070 |

**Table 3.7:** Results achieved by unbalancing the predictor class favoring class 0, namely 75% for class 0 and 25% for class 1. The table contains the mean and standard deviation of the evaluation metrics for different machine learning algorithms.

| Confidence Interval | SVM | Random Forest | Decision Tree | Gaussian Naive Bayes | k Nearest Neighbor |
|---|---|---|---|---|---|
| **Accuracy** | 0.741–0.744 | **0.786–0.789** | 0.775–0.778 | 0.734–0.738 | 0.721–0.723 |
| **Precision** | 0.449–0.467 | 0.596–0.606 | 0.620–0.636 | **0.800–0.803** | 0.246–0.262 |
| **Recall** | 0.116–0.124 | 0.425–0.435 | 0.220–0.235 | **0.859–0.863** | 0.075–0.080 |
| **Specificity** | 0.948–0.952 | 0.900–0.904 | **0.949–0.953** | 0.355-0.364 | 0.921–0.925 |
| **f-Score** | 0.203–0.211 | 0.492–0.500 | 0.356–0.367 | **0.829–0.831** | 0.118–0.125 |
| **MCC** | 0.111–0.123 | **0.371–0.380** | 0.245–0.260 | 0.238–0.249 | 0.001–0.007 |
| **MCEN** | 0.494–0.505 | 0.588–0.594 | 0.485–0.498 | **0.660–0.665** | 0.563–0.572 |

**Table 3.8:** Results achieved by unbalancing the predictor class favoring class 0, namely 75% for class 0 and 25% for class 1. The table contains the confidence interval of evaluation metrics for different machine learning algorithms.

Then, we also experimented unbalancing the class 0. Decreasing the class 0 randomly, we got 504 instances for class 0 and 1536 instances for class 1. Thus, the predictor class is composed of 25% of class 0 and 75% of class 1.

The results are considerably higher than the previous experiments, especially the recall, since our instances for positive class is much higher. Moreover, these experiments are done over generated data basing on the probability, that is why the high performance. The models performs quite well except the Gaussian Naive Bayes. Gaussian Naive Bayes results are considerably lower than the expected.

Statistically it do not follow normal distribution and comparing the models we got significant difference between random forest and decision tree.

| **Mean / Standard deviation** | **SVM** | **Random Forest** | **Decision Tree** | **Gaussian Naive Bayes** | **k Nearest Neighbor** |
|---|---|---|---|---|---|
| **Accuracy** | 0.752 / ±0.001 | **0.816** / ±0.023 | 0.816 / ±0.024 | 0.728 / ±0.028 | 0.730 / ±0.017 |
| **Precision** | 0.752 / ±0.001 | **0.854** / ±0.018 | 0.848 / ±0.031 | 0.456 / ±0.051 | 0.760 / ±0.009 |
| **Recall** | **1** / ±0 | 0.909 / ±0.023 | 0.909 / ±0.034 | 0.504 / ±0.077 | 0.917 / ±0.030 |
| **Specificity** | 0.000 / ±0.000 | **0.526** / ±0.068 | 0.499 / ±0.145 | 0.801 / ±0.035 | 0.119 / ±0.045 |
| **f-Score** | 0.859 / ±0.001 | **0.880** / ±0.015 | 0.877 / ± 0.016 | 0.477 / ±0.055 | 0.831 / ± 0.015 |
| **MCC** | 0 / ± 0 | **0.472** / ±0.067 | 0.439 / ±0.126 | 0.296 / ±0.071 | 0.057 / ±0.072 |
| **MCEN** | 0.306 / ±0.000 | 0.563 / ±0.042 | 0.555 / ±0.072 | **0.694** / ±0.030 | 0.583 / ±0.062 |

**Table 3.9:** Results achieved by unbalancing the predictor class favoring class 1, namely 25% for class 0 and 75% for class 1. The table contains the mean and standard deviation of evaluation metrics for different machine learning algorithms.

| **Confidence Interval** | **SVM** | **Random Forest** | **Decision Tree** | **Gaussian Naive Bayes** | **k Nearest Neighbor** |
|---|---|---|---|---|---|
| **Accuracy** | 0.752–0.753 | 0.814–0.817 | **0.816–0.819** | 0.726–0.730 | 0.729–0.731 |
| **Precision** | 0.752–0.753 | **0.853–0.855** | 0.846–0.850 | 0.453–0.459 | 0.759–0.761 |
| **Recall** | **1–1** | 0.907–0.910 | 0.907–0.911 | 0.499–0.509 | 0.915–0.919 |
| **Specificity** | 0.000–0.000 | 0.522–530 | 0.490–0.508 | **0.799–0.803** | 0.116–0.122 |
| **f-Score** | 0.858–0.859 | **0.879–0.881** | 0.876–0.878 | 0.473–0.480 | 0.830–0.832 |
| **MCC** | 0–0 | **0.468–0.476** | 0.431–0.447 | 0.292–0.301 | 0.052–0.061 |
| **MCEN** | 0.306–0.306 | 0.561–0.566 | 0.550–0.559 | **0.692–0.696** | 0.579–0.586 |

**Table 3.10:** Results achieved by unbalancing the predictor class favoring class 1, namely 25% for class 0 and 75% for class 1. The table contains the confidence interval of evaluation metrics for different machine learning algorithms.

CHAPTER 4

# Conclusion

We have introduced a classification problem based on machine learning algorithms, where we had to classify a download if it is going to lead to a construction work or not using different machine learning models. The different algorithms has been demonstrated over this concrete geospatial dataset, that includes the download dataset and the construction work dataset. Also, this project can be used in datasets of different areas, since those datasets will have a similar structure. Therefore, we can apply the previous steps to take advantage of the prediction, such as making most out of the construction area, making business plan for the downloading company, etc.

## 4.1 Limitations

During the realization of this project we had one major limitation that prevented a optimal execution of the project in real life, since we could not acquire the construction work dataset, which we had to associate with the download dataset. This conditioned the whole project, after all we had to rely on artificial data based on some reasoning, that clearly affected the results of the project.

## 4.2 Future works

In future works, first we will carry out the previous steps using the real dataset for construction works. Thus, our results will have more meaning. Then, crossing the construction work dataset with the data of the use of credit cards, we will analyze the economic impact of the construction work on local commerce. Hence, it can help on planning financial aid to the commerces of the area. Finally, we will use the predictions for a business plan, such as reaching the materials company before the construction work, using the predicted construction work area for additional plans, etc.

# Bibliography

[1] Jae-Gil Lee and Minseo Kang. Geospatial big data: Challenges and opportunities. *Big Data Research*, 2(2):74–81, 2015. Visions on Big Data. See page 2.

[2] Martin Breunig, Patrick Erik Bradley, Markus Jahn, Paul Kuper, Nima Mazroob, Norbert Rösch, Mulhim Al-Doori, Emmanuel Stefanakis, and Mojgan Jadidi. Geospatial data management research: Progress and future directions. *ISPRS International Journal of Geo-Information*, 9(2), 2020. See page 2.

[3] Fabian Dembski, Uwe Wössner, Mike Letzgus, Michael Ruddat, and Claudia Yamu. Urban digital twins for smart cities and citizens: The case study of herrenberg, germany. *Sustainability*, 12(6):2307, 2020. See page 2.

[4] Lin Liu and John Eck. *Artificial Crime Analysis Systems: using computer simulations and geographic information systems.* 01 2008. See page 2.

[5] Meysam Effati, Jean-Claude Thill, and Shahin Shabani. Geospatial and machine learning techniques for wicked social science problems: analysis of crash severity on a regional highway corridor. *Journal of Geographical Systems*, 17(2):107–135, Apr 2015. See page 2.

[6] Hossein Mojaddadi, Biswajeet Pradhan, Haleh Nampak, Noordin Ahmad, and Abdul Halim bin Ghazali. Ensemble machine-learning-based geospatial approach for flood risk assessment using multi-sensor remote-sensing data and gis. *Geomatics, Natural Hazards and Risk*, 8(2):1080–1102, 2017. See page 2.

[7] Georgios Kouziokas. An information system for judicial and public administration using artificial intelligence and geospatial data. 09 2017. See page 2.

[8] Yongyao Jiang, Yun Li, Chaowei Yang, Fei Hu, Edward M. Armstrong, Thomas Huang, David Moroni, Lewis J. McGibbney, and Christopher J. Finch. Towards intelligent geospatial data discovery: a machine learning framework for search ranking. *International Journal of Digital Earth*, 11(9):956–971, 2018. See page 2.

[9] Trang VoPham, Jaime E. Hart, Francine Laden, and Yao-Yi Chiang. Emerging trends in geospatial artificial intelligence (geoai): potential applications for environmental epidemiology. *Environmental Health*, 17(1):40, Apr 2018. See page 2.

[10] Mahyat Shafapour Tehrany, Simon Jones, Farzin Shabani, Francisco Martínez-Álvarez, and Dieu Tien Bui. A novel ensemble modeling approach for the spatial prediction of tropical forest fire susceptibility using logitboost machine learning classifier and multi-source geospatial data. *Theoretical and Applied Climatology*, 137(1):637–653, Jul 2019. See page 2.

[11] Anna Kovacs-Györi, Alina Ristea, Clemens Havas, Michael Mehaffy, Hartwig H. Hochmair, Bernd Resch, Levente Juhasz, Arthur Lehner, Laxmi Ramasubramanian, and Thomas Blaschke. Opportunities and challenges of geospatial analysis for promoting urban livability in the era of big data and machine learning. *ISPRS International Journal of Geo-Information*, 9(12), 2020. See page 3.

[12] Joel Podgorski, Ruohan Wu, Biswajit Chakravorty, and David A. Polya. Groundwater arsenic distribution in india by machine learning geospatial modeling. *International Journal of Environmental Research and Public Health*, 17(19), 2020. See page 3.

[13] Jürgen Döllner. Geospatial artificial intelligence: Potentials of machine learning for 3d point clouds and geospatial digital twins. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1):15–24, Feb 2020. See page 3.

[14] A review of recent researches and reflections on geospatial artificial intelligence. *Geomatics and Information Science of Wuhan University*, 45(1671-8860(2020)12-1865-10):1865, 2020. See page 3.

[15] Mihai Andronie, George Lăzăroiu, Mariana Iatagan, Cristian Uță, Roxana Stefanescu, and Mădălina Cocoșatu. Artificial intelligence-based decision-making algorithms, internet of things sensing networks, and deep learning-assisted smart process management in cyber-physical production systems. *Electronics*, 10:2497, 10 2021. See page 3.

[16] Mohammad Ahsanullah, BM Kibria, and Mohammad Shakil. Normal distribution. In *Normal and Student st Distributions and Their Applications*, pages 7–50. Springer, 2014. See page 7.

[17] Edmond A Murphy. One cause? many causes?: The argument from the bimodal distribution. *Journal of Chronic Diseases*, 17(4):301–324, 1964. See page 7.

[18] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3), jul 2017. See page 12.

[19] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002. See page 12.

[20] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of database systems*, 5:532–538, 2009. See page 13.

[21] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995. See page 13.

[22] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006. See page 14.

[23] Martin Hofmann. Support vector machines-kernels and the kernel trick. *Notes*, 26(3):1–16, 2006. See page 14.

[24] J Ross Quinlan. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, 28(1):71–72, 1996. See page 14.

[25] Adele Cutler, D Richard Cutler, and John R Stevens. Random forests. In *Ensemble machine learning*, pages 157–175. Springer, 2012. See page 15.

[26] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998. See page 15.

[27] Ali Haghpanah Jahromi and Mohammad Taheri. A non-parametric mixture of gaussian naive bayes classifiers based on local independent features. In *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pages 209–212. IEEE, 2017. See page 15.

[28] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer, 2003. See page 16.

[29] Rosario Delgado and J. David Núñez-González. Enhancing confusion entropy (cen) for binary and multiclass classification. *PLOS ONE*, 14(1):1–30, 01 2019. See page 20.

[30] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951. See page 20.

[31] Patrick E McKight and Julius Najab. Kruskal-wallis test. *The corsini encyclopedia of psychology*, pages 1–1, 2010. See page 20.