ISCTE IUL
Instituto Universitário de Lisboa

Department of Sciences and Information Technologies

Predicting Space Occupancy for Street Paid Parking

Marco Hélio Ramos Silva

A Dissertation presented in partial fulfillment of the Requirements for the Degree of Master in Computer Science Engineering

Supervisor:
PhD. Professor Luis Nunes,
Assistant professor at ISCTE-IUL

October 2017

Abstract

This dissertation discusses how to develop a prediction method for on-street parking space availability, using only historical occupancy data collected from on-street multi-space parking meters.

It is analyzed how to transform the raw data into a dataset representing the occupancy and how can this information be used to detect when the parking spaces on a street are Vacant or Full. Attributes like weather conditions and holidays are added to the data, giving them more context and comprehension.

After the data preparation and analysis, a prediction model is developed using machine-learning techniques that can forecast the availability of the parking spaces on a street at a specific day and on a given moment.

For that, a classification method is implemented based on decision trees and neural networks, comparing both methods regarding results and development time. Particular attention is given to the algorithm parameters, to achieve the right balance between accuracy and computational time.

The developed model proved effective, correctly capturing the different behavior of each street through the different weeks, and returning results useful to drivers searching for parking and to the business owners while monitoring their parking investments and returns.

Keywords: Parking space forecasting, street parking, traffic, parking guidance, machine learning, decision trees, neural networks

Resumo

Esta dissertação apresenta como pode ser desenvolvido um método para previsão de disponibilidade de lugares de estacionamento em rua, utilizando dados históricos obtidos através de parquímetros de controlo a múltiplos lugares.

É analisado como os dados em bruto dos parquímetros podem ser transformados num conjunto de dados que represente qual a ocupação dos lugares, e posteriormente como esta informação pode ser utilizada para detetar se o estacionamento em uma rua está livre ou ocupado. São adicionados também mais alguns atributos, como por exemplo informação sobre as condições meteorológicas ou que dias são feriados, dando mais algum contexto e compreensão à informação já existente.

Após a preparação e análise dos dados, é desenvolvido um método de previsão utilizando técnicas de aprendizagem automática de modo a que seja possível saber qual a disponibilidade de estacionamento em uma rua, a um dia específico e a um determinado momento.

Para isso, foi implementado um método de classificação baseado em árvores de decisão e redes neuronais, comparando ambos os métodos do ponto de vista dos resultados e do tempo de desenvolvimento. Foi dada especial atenção aos parâmetros utilizados em cada algoritmo, de modo a que haja um balanço entre a precisão e tempo de computação.

O modelo desenvolvido mostrou ser eficaz, captando corretamente o comportamento de cada rua nas diferentes semanas, devolvendo resultados uteis aos condutores que procurem lugares de estacionamento e aos proprietários do negócio por lhes permitir monitorizar o desempenho dos seus investimentos em parques de estacionamento e qual o retorno.

Palavras-chave: Previsão de lugares de estacionamento, estacionamento em rua, tráfego, assistência ao estacionamento, aprendizagem automática, árvores de decisão, redes neuronais.

Acknowledgements

First of all, I would like to thank my family and close friends for their comprehension, giving me the support to stay focused and finish my dissertation.

My greatest acknowledgment is for Professor Luis Nunes for giving me the opportunity to develop this study and being my supervisor, having always time available to answer questions and discuss results.

A big thanks to all the colleagues that worked with me during the master degree, their work and effort helped me to achieve good results through the master and obtain valuable knowledge about every theme.

At last, I would like to thanks my friends and colleagues at work for giving me the flexibility and comprehension about the effort required to finish this dissertation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANN – Artificial Neural Networks

ARIMA – Auto-Regressive Integrated Moving Average

AVD - Absolute Value of the Difference

BPNN – Back Propagation Neural Network

CSV – Comma Separated Values

EMEL – Empresa Municipal de Mobilidade e Estacionamento de Lisboa

FFNN – Feed Forward Neural Network

LR – Linear Regression

GA – Genetic Algorithm

GPS – Global Position System

IQR – Interquartile Range

MAE – Medium Average Error

MLP – Multi-Layer Perceptron

MLR – Multiple Linear Regression

PGI – Parking Guidance Systems

PUK – Pearson VII Universal Kernel

QREN – Quadro de Referência Estratégico Nacional

REP – Reduced Error Pruning

RBF – Radial Basis Function

RT – Regression Tree

SLR – Simple Linear Regression

SMOTE – Synthetic Minority Over-sampling Technique

SVM – Support Vector Machine

SVR – Support Vector Regression

VBA – Visual Basic for Applications

WEKA – Waikato Environment for Knowledge Analysis

WNN – Wavelet Neural Network

1. Introduction

This chapter describes the study theme presented in this document and the research motivations. It explains the addressed questions and their relevance in society.

1.1 Overview

In modern cities and urban areas the traffic is steadily rising, leading to an increase in travel time, congestions and health risks (Pflügler, Köhn, Schreieck, Wiesche, & Krcmar, 2016). Under these circumstances, locate an empty parking place can be a problem and a time-consuming task that can take an up to 30% the cruising time (Shoup, 2006), and this issue can be even more severe under specific conditions, like rainy weather or special events.

These situations cause a decrease in the quality of life, because a citizen may have preferred to use an alternative transportation, especially if he is non-resident or with little knowledge about parking locations and possible availability.

One way to reduce the unneeded cruising time and its consequences are by providing for the drivers a system where they can predict in advance if there are any parking spaces available, in a given area at a given time.

Parking spaces are a limited and valuable resource in cities, which if not regulated will lead to an increase in traffic and even worst life conditions because every citizen will tend to drive their cars to the destiny location. Directly related to the increase in cruising time is the underpricing of on-street parking, generating a waste of time, resources and revenue. The correct adjustments in the prices according to the demand will create more vacant parking spaces and reduce cruising, without decreasing the revenue (Van Ommeren, Wentink, & Rietveld, 2012). The earnings from parking’s are important to the cities because they use them as an asset to fund their finances, so a system that allows a better management of the parking spaces may allow an increase in the income, better zone pricing and better investment plans (Tiedemann, Thomas, Krell, Metzen, & Kirchner, 2015).

In the last years, many parking lots implemented Parking Guidance Systems (PGI), but these systems can be expensive because they rely on external sensors that are not always easy to implement and maintain.

1.2 Motivation

The purpose of this study is to analyze parking space occupancy using historical data from on-street parking meters to develop a prediction method for parking space availability. The objective is to develop a method suited for on-street parking, where is not easy to install sensors to collect data and provide real-time information.

Sensors require a good maintenance to keep the system accurate and precise, assuring that they are not damaged due to their exposition to the public and the weather. In the case of malfunction, sensors can easily have timing failures, compromising the system's reliability. Taking into consideration the maintenance costs is important because they represent an extensive effort that is vital to the system. If a failure happens and the system generates incorrect results the drivers eventually will stop trusting them. By only using data from the already existing parking meters, it is ensured that no that no other investments are necessary and that the existing equipment is enough to collect the required data. These type of systems are more affordable to the municipalities and allow a quicker implementation.

Searching for parking locations may originate slow traffic and jams, this is increased by the traffic of drivers that are nonresidents, since residents may already have some knowledge, based on previous experience, of where free parking space may be located. Tourists or outsiders represent a significant percentage of the cruising drivers, and these do not have any knowledge of the parking locations, having to slow down to search for one in their eye of sight originating heavy traffic.

Predicting parking space usage allows the drivers to know in advance the feasibility of using their cars in a specific area according to the free parking space prediction. By doing this, the driver can better plan his schedule and route, or even decide to travel using alternative transportation.

As a consequence, the fuel consumption is reduced, benefiting the environment and creating cleaner cities, allowing them to reach their pollution targets easily (Richter, Di Martino, & Mattfeld, 2014).

1.3 Research Questions

The focus of this research is on data exploration and prediction models using machine learning methods. With the collected information, the objective is to use machine learning and their recent developments to create optimized models that generate predictions with an acceptable confidence level.

This research seeks the answer to the following questions:

- In the parking spaces of the studied areas, is it possible to establish any patterns between the different days of the week and the different hours of the day?
- The data collected from the on-street parking meters is enough to create predictions with an acceptable confidence level?
- Is it possible to obtain accurate short (a few hours) and medium-term (2 or 3 days in advance) predictions with a dataset containing only historical data from a short time-span (3 months)?
- How effective is the precision of a prediction model built using decision trees when compared to a model using neural networks?
- Is the training of the models efficient enough to be executed daily without requiring dedicated or external computational systems?

1.4 Objectives

This work presents the results of applying prediction models based only on historical data collected from the existent on-street parking meters. After the creation of the models, their accuracy is tested in predicting if a parking space is full or not at a given moment in a specific parking zone.

From the business point of view, knowing in advance the parking occupancy at a given time allows the city entities to anticipate specific actions, like creating temporary parking spaces or route the traffic to another parking in the surrounding areas. Having the knowledge of how the parking demand increases in a specific zone allows for a better strategy choice when deciding the creation of new parking zones, enabling the business to create opportunities where they have bigger earnings and better long time return.

To the business, this type of solution has the following usages:

1 - Inform the clients through the website of the predicted occupation of a parking area at a given moment in the future;

2 - Detect anomalies in the regular pattern of parking occupation;

3 - Reorganize inspection teams to the zones with the biggest parking occupancy prediction;

4 - Readjust parking fees according to the average occupancy of a given area;

5 - Decide the location of new parking spaces according to the average occupation prediction.

From the user's point of view, the resolution of this problem allows them to know in advance if it is feasible to use their vehicle to reach a destiny or if is better go to a different location with an alternative parking, or even in another way of transportation. This reduces their cruising time, fuel consumption and, as a consequence, the air quality increases.

The creation of a system that provides parking availability forecast allows the business to increase the user satisfaction, establishing a proximity relationship with the clients and enhancing the company image.

1.5 Contributions

This thesis consists in the prediction of parking spaces availability, but it applies to on-street parking, which is a topic less studied since is more difficult to collect data outside closed parkings and to verify its accuracy.

The studied developed used historical data acquired from on-street multi-space parking meters, while other studies on this theme used data collected from individual sensors in closed parking. Individual sensors may have the capacity to provide real-time data, allowing a precise control of the parking status and to feed data into algorithms used in short-term occupancy prediction, however the application of sensors is not always feasible. By using historical data, this study demonstrates how this data is useful in the identification of parking patterns and the precision of the predictions developed with it.

The usage on multi-space parking meters creates more difficulties in obtaining the occupancy for a specific street, since these meters are not street specific and they may have registered parkings of neighbor streets.

In the city where this study is based is possible to obtain parking permissions, which will result in parking places that are occupied but not registered in the parking meters. On-Street parking is also more prone to illegal parking that will also be occupied spaces not registered.

To deal with these issues and obtain the parking occupancy status, it is developed a method to analyze the data and then calculate if the parking on a street is “Vacant” or “Full”, using the number of registries combined with peak occupation values.

While neural networks are used in many parking prediction studies, decision trees have also been used with success, being faster to execute and easier to develop. This study compares these two methods, analyzing how they perform and the effort required to develop them.

1.6 Research Method

This study performs an explanatory research in the forecast of parking places availability, seeking to establish a relation between a set of relevant attributes in the dataset and the prediction result, without individual study of each specific variable.

In a first stage, an exploratory research on the dataset is executed, analyzing it and searching how to extract information from it.

In a second stage, it is done an explanatory research developing a prediction method based on the latest studies in Machine Learning.

Using the dataset, several models are developed to verify if a generic model of all zones is accurate, or if better results are achieved splitting the dataset according to other characteristics, such as zones or by behavioral patterns, like the number of registries in a day.

Figure 1 exhibits the several steps that occur in the used research method, beginning with problem definition, dataset preparation and then model development. This is an iterative process. The dataset is refined and the model is readjusted at each iteration to obtain more accurate results. The iteration process is important because the fine tuning of the dataset and model parameters has a relevant impact on the results, so this approach allows small optimizations to be made and compared with previous results (Zheng, Sutharshan, & Christopher, 2015).

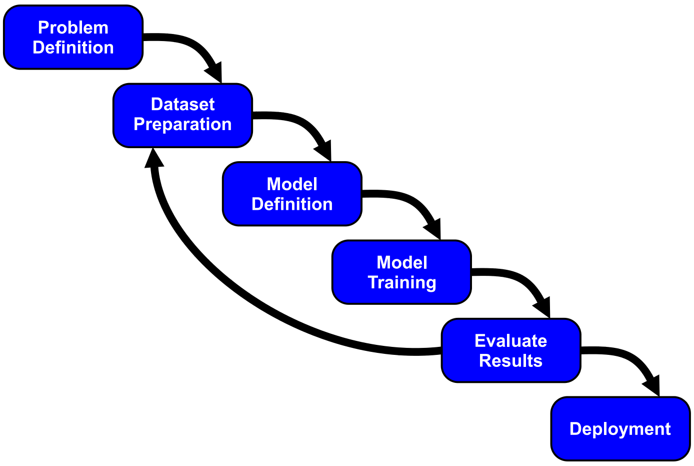


Figure 1 – Research method steps

The developed model is tested by applying it to a portion of the dataset that was not used in training, being the results of this test evaluated using quantitative analyses, based on statistical metrics and execution time. The expected performance will be a good balance between accurate prediction results and a fast execution time.

1.7 Dissertation Structure

This dissertation is divided into seven chapters, each one having the following content:

- Chapter 1. – Introduction: Present chapter, explaining the theme and objectives of this dissertation;
- Chapter 2. – State of Art: Describes all research made during the development of this study;
- Chapter 3. – Case Study Background: Introduces the dataset and context where this dissertation is developed, tools and methods used;
- Chapter 4. – Dataset Analysis: Exploration and comprehensive analyses of the dataset, observing the data behavior and the type of attributes included;
- Chapter 5. – Prediction Models Development: Elaboration of the models using different samples of the dataset and algorithms, evaluating their performance and how can they be optimized;
- Chapter 6. – Result Analysis: Present and compare the results according to the different experiments and contextualize them;
- Chapter 7. – Conclusions and Future Work: Dissertation conclusions and what type of works can be done in the future to improve results;

2. Literature Review

This section presents the current state of the art in prediction methodologies and data acquisition. Some of the authors focused on problems similar to the subject of this document, and this review shows the approaches they used and the results obtained.

The first part of this chapter describes the different techniques and devices used to acquire parking allocation data, presenting the studies conducted according to each technique.

The second part introduces the type of methods used to predict parking lot occupancy, the performance obtained with each one and the type of data used.

2.1 Parking Space Availability Methodologies

Predicting parking space occupancy has been a subject of study for some time but with an increase in recent years. The advancements in computing processing power, mobility, smart vehicle technologies, IoT, integrated sensors and communications lead to new approaches to the problem, seeking to obtain more efficient and accurate results.

The different methods used vary in the way that the data is collected and calculated, having this direct impact on the type of infrastructure needed, implementation time and costs.

2.1.1 **Based on Real-time Data and Infrastructures**

Acquire the parking occupancy data and calculate the demand in real-time is possible with the current technologies by using sensors or cameras, and networks (Klappenecker, Lee, & Welch, 2014). Selecting which technology to use depends on the project objectives, environmental factors, parking type and available infrastructure (Mainetti et al., 2016).

Technologies based on image use cameras as a sensor to monitor parking spaces and then process those images with software to detect parking space occupation. One method is to use them to track the incoming and outgoing cars and then calculate the lot occupancy, another method is to monitor the occupancy of each parking space, and then calculate how many remain vacant. Cameras can be difficult to tune to provide good accuracy, since they are sensitive to weather and changes in light conditions, resulting in distortions, oscillations, occlusions, and shadows that can be misinterpreted (Huang & Wang, 2010).

Wu, Huang, Wang, Chiu, & Chen (2007) developed a method that used images captured by a camera to detect empty or occupied parking, making it possible to use this system with existing surveillance cameras. Each image is subject to a preprocessing that divides them into patches containing three parking spaces that are then classified using an SVM (Support Vector Machine), obtaining an average accuracy of 85%. Another possibility is to capture an aerial image from multiple cameras and analyze the parking lot structure using an algorithm based on terrain elevation and intensity, recognizing by the elevation which parking spaces are vacant. The result of this allows extracting the parking lot activity to later visualize in the form of parking activity simulations (Wang & Hanson, 1998).

Another method uses only one aerial camera to detect the parking structure using block prediction, interpolation, and extrapolation. This method proved effective but sensitive to light conditions (Seo, Urmson, & Ratliff, 2009). Delibaltov, Wu, Loce, & Bernal (2013) used cameras mounted on public lamp-post to automatically detect parking space occupation based on the parking space geometry. The method, when tested at the University of California Santa Barbara, had an average accuracy of 76% under different weather conditions.

Technologies based on another type of sensors use one sensor by parking space to detect if it is vacant or not (Yamada & Mizuno, 2001). According to the purpose and the type of conditions in the parking lot different type of sensors are used, and they can be divided into two categories: intrusive and non-intrusive. Intrusive sensors are installed under the pavement and typically are piezoelectric sensors, weight-in-motion or magnetometers. Non-intrusive sensors are usually fixed on the ground or in the ceiling, being usually ultrasonic sensors.

In Italy, the city of Pisa monitors the parking in Piazza Carrara using sensors mounted on the ground that detect if a parking space is occupied or vacant. This information is then centrally processed and displayed in PGI and also in a mobile application to guide users searching for a free parking space (Tsiaras, Hobi, Hofstetter, Liniger, & Stiller, 2015).

Parknet is a platform that uses ultrasonic sensors and GPS mounted in cruising vehicles to monitor parking availability and then send this information through the network to a centralized server, being this solution tested in the city of San Francisco with an accuracy of 90% (Mathur et al., 2010).

2.1.2 Based on Offline Data

In some parking lots obtaining real-time data is either difficult or too expensive, especially if it is on-street parking. In this type of parking, the outdoor conditions make sensors installation difficult, sometimes requiring a bigger infrastructure when compared to off-street parking, which increases the final costs. In the outdoor, the exposition of the sensors to an uncontrolled environment creates difficulties in their accuracy and reliability (Tamrazian, Qian, & Rajagopal, 2015).

To control occupancy, most parks rely on parking terminals for off-street parking and parking meters for on-street parking. These systems can store the parking registries they make, and that data can be used to create a historical record of past parking occupancy behavior and predict occupancy rates. (Tamrazian et al., 2015)

The prediction methods based on offline data have proven useful even when real-time monitoring sensors are available. Caicedo (2009) developed a statistical model that combines the use of historical and real-time data from sensors to predict parking occupancy and then inform in advance if an off-street parking facility is full.

To increase the accuracy, this model was enhanced with simulations, being later tested and validated with data from a parking facility on Barcelona (Caicedo, Blazquez, & Miranda, 2012).

Real-time data can detect that there are parking spaces vacant at a given moment, but can’t assure that those spaces are still available when the vehicle arrives. Because of this, Rajabioun (2013) developed a system where he combined the real-time data with historical data to ensure that the parking place is still available at the arrival time. Pullola, Atrey, & Saddik (2007) developed a GPS navigation system to find available parking spaces that also used real-time data mixed with historical data to build an accurate prediction model.

The city of San Francisco hosted an experiment where GPS and phone accelerometers were used to detect when a car was parked or moving, registering those values to build a historical profile. That data was used to create a historical model that predicts the parking occupancy of a specific zone. (Xu et al., 2013)

Using only historical data Liu, Guan, Yan, & Yin (2010) developed a prediction model for parking space occupation using chaotic time series forecasting methods. The model was successfully tested in the city of Beijing, China.

According to Richter (2014), it is possible to build long-term prediction models using only historical data, executing them later on centralized computers or small devices like smartphones or in-car systems. The challenge in these cases is to obtain the maximum accuracy using the smallest set of data so that models can be quickly retrained with new data.

2.1.3 Comparison Between Methodologies

Collecting data using sensors can provide real-time information about the occupancy status of a parking lot, but the type of sensors used has an impact on the result. Magnetic field sensors are typically applied in large parks to obtain individual parking space occupancy status, but they require some construction in the parking lot. They work relatively well because they interact with the ferrous parts of the vehicles detecting if the parking space is occupied. By measuring the changes in the sensor magnetic field is also possible to detect if a car has passed or not and the type of vehicle that passed or parked (Burgstahler, Knapp, Zöller, Rückelt, & Steinmetz, 2014). The sensitivity of these sensors can be a problem, since residual magnetism or imprecise parameters during the manufacturing process can affect the sensor behavior, creating difficulties in calibrating them. Also, the same model or type of vehicle can have variations in their magnetic resistance, causing invalid reads from the sensors (Markevicius et al., 2016).

Ultrasonic is another type of sensors, having a lower cost and the possibility of being installed on the ceiling above each parking space, making them easier to install (Mimbela & Klein, 2000). They work by emitting an ultrasonic wave and then reading the time that takes the wave to return. By measuring the differences in the wave return time is possible to detect if a parking space is occupied or vacant. The disadvantage of these sensors is that they are sensitive to temperature changes and heavy air turbulence, generating wrong values that can give misleading results. (Kianpisheh, Mustaffa, Limtrairut, & Keikhosrokiani, 2012)

Sensors of another type are less common but can also be found in parking lots, like infrared, microwave radar or RFID (Geng & Cassandras, 2013).

Despite the type of the sensor, their usage is not always feasible to every parking because of their installation and maintenance costs that increase under some parking conditions like on-street parking. Another issue is that they have to be individually installed in each parking space, requiring labor time and arrangements in the existent infrastructure. The costs can be partially recovered by an aggressive ticketing of parking violations (Klappenecker et al., 2014). Installing sensors usually requires the installation of a wired network, since they generate a significant amount of data that sometimes is complicated to transmit wirelessly, especially in the case of cameras (Mainetti et al., 2016).

Cameras can be used to monitor large portions of a parking lot, but their sensitiveness requires correct placement, being image filtering also crucial to obtain images with enough quality (Burgstahler et al., 2014). As previously discussed, Wang & Hanson (1998) also stated these problems, and despite their system proved effective on parking lots with sparse vehicles it is difficult to install the camera at the required height and angle. The poor quality line parking marks are difficult to capture on camera, making the process of detecting available parking spaces a difficult task (Seo et al., 2009).

Real-time data gives a good representation of the status of a parking lot, but this information is valid only for a limited amount of time (Caliskan, Barthels, Scheuermann, & Mauve, 2007). Sometimes a delay in data refresh causes the system to work with outdated information, meaning that when the driver arrives the parking place is no longer free (Rajabioun & Ioannou, 2015). Several reasons contribute to the outdated information, like the time that cameras take to process the images or that the sensors take to communicate or refresh their status. These failures inevitably originate noise, corrupted data and missing observations, requiring the use of prediction algorithms that work well in these conditions (H. Chen, Grant-Muller, Mussone, & Montgomery, 2001).

Offline data does not give real-time feedback, but its usage has some advantages because it allows data filtering and analysis. It also does not require dedicated equipment because raw data is collectible from already existing sources like parking meters or even accumulated data from real-time sensors. The analysis of this type of data allows the observation of how the occupancy behaved in the past and what patterns emerge from it, creating a better understating of peak hours during the day and constantly occupied areas (Tamrazian et al., 2015).

Because offline data is usually analyzed as a trend in time, it is important to detect outliers in data since they can negatively affect the results. Like in real-time data, these outliers can appear because of failures in the equipment or the network, but they can also appear from external factors that are not possible to determine when analyzing the data. The advantage of offline data is that it is possible to detect these outliers and eliminate them, obtaining a cleaner data and more accurate results (Piovesan, Turi, Toigo, Martinez, & Rossi, 2016).

It is also possible to search for patterns to create clusters that contain specific behaviors under certain conditions, allowing a more clear comprehension of the data (X. Chen, 2014). Another possibility is to verify similar behaviors in a way that is possible to aggregate parts of the data, thus allowing a reduction in size and better computation times (Yan-jie, Tang, Wei-hong, Phil, & Wang, 2014).

With historical data it is possible to develop prediction models and simulations that allow verifying the data behavior in the future, making it possible to modify certain attributes and observe if the current behavior is maintained or changed. (Caicedo et al., 2012). It is also possible to add contextual attributes to the data, like weather conditions, special events or roadworks. This allows a more comprehensive learning of how changes in the environment affect the obtained data and provide a better understanding of which situations are addressable and which ones are difficult to control (Andrea, Klaus, & Walter, 2000).

The prediction models and simulations developed present the trend behavior of the occupancy at a given moment under certain conditions, and do not take into account sudden changes. When compared to real-time data, offline data cannot produce precise results when a deep or sudden change occurs in a way that affects the results (Tamrazian et al., 2015).

The type of attributes originally contained in the offline data can influence the type of study conducted, mainly when using passive elements that do not have detection capabilities, like on-street parking meters. This kind of devices usually relies on manual interaction with them to detect that a vehicle has arrived at the parking space, but they cannot automatically detect when that vehicle leaves. With parking meters is also not possible to detect if the car left exactly when the paid time ended, or if a car is parked and has not paid at all like in the case of cars with parking licenses. This type of situations should be addressed when analyzing the data by verifying how all the attributes interact with each other and what kinds of patterns the data follows (Caicedo et al., 2012).

Table 1 presents a list of studies performed using different data acquisition methodologies and the results obtained.

Paper and Author	Type of Data	Limitations	Results
Robust Parking Space Detection Considering Inter-Space Correlation (Wu et al., 2007)	Real-time data collected from multiple aerial cameras.	Vehicle shadows and occlusion may difficult the detection of a parked vehicle.	85% accuracy in identifying occupied spaces using SVM.
Parking lot occupancy determination from lamp-post camera images (Delibaltov et al., 2013)	Real-time data collected from cameras on a public lamp-post.	Different light conditions, fog, rain or image resolution, may affect the result.	76% average accuracy in identifying occupied spaces.
ParkITsmart: Minimization of cruising for parking (Tsiaras et al., 2015)	Real-time data collected from individual ground sensors.	Sensor cost and placement have a big deploy and maintenance cost.	80% reduction in the time searching for parking.
Intelligent parking assist (Rajabioun et al., 2013)	Real-time data combined with historical data of parking occupancy.	The performance decreases as the prediction time distance increases.	1.2% average error for 10 minutes ahead prediction 2.8% average error for 40 minutes ahead prediction.
Where Is My Parking Spot? (Tamrazian et al., 2015)	Offline historical data from parking meters, enriched with contextual attributes.	Precise arrival time is difficult to estimate because the parking meter resides at the entrance of the lot.	7.4% medium average error.
Unoccupied Parking Space Prediction of Chaotic Time Series (LIU et al., 2010)	Offline data, collected from individual sensors.	The system is effective, but the data collection is based on individual sensors.	2.33% average prediction error.

Table 1 – Studies comparison using different data acquisition methods

2.2 Prediction Modeling Techniques

Predictive models are an advanced way to forecast events in the future based on data from past events. From all the existent prediction techniques, Artificial Neural Networks (ANN) is often considered in the literature as a separated approach, mainly because of the differences in the way it is modeled and interpreted (Karlaftis & Vlahogianni, 2011).

ANN is a method of building predictive models that use algorithms to simulate the functions of a human brain, containing several layers each one with a number of neurons, being these interconnected through links with weights (M. Al-Maqaleh, A. Al-Mansoub, & N. Al-Badani, 2016). They have excellent ability to recognize and classify patterns (E. I. Vlahogianni, Golias, & Karlaftis, 2004), being able to learn and adapt from the original data (Blythe, Ji, Guo, Wang, & Tang, 2015).

Many studies used prediction models based on other techniques, like Support Vector Machine, Regression Trees or Markov Chain, that have been the prediction standard for several years because of their simplicity and easy comprehension (Sargent, 2001). They can be fast, efficient and easier to develop, but may lose accuracy when used with high complexity data and with low correlation (Karlaftis & Vlahogianni, 2011).

In the literature of parking availability prediction, many studies focused specifically on ANN, while others try to establish a comparison between ANN and other techniques.

Because of these differences between the techniques, there is a subchapter dedicated to the usage of ANN in parking prediction, where a literature review on these techniques is presented as well as the obtained results, being this latter discussed and compared to the other techniques.

2.2.1 Usage of Neural Networks in Parking Availability

ANN is shown in literature as a good alternative to the other techniques since they can approximate almost to any function despite its degree of linearity and without knowledge of its functional form (Kumar, Parida, & Katiyar, 2013). They try emulate the cognitive capability of the brain using a set of interconnected neurons, each one having its specific input/output and characteristics (Vasudevan & Parthasarathy, 2007).

They are efficient when working with noisy data or with low correlation and also very flexible and robust when dealing with multi-dimensional data. Their capacity to generalize and learn make them effective in the development of prediction models (Karlaftis &

Vlahogianni, 2011) and very suitable for predicting events where little is known about the relationship between the different data attributes. In this case and if enough training data is available is possible to use the neural network to derive the required information by training with the data. ANN also has the advantage of continuous learning, because their training dataset can gradually be enriched with more data so that the prediction performance continuously increases. The ANN hidden layers allow the prediction models to consider nonlinear relationships and interactions between the data attributes (Pflügler et al., 2016).

Yang, Liu, & Wang (2003) developed a prediction model based on ANN to forecast parking spaces using as input a set of variables with contextual data like road traffic flow, weather, specials events or road conditions. The system was implemented in the city of Beijing and proved effective in the reduction of recirculating traffic.

E. Vlahogianni, Kepaptsoglou, Tsetsos, & Karlaftis (2014) used an ANN to predict parking occupancy up to one hour ahead. Their system has two models; the first applies survival analysis to predict the probability of a parking space being free for the next time intervals, and the second introduces ANN for the prediction of the time series of parking occupancy in different regions of an urban network. They tested the model in the city of Santander, Spain, and proved that neural networks adequately captures the temporal evolution of parking space availability and can accurately predict occupancy up to one hour ahead.

Fengquan, Jianhua, Xiaobo, & Guogang (2015) developed a prediction model based on back propagation neural networks (BPNN) to calculate the number of remaining parking spaces. They used historical data collected from sensors, and the dataset is constantly updated in real-time with the data gathered from those same sensors. The system was tested with data from a closed parking space in the city of Nanjing in China and proved effective, despite the fact that some error always exists due to big and sudden variations in the number vacant parking spaces during the day.

Blythe et al. (2015) used the parking occupancy data from several off-street parking spaces in Newcastle, England to study and develop a model for short-term parking availability prediction. They analyzed and enriched the data with contextual information and then used it to build a Wavelet Neural Network (WNN) for prediction, creating different models for workdays and weekends since the parking behavior is different between these days. They concluded that the accuracy and training time of this method is appropriate for short-term forecast of available parking spaces and can provide valuable information to the drivers while searching for park.

2.2.2 Usage of Other Prediction Techniques in Parking Availability

Prediction techniques like Regression Trees or Support Vector Machine have widespread usage, and provided good results in the forecast of parking space availability (Zheng et al., 2015). The statistical theory used by them is well known, allowing the users to understand the influence of each attribute in the dataset, giving them the ability to verify the model established assumptions including issues of adequacy and fit (Sargent, 2001).

With these techniques it is possible to make prediction models for regression or classification, depending if the desired outcome is a number or category. The adequacy of each method and the developed model varies according to the dataset and its attributes. It is necessary to evaluate the different methods performance by comparing the error between the predicted values and the known values (Lijbers, 2016).

Zheng et al. (2015) developed prediction models using Regression Trees (RT) and Support Vector Regression (SVR). RT is a type of decision tree, it consists of multiple nodes with each branch representing a different outcome, having leafs that represent the prediction result. This method is fast to train and is easy to interpret the built model by simply analyzing the tree, verifying what are the most important attributes. SVR is a type of Support Vector Machine (SVM) modified to predict numeric values, having the advantage of being sensitive to changes in patterns of the dataset (Lijbers, 2016). Using two parking spaces datasets, one from the city of Melbourne, Australia, and another from San Francisco, USA, they compare the performance and accuracy of the models in predicting the occupancy rate of the parks. They concluded that the regression tree was the more accurate model and the least computationally intensive (Zheng et al., 2015).

Caliskan, Barthels, Scheuermann, & Mauve (2007) developed a short-term prediction algorithm using Markov Chain theory and tested it with real-time information from parking lots. The objective was to predict the available parking at the driver’s time of arrival based on the current park occupation, and by consequence reduce the amount of traffic generated by the search for free parking. The algorithm consists of modeling the parking lot as queue and using a Markov Chain with distributed inter-arrival and parking times to describe it. They tested the model with a simulation using historical data from the city of Brunswick, Germany, and results proved that the model was effective in reducing the effort to search for free parking. Yanjie, Wey, & Wey (2007) developed a prediction model where a weighted Markov Chain was combined with wavelet analysis of the parking occupation time series. The objective was to use the wavelet analysis to detect low signals that will represent trends and high signals that will represent unusual events in the parking occupation. The time series wavelet is then reconstructed according to this analysis, and the weighted Markov Chain is used to create the forecast. This method was tested in Nanjing, China and proved to be accurate when using a system that can provide real-time occupancy data.

Clustering analysis is a statistical estimation method that recognizes patterns in data based on the distance and correlation between the elements. An, Han, & Wang (2004) used this method to develop a system to dynamically guide the drivers to the zone with most probability of having vacant parking lots, where they proved that their system creates more balance between parking lot occupancy in situations of low or high demand. Richter et al. (2014) also used clustering to detect similar patterns in data so that will be possible to reduce the amount of historical data to be stored while assuring that the data can still provide enough information to build accurate prediction models. They concluded that the clustered models are accurate but with an inferior performance when compared to a full dataset model, but the storage needs are reduced by 99.03% when working with clusters.

Cherian, Luo, Guo, Ho, & Wisbrun (2016) conducted a study where they used regression to build a prediction model for parking garages occupancy in urban areas. They compared several statistical regressions methods namely, Simple Linear Regression (SLR) with only one attribute, Multiple Linear Regression (MLR) with seven attributes, and SVR with different kernel functions. In their experiments, they concluded that the best performance was achieved with SVR when using Pearson VII Universal Kernel (PUK) and that the performance accuracy increased when more contextual attributes were added to the data.

Fengquan et al. (2015) used a linear time series Autoregressive Integrated Moving Average (ARIMA) model to predict the number of unoccupied parking spaces. The forecast used real-time data acquisition of remaining vacant spaces, constant update of the datasets time series and constant correction of model parameters by applying real-time data. The model was tested in a central mall parking lot in Nanjing, China, and proved accurate for real-time usage.

Tamrazian et al. (2015) used k–nearest neighbors to develop a model of parking prediction that was constantly updated with real-time data. This algorithm has the advantage of quickly identifying the different trends that the occupancy is following based on the real-time data it is observing, allowing for more accurate predictions. It also has the advantage of being efficient since no training is required. The model was tested in two parking lots in the Stanford University, California, and provided accurate predictions with a decrease in mean and maximum error rates as the time progressed.

2.2.3 Comparison Between Prediction Techniques

One of the problems with ANN is their “black box” concept. In the other techniques, it is usually possible to find the effect and influence of each variable, while in neural networks is difficult or not possible at all to find these types of relations (Kumar et al., 2013). When using a technique like regression is possible to sequentially explore and eliminate variables that do not contribute to the model. It is also possible to make hypothesis testing between the explanatory variables and the intended result. This type of exploration is not feasible while using ANN, and they have the additional drawback of being computationally intensive (Sargent, 2001). Yang, Liu, & Wang (2003) stated that in the ANN forecast model he developed the precision increased with bigger datasets, but that will also increase the computing time which may not be acceptable for a real-time application. He also pointed out that selecting the appropriate number of implied neural units is difficult and important since it affects learning time, precision and convergence.

Because of these concepts, it is difficult to model and correctly define the parameters for an ANN. There is no specified general methodology to design them, and several authors used a trial and error approach leading to uncertainty when designing the neural network (E. Vlahogianni, Karlaftis, & Golias, 2005). This approach tends to be time-consuming for the researchers that have to rely on their previous experiences to try to define the neural network architecture, number of input variables, number of hidden layers, activation or transfer function and selection of learning or training algorithm (Kumar et al., 2013). Methods based on Genetic Algorithm (GA) try to solve this problem, helping in the selection of the parameters that model the neural network (E. Vlahogianni et al., 2005). Bashiri & Farshbaf Geranmayeh (2011) also proposed a method to find optimal parameters for a neural network using GA concluding that using them the network will have better performance that when compared to the random parameter selection.

While ANN emphasizes on implementation, the other techniques emphasize on estimation and inference, so that it is possible to provide a model and to offer insights on the data and its structure. ANN does not target interpretation, but instead aim to provide an efficient way regarding accuracy and development time to represent the underlying properties of the data and offer good predictions for the subject in study (Karlaftis & Vlahogianni, 2011).

Some studies used alternative prediction methods or variations like event-driven models combined with prediction techniques. Andrea, Klaus, & Walter (2000) developed an event-driven forecasting model to predict on-street parking availability without real-time data and successfully validated the model in the city of Munich, Germany. Teodorović & Lučić (2006) also develop a method that combines simulation, optimization, and fuzzy logic, and they were able to successfully forecast if a park is full or not at a given moment.

Fuzzy logic can also be used alternatively as prediction technique, as demonstrated by Chen, Xia, & Irawan (2013) that developed a model to predict short-time parking availability using historical data from 2 weeks of parking occupancy. Then taking into consideration the current location of the driver, time of the day and his arrival location, the model estimates the trip duration and predicts the expected occupancy in nearby parks at the time of arrival, indicating to the driver the ones expected to have vacant spaces. The model was successfully applied to the city of Perth, Western Australia, providing accurate results quick enough to be used in real-time.

Some authors directly compared prediction models developed with ANN and other techniques, but the achieved results vary according to the characteristics of the dataset and the desired performance.

H. Chen et al. (2001) developed a prediction model based on ARIMA and compared it to other models: two models based on ANN, one based on Radial Basis Function (RBF) and another one based on Multi-Layer Perceptron (MLP). They use the models to predict traffic, which is a different problem but faces the same time series difficulties as parking prediction. The models were used to forecast traffic in a highway and to observe its tolerance to missing data, concluding that when compared to ARIMA both neural networks are more accurate and less sensitive to missing data, generating predictions that are reliable even in a 12 month forecast horizon (Karlaftis & Vlahogianni, 2011).

Another study used a dataset from San Francisco to compare the prediction performance of models using ARIMA, Linear Regression (LR), Support Vector Regression (SVR) and Feed Forward Neural Network (FFNN). They concluded that the FFNN has the best prediction performance but with the longest training time which is over 90 minutes. Whereas ARIMA, LR, SVR only need 39, 12, 20 minutes respectively (X. Chen, 2014). Fengquan et al. (2015) also compared the parking prediction models they developed based on ARIMA and Neural Networks and despite both models provided good accuracy, the ARIMA model was more accurate and with a smaller error.

Cherian et al. (2016) verified in his study that when using the specified dataset the ANN model performed worst when compared to the other prediction methods, and it seemed that the train overfitted the model generating the highest prediction error.

Zheng et al. (2015) in their study compared the developed models using RT and SVR with one that uses ANN. They concluded that both statistical models perform better, but they also stated that the reason for the ANN poor performance is because ANN considers the inner correlation of its input attributes in the modeling, pointing out that the time of day and the day of the week do not have a strong correlation among them.

Table 2 presents some of the studies where they used multiple techniques and compared the results obtained.

Paper and Author	Technique	Limitations	Results
Real Time Prediction of Unoccupied Parking Space Using Time Series Model (Fengquan et al., 2015)	ARIMA	The error increases when high variation occurs in the uncopied parking space.	MAE – 2.22 MAPE – 9.12% RMSE – 4.47
Real Time Prediction of Unoccupied Parking Space Using Time Series Model (Fengquan et al., 2015)	BPNN	Less accurate when compared to the ARIMA Model.	MAE – 3.58 MAPE – 15.55% RMSE – 5.45
ParkGauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics (Cherian et al., 2016)	SVR with PUK	Support Vector Regression results are sensitive to the type of kernel used.	Normalized Root-Mean Squared: 0.0993
ParkGauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics (Cherian et al., 2016)	ANN	The model seems to overfit the training data.	Normalized Root-Mean Squared: > 0.2
Parking availability prediction for sensor-enabled car parks in smart cities (Zheng et al., 2015)	RT	The performance decreases with the increasing number of prediction steps.	(S.F. Dataset) Mean MSE - 0.010 Mean MAE - 0.057 Mean R2 - 0.825
Parking availability prediction for sensor-enabled car parks in smart cities (Zheng et al., 2015)	ANN	Has poor performance because it considers the inner correlation of its input attributes in the modeling.	(S.F. Dataset) Mean MSE - 0.054 Mean MAE - 0.194 Mean R2 - 0.059

Table 2 – Comparison of studies results using different prediction techniques

3. Case Study Background

The following subchapters present the context of this study, starting by describing dataset context and collection method, analyzing what attributes and limitations are in it. The last subchapter describes the tools that analyzed the dataset, and the methods used to elaborate and evaluate the prediction models.

3.1 Dataset Context

The elaboration of this study is in the scope of the urban mobility and traffic management and associated with the MOBIN project application, namely by the use of the dataset made available through Empresa Municipal de Mobilidade e Estacionamento de Lisboa (EMEL), the main project partner to allow an initial exploration of the parking prediction tasks. This dataset contains 3 months of parking occupancy recorded by the on-street parking meters of two parking zones in the city of Lisbon.

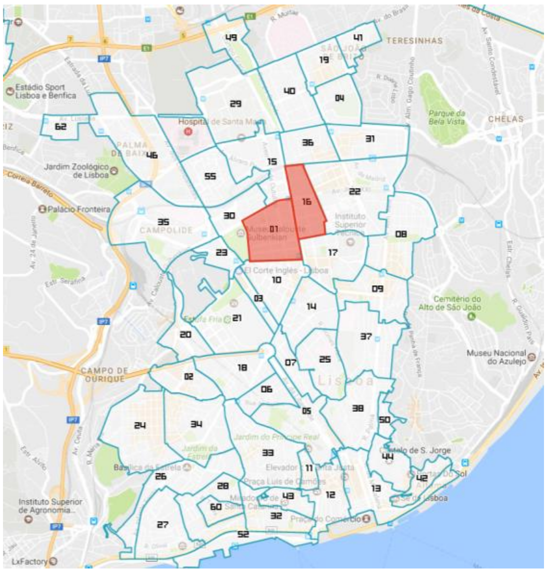


Figure 2 – EMEL parking zones in the city of Lisbon (EMEL, 2015)

Figure 2 shows a map of Lisbon divided into parking zones, each one labeled with a number. EMEL administrates these zones, and their purpose is to establish the boundaries for the free parking permissions assigned to citizens. The dataset used contains parking data from the zone 1, Berna/Valbom, and zone 16, Campo Pequeno, both overlaid with red in Figure 2. It contains 3 months of parking registries collected between 01 September 2015 and 31 December 2015. The original dataset is a raw dump of the information registered in each multi-space parking meter, and they are operational 24 hours a day for 7 days in a week. Each registry corresponds to one parking payment. This is discussed in detail in chapter 4.

The parking zones are located in a central area of Lisbon, and these parking spaces are surrounded by residences, commerce, universities and event venues. These are heavy traffic areas, and consequently, the pollution reached levels above safe, having since 2011 the circulation restricted to vehicles that comply with the emission standard EURO 1.

Each parking zone contains one or more streets, and each street has several parking spaces that may be together or sparse through the street. The parking spaces may be parallel or angled, and they do not have individual parking meters, having instead a set of multi-space parking meters distributed along the street. Figure 3 shows a street with parking spaces and a multi-space parking meter.

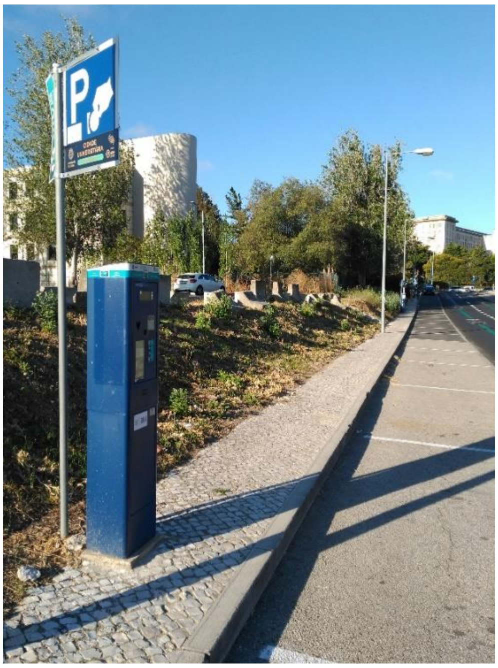
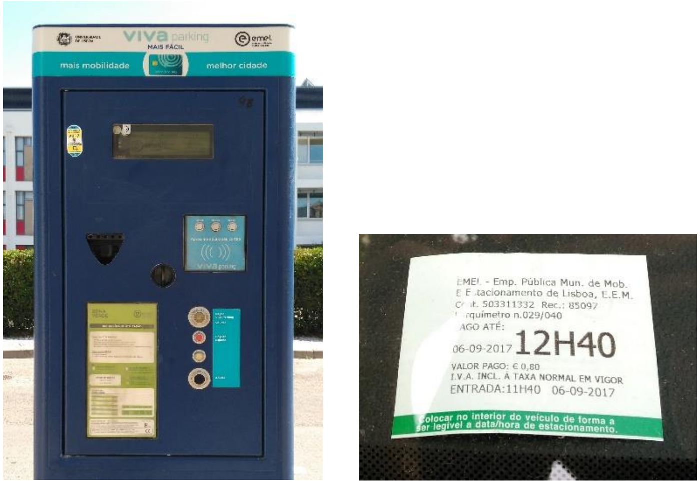


Figure 3 – On-street multi-space parking meter and parallel parking lots

After the driver parks the vehicle, he must search for the nearest parking meter and make the payment in advance. The amount to pay corresponds to the time that the driver wishes to park, and there is no refund if the driver leaves the parking earlier.

The parking meter emits a ticket after the payment indicating the park expiration time, and the driver must put this ticket visible on the vehicle dashboard. To increase the parking time a new payment has to be made, generating a new ticket. On the left of Figure 4 there is a detail view of a multi-space parking meter and on the right an example of a parking ticket in a dashboard.



EMEL - Emp. Pública Mun. de Mob.
E Estacionamento de Lisboa, E.E.M.
Cont. 503311332 Rec.: 85097
Parquímetro n.029/040
PAGO ATÉ:
06-09-2017 12H40
VALOR PAGO: € 0,80
I.V.A. INCL. À TAXA NORMAL EM VIGOR
ENTRADA:11H40 06-09-2017
Colocar no interior do veículo de forma a ser legível a data/hora de estacionamento.

Figure 4 – Detail of a multi-space parking meter and parking ticket

The parking paid time and price varies according to a color code assigned to each street as shown in Figure 5. A zone may have several different colors located at various places, and the paid time may have variations from street to street, having each street a sign indicating the color of that parking as shown in Figure 3.

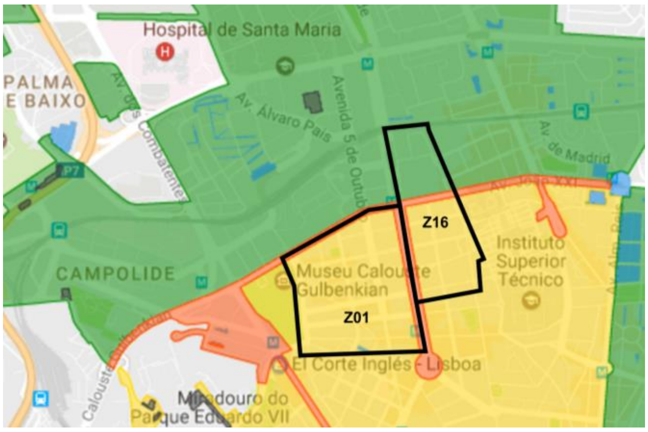


Figure 5 – EMEL parking zone color classification (EMEL, 2017e)

In each parking meter, there is a label indicating the color assigned to it, the parking price and paid times, as shown in the parking meter detail in Figure 4. In the streets near the parking places, there are signs like the ones in Figure 6 that indicate the color that regulates that parking zone.

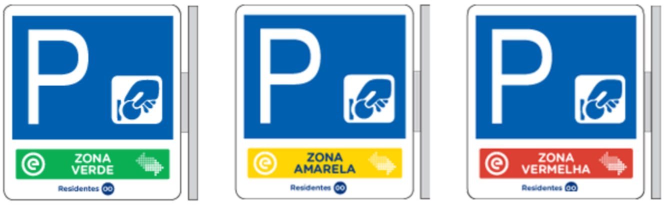


Figure 6 – Signs that indicate the street parking color (EMEL, 2017d)

The price and paid times vary according to the color and street. Green is the cheapest and red the most expensive. Streets with the same color may have different prices.

As shown in Figure 5, zone 1 has yellow and red parking colors, and zone 16 has green, yellow and red parking colors. Table 3 presents the description and paid times according to the parking color in the zones used in the dataset.

Zone Color	Description	Paid Time (Zone 1 and 16)		
		Weekdays	Saturday	Sunday
Green	62% of the parking places; low parking rotation; very residential	9h - 19h	None	None
Yellow	35% of the parking places; medium parking rotation	9h - 19h	None	None
Red	3% of the parking places; high parking rotation; heavy concentration of commerce and services	9h - 19h	9h - 13h	None

Table 3 – Parking zone colors and respective paid times (EMEL, 2017c, 2017d)

EMEL has specific parking permission for the residents and businesses. These permissions are valid for a year, and after that, they may be renewed.

For residents, the parking permission allows parking at any time without payment in the zone corresponding to the residence. Each residence area has a maximum of three permissions assigned to it (EMEL, 2017b). For businesses, is possible to have one parking permit per office or store with a fixed fee, and that allows parking with no payment in the zone where the office or store is located (EMEL, 2017a).

According to EMEL and as displayed in Table 4, the number of emitted parking permissions outnumbers the number of available parking spaces. The total number of parking spaces for zone 1 is 1600 against 2134 permissions, and for zone 16 is 1024 against 1366 permissions.

Zone Number	Zone Name	Available Parking	Resident Permissions	Business Permissions	Total Permissions	Ratio
1	Berna/Valbom	1600	2003	131	2134	1.33
16	Campo Pequeno	1024	1328	38	1366	1.33

Table 4 – Available parking places in the dataset zones (EMEL, 2015)

The Ratio column in Table 4 shows that the number of parking permissions is 1.33 times higher than the number of available spaces, and this means that a vehicle with permission does not have a vacant parking place assured. In this scenario, will be unlikely for a vehicle without permission to find a vacant parking space.

EMEL considers that the vehicles with permission will not park all at the same time. For example, residents with permission will park more during the night, while businesses with permission will park more during the day. This way everyone with or without permission always have the possibility of finding a vacant park. The vehicles parked with these permissions occupy a parking space without the need of registering it in the parking meter, having this a direct effect on how to calculate the occupancy. A street may be full of vehicles parked with permission, but the parking meter has no registers. This may lead to the misleading conclusion that the parking spaces are vacant. This situation is analyzed and discussed in Chapter 4.

3.2 Development Process

As previously stated in section 1.6, this study has several steps between the data acquisition and the development of the prediction model, being these steps represented in Figure 7.

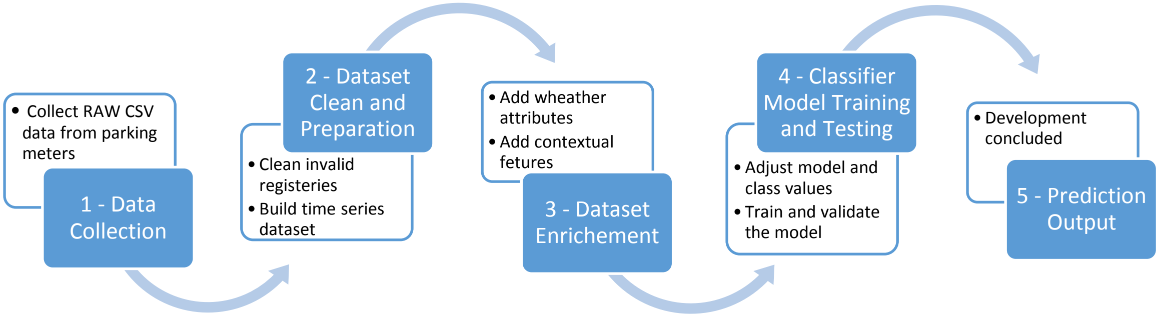

• Collect RAW CSV data from parking meters
1 - Data Collection
2 - Dataset Clean and Preparation
• Clean invalid registeries
• Build time series dataset
• Add wheather attributes
• Add contextual fetures
3 - Dataset Enrichement
4 - Classifier Model Training and Testing
• Adjust model and class values
• Train and validate the model
• Development concluded
5 - Prediction Output

Figure 7 – Development process steps

In the first step, the raw files with parking registries from the parking meters are collected in CSV format. Because most of the parking meters work offline, the data is gathered manually at each one.

In a second step, the data is cleaned to remove registries with errors and to ensure that all registries have their fields correctly inserted. Then, using the data, a time series dataset is built, indicating how many parking spaces are occupied at a given moment and location. Steps 1 and 2 are detailed in chapter 4.

At the third step, the time series dataset is enriched with information about the weather conditions at each time of the day and contextual information about that day, like holidays, strikes, or cultural events.

In the fourth step the classes that will set the status of the parking space are defined. These classes will be the output result of the prediction, where the classifier outputs if the parking is full or not. Then, a classifier is tuned and trained with a split of the dataset while the remaining dataset is used to test the produced model verifying its accuracy.

Steps 3, and 4 are detailed in chapter 5.

3.3 Evaluation

The performance of the classification models is evaluated using a confusion matrix to correlate the actual values with the desired values. In Figure 8 is an example of a confusion matrix where the diagonal represents the number of correct predicted values, being these visible as True Positive (TP) and True Negative (TN). The values in grey represent the prediction errors, False Positive (FP) as the number of negative values predicted as positive and False Negative (FN) as the number of positives predicted as negatives (Powers, 2007).

		Predicted Values	
		Positive	Negative
Actual Values	Positive	TP	FP
	Negative	FN	TN

Figure 8 – Confusion matrix example

The performance of the classifier is obtained by evaluating the Precision, Recall, and Accuracy of the classification.

Precision (1) represents how many of the values predicted for a class really belong to that class. Recall (2) is the percentage of correctly predicted values from all the predictions for a class. Accuracy (3) represents the overall performance of the model.

$$Precision = \frac{TP}{TP + FP} \qquad (1)$$

$$Recall = \frac{TP}{TP + FN} \qquad (2)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad (3)$$

Because Precision and Recall are ratios, the Accuracy does not always give a good representation of the model performance. F1-Score (4) offers an alternative performance metric of the classifier by calculating the harmonic mean between Precision and Recall.

$$F1\,Score = 2\ \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \qquad (4)$$

All these measures output a value between 0 and 1, being better a value closer to 1.

3.4 Development Tools

This section introduces the tools used in this study, explaining where they are applied and the benefits offered.

3.4.1 **Dataset Analysis Using Excel and Visual Basic**

Microsoft Excel in conjunction with Visual Basic for Applications (VBA) is used for the dataset transformation, analysis, and manipulation. There was already previous knowledge of Excel functionalities and VBA programming. This is an important factor because it allows a more efficient use of the tools.

The calculation and graphical possibilities of Excel are adequate to analyze sets of data and perform statistical operations on it. Excel offers many functionalities as standard, and this reduces the necessity of developing custom code, and because it has graphical interface it is easy to visualize the data and quickly gather insights.

VBA is embedded in Excel as macros, and this allows the development of custom code in Visual Basic to interact with the data or use Excel methods sequentially to obtain sets of results. This functionality is used when transforming the dataset, allowing to generate new fields or values by performing calculations on existing fields.

Because Excel works as live sheet it is possible to make quick recalculations and visualize them in a summary or graphical form. This is very useful when making experiments on the dataset.

3.4.2 Prediction Models Development and Evaluation Using Weka

Waikato Environment for Knowledge Analysis (WEKA) is a data mining and machine learning software with widespread acceptance in academic studies and business projects (Hall et al., 2009).

WEKA was already used in previous projects, and the acquired knowledge makes it the preferred tool for this study.

This software has a set of built-in algorithms for regression or classification and a graphical interface that allows easy access to their parameters, providing a fast way to try them and compare their results. It also has tools to preprocess the data and analyze it using the available statistical and graphical options, being possible to apply filters or transformations in the dataset and visualize the result. WEKA is an open source software designed to be modular and expandable, being easy to add new algorithms or functionalities (Hall et al., 2009).

It is also possible to develop a software solution that uses the WEKA functionalities and algorithms through its Java API, making it possible to create an integrated solution for the end user where all the conclusions and experiments of this dissertation may be applied.

In this study, WEKA is used to develop and test the prediction models using different algorithms and to filter the data according to the required experiments.

3.5 Prediction Algorithms

The objective of this study is to obtain occupancy classes and forecast the probability of finding a vacant parking space. To achieve these results, different algorithms are used as classifiers, being their accuracy and computational time analyzed and compared. This subchapter presents the used algorithms with a brief high-level explanation about each one.

3.5.1 J48

J48 is one of the most used decision tree algorithms, being a WEKA implementation of the C4.5 algorithm. It creates a binary tree where each node corresponds to the different attributes, and the branches indicate the possible values that those attributes may assume. This algorithm omits the missing values, calculating these based on the known values (Devasena, 2014; Salzberg, 1994).

3.5.2 Random Forest

This algorithm is considered one of the most efficient when using data with many dimensions. Random forests are a mixture of tree predictors, such that each tree depends on the values of a random vector containing a subset selected randomly from the main dataset, being this applied with the same distribution to all trees in the forest. The final tree classifier is an aggregation of all the results of all the generated trees. Random Forest has the advantage of being a fast algorithm that produces good results without almost no parameter tuning (Breiman, 2001; Devasena, 2014).

3.5.3 REP Tree

Reduced Error Pruning (REP) Tree works by applying regression tree logic to generate multiple trees in altered iterations and then section the best one from all the generated trees. The decision tree is constructed using variance and information gain and the tree is pruned using reduced-error pruning with back fitting method. It handles missing values with C4.5’s method of using fractional instances (Devasena, 2014; Quinlan, 1987).

3.5.4 Multilayer Perceptron

Multilayer Perceptron (MLP) is a feedforward Neural Network that contains at least one hidden layer. Each neuron is connected by weights and his output is a sum of their inputs after being modified by the activation function. During training, it uses backpropagation to set the weights of the neurons in order to find the combination with the smallest error. (Gardner & Dorling, 1998).

4. Dataset Analysis

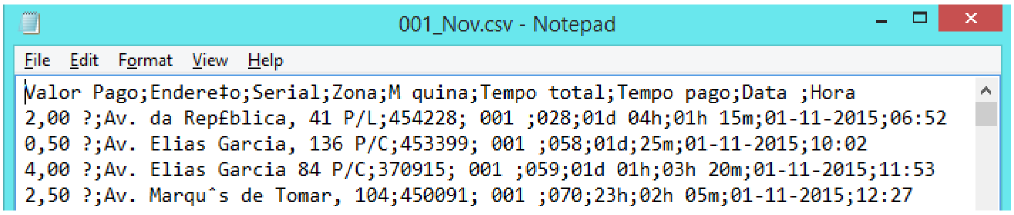
This chapter describes the raw data obtained and how the transformation of the dataset occurred. The dataset is then analyzed to observe how it behaves and patterns it follows.

4.1 Original Data Characteristics

As previously described in section 3.1, the original dataset contains historical observations of the parking occupancy from 2 zones in the city of Lisbon, collected from on-street parking meters during 3 months, being this a sample from a full year of parking occupancy.

4.1.1 Data Attributes

The raw data obtained from the individual parking meters is in CSV format and using ASCII charset encoding. This encoding originates some errors in chars with accents or special chars, as shown in Figure 9.



```
Valor Pago;Endere‡o;Serial;Zona;M quina;Tempo total;Tempo pago;Data ;Hora
2,00 ?;Av. da Rep£blica, 41 P/L;454228; 001 ;028;01d 04h;01h 15m;01-11-2015;06:52
0,50 ?;Av. Elias Garcia, 136 P/C;453399; 001 ;058;01d;25m;01-11-2015;10:02
4,00 ?;Av. Elias Garcia 84 P/C;370915; 001 ;059;01d 01h;03h 20m;01-11-2015;11:53
2,50 ?;Av. Marqu^s de Tomar, 104;450091; 001 ;070;23h;02h 05m;01-11-2015;12:27
```

Figure 9 – Sample of the CSV file with original data

The errors in the chars originate mistakes mainly when grouping the parking meters by street, because the address of each parking meter needs to correspond to one of the possible addresses, so that the registries in it can be assigned to a specific street.

The first line of the CSV is a header with the name of the registry attributes, being each one of the following lines a parking registry. As shown in Table 5, there are 9 attributes in the original data.

The attribute “Amount Paid” indicates the amount of money received in that registry, but this information will not be used in this study. The amount of parking time that the payment corresponds to is another attribute.

Original Attribute	Description
Valor Pago - (Amount Paid)	Amount of money received
Endereço – (Address)	Address with the location of the parking meter
Serial	Parking meter serial number (unique)
Zona – (Zone)	Parking zone where the parking meter is located
Máquina – (Machine)	Parking meter number in that zone (unique by zone)
Tempo Pago – (Time Paid)	PAID time the vehicle is allowed to park (Calculated according to the Amount Paid and the fees for that zone)
Tempo Total – (Total Time)	TOTAL time the vehicle is allowed to park (Calculated according to the time paid and free periods)
Data – (Date)	Payment date
Hour – (Hour)	Payment time in hours and minutes

Table 5 – Attributes in the original data

The “Address” and “Zone” attributes indicate the location of the parking meter and are used to group the registries by street and by zone if the street has 2 zones.

Attribute “Serial” is the serial number of the parking meter and is unique to each one, while attribute “Machine” is the parking meter number in the zone he belongs, being unique by zone.

“Date” and “Hour” indicates the time at which the registry was made.

“Time Paid” is an attribute calculated by the parking meter according to the fares configured on it, indicating the duration of paid parking time according to the “Amount Paid”.

“Total Time” takes into consideration the time of registry, the “Time Paid” and the hours where the parking is paid or free, according to park color and the fees in the parking meter. If the end of “Time Paid” overlaps a free parking hour, the remaining paid time will pass to the next paid period. This calculation takes into consideration the free periods during the weekends and holidays. This attribute returns the absolute allowed parking time, including paid times and overlapped free times.

4.1.2 Data Distribution

The original data contains 250995 parking registries originated by 110 parking meters distributed along 23 streets. Each street has a zone assigned, except 5 streets that have parking meters in both zones. This information is detailed in Appendix A.

As shown in Table 6 each zone has 14 streets, but zone 1 has more parking meters and more registries.

Park Zone	Number of Streets	Number of Parking Meters	Number of Registered Parks	% of Total Registered Parks
Zone 1	14	76	177254	70.82%
Zone 16	14	34	73041	29.18%
TOTAL	**28**	**110**	**250295**	**100.00%**

Table 6 - Registries distribution by zone

Table 7 shows that most of the registries are located in streets with yellow color, which means medium parking rotation. It is expected that the yellow parking color has a smaller number of average registers by parking meter when compared to the red parking color, which has the highest average rotation by parking meter.

Only 5 parking meters are located in the green parking color, and these are also the ones with the smallest average number of registries by parking meter, which is expected since this is a low rotation parking color.

Park Color	Number of Streets	Number of Parking Meters	Number of Registered Parks	Average Registers by Park. Meter	% of Total Registered Parks
Red	6	27	71286	2640	28.48%
Yellow	20	78	171861	2203	68.66%
Green	2	5	7148	1430	2.86%
TOTAL	**28**	**110**	**250295**		**100.00%**

Table 7 – Registries distribution by parking color

The streets in the data have different lengths and types of rotation, so the number of parking meters varies and, consequently, the number of registries also varies. The graph in Figure 10 displays how the numbers of registries are distributed along the streets.

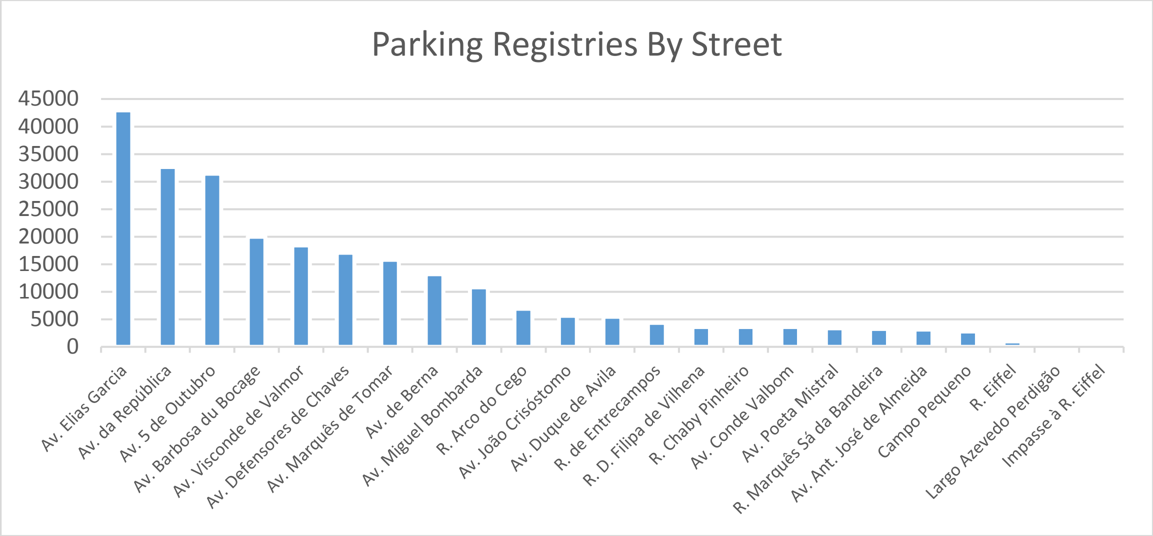


Figure 10 – Registries distribution by street

Is Figure 10 is visible that there is a concentration of the registries in a small number of streets, with almost half of the streets in the data having a low number of registries. Table 8 shows the number of registries by street and it is visible that half of them are concentrated in 4 streets. This information is detailed in Appendix B, and detailed by parking meter in Appendix E and Appendix F.

Zone	Street	Park Color	Number of Parking Meters	Number of Registered Parks	% of Total Registered Parks	Accumulated % of Total Registered Parks
1 / 16	Av. Elias Garcia	Yellow	8	42937	17.15%	17.15%
1 / 16	Av. da República	Red	10	32642	13.04%	30.20%
1	Av. 5 de Outubro	Yellow	12	31439	12.56%	42.76%
1 / 16	Av. Barbosa du Bocage	Red	4	19992	7.99%	50.74%

Table 8 – Streets that contain 50% of the registries

The concentration of registries happens mostly in streets with two zones what is expected since they may have a greater area. However, Table 8 also shows that the street with most registers (Av. Elias Garcia) is not the one with more parking meters, meaning that is not possible to correlate these two characteristics. This street is also on yellow park color, what is not expected since the color with more supposed parking rotations, and by consequence registries, is the red park color (EMEL, 2017d).

This uneven distribution of registries by street may cause problems in the prediction models, because the noise introduced by the streets with low registries may reduce the accuracy when predicting values for streets with most registries. It is also more difficult to train a model for the streets with less registries because of the small amount of data.

As stated previously in section 3.1, the raw data was collected between 01 September 2015 and 31 December 2015. The graph in Figure 11 represents how the registries are distributed along each week, being Monday the first day of the week. This information is detailed in Appendix C.

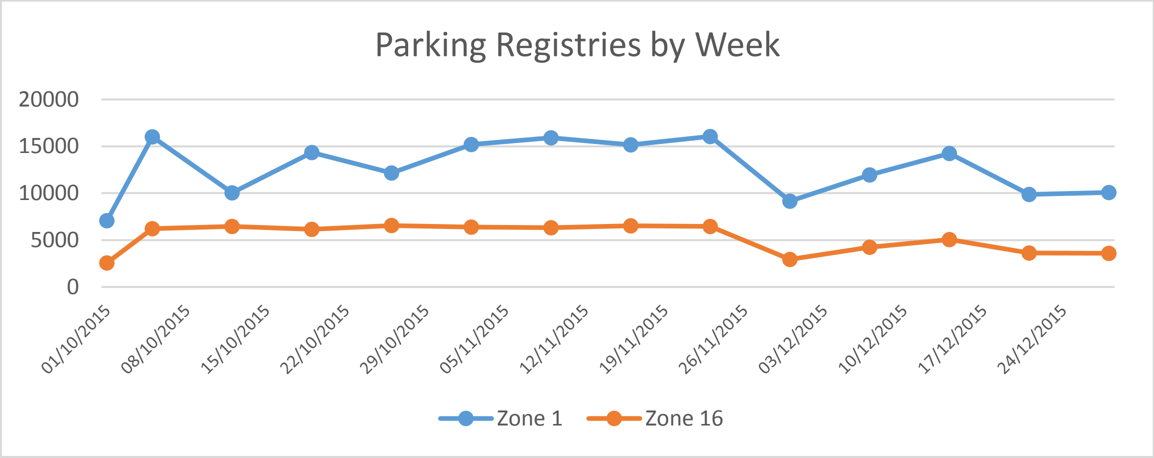


Figure 11 – Registries distribution by week

Figure 11 shows that in October and November zone 16 has an even distribution of registries along the weeks. However, zone 1 has an uneven distribution for this same time. These differences in zone 1 occur consistently in the streets with most registries and are less evident in the other streets. This consistent behavior indicates that the parking demand may have been affected by an environmental factor. However, the cause for this is difficult to trace because this study is made 2 years after these registries occur.

In December, both zones have an uneven distribution of the number of registries along the weeks, what is expected due to the holidays, christmas and vacations, being these factors responsible for abnormalities and spikes in parking demand. These results are detailed in Appendix C an Appendix D.

The uneven distribution of the registries through the weeks may cause problems when training and testing the models, since the number of instances may not be enough to represent the parking pattern correctly.

Table 9 shows how the registries are distributed along the months. As already seen in Figure 11, December has less registries mainly because of holidays and vacations. Although there are differences in the registries distribution along the weeks during October and November, Table 9 shows that both months have a similar quantity of registries.

Month	Start Date	End Date	Registries Zone 1	Registries Zone 16	Number of Registries	% of Total Registries
October	01/10/2015	31/10/2015	59587	27906	87493	34.96%
November	01/11/2015	30/11/2015	65531	26951	92482	36.95%
December	01/12/2015	31/12/2015	52136	18184	70320	28.09%
TOTAL			**177254**	**73041**	**250295**	**100.00%**

Table 9 – Registries distribution per month

The registry distribution along the weekdays for the whole dataset is shown in Figure 12.

The number of registries does not have significant oscillations, however different parking zones have different behavior. In zone 1 the average number of registries continually decreases until Thursday, where it has less 7.85% than Monday, having then an increase of 3.15% on Friday. Zone 16 also has a decrease of 7.07% after Monday, but then the number continually increases until Thursday, where it has gained 7.95%, having then a decrease of 7.93% on Friday. In both zones, there is a small amount of the registries in the weekend days because only 6 streets are in the red park color, which is the only one with paid parking on weekends. These are the source of most of the weekend registers.

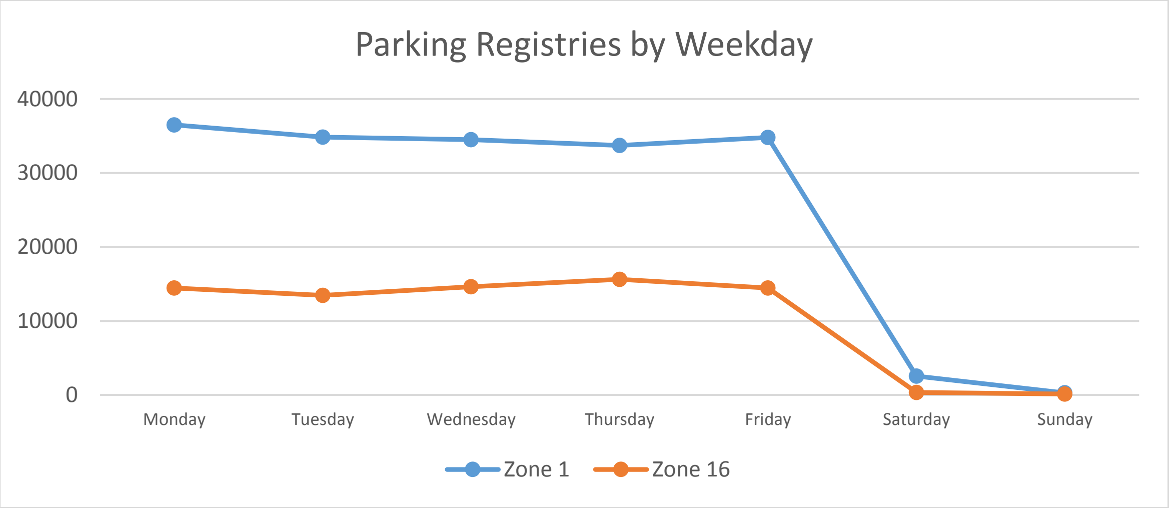


Figure 12 – Registries distribution by weekday

Table 9 and Figure 12 data are detailed in Appendix C.

4.1.3 Data Limitations

The content and attributes of the data have some limitations in the information they provide about the parking occupancy.

As stated in section 4.1.1, the attribute “Machine” is the parking meter number in the zone it belongs, being unique by zone. Observing the dataset is visible that for each zone this is a continuous number count starting at 1. However, there are some missing machine numbers in the series, and it is not possible to understand if the parking meters associated with those “Machine” numbers have been deactivated, replaced or are simply missing in the data. This information is detailed in Appendix E and Appendix F. If data is missing, the number of registries for a street is inaccurate and may cause an error in the occupancy classification. The data also does not provide information about parking meters with malfunctions or failed parking registries.

All the parking meters are located on a street and register the parking for that street, however they can also emit registers for a neighbor street as long as it has the same parking color and fares. This causes “*bleed*” in street registries, since a vehicle parked in one street may be registered in another. It is assumed that all the parking meters suffer from this register “*bleed*”, so this factor was not considered when calculating the occupancy.

It is not possible to know the real number of parked vehicles because the dataset only counts registered parks. However, a vehicle can be parked with a residential or business permission or even illegally parked. Also, it is not possible to know the vehicle departure time, since the vehicle can leave the parking place before or after the end of the paid time. In these situations, the parking meter does not have any registry but the parking place can be occupied, being difficult to obtain the real number of cars parked in the street.

The registries that occur during the weekdays free periods and weekends are assumed to be of vehicles that wish to stay parked when the next paid period starts. Therefore, no registries exist for all the other cars that parked during non-paid periods, being impossible to calculate the occupancy during free periods and weekends. The exception are the streets with red parking color, where is possible to calculate the occupancy for the Saturday paid period.

Contextual factors may have impact on the number of parking registries, like construction works or roadblocks, but these factors are difficult to obtain and could not be traced at the time of this study.

4.2 Dataset Transformation

To develop this study data must be transformed into a time series dataset with the number of vehicles with valid paid parking and the number of registries in that period.

After some experiments to obtain a balance between the resolution of the dataset and its size, the time series resolution was set to 10 minutes with granularity at street level. Smaller time resolutions or lower granularity exponentially increase the computation time and generated many observations with 0 occupancy and 0 registries, mainly because of the streets with low registries.

Table 10 presents the attributes included in the time series dataset.

Dataset Attribute	Description
Zone	The zone where the street is located
Street	Street where the occupancy is located
ParkColor	Parking color assigned to the street
TimeStamp	Timestamp to which the occupancy corresponds
Date	Date to which the occupancy corresponds [dd/MM/YYY]
Hour	Time to which the occupancy corresponds [hh:mm]
Weekday	Weekday corresponding to the occupancy date
Registries	Number of registered parks
Occupancy	Number of vehicles that have a valid paid parking

Table 10 – Attributes in the time series dataset

The “Zone” and “Street” attributes indicate the location of the registries and occupancy, and “ParkColor” indicates the parking color of the street.

“TimeStamp”, “Date” and “Hour” indicate the time of the observation. “Weekday” contains the name of the day where the observation was made.

“Registries” contain how many parking registries have been made between the last observation and the current. This attribute is generated by counting the parking registries made in the previous 10 minutes (if this is the resolution) in the parking meters that belong to the street being observed.

The attribute “Occupancy” indicates the number of vehicles in the current street that have a parking registry with an expire time equal or after the observation time.

The expire time is calculated by summing the register time to the “Total Time” attribute in the original data. “Total Time” indicates the total time of allowed park including paid and free periods, as already explained in section 4.1.1. At the end of the paid period, there are situations where the user does not have the exact amount of money and the payment made exceeds the limit of the paid period, but the vehicle does not stay overnight (all the next free period) and still leaves at the end of the day. To filter these type of situations and prevent ghost occupations, if the "Total Time" expires less than 15 minutes after the end of the paid period, the expire time to consider will be the end of the paid period.

4.3 Occupation Patterns

The time series dataset is analyzed to obtain insights about the occupation and registries patterns in time, identifying peak hours and daily averages.

The measurements only consider paid parking periods with a slight time offset before and after, because it is only possible to control the occupancy when parking registry is mandatory.

The considered periods are weekdays between 07h30 and 19h30. The values in the measurements are the average values of the corresponding attribute in the time series dataset.

4.3.1 **Occupancy by Zone**

Analyzing the occupancy by zone gives an overall view of the occupancy and peak values. The graph in Figure 13 presents the maximum, average and median occupancy for each zone and show that the values tend to have a normal distribution, with the median value slight above the average. These values are detailed in Appendix G.

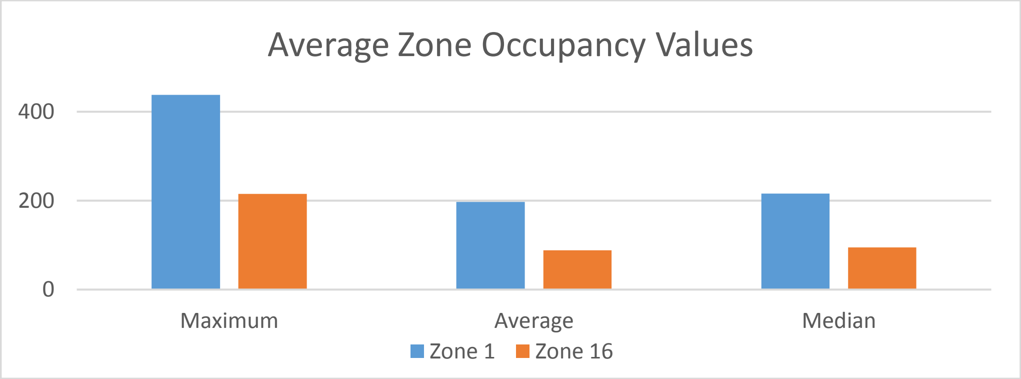


Figure 13 – Average zone occupancy values

This graph shows a maximum occupancy value of 438 to zone 1 and 215 to zone 16, however these values differ from the maximum available parking announced by EMEL of 1600 to zone 1 and 1024, as stated in section 3.1. This leads to the conclusion that at the peak hours the majority of the parking spaces are occupied by cars without parking registry, probably because they have parking permission or are illegally parked.

Figure 14 shows that both zones have a similar occupancy behavior during the day, starting at the beginning of morning during the free period (07h30) with the vehicles making registries because they pretend to only leave after the start of the paid period (09h00). After the start of the paid period, the occupancy values steadily increase until mid of the morning (11h00) having then a drop until the start of the lunchtime (13h00). The values then increase until 15h30, having stepper values in the period after lunch (14h30). After this, the values grow at a steady rhythm until 17h30 where the occupation values remain flat due to the rate of cars leaving being equal to the rate of cars arriving to stay overnight. After 18h30 the occupation registries heavily drop until the end of the paid period (19h00).

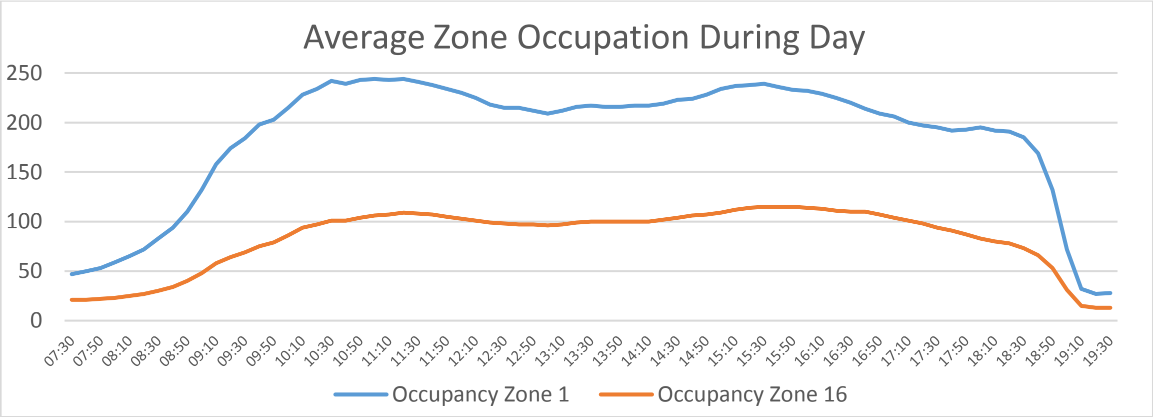


Figure 14 – Average zone occupation during day

Figure 15 shows that the number of registries follows a pattern similar to occupancy, but when the occupancy reaches its peak the number of registries decreases, leading to the conclusion that the parking is full and few or no registries can be made. Registries have a less uniform pattern when compared to the occupancy, indicating that many short-term parking events occur at certain hours, as can be seen around 13h00. At this time there is a peak of registries, but the occupancy remains flat indicating that this is short-term parking probably because of many services and commerce that close at 13h00.

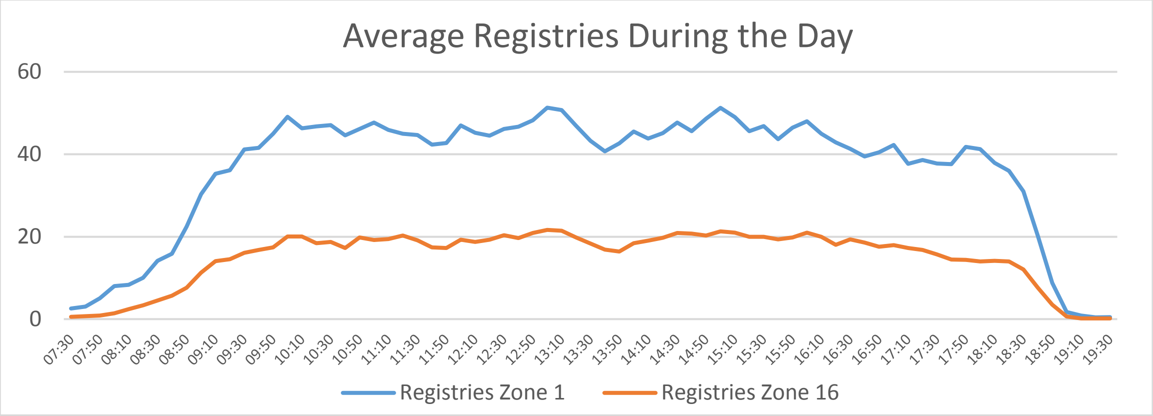


Figure 15 – Average zone registries during the day

4.3.2 **Occupancy by Street**

The average maximum occupancy in each street varies according to the parking availability and the size of the street. Figure 16 shows these variations, with only 2 streets having an average maximum occupancy of over 100 vehicles. This information is shown in detail in Appendix G.

The streets with greater occupancy values are the ones with higher number of registries, as shown in section 4.1.2. However, these values are not directly proportional since “Av. Da Republica” has more registries than “Av. 5 de Outubro”, but the latest has a higher average maximum occupancy. Several situations may be responsible for this situation as stated in section 4.1.3.

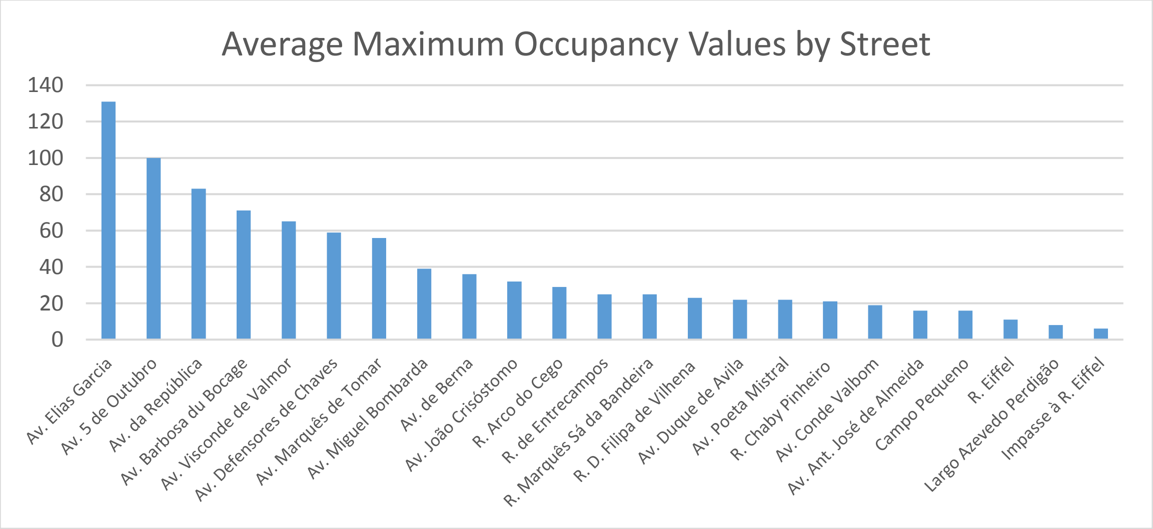


Figure 16 – Average maximum occupancy values by street

The streets follow the occupancy and registry patterns identified for the zones, as can be seen in Figure 17 and Figure 18. These patterns are more pronounced in the streets with higher occupancy, becoming less evident as the streets have less occupancy and getting almost flat in the streets with a low maximum occupancy. Because of this in the following graphs, only the 8 streets with higher occupancy are shown.

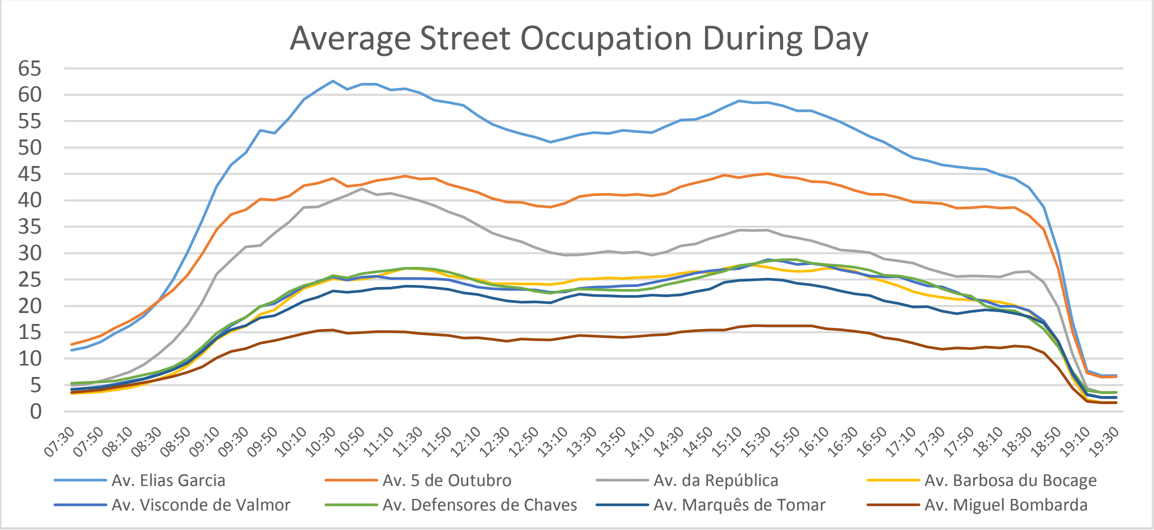


Figure 17 – Average street occupation during day

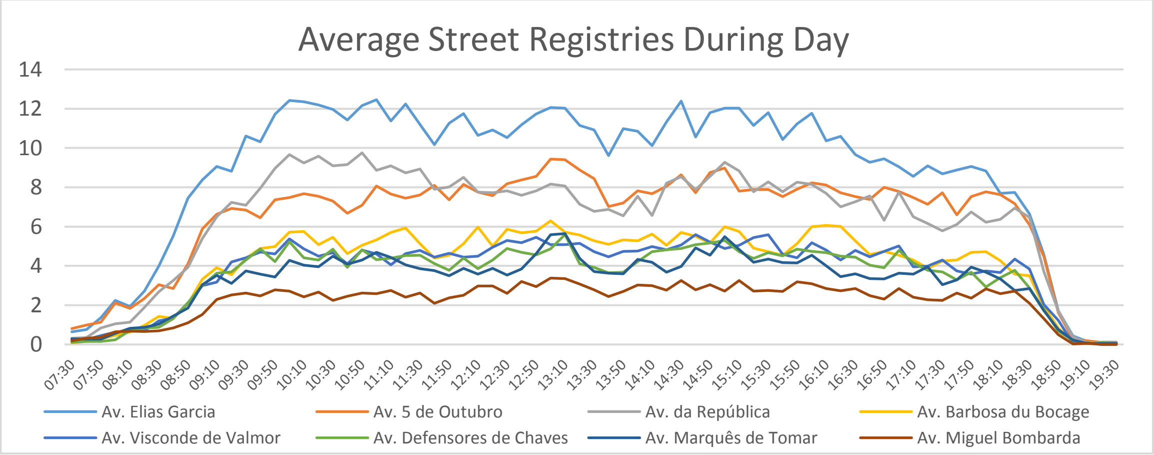


Figure 18 – Average street registries during day

4.4 Outlier Days

Occupation patterns may change because of external events or special situations, being these cases difficult to detect and resulting in data with noise. To detect these abnormalities, an outlier detection is performed in the dataset by comparing each time unit with the similar ones from the previous weeks. To ignore invalid outliers and provide a more robust model the time unit is only considered an outlier if at least half of time units in a day are also considered outliers.

Outlier detection is made using Interquartile Range (IQR). The first step is to calculate the quartiles (5) of the values in the analyses to obtain the value of the IRQ. Using IQR and the quartile values, it is established the upper and lower outlier limits (6), considering outlier the values under or above these limits.

$$\begin{gathered} Q1 = \frac{1}{4}(n+1) \qquad Q3 = \frac{3}{4}(n+1) \\ n = number\ of\ values\ in\ the\ dataset \end{gathered} \tag{5}$$

$$\begin{gathered} IQR = Q3 - Q1 \\ Lower\ Outlier\ Limit = Q1 - 1.5(IQR) \\ Upper\ Outlier\ Limit = Q3 + 1.5(IQR) \end{gathered} \tag{6}$$

In Figure 19 are the number outliers days for each street, being the streets ordered by occupancy. The weekends and holidays are not considered since they are free parking periods.

Streets with more data are more prone to have outliers since their behavior is more detailed and with higher dynamics. This is the situation for “Av. 5 de Outubro” that has 6 outlier days. However, a direct relation between occupancy and number of outliers cannot be made, since there are other streets with high occupancy values that don’t have outliers and streets with low occupancy values that have outliers.

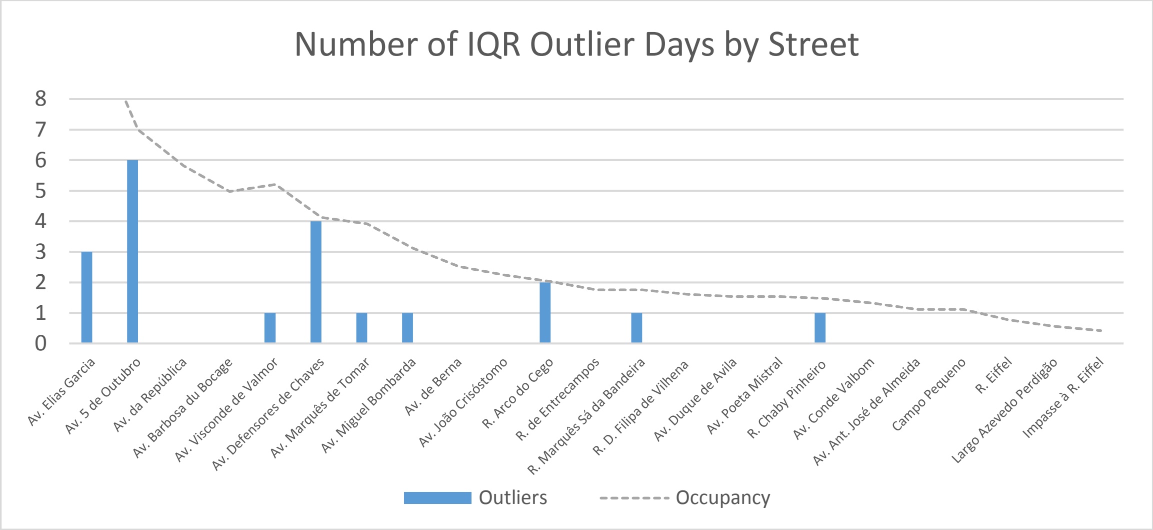


Figure 19 – Number of outlier days by street

Considering only the streets with outliers, the average number of outliers per street is 2.22 days, which in the universe of 64 weekdays represents 3.5% of the data. This is a small percentage of the dataset, however outliers may influence the prediction models, reducing their precision because of the added noise or in other situations increasing the precision by producing a less overfit model, being this discussed in section 6.2.

The number of outlier days has higher incidence during December as shown in Figure 20, and this behavior may be explained by the number of holidays, vacations and abnormal commerce activity during December. The week of 30-11-2015 is the only one that registered an exceptional number of outliers, and because of this it may be considered as an outlier week. This information is further detailed in Appendix H.

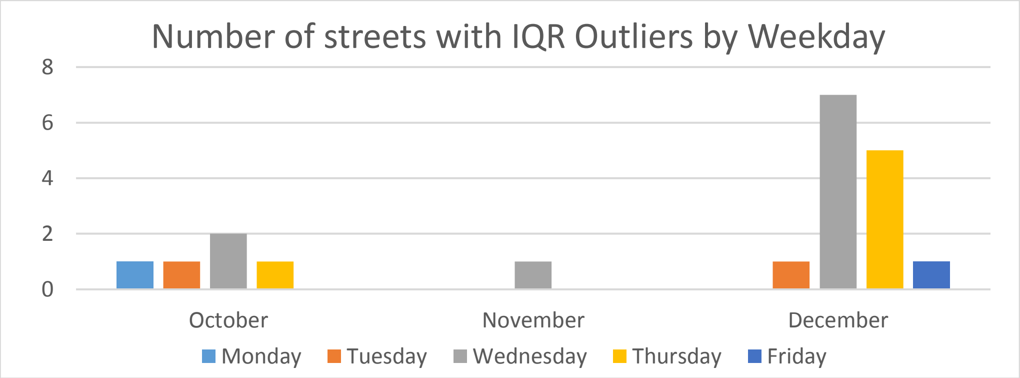


Figure 20 – Number of streets with outlier by weekday

4.5 Weeks and Weekdays Behavior

To observe how similar the occupation behavior is between different weeks and weekdays a correlation study was performed using the person coefficient (7). In this formula, X and Y correspond to two matching occupation observations for different days, which will result in a value between -1 and 1, having a smaller correlation as the values get closer to 0.

$$-1 \geq pearson = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2 \sum(Y-\bar{Y})^2}} \leq 1 \qquad (7)$$

If results show high correlation, it is an indication that the days have a very similar behavior and a lower prediction error can be achieved, also meaning that the behavior is constant and that the dataset can be reduced while still maintaining the accuracy.

Pearson can measure difference in the behavior between two days regardless of scale. This disregard for scale enables the system to detect similar dynamics, but often these occur at very different scales. This makes possible that days with very different parking flows are considered similar by this measure, however it is not possible to know how similar they are.

To verify this similarity, it is used the Absolute Value of the Difference (AVD) (8) where the result indicates the difference in occupation values between two days.

$$AVD = \frac{\sum|X-Y|}{n} \qquad (8)$$

Being X and Y two matching observations for different days, this formula is the sum of the absolute value of the difference between each observation in a day, dividing by the total number of observations in both days (n). The larger the average distance the less similar are the values, meaning that the days may have the same behavior but have different occupancy values.

The graph in Figure 21 shows the average weekly occupation of the dataset and is visible that despite the occupation patterns having slight variations from week to week, the overall behavior is consistent with the patterns previously analyzed in section 4.3.1. This is confirmed by the high correlation between weeks, with an average value of 0.83 for zone 1 and 0.85 for zone 16.

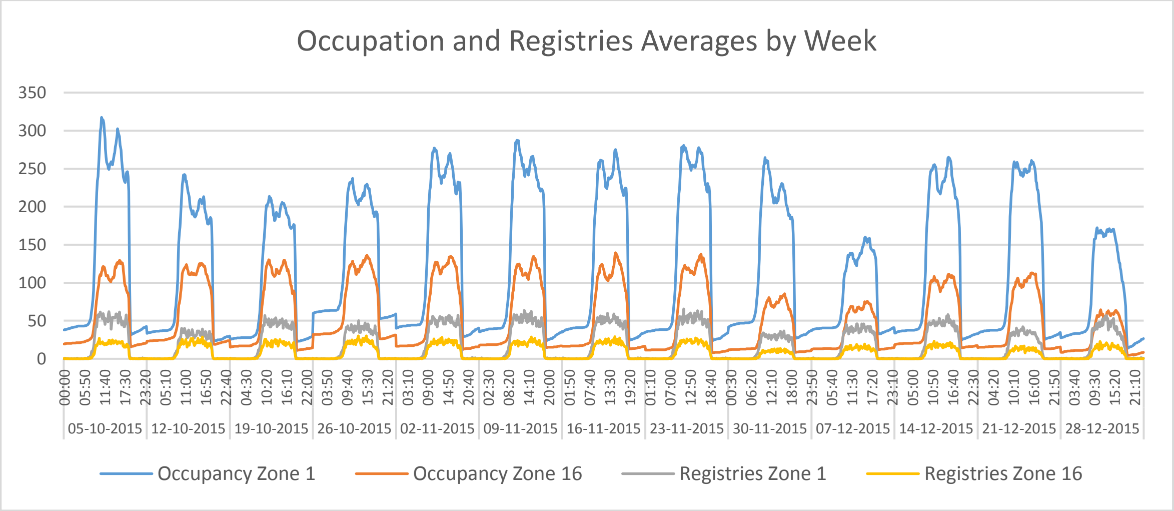


Figure 21 – Occupation and registries averages by week

However, the average occupation values show an inconsistent behavior along the weeks reflecting the registries behavior already analyzed in section 4.1.2, where the weeks in zone 1 have an inconsistent behavior during October and December, while in zone 16 the weeks are only inconsistent during December. Appendix I shows the occupancy values by street and is visible that this behavior is consistent through the streets with higher occupation. The average distance obtained for zone 1 is 48.30, that when considering the average of occupancy of 197 represents a variation of 24.52%. For zone 16 the average distance is 22.32 that indicates a variation of 25.36% considering an average occupancy of 88.

The relation between weekdays is in Figure 22, where it is shown that the days have a similar occupancy pattern, being this confirmed by the average correlation value of 0.87 for zone 1 and 0.86 for zone 16.

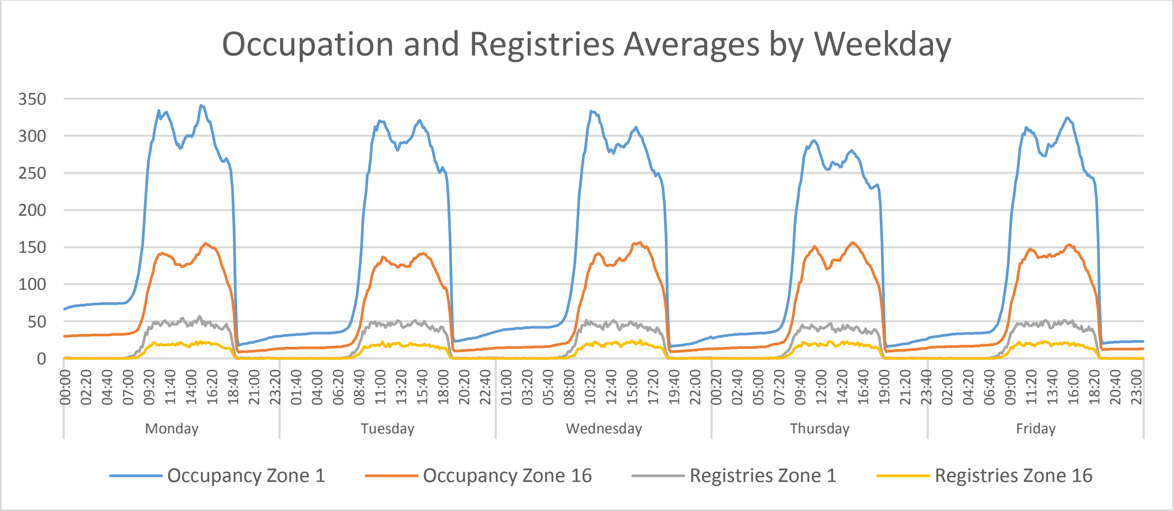


Figure 22 – Occupation and registries averages by weekday

Despite this, the occupancy values have differences, with Thursday showing a lower average occupancy when compared to the other days. The average distance for zone 1 is 50.73 and 20.33 for zone 16, which are relevant values as previously seen.

Appendix I shows the occupancy by street and weekday, where is shown that the behavior on Thursday is consistent through the streets with higher occupancy. Some Thursdays are considered outlier as already seen in section 4.4, however the number of outliers is not relevant and is still smaller than the outliers found on Wednesdays. No reason was found for this discrepancy and is believed that this may be either a fluctuation that would disappear in larger datasets, or an environmental factor that could not be traced at this time.

According to the results obtained from this analysis is possible to conclude that the occupation follows the same patterns through the different days, however the occupancy values have significant variations through the days, indicating that the data behaves differently according to the date and environmental events.

The correlation values and AVD for each street are presented in detail in Appendix J.

4.6 Occupation Classes Classification

As previously stated in 3.2, the objective of this study is to develop a classifier that indicates the status of the parking in a specific street, being able to provide binary or multiple classifications, like vacant, almost full and full. This approach is more effective than trying to predict the number of occupied parking’s, since that would be misleading to the drivers because the dataset does not contain real-time information and many variables can affect the exact number of parked vehicles in a 10 minute interval. Also, in the case of this dataset, the number of registered parked vehicles cannot be used alone to establish if a parking is full or vacant, since many more vehicles can be parked without registration, being those not taken into account by the dataset as previously discussed in section 4.1.3.

To determine the parking occupancy status, first it is considered that a parking location is in the high demand period when its occupancy is close to reaching the maximum value. At these peak hours, it is expected to have a high number of registries, but if the parking location is full these registries will not occur. However, at these peaks hours the number of vehicles departing can also be high, allowing newer vehicles to park and by consequence generate registries. In this case, the parking is the high demand period but the vehicle rotation allows new vehicles to park. This rotation value (9) is calculated dividing the number of registries by the occupation value at a specific moment.

$$rotation = \frac{registries}{occupation} \quad (9)$$

When the parking in a street is close to reach its maximum, the rotation value is used to verify if new vehicles are still able to park. A rotation value closer to 0 indicates that a very small number of vehicles are being able to park, so the parking in that street is considered full. In Appendix L is an example of how the rotation values behave across the day in function of occupation and registries.

The maximum occupancy is only an indicator that the parking is in the high demand period, so it is necessary to define a threshold below this maximum from where the parking is considered to be in this state. This threshold point is sensitive and has a direct impact on the results. If it is too low, the rotation is calculated for periods with low registries but with low occupancy generating false “Full” status. If it is too high, periods with high occupancy and low registries may be discarded originating fake “Vacant” status. In Appendix K is an example of a classifier with multiple classes and a diagram demonstrating this behavior.

As previously discussed at section 4.5, the parking occupancy has significant variations from street to street and from weekday to weekday, so an accurate threshold needs to be calculated relative to the maximum occupancy value in each street and for each weekday. To compensate the fluctuations in maximum occupancy values through the different weeks, the considered maximum value for a weekday is an average of the 12 highest occupancy values (1 for each week in the dataset) in a street for that specific weekday.

4.7 Prediction Based on Previous Weeks Values

Variations in the data along the time and across the different streets were already discussed in sections 4.3 and 4.5, however it is relevant to observe how much these variations are relevant if the prediction is solely based on the previous observations. The conclusions and results of this section are used as a baseline for the results using the prediction algorithms.

4.7.1 Prediction Method

The objective of the prediction based on previous values is to obtain the occupancy status for a specific moment in a weekday using the values of the same weekday in the previous weeks. The result of the prediction is compared with the actual result to verify its accuracy. The results are then presented in a confusion matrix to evaluate the precision.

To explore different possibilities and verify how the precision changes, several methods are used to obtain the previous values:

- **Week Before** – It is considered that the current week will have the same occupancy status as the week before.
- **Previous Weeks (Average Values) -** It will obtain all the occupancy values and registries from the previous weeks, average them and then obtain what will be the occupation status for the current moment.
- **Previous Weeks (Status Frequency) -** It will obtain the occupancy status from the previous weeks and verify which one is the more frequent, and then assign it to the current week.
- **Previous Weeks Excluding Outliers –** It will exclude from the calculation the previous week's where the occupancy at that moment is an outlier and at least 6 values in the neighborhood are also outliers. The predictions using previous weeks by average and by status frequency are recalculated without outliers, being the results presented separately.

This prediction method will only start in the fourth week of the dataset. For the first weeks, there are no previous values available or a significant amount of data to establish averages, which may generate misleading results.

4.7.2 Occupancy Classes Definition

To perform this test, the occupation status at each moment is set by 3 classes: “Vacant”, “Almost Full” and “Full”. Compared to the binary classification, the 3 class system gives more information to the driver, allowing him/her to make the decision of take a chance to park in a specific street knowing in advance that a free space is difficult to find, or to choose a nearby street with vacant park.

As previously explained in section 4.6, the prediction results are sensitive to the definition of the high demand period threshold and the class rotation limits. As seen in section 4.3.2 the most relevant occupation values are concentrated in 7 of the 23 streets, with the other having very low occupancy values.

To give clearer results only these 7 streets are considered in Table 11, where is shown the average occupancy of these streets by weekday. The “Absolute Maximum” is an average of only the peak value for each weekday, where the column “Top 12 Max” is an average of 12 peak values for each weekday.

As expected, the average using the 12 peak values (Top 12 Max) is slightly smaller, what is consistent with the variations in the maximum occupancy observed through the weeks.

Average Weekday (Absolute Max) Occupancy	**Average Weekday (Top 12 Max) Occupancy**	**Remain Vacant Parking Spaces at 10% Threshold**	**Average Rotation in High Demand Periods (10% Max Threshold)**
76.7 +/-22.6	73.6 +/-20.2	7.36	15.07%

Table 11 – Average rotation values for the 7 streets with higher occupancy

After analyzing the average street occupancy values, the threshold for the high demand period it is set to 10%, which according to Table 11 starts when only 7.36 parking spaces remain to reach the maximum. Analyzing this by street is possible to verify that on streets with higher occupancy this threshold value starts when 12 parking spaces remain to reach the maximum occupancy, which is the expected for a longer street where 12 vacant parking spaces may be difficult to find. In the streets with lower occupancy, the high demand period starts when 1 parking space remains below the maximum. With the 10% threshold set, Table 11 shows that the average rotation during the high demand periods is of 15.07%, so this value is used as the rotation threshold for the prediction using 2 occupation classes, setting it to 15%. This information is detailed by street in Appendix L.

In summary, these are the values used for the prediction based on previous values:

- **High Demand Period Threshold:** 10% before reaching maximum occupancy;
- **Full Class:** High Demand Period + Rotation value under 15%;
- **Almost Full Class:** High Demand Period + Rotation value under 20%;
- **Vacant Class:** Remaining situations.

4.7.3 Results

Table 12 shows the average results obtained for the parking occupancy prediction through the different weeks using the different methods.

In all the methods the accuracy is around 79%, however the F1-Score shows a value around 27%, which is a low value indicating poor prediction performance by the models. The high accuracy values occur because the “Vacant” class has a high precision and recall values, both with an average value of 79%.

However, in all the methods the class "Full" as only a precision around 1%, being this even smaller for the class "Almost Full". Both these classes are less frequent in the dataset than the “Vacant” class and, as a consequence, they are harder to predict. For example, a “Full” state can last for only 1 period, however he may be surrounded by many “Vacant” classes.

Type of Prediction	**Precision Full**	**Recall Full**	**Precision Almost Full**	**Recall Almost Full**	**Precision Vacant**	**Recall Vacant**	**Accuracy**	**Average F1-Score**
Value of the week before	1.10%	1.19%	0.32%	0.54%	79.59%	79.55%	**79.14%**	**27.02%**
Average values on the weeks before	0.43%	0.32%	0.39%	0.39%	79.59%	79.92%	**79.51%**	**26.84%**
Average values on the weeks before **(without outliers)**	0.43%	0.32%	0.39%	0.39%	79.59%	79.93%	**79.52%**	**26.84%**
Class with higher frequency on the weeks before	1.10%	1.19%	0.32%	0.54%	79.60%	79.56%	**79.15%**	**27.03%**
Class with higher frequency on the weeks before **(without outliers)**	1.10%	1.19%	0.31%	0.54%	79.60%	79.61%	**79.20%**	**27.04%**

Table 12 – Averages results of prediction based on previous weeks

The even lower precision in the class “Almost Full” is an indicator that this state seldom occurs and, consequently, is harder to predict, mostly because he may occur only for briefs moments before or after the parking “Full” status.

The results on Table 12 show that the method that averages the values and then generates the classification class obtains worst results when compared to the method of selecting the past class that most frequently occurs. The reason for this is that an average containing several weeks with low occupancy values will have a resulting low value, and by consequence will be classified as “Vacant”, resulting in even less precision.

The results based on the most frequent class in the last weeks also had poor results, however these are better than using last week’s averages.

When comparing the prediction using the previous week class with the prediction using the most frequent class in the previous weeks is visible that the behavior is almost equal, differing slightly only on the two streets that have higher occupancy values.

As previously discussed in section 4.4, there are some outliers but representing a small percentage of the data, so they may have reduced influence over it. As shown in Table 12 the results are almost equal with or without outliers, with reduced when removed. However, leaving them on the data may have the benefit of building a more robust model that reacts better to sudden changes.

The results from this prediction allow to conclude that the occupancy behavior has significant changes along the time and does not follow a precise or standard timing, resulting in most of the prediction results falling into the most common class, "Vacant".

The low precision of all methods indicates that predicting the parking occupancy based strictly on past values provides inaccurate results. It is also shown that the results are similar even when using the simplest method that uses only the previous week value, since more complicated methods only show a residual gain.

5. Prediction Models Development

This chapter presents the final steps in the preparation of the dataset, by enriching it with contextual attributes and performing the class balance. After this, it is analyzed how the prediction models are generated, how the algorithms are used, trained and tested.

5.1 Dataset Attributes Enrichment

Since the dataset consists of historical data, it is possible to enrich it with more attributes in order to understand certain data behaviors, like the change in occupancy according to the weather conditions or due to special events. It is expected that these attributes, presented in Table 13, increase the accuracy of the models because of the extra correlation that can be established between them.

Dataset Contextual Attribute	Description
Hour Class	Nominal class assigned to different periods of the day according to the occupancy pattern
Weather Conditions	Nominal class that describes the weather conditions at each moment
Temperature	Nominal class based on numerical temperature that describes the temperature sensation
Precipitation	Nominal class based on amount of precipitation that describes how heavy is the rain
Holidays	Binary class that indicates if a day is holiday or not.
Vacations	Binary class that indicates if a day is in a period where it is common to be considered for vacations
Week Number in Month	Nominal class indicating the number of the week in the month
Begin Month	Binary class that assigns if a day belongs to the begin of the month
End Month	Binary class that assigns if a day belongs to the end of the month
Special Events	Binary class that assigns if any special events happened during that day
Outlier	Binary class indicating that a day is considered an outlier

Table 13 – Contextual attributes added to the dataset

The attribute “Hour Class” assigns a classification to a specific group of hours in the day, according to the dataset analysis made previously in section 4.3.1, where the following classes are assigned: Pre-Paid (07h30-08h59), Morning 1 (09h00 - 10h59), Morning 2

(11h00 - 12h59), Lunch (13h00 - 14h29), Afternoon 1 (14h30 - 16h29), Afternoon 2 (16h30 - 17h59) and Afternoon 3 (18h00 - 19h00).

The Attribute “Weather Conditions” is a nominal attribute that describes the weather at each specific moment. The objective is to capture if sunny or rainy weather affects the parking demand and if this helps to improve the precision. “Temperature” and “Precipitation” are two additional nominal attributes that describe the temperature sensation and how heavy is the rain at a specific moment. They are added to give more context to the attribute “Weather Conditions”. All the weather data was obtained from the internet (TuTiempo.net, 2017).

The attribute “Holidays” and “Vacations” are binary attributes assigned to every moment of a day that belongs to a holiday or to a day that is considered to be in a typical period of vacations. For example, the Christmas is typically a period where many people have vacations, being the schools are also closed. This attribute is present to try to capture how these events have an impact on the parking occupancy.

“Week Number in Month” is a numerical attribute that indicates the number of the week in the month where a day belongs. As previously discussed in section 4.5, the occupation behavior has significant changes through the weeks, so this attribute tries to establish a relationship between the occupations in the different weeks of the month.

"Begin Month" and "End Month" are two binary attributes that are assigned to the first and the last 7 days of the month. The analyses in section 4.5 show that significant variations occur in the weeks at the beginning and end of the month, being this most visible in zone 1. The objective is try to establish a relation between the occupancy values and the available income, given that it is believed that in the beginning of the month the drivers may be more prone to spend money on fuel and parking, while at the end of the month they may prefer to use public transportation, by consequence reducing the parking demand.

The parking demand may be affected due to contextual events, like strikes or cultural events. "Special Events" is a binary attribute assigned to the days where is known that an event occurred, trying this way to relate this characteristic to the occupancy. This information was gathered from online news for the period 01-10-2015 through 31-12-2015.

The attribute “Outlier” is a binary attribute that indicates if a day is an outlier, according to the analyses in section 4.4. The objective is to verify how the presence of outliers affects the precision of the model and his generalization capabilities.

5.2 Occupation Classes Balance

When developing a prediction model using classifiers it is important to verify the balance of the classes, because this type of algorithms work better when the number of instances in the classes are equal. If the classes are imbalanced, the classifier will tend to classify everything as the larger class and to ignore the smaller classes, resulting in poor classification. In these situations, if the model is evaluated using only the accuracy the results will appear to be good, since almost everything is classified as the most common class, however the precision of the other classes is almost none. In this case, the use of a measurement like the F1-Score will reveal that the classifier had poor performance, as previously discussed in section 4.7.3. (Longadge, Dongre, & Malik, 2013).

The class balance can be achieved by two ways: Oversampling and Under-sampling. Under-sampling consists of randomly remove samples from the class with more instances. While effective, this technique implies the loss of information, and may not be suitable for a small dataset. Oversampling consists in duplicating the data in the minority classes, however this method may create overfit affecting the models' performance. Another consequence of oversampling is that a bigger dataset will require more computational time. Instead of duplicating data, the oversampling can be done by generating synthetic samples using Synthetic Minority Over-sampling Technique (SMOTE). This algorithm uses a distance measure to selects existent instances and randomly generates attribute values with a certain difference to the neighbor instances (Wallace, Small, Brodley, & Trikalinos, 2011) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

In section 4.7, it is stated that the accuracy results in the confusion matrix are misleading, because the class imbalance will result in many instances predicted as "Vacant" but few as "Full" or "Almost Full". This is confirmed by the reduced F1-Score obtained by the different predictions.

To verify the number of instances classified in each class the dataset was tested with different rotation percentages assigned for each class, and also different maximum occupancy threshold percentages for the start of the high demand period, as shown in Table 14.

High Demand Period			Full			Almost Full			Vacant	
% Tolerance	Number of Instances	% Classified	% Rotation	Classified Instances	% Classified	% Rotation	Classified Instances	% Classified	Classified Instances	% Classified
10%	3653	3.00%	**10%**	539	0.44%	**20%**	1101	0.90%	120328	98.66%
10%	3653	3.00%	**10%**	539	0.44%	**15%**	557	0.46%	120872	99.10%
10%	3653	3.00%	**15%**	1096	0.90%	**20%**	544	0.45%	120328	98.66%
15%	4260	3.49%	**10%**	954	0.78%	**20%**	1946	1.60%	119068	97.62%
15%	4260	3.49%	**10%**	954	0.78%	**15%**	1012	0.83%	120002	98.39%
15%	4260	3.49%	**15%**	1966	1.61%	**20%**	934	0.77%	119068	97.62%

Table 14 – Number instances in each class according to percentage and threshold

The results show that only a maximum of 3.5% of the instances will be considered inside the high demand period, automatically leaving 96.5% of the instances as “Vacant” and creating a difference between the instances in each class.

Varying between 10% or 15% the threshold for high demand period does not have a significant impact on the number of instances classified has been inside this period. However, the threshold of 15% almost doubled the number of instances classified as “Full” and “Almost Full”. This demonstrates the sensitivity of this parameter and how its tuning affects the results.

In the class “Full” changing the rotation value from 10% to 15% doubles the number of instances classified. This demonstrates that the occupancy reaches his peak between these two values, and is an indicator of how small changes in this parameter may affect the precision of the model.

In the tested values, a maximum of 1.6% of the instances is classified as “Full”, indicating that the use of low rotation values may not provide enough instances to build de prediction model accurately.

The class “Almost Full” shows that the number of classified instances increases as the rotation values decrease. As an example, with a high demand period threshold of 10% the number of instances classified between 0 and 10% (class “Full”) is almost the same as between 10% and 15% (class “Almost Full”). This is expected since the parking spaces should be “Almost Full” before reaching “Full”, however sometimes “Full” is not reached and the park will remain in “Almost Full” or “Vacant”.

Table 15 shows the balance between “Full” and “Almost Full” according to the different tests, and in column “Dataset Difference Between Classes” is visible that most of the tests performed have a class imbalance, with the exception of tests with class “Full” at 10% and “Almost Full” at 15%.

High Demand Period % Tolerance	Full % Rotation	Almost Full % Rotation	Average Difference by Street Between Classes	Dataset Difference Between Classes
10%	10%	20%	-55.86%	-68.54%
10%	10%	15%	9.56%	-3.28%
10%	15%	20%	71.69%	67.32%
15%	10%	20%	-45.94%	-68.41%
15%	10%	15%	21.91%	-5.90%
15%	15%	20%	75.11%	71.17%

Table 15 – Balance between occupation classes

Verifying the average class balance in each street we can see that the data distribution is more balanced when using a High Demand Period Tolerance of 10% in conjunction with a rotation of 10% for the class “Full” and 15% for the “Almost Full”. However, with these values, a reduced number of instances is classified of each class. Setting the High Demand Period Tolerance to 15% obtains the double of the instances for each class, creating a more detailed representation of the parking behavior.

As previously stated in this section, the definition of these values is important to ensure that the generated occupation classes accurately represent the real parking occupancy. These values should be validated and tuned on the streets, what is not possible at the time of this study. Without this validation, the exploration of these values pretends to understand what influence they have on the occupancy classes and interpret how these values correctly represent the occupancy behavior, trying to obtain meaningful results that are used through the rest of the study. Analyzing these results is essential to define the following work, where the dataset is prepared to be used in the prediction models.

5.3 Final Dataset

According to the discussion in the previous chapter, to develop an accurate prediction model the dataset must have balanced classes and provide enough instances to train and test data. With the conclusions obtained from the class balance analysis, the values used in the final dataset are:

- **High Demand Period Threshold:** 15% before reaching maximum occupancy
- **Full Class:** High Demand Period + Rotation value under 10%
- **Almost Full Class:** High Demand Period + Rotation value under 15%
- **Vacant Class:** Remaining situations.

The usage of these values provides a good balance between the "Full" and "Almost Full" classes, however the "Vacant" class is still imbalanced. This is solved by making an undersampling of the "Vacant" class using selective instance removal. This selective removal method has the advantage of still keep the "Vacant" class distribution through the dataset, while a random removal of instances may create imbalances, originating days without any "Vacant" class.

The removal process, represented in Figure 23, consists in verify for each day how many "Full" and "Almost Full" instances exist, then randomly remove "Vacant" instances from that day until the number of "Vacant" instances is equal to the sum of instances in the other classes. This is defined as an undersample by 100%, since the number of "Vacant" instances will be equal to the sum of "Full" and "Almost Full". If a day has only "Vacant" classes, only one "Vacant" instance will be kept, so that data is still available for that day. This procedure is repeated for every street.

```
for each day in dataset

    numberOfNonVacants = 0
    numberOfVacants = 0

    for each time-unit in day
        if time-unit != "Vacant"
            numberOfNonVacants = numberOfNonVacants + 1
        else
            numberOfVacants = numberOfVacants + 1
        end if
    end for

    while ((numberOfNonVacants >= numberOfVacants) || (numberOfVacants > 1))
        readedTimeUnit = readRandomTimeUnit(beginDay,endDay)

        if readedTimeUnit == "Vacant"
            delete readedTimeUnit
            numberOfVacants = numberOfVacants - 1
        end if
    do

end for
```

Figure 23 – Pseudo-code with the class balance removal method

Table 16 shows that under sampling the dataset by 100% creates more balance between classes, however this method still creates a high number of "Vacant" instances when compared to the other classes.

To test if this imbalance affects the performance, another dataset is created using 40% undersampling, reducing to 40% the number of "Vacant" instances considered for each day.

In this dataset the number of “Vacant” instances is still higher than in the other classes, but this is because one “Vacant” instance is always inserted even if the day has no other type of instances.

The class balance may have been improved by using a lower undersample percentage, however that will remove too many “Vacant” instances from each day and will modify the occupation pattern.

To verify how the model accuracy changes when all classes are balanced, two more datasets are created using SMOTE to generate instances that oversample by 200% the “Full” and “Almost Full” classes. This is applied to both previously created datasets, generating the new SMOTE datasets as shown in Table 16.

FULL DATASET - 3 CLASSES	Full Instances	Almost Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	954	1012	3268	5234	75.63%
Undersample 40% Vacant	954	1012	2111	4077	47.97%
SMOTE Oversample - 100% Vacant	1908	2024	3268	7200	31.41%
SMOTE Oversample - 40% Vacant	1908	2024	2111	6043	5.06%

Table 16 – Final dataset instances distribution using 3 classes

As stated previously in 4.3, the occupancy levels and registries have variations between zones and between streets. A zone or a street with higher occupancy levels may reveal a more predictable behavior, while the opposite may have data with more noise.

To verify this, using the previously created datasets a new dataset is created containing only zone 1 and another one containing zone 16. The same is performed for the streets, creating a dataset containing only the streets with higher occupancy, and another subset with the remaining of the streets. The instances resulting in each dataset are detailed in Appendix O.

The usage of 3 classes may result in low accuracy in one of the classes or an increased time in the creation of the model. To verify how these factors change, two more datasets are created where all the “Almost Full” instances are converted to “Full”, as shown in Table 17.

FULL DATASET - 2 CLASSES	Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	1966	3268	5234	35.18%
Undersample 40% Vacant	1966	2111	4077	5.03%

Table 17 – Final dataset instances distribution using 2 classes

5.4 Prediction Models Testing

This section discusses how the different algorithm parameters can be tuned to increase their precision, and the type of dataset split used to train and test their results.

5.4.1 Algorithm Modelling

Using the algorithms previously described in section 3.5, the initial objective is to verify the performance of different classifiers and compare them with performance obtained by the MLP, using in all algorithms the default parameters set in WEKA.

These initial tests are used as a baseline to verify the accuracy and precision of the algorithms, also allowing to verify which datasets provide better results. After this analysis, the parameters of each algorithm are modified to verify how their performance changes, training and testing them with the datasets that provided the most accurate results.

In J48, the parameter confidence level affects how the pruning of the decision tree is made and is used to prevent overfitting, so different experiments are made changing this value between 0.1 and 0.5 to increase the model performance.

In Random Forest, the size of each tree is set to unlimited since it provides the best results and is not computationally intensive with this dataset. The number of trees is by default set to 100, and different tests are performed changing this parameter since it affects the accuracy of the model and the computational time.

The REPTree algorithm also has the size of the tree set to unlimited for the same reasons as the RandomForest. Different experiments are made changing the minimum of instances in a leaf since it affects the size of the tree and the algorithm performance.

The MLP models have 16 input nodes, each one corresponding to an attribute in the dataset, and one output node for each classification class, creating a total of 3 output nodes. These models are developed with a variable number of hidden layers, that can be the number of attributes (16), of classes (3), the sum of attributes and classes (19), and the sum of attributes and classes divided by 2 (10). The learning rate influences the speed and capacity of learning in the MLP, so experiments are made by changing this value between 0.1 and 0.5.

MLP and Random Forest algorithms use random values to generate starting values, which may originate variations in the results. Because of this, in each experience the result of these algorithms is an average of 10 experiences with the same parameters but using a different seeds.

The evaluation of all the models is made using the confusion matrix as previously described in section 3.3.The evaluation of all the models is made using the confusion matrix as previously described in section 3.3.

5.4.2 Training and Validation Data

To establish a baseline for the algorithms' performance and the different datasets described in section 5.3, all of them are used to train and test the models using the two methods described in Table 18. For these methods, the dataset is randomly split without any specific instances going to the train or the test dataset. This way is possible to verify how the algorithm performs without having the train or test dataset influenced by occupancy patterns change through the days since a mix of those patterns is used to train and to test.

Training the model with different percentages of the dataset allows to verify how the number of instances in the train affect the model precision. With a small number of instances, the model may not correctly capture the occupation patterns, while with a higher number of instances the model may become overfit.

Description	Training Set	Testing Set
65% Split	65% of the data	35% of the data
80% Split	80% of the data	20% of the data

Table 18 – Training and validation data random split

After the analysis of the results of the previous experiences, the datasets that performed better are split into training and test sets according to the timeframes described in Table 19. The objective is to analyze how accurate the prediction is using a limited amount of historical data and if these smaller datasets capture the pattern changes that occur between weeks.

Description	Training Set	Testing Set
2 Weeks Test	All the data except 01-11-2015 to 15-11-2015	Data from 01-11-2015 to 15-11-2015
Monthly Test Prediction	Weeks 1,2,3 of each month	Weeks 4 of each month
Daily Test Prediction	Week 2 of each month	Monday of week 3 in each month

Table 19 – Training and validation data specific split

A first test is elaborated using all the data with the exception of the first two weeks of November to train the models, and then use those two weeks to test that model. These two specific weeks are chosen because they have a more constant behavior when compared to others. As discussed on section 4.5, the weeks occupancy has significant variations through the dataset, so it will not be effective to use weeks from September to November to train and the weeks of December to test, since their variation affects the models' performance deeply.

A second test is performed for each month, where the week 1, 2 and 3 are used to train, and then week 4 is used to test. Because the occupation patterns are more consistent by month, this test allows to verify how the models behave when trained with only 3 weeks of data.

A final test is performed by month for short-term prediction, using 5 days to train the model and then predict the occupancy for the next day. It uses week 2 to train and the Monday of week 3 to test. This week and day are specifically chosen because they are in the middle of the month, where the occupancy pattern is more constant, which may result in better model accuracy.

Table 20 contains a resume of the different dataset contents and data splits used to develop the prediction models.

Dataset Contents Description	Training Set	Testing Set
All Streets	65%	35%
	80%	20%
By Zone (Zone 1/Zone 16)	65%	35%
	80%	20%
By Occupancy (Most Occupied/Less Occupied)	65%	35%
	80%	20%
2 Weeks Test Prediction	Full dataset less 01-11-2015 to 15-11-2015	01-11-2015 to 15-11-2015
Week Test Prediction	Weeks 1,2,3 October	Week 4 October
	Weeks 1,2,3 November	Week 4 November
	Weeks 1,2,3 December	Week 4 December
Day Test Prediction	5 days of 2nd Week October	Monday of 3rd week October
	5 days of 2nd Week November	Monday of 3rd week November
	5 days of 2nd Week December	Monday of 3rd week December

Table 20 – Training datasets contents and validation splits

6. Results Analysis

This chapter presents the prediction results according to the different experiments. After each experience, the results are compared and analyzed to then refine the dataset and algorithms, allowing the realization of optimized experiences as shown at the end of the chapter.

The tables presented in this chapter show the best 5 results of the experiment being analyzed, ordered by the number of correct instances with the results having the following color code:

- 🟥 Results smaller than 50%, not having enough precision to be considered
- 🟨 Results between 50% and 65%, indicating small precision
- 🟩 Results higher than 65%, indicating a significant precision level

6.1 Algorithm Results using Default Values

To compare and discuss how the different algorithms perform, several models are developed using the different dataset splits. All the models use the default parameters set by WEKA, allowing to establish a baseline for the performance of the algorithms.

The results of all the experiments are shown in Appendix P.

6.1.1 Models with All Streets

The models in this section are developed using the datasets containing all streets. The objective is to verify the performance of these generic models and compare them to the models using specific splits of the datasets, as will be studied in the next sections.

Table 21 shows the best results while predicting for 3 occupancy classes.

DATASET (3 Classes)	ALGORITHM	TRAIN	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-score
			Precision	Recall	Precision	Recall	Precision	Recall		
SMOTE Oversample - 100% Vacant	RandomForest	65%	62.80%	66.80%	66.00%	65.60%	81.10%	78.50%	71.83%	71.90%
SMOTE Oversample - 100% Vacant	RandomForest	80%	61.00%	67.30%	65.40%	65.80%	82.50%	77.60%	71.60%	71.80%
SMOTE Oversample - 40% Vacant	RandomForest	65%	66.60%	71.00%	68.60%	69.20%	78.00%	72.80%	71.02%	71.10%
SMOTE Oversample - 40% Vacant	RandomForest	80%	65.70%	71.80%	68.50%	68.70%	78.20%	71.50%	70.64%	70.70%
SMOTE Oversample - 40% Vacant	MLP	65%	66.00%	68.90%	64.80%	69.30%	73.80%	65.90%	67.99%	68.00%

Table 21 – Results using default parameters, for all streets with 3 classes

The models developed with RandomForest offered the best results independently of the dataset used. The best RandomForest has 71.83% correct instances and an F1-Score of 71.90%, which confirms that the model has a reasonable precision across all classes. However, the precision in the "Full” class is only 62.80%, which is considerably less than the precision of 81.10% in the “Vacant” class.

The MLP also offered a reasonable performance, having only 4% less correct instances than the best RandomForest, but still performing better than J48 or REPTree. Despite this, MLP obtained the highest precision for the “Full” class, however it performed slightly worst in the other classes, having a lower F1-Score than the RandomForest. This indicates that with parameter tuning, the MLP performance may be equal or better than the RandomForest.

The Dataset “SMOTE Oversample - 100% Vacant” offered the best results. However, this dataset contains a high number of “Vacant” instances, which when combined with the good precision for the “Vacant” class result in a high number of correct instances, masking the less good precision in the other classes.

When analyzing the results is visible that the best precision results for the “Full” and “Almost Full” classes are when using the dataset “SMOTE Oversample - 40% Vacant”, having a slight loss of precision for the “Vacant” class. The reason for this is that the dataset used in this model has a higher balance between classes, as shown in chapter 5.3, enabling the algorithm to have better overall performance.

The best results for each dataset are achieved when the models are trained with 65% of the data, indicating that this percentage offers a higher capacity of generalization in the model, being able to adapt better to different conditions. The results when the model is trained with 80% of the data show that the model becomes overfit, responding worse to the variations present in this dataset, as previously discussed in chapter 4.3.

Table 22 shows the prediction results using the same criteria as before but for the prediction of only 2 classes.

DATASET (2 Classes)	ALGORITHM	TRAIN	FULL		VACANT		Correct Instances	Weighted Average F1-score
			Precision	Precision	Precision	Recall		
Undersample 100% Vacant	RandomForest	80%	67.30%	66.90%	79.90%	80.20%	75.17%	75.20%
Undersample 100% Vacant	RandomForest	65%	66.90%	64.40%	78.20%	80.00%	74.02%	73.90%
Undersample 40% Vacant	RandomForest	80%	68.10%	80.80%	79.90%	66.90%	73.37%	73.30%
Undersample 40% Vacant	RandomForest	65%	70.50%	76.50%	76.10%	70.00%	73.16%	73.20%
Undersample 100% Vacant	MLP	80%	61.00%	62.40%	76.80%	75.70%	70.68%	70.70%

Table 22 – Results using default parameters, for all streets with 2 classes

In the 2 class prediction, the performance of the algorithms is similar to the ones with 3 classes, having RandomForest the best performance followed by the MLP. The main difference is that the precision values are higher when using only 2 classes, demonstrating that the algorithms perform better when using a higher number of instances in each class and simpler decision process. However, the MLP showed a significant loss of precision in the "Full" class when compared to the RandomForest.

The dataset "Undersample 100% Vacant" offered the higher number of correct instances in the 2 class prediction, however the precision between classes is more balanced when using "Undersample 40% Vacant". These results are similar to the ones in 3 class prediction.

The main difference is that the better results are obtained when training the models with 80% of the data, indicating that the variations in the dataset are smaller when using the higher occupancy tolerances set by the 2 classes.

6.1.2 Models by Zone

These models are trained using datasets that contain the occupation data from only one zone, with the objective of verifying how much this split reduces the confusion in the dataset and improves the results.

The result of the models for 3 classes are shown in Table 23, and they demonstrate that the best result is achieved when using data from only zone 1. When compared with the results in 6.1.1, these have an improvement of almost 8% precision in the "Full" and "Almost Vacant" classes, and of 2% in the number of correct instances.

The results are worse for zone 16, indicating that the higher number of instances present in zone 1 allows a better training of the algorithms and representation of the occupation patterns.

DATASET (3 Classes)	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-score
				Precision	Recall	Precision	Recall	Precision	Recall		
SMOTE Oversample - 40% Vacant	ZONE 1	RandomForest	80%	71.40%	72.80%	73.20%	74.60%	76.50%	73.20%	73.64%	73.60%
SMOTE Oversample - 100% Vacant	ZONE 1	RandomForest	80%	62.90%	69.10%	69.90%	67.90%	80.50%	77.90%	72.37%	72.50%
SMOTE Oversample - 100% Vacant	ZONE 16	RandomForest	80%	67.70%	64.30%	52.00%	62.90%	84.20%	79.90%	71.72%	72.10%
SMOTE Oversample - 100% Vacant	ZONE 16	RandomForest	65%	68.00%	66.40%	54.10%	58.90%	81.10%	79.30%	71.21%	71.40%
SMOTE Oversample - 40% Vacant	ZONE 1	J48	80%	64.10%	68.70%	74.00%	78.50%	74.70%	63.40%	71.13%	71.10%

Table 23 – Results using default parameters, training by zone with 3 classes

This experience also shows that the performance of the algorithms improved when training specific models for each zone since the noise and confusion of the dataset get reduced. This happens because each zone has his own occupancy pattern as discussed in section 4.3.1.

RandomForest is the algorithm that offered better results and “SMOTE Oversample - 40% Vacant” is the dataset that performed better, being these the same that also had best results in the generic model for all streets with the reasons for this already discussed in section 6.1.1. J48 is the second best algorithm, probably because the smaller confusion in this dataset allows it to perform better and get slightly better results than the MLP.

The results for the 2 class prediction are shown in Table 24 and they also demonstrate an improvement in results when compared to the more generic model developed in 6.1.1. They also have the best results when using the same algorithms and datasets as in the generic model.

DATASET (2 Classes)	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		VACANT		Correct Instances	Weighted Average F1-score
				Precision	Precision	Precision	Recall		
Undersample 100% Vacant	ZONE 1	RandomForest	80%	67.30%	74.40%	84.90%	79.90%	77.92%	78.10%
Undersample 100% Vacant	ZONE 16	RandomForest	80%	64.00%	67.10%	82.50%	80.50%	75.92%	76.00%
Undersample 100% Vacant	ZONE 1	RandomForest	65%	67.30%	69.20%	79.90%	78.50%	74.85%	74.90%
Undersample 40% Vacant	ZONE 16	RandomForest	65%	70.90%	77.20%	78.30%	72.20%	74.56%	74.60%
Undersample 40% Vacant	ZONE 1	RandomForest	80%	68.60%	82.20%	80.70%	66.50%	73.89%	73.80%

Table 24 – Results using default parameters, training by zone with 2 classes

For the 2 class prediction, the models is also more effective when using data from only zone 1, has been previously stated for the 3 class prediction, having the same reasons to justify the performance of this dataset split. In both predictions, the better result are achieved when training the models with 80% of the data. Since both of these datasets are smaller, this may indicate that a higher number of instances is required to capture the occupation pattern correctly.

However, the “Full” class has the highest precision when the model is trained with 65% of the data, having the “Vacant” class a prediction loss of 5%. Since the “Full” class is more difficult to predict, these results indicate that the model may be less overfit when trained with only 65% of the data.

6.1.3 Models by Street Occupancy

To observe how the levels of occupancy and the number of instances in the streets affect the results of the prediction, these models are trained using one dataset containing only the streets with higher occupancy and another one containing the remaining streets, as previously explained in 5.3.

DATASET (3 Classes)	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-score
				Precision	Recall	Precision	Recall	Precision	Recall		
SMOTE Oversample - 100% Vacant	MOST Occupied	RandomForest	80%	68.30%	65.90%	69.30%	71.80%	80.50%	79.50%	73.51%	73.50%
SMOTE Oversample - 100% Vacant	LESS Occupied	RandomForest	80%	68.00%	68.90%	56.90%	55.50%	82.00%	82.00%	72.99%	73.00%
SMOTE Oversample - 40% Vacant	LESS Occupied	RandomForest	80%	66.80%	70.90%	62.80%	58.70%	80.30%	79.30%	71.45%	71.40%
SMOTE Oversample - 100% Vacant	MOST Occupied	J48	80%	65.20%	61.80%	66.40%	76.80%	79.00%	69.90%	70.60%	70.60%
SMOTE Oversample - 40% Vacant	MOST Occupied	J48	80%	63.70%	65.10%	71.20%	79.80%	76.50%	61.90%	70.36%	70.20%

Table 25 – Results using default parameters, training by occupancy with 3 classes

Table 25 shows the results for the 3 class prediction demonstrating that both datasets had similar levels of precision and offered better results than the generic model developed in 6.1.1, but are worse than the model with the dataset split by zone presented 6.1.2.

Despite the similar level of precision, the dataset with “Less Occupied Streets” had worst precision in the “Almost Full” class, indicating that this dataset may have more noise in the data because of the higher number of streets included and the reduced number of instances in each one.

The algorithm, dataset and training split that offered better performance are the same as stated in the previous models, being this discussed in 6.1.1 and 6.1.2.

The results for the 2 class prediction are shown in Table 26, and despite the good results, the overall performance is worse than the obtained for the 2 class prediction with the generic models in 6.1.1 and with models split by zone in 6.1.2.

DATASET (2 Classes)	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		VACANT		Correct Instances	Weighted Average F1-score
				Precision	Recall	Precision	Recall		
Undersample 40% Vacant	LESS Occupied	RandomForest	80%	74.80%	74.40%	76.80%	77.10%	75.82%	75.80%
Undersample 100% Vacant	LESS Occupied	RandomForest	80%	65.50%	63.00%	80.90%	82.50%	75.80%	75.70%
Undersample 100% Vacant	MOST Occupied	RandomForest	80%	69.30%	70.70%	79.80%	78.80%	75.49%	75.50%
Undersample 40% Vacant	LESS Occupied	RandomForest	65%	72.00%	71.80%	75.90%	76.10%	74.13%	74.10%
Undersample 40% Vacant	MOST Occupied	J48	80%	76.00%	77.10%	71.50%	70.30%	74.04%	74.00%

Table 26 – Results using default parameters, training by occupancy with 2 classes

This strict split of the dataset by street occupancy appears to remove generalization capability of the models for 3 and 2 class prediction, since in each dataset all the streets have a similar occupation behavior which may result in an overfitted model.

6.1.4 Models with 2 Week Split

In section 4.5 was verified that the occupation suffers strong variations during the weeks, so the objective is to verify if it is possible to build a model that predicts the occupation of 2 specific weeks using the rest of the data to train the model, as explained in 5.3.

Table 27 shows the result of the models for this type of prediction. The results are poor, with the "Full" and "Almost Full" classes having low precision. The number of correct instances is reasonable, but it is mostly because of the higher precision in the "Vacant" class, being this confirmed by the low average F1-Score obtained by these models.

DATASET (3 Classes)	ALGORITHM	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-score
		Precision	Recall	Precision	Recall	Precision	Recall		
Undersample 100% Vacant	J48	29.40%	10.00%	46.80%	33.60%	67.60%	87.20%	67.60%	57.50%
Undersample 100% Vacant	RandomForest	33.30%	5.30%	46.70%	9.20%	63.30%	95.30%	63.30%	51.80%
Undersample 100% Vacant	REPTree	30.40%	14.00%	48.60%	23.00%	66.60%	88.40%	66.60%	56.60%
Undersample 100% Vacant	MLP	34.40%	7.30%	28.80%	30.30%	64.60%	79.00%	64.60%	51.90%
Undersample 40% Vacant	RandomForest	21.10%	2.70%	53.10%	22.40%	55.60%	93.50%	55.60%	44.70%

Table 27 – Results using default parameters, testing for 2 weeks with 3 classes

The dataset and algorithms with better results are also different from the ones in the previous models, with RandomForest and MLP having difficulties when building the models to this specific prediction.

In Table 28 are displayed the results for the 2 class prediction, and they are slightly better than the prediction for 3 classes. However, the low F1-Score of the models still indicates that their performance is reduced, confirming that the variations through the weeks presented in this dataset difficult the models training.

DATASET (2 Classes)	ALGORITHM	FULL		VACANT		Correct Instances	Weighted Average F1-score
		Precision	Recall	Precision	Recall		
Undersample 100% Vacant	REPTree	53.50%	43.40%	68.80%	76.80%	64.06%	47.90%
Undersample 100% Vacant	J48	53.20%	38.70%	67.70%	79.00%	63.68%	44.80%
Undersample 100% Vacant	RandomForest	51.00%	16.60%	63.70%	90.20%	62.17%	25.00%
Undersample 40% Vacant	RandomForest	66.50%	38.10%	58.40%	81.90%	60.67%	48.40%
Undersample 100% Vacant	MLP	46.90%	27.20%	64.40%	81.10%	60.53%	34.40%
Undersample 40% Vacant	J48	60.90%	51.00%	60.00%	69.20%	60.35%	55.50%

Table 28 – Results using default parameters, testing for 2 weeks with 2 classes

The obtained results confirm that this type of prediction is difficult to make using this dataset because the occupation values suffer significant variations from week to week, being this most noticeable in zone 1 as previously discussed in section 4.5. In the previous experiences, the models performed better because they are trying to predict a random sample of the occupation, where all type of values is present. In this experiment, the models are trying to predict the behavior of a specific week, being this more difficult especially if the dataset does not follow any specific occupation pattern.

6.1.5 Algorithm and Dataset Performance Conclusions

To select which algorithms and datasets performed better, this section resumes their overall performance in the previous tests.

The tables present the different algorithms and datasets, with the first column indicating the number of experiences where they had an F1-Score higher than 65%, being this considered a good result. The next column indicates the average F1-Score of the experiences that had this value higher than 65%. The last column indicates the average algorithm F1-Score of all experiences.

The algorithms performance using 3 classes are shown in Table 29 and, as verified during the experiments, RandomForest is the algorithm that most frequently provided good results and with a higher average F1-Score. MLP and J48 presented similar average F1-Score results, however MLP had an F1-Score above 65% on more tests. REPTree had the lowest classification despite having only a small different in the average F1-Score however, was the one with least frequent good results.

Algorithm Name (3 Classes)	**Number of Models With F1-Score > 65%**	**Average F1-Score**	
		Results > 65%	All Results
RandomForest	25	70.32%	63.85%
MLP	20	68.45%	60.63%
J48	15	68.19%	59.97%
REPTree	6	66.18%	58.19%

Table 29 – Algorithm performance results using default parameters with 3 classes

The same analysis is performed in Table 30 for the algorithms prediction performance using 2 classes, being the results similar to the previous analysis. All the algorithms have slightly better average results than in the prediction for 3 classes, with RandomForest providing the best results. MLP and J48 have almost equal results, with REPTree being the worst but performing better than in the 3 classes prediction. This demonstrates that the dataset confusion diminishes when using only 2 classes, with the number of instances becoming more balanced between classes, creating a dataset where is easier to understand the occupation patterns and simplifying the decision process.

Algorithm Name (2 Classes)	**Number of Models With F1-Score > 65%**	**Average F1-Score**	
		Results > 65%	All Results
RandomForest	22	73.67%	70.59%
MLP	20	69.72%	66.50%
J48	20	69.09%	67.07%
REPTree	16	68.06%	65.45%

Table 30 – Algorithm performance results using default parameters with 2 classes

These results show that RandomForest is the most effective. However, the small differences obtained in these comparisons demonstrate that MLP and J48 also have a similar performance using this dataset. Despite this, during the experiments was possible to verify that the RandomForest always performed more consistently, being the first or second more precise, demonstrating his flexibility and capacity of generalization. J48 had oscillations in his performance, indicating that he is sensitive to the content of the training, having more difficulties as the data gets more noise. MLP offered good results, however the training time is much longer than with the RandomForest, incrementing exponentially as the dataset size increases.

Analyzing the different datasets, Table 31 shows their overall performance of through the different experiences when used for the 3 class prediction.

Dataset Name (3 Classes)	**Number of Models With F1-Score > 65%**	**Average F1-Score**	
		Results > 65%	All Results
SMOTE Oversample - 100% Vacant	33	69.03%	64.37%
SMOTE Oversample - 40% Vacant	28	69.15%	63.90%
Undersample 100% Vacant	5	66.52%	59.21%
Undersample 40% Vacant	0	0.00%	55.16%

Table 31 – Datasets performance results using 3 classes

The results show that the best results are achieved in both datasets oversampled with SMOTE, with them having the highest number of experiences with an F1-Score above 65%. “Undersample 100% Vacant” only had an F1-Score higher than 65% in 5 models, and

"Undersample 40% Vacant" had 0. This shows that the performance of the models increases when datasets have an evener class balance, as is the case in the oversamples datasets.

The number of instances in the dataset is important but not deterministic since "SMOTE Oversample - 40% Vacant" has a lower number of instances but a slightly higher average F1-Score despite not having the higher number of models with an F1-Score higher than 65%.

Table 32 shows the performance of datasets used in 2 class prediction with both of them having similar results. These are slightly better for the dataset "Undersample 40% Vacant", that once again have a smaller the number of instances and the classes more balanced, as discussed previously and in section 5.3.

Dataset Name (2 Classes)	**Number of Models With F1-Score > 65%**	**Average F1-Score**	
		Results > 65%	All Results
Undersample 40% Vacant	39	70.72%	68.24%
Undersample 100% Vacant	39	69.94%	66.56%

Table 32 – Dataset performance results using 2 classes

According to these results, the optimization of the models in the next experimentations are developed only for the algorithms that offered better performance, being these the RandomForest and MLP. J48 did not offer such consistent results, however it is also included in the optimization process because of his fast execution time and to allow a performance improvement comparison.

The datasets used for these optimizations in the 3 class prediction are the "SMOTE Oversample - 40% Vacant" and the "SMOTE Oversample - 100% Vacant" since they offered the best and more consistent results. In the 2 class prediction, both datasets are used since they offered a similar performance.

6.2 Attributes Relevance in Results

The attributes added to the dataset in section 5.1 are intended to give the data more comprehension and to allow the classifiers to perform better since they have extra attributes to correlate with the occupation patterns. Figure 24 is a sample of the decision tree built by J48, and it demonstrates how the attributes influence the decision process. This decision tree is expanded with more nodes in Appendix Q.

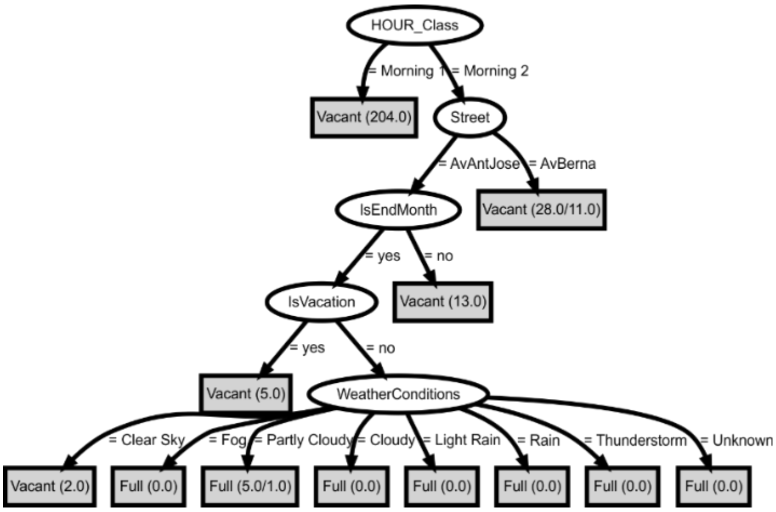


Figure 24 – Sample of a decision tree generated by J48

The addition of attributes may not always improve the results, since they may be considered nonrelevant. They can also degrade the results because of the additional complexity of the models or because they may generate overfitted models giving them low capacity of generalization.

To verify the relevance of the attributes, it is performed an evaluation using "Ranking", which is a "Filter" technique where the attributes are individually evaluated with a statistical method, being then a ranking applied to each one. This method has the advantage of being fast to execute and to allow the detection of irrelevant attributes (Hall & Holmes, 2003).

Table 33 shows the results of the ranking evaluation, where the "Street", "Hour_Class" and "Zone" are the 3 most relevant attributes, with a merit value higher than the rest, being this and indicator of their relevance in the decision process.

Rank	Merit	Attribute
1	0.1517491	Street
2	0.1268967	HOUR_Class
3	0.0633586	Zone
4	0.0174306	WeatherTemperature
5	0.0151109	TimeStamp
6	0.0141778	IsHoliday
7	0.0124293	WeatherConditions
8	0.0033928	WeekNumberMonth
9	0.0030340	WeekDay
10	0.0023966	Outlier
11	0.0014437	WeatherRain
12	0.0011458	IsVacation
13	0.0004557	IsBeginMonth
14	0.0002167	ExistSpecialEvent
15	0.0000807	IsEndMonth

Table 33 – Raking and merit of the attributes in the dataset

The attributes relative to the weather have the 4th, 7th and 11th position and, as expected, are considered relevant to the decision process. From ranking 8th to 15th the merit of each attribute drops below 0.01, indicating that they have low relevance in the decision process. To verify how much the attributes have an impact on the precision of the models, each one is individually removed from the dataset and then a prediction model is developed using that dataset. The precision of this model is then compared to the model that used all attributes, represented in the tables as “NONE”.

Table 34 shows that the average F1-Score of the RandomForest model is improved by 0.40% when removed the attribute “Outlier”, and by 0.10% when “IsHoliday” is removed. All the other attributes reduce the precision of the data when removed. The attributes with the weather information proved to be the most relevant because when removed the average F1-Score reduces 1.1% and the “Full” precision reduces 2.8%, being this consistent with the classification of the weather condition attributes in the ranking.

REMOVED ATTRIBUTES (Using RandomForest Models)	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall	Precision	Recall		
Outlier	66.60%	71.00%	68.60%	69.20%	78.00%	72.80%	71.02%	71.10%
IsHoliday	66.10%	70.60%	67.90%	69.20%	78.50%	72.30%	70.69%	70.80%
NONE	66.20%	70.60%	68.20%	69.10%	77.80%	72.30%	70.64%	70.70%
WeekNumberMonth	65.90%	70.10%	68.50%	68.90%	77.30%	72.60%	70.54%	70.60%
IsBeginMonth/EndMonth	66.40%	70.40%	67.80%	68.90%	77.50%	72.10%	70.50%	70.60%
Weekday	66.00%	70.70%	68.20%	68.50%	77.40%	72.10%	70.45%	70.50%
ExistSpecialEvent	66.20%	69.90%	67.50%	69.50%	77.60%	71.50%	70.31%	70.40%
IsVacation	65.70%	70.70%	67.90%	68.50%	77.60%	71.70%	70.31%	70.40%
All Weather Attributes	63.40%	68.90%	66.80%	68.00%	78.90%	71.60%	69.50%	69.60%

Table 34 – Performance of RandomForest models by removed attribute

Table 35 shows the results using MLP, where the removal of the attributes “IsBeginMonth/EndMonth” improved the F1-Score 0.8%, also improving the precision of the “Full” class by 1.5%. The removal of “IsVacation” increases the number of correct instances by 0.10%, however the F1-Score remains equal to the value in the base prediction. The “Outlier” attribute has only 0.05% improvement in the number of correct instances. However, its F1-Score has an improvement of 0.7% and the “Full” precision improves by 4.40%, indicating that the removal of this attribute has an impact on the final results. All the other attributes improve the precision of the model, with the weather attributes being again the ones with greater impact, generating a significant improvement in the precision of “Full” and “Almost Full” classes.

REMOVED ATTRIBUTES (Using MLP Models)	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall	Precision	Recall		
IsBeginMonth/EndMonth	63.10%	72.40%	66.90%	69.30%	78.00%	65.10%	68.79%	68.90%
IsVacation	62.70%	70.40%	65.70%	70.00%	77.50%	64.00%	68.04%	68.10%
Outlier	66.00%	68.90%	64.80%	69.30%	73.80%	65.90%	67.99%	69.60%
NONE	61.60%	68.40%	66.20%	71.30%	77.80%	64.30%	67.94%	68.10%
IsHoliday	63.40%	63.20%	63.60%	74.70%	77.60%	64.40%	67.57%	67.60%
Weekday	61.80%	66.90%	66.80%	67.10%	73.20%	67.70%	67.23%	67.30%
ExistSpecialEvent	61.20%	66.10%	65.10%	69.70%	74.70%	64.10%	66.67%	66.80%
WeekNumberMonth	59.70%	72.20%	66.80%	65.50%	75.00%	62.50%	66.52%	66.60%
All Weather Attributes	55.80%	67.80%	61.80%	62.40%	74.70%	59.60%	63.12%	63.30%

Table 35 – Performance of MLP models by removed attribute

The impact of the attributes in the J48 models is shown in Table 36, where the removal of "ExistSpecialEvent" improves the F1-Score by 0.30% and the "Full" precision by 0.50%. Removing the "IsBeginMonth/EndMonth" improves the F1-Score 0.20% and the "Full" precision in 0.60%. As in RandomForest, the removal of "IsHoliday" improve the results, but in J48 this impact is smaller, with only 0.04% improvement of in the number of correct instances with the other values remaining the same as the baseline. The presence of the remaining attributes improves the performance of the model, but when compared to the other algorithms these are smaller and with less consistent improvements in the precision of the classes, showing that J48 has difficulties evaluating the attributes and by consequence obtaining worse results.

REMOVED ATTRIBUTES (Using J48 Models)	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall	Precision	Recall		
ExistSpecialEvent	61.90%	69.60%	67.00%	69.90%	73.70%	62.40%	67,19%	67,20%
IsBeginMonth/EndMonth	62.10%	68.10%	67.40%	70.40%	72.20%	62.80%	67,04%	67,10%
IsHoliday	61.40%	69.00%	66.80%	69.90%	73.50%	62.10%	66,90%	66,90%
NONE	61.40%	69.00%	66.80%	69.90%	73.40%	62.00%	66,86%	66,90%
IsVacation	60.90%	68.70%	66.80%	70.40%	74.20%	61.70%	66,86%	66,90%
Weekday	61.40%	69.00%	66.80%	69.90%	73.40%	62.00%	66,86%	66,90%
All Weather Attributes	60.50%	69.30%	66.00%	69.70%	75.70%	61.70%	66,81%	66,90%
Outlier	61.00%	68.90%	66.80%	69.90%	73.10%	61.40%	66,62%	66,60%
WeekNumberMonth	62.40%	68.10%	65.60%	70.70%	72.70%	61.10%	66,57%	66,60%

Table 36 – Performance of J48 models by removed attribute

The results of these tests demonstrate that the presence of attributes as "IsBeginMonth/EndMonth" and "Outlier" are consistently considered to reduce the performance of the data, while attributes as "All Weather Attributes" have a significant relevance in the performance of the models.

In section 5.1, it was discussed the importance of the attributes "Outlier" and "IsBeginMonth/EndMonth", however according to these results, they appear to increase the confusion in the data. It is also possible that their presence generates overfitted models, with them having difficulties to adapt to the variations in the testing data.

As discussed in section 4.4, only a small number of days are an outlier, and they are sparse. This may be the cause for the confusion they create in the data, since the model is being trained for a situation that has a low probability of occurrence.

The "IsBeginMonth/EndMonth" attribute appears to be relevant according to the occupancy patterns shown in section 4.5, but these results indicate the opposite. This may occur because of the relevance assigned to the attribute "WeekNumberMonth", that can still be used to understand where is the begin and end of a month, making the attribute "IsBeginMonth/EndMonth" redundant.

The removal of the attributes does not drastically change the final results, but it offers a small improvement that deserves to be explored in how this change precision through the different datasets.

In the next section, the datasets are tested with the removal of the following attributes:

RandomForest

- Outlier
- IsHoliday
- Outlier + IsHoliday

- MLP
  - IsBeginMonth/EndMonth
  - Outlier
  - IsBeginMonth/EndMonth + Outlier
- J48
  - ExistSpecialEvent
  - IsBeginMonth/EndMonth
  - ExistSpecialEvent + IsBeginMonth/EndMonth

6.3 Algorithm Optimized Models

This section discusses how the algorithm parameters' tuning affects their performance and changes the predictions results. The models are also tested with the optimal datasets and with the removal of irrelevant attributes. It is also presented the standard deviation of the RandomForest and MLP experiences to verify the amount of variation in their results.

6.3.1 Algorithm Parameters Performance

As previously discussed in 5.4.1, the parameters of the algorithms selected in 6.1.5 are tested using the datasets with the best performance in the models using default parameters.

Table 37 shows the results of the models for 3 class prediction after these being tested with the dataset "SMOTE Oversample - 40% Vacant" with all streets. In RandomForest the best result is obtained when the number of trees is 1000, however the gains are reduced with only an improvement of 0.10% in the F1-Score. In MLP, changing the number of hidden layers to match the number of attributes (17) and the learning rate to 0.1 resulted in an F1-Score improvement of 2.60% and an improvement of 6.6% in the precision of "Full" class, indicating that the change of parameters has a significant impact on performance. In the J48, changing the confidence level to 0.35 led to a small improvement of 0.5% in the F1-Score and of 1% in the "Full" class.

Algorithm (3 Classes)	Parameters	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
		Precision	Recall	Precision	Recall	Precision	Recall		
RandomForest	Iteration = 1000	66.20%	69.90%	68.30%	69.30%	77.90%	72.80%	70.73%	70.80%
	Iteration = 100 **(DEFAULT)**	66.20%	70.60%	68.20%	69.10%	77.80%	72.30%	70.64%	70.70%
MLP	'i'attribs+LR 0.1	68.20%	66.30%	67.20%	76.50%	77.40%	68.90%	70.69%	70.70%
	'a'((att+cls)/2)+LR 0.3 **(DFT)**	61.60%	68.40%	66.20%	71.30%	77.80%	64.30%	67.94%	68.10%
J48	Conf 0.35	62.40%	69.80%	66.80%	71.00%	74.20%	61.80%	67.42%	67.40%
	Conf 0.25 **(DEFAULT)**	61.40%	69.00%	66.80%	69.90%	73.40%	62.00%	66.86%	66.90%

Table 37 – Algorithm parameters optimization comparison - 3 classes

In these results, RandomForest demonstrates that an increase the number of trees does not translate into a higher precision, since the value of 100 offered almost the best results and a value of 1000 increases the computational time and memory usage. This also indicates that RandomForest is an optimized algorithm and has low sensitivity the input parameters.

The results for the MLP demonstrate how much a Neural Network is sensitive to the input parameters and how his performance improves when they are tuned. The optimal parameters values found can be understood as a consequence of the complexity in this dataset, since the number of hidden layers has increased and the learning rate had to be diminished to a slower rate in order to make the neural network correctly learn from the data and converge.

In the J48, the optimized parameter consisted in an increase of the confidence value, indicating once more that the dataset is complex and the pruning of the decision tree has to be done later.

Table 38 shows the results of the model parameters for the 2 class prediction using the dataset “Undersample 40% Vacant”. There are still improvements, but they are more reduced, namely in the MLP where the improvement is only 0.5% in the “Full Class”. However, MLP improved the performance by 7.9% in the “Vacant” class. The optimal parameters for MLP algorithm also changed, with a different the number of hidden layers and a different learning rate. The J48 had their optimal result when the Confidence Level is set to 0.15. These parameters demonstrate that the dataset becomes less complex when using only 2 classes, since the learning rate of the MLP increased and the confidence level of the J48 had been reduced. The RandomForest showed his robustness by keeping the same parameters and similar results as when used for 3 class prediction.

Algorithm (3 Classes)	**Parameters**	**FULL**		**VACANT**		**Correct Instances**	**Weighted Average F1-Score**
		Precision	**Recall**	**Precision**	**Recall**		
RandomForest	Iteration = 1000	67.10%	67.40%	80.10%	79.90%	75.17%	75.20%
	Iteration = 100 **(DEFAULT)**	66.90%	67.40%	80.10%	79.70%	75.07%	75.10%
MLP	't' (att+cls)+LR 0.4	65.00%	54.00%	74.70%	82.30%	71.63%	71.00%
	'a' ((att+cls)/2)+LR 0.3 **(DEFAULT)**	57.90%	57.30%	74.20%	74.70%	68.10%	68.10%
J48	Conf 0.15	63.30%	42.20%	70.80%	85.10%	68.86%	67.20%
	Conf 0.25 **(DEFAULT)**	57.50%	47.20%	71.10%	78.80%	66.86%	66.10%

Table 38 – Algorithm parameters optimization comparison - 2 classes

The tables presented in this section contains the best results of each algorithm, being in Appendix R the remaining results of each experience.

6.3.2 Algorithm Optimized Prediction Results

Using the parameters defined in 6.3.1, new prediction models are developed and then compared with ones using standard parameters to verify how the performance improves. The attributes defined in 6.2 are also removed from the datasets in some experiments to verify how the performance of the model's changes. Using these optimizations, two new short-term prediction models are created, one for 1 week ahead prediction and another one for 1 day ahead.

The presented result are for the best results of each algorithm, being the remaining experiences in Appendix S.

6.3.2.1 Models with All Streets

Table 39 and Table 40 demonstrate the best results obtained for each algorithm while predicting the occupancy for all streets using 3 and 2 classes. At the bottom of tables are the previously obtained baseline prediction values that will be used for comparison. In both situations RandomForest is still the algorithm offering better performance, with a small performance improvement over the baseline values, demonstrating that he is optimized by default.

Despite this, MLP is the algorithm that obtained the higher precision at the "Full" class, being this an indicator of algorithm ability to work with complex data. Compared to the previous results, with the optimization MLP had precision gains of 2.20% for the 3 class prediction and of and 7.6% for the 2 class prediction. With further parameter adjustment for each dataset, the gains may improve, but the time required to train and adjust the model may not justify it. When compared to RandomForest, MLP requires more training and parameter adjustment time, offering only slightly improved results in some situations. This increased MLP precision at 3 class prediction is also obtained using the dataset where none of the attributes are removed, demonstrating that with the lower learning rate the MLP can work with the increased dataset complexity and to obtain benefits from it.

In these tests J48 had the lower performance, however when compared to the other results his values do not have a significant difference, indicating again that all the algorithms have a similar performance, and that the complexity of the dataset limits their improvements.

Dataset (3 Classes)	Removed Attributes	Algorithm	Parameters	Train	FULL Precision	ALMOST FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	62.80% (+/- 0.0032%)	66.10% (+/- 0.0026%)	81.50% (+/- 0.0012%)	72.02% (+/- 0.0013%)	72.10% (+/- 0.0013%)
SMOTE Oversample - 40% Vacant	NONE	MLP	'i'attribs+LR 0.1	65%	68.20% (+/- 0.0165%)	67.20% (+/- 0.0249%)	77.40% (+/- 0.0222%)	70.69% (+/- 0.0096%)	70.70% (+/- 0.0097%)
SMOTE Oversample - 100% Vacant	IsBeginMonth EndMonth	J48	Conf 0.35	80%	60.00%	63.70%	77.00%	68.47%	68.60%
BASELINE	NONE	RandomForest	DEFAULT	65%	62.80% (+/- 0.0043%)	66.00% (+/- 0.0025%)	81.10% (+/- 0.0049%)	71.83% (+/- 0.0017%)	71.90% (+/- 0.0019%)

Table 39 – Algorithm modified parameters results for all streets - 3 classes

Dataset (2 Classes)	Removed Attributes	Algorithm	Parameters	Train	FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	67.90% (+/- 0.0012%)	80.30% (+/- 0.0019%)	75.64% (+/- 0.0014%)	75.60% (+/- 0.0011%)
Undersample 40% Vacant	IsBeginMonth /EndMonth	MLP	't' (att+cls)+LR 0.4	80%	70.60% (+/- 0.0329%)	73.00% (+/- 0.018%)	71.90% (+/- 0.0106%)	71.90% (+/- 0.0112%)
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	80%	63.70%	71.50%	69.44%	68.00%
BASELINE	NONE	RandomForest	DEFAULT	80%	67.30% (+/- 0.003%)	79.90% (+/- 0.043%)	75.17% (+/- 0.0026%)	75.20% (+/- 0.0025%)

Table 40 – Algorithm modified parameters results for all streets - 2 classes

6.3.2.2 Models by Zone

Table 41 and Table 42 present the prediction result by zone after the algorithms optimization, and it demonstrates that all of them had higher precision using the dataset split by zone, as already discussed in section 6.1.2.

RandomForest still has the highest precision, but the optimization resulted in few improvements, with slight increases in precision and F1-Score. This occurs in both 3 and 2 class prediction, leading to the conclusion that improvements obtained with the optimized parameters may not justify the increased computational time.

MLP has the highest improvements, increasing the precision of "Full" class by 4.7% in 3 class prediction and 4.1% in 2 class prediction. As in the previous predictions, this demonstrates the importance of the parameter adjustment in the MLP and the relevant gains obtained from it.

Both MLP and RandomForest obtained improved results when the attribute "Outlier" is removed, with MLP also having improvements after removing the attribute "IsBeginMonth/EndMonth", being this coherent with the discussion in section 6.2, where these attributes are considered to have few relevance. However, RandomForest has the best result for the 2 class prediction and using a dataset with all attributes, indicating that due to the low merit of all attributes sometimes their presence does not significantly increase the confusion, with the algorithms benefitting for their inclusion in the dataset. This is also valid for the J48 algorithm, which had is highest precision when all the dataset attributes are present, despite the fact that the parameter optimizations did not offer any improvement in his performance.

Dataset (3 Classes)	**Removed Attributes**	**Algorithm**	**Parameters**	**Train**	**FULL Precision**	**ALMOST FULL Precision**	**VACANT Precision**	**Correct Instances**	**Weighted Average F1-Score**
SMOTE Oversample - 40% Vacant – **ZONE 1**	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	70.40% (+/- 0.0028%)	74.60% (+/- 0.0038%)	76.10% (+/- 0.0069%)	73.79% (+/- 0.0021%)	73.80% (+/- 0.002%)
SMOTE Oversample - 100% Vacant – **ZONE 16**	IsHoliday	RandomForest	Interation = 1000	80%	68.60% (+/- 0.0048%)	52.00% (+/- 0.0029%)	85.20% (+/- 0.0024%)	72.51% (+/- 0.0021%)	72.90% (+/- 0.0019%)
SMOTE Oversample - 40% Vacant – **ZONE 1**	Outlier	MLP	'i'attribs+LR 0.1	80%	68.60% (+/- 0.0258%)	71.50% (+/- 0.0189%)	77.30% (+/- 0.0221%)	72.16% (+/- 0.0089%)	72.10% (+/- 0.0088%)
SMOTE Oversample - 40% Vacant – **ZONE 1**	NONE	J48	Conf 0.35	80%	64.00%	74.10%	74.70%	71.13%	71.10%
SMOTE Oversample - 100% Vacant – **ZONE 16**	IsBeginMonth/EndMonth	J48	Conf 0.35	80%	67.20%	51.50%	79.10%	69.51%	69.70%
SMOTE Oversample - 100% Vacant – **ZONE 16**	Outlier	MLP	'i'attribs+LR 0.1	80%	67.20% (+/- 0.0221%)	51.30% (+/- 0.0196%)	77.70% (+/- 0.0171%)	69.51% (+/- 0.011%)	69.40% (+/- 0.0101%)
BASELINE – ZONE 1	NONE	RandomForest	DEFAULT	80%	71.40% (+/- 0.007%)	73.20% (+/- 0.0042%)	76.50% (+/- 0.006%)	73.64% (+/- 0.0037%)	73.60% (+/- 0.0039%)
BASELINE – ZONE 16	NONE	RandomForest	DEFAULT	80%	67.70% (+/- 0.0077%)	52.00% (+/- 0.0034%)	84.20% (+/- 0.0041%)	71.72% (+/- 0.0039%)	71.90% (+/- 0.0038%)

Table 41 – Algorithm modified parameters results by zone - 3 classes

Dataset (2 Classes)	Removed Attributes	Algorithm	Parameters	Train	FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
Undersample 100% Vacant – **ZONE 1**	NONE	RandomForest	Interation = 1000	80%	68.00% (+/- 0.0014%)	85.20% (+/- 0.0048%)	78.46% (+/- 0.0029%)	78.00% (+/- 0.0027%)
Undersample 100% Vacant – **ZONE 16**	Outlier	RandomForest	Interation = 1000	80%	65.10% (+/- 0.0011%)	82.70% (+/- 0.0026%)	76.53% (+/- 0.0014%)	76.60% (+/- 0.0013%)
Undersample 40% Vacant – **ZONE 1**	IsBeginMonth /EndMonth	MLP	't' (att+cls)+LR 0.4	80%	69.20% (+/- 0.0262%)	81.00% (+/- 0.0175%)	74.36% (+/- 0.0133%)	74.30% (+/- 0.0138%)
Undersample 40% Vacant – **ZONE 16**	Outlier	MLP	't' (att+cls)+LR 0.4	80%	70.20% (+/- 0.0144%)	78.20% (+/- 0.0123%)	74.09% (+/- 0.0096%)	74.10% (+/- 0.0095%)
Undersample 100% Vacant – **ZONE 1**	NONE	J48	Conf 0.15	80%	67.10%	74.40%	70.63%	70.60%
Undersample 100% Vacant – **ZONE 16**	IsBeginMonth/EndMonth	J48	Conf 0.15	80%	61.10%	61.10	72.40%	67.60%
BASELINE – ZONE 1	NONE	RandomForest	DEFAULT	80%	67.30% (+/- 0.0029%)	84.90% (+/- 0.0066%)	77.92% (+/- 0.0043%)	78.10% (+/- 0.0045%)
BASELINE – ZONE 16	NONE	RandomForest	DEFAULT	80%	64.00% (+/- 0.004%)	82.50% (+/- 0.0082%)	75.92% (+/- 0.0053%)	76.00% (+/- 0.0052%)

Table 42 – Algorithm modified parameters results by zone - 2 classes

6.3.2.3 Models by Street Occupancy

As discussed in section 6.1.3, the levels of occupancy and the number of instances in a street affect the prediction results. To analyze this the dataset is split into two, one containing the 7 streets with higher occupancy levels (with the label MOST), and another one with the remaining streets (with the label LESS). This allowed the algorithms to have better results when compared to the prediction values obtained when using all streets. However, the precision is smaller when compared to the dataset split by zone.

As in the previous predictions models, RandomForest has the best result for the 3 and 2 class predictions. Compared to the baseline values, the parameters optimizations also had slight improvements in the 3 class and the 2 class prediction, with some differences in the precision of the classes but a similar average F-1 Score. Each of these datasets contains different occupation behavior, with dataset “most occupied streets” having a more defined occupation pattern and occupation values, resulting in a slightly higher prediction precision.

With optimized parameters, the MLP precision had small improvements but worst than the results of the dataset split by zone. Compared to the previous tests, these had a slight increase of 2% on the F1-Score, however they had a decrease of 7% precision in the 3 class prediction for the “most occupied streets”. MLP had difficulties using the patterns in these datasets splits, probably creating overfitted models with high precision on “Vacant” class but significantly lower on the others. These difficulties are also visible in how MLP performs according to the attributes, having his best precision when using all of them.

J48 had a precision improvement of 2% using the optimized parameters and removing the attributes that he considered irrelevant. Despite being the worst, the differences to the other algorithms are small and show that J48 had a similar performance while being a much simpler and faster algorithm when compared to RandomForest or MLP.

Dataset (3 Classes)	**Removed Attributes**	**Algorithm**	**Parameters**	**Train**	**FULL Precision**	**ALMOST FULL Precision**	**VACANT Precision**	**Correct Instances**	**Weighted Average F1-Score**
SMOTE Oversample - 100% Vacant - **MOST**	Outlier	RandomForest	Interation = 1000	80%	68.60% (+/- 0.0023%)	69.80% (+/- 0.0031%)	80.40% (+/- 0.0038%)	73.77% (+/- 0.0026%)	73.80% (+/- 0.0025%)
SMOTE Oversample - 100% Vacant - **LESS**	NONE	RandomForest	Interation = 1000	80%	68.90% (+/- 0.0027%)	55.10% (+/- 0.0028%)	82.60% (+/- 0.0054%)	73.28% (+/- 0.0015%)	73.30% (+/- 0.0014%)
SMOTE Oversample - 100% Vacant - **MOST**	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.35	80%	67.50%	69.40%	79.20%	72.58%	72.60%
SMOTE Oversample - 100% Vacant - **MOST**	NONE	MLP	'i'attribs+LR 0.1	80%	59.80% (+/- 0.0329%)	69.30% (+/- 0.0151%)	80.10% (+/- 0.0089%)	70.60% (+/- 0.0125%)	70.80% (+/- 0.0122%)
SMOTE Oversample - 100% Vacant - **LESS**	NONE	MLP	'i'attribs+LR 0.1	80%	61.00% (+/- 0.018%)	56.40% (+/- 0.0382%)	82.60% (+/- 0.0267%)	69.64% (+/- 0.0085%)	69.90% (+/- 0.0082%)
SMOTE Oversample - 100% Vacant - **LESS**	IsBeginMonth /EndMonth	J48	Conf 0.35	80%	62.1%	54.6%	79.7%	68.76%	69%
BASELINE – MOST	NONE	RandomForest	DEFAULT	80%	68.30% (+/- 0.0048%)	69.30% (+/- 0.0032%)	80.50% (+/- 0.0034%)	73.51% (+/- 0.0021%)	73.60% (+/- 0.0021%)
BASELINE – LESS	NONE	RandomForest	DEFAULT	80%	68.00% (+/- 0.0028%)	56.90% (+/- 0.006%)	82.00% (+/- 0.0048%)	72.99% (+/- 0.0026%)	73.00% (+/- 0.0026%)

Table 43 – Algorithm modified parameters results by street occupation - 3 classes

Dataset (2 Classes)	**Removed Attributes**	**Algorithm**	**Parameters**	**Train**	**FULL Precision**	**VACANT Precision**	**Correct Instances**	**Weighted Average F1-Score**
Undersample 40% Vacant - **MOST**	Outlier	RandomForest	Interation = 1000	80%	77.60% (+/- 0.0035%)	77.80% (+/- 0.0045%)	77.70% (+/- 0.0035%)	77.70% (+/- 0.0035%)
Undersample 100% Vacant - **LESS**	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	65.90% (+/- 0.0029%)	80.60% (+/- 0.0078%)	75.80% (+/- 0.0041%)	75.60% (+/- 0.0039%)
Undersample 40% Vacant – **MOST**	NONE	MLP	't' (att+cls)+LR 0.4	80%	75.30% (+/- 0.0256%)	73.80% (+/- 0.0254%)	74.41% (+/- 0.0192%)	74.30% (+/- 0.0196%)
Undersample 40% Vacant – **LESS**	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	80%	74.10% (+/- 0.0147%)	72.00% (+/- 0.0173%)	73.26% (+/- 0.0123%)	73.10% (+/- 0.0173%)
Undersample 40% Vacant - **LESS**	ExistSpecialEvent	J48	Conf 0.15	65%	70.50%	77.60%	72.98%	72.40%
Undersample 100% Vacant – **MOST**	ExistSpecialEvent	J48	Conf 0.15	65%	66.9%	73.6%	71.19%	70.9%
BASELINE – MOST	NONE	RandomForest	DEFAULT	80%	67.30% (+/- 0.0009%)	79.90% (+/- 0.0036%)	75.49% (+/- 0.002%)	75.50% (+/- 0.0021%)
BASELINE – LESS	NONE	RandomForest	DEFAULT	80%	74.80% (+/- 0.0073%)	76.80% (+/- 0.0106%)	75.82% (+/- 0.0083%)	75.80% (+/- 0.0081%)

Table 44 – Algorithm modified parameters results by street occupancy - 2 classes

6.3.2.4 **Models with 2 Week Split**

Table 45 and Table 46 contains the results of the prediction for the two first weeks of November, training the models with the remaining dataset.

Dataset (3 Classes)	**Removed Attributes**	**Algorithm**	**Parameters**	**Train**	**FULL Precision**	**ALMOST FULL Precision**	**VACANT Precision**	**Correct Instances**	**Weighted Average F1-Score**
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	2 WEEK	43.10% (+/- 0.0373%)	67.10% (+/- 0.0119%)	48.10% (+/- 0.0017%)	49.32% (+/- 0.0037%)	40.80% (+/- 0.0048%)
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	2 WEEK	48.30% (+/- 0.0544%)	45.60% (+/- 0.0575%)	50.20% (+/- 0.0167%)	49.32% (+/- 0.0196%)	43.50% (+/- 0.0256%)
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth	J48	Conf 0.35	2 WEEK	38.80%	48.40%	51.30%	48.95%	46.20%
BASELINE	NONE	J48	DEFAULT	2 WEEK	29.40%	46.80%	67.60%	67.60%	57.50%

Table 45 – Algorithm modified parameters results for 2 week split - 3 classes

Dataset (2 Classes)	Removed Attributes	Algorithm	Parameters	Train	FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
Undersample 100% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.15	2 WEEK	62.90%	68.90%	67.59%	65.10%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	2 WEEK	57.80% (+/- 0.0172%)	67.90% (+/- 0.0309%)	65.57% (+/- 0.0175%)	63.30% (+/- 0.0259%)
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	2 WEEK	71.60% (+/- 0.0044%)	60.10% (+/- 0.0064%)	63.24% (+/- 0.0052%)	61.20% (+/- 0.0066%)
BASELINE	NONE	REPTree	DEFAULT	2 WEEK	53.50%	68.80%	64.06%	47.90%

Table 46 – Algorithm modified parameters results for 2 week split - 2 classes

For the 3 class prediction, there are improvements in the in the precision of the "Full" class, however they are reduced in the other classes, resulting in a lower F1-Score when compared to the baseline values. This indicates that this dataset split offer different challenges to the models, as discussed in section 6.3.2.4, making these parameters not adequate to this dataset.

In the 2 class prediction, the improvements are more noticeable, with the precision of the "Full" class and the average F1-Score getting higher values, demonstrating that these algorithm parameters are effective in understanding the occupation behavior. All of them performed better with the removal of attributes from the dataset, also showing that the attributes "Outlier" and "IsBeginMonth/EndMonth" increase the confusion the dataset.

J48 offered the best performance in the 2 class prediction, however RandomForest has higher precision in the "Full" class. This indicates that RandomForest may have a performance improvement with the parameters tuned explicitly for this dataset, however the small differences demonstrate that J48 is a good algorithm if fast computational is a requirement.

6.3.2.5 **Models for Next Week Test Prediction**

This section presents the result of the experiments where the models are developed for short-term prediction, using 3 weeks to predict the occupancy of 1 week. A test is performed for each month, training the model with week 1,2,3 and testing it with week 4.

Table 47 shows the prediction results for 3 classes, with all the models having a reduced performance. The differences in occupation behavior between the weeks difficult this type of prediction, being this more challenging than the experiments previously made and demonstrating that is difficult to capture the different occupation patterns using a reduced dataset. The fact that the prediction is made for a full week also increases the difficulty, since the weekdays also have different behavior between them. MLP was the algorithm that more often performed better, showing that is more effective when working with noisy data but still having a low precision in the “Full” and “Almost Full” classes. J48 obtained better results for the month of November and the best results for the “Full” class prediction. The results show that the difference between the weeks causes difficulties to the algorithms, making them have inconsistent results through the months.

Train	Dataset (3 Classes)	Removed Attributes	Algorithm	Parameters	FULL Precision	ALMOST FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
October	SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth	MLP	'i'attribs+LR 0.1	29.00% (+/- 0.066%)	45.10% (+/- 0.0533%)	69.60% (+/- 0.0516%)	58.76% (+/- 0.0357%)	55.40% (+/- 0.0385%)
November	SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth/ EndMonth	J48	Conf 0.35	50.00%	38.00%	56.00%	49.54%	49.00%
December	SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	46.20% (+/- 0.0963%)	46.70% (+/- 0.0605%)	58.70% (+/- 0.0148%)	47.59% (+/- 0.0205%)	37.10% (+/- 0.0345%)

Table 47 – Algorithm results for next week prediction - 3 classes

Table 48 shows the results when predicting is for 2 classes, and there is an improvement in the performance of the algorithms, indicating that with 2 classes the complexity of the dataset is reduced and the occupation patterns are more consistent. MLP had worst performance in this dataset, meaning that the parameters used are not adequate for this test. This contrasts with the performance of J48, which had the best results for the month of November without requiring any specific parameter tuning.

Train	Dataset (2 Classes)	Removed Attributes	Algorithm	Parameters	FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
October	Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	55.80% (+/- 0.0076%)	74.40% (+/- 0.0314%)	71.58% (+/- 0.0105%)	68.20% (+/- 0.0129%)
November	Undersample 100% Vacant	ExistSpecialEvent + IsBeginMonth/ EndMonth	J48	Conf 0.15	64.20%	66.60%	65.86%	64.60%
December	Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	57.70%	64.90%	63.98%	57.90%

Table 48 – Algorithm results for next week prediction - 2 classes

In all the experiments, the best results occur when attributes are removed, confirming the results of the previous experiences where the “Outlier” and “IsBeginMonth/EndMonth” are considered to increase the data confusion.

6.3.2.6 Models for Next Day Test Prediction

To test how the models perform using this data for short-term predicting several experiments has been made, training the model with 1 week and testing it with 1 day. The experiments are made by month, training the model with the 2nd week and testing it with the Monday of the 3rd week.

Table 49 and Table 50 shows the results using 3 class and 2 class prediction, having different performance results between them.

Train	Dataset (3 Classes)	Removed Attributes	Algorithm	Parameters	FULL Precision	ALMOST FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
October	SMOTE Oversample - 40% Vacant	IsBeginMonth EndMonth	MLP	'i'attribs+LR 0.1	35.70% (+/- 0.1717%)	75.00% (+/- 0.1318%)	55.10% (+/- 0.0604%)	56.63% (+/- 0.0607%)	53.70% (+/- 0.074%)
November	SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	46.30%	0.00%	73.00%	54.24%	48.30%
December	SMOTE Oversample - 40% Vacant	Outlier	MLP	'i'attribs+LR 0.1	50.00% (+/- 0.1553%)	61.10% (+/- 0.1509%)	100.00% (+/- 0.0434%)	70.00% (+/- 0.0948%)	70.60% (+/- 0.0687%)

Table 49 – Algorithm results for next day prediction - 3 classes

Train	Dataset (2 Classes)	Removed Attributes	Algorithm	Parameters	FULL Precision	VACANT Precision	Correct Instances	Weighted Average F1-Score
October	Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	72.00% (+/- 0.0541%)	78.80% (+/- 0.0522%)	75.86% (+/- 0.0474%)	75.90% (+/- 0.0537%)
November	Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	64.30% (+/- 0.0396%)	75.90% (+/- 0.075%)	72.09% (+/- 0.0672%)	71.70% (+/- 0.0656%)
December	Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	83.30% (+/- 0.0045%)	81.80% (+/- 0.0045%)	82.05% (+/- 0%)	80.10% (+/- 0.0003%)

Table 50 – Algorithm results for next day prediction - 2 classes

Using 3 classes, the models offered reduced performance, showing that is difficult to capture all the patterns that define the status of 3 classes using a small amount of data. MLP had the more consistent results but still with reduced performance, being this more evident in the "Full" class.

When using 2 classes, the results are better and with a reasonable level of precision, demonstrating that when using 2 classes 1 week of data may contain enough information to make next day predictions. MLP had again the most consistent results, with RandomForest having a good performance in December. In the 3 class prediction, the week in December also offered the best performance indicating that, when compared to the others, this week may have a more consistent behavior allowing RandomForest to outperform MLP.

The type of attributes in the dataset also changed between the 3 and 2 class prediction. In the 3 class prediction, the results are better when removing attributes, reducing the noise in an already noisy data. In the 2 class prediction, only the "Outlier" was removed once, indicating that the data is more understandable and the amount of confusion induced by the attributes is reduced.

6.4 Comparison with Original Dataset Prediction

The results obtained through this chapter demonstrates that is possible to predict the parking occupation status within a reasonable level of precision between 70% and 80%, depending on the datasets.

The study developed in section 4.7, used the occupation status of the previous weeks to predict the occupancy status for the current week. This method proved to be ineffective, since it only had a reasonable precision for the most common class, “Vacant”, having very reduced performance in the most relevant classes “Full” and “Almost Full”. These results are explained by the inconsistent occupation values through the days and weeks, as demonstrated in section 4.5, being difficult to have two specific time units with the same behavior at different days.

The complexity of the occupation patterns indicates that is not easy to forecast simply by using the past values at the same time, and that machine learning methods do obtain more accurate results.

The accuracy found through the different experiments showed that the method used can offer reasonable results in different scenarios, being possible to refine the results by adjusting the dataset and model parameters, the granularity of the dataset and the prediction window.

7. Conclusions and Future Work

This chapter presents the main conclusions of this study and the possible future works.

7.1 Conclusions

The primary motivation for this study is to develop a prediction method for parking occupancy, enabling the drivers to know in advance if parking is available on a specific street at a given moment and assert its accuracy.

The data consists of the parking registries collected from on-street multi-space parking meters, and the developed work started by transforming this data into a time series dataset. To obtain the parking occupancy status, that was not available in these data-sets, this study used the number of registries and peak occupation values to assign the status “Full”, “Almost Full” or “Vacant” to a parking in a street.

The development of the prediction models was made using J48, RandomForest, REPTree and Multilayer Perceptron, with the objective of comparing the performance between these methods.

After all the research, experiments and result analysis, the following conclusions were drawn:

- The number of parking registries contained in data collected from the parking meters does not match the number of available parking spaces, indicating that there are many non-registered parking’s or information missing;
- The parking occupancy follows a pattern through the different hours of the day. However, the occupancy values have significant variations through the weekdays, weeks and months, making it impossible to establish a pattern between them;
- The usage of a simple prediction method based on previous week’s values is not effective, since the variations in the occupation pattern result in low precision (the predictive power of this method is, in this case, below 30%).
- The definition of the threshold values in parking occupancy classification system demonstrated to be sensitive and with significant impact on the results. These

values should be carefully adjusted and validated in the streets to verify if the obtained classification is accurate.

- The class imbalance of the dataset demonstrated to affect the precision of the prediction models. The leveling by oversampling proved to be the most effective balancing method.
- The addition to dataset of the contextual attributes about the weather conditions increased the prediction precision up to 5%. The also added attributes “Outliers” and “IsBeginMonth/EndMonth” had the opposite effect and increased the dataset complexity, with the models performing better when they are removed.
- During the experiments, the algorithms do not have significant performance variations between them. However, RandomForest had the more consistent results usually outperforming MLP. J48 is the simpler and faster algorithm, offering slightly worst results but still accurate, while REPTree presented the weaker results. This conclusion is contrary to some of the state-of-the-art reports, where MLP is often the best rated algorithm for this type of problems. In most of these experiments though Random Forest algorithms are not tested or directly compared with MLP.
- MLP had significant improvements when using optimized parameters, however when using some dataset splits the performance decreases, indicating that the model has become more sensitive.
- In this dataset, RandomForest’s precision is higher than the obtained with MLP. When using optimized parameters, MLP best results equals the ones obtained by RandomForest, but with the disadvantage of being slower to train and more challenging to set the parameter values.
- The low standard deviation obtained from the 10 experiences with RandomForest and MLP demonstrate their consistency in the results. However, RandomForest had a significantly lower standard deviation, indicating their results suffer less variation and are less sensitive to the initial conditions.
- RandomForest has a low execution time, the default parameters are robust, and is not computationally intensive, making it possible to execute a daily model training without requiring dedicated or external computational systems.

- The developed prediction models showed an average precision of 70% using 3 classes and of 75% using 2 classes. These values increased when the dataset was split by zone or by streets with higher occupancy.
- The variations in the occupancy patterns make difficult to predict the result of a week using only the previous weeks. However, it was possible to accurately predict the occupancy for one day ahead using data from only the previous week.

7.2 Future Work

Following this research, some experiments can be to improve the results.

- Verify with EMEL the parking occupancy behavior on each street, identifying the peak hours and the number of parking's that occur without having a registry. This will allow to adjust the threshold values better and to improve the results.
- Obtain a dataset containing a larger period of time and analyze the parking behavior. The objective is to verify if the occupation patterns in other months also have significant variations or if they are more consistent.
- Utilize a dataset where was possible to verify on the street when the parking was full or vacant, and then use it to test the models and verify how precise is the prediction.
- Develop the models using a dataset with a smaller time series unit, for example 2 minutes, and verify if the results are more accurate. The additional information may help the algorithms, but it may also increase the noise in the dataset. With this, it is possible to verify how much the computational time increases and to conclude it this leads to better results.
- Analyze if the development of models specific for each street increases the precision.

References

An, S., Han, B., & Wang, J. (2004). Study of the mode of real-time and dynamic parking guidance and information systems based on fuzzy clustering analysis. *Conference on Machine Learning and Cybernetics*, (August), 26–29. https://doi.org/10.1109/ICMLC.2004.1378506

Andrea, D., Klaus, O., & Walter, S. (2000). Event-oriented forecast of the occupancy rate of parking spaces as part of a parking information service. *Proceedings of the 7th World Congress on Intelligent Systems*, (1), 1–8.

Bashiri, M., & Farshbaf Geranmayeh, A. (2011). Tuning the parameters of an artificial neural network using central composite design and genetic algorithm. *Scientia Iranica*, *18*(6), 1600–1608. https://doi.org/10.1016/j.scient.2011.08.031

Blythe, P., Ji, Y., Guo, W., Wang, W., & Tang, D. (2015). Short-term forecasting of available parking space using wavelet neural network model. *IET Intelligent Transport Systems*, *9*(November 2013), 202–209. https://doi.org/10.1049/iet-its.2013.0184

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Burgstahler, D., Knapp, F., Zöller, S., Rückelt, T., & Steinmetz, R. (2014). Where is that car parked? A wireless sensor network-based approach to detect car positions. *Proceedings - Conference on Local Computer Networks, LCN*, *2014–Nov*(November), 514–522. https://doi.org/10.1109/LCNW.2014.6927697

Caicedo, F. (2009). The use of space availability information in “PARC” systems to reduce search times in parking facilities. *Transportation Research Part C: Emerging Technologies*, *17*(1), 56–68. https://doi.org/10.1016/j.trc.2008.07.001

Caicedo, F., Blazquez, C., & Miranda, P. (2012). Prediction of parking space availability in real time. *Expert Systems With Applications*, *39*(8), 7281–7290. https://doi.org/10.1016/j.eswa.2012.01.091

Caliskan, M., Barthels, A., Scheuermann, B., & Mauve, M. (2007). Predicting Parking Lot Occupancy in Vehicular Ad Hoc Networks. *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, 277–281. https://doi.org/10.1109/VETECS.2007.69

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chen, H., Grant-Muller, S., Mussone, L., & Montgomery, F. (2001). A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. *Neural Computing & Applications*, *10*(3), 277–286. https://doi.org/10.1007/s521-001-8054-3

Chen, X. (2014). *Parking Occupancy Prediction and Pattern Analysis*. *Machine Learning Final Projects*.

Chen, Z., Xia, J. C., & Irawan, B. (2013). Development of Fuzzy Logic Forecast Models for Location-Based Parking Finding Services. *Mathematical Problems in Engineering*, *2013*, 1–6. https://doi.org/10.1155/2013/473471

Cherian, J., Luo, J., Guo, H., Ho, S. S., & Wisbrun, R. (2016). ParkGauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics. *Proceedings - IEEE International Conference on Mobile Data Management*, *2016–July*, 92–101. https://doi.org/10.1109/MDM.2016.26

Delibaltov, D., Wu, W., Loce, R. P., & Bernal, E. A. (2013). Parking lot occupancy determination from lamp-post camera images. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Itsc), 2387–2392. https://doi.org/10.1109/ITSC.2013.6728584

Devasena, L. (2014). Comparative Analysis of Random Forest, REP Tree and J48 Classifiers for Credit Risk Prediction. *International Journal of Computer Applications*, 975–8887.

EMEL. (2015). *Relatório & Contas 2015*. Lisbon.

EMEL. (2017a). Dístico de Empresa. Retrieved July 1, 2017, from https://www.emel.pt/pt/disticos/estacionamento-na-via-publica/distico-de-empresa/

EMEL. (2017b). Dístico de Residente. Retrieved July 1, 2017, from https://www.emel.pt/pt/disticos/estacionamento-na-via-publica/distico-de-residente/

EMEL. (2017c). ePark Mobile App. Lisbon: EMEL.

EMEL. (2017d). Identificação das zonas. Retrieved July 1, 2017, from https://www.emel.pt/pt/onde-estacionar/via-publica/tarifarios/identificacao-das-zonas/

EMEL. (2017e). Onde Estacionar. Retrieved July 1, 2017, from https://www.emel.pt/pt/onde-estacionar/via-publica/pesquisa-de-estacionamento/

Fengquan, Y., Jianhua, G., Xiaobo, Z., & Guogang, S. (2015). Real Time Prediction of Unoccupied Parking Space Using Time Series Model. *2015 International Conference on Transportation Information and Safety (ICTIS)*, 370–374. https://doi.org/10.1109/ICTIS.2015.7232145

Gardner, M. ., & Dorling, S. . (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14–15), 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Geng, Y., & Cassandras, C. G. (2013). A New “Smart Parking” System Based on Resource Allocation and Reservations. *Intelligent Transportation Systems, IEEE Transactions on*, *14*(3), 1129–1139. https://doi.org/10.1109/TITS.2013.2252428

Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, *11*(1), 10–18. https://doi.org/10.1145/1656274.1656278

Hall, M. A., & Holmes, G. (2003). Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, *15*(6), 1437–1447. https://doi.org/10.1109/TKDE.2003.1245283

Huang, C. C., & Wang, S. J. (2010). A hierarchical bayesian generation framework for vacant parking space detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *20*(12), 1770–1785. https://doi.org/10.1109/TCSVT.2010.2087510

Karlaftis, M. G., & Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, *19*(3), 387–399. https://doi.org/10.1016/j.trc.2010.10.004

Kianpisheh, A., Mustaffa, N., Limtrairut, P., & Keikhosrokiani, P. (2012). Smart Parking System (SPS) architecture using ultrasonic detector. *International Journal of Software Engineering and Its Applications*, *6*(3), 51–58.

Klappenecker, A., Lee, H., & Welch, J. L. (2014). Finding available parking spaces made easy. *Ad Hoc Networks*, *12*(1), 243–249. https://doi.org/10.1016/j.adhoc.2012.03.002

Kumar, K., Parida, M., & Katiyar, V. K. (2013). Short Term Traffic Flow Prediction for a Non Urban Highway Using Artificial Neural Network. *Procedia - Social and Behavioral Sciences*, *104*, 755–764. https://doi.org/10.1016/j.sbspro.2013.11.170

Lijbers, J. (2016). *Predicting parking lot occupancy using Prediction Instrument Development for Complex Domains*. University of Twente.

LIU, S., GUAN, H., YAN, H., & YIN, H. (2010). Unoccupied Parking Space Prediction of Chaotic Time Series. *Research on Influence of Aggregate Gradation on the Performance of Porous Asphalt Pavement*, *c*, 3738–3746. https://doi.org/10.1061/41127(382)228

Longadge, R., Dongre, S. S., & Malik, L. (2013). Class imbalance problem in data mining: review. *International Journal of Computer Science and Network*, *2*(1), 83–87. https://doi.org/10.1109/SIU.2013.6531574

M. Al-Maqaleh, B., A. Al-Mansoub, A., & N. Al-Badani, F. (2016). Forecasting using Artificial Neural Network and Statistics Models. *International Journal of Education and Management Engineering*, *6*(3), 20–32. https://doi.org/10.5815/ijeme.2016.03.03

Mainetti, L., Marasovic, I., Patrono, L., Solic, P., Stefanizzi, M. L., & Vergallo, R. (2016). A Novel IoT-aware Smart Parking System based on the integration of RFID and WSN technologies. *International Journal of RF Technologies: Research and Applications*, *7*(4), 175–199. https://doi.org/10.3233/RFT-161523

Markevicius, V., Navikas, D., Zilys, M., Andriukaitis, D., Valinevicius, A., & Cepenas, M. (2016). Dynamic Vehicle Detection via the Use of Magnetic Field Sensors. *Sensors*, *16*(1), 78. https://doi.org/10.3390/s16010078

Mathur, S., Jin, T., Kasturirangan, N., Chandrasekaran, J., Xue, W., Gruteser, M., & Trappe, W. (2010). ParkNet: drive-by sensing of road-side parking statistics. *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services - MobiSys '10*, 123. https://doi.org/10.1145/1814433.1814448

Mimbela, L. E. Y., & Klein, L. a. (2000). A Summary Of Vehicle Detection And Surveillance Technologies Used In Intelligent Transportation Systems. *UC Berkeley Transportation Library*, 211.

Pflügler, C., Köhn, T., Schreieck, M., Wiesche, M., & Krcmar, H. (2016). Predicting the Availability of Parking Spaces with Publicly Available Data. *Lecture Notes in Informatics (LNI), Gesellschaft Für Informatik, Bonn*, 361–374.

Piovesan, N., Turi, L., Toigo, E., Martinez, B., & Rossi, M. (2016). Data analytics for smart parking applications. *Sensors (Switzerland)*, *16*(10), 1–25. https://doi.org/10.3390/s16101575

Powers, D. M. W. (2007). Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation, (December).

Pullola, S., Atrey, P. K., & Saddik, A. El. (2007). Towards an intelligent GPS-based vehicle navigation system for finding street parking lots. *ICSPC 2007 Proceedings - 2007 IEEE*

International Conference on Signal Processing and Communications, (November), 1251–1254. https://doi.org/10.1109/ICSPC.2007.4728553

Quinlan, J. R. (1987). Simplifying Decision Trees. *Int. J. Man-Mach. Stud.*, *27*(3), 221–234. https://doi.org/10.1016/S0020-7373(87)80053-6

Rajabioun, T., Foster, B., & Ioannou, P. (2013). Intelligent parking assist. *2013 21st Mediterranean Conference on Control and Automation, MED 2013 - Conference Proceedings*, 1156–1161. https://doi.org/10.1109/MED.2013.6608866

Rajabioun, T., & Ioannou, P. (2015). On-Street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2913–2924. https://doi.org/10.1109/TITS.2015.2428705

Richter, F., Di Martino, S., & Mattfeld, D. C. (2014). Temporal and Spatial Clustering for a Parking Prediction Service. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence* (Vol. November, pp. 278–282). IEEE. https://doi.org/10.1109/ICTAI.2014.49

Salzberg, S. L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, *16*(3), 235–240. https://doi.org/10.1007/BF00993309

Sargent, D. J. (2001). Comparison of artificial neural networks with other statistical approaches. *Cancer*, *91*(S8), 1636–1642. https://doi.org/10.1002/1097-0142(20010415)91:8+<1636::AID-CNCR1176>3.0.CO;2-D

Seo, Y., Urmson, C., & Ratliff, N. (2009). Self-Supervised Aerial Image Analysis for Extracting Parking Lot Structure. *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, (January), 1837–1842.

Shoup, D. C. (2006). Cruising for parking. *Transport Policy*, *13*(6), 479–486. https://doi.org/10.1016/j.tranpol.2006.05.005

Tamrazian, A., Qian, Z. (Sean), & Rajagopal, R. (2015). Where Is My Parking Spot? *Transportation Research Record: Journal of the Transportation Research Board*, *2489*, 77–85. https://doi.org/10.3141/2489-09

Teodorović, D., & Lučić, P. (2006). Intelligent parking systems. *European Journal of Operational Research*, *175*(3), 1666–1681. https://doi.org/10.1016/j.ejor.2005.02.033

Tiedemann, T., Thomas, V., Krell, M. M., Metzen, J. H., & Kirchner, F. (2015). Concept of a Data Thread Based Parking Space Occupancy Prediction in a Berlin Pilot Region, 58–63.

Tsiaras, C., Hobi, L., Hofstetter, F., Liniger, S., & Stiller, B. (2015). ParkITsmart: Minimization of cruising for parking. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, *2015–Octob*. https://doi.org/10.1109/ICCCN.2015.7288448

TuTiempo.net. (2017). TuTiempo.net. Retrieved June 1, 2017, from https://pt.tutiempo.net/

Van Ommeren, J. N., Wentink, D., & Rietveld, P. (2012). Empirical evidence on cruising for parking. *Transportation Research Part A: Policy and Practice*, *46*(1), 123–130. https://doi.org/10.1016/j.tra.2011.09.011

Vasudevan, N., & Parthasarathy, G. C. (2007). Comparative Analysis of Neural Network Techniques Vs Statistical Methods in Capacity Planning. In *5th ACIS International*

Conference on Software Engineering Research, Management & Applications (SERA 2007) (pp. 799–806). IEEE. https://doi.org/10.1109/SERA.2007.66

Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short term traffic forecasting: Overview of objectives and methods. *Transport Reviews*, *24*(June 2015), 533–557. https://doi.org/10.1080/0144164042000195072

Vlahogianni, E., Karlaftis, M., & Golias, J. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. *Transportation Research Part C: Emerging Technologies*, *13*(3), 211–234. https://doi.org/10.1016/j.trc.2005.04.007

Vlahogianni, E., Kepaptsoglou, K., Tsetsos, V., & Karlaftis, M. (2014). Exploiting New Sensor Technologies for Real-Time Parking Prediction in Urban Areas. *Transportation Research Board 93rd Annual Meeting Compendium of Papers*, *14–1673*, 1–19.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2011). Class imbalance, redux. In *Proceedings - IEEE International Conference on Data Mining, ICDM* (pp. 754–763). https://doi.org/10.1109/ICDM.2011.33

Wang, X., & Hanson, A. R. (1998). Parking lot analysis and visualization from aerial images. *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*, 36–41. https://doi.org/10.1109/ACV.1998.732855

Wu, Q. W. Q., Huang, C. H. C., Wang, S. W. S., Chiu, W. C. W., & Chen, T. C. T. (2007). Robust Parking Space Detection Considering Inter-Space Correlation. *Multimedia and Expo, 2007 IEEE International Conference on*, 659–662. https://doi.org/10.1109/ICME.2007.4284736

Xu, B., Wolfson, O., Yang, J., Stenneth, L., Yu, P. S., & Nelson, P. C. (2013). Real-time street parking availability estimation. *Proceedings - IEEE International Conference on Mobile Data Management*, *1*, 16–25. https://doi.org/10.1109/MDM.2013.12

Yamada, K., & Mizuno, M. (2001). A vehicle parking detection method using image segmentation. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, *84*(10), 25–34. https://doi.org/10.1002/ecjc.1039

Yang, Z., Liu, H., & Wang, X. (2003). The research on the key technologies for improving efficiency of parking guidance system. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2*, 1177–1182. https://doi.org/10.1109/ITSC.2003.1252670

Yan-jie, J., Tang, D., Wei-hong, G., Phil, B. T., & Wang, W. (2014). Forecasting available parking space with largest Lyapunov exponents method. *J. Cent. South Univ*, *21*(4), 1624–1632. https://doi.org/10.1007/s11771-014-2104-3

Yanjie, J., Wey, W., & Wey, D. (2007). Available parking space occupancy change characteristicsand short-term forecasting model. *Journal of Southeast Universit*, *23*(4), 604–608.

Zheng, Y., Sutharshan, R., & Christopher, L. (2015). Parking availability prediction for sensor-enabled car parks in smart cities. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (pp. 1–6). IEEE. https://doi.org/10.1109/ISSNIP.2015.7106902

Annexes and Appendices

Appendix A – Parking Records Distribution Ordered by Street

Parking records distribution by Street, alphabetically ordered by street.

Streets highlighted with blue have registries from two zones.

Zone	Street	Park Color	Number of Parking Meters	Number of Registered Parks	% of Total Registered Parks
1	Av. 5 de Outubro	Yellow	12	31439	12.56%
16	Av. Ant. José de Almeida	Yellow	2	3104	1.24%
1	Av. Barbosa du Bocage	Red	2	10382	4.15%
16	Av. Barbosa du Bocage	Red	2	9610	3.84%
1	Av. Conde Valbom	Yellow	3	3587	1.43%
1	Av. da República	Red	7	27100	10.83%
16	Av. da República	Red	3	5542	2.21%
1	Av. de Berna	Red	8	13194	5.27%
16	Av. Defensores de Chaves	Yellow	8	17097	6.83%
1	Av. Duque de Avila	Red	5	5458	2.18%
1	Av. Elias Garcia	Yellow	7	35954	14.36%
16	Av. Elias Garcia	Yellow	1	6983	2.79%
1	Av. João Crisóstomo	Yellow	7	5654	2.26%
1	Av. Marquês de Tomar	Yellow	8	15795	6.31%
1	Av. Miguel Bombarda	Yellow	9	8987	3.59%
16	Av. Miguel Bombarda	Yellow	2	1821	0.73%
1	Av. Poeta Mistral	Yellow	1	3389	1.35%
1	Av. Visconde de Valmor	Yellow	3	12393	4.95%
16	Av. Visconde de Valmor	Yellow	3	5999	2.40%
16	Campo Pequeno	Green	3	2801	1.12%
16	Impasse à R. Eiffel	Yellow	1	665	0.27%
1	Largo Azevedo Perdigão	Yellow	1	678	0.27%
16	R. Arco do Cego	Yellow	3	6885	2.75%
16	R. Chaby Pinheiro	Yellow	1	3592	1.44%
16	R. D. Filipa de Vilhena	Yellow	2	3621	1.45%
16	R. de Entrecampos	Green	2	4347	1.74%
16	R. Eiffel	Yellow	1	974	0.39%
1	R. Marquês Sá da Bandeira	Yellow	3	3244	1.30%
	TOTAL		**110**	**250295**	**100.00%**

Appendix B – Parking Records Distribution Ordered by % Registries

Parking records distribution by Street and ordered by the number of registries in each street.

The column to the most right display the accumulated percentage of registries.

Zone	Street	Park Color	Number of Parking Meters	Number of Registered Parks	% of Total Registered Parks	Accumulated % of Total Registered Parks
1 / 16	Av. Elias Garcia	Yellow	8	42937	17.15%	17.15%
1 / 16	Av. da República	Red	10	32642	13.04%	30.20%
1	Av. 5 de Outubro	Yellow	12	31439	12.56%	42.76%
1 / 16	Av. Barbosa du Bocage	Red	4	19992	7.99%	50.74%
1	Av. Visconde de Valmor	Yellow	6	18392	7.35%	58.09%
16	Av. Defensores de Chaves	Yellow	8	17097	6.83%	64.92%
1	Av. Marquês de Tomar	Yellow	8	15795	6.31%	71.23%
1	Av. de Berna	Red	8	13194	5.27%	76.50%
1 / 16	Av. Miguel Bombarda	Yellow	11	10808	4.32%	80.82%
16	R. Arco do Cego	Yellow	3	6885	2.75%	83.57%
1	Av. João Crisóstomo	Yellow	7	5654	2.26%	85.83%
1	Av. Duque de Avila	Red	5	5458	2.18%	88.01%
16	R. de Entrecampos	Green	2	4347	1.74%	89.75%
16	R. D. Filipa de Vilhena	Yellow	2	3621	1.45%	91.20%
16	R. Chaby Pinheiro	Yellow	1	3592	1.44%	92.63%
1	Av. Conde Valbom	Yellow	3	3587	1.43%	94.07%
1	Av. Poeta Mistral	Yellow	1	3389	1.35%	95.42%
1	R. Marquês Sá da Bandeira	Yellow	3	3244	1.30%	96.72%
16	Av. Ant. José de Almeida	Yellow	2	3104	1.24%	97.96%
16	Campo Pequeno	Green	3	2801	1.12%	99.07%
16	R. Eiffel	Yellow	1	974	0.39%	99.46%
1	Largo Azevedo Perdigão	Yellow	1	678	0.27%	99.73%
16	Impasse à R. Eiffel	Yellow	1	665	0.27%	100.00%
	TOTAL		**110**	**250295**	**100.00%**	

Appendix C – Parking Records Distribution by Date

Parking records distribution by week number, each week starting on Monday and ending on Saturday.

Week	Start Date	End Date	Registries Zone 1	Registries Zone 16	Total Registries	% of Total Registries
Week 01	01/10/2015	04/10/2015	7068	2570	9638	3.85%
Week 02	05/10/2015	11/10/2015	16016	6221	22237	8.88%
Week 03	12/10/2015	18/10/2015	10034	6440	16474	6.58%
Week 04	19/10/2015	25/10/2015	14330	6138	20468	8.18%
Week 05	26/10/2015	01/11/2015	12160	6545	18705	7.47%
Week 06	02/11/2015	08/11/2015	15192	6390	21582	8.62%
Week 07	09/11/2015	15/11/2015	15923	6336	22259	8.89%
Week 08	16/11/2015	22/11/2015	15170	6533	21703	8.67%
Week 09	23/11/2015	29/11/2015	16059	6444	22503	8.99%
Week 10	30/11/2015	06/12/2015	9157	2946	12103	4.84%
Week 11	07/12/2015	13/12/2015	11956	4239	16195	6.47%
Week 12	14/12/2015	20/12/2015	14245	5055	19300	7.71%
Week 13	21/12/2015	27/12/2015	9883	3611	13494	5.39%
Week 14	28/12/2015	31/12/2015	10061	3573	13634	5.45%
TOTAL			**177254**	**73041**	**250295**	**100.00%**

Parking records distribution by month, each month starting on day 1 and ending on his last day.

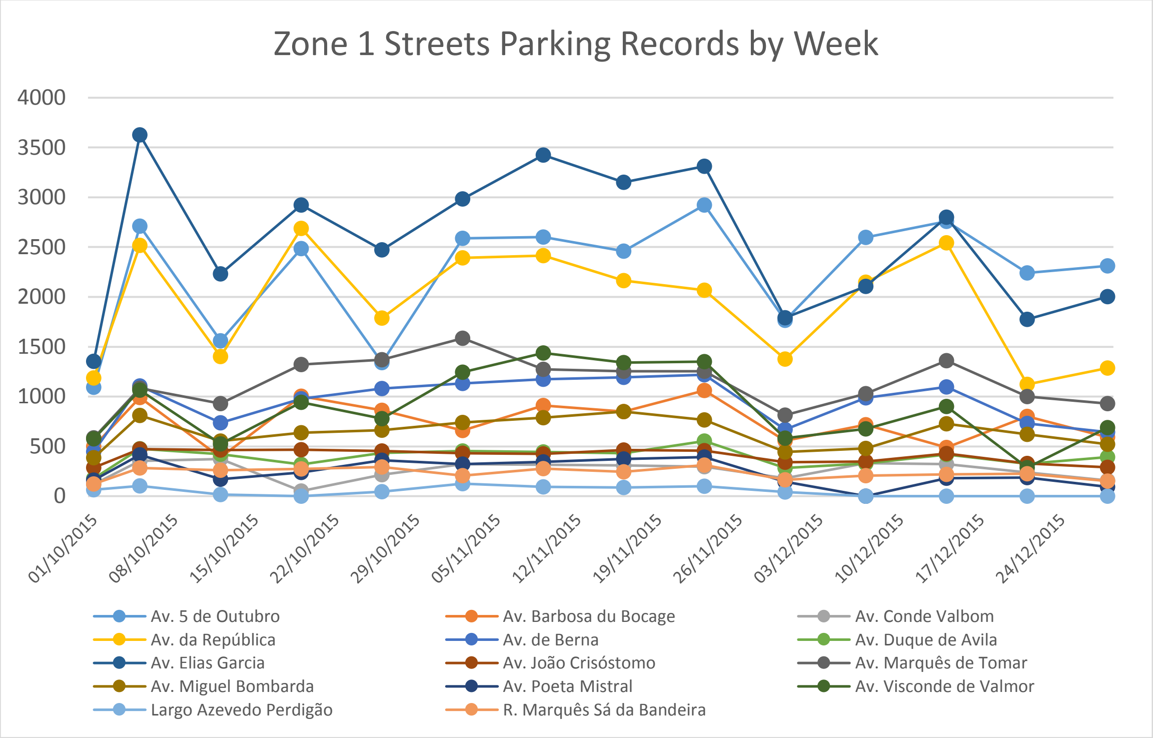
Month	Start Date	End Date	Registries Zone 1	Registries Zone 16	Number of Registries	% of Total Registries
October	01/10/2015	31/10/2015	59587	27906	87493	34.96%
November	01/11/2015	30/11/2015	65531	26951	92482	36.95%
December	01/12/2015	31/12/2015	52136	18184	70320	28.09%
TOTAL			**177254**	**73041**	**250295**	**100.00%**

Parking records distribution by weekday, with the week starting on Monday.

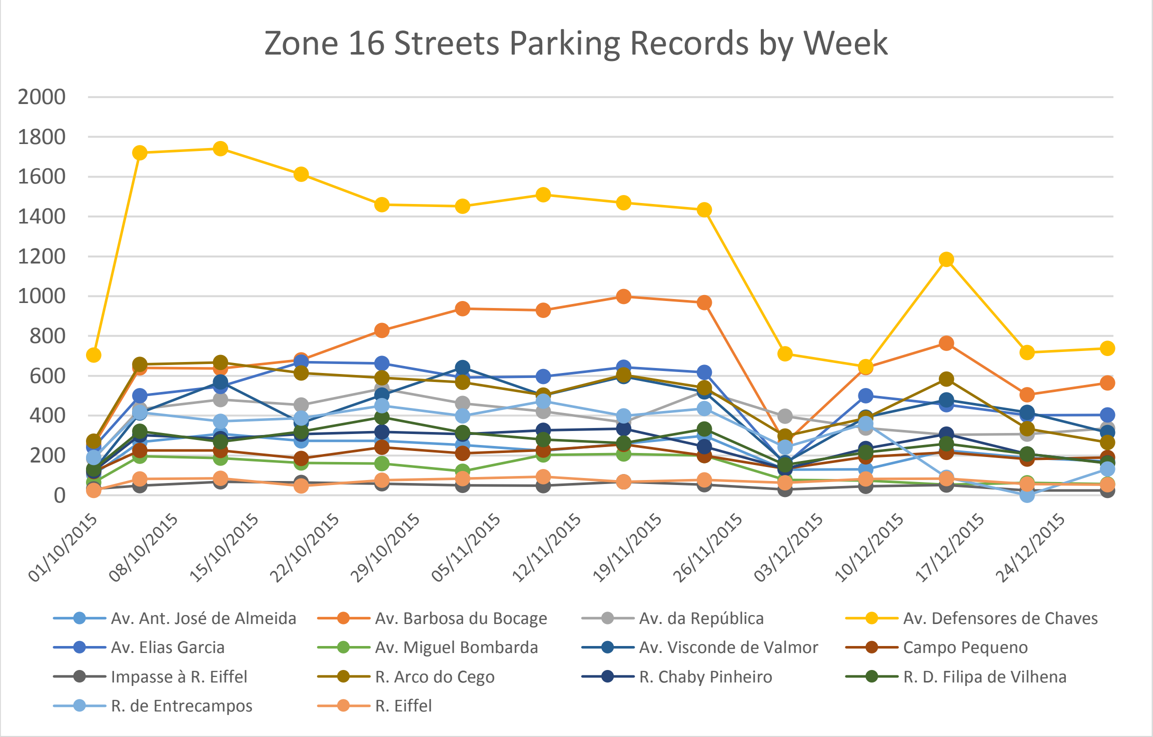
Weekday	Registries Zone 1	Registries Zone 16	Number of Registries	% of Total Registries
Monday	36500	14432	50932	20.35%
Tuesday	34870	13446	48316	19.30%
Wednesday	34505	14634	49139	19.63%
Thursday	33744	15627	49371	19.73%
Friday	34826	14434	49260	19.68%
Saturday	2535	344	2879	1.15%
Sunday	274	124	398	0.16%
TOTAL	**177254**	**73041**	**250295**	**100.00%**

Appendix D – Graph of Parking Records Distribution by Date

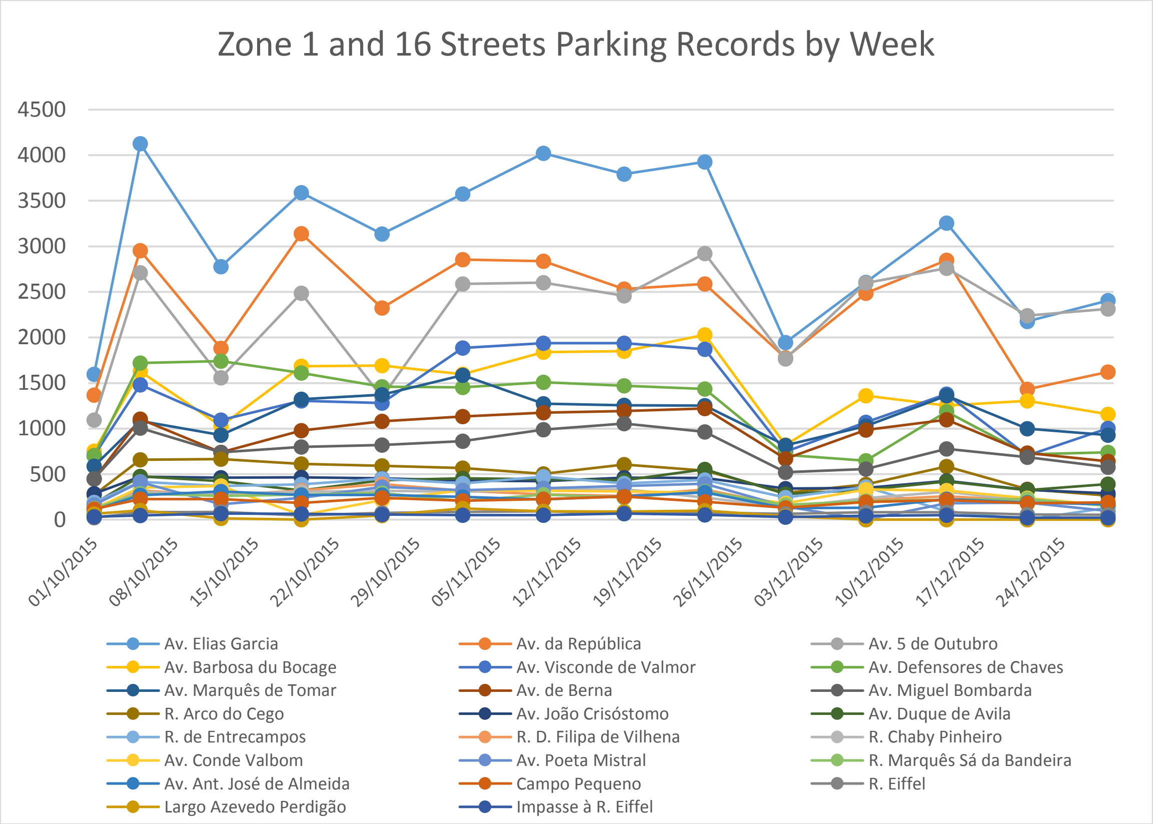
Parking registries from streets in Zone 1 distributed, by week starting on Monday.



Parking registries from streets in Zone 16, distributed by week starting on Monday.



Parking registries from streets in Zone 1 and 16, distributed by week starting on Monday.



Appendix E – Parking Records by Parking Meter – Zone 1

Parking recordings of zone 1 distributed by the Parking Meter address of this zone, and ordered by the Parking Meter Zone Machine Number.

Zone	Address	Parking Color	Machine Serial	Zone Machine Number	Registries
1	Av. 5 de Outubro, 61 P/C	Yellow	453831	1	2199
1	Av. 5 de Outubro, 40	Yellow	453832	3	1663
1	Av. 5 de Outubro, 75	Yellow	453828	4	1715
1	Av. 5 de Outubro, 56	Yellow	453830	6	2594
1	Av. 5 de Outubro, 91 D	Yellow	454656	7	2177
1	Av. 5 de Outubro, 70	Yellow	453835	9	1794
1	Av. 5 de Outubro, 104 P/C	Yellow	453837	10	4799
1	Av. 5 de Outubro, 104	Yellow	453836	11	2378
1	Av. 5 de Outubro, 125	Yellow	450084	12	3757
1	Av. 5 de Outubro, 142	Yellow	453868	14	2870
1	Av. 5 de Outubro, 158 P/C	Yellow	450015	15	4482
1	Av. 5 de Outubro, 164	Yellow	450101	17	1011
1	Av. Barbosa du Bocage, 126 A P/C	Red	450053	18	3867
1	Av. Marquês de Tomar, 42	Yellow	450115	20	438
1	Av. Conde Valbom, 107 P/C	Yellow	450109	22	1472
1	Av. Conde Valbom, 67 A P/C	Yellow	453398	23	1146
1	Av. da Republica, 75	Red	450027	24	1326
1	Av. da Republica, 71 P/L	Red	450026	25	6032
1	Av. da Republica, 63 P/L	Red	450025	26	3008
1	Av. da Republica, 49 B P/L	Red	450022	27	1874
1	Av. da Republica, 41 P/L	Red	454228	28	5619
1	Av. da Republica, 27 B Oposto	Red	450032	29	4930
1	Av. da Republica, 17 P/L	Red	123930	30	4311
1	Av. de Berna, 35	Red	450010	31	1338
1	Av. de Berna, 39 B Oposto	Red	450012	32	1253
1	Av. de Berna - Igreja N. S. Fátima	Red	450013	33	1929
1	Av. de Berna - Fac. Ciências	Red	450011	34	2191
1	Av. de Berna, 27 A	Red	450018	35	1515
1	Av. de Berna, 20	Red	450020	38	992
1	Av. de Berna, 5	Red	370931	39	770
1	Av. de Berna, 2	Red	450017	41	3206
1	Av. Poeta Mistral, 17 A P/C	Yellow	450112	43	3389
1	Av. Duque de Avila, 40	Red	372170	44	2102
1	Av. Duque de Avila, 169 Oposto	Red	372148	45	1193
1	Av. Duque de Avila, 72	Red	372166	47	623
1	Av. Duque de Avila, 98	Red	372149	48	905
1	Av. Duque de Avila, 116	Red	372143	49	635
1	Av. Conde Valbom, 18 Oposto	Yellow	453400	50	969

1	Av. Elias Garcia, 184 C	Yellow	450111	54	894
1	Av. Elias Garcia, 179 A P/C	Yellow	450113	55	3717
1	Av. Elias Garcia, 147 A P/C	Yellow	453396	56	5088
1	Av. Elias Garcia, 136 P/C	Yellow	453399	58	5902
1	Av. Elias Garcia, 84 P/C	Yellow	370915	59	6020
1	Av. Elias Garcia, 96 P/C	Yellow	453866	60	8885
1	Av. Elias Garcia, 74 B P/C	Yellow	368825	61	5448
1	Av. João Crisóstomo, 81	Yellow	454664	62	776
1	Av. João Crisóstomo, 63	Yellow	450052	63	584
1	Av. João Crisóstomo, 58	Yellow	450051	64	1958
1	Av. João Crisóstomo, 39	Yellow	450085	65	651
1	Av. João Crisóstomo, 38 D	Yellow	450088	66	996
1	Av. João Crisóstomo, 25	Yellow	450089	67	47
1	Av. Marquês de Tomar,102 Oposto	Yellow	450049	69	3522
1	Av. Marquês de Tomar, 104	Yellow	450091	70	4317
1	Av. Marquês de Tomar, 94 D P/C	Yellow	450096	72	3190
1	Av. Marquês de Tomar, 66 P/C	Yellow	450094	73	1887
1	Av. Marquês de Tomar, 35 B	Yellow	450100	74	1006
1	Av. Barbosa du Bocage, 90 C P/C	Red	450098	75	6515
1	Av. Marquês de Tomar, 2	Yellow	450054	76	387
1	Av. Marquês de Tomar, 5 A Oposto	Yellow	450099	77	1048
1	Av. Miguel Bombarda, 159	Yellow	450050	78	1719
1	Av. Miguel Bombarda, 139	Yellow	450046	80	1714
1	Av. Miguel Bombarda, 98 D	Yellow	450043	81	513
1	Av. Miguel Bombarda, 54	Yellow	450045	83	1537
1	Av. Miguel Bombarda, 69 A	Yellow	450042	84	786
1	Av. Miguel Bombarda, 40	Yellow	450041	85	1100
1	Av. Miguel Bombarda, 59	Yellow	450086	86	791
1	Av. Miguel Bombarda, 20 C	Yellow	450087	87	109
1	Av. Miguel Bombarda, 21	Yellow	450097	88	718
1	Av. Visconde Valmor, 71 P/C	Yellow	453392	89	3218
1	Av. Visconde Valmor, 48 P/C	Yellow	453393	91	3605
1	Av. Visconde Valmor, 37 P/C	Yellow	453391	93	5570
1	Av. João Crisóstomo, 68	Yellow	450047	95	642
1	R. Marquês Sá da Bandeira, 46 A	Yellow	450081	96	464
1	R. Marquês Sá da Bandeira, 94	Yellow	450080	98	1227
1	R. Marquês Sá da Bandeira, 110 Oposto	Yellow	450082	99	1553
1	Largo Azevedo Perdigão	Yellow	450083	100	678
	TOTAL				**177254**

Appendix F – Parking Records by Parking Meter – Zone 16

Parking recordings of zone 16 distributed by the Parking Meter address of this zone, and ordered by the Parking Meter Zone Machine Number.

Zone	Address	Parking Color	Machine Serial	Zone Machine Number	Registries
16	Av. Barbosa du Bocage, 47 A P/C	Red	454666	1	4828
16	Av. da Republica, n.º 56	Red	450102	5	1444
16	Av. da Republica, 42 P/L	Red	450108	6	1612
16	Av. Miguel Bombarda, 4 A	Yellow	450106	7	976
16	Av. da Republica, 32 B P/L	Red	365919	8	2486
16	Av. Miguel Bombarda, 7 A	Yellow	450105	9	845
16	Av. Defensores de Chaves, 93 C	Yellow	367901	10	3120
16	Av. Defensores de Chaves, 58	Yellow	368795	12	610
16	Av. Defensores de Chaves, 79 B	Yellow	368819	13	1717
16	Av. Ant. José de Almeida, 40 Oposto	Yellow	361042	14	1804
16	Av. Defensores de Chaves, 48 B	Yellow	368621	15	2016
16	Av. Defensores de Chaves, 69 E	Yellow	365320	16	2377
16	Av. Defensores de Chaves, 42	Yellow	367839	18	2599
16	Av. Defensores de Chaves, 61 F	Yellow	361458	19	2722
16	Av. Ant. José de Almeida, 42	Yellow	368606	20	1300
16	Av. Defensores de Chaves, 34 A	Yellow	364451	21	1936
16	Av. Elias Garcia, 56 C P/C	Yellow	365316	22	6983
16	Av. Barbosa du Bocage, 19 P/C	Red	454669	23	4782
16	Av. Visconde Valmor, 19 A	Yellow	365944	29	4613
16	Av. Visconde Valmor, 10	Yellow	361431	30	588
16	Av. Visconde Valmor, 7	Yellow	361076	31	798
16	Campo Pequeno, 39	Green	454690	39	536
16	Campo Pequeno, 51	Green	454672	40	402
16	R. Chaby Pinheiro, 21	Yellow	361481	49	3592
16	Campo Pequeno, 14	Green	361483	50	1863
16	R. de Entrecampos, 18 A	Green	361486	51	2557
16	R. de Entrecampos, 24 C	Green	361485	53	1790
16	R. Arco do Cego, 59 P/C	Yellow	454670	54	2658
16	R. Arco do Cego, 19 oposto	Yellow	365469	55	1467
16	R. Arco do Cego, 19 A P/C	Yellow	365474	56	2760
16	R. D. Filipa de Vilhena, 38	Yellow	364463	59	1832
16	R. D. Filipa de Vilhena, 7	Yellow	364465	60	1789
16	R. Eiffel, 6 A	Yellow	365473	61	974
16	Impasse à R. Eiffel, 15	Yellow	454668	100	665
	TOTAL				**73041**

Appendix G – Parking Occupancy Averages

Parking occupancy averages distributed by Zone and ordered by Average Maximum Occupancy

Zone	Average Maximum Occupancy	Average Occupancy	Average Median	Average Variation Coefficient	Average Registries by Hour
Zone 1	438	197	216	37%	256
Zone 16	215	88	95	37%	107

Parking occupancy averages distributed by Street and ordered by Average Maximum Occupancy

Street	Average Maximum Occupancy	Average Occupancy	Average Median	Average Variation Coefficient	Average Registries by Hour
Av. Elias Garcia	131	49	53	42%	62
Av. 5 de Outubro	100	38	41	48%	45
Av. da República	83	29	31	50%	45
Av. Barbosa du Bocage	71	21	23	63%	30
Av. Visconde de Valmor	65	22	23	51%	27
Av. Defensores de Chaves	59	22	23	48%	25
Av. Marquês de Tomar	56	19	20	47%	23
Av. Miguel Bombarda	39	13	13	48%	16
Av. de Berna	36	12	12	48%	19
Av. João Crisóstomo	32	7	7	51%	8
R. Arco do Cego	29	9	9	58%	10
R. de Entrecampos	25	6	7	82%	5
R. Marquês Sá da Bandeira	25	4	4	65%	6
R. D. Filipa de Vilhena	23	4	4	69%	5
Av. Duque de Avila	22	6	6	81%	8
Av. Poeta Mistral	22	4	5	89%	5
R. Chaby Pinheiro	21	5	5	67%	5
Av. Conde Valbom	19	4	4	83%	5
Av. Ant. José de Almeida	16	4	4	78%	5
Campo Pequeno	16	4	4	56%	4
R. Eiffel	11	1	1	65%	1
Largo Azevedo Perdigão	8	1	0	173%	1
Impasse à R. Eiffel	6	1	1	143%	1

Appendix H – Outlier Days

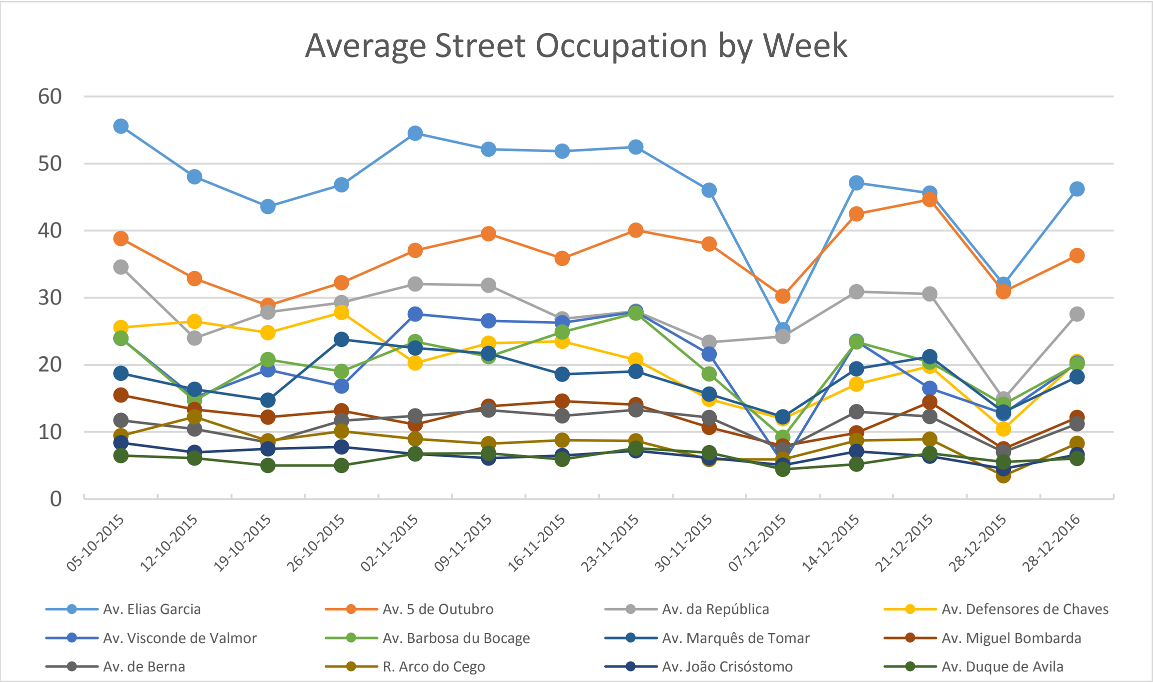
Number of outlier days in each street ordered by street with average maximum occupancy.

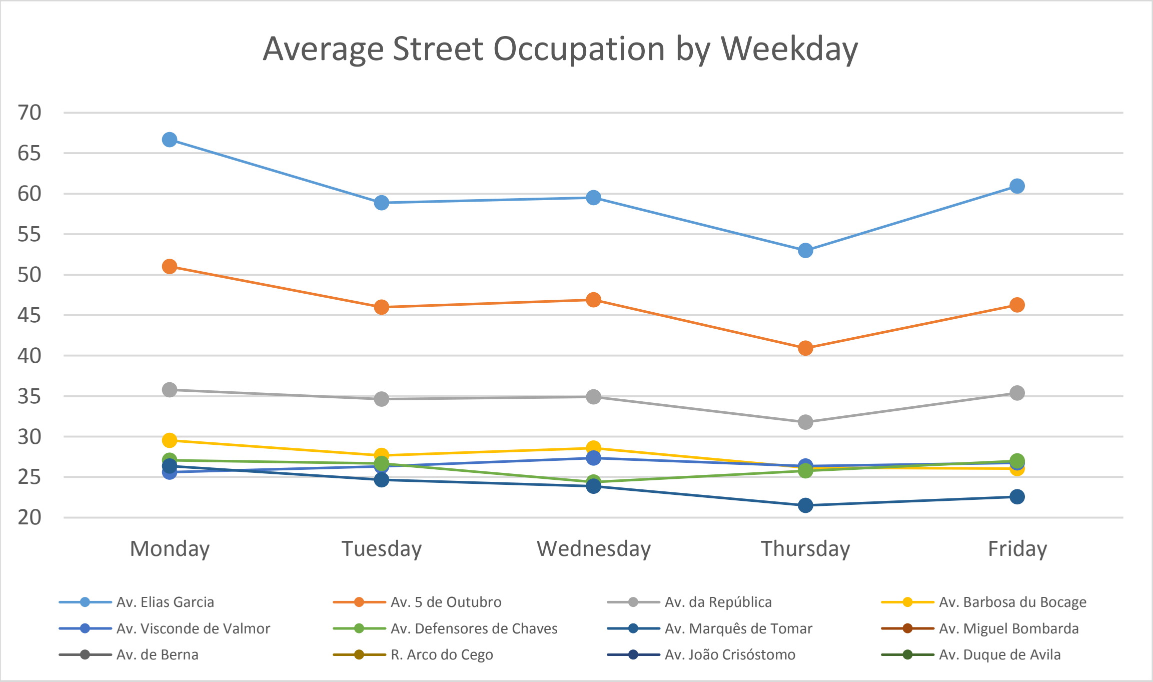
Street	Average Maximum Occupancy	Monday	Tuesday	Wednesday	Thursday	Friday	TOTAL
Av. Elias Garcia	131	0	0	2	0	1	3
Av. 5 de Outubro	100	1	1	2	2	0	6
Av. da República	83	0	0	0	0	0	0
Av. Barbosa du Bocage	71	0	0	0	0	0	0
Av. Visconde de Valmor	65	0	0	0	0	1	1
Av. Defensores de Chaves	59	0	1	1	1	1	4
Av. Marquês de Tomar	56	0	0	1	0	0	1
Av. Miguel Bombarda	39	0	0	1	0	0	1
Av. de Berna	36	0	0	0	0	0	0
Av. João Crisóstomo	32	0	0	0	0	0	0
R. Arco do Cego	29	0	0	1	1	0	2
R. de Entrecampos	25	0	0	0	0	0	0
R. Marquês Sá da Bandeira	25	0	0	1	0	0	1
R. D. Filipa de Vilhena	23	0	0	0	0	0	0
Av. Duque de Avila	22	0	0	0	0	0	0
Av. Poeta Mistral	22	0	0	0	0	0	0
R. Chaby Pinheiro	21	0	0	1	0	0	1
Av. Conde Valbom	19	0	0	0	0	0	0
Av. Ant. José de Almeida	16	0	0	0	0	0	0
Campo Pequeno	16	0	0	0	0	0	0
R. Eiffel	11	0	0	0	0	0	0
Largo Azevedo Perdigão	8	0	0	0	0	0	0
Impasse à R. Eiffel	6	0	0	0	0	0	0

Number of streets with outlier by week.

Week	Monday	Tuesday	Wednesday	Thursday	Friday	TOTAL
01-10-2015	-	-	-	0	0	0
05-10-2015	0	0	0	0	0	0
12-10-2015	0	0	2	1	0	3
19-10-2015	0	0	0	0	0	0
26-10-2015	1	1	0	0	0	2
02-11-2015	0	0	0	0	0	0
09-11-2015	0	0	0	0	0	0
16-11-2015	0	0	0	0	0	0
23-11-2015	0	0	1	0	0	1
30-11-2015	0	1	7	1	1	10
07-12-2015	0	0	0	0	0	0
14-12-2015	0	0	0	0	0	0
21-12-2015	0	0	0	2	0	2
28-12-2015	0	0	0	2	-	2

Appendix I – Average Street Occupancy by Week and Weekday





Appendix J – Occupation Correlation and AVD for Weeks and Weekdays

Average correlation and average distance by zone and street between weeks.

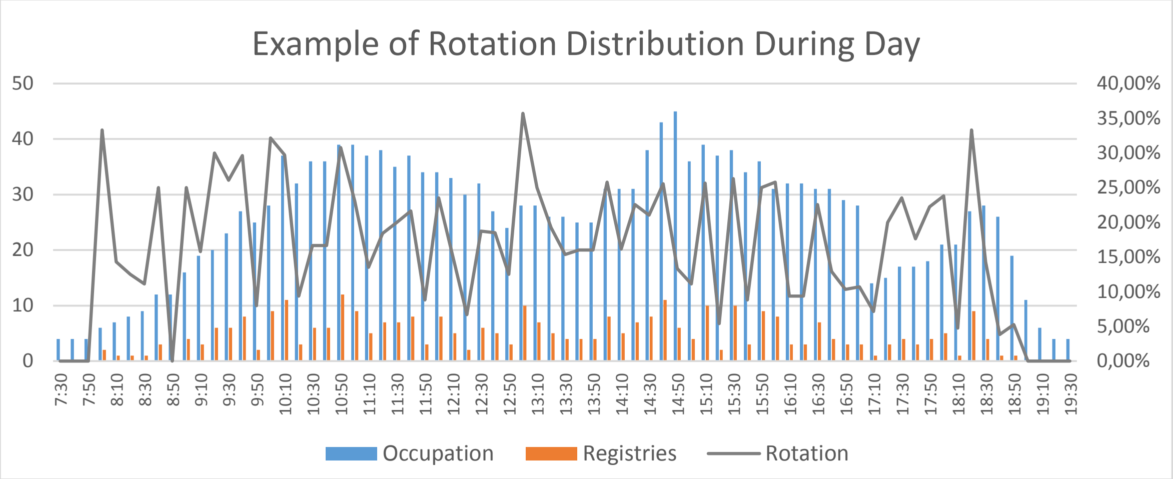
Street	Type	01-10-2015	05-10-2015	12-10-2015	19-10-2015	26-10-2015	02-11-2015	09-11-2015	16-11-2015	23-11-2015	30-11-2015	07-12-2015	14-12-2015	21-12-2015	28-12-2015	Average
ZONE 1	Corr.	0.88	0.91	0.80	0.87	0.78	0.90	0.90	0.90	0.91	0.63	0.76	0.90	0.69	0.85	0.83
	AVD	83.82	36.47	49.69	53.12	50.18	33.54	35.63	34.14	37.08	59.80	47.05	35.17	58.85	60.77	48.24
ZONE 16	Corr.	0.92	0.91	0.91	0.90	0.91	0.91	0.91	0.91	0.91	0.63	0.77	0.90	0.64	0.82	0.85
	AVD	35.89	18.90	18.00	23.11	19.56	17.02	16.74	18.19	17.63	29.82	21.68	17.95	28.85	29.24	22.33
Av. Barbosa du Bocage	Corr.	0.88	0.83	0.77	0.85	0.84	0.86	0.84	0.85	0.87	0.57	0.64	0.78	0.66	0.80	0.79
	AVD	9.44	5.66	6.45	5.89	5.88	5.31	5.95	5.79	6.42	7.97	7.84	6.51	7.79	6.82	6.69
Av. Elias Garcia	Corr.	0.83	0.87	0.75	0.84	0.72	0.87	0.88	0.87	0.87	0.56	0.72	0.87	0.68	0.80	0.80
	AVD	21.04	11.12	13.13	15.68	14.85	10.35	10.50	10.38	10.64	16.50	12.76	10.34	15.64	16.78	13.55
Av. Miguel Bombarda	Corr.	0.83	0.85	0.83	0.84	0.85	0.84	0.85	0.85	0.85	0.58	0.74	0.79	0.62	0.78	0.79
	AVD	6.56	3.89	3.72	3.68	3.35	3.35	3.58	4.00	3.47	4.68	3.83	3.49	5.00	4.50	4.08
Av. da República	Corr.	0.86	0.85	0.75	0.84	0.70	0.87	0.86	0.85	0.86	0.65	0.74	0.84	0.58	0.74	0.79
	AVD	12.21	7.68	8.30	10.39	9.44	7.33	7.20	7.04	6.78	9.32	9.20	8.35	10.78	10.92	8.92
Av. Visconde de Valmor	Corr.	0.85	0.83	0.77	0.81	0.65	0.84	0.84	0.84	0.84	0.51	0.58	0.81	0.52	0.70	0.74
	AVD	9.53	6.24	6.62	6.84	7.94	6.84	6.47	7.05	6.84	9.53	7.90	6.52	8.98	8.84	7.58
Av. 5 de Outubro	Corr.	0.84	0.84	0.68	0.72	0.50	0.85	0.84	0.83	0.85	0.55	0.72	0.85	0.63	0.80	0.75
	AVD	17.20	9.04	12.97	15.60	15.10	9.14	9.00	9.12	9.50	14.15	13.33	9.21	13.47	14.52	12.24
Av. de Berna	Corr.	0.78	0.79	0.65	0.80	0.81	0.82	0.82	0.82	0.78	0.54	0.67	0.80	0.57	0.73	0.74
	AVD	5.45	3.49	4.36	3.35	3.31	3.39	3.40	3.15	4.41	4.39	4.24	3.51	4.34	4.33	3.94
Av. Conde Valbom	Corr.	0.00	0.73	0.72	0.00	0.45	0.72	0.71	0.71	0.74	0.59	0.69	0.71	0.55	0.64	0.57
	AVD	1.85	1.69	1.63	2.02	2.06	1.58	1.43	1.63	1.49	1.62	1.81	1.62	1.84	1.64	1.71
Av. Duque de Avila	Corr.	0.76	0.77	0.75	0.65	0.69	0.80	0.79	0.75	0.74	0.50	0.64	0.75	0.59	0.71	0.71
	AVD	3.35	2.31	2.23	2.59	2.34	2.10	2.17	2.42	3.06	3.06	2.64	2.22	2.63	2.57	2.55
Av. João Crisóstomo	Corr.	0.76	0.80	0.78	0.80	0.80	0.79	0.80	0.79	0.81	0.64	0.62	0.76	0.65	0.72	0.75
	AVD	3.82	2.04	2.43	2.19	2.36	2.13	2.08	2.14	2.31	2.56	2.90	2.50	2.55	2.77	2.49
Av. Marquês de Tomar	Corr.	0.83	0.86	0.77	0.81	0.87	0.87	0.88	0.87	0.86	0.66	0.73	0.86	0.68	0.77	0.81
	AVD	8.97	4.73	5.72	6.73	5.36	5.57	4.52	4.44	4.60	6.19	5.60	4.84	6.58	6.78	5.76
Av. Poeta Mistral	Corr.	0.60	0.62	0.33	0.38	0.61	0.60	0.57	0.60	0.60	0.29	0.00	0.39	0.42	0.26	0.45
	Avg D	2.64	2.43	2.35	2.66	2.25	2.04	2.60	2.42	2.41	2.32	2.46	2.26	2.13	2.31	2.38
Largo Azevedo Perdigão	Corr.	0.28	0.27	0.14	0.00	0.19	0.31	0.30	0.24	0.21	0.14	0.00	0.00	0.00	0.00	0.15
	Avg D	0.92	1.00	0.56	0.53	0.66	1.18	0.78	0.78	1.29	0.76	0.53	0.53	0.53	0.53	0.76
R. Marquês Sá da Bandeira	Corr.	0.66	0.62	0.67	0.46	0.67	0.57	0.68	0.66	0.53	0.43	0.49	0.61	0.54	0.63	0.59
	Avg D	2.27	1.93	1.73	2.69	1.91	1.85	1.71	1.62	2.38	1.94	2.01	1.73	1.85	1.99	1.97
Av. Ant. José de Almeida	Corr.	0.70	0.74	0.70	0.48	0.68	0.69	0.71	0.68	0.71	0.50	0.56	0.70	0.50	0.66	0.64
	AvgD	1.86	1.45	1.63	2.70	1.75	1.65	1.40	1.87	1.57	1.67	1.61	1.41	1.77	1.61	1.71
Av. Defensores de Chaves	Corr.	0.85	0.84	0.85	0.83	0.80	0.84	0.83	0.84	0.85	0.53	0.62	0.79	0.57	0.71	0.77
	Avg D	9.83	6.39	6.19	7.38	6.93	5.54	5.76	5.81	5.62	8.25	7.52	6.48	8.23	8.21	7.01
Campo Pequeno	Corr.	0.72	0.63	0.63	0.66	0.66	0.53	0.63	0.62	0.70	0.47	0.63	0.66	0.51	0.53	0.61
	Avg D	2.06	1.95	1.60	1.51	1.68	2.06	1.76	1.82	1.47	1.84	1.57	1.53	1.96	2.10	1.78
Impasse à R. Eiffel	Corr.	0.50	0.40	0.29	0.47	0.39	0.37	0.47	0.38	0.39	0.37	0.37	0.41	0.20	0.23	0.37
	Avg D	0.41	0.42	0.68	0.49	0.50	0.45	0.40	0.73	0.45	0.40	0.43	0.48	0.44	0.52	0.49
R. Arco do Cego	Corr.	0.79	0.76	0.81	0.82	0.81	0.83	0.82	0.82	0.81	0.55	0.65	0.81	0.55	0.68	0.75
	Avg D	4.16	4.19	3.20	2.86	2.81	2.45	2.59	2.84	2.63	3.34	3.00	2.68	3.61	3.67	3.14
R. Chaby Pinheiro	Corr.	0.73	0.75	0.79	0.75	0.73	0.78	0.79	0.75	0.76	0.60	0.70	0.73	0.47	0.65	0.71
	AvgD	2.48	1.89	1.71	1.93	2.22	1.74	1.82	1.98	1.73	2.07	1.85	1.94	2.30	2.01	1.98
R. Eiffel	Corr.	0.55	0.48	0.54	0.44	0.59	0.53	0.57	0.59	0.61	0.37	0.53	0.57	0.44	0.47	0.52
	Avg D	0.52	0.51	0.46	0.48	0.43	0.44	0.56	0.40	0.42	0.51	0.57	0.53	0.50	0.46	0.49
R. de Entrecampos	Corr.	0.60	0.60	0.62	0.49	0.59	0.58	0.62	0.62	0.64	0.48	0.62	0.25	0.00	0.25	0.50
	Avg D	3.97	3.16	2.62	4.15	3.13	3.38	3.39	2.84	2.70	2.93	2.68	3.55	3.89	3.50	3.28
R. D. Filipa de Vilhena	Corr.	0.77	0.75	0.67	0.74	0.76	0.75	0.73	0.73	0.76	0.50	0.53	0.73	0.49	0.68	0.68
	Avg D	2.19	1.92	1.87	1.68	1.77	1.74	1.62	1.55	1.58	1.94	2.09	1.67	2.16	1.76	1.82

Average correlation and average distance by zone and street between weekdays.

Street	Type	Monday	Tuesday	Wednesday	Thursday	Friday	Average
ZONE 1	Corr.	0.88	0.86	0.85	0.89	0.84	0.87
	AVD	51.32	50.67	44.38	51.12	56.16	50.73
ZONE 16	Corr.	0.89	0.80	0.90	0.88	0.82	0.86
	AVD	21.48	20.01	17.51	19.99	22.66	20.33
Av. Barbosa du Bocage	Corr.	0.85	0.81	0.82	0.80	0.86	0.83
	AVD	7.69	7.28	6.73	7.68	4.76	6.83
Av. Elias Garcia	Corr.	0.85	0.83	0.84	0.88	0.91	0.86
	AVD	15.67	14.28	12.93	14.95	9.28	13.42
Av. Miguel Bombarda	Corr.	0.86	0.81	0.81	0.84	0.86	0.84
	AVD	4.34	4.14	4.01	4.33	8.05	4.97
Av. da República	Corr.	0.84	0.85	0.86	0.88	0.74	0.83
	AVD	9.36	9.14	8.64	8.87	14.09	10.02
Av. 5 de Outubro	Corr.	0.77	0.80	0.81	0.79	0.74	0.78
	AVD	8.55	7.57	6.67	7.25	4.35	6.88
Av. Visconde de Valmor	Corr.	0.77	0.78	0.78	0.86	0.78	0.79
	AVD	13.16	12.51	11.04	12.19	1.89	10.16
Av. de Berna	Corr.	0.80	0.77	0.76	0.82	0.67	0.77
	AVD	4.18	4.34	3.99	4.21	3.08	3.96
Av. Conde Valbom	Corr.	0.79	0.73	0.77	0.79	0.78	0.77
	AVD	1.94	1.81	1.75	1.77	3.07	2.07
Av. Duque de Avila	Corr.	0.76	0.71	0.70	0.75	0.88	0.76
	AVD	2.68	2.70	2.79	2.83	6.42	3.48
Av. João Crisóstomo	Corr.	0.78	0.77	0.79	0.78	0.83	0.79
	AVD	2.84	2.67	2.68	2.84	2.15	2.63
Av. Marquês de Tomar	Corr.	0.87	0.83	0.82	0.84	0.55	0.78
	AVD	5.86	6.07	5.56	6.11	0.72	4.86
Av. Poeta Mistral	Corr.	0.77	0.66	0.71	0.72	0.68	0.71
	Avg D	2.22	2.29	2.13	2.33	2.10	2.21
Largo Azevedo Perdigão	Corr.	0.48	0.47	0.47	0.51	0.76	0.54
	Avg D	0.68	0.64	0.62	0.66	1.67	0.85
R. Marquês Sá da Bandeira	Corr.	0.62	0.64	0.66	0.66	0.73	0.66
	Avg D	2.11	2.00	2.08	2.03	7.86	3.22
Av. Ant. José de Almeida	Corr.	0.68	0.72	0.73	0.72	0.67	0.71
	AvgD	1.73	1.48	1.54	1.58	2.14	1.70
Av. Defensores de Chaves	Corr.	0.69	0.68	0.77	0.72	0.51	0.67
	Avg D	8.94	7.84	7.45	7.52	0.58	6.47
Campo Pequeno	Corr.	0.60	0.51	0.70	0.66	0.82	0.66
	Avg D	1.88	1.87	1.73	1.76	3.47	2.14
Impasse à R. Eiffel	Corr.	0.45	0.49	0.48	0.47	0.85	0.55
	Avg D	0.57	0.52	0.48	0.55	2.20	0.86
R. Arco do Cego	Corr.	0.80	0.69	0.84	0.78	0.55	0.73
	Avg D	3.23	3.20	3.06	3.34	0.65	2.69
R. Chaby Pinheiro	Corr.	0.79	0.78	0.79	0.76	0.74	0.77
	AvgD	2.01	2.03	2.01	2.05	2.83	2.19
R. Eiffel	Corr.	0.58	0.56	0.58	0.54	0.77	0.60
	Avg D	0.62	0.67	0.66	0.66	2.03	0.93
R. de Entrecampos	Corr.	0.74	0.62	0.76	0.73	0.74	0.72
	Avg D	3.54	2.89	2.91	2.90	2.83	3.01
R. D. Filipa de Vilhena	Corr.	0.78	0.80	0.79	0.80	0.77	0.79
	Avg D	1.77	1.81	1.72	1.82	2.03	1.83

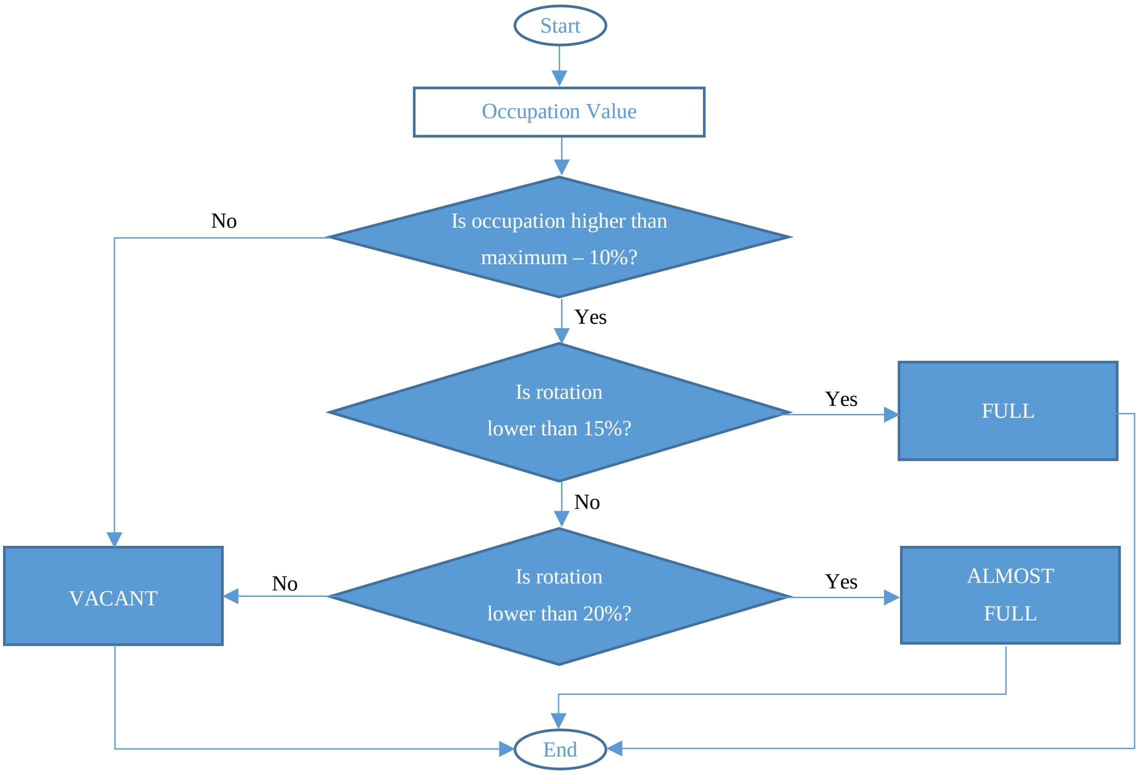
Appendix K – Example of Parking Rotation and Classification

Example of the parking rotation in one street during one day.



Example of the parking status classification algorithm using three classes according to the following values:

Occupancy Status	Rotation	Outcome
> 90% maximum (10% threshold to maximum value)	<= 15%	Full
> 90% maximum (10% threshold to maximum value)	>15% and <= 20%	Almost Full
< 90% maximum (10% threshold to maximum value)	All Values	Vacant


Start
Occupation Value
Is occupation higher than maximum – 10%?
No
Yes
Is rotation lower than 15%?
Yes
FULL
No
Is rotation lower than 20%?
No
VACANT
Yes
ALMOST FULL
End

Appendix L – Street Average Rotation at 10% High Demand Period Threshold

Street average weekday occupancy and rotation with the high demand period set to a threshold of 10% maximum occupancy. The table is order by the streets with higher occupancy.

Street	Average Weekday (Absolute Max) Occupancy	Average Weekday (Top 12 Max) Occupancy	Remain Vacant Parking Spaces at 10% Threshold	Average Rotation in High Demand Periods (10% Threshold)
Av. Elias Garcia	123.0	120.0	12.0	15.29%
Av. 5 de Outubro	92.8	88.2	8.8	14.79%
Av. da República	78.8	75.0	7.5	18.79%
Av. Barbosa du Bocage	68.4	64.9	6.5	15.06%
Av. Visconde de Valmor	64.0	61.0	6.1	14.04%
Av. Defensores de Chaves	57.0	55.2	5.5	13.62%
Av. Marquês de Tomar	52.6	50.7	5.1	13.91%
Av. Miguel Bombarda	36.0	34.3	3.4	15.03%
Av. de Berna	34.0	32.2	3.2	18.67%
R. Arco do Cego	27.4	25.8	2.6	13.42%
Av. João Crisóstomo	23.4	22.1	2.2	14.36%
R. de Entrecampos	23.2	21.6	2.2	14.77%
Av. Duque de Avila	20.0	19.0	1.9	14.22%
Av. Poeta Mistral	18.2	17.3	1.7	13.04%
R. Chaby Pinheiro	17.6	16.8	1.7	13.13%
R. D. Filipa de Vilhena	17.2	15.4	1.5	17.90%
R. Marquês Sá da Bandeira	17.0	15.9	1.6	14.43%
Av. Conde Valbom	16.8	15.6	1.6	17.60%
Av. Ant. José de Almeida	14.2	13.1	1.3	14.95%
Campo Pequeno	14.0	13.1	1.3	12.83%
R. Eiffel	8.4	7.1	0.7	13.66%
Largo Azevedo Perdigão	7.2	6.7	0.7	12.30%
Impasse à R. Eiffel	5.2	4.8	0.5	16.18%

Appendix M – Occupancy Prediction Based on Previous Weeks Values

Results of Prediction Based On The Value of the Week Before

Street	Precision Full	Recall Full	Precision Almost Full	Recall Almost Full	Precision Vacant	Recall Vacant	Accuracy	Average F1-Score
Av. Elias Garcia	0.00%	0.00%	0.00%	0.00%	79.49%	79.43%	**78.82%**	**26.39%**
Av. 5 de Outubro	0.00%	0.00%	0.00%	0.00%	79.24%	79.38%	**78.64%**	**26.44%**
Av. da República	0.00%	0.00%	1.82%	5.00%	79.71%	79.59%	**79.36%**	**27.36%**
Av. Barbosa du Bocage	0.00%	0.00%	0.00%	0.00%	79.54%	79.49%	**79.02%**	**26.50%**
Av. Visconde de Valmor	1.60%	1.60%	3.33%	3.33%	79.17%	79.16%	**78.34%**	**28.03%**
Av. Defensores de Chaves	6.39%	7.50%	2.22%	4.00%	79.35%	79.04%	**78.35%**	**29.65%**
Av. Marquês de Tomar	2.86%	2.86%	0.00%	0.00%	79.34%	79.40%	**78.70%**	**27.41%**
Av. Miguel Bombarda	10.00%	10.00%	0.00%	0.00%	79.73%	79.71%	**79.43%**	**29.91%**
Av. de Berna	0.00%	0.00%	0.00%	0.00%	79.65%	79.63%	**79.29%**	**26.55%**
Av. João Crisóstomo	0.00%	0.00%	0.00%	0.00%	79.72%	79.60%	**79.32%**	**26.55%**
R. Arco do Cego	0.00%	0.00%	0.00%	0.00%	79.63%	79.61%	**79.24%**	**26.54%**
R. de Entrecampos	0.00%	0.00%	0.00%	0.00%	79.81%	79.73%	**79.53%**	**26.59%**
R. Marquês Sá da Bandeira	0.00%	0.00%	0.00%	0.00%	79.51%	79.49%	**78.99%**	**26.50%**
R. D. Filipa de Vilhena	0.00%	0.00%	0.00%	0.00%	79.69%	79.68%	**79.37%**	**26.56%**
Av. Duque de Avila	0.00%	0.00%	0.00%	0.00%	79.36%	79.41%	**78.78%**	**26.46%**
Av. Poeta Mistral	4.35%	5.36%	0.00%	0.00%	79.59%	79.48%	**79.07%**	**28.11%**
R. Chaby Pinheiro	0.00%	0.00%	0.00%	0.00%	79.65%	79.65%	**79.30%**	**26.55%**
Av. Conde Valbom	0.00%	0.00%	0.00%	0.00%	79.75%	79.68%	**79.42%**	**26.57%**
Av. Ant. José de Almeida	0.00%	0.00%	0.00%	0.00%	79.60%	79.57%	**79.17%**	**26.53%**
Campo Pequeno	0.00%	0.00%	0.00%	0.00%	79.73%	79.69%	**79.43%**	**26.57%**
R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.91%	79.88%	**79.79%**	**26.63%**
Largo Azevedo Perdigão	0.00%	0.00%	0.00%	0.00%	79.74%	79.73%	**79.46%**	**26.58%**
Impasse à R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.74%	79.70%	**79.44%**	**26.57%**

Results of Prediction Based On The Average Values of the Weeks Before

Street	Precision Full	Recall Full	Precision Almost Full	Recall Almost Full	Precision Vacant	Recall Vacant	Accuracy	Average F1-Score
Av. Elias Garcia	0.00%	0.00%	0.00%	0.00%	79.54%	79.76%	**79.30%**	**26.55%**
Av. 5 de Outubro	0.00%	0.00%	0.00%	0.00%	79.25%	79.98%	**79.23%**	**26.54%**
Av. da República	0.00%	0.00%	5.56%	5.00%	79.77%	79.86%	**79.63%**	**28.36%**
Av. Barbosa du Bocage	0.00%	0.00%	0.00%	0.00%	79.53%	79.95%	**79.48%**	**26.58%**
Av. Visconde de Valmor	0.00%	0.00%	0.00%	0.00%	79.14%	79.98%	**79.11%**	**26.52%**
Av. Defensores de Chaves	4.88%	4.58%	3.33%	4.00%	79.31%	79.40%	**78.68%**	**29.24%**
Av. Marquês de Tomar	0.00%	0.00%	0.00%	0.00%	79.28%	80.00%	**79.28%**	**26.55%**
Av. Miguel Bombarda	0.00%	0.00%	0.00%	0.00%	79.70%	79.98%	**79.67%**	**26.61%**
Av. de Berna	0.00%	0.00%	0.00%	0.00%	79.65%	79.99%	**79.64%**	**26.61%**
Av. João Crisóstomo	0.00%	0.00%	0.00%	0.00%	79.72%	79.87%	**79.59%**	**26.60%**
R. Arco do Cego	0.00%	0.00%	0.00%	0.00%	79.63%	79.98%	**79.61%**	**26.60%**
R. de Entrecampos	0.00%	0.00%	0.00%	0.00%	79.81%	79.92%	**79.73%**	**26.62%**
R. Marquês Sá da Bandeira	0.00%	0.00%	0.00%	0.00%	79.50%	79.98%	**79.48%**	**26.58%**
R. D. Filipa de Vilhena	0.00%	0.00%	0.00%	0.00%	79.69%	79.99%	**79.68%**	**26.61%**
Av. Duque de Avila	0.00%	0.00%	0.00%	0.00%	79.37%	79.99%	**79.36%**	**26.56%**
Av. Poeta Mistral	5.00%	2.86%	0.00%	0.00%	79.57%	79.90%	**79.45%**	**27.79%**
R. Chaby Pinheiro	0.00%	0.00%	0.00%	0.00%	79.65%	80.00%	**79.65%**	**26.61%**
Av. Conde Valbom	0.00%	0.00%	0.00%	0.00%	79.74%	79.93%	**79.67%**	**26.61%**
Av. Ant. José de Almeida	0.00%	0.00%	0.00%	0.00%	79.60%	79.97%	**79.57%**	**26.59%**
Campo Pequeno	0.00%	0.00%	0.00%	0.00%	79.74%	79.94%	**79.68%**	**26.61%**
R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.91%	79.97%	**79.88%**	**26.65%**
Largo Azevedo Perdigão	0.00%	0.00%	0.00%	0.00%	79.74%	79.99%	**79.73%**	**26.62%**
Impasse à R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.74%	79.97%	**79.71%**	**26.62%**

Results of Prediction Based On The Average Values of the Weeks Before Without Outliers								
Street	**Precision Full**	**Recall Full**	**Precision Almost Full**	**Recall Almost Full**	**Precision Vacant**	**Recall Vacant**	**Accuracy**	**Average F1-Score**
Av. Elias Garcia	0.00%	0.00%	0.00%	0.00%	79.54%	79.76%	**79.30%**	**26.55%**
Av. 5 de Outubro	0.00%	0.00%	0.00%	0.00%	79.25%	79.98%	**79.23%**	**26.54%**
Av. da República	0.00%	0.00%	5.56%	5.00%	79.77%	79.86%	**79.63%**	**28.36%**
Av. Barbosa du Bocage	0.00%	0.00%	0.00%	0.00%	79.53%	79.95%	**79.48%**	**26.58%**
Av. Visconde de Valmor	0.00%	0.00%	0.00%	0.00%	79.14%	79.98%	**79.11%**	**26.52%**
Av. Defensores de Chaves	4.88%	4.58%	3.33%	4.00%	79.31%	79.40%	**78.68%**	**29.24%**
Av. Marquês de Tomar	0.00%	0.00%	0.00%	0.00%	79.28%	79.98%	**79.25%**	**26.54%**
Av. Miguel Bombarda	0.00%	0.00%	0.00%	0.00%	79.70%	79.98%	**79.67%**	**26.61%**
Av. de Berna	0.00%	0.00%	0.00%	0.00%	79.65%	79.99%	**79.64%**	**26.61%**
Av. João Crisóstomo	0.00%	0.00%	0.00%	0.00%	79.72%	80.00%	**79.72%**	**26.62%**
R. Arco do Cego	0.00%	0.00%	0.00%	0.00%	79.63%	79.98%	**79.61%**	**26.60%**
R. de Entrecampos	0.00%	0.00%	0.00%	0.00%	79.81%	79.92%	**79.73%**	**26.62%**
R. Marquês Sá da Bandeira	0.00%	0.00%	0.00%	0.00%	79.50%	80.00%	**79.50%**	**26.58%**
R. D. Filipa de Vilhena	0.00%	0.00%	0.00%	0.00%	79.69%	79.99%	**79.68%**	**26.61%**
Av. Duque de Avila	0.00%	0.00%	0.00%	0.00%	79.37%	79.99%	**79.36%**	**26.56%**
Av. Poeta Mistral	5.00%	2.86%	0.00%	0.00%	79.57%	79.90%	**79.45%**	**27.79%**
R. Chaby Pinheiro	0.00%	0.00%	0.00%	0.00%	79.65%	80.00%	**79.65%**	**26.61%**
Av. Conde Valbom	0.00%	0.00%	0.00%	0.00%	79.74%	79.94%	**79.68%**	**26.61%**
Av. Ant. José de Almeida	0.00%	0.00%	0.00%	0.00%	79.60%	79.98%	**79.58%**	**26.60%**
Campo Pequeno	0.00%	0.00%	0.00%	0.00%	79.74%	79.97%	**79.70%**	**26.62%**
R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.91%	80.00%	**79.91%**	**26.65%**
Largo Azevedo Perdigão	0.00%	0.00%	0.00%	0.00%	79.74%	79.99%	**79.73%**	**26.62%**
Impasse à R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.74%	79.98%	**79.72%**	**26.62%**

Results of Prediction Based On The Class With Higher Frequency on The Weeks Before

Street	Precision Full	Recall Full	Precision Almost Full	Recall Almost Full	Precision Vacant	Recall Vacant	Accuracy	Average F1-Score
Av. Elias Garcia	0.00%	0.00%	0.00%	0.00%	79.53%	79.46%	**79.00%**	**26.50%**
Av. 5 de Outubro	0.00%	0.00%	0.00%	0.00%	79.24%	79.38%	**78.64%**	**26.44%**
Av. da República	0.00%	0.00%	1.82%	5.00%	79.77%	79.64%	**79.41%**	**27.46%**
Av. Barbosa du Bocage	0.00%	0.00%	0.00%	0.00%	79.54%	79.49%	**79.02%**	**26.50%**
Av. Visconde de Valmor	1.60%	1.60%	3.33%	3.33%	79.17%	79.16%	**78.34%**	**28.03%**
Av. Defensores de Chaves	6.39%	7.50%	2.22%	4.00%	79.35%	79.04%	**78.35%**	**29.65%**
Av. Marquês de Tomar	2.86%	2.86%	0.00%	0.00%	79.34%	79.40%	**78.70%**	**27.41%**
Av. Miguel Bombarda	10.00%	10.00%	0.00%	0.00%	79.73%	79.71%	**79.43%**	**29.91%**
Av. de Berna	0.00%	0.00%	0.00%	0.00%	79.65%	79.63%	**79.29%**	**26.55%**
Av. João Crisóstomo	0.00%	0.00%	0.00%	0.00%	79.72%	79.60%	**79.32%**	**26.55%**
R. Arco do Cego	0.00%	0.00%	0.00%	0.00%	79.63%	79.61%	**79.24%**	**26.54%**
R. de Entrecampos	0.00%	0.00%	0.00%	0.00%	79.81%	79.73%	**79.53%**	**26.59%**
R. Marquês Sá da Bandeira	0.00%	0.00%	0.00%	0.00%	79.51%	79.49%	**78.99%**	**26.50%**
R. D. Filipa de Vilhena	0.00%	0.00%	0.00%	0.00%	79.69%	79.68%	**79.37%**	**26.56%**
Av. Duque de Avila	0.00%	0.00%	0.00%	0.00%	79.36%	79.41%	**78.78%**	**26.46%**
Av. Poeta Mistral	4.35%	5.36%	0.00%	0.00%	79.59%	79.48%	**79.07%**	**28.11%**
R. Chaby Pinheiro	0.00%	0.00%	0.00%	0.00%	79.65%	79.65%	**79.30%**	**26.55%**
Av. Conde Valbom	0.00%	0.00%	0.00%	0.00%	79.75%	79.68%	**79.42%**	**26.57%**
Av. Ant. José de Almeida	0.00%	0.00%	0.00%	0.00%	79.60%	79.57%	**79.17%**	**26.53%**
Campo Pequeno	0.00%	0.00%	0.00%	0.00%	79.73%	79.69%	**79.43%**	**26.57%**
R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.91%	79.88%	**79.79%**	**26.63%**
Largo Azevedo Perdigão	0.00%	0.00%	0.00%	0.00%	79.74%	79.73%	**79.46%**	**26.58%**
Impasse à R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.74%	79.70%	**79.44%**	**26.57%**

Results of Prediction Based On The Class With Higher Frequency on The Weeks Before Without Outliers

Street	Precision Full	Recall Full	Precision Almost Full	Recall Almost Full	Precision Vacant	Recall Vacant	Accuracy	Average F1-Score
Av. Elias Garcia	0.00%	0.00%	0.00%	0.00%	79.53%	79.43%	**78.98%**	**26.49%**
Av. 5 de Outubro	0.00%	0.00%	0.00%	0.00%	79.26%	79.40%	**78.66%**	**26.44%**
Av. da República	0.00%	0.00%	1.82%	5.00%	79.77%	79.64%	**79.41%**	**27.46%**
Av. Barbosa du Bocage	0.00%	0.00%	0.00%	0.00%	79.54%	79.48%	**79.01%**	**26.50%**
Av. Visconde de Valmor	1.60%	1.60%	3.33%	3.33%	79.17%	79.16%	**78.34%**	**28.03%**
Av. Defensores de Chaves	6.39%	7.50%	2.00%	4.00%	79.35%	79.03%	**78.34%**	**29.59%**
Av. Marquês de Tomar	2.86%	2.86%	0.00%	0.00%	79.34%	79.44%	**78.75%**	**27.42%**
Av. Miguel Bombarda	10.00%	10.00%	0.00%	0.00%	79.73%	79.76%	**79.48%**	**29.92%**
Av. de Berna	0.00%	0.00%	0.00%	0.00%	79.65%	79.65%	**79.30%**	**26.55%**
Av. João Crisóstomo	0.00%	0.00%	0.00%	0.00%	79.72%	79.74%	**79.47%**	**26.58%**
R. Arco do Cego	0.00%	0.00%	0.00%	0.00%	79.63%	79.63%	**79.26%**	**26.54%**
R. de Entrecampos	0.00%	0.00%	0.00%	0.00%	79.81%	79.74%	**79.55%**	**26.59%**
R. Marquês Sá da Bandeira	0.00%	0.00%	0.00%	0.00%	79.51%	79.64%	**79.14%**	**26.52%**
R. D. Filipa de Vilhena	0.00%	0.00%	0.00%	0.00%	79.69%	79.73%	**79.42%**	**26.57%**
Av. Duque de Avila	0.00%	0.00%	0.00%	0.00%	79.36%	79.51%	**78.88%**	**26.48%**
Av. Poeta Mistral	4.35%	5.36%	0.00%	0.00%	79.59%	79.48%	**79.07%**	**28.11%**
R. Chaby Pinheiro	0.00%	0.00%	0.00%	0.00%	79.65%	79.76%	**79.42%**	**26.57%**
Av. Conde Valbom	0.00%	0.00%	0.00%	0.00%	79.75%	79.75%	**79.49%**	**26.58%**
Av. Ant. José de Almeida	0.00%	0.00%	0.00%	0.00%	79.60%	79.60%	**79.20%**	**26.53%**
Campo Pequeno	0.00%	0.00%	0.00%	0.00%	79.73%	79.81%	**79.54%**	**26.59%**
R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.91%	79.96%	**79.87%**	**26.64%**
Largo Azevedo Perdigão	0.00%	0.00%	0.00%	0.00%	79.74%	79.83%	**79.56%**	**26.59%**
Impasse à R. Eiffel	0.00%	0.00%	0.00%	0.00%	79.74%	79.81%	**79.55%**	**26.59%**

Appendix N – Classified Instances by Street

All the presented tables are in descent order by the occupancy of the street.

Street	**High Demand Period 10%Torelance**		**Classified Full 10% Rotation**		**Classified Almost Full 20% Rotation**		**Class Balance**
	Instances	% of data	Instances	% of data	Instances	% of data	
Av. Elias Garcia	173	1.99%	11	0.13%	119	1.37%	**-166.15%**
Av. 5 de Outubro	111	2.55%	13	0.30%	102	2.34%	**-154.78%**
Av. da República	110	1.26%	14	0.16%	78	0.90%	**-139.13%**
Av. Barbosa du Bocage	94	1.08%	40	0.46%	81	0.93%	**-67.77%**
Av. Visconde de Valmor	140	1.61%	59	0.68%	105	1.21%	**-56.10%**
Av. Defensores de Chaves	84	1.93%	39	0.90%	95	2.18%	**-83.58%**
Av. Marquês de Tomar	101	2.32%	22	0.51%	71	1.63%	**-105.38%**
Av. Miguel Bombarda	186	2.13%	25	0.29%	59	0.68%	**-80.95%**
Av. de Berna	33	0.76%	10	0.23%	43	0.99%	**-124.53%**
Av. João Crisóstomo	96	2.20%	17	0.39%	29	0.67%	**-52.17%**
R. Arco do Cego	50	1.15%	21	0.48%	35	0.80%	**-50.00%**
R. de Entrecampos	59	1.35%	14	0.32%	25	0.57%	**-56.41%**
R. Marquês Sá da Bandeira	139	3.19%	29	0.67%	26	0.60%	**10.91%**
R. D. Filipa de Vilhena	101	2.32%	17	0.39%	18	0.41%	**-5.71%**
Av. Duque de Avila	56	1.29%	27	0.62%	50	1.15%	**-59.74%**
Av. Poeta Mistral	85	1.95%	31	0.71%	33	0.76%	**-6.25%**
R. Chaby Pinheiro	98	2.25%	18	0.41%	26	0.60%	**-36.36%**
Av. Conde Valbom	211	4.84%	17	0.39%	24	0.55%	**-34.15%**
Av. Ant. José de Almeida	85	1.95%	29	0.67%	18	0.41%	**46.81%**
Campo Pequeno	74	1.70%	27	0.62%	17	0.39%	**45.45%**
R. Eiffel	1240	28.47%	11	0.25%	11	0.25%	**0.00%**
Largo Azevedo Perdigão	87	2.00%	20	0.46%	19	0.44%	**5.13%**
Impasse à R. Eiffel	240	5.51%	28	0.64%	17	0.39%	**48.89%**
FINAL RESULT	**3653**	**3.00%**	**539**	**0.44%**	**1101**	**0.90%**	**-68.54%**

Street	High Demand Period 10%Torelance		Classified Full 10% Rotation		Classified Almost Full 15% Rotation		Class Balance
	Instances	% of data	Instances	% of data	Instances	% of data	
Av. Elias Garcia	173	1.99%	11	0.13%	68	0.78%	**-144.30%**
Av. 5 de Outubro	111	2.55%	13	0.30%	52	1.19%	**-120.00%**
Av. da República	110	1.26%	14	0.16%	36	0.41%	**-88.00%**
Av. Barbosa du Bocage	94	1.08%	40	0.46%	36	0.41%	**10.53%**
Av. Visconde de Valmor	140	1.61%	59	0.68%	58	0.67%	**1.71%**
Av. Defensores de Chaves	84	1.93%	39	0.90%	58	1.33%	**-39.18%**
Av. Marquês de Tomar	101	2.32%	22	0.51%	37	0.85%	**-50.85%**
Av. Miguel Bombarda	186	2.13%	25	0.29%	42	0.48%	**-50.75%**
Av. de Berna	33	0.76%	10	0.23%	13	0.30%	**-26.09%**
Av. João Crisóstomo	96	2.20%	17	0.39%	15	0.34%	**12.50%**
R. Arco do Cego	50	1.15%	21	0.48%	19	0.44%	**10.00%**
R. de Entrecampos	59	1.35%	14	0.32%	17	0.39%	**-19.35%**
R. Marquês Sá da Bandeira	139	3.19%	29	0.67%	14	0.32%	**69.77%**
R. D. Filipa de Vilhena	101	2.32%	17	0.39%	10	0.23%	**51.85%**
Av. Duque de Avila	56	1.29%	27	0.62%	27	0.62%	**0.00%**
Av. Poeta Mistral	85	1.95%	31	0.71%	15	0.34%	**69.57%**
R. Chaby Pinheiro	98	2.25%	18	0.41%	15	0.34%	**18.18%**
Av. Conde Valbom	211	4.84%	17	0.39%	11	0.25%	**42.86%**
Av. Ant. José de Almeida	85	1.95%	29	0.67%	1	0.02%	**186.67%**
Campo Pequeno	74	1.70%	27	0.62%	5	0.11%	**137.50%**
R. Eiffel	1240	28.47%	11	0.25%	3	0.07%	**114.29%**
Largo Azevedo Perdigão	87	2.00%	20	0.46%	5	0.11%	**120.00%**
Impasse à R. Eiffel	240	5.51%	28	0.64%	0	0.00%	**200.00%**
FINAL RESULT	**3653**	**3.00%**	**539**	**0.44%**	**557**	**0.46%**	**-3.28%**

Street	High Demand Period 10%Torelance		Classified Full 15% Rotation		Classified Almost Full 20% Rotation		Class Balance
	Instances	% of data	Instances	% of data	Instances	% of data	
Av. Elias Garcia	173	1.99%	79	0.91%	51	0.59%	**43.08%**
Av. 5 de Outubro	111	2.55%	65	1.49%	50	1.15%	**26.09%**
Av. da República	110	1.26%	50	0.57%	42	0.48%	**17.39%**
Av. Barbosa du Bocage	94	1.08%	76	0.87%	45	0.52%	**51.24%**
Av. Visconde de Valmor	140	1.61%	117	1.34%	47	0.54%	**85.37%**
Av. Defensores de Chaves	84	1.93%	97	2.23%	37	0.85%	**89.55%**
Av. Marquês de Tomar	101	2.32%	59	1.35%	34	0.78%	**53.76%**
Av. Miguel Bombarda	186	2.13%	67	0.77%	17	0.20%	**119.05%**
Av. de Berna	33	0.76%	23	0.53%	30	0.69%	**-26.42%**
Av. João Crisóstomo	96	2.20%	32	0.73%	14	0.32%	**78.26%**
R. Arco do Cego	50	1.15%	40	0.92%	16	0.37%	**85.71%**
R. de Entrecampos	59	1.35%	31	0.71%	8	0.18%	**117.95%**
R. Marquês Sá da Bandeira	139	3.19%	43	0.99%	12	0.28%	**112.73%**
R. D. Filipa de Vilhena	101	2.32%	27	0.62%	8	0.18%	**108.57%**
Av. Duque de Avila	56	1.29%	54	1.24%	23	0.53%	**80.52%**
Av. Poeta Mistral	85	1.95%	46	1.06%	18	0.41%	**87.50%**
R. Chaby Pinheiro	98	2.25%	33	0.76%	11	0.25%	**100.00%**
Av. Conde Valbom	211	4.84%	28	0.64%	13	0.30%	**73.17%**
Av. Ant. José de Almeida	85	1.95%	30	0.69%	17	0.39%	**55.32%**
Campo Pequeno	74	1.70%	32	0.73%	12	0.28%	**90.91%**
R. Eiffel	1240	28.47%	14	0.32%	8	0.18%	**54.55%**
Largo Azevedo Perdigão	87	2.00%	25	0.57%	14	0.32%	**56.41%**
Impasse à R. Eiffel	240	5.51%	28	0.64%	17	0.39%	**48.89%**
FINAL RESULT	3653	3.00%	1096	0.90%	544	0.45%	67.32%

Street	High Demand Period 15%Torelance		Classified Full 10% Rotation		Classified Almost Full 20% Rotation		Class Balance
	Instances	% of data	Instances	% of data	Instances	% of data	
Av. Elias Garcia	226	2.59%	38	0.44%	220	2.53%	**-141.09%**
Av. 5 de Outubro	157	3.60%	30	0.69%	204	4.68%	**-148.72%**
Av. da República	153	1.76%	23	0.26%	124	1.42%	**-137.41%**
Av. Barbosa du Bocage	136	1.56%	71	0.81%	150	1.72%	**-71.49%**
Av. Visconde de Valmor	232	2.66%	90	1.03%	183	2.10%	**-68.13%**
Av. Defensores de Chaves	148	3.40%	78	1.79%	207	4.75%	**-90.53%**
Av. Marquês de Tomar	130	2.98%	52	1.19%	125	2.87%	**-82.49%**
Av. Miguel Bombarda	219	2.51%	52	0.60%	116	1.33%	**-76.19%**
Av. de Berna	108	2.48%	20	0.46%	81	1.86%	**-120.79%**
Av. João Crisóstomo	109	2.50%	34	0.78%	46	1.06%	**-30.00%**
R. Arco do Cego	84	1.93%	53	1.22%	84	1.93%	**-45.26%**
R. de Entrecampos	87	2.00%	29	0.67%	44	1.01%	**-41.10%**
R. Marquês Sá da Bandeira	141	3.24%	31	0.71%	29	0.67%	**6.67%**
R. D. Filipa de Vilhena	116	2.66%	38	0.87%	28	0.64%	**30.30%**
Av. Duque de Avila	72	1.65%	46	1.06%	72	1.65%	**-44.07%**
Av. Poeta Mistral	94	2.16%	40	0.92%	47	1.08%	**-16.09%**
R. Chaby Pinheiro	118	2.71%	50	1.15%	45	1.03%	**10.53%**
Av. Conde Valbom	236	5.42%	47	1.08%	45	1.03%	**4.35%**
Av. Ant. José de Almeida	87	2.00%	32	0.73%	20	0.46%	**46.15%**
Campo Pequeno	82	1.88%	39	0.90%	23	0.53%	**51.61%**
R. Eiffel	1244	28.56%	13	0.30%	17	0.39%	**-26.67%**
Largo Azevedo Perdigão	87	2.00%	20	0.46%	19	0.44%	**5.13%**
Impasse à R. Eiffel	240	5.51%	28	0.64%	17	0.39%	**48.89%**
FINAL RESULT	**4306**	**3.53%**	**954**	**0.78%**	**1946**	**1.60%**	**-68.41%**

Street	High Demand Period 15%Torelance		Classified Full 10% Rotation		Classified Almost Full 15% Rotation		Class Balance
	Instances	% of data	Instances	% of data	Instances	% of data	
Av. Elias Garcia	226	2.59%	38	0.44%	119	1.37%	**-103.18%**
Av. 5 de Outubro	157	3.60%	30	0.69%	111	2.55%	**-114.89%**
Av. da República	153	1.76%	23	0.26%	52	0.60%	**-77.33%**
Av. Barbosa du Bocage	136	1.56%	71	0.81%	74	0.85%	**-4.14%**
Av. Visconde de Valmor	232	2.66%	90	1.03%	105	1.21%	**-15.38%**
Av. Defensores de Chaves	148	3.40%	78	1.79%	132	3.03%	**-51.43%**
Av. Marquês de Tomar	130	2.98%	52	1.19%	70	1.61%	**-29.51%**
Av. Miguel Bombarda	219	2.51%	52	0.60%	70	0.80%	**-29.51%**
Av. de Berna	108	2.48%	20	0.46%	33	0.76%	**-49.06%**
Av. João Crisóstomo	109	2.50%	34	0.78%	23	0.53%	**38.60%**
R. Arco do Cego	84	1.93%	53	1.22%	48	1.10%	**9.90%**
R. de Entrecampos	87	2.00%	29	0.67%	31	0.71%	**-6.67%**
R. Marquês Sá da Bandeira	141	3.24%	31	0.71%	17	0.39%	**58.33%**
R. D. Filipa de Vilhena	116	2.66%	38	0.87%	10	0.23%	**116.67%**
Av. Duque de Avila	72	1.65%	46	1.06%	38	0.87%	**19.05%**
Av. Poeta Mistral	94	2.16%	40	0.92%	22	0.51%	**58.06%**
R. Chaby Pinheiro	118	2.71%	50	1.15%	30	0.69%	**50.00%**
Av. Conde Valbom	236	5.42%	47	1.08%	12	0.28%	**118.64%**
Av. Ant. José de Almeida	87	2.00%	32	0.73%	1	0.02%	**187.88%**
Campo Pequeno	82	1.88%	39	0.90%	5	0.11%	**154.55%**
R. Eiffel	1244	28.56%	13	0.30%	4	0.09%	**105.88%**
Largo Azevedo Perdigão	87	2.00%	20	0.46%	5	0.11%	**120.00%**
Impasse à R. Eiffel	240	5.51%	28	0.64%	0	0.00%	**200.00%**
FINAL RESULT	**4260**	**3.49%**	**954**	**0.78%**	**1012**	**0.83%**	**-5.90%**

Street	High Demand Period 15%Torelance		Classified Full 15% Rotation		Classified Almost Full 20% Rotation		Class Balance
	Instances	% of data	Instances	% of data	Instances	% of data	
Av. Elias Garcia	226	2.59%	157	1.80%	101	1.16%	**43.41%**
Av. 5 de Outubro	157	3.60%	141	3.24%	93	2.13%	**41.03%**
Av. da República	153	1.76%	75	0.86%	72	0.83%	**4.08%**
Av. Barbosa du Bocage	136	1.56%	145	1.66%	76	0.87%	**62.44%**
Av. Visconde de Valmor	232	2.66%	195	2.24%	78	0.90%	**85.71%**
Av. Defensores de Chaves	148	3.40%	210	4.82%	75	1.72%	**94.74%**
Av. Marquês de Tomar	130	2.98%	122	2.80%	55	1.26%	**75.71%**
Av. Miguel Bombarda	219	2.51%	122	1.40%	46	0.53%	**90.48%**
Av. de Berna	108	2.48%	53	1.22%	48	1.10%	**9.90%**
Av. João Crisóstomo	109	2.50%	57	1.31%	23	0.53%	**85.00%**
R. Arco do Cego	84	1.93%	101	2.32%	36	0.83%	**94.89%**
R. de Entrecampos	87	2.00%	60	1.38%	13	0.30%	**128.77%**
R. Marquês Sá da Bandeira	141	3.24%	48	1.10%	12	0.28%	**120.00%**
R. D. Filipa de Vilhena	116	2.66%	48	1.10%	18	0.41%	**90.91%**
Av. Duque de Avila	72	1.65%	84	1.93%	34	0.78%	**84.75%**
Av. Poeta Mistral	94	2.16%	62	1.42%	25	0.57%	**85.06%**
R. Chaby Pinheiro	118	2.71%	80	1.84%	15	0.34%	**136.84%**
Av. Conde Valbom	236	5.42%	59	1.35%	33	0.76%	**56.52%**
Av. Ant. José de Almeida	87	2.00%	33	0.76%	19	0.44%	**53.85%**
Campo Pequeno	82	1.88%	44	1.01%	18	0.41%	**83.87%**
R. Eiffel	1244	28.56%	17	0.39%	13	0.30%	**26.67%**
Largo Azevedo Perdigão	87	2.00%	25	0.57%	14	0.32%	**56.41%**
Impasse à R. Eiffel	240	5.51%	28	0.64%	17	0.39%	**48.89%**
FINAL RESULT	**4306**	**3.53%**	**1966**	**1.61%**	**934**	**0.77%**	**71.17%**

Appendix O – Final Dataset Classes Distribution

FULL DATASET - 3 CLASSES	Full Instances	Almost Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	954	1012	3268	5234	75.63%
Undersample 40% Vacant	954	1012	2111	4077	47.97%
SMOTE Oversample - 100% Vacant	1908	2024	3268	7200	31.41%
SMOTE Oversample - 40% Vacant	1908	2024	2111	6043	5.06%

FULL DATASET - 2 CLASSES	Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	1966	3268	5234	35.18%
Undersample 40% Vacant	1966	2111	4077	5.03%

ONLY ZONE 1 - 3 CLASSES	Full Instances	Almost Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	454	621	1709	2784	73.44%
Undersample 40% Vacant	454	621	1070	2145	44.56%
SMOTE Oversample - 100% Vacant	971	1354	1709	4034	27.45%
SMOTE Oversample - 40% Vacant	971	1356	1070	3397	17.66%

ONLY ZONE 16 - 3 CLASSES	Full Instances	Almost Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	500	391	1559	2450	79.00%
Undersample 40% Vacant	500	391	1041	1932	54.05%
SMOTE Oversample - 100% Vacant	937	670	1559	3166	43.22%
SMOTE Oversample - 40% Vacant	937	668	1041	2646	21.82%

ONLY MOST OCCUPIED STREETS- 3 CLASSES	Full Instances	Almost Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	382	663	1523	2568	69.45%
Undersample 40% Vacant	382	663	900	1945	40.00%
SMOTE Oversample - 100% Vacant	835	1419	1523	3777	29.46%
SMOTE Oversample - 40% Vacant	835	1418	900	3153	30.40%

ONLY LESS OCCUPIED STREETS- 3 CLASSES	Full Instances	Almost Full Instances	Vacant Instances	Total Instances	Class Balance
Undersample 100% Vacant	572	349	1745	2666	84.39%
Undersample 40% Vacant	572	349	1211	2132	62.96%
SMOTE Oversample - 100% Vacant	1073	605	1745	3423	50.22%
SMOTE Oversample - 40% Vacant	1073	606	1211	2890	32.91%

Appendix P – Algorithm Models Result - Default Parameters

Results of the prediction models ordered by the number of correct instances.

3 Classes – Dataset with All Streets

DATASET	ALGORITHM	TRAIN	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
			Precision	Recall	Precision	Recall	Precision	Recall		
SMOTE Oversample - 100% Vacant	RandomForest	65%	62.80%	66.80%	66.00%	65.60%	81.10%	78.50%	71.83%	71.90%
SMOTE Oversample - 100% Vacant	RandomForest	80%	61.00%	67.30%	65.40%	65.80%	82.50%	77.60%	71.60%	71.80%
SMOTE Oversample - 40% Vacant	RandomForest	65%	66.60%	71.00%	68.60%	69.20%	78.00%	72.80%	71.02%	71.10%
SMOTE Oversample - 40% Vacant	RandomForest	80%	65.70%	71.80%	68.50%	68.70%	78.20%	71.50%	70.64%	70.70%
SMOTE Oversample - 40% Vacant	MLP	65%	66.00%	68.90%	64.80%	69.30%	73.80%	65.90%	67.99%	68.00%
SMOTE Oversample - 40% Vacant	MLP	80%	65.70%	65.10%	63.20%	71.60%	74.80%	65.60%	67.49%	67.60%
SMOTE Oversample - 40% Vacant	J48	80%	61.50%	72.70%	69.20%	68.20%	72.70%	62.00%	67.41%	67.40%
SMOTE Oversample - 100% Vacant	J48	80%	58.00%	63.80%	63.30%	62.80%	75.90%	72.00%	67.29%	67.40%
SMOTE Oversample - 100% Vacant	MLP	80%	56.80%	68.10%	61.50%	63.80%	79.50%	68.80%	67.22%	67.60%
SMOTE Oversample - 40% Vacant	J48	65%	61.00%	68.90%	66.80%	69.90%	73.10%	61.40%	66.62%	66.60%
Undersample 100% Vacant	RandomForest	80%	42.40%	35.50%	40.90%	41.70%	79.70%	83.10%	66.28%	65.70%
SMOTE Oversample - 100% Vacant	MLP	65%	58.80%	62.20%	64.80%	56.00%	70.90%	74.70%	66.15%	66.00%
SMOTE Oversample - 100% Vacant	J48	65%	56.50%	60.30%	62.70%	64.50%	73.60%	69.50%	65.75%	65.90%
Undersample 100% Vacant	RandomForest	65%	41.30%	33.00%	40.10%	38.20%	77.90%	83.90%	65.34%	64.30%
SMOTE Oversample - 40% Vacant	REPTree	80%	57.40%	69.40%	65.80%	66.70%	72.40%	58.00%	64.52%	64.60%
Undersample 100% Vacant	REPTree	80%	37.10%	13.20%	43.90%	34.20%	69.70%	88.00%	63.71%	59.30%
Undersample 100% Vacant	J48	80%	39.80%	18.80%	41.30%	31.20%	70.00%	86.50%	63.23%	59.70%
Undersample 100% Vacant	J48	65%	38.10%	17.30%	44.00%	30.90%	68.50%	86.50%	62.50%	58.60%
SMOTE Oversample - 100% Vacant	REPTree	80%	53.50%	57.60%	59.30%	59.10%	70.00%	67.20%	62.43%	62.60%
SMOTE Oversample - 40% Vacant	REPTree	65%	56.30%	66.60%	63.00%	64.50%	69.50%	56.70%	62.41%	62.40%
Undersample 100% Vacant	REPTree	65%	33.00%	9.10%	45.00%	28.00%	66.70%	89.60%	62.28%	56.40%
Undersample 40% Vacant	RandomForest	80%	43.70%	43.70%	40.90%	42.60%	78.90%	77.50%	61.47%	61.70%
SMOTE Oversample - 100% Vacant	REPTree	65%	50.10%	59.90%	60.80%	59.10%	69.70%	63.50%	61.31%	61.60%
Undersample 100% Vacant	MLP	80%	31.40%	29.90%	36.50%	33.20%	73.90%	77.00%	59.79%	59.20%
Undersample 40% Vacant	RandomForest	65%	40.70%	34.90%	43.50%	45.30%	74.20%	77.50%	59.50%	59.00%
Undersample 100% Vacant	MLP	65%	34.00%	27.80%	36.80%	23.20%	68.70%	80.50%	59.33%	57.00%
Undersample 40% Vacant	MLP	80%	39.90%	52.10%	44.40%	35.30%	74.30%	71.00%	58.28%	58.50%
Undersample 40% Vacant	REPTree	80%	41.10%	31.60%	46.80%	46.30%	66.30%	73.30%	57.30%	56.40%
Undersample 40% Vacant	J48	80%	41.60%	33.70%	39.20%	52.60%	72.20%	67.40%	56.07%	56.30%
Undersample 40% Vacant	J48	65%	35.80%	26.50%	42.70%	43.60%	64.70%	71.60%	54.10%	53.00%
Undersample 40% Vacant	MLP	65%	33.00%	34.60%	41.00%	45.80%	71.90%	66.20%	53.75%	54.30%
Undersample 40% Vacant	REPTree	65%	31.60%	28.30%	42.00%	39.70%	64.30%	69.10%	52.21%	51.60%

2 Classes – Dataset with All Streets

DATASET	ALGORITHM	TRAIN	FULL		VACANT		Correct Instances	Weighted Average F1-Score
			Precision	Recall	Precision	Recall		
Undersample 100% Vacant	RandomForest	80%	67.30%	66.90%	79.90%	80.20%	75.17%	75.20%
Undersample 100% Vacant	RandomForest	65%	66.90%	64.40%	78.20%	80.00%	74.02%	73.90%
Undersample 40% Vacant	RandomForest	80%	68.10%	80.80%	79.90%	66.90%	73.37%	73.30%
Undersample 40% Vacant	RandomForest	65%	70.50%	76.50%	76.10%	70.00%	73.16%	73.20%
Undersample 100% Vacant	MLP	80%	61.00%	62.40%	76.80%	75.70%	70.68%	70.70%
Undersample 40% Vacant	J48	80%	65.50%	76.60%	76.00%	64.80%	70.31%	70.30%
Undersample 40% Vacant	MLP	65%	66.40%	75.50%	73.70%	64.20%	69.66%	69.60%
Undersample 40% Vacant	MLP	80%	65.60%	73.20%	73.90%	66.40%	69.57%	69.60%
Undersample 100% Vacant	REPTree	80%	60.40%	47.70%	71.80%	81.00%	68.39%	67.50%
Undersample 40% Vacant	J48	65%	66.10%	70.60%	70.60%	66.10%	68.26%	68.30%
Undersample 100% Vacant	J48	65%	60.30%	51.10%	72.10%	79.00%	68.23%	67.60%
Undersample 100% Vacant	REPTree	65%	59.80%	46.40%	70.60%	80.50%	67.36%	66.40%
Undersample 100% Vacant	J48	80%	58.30%	48.00%	71.40%	79.10%	67.34%	66.60%
Undersample 40% Vacant	REPTree	80%	62.40%	73.70%	72.70%	61.10%	66.99%	66.90%
Undersample 40% Vacant	REPTree	65%	64.90%	69.30%	69.30%	64.90%	66.99%	67.00%
Undersample 100% Vacant	MLP	65%	58.30%	49.50%	71.10%	77.80%	66.92%	66.30%

3 Classes – Dataset Split By Zone

DATASET	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
				Precision	Recall	Precision	Recall	Precision	Recall		
SMOTE Oversample - 40% Vacant	ZONE 1	RandomForest	80%	71.40%	72.80%	73.20%	74.60%	76.50%	73.20%	73.64%	73.60%
SMOTE Oversample - 100% Vacant	ZONE 1	RandomForest	80%	62.90%	69.10%	69.90%	67.90%	80.50%	77.90%	72.37%	72.50%
SMOTE Oversample - 100% Vacant	ZONE 16	RandomForest	80%	67.70%	64.30%	52.00%	62.90%	84.20%	79.90%	71.72%	72.10%
SMOTE Oversample - 100% Vacant	ZONE 16	RandomForest	65%	68.00%	66.40%	54.10%	58.90%	81.10%	79.30%	71.21%	71.40%
SMOTE Oversample - 40% Vacant	ZONE 1	J48	80%	64.10%	68.70%	74.00%	78.50%	74.70%	63.40%	71.13%	71.10%
SMOTE Oversample - 100% Vacant	ZONE 1	RandomForest	65%	63.90%	66.10%	70.50%	69.80%	76.10%	75.20%	71.10%	75.70%
SMOTE Oversample - 40% Vacant	ZONE 1	MLP	65%	63.90%	66.10%	70.20%	76.70%	77.80%	66.50%	70.48%	70.50%
SMOTE Oversample - 40% Vacant	ZONE 1	RandomForest	65%	65.00%	66.40%	70.20%	74.00%	76.00%	69.40%	70.40%	70.40%
SMOTE Oversample - 100% Vacant	ZONE 1	MLP	80%	59.60%	70.60%	71.30%	69.00%	77.20%	71.10%	70.26%	70.40%
SMOTE Oversample - 40% Vacant	ZONE 16	RandomForest	80%	62.00%	73.40%	65.30%	53.60%	81.50%	79.50%	70.13%	70.00%
SMOTE Oversample - 40% Vacant	ZONE 1	MLP	80%	65.20%	69.20%	70.00%	74.60%	74.90%	63.90%	69.81%	69.80%
SMOTE Oversample - 100% Vacant	ZONE 16	MLP	80%	65.60%	69.90%	53.90%	50.00%	77.70%	76.70%	69.35%	69.30%
SMOTE Oversample - 40% Vacant	ZONE 1	J48	65%	60.90%	66.70%	71.30%	76.50%	75.40%	62.30%	69.22%	69.20%
SMOTE Oversample - 40% Vacant	ZONE 16	MLP	80%	62.10%	74.00%	66.20%	59.60%	79.10%	72.20%	69.19%	69.30%
SMOTE Oversample - 100% Vacant	ZONE 16	J48	80%	65.50%	66.80%	52.70%	54.80%	78.30%	76.00%	69.04%	69.20%
SMOTE Oversample - 100% Vacant	ZONE 1	J48	80%	61.90%	64.40%	66.80%	69.00%	75.50%	71.70%	69.02%	69.10%
SMOTE Oversample - 40% Vacant	ZONE 16	RandomForest	65%	59.70%	72.30%	65.70%	53.80%	80.90%	77.10%	69.01%	69.00%
Undersample 100% Vacant	ZONE 1	RandomForest	80%	40.90%	44.70%	41.60%	41.20%	84.30%	82.70%	68.40%	68.70%
SMOTE Oversample - 100% Vacant	ZONE 16	MLP	65%	65.40%	64.60%	49.40%	57.10%	79.20%	74.70%	68.05%	68.40%
SMOTE Oversample - 40% Vacant	ZONE 1	REPTree	80%	59.00%	65.60%	67.80%	77.10%	81.40%	57.60%	67.89%	67.80%
SMOTE Oversample - 100% Vacant	ZONE 16	J48	65%	64.00%	62.50%	53.20%	51.30%	75.00%	77.20%	67.51%	67.40%
SMOTE Oversample - 40% Vacant	ZONE 16	J48	80%	60.40%	74.00%	70.00%	55.60%	73.10%	70.20%	67.30%	67.20%
SMOTE Oversample - 40% Vacant	ZONE 16	MLP	65%	60.40%	72.90%	62.40%	56.90%	78.60%	70.00%	67.28%	67.40%
SMOTE Oversample - 100% Vacant	ZONE 1	J48	65%	60.30%	62.00%	66.00%	71.20%	72.70%	66.60%	67.07%	67.10%
SMOTE Oversample - 100% Vacant	ZONE 1	MLP	65%	59.50%	66.40%	66.20%	69.60%	73.20%	65.00%	66.93%	67.00%
Undersample 100% Vacant	ZONE 16	RandomForest	80%	38.50%	37.60%	28.80%	25.70%	82.00%	84.50%	66.73%	66.20%
Undersample 100% Vacant	ZONE 1	RandomForest	65%	42.90%	34.30%	41.50%	42.10%	79.30%	83.30%	65.91%	65.20%
SMOTE Oversample - 100% Vacant	ZONE 16	REPTree	80%	61.60%	58.20%	51.10%	54.00%	74.10%	75.10%	65.72%	65.70%
Undersample 100% Vacant	ZONE 16	J48	80%	31.30%	16.10%	35.10%	17.60%	71.90%	90.10%	65.10%	60.30%
SMOTE Oversample - 100% Vacant	ZONE 1	REPTree	65%	60.00%	54.80%	64.50%	68.20%	68.40%	68.60%	65.09%	65.00%
SMOTE Oversample - 40% Vacant	ZONE 1	REPTree	65%	56.20%	60.60%	66.50%	77.50%	73.40%	53.00%	65.01%	64.70%
Undersample 100% Vacant	ZONE 16	RandomForest	65%	39.70%	33.10%	29.70%	24.80%	77.40%	84.60%	64.64%	63.20%
Undersample 100% Vacant	ZONE 16	J48	65%	40.00%	20.10%	41.50%	15.60%	69.10%	90.90%	64.53%	59.10%
Undersample 100% Vacant	ZONE 16	REPTree	80%	34.50%	21.50%	31.30%	13.50%	71.50%	88.50%	64.49%	60.00%
SMOTE Oversample - 100% Vacant	ZONE 16	REPTree	65%	59.10%	58.70%	47.80%	49.60%	73.50%	72.70%	63.72%	63.00%
Undersample 100% Vacant	ZONE 1	REPTree	80%	23.90%	12.90%	45.30%	29.80%	70.90%	86.30%	63.55%	60.00%

SMOTE Oversample - 40% Vacant	ZONE 16	J48	65%	57.10%	73.30%	61.80%	48.50%	72.10%	66.10%	63.50%	63.30%
SMOTE Oversample - 100% Vacant	ZONE 1	REPTree	80%	56.80%	51.50%	63.90%	69.70%	66.60%	65.20%	63.44%	63.30%
Undersample 100% Vacant	ZONE 1	J48	80%	30.40%	16.50%	45.40%	38.60%	71.30%	82.40%	63.38%	60.90%
SMOTE Oversample - 40% Vacant	ZONE 16	REPTree	80%	58.00%	56.60%	64.80%	53.60%	65.10%	74.60%	62.76%	62.40%
Undersample 100% Vacant	ZONE 16	REPTree	65%	41.20%	24.90%	23.70%	9.90%	68.80%	87.60%	62.43%	57.60%
SMOTE Oversample - 40% Vacant	ZONE 16	REPTree	65%	55.00%	64.00%	62.40%	50.40%	68.00%	68.00%	61.77%	61.70%
Undersample 100% Vacant	ZONE 1	J48	65%	26.40%	8.40%	41.30%	40.20%	69.10%	83.00%	60.88%	57.10%
Undersample 100% Vacant	ZONE 1	MLP	65%	38.30%	35.50%	37.60%	38.80%	74.80%	75.40%	60.57%	60.50%
Undersample 100% Vacant	ZONE 1	MLP	80%	36.50%	31.80%	32.90%	40.40%	77.00%	73.70%	60.50%	61.00%
Undersample 40% Vacant	ZONE 1	RandomForest	65%	41.10%	36.00%	45.90%	47.40%	75.50%	78.10%	60.45%	60.10%
Undersample 40% Vacant	ZONE 1	RandomForest	80%	38.60%	35.80%	39.50%	45.80%	80.20%	76.70%	59.91%	60.30%
Undersample 100% Vacant	ZONE 1	REPTree	65%	22.90%	4.80%	40.60%	31.30%	65.60%	85.50%	59.86%	54.40%
Undersample 40% Vacant	ZONE 16	RandomForest	65%	43.10%	41.40%	34.50%	28.20%	73.80%	80.60%	59.47%	58.40%
Undersample 100% Vacant	ZONE 16	MLP	65%	32.40%	33.70%	25.90%	20.60%	74.30%	77.30%	59.39%	58.70%
Undersample 100% Vacant	ZONE 16	MLP	80%	32.40%	36.60%	24.70%	28.40%	78.70%	73.10%	59.39%	60.40%
Undersample 40% Vacant	ZONE 16	J48	80%	41.10%	38.50%	41.20%	33.30%	70.60%	78.20%	58.55%	57.50%
Undersample 40% Vacant	ZONE 16	RandomForest	80%	37.20%	36.50%	35.20%	29.80%	74.20%	79.60%	58.03%	57.20%
Undersample 40% Vacant	ZONE 16	MLP	65%	41.10%	42.50%	38.80%	36.60%	71.50%	71.90%	56.95%	56.90%
Undersample 40% Vacant	ZONE 1	J48	80%	31.00%	18.90%	41.00%	53.30%	71.60%	73.10%	56.18%	55.00%
Undersample 40% Vacant	ZONE 1	MLP	80%	37.30%	40.00%	37.50%	50.50%	80.30%	64.80%	55.71%	57.20%
Undersample 40% Vacant	ZONE 1	REPTree	80%	34.20%	13.70%	39.60%	55.10%	68.60%	73.10%	55.48%	53.30%
Undersample 40% Vacant	ZONE 16	REPTree	65%	37.20%	38.50%	42.30%	38.70%	68.90%	70.00%	55.33%	55.20%
Undersample 40% Vacant	ZONE 16	MLP	80%	36.00%	41.70%	36.80%	33.30%	72.90%	70.40%	55.18%	55.40%
Undersample 40% Vacant	ZONE 16	J48	65%	40.80%	35.60%	33.00%	26.80%	66.50%	75.60%	55.03%	53.70%
Undersample 40% Vacant	ZONE 16	REPTree	80%	38.50%	41.70%	40.40%	22.60%	64.70%	73.80%	54.66%	53.00%
Undersample 40% Vacant	ZONE 1	MLP	65%	38.50%	34.20%	39.50%	48.30%	71.40%	66.00%	54.19%	54.60%
Undersample 40% Vacant	ZONE 1	REPTree	65%	33.10%	26.10%	43.90%	51.20%	63.20%	63.10%	51.80%	51.40%
Undersample 40% Vacant	ZONE 1	J48	65%	24.60%	18.00%	42.90%	44.50%	61.60%	67.30%	50.33%	49.20%

2 Classes – Dataset Split By Zone

DATASET	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		VACANT		Correct Instances	Weighted Average F1-Score
				Precision	Recall	Precision	Recall		
Undersample 100% Vacant	ZONE 1	RandomForest	80%	67.30%	74.40%	84.90%	79.90%	77.92%	78.10%
Undersample 100% Vacant	ZONE 16	RandomForest	80%	64.00%	67.10%	82.50%	80.50%	75.92%	76.00%
Undersample 100% Vacant	ZONE 1	RandomForest	65%	67.30%	69.20%	79.90%	78.50%	74.85%	74.90%
Undersample 40% Vacant	ZONE 16	RandomForest	65%	70.90%	77.20%	78.30%	72.20%	74.56%	74.60%
Undersample 40% Vacant	ZONE 1	RandomForest	80%	68.60%	82.20%	80.70%	66.50%	73.89%	73.80%
Undersample 40% Vacant	ZONE 16	MLP	80%	70.50%	75.60%	77.20%	72.30%	73.83%	73.90%
Undersample 40% Vacant	ZONE 16	RandomForest	80%	70.20%	74.40%	76.40%	72.30%	73.32%	73.30%
Undersample 40% Vacant	ZONE 1	RandomForest	65%	70.30%	79.00%	76.60%	67.30%	73.10%	73.00%
Undersample 100% Vacant	ZONE 16	RandomForest	65%	62.70%	59.70%	77.80%	79.90%	72.58%	72.40%
Undersample 40% Vacant	ZONE 1	J48	80%	67.70%	77.70%	77.20%	67.00%	72.03%	72.00%
Undersample 40% Vacant	ZONE 1	MLP	80%	65.10%	86.60%	83.10%	58.60%	71.79%	71.40%
Undersample 100% Vacant	ZONE 16	MLP	80%	56.90%	66.50%	81.00%	74.00%	71.43%	71.90%
Undersample 40% Vacant	ZONE 16	J48	80%	69.10%	69.40%	73.20%	72.80%	71.24%	71.20%
Undersample 40% Vacant	ZONE 1	REPTree	80%	66.20%	76.70%	75.90%	65.20%	70.63%	70.60%
Undersample 100% Vacant	ZONE 16	J48	65%	60.80%	51.60%	74.70%	81.20%	70.48%	69.90%
Undersample 40% Vacant	ZONE 16	MLP	65%	66.70%	72.20%	73.70%	68.30%	70.12%	70.10%
Undersample 100% Vacant	ZONE 1	J48	80%	57.40%	60.30%	77.30%	75.10%	69.84%	70.00%
Undersample 100% Vacant	ZONE 1	MLP	80%	56.40%	68.80%	80.30%	70.40%	69.84%	70.40%
Undersample 40% Vacant	ZONE 16	REPTree	80%	66.10%	70.60%	72.70%	68.40%	69.43%	69.50%
Undersample 40% Vacant	ZONE 16	J48	65%	65.70%	72.20%	73.30%	66.90%	69.38%	69.40%
Undersample 100% Vacant	ZONE 16	REPTree	65%	58.50%	51.90%	74.40%	79.20%	69.31%	68.90%
Undersample 100% Vacant	ZONE 16	J48	80%	55.30%	50.30%	75.40%	78.90%	69.18%	68.80%
Undersample 100% Vacant	ZONE 16	REPTree	80%	54.90%	53.90%	76.40%	77.10%	69.18%	69.10%
Undersample 100% Vacant	ZONE 16	MLP	65%	56.40%	54.20%	74.60%	76.20%	68.26%	68.10%
Undersample 40% Vacant	ZONE 1	MLP	65%	66.90%	70.20%	69.30%	66.00%	68.04%	68.00%
Undersample 100% Vacant	ZONE 1	REPTree	80%	55.40%	48.70%	73.30%	78.20%	67.68%	67.20%
Undersample 100% Vacant	ZONE 1	MLP	65%	58.40%	59.20%	73.70%	73.10%	67.66%	67.70%
Undersample 40% Vacant	ZONE 16	REPTree	65%	65.30%	64.90%	69.30%	69.70%	67.46%	67.40%
Undersample 40% Vacant	ZONE 1	REPTree	65%	65.10%	73.70%	70.30%	61.20%	67.38%	67.00%
Undersample 40% Vacant	ZONE 1	J48	65%	66.30%	67.70%	67.70%	66.20%	66.98%	67.00%
Undersample 100% Vacant	ZONE 1	J48	65%	60.40%	44.50%	69.60%	81.30%	66.94%	65.70%
Undersample 100% Vacant	ZONE 1	REPTree	65%	57.60%	41.10%	68.10%	80.60%	65.20%	63.70%

3 Classes – Dataset Split By Street Occupancy

DATASET	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
				Precision	Recall	Precision	Recall	Precision	Recall		
SMOTE Oversample - 100% Vacant	MOST Occupied	RandomForest	80%	68.30%	65.90%	69.30%	71.80%	80.50%	79.50%	73.51%	73.50%
SMOTE Oversample - 100% Vacant	LESS Occupied	RandomForest	80%	68.00%	68.90%	56.90%	55.50%	82.00%	82.00%	72.99%	73.00%
SMOTE Oversample - 40% Vacant	LESS Occupied	RandomForest	80%	66.80%	70.90%	62.80%	58.70%	80.30%	79.30%	71.45%	71.40%
SMOTE Oversample - 100% Vacant	MOST Occupied	J48	80%	65.20%	61.80%	66.40%	76.80%	79.00%	69.90%	70.60%	70.60%
SMOTE Oversample - 40% Vacant	MOST Occupied	J48	80%	63.70%	65.10%	71.20%	79.80%	76.50%	61.90%	70.36%	70.20%
SMOTE Oversample - 100% Vacant	LESS Occupied	RandomForest	65%	67.80%	62.90%	52.80%	52.50%	77.60%	81.40%	70.12%	69.90%
SMOTE Oversample - 40% Vacant	MOST Occupied	RandomForest	80%	65.30%	62.30%	67.70%	79.40%	80.10%	64.00%	70.05%	69.90%
SMOTE Oversample - 40% Vacant	MOST Occupied	MLP	80%	67.70%	60.00%	67.00%	82.00%	78.50%	61.90%	69.89%	69.60%
SMOTE Oversample - 40% Vacant	LESS Occupied	RandomForest	65%	66.80%	70.80%	60.00%	54.50%	77.10%	76.90%	69.63%	69.50%
SMOTE Oversample - 100% Vacant	MOST Occupied	MLP	80%	59.70%	69.40%	67.20%	68.90%	78.70%	69.50%	69.27%	69.50%
SMOTE Oversample - 100% Vacant	MOST Occupied	RandomForest	65%	58.60%	64.60%	67.30%	66.60%	77.70%	74.10%	69.14%	69.30%
SMOTE Oversample - 100% Vacant	LESS Occupied	MLP	80%	61.60%	71.10%	56.30%	56.30%	80.20%	71.90%	68.91%	69.20%
SMOTE Oversample - 40% Vacant	MOST Occupied	RandomForest	65%	67.30%	59.90%	66.00%	78.50%	76.60%	63.10%	68.84%	68.60%
SMOTE Oversample - 40% Vacant	LESS Occupied	MLP	80%	62.40%	74.40%	61.40%	58.70%	80.40%	69.70%	68.69%	68.80%
SMOTE Oversample - 40% Vacant	MOST Occupied	MLP	65%	65.70%	57.10%	65.60%	79.80%	77.60%	62.50%	68.39%	68.10%
Undersample 100% Vacant	LESS Occupied	RandomForest	80%	51.60%	39.80%	21.80%	19.70%	78.90%	86.50%	68.11%	66.80%
SMOTE Oversample - 100% Vacant	LESS Occupied	MLP	65%	65.90%	64.70%	53.40%	50.70%	74.00%	76.30%	67.86%	67.70%
SMOTE Oversample - 40% Vacant	LESS Occupied	MLP	65%	63.50%	71.70%	60.90%	53.20%	75.70%	72.60%	67.85%	67.80%
SMOTE Oversample - 100% Vacant	LESS Occupied	J48	80%	61.10%	67.50%	54.10%	50.40%	76.70%	73.10%	67.30%	67.40%
SMOTE Oversample - 100% Vacant	MOST Occupied	REPTree	80%	62.00%	56.60%	63.20%	72.90%	74.80%	67.90%	67.15%	67.10%
SMOTE Oversample - 100% Vacant	MOST Occupied	J48	65%	64.60%	55.80%	63.30%	72.80%	72.80%	68.00%	67.10%	67.00%
SMOTE Oversample - 100% Vacant	MOST Occupied	MLP	65%	56.50%	66.30%	67.40%	63.20%	73.40%	70.60%	66.87%	67.00%
SMOTE Oversample - 40% Vacant	MOST Occupied	J48	65%	66.30%	57.40%	64.40%	79.60%	64.40%	79.60%	66.67%	63.50%
Undersample 100% Vacant	LESS Occupied	RandomForest	65%	44.70%	34.50%	31.40%	23.30%	76.50%	86.80%	66.56%	64.50%
SMOTE Oversample - 40% Vacant	MOST Occupied	REPTree	80%	62.30%	65.10%	67.60%	74.90%	67.10%	54.00%	65.93%	65.60%
SMOTE Oversample - 100% Vacant	MOST Occupied	REPTree	65%	56.20%	60.50%	65.00%	67.60%	73.00%	67.00%	65.81%	65.90%
Undersample 100% Vacant	LESS Occupied	J48	80%	0.00%	0.00%	0.00%	0.00%	65.50%	100.00 %	65.48%	51.80%
SMOTE Oversample - 100% Vacant	LESS Occupied	J48	65%	62.50%	60.90%	53.80%	41.90%	69.00%	75.80%	64.77%	64.20%
SMOTE Oversample - 40% Vacant	LESS Occupied	J48	65%	62.00%	71.10%	58.70%	48.10%	69.90%	68.10%	64.59%	64.30%
SMOTE Oversample - 40% Vacant	LESS Occupied	J48	80%	59.00%	69.30%	63.30%	55.10%	71.00%	66.00%	64.53%	64.50%
SMOTE Oversample - 40% Vacant	MOST Occupied	REPTree	65%	61.00%	53.20%	62.10%	78.30%	73.00%	53.60%	64.13%	63.60%
Undersample 100% Vacant	LESS Occupied	J48	65%	0.00%	0.00%	0.00%	0.00%	64.00%	100.00 %	63.99%	49.90%
SMOTE Oversample - 100% Vacant	LESS Occupied	REPTree	80%	58.80%	67.50%	44.40%	40.30%	73.30%	68.30%	63.21%	63.20%
Undersample 100% Vacant	MOST Occupied	J48	80%	48.10%	14.90%	44.50%	43.80%	70.10%	84.30%	63.04%	59.80%
Undersample 100% Vacant	MOST Occupied	RandomForest	80%	38.10%	27.60%	39.70%	46.30%	78.70%	79.70%	63.04%	62.60%
Undersample 100% Vacant	MOST Occupied	J48	65%	50.00%	15.80%	41.90%	42.90%	70.40%	83.60%	62.29%	59.40%

Undersample 100% Vacant	MOST Occupied	RandomForest	65%	38.50%	29.60%	41.60%	47.90%	77.30%	77.50%	62.29%	62.00%
Undersample 100% Vacant	LESS Occupied	MLP	65%	38.70%	37.30%	27.10%	22.40%	75.20%	78.70%	61.95%	61.20%
Undersample 40% Vacant	LESS Occupied	RandomForest	65%	50.30%	42.40%	27.90%	24.20%	74.40%	83.80%	61.80%	60.50%
SMOTE Oversample - 40% Vacant	LESS Occupied	REPTree	80%	56.60%	71.40%	58.60%	49.30%	69.70%	61.00%	61.76%	61.60%
SMOTE Oversample - 40% Vacant	LESS Occupied	REPTree	65%	57.20%	74.20%	56.00%	40.30%	69.60%	62.60%	61.62%	61.60%
Undersample 100% Vacant	LESS Occupied	REPTree	65%	28.60%	10.90%	7.70%	0.90%	64.70%	90.60%	60.66%	52.20%
Undersample 100% Vacant	MOST Occupied	REPTree	65%	35.90%	9.20%	40.70%	45.20%	69.60%	81.30%	60.40%	57.00%
Undersample 40% Vacant	LESS Occupied	RandomForest	80%	51.80%	43.10%	19.70%	18.20%	74.10%	83.40%	60.33%	59.10%
Undersample 100% Vacant	LESS Occupied	MLP	80%	38.80%	38.20%	20.70%	27.90%	77.90%	73.60%	60.23%	61.20%
SMOTE Oversample - 100% Vacant	LESS Occupied	REPTree	65%	56.00%	54.30%	48.50%	44.20%	66.30%	69.80%	60.10%	59.90%
Undersample 100% Vacant	LESS Occupied	REPTree	80%	29.30%	17.90%	6.30%	1.60%	67.20%	85.10%	60.04%	54.60%
Undersample 100% Vacant	MOST Occupied	REPTree	80%	32.40%	13.80%	36.10%	39.70%	71.50%	80.40%	59.53%	57.20%
Undersample 40% Vacant	MOST Occupied	RandomForest	65%	36.50%	40.30%	52.40%	49.60%	74.40%	74.40%	59.47%	59.60%
Undersample 40% Vacant	MOST Occupied	RandomForest	80%	35.70%	41.10%	55.10%	53.20%	74.60%	72.00%	59.38%	59.80%
Undersample 40% Vacant	MOST Occupied	J48	80%	25.00%	15.10%	53.70%	61.70%	69.90%	73.10%	58.10%	56.50%
Undersample 40% Vacant	MOST Occupied	J48	65%	32.90%	21.00%	53.40%	56.30%	67.00%	73.80%	58.00%	56.70%
Undersample 40% Vacant	LESS Occupied	MLP	65%	44.90%	39.70%	26.80%	28.30%	72.90%	76.40%	57.64%	57.30%
Undersample 100% Vacant	MOST Occupied	MLP	65%	32.40%	23.70%	36.70%	40.10%	71.70%	74.50%	57.62%	57.00%
Undersample 40% Vacant	LESS Occupied	MLP	80%	50.40%	41.60%	21.30%	25.80%	73.00%	76.20%	57.28%	57.30%
Undersample 100% Vacant	MOST Occupied	MLP	80%	27.10%	18.40%	36.50%	47.90%	74.30%	71.90%	57.20%	57.00%
Undersample 40% Vacant	LESS Occupied	J48	80%	46.00%	33.60%	25.00%	15.20%	64.00%	82.10%	56.10%	53.00%
Undersample 40% Vacant	LESS Occupied	J48	65%	43.90%	33.90%	30.00%	20.00%	64.50%	79.10%	56.03%	53.70%
Undersample 40% Vacant	MOST Occupied	MLP	65%	31.60%	29.80%	47.70%	52.10%	70.50%	67.20%	55.07%	55.20%
Undersample 40% Vacant	MOST Occupied	MLP	80%	27.50%	26.00%	48.70%	52.50%	70.80%	68.00%	54.50%	54.50%
Undersample 40% Vacant	MOST Occupied	REPTree	65%	31.00%	29.00%	50.40%	50.40%	65.80%	67.50%	54.48%	54.30%
Undersample 40% Vacant	MOST Occupied	REPTree	80%	21.40%	16.40%	50.70%	51.80%	66.70%	72.00%	54.24%	53.20%
Undersample 40% Vacant	LESS Occupied	REPTree	80%	43.40%	35.80%	28.60%	12.10%	59.60%	76.20%	53.29%	50.30%
Undersample 40% Vacant	LESS Occupied	REPTree	65%	35.30%	24.10%	9.40%	2.50%	57.80%	80.60%	51.07%	45.50%

2 Classes – Dataset Split By Street Occupancy

DATASET	DATASET CONTENTS	ALGORITHM	TRAIN	FULL		VACANT		Correct Instances	Weighted Average F1-Score
				Precision	Recall	Precision	Recall		
Undersample 40% Vacant	LESS Occupied	RandomForest	80%	74.80%	74.40%	76.80%	77.10%	75.82%	75.80%
Undersample 100% Vacant	LESS Occupied	RandomForest	80%	65.50%	63.00%	80.90%	82.50%	75.80%	75.70%
Undersample 100% Vacant	MOST Occupied	RandomForest	80%	69.30%	70.70%	79.80%	78.80%	75.49%	75.50%
Undersample 40% Vacant	LESS Occupied	RandomForest	65%	72.00%	71.80%	75.90%	76.10%	74.13%	74.10%
Undersample 40% Vacant	MOST Occupied	J48	80%	76.00%	77.10%	71.50%	70.30%	74.04%	74.00%
Undersample 40% Vacant	MOST Occupied	J48	65%	74.40%	78.30%	73.50%	69.10%	74.01%	73.90%
Undersample 40% Vacant	MOST Occupied	RandomForest	80%	73.90%	80.80%	73.50%	65.10%	73.78%	73.60%
Undersample 100% Vacant	LESS Occupied	RandomForest	65%	64.30%	56.80%	77.20%	82.20%	73.10%	72.70%
Undersample 40% Vacant	MOST Occupied	RandomForest	65%	71.90%	80.80%	74.30%	63.70%	72.83%	72.60%
Undersample 40% Vacant	MOST Occupied	MLP	80%	73.10%	79.90%	72.30%	64.00%	72.75%	72.50%
Undersample 100% Vacant	MOST Occupied	RandomForest	65%	65.10%	69.60%	77.80%	74.00%	72.19%	72.30%
Undersample 40% Vacant	LESS Occupied	MLP	80%	73.70%	63.50%	70.50%	79.40%	71.83%	71.60%
Undersample 40% Vacant	LESS Occupied	MLP	65%	68.10%	70.60%	74.00%	71.60%	71.18%	71.20%
Undersample 100% Vacant	MOST Occupied	REPTree	80%	63.30%	65.40%	75.90%	74.20%	70.62%	70.70%
Undersample 40% Vacant	MOST Occupied	REPTree	80%	71.10%	77.10%	68.80%	61.70%	70.18%	70.00%
Undersample 40% Vacant	MOST Occupied	MLP	65%	68.80%	80.50%	72.20%	58.00%	70.04%	69.60%
Undersample 100% Vacant	MOST Occupied	J48	80%	62.60%	61.10%	74.00%	75.20%	69.46%	69.40%
Undersample 100% Vacant	MOST Occupied	J48	65%	62.90%	60.70%	73.30%	75.10%	69.19%	69.10%
Undersample 100% Vacant	MOST Occupied	MLP	80%	59.40%	73.10%	78.30%	66.00%	68.87%	69.20%
Undersample 100% Vacant	LESS Occupied	MLP	65%	56.70%	55.70%	75.30%	76.00%	68.70%	68.60%
Undersample 100% Vacant	LESS Occupied	MLP	80%	53.20%	66.80%	79.80%	69.10%	68.29%	68.90%
Undersample 40% Vacant	LESS Occupied	J48	80%	65.60%	68.50%	70.10%	67.30%	67.84%	67.90%
Undersample 100% Vacant	MOST Occupied	REPTree	65%	61.50%	55.80%	71.10%	75.70%	67.52%	67.20%
Undersample 40% Vacant	MOST Occupied	REPTree	65%	67.60%	71.70%	65.10%	60.60%	66.52%	66.40%
Undersample 100% Vacant	LESS Occupied	J48	80%	51.40%	49.50%	73.90%	75.40%	66.42%	66.30%
Undersample 40% Vacant	LESS Occupied	J48	65%	62.90%	63.10%	68.30%	68.20%	65.82%	65.80%
Undersample 100% Vacant	LESS Occupied	REPTree	65%	53.80%	35.40%	69.50%	82.90%	65.81%	63.80%
Undersample 100% Vacant	MOST Occupied	MLP	65%	57.60%	56.40%	70.10%	71.10%	65.07%	65.00%
Undersample 100% Vacant	LESS Occupied	J48	65%	51.40%	37.50%	69.50%	80.10%	64.74%	63.20%
Undersample 100% Vacant	LESS Occupied	REPTree	80%	47.10%	35.90%	70.00%	78.80%	63.98%	62.60%
Undersample 40% Vacant	LESS Occupied	REPTree	80%	63.30%	52.70%	62.60%	72.20%	62.91%	62.50%
Undersample 40% Vacant	LESS Occupied	REPTree	65%	58.40%	52.60%	62.60%	67.90%	60.86%	60.60%

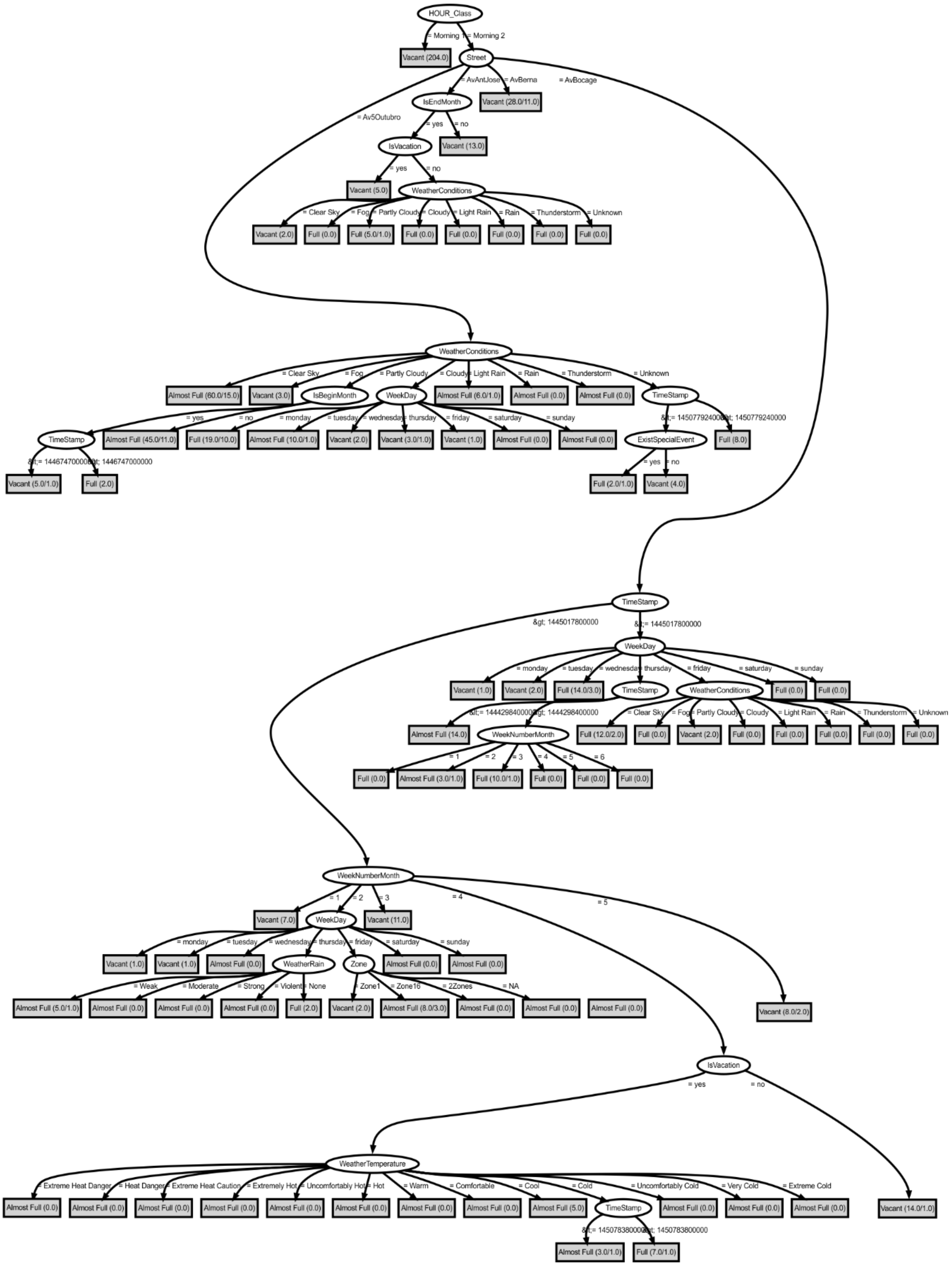
3 Classes – Dataset with All Streets Testing with data from 01-11 to 15-11

DATASET	ALGORITHM	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
		Precision	Recall	Precision	Recall	Precision	Recall		
Undersample 100% Vacant	J48	29.40%	10.00%	46.80%	33.60%	67.60%	87.20%	67.60%	57.50%
Undersample 100% Vacant	RandomForest	33.30%	5.30%	46.70%	9.20%	63.30%	95.30%	63.30%	51.80%
Undersample 100% Vacant	REPTree	30.40%	14.00%	48.60%	23.00%	66.60%	88.40%	66.60%	56.60%
Undersample 100% Vacant	MLP	34.40%	7.30%	28.80%	30.30%	64.60%	79.00%	64.60%	51.90%
Undersample 40% Vacant	RandomForest	21.10%	2.70%	53.10%	22.40%	55.60%	93.50%	55.60%	44.70%
Undersample 40% Vacant	REPTree	32.90%	18.00%	45.20%	46.70%	62.00%	74.10%	62.00%	51.60%
Undersample 40% Vacant	J48	36.80%	14.00%	48.90%	30.30%	56.80%	83.50%	56.80%	48.80%
SMOTE Oversample - 100% Vacant	REPTree	38.10%	35.30%	53.50%	28.00%	53.30%	71.50%	53.30%	47.60%
SMOTE Oversample - 100% Vacant	J48	37.80%	17.00%	51.90%	36.50%	49.30%	74.90%	49.30%	45.00%
Undersample 40% Vacant	MLP	33.30%	16.70%	34.70%	27.60%	54.80%	72.90%	54.80%	45.10%
SMOTE Oversample - 100% Vacant	MLP	38.30%	19.70%	40.00%	24.30%	51.50%	79.20%	51.50%	43.50%
SMOTE Oversample - 40% Vacant	REPTree	45.80%	36.00%	50.60%	40.50%	46.40%	64.50%	46.40%	46.60%
SMOTE Oversample - 100% Vacant	RandomForest	23.30%	4.70%	61.80%	15.50%	47.20%	92.30%	47.20%	37.00%
SMOTE Oversample - 40% Vacant	RandomForest	43.10%	15.70%	71.60%	25.70%	41.20%	90.70%	41.20%	39.50%
SMOTE Oversample - 40% Vacant	MLP	40.60%	26.70%	63.40%	27.30%	41.70%	77.60%	41.70%	41.80%
SMOTE Oversample - 40% Vacant	J48	45.70%	23.00%	59.20%	29.60%	39.70%	76.90%	39.70%	41.10%

2 Classes – Dataset with All Streets Testing with data from 01-11 to 15-11

DATASET	ALGORITHM	FULL		VACANT		Correct Instances	Weighted Average F1-Score
		Precision	Recall	Precision	Recall		
Undersample 100% Vacant	REPTree	53.50%	43.40%	68.80%	76.80%	64.06%	47.90%
Undersample 100% Vacant	J48	53.20%	38.70%	67.70%	79.00%	63.68%	44.80%
Undersample 100% Vacant	RandomForest	51.00%	16.60%	63.70%	90.20%	62.17%	25.00%
Undersample 40% Vacant	RandomForest	66.50%	38.10%	58.40%	81.90%	60.67%	48.40%
Undersample 100% Vacant	MLP	46.90%	27.20%	64.40%	81.10%	60.53%	34.40%
Undersample 40% Vacant	J48	60.90%	51.00%	60.00%	69.20%	60.35%	55.50%
Undersample 40% Vacant	REPTree	60.20%	53.60%	60.50%	66.70%	60.35%	56.70%
Undersample 40% Vacant	MLP	58.90%	40.40%	56.70%	73.50%	57.46%	47.90%

Appendix Q – Sample of Decision Tree Generate by J48



Appendix R – Algorithm Parameters Performance Results

RandomForest Parameters (3 Classes)	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall	Precision	Recall		
Iteration = 1000	66.20%	69.90%	68.30%	69.30%	77.90%	72.80%	70.73%	70.80%
Iteration = 100 **(DEFAULT)**	66.20%	70.60%	68.20%	69.10%	77.80%	72.30%	70.64%	70.70%
Iteration = 750	66.20%	69.90%	68.00%	69.20%	77.80%	72.60%	70.59%	70.70%
Iteration = 500	65.90%	69.80%	68.20%	69.20%	77.70%	72.60%	70.54%	75.10%
Iteration = 300	66.10%	69.90%	68.00%	68.90%	77.50%	72.40%	70.45%	70.50%
Iteration = 400	65.80%	69.90%	68.10%	68.90%	77.40%	72.10%	70.35%	70.40%
Iteration = 200	65.70%	70.20%	67.50%	68.80%	77.50%	71.30%	70.12%	70.20%

MLP Parameters (3 Classes)	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall	Precision	Recall		
'i'attribs+LR 0.1	68.20%	66.30%	67.20%	76.50%	77.40%	68.90%	70.69%	70.70%
'i'attribs+LR 0.2	66.40%	70.60%	68.50%	73.50%	77.90%	68.10%	70.69%	70.70%
't' (att+cls)+LR 0.5	63.20%	76.20%	70.40%	67.00%	78.00%	67.40%	69.98%	70.00%
'i'attribs+LR 0.5	63.60%	73.30%	69.40%	67.10%	77.70%	69.70%	69.93%	70.00%
't' (att+cls)+LR 0.4	65.20%	69.00%	67.60%	71.30%	76.50%	68.50%	69.60%	69.70%
'a' ((att+cls)/2)+LR 0.4	62.70%	73.00%	70.00%	68.60%	77.00%	67.30%	69.50%	69.60%
'a'((att+cls)/2)+LR 0.1	65.60%	69.50%	66.00%	73.00%	76.70%	64.50%	68.98%	69.00%
'i'attribs+LR 0.3	63.20%	67.90%	66.50%	72.80%	78.40%	65.90%	68.89%	69.00%
'i'attribs+LR 0.4	63.20%	67.90%	66.50%	72.80%	78.40%	65.90%	68.89%	69.00%
't' (att+cls)+LR 0.3	64.30%	70.10%	66.30%	71.80%	77.10%	64.70%	68.79%	68.90%
't' (att+cls)+LR 0.1	67.80%	64.60%	65.30%	75.00%	74.10%	66.30%	68.75%	68.70%
't' (att+cls)+LR 0.2	65.40%	69.90%	64.70%	73.60%	77.00%	61.80%	68.37%	68.40%
'a'((att+cls)/2)+LR 0.2	64.90%	67.00%	63.90%	73.20%	77.30%	64.00%	68.09%	68.20%
'a'((att+cls)/2)+LR 0.3 **(Default)**	61.60%	68.40%	66.20%	71.30%	77.80%	64.30%	67.94%	68.10%
'a'((att+cls)/2)+LR 0.5	63.70%	66.40%	67.30%	70.00%	72.50%	66.80%	67.80%	67.80%
'o'classes+LR 0.2	54.50%	53.20%	59.90%	66.70%	65.00%	59.00%	59.86%	59.80%
'o'classes+LR 0.1	55.20%	49.80%	62.70%	58.00%	59.00%	68.30%	59.10%	58.90%
'o'classes+LR 0.3	54.10%	47.20%	58.70%	67.50%	63.20%	60.90%	58.96%	58.70%
'o'classes+LR 0.5	54.70%	53.20%	51.60%	79.80%	82.60%	39.90%	57.73%	56.90%
'o'classes+LR 0.4	52.80%	41.40%	54.30%	70.70%	63.50%	56.70%	56.78%	56.30%

J48 Parameters (3 Classes)	FULL		ALMOST FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall	Precision	Recall		
Conf 0.35	62.40%	69.80%	66.80%	71.00%	74.20%	61.80%	67.42%	67.40%
Conf 0.50	62.10%	70.10%	66.70%	70.00%	74.50%	62.40%	67.38%	67.40%
Conf 0.30	61.90%	69.50%	67.10%	70.70%	73.70%	61.80%	67.23%	67.30%
Conf 0.40	61.90%	69.30%	66.80%	70.80%	74.10%	61.80%	67.23%	67.30%
Conf 0.45	61.90%	69.60%	66.90%	70.00%	73.70%	62.10%	67.14%	67.20%
Conf 0.2	61.50%	67.50%	66.80%	71.50%	73.40%	61.80%	66.90%	66.90%
Conf 0.25 **(DEFAULT)**	61.40%	69.00%	66.80%	69.90%	73.40%	62.00%	66.86%	66.90%
Conf 0.15	60.70%	69.20%	67.10%	69.30%	72.10%	60.90%	66.34%	66.40%
Conf 0.1	60.30%	68.40%	67.70%	68.60%	71.60%	62.10%	66.29%	66.30%

RandomForest Parameters (2 Classes)	FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall		
Iteration = 1000	67.10%	67.40%	80.10%	79.90%	75.17%	75.20%
Iteration = 100 **(DEFAULT)**	66.90%	67.40%	80.10%	79.70%	75.07%	75.10%
Iteration = 400	66.90%	67.40%	80.10%	79.70%	75.07%	75.10%
Iteration = 500	66.90%	67.40%	80.10%	79.70%	75.07%	75.10%
Iteration = 200	66.80%	67.40%	80.10%	79.60%	74.98%	75.00%
Iteration = 300	66.80%	67.40%	80.10%	79.60%	74.98%	75.00%
Iteration = 750	66.70%	67.20%	79.90%	79.60%	74.88%	74.90%

MLP Parameters (2 Classes)	FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall		
't' (att+cls)+LR 0.4	65.00%	54.00%	74.70%	82.30%	71.63%	71.00%
'i'attribs+LR 0.5	63.20%	59.10%	76.10%	79.10%	71.54%	71.30%
't' (att+cls)+LR 0.3	62.00%	60.10%	76.20%	77.60%	70.96%	70.90%
't' (att+cls)+LR 0.2	60.40%	64.40%	77.40%	74.30%	70.58%	70.80%
't' (att+cls)+LR 0.1	60.80%	61.90%	76.60%	75.70%	70.49%	70.50%
't' (att+cls)+LR 0.5	61.70%	57.80%	75.30%	78.20%	70.49%	70.30%
'a' ((att+cls)/2)+LR 0.1	60.00%	63.10%	76.80%	74.30%	70.11%	70.20%
'a' ((att+cls)/2)+LR 0.4	60.70%	57.30%	74.90%	77.40%	69.82%	69.60%
'i'attribs+LR 0.1	59.80%	60.10%	75.70%	75.40%	69.63%	69.60%
'i'attribs+LR 0.2	58.90%	60.90%	75.70%	74.20%	69.15%	69.20%
'a' ((att+cls)/2)+LR 0.2	58.90%	58.30%	74.80%	75.30%	68.86%	68.80%
'i'attribs+LR 0.3	59.80%	54.00%	73.60%	77.90%	68.86%	68.50%
'i'attribs+LR 0.4	59.40%	55.80%	74.10%	76.80%	68.86%	68.70%
'a' ((att+cls)/2)+LR 0.5	59.70%	53.00%	73.20%	78.20%	68.67%	68.30%
'a' ((att+cls)/2)+LR 0.3 **(DEFAULT)**	57.90%	57.30%	74.20%	74.70%	68.10%	68.10%
'o'classes+LR 0.1	54.10%	53.50%	71.90%	72.40%	65.23%	65.20%
'o'classes+LR 0.2	53.60%	54.30%	72.00%	71.40%	64.95%	65.00%
'o'classes+LR 0.5	53.00%	62.90%	74.50%	66.10%	64.85%	65.30%
'o'classes+LR 0.4	52.30%	66.20%	75.50%	63.30%	64.37%	64.90%
'o'classes+LR 0.3	52.30%	57.10%	72.40%	68.40%	64.09%	64.40%

J48 Parameters (2 Classes)	FULL		VACANT		Correct Instances	Weighted Average F1-Score
	Precision	Recall	Precision	Recall		
Conf 0.15	63.30%	42.20%	70.80%	85.10%	68.86%	67.20%
Conf 0.1	62.90%	40.70%	70.30%	85.40%	68.48%	66.60%
Conf 0.45	57.80%	59.10%	74.80%	73.70%	68.19%	68.30%
Conf 0.40	57.50%	59.30%	74.80%	73.30%	68.00%	68.10%
Conf 0.30	57.70%	56.10%	73.70%	75.00%	67.81%	67.70%
Conf 0.50	57.10%	60.10%	74.90%	72.50%	67.81%	68.00%
Conf 0.35	57.50%	56.10%	73.70%	74.80%	67.72%	67.60%
Conf 0.25 **(DEFAULT)**	57.50%	47.20%	71.10%	78.80%	66.86%	66.10%
Conf 0.2	57.10%	47.50%	71.00%	78.30%	66.67%	65.90%

Appendix S – Results With Parameter and Attributes Adjustment

3 Classes – Dataset With all Streets (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	62.80%	66.10%	81.50%	72.02%	72.10%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	62.60%	66.70%	81.20%	72.02%	72.10%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	62.30%	66.20%	81.30%	71.83%	72.00%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	61.50%	65.50%	82.40%	71.81%	72.00%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	61.10%	65.60%	82.40%	71.67%	71.90%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	65%	62.40%	65.80%	81.20%	71.63%	71.80%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	80%	61.30%	65.40%	82.40%	71.60%	71.80%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	61.10%	65.40%	82.40%	71.60%	71.80%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	66.70%	68.60%	78.30%	71.16%	71.20%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	66.90%	68.30%	78.10%	71.02%	71.10%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	66.70%	68.30%	77.80%	70.87%	70.90%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	66.30%	68.70%	77.70%	70.80%	70.90%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	80%	65.90%	68.50%	78.60%	70.80%	70.90%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	65%	66.20%	68.30%	77.90%	70.73%	70.80%
SMOTE Oversample - 40% Vacant	NONE	MLP	'i'attribs+LR 0.1	65%	68.20%	67.20%	77.40%	70.69%	70.70%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	65.50%	68.30%	78.20%	70.47%	70.60%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	65.50%	68.40%	78.00%	70.47%	70.60%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	65%	64.60%	68.80%	76.70%	69.88%	70.00%
SMOTE Oversample - 40% Vacant	Outlier	MLP	'i'attribs+LR 0.1	80%	66.40%	65.20%	78.00%	69.56%	69.60%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	65%	65.40%	65.50%	78.50%	69.17%	69.20%

2 Classes – Dataset With All Streets (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	67.90%	80.30%	75.64%	75.60%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	67.50%	79.90%	75.26%	75.20%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	67.40%	80.00%	75.26%	75.20%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	80%	67.10%	80.10%	75.17%	75.20%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	80%	69.50%	80.30%	74.48%	74.50%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	67.50%	78.60%	74.45%	74.40%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	67.40%	78.60%	74.40%	74.30%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	69.30%	80.40%	74.36%	74.30%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	69.60%	79.90%	74.36%	74.40%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	67.30%	78.50%	74.35%	74.30%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	65%	66.90%	78.80%	74.29%	74.20%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	69.10%	80.30%	74.23%	74.20%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	70.60%	75.80%	73.09%	73.10%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	65%	70.20%	76.10%	72.95%	72.90%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	70.20%	76.10%	72.95%	72.90%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	69.70%	75.80%	72.60%	72.60%
Undersample 40% Vacant	IsBeginMonth /EndMonth	MLP	't' (att+cls)+LR 0.4	80%	70.60%	73.00%	71.90%	71.90%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	80%	65.00%	74.70%	71.63%	71.00%
Undersample 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	65%	68.90%	74.30%	71.48%	71.50%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	80%	66.60%	75.80%	70.92%	70.90%

3 Classes – ZONE 1 (Best 20)									
Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	70.40%	74.60%	76.10%	73.79%	73.80%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	80%	70.00%	74.60%	76.00%	73.64%	73.70%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	70.40%	74.00%	76.40%	73.64%	73.70%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	70.40%	74.00%	75.90%	73.49%	73.50%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	63.20%	70.50%	80.20%	72.61%	72.70%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	62.60%	69.70%	80.90%	72.37%	72.50%
SMOTE Oversample - 40% Vacant	Outlier	MLP	'i'attribs+LR 0.1	80%	68.60%	71.50%	77.30%	72.16%	72.10%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	62.00%	69.70%	80.40%	72.00%	72.10%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	64.90%	71.30%	76.70%	71.88%	71.90%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	80%	62.60%	68.50%	80.80%	71.87%	72.00%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	64.70%	71.40%	76.60%	71.81%	71.80%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	64.70%	70.80%	76.70%	71.67%	71.70%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	65%	64.70%	70.50%	76.30%	71.39%	71.40%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	63.90%	71.60%	77.80%	71.15%	71.20%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	80%	64.00%	74.10%	74.70%	71.13%	71.10%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	80%	65.10%	73.20%	75.10%	71.13%	71.10%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	65%	66.10%	69.40%	79.60%	71.07%	71.00%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	63.80%	71.30%	77.60%	70.90%	71.00%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	80%	62.90%	75.30%	73.00%	70.84%	70.80%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth EndMonth	J48	Conf 0.35	80%	63.60%	74.10%	73.30%	70.69%	70.70%

2 Classes – ZONE 1 (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	80%	68.00%	85.20%	78.46%	78.00%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	67.70%	85.20%	78.28%	78.50%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	67.70%	85.20%	78.28%	78.50%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	67.10%	85.10%	77.92%	70.80%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	68.70%	80.40%	75.77%	75.80%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	67.90%	80.20%	75.26%	75.30%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	65%	67.70%	80.10%	75.15%	75.20%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	67.50%	80.10%	75.05%	75.10%
Undersample 40% Vacant	IsBeginMonth /EndMonth	MLP	't' (att+cls)+LR 0.4	80%	69.20%	81.00%	74.36%	74.30%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	68.60%	80.70%	73.89%	73.80%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	70.60%	76.90%	73.37%	73.30%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	67.80%	80.40%	73.19%	73.10%
Undersample 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	80%	67.20%	81.80%	73.19%	73.00%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	70.20%	76.70%	73.10%	73.00%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	70.40%	76.40%	73.10%	73.00%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	65%	70.20%	76.50%	72.97%	72.90%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	80%	67.30%	80.70%	72.96%	72.80%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	67.30%	80.70%	72.96%	72.80%
Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	80%	66.40%	81.50%	72.49%	72.30%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	80%	64.40%	83.80%	71.33%	70.80%

3 Classes – ZONE 16 (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	68.60%	52.00%	85.20%	72.51%	72.90%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	68.30%	51.70%	84.80%	72.04%	72.50%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	68.60%	51.30%	84.80%	72.04%	72.50%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	80%	67.70%	51.30%	84.50%	71.72%	72.10%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	67.90%	53.90%	81.50%	71.39%	71.50%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	68.30%	53.90%	81.20%	71.39%	71.50%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	67.70%	54.30%	81.40%	71.39%	71.50%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	65%	67.10%	54.30%	81.30%	71.12%	71.30%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	62.00%	64.60%	81.70%	69.94%	69.90%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	61.70%	65.10%	81.70%	69.94%	69.90%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	80%	61.70%	64.30%	81.70%	69.75%	69.70%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	61.70%	61.70%	64.30%	69.75%	80.10%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	60.40%	66.70%	80.70%	69.55%	69.50%
SMOTE Oversample - 100% Vacant	IsBeginMonth/ EndMonth	J48	Conf 0.35	80%	67.20%	51.50%	79.10%	69.51%	69.70%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	80%	67.20%	51.30%	77.70%	69.51%	69.40%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	65%	60.20%	66.00%	80.90%	69.33%	69.30%
SMOTE Oversample - 100% Vacant	IsBeginMonth/ EndMonth + Outlier	MLP	'i'attribs+LR 0.1	65%	62.70%	55.30%	79.30%	69.04%	69.10%
SMOTE Oversample - 100% Vacant	IsBeginMonth/ EndMonth + Outlier	MLP	'i'attribs+LR 0.1	80%	67.40%	49.70%	79.30%	69.04%	69.30%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	60.10%	66.30%	79.70%	69.01%	68.90%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	65%	65.40%	53.80%	77.10%	68.95%	68.80%

2 Classes – ZONE 16 (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	65.10%	82.70%	76.53%	76.60%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	64.90%	82.40%	76.33%	76.40%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	64.20%	82.30%	75.92%	76.00%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	80%	63.10%	82.20%	75.31%	75.50%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	71.30%	78.90%	75.00%	75.00%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	71.00%	78.50%	74.70%	74.70%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	71.00%	78.50%	74.70%	74.70%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	80%	71.30%	77.80%	74.56%	74.60%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	71.50%	76.50%	74.09%	74.10%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	80%	70.20%	78.20%	74.09%	74.10%
Undersample 40% Vacant	IsBeginMonth/ EndMonth	MLP	't' (att+cls)+LR 0.4	80%	73.80%	74.30%	74.09%	74.00%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	70.70%	76.90%	73.83%	73.90%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	70.50%	76.50%	73.58%	73.60%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	64.50%	78.10%	73.51%	73.30%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	64.10%	77.90%	73.28%	73.00%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	63.60%	77.60%	72.93%	72.70%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	80%	69.80%	75.60%	72.80%	72.80%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	65%	63.10%	77.60%	72.70%	72.50%
Undersample 100% Vacant	IsBeginMonth/ EndMonth	MLP	't' (att+cls)+LR 0.4	80%	58.70%	80.70%	72.45%	72.70%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	65%	68.70%	75.40%	72.04%	72.10%

3 Classes – Most Occupied Streets (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	68.60%	69.80%	80.40%	73.77%	73.80%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	68.60%	69.40%	80.50%	73.64%	73.70%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	80%	67.80%	69.70%	80.50%	73.51%	73.50%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	68.60%	69.10%	80.20%	73.38%	73.40%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	80%	67.50%	69.40%	79.20%	72.58%	72.60%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	80%	65.20%	72.60%	77.90%	71.79%	71.60%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	80%	64.10%	73.00%	76.60%	71.32%	71.20%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	80%	65.20%	72.30%	76.00%	71.16%	71.00%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	80%	65.10%	68.20%	78.20%	71.13%	71.20%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	80%	64.40%	67.60%	79.10%	70.86%	70.90%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	80%	65.20%	71.10%	77.20%	70.84%	70.60%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	80%	63.50%	67.80%	79.00%	70.73%	70.80%
SMOTE Oversample - 100% Vacant	NONE	MLP	'i'attribs+LR 0.1	80%	59.80%	69.30%	80.10%	70.60%	70.80%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	65.30%	68.00%	79.10%	70.05%	70.00%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	65.30%	67.80%	79.60%	70.05%	70.00%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	59.80%	67.40%	78.70%	69.89%	70.00%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	64.90%	67.80%	79.50%	69.89%	69.80%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	64.90%	67.80%	79.50%	69.89%	69.80%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	80%	64.50%	67.90%	79.00%	69.73%	69.70%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	80%	67.30%	66.20%	79.90%	69.57%	69.30%

2 Classes – Most Occupied Streets (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	77.60%	77.80%	77.70%	77.70%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	76.80%	77.60%	77.23%	77.20%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	80%	75.60%	77.30%	76.53%	76.50%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	70.50%	80.30%	76.26%	76.30%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	70.30%	80.00%	76.07%	76.10%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	70.30%	80.00%	76.07%	76.10%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	75.10%	76.90%	76.06%	76.10%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	80%	70.00%	79.90%	75.88%	75.90%
Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	80%	75.30%	73.80%	74.41%	74.30%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	72.00%	76.40%	74.40%	74.40%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	65%	71.80%	76.40%	74.26%	74.30%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	71.90%	76.10%	74.13%	74.10%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	71.50%	75.60%	73.73%	73.70%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	66.00%	78.10%	72.86%	73.00%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	65.90%	78.00%	72.75%	72.80%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	65%	65.70%	78.00%	72.64%	72.70%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	65.70%	78.00%	72.64%	72.70%
Undersample 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	65%	68.70%	75.40%	72.12%	72.20%
Undersample 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	80%	72.20%	71.50%	71.83%	71.70%
Undersample 40% Vacant	IsBeginMonth/EndMonth	MLP	't' (att+cls)+LR 0.4	80%	72.50%	70.90%	71.60%	71.50%

3 Classes – Less Occupied Streets (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	80%	68.90%	55.10%	82.60%	73.28%	73.30%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	68.60%	55.70%	82.10%	73.14%	73.10%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	68.30%	55.60%	82.20%	72.99%	73.00%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	68.00%	56.00%	82.20%	72.99%	73.00%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	66.40%	63.30%	80.90%	71.63%	71.60%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	65%	66.20%	62.30%	80.40%	71.11%	71.10%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	80%	66.20%	62.30%	80.40%	71.11%	71.10%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	65.60%	62.50%	80.40%	70.93%	70.90%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	69.00%	53.80%	77.70%	70.87%	70.60%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	68.70%	53.70%	77.80%	70.78%	70.60%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	65.40%	62.50%	80.10%	70.76%	70.70%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	68.70%	53.70%	77.70%	70.70%	70.50%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	65%	68.10%	52.80%	77.60%	70.20%	70.00%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	66.90%	60.30%	76.90%	69.73%	69.60%
SMOTE Oversample - 100% Vacant	NONE	MLP	'i'attribs+LR 0.1	80%	61.00%	56.40%	82.60%	69.64%	69.90%
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth	MLP	'i'attribs+LR 0.1	80%	63.10%	53.60%	82.00%	69.64%	70.00%
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	80%	61.80%	54.70%	83.40%	69.64%	70.10%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	67.00%	60.00%	76.80%	69.63%	69.50%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	67.20%	60.30%	76.10%	69.54%	69.40%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	80%	64.10%	56.20%	83.40%	68.86%	69.30%

2 Classes – Less Occupied Streets (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	65.90%	80.60%	75.80%	75.60%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	65.50%	80.50%	75.61%	75.40%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	80%	65.10%	80.40%	75.42%	75.30%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	80%	64.00%	79.90%	74.67%	74.50%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	80%	74.80%	73.60%	74.29%	74.10%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	65%	66.30%	77.50%	74.06%	73.60%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	66.50%	77.30%	74.06%	73.50%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	80%	74.10%	73.20%	73.78%	73.60%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	65%	66.10%	77.10%	73.74%	73.20%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	65%	65.60%	77.20%	73.63%	73.10%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	80%	73.80%	73.10%	73.52%	73.30%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	80%	73.50%	72.90%	73.26%	73.10%
Undersample 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	80%	74.10%	72.00%	73.26%	73.10%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	65%	72.00%	74.50%	72.98%	72.70%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	65%	70.50%	77.60%	72.98%	72.40%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	65%	71.70%	74.60%	72.83%	72.60%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	65%	71.70%	74.60%	72.83%	72.60%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	80%	71.40%	75.20%	72.75%	72.20%
Undersample 40% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.15	80%	72.50%	73.20%	72.75%	72.40%
Undersample 40% Vacant	NONE	J48	Conf 0.15	65%	71.30%	73.30%	72.10%	71.90%

3 Classes – Test 2 Week November (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	2 WEEK	43.10%	67.10%	48.10%	49.32%	40.80%
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	2 WEEK	48.30%	45.60%	50.20%	49.32%	43.50%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	2 WEEK	41.80%	67.10%	48.10%	49.13%	40.50%
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth	J48	Conf 0.35	2 WEEK	38.80%	48.40%	51.30%	48.95%	46.20%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.35	2 WEEK	37.90%	47.80%	51.40%	48.86%	45.70%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	2 WEEK	42.70%	62.40%	47.80%	48.68%	40.80%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	2 WEEK	40.00%	66.30%	47.80%	48.68%	40.00%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	2 WEEK	37.80%	51.90%	49.20%	48.31%	44.90%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	2 WEEK	37.90%	47.00%	50.30%	48.22%	44.60%
SMOTE Oversample - 100% Vacant	NONE	MLP	'i'attribs+LR 0.1	2 WEEK	48.60%	45.60%	48.50%	48.04%	42.10%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth	J48	Conf 0.35	2 WEEK	50.30%	59.90%	42.20%	47.24%	45.20%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	2 WEEK	46.30%	41.20%	47.60%	46.94%	38.30%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	2 WEEK	48.00%	67.10%	41.30%	46.05%	41.50%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.35	2 WEEK	43.70%	60.90%	41.60%	46.05%	44.00%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	2 WEEK	50.00%	70.80%	40.60%	45.84%	40.50%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	2 WEEK	42.00%	53.30%	44.40%	45.73%	42.50%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	2 WEEK	47.30%	62.20%	39.80%	44.76%	42.20%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	2 WEEK	43.70%	70.30%	40.40%	44.54%	38.70%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	2 WEEK	50.00%	68.30%	39.90%	44.54%	38.90%
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth	MLP	'i'attribs+LR 0.1	2 WEEK	38.50%	35.20%	48.30%	44.47%	40.30%

2 Classes – Test 2 Week November (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.15	FULL - 2 WEEK	62.90%	68.90%	67.59%	65.10%
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	FULL - 2 WEEK	61.30%	68.40%	66.83%	64.30%
Undersample 100% Vacant	IsBeginMonth /EndMonth	J48	Conf 0.15	FULL - 2 WEEK	59.80%	68.00%	66.20%	63.60%
Undersample 100% Vacant	NONE	J48	Conf 0.15	FULL - 2 WEEK	58.80%	67.90%	65.83%	63.30%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	FULL - 2 WEEK	57.80%	67.90%	65.57%	63.30%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	FULL - 2 WEEK	71.60%	60.10%	63.24%	61.20%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	FULL - 2 WEEK	66.70%	61.20%	63.08%	62.20%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	FULL - 2 WEEK	54.30%	64.10%	62.93%	56.30%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	FULL - 2 WEEK	52.50%	64.00%	62.55%	56.40%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	FULL - 2 WEEK	52.50%	64.00%	62.55%	56.30%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	FULL - 2 WEEK	69.30%	59.70%	62.44%	60.60%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	FULL - 2 WEEK	51.80%	64.20%	62.42%	56.90%
Undersample 100% Vacant	IsBeginMonth/ EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	FULL - 2 WEEK	50.80%	67.50%	62.42%	61.40%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	FULL - 2 WEEK	50.80%	65.80%	62.30%	59.80%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	FULL - 2 WEEK	68.90%	59.60%	62.28%	60.40%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	FULL - 2 WEEK	69.20%	59.40%	62.12%	60.10%
Undersample 40% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.15	FULL - 2 WEEK	62.10%	62.10%	62.12%	62.00%
Undersample 100% Vacant	IsBeginMonth /EndMonth	MLP	't' (att+cls)+LR 0.4	FULL - 2 WEEK	49.40%	64.70%	61.66%	58.10%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	FULL - 2 WEEK	61.40%	61.50%	61.48%	61.30%
Undersample 40% Vacant	IsBeginMonth /EndMonth	J48	Conf 0.15	FULL - 2 WEEK	61.70%	61.00%	61.32%	61.10%

3 Classes – 3 Week – 1 Week October - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth	MLP	'i'attribs+LR 0.1	OCT - 3 WEEK	29.00%	45.10%	69.60%	58.76%	55.40%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 3 WEEK	32.80%	49.40%	67.00%	56.48%	54.30%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	32.30%	50.60%	67.20%	56.48%	54.60%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	31.10%	50.00%	66.70%	55.86%	53.90%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	15.00%	41.80%	61.10%	55.80%	48.20%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	15.00%	41.80%	61.10%	55.80%	48.20%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	OCT - 3 WEEK	13.60%	43.10%	61.20%	55.53%	48.40%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 3 WEEK	14.30%	41.80%	61.00%	55.53%	48.00%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	OCT - 3 WEEK	30.60%	49.40%	66.30%	55.25%	53.40%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth	MLP	'i'attribs+LR 0.1	OCT - 3 WEEK	38.00%	48.10%	67.70%	54.63%	54.60%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	OCT - 3 WEEK	25.40%	39.70%	72.10%	53.10%	53.10%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	OCT - 3 WEEK	25.00%	39.70%	72.00%	52.83%	52.80%
SMOTE Oversample - 100% Vacant	IsBeginMonth/ EndMonth + Outlier	MLP	'i'attribs+LR 0.1	OCT - 3 WEEK	27.70%	37.80%	70.20%	52.83%	52.90%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth/ EndMonth	J48	Conf 0.35	OCT - 3 WEEK	25.40%	38.40%	70.30%	52.56%	52.30%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	OCT - 3 WEEK	33.70%	41.10%	69.30%	52.16%	52.10%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	OCT - 3 WEEK	23.10%	38.70%	72.00%	51.75%	52.20%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	OCT - 3 WEEK	30.60%	32.90%	70.50%	51.48%	51.80%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	OCT - 3 WEEK	21.20%	47.40%	63.20%	50.00%	48.90%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	OCT - 3 WEEK	21.20%	47.40%	63.20%	50.00%	48.90%
SMOTE Oversample - 40% Vacant	Outlier	MLP	'i'attribs+LR 0.1	OCT - 3 WEEK	18.80%	39.80%	63.40%	50.00%	47.10%

2 Classes – 3 Week – 1 Week October - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 3 WEEK	55.80%	74.40%	71.58%	68.20%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	OCT - 3 WEEK	54.50%	74.30%	71.23%	68.00%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	54.50%	74.30%	71.23%	68.00%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	53.30%	74.20%	70.88%	67.70%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	49.50%	79.80%	69.47%	70.00%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	58.90%	73.90%	69.33%	68.70%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 3 WEEK	57.70%	73.10%	68.49%	67.70%
Undersample 100% Vacant	IsBeginMonth/EndMonth	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	48.30%	82.20%	68.42%	69.50%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 3 WEEK	56.10%	71.50%	67.23%	66.10%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	OCT - 3 WEEK	54.80%	72.10%	66.81%	66.10%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	51.90%	76.90%	65.55%	66.20%
Undersample 100% Vacant	IsBeginMonth/EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	43.80%	78.60%	64.91%	66.00%
Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	50.80%	78.80%	64.71%	65.40%
Undersample 40% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.15	OCT - 3 WEEK	50.50%	74.50%	64.29%	64.80%
Undersample 40% Vacant	IsBeginMonth/EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	50.50%	73.10%	64.29%	64.60%
Undersample 100% Vacant	NONE	J48	Conf 0.15	OCT - 3 WEEK	42.50%	77.90%	63.86%	65.00%
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	OCT - 3 WEEK	42.50%	77.90%	63.86%	65.00%
Undersample 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.15	OCT - 3 WEEK	42.50%	77.90%	63.86%	65.00%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	OCT - 3 WEEK	41.40%	75.80%	63.86%	64.50%
Undersample 40% Vacant	NONE	J48	Conf 0.15	OCT - 3 WEEK	49.50%	74.10%	63.45%	64.00%

3 Classes – 3 Week – 1 Week November - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	NOV - 3 WEEK	50.00%	38.00%	56.00%	49.54%	49.00%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	NOV - 3 WEEK	48.40%	39.30%	55.40%	49.02%	48.50%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	NOV - 3 WEEK	48.70%	38.00%	54.60%	48.63%	48.00%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	NOV - 3 WEEK	48.40%	38.00%	54.50%	48.50%	47.80%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	33.30%	49.50%	42.90%	43.79%	33.00%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	NOV - 3 WEEK	25.00%	49.10%	42.80%	43.66%	33.00%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	NOV - 3 WEEK	55.60%	41.30%	43.10%	43.53%	36.70%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	NOV - 3 WEEK	15.70%	31.50%	36.20%	43.08%	43.10%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	NOV - 3 WEEK	70.80%	43.90%	41.60%	42.75%	33.60%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	NOV - 3 WEEK	16.70%	44.30%	42.40%	42.48%	31.20%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	16.70%	43.90%	42.40%	42.35%	31.10%
SMOTE Oversample - 100% Vacant	NONE	MLP	'i'attribs+LR 0.1	NOV - 3 WEEK	41.90%	40.50%	41.80%	41.70%	32.00%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth + Outlier	MLP	'i'attribs+LR 0.1	NOV - 3 WEEK	46.20%	38.90%	41.80%	41.44%	33.30%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	NOV - 3 WEEK	54.80%	36.30%	38.50%	41.02%	39.90%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	NOV - 3 WEEK	54.80%	36.30%	38.50%	41.02%	39.90%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	NOV - 3 WEEK	52.90%	35.80%	38.90%	40.54%	39.10%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	55.60%	56.30%	30.80%	37.04%	30.80%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	NOV - 3 WEEK	58.50%	54.80%	30.40%	36.73%	31.60%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	NOV - 3 WEEK	61.50%	54.50%	30.10%	35.93%	30.50%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	55.60%	52.10%	29.90%	35.29%	29.00%

2 Classes – 3 Week – 1 Week November - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	NOV - 3 WEEK	64.70%	66.30%	65.86%	64.40%
Undersample 100% Vacant	ExistSpecialEvent + IsBeginMonth/ EndMonth	J48	Conf 0.15	NOV - 3 WEEK	64.20%	66.60%	65.86%	64.60%
Undersample 100% Vacant	NONE	J48	Conf 0.15	NOV - 3 WEEK	64.70%	66.10%	65.67%	64.00%
Undersample 100% Vacant	IsBeginMonth/ EndMonth	J48	Conf 0.15	NOV - 3 WEEK	64.20%	66.30%	65.67%	64.30%
Undersample 40% Vacant	NONE	J48	Conf 0.15	NOV - 3 WEEK	70.60%	55.70%	63.00%	63.20%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	NOV - 3 WEEK	70.30%	55.10%	62.50%	62.70%
Undersample 40% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.15	NOV - 3 WEEK	69.80%	55.20%	62.50%	62.70%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	NOV - 3 WEEK	76.70%	53.90%	62.25%	61.80%
Undersample 40% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.15	NOV - 3 WEEK	69.50%	54.70%	62.00%	62.20%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	NOV - 3 WEEK	76.40%	53.10%	61.25%	60.60%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	75.40%	52.30%	60.25%	59.50%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	61.20%	59.90%	60.07%	53.30%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	NOV - 3 WEEK	58.70%	59.90%	59.70%	53.50%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	74.80%	50.90%	58.50%	57.30%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	NOV - 3 WEEK	55.10%	58.50%	58.21%	49.40%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV - 3 WEEK	53.70%	58.50%	58.02%	49.70%
Undersample 100% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	NOV - 3 WEEK	51.90%	59.50%	58.02%	53.80%
Undersample 100% Vacant	IsBeginMonth/ EndMonth	MLP	't' (att+cls)+LR 0.4	NOV - 3 WEEK	50.60%	60.70%	57.65%	56.10%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	NOV - 3 WEEK	48.20%	59.70%	56.16%	54.70%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	NOV - 3 WEEK	46.00%	58.30%	55.41%	52.10%

3 Classes – 3 Week – 1 Week December - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	46.20%	46.70%	58.70%	47.59%	37.10%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 3 WEEK	33.80%	49.60%	41.80%	46.05%	42.40%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	DEZ - 3 WEEK	32.40%	49.50%	41.80%	45.88%	42.00%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	28.60%	45.40%	48.10%	44.85%	35.90%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth + Outlier	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	22.70%	45.80%	51.60%	44.67%	37.40%
SMOTE Oversample - 100% Vacant	NONE	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	29.50%	50.20%	28.30%	44.16%	38.10%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 3 WEEK	28.20%	47.70%	41.80%	43.64%	40.70%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	DEZ - 3 WEEK	26.80%	47.60%	41.80%	43.47%	40.30%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 3 WEEK	41.70%	36.90%	63.00%	41.49%	35.10%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	42.40%	38.10%	58.80%	40.87%	35.40%
SMOTE Oversample - 40% Vacant	Outlier	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	44.00%	39.40%	40.40%	40.04%	34.60%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth + Outlier	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	50.90%	37.80%	43.60%	40.04%	34.00%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 3 WEEK	43.50%	36.20%	58.30%	39.83%	32.90%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 3 WEEK	42.90%	35.90%	58.30%	39.63%	33.30%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 3 WEEK	40.90%	36.00%	57.70%	39.42%	32.20%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 3 WEEK	36.40%	37.40%	45.20%	39.21%	37.40%
SMOTE Oversample - 40% Vacant	NONE	MLP	'i'attribs+LR 0.1	DEZ - 3 WEEK	39.50%	35.00%	50.90%	37.14%	30.60%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	DEZ - 3 WEEK	37.60%	37.00%	33.70%	36.51%	33.80%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 3 WEEK	36.40%	34.30%	30.90%	34.02%	31.10%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 3 WEEK	27.30%	28.90%	95.10%	33.23%	23.40%

2 Classes – 3 Week – 1 Week December - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	DEZ - 3 WEEK	57.70%	64.90%	63.98%	57.90%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 3 WEEK	55.60%	62.90%	62.56%	51.60%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 3 WEEK	50.00%	62.60%	62.09%	51.00%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 3 WEEK	50.00%	62.70%	62.09%	51.30%
Undersample 100% Vacant	NONE	J48	Conf 0.15	DEZ - 3 WEEK	49.20%	64.00%	61.85%	56.30%
Undersample 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.15	DEZ - 3 WEEK	49.20%	64.00%	61.85%	56.30%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 3 WEEK	43.80%	62.30%	61.61%	50.00%
Undersample 100% Vacant	IsBeginMonth EndMonth	MLP	't' (att+cls)+LR 0.4	DEZ - 3 WEEK	46.00%	64.20%	60.43%	57.00%
Undersample 100% Vacant	ExistSpecialEvent + IsBeginMonth EndMonth	J48	Conf 0.15	DEZ - 3 WEEK	46.50%	65.30%	60.19%	58.50%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	DEZ - 3 WEEK	45.40%	64.60%	59.72%	57.60%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	DEZ - 3 WEEK	62.80%	55.10%	57.14%	54.70%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	DEZ - 3 WEEK	41.00%	63.30%	56.87%	55.40%
Undersample 100% Vacant	IsBeginMonth EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	DEZ - 3 WEEK	37.90%	62.10%	56.64%	53.50%
Undersample 40% Vacant	IsBeginMonth/EndMonth + Outlier	MLP	't' (att+cls)+LR 0.4	DEZ - 3 WEEK	60.00%	55.00%	56.52%	54.90%
Undersample 40% Vacant	IsBeginMonth EndMonth	MLP	't' (att+cls)+LR 0.4	DEZ - 3 WEEK	60.00%	54.30%	55.90%	53.60%
Undersample 40% Vacant	ExistSpecialEvent + IsBeginMonth EndMonth	J48	Conf 0.15	DEZ - 3 WEEK	61.90%	53.30%	54.97%	50.40%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 3 WEEK	59.70%	52.70%	54.04%	49.20%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 3 WEEK	61.20%	52.40%	53.73%	47.30%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 3 WEEK	58.10%	52.30%	53.42%	48.50%
Undersample 40% Vacant	NONE	J48	Conf 0.15	DEZ - 3 WEEK	55.00%	52.10%	52.80%	49.60%

3 Classes – 1 Week – 1 Day October - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 40% Vacant	IsBeginMonth EndMonth	MLP	'i'attribs+LR 0.1	OCT - 1 WEEK	35.70%	75.00%	55.10%	56.63%	53.70%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	OCT - 1 WEEK	30.00%	62.50%	53.70%	54.22%	49.50%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	OCT - 1 WEEK	30.00%	62.50%	53.70%	54.22%	49.50%
SMOTE Oversample - 40% Vacant	IsBeginMonth EndMonth	J48	Conf 0.35	OCT - 1 WEEK	30.00%	62.50%	53.70%	54.22%	49.50%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth EndMonth	J48	Conf 0.35	OCT - 1 WEEK	30.00%	62.50%	53.70%	54.22%	49.50%
SMOTE Oversample - 40% Vacant	IsBeginMonth EndMonth + Outlier	MLP	'i'attribs+LR 0.1	OCT - 1 WEEK	25.00%	60.00%	58.70%	54.22%	49.80%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 1 WEEK	16.70%	66.70%	54.00%	52.53%	42.30%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 1 WEEK	16.70%	66.70%	54.00%	52.53%	42.30%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	OCT - 1 WEEK	25.00%	30.40%	84.10%	52.53%	53.80%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	OCT - 1 WEEK	25.00%	30.40%	84.10%	52.53%	53.80%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	OCT - 1 WEEK	25.00%	30.40%	84.10%	52.53%	53.80%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth EndMonth	J48	Conf 0.35	OCT - 1 WEEK	25.00%	30.40%	84.10%	52.53%	53.80%
SMOTE Oversample - 100% Vacant	IsBeginMonth EndMonth	MLP	'i'attribs+LR 0.1	OCT - 1 WEEK	33.30%	42.30%	75.00%	52.53%	53.80%
SMOTE Oversample - 40% Vacant	NONE	MLP	'i'attribs+LR 0.1	OCT - 1 WEEK	25.00%	57.10%	55.80%	51.81%	47.90%
SMOTE Oversample - 100% Vacant	IsBeginMonth EndMonth + Outlier	MLP	'i'attribs+LR 0.1	OCT - 1 WEEK	0.00%	50.00%	52.60%	50.51%	42.70%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 1 WEEK	9.10%	66.70%	52.40%	48.48%	40.40%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	OCT - 1 WEEK	20.00%	62.50%	46.80%	48.19%	40.80%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 1 WEEK	42.90%	55.60%	46.60%	48.19%	43.10%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	OCT - 1 WEEK	8.30%	66.70%	51.90%	47.47%	39.80%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	OCT - 1 WEEK	40.00%	28.60%	54.00%	46.46%	44.20%

2 Classes – 1 Week – 1 Day October -

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	OCT - 1 WEEK	72.00%	78.80%	75.86%	75.90%
Undersample 100% Vacant	NONE	J48	Conf 0.15	OCT - 1 WEEK	57.70%	79.20%	71.62%	71.80%
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	OCT - 1 WEEK	57.70%	79.20%	71.62%	71.80%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	OCT - 1 WEEK	68.20%	72.20%	70.69%	70.40%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 1 WEEK	68.20%	72.20%	70.69%	70.40%
Undersample 40% Vacant	NONE	J48	Conf 0.15	OCT - 1 WEEK	65.20%	71.40%	68.97%	68.80%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	OCT - 1 WEEK	65.20%	71.40%	68.97%	68.80%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	OCT - 1 WEEK	53.60%	78.30%	68.92%	69.30%
Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	OCT - 1 WEEK	61.90%	67.60%	65.52%	65.00%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	OCT - 1 WEEK	45.50%	68.30%	64.86%	60.20%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	OCT - 1 WEEK	45.50%	68.30%	64.86%	60.20%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 1 WEEK	45.50%	68.30%	64.86%	60.20%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 1 WEEK	45.50%	68.30%	64.86%	60.20%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	OCT - 1 WEEK	60.00%	62.80%	62.07%	59.80%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	OCT - 1 WEEK	60.00%	62.80%	62.07%	59.80%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	OCT - 1 WEEK	39.10%	68.60%	59.46%	59.00%

3 Classes – 1 Week – 1 Day November - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	NOV -1 WEEK	62.10%	0.00%	56.20%	57.63%	45.70%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	NOV -1 WEEK	46.30%	0.00%	73.00%	54.24%	48.30%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.35	NOV -1 WEEK	46.30%	0.00%	73.00%	54.24%	48.30%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	NOV -1 WEEK	52.90%	0.00%	53.60%	53.39%	41.60%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	52.90%	0.00%	53.60%	53.39%	41.60%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	NOV -1 WEEK	46.30%	0.00%	72.20%	53.39%	47.90%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	NOV -1 WEEK	46.30%	0.00%	72.20%	53.39%	47.90%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	NOV -1 WEEK	53.80%	0.00%	52.40%	51.02%	39.10%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	NOV -1 WEEK	53.80%	0.00%	52.40%	51.02%	39.10%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth	J48	Conf 0.35	NOV -1 WEEK	53.80%	0.00%	52.40%	51.02%	39.10%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.35	NOV -1 WEEK	53.80%	0.00%	52.40%	51.02%	39.10%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	NOV -1 WEEK	25.00%	75.00%	46.50%	45.76%	45.90%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	33.30%	0.00%	45.50%	44.92%	31.60%
SMOTE Oversample - 100% Vacant	NONE	MLP	'i'attribs+LR 0.1	NOV -1 WEEK	13.30%	69.20%	46.70%	44.92%	40.90%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	NOV -1 WEEK	13.00%	66.70%	43.80%	40.68%	38.90%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	NOV -1 WEEK	57.90%	0.00%	32.90%	36.73%	33.90%
SMOTE Oversample - 100% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	NOV -1 WEEK	22.20%	0.00%	43.20%	36.44%	26.90%
SMOTE Oversample - 40% Vacant	IsBeginMonth/ EndMonth + Outlier	MLP	'i'attribs+LR 0.1	NOV -1 WEEK	69.20%	0.00%	32.90%	35.71%	38.30%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	56.30%	0.00%	31.60%	34.69%	32.80%
SMOTE Oversample - 40% Vacant	NONE	MLP	'i'attribs+LR 0.1	NOV -1 WEEK	0.00%	60.00%	32.90%	34.69%	27.30%

2 Classes – 1 Week – 1 Day November

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	NOV -1 WEEK	64.30%	75.90%	72.09%	71.70%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	NOV -1 WEEK	59.00%	80.90%	70.93%	71.40%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	100.00%	65.10%	66.28%	55.90%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	NOV -1 WEEK	100.00%	65.10%	66.28%	55.90%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	100.00%	65.10%	66.28%	55.90%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	NOV -1 WEEK	50.00%	64.10%	62.79%	55.00%
Undersample 100% Vacant	NONE	J48	Conf 0.15	NOV -1 WEEK	0.00%	62.80%	62.79%	48.40%
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	NOV -1 WEEK	0.00%	62.80%	62.79%	48.40%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	NOV -1 WEEK	48.80%	73.30%	61.63%	62.20%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	45.70%	68.60%	59.30%	59.60%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	NOV -1 WEEK	39.10%	63.50%	56.98%	2.40%
Undersample 40% Vacant	NONE	J48	Conf 0.15	NOV -1 WEEK	44.40%	70.70%	56.98%	57.70%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	NOV -1 WEEK	44.40%	70.70%	56.98%	57.70%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	NOV -1 WEEK	41.40%	64.90%	56.98%	56.50%
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	NOV -1 WEEK	37.90%	63.20%	54.65%	54.10%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	NOV -1 WEEK	33.30%	60.70%	51.16%	50.80%

3 Classes – 1 Week – 1 Day December - (Best 20)

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	ALMOST FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision	Precision		
SMOTE Oversample - 40% Vacant	Outlier	MLP	'i'attribs+LR 0.1	DEZ - 1 WEEK	50.00%	61.10%	100.00%	70.00%	70.60%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	MLP	'i'attribs+LR 0.1	DEZ - 1 WEEK	38.10%	69.20%	93.80%	64.00%	65.80%
SMOTE Oversample - 100% Vacant	NONE	J48	Conf 0.35	DEZ - 1 WEEK	14.30%	42.90%	85.70%	60.71%	60.80%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent	J48	Conf 0.35	DEZ - 1 WEEK	14.30%	42.90%	85.70%	60.71%	60.80%
SMOTE Oversample - 100% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 1 WEEK	14.30%	42.90%	85.70%	60.71%	60.80%
SMOTE Oversample - 100% Vacant	ExistSpecialEvent + IsBeginMonth /EndMonth	J48	Conf 0.35	DEZ - 1 WEEK	14.30%	42.90%	85.70%	60.71%	60.80%
SMOTE Oversample - 40% Vacant	NONE	J48	Conf 0.35	DEZ - 1 WEEK	0.00%	47.60%	82.60%	58.00%	56.30%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent	J48	Conf 0.35	DEZ - 1 WEEK	0.00%	47.60%	82.60%	58.00%	56.30%
SMOTE Oversample - 40% Vacant	IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 1 WEEK	0.00%	47.60%	82.60%	58.00%	56.30%
SMOTE Oversample - 40% Vacant	ExistSpecialEvent + IsBeginMonth/EndMonth	J48	Conf 0.35	DEZ - 1 WEEK	0.00%	47.60%	82.60%	58.00%	56.30%
SMOTE Oversample - 100% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	16.70%	69.80%	57.14%	50.90%
SMOTE Oversample - 100% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	16.70%	85.70%	57.14%	56.40%
SMOTE Oversample - 100% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	16.70%	69.80%	57.14%	50.90%
SMOTE Oversample - 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	16.70%	69.80%	57.14%	50.90%
SMOTE Oversample - 40% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	12.50%	81.80%	56.00%	52.00%
SMOTE Oversample - 40% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	12.50%	81.80%	56.00%	52.00%
SMOTE Oversample - 40% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	11.10%	81.80%	56.00%	51.90%
SMOTE Oversample - 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	0.00%	12.50%	81.80%	56.00%	52.00%
SMOTE Oversample - 40% Vacant	IsBeginMonth /EndMonth + Outlier	MLP	'i'attribs+LR 0.1	DEZ - 1 WEEK	10.50%	12.50%	100.00%	38.00%	45.30%
SMOTE Oversample - 100% Vacant	Outlier	MLP	'i'attribs+LR 0.1	DEZ - 1 WEEK	20.00%	24.40%	100.00%	32.14%	28.90%

2 Classes – 1 Week – 1 Day December

Dataset	Removed Attributes	Algorithm	Parameters	Train	FULL	VACANT	Correct Instances	Weighted Average F1-Score
					Precision	Precision		
Undersample 40% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 1 WEEK	83.30%	81.80%	82.05%	80.10%
Undersample 40% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	83.30%	81.80%	82.05%	80.10%
Undersample 40% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 1 WEEK	83.30%	81.80%	82.05%	80.10%
Undersample 40% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	83.30%	81.80%	82.05%	80.10%
Undersample 100% Vacant	IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	60.00%	85.70%	80.00%	79.70%
Undersample 100% Vacant	Outlier	RandomForest	Interation = 1000	DEZ - 1 WEEK	54.50%	85.30%	77.78%	77.80%
Undersample 100% Vacant	Outlier + IsHoliday	RandomForest	Interation = 1000	DEZ - 1 WEEK	54.50%	85.30%	77.78%	77.80%
Undersample 40% Vacant	NONE	J48	Conf 0.15	DEZ - 1 WEEK	58.30%	85.20%	76.92%	77.20%
Undersample 40% Vacant	ExistSpecialEvent	J48	Conf 0.15	DEZ - 1 WEEK	58.30%	85.20%	76.92%	77.20%
Undersample 100% Vacant	NONE	RandomForest	Interation = 1000	DEZ - 1 WEEK	50.00%	84.80%	75.56%	75.90%
Undersample 40% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	DEZ - 1 WEEK	47.80%	100.00%	69.23%	70.50%
Undersample 40% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	DEZ - 1 WEEK	44.00%	100.00%	64.10%	65.10%
Undersample 100% Vacant	NONE	J48	Conf 0.15	DEZ - 1 WEEK	35.00%	84.00%	62.22%	64.80%
Undersample 100% Vacant	ExistSpecialEvent	J48	Conf 0.15	DEZ - 1 WEEK	35.00%	84.00%	62.22%	64.80%
Undersample 100% Vacant	NONE	MLP	't' (att+cls)+LR 0.4	DEZ - 1 WEEK	34.40%	100.00%	53.33%	54.30%
Undersample 100% Vacant	Outlier	MLP	't' (att+cls)+LR 0.4	DEZ - 1 WEEK	28.90%	100.00%	40.00%	36.80%