CHAPTER 1

INTRODUCTION

1.1 HEALTHCARE

Healthcare is the act of taking preventative or necessary medical procedures to improve a person's well-being. This may be done with surgery, the administering of medicine, or other alterations in a person's lifestyle. These services are typically offered through a healthcare system made up of hospitals and physicians.

1.1.1 OUTPATIENT CARE

Outpatient services are the most important service provided by the hospitals as it provides services to a large number of patients at affordable cost. Outpatient services include diagnosis, treatment and prevention. Most people can choose an outpatient services instead of inpatient care because outpatient systems move through full continuation of care, or suffer from less severe addiction. Programs for outpatient treatment vary depending upon the patient's needs and the facility but they typically meet a couple of times every week for a few hours at a time. Unlike inpatient treatment, outpatient treatment does not often address medical conditions and nutritional needs. Outpatient services is desired by many people because of its flexibility. For adults with children, who cannot afford to attend treatment for treatment for months at a time or who do not have the insurance to cover their stay, outpatient treatment can be very helpful means of recovery.

Outpatient services include

- **Wellness and prevention**, such as counselling and weight-loss programs.
- **Diagnosis**, such as lab tests and MRI scans.
- **Treatment**, such as some surgeries and chemotheraphy

- **Rehabititation**, such as drug or alcohol rehab and physical theraphy.

1.1.2 INPATIENT CARE

Inpatient care is the care of patients whose condition requires admission to the hospital. Inpatient treatment requires continuous monitoring and it is an expensive treatment. The costs are much higher because there is the need for more staff members and possibly medications. In inpatient treatment constant medical supervision is placed over each resident. In the case of someone with an eating disorder, inpatient treatment through an eating disorder residential program will be more effective in monitoring positive or negative health levels. If the health of a person is declining, the facility can appropriately take care of the person, providing them with care from a local hospital if necessary.

1.1.3 OUTPATIENT WAITING TIME

Outpatient waiting time refers to the time a patient waits in the clinic before being seen by one of the clinical medical staff. Patient waiting time is an important indicator of quality of services offered by hospitals. The amount of time a patient waits to be seen is one factor which affects utilization of healthcare services. Keeping patients waiting unnecessarily can be a cause of stress for both patient and doctor. The institute of medicine recommends that, at least 90% of patients should be seen within 30 minute of their scheduled appointment time.

1.2 DATA MINING

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), a field at the intersection of computer science and statistics, is the process that attempts to discover patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use Aside from

The raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Generally, data mining (sometimes called data or knowledge discovery) is the process of analysing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analysing data. It allows users to analyse data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Data

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. Types of data includes;

- **Categorical data**
- **Numerical data**

Categorical data

Categorical data can take on numerical values (such as”1” indicating male and “2” indicating female), but those numbers do not have mathematical meaning.

Numerical data

Numerical data is data that is measurable, such as time, height, weight, amount, and so on. An example of numerical data would be the number of people that attended the movie theatre over the course of a month.

1.3 MACHINE LEARNING

Machine learning is the science of getting computers to act intelligently without being explicitly programmed. It grew out of the work of Artificial Intelligence. It gives new capability to the computers. Arthur Samuel says that machine learning is a field of study that gives computers the ability to learn without being explicitly programmed. Data Mining can be done with the machine learning algorithms. It is useful in applications that can’t be programmed by hand. The examples are autonomous helicopter, handwriting recognition, language processing (NLP) and Computer Vision. They serve as the self-customizing programs in Amazon or Netflix product recommendations. The future will be governed by machines. They are going to make our lives easier and more comfortable than ever.

Machine learning algorithms can be broadly classified into supervised learning and unsupervised learning methods.

Supervised learning algorithms

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyses the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

CLASSIFICATION

Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on training set and class labels and classifies data based on training set and class labels and it can be used for classifying newly available data. The term could cover any context in which some decision or forecast is made on the basis of presently available information. Classification procedure is recognized method for repeatedly making such decisions in new situations. Classification techniques predict discrete responses-for example, whether an email is genuine or spam, or whether a tumour is cancerous or beginning. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring. Common algorithms for performing classification include support vector machine (SVM), boosted and bagged decision trees, k-nearest neighbour, naïve bayes, discriminant analysis, logistic regression, and neural networks.

REGRESSION

Regression techniques predict continuous responses-for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading. Common regression algorithms include linear model, nonlinear model, regularization, stepwise regression and bagged decision trees, neural networks and adaptive neuro-fuzzy learning.

Common supervised learning algorithms include:

Decision Trees A decision tree is a decision support tool that uses a treelike graph or model of decisions and their possible consequences, including chance-event outcomes, resource costs, and utility.

Naive Bayes Classification Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Random forest algorithm Random forest algorithm are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of individual trees.

Linear Regression Linear Regression is a linear approach to modelling the relationship between a dependent variable and one or more independent variables.

Support Vector Machines SVM is binary classification algorithm. Given a set of points of 2 types in N dimensional place, SVM generates a (N—1) dimensional hyper plane to separate those points into 2 groups.

Unsupervised learning algorithms

There are no labels associated with data points. These machine learning algorithms organize the data into a group of clusters to describe its structure and make complex data look simple and organized for analysis. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modelled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance.

Clustering

Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for cluster analysis include gene sequence analysis, market research, and object recognition. Common algorithms for performing clustering include k-means and k-medoids, hierarchical clustering, Gaussian mixture models, hidden markov models, self organizing maps, fuzzy c-means clustering and subtractive clustering.

Common clustering algorithms include:

Hierarchical clustering builds a multilevel hierarchy of clusters by creating a cluster tree.

K-Means clustering partitions data into k distinct clusters based on distance to the centroid of a cluster.

Gaussian mixture models models clusters as a mixture of multivariate normal density components.

Self-organizing maps uses neural networks that learn the topology and distribution of the data.

Hidden Markov models uses observed data to recover the sequence of states.

1.4 OBJECTIVE

- To determine the average time spent by the patient in the OPD.
- To identify the factors responsible for prolonged waiting time in the OPD.
- To study the causes of the delays and suggest interventions.
- To assess the patients satisfaction with the OPD services provided.

CHAPTER 2

LITERATURE SURVEY

In this chapter we focus on the literature survey. The literature survey gives as the clear idea about the existing system.

Wen- Jen Chang et al., [1] presented his work on "Designing a patient-centred appointment scheduling with artificial neural network and discrete even simulation". This study aims to propose a framework for individualized outpatient appointment scheduling (OAS) in a dental clinic which composed of one attending dentist and two resident dentists. To design the OAS, the prediction model of the treatment duration of an individual patient was established by using artificial neural network. Secondly, discrete event simulation method was used to develop the simulation model which simulates the operations of the studied dental clinic. Finally, the established simulation model was used to evaluate the performance of the appointment scheduling. The proposed model consists of number of main features, the service providers composed of multiple dentists with different professional competence levels there are two types of patients, patient no-shows was considered, a variety of medical treatments are provided to the patients. The results of the study show that the proposed OAS can effectively improve the service performance of the dental clinic, this could be caused by patient's characteristics were taken into consideration of building an appropriated appointment interval.

Kay Ross et al., [2] presented his work on "Reducing Length of Stay in Emergency Department". In this paper, a simulation model of an emergency department(ED) at a large community hospital in Lexington is developed. Using such a model, we can accurately emulate the patient flow in the ED and carry out sensitivity analysis to determine the most critical process for improvement in quality of care. In addition, a what if analysis performed to investigate the potential change in operation policies and its impact. Floating nurse, combining registration with triage, a mandatory requirement of physicians visit within 30 minutes, and simultaneous reduction of operation times of most sensitive procedures can all result in substantial improvement. These recommendations have been submitted to the hospital leadership, and implementations are in progress. Discrete event simulation is an effective tool for hospital management to access the inefficiency of existing system, re-evaluate and design the processes, study the impact of potential changes , and investigate the complex relationships of system variables. The goal of simulation is to emulate the patient flow and provide insights for potential development to reduce length of stay. Specifically the model should be able to evaluate the potential ways to reduce length of stay, through identifying the resource shortages, the most criteria operations, and managerial alternatives.

Sharmila Savanth et al., [3] presented his work on “Hospital queuing recommendation system based on patient treatment time”. One of the major problems faced by the today hospitals is the lack of effective technique to manage the patient queue which has led to the patient overcrowding and increased patient waiting delays. Unnecessary waits for longer periods causes negative impact on human resource and also on the patient’s valuable time and thereby increases patient frustration. The waiting time of each patient in the queue is the sum of the treatment time of all the patients standing before him/her in the queue. It will be good if the patient is able to receive an effective treatment plan recommendation and is able to know in advance his/her total waiting time in the hospital. Therefore, a process named patient treatment time prediction is proposed that calculates the waiting time of patients for each of the treatments. On the basis of the waiting time calculated by the PTTP, a system called hospital queuing recommendation system is developed whose task is to recommend an efficient treatment plan for each of the patients in the queue, thereby minimizing the patient waiting time delays. This paper focusses on helping the patients to complete all the treatments in a predictable amount of time and aiding the hospitals to overcome the problem of patient overcrowdings and ineffective management of queues.

Tannaz Sattari et al.,[4] presented his work on “Towards a patient satisfaction based hospital recommendation system”. Surveys are used by hospitals to evaluate patient satisfaction and to improve operation. Collected satisfaction data is usually represented to the hospital administration using statistical charts and graphs. Although this statistical data and visualization is helpful, but because of the size and dimension of the dataset, it is very difficult if not impossible, to identify important factors that could be evaluated and improved for better patient satisfaction. This work presents an unsupervised data-driven methodology that discovers the specific issues reflected by the dataset from the patients’ point of view. The goal of the introduced exploratory data analysis methodology is to determine hidden patterns in the dataset and to identify the main causes of dissatisfaction. To this end, two layers of data analysis is performed. In the first layer, the analysis is only performed on the satisfaction questions. The analysis consists of handling the high dimensionality using self-organizing maps, grouping similar patients using clustering methods and labelling each cluster according to their salient features. In the second layer, demographic data of patients of each cluster is fed to the same analysis process. Putting the salient features of a cluster and its sub-clusters together, one can extract correlations. The correlations are validated using multiple statistical methods applied to the dataset. In a following work, the correlations extracted using this methodology will be ranked and converted to recommendations that can be used by healthcare providers as well as patients.

Hanqing chao et al., [5] presented his work on “Population density based hospital recommendation”. The difficulty of getting medical treatment is one of major livelihood issues in China. Since patients lack prior knowledge about the spatial distribution and the capacity of hospitals, some hospitals have abnormally high or sporadic population densities. This paper presents a new model for estimating the spatiotemporal population density in each hospital based on location-based service (LBS) big data, which would be beneficial to guiding and dispersing outpatients. To improve the estimation accuracy, several approaches are proposed to de-noise the LBS data and classify people by detecting their various behaviours. In addition, a long short-term memory (LSTM) based deep learning is presented to predict the trend of population density. By using large-scale LBS logs database, we apply the proposed model to 113 hospitals in Beijing, P. R. China, and constructed an online hospital recommendation system which can provide users with a hospital rank list basing the real-time population density information and the hospitals' basic information such as hospitals' levels and their distances. We also mine several interesting patterns from these LBS logs by using our proposed system.

Nang Laik MA et al., [6] presented his work on "Predictive analytics for outpatient appointments". Healthcare is a very important industry where analytics has been applied successfully to generate insights about patients, identify bottleneck and to improve the business efficiency. In this paper, we aim to look at the patient appointment process as the hospital is experiencing high volume of "no shows" have a high impact on longer appointment lead time for patients, poor patient satisfaction and loss of revenue for the hospital. We use data analytics to identify pattern of "no shows" and finally operationalizing the model to embed the analytics solution in the business process to reduce the number of "no shows" in the hospital. Exploratory data analysis (EDA) was used to find out the major causes of no shows based on patient demographic information, patient appointment detail and SMS reminder response. Data mining techniques such as logistic regression and recursive partitioning were used on training, test and validation data to predict patients who have high probability of "no show". Our logistic regression model could predict around 70% of no show cases correctly with a kappa coefficient of 0.41 on validation data. Based on finding, we have recommended different strategies to the operations staff for possible reduction of no show slots.

sourabh teli et al., [7] presented his work on “Predicting waiting time of patients using random forest algorithm and design of HQR system”. Patients wait delay and patient overcrowding is one of the major problems faced by hospital. A patient is usually required to undergo various examinations, inspection or tests according to his conditions. This waiting time increases the frustration on patients. Patient Queue Management and wait time prediction form challenging and complicated job because each patient might require different phases and operations such as check-up and various tests. Random Forest Algorithm (RFA) is used for Data mining of big data Furthermore, this implementation can also be applied to Time Prediction. Use of H-Base will give historical data of patients. HQR and RF are parallelized on hadoop platform. Android Platform is used for providing Graphical User interface. Patient treatment time prediction (PTTP) uses RF algorithm for its implementation. Based on hospital queuing recommendation system is diagnosed. PTTP algorithm is proposed as algorithm for calculating the waiting time of patients.

C.Elvira et al., [8] presented his work on “Machine learning based no show prediction in outpatient visits”. A recurring problem in healthcare is the high percentage of patients who miss their appointment, be it a consultation or a hospital test. The present study seeks patient’s behavioural patterns that allow predicting the probability of no-shows. We explore the convenience of using Big Data Machine Learning models to accomplish this task. To begin with, a predictive model based only on variables associated with the target appointment is built. Then the model is improved by considering the patient’s history of appointments. In both cases, the Gradient Boosting algorithm was the predictor of choice. Our numerical results are considered promising given the small amount of information available. However, there seems to be plenty of room to improve the model we manage to collect additional data for both patients and appointments.

Erjie Ang et al., [9] presented his work on “Accurate ED waiting time prediction”. This paper proposes the Q-Lasso method for wait time prediction, which combines statistical learning with fluid model estimators. In historical data from four remarkably different hospitals, Q-Lasso predicts the emergency department (ED) wait time for low-acuity patients with greater accuracy than rolling average methods (currently used by hospitals), fluid model estimators (from the service OM literature), and quantile regression methods (from the emergency medicine literature). Q-Lasso achieves greater accuracy largely by correcting errors of underestimation in which a patient waits for longer than predicted. Implemented on the external website and in the triage room of the San Mateo Medical centre (SMMC), Q-Lasso achieves over 30% lower mean squared prediction error than would occur with the best rolling average method. The paper describes challenges and insights from the implementation at SMMC.

Natchaya et al., [10] presented his work on “Patients waiting time reduction in outpatient department”. Outpatient department has become an essential part of the hospital due to the fact that it is the first step of the treatment system. This leads to the long waiting times especially in public hospitals in Thailand. Patients always have long waiting times for a treatment followed by short consultations. In this study, Operation Research is applied to improve the system. We found out that the reason for long waiting times is that there are too many patients who come in at the same period of time. A discrete event simulation of the outpatient department was developed to examine the system by applying an appointment system to reduce the congestion of patients. The results identified the best appointment system suited for the outpatient department which has more walk-in patients and a high variability of consultation time. This could reduce the waiting times of patients without adding more resources.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

A recurring problem in healthcare is the high percentage of patients who miss their appointment, be it is a consultation or a hospital test. In our existing system random forest algorithm was used in reducing the outpatient waiting time. Outpatient data records are unstructured and inappropriate to build machine learning model. Huge pre-processing is required to transform it into a model suitable for prediction. Almost all the existing models only helps in predicting the waiting time. They operate only for binary class predictions. In our existing system patient no shows visits are mainly considered. Regarding the waiting time reduction we want to make, we may find three possible scenarios:

- Patient who had previously requested an appointment and attend the doctor's consultation.
- Patient who had previously requested an appointment and do NOT attend the doctor's consultation.
- Patient who had previously not requested the appointment and attend the doctor's consultation.

Random forest algorithm were used for the outpatient waiting time reduction. The working of random forest algorithm is given below.

- Randomly select subset of records from the available records.
- Randomly select some features from the available number of features in each subset of records.

For a selected node, calculate the best split among the selected features.

- Split the node into two daughter nodes using the best split.

- Repeat the first three steps until certain number of nodes has been reached
- Build the forest by repeating the first four steps for the total number of trees to be constructed.

3.1.1 LIMITATIONS

- The main disadvantage of random forests is their complexity. They are much harder and time consuming to construct than decision trees.
- They also require more computational resources and also less intuitive. When you have a large collection of decision trees it is hard to have an intuitive grasp of the relationship existing in the input data.

3.2 PROPOSED SYSTEM

In our system, we have used support vector machine algorithm for reducing the outpatient waiting time as this algorithm provides higher accuracy than random forest algorithm which was used in the earlier methods.

Linear Regression

Linear Regression is the most common predictive model to identify the relationship among the variables. Apart from univariate or multivariate data types the concept is linear. Linear regression can be either simple linear or multiple linear regression. The linear regression is the process of prediction using single independent variable which is univariate regression analysis. Simple linear regression distinct the dependent variables and independent variables to extent the relationship between two variables as similar to correlation but correlation does not distinct the dependent and independent variables.

$Y = a + bx$ ------------→equation 1

Where Y - dependent variable

X - independent variable Y- Patient who doesn't appeared

a - intercept of y X- Patients waiting time

b - slope of the line

Decision tree

A decision tree is a flowchart like structure in which each internal node represents a "test" on the attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules. Decision trees are commonly used in operations research and operations management. Another use of decision tree is a descriptive means for calculating conditional probabilities.

Gini Index

Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.

Steps to Calculate Gini for a split,

- Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure **(p^2+q^2)-------→** equation 2
- Calculate Gini for split using weighted Gini score of each node of that split.

Information Gain

Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one.

Entropy can be calculated using formula

Entropy=-plog2p-qlog2q-------------→ equation 3

Here p and q is probability of success and failure respectively in that node. Entropy is also used with categorical target variable. It chooses the split which has lowest entropy compared to parent node and other splits.

Steps to calculate entropy for a split:

- Calculate entropy of parent node
- Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

Naive Bayes

The Bayesian classifiers are statistical classifiers. Naïve Bayes algorithm is one of the most robust machine learning algorithms for rainfall outpatient waiting time reduction. The Naïve Bayes classifier is based on Bayes rule of conditional probability. It analysis each attribute individually and assumes that all of them are independent and important. Naive Bayes classifiers have been used extensively in fault-proneness prediction. An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters necessary for classification.

P(c/x) = P(x/c) P(c)/P(x) ---------→ equation 3

Where P(c/x) – posterior probability of class

P(c) – prior probability of class

P(x/c) – probability of predictor

P(x) – prior probability of predictor

Support Vector Machine

SVM is a supervised Machine Learning algorithm which can be used for both classification and regression challenges. However it is mostly used in classification problems. SVM are a subclass of supervised classifiers that attempt to partition a feature space into two or more groups. Then, we perform classification by finding the hyper-plane that differentiates the two classes.

Support Vector Machines are particularly suited to handle such tasks. Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyper planes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE SPECIFICATIONS

Hardware is the physical components of the computer like microprocessor, hard disks, RAM, and motherboard. Hardware devices are the executors of the commands provided by software applications. Computer hardware as the electronic, magnetic, and electric devices that carry out the computing functions.

Platform: Windows 10

Processor: INTEL Pentium

RAM Capacity: 4 GB RAM

Hard disk: 40 GB

4.2 SOFTWARE SPECIFICATIONS

Software includes all the various forms and roles that digitally stored data may have and play in a computer (or similar system), regardless of whether the data is used as code for a CPU, or other interpreter .Software thus encompasses a wide array of products that may be developed using different techniques such as ordinary programming languages, scripting languages and etc.

Operating system: windows 10

Language : Python

Tool : Jupyter Notebook

4.3 ABOUT THE SOFTWARE

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

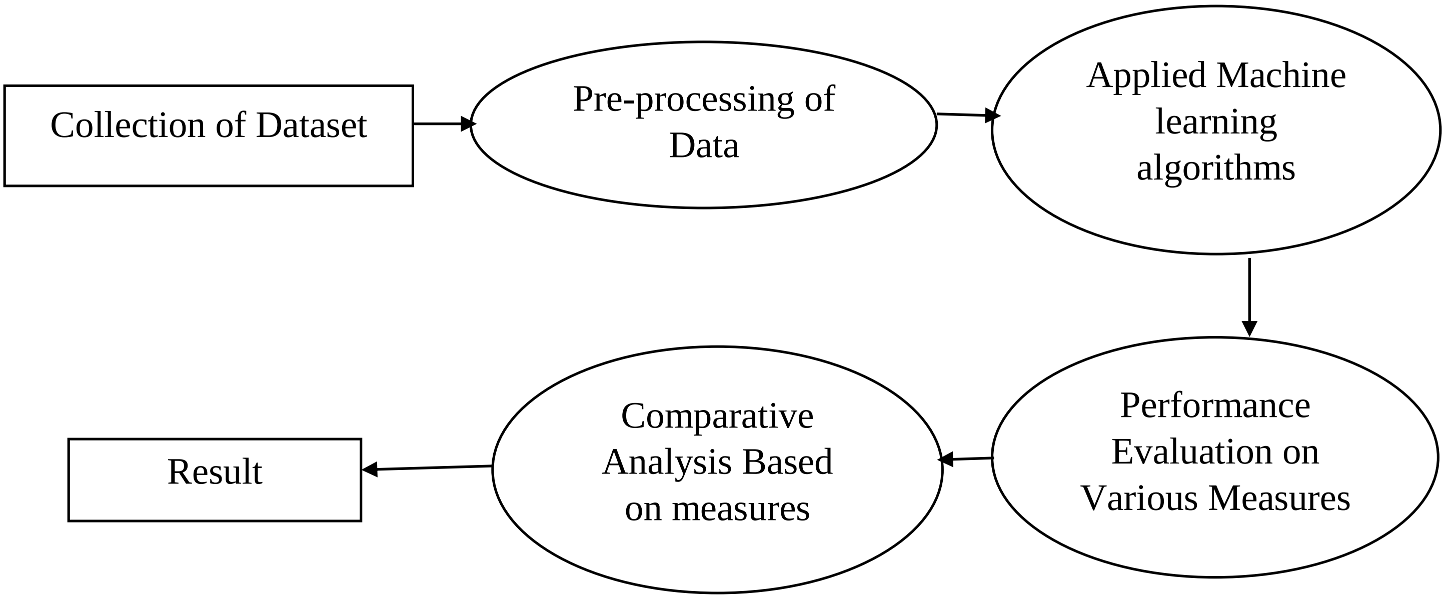

Collection of Dataset
Pre-processing of Data
Applied Machine learning algorithms
Performance Evaluation on Various Measures
Comparative Analysis Based on measures
Result

Fig 5.1 System architecture

5.2 METHODOLOGY

Step1: Take the outpatient dataset for preceding algorithm.

Step2: Pre-process the dataset.

Step 3: Take away attributes on both train and test dataset.

Step 4: Initialize the algorithms viz., linear regression, decision tree, naïve bayes, random forest and support vector machine.

Step 5: To apply the input to the algorithms and gather the computed output.

Step 6: To compute MSE, RMSE and ACCURACY.

Step 7: Apply the above steps for each training model until the desired output.

5.3 DATAFLOW DIAGRAM

A dataflow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

5.3.1 LEVEL 0 DFD

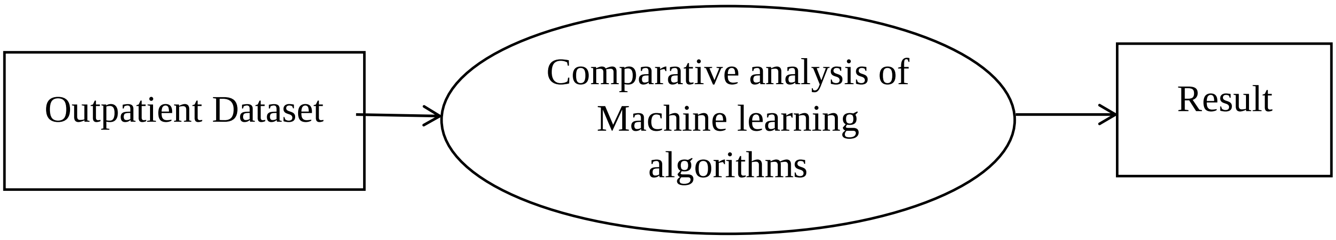


Fig 5.3.1 level 0 DFD

5.3.2 LEVEL 1 DFD:

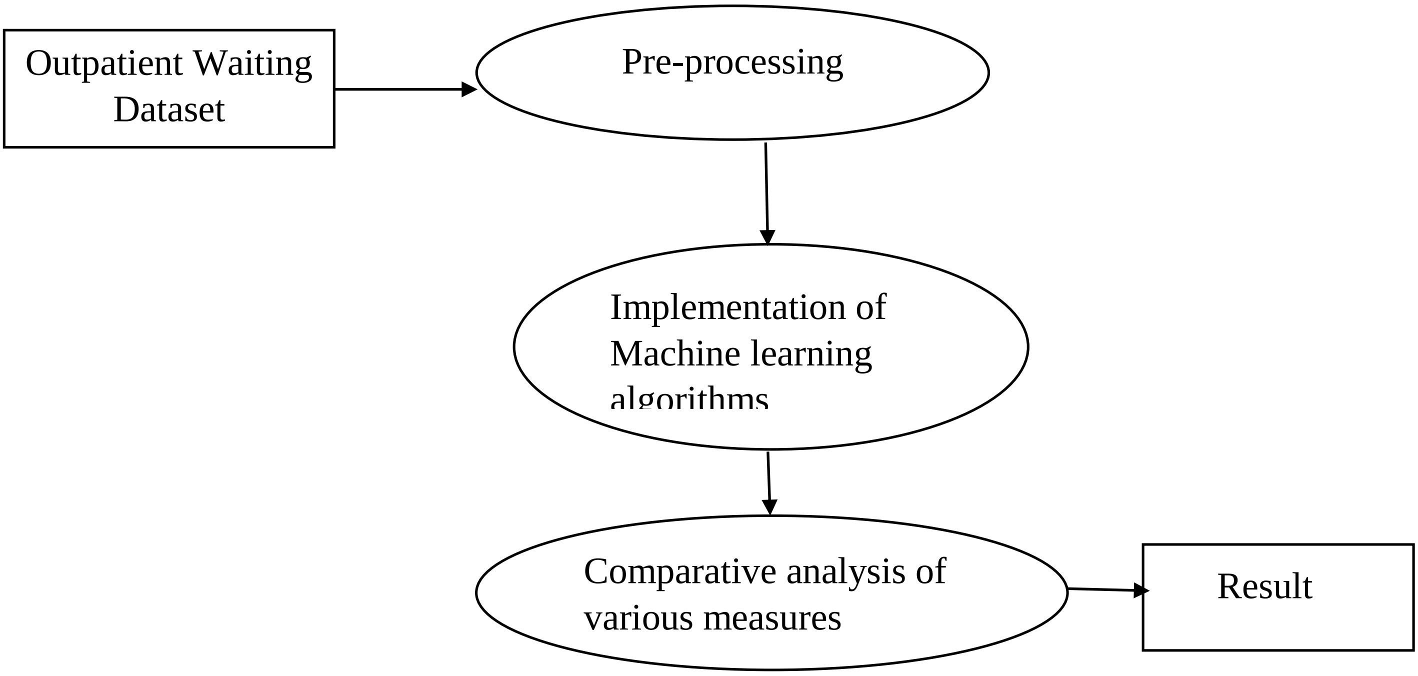

Outpatient Waiting Dataset
Pre-processing
Implementation of Machine learning algorithms
Comparative analysis of various measures
Result

Fig 5.3.2 level 1 DFD

CHAPTER 6

IMPLEMENTATION

6.1 DATA COLLECTION

Data is a piece of information that should be collected carefully so that the collected information is useful. Data collection is an important step while doing experiments or researches. Data collection is the process of gathering or processing information that is used for obtaining outcomes in experiments. The data is collected from kaggle dataset. It contains sanfransico outpatient data set. The dataset contains 8 attributes with 180 records. These data are outpatient waiting time details under the disease related to eye, ear and teeth. The patient who doesn’t appeared under the above diseases also given in the dataset. They are pre-processed and converted into csv format. Later the csv format is loaded into Jupyter notebook.

6.2 DATA PREPROCESSING

The main challenge in reducing outpatient waiting time is the poor data quality and selection. For this reason we try to pre-process data carefully to obtain accurate and correct prediction results. In this phase unwanted data or noise is removed from the collected data set which is done by removing the unwanted attributes and keeping the most relevant attributes that help in better prediction. Another major issue that is to be rectified is the missing values in the collected data set. Missing values in the data set is filled by using various techniques. In this work, the missing values for attributes in the dataset are replaced with the modes and means based on existing data. Adding the missing values provides a more complete dataset for the classifiers to be trained on. Data mining is the process of extracting the useful information from a large collection of data which was previously unknown. For extracting useful information we need to follow data mining process model that will give us clean valuable dataset for model computation and better prediction. Very rarely data are available in the form required by the data mining algorithms. Most of the data mining algorithms would require data to be structured in a tabular format with records in rows and attributes in columns. Not all discovered patterns leads to knowledge. It is up to the practitioner to invalidate the irrelevant patterns and identify meaningful information.

6.3 IDENTIFICATION OF SUITABLE MACHINE LEARNING ALGORITHM

There are numerous machine learning algorithms available today. The suitable algorithm for the particular dataset must be researched to have a greater accuracy in the prediction of the dependent variable. SVM algorithm is the most popular and highly accurate algorithm in finding out the correct output. Random Forest is intrinsically suited for multiclass problems, while SVM is intrinsically two-class. Linear regression works well with one or more independent variables but accuracy is lower than other classification algorithms. Decision tree works well for both continuous and categorical variables but less appropriate for estimation tasks. Naïve bayes algorithm works well with smaller input data but accuracy is lower than support vector machine algorithm. Hence, support vector machine has better accuracy than other classification algorithms.

6.3.1 ALGORITHMIC IMPLEMENTATION USING JUPYTER

To install the jupyter Notebook,

- Download Anaconda. We recommend downloading Anaconda’s latest python 3 version (currently python 3.5).
- Install the version of Anaconda which you downloaded, following the instructions on the download page.
- You have installed jupyter notebook successfully.

Working with jupyter,

- You add a folder inside the working directory that contains all the notebooks.
- Open the terminal and write
    - mkdir jupyter
    - jupyter notebook
- You can see the new folder inside the environment. Click on the folder jupyter_tf.
- Inside this folder, you will create your first notebook. Click on the button new and python 3.
- You are inside the jupyter environment. So far, your notebook is called untitled. Ipynb . This is the default name given by Jupyter. Let's rename it by clicking on File and Rename.
- You are ready to write your first line of code.

6.4 VARIOUS MEASURES

6.4.1 MEAN SQUARED ERROR

The mean squared error (MSE) or mean squared deviation (MSD) of an estimator measures the average of the squares of the errors that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate.

MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

$$\mathrm{MSE}(t) = \frac{1}{n}\sum_{i=1}^{k} f_i (x_i - t)^2 = \sum_{i=1}^{k} p_i (x_i - t)^2$$

6.4.2 ROOT MEAN SQUARE ERROR:

The root mean square deviation (RMSD) or root mean square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values observed. The RMSD represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors when computed out of sample.

RMSE= sqrt(f-o)^2

Where, **f**= forecasts (expected values or unknown results)

o= observed values (known results)

6.4.3 ACCURACY

Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For binary classification, accuracy can be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

TP = True Positive FP = False Positive

TN = True Negative FN = False Negative

6.5 EXPERIMENTAL RESULTS

ALGORITHMS	MSE	RMSE	ACCURACY
Linear regression	0.149	0.386	0.527
Decision tree	0.141	0.375	0.777
Naïve bayes	0.170	0.412	0.527
Random forest	0.141	0.375	0.833
Support vector machine	0.141	0.375	0.920

Fig 6.5.1 comparison table

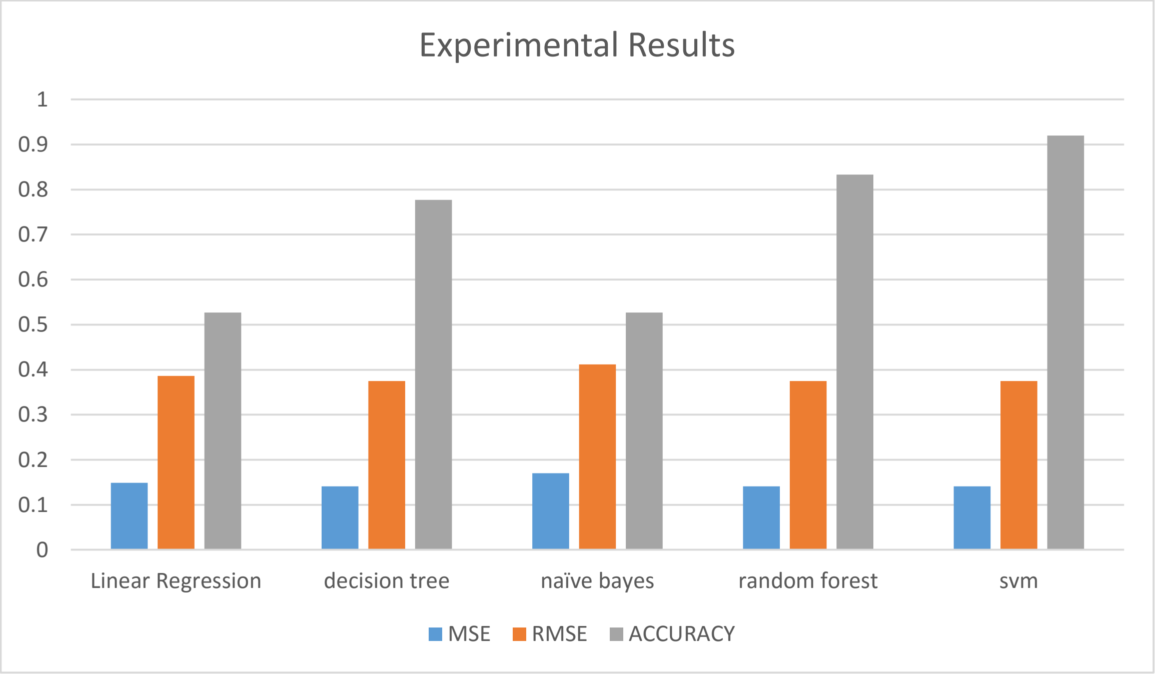


Fig 6.5.2 Experimental results

The experimental results proved that the support vector machine has better accuracy of 0.920 compared to other methods. The support vector machine efficiently classifies the data points with the constructed hyper-plane. The linear regression compares the relationship between two or more variables which has accuracy 0.527. The decision tree and random forest algorithm helps in calculating the conditional probabilities which has accuracy equal to 0.77 and 0.83. Naïve bayes works with smaller training data and predicts the most important parameters for classification which has accuracy equal to 0.52. The proposed system has proved that support vector machine algorithm efficiently classifies data points within the hyper-plane.

CHAPTER 7

CONCLUSION AND FUTURE WORK

CONCLUSION

In day to day life outpatients were so busy and cannot have enough time to wait and see the doctor. The proposed system discussed about the reasons behind the outpatients prolonged waiting time and discussed about the ways to reduce the outpatient waiting time such as reducing the patient cancellation behaviours and many other factors. The existing system have proved that using random forest algorithm outpatient waiting time was greatly reduced. The proposed system have proved that support vector machine classification algorithm has better accuracy than other classification algorithms.

FUTURE WORK

The proposed system compared the accuracy of classification and regression algorithms and estimated that support vector machine has better accuracy than other classification algorithms. In our future work, need to develop the SMS gateway and display the outpatients scheduled time to their mobile phones through an interface. The proposed system works with small amount of outpatient data with hundred and eighty records and working with large set of records is the future scope of this project.

APPENDIX 1

SOURCE CODE

```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from math import sqrt
df =pd.read_csv('C:/Users/prasanth/Desktop/outpatientdata2.csv',encoding='latin1')
df.head()
sns.countplot(x='Age', hue='Patientdoesntappeared', data=df);
df['Timetaken'].describe()
sns.countplot(x='Diseaserelatedtoear', hue='Patientdoesntappeared', data=df);
%matplotlib inline
sns.countplot(x='Diseaserelatedtoear', hue='Patientdoesntappeared', data=df);
names = df['Name']
mobile = df['MobileNumber']
y = df['Patientdoesntappeared']
df.drop(['Name','MobileNumber','Patientdoesntappeared'], axis=1, inplace=True)
from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
X_train, X_holdout, y_train, y_holdout = train_test_split(df.values,
y,test_size=0.3,random_state=17)
tree = RandomForestClassifier(max_depth=5, random_state=17)
knn = KNeighborsClassifier(n_neighbors=10)
tree.fit(X_train, y_train)
# for kNN, we need to scale features
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_holdout_scaled = scaler.transform(X_holdout)
knn.fit(X_train_scaled, y_train)
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from collections import Counter
from sklearn.model_selection import cross_val_score
tree_pred = tree.predict(X_holdout)
print(tree_pred)
print(Counter(tree_pred))
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
mse =  cross_val_score(lm, X_train, y_train, cv=10 , scoring = 'neg_mean_squared_error')
print ("Mean of MSE: ", abs(mse.mean()))
rootMeanSquaredError = sqrt( abs(mse.mean()))
print ("RMSE:",rootMeanSquaredError)
print ("R Squared error =",r2_score(y_test,y))
confusion_matrix(y_holdout,tree_pred)
accuracy_score(y_holdout, tree_pred)
from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
X=df.values
X_train, X_holdout, y_train, y_holdout = train_test_split(X, y, test_size=0.3, random_state=17)
tree = DecisionTreeClassifier(max_depth=5, random_state=17)
```

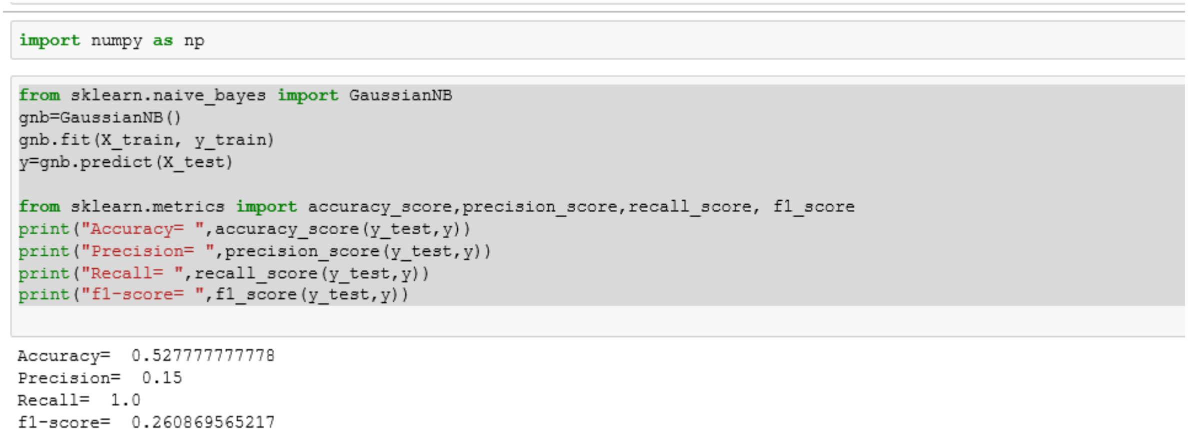
```
knn = KNeighborsClassifier(n_neighbors=10)
tree.fit(X_train, y_train)
# for kNN, we need to scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_holdout_scaled = scaler.transform(X_holdout)
knn.fit(X_train_scaled, y_train)
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.cross_validation import*
from collections import Counter
tree_pred = tree.predict(X_holdout)
print(tree_pred)
print(Counter(tree_pred))
from sklearn.metrics import mean_squared_error
mse =  cross_val_score(lm, X_train, y_train, cv=10 , scoring = 'neg_mean_squared_error')
print ("Mean of MSE: ", abs(mse.mean()))
rootMeanSquaredError = sqrt( abs(mse.mean()))
print ("RMSE:",rootMeanSquaredError)
print(confusion_matrix(y_holdout,tree_pred))
print(accuracy_score(y_holdout, tree_pred))
print(precision_score(y_holdout,tree_pred))
from sklearn.svm import LinearSVC
svc =LinearSVC()
svc.fit(X_train, y_train)
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_squared_error
mse =  cross_val_score(lm, X_train, y_train, cv=10 , scoring =
'neg_mean_squared_error')
print ("Mean of MSE: ", abs(mse.mean()))
rootMeanSquaredError = sqrt( abs(mse.mean()))
print ("RMSE:",rootMeanSquaredError)
accuracy = cross_val_score(svc, X_train, y_train,cv=10,scoring="accuracy")
print("Accuracy= ",accuracy_score(y_train,y1))
dimensions =list(X_train)
weights =svc.coef_
weights=weights[0]
f= zip(weights, dimensions)
l=sorted(f, reverse=True)
l[:5]
significance = tree.feature_importances_
f=zip(significance, dimensions)
l=sorted(f, reverse=True)
l[:5]
y1=tree.predict(X_train)
from sklearn.metrics import accuracy_score,precision_score,recall_score, f1_score
print("Accuracy= ",accuracy_score(y_train,y1))
print("Precision= ",precision_score(y_train,y1))
print("Recall= ",recall_score(y_train,y1))
print("f1-score= ",f1_score(y_train,y1))
dimensions= list(X)
tree.feature_importances
significance = tree.feature_importances_
f=zip(significance, dimensions)
```

```
l=sorted(f, reverse=True)
l[:5]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2,
random_state=42)
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train,y_train)
from sklearn.metrics import mean_squared_error
mse =  cross_val_score(lm, X_train, y_train, cv=10, scoring
'neg_mean_squared_error')
print ("Mean of MSE: ", abs(mse.mean()))
rootMeanSquaredError = sqrt( abs(mse.mean()))
print ("RMSE:",rootMeanSquaredError)
from sklearn.model_selection import cross_val_score
accuracy =  cross_val_score(gnb, X_train, y_train, cv=10)
print("Accuracy= ",accuracy_score(y_test,y))
from sklearn.naive_bayes import GaussianNB
gnb=GaussianNB()
gnb.fit(X_train, y_train)
y=gnb.predict(X_test)
from sklearn.metrics import accuracy_score,precision_score,recall_score, f1_score
print("Accuracy= ",accuracy_score(y_test,y))
print("Precision= ",precision_score(y_test,y))
print("Recall= ",recall_score(y_test,y))
print("f1-score= ",f1_score(y_test,y))
from sklearn.model_selection import cross_val_score
accuracy =  cross_val_score(gnb, X_train, y_train, cv=11)
precision =  cross_val_score(gnb, X_train, y_train, cv=11 , scoring = 'precision')
```

```
recall =  cross_val_score(gnb, X_train, y_train, cv=11 , scoring = 'recall')
print ("Mean Accuracy = ", accuracy.mean())
print ("Mean Recall = ", recall.mean())
print ("Mean Precision = ", precision.mean())
from sklearn.metrics import mean_squared_error
mse =  cross_val_score(lm, X_train, y_train, cv=11 , scoring = 'neg_mean_squared_error')
print ("Mean of MSE: ", abs(mse.mean()))
rootMeanSquaredError = sqrt( abs(mse.mean()))
print ("RMSE:",rootMeanSquaredError)
```

APPENDIX 2

SCREENSHOTS

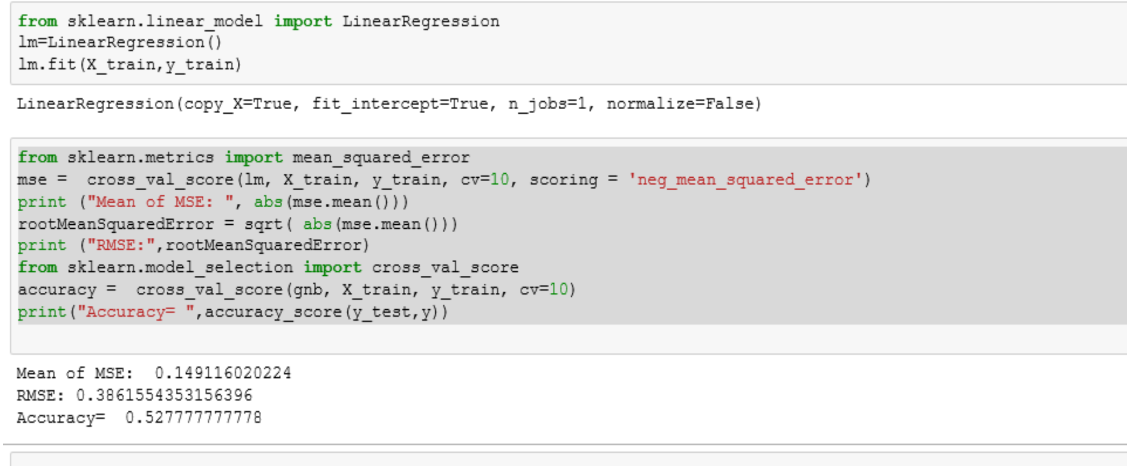


```
import numpy as np
```

```
from sklearn.naive_bayes import GaussianNB
gnb=GaussianNB()
gnb.fit(X_train, y_train)
y=gnb.predict(X_test)

from sklearn.metrics import accuracy_score,precision_score,recall_score, f1_score
print("Accuracy= ",accuracy_score(y_test,y))
print("Precision= ",precision_score(y_test,y))
print("Recall= ",recall_score(y_test,y))
print("f1-score= ",f1_score(y_test,y))
```

```
Accuracy=  0.527777777778
Precision=  0.15
Recall=  1.0
f1-score=  0.260869565217
```

S1: Naïve bayes



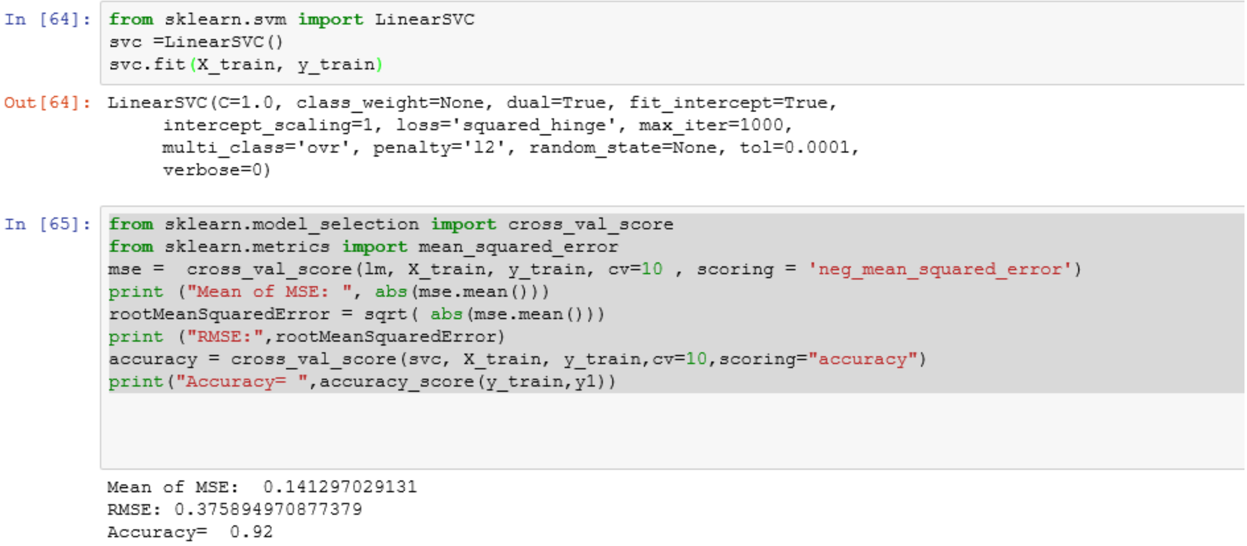
```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

```
from sklearn.metrics import mean_squared_error
mse =  cross_val_score(lm, X_train, y_train, cv=10, scoring = 'neg_mean_squared_error')
print ("Mean of MSE: ", abs(mse.mean()))
rootMeanSquaredError = sqrt( abs(mse.mean()))
print ("RMSE:",rootMeanSquaredError)
from sklearn.model_selection import cross_val_score
accuracy =  cross_val_score(gnb, X_train, y_train, cv=10)
print("Accuracy= ",accuracy_score(y_test,y))
```

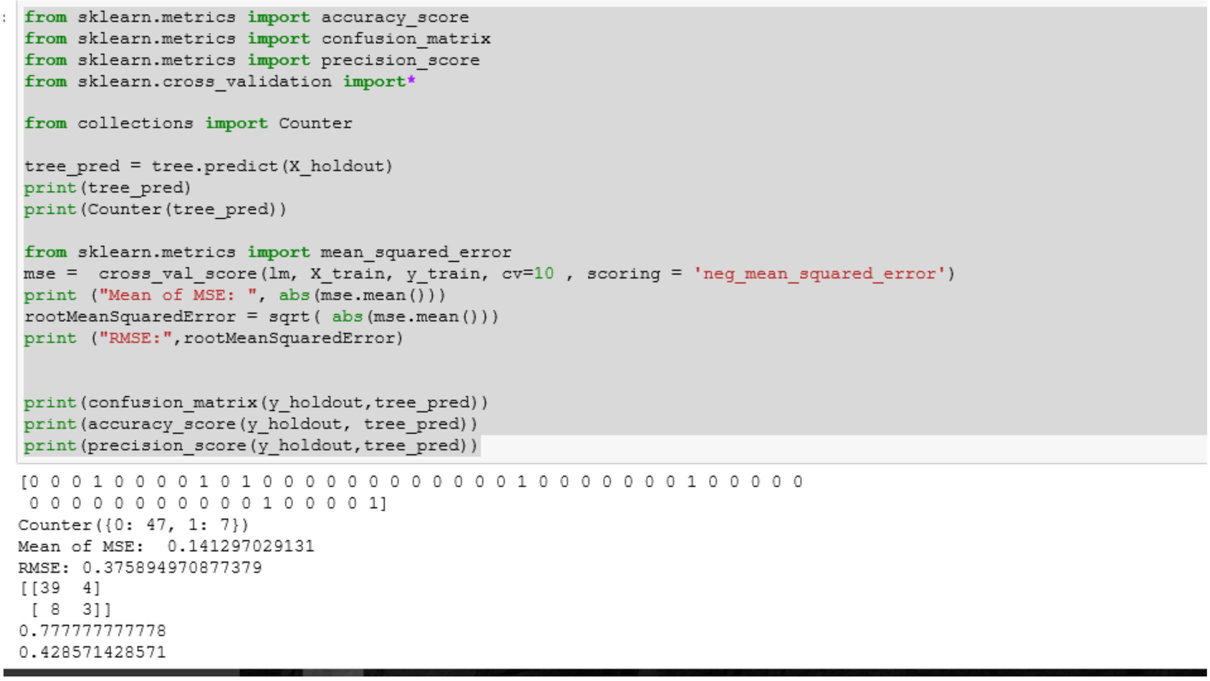
```
Mean of MSE:  0.149116020224
RMSE: 0.3861554353156396
Accuracy=  0.527777777778
```

S2: Linear regression



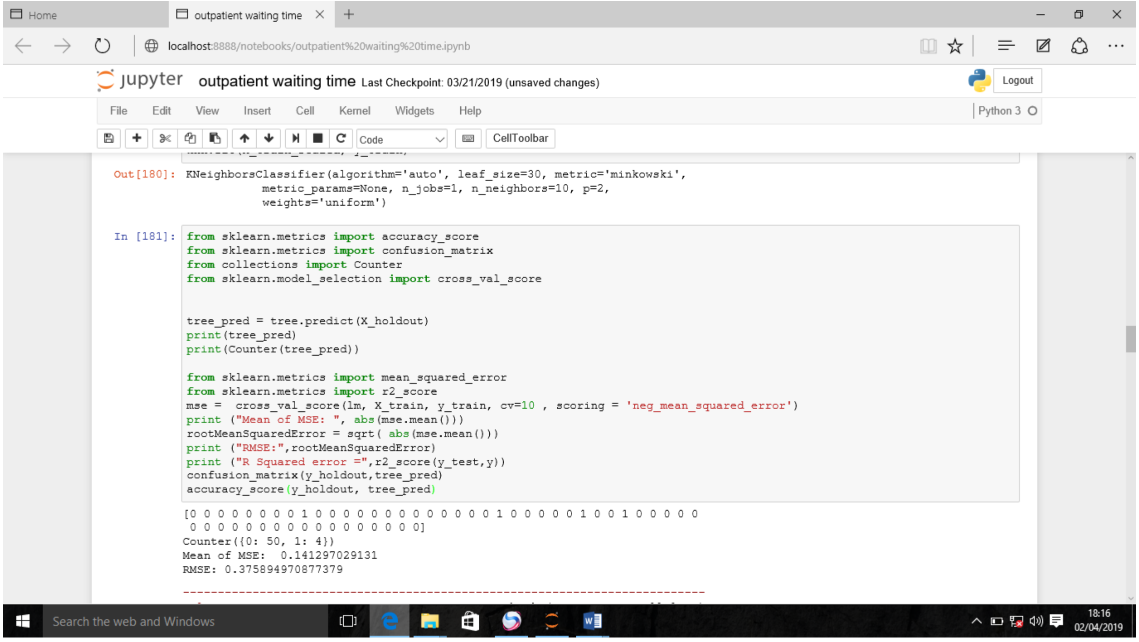
```
In [64]: from sklearn.svm import LinearSVC
         svc =LinearSVC()
         svc.fit(X_train, y_train)

Out[64]: LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
              intercept_scaling=1, loss='squared_hinge', max_iter=1000,
              multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
              verbose=0)

In [65]: from sklearn.model_selection import cross_val_score
         from sklearn.metrics import mean_squared_error
         mse =  cross_val_score(lm, X_train, y_train, cv=10 , scoring = 'neg_mean_squared_error')
         print ("Mean of MSE: ", abs(mse.mean()))
         rootMeanSquaredError = sqrt( abs(mse.mean()))
         print ("RMSE:",rootMeanSquaredError)
         accuracy = cross_val_score(svc, X_train, y_train,cv=10,scoring="accuracy")
         print("Accuracy= ",accuracy_score(y_train,y1))

         Mean of MSE:  0.141297029131
         RMSE: 0.375894970877379
         Accuracy=  0.92
```

S3: Support vector machine



```
: from sklearn.metrics import accuracy_score
  from sklearn.metrics import confusion_matrix
  from sklearn.metrics import precision_score
  from sklearn.cross_validation import*

  from collections import Counter

  tree_pred = tree.predict(X_holdout)
  print(tree_pred)
  print(Counter(tree_pred))

  from sklearn.metrics import mean_squared_error
  mse =  cross_val_score(lm, X_train, y_train, cv=10 , scoring = 'neg_mean_squared_error')
  print ("Mean of MSE: ", abs(mse.mean()))
  rootMeanSquaredError = sqrt( abs(mse.mean()))
  print ("RMSE:",rootMeanSquaredError)


  print(confusion_matrix(y_holdout,tree_pred))
  print(accuracy_score(y_holdout, tree_pred))
  print(precision_score(y_holdout,tree_pred))

  [0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1]
  Counter({0: 47, 1: 7})
  Mean of MSE:  0.141297029131
  RMSE: 0.375894970877379
  [[39  4]
   [ 8  3]]
  0.777777777778
  0.428571428571
```

S4: Decision tree



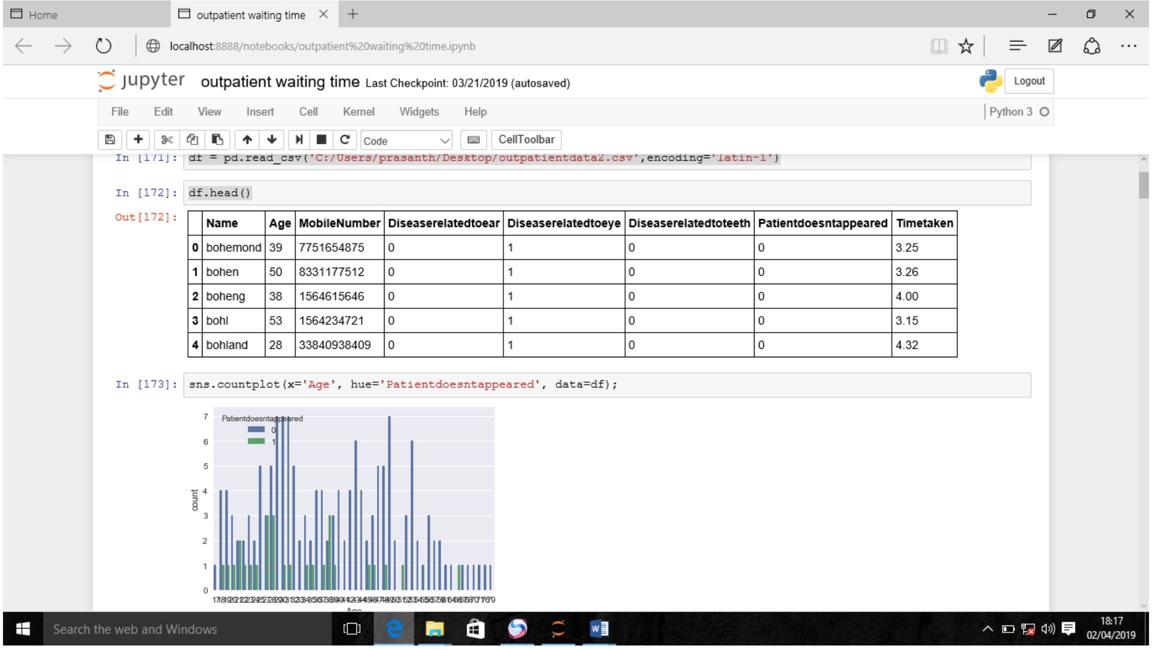
Home
outpatient waiting time
localhost:8888/notebooks/outpatient%20waiting%20time.ipynb
jupyter outpatient waiting time Last Checkpoint: 03/21/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Python 3
Code
CellToolbar
Search the web and Windows
18:16
02/04/2019

```
Out[180]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=1, n_neighbors=10, p=2,
                     weights='uniform')
```

```
In [181]: from sklearn.metrics import accuracy_score
          from sklearn.metrics import confusion_matrix
          from collections import Counter
          from sklearn.model_selection import cross_val_score


          tree_pred = tree.predict(X_holdout)
          print(tree_pred)
          print(Counter(tree_pred))

          from sklearn.metrics import mean_squared_error
          from sklearn.metrics import r2_score
          mse =  cross_val_score(lm, X_train, y_train, cv=10 , scoring = 'neg_mean_squared_error')
          print ("Mean of MSE: ", abs(mse.mean()))
          rootMeanSquaredError = sqrt( abs(mse.mean()))
          print ("RMSE:",rootMeanSquaredError)
          print ("R Squared error =",r2_score(y_test,y))
          confusion_matrix(y_holdout,tree_pred)
          accuracy_score(y_holdout, tree_pred)
```

```
[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Counter({0: 50, 1: 4})
Mean of MSE:  0.141297029131
RMSE: 0.375894970877379

---------------------------------------------------------------------------
```

S5: Random forest Algorithm



```
In [171]: df = pd.read_csv('C:/Users/prasanth/Desktop/outpatientdata2.csv',encoding='latin-1')
```

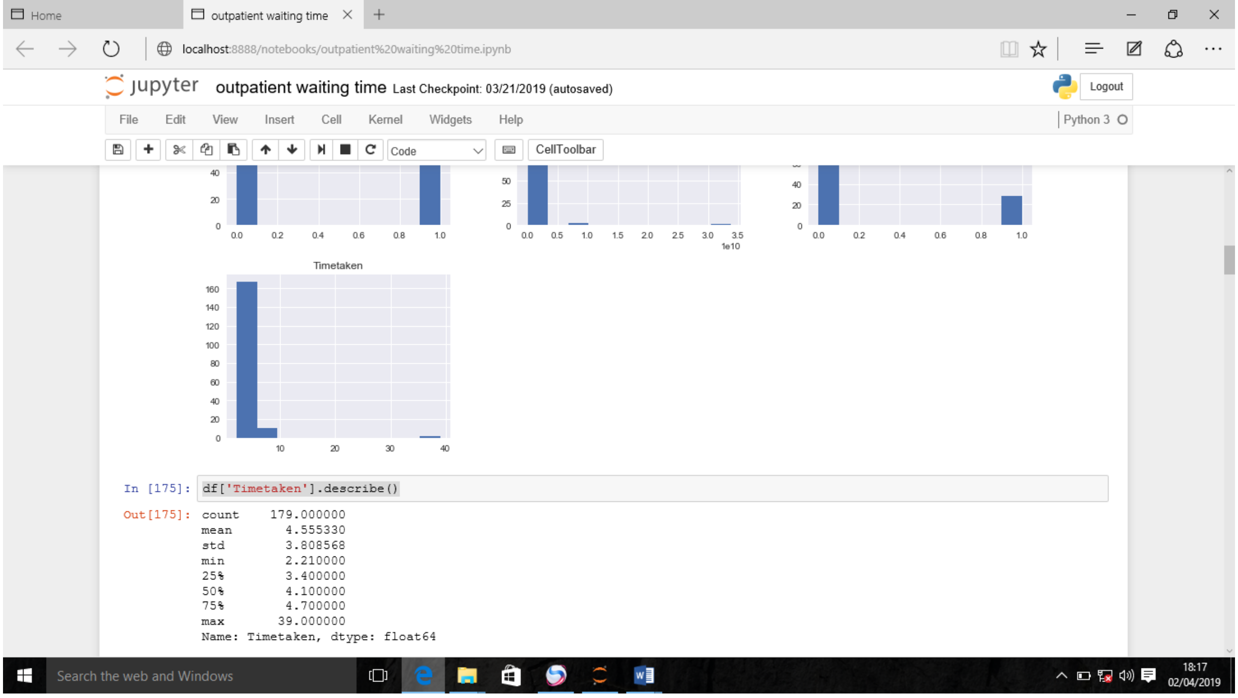
```
In [172]: df.head()
```

Out[172]:

	Name	Age	MobileNumber	Diseaserelatedtoear	Diseaserelatedtoeye	Diseaserelatedtoteeth	Patientdoesntappeared	Timetaken
0	bohemond	39	7751654875	0	1	0	0	3.25
1	bohen	50	8331177512	0	1	0	0	3.26
2	boheng	38	1564615646	0	1	0	0	4.00
3	bohl	53	1564234721	0	1	0	0	3.15
4	bohland	28	33840938409	0	1	0	0	4.32

```
In [173]: sns.countplot(x='Age', hue='Patientdoesntappeared', data=df);
```

S6: outpatient Dataset


outpatient waiting time
localhost:8888/notebooks/outpatient%20waiting%20time.ipynb
jupyter outpatient waiting time Last Checkpoint: 03/21/2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Python 3
Code
CellToolbar
Timetaken
160
140
120
100
80
60
40
20
0
10
20
30
40
Search the web and Windows
18:17
02/04/2019

```
In [175]: df['Timetaken'].describe()
```

```
Out[175]: count    179.000000
          mean       4.555330
          std        3.808568
          min        2.210000
          25%        3.400000
          50%        4.100000
          75%        4.700000
          max       39.000000
          Name: Timetaken, dtype: float64
```

S7: Time taken

REFERENCES

[1] Wen- Jen Chang and yen-hsiang chang,"Designing a patient-centred appointment scheduling with artificial neural network and discrete even simulation", journal of service science and management, pp.71-82,January,2018, DOI 10.4236.

[2] Kay Ross and Kathy Tussey,"Reducing Length of Stay in Emergency Department", journal of IEEE, pp. 395-404,2012,DOI 10.1109

[3] Sharmila Savanth and K.N.Rama Mohan Babu,"Hospital queuing recommendation system based on patient treatment time" , journal of international conference, pp. 579-584,2017.

[4] Tannaz Sattari, Mohammad Reza Khoie , shahram Rahimi, "Towards a patient satisfaction based hospital recommendation system", journal of international joint conference, pp. 1-14,2016.

[5] Hanqing chao and Yuan Chao, "Population density based hospital recommendation", journel of IEEE, pp. 2375-9356,2018, DOI 10.1109.

[6] Nang Laik MA and Dan WU, "Predictive analytics for outpatient appointments", school of information systems, vol.10, no.3, pp. 2615-2618, 2014.

[7] sourabh teli, "Predicting waiting time of patients using random forest algorithm and design of HQR system", journal of IEEE, pp.18–20 December 2014.

[8] C.Elvira and Z.Zeng, "Machine learning based no show prediction in outpatient visits", journal of IEEE, vol.38, no.4, pp. 24-57, 2017

[9] Erjie Ang and C.W.burt, "Accurate ED waiting time prediction", journal of IEEE,vol.13,no.1, pp. 2324-2386, 2016.

[10] Natchaya and S.Samha, "Patients waiting time reduction in outpatient department", journal of joint International Conference ,vol.12,no.6, pp. 20-46, 2018.

[11] J.Feldman, “Appointment scheduling under patient preference and no show behaviour”, Oper.Res., vol.62,no.4, pp. 794-811, 2014.

[12] N.Liu, S. Ziya and V.G.Kulkarni, ”Dynamic scheduling of outpatient appointments under patient no-shows and cancellations”, journal of IEEE.,vol.12,no.2, pp. 347-364, 2010.

[13] R.R.Chen and L.W.Robinson, “Sequencing and scheduling appointments with potential call in patients”, Prod.Oper.Manage.,vol.23, no.9, pp. 1522-1538, 2014.

[14] C.Zacharias and M.Armony,”Joint panel sizing and appointment scheduling in outpatient care”, Manage.Sci.,vol.63, no.11, pp. 3978-3997, 2016.

[15] L.V.Green and S.Savin, ”Reducing delays for medical appointments a queuing approach”, Oper.Res.,,vol.56,no.6, pp. 1526-1538, 2008.

[16] T.Cayiril and E.veral,“Outpatient scheduling in healthcare”, Prod.Oper.Manage., vol.12,no.4, pp. 519-549,2003.

[17] D.Gupta and B.Denton,“Appointment scheduling in healthcare challenges and opportunities”, journal of IEEE ,vol.40,no.9,pp. 800-819,2008.

[18] L.W.Robinson and R.R.Chen,“ A comparison of traditional and open access policies for appointment scheduling”,Manuf.Service Oper.Manage.,vol.12,no.2, pp. 330-346, 2010.

[19] R.Natarajan, “Discrete-time bulk service queuing processes”, Defence sci.j.,vol.12,no.6, pp. 317-326, 2016.

[20] J.Patrick ,“ A markov decision model for determining optimal outpatient scheduling”, Health Care Manage.Sci.,vol.15,no.2, pp. 91-102, 2012.