Indian Liver Patient

Pinar Asker

7/12/2021

1. INTRODUCTION

The "Indian Liver Patient Records" dataset available on the Kaggle website was used in this project. The project aims to examine the factors related to liver cancer for a group of people from India and develop machine learning models based on these factors to predict whether a person has liver disease or not. Thus, the inspiration of this project is to provide statistical modeling for liver disease detection and help doctors predict the disease. The dataset consists of 416 liver patient records and 167 non-liver patient records, along with related variables. There are 11 variables which are,

• Age of the patient • Gender of the patient • Total Bilirubin • Direct Bilirubin • Alkaline Phosphatase • Alamine Aminotransferase • Aspartate Aminotransferase • Total Proteins • Albumin • Albumin and Globulin Ratio • Disease: field used to split the data into two sets (patient with liver disease or no disease)

The scientific definition of the variables above is beyond this project's scope. Instead, I characterized variables are only by quantitative or qualitative variables in this project. Quantitative variables are numerical variables, whereas qualitative variables take on categorical variables. "Dataset" variable, which indicates a patient with liver disease or no disease, and "Gender" variables are qualitative, and remaining variables are quantitative.

In the statistical models in this project, the "Disease" variable is used as a dependent(response) variable, and the remaining variables are used as dependent variables. I am interested in the prediction of the dependent variable ("Disease") based on independent variables (remaining variables).

Since the response (dependent) variable is qualitative(categorical), I considered classification models in this project. Classification models assign each record to a class that is most likely based on dependent variables. However, linear regression models are also can be considered for categorical response variables.

I built two different classification models to predict the "Disease." First, I built logistic regression models with varying combinations of dependent variables and with different thresholds values. Second, I built KNN models with the different "k" values.

In order to evaluate how well a model's prediction matches the actual data, I used three measures to quantify it such as model accuracy, specificity, and sensitivity. Accuracy is a measure of fit of a model, and it is a proportion of correctly predicted values to the total number of predicted values. Specificity refers to the percentage of predicting liver patients as liver patients. Sensitivity is the percentage of predicting non-liver patients as non-liver patient in this project.

I obtained the models' accuracy, specificity, and sensitivity values by applying these models to test data.

I concluded the final model as the logistic regression model with an accuracy of 0.752.

2. DATA LOADING

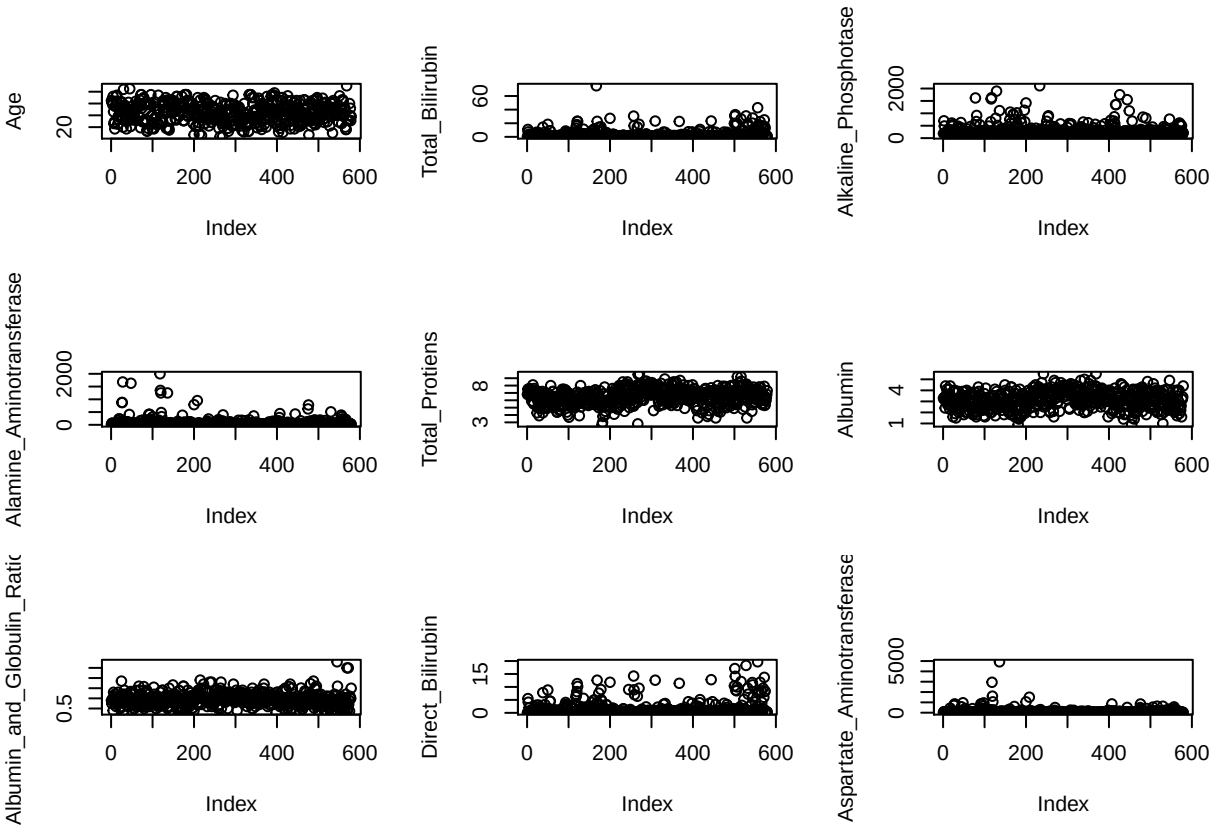
As the first step, I loaded the data into R. Second, I checked for missing values and removed them from the data. In addition, I checked data for columns names, types of variables, and any anomalies that might be in

the dataset.

3. VISUAL SUMMARY OF THE DATA

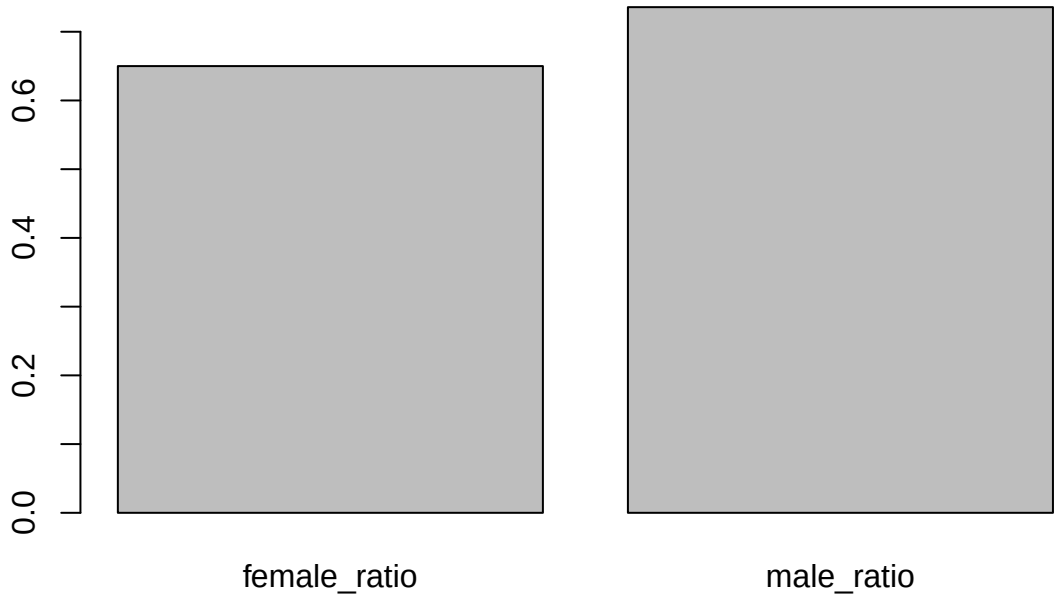
Visual summaries can be very helpful to understand the data and reveal the relationship between variables.

3.1.Scatterplots



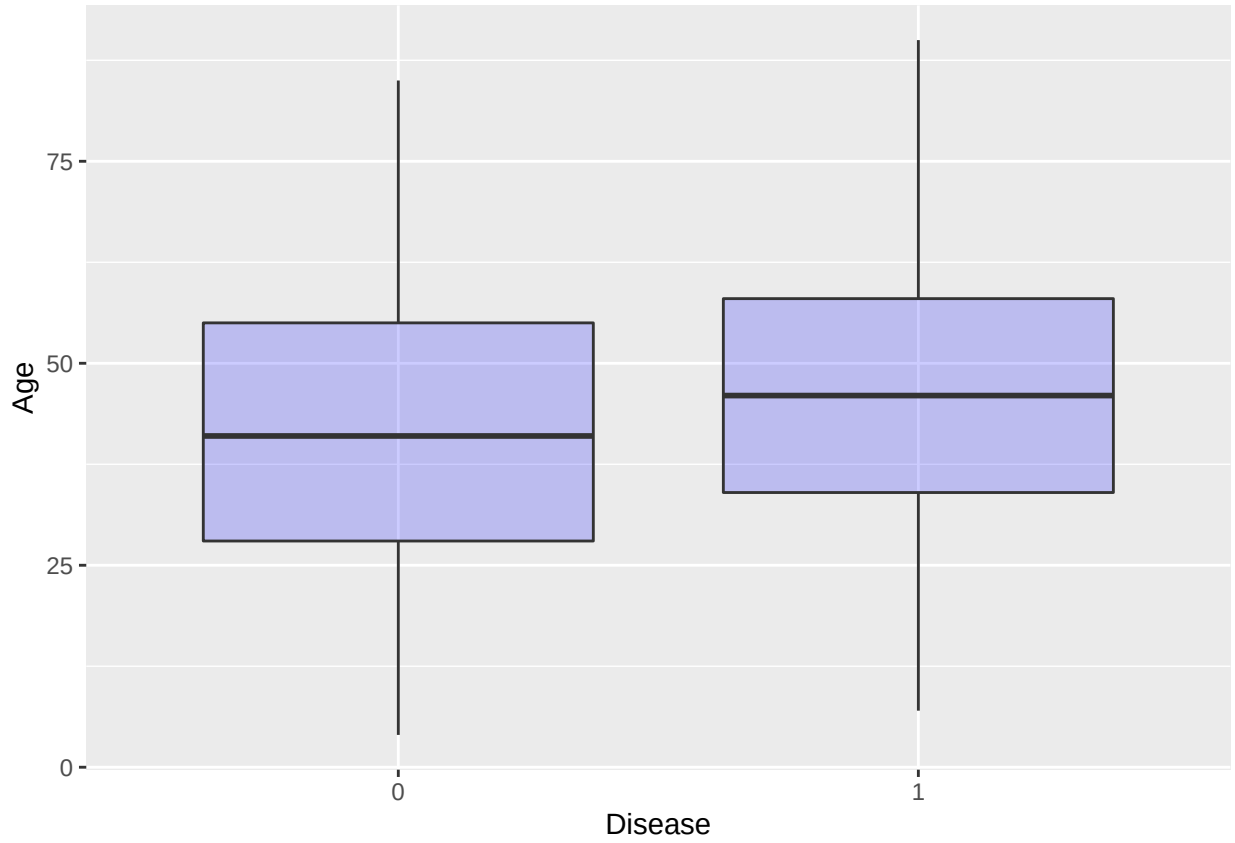
The chart above shows the scatterplot of each forecast variable in the dataset. There seem to be some high leverage points in the dataset; in other words, some observations in the data set are far away from other observations. However, we do not have enough information to decide whether these are true outliers or not. Therefore, I did not remove these data points from the dataset.

3.2. The Effect of the Gender on Liver Disease



According to the graph above, males tend to have slightly more liver disease than women. Since there seems no significant difference between the two genders in liver disease, I excluded the gender from the models.

3.3. The Effect of Age on Liver Disease



The boxplots show older patients more subject to the disease compared to younger patients in the dataset. Therefore, I include the "Age" variable in the models.

4. DATA ANALYSIS

4.1 Data Partitioning

I divided the data into train set and test set by 30 percent and 70 percent, respectively. Although the data can be divided by other percentages, I decided to split the dataset by this ratio due to the limited observation in the data.

```
set.seed(1)
test_index<-createDataPartition(data$Disease,times=1,p=0.3,list=F)
train_data<-data %>% slice(-test_index)
test_data<-data %>% slice(test_index)
dim(train_data)
dim(test_data)
```

4.2. Logistic Regression and Collinearity Diagnosis

4.2.1. Model 1 : Logistic Regression Model with All Variables

First, I built a logistic regression model with all variables in the dataset.

```
glm_fit_1<-glm(Disease~.,train_data, family=binomial)
summary(glm_fit_1)
```

```
##
## Call:
## glm(formula = Disease ~ ., family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3032  -1.0436   0.3221   0.8976   1.5130
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -3.468466   1.671076  -2.076   0.0379 *
## Age                         0.021284   0.007919   2.688   0.0072 **
## GenderMale                 -0.002745   0.289992  -0.009   0.9924
## Total_Bilirubin             0.168090   0.337433   0.498   0.6184
## Direct_Bilirubin            0.303504   0.585290   0.519   0.6041
## Alkaline_Phosphotase        0.002742   0.001215   2.256   0.0240 *
## Alamine_Aminotransferase    0.010694   0.005928   1.804   0.0712 .
## Aspartate_Aminotransferase  0.001044   0.003355   0.311   0.7555
## Total_Protiens              0.705397   0.467817   1.508   0.1316
## Albumin                    -1.308415   0.907529  -1.442   0.1494
## Albumin_and_Globulin_Ratio  1.366711   1.374991   0.994   0.3202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 481.42  on 404  degrees of freedom
## Residual deviance: 386.50  on 394  degrees of freedom
## AIC: 408.5
##
## Number of Fisher Scoring iterations: 9
```

The confusion matrix is shown below.

```
##           Reference
## Prediction   0   1
##          0  21  20
##          1  30 103
```

Evaluation of the model on the test data is shown below.

```
results=tibble(Model="glm_fit_1", Accuracy=cm1$overall["Accuracy"],
Sensitivity=cm1$byClass["Sensitivity"], Specificity= cm1$byClass["Specificity"])
results%>%knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity
glm_fit_1	0.7126437	0.4117647	0.8373984

Before moving forward on analysis, I checked the collinearity between variables to improve model. Collinearity means a close relationship between dependent variables (predictor variables) in the dataset. In other words, two or more predictor variables might have a linear relationship and move together in the same direction. Collinearity might be a problem in a regression model. Regression analysis can be adversely affected by collinearity because it may be hard to differentiate individual effects of variables in the model.

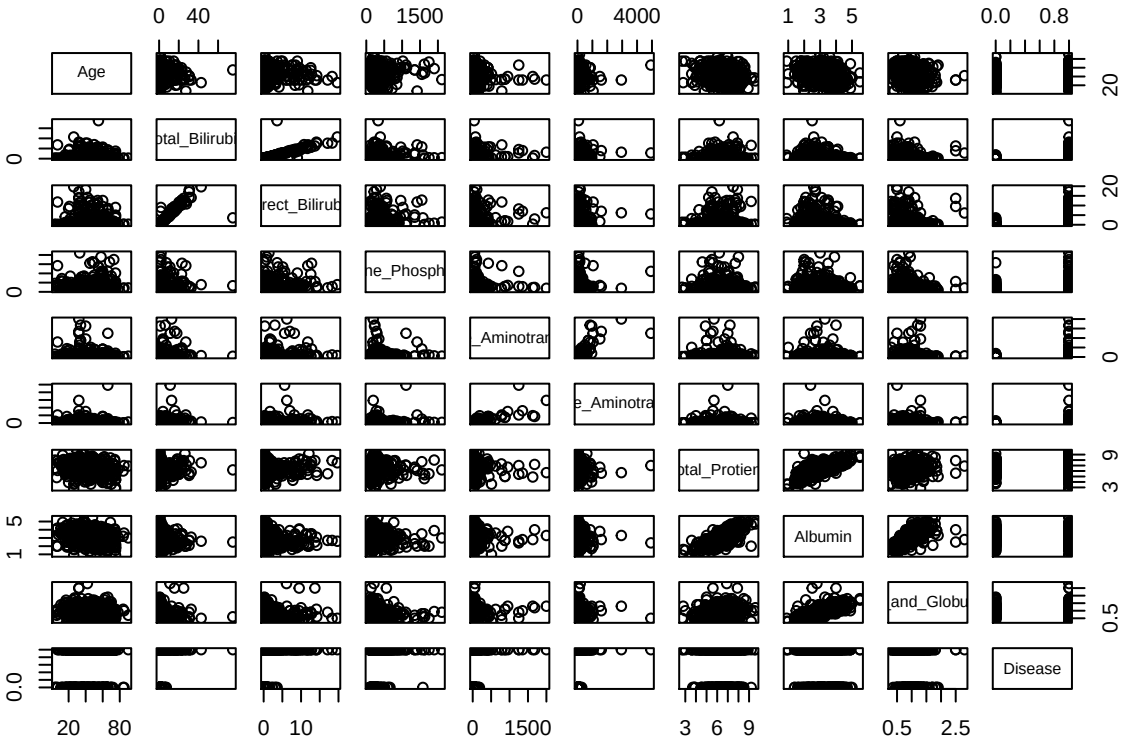
I used variance inflation factor (VIF) to quantify collinearity. VIF is a calculation for collinearity. As a thumb of rule, VIF values greater than 5 or 10 are considered as high collinearity between any two predictor variables in the model. I removed one of the variables with a VIF value greater than 10 and re-built the model with the remaining variables.

4.2.2 VIF Values to Detect Collineartiy

```
vif(glm_fit_1)
```

```
##                         Age                      Gender
##                    1.122887                    1.067053
##             Total_Bilirubin            Direct_Bilirubin
##                    7.991095                    7.954889
##       Alkaline_Phosphotase   Alamine_Aminotransferase
##                    1.128022                    2.082697
## Aspartate_Aminotransferase             Total_Protiens
##                    2.045726                   17.515456
##                     Albumin Albumin_and_Globulin_Ratio
##                   33.007903                   10.220959
```

```
pairs(data[,-2])
```



According to the VIF values and the matrix of the scatterplots above, "Total Bilirubin" and "Direct Bilirubin" are correlated. Additionally, "Total Proteins" and "Albumin" variables are also correlated. Therefore, I removed one of the correlated variables from the model. Those are "Total Bilirubin" and "Albumin." I also removed the "Gender" variable since there is no significant difference between the two genders in liver disease.

4.2.3. Model 2: Logistic Regression After Removing Correlated Predictor Variables

I built the logistic regression model again with remaining variables.

```
glm_fit_2<-glm(Disease~Age+Direct_Bilirubin+Alkaline_Phosphotase+
Alamine_Aminotransferase+Aspartate_Aminotransferase+
Total_Protiens+Albumin_and_Globulin_Ratio,train_data,family=binomial)

summary(glm_fit_2)
```

```
##
## Call:
## glm(formula = Disease ~ Age + Direct_Bilirubin + Alkaline_Phosphotase +
##     Alamine_Aminotransferase + Aspartate_Aminotransferase + Total_Protiens +
##     Albumin_and_Globulin_Ratio, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2759  -1.0812   0.3438   0.8910   1.5884
```

```
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.543709   1.046482  -1.475  0.14017
## Age                         0.021939   0.007879   2.784  0.00536 **
## Direct_Bilirubin            0.649538   0.219026   2.966  0.00302 **
## Alkaline_Phosphotase        0.002620   0.001201   2.182  0.02914 *
## Alamine_Aminotransferase    0.009132   0.005685   1.606  0.10820
## Aspartate_Aminotransferase  0.001861   0.003387   0.549  0.58275
## Total_Protiens              0.045419   0.120478   0.377  0.70618
## Albumin_and_Globulin_Ratio -0.469042   0.444065  -1.056  0.29086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 481.42  on 404  degrees of freedom
## Residual deviance: 388.90  on 397  degrees of freedom
## AIC: 404.9
##
## Number of Fisher Scoring iterations: 7
```

The confusion matrix of the model 2 is shown below.

```
##            Reference
## Prediction   0   1
##          0  19  20
##          1  32 103
```

Evaluation of the models on test data is shown below.

Model	Accuracy	Sensitivity	Specificity
glm_fit_1	0.7126437	0.4117647	0.8373984
glm_fit_2	0.7011494	0.3725490	0.8373984

4.2.4. Model 3: Logistic Regression After Removing Insignificant Variables

I removed the insignificant variables from the previous model according to the p-values and rebuilt the model.

```
glm_fit_3 <- glm(Disease~Age+Direct_Bilirubin+Alkaline_Phosphotase, train_data,
family=binomial)
summary(glm_fit_3)
```

```
##
## Call:
## glm(formula = Disease ~ Age + Direct_Bilirubin + Alkaline_Phosphotase,
##     family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0138  -1.1432   0.4102   0.9216   1.3716
```

```
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.397947   0.463334  -3.017 0.002552 **
## Age                   0.018245   0.007407   2.463 0.013773 *
## Direct_Bilirubin      0.853165   0.221603   3.850 0.000118 ***
## Alkaline_Phosphotase  0.003717   0.001239   3.000 0.002701 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 481.42  on 404  degrees of freedom
## Residual deviance: 402.25  on 401  degrees of freedom
## AIC: 410.25
##
## Number of Fisher Scoring iterations: 7
```

The confusion matrix of the model 3 is shown below.

```
##           Reference
## Prediction   0   1
##          0  18  10
##          1  33 113
```

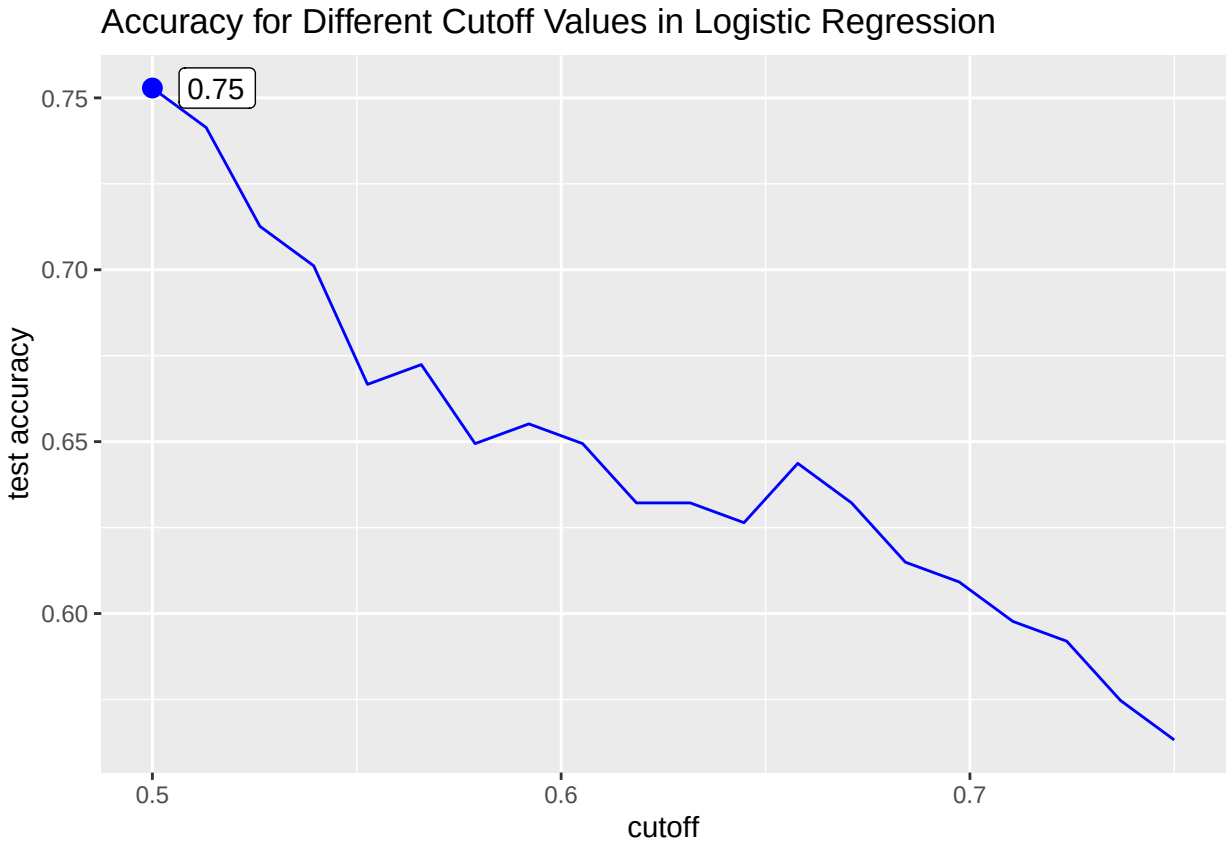
Evaluation of the models on test data is shown below.

Model	Accuracy	Sensitivity	Specificity
glm_fit_1	0.7126437	0.4117647	0.8373984
glm_fit_2	0.7011494	0.3725490	0.8373984
glm_fit_3	0.7528736	0.3529412	0.9186992

Now, all variables in the model are significant. I finalized the variable selection for logistic regression and built the model with only "Age", "Direct_Bilirubin", and "Alkaline_Phosphotase" variables.

I performed logistic regression with 0.5 as the cut-off for the predicted probability so far. Although logistic regression models are created with mostly 0.5 probability cut-off values, we can use other cut-off values to predict the classes. I used different thresholds ranging from 0.5 to 0.75 to predict liver cancer patients to be sure that 0.5 cutoff value is the best option for our model.

4.2.5. Different Predicted Probabilty Cutoff Values for Logistic Regression.



As we can see in the graph, the highest accuracy (0.75) was calculated with a threshold of 0.5. Therefore, I used 0.5 as the predicted probability cutoff for logistic regression to predict whether a person has liver disease or not.

4.3. KNN Models

The K-nearest neighbors method is the second classification model in this project.

4.3.1 Scaling the Variables

KNN models are based on the distance between variables. It predicts the class of a new (test) variable which is nearest to that point. Variables in different scales may lead to unrealistic results in the model, so all variables should be on the same scale. Therefore, I applied KNN models to Indian Liver Patient data after standardizing all quantitive predictor variables. Since variable 2 (Gender) and variable 11 (Disease) are categorical, I exclude them during standardization.

4.3.2 Data Partitioning

After standardization, I partitioned the data into a train set and a test set. The test set contains 30% of the total observations, and the train set contains the remaining observations.

```
set.seed(100)
test_index_st<-createDataPartition(data_standardized$Disease,times=1,p=0.3,list=F)
train_data_st<-data_standardized %>% slice(-test_index_st)
test_data_st<-data_standardized %>% slice(test_index_st)
train_data_st$Disease=as.factor(train_data_st$Disease)
test_data_st$Disease=as.factor(test_data_st$Disease)
```

4.3.3 The First KNN Model

I built k-nearest neighbors (kNN) model with k=5 to predict liver patients based on the variables in the dataset and test the final KNN model's accuracy on test data.

```
set.seed(1)
knn_fit_1 <- knn3(Disease ~ ., data = train_data_st, k=5)
y_hat_knn_1 <- predict(knn_fit_1, test_data_st, type = "class")
```

The confusion matrix is shown below.

```
##            Reference
## Prediction   0   1
##          0  20  25
##          1  27 102
```

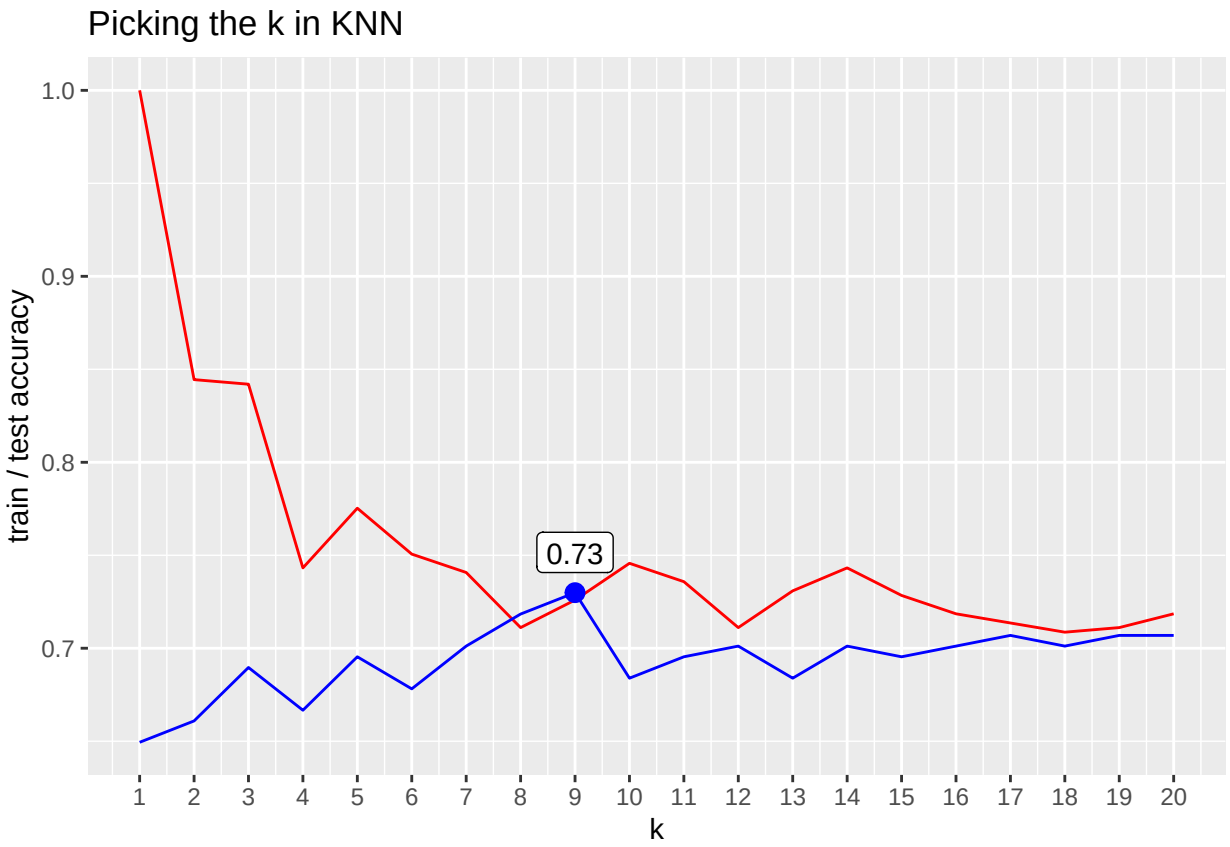
Evaluation of the models on test data is shown below.

Model	Accuracy	Sensitivity	Specificity
glm_fit_1	0.7126437	0.4117647	0.8373984
glm_fit_2	0.7011494	0.3725490	0.8373984
glm_fit_3	0.7528736	0.3529412	0.9186992
knn_1	0.7011494	0.4255319	0.8031496

4.3.4. Selecting optimum 'k'value.

'k' refers to the number of nearest neighbors in a KNN model. The choice of a "k" is crucial and can yield dramatically different results.

To pick up the best 'k' value, I built KNN models with the different 'k' values ranging from 1 to 20.



The red line and blue line reflect train accuracy and test accuracy, respectively. As seen in the graph, the best test accuracy was achieved with the 'k' value of 9. Therefore, I repeated the KNN model with 9 nearest neighbors.

```
set.seed(1)
knn_fit_1 <- knn3(Disease ~ ., data = train_data_st, k=9)
y_hat_knn_1 <- predict(knn_fit_1, test_data_st, type = "class")
```

The confusion matrix is shown below.

```
##            Reference
## Prediction   0   1
##          0  13  13
##          1  34 114
```

Evaluation of the models on test data is shown below.

Model	Accuracy	Sensitivity	Specificity
glm_fit_1	0.7126437	0.4117647	0.8373984
glm_fit_2	0.7011494	0.3725490	0.8373984
glm_fit_3	0.7528736	0.3529412	0.9186992
knn_1	0.7011494	0.4255319	0.8031496
knn_2	0.7298851	0.2765957	0.8976378

5. RESULTS AND CONCLUSION

The evaluation of all models is summarized in the table above. Values show the models' prediction performance on the test data. The first three models (glm_fit_1, 2, 3) were developed with a logistic regression model with a different set of predictors variables. The selection of the variables is detailed in the analysis section. The remaining models (knn_1 and knn_2) were developed with the KNN method.

According to the table, the glm_fit_3 model predicts the liver patient with the highest accuracy and Specificity among all models. However, this model comes at a cost on the sensitivity. Sensitivity is the percentage of predicting non-liver patients as non-liver patients in this project. Specificity refers to the percentage of predicting liver patients as liver patients.

There is always a trade-off between sensitivity and Specificity. Selecting the best model depends on the need of the project. In some projects, the sensitivity might be more critical compared to Specificity, or vice versa.

In this project, I decided on the final model based on accuracy. Therefore, I concluded that logistic regression model 3 (glm_fit_3) is the best model to predict liver patients based on their characteristics.

For future works, some other classification methods such as classification trees can be applied to improve accuracy.