Investigation of hidden patterns in Qmatic Customer Journeys in order to minimize the average waiting time

By

D.D.U.B.Wijerathna

Registration No. : 2015/CS/152

This dissertation is submitted to the University of Colombo School of Computing
In partial fulfillment of the requirements for the Degree of Bachelor of Science
Honours in Computer Science
(SCS4124)

University of Colombo School of Computing
35, Reid Avenue, Colombo 07,
Sri Lanka

July 22, 2020

Declaration

I, D.D.U.B. Wijerathna (2015/CS/152), hereby certify that this dissertation entitled investigation of hidden patterns in Qmatic Customer Journeys in order to minimize the average waiting time, is entirely my own work and it has never been submitted nor is currently been submitted for any other degree.

Date Student's Signature

I, Dr. M. G. N. A. S. Fernando, certify that I supervised this dissertation entitled investigation of hidden patterns in Qmatic Customer Journeys in order to minimize the average waiting time, conducted by D.D.U.B.Wijerathna in partial fulfillment of the requirements for the degree of Bachelor of Science Honours in Computer Science.

Date Signature of Supervisor

Abstract

A queue management system is a critic component in any sector of business. In fact, all of the queue management systems have the same goal which is to minimize the waiting time for the customers who are in queues.

The aim of this study is to investigate the predictive variables which explain waiting time in bank virtual queues of Qmatic customer journeys. The Knowledge Discovery in Databases which is the standard data mining process, is employed as the methodology to reach the target of discovering the hidden patterns. As the data mining engine a GBM model is used for waiting time estimation. One of the most substantial measures in creating the regression model is selecting the optimal feature sets for predicting waiting time. According to the results, the best results obtained when learning rate equals to 0.1, max depth equals to 4, min sample leaves equals to 2 and max feature equals to 0.3. For the training processes, 10-Fold cross-validation is applied. The overall accuracy of the model is 71%.

Keywords: Qmatic Customer Journeys, Waiting time, Hidden patterns, Queueing Theory

Preface

The work presented in this study has utilized a queue management system, namely Qmatic in order to make an estimate and reduce the waiting time of Qmatic bank queues.

The objectives and aims of this study has not been explored by any other previous research of this particular domain. A novel design was introduced in order to facilitate the minimization of waiting time. There are several machine learning approaches for queue management such as Neural Networks and Deep learning models as well as approaches based on queuing theory . This dissertation explores a Gradient Boosting Machine (GBM) approach which uses unique set of features for Qmatic based on the elements of queuing theory.

Apart from these, the specific code segments mentioned under Chapter 4, the body of work mentioned herein is the work of the author of this document. The python codes are based on the work found in the shankarmsy repository on GitHub. The extended code segments can be found in the Appendices of this document. The evaluation was conducted by the author.

Acknowledgement

I would like to convey my sincere appreciation to my principal supervisor, Dr. Noel Fernando for his guidance and encouragement. without for his constant encouragement, this work would not have been possible. I am truly grateful for his unwavering support,

I would also like to extend my sincere gratitude to all the examiners and evaluators of my research for providing feedback on my research proposal and interim evaluation to improve my studies.

My sincere thanks go out to our final year computer science project coordinator Dr. H. E. M. H. B. Ekanayake for his encouragement and support in keeping this research focused and on-track.

Foremost my special thanks to my parents for providing me a solid foundation in education and all the courage and love gave me on every moment. They are the guiding stars which strengthen me to become the person who I am today.

Finally, I express my sincere appreciation to all my friends who supported and encouraged me on all cause of challenges I faced during this research. All the help given by everyone to make this research a success owns my great appreciation.

Contents

List of Figures

List of Tables

List of Acronyms

GBM	Gradient Boosting Machine
QT	Queueing Theory
SVM	Support Vector Machine
RF	Random Forest
SQL	Structured Query Language
MSE	Mean Squared Error
FCFS	First Come First Served
LCFS	Last Come First Served
MMRE	Mean Magnitude Relative Error
KDD	Knowledge Discovery in Database
CV	Cross Validation
GS	Grid Search

Chapter 1

Introduction

1.1 Background to the Research

The research is conducted for the Qmatic queueing management system. Qmatic is a queue management and customer journey management solution. They control the functioning in more than one hundred and twenty countries while having global headquarters in Sweden. Qmatic provides the solution which best met the needs of both customers and employees while bridging the virtual and physical world.
At present, the customers are checking in at a simple self served kiosk where an alphanumeric ticket is printed by a ticket printer. The type of service requested is correlated with the alphanumeric number(Figure 1.1).

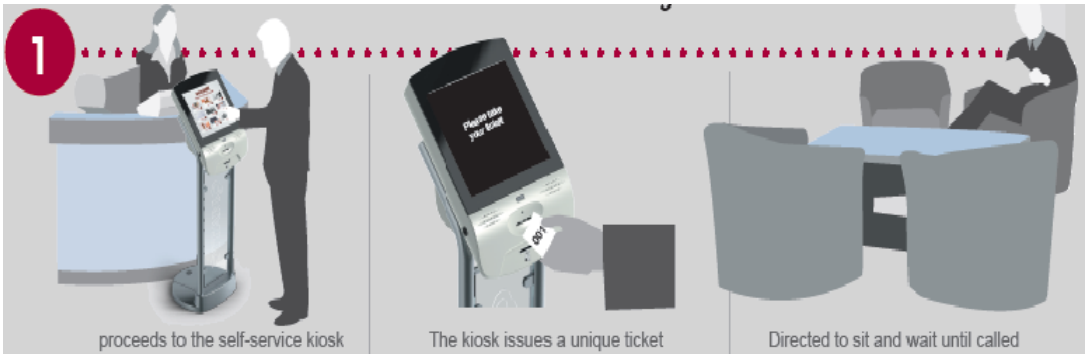

Figure 1.1: Step 1: Queueing behavior in Qmatic customer queues

Source: www.qmatic.com

Next, The customers are waiting in the virtual queue till they are called for the service. While the customers are in the queue, the Qmatic system manages to

which service counter, the customers should go and when. It eliminates the time spent unsure of where to go. Staff members are able to call, serve, and transfer customers without leaving the seats(Figure 1.2). The Qmatic solution allows the

Figure 1.2: Step 2: Queueing behavior in Qmatic customer queues

Source: www.qmatic.com

staff to easily manage the customer transactions from the time the customer check in for a service and up until the transaction closes(Figure 1.3). Each step of the

Figure 1.3: Step 3: Queueing behavior in Qmatic customer queues

Source: www.qmatic.com

transaction information of each customer is time stamped and reserved in the system in order to easily report and retrieve . If the waiting times and transaction times exceed the preset limits managers will receive alerts. This allows them to properly calibrate by expanding the number of workstations or adding additional staff(Figure 1.4).

Figure 1.4: Step 4: Queueing behavior in Qmatic customer queues

Source: www.qmatic.com

1.2 Research Problem and Research Question

This section elaborates the research problem in sub section 1.2.1 and the research questions in sub section 1.2.2.

1.2.1 Research Problem

Qmatic links people all over the world to the services. It posses a huge volume of data that is not used to maximize the advantages by making business decision and predictions. Therefore, there is a requirement in Qmatic to identify possible application areas based on the data available which leads for better forecasting.
In this context, an investigation will be done in order to estimate and reduce the average waiting time which will result in a remarkable customer experience while achieving the efficiency of delivering the service. The estimation of waiting time of a queue that has a considerably low number of journeys after taking whether the queue is a normal or a priority queue into consideration, the number of counters or the number of staff members needed to be in operation can be predicted which will result in a better utilization of resources to achieve desired profit margins.
Further, when there are lengthy waiting times, customers will not be satisfied and they will be in a situation where they could not decide whether to continue with the process or to quit because the queues are invisible to the customers. At the same time, customer may be called for a service prior to the provided initial waiting time. If the customer is not available at that moment, that the service will be

rollback.Therefore, it is perceived although the Qmatic system is having an initial waiting time for a service, it can be changed due to several reasons as follows.

- While the customer is in the queue, he decided to quit the service he requested. But until he is called for the service, quitting out from the service of a customer cannot be figured out.
- Due to the customer leaving the service, waiting time of other customers who came after that particular customer will be changed. It can be a time prior to the given initial waiting time.
- When there are transfers taken place due to mistakenly taking the wrong service ticket by the customer.

Therefore, identifying waiting time patterns and keeping the average waiting time to a minimum should be concerned.

.

1.2.2 Research Questions

- Research Question 1:
  Why does the given initial waiting time for a customer change after the customer checks in at a simple self served kiosk?
- Research Question 2:
  How can the average waiting time be predicted by condensing and evaluating the large amount of data which has already been collected in Qmatic system?
- Research Question 3:
  What will be the approach to minimize the waiting time according to the patterns observed and prediction?

1.3 Justification for the Research

The significance of this study lies in several aspects. Principally, the importance of this research is pointed towards the business domain and the computer science

research areas. According to the literature survey carried out, although there are several evidences of estimating the average waiting time of virtual queues, Qmatic queue management system does not have waiting-time estimation feature when the situations mentioned in 1.2.1 is taken place. Therefore, recognizing patterns of Qmatic customer queues and coming up with an average waiting time for queues and keeping it to a minimum value will be a significant contribution to the business domain as well as to the computer science research area. Qmatic customers will be benefited when waiting is properly tracked and managed. With this research, the staff members can focus on their job and will spend their time interacting with a more satisfied customer base. Apart from that, the identification of bottlenecks and other issue areas which slow down the productivity and wreak havoc on the operations will lead to the development of solutions to maximize efficiency and minimize waste. The opportunity given by this investigation in Qmatic queue management system to learn about their business and improve operational procedure will finally result in customer retention and sustainability of organization.

1.4 Design Approach

The research methodology is the systematic way of solving the research problem. As the research methodology, Knowledge Discovery in Databases (KDD)(Figure 1.5), which is the standard data mining process, is followed.

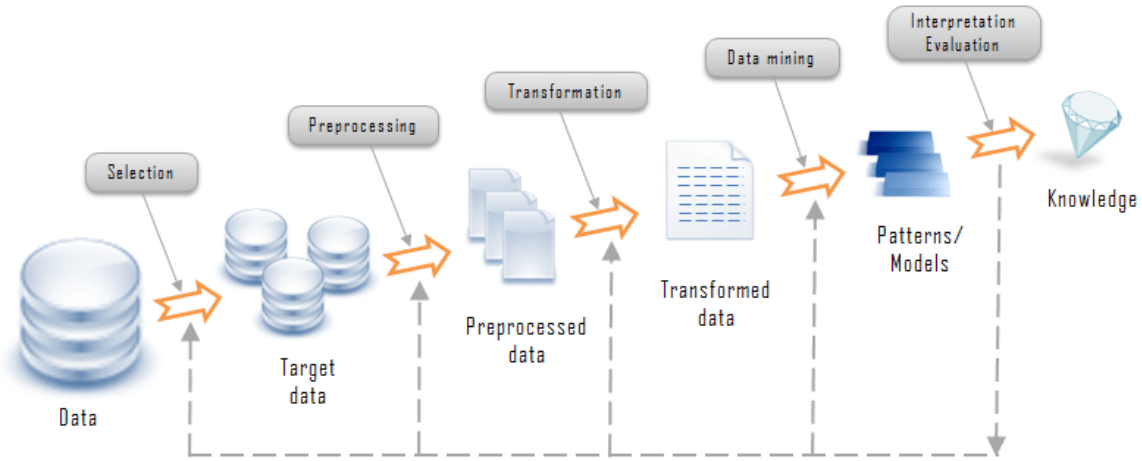

Figure 1.5: Knowledge Discovery in Databases (KDD)

This research is based on a particular bank data set containing information

about bank services which are generated and gathered by qmatic queue management and customer journey management system. The nature of this research is positivism where the problem is studied, then conduct experiments to identify and evaluate the causes that are influenced to the outcomes. This research can be categorized under an empirical study. In the empirical research life cycle contains of four stages such as observation, experiments, testing, and evaluation. In the observation phase, acquisition information from the data set should be carried out. In the pre-processing stage inconsistencies are eliminated and missing values are filled using mean value.In the feature selection phase, it removes features that are highly correlated with each other and if its values change very similarly to another. These features provide redundant information. Then, the collection of feature vectors are input to the learning model. Then the model will be trained to predict waiting time based on its feature vectors.Here, above mentioned data set is used as the data source which is commonly known as raw data. These raw data contains hidden patterns that can be interpreted while obtaining meaningful high-level knowledge from the raw data. In this research also, the input data definitely contains hidden patterns which are complex and are not sensible for the human eye. Due to that complexity, a machine learning approach is employed to address the interpretation and prediction process.

In addition to that, there are two broad categories of techniques available[2], memory-based approach where prediction is generated directly from examples without explicitly generalizing relationship between the predictive variables and the response; and rule-based (or model-based) approach where rules are established between the predictive variables and the response, ready to be applied to new values of predictor vectors. In this research, the rule-based approach is being employed to establish rules based on the quantitative features of the data set.

In a quantitative research, it is mainly focused on the design of a research method. In order to have better performance from this experiments, proper organization of the experiments should be considered. The design of the training model and experiments will be explained in Chapter 3 with more details. When creating the training data set, testing data set and validation data set, data set should be randomized in order to eliminate the bias. After having the experiments, testing and

evaluation process is executed to assert the degree of the achievements to calculate the accuracy.

1.5 Outline of the Dissertation

This dissertation is organized into six chapters as follows. This chapter is given a precise introduction to the research. Chapter 2 explores previous work related to this research and compare and contrast the different approaches used previously. Chapter 3 demonstrates the design of the proposed methodology. Chapter 4 provides the implementation details including the data set, tools, and formulas used in the research. Chapter 5 presents the evaluation results. The proposed methodology is concluded along with future works in Chapter 6.

1.6 Delimitation of Scope

This project is restricted to its scope of mitigating the estimated average waiting time in Qmatic customer queues, with an intention to work on predicting the number of staff members who are needed to be operated or the number of service counters that should be kept open to provide a quality service to the Qmatic customers while keeping the waiting time under a feasible value. Therefore, the research areas concerning the project are data mining, machine learning and pattern recognition.

This research will only consider the data generated and gathered by Qmatic queue management system and customer journey solutions. Another restriction of this research is, due to the time and resource limitations, data set containing banks' queueing details will be considered. Due to GDPR some of the data will be encoded.

1.7 Summary

Initially, this chapter has precisely introduced the background details of the research domain. As seen in the background, Qmatic queue management can be

taken as a major research area which aids the bank virtual queue monitoring process. However, there exists a knowledge gap to predict the waiting time of virtual bank queues and keeping it to a minimum. Therefore, the research questions are built through the identified problem and presented with the aim and objectives. Subsequently, the novelty of the research was justified. The proposed methodology was briefly described along with data analyzing, pre-processing, feature extraction, prediction and evaluation phases. Finally, the dissertation was outlined according to the chapters such as introduction,literature review, design, implementation,results and evaluation and conclusions.

Chapter 2

Literature Review

2.1 Introduction

The study focuses on minimizing the waiting time in Qmatic queue management system. Since this research belongs to queue management, a considerable amount of related works can be examined. As seen in the related works, data mining algorithmic approaches and approaches based on queueing theory are the traditional approaches that are followed by the researchers for waiting time prediction.

2.2 Theories

This section elaborates the queueing theory in sub section 2.2.1 and the theory of Gradient Boosting Machine in sub section 2.2.2.

2.2.1 Queueing Theory

Queueing Theory (QT) is a sub division of Operations Research, according to Hillier and Lieberman[11]. Queuing systems can be explained by some of the elements such as the arrival pattern of the customer, the service pattern of the staff, the number of staff members, the capacity of the system, and the discipline of the queue. Arrival pattern shows how the customers are arriving on the queue. An arrival pattern of a customer can be stochastic when there is a known distribution of probability which describes the gap among the arrivals of customers on a queue[12]. This metric is named as inter arrival time. How much time a particular customer spends with

one staff member is considered in the service pattern. Similar to arrival pattern, The service time is the metric related to this queue element. How many number of attendants are serving a queue is usually referred as the number of servers. The maximum number of customers who are served by a queue, is the capacity of the system. The default is infinite. The service order is the Queue discipline. Examples: First Come First Served (FCFS), Last Come First Served (LCFS), and General Discipline (GD). The elements of a queuing system is summarized by Kendall's Notation Formula[11] as follows.

$$a/s/n/c/d \tag{2.1}$$

Where a is the arrival pattern of the customers, s is the service pattern of the servers, n is the number of servers, c is the capacity of the system and d is the discipline of the queue. In most of the situations, the last two parts are suppressed. The assumptions are infinite capacity for the system and a First Come First Served discipline. M/M/1 is the simplest queue management system, where the M letters are corresponding to Markov Processes. These processes are related to exponential distribution for both arrival patterns and service patterns. For a M/M/1 queue management system, the traffic intensity is explained by the utilization factor.

$$\rho = \lambda/\mu \tag{2.2}$$

Where λ is the average rate of customer arrival and μ is the average rate of service of staff member. $1/\lambda$ is the inter arrival time and $1/\mu$ is the service time.
According to the work[11], when

$$\rho < 1 \tag{2.3}$$

the queuing system is in state of steady. It means the queue is operating normally. In this situation, it is possible to explore the queue management system by making use of probabilistic results. And also, the occupation of system or the ratio of time that the attendants are engaged is represented by ρ. When $\rho = 1$, the queue management system is operating at its fullest capacity. Finally, if

$$\rho > 1 \tag{2.4}$$

, the rate of customer arrival will grow faster than the rate of service provided and the queue will grow indefinitely.

2.2.2 The Theory of Gradient Boosting Machine

Gradient Boosting Machine is built on the Decision Tree algorithm[12], the technique of boosting , and the Gradient Descent algorithm. Because Gradient Boosting Machine uses the technique of boosting, it defines a set of weak models which is able to turn in to a strong model. In boosting technique, models are built in series and in each successive model the weights are adjusted based on the previous model's learning, where gradient descent works on reducing errors sequentially. Gradient Boosting Machine starts with a tree and the following trees are added according to a loss function. Each added tree makes a model regularization by reducing the value of the loss function. However, Gradient Boosting Machine tends to lead to over fitting[12]. In order to overcome this problem, Gradient Boosting Machine can modified its behaviour according to the importance of new trees added and by changing the parameters which are related to trees constraints.

1. Loss Function

   The loss function is a function used to ascertain the amount of loss and error, which shows the model's credibility. The smaller the loss function is, higher the model's accuracy is[4]. At the present time, there are different type of loss functions in GBM. The most common loss functions in regression applications are shown in Table 2.1.

Table 2.1: Most commonly used loss functions in regression applications

#	Name	Loss Function	Negative Gradient
1	Squared Loss Function	$1/2(y_t\text{-}f(x_t))^2$	$y_t\text{-}f(x_t)$
2	Absolute Loss Function	$[y_t\text{-}f(x_t)]$	$\text{sign}[y_t\text{-}f(x_t)]$

2. Gradient Descent Algorithm

   In supervised learning, it is supposed that there are N training samples where Xi is the feature vector, and Yi is the target variable of the sample. The target variable of a sample can either be a continuous value or a discrete value. The objective of machine learning is to find a mapping function F(X) between Xi and Yi by using the training data. In order to find the optimal function, a loss function is set for the model. Minimization of loss function is the way to obtain it.

   The optimization problem can be solved by Gradient Descent (GD) algorithm[4]. The loss function always reduces the fastest in the negative gradient direction. Gradient Descent decreases value of the function along the direction of negative gradient when optimizing.

3. Gradient Boosting Algorithm

   Boosting method generates base models sequentially [4].The steps of boosting method are shown below[4].

   - Step 1: Each sample has weights, initial value of weight is identical
   - Step 2: The training samples are learnt from Base-learner 1
   - Step 3: When the learning of training samples is completed, the weight of the wrong samples is increased, and the weight of the correct samples is decreased
   - Step 4: The training samples are learnt from Base-learner 2
   - Step 5: M base-learners are obtained by repeating Steps 2-4
   - Step 6:The results of the M base-learners are combined as the ultimate learning result.

2.3 Literature

There is a considerable amount of related work found in the context of waiting time prediction[12]. In a research where its purpose was to design a classification model to decrease the number of waiting time overflows on bank teller queues in Brazil[12],

authors have tested four predictive models that were able to provide a probability of time overflow. One is making use of a queuing theory's formula (QT) and the other three making use of data mining algorithms such as DL, GBM, and RF. In order to make the appropriate Queuing Theory model, the authors have used M/M/1 queue configuration because the most queues that were used at this bank bear the main characteristic of this simplification. In order to have a verification on which queues on data are having M/M/1 characteristics, a Kolmogorov-Smirnov Goodness of Fit Test was used. This would help increase the acceptability of the results generated in QT model. Due to the constraints imposed by the Queueing Theory, researchers proposed other three models. These three models made use of the H2O platform for easy parallel processing and for grid search to find the optimal hyper parameters of each model which results in the most accurate predictions. In models' evaluation stage, performance of the models were analyzed by making use of two metrics such as the accuracy and F1-Measure. The results showed the Gradient Boosting model[12] as the most efficient model because it held an 97% of accuracy and a 75% of F1-measure. The QT model possess an accuracy of 78% and a 12% of F1-Measure, identifying only 32% cases of time overflow . When compared to QT model, models that used data mining algorithms generated satisfactory performance[12].

In another research, authors have employed a gradient boosting regression tree method (GBM) in order to examine methodically in detail and model freeway travel time to enhance the accuracy of prediction and interpretability of the model. The gradient boosting tree method strategically connects additional trees by rectifying mistakes made by its previous base models, therefore, potentially enhances accuracy of the prediction[20]. In a study which purpose has been to implement a model that conjugates the regular operational information with predictive capabilities and allows the owners of a telecommunications store located in Braga, Portugal[3] to regulate the quality of service and the satisfaction levels of customers of the referred store in useful time. Because the telecommunications store is having queuing models of unlimited length in the store but, a finite number of population of source for the existing queue management system, store faces problems coming often from low quality service in customer attendance processes. Therefore, in order to conceive

an adjustable predictive model that fits the operational requirements, the authors have used several data mining techniques and applied over a real data sample; a meaningful extracted from the service records maintained in the operational system of store. First authors have provided a detailed explanation of the case study that they are addressing and have exposed the way they prepared it for analysis. In the data preparation stage, features that are oriented to the formation of service profiles of store and patterns of customers' behavior, were extracted. Then they have performed some initial prediction tests by using one month (February 2016) of attendance service data. After the initial statistical analysis, researchers have designed and implemented two distinct data mining processes such as a classification process and an association process. The classification process was designed with an intention to provide a real picture about how the several attendance services were made in the past. Decision Tree J48 algorithm provided by Weka was used with a training level of 10 for this process. The association process which was designed to know the several aspects that could lead to a certain type of attendance service, was performed using the Apriori algorithm given by Weka with a minimum support bound of 0.1, delta value of 0.05 and a metric confidence interval of greater than 60%. According to the given facts, it can be agreed that the approach is commendable. However, the paper lacks explanation of most underlying concepts which would add value to this paper if they were presented.

A significant study was done by Ibrahim and Whitt[9] to predict waiting time for customer service. A parametric approach is employed by deriving a formula to predict the waiting time. As a result, several modified predictors based on queue length (QL) and the elapsed waiting time of the customer were tested[9]. They proposed alternative real time delay predictors for non stationary many server queue management systems and showed that they are effective in the M(t)/M/S(t) + GI queueing model which has time-varying arrival rates and a time-varying number of servers. With an aim to explore the predictive variables that explain the duration of waiting time in bank customer queues, in a significant study, researchers have used fast artificial neural network engine. In order to train the neural network, they have employed a resilient propagation[8]. When considering the research on minimizing the customers' waiting time and gaining more profit in restaurant, the

researchers have used two concepts in queuing theory such as Little's theorem and Queuing Models & Kendall's Notation[10]. They have derived the rate of arrival, rate of provisioning the service, rate of utilization, waiting time in queue and the probability of potential customers to balk based on the data using Little's theorem and M/M/1 queuing model. The research on prediction of queuing behavior through the use of artificial neural networks illustrates that the artificial neural networks trained by a supervised learning algorithm, and combined with queueing theory could be used to optimize staff scheduling while notifying the requirement of more training and testing [14].

Another related study used a simulation model to estimate the virtual waiting time for a passenger arriving at a given time in airport[19]. It illustrates how the variability of daily waiting-time averages are different from the variability of individual customer experiences. An overview of state of the art Customer Experience Management (CEM) requirements and challenges facing are provided in a study where they discuss an artificial intelligence driven concept in vision of autonomous CEM. In the state of the art CEM, it necessitates the need for an integrated approach of two CEM components such as Network Quality of Service and Customer Quality of Experience while introducing CEM with Data Analytics[5].

The Emergency service of a public hospital is investigated by applying the relations and concepts of queues because there was a significant raise in the number of patients waiting in line for assistance in the hospital's emergency area[15]. In the research on waiting time analysis of pharmaceutical services with queuing method in a hospital, the authors have identified a model of queuing management system, and any factors that influence waiting time of services of outpatient pharmacy. This research is a quantitative research which uses a descriptive observational research method from samples taken and to support this approach.[13].

In the work of queue mining for predicting the delay in multi-class service processes researchers establish a queueing perspective in process mining to address the problem of predicting the delay, which refers to the time which the execution of an activity for a running instance of a service process is delayed because of the queueing effects [18]. They presented predictors that treat queues as first-class citizens and either enhance existing regression-based techniques for process mining

or are directly grounded in queueing theory. In the research on feature selection in principal component analysis of analytical data, researchers have proposed to select a subset of variables in principal component analysis (PCA). It helps to preserve as much information exist in the complete data as possible. The information is measured by means of the percentage of consensus in generalized Procrustes analysis. By applying a genetic algorithm (GA),the best subset of variables is obtained with an intention to optimize the consensus between the subset and the complete data set while avoiding the exhaustive searching [7] article.

2.4 Summary

This chapter focuses on investigating extensive details of related works including the performance and drawbacks of each approach. The potentials of these approaches vary from problem domain to domain. While reviewing the literature, the best approach that is compatible with the problem domain was observed as follows.

The majority of related works subject to Queueing theory and machine learning approaches where it uses data mining techniques to build a model. Most often, Queueing theory has been adapted in waiting time prediction. Due to the constraints imposed by the Queueing Theory, Researchers have proposed other models that were analyzed by making use of data mining algorithms where best results were obtained in GBM model[12]. Hence, a Regression Gradient Boosting Algorithm based machine learning approach is used in this context. Since the proposed approach, GBM, is a supervised learning method, it defines a loss function and minimizes it. The Mean Squared Error (MSE)[16] is defined as the loss function. By using gradient descent and updating the predictions based on a learning rate, values where loss function is minimum can be found.

Chapter 3

Design

3.1 Introduction

This chapter elaborates the overview of the research design for the proposed methodology. Section 3.2 described how the data is structured. Figure 3.1 shows a high-level diagram of the design of the methodology. As shown in the diagram, the design has five main components as data collection, data pre-processing, data transformation, the design of the gradient boosting regression model for prediction and evaluation. Section 3.3 describes all the details of analyzing the bank data set and the features that are selected. Section 3.4 explains the data pre-processing component with all considerations and construction steps. The data transformation process is outlined in Section 3.5. Section 3.6 illustrates the design of the regression model and section 3.7 briefly mentions the evaluation plan.

3.2 Data set

The bank data set of Qmatic queue management is the primary data source of this research. It is an aggregated data set with structured data. Statistics of the customer journey is recorded in the database. The time that the customer check in for the service through getting the service done and customer feedback with whether customer is transferred for another service are captured and stored. This primary data set consists of 100000 records of waiting time of bank services. At the beginning of the research, the data set is annotated by observing it manually.

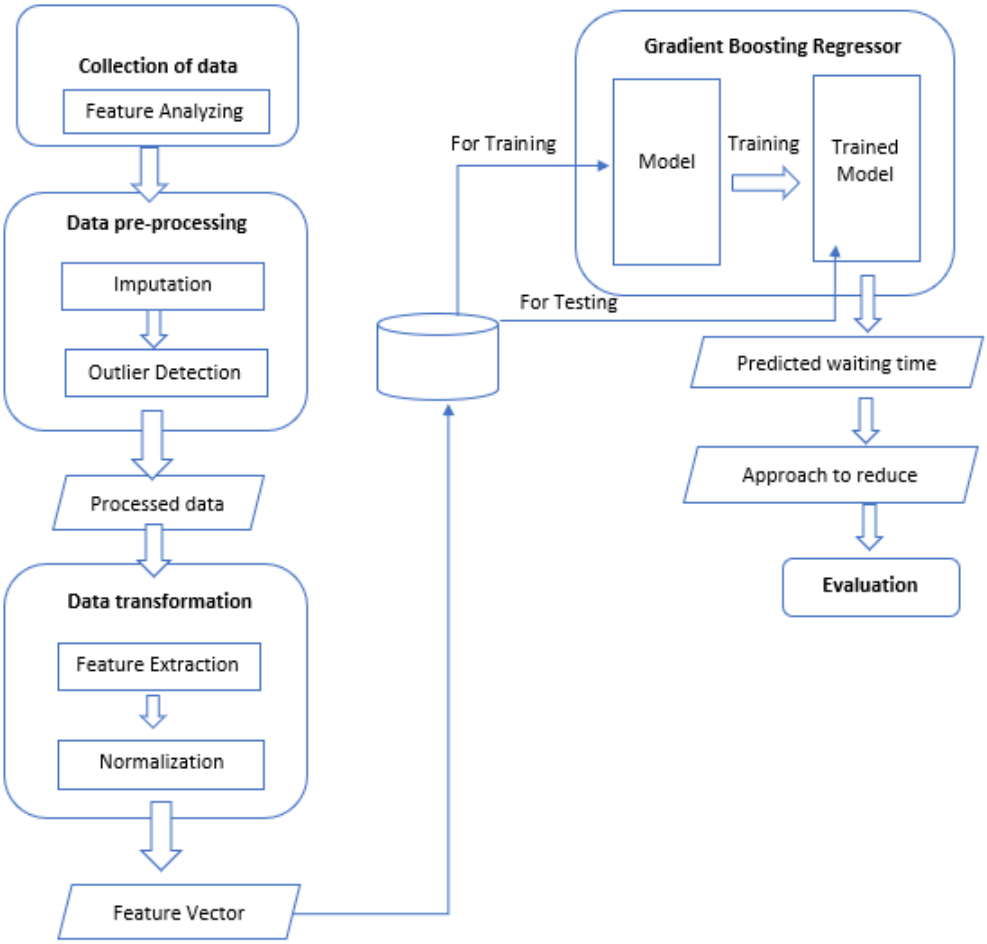

Figure 3.1: The abstract image of the research architecture

Initially, the whole data set is divided as 60% and 40% for training and testing purposes respectively as depicted in Figure 3.2.

3.2.1 Training Data set

The training Data set consists of 60000 records.As shown in Figure 3.2, training data set is again divided into 70% for train the regression model and the rest 30% to get the training accuracy of it.

The Cross-Validation (CV) technique is employed.

3.2.2 Testing Data set

40% of the data set is belong to the testing data set which includes 40000 number of records from whole data set (Figure 3.2). The important fact is that this testing data is never being in the training data set.

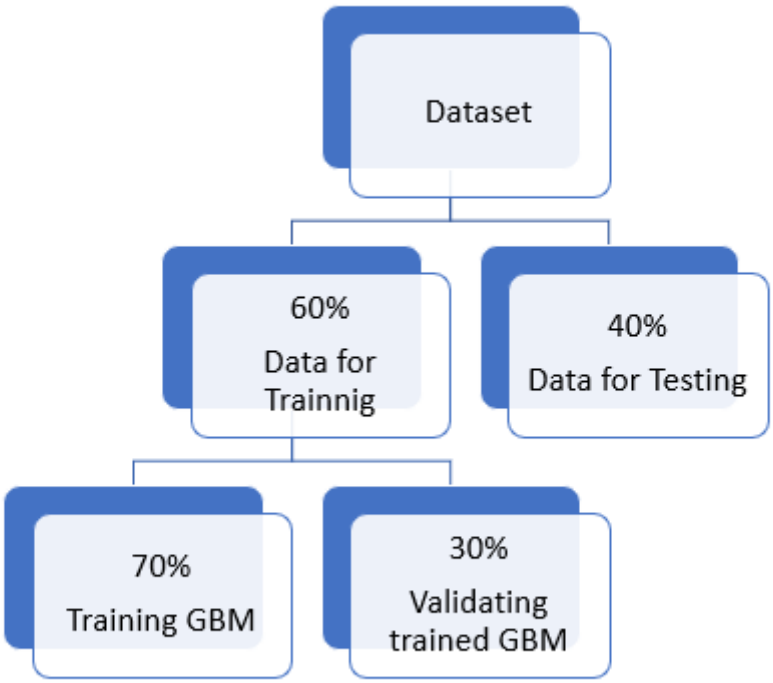

Figure 3.2: Data set

3.3 Data Collection and Feature Analyzing

The Features are used to capture the most useful information in the data set. In this regression model, identifying the most appropriate features is essential to predict waiting time. In order to select the appropriate features, the data should be analyzed well.

First, sample data set gathered by Qmatic System was observed well. After observing the sample data set, predictive variables that were identified through the literature review were queried out by using SQL statements. Below Figure 3.3 shows the final query and results in the Microsoft SQL server management studio by joining all other data with the visit transaction table. Table 3.2 shows features used in the study.

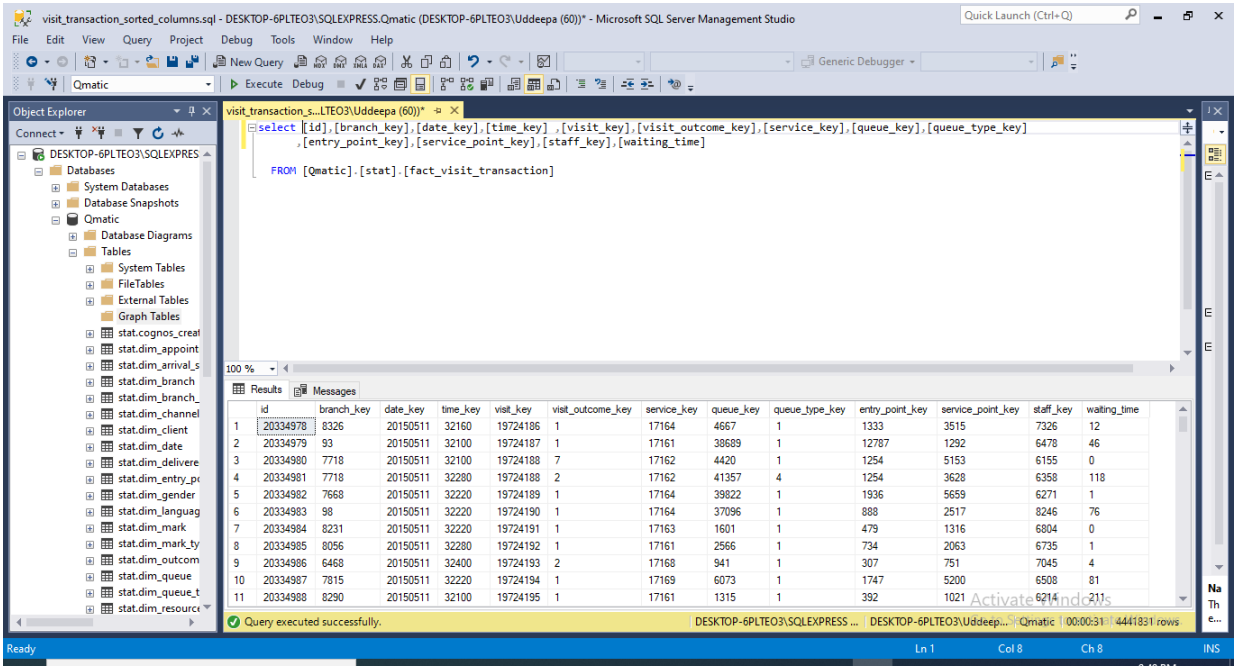

Figure 3.3: Sample Data set

Table 3.1: Features used in the study

#	Name	Description
1	Branch	Branch of the bank
2	Date	Date of transaction
3	Time	Time of transaction time
4	Visit Outcome	The previous customer's outcome. eg: normal, no show, transfer
5	Service Type	Service required by the customer
6	Queue Type	Type of the queue that the customer is assigned
7	Service Point	Location where the ticket is obtained
8	Staff Member	Staff member who serves the customer
9	Waiting Time	Target variable

3.4 Data Pre-processing

Next Phase is data pre-processing. Pre-processing prepares the data in a consistent way before extract the features. Data pre-processing stage consists of imputation and outlier detection that are discussed in Sections 3.4.1 and 3.4.2 respectively.

3.4.1 Imputation

The missing values of the data set that were considered for the experiment were denoted with null value or '?' value were first encoded as np.nan and were imputed by using the statistics (mean) of each column which the missing values are positioned. The SimpleImputer class in scikit-learn library which is a univariate feature (that imputes values in the i-th position of feature dimension by making use of only the non missing values in that feature dimension) was used.

3.4.2 outlier detection

In outlier detection phase, the Outlier detection estimator tries to fit the regions. It ignores the deviant observations which can be interpreted as observations far from the other observations. In this study, 'isolation forest' in scikit-learn library was used as the Outlier detection estimator. Here, partitions were generated by choosing a feature randomly and then choosing a split value randomly between the minimum value and maximum value of the selected feature.

In the case of Isolation Forest, anomaly score is defined as:

$$s(x, n) = 2^{-E(h(x))/c(n)} \tag{3.1}$$

where the path length of observation x is denoted by h(x), the average path length of unsuccessful search in a Binary Search Tree(BST) is denoted by c(n) and the number of external nodes is denoted by n.

3.5 Data Transformation

In the KDD process, the next Phase is data transformation. Data transformation stage consists of feature extraction and normalization that are discussed in Sections 3.5.1 and 3.5.2 respectively.

3.5.1 Feature Extraction

The subset selection of features is carried out by identifying and pulling out as much irrelevant and redundant features as possible from the training data set. This de-

creases the dimensionality of the data and may enable regression algorithms[6] to operate faster and more effectively. Feature wrappers often achieve better desired results than filters because of they are tuned to the specific interaction between an induction algorithm and its training data[1].At the same time, wrapper based feature selection methods have high chance of over fitting because it involves training of machine learning models with different combination of features. Therefore, an optimal features set is obtained by making use of both Wrapper and filter method[1]. for feature selection.

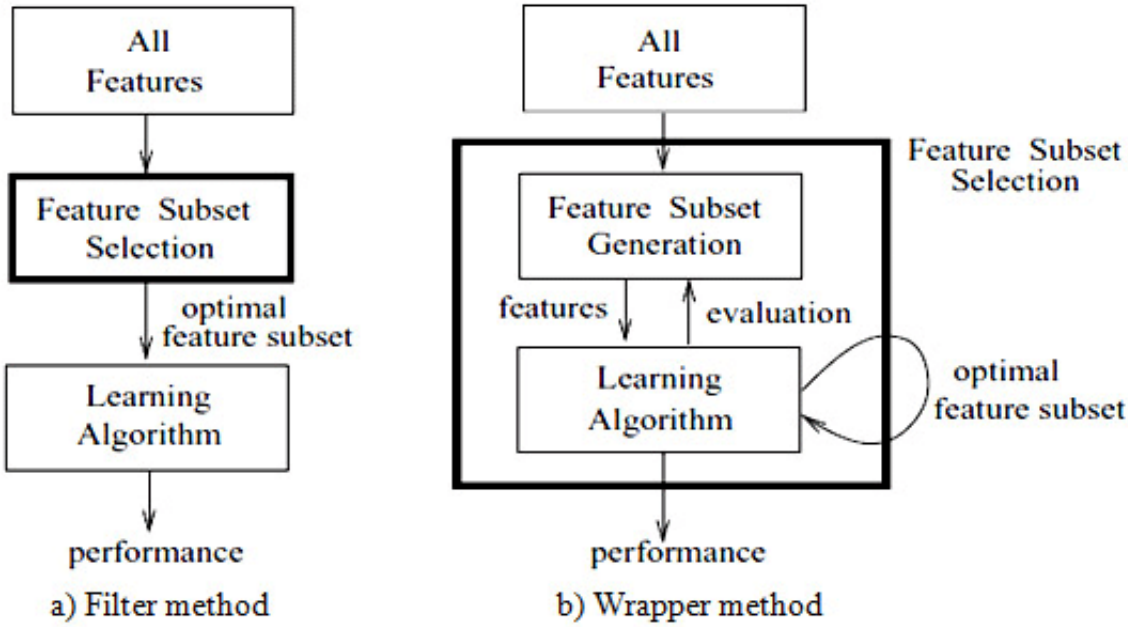

Figure 3.4: Feature Subset Selection Methods

Table 3.2 shows features used in the study.

Table 3.2: Features used in the study

#	Name	Description
1	Branch	Branch of the bank
2	Date	Date of transaction
3	Time	Time of transaction
4	Visit Outcome	The previous customer's outcome. eg: normal, no show, transfer
5	Service Type	Service required by the customer
6	Queue Type	Type of the queue that the customer is assigned
7	Service Point	Location where the ticket is obtained
8	Staff Member	Staff member who serves the customer
9	Waiting Time	Target variable

3.5.2 Normalization

The goal of normalization phase is to change the values of numeric columns in the data set to a common scale, without misrepresenting differences in the ranges of values. Because the extracted features of the data set used in this study have different ranges, it was required to normalize. The normalizer class in scikit-learn library was used where it normalizes samples individually to unit norm.

3.6 Design of Regression Model

The goal is to investigate hidden patterns in Qmatic customer journeys to predict and reduce the waiting time in bank virtual queues. As concluded in the literature review, Gradient Boosting Model is selected as the best approach which fits into the problem domain. This section demonstrates the designs of GBM
When illustrating the regression model, Gradient Boosting Machine was designed. It is based on the Decision Tree algorithm, a Boosting technique (Figure 3.5), and the Gradient Descent algorithm.

Because GBM uses Boosting, that defines that a group of weak models can turn out to a strong group. In Boosting, models are constructed in series. And, in each successive model, the weights are adjusted based on the learning curve of previous

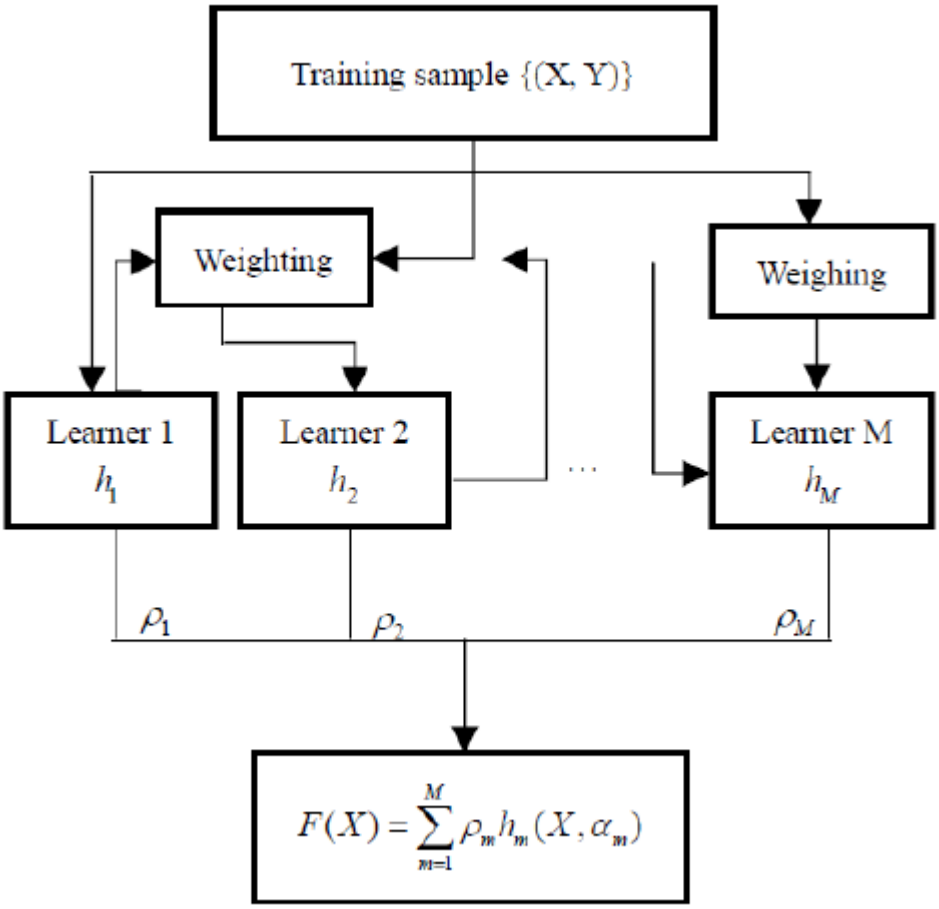

Figure 3.5: Boosting

model where gradient descent works on reducing errors sequentially.

GBM begins with one tree and then adds the subsequent trees according to a loss function. Each added tree makes a regularization of the model by minimizing the value of the loss function. However, GBM tends to lead to over fitting[12]. To overcome this problem, the behavior of GBM can be modified by changing its parameters which are related to trees constraints. Apart from that according to the importance, new trees are added to the model. Therefore, grid feature which generates multiple models by varying its parameters is used to get the optimal hyper parameters of a model which results in the most accurate predictions.

More precisely, Gradient Boosting Regression algorithm can be explained for waiting time prediction of Qmatic bank queues as follows.

1. The average value of waiting time is calculated (target label)

   When it is the case of tackling regression problems, a leaf is the starting

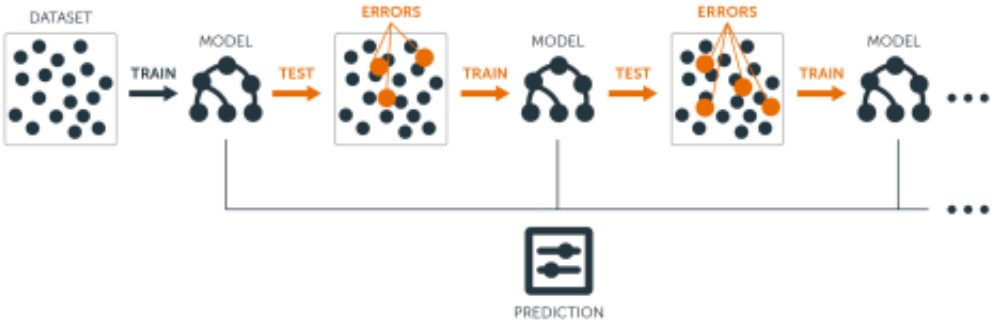

Figure 3.6: Gradient Booting Regression

point. It is the average value of the variable to be predicted,. This leaf is used as a baseline to approach the correct solution in the following steps.

$$Average = W(x)/N(x) \tag{3.2}$$

where W(x) is the sum of the values of all target variables and N(x) is the number of records

2. The residuals are calculated

   For each and every sample,the residual is calculated with the following formula.

   $$residualV = actualV - predictedV \tag{3.3}$$

   the predicted value is identical to the mean that is calculated in the previous step and the actual value can be get from the waiting time column of each sample.

3. A decision tree is constructed

   A tree with the aim of predicting the residual values is built. In other words, every leaf in the decision tree will contain a prediction as to the residual value

4. The target label is predicted by making use of all of the trees within the ensemble

   Each record is passed through the decision nodes of the tree that is newly formed, up until it reaches a given lead. The residual value in previously

mentioned leaf is used to predict the waiting time

$$predicted = Average + (\eta * R(x)) \tag{3.4}$$

Where η is learning rate and R(x) is residual predicted by decision tree

5. The new residuals are computed

   A new residual values set is computed by subtracting the actual waiting time from the predicted waiting time made in the previous step. Then, the residual values are used for the next decision tree's leaves as described in step 3.

6. The steps 3 to 5 is repeated until the number of repetitions matches the number specified by the hyper parameter.

3.7 Evaluation

There will be a quantitative evaluation. The intention of the quantitative evaluation is to analyze the accuracy of the prediction of waiting time. These analysis have been designed to address the research questions that were posted in Section 1.2.2

3.8 Summary

With this chapter the conceptual overview of the research design has been discussed. In each subsequent sections, the individual components in the research design were outlined with a description of the connectivity between these components. Also, a brief opening of the evaluation plan was given. The justifications to the selections and all considerations were mentioned within components. The implementation of these components will be offered further in Chapter 4.

Chapter 4

Implementation

4.1 Introduction

This chapter addresses the implementation details of the proposed design. Section 4.2 outlines the tools that are used in the process. Section 4.3 establishes the implementation details of pre processing, data transformation, and GBM model.

4.2 Software Tools

This Section outlines the tools that are used in the process.

4.2.1 Scikit-learn

Scikit-learn is a simple and efficient library used in python for machine learning and data mining. It provides functionalities such as model selection, dimensionality reduction, clustering, classification and regression.

4.2.2 Numpy

NumPy is the fundamental package for scientific computing with Python. It is also used as an efficient multi-dimensional container of generic data..

4.2.3 Matplotlib

'Matplotlib' is a two-dimensional plotting library in python which provides publication quality figures. This is a numerical extension of 'Numpy' library. For a simple plotting 'pyplot' module provides interfaces which are very close to Matlab.

4.2.4 Pandas

pandas is a powerful and easy to use open source data analysis tool, that is built on top of the Python programming language.

4.2.5 Weka

Waikato Environment for analysis of knowledge which is a machine learning software, is written in Java language. It is used for feature ranking.

4.2.6 SQL Server Management Studio

SQL Server Management Studio is an integrated environment for managing any SQL infrastructure. The obtained data set was imported to 2017 version and queried out related features.

4.3 Implementation Details

This section establishes the implementation details of pre processing, data transformation, and GBM model.

4.3.1 Pre Processing

In data pre processing, Imputation and outlier removal were performed. This has already been described in the sub sections 3.4.1 and 3.4.2. The code segments related to imputation (Figure 5.1) and outlier removal (Figure **??**) are as follows.

```
In [1]: #import packages
        from sklearn import preprocessing
        import numpy as np
        import pandas as pd
        from sklearn.model_selection import train_test_split
        import seaborn as sns
        import matplotlib.pyplot as plt
        from scipy import stats
```

Loading data

```
In [2]: dataset = pd.read_excel (r'featurelistFinal.xlsx')
```

Figure 4.1: Data Loading

Imputation

```
In [ ]: # dataset has 'NULL' and '?' in it, convert these into NaN
        dataset = dataset.replace('NULL', np.nan)
        dataset = dataset.replace('?', np.nan)
```

```
In [ ]: from sklearn.impute import SimpleImputer
```

```
In [ ]: # replace missing values, encoded as np.nan, using the mean value of
        #the columns (axis 0) that contain the missing values:
        imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
```

```
In [ ]: dataset = imp_mean.fit_transform(dataset)
        print (dataset)
```

Figure 4.2: Imputation

outlier removal

```
In [ ]: from sklearn.ensemble import IsolationForest
```

```
In [ ]: clf = IsolationForest(n_estimators=100, warm_start=True)
        clf.fit(dataset)  # fit 10 trees
        #clf.set_params(n_estimators=20)  # add 10 more trees
        #clf.fit(dataset)
```

```
In [ ]: print(dataset)
```

Figure 4.3: Outlier Removal

Normalization

```
In [ ]: normalizer = preprocessing.Normalizer().fit(dataset)

In [ ]: # Creating pandas dataframe from numpy array
        dataset = pd.DataFrame({'Branch': dataset[:, 0], 'Date': dataset[:, 1],'Time': dataset[:, 2],
                                'Visit_Outcome': dataset[:, 3],'Service_Type': dataset[:, 4],
                                'Queue_Type': dataset[:, 5],'Service_Point': dataset[:, 6],
                                'Staff': dataset[:, 7],'Waiting_Time': dataset[:, 8]})
        #, 'Column11': dataset[:, 10], 'Column12': dataset[:, 11],'Column13': dataset[:, 12]

In [ ]: print(dataset)
```

splitting the dataset into inputs and output

```
In [ ]: #until the last column #The iloc indexer for Pandas Dataframe is used for
        #integer-location based indexing / selection by position
        sourcevars = dataset.iloc[:,:-1]

        #last column
        targetvar = dataset.iloc[:,-1]
        print(targetvar)
```

Figure 4.4: Normalization

Correlation of attributes

```
In [ ]: f, ax = plt.subplots(figsize=(10, 6))
        corr = dataset.corr()
        hm = sns.heatmap(round(corr,2), annot=True, ax=ax, cmap="coolwarm",fmt='.2f',
                         linewidths=.05)
        f.subplots_adjust(top=0.93)
        t= f.suptitle('Waiting Time Attributes Correlation Heatmap', fontsize=14)
```

Figure 4.5: Feature Extraction

4.3.2 Data Transformation

In data transformation, feature extraction and normalizing were performed. This has already been described in the sub sections 3.5.1 and 3.5.2. The code segments related to feature extraction (Figure **??**) and normalizing (Figure **??**) are as follows.

Feature Extraction

In this phase, feature subset selection was carried out by using Weka application's wrapper subset evaluator. Then, the correlation heat map was plotted in order to depicts the correlation of independent variables with the target variable waiting time. The features which has correlation of above 0.5 (taking absolute value) with the target variable were selected

Training

In [ ]:
```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_validate
from sklearn.naive_bayes import GaussianNB
from sklearn.datasets import load_digits
from sklearn.model_selection import learning_curve, GridSearchCV

def GradientBooster(param_grid, n_jobs):
    estimator = GradientBoostingRegressor()
    cv = ShuffleSplit(n_splits=10, test_size=0.3, random_state=0)
    classifier = GridSearchCV(estimator=estimator, cv=cv, param_grid=param_grid, n_jobs=n_jobs)
    classifier.fit(X_train, y_train)
    # best estimator that was found by GridSearchCV
    print ("Best Estimator learned through GridSearch")
    print (classifier.best_estimator_)
    return (cv, classifier.best_estimator_)
```

In [ ]:
```
param_grid={'n_estimators':[100], 'learning_rate': [0.1],# 0.05, 0.02, 0.01],
            'max_depth':[4],#4,6],
            'min_samples_leaf':[2],#,5,9,17],
            'max_features':[0.3],#,0.3]#,0.1],
            }
n_jobs=4
 #Let's fit GBRT to the digits training dataset by calling the function we just created.
cv,best_est=GradientBooster(param_grid, n_jobs)
```

Figure 4.6: Training the GBM

4.3.3 Gradient Boosting Model

A cross validation generator which is a built-in package in scikit-learn, was used to train the model by tuning the parameters based on a cross-validation subset that is picked from within the training set. If the parameters are tuned against the test data set, it will end up biasing towards the test set and will not generalize very well. A different cross-validation subset was picked for each repetition, The number of repetitions was given explicitly. Next,the cv/train splits were used and a grid search function was executed that evaluates the model with each split and tune parameters to give the best parameter which gives the optimal result.

Testing

In [ ]:
```
model_score = model.score(X_train,y_train)
# Have a look at R sq to give an idea of the fit ,
# Explained variance score: 1 is perfect prediction
print('Train Variance: ',model_score)
y_predicted = model.predict(X_test)
print("predicted waiting time")
print(y_test)
print(y_predicted)
# The mean squared error
#print("Mean squared error: %.2f"% mean_squared_error(y_test, y_predicted))
# Explained variance score: 1 is perfect prediction
print('Test Variance score: %.2f' % r2_score(y_test, y_predicted))
```

In [ ]:
```
fig, ax = plt.subplots()
ax.scatter(y_test, y_predicted, edgecolors=(0, 0, 0))
ax.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], lw=1)
ax.set_xlabel('Actual Waiting Time')
ax.set_ylabel('Predicted Waiting Time')
ax.set_title("Ground Truth vs Predicted")
plt.show()
```

Figure 4.7: Testing the GBM

4.4 Summary

This chapter presents the utilized software tools and libraries followed by the important characteristics of them. Then, the implementation details of the major components in the proposed design were described in each section including codes of the functionalities. Pre-processing, data transformation, GBM model are the major components

Other code segments of the research are included in Appendix A.

Chapter 5

Results and Evaluation

5.1 Introduction

This chapter illustrates the success level of the proposed methodology with the results. Section 5.2 presents the evaluation model. Section 5.3 demonstrates the obtained results under different criteria by utilizing tables and diagrams. The observations are discussed in Section 5.4. The summary of the chapter discussed in Section 5.5.

5.2 Evaluation Model

In the evaluation phase, the performance of Gradient Boosting Regression Model was compared and contrasted. Ground truth data was required to evaluate the model's accuracy. 40% of data from the whole data set was taken as the ground truth data which was never being in the training set.

The model was evaluated under different criteria. K Fold Cross Validation is used to evaluate a model's performance by handling the variance problem of the result set. Furthermore, in order to identify the best parameters, Grid Search algorithm is used. Hyper parameters that are considered for evaluation are as follows.

- Evaluate with Different estimators
- Evaluate with Different learning rates

- Evaluate with Different max depths
- Evaluate with Different Max Feature values

To determine the accuracy of the model estimates and also for evaluating and validating the estimates, R squared error was used which is a measure of statistics that represents the goodness of fit of a regression model [17]. The ideal value for r-squared error is 1. The closer the value of r-squared error to 1, the better the model fitted. R squared error is calculated by using the following formula :

$$R^2 = 1 - (SS_{res}/SS_{tot}) \tag{5.1}$$

where

$$SS_{res}$$

is the residual sum of squares and

$$SS_{tot}$$

is the total sum of squares. In addition to that, the performances of the model is presented using graphs.

5.3 Results

5.3.1 Pre processing

The missing values that were encoded as np.nan were replaced by making use of the mean value of the columns which include the missing values (Figure 5.1).

5.3.2 Feature Extraction

The feature subset selection was carried out by using Weka application's wrapper subset evaluator (Figure 5.2) and classifier subset evaluator (Figure 5.3). In wrapper subset evaluation, it wraps a classifier in a cross-validation loop and searches through the feature space and uses the classifier to find a good feature set. Best first search method was employed here. In classifier subset evaluation, ranker search

```
n [7]: dataset = imp_mean.fit_transform(dataset)
       print (dataset)

[[8.3260000e+03 2.0150511e+07 3.2160000e+04 ... 3.5150000e+03
  7.3260000e+03 1.2000000e+01]
 [7.7180000e+03 2.0150511e+07 3.2100000e+04 ... 5.1530000e+03
  6.1550000e+03 0.0000000e+00]
 [7.7180000e+03 2.0150511e+07 3.2280000e+04 ... 3.6280000e+03
  6.3580000e+03 1.1800000e+02]
 ...
 [8.2310000e+03 2.0150513e+07 4.4460000e+04 ... 1.3160000e+03
  6.8040000e+03 2.0000000e+00]
 [7.8830000e+03 2.0150513e+07 4.4520000e+04 ... 3.3270000e+03
  6.3840000e+03 1.0000000e+00]
 [7.6970000e+03 2.0150413e+07 4.6140000e+04 ... 5.7340000e+03
  7.7130000e+03 1.8000000e+01]]
```

Figure 5.1: Imputation Results

method is used. It evaluates each feature and lists the results according to the order of the rank.

The correlation heat map (Figure 5.4) depicts the correlation of independent variables with the target variable (waiting time)

5.3.3 Evaluate with K Fold Cross Validation for Different K values

K fold cross validation is a process used to evaluate machine learning models. When K is defined, the training data set is split into K samples as one sample for validation and others for training. This process iterates K times by changing the validation sample in each iteration. Because of the limited data set, K-fold cross-validation is used to train the model over newcomers. Here we define K = 2; 3; :::; 10 to evaluate the model against different K values. The obtained r squared errors are shown in Table 5.2. Figures 5.5, 5.6,5.7, 5.8, 5.9 and 5.10 depicts Learning curves of training and cross validation score when increasing the K value.The parameters of Gradient Boosting Regression Model remain constant, where

- number of estimators equal to 100
- maximum depth equals to 4
- learning rate equals to 0.1

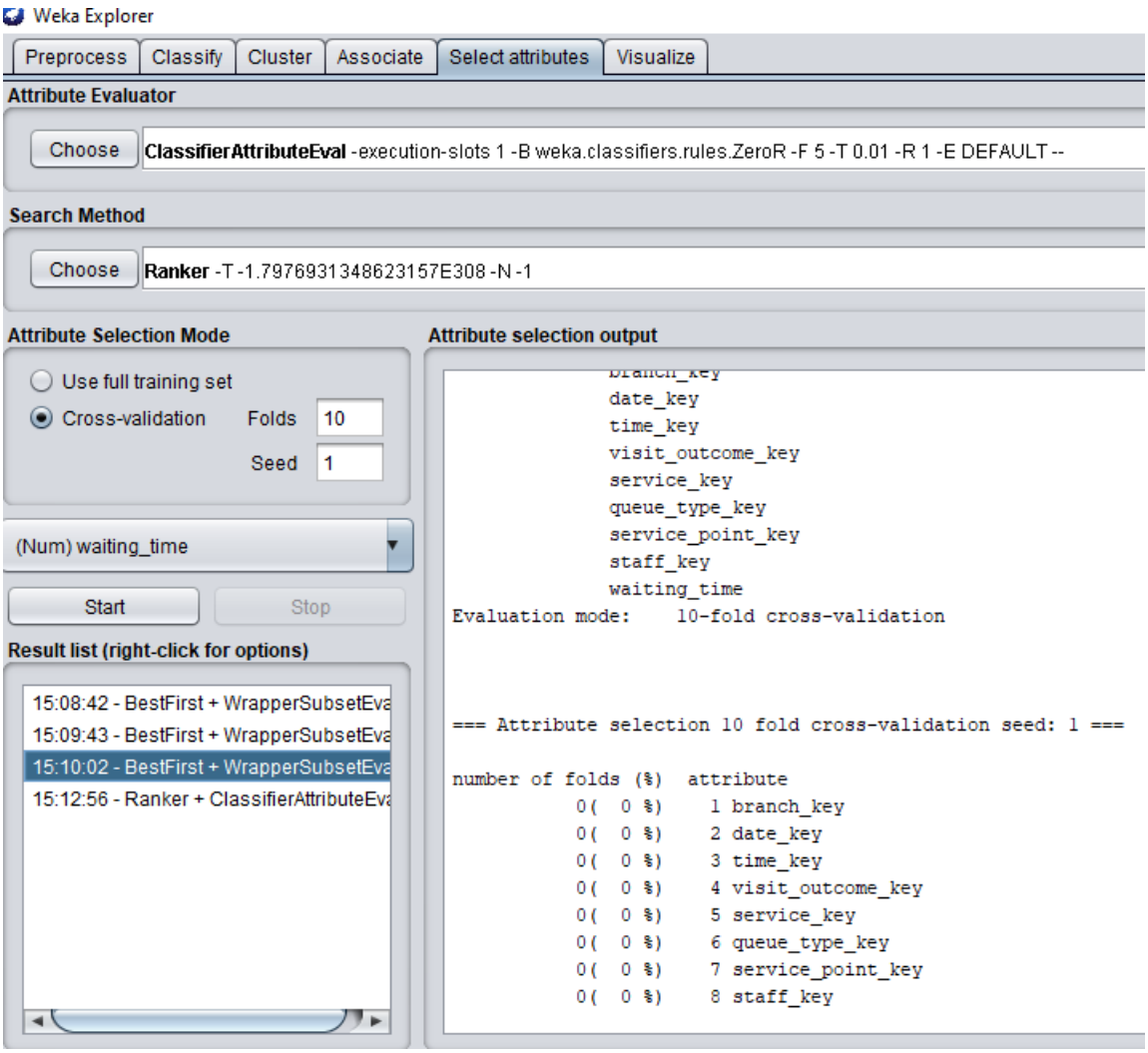

Figure 5.2: Results of wrapper subset evaluator

- minimum samples leaf equals to 2
- maximum features equals to 0.3
- loss function equals to least squares regression (ls)

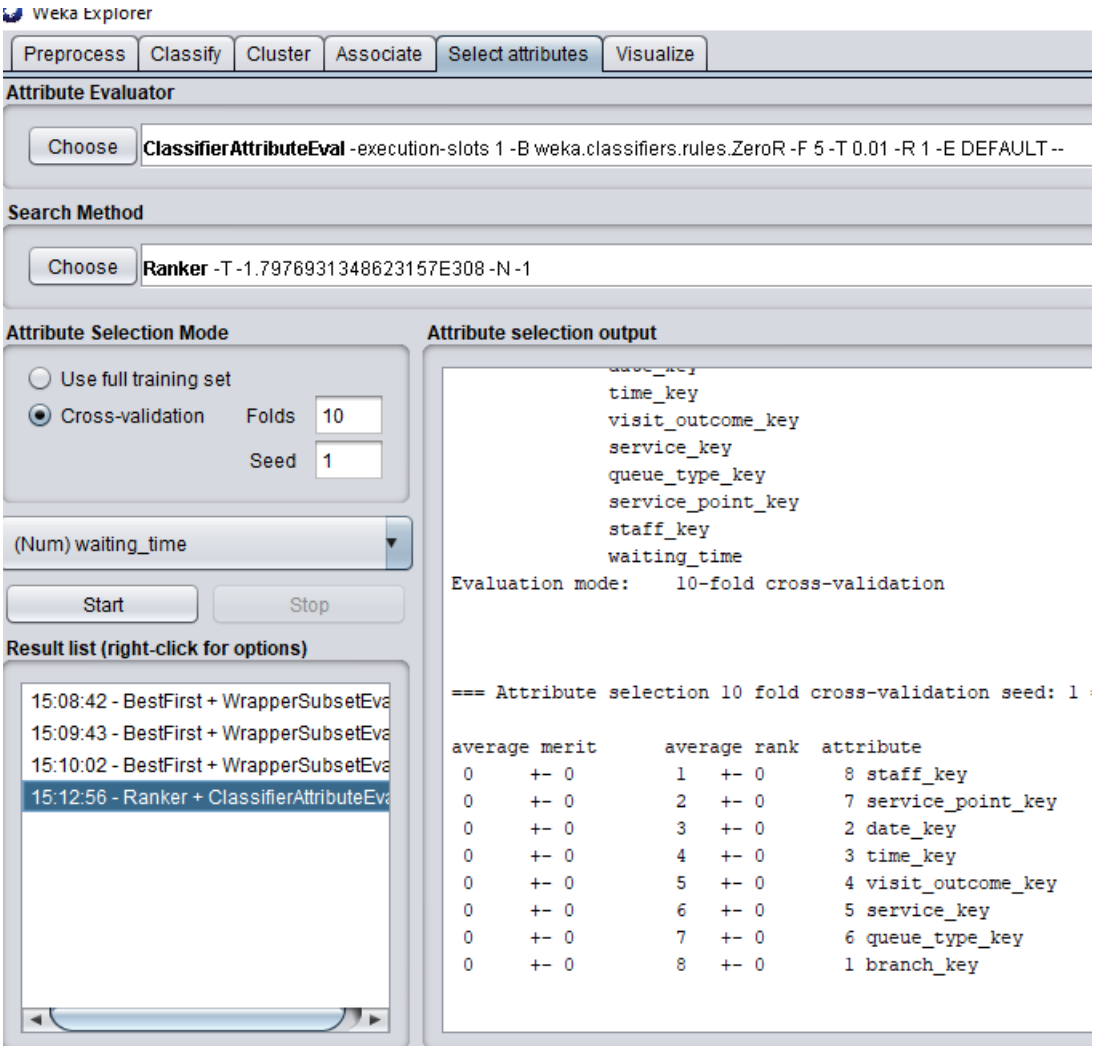

Figure 5.3: Results of classifier subset evaluator

Table 5.1: Test Variance using R-squared error for different K values

#	K Value	Test Variance
1	2	67%
2	3	67.5%
3	4	69%
4	5	69.1%
5	6	69.23%
6	10	71%

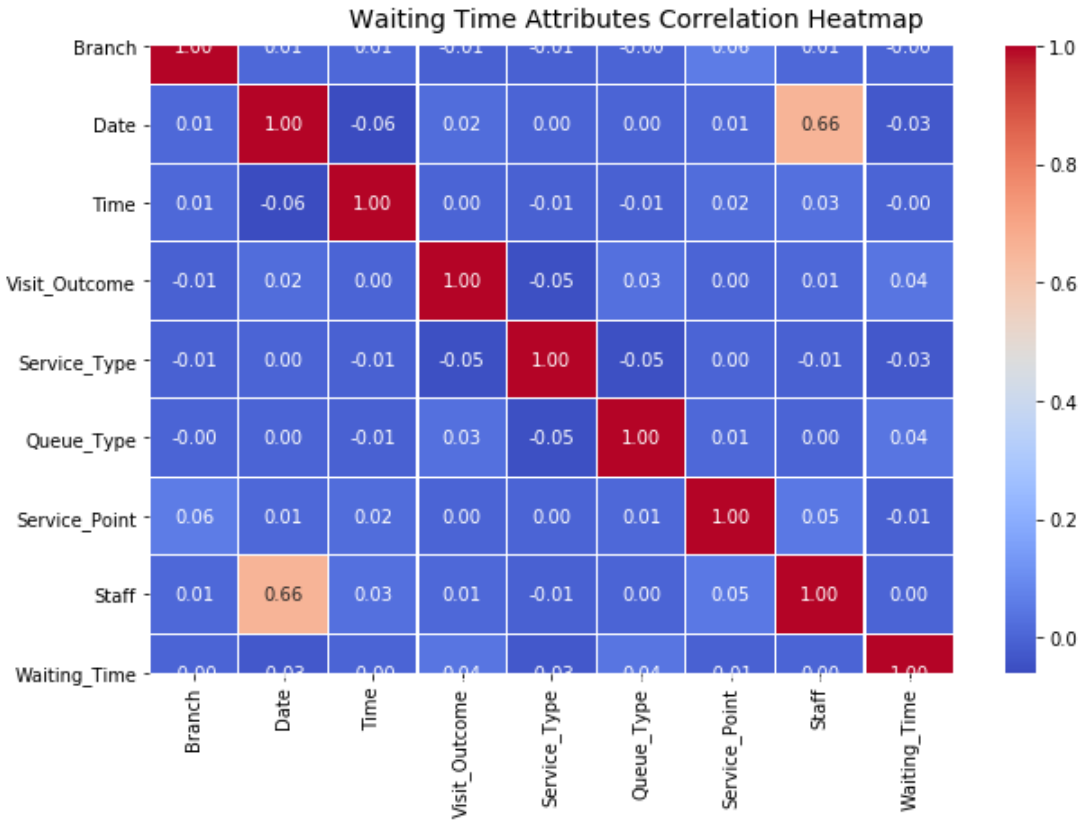

Figure 5.4: Correlation Heat Map

5.3.4 Evaluate with Different n estimators

n estimator parameter is the number of stages to be performed. Because of over-fitting in GBM, a large number can be used for better performance. Figure 5.11 depicts how the two learning curves behave when n estimator equals to 50.

5.3.5 Evaluate with Different learning rates

The contribution of each tree is shrunk by learning rate. In between learning rate and n estimators, there is a trade off.

5.3.6 Evaluate with Different Max Depth

Max depth refers to the maximum depth of a tree. It can be used to control over-fitting when higher depth values are given. It allows model to learn relations very specific to a particular observation. The comparison of different max depth values are depicted in the Figure 5.15

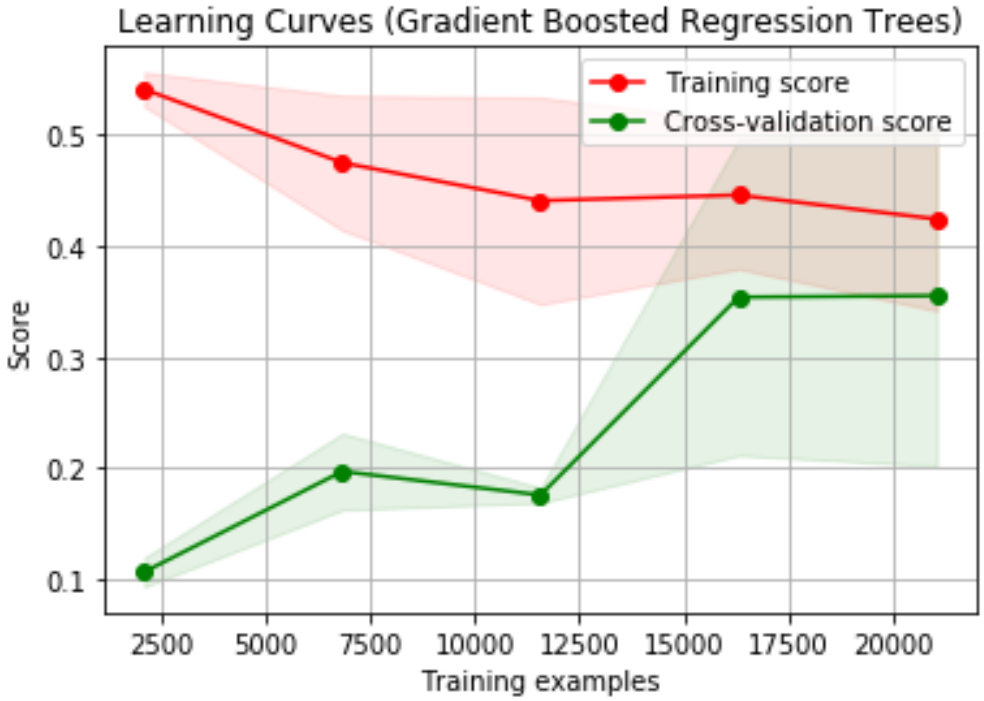

Figure 5.5: Learning curves when K=2

5.3.7 Evaluate with Different Min Sample Leaves

The minimum number of observations required in a terminal node or in a leaf is referred to the Min Sample Leaves.This hyper parameter should be tuned to control over fitting. The comparison of different min sample leaves values are depicted in the Figure 5.16

5.3.8 Evaluate with Different Max Feature Values

The hyper parameter 'max feature' is the number of features to consider while searching for a best split. Max feature is selected randomly. The square root of the total number of features works great [12] but check up to 30-40% of the total number of features should be checked. The comparison of different max feature values are depicted in the Figure 5.17

5.4 Discussion

The gradient boosting regression model was evaluated under different conditions. First, optimal feature subset was obtained by using both wrapper based method

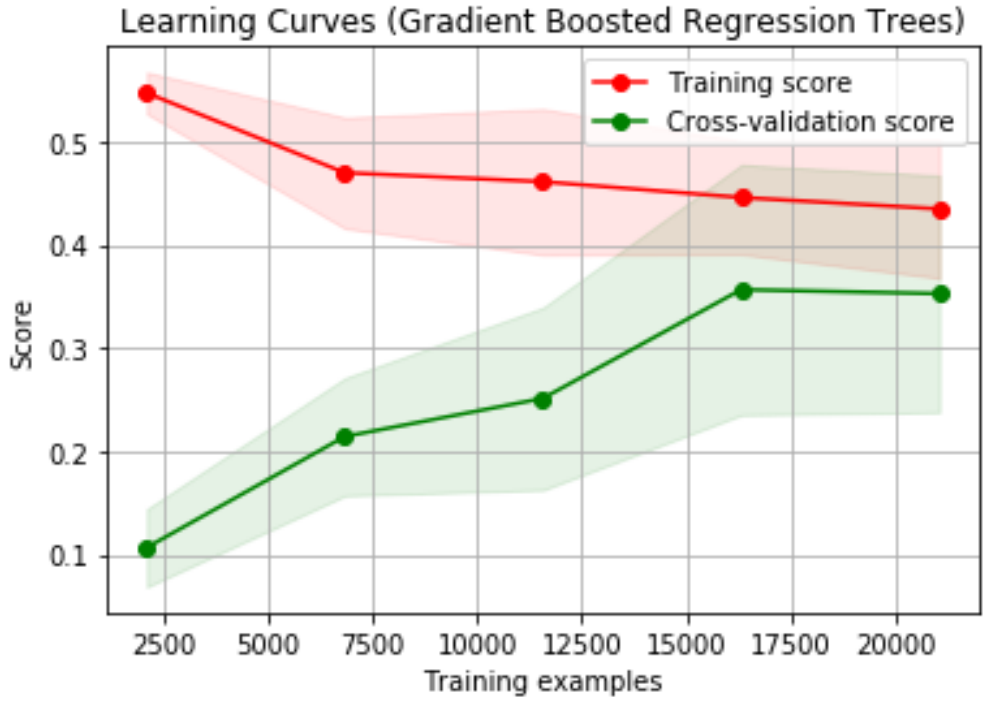

Figure 5.6: Learning curves when K=3

and filter method. According to the results obtained from Weka application, features including branch, date, time, visit outcome,service type,queue type,service point and staff member is the subset of features that is highly correlated with the class while having low inter correlation. Additionally, the correlation heat map depicts the same phenomena.

When considering the K fold cross validation, Figure 5.5, Figure 5.6, Figure 5.7, Figure 5.8, Figure 5.9 and Figure 5.10 infer that over fitting can be addressed better with the increment of K values. In addition to that, Table 5.1 shows the highest r squared score when K equals to 10

According to the Figure 5.16, figure depicting 2 sample leaves gives the best performance. Due to the fact that some features are imbalanced, lower values perform more desirable. It is due to the regions, in which the minority class will be in majority will be small.

As mentioned in the section 5.3.8, when max depth equals to 4, the gap between train and cv curves are minimum. When it is the case of value 3, the curves are under fitted.

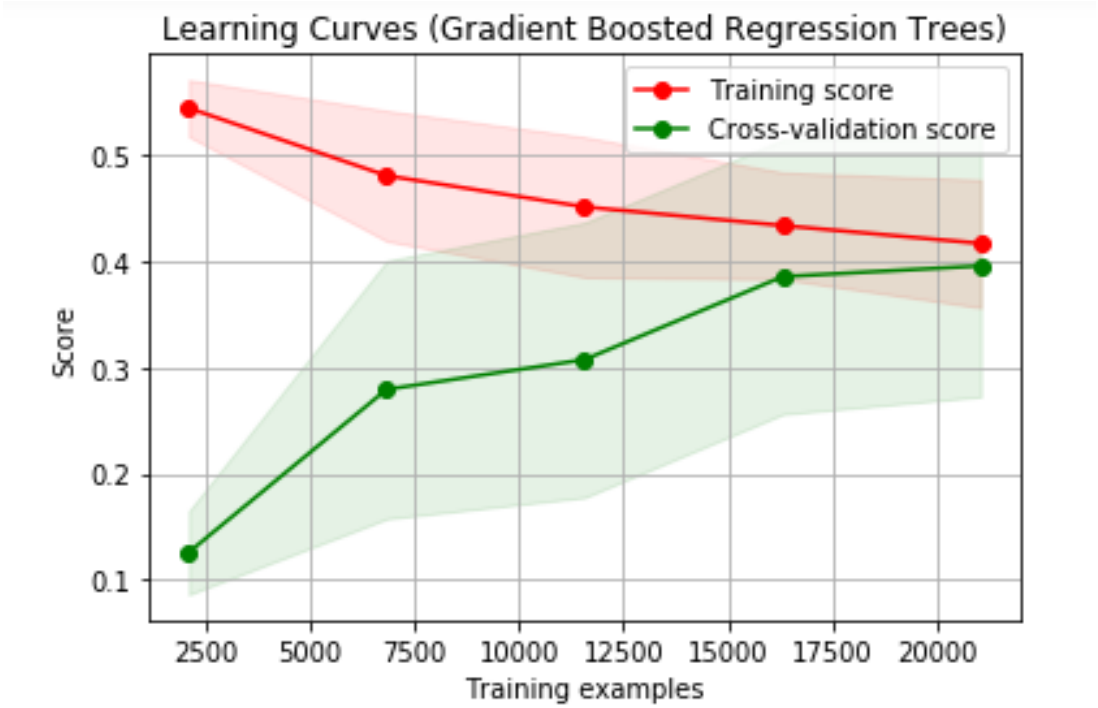

Figure 5.7: Learning curves when K=4

The figure 5.1 depicts the train set and test set deviance curves when the number of estimators (boosting iterations) increase. When comparing the figure 5.11 and the figure 5.13, a large number usually results in better performance. It is due to over-fitting.

5.5 Summary

This chapter reviews the evaluation results and findings with regard to gradient Boosting regression model for waiting time prediction in Qmatic bank virtual queues. Experiments have been carried out to validate the feature set, K value for cross-validation and to tune the hyper parameters such as learning rate, max depth,min sample leaves and max feature.

According to the results,best results obtained when learning rate equals to 0.1, max depth equals to 4, min sample leaves equals to 2 and max feature equals to 0.3. For the training processes, 10-Fold cross-validation is applied. The overall accuracy of the model is 71%.

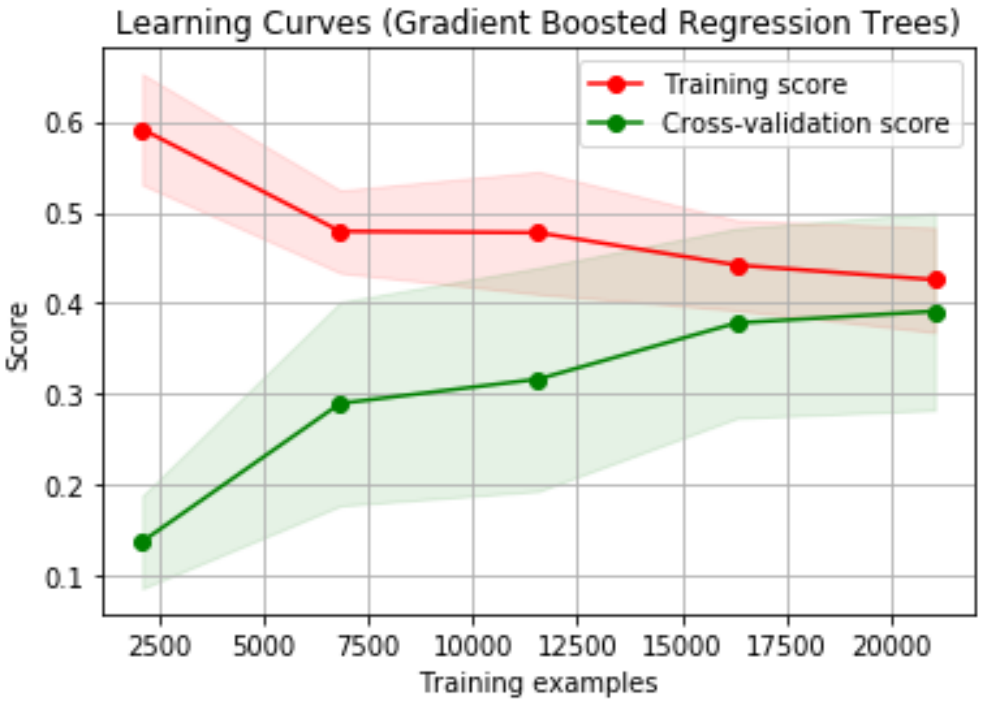

Figure 5.8: Learning curves when K=5

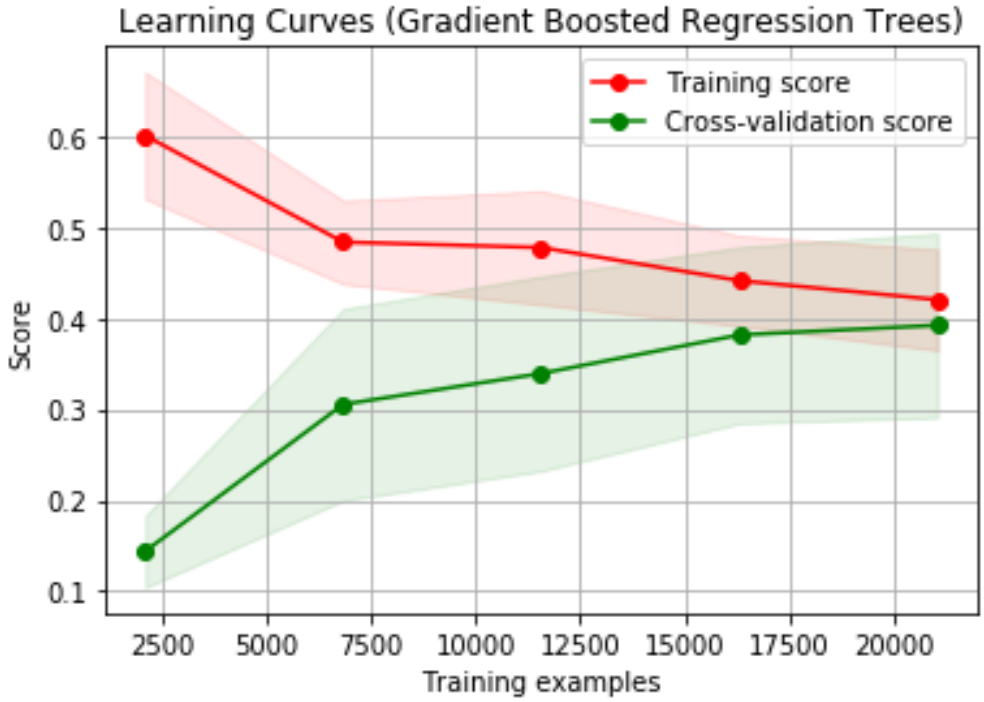

Figure 5.9: Learning curves when K=6

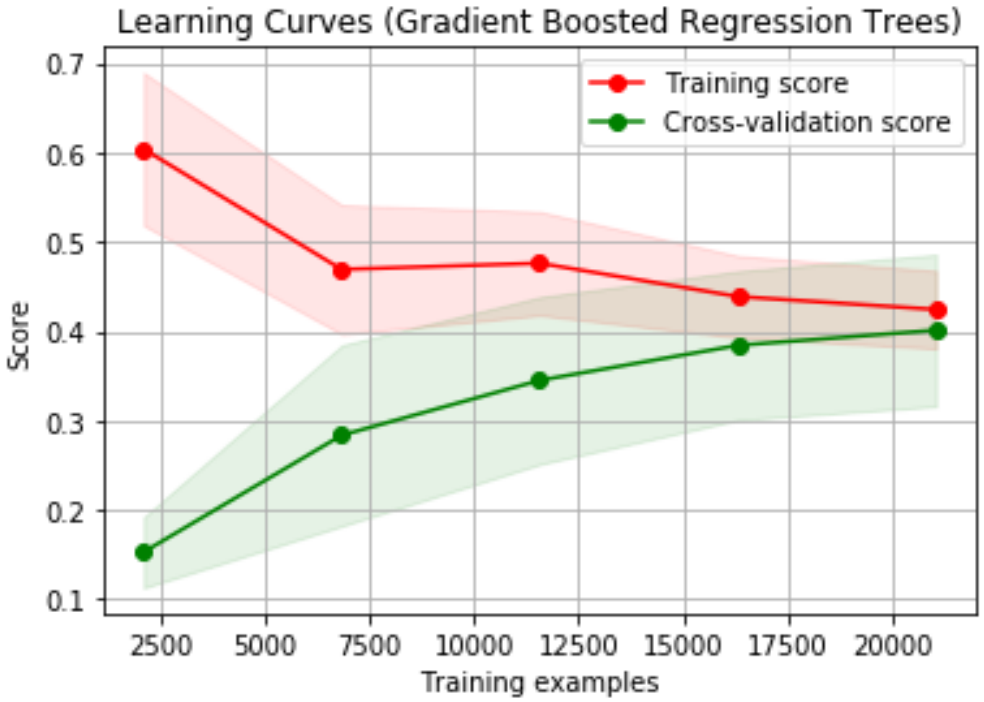

Figure 5.10: Learning curves when K=10

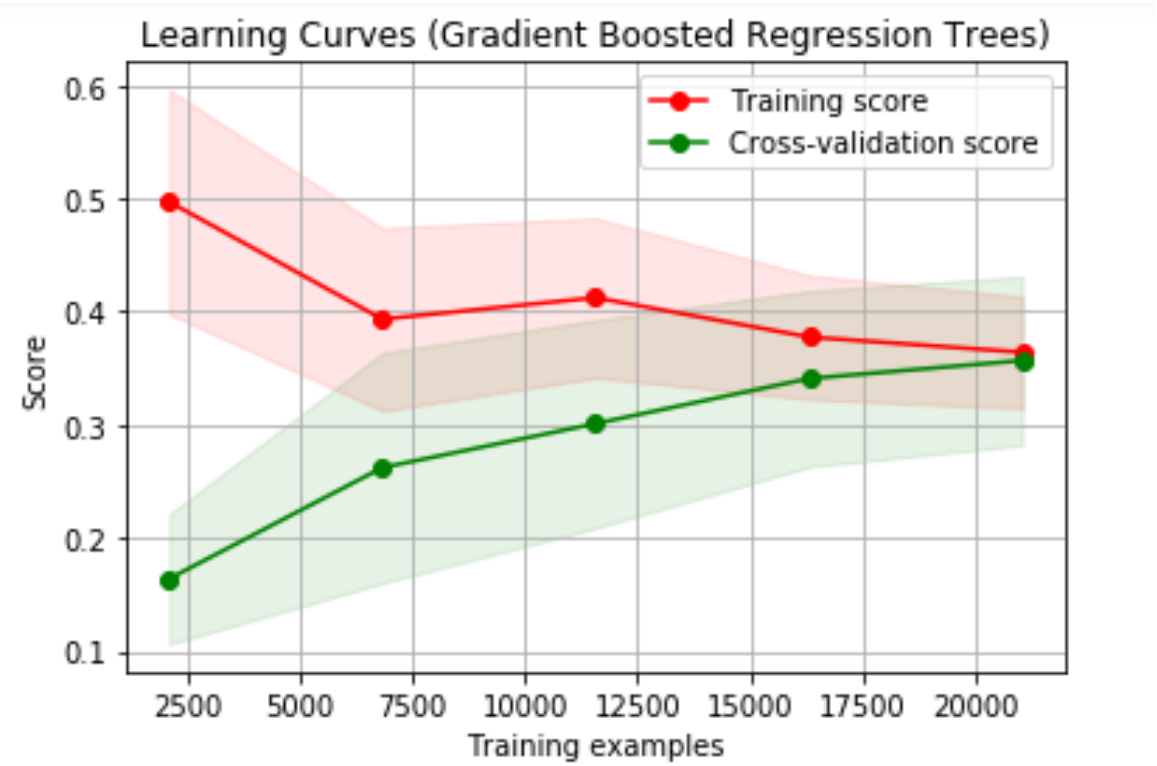

Figure 5.11: n estimators=50

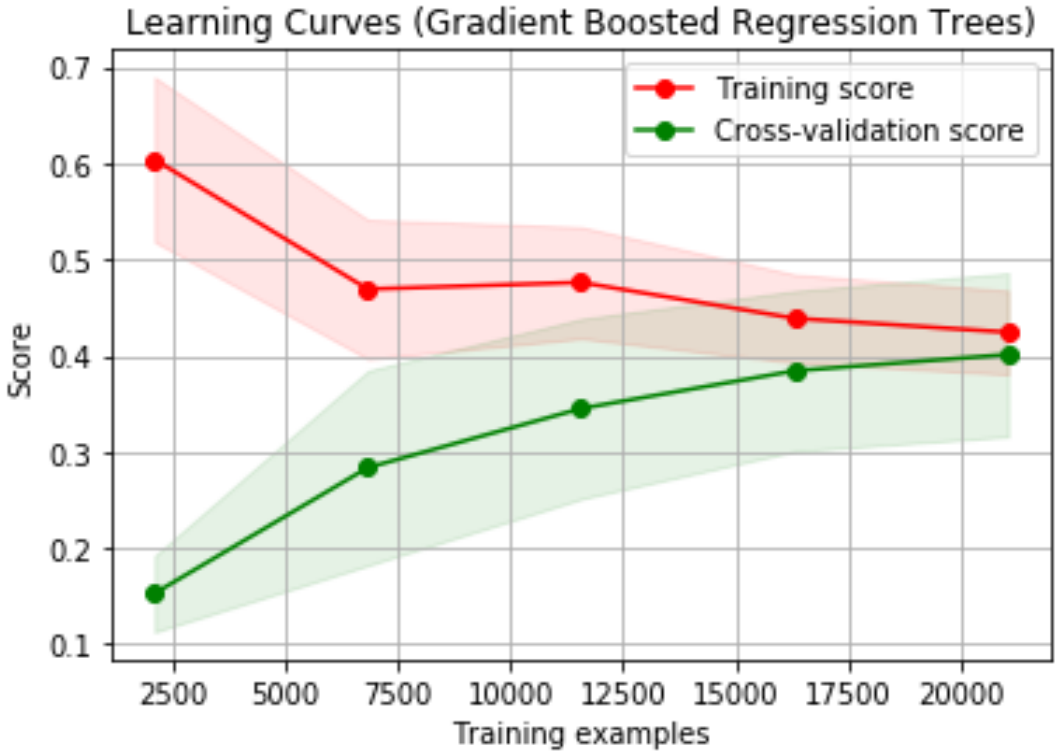

Figure 5.12: n estimators=100

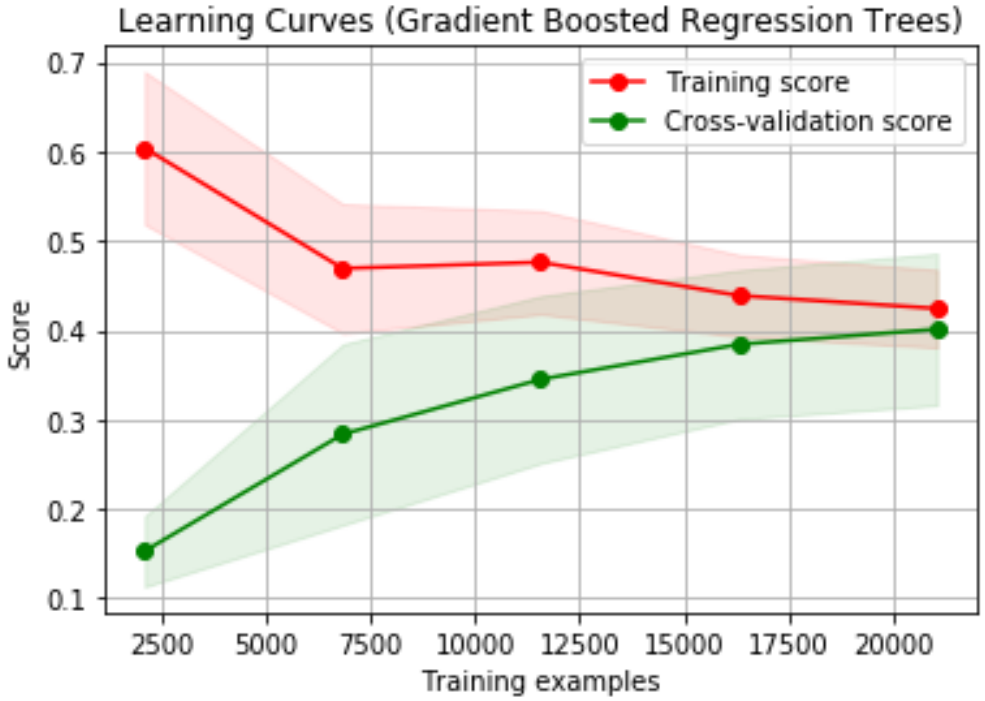

Figure 5.13: Learning rate = 0.1

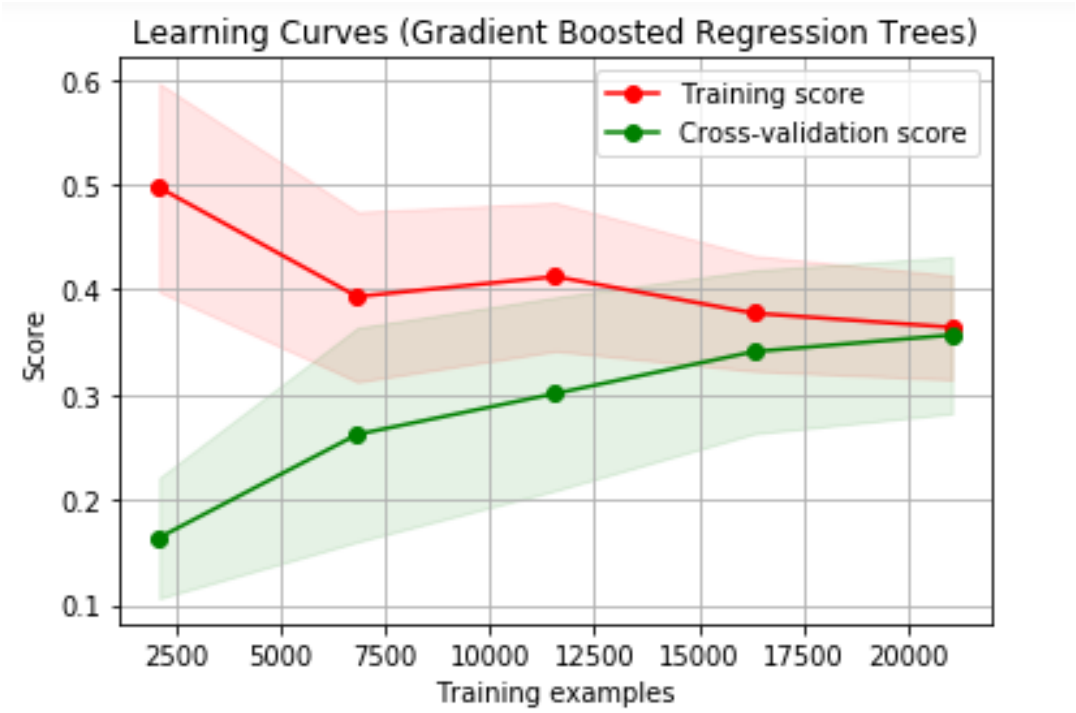

Figure 5.14: Learning rate = 0.05

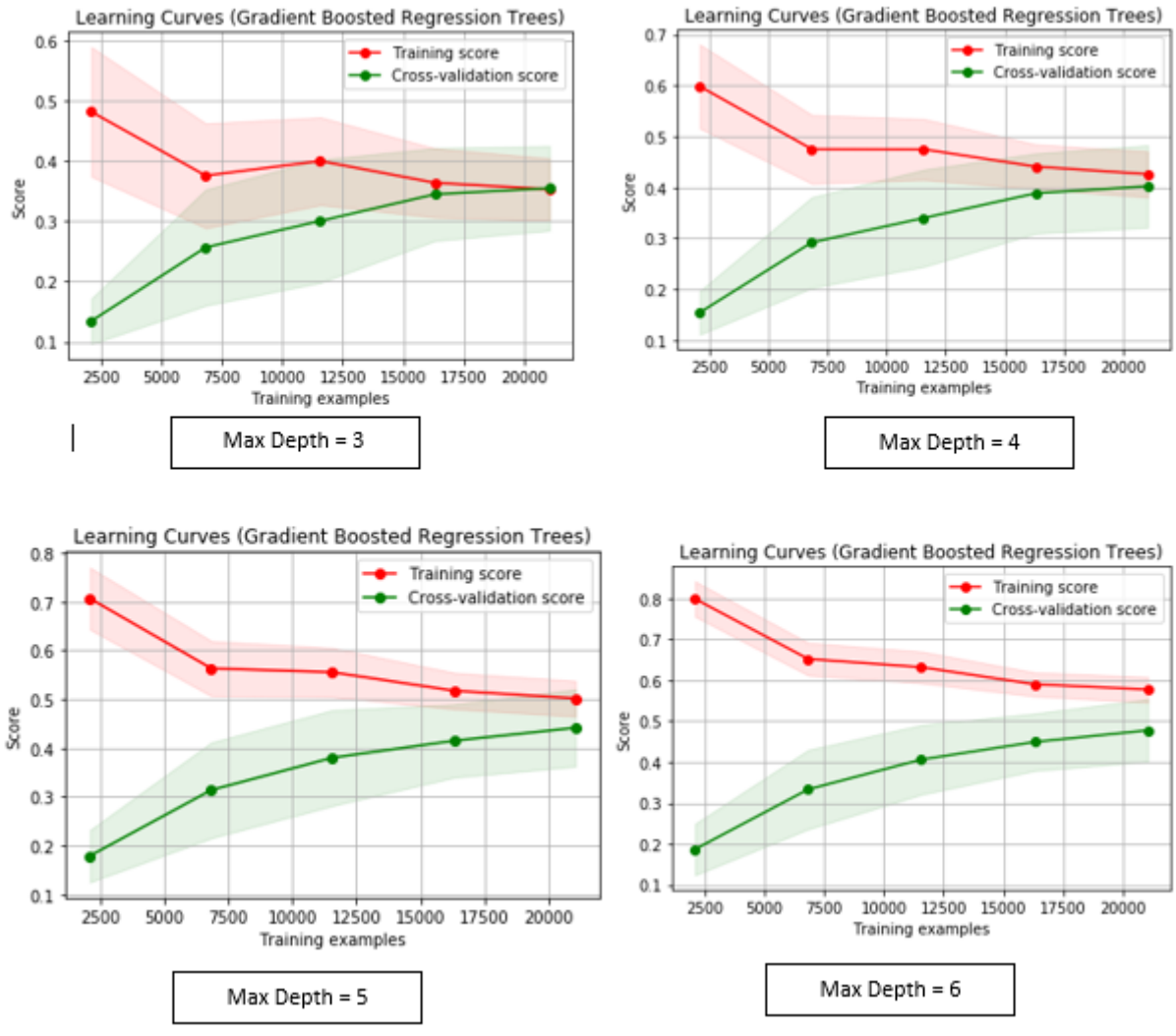

Figure 5.15: Max Depth Comparison of Learning Curves

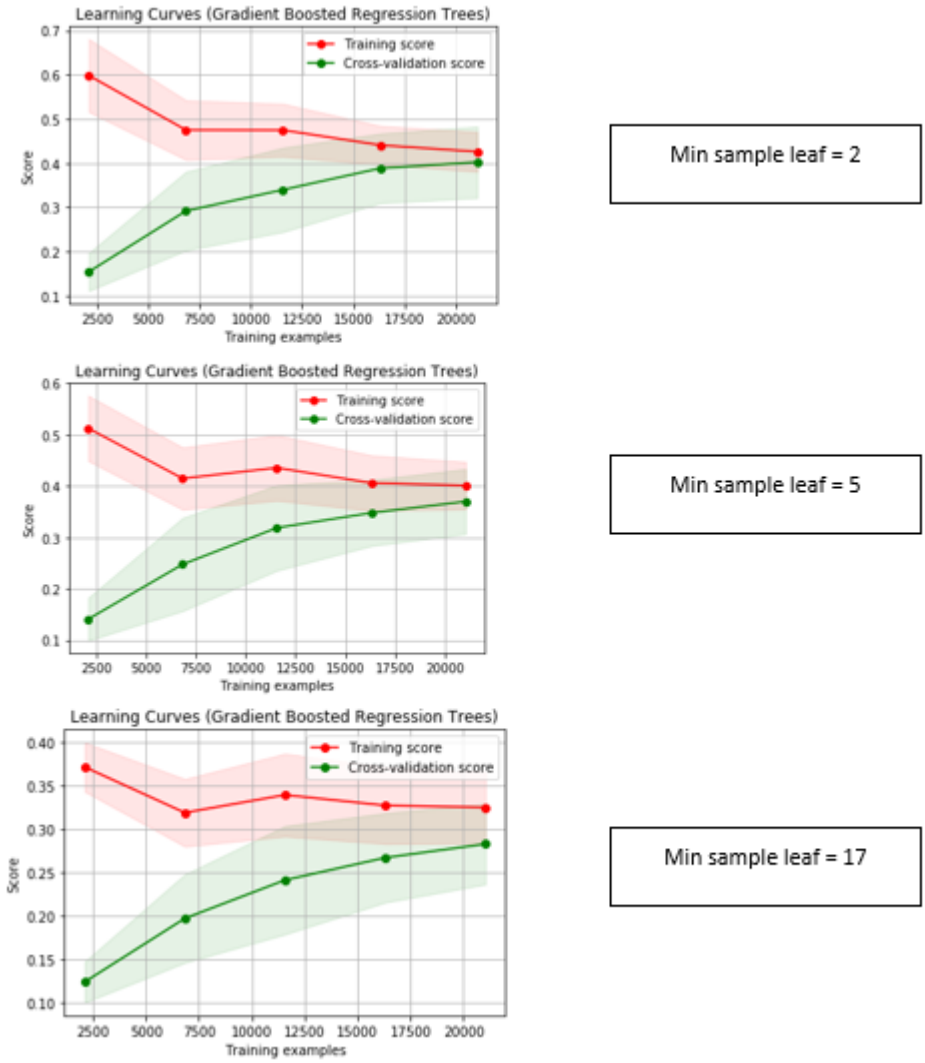

Figure 5.16: Min Sample Leaf Comparison of Learning Curves

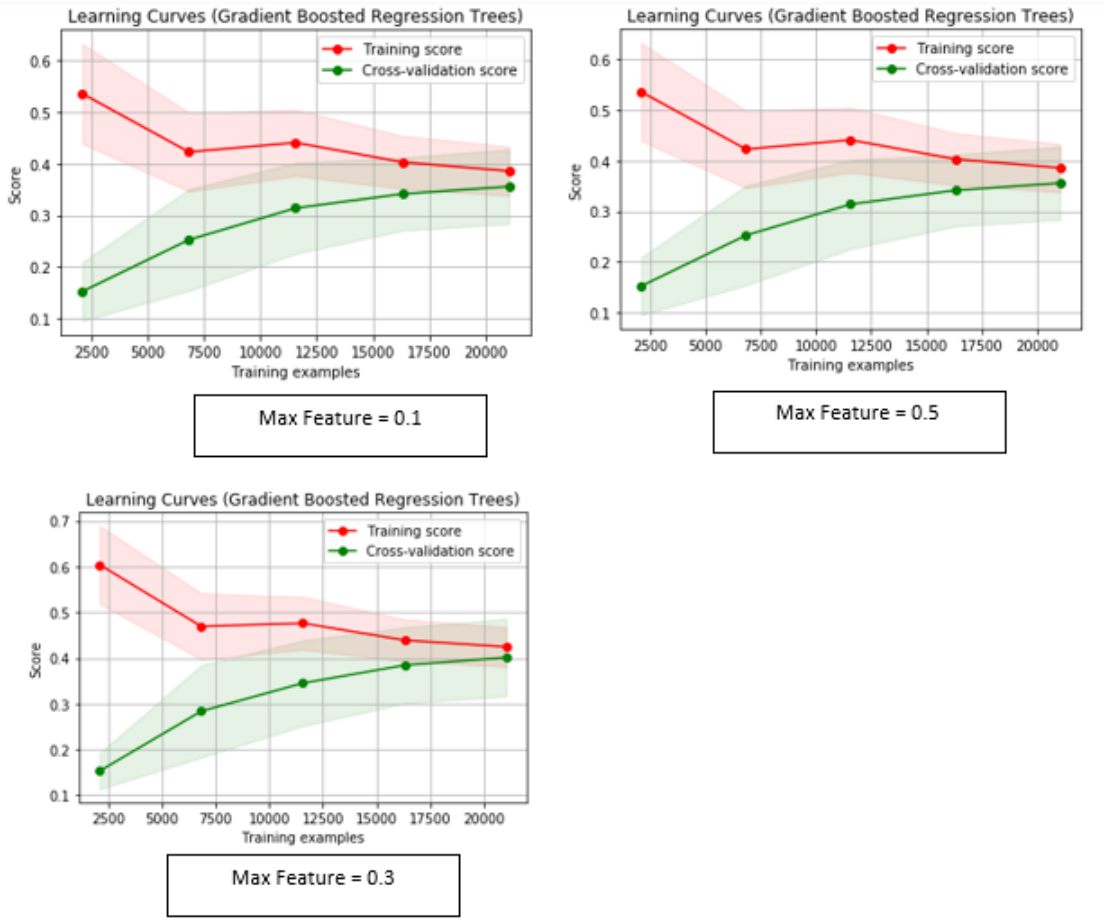

Figure 5.17: Comparison of Max Feature of Learning Curves

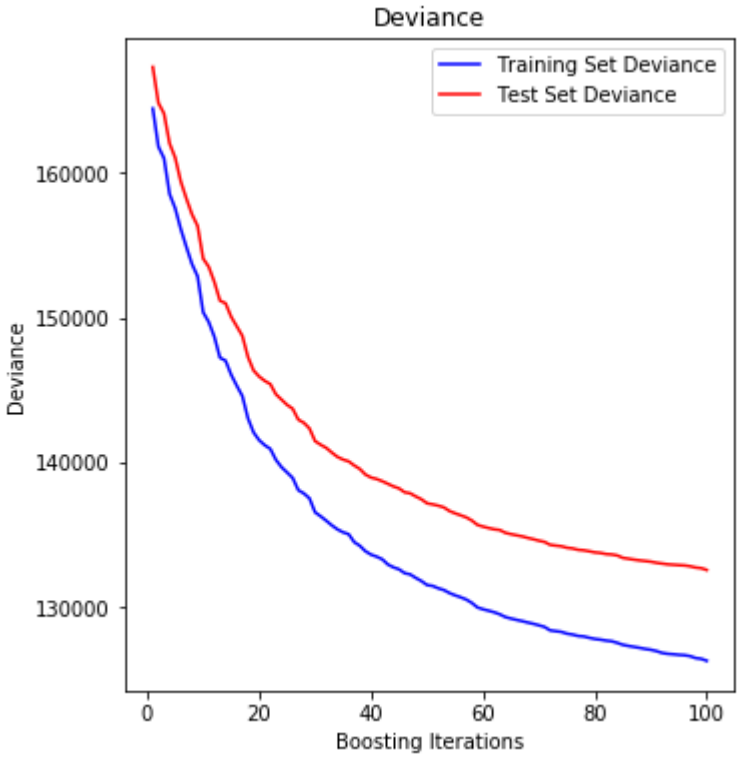

Figure 5.18: Train and Test set Deviance

Chapter 6

Conclusion

6.1 Introduction

This chapter reviews the conclusions formed after completion of the research. Section 6.2 concludes the research questions stated in Section 1.2 followed by the aim and objectives. In Section 6.3, the contribution of the dissertation for waiting time prediction and minimization in Qmatic queue management system is presented. The subsequent section will review the limitations of this research apparent during the progress of the research as well as the implications for further research.

6.2 Conclusion about the research questions

The goal of this research is to contribute to the prediction and mitigation of waiting time in Qmatic virtual customer queues by building a model for prediction and proposing an approach for mitigation in order to have an effective usage of resources such as staff members and service counters which will result in providing a satisfactory service to Qmatic customers. In order to achieve the ultimate goal, the following key objectives were set.

- Explore waiting time patterns in Qmatic queues
- Identify a suitable model for waiting time pattern recognizing and prediction.
- Study how to minimize the waiting time with pattern observed.

- Propose a machine learning engine that can predict the waiting time and an approach to mitigate it to improve Qmatic services.

Because each step of the customer transaction information was time stamped and stored in the system, customers' waiting time duration patterns were observed. After achieving the first objective, employing an appropriate learning model was the next objective. The efficiency and the effectiveness also were taken into account when selecting a learning model. The literature says data mining algorithm based approaches certainly benefited for most of the waiting time prediction scenarios [12]. Queueing theory, neural networks, deep learning, RF, SVM, and GBM are approaches that were found from related works. Therefore, when selecting an appropriate learning method the advantages, disadvantages, applicability and feasibility of each approach were observed. According to the observation, as the data mining engine a GBM model is used for waiting time estimation. One of the most substantial measures in creating the regression model is selecting the optimal feature sets for predicting waiting time. According to the results, the best results obtained when learning rate equals to 0.1, max depth equals to 4, min sample leaves equals to 2 and max feature equals to 0.3. For the training processes, 10-Fold cross-validation is applied. The overall accuracy of the model is 71%

6.3 Conclusion about research problem

The ultimate aim of any organization is to provide better service to their customers and to obtain higher profit while maintaining their reputation for sustainability. To maintain the above factors in a balanced manner by Qmatic as an organization, waiting time for virtual queues should be minimized to compete with the competitors. In the present context, several studies on estimation of waiting time in virtual queues has been conducted. However, in Qmatic queue management system, there is a lack of computer science research for making predictions and business decision making. It has been my personal motivation of interest to explore the hidden insights in Qmatic customer journeys and develop a prediction mechanism to estimate average waiting time and an approach to mitigate it because it is a service to the society while getting intellectual joy by facing the challenge in solving the

unsolved problem.

6.4 Limitations and Implications for further research

The major limitation of this research is that the model is trained and tested only for data set provided by Qmatic Queue Management System.The data is based on only for the recorded waiting time of bank services. However, this study creates a platform to further generalize this model to all Qmatic services. When analyzing the features that should be fed to the Gradient Boosting Regression model, it was hard because the attributes of database schema's tables are not well described. Altogether, 8 features were identified which contribute to waiting time duration. Therefore, identification of more distinguishable features can be lead the model to provide more accurate results. In addition to that, although it is said that Qmatic queue management system is having an initial waiting time for a service, data related to initial waiting time for a service was not provided. As an further implication to this research, dynamic programming approach can be employed once the above requirement is fulfilled where the better results can be obtained.

References

[1] R. C. Anirudha, R. Kannan, and N. Patil. Genetic algorithm based wrapper feature selection on hybrid prediction model for analysis of high dimensional data. In *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6, Dec 2014.

[2] Anon. Research methodology: methods and techniques,. 2019.

[3] Andre Carvalho and Orlando Belo. Predicting waiting time in customer queuing systems. pages 155–159, 09 2016.

[4] J. Cheng, G. Li, and X. Chen. Research on travel time prediction model of freeway based on gradient boosting decision tree. *IEEE Access*, 7:7466–7480, 2019.

[5] Haris Gaanin and Mark Wagner. Artificial intelligence paradigm for customer experience management in next-generation networks: Challenges and perspectives. *IEEE Network*, 33:188–194, 2018.

[6] Jean Golay, Michael Leuenberger, and Mikhail Kanevski. Feature selection for regression problems based on the morisita estimator of intrinsic dimension. *Pattern Recognition*, 70:126–138, 10 2017.

[7] Qiuming Guo, W Wu, D.L Massart, C Boucon, and S Jong. Feature selection in principal component analysis of analytical data. *Chemometrics and Intelligent Laboratory Systems*, 61:123–132, 02 2002.

[8] Rinda Parama Satya Hermanto, Suharjito, Diana, and Ariadi Nugroho. Waiting-time estimation in bank customer queues using rprop neural networks.

Procedia Computer Science, 135:35 – 42, 2018. The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life.

[9] Rouba Ibrahim and Ward Whitt. Wait-time predictors for customer service systems with time-varying demand and capacity. *Oper. Res.*, 59(5):1106–1118, September 2011.

[10] Qamar Iqbal, Lawrence Whitman, and Don Malzahn. Reducing customer wait time at a fast food restaurant on campus. *Journal of Foodservice Business Research*, 15:319–334, 10 2012.

[11] Hillier F Lieberman G. Introduction to operations research tenth edition. 2015.

[12] R. N. Mourõo, R. S. Carvalho, R. N. Carvalho, and G. N. Ramos. Predicting waiting time overflow on bank teller queues. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 842–847, Dec 2017.

[13] Sri Sundari Dwi Pudjaningsih Nanda Surya Febrianta. Waiting time analysis of pharmaceutical services with queue method in pku muhammadiyah hospital bantul. *International Journal of Scientific and Research Publications*, 9:54–58, 2017.

[14] Fredrik Norrman and Josefin Stintzing. Prediction of queuing behaviour through the use of artificial intelligence, 2017.

[15] G.R. Rodríguez-Jáuregui, A. González-Pérez, Salvador Hernandez, and M.D. Hernández-Ripalda. Análisis del servicio del área de urgencias aplicando teoría de líneas de espera. *Revista Contaduria y Administración*, 1, 03 2016.

[16] Saharon Rosset, Claudia Perlich, and Bianca Zadrozny. Ranking-based evaluation of regression models. *Knowl. Inf. Syst.*, 12:331–353, 08 2007.

[17] Parasana SankaraRao and Kiran Reddi. Performance evaluation of regression techniques for effort estimation. *International Journal of Computer Applications*, 52:8–12, 08 2012.

[18] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. Queue mining for delay prediction in multi-class service processes. *Information Systems*, 53, 04 2015.

[19] Jeffrey S. Smith and Barry L. Nelson. Estimating and interpreting the waiting time for customers arriving to a non-stationary queueing system. In *Proceedings of the 2015 Winter Simulation Conference*, WSC '15, page 2610–2621. IEEE Press, 2015.

[20] Yanru Zhang and Ali Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C Emerging Technologies*, 58, 03 2015.

Appendices

Appendix A

Code Listings

Listing 1.PNG Listing 1.PNG

```
#import packages
import pandas as pd
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats

#loading data
dataset = pd.read_excel (r'featureListFinal.xlsx')

#Preprocessing

#import packages
from sklearn import preprocessing
import numpy as np

#imputation
# dataset has 'NULL' and '?' in it, convert these into NaN
dataset = dataset.replace('NULL', np.nan)
dataset = dataset.replace('?', np.nan)


#Replace missing values, encoded as np.nan, using the mean value
of the columns (axis 0) that contain the missing values:

imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
dataset = imp_mean.fit_transform(dataset)
print (dataset)

#outlier removal

from sklearn.ensemble import IsolationForest
clf = IsolationForest(n_estimators=100, warm_start=True)
clf.fit(dataset)


#normalizing

normalizer = preprocessing.Normalizer().fit(dataset)

# Creating pandas dataframe from numpy array

dataset = pd.DataFrame({'Branch': dataset[:, 0],
                        'Date': dataset[:, 1],
                        'Time': dataset[:, 2],
                        'Visit_Outcome': dataset[:, 3],
                        'Service_Type': dataset[:, 4],
                        'Queue_Type': dataset[:, 5],
                        'Service_Point': dataset[:, 6],
                        'Staff': dataset[:, 7],
                        'Waiting_Time': dataset[:, 8]})

#splitting the dataset into inputs and output

#until the last column
sourcevars = dataset.iloc[:,:-1]

#last column
targetvar = dataset.iloc[:,-1]
```

Figure A.1: Code Listing 1

Listing 2.PNG Listing 2.PNG

```
#Feature extraction
#Correlation of features

f, ax = plt.subplots(figsize=(10, 6))
corr = dataset.corr()
hm = sns.heatmap(round(corr,2),
                 annot=True, ax=ax,
                 cmap="coolwarm",fmt='.2f',
                 linewidths=.05)
f.subplots_adjust(top=0.93)
t= f.suptitle('Waiting Time Attributes Correlation Heatmap',
              fontsize=14)

#train/test set splitting

X_train, X_test, y_train, y_test = train_test_split(sourcevars,
                 targetvar,
                 test_size=0.4,
                 random_state=42)
print (X_train.shape, X_test.shape)
print(X_train)


#import packages

from sklearn import ensemble
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_squared_error, r2_score

def GradientBooster(param_grid, n_jobs):

    estimator = GradientBoostingRegressor()
    cv = ShuffleSplit(X_train.shape[0], test_size=0.3)

    classifier = GridSearchCV(estimator=estimator,
                              cv=cv,
                              param_grid=param_grid,
                              n_jobs=n_jobs)
    classifier.fit(X_train, y_train)

    # best estimator that was found by GridSearchCV
    print ("Best Estimator learned through GridSearch")
    print (classifier.best_estimator_)

    return (cv, classifier.best_estimator_)
```

Figure A.2: Code Listing 2

Listing 3.PNG Listing 3.PNG

```
#Traning the GBM

param_grid={'n_estimators':[100],
            'learning_rate': [0.1],# 0.05, 0.02, 0.01],
            'max_depth':[4],#4,6],
            'min_samples_leaf':[2],#,5,9,17],
            'max_features':[0.3],#,0.3]#,0.1],
            }
n_jobs=2

cv,best_est=GradientBooster(param_grid, n_jobs)

print ("Best Estimator Parameters" )
print("---------------------------" )
print ("n_estimators: %d" %best_est.n_estimators)
print ("max_depth: %d" %best_est.max_depth )
print("Learning Rate: %.1f" %best_est.learning_rate)
print ("min_samples_leaf: %d" %best_est.min_samples_leaf)
print ("max_features: %.1f" %best_est.max_features )
print()
print("Train R-squared: %.2f" %best_est.score(X_train,y_train))

params = {'n_estimators': 100,
          'max_depth': 4,
          'min_samples_leaf': 2,
          'max_features':0.3,
          'learning_rate': 0.1,
          'loss': 'ls'}
          #loss function ls refers to least squares regression

model = ensemble.GradientBoostingRegressor(**params)
model.fit(X_train, y_train)


#Testing the GBM

model_score = model.score(X_train,y_train)

# Explained variance score: 1 is perfect prediction
print('R2 sq: ',model_score)
y_predicted = model.predict(X_test)
print("predicted waiting time")
print(y_predicted)

# The mean squared error
print("Mean squared error: %.2f"% mean_squared_error(y_test,
y_predicted))
print('Test Variance score: %.2f' % r2_score(y_test, y_predicted))
```

Figure A.3: Code Listing 3

Listing 4.PNG Listing 4.PNG

```
fig, ax = plt.subplots()

ax.scatter(y_test, y_predicted, edgecolors=(0, 0, 0))
ax.plot([y_test.min(),
        y_test.max()],
        [y_test.min(),
        y_test.max()],
        lw=1)

ax.set_xlabel('Actual Waiting Time')
ax.set_ylabel('Predicted Waiting Time')
ax.set_title("Ground Truth vs Predicted")
plt.show()

#Evaluation

def plot_learning_curve(estimator,
                        title, X, y,
                        ylim=None, cv=None,
                        n_jobs=1,
                        train_sizes=np.linspace(.1, 1.0, 5)):

    plt.figure()
    plt.title(title)

    if ylim is not None:
        plt.ylim(*ylim)

    plt.xlabel("Training examples")
    plt.ylabel("Score")

    train_sizes, train_scores, test_scores
            = learning_curve( estimator,
                              X, y, cv=cv,
                              n_jobs=n_jobs,
                              train_sizes=train_sizes)

    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)

    plt.grid()

    plt.fill_between(train_sizes,
                     train_scores_mean - train_scores_std,
                     train_scores_mean + train_scores_std,
                     alpha=0.1, color="r")

    plt.fill_between(train_sizes,
                     test_scores_mean - test_scores_std,
                     test_scores_mean + test_scores_std,
                     alpha=0.1, color="g")
```

Figure A.4: Code Listing 4

Listing 5.PNG Listing 5.PNG

```
    plt.plot(train_sizes, train_scores_mean,
             'o-', color="r",
             label="Training score")

    plt.plot(train_sizes,
             test_scores_mean,
             'o-', color="g",
             label="Cross-validation score")

    plt.legend(loc="best")
    return plt


# plotting training and cross validation score
title = "Learning Curves (Gradient Boosted Regression Trees)"
estimator = GradientBoostingRegressor(
                n_estimators=best_est.n_estimators,
                max_depth=best_est.max_depth,
                learning_rate=best_est.learning_rate,
                min_samples_leaf=best_est.min_samples_leaf,
                max_features=best_est.max_features)
plot_learning_curve(estimator,
                    title,
                    X_train,
                    y_train,
                    cv=cv,
                    n_jobs=n_jobs)

plt.show()
# compute test set deviance
test_score = np.zeros((params['n_estimators'],), dtype=np.float64)

for i, y_pred in enumerate(model.staged_predict(X_test)):
    test_score[i] = model.loss_(y_test, y_pred)

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.title('Deviance')
plt.plot(np.arange(params['n_estimators']) + 1, model.train_score_, 'b-',
         label='Training Set Deviance')
plt.plot(np.arange(params['n_estimators']) + 1, test_score, 'r-',
         label='Test Set Deviance')
plt.legend(loc='upper right')
plt.xlabel('Boosting Iterations')
plt.ylabel('Deviance')
plt.show()
```

Figure A.5: Code Listing 5