George Mason University

CSI 703 Project Report

Analysis of Letter Recognition Dataset for Letter Classification

Submitted by

Ajay Kulkarni **G01024139**
Dagamawi Woldesenbet **G01047478**

Table of contents

Abstract 3

Chapter 1- Introduction 4

Chapter 2 – Exploratory Data Analysis and Data Preprocessing 6
- 2.1 Histograms
- 2.2 Box plots
- 2.3 Pearson's correlation

Chapter 3- Feature Selection and Model Selection 13
- 3.1 Bayesian Information Criterion
- 3.2 Boruta
- 3.3 Model Selection

Chapter 4 – Classification 16
- 4.1 Random Forest
- 4.2 Support Vector Machine (SVM)
- 4.3 KNN

Chapter 5 – Conclusion 21

References 22

Appendix – A (Exploratory Data Analysis using Shiny) 23

Appendix – B (Shiny and R scripts) 26

Abstract

Letter Recognition dataset consist of 20,000 data points and 17 variables. Among those 17 variables 'Letter' is the dependent variable. The main objective of the project was to identify the important features which majorly affect the classification of letters and then use those features to classify the letters. We started this project by cleaning the data and studying the data using different visualization techniques such as histograms, boxplots etc. After performing data preprocessing and exploratory data analysis we used Bayesian Information Criterion & Boruta for feature selection. Based on the selected features we built the model and we used the model to classify the letters. For classification purpose, we used three different methods Random Forest, SVM, and KNN. In the end, we calculated accuracies and decided which method is better for classification.

Chapter 1: Introduction

Letter Recognition dataset has been studied in this project to classify the letters based on the sixteen attributes. Letter Recognition dataset has been taken from the UCI Machine Learning Repository. Dataset consist of 20 different fonts and each letter within these 20 fonts were randomly distorted to produce file consist of 20,000 unique stimuli. Each stimulus was converted into 16 numerical attributes that were in turn submitted to classifier system. The 20 different fonts were designed by Dr. Allen V. Hershey, a mathematical physicist at U.S. Naval Weapons Laboratory.

Each item in character file was generated in the following manner. Twenty thousand calls were made to character-image generating program with random uniformly distributed parameter values for font type, letter of the alphabet, linear magnification, aspect ratio, horizontal and vertical "warp". Each character image was first produced in the form of vector coordinates of the end-points of its constituent line segments. The specified scale changes and "warping" were applied to these coordinates. The line segments were then "rasterized" to form a rectangular array of pixels, each of which was "on" or "off". The "on" pixels represented the image of the desired character. These arrays averaged about 45 pixels high by 45 pixels wide.

The linear magnification ranged from 1.0 to 1.6. The additional horizontal magnification, which changed the aspect ratio, ranged from 1.0 to 1.5. The horizontal warp parameter controlled a quadratic transformation of the horizontal coordinates that distorted the horizontal scale by stretching either the left or right region of the image. The vertical warp parameter operated similarly in the vertical direction. The range of the warp parameters was chosen so that even their values were at the limits of their range, the resulting character images, although rather misshapen, were fairly recognizable to humans.

Each character image was then scanned pixel by pixel, to extract 16 numerical attributes. These attributes represent primitive statistical features of the pixel distribution. To achieve compactness, each attribute was then scaled linearly to a range of integer values from 0 to 15. This final set of values was adequate to provide a perfect separation of the 26 classes. The 16 attributes are given below,

X1: The horizontal position, counting pixels from the left edge of the image, of the center of the smallest rectangular box that can be drawn with all "on" pixels inside the box.

X2: The vertical position, counting pixels from the bottom, of the above box.

X3: The width, in pixels, of the box.

X4: The height, in pixels, of the box.

X5: The total number of "on" pixels on the character image.

X6: The mean horizontal position of all "on" pixels relative to the center of the box and divided by the width of the box. This feature has a negative value if the image is "left heavy" as would be the case for the letter L.

X7: The mean vertical position of all "on" pixels relative to the center of the box and divided by the height of the box.

X8: The mean squared value of the horizontal pixel distances as measured in X6 above. This attribute will have a higher value for images whose pixels are more widely separated in the horizontal direction as would be the case for the letters W or M.

X9: The mean squared value of the vertical pixel distances as measured in X7 above.

X10: The mean product of the horizontal and vertical distances for each "on" pixel as measured in X6 and X7 above. This attribute has a positive value for diagonal lines that run from bottom left to top right and a negative value for diagonal lines from top left to bottom right.

X11: The mean value of the squared horizontal distance times the vertical distances for each "on" pixel. This measures the correlation of the horizontal variance with the vertical position.

X12: The mean value of the squared vertical distance times the horizontal distance for each "on" pixel. This measures the correlation of the vertical variance with the horizontal position.

X13: The mean number of edges encountered when making systematic scans from left to right at all vertical positions within the box. This measure distinguished between letters like "W" or "M" and letters like "I" or "L".

X14: The sum of the vertical positions of edges encountered as measured in X13 above. This feature will give a higher if there are more edges at the top of the box, as in the letter "Y".

X15: The mean number of edges encountered when making systematic scans of the image from bottom to top over all horizontal positions within the box.

X16: The sum of horizontal positions of edges encountered as measured in X15 above.

To analyze the Letter Recognition dataset, we started with the Exploratory data analysis and data preprocessing which is explained in chapter 2. After preprocessing we selected important features and built the model. The details about feature selection and model building are explained in chapter 3. Further, we used selected model to classify 26 letters using different classification algorithms. The details about classification are explained in chapter 4. The findings of this analysis are concluded in chapter 5 of the report.

Chapter 2: Exploratory Data Analysis and Data Preprocessing

Exploratory Data Analysis and Data Preprocessing mainly deals with checking the quality of the data and make the data ready for further analysis. In data preprocessing we analyzed the data for missing and null values. In addition to that, we also determined outliers in the data. The data was present in a "Comma-Separated Values" (CSV) format and we used R for analyzing the data. All the variables in the data are plotted using histograms and boxplots. Further, we also have calculated correlations among 16 attributes using Pearson's correlation method to enhance the understanding of the attributes. The general structure of data is shown in figure 1 below,

```
> head(mydata)
  Letter X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16
1      T  2  8  3  5  1  8 13  0  6   6  10   8   0   8   0   8
2      I  5 12  3  7  2 10  5  5  4  13   3   9   2   8   4  10
3      D  4 11  6  8  6 10  6  2  6  10   3   7   3   7   3   9
4      N  7 11  6  6  3  5  9  4  6   4   4  10   6  10   2   8
5      G  2  1  3  1  1  8  6  6  6   6   5   9   1   7   5  10
6      S  4 11  5  8  3  8  8  6  9   5   6   6   0   8   9   7
```

Figure 1: Structure of Letter Recognition data set

From the preliminary analysis in R, it became clear that data doesn't have any null or missing values. So we proceeded forward and studied histograms as well as boxplots. Details about histograms and boxplots are given below.

2.1 Bar chart and Histograms

Histograms are graphical representation of distribution of the numerical data. The purpose of histogram is to graphically summarize the distribution of a data. We have used histograms for studying the spread and skewness of the data. The bar chart for all the variables is given in figure 2.

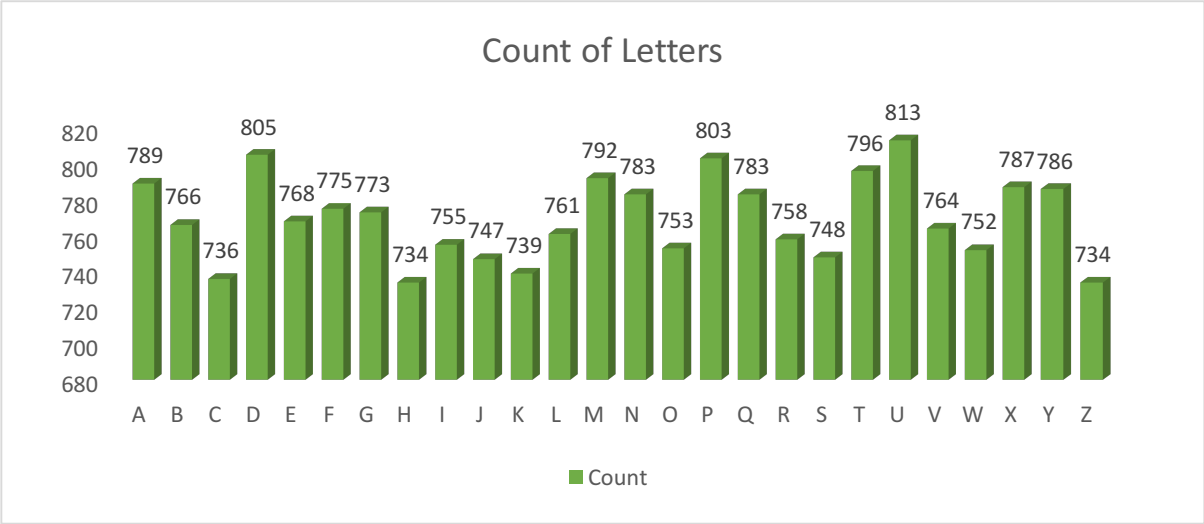


Figure 2: Total count of letters

The purpose of this bar chart (figure 2) is to understand the data and to know the count of the letters. It can be observed from figure 2 that count of U (count = 813) is more in the data as compared to any other letters. In addition to that, it also can observe that H's and Z's have least count which is about 734.

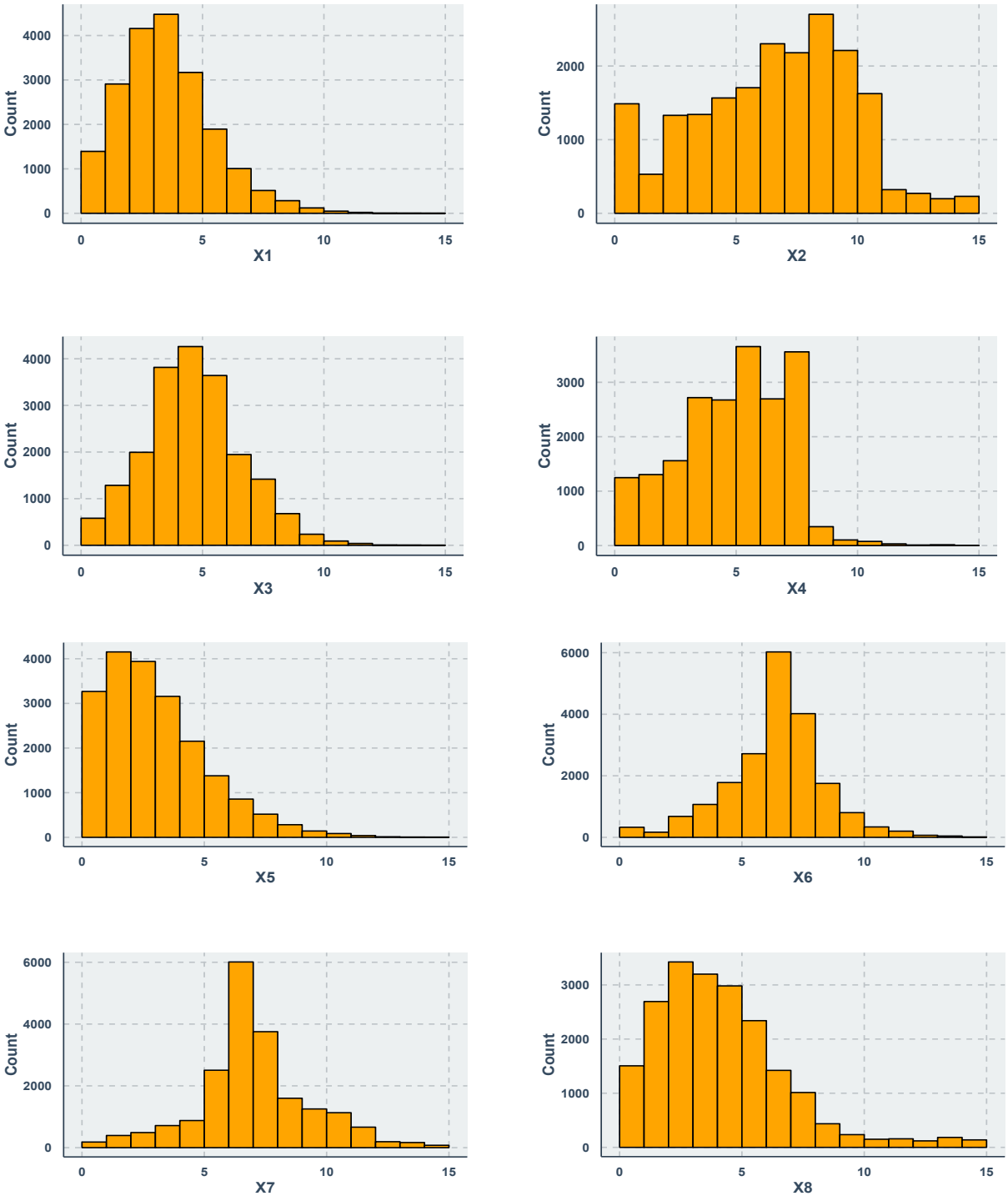


Figure 3: Histograms for X1-X8 attributes

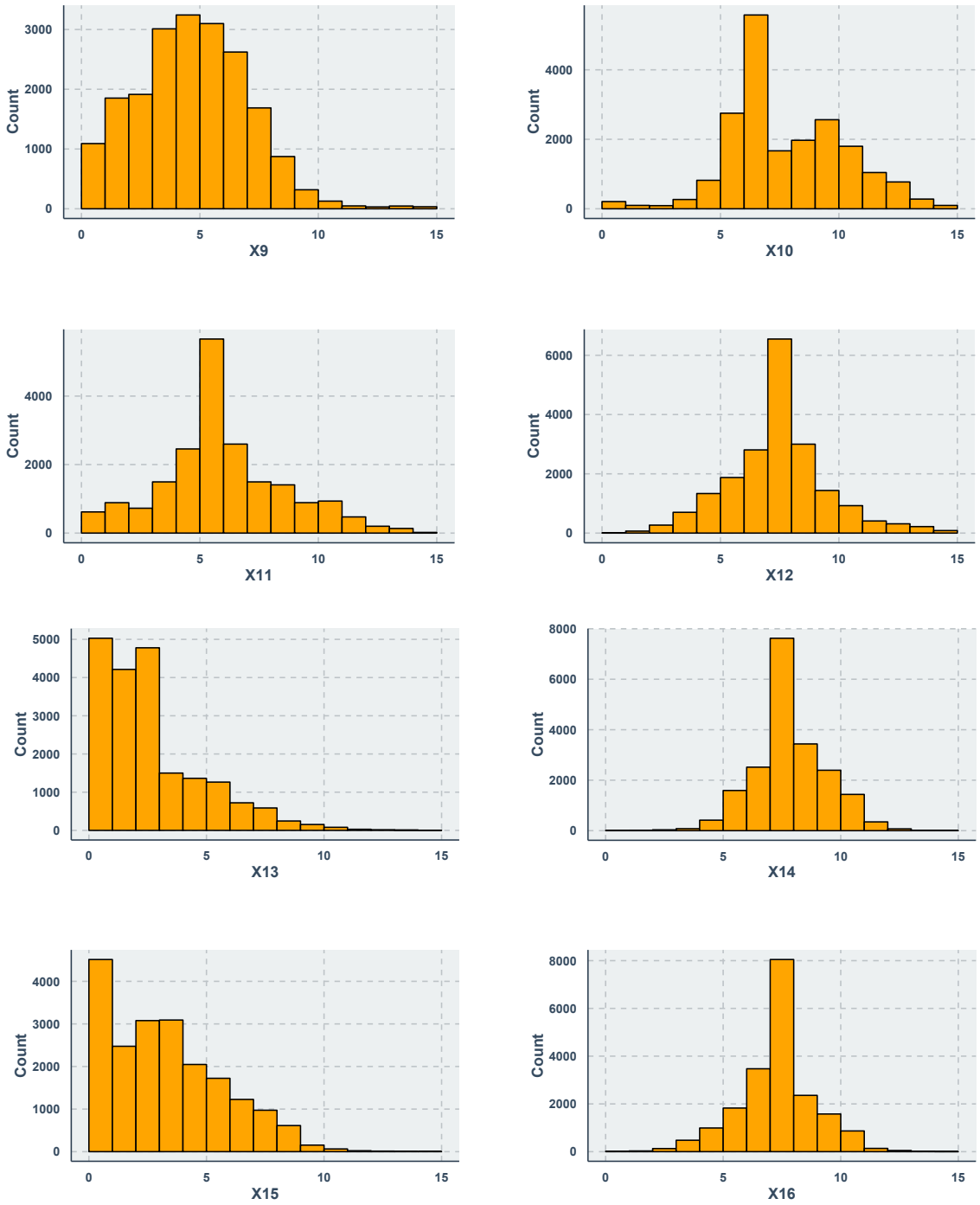


Figure 4: Histograms for X8-X16 attributes

It can be observed from figure 3 and figure 4 that majority of the attributes are right skewed and no attribute can be found where data is left skewed. The attributes X1, X2, X3, X4, X5, X8, X9, X13, X14, and X15 are right skewed and X11 is slightly right skewed. Other attributes such as X6, X7, X10, X14 and X16 are almost symmetric. While it is also observed that X12 is symmetric.

2.2 Boxplots

Boxplots are excellent tools for understanding the range of the data and detecting the outliers. The boxplots also explain about the quartiles, mean and median of the data. We used boxplots for aforementioned purpose. Boxplots for all the variables are given below,

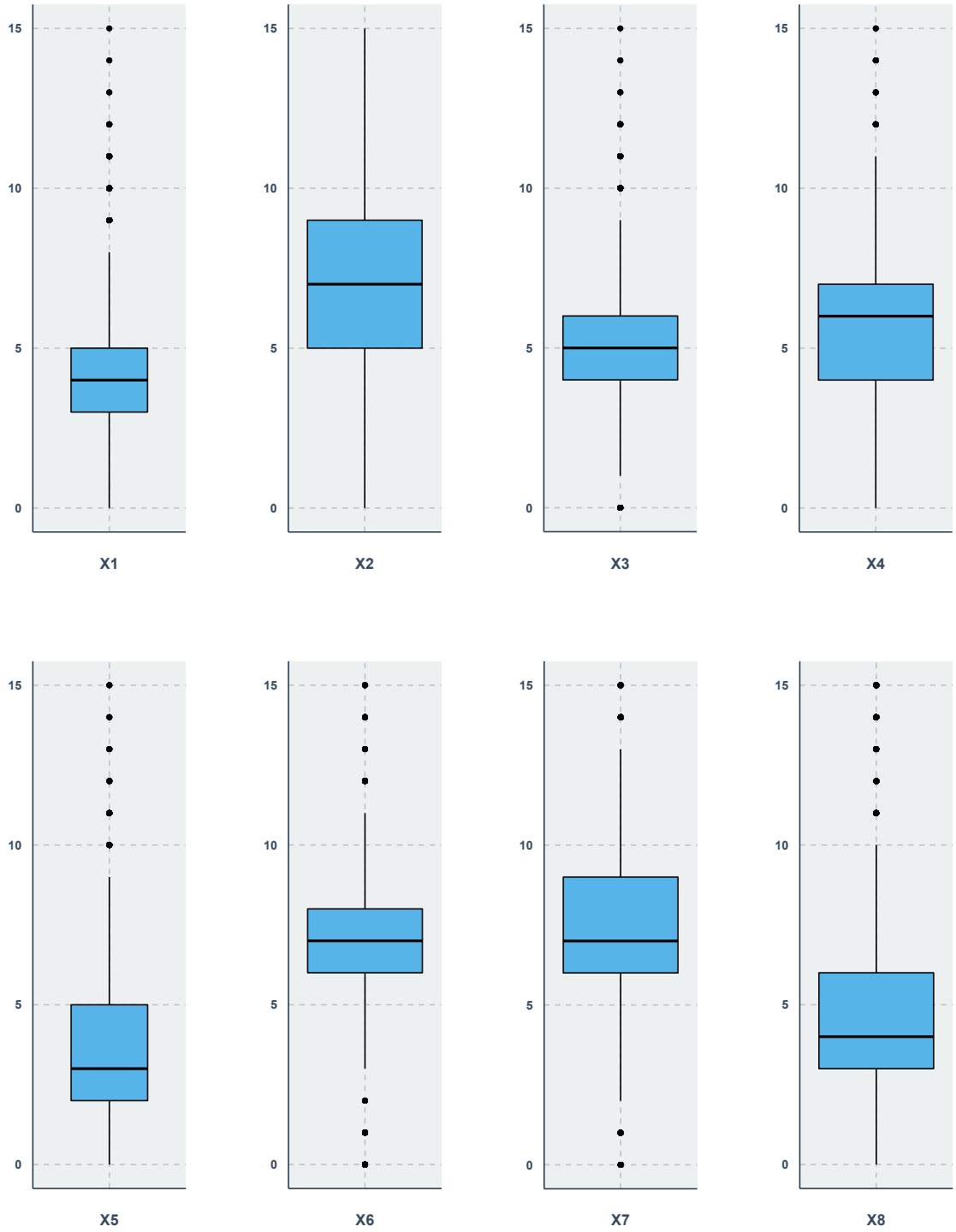


Figure 5: Boxplots for X1-X8 attributes

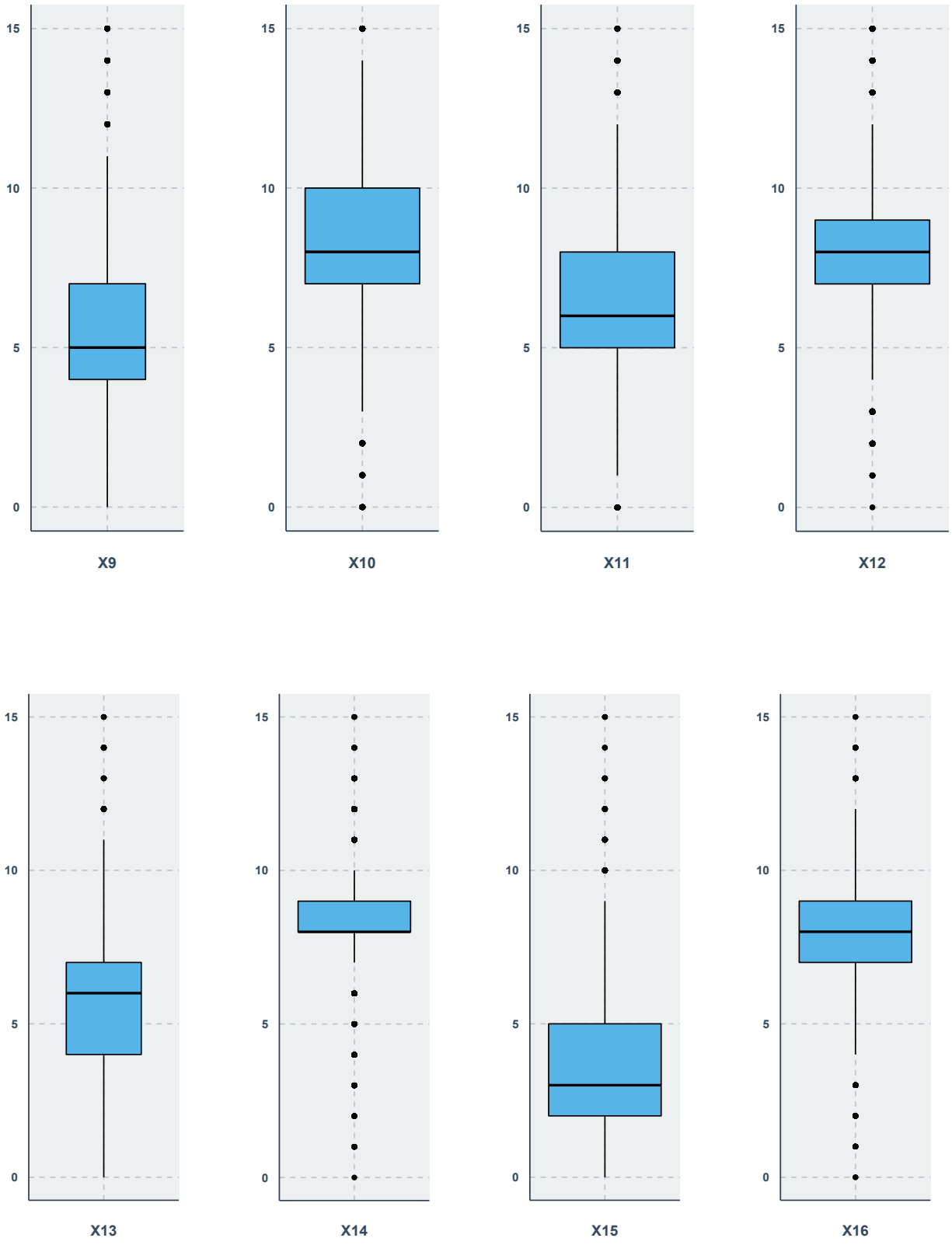


Figure 6: Boxplots for X8-X16 attributes

From the above plots (figure 5 and figure 6) it can be observed that the count of outliers is less. It is also observed that for attributes X1, X3, X4, X5, X8, X9, X13 and X15 outliers are present on the higher end of the plot. For other variables except X2 outliers are present on both sides of the plot and for X2 no outliers are observed. From the above plots, we can say that the outliers are less and thus, we decided to use the data without removal of outliers. The summary of the data from boxplots can be given as follows:

Parameters	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
X1	0.000	3.000	4.000	4.024	5.000	15.000
X2	0.000	5.000	7.000	7.035	9.000	15.000
X3	0.000	4.000	5.000	5.122	6.000	15.000
X4	0.000	4.000	6.000	5.327	7.000	15.000
X5	0.000	2.000	3.000	3.506	5.000	15.000
X6	0.000	6.000	7.000	6.898	8.000	15.000
X7	0.000	6.000	7.000	7.500	9.000	15.000
X8	0.000	3.000	4.000	4.629	6.000	15.000
X9	0.000	4.000	5.000	5.179	7.000	15.000
X10	0.000	7.000	8.000	8.282	10.000	15.000
X11	0.000	5.000	6.000	6.454	8.000	15.000
X12	0.000	7.000	8.000	7.929	9.000	15.000
X13	0.000	1.000	3.000	3.046	4.000	15.000
X14	0.000	8.000	8.000	8.339	9.000	15.000
X15	0.000	2.000	3.000	3.692	5.000	15.000
X16	0.000	7.000	8.000	7.801	9.000	15.000

Table 1: Summary of data

2.3 Pearson's correlation

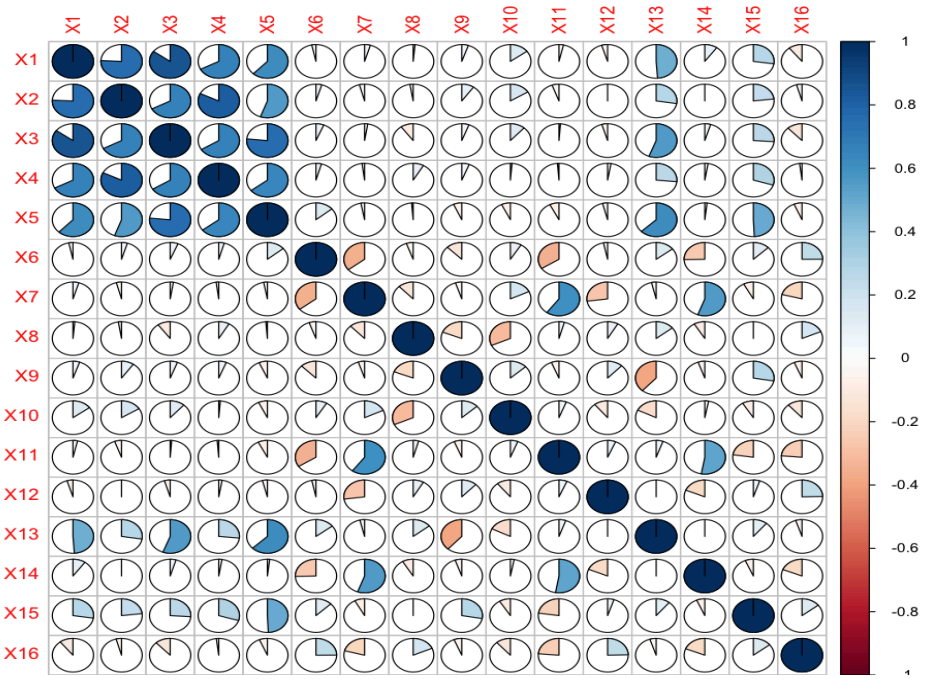


Figure 7: Correlation plot

The main purpose of the Pearson's correlation is to get a brief idea about the relationship of one variable with other variables. The correlation plot for all the variables is given in figure 7. The plot will help us to understand the dependency between the variables. In the above plot red color indicates negative correlation and blue color indicates positive correlation.

It can be observed from figure 7 that X1 have significant positive correlations with X2, X3, X4, X5, and X13. Similarly, for X2 significant positive correlations can be observed with X1, X3, X4, X5. For X3 significant positive correlations can be observed with X1, X2, X4, X5 and X13. For X4 significant positive correlations can be observed with X1, X2, X3 and X5. For X5 significant positive correlations can be observed with X1, X2, X3, X4, X13 and X15. For X7 significant positive correlations are observed with X11 and X14. X11 is showing significant positive correlation with X14. For X13 positive correlations can be observed with X1, X3, and X5. X14 shows significant positive correlations with X8 and X11. X15 shows significant positive correlation with X5. Negative correlations are also observed for attributes X6, X7, X8, X9, X10, X11 X12 and X13.

Chapter 3: Feature Selection and Model Selection

Feature selection is the process of selecting different relevant parameters or variables which are most important to the data. Feature selection is also called as variable selection or attribute selection. Feature selection techniques are used for the three reasons:

- Simplification of models to make them easier to interpret.
- Shorten training time.
- Enhanced generalization by reducing overfitting.

The main theme behind the feature selection is that there are many features present in the data. Some of them are irrelevant or redundant and if we remove those features it will not result in much loss of information. But if we select the most important features of the data it can be used for building the good predictive model.

For Letter Recognition data set 16 features are present in the data and we need to select the features which majorly affect the classification of letters. For feature selection, we have selected features based on BIC and then verified it by using Boruta. The details about feature selection are as follows,

3.1 Bayesian Information Criterion (BIC)

Bayesian Information Criterion (BIC) is a criterion for model selection among a finite set of models with the lowest BIC is preferred. BIC was developed by Gideon E. Schwarz and published in a 1978 paper where he gave Bayesian argument for adopting it. The BIC is formally defined as,

$$BIC = \ln(n)\, k - 2\ln(\hat{L})$$

Where,
$\hat{L}$ is the maximized value of the likelihood function of the model M,
n is the number of data points in x (the observed data),
k is the number of free parameters to be estimated.

The BIC plot in R is constructed using *"regsubsets"* command from the *"leaps"* package. In BIC plot each row represents a model. The shaded rectangles in the columns indicate the variables included in the given model. The number on the left margin are the values of Schwartz' Bayesian Information Criterion. The darkness of the shading simply represents the ordering of the BIC values. The BIC plot for the Letter Recognition dataset is given below,

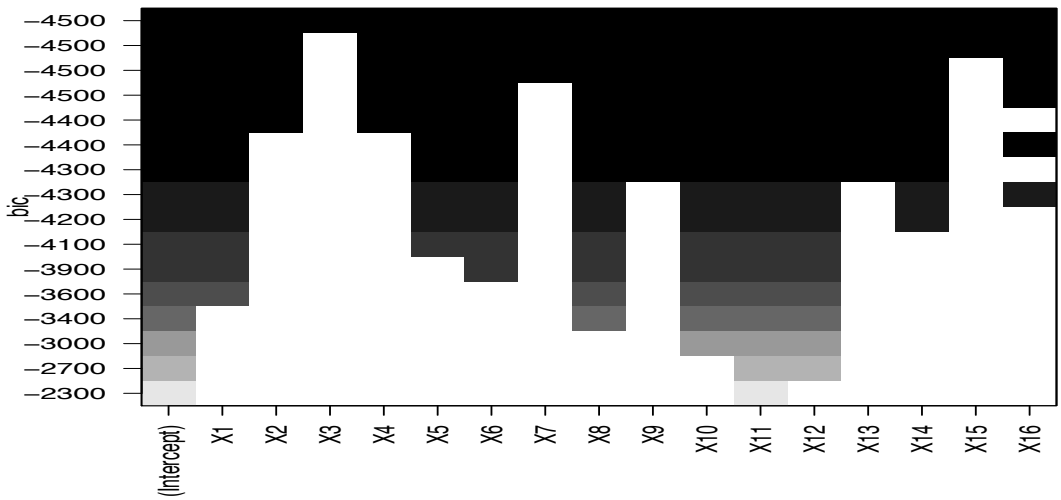


Figure 8: BIC plot

Figure 8 shows the BIC plot for the Letter Recognition dataset. In the plot, black color indicates the most important variables and white color indicates the least important variables. It can be observed from the plot that all the variables are important for model selection. The BIC is the minimum for the model containing all the features of the data. To verify this, we used one more method Boruta and details about that are given below.

3.2 Boruta

Boruta algorithm is a wrapper built around the Random Forest classification algorithm implemented in the R package randomForest. The method performs a top-down search for relevant features by comparing original attributes importance with importance achievable at

```
>
> output <- Boruta(lenum ~ X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16, data=t1,doTrace=2)
 1. run of importance source...
 2. run of importance source...
 3. run of importance source...
 4. run of importance source...
 5. run of importance source...
 6. run of importance source...
 7. run of importance source...
 8. run of importance source...
 9. run of importance source...
 10. run of importance source...
 11. run of importance source...
Confirmed 16 attributes: X1, X10, X11, X12, X13 and 11 more.
> output
Boruta performed 11 iterations in 3.593628 mins.
 16 attributes confirmed important: X1, X10, X11, X12, X13 and 11 more.
 No attributes deemed unimportant.
>
```

Figure 9: Boruta feature selection result

random, estimated using their permuted copies and progressively eliminating irrelevant features to stabilize that test. We have used this function for variable selection and result obtained from Boruta is given above. From the above result, it became clear that all the

variables from X1 to X16 are very important for classifying the letters present in the dataset.

3.3 Model Selection

Model building and selection are completely dependent on the feature selection. We have used BIC criterion and Boruta for feature selection. Both the methods have given similar results i.e. all the sixteen variables are important. So, we have used all the sixteen variable in our model and selected model is given as follows,

Selected Model

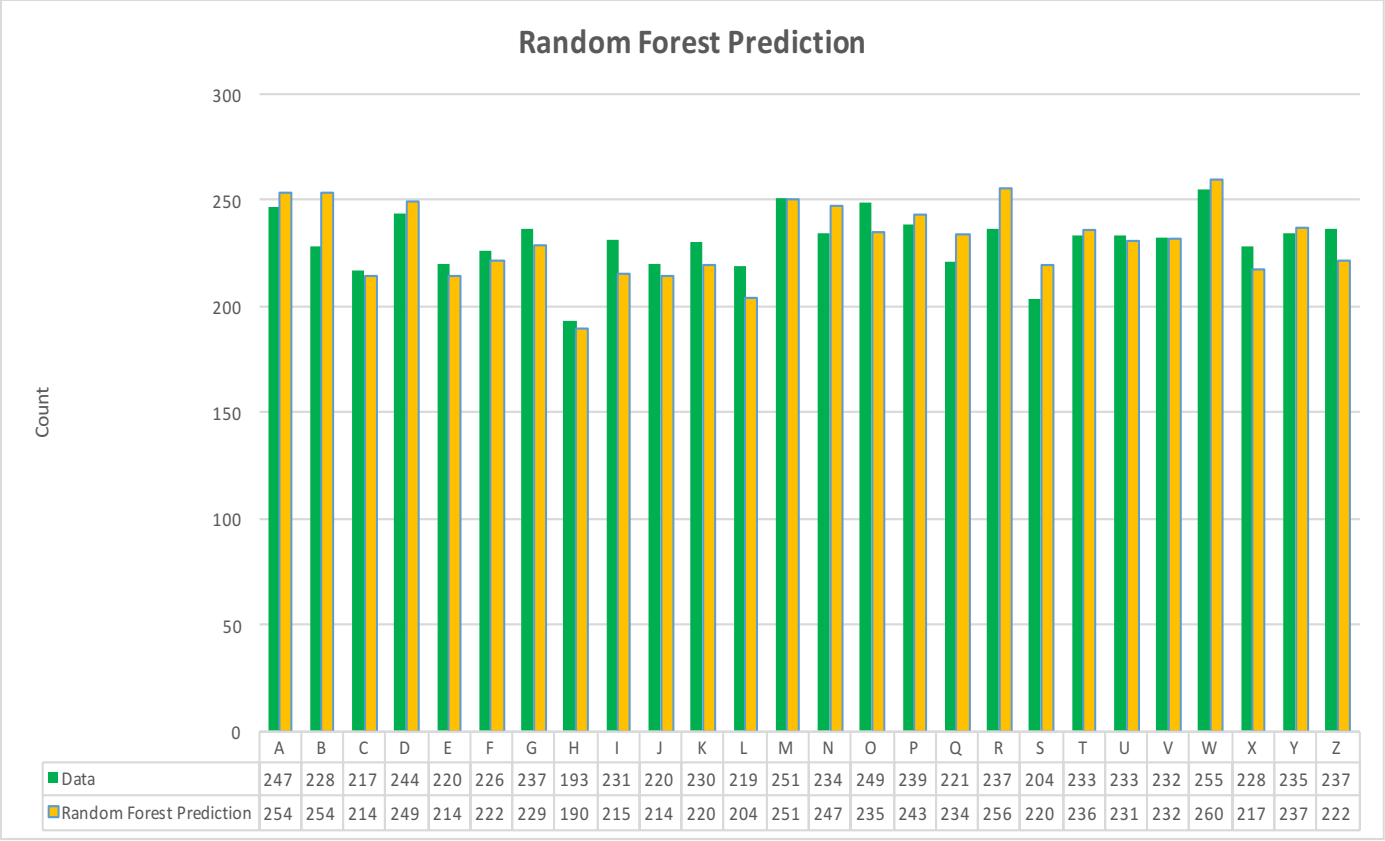
Letter ~ X1 + X2 + X3 + X4 + X5 +X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15 + X16

Chapter 4 – Classification

Classification is the problem of identifying to which of a set of categories a new observation belongs on the basis of a training set of data whose category membership is known. The goal of the classification is to accurately predict the target class of each case in the data. To implement classification, we divided data into two parts i.e. training data (70%) and testing data (30%). The training data was used by the model to train 70% of the data and remaining 30% of the data is used by the model for testing purpose. Different classification methods have been implemented on the selected model. The training data consist of 14,000 data points and testing data consist of 6,000 data points. In prediction plots, green color indicates the testing data and yellow color indicates the predicted data generated by an algorithm. Also, in accuracy plots black color indicates the accuracy for the specific letter and red color indicates the overall accuracy of the model.

4.1 Random Forest

Random Forest is the algorithm which we tested for classification of letters. Random Forest is a computationally expensive algorithm and because of our computational limitations, we are able to produce only 10 trees in the Random Forest. The result obtained from the Random Forest are given below



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Data	247	228	217	244	220	226	237	193	231	220	230	219	251	234	249	239	221	237	204	233	233	232	255	228	235	237
Random Forest Prediction	254	254	214	249	214	222	229	190	215	214	220	204	251	247	235	243	234	256	220	236	231	232	260	217	237	222

Figure 10: Random Forest Prediction for all letters

The purpose of the above plot is to know the classification of letters from Random Forest. The above classification is the total number of letters predicted by Random Forest and not the correct letters predicted by Random Forest. It can be observed that Random Forest is able to predict 251 M's which are equal to the total number of M's present in the testing data. But the correct number of predicted M's by Random Forest is 226. The accuracy of Random Forest for predicting accurate letters is given below,

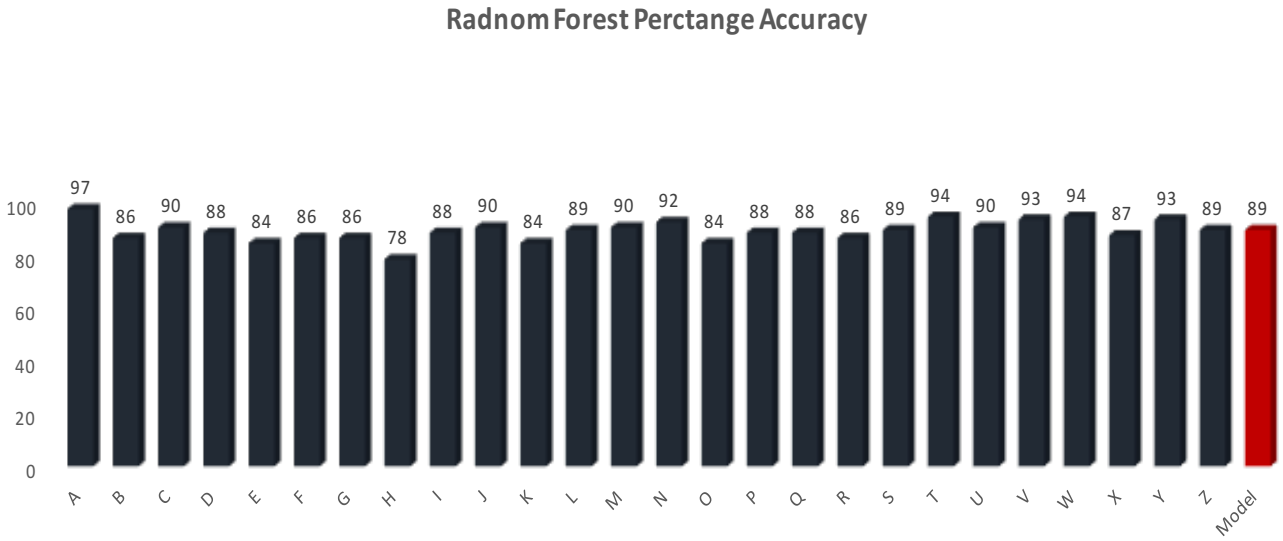


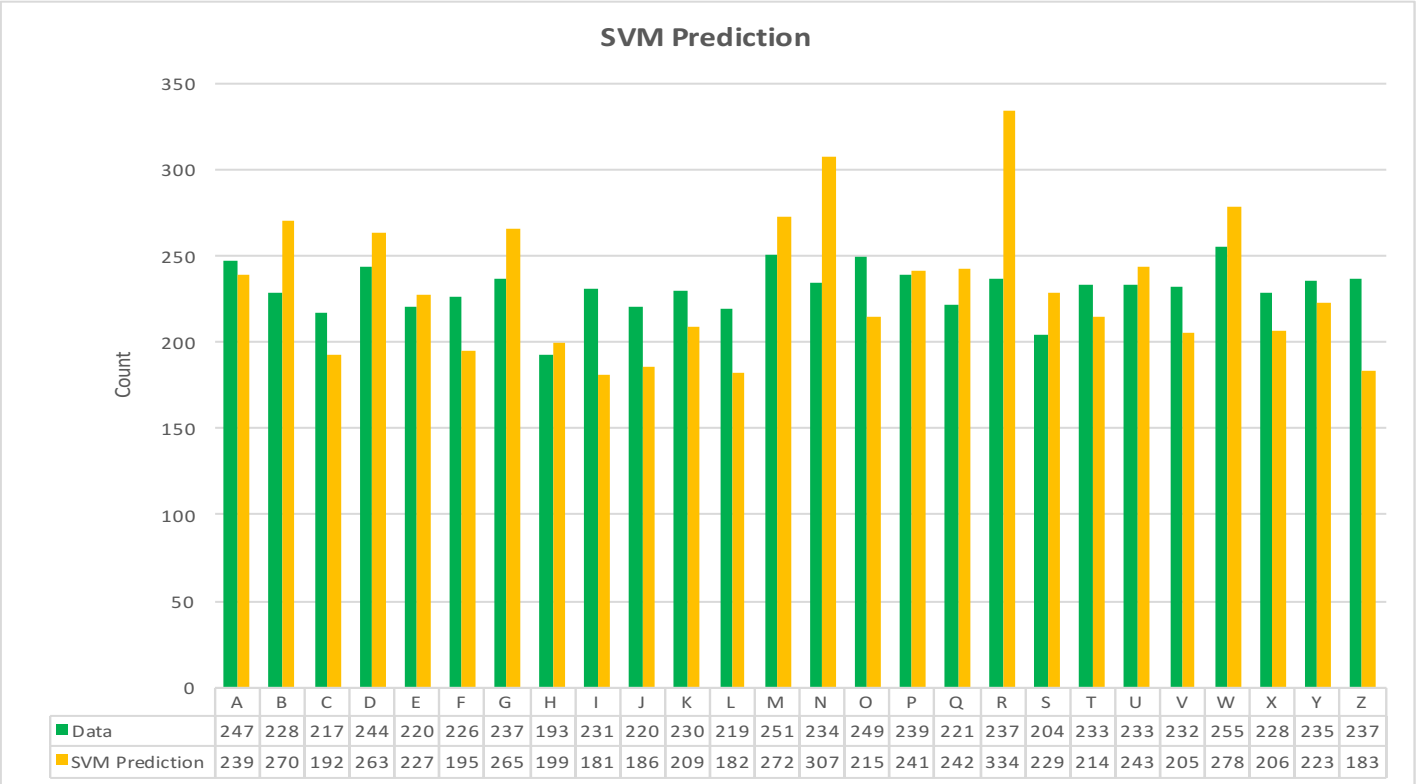
Figure 11: Random Forest Prediction Accuracy

It can be observed from the above figure that overall accuracy of Random Forest for the selected model is about 89%. It can also be observed that Random Forest is able to predict 97% correct A's which is the maximum correct accuracy for any letter. Also, the least accuracy of Random Forest is observed for H which is about 78%.

4.2 Support Vector Machine (SVM)

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. We have used SVM here for the classification purpose. The results obtained from SVM are given on next page.

It can be observed from figure 12 that SVM is predicting more number of R's, N's as well as some other letters. The results obtained from SVM are showing less accuracy as compared to Random Forest. The overall accuracy of the SVM can be observed from figure 13 is about 76% which is less as compared to Random Forest. Also, the least accuracy of SVM is observed for letter S which is about 66% and most accuracy is observed for letter A which is 87%.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Data	247	228	217	244	220	226	237	193	231	220	230	219	251	234	249	239	221	237	204	233	233	232	255	228	235	237
SVM Prediction	239	270	192	263	227	195	265	199	181	186	209	182	272	307	215	241	242	334	229	214	243	205	278	206	223	183

Figure 12: SVM Prediction results

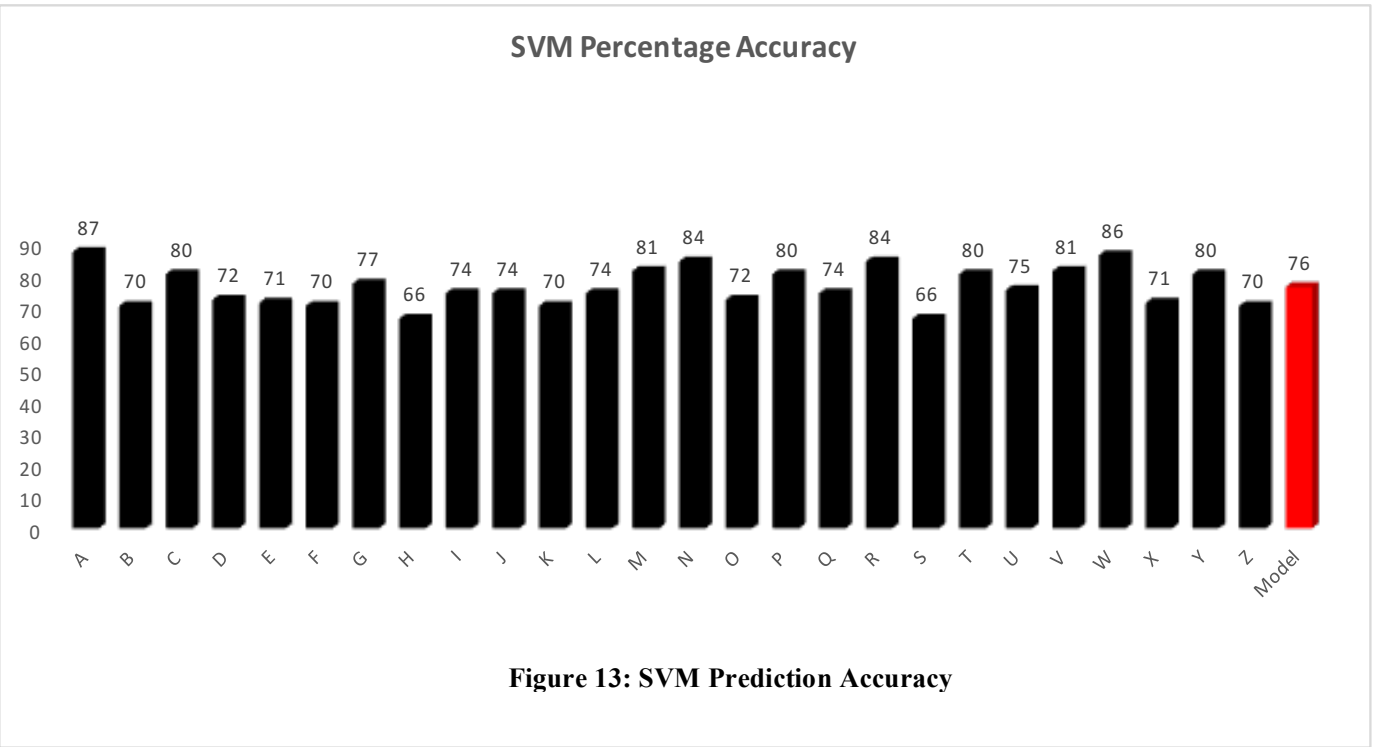


Figure 13: SVM Prediction Accuracy

4.3 KNN (K-Nearest Neighbors)

KNN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. When KNN is used for classification, the output can be calculated as the class with the highest frequency from the K-most similar instances. Each instance is essence votes for their class and the class with the most votes is taken as the prediction. To select the value for K we selected values from 1 to 10 for K and then we calculated the error. The plot for the error rate is given below.

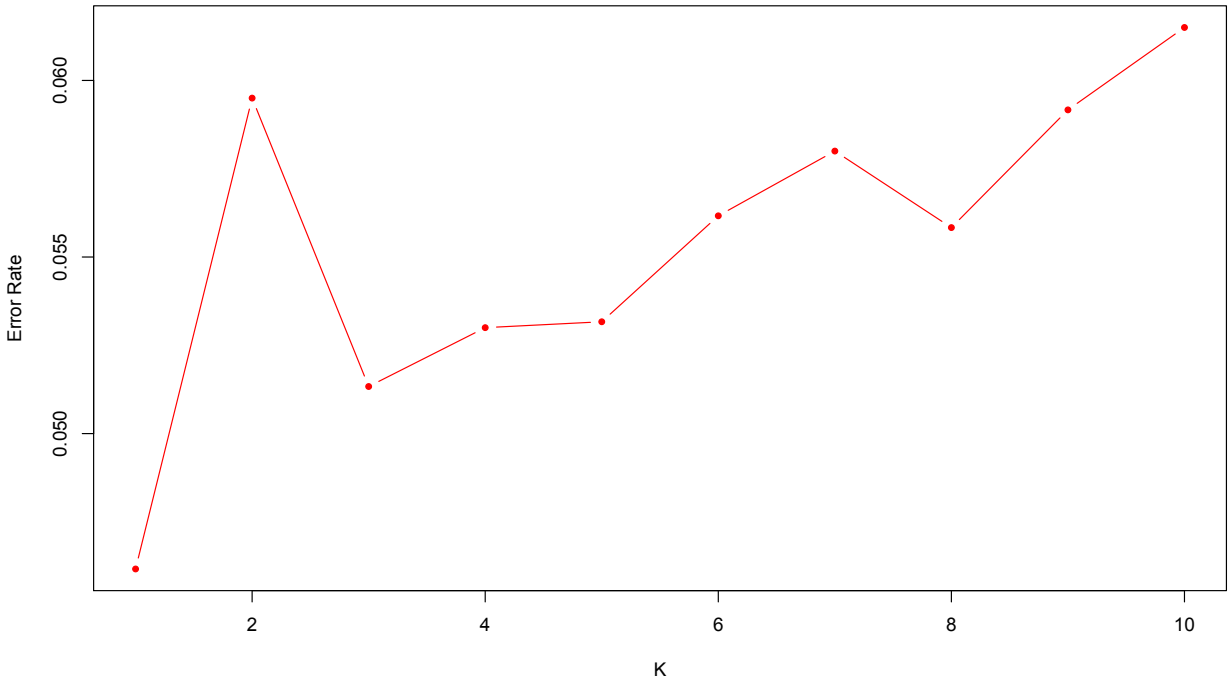
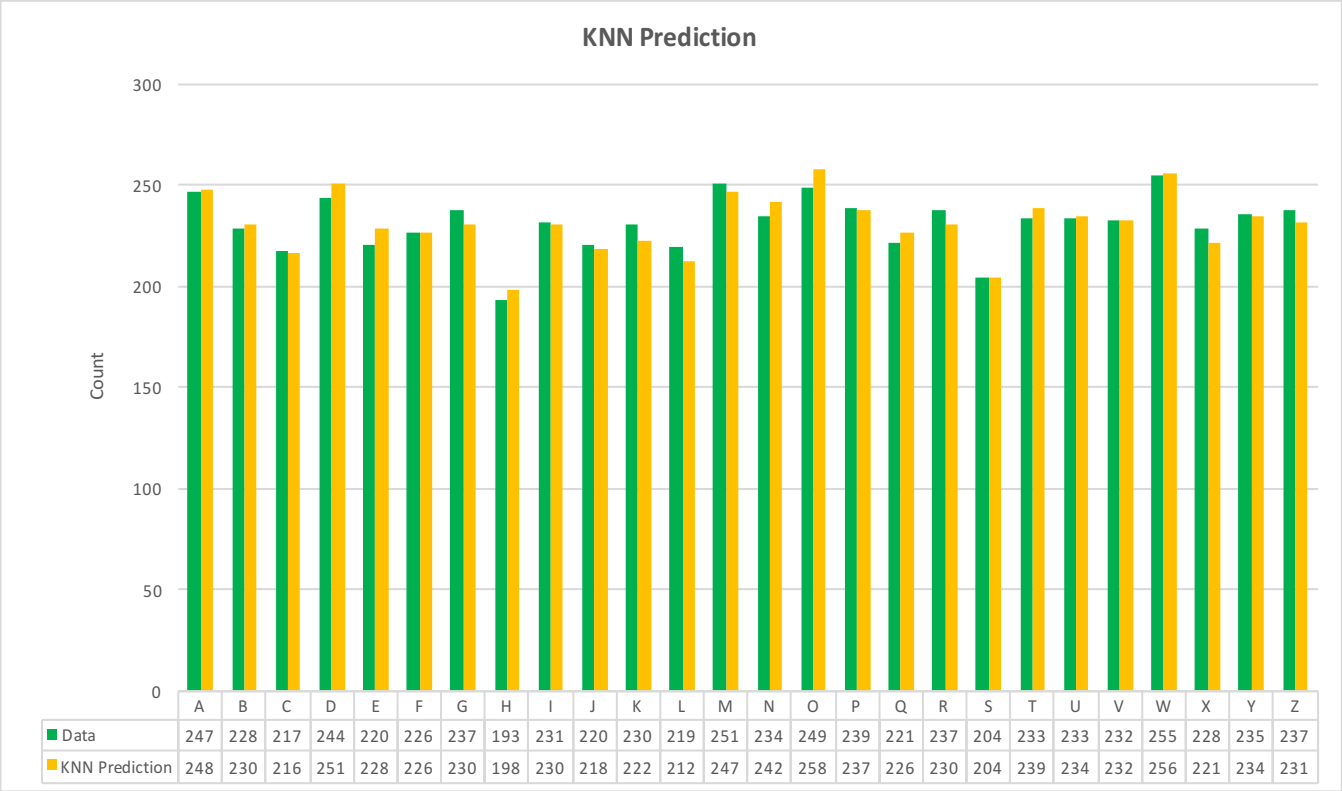


Figure 14: Error rate for KNN

It can be observed that error rate is increasing as the value for K is increasing except for K = 3. For K= 10 error rate is maximum and for K = 1 error rate is minimum. So we decided to use K = 1 for our prediction. The results obtained from the KNN are given on the next page. It can be observed from the results that the prediction accuracy of KNN is better as compared to other two algorithms. The overall accuracy of the model using KNN is about 95%. KNN is able to predict about 99% correct A's and least accuracy is observed for H which is about 89%.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Data	247	228	217	244	220	226	237	193	231	220	230	219	251	234	249	239	221	237	204	233	233	232	255	228	235	237
KNN Prediction	248	230	216	251	228	226	230	198	230	218	222	212	247	242	258	237	226	230	204	239	234	232	256	221	234	231

Figure 15: KNN Prediction results

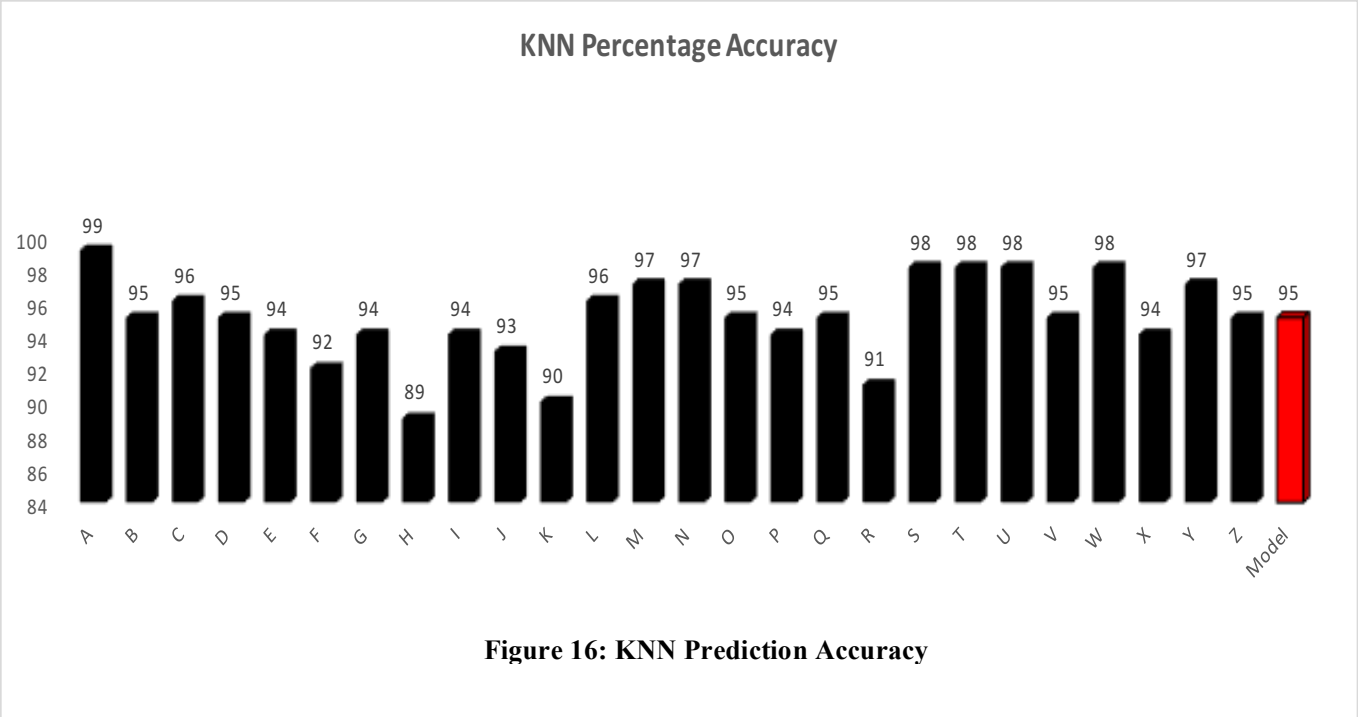


Figure 16: KNN Prediction Accuracy

Chapter 5: Conclusion

Letter Recognition data set has been successfully analyzed by finding important features in the data and then classifying the data based on different classification algorithms. We started feature selection using Bayesian Information Criterion and Boruta. Both the techniques indicated that all the variables are important for building the model.

After selecting all the features, we built the model and we compared three classification algorithms which are Random Forest, SVM, and KNN. It can be observed that overall accuracy of the Random Forest is about 89% and it is able to predict 97% correct A's which is the highest prediction accuracy for any letter. For SVM the overall prediction accuracy observed is around 76% and the SVM is able to predict 87% of correct A's. In the case of KNN, we used value K = 1 based on the error rate. The overall accuracy observed for the KNN is about 95% and KNN is able to predict 99% of correct A's.

Thus, we concluded that all the 16 variables are important for classifying the letters in Letter Recognition data set. We also conclude that KNN has better accuracy as compared to Random Forest and SVM. KNN is also able to classify 99% A's correctly which is the greater percentage of prediction for A as compared to any other letter.

References

1. P. W. Frey and D. J. Slate. "Letter Recognition Using Holland-style Adaptive Classifiers". (Machine Learning Vol 6 #2 March 91)

2. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] Irvine, CA: University of California, School of Information and Computer Science.

3. https://www.r-bloggers.com/predicting-wine-quality-using-random-forests/

4. https://www.r-bloggers.com/correlation-and-linear-regression/

5. http://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/a-comparison-of-the-pearson-and-spearman-correlation-methods/

6. https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

7. http://www2.hawaii.edu/~taylor/z632/Rbestsubsets.pdf

8. http://www.svm-tutorial.com/2014/10/support-vector-regression-r/

9. https://en.wikipedia.org/wiki/Bayesian_information_criterion

10. https://www.r-bloggers.com/feature-selection-all-relevant-selection-with-the-boruta-package/

11. http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.htm

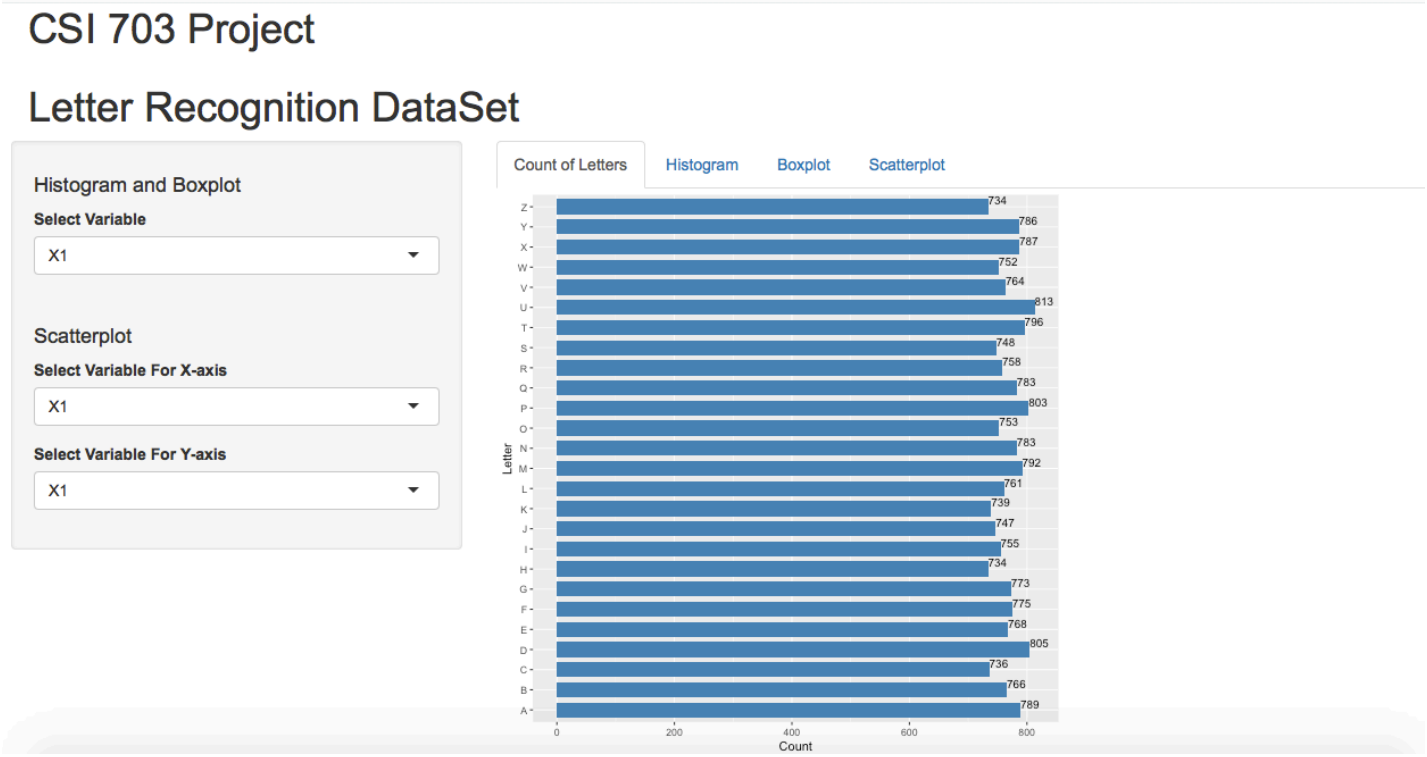
12. http://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

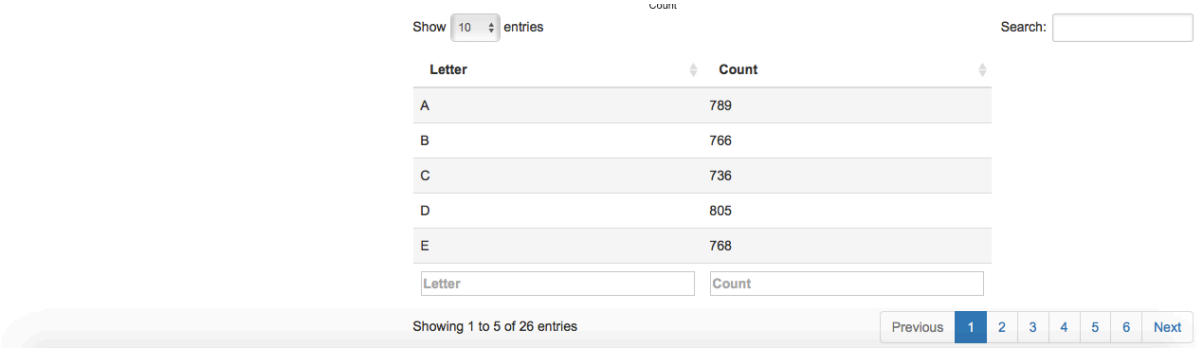
13. http://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/

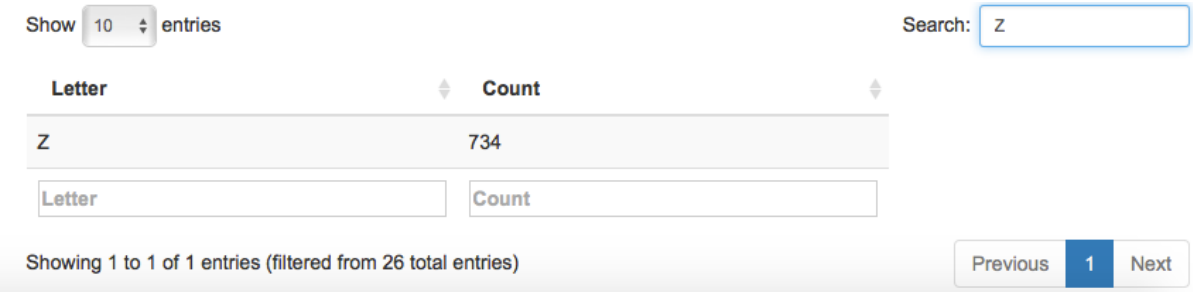
14. http://www.statsoft.com/Textbook/k-Nearest-Neighbors/

Appendix – A (Exploratory Data Analysis using Shiny)

- **Bar chart and Interactive Table**







- **Interactive Histograms and Boxplots**

CSI 703 Project

Letter Recognition DataSet

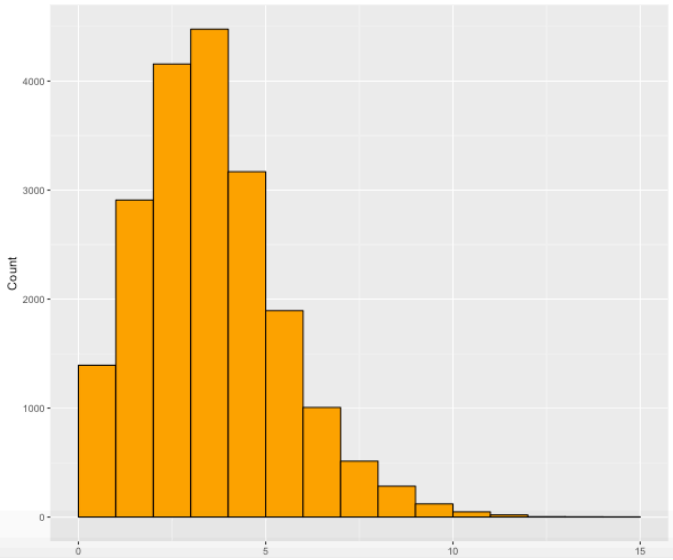
Histogram and Boxplot

Select Variable

X1

X2
X3
X4
X5
X6
X7
X8
X9

Count of Letters | Histogram | Boxplot | Scatterplot



Histogram and Boxplot

Select Variable

X5

Scatterplot

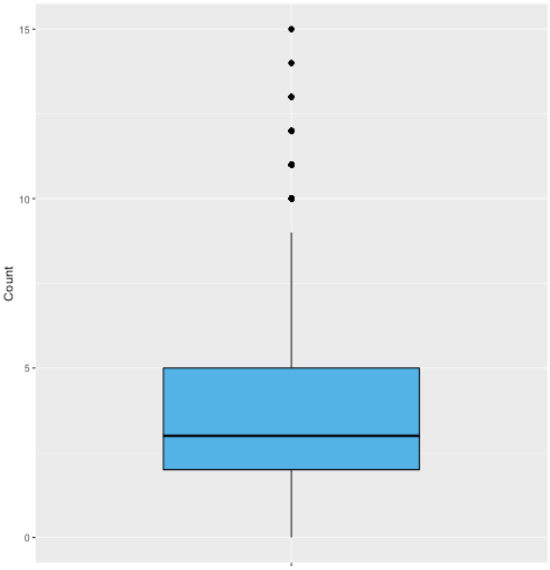
Select Variable For X-axis

X1

Select Variable For Y-axis

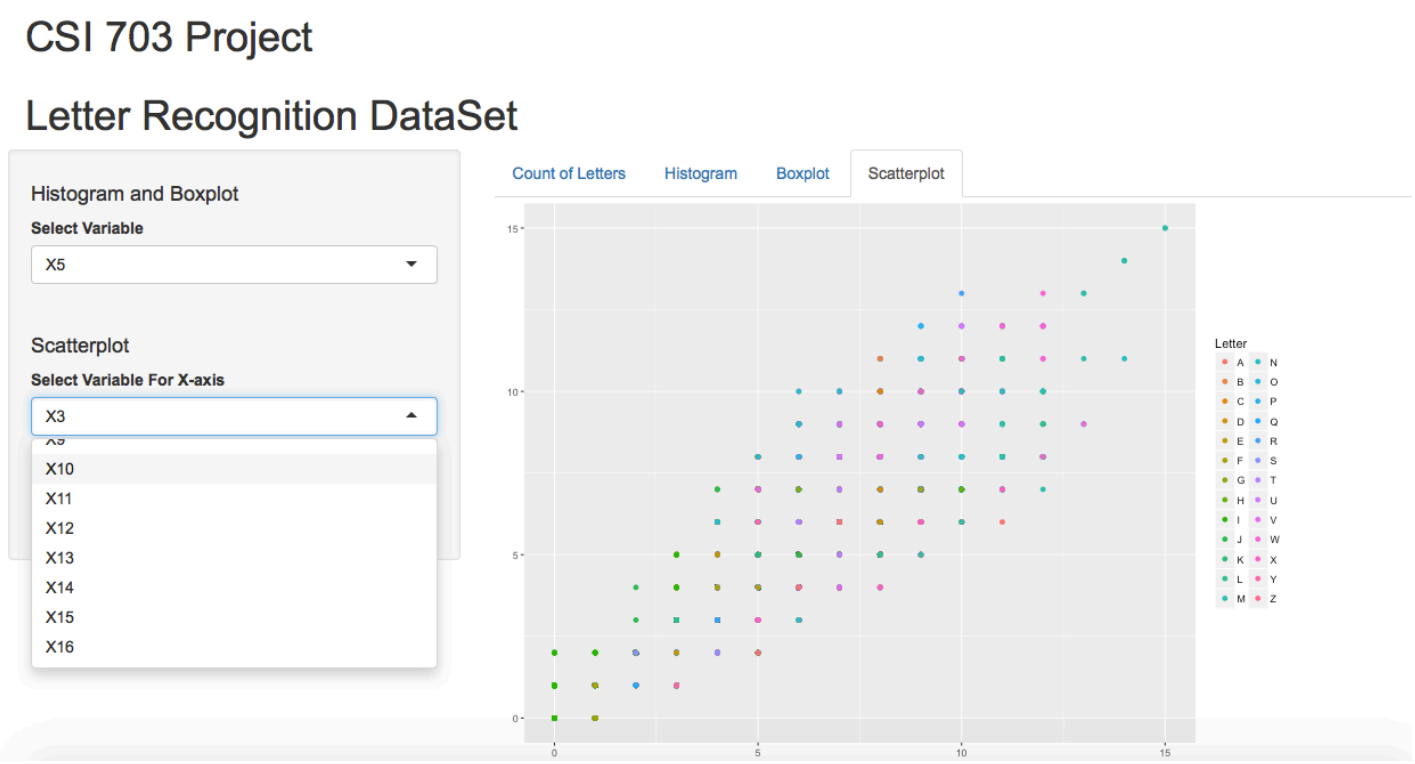
X1

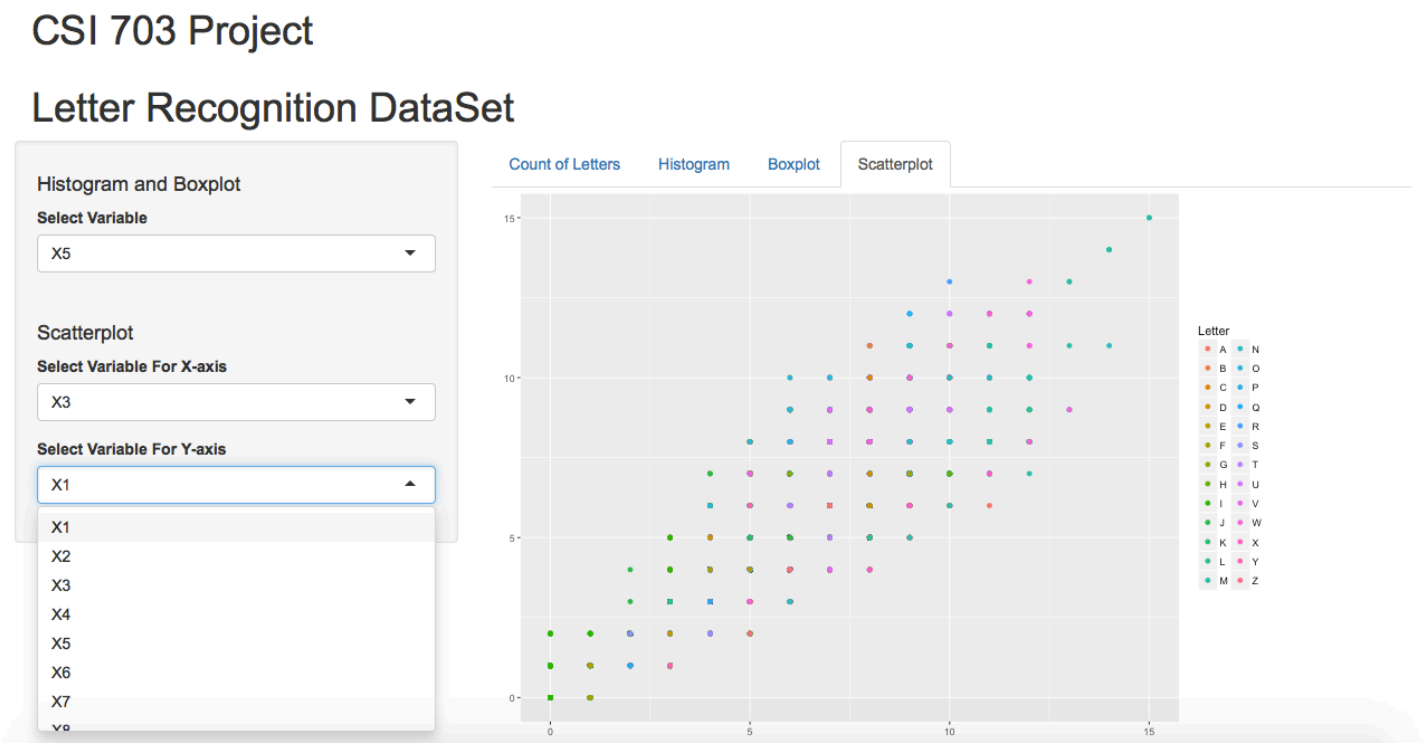
Count of Letters | Histogram | Boxplot | Scatterplot




```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000   2.000   3.000   3.506   5.000  15.000
```

- **Interactive Scatterplot**





Appendix – B (Shiny and R Scripts)

- **Shiny code**

```
library(shiny)
library(ggplot2)
library(corrplot)

ed<-read.csv("Master Data.csv", stringsAsFactors = FALSE)

df <- data.frame(Letter = c("A","B","C","D","E","F","G","H","I",
                            "J","K","L","M","N","O","P","Q","R",
                            "S","T","U","V","W","X","Y","Z"),
                 Count = c(789,766,736,805,768,775,773,734,755,747,
                           739,761,792,783,753,803,783,758,748,796,
                           813,764,752,787,786,734)
)

ui<-fluidPage (

  tags$head(tags$style(
    HTML('
         sidebar {
         background-color: blue;
         }

         body, label, input, button, select {
         font-family: "Arial";
         }')
  )),

  headerPanel("CSI 703 Project"),
  headerPanel("Letter Recognition DataSet"),
  sidebarLayout(
    sidebarPanel(
      h4("Histogram and Boxplot"),
      selectInput("var", "Select Variable", choices=colnames(ed[2:17]),
                  selected = NULL),
      br(),
      h4("Scatterplot"),
       selectInput("var1", "Select Variable For X-axis", choices=colnames(ed[2:17]),
                   selected = NULL),
      selectInput("var2", "Select Variable For Y-axis", choices=colnames(ed[2:17]),
                  selected = NULL)
    ),
    mainPanel(
      tabsetPanel(
        tabPanel("Count of Letters",plotOutput(outputId =
"bar",width="500px",height="500px"),dataTableOutput('table')),
        tabPanel("Histogram",plotOutput(outputId = "hist",width="600px",height="500px")),
```

```
       tabPanel("Boxplot",plotOutput(outputId =
"boxplot",width="500px",height="500px"),verbatimTextOutput("results")),
       tabPanel("Scatterplot",plotOutput(outputId = "scatter",width="700px",height="500px"))
       )
     )
   )
 )

server <- function(input, output) {

 output$table <- renderDataTable(df,options = list(pageLength=5))

 output$bar <- renderPlot({
   ggplot(data=df,aes(x=Letter,y=Count,width=.75)) +
geom_bar(position="dodge",stat="identity",fill="steelblue") + coord_flip()+ geom_text(aes(label=Count),
vjust=-0.3,hjust=0, size=3.5)
 })

 output$hist <- renderPlot({
    ggplot(ed, aes(ed[,input$var])) +
     geom_histogram(breaks=seq(0,15,by=1),fill="orange",color="black")+
     labs(x= colnames(ed[,input$var]),y="Count")
 })

 output$boxplot <- renderPlot({
   ggplot(ed, aes(x="",y=(ed[,input$var]))) +
     geom_boxplot(fill = "#56B4E9", color = "black",width=0.5)+
     labs(x= colnames(ed[,input$var]),y="Count")
 })

 output$results <- renderPrint({
  summary(ed[,input$var])
 })

 output$scatter <- renderPlot({
   qplot(ed[,input$var1],ed[,input$var2],data=ed,color=Letter) + labs(x= ed[0,input$var1],y=ed[0,input$var2])

 })

}
shinyApp(ui = ui, server = server)
```

- **R scripts**

  - Feature Selection

```
library(xlsx)
library(MASS)
library(randomForest)
library(caret)
library(Boruta)
library(earth)
library(plotmo)
library(plotrix)
library(TeachingDemos)
library(leaps)
library(faraway)

t1 <- read.xlsx2("/Users/AjayKulkarni/Study/Master of Science/Sem 2/CSI
703/Letter_Recognition_Data/Training.xlsx",1)

head(t1)

# Variable Selection

# 1. BIC plot

subsets <- regsubsets(Letter ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 +
X14 + X15 + X16,t1)
plot(subsets)

#2. Boruta
output <- Boruta(lenum ~ X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16,
data=t1,doTrace=2)
```

  - Classification

```
library(xlsx)
library(randomForest)
library(caret)
library(plotly)
library(e1071)
library(class)

t1 <- read.xlsx2("/Users/AjayKulkarni/Study/Master of Science/Sem 2/CSI
703/Letter_Recognition_Data/Training.xlsx",1)

t2 <- read.xlsx2("/Users/AjayKulkarni/Study/Master of Science/Sem 2/CSI
703/Letter_Recognition_Data/Testing.xlsx",1)

xtest <- rbind(t1[1,],t2)
xtest <- xtest[-1,]

# 1. Random Forest
model <- randomForest(Letter ~
X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16,data=t1,ntree=10,importace=TR
UE)
pred <- predict(model,newdata=xtest)
```

```
write.xlsx(pred,"/Users/AjayKulkarni/Study/Master of Science/Sem 2/CSI
703/Letter_Recognition_Data/Random_Forest.xlsx")
table(pred,t2$Letter)

# 2. Support Vector Machine (SVM)
model <- svm(Letter ~ .,data=t1)
pred <- predict(model,newdata=xtest)
write.xlsx(pred,"/Users/AjayKulkarni/Study/Master of Science/Sem 2/CSI
703/Letter_Recognition_Data/SVM.xlsx")
table(pred,t2$Letter)

# 3. KNN

# Selecting optimal value for K in KNN

labels <- t1[,1]
error.rate <- numeric(10)
for(i in 1:10)
{
  knn.pred <- knn(t1[,-1],t2[,-1],labels, k = i)
  error.rate[i] <- 1-mean(knn.pred == t2[,1])
}
plot(1:10, error.rate,"b", pch = 20, col = "red", xlab = "K", ylab = "Error Rate")

pred <- knn(t1[,-1],t2[,-1],labels, k = 1)
table(pred,t2$Letter)
write.xlsx(pred,"/Users/AjayKulkarni/Study/Master of Science/Sem 2/CSI
703/Letter_Recognition_Data/knn.xlsx")
```