Reduced Feature based Prediction of Chronic Kidney Disease by Using Machine Learning Classifiers in A Comparative Way

By

SAJEEB CHANDAN SAHA
ID: 151-35-858

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering
Daffodil International University

Spring – 2019

APPROVAL

This **Thesis** titled "**Reduced Feature Based Prediction of Chronic Kidney Disease by Using Machine Learning Classifiers in a Comparative Way**", submitted by **Sajeeb Chandan Saha, 151-35-858** to the Department of Software Engineering,Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Scin Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head — **Chairman**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

07/05/2019

Dr. Md. Asraf Ali
Associate Professor — **Internal Examiner 1**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Mohammad Khaled Sohel
Assistant Professor — **Internal Examiner 2**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Prof Dr. Mohammad Abul Kashem
Professor — **External Examiner**
Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

DECLARATION

It hereby declere that this thesis has been done by **Sajeeb Chandan Saha** under the supervission of **Mr. Md. Anwar Hossen, Lecturer (Senior Scale)**, Department of Software Engineering, Daffodil International University. It also declare that nither this thesis nor any part of this has been submitted elesewhere for award of any degree.

Sajeeb
07/05/19

Sajeeb Chandan Saha
Student ID: 151-35-858
Batch: 16
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:

07.05.19

Mr. Md. Anwar Hossen
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

iii ©Daffodil International University

ACKNOWLEDGEMENT

First of all I am very grateful to the Almighty for allowing me to complete this work as well as my BSc degree. Praise to be God.

I would like to express my sincere gratitude to my respected supervisor **Mr. Md. Anwar Hossen** for consistently supporting me throughout this journey. Without his guidance this work would be incomplete. I would like to appreciate his way of leading the journey throughout the time.

Besides my supervisor I would like to thank my rest of my thesis committee **Prof. Dr. Md. Ashraf Ali** and my department head **Prof. Dr. Touhid Bhuiyan**.

I would like to thank my friends for their support.

Last but not the least, I would like to thank my family: my parents for giving birth to me at the first place and supporting me spiritually throughout my life.

TABLE OF CONTANT

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Background: Chronic kidney disease (CKD) is a long-term condition where the kidneys doesn't work as they should. CKD can range from a mild condition with no or few symptoms, to a very serious condition where the kidneys stop working, sometimes called kidney failure. It can cause several health problem which may lead to serious issue by time. Some of them are high blood pressure, diabetes, high cholesterol, Glomerulonephritis (Damage to tiny filter inside kidneys). This can be prevented by early detection of disease.

Objective: Our main objective is to use machine learning classifier to predict chronic kidney disease on reduced feature based dataset.

Results: In this work I've applied a handful list of machine learning classifier in different way including ensemble learning. Algorithm list includes Naïve Bayes (NB), Random Forest (RF), K-Nearest Neighbor (KNN), Decision Tree (DT), Logistic Regression (LR), and Support Vector Machine (SVM). Among all the test SVM which is a machine learning classifier which tries to find a hyperplane among N-dimensional dataset where N is the number of feature performed as best in terms of accuracy, precision, recall, AUC score, and most importantly lower false negative count.

Conclusions: As medical science is collaborating with different data mining related fields of computer science in future more comprehensive and robust work is possible with larger data set and complex algorithms or neural network. A fuzzy based expert system can be a blessing in this field.

Keywords: Chronic Kidney Disease, Diabetes, Glomerulonephritis, Ensemble, SVM, RF, KNN, NB, DT, LR, Fuzzy, Neural Network;

CHAPTER 1

INTRODUCTION

Chronic kidney disease is a disorder against proper function regarding kidneys. Our kidneys balance the salt and minerals such as calcium, phosphorus, sodium, and potassium that circulates in our blood and also filters wastes from the blood and remove them through urination. This filtering process includes excess fluids from our body. As kidneys filter our blood whenever kidney disease gets worse our blood receives wastes at a higher level which results in sickness. Reduced glomerular filtration rate, increased urinary albumin excretion or both are the key definition terms for chronic kidney disease. This is currently a public health issue. Worldwide 10% of the population is affected by chronic kidney disease (CKD), and millions die each year because they do not have access to affordable treatment (World Kidney Day: Chronic Kidney Disease, 2015). According to the 2010 Global Burden of Disease study, chronic kidney disease was ranked 27th in the list of causes of a total number of deaths worldwide in 1990 but rose to 18th in 2010. This degree of movement up the list was second only to that for HIV and AIDs (Jha V, Garcia-Garcia G, Iseki K, et al. 2013). Chronic kidney disease is a worldwide health crisis. For example, in the year 2005, there were approximately 58 million deaths worldwide, with 35 million attributed to chronic disease, according to the World Health Organization (Levey AS, Atkins R, Coresh J, et al. 2007). There is no early symptoms or sign for this disease at an early stage. The two main causes of chronic kidney disease are diabetes and high blood pressure, which are responsible for up to two-thirds of the cases. The chronic kidney disease also creates some complications like Gout, Anemia, Hyperphosphatemia (Bone disease and high phosphorus), Hyperkalemia (High potassium), and Fluid buildup. This gradual process of kidney damage happens in 5 stages. Early detection of CKD is very helpful for a patient as there are more chances of survival.

1.1 Background

1.1.1 How Kidneys Work

The kidneys lie in the side of the spine between the parietal peritoneum and the posterior abdominal wall. They are protected by ribs, muscles and fat. They are bean-shaped and roughly sized like a fist. Typically a male kidney (125-175 g) is larger than a female kidney (115-155 g). Two main jobs of kidneys are following:

- Filtration
- Execration

As per the internal anatomy of a kidney the outer region is called renal cortex and the inner region is called medulla. Renal pyramids and renal papillae the most characteristic features of medulla are separated by the renal columns which are connective tissue extensions which emit toward down from the cortex through the medulla. Collecting ducts that transport urine made by nephrons to the calyces of the kidney for excretion are bundled by papillae. The main working and functioning unit of a kidney are nephrons. Actually, they are the filtering unit of kidneys. Each of 2 kidneys contains about a million filtering units called nephron. Each nephron works in 2-steps: Filters blood by the glomerulus and then renal tubules absorbs necessary substance into the blood. Renal artery brings blood into the kidney. This large blood vessel reaches nephrons through smaller blood vessels. When the blood enters into nephron it goes into renal corpuscle also known as Malpighian body. Renal corpuscle contains the glomerulus and the Bowman capsule. The glomerulus is the cluster of tiny blood vessels. It filters blood and absorbs larger molecules such as blood cells and proteins into the blood vessel and remaining smaller molecules called capsular urine passes through the Bowman capsule into renal tubules. The renal tubules end up at collecting ducts. Each renal tubule has consisted of PCT (Proximal convoluted tubule), Loop of Henle and DCT (Distal convoluted tubule). A blood vessel runs alongside each tubule. As the filtered fluid moves along the tubule, PCT absorbs water, sodium, and glucose back into blood then Loop of Henle absorbs more potassium, chloride, and sodium into blood then DCT removes excess acid from the blood. By this time fluid is filled with urea which is a byproduct of protein metabolism. At the end of the nephron, there is a

collecting duct where filtered fluids exit the nephrons. Calyces collect remaining fluids and wastes toward the bladder. Here extra fluids and wastes become urine. Finally, the filtered blood flows out from the kidney through the renal vein. If the filter rate of glomerulus falls down gradually the toxic material rate into blood starts increasing. Which ends up in the form of CKD (Chronic Kidney Disease) (Your Kidneys & How They Work, 2018).

1.1.2 Symptoms of CKD

Early CKD does not have any symptoms. Only 50% of renal function can keep a human body stable with no problem. As 2 normal kidneys have 100% renal function capability it is clear that our kidney has greater capability than our body needs. If anyone donates 1 kidney and still can lead a healthy life. Thus despite any severe damage, our kidney keeps working to keep us well. At early stage kidney disease can be diagnosed by conducting the blood and tests. Following are the symptoms of advanced CKD (National Institute of Diabetes and Digestive and Kidney Diseases, 2017):

- chest pain
- dry skin
- itching or numbness
- feeling tired
- headaches
- increased or decreased urination
- loss of appetite
- muscle cramps
- nausea
- shortness of breath
- sleep problems
- trouble concentrating
- vomiting
- weight loss

1.1.3 Cause of CKD

Too much sugar into blood damages kidney's filter over time. Thus an important protein called albumin gets filtered out from the blood through the weak filter and passes through the urine. Also if the blood vessels into the nephrons damage over time blood filtration do not occur properly which leads to CKD. Thus diabetes and high blood pressure is the main cause of CKD as mentioned earlier. Following are the other cause of CKD (National Institute of Diabetes and Digestive and Kidney Diseases, 2016):

- A genetic disorder that causes many cysts to grow in the kidneys, polycystic kidney disease (PKD).
- an infection
- a drug that is toxic to the kidneys
- A disease that affects the entire body, such as diabetes or lupus. Lupus nephritis is the medical name for kidney disease caused by lupus
- IgA glomerulonephritis
- disorders in which the body's immune system attacks its own cells and organs, such as Goodpasture syndrome
- heavy metal poisoning, such as lead poisoning
- rare genetic conditions, such as Alport syndrome
- hemolytic uremic syndrome in children
- Henoch-Schönlein purpura
- renal artery stenosis

1.1.4 Stages of CKD

Chronic kidney disease attacks gradually from stage 1 as a mild attack to stage 5 as complete kidney failure. Following are the kidney disease stage with details (American Kidney Fund):

- Stage 1 CKD: Kidney damage and estimated Glomerular Filtration Rate (eGFR) greater than 90
- Stage 2 CKD: Kidney damage and an eGFR between 60 and 89

- Stage 3: eGFR between 30 and 59
- Stage 4: eGFR between 15 and 30
- Stage 5: eGFR less than 15

1.1.5 Test and Diagnosis

Blood test for Glomerular Filtration Rate can indicate whether you have CKD or not. GFR between 16 and 60 indicated kidney disease where GFR less than 15 is known as complete kidney failure. Creatinine is a waste product from the normal breakdown of muscles in the human body. Kidneys filter out blood by removing creatinine. GFR is calculated from the amount of creatinine residing in the blood. Albumin in the urine indicates the signature of diabetes. Kidney filters are damage due to the heavy level of sugar into the blood. As a result, the kidney fails to absorb albumin into blood thus albumin filters out through urine. UACR (Urine albumin to creatinine ratio) measures the sign of CKD. UACR measure of 30 mg/g or less is normal and more than 30 mg/g is an indication of CKD.

1.1.6 Prevention of CKD

Following are the prevention of CKD (National Institute of Diabetes and Digestive and Kidney Diseases, 2016)

- Make healthy food choice
- Make physical activity part of your routine
- Aim for a healthy weight
- Get enough sleep
- Stop smoking
- Limit alcohol intake
- Explore stress-reducing activities
- Manage diabetes, high blood pressure, and heart disease
- Ask your health care provider questions.
    - What is my glomerular filtration rate (GFR)?
    - What is my urine albumin result?

- What is my blood pressure?
- What is my blood glucose (for people with diabetes)?
- How often should I get my kidneys checked?

1.2 Motivation of the Research

Including bacterial and viral disease, prediction cause the earlier detection of any sort of disease. Earlier detection results in earlier medication. Currently, chronic kidney disease is a public health issue. Complications due to CKD such that anemia, hyperphosphatemia, heart disease, hyperkalemia which may lead to premature death. As there is no early symptom for this disease the only prediction on the test result of blood and urine can uncover the likelihood of CKD. Moreover, early detection can help patients to recover from this state. Early detection depends upon the prediction over the test result. These are my motivation to work on prediction of CKD using machine learning classifiers. Another motivation comes whenever I saw the dataset on chronic kidney disease on the UCI machine learning repository (UCI Machine Learning Repository: Chronic_Kidney_Disease Data Set, 2015).

1.3 Problem Statement

Using Chronic Kidney Disease dataset of UCI machine learning repository I need to create best predictive model regarding accuracy, time and learning rate among selected models by applying various machine learning classifier and comparing their result after data preprocessing and transformation.

1.4 Research Questions

- Can I estimate a better reduced feature standpoint to predict CKD more accurately from the dataset on UCI Machine Learning Repository?
- Can I find a better predictive machine learning model which can predict the presence of CKD in a human body from reduced feature derived from statistical experiment more accurately with lower FN (False Negative) count?

1.5 Research Objectives

- To estimate a better reduced feature standpoint on the dataset used.
- To find a better predictive machine learning model which can predict the presence of CKD in a human body from reduced feature derived from statistical experiment more accurately with lower FN (False Negative) count

1.6 Research Scope

Classification based machine learnings algorithm can predict disease availability or not. But from background study new know that CKD can attack in 5 stages. So in the future, we can work on making a fuzzy based expert system for predicting the accurate likelihood of a patient having CKD.

1.7 Thesis Organization

This thesis is organized in 5 different section. First section describes the complete background of topic. As this topic is related to medical science a few description is available here. Also some statistics of this topic is available. In section 2 some previous works done by others has been described. In chapter 3 complete methodology has been described with proper figure. In chapter 4 thesis result has been discussed with pooper and adequate visualization. In chapter 5 the whole topic has been concluded with current thesis work.

CHAPTER 2

LITERATURE REVIEW

Medical science has been aided by various aspect of computer vision and data science. Early detection of diseases is mostly dependent on enough data which is regulated by well-known physician. According to many research data science has a great impact on chronic kidney disease.

Dr. S. Vijayarani (Dr. S. Vijayarani, et al. 2015) have used two machine learning classifiers named Artificial Neural Network (ANN) and Support Vector Machine (SVM) to predict Chronic Kidney Disease and compared their accuracy and efficiency. They used WEKA tool for finding the best prediction algorithm. Among them they found ANN has the best classification accuracy

Parul Sinha, (Parul Sinha, et al. 2015) has done a comparative study between two algorithms named SVM (Support Vector Machine), KNN (K-Nearest Neighbor). They have used CKD Dataset on UCI Machine Learning Repository and found KNN as best classification algorithm.

Abdulhamit, (Abdulhamit, et al. 2017) has applied Artificial Neural Network (ANN), SVM (Support Vector Machine), C4.5, KNN (K-Nearest Neighbor) and Random Forest (RF) algorithm on CKD Dataset on UCI Machine Learning Repository. Among them they found Random Forest as best classifier among all the algorithms.

Lamborder, (Lamborder, et al. 2015) has used WEKA tool to apply algorithms named Naive-Bayes, Multilayer Perception, Support Vector Machine, J48, Conjunctive Rule and Decision Table on CKD Dataset on UCI Machine Learning to predict CKD. According to them Multilayer Perception has performed better in classification and prediction.

Abhishek, (Abhishek, et al. 2012) have used Back Propagation Algorithm (BPA), Radial Basis Function (RBF) and Support Vector Machine (SVM) to predict Kidney Stone by using WEKA tool. Among them they found Back Propagation Algorithm significantly improved the conventional classification technique.

Ashfaq Ahmed, (Ashfaq Ahmed, K et al. 2013) have used machine learning algorithm to predict Cancer, Liver and Heart Disease. Between the algorithms they have used

named Support Vector Machine (SVM) and Random Forest (RF) they found SVM as the best classification algorithm

Manish Kumar (Manish Kumar, 2016) have used 6 machine learning classifier including RF, SMO, NB, RBF, MLPC, SLG to predict chronic kidney disease on UCI CKD Dataset and among the 6 classifiers he found Random Forest (RF) to be more accurate in terms of AUC, ROC, and MCC score.

Rubini (Rubini, et al. 2015) have used 3 machine learning classifier named as RBF Network, MLP and LR on CKD dataset by UCI to predict chronic kidney disease. They have applied a 10-fold cross validation on their models. Among all 3 classifier they found MLP to be highest accurate classifier in their test.

Chatterjee (Chatterjee, et al. 2017) has proposed a Cuckoo Search (CS) trained Neural Network (NN) or NN-CS based model. This model has been compared well-known classifiers like Multilayer Perceptron Feedforward Network (MLP-FFN) (trained with scaled conjugate gradient descent) and also with NN supported by Genetic Algorithm (NN-GA). Among all of them they found NN-CS based model to be best performer in terms of accuracy, recall and F-measure.

Chen (Chen, et al. 2016) has proposed a fuzzy rule-building expert system (FuRES) and fuzzy optimal associative memory (FOAM) system for predicting CKD on UCI CKD dataset. For comparative study they have applied partial least squares discriminant analysis (PLS-DA). Among all of the algorithms they found FuRES and FOAM both best in terms of accuracy but FOAM is best for sensitivity

Zewei (Chen, et al. 2016) has applied three multivariate machine learning classifier which are KNN, SVM, soft independent modeling of class analogy (SIMCA) on UCI CKD dataset. Among them they have found SVM as the best classification performer in terms of accuracy.

Parthiban and Srivatsa (Parthiban, et al. 2012) used machine learning classifier named Naïve Bayes and SVM to predict diabetes on the dataset of 500 patients collected from Research Institute of Chennai. Among the 500 patients 142 are affected with diabetes where 358 were not affected. In this experiment Naïve Bayes has scored 74% accuracy and SVM performed as the best with 94.60% accuracy.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Almost every research work seeks for an outcome. To conquer the way towards outcome we need to follow some precise rules. Combination of these rules are called methodology for a research work. In this research of finding chronic kidney disease using machine learning classifier the research methodology separated into a few section including data collection, preprocessing, analysis and visualizations of the outcome. Following figure shows the proposed model:

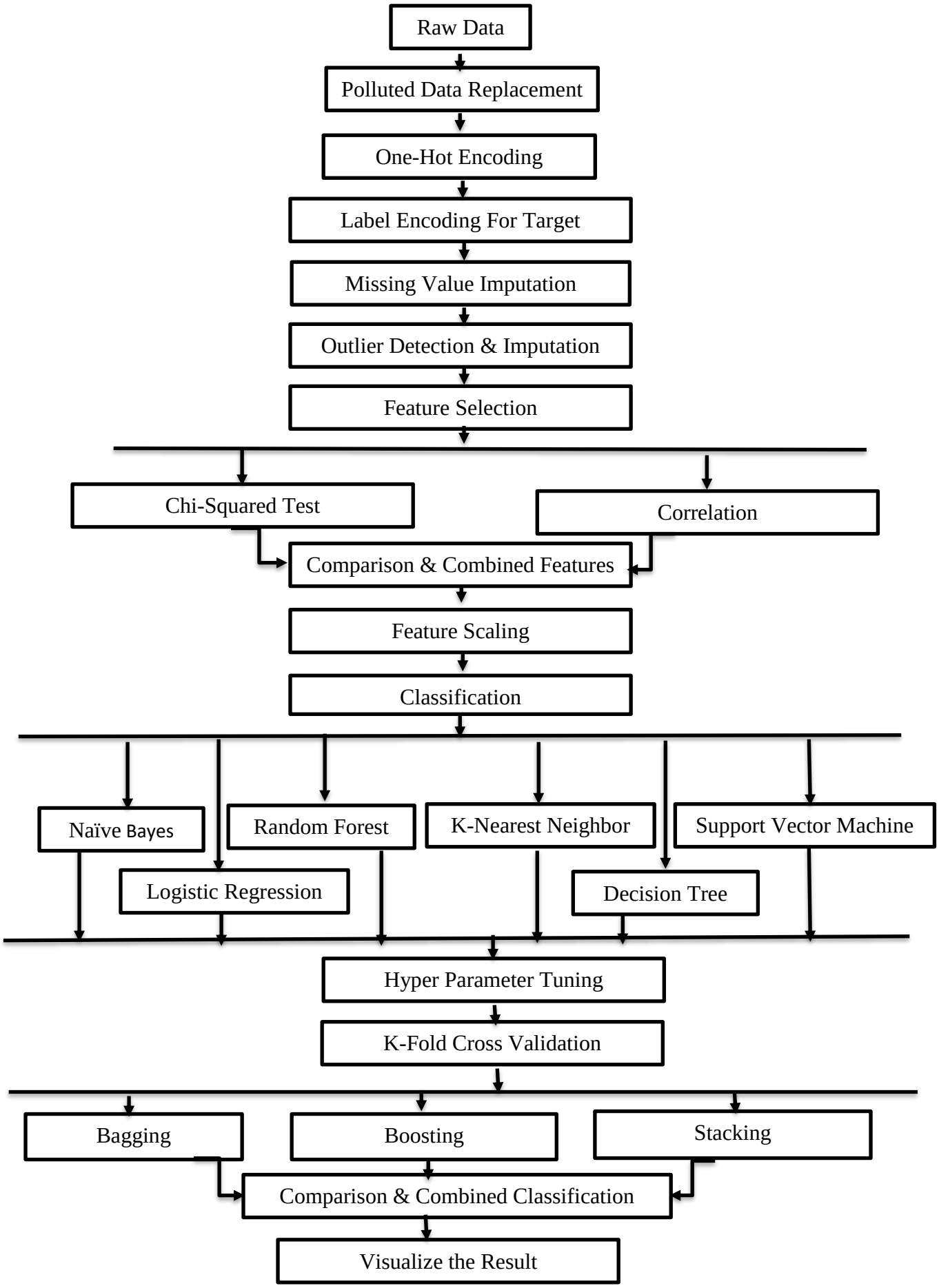

Figure: Proposed machine Learning based model

In this thesis I've prepared the dataset using some data preprocessing steps which includes

- I've replaced the polluted data elements (e.g.: \t, \n) with the pure data.
- Encoded categorical data using one-hot system.
- Encoded target column using label encoder method.
- Replaced missing values.
- Detected outlier using Tukey method and replaced the outlier value with mean value of each column

Now before applying machine learning algorithm I've made to hypothesis. I've applied algorithms on the dataset having all features.

Then I've reduced features using the following method.

- Removed correlated column
  Here I've tested another hypothesis for finding the best threshold value over which value we have removed correlated columns
- Apply Chi square test and selected best feature.
  Here another hypothetical test have been applied to collect the lowest score over which I've taken all the features.

I've ran these hypothetical test using trial and error method to find the best suitable value.

For machine learning task I've applied 6 different algorithms.

- Naive Bayes
  This is a conditional probability model based on Bayes theorem (John, et al. 1995).
- Random Forest
  This uses a forest of tree. Every tree in the forest is influenced by the values of random vectors sampled separately and has identical distribution as any other tree in the forest (L. Breiman, et al. 2001).
- K-Nearest Neighbor
  This algorithm uses the data directly for classification without building a model first (L. E. Peterson, et al. 1883, 2009).
- Support Vector Machine
  This algorithm uses a hyper-plane to make classification among targets (V. Vapnik, et al. 1995).
- Logistic Regression
  It uses a linear regression based model with a sigmoid function which converts the result into binary under some threshold (Hosmer, et al. 1989).

- Decision Tree
  It is a tree based classifier where a leaf is a decision (class) and each non-leaf nodes represents a test (Quinlan J. R., et al. 1985).

All the models have been trained and tested using 10-Fold Cross Validation.

With these algorithms I've applied some ensemble techniques as follows:

- Bagging
  Bootstrap aggregation. It trains multiple instances of same model by resampling subset of data from the dataset.
- Boosting
  In this thesis we have used AdaBoost (Adaptive Boosting) technology which used a weighted graph to treat each missed classification on each run.
- Stacking
  This uses multiple weak learners as base classifier and one learner as Meta classifiers. The Meta classifier learns form then base classifiers and finally expose the result.

Finally I've compared all the result from different classifiers in different methods and found SVM to be best performer.

3.2 Data Collection

Data collection is the most crucial task for building machine learning models as machine learning models learn from data. Data collection is expedient of proper trustworthiness towards a machine learning research. Proper data collection helps organizations to answer question and predict hidden possibilities. For this research the dataset is gathered from an internet source named "UCI Machine Learning Repository" (Chronic_Kidney_Disease Data Set, 2015). The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The archive was created as an ftp archive in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning data sets.

3.3 Data Preprocessing

Data preprocessing is one of the most fundamental task to be performed in data mining techniques. It refers to the task which includes transforming raw data into a reasonable configuration. Raw data or real word data contains missing information, unnecessary, bad formatted, invalid information. And these types of data leads to disaster in prediction by machine learning algorithms. Data preprocessing steps demonstrate the way for handling these issue. For this research different types of fata preprocessing steps are used. All of them are being described as follows.

3.3.1 Polluted Data Replacement

From real word some data inputs come with unnecessary value with them. For example in a random record the value for the attribute age is 30 but it has comes with some special character like 30\t. Which is not necessary as an information. Thus this value should be replaced with the original pure value. Also in some cases where the data is missing the missing field is being replaced with some random character instead of letting the place empty. The algorithm for handling the polluted data is shown in **Table 3.1:**

Table 3. 1 Algorithm for handling polluted data

```
Load_Data
for each cell in row number of total records do
   if the cell contains value with special character
       orig_val ← remove_special_char value
       Value ← orig_val

   else if the cell contains only special character
       then replace the cell with NaN
   else
       continue
   end if
end for
Save_Data
```

3.3.2 One Hot Encoding

One Hot encoding means transforming categorical feature's representation into binary way of representation. As we know that machine learning models perform better for numerical value. One-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0). A similar implementation in which all bits are '1' except one '0' is sometimes called one-cold. In statistics, dummy variables represent a similar technique for representing categorical data. In this techniques all unique category of a nominal feature is taken under consideration. Then those unique values create their own one-hot encoded new feature. Their values are represented in such way that if there is a presence of that feature in any random record then the value is represented by 1 and otherwise 0. Following **Figure 3.1** shows the example of one hot encoding.

pe	ane	ane_no	ane_yes	pe_no	pe_yes
no	no	1	0	1	0
no	no	1	0	1	0
no	yes	0	1	1	0
yes	yes	0	1	0	1
no	no	1	0	1	0
yes	no	1	0	0	1
no	no	1	0	1	0
yes	no	1	0	0	1
no	yes	0	1	1	0
no	yes	0	1	1	0

Figure 3. 1 Example of One-Hot Encoding

3.3.3 Label Encoding for Target Column

Label encoding means encoding nominal feature's representation into numerical representation. As we encode the original value into digit we need to keep track or

evidence of that value or information to map with the original information later. As we already know that for fitting data into a model we need to prepare that data for that model. Most of the classification based model requires numerical representation of data so we need to encode that data and keep the original information evidence for human readability on later. The algorithm for label encoding is shown in **Table 3.2**.

Table 3. 2 Label Encoding Algorithm

```
Load_DataSet
for column of columns
  unique_values ← Finding unique values of the column
    for I = 0 to M – 1 M number of unique values
       Encoding ← Encoded index of I unique_values
    end for
end for
Save_DataSet
```

3.3.4 Missing Value Imputation

In real word data collection scenario there occurs many problem. Some of them causes data missing issue. In this case some observation for data can't be completed with data. Here data can't be gathered properly. Data missing can be occurred due to some reason.

- A few data can be missing while collecting.
- Some data can be damaged due to data breach
- Medical data can be missing due to diagnostic issue.

Missing data is a great problem. There are a few reason for that. Many of machine learning algorithms can be operated on missing data. Many model create biased result due to missing data. All of these issue result in bad predictive model. So missing data should be handled. There are many ways to handle missing data. Some of them are following:

- Delete the feature having missing data
- Delete the record having missing data
- Fill the missing data with some value.

Deleting the feature which have missing data is not a wise decision. As features are most valuable element for data mining process.

Deleting the record is not always a wise suggestion if the total number of record is not large. As machine learning models lean form data and from more data they can learn more.

On another way missing data can be replaced with some value. Best possible way to refill the missing data is to recollect the data. If it is not possible then data can be imputed for every feature by any of the following way:

- Mean
- Median

Mean is calculated from sum of all record when the sum is divided by the total count of all record.

$$\overline{x} = \frac{1}{n}(\sum_{i=1}^{n} x_i) = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{3.1}$$

Median is the middle point of an arranged data set. Thus median separate the higher portion from the lower portion.

$$\text{median}(a) = \frac{a_{\lceil \#x \div 2 \rceil} + a_{\lceil \#x \div 2+1 \rceil}}{2} \tag{3.2}$$

In this research missing value has been imputed from the mean of that feature.

3.3.5 Outlier Detection and Imputation

In statistics an observation is called outlier if that observation is far from other observation. Outlier may be occurred in a data set due to unevenness in the measurement. Otherwise experimental error can be the cause for outlier.

Observed variables often contain outliers that have unusually large or small values when compared with others in a data set. Some data sets may come from homogeneous groups; others from heterogeneous groups that have different characteristics regarding a specific variable, such as height data not stratified by gender. Outliers can be caused

by incorrect measurements, including data entry errors, or by coming from a different population than the rest of the data. If the measurement is correct, it represents a rare event.

Tukey's (1977) method, constructing a boxplot, is a well-known simple graphical tool to display information about continuous univariate data, such as the median, lower quartile, upper quartile, lower extreme, and upper extreme of a data set. It is less sensitive to extreme values of the data than the previous methods using the sample mean and standard variance because it uses quartiles which are resistant to extreme values. The rules of the method are as follows:

1. The IQR (Inter Quartile Range) is the distance between the lower (Q1) and upper (Q3) quartiles.

2. Inner fences are located at a distance 1.5 IQR below Q1 and above Q3 [Q1-1.5 IQR, Q3+1.5IQR].

3. Outer fences are located at a distance 3 IQR below Q1 and above Q3 [Q1-3 IQR, Q3+3 IQR].

4. A value between the inner and outer fences is a possible outlier. An extreme value beyond the outer fences is a probable outlier. There is no statistical basis for the reason that Tukey uses 1.5 and 3 regarding the IQR to make inner and outer fences.

For the previous example data set, Q1=3.725, Q3=4.575, and IQR=0.85. Thus, the inner fence is [2.45, 5.85] and the outer fence is [1.18, 7.13]. Two extreme values, 14 and 15, are identified as probable outliers in this method. Figure 4 is a boxplot generated using the statistical software STATA for the example data set (Seo, et al. 2006).

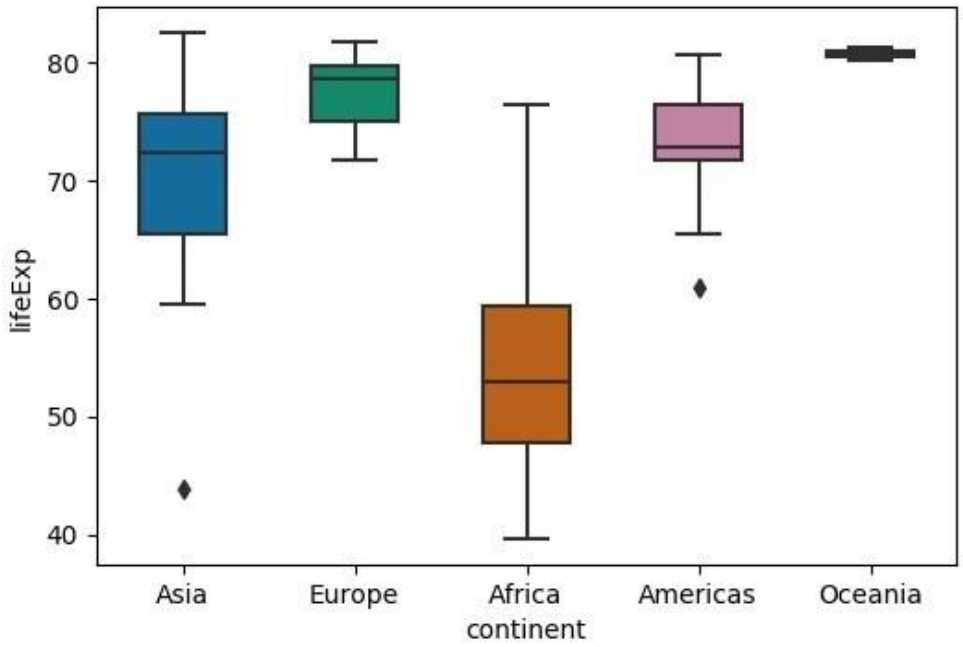

Figure 3. 2 Tukey Boxplot

Removal of records having outlier value is harmful for dataset if the dataset has lesser records. Outlier can be imputed with new value. New value come from mean, median or even from the 1st quartile or 3rd quartile. In this research outlier has been handled only for numerical features as nominal value can never have outlier. Here outlier value has been imputed from the mean of the feature.

Following is the figure of boxplot with outlier:

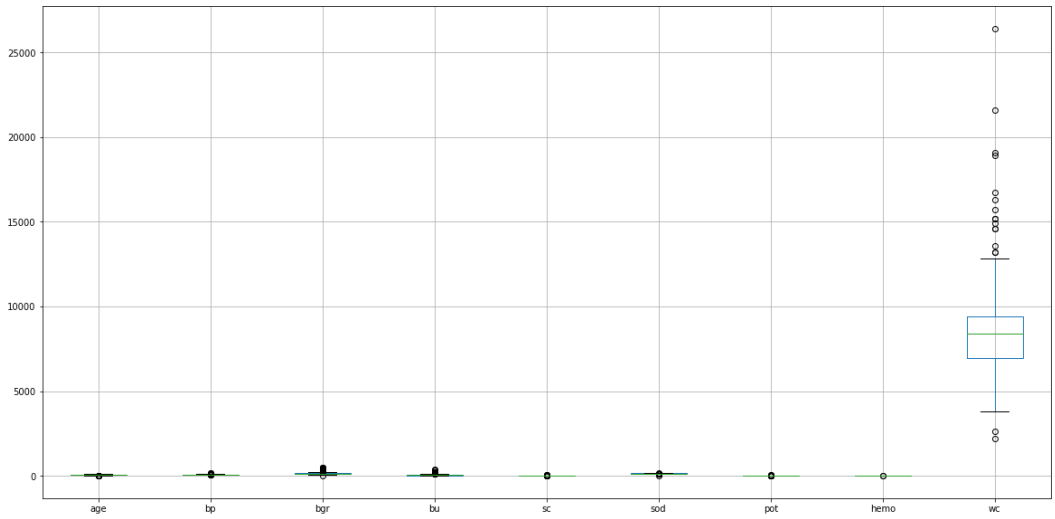

Figure 3. 3 Showing Outlier via Boxplot

After removing outlier following is the figure

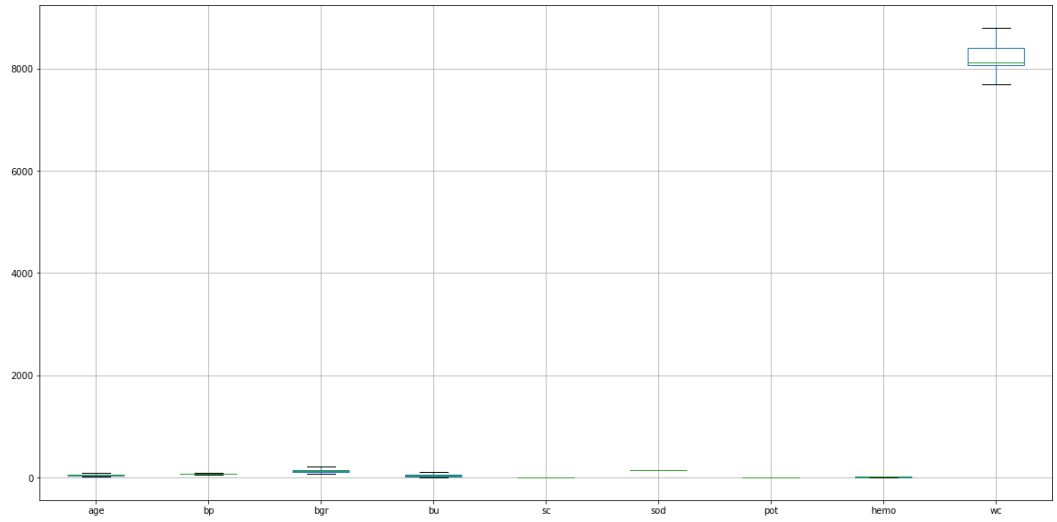

Figure 3. 4 Outlier Removed

3.3.6 Feature Selection

Feature selection has been a functioning exploration territory in design acknowledgment, insights, and data mining networks. The fundamental thought of feature selection is to pick a subset of info factors by wiping out features with almost no prescient data. Feature selection can fundamentally enhance the conceivability of the subsequent classifier models and frequently manufacture a model that sums up better to concealed focuses. Further, usually the case that finding the right subset of prescient features is a vital issue in its own particular right. For instance, doctor may settle on a choice in view of the chose highlights whether a perilous medical procedure is essential for treatment or not. Feature selection is basic to building a decent model for a few reasons. One is that component choice suggests some level of cardinality decrease, to force a cutoff on the quantity of characteristics that can be considered when constructing a model. Data quite often contains more data than is expected to construct the model, or the wrong sort of data. For instance, someone have a dataset with one thousand sections that portray the attributes of clients; in any case, if the information in a portion of the segments is extremely inadequate that would increase almost no

advantage from adding them to the model, and if a portion of the segments copy one another, utilizing the two segments could influence the model.

For feature selection in the research apply two different algorithms such as Chi-square and Correlation Coefficient algorithm.

3.3.6.1 Chi Square

The Chi-Square test of freedom is utilized to decide whether there is a significant connection between two categorical variables. The recurrence of every classification for one ostensible variable is looked at over the classifications of the second ostensible variable. The information can be shown in a possibility table where each line speaks to a class for one factor and every segment speaks to a classification for the other variable. For instance, say a scientist needs to look at the connection between sex (male versus female) and compassion (high versus low). The chi-square trial of autonomy can be utilized to look at this relationship. The invalid speculation for this test is that there is no connection among sexual orientation and sympathy. The elective speculation is that there is a connection among sex and compassion. If O is the observed value and E is the expected value then the chi-square is,

$$X^2 = \Sigma \frac{(O - E^2)}{E} \tag{3.3}$$

In this thesis a library named scikit-learn has been used for calculating the K best feature as per their score indicating the feature importance. Before removing feature I've total 49 feature excluding target class. After removing unnecessary features finally I've got total 25 features. Following is the horizontal bar chart representation of feature scores in chi square:

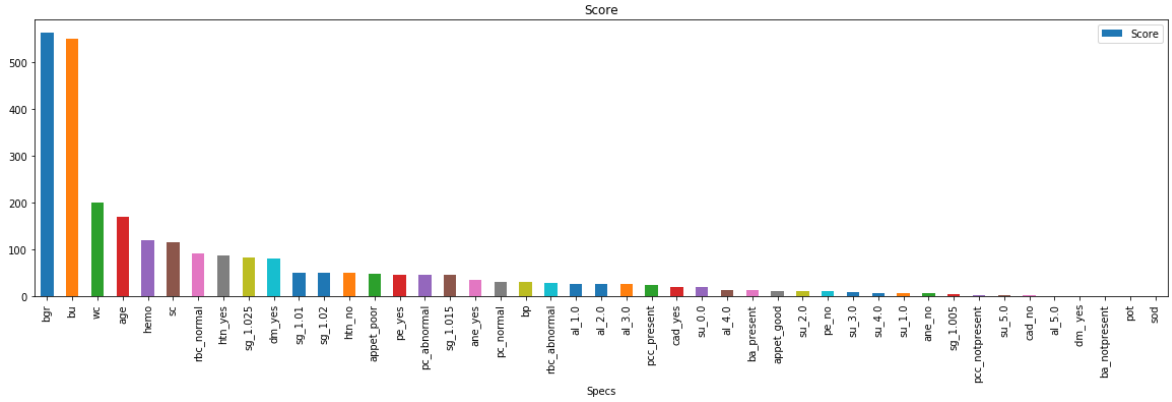

Figure 3. 5 Feature Score in Chi Square Test

3.3.6.2 Correlation

Correlation is a statistical method that measures and break down the level of connection between two variables. Correlation investigation manages the relationship between at least two variables. Correlation signifies the interdependency among the factors for corresponding two wonder, it is basic that the two marvels ought to have cause-impact relationship, & if such relationship does not exist then the two marvels cannot be associated. On the off chance that two factors shift so that development in one are joined by development in other, these factors are called circumstances and end results relationship. The correlation coefficient, r, is an outline measure that portrays the degree of the factual connection between two interim or proportion level variables. The correlation coefficient is scaled with the goal that it is dependably between - 1 and +1. At the point when r is near 0 this implies there is little correlation between the factors and the more distant far from 0 r is, in either the positive. The degree of relationship between the variables under consideration is measure through the correlation analysis. If X is an attribute whereas Y is a target attribute, then the correlation of X & Y is

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (3.4)$$

.

Where N is the total number of records. The correlation esteem is should between - 1 to +1. On the off chance that the connection is lower than 0 and close to - 1 then X and Y are negative related. In the event that connection esteem is close to positive 1 at that

point there is a positive connection between X and Y. In the event that the correlation is 0, at that point there is no connection between X and Y. Following is the correlation heatmap generated using seaborn library of python. From the heatmap we can see that cell having deeper red color indicating extreme positive correlation. On the other hand cell having deeper blue color indicating extreme negative correlation.

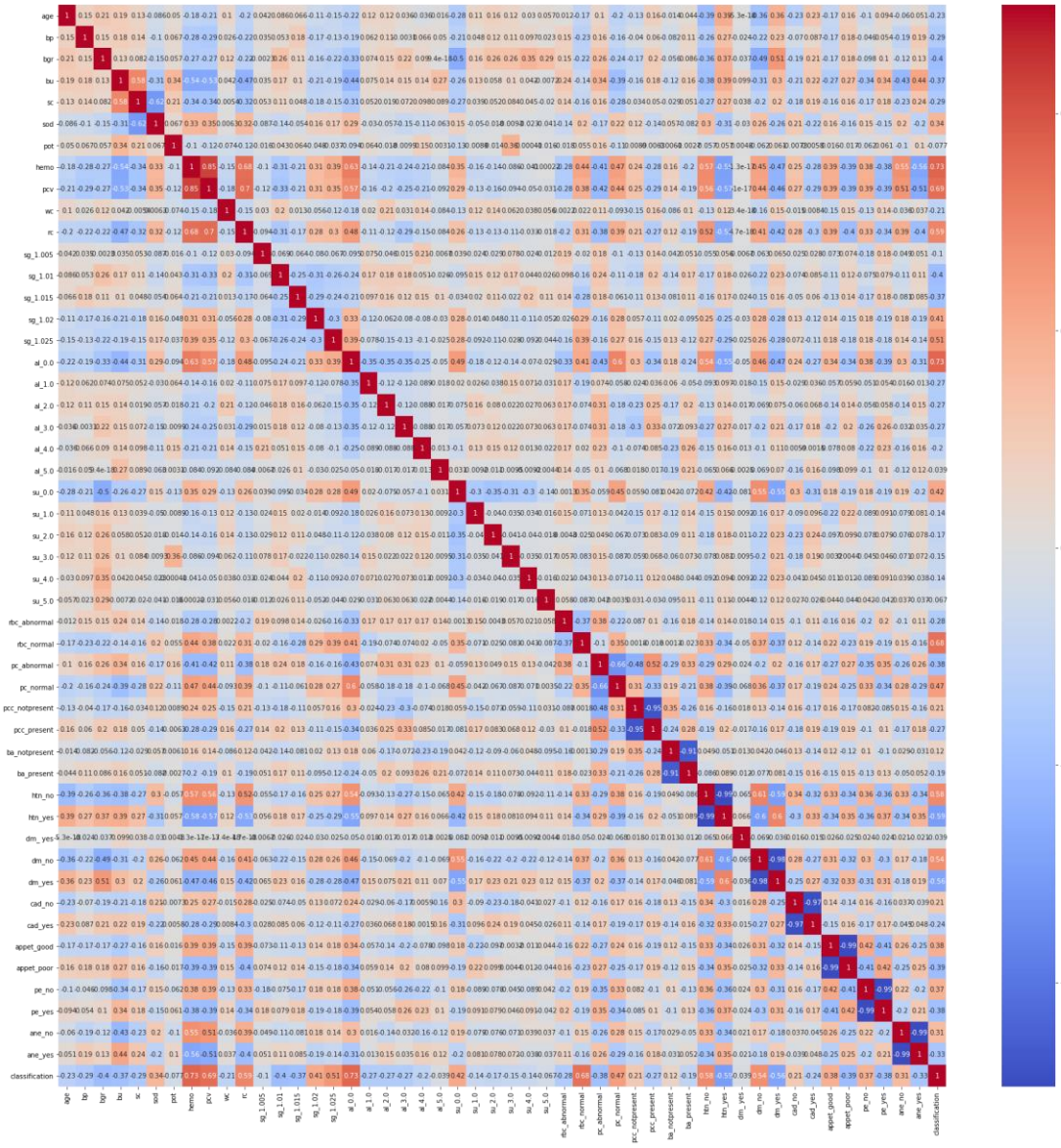

Figure 3. 6 Correlation Heat Map

3.3.7 Feature Scaling

Feature scaling is a method which is used to normalize the range of independent variables of data. In real word dataset there are always varying information. Suppose in

a dataset there is a column named age which varies from 1 to 120. On the other hand there is another column named upload_size which varies from 2000 to 2000000. So clearly we can see that there is a variation in magnitude, unit and range. As of now many machine learning algorithms use Euclidian distance between two data point for computation this variance will result in huge problem. There are many ways for feature scaling. Some of them are as follows:

- Rescaling (Min-Max Normalization)

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.5}$$

- Mean Normalization

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)} \tag{3.6}$$

- Standardization

$$x' = \frac{x - \overline{x}}{\sigma} \tag{3.7}$$

- Scaling to unit length

$$x' = \frac{x}{||x||} \tag{3.8}$$

In this research Min-Max Normalization technique is used for feature scaling. Following is the box plot image before feature scaling.

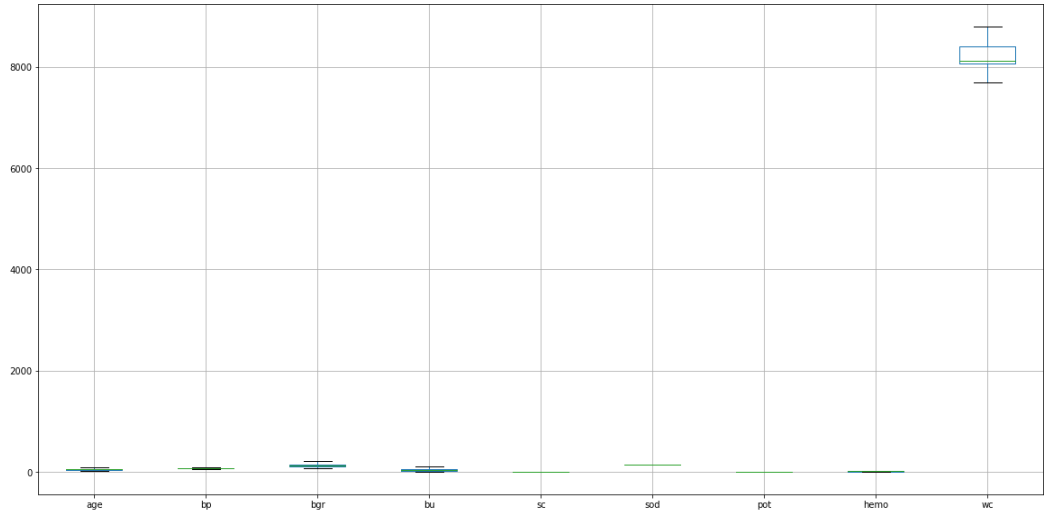

Figure 3. 7 Dataset without magnitude scaling

After scaling the dataset magnitude using min-max scaler following figure is plotted:

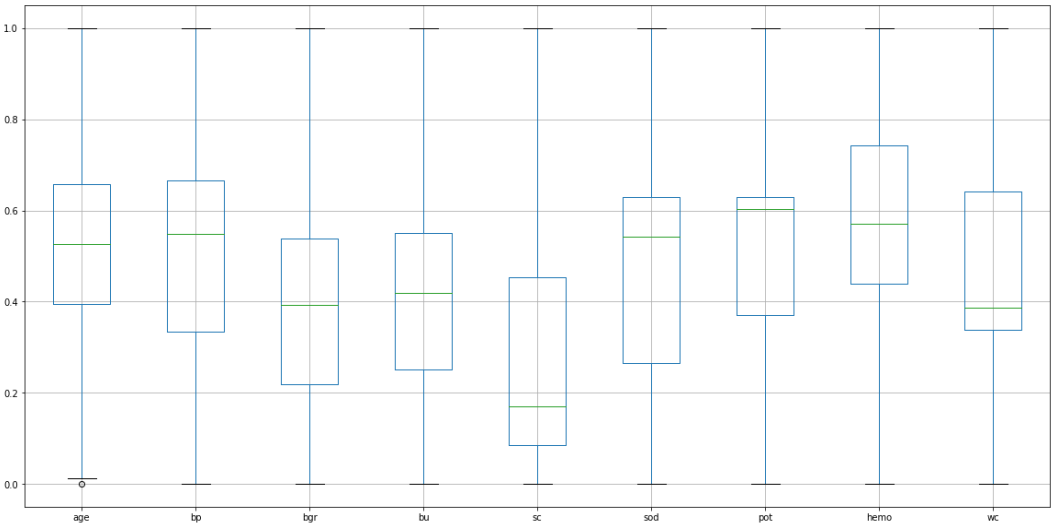

Figure 3. 8 Scaling the magnitude of dataset

3.4 Classification

In machine learning and statistics, classification is the puzzle of recognizing to which of a set of categories (sub-populations) a new observation belongs, on the basis of a

training set of data containing observations (or instances) whose category membership is known.

3.4.1 Naïve Bayes

It is one of the fastest statistical classifier algorithm works on probability of all attribute contained in data sample individually and then classifies them accurately. It works on conditional probabilities and class probabilities (John, et al. 1995).

- Class Probability: - This is the probability of each class in the training dataset (Zhang, et al. 2004).
- Conditional Probabilities: - Probabilities of each input values when the given hypothesis is true for each input (Zhang, et al. 2004).

$$p(C_k|X) = \frac{p(C_k)p(X|C_k)}{p(X)} \quad (3.9)$$

The problem statement is to find the max probability of hypothesis for given data.

$$MAX(p(C_k|X)) \quad (3.10)$$

The class probabilities is calculated by dividing the frequency of instances that belong to given class by the total number of class instances (Zhang, et al. 2004).

$$P(class = 1) = \frac{count(class = 1)}{count(class = 1) + count(class = 0)} \quad (3.11)$$

Conditional probability is calculated using this formula as follows.

$$P(X = V|C_k = 1) = count(X = V\ and\ C_k = 1)/count(C_k = 1) \quad (3.12)$$

3.4.2 Random Forest

Random Forest (RF), proposed by Leo Breiman, is fast, highly accurate, noise resistant classification method. Bagging and random feature selection are combined together.

Every tree in the forest is influenced by the values of random vectors sampled separately and has identical distribution as any other tree in the forest (L. Breiman, et al. 2001).

RF consists of outsized number of decision trees where decision tree select their separating features from bootstrap training set S_i where i represent ith internal node. Trees in RF are grown by means of Classification and Regression Tree (CART) method with no pruning. As number of trees in the forest turns into outsized number, generalization error will also increase until it converges to some boundary level (Subasi, et al. 2017).

3.4.3 K-Nearest Neighbor

The k-nearest neighbor (L. E. Peterson, et al. 1883, 2009) algorithm uses the data directly for classification without building a model first. As such, no details of model construction need to be considered, and the only adjustable parameter in the model is k, the number of nearest neighbors to include in the estimate of class membership: the value of $P(y|x)$ is calculated simply as the ratio of members of class y among the k-nearest neighbors of x. By varying k, the model can be made more or less flexible. The advantage of the k-nearest neighbor classifier is, it is robust to noisy training data and effective with large training datasets. The major drawback lies in the calculation of the case neighborhood: for this, one needs to define a metric that measures the distance between data items. In most cases it is done by trial and error (Salekin, et al. 2016).

3.4.4 Support Vector Machine

Support Vector Machines (SVMs) are a set of supervised learning techniques, and can be used to perform both classification and regression tasks. SVM technique builds a maximum- margin hyper-plane that is positioned in transformed input space and divides the pattern classes, while the distance to the closest plainly divided patterns is maximum. The solution hyper-plane parameters are obtained from a quadratic programming optimization task. SVM gained huge reputation because of its firm theoretical foundation since it was introduced by Vapnik (V. Vapnik, et al. 1995) in 1995 (S. Armin, et al. 2010).

A hyper-plane can separate the training set. In (V. Vapnik, et al. 1995), it is shown that for the class of hyper-planes, the complexity of the hyper-plane can be constrained by an additional measure, the margin. The margin can be understood as the minimal distance between a pattern and a decision surface. Therefore, complexity can be controlled in the case that we constraint the margin of a function class from below. Support vector learning performs this understanding that when margin is maximized, then risk is minimized. A SVM chooses a maximum-margin hyper-plane that is found in a transformed input space and divides the pattern classes, whereas trying to obtain the maximum distance to the nearest plainly divided pattern. The parameters are computed from a quadratic programming optimization task (S. Armin, et al. 2010).

By employing the kernel trick for SVM, it is possible that the maximum margin hyper-plane fits in a feature space S. The feature space S represents a non-linear mapping $\Phi: R^N \to S$ from the initial input space, commonly of far more bigger dimensionality compared to the initial input space. By using the kernel trick, non-linear SVM can create the maximum margin hyper-plane to fit in a feature space (S. Armin, et al. 2010).

3.4.5 Logistic Regression

Logistic regression examines the relationship between a binary outcome (dependent) variable such as presence or absence of disease and predictor (explanatory or independent) variables such as patient demographics or imaging findings (Hosmer, et al. 1989).

For example, the presence or absence of chronic kidney disease within a specified time period might be predicted from knowledge of the patient's age, albumin, specific gravity, and any prior diabetes mellitus. The outcome variables can be both continuous and categorical. If $X_1, X_2, \dots, X_n$ denote n predictor variables (e.g., calcification types, specific gravity, patient age, and so on), Y denotes the presence $(Y = 1)$ or absence $Y = 0$ of disease, and p denotes the probability of disease presence (i.e., the probability that $Y = 1$), the following equation describes the relationship between the predictor variables and p :

$$Log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n, \tag{3.13}$$

Where $\beta_0$is a constant and $\beta_1, \beta_2, \dots, \beta_n$ are the regression coefficients of the predictor variables $X_1, X_2, \dots, X_n$. The regression coefficients are estimated from the available data. The probability of disease presence p can be estimated with this equation. Each regression coefficient describes the size of the contribution of the corresponding predictor variable to the outcome. The effect of the predictor variables on the outcome variable is commonly measured by using the odds ratio of the predictor variable, which represents the factor by which the odds of an outcome change for a one-unit change in the predictor variable. The odds ratio is estimated by taking the exponential of the coefficient (e.g. $[\beta_1]$). For example if β_1 is the coefficient of variable X_{SG} ("specific gravity"), and p represents the probability of chronic kidney disease, (β_1) is the odds ratio corresponding to the any prior diabetes mellitus. The odds ratio in this case represents the factor by which the odds of having chronic kidney disease increase if the patient has any prior diabetes mellitus and all other predictor variables remain unchanged.

Logistic regression models generally include only the variables that are considered "important" in predicting an outcome. With use of *P* values, the importance of variables is defined in terms of the statistical significance of the coefficients for the variables. The significance criterion $P \leq 0.5$ is commonly used when testing for the statistical significance of variables; however, such criteria can vary depending on the amount of available data. For example, if the number of observations is very large, predictors with small effects on the outcome can also become significant. To avoid exaggerating the significance of these predictors, a more stringent criterion (e.g.,$P \leq .001$) can be used. Significant variables can be selected with various methods. In forward selection, variables are sequentially added to an "empty" model (i.e., a model with no predictor variables) if they are found to be statistically significant in predicting an outcome. In contrast, backward selection starts with all of the variables in the model, and the variables are removed one by one as they are found to be insignificant in predicting the outcome. The stepwise logistic regression method is a combination of these two methods and is used to determine which variables to add to or drop from the model in a sequential fashion on the basis of statistical criteria. Although different techniques can yield different regression models, they generally work similarly. Sometimes, clinically important variables may be found to be statistically insignificant with the selection methods because their influence may be attenuated by the presence of other

strong predictors. In such cases, these clinically important variables can still be included in the model irrespective of their level of statistical significance.

3.4.6 Decision Tree

Decision trees are an exceptionally compelling strategy for supervised learning. Its points are the partition of a dataset into bunches as homogeneous as conceivable as far as the variable to be predicted. It takes as info an arrangement of characterized information, and yields a tree that looks like to an introduction outline where each node (leaf) is a decision (a class) and each non-last node (inward) represents to a test. Each leaf present to the decision of having a place with a class of information checking all tests way from the root to the leaf.

The tree is more straightforward, and in fact it appears to be anything but difficult to utilize. In truth, it is all the more fascinating to get a tree that is adjusted to the probabilities of factors to be tried. Generally adjusted tree will be a decent outcome. In the event that a sub-tree can just prompt a special arrangement, at that point all sub-tree can be lessened to the straightforward. Ross Quinlan (Quinlan J. R., et al. 1985) first introduced the decision tree and its features in 1985. Ross Quinlan (Quinlan J. R., et al. 1985) summarizes in his paper, an approach to synthesizing decision trees that has been used in a variety of systems and also describes one such system that is familiar named ID3. But ID3 has some limitations. Maimon (Maimon & Rokach, et al. 2010) said that ID3 does not apply any pruning procedures or does not handle numeric values or missing values. Another limitation describes by Hssina (Hssina, et al. 2014), that ID3 is overly sensitive to features with large numbers of values. ID3 algorithm create tree based on Information Gain (IG) that's are get from training instances and apply to classify the test data.

Ross Quinlan, introduced another algorithm named C4.5 that is an evolution of ID3 (Quinlan J. R., et al. 2014). ID3's limitation is overcome in C4.5 algorithm. The C4.5 follow the same approach of ID3 but includes some additional features that makes C4.5 more powerful than ID3. C4.5 algorithm can handle missing and numeric values by using gain ratio. Gain ratio, is defined as follows (Hssina, et al. 2014):

$$GainRation(p,T) = \frac{InformationGain(p,T)}{SplitInfo(p,T)} \quad (3.14)$$

Where SplitInfo is,

$$SplitInfo(p,test) = -\sum_{j=1}^{n} \mathrm{P}\left(\frac{j}{p}\right) * \log(P(\frac{j}{p})) \quad (3.15)$$

Where P' (j/p) is the proportion of elements present at the position, taking the value of j-th test.

Advantages of decision trees (Prajwala, et al. 2015) are such as Simple to interpret the decision tenets, Nonparametric so it is anything but difficult to consolidate a scope of numeric or clear-cut information layers and there is no compelling reason to choose unimodal preparing information and Powerful with respect to exceptions in preparing data.

Disadvantage of decision tress (Prajwala, et al. 2015) such as decision trees tend to over fit preparing information which can give poor outcomes when connected to the full dataset and Impractical to anticipate past the base and most extreme breaking points of the reaction variable in the preparation information.

3.5 Hyper Parameter Tuning

Every machine learning algorithm has a set of variable which hold values from various source. Sometimes they are filled by user input, sometimes they are learned by data. These variables are called parameter. Parameter whose values are learned from data known as parameter. But there are some set of parameter whose values are set before the learning process start. These are called hyper parameter. In scikit-learn machine learning library in python hyper parameter is passed in as the argument in model constructor class.

In this research model's hyper parameter is tuned as follows:

- Random Forest Classifier :- random_state=42, n_estimator=1
- K-Nearest Neighbor :- n_neighbors=5
- Decision Tree Classifier :- criterion = "gini",random_state = 42,max_depth=1, min_samples_leaf=5

3.6 K-Fold Cross Validation

Cross validation is a statistical procedure which is used to calculate the expertise of machine learning models. Here the word K means the total number of fold will be used for cross validation. In this process the whole dataset is shuffled randomly. Then they are split into K sub set. Splitting is done is such way that all subset have same ratio of target value. After splitting:

- For each unique subset
    - Take this sub-set as test set
    - Take rest subset as train set
    - Fit a model on training set and evaluate the model performance on test set
    - Store the evaluation score and discard the model

Continue the process until there is any unique subset not yet used for as test set as well as train set.

In this research 10-Fold cross valuation is used for all model stated in section 3.4.

3.7 Ensemble Learning

Ensemble method combines several machine learning models to produce better model. In most of the case single model's performance can be biased. Sometimes a single model is biased on some types of data or on some feature. Sometimes single model's variance can be large due to data variability in data and model's greedy approach towards variability. To avoid these issues we need ensemble learning. Ensemble learning controls bias and variance. It produce better accuracy and evaluation score. Actually it combines many weak learners to form a strong learner. There are a few types if ensemble learners. They are as follows:

3.7.1 Bagging

Bagging stands for bootstrap aggregation. Here bootstrap means sampling with replacement. Working procedure of bagging is as follows:

- A ransom sample from a dataset is picked with replacement.
- Here replacement means in many sample there may be repetition of many observation.
- A subset of feature is picked randomly.
- A model is created and fitted with random subsample and random feature set.
- This procedure is repeated and created many models
- Finally every model's prediction is aggregated as final prediction.

$$f(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x) \tag{3.16}$$

3.7.2 Boosting

Boosting uses weighted average to make a weak learner into a strong learner. At first all the record of a dataset is weighted equally. Then they are used for taring a model. Then the model is being evaluated. There must be some observation point where model will not be able to correctly generalize. The rest of the procedure is as follows:

- A higher weight is added to the observation which are incorrectly classified.
- Again the model is being trained to reduce the incorrectness of highly weighted records.
- This process continues until the model generalize almost all the record correctly.

Thus the final model is enough complex to generalize almost all the records.

In this research AdaBoost technology is used for boosting. **Figure 3.3** is the algorithm for AdaBoost (Vineet Maheshwari, 2019):

Table 3. 3 Algorithm for AdaBoost

Init data weights $\{w_n\}$ to $1/N$
for $m = 1$ to M do
fit a classifier $y_m(x)$ by minimizing weighted error function J_m:
$J_m = \sum_{n=1}^{N} w_n^{(m)} 1[y_m(x_n) \neq t_n]$
compute $\epsilon_m = \sum_{n=1}^{N} w_n^{(m)} 1[y_m(x_n) \neq t_n] / \sum_{n=1}^{N} w_n^{(m)}$
compute $\alpha_m = \log\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$
update the data weights: $w_n^{m+1} = w_n^{(m)} \exp\{\alpha_m 1[y_m(x_n) \neq t_n]\}$
end for
Make prediction using final model: $Y_M(x) = sign(\sum_{m=1}^{M} \alpha_m y_m(x))$

3.7.3 Stacking

Stacking is an ensemble learning technique where multiple classifier or regression models are combined with the help of a Meta classifier or regression model. The base level models are trained based on a complete training set, then the meta-model is trained on the outputs of the base level model as features. **Figure 3.4** is the algorithm for stacking (Vineet Maheshwari, 2019):

Table 3. 4 Algorithm for Stacking

Input: training data $D = \{x_i, y_i\}_{i=1}^{m}$
Output: ensemble classifier H
Step 1: learn base level classifiers
for $t = 1$ to T do
learn h_t based on D
end for
Step 2: construct new dataset of predictions
for $i = 1$ to m do
$D_h = \{x_i', y_i\}$ where $x_i' = \{h_1(x_i), \dots, h_T(x_i)\}$
end for
Step 3: learn a meta classifier
learn H based on D_h
return H

3.8 Performance Evaluation Metrics

After finishing data preprocessing task dataset has been split as train and test subset for training the machine learning model and then testing the performance of the model. Here the term performance metrics comes. Performance evaluation metrics refers to the metrics used for evaluating the model's performance. There are a large number of evaluation metrics are available to evaluate and compare the performance of Supervised Machine learning models (Powers, et al. 2010).

3.8.1 Accuracy

Accuracy in classification means what it actually is meant by the word; how accurately the classifier can predict the target variable's value. It is calculated from ratio of all correct prediction and all prediction.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ prediction\ made} \qquad (3.17)$$

But as we know that a specific performance metrics is not sufficient to evaluate a model. The classifier accuracy will work better if there are equal number of instance of each target class. Luckily in this dataset almost there are equal number of each class available. Here is the evidence for equality.

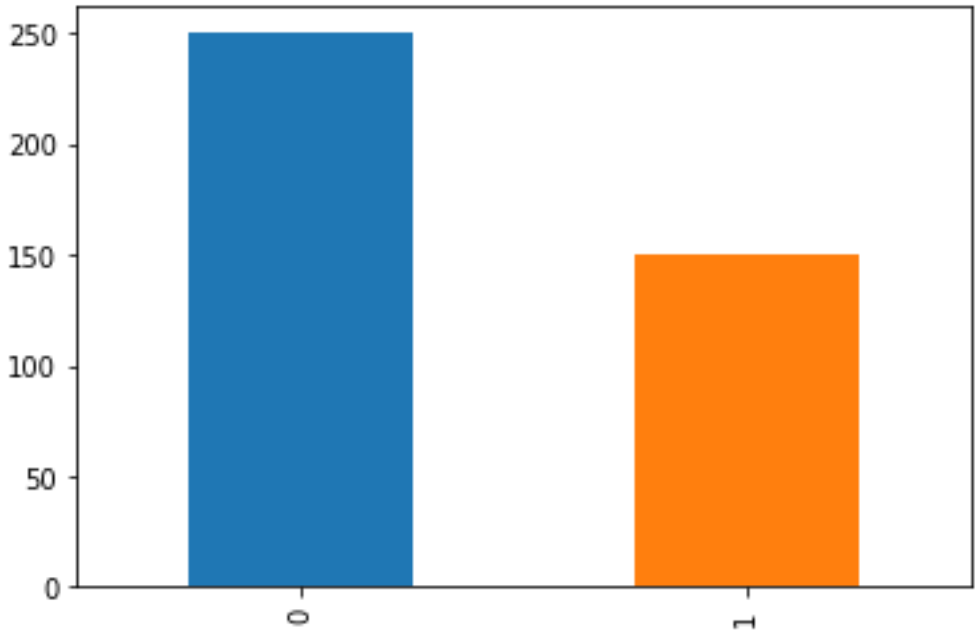

Figure 3. 9 Total number of records for each class

Here 0 refers to positive result which is ckd and 1 refers to the negative result which is notckd.

Suppose we have a training set where 98% of samples are of class A and rest 2% of samples are of class B. Here our model can get up to 98% of trading accuracy by simply labeling each sample as of class A.

But when we test the model using our test set where 60% of the sample is of class A and rest 40% of sample is of class B then the test accuracy will decreased to 60%. That's why classification accuracy is not always the best performance metrics.

3.8.2 Confusion Matrix

Confusion matrix is table-alike representation which shows the complete and detailed performance report of a model. This is a table having two dimension ("Actual", "Predicted").

Table 3. 5 Confusion Matrix Details

		Actual	
		Positive(1)	Negative(0)
Predicted	Positive(1)	**TP**	**FP**
	Negative(0)	**FN**	**TN**

- **TP**: - True positive. The case where all the actual data points are positive and predicted results are also positive.
- **FP**: - False positive. The case where actual data points are negative but predicted as positive.
- **FN**: - False negative. Here actually all data points are positive but they are predicted as negative.
- **TN**: - True negative. Here actually all data points are negative and also predicted as negative.

So the formula of accuracy is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.18)$$

3.8.3 Precision

It means among all the record classified as positive how many of them are actually positive. This is the ratio of all predicted positive record which is TP+FP over all actually positive record which is TP. Precision is all about being precise to the result.

Thus if the model manage to capture only 1 positive case correctly then the model is 100% precise.

$$Precision = \frac{TP}{TP + FP} \tag{3.19}$$

3.8.4 Recall/Sensitivity

It means among all the positive record how many of them are predicted as positive. This is the ratio of all positive record which is TP+FN over all actually positive record which is TP. Recall is not the case capturing the record correctly but capturing all the positive case as positive.

$$Recall = \frac{TP}{TP + FN} \tag{3.20}$$

3.8.5 Specificity

It means among all the negative record how many of them are predicted as negative. This is the complete opposite of recall. Specificity is all about capturing all the negative case as negative.

$$Specificity = 1 - Recall \tag{3.21}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.22}$$

3.8.6 F1-Score

This is the harmonic mean between precision and recall. Range for f1-score is from 0 to 1. It describe the preciseness (how many record can be correctly classified by the model) and robustness (it avoid missing any significant number of record) of a model.

High precision and low recall produce am extremely accurate model but still there is a large number of instances which are difficult to classify. F1-Score positively accelerate the model's performance. Mathematical expression for f1-score is as follows:

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{3.23}$$

3.8.7 AUC-ROC Curve

AUC is only used for binary classification problem. It represents the degrees of separate ability under a set of calibrated threshold. Thus how well the model is able to discriminate between positive and negative result. The value of AUC 1 means the model can perfectly distinguish between positive and negative result. The worst score for AUC is 0.5 which means complete inability to separate. ROC curve is plotted in TPR (y-axis) against FPR (x-axis) for a number of different candidate threshold values between 0.0 and 1.0. ROC curve is appropriate choice where the distribution of positive and negative classes are distributed equally. Otherwise the precision-recall curve. For a ROC curve the area under the curve is the summary for the model performance.

CHAPTER 4

RESULTS AND DISCUSSION

In this thesis we have performed total 4 experiment from 4 different standpoint. This section is fulfilled with the result of those 4 experiments. Those 4 experiments are done using all features, all features and reduced features.

4.1 Experiment 1

Here in this experiment all the features have been used. Total feature count is 49 excluding target class. I've applied different machine learning algorithms and different validation techniques on the dataset. As per the proposed model I've applied total 6 machine learning algorithms on preprocessed dataset. I've tried these models on different criteria. I've applied these models in following criteria:

- Single learner
- 10 Fold Cross validation
- Ensemble learners
    - Bagging
    - Boosting
    - Stacking

4.1.1 Single Learner

Table 4. 1 Performance Report for Single Data Fold Learner

Metrics	NB		RF		KNN		DT		SVM		LR	
Accuracy	0.9848		0.9621		0.9772		0.9469		0.9772		0.9772	
Precision	0.98		0.96		0.98		0.95		0.98		0.98	
Recall	0.98		0.96		0.98		0.95		0.98		0.98	
F1-Score	0.98		0.96		0.98		0.95		0.98		0.98	
AUC Score	0.984		0.948		0.999		0.958		0.995		0.999	
Confusion Matrix	47	1	43	5	47	1	48	0	46	2	46	2
	1	83	0	84	2	82	7	77	1	83	1	83

Here **Naïve Bayes has performed as best in terms of accuracy, precision, recall, f1-score, and AUC**. It also has got very good FP and FN count. Though Random Forest has got best FN count and Decision Tree has got best FP count.

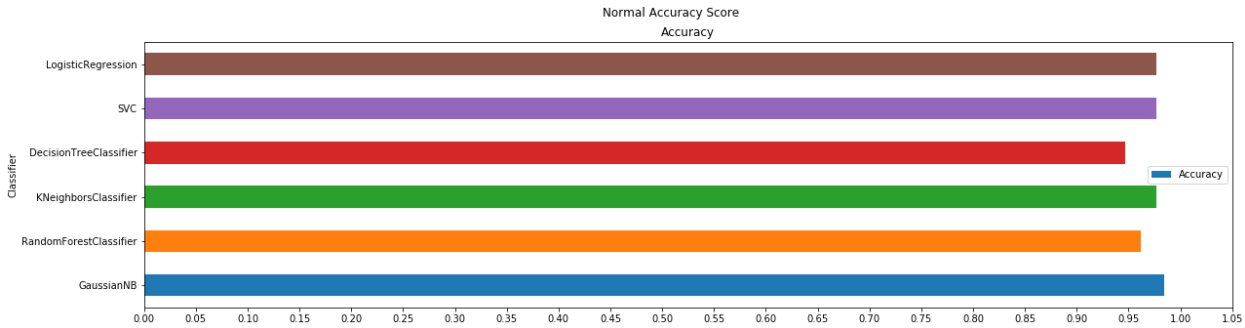

Figure 4. 1 Normal Accuracy Score Chart

Following is the ROC curve for all models as single learner.

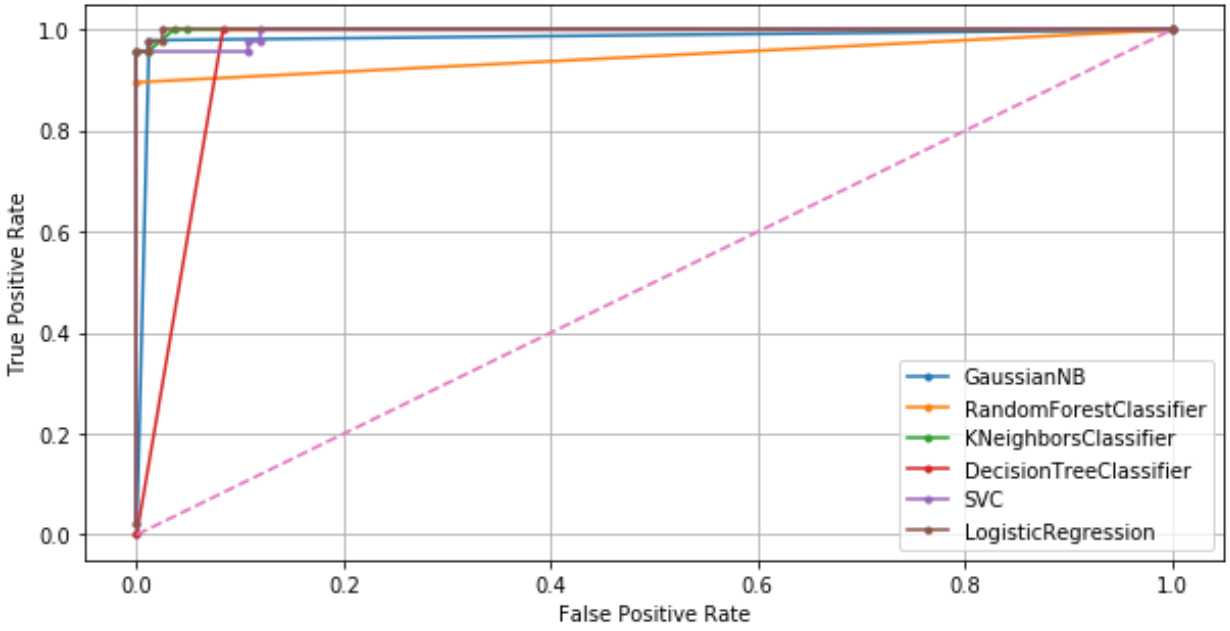

Figure 4. 2 ROC Curve

As per the Curve We can see that almost all learners have covered most of the area under the curve. In terms of AUC score K-Nearest Neighbor has scored best as 0.999.

Following are the learning curves for all the models.

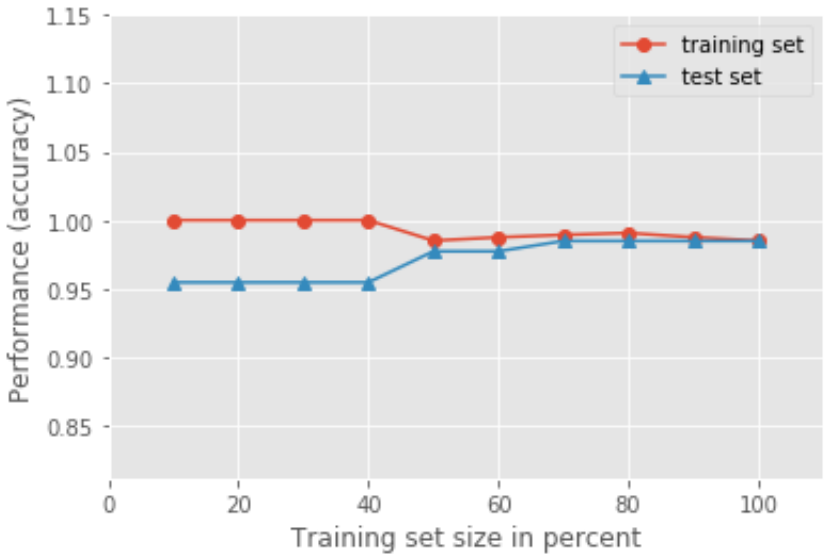

Figure 4. 3 Learning Curve Naive Bayes

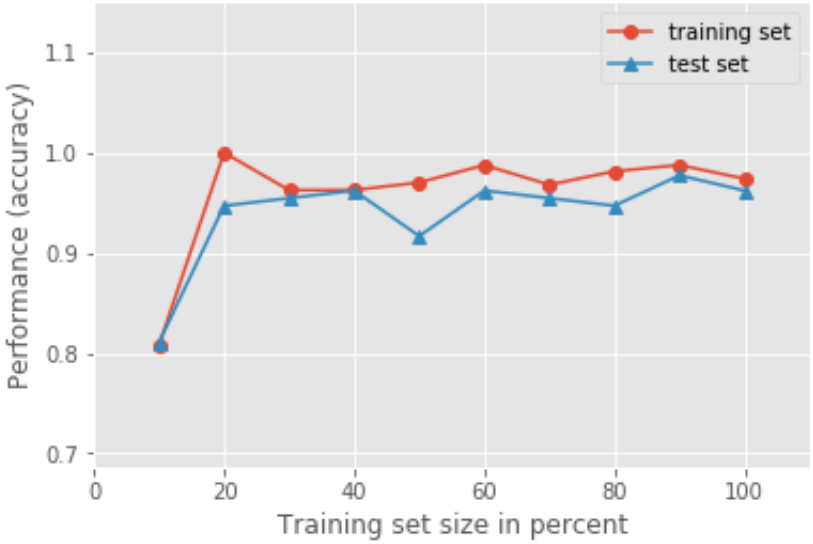

Figure 4. 4 Learning Curve Random Forest

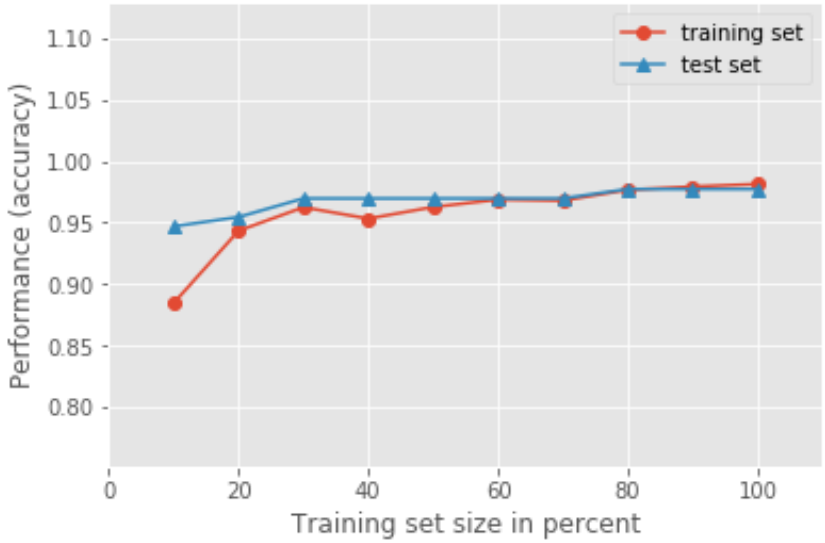

Figure 4. 5 Learning Curve K-Nearest Neighbor

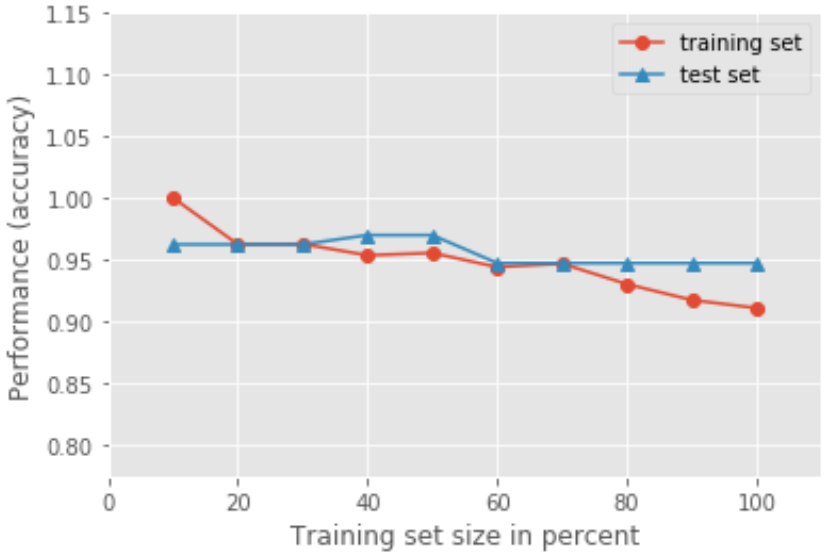

Figure 4. 6 Learning Curve Decision Tree

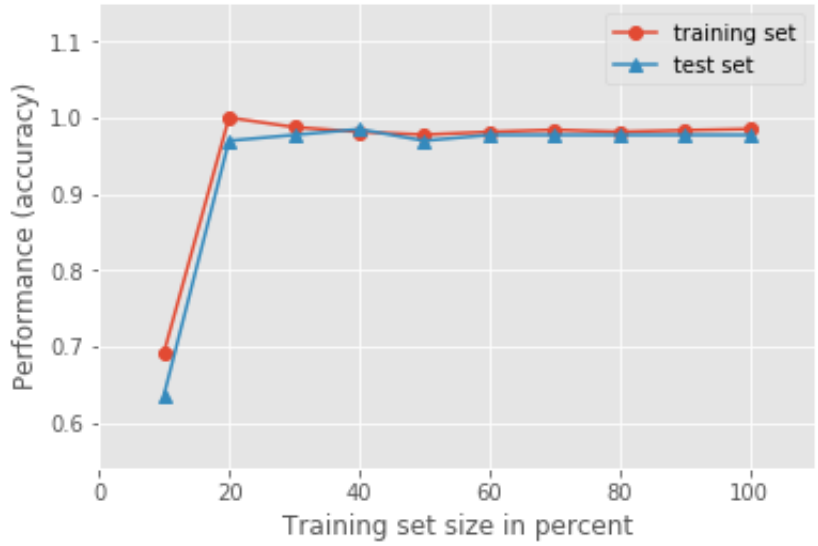

Figure 4. 7 Learning Curve Support Vector Machine

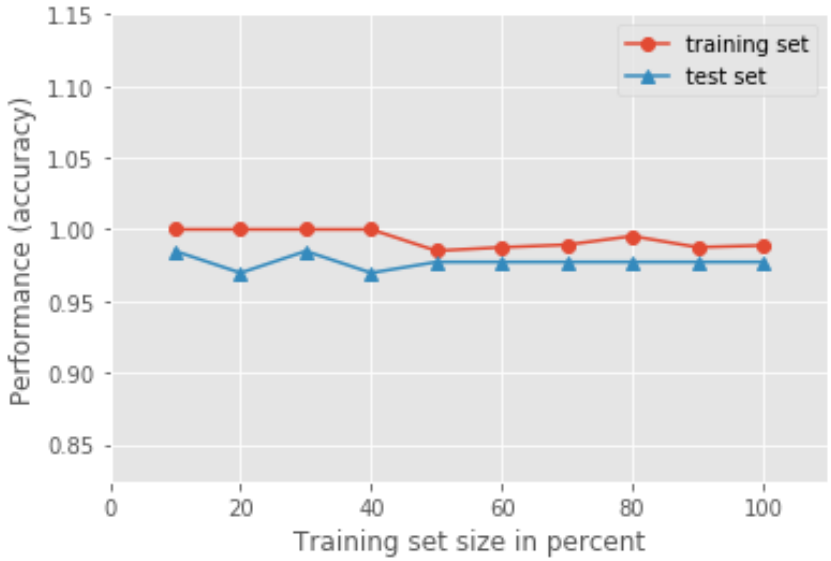

Figure 4. 8 Learning Curve Logistic Regression

From the learning curve we can see that Naïve Bayes has converged at 90% of dataset. Random Forest has converged at almost 98% of data. K-Nearest Neighbor has converged at 60% of data. Decision tree has failed to converge. Support Vector Machine has converged at 35% of the data. Logistic Regression has failed to converge.

4.1.2 10 Fold Cross Validation

Here I've applied a 10 fold cross validation on each learner to measure their performance. Following is the result:

Table 4. 2 Performance Report for Cross Validation Learner

Metrics	NB	RF	KNN	DT	SVM	LR
Accuracy	0.980	0.935	0.970	0.917	0.977	0.985
Precision	0.98	0.94	0.97	0.92	0.98	0.98
Recall	0.98	0.94	0.97	0.92	0.98	0.98
F1-Score	0.98	0.94	0.97	0.92	0.98	0.98
Confusion Matrix	147 3 5 245	137 13 13 237	150 0 12 238	140 10 23 227	146 4 2 245	147 3 3 247

From the table we can see that **Naïve Bayes has performed as best in terms of accuracy, precision, recall and f1-score**. From the confusion matrix we can see that K-Nearest Neighbor has the lowest FP identification and Support Vector Machine has the lowest FN identification.

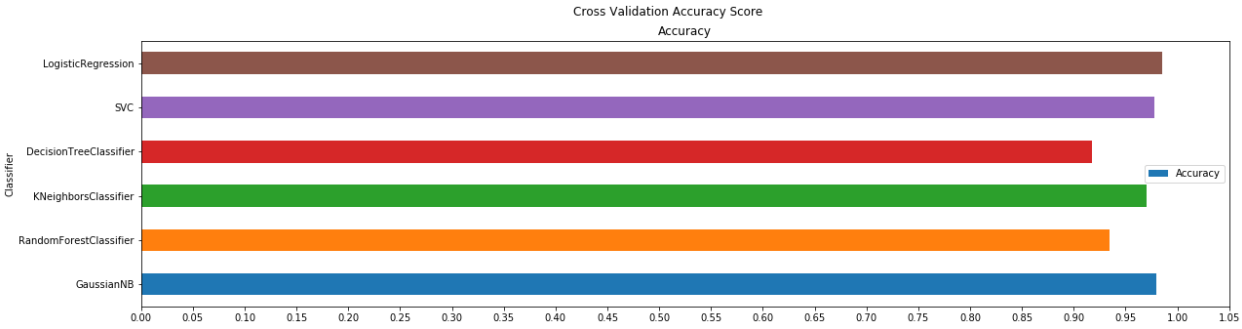

Figure 4. 9 Cross Validation Accuracy Score Chart

4.1.3 Bagging (Ensemble)

We have applied bootstrap aggregation ensemble learning technique on the models we have chosen. We used a 5 fold cross validation and 50% max sample and 50% max feature to train the model. Following is the result:

Table 4. 3 Performance Report for Bagging Classifier

Metrics	NB	RF	KNN	DT	SVM	LR
Accuracy	0.960	0.985	0.970	0.930	0.980	0.972
Precision	0.96	0.99	0.97	0.94	0.98	0.97
Recall	0.96	0.98	0.97	0.93	0.98	0.97
F1-Score	0.96	0.98	0.97	0.93	0.98	0.97
Confusion Matrix	145 5 11 239	144 6 0 250	146 4 8 242	145 5 23 227	145 5 3 247	146 4 7 243

After applying bagging ensemble learning method **Random Forest** have got improvement. It has performed as best with the highest accuracy of 0.985 including highest precision of 0.99. Both random forest and support vector machine has same highest recall and f1-score of 0.98. Random Forest has lowest FN and K-Nearest Neighbor has the lowest FP.

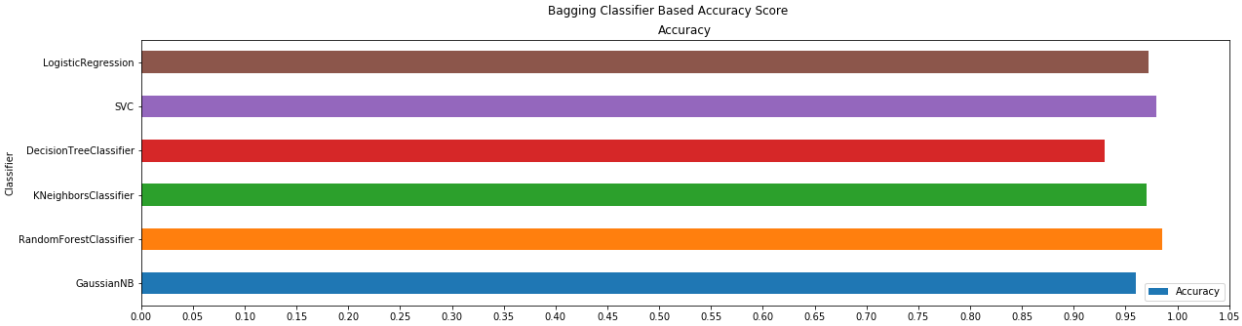

Figure 4. 10 Bagging Accuracy Score Chart

4.1.4 Boosting (Ensemble)

We have applied boosting ensemble learning technique on every model except K-Nearest Neighbor as this model does not support weighted sampling which is the fundamental of boosting. We have used learning rate of 1 and 10 estimator which 5 fold cross validation. Following is the result:

Table 4. 4 Performance Report for boosting leaners

Metrics	NB		RF		KNN		DT		SVM	
Accuracy	0.970		0.972		0.967		0.977		0.970	
Precision	0.97		0.97		0.97		0.98		0.97	
Recall	0.97		0.97		0.97		0.98		0.97	
F1-Score	0.97		0.97		0.97		0.98		0.97	
Confusion Matrix	142	8	144	6	144	6	146	4	146	4
	4	246	4	245	7	243	5	245	4	242

After applying boosting ensemble learning technique we can see that **Support Vector Machine has performed as best in terms of accuracy, precision, recall and f1-score**. It has the lowest FP.

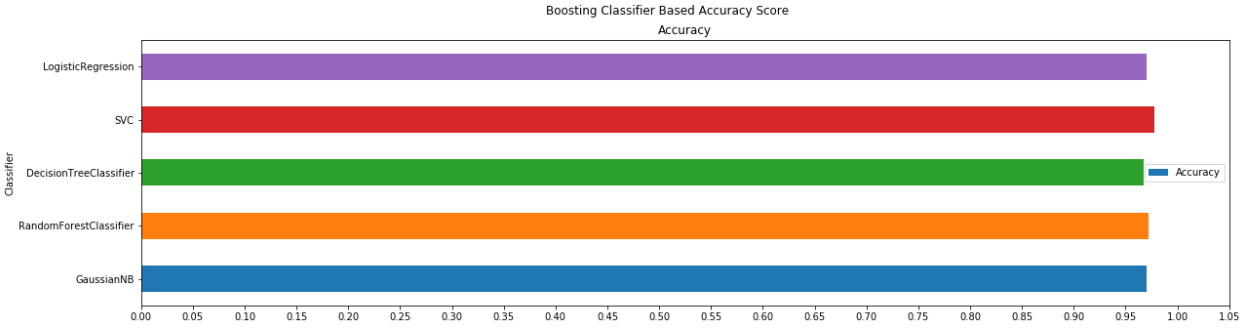

Figure 4. 11 Boosting Accuracy Score Chart

4.1.5 Stacking W/O Cross Validation (Ensemble)

Table 4. 5 Performance Report for stacking

Metrics	Meta: NB Base: KNN, DT		Meta: RF Base: NB, KNN		Meta: KNN Base: NB, DT		Meta: DT Base: SVM, LR		Meta: SVM Base: DT, LR		Meta: LR Base: SVM, NB	
Accuracy	0.9772		0.9772		0.9848		0.9772		0.9772		0.9696	
Precision	0.98		0.98		0.98		0.98		0.98		0.97	
Recall	0.98		0.98		0.98		0.98		0.98		0.97	
F1-Score	0.98		0.98		0.98		0.98		0.98		0.97	
AUC Score	0.989		0.973		0.989		0.973		0.986		0.994	
Confusion Matrix	47	1	46	2	47	1	46	2	46	2	45	3
	2	82	1	83	1	83	1	83	1	83	1	83

Here **K-Nearest Neighbor** has performed best in terms of every metrics.

Following is the ROC curve for stacking based model without cross validation.

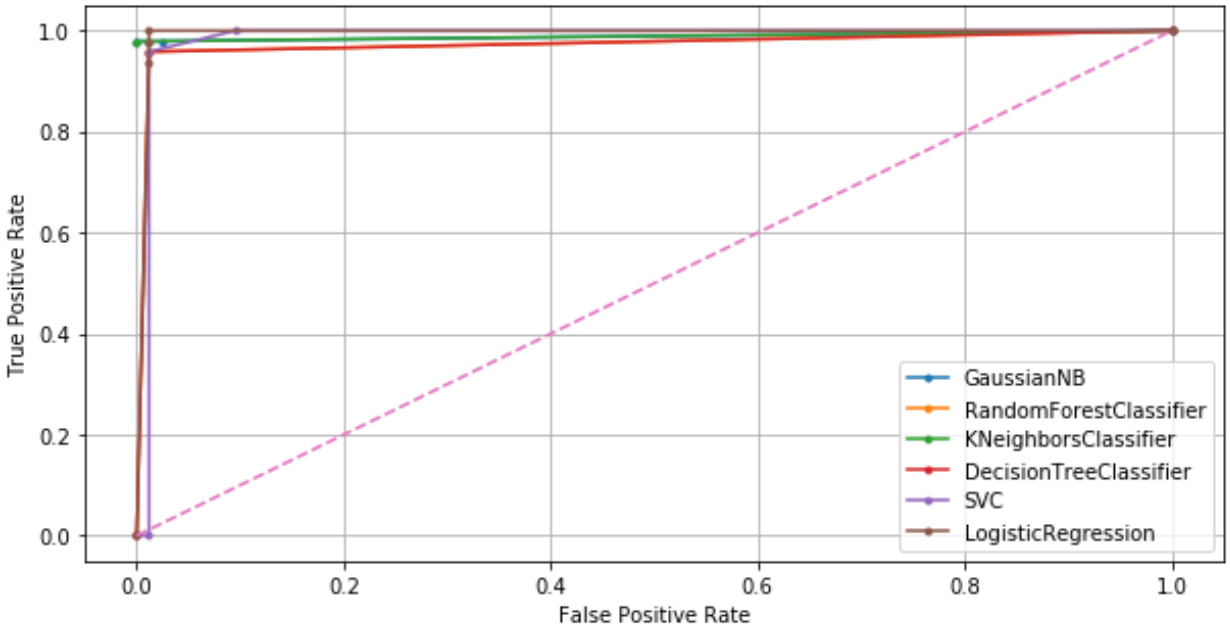

Figure 4. 12 ROC Curve Stacking Based Model

Following are the learning curve for stacking based model.

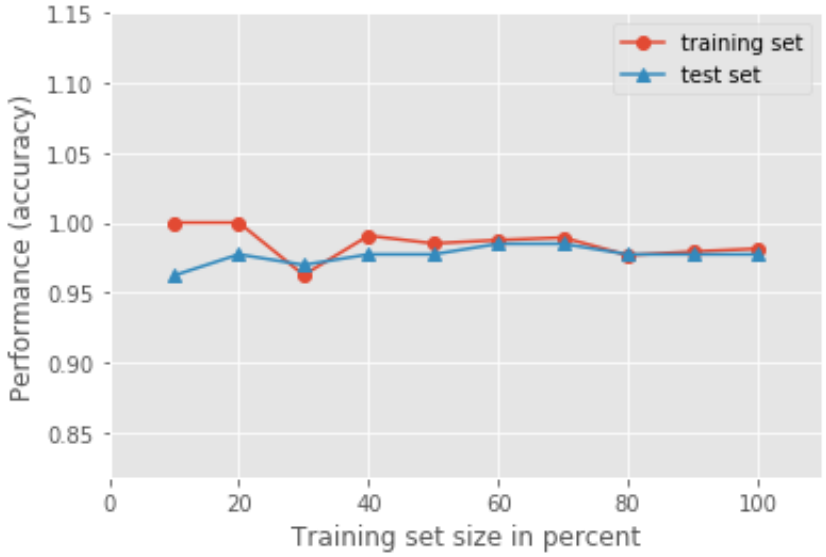

Figure 4. 13 Learning Curve Stacking Naive Bayes

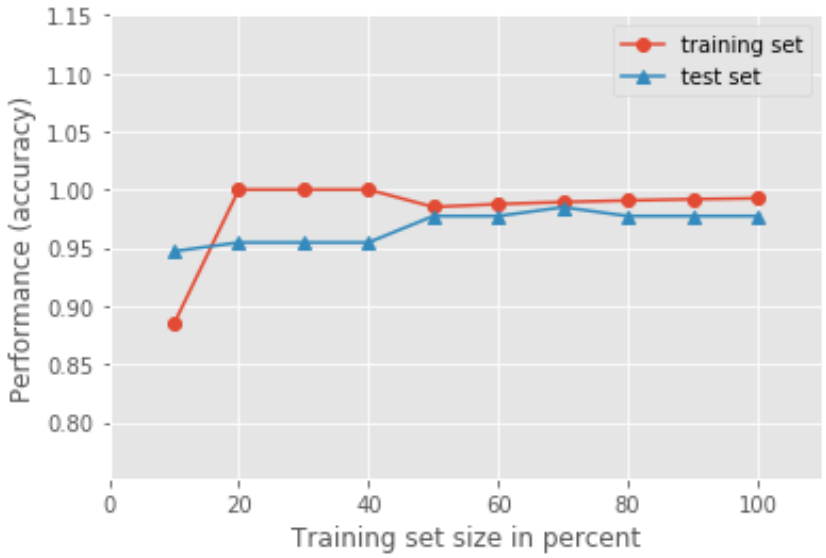

Figure 4. 14 Learning Curve Stacking Random Forest

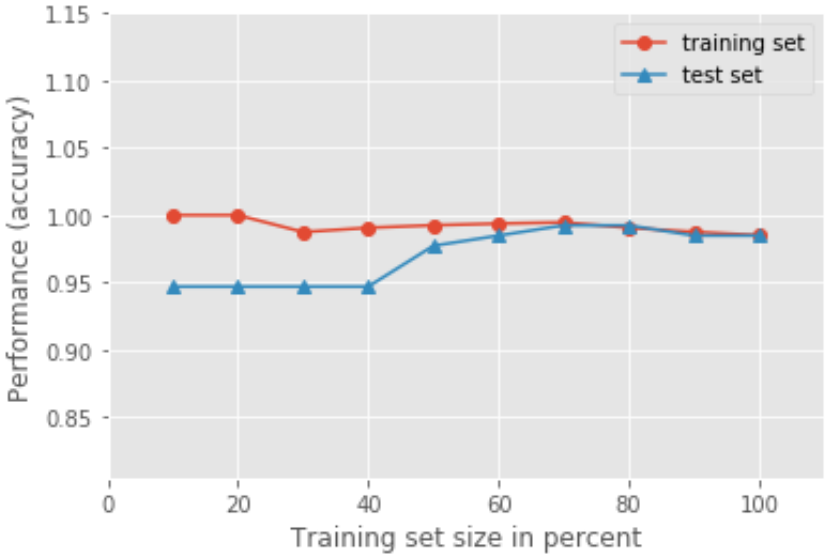

Figure 4. 15 Learning Curve Stacking K-Nearest Neighbor

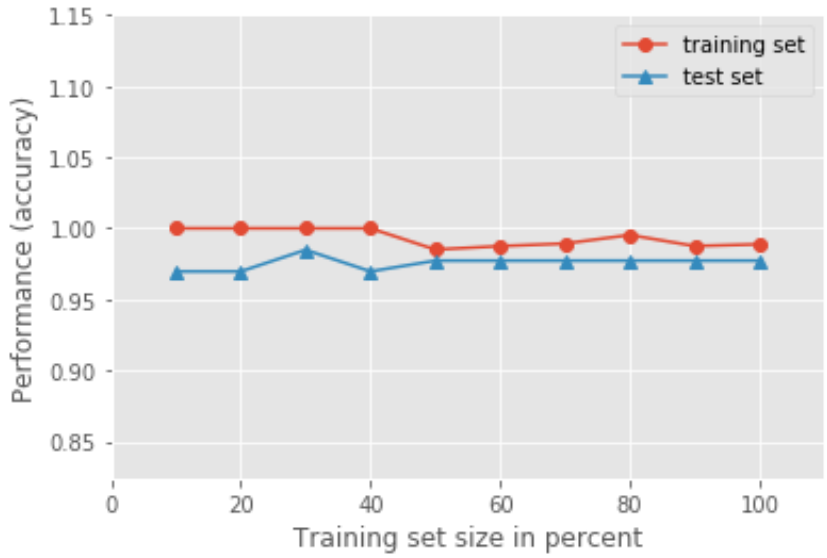

Figure 4. 16 Learning Curve Stacking Support Vector Machine

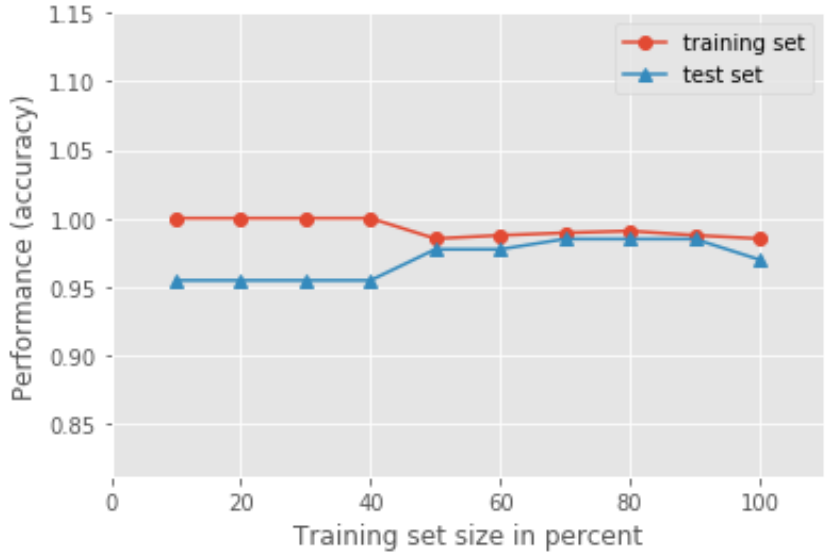

Figure 4. 17 Learning Curve Stacking Logistic Regression

From the learning curve we can see that:

- Naïve Bayes converged at 60% of data
- Random Forest converged at 70% but failed to converge till end
- K-Nearest Neighbor converged at 70% data
- Decision Tree converged at 90% data
- Support Vector Machine behaved like Decision Tree
- Logistic Regression started to converge at 70% of data but failed convergence after 90% of data. Finally no convergence at 100% of data.

In this ensemble Naïve Bayes and Decision Tree has performed as best together where beta classifier was the K-Nearest Neighbor with highest score in all sectors including accuracy, precision, recall, AUC, and confusion matrix.

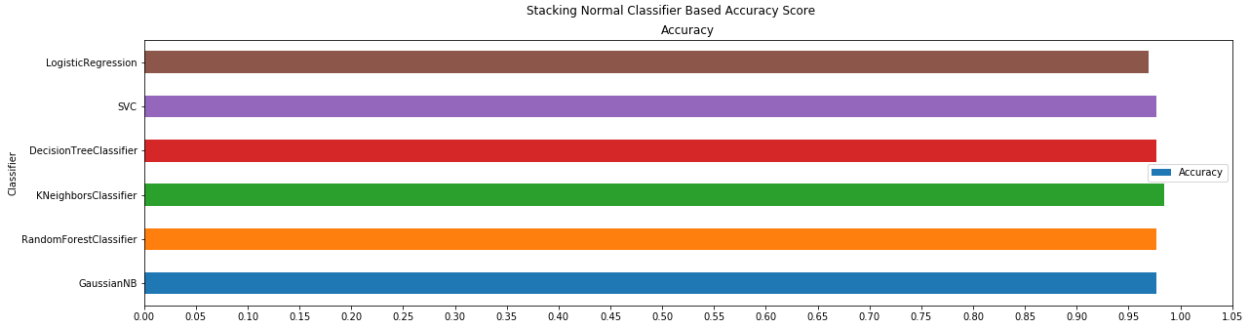

Figure 4. 18 Stacking Accuracy Score Chart

4.1.6 Stacking With Cross Validation (Ensemble)

Here we have used 5 fold cross validation for each stacking model.

Table 4. 6 Performance Report for Stacking Cross Validation

Metrics	Meta: NB Base: KNN, DT	Meta: RF Base: NB, KNN	Meta: KNN Base: NB, DT	Meta: DT Base: SVM, LR	Meta: SVM Base: DT, LR	Meta: LR Base: SVM, NB
Accuracy	0.970	0.985	0.965	0.985	0.985	0.977
Precision	0.97	0.99	0.97	0.98	0.98	0.98
Recall	0.97	0.98	0.96	0.98	0.98	0.98
F1-Score	0.97	0.99	0.96	0.98	0.98	0.98
Confusion Matrix	150 0 12 238	150 0 6 244	139 11 3 247	147 3 3 247	147 3 3 247	146 4 5 245

From the above table we can conclude that **Random Forest is the best classifier among all classifiers in terms of accuracy, precision, recall, f1-score**. From confusion matrix Random Forest has got the lowest FP and K-Nearest Neighbor, Decision Tree, Support Vector Machine has got the lowest FN count.

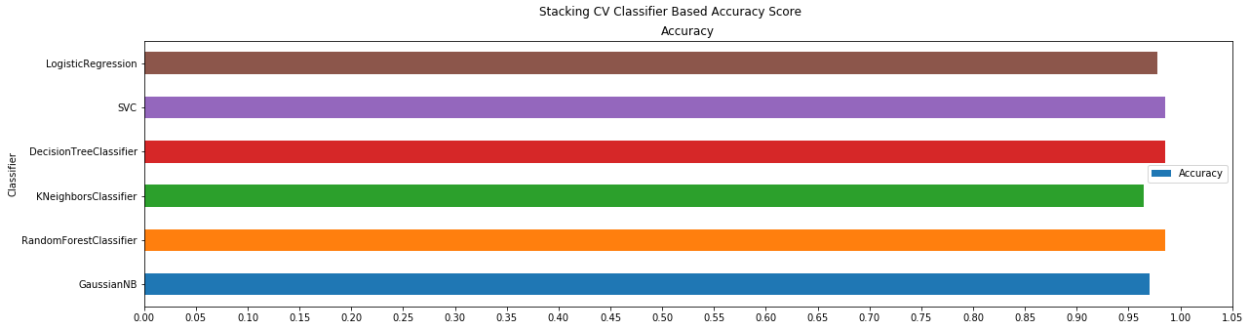

Figure 4. 19 Stacking Cross Validation Accuracy Score Chart

4.2 Experiment 2

Here in this experiment we have used reduced feature. Feature has been reduced by removing highly correlated features and by using highly scored features in Chi Square test. We have done a hypothetical test for finding best correlation threshold value. For finding minimum threshold of Chi Square test score we run another hypothetical test. And finally I've got convergence on a fixed threshold of both correlation and Chi Square score. Finally we have applied them before applying machine learning algorithms. Here in this experiment I'm working with 25 features only. Here I've followed same criteria and experimental job as we did in experiment 4.1.

4.2.1 Single Learner

Table 4. 7 Performance Report for Single Data Fold Learner with Reduced feature

Metrics	NB		RF		KNN		DT		SVM		LR	
Accuracy	0.9924		0.9016		0.9772		0.9469		0.9772		0.9772	
Precision	0.99		0.91		0.98		0.95		0.98		0.98	
Recall	0.99		0.90		0.98		0.95		0.98		0.98	
F1-Score	0.99		0.90		0.98		0.95		0.98		0.98	
AUC Score	0.994		0.914		0.999		0.958		0.997		0.999	
Confusion Matrix	48	0	46	2	47	1	48	0	45	3	46	2
	1	83	11	73	2	82	7	77	0	84	1	83

Here **Naïve Bayes has performed as best in terms of accuracy, precision, recall and f1-score**. It also has got best FP and FN count.

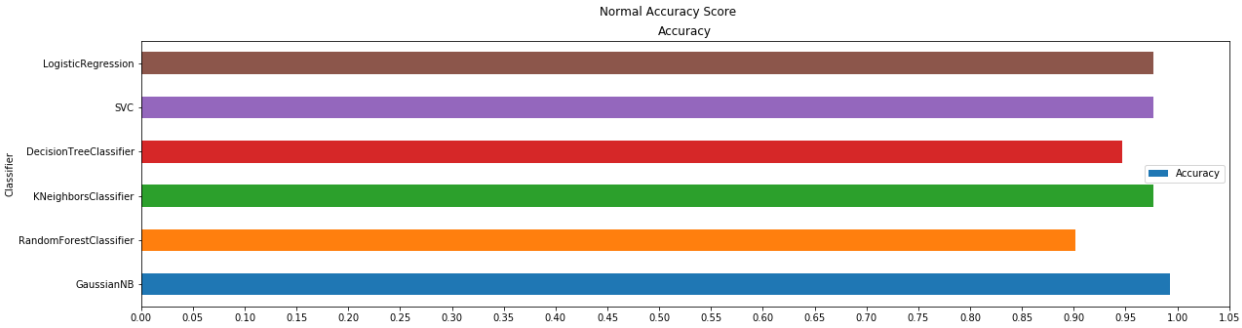

Figure 4. 20 Normal Accuracy Score Chart

Following is the ROC curve for all models as single learner.

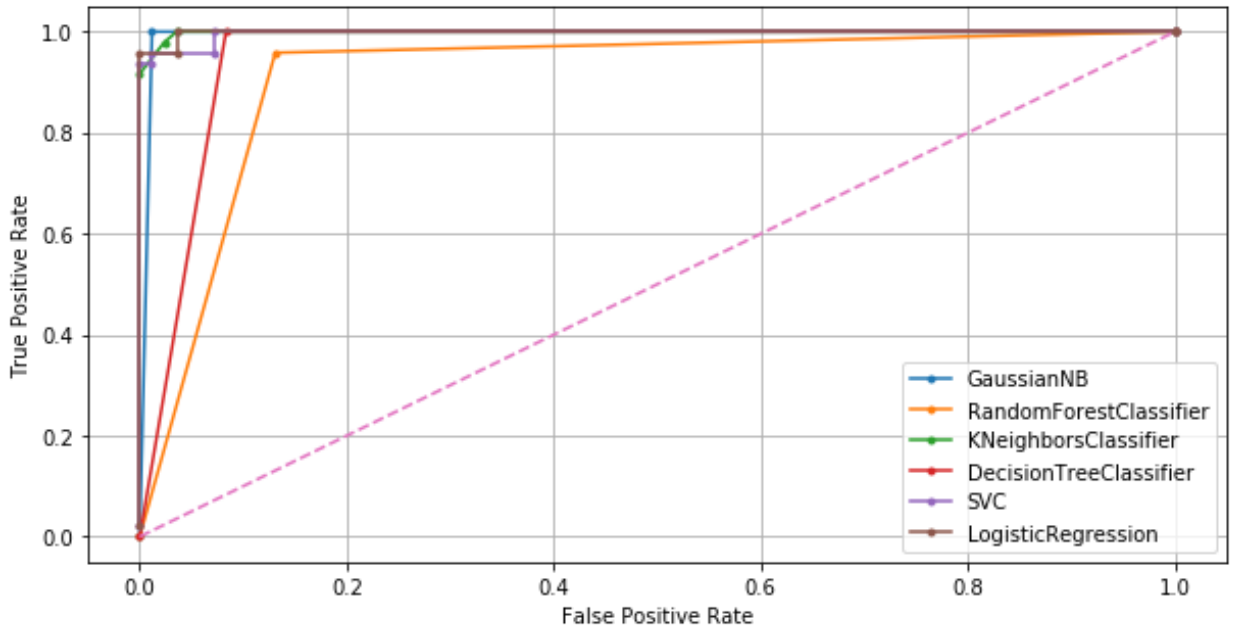

Figure 4. 21 ROC Curve

As per the Curve We can see that almost all learners have covered most of the area under the curve. In terms of AUC score Logistic Regression has scored best as 0.997.

Following are the learning curves for all the models.

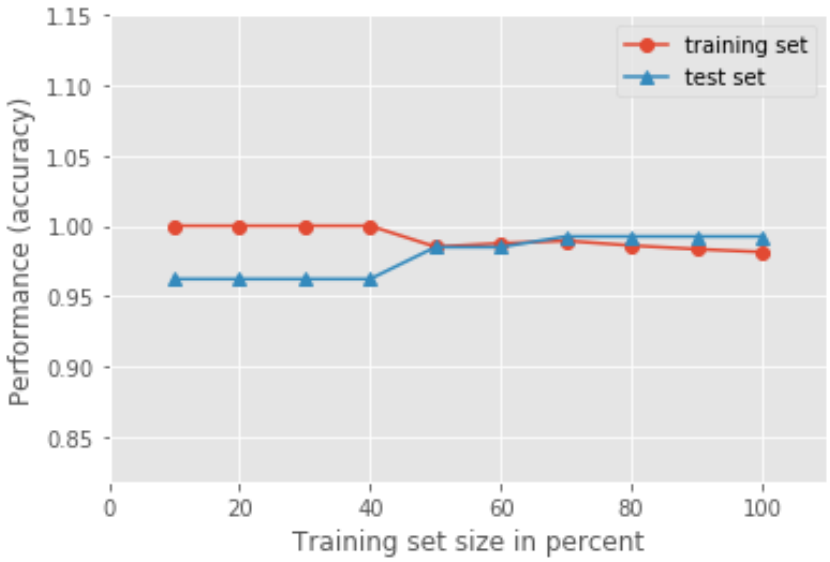

Figure 4. 22 Learning Curve Naive Bayes (Reduced Feature)

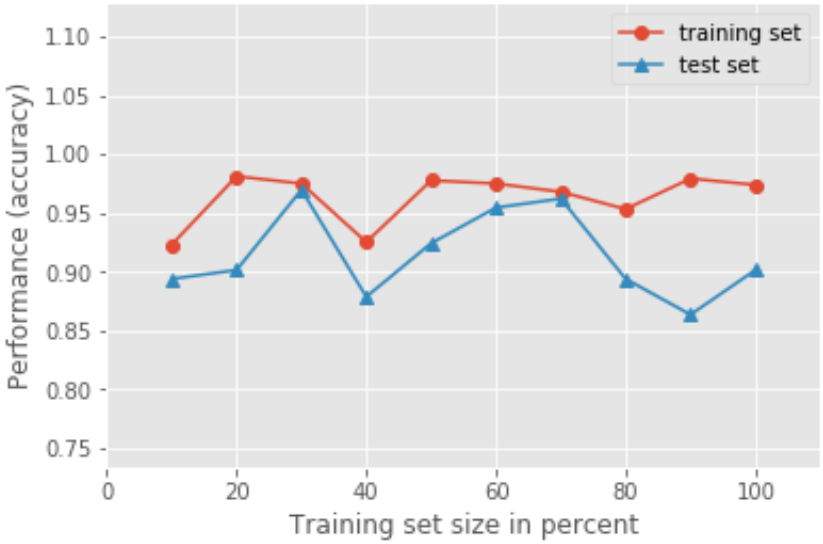

Figure 4. 23 Learning Curve Random Forest (Reduced Feature)

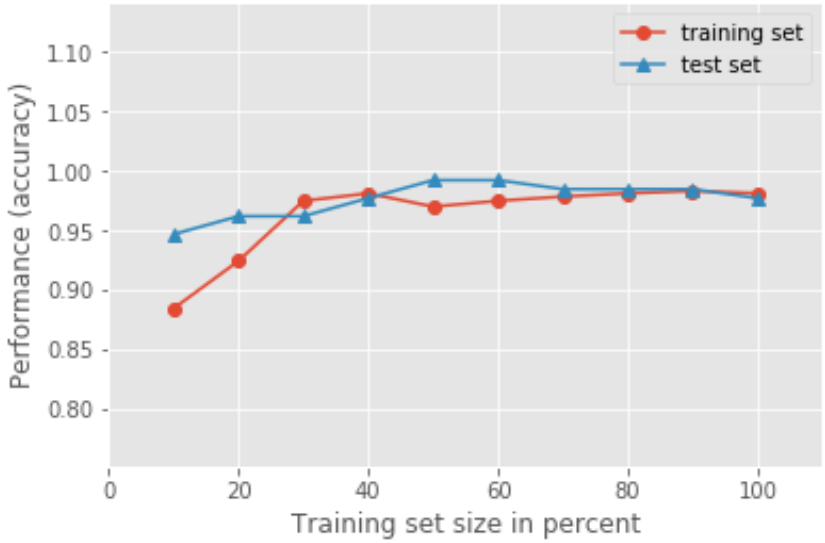

Figure 4. 24 Learning Curve K-Nearest Neighbor (Reduced Feature)

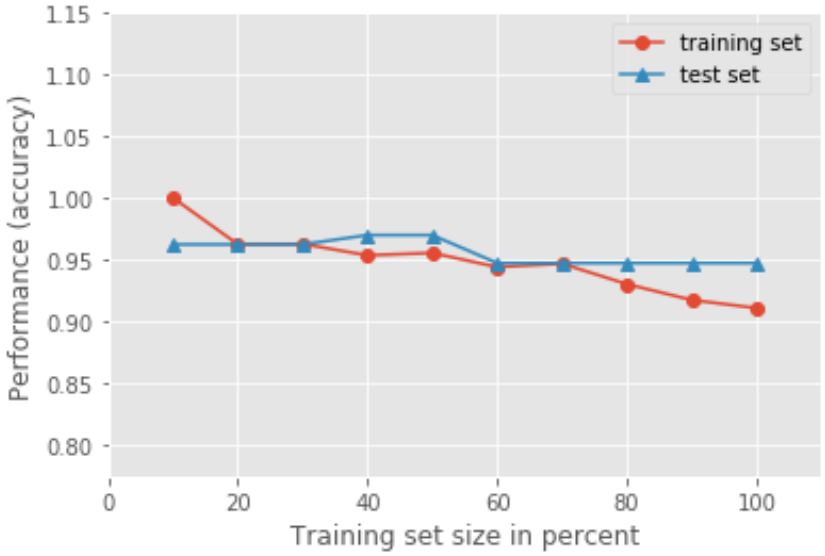

Figure 4. 25 Learning Curve Decision Tree (Reduced Feature)

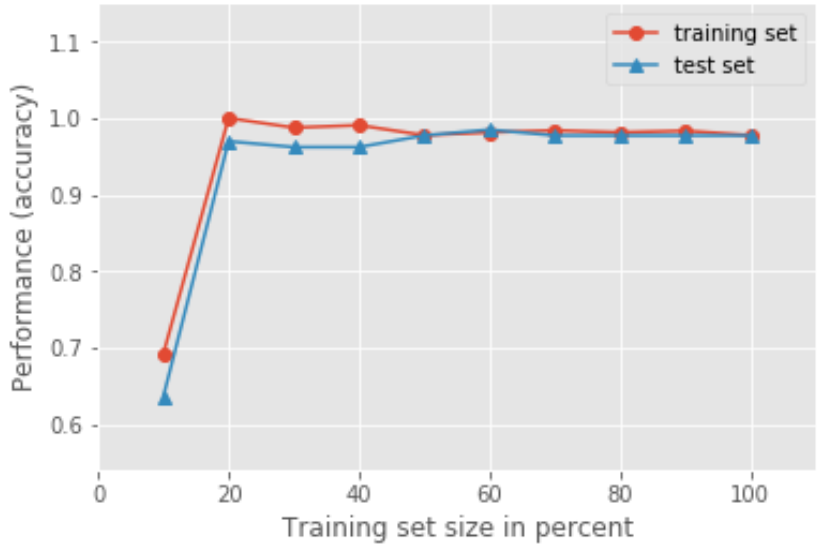

Figure 4. 26 Learning Curve Support Vector Machine (Reduced Feature)

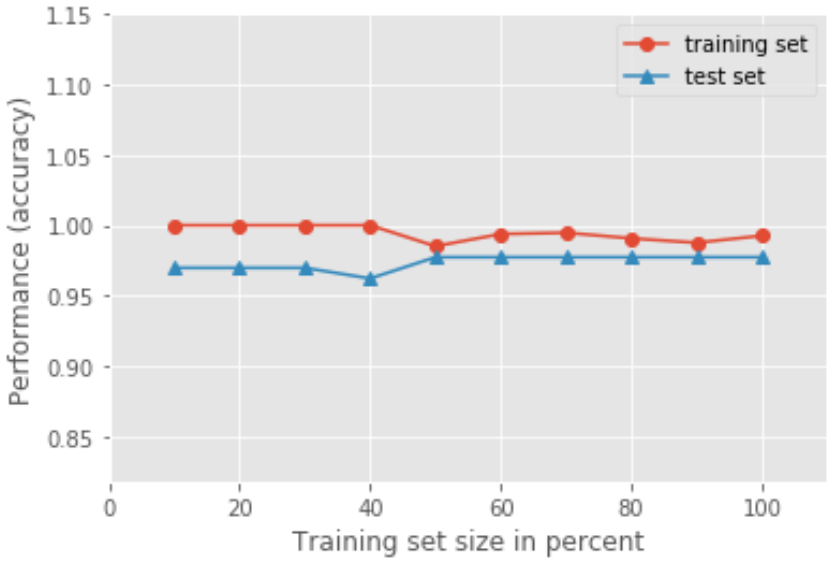

Figure 4. 27 Learning Curve Logistic Regression (Reduced Feature)

4.2.2 10 Fold Cross Validation

Here I've applied a 10 fold cross validation on each learner to measure their performance. Following is the result:

Table 4. 8 Performance Report for Cross Validation learner with reduced feature

Metrics	NB		RF		KNN		DT		SVM		LR	
Accuracy	0.980		0.945		0.972		0.917		0.975		0.982	
Precision	0.99		0.94		0.97		0.92		0.97		0.98	
Recall	0.98		0.94		0.97		0.92		0.97		0.98	
F1-Score	0.98		0.94		0.97		0.92		0.97		0.98	
Confusion Matrix	148	2	138	12	148	2	140	10	144	6	147	3
	6	244	10	240	9	241	23	227	4	246	4	246

From the table we can see that **Logistic Regression has performed as best in terms of accuracy, precision, recall and f1-score**. From the confusion matrix we can see that Naïve Bayes and K-Nearest Neighbor has the lowest FP count and Support Vector Machine has the lowest FN count.

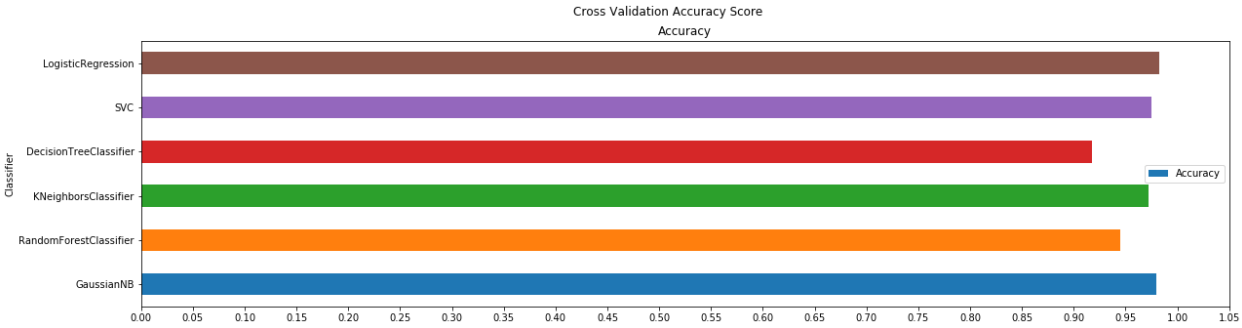

Figure 4. 28 Cross Validation Accuracy Score Chart

4.2.3 Bagging (Ensemble)

We have applied bootstrap aggregation ensemble learning technique on the models we have chosen. We used a 5 fold cross validation and 50% max sample and 50% max feature to train the model. Following is the result:

Table 4. 9 Performance Report for bagging classifier with reduced feature

Metrics	NB		RF		KNN		DT		SVM		LR	
Accuracy	0.957		0.982		0.987		0.932		0.982		0.981	
Precision	0.96		0.98		0.99		0.93		0.98		0.98	
Recall	0.96		0.98		0.99		0.93		0.98		0.98	
F1-Score	0.96		0.98		0.99		0.93		0.98		0.98	
Confusion Matrix	148	2	145	5	149	1	142	8	144	6	147	3
	15	235	2	248	4	246	19	231	1	249	5	245

After applying bagging ensemble learning method **Random Forest** have got improvement. It has performed as best with the highest accuracy of 0.994 including highest precision, recall and f1-score of 0.99. Random Forest has lowest FN lower FP. Naïve Bayes has got lowest FP count.

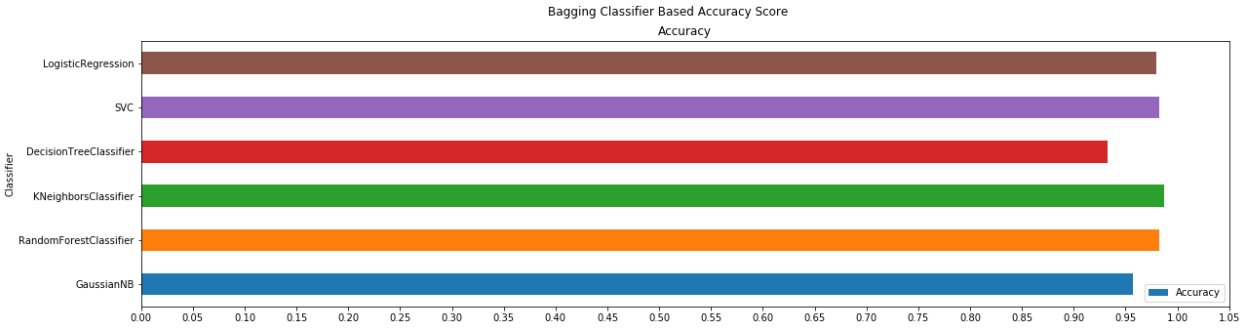

Figure 4. 29 Bagging Accuracy Score Chart

4.2.4 Boosting (Ensemble)

We have applied boosting ensemble learning technique on every model except K-Nearest Neighbor as this model does not support weighted sampling which is the fundamental of boosting. We have used learning rate of 1 and 10 estimator which 5 fold cross validation. Following is the result:

Table 4. 10 Performance Report for boosting with reduced feature

Metrics	NB		RF		DT		SVM		LR	
Accuracy	0.981		0.975		0.975		0.981		0.967	
Precision	0.98		0.98		0.98		0.98		0.97	
Recall	0.98		0.97		0.97		0.98		0.97	
F1-Score	0.98		0.98		0.98		0.98		0.97	
Confusion Matrix	146	4	147	3	148	2	149	1	146	4
	4	246	7	243	8	242	7	243	9	241

After applying boosting ensemble learning technique we can see that **Support Vector Machine has performed as best in terms of accuracy, precision, recall and f1-score**. It has the lowest FN. Both Decision Tree and Support vector Machine has got lowest FP.

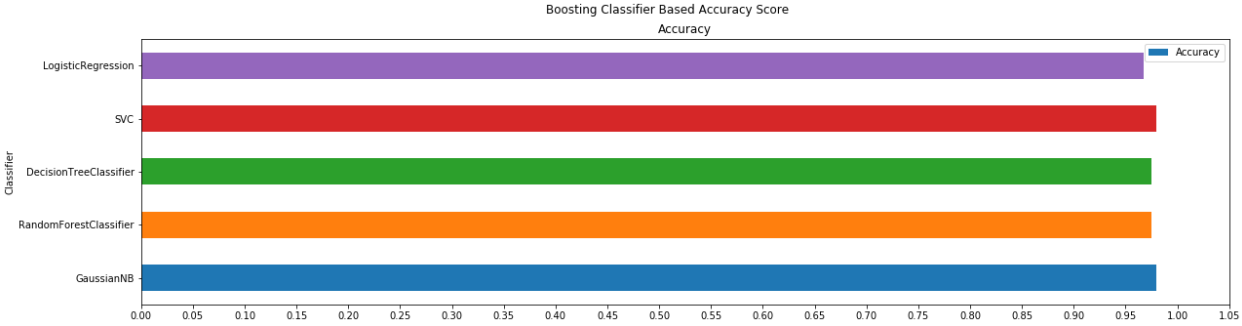

Figure 4. 30 Boosting Accuracy Score Chart

4.2.5 Stacking W/O Cross Validation (Ensemble)

Table 4. 11 Performance Report for stacking without cross validation with reduced feature

Metrics	Meta: NB Base: KNN, DT		Meta: RF Base: NB, KNN		Meta: KNN Base: NB, DT		Meta: DT Base: SVM, LR		Meta: SVM Base: DT, LR		Meta: LR Base: SVM, NB	
Accuracy	0.9772		0.9848		1.0000		0.9772		0.9772		0.9924	
Precision	0.98		0.98		1.00		0.98		0.98		0.99	
Recall	0.98		0.98		1.00		0.98		0.98		0.99	
F1-Score	0.98		0.98		1.00		0.98		0.98		0.99	
AUC Score	0.999		0.984		1.000		0.973		0.986		1.000	
Confusion Matrix	47	1	47	1	48	0	46	2	46	2	48	0
	2	82	1	83	0	84	1	83	1	83	1	83

Here **K-Nearest Neighbor** has performed a top-notched scores of 100% accuracy.

Following is the ROC curve for stacking based model without cross validation.

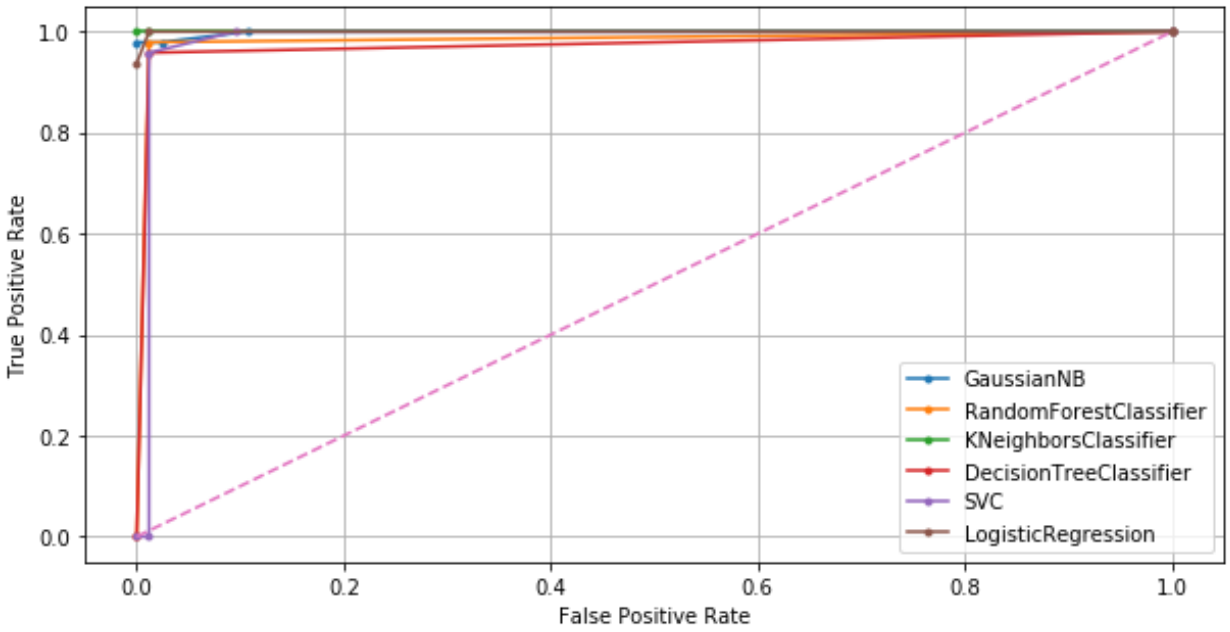

Figure 4. 31 ROC Curve Stacking Based Model

Following are the learning curve for stacking based model.

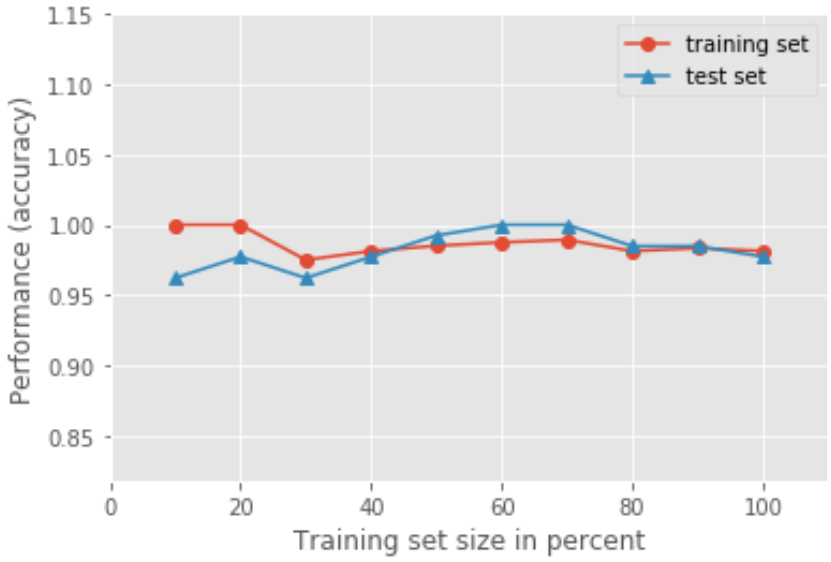

Figure 4. 32 Learning Curve Stacking Naive Bayes (Reduced Feature)

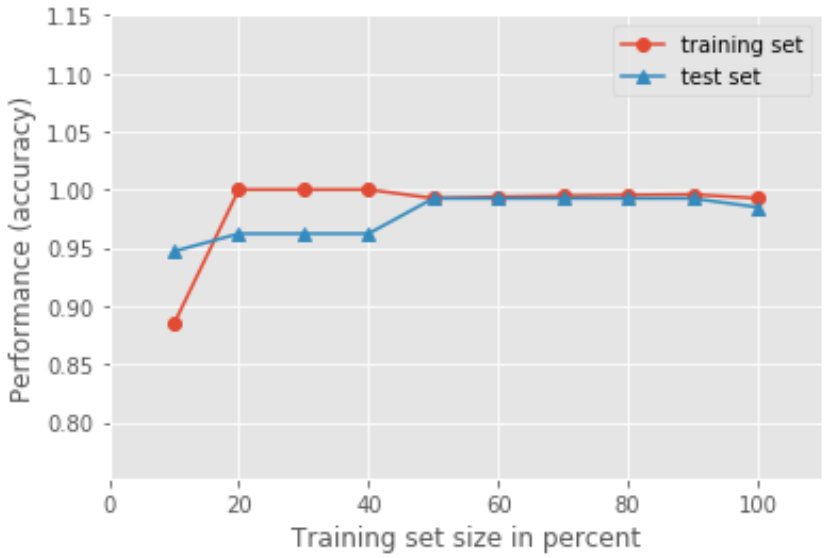

Figure 4. 33 Learning Curve Stacking Random Forest (Reduced Feature)

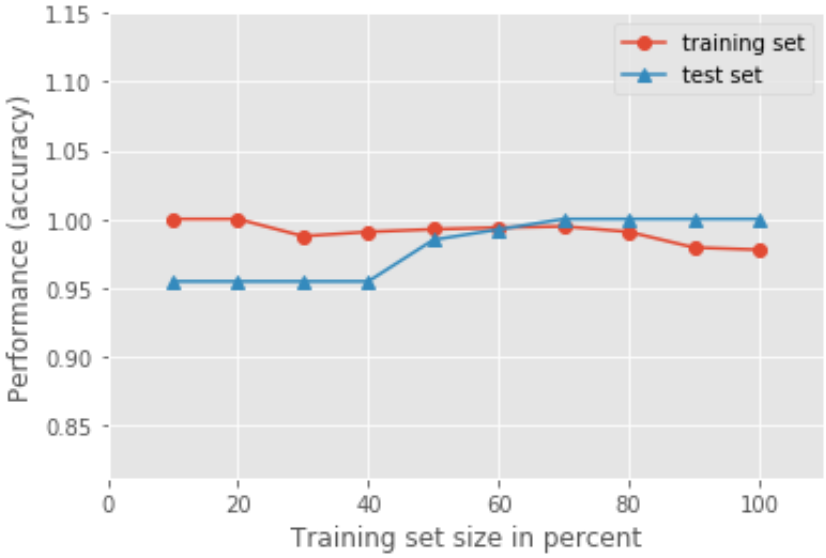

Figure 4. 34 Learning Curve Stacking K-Nearest Neighbor (Reduced Feature)

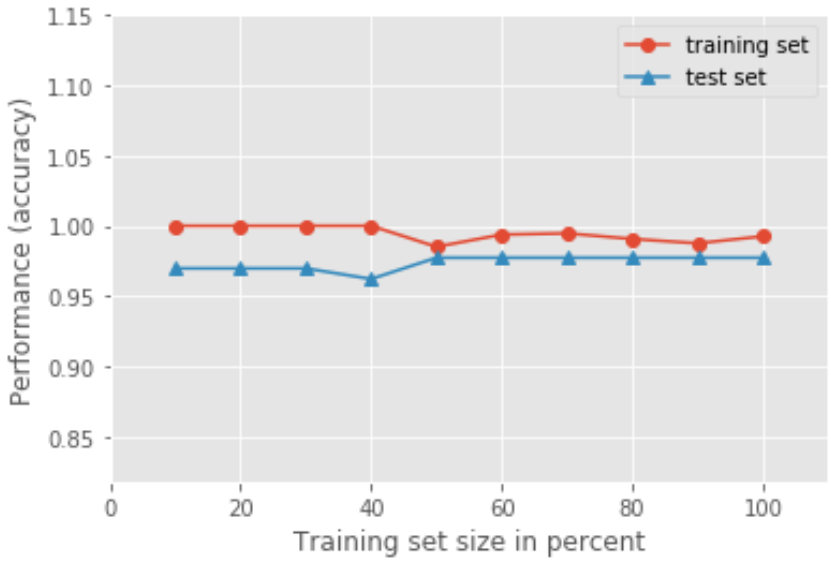

Figure 4. 35 Learning Curve Stacking Decision Tree (Reduced Feature)

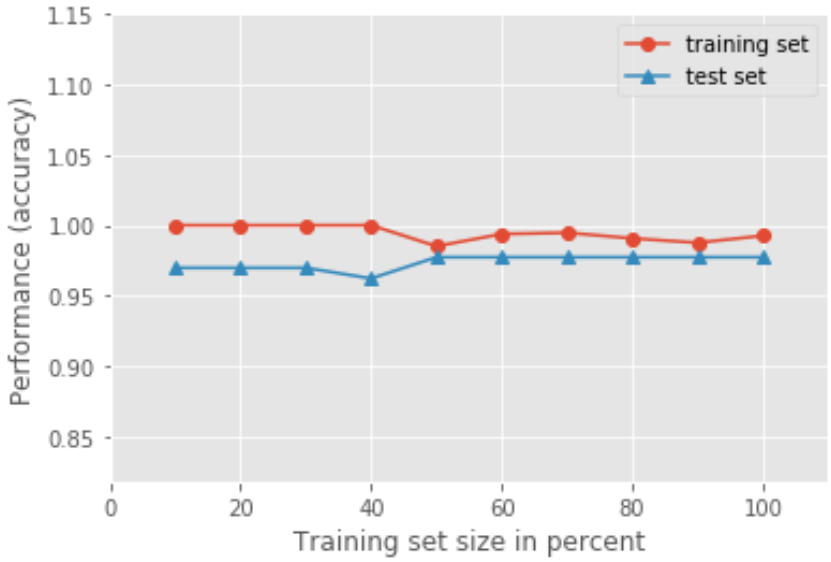

Figure 4. 36 Learning Curve Stacking Support Vector Machine (Reduced Feature)

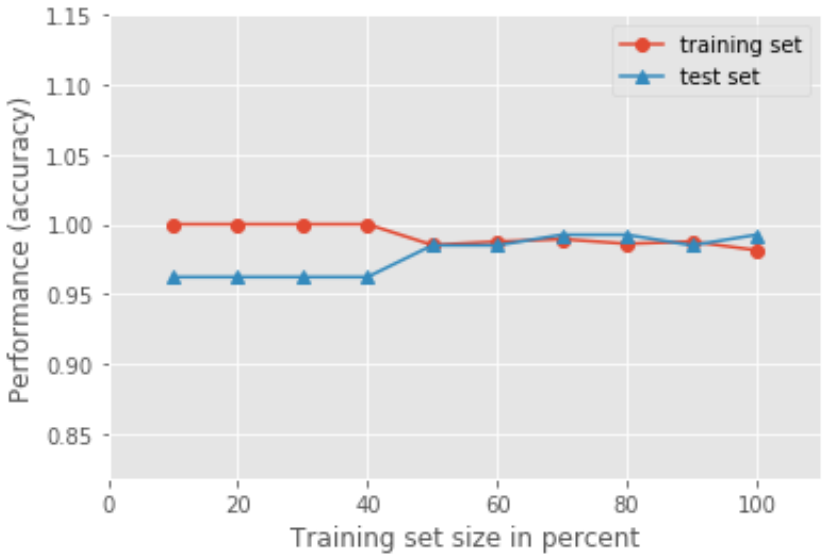

Figure 4. 37 Learning Curve Stacking Logistic Regression (Reduced Feature)

From the learning curve we can see that:

- Naïve Bayes converged at 90% of data. After that it has lost convergence. And finally at 100% data point it has converged.
- Random Forest converged at 50% and continue this up to 90%
- K-Nearest Neighbor converged at 60% data but after that lost convergence
- Decision Tree converged has no convergence data point
- Support Vector Machine behaved like Decision Tree
- Logistic Regression started to converge at 50% of data but failed convergence after 70% of data. Finally no convergence at 100% of data.

In this ensemble Naïve Bayes and Decision Tree has performed as best together where beta classifier was the K-Nearest Neighbor with highest score in all sectors including accuracy, precision, recall, AUC, and confusion matrix.

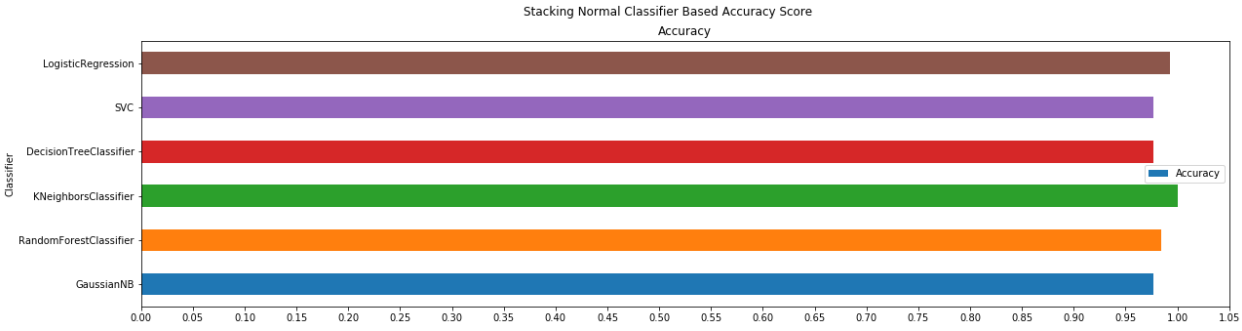

Figure 4. 38 Stacking Accuracy Score Chart

4.2.6 Stacking With Cross Validation (Ensemble)

Here we have used 5 fold cross validation for each stacking model.

Table 4. 12 Performance Report for stacking with cross validation and reduced feature

Metrics	Meta: NB Base: KNN, DT	Meta: RF Base: NB, KNN	Meta: KNN Base: NB, DT	Meta: DT Base: SVM, LR	Meta: SVM Base: DT, LR	Meta: LR Base: SVM, NB
Accuracy	0.972	0.981	0.971	0.982	0.982	0.972
Precision	0.97	0.98	0.97	0.98	0.98	0.97
Recall	0.97	0.98	0.97	0.98	0.98	0.97
F1-Score	0.97	0.98	0.97	0.98	0.98	0.97
Confusion Matrix	148 2 9 241	146 4 4 246	138 12 0 250	147 3 4 246	147 3 3 246	142 8 3 247

From the above table we can conclude that **Decision Tree and Support Vector Machine both are the best classifier among all classifiers in terms of accuracy, precision, recall, f1-score**. From confusion matrix Naïve Bayes and Logistic Regression has got the lowest FP and K-Nearest Neighbor has got the lowest FN count.

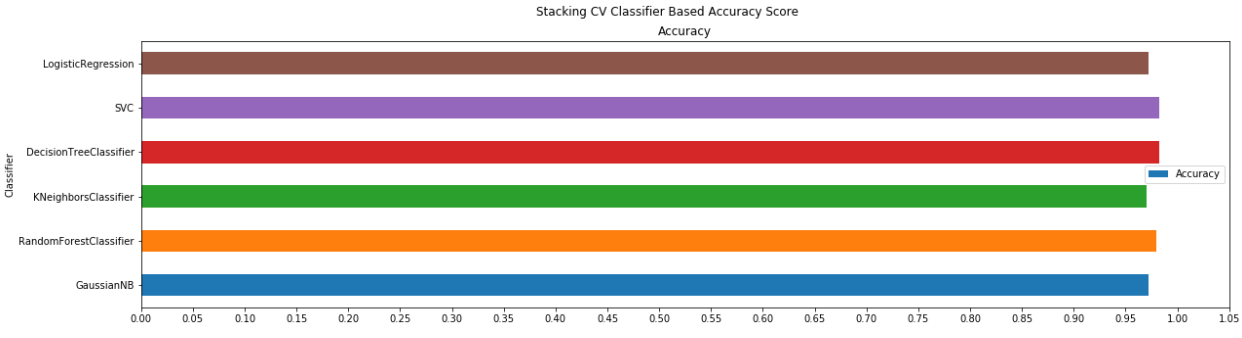

Figure 4. 39 Stacking Cross Validation Accuracy Score Chart

4.3 Decision Making

Following is the line plot of all classifier, all technique summary at a glance.

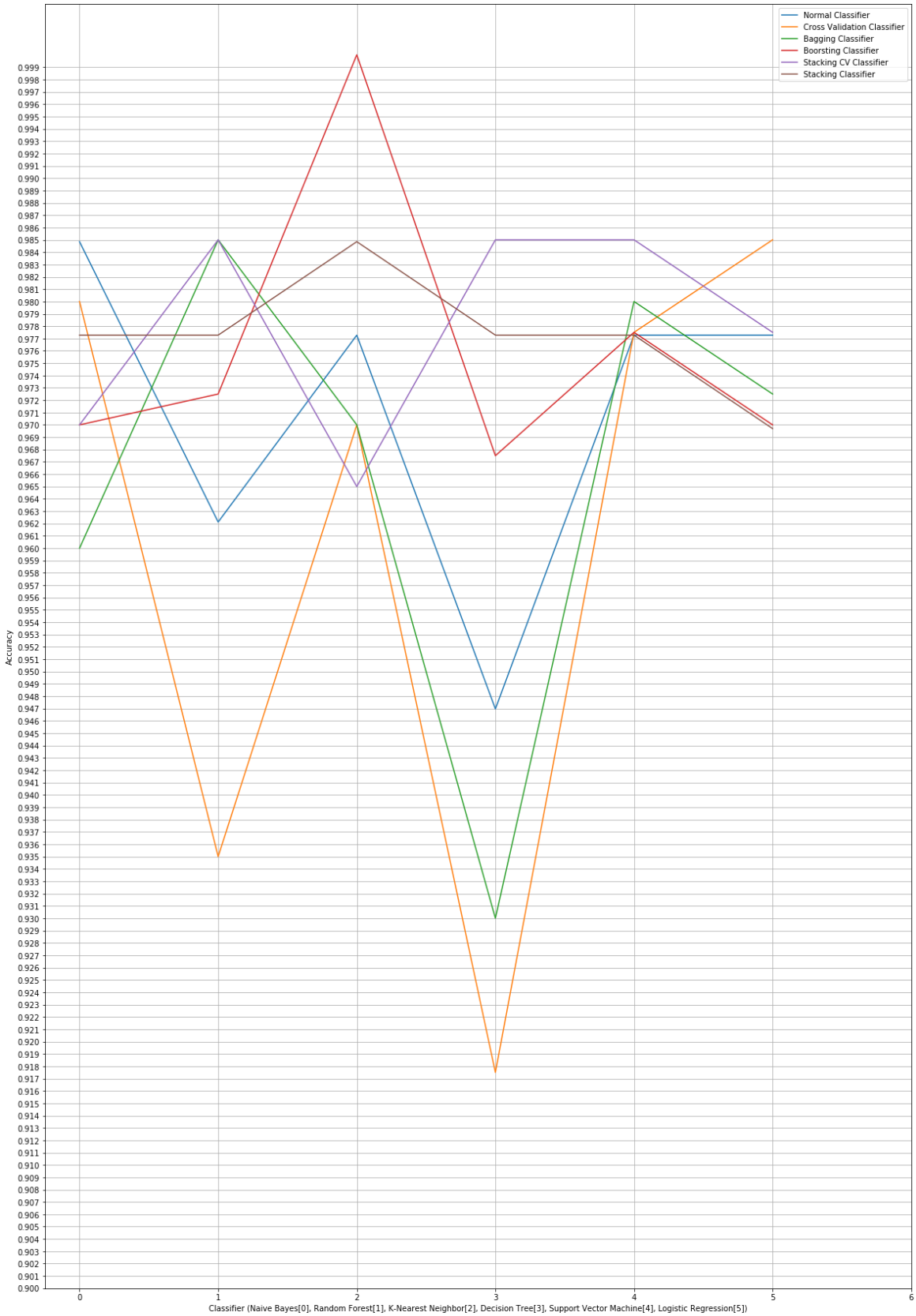

Figure 4. 40 All Feature Standpoint Summary

Figure 4. 41 Reduced Feature Standpoint Summary

Please note that for both of the image in terms of the line plot of “Boosting” please ignore the accuracy point of K-Nearest Neighbor as KNN doesn’t support weighted sampling.

In both graph X-axis represents some integer number. Each number represent each classifier.

0 → Naïve Bayes

1 → Random Forest

2 → K-Nearest Neighbor

3 → Decision Tree

4 → Support Vector Machine

5 → Logistic Regression

Y-axis represent the accuracy in decimal point.

From the figure of *All Feature Standpoint Summary:*

- Naïve Bayes has performed best as single model based learning (Accuracy: -0.9848, FN: - 1).
- Logistic Regression has performed best in 10 fold cross validation based model (Accuracy: -0.9850, FN: - 3).
  Closer Support Vector Machine (Accuracy: -0.9770, FN: - 2)
- Random Forest has performed best in bagging based model (Accuracy: -0.9850, FN: - 0).
  Closer Support Vector Machine (Accuracy: -0.9800, FN: - 3)
- Support Vector Machine has performed best in boosting model (Accuracy: -0.9700, FN: - 4).
  Closer Random Forest (Accuracy: -0.9720, FN: - 4)
- Decision Tree and Support Vector Machine has performed best in stacking cross validation based model (Accuracy: -0.9850, FN: - 3).
- K-Nearest Neighbor has performed best in stacking w/0 cross validation based model (Accuracy: -0.9848, FN: - 1).

From the figure of *Reduced Feature Standpoint Summary:*

- Naïve Bayes has performed best as single model based learning (Accuracy: - 0.9924, FN: - 1).
- Logistic Regression has performed best in 10 fold cross validation based model (Accuracy: - 0.9825, FN: - 4).
  Closer Support Vector Machine (Accuracy: - 0.9750, FN: - 4)
- Support Vector Machine has performed best in bagging based model (Accuracy: - 0.9820, FN: - 1).
  Closer Random Forest (Accuracy: - 0.9820, FN: - 2)
- Naïve Bayes has performed best in boosting model (Accuracy: - 0.9810, FN: - 4).
  Closer Support Vector Machine (Accuracy: - 0.9810, FN: - 7)
- Support Vector Machine has performed best in stacking cross validation based model (Accuracy: - 0.9820, FN: - 3). Closer K-Nearest Neighbor (Accuracy: - 0.9710, FN: - 0) and Logistic Regression (Accuracy: - 0.9720, FN: - 3)
- K-Nearest Neighbor has performed best in stacking w/o cross validation based model (1.0000). Closer Logistic Regression (0.9924)

As accuracy has improved in reduced feature based model thus we will further discuss considering the result generated from reduced feature based model.

Among the complete result we will select a few model for further discussion.

Naïve Bayes: - As it has performed well as single model and cross validation based model

Random Forest: - As it has performed well in cross validation based model and stacking based cross validated model

Support Vector Machine: - As it has a significant best performance in boosting and cross validation based stacking model.

Let's take a deeper dive into these model in given perspective.

Table 4. 13 Selected Single Model Based Learner

Metrics	NB		RF		SVM	
Accuracy	0.9924		0.9016		0.9772	
Precision	0.99		0.91		0.98	
Recall	0.99		0.90		0.98	
F1-Score	0.99		0.90		0.98	
AUC Score	0.994		0.914		0.997	
Confusion Matrix	48	0	46	2	45	3
	1	83	11	73	0	84

Here I've chosen Support Vector Machine because it has better AUC as well as lowest FN (False Negative) count.

Table 4. 14 Selected Cross Validation Based Learner

Metrics	NB		RF		SVM	
Accuracy	0.980		0.945		0.975	
Precision	0.99		0.94		0.97	
Recall	0.98		0.94		0.97	
F1-Score	0.98		0.94		0.97	
Confusion Matrix	148	2	138	12	144	6
	6	244	10	240	4	246

Here I've chosen Support Vector Machine because it has lowest FN (False Negative) count.

Table 4. 15 Selected Bagging Based Learners

Metrics	NB		RF		SVM	
Accuracy	0.957		0.982		0.982	
Precision	0.96		0.98		0.98	
Recall	0.96		0.98		0.98	
F1-Score	0.96		0.98		0.98	
Confusion Matrix	148	2	145	5	144	6
	15	235	2	248	1	249

Here I've chosen Support Vector Machine because it has best accuracy, precision, recall, f1-score and AUC as well as lowest FN (False Negative) count.

Table 4. 16 Selected Boosting Based Learners

Metrics	NB		RF		SVM	
Accuracy	0.981		0.975		0.981	
Precision	0.98		0.98		0.98	
Recall	0.98		0.97		0.98	
F1-Score	0.98		0.98		0.98	
Confusion Matrix	146	4	147	3	149	1
	4	246	7	243	7	243

Here I've chosen Naïve Bayes because it has best accuracy, precision, recall, f1-score and AUC as well as lowest FN (False Negative) count.

Table 4. 17 Selected Stacking Based Single Fold Learners

Metrics	Meta: NB Base: KNN, DT		Meta: RF Base: NB, KNN		Meta: SVM Base: DT, LR	
Accuracy	0.9772		0.9848		0.9772	
Precision	0.98		0.98		0.98	
Recall	0.98		0.98		0.98	
F1-Score	0.98		0.98		0.98	
AUC Score	0.999		0.984		0.986	
Confusion Matrix	47	1	47	1	46	2
	2	82	1	83	1	83

Here I've chosen Random Forest because it has best accuracy, precision, recall, f1-score as well as lowest FN (False Negative) count.

Table 4. 18 Selected Stacking Based Cross Validation Learners

Metrics	Meta: NB Base: KNN, DT		Meta: RF Base: NB, KNN		Meta: SVM Base: DT, LR	
Accuracy	0.972		0.981		0.982	
Precision	0.97		0.98		0.98	
Recall	0.97		0.98		0.98	
F1-Score	0.97		0.98		0.98	
Confusion Matrix	148	2	146	4	147	3
	9	241	4	246	3	246

Here I've chosen Support Vector Machine because it has best accuracy, precision, recall, f1-score and AUC as well as lowest FN (False Negative) count.

4.4 Result Summary

Here I've chosen lowest FN count over accuracy in many cases. This is because in case of disease prediction a false negative may lead to national health damage into catastrophic level. Because false negative means the patient has been predicted with having no disease wrongly. This is not tolerable I terms of health and medical science. That's why I choose for lowest FN count over accuracy.

Table 4. 19 Selected Classifier Summary

Single Model	Cross Validation	Bagging	Boosting	Stacking Single Model	Stacking Cross Validation
SVM	SVM	SVM	NB	RF	SVM

Table 4. 20 Performance comparison report

Finally Support Vector Machine is the best performer in terms of accuracy as well as other metrics.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Findings and Contributions

In this thesis I've started with an imbalanced and noisy dataset. We has performed 7 different data preprocessing steps to make the dataset workable with machine learning algorithms. From all the 49 features (features increased after one-hot encoding) I've been able to reduce the length of feature list to 25. I've also proved that performance improved with the reduced features. Finally among all the classifiers and among different ways for applying them we have been able to choose the best classifier in terms of many performance metrics. We have not gone by the tradition where we could pick a model having great accuracy. I've applied common sense of medical disease diagnosis to pick our best model. While picking our best model we have keep an eye on traditional thing also like accuracy. Also we have chosen a model having lowest false negative. Because getting diagnosed with no disease while having disease may be worse by time.

5.2 Recommendations for Future Works

For data science related problem data is everything. Our dataset was very small to work with. Only 400 records is not that much good for better prediction. More data will add more information, more hidden pattern, and more dimension. In future more data can be added into this dataset and most important more variety of data will be a plus point for further research.

REFERENCES

Jha, V., Garcia-Garcia, G., Iseki, K., Li, Z., Naicker, S., Plattner, B., ... & Yang, C. W. (2013). Chronic kidney disease: global dimension and perspectives. *The Lancet*, *382*(9888), 260-272.

Levey, A. S., Atkins, R., Coresh, J., Cohen, E. P., Collins, A. J., Eckardt, K. U., ... & Powe, N. R. (2007). Chronic kidney disease as a global public health problem: approaches and initiatives–a position statement from Kidney Disease Improving Global Outcomes. *Kidney international*, *72*(3), 247-259.

Seo, S. (2006). *A review and comparison of methods for detecting outliers in univariate data sets* (Doctoral dissertation, University of Pittsburgh).

Jena, L., & Kamila, N. K. (2015). Distributed data mining classification algorithms for prediction of chronic-kidney-disease. *Int. J. Emerg. Res. Manag. &Technology*, *9359*(11), 110-118.

L. Breiman, "Random Forests," Machine Learning, vol. 45, p. 5–32, 2001.

Subasi, A., Alickovic, E., & Kevric, J. (2017). Diagnosis of chronic kidney disease by using random forest. In *CMBEBIH 2017* (pp. 589-594). Springer, Singapore.

L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, no. 2, p. 1883, 2009.

Salekin, A., & Stankovic, J. (2016, October). Detection of chronic kidney disease and selecting important predictive attributes. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 262-270). IEEE.

V. Vapnik, The Nature of Statistical Learning Theory, New York: Springer Verlag, 1995.

S. Armin, Data Mining and Knowledge Discovery Handbook, 2nd ed., O. Maimon and L. Rokach, Eds., New York: Springer, 2010.

Hosmer DW, Lemeshow S. Applied logistic regression. New York, NY: Wiley, 1989.

Ayer, T., Chhatwal, J., Alagoz, O., Kahn Jr, C. E., Woods, R. W., & Burnside, E. S. (2010). Comparison of logistic regression and artificial neural network models in breast cancer risk estimation. *Radiographics*, *30*(1), 13-22.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Matatov, N., Rokach, L., & Maimon, O. (2010). Privacy-preserving data mining: A feature set partitioning approach. *Information Sciences*, *180*(14), 2696-2720.

Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014). A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*, *4*(2), 0-0.

Prajwala, T. R. (2015). A comparative study on decision tree and random forest using R tool. *International journal of advanced research in computer and communication engineering*, *4*(1), 196-199.

Vijayarani, S., Dhayanand, S., & Phil, M. (2015). Kidney disease prediction using SVM and ANN algorithms. *International Journal of Computing and Business Research (IJCBR)*, *6*(2).

Sinha, P., & Sinha, P. (2015). Comparative study of chronic kidney disease prediction using KNN and SVM. *International Journal of Engineering Research and Technology*, *4*(12), 608-12.

Kaur, G., & Sidhu, E. B. K. (2014). Proposing Efficient Neural Network Training Model for Thyroid Disease Diagnosis. *International Journal for Technological Research in Engineering*, *1*(11).

Aljahdali, S., & Hussain, S. N. (2013). Comparative prediction performance with support vector machine and random forest classification techniques. *International Journal of Computer Applications*, *69*(11).

Kumar, M. (2016). Prediction of chronic kidney disease using random forest machine learning algorithm. *International Journal of Computer Science and Mobile Computing*, *5*(2), 24-33.

Rubini, L. J., & Eswaran, P. (2015). Generating comparative analysis of early stage prediction of Chronic Kidney Disease. *International Journal of Modern Engineering Research (IJMER)*, *5*(7), 49-55.

Chatterjee, S., Banerjee, S., Basu, P., Debnath, M., & Sen, S. (2017, April). Cuckoo search coupled artificial neural network in detection of chronic kidney disease. In *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)* (pp. 1-4). IEEE.

Chen, Z., Zhang, Z., Zhu, R., Xiang, Y., & Harrington, P. B. (2016). Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers. *Chemometrics and Intelligent Laboratory Systems*, *153*, 140-145.

Chen, Z., Zhang, X., & Zhang, Z. (2016). Clinical risk assessment of patients with chronic kidney disease by using clinical data and multivariate models. *International urology and nephrology*, *48*(12), 2069-2075.

John, G. H., & Langley, P. (1995, August). Estimating continuous distributions in Bayesian classifiers. In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence (pp. 338-345). Morgan Kaufmann Publishers Inc..

Zhang, H. (2004). The optimality of naive Bayes. AA, 1(2), 3.

Parthiban, G., & Srivatsa, S. K. (2012). Applying machine learning meth

Vineet Maheshwari. (2019, January 05). Retrieved from https://medium.com/datadriveninvestor/ensemble-learning-9e5924fd6567

Your Kidneys & How They Work (2018, June 01). Retrieved from https://www.niddk.nih.gov/health-information/kidney-disease/kidneys-how-they-work

How Your Kidneys Work (2017, March 10). Retrieved from https://www.kidney.org/kidneydisease/howkidneyswrk

Your Kidneys and How They Work. (n.d.). Retrieved from https://www.webmd.com/a-to-z-guides/function-kidneys

UCI Machine Learning Repository: Chronic_Kidney_Disease Data Set (2015, July 03). Retrieved from https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease

Appendix – A

Early stage of Indians Chronic Kidney Disease (CKD) Dataset

Feature Name	Description
Age	Age(numerical) age in years
Bp	Blood Pressure(numerical) bp in mm/Hg
Sg	Specific Gravity(nominal) sg - (1.005,1.010,1.015,1.020,1.025)
Al	Albumin(nominal) al - (0,1,2,3,4,5)
Su	Sugar(nominal) su - (0,1,2,3,4,5)
Rbc	Red Blood Cells(nominal) rbc - (normal,abnormal)
Pc	Pus Cell (nominal) pc - (normal,abnormal)
Pcc	Pus Cell clumps(nominal) pcc - (present,notpresent)
Ba	Bacteria(nominal) ba - (present,notpresent)
Bgr	Blood Glucose Random(numerical) bgr in mgs/dl
Bu	Blood Urea(numerical) bu in mgs/dl
Sc	Serum Creatinine(numerical) sc in mgs/dl
Sod	Sodium(numerical) sod in mEq/L

Pot	Potassium(numerical) pot in mEq/L
Hemo	Hemoglobin(numerical) hemo in gms
Pcv	Packed Cell Volume(numerical)
wc	White Blood Cell Count(numerical) wc in cells/cumm
Rc	Red Blood Cell Count(numerical) rc in millions/cmm
Htn	Hypertension(nominal) htn - (yes,no)
Dm	Diabetes Mellitus(nominal) dm - (yes,no)
Cad	Coronary Artery Disease(nominal) cad - (yes,no)
Appet	Appetite(nominal) appet - (good,poor)
Pe	Pedal Edema(nominal) pe - (yes,no)
Ane	Anemia(nominal) ane - (yes,no)
Class	Class (nominal) class - (ckd,notckd)

Table 5. 1 Dataset Description