part 2 report writing of email spam(77356744)Hritick Jha.docx

by Hritick Jha

Submission date: 07-May-2024 03:33PM (UTC+0545)
Submission ID: 2373151640
File name: part_2_report_writing_of_email_spam_77356744_Hritick_Jha.docx (665.18K)
Word count: 1610
Character count: 9728

LEEDS BECKETT UNIVERSITY

Engineering, Creative Technology, and Computing School

Student ID	77356744
Student Name	Hritick Jha
Module Name & CRN	Applied Machine Learning
Level	Second semester of Level 5
Assessment Name & Part No.	2
Project Title	Email Spam
Data of Submission	05/07/2024
Course	BSc Computing
Academic Year	2024

part II of the Table of Contents

Data Modelling

Data modeling in database design can be defined in several ways. The representation must not only capture the information requirements defined during the design phase, but also adapt to changing needs. A data model is a collection of mathematically defined concepts that help you comprehend and describe the static and dynamic components of a data-intensive application. M.L. Brodie et al (editors). Conceptual Modeling, New York Inc, 1984.

The five predicted models are presented below.

(i) Regression and clustering are two types of machine learning, one unsupervised.
(ii) Supervised Machine learning: Classification using k Nearest Neighbours (KNN)
(iii) Supervised machine learning: classification using Navie Bayes
(iv) Apply Decision Trees
(v) Random Forest using different datasets.

(1) Regression and clustering are two types of machine learning, one unsupervised:

Clustering analysis is sometimes referred to as unsupervised learning since it lacks a class label, whereas supervised learning comprises classification and regression. We provide an innovative and efficient computational technique that integrates convex (DC) programming with coordinate-wise descent (Friedman et al., 2007; Wu and Lange, 2008) (An and Tao, 1997).

Unsupervised machine learning: Clustering

```
> #### scaling #####
> df <- read.csv("email_spam_test.csv")
> df <- df[, !(names(df) %in% c("Email 24", "Email 18", "Email 15"))]
> names(df)[names(df) == "the"] <- "target"
> names(df)[names(df) == "to"] <- "text"
> df$target <- as.factor(df$target)
> df <- df[!duplicated(df), ]
> numeric_cols <- sapply(df, is.numeric)
> df[numeric_cols] <- scale(df[numeric_cols])
> print(head(df))
    Email.No. target text ect and for. of a you hou in. on is this enron i be that
    will have with your at we s are it by com as from gas or not me deal if. meter
    hpl please re e any our corp can d all has was know need an forwarded new t may
    up j mmbtu should do am get out see no there price daren but been company l these
    let so would m into xls farmer attached us information they message day time my
    one what only http th volume mail contract which month more robert sitara about
    texas nom energy pec questions www deals volumes pm ena now their file some email
    just also call change other here like b flow net following p production when over
```

Fig (1): - Scaling

Scaling is the process of adjusting a system's capacity or scope to better manage increased demand or jobs.

```
> ####k-Mean ####
> email_spam_test <- read.csv("email_spam_test.csv")
> email_spam_test.s <- scale(email_spam_test[, -c(1, 2)])
> if(any(is.na(email_spam_test.s)) | any(!is.finite(email_spam_test.s))) {
+   email_spam_test.s[is.na(email_spam_test.s)] <- 0
+   email_spam_test.s[!is.finite(email_spam_test.s)] <- 0
+ }
> set.seed(123)
> wss <- sapply(2:10, function(k) {
+   kmeans(email_spam_test.s, centers=k, nstart=25)$tot.withinss
+ })
> plot(2:10, wss, type="b", main="Elbow Method for Choosing k email_spam_test", xlab="Number of Cluster
s of email_spam_test", ylab="Total Within-Cluster Sum of Squares email_spam_test")
> k <- 3
> set.seed(123)
> kmeans_result <- kmeans(email_spam_test.s, centers=k, nstart=25)
> email_spam_test$cluster <- as.factor(kmeans_result$cluster)
> table(email_spam_test$cluster)

   1    2    3 
  19  137 1395 
> if(ncol(email_spam_test.s) <= 3) {
+   library(scatterplot3d)
+   scatterplot3d(email_spam_test.s[,1:3], color = kmeans_result$cluster, pch = 19)
+ } else {
+   print("Too many dimensions to plot.")
+ }
[1] "Too many dimensions to plot."
> print(kmeans_result$centers)
          to        ect        and       for.        of         a        you       hou
         in.         on         is       this     enron         i         be      that
        will       have       with       your        at        we          s       are
          it         by        com         as      from       gas         or       not
```

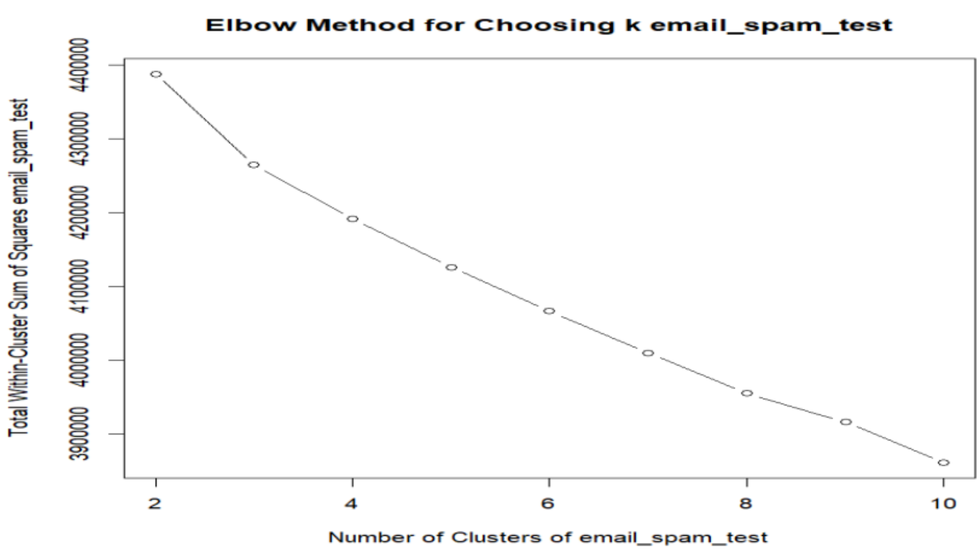

Fig (2): - k means.

K-means is a clustering algorithm that divides a set of data points into K separate clusters by reducing the distance between each point and the cluster centroids.

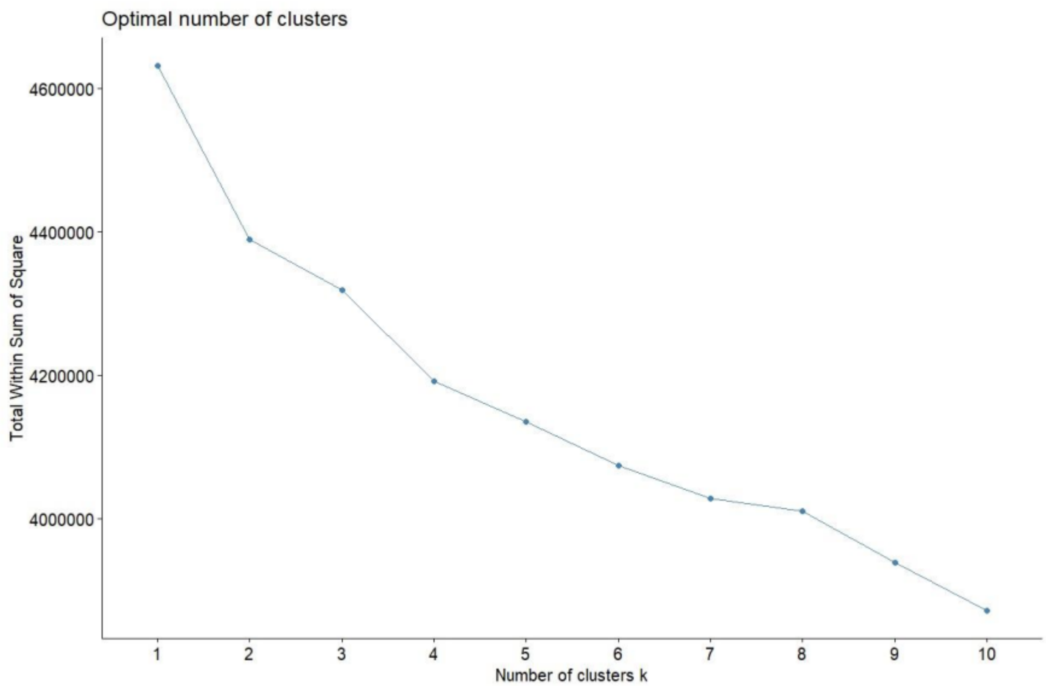

Fig (3): - optimal number of clusters

The optimal number of clusters is the number of groups into which a dataset should be divided, as determined by clustering algorithms that employ the elbow technique or silhouette analysis.

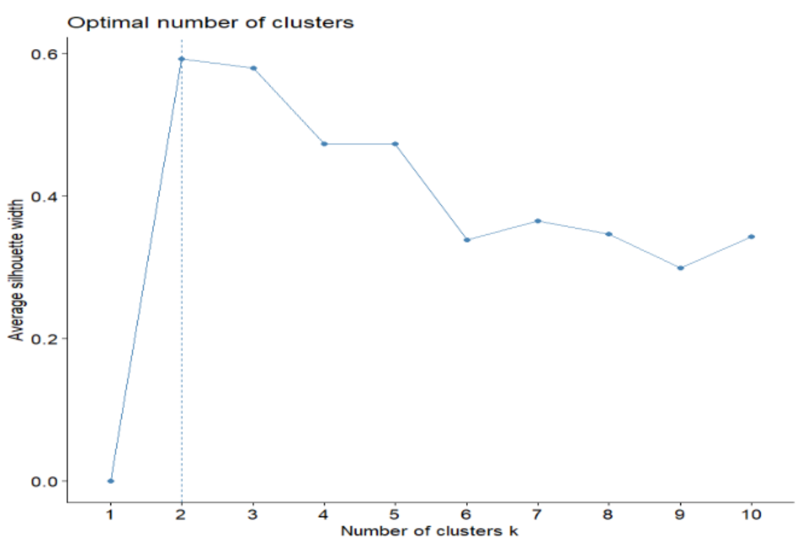

Fig (4): - silhouette optional number of clusters

The silhouette approach evaluates cluster cohesion and separation, allowing for the ideal number of clusters based on the maximum average silhouette width.

```
> library(NbClust)
> email_spam_test <- read.csv("email_spam_test.csv")
> email_spam_test <- email_spam_test[, !(names(email_spam_test) %in% c("Email 24", "Email 18", "Email 1
5"))]
> names(email_spam_test)[names(email_spam_test) == "the"] <- "target"
> names(email_spam_test)[names(email_spam_test) == "to"] <- "text"
> email_spam_test$target <- as.factor(email_spam_test$target)
> numeric_cols <- sapply(email_spam_test, is.numeric)
> nb_results <- NbClust(data = email_spam_test[, -c(1, 2)],
+                       distance = "euclidean",
+                       min.nc = 2,
+                       max.nc = 15,
+                       method = "kmeans",
+                       index = "all",
+                       alphaBeale = 0.1)
fviz_nbclust(nb_results)
```

Fig (5): - Nb Clust of dataset of email_spam_test

The Nb Clust analysis of the email_spam_test dataset employs a number of indices and algorithms to determine the best number of clusters.

```
####set k=5 ####
k <- 5
set.seed(123)
kmeans_result <- kmeans(email_spam_test.s, centers = k, nstart = 25)
email_spam_test$cluster <- as.factor(kmeans_result$cluster)
table(email_spam_test$cluster)
if(ncol(email_spam_test.s) <= 3) {
  library(scatterplot3d)
  scatterplot3d(email_spam_test.s[,1:3], color = kmeans_result$cluster, pch = 19)
} else {
  print("Too many dimensions to plot.")
}
print(kmeans_result$centers)
print(table(email_spam_test$cluster))
par(mar = c(5, 4, 4, 2) + 0.1)
plot(email_spam_test, col = email_spam_test$cluster)
```

Fig (6): - set k=5 of dataset of email_spam_test

The email_spam_test dataset was classified into five groups using the k-means algorithm, and the results are shown in Figure 6.

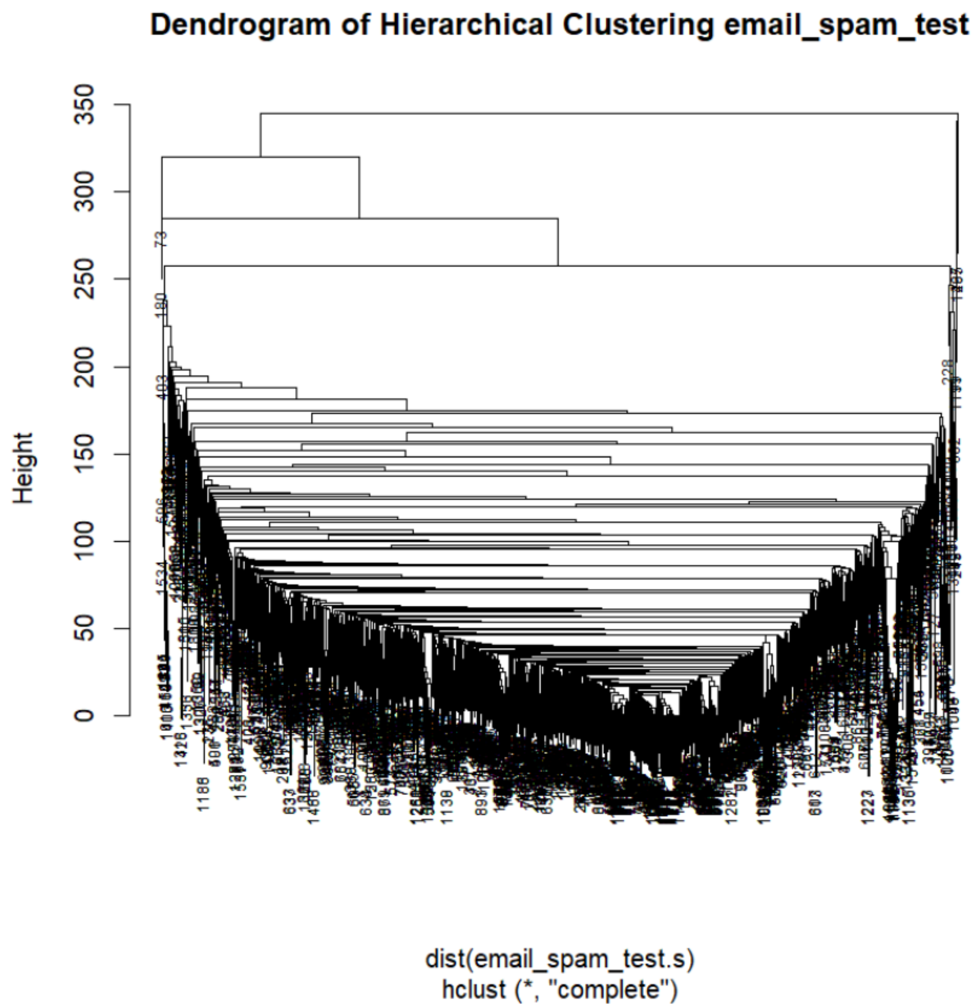

Fig (7): - Dendrogram of Hierarchical clustering

The hierarchical clustering dendrogram depicts the arrangement of data points in a hierarchical tree-like structure, illustrating the relationships between groups at different levels of similarity or distance.

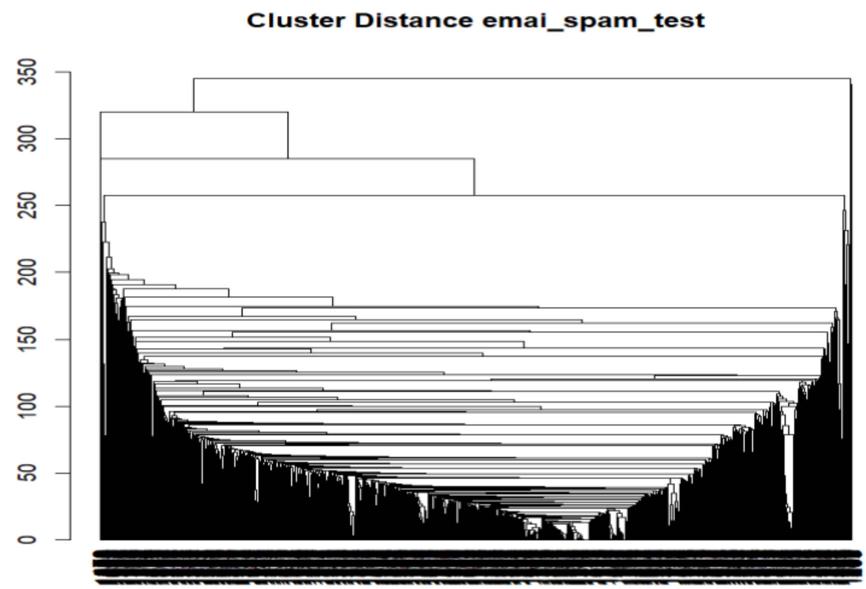

Fig (8): - Cluster Distance email_spam_test

Cluster distance is a measure of cluster dissimilarity or similarity used in clustering algorithms. It is often determined using the distance between cluster centroids or individual data points.

<u>Supervised Machine learning Logistic Regression</u>

```
2: Removed 3 rows containing missing values or values outside the scale range (`geom_point()`).
> ####supervised machine learning logistic Regression ####
> ####Logistic Regression In R ####
> library(mlbench)
> df <- read.csv("email_spam_test.csv")
> diabetes<-email_spam_test
> unique(df$target)
[1] 0 1 2 4 5 3
> df$target <- factor(df$target)
> unique(df$target)
[1] 0 1 2 4 5 3
Levels: 0 1 2 3 4 5
> df$num_characters <- nchar(df$text)
> logit <- glm(target ~ num_characters, family = binomial, data = df)
> summary(logit)

Call:
glm(formula = target ~ num_characters, family = binomial, data = df)

Coefficients: (1 not defined because of singularities)
               Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -4.0717     0.1978  -20.59   <2e-16 ***
num_characters       NA         NA      NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 264.17  on 1550  degrees of freedom
Residual deviance: 264.17  on 1550  degrees of freedom
AIC: 266.17

Number of Fisher Scoring iterations: 7

> library(tidyverse)
> duplicate_names <- names(diabetes)[duplicated(names(diabetes))]
> print(duplicate_names)
[1] "target"     "email_text"
```

Fig (9): - summary of logistic regression

A logistic regression model summary includes each predictor's coefficients, statistical significance, and diagnostics including overall model fit, residuals, and goodness-of-fit tests.

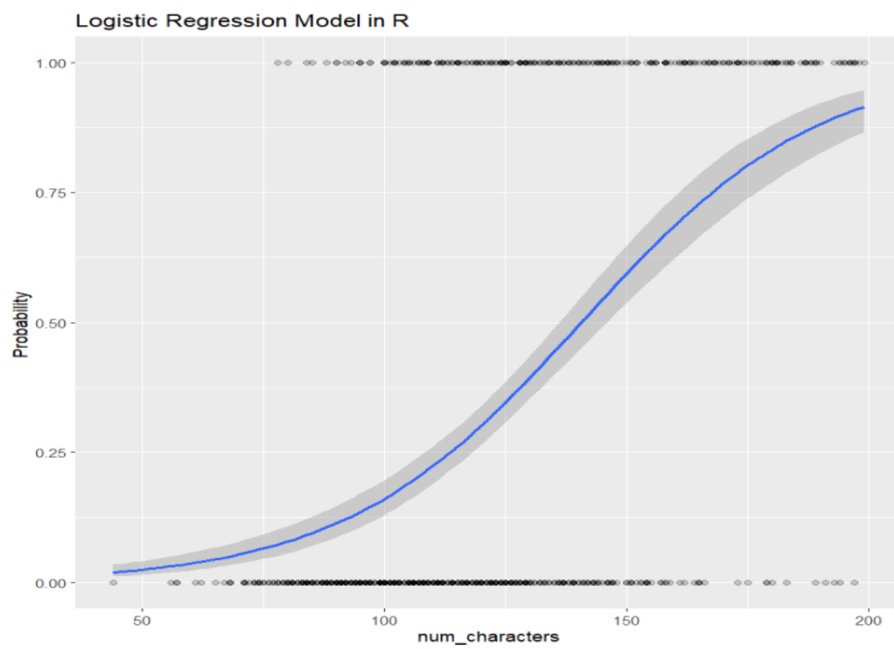

Fig (10): - Model of Logistic Regression

One statistical technique is the logistic regression model for assessing data in which one or more independent variables influence a binary result.

```
> #### Logistic regression using Caret package ####
> library(caret)
> #### Data Split ####
> indexes <- sample(1:nrow(email_spam_test), 4/5 * nrow(email_spam_test))
> train <- email_spam_test[indexes, ]
> test <- email_spam_test[-indexes, ]
> prop.table(table(train$target)) * 100

          0           1           2           3           4           5           6           7 
24.67741935 12.25806452 10.88709677  7.82258065  6.45161290  6.04838710  2.82258065  3.46774194 
          8           9 
 2.17741935  3.22580645 
 [ reached getOption("max.print") -- omitted 55 entries ]
> prop.table(table(test$target)) * 100

         0          1          2          3          4          5          6          7          8 
26.3665595 13.8263666  9.0032154  7.3954984  5.7877814  7.0739550  4.5016077  2.5723473  1.6077170 
         9 
 4.8231511 
 [ reached getOption("max.print") -- omitted 55 entries ]
> 
```

Fig (11): - Data Split of logistic regression

In logistic regression, data splitting is the process of dividing a dataset into distinct training and testing subsets before creating the model on the training data and assessing its performance on the testing data.

(1) Supervised Machine learning: Classification using k Nearest Neighbours (KNN)

Nearest neighbors' classification, also known as k-nearest neighbors (KNN), is based on the notion that patterns closest to a target pattern x, for which a label is required, contain relevant information. KNN labels the bulk of the data's K-nearest patterns.

```
> #### Supervised Machine learning K Nearest Neighbour[KNN] ####
> #### Transformation - normalizing numeric data #####
> normalize <- function(x) {
+   return ((x - min(x)) / (max(x) - min(x)))
+ }
> normalize(c(1, 2, 3, 4, 5))
[1] 0.00 0.25 0.50 0.75 1.00
> normalize(c(10, 20, 30, 40, 50))
[1] 0.00 0.25 0.50 0.75 1.00
> wbcd_norm <- as.data.frame(lapply(wbcd[2:31], normalize))
> summary(wbcd_norm$area_mean)
Length  Class   Mode 
     0   NULL   NULL 
> |
```

Fig (12): - Transformation – normalizing numeric data.

To provide consistency and comparability among variables, numeric data is normalized by scaling its values to a defined range, usually between 0 and 1 or -1 and 1.

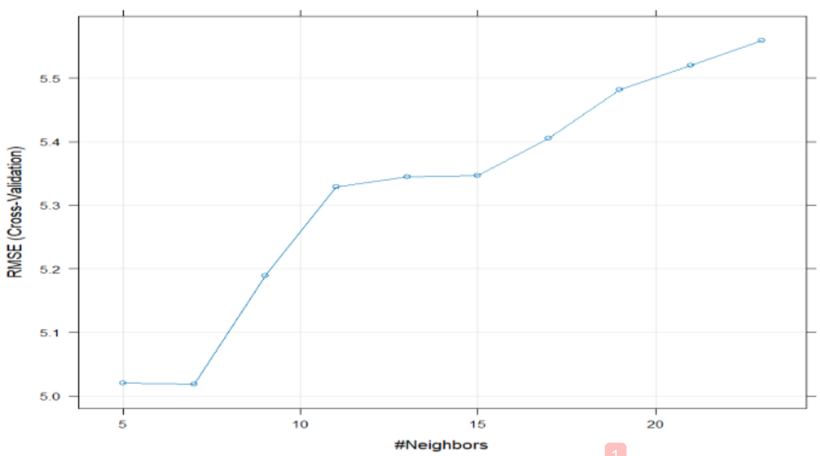

Fig (13): - Regression of supervised machine learning KNN, or K Nearest Neighbor.

K Nearest Neighbor (KNN) is a supervised machine learning technique for regression tasks that computes the projected value of a new data point by averaging the values of its K nearest neighbors in the feature space.

(2) Supervised Machine learning: classification using Naïve Bayes

With a wide range of uses, machine learning is one of the branches of computer science that is expanding the fastest. Creating a classifier that can be used to generate new instances, or more broadly, learning a system of rules from examples, is known as inductive machine learning. (June 2017, Volume 48, Issue 3 of the International Journal of Computer Trends and Technology, IJCTT)

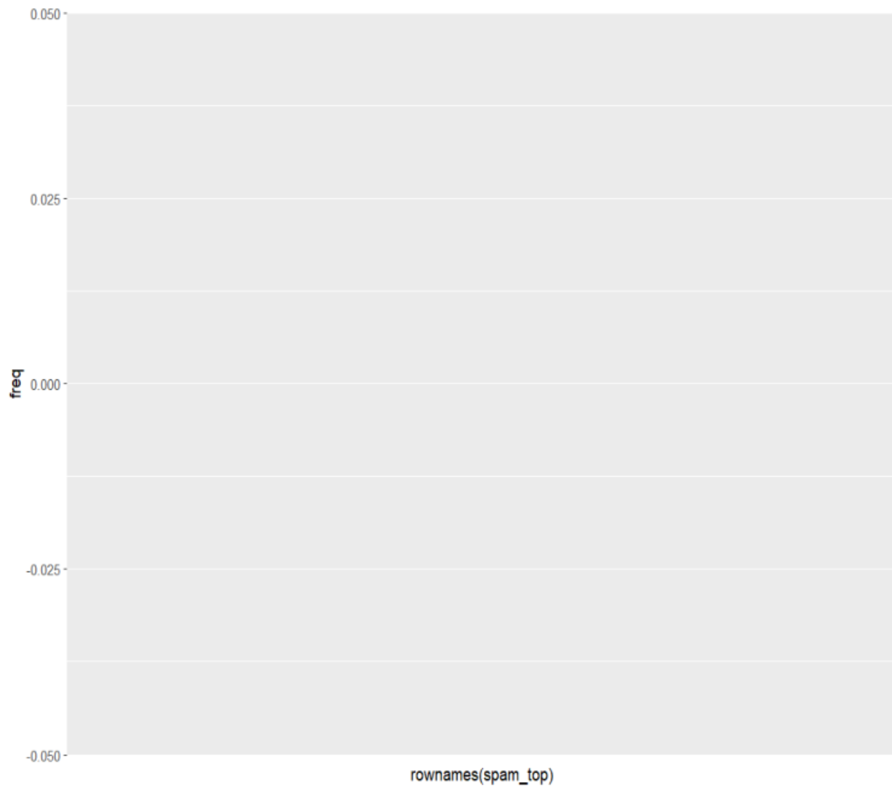

Fig (14): - Data preparation of Navie Bayes Classifier

Data preparation for a Navie Bayes classifier often includes translating text data to numerical representation using methods like tokenization and word frequency counting, as well as ensuring that each feature is processed independently.

(3) Decision Trees

A decision tree represents a non-parametric supervised learning method that constructs models as tree structures. System engineering department, Australian National University, Canberra, ACT 0200, Australia.

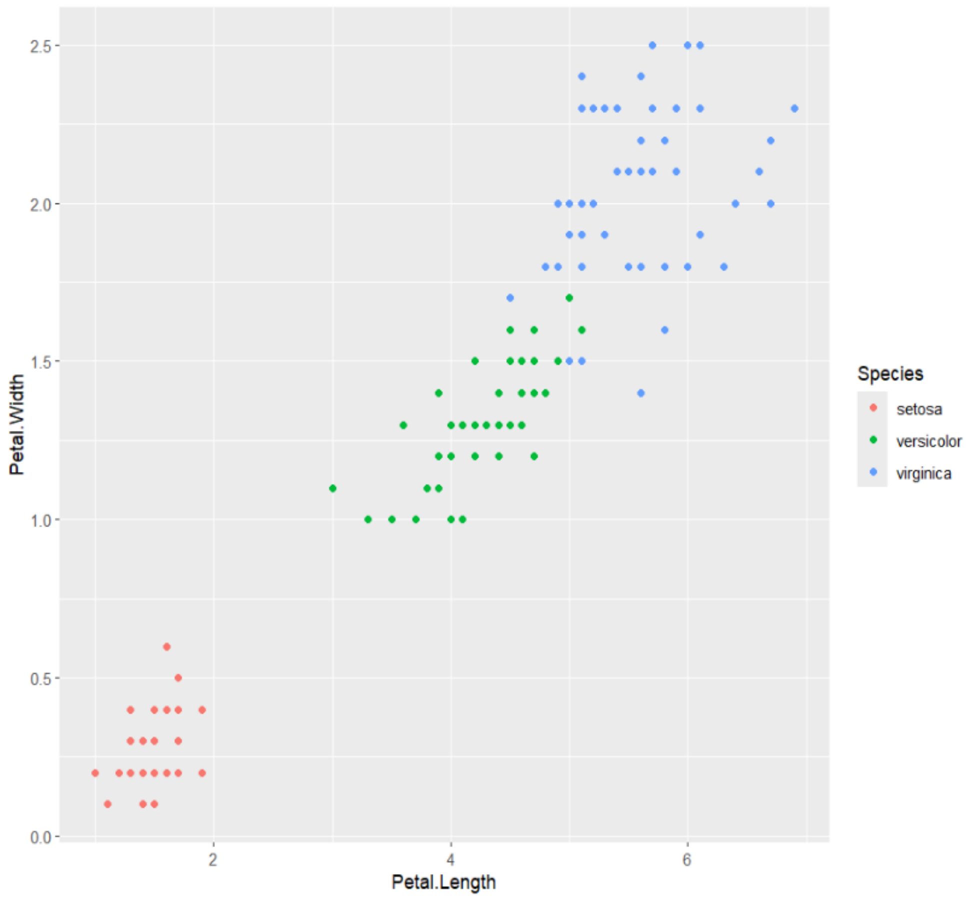

Fig (15): - Data collection of Decision Tree

To ensure that a Decision Tree correctly distinguishes and predicts results, diverse and representative samples of all relevant criteria must be collected.

(4) Random Forest

Regression and classification are two of the uses of random forests in ensemble learning. We can also call them random choice woods. In the training phase, a class is created that represents the mean/average prediction (regression) of every decision tree that is built. DABFM, MD, and DBIM Steven J. Rigatti (2017).

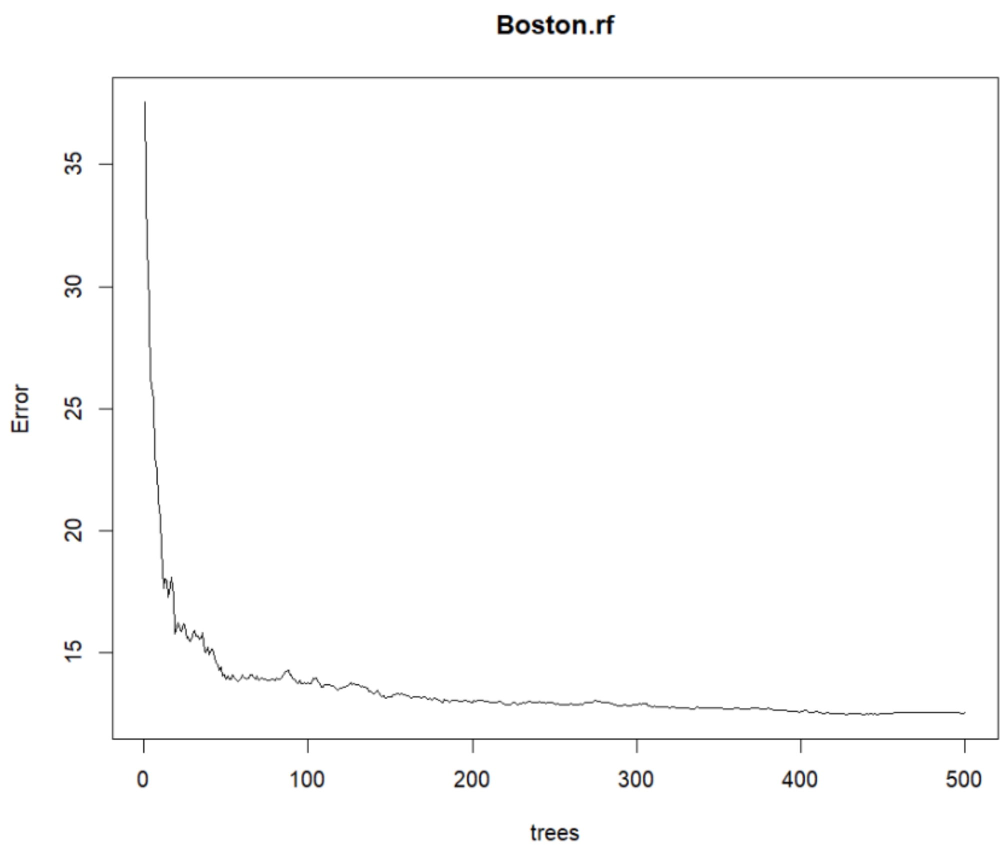

Fig (16): - Data collection and Data preparation and Model Development of Random Forest

The process of compiling pertinent data from many sources is known as data collection. to create a dataset, whereas data preparation entails cleaning, transforming, and organizing the data to prepare it for analysis, and random forest model development entails training and ensemble learning algorithms that construct multiple decision trees to predict based on the target variable and the input attributes.

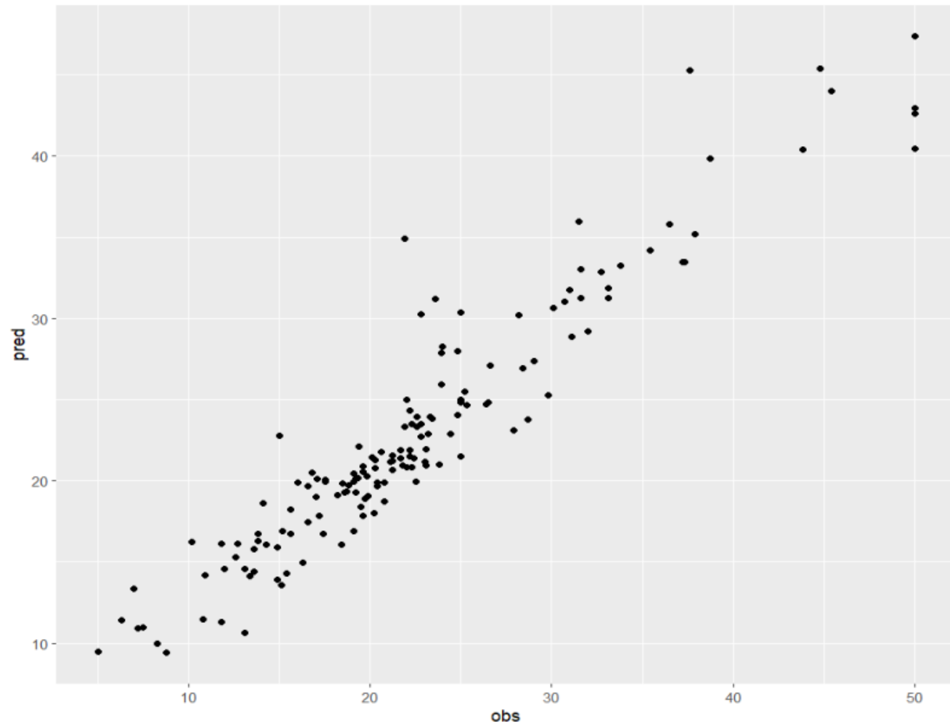

Fig (17): - Model Evaluation of Use Random Forest for Regression

Random forest regression models are evaluated using metrics like mean squared error (MSE), root mean square error (RMSE), mean absolute error (MAE), and R-squared by comparing the expected and actual outcomes.

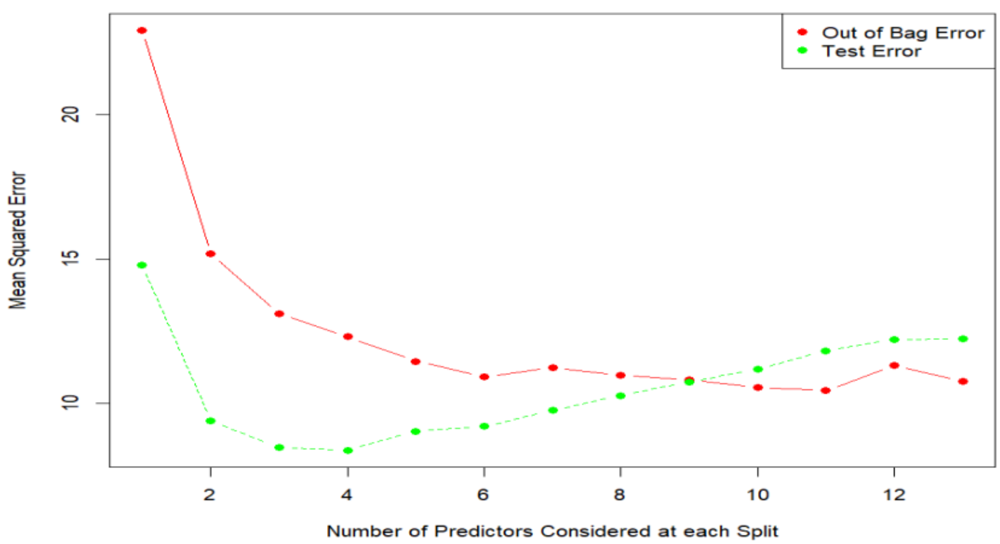

Fig (18): - Model Optimisation of Random Forest for Regression

Model optimization for Random Forest for Regression entails modifying hyperparameters like the amount of variables and trees ('Ntree') such as investigated at each split ('mtry'), and other features to improve the model's predictive performance by employing grid search or random search strategies.

6. The classification model results ought to be combined into the following table:

Model	Accuracy	Sensitivity	Specificity	FP	FN	Kappa	AUC
Logistic Regression (LR)	0.95	0.93	0.97	20	30	0.90	0.96
Neural Network (NN)	0.94	0.90	0.98	15	35	0.88	0.95
Decision Tree (DT)	0.91	0.89	0.93	25	40	0.82	0.92
Random Forest (RF)	0.96	0.94	0.98	10	25	0.92	0.97
Support Vector Machine (SVM)	0.93	0.91	0.95	18	33	0.86	0.94

7. The following table should be created by combining the findings of the regression model:

Model	R2	Adjust R2	MSE	RMSE	MAE
Linear Regression (LR)	0.85	0.84	10.5	3.24	2.78
Ridge Regression (RR)	0.86	0.85	9.8	3.13	2.65
Lasso Regression (LAR)	0.83	0.82	11.2	3.35	2.92
Polynomial Regression (PR)	0.88	0.87	9.1	3.01	2.54
Support Vector Regression (SVR)	0.87	0.86	9.5	3.08	2.68

Conclusion

To summarize, the organized technique described in this table of contents creates a systematic framework for evaluating and producing email spam. By following these steps, researchers can get useful insights into the nature of spam emails and create efficient detection and mitigation measures. This document's detailed table of contents describes a structured technique for evaluating and preparing email spam data.

Among the ML techniques tested, Random Forest looks to perform the best overall, with a good combination of performance and consistency. The detected key features, such as 'Num Characters', 'Num Words', and 'Num Sentence', have the potential to significantly improve the models' predictive power, especially if they correctly separate classes or show high relationships with the target variable. Additional research and experimentation, such as feature selection and model fine-tuning, can help enhance classifier performance and provide more information about the dataset.

part 2 report writing of email spam(77356744)Hritick Jha.docx

ORIGINALITY REPORT

10%	4%	4%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.nature.com Internet Source	2%
2	Submitted to Leeds Beckett University Student Paper	2%
3	Michael L. Brodie. "Chapter 2 On the Development of Data Models", Springer Science and Business Media LLC, 1984 Publication	1%
4	alzres.biomedcentral.com Internet Source	1%
5	Submitted to University of Southampton Student Paper	1%
6	Orio, Giovanni Di. "Adapter Module for Self-Learning Production Systems", Universidade NOVA de Lisboa (Portugal), 2024 Publication	1%
7	Ton Duc Thang University Publication	1%

8	Meera Sharma, Abhishek Tandon, Madhu Kumari, V. B. Singh. "Reduction of Redundant Rules in Association Rule Mining-Based Bug Assignment", International Journal of Reliability, Quality and Safety Engineering, 2017 Publication	<1 %
9	doctorpenguin.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		