MLPH FINAL PROJECT

Keith Lee & Etornam Amesimeku

2023-04-28

```
library(haven)
library(MASS)
library(tidyverse) ## Data  manipulation
library(caret) ## KNN
library(dplyr)
library(caTools)
library(randomForest)
library(ggplot2)
library(reshape2)
library(gridExtra)
library(corrplot)
library(FNN)
library(class)
library(rpart)
library(randomForest)
library(ISLR)
library(boot)
library(pROC)
library(ROCR)
library(glmnet)
```

Read dataset into R

```
library(readr)
lung_cancer<- read_csv("C:/Users/khlee/OneDrive/Documents/GWANGJAAA/NYU/Spring/ML/survey lung cancer.csv

## Checking for missing value and duplicates
sum(is.na(lung_cancer)) ##Should return 0
```

```
## [1] 0
```

Recoding variables

```
lung_cancer$GENDER[lung_cancer$GENDER=="M"]<-1
lung_cancer$GENDER[lung_cancer$GENDER=="F"]<-0
lung_cancer$LUNG_CANCER[lung_cancer$LUNG_CANCER=="YES"]<-1
lung_cancer$LUNG_CANCER[lung_cancer$LUNG_CANCER=="NO"]<-0
```

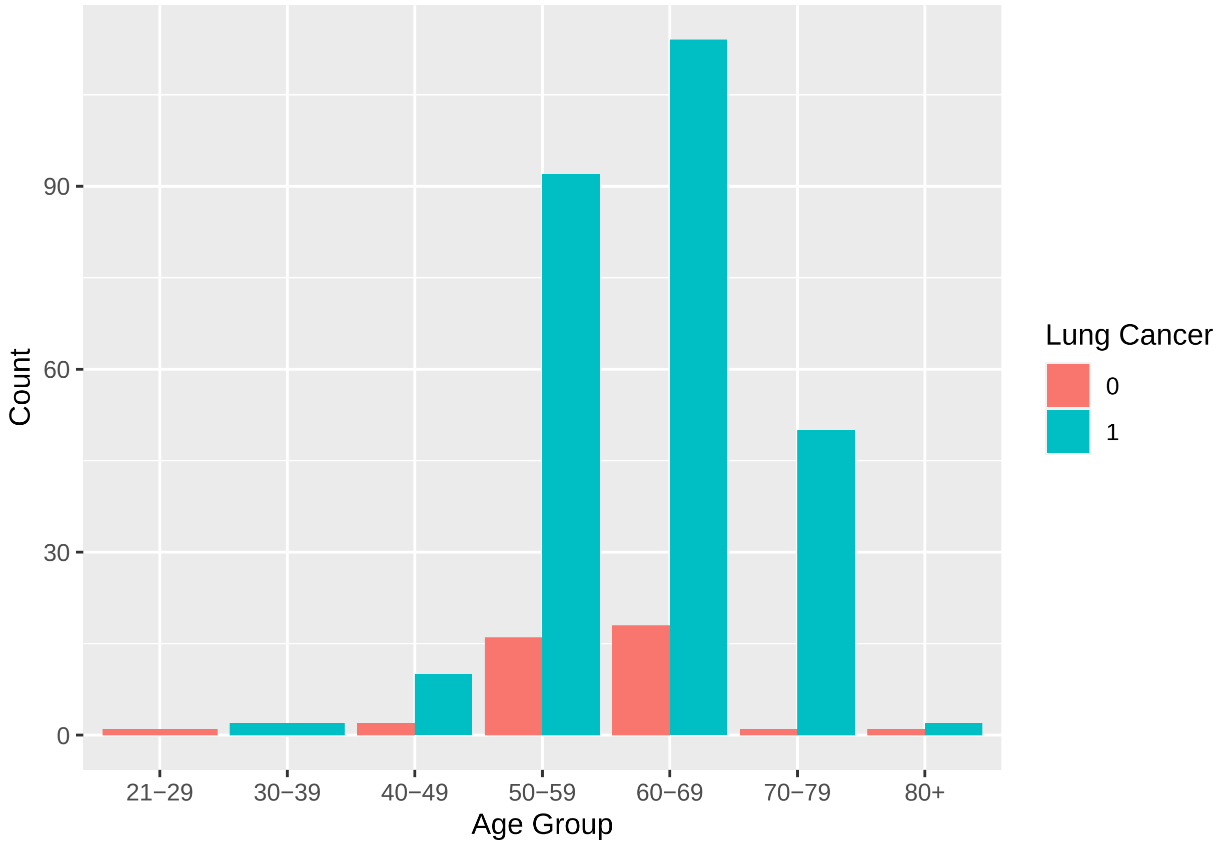
```
# Create a new variable with the encoded LUNG_CANCER variable
lung_cancer$LUNG_CANCER <- factor(lung_cancer$LUNG_CANCER)

# Create a new variable with the encoded GENDER variable
lung_cancer$GENDER <- as.numeric(lung_cancer$GENDER)
```

Bar plot showing the age distribution of Age and lung cancer

```
# Create age categories
lung_cancer$AGE <- cut(lung_cancer$AGE, breaks = c(20, 30, 40, 50, 60, 70, 80, 90),
                                  labels = c("21-29", "30-39", "40-49", "50-59", "60-69", "70-79", "80+

# Create bar chart
ggplot(lung_cancer, aes(x = AGE, fill = LUNG_CANCER)) +
  geom_bar(position = "dodge") +
  labs(x = "Age Group", y = "Count", fill = "Lung Cancer")
```

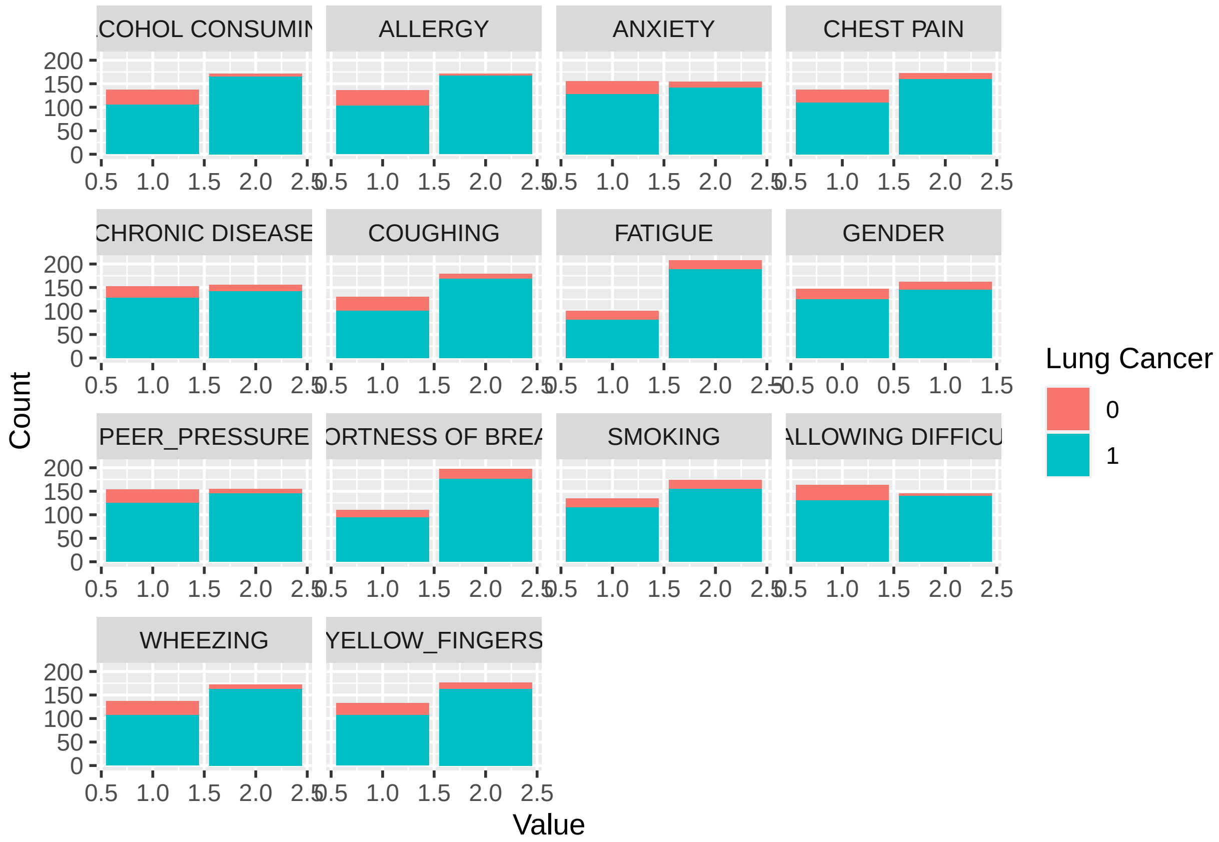


Exploratory Data Analysis(EDA)

```
library(ggplot2)
library(tidyr)
```

```
# Gather data into long format
lung_cancer_long <- lung_cancer %>%
  pivot_longer(cols = -c("AGE","LUNG_CANCER"), names_to = "variable", values_to = "value")

# Create bar chart
ggplot(lung_cancer_long, aes(x = value, fill = LUNG_CANCER)) +
  geom_bar() +
  facet_wrap(~ variable, scales = "free_x") +
  labs(x = "Value", y = "Count", fill = "Lung Cancer")
```



**Some of the dataset are skewed and imbalanced. Greater numbers of samples contain lung cancer.

Factoring variables

```
lung_cancer$GENDER <- factor(lung_cancer$GENDER)
lung_cancer$LUNG_CANCER <- factor(lung_cancer$LUNG_CANCER)
lung_cancer$SMOKING<-factor(lung_cancer$SMOKING)
lung_cancer$YELLOW_FINGERS<-factor(lung_cancer$YELLOW_FINGERS)
lung_cancer$ANXIETY<-factor(lung_cancer$ANXIETY)
lung_cancer$PEER_PRESSURE<-factor(lung_cancer$PEER_PRESSURE)
lung_cancer$`CHRONIC DISEASE`<-as.numeric(lung_cancer$`CHRONIC DISEASE`)
lung_cancer$FATIGUE<-factor(lung_cancer$FATIGUE)
```

```
lung_cancer$ALLERGY<-factor(lung_cancer$ALLERGY)
lung_cancer$WHEEZING<-factor(lung_cancer$WHEEZING)
lung_cancer$`ALCOHOL CONSUMING`<-factor(lung_cancer$`ALCOHOL CONSUMING`)
lung_cancer$COUGHING<-factor(lung_cancer$COUGHING)
lung_cancer$`SHORTNESS OF BREATH`<-factor(lung_cancer$`SHORTNESS OF BREATH`)
lung_cancer$`SWALLOWING DIFFICULTY`<-factor(lung_cancer$`SHORTNESS OF BREATH`)
lung_cancer$`CHEST PAIN`<-factor(lung_cancer$`CHEST PAIN`)
lung_cancer$AGE<-as.numeric(lung_cancer$AGE)
```

Data processing

Splitting dataset into train (70%) and test(30%)

```
set.seed(123)
split <- sample.split(lung_cancer$LUNG_CANCER, SplitRatio = 0.7)
lung_cancer_tr <- subset(lung_cancer, split == TRUE)
lung_cancer_te <- subset(lung_cancer, split == FALSE)
```

scaling age variable

```
# Scale the AGE column in lung_cancer_tr and lung_cancer_te
lung_cancer_tr <- lung_cancer_tr %>%
  mutate(AGE = scale(AGE))

lung_cancer_te <- lung_cancer_te %>%
  mutate(AGE = scale(AGE))
head(lung_cancer_tr)
```

```
## # A tibble: 6 x 16
##   GENDER AGE[,1] SMOKING YELLO~1 ANXIETY PEER_~2 CHRON~3 FATIGUE ALLERGY WHEEZ~4
##   <fct>    <dbl> <fct>   <fct>   <fct>   <fct>     <dbl> <fct>   <fct>   <fct>
## 1 1        0.314 1       2       2       1             1 2       1       2
## 2 1        0.314 2       2       2       1             1 1       1       1
## 3 0        0.314 1       2       1       1             1 1       1       2
## 4 0        1.41  1       2       1       1             2 2       2       2
## 5 0        0.314 2       1       2       1             1 2       1       1
## 6 1       -0.779 2       2       2       2             2 1       2       1
## # ... with 6 more variables: ‘ALCOHOL CONSUMING‘ <fct>, COUGHING <fct>,
## #   ‘SHORTNESS OF BREATH‘ <fct>, ‘SWALLOWING DIFFICULTY‘ <fct>,
## #   ‘CHEST PAIN‘ <fct>, LUNG_CANCER <fct>, and abbreviated variable names
## #   1: YELLOW_FINGERS, 2: PEER_PRESSURE, 3: ‘CHRONIC DISEASE‘, 4: WHEEZING
```

Model Building

Run 10 fold cross validation algorithm

KNN model

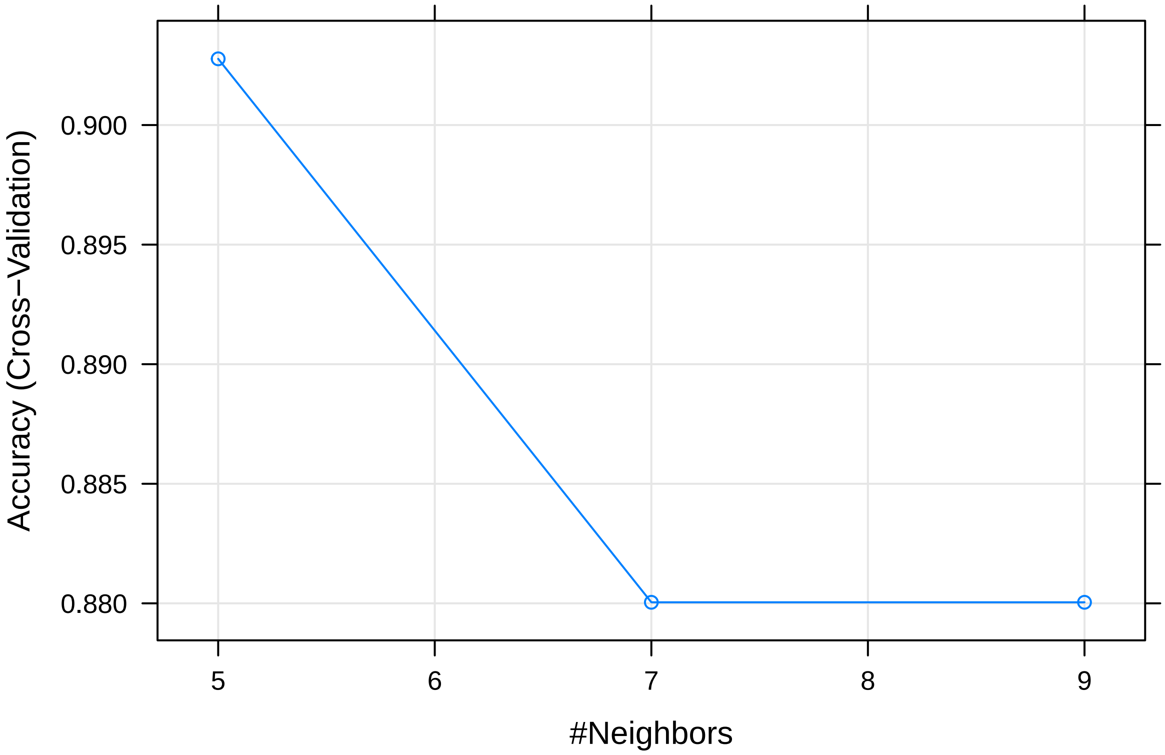
```
set.seed(123)

# define the training control
train_control <- trainControl(method = "cv", number = 10)

# K-Nearest Neighbor model
knn_model <- train(LUNG_CANCER ~ ., data = lung_cancer_tr, method = "knn", trControl = train_control)
knn_pred <- predict(knn_model, newdata = lung_cancer_te)
knn_pred_prob <- predict(knn_model, newdata = lung_cancer_te, type = "prob")
knn_accuracy <- sum(diag(table(knn_pred, lung_cancer_te$LUNG_CANCER)))/nrow(lung_cancer_te)
knn_accuracy
```

```
## [1] 0.8924731
```

```
plot(knn_model)
```



```
knn_model
```

```
## k-Nearest Neighbors
##
## 216 samples
##  15 predictor
##   2 classes: ’0’, ’1’
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 194, 194, 194, 196, 194, 195, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9027706  0.3682591
##   7  0.8800433  0.1692747
##   9  0.8800433  0.1756460
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

Random Forest

```
set.seed(7)
rf_model <- train(LUNG_CANCER~., data=lung_cancer_tr, method="rf", trControl=train_control)
rf_pred <- predict(rf_model, newdata = lung_cancer_te)
rf_pred_prob <- predict(rf_model, newdata = lung_cancer_te, type = "prob")
rf_accuracy <- sum(diag(table(rf_pred, lung_cancer_te$LUNG_CANCER)))/nrow(lung_cancer_te)
rf_accuracy
```

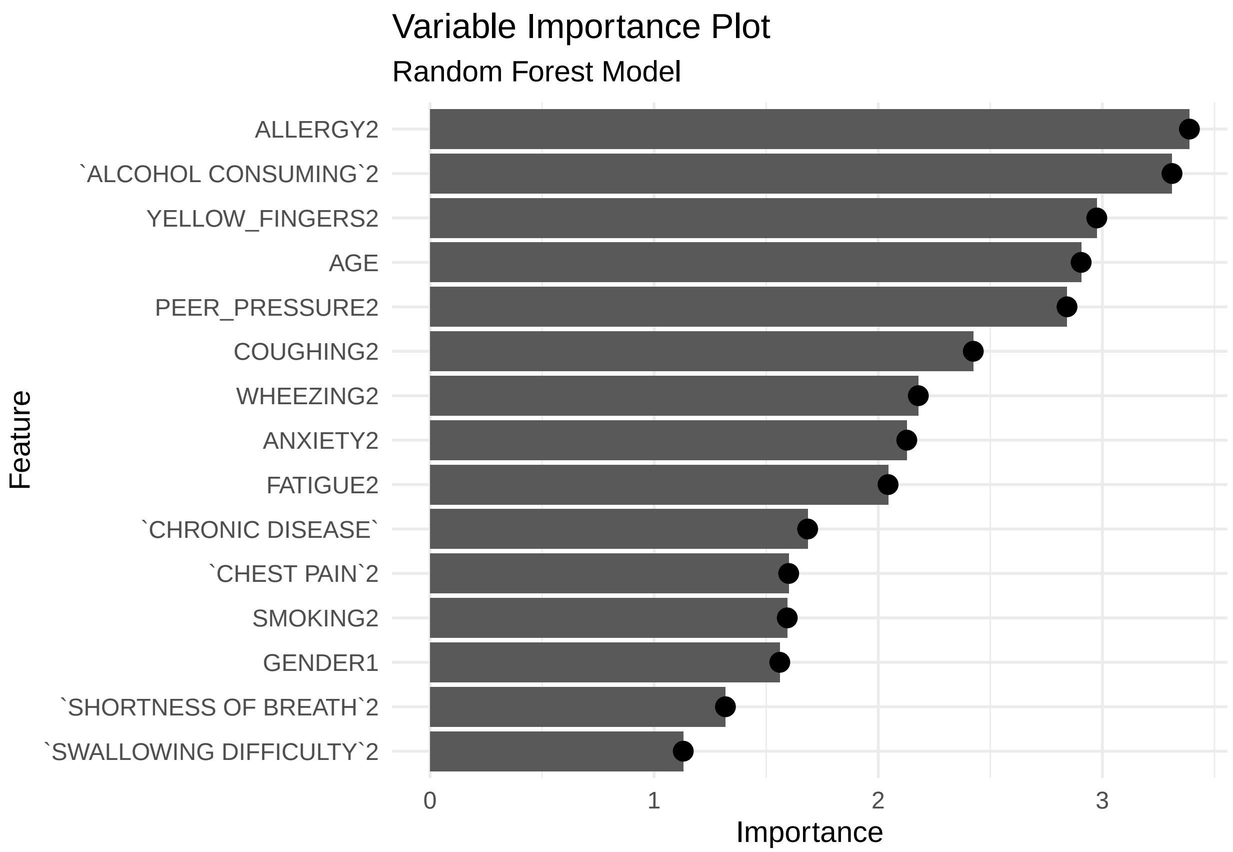
```
## [1] 0.9354839
```

Random forest plot

```
set.seed(7)
rf_model <- train(LUNG_CANCER ~ ., data = lung_cancer_tr, method = "rf", trControl = train_control)

# variable importance information
var_imp <- varImp(rf_model, scale = FALSE)

# Plotting variable importance
ggplot(var_imp, aes(x = Overall, y = names, color = Overall)) +
  geom_point(size = 3) +
  labs(title = "Variable Importance Plot",
       subtitle = "Random Forest Model") +
  theme_minimal()
```



Decision Tree

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.3
```

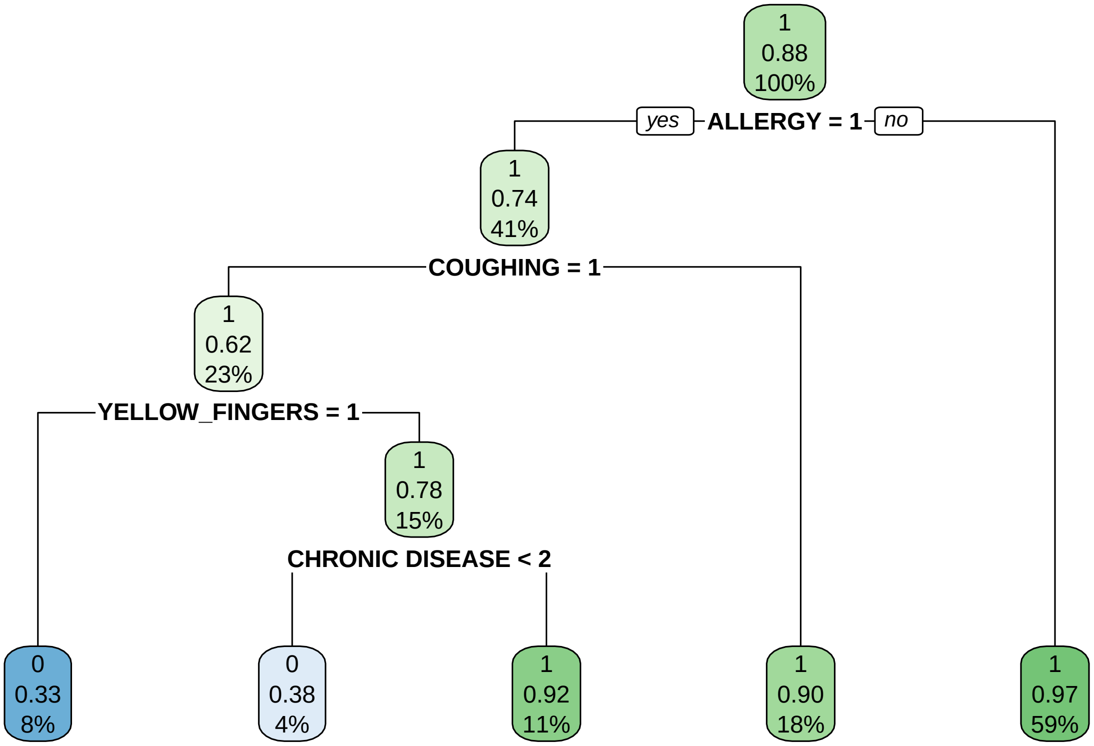
```
dt_model <- rpart(LUNG_CANCER ~ ., data = lung_cancer_tr, method = "class")
dt_pred <- predict(dt_model, lung_cancer_te, type = "class")
dt_confusion <- table(dt_pred, lung_cancer_te$LUNG_CANCER)
dt_confusion
```

```
##
## dt_pred  0  1
##       0  7  8
##       1  5 73
```

```
dt_accuracy <- sum(diag(dt_confusion))/sum(dt_confusion)
dt_accuracy
```

```
## [1] 0.8602151
```

```
rpart.plot(dt_model)
```



```
dt_pred_prob <- predict(dt_model, lung_cancer_te, type = "prob")[,2]
```

Logistic Regression model

```
glm_model <- train(LUNG_CANCER ~ ., data = lung_cancer_tr, method = "glmnet", family = "binomial", trCo
glm_pred <- predict(glm_model, newdata = lung_cancer_te)
glm_pred_prob <- predict(glm_model, newdata = lung_cancer_te, type = "prob")
glm_accuracy <- sum(diag(table(glm_pred, lung_cancer_te$LUNG_CANCER)))/nrow(lung_cancer_te)
glm_accuracy
```

```
## [1] 0.9032258
```

Models' comparison in terms of accuracy

```
# Create a table to compare model accuracy, test error, and train error
results_table <- data.frame(Model = c("KNN", "Decision Trees", "Logistic Regression", "Random Forest"),
                            Accuracy = c(knn_accuracy, dt_accuracy, glm_accuracy, rf_accuracy),
                            Test_Error = c(1 - knn_accuracy, 1 - dt_accuracy, 1 - glm_accuracy, 1 - rf_a
                            Train_Error = c(1 - knn_model$results$Accuracy[1], 1 - dt_model$cptable[whic
                                             1 - glm_model$results$Accuracy[1], 1 - rf_model$results$Acc

results_table
```

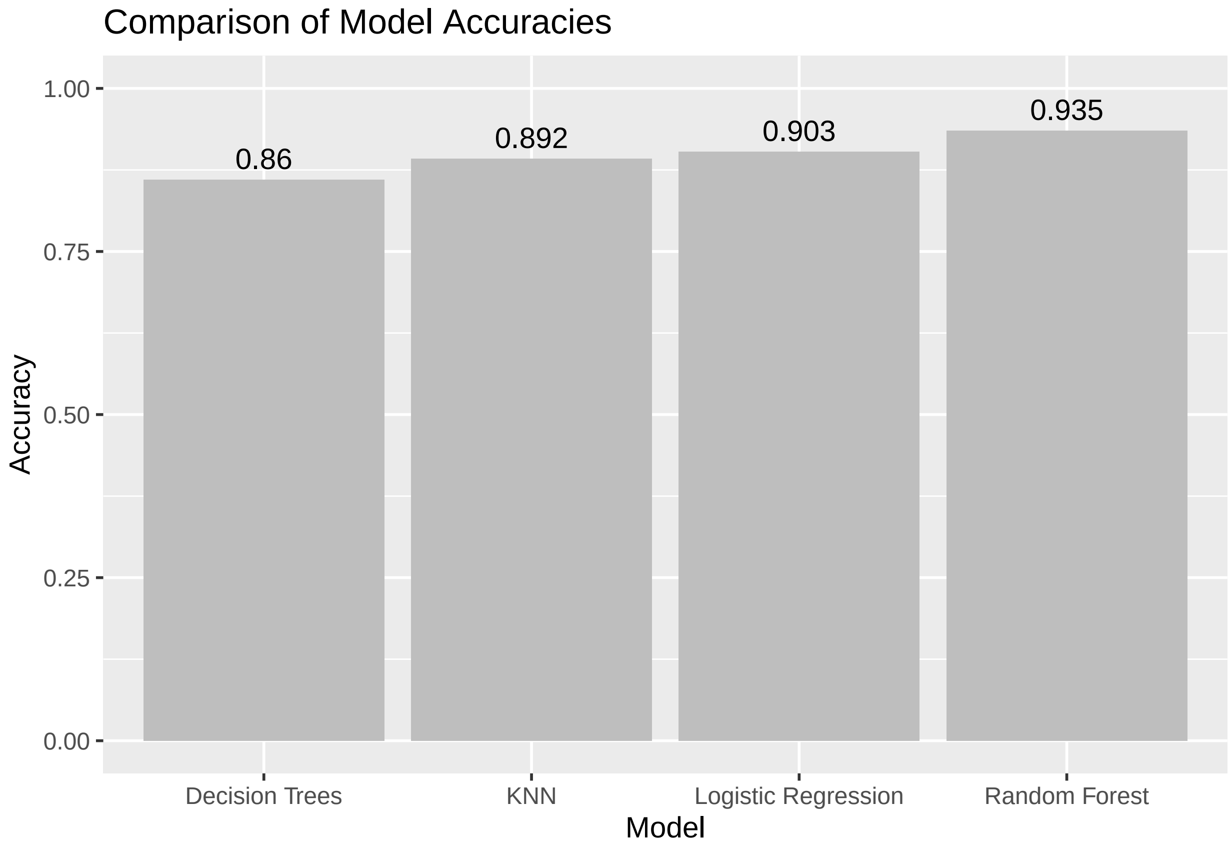
```
##                 Model  Accuracy Test_Error Train_Error
## 1                 KNN 0.8924731 0.10752688  0.09722944
## 2      Decision Trees 0.8602151 0.13978495  0.00000000
## 3 Logistic Regression 0.9032258 0.09677419  0.08268398
## 4       Random Forest 0.9354839 0.06451613  0.08311688
```

Model performance visualization

```
library(ggplot2)

# Create a data frame of model names and accuracies
accuracy_df <- data.frame(Model = c("KNN", "Decision Trees", "Logistic Regression", "Random Forest"),
                          Accuracy = c(knn_accuracy, dt_accuracy, glm_accuracy, rf_accuracy))

# Create the plot
ggplot(accuracy_df, aes(x = Model, y = Accuracy)) +
  geom_bar(stat = "identity", fill = "gray") +
  geom_text(aes(label = round(Accuracy, 3)), vjust = -0.5) +
  ylim(c(0, 1)) +
  labs(x = "Model", y = "Accuracy", title = "Comparison of Model Accuracies")
```



AUC ROC curve for all models

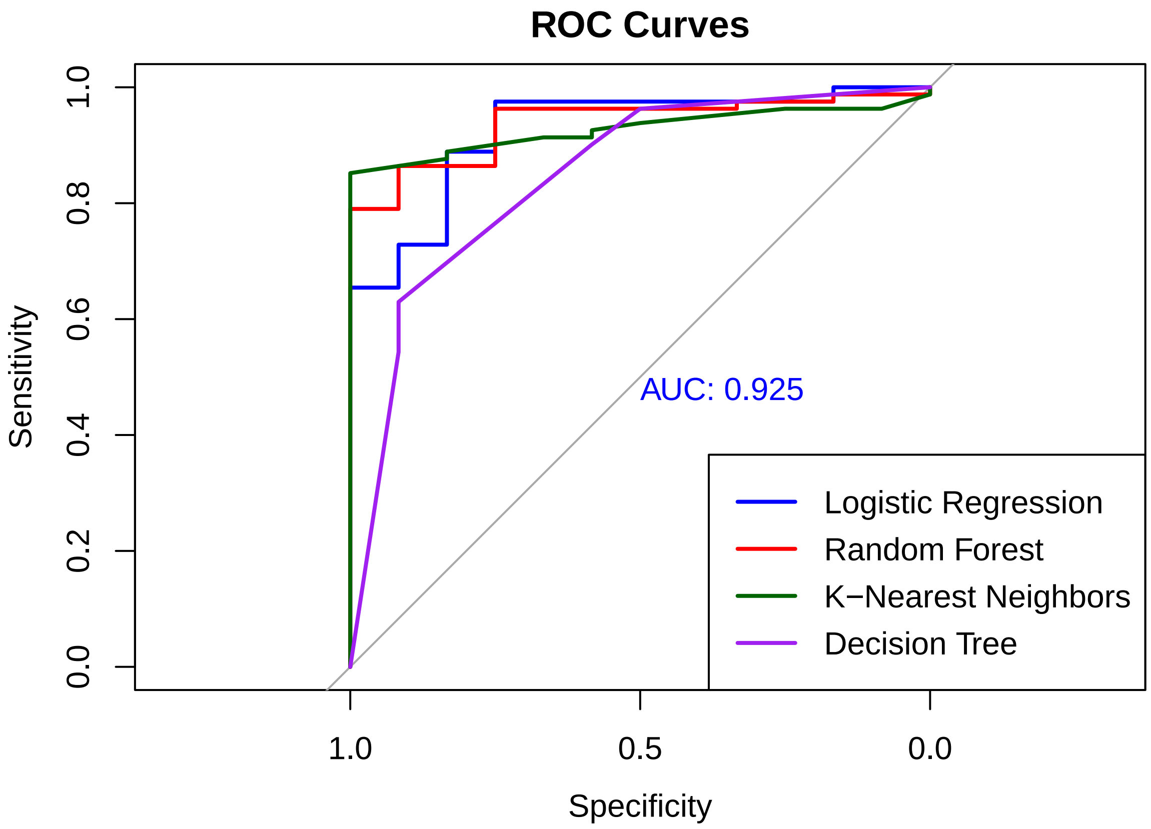
```
# # Calculate AUC for Random Forest model
# auc_rf <- roc(lung_cancer_te$LUNG_CANCER, rf_pred_prob[,2])
# auc_rf
#
# # Plot ROC curve for Random Forest model
# plot(auc_rf, main = "Random Forest Model ROC Curve", print.auc = TRUE, legacy.axes = TRUE, col="#D55E
# legend("bottomright", legend = paste("AUC =", round(auc_rf$auc,2)), col = "#D55E00", lwd = 2, cex=0.8

# Calculate ROC with each prediction probability and the true values
roc_obj_glm <- roc(lung_cancer_te$LUNG_CANCER, glm_pred_prob[,2])
roc_obj_rf <- roc(lung_cancer_te$LUNG_CANCER, rf_pred_prob[,2])
roc_obj_knn <- roc(lung_cancer_te$LUNG_CANCER, knn_pred_prob[,2])
roc_obj_dt <- roc(lung_cancer_te$LUNG_CANCER, dt_pred_prob)

# Plot ROC curve
plot(roc_obj_glm, main="ROC Curves", col="blue", print.auc=TRUE)
lines(roc_obj_rf, col="red", print.auc=TRUE)
lines(roc_obj_knn, col="darkgreen", print.auc=TRUE)
lines(roc_obj_dt, col="purple", print.auc=TRUE)

# Add a legend
```

```
legend("bottomright", legend=c("Logistic Regression", "Random Forest", "K-Nearest Neighbors", "Decision
       col=c("blue", "red", "darkgreen", "purple"), lwd=2)
```



‘