

# **Shared Bike Scheme in New York: Predicting the end of trips with Supervised Machine Learning**

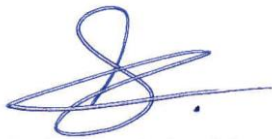
Dissertation

**Higher Diploma in Science in Data Analytics**

**X13122746 - Pedro Sacristan**

## Declaration

I hereby declare that this is entirely my own work and that it has not previously been submitted as an exercise for the award of a degree at this or any other College or University. I agree that the Library or other staff of National College of Ireland may lend or copy this dissertation on request.

A handwritten signature in blue ink, consisting of a large, stylized 'S' followed by a horizontal line and a small dot.

Signed by Pedro Sacristan

31/07/2014

## Summary

Share bike schemes are increasingly reliant on technology to improve and expand its services (AAAI, 2009). Being able to develop efficient, reliable and sustainable bike scheme systems will facilitate the accommodation of the rapidly increasing population in most of the cities around the world (IEEE, 2013).

This paper examines the possibility of predicting the destination of a trip by bike after the subscriber starts the journey. For that purpose some machine learning techniques have been applied to a dataset which contains almost one year of bike trips in the Share Bike Scheme of New York City.

Two supervised machine learning algorithms developed in a package of R have been used. “Naïve Bayes” was the first one but, because of the poor performance, a second one was used too, “Gradient Boosting Machine”.

Due to the lack of investment and means to carry out a complete analysis over the whole dataset a simplification was done by studying the performance of both algorithms in a sub-dataset of nine stations. It has been proven that the accuracy of the model is larger than was expected and that having the anonymized data the accuracy could reach a higher value.

The main contribution that this project makes to the subject of bike urban mobility is the potential for more precision than the time average of past events in predicting times for a bike in an empty station.

## Acknowledgements

I would like to thank Oisin Creaner for his help and advice in general and in particular about how to face the problem of lack of processing power.

Thanks to Dr. Jer Hayes who recommended some literature reviews which gave me an insight into what had been done before in this field. Also for his classes in Advanced Business Data Analysis and Data & Web Mining providing me with the knowledge of statistical analysis and Machine Learning Algorithms which have been used on this project.

Big thanks to my family who supported me during this challenging time especially my wife who has made it possible taking up some of my responsibilities as father of two daughters.

I would also like to thank Margarita, Fernando and Gerry, fellow classmates for their help in my approach to the project, suggestions and ideas.

Thanks as well to Lorraine for the English proof reading and her valuable point of view.

Thanks to my manager in Microsoft Gerry Hession for being so patient in the stressful moments I had during this period of hard working.

Last but not least, thanks to New York Bikes for making the data freely available.

## Table of Contents

1	Introduction.....	1
1.1	Domain Description.....	2
1.2	Motivation .....	2
1.3	Aims.....	3
1.4	Scope .....	3
1.5	Dissertation Structure .....	4
2	Literature Review.....	5
2.1	Urban Mobility .....	5
2.2	Predicting waiting times approaches.....	5
2.3	Statistical background.....	6
3	System and Datasets.....	9
3.1	Datasets.....	9
3.2	Data Requirements .....	9
3.3	Datasets Detailed Description .....	10
3.4	System Architecture .....	12
3.5	Technologies.....	14
3.5.1	Hardware .....	14
3.5.2	Software .....	15
3.6	Procedure .....	15
3.6.1	Data Acquisition .....	15
3.6.2	Data Cleaning .....	16
3.6.3	Data Preparing.....	19
3.6.4	Issues during these processes.....	22
4	Research Findings .....	24
4.1	An in-depth view of the dataset.....	24
4.2	Simplification .....	31
4.3	Analysis 1: Naïve Bayes.....	31
4.3.1	Research question: .....	32
4.3.2	Description.....	32
4.3.3	Method .....	32
4.3.4	Results .....	38
4.4	Analysis 2: Gradient Boosting Machine. ....	38

4.4.1	Research question: .....	39
4.4.2	Description.....	39
4.4.3	Method .....	40
4.4.4	Results .....	48
4.5	Analysis 3: Estimation with the User ID .....	48
4.5.1	Research question .....	48
4.5.2	Description.....	48
4.5.3	Method .....	49
4.5.4	Results .....	51
4.6	Analysis 4: Over-fitting and Under-fitting .....	52
4.6.1	Over-fitting .....	52
4.6.2	Under-fitting.....	53
4.7	Final Prototype.....	55
5	Conclusions .....	57
6	Further developments.....	59
6.1	Improving the current analysis .....	59
6.2	Develop a prototype to offer some companies .....	59
7	References .....	61
8	Literature.....	62
9	Appendix .....	63
9.1	Appendix 1 Online Repository OneDrive .....	63
9.2	Appendix 2 Study of Correlation between rain & trips .....	63
9.3	Appendix 3 Table of values comparing the accuracy in under-fitting.....	64
9.4	Appendix 4 Code of the programs (Images) .....	65
9.5	Appendix 5 Plot of the trips by day of the week.....	67
9.6	Appendix 6 Patterns of movements user 114 years old.....	68
9.7	Appendix 7 Complete results of the Confusion Matrix (R console) .....	70

## Table of Figures

FIGURE 1 OVERVIEW OF THE BIKES DATASET .....	11
FIGURE 2 OVERVIEW OF THE WEATHER DATASET .....	12
FIGURE 3 STRUCTURE OF ARCHITECTURE USED FOR THE PROJECT. ....	13
FIGURE 4 DIFFERENCE IN TIME UPLOADING DATA USING CSV OR SQL.....	14
FIGURE 5 MEMORY CONSUMING WHEN USING CSV FORMAT .....	14
FIGURE 6 VISUALIZATION OF THE MISSING VALUES .....	16
FIGURE 7 VISUALIZATION OF THE OUTLIERS .....	17
FIGURE 8 DISTRIBUTION OF THE TRIPS BY AGE OF THE USER .....	18
FIGURE 9 OUTLIERS IN THE WEATHER DATASET.....	19
FIGURE 10 CODE OF THE BULK INSERT.....	19
FIGURE 11 STRUCTURE OF THE TABLE TRIPS.....	20
FIGURE 12 SQL QUERY TO POPULATE THE NEW FIELDS .....	20
FIGURE 13 DISTRIBUTION MAP OF THE METEOROLOGICAL STATIONS.....	21
FIGURE 14 STRUCTURE OF THE TABLE WEATHER .....	22
FIGURE 15 DISTRIBUTION MAP OF THE BIKES STATION AROUND NEW YORK .....	24
FIGURE 16 TOTAL TRIPS BY MONTH.....	25
FIGURE 17 TOTAL TRIPS BY DAY DURING THE YEAR.....	26
FIGURE 18 TOTAL TRIPS BY HOUR THE SEVEN DAYS OF THE WEEK.....	26
FIGURE 19 TOTAL TRIPS BY STATION IN A MAP OF NEW YORK .....	27
FIGURE 20 TOTAL TRIPS STARTED IN "1 AVE & E 15 St" BY THE STATION OF ENDING.....	28
FIGURE 21 HISTOGRAM OF TRIP DURATION BETWEEN TWO STATIONS BY GENDER.....	29
FIGURE 22 TOTAL TRIPS BY AGE IN THREE DIFFERENT STATIONS.....	29
FIGURE 23 PERCENTAGE OF TRIPS BY GENDER IN THREE DIFFERENT STATIONS.....	30
FIGURE 24 LOCATION IN A MAP THE EIGHT CHOSEN STATIONS FOR THE ANALYSIS.....	31
FIGURE 25 SCREEN SHOOT OF THE PROGRAM NAIVE BAYES.....	33
FIGURE 26 MAIN CHARACTERISTICS OF THE DATASET AND THE ACCURACY OF THE MODEL.....	33
FIGURE 27 STANDARD DEVIATION OF THE ACCURACY AND KAPPA.....	34
FIGURE 28 SCREEN SHOT TO SHOW THE FILTER OF WORK DAYS.....	35
FIGURE 29 MAIN CHARACTERISTICS OF THE DATASET AND THE ACCURACY OF THE MODEL.....	35
FIGURE 30 STANDARD DEVIATION OF THE ACCURACY AND KAPPA .....	36
FIGURE 31 FILTER: DAYS OF THE WEEK AND ONLY FOR THE RUSH HOURS DURING THE MORNING.....	36
FIGURE 32 MAIN CHARACTERISTICS OF THE DATASET AND ACCURACY OF THE MODEL DURING 7 TO 9AM .....	37
FIGURE 33 STANDARD DEVIATION OF THE ACCURACY AND KAPPA.....	37
FIGURE 34 EVOLUTION OF THE ACCURACY DURING A WORK DAY WITH NAÏVE BAYES ALGORITHM.....	38
FIGURE 35 RATE USED TO SPLIT THE DATASET INTO TRAINING DATA AND TEST DATA .....	40
FIGURE 36 MAIN STEPS OF THE PROGRAM.....	41
FIGURE 37 VIEW OF THE DIFFERENT DATASET AND VARIABLES USED .....	41
FIGURE 38 SCREEN SHOT OF THE RESULTS .....	42
FIGURE 39 INDEPENDENT VARIABLE BY IMPORTANCE IN THE MODEL .....	43
FIGURE 40 EVOLUTION OF THE ACCURACY WITH THE NUMBER OF TREES.....	44
FIGURE 41 EVOLUTION OF THE KAPPA WITH THE NUMBER OF TREES .....	44
FIGURE 42 EVALUATION OF THE ACCURACY VS THE COMPLEXITY OF THE MODEL .....	45
FIGURE 43 EVOLUTION OF THE ACCURACY WITH THE NUMBER OF TREES.....	45
FIGURE 44 ACCURACY RESULT WITH THE MODEL .....	46
FIGURE 45 DENSITY PLOTS OF THE 200 BOOTSTRAP ESTIMATES THE ACCURACY AND KAPPA .....	46
FIGURE 46 ACCURACY PER HOURS DURING THE WORK DAYS .....	47

FIGURE 47 ACCURACY PER HOURS DURING THE WEEKEND DAYS .....	47
FIGURE 48 SCREEN SHOT OF THE RESULT .....	50
FIGURE 49 PLOT THE IMPORTANCE OF THE INDEPENDENT VARIABLE FOR THE MODEL.....	50
FIGURE 50 EVOLUTION OF THE ACCURACY WITH THE NUMBER OF TREES.....	51
FIGURE 51 ACCURACY R OF THE MODEL .....	51
FIGURE 52 CONFUSION MATRIX (1) TESTING THE MODEL.....	53
FIGURE 53 CONFUSION MATRIX (2) TESTING THE MODEL.....	53
FIGURE 54 IMPORTANCE OF THE INDEPENDENT VARIABLE FOR THE MODEL.....	54
FIGURE 55 ERROR FOR HAVING FEW ROWS .....	54
FIGURE 56 COMPARISON THE ACCURACY OF THE MODEL WITH 75% OF THE ROWS AND 100% (1) .....	54
FIGURE 57 COMPARISON THE ACCURACY OF THE MODEL WITH 75% OF THE ROWS AND 100% (2) .....	55

## Table of tables

TABLE 1 FIELDS OF THE BIKES DATASET .....	10
TABLE 2 FIELDS OF THE WEATHER DATASET .....	11
TABLE 3 DISTRIBUTION OF TRIPS BY GENDER IN THREE STATIONS.....	30
TABLE 4 ACCURACY IN THE SAME DATASET TRAINING THE MODEL WITH 75% OF ROWS AND 100%.....	65



# 1 Introduction

Shared mobility scheme programs are being developed all around the world as an answer to the demand from citizens of the big cities to have green, efficient and reliable public transportation (World Scientific Publishing Company, 2011).

From this type of community mobility schemes it is possible to extract an insight into the movements of citizens around cities and even into the cities activity. The actual system of collecting all the data provides the community with a digital footprint that can be studied to unveil the patterns of movements of people in a city over time (AAAI, 2009).

Bike schemes are one of the most successful share mobility programs due to their economical accessibility for all the people, their easily to use, the small space they take up in parking, the availability all around the cities and the reliability. For all these reasons and many more, having a handy share bike scheme is one of the indicators of a green city.

One of the principal measures of quality in a bike service is the availability of bikes to the users when they arrive at a station and the amount of information that the user has available to avoid waiting times. In most of the cities that have this public service of bikes an app to allow users find whether there is a free bike or dock in a station, is available.

The research focused on discovering by the use of Supervised Machine Learning algorithms whether it is possible to predict with a larger degree of accuracy the destiny of a bike after the user has started the journey or if it is just a question of luck to make correct predictions.

Findings in this dissertation could potentially be used as a tool for companies which have this service and want to improve the quality of the service.

## 1.1 Domain Description

The chosen domain for the present Project is the application of Machine Learning algorithms to find out patterns in the movement of bikes between stations and build a model to be able to predict the final station of bike in use.

This project analyses the trips by bicycle that have been made by the commuters of New York and which have been registered in their systems. The New York shared bike scheme was introduced in May 2013 and it has 330 stations and thousands of bikes. The bikes are available 24h/7, 365 days a year.

It is clear that the citizens of a city have daily routines and movement patterns that can be observed in the transport they use. As this economical and fast method of transport is becoming more and more popular it can give an insight into the activity of the city.

## 1.2 Motivation

Nowadays most of the companies try to exploit data to obtain a better understanding of their customers and their needs in order to provide them with a better service which in turn increase their sales. On the other hand, non-profit organizations or organizations providing a public service may not have the resources to invest money in this kind of research. Therefore, it is difficult for them to obtain the new technologies and skills to exploit the data stored which would enable them, for instance, to provide a better service to their customers, and ultimately the whole community. Because maintaining and enhancing a green share bike scheme mobility is important for the sustainability of cities and the comfort of those live in them, it is important that the scientific community collaborates with these organizations in an altruistic way to help them to increase the efficiency and reliability of their systems.

Different questions motivate the study of this shared mobility scheme. Some are about the correlation between the use of the bikes and the age, gender or type of user, while others are about the patterns of movements around the city and the correlation with the day of the week, time of the day and weather. Nevertheless, the main issue that this project wants to address is the provision of a model for

the accurate prediction of bikes availability at a specific station during the short periods of time when the station is empty of bikes.

### **1.3 Aims**

The goal of the project is to identify patterns in bikes journeys. The results of this analysis could be beneficial for strategic and planning purposes for companies which operate this kind of service.

Strategic: Understand better the movements that are predictable and consequently take action to provide a better service building more docks or new stations around the city.

Planning: Being able to predict the final stop of a bike after it leaves the station could be used not only to prevent people waiting unnecessarily, but also to move bikes from stations more effectively and offer publicity specific for business around the station where the user is going to finish their journey.

### **1.4 Scope**

Although in the Project Proposal the initial set of items declared in scope were more numerous, time constraints has meant that it was necessary to make some changes to this set of items and the final scope of the project focuses on the most challenging and ambitious items.

The general scope of the project is to predict how long a user will have to wait for a bike in a particular station when there are not any available. To solve this problem there are different approaches. Because some studies have used the logical approach of the average time of previous days at the same time on the same day of the week, as it is described in the literature review, the scope of this project is to focus on finding a model based on Machine Learning algorithms to predict the end station of the bikes which are in use at the moment another user is waiting.

This information will help the user to decide whether it is worth waiting or if it is better to go to another station. With this information will be included the accuracy of the model for that particular prediction.

## 1.5 Dissertation Structure

This paper is organized into four main sections as follows:

- Introduction, a general presentation of the problem to be addressed followed by an overview of the shared bike scheme of New York and highlighting its particular features.
- System and datasets section, which is concerned with the description of the tools which have been used as well as an overview of the datasets and the process carried out to prepare the data for the research.
- Research and Findings - this section relates to the analysis carried out during the research process. For each analysis the paper has different sections to describe the analysis to explain the method and the results.
- Conclusions highlights the key findings and describes the results of the research.

## **2 Literature Review**

This section summarize the papers that the researcher has read to study what has been done so far in the domain of this project and to learn about the algorithms which are going to be used.

### **2.1 Urban Mobility**

Observing and finding models for human movement in cities is a key factor to understanding the need for infrastructure and improving the efficiency of the ones the city already has (AAAI, 2009).

Nevertheless, little has changed since “Street Live Project” was written in 1980 observing and describing the usage of New York’s streets. One principal difficulty that urban planners, social scientists and virologists had to face is obtaining large amounts of data from real observations of human movements (Brockman et al., 2006). As city big transportation systems such as buses, underground, tolls and public utilities become digitized, the scientific community has other sources of information to study human movement around the city (Ratti D., 2006).

In the same way, the footprint that the new type of urban mobility scheme “Shared bicycling systems” leaves, has begun to be used for the purpose of studying not only the movements in cities but to infer cultural and geographical aspects of the city (AAAI, 2009).

### **2.2 Predicting waiting times approaches**

Community shared bicycle schemes have been under development over the last few years all around the world. Besides their interest as a new way to study public transportation and patterns of movements, there are a few studies which try to improve the accurate prediction of availability of both bikes and docks at a particular station (Word Scientific Publishing Company, 2011).

In 2009 Froehlich published a paper where studying bicycle usage over 13 weeks in the city of Barcelona would be able to infer cultural and geographical aspects of the city and predict future usage behaviour in new stations. In particular this paper makes a big contribution in:

- Demonstrating that the successful use of digital footprints of shared bicycling acts as a window to gain insights into the city dynamics and human behaviour.
- Studying the relationship between spatiotemporal patterns of bikes usage and city geography and behaviour.
- Exploring the patterns of usage including predictions of usage an analysis of how factors like day or time of the day affect this prediction (AAAI, 2009)

Bayesian Networks has been used in this paper to predict, for short and medium term (five minutes to two hours) ahead, the availability of bikes.

Kaltenbrunner extends the previous study by using Autoregressive Moving Average (ARMA) for time series models.

In 2010 Borgnat proposed signal processing methods to develop a model to predict the demand of bikes and docks in the Shared bike scheme in Lyon. (Word Scientific Publishing Company, 2011). Later on, in 2013 a team of IBM published a paper about the prediction of bike availability taking into account the exogenous factors such as weather or time of the day. They split the problem into two stages. In the first, they predict the availability of bikes with a Generalized Additive Model (GAM) and in the second the prediction of waiting times when there are no bikes available (IEEE, 2013).

No published paper or study has been found where the objective is to predict the end of the trips started at the time someone is waiting for a bike.

## 2.3 Statistical background

The statistical problem is to prove whether two supervised Machine Learning algorithms are able to predict in advance where a trip by bike in the city of New York is going to finish. These movements could seem random but are based on previous studies which have proven that the movements have temporal and spatiotemporal patterns which could help the algorithm to find a model of prediction.

The package “caret” (classification and regression training) contains numerous functions for developing predictive models in complex regression and

classification problems. It uses the huge set of models available in the open software R. Caret has been developed focusing on simplifying the model training and tuning. It includes as well methods for pre-processing data, visualization models, calculating variable importance and other important functionality that make it easier to face machine learning problems (R Project Org. 2014).

Naïve Bayes is the first of the ML algorithms used in this research. This algorithm was studied in Data & Web Mining and as an initial thought it was expected to be the better model for the problem faced. Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions. Although it is popular for text categorization and predicting spam mails it has been used for other complex classification problems performing quite well (R Project Org. 2014).

“Gbm” or Generalized Boosted Regression Models is a recent data mining technique that has proven considerable accuracy in predictions. It is based on gradient boosting ML technique for regression problems which produces a model typically decision trees (University of Washington, 2012).

Over-fitting happens when a model violates the principle of parsimony. This principle says that models and procedures must contain what is necessary but nothing more. There are several types of over-fitting:

- Using a model which is more flexible than it needs to be.
- The model includes not relevant features.
- The model catches even the noise of the dataset.
- The model only performs well in the training dataset.

Under-fitting is just the opposite problem with a model. It is not even able to capture the underlying trend between the predictors or independent variable and the dependent variable (University of Washington, 2012).

It has been proven that one good way to overcome the over-fitting problem is to use K-fold cross-validation. This method works as is described in the following steps:

- Split the dataset into K partitions.

- Designate one partition for testing the model and the rest of the partitions for training the model.
- Use the test dataset to assess the predictive accuracy of the model and the measure of predictive accuracy is mean squared error.
- Repeat the first three steps for the  $K-1$  partitions changing the test partition every time.
- Calculate the average of the mean squared error from all  $K$  validations.

Comparing all the average mean squared errors of the  $k$  models the system will pick up the one with the smallest as the best model.

If the dataset contains  $N$  rows and the  $K$  value chosen is equal to  $N$ , this type of cross validation receives the name: leave-one-out cross-validation

There are some advantages and disadvantages between a large and a small  $K$  value. In the case of a larger  $K$  results in:

- Less bias
- Higher variance
- Slower computation (University of Washington, 2012).

Although in this project both algorithms will use resampling (Cross-validation) with a different  $K$  value to evaluate the accuracy of the model and could be sufficient in most cases, sometimes resampling alone may be not enough. For this reason all the dataset used for predicting models will be split into two to have new data and test the accuracy of the model in data which has not been used for developing the model.



## 3 System and Datasets

### 3.1 Datasets

The datasets used for this research project comes from the combination of two open data sources:

In the first dataset a list of all the trips which have been made using the share bikes in the city of New York since it was launched in May of last year.

The second dataset contains several measurements of the main variables which define the weather of a region. It contains the data of more than one hundred stations around the city of New York.

Both datasets contains the same period of time with the purpose of being joined in a unique dataset which allows the researcher work in an easier and quicker way.

### 3.2 Data Requirements

To achieve the scope of the research a set of requirements was established and a long time was spent looking for the right dataset with the critical information.

#### **Data from the bike trips**

It was required to have at least one year of historical data in order to obtain sufficient data to carry out the research. Because the nature of the issue studied is seasonal it would have been great to have more than one year to appreciate some patterns and relations but in the end it was not possible.

The set of data has been made available for the owner with the only purpose of allowing the scientific community access to study whatever they are interested in.

It is required to have included in the dataset several features related to the information about the trip, for instance: trip duration, hour and date of the trip, initial and end station, id of the user, gender, age, type of user, bikes and docks available in the initial and end station.

### Data from the weather

It is required to have the data from the same period of time as the previous dataset in order to join both and be able to look for the correlation between the weather and the use of bikes.

Weather data is sometimes under a fee but because in the EE.UU. most of the public agencies are aware of the research that the scientific community is carrying out they share it for free.

Details: It is required that the dataset contains the temperature of the area, the measure of rain, snow, wind and whether it was foggy. It would be great to have this data each hour to be able to appreciate a high tide correlation.

### 3.3 Datasets Detailed Description

Although the process adapting the data to the needs of the project has been done in the procedure section, the original datasets are described in this section to have a first overview of the raw data.

The first dataset used for the purpose of this project was provided by the “NYC Bike Share LLC” a wholly-owned subsidiary of Alta Bicycle Share which operates large-scale bikes share systems. The dataset is available on the company website and it has been divided into a file per month. As the scheme was launched in May of last year there are only eleven months available so far today.

Scheme	Citi bike
City	New York
Country	United States of America
Owner	Alta Bicycle Share
Start data Collected	01/06/2013 00:00
End data Collected	31/05/2014 23:59
Frequency	At any time there is a change
Number of rows	7,518,335
Original size	1.86 Gb
Data Format	CSV
URL Source	<a href="http://www.citibikenyc.com/system-data">http://www.citibikenyc.com/system-data</a>

Table 1 Fields of the Bikes dataset

According to the specifications of the owner of the data all movements of bikes made by the organization has been removed. As well, all trips with a duration

shorter than 60 seconds have been removed as they are likely to be attempts to park the bike. Nevertheless, some techniques, which are explained later, have been applied to detect outliers and remove them. The table below contains an overview of the row data and the name of the fields.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	tripduration	starttime	stoptime	start station id	start station name	start station latitude	start station longitude	end station id	end station name	end station latitude	end station longitude	bikeid	usertype	birthyear	gender
2	362	01/02/2014 00:00	01/02/2014 00:06	234	Washington Square E	40.7304333	-73.997214	265	Stanton St & Chryslr St	40.722	-73.9947535	21101	Subscriber	1931	1
3	372	01/02/2014 00:00	01/02/2014 00:06	285	Broadway & E 14 St	40.73454567	-73.9974142	439	E 4 St & 2 Ave	40.726	-73.98978041	65456	Subscriber	1979	2
4	591	01/02/2014 00:00	01/02/2014 00:10	247	Perry St & Bleecker St	40.7353538	-74.00463091	251	Mont St & Prince St	40.723	-73.99460012	16281	Subscriber	1948	2
5	583	01/02/2014 00:00	01/02/2014 00:10	357	E 11 St & Broadway	40.7326787	-73.9958043	284	Greenwich Ave & 8 Ave	40.739	-74.00263761	17400	Subscriber	1981	1
6	223	01/02/2014 00:00	01/02/2014 00:04	401	Allen St & Rivington St	40.7203976	-73.98978025	439	E 4 St & 2 Ave	40.726	-73.98978041	13341	Subscriber	1990	1
7	541	01/02/2014 00:00	01/02/2014 00:09	452	Warren St & Church St	40.74173993	-74.00318627	331	Pike St & Monroe St	40.712	-73.99393043	18674	Subscriber	1930	1
8	354	01/02/2014 00:01	01/02/2014 00:06	325	E 19 St & 3 Ave	40.73624527	-73.98473765	439	E 4 St & 2 Ave	40.726	-73.98978041	16375	Subscriber	1931	1
9	916	01/02/2014 00:01	01/02/2014 00:16	354	Emerson Pl & Myrtle Av	40.69363137	-73.96223558	335	Bond St & Schermerhorn	40.688	-73.98410637	16020	Subscriber	1978	1
10	277	01/02/2014 00:01	01/02/2014 00:06	375	Mercer St & Bleecker St	40.72679454	-73.98965094	369	Washington Pl & 6 Ave	40.732	-74.00026394	18891	Subscriber	1944	1
11	439	01/02/2014 00:02	01/02/2014 00:09	285	Broadway & E 14 St	40.73454567	-73.9974142	247	Perry St & Bleecker St	40.726	-74.00463091	20875	Subscriber	1983	2
12	959	01/02/2014 00:02	01/02/2014 00:18	518	E 39 St & 2 Ave	40.74780373	-73.9734419	439	E 4 St & 2 Ave	40.726	-73.98978041	15263	Subscriber	1969	1
13	359	01/02/2014 00:02	01/02/2014 00:08	501	FDR Drive & E 35 St	40.744219	-73.97121214	487	E 20 St & FDR Drive	40.733	-73.97573881	19377	Subscriber	1986	1
14	1040	01/02/2014 00:02	01/02/2014 00:19	388	W 26 St & 10 Ave	40.74971775	-74.00295035	336	Sullivan St & Washington	40.73	-73.99060695	17271	Subscriber	1981	1
15	477	01/02/2014 00:02	01/02/2014 00:10	518	E 39 St & 2 Ave	40.74780373	-73.9734419	538	2 Ave & E 31 St	40.743	-73.97706058	19368	Subscriber	1930	1
16	707	01/02/2014 00:02	01/02/2014 00:14	257	Lispenard St & Broadw	40.71933226	-74.00247214	345	W 13 St & 8 Ave	40.736	-73.9704374	17757	Subscriber	1962	1
17	343	01/02/2014 00:03	01/02/2014 00:08	477	W 41 St & 8 Ave	40.75640548	-73.9900262	493	W 45 St & 8 Ave	40.757	-73.98291153	18734	Subscriber	1965	1
18	813	01/02/2014 00:03	01/02/2014 00:16	317	E 6 St & Avenue B	40.72453734	-73.98185424	223	W 13 St & 7 Ave	40.738	-73.99994661	18003	Subscriber	1942	1
19	451	01/02/2014 00:03	01/02/2014 00:28	527	E 33 St & 1 Ave	40.74375566	-73.97434726	412	Forsyth St & Canal St	40.716	-73.99422366	17630	Subscriber	1986	1
20	292	01/02/2014 00:04	01/02/2014 00:06	504	1 Ave & E 15 St	40.7322853	-73.98185557	487	E 20 St & FDR Drive	40.733	-73.97573881	16115	Subscriber	1989	2
21	259	01/02/2014 00:05	01/02/2014 00:09	316	Fulton St & William St	40.70955358	-74.00653609	415	Pearl St & Hanover Square	40.705	-74.00326027	20162	Subscriber	1980	2
22	231	01/02/2014 00:05	01/02/2014 00:09	430	8 Ave & W 33 St	40.757551	-73.993334	512	W 29 St & 9 Ave	40.75	-73.98393279	17141	Subscriber	1930	1
23	458	01/02/2014 00:05	01/02/2014 00:13	518	E 39 St & 2 Ave	40.74780373	-73.9734419	326	E 11 St & 1 Ave	40.73	-73.98426726	20774	Subscriber	1957	1
24	297	01/02/2014 00:05	01/02/2014 00:10	450	W 49 St & 8 Ave	40.76272205	-73.98788205	478	9 Ave & W 45 St	40.76	-73.9812551	21025	Subscriber	1951	2
25	497	01/02/2014 00:06	01/02/2014 00:14	300	Shevchenko Pl & E 6 St	40.728145	-73.990214	174	E 25 St & 7 Ave	40.738	-73.97173862	14827	Subscriber	1932	1
26	378	01/02/2014 00:06	01/02/2014 00:12	474	5 Ave & E 29 St	40.7451877	-73.98683077	442	W 27 St & 7 Ave	40.747	-73.99395	20167	Subscriber	1971	2
27	695	01/02/2014 00:06	01/02/2014 00:18	430	8 Ave & W 33 St	40.757551	-73.993334	468	Broadway & W 55 St	40.765	-73.98192338	21122	Subscriber	1979	1
28	189	01/02/2014 00:07	01/02/2014 00:10	540	Lexington Ave & E 28 St	40.74472288	-73.98323528	507	E 25 St & 2 Ave	40.739	-73.97973776	14945	Subscriber	1990	2
29	298	01/02/2014 00:07	01/02/2014 00:11	347	W Houston St & Hudson	40.72873888	-74.00748842	346	Bank St & Hudson St	40.737	-74.00618026	16842	Subscriber	1982	1
30	892	01/02/2014 00:07	01/02/2014 00:22	439	Broadway & W 60 St	40.7695505	-73.981941	430	8 Ave & W 33 St	40.752	-73.993334	20799	Subscriber	1968	1
31	636	01/02/2014 00:08	01/02/2014 00:19	285	Broadway & E 14 St	40.73454567	-73.9974142	393	E 5 St & Avenue C	40.723	-73.97995466	18764	Subscriber	1984	1
32	372	01/02/2014 00:08	01/02/2014 00:14	403	E 2 St & 2 Ave	40.72502876	-73.98069656	349	Rivington St & Ridge St	40.719	-73.98323659	19507	Subscriber	1944	1
33	664	01/02/2014 00:08	01/02/2014 00:15	237	E 11 St & 2 Ave	40.73047319	-73.98672378	349	Rivington St & Ridge St	40.719	-73.98323659	17540	Customer	IN	0
34	213	01/02/2014 00:08	01/02/2014 00:12	148	Hudson St & Reade St	40.71825008	-74.0091059	329	Greenwich St & N Moore	40.72	-74.01020609	18790	Subscriber	1956	1
35	678	01/02/2014 00:09	01/02/2014 00:23	437	E 17 St & Broadway	40.73704384	-73.99003296	477	W 41 St & 8 Ave	40.756	-73.9900262	19897	Subscriber	1987	1

Figure 1 Overview of the Bikes dataset

The second dataset used for the purpose of the project was provided by the National Oceanic and Atmospheric Administration, an agency which belongs to the National Climate Data Center. To download this dataset the agency provides a web tool where you can select the area you are interested in and, the weather variables you want in your dataset and it provides you with a link in your email to download the data.

City	New York
Country	United States of America
Owner	NOAA
Start data Collected	01/06/2013
End data Collected	31/05/2014
Frequency	Once a day
Number of rows	19,927
Original size	10 Mb
Data Format	CSV
URL Source	<a href="http://www.ncdc.noaa.gov/cdo-web/datasets">http://www.ncdc.noaa.gov/cdo-web/datasets</a>

Table 2 Fields of the weather dataset

According to the owner, the dataset contains all the data and no process has been carried out to clean or remove data. The table below contains an overview of the row data and the name of the fields.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
	STATION	STATION_NAME	ELEVATION	LATITUDE	LONGITUDE	DATE	MDPR	Measurement Flag	Quality Flag	Source Flag	Time of Observation	DWPR	Measurement Flag	Quality Flag	Source Flag	Time of Observation	PRCP	Measurement Flag	Quality Flag	Source Flag	Time of Observation	SNWD	Measurement Flag	Quality Flag	Source Flag	Time of Observation	SNOW	Measurement Flag	Quality Flag	Source Flag	Time of Observation	TMAX	Measurement Flag
2	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130608	-9999				9999	-9999				9999	1636		N	9999	-9999				9999	-9999					9999	-9999	
3	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130611	-9999				9999	-9999				9999	485		N	9999	-9999				9999	-9999					9999	-9999	
4	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130614	-9999				9999	-9999				9999	318		N	9999	-9999				9999	-9999					9999	-9999	
5	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130722	-9999				9999	-9999				9999	0		N	9999	-9999				9999	-9999					9999	-9999	
6	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130723	-9999				9999	-9999				9999	127		N	9999	-9999				9999	-9999					9999	-9999	
7	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130724	-9999				9999	-9999				9999	3		N	9999	-9999				9999	-9999					9999	-9999	
8	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130725	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
9	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130726	-9999				9999	-9999				9999	28		N	9999	-9999				9999	-9999					9999	-9999	
10	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130727	-9999				9999	-9999				9999	3		N	9999	-9999				9999	-9999					9999	-9999	
11	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130728	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
12	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130729	-9999				9999	-9999				9999	33		N	9999	-9999				9999	-9999					9999	-9999	
13	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130730	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
14	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130731	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
15	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130801	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
16	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130802	-9999				9999	-9999				9999	127		N	9999	-9999				9999	-9999					9999	-9999	
17	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130803	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
18	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130804	-9999				9999	-9999				9999	3		N	9999	-9999				9999	-9999					9999	-9999	
19	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130805	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
20	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130806	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
21	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130807	-9999				9999	-9999				9999	0		N	9999	-9999				9999	0			N	9999	-9999		
22	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130808	-9999				9999	-9999				9999	145		N	9999	-9999				9999	-9999					9999	-9999	
23	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130809	-9999				9999	-9999				9999	8		N	9999	-9999				9999	-9999					9999	-9999	
24	GHEND-U	LEVITTOWN 0.2 E NY US	27.1	40.7245	-73.5086	20130810	-9999				9999	-9999				9999	0		N	9999	-9999				9999	-9999					9999	-9999	

Figure 2 Overview of the Weather dataset

### 3.4 System Architecture

Dealing with the dataset of a Big-data project is challenging because the tools that are used in normal projects do not work very well within big-data projects most of the time. It is easy to get lost without ideas about how to tackle the problem. In this project it was not possible to join all the files in just one because Excel is not able to hold seven million lines. It is possible to save all the lines in a CSV file but I do not have a program able to open that number of files but just to upload it to memory by RStudio. The problem that comes up with the seven million in only one csv file is that it is very difficult to move 1.86GB and RStudio has to upload to memory every time the program needs it. Because the lack of memory in the computers used and the need of memory for the processing steps another solution had to be found.

In the image below a diagram of the processing has been drawn to clarify how the project deals with the big dataset.

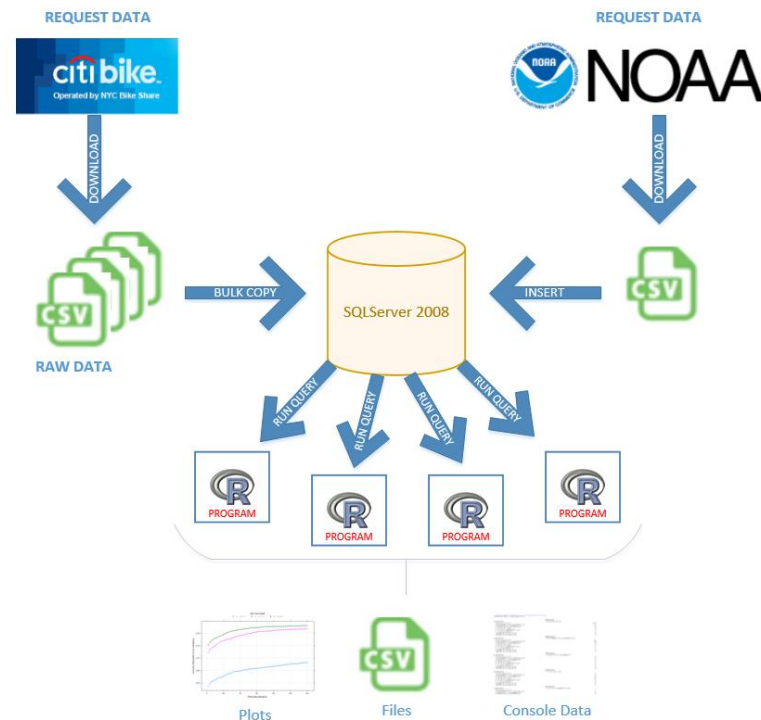


Figure 3 Structure of architecture used for the project.

Although the time that RStudio takes to upload the seven million lines in memory through an ODBC connection is greater than reading it directly from a CSV file, as it can be seen in the images below, it was decided to upload all the files into a SQLServer database to be more accessible at any time. Also it is easier to work with the ODBC connection ready at any time at anyplace no matter what “Working Directory” your RStudio is working on.

```

> Sys.time()
[1] "2014-07-09 18:19:04 BST"
> bikes7 = read.csv("July.csv", header=TRUE, sep=",")
> bikes8 = read.csv("August.csv", header=TRUE, sep=",")
> bikes9 = read.csv("September.csv", header=TRUE, sep=",")
> bikes10 = read.csv("October.csv", header=TRUE, sep=",")
> bikes11 = read.csv("November.csv", header=TRUE, sep=",")
> bikes12 = read.csv("December.csv", header=TRUE, sep=",")
> bikes1 = read.csv("January.csv", header=TRUE, sep=",")
> bikes2 = read.csv("February.csv", header=TRUE, sep=",")
> bikes3 = read.csv("March.csv", header=TRUE, sep=",")
> bikes4 = read.csv("April.csv", header=TRUE, sep=",")
> bikes5 = read.csv("May.csv", header=TRUE, sep=",")
> Sys.time()
[1] "2014-07-09 18:23:47 BST"

```

Total time: 4' 43"

Data	
bikes1	300400 obs. of 15 variables
bikes10	1037712 obs. of 15 variables
bikes11	675774 obs. of 15 variables
bikes12	443966 obs. of 15 variables
bikes2	224736 obs. of 15 variables
bikes3	439117 obs. of 15 variables
bikes4	670780 obs. of 15 variables
bikes5	866117 obs. of 15 variables
bikes7	843416 obs. of 15 variables
bikes8	1001958 obs. of 15 variables
bikes9	1034359 obs. of 15 variables

```

> library(RODBC)
> Sys.time()
[1] "2014-07-09 18:28:11 BST"
> # Connect to the SQLServer
> channel1 <- odbcconnect("SQLodbc")
> # Define the query with the four stations
> querystations = "SELECT * FROM TRIPSTEST"
> # Run the query
> bikes=sqlQuery(channel1,querystations)
> Sys.time()
[1] "2014-07-09 18:38:49 BST"

```

Total time: 10' 38"

Data	
bikes	7006181 obs. of 22 variables
values	
channel1	class 'RODBC' atomic [1:1] 1
querystations	"SELECT * FROM TRIPSTEST"

```

> Sys.time()
[1] "2014-07-09 18:39:25 BST"
> bikes = read.csv("BikesTotalStationExport.csv", header=TRUE, sep=",")
> Sys.time()
[1] "2014-07-09 18:45:27 BST"

```

Total time: 4' 2"

**Data**

bikes 7006181 obs. of 22 variables

Figure 4 Difference in time uploading data using CSV or SQL

The time that RStudio takes to upload the entire dataset from a csv file is shorter than through the ODBC connection but the amount of memory that the computer needs is lower because with the ODBC connection the system will bring only the data which at that particular moment is required.

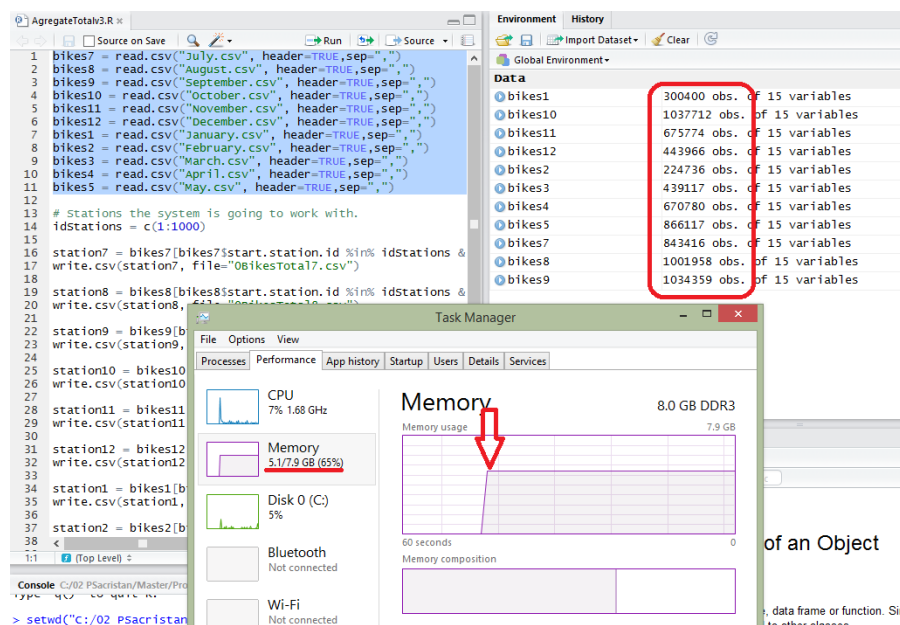


Figure 5 Memory consuming when using CSV format

## 3.5 Technologies

This section summarizes the different technologies which have been used for the project. Some of them have been chosen for the reason that they are the standard for the purpose of the action which was going to be carried out. However, others have been chosen because the researcher has the needed skills to use them.

### 3.5.1 Hardware

**Personal Computer** Asus Ultrabook S56C with 2.3 GHz Quad-core Intel Core-i7 processor, 750 GB of hard disk plus 24 GB of solid hard disc (SSD) and 8 G of memory RAM.

**The laptop of the company:** HP EliteBook 8540w with 1.6 GHz Intel Core i7 processor, 500 GB and 8 GB of memory RAM.

### 3.5.2 Software

**SQL Server 2008 R2 Express:** Is a reliable data management system which allows the creation of relational databases and delivers a rich set of features, data protection and free tools to connect.

**Microsoft Office 365:** It is similar to the standard Office 2012 but it includes new utilities like the geo-localization and the map utility which has been used for the purpose of this project.

**RStudio:** Version 0.98.507. It is an open software programming language and environment for computing, statistical and graphics purpose. It is widely used among the Scientifics and data analytics. It is free and a lot of packages have been developed altruistically. Some of the packages have been used in this project: (R Project Org., 2013)

- **RODBC:** To connect with a Microsoft SQL Server
- **caret:** Classification and Regression Training
- **gplot2:** An implementation of the Grammar of Graphics
- **e1071:** Miscellaneous Functions of the Department of Statistics

**OneDrive:** Document repository provided by Microsoft that lets the user access the information stored in it from anywhere at the same time that it provides backup automatically and the possibility to work offline with the same documents.

## 3.6 Procedure

In this section the different processes which have been carried out within the datasets have been summarized.

### 3.6.1 Data Acquisition

The NYC Bike share scheme operates in New York City. They share data of trips and use of bikes with the entire scientific community. To acquire the data just an internet connection and free space in the Hard Drive of your computer is needed.

No forms are required or permission from any authority. On the other hand, to obtain the weather dataset it was necessary to fill in a form and give an email. They want to know who downloads the data and your email. Note: My first thought was to do this project about Dublin Bikes. See the “Issues” sub-section below.

### 3.6.2 Data Cleaning

The first dataset used for this project has already been processed by the owner and some fields and rows have been removed for different reasons. By default the trips with a duration below 60 seconds in length have been removed because they are considered false starts or users trying to ensure that their bikes were secure. All the trips taken by the staff as they service or inspect the system have been removed as well. All the trips that are taken to/from any of their “test” stations have been removed too.

Although the quality of the dataset is very high some operations have been carried out. After a visual inspection of the dataset it was found that both the gender and the year of birth do not have a valid value in all the trips made by customers. This does not make sense for a 24 hours or week pass give any personal data. This is 8.70% of the total dataset, initially the thought was to remove them but finally they were kept as they are valid trips made by more temporary users. For those rows the “year of birth field” contains the value “\N” and the “gender” 0 (male has 1 and female 2).

```
> str(bikes7)
'data.frame': 7413680 obs. of 16 variables:
 $ x          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ tripduration : int  634 1547 178 1580 757 861 550 288 766 773 ...
 $ starttime    : Factor w/ 4290498 levels "01/07/2013 00:00",...: 1 1 2 2 2 2 2 3 3 3 ...
 $ stoptime     : Factor w/ 4304906 levels "01/07/2013 00:04",...: 5 12 1 13 8 9 6 3 9 9 ...
 $ start.station.id : int  164 388 293 531 382 511 293 224 432 173 ...
 $ start.station.name : Factor w/ 322 levels "1 Ave & E 15 St",...: 132 273 189 153 254 100 189 243 147 53 ...
 $ start.station.latitude : num  40.8 40.7 40.7 40.7 40.7 ...
 $ start.station.longitude: num  -74 -74 -74 -74 -74 ...
 $ end.station.id : int  504 459 237 499 410 454 394 376 336 479 ...
 $ end.station.name : Factor w/ 322 levels "1 Ave & E 15 St",...: 1 265 96 58 252 136 148 182 253 22 ...
 $ end.station.latitude : num  40.7 40.7 40.7 40.8 40.7 ...
 $ end.station.longitude : num  -74 -74 -74 -74 -74 ...
 $ bikeid       : int  16950 19816 14548 16063 19213 16223 16746 16062 17963 19365 ...
 $ usertype     : Factor w/ 2 levels "Customer","Subscriber": 1 1 2 1 2 2 1 2 2 2 ...
 $ birth_year   : Factor w/ 85 levels "\N","1899","1900",...: 1 1 67 1 73 75 1 72 67 76 ...
 $ gender       : int  0 0 2 0 1 1 0 2 2 1 ...
```

Figure 6 Visualization of the missing values

The final approach taken was to keep the value of 0 for the gender as a third value and replace the year of birth for 2012 to have an age of one year old instead



of zero. A program with two lines of code was developed to make the change in all dataset. This is available in the appendix and in the online repository.

After studying the data, one of the main problems of the dataset was to define what is considered an outlier. If we look at the field “trip duration” it is difficult to establish the minimum and the maximum time a trip can last. To decide what the correct maximum limit may be a graph was drawn to see if there was a clear number for that purpose. All the trips with a duration higher than 7200 seconds were rejected and they were not upload to SQL in a first process. It was assumed that a trip with a duration longer than two hours is out of our scope as that journey cannot be considered as a direct trip from one station to another.

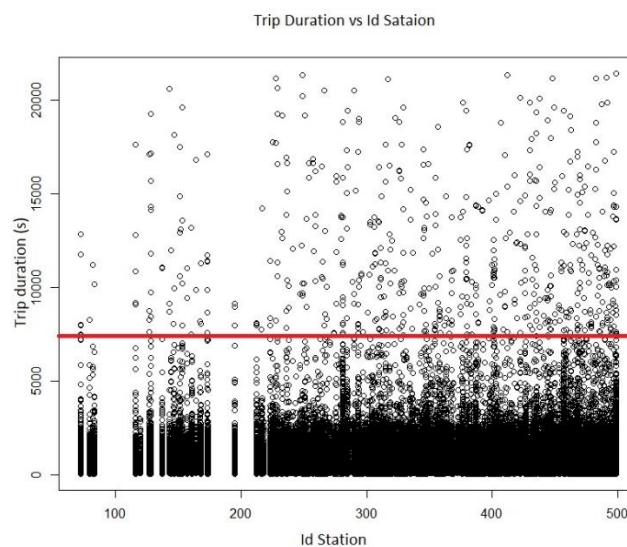


Figure 7 Visualization of the outliers

Five queries with different superior limits were run as well to determine what the percentage of rows out of scope would be. With a limit of 3600 seconds 42,000 trips would be out of scope which is only 0.6% of the total trips. We can consider those trips as noise of the dataset. Although they can be valid trips and not trips with problems, they could affect the accuracy of the Machine Learning model because this type of trips could be considered as unpredictable.

As it can be appreciated in the graph of trips distributed by the age of the subscriber there are some values that look very illogical. It is very unlikely that a person older than 90 years rides a bike. The most probable is that the user lied or made a mistaken when introducing his/her birthday. Nevertheless, after

studying the consequences it could have in the project, not action where carried out with this data. They could facilitate the algorithm to identify patterns more easily for them.

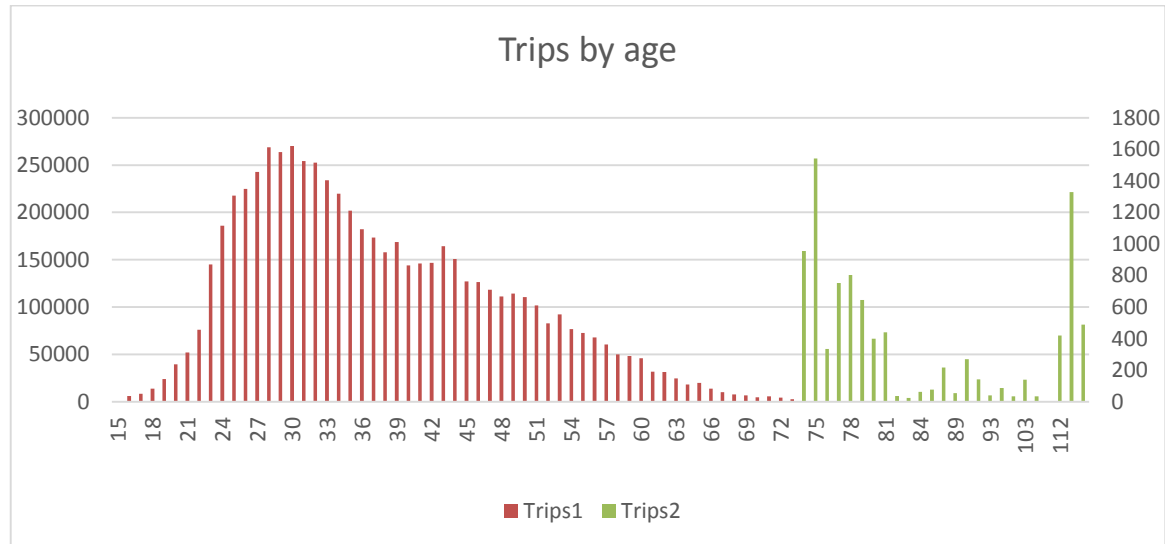


Figure 8 Distribution of the trips by age of the user

Note: The graph represents the number of trips by age. The red bars are related with the left axis whereas the green bars are related with the right axis. Although all are the same units the quantity on the red bars is huge and the green bars were invisible.

The second dataset has been processed by the owner to fill the fields that were missing with “-9999”. According to the data provider, the missing values are probably a consequence of, at that moment, the sensor not working on that particular day, there was a communication problem or it was in maintenance.

As it has already been explained the dataset has a lot of weather parameters that are not being used in our project so they were removed and kept the ones needed are kept.

After a visual inspection of the graph representing the maximum and minimum temperature, the rain and the snow, nothing abnormal was found as can be appreciated in the plot below.

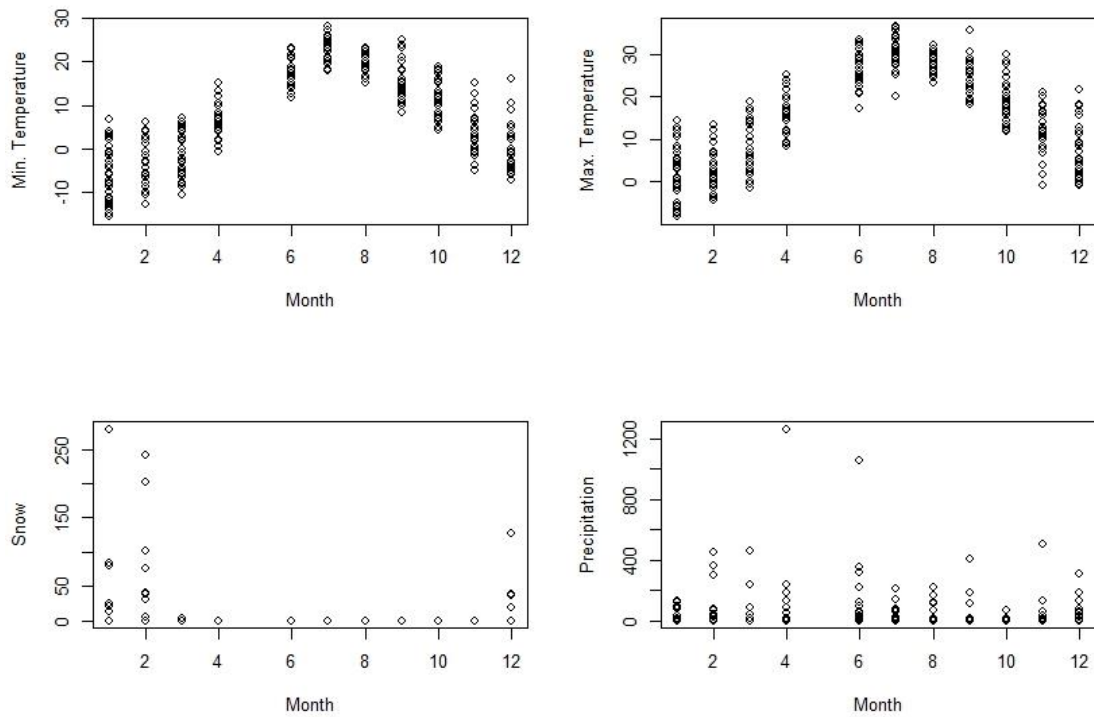


Figure 9 Outliers in the weather dataset

### 3.6.3 Data Preparing

Although all the operations that have been carried out to prepare the data could have been done with into the R programs which have been used for the Research purpose, it was decided to proceed before the program and have the data ready in the database. The first step was to upload the data into the SQL Server database. In the local SQL Server a database with the name NewyorkBikes was created and a table named trips with the same structure of the dataset was created too. In order to upload the seven million rows quickly, the bulk insert command of Microsoft SQL server was used:

```
/*This function insert the 7 million rows in the table Trips */
BULK INSERT Trips
FROM '\\V-PESAC-LAPTOP\ProjectData\TotalBikes.csv'
WITH (FIELDTERMINATOR = ',', ROWTERMINATOR = '\r')
```

Figure 10 Code of the Bulk Insert

It took just three minutes to upload the whole bunch of data, more than seven million rows. Some issues are described later on in this section.

Column Name	Data Type	Allow Nulls
idtrip	numeric(18, 0)	<input type="checkbox"/>
tripduration	numeric(18, 0)	<input checked="" type="checkbox"/>
starttime	nvarchar(16)	<input checked="" type="checkbox"/>
stoptime	nvarchar(16)	<input checked="" type="checkbox"/>
startstationid	int	<input checked="" type="checkbox"/>
startstationname	nchar(50)	<input checked="" type="checkbox"/>
startstationlatitude	float	<input checked="" type="checkbox"/>
startstationlongitude	float	<input checked="" type="checkbox"/>
endstationid	int	<input checked="" type="checkbox"/>
endstationname	nchar(50)	<input checked="" type="checkbox"/>
endstationlatitude	float	<input checked="" type="checkbox"/>
endstationlongitude	float	<input checked="" type="checkbox"/>
bikeid	numeric(5, 0)	<input checked="" type="checkbox"/>
usertype	nchar(20)	<input checked="" type="checkbox"/>
birthyear	nchar(10)	<input checked="" type="checkbox"/>
gender	int	<input checked="" type="checkbox"/>
startyear	nvarchar(4)	<input checked="" type="checkbox"/>
startmonth	nvarchar(2)	<input checked="" type="checkbox"/>
startday	nvarchar(2)	<input checked="" type="checkbox"/>
starthour	nvarchar(2)	<input checked="" type="checkbox"/>
startmin	nvarchar(2)	<input checked="" type="checkbox"/>
startdayofweek	nvarchar(2)	<input checked="" type="checkbox"/>
age	int	<input checked="" type="checkbox"/>
endyear	nvarchar(4)	<input checked="" type="checkbox"/>

Figure 11 Structure of the table trips

After having the data uploaded in the server more fields were created from the same data. For the start.time the field startyear, startmonth, startday, starthour, startmin and startdayofweek was created. The same set of fields was created for the field end.time. To have the age of the user instead of the year of birth another field was created. The structure of the new table can be seen in the image below.

Note: Although it is clear it is not the best practice for a big-data project my problem

was not the storage space but the power of process. For this reason some fields were calculated in the database to have them ready for the R program to deal with.

To populate those new fields from the data which has already been uploaded, the SQL query below was created and executed.

```

update tripstest set
startyear = SUBSTRING(starttime, 7, 4),
startmonth = SUBSTRING(starttime, 4, 2),
startday = LEFT(starttime, 2),
starthour = SUBSTRING(starttime, 12, 2),
startmin = RIGHT(starttime, 2),
startdayofweek = datepart(dw, CONVERT(DATE, starttime, 103)),
endyear = SUBSTRING(endtime, 7, 4),
endmonth = SUBSTRING(endtime, 4, 2),
endday = LEFT(endtime, 2),
endhour = SUBSTRING(endtime, 12, 2),
endmin = RIGHT(endtime, 2),
enddayofweek = datepart(dw, CONVERT(DATE, endtime, 103)),
age = (2013 - cast(birthyear as int))

```

Figure 12 SQL Query to populate the new fields

The last process that was carried out in the dataset was to calculate the age of the user as an integer value instead of having the year of birthday. This was done in the same query as the previous process but what has to be underlined is that because the dataset has only the birth year the age resulting from the calculation

could be wrong for some users because the data goes from July 2013 to May 2014. Because it does not have influence in the result, the reference to calculate the age was 2013.

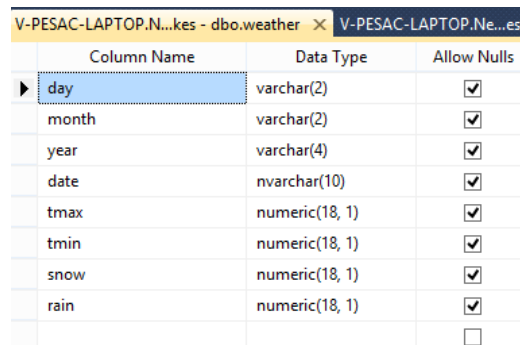
The processes with the second dataset were similar. In the first step the stations were plotted on a map to find the six nearest meteorological stations to the region where the scheme operates. As it can be appreciated in the image, five of the stations are around the area and one is in the middle of the area of interest. Bear in mind that from the weather point of view there is not a big difference in temperature or in other weather conditions in one kilometre.



Figure 13 Distribution map of the meteorological stations

The rest of the data from other stations was removed from the data set and only the data from the six meteorological stations in scope were kept. From the initial 19927 rows 2304 were kept, which means a row per day per station approximately. Before uploading the data to SQL Server two more steps were carried out with the dataset. The first was to remove all the fields not of interest, like wind, humidity, ice, and so on. The second step was to calculate the average of the six stations with R. The “NA” values were removed to calculate the average of every field for every day data. Note: If for example the second of April has only three stations with data and two without, the three values were summed up and divided by three. The same operations were executed for the rain and the snow in the entire dataset.

Once the data was ready to upload to the SQL Server a table was created in the “NewYorkBikes” database to storage weather data. The structure of the table is shown in the next image.



Column Name	Data Type	Allow Nulls
day	varchar(2)	<input checked="" type="checkbox"/>
month	varchar(2)	<input checked="" type="checkbox"/>
year	varchar(4)	<input checked="" type="checkbox"/>
date	nvarchar(10)	<input checked="" type="checkbox"/>
tmax	numeric(18, 1)	<input checked="" type="checkbox"/>
tmin	numeric(18, 1)	<input checked="" type="checkbox"/>
snow	numeric(18, 1)	<input checked="" type="checkbox"/>
rain	numeric(18, 1)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Figure 14 Structure of the table Weather

Having uploaded the 334 rows in the table of SQL, two new fields were created. The purpose of these new fields was to hold the day and the month of the date.

Because one of the initial purposes of the project was to find out the correlation between the weather and the use of bikes this dataset was very important. With the

progress of the project and having in mind the opinion of some lecturers the main objective was changed and this dataset was not important at all for the new research. Nevertheless some graphs have been plotted.

### 3.6.4 Issues during these processes

This section describes the most important issues encountered during the process of acquiring, cleaning and processing the datasets from the technical point of view and from the project management point of view.

- Because the College and the people involved in the whole process of the project are located in Dublin, at the first stage of the project the purpose was to get the dataset from the shared bike scheme of Dublin. Several requests were sent to them but in the end it was not possible to obtain the dataset. Their answer was that: “**Unfortunately, we are not in a position to assist you at this time.**” It was decided to change to another shared scheme in a big city around the world which would allow access to their dataset. However, a lot of time was consumed waiting for an answer which could have had a big impact on the progress of the project.
- When the New York Scheme was chosen to work with, the months of March, April and May were missing from their website. I wrote an email to the person in charge and one week later those monthly datasets were



uploaded to the website, making them available for everybody. It is thanks to that the project can count on more recent data.

- The data requirements established that the id of the user who uses the bike should be on the dataset to obtain a better result. After exchanging four emails regarding request for access to this data with the person in charge of the data, our request was definitely refused and therefore this data could not be included in the dataset for this project. Although it could have a big impact on the achievement of the project it was decided to continue with the project anyway.
- From the technical point of view a lot of problems were solved during the processes of data acquisition, cleaning and preparing. The one which has been chosen to be underlined in this section is about the format of the fields “start.time” and “end.time” that in the csv file had a long date format “dd/mm/yyyy hh:mm”, “21/05/2014 08:23”. The problem was that in the upload to the SQL database no matter what the format of the field was it was always uploaded with the American format “mm/dd/yyyy hh:mm”. The problem was that the day was exchanged with month and vice versa. On top of that a lot of errors popped up because the days greater than 12 were not possible to convert to month. After looking for a solution on the internet and not having the possibility to change the data format in the computer, it was decided to upload it as a text field and work out the day, month, hour and minute in another field.

## 4 Research Findings

As it has been established in the scope of the project the main purpose is to find out whether it is possible to predict the end station after the user leaves the start station or is something random and unpredictable.

### 4.1 An in-depth view of the dataset

In any project of big-data the first process which has to be carried out is the understanding of the dataset and the business involved. From a superficial look at the dataset seems, from the logical point of view, that all the trips has a random star / end station. This section sheds light on the fact that in the apparently chaotic movements in the scheme there are some patterns that could help to understand better the relation between the different fields and to focus the research on some Machine Learning algorithms.

As a first approach to the dataset the distribution of the stations around the city was plotted in a new feature that Microsoft Excel 365 has implemented. It will allow the reader to understand how they are distributed around the city.

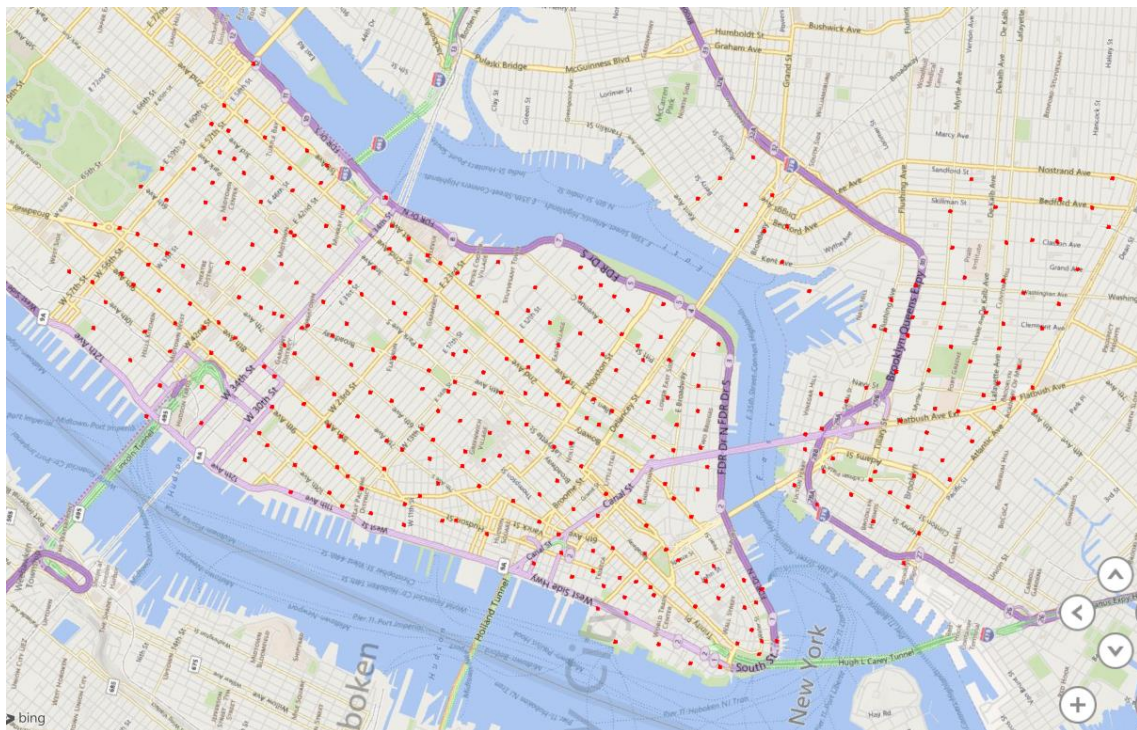


Figure 15 Distribution map of the bikes station around New York



The share bikes scheme extends for around 17 Km from one station on one side of the scheme to the farthest station.

The first approach to understanding the data and the business is to plot the trips along the time. It will show whether there are patterns in the use of bikes or not and between which variables. The graphs below show the use of bikes during the year during different periods of time: months, days and during a work day or weekend day.

The first image below shows the chart of the number of trips have been made in the scheme of New York Bikes by month. From this image can be deducted that the use of bikes is very seasonal and it depends on the month of the year. It is clear that the use of bikes depends directly on the temperature and on the rain or snow. Note: Some analysis has been carried out to prove this statement which has been included in the Appendix because the analysis of the correlation was not in the scope of the project anymore.

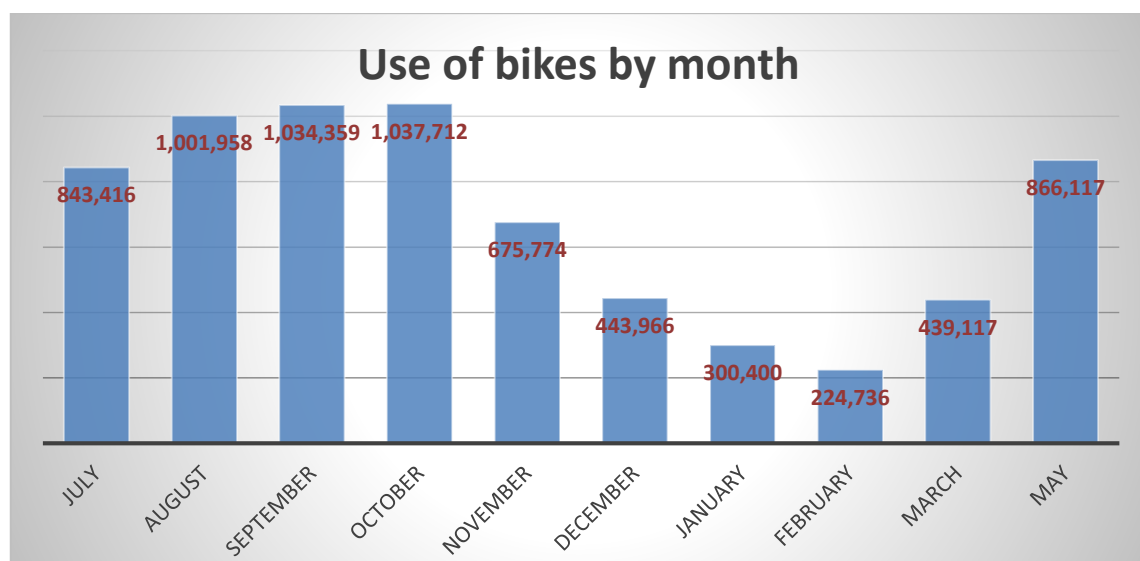


Figure 16 Total trips by month

The image below shows the distribution of the number of trips by days in the months. It seems clear that there is no pattern of use during the month. Although it could be expected, there is no weekly pattern or other type of behaviour in the period of a month. There is an example in the appendix 4 that shows the data of two months by the day of the week and nothing can be inferred. Note: The program developed to draw this series of plots is in the appendix.

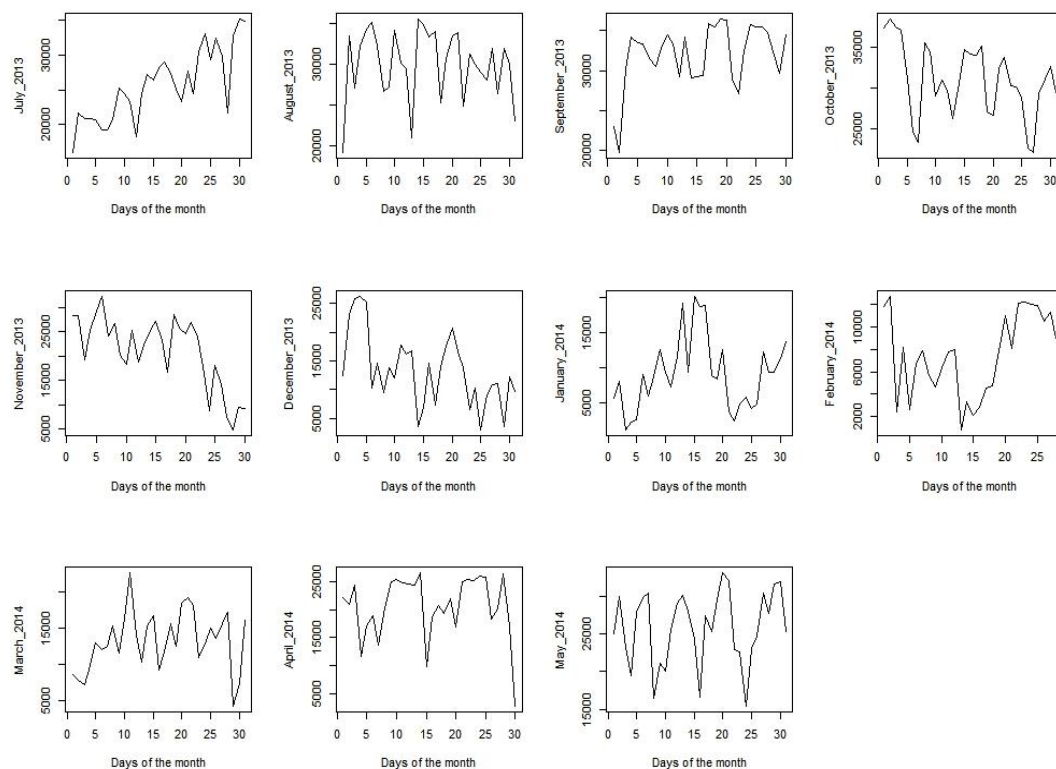


Figure 17 Total trips by day during the year

On the other hand, in the image below it is clear that the use of bikes on the different days of the week follows two clear patterns. All the work days of the week have the same shape whereas weekend days have a totally different one but are similar between Saturday and Sunday. This graph gives the first indication that there are certain patterns in the movements of bikes around the city.

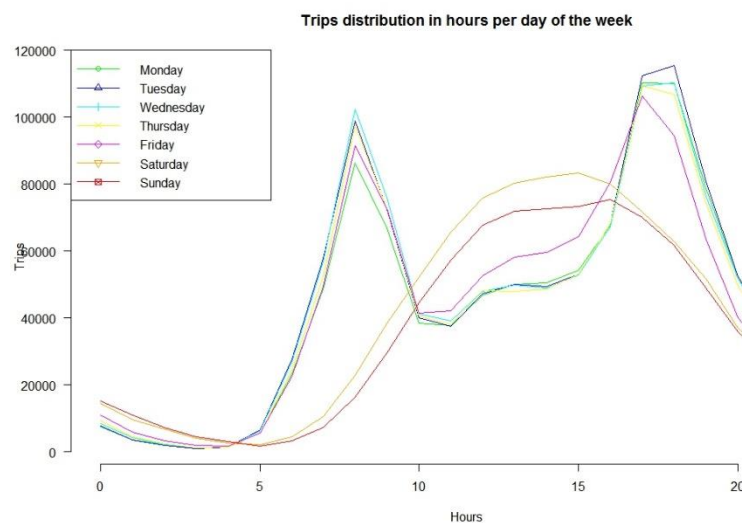


Figure 18 Total trips by hour the seven days of the week.

Note: The program in R developed to plot this graph is in the Appendix named as “Plot the trips hour by day of the week.R”

It would be logical to think that just as there are streets in the cities busier than others, there should also be bike stations more frequently used than others. To check this statement the graph below was develop, which shows that the stations on the outskirts of the city are far less used than some in the city centre. However, even in the same area, there are different frequencies of use for stations very near each other. One reason that came to my mind is that it could be because the number of bikes/docks is larger or more conveniently located than other.

If the model had to predict the final station of a trip, having a look at that graph, it could be based on the chances of getting a final station by previous amount of use. As a question of probability the most used have more chances of receiving a bike. This is a very simple approach but it has been the foundation of some studies which have searched for an average time prediction.

For the purpose of this project and with images like the one below for each month some videos have been developed. They show from different view points around the city the evolution of use of bikes by stations. The videos can be found in the share folder.



Figure 19 Total trips by station in a map of New York

Another logical finding, but not as obvious as the previous one, is the distribution of the end stations having a particular station as start. As it can be appreciated in the example that is represented in the image below not all the stations have the same probability of being the end station for a trip started in the “1 Ave & E 15 St” station.

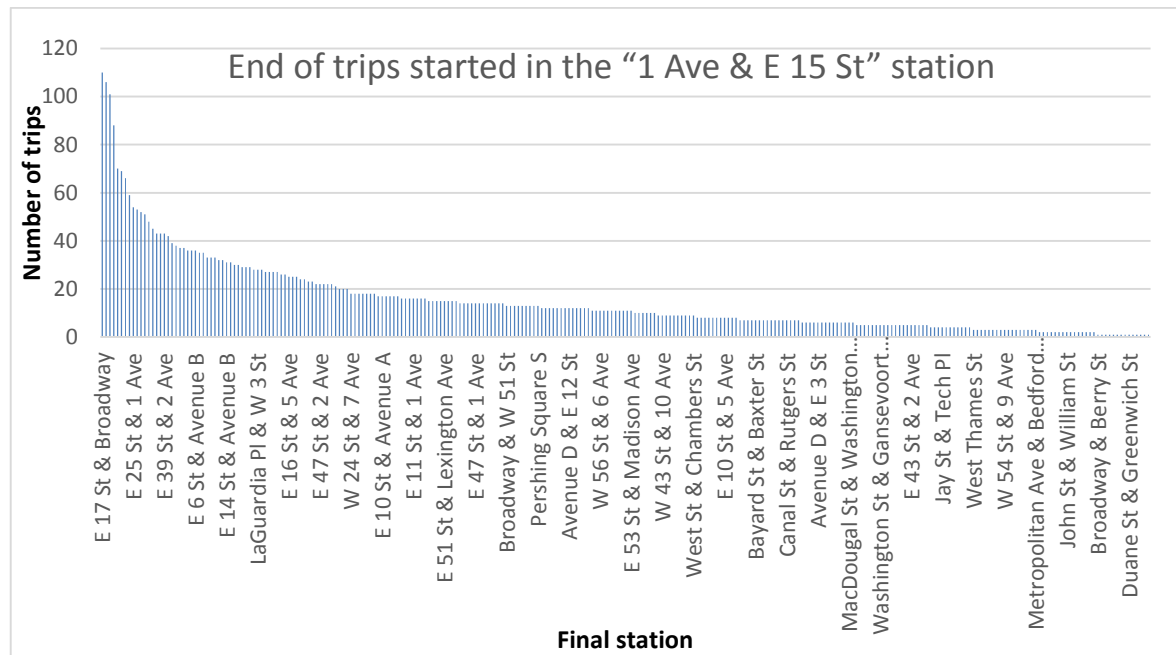


Figure 20 Total trips started in "1 Ave & E 15 St" by the station of ending

From the point of view of the data trip-duration, for all the trips which are common on the same start station and end station or vice versa, it seems that they obey a Normal distribution. Although it is not very significant, it was decided to plot two different graphs for men and women. Sometimes the normality of a variable is hide for plotting together both genders.

The image below shows the histogram of the trip-duration in seconds for Males and Females from the station 229 to 497 top / 301 bottom.

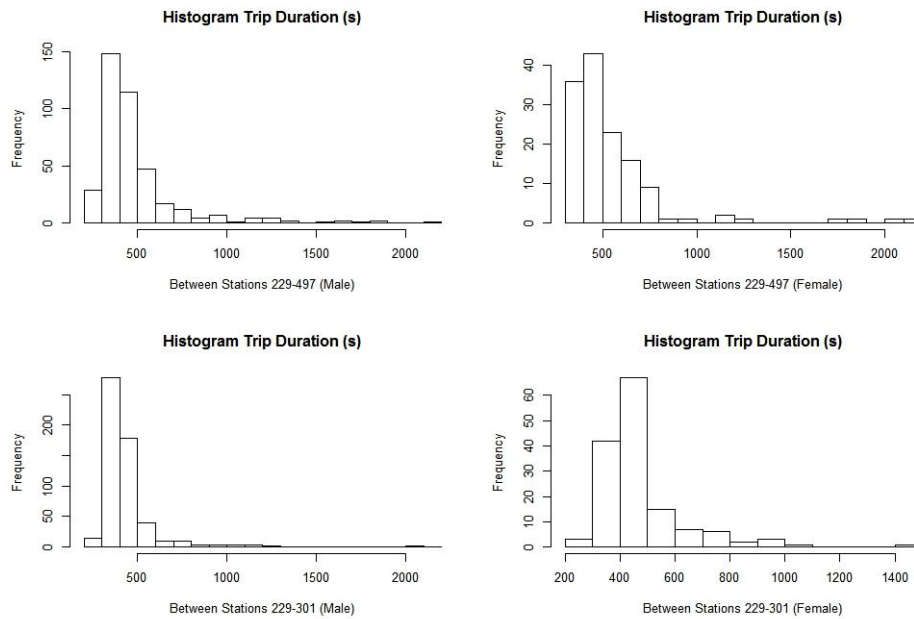


Figure 21 Histogram of trip duration between two stations by gender

Analysing the dataset from the point of view of the field “age”, as it is shown in the image below, there are some patterns that should be underlined. The graph shows the number of trips by age in three stations chosen on purpose to prove that age can be also be used to look for patterns. One station is beside the University and the chart shows that the average is lower than in another station that is beside a Business Area. The third station is beside a hospital and the distribution of trips by age is different from the other two. What can be concluded is that the end station also depends on the age so the probability of ending in a station depends to some extent on the age in some cases.

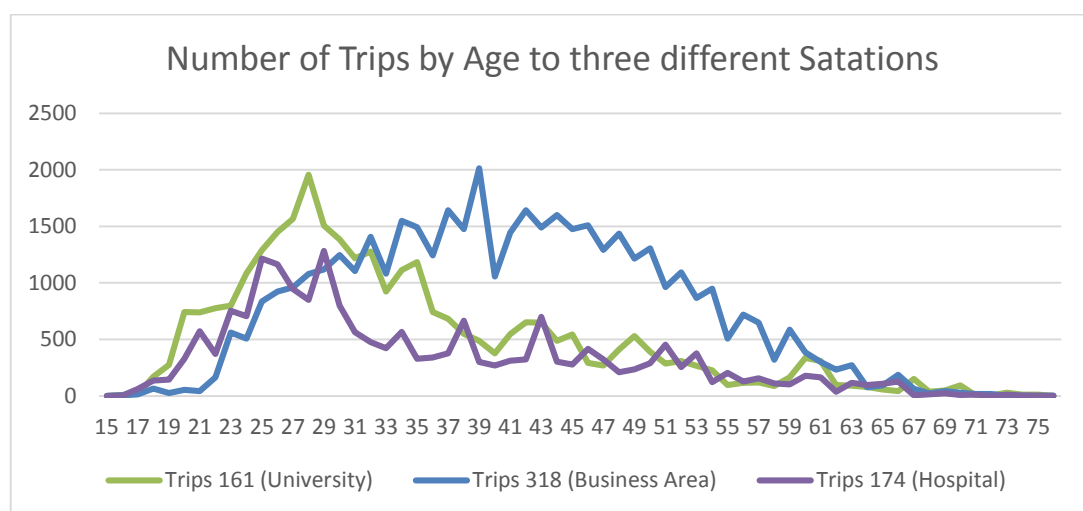


Figure 22 Total trips by age in three different stations

Although the percentage of use of bikes is higher among males it could be expected that the percentage of trips that finish in one stations by gender should be similar in all the stations but it is not. The percentage of females that arrive in the business station by bike is much lower than in the university for example. It could be because business women are usually dressed in clothes unsuitable for riding a bike. However, this statement is not based on anything scientific or any test carried out in this project.

GENDER	NUMBER TRIPS			PERCENTAGE		
	174 Hospital	161 University	318 Business Area	174 Hospital	161 University	318 Business Area
Unknown	1320	3846	1710	6%	11%	4%
Male	13351	20043	40549	63%	60%	85%
Female	6492	9607	5298	31%	29%	11%
Total	21163	33496	47557	100%	100%	100%

Table 3 Distribution of trips by gender in three stations

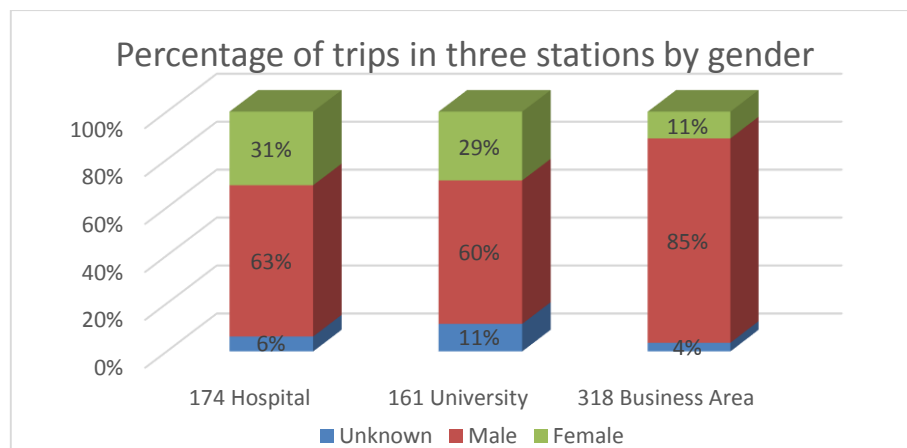


Figure 23 Percentage of trips by gender in three different stations

From the analysis of all these graphs can be extracted enough evidence can be extracted to support the idea that there are different probability distributions for some events to happen. The conclusion of this section is that the dataset has enough information from numerous attributes than can be considered simultaneously to estimate the probability of one event. Although the effect of each attribute may have a relatively minor impact on the rest, taken together their combination could have a larger impact between each other. To see whether this



statement is true or not further analysis will be carried out with the help of some Machine Learning algorithms that can be applied to solve the question.

## 4.2 Simplification

Due the complexity of the problem that the project is addressing and the lack of budget and powerful machines to execute the algorithms across the entire dataset, the analysis is going to be implemented in a sub-dataset of nine id stations. It contains approximately 18000 trips that will be used to apply the Machine Learning Algorithms. Although the dataset has been reduced the program which has been developed is able to select different stations and number of them even carry out the same analysis without making any change in the code. Note: Although the first number of stations chosen was nine one of the numbers does not correspond to a real station.

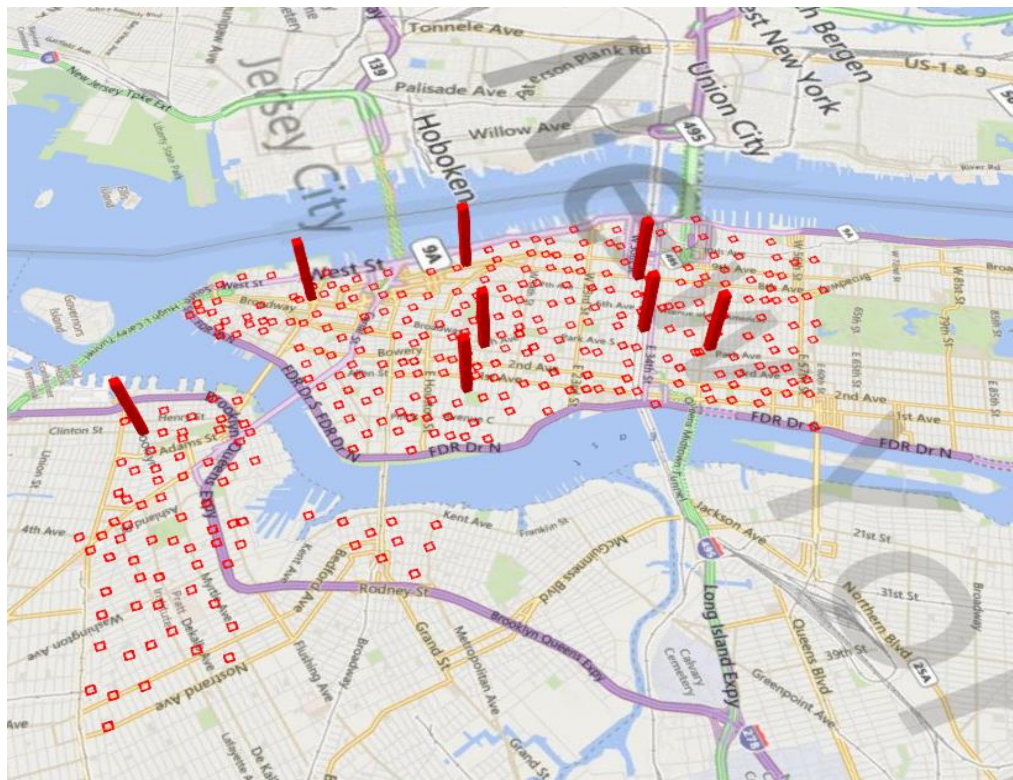


Figure 24 Location in a map the eight chosen stations for the analysis

The stations were chosen randomly and without any other criteria. After completing the analysis for the sub-dataset an estimation of the time and cost will be done for a further analysis with the entire dataset.

## 4.3 Analysis 1: Naïve Bayes

Having proven that the probability of some features in the dataset are conditioned to other attributes of the same dataset, Naive Bayes seems to be the best and simplest algorithm to find a classifier model to predict the most likely result for the new trips. This is a Supervised Machine Learning algorithm which has been taught during the course so the student is familiar with it.

#### **4.3.1 Research question:**

- Will Machine Learning work in this dataset to predict the end station of a trip?
- What will the accuracy be for that prediction?

#### **4.3.2 Description**

For some probability models, Naive-Bayes classifier can be trained efficiently in a supervised learning setting. (Ref Wikipedia) The purpose of this analysis is to develop a program in R to apply this algorithm to a random part of the reduced dataset to train the model and then test how well it works with the rest of the dataset. It will give a result that can be considered as the accuracy of the model.

#### **4.3.3 Method**

Although from the previous section I have learnt that there are some patterns in the data that would give more accuracy in the model, in the first try the algorithm was applied to the whole sub-dataset of eight stations without any restriction of hours, days, or type of user. For this purpose an R package called “caret” was used which lets the user apply different machine learning algorithms and the method that the algorithm is going to use to learn or fit the model. In this case, to evaluate the efficacy of the statistical mode, “Cross-Validation” has been chosen as a method of resampling which is one of the more tested.

A screen shoot of the code program can be appreciated in the image below with the details of the parameters applied. The entire code is available in the appendix and in the share folder created for this purpose.



```

1 #####
2 # program to apply Naive Bayes in the stations during one day #
3 #####
4
5 #Load the libraries the program is going to use
6
7 library("k1ar")
8 library("caret")
9 library(Rodbc)
10
11 #open the ODBC connection to bring the data
12 channel1 <- odbcconnect("SQLodbc")
13 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
14 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
15 (143,152, 186, 293,318,358,432, 521, 526)"
16 # run the query
17 tripsbetweenstations=sqlQuery(channel1,querystations)
18 #Define some variable to filter more the data
19 morning = c(0,8,9)
20 afternoon = c(17,18,19)
21 day = c(0:23)
22 workday = c(2,3,4,5,6)
23 weekend = c(1,7)
24 #divide the dataset into the independent variables and the dependent variable
25 xtrain = tripsbetweenstations[,c(1:7)]
26 ytrain = tripsbetweenstations[,8]
27 #Run the model with the package caret and the funtion train
28 model = train(xtrain,ytrain,'nb', trcontrol=traincontrol(method='cv', number=12))
29 write.csv(model$results, file="model.csv")
30 plot(model)
31 model

```

Figure 25 Screen shoot of the program Naive Bayes

As a result of the program some key details of the dataset used comes up. From the image below it can be appreciated that the sub-dataset contains around 18,000 rows and seven predictors or independent variables to determine the dependent variable which has eight classes or possible values.

Naive Bayes

17146 samples  
 7 predictors  
 8 classes:

linton St & Joralemon St, 'E 33 St & 5 Ave', 'Christopher St & Greenwich St', 'E 43 St & Vanderb  
 it Ave, 'E 7 St & Avenue A', 'Lafayette St & E 8 St  
 ', 'warren St & Church St

No pre-processing  
 Resampling: cross-validated (12 fold)

Summary of sample sizes: 15718, 15717, 15715, 15717, 15720, 15715, ...

Resampling results across tuning parameters:

usekernel	Accuracy	Kappa	Accuracy SD	Kappa SD
FALSE	0.409	0.258	0.0149	0.0196
TRUE	0.448	0.319	0.0173	0.0217

Tuning parameter 'fl' was held constant at a value of 0  
 Accuracy was used to select the optimal model using the largest value.  
 The final values used for the model were fl = 0 and usekernel = TRUE.

Figure 26 Main characteristics of the dataset and the accuracy of the model.

After running the program, the caret package using Naïve Bayes algorithm has found within this dataset a model with an accuracy of 0.448 with a Standard Deviation of 0.018. The “Accuracy” data is the average of the accuracy of the 200 held-out samples. The “Accuracy SD” column is the Standard Deviation of the 200 accuracies. (All the programs have been modified with number=200 instead of the 12. Nevertheless the accuracy was practically the same). “The Kappa statistic is a measure of concordance for categorical data that measures agreement relative to what would be expected by chance. Values of 1 indicate perfect agreement, while zero would indicate lack of agreement”

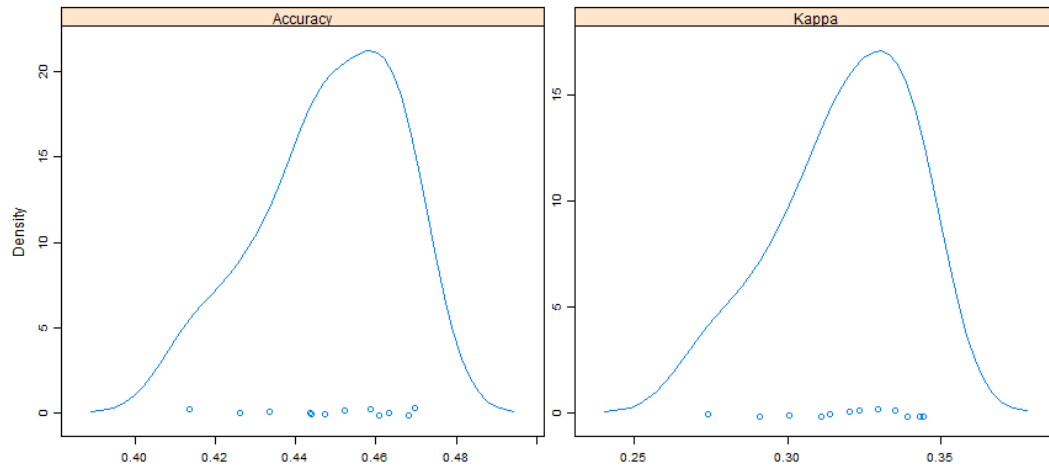


Figure 27 Standard Deviation of the Accuracy and Kappa

With the entire sub-dataset and without any filter in the data trying to find a model in it without having in mind the patterns found in the previous section the accuracy of the system is quite low. The Naïve Bayes has only been able to predict in the best case 48% of the final stations. In the image the density of the accuracy has been plotted and most of the values are in 44% of accuracy with a Standard Deviation of 0.017.

In a second try with the same Machine learning algorithm some of the characteristics that the study of the dataset has taught are going to be considered. This second approach to finding a model has been developed in a program in R which splits the trips of the dataset into work days and weekend days. They have totally different patterns of behaviour so the logic says that it will be more accurate if the data for the model is separated. In this second approach the model takes into account the effect of the variable: “day of the week”. In the vector the five work days of the week has been defined. The first day of the week is Sunday so that is the reason that the vector contains from 2 to 6. Note: It can be appreciated in red in the image below.

```

1
2- #####
3 # program to apply Naïve Bayes in the stations during one day #
4- #####
5
6 #Load the libraries the program is going to use
7
8 library("kLar")
9 library("caret")
10 library(RODBC)
11 #Open the ODBC connection to bring the data
12 channel1 <- odbcConnect("SQLodbc")
13 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
14 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
15 (143,152, 186, 293,318,358,432, 521, 526)"
16 # run the query
17 tripsbetweenstations=sqlQuery(channel1,querystations)
18 #Define some variable to filter more the data
19 morning = c(0,8,9)
20 afternoon = c(17,18,19)
21 day = c(0-23)
22 workday = c(2,3,4,5,6)
23
24 weekend = c(1,7)
25 #Divide the dataset into the independent variables and the dependent variable
26 #As well it has been filter by the workdays
27 xtrain = tripsbetweenstations[tripsbetweenstations$startdayofweek %in% workday, c(1,2,3,4,5,6,7)]
28 ytrain = tripsbetweenstations[tripsbetweenstations$startdayofweek %in% workday, 8]
29
30 #Run the model with the package caret and the funtion train
31 model = train(xtrain,ytrain,'nb', trControl=trainControl(method='cv', number=12))
32 write.csv(model$results, file="model.csv")
33 #Write the model in the console
34 model
35 #Plot the density distribution of the accuracy
36 resampleHist(model)
37 plot(model)

```

Filter to reduce the dataset to the work-days

Figure 28 Screen Shot to show the filter of work days

Like in the previous try the result describes in the header the characteristics of the dataset. The main characteristics are the same but the rows that the algorithm has now to find the model has been reduced from the initial 18.000 to just 12,000 or a 33% less.

```

Naive Bayes
12755 samples
7 predictors
8 classes: '8 Ave & W 31 St', 'Christopher St & Greenwich St', 'E 43 St & Va
linton St & Joralemon St', 'E 33 St & 5 Ave', 'E 7 St & Avenue A', 'Lafayette St & E 8 St
lt Ave', 'warren St & Church St'

No pre-processing
Resampling: Cross-Validated (12 fold)

Summary of sample sizes: 11693, 11691, 11692, 11692, 11693, ...

Resampling results across tuning parameters:

usekernel Accuracy Kappa Accuracy SD Kappa SD
FALSE 0.411 0.267 0.00558 0.00711
TRUE 0.458 0.329 0.0118 0.015

Tuning parameter 'fL' was held constant at a value of 0
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0 and usekernel = TRUE.

```

Figure 29 Main characteristics of the dataset and the accuracy of the model.

In this case the accuracy of the model has increased just a little reaching 46% of cases. In the image below we can see that the density of the graph is different and the standard deviation is lower than in the previous model so the values are closer, less spread out.

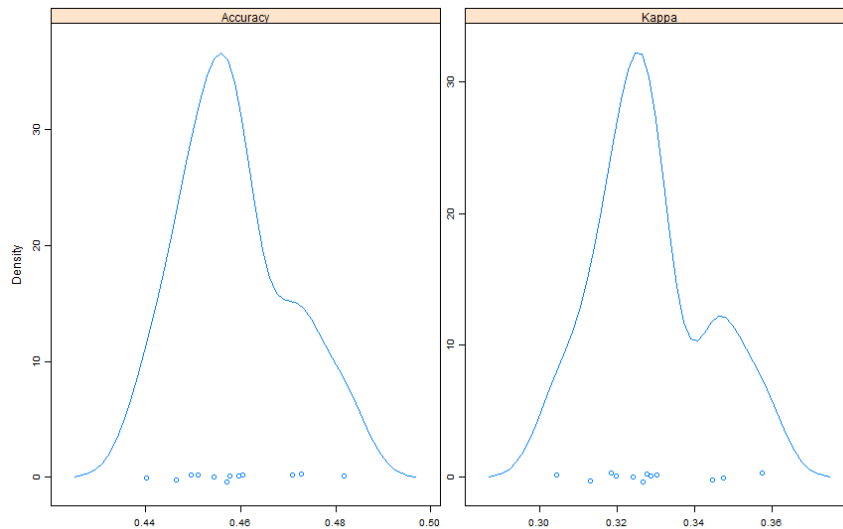


Figure 30 Standard deviation of the Accuracy and Kappa

Going further, in a third try the hours of the day has been reduced to the three rush hours of the morning to see whether the algorithm Naïve Bayes performs better and the accuracy of the model is larger than considering the whole day. As it can be appreciated in red in the picture below the vector of hours contains only from 7 am to 9 am.

```

1
2 #####
3 # program to apply Naïve Bayes in the stations during one day #
4 #####
5
6 #Load the libraries the program is going to use
7
8 library("k1ar")
9 library("caret")
10 library(RODBC)
11 #Open the ODBC connection to bring the data
12 channel1 <- odbcconnect("SQLodbc")
13 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
14 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
15 (143,152, 186, 293,318,358,432, 521, 526)"
16 # run the query
17 tripsbetweenstations=sqlQuery(channel1,querystations)
18 #define some variable to filter more the data
19 morning = c(7:9)
20 afternoon = c(17,18,19)
21 day = c(0:23)
22 workday = c(2,3,4,5,6)
23
24 weekend = c(1,7)
25 #Divide the dataset into the independent variables and the dependent variable
26 #As well it has been filter by the weekdays and the hours 7-9
27 xTrain = tripsbetweenstations[,tripsbetweenstations$starthour %in% morning &
28 tripsbetweenstations$startdayofweek %in% workday,c(1,2,3,4,5,6,7)]
29 yTrain = tripsbetweenstations[,tripsbetweenstations$starthour %in% morning &
30 tripsbetweenstations$startdayofweek %in% workday,8]
31
32 #Run the model with the package caret and the funtion train
33 model = train(xTrain,yTrain,'nb', trControl=trainControl(method='cv', number=12))
34 write.csv(model$results, file="model.csv")
35 #Write the model in the console
36 model
37 #Plot the density distribution of the accuracy
38 resampleHist(model)
39 plot(model)

```

Figure 31 Filter: days of the week and only for the rush hours during the morning

Like in the previous test the result describes in the header the characteristics of the dataset. The total of rows after the filter has been reduced to 3,500 with

seven predictors or independent variables and eight classes in the dependent variable.

```
Naive Bayes
3549 samples
7 predictors
8 classes: 'S Ave & W 31 St', 'Christopher St & Greenwich St', 'E 43 St &
inton St & Joralemon St', 'E 33 St & 5 Ave', 'E 43 St &
t Ave', 'Lafayette St & E 8 St',
', 'Warren St & Church St', 'E 7 St & Avenue A',

No pre-processing
Resampling: cross-validated (12 fold)

Summary of sample sizes: 3254, 3254, 3254, 3253, 3253, 3253, ...

Resampling results across tuning parameters:

usekernel Accuracy Kappa Accuracy SD Kappa SD
FALSE      0.599    0.435    0.0135    0.0196
TRUE       0.626    0.472    0.0164    0.0238

Tuning parameter 'fL' was held constant at a value of 0
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0 and usekernel = TRUE.
```

Figure 32 Main characteristics of the dataset and accuracy of the model during 7 to 9am

As it can be appreciated in red in the image above that the accuracy of the model has improved to 0.63 which means that the movements that take place in the rush hours during the morning have a more defined pattern.

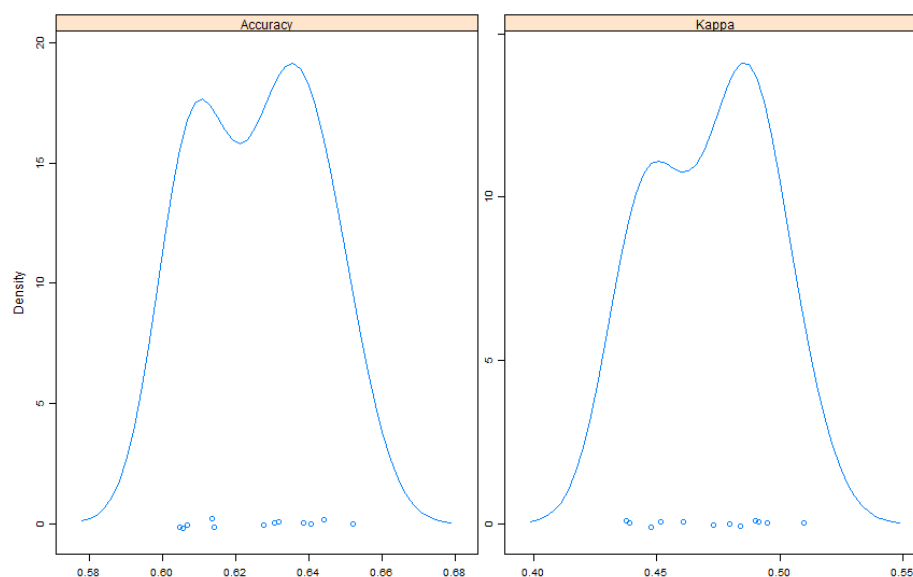


Figure 33 Standard Deviation of the accuracy and kappa

The standard deviation for this try is similar to previous tests and means that the spread of the predictions is quite close to the mean.

As a last try with the supervised Machine Learning algorithm Naïve Bayes a program has been developed to split the dataset in bunches of trips per hour, so 23 bunches. The program carries out the algorithm for each bunch and saves the

result models in order to compare the accuracy. The program is in the appendix “NaiveTotal Work-Days Different Hours.R”

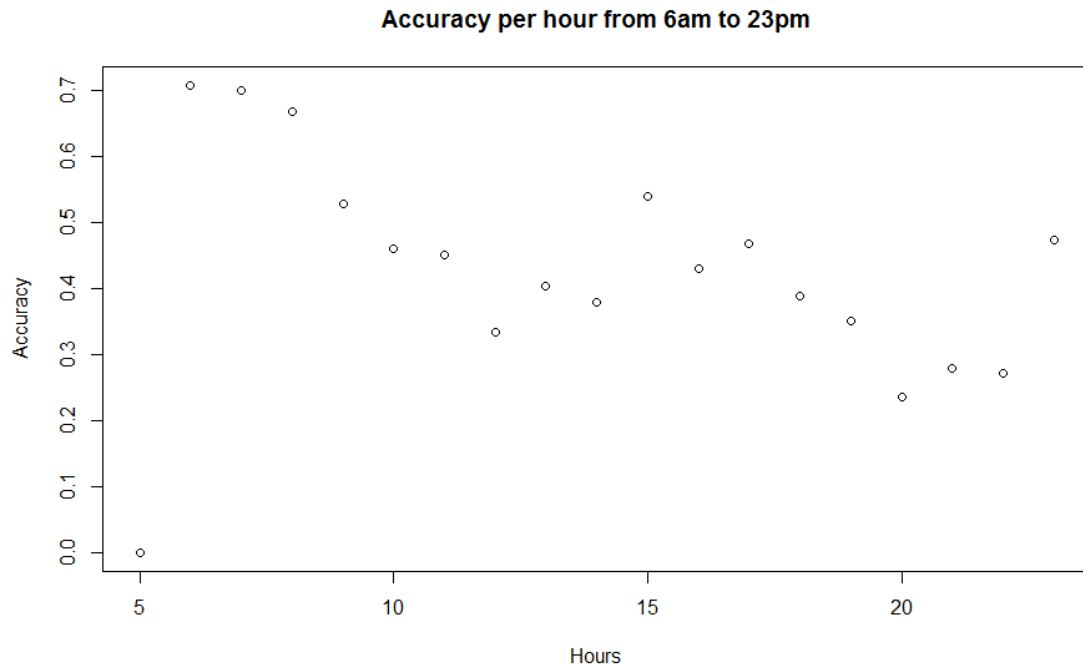


Figure 34 Evolution of the accuracy during a work day with Naïve Bayes Algorithm

After seeing that the accuracy depends on the day of the week and the hour of the day, the development of a new program to keep all the data from the whole day was considered.

#### 4.3.4 Results

The Naïve Bayes algorithm has not performed very well for this particular dataset and it could be because of the short number of variables and the small impact that some of them have on the final result. The accuracy of the model is very low with an average of 0.48 and the maximum during the morning hours of 7, 8 and 9 am with around 0.71 of accuracy. Due to the poor performance of this Machine Learning algorithm another Machine Learning Algorithm was tried in order to find out whether a larger accuracy is not possible or the model is not the right one.

#### 4.4 Analysis 2: Gradient Boosting Machine.

This algorithm is a supervised Machine Learning algorithm that belongs to regression techniques family. This produces a prediction model in the form of an ensemble of weak prediction models, normally decision-trees. This method was developed by Jerone H.Friedman in 1999 and performs very well in many cases. This method was developed with an example where the author described an important tweak to the algorithm which allows it to improve its accuracy and performance (StatSoft Company, 2013)

One of the advantage of this model over the Naïve Bayes is that more control can be had over the grid of tuning with some parameters as it will be explained in the method section of this point 4.43.

#### **4.4.1 Research question:**

- How well will GBM perform?
- What will the accuracy be for the model obtained by this algorithm?

#### **4.4.2 Description**

For some probability models, Naive-Bayes classifier can be trained efficiently in a supervised learning setting. (Ref Wikipedia) The purpose of this analysis is to develop a program in R to apply this algorithm to a random part of the reduced dataset to train the model and then test how well it works with the rest of the dataset. It will give a result that can be considered as the accuracy of the model.

The same R package “caret” is used for this analysis but changing the algorithm parameter method to “gbm” instead of “nb”. This method has more control over the grid of tuning parameters which will let them adjust to improve the performance of the model. As an example, the number of trees “n.trees”, the complexity of the trees with the parameter “depth” and the learning rate with “shrinkage” could be tuned to improve the accuracy. From the point of the cross validation can be decide how many times the algorithm is going to be performed which in the case of this project was 10 times with 10 different sets of data.

Since in the project there is a finite amount of rows to use for training and evaluating the model, one of the first decisions was to determine how the samples for the evaluation should be utilized. Although in this project both algorithms use

resampling (Cross-validation) to evaluate the accuracy of the model and could be sufficient in most cases, sometimes resampling alone may be not enough. To make sure that the model performs as well as in the Cross-Validation, the dataset used in each model has been split randomly. In all cases, 75% of the data has been used for the training and the other 25% for evaluating the performance of the model. The piece of code to do it has been included in the image below.

```
29
30 set.seed(14*(iI+1))
31 # Create a partition with data for the model and the rest for testing the model.
32 # 75% for the model and 25% for testing
33 inTraining <- createDataPartition(bikeswork$endstationname, p = 0.75, list = FALSE)
34 training <- bikeswork[inTraining, ]
35 testing <- bikeswork[-inTraining, ]
36
```

Figure 35 Rate used to split the dataset into training data and test data

#### 4.4.3 Method

In the first try a program which was developed gets the data of the nine stations from the SQL Server and then applies a new filter in R to leave only the rows of trips which have been made on work days. Because all these tests have been done hundreds of times the control parameters over the grid of tuning have been optimised in previous attempts. All those tries have been omitted due to the lack of time and word count in this paper.

In the image below the code of the program has been included although the entire code has been included in the appendix and in the share folder like the rest of the code programs.

The program has four main steps. The first step is get the data and filter for the days that we want to analyse. Step two is to define the grid fit and divide randomly the dataset into training data and testing data. The third is to run the algorithm to find the model and the last is to show the data of the model in numbers and graphs that will be included later on.



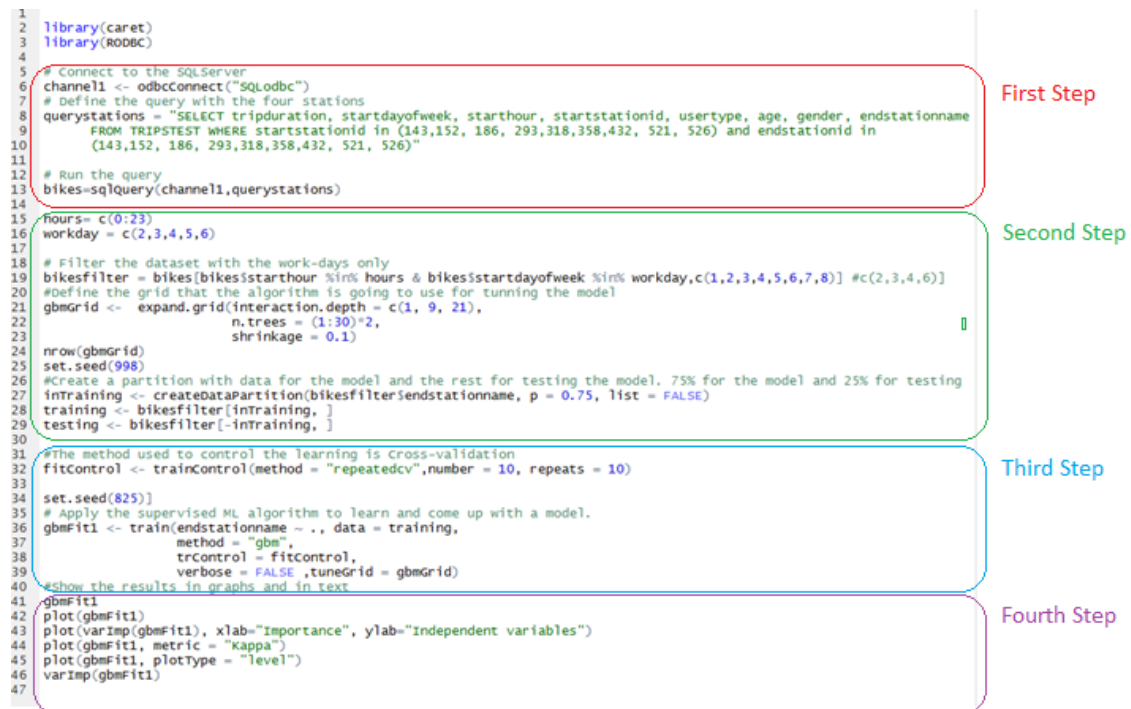


Figure 36 Main steps of the program

The image below is a screen shoot of the amount of data that the algorithm is going to use in the first attempt. The first dataframe “bikes” has the entire number of rows for the nine stations. The second one “bikesfilter” has the rows for the work days. The “testing” dataframe with a 25% and the “training” with the 75% rest.

Data		
① bikes	17146 obs. of 8 variables	
① bikesfilter	12755 obs. of 8 variables	
① gbmGrid	90 obs. of 3 variables	
intraining	int [1:9569, 1] 1 3 8 11 15 20 21 29 40 43 ...	
① testing	3186 obs. of 8 variables	
① training	9569 obs. of 8 variables	
values		
channel1	Class 'RODBC' atomic [1:1] 1	
① fitControl	List of 22	
① gbmFit1	Large train (23 elements, 13.4 Mb)	
hours	int [1:24] 0 1 2 3 4 5 6 7 8 9 ...	
querystations	"SELECT tripduration, startdayofweek, starthour, s...	
workday	num [1:5] 2 3 4 5 6	

Figure 37 View of the different dataset and variables used

```

Console ~/
Stochastic Gradient Boosting

9569 samples
 7 predictors
 8 classes: '8 Ave & W 31 St', 'Clinton St & Joralemon St', 'Christopher St &
Greenwich St', 'E 33 St & 5 Ave', 'E 43 St & Vanderbilt Ave',
'E 7 St & Avenue A', 'Lafayette St & E 8 St',
'Warren St & Church St'

No pre-processing
Resampling: Cross-validated (10 fold, repeated 10 times)

Summary of sample sizes: 8614, 8611, 8612, 8612, 8611, 8613, ...

Resampling results across tuning parameters:

interaction.depth n.trees Accuracy Kappa Accuracy SD Kappa SD
1 2 0.393 0.235 0.0366 0.049
1 4 0.398 0.243 0.03 0.0451
1 6 0.425 0.282 0.0175 0.0254
1 8 0.442 0.308 0.0168 0.0219
1 10 0.46 0.331 0.0176 0.0223
1 12 0.458 0.329 0.0178 0.0224
1 14 0.459 0.331 0.0159 0.0202
1 16 0.463 0.336 0.0165 0.0211
1 18 0.466 0.34 0.016 0.0202
1 20 0.471 0.346 0.0145 0.0184
1 22 0.475 0.352 0.015 0.019
1 24 0.48 0.359 0.0143 0.0181
1 26 0.485 0.365 0.0151 0.0191
1 28 0.488 0.37 0.015 0.019
1 30 0.492 0.375 0.0145 0.0183
1 32 0.495 0.379 0.0151 0.0189
1 34 0.498 0.383 0.0147 0.0185
1 36 0.5 0.386 0.0148 0.0186
1 38 0.502 0.388 0.0149 0.0187
1 40 0.504 0.391 0.0145 0.0182
1 42 0.505 0.393 0.0153 0.0191
1 44 0.506 0.395 0.0151 0.0188
1 46 0.507 0.396 0.015 0.0186
1 48 0.508 0.398 0.0153 0.0189
1 50 0.509 0.399 0.0147 0.0183
1 52 0.51 0.401 0.0152 0.0188
1 54 0.511 0.402 0.0148 0.0182
1 56 0.512 0.403 0.0149 0.0185
1 58 0.513 0.404 0.0152 0.0187
1 60 0.513 0.405 0.0148 0.0183
9 2 0.647 0.571 0.0166 0.0201
9 4 0.659 0.586 0.0158 0.0191
9 6 0.664 0.592 0.0151 0.0181
9 8 0.667 0.596 0.0142 0.0171
9 10 0.671 0.601 0.0139 0.0166
9 12 0.675 0.605 0.0135 0.0162
.....
21 24 0.709 0.648 0.0118 0.0143
21 26 0.711 0.648 0.0123 0.0149
21 28 0.711 0.649 0.0123 0.0149
21 30 0.712 0.649 0.0123 0.0148
21 32 0.713 0.65 0.0118 0.0142
21 34 0.713 0.651 0.012 0.0145
21 36 0.714 0.652 0.0117 0.0141
21 38 0.714 0.653 0.0118 0.0143
21 40 0.714 0.653 0.0124 0.015
21 42 0.715 0.653 0.0123 0.0148
21 44 0.716 0.654 0.0118 0.0143
21 46 0.716 0.655 0.0119 0.0144
21 48 0.716 0.655 0.0118 0.0143
21 50 0.717 0.656 0.0114 0.0137
21 52 0.717 0.656 0.0111 0.0134
21 54 0.717 0.656 0.0109 0.013
21 56 0.718 0.657 0.0108 0.013
21 58 0.718 0.657 0.0108 0.0129
21 60 0.718 0.657 0.0111 0.0133

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were n.trees = 60, interaction.depth = 21 and shrinkage = 0.1.

```

Figure 38 Screen shot of the results

As it can be appreciated in the image above the algorithm has worked with 9596 rows with seven predictors and eight classes for the dependent variable. It shows the accuracy for each tree and it can be appreciated how the algorithm increases the accuracy of the model one by one.

Additionally, this package has a function to evaluate the importance of each of the independent variables to predict the dependent one. In this case the “startstationid” variable is the most important followed very closely by the “tripduration” and very far in importance, the rest of the variables. The detail that can be extract from this graph is that some variables do not have any effect in the model.

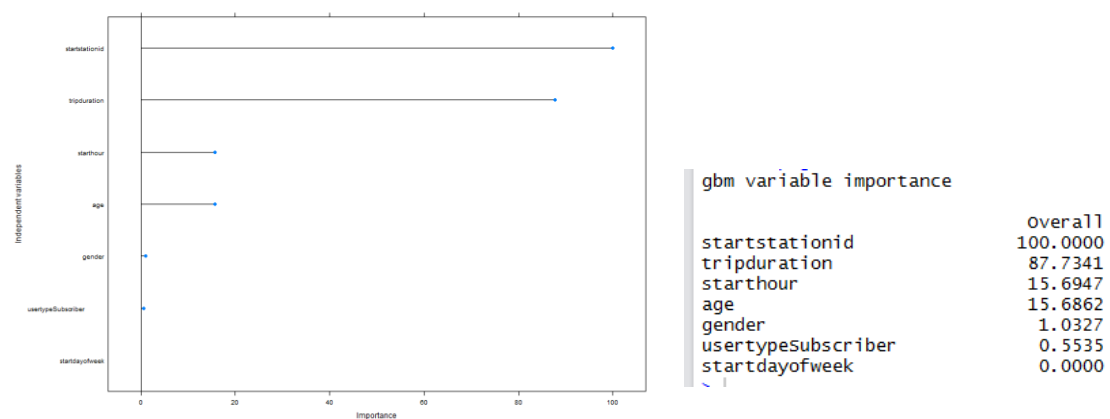


Figure 39 Independent variable by importance in the model

The right side of the image shows the importance of each variable in the model in numbers.

Appreciating the improvement of the model as the algorithm tries many times is difficult in the visual text result so it was decided to plot the progression in the different repeated cross validation of the variable “Accuracy” in the first graph below, and the “Kappa” in the second one. In this first try the accuracy reaches 0.73 whereas Naïve Bayes was only able to reach 0.46

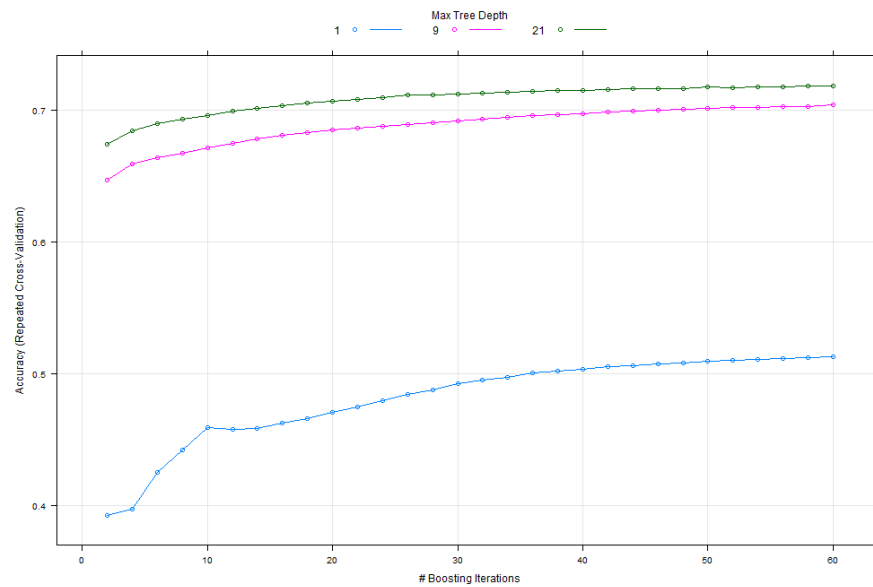


Figure 40 Evolution of the accuracy with the number of trees

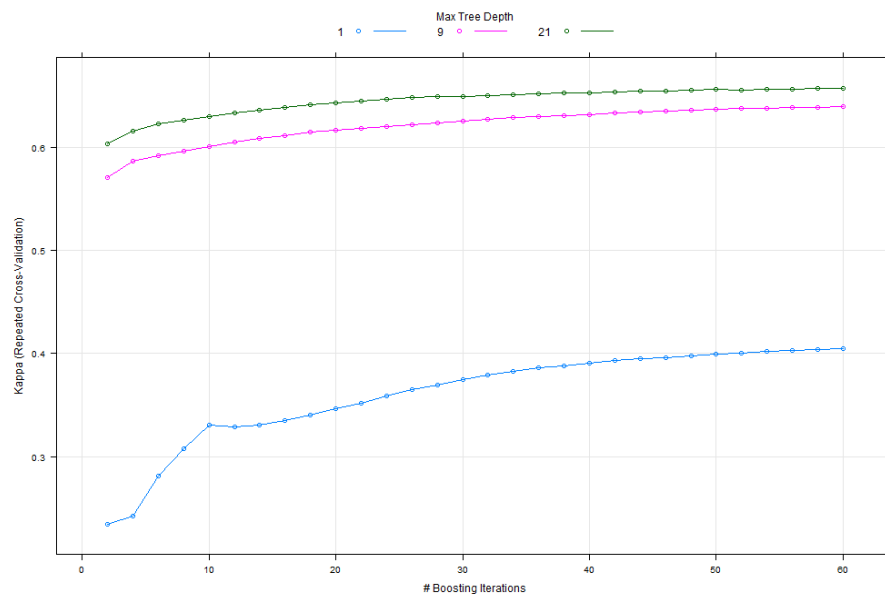


Figure 41 Evolution of the Kappa with the number of trees

It is clear that for a certain number of boosting iterations or number of trees it is not worth the time that the processor has to spend for the small increase of Accuracy. The same conclusion is applicable for the complexity of the model that no matter how big it is, if it is larger than ten the algorithm reaches an accuracy more or less similar. In the image below that statement can be appreciated clearer. If the colour of the graph does not change that means that the model has

reached the maximum accuracy and going further does not make any sense but spend time of processing.

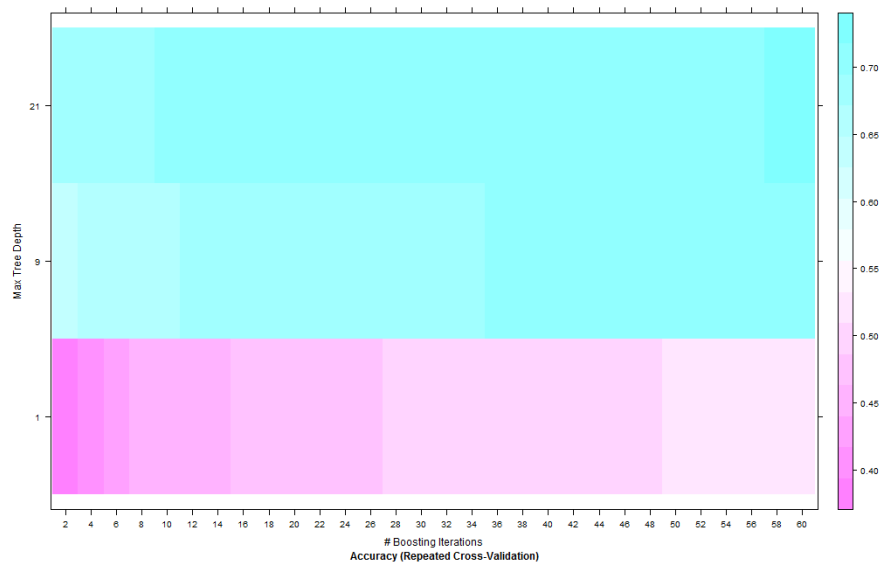


Figure 42 Evaluation of the accuracy vs the complexity of the model

As in the previous analysis with Naïve Bayes algorithm, an attempt has been made with the rush hours of the day during the morning. For this analysis only the trips between 7 am and 9am have been considered to find a model.

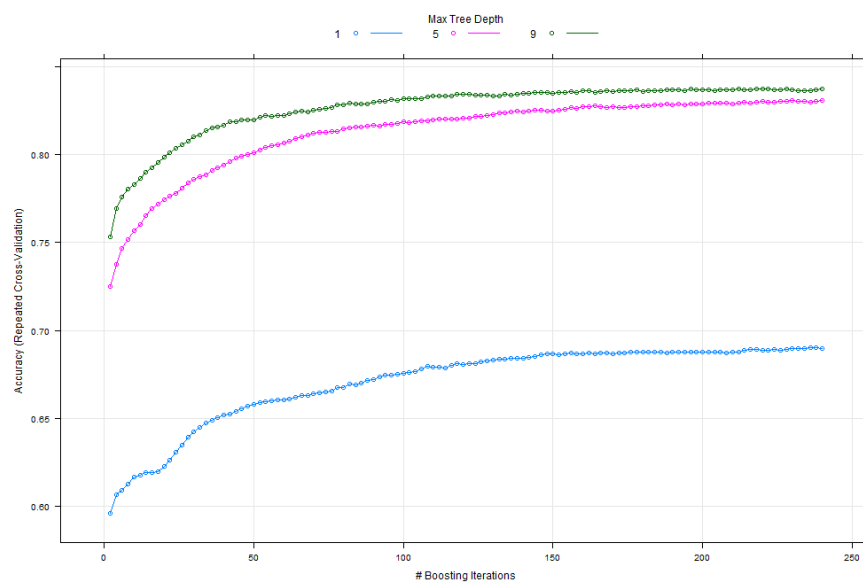


Figure 43 Evolution of the accuracy with the number of trees

The accuracy of the model reaches in this interval of time 84% which is a very high level of precision in the predictions bearing in mind that only three of the seven independent variables have a significant level of importance for developing

the model. It was expected that in this interval of time the accuracy of the model found by the GBM algorithm was larger because performance must be parallel to the Naïve Bayes and it increased in this interval too. (Although all the graphs were plotted they were not include due to space restrictions)

9	238	0.837	0.782	0.0213	0.0283
9	240	0.837	0.782	0.0211	0.028

Tuning parameter 'shrinkage' was held constant at a value of 0.1  
 Accuracy was used to select the optimal model using the largest value.  
 The final values used for the model were n.trees = 218, interaction.depth = 9 and sl

Figure 44 Accuracy result with the model

The image below shows the standard deviation for the accuracy and the kappa and it proves that for most of the sets of data the prediction is quite close to the 0.84 of accuracy which means that most of the values are predicted with that accuracy.

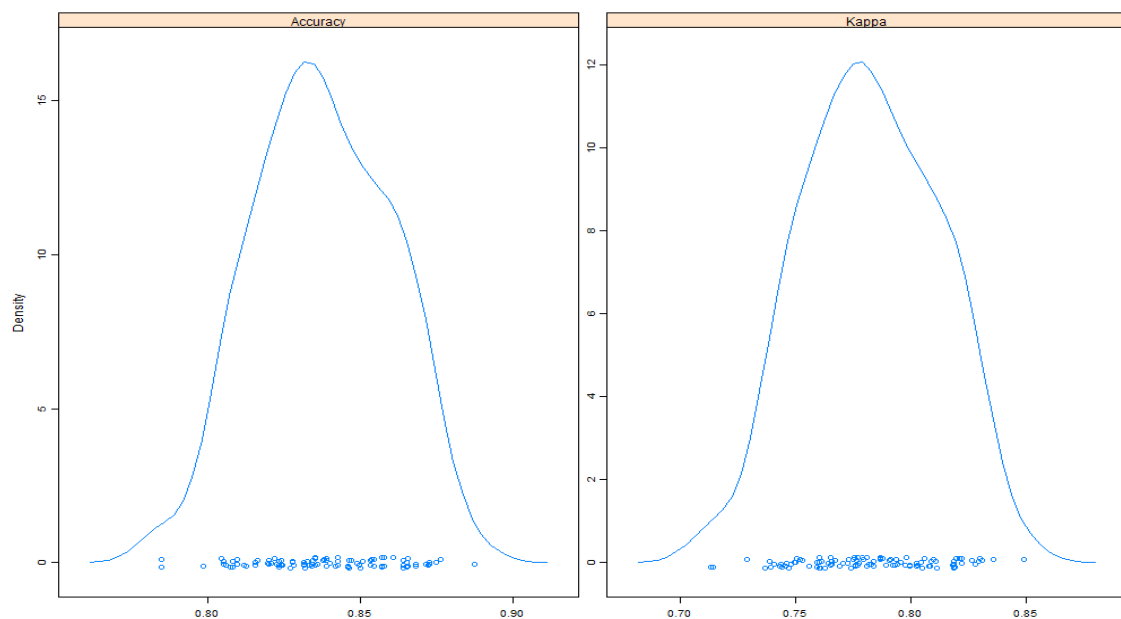


Figure 45 Density plots of the 200 bootstrap estimates the accuracy and Kappa

During the different analysis carried out in this project it was proven that the accuracy of the model depends on the days of the week and on the hours of the day. To find out how this algorithm performs over the 24 hours of the day a new

program was created to run the algorithm in a function called for a loop. The code of the program can be found both in the appendix and in the share folder.

This program was run for both work and weekend days in order to make a comparison of how the model would fit to the different patterns of movements. As a result the two graphs below were obtained. In both cases the shape of the accuracy during the hours of the day is very similar to the one of the amount of trips per hour. It is due to two main reasons. The first one is the type of movements that take place in the different hours of the day and the number of rows that the algorithm has to produce the model.

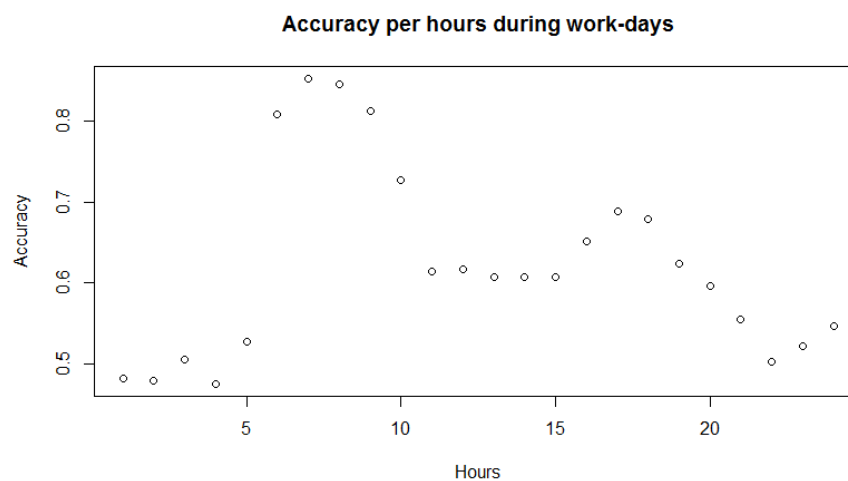


Figure 46 Accuracy per hours during the work days

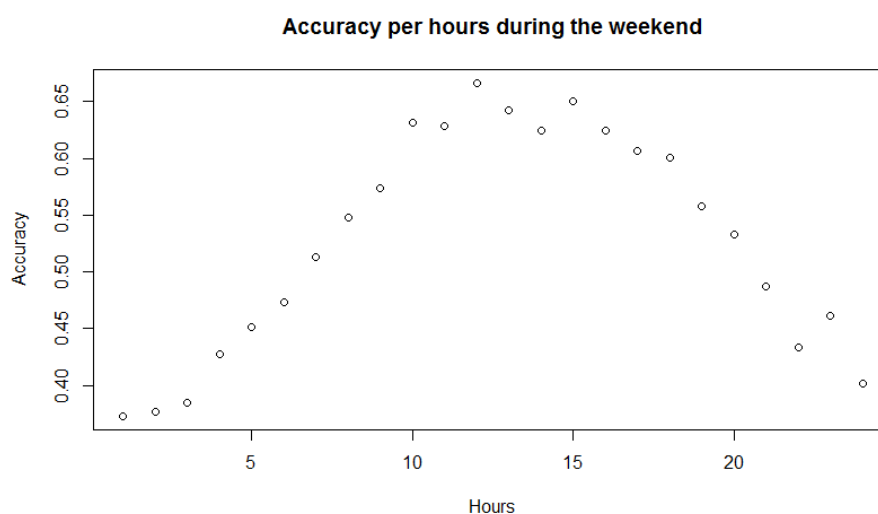


Figure 47 Accuracy per hours during the weekend days

#### **4.4.4 Results**

The Gradient Boosting Machine algorithm has performed very well for this particular dataset achieving an accuracy of 84%. Not only that, this package lets the model be saved and used in a posterior prediction. (It has been used in the testing of the models. An example is in next analysis). Another important result from the hundreds of times that those models have been run is the value that the different parameters of the grid fit can optimize the accuracy and the time that the computer spends processing it. The depth of the tree should not be more than 10 and with values 3, 5 and 9 it is more than enough. It would be worthless for the number of trees to be larger than 60 although in some of the examples shown here that number was 200. It is clear, in the same way that in the previous analysis with another algorithm that the accuracy of the model depends on the hours of the day. The odd thing about this dependency is that if the algorithm is performed in the entire dataset in one go the variable “start hour” has near zero importance in the model.

### **4.5 Analysis 3: Estimation with the User ID**

One of the requirements established at the beginning of this project was the need to have the User Id in the dataset. It would be expected that the accuracy of the model would be improved. Because it was not possible to obtain this data despite several attempts, the researcher made an approximation that gives an idea of how the algorithm could perform if the dataset had contained the User Id.

#### **4.5.1 Research question**

- How well would GBM perform if the dataset contained the user ID?

#### **4.5.2 Description**

After having an in-depth look at the dataset it was found that some users lie about their age or mistype their year of birth. No matter the reason, it gives the opportunity to study the performance of the algorithm in a sub-dataset from a different point of view.



The foundation of this approach is this: There are two users with an age of 114 years, one is female and the other is male. After studying in detail all the movements of the male it can be concluded where he lives, works and the patterns of all his movements around the city. Going a step further but without having the User Id, the sub-dataset is going to be widened. Because there are several trips made by users with an age higher than 100 years the number of users is lower and the algorithm could extract the patterns more easy than in other parts of the dataset.

The same Supervised Machine Learning algorithm “Gradient Boosting Machine” than in the previous analysis is going to be used in this new analysis. It has been concluded that the algorithm performs very well in a sub-dataset that includes only eight stations but in this dataset the number of stations will be around 140 but with patterns easier to identify.

The two images and the table that describe the movements of the 114 years old user are in the appendix 6.

### **4.5.3 Method**

For this analysis a new program was developed in R which runs a SQL Query to bring up the trips of all subscribers older than 100 years. The next screen shots were taken from the analysis in the interval time of 7 am. It had 372 rows or samples with 6 predictors as independent variables and 138 classes for the independent variable. It would be expected that with this short number of rows and huge number of stations to predict, the performance of the algorithm is going to be very bad. The image below shows the header of the results by console. The most important data for understanding the complexity the analysis and the difficulty of the algorithm has to tackle is underlined in red.

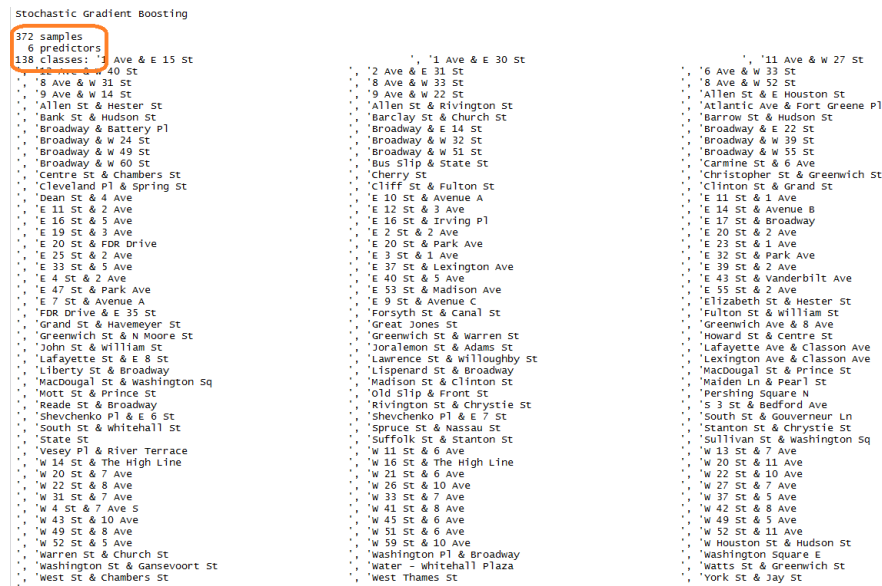


Figure 48 Screen shot of the result

Another very important thing that can be appreciated in this analysis is the change that has been produced in the importance of the independent variable to predict the dependent one. The following picture shows that the age of the user is the second most important variable with a range of 76 over 100. This is the much more similar to what could be expected in a dataset with the user id.

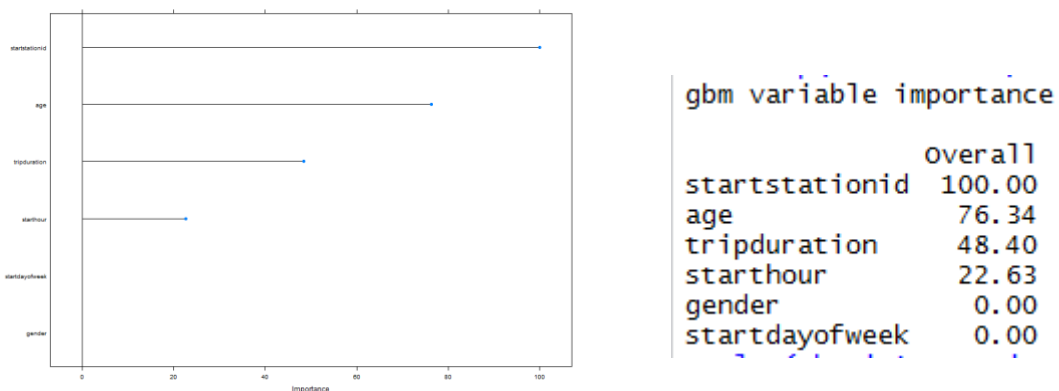


Figure 49 Plot the importance of the independent variable for the model

The image below shows the accuracy of the model in all the iterations for all the trips which started at seven on any of the work days. The detail that should be highlighted from this picture is that the simplest model, with a depth of 1, has better performance than the one with five or nine. It means that the algorithm could take less time simplifying the fit grid with just one depth at the lowest value.

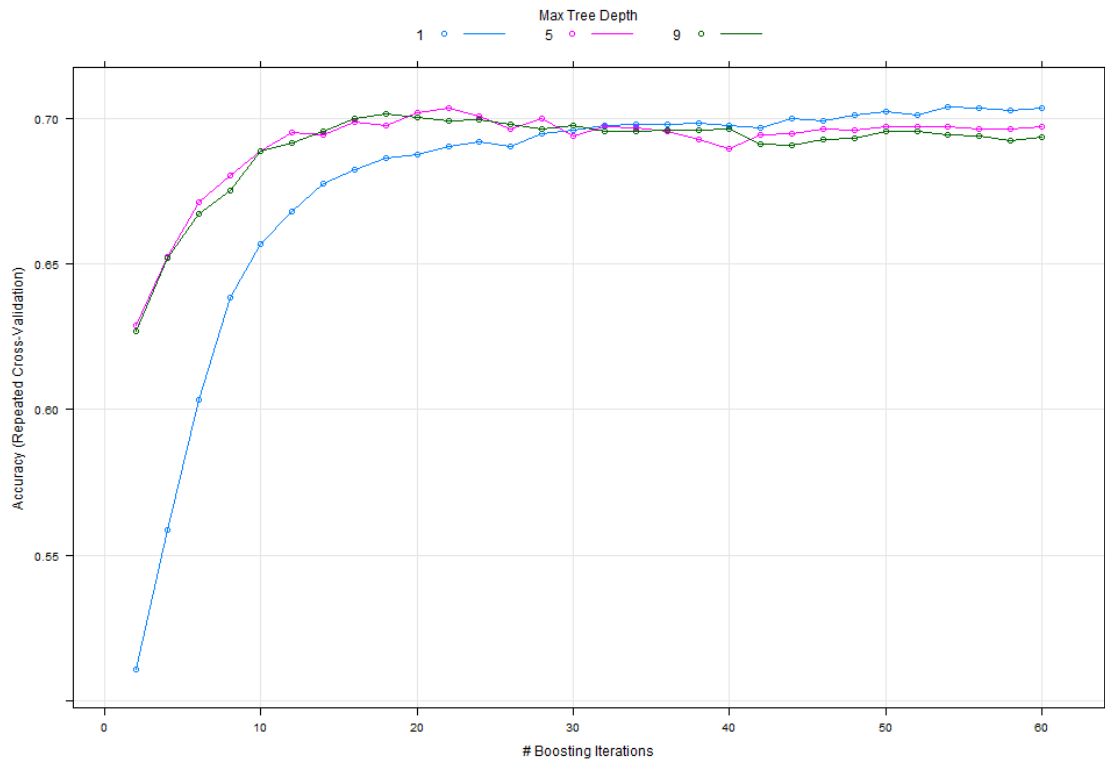


Figure 50 Evolution of the accuracy with the number of trees

Because it is difficult to have the exact value of accuracy in the plot, in the image below a screen shot has been cut with the exact value of the maximum value.

```
> max(model$model$results$Accuracy)
[1] 0.7038974
> |
```

Figure 51 Accuracy r of the model

#### 4.5.4 Results

From this last analysis can be concluded that the algorithm “Gradient Boosting Machine” could performs with unexpected precision if the entire dataset would had the user id.

To appreciate the real value of this accuracy a similar analysis was done with the following characteristics. The dataset has been selected with the same number of rows the same number of classes as the previous one 138 stations and all the trips at the 7am hour.

## 4.6 Analysis 4: Over-fitting and Under-fitting

In Machine Learning there are some cases where the statistical model does not performs very well in new data.

### 4.6.1 Over-fitting

Over-fitting occurs when the machine learning algorithm describes too well the dataset, including errors and noise instead of the underlying trend. It normally happens when the model has too many parameters in relation to the number of observations, in other words the model usually is very complex. It is more common nowadays in the Supervised Machine Learning than under-fitting(University of Washington 2012).

The main measure which was taken during the analysis of the project to detect whether there was a problem of over-fitting or not was to check some of the models to find how well they perform with the testing data which has been split from the training data.

As described earlier on, for each model the dataset was split into two parts. In at least five cases the model was tested with the test-data and in all of them the accuracy of the model was very similar to the accuracy that the model had in the training data.

The image below shows the result of the prediction in the Analysis 2 for the hours (7, 8, 9) in the morning where the model had an accuracy of 83% and the result of the test was 86%. Normally the accuracy is a bit lower in the test dataset than in the training dataset but in this case it is higher. To run the model over the test dataset two functions of the caret package have been used. The first one is the “predict” function which applies the model to the new dataset and the second one is the “confusionMatrix” which compares the prediction with the real result in the test dataset and creates a matrix with the results giving a lot of parameters that measure how well the model predicts over the test dataset.

Note: In the appendix the entire result of these functions have been pasted.

```

> prediction <- predict(gbmFit1, newdata =testing )
> confusionMatrix(prediction, testing$endstationname)
Confusion Matrix and Statistics

Prediction
8 Ave & W 31 St
Christopher St & Greenwich St
Clinton St & Joralemon St
E 33 St & 5 Ave
E 43 St & Vanderbilt Ave
E 7 St & Avenue A
Lafayette St & E 8 St
Warren St & Church St

Reference
8 Ave & W 31 St
103
5
0
0
2
1
4
4

Prediction
8 Ave & W 31 St
Christopher St & Greenwich St
Clinton St & Joralemon St
E 33 St & 5 Ave
E 43 St & Vanderbilt Ave
E 7 St & Avenue A
Lafayette St & E 8 St
Warren St & Church St

Reference
Christopher St & Greenwich St
5
88
0
1
7
1
2
4

Prediction
8 Ave & W 31 St
Christopher St & Greenwich St
Clinton St & Joralemon St
E 33 St & 5 Ave
E 43 St & Vanderbilt Ave
E 7 St & Avenue A
Lafayette St & E 8 St
Warren St & Church St

Reference
Clinton St & Joralemon St
0
0
2
0
1
0
1
0

Prediction
8 Ave & W 31 St
Christopher St & Greenwich St
Clinton St & Joralemon St
E 33 St & 5 Ave
E 43 St & Vanderbilt Ave
E 7 St & Avenue A
Lafayette St & E 8 St
Warren St & Church St

Reference
E 33 St & 5 Ave
0
0
0
0
1
1
0
0

```

Figure 52 Confusion Matrix (1) testing the model

```

E 7 St & Avenue A
Lafayette St & E 8 St
Warren St & Church St
~
0
5
14

Overall Statistics

Accuracy : 0.8664
95% CI : (0.8421, 0.8881)
No Information Rate : 0.4009
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8207
McNemar's Test P-value : NA

Statistics by Class:

Class: 8 Ave & W 31 St
Sensitivity 0.8655
Specificity 0.9647
Pos Pred Value 0.7923
Neg Pred Value 0.9788
Prevalence 0.1348
Detection Rate 0.1166
Detection Prevalence 0.1472
Balanced Accuracy 0.9151

Class: Christopher St & Greenwich St
Sensitivity 0.81481
Specificity 0.97548
Pos Pred Value 0.82243
Neg Pred Value 0.97423

```

Figure 53 Confusion Matrix (2) testing the model

## 4.6.2 Under-fitting

On the other hand, under-fitting occurs when the statistical model cannot describe the underlying relationship of the dataset. It occurs when the model does not fit well enough and it will perform a poor prediction on the testing data.

Under-fitting is often a result of a too simple model which is not able to describe the trend of the dataset. In the case of this project we have two possible causes of an under-fitting model.

- The lack of predictors or independent variables to build the model. As it can be appreciated in the images below there are only seven predictors but only two of them are very important for fitting the model.

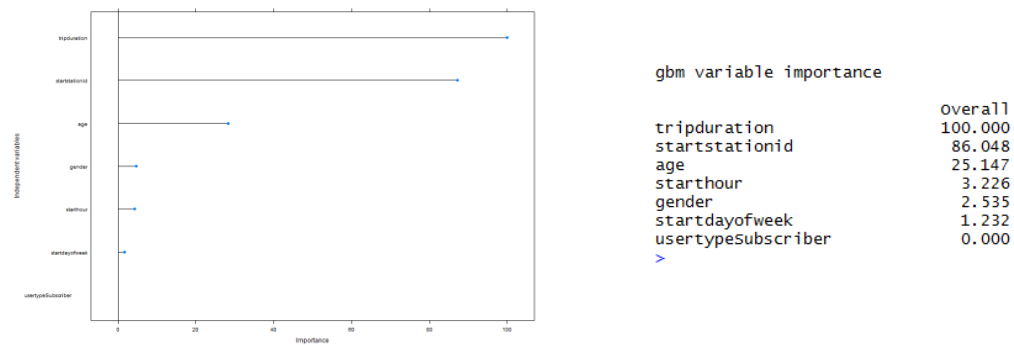


Figure 54 Importance of the independent variable for the model

- The number of rows is not enough to complete the whole process of fitting the model for some cases. (During the night time there are not many trips and the model by hours cannot fit very well and the accuracy decreases) To solve this problem two hours were studied together at the same time. A screen shot of the error can be seen in the picture below.

```
> # Apply the supervised ML algorithm to learn and come up with a model
> gbmFit1 <- train(endstationname ~ ., data = training,
+                 method = "gbm",
+                 trControl = fitControl,
+                 ## This last option is actually one
+                 ## for gbm() that passes through
+                 verbose = FALSE, tuneGrid = gbmGrid)
Error in train.default(x, y, weights = w, ...) :
  final tuning parameters could not be determined
```

Figure 55 Error for having few rows

To analyse whether the model was giving a shorter accuracy as a consequence of the low number of rows in the same dataset the following process was carried out. As usual, the first analysis was carried out with 75% of the dataset and left the other 25% for testing. With the same set of data the analysis was carried out again with the whole dataset. The result, as it can be appreciated, is better in the case where 100% of the rows to train the model were used. The table with the results is in the appendix.

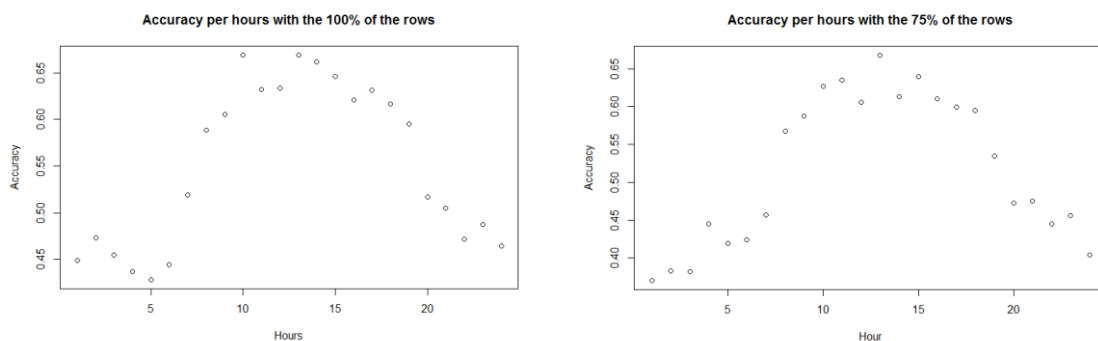


Figure 56 Comparison the accuracy of the model with 75% of the rows and 100% (1)

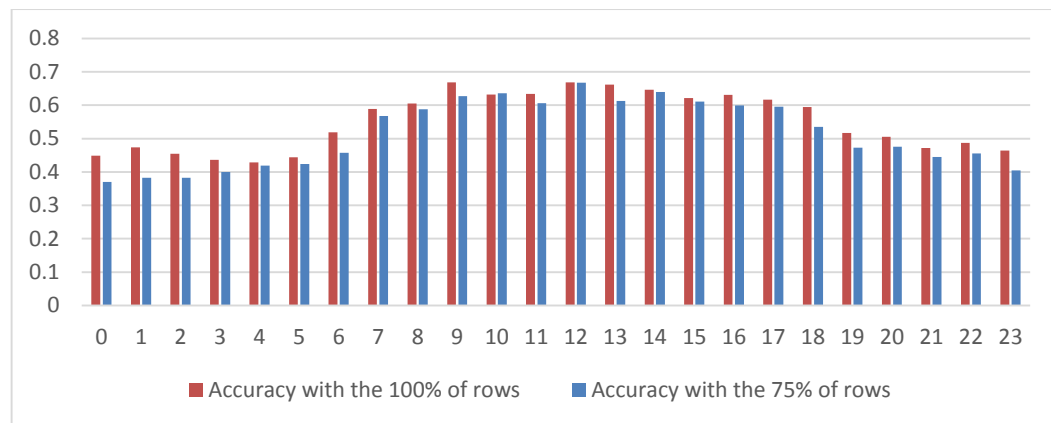


Figure 57 Comparison the accuracy of the model with 75% of the rows and 100% (2)

## 4.7 Final Prototype

As a final objective is to apply the results of this project, offering a service to users of the Share Bikes Scheme that gives them a prediction of whether a bike is coming to the station where they are waiting and the time that bike will take to appear.

It has been proved that finding a model to predict the final station of a trip is possible. As well, it has been proved that the accuracy is very large and it depends on the time of the day, the day of the week and the trip duration (from the independent variables that our dataset has). As an estimation, a different analysis was carried out to prove that the accuracy of the model could be increased if the user ID was included in the dataset. For the final prototype has been taken into consideration that the user Id is going to be available both because it has to be developed with the owner of the Share Bikes Schemes or the data has to be bought from the owner.

The good thing about this system is that the model could be calculated in advance and the prediction would take just seconds so the system could come up with a simulation of the situation very quickly.

The system will store all the models in a database. Models for specific users and for hours of the work day and weekend days will be developed every weekend. Once the application has available all the models the system will be ready to attend requests. After an estimation is requested the system will answer with the following steps: firstly it will run a query to the real time database and will bring

all the data of the bikes which have started a trip and have not finished it yet. With one row per bike the system will have some of the independent variables like the start hour, the id of the station where the journey was started, the user id and all the user's data.

The system does not have include the trip duration and it is one of the most important independent variables to predict the dependent variable, in our case, the Id of the final station. With each row that the system has in return from the real time database the system will create a bunch of 30 rows. In each of the new rows the trip duration will be the difference between the time it started and the time that the prediction is requested plus 20 seconds in each. The first will have the difference of times, the second will have the difference of times plus 20 seconds, the third will have the difference of times plus 40 seconds and so on up to ten minutes in advance.

In order to be used for prediction, the system will upload the correspondent model for the specific user and for the hour within the request has been made. For each of those new rows that have been created the system will come up with a prediction using the model it has stored before. If any of the predictions match with the station where the request was demanded, then the system will determine the time that the bike will be there with the seconds that have been added.

The system will be refreshing the data each twenty seconds in order to remove the trips that have been finished and to add and recalculate the new trips that have been started.



## 5 Conclusions

The analysis carried out in this project shows a number of interesting findings that could be applied in the real world. Although the movements of bikes seem to be very random and chaotic it has been proven that there are a lot of patterns that the customers and the subscribers follow to move around the city.

Understanding all those patterns and predicting movements could make a significant impact on the improvement of the quality of the service and on the accurate prediction of availability in the short term. This accuracy will help to enhance sustainable urban share bikes schemes into the medium and big cities.

What has been proven in this project is that the machine learning algorithms, and more specifically the “Gradient Boosting Machine” algorithm, could be used satisfactorily to predict movements of people in advance, based on past movements, once the trip has started.

The analysis has demonstrated that the accuracy of the prediction is different along the hours of the day. The shape of that graph is very similar to the shape of the use of bikes per day of the week. As it could be expected, the accuracy is larger when the movements in the city (as well with the bikes) are more daily routine, for instance in the mornings to commute, and in the afternoon to go back home. On the other hand we have more random movements during midday due to tourist and sporadic users that give us a shorter accuracy in the model’s predictions.

The study has proven that the key independent variables that the algorithm uses to predict the destination station, the dependent variable, are: the “Start station”, the “trip duration” and in a lower percentage the “age” and the “hour of the day” depending on the model that wants to be found.

In this particular project and for instance in the dataset used for its purpose it has been proven that the method that the algorithm uses to validate the model or resampling (Cross-Validation) during the training period performs very well. In most of the cases checked, the accuracy has been very similar to the cross-validation. It could be concluded that for further analysis the entire dataset could

be used for training the model instead of splitting it without caring about the over-fitting and reducing the possibility of under-fitting, obtaining a larger accuracy.

The study of the dataset has shed light on the different type of people who use the bikes per station and a deeper study could be carried out on the different movements depending on the age, gender and the place where they live or work.

Although having the Id of the user in the dataset was one of the requirements it was impossible to obtain it. This project wanted to demonstrate that the user id would be key for further analysis so in the fourth analysis it has been proven in a small part of the dataset that although the number of classes to predict rises to 136 the accuracy of the model decreases slightly to 71%. Therefore, if the dataset had the user id the accuracy of the model could be around the 95%.

## 6 Further developments

Following this project there are several points in which further developments should focus on.

### 6.1 Improving the current analysis

During this project there have been several limitations of time, investment and resources that the scope had to be reduced and leave out of scope some important points. It would be recommended for the future:

- Going hand in hand with the National College of Ireland to reach an agreement with the owner of the data and with a signed contract of confidentiality get the entire dataset with the Id of the users anonymized.
- Develop a model with the entire dataset in a cloud server (Amazon or Windows Azure) that facilitates a real view of the accuracy with all the possible independent variables and all the possible classes or stations for the dependent variable.
- It will be very important to have in real time the availability or not of bikes or docks in any station in order to improve the accuracy of a real model. You can predict that a bike is going to finish in a station but if there are no available docks it is going to be a wrong prediction.
- Study how events like concerts, theatres, cinemas, season reductions and all kinds of events that happen around the city affect the patterns of movements in the share bikes scheme.
- Study how the lack of bikes or docks affects the punctuality of the people who rely on this transportation system to improve the scheme and minimize the number of times it happens.

### 6.2 Develop a prototype to offer some companies

From the practical point of view it is clear that the findings of this research have a real business application. It would be recommended in order to have a return on investment of this project and further developments to:

- Develop a small prototype, which would not take more than one week with the programs that have already been developed, and offer it to some companies which run similar businesses around the world and who may be interested in offer a better service to their customers with this new feature.
- Offer over this service some type of advertisement for restaurants, bars, shops, cinemas, theatres or business around the area where the user is going to finish his/her trip. Because the application knows in advance where the user is going to finish his/her trips and therefore the area where the user is going to be, it would be possible to show in the application some advertisements of business from that particular area.

## 7 References

AAAI (2009) ‘Sensing and Predicting the Pulse of the City through Shared Bicycling’. [Online]. *Association for the advancement of artificial intelligence*. Available from: [www.aaai.org/ocs/index.php/IJCAI/IJCAI-09/paper/download/578/910](http://www.aaai.org/ocs/index.php/IJCAI/IJCAI-09/paper/download/578/910) [Accessed 29<sup>th</sup> June 2014].

Cran r-Project (2014) ‘Generalized Boosted Regression Models’. [Online]. *Cran.r-Project* Available from: <http://cran.r-project.org/web/packages/gbm/index.html> [Accessed 14<sup>th</sup> June 2014].

Eric Cai (2014) ‘Machine Learning Lesson of the Day – Cross-Validation’. [Online]. *The Chemical Statistician*. Available from: <http://chemicalstatistician.wordpress.com/2014/01/17/machine-learning-lesson-of-the-day-cross-validation/> [Accessed 19<sup>th</sup> July 2014].

Ian C. (2014) ‘Running R on the Amazon Cloud’. [Online]. *Slide Share*. Available from: <http://www.slideshare.net/ianmcook/running-r-on-the-amazon-cloud-2013-0620> [Accessed 24<sup>th</sup> July 2014].

IEEE (2013) ‘Uncertainty in urban mobility: Predicting waiting times for shared bicycles and parking lots’. [Online]. *IEEE Explore digital library* <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6728210&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel7%2F6712176%2F6728201%2F06728210.pdf%3Farnumber%3D6728210> [Accessed 24<sup>th</sup> June 2014].

Journal of Statistical Software (2008) ‘Building Predictive Models in R using the caret package’. [Online]. *Journal of Statistical Software*. Available from: <http://www.jstatsoft.org/v28/i05> [Accessed 28<sup>th</sup> June 2014].

Project Euclid (2017) ‘Boosting Algorithms: Regularization, Prediction and Model Fitting’. [Online]. Project Euclid mathematics and statistics online. Available from: <http://projecteuclid.org/DPubS?verb=Display&version=1.0&service=UI&handle=euclid.ss/1207580163&page=record> [Accessed 21<sup>st</sup> June 2014].

R Project Org. (2014) ‘the caret package’. [Online]. *The caret package*. Available from: <http://caret.r-forge.r-project.org/training.html> [Accessed 13<sup>th</sup> July 2014]

Ridgeway G. (2003) ‘The Comprehensive R Archive Network’ [Online]. *Cran.r-Project* Available from: <http://cran.r-project.org/web/packages/gbm/index.html> [Accessed 14<sup>th</sup> June 2014].

StatSoft Company (2013) ‘StatSoft Electronic Statistics Textbook’. [Online]. *Introduction to Boosting Trees for Regression and Classification*. Available from: <http://www.statsoft.com/Textbook/Boosting-Trees-Regression-Classification> [Accessed 15<sup>th</sup> July 2014].

University of Washington (2012) 'Linear Regression Bias Variance Tradeoff'. [Online]. University of Washington Available from: <http://courses.cs.washington.edu/courses/cse546/12wi/slides/cse546wi12LinearRegression.pdf> [Accessed 26<sup>th</sup> July 2014].

World Scientific Publishing Company (2011) 'Shared bicycles in a City: A signal processing and data analysis perspective' *World Scientific Connecting Great Minds* Available from: <http://www.worldscientific.com/doi/abs/10.1142/S0219525911002950> [Accessed 27<sup>th</sup> June 2014].

## 8 Literature

Cortinhas, C. and Black, K. (2012) *Statistics for Business and Economics*, 1st European Edition Ed., John Wiley & Sons

Alex Smola and S.V.N. Vishwanathan (2008) 'Introduction to Machine Learning' 1<sup>st</sup> Edition. Ed., Cambridge University Press.

## 9 Appendix

This section contains programs and other material that have been used for this project. Most of this appendix are images which do not make an impact on the number of words. Just the last one had to be by text and it has around 700 words.

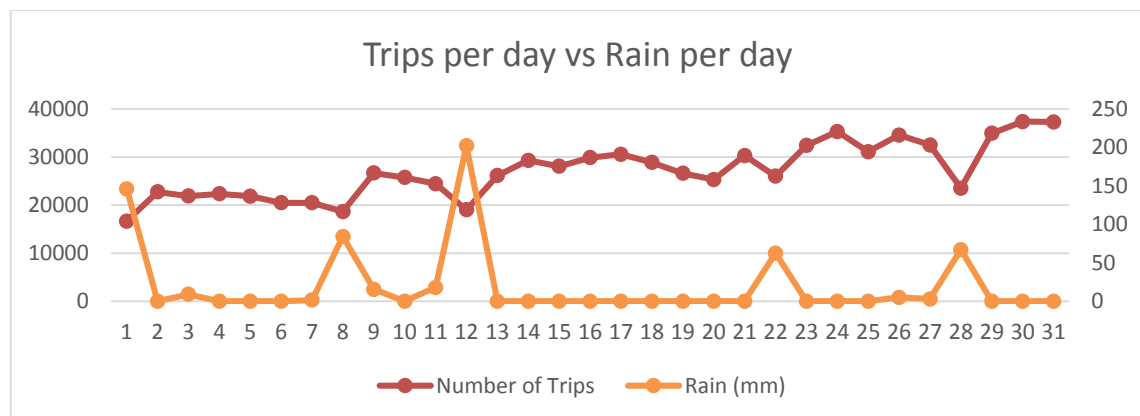
### 9.1 Appendix 1 Online Repository OneDrive

As requested, all the files used for the purpose of this research have been made available on the cloud. Please, find below the link to the repository.

<https://onedrive.live.com/redir?resid=2FBCBC80E4A1444C!4500&authkey=!ANzaeovxEH8PtJY&ithint=folder%2c>

### 9.2 Appendix 2 Study of Correlation between rain & trips

Because finally this analysis was removed from the scope of the project but was already done, it has been included here for the sake of information. It was removed for two main reasons, lack of time and evidence of the correlation without analysis.



Correlation between Number of trips and Weather

```
> # correlations
> data = read.csv("Tripsweatherv1.csv", header=TRUE, sep=",")
> cor(data[c("NumberTrips", "TMax", "TMin", "Rain", "Snow")])
```

	NumberTrips	TMax	TMin	Rain	Snow
NumberTrips	1.0000000	0.76518391	0.761001418	-0.205682914	-0.3716160
TMax	0.7651839	1.00000000	0.975030169	-0.023021776	-0.2937024
TMin	0.7610014	0.97503017	1.000000000	-0.005524572	-0.2883228
Rain	-0.2056829	-0.02302178	-0.005524572	1.000000000	0.1660353
Snow	-0.3716160	-0.29370238	-0.288322761	0.166035262	1.0000000

```
> summary(model)
```

```
Call:
```

```
lm(formula = NumberTrips ~ TMax + TMin + Rain + Snow, data = data)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-17579.0 -4288.9   291.6   3871.6  20597.9
```

```
Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13783.964   1145.690   12.031 < 2e-16 ***
TMax         331.917    131.947    2.516 0.012361 *
TMin         323.384    134.871    2.398 0.017055 *
Rain        -15.432     2.983   -5.173 4e-07 ***
Snow        -61.353    16.543   -3.709 0.000245 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 5917 on 329 degrees of freedom
```

```
Multiple R-squared:  0.6425, Adjusted R-squared:  0.6382
```

```
F-statistic: 147.8 on 4 and 329 DF,  p-value: < 2.2e-16
```

From the data obtained in the analysis it is pretty clear that the rain and the snow have a direct impact on the use of bikes as could be expected.

### 9.3 Appendix 3 Table of values comparing the accuracy in under-fitting

As a test for proving that the Machine Learning algorithm could under-fitting when the number of rows is low, a comparison was carried out between the same dataset training the model with 75% of the rows, and 100% of the rows.

Hours	Accuracy with the 100% of rows	Accuracy with the 75% of rows	Number of Rows 100%	Number of Rows 75%
0	0.448343505	0.370554557	133	107
1	0.473357171	0.382710317	101	84
2	0.454907925	0.382281746	101	84
3	0.436496004	0.399492244	101	84
4	0.42827381	0.419243986	57	49
5	0.444186508	0.424074074	57	49
6	0.518833333	0.457312925	58	50
7	0.588739372	0.567787879	113	92
8	0.60521499	0.587927244	273	214
9	0.668677846	0.626857749	482	371
10	0.631788976	0.635451382	607	464
11	0.633396003	0.605947168	653	501
12	0.668803105	0.667720287	691	527
13	0.661224875	0.612994188	741	565



14	0.646329485	0.639243308	761	581
15	0.620930577	0.610570945	749	572
16	0.630955853	0.599202125	725	553
17	0.61667535	0.595199996	635	486
18	0.59475576	0.534944019	489	375
19	0.51688275	0.47265381	380	295
20	0.505116829	0.475519593	308	240
21	0.471616419	0.444988597	254	199
22	0.487001261	0.455699418	194	155
23	0.463959087	0.404190809	155	123

Table 4 Accuracy in the same dataset training the model with 75% of rows and 100%

## 9.4 Appendix 4 Code of the programs (Images)

In this Appendix has been included some of the most important programs used for the purpose of this project. Note: It has been decide to include them as image in order to not increase the number of words. Only five of the 27 programs have been included due to them are very similar but they are in the shared repository with the rest of the programs.

“Naïve Bayes Work-days 7-9am.R”

```

1
2- #####
3 # program to apply Naive Bayes in the stations during one day #
4- #####
5
6 #Load the libraries the program is going to use
7
8 library("kIAR")
9 library("caret")
10 library(RODBC)
11 #Open the ODBC connection to bring the data
12 channel1 <- odbcConnect("SQLodbc")
13 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
14 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
15 (143,152, 186, 293,318,358,432, 521, 526)"
16 # run the query
17 tripsbetweenstations=sqlQuery(channel1,querystations)
18 #Define some variable to filter more the data
19 morning = c(0,8,9)
20 afternoon = c(17,18,19)
21 day = c(0:23)
22 workday = c(2,3,4,5,6)
23 weekend = c(1,7)
24 #Divide the dataset into the independent variables and the dependent variable
25 xTrain = tripsbetweenstations[,c(1:7)]
26 yTrain = tripsbetweenstations[,8]
27 #Run the model with the package caret and the funtion train
28 model = train(xTrain,yTrain,'nb', trControl=trainControl(method='cv', number=12))
29 write.csv(model$results, file="model.csv")
30 model
31
32 xTrain = tripsbetweenstations[tripsbetweenstations$starthour %in% day
33 & tripsbetweenstations$startdayofweek %in% workday,c(1,2,3,4,5,6,7)]
34 yTrain = tripsbetweenstations[tripsbetweenstations$starthour %in% day
35 & tripsbetweenstations$startdayofweek %in% workday,8]
36 model = train(xTrain,yTrain,'nb', trControl=trainControl(method='cv', number=6))
37 model
38
39 #xTest = bikes[170:194,c(1,2,3,4,6,7)] #c(2,3,4,6)
40 #yTest = bikes[170:194,8]
41 #prop.table(table(predict(model$finalModel,xTest)$class,yTest))

```

“Naïve Bayes whole day.R”

```

1
2- #####
3 # program to apply Naive Bayes in the stations during one day #
4- #####
5
6 #Load the libraries the program is going to use
7
8 library("kIAR")
9 library("caret")
10 library(RODBC)
11 #Open the ODBC connection to bring the data
12 channel1 <- odbcConnect("SQLodbc")
13 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
14 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
15 (143,152, 186, 293,318,358,432, 521, 526)"
16 # run the query
17 tripsbetweenstations=sqlQuery(channel1,querystations)
18 #Define some variable to filter more the data
19 morning = c(0,8,9)
20 afternoon = c(17,18,19)
21 day = c(0:23)
22 workday = c(2,3,4,5,6)
23 weekend = c(1,7)
24 #Divide the dataset into the independent variables and the dependent variable
25 xTrain = tripsbetweenstations[,c(1:7)]
26 yTrain = tripsbetweenstations[,8]
27 #Run the model with the package caret and the funtion train
28 model = train(xTrain,yTrain,'nb', trControl=trainControl(method='cv', number=12))
29 write.csv(model$results, file="model.csv")
30 model
31
32 xTrain = tripsbetweenstations[tripsbetweenstations$starthour %in% day
33 & tripsbetweenstations$startdayofweek %in% workday,c(1:7)]
34 yTrain = tripsbetweenstations[tripsbetweenstations$starthour %in% day
35 & tripsbetweenstations$startdayofweek %in% workday,8]
36 model = train(xTrain,yTrain,'nb', trControl=trainControl(method='cv', number=6))
37 model
38
39 #xTest = bikes[170:194,c(1,2,3,4,6,7)] #c(2,3,4,6)]
40 #yTest = bikes[170:194,8]
41 #prop.table(table(predict(model$finalModel,xTest)$class,yTest))

```

## “GradientBoostingMachine Work-Days loop 0-23.R”

```

1
2 library(caret)
3 library(RODBC)
4- #####
5 # FUNCTION STARTS HERE
6- #####
7 ftrain <- function(iI, bikesf){
8   print(iI)
9
10-   if (iI == 23) {
11     hours <- c(iI, 0)
12-   }else if (iI == 1 || iI == 2 || iI == 3) {
13     hours <- c(1,2,3)
14-   }else{
15     hours <- c(iI, iI+1)
16   }
17   bikeswork = bikesf[bikesf$starthour %in% hours,c(1,2,4,6,7,8)] #c(2,3,4,6)]
18   #Define the grid that the algorithm is going to use for tuning the model
19   gbmGrid <- expand.grid(interaction.depth = c(1, 5, 9),
20                         n.trees = (1:30)*2,
21                         shrinkage = 0.1)
22   nrow(gbmGrid)
23   set.seed(14*(iI+1))
24   #Create a partition with data for the model and the rest for testing the model. 75% for the model and 25% for testing
25   inTraining <- createDataPartition(bikeswork$endstationname, p = 0.75, list = FALSE)
26   training <- bikeswork[inTraining, ]
27   testing <- bikeswork[-inTraining, ]
28   #The method used to control the learning is Cross-validation
29   fitControl <- trainControl(method = "repeatedcv",number = 10, repeats = 10)
30   set.seed(7*(iI+1))
31   # Apply the supervised ML algorithm to learn and come up with a model
32   gbmFit1 <- train(endstationname ~ ., data = training,
33                  method = "gbm",
34                  trControl = fitControl,
35                  ## This last option is actually one
36                  ## for gbm() that passes through
37                  verbose = FALSE ,tuneGrid = gbmGrid)
38   # Returns the results to the loop
39   graphf <- plot(gbmFit1)
40   total <- list("model"=gbmFit1,"plot"=graphf)
41   return(total)
42 }

```

```

43 - #####
44 # PROGRAM STARTS HERE
45 #####
46 # Connect to the SQLServer
47 channel1 <- odbcConnect("SQLodbc")
48 # Define the query with the four stations
49 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
50 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
51 (143,152, 186, 293,318,358,432, 521, 526)"
52
53 # Run the query
54 bikes=sqlQuery(channel1,querystations)
55
56
57 workday = c(2,3,4,5,6)
58
59 # Filter the dataset with the work-days only
60 bikesfilter = bikes[bikes$startdayofweek %in% workday,c(1,2,3,4,5,6,7,8)] #c(2,3,4,6)]
61 listmodels = list()
62 u = c(0)
63 for (ij in 0:23) {
64   listmodels[ij+1] <- ftrain(ij, bikesfilter)
65   u = rbind(u,max(listmodels[[ij+1]]$results$Accuracy))
66 }
67
68 max(model$model$result$Accuracy)
69
70
71 #Show the results in graphs and in text
72 gbmFit1
73 plot(gbmFit1)
74 plot(varImp(gbmFit1), xlab="Importance", ylab="Independent variables")
75 plot(gbmFit1, metric = "Kappa")
76 plot(gbmFit1, plotType = "level")
77 varImp(gbmFit1)
78 resampleHist(gbmFit1)

```

### “GradientBoostingMachine Work-Days 7-9.R”

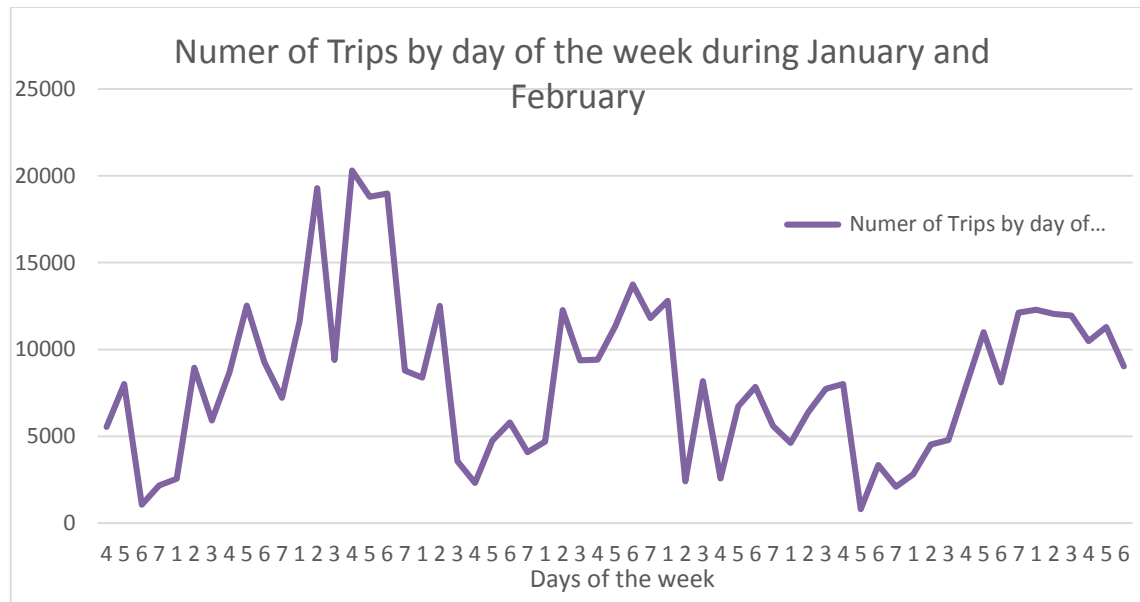
```

2 library(caret)
3 library(RODBC)
4
5 # Connect to the SQLServer
6 channel1 <- odbcConnect("SQLodbc")
7 # Define the query with the four stations
8 querystations = "SELECT tripduration, startdayofweek, starthour, startstationid, usertype, age, gender, endstationname
9 FROM TRIPSTEST WHERE startstationid in (143,152, 186, 293,318,358,432, 521, 526) and endstationid in
10 (143,152, 186, 293,318,358,432, 521, 526)"
11
12 # Run the query
13 bikes=sqlQuery(channel1,querystations)
14
15 hours= c(7:9)
16 workday = c(2,3,4,5,6)
17
18 # Filter the dataset with the work-days only
19 bikesfilter = bikes[bikes$starthour %in% hours & bikes$startdayofweek %in% workday,c(1,2,3,4,5,6,7,8)] #c(2,3,4,6)]
20 #Define the grid that the algorithm is going to use for tuning the model
21 gbmGrid <- expand.grid(interaction.depth = c(1, 5, 9),
22                       n.trees = (1:60)*2,
23                       shrinkage = 0.1)
24 nrow(gbmGrid)
25 set.seed(998)
26 #Create a partition with data for the model and the rest for testing the model. 75% for the model and 25% for testing
27 intraining <- createDataPartition(bikesfilter$endstationname, p = 0.75, list = FALSE)
28 training <- bikesfilter[intraining, ]
29 testing <- bikesfilter[-intraining, ]
30
31 #The method used to control the learning is Cross-validation
32 fitControl <- trainControl(method = "repeatedcv",number = 10, repeats = 10)
33
34 set.seed(825)
35 # Apply the supervised ML algorithm to learn and come up with a model.
36 gbmFit1 <- train(endstationname ~ ., data = training,
37               method = "gbm",
38               trControl = fitControl,
39               verbose = FALSE ,tuneGrid = gbmGrid)
40 #Show the results in graphs and in text
41 gbmFit1
42 plot(gbmFit1)
43 plot(varImp(gbmFit1), xlab="Importance", ylab="Independent variables")
44 plot(gbmFit1, metric = "Kappa")
45 plot(gbmFit1, plotType = "level")
46 varImp(gbmFit1)
47 resampleHist(gbmFit1)

```

## 9.5 Appendix 5 Plot of the trips by day of the week

It is clear in this graph that there is no pattern in the amount of movements between weeks of the months.



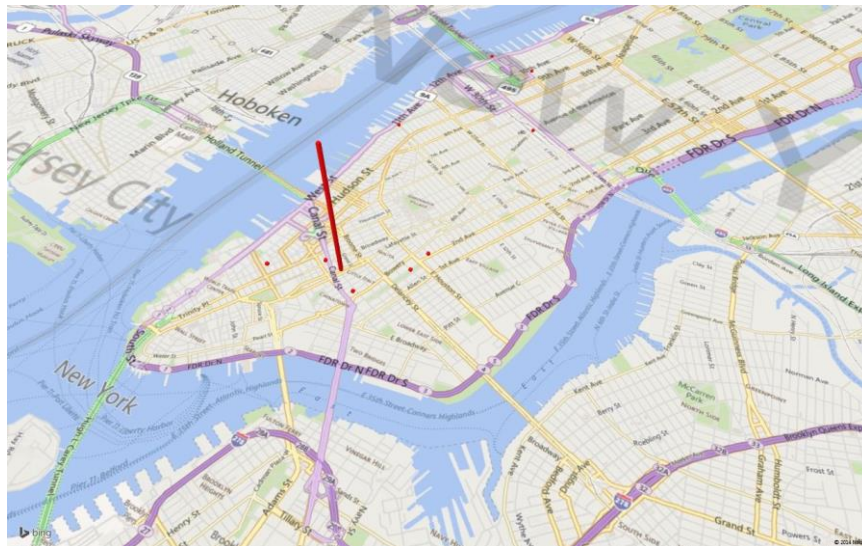
## 9.6 Appendix 6 Patterns of movements user 114 years old.

This is a simple overview about the patterns that having the user id could be found from the users movements.

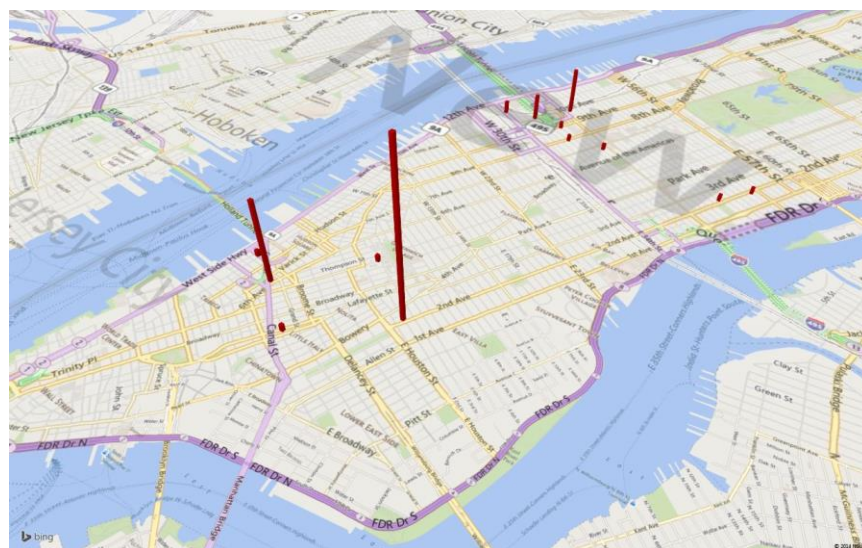
		D-A	Name	Latitude	Longitude	Quantity
Arrivals 7,8,9 in the morning	12 Ave & W 40 St	2	12 Ave & W 40 St	40.76087502	74.00277668	1
	Broadway & W 32 St	2	Broadway & W 32 St	40.74854862	73.98808416	1
	E 2 St & 2 Ave	2	E 2 St & 2 Ave	40.72502876	73.99069656	1
	Elizabeth St & Hester St	2	Elizabeth St & Hester St	40.71729	-73.996375	1
	Howard St & Centre St	2	Howard St & Centre St	40.71910537	73.99973337	61
	Hudson St & Reade St	2	Hudson St & Reade St	40.71625008	-74.0091059	1
	Lispenard St & Broadway	2	Lispenard St & Broadway	40.71939226	74.00247214	1
	Stanton St & Chrystie St	2	Stanton St & Chrystie St	40.72229346	73.99147535	1
	W 14 St & The High Line	2	W 14 St & The High Line	40.74195138	74.00803013	1
	W 43 St & 10 Ave	2	W 43 St & 10 Ave	40.76009437	73.99461843	1
Departures 7,8,9 in the morning	12 Ave & W 40 St	1	12 Ave & W 40 St	40.76087502	74.00277668	4
	6 Ave & Canal St	1	6 Ave & Canal St	40.72243797	74.00566443	12
	Broadway & W 41 St	1	Broadway & W 41 St	40.75513557	73.98658032	1
	E 2 St & 2 Ave	1	E 2 St & 2 Ave	40.72502876	73.99069656	24
	E 47 St & 2 Ave	1	E 47 St & 2 Ave	40.75323098	73.97032517	1
	E 52 St & 2 Ave	1	E 52 St & 2 Ave	40.756014	-73.967416	1
	Howard St & Centre St	1	Howard St & Centre St	40.71910537	73.99973337	1

LaGuardia Pl & W 3 St	1	LaGuardia Pl & W 3 St	40.72917025	73.99810231	1
W 34 St & 11 Ave	1	W 34 St & 11 Ave	40.75594159	-74.0021163	2
W 37 St & 10 Ave	1	W 37 St & 10 Ave	40.75660359	-73.9979009	4
W 38 St & 8 Ave	1	W 38 St & 8 Ave	40.75466591	73.99138152	1
W 39 St & 9 Ave	1	W 39 St & 9 Ave	40.75645824	73.99372222	1
W 43 St & 10 Ave	1	W 43 St & 10 Ave	40.76009437	73.99461843	7
Watts St & Greenwich St	1	Watts St & Greenwich St	40.72405549	74.00965965	1

Number of arrival to different stations during the rush hours of the morning. Clearly can be appreciated that this user works near the “Howard St & Centre St” station.



Number of departures from different stations.



## 9.7 Appendix 7 Complete results of the Confusion Matrix (R console)

Test of the Over-fitting problem: Turn the sheets to Landscape orientation.

```
> prediction <- predict(gbmFit1, newdata =testing )
> confusionMatrix(prediction, testing$endstationname)
Confusion Matrix and Statistics
```

Prediction	Reference	
8 Ave & W 31 St	8 Ave & W 31 St	103
Christopher St & Greenwich St		5
Clinton St & Joralemon St	0	
E 33 St & 5 Ave		0
E 43 St & Vanderbilt Ave		2
E 7 St & Avenue A		1
Lafayette St & E 8 St		4
Warren St & Church St		4
Prediction	Reference	
8 Ave & W 31 St	Christopher St & Greenwich St	5
Christopher St & Greenwich St		88
Clinton St & Joralemon St		0
E 33 St & 5 Ave		1
E 43 St & Vanderbilt Ave		7
E 7 St & Avenue A		1
Lafayette St & E 8 St		2
Warren St & Church St		4
Prediction	Reference	
8 Ave & W 31 St	Clinton St & Joralemon St	0
Christopher St & Greenwich St		0
Clinton St & Joralemon St		2
E 33 St & 5 Ave		0
E 43 St & Vanderbilt Ave		1
E 7 St & Avenue A		0
Lafayette St & E 8 St		1
Warren St & Church St		0



Prediction	Reference	
8 Ave & W 31 St	E 33 St & 5 Ave	6
Christopher St & Greenwich St		0
Clinton St & Joralemon St		0
E 33 St & 5 Ave		22
E 43 St & Vanderbilt Ave		6
E 7 St & Avenue A		0
Lafayette St & E 8 St		0
Warren St & Church St		1
Prediction	Reference	
8 Ave & W 31 St	E 43 St & Vanderbilt Ave	7
Christopher St & Greenwich St		5
Clinton St & Joralemon St		0
E 33 St & 5 Ave		6
E 43 St & Vanderbilt Ave		181
E 7 St & Avenue A		1
Lafayette St & E 8 St		2
Warren St & Church St		1
Prediction	Reference	
8 Ave & W 31 St	E 7 St & Avenue A	0
Christopher St & Greenwich St		4
Clinton St & Joralemon St		0
E 33 St & 5 Ave		0
E 43 St & Vanderbilt Ave		0
E 7 St & Avenue A		16
Lafayette St & E 8 St		4
Warren St & Church St		1
Prediction	Reference	
8 Ave & W 31 St	Lafayette St & E 8 St	4
Christopher St & Greenwich St		3
Clinton St & Joralemon St		0
E 33 St & 5 Ave		1
E 43 St & Vanderbilt Ave		5
E 7 St & Avenue A		0
Lafayette St & E 8 St		339
Warren St & Church St		2
Prediction	Reference	
	Warren St & Church St	

8 Ave & W 31 St	5
Christopher St & Greenwich St	2
Clinton St & Joralemon St	0
E 33 St & 5 Ave	1
E 43 St & Vanderbilt Ave	8
E 7 St & Avenue A	0
Lafayette St & E 8 St	5
Warren St & Church St	14

### Overall Statistics

Accuracy : 0.8664  
 95% CI : (0.8421, 0.8881)  
 No Information Rate : 0.4009  
 P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8207  
 McNemar's Test P-Value : NA

### Statistics by Class:

Class: 8 Ave & W 31 St	
Sensitivity	0.8655
Specificity	0.9647
Pos Pred Value	0.7923
Neg Pred Value	0.9788
Prevalence	0.1348
Detection Rate	0.1166
Detection Prevalence	0.1472
Balanced Accuracy	0.9151
Class: Christopher St & Greenwich St	
Sensitivity	0.81481
Specificity	0.97548
Pos Pred Value	0.82243
Neg Pred Value	0.97423
Prevalence	0.12231
Detection Rate	0.09966
Detection Prevalence	0.12118
Balanced Accuracy	0.89515
Class: Clinton St & Joralemon St	
Sensitivity	0.500000
Specificity	1.000000



Pos Pred Value	1.000000
Neg Pred Value	0.997730
Prevalence	0.004530
Detection Rate	0.002265
Detection Prevalence	0.002265
Balanced Accuracy	0.750000
Class: E 33 St & 5 Ave	
Sensitivity	0.62857
Specificity	0.98939
Pos Pred Value	0.70968
Neg Pred Value	0.98474
Prevalence	0.03964
Detection Rate	0.02492
Detection Prevalence	0.03511
Balanced Accuracy	0.80898
Class: E 43 St & Vanderbilt Ave	
Sensitivity	0.8916
Specificity	0.9574
Pos Pred Value	0.8619
Neg Pred Value	0.9673
Prevalence	0.2299
Detection Rate	0.2050
Detection Prevalence	0.2378
Balanced Accuracy	0.9245
Class: E 7 St & Avenue A	
Sensitivity	0.64000
Specificity	0.99650
Pos Pred Value	0.84211
Neg Pred Value	0.98958
Prevalence	0.02831
Detection Rate	0.01812
Detection Prevalence	0.02152
Balanced Accuracy	0.81825
Class: Lafayette St & E 8 St	
Sensitivity	0.9576
Specificity	0.9660
Pos Pred Value	0.9496
Neg Pred Value	0.9715
Prevalence	0.4009
Detection Rate	0.3839
Detection Prevalence	0.4043
Balanced Accuracy	0.9618

Class: Warren St & Church St		
Sensitivity		0.40000
Specificity		0.98467
Pos Pred Value		0.51852
Neg Pred Value		0.97547
Prevalence		0.03964
Detection Rate		0.01586
Detection Prevalence		0.03058
Balanced Accuracy		0.69233