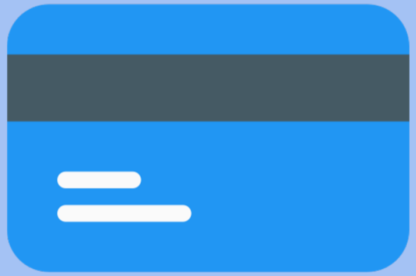


Credit Card Approval Prediction

By: Bean, Dennis, Leora, Vincent, Zoey

Lack of Transparency

How do banks decide who to approve or reject?

Banks have complex algorithms to determine whether an individual will be approved for a credit card or not

But what do these algorithms use to judge a person's risk level?

Credit Card Approval Dataset

What dataset are we working with?

6

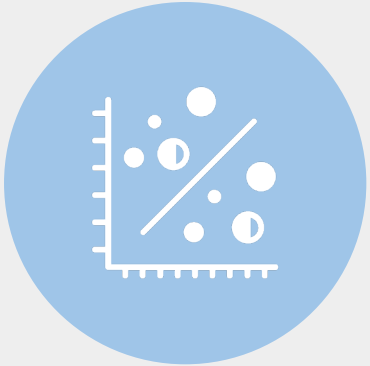
Categorical

15

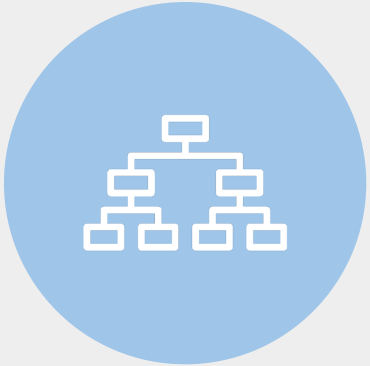
Numerical

Which models were used?

The different classification models used for approval prediction



Logistic Regression

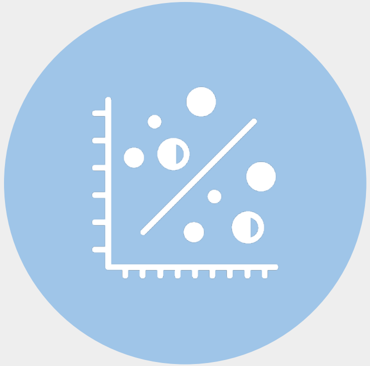


Decision Tree

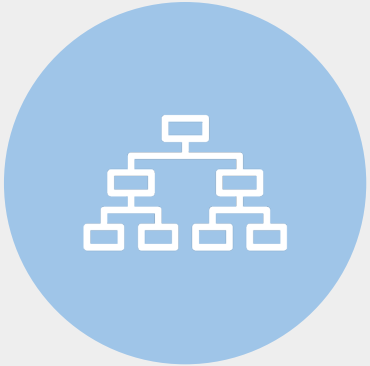
Random Forest & XGBoost

Which models were used?

The different classification models used for approval prediction



Logistic Regression



Decision Tree

Random Forest &
XGBoost

Logistic Regression

We chose 6 independent variables, which are:
Gender, Income, Education, Age, Years of Work Experience, and Good Debt.
The Education Type has a negative impact on the application status.

```
## Education_TypeHigher education                         -9.663e+00
## Education_TypeIncomplete higher                         -9.778e+00
## Education_TypeLower secondary                           -1.064e+01
## Education_TypeSecondary / secondary special             -9.505e+00
```

However, if we run all the variables together, Education Type becomes a helpful element when applying for a credit card.

```
## Education_TypeHigher education                          4.800e+04
## Education_TypeIncomplete higher                         4.809e+04
## Education_TypeLower secondary                           4.928e+04
## Education_TypeSecondary / secondary special             4.800e+04
```

Logistic Regression

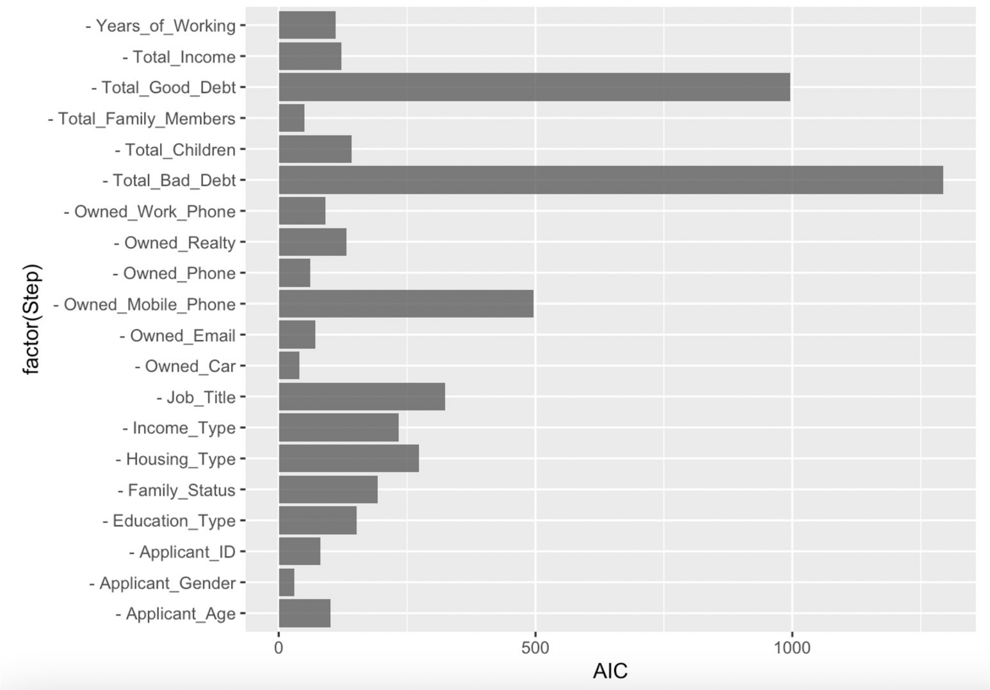
```
summary(lm_bwd_all)
```

```
##
## Call:
## glm(formula = Status ~ Total_Bad_Debt + Total_Good_Debt, family = binomial(link = "logit"),
##     data = total_data)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -8.222e-04   2.000e-08   2.000e-08   2.000e-08   9.526e-04
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -14.83     201.51  -0.074    0.941
## Total_Bad_Debt      -31.58     239.32  -0.132    0.895
## Total_Good_Debt      31.51     240.90   0.131    0.896
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.5327e+03  on 25127  degrees of freedom
## Residual deviance: 5.4921e-05  on 25125  degrees of freedom
## AIC: 6.0001
##
## Number of Fisher Scoring iterations: 25
```

For logistic regression, we know that good and bad debt records are the most important.

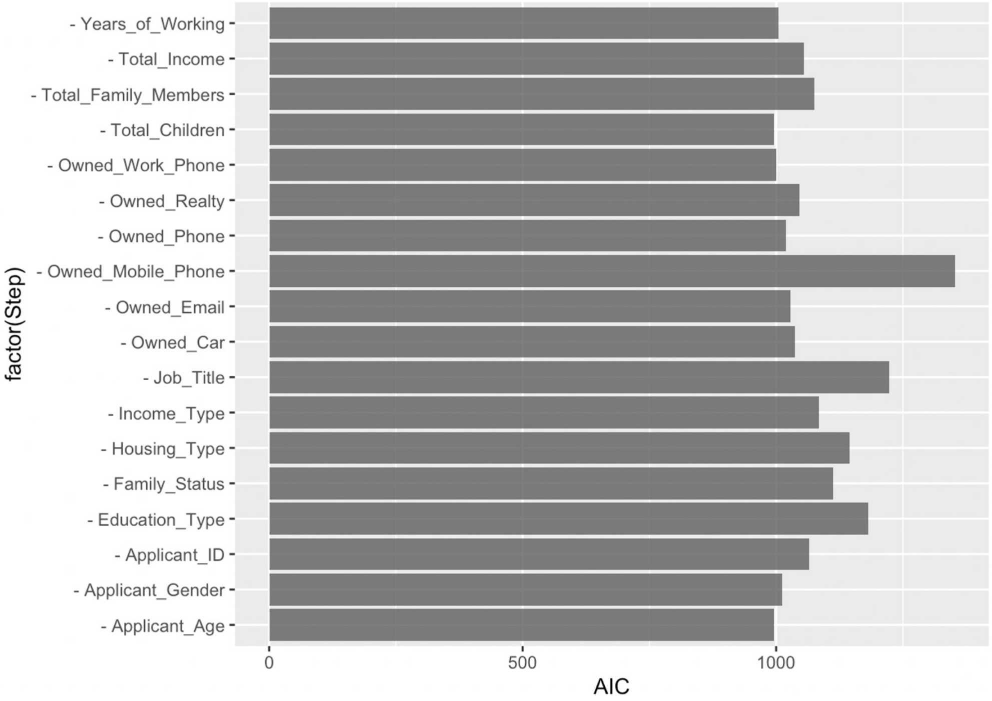
We can use backward selection to get that answer easily.

Logistic Regression



Using visualization is also a good way to notice the huge difference between important ones and not important ones.

Logistic Regression

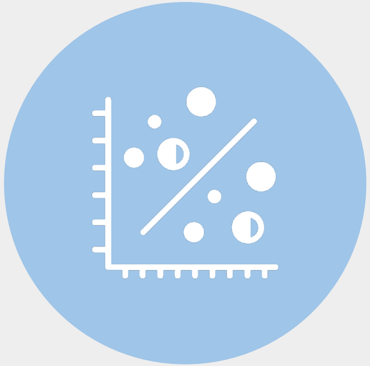


Then remove “Total_Bad_Debt” and ”Total_Good_Debt”.

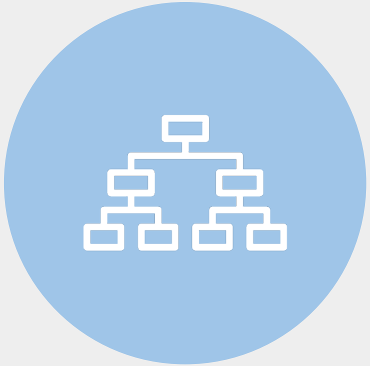
In summary, “Owned_Mobile_Phone”, “Job_Title” and “Education_Type” are now the top 3.

Which models were used?

The different classification models used for approval prediction

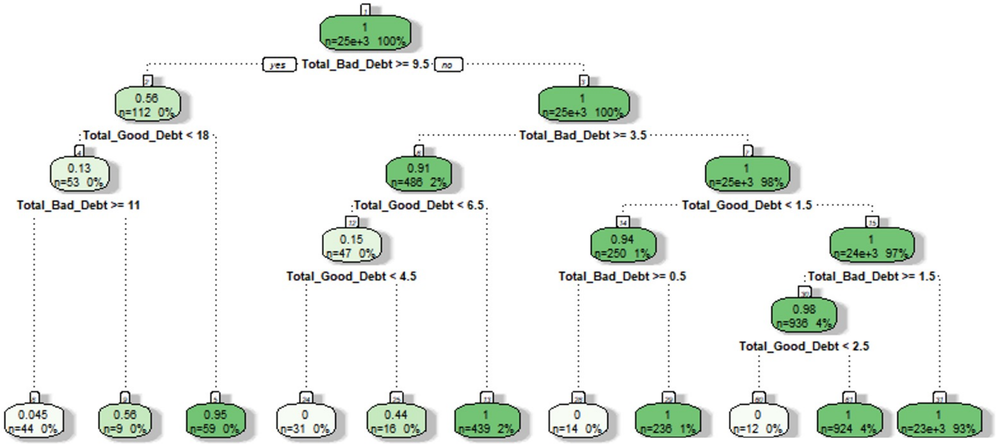


Logistic Regression



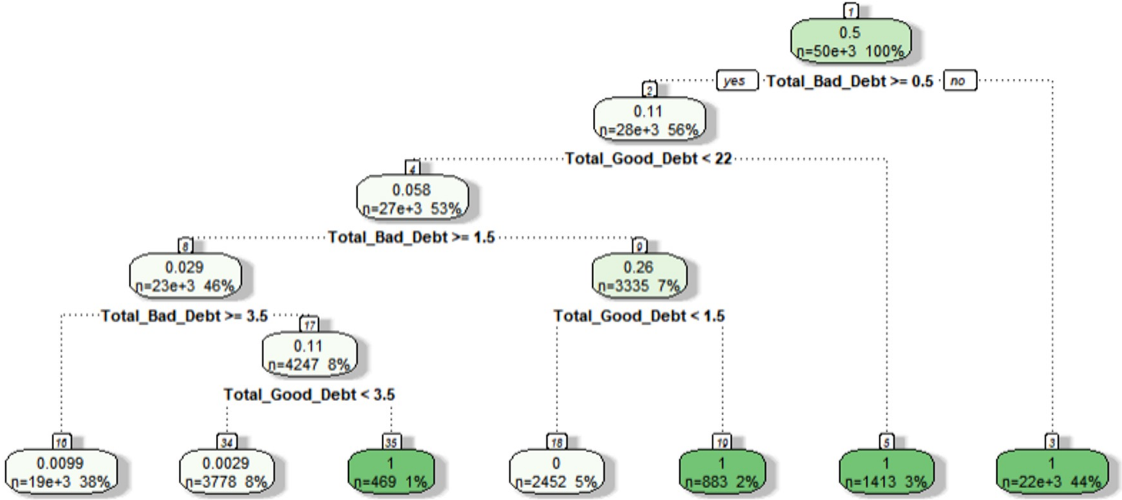
Decision Tree

Random Forest &
XGBoost

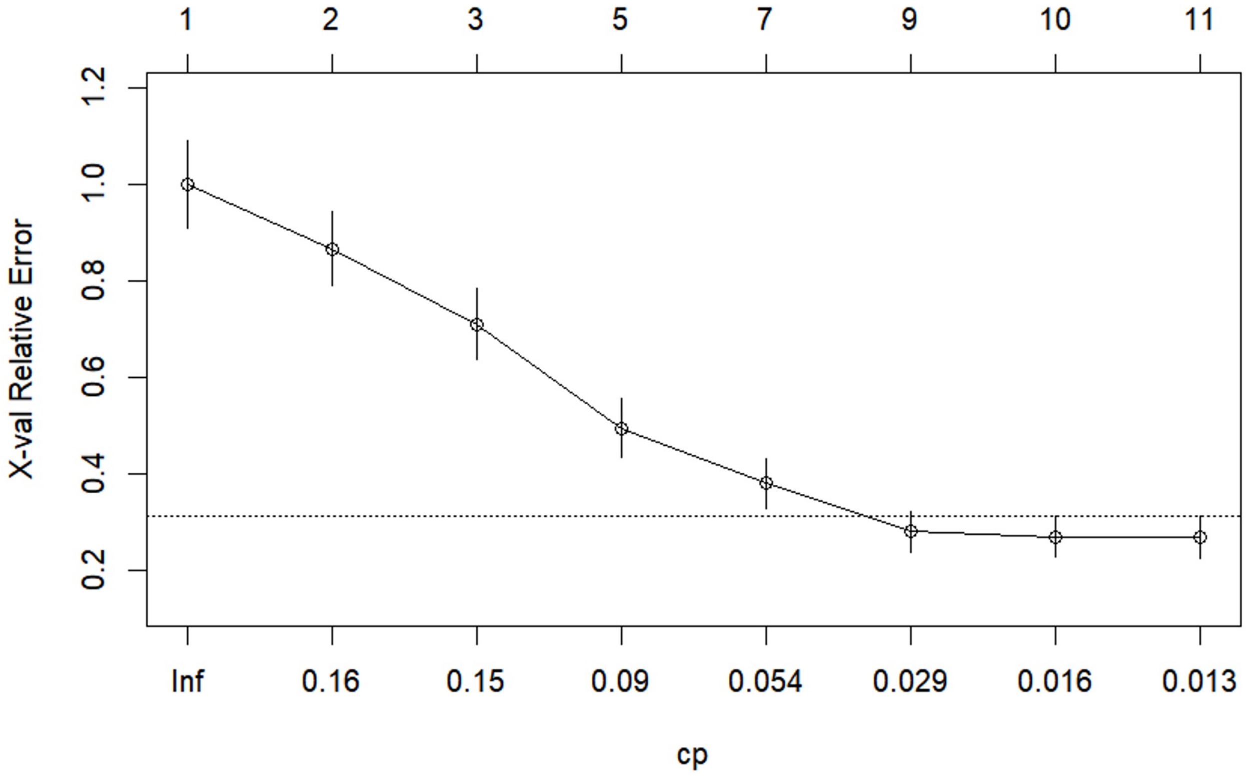


Initial decision tree (all hyperparameters are default and no processing of the data)

The final decision tree
(much more concise)



Find the best cp value



Set the threshold value to 0.5

```
Confusion Matrix and Statistics

preds_char    0    1
         0   21    0
         1    3 5001

               Accuracy : 0.9994
                 95% CI : (0.9983, 0.9999)
    No Information Rate : 0.9952
    P-Value [Acc > NIR] : 9.451e-08

                  Kappa : 0.933

 Mcnemar's Test P-Value : 0.2482

            Sensitivity : 1.0000
            Specificity : 0.8750
         Pos Pred Value : 0.9994
         Neg Pred Value : 1.0000
             Prevalence : 0.9952
         Detection Rate : 0.9952
   Detection Prevalence : 0.9958
      Balanced Accuracy : 0.9375

       'Positive' Class : 1
```

Set the threshold value to 0.3

```
Confusion Matrix and Statistics

preds_char    0    1
         0    2    0
         1   22 5001

               Accuracy : 0.9956
                 95% CI : (0.9934, 0.9973)
    No Information Rate : 0.9952
    P-Value [Acc > NIR] : 0.3913

                  Kappa : 0.1532

 Mcnemar's Test P-Value : 7.562e-06

            Sensitivity : 1.00000
            Specificity : 0.08333
         Pos Pred Value : 0.99562
         Neg Pred Value : 1.00000
             Prevalence : 0.99522
         Detection Rate : 0.99522
   Detection Prevalence : 0.99960
      Balanced Accuracy : 0.54167

       'Positive' Class : 1
```

The number of ‘0’ and ‘1’ contained in status in the raw data is very imbalanced

```
  0     1
121 25007
```

Bootstrap resampling

```
# Split data into approved and rejected classes
rejected <- total_data[which(total_data$Status == 0),] # Select minority samples
approved <- total_data[which(total_data$Status == 1),] # Select majority samples
nrow(rejected) # Rows in rejected
nrow(approved)
set.seed(123456) # Set seed for sampling
rejected_boot <- rejected[sample(1:nrow(rejected), size = nrow(approved), replace =TRUE),]
nrow(rejected_boot) # Check rows of bootstrap sample
```

```
[1] 121
[1] 25007
[1] 25007
```

```
Confusion Matrix and Statistics


preds_char    0    1
         0 5001   32
         1    0 4969

               Accuracy : 0.9968
                 95% CI : (0.9955, 0.9978)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9936

 Mcnemar's Test P-Value : 4.251e-08

            Sensitivity : 0.9936
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9936
             Prevalence : 0.5000
         Detection Rate : 0.4968
   Detection Prevalence : 0.4968
      Balanced Accuracy : 0.9968

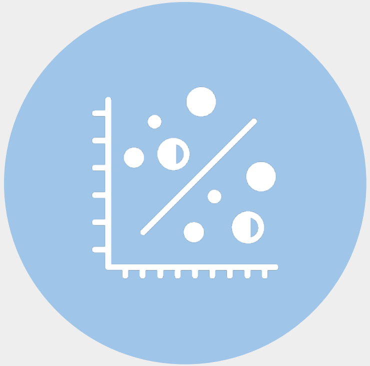
       'Positive' Class : 1
```

Decision Tree

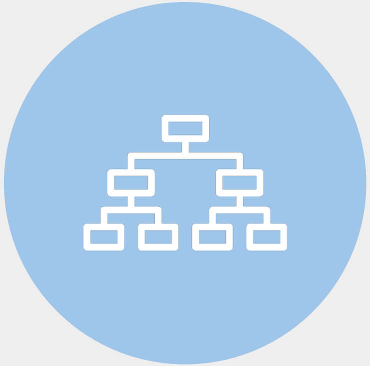
Final Evaluation

Which models were used?

The different classification models used for approval prediction



Logistic Regression



Decision Tree

Random Forest & XGBoost

Random Forest

Random Forest Without Good or Bad Debt

```
## Confusion Matrix and Statistics
##
##
## rf_preds    0    1
##        0    3   10
##        1   26 7500
##
##                Accuracy : 0.9952
##                  95% CI : (0.9934, 0.9967)
##     No Information Rate : 0.9962
##     P-Value [Acc > NIR] : 0.91478
##
##                   Kappa : 0.1408
##
##  Mcnemar's Test P-Value : 0.01242
##
##             Sensitivity : 0.9987
##             Specificity : 0.1034
##          Pos Pred Value : 0.9965
##          Neg Pred Value : 0.2308
##              Prevalence : 0.9962
##          Detection Rate : 0.9948
##    Detection Prevalence : 0.9983
##       Balanced Accuracy : 0.5511
##
##        'Positive' Class : 1
##
```

Random Forest Without Good Debt

```
## Confusion Matrix and Statistics
##
##
## rf_preds    0    1
##        0   12    4
##        1   17 7506
##
##                Accuracy : 0.9972
##                  95% CI : (0.9957, 0.9983)
##     No Information Rate : 0.9962
##     P-Value [Acc > NIR] : 0.076298
##
##                   Kappa : 0.5321
##
##  Mcnemar's Test P-Value : 0.008829
##
##             Sensitivity : 0.9995
##             Specificity : 0.4138
##          Pos Pred Value : 0.9977
##          Neg Pred Value : 0.7500
##              Prevalence : 0.9962
##          Detection Rate : 0.9956
##    Detection Prevalence : 0.9979
##       Balanced Accuracy : 0.7066
##
##        'Positive' Class : 1
##
```

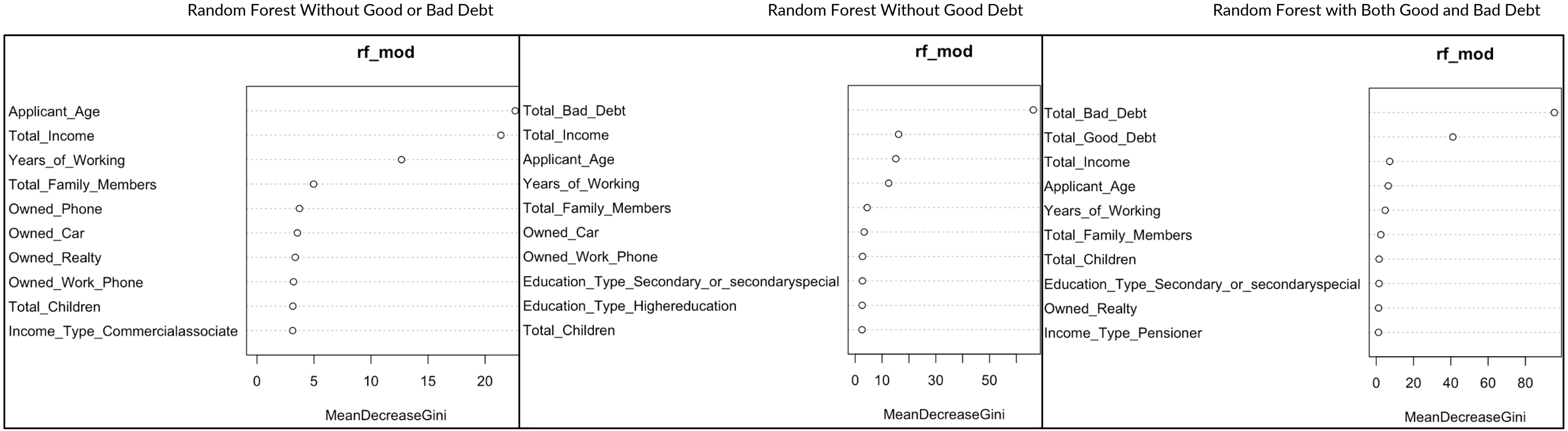
Random Forest with Both Good and Bad Debt

```
## Confusion Matrix and Statistics
##
##
## rf_preds    0    1
##        0   22    0
##        1    7 7510
##
##                Accuracy : 0.9991
##                  95% CI : (0.9981, 0.9996)
##     No Information Rate : 0.9962
##     P-Value [Acc > NIR] : 1.097e-06
##
##                   Kappa : 0.8623
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 1.0000
##             Specificity : 0.7586
##          Pos Pred Value : 0.9991
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9962
##          Detection Rate : 0.9962
##    Detection Prevalence : 0.9971
##       Balanced Accuracy : 0.8793
##
##        'Positive' Class : 1
##
```

Balanced Accuracy		
0.5511	0.7066	0.8793 Best

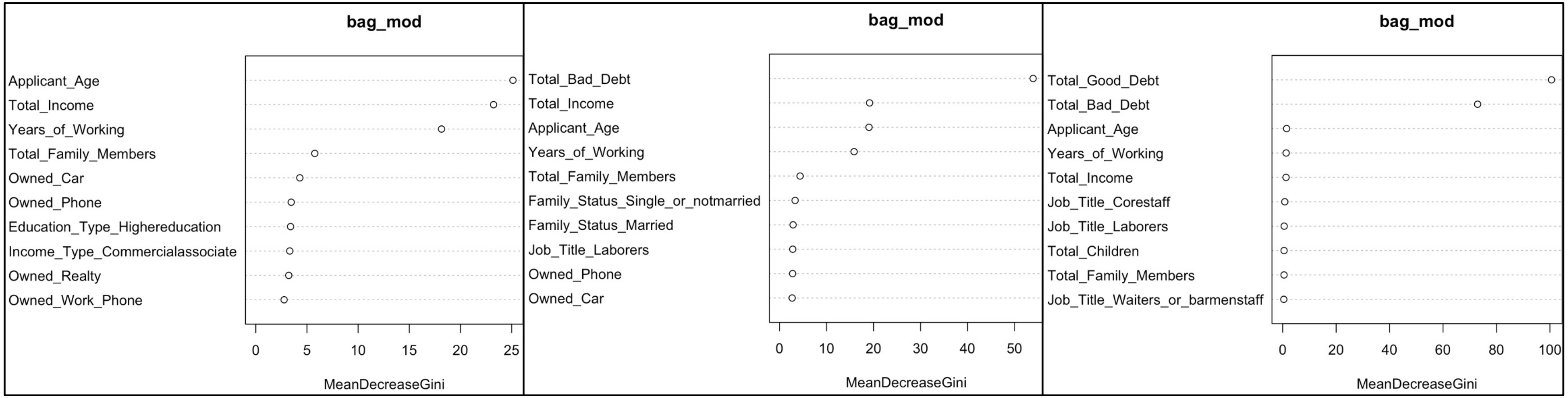
Balanced Accuracy goes down as we take away significant variables

Random Forest Variable Importance



Total_Bad_Debt and Total_Good_Debt are by far the most important features of the model with Applicant_Age, Total_Income, and Years_of_Working also being somewhat significant.

Checking with Bagging



No visible difference which shows that our random forest model is quite stable and reliable

XGBoost

Imbalanced Data Problem

We have 121 rejection cases and 25007 approval cases.

Thus, we create a weight which we can use to scale the positive class weight so that we have equal representation in the dataset in terms of initial weights.

```
```{r zero weight}
zero_weight <- 121/25007
```
```

We can then feed this vector to xgboost using the weight parameter.

```
```{r }
set.seed(886)
xg_mod_bal <- xgboost(data = dtrain,
              eta = 0.05,
              max.depth =  7,
              min_child_weight = 10,
              gamma = 0,
              subsample =  0.9,
              colsample_bytree = 0.9,

              nrounds = 50,

              verbose = 1,
              nthread = 1,
              print_every_n = 20,

              scale_pos_weight = zero_weight,

              objective = "binary:logistic",
              eval_metric = "auc",
              eval_metric = "error")
```
```

XGBoost

Model Accuracy

Imbalanced Data

```
Confusion Matrix and Statistics

boost_pred_class    0    1
               0    8   16
               1   21 7494

               Accuracy : 0.9951
                 95% CI : (0.9932, 0.9965)
    No Information Rate : 0.9962
    P-Value [Acc > NIR] : 0.9385

                  Kappa : 0.2994

 Mcnemar's Test P-Value : 0.5108

            Sensitivity : 0.9979
            Specificity : 0.2759
         Pos Pred Value : 0.9972
         Neg Pred Value : 0.3333
             Prevalence : 0.9962
         Detection Rate : 0.9940
   Detection Prevalence : 0.9968
      Balanced Accuracy : 0.6369

       'Positive' Class : 1
```

Balanced Data

```
Confusion Matrix and Statistics

     0    1
0    0    0
1   29 7510

               Accuracy : 0.9962
                 95% CI : (0.9945, 0.9974)
    No Information Rate : 0.9962
    P-Value [Acc > NIR] : 0.5492

                  Kappa : 0

 Mcnemar's Test P-Value : 1.999e-07

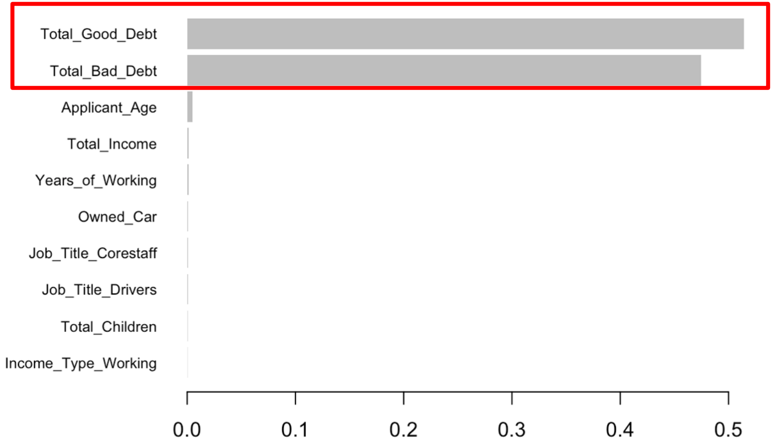
            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.9962
         Neg Pred Value :    NaN
             Prevalence : 0.9962
         Detection Rate : 0.9962
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

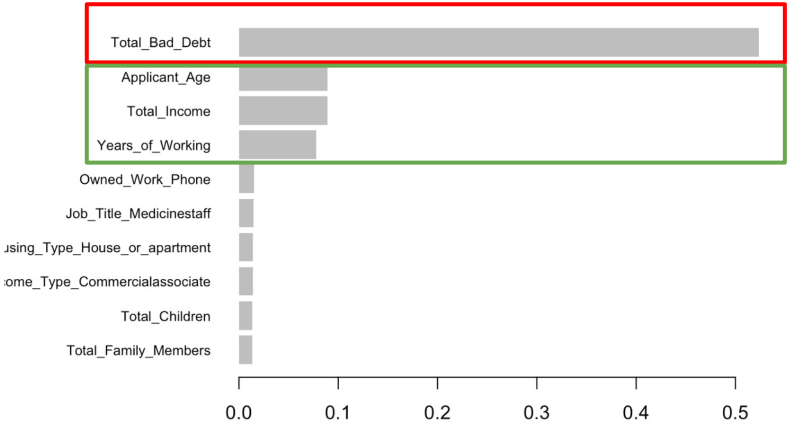
       'Positive' Class : 1
```

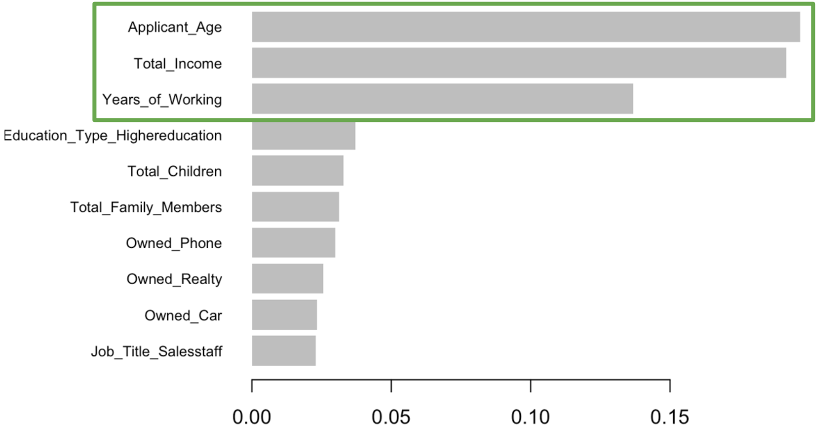
XGBoost

Variable Importance

(Balanced Data)

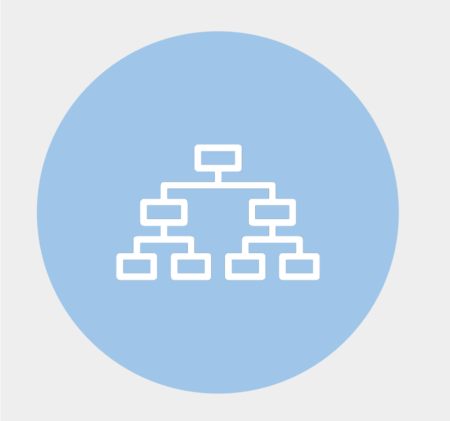






Which model is the ideal one ?

Logistic Regression



Decision Tree

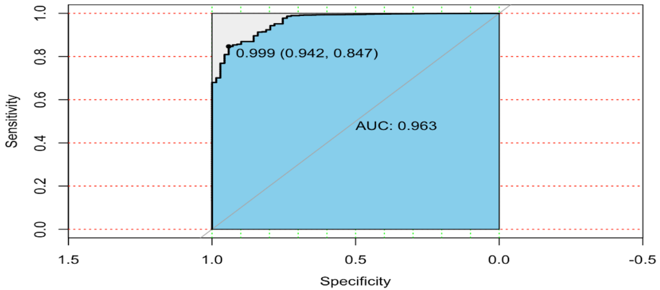
Random Forest & XGBoost

Important variables: Total_Good_Debt, Total_Bad_Debt, Applicant_Age, Total_Income, Years_of_working

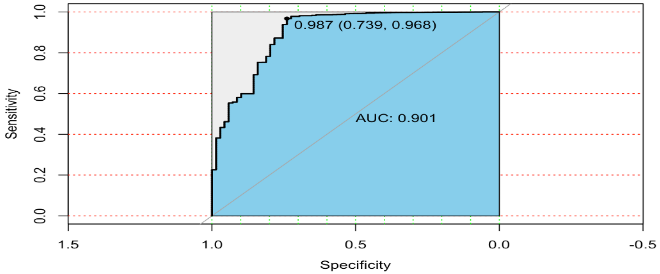
Naive Bayes Model

(-) Accuracy

With Good & Bad Debts



Without Good Debts



Without both Good & Bad Debts