A Recommendation System based on Clustering and Classification for Optimal Trajectory Analysis

by

Rami Ibrahim

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Information Technology

in

Digital Media

Carleton University
Ottawa, Ontario

Abstract

Moving objects such as people, animals, and vehicles have generated a huge amount of spatiotemporal data by using location-capture technologies and mobile devices. There is a high demand to analyze this collected data and extract the desired knowledge. In this study, we built a recommendation system based on four data mining techniques which are clustering, classification, sequential pattern mining, and time series analysis. We have focused on predicting traffic status in an effective way by considering the trip destination which can be useful for passengers. We applied clustering and sequential pattern mining to detect taxi trips movement in different areas, then we applied the Naïve Bayes classifier to predict the traffic status of each trip. With the real taxi trips data of 441 taxis, we performed qualitative and quantitative analysis for our clustering method, then we evaluated the accuracy of the classification models. The results show that our recommendation system can achieve 70% accuracy in predicting traffic status. Additionally, it can detect 37 taxi movement patterns and 146 sequential patterns in different areas of the city. Based on the proposed approach, this system can be integrated with current traffic control applications to provide useful guidelines for detecting taxi trips movement and predicting origin, destination, and future route.

Acknowledgements

Foremost, I would like to thank my advisor Prof. Omair Shafiq for his continuous support during my master's study and research. I appreciate his patience, motivation, and deep knowledge. I could not have finished my research and writing of this thesis without his guidance and enthusiasm.

I would like to present my gratitude to my deceased parents for their great sacrifices for me and my brothers. Many thanks for my brothers Hamid and Adham for their support and for being true brothers when I needed them.

Finally, I would like to express my deep thankfulness to my wife Marwa, for her patience over the last two years. I could not be able to finish my research without her support. Thank you my cute little daughter Leen, and my kids Yamen and Mohammed for being good with your mother while I was studying at school.

Table of Contents

Abstract **1**

Acknowledgements **3**

Table of Contents **4**

List of Tables **6**

List of Figures **7**

Chapter 1: Introduction **9**

1.1 Trajectory Data 9

1.2 Trajectory Mining Application Areas 9

1.3 Taxi Trajectory Data 10

1.4 Trajectory Mining Techniques 11

1.5 Trajectory Mining Challenges 13

Chapter 2: Literature Review **15**

2.1 Trajectory Clustering 15

2.2 Time Series Analysis, Forecasting, and Anomaly Detection 17

2.3 Sequential Pattern Mining 19

2.4 Trajectory Classification 20

2.5 Trajectory Data Visualization 23

2.6 Comparative Analysis 25

Chapter 3: Methodology **27**

3.1 Data Description 27

3.2 Data Preprocessing 28

3.3 Descriptive Analysis 30

3.4 HDBSCAN Spatiotemporal Clustering 38

3.5 Trajectory Classification 44
3.6 Sequential Pattern Mining 61
3.7 Time Series Analysis 65
3.8 Interactive Visualization 71

Chapter 4: Evaluation 75
4.1 Clustering Evaluation 75
4.2 Classification Evaluation 93
4.3 Results Verification 96

Chapter 5: Implications 98
5.1 Transportation Recommender System 98
5.2 Urban Computing 100
5.3 Pollution and Energy Consumption 100
5.4 Anomalies Detection 101
5.5 Human and Machine Intelligence Combination 102
5.6 Business Potentials 102

Chapter 6: Limitations and Future Work 104
6.1 Clustering Improvement 104
6.2 Enhancement of Classifier Accuracy 105
6.3 Anomaly Detection 106
6.4 Use of Interactive Visualization 107

Chapter 7: Conclusions 108

Bibliography 112

List of Tables

Table 1. Previous trajectory analysis approaches 25

Table 2. Part of the preprocessed dataset 30

Table 3. Porto city districts border coordinates 49

Table 4. Traffic status values distribution 56

Table 5. Degree of membership thresholds for 3 CSV files 56

Table 6. Structure for each CSV file before classification 57

Table 7. Trip sequence distribution over time 62

Table 8. Patterns generated by SPADE algorithm 64

Table 9. Time series sequence for a given trip 66

Table 10. Silhouette coefficient for 3 datasets 93

Table 11. NB Confusion Matrix for origin traffic status 94

Table 12. RF Confusion Matrix for origin traffic status 94

Table 13. NB Confusion Matrix for destination traffic status 95

Table 14. RF Confusion Matrix for destination traffic status 95

Table 15. NB Confusion Matrix for route traffic status 96

Table 16. RF Confusion Matrix for route traffic status 96

List of Figures

Figure 1. Number of trips per call type 32

Figure 2. Number of trips per weekdays 32

Figure 3. Number of trips per month 33

Figure 4. Number of trips in 24 hours 34

Figure 5. Average duration of trips per call type 35

Figure 6. Average duration of trips per weekdays 36

Figure 7. Average duration of trips per month 37

Figure 8. Average duration of trips in 24 hours 37

Figure 9. Spatial distribution of taxi trips based on start locations 39

Figure 10. Spatiotemporal clustering of taxi trips dataset 41

Figure 11. K-means clustering of Porto taxi trips 42

Figure 12. HDBSCAN clustering Pseudo-Code (Campello Ricardo J. G. B, 2013) 44

Figure 13. Porto city districts in Google Maps (Google Inc., 2018) 48

Figure 14. Cluster #7 trips distribution 53

Figure 15. Cluster #4 trips distribution 54

Figure 16. Heatmap for origin traffic status in districts 59

Figure 17. Heatmap for destination traffic status in districts 60

Figure 18. Taxi trips patterns over districts (Pereira, 2018) 64

Figure 19. Taxi trips for trips started on Friday 67

Figure 20. Taxi trips for trips started on Friday 67

Figure 21. WSS plot with an elbow 69

Figure 22. Hierarchical clustering dendrograms 70

Figure 23. Trips on Friday morning from Sao Nicolau to Se..... 70

Figure 24. Trips on Saturday night from Do Ouro to Massaleros..... 71

Figure 25. R Shiny slide for sequential pattern mining..... 72

Figure 26. R Shiny slide for time series clustering 74

Figure 27. Number of clusters in 24 hours..... 75

Figure 28. Number of clusters on Wednesday in 24 hours 76

Figure 29. Number of clusters on Saturday in 24 hours..... 77

Figure 30. Clusters visualization for one timeframe (Tableau Software Inc., 2018)..... 78

Figure 31. High traffic heading toward the city airport at 2:00 AM 81

Figure 32. Traffic heading from center areas toward city park at 8:00 PM 83

Figure 33. High traffic in Matosinhos at 3:00 PM 83

Figure 34. High traffic detected in the central area on Monday at 10:00 AM 85

Figure 35. High traffic heading north via on Wednesday at 11:00 AM..... 87

Figure 36. High traffic in Matosinhos on Friday at 3:00 PM..... 89

Figure 37. Trips heading to Gaia via Luis I bridge 90

Chapter 1: Introduction

1.1 Trajectory Data

A spatiotemporal trajectory is a timestamped sequence generated by tracking the location of a moving object like humans, animals, vehicles, and tornados (Z. Feng, 2016). This sequence is represented by a series of space and time instances. A wide number of real-life applications are using positioning services technologies such as Global Position Systems (GPS) and Radio Frequency Identification (RFID) to collect trajectory data. Trajectory data mining can uncover the behavior of moving objects by providing valuable information for many application areas like social networks, transportation systems, environmental planning, and urban computing.

1.2 Trajectory Mining Application Areas

The extracted knowledge from trajectory data mining can be beneficial in multiple application areas. It can help transportation management systems to discover hotspots by studying the behavior of moving objects. With further analysis, we can predict the trajectory movement and make some recommendations for taxis and buses (Xiaolong Li, 2012). By mining trajectories, we can identify the nature of regions and the connectivity between them which helps in exploring urban boundaries and improves urban planning (Jean Damascène Mazimpaka, 2015). In addition, trajectory mining can help the city to detect road networks by tracing the movement of people in the city (Zheng Yu C. L., 2014). It can translate collected geospatial coordinates and transform them into text information such as Points of Interests (POIs). Identifying regions and moving vehicles behavior can help to address areas with high energy consumption, thus, can provide an obvious indication of the pollution level in these areas (Zheng Yu L. F.-P., 2013), (Shang Jingbo,

2014). Additionally, mining people's movement and studying their patterns can be beneficial for identifying ideal business locations and advertisement spots. A critical application of trajectory mining is to detect places and objects that can cause threats to public security. Trajectory analysis can be extended to predict these threats by monitoring objects movement and stops, and by identifying any outliers or anomalies in their movements (Felipe Pinto da Silva, 2015). In addition, capturing people's movement by using GPS devices can be utilized to uncover useful information such as mode of transport (e.g. walk, bike, bus, car) and locations of significance (Lin Miao, 2014). When the semantic description is added along with trajectory data, activities associated with each location can be recognized. For example, we can recognize restaurants, shopping centers, workplaces, and other activities. in addition to the GPS devices, RFID has been used in a wide number of applications like warehouse management (Ray Y. Zhong, 2015). These sensors are installed on shopping carts to collect manufacturing data and discover trajectory patterns for shoppers.

1.3 Taxi Trajectory Data

One of the interesting moving objects is the taxi. People use this reliable mode of transportation to visit places and discover new attractions. However, taxi drivers suffer from high traffic as they drive for long distances. Most of the times they need a guide for their trips more than relying on their driving experience. To solve this issue, taxis can be equipped with GPS devices which they use for trips navigation. The GPS device can record the sequence (i.e. trajectory) of geographical coordinates for the moving vehicle. A trajectory sequence is expressed by several points, each point is represented by an instance of longitude and latitude (Z. Feng, 2016). Analyzing raw taxi trajectory data can uncover

attractive and busy areas, but with semantic entities where each place is described by visitors, it can lead to more insights into the context of these places. Furthermore, if we know the road network for this data, we can detect the travel direction and the flow of moving taxis in the city. In this study, we use raw taxi trajectory data to identify attractive places in Porto such as shopping malls, leisure places and living areas based on taxi movement patterns.

1.4 Trajectory Mining Techniques

One major technique of mining trajectory data is the data visualization. For example, traffic data has become a major part of human life as people spend a decent time on roads every day (Wei Chen, 2015). Visualization of traffic data can provide an easy interpretation that incorporates human capabilities. Taxi trajectories graphs can detect congestion areas and traffic jams. Taxi flow patterns can be identified which can lead to predicting the traffic situation in each place at a given time. Multiple visualization approaches were proposed such as time visualization, spatial visualization, and spatiotemporal visualization. In our paper, we use time series visualization for taxi trips as they travel from origin to destination. Furthermore, we use spatiotemporal visualization heatmaps to describe the traffic status in each area for a given period.

To extract taxi movement patterns, we use Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) clustering approach (Ricardo J. G. B. Campello, 2013) to group similar points in each timeframe during the taxi trip. Each trip has a start point which is the origin, and a group of points representing the track where the taxi is moving, and an end which is the destination. Clusters of each timeframe are visualized on the city's map, the travel distribution and trips flowing among the city's areas are

recognized by detecting the density of each cluster. HDBSCAN clustering approach is based on DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (M Ester, 1996). DBSCAN is an efficient algorithm for finding arbitrary shapes with noise and outliers. It is based on the Euclidean distance where close points are grouped in one cluster and fewer density points are considered as noise. In addition, other approaches can be used in trajectory data mining such as classification. In this approach, a set of features can be initialized such as trajectory speed, duration, and length. Then this list of features is used to train a classification model and predict the class attribute.

Additionally, we apply the naïve Bayes classification to predict the traffic status. Trips which start at the same timeframe are grouped together. We build a classification model which relies on a set of features such as trip day, trip time, and area to predict the class feature which is the traffic status. Naïve Bayes classifier assumes that features used to predict a class attributes are independent and not highly correlated (Han Jiawei, 2011). This classifier can predict the class attribute conditional probabilities with other attributes, the classification model first learns the target function of mapping attributes by using a training set, then a test dataset is used by the model to predict unseen data. Naïve Bayes classifier can perform well with some types of attributes such as discrete and Boolean. For each instance, it uses the prior probability to calculate the posterior probabilities, after that the class value with the higher probability is assigned to the class attribute. Naïve Bayes classifier is easy to build and implement, however, some factors could affect the accuracy of the model such as dependencies among attributes, improper discretization of continuous attributes and imbalanced class values.

Another technique that we apply is the SPADE sequential pattern mining algorithm (Zaki, 2001). The Apriori-based technique can detect sequential patterns using lattice trees in three database scans. We group areas together and eliminate points in each trip where the area is duplicated. Eliminated points in our dataset represent the status where the taxi is still in the same area and is not moving to a different one, so we keep the distinct areas for each trip then we apply SPADE on the list of trips to extract the most repetitive taxi patterns among Porto city areas.

Some previous approaches were concerned with finding outlier trajectories that do not match the overall behavior of the trajectory data. Route and location prediction are important approaches, the route prediction aims to find a match between the current trip and a previous trip to adapt its route. Some approaches built probabilistic models such as Markov Model to predict the next road segment (Krumm, 2008). In our approach, we apply trajectory time series analysis. Each trip represents a sequence of time series longitudinal coordinates, we measure the similarity of these trips by using Euclidean Distance and Dynamic Time Warping (DTW) distance measures. Then we cluster trips based on their similarity using hierarchical clustering. Finally, we perform a visual analysis to detect outlier clusters which behave differently.

1.5 Trajectory Mining Challenges

Overall, trajectory data mining analysis can enhance our understanding of objects behavior, applying its approaches can uncover movement patterns and attraction areas. Additionally, extending this analysis can be used to build a useful recommendation system that carries out predictions for potential routes, places, and traffic conditions.

However, some challenges should be addressed in terms of trajectory data analysis (Lee Jae-Gil, 2015). One challenge is handling the massive amount of trajectory data generated from various types of sensors. Dealing with different formats of heterogeneous data and solving the missing data issue is another challenge. The scale and granularity level adapted in analyzing trajectory data is another factor that can impact the outcome of the analysis. The lack of collaboration between trajectory data analysts and domain experts in setting methods and parameters is a major issue that can cause irrelevant results. One main challenge that faces trajectory data analysis is the human privacy issue, tracking individuals are not openly available like animals, most studies require a signed consent form from participants to run such experiments (Mazimpaka Jean Damascène, 2016).

Another challenge is finding an interactive effective module which can collect, store, and analyze spatiotemporal data coming at a high speed. The massive amount of trajectory data requires a powerful framework with parallel processing features such as Apache Hadoop to adopt the scalability and reduce any latency issues. Another challenge is the trajectory data which requires some preprocessing techniques before Analysis (Z. Feng, 2016). One technique is noise filtering, collected trajectory data is not optimal since the signal of sensors can be affected, noise points need to be filtered and excluded before trajectory data is mined. Another issue is the delay in points detection, some points are not significant as moving object remain stationary for a while, these points need to be excluded.

Chapter 2: Literature Review

Various trajectory data mining techniques were proposed such as clustering, pattern mining, time series analysis, and classification.

2.1 Trajectory Clustering

(Palma Andrey, 2008) proposed a clustering approach to finding major places on a trajectory based on its speed. They extracted the knowledge from trajectory data by combining geographical characteristics with a semantic description for each trajectory stop (e.g. hotel, museum, nightclub). They applied DBSCAN algorithm to identify stops with minimal time instead of minimal density. However, some points in the trajectory were missing and some points were identified as unknown points as they lacked the proper semantic description. (Zhao Liangbin, 2017) adapted a hierarchical clustering approach based on DBSCAN to find similarity in ships trajectories. They were able to discover the ships traffic flow into the sea by identifying clusters with various densities which contributed to the maritime application area.

Meanwhile, some previous studies divided the trajectory datasets into equal-sized grids then applied DBSCAN clustering on each grid. (Shen Ying, 2015) applied DBSCAN clustering to detect hotspots for passenger's pick-up and drop-off locations. They enhanced the algorithm parameters sensitivity by applying clustering on each grid and initializing the density threshold for each grid. Also, they created a weighted tree with multiple factors such as speed, time and distance to recommend the most suitable routes. (Irrevaldy, 2017) partitioned trajectory dataset into small grids, then they applied spatiotemporal clustering as they clustered each grid for each time interval. They were able to find congestion areas with the high-density population in a specific time interval.

(Feng Mao, 2016) proposed a novel approach to detect spatiotemporal patterns for household travel. They first clustered trips origin and destination points (OD) to identify attractive places, then they visualized their clusters to analyze the behavior of the urban areas based on the spatial distribution and temporal trend. (You Dabin, 2017) proposed an urban mobility model based on DBSCAN clustering. They extracted clusters from the distribution of source locations during time intervals, also they measured the degree of mobility by analyzing the connectivity between destination and source clusters. (Yang Yue, 2009) explored attractive areas and movement patterns by clustering pick-up and drop-off points in five-time spans. They extracted the time-dependent level of attractiveness by studying the distribution of clusters and the flow interactions between different clusters.

Some previous studies proposed semantic trajectories by adding information such as text and photos to the trajectory points. (Takimoto Yoshiaki, 2017) analyzed semantic trajectories to extract frequent patterns and regions of interest (ROI). They proposed a new clustering algorithm SimDBSCAN which was an extension of DBSCAN algorithm. They clustered regions based on similarity and used Flickr photographs of trajectory points to describe user preferences. Some models combined trajectory points with geographical location information. However, these models were efficient only when trajectory points were matched with geographical regions of interest (ROI). ROI places are usually distributed and located along the trajectory track; therefore, it is hard to locate these places unless they are on intersections or on a trajectory turn point.

Some studies focused on finding outlier patterns instead or regular patterns in a trajectory data. (Lei Bao, 2018) proposed a distance-based approach to detect outliers. They applied DBSCAN clustering algorithm to reduce the size of data and to extract patterns, then they

produced gravity vectors for each trajectory. Finally, they used relative distance and angular distance equations to measure the similarity of clusters and trajectory points. Their approach can be used in anomaly detection and in identifying any unusual or suspicious movement behavior.

2.2 Time Series Analysis, Forecasting, and Anomaly Detection

Time series analysis is used to study the behavior of trajectories over time. In this analysis, trajectory data is visualized so the human can recognize shapes of trajectories and identify their similarity. However, when dealing with massive datasets with high dimensions, time series analysis needs to apply some techniques like indexing to enable fast query of the data. Another solution is to reduce the dimensionality of data by applying techniques like Discrete Fourier Transform (DFT) and segmentation. To divide time series trajectories into distinct groups, a clustering technique is applied. Each cluster will contain a set of time series trajectories which are similar (Esling Philippe, 2012).

(Khashei Mehdi, 2011) applied time series forecasting technique. They integrated Artificial Neural Networks (ANN) with (ARIMA) models to enhance the accuracy of the forecasting. This hybrid model was proposed to overcome the limitation of existing linear models. Their experiments showed that the ARIMA-ANN model was more accurate than other hybrid models. (Chujai Pasapitch, 2013) used Autoregressive Integrated Moving Average (ARIMA) to forecast the household electricity consumption. They applied time series decomposition to extract seasonal trend for the consumption, then they predicted the monthly and quarterly consumption. For weekly and daily consumption prediction, they applied the Autoregressive Moving Average (ARMA) model which performed better for this forecasting period.

Anomaly detection technique was used to find rear events in time series data (Esling Philippe, 2012). One method to perform anomaly detection was to discover subsequences (i.e. motifs) which repeat in longer time series. (Kumar Vipin, 2014) detected global changes in climate such as extreme rainfalls and droughts. They were able to identify one anomaly which was an extreme change in the vegetation level in years 2003 and 2006. However, the authors mentioned that it was hard to decide if a significant change in time series was a shift or just a fluctuation in the data. (Wang Xiaoyue, 2013) conducted an experimental comparison among similarity measures for time series data. In terms of time series classification, they concluded that Dynamic Time Warping (DTW) could be more accurate with small training sets, while Euclidean Distance (ED) was more accurate when using large training sets. They recommend enlarging the training data to minimize the error rate for both DTW and ED measures.

(Aghabozorgi Saeed, 2015) described time series clustering as a useful technique to aggregate and visualize data in a clear way. They mentioned that clustering could help to explore and extract frequent patterns and anomalies in time series data. They listed application areas that applied time series clustering such as Biology, Climate, Finance, and Medicine. Additionally, the authors discussed multiple similarity measures used in clustering such as Dynamic Time Warping (DTW), Euclidean Distance (ED), and Pearson's correlation coefficient. In our study, we use time series analysis to visualize a set of taxi trips trajectories, our time series trajectories have two dimensions, time and geographical location. Additionally, we perform hierarchical clustering on taxi trips to group similar trips and study their trend over time. Our clustering is based on two distance measures, Euclidean Distance (ED) and Dynamic Time Warping (DTW).

2.3 Sequential Pattern Mining

Sequential pattern mining is a technique that is used to discover repeated subsequences in a set of sequences. Time series data is an ordered form of sequences where data is represented over time. Trajectory data can be considered as a time series sequence which consists of a set of ordered longitudinal locations. (P. Fournier-Viger, 2017) mentioned that a discretization process was required to perform sequential pattern mining. for example, when dealing with financial time series data, numerical amounts of money should be associated with symbols based on the defined intervals. (Bermingham Luke, 2014) performed spatiotemporal trajectory region-of-interest (ROI) sequential pattern mining. they used Flickr photos for Queensland taken by tourists to represent each ROI point in a trajectory. They applied the SPM framework to mine the ROI dataset and they were able to uncover interesting patterns in the east coast and Brisbane.

(Dongzhi Zhang, 2015) discussed details of spatiotemporal trajectory periodic pattern mining (PPM). they categorized PPM methods into three types, one-dimensional sequence, two-dimensional time series or spatial data, and spatiotemporal trajectory data. The one-dimensional type patterns could be detected using association rule mining algorithms such as SPADE (Zaki, 2001) and WPPM (Chanda Ashis Kumar, 2017). For time series data, previous mentioned time series analysis techniques could be used to detect the change of data over time. For spatial data, spatial distributions could be compared over different areas to find patterns, the time element was ignored in this PPM type. Finally, for spatiotemporal trajectories, each moving object was represented by a line over time, and shapes of lines were analyzed to detect periodic patterns. In our study, we use a one-dimensional type to represent each taxi trip trajectory as a sequence of locations. After that, we eliminate

repeated locations in each trajectory (i.e. sequence) and perform SPADE algorithm over a set of trajectories (i.e. taxi trips) to extract sequential patterns of visited locations. In addition, we perform a time series analysis to visualize and detect any periodic patterns among the set of taxi trips trajectories.

2.4 Trajectory Classification

Some studies applied different trajectory data analysis approaches such as classification. (Tang Luliang, 2015) proposed a novel approach to determine the number of lanes and turn rules on a traffic road. They created a naïve Bayesian classifier based on the features of the road and number of lanes, then they used a test sample on the trained classifier to extract the number of lanes and turn rules on the road. They made experiments on simple roads and intersections, not on complex roads such as tunnels and bridges.

Some previous studies applied trajectory data mining techniques such as clustering and classification to build recommendation systems. For example, on taxi trips, services offered to passengers is not efficient as taxi drivers can experience traffic jams and congestions. This could affect both driver and passenger, the taxi driver will not be able to drop off the passenger on time. Additionally, his car will consume more gas while waiting in the high traffic areas. Moreover, if a passenger is looking for a vacant taxi, it would be difficult to get one in the low traffic areas where they can find a few numbers of taxis (Yuanhang Hu, 2015). Authors proposed several techniques to build an effective recommendation system that could help both drivers and passengers such as clustering, classification, and regression prediction using naïve Bayes and Markov model. Their approach was based on taxi driver personal preference to predict route or location. However, in our study, we perform both clustering and classification to predict traffic status in origin and destination. Predicting the

origin traffic status will help taxi drivers to identify the hot spots in the city, they can detect places where they can likely pick up passengers. While predicting trips destination traffic status can help passengers to identify hot areas where trips end, thus, where they can find more vacant taxis to ride.

(Hwang Ren-Hung, 2015) proposed a taxi recommender system to predict the next cruising location. They used the location-location model to detect the relation between current passenger drop-off location and the next passenger pick-up location based on multiple factors such as distance, waiting time, fare and driver experience. This recommendation system could increase the revenue of a taxi driver by analyzing historical taxi trips and providing an optimal potential location. (Ye Ding, 2013) presented a system called HUNTS. This system was based on historical GPS data and business data, each road was evaluated in multiple timeframes based on location and profit factors. Then a score was assigned to each road and time interval, roads with a higher score were busy roads with congestion. (Chen Ling, 2010) proposed an approach to predict route and destination of person trajectory data. Their prototype used historical data of the person moves generated from his smartphone. Then they applied a clustering algorithm to identify the important places the person visits, then they extracted the person's movement patterns. Finally, they predicted destination and future route based on the movement patterns and historical data without relying on road network or semantic information. (Zhang Shu-kai, 2018) proposed a novel approach to predict the route of ships trajectories in China. They preprocessed the trajectory dataset, then they clustered trajectory points using DBSCAN algorithm. After that, they extracted spatial patterns and turning nodes. Finally, they detected the connectivity between turning nodes to find the optimal route to the destination.

(Gang Pan, 2013) also used taxi trips datasets in China to predict the urban land-use. They used DBSCAN clustering to identify regions with high pick-up density, they were able to identify regions such as train station, hospital, commercial, entertainment, and residential. After that, they used multiple classifiers such as Support Vector Machine (SVM) and K-nearest neighbor (KNN) to predict the region type for a given taxi trip. Their classifiers could reach an accuracy of 95% for land-use prediction. In our study, we use Hierarchical Density-based spatial clustering of applications with noise (HDBSCAN) algorithm (Campello Ricardo J. G. B, 2013) to cluster the pick-up and drop-off points, also we cluster each timeframe point during the trip to identify traffic status as the taxi travels through the city areas. Other studies relied on taxi speed to predict transportation mode (Sun Zhanbo, 2013). They used data collected from GPS and applied SVM classifier to predict the vehicle type, passenger vehicle or truck. Their model accuracy varied and was related to the technology used such as inductive loop detectors, radar sensors, infrared sensors, and acoustic sensors.

(Nick Theresa, 2010) classified the means of transportation using mobile phone sensor data. They applied the naïve Bayes classifier to predict the mean of transportation like a car, train, or pedestrian, the accuracy of their classifier was 93%. Multiple classifiers were applied to predict mode features, (Gonzalez P.A, 2010) used neural networks (NN) and GPS mobile phones to predict the mode of transportation in Florida such as a car, bus, and walk. The highest accuracy accomplished for their model using 10-fold cross-validation was 91.23%. (Rodriguez-Galiano V.F, 2012) measured the effectiveness of the random forest classifier in predicting land cover in Granada provinces such as water, urban, greenhouse, grasslands, and bare soils. With the proper configuration of random forest

parameters such as the number of trees and random split variables, the error rate for their model was less than 10%.

2.5 Trajectory Data Visualization

The exploration of spatiotemporal trajectories faces multiple challenges such as the large scale of data and storage limitations. To extract the knowledge from this data, efficient systems should be provided to store, analyze, and visualize trajectory datasets. (Ferreira Nivan, 2013) used taxi trips dataset to provide more insights into the activity index and human patterns of New York City. They proposed a model that executed a wide range of spatiotemporal queries based on the origin-destination factor. The user could select a sample of the dataset and visualize the spatial distribution of taxi trips in different areas on the map of New York.

(H. Xiong, 2017) designed a visualization system which described passenger status (e.g. occupied, vacant) and the speed of the vehicle in a taxi trajectory dataset. They used MongoDB database to store the taxi trajectory dataset, then they applied DBSCAN algorithm to cluster the passenger pickup points and visualize them. Additionally, they designed an interactive visualization of the movement of taxis on a sample of this dataset.

(Jianqin Zhang, 2015) proposed a method for spatiotemporal visualization of Beijing taxi trajectory dataset. They considered multiple factors in their analysis such as driver residence location, daily operation time and driver rest periods. Their analysis indicated that taxi traffic was 20% of the city traffic and the rate of vacant taxis was higher than busy taxis. Relying on these factors and analysis, major information could be provided to the city's taxi operations management.

Our methodology is different from previous studies in many ways. In terms of clustering, we analyze, visualize and cluster a large amount of data. We extract 24 CSV files where each file represents a specific hour of the day. Furthermore, we extract 168 CSV files representing 24 hours of the day and 7 days of the week (i.e. weekday vs day hour). Each CSV file contains a set of trips, each trip has a starting point, an end and intermediate moving points for different timeframes. In addition, we cluster all trips and their moving points, our clustering granularity is one hour. It is more detailed than previous studies with a lag of 15 seconds between successive locations. We apply HDBSCAN which is a modified version of DBSCAN algorithm. However, HDBSCAN does not require Epsilon parameter which reduces the parameters sensitivity for our study.

In terms of classification, we use the naïve Bayes classifier and collected GPS data to predict traffic status in origin, destination, and any given point during taxi trips in Porto city. To extract the traffic status feature, we apply clustering on each timeframe point (i.e. origin, destination, and in-between points), then we rely on the degree of membership for each point to determine its traffic status. Moreover, we manipulate our threshold to get the most balanced dataset to avoid the low accuracy of our classifier as discussed in (Mujalli Randa Oqab, 2016) study. They described the misclassification issue where the imbalanced dataset can cause the model to frequently predicts the majority class. In their study, they evaluated different types of classifiers such as naïve Bayes and decision trees. Their experiments proved that balancing the dataset will improve the accuracy of the naïve Bayes classification. Additionally, since many real-world datasets are imbalanced, they proposed to apply re-sampling techniques such as oversampling and under-sampling.

The main contribution of this paper is to establish a data driven approach to optimize the traffic operations in Porto city by building a recommendation system based on trajectory analysis.

2.6 Comparative Analysis

Previous studies worked on different types of trajectory analysis. They applied various data mining techniques and algorithms such as clustering and classification based on the dataset and knowledge they targeted to extract. Table 1 classifies the most relevant trajectory analysis algorithms applied on taxi datasets. Features and findings of these algorithms are discussed in this table.

Table 1. Previous trajectory analysis approaches

Dataset	Algorithms	Findings
Porto taxi trips (Moreira-Matias Luis, 2013)	➢ Time series forecasting (ARIMA)	➢ predicted passenger demand ➢ had a low error rate prediction with 26%
Jordan traffic accidents (Mujalli Randa Oqab, 2016)	➢ Naïve Bayes classification	➢ improved the ability of classifier by applying sampling to balance the dataset
Hangzhou taxi trips, China (Gang Pan, 2013)	➢ DBSCAN ➢ SVM, KNN, LDA, BP	➢ Extracted regions in province ➢ Predicted the land-use with an accuracy of 95%
Beijing taxi trips (Jianqin Zhang, 2015)	➢ Visualization	➢ High rate of empty driving ➢ Provided decision-making guidelines for city managers

Nanjing taxi trips, China (Shen Ying, 2015)	➢ DBSCAN ➢ Weighted tree	➢ Found optimal route for vacant taxi drivers to pick up passenger
Large city taxi trips in China (Ye Ding, 2013)	➢ Dynamic scoring system	➢ Discovered patterns of red and green taxis ➢ Discovered the variation in driving patterns before and after picking up a passenger
Shanghai taxi trips, China (Feng Mao, 2016)	➢ Grid-based clustering method	➢ Discovered spatiotemporal clusters patterns of household travel
Wuhan taxi trips, China (Yang Yue, 2009)	➢ K-Nearest neighbor	➢ Detected attractive areas ➢ Detected travel patterns in urban areas which helped transport management
Seoul Metropolitan Area taxi trips, Korea (You Dabin, 2017)	➢ DBSCAN ➢ GMM	➢ Identified urban mobility models ➢ Visualized degree of mobility

Chapter 3: Methodology

3.1 Data Description

The dataset used in our study is a taxi service trajectory dataset (Moreira-Matias Luis, 2013). This dataset is provided by the UCI Machine Learning Repository (Dheeru Dua, 2017). It was originally acquired by telematics installed on each one of the 441 taxis. The dataset has information about all taxi trips in Porto, Portugal from June 2013 to June 2014. The raw CSV file was extracted with a total size of 2 GB and contained approximately 1.7 million taxi trips. Each taxi trip in this dataset is a sequence of timestamped longitudinal coordinates, it has an origin and a destination. However, a trip could be vacant or occupied since we did not have an attribute to distinguish the two states. Each trip instance consisted of 9 attributes, Trip id is the unique identifier where we can distinguish between different trips. Each trip has a call type, this attribute determined the area from which the taxi was requested (i.e. center area, taxi station or a random street). Origin call attribute indicated the phone number used to request the taxi, and origin stand attribute indicated the start point of the trip if the start was from a taxi station.

Taxi id is an identifier for the driver who made this trip, timestamp attribute is the time in Unix format which identified the start time of the trip. The day-type attribute is used to indicate the nature of the day (i.e. holiday, weekday or weekend). To describe the accuracy of the GPS device reading, missing data attribute was added to this dataset. If the location was not captured correctly, this attribute indicated that the reading was missing by a “true” value. Otherwise, it indicated that the reading was captured, and the value was “false”. Polyline is a major attribute that expressed the trajectory nature of the taxi trips. Each trip has a sequence of GPS coordinates, each pair of coordinates is described by longitude and

latitude values. For every 15 seconds of the trip, a new pair of coordinates was added to the sequence indicating the new location of the taxi. The length of the Polyline sequence attribute varied. Hence, short taxi trips had a few pairs of coordinates, while long trips had a more pairs of coordinates.

3.2 Data Preprocessing

First, we downloaded the dataset as a CSV file from the UCI website (Dheeru Dua, 2017), then we used DB Browser for SQLite (Free Software Foundation, Mozilla Foundation) to import the CSV file into an SQLite table called Taxi Data. After that we deleted all records with a missing data attribute value of "true", the number of missing records was approximately 162 records. For the Unix timestamp attribute, it was converted into a windows timestamp with the following format "yyyy-mm-dd hh:mm:ss".

By having the time and date format, it was applicable to extract the weekday and apply queries with date and time comparisons. After that, we partitioned the Polyline attribute to extract the first pair and last pair of coordinates, the first pair represented the trip start location and the last pair of coordinates represented the trip end location. To calculate the duration of each trip in seconds, we calculated the number of coordinate pairs for each trip, then we multiplied the count of pair coordinates with 15 seconds which is the time resolution of each moving point.

Then from the trip start time attribute, we extracted the weekday, month and year for each trip. The first trip started on the 30th of June 2013 at 20:00, and the last trip in this dataset started on the 30th of June 2014 at 19:59. Then, we exported the taxi data table with the new calculated attributes (i.e. trip start time, trip end time, trip duration, day, month, year, trip start location, trip end location) into a CSV file. Finally, we used the Delimit software

(Delimitware, 2018) to open this CSV file and to partition the Polyline string attribute into single columns. Each pair of coordinates inside this attribute had brackets on each side, and a comma was used to separate each pair. We eliminated the brackets and used a delimiter of type comma as all CSV files are comma separated files. We were able to split each string of coordinates into separate columns; accordingly, the final CSV file had the original attributes along with the calculated attributes. Additionally, as an output from the Delimit splitting process, new columns were added to represent the moving points of each trip, each moving point was divided into two sub-columns which were the longitude and the latitude coordinates.

After applying this approach, each trip had a different number of columns since the duration of trips varies. Hence, short trips will have fewer columns than long trips as they have fewer moving points. For example, if we had a trip with a duration of 5 minutes, then the duration of the trip in seconds is 300, considering that the moving points will change every 15 seconds (i.e. time resolution), therefore, we will have 20 moving points for this trip. These 20 moving points will be represented by 40 columns in the CSV file as each moving point will be divided into two columns, one for the longitude and one for the latitude. Furthermore, if we take a longer trip with 30 minutes duration, then its duration in seconds is 1800. In this case, we will have 120 moving points, which means 240 columns for this trip in the CSV file.

In general, these steps represented the higher level of preprocessing for taxi trips dataset. This level of preprocessing added useful attributes which were a need in our trajectory data mining techniques as we can observe from table 2. However, we will conduct a detailed level of dataset preprocessing such as data sampling, unsupervised discretization, and long

formatting when we apply trajectory data mining techniques such as clustering and classification.

Table 2. Part of the preprocessed dataset

Id	Missing	Polyline	Start time	Duration	Day	Month	Year
1	FALSE	-8.611155,41.149764	2014-05-02 5:13	15	Friday	MAY	2014
2	FALSE	-8.580141,41.159475	2014-05-02 5:23	15	Friday	MAY	2014
3	FALSE	-8.585604,41.148567	2014-05-02 5:37	15	Friday	MAY	2014
4	FALSE	-8.585658,41.148513	2014-05-02 5:37	15	Friday	MAY	2014
5	FALSE	-8.585631,41.148549	2014-05-02 5:44	15	Friday	MAY	2014
6	FALSE	-8.575083,41.151987	2014-05-02 5:50	15	Friday	MAY	2014
7	FALSE	-8.606502,41.144589	2014-05-09 5:14	15	Friday	MAY	2014
8	FALSE	-8.614755,41.146083	2014-05-09 5:26	15	Friday	MAY	2014
9	FALSE	-8.63037,41.158269	2014-05-09 5:22	15	Friday	MAY	2014
10	FALSE	-8.596431,41.15385	2014-05-16 5:00	15	Friday	MAY	2014

3.3 Descriptive Analysis

In our study, we perform two levels of trajectory data analysis. The high level of analysis where we perform aggregation SQL queries to extract information about counts of trips and average trips duration. The next level is the detailed analysis, in this level we perform multiple trajectory mining techniques such as DBSCAN clustering, naïve Bayes classification, sequential pattern mining, and time series analysis. The detailed analysis will provide more insights into the behavior of taxi trips and trips movement patterns in

different time spans. For the high-level analysis (i.e. descriptive), we analyzed two main factors in the taxi trips dataset, number of trips and average trip duration in minutes.

In our descriptive analysis we used two types of charts, the bar chart and the line chart. Both types are useful in showing comparisons and trends over time. We used bar chart for studying the trend of taxi trips over distinct attributes (e.g. the trend of taxi trips over call type attribute). Meanwhile, we used line charts to visualize the trend of taxi trips over various timestamps of the day (e.g. the trend of taxi trips over 24 hours). However, the two charts can be used for the same purpose, but we believe that using bar charts is more appropriate when we have a small number of labels.

The overall number of trips in our dataset was approximately 1.7 million. We analyzed the number of trips for multiple factors such as call type, weekdays, month and day. First, we analyzed the number of trips per call type. Figure 1 describes the number of trips in the call types, central, stand, and random street. The largest number of trips was for the trips which started from a stand dispatch with 817K trips. Trips which started from a random street came next with 527K trips while trips that started from the central call types came last with 364K trips. The low number of trips for central call type is because people preferred to avoid requesting taxi by calling central since they had to pay for extra 0.8 € (Guedes, n.d.).

We next analyzed the number of trips per weekdays. We can notice from figure 2 that the smallest number of trips was on Sunday with 178K trips, that is probably because people want to spend some time at home before going back to work the next day. Meanwhile, the largest number of trips was on Friday and Saturday with 306K and 266K trips respectively. It might be busy in these two days because most people start to hang out and spend more time in restaurants, bars, parks and other attraction areas since the next day is a weekend.

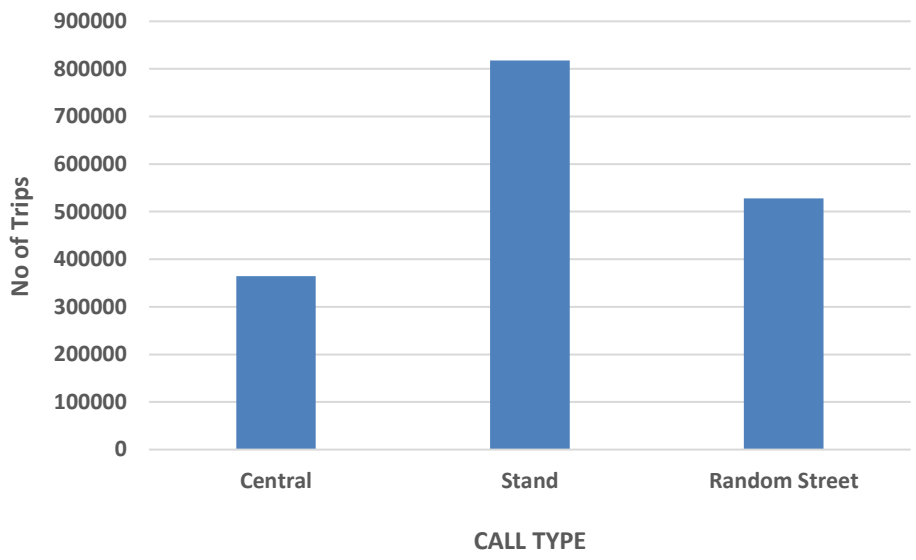

Figure 1. Number of trips per call type

Additionally, people who live outside the city start to leave their work early in the last day and move to the train station or the airport to travel to their families. For the weekdays, the number of trips was approximately the same.

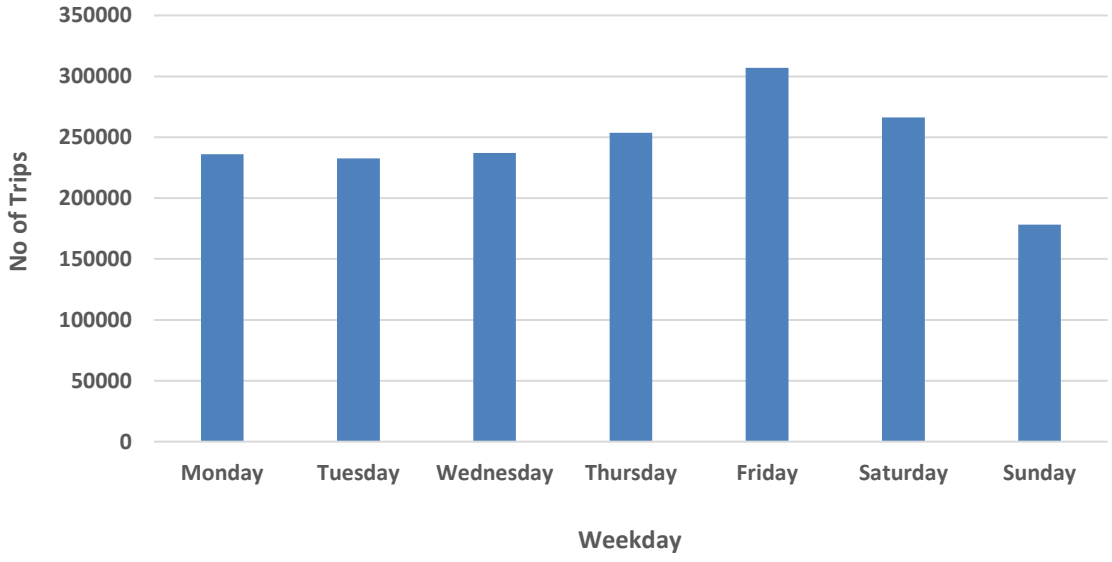

Figure 2. Number of trips per weekdays

We next analyzed the number of trips per month, figure 3 shows the distribution of taxi trips over 12 months. We can observe that the highest number of trips was in May with 162K trips followed by October with 153K trips. One reason for the high number of trips in the city is that Porto was selected as the top European destination in 2014 by approximately 228K voters in an online competition on Europe's Best Destinations (European Best Destinations, 2018).

Furthermore, there are multiple reasons behind the high number of trips in May and October. Most rainfalls in Porto do not come in these two months, the rainfalls season in Porto starts from November until March of every year. The prices of hotels and accommodation play the main role in the taxi movements, the peak season where the prices are high is not in May and October, but in the summer months of July and August. In addition, in May and October, the weather is not hot, it is cool and nice, tourists and visitors can spend more time hiking and cycling during these two months.

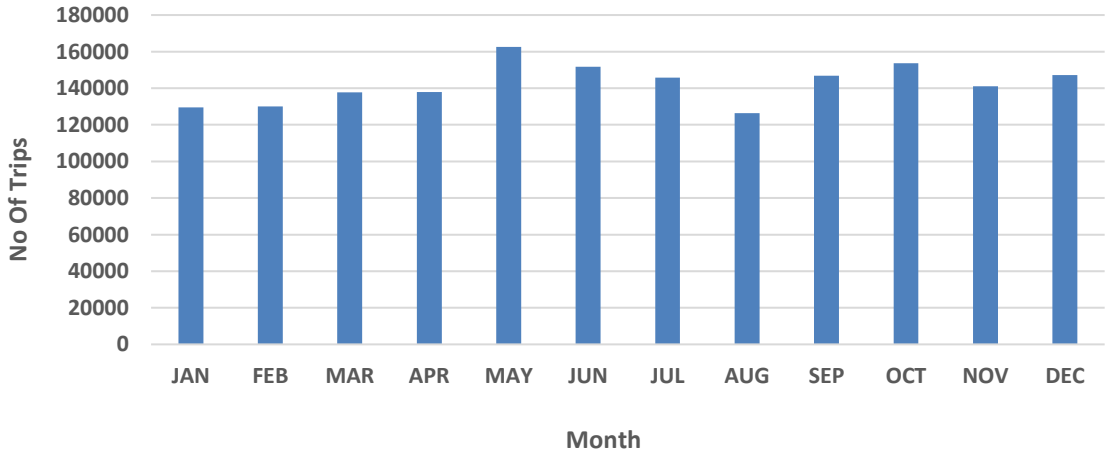

Figure 3. Number of trips per month

Our next analysis was the number of trips per day. In this analysis we studied the number of trips during different time slots of the day, we compared the taxi movements in the

daytime and at night. Figure 4 describes the change of the number of trips across daytimes, we can observe that number of trips for the hours after the midnight (1:00 AM – 2:00 AM) was low with 53K trips. After that, the number of trips significantly increased until it reached the peak in the early morning hours (4:00 AM – 5:00 AM) with 97K trips. Then it slightly decreased until the number of trips became 82K at 8:00 AM. Then it gradually increased until 11:00 AM with 91K trips, the number of trips gradually decreased for the next 11 hours until it became at the lowest level with 47K trips at 10:00 PM. Finally, the trips slightly increased until it is midnight with 56K trips.

The flow of the line chart provided a good indication that the busiest period of the day started early at 4:00 AM and ended at 8:00 AM. This is reasonable because people start to go to their work in the early morning hours. The slowest period started in the afternoon until night (11:00 AM–10:00 PM), which is the time people were not active. The last period is the midnight period (10:00 PM–Midnight), people during this time were probably more active on the weekends as they enjoyed their time and visited more places.

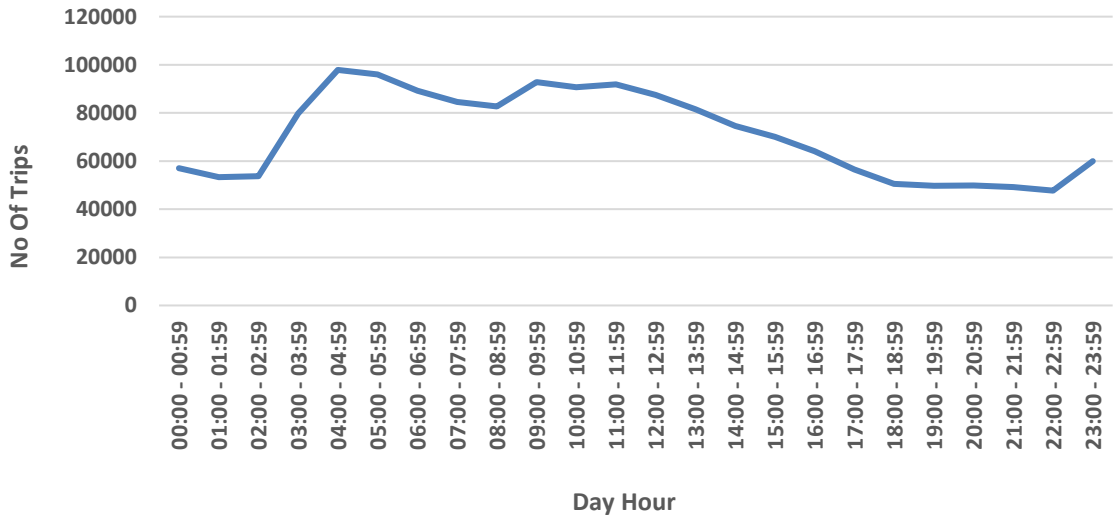

Figure 4. Number of trips in 24 hours

We next analyzed the average trip's duration in minutes. Figure 5 describes the average duration of trips per call type, we can notice from the figure that trips took longer when they started from a random street with 13.1 minutes. The reason behind this might be that drivers took more time to search for the optimal path while they pick up their passengers in a random area. Trips took the shortest time when they started from stand dispatch with 11.3 minutes. Moreover, trips took more time to reach their destination while dispatched from the central. This is because the time a taxi required to reach the place of the passenger was added to the trip duration.

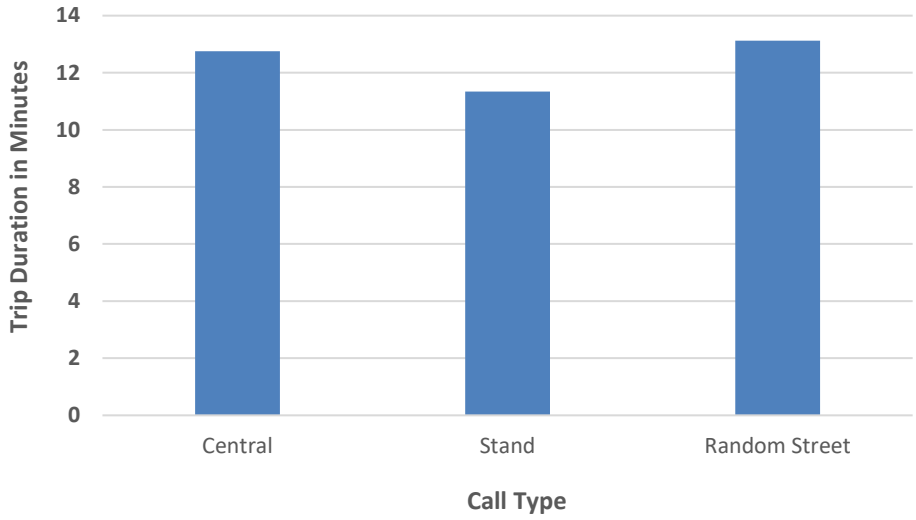

Figure 5. Average duration of trips per call type

We next analyzed the average duration of trips per weekdays. Figure 6 describes the trip's duration, we can observe from the figure that trips took a shorter time on weekends while they took a long time on weekdays. This is reasonable since traffic during weekdays is much higher than traffic at the weekends. The days where trips took the longest time where Friday and Monday with 12.5 minutes and 12.6 minutes respectively. We might relate this

delay to the high traffic status in these two days, on Friday people are more active since they are planning to visit more places and spend more time outside, thus, they are causing a higher traffic. While Monday is the start of the week and the traffic seems to be higher on this day compared to other weekdays.

We next analyzed the average duration of trips per month. As we can observe from figure 7, the trips were taking a longer time in October with an average of 12.72 minutes. November and May months followed with 12.37 minutes and 12.35 minutes respectively, while trips were taking the shortest time in August with 11.53 minutes. These results are relative to the number of trips in figure 3 where we had more trips in May and October. Therefore, the more trips we have on the road, the highest traffic is likely to happen. Thus, the more time trips will take to reach their destination.

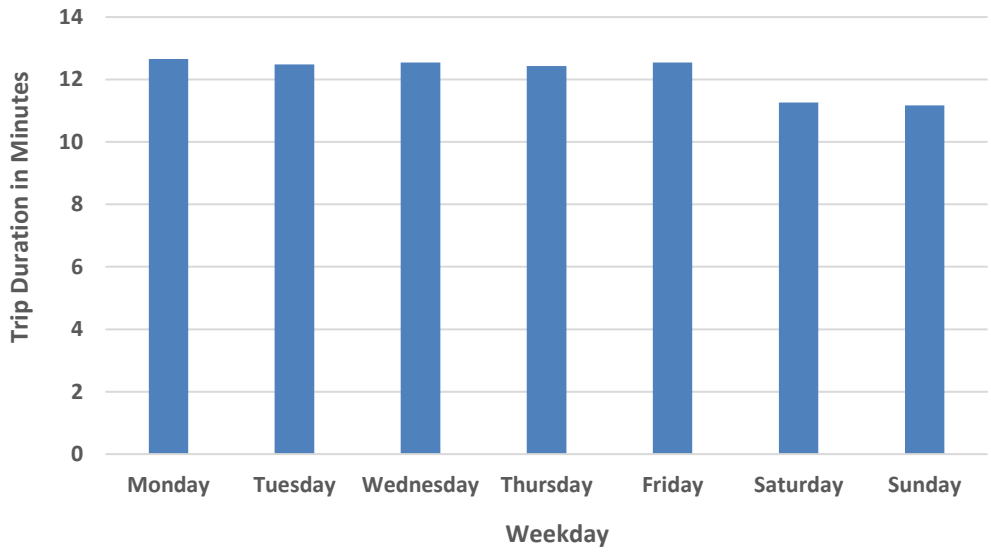

Figure 6. Average duration of trips per weekdays

Finally, we analyzed the average duration of trips in terms of day hours. Figure 8 describes the flow of taxi duration along 24 hours of the day. We can notice that trips were not taking

much time to reach their destination in the couple of hours following the midnight (1:00 AM – 2:00 AM) with approximately 11 minutes. Then the trip's duration slightly increased to reach 13 minutes at 4:00 AM, after that the trip's duration remained steady until 8:00 AM where the trip's duration started to increase until it reached the peak duration at 02:00 PM with 14.25 minutes. Finally, the trip's duration started to decrease until the end of the day (Midnight).

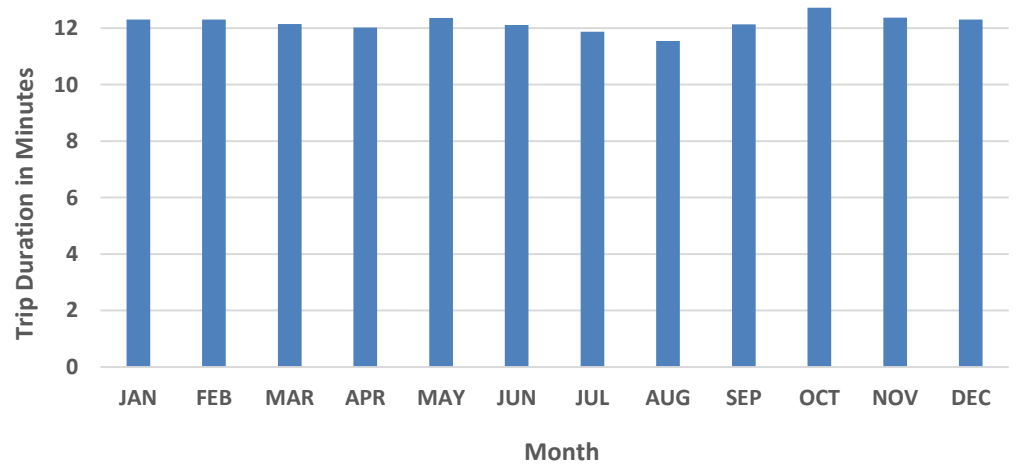

Figure 7. Average duration of trips per month

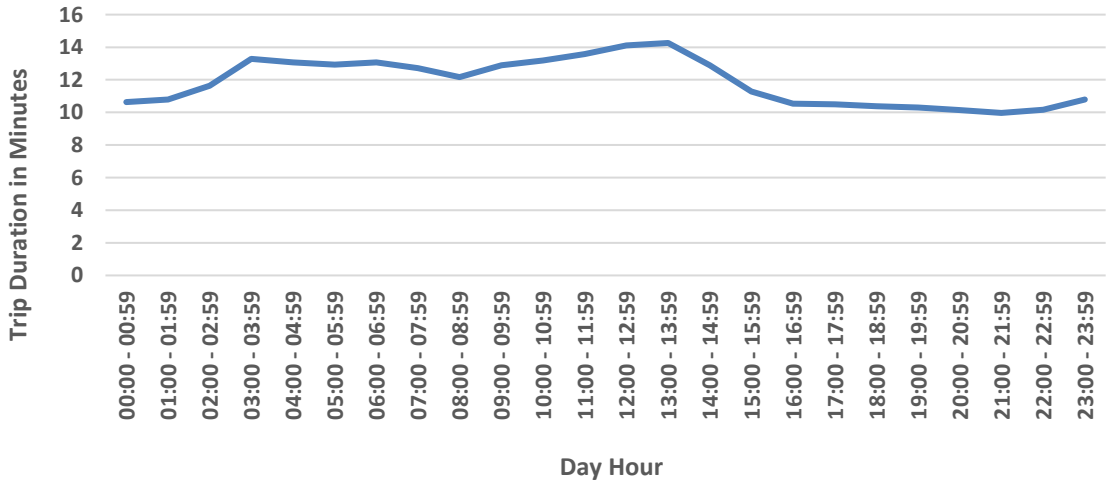

Figure 8. Average duration of trips in 24 hours

The line chart flow in figure 8 provided an indication that trips were taking more time to reach their destination in the early morning hours and afternoon hours (3:00 AM–2:00 PM). This might be related to the high traffic in the early morning which means taxi drivers were taking more time to end their trips. Meanwhile, For the evening and night hours, roads were less busy, so the trips were taking less time to reach their destination.

3.4 HDBSCAN Spatiotemporal Clustering

A. Data Sampling

The next step is to perform the detailed analysis on the taxi trajectory dataset. In this analysis, we applied clustering on taxi trips moving points. However, the number of trips in this dataset was approximately 1.7 million trips, figure 9 shows the distribution of taxi trips based on their start points. We used Tableau Desktop (Tableau Software Inc., 2018) to plot each trip's coordinates on Porto city map. As we can observe from this figure, it was hard to cluster this data, applying clustering on the whole dataset will suffer from space and memory limitations. Therefore, we need to perform our analysis on samples of this dataset, we relied on our descriptive analysis and we extracted a sample from the original dataset. This sample represented taxi trips which occurred in May. Figure 3 shows that the highest number of trips in Porto city was in May, this month is the optimal time to visit Porto because of the reasons we discussed earlier. By considering this time of the year, we will be able to study the behavior of both tourists and residents. The number of taxi trips in May was approximately 161K trips.

To have a deep understanding of the taxi trips behavior in this month, we clustered the taxi moving points based on two main factors, 24 hours of the day and weekdays. For the first factor, we divided the original CSV file for taxi trips in May into 24 smaller CSV files

based on the day hours (00 – 23). Each one of these files contained all trips which started at that hour of the day.

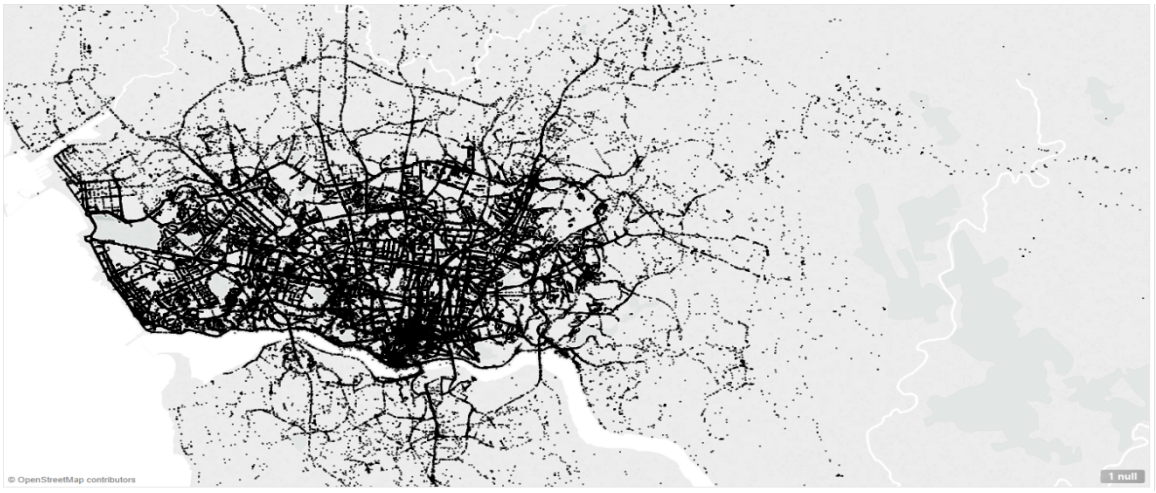

Figure 9. Spatial distribution of taxi trips based on start locations

For example, if we have a CSV file with the name 00, then we can say that this file includes all taxi trips which started at 00:00:00 to 00:59:59 (i.e. the midnight hour). And if we have a CSV file with the name 14, then we say that this file includes taxi trips which started at 14:00:00 to 14:59:59. Furthermore, we extracted only taxi trips which had a maximum duration of 1 hour. We excluded trips that took more than one hour to reach their destination, we did not want to include trips that overlap on two hours timeframes. In addition, the number of these long trips was approximately 1K trips, excluding these trips would not impact the behavior analysis for the dataset.

For the second factor, we divided the original CSV file for taxi trips in May into 7 smaller CSV files based on the weekday (e.g. Saturday, Sunday). Additionally, we further divided each of these 7 files into 24 small CSV files based on the day hours (00-23). At the end, we had 7X24 files which were 168 files, each file contained all trips which started in each weekday and hour. For example, if we have a CSV file with the name FRI-22, then we say

that this file includes all taxi trips which started on Friday and at 22:00:00 to 20:59:59. Also, we excluded trips that take more than one hour as we did for the first factor.

At the end of this sampling process, we created two folders with CSV files ready to cluster. The first folder contained taxi trips in May grouped by the 24-day hours, they were 24 CSV files labeled from 00 to 23. The second folder was taxi trips in May grouped into weekdays and day hours, this folder contained 168 CSV files ready for clustering. These files were labeled with a combination of weekday and day hour.

B. HDBSCAN Algorithm

After preparing the two folders which have CSV files, one folder with 24 CSV files and the other folder with 168 CSV files, we needed to cluster each one of these CSV files separately. Each CSV file contained a set of trips, each trip had several moving points starting from the origin point and ending with the destination point. To study the behavior of this dataset, we needed to cluster it as a spatiotemporal trajectory dataset and not as a spatial dataset. Previous studies focused on clustering taxi trips based on their origin or destination (i.e. spatial clustering). However, we intend to cluster every moving point for each trip to extract patterns from different timeframes. To perform a spatiotemporal clustering, we took all trips moving points for the same timeframe and we grouped them as a separate dataset, then we clustered this dataset. We kept clustering all moving points until we had no points to cluster as shown in figure 10.

For our clustering process, we used open source software RStudio (RStudio Inc., 2018) to write R scripts for reading the CSV file and clustering the trips moving points. RStudio is an integrated development environment that is a top layer which is built on the top of the

R console (R Foundation for Statistical Computing, 2018), it provides multiple windows where the user can simultaneously browse help information and R packages list.

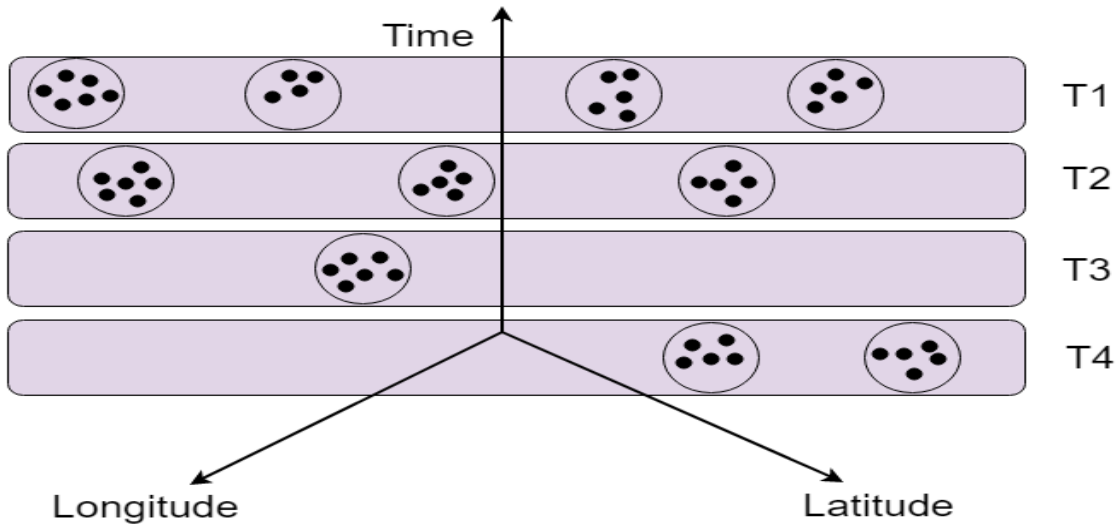

Figure 10. Spatiotemporal clustering of taxi trips dataset

We started with applying K-Means (Hartigan J. A, 1979) portioning clustering algorithm to the CSV files. (Hari Krishna Kanagala, 2016) experiments indicated that K-means was fast compared with other algorithms like DBSCAN and OPTICS. However, it cannot detect arbitrary shapes and outliers in data. Based on their findings and Porto city taxi trips visualization of K-Means clusters in figure 11, we assumed that this clustering algorithm was not appropriate for our analysis because of two reasons. One reason was that the number of clusters parameter *k* needed to be initialized prior to the clustering process, this was not applicable since we did not have enough knowledge regarding the attraction places of Porto city. The second reason was that K-Means assigns all trips to clusters, it will not identify outliers. this was not realistic because we had some moving points (i.e. trips) that were in far places of the city, these points should not be assigned to any cluster as they were noise. Accordingly, we decided to apply DBSCAN density-based clustering

algorithm for applications with noise (M Ester, 1996), it is more efficient in recognizing arbitrary shapes in the city's neighborhoods and identifying noise trips. This algorithm required two parameters to be initialized before clustering, the maximum distance between neighbors *Epsilon* and minimum points to form a cluster *MinPts*. The clustering result was sensitive to these two parameters, if we changed the value of *Epsilon*, this would impact the number and size of clusters produced.

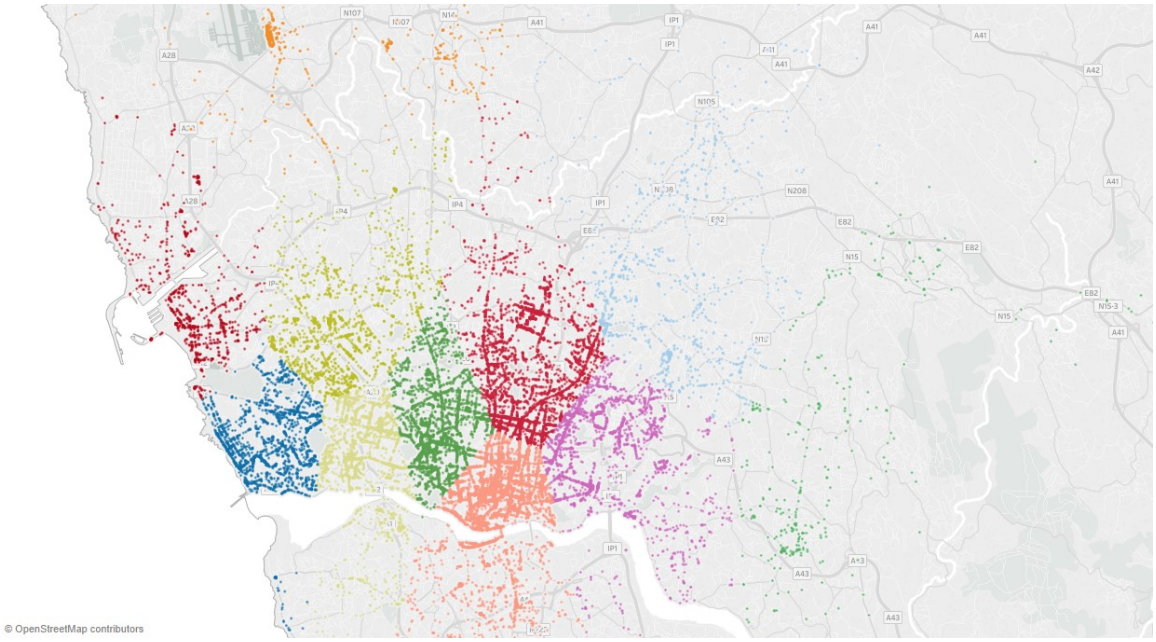

Figure 11. K-means clustering of Porto taxi trips

To assign the optimal value to DBSCAN *Epsilon* parameter, multiple methods were proposed such as k-nearest neighbor (KNN). The approach is to define a value of *k*, which usually corresponds to *MinPts* value, then it calculates the average distance of each point to its *k* nearest neighbors. After that, all calculated distances are plotted in ascending order, then a knee point (i.e. elbow) where there is a sharp change in the curve is assumed to be *Epsilon* value. However, the KNN method was not practical for our analysis because of two reasons.

First, when we applied KNNdistplot function which plots the graph of distances to find a knee point (i.e. *Epsilon* value), we had graphs with multiple knee points, in this case, it was impossible to determine the value for the *Epsilon* parameter. Second, we needed to cluster two folders with a total of 192 CSV files. Furthermore, in each CSV file, we needed to run the DBSCAN algorithm once for every group of moving objects. We had up to 240 clustering process in some CSV files, therefore, initializing the right value for the *Epsilon* parameter prior to each clustering was impractical and a time-consuming process.

Finally, we adopted a hierarchical clustering method HDBSCAN (Campello Ricardo J. G. B, 2013). This algorithm enhances DBSCAN by constructing a tree of significant clusters and using a single input parameter *MinPts*. This algorithm computes the core distance and a symmetric reachability distance among nodes, then it builds the dendrogram of significant clusters. This algorithm was suitable for our analysis since it was a density-based approach and it required only one input parameter which was *MinPts*. Initializing one parameter instead of two was more efficient as it saved time and increased the consistency of our clustering process.

To apply HDBSCAN, we wrote an R script which reads each CSV and performs clustering. We used DBSCAN package, we imported the CSV file, then we created a loop to go through each group of trips moving points (i.e. trips pair of coordinates). In each iteration of the loop, we clustered each group using HDBSCAN function, then we stored the clustering result as a new attribute in the same CSV file. for the null values, we skipped them, and we did not count them in the clustering.

HDBSCAN CSV Clustering
Input: CSV files with taxi trips grouped based on 24-day hours and weekdays-day hours
Output: clustered CSV files with extra attributes, each attribute is a clustering result for a pair of coordinates

Algorithm: ***HDBSCAN (CSV files)***
1. Read the CSV file
2. Iterate ***i*** from 1 to ***nPairCoordinates***
3. Extract ***longitude*** and ***latitude*** for pair ***i***
4. ***Count*** = number of nonempty records of ***longitude***
5. MinPts = Round of Log (Count)
6. If ***MinPts = 1***
// The cluster cannot contain one point only
Break;
Else
// Apply the clustering for this pair of coordinates
Ci = HDBSCAN (***longitude, latitude, MinPts)***
7. add ***Ci*** attribute to the CSV file
8. Export a new CSV file for visualization

Figure 12. HDBSCAN clustering Pseudo-Code **(Campello Ricardo J. G. B, 2013)**

3.5 Trajectory Classification

A. Naïve Bayes and Random Forest

For trajectory classification, we can apply multiple algorithms such as Naïve Bayes, Decision trees, Random Forest, Support Vector Machines, and K-Nearest Neighbor

(KNN). Random Forest (Andy Liaw, 2002) is a classification algorithm based on a generation of compact decision trees using techniques like bagging. However, it overcomes the decision trees overfitting drawback by growing small trees to a limited depth, then it predicts the value by using all decision trees. The class value with the highest vote will be selected in the end. Based on (Ahmad Ashari, 2013) study, decision trees outperformed Naïve Bayes and KNN in terms of execution time. Decision trees had no calculations to perform while predicting the class value. Additionally, in terms of prediction accuracy, Naïve Bayes outperformed decision trees and KNN because there were no dependencies among attributes. For this reason, we decided to select Naïve Bayes and Random Forest algorithms for building our trajectory classification model.

B. Data Sampling

As the number of taxi trips in the dataset is 1.7 million trips, we need to sample this dataset to apply trajectory mining techniques such as classification and sequential pattern mining. These techniques will face limitations such as space and memory issues when trying to process this massive amount of taxi trips. Therefore, we used the same subset that represents tax trips which started in May. We chose this month as it is one of the best times to visit Porto city because of the nice weather and cheap hotels.

C. Districts Extraction

In our dataset, we had longitude and latitude coordinates but with no description of the district for each location. To perform techniques like classification and periodic pattern mining, we need to use the name of the district where the taxi is located, if we want to predict the location at any point and the data is still continuous, then we will have a wide range of class values which makes it difficult for classifiers to perform properly. Therefore,

we used R Studio (RStudio Inc., 2018) to create a script that extracts the address for each pair of coordinates (i.e. longitude and latitude). In this script, we used R ggmap package (D Kahle, 2013), this package uses Google Maps (Google Inc., 2018) online services to reverse geocoding the longitudinal coordinates in our CSV file.

To perform the naïve Bayes classification on the CSV files, we planned to predict three factors. The first factor is the traffic status at the start location (i.e. origin) of taxi trips. The second factor is the traffic status at the end of taxi trips (i.e. destination). The final factor is to predict the traffic status at each point (i.e. trip route) during taxi trips. Therefore, we had to extract three CSV files before performing the R script on each one of them. The first CSV file was called traffic-origin, this file had four main attributes, the weekday, the start time, the longitude and latitude coordinates of the start location of trips, the file had 161016 taxi trips. the second CSV file was called traffic-destination, it had four attributes, the weekday, the start time, the longitude and latitude coordinates of end location of trips, the file had 161016 taxi trips. the last CSV file was called traffic-route, for this CSV file, we couldn't consider all taxi trips which started in May. The reason is that taxi trips in May have various durations, thus, they have a different number of longitudinal attributes. Trips which are longer had more longitudinal attributes than other trips. To perform our classification on similar trips, we had to extract taxi trips with a duration of 5 minutes (i.e. 40 longitudinal attributes), this CSV file had 7333 taxi trips.

After preparing our CSV files, we applied an R script to extract the address for each pair of longitudinal coordinates in each one of the three CSV files. The ggmap package used a function called revgeocode, this function passes the pair of coordinates to the Google services, then it returns the address which corresponds to these coordinates. However, we

had several drawbacks while running our script on the CSV files. The first issue was the number of revgeocode function calls. Since this function uses Google servers to call some services and return the address, Google restricts the usage of this function calls to 2500 per day. If we exceeded the quote, the service will return an error message each time the revgeocode function is called. Since we have large CSV files, we extracted smaller subsets so that we will not exceed the Google service daily limit. After minimizing the size of our CSV files, the traffic-route CSV file had 750 trips with 20 longitudinal coordinates, the traffic-origin and traffic-destination files had 1231 trips each. The total number of revgeocode calls for three CSV files was approximately 9962 calls, which took us nearly 4 days to extract all addresses. However, we faced another issue with the addresses when we executed our R script every time. In our script, we extracted a part of the address of each longitudinal coordinate, this part was the street name and street number. Extracting the street name and number generated a wide range of values which made it hard to classify and predict. However, even when we extracted only the district, 95% of taxi trips coordinates were assigned to Porto district, just a few coordinates were assigned to other districts such as Moreira, Matosinhos, and Vila Nova de Gaia. It seemed that all districts that were in the central area of the city like Sao Nicolau, Se, Santo Ildefonso, Vitoria, and Bonfim were included in Porto district.

Applying this approach resulted in an imbalanced dataset where districts in each CSV file were not evenly-distributed as 95% of taxi trips belonged to Porto district. This approach could cause a poor classification performance with a low accuracy. Therefore, we adopted another approach which generates fewer districts with well-distributed trips among Porto city districts. Since we did not use revgeocode in our new approach, we did not extract

smaller subsets from CSV files, and we used the same CSV files we created at the first time. In the new approach, we used Google Maps (Google Inc., 2018) to divide Porto city into smaller rectangles. For the districts of Porto city, we relied on the Porto urban distinctions research conducted by Pereira in 2018 (Pereira, 2018). In his research, he divided Porto city into 18 various districts, he identified the location (i.e. central, periphery, Atlantic) and the nature (i.e. historical) of each district. In Google maps, we used the distance measurement feature to plot the corner points for each district, after that we plotted another point as Google Maps drew a line between these two points to calculate the distance between them. We kept plotting new points and lines until we had rectangles of all districts in Porto city, each district was a rectangle area with four corners. In the end, we had 18 districts as shown in figure 13.

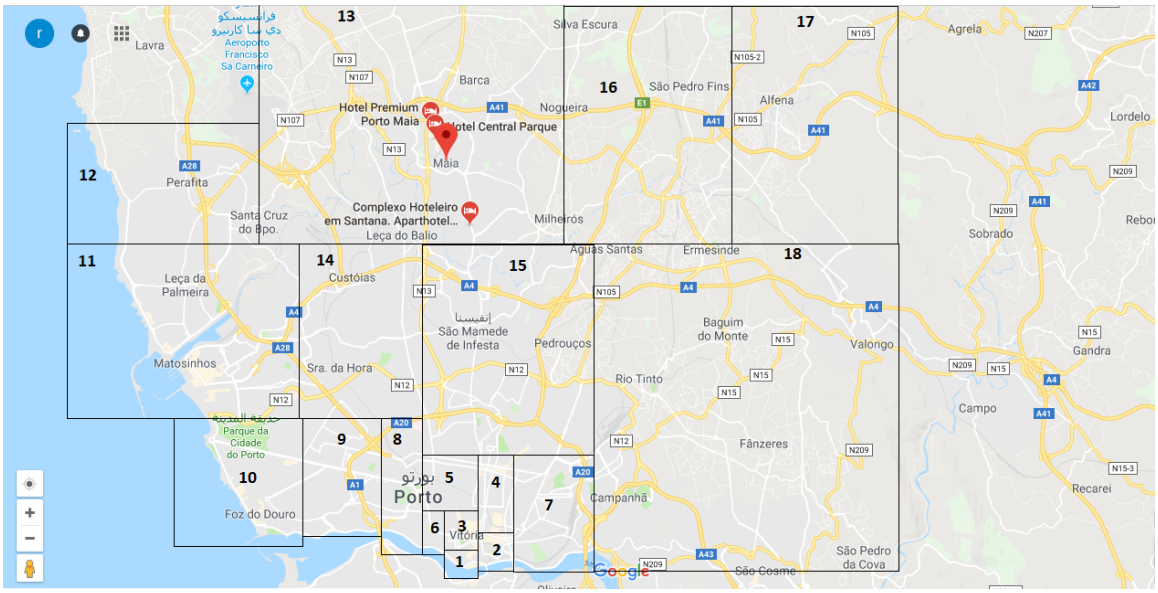

Figure 13. Porto city districts in Google Maps **(Google Inc., 2018)**

To distinguish each district, we extracted the four corners coordinates for each district. We went to each plotted point on Google Maps and with the right click, we could identify

longitude and latitude coordinates of each corner. As we divided the city into multiple rectangles, we assumed that we produced equiangular, which means we generated a rectangle where each parallel side are equal in length. Therefore, we said that a taxi belongs to a district if its current longitude lies between right and left longitude coordinates of the district, and its current latitude lies between lower and upper latitude coordinates of the district. The following table describes the extracted districts and the longitude and latitude coordinates for each border. We can notice that each district has four borders with nearly equal parallel sides.

Table 3. Porto city districts border coordinates

District	Lower Lat.	Upper Lat.	Right Long.	Left Long.
Sao Nicolau	41.139357	41.143865	-8.611466	-8.621299
Se	41.139193	41.148208	-8.600352	-8.611466
Vitoria	41.144025	41.152879	-8.611251	-8.621298
Santo Ildefonso	41.148047	41.164788	-8.60056	-8.611249
Cedofeita	41.152879	41.165274	-8.611247	-8.628137
Miaragaia	41.14274	41.152881	-8.621298	-8.627924
Bonfim&Campnha	41.139827	41.165101	-8.577693	-8.600565
Massarelos	41.14435	41.17252	-8.627924	-8.639681
Lordelo Do Ouro	41.146765	41.172843	-8.639467	-8.661271
Foz Do Douro	41.144509	41.17284	-8.661271	-8.703385
Matosinhos	41.172523	41.210809	-8.664262	-8.732013
Perafita	41.210875	41.262566	-8.675873	-8.73124

Maia	41.210921	41.262463	-8.585343	-8.675948
Ramalde&Aldoar	41.172683	41.210647	-8.628359	-8.664262
Paranhos	41.164609	41.210949	-8.577473	-8.627925
Sao Pedro Fins	41.210791	41.263121	-8.540443	-8.58532
Alfena	41.21103	41.262843	-8.492282	-8.540443
Corujeiera&S.Roque	41.140778	41.211122	-8.491102	-8.57768
Vila Nova De Gaia	41.07919	41.136052	-8.571571	-8.67612

The final step was to scan the three CSV files and insert the district for each longitudinal coordinate according to the districts coordinates in table 3. To accomplish this, we imported the districts coordinates into an Oracle table using Oracle Application Express (Oracle Corporation, 2018). Then we imported the three CSV files into three Oracle tables. After that, we created a PL/SQL script which scanned all CSV files in a loop and compared each longitudinal coordinate with the border coordinates of each district in table 3. The trip coordinates were assigned to a district if they lied within that district's borders. If the trip coordinates did not lie within any district, then the value 'no district' was assigned to that point. Furthermore, we had a few trip points which were lying in more than one district, this could be due to the error of the manual rectangles plotting. There could be a slight overlap between rectangles (i.e. districts) when we plotted the corners and drew the borders in Google Maps. If we had more than one district, we assigned the trip point to one of these districts. At the end of this approach, we had three tables with the following attributes, weekday, start time, longitudinal coordinates, and districts corresponding to each pair of longitudinal coordinates.

D. Traffic Status Extraction

After extracting the districts from longitudinal coordinates in the CSV files, we planned to apply the classification technique on these files to predict the traffic status over time. However, we needed to define the methods which decide if a trip status in a certain place at a given time is a high or low traffic. In our study, we used clustering to detect the density of each point, thus, to define the traffic status at the point (e.g. low or high traffic). Additionally, we applied spatiotemporal clustering, all trips that started at the same time on the same day should be clustered together. Therefore, we divided our three CSV files into smaller files, each file was divided into 7X24 subfiles (i.e. weekday vs hour). Each subfile represented the weekday and the hour when each trip started, applying clustering on the whole CSV file was improper as we assumed that all trips started together. In spatiotemporal clustering, we clustered taxi trips which were in different places in the city but started on the same day and hour. After dividing traffic-origin CSV file into 168 subfiles, we applied clustering on a single pair of longitudinal coordinates (i.e. start) for each subfile. For traffic-destination CSV file, we applied clustering on one pair of longitudinal coordinates (i.e. end). For traffic-route CSV file, we divided it into 168 subfiles, and for each subfile, we applied clustering on 20 pairs of longitudinal coordinates (i.e. the entire trip points).

We applied the same HDBSCAN algorithm (Ricardo J. G. B. Campello, 2013) described in figure 12. This algorithm is based on the traditional density-based clustering algorithm DBSCAN (M Ester, 1996). To implement HDBSCAN, we created an R script which reads each CSV subfile, we used R hdbscan package (Michael Hashler, 2018). The script scanned every pair of longitudinal coordinates separately. Each pair is the dataset where the

HDBSCAN was executed, in each iteration we called hdbscan function which took two parameters, the CSV pair of coordinates and the minimum number of points *MinPts*. In terms of *MinPts* parameter, the optimal case was to set it by a domain expert, and since we did not have the chance to work along with domain experts in our study, we set the *MinPts* to log(*n*), where *n* is the number of points to be clustered. After executing hdbscan function, we extracted two outputs from this function and stored them as new attributes in our CSV files. One output was the cluster number the point was assigned to *C*, and the second output was the degree of membership for each point with respect to its cluster *M*.

The HDBSCAN algorithm can measure the strength of cluster membership of each point in the dataset. The range of this measurement is 0 – 1, where 0 means that the point is a weak member of its cluster, and 1 means that the point is a strong member in its cluster. Therefore, by adopting the degree of membership factor, we could identify the density of that point in its cluster. For example, if a point had a degree of membership of 0.8, that meant it was strongly connected to the cluster, thus, it was most likely in the high-density area of its cluster. And if a point had a degree of membership of 0.1, that meant it was weakly connected to its cluster, thus, it was in the low-density area of its cluster.

To validate the degree of membership measure, we used Tableau desktop software (Tableau Software Inc., 2018) to visualize one clustered CSV file. This CSV file had 80 points with 7 clusters. When we visualized all clusters, we could correspond each point on the map to its degree of membership. For example, in cluster 7 as we can see from figure 14, we had 6 points (i.e. trips) in this cluster. Furthermore, there were two points which belonged to cluster 7, but they were in less dense areas and not in the center of the cluster. When we extracted the degree of membership for these two points, we had the values of 0

and 0.49 respectively. While the other four points had values of 0.93, 0.93, 0.95, 0.96 respectively. Which meant that this cluster had four points in high-density area and two points in less dense areas. Another example was cluster number 4 which is shown in figure 15. This cluster had 8 points, two points were far from the high-dense area and 6 points were in the center of the high-dense area. The two points had less degree of membership with 0 and 0.05 respectively, and the other 6 points had a higher degree of membership with values of 0.67, 0.67, 0.7, 0.75, 0.75, and 0.78 respectively.

We relied on this approach to determine the traffic status for each point in our CSV files. The lower the degree of membership was, the less traffic the trip was likely to have. On the other hand, the high degree of membership meant that the traffic was higher in the trip's location and time.

Figure 14. Cluster #7 trips distribution

E. Classification Preprocessing

The main reason for applying the classification techniques is to predict three factors, the traffic status of trips initiation (i.e. trips origin), the traffic status of trips destination to

detect the availability of taxis, and the traffic status at any given place and time on the route to provide an exploration to taxi drivers of potential visited areas in the city. Each classifier was applied on three CSV files to carry out predictions, where each CSV file represented one of the three factors (traffic status in origin, traffic status in the destination, traffic status on route). Before applying classification, we needed to implement some preprocessing operations on the CSV files. The classifier required a set of variables to predict the class variable, in our CSV files the class variable was the traffic status which was the attribute that we are predicting. The first variable in our CSV files was the trip day, we kept the same values for this variable as we could not discretize them. Trip day variable would have 7 values which are the weekdays (e.g. Monday, Tuesday, Wednesday, etc.).

Figure 15. Cluster #4 trips distribution

The second variable was the trip time, this variable had 24 values representing the hours of the day. For an effective classification, we decided to perform an unsupervised discretization on this variable and categorize it into three periods, morning, afternoon, and

night. We assumed that the period from 21:00 to 4:00 was the night period, the period from 5:00 to 11:00 was the morning period, while the period from 12:00 to 20:00 was the afternoon period. The third variable was the trip area which we extracted previously using Google Maps and districts borders coordinates in table 3. As we had 18 values in this variable, we decided to discretize it and reduce the number of values to enhance the performance of our classifier. we merged some districts into one district, at the end of this process, we had 7 main districts which are East, North, West, South, Central, Se&Santo Ildefonso, and Vitoria&Miragaia.

Finally, we had the class variable which was the traffic status. As we discussed earlier, we relied on the cluster number and degree of membership factors to assign a traffic status for each trip. We decided to have three traffic status values, high, medium, and low traffic. First, we checked the value of cluster number produced from HDBSCAN clustering, if the value of this attribute in the CSV file was 0, that meant the trip point was noise, therefore, we excluded that point and assigned its traffic status to “no traffic”. If the cluster number was not 0, that meant there was a cluster. In this case, we checked the degree of membership *M* and we defined a manual threshold to determine which traffic status value to assign to that point, low traffic, medium traffic, or high traffic. However, when setting the threshold and assigning the traffic status values to the class variable, we wanted to ensure that the values were evenly distributed. If the manual threshold led to an imbalanced dataset, this could cause a poor classification where the classifier would likely prefer one value over the other (Mujalli Randa Oqab, 2016). To avoid the skewed probabilities, we kept manipulating our threshold until we got a balanced distribution of traffic status values in each CSV file as shown in table 4.

We got the balanced distribution described in table 4 by setting the manual threshold for the degree of membership *M* to the values shown in table 5. As we could observe from this table, the threshold ranges were different for the three CSV files. Therefore, categorizing traffic status using equal intervals of *M* (i.e. 0-0.33, 0.34-0.66, 0.67-1) will cause a skewness in the traffic status values. Overall, various threshold ranges for different CSV files is normal since they had a different combination of taxi trips attributes, some files had many high *M* value, while some files had many low *M* value. After preprocessing each variable, each one of the three CSV files had four variables as shown in table 6.

Table 4. Traffic status values distribution

CSV File/Traffic Status	Low traffic	Medium traffic	High traffic	Total trips
Trip origin	58600	87610	103544	249754
Trip destination	82650	72638	42832	198120
Trip route	28424	46954	52020	127398

Table 5. Degree of membership thresholds for 3 CSV files

CSV File/Membership Thresholds	Low membership thresholds		Medium membership thresholds		High membership thresholds	
	From	To	From	To	From	To
Trip origin	0	0.7	0.71	0.93	0.94	1
Trip destination	0	0.7	0.71	0.93	0.94	1
Trip route	0	0.3	0.31	0.8	0.81	1

Table 6. Structure for each CSV file before classification

Variable name	No of values	Nature
Trip Day	7	Domain attribute
Trip Time	3	Domain attribute
Trip District	7	Domain attribute
Traffic Status	3	Class attribute

F. Naïve Bayes and Random Forest (Origin)

To predict the three factors which are trip origin, trip destination, and trip route traffic status, we planned to apply two models of classifiers. The first model was the Naïve Bayes classifier, the second model was the random forest classifier. For the Naïve Bayes classifier, this model relies on the probabilities approach where each value is between 0 and 1. In general, classifiers learn from a set of data called training data, then after applying them on training data, they use the knowledge they learned to make their prediction on the unseen data which we call testing data. However, the naïve Bayes classifier works properly when all attributes are independent, so we applied this classifier as we did not have significant correlations among the four attributes described in table 6.

First, the naïve Bayes classifier would try to predict the traffic status in the areas where trips started. So, we used the traffic-origin CSV file in our classification. The class attribute would be traffic status, while the other attributes were the trip day, trip time, and trip district. Furthermore, we wanted to get a general idea of the origin traffic status in each area before we apply our classifier. As shown in figure 16, the traffic status in each area was represented using a heat map. We extracted the number of trips for each day, time, area, and traffic status from traffic-origin CSV file. after that, we used a heatmap function

in R to create a matrix and visualize the traffic status for each timeframe. The heatmap colors were scaled based on traffic status columns (i.e. high, low, medium). Hence, in each area, the timeframes with high traffic had darker colors than timeframes with less traffic.

For example, in the Central district, we could observe that the highest traffic started on Friday morning followed by Thursday morning and Friday afternoon periods. While the less traffic started was on Sunday night. In the South district, there was no significant traffic initiation most of the days except for Thursday night where we had a moderate-high traffic. For the West district, we had a very high traffic initiation on two periods, Saturday afternoon and Friday morning.

To apply the naïve Bayes classification, we created an R script and used R package e1071 (David Meyer, 2018). We imported traffic-origin CSV file with 249754 rows and split the file into two subfiles, training set, and testing set. The split ratio was 90:10, the training set subfile formed 90% of the original data, while the testing data formed 10% of the original data. Since we had enough data to classify, we did not perform any techniques such as cross-validation to observe the variance and determine the optimal size of the training set. Our training set had 224350 rows and the testing data had 25404 rows.

To measure the accuracy of the classifier, we calculated the counts of traffic status records that were correctly and incorrectly predicted, then we visualized the percentage of correct values for each traffic status value (i.e. high, low, medium).

In addition, we used a random forest classifier to predict the traffic status on the same dataset. Unlike the naïve Bayes classifier which depends on prior and posterior probabilities, random forest classifier builds several decision trees for the dataset, then it uses trees with highest votes to predict the class attribute.

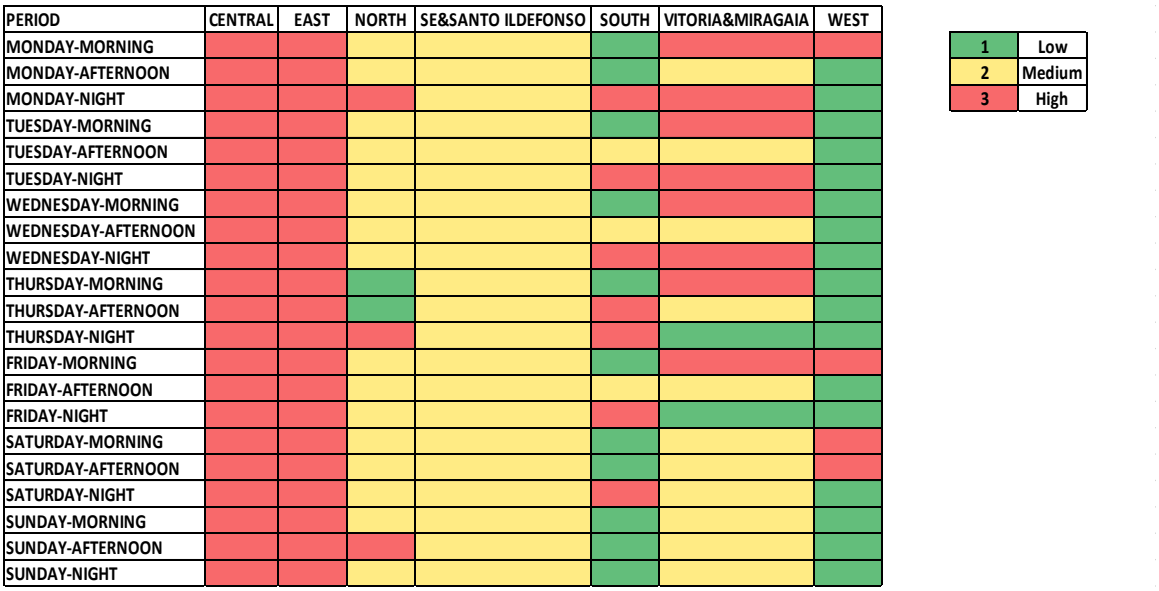

Figure 16. Heatmap for origin traffic status in districts

These trees are based on random subsets of data and independent variables, generated decision trees are different from each other. Decision trees usually provide a correct prediction for the most part of the dataset, attribute values in the dataset will be provided to these decision trees, then passed through the tree nodes, then random forest will combine the output of several decision trees to predict the class attribute in the end. One input parameter is required for the random forest classifier which is the number of trees. If we define many decision trees to be constructed, the random forest classifier will take more time to train the data and predict the class attribute. In our model, we avoided speed and memory limitations for classifying this huge CSV file, so we defined the number of decision trees to be 200.

To apply random forest classification, we created an R script and used R package randomForest (Andy Liaw, 2002). We imported traffic-origin CSV file, then we split the file into two subfiles with a ratio of 90:10.

G. Naïve Bayes and Random Forest (Destination)

We applied the same two models, the naïve Bayes and the random forest classifiers to predict the destination traffic status. We used traffic-destination CSV dataset in our classification, then we predicted the class attribute which was the traffic status. Figure 17 describes the traffic status in each area, we used the heatmap function in R to visualize traffic status columns in various destination areas.

We could observe from the figure that in Central district, the traffic where trips ended was high on Monday morning. Whereas, in the West district, there was a high traffic where trips ended on Saturday afternoon. Meanwhile, in the South district, there were no significant high traffic periods in places where trips ended.

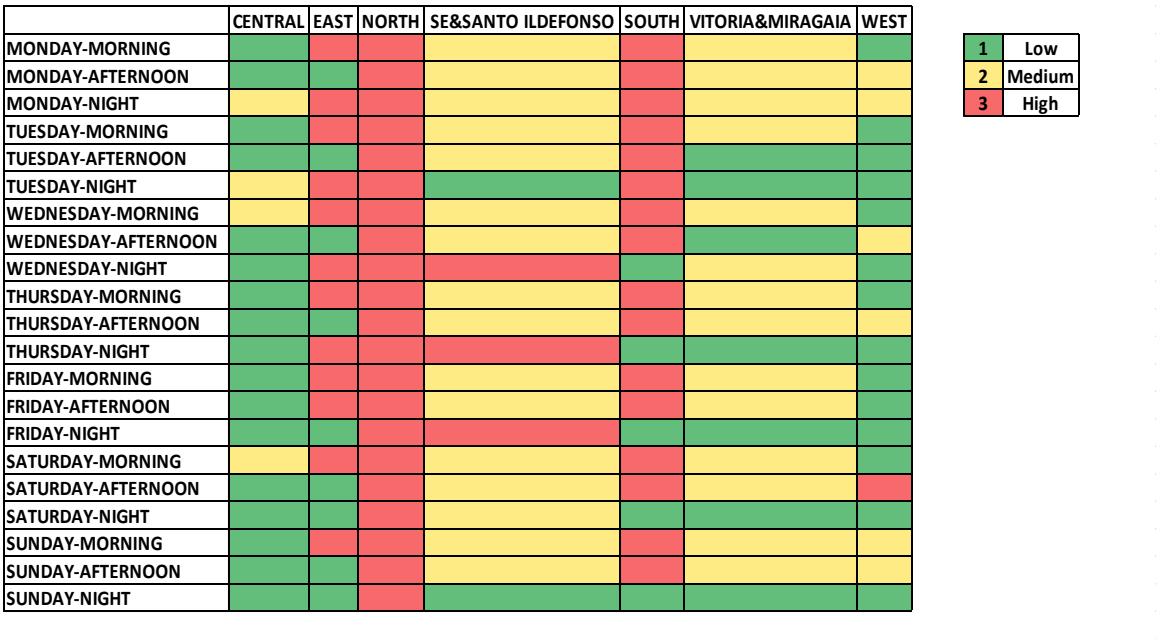

Figure 17. Heatmap for destination traffic status in districts

We used the same R script created previously to import traffic-destination CSV file which had 198120 rows. After that, we split the file into two subfiles with a ratio of 90:10. The

training dataset had 178004 rows, and the testing dataset had 20116 rows. We calculated the number of correct and incorrect predictions, then we visualized the percentage of correct predictions of traffic status (i.e. low, medium, high).

Next, we applied a random forest classifier with setting the number of decision trees parameter to 200. We used the same R script created previously, we imported traffic-destination CSV file, then we split the file into two subfiles with a ratio of 90:10.

H. Naïve Bayes and Random Forest (Route)

We applied the same two models, the naïve Bayes and the random forest classifiers to predict the route traffic status. We used traffic-route CSV dataset in our classification, we predicted the traffic status class attribute. We used the same R script created previously to import traffic-route CSV file which had 127398 rows. After that, we split the file into two subfiles with a ratio of 90:10. The training dataset had 114509 rows, and the testing dataset had 12889 rows. We calculated the number of correct and incorrect predictions, then we visualized the percentage of correct predictions of traffic status (i.e. low, medium, high).

We next applied the random forest classifier, we set the number of trees parameter to 200. We used the same R script created previously, we imported traffic-route CSV file, then we split the file into two subfiles with a ratio of 90:10.

3.6 Sequential Pattern Mining

To help the transportation authorities to have more insights into the flow of taxi trips among different districts, we applied the sequential pattern mining technique. Sequential pattern mining is used to discover patterns in the dataset where data is displayed as a sequence. In our dataset, we considered a set of taxi trips as time series sequences of districts, each trip travels and navigates into a different district over time. Each time series sequence should

have an alphabet which represents the items in that sequence, in our study our alphabet contained 7 elements, each element was the name of a district in Porto city.

The first step in our analysis was to prepare the CSV file for sequential pattern mining. The CSV file should be in a long format where each taxi trip sequence is split into subsequences, each subsequence will have one event. All subsequences of a given trip should be ordered in the CSV file according to their timestamp. For example, if we have a taxi trip which navigated through for districts, T1 = {Santo Ildefonso, Se, Vitoria, Miragaia}. Then this trip will be displayed in the CSV file as follows.

Table 7. Trip sequence distribution over time

Trip id	Time	Event	Size
T1	1	Santo Ildefonso	1
T1	2	Se	1
T1	3	Vitoria	1
T1	4	Miragaia	1

In addition, in the CSV file, we assumed that all events had a size if 1 as we wanted each district to be a separate event. We extracted all trips with a duration of 5 minutes and converted them from a wide format into a long format as shown in table 7. To convert the CSV file from wide to long format, we created a PL/SQL script which read all districts in a row, then inserted each district in that row as a new record in the CSV file. When inserting districts, we ignored repeated districts in the same row as many trips stayed in the same district for a while. We wanted to avoid patterns which had repeated districts such as

{Paranhos, Paranhos}. In the CSV file, some trips navigated two districts, other trips navigated four districts. After preparing the CSV file, we had 5542 trips and 15921 rows.

Next, we created an R script using arules packages (Michael Hahsler, 2018), (Hahsler Michael G. B., 2005), (Hahsler Michael C. S., 2011) and arulesSequences packages (Christian Buchta, 2018). We read the CSV file which had the taxi trips in the long format, then we called cspade function which was based on the SPADE sequential pattern mining algorithm (Zaki, 2001). SPADE algorithm builds a lattice tree for the time series sequence, then it performs three scans to decompose the lattice tree into smaller subtrees processed separately. Finally, it generates patterns with a support value associated with each pattern. The cspade function took one parameter which was the minimum support value, if a generated sequence support value was greater than the minimum support value, then it was a frequent sequence (i.e. pattern). However, setting an improper value for minimum support value could cause poor performance in terms of results and execution time. For example, if we set a high minimum support value, this could generate few patterns and ignore some significant patterns in the time series sequence. And if we set a low minimum support value, the algorithm could generate many insignificant patterns.

In the R script, if we set the minimum support value to 0.01, we got 146 patterns. We wanted to shortlist the generated patterns to visualize them on the Port city map, so we set the minimum support value to 0.04 which generated 41 patterns in the end. From the 41 patterns, 14 patterns were single patterns with one district (i.e. the trip path was within one district), 24 patterns were two-districts and 3 patterns were three-districts patterns. Table 8 shows the top five generated two-districts patterns with highest support values.

Table 8. Patterns generated by SPADE algorithm

Sequence	Support value
<{VITORIA}, {SE}>	0.115662
<{MASSARELOS}, {LORDELO DO OURO}>	0.099603
<{VITORIA}, {MIRAGAIA}>	0.098
<{SE}, {SANTOILDEFONSO}>	0.095
<{SE}, {VITORIA}>	0.094

Furthermore, to have a better understanding of patterns generated, we visualized 25 patterns with highest support values to analyze the flow of taxi trips among Porto city's districts. Figure 18 shows Porto city urban distinctions based on the author's work (Pereira, 2018) along with our sequential patterns. We plotted each pattern as an arrow to define its flow and the districts it navigates to. In this figure, the arrows in red represented trips patterns over two districts, and blue arrows represented trip patterns over three districts.

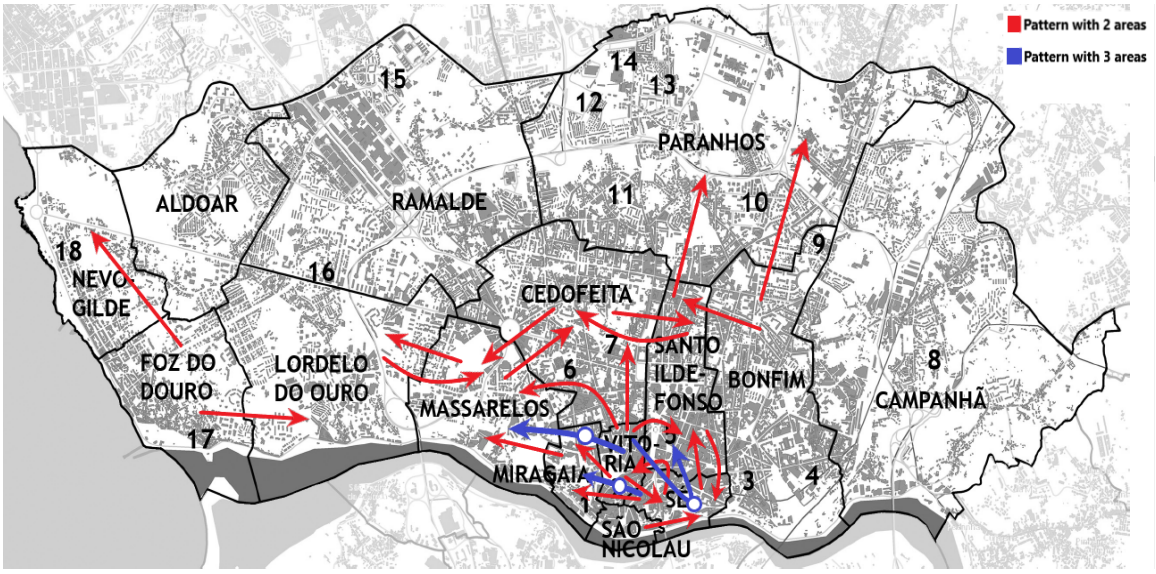

Figure 18. Taxi trips patterns over districts **(Pereira, 2018)**

We could notice that most of the patterns generated were trips with two districts. Furthermore, most trips patterns were flowing within the Central areas and mainly in Sao Nicolau, Se, Vitoria, Miaragaia, Santo Ildefonso, Massarelos. This meant that these districts have a lot of attractions and facilities that keep people and tourists strongly attached to them. However, we identified few patterns heading North to Paranhos and West to Lordelo Do Ouro, Foz Do Douro, and Matosinhos.

3.7 Time Series Analysis

This is the last analysis technique we applied. Each taxi trip in our dataset is a time series sequence, it is a sequence of longitudinal coordinates (i.e. longitude and latitude) over timestamps. We created a two-dimensional time series dataset; the first dimension was the time where the second dimension was the longitude. To apply time series analysis, we had to convert our CSV file from wide format into long format. First, we extracted all trips with a duration of 5 minutes into a CSV file, then we created a PL/SQL script to iterate through the longitude columns in each row and insert each longitude column as a separate row in the same taxi trip table. When inserting new rows, we inserted both the longitude coordinate and its timestamp. Since we knew that the time frequency for our dataset is 15 seconds, we started with a timestamp of 15 for the first longitudinal coordinate, then we incremented each new record by 15 seconds until we reached the end of the trip (i.e. last row with a time of 300 seconds). Table 9 shows a sample from the long-format CSV file for one trip where first 6 longitudinal coordinates are shown.

After that, we divided the long-format CSV file into 7 subfiles, each subfile represented trips which started on a specific day (i.e. Friday, Saturday, etc.). Then we used R script to plot time with longitude, so we could visualize the trend of locations over time.

Table 9. Time series sequence for a given trip

Time	Trip	Day	Longitude
15	1	Friday	-8.636049
30	1	Friday	-8.635986
45	1	Friday	-8.635968
60	1	Friday	-8.635896
75	1	Friday	-8.635077
90	1	Friday	-8.633664

However, we had hundreds of trips in each subfile. Therefore, the visualization of time series trips in each weekday did not provide the useful information about the trend of trips on that day as we notice from figure 19 which describes the trips time series on Friday.

To extract the useful knowledge, we decided to apply a different approach. So, we divided the weekdays CSV files into smaller subfiles. We assumed that visualizing a fewer number of trips will provide more understanding of the trips time series behavior. We created two CSV files, one CSV file represented taxi trips which started on Friday morning in Sao Nicolau district and ended at Se district.

By following this approach, we grouped all similar taxi trips together. The second CSV file contained taxi trips which started on Saturday night in Lordelo Do Ouro and ended at Massaleros. Each CSV file had 17 trips with a duration of 5 minutes each. Next, we created an R script and used two packages, the zoo package (Zeileis Achim, 2005) and Tclust package (Montero Pablo, 2014). The zoo package was used to combine each longitude with its timestamp, while the TSclust package was used to perform clustering on the 17 trips.

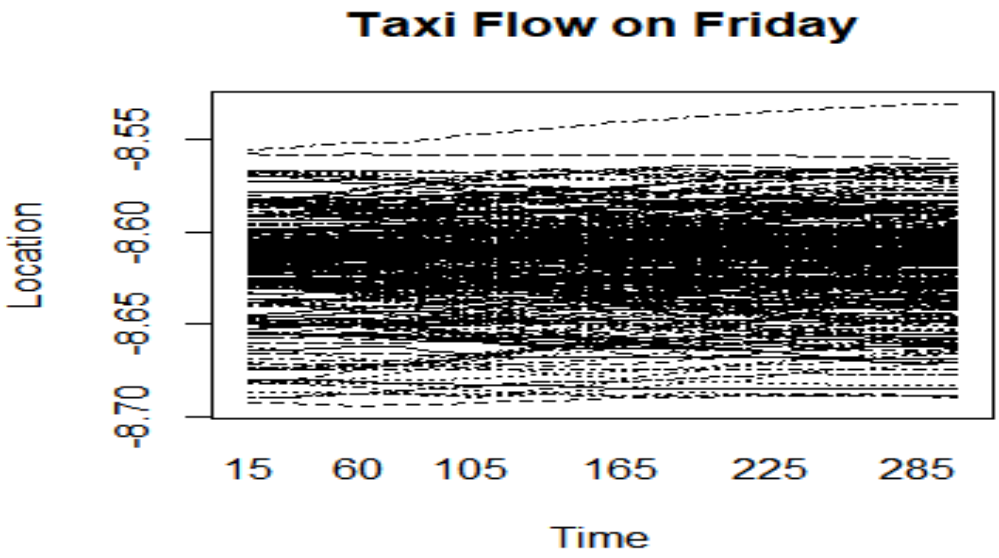

Figure 19. Taxi trips for trips started on Friday

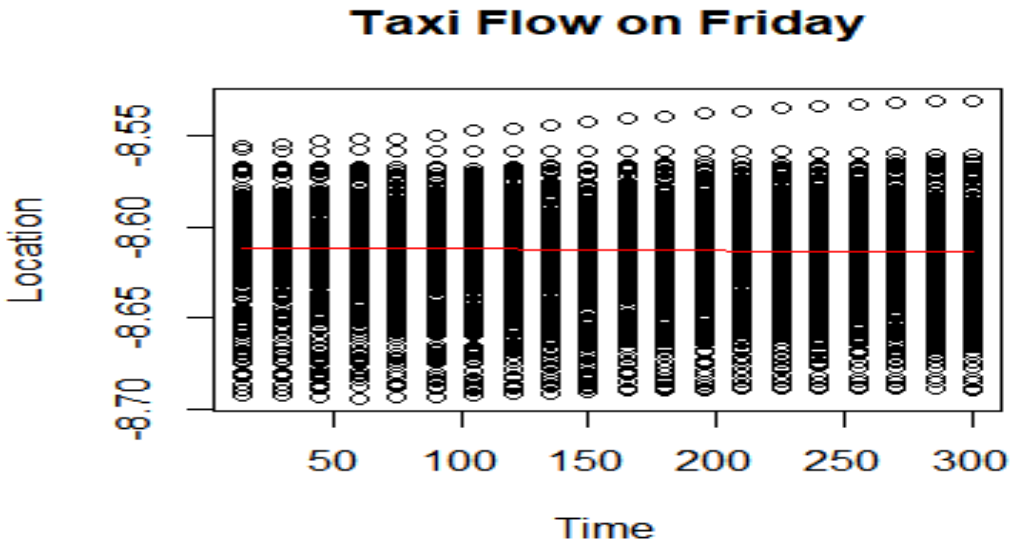

Figure 20. Taxi trips for trips started on Friday

First, we imported the Friday trips CSV file, then we applied hierarchical clustering based on the similarity among trips. Before that, we had to determine the optimal number of clusters to produce. So, we grouped all points in the dataset based on with-in-sum-of-

squares (WSS), then we drew a WSS plot to visualize the number of clusters against the sum of squares as shown in figure 21, then we identified the elbow criteria which would be the number of clusters. As we could notice, the elbow criteria, in this case, was 3, which meant that we will consider the number of clusters $k = 3$.

The next step was to compute the similarity matrix among the 17 trips. To do this, we used the diss function which required two parameters, the dataset and the distance measure. For our analysis, we used the Euclidean distance measure, so we passed the value "EUC" to the diss function. Then we called the hclust function to perform hierarchical clustering for the 17 trips based on their similarity matrix. We passed the similarity matrix calculated previously along with the number of clusters k=3 which was extracted from the WSS plot in figure 21.

To visualize the clustering outcome, we plotted the dendrogram which is a diagram that is used for this purpose. Figures 22 shows the clustering outcome for both Friday and Saturday CSV files. We could notice from the figure that on Friday morning, trips T4 and T11 were grouped together which meant they were very similar in terms of the distance measured. However, trip T7 was not within any group which meant it was somehow far from other trips. On Saturday night, trips T11 and T16 were grouped together while trip T17 was far from other trips.

To correspond the clustering dendrograms with the time series trend of each trip, we plotted the time series of the 17 trips on one diagram, so we could analyze their trend and relate their visualization to the clusters we had in previous dendrograms. Figure 23 shows the time series plot for the 17 trips in the Friday morning CSV file. While figure 24 shows the time series plot for the 17 trips in the Saturday night CSV file.

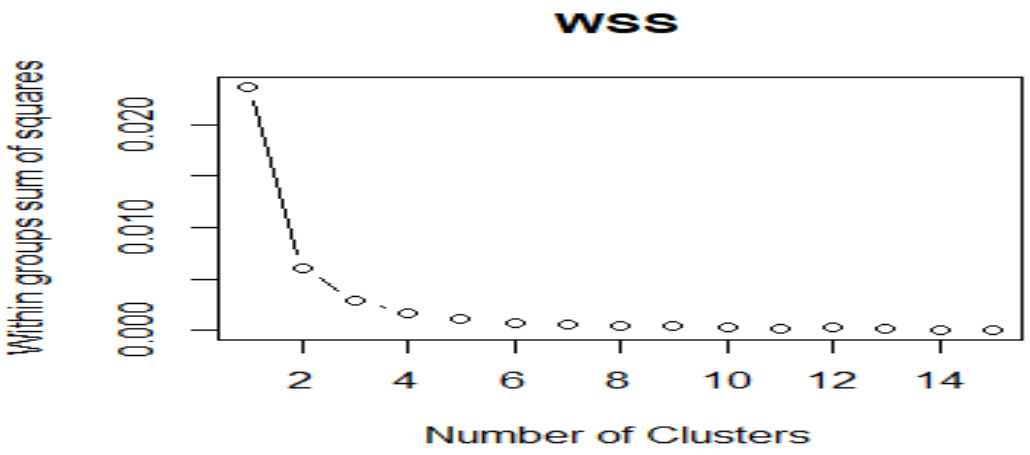

Figure 21. WSS plot with an elbow

If we correlated the visualizations of Friday morning trips in figure 22 and figure 23, we could realize that trip T7 did not belong to any group in the dendrogram, and this conclusion was confirmed in figure 23 where the red arrow points to trip T7. We could observe that the time series line for trip T7 was far from other trips and was somehow taking a different path. In terms of Saturday trips, we observed that trip T7 was taking a different path. Additionally, we could see a couple of trips T11 and T16 located in the same group. In figure 24, both trips T11 and T16 were pointed to using the red arrow and a circle, they appeared to be close to each other. However, both trips were taking a different path and not following the same trend as other trips.

Overall, we could cluster time series for taxi trips based on their similarity. This approach was beneficial for identifying similar groups of trips where taxi drivers shared the same behavior. Furthermore, we could detect trips which were not following the taxi trips general trend. This could be useful for identifying taxi drivers who behaved differently and followed other paths to reach the same destination.

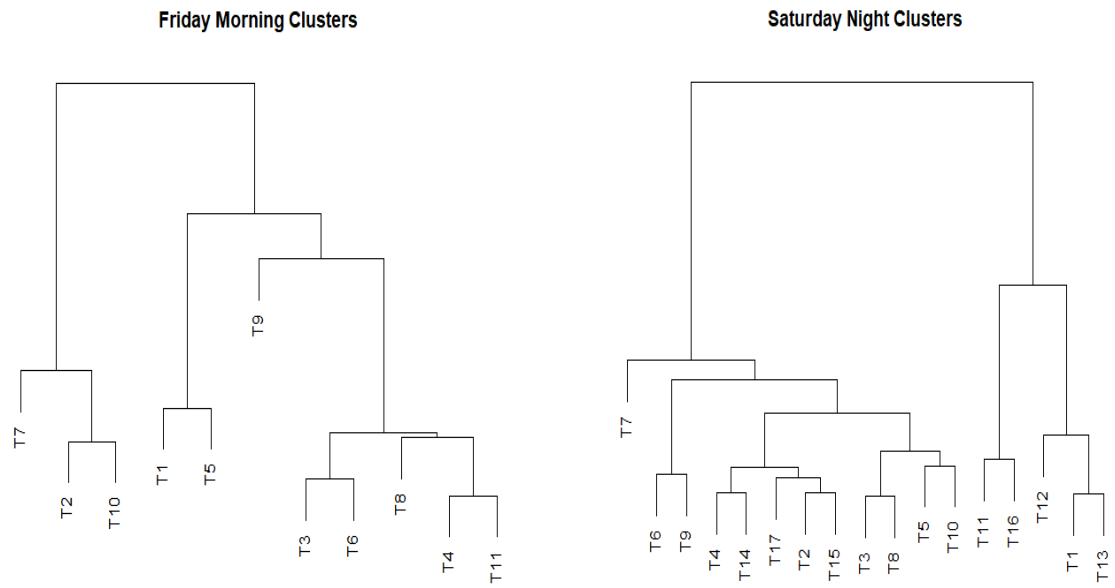

Figure 22. Hierarchical clustering dendrograms

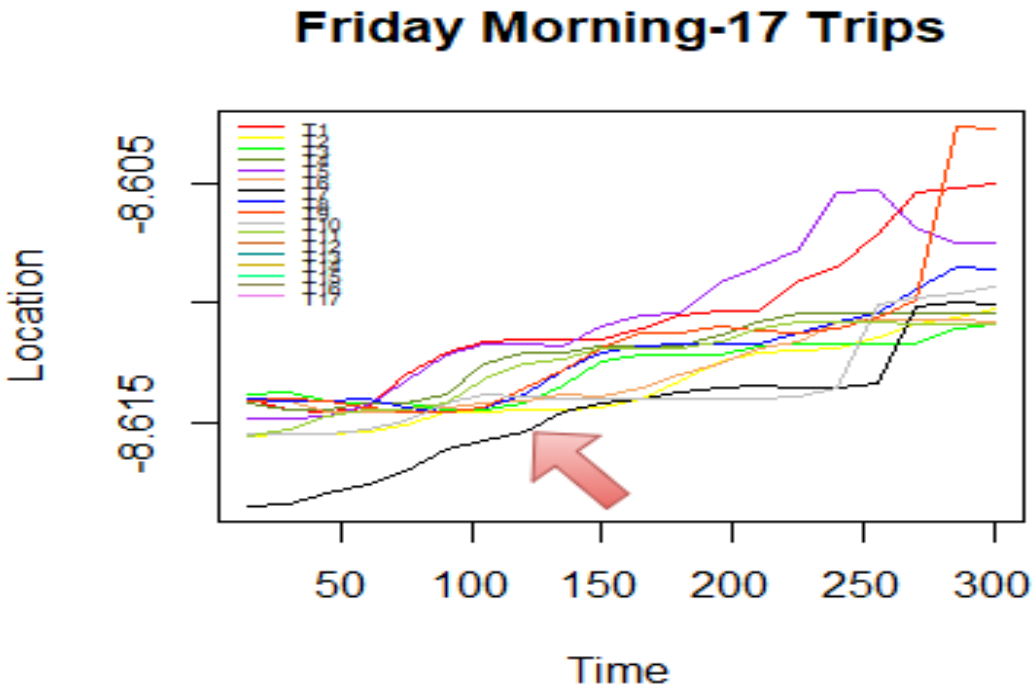

Figure 23. Trips on Friday morning from Sao Nicolau to Se

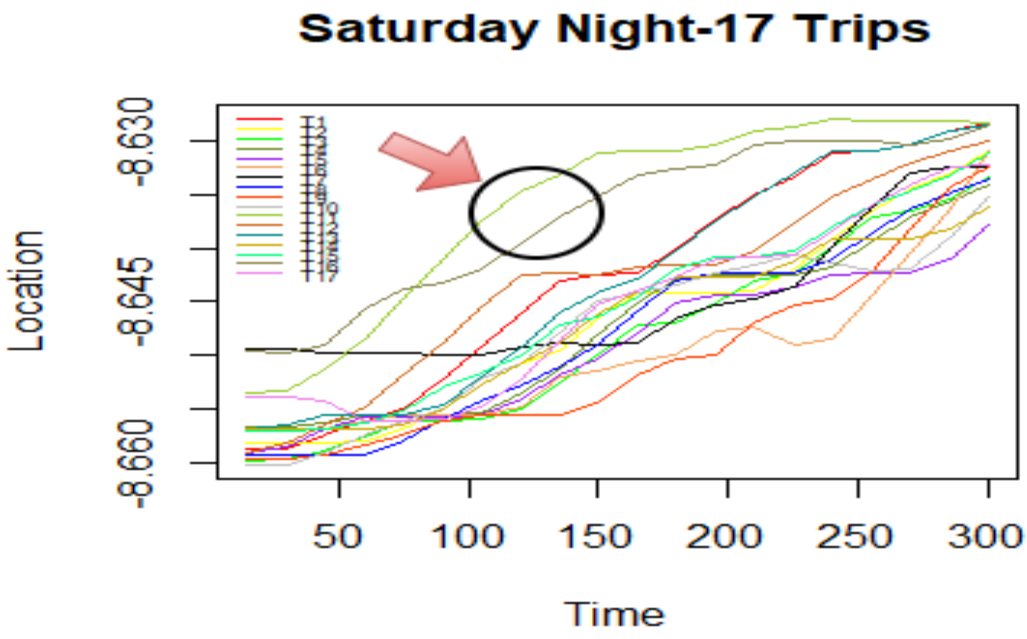

Figure 24. Trips on Saturday night from Do Ouro to Massaleros

3.8 Interactive Visualization

To provide an interactive visualization which could help people to learn more from our dataset, and to give the chance to explore the dataset by passing different parameters, we created R shiny (Winston Chang, 2018) slides for both sequential pattern mining and time series analysis techniques applied previously. By applying R shiny, we could create applications which run on the local machine or through web browsers without the need to learn other programming languages.

In terms of sequential pattern mining, we took the same R script we created previously. However, we used the shinyapp function which had two parts, ui, and server. The ui part defined what appeared to the user in the slides, and server part defined what we want to do internally to visualize the data. In the ui part, we usually defined all fields (i.e. parameters)

which were visible the first time we run our R shiny presentation. The idea behind using R shiny was to create an interactive sequential pattern mining with dynamic parameters instead of static values.

For instance, instead of passing a minimum support value of 0.04 to the cspade function as we did previously, we create an input parameter which can take any given value and retrieve sequential patterns accordingly. In this case, the user can run the SPADE algorithm several times, each time with a different support value. We did this by creating an input parameter in the ui part, this input field was a dropdown list where the user could pick one of the displayed support values which were 0.01, 0.02, 0.03, and 0.04.

After that, we called the cspade function in the server part and we passed the value of the input field we created. We plotted the patterns and their support values in a table on the R shiny slide. Figure 25 shows the slides generated when we run the presentation.

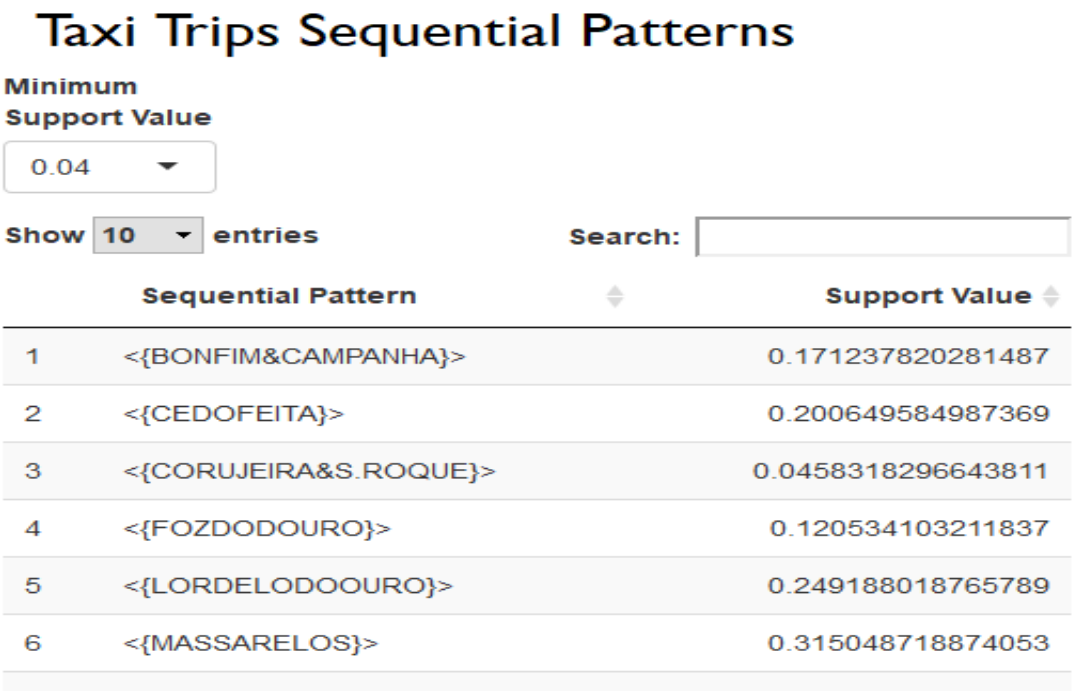

Figure 25. R Shiny slide for sequential pattern mining

As we can notice from the slide, R shiny gave us the ability to view any number of patterns per page. The default value for the minimum support value was 0.04 and once we run the presentation, R shiny would display all the patterns with support value larger or equal to 0.04. Additionally, we could sort patterns based on their support value in an ascending or descending order. And if we wanted to search for a specific pattern, we could enter the district name in the search box.

For example, if we wanted to display all the patterns which Se district is part of, we could fill the search box with the value “Se” and press return to refresh the page and view the filtered patterns again. Furthermore, when we selected different support value from the list like 0.01, we noticed that more patterns were generated, and the number of pages increased. This was because of the cspade function in the server part, it generated more insignificant patterns when we provided a low support value as we discussed earlier.

For the time series analysis, we defined the distance measure as a parameter. We created an R script and defined an input parameter in the ui part which was the distance measure. This input field was a drop-down list with two value, “EUC”, and “DTW”. “EUC” stands for Euclidean distance and “DTW” stands for dynamic time warping.

After that, we called the diss function from inside the server part and we passed the value of the input parameter. Then we called the hclust function to perform hierarchical clustering based on the diss similarity matrix, the last step was to display the dendrogram for the trips. Figure 26 shows the dendrogram for clusters based on Euclidean distance since the default value for the input field was “EUC”. However, when choosing “DTW” as a distance measure, the dendrogram clusters looked different as dynamic time warping had a different mechanism based on optimal timeseries match.

Euclidean distance was used to be a one to one measure, the shortest distance between corresponding points on time series lines is calculated. But in dynamic time warping measure, it could distinguish the underlying patterns and map curves together. DTW could overcome machine limitations of identifying similar shapes with various magnitudes. Therefore, DTW would perform better than Euclidean distance for shapes which are similar.

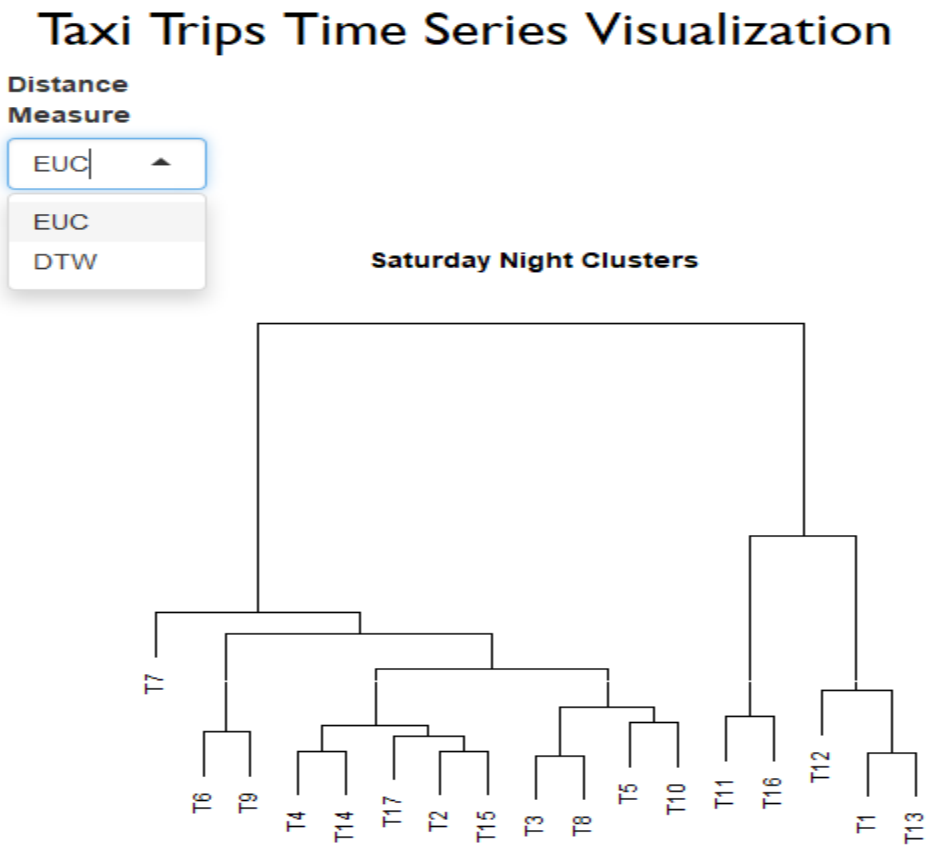

Figure 26. R Shiny slide for time series clustering

Overall, R shiny interactive visualization could provide the ability to study the trajectory data mining techniques. Therefore, transforming data mining techniques into the interactive visualization mode would enhance the understanding of their mechanism.

Chapter 4: Evaluation

4.1 Clustering Evaluation

A. Activity Index

One method to evaluate our clustering analysis is by studying the activity index of taxi trips. In this evaluation, we relied on the number of clusters produced for the trip's origin points coordinates in each CSV file. In figure 27, we can observe the number of clusters produced for each CSV file in 24-day hours. As we can notice from this figure, the highest number of clusters was in the early morning hours. The peak number of clusters was at 3:00 AM with 156 clusters, after that the number of clusters slightly decreased until midnight. This is a reasonable outcome as people tend to be more active in the early morning hours while they are heading to their work or school.

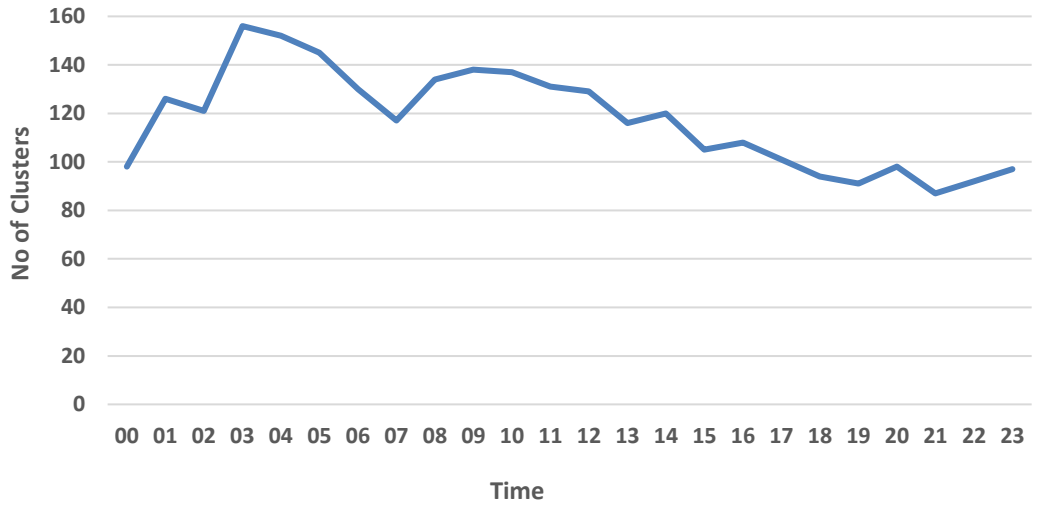

Figure 27. Number of clusters in 24 hours

To perform a comparative evaluation of our clustering analysis, we studied the active index for taxi trips on two different weekdays and extracted the behavior of people in these two

days. Figures 28 and 29 show the number of clusters produced for 24-day hours on Wednesday and Saturday respectively. In figure 28, we can notice that the general trend of the number of clusters was decreasing over time. The early morning hours had the highest number of clusters, then the number of clusters started to decrease slightly until it reached the bottom at midnight hours. This is an indication that for regular weekdays, people tend to be more active at early morning hours where they plan to go to their work or school, while they prefer to stay inactive at the evening hours preparing for their next day. Meanwhile, we can observe from figure 29 that the general trend of the number of clusters was increasing over time, unlike the regular weekdays. We had an approximate steady number of clusters during early morning hours. However, the number of clusters significantly increased in the evening hours. This result indicates that on the weekends, people tend to be active in the afternoon and evening hours. Since they are not working the next day, people spend more time visiting restaurants and other attraction areas in the city.

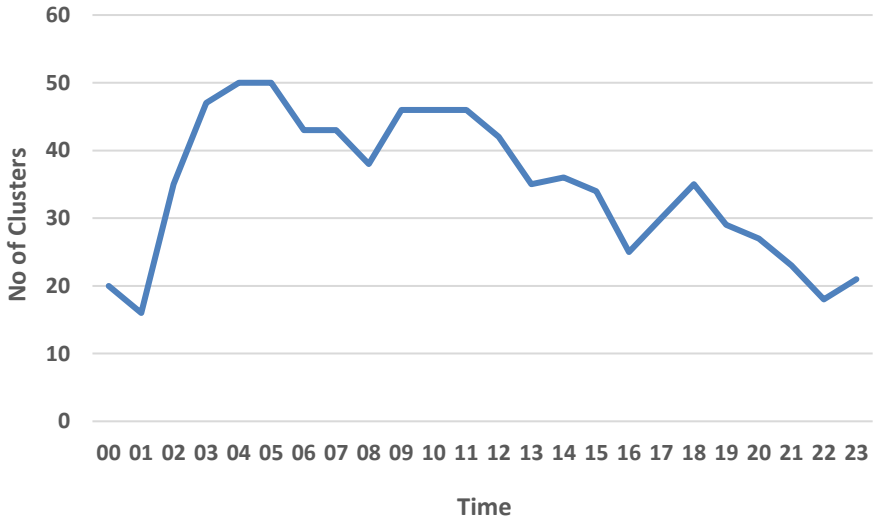

Figure 28. Number of clusters on Wednesday in 24 hours

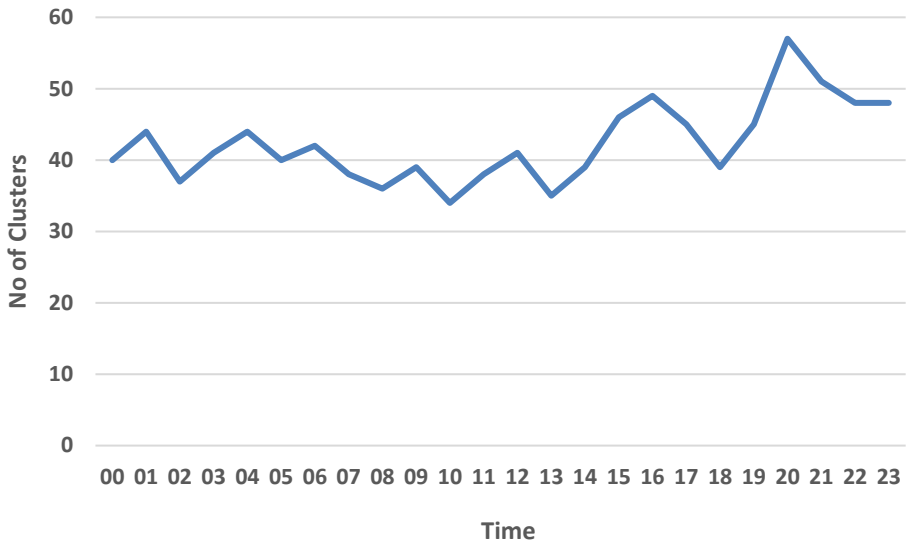

Figure 29. Number of clusters on Saturday in 24 hours

B. Qualitative Analysis

To perform this analysis, we took all CSV files generated from HDBSCAN clustering process described earlier, then we visualized each one of these CSV files using Tableau Desktop Software (Tableau Software Inc., 2018). First, we established a connection and defined each CSV file as a data source for the Tableau workbook. Then we defined each pair of longitude and latitude coordinates as geographical attributes, after that we created a new datasheet and we plotted longitude and latitude as columns and rows respectively. To show clusters on Porto city map, we defined the newly added cluster column as an attribute in the datasheet, then we assigned the color property to the attribute so that we can distinguish different clusters while visualizing the map as shown in figure 30. We kept creating a datasheet for each cluster column (i.e. a pair of coordinates), each pair of coordinates represented clusters within a specific timeframe. So, if we have a trip with 100 cluster columns, that means this trip have 100 moving points. Therefore, we created 100

datasheets in Tableau, each sheet visualized taxi trips clustering within a specific timeframe. At the end of this visualization process, we produced one visualization workbook for each CSV file. Thus, we had 24 Tableau workbooks for the 24-hours CSV files, and 168 Tableau workbooks for the weekday vs 24-hours CSV files.

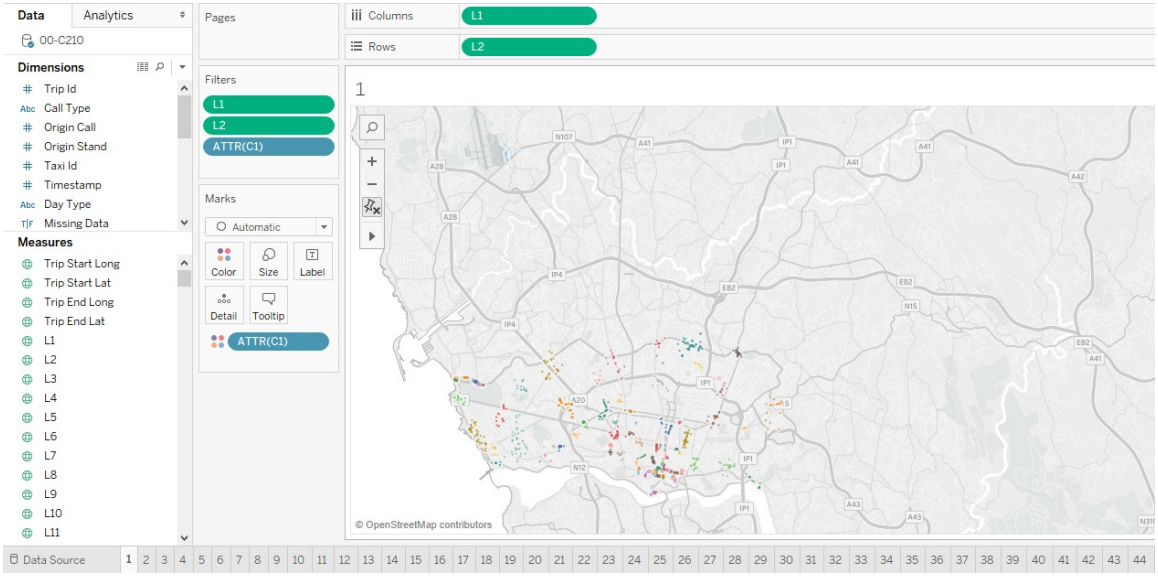

Figure 30. Clusters visualization for one timeframe **(Tableau Software Inc., 2018)**

After visualizing taxi trips clusters on the map of Porto city, we needed to determine the attractive areas in this city. We expected to observe the distribution of clusters and decided if trips followed a pattern by determining hotspots and attractive areas. In general, we expected to find some area which was regularly attractive. Meanwhile, other areas in the city might have a level of attraction which varies over time.

To study the attractive areas in Porto city, we listed the most important areas by reading websites that provide guidelines for tourists when visiting Porto (PlanetWare Inc., 2018). These websites give people detailed information and some recommendations for the places to visit and stay in. Furthermore, we used other resources such as Google Maps (Google

Inc., 2018) and Turinta Maps (Web2Business Inc., 2016) to browse Porto city and discover the main districts and neighborhoods. The following is the list of main attractive areas and a brief description for each area (Wikimedia Foundation Inc., 2018):

1- **Porto Airport (north area):** The only international airport in Porto. The second busiest airport in Portugal, it is located northwest of the center of the city.

2- **Luis I Bridge:** Double deck metal bridge, lower level for buses and cars, upper level for metro and pedestrians. It crosses the Douro river and connects Porto city with Gaia and is located to the south of the center of Porto.

3- **Gaia (south area):** A city that is located to the south of Porto on the other side of the Douro river. It is famous for the port wine cellars which is a major attraction for most tourists.

4- **Boavista (central area):** It has Casa da Musica where the popular orchestra and musical concerts take place. Also, there is the Boavista circle, where people can find shopping centers, hospitals and a lot of restaurants.

5- **Miragaia (historical central area):** it has many historical buildings such as Palacio da Bolsa (i.e. the stock exchange palace) and St Francis church of Porto which was built in the 13th century.

6- **Vitoria (historical central area):** it has many attractive areas such as Igreja do Carmo church which was built in the 18th century, and University of Porto which is the second largest university in Portugal. Furthermore, people in this area can find the famous Lello & Irmao bookstore which has a marvelous stairway and exquisite wooden walls. It is one of the top-rated bookstores in the world.

7- **Santo Ildefonso (central area):** It has the Bolhao market which is a famous outdoor market. Also, there is the famous Porto city hall with the tall clock tower. Tourists can go to Café Majestic which is one of the most beautiful cafes in the world. This area has the Sao Bento railway station with all tiles which tell the historical life scenes of Portugal.

8- **Se: (historical central area):** this area has the Porto cathedral church which has one of the oldest monuments. Also, it has Ribeira square with many cafes and restaurants alongside the Douro river.

9- **Aldoar (southwest area):** it has several restaurants, also tourists can visit Porto city park which is the largest urban park in Portugal. In this 83-hectare park, visitors can enjoy the beautiful view of the Atlantic Ocean.

10- **Matosinhos (north area):** It is well known for its beaches where tourists can find Seafood restaurants and barbeque.

11- **Maia (north area):** It is one of the industrial areas in the city. Major factories and leading products companies are in this area such as textiles and food products companies.

12- **Norte Shopping (southwest area):** It is a huge shopping center where visitors can find everything they are looking for. It has 30 restaurants, 8 cinemas, childcare, health, and fitness clubs.

13- **Bonfim (east area):** It is an industrial area. However, people can find some historical buildings such as the church of Bonfim which was built in the 19th century and the old textile factory.

We studied each one of the clusters visualization files generated in Tableau along with the previous list of attraction areas. In our analysis, we focused on observing the attraction areas with the highest density clusters, and how did the density of clusters change over

different timeframes. Additionally, we analyzed the flow interactions among clusters in different timeframes. This flow interaction was observed from the change in clusters density and size in different areas. For the pattern's identification, we extracted trips movement patterns based on two factors. Movement patterns occurrence in 24 hours, and movement patterns occurrence on weekdays vs 24 hours.

Airport trips: there was a high flow of taxi trips heading from south area to the airport, the high traffic started from midnight hours at 11:00 PM until 10:00 AM in the morning. High- density clusters in this period indicated the high traffic towards Porto international airport as shown in figure 31. It seems that most departure flights took off at the late night or early morning hours. Unlikely, few trips were heading from the south area to the airport in the afternoon and evening hours.

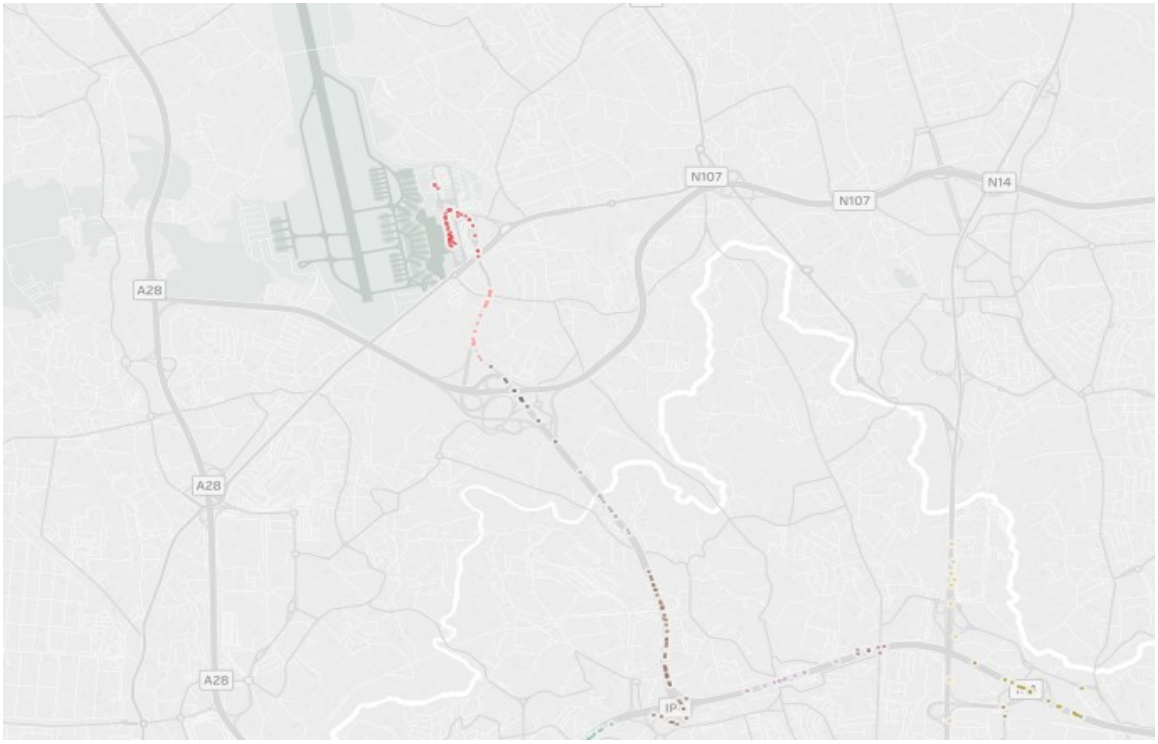

Figure 31. High traffic heading toward the city airport at 2:00 AM

Luis I Bridge traffic: there was a daily high flow of trips crossing the bridge at the midnight hour, then the traffic slowed on the bridge until it was 5:00 AM at the early morning. after that, in the morning hours to 04:00 PM in the afternoon, a high traffic flow of trips was crossing the bridge towards Gaia area. Then for the rest of the day, the traffic on the bridge remained slow. It seems that the bridge gets busy when people are heading toward Gaia in the morning and afternoon. However. In the evening and night hours, they tend to stay in Porto city.

Gaia trips: the traffic in Gaia was nearly relative to the traffic on Luis I Bridge. When there was a heavy traffic on the bridge, there were some busy areas in Gaia. Meanwhile, when the bridge was not busy, traffic in most areas of Gaia was slow.

Central areas traffic: when observing the traffic in Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso. We noticed that traffic was high in these areas throughout the day and night hours. This means that the central areas are always busy since they have the most attractions in Porto city such as restaurants, churches, cafes, and markets.

Porto city park: based on our observations, taxi trips started to travel from the city center to the city park in the west at 6:00 PM, the high traffic in the park stayed the same until midnight hours when people started to travel back to the center areas of the city as we can see in figure 32. The traffic around the park stayed slow for the morning and afternoon hours, then it started to get busy again in the evening hours. It seems that people prefer to finish their work then spend a few hours enjoying the park.

Matosinhos trips: the traffic in this area was slow in the morning hours, then in the afternoon (2:00 PM–4:00 PM) when it is lunchtime, the traffic started to get busy as shown in figure 33. After that, the traffic slowed again in the evening and night hours. However,

the traffic returned to get busy again around the midnight hours. It is obvious that this place had two active periods, the lunch and dinner periods. This is reasonable due to the wide collection of seafood restaurants.

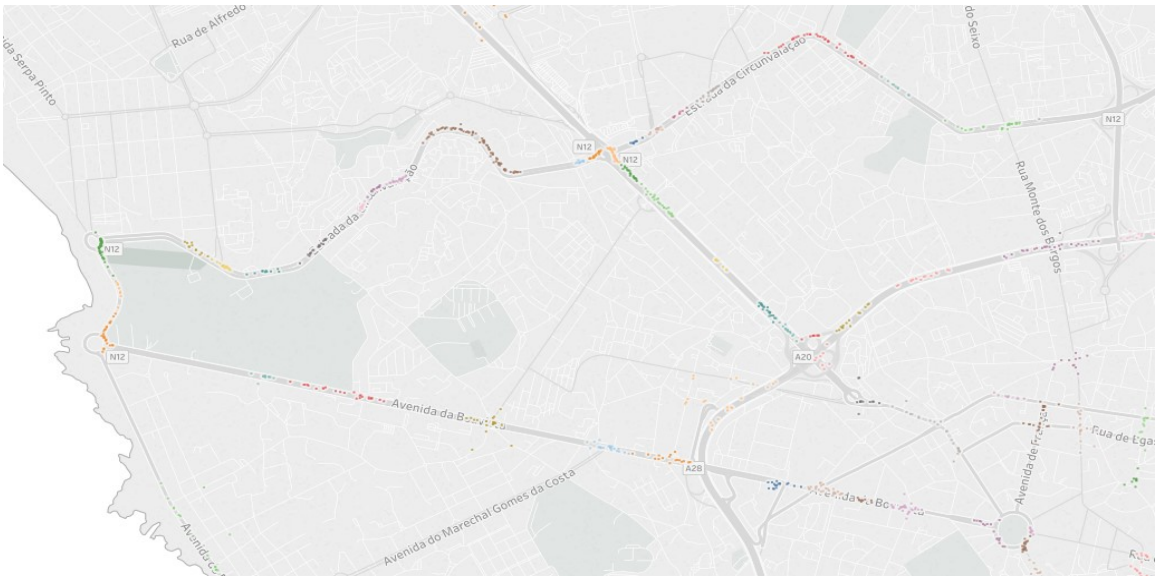

Figure 32. Traffic heading from center areas toward city park at 8:00 PM

Figure 33. High traffic in Matosinhos at 3:00 PM

Norte shopping center: the busy hours of the shopping center started from the early morning around 05:00 AM and remained busy until the evening hours around 05:00 PM. After that, the traffic around the shopping center stayed slow until the end of the day. It seems that people tend to spend more time in the shopping center during the day hours, while they prefer to spend the evening and night hours in the outdoor areas enjoying the cool weather and nightlife.

For the weekday's patterns extraction, we went through the visualization files day by day to observe the trips behavior. We described the behavior of trips in mentioned attraction areas for every day of the week. Visualizing trips in each day and for each hour produced a smaller number of trips. However, it provided a deeper understanding and a detailed comprehension of the trip's behavior and flow among different areas.

Monday/Airport trips: the period where we found a high traffic was the midnight hours (11:00 PM–00:00 AM). There was a high flow of trips heading from the south area to the airport during these couple of hours.

Monday/Central areas traffic: the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) got busy starting from early morning hours at 3:00 AM until the evening hours at 6:00 PM as shown in figure 34. Most trips at this period flowed from west and east areas to the central area. After that, the traffic got slower for the rest of the day.

Monday/Porto city park: the traffic in the city park started to get busy in the evening hours at 6:00 PM. Most trips flowed from east and central areas to the west area where the park is located. The traffic in the park remained busy until the midnight hours. The flow of trips started to take the opposite direction at 9:00 PM where trips headed from west areas to east and central areas.

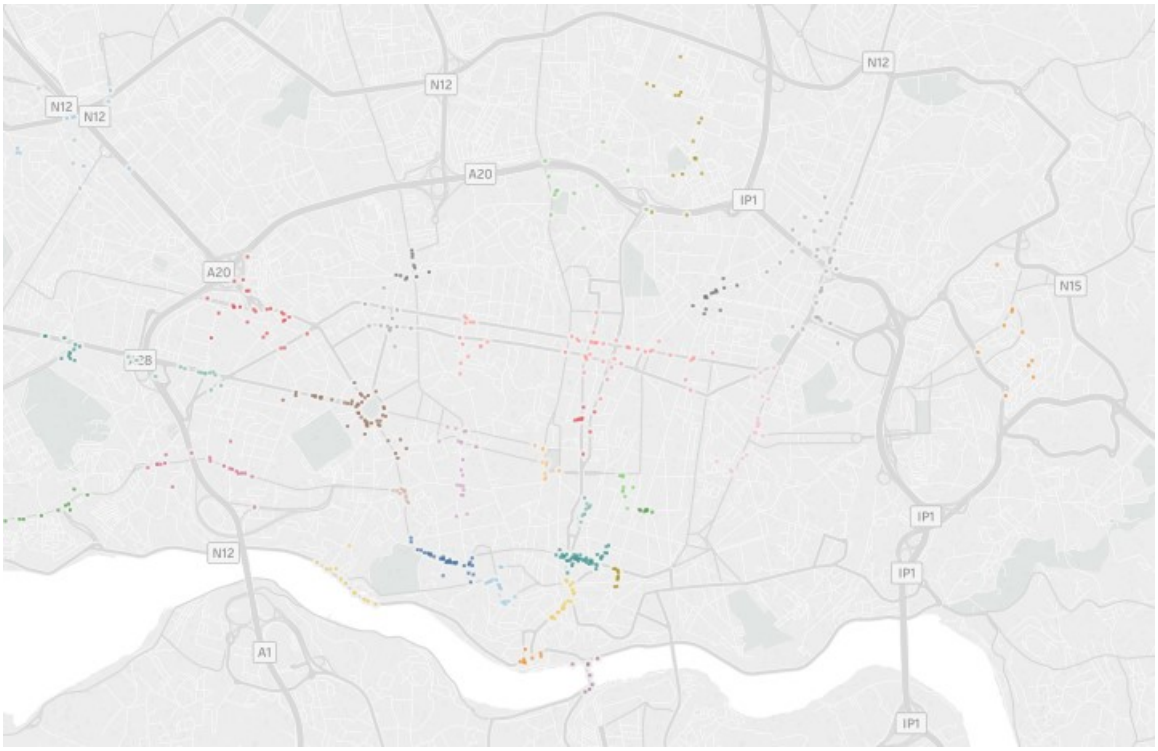

Figure 34. High traffic detected in the central area on Monday at 10:00 AM

Tuesday/Airport trips: the traffic was busy in the midnight hours (11:00 PM–01:00 AM). There was a high flow of trips heading from the south area to the airport during this period.

Tuesday/Central areas traffic: the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) got busy starting from early morning hours at 3:00 AM until the evening hours at 7:00 PM. Most trips at this period flowed from west and east areas to the central area. After that, the traffic got slower for the rest of the day.

Tuesday/Porto city park: the traffic in the city park started to get busy in the evening hours at 7:00 PM. Most trips flowed from east and central areas to the west area where the park is located. The traffic in the park remained busy until the midnight hours. The flow of trips started to take the opposite direction at 9:00 PM where trips headed from west areas to east and central areas.

Wednesday/Airport trips: the period where we found a high traffic was the midnight hours (11:00 PM–02:00 AM). There was a high flow of trips heading from the south area to the airport during this period.

Wednesday/Central areas traffic: the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) got busy starting from early morning hours at 3:00 AM until the evening hours at 6:00 PM. Most trips at this period flowed from west and east areas to the central area. After that, the traffic got slower for the rest of the day.

Wednesday/Porto city park: the traffic in the city park started to get busy in the evening hours at 7:00 PM. Most trips flowed from east and central areas to the west area where the park is located. The traffic in the park remained busy until the midnight hour (00:00 AM). The flow of trips started to take the opposite direction at 10:00 PM where trips headed from west areas to east and central areas.

Wednesday/Bonfim Taxi Stop: there was a high traffic in the Bonfim east area, it started to get busy from the morning hours at 07:00 AM to the afternoon at 4:00 PM. It seems that trips at this period usually flow to the north areas via Dragon's stadium circle as shown in figure 35.

Wednesday/Ponte Arrabida bridge: in the morning at 11:00 AM, some trips flowed from Porto city to Gaia via this bridge.

Thursday/Airport trips: the period where we found a high traffic was the midnight hours (11:00 PM–02:00 AM). There was a high flow of trips heading from the south area to the airport during this period.

Thursday/Central areas traffic: the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) got busy starting from early morning hours at 3:00 AM until the

evening hours at 6:00 PM. Most trips at this period flowed from west and east areas to the central area. After that, the traffic got slower for the rest of the day.

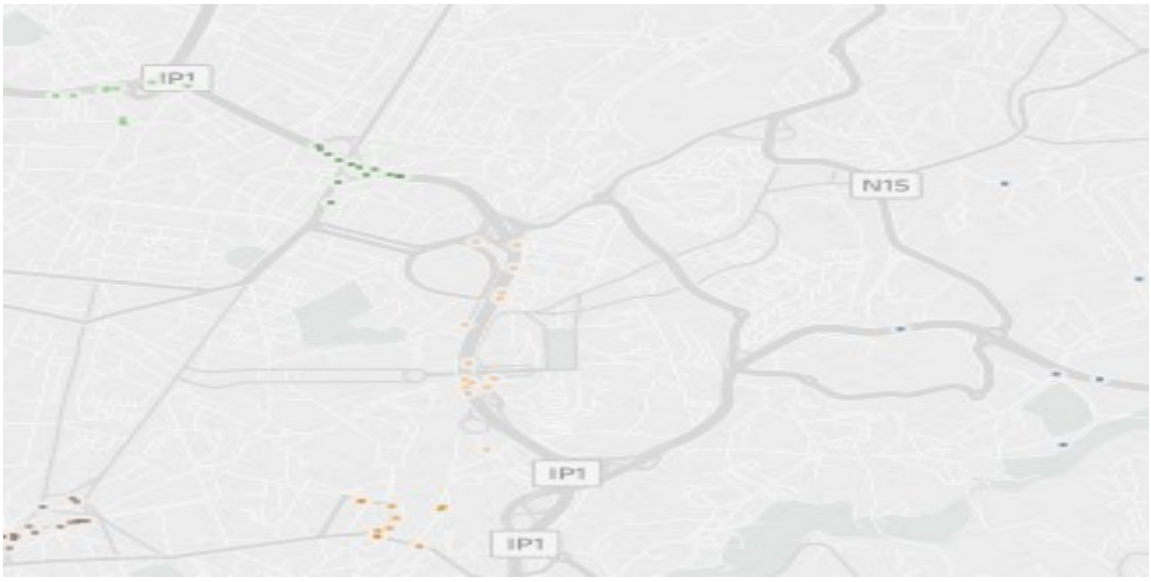

Figure 35. High traffic heading north via on Wednesday at 11:00 AM

Thursday/Porto city park: the traffic in the city park started to get busy in the evening hours at 7:00 PM. Most trips flowed from east areas to the west area where the park is located. The traffic in the park remained busy until the midnight hours. The flow of trips started to take the opposite direction at 10:00 PM where trips headed from west areas to east and central areas.

Thursday/Ponte Arrabida bridge: we found high traffic on this bridge in two main periods, one period in the morning (10:00 AM – 11:00 AM), and one period in the afternoon at 3:00 PM. Trips in these periods heavily travelled from Porto city to Gaia area via this bridge.

Thursday/Luis I Bridge traffic: in the afternoon at 4:00 PM, we found high traffic of trips flowing from Porto city to Gaia via this bridge.

Friday/Airport trips: the period where we found a high traffic was the midnight hours (11:00 PM–02:00 AM). There was a high flow of trips heading from the south area to the airport during this period.

Friday/Central areas traffic: the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) got busy starting from early morning hours at 3:00 AM until the afternoon hours at 3:00 PM. It seems that people are leaving their work at an earlier time on this day. Most trips at the busy period flowed from west and east areas to the central area. After that, the traffic got slower for the rest of the day.

Friday/Porto city park: the traffic in the city park started to get busy in the evening hours at 7:00 PM. Most trips flowed from east areas to the west area where the park is located. The traffic in the park remained busy until the midnight hours (02:00 AM). The flow of trips started to take the opposite direction at 09:00 PM where trips headed from west areas to east and central areas.

Friday/Matosinhos trips: in the afternoon hours (3:00 PM–4:00 PM), traffic started to get busy in this area. Trips started to flow from center areas to the seafood restaurants and barbeque places as shown in figure 36. After that, the traffic became slow until it is the night. At the night hours (9:00 PM-10:00 PM), the traffic got busy again then it slowed for the rest of the day.

Friday/Ponte Arrabida bridge: we found high traffic on this bridge at 10:00 AM, trips at this time flowed from Porto city to Gaia via this bridge.

Friday/Luis I Bridge traffic: we found high traffic of trips flowing from Porto city to Gaia via this bridge after midnight at 2:00 AM.

Figure 36. High traffic in Matosinhos on Friday at 3:00 PM

Saturday/Airport trips: the period where we found a high traffic was the midnight and early morning hours (11:00 PM–04:00 AM). The airport got busier at the weekend compared with other days. There was a high flow of trips heading from the south area to the airport during this period.

Saturday/Central areas traffic: unlike other weekdays, the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) at weekend was slow at the morning hours. However, it got busy starting from afternoon hours at 3:00 PM until the midnight hours at 11:00 PM. Most trips at this busy period flowed from west and east areas to the central area.

Saturday/Porto city park: the traffic in the city park started to get busy in the evening hours at 8:00 PM. Most trips flowed from east areas to the west area where the park is located. The traffic in the park remained busy until the midnight hours. The flow of trips started to take the opposite direction at 10:00 PM where trips headed from west areas to

east and central areas. However, we can observe that trips were still flowing from east to the city park at the late time of the night (10:00 PM–11:00 PM), this is probably because of the weekend where people can stay outside for longer periods.

Saturday/Matosinhos trips: in the afternoon hours (3:00 PM–4:00 PM), traffic started to get busy in this area. Trips started to flow from center areas to the seafood restaurants and barbeque places. After that, the traffic became slow for the rest of the day.

Saturday/Gaia trips: we notice high traffic of trips heading from Porto city to Gaia started from night hours until midnight (8:00 PM–02:00 AM). The high traffic was distributed on both bridges, Luis I bridge, and Ponte Arrabida bridge as shown in figure 37.

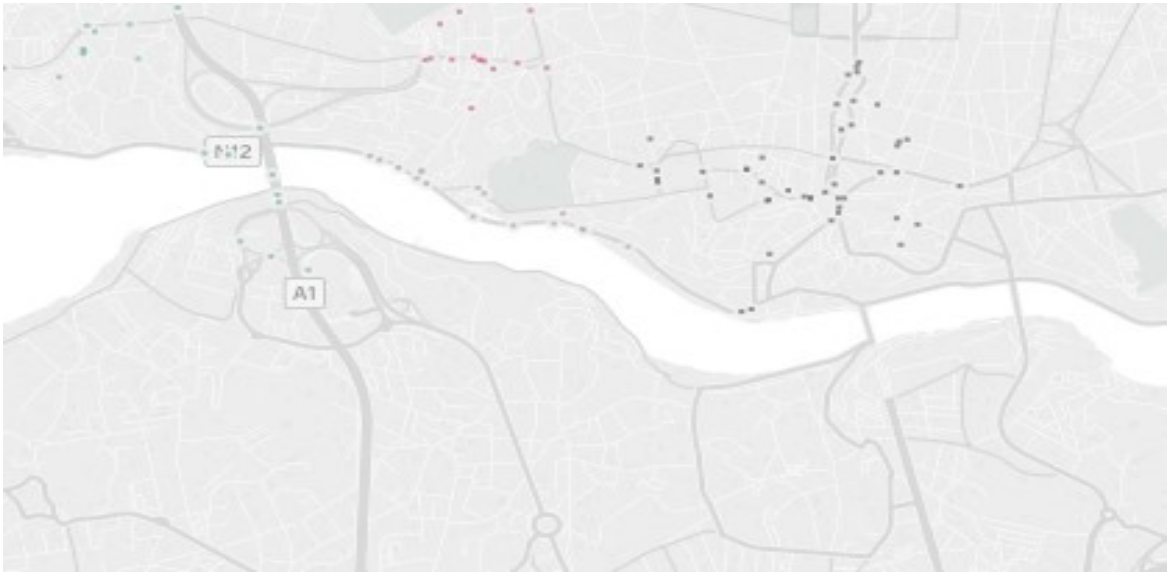

Figure 37. Trips heading to Gaia via Luis I bridge

Sunday/Airport trips: the period where we found a high traffic was the midnight and early morning hours (11:00 PM–03:00 AM). Also, there was a high traffic in the morning hours (07:00 AM – 08:00 AM). There was a high flow of trips heading from the south area to the airport during these two periods.

Sunday/Central areas traffic: unlike Saturday and other weekdays, the traffic in central areas (Boavista, Miragaia, Vitoria, Se, and Santo Ildefonso) at Sunday started to get high only for a short period in the morning hours (08:00 AM). Then it stayed slow until the midnight hours where it got busy again for a couple of hours (00:00 AM – 01:00 AM). It seems that people tend to spend fewer hours in the central area on Sunday. Most trips at busy periods flowed from west and east areas to the central area.

Sunday/Porto city park: the traffic in the city park started to get busy in the evening hours at 7:00 PM. Most trips flowed from east areas to the west area where the park is located. The traffic in the park remained busy until the midnight hours. The flow of trips started to take the opposite direction at 10:00 PM where trips headed from west areas to east areas.

Sunday/Gaia trips: we can notice that a high traffic of trips flowed from Porto city to Gaia in the midnight hours (11:00 PM–01:00 AM). The high traffic was distributed on both bridges, Luis I bridge and Ponte Arrabida bridge.

Sunday/Matosinhos trips: there was a high traffic in this area at 1:00 AM. Some trips flowed from center areas to the seafood restaurants and barbeque places at this hour.

C. Quantitative Analysis

In this analysis, we evaluate the quality of the HDBSCAN algorithm we used in the clustering process. In our evaluation, we applied Silhouette coefficient (Rousseeuw, 1987). In this method, we measure the average distance between a given point *p* and other points in its cluster, let us define the distance as b(*p*), where *p* is the given point. Then we measure the average distance between this point p and other points in the closest cluster and let us call it a(*p*), where *p* is the given point. Then we calculate the Silhouette score which is the

difference between b(p) and a(p), then we normalize the score by dividing the difference by the maximum value of both a and b as shown in the following equation:

$s(p) = \frac{b(p)-a(p)}{\max\{a(p),b(p)\}}$, where $-1 \leq s(p) \leq 1$

The Silhouette score s(p) is an indication of how much closer a point to its cluster. If the Silhouette score is in negative, that means it is far from its cluster. If it is zero, that means the point lies in between two clusters. If the score is positive, that means that the point is close to its cluster, thus, it is well clustered. Therefore, to get good clusters with high quality, we require b(p)≫a(p) as we want the Silhouette score to be closer to 1.

We created an R script to apply the Silhouette analysis on HDBSCAN (Ricardo J. G. B. Campello, 2013) clustering algorithm. In our script, we used two R packages, cluster package (Martin Maechler, 2018) and vegan package (Jari Oksanen, 2018). We considered the Euclidean distance criteria in our analysis, we calculated the average Euclidean distance among all points, then we called Silhouette function to evaluate the clusters based on the calculated distances. Finally, we plotted the Silhouette analysis results to show if the observations indicate a good clustering structure. We applied our R script on 35 CSV files, these files represented the taxi trips clustered based on weekdays vs 24 hours and were selected randomly.

Table 10 shows the Silhouette coefficients for 3 selected datasets. For the trips which started on Friday at 00:00 AM, 29 clusters were produced from HDBSCAN for 610 taxi trips. The average Silhouette score was 0.66 which means a reasonable structure has been found. Meanwhile, for trips which started on Monday at 05:00 AM, we can notice that 53 clusters were produced for 1004 trips. The average Silhouette score was 0.79 which means a strong structure has been found.

Table 10. Silhouette coefficient for 3 datasets

Dataset	Trips number	Clusters number	Silhouette Coefficient	Clustering Status
Friday Midnight trips	610	29	0.66	Good structure
Sunday 11 AM trips	511	30	0.81	Strong structure
Monday 5 AM trips	1004	53	0.79	Strong structure

4.2 Classification Evaluation

To evaluate our classifier models, we used the confusion matrix to summarize the performance of the naïve Bayes classifier. The confusion matrix is a tabular visualization of the classifier's accuracy. The table rows represent the actual values of traffic status, while the columns represent the predicted values of our models. We calculated the accuracy by dividing the percentage of correct predictions by the total predictions. For the Random Forest classifier, we used the confusion matrix with the number of predictions with respect to the size of the dataset. In addition, we added the error rate attribute which indicates how much accurate was the Random Forest classifier.

A. Naïve Bayes and Random Forest (Origin)

In table 11, we could observe the Naïve Bayes confusion matrix which listed the correct predicted percentage of traffic status. We could conclude from the confusion matrix that the prediction accuracy for our classifier was satisfying, the classifier was able to predict 77% of high traffic trips, 71% of low traffic trips, and 78% of medium traffic trips.

In terms of Random Forest classifier, the OOB estimation error rate for random forest classifier was 21.57%. Table 12 shows a confusion matrix of the numbers of correct predictions for traffic status values. As we could observe from the table; the highest

prediction accuracy was for high traffic status with an error rate of 12.7%. while the lowest prediction accuracy was for low traffic status with an error rate of 41.3%.

Table 11. NB Confusion Matrix for origin traffic status

Traffic status	High	Low	Medium
High	77	12	11
Low	20	71	9
Medium	11	11	78

Table 12. RF Confusion Matrix for origin traffic status

Traffic status	High	Low	Medium	Error rate
High	90353	3995	9196	0.1273951
Low	14460	34360	9780	0.4136519
Medium	12118	4329	71163	0.1877297

B. Naïve Bayes and Random Forest (Destination)

We created a naïve Bayes confusion matrix as shown in table 13, we noticed that the prediction accuracy for our classifier was 70% for high traffic status, 69% for low traffic status, and 79% for medium traffic status.

In terms of Random Forest classifier, the OOB estimation error rate for random forest classifier was 13.55%. Table 14 shows a confusion matrix of the numbers of correct predictions for traffic status values. As we could observe from the table; the highest prediction accuracy was for low traffic status with an error rate of 12.6%. while the lowest prediction accuracy was for high traffic status with an error rate of 14.5%.

Table 13. NB Confusion Matrix for destination traffic status

Traffic status	High	Low	Medium
High	70	24	6
Low	6	69	24
Medium	7	15	79

Table 14. RF Confusion Matrix for destination traffic status

Traffic status	High	Low	Medium	Error rate
High	36618	3998	2216	0.1450784
Low	3362	72166	7122	0.1268482
Medium	2664	7484	62490	0.1397065

C. Naïve Bayes and Random Forest (Route)

We created a naïve Bayes confusion matrix as shown in table 15, we noticed from the table that the prediction accuracy for the naïve Bayes classifier was 79% for high traffic status, 70% for low traffic status, and 76% for medium traffic status.

In terms of Random Forest classifier, the OOB estimation error rate for random forest classifier was 18.68%. Table 16 shows a confusion matrix of the numbers of correct predictions for traffic status values. As we noticed from the table; the highest prediction accuracy was for high traffic status with an error rate of 10%. Meanwhile, the lowest prediction accuracy was for low traffic status with an error rate of 38.3%. Overall, the accuracy of Naïve Bayes for predicting traffic status in different locations was around 70%. For the Random Forest, the error rate varied with a range between 12% and 41%

Table 15. NB Confusion Matrix for route traffic status

Traffic status	High	Low	Medium
High	79	11	10
Low	22	70	9
Medium	11	13	76

Table 16. RF Confusion Matrix for route traffic status

Traffic status	High	Low	Medium	Error rate
High	46472	1820	3728	0.1066513
Low	6696	17532	4196	0.3831973
Medium	5355	2002	39597	0.1566853

4.3 Results Verification

Our key findings provided efficient recommendations to enhance the traffic operations in Porto city. However, since we decided to analyze taxi trips in May, we should mention that the number of tourists is increasing significantly over years. Therefore, taxi trips movement patterns do not only reflect inhabitants, but they also include tourists who are moving around to discover the city. (Joao Gomes, 2014) found in their study that tourism in Porto continues to grow while observing a significant decrease in the domestic tourism. They explained some reasons like locals tend to avoid the peak summer time.

For our moving patterns, we were able to detect some interesting patterns over major districts such as the city park, Porto airport, Gaia, and Matosinhos. However, we were not able to detect moving patterns in smaller areas. For example, the moving patterns among central areas like Se, Vitoria, and Miragaia were hard to detect. We had a high traffic in

these areas for the most hours of the day which generated large clusters constantly. So, we assume that our HDBSCAN clustering model can be applied be detect some patterns among the main districts in Porto city. However, our sequential patterns can fill this gap by adding more insights into the frequent movements of taxi trips in central areas. Most sequential patterns were detected among central areas as we noticed from figure 18. We can refer our findings to (Ramos Delfinaa, 2016) study which found that most attractions in Porto were located in central areas, and they stated that the historical locations in these areas attracted tourists as these locations were proclaimed a World Heritage site by UNESCO in 1996. For predicting traffic status, our model had an accuracy of 70% which is reasonable. We could not get a higher accuracy because of the dataset behavior and the combination of domain and class attributes. For the time series analysis, we can rely on our model to detect anomalies visually by relying on both dendrograms and time series visualizations. We could detect an outlier among trips which started on Friday morning. However, we visualized a dataset of 17 trips to detect anomalies, using larger datasets can make this task harder.

Chapter 5: Implications

5.1 Transportation Recommender System

In this analysis, we were able to recognize taxi movement patterns over the main regions in Port city by identifying large clusters and their change over time. By using this knowledge, we can improve the driving experience in Porto. For example, based on our analysis, the traffic is usually high in the city park area in the evening hours, people finish their work in the evening and head to the west area.

We can use these patterns as a recommender system which advises drivers to avoid the west area at this time, or to estimate the travel time when taking a route towards the city park. Another example is the airport traffic, trips heading to the airport are more at midnight hours. This pattern can be used to provide the driver and the passenger with the fastest route to the airport if they need to catch up a plane in midnight hours. Furthermore, we can provide the driver with the time-dependent hotspots in Porto city. For example, if the taxi is vacant and it is afternoon, we can advise the driver to head to Matosinhos area since it is usually busy at this time as people are enjoying their seafood lunch.

We can use the interpreted patterns to study the behavior of taxi drivers. Relying on the driver's location and timeframe, we can advise the driver and provide him with the best strategy to take. Hence, if the location is far from the city hotspots at that moment, the best strategy is to wait in the same area if there is a coming pattern with a high traffic. Meanwhile, if there is no close pattern, we advise the driver to travel to a specific area with an existing high traffic. For example, if it is Sunday night and the driver is in Matosinhos area looking for passengers, we advise the driver to travel to the city park area since it is busy this time, people start to head back to the central areas. However, if it is Wednesday

evening and the driver is near the city park area, we advise him to wait there as this area will become busy soon.

In addition, we were able to identify the traffic status in different districts and timeframes. We built a classifier that was able to predict traffic status based on multiple factors (i.e. traffic origin, traffic destination, and traffic route) with an accuracy of around 70%. Moving one step forward to predict traffic status instead of visualizing historical data can provide useful guidelines for taxi drivers, passengers, and Porto transportation authorities. Additionally, these guidelines can be considered as an effective recommender system which can be utilized by all parties. In terms of taxi drivers, they can use route traffic status recommendations to avoid high traffic districts during their trip as they want to take the fastest route and to drop off their passenger with no delays. Meanwhile, they can rely on origin traffic status recommendations to seek high traffic districts when they are vacant. Usually, high traffic districts of trips initiations mean that there is a high demand for taxis in that area, thus, taxi drivers can more likely find passengers. In terms of passengers, the recommender system should be able to decrease their waiting time. Our recommender can predict the traffic status of trips destinations, this prediction can help passengers to identify taxi availability in a certain area. For example, if the traffic status of trips destination is high, this means that many taxis are ending their trips at that place, thus, passengers are more likely to find vacant taxis in this area and timeframe.

Overall, our recommender system was able to detect taxi movement patterns and identify traffic status in various regions and timeframes. This system can save time and effort for taxi drivers. Additionally, it can maximize their profit by providing them with traffic status

on a given route. Meanwhile, it can reduce the waiting time for passengers who are trying to find a vacant taxi.

5.2 Urban Computing

The trajectory data analysis uncovered busy regions and their interaction over different time intervals. Transportation management can use this knowledge to study the behavior of each region and how does each land type interact such as industrial, residential and institutional regions. By knowing the change in traffic for different land types, and analyzing points of interest in each region, urban planners and decision makers can improve the city traffic. They can add more public services and facilities in some areas which are usually busy.

Furthermore, this knowledge can be utilized to enhance the services provided to people. For example, sequential pattern mining technique we used previously can help authorities to identify the taxi trips flow over various regions. High flow areas can be a good indication of the functionality of certain areas (e.g. educational, business, industrial, etc.). For example, most patterns we discovered were in the central city areas, this could be a strong sign that most facilities such as universities, shopping centers, hospitals, and restaurants are not distributed over Porto city districts. Therefore, we can conclude that most places which attract people are in the central areas of Porto city, not in the north or west areas. This information can help transportation authorities to reduce traffic in the central areas by providing a reliable bus and subway services.

5.3 Pollution and Energy Consumption

By using extracted knowledge from our analysis, we can provide drivers with the fastest routes which can be significant for cost-efficient driving where gas is not wasted. Our

findings can be combined with other factors such as travel speed and vehicle acceleration to monitor the city's traffic conditions and enable time-dependent pollution alerts. These alerts can control the energy consumption and can support the use of alternate methods of transportation such as electric vehicles.

By using other types of sensors which can be fixed to collect information about air condition and its pollution level, we can combine our trajectory data analysis with the air pollution analysis to provide drivers with the fastest and cleanest routes at the same time. Furthermore, this approach can indicate if the drivers are environmentally friendly in the high traffic areas.

Furthermore, we can use our extracted patterns and combine them with the 311 noise complaints datasets. In this case, we can detect if the noise complaints are following a certain pattern by matching the complaint location and time with the traffic status at the that location and time. At the end, urban planners can find a correlation between the noise level and traffic status; thus, they can estimate the source of the noise if it is a noise by people or by vehicles.

5.4 Anomalies Detection

By applying trajectory data mining techniques such as time series analysis, we can detect outliers in the observations. An outlier trajectory can be a taxi trip which is significantly different from other trips in terms of distance metric (e.g. Euclidean distance). There are some techniques which detect outliers by decomposing time series into seasonal, trend and random data. Then they use the random data they extracted to detect anomalies, this process requires the time series data to be periodic with certain frequencies. Overall, time series visualization and anomaly detection can help authorities to interpret the unusual behavior

of taxi drivers and other moving objects. This behavior could be related to some issues like accidents, celebrations, protests, etc.

5.5 Human and Machine Intelligence Combination

In our study, we used visual analytics in all the techniques we applied. In classification, we created heatmaps to visualize the traffic status for each district, weekday, and timeframe. In clustering, we used Tableau maps to visualize taxi trips and their density. In sequential pattern mining, we used Porto city map to visualize taxi trips flow among various districts in the city. And in the time series analysis, we used dendrograms to visualize the trips time series clusters, and line charts to visualize trips flow over time and identify outliers.

Furthermore, we applied R shiny interactive visualizations to show how can the user choose different parameters and get a better understanding of sequential patterns and clustering of time series trips. Visualization is an essential technique for intelligent transportation systems, it can enhance the understanding of moving vehicles and traffic data. Visualization analytics can provide a useful interpretation of traffic data which can help the humans and decision makers in accident monitoring, route planning, and traffic jams reduction.

5.6 Business Potentials

Extracted patterns and trips movement behavior can be beneficial for business investors. By identifying busy regions in the city, investors can utilize the semantic description for each busy region to know different commercial places. Trajectory data analysis can help investors in identifying popular areas where they can start a successful retail business. Additionally, we can analyze the number of visits to a certain location. A high number of visits to the same place means that it is popular. Therefore, business owners can use outdoor

advertising posters at these locations to market their products. For example, based on our analysis, the chance of success to open a retail store inside the Norte shopping center is very high. This shopping center had a high traffic for the whole day (05:00 AM–05:00 PM), which means it is one of the main attractions in Porto city.

Chapter 6: Limitations and Future Work

6.1 Clustering Improvement

Our trajectory data analysis was based on taxi trips dataset in Porto city. However, this dataset did not have an attribute that indicates if the taxi is occupied or vacant. the behavior of taxi drivers is different if they have a passenger or not. Furthermore, if the taxi is occupied, the driver will try to find the fastest route to reach the destination. Meanwhile, if the taxi is empty, the driver will take random routes without any guidance to search for a passenger. Analyzing a trajectory dataset with taxi occupancy index can help us to extract more patterns and understand how taxi driver behave in each case.

A good approach in trajectory data analysis is to utilize the semantic description included in some points of interest. This description can provide more details about the nature of each place. Applying semantic trajectory analysis on datasets with enough description of points of interest is more efficient. Unlike raw trajectory data analysis where main neighborhoods are studied, semantic trajectory analysis can provide more information about the features of the city which can enhance the extracted knowledge. In our analysis, we could not recognize the nature of the moving points in a trajectory. However, if we had the semantic description, we could have a better understanding of the movement of trips and the reason behind taking a certain route.

Furthermore, we used trajectory data clustering to study the behavior of taxi trips movement. The HDBSCAN clustering algorithm was applied to extract clusters in different regions of Porto city. However, we used one parameter for this algorithm which is *MinPts*. This parameter was initialized by a mathematical equation and not by the domain expert. Meanwhile, in real life applications, to get the best relevant clusters, we can collaborate

with domain specialists. Such collaboration will provide relevant clustering process and helps in setting suitable values for clustering algorithm parameters.

Additionally, our clustering model was not able to detect trips on a higher granularity level. For example, trips movement in areas like Se and Miragaia were hard to detect due to the granularity level of our clustering which is one hour. A good approach is to apply a higher level of granularity like one minute. However, we need to visualize a small sample of taxi trips to be able to detect patterns in central areas of the city.

6.2 Enhancement of Classifier Accuracy

We applied the naïve Bayes classification on our dataset to predict the traffic status in each area and timeframe. However, the accuracy of our classifier was not very high despite selecting an uncorrelated set of attributes, the classifier accuracy was around 70%. This might be due to a few reasons; one reason is the discretization approach we applied. We applied unsupervised discretization on each attribute in our dataset, this type of discretization does not consider the values stored inside each attribute which could cause a missing in some data when converted into the nominal state. In our study, we divided each attribute into intervals based on the manual thresholds we defined. For example, we discretized the trips start time by dividing the attribute into 3 intervals (i.e. morning, afternoon, night) by applying a user-defined threshold.

Additionally, we did the same process for trip districts and traffic status attribute (i.e. low, medium, high). Meanwhile, supervised discretization recursively divides the class into several bins relying on the information stored inside that class. After that, it keeps choosing split points until it meets the stopping criterion. Applying the supervised discretization on our attributes could enhance the accuracy of our classifier.

Another possible reason is the distribution of the class values for the traffic status attribute. Therefore, if the discretization process generated imbalanced class values, this would affect the performance of the naïve Bayes classifier. We avoided this by manipulating the user-defined thresholds as shown in table 5 until we had a well-balanced dataset. However, previous studies indicated that using sampling techniques such as oversampling and under-sampling could reduce the misclassification and improve the classifier performance. Therefore, using these techniques to increase the size of minority class values or decrease the size of the majority class could lead to a balanced dataset, thus, could improve the accuracy of our classification.

6.3 Anomaly Detection

A critical task in trajectory data mining is to detect anomalies and outlier trajectory. In our study, we adopted the visual detection of anomaly taxi trips for a small subset of data. However, it might be a good approach to use longer trips with more longitudinal coordinates. Then we could apply some time series techniques such as smoothing to detect if the time series data has a periodic trend. After that, we can apply some anomaly detection techniques to decompose time series data and extract seasonal and random data. By following this approach, we could detect outlier taxi trips even when we analyze a large set of taxi trips.

Furthermore, detecting outliers visually can be a hard task when we have many trips. Using some packages to filter the time series data and extract anomalies will be more reliable.

In time series analysis, forecasting time series is an essential task to predict the future trends of taxi trips. Applying some techniques like Autoregressive integrated moving average (ARIMA) can be beneficial for predicting potential behavior for taxi trips. However,

applying ARIMA requires a seasonal trend of data, which means it can be applied on longer trips duration with different trends.

6.4 Use of Interactive Visualization

We plan to build a comprehensive interactive visualization application. By using R shiny, we can create many slides of presentation where each slide represents a separate trajectory mining technique such as clustering, classification, and sequential pattern mining. We plan to create dynamic slides which can be fed by any online trajectory mining datasets. This application can help users to understand the mechanism of different techniques when they have the chance to interact with various visualizations.

Chapter 7: Conclusions

Applying trajectory data mining techniques on Porto taxi trips dataset was fruitful. We presented a detailed analysis of spatiotemporal trajectories of taxi trips. We performed a descriptive analysis to understand the nature of the dataset. We noticed that the highest number of trips occurred in May and October. Additionally, during the day hours, the number of trips was high in the early morning hours and then dropped in the evening and night hours. After that we decided to analyze taxi trips for May, we grouped May taxi trips with the same start time into one CSV file, we excluded trips with a duration of one hour or more. We performed our analysis based on two factors, 24-hours and weekdays vs 24-hours. We made some experiments to cluster taxi trips stored in CSV files, then we excluded K-means and DBSCAN algorithms. We selected HDBSCAN clustering algorithm which only required one parameter. We took each CSV file, then we applied HDBSCAN clustering algorithm on each group of moving points at a specific timeframe. For each clustering process, we added a new column in each CSV file, this column stored the clusters outcome. For some CSV files, we executed HDBSCAN algorithm for 240 timeframes, thus, 240 new columns were added to the CSV file after clustering.

We used Tableau desktop to visualize each clustered CSV file. We plotted each pair of coordinates (longitude and latitude) on the Porto city map, then we added the colored clusters. To evaluate our clustering, we performed a quantitative analysis using Silhouette coefficient. Silhouette score index proved that our clustering structure was strong and reasonable. Furthermore, we performed an activity index analysis on CSV files. We studied the number of clusters produced in each CSV file to determine the activity on that day and time. Results indicated that on weekdays, people tend to be more active in early morning

since they are heading to work and school. Meanwhile, on weekends, people tend to be more active in the evening and night hours were they like to enjoy outdoor city attractions. After that, we used a qualitative analysis to extract patterns from the taxi movements. We were able to identify the main attractions in Porto, then we discovered some interesting patterns in the city. In the central area, it was busy in the daytime during weekdays. However, on weekends it was busy in the evening and at night. In Porto city park, it got busy in the evening when people finished their work, they liked to spend a couple of hours in this park. For Matosinhos area, it has a lot of seafood restaurants, it got busy in two periods, the lunchtime and the dinner time. For Norte shopping center, people tend to spend time at this place in the daytime only. but for the night, they tend to spend more time in outdoor places.

For the flow of the trips, it was changing over time. In the early morning hours, trips were flowing from east and west areas to the central areas where most attractions and facilities can be found. In the afternoon, some trips flowed from the central area to Norte shopping center in the west. At the evening hours, trips started to flow from the central areas to Porto city park in the west. After that, trips flowed back from the city park to the central areas, while some trips flowed from west area to the airport in the midnight.

After that, we conducted the naïve Bayes and the random forest classification to predict the traffic status for any point on the trip route, on taxi trips initiations, and on taxi trips destinations. We performed classification on three CSV files, each file had four attributes including the class attribute which was traffic status. We took a subset of the original dataset which represented trips that started in May, then we extracted districts names corresponded to each longitudinal coordinate by using Google maps. Then we clustered

each group of moving points in each CSV file using HDBSCAN, we extracted the degree of membership for each trip point. We relied on the degree of membership in each point to categorize the class attribute into three values, high traffic, medium traffic, and low traffic. In terms of classification, the naïve Bayes classifier was able to predict the traffic status with an accuracy of around 70%, while the random forest classifier predicted the traffic status with an approximate error rate of 18%.

Furthermore, we applied sequential pattern mining on the taxi trips dataset. We took a subset of the original dataset which represented trips that started in May and had a duration of 5 minutes. We preprocessed the CSV file by converting its structure from a wide format into a long format. After that, we applied the SPADE technique with a support value of 0.04 to extract taxi trips patterns over several districts in Porto city. Interestingly, we found that most taxi trips patterns flew over the central districts of the city. In addition, some patterns had a sequence of 3 districts, while most patterns had sequences of two districts.

We performed a time series analysis. To have a clear visualization, we took two small subsets of the original dataset. These subsets represented two groups of trips that occurred in the same timeframe and started and ended in the same places. The first subset contained trips which started on Friday morning in Sao Nicolau and ended in Se. the second subset contained trips which started on Saturday night in Lordelo Do Ouro and ended in Massaleros. We measured the similarity of trips in each group by applying hierarchical clustering based on the Euclidean distance measure. Then we visualized each group of trips as time series lines to analyze the trend of taxi trips. We could find that most trips were taking the same trend and behavior. However, we could detect a few trips which had different behavior and we assumed that they are anomalies or outlier trajectories.

Overall, we had a good understanding of people's movements and attraction areas they liked to visit. Our findings can be counted as a recommender system for taxi trips in Porto city. Our recommender can provide useful guidelines for multiple parties such as taxi drivers, passengers, and transportation authorities in the city. It can help taxi drivers to save time and gas by identifying high traffic areas and avoiding traffic jams. Additionally, it can reduce passengers waiting time for a vacant taxi by providing them with high traffic drop off locations with high taxi availability. In terms of transportation authorities, it can provide them with useful knowledge about the flow of trips, traffic status, and taxi drivers behavior. This knowledge helps authorities to enhance the road network services and improve the city's infrastructure by reducing traffic jams and establishing new facilities.

Bibliography

Aghabozorgi Saeed, S. S. (2015). Time-series clustering – A decade review. *Information Systems*, 16 - 38. doi:https://dx.doi.org/10.1016/j.is.2015.04.007

Ahmad Ashari, I. P. (2013). Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 7.

Andy Liaw, M. W. (2002). Classification and Regression by randomForest. 18-22. R News. Retrieved from https://www.researchgate.net/profile/Andy_Liaw/publication/228451484_Classification_and_Regression_by_RandomForest/links/53fb24cc0cf20a45497047ab/Classification-and-Regression-by-RandomForest.pdf

Bermingham Luke, L. I. (2014). Spatio-temporal Sequential Pattern Mining for Tourism Sciences. *Procedia Computer Science*, 379 - 389. doi:https://dx.doi.org/10.1016/j.procs.2014.05.034

Campello Ricardo J. G. B, M. D. (2013). Density-based clustering based on hierarchical density estimates. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 160 - 172). Gold Coast, Australia: Springer. doi:https://dx.doi.org/10.1007/978-3-642-37456-2_14

Chanda Ashis Kumar, A. C. (2017). A new framework for mining weighted periodic patterns in time series databases. *Expert Systems With Applications*, 207 - 224. doi:https://dx.doi.org/10.1016/j.eswa.2017.02.028

Chen Ling, C. G. (2010). A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 657 - 676. doi:https://dx.doi.org/10.1016/j.pmcj.2010.08.004

Christian Buchta, M. H. (2018). arulesSequences: Mining Frequent Sequences. Retrieved from https://CRAN.R-project.org/package=arulesSequences

Chujai Pasapitch, K. N. (2013). Time series analysis of household electric consumption with ARIMA and ARMA models. *Lecture Notes in Engineering and Computer Science* (pp. 295 - 300). Hong Kong: International MultiConference of Engineers and Computer Scientists.

D Kahle, H. W. (2013). ggmap: Spatial Visualization with ggplot2. *R Journal*, 144-160. Retrieved from https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

David Meyer, E. D.-C.-C. (2018). Misc Functions of the Department of Statistics version 1.7-0. Retrieved from https://cran.r-project.org/package=e1071

Delimitware. (2018). Delimit version 4.0.0. Retrieved from www.delimitware.com

Dheeru Dua, K. T. (2017). {UCI} Machine Learning Repository. *{UCI} Machine Learning Repository*. CA, California, USA: University of California, Irvine, School of Information and Computer Sciences. Retrieved from UCI Machine Learning Repository: http://archive.ics.uci.edu/ml

Dongzhi Zhang, L. K. (2015). Periodic Pattern Mining for Spatio-Temporal Trajectories: A Survey. *2015 10th International Conference on Intelligent Systems and*

Knowledge Engineering (ISKE) (pp. 306 - 313). Taipei, Taiwan: IEEE. doi:https://dx.doi.org/10.1109/ISKE.2015.92

Esling Philippe, A. C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 1 - 34. doi:https://dx.doi.org/10.1145/2379776.2379788

European Best Destinations. (2018). European Best Destinations. Brussels, Belgium. Retrieved from https://www.europeanbestdestinations.com/

Felipe Pinto da Silva, R. F. (2015). A Method to Detect and Classify Inconsistencies of Moving. *Journal of Information and Data Management*, 71-80.

Feng Mao, M. J. (2016). Mining spatiotemporal patterns of urban dwellers from taxi trajectory data. *Frontiers of Earth Science*, 205 - 221. doi:https://dx.doi.org/10.1007/s11707-015-0525-4

Ferreira Nivan, P. J. (2013). Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. *IEEE Transactions on Visualization and Computer Graphics, 19*(12), 2149 - 2158. doi:https://dx.doi.org/10.1109/TVCG.2013.226

Free Software Foundation, Mozilla Foundation. (n.d.). DB Browser for SQLite. Retrieved from http://sqlitebrowser.org/

Gang Pan, G. Q. (2013). Land-Use Classification Using Taxi GPS Traces. *IEEE Transactions on Intelligent Transportation Systems*, 113 - 123. doi:https://dx.doi.org/10.1109/TITS.2012.2209201

Gonzalez P.A, W. J. (2010). Automating mode detection for travel behaviour analysis by using global positioning systemsenabled mobile phones and neural networks. *IET*

Intelligent Transport Systems, 4(1), 37 - 49. doi:https://dx.doi.org/10.1049/iet-its.2009.0029

Google Inc. (2018). Google Maps. Porto, Portugal. Retrieved from https://www.google.ca/maps/place/porto+portugal

Guedes, A. (n.d.). *Travelling by taxi in Porto*. Retrieved from Local Porto: https://localporto.com/travelling-by-taxi-in-porto/

H. Xiong, L. C. (2017). A WEB-BASED PLATFORM FOR VISUALIZING SPATIOTEMPORAL DYNAMICS OF BIG TAXI DATA. *The International Archives of the Photogrammetry*, 1407 - 1412. doi:https://dx.doi.org/10.5194/isprs-archives-XLII-2-W7-1407-2017

Hahsler Michael, C. S. (2011). The arules R-package ecosystem: Analyzing interesting patterns from large transaction data sets. *Journal of machine learning research*, 2021 - 2025.

Hahsler Michael, G. B. (2005). Arules - A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 1 - 25. doi:https://dx.doi.org/10.18637/jss.v014.i15

Han Jiawei, K. M. (2011). *Data mining: concepts and techniques* (3rd ed ed.). Elsevier. doi:10.1016/C2009-0-61819-5

Hari Krishna Kanagala, V. J. (2016). A comparative study of K-Means, DBSCAN and OPTICS. *2016 International Conference on Computer Communication and Informatics, ICCCI 2016.* Coimbatore, India: IEEE.

Hartigan J. A, W. M. (1979). AS136 A K-means clustering algorithm. *Applied Statistics*, 90.

Hwang Ren-Hung, H. Y.-L.-T. (2015). An effective taxi recommender system based on a spatio-temporal factor analysis model. *Information Sciences*, 28 - 40. doi:https://dx.doi.org/10.1016/j.ins.2015.03.068

Irrevaldy, S. G. (2017). Spatio-temporal mining to identify potential traffic congestion based on transportation mode. *2017 International Conference on Data and Software Engineering (ICoDSE)* (pp. 1 - 6). Palembang, Indonesia: IEEE. doi:https://dx.doi.org/10.1109/ICODSE.2017.8285857

Jari Oksanen, F. G. (2018). vegan: Community Ecology Package. Retrieved from https://CRAN.R-project.org/package=vegan

Jean Damascène Mazimpaka, S. T. (2015). Exploring the potential of combining taxi gps and flickr data for discovering functional regions. *Lecture Notes in Geoinformation and Cartography* (pp. 3-18). Elsevier B.V. doi:10.1007/978-3-319-16787-9_1

Jianqin Zhang, P. Q. (2015). A space-time visualization analysis method for taxi operation in Beijing. *Journal of Visual Languages and Computing*, 1 - 8. doi:https://dx.doi.org/10.1016/j.jvlc.2015.09.002

Joao Gomes, D. R. (2014). Holiday intentions of Portuguese residents: a profile analysis for the years 2010 to 2014. *Worldwide Hospitality and Tourism Themes*, 429-441.

Khashei Mehdi, B. M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing Journal*, 2664 - 2675. doi:https://dx.doi.org/10.1016/j.asoc.2010.10.015

Krumm, J. (2008). A markov model for driver turn prediction. *Society of automotive engineers (SAE)*. doi:https://dx.doi.org/10.4271/2008-01-0195

Kumar Vipin, F. J. (2014). Spatio-temporal Data Mining for Climate Data: Advances, Challenges, and Opportunities. In W. W. Chu, *Data Mining and Knowledge Discovery for Big Data* (pp. 83 - 116). Los Angeles, USA: Springer, Berlin, Heidelberg. doi:https://doi-org.proxy.library.carleton.ca/10.1007/978-3-642-40837-3

Lee Jae-Gil, K. M. (2015). Geospatial Big Data: Challenges and Opportunities. *Big Data Research*, 74-81. doi:https://dx.doi.org/10.1016/j.bdr.2015.01.003

Lei Bao, M. D. (2018). A distance-based trajectory outlier detection method on maritime traffic data. *2018 4th International Conference on Control, Automation and Robotics (ICCAR)* (pp. 340 - 343). Auckland, New Zealand: IEEE. doi:https://dx.doi.org/10.1109/ICCAR.2018.8384697

Lin Miao, H. W.-J. (2014). Mining GPS data for mobility patterns: A survey. *Pervasive and Mobile Computing*, 1-16. doi:10.1016/j.pmcj.2013.06.005

M Ester, H. K. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *2nd International Conference on Knowledge Discovery and Data Mining* (pp. 226–231). AAAI Press.

Martin Maechler, P. R. (2018). cluster: Cluster Analysis Basics and Extensions.

Mazimpaka Jean Damascène, T. S. (2016). Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, 61 - 99. doi:https://dx.doi.org/10.5311/JOSIS.2016.13.263

Michael Hahsler, C. B. (2018). arules: Mining Association Rules and Frequent Itemsets. Retrieved from https://CRAN.R-project.org/package=arules

Michael Hashler, M. P. (2018). Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms. Retrieved from https://cran.r-project.org/web/packages/dbscan/dbscan.pdf

Montero Pablo, V. J. (2014). TSclust: An R Package for Time Series Clustering. *Journal of Statistical Software*, 1 - 43. doi:https://dx.doi.org/10.18637/jss.v062.i01

Moreira-Matias Luis, G. J.-M. (2013). Predicting Taxi–Passenger Demand Using Streaming Data. *IEEE Transactions on Intelligent Transportation Systems*, 1393 - 1402. doi:https://dx.doi.org/10.1109/TITS.2013.2262376

Mujalli Randa Oqab, L. G. (2016). Bayes classifiers for imbalanced traffic accidents datasets. *Accident Analysis and Prevention*, 37 - 51. doi:https://dx.doi.org/10.1016/j.aap.2015.12.003

Nick Theresa, C. E. (2010). Classifying means of transportation using mobile sensor data. *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1 - 6). Barcelona, Spain: IEEE. doi:https://dx.doi.org/10.1109/IJCNN.2010.5596549

Oracle Corporation. (2018). Oracle Application Express (APEX) version 4.0.2.00.09 Oracle 11g. California, USA. Retrieved from https://apex.oracle.com

P. Fournier-Viger, J. C. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition (DSPR), 1*, 54–77.

Palma Andrey, B. V. (2008). A clustering-based approach for discovering interesting places in trajectories. *Proceedings of the 2008 ACM symposium on applied computing* (pp. 863 - 868). Fortaleza, Ceara, Brazil: ACM. doi:https://dx.doi.org/10.1145/1363686.1363886

Pereira, V. B. (2018). Urban Distinctions: Class, Culture and Sociability in the City of Porto. *International Journal of Urban and Regional Research*, 126-137. Retrieved from https://doi.org/10.1111/1468-2427.12532

PlanetWare Inc. (2018). 17 Top-Rated Tourist Attractions in Porto. Retrieved from https://www.planetware.com/tourist-attractions-/oporto-p-porto-porto.htm

R Foundation for Statistical Computing. (2018). R version 3.4.4.

Ramos Delfinaa, A. L. (2016). Tourism Porto and North of Portugal – Case Study Concerning Private Accommodation. *The 5th Jubilee International Scientific Congress.* Skopje, Republic of Macedonia: Researchgate.

Ray Y. Zhong, G. Q. (2015). A big data approach for logistics trajectory discovery from RFID-enabled production data. *International Journal of Production Economics*, 260-272. doi:doi: 10.1016/j.ijpe.2015.02.014

Ricardo J. G. B. Campello, D. M. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 7819*, pp. 160 - 172. Gold Coast, Australia: Springer Berlin Heidelberg. doi:https://dx.doi.org/10.1007/978-3-642-37456-2_14

Rodriguez-Galiano V.F, G. B.-O.-S. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93 - 104. doi:https://dx.doi.org/10.1016/j.isprsjprs.2011.11.002

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 53 - 65. doi:https://dx.doi.org/10.1016/0377-0427(87)90125-7

RStudio Inc. (2018). RStudio version 1.1.442. Retrieved from www.rstudio.com

Shang Jingbo, Z. Y. (2014). Inferring gas consumption and pollution emission of vehicles throughout a city. *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1027-1036). New York, New York, USA: ACM. doi:10.1145/2623330.2623653

Shen Ying, Z. L. (2015). Analysis and visualization for hot spot based route recommendation using short dated taxi GPS traces. *Information*, 134 - 151. doi:10.3390/info6020134

Sun Zhanbo, B. X. (2013). Vehicle classification using GPS data. *Transportation Research Part C*, 102 - 117. doi:https://dx.doi.org/10.1016/j.trc.2013.09.015

Tableau Software Inc. (2018). Tableau Desktop version 10.4.5. Retrieved from www.tableau.com

Takimoto Yoshiaki, S. K. (2017). Extraction of Frequent Patterns Based on Users' Interests from Semantic Trajectories with Photographs. *Proceedings of the 21st International Database Engineering & Applications Symposium* (pp. 219 - 227). Bristol, United Kingdom: ACM. doi:https://dx.doi.org/10.1145/3105831.3105870

Tang Luliang, Y. X. (2015). Lane-level road information mining from vehicle GPS trajectories based on Naïve Bayesian classification. *ISPRS International Journal of Geo-Information*, 2660 - 2680. doi:https://dx.doi.org/10.3390/ijgi4042660

Wang Xiaoyue, M. A. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 275 - 309. doi:https://dx.doi.org/10.1007/s10618-012-0250-5

Web2Business Inc. (2016). *Turinta Maps*. Retrieved from Turinta Maps: http://www.turinta.pt/

Wei Chen, F. G.-Y. (2015). A Survey of Traffic Data Visualization. *IEEE Transactions on Intelligent Transportation Systems, 16*(6), 2970 - 2984. doi:10.1109/TITS.2015.2436897

Wikimedia Foundation Inc. (2018). Porto. Retrieved from https://en.wikipedia.org

Winston Chang, J. C. (2018). shiny: Web Application Framework for R. Retrieved from https://CRAN.R-project.org/package=shiny

Xiaolong Li, G. P. (2012). Prediction of urban human mobility using large-scale taxi traces and its applications. *SP Higher Education Press*, 111-121. doi:10.1007/s11704-011-1192-6

Yang Yue, Y. Z. (2009). Mining time-dependent attractive areas and movement patterns from taxi trajectory data. *2009 17th International Conference on Geoinformatics* (pp. 1 - 6). Fairfax, VA, USA: IEEE. doi:https://dx.doi.org/10.1109/GEOINFORMATICS.2009.5293469

Ye Ding, S. L. (2013). HUNTS: A Trajectory Recommendation System for Effective and Efficient Hunting of Taxi Passengers. *2013 IEEE 14th International Conference on Mobile Data Management* (pp. 107 - 116). Milan, Italy: IEEE. doi:https://doi.org/10.1109/MDM.2013.21

You Dabin, S. H. (2017). Urban Mobility Model Generation with Public Taxi Transportation Data. *Proceedings of the 15th International Conference on Advances in Mobile Computing & Multimedia* (pp. 13 - 21). Salzburg, Austria: ACM. doi:https://dx.doi.org/10.1145/3151848.3151852

Yuanhang Hu, Y. Y. (2015). A Comprehensive Survey of Recommendation System Based on Taxi GPS Trajectory. *2015 International Conference on Service Science (ICSS)* (pp. 99 - 105). IEEE. doi:https://dx.doi.org/10.1109/ICSS.2015.31

Z. Feng, Y. Z. (2016). A Survey on Trajectory Data Mining: Techniques and Applications. *IEEE Access*, 2056-2067. doi:10.1109/ACCESS.2016.2553681

Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, 31 - 60. doi:https://dx.doi.org/10.1023/A:1007652502315

Zeileis Achim, G. G. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software*, 27.

Zhang Shu-kai, S. G.-y.-j.-w.-l. (2018). Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity. *Ocean Engineering*, 240 - 250. doi:https://dx.doi.org/10.1016/j.oceaneng.2018.02.060

Zhao Liangbin, S. G. (2017). An adaptive hierarchical clustering method for ship trajectory data based on DBSCAN algorithm. *IEEE 2nd International Conference on Big Data Analysis* (pp. 329 - 336). ICBDA 2017. doi:https://dx.doi.org/10.1109/ICBDA.2017.8078834

Zheng Yu, C. L. (2014). Urban Computing: Concepts, Methodologies, and Applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 1-55. doi:10.1145/2629592

Zheng Yu, L. F.-P. (2013). U-Air: when urban air quality inference meets big data. *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1436 - 1444). Chicago, Illinois, USA: ACM. doi:10.1145/2487575.2488188