Predictive Analysis

library(e1071) library(caret) library(caTools) library(rpart) library(rpart.plot) library(kernlab) library(readr) library(ggplot2)

Working directory

getwd()

Extracting data from csv file

df_admission <- read.csv('College_admission.csv')

class(df_admission) dim(df_admission) View(df_admission) names(df_admission)

Finding the missing values

sum(is.na(df_admission))

Finding outliers using boxplot

boxplot(df_admission$gre, main="College admission ", xlab ="Gre Score", ylab ="Gre", horizontal=TRUE)

summary(df_admission)

Creating copy of the dataframe for further analysis

admin <- data.frame(df_admission)

Finding the quartile to remove outliers

Q <- quantile(admin$gre,probs=c(.25, .75), na.rm = FALSE) iqr<- IQR(admin$gre) up <- Q[2]+1.5*iqr # Upper Range low<- Q[1]-1.5*iqr # Lower Range

admin <- subset(admin, admin$gre > low & admin$gre < up) View(admin) dim(admin)

Boxplot showing no outliers

boxplot(admin$gre, main="College admission ", xlab ="Gre Score", ylab ="Gre", horizontal=TRUE)

Find the structure of the data set and if required, transform the numeric data type to factor and vice-versa.

str(admin)

Finding whether data is normally distributed or not

mean(admin$gre) sd(admin$gre)

d_norm <- dnorm(admin$gre, mean = 590.8081, sd = 111.7971) d_norm plot(density(d_norm, adjust = 10))# The density plot talks that dataframe is normally distributed

Use variable reduction techniques to identify significant variables

Finding correlation between dependent variable and independent variable

View(cor(admin))

Dropping insignificant column

admin <- subset(admin , select = -c(ses, Race, Gender_Male)) View(admin)

Run logistic model to determine the factors that influence the admission process of a student

(Drop insignificant variables)

Separating the categorical variable and converting into factor

rank_cat<-admin[,-c(1,2,3)] rank_cat <- as.factor(rank_cat) View(rank_cat)

Creating dummy variable

dummy_rank <- data.frame(model.matrix(~rank_cat-1)) View(dummy_rank)

Combing the dummy columns and the other columns

admin_final <- cbind(admin[,c(1,2,3)],dummy_rank) View(admin_final)

Spliting the data into train and test

set.seed(123)

indices = sample.split(admin_final$admit, SplitRatio = 0.7)

train = admin_final[indices,]

test = admin_final[!(indices),]

dim(train) dim(test)

Model Building

Logistic model

model_1 = glm(admit ~ ., data = train, family = "binomial")

summary(model_1)

View(test)

Test Data

Probabilities prediction of admit variable for test data

test_pred = predict(model_1, type = "response", newdata = test)

test_pred

test$prob <- test_pred

View(test)

Using the probability cutoff of 50%.

test_pred_admit <- factor(ifelse(test_pred >= 0.50, "Yes", "No")) test_actual_admit <- factor(ifelse(test$admit==1,"Yes","No")) test_actual_admit test_pred_admit

table(test_actual_admit,test_pred_admit)

test_conf <- confusionMatrix(test_pred_admit, test_actual_admit, positive = "Yes") test_conf

Based on specificity we can say that it is good in predicting class 0

Decision tree

admin_tree <- admin

prop.table(table(admin_tree$admit))

table(admin_tree$admit)

set.seed(123) split.indices <- sample(nrow(admin_tree), nrow(admin_tree)*0.7, replace = F) train <- admin_tree[split.indices,] test <- admin_tree[-split.indices,]

View(train)

Decision Tree

tree.model <- rpart(admit ~ ., # formula data = train, # training data method = "class") # classification not regression

display decision tree

prp(tree.model)

make predictions on the test set

tree.predict <- predict(tree.model, test, type = "class") tree.predict

evaluate the results

confusionMatrix(tree.predict, as.factor(test$admit), positive = "1")

Based on specificity above model is good in prediction of class 0

SVM Model

Converting dependent variable into factor

admin_svm <- admin_final

admin_svm$admit <- as.factor(admin_svm$admit)

set.seed(123)

indices = sample.split(admin_svm$admit, SplitRatio = 0.7)

train = admin_svm[indices,]

test = admin_svm[!(indices),]

dim(train)

dim(test)

model_svm<- ksvm(admit ~ ., data = train,scale = FALSE , C=1)

Predicting the model results

evaluate_1<- predict(model_svm, test)

levels(as.factor(evaluate_1))

Confusion Matrix - Finding accuracy, Sensitivity and specificity

confusionMatrix(evaluate_1, as.factor(test$admit))

Based on sensitivity above model is good in prediction of class 1

Based on the three model Logistic Regression, Decision tree and SVM model

We can conclude that Logistic regression and SVM model are better fit model

Accuracy, Sensitivity and Specificity is very good as compared to Decision tree

Descriptive Analysis

admin1 <- admin admin1 <- within(admin1, { gre_cat <- NA # need to initialize variable gre_cat[gre <= 440] <- "Low" gre_cat[gre > 440 & gre <= 580] <- "Middle" gre_cat[gre > 580] <- "High" }) str(admin1) dim(admin1) admin1$gre_cat <- factor(admin1$gre_cat, levels = c("High", "Middle", "Low"))

str(admin1$gre_cat)

sum_Desc <- aggregate(admit~gre_cat, admin1 , FUN = sum) length_Desc <- aggregate(admit~gre_cat, admin1, FUN = length) probability_table <- cbind(sum_Desc, recs = length_Desc[,2])

probability_table_final <- transform(probability_table,probability_admission = admit/recs) probability_table_final

Creating the point chart

ggplot(probability_table_final, aes(x = gre_cat, y = probability_admission))+geom_point() table(admin1$admit, admin1$gre_cat)

knitr::stitch('Assignment.R') Sys.which("pdflatex") pdflatex ""