**PROJECT2**

We are performing different classification techniques for the given data set ‘project2.txt’ and estimating the misclassification rate by using cross-validation.

The given data set is project2.txt first we load the data using read csv and there are no headers so header=FALSE and the columns are separated with ‘,’ so sep=’,’.

Now the first four columns represent four quantitative predictors. The fifth column stands for a categorical response, with two categories coded as 0 and 1.

The Dimension of the data set is 1375 rows and 5 columns.

we produce a matrix that contains all the pair wise correlations among the predictors in data set.

Here, we performed 4 types of classification:

* Logistic Regression
* QDA - Quadratic Discriminant Analysis
* LDA - Linear Discriminant Analysis
* KNN - K nearest neighbor

And done the K-Fold validation for all the four classifications from k=1 to k=10

sdm.prj2 = read.csv('/Users/vinithavudhayagiri/Downloads/project2.txt',sep = ",",header=FALSE)

dim(sdm.prj2)

names(sdm.prj2)

A picture containing text

Description automatically generated

summary(sdm.prj2)

A picture containing text

Description automatically generated

attach(sdm.prj2)

Graphical user interface, text, application

Description automatically generated

cor(sdm.prj2)

Text

Description automatically generated

**LOGISTIC REGRESSION**

Logistic regression is an example of supervised learning. It is used to calculate or predict the probability of a binary (yes/no) event occurrence.

Logistic = glm ( V5 ~ V1 + V2 + V3 + V4 , family = binomial ,data = sdm.prj2 )

coef (Logistic)

**Text

Description automatically generated with medium confidence**

summary (Logistic)

Table

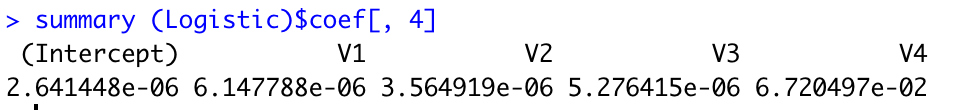
Description automatically generated

summary (Logistic)$coef

Text, letter

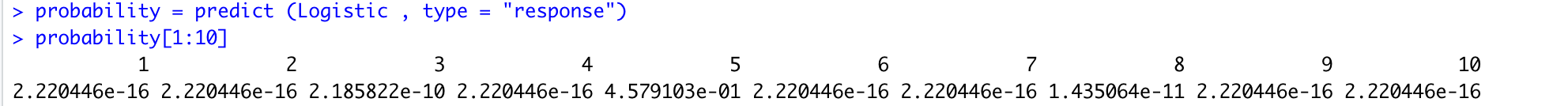
Description automatically generated

summary (Logistic)$coef[, 4]



probability = predict (Logistic , type = "response")

probability[1:10]



predict = rep ("0", 1372)

predict[probability > .6] = "1"

mean (predict == V5)

table (predict , V5)

Text

Description automatically generated

Now we are splitting our data to train.data as training data and test.data as test data as per the probability of 0.45 and 0.55 with respectively.

data.sample <- sample(c(TRUE, FALSE), nrow(sdm.prj2), replace=TRUE, prob=c(0.45,0.55))

train.data <- sdm.prj2[!data.sample, ]

dim(train.data)

A picture containing graphical user interface

Description automatically generated

test.data<- sdm.prj2[data.sample, ]

dim(test.data)

A picture containing application

Description automatically generated

final <- predict (Logistic , newdata = test.data, type = "response")

final <- ifelse(final > 0.5,1,0)

Error.data <- mean(final != test.data$V5)

print(paste('Accuracy', 1 - Error.data))

Graphical user interface, text

Description automatically generated

**K-FOLD FROM 1 TO 10 CROSS VALIDATION FOR LOGISTIC REGRESSION**

In here we are validating our data set using logistic regression with the help of K-FOLD validation.

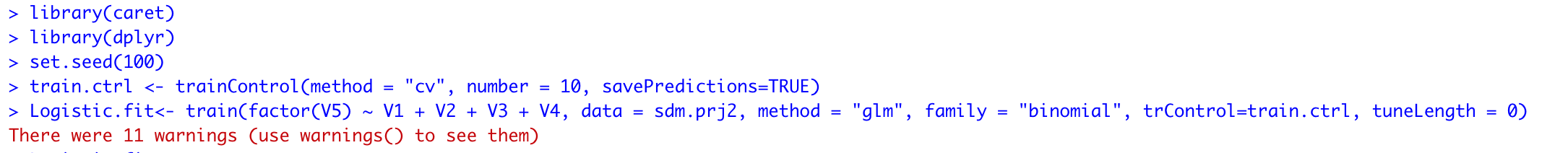
library(caret)

library(dplyr)

set.seed(100)

train.ctrl <- trainControl(method = "cv", number = 10, savePredictions=TRUE)

Logistic.fit<- train(factor(V5) ~ V1 + V2 + V3 + V4, data = sdm.prj2, method = "glm", family = "binomial", trControl=train.ctrl, tuneLength = 0)



Logistic.fit

Graphical user interface, text, application, email

Description automatically generated

Here the accuracy for logistic regression is 0.9897916.

predict <- Logistic.fit$pred

predict$equal <- ifelse(predict$pred == predict$obs, 1,0)

fold <- predict %>%

group\_by(Resample) %>%

summarise\_at(vars(equal),

list(Accuracy = mean))

fold

Graphical user interface, text, application

Description automatically generated

**QUADRATIC DISCRIMINANT ANALYSIS**

Quadratic Discriminant Analysis (QDA) is a generative model. QDA assumes that each class follow a Gaussian distribution. The class-specific prior is simply the proportion of data points that belong to the class. The class-specific mean vector is the average of the input variables that belong to the class.

library(MASS)

library(ggplot2)

qda.fit = qda(V5~V1+V2+V3+V4, data=train.data)

qda.fit

Text

Description automatically generated

qda.class=predict(qda.fit,train.data)$class

table(qda.class)

mean(qda.class==V5)

Text

Description automatically generated

**K-FLOD FROM 1 TO 10 CROSS VALIDATION FOR QUADRATIC DISCRIMINANT ANALYSIS**

We are validating our data set using Quadratic discriminant analysis with the help of K-FOLD validation.

library(caret)

library(dplyr)

set.seed(100)

train.ctrl <- trainControl(method = "cv", number = 10, savePredictions=TRUE)

qda.fit<- train(factor(V5) ~ V1 + V2 + V3 + V4, data = sdm.prj2, method = "qda", trControl=train.ctrl, tuneLength = 0)

qda.fit

Graphical user interface, text, application, email

Description automatically generated

Here, the accuracy for Quadratic discriminant analysis is 0.9839733 comparing with logistic regression it has less accuracy.

predict <- qda.fit$pred

predict$equal <- ifelse(predict$pred == predict$obs, 1,0)

fold <- predict %>%

group\_by(Resample) %>%

summarise\_at(vars(equal),

list(Accuracy = mean))

fold

Graphical user interface, text, application, chat or text message

Description automatically generated

**LINEAR DISCRIMINANT ANALYSIS**

Linear Discriminant Analysis or Normal Discriminant Analysis or Discriminant Function Analysis is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e., separating two or more classes.

sdm.prj2[1:4] = scale(sdm.prj2[1:4])

lda <- lda(V5 ~ V1 + V2 + V3 + V4, data=train.data)

lda

lda.pred <- predict(lda, train.data)

**Graphical user interface, text

Description automatically generated**

names(lda.pred)

A picture containing company name

Description automatically generated

head(lda.pred$class)

A picture containing text

Description automatically generated

head(lda.pred$posterior)

Text

Description automatically generated

mean(lda.pred$class == test.data$V5)

**Text

Description automatically generated with medium confidence**

**K-FLOD FROM 1 TO 10 CROSS VALIDATION FOR LINEAR DISCRIMINANT ANALYSIS**

We are validating our data set using Linear discriminant analysis with the help of K-FOLD validation.

library(caret)

library(dplyr)

set.seed(100)

train.ctrl <- trainControl(method = "cv", number = 10, savePredictions=TRUE)

lda.fit<- train(factor(V5) ~ V1 + V2 + V3 + V4, data = sdm.prj2, method = "lda", trControl=train.ctrl, tuneLength = 0)

lda.fit

Graphical user interface, text, application, email

Description automatically generated

Here, the accuracy for Linear discriminant analysis is 0.9766847 comparing with logistic regression it has less accuracy.

predict <- lda.fit$pred

predict$equal <- ifelse(predict$pred == predict$obs, 1,0)

fold <- predict %>%

group\_by(Resample) %>%

summarise\_at(vars(equal),

list(Accuracy = mean))

fold

**Graphical user interface, text, application

Description automatically generated**

**KNN CLASSIFIER**

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

library(e1071)

library(class)

library(caTools)

test = scale(test.data[, 1:4])

train = scale(train.data[, 1:4])

knn.clf = knn(train = train, test = test, cl = train.data$V5, k = 5)

clf<- table(test.data$V5,knn.clf)

clf

Graphical user interface, text, application

Description automatically generated

mean (test.data$V5 ==knn.clf)

A picture containing graphical user interface

Description automatically generated

knn.clf = knn(train = train, test = test, cl = train.data$V5, k = 10)

clf<- table(test.data$V5,knn.clf)

clf

mean (test.data$V5 ==knn.clf)

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**K-FOLD FROM 1 TO 10 CROSS VALIDATION FOR KNN**

We are validating our data set using KNN with the help of K-FOLD validation.

library(caret)

library(dplyr)

set.seed(100)

train.ctrl <- trainControl(method = "cv", number = 10, savePredictions=TRUE)

knn.fit<- train(factor(V5) ~ V1 + V2 + V3 + V4, data = sdm.prj2, method = "knn", trControl=train.ctrl, tuneLength = 0)

knn.fit

Graphical user interface, text, application, email

Description automatically generated

Here, the accuracy for KNN is 0.9985454 comparing with other classifications it has high accuracy and low-test misclassification rate.

predict <- knn.fit$pred

predict$equal <- ifelse(predict$pred == predict$obs, 1,0)

fold <- predict %>%

group\_by(Resample) %>%

summarise\_at(vars(equal),

list(Accuracy = mean))

fold

Graphical user interface, text, application

Description automatically generated

**Team Members:**

**Alekhya Monavarthi -50469035**

**Preethi Abhilasha Vaddi-50483865**

**Shanawaz Pathan-50471026**

**Vinitha Vudhayagiri-50478854**