Predicting heart disease using Machine Learning Models

Ashesh Shrestha

4/27/2021

Abstract

In this project I have tried to build the best model in terms of accuracy in order to predict the presence of heart disease in a person. In order to do so I have used data from University of California Irvine (UCI) repository which consists of data related to presence or absence of heart disease along with other 13 demographic and health attributes of 301 individuals. I used several classification models, namely, linear probability model, logistic regression model, bagging model, random forest model and gradient boosting model. Out of all these models, I get the best prediction from random forest model.

Introduction

Heart disease is the leading cause of death in the United States. According to Centerz for Disease Control and Prevention (CDC), about 6550,000 Americans die each year because of heart disease, which is 1 in every 4 deaths in the United States. There are mainly four types of heart disease: i. Coronary artery disease (CAD), ii. Valvular heart disease, iii. Arrhythmia, and iv. Heart failure. CAD is the most common type of heart disease in the United States. CAD is a condition in which plaque grows in the walls of the coronary arteries and limits the flow of blood to the heart's muscle. It can ultimately lead to heart attack.

In this project, my goal to build a classification model for prediction of heart disease. I have used 4 classification models namely, linear probability model, logistic regression model, random forest model and gradient boosting in order to predict the presence of heart disease in an individual.

Method

As mentioned above I have used 4 classification models. In order to fit these models I have used data from UCI repository. The database consists of 14 attributes of 301 individuals, out of which about 55 percent suffer from heart disease of one of the four kind mentioned above. However, the heart disease has not been categorized. The data set has simply distinguished the presence of heart diseases from its absence. The various features/attributes used and the values they take are:

1. Age : age of the person
2. Sex :(1 if male, 0 if female)
3. cp : chest pain type – Value 0: asymptomatic – Value 1: atypical angina – Value 2: non-anginal pain – Value 3: typical angina
4. trestbps : resting blood pressure (in mm Hg on admission to the hospital)
5. chol : serum cholesterol in mg/dl
6. fbs : fasting blood sugar $>$ 120 mg/dl (1 = true ; 0 =false)
7. restecg: resting electrocardiographic result – Value 0 : normal – Value 1 : having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV) – Value 2 : showing probable or definite left ventricular hypertrophy by Estes' criteria

8. thalach: maximum heart rate achieved
9. exang: exercise induced angina (1 = yes; 0 = no)
10. oldpeak = ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment – Value 0: downsloping – Value 1: flat – Value 2: upsloping
12. ca: number of major vessels (0-4) colored by fluoroscopy
13. thal: A blood disorder called thalassemia – Value 1 : fixed defect (no blood flow in some part of the heart) – Value 2: normal blood flow – Value 3: reversible defect (a blood flow is observed but it is not normal)
14. target: presence of heart disease (1= yes, 0= no)

I have randomly split the data into training and testing sets. The training set which comprises of 80% of the data set has been used for the purpose of building models, whereas rest 20% has been used for testing the accuracy of the models.

Models and their repective results

Linear Probability Model without interactions

```
## (Intercept)         age         sex         cp1         cp2         cp3
##       0.275       0.006      -0.146       0.209       0.277       0.262
##    trestbps        chol         fbs    restecg1    restecg2     thalach
##      -0.003       0.000      -0.008       0.052      -0.115       0.003
##       exang     oldpeak      slope1      slope2         ca1         ca2
##      -0.064      -0.011       0.070       0.217      -0.317      -0.379
##         ca3         ca4       thal2       thal3
##      -0.392       0.049       0.024      -0.164
```

```
##       target_hat
## target  0  1
##      0 23  8
##      1  5 24
```

After splitting the data into training and testing sets, I fitted a linear probability model using all of the 13 features without any interactions into the training data. Then, I used the model to make prediction on testing set. In order to test for out-of-sample accuracy, I created a confusion matrix with threshold of 0.5, that is, any prediction above the probability of 0.5 would be considered as presence of heart diseases and any prediction below the probability of 0.5 would be considered as no presence of heart disease. The out of sample accuracy from the confusion matrix is

```
## [1] 0.7833333
```

Linear Probability Model with interactions

```
##       target_hat
## target  0  1
##      0 24  7
##      1 12 17
```

```
## [1] 0.6833333
```

In order to check if I can improve the out of sample accuracy of the model, I considered a linear probability model with all possible two-way interactions. As a result, I obtained a model with variables with an intercept. In order to test for out-of-sample accuracy, I again created a confusion matrix with threshold of 0.5, however, the out-of-sample accuracy turned out to be, which is much lower than that of the linear probability model without any interactions. Thus, linear probability model without interactions clearly wins over the linear probability model with all possible two way interactions.

Logistic Regression Model without interactions

```
## (Intercept)         age         sex         cp1         cp2         cp3
##      -0.870       0.054      -1.685       1.244       2.889       2.434
##    trestbps        chol         fbs    restecg1    restecg2     thalach
##      -0.036      -0.006      -0.213       0.557      -0.665       0.035
##       exang     oldpeak      slope1      slope2         ca1         ca2
##      -0.479      -0.201       1.013       2.537      -2.910      -3.567
##         ca3         ca4       thal2       thal3
##      -3.540       1.490      -0.740      -2.210
```

```
##       target_hat
## target  0  1
##      0 22  9
##      1  5 24
```

```
## [1] 0.7666667
```

Secondly, I fit logistic regression model on training data using all the 13 variables without any interactions. The, the model was used to make prediction of testing set. Like in linear probability model, I created a confusion matrix with threshold of 0.5. The out of sample accuracy that I found is which is almost equal to that of linear probability model without interactions.

Logistic Regression Model with interactions

```
##       target_hat
## target  0  1
##      0 18 13
##      1 11 18
```

```
## [1] 0.6
```

Like in the case of linear probability model, I also fit logistic regression with all possible two-way interactions. The out of sample accuracy calculated using the confusion matrix turned out to be much lower than the logistic regression model without any interactions. Hence, the model without interactions is clearly the winner in this case as well.

Linear probability model vs Logistic regression model

As seen above out-of- sample accuracy calculated for the linear probability model and logistic regression model without interaction are almost the same. Therefore, I have taken a slightly more nuanced look at the performance of the classifier that simply calculation an overall accuracy. I have calculated:

- true positive rate (TPR) : among the people who have heart disease (y= 1), how may are correctly identified ($\hat{y}$ =1)
- false positive rate (FPR) : among the people who do not have heart disease(y= 0), how many are incorrectly identified as having heart disease ($\hat{y}$ =1)

The TPR and FPR for the linear probability model are

```
## [1] 0.8275862
```

```
## [1] 0.2580645
```

respectively.

The TPR and FPR for the logit model are

```
## [1] 0.8275862
```

```
## [1] 0.2903226
```

respectively.

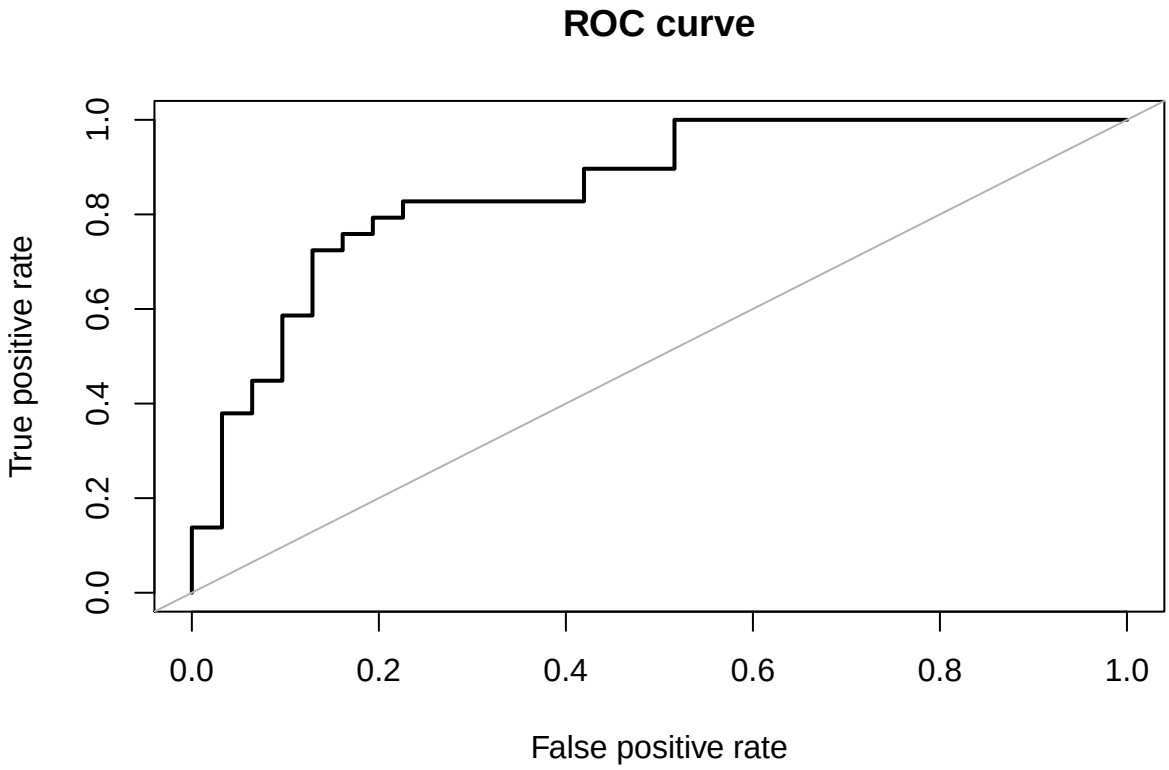
For calculation of error rates for our heart disease classifier, we used the threshold of 50%.

P(y = 1|x) > 0.5 - presence of heart disease

P(y = 1|x) < 0.5 - no presence of heart disease

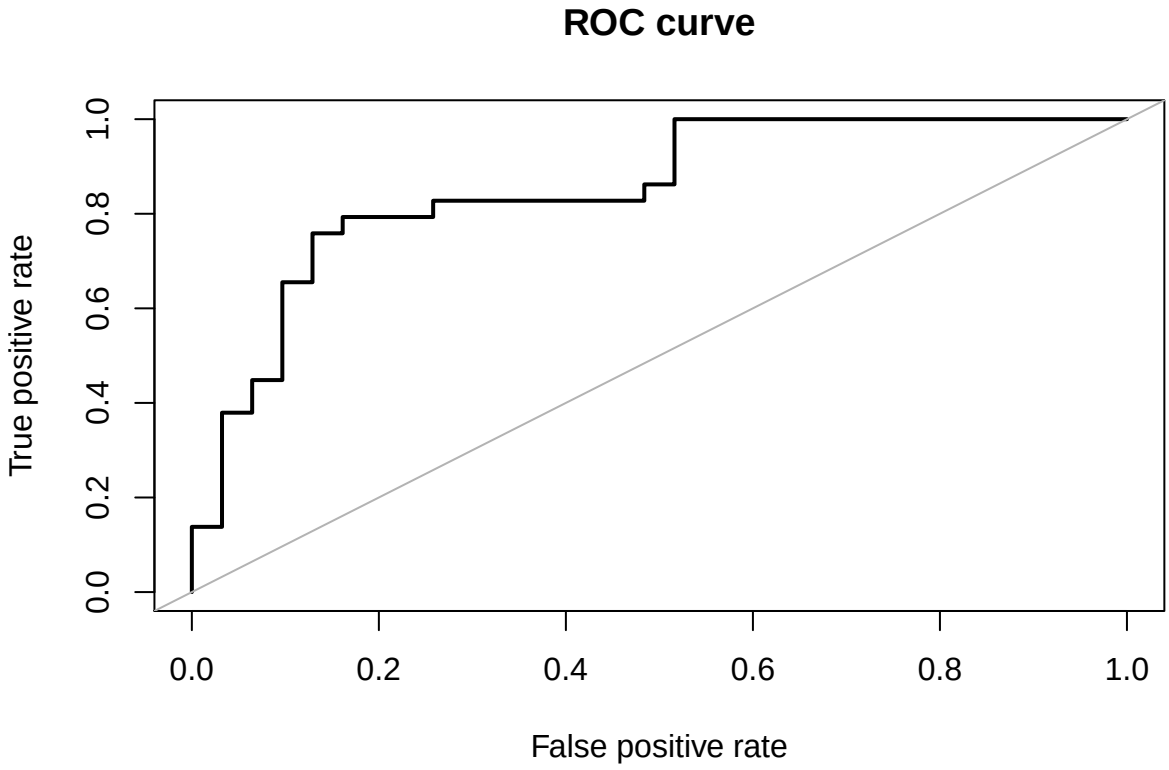
So, the question is what will happen to the performance and error rates if we vary the threshold. This question is addressed by Receiver Operation Characteristics (ROC) curve. ROC curve is a graph showing the performance of a binary classifier at all classification thresholds. At each threshold, TPR and FPR are computed. The, ROC curve plots TPR v FPR as functions of classification threshold. A ROC curve that is more "up and to the left" represents better performance, i.e. better detection of true positives at a fixed false positive rate. Moreover, we can also report the area under the ROC curve (AUC) as an overall measure of classifier performance. The close the AUC is to 'one', better the performance.

Figure 1: ROC curve for the Linear Probability Model




```
## Area under the curve (AUC): 0.854
```

Figure 2: ROC curve for the Logistic Regression Model




```
## Area under the curve (AUC): 0.852
```

To measure the relative performance across various classification threshold, I made ROC curves for both linear probability model and logistic regression model. I have reported AUC as the overall measure of performance. The AUC of the linear probability model is slightly higher that the AUC of logistic regression model.

Thus, we conclude that the better model out of the two is the linear probability model.

Tree based models

The major advantage of using decision trees is that they are intuitively easy to interpret and they can automatically detect non-linearities and interactions. However, the decision trees lack prediction accuracy. We can get a largely different decision tree with change in data. If we use split out data into multiple training sets, the structure of the tree might significantly differ for each training sets.

In order to overcome this limitation, we can aggregate across many decision trees. Aggregation techniques like bagging, random forest and boosting can help us improve the prediction accuracy significantly.

Bagging

The first tree related model which I fit is bagging or bootstrap aggregating model. As the name suggests, bagging involves taking multiple bootstrap sample from the training data and fitting a classification tree for

each bootstrap sample. Finally, prediction is done by averaging predictions from all the trees obtained from each bootstrap samples.

```
##       target_hat
## target  0  1
##      0 23  8
##      1  8 21
```

For a threshold of 0.5, the out-of-sample accuracy for the bagging model is

```
## [1] 0.7333333
```

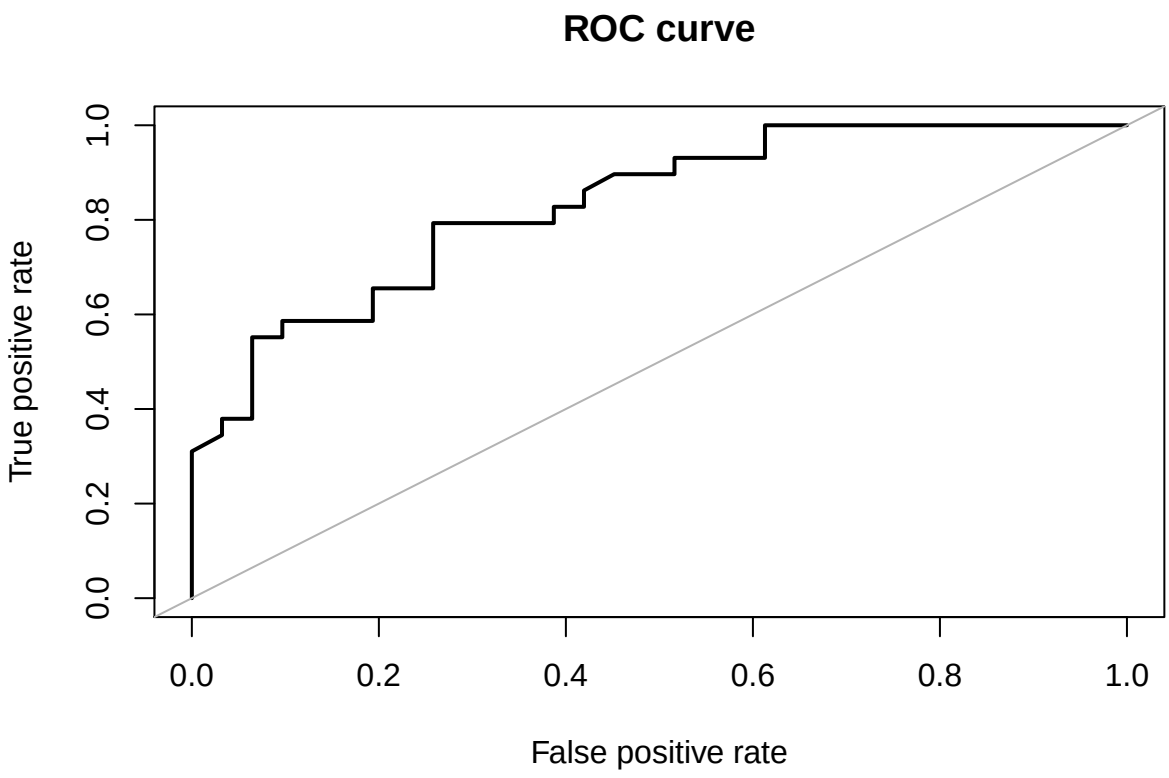
The TPR and FPR are

```
## [1] 0.7241379
```

```
## [1] 0.2580645
```

respectively.

Figure 3: ROC curve for the Tree Bagging Model




```
## Area under the curve (AUC): 0.832
```

Random Forest Model

Random forest model is very similar to bagging except for it adds more randomness. We still build a number of trees on bootstrap samples, but instead of all the feature variables m , a random subset of m<p is chosen as split candidates from the full set of m variables each time a split in a tree is considered.

```
##       target_hat
## target  0  1
##      0 23  8
##      1  7 22
```

For a threshold of 0.5, the out-of-sample accuracy for the random forest model is

```
## [1] 0.75
```

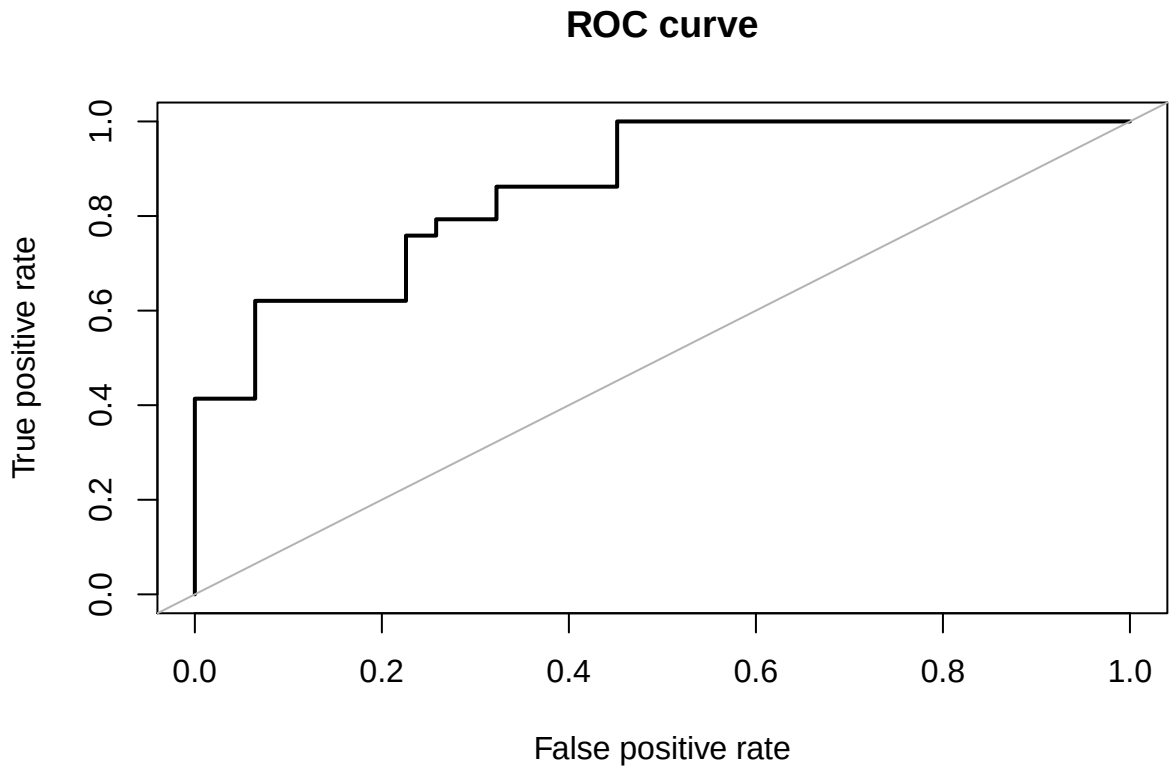
The TPR and FPR are

```
## [1] 0.7586207
```

```
## [1] 0.2580645
```

respectively.

Figure 4: ROC curve for the Random Forest Model




```
## Area under the curve (AUC): 0.862
```

Boosting model

The last model that I use is Boosting model. Boosting is also an ensemble method like random forest in which overall fit is produced from many trees. However, it is quite different. Trees are grown sequentially by using the information from previously grown trees, In boosting, we fit the data with a single tree, then crush the tree so that it does not fit very well. Then, by using the part of the target variable not captured by the crushed tree, we fit a new tree. Our new fit is sum of the two trees. This process is conducted repeatedly and our final fit is sum of the many tree thus created.

```
##       target_hat
## target  0  1
##      0 23  8
##      1  8 21
```

For a threshold of 0.5, the out-of-sample accuracy for the random forest model is

```
## [1] 0.7333333
```

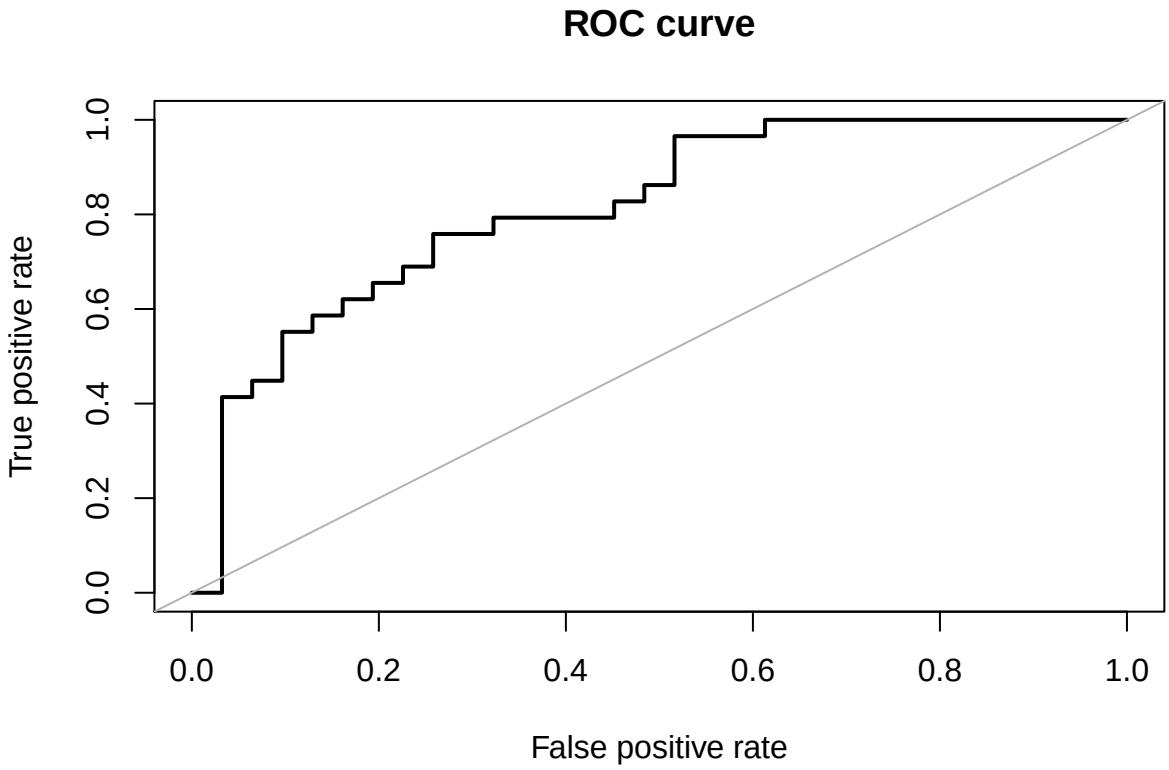
The TPR and FPR are

```
## [1] 0.7241379
```

```
## [1] 0.2580645
```

respectively.

Figure 5: ROC curve for the Boosting Model




```
## Area under the curve (AUC): 0.814
```

Conclusion

From the results based on the analyses made above, we can conclude that random forest model made the most accurate prediction. However, I have to acknowledge the fact that the other models were not very far behind in terms of accuracy. Looking at the accuracy level of linear probability model, there is a possibility that for a different training and testing split it could be the winner. Usually, tree based models perform better than linear probability model or logistic regression model, but owing to low number of observations, linear probability model has performed better that bagging and gradient boosting model, and the random forest model has performed only marginally better than linear probability model.