Bachelor Thesis

The power of machine learning and its application to classification problems in the start-up VivaDrive

Stefanos Kypritidis

Supervisor:
Professor Panos Louridas

Warsaw 2022

Summary

The increasing use of technology, and in particular machine learning, to solve a variety of problems is relevant to all fields. Not only does it make everyone's life easier, but it can also contribute to making important decisions. This paper starts with an introduction to machine learning, its various models, and terms. It also includes an extensive analysis of data and machine learning. In particular, it analyzes and addresses three separate classification problems of a real start-up company regarding the replacement of conventional vehicles with suitable electric ones. For each problem, a multitude of different algorithms and machine learning techniques are tested to find the combinations of those that address each problem most effectively. It is worth noting that only a few of the possibilities of machine learning are shown by this analysis. However, it hints at the zillions of possibilities for its application in the present and the future.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

In today's world, technology is an integral part of our daily lives. The internet, computers, smartphones, and smart-devices in general are objects of daily use for most of us. However, alongside their use, a huge amount of data is generated, which needs to be stored as it can lead to the acquisition of valuable knowledge. In particular, machine learning is one of the ways of gaining intelligence from data. According to [1], machine learning is not only about storing large amounts of data, but it is also part of artificial intelligence. In detail, artificial intelligence is used to improve computer programs to perform tasks that require human involvement, such as decision-making.

Machine learning is a vast interdisciplinary field that builds on concepts from computer science, statistics, cognitive science, engineering, optimization theory, and many other disciplines from mathematics and science [2]. In addition, machine learning can be applied to various areas of our lives such as healthcare, manufacturing, education, financial modeling, policing, and marketing [3]. A typical example of a machine learning application is the detection and filtering of spam emails [1].

The purpose of this paper is to analyze the impact of technology and specifically machine learning on solving classification problems and facilitating decision making. To do this, a case analysis of several real-world classification problems was implemented within the technology company VivaDrive. This start-up is a technology company that deals with vehicle fleet management and the gradual replacement of conventional vehicles with electric ones. In detail, the case analysis constitutes a data and machine learning analysis since data are explored and then machine learning models of classification are implemented to predict vehicles suitable for replacement with electric vehicles.

The case analysis focuses on the solution of three separate and distinct classification problems. The first problem concerns the building of machine learning models to predict conventional vehicles as suitable or unsuitable for replacement by electric vehicles. Additionally, the second problem is associated with building models that are capable of predicting how suitably a conven-

tional vehicle can be replaced by an electric vehicle. Finally, the third problem relates to building models to predict all brands of electric vehicles that can replace the considered conventional one.

Chapter 2

Literature Review

This section includes a literature review on machine learning. In detail, the existing knowledge is presented, the different types of machine learning are described, and the terms and models used in this paper are analyzed. As a result, the terms and the use of the models are comprehended, following a smooth transition from chapter to chapter and finally understanding the importance and power of machine learning. Specifically, in the first subsection, the division of machine learning into supervised, semi-supervised, and unsupervised learning is examined, and each type is analyzed. In the second subsection, the different types of classification follow, explaining each one of them. Next, the different algorithms used for classification are examined, highlighting the advantages and disadvantages of each. Finally, various machine learning terms that are used in the Vivadrive case analysis are discussed.

2.1 Machine Learning Types

2.1.1 Supervised

According to [4], supervised machine learning models try to discover the relationship between input features (independent variables) and a specific target feature (dependent variable). These models learn from a set of pre-labeled data, in other words having for each data the label (target) that corresponds to it. Thus, from the data and their corresponding labels, the model eventually constructs its own probabilistic mapping system (from data to label) to be used for new data inputs [5].

Some applications of supervised learning are diagnostics, personality prediction [6], student performance prediction [7], and image recognition [8]. In fact, according to [2], supervised machine learning is further divided into two categories, classification and regression. More specifically, in regression the target feature takes continuous values while in classification the target

feature takes labels corresponding to different classes. Finally, decision trees, Naïve Bayes, support vector machine and random forest are examples of classification algorithms, while linear regression is an example of a regression algorithm [6].

2.1.2 Unsupervised

In this particular type of machine learning, there is no labeled data and clear instructions for pre-training a particular model. Thus, in these models training is absent. In particular, the analysis is performed based on the existing data and at the same time focuses on the common features and structures in the group [9]. Similarly, according to [6], unsupervised learning models are given a huge amount of data along with specific instructions to find some pattern among the data.

Regarding the output of these models, the data can be organized into different types, such as clustering, anomaly detection, correlation, and automatic coding [10]. Additionally, various popular applications of unsupervised machine learning include customer segmentation, targeted marketing, and recommendations such as on the YouTube homepage [6]. Finally, the main unsupervised learning algorithms are K-means clustering, K-Nearest neighbors (K-NN), principal component analysis (PCA), and association rules [11].

2.1.3 Semi-Supervised

In this approach, the system is trained not only with labeled data but also with unlabeled data in the testing phase, while at the same time, both types of machine learning are used. The main goal of this approach is to achieve better accuracy and precision compared to traditional types of supervised and unsupervised learning [10]. Accordingly, according to [3], semi-supervised learning uses unlabeled data to augment labeled data within a supervised learning model and can even use unsupervised learning architectures in combination with optimization procedures that make use of labels. Additionally, semi-supervised clustering and semi-supervised classification are the two different output types that exist [10]. Finally, some examples of algorithms of this approach are self-training, active learning, and expectation maximization [12].

2.2 Classification Types

According to [13], classification can be distinguished into binary, multi-class, and multi-label classification.

2.2.1 Binary classification

According to [14], binary classification plays an important role in the machine learning process. In detail, binary classification is used for binary problems, in other words, situations where the prediction constitutes a yes or a no decision [15]. Hence, the specific type is used for classifications where there are only two class-labels [13]. Additionally, according to [16], when there are only two classes and at the same time there is a reasonable level of balance between the data of the two classes, the use of binary classification is strongly recommended.

As already mentioned, binary classification includes a normal state class and another abnormal state class. Some examples of binary classification are spam detection, conversion prediction (purchase or no purchase), cancer detection [13], and fake profile detection [14]. Lastly, popular algorithms used for binary classification problems are decision trees, logistic regression, Naïve Baynes, and Support Vector Machines [13].

2.2.2 Multi-class classification

According to [17], as more emphasis is placed on data aggregation by institutions, multi-class problems are becoming increasingly important. Specifically, multi-class classification refers to classification where there are more than two class-labels. In fact, the concept of normal and abnormal is absent since the data is assigned to one of the various known classes [13].

According to [13], the number of class labels varies depending on the type of problem. For example, a face recognition model can predict that a photo belongs to one person among the thousands or tens of thousands of people in the system. In contrast, in the brain cancer classification example, there are five different class-labels of which only one corresponds to a normal sample [18]. Finally, decision trees, random forest, Gradient Boosting, and Naïve Baynes are among the most popular algorithms for multi-class classification problems [13].

2.2.3 Multi-label classification

Multi-label classification differs from both binary and multi-class classification since each instance can be associated with several class-labels [19]. In fact, multi-label classification is increasingly required in applications such as protein function classification, music classification [20], text classification, video classification and bioinformatics [19]. Finally, a simple example of multi-label classification is the categorization of a song that can belong to different musical genres such as rock and ballad [20].

Multi-label problems are handled either by the method of problem transformation or by the method of algorithm adaptation [21]. More specifically, adaptations of traditional classification algorithms used to solve multi-label problems are called multi-label versions. Some examples

are multi-label decision trees, multi-label Random Forests, and multi-label Gradient Boosting [13]. In contrast, with the problem transformation, a multi-label problem is transformed into one or more binary or multi-class problems [19].

In detail, all different approaches to solving multi-label type problems are presented:

- Binary Relevance is a problem transformation approach that treats each label as a separate binary classification problem [22].
- Classifier Chains is another problem transformation approach where the first classifier is trained only on the input data and then each subsequent classifier is trained on the input data but also on all the previous classifiers in the chain [22].
- Label Powerset is also a problem transformation method where the problem is transformed from multi-label to multi-class with a classifier trained on all unique combinations of labels found in the training data [22].
- Custom algorithms adapt the algorithm to perform multi-label classification directly, instead of breaking the problem into different subsets of problems. An example of such an algorithm is MLkNN which is the multi-label version of k-NearestNeighbors [22].
- Ensemble methods can also solve multi-label problems usually producing better results [22].
- Neural networks are another example of an algorithm that supports multi-label classification problems and can even be tuned to solve them effectively [23].

2.3 Classification Algorithms

Classification problems can be tackled by various variants of machine learning algorithms with accurate and desirable outcomes [24]. The presence of the open source library Scikit-learn greatly facilitates researchers to solve classification problems using machine learning algorithms and their variants [25]. At this point, various machine learning algorithms and variants will be discussed along with their advantages and disadvantages.

2.3.1 Decision Trees

Decision trees are one of the algorithms used to a great extent in statistics and machine learning [1]. Certainly, it is one of the simplest machine learning algorithms since it creates association rules to find and predict target labels [24]. Specifically, it is a hierarchical design that applies the divide-and-conquer approach, while at the same time, it is a non-parametric technique [1]. Moreover, in trees, data can be modeled in hierarchical structures using a series of if-else comparisons [26].

Decision trees are used not only on large but also on small datasets, and they can handle both numerical and categorical data [27]. Furthermore, decision trees are usually constructed in two phases, tree growth and tree pruning [2]. Illustratively, tree growth involves iteratively splitting the training data based on optimal criteria until the majority of records belong to a class-label [28]. In contrast, tree pruning is about reducing the size of the tree to make it easier to understand [29]. Additionally, according to [30], tree pruning can also be used to deal with the overfitting phenomenon where the algorithm learns to perfectly categorize all training data.

According to [4], decision trees show several advantages as a classification algorithm. To begin with, trees contain self-evident logic and are very easy to follow when compressed. In fact, when they are of reasonable size, they are easily understood by non-professional users. This is also achieved by the fact that they are easily converted into a set of rules [4]. Additionally, trees can easily handle not only outliers [31] but also missing values [4]. Finally, since decision trees are a non-parametric technique, there is no need for any functional form parameterization [31].

However, according to [1], trees also present limitations. First of all, sometimes trees can be computationally expensive. In addition, trees can easily end up overfitting the data when it is not attempted to avoid it. Besides, according to [32], sometimes you get very complex trees that don't generalize well. At the same time, decision trees can be unstable since small variations in the data can lead to the creation of a completely different tree. Finally, trees are practically more used for classification problems and are less suitable for regression problems [1].

2.3.2 Support Vector Machines

Support Vector Machines (SVMs) are one of the most important and convenient techniques for solving data classification problems [2]. At the same time, SVMs are also a well-established and frequently used classification technique [10]. According to [33], no matter that new classification techniques are proposed, SVMs remain one of the most popular and widely used classification algorithms. Although initially developed for binary classification problems, multi-class classification extensions have been implemented.

SVMs can be used for any number of vector dimensions of data [10]. In detail, in classification SVMs determine an optimal separating hyperplane using the concept of margin. The margin is essentially the distance between the hyperplane and the nearest points on either side of it. The aim is to maximize the margin for better generalization of the data points [1]. More precisely, in the two dimensions the hyperplane is a line [10].

Using SVMs offers several advantages in classification problems. First of all, according to [32], SVMs are efficient in high-dimensional spaces. In addition, the algorithm is also memory efficient since it only uses a subset of training points to find the hyperplane. Additionally, another advantage is the ability of SVMs to deal with a variety of classification problems such as high-dimensional problems and non-linear separable problems [2]. Besides, training is relatively easy, the model can be used with both continuous and categorical data, and it can deal with data

errors. Still, the trade-off between model complexity and error is easily controlled. Finally, the accuracy of predictions is very high, good generalizations are made and a unique optimal solution is produced [1].

At the same time, the use of SVMs comes with some disadvantages. Firstly, the algorithm does not directly provide probability estimates, so these are calculated using the exact five-fold cross-validation [32], which is discussed next. Also, achieving excellent results requires the correct setting of some basic parameters [2]. Besides that, SVMs are very difficult to interpret unless the features of the data are interpretable. Finally, sometimes using SVMs can be computationally expensive [1].

2.3.3 K-Nearest-Neighbors

The K-Nearest-Neighbors (KNN) algorithm is based on the principle that within a dataset, data observations are in close proximity relative to other observations that have similar characteristics [30]. In particular, data observations are represented in an n-dimensional space, where n is the number of data attributes. Thus, a new point is classified according to its similarity to the rest of the data points already stored known as training data [1]. Specifically, K specifies the number of nearest neighbors to be examined in order to classify the observation being examined [2]. Finally, for K≥3, a vote is taken to implement the classification of the new observation based on the most common class among the K nearest neighbors [1].

The choice of K affects the performance of the KNN algorithm. Illustratively, when there is noise in the data, a small K may result in the noisy cases winning the majority [30]. Conversely, if K is too large, the algorithm may misclassify the new observation because the majority of nearest neighbors may be too far from that particular observation [1].

The KNN algorithm shows its own advantages. First, the algorithm is efficient for large data sets and simultaneously suitable for noisy training data. In addition, KNN is characterized by simplicity and transparency and thus can be easily implemented and understood [2]. Finally, the training process is very fast and has zero cost [1].

On the other hand, KNN is characterized by several disadvantages. Firstly, the computational cost is high as it is required to calculate the distance of each test observation from all training samples. Also, the number K of nearest neighbors is required to be determined [32] which can make a big difference as mentioned earlier. In addition, features of the data that are not very important may cause problems in the results of KNN [2]. Finally, according to [1], the algorithm needs huge memory to store all the training data.

2.3.4 Naïve Bayes

The Naïve Bayes (NB) classifier is one of the most commonly used supervised machine learning algorithms [10]. Also, NB is often used for text categorization such as file categorization and spam detection. Alongside, NB calculates a set of probabilities from value combinations of the dataset [33].

More specifically, the operation of NB is based on the Bayes' theorem. This theorem attempts to calculate the probability of an event occurring based on relevant prior knowledge and conditions [10]. Effectively, NB assumes independence between the different features of the data. In fact, with the help of statistical methods, the probabilities of an input sample corresponding to the various classes are calculated. Finally, the output of the algorithm is the class with the highest probability [12].

Various advantages come with the use of NB. Firstly, the main advantage of NB is the short computational time for the training process [30]. Also, the decision-making process in NB is very fast in comparison with other classifiers and works with good results even with a small amount of training data [33]. Besides, NB's algorithm is extremely scalable, fast, and easy to implement in a system [10]. Also, according to [34], NB not only requires little storage space during both training and classification but it also requires much fewer parameters than other classifiers such as SVM. Finally, elements from several features are taken into account to make the final prediction and at the same time, NB can handle missing values and noisy data.

However, the use of NB also comes with several problems. Firstly, the main disadvantage is that it can only be used if the data features are completely independent of each other. In practice, this is not always possible [10]. However, according to the research of [34], NB can perform satisfactorily in some cases where the features are not fully independent but with less accurate estimates and reduced performance. This is because the algorithm may increase the influence of the two interdependent features and decrease the influence of the others, leading to bias in the classification. Finally, NB may experience problems due to hypersensitivity to unnecessary or irrelevant features.

2.3.5 Logistic regression

Logistic regression is one of the simplest machine learning algorithms. Specifically, it is used in various classification problems such as text analysis, data mining, and information retrieval [35]. Indeed, logistic regression examines the relationship between a binary (dependent) variable such as the presence or absence of disease, and various predictor (explanatory or independent) variables such as patient demographics [36]. However, logistic regression can be multinomial in nature, in other words, it has three or more class-labels. Finally, the algorithm is known for its ability to predict the probability of the target variable [35] using the basic logistic function [33].

Several benefits and drawbacks are included in the use of logistic regression. Firstly, logistic regression is ideal for binary classification problems since it was designed for them [33]. In addition, it is very useful for understanding the influence of several independent variables on a single target variable. On the other hand, all predictor variables are assumed to be independent of each other [32]. Hence, before running the algorithm all-important independent variables must be identified, and irrelevant and dependent ones should be removed [33]. Finally, logistic regression requires the absence of missing data – missing values [32].

2.3.6 Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) algorithm is a simple and very efficient approach to fitting linear models [32]. More specifically, it is based on the function of convex loss functions of SVM and logistic regression. It proves to be a powerful classifier for multi-class problems since it combines multiple binary classifiers with the one-vs-all method [24]. Finally, the algorithm repeatedly examines the training data and each time a training sample is used to modify the parameters according to the slope of the error relative to the single training sample [37].

SGD notes its own advantages and disadvantages. To begin with, SGD is a relatively easy-to-understand algorithm. Also, it can be used efficiently on large datasets since it uses only one training sample (batch size) per iteration [24]. Furthermore, according to [32], SGD is characterized by efficiency and ease of implementation and supports various loss functions and classification penalties. However, to achieve good results the SGD hyperparameters must be properly tuned. Finally, the algorithm can present problems with noisy data since the iteratively selected training samples are random, and at the same time, SGD shows sensitivity to feature scaling [24].

2.3.7 Random Forest

The random forest (RF) algorithm is an ensemble learning method that uses a multitude of decision trees during training [38]. In particular, RF randomly selects subsets of features from the total training set to generate multiple decision trees and then returns the average prediction of all trees [35]. Finally, RF generally trains better models than traditional machine learning models [4].

The use of RF comes with its benefits and limitations. Firstly, RF solves the problem of overfitting during training that had been cited as a drawback of decision trees [35]. Besides that, the RF algorithm has more accurate results than decision trees in most cases [32]. On the other hand, with the use of RF, there is no longer any tree that can be interpreted making the interpretability which is an advantage of decision trees very difficult [4]. Finally, RF is a complex algorithm that is difficult to implement and has slow real-time predictions [32].

2.3.8 Extra Trees Classifier

The extra trees algorithm, also called extremely randomized forest, is an ensemble learning type as well. In detail, a set of unpruned decision trees is created where both features and separation are chosen randomly [39]. In fact, the operation of the algorithm is similar to the random forest one but differs in the way of creating trees in a decision tree forest. Specifically, random samples of the K best features are used for the decision, and the Gini criterion is used to select the best feature for data partitioning in the tree. Thus, decorrelated trees are constructed with this approach [24].

Extra trees have their pros and cons. Firstly, the algorithm is known for its high accuracy and computational efficiency [40]. In addition, the algorithm can deal with outliers, identify important features among irrelevant ones, and be used in large-scale mining applications [39]. In contrast, extra trees exhibit greater complexity since the number of leaves on extra trees is 1.5–3 times more than that of Random Forests. Finally, the extra-trees algorithm has several parameters that need to be tuned, which play a crucial role in producing good results [41].

2.3.9 Neural Networks

Neural networks are a well-known tool and belong to the field of artificial intelligence. Additionally, there are several variants of neural networks, and their applications cover areas such as pattern recognition, localization, and classification problems [35]. Neural networks are inspired by the neurons of the human brain where the nodes are interconnected in such a way that the output of each node is transformed into the input of another node. Although each node receives more than one input, the output produced consists of a single value [6].

Several advantages characterize the use of neural networks. First of all, neural networks tend to produce better results when there are multiple dimensions and continuous features [30]. Moreover, they can be used to solve linear and non-linear programming problems, while they are successful in solving different kinds of problems such as classification, clustering, and regression. Finally, they are characterized as powerful and flexible since they learn from the training data without requiring knowledge of their production process [1].

On the other hand, neural networks also have their drawbacks. First, the presence of irrelevant features can lead to inefficient training of the neural network. Also, a large amount of data is required to achieve maximum prediction accuracy [30]. Finally, choosing the optimal neural network architecture usually cannot be known in advance since neural networks involve the process of trial and error [1].

2.4 Machine Learning Terms

As already mentioned in the supervised learning subsection, a set of pre-labeled data (training data) is used initially to train the model appropriately. Then, after the training is complete and the probabilistic mapping system is constructed from data to label-class, it can be tested on new data inputs known as testing data [5].

2.4.1 Metrics

To evaluate the performance of each algorithm in terms of prediction results, there are several metrics [42, p. 214]:

- The metric accuracy is defined as

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

  where TN is the number of true negatives, TP is the number of true positives, FN is the number of false negatives, and so on. In detail, the accuracy metric calculates how often the specific classifier predicts correctly. This metric is very useful when there is a balance of data in the classes predicted [43].

- The precision metric is defined as

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

  This metric explains how many of the cases predicted as positive turned out to be positive. Precision is useful in cases where false positives have worse consequences than false negatives [43].

- The recall metric is defined as

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

  Recall explains how many of the actual positive cases were correctly predicted with the model. It is also a useful metric in cases where false negatives have a worse impact than false positives, as often in medical cases [43].

- The F-measure metric is defined as

$$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

  This metric is the harmonic mean of precision and recall. In fact, it is maximized when precision is equal to recall. In addition to this, the F-measure penalizes outliers more. Finally, it is useful when false positives and false negatives cost the same and when the number of true negatives is high [43].

The hamming loss metric is characterized as the most common metric in the literature related to multi-label classification [44]. Specifically, it counts the number of labels for which the prediction was wrong and then normalizes it [45]. As a result, the lower the value of the hamming loss metric, the better the results. This means that the model performance is perfect when the value of the metric is zero [44].

2.4.2 Hyperparameters tuning

Another important term that is used later is grid search. Grid search is a traditional hyperparameter tuning method [46]. It essentially tests all combinations of a given set of hyperparameters [47] to detect the best ones while using cross-validation to measure the performance of the results [48]. However, a drawback of grid search is that the models being tested grow exponentially with the number of hyperparameters [47].

Another hyperparameter tuning technique is Bayesian optimization with Gaussian processes. This technique assumes that the evaluation function of the machine learning model follows a multivariate Gaussian distribution. Thus, smarter choices are made regarding the selection of the next hyperparameter to evaluate, compared to the grid-search approach [49].

2.4.3 Cross validation

According to [50], the term cross-validation describes a method of evaluating and comparing learning algorithms. In particular, this method divides the data into a part used for training the model and another part used for validation of the model. The basic form of cross-validation is k-fold cross-validation where the data is divided into k equal (or approximately equal) parts known as folds. Consequently, k iterations of the model follow where in each iteration there is a different fold for validation, while at the same time, the remaining k-1 folds are used for training.

A specific case of k-fold cross-validation is the leave-one-out cross-validation. In this case, k is equal to the total number of observations. Effectively, in each iteration, a single observation is used as a fold for validation and all remaining observations are used for training. With this method, the testing error is approximately an unbiased estimate of the true prediction error. However, the computational cost of this approach can be particularly high for large data sets [51].

2.4.4 Oversampling techniques

Oversampling techniques solve the issue of data imbalance. This is achieved by generating additional training samples for the class with a minority of samples and focusing on improving the performance of the classifier [52]. In detail, in the random-oversampler technique, the samples are duplicated at random times and combined with the sample population of the majority class.

On the other hand, the smote technique produces new artificial samples using the feature space instead of repeating existing samples [53].

2.4.5 PCA

Principal Component Analysis (PCA) is one of the most popular multivariate statistical techniques and is used in most scientific disciplines [54]. In detail, PCA is a mathematical algorithm that reduces the dimensionality of the data, retaining most of the variance in the dataset. This is achieved by identifying the principal components which are directions along which the variance in the data is maximum [55].

Chapter 3

Case Analysis

3.1 Dataset

The specific analysis was completed in the context of the Polish start-up company VivaDrive. The data used belongs to the specific company and stems from the use of IoT technologies in fleet vehicles of another client company. After the necessary data is collected for the appropriate period, various metrics and optimizations are calculated through the company's algorithm and finally specific electric vehicles are recommended according to their suitability. The company's goal is to replace as many conventional vehicles with electric ones as possible.

The initial data concerns two datasets. Firstly, the first dataset contains all vehicle profiles and consists of 981 rows and 20 columns. In particular, a vehicle profile is considered to be a unique vehicle tracked for a specific period of time. The same vehicle can correspond to several vehicle profiles when it is tracked for different periods. Hence, the vehicle profile dataset contains data about the vehicle driving for the relevant time period. Additionally, the second dataset concerns the vehicles themselves. Specifically, it includes 72 rows and 53 columns. This means there is data for 72 unique vehicles. Finally, the second dataset has information about the characteristics of the vehicle and general information about its driving.

The vehicles in the dataset are located in Poland since both the start-up and the company that owns the vehicles are Polish. Also, vehicle profiles contain data for a duration of one month or three months. Aggregately, there is data for vehicle profiles from June 2020 until July 2021.

3.2 Data pre-processing

Before the execution of any analysis and machine learning model, the data must be collected, cleaned, and transformed into the appropriate format. This particular step required a lot of effort and time since the initial datasets included a lot of useless information, columns in the wrong formats, and data that needed to be processed into aggregated features.

Starting with the attributes with useless information, the columns with fleet code, vehicle code, and last update were deleted. In addition, new columns were created for the most frequent driver of the vehicle and the start month, in which the profile was being tracked. At the same time, several new columns were created for statistical characteristics of the vehicle which were previously stored in a column in dictionary format. Also, average, percentage, minimum, and maximum metrics such as average distance and percentage of trips with an electric vehicle charger nearby were constructed from the trip data. Finally, vehicle profiles with limited road speed limit information, limited traffic activity, and limited recorded trip speeds were removed.

The final dataset contains 934 rows and 73 columns. The last column concerns the target attribute, in other words, the label-class to be predicted. Of course, the target attribute column has a different format and values for each distinct type of classification problem to be solved. Below is the table that describes the various columns.

Table 3.1: Attribute descriptions

Attribute	Description
monitoring start month	first month of profile monitoring
monitoring days	number of days of monitoring the profile
active days	number of days the vehicle made at least one trip
data quality score	metric 1-10 describing the sufficiency and quality of the trips to draw accurate conclusions
reliability score	metric 1-10 of the variability of a driver's daily pattern
car use score	metric 1-10 describing the drop in range of the potential electric vehicle due to adverse temperature conditions and roads driven by the vehicle
driver behavior score	1-10 metric describing the suitability of a driver to drive an electric vehicle in terms of their behavior (overspeeding tendencies) and speed (exceeding the cutoff point of electric vehicles of 90 Km/h)
number of trips	number of recorded trips
fuel costs	total fuel costs
body type	car body type
driver name	vehicle driver name of majority of trips
fuel cost per trip	average fuel cost per trip

Continued on the next page

Table 3.1 – continued from the previous page

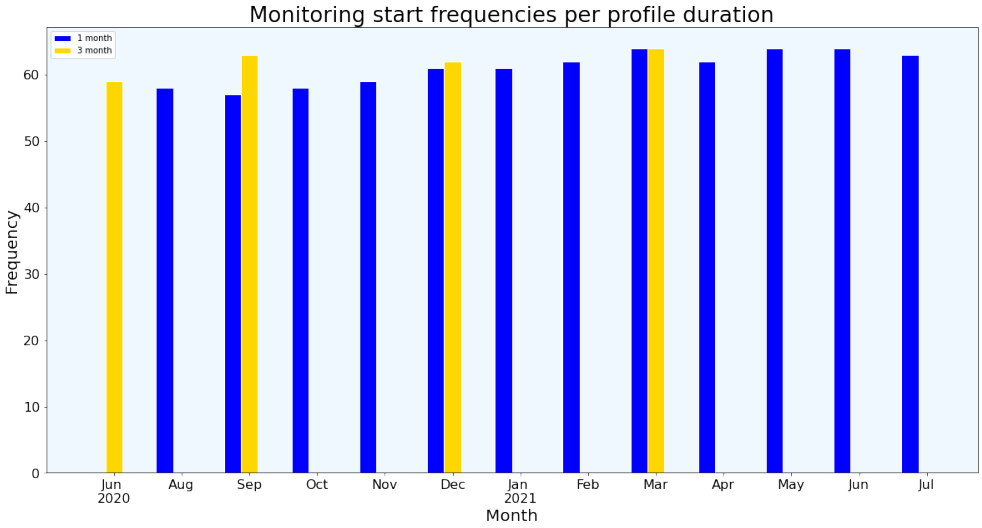
Attribute	Description
distance per speed limit 0-30, 30-50, etc.	total distance covered per category of speed limits
speed limit distribution 0-30, 30-50, etc.	distance distribution per speed limit category
distance per temp class 0-10, 10-20, etc.	total distance covered per category of temperature
distance with temp ratio	total distance ratio where the temperature was known
states availability	percentage of trips with recorded speed
range drop speed	percentage of range that will be lost due to road conditions the vehicle was driven
range drop temp	percentage of range that will be lost due to temperature conditions the vehicle was driven
total distance	total trip distance
distance with limits ratio	distance ratio with known speed limits
tot overspeeding distance	total distance where the driver exceeded the speed limit speed
overspeeding ratio	distance ratio where the driver exceeded the limit speed
overspeeding distance per speed limit 0-30, 30-50 etc.	distance the driver exceeded the speed limit per speed limit category
overspeeding ratio per speed limit 0-30, 30-50 etc.	distance ratio that the driver exceeded the speed limit per speed limit category
total over 90 distance	total distance traveled at a higher speed than 90 Km/h
over 90 ratio	ratio of distance traveled at a higher speed than 90 Km/h
total segments	number of road segments crossed on trips
segments with limits	number of road segments with known speed limits
speed limits availability	ratio of road segments with known speed limits
active days score	percentage of days that the vehicle was active
recorded distance ratio	ratio of recorded distance
average-max-median trip distance	trip distance statistics
average-max-median pause time	statistics about the pause between trips
percentage start home	percentage of trips starting from the driver's home location
percentage start office	percentage of trips starting from the driver's office location
average trip duration sec	average trip duration in seconds
percentage charger nearby	percentage of trips with an electric vehicle charger nearby
percentage charger fast nearby	percentage of trips with a fast electric vehicle charger nearby
percentage charger 3F nearby	percentage of trips with 3F electric vehicle charger nearby

3.3 Data visualization

At this point some visualizations of the attributes will be presented to achieve a better understanding of the data, its format, and some relationships between them. Obviously, with 73 different features countless charts could be produced with each having its value. However, it was decided to present some of the important diagrams since this analysis is mainly about machine learning.

The diagram 3.1 describes the monitoring start periods of the different profiles. Firstly, each profile has either three months or one month of vehicle tracking data. The visualization shows the number of profiles for each start month when the profile was being tracked. In fact, there are separate colors for profiles with a duration of one month and for profiles with a duration of three months. As can be seen, there is data for all different months and seasons.

Figure 3.1: Monitoring period start frequencies



The visualization 3.2 is about the boxplots of the various attribute scores. Specifically, the characteristics described are the data quality score, reliability score, car use score, and driver behavior score. Additionally, boxplots show the median, quartiles, minimum, and maximum based on the distribution, and outliers. All scores appear to take values from 1 to 10 and each score has separate statistical measures.

The diagram 3.3 describes the evolution of the number of vehicle trips over time. In detail, for each month the total number of unique trips driven by the fleet vehicles is shown. The month of June 2021 had the most trips, while November 2020 had the fewest.

The chart 3.4 displays the average number of trips per day of the week. It is clear that on average the weekend days are the least traveled days by fleet vehicles. On the other hand, Friday is the day when most vehicle trips are recorded.

Figure 3.2: Scores boxplots

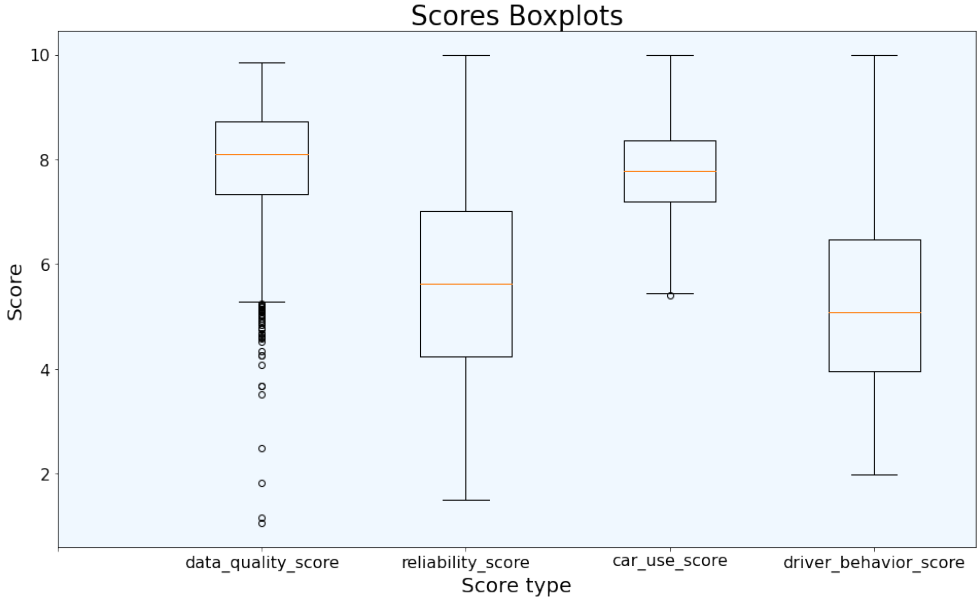


Figure 3.3: Number of trips over time

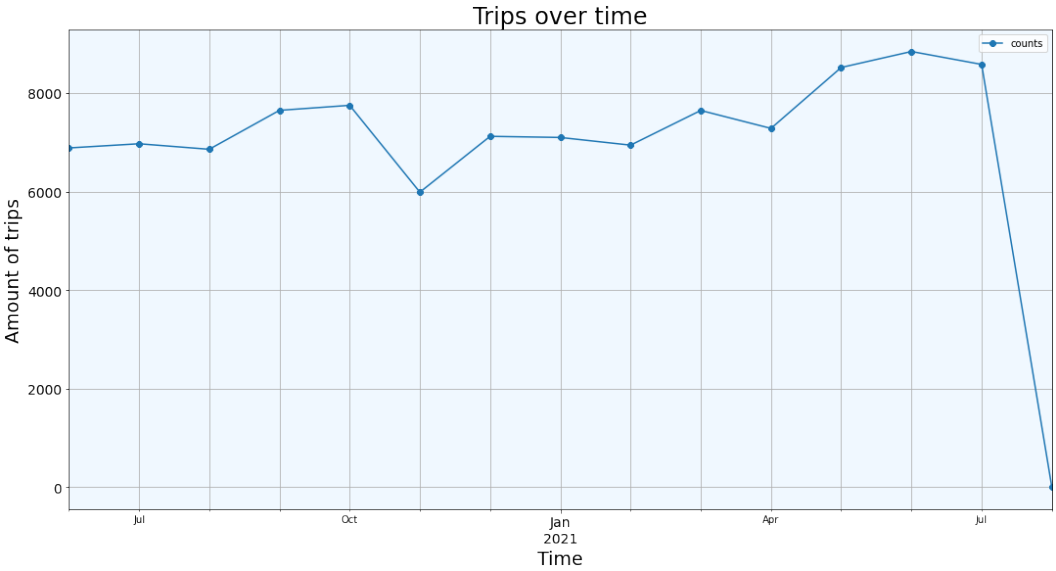
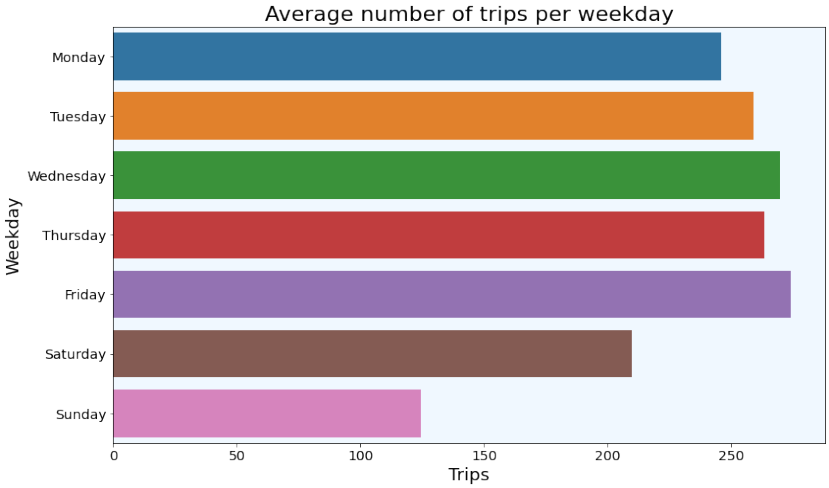
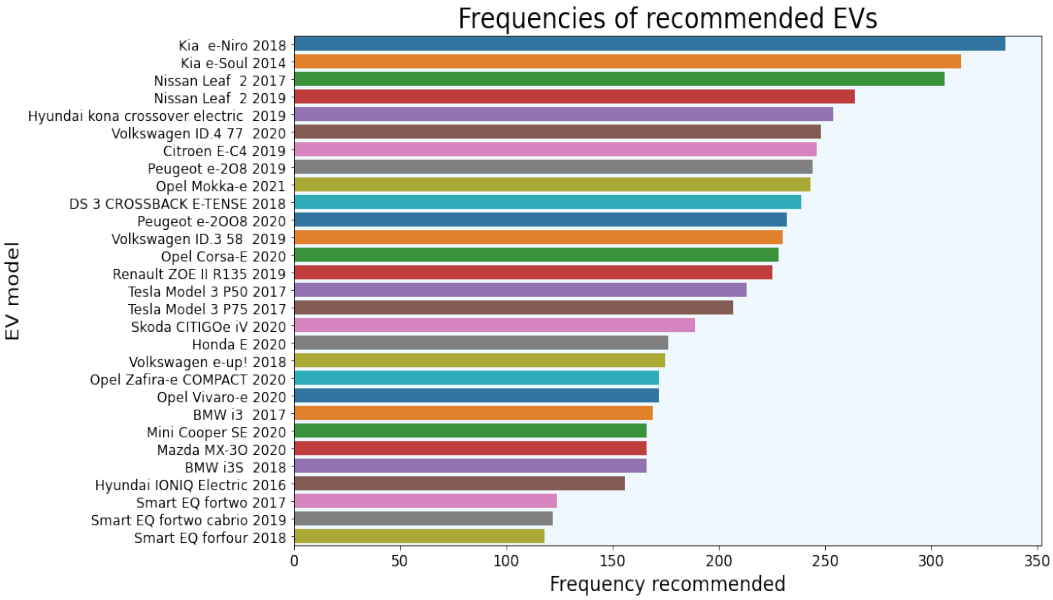


Figure 3.4: Average number of trips per weekday



The visualization 3.5 concerns the frequencies that different models of electric vehicles are recommended to replace a conventional vehicle. Specifically, the Kia e-Niro 2018, Kia e-Soul 2014, and Nissan Leaf 2 2017 models were recommended the most often, while the various Smart EQ models were recommended the least. However, this chart, like the analysis below, only shows reasonably priced electric vehicles. In other words, luxury and expensive electric models are not included.

Figure 3.5: Frequencies of recommended Electric Vehicles



3.4 Classification tasks

This analysis consists of 3 different parts. In detail, it tackles 3 separate classification problems using machine learning models and compares their performance on each separate problem with the appropriate metrics. Each problem is also a different type of classification as described in the literature review chapter.

A crucial attribute of the classification of the various vehicle profiles is the recommended electric vehicles. Of course, not every recommended electric vehicle is equally suitable for a specific conventional vehicle profile. This is why every recommended electric vehicle of a profile has a specific score called real score. The real score metric is the range-distance that can be achieved through existing charging stations. In detail, it is the weighted ratio of the total mileage recovered by charging the car with the currently available chargers compared to that of an ideal recovery by following the suggested charging strategy. Simply put, the metric takes values from 1-10, where 1 is the worst and 10 is the best. Thus, a score of 10 means that the current charging infrastructure achieves the optimal charging range.

Based on this metric, an electric vehicle model is considered suitable or unsuitable to replace the examined conventional vehicle profile. More specifically, after analyzing the distributions of all the real score metrics and based on the experience of the data science team manager, the real score of 5.2 was set as the threshold that determines whether the specific electric vehicle can replace the conventional one. In other words, an electric vehicle rated with a real score of 7 for a conventional vehicle profile is considered suitable to replace that particular conventional vehicle profile. On the other hand, an electric vehicle rated with a real score of 5 for a vehicle profile is considered unsuitable to replace that profile. Finally, the specific hypothesis concerns binary and multi-label classification problems, as will be analyzed below.

Another key point is that only electric vehicles with a price below 250,000 Polish zlotys, which corresponds to around 55,000 euros, are considered. Of course, more expensive electric vehicles can achieve better range and performance, but at the same time require more capital to acquire. Effectively, only cheap and affordable electric vehicle models are considered throughout the analysis. Finally, the specific electric vehicle models are also shown in the diagram 3.5.

3.4.1 Binary classification

The first machine learning application concerns the prediction of a conventional vehicle profile as suitable or unsuitable to be replaced by an electric vehicle. Of course, for each conventional vehicle profile, there are several recommended electric vehicles, each of which has a separate real score for the considered profile. More specifically, this relationship can be considered many-to-many, since many conventional vehicle profiles map to many electric vehicle models. For this reason, for each vehicle profile only the electric vehicle model with the highest real score is kept.

Table 3.2: Binary classification class frequencies

Class	Number of vehicle profiles
Unsuitable for replacement by an electric vehicle	580
Suitable for replacement by an electric vehicle	354

The table 3.2 describes how the 934 conventional vehicle profiles are categorized. Specifically, a vehicle profile classified as unsuitable means that all electric vehicle models had scores of 5.2 or less in the real score metric. On the other hand, a vehicle profile classified as suitable means that at least one electric vehicle model was rated more than 5.2 in the real score metric. Finally, the accuracy metric was used to evaluate the models of this specific machine-learning application.

Steps

For each algorithm, the following steps were followed to calculate the accuracy metric:

- Finding the best parameters for each algorithm using grid-search and 10-fold cross-validation. In other words, each declared parameter combination was trained for 10 random but repeated splits of training and validation data. The combination of parameters with the highest average accuracy metric for the training-validation data splits, in other words, the combination of parameters that made the most correct predictions, was saved.
- The best parameters from the previous step were then used to train the classifier and calculate the accuracy metric using leave-one-out cross-validation. This means that for each split of the data into training and validation, where the validation data only contains one sample at a time, the classifier was trained, the validation sample was predicted, and finally, the average accuracy metric was calculated.

Then, the classifier with the highest average accuracy value was used with the smote technique to further increase the average accuracy. Different resampling strategies along with grid-search and 20-fold cross-validation were used to find the optimal smote strategy and classifier parameters. Then, the optimal resampling strategy and parameters were used to train the classifier and calculate the average accuracy using leave-one-out cross-validation.

Finally, Bayesian optimization using Gaussian processes was used to find the optimal value of the *n_estimators* parameter. This parameter is a natural number which would be very difficult to optimize with grid-search since grid-search constitutes an exhaustive parameter search. After the optimal value was found, it was used together with the smote technique and the classifier to recalculate the metric accuracy using leave-one-out cross-validation.

3.4.2 Multi-class classification

The next machine learning application deals with a multi-class classification problem. Specifically, different classes were created that describe how well a conventional vehicle profile can be replaced by an electric vehicle. As in the binary classification application, for each vehicle profile, only the proposed electric vehicle model with the highest real score is considered.

Table 3.3: Classes of multi-class classification

Class	Description	Real score interval	Number of profiles
0	No electric vehicle recommendation	-	90
1	Disappointing real score	$real\_score < 4$	375
2	Low real score	$4 \leq real\_score < 6.5$	207
3	Good real score	$6.5 \leq real\_score < 10$	198
4	Perfect real score	$real\_score = 10$	64

The table 3.3 explains how the 934 profiles of the final dataset are classified. In addition, the different classes and real score intervals for each are described. It seems that the dataset shows some imbalance regarding the frequencies of vehicle profiles belonging to the different classes. Finally, the accuracy metric was used to evaluate the performance of the machine learning models.

Steps

Initially, the following steps were implemented for each classifier to calculate the average accuracy metric:

- Finding the best parameters for each algorithm using grid-search and 10-fold cross-validation. Then, the parameters with the highest average accuracy metric for the 10 training-validation data splits were saved.
- The best parameters from the previous step were used to train the classifier and calculate the accuracy metric using leave-one-out cross-validation.

Additionally, for the SVC, random forest, Naive Baynes, and Logistic regression classifiers:

- Creating a pipeline of a standard scaler, PCA, and the classifier. The best parameters were found for the PCA technique ($n\_components$) and the classifier using grid-search and 10-fold cross-validation.
- The best parameters found were used to train each pipeline to calculate the average accuracy with leave-one-out cross-validation.

For the classifiers with the highest average accuracy, the following steps were applied together with the techniques of random-oversampler and smote:

- Different resampling strategies were defined for the use of smote and random-oversampler.
- Grid-search was used with 250-fold cross-validation to find the pipeline consisting of the optimal oversampling technique, the optimal resampling strategy, and the optimal classifier parameters.
- The optimal oversampling technique, the optimal resampling strategy, and the optimal classifier parameters were determined to train a pipeline to calculate the average accuracy using leave-one-out cross-validation.

For the oversampling technique and the classifier that produced the highest average accuracy metric, Bayesian optimization with Gaussian processes was used to find the optimal value of the *n_estimators* parameter that takes discrete values. After the optimal value was found, the average accuracy value was again calculated using leave-one-out cross-validation.

3.4.3 Multi-label classification

The third machine learning application solves a multi-label classification problem. This particular application focuses on predicting all brands of electric vehicles that are suitable to replace one particular conventional vehicle profile. As in the binary classification application, for an electric vehicle to be considered suitable to replace a conventional one, it must be rated more than 5.2 in the real score metric. Suppose a profile can be replaced by the electric vehicle models Kia e-Soul 2014, Kia e-Niro 2018, and BMW i3s 2018, then the classifier should predict Kia and BMW as the suitable electric-vehicle brands for that profile and all the other brands as unsuitable for the profile.

In total, there are 16 different brands of electric vehicles in the dataset, each of which is also a separate label to be predicted as true or false. Specifically, these brands are Nissan, Peugeot, Hyundai, Renault, Skoda, Honda, Mazda, Volkswagen, DS, BMW, Smart, Kia, Mini Cooper, Citroen, Tesla, and Opel. Additionally, since the problem is multi-label, the performance of the models is evaluated by the hamming loss metric.

Steps

For each approach and algorithm, the following steps were implemented to calculate the average hamming loss metric:

- Finding the best parameters for each approach-algorithm using grid search and 10-fold cross-validation. The combination of parameters with the lowest average hamming loss for the training-validation data splits is saved. In other words, for each approach-algorithm, the parameters that achieve the accurate prediction of the most individual labels, are kept.
- The best parameters of the previous step were used to train the approach-algorithm and calculate the average hamming loss metric using leave-one-out cross-validation.

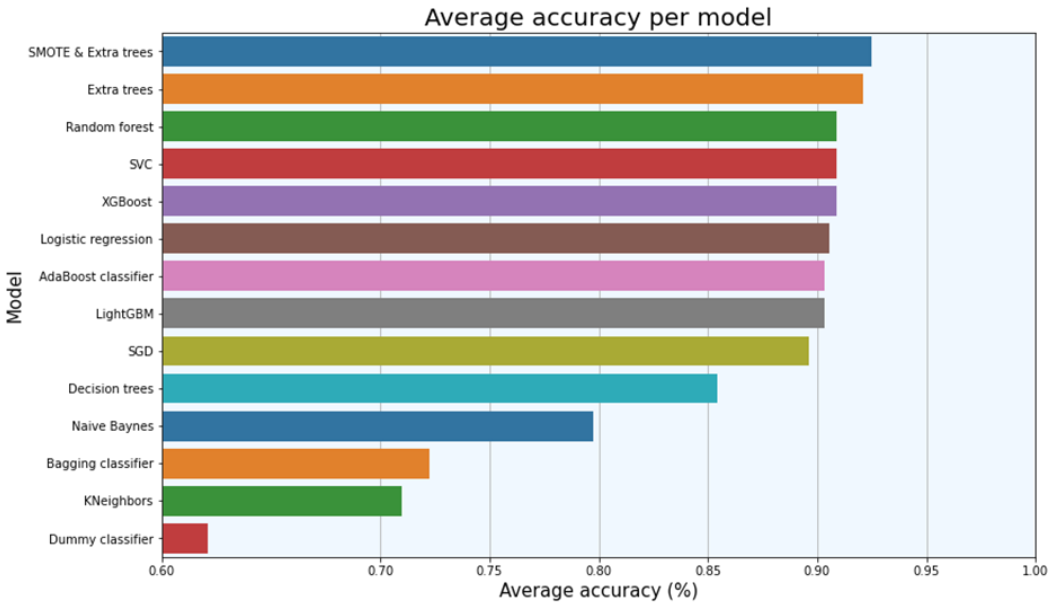
Chapter 4

Results and analysis

At this point, the performance of the machine learning algorithms in each of the classification problems is examined. For each problem, the different algorithms and pipelines are compared and finally, the one that achieves the best results based on the problem metric is presented.

4.1 Binary classification

Figure 4.1: Binary classification algorithm performance



In the binary classification problem, the dummy classifier had the lowest accuracy. In detail, the dummy classifier always predicts the most frequent class and is only used for comparison with

the other smarter models. The KNeighbors, bagging, and Naive Baynes classifiers do have higher accuracy than the dummy classifier, but compared to the rest of the models, do not bring good results. Also, decision trees exceed 85% accuracy and SGD reaches 89.6% accuracy. However, the best classifiers are considered extra trees, random forest, SVC, XGBoost, logistic regression, AdaBoost classifier, and LightGBM since they achieve over 90% average accuracy.

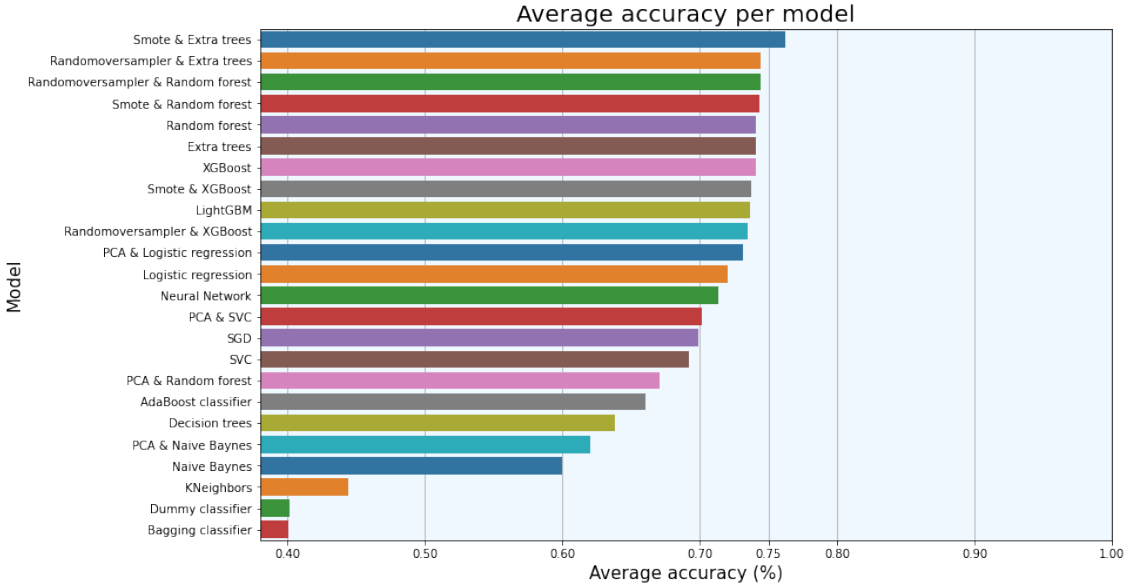
The extra trees are the algorithm that achieves the highest accuracy with 92%. Using the smote technique led to a 0.32 increase in the metric accuracy. Additionally, using Bayesian optimization with Gaussian Processes to find the optimal value of the parameter *n_estimators* resulted in a further increase of 0.107 in accuracy. Finally, the final model reaches 92.5% accuracy and is shown in the figure 4.2

Figure 4.2: Best pipeline in binary classification

```
Pipeline(steps=[('smote',
                 SMOTE(random_state=42, sampling_strategy={0: 580, 1: 400})),
                ('extratreesclassifier',
                 ExtraTreesClassifier(class_weight='balanced', max_depth=10,
                                      max_features=None, n_estimators=65,
                                      random_state=7, warm_start=True))])
```

4.2 Multi-class classification

Figure 4.3: Multi-class classification algorithm performance



A plethora of different algorithms and pipelines were used in the multi-class classification problem. The lowest performances are noted by the dummy, bagging classifier, and KNeighbors. On the other hand, the algorithms and pipelines containing extra trees, random forest, XGBoost, and LightGBM record the highest accuracy with over 73%. An important observation is that the

use of the PCA technique managed to successfully increase the average accuracy of the Naive Baynes, SVC, and logistic regression models but not of the random forest model. Moreover, the use of oversampling techniques in the XGBoost and LightGBM models not only did not increase the metric accuracy but decreased it as well. However, the use of oversampling techniques in the random forest and extra trees models resulted in a small increase in accuracy.

Again the extra trees classifier managed to achieve the highest accuracy. In detail, the use of extra trees scored 74% accuracy. When combined with the use of the smote technique, the accuracy increased by 2 whole units, reaching 76%. Also, using Bayesian optimization with Gaussian Processes to find the optimal value of *n_estimators* resulted in an accuracy increase of 0.21. Finally, the smote-extra trees pipeline reaches 76.23% accuracy and is described in figure 4.4

Figure 4.4: Best pipeline in multi-class classification

```
Pipeline(steps=[('smote',
                 SMOTE(random_state=42,
                       sampling_strategy={0: 95, 1: 375, 2: 207, 3: 198,
                                          4: 67})),
                ('extratreesclassifier',
                 ExtraTreesClassifier(criterion='entropy', max_features=None,
                                      n_estimators=52, random_state=7,
                                      warm_start=True))])
```

4.3 Multi-label classification

The multi-label classification models are evaluated on the hamming loss metric. Thus, the lower the value of the metric, the better the performance of the model. This is due to the metric describing losses. Based on the graph 4.5 the worst performances are shown for the adapted algorithms and the dummy classifier. Then, decision trees and logistic regression in various problem transformation approaches performed much better with a hamming loss of about 0.07. Moving forward, the neural network achieved a hamming loss of 0.062, outperforming the aforementioned models. Furthermore, random forest, extra trees, XGBoost, and LightGBM used in different problem transformation approaches produced the best results.

The extra trees algorithm reached the lowest recorded hamming loss value of 0.0467. This particular algorithm belongs to the ensemble approach. The exact extra trees model is described in figure 4.6

Now it will be shown what exactly a hamming loss value of 0.0467 means in this particular application. If the appropriate electric vehicle brands were to be predicted for 4 different conventional vehicle profiles (a total of $4 \cdot 16 = 64$ brands), the extra trees classifier would accurately predict on average all but 3 electric-vehicle brand labels since $3/64 = 0.046875$ which is almost as much as the hamming loss. In other words, only 3 brand-labels will be miscategorized (either a suitable brand as unsuitable or an unsuitable brand as suitable for a specific conventional vehicle profile) out of the total number of 64 brand-labels.

Figure 4.5: Multi-label classification algorithm performance

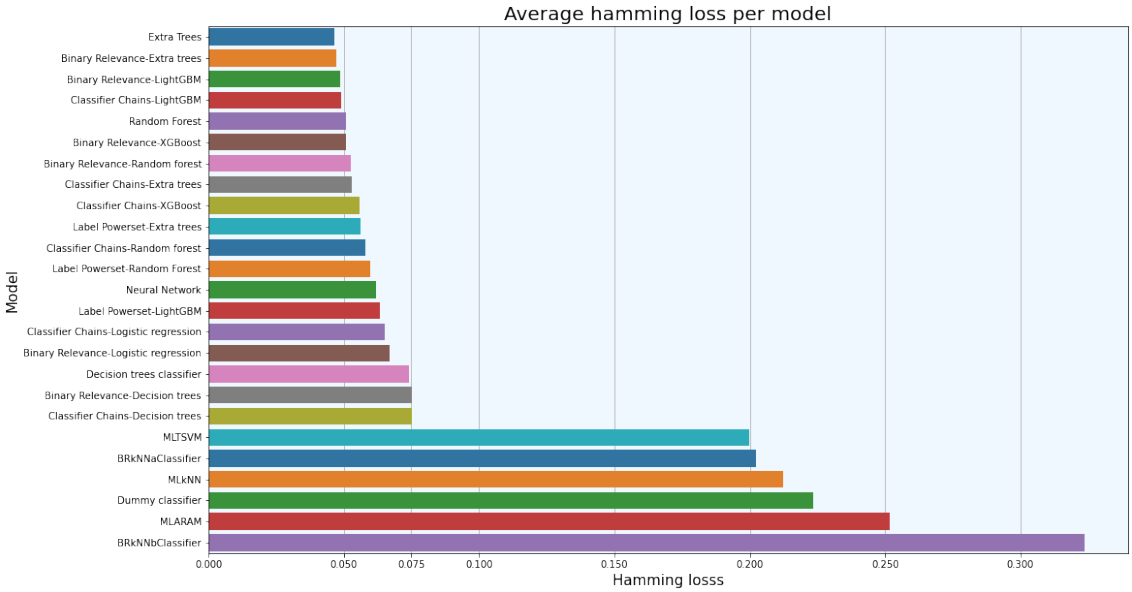


Figure 4.6: Best algorithm in multi-label classification

```
ExtraTreesClassifier(criterion='entropy', max_features=None, n_estimators=125,
                     random_state=7, warm_start=True)
```

4.4 Probabilities application

In all the examined problems, extra trees were part of the best machine-learning model. A very good use of the extra trees model is to calculate the probabilities that a vehicle profile belongs to each of the classes of the classification problem. To understand this, an example from the multi-class classification problem already described will be showcased.

Table 4.1: Application of probabilities in testing data

Class	Description	Profile 1	Profile 2	Profile 3
0	No EV recommendation	19	19	38.5
1	Disappointing real score	0	**71**	25
2	Low real score	6	10	**36.5**
3	Good real score	**75**	0	0
4	Perfect real score	0	0	0

The table 4.1 describes the probabilities of various conventional vehicle profiles belonging to the different classes based on the best extra-trees model for the analyzed multi-class classification problem. In each column, the probabilities are presented and the true prediction is shown in bold. In detail, the first profile of the testing data was predicted as a good real score in terms of its suitability to be replaced by an electric vehicle. In fact, this prediction corresponds to a

probability of 75% and is true since the profile actually had a good real score. Additionally, the second profile was predicted to belong to the disappointing real score class with a probability of 71%. Indeed, this prediction is correct. Moreover, the third profile was predicted as no electric vehicle recommendation with a probability of 38.5% which is the highest among the other classes. However, the specific profile corresponds to the low real score class with a probability of 36.5%.

The use of probabilities is quite important in classification problems. Firstly, in the case of multi-class classification, the pipeline extra-trees correctly predicts 76.23% of the cases which is a satisfactory result. Of course, the use of probabilities can give a more complete picture of each prediction, excluding some classes and giving more weight to others. This is shown in the example of the third testing vehicle profile. The prediction might not have been accurate, but the possibility that the profile belonged to the low real score class had become known. Effectively, if a second prediction was allowed, it would be correct.

Chapter 5

Conclusions

To recap, this paper analyzed three different types of classification problems in the context of a real company that uses technology and machine learning to its advantage. After all, a multitude of machine learning techniques and algorithms and their combinations have been presented and tested to detect those that can solve each problem best. The final machine learning models make predictions on the replacement of conventional vehicles with electric ones, their suitability for replacement by electric vehicles, and the brands of electric vehicles suitable for each conventional vehicle profile.

The described models can facilitate the decision-making of vehicle fleet users regarding the replacement of conventional vehicles. In particular, the models quickly give an overview of the vehicle by instantly predicting important knowledge about it. Hence, valuable time can be saved, since the user does not need to wait for the entire electric-vehicle-recommendation algorithm to be executed. This algorithm is very expensive since it has to calculate the optimal charging strategy. Finally, at a future stage, the models can be developed and combined with more data to make personalized recommendations of electric vehicle models to each user along with the cost that will be saved.

Through this case analysis, the power of machine learning is reflected. In particular, a small-scale problem of a niche industry was addressed with considerable success. Certainly, machine learning can be used, if it is not already being used, to tackle similar and different classification problems in other industries. There is already a huge variety of machine-learning techniques and models that can be easily implemented for various purposes and problems. In conclusion, the possibilities of applying machine learning in today's era are countless and it seems that in the future the use of machine-learning will be a must for all fields.

Bibliography

[1] A. E. Mohamed, "Comparative study of four supervised machine learning techniques for classification," *International Journal of Applied*, vol. 7, no. 2, 2017.

[2] A. A. Soofi and A. Awan, "Classification techniques in machine learning: applications and issues," *Journal of Basic and Applied Sciences*, vol. 13, pp. 459–465, 2017.

[3] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[4] O. Maimon and L. Rokach, "Data mining and knowledge discovery handbook," 2005.

[5] R. Sathya, A. Abraham *et al.*, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013.

[6] G. Sharma, G. Rani, V. S. Dhaka *et al.*, "A review on machine learning techniques for prediction of cardiovascular diseases," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE, 2020, pp. 237–242.

[7] R. Katarya *et al.*, "A review: Predicting the performance of students using machine learning classification techniques," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2019, pp. 36–41.

[8] P. Meshram, S. Ray *et al.*, "Field-level crop classification using an optimal dataset of multi-temporal sentinel-1 and polarimetric radarsat-2 sar data with machine learning algorithms," *Journal of the Indian Society of Remote Sensing*, vol. 49, no. 12, pp. 2945–2958, 2021.

[9] F. Qian, A. Pathak, Y. C. Hu, Z. M. Mao, and Y. Xie, "A case for unsupervised-learning-based spam filtering," *ACM SIGMETRICS performance evaluation review*, vol. 38, no. 1, pp. 367–368, 2010.

[10] R. Mansoor, N. D. Jayasinghe, and M. M. A. Muslam, "A comprehensive review on email spam classification using machine learning algorithms," in *2021 International Conference on Information Networking (ICOIN)*. IEEE, 2021, pp. 327–332.

[11] D. Johnson, "Unsupervised machine learning: What is, algorithms, example," Dec 2021, accessed: 2021-12-22. [Online]. Available: https://www.guru99.com/unsupervised-machine-learning.html#6

[12] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Systems with Applications: X*, vol. 1, p. 100001, 2019.

[13] J. Brownlee, "4 types of classification tasks in machine learning," Aug 2020, accessed: 2021-12-24. [Online]. Available: https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[14] R. Kumari and S. K. Srivastava, "Machine learning: A review on binary classification," *International Journal of Computer Applications*, vol. 160, no. 7, 2017.

[15] P. Jeatrakul and K. Wong, "Comparing the performance of different neural networks for binary classification problems," in *2009 Eighth International Symposium on Natural Language Processing*. IEEE, 2009, pp. 111–115.

[16] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?" in *2012 11th International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2012, pp. 102–106.

[17] A. Chatterjee, M. Vallières, and J. Seuntjens, "Overlooked pitfalls in multi-class machine learning classification in radiation oncology and how to avoid them," *Physica Medica*, vol. 70, pp. 96–100, 2020.

[18] V. Panca and Z. Rustam, "Application of machine learning on brain cancer multiclass classification," in *AIP Conference Proceedings*, vol. 1862, no. 1. AIP Publishing LLC, 2017, p. 030133.

[19] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[20] G. Tsoumakas and I. Katakis, "Multi-label classification," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, p. 1–13, 2007.

[21] K. Nooney, "Deep dive into multi-label classification..! (with detailed case study)," Jun 2018, accessed: 2021-12-24. [Online]. Available: https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff

[22] S. Jain, "Multi label classification: Solving multi label classification problems," Dec 2020, accessed: 2022-02-22. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/

[23] J. Brownlee, "Multi-label classification with deep learning," Aug 2020, accessed: 2022-02-22. [Online]. Available: https://machinelearningmastery.com/multi-label-classification-with-deep-learning/

[24] A. Ishaq, M. Umer, M. F. Mushtaq, C. Medaglia, H. U. R. Siddiqui, A. Mehmood, and G. S. Choi, "Extensive hotel reviews classification using long short term memory," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 9375–9385, 2021.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[26] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[27] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168 261–168 295, 2019.

[28] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "Decision trees for mining data streams based on the gaussian approximation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 108–119, 2013.

[29] D. D. Patil, V. Wadhai, and J. Gokhale, "Evaluation of decision tree pruning algorithms for complexity and classification accuracy," *International Journal of Computer Applications*, vol. 11, no. 2, pp. 23–30, 2010.

[30] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[31] R. Timofeev, "Classification and regression trees (cart) theory and applications," *Humboldt University, Berlin*, pp. 1–40, 2004.

[32] R. Garg, "7 types of classification algorithms," January 2018, accessed: 2021-12-25. [Online]. Available: https://analyticsindiamag.com/7-types-classification-algorithms/

[33] A. Sihombing and A. C. M. Fong, "Fake review detection on yelp dataset using classification techniques in machine learning," in *2019 International Conference on contemporary Computing and Informatics (IC3I)*. IEEE, 2019, pp. 64–68.

[34] K. M. Al-Aidaroos, A. A. Bakar, and Z. Othman, "Naive bayes variants in classification learning," in *2010 International Conference on Information Retrieval & Knowledge Management (CAMP)*. IEEE, 2010, pp. 276–281.

[35] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, "Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions," *Computer Science Review*, vol. 38, p. 100311, 2020.

[36] T. Ayer, J. Chhatwal, O. Alagoz, C. E. Kahn Jr, R. W. Woods, and E. S. Burnside, "Comparison of logistic regression and artificial neural network models in breast cancer risk estimation," *Radiographics*, vol. 30, no. 1, pp. 13–22, 2010.

[37] A. Ng, "Cs229 lecture notes," *CS229 Lecture notes*, vol. 1, no. 1, pp. 1–3, 2000.

[38] L. Abhishek, "Optical character recognition using ensemble of svm, mlp and extra trees classifier," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–4.

[39] V. Saxena and A. Aggarwal, "Comparative study of select non parametric and ensemble machine learning classification techniques," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2020, pp. 110–115.

[40] C. Désir, C. Petitjean, L. Heutte, M. Salaun, and L. Thiberville, "Classification of endomicroscopic images of the lung based on random subwindows and extra-trees," *IEEE transactions on biomedical engineering*, vol. 59, no. 9, pp. 2677–2683, 2012.

[41] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[42] Y. Li and Z. Chen, "Performance evaluation of machine learning methods for breast cancer prediction," *Appl Comput Math*, vol. 7, no. 4, pp. 212–216, 2018.

[43] S. K. Agrawal, "Evaluation metrics for classification model: Classification model metrics," Jul 2021, accessed: 2021-12-29. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

[44] M. Pushpa and S. Karpagavalli, "Multi-label classification: problem transformation methods in tamil phoneme classification," *Procedia computer science*, vol. 115, pp. 572–579, 2017.

[45] S. Destercke, "Multilabel prediction with probability sets: the hamming loss case," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 2014, pp. 496–505.

[46] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: a big comparison for nas," *arXiv preprint arXiv:1912.06059*, 2019.

[47] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[48] I. Syarif, A. Prugel-Bennett, and G. Wills, "Svm parameter optimization using grid search and genetic algorithm to improve classification performance," *Telkomnika*, vol. 14, no. 4, p. 1502, 2016.

[49] "Bayesian optimization using gaussian processes," accessed: 2022-02-23. [Online]. Available: https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html

[50] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation." *Encyclopedia of database systems*, vol. 5, pp. 532–538, 2009.

[51] D. Berrar, "Cross-validation." 2019.

[52] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Applied Soft Computing*, vol. 83, p. 105662, 2019.

[53] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.

[54] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[55] M. Ringnér, "What is principal component analysis?" *Nature biotechnology*, vol. 26, no. 3, pp. 303–304, 2008.