Modeling

AUTHOR
Varun Selvam

Load Libraries

```
pacman::p_load(tidyverse,e1071,caret,rminer,rpart,C50,tictoc,kernlab,snow,parallel,doParallel,rand
```

Import Dataset

```
cleaned_dataset <- read_csv("C:/Users/User/Box Sync/Business Analytics Degree/Semesters/Fall Semes
```

Clean the Dataset

There were some changes that I did to the dataset like converting certain numeric and character columns to factors. However, when the file was exported to a csv after completing the EDA, that information was lost. Thus, to deal with that issue, some of the columns must be reconverted to factors.

```
# Mutate any character variables into factors
cleaned_dataset <- cleaned_dataset %>% mutate_if(is.character, as.factor)

# Numeric Variables which should be factors stored as a vector
cat_values <- c("FLAG_EMP_PHONE","FLAG_WORK_PHONE","FLAG_EMAIL","FLAG_PHONE",'REG_REGION_NOT_WORK_

# Utilize mutate to transform all the columns in the cat_values vector to factors.
cleaned_dataset <- cleaned_dataset %>%
  mutate(across(all_of(cat_values), as.factor))

# Display the summary of train_clean
str(cleaned_dataset)
```

```
tibble [237,776 × 64] (S3: tbl_df/tbl/data.frame)
 $ SK_ID_CURR                  : num [1:237776] 1e+05 1e+05 1e+05 1e+05 1e+05 ...
 $ TARGET                      : num [1:237776] 1 0 0 0 0 0 0 0 0 0 ...
 $ NAME_CONTRACT_TYPE          : Factor w/ 2 levels "Cash loans","Revolving loans": 1 1 1 1 1 1 2
1 1 1 ...
 $ CODE_GENDER                 : Factor w/ 2 levels "F","M": 2 1 1 2 2 1 2 1 1 1 ...
 $ FLAG_OWN_CAR                : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...
```

```
 $ FLAG_OWN_REALTY            : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 2 2 2 ...
 $ CNT_CHILDREN               : num [1:237776] 0 0 0 0 0 0 0 1 0 0 ...
 $ AMT_INCOME_TOTAL           : num [1:237776] 202500 270000 135000 121500 99000 ...
 $ AMT_CREDIT                 : num [1:237776] 406598 1293503 312683 513000 490496 ...
 $ AMT_ANNUITY                : num [1:237776] 24701 35699 29687 21866 27518 ...
 $ AMT_GOODS_PRICE            : num [1:237776] 351000 1129500 297000 513000 454500 ...
 $ NAME_TYPE_SUITE            : Factor w/ 7 levels "Children","Family",..: 7 2 7 7 6 1 7 7 1 7
...
 $ NAME_INCOME_TYPE           : Factor w/ 8 levels "Businessman",..: 8 5 8 8 5 4 8 8 4 8 ...
 $ NAME_EDUCATION_TYPE        : Factor w/ 5 levels "Academic degree",..: 5 2 5 5 5 5 5 2 5 5 ...
 $ NAME_FAMILY_STATUS         : Factor w/ 6 levels "Civil marriage",..: 4 2 1 4 2 2 4 2 2 2 ...
 $ NAME_HOUSING_TYPE          : Factor w/ 6 levels "Co-op apartment",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ REGION_POPULATION_RELATIVE  : num [1:237776] 0.0188 0.00354 0.00802 0.02866 0.03579 ...
 $ DAYS_BIRTH                 : num [1:237776] 9461 16765 19005 19932 16941 ...
 $ DAYS_REGISTRATION          : num [1:237776] 3648 1186 9833 4311 4970 ...
 $ DAYS_ID_PUBLISH            : num [1:237776] 2120 291 2437 3458 477 ...
 $ OWN_CAR_AGE                : num [1:237776] 0 0 0 0 0 0 0 0 0 0 ...
 $ FLAG_EMP_PHONE             : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 2 1 2 ...
 $ FLAG_WORK_PHONE            : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 2 ...
 $ FLAG_PHONE                 : Factor w/ 2 levels "0","1": 2 2 1 1 2 1 1 1 2 2 ...
 $ FLAG_EMAIL                 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ OCCUPATION_TYPE            : Factor w/ 19 levels "Accountants",..: 9 4 9 4 9 18 9 4 18 9 ...
 $ CNT_FAM_MEMBERS            : num [1:237776] 1 2 2 1 2 2 1 3 2 2 ...
 $ REGION_RATING_CLIENT       : Factor w/ 3 levels "1","2","3": 2 1 2 2 2 2 2 2 2 2 ...
 $ REGION_RATING_CLIENT_W_CITY : Factor w/ 3 levels "1","2","3": 2 1 2 2 2 2 2 2 2 2 ...
 $ WEEKDAY_APPR_PROCESS_START  : Factor w/ 7 levels "FRIDAY","MONDAY",..: 7 2 7 5 7 7 5 3 1 1 ...
 $ HOUR_APPR_PROCESS_START    : num [1:237776] 10 11 17 11 16 14 8 15 7 10 ...
 $ REG_REGION_NOT_WORK_REGION  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ REG_CITY_NOT_LIVE_CITY     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ REG_CITY_NOT_WORK_CITY     : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ LIVE_CITY_NOT_WORK_CITY    : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ ORGANIZATION_TYPE          : Factor w/ 58 levels "Advertising",..: 6 40 6 38 34 57 10 31 57 5
...
 $ EXT_SOURCE_1               : num [1:237776] 0.083 0.311 0 0 0 ...
 $ EXT_SOURCE_2               : num [1:237776] 0.263 0.622 0.65 0.323 0.354 ...
 $ EXT_SOURCE_3               : num [1:237776] 0.139 0.64 0.555 0.458 0.621 ...
 $ APARTMENTS_MEDI            : num [1:237776] 0.025 0.0968 0.0323 0.0749 0.1135 ...
 $ YEARS_BUILD_MEDI           : num [1:237776] 0.624 0.799 0.591 0.644 0.698 ...
 $ COMMONAREA_MEDI            : num [1:237776] 0.0144 0.0608 0 0.243 0.0076 0.0109 0.0338 0.0276
0.0207 0.0448 ...
 $ ELEVATORS_MEDI             : num [1:237776] 0 0.08 0.08 0.04 0.16 0.16 0.24 0.04 0 0 ...
 $ ENTRANCES_MEDI             : num [1:237776] 0.069 0.0345 0.9655 0.3103 0.0345 ...
 $ FLOORSMAX_MEDI             : num [1:237776] 0.0833 0.2917 0.3333 0.3333 0.0833 ...
 $ FLOORSMIN_MEDI             : num [1:237776] 0.125 0.333 0.375 0.708 0.208 ...
 $ LIVINGAPARTMENTS_MEDI      : num [1:237776] 0.0205 0.0787 0.0513 0.041 0.053 ...
 $ LIVINGAREA_MEDI            : num [1:237776] 0.0193 0.0558 0.0897 0.0618 0.0096 ...
 $ NONLIVINGAPARTMENTS_MEDI   : num [1:237776] 0 0.0039 0 0 0.0155 0.0155 0 0 0.0155 0 ...
 $ OBS_30_CNT_SOCIAL_CIRCLE   : num [1:237776] 2 1 2 0 0 1 2 0 0 0 ...
 $ DEF_30_CNT_SOCIAL_CIRCLE   : num [1:237776] 2 0 0 0 0 0 0 0 0 0 ...
 $ OBS_60_CNT_SOCIAL_CIRCLE   : num [1:237776] 2 1 2 0 0 1 2 0 0 0 ...
 $ DEF_60_CNT_SOCIAL_CIRCLE   : num [1:237776] 2 0 0 0 0 0 0 0 0 0 ...
```

```
 $ DAYS_LAST_PHONE_CHANGE      : num [1:237776] 1134 828 617 1106 2536 ...
 $ FLAG_DOCUMENT_3             : Factor w/ 2 levels "0","1": 2 2 2 1 2 2 1 2 1 2 ...
 $ FLAG_DOCUMENT_6             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
 $ FLAG_DOCUMENT_8             : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ AMT_REQ_CREDIT_BUREAU_HOUR  : num [1:237776] 0 0 0 0 0 0 0 0 0 0 ...
 $ AMT_REQ_CREDIT_BUREAU_DAY   : num [1:237776] 0 0 0 0 0 0 0 0 0 0 ...
 $ AMT_REQ_CREDIT_BUREAU_WEEK  : num [1:237776] 0 0 0 0 0 0 0 0 0 0 ...
 $ AMT_REQ_CREDIT_BUREAU_MON   : num [1:237776] 0 0 0 0 0 0 0 1 0 1 ...
 $ AMT_REQ_CREDIT_BUREAU_QRT   : num [1:237776] 0 0 0 0 1 0 0 0 0 0 ...
 $ AMT_REQ_CREDIT_BUREAU_YEAR  : num [1:237776] 1 0 1 0 1 1 1 0 2 0 ...
 $ House_Attribute_Low_Variance: num [1:237776] -1.463 -1.389 -0.522 -0.522 -0.522 ...
```

The code has read in the file and is now displaying some information such as whether it's a character, factor, etc. for each variable.

Build Majority Classifier

A majority classifier just uses the majority of a class to make predictions about the entire dataset. For instance, in Home Credit's Case, `Target` is a binary variable which can take two values:

Default - 0 No Default - 1.

If the majority of applicants in this dataset do not default, then the majority class is "No Default". The majority classifier will then predict "No Default" for everyone in the dataset.

Check Class Prevalence

```
# Display counts for the target variable and convert to a proportion.
prop.table(table(cleaned_dataset$TARGET)) %>% round(2) # Round to 2 decimal places.
```

```

   0    1 
0.92 0.08 
```

The proportion of 0 and 1 for the cleaned dataset is still the same. 0 represents `No Default` and while 1 represents `Default`. This means that the majority of clients in this dataset did not default.

Get Counts of the Variables

```
# Display counts for each level of the "TARGET" Variable
table(cleaned_dataset$TARGET)
```

```

     0      1 
218531  19245 
```

This just displays the counts for the target variable. There are 218,531 customers who did not default and there are 19,245 customers who did default.

Code Majority Classifier

```r
# No and Yes Count
yes_count <- 218531
no_count <- 19245

# Calculate the total number of observations
total_count <- yes_count + no_count

# Majority class
majority_class <- ifelse(yes_count > no_count, 0, 1) # 1 if yes_count is greater than no count

# Convert Majority Class to factor
majority_class <- factor(majority_class)

#Generate the majority classifier predictions for all instances
predicted <- rep(majority_class, total_count) # Total count represents the total number of instan
# since majority class is zero, this will be repeated for every prediction or total number of ins

# Create metrics list for Majority Classifier
metrics_list = c("ACC","TPR","PRECISION", "F1", "CONF", "AUC", "ROC")

# Generate metrics
majority_output <- mmetric(factor(cleaned_dataset$TARGET), predicted, metrics_list) #Set all the 

majority_output_rounded <- lapply(majority_output, function(x) { # Utilize function to round off 
  if (is.numeric(x)) {
    return(round(x, 2))
  } else {
    return(x)
  }
})

print(majority_output_rounded)
```

```
$res
     ACC       TPR       TPR2 PRECISION1 PRECISION2        F11        F12
   91.91    100.00       0.00      91.91       0.00      95.78       0.00

$conf
      pred
target      0      1
     0 218531      0
     1  19245      0
```

TPR stands for Recall

The above code shows that the accuracy is 91.92% which rounds to 92%. The reason for this is because the majority classifier predicts the majority class which is “No Default”.

The majority classifier has an overall accuracy of 91.93%. This is however because the majority class is 0 which is “No Default”. (Majority Classifiers predict the majority class for everything which means that accuracy will be the majority class value.)

However since the objective is to predict the clients who can pay back their loans, the majority class accuracy can be used as the baseline accuracy. This also means that any future model will have to beat this majority class accuracy. (91.93%)

Although the accuracy of the majority model is interesting as a baseline metric, it might not be very useful due to the high class imbalance.

Thus, some additional metrics that may be desirable to be beat is `Precision 1`, `F1-1` or `ROC-AUC` .

It should be noted that beating the recall for 1(pay back loan) will not be possible since it is 100%. This is because TPR1 represents the majority class and the majority classifier will make no false negative errors. Thus, everything will be classified as positive which divided by the total positive observations results in 100.

ROC-AUC however is a measure of how well the model makes predictions in regards to different classes of the target variable. (This is measured by comparing the True Positive to the True Negative Rate.) This might be a more useful metric due to the imbalance and will reveal more interesting insights about the model.

- True Positive represents correctly predicting that somebody will be able to pack a loan.
- True Negative represents correctly predicting that somebody will have difficulty paying back a loan.
- False Negative represents incorrectly predicting difficulty paying back the loan.
- False Positive represents incorrectly predicting that somebody will pay back the loan when they actually will have difficulty paying back the loan.

Build Random Classifier

```
class_labels = c(0,1) # Create a class label with 2 labels, 0 and 1
class_labels <- factor(class_labels) # Convert Class labels to a factor

set.seed(123) # Set seed for reproducibility
predictions <- sample(x = class_labels, size = total_count,replace = TRUE, prob = c(.5,.5)) # Sam

predictions <- factor(predictions) # Convert the predictions into a factor

random_output <- mmetric(factor(cleaned_dataset$TARGET),predictions,metrics_list) # Generate metr


random_output_rounded <- lapply(random_output, function(x) { # Utilize function to round off metr
  if (is.numeric(x)) {
    return(round(x, 0))
```

```
  } else {
    return(x)
  }
})
print(random_output_rounded) # Print out the rounded metrics
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11        F12
        50         50         50         92          8         65         14

$conf
      pred
target      0      1
     0 109095 109436
     1   9704   9541

$roc
NULL

$lift
NULL
```

The Random Classifier model has an overall accuracy of 50% due to randomly assigning an observation to either default or no default with 50% probability.

Consequently, this has also affected recall which measures the following: predicted class/total observations in the actual class

The objective of any model built will be to have an accuracy higher than 50% and to have a higher recall (greater than TPR1), higher precision1, and a higher F11 score as well.

Partition the Dataset

The below code partitions the dataset into a train and test set with 80% of the data in the train set and 20% of the data in the test set.

This is an important step, because it is possible that models can over fit. In other words, the model may learn noise in the data as opposed to actual patterns. Consequently, when the model sees new data, it won't make accurate predictions and performance will instead plummet.

Thus, to avoid this, cross-validation can be utilized where the model is split into 80% train and 20% test. The 20% test will be used to test the model to ensure that the model is actually learning the patterns and not overfitting.

```
# Make Zero the level to be modelled aka the positive class
cleaned_dataset$TARGET <- factor(cleaned_dataset$TARGET, levels = c("1", "0"))  # Set 0 as the po
```

```
cleaned_dataset <- cleaned_dataset %>%
  select(-SK_ID_CURR)

set.seed(123)
row_indexes <- createDataPartition(cleaned_dataset$TARGET,p=.7,list = FALSE) # This splits the da

train_set <- cleaned_dataset[row_indexes,] # subset 70% of rows from cleaned dataset to the train
test_set <- cleaned_dataset[-row_indexes,] # take all remaining rows which is 30% and move to the
```

The Target Variable has been changed with Target becoming a factor and 0 becoming the second level. This is because R always models the second level of a factor by default.

Additionally, it should be noted that 0 - `No Default` is considered the positive class while 1 - `Default` is considered the negative class.

Decision Tree

Train Decision Tree

We'll first build a simple decision tree model using all the predictors and the default hyper-parameters.

```
tic() # See how long decision tree takes to model
tree_model <- rpart(formula = TARGET ~ ., data=train_set) # Create tree model using the rpart alg
toc() # See how long decision tree takes to model
```

```
13.86 sec elapsed
```

This decision tree trained relatively quickly at 13.86 seconds.

Display Tree Model

```
tree_model # Display the tree model
```

```
n= 166444

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 166444 13472 0 (0.08094014 0.91905986) *
```

```
summary(tree_model) # Important information about the tree model
```

```
Call:
rpart(formula = TARGET ~ ., data = train_set)
  n= 166444

  CP nsplit rel error xerror xstd
1  0      0         1      0    0

Node number 1: 166444 observations
  predicted class=0  expected loss=0.08094014  P(node) =1
    class counts: 13472 152972
   probabilities: 0.081 0.919
```

This decision tree is very simple, with no splits made due to the high class imbalance (92% positive class - No Default). Decision trees typically create splits to group similar observations, aiming to separate classes effectively. However, the imbalance in this dataset made it challenging to find meaningful splits.

While decision trees are powerful, they can overfit if the splits become too precise, capturing noise rather than meaningful patterns.

Compute Evaluation Metrics

The below section will evaluate how this decision tree performs on various metrics for the train and test sets. The most important metric however is AUC-ROC.

Evaluation Train Metrics

Evaluate Decision Tree Performance on the train set.

```
tree_model_train_pred <- predict(tree_model,newdata = train_set) # Make predictions with the tree

mmetric(train_set$TARGET,tree_model_train_pred,metrics_list) # Generate predictions with the tree
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
  91.90599    0.00000  100.00000    0.00000   91.90599    0.00000

       F12        AUC
  95.78230    0.50000

$conf
      pred
target      1      0
     1      0  13472
     0      0 152972
```

The decision tree has a slightly lower accuracy then a majority classifier at 91.91%. The AUC-ROC is however not great at 0.5. A score of 0.5 indicates that the decision's tree ability to distinguish between the two classes (default and no default) is as good as random guessing.

The model's performance however on the minority class is not great with 0% on the minority F1 Score, Precision and Recall. In terms of recall, this means that the model failed to identify any instance of the negative class. This also adversely affects precision because if the model failed to identify any negative class, then one can't correctly identify how much of the model's predictions are negative.

The model however performs fairly well with the positive class being able to identify correctly identify all the positive instances (successfully pay loan) and having a precision of 91.91%. This means that out all the identified positive instances, 95.78% were indeed actually positive. The positive F1 score is also great which indicates that the model can balance between precision and recall for the positive class. (In the context of Home Credit, the model is able to identify all instances of successful loan payback and in instances where it has identified loan payback, it has done so successfully at 91.91%.)

Thus, this model is really good at identifying those who can pay back their loans, since they are the majority class. It however is not great at identifying defaulters since they are the minority class. (It would seem this tree model is almost exactly like a majority classifier since it's performance with Accuracy and Recall is identical. Accuracy is slightly lower at 91.91% vs 92% for a majority classifier.)

Evaluation Test Metrics

```
tree_model_test_pred <- predict(tree_model,newdata = test_set) # Make predictions with tree model

mmetric(test_set$TARGET,tree_model_test_pred,metrics_list) # Generate metrics with tree model pre
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
  91.90686    0.00000  100.00000    0.00000   91.90686    0.00000

       F12        AUC
  95.78278    0.50000

$conf
      pred
target     1     0
     1     0  5773
     0     0 65559
```

The model maintains the same performance as the test set with an accuracy of 91.92%. All of the other metrics are the same for both the positive and negative class as well.

Overall, this decision tree generalizes well to new data since it maintains the same performance on the train set as well. It however also has the same issues like being unable to effectively distinguish between the majority class (no default) and minority class (default). The metrics are also similarly not very great for defaulters. The model struggles broadly based on precision, recall and F1 to identify defaulters.

Decision Tree with Weights

A potential solution to deal with this class imbalance is to assign a greater weight to the minority class. We'll use weight implementation with a Decision Tree.

Create Weights

```
# Set class weights (example: 10x weight for the minority class)
class_weights <- ifelse(cleaned_dataset$TARGET == 1, 10, 1)
```

Assign the weights to the class_weights variable. For minority, a weight of 10 is assigned. This means that the observations in the minority class will be considered 10 time more important than observations in the majority class.

It will also force the model to do splits that maximize the separation of the minority class due to the higher weights.

Create Model with Weights

```
# Fit rpart model with class weights
tic() # See how long model takes to train
tree_model_w <- rpart(TARGET ~ ., data = cleaned_dataset, weights = class_weights, method = "clas
toc() # See how long model takes to train
```

```
146.88 sec elapsed
```

This model was relatively quick to train at 2.45 minutes.

Summary of Weighted Model

```
tree_model_w
```

```
n= 237776

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 237776 192450 0 (0.4682698 0.5317302)
   2) EXT_SOURCE_2< 0.4597299 78506  68187 1 (0.6021228 0.3978772)
```

```
      4) EXT_SOURCE_3< 0.3581222 21925  17328 1 (0.7262473 0.2737527) *
      5) EXT_SOURCE_3>=0.3581222 56581  50859 1 (0.5294275 0.4705725)
       10) EXT_SOURCE_2< 0.160342 10226   8535 1 (0.6645706 0.3354294) *
       11) EXT_SOURCE_2>=0.160342 46355  40310 0 (0.4878137 0.5121863)
         22) OCCUPATION_TYPE=Cleaning staff,Cooking staff,Drivers,HR staff,IT staff,Laborers,Low-skill Laborers,Sales staff,Security staff,Waiters/barmen staff 20020  17757 1 (0.5603288 0.4396712) *
         23) OCCUPATION_TYPE=Accountants,Core staff,High skill tech staff,Managers,Medicine staff,Private service staff,Realty agents,Secretaries,Unemployed 26335  17680 0 (0.4184913 0.5815087) *
    3) EXT_SOURCE_2>=0.4597299 159270  89260 0 (0.3725313 0.6274687)
      6) EXT_SOURCE_3< 0.4127227 44690  40448 1 (0.5118984 0.4881016)
       12) EXT_SOURCE_3< 0.2157927 12465  10711 1 (0.6208630 0.3791370) *
       13) EXT_SOURCE_3>=0.2157927 32225  24880 0 (0.4555358 0.5444642) *
      7) EXT_SOURCE_3>=0.4127227 114580  46840 0 (0.2988465 0.7011535) *
```

```
summary(tree_model_w)
```

```
Call:
rpart(formula = TARGET ~ ., data = cleaned_dataset, weights = class_weights,
    method = "class")
  n= 237776

          CP nsplit rel error    xerror        xstd
1 0.18188101      0 1.0000000 1.0000000 0.001662213
2 0.01774227      1 0.8181190 0.8201715 0.001620175
3 0.01192864      3 0.7826345 0.7732398 0.001600957
4 0.01000000      6 0.7468485 0.7569966 0.001593468

Variable importance
               EXT_SOURCE_2                EXT_SOURCE_3
                         44                          34
       REGION_RATING_CLIENT REGION_RATING_CLIENT_W_CITY
                          4                           4
            OCCUPATION_TYPE      DAYS_LAST_PHONE_CHANGE
                          3                           3
                 DAYS_BIRTH           ORGANIZATION_TYPE
                          2                           2
    HOUR_APPR_PROCESS_START            NAME_INCOME_TYPE
                          1                           1
             FLAG_EMP_PHONE                 CODE_GENDER
                          1                           1

Node number 1: 237776 observations,    complexity param=0.181881
  predicted class=0  expected loss=0.4682698  P(node) =1
    class counts: 192450 218531
   probabilities: 0.468 0.532
  left son=2 (78506 obs) right son=3 (159270 obs)
  Primary splits:
      EXT_SOURCE_2    < 0.4597299  to the left,  improve=10533.350, (0 missing)
```

```
      EXT_SOURCE_3    < 0.4338475  to the left,  improve= 9857.361, (0 missing)
      EXT_SOURCE_1    < 0.4990603  to the left,  improve= 4347.691, (0 missing)
      OCCUPATION_TYPE splits as  RLLRLRRRLLRRRRLRLRL, improve= 3816.934, (0 missing)
      DAYS_BIRTH      < 15342.5    to the left,  improve= 3429.655, (0 missing)
  Surrogate splits:
      REGION_RATING_CLIENT        splits as  RRL, agree=0.623, adj=0.096, (0 split)
      REGION_RATING_CLIENT_W_CITY splits as  RRL, agree=0.623, adj=0.095, (0 split)
      DAYS_LAST_PHONE_CHANGE      < 238.5      to the left,  agree=0.611, adj=0.067, (0 split)
      HOUR_APPR_PROCESS_START     < 8.5        to the left,  agree=0.597, adj=0.034, (0 split)
      DAYS_BIRTH                  < 10328.5    to the left,  agree=0.595, adj=0.029, (0 split)

Node number 2: 78506 observations,    complexity param=0.01192864
  predicted class=1  expected loss=0.3978772  P(node) =0.4169949
    class counts: 103190 68187
   probabilities: 0.602 0.398
  left son=4 (21925 obs) right son=5 (56581 obs)
  Primary splits:
      EXT_SOURCE_3      < 0.3581222  to the left,  improve=3092.759, (0 missing)
      EXT_SOURCE_2      < 0.1526391  to the left,  improve=1870.726, (0 missing)
      ORGANIZATION_TYPE splits as  LLRLLLLLRLLRRLLRLRLRLLRLLLRRRLRRRLRRLLLRLRLRRLRLLRRLRLLLRR,
improve=1575.981, (0 missing)
      DAYS_BIRTH        < 19675.5    to the left,  improve=1547.510, (0 missing)
      EXT_SOURCE_1      < 0.4201047  to the left,  improve=1408.974, (0 missing)
  Surrogate splits:
      EXT_SOURCE_2               < 0.01231413 to the left,  agree=0.636, adj=0.014, (0 split)
      DAYS_BIRTH                 < 8808.5     to the left,  agree=0.633, adj=0.007, (0 split)
      AMT_REQ_CREDIT_BUREAU_QRT  < 3.5        to the right, agree=0.631, adj=0.002, (0 split)
      AMT_REQ_CREDIT_BUREAU_YEAR < 12.5       to the right, agree=0.631, adj=0.001, (0 split)
      ORGANIZATION_TYPE          splits as
LRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRLRRRRRRRRRRLLRRRRRRRR, agree=0.631, adj=0.001, (0 split)

Node number 3: 159270 observations,    complexity param=0.01774227
  predicted class=0  expected loss=0.3725313  P(node) =0.5830051
    class counts: 89260 150344
   probabilities: 0.373 0.627
  left son=6 (44690 obs) right son=7 (114580 obs)
  Primary splits:
      EXT_SOURCE_3      < 0.4127227  to the left,  improve=4921.105, (0 missing)
      EXT_SOURCE_2      < 0.6174253  to the left,  improve=2024.934, (0 missing)
      EXT_SOURCE_1      < 0.5717928  to the left,  improve=1648.520, (0 missing)
      OCCUPATION_TYPE   splits as  RLLRLRRRLLRRRLLLLRL, improve=1586.875, (0 missing)
      ORGANIZATION_TYPE splits as  LLRLLLLLRLRRRLLLLRLLLLLRLLRRRLRRLRRLLRLRLRLLLLRLRRRLRLLLRR,
improve=1202.740, (0 missing)
  Surrogate splits:
      DAYS_BIRTH                 < 9091.5     to the left,  agree=0.657, adj=0.008, (0 split)
      AMT_REQ_CREDIT_BUREAU_YEAR < 8.5        to the right, agree=0.655, adj=0.001, (0 split)
      AMT_REQ_CREDIT_BUREAU_QRT  < 5.5        to the right, agree=0.654, adj=0.000, (0 split)
      ORGANIZATION_TYPE          splits as
RRRRRRRRRRRRRRRRRRLRRRRRRRRRRRRRLRRRRRRRRRRRRRRRRRRRRRRRRR, agree=0.654, adj=0.000, (0 split)
      NAME_INCOME_TYPE           splits as  RRRRRRLR, agree=0.654, adj=0.000, (0 split)
```

```
Node number 4: 21925 observations
  predicted class=1  expected loss=0.2737527  P(node) =0.1540169
    class counts: 45970 17328
   probabilities: 0.726 0.274

Node number 5: 56581 observations,    complexity param=0.01192864
  predicted class=1  expected loss=0.4705725  P(node) =0.2629781
    class counts: 57220 50859
   probabilities: 0.529 0.471
  left son=10 (10226 obs) right son=11 (46355 obs)
  Primary splits:
      EXT_SOURCE_2      < 0.160342   to the left,  improve=1215.6340, (0 missing)
      ORGANIZATION_TYPE splits as  LLRLLLLLRLLRRRLRRRLRLLRLLLRRRLRRRLRRLRLRLRLRRLRLLRRLRLLLRR,
improve=1143.8400, (0 missing)
      OCCUPATION_TYPE   splits as  RLLRLRLRLLRRRRLRLRL, improve=1139.6290, (0 missing)
      EXT_SOURCE_1      < 0.4201047  to the left,  improve=1062.0740, (0 missing)
      DAYS_BIRTH        < 18058      to the left,  improve= 921.2649, (0 missing)
  Surrogate splits:
      REGION_POPULATION_RELATIVE   < 0.0007355  to the left,  agree=0.765, adj=0.001, (0 split)
      NAME_INCOME_TYPE             splits as  -RRRRRLR, agree=0.765, adj=0.000, (0 split)
      ORGANIZATION_TYPE            splits as
RRRRRRRRRRRRRRRRRLRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR, agree=0.765, adj=0.000, (0 split)
      House_Attribute_Low_Variance < 37.28613   to the right, agree=0.765, adj=0.000, (0 split)
      DAYS_BIRTH                   < 25146      to the right, agree=0.765, adj=0.000, (0 split)

Node number 6: 44690 observations,    complexity param=0.01774227
  predicted class=1  expected loss=0.4881016  P(node) =0.2016346
    class counts: 42420 40448
   probabilities: 0.512 0.488
  left son=12 (12465 obs) right son=13 (32225 obs)
  Primary splits:
      EXT_SOURCE_3        < 0.2157927  to the left,  improve=1017.8720, (0 missing)
      EXT_SOURCE_2        < 0.6160659  to the left,  improve= 675.6920, (0 missing)
      EXT_SOURCE_1        < 0.4794052  to the left,  improve= 675.4821, (0 missing)
      OCCUPATION_TYPE     splits as  RLLRLRRLLLRRRLLRLRL, improve= 518.2478, (0 missing)
      NAME_EDUCATION_TYPE splits as  RRLLL, improve= 432.3680, (0 missing)
  Surrogate splits:
      ORGANIZATION_TYPE         splits as  RRRRRRRRRRRRRRRRRRLRRRRLR-
RRRRRRLRRRRRRRRRRRRRRRRRLRRRRRRRR, agree=0.660, adj=0.002, (0 split)
      AMT_REQ_CREDIT_BUREAU_QRT < 4.5        to the right, agree=0.659, adj=0.001, (0 split)
      EXT_SOURCE_2              < 0.4599961  to the left,  agree=0.659, adj=0.001, (0 split)
      COMMONAREA_MEDI           < 0.8491     to the right, agree=0.659, adj=0.000, (0 split)
      AMT_ANNUITY               < 2862       to the left,  agree=0.659, adj=0.000, (0 split)

Node number 7: 114580 observations
  predicted class=0  expected loss=0.2988465  P(node) =0.3813704
    class counts: 46840 109896
   probabilities: 0.299 0.701

Node number 10: 10226 observations
  predicted class=1  expected loss=0.3354294  P(node) =0.06191284
```

```
     class counts: 16910  8535
    probabilities: 0.665 0.335 

Node number 11: 46355 observations,    complexity param=0.01192864
  predicted class=0  expected loss=0.4878137  P(node) =0.2010653
    class counts: 40310 42324
   probabilities: 0.488 0.512 
  left son=22 (20020 obs) right son=23 (26335 obs)
  Primary splits:
      OCCUPATION_TYPE   splits as  RLLRLRLLLLRRRRLRLRL, improve=830.7899, (0 missing)
      ORGANIZATION_TYPE splits as  LLRLLLLLRLLRRRLLRRLLLLRRLLRRRLRRRRRRLRLRLRLRRLRLLRRLRLLLRR, improve=779.6750, (0 missing)
      EXT_SOURCE_1      < 0.4195094  to the left,  improve=749.4841, (0 missing)
      CODE_GENDER       splits as  RL, improve=634.4173, (0 missing)
      DAYS_BIRTH        < 18254.5    to the left,  improve=612.1099, (0 missing)
  Surrogate splits:
      ORGANIZATION_TYPE splits as  RLRLLLLLRLRRRLLLLRLLLLLRLLLRRRRRRRRRRRLRLRLRRLRLLRRLRLLLRR, agree=0.757, adj=0.502, (0 split)
      NAME_INCOME_TYPE  splits as  -LRRRRRL, agree=0.715, adj=0.416, (0 split)
      FLAG_EMP_PHONE    splits as  RL, agree=0.688, adj=0.361, (0 split)
      DAYS_BIRTH        < 19505.5    to the left,  agree=0.658, adj=0.301, (0 split)
      CODE_GENDER       splits as  RL, agree=0.620, adj=0.223, (0 split)

Node number 12: 12465 observations
  predicted class=1  expected loss=0.379137  P(node) =0.0687404
    class counts: 17540 10711
   probabilities: 0.621 0.379 

Node number 13: 32225 observations
  predicted class=0  expected loss=0.4555358  P(node) =0.1328942
    class counts: 24880 29737
   probabilities: 0.456 0.544 

Node number 22: 20020 observations
  predicted class=1  expected loss=0.4396712  P(node) =0.09826975
    class counts: 22630 17757
   probabilities: 0.560 0.440 

Node number 23: 26335 observations
  predicted class=0  expected loss=0.4184913  P(node) =0.1027955
    class counts: 17680 24567
   probabilities: 0.418 0.582 
```

The weighted decision tree has performed much better with 12 splits in contrast to the un-weighted decision tree. Additionally, we can see that the top 3 important predictors for predicting successful loan repayment were `EXT_SOURCE_2`, `EXT_SOURCE_3` and `OCCUPATION_TYPE`.

Compute Evaluation Metrics

The weighted decision tree will now be cross-validated with the train and test sets.

Evaluation Train Metrics

```
tree_model_w_train <- predict(tree_model_w,newdata = train_set) # Make predictions for weighted tr
mmetric(train_set$TARGET,tree_model_w_train,metrics_list) #Generate Metrics to analyze weighted tr
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
73.4493283 53.5926366 75.1980755 15.9875996 94.8451568 24.6281894

       F12        AUC
83.8865877 0.6815838

$conf
      pred
target      1      0
     1   7220   6252
     0  37940 115032
```

The weighted decision tree does much better on the ROC-AUC metric at 0.68. This is much better than the regular Decision Tree which had a metric of 0.50. This means that model is doing a much better job at discriminating between the positive class (no default) and negative class (default).

ROC-AUC can go up to 1, so this model is doing relatively well on this metric. It also does better on all the metrics for the minority class as well. For instance, this model is able to identify 53.59% of all negative instances (default). This is significant progress since the un-weighted decision tree was at 0% for this recall.

Additionally, for precision, the minority class metric is at 16%. This much better then the un-weighted decision tree which was at 0%. In the context of Home Credit, this means that the model out of all the negative instances that weighted tree identified, 16% were actually negative. This in some ways is more concerning then accuracy because it costs more money (median of 24,412 dollars when someone defaults) compared to rejecting someone who could pay back the loan. (median of 23,800 dollars for incorrect rejections)

The F1 Score is much better when compared to the un-weighted decision tree model for the negative class at 24.63%. Although better than the un-weighted decision tree, it could still be better at striking a balance between the precision ad recall. 24.63% is a bit low.

The positive class (No Default) for Precision and Accuracy has a decline of approximately 23 to 25 percent. All the other metrics have stayed the same. This indicates that the weights enabled the decision tree to better predict some of the negative class. It however has reduced the performance with identifying the positive class. The precision for the un-weighted decision tree was 100% which meant that out of all the positive instances predicted by the model, all of them were correct. However, in this case, it decreased to 75.2%. This means that out all the positive predicted instances, 75.2% of the positive predicted instances were correct.

- The overall accuracy which represents overall model predictions also decreased to 75.2% from 91.92%. This means that the overall ability to correctly predict predictions has gone down. This however is not a great metric to gauge model performance due to high class imbalance. (Correctly predicting the majority class alone would give a very high accuracy since it comprises 91% of the dataset. Hence, although this is concerning, it's not very important when compared to the decline in other models.)

Evaluation Test Metrics

```
# Make predictions with the Weighted Tree Model on the test set.
tree_model_w_test <- predict(tree_model_w,newdata = test_set)
mmetric(test_set$TARGET,tree_model_w_test,metrics_list) # Generate predictions on the test set to
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
73.2532384 53.4384202 74.9980933 15.8400082 94.8164147 24.4366114

       F12        AUC
83.7507984  0.6787567

$conf
      pred
target     1     0
     1  3085  2688
     0 16391 49168
```

The metrics from the test set are basically the same as the train set metrics. This means that the model has low variance because it's able to generalize well to new data and maintain the same performance that it maintained on the train set. It however has high bias since the metrics for the minority class are very low. (The minority class metrics are the same when compared to the train set, but it's still low overall).

Thus this is a low variance model since it cannot adequately predict the minority class, but it does this consistently for both the train and test sets. It however has high bias because as previously mentioned, the performance is not great on any of the metrics for the minority class. (Model does however perform better than an un-weighted decision tree.)

Naive_Bayes

The next model that will be examined is Naive Bayes, where the Bayes Theorem is used to classify instances.

Build Model

The code chunk down below trains a Naive Bayes model on the train set.

```
# Build the Naive Bayes Model utilizing the e1701 package.
tic() # See how long it takes model to train
nb_model <- naiveBayes(TARGET ~ ., data = train_set) # train the model on the train dataset.
toc() # See how long it takes model to train\
```

```
1.3 sec elapsed
```

This model was very quick to train with a total training time of 1.3 seconds.

Summary of Naive_Bayes Model

The below code will display more information about the Naive Bayes Model.

```
nb_model # Display naive bayes model
```

```

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         1          0 
0.08094014 0.91905986 

Conditional probabilities:
   NAME_CONTRACT_TYPE
Y   Cash loans Revolving loans
  1 0.93519893      0.06480107
  0 0.90263578      0.09736422

   CODE_GENDER
Y           F         M
  1 0.6209175 0.3790825
  0 0.7177261 0.2822739

   FLAG_OWN_CAR
Y           N         Y
  1 0.8210362 0.1789638
  0 0.7698468 0.2301532

   FLAG_OWN_REALTY
Y           N         Y
  1 0.3225950 0.6774050
  0 0.3043367 0.6956633

   CNT_CHILDREN
```

```
Y        [,1]      [,2]
  1 0.4053593 0.6565812
  0 0.3587062 0.6316833

   AMT_INCOME_TOTAL
Y       [,1]     [,2]
  1 145979.2 55964.00
  0 148537.0 58329.06

   AMT_CREDIT
Y       [,1]     [,2]
  1 537284.9 331122.8
  0 570239.4 380628.3

   AMT_ANNUITY
Y       [,1]     [,2]
  1 25523.72 11738.60
  0 25706.08 13210.51

   AMT_GOODS_PRICE
Y       [,1]     [,2]
  1 470800.3 296968.7
  0 512669.5 347981.4

   NAME_TYPE_SUITE
Y       Children       Family Group of people      Other_A      Other_B
  1 0.0103176960 0.1186906176    0.0011876485 0.0025979810 0.0074970309
  0 0.0114726878 0.1318999555    0.0008825144 0.0028044348 0.0059030411
   NAME_TYPE_SUITE
Y   Spouse, partner Unaccompanied
  1    0.0335510689  0.8261579572
  0    0.0355489894  0.8114883770

   NAME_INCOME_TYPE
Y    Businessman Commercial associate Maternity leave    Pensioner
  1 0.000000e+00         2.071704e-01    1.484561e-04 1.313836e-01
  0 1.961143e-05         2.205436e-01    6.537144e-06 2.050702e-01
   NAME_INCOME_TYPE
Y   State servant      Student   Unemployed      Working
  1  5.262767e-02 0.000000e+00 4.453682e-04 6.082245e-01
  0  7.092148e-02 9.805716e-05 4.576001e-05 5.032947e-01

   NAME_EDUCATION_TYPE
Y   Academic degree Higher education Incomplete higher Lower secondary
  1    0.0000000000     0.1564726841      0.0345902613    0.0164786223
  0    0.0004379887     0.2340820542      0.0328949089    0.0126362995
   NAME_EDUCATION_TYPE
Y   Secondary / secondary special
  1                  0.7924584323
  0                  0.7199487488
```

```
   NAME_FAMILY_STATUS
Y   Civil marriage      Married    Separated Single / not married      Unknown
  1   1.203236e-01 5.765291e-01 6.791865e-02         1.906176e-01 0.000000e+00
  0   9.633789e-02 6.224603e-01 6.745679e-02         1.519821e-01 6.537144e-06
   NAME_FAMILY_STATUS
Y          Widow
  1 4.461105e-02
  0 6.175640e-02

   NAME_HOUSING_TYPE
Y   Co-op apartment House / apartment Municipal apartment Office apartment
  1     0.003562945       0.852360451         0.040676960      0.007051663
  0     0.002954789       0.890522449         0.037176738      0.007981853
   NAME_HOUSING_TYPE
Y   Rented apartment With parents
  1      0.025089074  0.071258907
  0      0.014787020  0.046577151

   REGION_POPULATION_RELATIVE
Y         [,1]       [,2]
  1 0.01904157 0.01161157
  0 0.02052782 0.01317459

   DAYS_BIRTH
Y       [,1]     [,2]
  1 14936.41 4277.300
  0 16294.58 4452.129

   DAYS_REGISTRATION
Y       [,1]     [,2]
  1 4591.003 3334.466
  0 5136.933 3579.263

   DAYS_ID_PUBLISH
Y       [,1]     [,2]
  1 2718.013 1520.271
  0 3018.923 1504.732

   OWN_CAR_AGE
Y       [,1]     [,2]
  1 1.170279 2.932282
  0 1.416298 3.073040

   FLAG_EMP_PHONE
Y           0         1
  1 0.1318290 0.8681710
  0 0.2051029 0.7948971

   FLAG_WORK_PHONE
Y           0         1
  1 0.7594270 0.2405730
```

```
  0 0.8029639 0.1970361

   FLAG_PHONE
Y          0         1
  1 0.7507423 0.2492577
  0 0.7129082 0.2870918

   FLAG_EMAIL
Y           0          1
  1 0.94885689 0.05114311
  0 0.94823889 0.05176111

   OCCUPATION_TYPE
Y   Accountants Cleaning staff Cooking staff  Core staff     Drivers
  1 0.019596200    0.020486936   0.028058195 0.074970309 0.066359857
  0 0.032770703    0.016584734   0.020899250 0.092716314 0.047505426
   OCCUPATION_TYPE
Y   High skill tech staff    HR staff    IT staff    Laborers
  1           0.027909739 0.001484561 0.001410333 0.233447150
  0           0.037032921 0.001961143 0.001542766 0.169573517
   OCCUPATION_TYPE
Y   Low-skill Laborers    Managers Medicine staff Private service staff
  1        0.015290974 0.040380048    0.025163302           0.007274347
  0        0.006086081 0.055899119    0.030175457           0.009230447
   OCCUPATION_TYPE
Y   Realty agents Sales staff Secretaries Security staff  Unemployed
  1   0.002078385 0.132348575 0.004230998    0.028058195 0.264251781
  0   0.002438355 0.109072249 0.004386424    0.020454724 0.337251262
   OCCUPATION_TYPE
Y   Waiters/barmen staff
  1          0.007200119
  0          0.004419109

   CNT_FAM_MEMBERS
Y       [,1]      [,2]
  1 2.102212 0.8647818
  0 2.077511 0.8346114

   REGION_RATING_CLIENT
Y           1          2          3
  1 0.05871437 0.72995843 0.21132720
  0 0.09408911 0.75789687 0.14801402

   REGION_RATING_CLIENT_W_CITY
Y           1          2          3
  1 0.06250000 0.73975653 0.19774347
  0 0.09997254 0.76535575 0.13467170

   WEEKDAY_APPR_PROCESS_START
Y       FRIDAY     MONDAY   SATURDAY     SUNDAY   THURSDAY    TUESDAY
  1 0.16730998 0.15743765 0.10666568 0.05181116 0.16448931 0.18215558
```

```
  0 0.16195121 0.16477525 0.11113799 0.05259132 0.16496483 0.17436524
   WEEKDAY_APPR_PROCESS_START
Y    WEDNESDAY
  1 0.17013064
  0 0.17021416

   HOUR_APPR_PROCESS_START
Y       [,1]     [,2]
  1 11.85474 3.259753
  0 12.10839 3.213373

   REG_REGION_NOT_WORK_REGION
Y            0          1
  1 0.94930226 0.05069774
  0 0.95648223 0.04351777

   REG_CITY_NOT_LIVE_CITY
Y            0          1
  1 0.88145784 0.11854216
  0 0.92607144 0.07392856

   REG_CITY_NOT_WORK_CITY
Y           0         1
  1 0.7007868 0.2992132
  0 0.7850979 0.2149021

   LIVE_CITY_NOT_WORK_CITY
Y           0         1
  1 0.7826603 0.2173397
  0 0.8334597 0.1665403

   ORGANIZATION_TYPE
Y    Advertising  Agriculture         Bank Business Entity Type 1
  1 1.781473e-03 9.204276e-03 5.864014e-03           1.996734e-02
  0 1.196297e-03 7.308527e-03 8.557122e-03           1.883351e-02
   ORGANIZATION_TYPE
Y   Business Entity Type 2 Business Entity Type 3     Cleaning Construction
  1           3.377375e-02           2.439875e-01 1.261876e-03 2.946853e-02
  0           3.372513e-02           2.075151e-01 8.563659e-04 1.836937e-02
   ORGANIZATION_TYPE
Y        Culture  Electricity    Emergency   Government        Hotel
  1 1.113420e-03 2.969121e-03 1.484561e-03 3.028504e-02 3.191805e-03
  0 1.274743e-03 2.889418e-03 1.555840e-03 3.403891e-02 3.360092e-03
   ORGANIZATION_TYPE
Y        Housing Industry: type 1 Industry: type 10 Industry: type 11
  1 7.793943e-03     5.121734e-03      3.711401e-04      9.278504e-03
  0 9.524619e-03     3.033235e-03      3.660801e-04      8.615956e-03
   ORGANIZATION_TYPE
Y   Industry: type 12 Industry: type 13 Industry: type 2 Industry: type 3
  1      5.195962e-04      3.711401e-04     1.855701e-03     1.350950e-02
  0      1.202835e-03      2.288000e-04     1.438172e-03     1.048558e-02
```

```
   ORGANIZATION_TYPE
Y   Industry: type 4 Industry: type 5 Industry: type 6 Industry: type 7
  1     3.266033e-03     1.707245e-03     2.969121e-04     4.230998e-03
  0     2.824046e-03     1.993829e-03     3.791544e-04     4.183772e-03
   ORGANIZATION_TYPE
Y   Industry: type 8 Industry: type 9    Insurance Kindergarten Legal Services
  1     7.422803e-05     7.868171e-03 1.336105e-03 2.048694e-02   7.422803e-04
  0     7.844573e-05     9.635750e-03 1.928457e-03 2.405015e-02   7.779201e-04
   ORGANIZATION_TYPE
Y       Medicine     Military       Mobile        Other       Police
  1 3.147268e-02 4.527910e-03 1.187648e-03 5.136580e-02 4.899050e-03
  0 3.864760e-02 7.105876e-03 9.740345e-04 5.380070e-02 7.099338e-03
   ORGANIZATION_TYPE
Y         Postal      Realtor     Religion   Restaurant       School
  1 8.981591e-03 1.484561e-03 2.226841e-04 8.833135e-03 2.278800e-02
  0 7.530790e-03 1.065554e-03 2.810972e-04 6.066470e-03 3.086186e-02
   ORGANIZATION_TYPE
Y       Security Security Ministries Self-employed     Services      Telecom
  1 1.336105e-02        3.266033e-03  1.552108e-01 4.453682e-03 1.633017e-03
  0 9.831865e-03        6.027247e-03  1.205449e-01 5.334310e-03 1.856549e-03
   ORGANIZATION_TYPE
Y   Trade: type 1 Trade: type 2 Trade: type 3 Trade: type 4 Trade: type 5
  1  1.707245e-03  5.864014e-03  1.558789e-02  1.484561e-04  7.422803e-05
  0  1.117852e-03  6.216824e-03  1.180608e-02  2.026515e-04  1.699657e-04
   ORGANIZATION_TYPE
Y   Trade: type 6 Trade: type 7 Transport: type 1 Transport: type 2
  1  1.410333e-03  3.050772e-02      3.711401e-04      7.719715e-03
  0  2.137646e-03  2.573674e-02      5.752687e-04      6.844390e-03
   ORGANIZATION_TYPE
Y   Transport: type 3 Transport: type 4   Unemployed   University
  1      6.903207e-03      1.840855e-02 1.318290e-01 2.597981e-03
  0      3.020161e-03      1.547342e-02 2.050767e-01 4.366812e-03

   EXT_SOURCE_1
Y        [,1]      [,2]
  1 0.1546697 0.2294750
  0 0.2213326 0.2894479

   EXT_SOURCE_2
Y        [,1]      [,2]
  1 0.4051446 0.2134464
  0 0.5204589 0.1866775

   EXT_SOURCE_3
Y        [,1]      [,2]
  1 0.4155140 0.2085214
  0 0.5205069 0.1911999

   APARTMENTS_MEDI
Y         [,1]       [,2]
  1 0.09271722 0.09968514
```

```
  0 0.09974832 0.10473049

   YEARS_BUILD_MEDI
Y         [,1]      [,2]
  1 0.7120983 0.1039413
  0 0.7180303 0.1049106

   COMMONAREA_MEDI
Y          [,1]       [,2]
  1 0.03858365 0.06819365
  0 0.03956110 0.06980448

   ELEVATORS_MEDI
Y          [,1]      [,2]
  1 0.06553741 0.1173198
  0 0.07283084 0.1242843

   ENTRANCES_MEDI
Y         [,1]      [,2]
  1 0.1145913 0.1012653
  0 0.1217426 0.1032362

   FLOORSMAX_MEDI
Y         [,1]      [,2]
  1 0.2033872 0.1270164
  0 0.2137140 0.1318733

   FLOORSMIN_MEDI
Y         [,1]      [,2]
  1 0.2167934 0.1518849
  0 0.2209746 0.1538992

   LIVINGAPARTMENTS_MEDI
Y          [,1]       [,2]
  1 0.08159729 0.09069968
  0 0.08475313 0.09246330

   LIVINGAREA_MEDI
Y          [,1]      [,2]
  1 0.09477972 0.1006433
  0 0.10078320 0.1049539

   NONLIVINGAPARTMENTS_MEDI
Y           [,1]       [,2]
  1 0.005724406 0.04135401
  0 0.006392661 0.04639039

   OBS_30_CNT_SOCIAL_CIRCLE
Y        [,1]     [,2]
  1 1.382349 2.055743
  0 1.320078 2.012631
```

```
   DEF_30_CNT_SOCIAL_CIRCLE
Y        [,1]      [,2]
  1 0.1946259 0.5156371
  0 0.1391889 0.4320328

   OBS_60_CNT_SOCIAL_CIRCLE
Y       [,1]     [,2]
  1 1.366538 2.038127
  0 1.304154 1.995002

   DEF_60_CNT_SOCIAL_CIRCLE
Y         [,1]      [,2]
  1 0.14095903 0.4305634
  0 0.09722694 0.3510257

   DAYS_LAST_PHONE_CHANGE
Y       [,1]     [,2]
  1 790.8948 749.4911
  0 963.3793 827.0158

   FLAG_DOCUMENT_3
Y           0         1
  1 0.2114014 0.7885986
  0 0.2864184 0.7135816

   FLAG_DOCUMENT_6
Y            0          1
  1 0.93401128 0.06598872
  0 0.89939989 0.10060011

   FLAG_DOCUMENT_8
Y            0          1
  1 0.94380938 0.05619062
  0 0.93693617 0.06306383

   AMT_REQ_CREDIT_BUREAU_HOUR
Y          [,1]       [,2]
  1 0.005344418 0.07590554
  0 0.005504275 0.07752460

   AMT_REQ_CREDIT_BUREAU_DAY
Y          [,1]      [,2]
  1 0.007348575 0.1056185
  0 0.006027247 0.1015160

   AMT_REQ_CREDIT_BUREAU_WEEK
Y         [,1]      [,2]
  1 0.02983967 0.1914986
  0 0.02972439 0.1913301
```

```
   AMT_REQ_CREDIT_BUREAU_MON
Y         [,1]      [,2]
  1 0.1855701 0.6598935
  0 0.2230670 0.8388589

   AMT_REQ_CREDIT_BUREAU_QRT
Y         [,1]      [,2]
  1 0.2137025 0.5801348
  0 0.2311534 0.5767163

   AMT_REQ_CREDIT_BUREAU_YEAR
Y        [,1]     [,2]
  1 1.853845 1.816936
  0 1.775632 1.769809

   House_Attribute_Low_Variance
Y            [,1]     [,2]
  1 -0.165075244 1.927351
  0  0.006168287 2.272093
```

```
summary(nb_model) # Display a summary/key highlights from the model
```

```
          Length Class  Mode
apriori    2     table  numeric
tables    62     -none- list
levels     2     -none- character
isnumeric 62     -none- logical
call       4     -none- call
```

The apriori probablities which can be thought of as "inital inferences" or guesses represents the probability of selecting each class based on their proportion before looking at any of the other data. In this case the "Apriori" probabilities are 8.09% for class 0 and 91.91% for class 1.

- Class 0 represents no default
- Class 1 represents default

The model has then calculated the conditional probabilities for each column. i.e

$$p(0) = p(ExtSource1|target = 0) * p(ExtSource2|target = 0) * \ldots p(target = 0)$$

The p(target = 0) represents the apriori probability.

Since the apriori probability is simply the proportion of target variable's levels this means that no default's apriori probability will be 91.91% while default's apriori probability is 8.09%.

The probability can also be calculated for the probability of no default as well:

i.e

$$p(1) = p(ExtSource1|target = 1) * p(ExtSource2|target = 1) * \ldots p(target = 1)$$

Afterwards, whichever probability is the greater for each observation will be used to make the prediction. For instance, if the probability of "No Default" is .78 and the probability of "Default" is .45, the algorithm will classify the observation as "No Default".

Make Predictions

```
tic() # Keep track of how long it took to make predictions
# Generate predictions on the train set using the naive bayes model
nb_train <- predict(nb_model,newdata = train_set,type = "raw")
# Generate predictions on the test set using the naive bayes model
nb_test <- predict(nb_model,newdata = test_set, type = "raw")
toc()
```

```
594.93 sec elapsed
```

Compute Evaluation Metrics

This section contains the train and test set metrics for the Naive Bayes Model.

Evaluation Train Metrics

```
# Generate metrics for the model's train predictions
mmetric(train_set$TARGET,nb_train,metrics_list)
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
82.5683113 37.2402019 86.5602855 19.6160463 93.9979271 25.6965786

       F12        AUC
90.1259189 0.6924399

$conf
      pred
target      1      0
     1   5017   8455
     0  20559 132413
```

The model seems to perform better than the regular Decision Tree.

The model did slightly better at discriminating between the two classes (default and no default) with an AUC of 0.6924. The weighted decision tree had an AUC of 0.6788.

In the context of Home Credit, this means that the Naive Bayes model is able to better distinguish between defaulting clients and non-defaulting clients in the dataset.

The Naive Bayes Model also performed better in some of the minority class metrics when compared to a decision tree. It should however be noted that the decision tree's performance was zero for the minority class. This means that any model that delivers non-zero results for the minority class will do better like the Weighted Decision Tree.

The Naive Bayes model does better in terms of precision and F1 Score when compared to the Weighted Decision Tree. The Weighted Decision Tree's precision score was 16% while Naive Baye's precision score was 19.62%. The F1 Score was also slightly better for Naive Bayes at 25.70% vs 24.63% for the Weighted Decision Tree.

This means that the Naive Bayes model is able to better balance recall and precision better than the Weighted Decision Tree for the negative class. The model's performance however for the positive class is better for recall and F1 score as well. For instance, the weighted decision tree's recall score was 75% while the Naive Bayes's score was 86.56%. The F1 score was also better at 90.13% for Naive Bayes compared to 83.75% for the Weighted Decision Tree. Naive Bayes does however come very close to the Weighted Decision Tree for the positive precision score as well. (Weighted Decision Tree Precision Score was 94.82% vs 94% for Naive Bayes Model.)

Accuracy is also higher overall, but as previously mentioned, not a very great metric due to the high class imbalance.

Hence, based on the train set, the Naive Bayes model seems comparable to Weighted Decision Tree. It also performs slightly better than the Weighted Decision Tree as well. The model however needs to be tested on the Test Set to fully gauge performance.

Evaluation Test Metrics

```r
# Generate metrics for the model's test predictions
mmetric(test_set$TARGET,nb_test,metrics_list)
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
82.5604217 36.6360644 86.6044326 19.4090117 93.9472160 25.3749250
       F12        AUC
90.1265140  0.6922606

$conf
      pred
target     1     0
     1  2115  3658
     0  8782 56777
```

This model’s test set metrics are very comparable to the train set metrics. Thus, this model is generalizing well to new data. This also indicates that the model is low on variance but moderately high on bias since some of the metrics for the negative class are low like precision which is at 19.41% or recall which is 36.64%.

Overall this model does not perform better then the Weighted Decision Tree which had more balanced metrics for the negative class. For instance, the Weighted Decision Tree's metrics vs the Naive Bayes Model are:

Recall (Negative):

- 53.44 - Weighted Decision Tree
- **36.64 - Naive Bayes Model**

Precision (Negative):

- 15.84 - Weighted Decision Tree
- **19.41 - Naive Bayes Model**

F1 (Negative):

- 24.44 - Weighted Decision Tree
- **25.37 - Naive Bayes Model**

AUC:

- 0.6787 - Weighted Decision Tree
- **0.6922 - Naive Bayes Model**

This Naive Bayes Model does have a slightly higher AUC score which indicates that it's better at discriminating between the positive and negative classes. However the Weighted Decision Tree has a much higher Recall at 53.44 vs 36.44 for the recall. Moreover all of the Weighted Decision Tree's metrics are slightly lower for Precision and F1.

Thus, the Weighted Decision Tree is still better since it does better in Recall and only slightly worse in Precision and F1.

Logistic Regression

This is the next model that will be built to predict default and no default.

Build Logistic Regression Model

```
tic() # Track how long model takes to train
logistic_model <- glm(TARGET ~ .  -House_Attribute_Low_Variance, data = train_set, family = binom
toc() # Track how long model takes to train
```

```
95.14 sec elapsed
```

This model took 1.34 minutes to train

Summary of Logistic Regression Model

```r
# Get a key highlight of this model
summary(logistic_model)
```

```
Call:
glm(formula = TARGET ~ . - House_Attribute_Low_Variance, family = binomial,
    data = train_set)

Coefficients: (1 not defined because of singularities)
                                                 Estimate Std. Error z value
(Intercept)                                     3.134e+01  3.979e+02   0.079
NAME_CONTRACT_TYPERevolving loans               1.361e-01  6.313e-02   2.156
CODE_GENDERM                                   -3.215e-01  2.500e-02 -12.861
FLAG_OWN_CARY                                   4.296e-01  4.822e-02   8.910
FLAG_OWN_REALTYY                               -1.849e-02  2.133e-02  -0.867
CNT_CHILDREN                                   -2.687e-02  1.572e-02  -1.709
AMT_INCOME_TOTAL                               -2.013e-07  2.027e-07  -0.993
AMT_CREDIT                                     -2.315e-06  1.566e-07 -14.781
AMT_ANNUITY                                    -1.178e-05  1.241e-06  -9.492
AMT_GOODS_PRICE                                 2.849e-06  1.786e-07  15.950
NAME_TYPE_SUITEFamily                           9.180e-02  9.499e-02   0.966
NAME_TYPE_SUITEGroup of people                 -2.511e-01  2.922e-01  -0.859
NAME_TYPE_SUITEOther_A                          3.015e-01  2.035e-01   1.482
NAME_TYPE_SUITEOther_B                         -7.490e-02  1.427e-01  -0.525
NAME_TYPE_SUITESpouse, partner                  1.490e-01  1.046e-01   1.424
NAME_TYPE_SUITEUnaccompanied                    3.780e-02  9.174e-02   0.412
NAME_INCOME_TYPECommercial associate           -1.040e+01  3.028e+02  -0.034
NAME_INCOME_TYPEMaternity leave                -1.448e+01  3.028e+02  -0.048
NAME_INCOME_TYPEPensioner                       6.707e-01  3.682e+02   0.002
NAME_INCOME_TYPEState servant                  -1.044e+01  3.028e+02  -0.034
NAME_INCOME_TYPEStudent                         8.120e-01  3.285e+02   0.002
NAME_INCOME_TYPEUnemployed                     -2.270e+00  3.682e+02  -0.006
NAME_INCOME_TYPEWorking                        -1.051e+01  3.028e+02  -0.035
NAME_EDUCATION_TYPEHigher education            -1.064e+01  6.131e+01  -0.174
NAME_EDUCATION_TYPEIncomplete higher           -1.068e+01  6.131e+01  -0.174
NAME_EDUCATION_TYPELower secondary             -1.093e+01  6.131e+01  -0.178
NAME_EDUCATION_TYPESecondary / secondary special -1.088e+01  6.131e+01  -0.177
NAME_FAMILY_STATUSMarried                       1.197e-01  3.055e-02   3.919
NAME_FAMILY_STATUSSeparated                    -3.702e-02  4.548e-02  -0.814
NAME_FAMILY_STATUSSingle / not married          2.943e-02  3.565e-02   0.825
NAME_FAMILY_STATUSUnknown                       1.003e+01  5.354e+02   0.019
NAME_FAMILY_STATUSWidow                         7.393e-02  5.344e-02   1.384
NAME_HOUSING_TYPEHouse / apartment              1.893e-01  1.584e-01   1.195
NAME_HOUSING_TYPEMunicipal apartment            5.068e-02  1.651e-01   0.307
NAME_HOUSING_TYPEOffice apartment               3.258e-01  1.929e-01   1.689
NAME_HOUSING_TYPERented apartment               6.233e-02  1.699e-01   0.367
NAME_HOUSING_TYPEWith parents                   1.556e-01  1.622e-01   0.960
```

REGION_POPULATION_RELATIVE	-2.310e+00	9.686e-01	-2.385
DAYS_BIRTH	2.708e-05	3.246e-06	8.343
DAYS_REGISTRATION	1.208e-05	2.972e-06	4.063
DAYS_ID_PUBLISH	4.594e-05	6.687e-06	6.870
OWN_CAR_AGE	-1.371e-02	6.217e-03	-2.205
FLAG_EMP_PHONE1	-1.049e+01	2.508e+02	-0.042
FLAG_WORK_PHONE1	-1.830e-01	2.485e-02	-7.366
FLAG_PHONE1	5.963e-02	2.306e-02	2.586
FLAG_EMAIL1	1.091e-02	4.336e-02	0.252
OCCUPATION_TYPECleaning staff	-2.983e-01	9.441e-02	-3.159
OCCUPATION_TYPECooking staff	-2.456e-01	8.846e-02	-2.776
OCCUPATION_TYPECore staff	-8.388e-02	7.567e-02	-1.109
OCCUPATION_TYPEDrivers	-2.220e-01	7.960e-02	-2.789
OCCUPATION_TYPEHigh skill tech staff	-3.399e-02	8.585e-02	-0.396
OCCUPATION_TYPEHR staff	-2.344e-01	2.460e-01	-0.953
OCCUPATION_TYPEIT staff	-1.472e-01	2.589e-01	-0.568
OCCUPATION_TYPELaborers	-2.399e-01	7.051e-02	-3.402
OCCUPATION_TYPELow-skill Laborers	-4.092e-01	1.075e-01	-3.808
OCCUPATION_TYPEManagers	-4.170e-02	8.020e-02	-0.520
OCCUPATION_TYPEMedicine staff	-9.308e-02	9.957e-02	-0.935
OCCUPATION_TYPEPrivate service staff	3.247e-02	1.315e-01	0.247
OCCUPATION_TYPERealty agents	3.999e-02	2.193e-01	0.182
OCCUPATION_TYPESales staff	-1.637e-01	7.188e-02	-2.278
OCCUPATION_TYPESecretaries	-3.399e-01	1.581e-01	-2.150
OCCUPATION_TYPESecurity staff	-2.682e-01	9.648e-02	-2.780
OCCUPATION_TYPEUnemployed	-1.405e-01	7.080e-02	-1.985
OCCUPATION_TYPEWaiters/barmen staff	-3.483e-01	1.327e-01	-2.625
CNT_FAM_MEMBERS	NA	NA	NA
REGION_RATING_CLIENT2	2.687e-01	1.606e-01	1.673
REGION_RATING_CLIENT3	2.843e-01	1.646e-01	1.727
REGION_RATING_CLIENT_W_CITY2	-4.194e-01	1.532e-01	-2.738
REGION_RATING_CLIENT_W_CITY3	-5.920e-01	1.589e-01	-3.727
WEEKDAY_APPR_PROCESS_STARTMONDAY	9.092e-02	3.281e-02	2.771
WEEKDAY_APPR_PROCESS_STARTSATURDAY	8.479e-02	3.668e-02	2.311
WEEKDAY_APPR_PROCESS_STARTSUNDAY	9.303e-02	4.711e-02	1.975
WEEKDAY_APPR_PROCESS_STARTTHURSDAY	3.830e-02	3.248e-02	1.179
WEEKDAY_APPR_PROCESS_STARTTUESDAY	1.496e-03	3.174e-02	0.047
WEEKDAY_APPR_PROCESS_STARTWEDNESDAY	2.872e-02	3.222e-02	0.891
HOUR_APPR_PROCESS_START	3.398e-04	3.065e-03	0.111
REG_REGION_NOT_WORK_REGION1	8.689e-02	4.604e-02	1.887
REG_CITY_NOT_LIVE_CITY1	-1.707e-01	4.673e-02	-3.652
REG_CITY_NOT_WORK_CITY1	2.445e-02	5.257e-02	0.465
LIVE_CITY_NOT_WORK_CITY1	-6.993e-02	5.091e-02	-1.373
ORGANIZATION_TYPEAgriculture	4.788e-01	2.479e-01	1.932
ORGANIZATION_TYPEBank	6.292e-01	2.573e-01	2.445
ORGANIZATION_TYPEBusiness Entity Type 1	4.788e-01	2.370e-01	2.021
ORGANIZATION_TYPEBusiness Entity Type 2	5.263e-01	2.331e-01	2.258
ORGANIZATION_TYPEBusiness Entity Type 3	3.611e-01	2.283e-01	1.582
ORGANIZATION_TYPECleaning	1.438e-01	3.526e-01	0.408
ORGANIZATION_TYPEConstruction	1.741e-01	2.346e-01	0.742
ORGANIZATION_TYPECulture	2.876e-01	3.585e-01	0.802

ORGANIZATION_TYPEElectricity	4.041e-01	2.855e-01	1.415
ORGANIZATION_TYPEEmergency	5.129e-01	3.318e-01	1.546
ORGANIZATION_TYPEGovernment	4.823e-01	2.337e-01	2.064
ORGANIZATION_TYPEHotel	5.499e-01	2.811e-01	1.956
ORGANIZATION_TYPEHousing	6.834e-01	2.505e-01	2.728
ORGANIZATION_TYPEIndustry: type 1	2.038e-01	2.657e-01	0.767
ORGANIZATION_TYPEIndustry: type 10	4.477e-01	5.467e-01	0.819
ORGANIZATION_TYPEIndustry: type 11	4.571e-01	2.474e-01	1.847
ORGANIZATION_TYPEIndustry: type 12	1.029e+00	4.541e-01	2.267
ORGANIZATION_TYPEIndustry: type 13	7.076e-01	5.474e-01	1.293
ORGANIZATION_TYPEIndustry: type 2	4.391e-01	3.172e-01	1.384
ORGANIZATION_TYPEIndustry: type 3	3.770e-01	2.418e-01	1.559
ORGANIZATION_TYPEIndustry: type 4	5.646e-01	2.810e-01	2.010
ORGANIZATION_TYPEIndustry: type 5	7.308e-01	3.177e-01	2.300
ORGANIZATION_TYPEIndustry: type 6	6.451e-01	5.742e-01	1.123
ORGANIZATION_TYPEIndustry: type 7	5.253e-01	2.690e-01	1.952
ORGANIZATION_TYPEIndustry: type 8	4.976e-01	1.084e+00	0.459
ORGANIZATION_TYPEIndustry: type 9	7.658e-01	2.504e-01	3.058
ORGANIZATION_TYPEInsurance	5.138e-01	3.371e-01	1.524
ORGANIZATION_TYPEKindergarten	5.187e-01	2.371e-01	2.187
ORGANIZATION_TYPELegal Services	4.015e-02	4.106e-01	0.098
ORGANIZATION_TYPEMedicine	5.427e-01	2.372e-01	2.288
ORGANIZATION_TYPEMilitary	1.008e+00	2.658e-01	3.791
ORGANIZATION_TYPEMobile	3.959e-01	3.551e-01	1.115
ORGANIZATION_TYPEOther	4.552e-01	2.312e-01	1.969
ORGANIZATION_TYPEPolice	7.118e-01	2.639e-01	2.697
ORGANIZATION_TYPEPostal	2.209e-01	2.485e-01	0.889
ORGANIZATION_TYPERealtor	-2.357e-01	3.420e-01	-0.689
ORGANIZATION_TYPEReligion	3.755e-01	6.561e-01	0.572
ORGANIZATION_TYPERestaurant	4.363e-01	2.508e-01	1.740
ORGANIZATION_TYPESchool	6.235e-01	2.358e-01	2.644
ORGANIZATION_TYPESecurity	3.773e-01	2.481e-01	1.521
ORGANIZATION_TYPESecurity Ministries	9.568e-01	2.777e-01	3.446
ORGANIZATION_TYPESelf-employed	2.824e-01	2.289e-01	1.234
ORGANIZATION_TYPEServices	3.865e-01	2.704e-01	1.429
ORGANIZATION_TYPETelecom	4.294e-01	3.213e-01	1.336
ORGANIZATION_TYPETrade: type 1	2.566e-02	3.244e-01	0.079
ORGANIZATION_TYPETrade: type 2	7.158e-01	2.590e-01	2.764
ORGANIZATION_TYPETrade: type 3	2.565e-01	2.401e-01	1.068
ORGANIZATION_TYPETrade: type 4	1.011e+00	7.929e-01	1.275
ORGANIZATION_TYPETrade: type 5	1.263e+00	1.073e+00	1.177
ORGANIZATION_TYPETrade: type 6	5.613e-01	3.333e-01	1.684
ORGANIZATION_TYPETrade: type 7	3.207e-01	2.341e-01	1.370
ORGANIZATION_TYPETransport: type 1	9.145e-01	5.229e-01	1.749
ORGANIZATION_TYPETransport: type 2	4.581e-01	2.519e-01	1.819
ORGANIZATION_TYPETransport: type 3	-2.658e-01	2.588e-01	-1.027
ORGANIZATION_TYPETransport: type 4	3.998e-01	2.380e-01	1.680
ORGANIZATION_TYPEUnemployed	-2.114e+01	3.268e+02	-0.065
ORGANIZATION_TYPEUniversity	6.295e-01	2.887e-01	2.180
EXT_SOURCE_1	6.011e-01	4.045e-02	14.859
EXT_SOURCE_2	2.051e+00	4.857e-02	42.223

```
EXT_SOURCE_3                                 2.157e+00  4.766e-02  45.270
APARTMENTS_MEDI                              2.573e-02  1.137e-01   0.226
YEARS_BUILD_MEDI                             2.769e-01  9.470e-02   2.924
COMMONAREA_MEDI                             -5.017e-02  1.405e-01  -0.357
ELEVATORS_MEDI                               1.307e-01  9.584e-02   1.364
ENTRANCES_MEDI                               2.236e-01  1.019e-01   2.194
FLOORSMAX_MEDI                               6.370e-02  1.018e-01   0.626
FLOORSMIN_MEDI                              -4.679e-02  6.836e-02  -0.684
LIVINGAPARTMENTS_MEDI                        5.569e-03  1.102e-01   0.051
LIVINGAREA_MEDI                             -3.872e-02  1.108e-01  -0.349
NONLIVINGAPARTMENTS_MEDI                     1.121e-01  2.171e-01   0.516
OBS_30_CNT_SOCIAL_CIRCLE                     3.566e-02  7.349e-02   0.485
DEF_30_CNT_SOCIAL_CIRCLE                    -1.405e-01  3.867e-02  -3.634
OBS_60_CNT_SOCIAL_CIRCLE                    -3.425e-02  7.419e-02  -0.462
DEF_60_CNT_SOCIAL_CIRCLE                    -6.438e-02  4.557e-02  -1.413
DAYS_LAST_PHONE_CHANGE                       6.850e-05  1.287e-05   5.323
FLAG_DOCUMENT_31                            -2.021e-01  5.475e-02  -3.691
FLAG_DOCUMENT_61                            -1.163e-01  7.068e-02  -1.646
FLAG_DOCUMENT_81                             4.263e-02  6.733e-02   0.633
AMT_REQ_CREDIT_BUREAU_HOUR                   6.412e-02  1.248e-01   0.514
AMT_REQ_CREDIT_BUREAU_DAY                   -1.696e-01  8.489e-02  -1.997
AMT_REQ_CREDIT_BUREAU_WEEK                   3.637e-02  5.013e-02   0.725
AMT_REQ_CREDIT_BUREAU_MON                    5.383e-03  1.341e-02   0.402
AMT_REQ_CREDIT_BUREAU_QRT                    6.420e-02  1.674e-02   3.836
AMT_REQ_CREDIT_BUREAU_YEAR                  -1.616e-02  5.339e-03  -3.027
                                            Pr(>|z|)
(Intercept)                                 0.937224
NAME_CONTRACT_TYPERevolving loans           0.031116 *
CODE_GENDERM                                 < 2e-16 ***
FLAG_OWN_CARY                                < 2e-16 ***
FLAG_OWN_REALTYY                            0.386067
CNT_CHILDREN                                0.087387 .
AMT_INCOME_TOTAL                            0.320806
AMT_CREDIT                                   < 2e-16 ***
AMT_ANNUITY                                  < 2e-16 ***
AMT_GOODS_PRICE                              < 2e-16 ***
NAME_TYPE_SUITEFamily                       0.333852
NAME_TYPE_SUITEGroup of people              0.390150
NAME_TYPE_SUITEOther_A                      0.138389
NAME_TYPE_SUITEOther_B                      0.599572
NAME_TYPE_SUITESpouse, partner              0.154570
NAME_TYPE_SUITEUnaccompanied                0.680283
NAME_INCOME_TYPECommercial associate        0.972605
NAME_INCOME_TYPEMaternity leave             0.961845
NAME_INCOME_TYPEPensioner                   0.998546
NAME_INCOME_TYPEState servant               0.972504
NAME_INCOME_TYPEStudent                     0.998028
NAME_INCOME_TYPEUnemployed                  0.995081
NAME_INCOME_TYPEWorking                     0.972309
NAME_EDUCATION_TYPEHigher education         0.862202
NAME_EDUCATION_TYPEIncomplete higher        0.861659
```

```
NAME_EDUCATION_TYPELower secondary              0.858543
NAME_EDUCATION_TYPESecondary / secondary special 0.859143
NAME_FAMILY_STATUSMarried                       8.90e-05 ***
NAME_FAMILY_STATUSSeparated                     0.415651
NAME_FAMILY_STATUSSingle / not married          0.409124
NAME_FAMILY_STATUSUnknown                       0.985059
NAME_FAMILY_STATUSWidow                         0.166505
NAME_HOUSING_TYPEHouse / apartment              0.232120
NAME_HOUSING_TYPEMunicipal apartment            0.758826
NAME_HOUSING_TYPEOffice apartment               0.091149 .
NAME_HOUSING_TYPERented apartment               0.713683
NAME_HOUSING_TYPEWith parents                   0.337175
REGION_POPULATION_RELATIVE                      0.017084 *
DAYS_BIRTH                                       < 2e-16 ***
DAYS_REGISTRATION                               4.85e-05 ***
DAYS_ID_PUBLISH                                 6.44e-12 ***
OWN_CAR_AGE                                     0.027470 *
FLAG_EMP_PHONE1                                 0.966637
FLAG_WORK_PHONE1                                1.76e-13 ***
FLAG_PHONE1                                     0.009722 **
FLAG_EMAIL1                                     0.801363
OCCUPATION_TYPECleaning staff                   0.001582 **
OCCUPATION_TYPECooking staff                    0.005499 **
OCCUPATION_TYPECore staff                       0.267633
OCCUPATION_TYPEDrivers                          0.005281 **
OCCUPATION_TYPEHigh skill tech staff            0.692154
OCCUPATION_TYPEHR staff                         0.340624
OCCUPATION_TYPEIT staff                         0.569854
OCCUPATION_TYPELaborers                         0.000668 ***
OCCUPATION_TYPELow-skill Laborers               0.000140 ***
OCCUPATION_TYPEManagers                         0.603086
OCCUPATION_TYPEMedicine staff                   0.349877
OCCUPATION_TYPEPrivate service staff            0.805005
OCCUPATION_TYPERealty agents                    0.855285
OCCUPATION_TYPESales staff                      0.022743 *
OCCUPATION_TYPESecretaries                      0.031551 *
OCCUPATION_TYPESecurity staff                   0.005438 **
OCCUPATION_TYPEUnemployed                       0.047145 *
OCCUPATION_TYPEWaiters/barmen staff             0.008674 **
CNT_FAM_MEMBERS                                       NA
REGION_RATING_CLIENT2                           0.094374 .
REGION_RATING_CLIENT3                           0.084198 .
REGION_RATING_CLIENT_W_CITY2                    0.006179 **
REGION_RATING_CLIENT_W_CITY3                    0.000194 ***
WEEKDAY_APPR_PROCESS_STARTMONDAY                0.005588 **
WEEKDAY_APPR_PROCESS_STARTSATURDAY              0.020811 *
WEEKDAY_APPR_PROCESS_STARTSUNDAY                0.048315 *
WEEKDAY_APPR_PROCESS_STARTTHURSDAY              0.238303
WEEKDAY_APPR_PROCESS_STARTTUESDAY               0.962395
WEEKDAY_APPR_PROCESS_STARTWEDNESDAY             0.372662
HOUR_APPR_PROCESS_START                         0.911735
```

```
REG_REGION_NOT_WORK_REGION1                    0.059112 .
REG_CITY_NOT_LIVE_CITY1                        0.000260 ***
REG_CITY_NOT_WORK_CITY1                        0.641857
LIVE_CITY_NOT_WORK_CITY1                       0.169608
ORGANIZATION_TYPEAgriculture                   0.053374 .
ORGANIZATION_TYPEBank                          0.014484 *
ORGANIZATION_TYPEBusiness Entity Type 1        0.043326 *
ORGANIZATION_TYPEBusiness Entity Type 2        0.023972 *
ORGANIZATION_TYPEBusiness Entity Type 3        0.113682
ORGANIZATION_TYPECleaning                      0.683369
ORGANIZATION_TYPEConstruction                  0.458040
ORGANIZATION_TYPECulture                       0.422339
ORGANIZATION_TYPEElectricity                   0.156976
ORGANIZATION_TYPEEmergency                     0.122158
ORGANIZATION_TYPEGovernment                    0.039011 *
ORGANIZATION_TYPEHotel                         0.050437 .
ORGANIZATION_TYPEHousing                       0.006379 **
ORGANIZATION_TYPEIndustry: type 1              0.442995
ORGANIZATION_TYPEIndustry: type 10             0.412902
ORGANIZATION_TYPEIndustry: type 11             0.064718 .
ORGANIZATION_TYPEIndustry: type 12             0.023379 *
ORGANIZATION_TYPEIndustry: type 13             0.196081
ORGANIZATION_TYPEIndustry: type 2              0.166300
ORGANIZATION_TYPEIndustry: type 3              0.118900
ORGANIZATION_TYPEIndustry: type 4              0.044482 *
ORGANIZATION_TYPEIndustry: type 5              0.021426 *
ORGANIZATION_TYPEIndustry: type 6              0.261245
ORGANIZATION_TYPEIndustry: type 7              0.050885 .
ORGANIZATION_TYPEIndustry: type 8              0.646159
ORGANIZATION_TYPEIndustry: type 9              0.002227 **
ORGANIZATION_TYPEInsurance                     0.127468
ORGANIZATION_TYPEKindergarten                  0.028710 *
ORGANIZATION_TYPELegal Services                0.922114
ORGANIZATION_TYPEMedicine                      0.022115 *
ORGANIZATION_TYPEMilitary                      0.000150 ***
ORGANIZATION_TYPEMobile                        0.264836
ORGANIZATION_TYPEOther                         0.048957 *
ORGANIZATION_TYPEPolice                        0.006988 **
ORGANIZATION_TYPEPostal                        0.374032
ORGANIZATION_TYPERealtor                       0.490810
ORGANIZATION_TYPEReligion                      0.567070
ORGANIZATION_TYPERestaurant                    0.081858 .
ORGANIZATION_TYPESchool                        0.008193 **
ORGANIZATION_TYPESecurity                      0.128359
ORGANIZATION_TYPESecurity Ministries           0.000569 ***
ORGANIZATION_TYPESelf-employed                 0.217331
ORGANIZATION_TYPEServices                      0.152990
ORGANIZATION_TYPETelecom                       0.181484
ORGANIZATION_TYPETrade: type 1                 0.936958
ORGANIZATION_TYPETrade: type 2                 0.005714 **
ORGANIZATION_TYPETrade: type 3                 0.285416
```

```
ORGANIZATION_TYPETrade: type 4                    0.202422
ORGANIZATION_TYPETrade: type 5                    0.239373
ORGANIZATION_TYPETrade: type 6                    0.092184 .
ORGANIZATION_TYPETrade: type 7                    0.170676
ORGANIZATION_TYPETransport: type 1                0.080276 .
ORGANIZATION_TYPETransport: type 2                0.068950 .
ORGANIZATION_TYPETransport: type 3                0.304427
ORGANIZATION_TYPETransport: type 4                0.092994 .
ORGANIZATION_TYPEUnemployed                       0.948437
ORGANIZATION_TYPEUniversity                       0.029241 *
EXT_SOURCE_1                                       < 2e-16 ***
EXT_SOURCE_2                                       < 2e-16 ***
EXT_SOURCE_3                                       < 2e-16 ***
APARTMENTS_MEDI                                   0.820940
YEARS_BUILD_MEDI                                  0.003452 **
COMMONAREA_MEDI                                   0.720954
ELEVATORS_MEDI                                    0.172618
ENTRANCES_MEDI                                    0.028207 *
FLOORSMAX_MEDI                                    0.531610
FLOORSMIN_MEDI                                    0.493677
LIVINGAPARTMENTS_MEDI                             0.959716
LIVINGAREA_MEDI                                   0.726844
NONLIVINGAPARTMENTS_MEDI                          0.605597
OBS_30_CNT_SOCIAL_CIRCLE                          0.627496
DEF_30_CNT_SOCIAL_CIRCLE                          0.000279 ***
OBS_60_CNT_SOCIAL_CIRCLE                          0.644315
DEF_60_CNT_SOCIAL_CIRCLE                          0.157721
DAYS_LAST_PHONE_CHANGE                            1.02e-07 ***
FLAG_DOCUMENT_31                                  0.000224 ***
FLAG_DOCUMENT_61                                  0.099828 .
FLAG_DOCUMENT_81                                  0.526578
AMT_REQ_CREDIT_BUREAU_HOUR                        0.607490
AMT_REQ_CREDIT_BUREAU_DAY                         0.045789 *
AMT_REQ_CREDIT_BUREAU_WEEK                        0.468184
AMT_REQ_CREDIT_BUREAU_MON                         0.688042
AMT_REQ_CREDIT_BUREAU_QRT                         0.000125 ***
AMT_REQ_CREDIT_BUREAU_YEAR                        0.002470 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93561  on 166443  degrees of freedom
Residual deviance: 83627  on 166281  degrees of freedom
AIC: 83953

Number of Fisher Scoring iterations: 12
```

This shows key information for the Logistic Regression model. The information however can be a lot, so I have isolated the coefficients. Afterwards I have organized it in descending order.

Order Coefficients in Descending Order

```
# Extract the coefficients from the logistic regression model
log_coefficients <- logistic_model[["coefficients"]]

# Reordering the coefficients in descending order with sort function
sorted_coefficients <- sort(log_coefficients, decreasing = TRUE)

# View the sorted coefficients
print(sorted_coefficients)
```

```
                              (Intercept)
                             3.133979e+01
              NAME_FAMILY_STATUSUnknown
                             1.002664e+01
                             EXT_SOURCE_3
                             2.157405e+00
                             EXT_SOURCE_2
                             2.050918e+00
           ORGANIZATION_TYPETrade: type 5
                             1.262775e+00
       ORGANIZATION_TYPEIndustry: type 12
                             1.029422e+00
           ORGANIZATION_TYPETrade: type 4
                             1.010649e+00
              ORGANIZATION_TYPEMilitary
                             1.007714e+00
     ORGANIZATION_TYPESecurity Ministries
                             9.568355e-01
       ORGANIZATION_TYPETransport: type 1
                             9.145273e-01
                 NAME_INCOME_TYPEStudent
                             8.119607e-01
        ORGANIZATION_TYPEIndustry: type 9
                             7.658051e-01
        ORGANIZATION_TYPEIndustry: type 5
                             7.308444e-01
           ORGANIZATION_TYPETrade: type 2
                             7.158432e-01
                 ORGANIZATION_TYPEPolice
                             7.117719e-01
       ORGANIZATION_TYPEIndustry: type 13
                             7.076323e-01
                ORGANIZATION_TYPEHousing
                             6.833523e-01
              NAME_INCOME_TYPEPensioner
                             6.707406e-01
        ORGANIZATION_TYPEIndustry: type 6
                             6.451178e-01
             ORGANIZATION_TYPEUniversity
```

```
                                   6.294795e-01
                         ORGANIZATION_TYPEBank
                                   6.292243e-01
                       ORGANIZATION_TYPESchool
                                   6.234617e-01
                                   EXT_SOURCE_1
                                   6.011137e-01
             ORGANIZATION_TYPEIndustry: type 4
                                   5.646363e-01
                ORGANIZATION_TYPETrade: type 6
                                   5.612789e-01
                        ORGANIZATION_TYPEHotel
                                   5.498949e-01
                     ORGANIZATION_TYPEMedicine
                                   5.427444e-01
       ORGANIZATION_TYPEBusiness Entity Type 2
                                   5.263464e-01
             ORGANIZATION_TYPEIndustry: type 7
                                   5.252763e-01
                 ORGANIZATION_TYPEKindergarten
                                   5.186699e-01
                    ORGANIZATION_TYPEInsurance
                                   5.137716e-01
                    ORGANIZATION_TYPEEmergency
                                   5.128511e-01
             ORGANIZATION_TYPEIndustry: type 8
                                   4.975787e-01
                   ORGANIZATION_TYPEGovernment
                                   4.822822e-01
                  ORGANIZATION_TYPEAgriculture
                                   4.788393e-01
       ORGANIZATION_TYPEBusiness Entity Type 1
                                   4.788344e-01
            ORGANIZATION_TYPETransport: type 2
                                   4.581172e-01
            ORGANIZATION_TYPEIndustry: type 11
                                   4.570859e-01
                        ORGANIZATION_TYPEOther
                                   4.552258e-01
            ORGANIZATION_TYPEIndustry: type 10
                                   4.476680e-01
             ORGANIZATION_TYPEIndustry: type 2
                                   4.390527e-01
                   ORGANIZATION_TYPERestaurant
                                   4.363307e-01
                                  FLAG_OWN_CARY
                                   4.296254e-01
                      ORGANIZATION_TYPETelecom
                                   4.293856e-01
                  ORGANIZATION_TYPEElectricity
                                   4.041278e-01
```

```
ORGANIZATION_TYPETransport: type 4
                         3.998273e-01
              ORGANIZATION_TYPEMobile
                         3.959350e-01
            ORGANIZATION_TYPEServices
                         3.864690e-01
            ORGANIZATION_TYPESecurity
                         3.773302e-01
   ORGANIZATION_TYPEIndustry: type 3
                         3.769959e-01
            ORGANIZATION_TYPEReligion
                         3.755459e-01
ORGANIZATION_TYPEBusiness Entity Type 3
                         3.610999e-01
   NAME_HOUSING_TYPEOffice apartment
                         3.258432e-01
      ORGANIZATION_TYPETrade: type 7
                         3.207349e-01
              NAME_TYPE_SUITEOther_A
                         3.014940e-01
             ORGANIZATION_TYPECulture
                         2.876115e-01
                REGION_RATING_CLIENT3
                         2.843011e-01
      ORGANIZATION_TYPESelf-employed
                         2.823642e-01
                     YEARS_BUILD_MEDI
                         2.769469e-01
                REGION_RATING_CLIENT2
                         2.686740e-01
      ORGANIZATION_TYPETrade: type 3
                         2.565199e-01
                       ENTRANCES_MEDI
                         2.236000e-01
              ORGANIZATION_TYPEPostal
                         2.208625e-01
   ORGANIZATION_TYPEIndustry: type 1
                         2.037935e-01
  NAME_HOUSING_TYPEHouse / apartment
                         1.892505e-01
        ORGANIZATION_TYPEConstruction
                         1.741060e-01
        NAME_HOUSING_TYPEWith parents
                         1.556335e-01
      NAME_TYPE_SUITESpouse, partner
                         1.489563e-01
            ORGANIZATION_TYPECleaning
                         1.438155e-01
   NAME_CONTRACT_TYPERevolving loans
                         1.360722e-01
                       ELEVATORS_MEDI
```

```
1.307084e-01
NAME_FAMILY_STATUSMarried
1.197066e-01
NONLIVINGAPARTMENTS_MEDI
1.121222e-01
WEEKDAY_APPR_PROCESS_STARTSUNDAY
9.303120e-02
NAME_TYPE_SUITEFamily
9.179562e-02
WEEKDAY_APPR_PROCESS_STARTMONDAY
9.091865e-02
REG_REGION_NOT_WORK_REGION1
8.689107e-02
WEEKDAY_APPR_PROCESS_STARTSATURDAY
8.478582e-02
NAME_FAMILY_STATUSWidow
7.392880e-02
AMT_REQ_CREDIT_BUREAU_QRT
6.420159e-02
AMT_REQ_CREDIT_BUREAU_HOUR
6.411864e-02
FLOORSMAX_MEDI
6.370416e-02
NAME_HOUSING_TYPERented apartment
6.233352e-02
FLAG_PHONE1
5.963349e-02
NAME_HOUSING_TYPEMunicipal apartment
5.067534e-02
FLAG_DOCUMENT_81
4.263295e-02
ORGANIZATION_TYPELegal Services
4.014624e-02
OCCUPATION_TYPERealty agents
3.998762e-02
WEEKDAY_APPR_PROCESS_STARTTHURSDAY
3.830268e-02
NAME_TYPE_SUITEUnaccompanied
3.780347e-02
AMT_REQ_CREDIT_BUREAU_WEEK
3.636759e-02
OBS_30_CNT_SOCIAL_CIRCLE
3.566124e-02
OCCUPATION_TYPEPrivate service staff
3.246987e-02
NAME_FAMILY_STATUSSingle / not married
2.943072e-02
WEEKDAY_APPR_PROCESS_STARTWEDNESDAY
2.872235e-02
APARTMENTS_MEDI
2.572706e-02
```

```
      ORGANIZATION_TYPETrade: type 1
                         2.566173e-02
             REG_CITY_NOT_WORK_CITY1
                         2.444897e-02
                          FLAG_EMAIL1
                         1.090753e-02
              LIVINGAPARTMENTS_MEDI
                         5.568573e-03
           AMT_REQ_CREDIT_BUREAU_MON
                         5.383470e-03
   WEEKDAY_APPR_PROCESS_STARTTUESDAY
                         1.496484e-03
             HOUR_APPR_PROCESS_START
                         3.397610e-04
              DAYS_LAST_PHONE_CHANGE
                         6.849524e-05
                      DAYS_ID_PUBLISH
                         4.594010e-05
                           DAYS_BIRTH
                         2.707775e-05
                    DAYS_REGISTRATION
                         1.207712e-05
                      AMT_GOODS_PRICE
                         2.849118e-06
                     AMT_INCOME_TOTAL
                        -2.012825e-07
                           AMT_CREDIT
                        -2.314917e-06
                          AMT_ANNUITY
                        -1.178047e-05
                          OWN_CAR_AGE
                        -1.370684e-02
          AMT_REQ_CREDIT_BUREAU_YEAR
                        -1.615962e-02
                     FLAG_OWN_REALTYY
                        -1.849254e-02
                         CNT_CHILDREN
                        -2.686796e-02
OCCUPATION_TYPEHigh skill tech staff
                        -3.398893e-02
            OBS_60_CNT_SOCIAL_CIRCLE
                        -3.425077e-02
         NAME_FAMILY_STATUSSeparated
                        -3.701819e-02
                      LIVINGAREA_MEDI
                        -3.871675e-02
             OCCUPATION_TYPEManagers
                        -4.170207e-02
                       FLOORSMIN_MEDI
                        -4.679154e-02
                      COMMONAREA_MEDI
```

-5.017434e-02
DEF_60_CNT_SOCIAL_CIRCLE
-6.438478e-02
LIVE_CITY_NOT_WORK_CITY1
-6.992652e-02
NAME_TYPE_SUITEOther_B
-7.489952e-02
OCCUPATION_TYPECore staff
-8.387762e-02
OCCUPATION_TYPEMedicine staff
-9.307985e-02
FLAG_DOCUMENT_61
-1.163201e-01
DEF_30_CNT_SOCIAL_CIRCLE
-1.405382e-01
OCCUPATION_TYPEUnemployed
-1.405430e-01
OCCUPATION_TYPEIT staff
-1.471506e-01
OCCUPATION_TYPESales staff
-1.637235e-01
AMT_REQ_CREDIT_BUREAU_DAY
-1.695623e-01
REG_CITY_NOT_LIVE_CITY1
-1.706854e-01
FLAG_WORK_PHONE1
-1.830460e-01
FLAG_DOCUMENT_31
-2.020829e-01
OCCUPATION_TYPEDrivers
-2.220234e-01
OCCUPATION_TYPEHR staff
-2.344237e-01
ORGANIZATION_TYPERealtor
-2.356509e-01
OCCUPATION_TYPELaborers
-2.399056e-01
OCCUPATION_TYPECooking staff
-2.455864e-01
NAME_TYPE_SUITEGroup of people
-2.510932e-01
ORGANIZATION_TYPETransport: type 3
-2.657598e-01
OCCUPATION_TYPESecurity staff
-2.682063e-01
OCCUPATION_TYPECleaning staff
-2.982612e-01
CODE_GENDERM
-3.215394e-01
OCCUPATION_TYPESecretaries
-3.398878e-01

```
                 OCCUPATION_TYPEWaiters/barmen staff
                                       -3.483105e-01
                   OCCUPATION_TYPELow-skill Laborers
                                       -4.092023e-01
                        REGION_RATING_CLIENT_W_CITY2
                                       -4.193870e-01
                        REGION_RATING_CLIENT_W_CITY3
                                       -5.920345e-01
                          NAME_INCOME_TYPEUnemployed
                                       -2.269898e+00
                          REGION_POPULATION_RELATIVE
                                       -2.310099e+00
                NAME_INCOME_TYPECommercial associate
                                       -1.039693e+01
                       NAME_INCOME_TYPEState servant
                                       -1.043559e+01
                                      FLAG_EMP_PHONE1
                                       -1.049159e+01
                             NAME_INCOME_TYPEWorking
                                       -1.050958e+01
                 NAME_EDUCATION_TYPEHigher education
                                       -1.064172e+01
                NAME_EDUCATION_TYPEIncomplete higher
                                       -1.068402e+01
    NAME_EDUCATION_TYPESecondary / secondary special
                                       -1.088036e+01
                  NAME_EDUCATION_TYPELower secondary
                                       -1.092728e+01
                     NAME_INCOME_TYPEMaternity leave
                                       -1.448357e+01
                         ORGANIZATION_TYPEUnemployed
                                       -2.113619e+01
```

`NAME_FAMILY_STATUSUnknown` and `Ext_Source_2` and `Ext_Source_3` are shown as important predictors in successful repayment. The coefficients are in log-odds which makes the exact interpretation difficult. However, since these coefficients are bigger then the other coefficients, we can tell that these predictors have the most influence. Moreover, the sign is positive which indicates that a one unit increase in these predictors increases the log-odds of successfully repaying a loan.

`NAME_FAMILY_STATUSUnknown` however is a factor but for anyone in the "Unknown" category, the log-odds of repayment increase. This does seem a little strange, one would think that one with a known family status might have a better chance of paying back a loan. It additionally is not statistically significant at a significance level of .05. Thus, the reliability of this third predictor is questionable. The other two predictors however are statistically significant at a significance level of .05

Overall though, 2 out of the 3 top predictors also match with the Weighted Decision Tree. `EXT_SOURCE_1` and `EXT_SOURCE_2` match in both Weighted Decision Tree and the Logistic Regression Model. (Weighted Decision Tree however predicted that `Occupation_Type` is the other important predictor while Logistic Regression is predicting "Unknown" Family Status.)

Make Predictions

Generate predictions for the train and test sets. It should be noted that Logistic Regression produces probabilities as predictions. Thus, a threshold has to be selected to convert the probabilities to class labels.

In the context, of Homecredit, there are 2 values in the `Target` Variable, "0" or "1". The Target Variable has already been converted to a factor with "0" as the second level. Thus, no changes are required for the Target Variable. However, a threshold must still be decided will be 0.9. This is because the classes are severely imbalanced which means that the model will be able to better predict the majority class. Thus, to mitigate this issue, a very high threshold of 0.9 will be selected. This will force the model to only classify someone as no default if their probability of paying the loan off is very high.

```
# Convert probabilities to binary class predictions (0 or 1)
threshold <- 0.9

# Train Predictions
lm_train_probs <- predict(logistic_model, newdata = train_set, type = "response") # Use Response t
lm_train <- factor(ifelse(lm_train_probs > threshold, "0", "1"), levels = c("1", "0")) # Convert t

# Test Predictions
lm_test_probs <- predict(logistic_model, newdata = test_set, type = "response") # Use Response to
lm_test <- factor(ifelse(lm_test_probs > threshold, "0", "1"), levels = c("1", "0")) # Convert th
```

Evaluate Model

Metrics will now be computed to evaluate the model's performance on the train and test metrics.

Evaluation Train Metrics

```
# Train Metrics
mmetric(train_set$TARGET, factor(lm_train,levels = c(1,0)),
metrics_list)
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
  75.34666   57.98694   76.87551   18.08920   95.40801   27.57598

       F12
  85.14499

$conf
      pred
target      1      0
     1   7812   5660
     0  35374 117598
```

The Logistic Regression Model's performance is comparable to the other models. For instance, it's Recall for the negative class is better at 58% vs the Weighted Decision Tree's recall which is only 53.44%. This is 4.41% better then the Weighted Decision Tree. Additionally, the precision (18.09%) and F1 (27.58%) scores are also higher when compared to a Weighted Decision Tree, (15.84%) and (24.44%) respectively.

Like the other models it does better on the positive class with fairly high metrics such as 95.41% for Precision. It also seems to do better then Naive Bayes. The model still however needs to be cross-validated to make sure that its performance is consistent.

Evaluation Test Metrics

```
# Test Metrics
mmetric(factor(test_set$TARGET,levels = c("1","0")),factor(lm_test,levels = c("1","0")),metrics_l
```

```
$res
      ACC      TPR1      TPR2 PRECISION1 PRECISION2       F11 
 75.22571  57.50909  76.78580   17.90819   95.35355  27.31162 

      F12 
 85.06827 

$conf
      pred
target     1     0
     1  3320  2453
     0 15219 50340
```

The Logistic Regression's model's performance on the test set is fairly consistent with its train set performance. There is a very minor drop in all the metrics when train and test sets are compared. This also means that the interpretations about the Weighted Decision Tree still hold. Since Logistic Regression's performance against the Weighted DecisionTree has been compared, we'll now compare this model's performance to Naive Bayes.

Logistic Regression does better in terms of Recall, and F1 score and is only slightly lower then Naive Bayes in terms of precision.

Recall (Negative):

- 36.64 - Naive Bayes Model
- **57.51 - Logistic Regression**

F1 (Negative):

- 27.31 - Naive Bayes Model
- **24.44 - Logistic Regression**

Precision (Negative): * 17.91 - Weighted Decision Tree * **19.41 - Naive Bayes Model** (Slightly higher then the Weighted Decision Tree)

Thus, Logistic Regression seems than Naive Bayes since it's performance with the positive class is comparable. Additionally it does better in all of the metrics for the negative class except for precision. This however is by a relatively low amount (1.5%)

Random Forest

Random Forest is an ensemble method that uses several decision trees to make classification predictions. It however is very computationally intensive and is prone to overfitting. Thus, to mitigate this issue, I will model a very simple Random Forest, since very complicated Random Forest Models overfit.

Pre Process the Data for Random Forest

The train dataset will be converted into a matrix and then all of the variables will be placed into "x" which is all the predictors. "y" will represent the "TARGET" variable.

```
# Convert categorical variables to dummy variables
train_data_clean <- model.matrix(TARGET ~ . - 1, data = cleaned_dataset) # Remove the target vari

# Set up independent and dependent variables
x <- as.data.frame(as.matrix(train_data_clean))
y <- as.factor(cleaned_dataset$TARGET)

# Reorder the levels in "y" to make "0" the second level
y <- factor(y,levels = c(1,0))
```

Partition the Dataset

The dataset will be partitioned into an 80/20 split. The train dataset will first be converted into a matrix and then partitioned with 80% of the data being used for training and 20% used for testing.

```
# Subset data for testing
set.seed(123) # Set seed for reproducibility

# Sample 20% of the data and assign to sample_index
sample_index <- sample(1:nrow(x), size = 0.2 * nrow(x))

# train set w/ all predictors and no target variable
x_sub_train <- x[sample_index, ] # Subset all predictors to x_sub
# target variable from test set
y_sub_train <- y[sample_index] # Subset target variable to y_sub

# test set w/ all predictors and no target variable
```

```
x_sub_test <- x[-sample_index,] # Subset all predictors to x_sub_test
y_sub_test <- y[-sample_index] # Subset all predictors to y_sub_test
```

The dataset has now been sucessfully partitioned into train and test sets with 80% in the train set and 20% in the test set.

Train the Model

```
tic()
set.seed(123)
# Train Random Forest on subset with reduced trees and features
rf_model <- randomForest(x_sub_train, y_sub_train, ntree = 50, mtry = log(ncol(x)), importance = 
toc()
```

```
431.79 sec elapsed
```

This model took 5.29 minutes to train which is relatively fast for a Random Forest. (This is however a farily simple model.)

Model Information

Model Summary

```
print(rf_model) # Print model summary
```

```

Call:
 randomForest(x = x_sub_train, y = y_sub_train, ntree = 50, mtry = log(ncol(x)),      importance 
= TRUE)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 5

        OOB estimate of  error rate: 8.01%
Confusion matrix:
   1     0  class.error
1  5  3797 0.9986849027
0 10 43743 0.0002285558
```

The overall out-of-bag (OOB) error is approximately 8%, which means the model misclassified around 8% of the observations. OOB error is an internal estimate of test error in Random Forests.

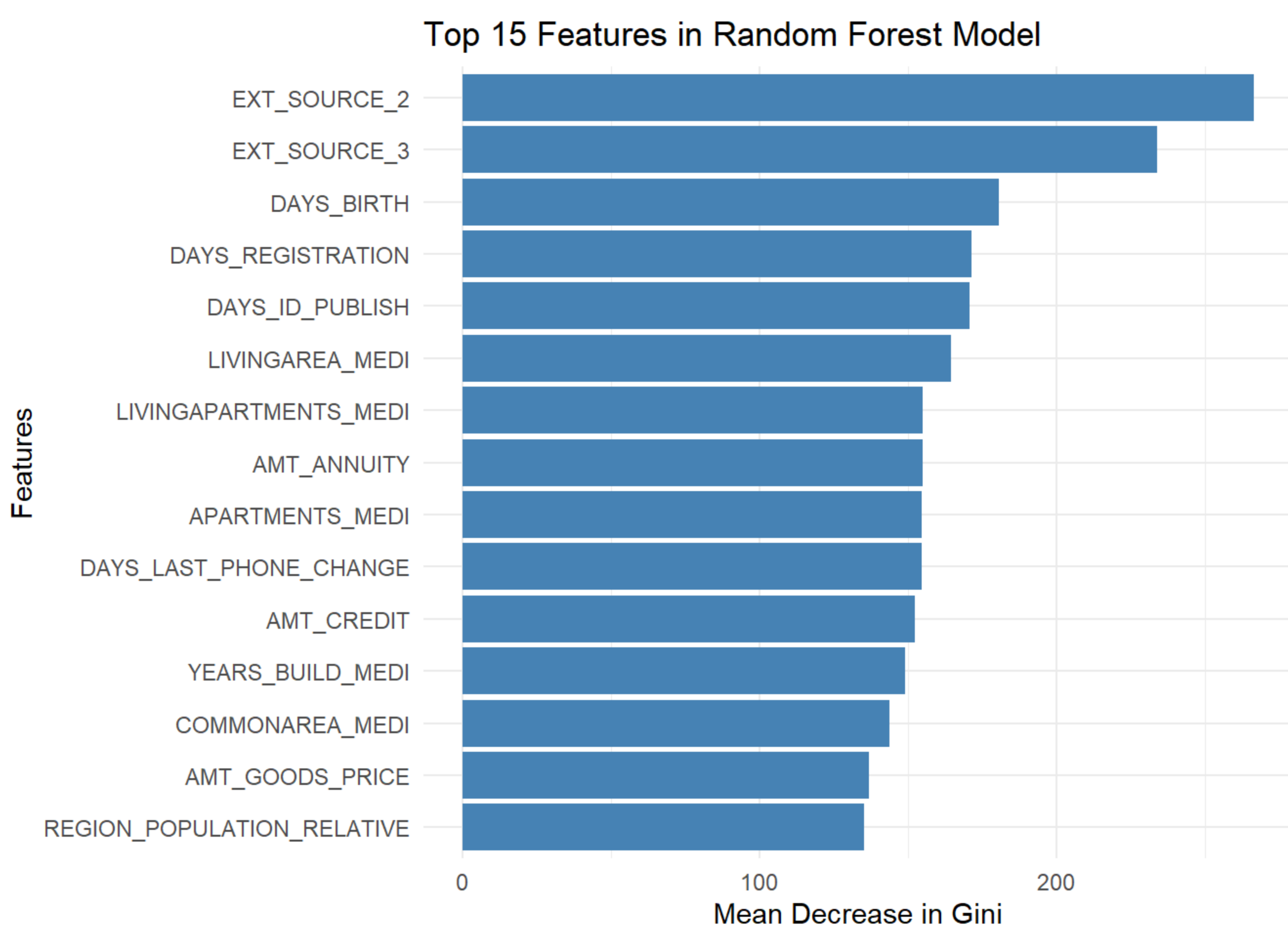
This code will extract the top 15 most important features from the XGboost Model. Thet model however has a very high error rate with the minority class (99.87%). In contrast, the majority class has a lower error rate that is less than 1%. This indicates that the class imbalance is affecting the model because Random Forest is only picking up the patterns for the majority class.

However to a get a full understanding, we'll still cross-validate Random Forest with the train and test sets.

Extract Important Features

```
# Extract feature importance and select top 15 features for better visualization
importance_df <- as.data.frame(importance(rf_model))
importance_df$Feature <- rownames(importance_df)
top_features <- importance_df %>%
  arrange(desc(MeanDecreaseGini)) %>%
  top_n(15, MeanDecreaseGini)

# Plot top 15 feature importance using ggplot2
ggplot(top_features, aes(x = reorder(Feature, MeanDecreaseGini), y = MeanDecreaseGini)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +  # Flip coordinates for horizontal bar chart
  labs(title = "Top 15 Features in Random Forest Model",
       x = "Features",
       y = "Mean Decrease in Gini") +
  theme_minimal()
```



This plot is based on the Gini Index which measures impurity or the probability of something being erroneously categorized. A predictor with a lower Gini Index indicates that the feature does a good job at reducing “uncertainty”.

The top 3 important features are `EXT_SOURCE_2`, `EXT_SOURCE_3` and `DAYS_BIRTH`. These predictors also very important for reducing impurity in the model as well.

`EXT_SOURCE_2` and `EXT_SOURCE_3` match with the other models. `DAYS_BIRTH` however is different and does not match up with the other models. For instance, the 3rd predictor for the Weighted Decision Tree was `OCCUPATION_TYPE`. The third predictor in Logistic Regression was `NAME_FAMILY_STATUSUnknown`

This indicates that at two of the top most important predictors are `EXT_SOURCE_2` and `EXT_SOURCE_3` since they appear in every model.

Generate Predictions

The code down below will generate predictions utilizing the Random Forest Model on the train and test sets. It will also help us get a full understanding of Random Forest’s performance. Random Forest’s model summary indicated that the results are not great for the minority class. Cross-Validation will however help us confirm or repudiate those results.

```
# Generate for the training data
predictions_train <- predict(rf_model, newdata = x_sub_train)

# Make predictions on the tesing data
predictions_test <- predict(rf_model, newdata = x_sub_test)
```

Predictions have been sucessfully generated for the train and test sets.

Generate Train Metrics

This code generates the metrics for the model’s performance on the train set.

```
# Train Metrics
mmetric(y_sub_train,predictions_train,metric = metrics_list)
```

```
$res
       ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
  96.62706   57.81168  100.00000  100.00000   96.46361   73.26667


       F12
  98.19998

$conf
      pred
target     1     0
```

```
     1  2198  1604
     0     0 43753
```

The model seems to do very well with high metrics for the positive class. classes. This model's metrics are actually higher than the rest of the models for both classes. The accuracy is even higher as well by 4.71%. (The majority classifier had an accuracy of 91.92%.)

These are the negative class metrics for the other 3 models:

Recall (Negative):

- 57.51 - Logistic Regression
- 36.64 - Naive Bayes Model
- **57.81 - Random Forest**

F1 (Negative):

- 24.44 - Weighted Decision Tree
- 25.37 - Naive Bayes Model
- **100 - Random Forest** (Much Higher)

Precision (Negative): * 17.91 - Weighted Decision Tree * 19.41 - Naive Bayes Model * **73.27 - Random Forest** (Much Higher)

The above metrics show that Random Forest is doing much better with the negative class when compared to the other models. This is spite of the high error rate for the minority class from the model summary.

However this great model performance must be cross-validated to ensure that the model is not overfitting.

Generate Test Metrics

This code generates the metrics for the model's performance on the test set.

```
# Test Metrics
mmetric(y_sub_test,predictions_test,metric = metrics_list)
```

```
$res
      ACC       TPR1       TPR2 PRECISION1 PRECISION2        F11
 91.88155    0.00000  100.00000    0.00000   91.88155    0.00000

      F12
 95.76903

$conf
      pred
target      1      0
     1      0  15443
     0      0 174778
```

The model maintains comparable performance, however it over-fits very severely for the negative class (default). For instance, Recall is 0.0064%, and F1 Score is 0.01295%. This is nearly a 100% drop in all of the metrics. The precision is 100% for the negative class, however this is not very informative. (Precision is only 100% because the model made only 1 prediction of a negative instance and it got it right. Precision is a measure of how many instances are actually negative out of all the predicted instances. So if the model makes one predicted instance and gets it right, then the precision will be 100%.)

This indicates that the Random Forest will need more hyper-parameter tuning to increase generalization to new data. I made the model fairly simple to avoid overfitting, but this does not seem to be very effective. It would appear that other parameters like regularization will need to be included, to improve Random Forest's performance.

Finally, it appears that the model's summary was correct, which cross validation has further proved. The model struggles to classify the minority class.

Final Thoughts

Overall this was an interesting experience with building all of these models. The key takeaways I have learned are that the predictors may not be the same across all of the models. However, there will be a few predictors that appear commonly across all the models. We can take those predictors and use them to make inferences about what the biggest associations are with successful repayment.

Additionally techniques like weights help with improving model performance along with using some simpler models like Logistic Regression or Naive Bayes to get some insights.

However Class Imbalance still remains an issue that must be addressed through other techniques like under-sampling, etc. I will be experimenting more on these techniques with my group.