POLITECNICO DI TORINO

Department of Control and Computer Engineering

Master Degree Course in Computer Engineering

Master Degree Thesis

Predicting Bicycle Availablility By Means Of Data Mining Algorithms

Supervisors:
Prof. Paolo Garza
Prof. Luca Cagliero
Prof. Silvia Chiusano

Sorath Asnani

Academic Year 2017-2018

This work is dedicated to my beloved Parents
for their unconditional love and support
throughout my graduate studies

Summary

The concerns about global warming, air and noise pollutions, unstable fuel prices and road safety have caused policy makers to examine the need for sustainable means of transport. In the context of better urban mobility systems, the public Bicycle Sharing Systems (BSSs) have seen a great development in recent years. Community shared bicycle programs are being promoted all over the world as a "green" transportation system.

A Bike Sharing System (BSS) is an innovative transportation service available to the public, usually aimed for short-distance trips. The core idea of a BSS is that a user takes a bicycle from station A, uses that bike to travel to another location and returns the bicycle at station B. Community shared bicycling programs offer an environmentally friendly and inexpensive means of inner-city transportation.

The bike sharing systems have some problems related to limited number of bicycles and limited number of free slots in stations. In some occasions, it is not possible to pick-up bicycle at a certain station because the station might be empty or it might contain broken bikes. It may also happen that the user do not find a free parking slot to drop the bicycle at a certain station close to the destination.

Although in some bike sharing systems, there are trucks which are used to balance the bicycles by taking them from stations which are full or have more bicycles and leaving those bicycles in stations which are empty or have lesser bicycles, but a user who has already arrived at a station to pick a bike cannot wait for the trucks to get a bike. Similarly when a user arrives at a station which is completely full of bikes, he cannot wait for trucks to take some bikes so that he can park the bike. These situations are the causes of problems for the customers.

One possible solution to these problems is to enable the users to know about the availability of bicycles before the departure or arrival time so that they can go directly to those stations where bikes/parking slots are available. This can be done by predicting the number of bikes at each station at some future point of time. The

future prediction of the bicycle availability will greatly enhance the performance and reliability of the Bike Sharing Systems.

The major research goal of the thesis is to study and compare some automatic models to predict the availability of bikes some minutes ahead. The study has been performed on the data set of Bicing, a Bike Sharing System of Barcelona.

Data pre-processing is the critical part of this research. Extensive efforts have been put in identifying the dirty data in the available data set and to perform data cleaning in order to ensure the correctness of the data to be trained and tested.

After reviewing the literature, we decided to use three models for predicting the number of bikes, the ARMA, the Decision Tree and the Random Forest models. After the implementation of these models, a Baseline model based on the Historic Mean was considered to measure the performance of the three models.

After data cleaning, the total number of stations used in this study is 268. The training data set comprises of 800 continuous hours (from 2008-05-16 05:00 to 2008-06-27 06:58) with the timestamps separated by 2 minutes. The testing data set is composed of non-overlapping data of 30 hours (from 2008-06-28 05:00 to 2008-06-29 15:58). The timestamps between 24:00 and 05:00 are not considered in this study. All the models were trained using the same data sets. The predictions are made for 10 minutes ahead to 60 minutes ahead, separated by 10-minute intervals.

The performance of these models is calculated in terms of the absolute errors. The minimum, the maximum and the mean absolute errors have been computed for all the models, including the Historic Mean Baseline model. Six prediction models are generated from each of the four algorithms, one prediction model for each prediction time. In total 24 prediction models are generated.

The mean absolute errors are calculated for all the prediction models in order to compare and to rank the performance of the models. The Random Forest algorithm showed the highest mean absolute errors for all the prediction time periods, hence its performance to predict the number of bikes is worst in our case.

The performance of Decision Tree model remained in between the ARMA and the Random Forest, both in terms of training time and also in the mean absolute errors. But its performance was less than the Historic Mean Baseline model.

On the other hand, an interesting fact was identified between the ARMA

and the Historic Mean Baseline model. For the predictions of 10, 20 and 30 minutes ahead, the Baseline model showed the least mean absolute errors, while for higher time instances, such as 40, 50 and 60 minutes ahead, the performance of ARMA model was the best among all. Such results can lead us to the conclusions that the ARMA Model can be the best choice for predictions beyond 30 minutes, while for shorter terms, the Historic Mean can be considered as the best.

Behind the success of ARMA model is its long computational time. Prediction models with ARMA were trained in about 42 hours which is about 7 times greater than the Random Forest models.

The results in this study have shown that the exact number of bikes are quite predictable for near future. The predictions can be useful for both the system administration and the users. From the point of view of the system administration, the predictions can help in re-balancing operation of the bicycles. The truck operators can exploit the information about the availability of bicycles in future at each station and can schedule the re-balancing operations beforehand. From the users' perspectives, the predictions can enhance user satisfaction and reliability and eventually will result in enhancing overall system performance.

Hence the research was concluded by considering the ARMA model to be the best among other three models i.e. the Decision Tree, the Random Forest and the Historic Mean Baseline model for predictions from 40 to 60 minutes ahead.

Acknowledgements

This is an opportunity for me to express my gratitude to those who have supported me during my graduate studies. My whole-hearted thanks goes to my thesis supervisor, Dr. Paolo Garza, for his guidance and support throughout the research period. I have learnt a lot from him. In particular, I was greatly inspired by his problem solving skills. Apart from that, his active availability either in lab or on emails, contributed a lot in the success of this work.

I gratefully acknowledge Dr. Luca Cagliero, my thesis co-supervisor, for his valuable recommendations and ideas in this work. His guidance about new experimental dimensions contributed a lot in this work. Besides that, I am admired by his friendly nature.

My special thanks goes to Dr. Muhammad Mujtaba Shaikh, my mentor, who have taught me during undergraduate studies. Throughout the conduct of this research study, there were times when I was emotionally low and de-motivated. He was the person to inculcate positivity in me and raise my confidence. Without his moral support, it was not possible for me to face the difficult situations easily.

My sincere gratitude goes to my family. It was the unconditional love and the support of my parents at every step of my life that enabled me to achieve my goals. Their unshakable believe and trust in me is the major cause behind my success. Besides them, the love and appreciation of my siblings mean a lot to me.

I am equally thankful to my life partner, Rohit Hansrajani, for his love and encouragement. It was his continuous belief in me that kept me going forward and ultimately reaching at my destination. His presence in every moment of my life means a lot to me. The moments we shared together during this journey are unforgettable.

I am truly blessed for being surrounded by true friends in my life. It was their encouragement and support that motivated me a lot.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Growing concerns about global warming, air and noise pollutions, unstable fuel prices and road safety have caused policy makers to examine the need for sustainable means of transport. In the context of better urban mobility systems, the public Bicycle Sharing Systems (BSSs) have seen a great development in recent years. The idea of shared use of bicycles was initiated by Europe and then was expanded throughout the world [1]. Community shared bicycle programs are being promoted all over the world as a "green" transportation system.

A BSS comprises of many self-service stations located throughout cities which gives flexibility to users to pick-up bikes from one station, use that bike to travel to their destinations, and finally drop-off the bike at another station. Community shared bicycle sharing programs are targeted to daily mobility in urban areas. They are typically used by commuters as a preferred means of transport to travel to and from home and work-place and vise versa, provided that the distance between home and work place is not too long. Furthermore, such public bicycle sharing programs are inexpensive and convenient transport modes for students to travel to their universities on regular basis.

Users are required to subscribe to a bicycle sharing program in order to access bicycles. The subscription cost typically covers bicycle purchase and maintenance costs along with storage and parking responsibilities. In this way, users do not need to worry about the maintenance as they would do in case of private vehicles.

Public bicycle sharing programs provide a number of environmental, social and transportation related benefits. A BSS is the most feasible solution for the

"last mile" and "first mile" problems. The last mile refers to the distance between existing public transport stop and the final destination that might be one's home or workplace. Similarly, the first mile is the distance from a place to the public transport point. That distance might be too far to cover on foot. In such cases the presence of bicycles seems to be the most convenient option for the users. Hence, a BSS bridges the gap created by the existing public transport systems. Furthermore, the availability of BSS gives a variety of mobility options. It has lower implementation and operational costs as compared to the tram and bus networks within a city. A BSS is an environmentally-friendly transport mode in a sense that it offers reduced traffic congestion and reduced fuel usage which eventually decreases air pollution and there-by increasing health benefits.

1.2 Problem Statement

Besides the above mentioned advantages, bike sharing systems also have some problems related to limited number of bicycles and limited number of free slots in stations. In some occasions, it is not possible to pick-up bicycle at a certain station because the station might be empty or it might contain broken bikes. It may also happen that the user do not find a free parking slot to drop the bicycle at a certain station close to the destination.

Problem of not finding a bike can be solved by taking another public transport such as a tram, a bus or a train. However, if a person is currently holding a bike, and cannot leave it at the destination station, it may create an inconvenience and a reason for avoiding the use of the system in future.

Under such conditions, a prediction about the number of bikes a user will find in the next hour or so at a certain station may improve the system reliability and increase its usage. For example, if a person knows that he need to pick-up a bicycle in next 30 minutes from a station, and the prediction shows the unavailability of bike at that station, he may change either the departure time or go to another station. Similarly, if a person knows in advance that in next hour there would not be any free parking slot at a certain station, he might plan his journey according to the arrival time or may goes to another station.

Bicycle availability prediction is useful to both the system administrators and the users. From the perspective of service operators, the prediction of bikes will enable them to build a schedule for their workers to redistribute the bikes across several stations in order to re-balance the stations. Typically they use trucks to collect bikes from stations which contain many bikes or are full and leave them to

the stations which are empty or have less number of bikes at a certain time.

This work particularly focuses on predicting the number of bikes in stations in some future point of time by using different prediction models.

1.3 Motivation

This study is based on the bike sharing system of Barcelona, known as Bicing. Figure 1.1 taken from the Bicing website [2], shows the number of available bikes and the parking spaces at the station number 458. It does not tells the availability of bikes in future time. This is the major motivation for this research i.e. to predict the availability of the bikes.

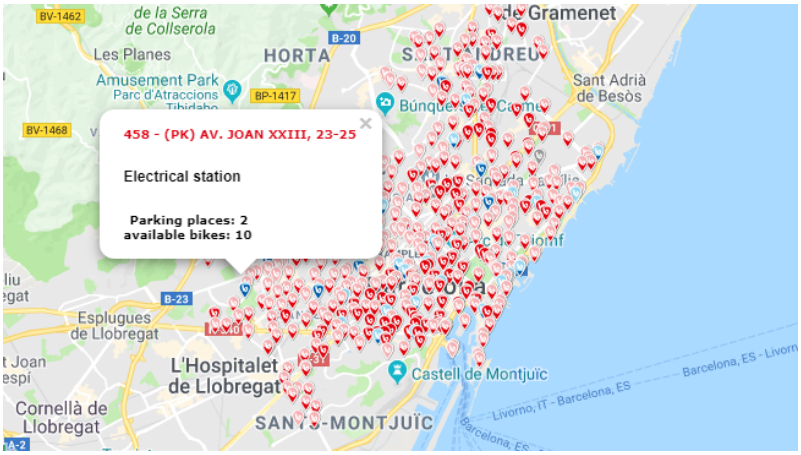

Figure 1.1: Availability of Bikes at a Bicing Station

1.4 Research Goals

Following are the major research goals:

- To predict the number of bikes in Bicing stations from 10 minutes ahead to 60 minutes ahead.
- To implement the Auto-Regressive Moving Average (ARMA) Model for the prediction of Bikes.
- To train the predictive models using also the Decision Tree and the Random Forest Algorithms.

- To create the Historic Mean Baseline Model to compare the performance of the above 3 models.
- To analyze the time taken by each of these models to generate the trained models.
- To identify the best and the worst model to be used for bicycle predictions based on the mean absolute error.

1.5 Document Organization

The complete thesis document is organized into five chapters.

- Chapter 1 provided the introduction to the research and its background study. It also described the problem statement and the motivation for the research. Further, the research goals were also discussed.
- Chapter 2 presents the literature review of the research. All the experimental study conducted, related to the research is briefly described. The weaknesses of the previous methods are also summarized.
- Chapter 3 discusses all the methodologies, the data set, experimental setup, and the experiments carried out in this study.
- Chapter 4 presents and discusses all the results of the experiments presented in the previous chapter. The training time duration of the models is presented and compared. The performance is calculated in terms of the absolute errors. Finally, the performance of the predictors is compared on the basis of the experimental results.
- Chapter 5 discusses the outcomes and results of the study. It presents the concluding statements on the basis of the experimental results. It describes the limitations of the study and future recommendations to the researchers and finally concludes the whole study.

Chapter 2

Literature Review

2.1 Introduction

A Bike Sharing System (BSS) is an innovative transportation service available to the public, usually aimed for short-distance trips. The core idea of a BSS is that a user takes a bicycle from station A, uses that bike to travel to another location and returns the bicycle at station B. Community shared bicycling programs offer an environmentally friendly and inexpensive means of inner-city transportation.

Shaheen et al. [1] provides the evolution and generations of community shared bicycle programs all over the world, including Europe, North and South America, Asia and Australia. According to the study, Europe has achieved much higher success in planning and implementing BSSs as compared to other continents of the world. Among other bicycle sharing programs, "Velo'v" in Lyon, France, "Velib" in Paris, France and "Bicing" in Barcelona, Spain have been studied widely.

In this thesis, the research has been carried out on the dataset of Barcelona's shared bicycle program, Bicing. Bicing was launched in March 2007. Currently, the network consists of over 420 stations with 6000 bicycles and over 106,635 yearly subscribers [2]. Stations are situated throughout the city with a distance of around 300 to 400 meters between each one. Many stations are situated next to public transport stops, which makes the BSS suitable for one-way travel [3]. Each station has between 15 and 39 parking slots [4].

To rent a bicycle, one swipes the contactless RFID card at a service station in order to be individually recognized by the system. The bike is then unlocked from its slot. Bicycles can be used for the first 30 minutes free of charge, every 30 minutes beyond that costs 0.70€ for 2 hours. The use of bicycles for more than

2 hours is discouraged with a penalty rate of 4.20€ per hour. When a certain number of warnings are exceeded, the membership might be cancelled. Bicycle can be returned by simply placing it in a spare slot at a Bicing station. The bike is recognized automatically and is then locked into place [3].

Besides the widespread use of community shared bicycles, the unavailability of bikes or free parking slots in the stations is still the major problem faced by the customers. There are 2 possible cases of the problem. The first is, when a user arrive at his nearest station to rent a bicycle, the station is either empty i.e. it does not contain any bike or it contains broken bikes which cannot be used. In that case, the user will have to look for other nearby stations for the bike availability. In the other case, when a user arrive at his destination station, that station might not have empty slots to leave the bicycle. In that case, the user will have to look for other nearby stations to return the bicycle. Both of these problems may contribute greatly to user frustration and dissatisfaction and eventually in decreasing the use and performance of the BSS.

One possible way to prevent the above mentioned problems is to provide the predictions for the number of available bikes and/or free slots to the customers. This will enable the users to know whether there would be any bikes or free slots in the desired station before actually going to that station. It can greatly contribute in enhancing customer satisfaction. The goal of this thesis is to implement prediction algorithms to forecast the number of available bikes upto 60 minutes ahead of time.

2.2 Background and Related Work

This section gives an overview of the prediction algorithms and the related work in literature in order to familiarize the readers with the core concepts of the algorithms. With the emergence of more and more BSSs all over the world, the research-areas are also extending in different dimensions related to the bike sharing systems. The first subsection gives a broad overview of BSS in general, the second subsection focuses on bike usage predictions for BSSs and the third subsection presents the major application areas of the prediction algorithms used in this study.

2.2.1 Bike Sharing Systems

The main goal of extensive research on BSSs is to enhance customer satisfaction by improving the performance of community shared bicycle services.

A number of studies have focused on finding the optimal locations for Bike Stations.

Chen et al. [5] proposed a system to find optimal locations for bike stations by predicting the user trip demands as opposed to the traditional urban planners. They devised a semi-supervised feature selection method to extract customized features from heterogeneous urban open data to predict bike trip demand and hence inferring optimal placement of stations. Their method outperformed the state-of-the-art approaches on recommending locations for optimal bike station placement and achieved particularly good results in feature selection based on city specific characteristics. Jimenez et al. [6] proposed a new characteristic known as a "Turnover Station Ratio" in order to measure the effectiveness degree of each station. This ratio indicates the number of times a station's capacity is used in a complete day. This new ratio together with other ratios such as "Number of Available Bikes" and "Cumulative Trips", allow to identify balanced, attractive and effective stations.

Another characteristics which has a great impact on customer satisfaction is the presence of faulty bikes in stations as investigated by Kaspi et al. [7]. They found out that faulty and unusable bikes seem to have significant impact on user satisfaction, even if the number of such bikes is relatively smaller. They concluded that operators should invest more resources in order to detect and recollect such damaged bikes. Vassimon [8] studied different factors impacting BSSs in more than 50 cities. He performed an evaluation of the performance and service quality of BSSs through a benchmarking analysis that relied on Key Performance Indicators (KPIs) and customer satisfaction. His study provides an interesting statistical analysis of BSS data and some insights on the related business models.

Optimal bike route planning has also been investigated by various researchers. Wakamiya et al. [9] proposed a system to enhance GPS-based navigation for bikes. In order to measure pedestrian congestion, they utilized location based social network services where they analyzed geo-tagged microblogs to provide the optimal route planning. Wu and Frias-Martinez [10] investigated the precision of biking times predicted by Google compared to real biking data from BSS. They concluded that Google's biking directions are generally good but longer trips and steep slopes pose a problem for Google's formulae and heuristic rules. Therefore they propsed a new predictive model for computing biking times that improved the accuracy of Google's biking time computations by 5%.

New possible improvements for BSSs using Internet of Things (IoT) were proposed by Razzaque and Clarke [11]. They proposed a potential communication infrastructure for next-generation BSSs that would offer new services by collecting, processing and using real-time and non-real-time data about the customers, the environments and the bikes.

2.2.2 Forecasting

Real-time monitoring of BSSs is not enough to ensure quality of service. In order to prevent problems and to improve effectiveness and customer satisfaction, forecasting of bikes usage is crucial. Various prediction models have been studied by researchers as discussed in the following paragraphs.

2.2.2.1 Univariate Methods

Prediction methods that use data which contains only one variable, which is to be predicted, are known as univariate methods. Time is used to index the observation in such methods. In other words, the univariate methods do not provide other variables as input for doing predictions. Time series forecasting comes in the category of univariate methods.

Time Series Forecasting

A time series is a sequence of observations taken sequentially in time [12]. Time Series Forecasting is the prediction of future values in a time series by taking into consideration the past and current values of that time series.

Many methods have been developed for the prediction of future values from sampled data. Naive prediction methods are based on statistical aggregate functions such as the average, the maximum, the minimum and the median values of the sampled data. Although they are the simplest methods with low computational costs, their results are quite constrained.

Time series methods are more powerful as compared to the Naive methods. They take into account the evolution of system based on historical values indexed by time. Time series methods vary from the simplest methods such as Auto-Regressive (AR) and Moving Average (MA) to the most complex variations of ARMA (VARMA, ARMAX, GARCH, etc) [12]. Such methods are usually applied in industry and economics to predict stock prices, but the applications can be extended to almost any area. Time series methods are widely used for one-dimensional data.

There are several approaches in the literature to predict the availability of bikes in bike sharing stations. Kaltenbrunner et al. [4] showed an Auto-Regressive Moving Average (ARMA) Model and Naive approach to forecast the available number of bikes and Froehlich et al. [13] showed a simple Bayesian Network based prediction algorithm. In [4], the authors have concluded that predicting the number of bikes in the next hour using the ARMA Model was more accurate than the

Naive results. Although the Naive Bayesian prediction based on Bayesian network in [13] seems to be very simple but useful in bike prediction, the authors used small number of classes according to percentages (for example, 5 classes: 0% to 20%, ..., 80% to 100%), rather than predicting the actual available number of bikes, which would be useful for accurate journey planning if a group of people is planning for a trip. The Bicing system is analyzed in both studies [4], [13].

Yoon et al. [14] devised a personal journey advisor for navigating in Dublin using BSS. They predicted the availability of bikes in a station in near future (5 and 60 minutes ahead) with a spatio-temporal prediction system based on Auto-Regressive Integrated Moving Average (ARIMA), which also takes into account seasonal trends and spatial correlations. Given the origin and destination, their application suggests the best pair of stations to take and return a bicycle.

Giot and Cherrier [15] predicted the number of bikes up to 24 hours ahead at a frequency of one hour using several different regression algorithms. It is interesting to note that their dataset comprised of 2 years as opposed to other studies which used several weeks' data, however, their dataset contained only network level data rather than station level data.

2.2.2.2 Multivariate Methods

A multivariate system has multiple variables, known as predictors, contributing in making predictions. Multivariate data will contain environmental characteristics along with the time value. Some recent studies have shown that considering the multivariate data is more efficient in terms of predictions as compared to the univariate data. In case of bike sharing systems, the data can be either univariate or multivariate. The multivariate methods also consider external factors such as weather conditions, holidays, festivals and events because these characteristics influence the bike usage. For such systems, Random Forest learning approach has been used in the literature.

Decision Tree

Decision Tree [16] is a popular algorithm applied to machine learning problems to make predictions. A decision tree is an n-ary tree. Its height is equal to one more than the number of predictors, where the last level contains the leaves and represents the output values. If the output value is a label instead of a number, the decision tree may be called as Classification tree.

Random Forest

Random Forest [17] is an algorithm that uses decision trees to create classification trees. In order to create classification trees, the procedure randomly selects a subset of predictors from the original data and build a classification tree based on it. After creating several trees, a final decision tree is built based on the average relevance of the predictors, and can be used to make predictions over other sets of observations. Besides building the decision tree, the Random Forest algorithm calculates the relevance of each predictor in the real environment.

Dias et al. [18] compared the results of Random Forest algorithm with ARIMA Model. They investigated the importance of various features and showed that the most important feature is "minute of the day", followed by weather parameters, specifically "average humidity". They demonstrated that the Random Forest algorithm greatly outperformed ARIMA and that predicting the status of stations two days ahead was feasible almost half of the time. They predicted upto 2 days ahead if the stations would either be completely full or empty. Although they have considered the external factors, but like [13], they also did not predicted the actual number of available bikes, instead they classified the status of stations in 5 classes such as, full, almost full, slots and bikes available, almost empty or empty.

Several studies focused on demand predictions and availability of bikes to improve customer experience. Chen at el. [19] proposed a dynamic cluster based framework for over-demand predictions in BSSs. Based on the contextual factors such as, time, weather, social and traffic events, they constructed a weighted correlation network to group stations into similar usage patterns into clusters. They proposed a Monte Carlo simulation in order to predict the over-demand probability of each cluster. They applied the proposed model to the real world data of New York City and Washington D.C. and demonstrated that their framework could accurately predict over-demand clusters.

Borgnat et al. [20] build a model of the cyclic temporal patterns with a linear regression and used that for forecasting. They utilized variables such as, weather, number of users and holiday markers. Their study reveals the spatial and temporal patterns of activity in the city along with the predictions of available bikes on hourly or daily basis. They have analyzed the data of Velo'v, the shared bicycle program of Lyon, France.

Gast el al. [21] focused on Velib, a BSS of Paris. They illustrated the probability distributions of bike availability. They ranked the stations according to expected number of available bikes. Their scoring rule lies on two criteria, "no bikes" and

"one or more bikes", because a customer is mostly interested if there is atleast one bike available. However, the prediction of the actual number of bikes would likely to benefit to the system providers.

2.3 Conclusion

This study considers the idea of using neighboring stations as described in [4] and using the ARMA Model for predicting the number of bikes in near future (10 to 60 minutes ahead). To the best of our knowledge, Decision Trees and Random Forest have not been used on such data which contains time information and neighbor stations, without considering the external environmental factors. This is the first study which will compare the performance of ARMA Model, Decision Tree and Random Forest with the Baseline Model.

Chapter 3

Methodology

3.1 Introduction

This chapter provides the detailed study of the Bicing data set and the methodology for experimental setup. The chapter first introduces the data set and explains the pre-processing steps to clean the data. It then explains the ARMA, the Decision Tree, the Random Forest and the Historic Mean Baseline models in detail and describes the preparation of the data for the experiments.

3.2 Bicing

Bicing is an urban community bike sharing program, managed and maintained by the city council of Barcelona and the Clear Channel Communications Corporation [4]. Bicing service is mainly used by the people to commute within the city of Barcelona.

The Bicing website [2] shows that there are currently more than 420 stations. Each station has fixed number of slots. Each slot can be in one of the three conditions at a time as shown in Table 3.1.

Table 3.1: Status of bicycle slots in Bicing stations

Slot Status	**Meaning**
empty	without a bicycle
occupied	holding a bicycle
out of service	either the slot itself or the bicycle it contained is marked as damaged

Users first need to register to the system by paying a fixed amount for yearly

subscription, after that they receive an RFID card which grants them the access to use the bicycles. The user swipes the RFID card at a service station to rent a bicycle. The system recognizes the user and stores information about the bike, which is then unlocked from its slot. The first 30 minutes of bicycle usage are free of charge and the subsequent 30 minute intervals cost 0.70€ for a maximum of 2 hours. The use of bicycles beyond that time limit is discouraged with a penalty rate of 4.20€ per hour. The continuous use of bicycle for more than 2 hours might result in the cancellation of membership. Bicycle can be returned by simply placing it in an empty slot at any Bicing station. The bike is recognized automatically and is then locked into place.

According to the Bicing website [2], bicycles can be rented all day long except between 02:00 and 05:00 from Monday to Thursday, while on Friday, the bicycles can not be withdrawn between 03:00 AM and 05:00 AM. During these hours, the bicycles can be returned but not withdrawn. The service is open for 24 hours on Saturdays, Sundays and holidays. However, the service timings were different in past years. As mentioned in [4], bicycles could be withdrawn from stations between 05:00 and 24:00 from Monday to Friday and on Saturdays and Sundays, the service remained open for 24 hours.

3.3 Data Set

The dataset used in this study comprises of 284 stations in total. For each station, the following information is given:

1. Station Id
2. Timestamp
3. Number of bikes available
4. Number of free slots
5. Longitude
6. Latitude

The data is available every 2 minutes from 05:00 to 24:00 for each day. Due to the unavailability of bike rental service from 24:00 to 05:00, our study is restricted to the time frame between 005:00 and 24:00. We have used approximately 6 weeks' data i.e. from May 16, 2008 to June 29, 2008.

3.4 Architecture of the Prediction Framework

The steps involved in the architecture are shown in figure 3.1. The following sections describe the overall architecture of the prediction framework.

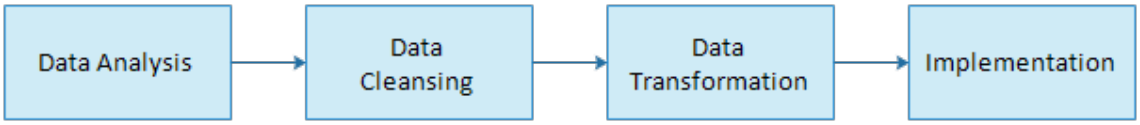

Figure 3.1: Architecture of the Prediction Framework

3.5 Data Analysis

In the first phase, the provided data set is analyzed and is divided in the training and the test set. The training data set consists of 800 hours, from 2008-05-16 05:00 to 2008-06-27 06:58. The testing data set comprises of 30 hours, from 2008-06-28 05:00 to 2008-06-29 15:58. As mentioned previously, the time series is in the increments of 2 minutes, so the training data should have 24,000 rows in total. Similarly, the testing data should have 900 rows. But the actual number of rows are less than the required rows because of the presence of some dirty data as described below.

3.5.1 Dirty Data

Time series data are often found with dirty or imprecise values. The Bicing data, which we have, is noisy due to temporary station closures, technical issues caused by the maintenance work, Internet connectivity failures, and broken bicycles and parking slots. Training algorithms can not be applied to the given data set because it contains dirty time series. In particular, the characteristics due to which we consider it as dirty data are discussed in the following subsections.

3.5.1.1 Incomplete Data

After analyzing the training and testing data sets, it was found that majority of the stations contain less than 24,000 and 900 rows for training and testing data sets respectively. It was due to the fact that there were some missing timestamps in the data set. No information about the number of bikes and slots was available in those missing timestamps. A sample of data set is shown in Figure 3.2 which exemplifies the missing data.

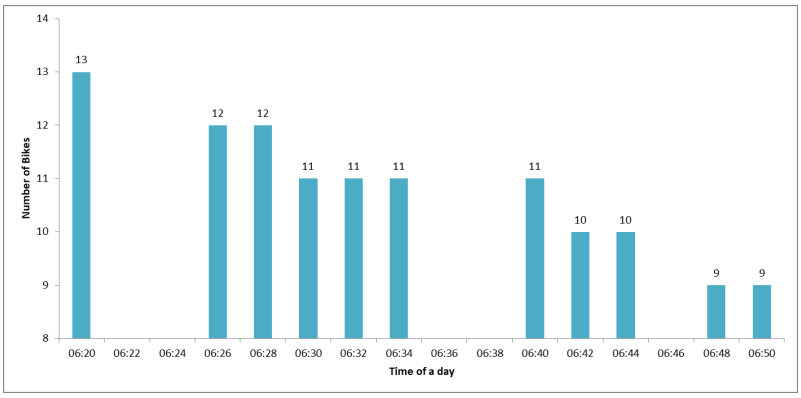

Figure 3.2: Example of Missing Values in Time Series

3.5.1.2 Incorrect Data

In some stations, incorrect values have been observed. For example, for some timestamps, the number of used slots (available bikes) and the number of free slots are both shown as 0, which is not correct. At any time, both of these values cannot be 0. An example of such data is shown in Figure 3.3.

station	date	used	free
2	2008-05-19 22:48	12	0
2	2008-05-19 22:50	12	0
2	2008-05-19 22:52	12	0
2	2008-05-19 22:58	12	0
2	2008-05-19 23:00	0	0
2	2008-05-19 23:02	11	1
2	2008-05-19 23:08	12	0
2	2008-05-19 23:10	0	0
2	2008-05-19 23:12	10	2
2	2008-05-19 23:14	12	0
2	2008-05-19 23:16	12	0
2	2008-05-19 23:18	12	0
2	2008-05-19 23:20	12	0

Figure 3.3: Example of Incorrect Data

3.5.1.3 Inaccurate Data

Inaccuracy in time series data has been also observed in most of the stations. Sometimes, the number of used and free slots drop significantly, usually near to

0 as shown in Figure 3.4a. On the other hand, when the time series is quite smooth, there is sudden deviation of values in just 2 minutes, and after that values become smooth again, as shown in Figure 3.4b. Such values seem to be practically impossible and hence are considered as inaccurate.

station	date	used	free
2	2008-05-16 05:34	8	13
2	2008-05-16 05:36	8	13
2	2008-05-16 05:38	0	1
2	2008-05-16 05:40	8	13
2	2008-05-16 05:46	8	13
2	2008-05-16 05:48	0	1
2	2008-05-16 05:50	8	13
2	2008-05-16 05:52	8	13
2	2008-05-16 05:54	9	12
2	2008-05-16 06:02	9	12
2	2008-05-16 06:04	0	2
2	2008-05-16 06:08	9	12
2	2008-05-16 06:12	9	12

(a) Sudden drop of values

station	date	used	free
2	2008-05-28 10:10	9	8
2	2008-05-28 10:14	9	8
2	2008-05-28 10:16	9	8
2	2008-05-28 10:18	9	8
2	2008-05-28 10:20	15	2
2	2008-05-28 10:22	9	8
2	2008-05-28 10:24	9	8
2	2008-05-28 10:26	9	8
2	2008-05-28 10:28	9	8
2	2008-05-28 10:30	8	9
2	2008-05-28 10:32	8	9
2	2008-05-28 10:34	8	9

(b) Sudden deviation in values

Figure 3.4: Examples of Inaccurate Data

3.6 Data Cleaning

Data cleaning is critical to ensure that the data used to train our prediction models is valid. Due to the presence of dirty data, a 3-step procedure, shown in Figure 3.5, has been performed to clean the data as described in the following subsections.

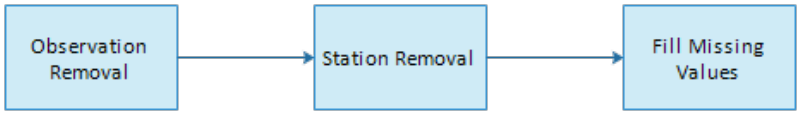

Figure 3.5: Data Cleaning Steps

3.6.1 Observation Removal

The observation removal phase is shown in Figure 3.6. As mentioned previously, the total number of slots in stations lies between 15 and 39, we considered this criteria while filtering the data. At first, the sum of used and free slots is computed. A threshold of value 10 is considered to compare the sum. A threshold value 10 is considered instead of 15 in order to provide flexibility because in some stations, some slots might be broken and should not be counted. If the sum of current observation of used and free slots is less than the threshold, that observation is discarded. This step also removes the incorrect data where both the used and the free slots are 0. If the sum is greater than 10, another criteria is established. In that case, the sum of current observations is compared with the sum of last observation and the next observation. If the current sum is neither equals to the sum of the last observation nor it is equal to the sum of next observation, that observation is also removed. This step gets rid of the inaccuracy which is shown in Figure 3.4b.

3.6.2 Station Removal

After removing the inconsistent observations, the number of rows are counted for each station. The number of rows in training data set should be 24,000 and that for testing data set should be 900, as mentioned previously. We discarded all those stations where the number of rows were less than 80% of the total data set in both the training and the test sets. After this step, 16 stations were discarded due to large number of missing values. The training models were created on the remaining 268 stations. This phase is shown in Figure 3.7.

3.6.3 Fill Missing Values

After observation and station removal phases, the next step is to fill the missing values in the training data set. To maintain the real-world conditions, this step is not performed on test data set. Before training the models, it is necessary to ensure that the training data should not contain any gaps. In order to do that, new rows are inserted in the time series data with the time field increment of 2 minutes. The values for the number of bikes and free slots were considered to be equal to the nearest available values in the data set.

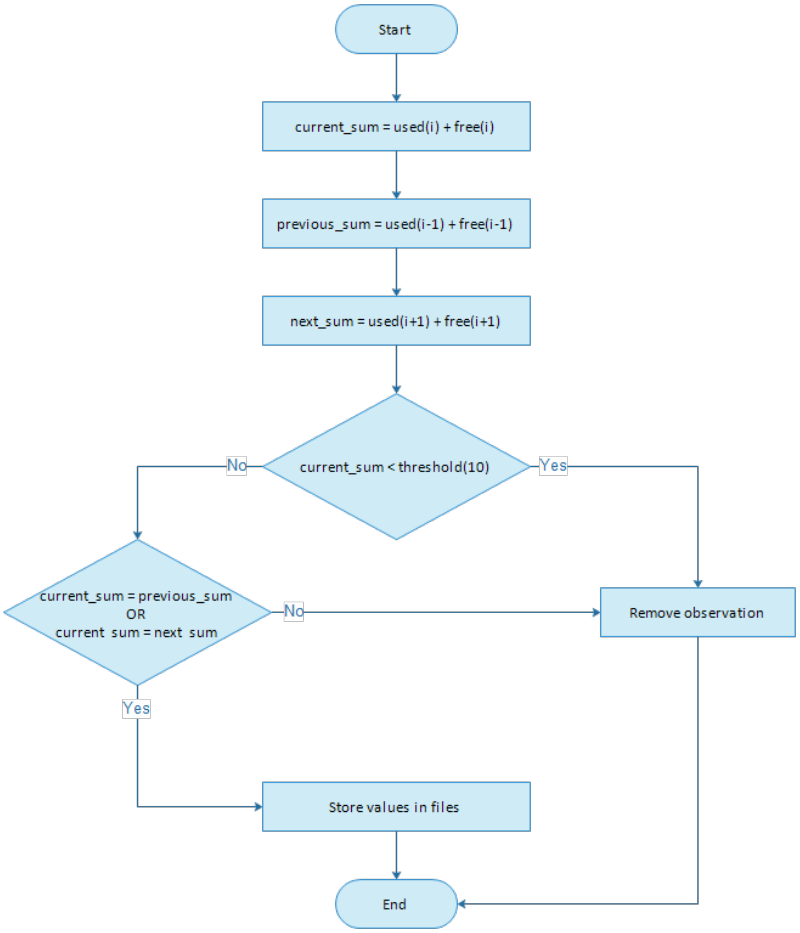

Figure 3.6: Observation Removal Steps

3.6.4 Outcome of Data Cleaning

The result of this 3-step procedure is that the training and testing data sets are free of the dirty values and are ready for further processing. Each station now contains 24,000 rows in the training data set, where as some rows are missing in the testing data set but they are free of inaccurate and incorrect values.

3.7 Data Transformation

After data cleaning, both the training and the testing data sets are transformed so that these datasets can be provided as inputs to the training models. The steps for transforming the training and the testing data are shown in Figure 3.8 and are discussed in the following subsections.

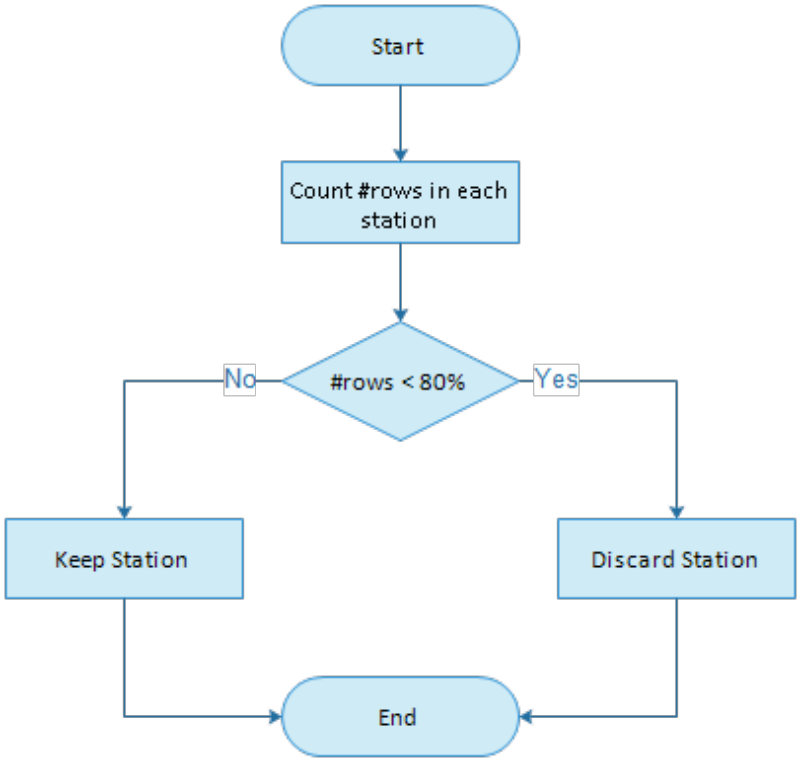

Figure 3.7: Station Removal Phase

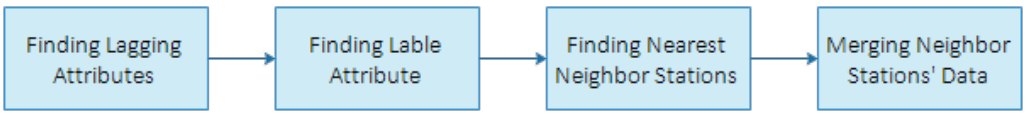

Figure 3.8: Data Transformation Steps

3.7.1 Finding Lagging Attributes

The first step to transform the data is to find the lagging attributes. The number of bikes at time "t" are given, the lagged time values are computed for each timestamp, such as the number of bikes at time "t-1", "t-2",...,"t-10". For all the experiments we are considering the history of 20 minutes, that is why we need to compute 10 lagged values, i.e. from "t-1" to "t-10", for the number of bikes for each station.

3.7.2 Finding Label Attribute

For training the models, it is necessary that the training and test set must contain an attribute known as "label attribute". The label attribute represents the target attribute for prediction. It specifies the number of bikes that will be available say

10 minutes ahead. The label attribute is different for each prediction time as shown in Table 3.2.

Table 3.2: Computation of Label Attribute for each Prediction Time

Prediction Time	**Label Attribute**
10 minutes ahead	$t + 4$
20 minutes ahead	$t + 9$
30 minutes ahead	$t + 14$
40 minutes ahead	$t + 19$
50 minutes ahead	$t + 24$
60 minutes ahead	$t + 29$

3.7.3 Finding Nearest Neighbor Stations

The models will be trained by exploiting the information of the current station as well as its 5 surrounding stations. In order to compute the nearest neighbors, information about the longitude and latitude, which is present in the Bicing dataset, has been used. The distance between 2 stations is computed by using the Haversine Formula described in [22]. As a result of this step, we now have 5 nearest neighbour stations for each station.

3.7.4 Merging Neighbor Stations' Data

In order to prepare the training and test sets, we now need to combine the information about the 5 nearest stations for each station. For example, 5 neighbor stations of Station # 1 are: Station # 26, 122, 3, 32 and 4. The training and testing files of Station # 1 contain attributes as shown in Table 3.3

Table 3.3: Attributes Involved in Transformed Training and Test Sets of Station#1

Attribute 1	Time
Attribute 2 to 11	s1 (t-1, ..., t-10)
Attribute 12 to 21	s26 (t-1, ..., t-10)
Attribute 22 to 31	s122 (t-1, ..., t-10)
Attribute 32 to 41	s3 (t-1, ..., t-10)
Attribute 42 to 51	s32 (t-1, ..., t-10)
Attribute 52 to 61	s4 (t-1, ..., t-10)
Attribute 62	s1(Lable Attribute)

At the end of this step, the training and testing files are ready to be served as input to the training models.

3.8 Training Model

As discussed in Chapter 1, three machine learning models have been used in this research work, namely, the ARMA model, the Decision Tree and the Random Forest Model. Along with them, the Historic Mean Baseline Model is also used. Each of these models are described as follows.

3.8.1 The Auto-Regressive Moving Average (ARMA) Model

An ARMA, also written as ARMA(p, q) model is a combination of AR(p) and MA(q) models. The ARMA model is suitable for univariate time series modeling. This model gives the feasibility to use the recent history of both, the current station and its surrounding stations, to predict the availability of bicycles. The AR component deals with the autocorrelated nature of current station's time series, while the MA component provide the information about other stations' time series generally known as denominated "inputs".

A general form of ARMA model, as given in [4], can be written as:

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \sum_{j=1}^{m} \sum_{i=1}^{q} b_{(i,j)} I_{j(t-1)}$$

The meaning of each of these symbols is described in Table 3.4.

Table 3.4: Description of variables used in ARMA Model

X	Number of Bikes to be predicted
p	order of AR model
q	order of MA model
I_j	input time series
m	total number of "input" time series
t	time index for each time series
a_i and $b_{(i,j)}$	model coefficient to be computed during training phase

3.8.1.1 Parameter Values

In all the experiments presented in this thesis, a history of 20 minutes (10 samples) has been used for the station for which the experiment is carried out (AR component) and also for its neighbor stations (MA component). So $p = q = 10$ for the experiments. The number of neighbor stations considered in these experiments is 5 i.e. $m = 5$.

3.8.1.2 AR Component

The AR part is related to the data of current station the prediction is done for. Since the order of AR is $p = 10$, the history of 10 samples (20 minutes) is considered and the lagging values such as $t-1$, $t-2$,..., $t-10$ are computed. For each station, the label attribute is also required. Predictions are done from 10 minutes ahead up to 60 minutes ahead and the label attribute is computed as shown in Table 3.2.

3.8.1.3 MA Component

After the AR component, we now need to prepare the MA component of the ARMA model. The MA component is related to the neighbor stations and the order of MA is considered to be $q = 10$ fo all the experiments, as mentioned previously. The number of neighbor stations used for experiments is $m = 5$. The lagging values such as $t-1$, $t-2$,..., $t-10$ are computed for 5 nearest stations of the current station.

3.8.1.4 Merging AR and MA Components

For the ARMA model we need to combine both the AR and MA parts, so the data computed above are merged in single file for each station. Each file contains the timestamp, lagging values of the current station, lagging values of 5 nearest stations and the label attribute of the current station as shown in Table 3.3. Implementation details are provided later in this chapter.

3.8.2 The Decision Tree Model

Decision tree is one of the most popular machine learning algorithms. The general motive of using a Decision Tree is to create a training model which can be used to predict a value of target variable, the number of bikes in our case, by learning decision rules inferred from training data. As the name implies, the decision tree algorithm solves a machine learning problem by using tree representation. Each internal node of the tree corresponds to an attribute and each leaf node corresponds to a label attribute.

3.8.2.1 Decision Tree Algorithm Pseudocode

1. The best attribute of the dataset is placed at the root of the tree.
2. The training set is splitted into subsets. Subsets are made in such a way that each subset contains data with the same value for an attribute.
3. Steps 1 and 2 are repeated on each subset until the leaf nodes are found on all branches.

The implementation details are provided later in this chapter.

3.8.3 The Random Forest Model

Random Forest is an algorithm that uses decision trees to create classification trees. In order to create classification trees, the procedure randomly selects a subset of predictors from the original data and build a classification tree based on it. After creating several trees, a final decision tree is built based on the average relevance of the predictors, and can be used to make predictions over other sets of observations. Besides building the decision tree, the Random Forest algorithm calculates the relevance of each predictor in the real environment. The implementation details are provided later in this chapter.

3.9 Implementation

This section discusses the implementation of the algorithms discussed in this thesis work. All the experiments are done in RapidMiner Studio.

The general block diagram of the implementation is shown in Figure 3.9. The training data is provided as input to the training model. The training model can be either Linear Regression for ARMA Model, Decision Tree or Random Forest. The training model is then applied on testing data. Then the performance is evaluated and is stored in a file for each station. The performance is measured in terms of absolute error. The experiments are executed separately for each prediction time from 10 minutes ahead to 60 minutes ahead.

3.9.1 Algorithm 1: The ARMA Model

The ARMA Model is trained by means of Linear Regression process of RapidMiner as shown in Figure 3.10. The entire process is repeated for each station. The operators used in the model are described below:

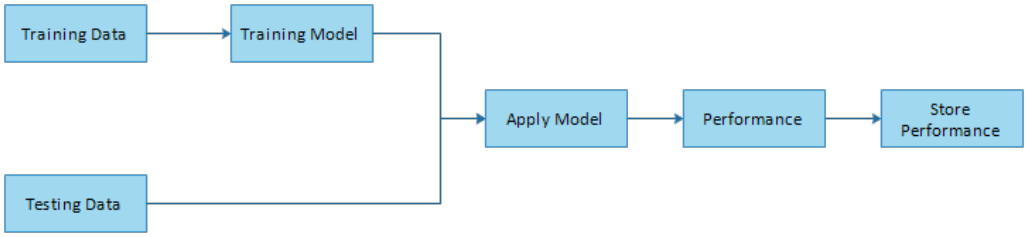

Figure 3.9: Block Diagram of Experimental Setup

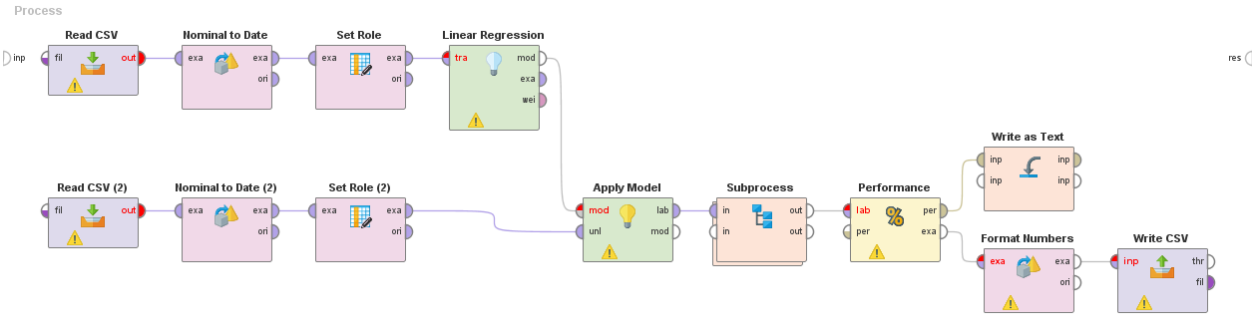

Figure 3.10: The ARMA Model

Read CSV: Reads the training data of a station.

Nominal to Date: The data type of all attributes is Nominal by default. This operator changes the data type of the timestamp attribute from Nominal to Date.

Set Role: Sets 'label' as target role of the last label attribute of the training data.

Linear Regression: Generates a trained model by using Linear Regression Algorithm.

Read CSV (2): Reads the testing data of a station.

Nominal to Date (2): This operator changes the data type of the timestamp attribute from Nominal to Date in test data.

Set Role (2): Sets 'label' as target role of the last label attribute of the testing data.

Apply Model: The trained model and the testing data is provided as input to this operator. It applies the generated model on the test data and outputs the labeled data by adding another column "Prdeiction(Label)" in the test data.

Performance: This operator is used for statistical performance evaluation. The performance for all the experiments is measured in terms of absolute error.

Write as Text: Stores the computed performance (absolute error) for each station in a file.

Format Numbers: It reformats the numeric attributes to integer type.

Write CSV: Stores the resulting test file with predictions in a file.

Subprocess: The subprocess includes some other operators as shown in Figure 3.11. When the model is applied to the test data, some of the predicted values for the number of bikes are negative, which needs to be processed further, because it is not possible for the predicted number of bikes to be negative. The purpose of the subprocess is to replace the negative values with '0'. The output of Apply Model is given as the input to the Subprocess and its output serves as the input to the Performance operator. The operators inside the subprocess are described below.

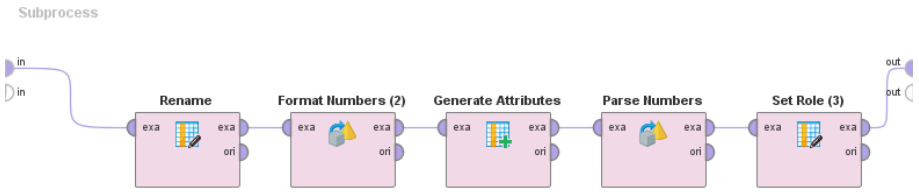

Figure 3.11: Subprocess

Rename: Renames the name of prediction attribute from prediction(label_attr) to prediction_label_attr.

Format Numbers (2): Reformats the label and prediction attributes to 'Number' type.

Generate Attributes: Generates a new attribute which is same as the predicted attribute but replacing negative values by '0'.

Parse Numbers: Changes the type of Nominal attributes to Numeric type.

Set Role (3): Sets the role to the label and predicted attributes to be used by the Performance operators.

3.9.2 Algorithm 2: The Decision Tree Model

Input to the Decision Tree Model is same as that of ARMA Model. The process is essentially the same as described above except the learning operator. Decision Tree operator is used instead of Linear Regression as shown in Figure 3.12.

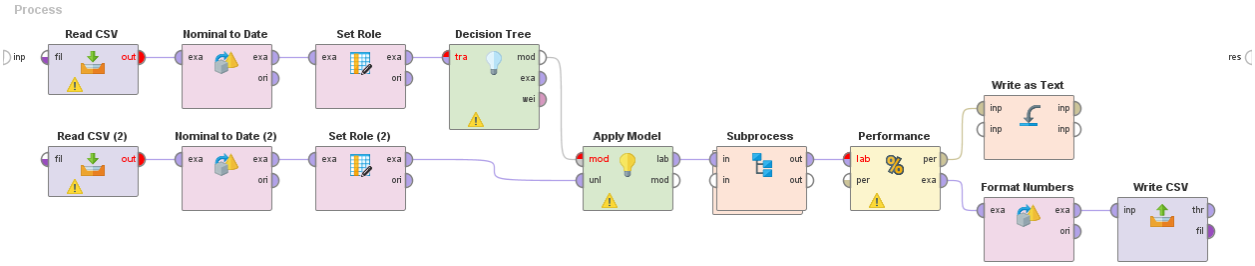

Figure 3.12: The Decision Tree Model

3.9.3 Algorithm 3: Random Forest Model

The third experiment is carried out by using Random Forest operator as shown in Figure 3.13.

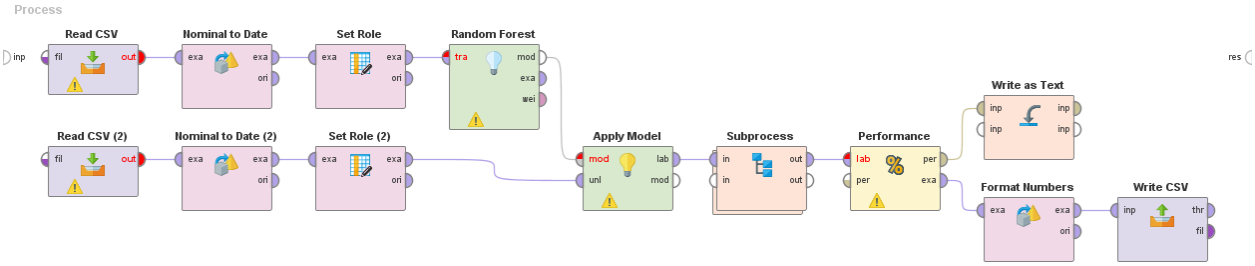

Figure 3.13: The Random Forest Model

3.9.4 Performance Evaluation

The performance of these machine learning algorithm is computed in terms of absolute errors. The average absolute errors are then compared with the Historic Mean Baseline Model as described below.

3.10 The Historic Mean Baseline Model

In order to compare the performance of the above three machine learning algorithms, the Historic Mean Baseline model is generated. Since in all the above experiments the history of 20 minutes (10 samples) is considered, the Historic Mean also considers the time window comprising of 10 samples. The steps for the computations of the baseline model are listed below.

3.10.1 Computations for 10-minute ahead prediction

1. The mean is computed from "t-5" to "t-15" for the number of bikes.
2. This mean is the predicted number of bikes at time instance "t".
3. Absolute difference of the actual number of bikes and the predicted number of bikes is computed for each timestamp.
4. The average of all the absolute difference is computed.

3.10.2 Computations for 20-minute ahead prediction

1. The mean is computed from "t-10" to "t-20" for the number of bikes.
2. This mean is the predicted number of bikes at time instance "t".
3. Absolute difference of the actual number of bikes and the predicted number of bikes is computed for each timestamp.
4. The average of all the absolute difference is computed.

3.10.3 Computations for 30-minute ahead prediction

1. The mean is computed from "t-15" to "t-25" for the number of bikes.
2. This mean is the predicted number of bikes at time instance "t".
3. Absolute difference of the actual number of bikes and the predicted number of bikes is computed for each timestamp.

4. The average of all the absolute difference is computed.

3.10.4 Computations for 40-minute ahead prediction

1. The mean is computed from “t-20” to “t-30” for the number of bikes.
2. This mean is the predicted number of bikes at time instance “t”.
3. Absolute difference of the actual number of bikes and the predicted number of bikes is computed for each timestamp.
4. The average of all the absolute difference is computed.

3.10.5 Computations for 50-minute ahead prediction

1. The mean is computed from “t-25” to “t-35” for the number of bikes.
2. This mean is the predicted number of bikes at time instance “t”.
3. Absolute difference of the actual number of bikes and the predicted number of bikes is computed for each timestamp.
4. The average of all the absolute difference is computed.

3.10.6 Computations for 60-minute ahead prediction

1. The mean is computed from “t-30” to “t-40” for the number of bikes.
2. This mean is the predicted number of bikes at time instance “t”.
3. Absolute difference of the actual number of bikes and the predicted number of bikes is computed for each timestamp.
4. The average of all the absolute difference is computed.

3.11 Conclusion

In this chapter details about the pre-processing of training and testing data sets are provided. Along with that, the architecture of the prediction framework is discussed in detail and finally the implementation details are also presented.

Chapter 4

Experimental Results

4.1 Introduction

This chapter illustrates the experimental results of the models trained using the ARMA, Decision Tree and Random Forest algorithms. In the end it provides the comparisons among all these results and also compares the results with the Historic Mean Baseline Model.

4.2 Results

The major goal of the thesis is the implementation of ARMA Model to predict the number of available bikes in Bicing stations from 10 minutes ahead to 60 minutes ahead. Apart from that, the performance of Decision Tree and Random Forest algorithms is also measured and compared with that of the ARMA Model. All the three models are then compared with the Baseline Model which is actually the Historic Mean for the prediction of the bicycle availability. The results are compared in terms of the training time duration and the prediction error as discussed below.

4.2.1 Training Time Duration

Training time duration is the time taken by an algorithm to generate trained model from the training data which can be applied to testing data.

4.2.2 Prediction Error

The prediction error is computed in terms of "Mean Absolute Error (MAE)". It is defined as:
"The Mean Absolute Error (MAE) measures the average magnitude of the errors

in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight" [23].

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

where:

- n = number of test samples
- y_j = actual number of bikes in time series
- $\hat{y}_j$ = predicted number of bikes in time series

4.3 Auto-Regressive Moving Average (ARMA)

This section discusses the results obtained by using the ARMA Model. First, the training time duration is shown followed by the prediction error.

4.3.1 Training Time Duration

The training time duration of ARMA model is shown in Table 4.1. It took approximately 7 hours to generate a model which could predict the number of bicycles up to 10 minutes ahead. In our case, we need to generate 6 models to predict from 10 minutes to 60 minutes ahead, hence the total training time was about 42 hours for generating all the required models.

Table 4.1: Training Time for ARMA Model

ARMA Model	
10 minutes ahead prediction	7 hours
10, 20, ..., 60 minutes ahead prediction	42 hours

4.3.2 Prediction Error

The trained model is tested on a test data set which comprises of 30 hours of data. The performance is calculated in terms of absolute errors. The minimum, the maximum and the mean absolute errors are computed for all the stations over all the prediction time frames i.e. from 10 to 60 minutes ahead as shown in Table 4.2.

Table 4.2: Prediction Errors by ARMA Model

Prediction Time	Maximum Error	Mean Error	Minimum Error
10 minutes ahead	3.13	1.296	0.1
20 minutes ahead	3.783	1.622	0.156
30 minutes ahead	4.364	1.892	0.211
40 minutes ahead	4.886	2.121	0.269
50 minutes ahead	5.372	2.320	0.328
60 minutes ahead	5.76	2.498	0.349

The graphical representation of the results in Figure 4.1 shows that the minimum, the maximum and the mean prediction errors increase as the prediction time increases. In other words, if we predict the number of bicycles 10 minutes ahead, the average error is relatively low as compared to the predictions made for 60 minutes ahead.

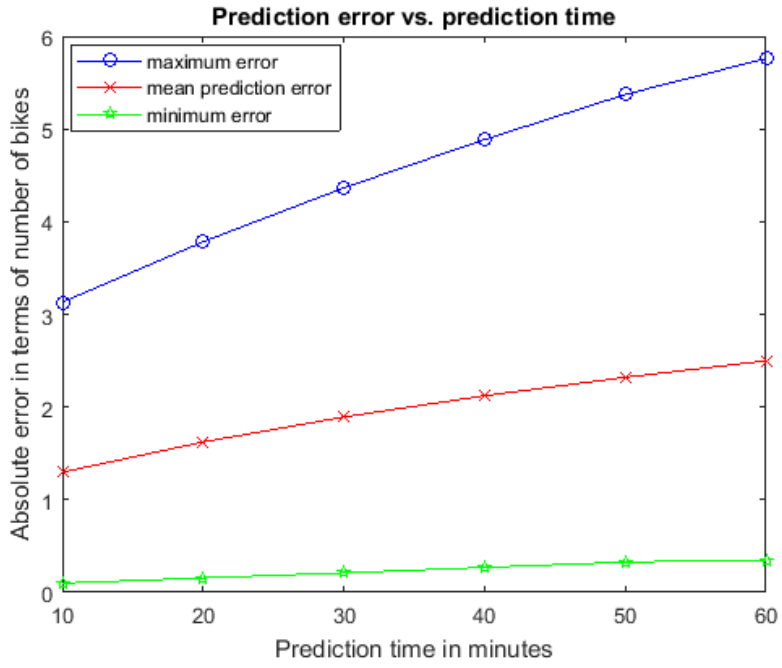

Figure 4.1: Average prediction error according to prediction interval, with minimum and maximum errors for ARMA Model

4.4 Decision Tree

This section discusses the results obtained by using the Decision Tree. First, the training time duration is shown followed by the prediction error.

4.4.1 Training Time Duration

The training time duration of Decision Tree model is shown in Table 4.3. It took approximately 1 hour and 30 minutes (1.5 hours) to generate a model which would be used to predict the number of bicycles up to 10 minutes ahead. In our case, we need to generate 6 models to predict from 10 minutes to 60 minutes ahead, hence the total training time was about 9 hours for generating all the required models.

Table 4.3: Training Time for Decision Tree

Decision Tree	
10 minutes ahead prediction	1.5 hours
10, 20, ..., 60 minutes ahead prediction	9 hours

4.4.2 Prediction Error

The trained model is tested on the same test data set which was used for ARMA Model. The performance is calculated in terms of absolute errors. The minimum, the maximum and the mean absolute errors are computed for all the stations over all the prediction time frames i.e. from 10 to 60 minutes ahead as shown in Table 4.4.

Table 4.4: Prediction Errors by Decision Tree Model

Prediction Time	**Maximum Error**	**Mean Error**	**Minimum Error**
10 minutes ahead	8.653	1.228	0.139
20 minutes ahead	8.669	1.741	0.26
30 minutes ahead	14.247	2.125	0.629
40 minutes ahead	11.227	2.419	0.64
50 minutes ahead	15.189	2.716	0.649
60 minutes ahead	11.218	2.873	0.512

The graphical representation of the results in Figure 4.2 shows that the minimum and the mean prediction errors increase as the prediction time increases, where as the maximum prediction error is approximately same for 10 and 20 minutes ahead

prediction. It then increases for 30 minutes ahead and again declines for 40 minutes ahead, and repeats the similar pattern for 50 and 60 minutes ahead prediction time.

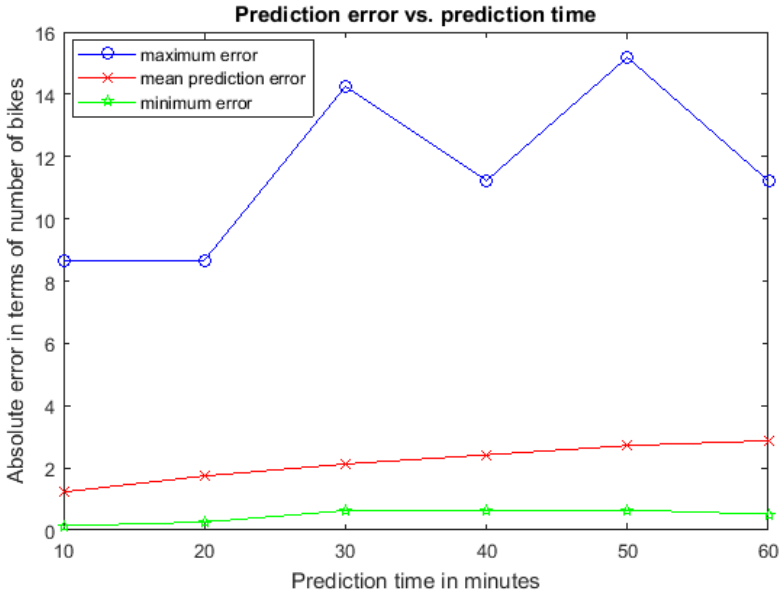

Figure 4.2: Average prediction error according to prediction interval, with minimum and maximum errors for Decision Tree

4.5 Random Forest

This section discusses the results obtained by using the Random Forest. First, the training time duration is shown followed by the experimental results.

4.5.1 Training Time Duration

The training time duration of Random Forest model is shown in Table 4.5. It took approximately 1 hour to generate a model which would be used to predict the number of bicycles up to 10 minutes ahead. In our case, we need to generate 6 models to predict from 10 minutes to 60 minutes ahead, hence the total training time was about 6 hours for generating all the required models.

4.5.2 Prediction Error

The trained model is tested on the same test data set which was used for ARMA Model. The performance is calculated in terms of absolute errors. The minimum,

Table 4.5: Training Time for Random Forest

Random Forest	
10 minutes ahead prediction	1 hour
10, 20, ..., 60 minutes ahead prediction	6 hours

the maximum and the mean absolute errors are computed for all the stations over all the prediction time frames i.e. from 10 to 60 minutes ahead as shown in Table 4.6.

Table 4.6: Prediction Errors by Random Forest Model

Prediction Time	Maximum Error	Mean Error	Minimum Error
10 minutes ahead	8.402	1.901	0.555
20 minutes ahead	8.381	2.229	0.691
30 minutes ahead	9.07	2.511	0.64
40 minutes ahead	9.996	2.757	0.533
50 minutes ahead	11.075	2.954	0.576
60 minutes ahead	9.389	3.133	0.543

The graphical representation of the results in Figure 4.3 shows that the mean prediction errors increase as the prediction time increases, where as the maximum prediction error is approximately same for 10 and 20 minutes ahead prediction. It then rises from 30 to 50 minutes ahead and finally falls for 60 minutes ahead. The minimum error is almost similar for all prediction times, with a little rise on 20 and 30 minutes ahead.

4.6 Historic Mean Baseline Model

In order to evaluate the performance of the above three models, a fourth model is generated which serves as the Baseline and is calculated in terms of the Historic Mean of the values. It is not a trained model, so its training time is not computed. However, the prediction error is discussed in the following subsection.

4.6.1 Prediction Error

The Historic Mean is computed over the same test data set which was used for above three model. The minimum, the maximum and the mean absolute errors are computed for all the stations over all the prediction time frames i.e. from 10 to 60

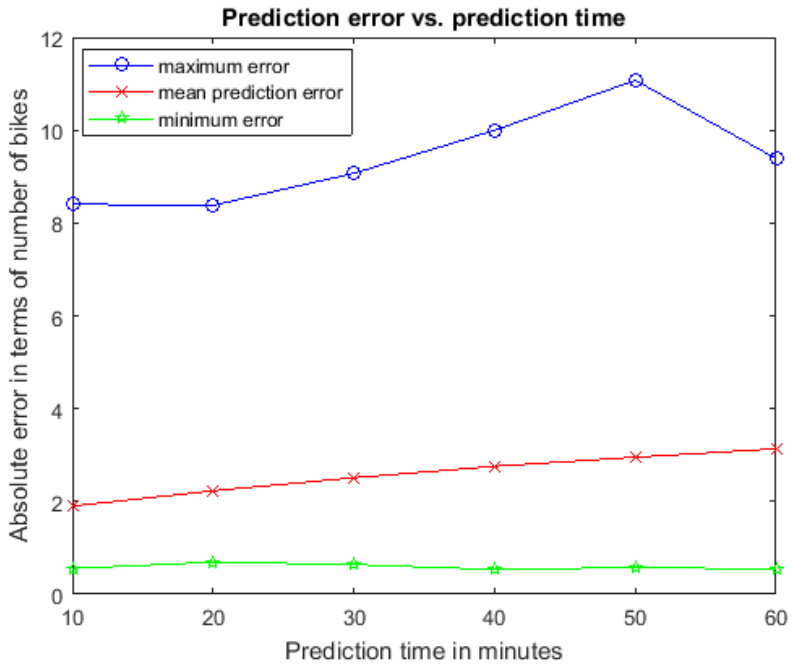

Figure 4.3: Average prediction error according to prediction interval, with minimum and maximum errors for Random Forest

minutes ahead as shown in Table 4.7.

Table 4.7: Prediction Errors by Historic Mean Baseline Model

Prediction Time	**Maximum Error**	**Mean Error**	**Minimum Error**
10 minutes ahead	3.305	1.186	0.046
20 minutes ahead	4.279	1.551	0.069
30 minutes ahead	4.896	1.857	0.092
40 minutes ahead	5.341	2.125	0.116
50 minutes ahead	5.721	2.367	0.14
60 minutes ahead	6.144	2.590	0.164

The graphical representation of the results in Figure 4.4 shows that the minimum, the maximum and the mean prediction errors increase as the prediction time increases.

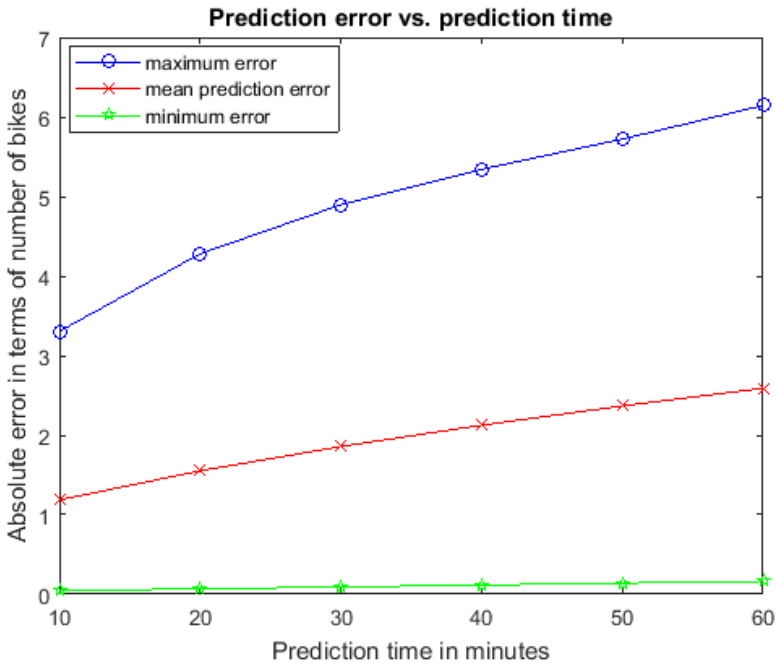

Figure 4.4: Average prediction error according to prediction interval, with minimum and maximum error for Historic Mean Baseline Model

4.7 Comparison of ARMA, Decision Tree and Random Forest

This section presents the comparative approach in order to identify the model which gives the best results. At first, the comparisons are made in terms of training time and which is then followed by the prediction error comparison.

4.7.1 Comparison in terms of Training Time

Figure 4.5 compares the time taken by the ARMA, the Decision Tree and the Random Forest models to generate the trained model from the training data set. It can be seen clearly that the training time duration for the ARMA model is 7 times that of the Random Forest Model and almost 5 times of the Decision Tree model. The least training time was consumed by the Random Forest Model (6 hours for generating 6 prediction models), while the ARMA model consumed the most (42 hours for generating 6 prediction models).

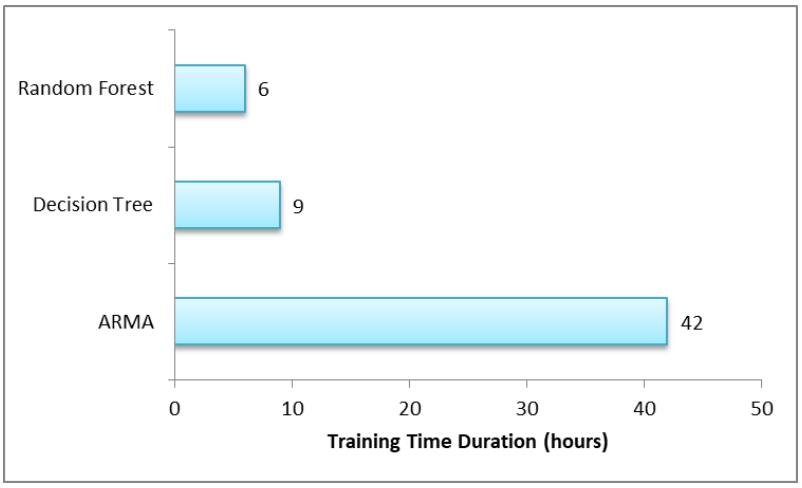

Figure 4.5: Comparison of ARMA, Decision Tree and Random Forest Models in terms of Training Time

4.7.2 Comparison in terms of Prediction Errors

The performance of all the models is compared in terms of the mean absolute error. Table 4.8 shows the mean absolute error of all the models for all the prediction times. The Random Forest algorithm showed the highest mean absolute errors for all the prediction time periods, hence its performance to predict the number of bikes is worst in our case.

On the other hand, an interesting fact was identified between the ARMA and the Historic Mean Baseline model. For the predictions of 10, 20 and 30 minutes ahead, the Baseline model showed the least mean absolute errors, while for higher time instances, such as 40, 50 and 60 minutes ahead, the performance of ARMA model was the best among all. Such results can lead us to the conclusions that the ARMA Model can be the best choice for predictions beyond 30 minutes, while for shorter terms, the Historic Mean can be considered as the best.

Figure 4.6 illustrates the graphical representation of the results. Random Forest Model shows the highest error for all the prediction time intervals. The second highest error is shown by the Decision Tree Model.

Table 4.8: Mean Absolute Errors of all the models

Prediction Time	ARMA	Decision Tree	Random Forest	Historic Mean
10 minutes ahead	1.296	1.228	**1.901**	**1.186**
20 minutes ahead	1.622	1.741	**2.229**	**1.551**
30 minutes ahead	1.892	2.125	**2.511**	**1.857**
40 minutes ahead	**2.121**	2.419	**2.757**	2.125
50 minutes ahead	**2.320**	2.716	**2.954**	2.367
60 minutes ahead	**2.498**	2.873	**3.133**	2.590

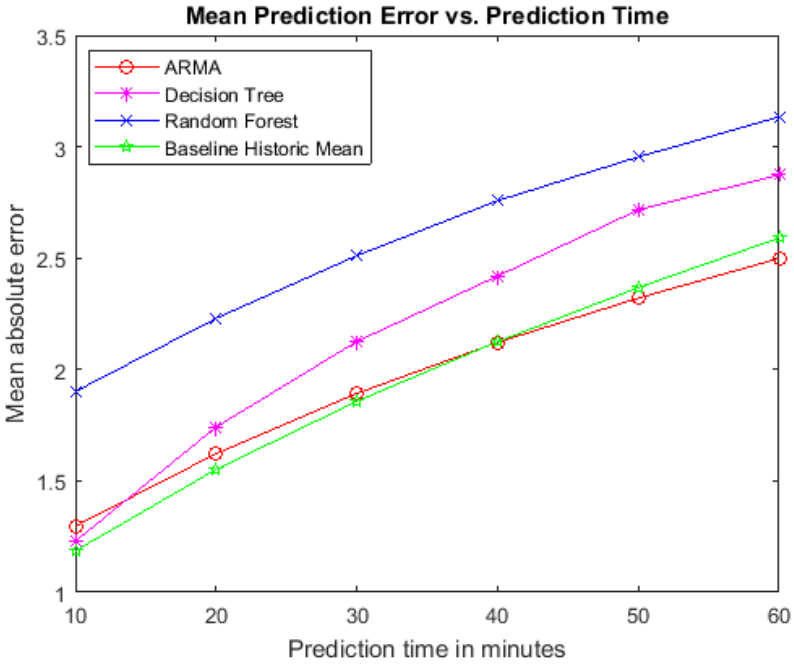

Figure 4.6: Average prediction error according to prediction interval, with minimum and maximum error

4.8 Discussion

It is clear from the results of all the models that both the Random Forest and the Decision Tree showed the highest errors in terms of the predictions. It is because of the fact that they are not suitable for such types of predictions or probably because the provided input was not enough for these models to generate the trained models. In [18] it is shown that Random Forest works better than ARMA Model but in that paper they also considered other inputs such as the information about weather and holidays. From such results it can be concluded that the Random Forest might

work better when more diverse sets of inputs are provided. In our case we had the data about the status of the stations only, and using that data for predicting the bicycle availability using Random Forest proved to be inaccurate.

The mean absolute errors of all the prediction models generated by the four algorithms were compared with each other to rank the performance of the models. The Random Forest algorithm showed the highest mean absolute errors for all the prediction time periods, hence its performance to predict the number of bikes is worst in our case.

On the other hand, an interesting fact was identified between the ARMA and the Historic Mean Baseline model. For the predictions of 10, 20 and 30 minutes ahead, the Baseline model showed the least mean absolute errors, while for higher time instances, such as 40, 50 and 60 minutes ahead, the performance of ARMA model was the best among all. Such results can lead us to the conclusions that the ARMA Model can be the best choice for predictions beyond 30 minutes, while for shorter terms, the Historic Mean can be considered as the best.

Behind the success of ARMA model is its long computational time. Prediction models with ARMA were trained in about 42 hours which is about 7 times greater than the Random Forest models.

The performance of Decision Tree model remained in between the ARMA and the Random Forest, both in terms of training time and also in the mean absolute errors. But its performance was less than the Historic Mean Baseline model.

From the results of ARMA and Historic Mean Baseline models, it can be concluded that ARMA works better for long time predictions (more than 40 minutes ahead). For short time predictions, the baseline line seems to be the best approach.

Although, ARMA model is expensive in terms of the computational time, but its results outperform the results achieved by the Decision Tree and the Random Forest.

Chapter 5

Conclusion and Future Work

5.1 Introduction

This chapter provides a brief overview of the study carried out in this thesis, including the problem statement and the major experiments carried out in this research. It then discusses the obtained results and the limitations of the study. Finally the future work is provided.

5.2 Problem Statement

With the increasing populations in cities, the need for better urban mobility systems is rapidly increasing. Bike sharing systems have been established all over the world as a means of environmentally-friendly transport [1].

Two problems have been widely discussed in the literature about bike sharing systems which are annoying from customer point of view.

1. Unavailability of bikes at stations when a user wants to rent a bike.
2. Impossibility to return the bike at a station due to unavailability of free parking slots.

The impossibility to rent a bike can be caused when the station is either completely empty, i.e. it does not contain any bikes, or some broken bikes are present which can not be used. Moreover, it could be impossible to return bikes when there are no free parking slots or when some slots are unusable due to any maintenance work.

Although in some bike sharing systems, there are trucks which balance the

bicycles by taking them from stations which are full or have more bicycles and leaving those bicycles in stations which are empty or have lesser bicycles, but a user who has already arrived at a station to pick a bike cannot wait for the trucks to get a bike. Similarly when a user arrives at a station which is completely full of bikes, he cannot wait for trucks to take some bikes so that he can park the bike. These situations are the causes of problems for the customers.

One possible solution to these problems is to enable the users to know beforehand about the availability of bicycles so that they can go directly to those stations where bikes/parking slots are available. This can be done by predicting the number of bikes at each station at some future point of time. For example, if a user needs to go to a station in 30 minutes, he should be able to know will there be any bikes/parking slots available at that time. The future prediction of the bicycle availability will greatly enhance the performance and reliability of the Bike Sharing Systems.

5.3 Discussion

The major research goal of the thesis is to study and compare some models to predict the availability of bikes in Bicing bicycle sharing stations some minutes ahead.

In this research, three learning models were considered, the ARMA, the Decision Tree and the Random Forest, based on their wide usage in the literature. The models are trained to predict the number of bikes in near future i.e. from 10 minutes ahead to 60 minutes ahead. Therefore, 6 prediction models were created for each of the three learning algorithms. The performance of these models was compared with the Historic Mean Baseline model. The models were compared with each other in terms of training time duration and in terms of performance which was measured by the mean absolute prediction error.

Chapter 4 showed the time consumed by these operators to train the models and the experimental results of all the trained models. The ARMA model took about 42 hours to generate the trained models for all the prediction time instance, i.e. from 10 minutes ahead to 60 minutes ahead, where as the total time consumption was 9 and 6 hours for the Decision Tree and the Random Forest Models respectively. It is clear that the time taken by the ARMA model is relatively much higher than the other two models.

Lets now discuss the experimental results by these models. According to

table 4.8, the mean prediction error of Random Forest was the worst among all models for all the prediction time periods. The results of Historic Mean Baseline showed the best performance for predictions from 10 to 30 minutes ahead, where as the ARMA Model exhibited the best performance from 40 to 60 minutes ahead predictions. The worst performance of the Random Forest can probably be due to two reasons:

1. Information of other parameters such as weather and holidays is not included in the data set.

2. The number of attributes used to train the models might not be enough for the Random Forest model. It may require more number of neighbor stations.

ARMA Model and the Historic Mean Baseline models showed interesting results. It can be concluded that for near future predictions (just few minutes ahead, 30 minutes in our case), the simple Historic Mean Model is the best choice, while for long term predictions, it is best to consider the ARMA Model.

5.4 Limitations

Few limitations exists in the current study:

- The current number of Bicing stations is over 420 but our study is based on 268 stations. We believe that by including more number of stations, better predictions can be made.

- The provided data set contained lots of erroneous data as discussed in Chapter 03. We believe that the provision of near-accurate and clean data can further improve the prediction accuracy.

- This study has shown predictions in near future only, i.e. from 10 minutes ahead to 60 minutes ahead. Long term predictions can contribute in enhancing the performance of Bicing.

5.5 Future Work

In future, the current study can be extended in the following possible ways:

- Chapter 1 described that there are Bicing trucks whose purpose is to take bicycles from stations which are full or have many bicycles and to leave those bicycles in stations which are empty or have small number of bikes. The

incorporation of knowledge about the interventions of Bicing trucks, their schedules and timings, and the number of bikes carried by these trucks can be included to improvise predictions.

- Information about other events that result in the deviation from normal Bicycle usage, can be considered to improve the prediction results such as, weather conditions, national holidays and festivals have great impact on the usage of bicycles, so information about all these aspects can be considered for future work.
- Additional features can be analyzed to further improve the accuracy of our predictions. In particular, long term predictions such as up to few days can be considered for future work.

5.6 Conclusion

In this work, we presented an analysis of the data of a public bicycle system in Barcelona, known as Bicing. In particular, the number of bicycles were predicted up to 60 minutes ahead of time.

Data pre-processing was the critical part of this research. Extensive efforts have been put in identifying the dirty data in the available data set and to perform data cleaning in order to ensure the correctness of the data to be trained and tested.

After reviewing the literature, we decided to use three automatic prediction algorithms for predicting the number of bikes: the ARMA, the Decision Tree and the Random Forest models. After the implementation of these models, a Baseline model based on the Historic Mean was considered to measure the performance of the three models.

After data cleaning, the total number of stations used in this study was 268. The training data set comprises of 800 continuous hours (from 2008-05-16 05:00 to 2008-06-27 06:58) with the timestamps separated by 2 minutes. The testing data set composed of non-overlapping data of 30 hours (from 2008-06-28 05:00 to 2008-06-29 15:58). The timestamps between 24:00 and 05:00 are not considered in this study. All the models were trained using the same data sets.

The performance of these models was calculated in terms of the absolute errors. The minimum, the maximum and the mean absolute errors were computed for all the models, including the Historic Mean Baseline model. Six prediction models were generated from each each of the four algorithms, one prediction model

for each prediction time. In total 24 prediction models were generated.

The mean absolute errors of all the prediction models generated by the four algorithms were compared with each other to rank the performance of the models. The Random Forest algorithm showed the highest mean absolute errors for all the prediction time periods, hence its performance to predict the number of bikes is worst in our case.

On the other hand, an interesting fact was identified between the ARMA and the Historic Mean Baseline model. For the predictions of 10, 20 and 30 minutes ahead, the Baseline model showed the least mean absolute errors, while for higher time instances, such as 40, 50 and 60 minutes ahead, the performance of ARMA model was the best among all. Such results can lead us to the conclusions that the ARMA Model can be the best choice for predictions beyond 30 minutes, while for shorter terms, the Historic Mean can be considered as the best.

Behind the success of ARMA model is its long computational time. Prediction models with ARMA were trained in about 42 hours which is about 7 times greater than the Random Forest models.

The performance of Decision Tree model remained in between the ARMA and the Random Forest, both in terms of training time and also in the mean absolute errors. But its performance was less than the Historic Mean Baseline model.

The results in this study have shown that the exact number of bikes are quite predictable for near future. The predictions can be useful for both the system administration and the users. From the point of view of the system administration, the predictions can help in re-balancing operation of the bicycles. The truck operators can exploit the information about the availability of bicycles in future at each station and can schedule the re-balancing operations beforehand. From the users' perspectives, the predictions can enhance user satisfaction and reliability and eventually will result in enhancing overall system performance.

Hence the research was concluded by considering the ARMA model to be the best among other three models i.e. the Decision Tree, the Random Forest and the Historic Mean Baseline models for predictions from 40 to 60 minutes ahead.

Bibliography

[1] Susan Shaheen, Stacey Guzman, and Hua Zhang. Bikesharing in europe, the americas, and asia: past, present, and future. *Transportation Research Record: Journal of the Transportation Research Board*, (2143):159–167, 2010.

[2] Bicing system information. https://www.bicing.cat/es/informacion/informacion-del-sistema. Accessed: 2018-06-18.

[3] Bicing. https://en.wikipedia.org/wiki/Bicing. Accessed: 2018-06-18.

[4] Andreas Kaltenbrunner, Rodrigo Meza, Jens Grivolla, Joan Codina, and Rafael Banchs. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455–466, 2010.

[5] Longbiao Chen, Daqing Zhang, Gang Pan, Xiaojuan Ma, Dingqi Yang, Kostadin Kushlev, Wangsheng Zhang, and Shijian Li. Bike sharing station placement leveraging heterogeneous urban open data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 571–575. ACM, 2015.

[6] Pilar Jiménez, María Nogal, Brian Caulfield, and Francesco Pilla. Perceptually important points of mobility patterns to characterise bike sharing systems: The dublin case. *Journal of Transport Geography*, 54:228–239, 2016.

[7] Mor Kaspi, Tal Raviv, and Michal Tzur. Bike-sharing systems: User dissatisfaction in the presence of unusable bicycles. *IISE Transactions*, 49(2):144–158, 2017.

[8] Pedro Pimentel de Vassimon. Performance evaluation for bike-sharing systems: a benchmarking among 50 cities. Technical report, 2016.

[9] Shoko Wakamiya, Yukiko Kawai, Hiroshi Kawasaki, Ryong Lee, Kazutoshi Sumiya, and Toyokazu Akiyama. Crowd-sourced prediction of pedestrian congestion for bike navigation systems. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 25–32. ACM, 2014.

[10] Mingsheng Wu and Vanessa Frias-Martinez. Crowdsourcing biking times. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM*

International Symposium on Wearable Computers, pages 1123–1131. ACM, 2015.

[11] Mohammad Abdur Razzaque and Siobhan Clarke. Smart management of next generation bike sharing systems using internet of things. In *Smart Cities Conference (ISC2), 2015 IEEE First International*, pages 1–8. IEEE, 2015.

[12] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[13] Jon Froehlich, Joachim Neumann, Nuria Oliver, et al. Sensing and predicting the pulse of the city through shared bicycling. In *IJCAI*, volume 9, pages 1420–1426, 2009.

[14] Ji Won Yoon, Fabio Pinelli, and Francesco Calabrese. Cityride: a predictive bike sharing journey advisor. In *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pages 306–311. IEEE, 2012.

[15] Romain Giot and Raphaël Cherrier. Predicting bikeshare system usage up to one day ahead. In *Computational intelligence in vehicles and transportation systems (CIVTS), 2014 IEEE symposium on*, pages 22–29. IEEE, 2014.

[16] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[18] Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. Predicting occupancy trends in barcelona's bicycle service stations using open data. In *SAI Intelligent Systems Conference (IntelliSys), 2015*, pages 439–445. IEEE, 2015.

[19] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. Dynamic cluster-based over-demand prediction in bike sharing systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 841–852. ACM, 2016.

[20] Pierre Borgnat, Patrice Abry, Patrick Flandrin, Céline Robardet, Jean-Baptiste Rouquier, and Eric Fleury. Shared bicycles in a city: A signal processing and data analysis perspective. *Advances in Complex Systems*, 14(03):415–438, 2011.

[21] Nicolas Gast, Guillaume Massonnet, Daniël Reijsbergen, and Mirco Tribastone. Probabilistic forecasts of bike-sharing systems for journey planning. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 703–712. ACM, 2015.

[22] Alexander Rubin. Geo (proximity) search with mysql. http://www.arubin.org/files/geo_search.pdf, 2006. Accessed: 2018-06-24.

[23] Mean absolute error. https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d. Accessed: 2018-07-16.