# Q1-Logistic model to predict success or Failure of students who are doing preliminary year to see if they will be admitted to college.

* The data is then read in from a CSV file using the read\_csv function from the readr package.
* The data is checked for missing values using the sapply function, and unnecessary columns are removed using the select function from the dplyr package. The last two rows are also removed using the slice function from the dplyr package.
* Next, numerical null values are imputed with the mean using the mutate\_all function from the dplyr package.
* Numerical variables are then scaled using the scale function, and outliers are detected and removed using a custom function that utilizes the quantile and interquartile range.
* The caret package is loaded, and categorical variables are converted to numerical using the dummyVars and predict functions.
* A correlation matrix is then created using the cor function, and highly correlated columns are identified using the which and abs functions. These highly correlated columns are removed from the dataset.
* The data is then split into training and testing sets using the createDataPartition function from the caret package.
* The logistic regression model is then fit on the training data using the cv.glmnet function from the glmnet package.
* Predictions are made on the testing data using the predict function, and the accuracy of the model is calculated. The F1 score is also calculated using the F1\_Score function from the MLmetrics package.
* The coefficients from the fitted model are extracted using the coef function, and a data frame is created with the variable names and coefficients.
* This data frame is then sorted by absolute value, and the top 10 features are selected. The top 10 features are then used to fit another logistic regression model on the training data and make predictions on the testing data.
* The accuracy of this model is also calculated and the value comes out to be 0.9032258 and F1 score is 0.8888889
* It can be observed that selecting the top 10 features and training on them doesn’t affect our accuracy by a significant amount, hence all the features are significant to our model and should be considered while training.

Code for Logistic model to predict Success or failure of students who are doing preliminary year.

library(readr)

library(tidyr)

library(dplyr)

library(MLmetrics)

library(corrplot)

library(glmnet)

data <- read\_csv("232408682483120\_File.csv")

View(data)

#checking for missing values column wise

sapply(data, function(x) sum(is.na(x)))

#drop unnecessary columns

data <- select(data, -c("Reason for not Completing Connect", "Reason not Retained"))

#drop unnecesaary rows

data <- slice(data, 1:(nrow(data)-2))

# Impute numerical null values with mean

data <- mutate\_all(data, funs(ifelse(is.na(.), mean(., na.rm=TRUE), .)))

# Scaling numerical variables

numerical\_var <- c("High School GPA", "SAT Score","Dropout Proneness (percentile score before start of semester)", "Predicted Academic Difficulty (percentile score before start of semester)", "Educational Stress (percentile score before start of semester)",

 "Receptivity to Institutional Help (percentile score before start of semester)", "Receptivity to Academic Assistance (percentile score before start of semester)", "Receptivity to Personal Counseling (percentile score before start of semester)",

 "Receptivity to Social Engagement (percentile score before start of semester)", "Receptivity to Career Guidance ((percentile score before start of semester)", "Receptivity to Financial Guidance (percentile score before start of semester)",

 "Desire to Transfer (percentile score before start of semester)", "F17 GPA", "S18 GPA", "CUM GPA", "Number of Credits Earned",

 "Number of Faculty Advisor Meetings Attended", "Number of Peer Mentor Meetings Attended", "Number of Workshops Attended")

data[numerical\_var] <- scale(data[numerical\_var])

#OUTLIER DETECTION

outliers <- function(x) {

 Q1 <- quantile(x, probs=.25)

 Q3 <- quantile(x, probs=.75)

 iqr = Q3-Q1

 upper\_limit = Q3 + (iqr\*1.5)

 lower\_limit = Q1 - (iqr\*1.5)

 x > upper\_limit | x < lower\_limit

}

remove\_outliers <- function(data, cols = names(data)) {

 for (col in cols) {

 data <- data[!outliers(data[[col]]),]

 }

 data

}

#OUTLIER DETECTION

remove\_outliers(data, numerical\_var)

library(caret)

#CATEGORICAL TO NUMERICAL

dmy <- dummyVars(" ~ .", data = data, fullRank = T)

data\_transformed <- data.frame(predict(dmy, newdata = data))

View(data\_transformed)

# Create correlation matrix

corr\_matrix <- cor(data\_transformed)

# Find highly correlated pairs of columns

high\_corr\_pairs <- which(abs(corr\_matrix) > 0.75 & lower.tri(corr\_matrix), arr.ind = TRUE)

# Identify columns to remove

cols\_to\_remove <- unique(high\_corr\_pairs[, 2])

# Remove highly correlated columns

data\_transformed <- data\_transformed[, -cols\_to\_remove]

# Split data into training and testing sets

set.seed(69)

train\_index <- createDataPartition(data\_transformed$X.Retained.F17.F18...1.yes..0.no.., p = 0.7, list = FALSE)

train\_data <- data\_transformed[train\_index, ]

test\_data <- data\_transformed[-train\_index, ]

View(train\_data)

View(test\_data)

# Fit logistic regression model on training data

x\_train <- as.matrix(train\_data[, -ncol(train\_data)])

y\_train <- train\_data$X.Retained.F17.F18...1.yes..0.no..

fit <- cv.glmnet(x\_train, y\_train, family = "binomial", type.measure = "class")

# Predict on test data and calculate accuracy

x\_test <- as.matrix(test\_data[, -ncol(test\_data)])

y\_test <- test\_data$X.Retained.F17.F18...1.yes..0.no..

y\_pred <- predict(fit, newx = x\_test, s = fit$lambda.min, type = "class")

accuracy <- sum(y\_pred == y\_test) / length(y\_test)

print(accuracy)

F1\_Score(y\_pred,y\_test)

# Extract coefficients from fitted model

coeficients <- coef(fit, s = fit$lambda.min)

coef\_data <- data.frame(variable = rownames(coeficients)[-1], coefficient = coeficients[-1])

# Sort coefficients by absolute value

coef\_data <- coef\_data[order(abs(coef\_data$coefficient), decreasing = TRUE), ]

View(coef\_data)

# Select top 10 features

top\_10 <- coef\_data$variable[1:10]

# Select top 10 features from training and testing data

train\_data\_10<- train\_data[, c(top\_10, "X.Retained.F17.F18...1.yes..0.no..")]

test\_data\_10 <- test\_data[, c(top\_10, "X.Retained.F17.F18...1.yes..0.no..")]

# Fit logistic regression model on training data with top features

x\_train\_10 <- as.matrix(train\_data\_10[, -ncol(train\_data\_10)])

y\_train\_10 <- train\_data\_10$X.Retained.F17.F18...1.yes..0.no..

fit\_10 <- cv.glmnet(x\_train\_10, y\_train\_10, family = "binomial", type.measure = "class")

# Predict on test data and calculate accuracy

x\_test\_10 <- as.matrix(test\_data\_10[, -ncol(test\_data\_10)])

y\_test\_10 <- test\_data\_10$X.Retained.F17.F18...1.yes..0.no..

y\_pred\_10 <- predict(fit\_10, newx = x\_test\_10, s = fit\_10$lambda.min, type = "class")

accuracy\_10 <- sum(y\_pred\_10 == y\_test\_10) / length(y\_test\_10)

accuracy\_10

F1\_Score(y\_pred\_10,y\_test\_10)

# Q2. Heart data set

* First, the necessary libraries are loaded, including readxl, tidyr, dplyr, corrplot, caret, and glmnet. The Excel file containing the dataset is then read into a dataframe called df.
* Missing values are detected using sapply and removed using na.omit. Unnecessary columns are dropped using select, and the continuous variable Age is scaled.
* Outliers are detected and removed using the outliers and remove\_outliers functions. The binary variable delay\_days\_binary is created using mutate based on the delaydays variable.
* The correlation matrix is computed using cor and visualized as a heatmap using corrplot. Highly correlated pairs of columns are identified using which and abs, and the dataset is split into training and testing sets using createDataPartition.
* A logistic regression model is fit on the training data using cv.glmnet. The model is evaluated on the testing data using predict and the accuracy and F1 score are calculated.
* The coefficients from the model are extracted using coef and sorted by absolute value. The top 10 features are selected using head and the model is refit using only these variables. The accuracy and F1 score are then calculated again for this model.
* In the second part of the code, the average delay days are calculated and a binary variable is created to indicate whether delay days are less than or equal to the cohort average.
* The original delaydays column is dropped, and the dataset is split into training and testing sets again. A logistic regression model is fit on the training data, evaluated on the testing data, and the accuracy and F1 score are calculated.
* It can be observed that selecting the top 10 features and training on them doesn’t affect our accuracy by a significant amount, hence all the features are significant to our model and should be considered while training.

|  |  |  |
| --- | --- | --- |
| Part | Accuracy | F1 Score |
| A: all features | 0.4761905 | 0.3125 |
| A: top 10 features | 0.4761905 | 0.3125 |
| B: all features | 0.6428571 | 0.6153846 |
| B: top 10 features | 0.6190476 | 0.5789474 |

|  |  |  |
| --- | --- | --- |
| C: all features | 0.7142857 | 0.8064516 |
| C: top 10 features | 0.7142857 | 0.8064516 |

Code for Heart data set:

library(readr)

library(tidyr)

library(dplyr)

library(MLmetrics)

library(corrplot)

library(glmnet)

data <- read\_csv("232408682483120\_File.csv")

View(data)

#checking for missing values column wise

sapply(data, function(x) sum(is.na(x)))

#drop unnecessary columns

data <- select(data, -c("Reason for not Completing Connect", "Reason not Retained"))

#drop unnecesaary rows

data <- slice(data, 1:(nrow(data)-2))

# Impute numerical null values with mean

data <- mutate\_all(data, funs(ifelse(is.na(.), mean(., na.rm=TRUE), .)))

# Scaling numerical variables

numerical\_var <- c("High School GPA", "SAT Score","Dropout Proneness (percentile score before start of semester)", "Predicted Academic Difficulty (percentile score before start of semester)", "Educational Stress (percentile score before start of semester)",

 "Receptivity to Institutional Help (percentile score before start of semester)", "Receptivity to Academic Assistance (percentile score before start of semester)", "Receptivity to Personal Counseling (percentile score before start of semester)",

 "Receptivity to Social Engagement (percentile score before start of semester)", "Receptivity to Career Guidance ((percentile score before start of semester)", "Receptivity to Financial Guidance (percentile score before start of semester)",

 "Desire to Transfer (percentile score before start of semester)", "F17 GPA", "S18 GPA", "CUM GPA", "Number of Credits Earned",

 "Number of Faculty Advisor Meetings Attended", "Number of Peer Mentor Meetings Attended", "Number of Workshops Attended")

data[numerical\_var] <- scale(data[numerical\_var])

#OUTLIER DETECTION

outliers <- function(x) {

 Q1 <- quantile(x, probs=.25)

 Q3 <- quantile(x, probs=.75)

 iqr = Q3-Q1

 upper\_limit = Q3 + (iqr\*1.5)

 lower\_limit = Q1 - (iqr\*1.5)

 x > upper\_limit | x < lower\_limit

}

remove\_outliers <- function(data, cols = names(data)) {

 for (col in cols) {

 data <- data[!outliers(data[[col]]),]

 }

 data

}

#OUTLIER DETECTION

remove\_outliers(data, numerical\_var)

library(caret)

#CATEGORICAL TO NUMERICAL

dmy <- dummyVars(" ~ .", data = data, fullRank = T)

data\_transformed <- data.frame(predict(dmy, newdata = data))

View(data\_transformed)

# Create correlation matrix

corr\_matrix <- cor(data\_transformed)

# Find highly correlated pairs of columns

high\_corr\_pairs <- which(abs(corr\_matrix) > 0.75 & lower.tri(corr\_matrix), arr.ind = TRUE)

# Identify columns to remove

cols\_to\_remove <- unique(high\_corr\_pairs[, 2])

# Remove highly correlated columns

data\_transformed <- data\_transformed[, -cols\_to\_remove]

# Split data into training and testing sets

set.seed(69)

train\_index <- createDataPartition(data\_transformed$X.Retained.F17.F18...1.yes..0.no.., p = 0.7, list = FALSE)

train\_data <- data\_transformed[train\_index, ]

test\_data <- data\_transformed[-train\_index, ]

View(train\_data)

View(test\_data)

# Fit logistic regression model on training data

x\_train <- as.matrix(train\_data[, -ncol(train\_data)])

y\_train <- train\_data$X.Retained.F17.F18...1.yes..0.no..

fit <- cv.glmnet(x\_train, y\_train, family = "binomial", type.measure = "class")

# Predict on test data and calculate accuracy

x\_test <- as.matrix(test\_data[, -ncol(test\_data)])

y\_test <- test\_data$X.Retained.F17.F18...1.yes..0.no..

y\_pred <- predict(fit, newx = x\_test, s = fit$lambda.min, type = "class")

accuracy <- sum(y\_pred == y\_test) / length(y\_test)

print(accuracy)

F1\_Score(y\_pred,y\_test)

# Extract coefficients from fitted model

coeficients <- coef(fit, s = fit$lambda.min)

coef\_data <- data.frame(variable = rownames(coeficients)[-1], coefficient = coeficients[-1])

# Sort coefficients by absolute value

coef\_data <- coef\_data[order(abs(coef\_data$coefficient), decreasing = TRUE), ]

View(coef\_data)

# Select top 10 features

top\_10 <- coef\_data$variable[1:10]

# Select top 10 features from training and testing data

train\_data\_10<- train\_data[, c(top\_10, "X.Retained.F17.F18...1.yes..0.no..")]

test\_data\_10 <- test\_data[, c(top\_10, "X.Retained.F17.F18...1.yes..0.no..")]

# Fit logistic regression model on training data with top features

x\_train\_10 <- as.matrix(train\_data\_10[, -ncol(train\_data\_10)])

y\_train\_10 <- train\_data\_10$X.Retained.F17.F18...1.yes..0.no..

fit\_10 <- cv.glmnet(x\_train\_10, y\_train\_10, family = "binomial", type.measure = "class")

# Predict on test data and calculate accuracy

x\_test\_10 <- as.matrix(test\_data\_10[, -ncol(test\_data\_10)])

y\_test\_10 <- test\_data\_10$X.Retained.F17.F18...1.yes..0.no..

y\_pred\_10 <- predict(fit\_10, newx = x\_test\_10, s = fit\_10$lambda.min, type = "class")

accuracy\_10 <- sum(y\_pred\_10 == y\_test\_10) / length(y\_test\_10)

accuracy\_10

F1\_Score(y\_pred\_10,y\_test\_10)