DMPM Lab Assignment -5

Decision Classification Model

Name :-Sagnik Ghosh
Roll no:-8
SRN:-202100424

- Read the dataset that is provided to you.

```
library(tidyverse)
library(rpart)
library(rpart.plot)
library(caTools)



# Read the dataset
Titanic_dataset <- read.csv("D:/Users/sagni/Downloads/Titanic_dataset.csv")
```

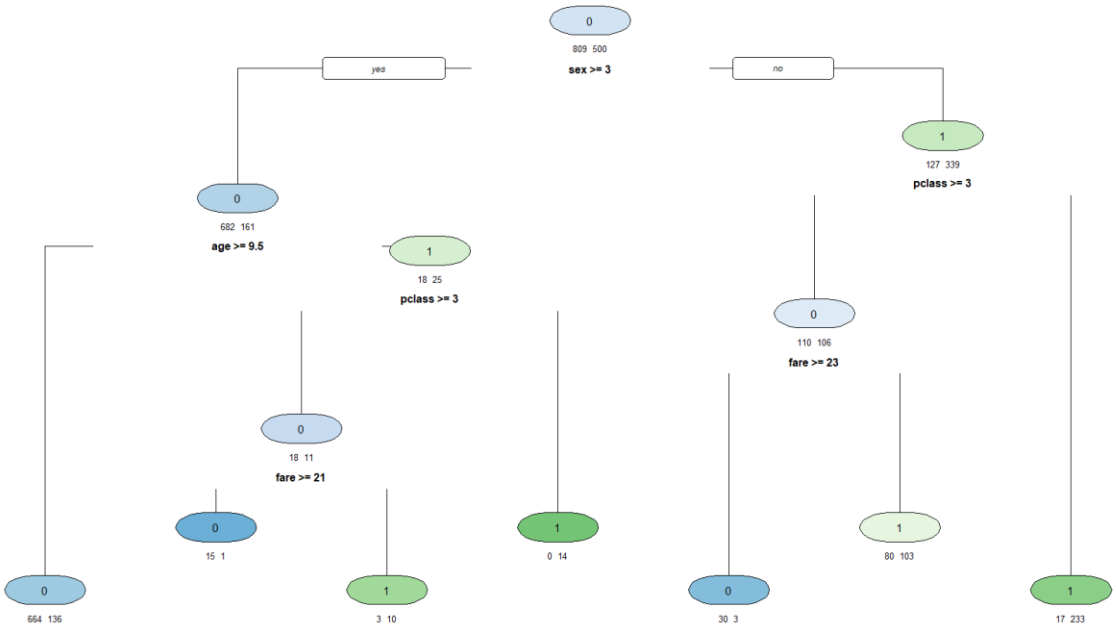
- Build a suitable decision tree predictive model to predict respective target values based on predictive features.

```
# Preprocess the data
selected_features <- Titanic_dataset %>%
  select(age, sex, pclass, fare, survived) %>%
  mutate(age = ifelse(is.na(age), median(age, na.rm = TRUE), age)) %>%
  mutate(sex = as.numeric(factor(sex))) # Encode categorical variable
# Split the dataset into features and target variable
X <- selected_features[, c("age", "sex", "pclass", "fare")]
y <- selected_features$survived
# Split the dataset into training and testing sets
set.seed(123)
split <- sample.split(y, SplitRatio = 0.7)
X_train <- X[split, ]
X_test <- X[!split, ]
y_train <- y[split]
y_test <- y[!split]
# Train Decision Tree Classifier
```

```
DT_clf <- rpart(survived ~ age + sex + pclass + fare, data = selected_features, method =
                "class")
# Prediction of survivals
predictions <- predict(DT_clf, X_test, type = "class")
# Remove observations with missing values from y_test and X_test
y_test <- y_test[!is.na(y_test)]
X_test <- X_test[!is.na(y_test), ]
# Convert y_test to factor
y_test <- factor(y_test)
```

- Plot the decision tree and develop some metrics to determine the accuracy of your model. (Compute various evaluation parameters of the tree model built).

```
# Plot Decision Tree
rpart.plot(DT_clf, extra = 1, under = TRUE, faclen = 0, cex = 0.8)
```



- Cross validate and optimize the model using hold back K-fold technique.

```
# Check for missing values again
print(colSums(is.na(selected_features)))
# Impute missing value in the "fare" column with the median
selected_features$fare[is.na(selected_features$fare)] <-
  median(selected_features$fare, na.rm = TRUE)
# Check for missing values again
print(colSums(is.na(selected_features)))
```

```
# Convert 'survived' to a factor with two levels
selected_features$survived <- factor(selected_features$survived)
# Remove rows with missing values
selected_features <- na.omit(selected_features)
# Cross-validate and optimize the model for classification
set.seed(123)
ctrl <- trainControl(method = "cv", number = 10)
DT_model_cv <- train(survived ~ ., data = selected_features, method = "rpart",
                     trControl = ctrl)
# Print the results
print(DT_model_cv)
# Print cross-validated accuracy
print(paste("Cross-validated Accuracy:", DT_model_cv$results$Accuracy))
```

```
CART

1309 samples
   4 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1178, 1178, 1178, 1178, 1178, 1178, ...
Resampling results across tuning parameters:

  cp     Accuracy   Kappa
  0.014  0.7967528  0.5618686
  0.027  0.7853142  0.5373980
  0.424  0.7227246  0.3525287

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.014.
> # Print cross-validated accuracy
> print(paste("Cross-validated Accuracy:", DT_model_cv$results$Accuracy))
[1] "Cross-validated Accuracy: 0.796752789195537"
[2] "Cross-validated Accuracy: 0.785314151497358"
[3] "Cross-validated Accuracy: 0.722724603640634"
```

INTERPRETATION:

The accuracy achieved in the first fold of cross-validation is approximately 0.797.

The accuracy achieved in the second fold of cross-validation is approximately 0.785.

The accuracy achieved in the third fold of cross-validation is approximately 0.723.

- Use method of pruning to avoid over-fitting and derive the best size of the decision tree.

```
> # Prune the Decision Tree
> DT_pruned <- prune(DT_clf, cp = DT_model_cv$bestTune$cp)
> DT_pruned
n= 1309

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1309 500 0 (0.61802903 0.38197097)
   2) sex>=1.5 843 161 0 (0.80901542 0.19098458) *
   3) sex< 1.5 466 127 1 (0.27253219 0.72746781)
     6) pclass>=2.5 216 106 0 (0.50925926 0.49074074)
      12) fare>=23.35 33   3 0 (0.90909091 0.09090909) *
      13) fare< 23.35 183  80 1 (0.43715847 0.56284153) *
     7) pclass< 2.5 250  17 1 (0.06800000 0.93200000) *
```