Executive Summary

This project aims to identify model that can be used to classify people, with certain socio-economic conditions, based on their capability to pay loan back. Data for the project is posted in Kaggle.com as a competition. Classification models were created with 14 variables and 20,000 rows. Out of SVM, logistic regression, and KNN models, SVM was the best model due to higher value of ROC, sensitivity, and specificity. Decision tree models were also performed, and C5.0 is the best decision tree in terms of confusion matrix and ROC value.

Introduction

Several people have hard time to receive loan due to lack of or non-existent credit history, so are end up with risky loans. Home credit wants to attract such population and is dedicated to provide the positive and safe borrowing experience. To make sure that the business is safe for home credit, home credit wants to find out what kind of customers have higher chance of loan repayment. So, home credit is opened one competition, where peoples capability to repay loan is estimated by analyzing the telco and transactional information. In the competition, competitors are expected to provide the possible solution using several statistical and machine learning methods. This project is using several classification machine learning methods to find the best model which will identify the target variable with higher accuracy.

Methods

1) Data Loading, cleaning, and preparation

This project will only use the train data posted in Kaggle.com, which contains 307511 rows and 122 columns. Due to capacity of working laptop, only 20000 rows are selected. Out of 122 columns, only 18 variables related with different socio-economic factors were selected based on the null value ratio and zero variances, which were available from the discussion of the competition. Variables with null proportion higher than 50% were not considered for the project.

1.1) Cleaning:

Data were thoroughly checked, and information posted in Kaggle.com were also checked to get the data preparation and cleaning ideas. Data preparation for the project is conducted in both csv file and R. Days variable included values of 365243, which were recognized as infinity values and are replaced with NA values. Likewise the XNA/XAP values were also changed to NA values. The negative date variables were changed to abs value in excel using (=abs()) function to perform skewness analysis.

1.2) Replacing null values:

The empty cells and other null (NA) values were replaced by the mean value of each variable, in R, by using the na.aggregate function.

1.3) Skewness and Transformation

Skewness test of the data shows all of the variables are +vely or -vely skewed (Fig. 1). Transformation of the data is necessary to remove distributional skewness. The transformation can be done with log, square root, or inverse functions. Though the transformation may not

entirely remove the symmetric distribution, the data are better behaved than they were in their previous state (Kuhn and Johnson, 2013). Appropriate transformation can be identified by using statistical analysis, one is Box and Cox method which uses lambda index (Box and Cox, 1964) . The main concept of the Box and Cox method is using maximum likelihood estimation to determine the transformation parameter (Kuhn and Johnson, 2013). Lambda value greater than zero requires the transformation process. Box and Cox procedure identifies the variables needed to be corrected or transformed (Fig. 2).

```
> skewValues
                      TARGET                  CODE_GENDER1                  FLAG_OWN_CAR1
                  2.97931284                    0.95785636                     0.98866771
          FLAG_OWN_REALTY_1                 CNT_CHILDREN                AMT_INCOME_TOTAL
                 -0.83518866                    1.88629988                    -0.13997049
                  AMT_CREDIT                  AMT_ANNUITY                AMT_GOODS_PRICE
                  1.12687557                    1.07632363                     1.24708896
             DAYS_ID_PUBLISH              CNT_FAM_MEMBERS          REGION_RATING_CLIENT
                 -0.35966160                    0.95017334                     0.40350787
REGION_RATING_CLIENT_W_CITY      REG_CITY_NOT_LIVE_CITY                 EXT_SOURCE_1
                  0.38649528                    3.11993085                    -0.01415223
                EXT_SOURCE_2                 EXT_SOURCE_3                 YEARS_BEGIN
                 -0.67715582                   -0.51614061                   -21.64755244
```

Figure 1. Skewness results.

Column	lambda
AMT_INCOME_TOTAL	1.1
AMT_CREDIT	0.2
AMT_ANNUITY	0.3
AMT_GOODS_PRICE	0.1
CNT_FAM_MEMBERS	0.2
REGION_RATING_CLIENT	0.7
REGION_RATING_CLIENT_W_CITY	0.8
EXT_SOURCE_1	1.1
EXT_SOURCE_2	1.3
EXT_SOURCE_3	1.3

Figure 2. Box-Cox transformation results.

10 columns are identified for transformation from the Box Cox (Fig. 2). Following formula in R is used to transform data or to reduce the skewness:

```
preProcValues <- preProcess(data1, method = "BoxCox")
preProcValues
dt1_tran <- predict(preProcValues, data1)
```

The plot of the datasets before and after transformation shows the reduction in the effect on the skewness (Fig. 3 and 4).

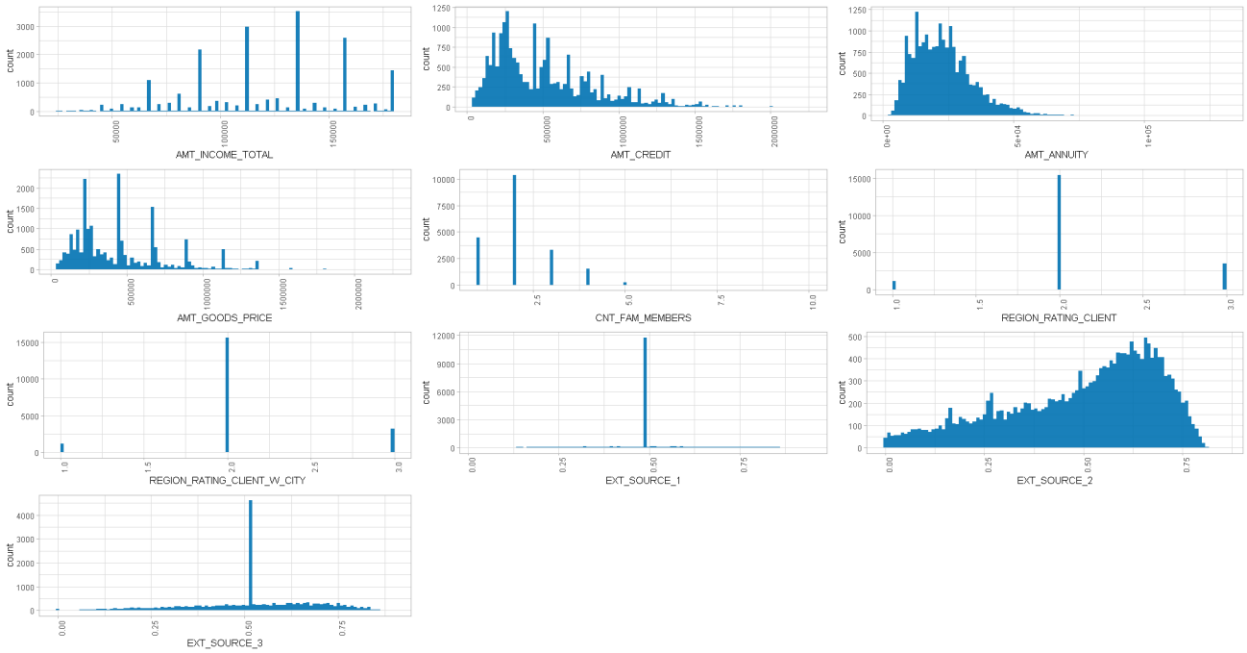


Figure 3. Dataset before transformation.

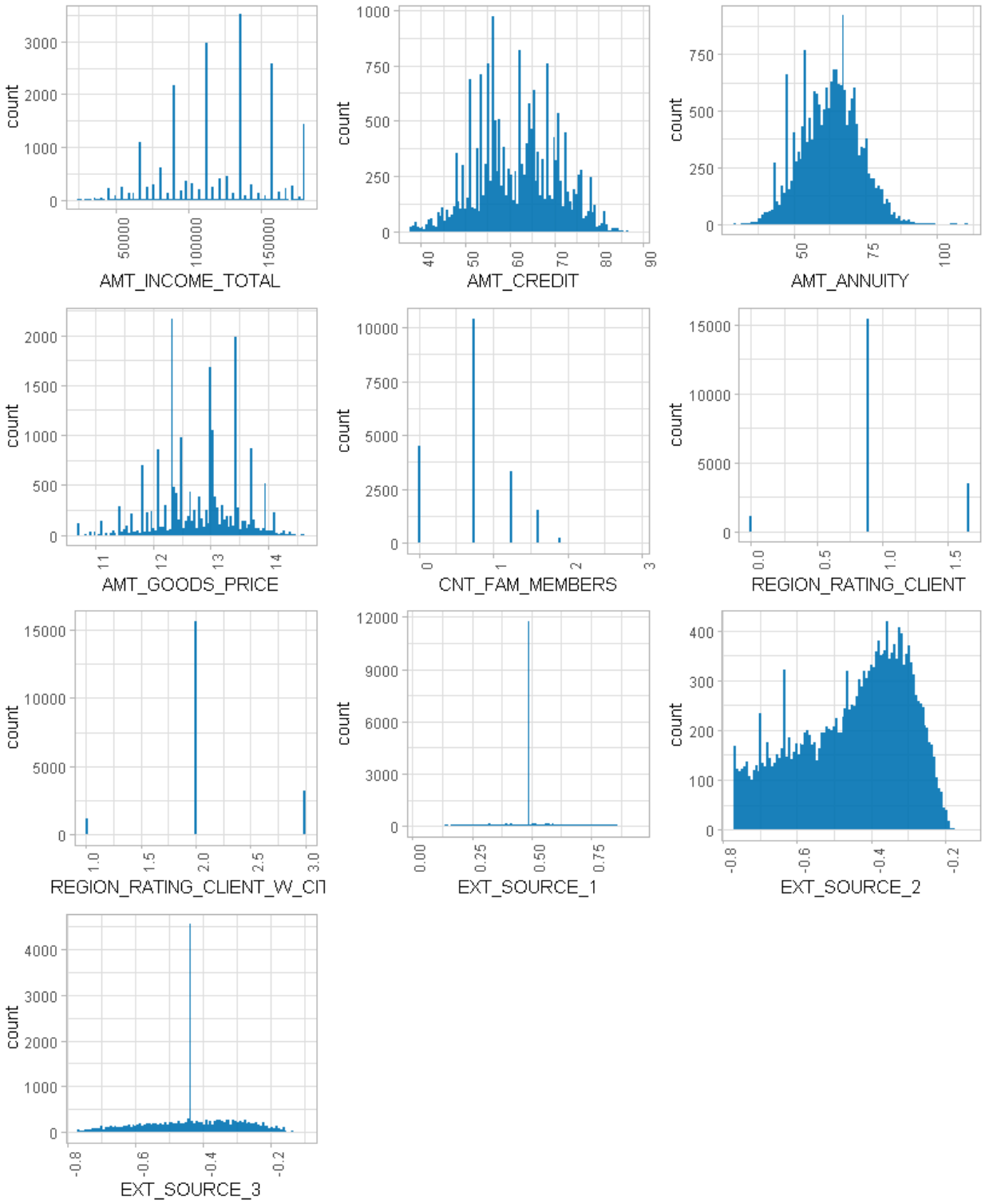
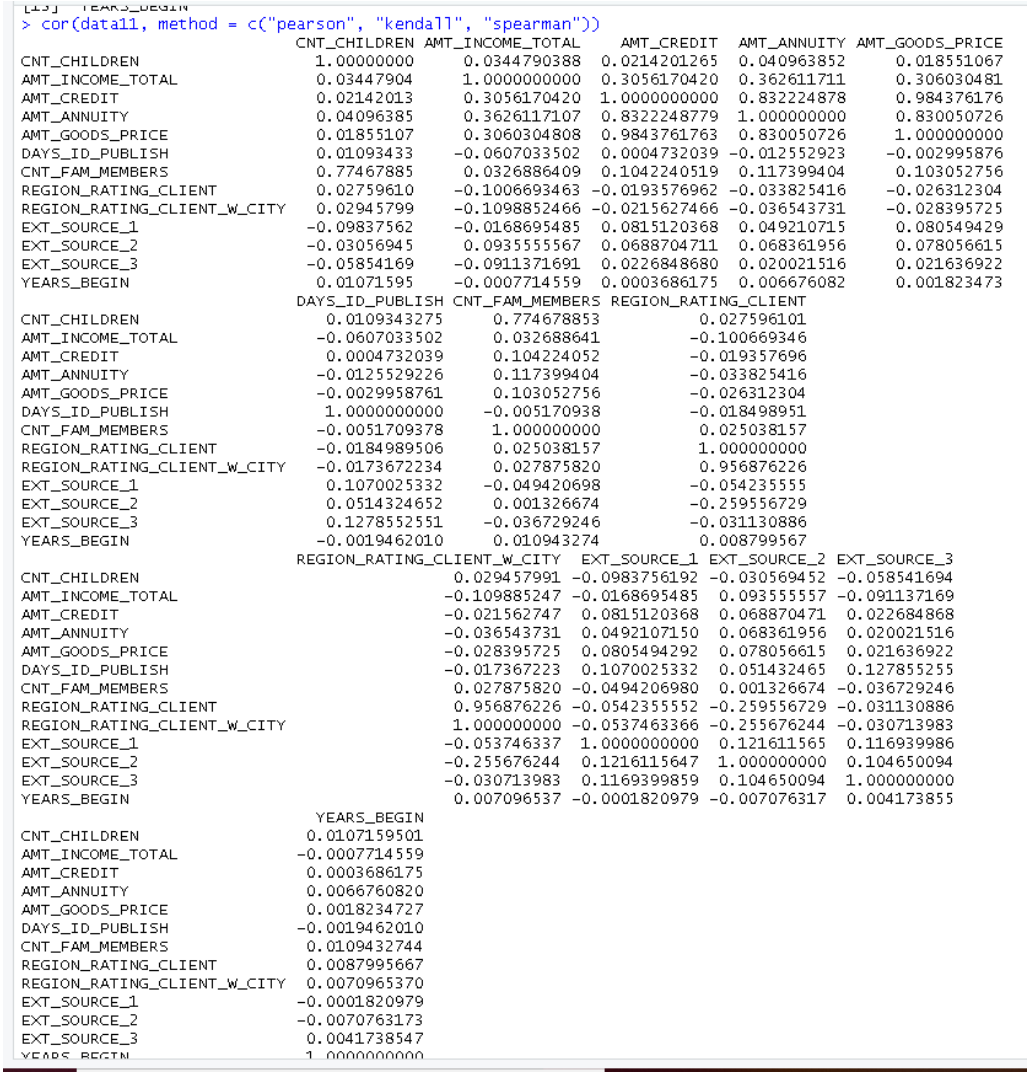


Figure 4. Dataset after transformation.

1.4) Correlation Coefficient:

Multicollinearity in the dataset is checked with the analysis of the correlation coefficients between non-characteristic dependent variables. The dataset consists of only 13 numeric or non-characteristic variables. The resultant table of the correlation coefficient (Fig. 5) is hard to read, so heatmaps were created to better visualize the correlation coefficients. The final heatmap (Fig. 6) is created in triangular form for effective visualization of the highly correlated dependent variables. The sequential transformation of the heatmap of the correlation coefficients into triangular form is shown in Appendix A. The heatmap (Fig. 6) shows higher correlation in three groups: high positive correlation of REGION_RATING_CLIENT with REGION_RATING_CLIENT_W_CITY; CNT_CHILDREN with CNT_FAM_MEMBERS; and AMT_ANNUITY with AMT_CREDIT and AMT_GOODS_PRICE. Only one from each correlated group is selected. AMT_CREDIT, REGION_RATING_CLIENT_W_CITY, and CNT_FAM_MEMBERS are selected due to less outliers identified from box plot. So, the final dataset included only 14 variables including the dependent variable.



```
[13] "YEARS_BEGIN"
> cor(data11, method = c("pearson", "kendall", "spearman"))
                        CNT_CHILDREN AMT_INCOME_TOTAL    AMT_CREDIT  AMT_ANNUITY AMT_GOODS_PRICE
CNT_CHILDREN              1.00000000     0.0344790388  0.0214201265  0.040963852     0.018551067
AMT_INCOME_TOTAL          0.03447904     1.0000000000  0.3056170420  0.362611711     0.306030481
AMT_CREDIT                0.02142013     0.3056170420  1.0000000000  0.832224878     0.984376176
AMT_ANNUITY               0.04096385     0.3626117107  0.8322248779  1.000000000     0.830050726
AMT_GOODS_PRICE           0.01855107     0.3060304808  0.9843761763  0.830050726     1.000000000
DAYS_ID_PUBLISH           0.01093433    -0.0607033502  0.0004732039 -0.012552923    -0.002995876
CNT_FAM_MEMBERS           0.77467885     0.0326886409  0.1042240519  0.117399404     0.103052756
REGION_RATING_CLIENT      0.02759610    -0.1006693463 -0.0193576962 -0.033825416    -0.026312304
REGION_RATING_CLIENT_W_CITY 0.02945799  -0.1098852466 -0.0215627466 -0.036543731    -0.028395725
EXT_SOURCE_1             -0.09837562    -0.0168695485  0.0815120368  0.049210715     0.080549429
EXT_SOURCE_2             -0.03056945     0.0935555567  0.0688704711  0.068361956     0.078056615
EXT_SOURCE_3             -0.05854169    -0.0911371691  0.0226848680  0.020021516     0.021636922
YEARS_BEGIN               0.01071595    -0.0007714559  0.0003686175  0.006676082     0.001823473
                        DAYS_ID_PUBLISH CNT_FAM_MEMBERS REGION_RATING_CLIENT
CNT_CHILDREN              0.0109343275     0.774678853          0.027596101
AMT_INCOME_TOTAL         -0.0607033502     0.032688641         -0.100669346
AMT_CREDIT                0.0004732039     0.104224052         -0.019357696
AMT_ANNUITY              -0.0125529226     0.117399404         -0.033825416
AMT_GOODS_PRICE          -0.0029958761     0.103052756         -0.026312304
DAYS_ID_PUBLISH           1.0000000000    -0.005170938         -0.018498951
CNT_FAM_MEMBERS          -0.0051709378     1.000000000          0.025038157
REGION_RATING_CLIENT     -0.0184989506     0.025038157          1.000000000
REGION_RATING_CLIENT_W_CITY -0.0173672234  0.027875820          0.956876226
EXT_SOURCE_1              0.1070025332    -0.049420698         -0.054235555
EXT_SOURCE_2              0.0514324652     0.001326674         -0.259556729
EXT_SOURCE_3              0.1278552551    -0.036729246         -0.031130886
YEARS_BEGIN              -0.0019462010     0.010943274          0.008799567
                        REGION_RATING_CLIENT_W_CITY  EXT_SOURCE_1 EXT_SOURCE_2 EXT_SOURCE_3
CNT_CHILDREN                          0.029457991 -0.0983756192 -0.030569452 -0.058541694
AMT_INCOME_TOTAL                     -0.109885247 -0.0168695485  0.093555557 -0.091137169
AMT_CREDIT                           -0.021562747  0.0815120368  0.068870471  0.022684868
AMT_ANNUITY                          -0.036543731  0.0492107150  0.068361956  0.020021516
AMT_GOODS_PRICE                      -0.028395725  0.0805494292  0.078056615  0.021636922
DAYS_ID_PUBLISH                      -0.017367223  0.1070025332  0.051432465  0.127855255
CNT_FAM_MEMBERS                       0.027875820 -0.0494206980  0.001326674 -0.036729246
REGION_RATING_CLIENT                  0.956876226 -0.0542355552 -0.259556729 -0.031130886
REGION_RATING_CLIENT_W_CITY           1.000000000 -0.0537463366 -0.255676244 -0.030713983
EXT_SOURCE_1                         -0.053746337  1.0000000000  0.121611565  0.116939986
EXT_SOURCE_2                         -0.255676244  0.1216115647  1.000000000  0.104650094
EXT_SOURCE_3                         -0.030713983  0.1169399859  0.104650094  1.000000000
YEARS_BEGIN                           0.007096537 -0.0001820979 -0.007076317  0.004173855
                          YEARS_BEGIN
CNT_CHILDREN            0.0107159501
AMT_INCOME_TOTAL       -0.0007714559
AMT_CREDIT              0.0003686175
AMT_ANNUITY             0.0066760820
AMT_GOODS_PRICE         0.0018234727
DAYS_ID_PUBLISH        -0.0019462010
CNT_FAM_MEMBERS         0.0109432744
REGION_RATING_CLIENT    0.0087995667
REGION_RATING_CLIENT_W_CITY 0.0070965370
EXT_SOURCE_1           -0.0001820979
EXT_SOURCE_2           -0.0070763173
EXT_SOURCE_3            0.0041738547
YEARS_BEGIN             1.0000000000
```

Figure 5. correlation coefficient table.

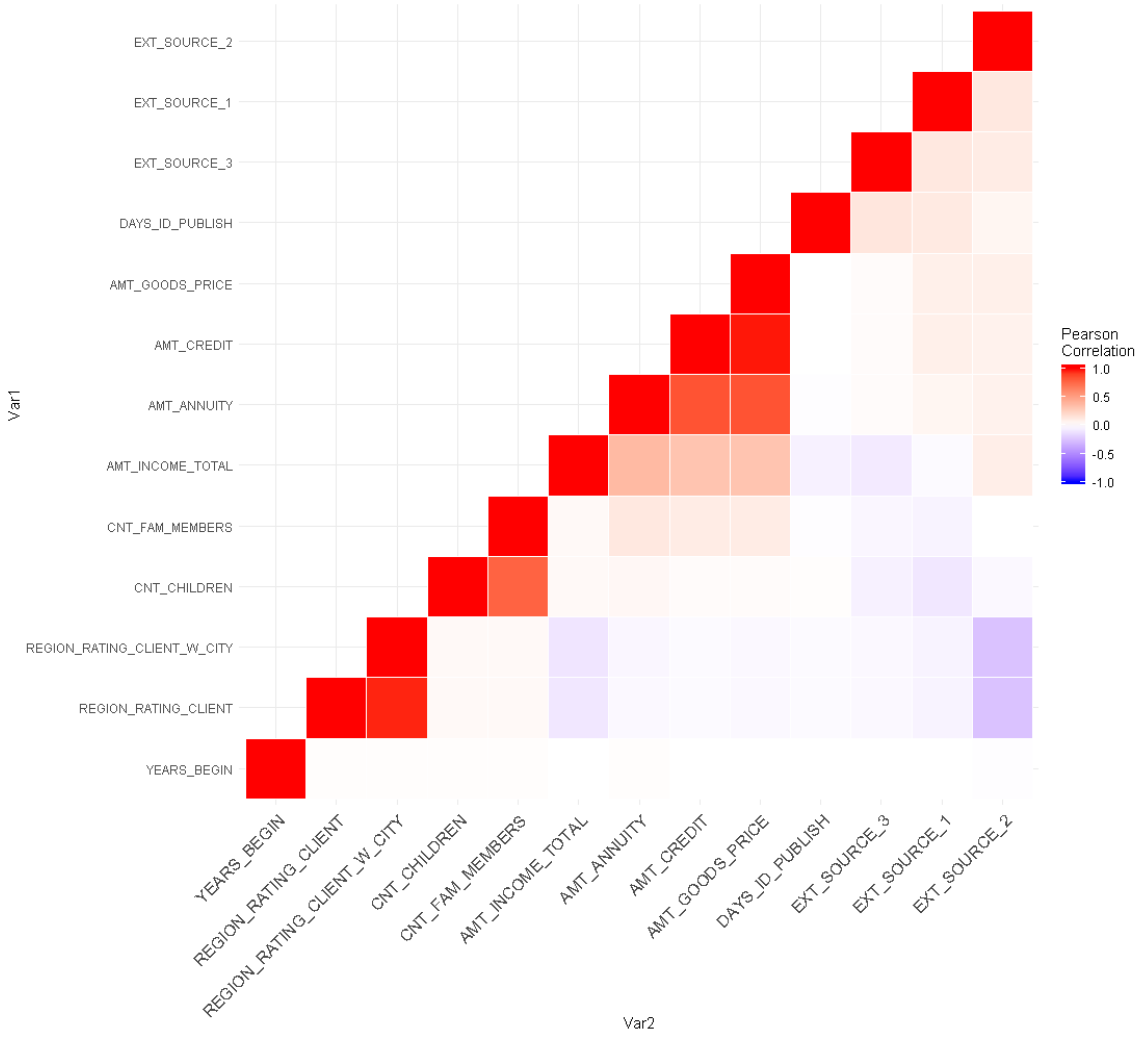


Figure 6. Reordered triangular shaped heatmap showing pearson correlation coefficient. Dark red colored box is +vely correlated and blue colored box is -vely correlated.

2) Modeling

To perform modeling, data is divided into the training and test set based on 80/20 percent. Target value of 0 is considered Yes (i.e., the client is good in terms of the payment capacity) and 1 is considered as No (i.e., the client is risky in terms of payment difficulties). The training and test data set are combined with repeated 10-fold cross-validation to increase the precision of the estimates with maintaining a small bias (Molinaro 2005; Kim 2009).

Classification method is considered for the modeling section due to the categorical nature of the dependent variable. Classification methods considered for this project are Logistic regression, SVM, KNN, and decision trees (i.e., rpart, treebag, rf, adaboost, and C5.0).

Results

1) Descriptive Statistics

The variables considered for the modeling section shows wide dispersion (Fig. 7). Target shows higher number of zero than 1 (18302 vs 1697), CODE_GENER1 also got higher number of zero than 1 (one) or higher female than male (14319 vs 5680). Income shows wide range with some extremely high income, and some low income (Fig. 7).

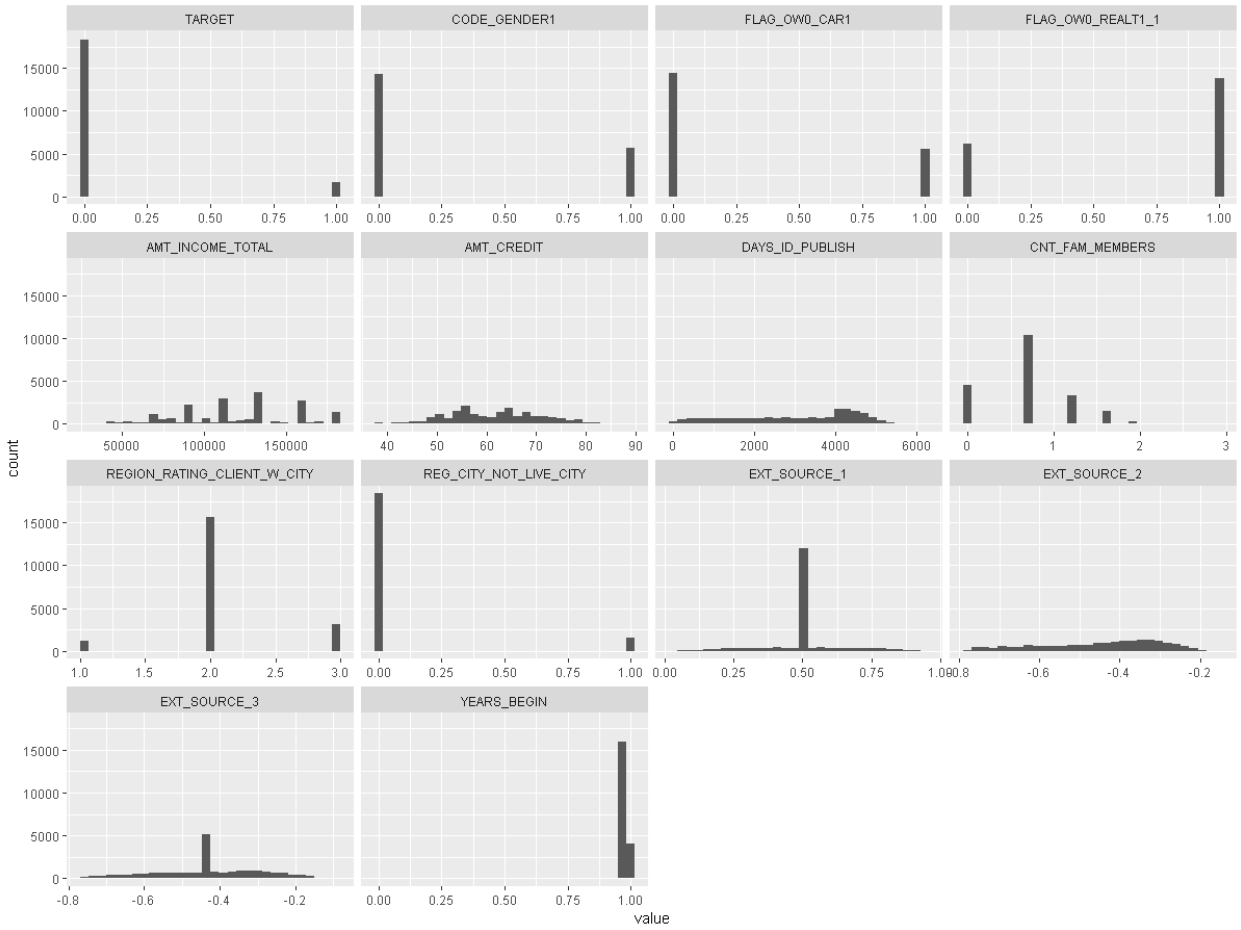


Figure 7. Histograms of the variables selected for the modeling part.

2) Models

2.1) *Logistic Regression*:

Both stepwise logistic model and full models are considered (Fig. 8 and 9). The stepwise logistic regression model is computed with R function of stepAIC() from the Mass package, which is based on the concept of model selection by AIC. Stepwise model selection remove variables which are not important (Fig. 8), while full model includes all the variables (Fig. 9). Income variable, client owning a house or apartment, family member number, and permanent address not matching with current address variables are shown as statistically insignificant by both the full and step-logistic models.

The comparison of the full and stepwise logistic regression model shows the same mean prediction accuracy (i.e., 0.085021). This project will consider the full logistic model for logistic regression. The final model of the logistic regression is run with repeated 10 fold cross validation and ROC, sensitivity, and specificity is calculated (Figure 10). ROC is around 0.7, and sensitivity and specificity are around 0.65.

```
> ## Logisitc Regression
> ## step logistic regression model
> model <- glm(TARGET~., data = train.data, family = binomial) %>% stepAIC(trace = FALSE)
> summary(model)

Call:
glm(formula = TARGET ~ CODE_GENDER1 + FLAG_OWN_CAR1 + AMT_CREDIT +
    DAYS_ID_PUBLISH + REGION_RATING_CLIENT_W_CITY + EXT_SOURCE_1 +
    EXT_SOURCE_2 + EXT_SOURCE_3, family = binomial, data = train.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0235   0.2420   0.3319   0.4500   1.4241

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                 5.975e+00  3.105e-01  19.242  < 2e-16 ***
CODE_GENDER1               -4.481e-01  6.519e-02  -6.874 6.24e-12 ***
FLAG_OWN_CAR1               2.573e-01  7.021e-02   3.664 0.000248 ***
AMT_CREDIT                 -1.164e-02  3.354e-03  -3.471 0.000519 ***
DAYS_ID_PUBLISH             4.894e-05  1.970e-05   2.484 0.012988 *
REGION_RATING_CLIENT_W_CITY -2.499e-01  6.385e-02  -3.913 9.10e-05 ***
EXT_SOURCE_1                1.719e+00  2.177e-01   7.896 2.89e-15 ***
EXT_SOURCE_2                2.931e+00  2.018e-01  14.526  < 2e-16 ***
EXT_SOURCE_3                3.869e+00  2.214e-01  17.477  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 9296.6  on 15999  degrees of freedom
Residual deviance: 8408.6  on 15991  degrees of freedom
AIC: 8426.6

Number of Fisher Scoring iterations: 6
```

Figure 8. Summary of step-wise logistic regression model.

```
> ## Full logistic regression model
> full.model <- glm(TARGET ~., data = train.data, family = binomial)
> summary(full.model)

Call:
glm(formula = TARGET ~ ., family = binomial, data = train.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0164   0.2416   0.3311   0.4507   1.4680

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                  5.548e+00  6.846e-01   8.104 5.33e-16 ***
CODE_GENDER1                -4.633e-01  6.620e-02  -6.998 2.59e-12 ***
FLAG_OWN_CAR1                2.595e-01  7.105e-02   3.653 0.000260 ***
FLAG_OWN_REALT1_1           -1.336e-02  6.331e-02  -0.211 0.832888
AMT_INCOME_TOTAL             1.075e-06  9.149e-07   1.175 0.240003
AMT_CREDIT                  -1.251e-02  3.532e-03  -3.543 0.000396 ***
DAYS_ID_PUBLISH              5.026e-05  1.978e-05   2.540 0.011077 *
CNT_FAM_MEMBERS             -6.772e-02  5.975e-02  -1.133 0.257020
REGION_RATING_CLIENT_W_CITY -2.411e-01  6.426e-02  -3.752 0.000176 ***
REG_CITY_NOT_LIVE_CITY      -5.447e-02  9.764e-02  -0.558 0.576972
EXT_SOURCE_1                 1.708e+00  2.193e-01   7.789 6.77e-15 ***
EXT_SOURCE_2                 2.915e+00  2.023e-01  14.408  < 2e-16 ***
EXT_SOURCE_3                 3.884e+00  2.235e-01  17.383  < 2e-16 ***
YEARS_BEGIN                  4.125e-01  6.184e-01   0.667 0.504764
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 9296.6  on 15999  degrees of freedom
Residual deviance: 8405.1  on 15986  degrees of freedom
AIC: 8433.1

Number of Fisher Scoring iterations: 6
```

Figure 9. Summary of full – logistic regression model.

```
G:/Summer_2019/AdvanceR/Project_Main/home_credit_default_risk/project/
nstead.
> logisticReg
Random Forest

16000 samples
   13 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 2 times)
Summary of sample sizes: 12799, 12801, 12800, 12799, 12801, 12799, ..
.
Addtional sampling using down-sampling

Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   2    0.7124582  0.6631200  0.6582769
   7    0.7024316  0.6498657  0.6596769
  13    0.6960686  0.6513241  0.6460861

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
> plot(logisticReg, scales = list(x=list(log(2))))
>
```

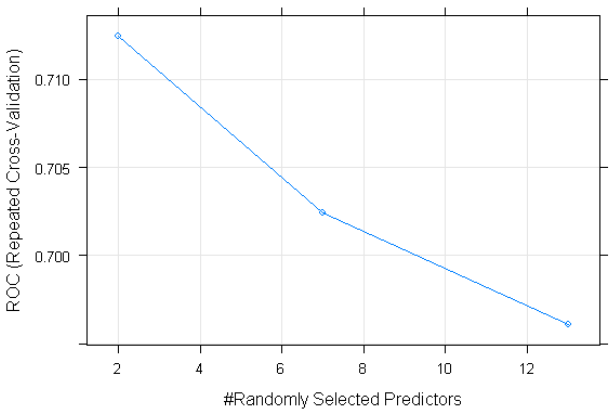


Figure 10. Result of the logistic regression model. ROC curve shows decreasing value with increasing predictors

2.2) <u>KNN model</u>:

KNN model uses the K-closest samples for prediction. KNN method depends on the distance between samples or data, so the scale of the predictors have a significant effect on the

prediction. This require the predicators to be centered and scaled, which is achieved by including the center and scale functions in the model formula. ROC, sensitivity, and specificity of the model are around 0.6 (Fig. 11).

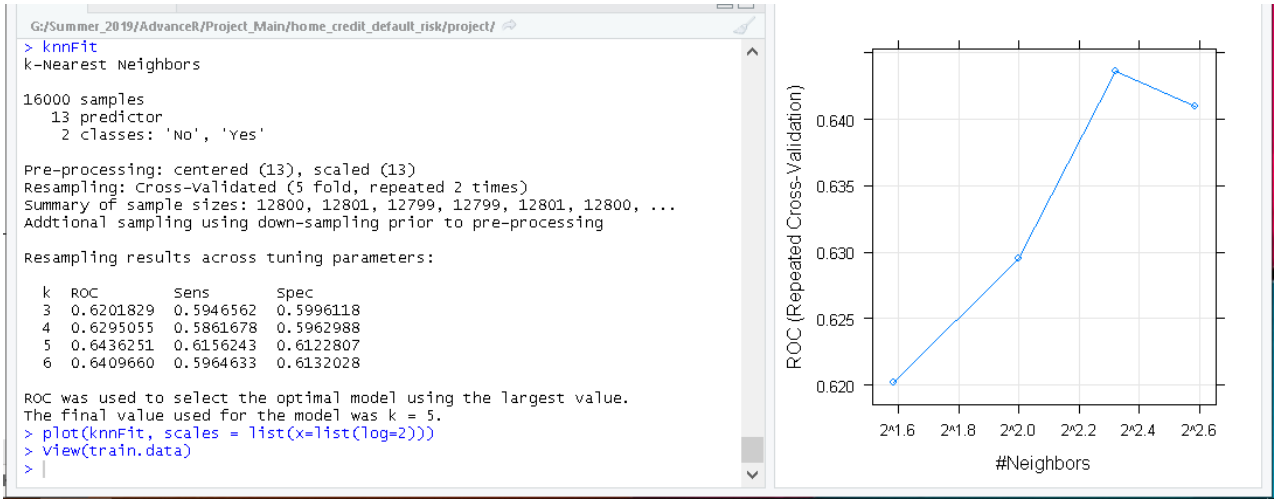


Figure 11. Result of the KNN model. ROC curve shows increasing ROC value with increasing number of neighbors except in top, which shows a decreasing trend.

2.3) *SVM model:*

Two SVM models (i.e., radial and linear) were considered. ROC of SVM_Linear is slightly higher than SVM_Radial (i.e., 0.7283681 > 0.7239669). SVM_Linear is considered for the SVM model.

```
G:/Summer_2019/Advanced/Project_Main/home_credit_default_risk/project/
> #### Comparing all the SVM models
> resamp <- resamples(list(SVM_Radial = svmFit, SVM_Linear = svmFitLinear))
> summary(resamp)

Call:
summary.resamples(object = resamp)

Models: SVM_Radial, SVM_Linear
Number of resamples: 10

ROC
               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM_Radial 0.6955847 0.7207751 0.7248820 0.7239669 0.7367944 0.7442583    0
SVM_Linear 0.7080784 0.7174344 0.7236128 0.7283681 0.7392829 0.7516236    0

Sens
               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM_Radial 0.5992647 0.6507353 0.6752768 0.6664492 0.6872117 0.7047970    0
SVM_Linear 0.6250000 0.6329363 0.6629992 0.6638661 0.6894129 0.7158672    0

Spec
               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM_Radial 0.6431011 0.6643900 0.6665527 0.6672578 0.6750622 0.6893138    0
SVM_Linear 0.6540301 0.6660976 0.6709358 0.6703667 0.6734100 0.6936475    0

> |
```

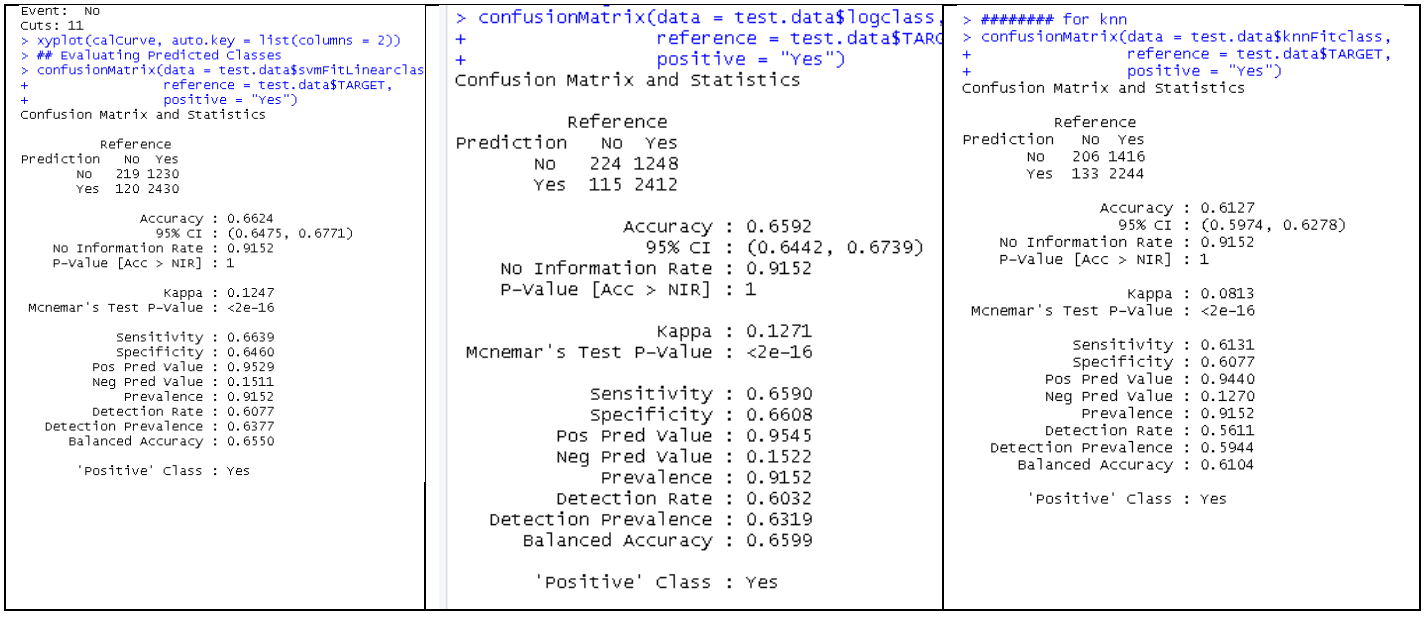
Figure 12. Comparison between SVM models.

2.4) *Comparison of SVM, Logistic Regression and KNN*

Comparison of the SVM (linear), LOGISTIC, and KNN models (Fig. 13 and 14) shows higher ROC of SVM (0.728) followed by Logistic (0.712). ROC of KNN is lowest (0.6436). Sensitivity and specificity values are also higher for SVM (sens: 0.6638, spec: 0.6703) than for logistic and KNN. Accuracy score is also higher for SVM (i.e. 0.6624) than other models (Fig. 14). ROC curve of SVM is higher than that of the logistic regression and KNN (Fig. 15).

```
> resamp <- resamples(list(SVM = svmFitLinear, Logistic = logisticReg, KNN = knnFit))
> summary(resamp)

Call:
summary.resamples(object = resamp)

Models: SVM, Logistic, KNN
Number of resamples: 10

ROC
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM      0.7080784 0.7174344 0.7236128 0.7283681 0.7392829 0.7516236    0
Logistic 0.6815121 0.6962957 0.7152400 0.7124582 0.7284146 0.7504514    0
KNN      0.6078588 0.6402872 0.6500906 0.6436251 0.6523519 0.6571479    0

Sens
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM      0.6250000 0.6329363 0.6629992 0.6638661 0.6894129 0.7158672    0
Logistic 0.5808824 0.6320341 0.6709559 0.6631200 0.6918039 0.7306273    0
KNN      0.5793358 0.6066176 0.6095751 0.6156243 0.6277574 0.6642066    0

Spec
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM      0.6540301 0.6660976 0.6709358 0.6703667 0.6734100 0.6936475    0
Logistic 0.6268351 0.6448087 0.6625683 0.6582769 0.6743769 0.6831683    0
KNN      0.5865483 0.6041667 0.6100389 0.6122807 0.6251179 0.6314891    0
```

Figure 13. Comparison between SVM, LOGISTIC, and KNN models. ROC is higher for SVM followed by logistic and KNN.



```
Event:  No
Cuts: 11
> xyplot(calCurve, auto.key = list(columns = 2))
> ## Evaluating Predicted Classes
> confusionMatrix(data = test.data$svmFitLinearclas
+                 reference = test.data$TARGET,
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No   219 1230
       Yes  120 2430

               Accuracy : 0.6624
                 95% CI : (0.6475, 0.6771)
    No Information Rate : 0.9152
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1247
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6639
            Specificity : 0.6460
         Pos Pred Value : 0.9529
         Neg Pred Value : 0.1511
             Prevalence : 0.9152
         Detection Rate : 0.6077
   Detection Prevalence : 0.6377
      Balanced Accuracy : 0.6550

       'Positive' Class : Yes
```

```
> confusionMatrix(data = test.data$logclass,
+                 reference = test.data$TARG
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No   224 1248
       Yes  115 2412

               Accuracy : 0.6592
                 95% CI : (0.6442, 0.6739)
    No Information Rate : 0.9152
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1271
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6590
            Specificity : 0.6608
         Pos Pred Value : 0.9545
         Neg Pred Value : 0.1522
             Prevalence : 0.9152
         Detection Rate : 0.6032
   Detection Prevalence : 0.6319
      Balanced Accuracy : 0.6599

       'Positive' Class : Yes
```

```
> ######## for knn
> confusionMatrix(data = test.data$knnFitclass,
+                 reference = test.data$TARGET,
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No   206 1416
       Yes  133 2244

               Accuracy : 0.6127
                 95% CI : (0.5974, 0.6278)
    No Information Rate : 0.9152
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0813
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6131
            Specificity : 0.6077
         Pos Pred Value : 0.9440
         Neg Pred Value : 0.1270
             Prevalence : 0.9152
         Detection Rate : 0.5611
   Detection Prevalence : 0.5944
      Balanced Accuracy : 0.6104

       'Positive' Class : Yes
```

Figure 14. comparison of the confusion matrixes of svm, logistic regression, and knn. Accuracy, sensitivity, and specificity are higher for SVM.

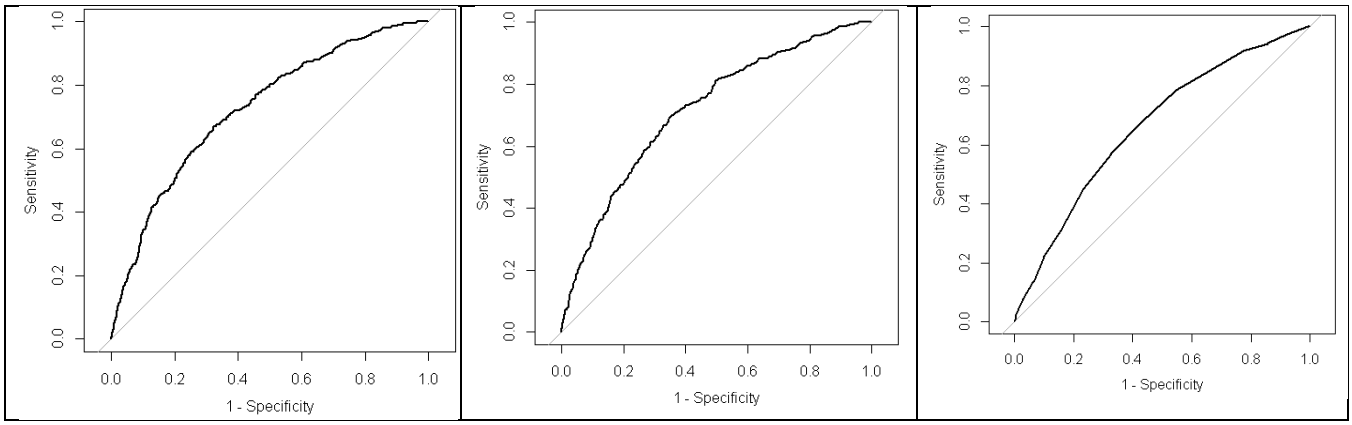


Figure 15. rocCurve of (A) svmFitLinear (the auc[rocCurve] or the area under the curve is 0.7246). (B) logisticRegression (Area under the curve is 0.7173), and (C) KNN (Area under the cruve is 0.6627). rocCurve of SVM covers higher area is more left than rocCurve of logisticRegressions and KNN.

2.5) *Classification Trees and Rule-Based Models*

Classification trees are a member of the tree-based models and are formed by the if-then statements (Kuhn and Johnson, 2013). Five decision trees are considered, and summary of all models (Fig. 16) show ROC is higher for C5.0 (0.7117495) and lowest in CART (0.6575516) (Fig. 16). Sensitivity is higher for C5.0 (0.6583528) and lowest for AdaBoost 0.6251886). Specificity is also highest for C5.0 (0.6802384) and lowest for Bagged (0.6320014) (Fig. 16). The confusion matrix of C5.0 shows accuracy of 0.6964 (Fig. 17).

```
> summary(alltreemodels)

Call:
summary.resamples(object = alltreemodels)

Models: CART, Bagged, RF, AdaBoost, C5.0
Number of resamples: 10

ROC
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART     0.6341874 0.6433476 0.6482112 0.6575516 0.6725847 0.6886037    0
Bagged   0.6470734 0.6682616 0.6749076 0.6730539 0.6776657 0.6892457    0
RF       0.6981621 0.7027915 0.7085138 0.7098469 0.7144631 0.7299098    0
AdaBoost 0.6485663 0.6644653 0.6720000 0.6701030 0.6751568 0.6958261    0
C5.0     0.6917628 0.6994081 0.7149448 0.7117495 0.7234018 0.7257363    0

Sens
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART     0.5588235 0.6259191 0.6353647 0.6266537 0.6421309 0.6801471    0
Bagged   0.5424354 0.6222426 0.6353579 0.6303288 0.6430534 0.6752768    0
RF       0.6139706 0.6397059 0.6470588 0.6561333 0.6771218 0.7121771    0
AdaBoost 0.6029412 0.6099448 0.6243081 0.6251886 0.6365314 0.6544118    0
C5.0     0.5477941 0.6452206 0.6629857 0.6583528 0.6724923 0.7564576    0

Spec
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART     0.5983607 0.6094230 0.6289892 0.6268271 0.6398873 0.6567623    0
Bagged   0.6010929 0.6091337 0.6245508 0.6226612 0.6320014 0.6523224    0
RF       0.6346876 0.6455828 0.6560792 0.6584140 0.6745219 0.6800956    0
AdaBoost 0.6188525 0.6284470 0.6332586 0.6345789 0.6415096 0.6523224    0
C5.0     0.5503585 0.6454064 0.6658700 0.6588272 0.6802384 0.7312158    0
```

Figure 16. Comparison of the decision tree models.

```
> confusionMatrix(data = test.data$C5.0class,
+                 reference = test.data$TARGET,
+                 positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No   213 1088
       Yes  126 2572

               Accuracy : 0.6964
                 95% CI : (0.6819, 0.7107)
    No Information Rate : 0.9152
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1447
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7027
            Specificity : 0.6283
         Pos Pred Value : 0.9533
         Neg Pred Value : 0.1637
             Prevalence : 0.9152
         Detection Rate : 0.6432
   Detection Prevalence : 0.6747
      Balanced Accuracy : 0.6655

       'Positive' Class : Yes
```

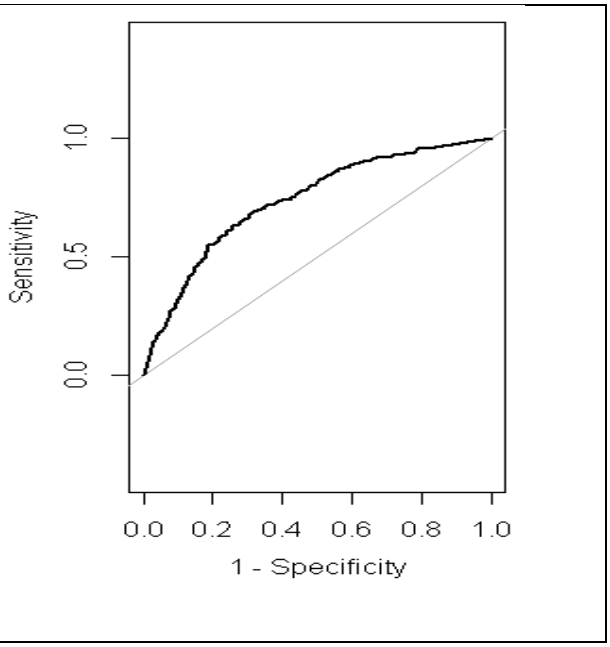


Figure 17. Confusion matrix and rocCurve of C5.0.

Discussion

The target variable is the categorical in nature, i.e., 0 and 1. So, classification machine learning models are used for this project. Classification categorizes samples into groups based on the predictor characteristics through either a mathematical path (e.g., logistic regression) or an algorithmic path (e.g., k-nearest neighbors).

Logistic, KNN, and SVM

Though there are several Logistic Regression models, this project will consider the binary logistic regression due to the 2 possible outcomes (categorical response) of the dependent variables. The possible outcomes are 0 or 1, or Yes or No. For determining the classes, a threshold is set at 0.5. So if predicted value is higher than 0.5, then it is considered as No or 1 to the target variable. Logistic regression model is a popular model due to the simplicity and ability to make inferential statements about the model terms (Kuhn and Johnson, 2013). Logistic

regression model can be effective in case of prediction goal but require user to identify effective representations of the predictor data that yield the best performance (Kuhn and Johnson, 2013).

The KNN method uses the sample's geographic neighborhood to predict the sample's classification (Kuhn and Johnson, 2013). The closeness of the predictors is determined by the distance metric, for which recall of the original measurement scales of the predictor is important. Which means, if the predictors are on widely different scales, distance between samples will be biased towards predictors with larger scales (Kuhn and Johnson, 2013). So centering and scaling all predicators is conducted in KNN equation so that each predictor contribute equally to the distance calculation (Kuhn and Johnson, 2014).

The SVM is based on finding the hyperplane that differentiates the classes of interest. In each classes, data are plotted as a point in n-dimensional space.

The Receiver Operating Characteristic (ROC) curve shows the true positive rate (sensitivity) against the false positive rate (100-Specificity) for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold (Kuhn and Johnson, 2013). Values above the threshold are indicative of a specific event (Altman and Bland 1994; Brown and Davis 2006). The ROC plot helps to choose a threshold that appropriately maximizes the trade-off between sensitivity and specificity. The best model with 1000% accurate prediction shows a ROC curve passing through the upper left corner (100% sensitivity and 100% specificity). So, closeness of the ROC curve to the upper left corner is the indication of the higher accuracy of the test (Zweig and Campbell, 1993). Advantage of the use of ROC curve in evaluation of the model is that, it is insensitive to disparities in the class proportions (Provost et al., 1998) due to being a function of sensitivity and specificity. Disadvantage is the obscure of the information (Kuhn and Johnson, 2013).

Based on the higher values of ROC, Sensitivity, Specificity, and Accuracy, SVM model is considered the best model than Logistic and KNN models. SVM model is expected to identify higher number of both risky and non-risky customer than other two models. Logistic regression model is close to the SVM. KNN model shows the weakest in terms of the confusion matrix and ROC values and curve.

Decision Tree

The strength of the decision tree is the high interpretability, handling many types of predictors and missing data, but the weakness includes the model instability and not stronger to produce optimal predictive performance (Kuhn and Johnson, 2013). The idea of the decision tree is to partition the data into smaller and homogeneous groups in terms of the purity of the nodes. The purity indicates the highest accuracy and lowest misclassification error. Based on the higher value of ROC, specificity, and sensitivity C5.0 is considered as the best decision tree model for the dataset with a accuracy of 0.6964.

Conclusion

The accuracy of the SVM(linear) is 0.6624, sensitivity of 0.6638661 and specificity of 0.6703667, and ROC of 0.7283681. Accuracy value is higher for C5.0 (0.6964) than SVM, but other values confusion matrix and ROC value are favoring SVM than C5.0. So, SVM is considered as the best model to differentiate the risky vs non-risky customer in terms of loan repayment capacity. KNN is the weakest model due to lower performance values. Logistic regression is close to SVM in prediction based on the performance values (i.e., confusion matrix,

ROC). In decision tree, CART is the weakest model. RF (random forest) is close to C5.0 in model performance.

REFERENCES

Altman D, Bland J (1994). "Diagnostic Tests 3: Receiver Operating Characteristic Plots." *British Medical Journal*, 309(6948), 188.

Box G, Cox D (1964). "An Analysis of Transformations." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 211–252.

Brown C, Davis H (2006). "Receiver Operating Characteristics Curves and Related Decision Measures: A Tutorial."*Chemometrics and Intelligent Laboratory Systems*, 80(1), 24–38.

Kim J, Basak J, Holtzman D (2009). "The Role of Apolipoprotein E in Alzheimer's Disease." *Neuron*, 63(3), 287–303.

Kuhn M, Johnson K (2013). "Applied Predictive Modeling".

Molinaro A (2005). "Prediction Error Estimation: A Comparison of Resampling Methods." *Bioinformatics*, **21**(15), 3301–3307.

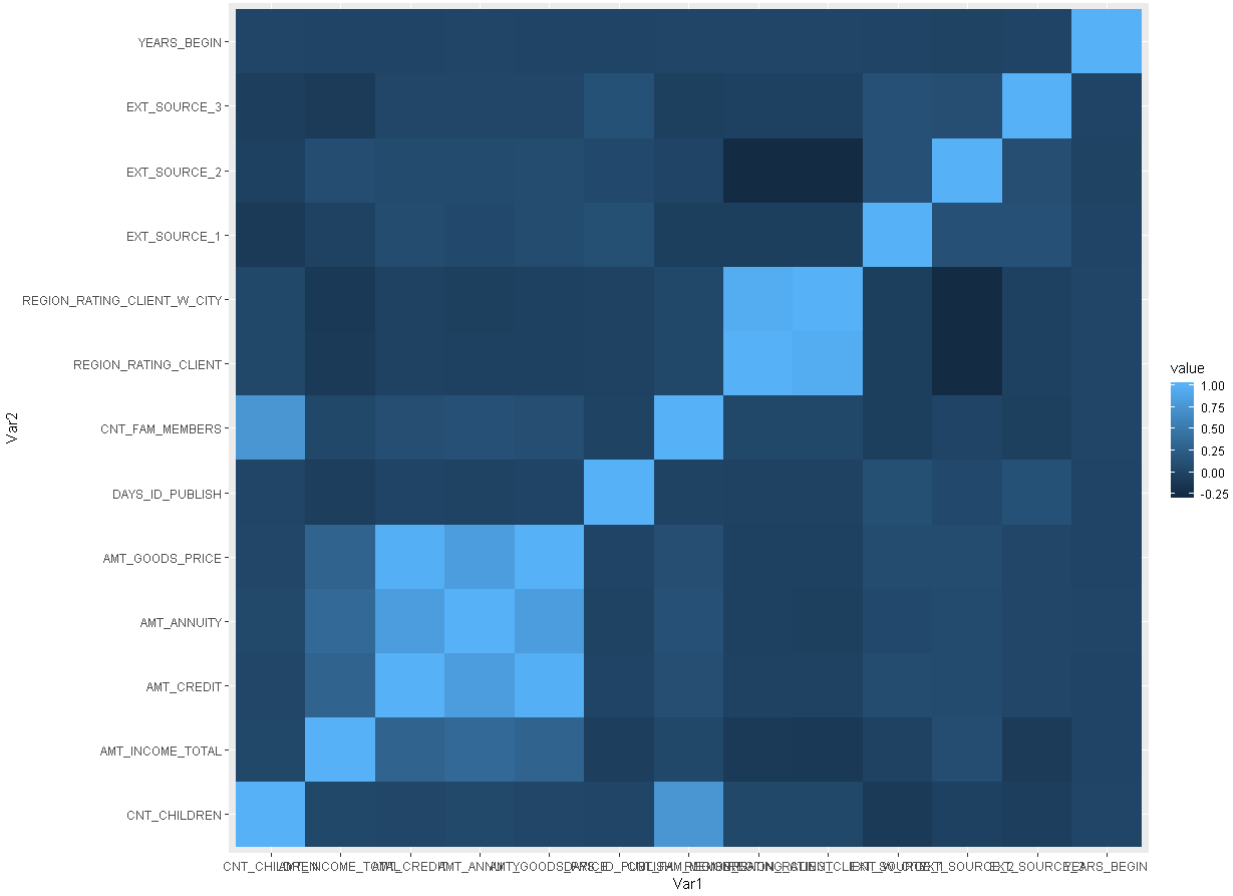
Provost F, Fawcett T, Kohavi R (1998). "The Case Against Accuracy Estimation for Comparing Induction Algorithms." *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453.

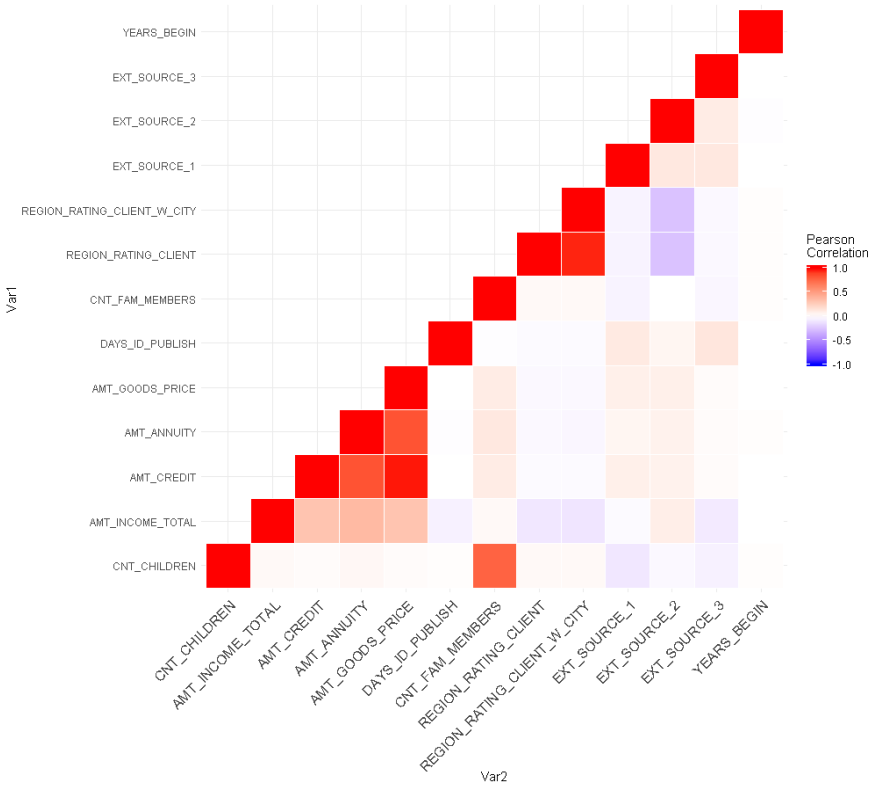
Zweig, M.H. and Campbell G (1993). "Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine". *Clinical Chemistry*, 39, 561-577.

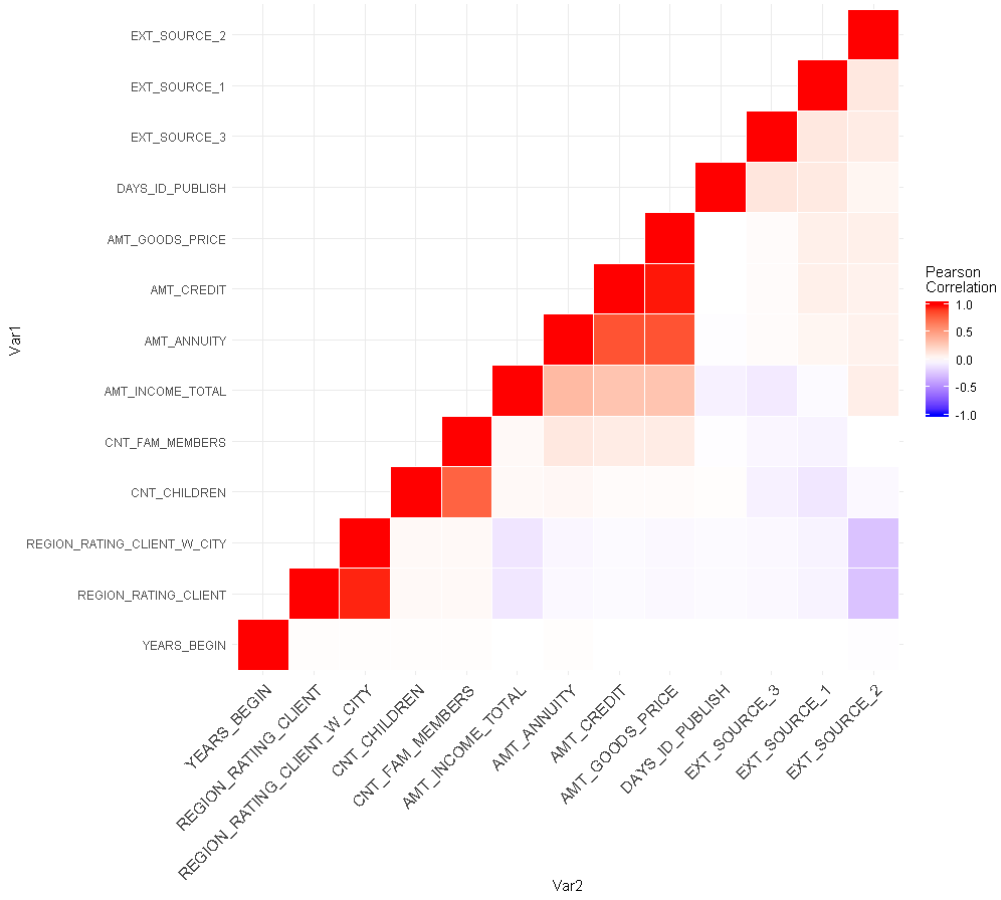
APPENDIXES

APPENDIXE A

Sequential transformation of the heatmap of the correlation coefficient for presentation.




YEARS_BEGIN
EXT_SOURCE_3
EXT_SOURCE_2
EXT_SOURCE_1
REGION_RATING_CLIENT_W_CITY
REGION_RATING_CLIENT
CNT_FAM_MEMBERS
DAYS_ID_PUBLISH
AMT_GOODS_PRICE
AMT_ANNUITY
AMT_CREDIT
AMT_INCOME_TOTAL
CNT_CHILDREN
Var1
Var2
Pearson Correlation
1.0
0.5
0.0
-0.5
-1.0


EXT_SOURCE_2
EXT_SOURCE_1
EXT_SOURCE_3
DAYS_ID_PUBLISH
AMT_GOODS_PRICE
AMT_CREDIT
AMT_ANNUITY
AMT_INCOME_TOTAL
CNT_FAM_MEMBERS
CNT_CHILDREN
REGION_RATING_CLIENT_W_CITY
REGION_RATING_CLIENT
YEARS_BEGIN
Var1
Var2
Pearson Correlation
1.0
0.5
0.0
-0.5
-1.0