Home Credit Default Risk: Modelling

Nikita Muddapati

Contents

Data Setup

```
pacman::p_load(tidyverse,skimr,janitor,knitr,caret,rminer,mice,dbscan,tictoc,dplyr,ranger,psy

library(xgboost)
library(Matrix)
library(pROC)   # to plot ROC-AUC
library(e1071)  # for svm
library(GGally) # to plot with ggpairs()
library(doParallel)  # for training xgboost using parallel processing
library(MLmetrics)

library(readxl)
library(themis)

library(knitr)

library(rmarkdown)

#library(xfun)

#install.packages("tinytex")   #to render to pdf
#tinytex::install_tinytex()

#library(tinytex)
```

```
# load cleaned application_train and application_test data from EDA HW (with removed columns

train_clean <- read_csv("C:/Users/nikit/Downloads/Capstone Project/train_clean.csv")
test_clean <- read_csv("C:/Users/nikit/Downloads/Capstone Project/test_clean.csv")

head(train_clean)  #first 6 rows
```

```
# A tibble: 6 x 64
  SK_ID_CURR TARGET NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY
       <dbl>  <dbl> <chr>              <chr>       <chr>        <chr>
1     100002      1 Cash loans         M           N            Y
2     100003      0 Cash loans         F           N            N
3     100006      0 Cash loans         F           N            Y
4     100007      0 Cash loans         M           N            Y
5     100008      0 Cash loans         M           N            Y
```

```
6     100011      0 Cash loans          F           N             Y
# i 58 more variables: CNT_CHILDREN <dbl>, AMT_INCOME_TOTAL <dbl>,
#   AMT_CREDIT <dbl>, AMT_ANNUITY <dbl>, AMT_GOODS_PRICE <dbl>,
#   NAME_TYPE_SUITE <chr>, NAME_INCOME_TYPE <chr>, NAME_EDUCATION_TYPE <chr>,
#   NAME_FAMILY_STATUS <chr>, NAME_HOUSING_TYPE <chr>,
#   REGION_POPULATION_RELATIVE <dbl>, DAYS_BIRTH <dbl>,
#   DAYS_REGISTRATION <dbl>, DAYS_ID_PUBLISH <dbl>, OWN_CAR_AGE <dbl>,
#   FLAG_EMP_PHONE <dbl>, FLAG_WORK_PHONE <dbl>, FLAG_PHONE <dbl>, ...
```

```r
dim(train_clean)  #shape of data
```

```
[1] 237776     64
```

Clean Train Data

```r
# remove identifier
train_clean <- train_clean %>% select(-SK_ID_CURR)


# factor target
#train_clean$TARGET <- factor(train_clean$TARGET, levels = c(0, 1))


# Modify all character variables into factors
tr_clean <- train_clean %>%
            mutate_if(is.character, as.factor)


# Store all numeric variables which should be factors in a vector

num_cat_values <- c("TARGET","FLAG_EMP_PHONE","FLAG_WORK_PHONE","FLAG_EMAIL","FLAG_PHONE",'RE

# transform the vector of num. columns to factors
tr_clean <- train_clean %>%
  mutate(across(all_of(num_cat_values), as.factor))


# structure of target
str(tr_clean$TARGET)
```

```
 Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
```

```r
# summary of cleaned dataset
summary(tr_clean)
```

```
 TARGET     NAME_CONTRACT_TYPE CODE_GENDER        FLAG_OWN_CAR
 0:218531   Length:237776      Length:237776      Length:237776
 1: 19245   Class :character   Class :character   Class :character
            Mode  :character   Mode  :character   Mode  :character


 FLAG_OWN_REALTY     CNT_CHILDREN    AMT_INCOME_TOTAL   AMT_CREDIT
 Length:237776      Min.   :0.0000   Min.   : 25650   Min.   :  45000
 Class :character   1st Qu.:0.0000   1st Qu.:108000   1st Qu.: 270000
 Mode  :character   Median :0.0000   Median :135000   Median : 491031
                    Mean   :0.3625   Mean   :148333   Mean   : 567430
                    3rd Qu.:1.0000   3rd Qu.:180000   3rd Qu.: 781695
                    Max.   :3.0000   Max.   :299700   Max.   :3860019
  AMT_ANNUITY     AMT_GOODS_PRICE   NAME_TYPE_SUITE    NAME_INCOME_TYPE
 Min.   :  1616   Min.   :  40500   Length:237776      Length:237776
 1st Qu.: 16006   1st Qu.: 229500   Class :character   Class :character
 Median : 23837   Median : 450000   Mode  :character   Mode  :character
 Mean   : 25678   Mean   : 509126
 3rd Qu.: 32603   3rd Qu.: 675000
 Max.   :225000   Max.   :3555000
 NAME_EDUCATION_TYPE NAME_FAMILY_STATUS NAME_HOUSING_TYPE
 Length:237776       Length:237776      Length:237776
 Class :character    Class :character   Class :character
 Mode  :character    Mode  :character   Mode  :character


 REGION_POPULATION_RELATIVE   DAYS_BIRTH    DAYS_REGISTRATION DAYS_ID_PUBLISH
 Min.   :0.00029            Min.   : 7673   Min.   :    0     Min.   :   0
 1st Qu.:0.01001            1st Qu.:12439   1st Qu.: 2108     1st Qu.:1721
 Median :0.01885            Median :15968   Median : 4601     Median :3262
 Mean   :0.02044            Mean   :16187   Mean   : 5097     Mean   :2996
 3rd Qu.:0.02639            3rd Qu.:19961   3rd Qu.: 7649     3rd Qu.:4299
 Max.   :0.07251            Max.   :25201   Max.   :24672     Max.   :7197
  OWN_CAR_AGE   FLAG_EMP_PHONE FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL
 Min.   : 0.0   0: 47432       0:190061        0:170246   0:225462
```

```
1st Qu.: 0.0   1:190344       1: 47715        1: 67530   1: 12314
Median : 0.0
Mean   : 1.4
3rd Qu.: 0.0
Max.   :12.0
OCCUPATION_TYPE    CNT_FAM_MEMBERS REGION_RATING_CLIENT
Length:237776      Min.   :1.00    1: 21787
Class :character   1st Qu.:2.00    2:179447
Mode  :character   Median :2.00    3: 36542
                   Mean   :2.08
                   3rd Qu.:2.00
                   Max.   :4.00
REGION_RATING_CLIENT_W_CITY WEEKDAY_APPR_PROCESS_START HOUR_APPR_PROCESS_START
1: 23152                    Length:237776              Min.   : 0.00
2:181331                    Class :character           1st Qu.:10.00
3: 33293                    Mode  :character           Median :12.00
                                                       Mean   :12.09
                                                       3rd Qu.:14.00
                                                       Max.   :23.00
REG_REGION_NOT_WORK_REGION REG_CITY_NOT_LIVE_CITY REG_CITY_NOT_WORK_CITY
0:227280                   0:219288               0:185117
1: 10496                   1: 18488               1: 52659



LIVE_CITY_NOT_WORK_CITY ORGANIZATION_TYPE   EXT_SOURCE_1
0:197449                Length:237776      Min.   :0.0000
1: 40327                Class :character   1st Qu.:0.0000
                        Mode  :character   Median :0.0000
                                           Mean   :0.2163
                                           3rd Qu.:0.4528
                                           Max.   :0.9516
 EXT_SOURCE_2        EXT_SOURCE_3       APARTMENTS_MEDI   YEARS_BUILD_MEDI
Min.   :0.0000001   Min.   :0.0005273   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.3859750   1st Qu.:0.3706496   1st Qu.:0.03440   1st Qu.:0.6578
Median :0.5627356   Median :0.5388627   Median :0.07290   Median :0.7048
Mean   :0.5111018   Mean   :0.5118501   Mean   :0.09919   Mean   :0.7177
3rd Qu.:0.6612173   3rd Qu.:0.6690567   3rd Qu.:0.12280   3rd Qu.:0.7920
Max.   :0.8549997   Max.   :0.8960095   Max.   :1.00000   Max.   :1.0000
COMMONAREA_MEDI   ELEVATORS_MEDI    ENTRANCES_MEDI   FLOORSMAX_MEDI
Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.00590   1st Qu.:0.00000   1st Qu.:0.0345   1st Qu.:0.1667
```

```
Median :0.01620   Median :0.00000   Median :0.1034   Median :0.1667
Mean   :0.03946   Mean   :0.07234   Mean   :0.1213   Mean   :0.2130
3rd Qu.:0.04580   3rd Qu.:0.12000   3rd Qu.:0.1379   3rd Qu.:0.3333
Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
FLOORSMIN_MEDI   LIVINGAPARTMENTS_MEDI LIVINGAREA_MEDI
Min.   :0.0000   Min.   :0.00000       Min.   :0.0000
1st Qu.:0.0833   1st Qu.:0.02740       1st Qu.:0.0388
Median :0.2083   Median :0.06070       Median :0.0677
Mean   :0.2207   Mean   :0.08435       Mean   :0.1003
3rd Qu.:0.3750   3rd Qu.:0.10180       3rd Qu.:0.1209
Max.   :1.0000   Max.   :1.00000       Max.   :1.0000
NONLIVINGAPARTMENTS_MEDI OBS_30_CNT_SOCIAL_CIRCLE DEF_30_CNT_SOCIAL_CIRCLE
Min.   :0.00000          Min.   : 0.000           Min.   :0.0000
1st Qu.:0.00000          1st Qu.: 0.000           1st Qu.:0.0000
Median :0.00000          Median : 0.000           Median :0.0000
Mean   :0.00639          Mean   : 1.322           Mean   :0.1432
3rd Qu.:0.00000          3rd Qu.: 2.000           3rd Qu.:0.0000
Max.   :1.00000          Max.   :10.000           Max.   :6.0000
OBS_60_CNT_SOCIAL_CIRCLE DEF_60_CNT_SOCIAL_CIRCLE DAYS_LAST_PHONE_CHANGE
Min.   : 0.000           Min.   :0.0000           Min.   :   0.0
1st Qu.: 0.000           1st Qu.:0.0000           1st Qu.: 266.0
Median : 0.000           Median :0.0000           Median : 741.0
Mean   : 1.306           Mean   :0.1005           Mean   : 949.7
3rd Qu.: 2.000           3rd Qu.:0.0000           3rd Qu.:1554.0
Max.   :10.000           Max.   :6.0000           Max.   :4292.0
FLAG_DOCUMENT_3 FLAG_DOCUMENT_6 FLAG_DOCUMENT_8 AMT_REQ_CREDIT_BUREAU_HOUR
0: 66846        0:214506        0:222941        Min.   :0.000000
1:170930        1: 23270        1: 14835        1st Qu.:0.000000
                                                Median :0.000000
                                                Mean   :0.005451
                                                3rd Qu.:0.000000
                                                Max.   :4.000000
AMT_REQ_CREDIT_BUREAU_DAY AMT_REQ_CREDIT_BUREAU_WEEK AMT_REQ_CREDIT_BUREAU_MON
Min.   :0.000000          Min.   :0.00000            Min.   : 0.0000
1st Qu.:0.000000          1st Qu.:0.00000            1st Qu.: 0.0000
Median :0.000000          Median :0.00000            Median : 0.0000
Mean   :0.006157          Mean   :0.02947            Mean   : 0.2221
3rd Qu.:0.000000          3rd Qu.:0.00000            3rd Qu.: 0.0000
Max.   :9.000000          Max.   :8.00000            Max.   :24.0000
AMT_REQ_CREDIT_BUREAU_QRT AMT_REQ_CREDIT_BUREAU_YEAR
Min.   : 0.00             Min.   : 0.00
1st Qu.: 0.00             1st Qu.: 1.00
Median : 0.00             Median : 1.00
```

```
 Mean   : 0.23             Mean   : 1.78
 3rd Qu.: 0.00             3rd Qu.: 3.00
 Max.   :19.00             Max.   :25.00
 House_Attribute_Low_Variance
 Min.   :-2.77162
 1st Qu.:-0.52169
 Median :-0.52169
 Mean   :-0.00629
 3rd Qu.:-0.34387
 Max.   :53.15290
```

Clean Test Data

```
# Modify all character variables into factors
te_clean <- test_clean %>%
            mutate_if(is.character, as.factor)

# Store all numeric variables which should be factors in a vector

num_cat_values2 <- c("FLAG_EMP_PHONE","FLAG_WORK_PHONE","FLAG_EMAIL","FLAG_PHONE",'REG_REGIO

# transform the vector of num. columns to factors

te_clean <- test_clean %>%
  mutate(across(all_of(num_cat_values2), as.factor))


# summary of cleaned dataset

summary(te_clean)
```

```
   SK_ID_CURR     NAME_CONTRACT_TYPE CODE_GENDER        FLAG_OWN_CAR
 Min.   :100001   Length:37740       Length:37740       Length:37740
 1st Qu.:186702   Class :character   Class :character   Class :character
 Median :275020   Mode  :character   Mode  :character   Mode  :character
 Mean   :274808
 3rd Qu.:362886
 Max.   :450458
 FLAG_OWN_REALTY     CNT_CHILDREN    AMT_INCOME_TOTAL   AMT_CREDIT
 Length:37740       Min.   :0.0000   Min.   : 26942   Min.   :  45000
```

```
Class :character   1st Qu.:0.0000   1st Qu.:112500   1st Qu.: 248760
Mode  :character   Median :0.0000   Median :153000   Median : 414612
                   Mean   :0.3456   Mean   :155613   Mean   : 480048
                   3rd Qu.:1.0000   3rd Qu.:202500   3rd Qu.: 610484
                   Max.   :3.0000   Max.   :299250   Max.   :2245500
 AMT_ANNUITY     AMT_GOODS_PRICE   NAME_TYPE_SUITE    NAME_INCOME_TYPE
Min.   :  2295   Min.   :  45000   Length:37740       Length:37740
1st Qu.: 17262   1st Qu.: 225000   Class :character   Class :character
Median : 24926   Median : 360000   Mode  :character   Mode  :character
Mean   : 27570   Mean   : 428636
3rd Qu.: 34826   3rd Qu.: 540000
Max.   :177827   Max.   :2245500
NAME_EDUCATION_TYPE NAME_FAMILY_STATUS NAME_HOUSING_TYPE
Length:37740        Length:37740       Length:37740
Class :character    Class :character   Class :character
Mode  :character    Mode  :character   Mode  :character



REGION_POPULATION_RELATIVE   DAYS_BIRTH    DAYS_REGISTRATION DAYS_ID_PUBLISH
Min.   :0.000253           Min.   : 7338   Min.   :    0     Min.   :   0
1st Qu.:0.010006           1st Qu.:12503   1st Qu.: 2004     1st Qu.:1711
Median :0.018850           Median :16004   Median : 4622     Median :3249
Mean   :0.020619           Mean   :16211   Mean   : 5092     Mean   :3058
3rd Qu.:0.028663           3rd Qu.:19942   3rd Qu.: 7659     3rd Qu.:4448
Max.   :0.072508           Max.   :25195   Max.   :23722     Max.   :6348
 OWN_CAR_AGE     FLAG_EMP_PHONE FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL
Min.   : 0.000   0: 7997        0:30036         0:27716    0:31937
1st Qu.: 0.000   1:29743        1: 7704         1:10024    1: 5803
Median : 0.000
Mean   : 1.403
3rd Qu.: 0.000
Max.   :12.000
OCCUPATION_TYPE    CNT_FAM_MEMBERS REGION_RATING_CLIENT
Length:37740       Min.   :1.000   1: 3787
Class :character   1st Qu.:2.000   2:28156
Mode  :character   Median :2.000   3: 5797
                   Mean   :2.077
                   3rd Qu.:2.000
                   Max.   :4.000
REGION_RATING_CLIENT_W_CITY WEEKDAY_APPR_PROCESS_START HOUR_APPR_PROCESS_START
-1:    1                    Length:37740               Min.   : 0.00
1 : 4059                    Class :character           1st Qu.:10.00
```

```
2 :28464                    Mode  :character           Median :12.00
3 : 5216                                               Mean   :12.04
                                                       3rd Qu.:14.00
                                                       Max.   :23.00
REG_REGION_NOT_WORK_REGION REG_CITY_NOT_LIVE_CITY REG_CITY_NOT_WORK_CITY
0:35988                    0:34829                0:29659
1: 1752                    1: 2911                1: 8081




LIVE_CITY_NOT_WORK_CITY ORGANIZATION_TYPE   EXT_SOURCE_1
0:31586                 Length:37740        Min.   :0.0000
1: 6154                 Class :character    1st Qu.:0.0000
                        Mode  :character    Median :0.2366
                                            Mean   :0.2853
                                            3rd Qu.:0.5464
                                            Max.   :0.9391
 EXT_SOURCE_2        EXT_SOURCE_3       APARTMENTS_MEDI  YEARS_BUILD_MEDI
Min.   :0.0000081   Min.   :0.0005273   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.4039882   1st Qu.:0.3656165   1st Qu.:0.0416   1st Qu.:0.6847
Median :0.5556799   Median :0.5208976   Median :0.0833   Median :0.7048
Mean   :0.5149308   Mean   :0.5020387   Mean   :0.1109   Mean   :0.7486
3rd Qu.:0.6555860   3rd Qu.:0.6545293   3rd Qu.:0.1374   3rd Qu.:0.8390
Max.   :0.8549997   Max.   :0.8825303   Max.   :1.0000   Max.   :1.0000
COMMONAREA_MEDI   ELEVATORS_MEDI    ENTRANCES_MEDI   FLOORSMAX_MEDI
Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.00760   1st Qu.:0.00000   1st Qu.:0.0690   1st Qu.:0.1667
Median :0.02200   Median :0.00000   Median :0.1379   Median :0.1667
Mean   :0.05325   Mean   :0.08353   Mean   :0.1438   Mean   :0.2270
3rd Qu.:0.05750   3rd Qu.:0.12000   3rd Qu.:0.2069   3rd Qu.:0.3333
Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
FLOORSMIN_MEDI   LIVINGAPARTMENTS_MEDI LIVINGAREA_MEDI
Min.   :0.0000   Min.   :0.00000       Min.   :0.0000
1st Qu.:0.0833   1st Qu.:0.03850       1st Qu.:0.0488
Median :0.2083   Median :0.06670       Median :0.0771
Mean   :0.2255   Mean   :0.09857       Mean   :0.1115
3rd Qu.:0.3750   3rd Qu.:0.12310       3rd Qu.:0.1359
Max.   :1.0000   Max.   :1.00000       Max.   :1.0000
NONLIVINGAPARTMENTS_MEDI OBS_30_CNT_SOCIAL_CIRCLE DEF_30_CNT_SOCIAL_CIRCLE
Min.   :0.000000         Min.   : 0.000           Min.   :0.0000
1st Qu.:0.000000         1st Qu.: 0.000           1st Qu.:0.0000
Median :0.000000         Median : 0.000           Median :0.0000
```

```
Mean   :0.008813         Mean   : 1.332           Mean   :0.1427
3rd Qu.:0.003900         3rd Qu.: 2.000           3rd Qu.:0.0000
Max.   :1.000000         Max.   :10.000           Max.   :6.0000
OBS_60_CNT_SOCIAL_CIRCLE DEF_60_CNT_SOCIAL_CIRCLE DAYS_LAST_PHONE_CHANGE
Min.   : 0.000           Min.   :0.0000           Min.   :   0
1st Qu.: 0.000           1st Qu.:0.0000           1st Qu.: 354
Median : 0.000           Median :0.0000           Median : 839
Mean   : 1.321           Mean   :0.1015           Mean   :1062
3rd Qu.: 2.000           3rd Qu.:0.0000           3rd Qu.:1755
Max.   :10.000           Max.   :5.0000           Max.   :4361
FLAG_DOCUMENT_3 FLAG_DOCUMENT_6 FLAG_DOCUMENT_8 AMT_REQ_CREDIT_BUREAU_HOUR
0: 7685         0:34055         0:35182         Min.   :0.000000
1:30055         1: 3685         1: 2558         1st Qu.:0.000000
                                                Median :0.000000
                                                Mean   :0.001696
                                                3rd Qu.:0.000000
                                                Max.   :1.000000
AMT_REQ_CREDIT_BUREAU_DAY AMT_REQ_CREDIT_BUREAU_WEEK AMT_REQ_CREDIT_BUREAU_MON
Min.   :0.000000          Min.   :0.000000           Min.   :0.000000
1st Qu.:0.000000          1st Qu.:0.000000           1st Qu.:0.000000
Median :0.000000          Median :0.000000           Median :0.000000
Mean   :0.001431          Mean   :0.002544           Mean   :0.008161
3rd Qu.:0.000000          3rd Qu.:0.000000           3rd Qu.:0.000000
Max.   :2.000000          Max.   :2.000000           Max.   :6.000000
AMT_REQ_CREDIT_BUREAU_QRT AMT_REQ_CREDIT_BUREAU_YEAR
Min.   :0.0000            Min.   : 0.000
1st Qu.:0.0000            1st Qu.: 1.000
Median :0.0000            Median : 2.000
Mean   :0.4734            Mean   : 1.999
3rd Qu.:1.0000            3rd Qu.: 3.000
Max.   :7.0000            Max.   :17.000
House_Attribute_Low_Variance
Min.   :-2.78776
1st Qu.:-0.53249
Median :-0.53249
Mean   :-0.00739
3rd Qu.:-0.29941
Max.   :48.18664
```

Class Imbalance

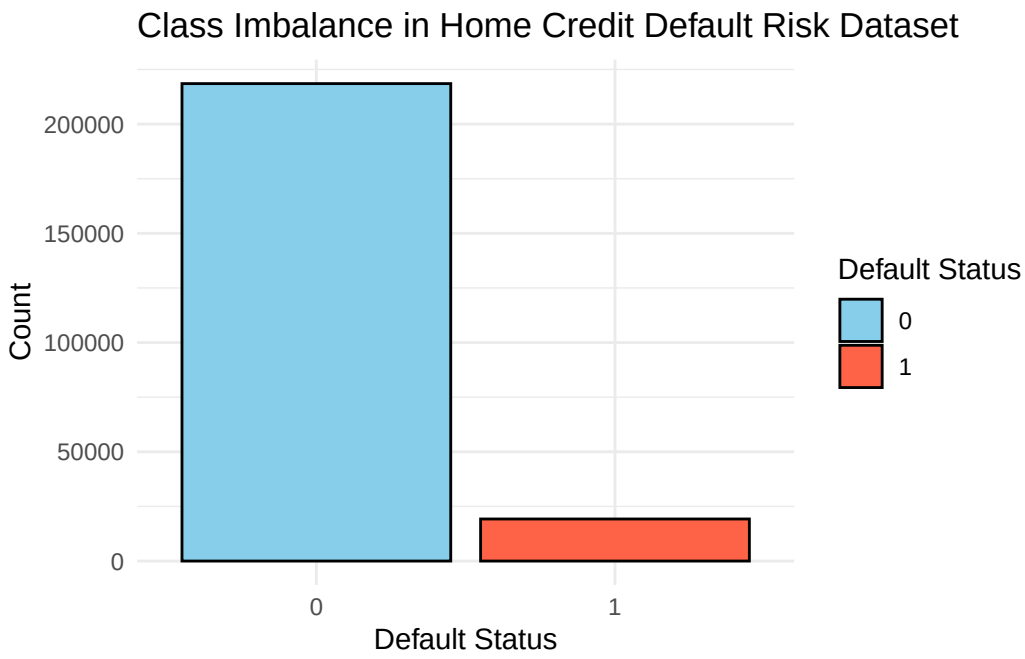
```
table(tr_clean$TARGET)  #class distribution
```

```
     0      1
218531  19245
```

```
round(prop.table(table(tr_clean$TARGET)),4) * 100  #percentages
```

```
    0     1
91.91  8.09
```

```
# bar plot to show class imbalance

ggplot(tr_clean, aes(x =TARGET, fill =TARGET)) +
  geom_bar(color = "black") +
  scale_fill_manual(values = c("0" = "skyblue", "1" = "tomato")) +
  theme_minimal() +
  labs(title = "Class Imbalance in Home Credit Default Risk Dataset",
    x = "Default Status",
    y = "Count",
    fill = "Default Status")
```

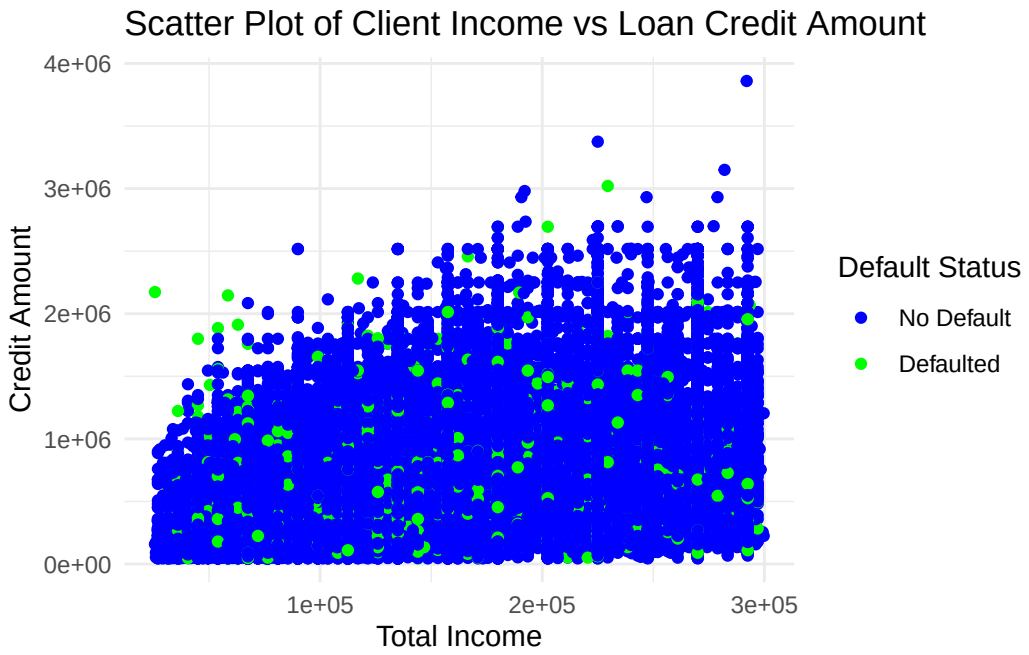


The plot shows there's a huge class imbalance and majority of the clients have re-payed the loan. 91.91% of them show successful repayment and our models have a lot to learn from this information.

Non-Linear Separability Check

```
# select any 2 random numeric variables and check scatter plot

tr_clean|> # use cleaned initial train data before split (as train_matrix doesn't contain ta
  ggplot(mapping = aes(x = AMT_INCOME_TOTAL,
                       y = AMT_CREDIT,
                       color = as.factor(TARGET))) +  # set color as TARGET
  geom_point() +

  labs(title = "Scatter Plot of Client Income vs Loan Credit Amount",
       x = "Total Income", # custom x-axis
       y = "Credit Amount", # custom y -axis
       color = "Default Status") +  # Add a label for the color legend

  scale_color_manual(values = c("0" = "blue", "1" = "green"),
```

```
                    labels = c("No Default", "Defaulted")) +
  theme_minimal()
```



We have randomly selected 2 numeric variables from the cleaned dataset and plotted the relationship between them filtered by default status. It is clearly seen there is a class overlap and the data is noisy. The relationships are complex and cannot be separated by a straight line, we hence implement the XGboost blackbox model to capture such patterns along with improved performance and bias.

We can also cross verify this by modelling linear models such as SVM with a linear kernel or a simple logistic regression. The models would likely perform poorly, a model like the linear SVM would struggle by taking a long training time and using many vectors to make the predictions which indicates non-linear data.

Data Split

```
# Set seed for reproducibility
set.seed(123)

# Ensure TARGET is a factor
#str(tr_clean$TARGET)
```

```r
# Split into training and validation sets

train_index <- createDataPartition(tr_clean$TARGET, p = 0.7, list = FALSE)

train_data <- tr_clean[train_index, ]

test_data <- tr_clean[-train_index, ]
```

Logistic Regression Performance

```r
# Fit glm model

logistic <- glm(TARGET~., data = train_data, family = binomial())

summary(logistic)
```

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

Coefficients: (1 not defined because of singularities)
                                                Estimate Std. Error z value
(Intercept)                                   -3.213e+01  3.969e+02  -0.081
NAME_CONTRACT_TYPERevolving loans             -7.497e-02  6.326e-02  -1.185
CODE_GENDERM                                   3.375e-01  2.495e-02  13.529
FLAG_OWN_CARY                                 -4.633e-01  4.829e-02  -9.594
FLAG_OWN_REALTYY                               1.954e-02  2.135e-02   0.915
CNT_CHILDREN                                   2.960e-02  1.569e-02   1.887
AMT_INCOME_TOTAL                              -4.992e-08  2.036e-07  -0.245
AMT_CREDIT                                     2.141e-06  1.568e-07  13.657
AMT_ANNUITY                                    1.203e-05  1.239e-06   9.707
AMT_GOODS_PRICE                               -2.665e-06  1.787e-07 -14.912
NAME_TYPE_SUITEFamily                         -4.353e-03  9.629e-02  -0.045
NAME_TYPE_SUITEGroup of people                 2.024e-01  3.085e-01   0.656
NAME_TYPE_SUITEOther_A                        -6.387e-02  1.926e-01  -0.332
NAME_TYPE_SUITEOther_B                         5.848e-02  1.461e-01   0.400
NAME_TYPE_SUITESpouse, partner                -1.064e-01  1.056e-01  -1.007
NAME_TYPE_SUITEUnaccompanied                  -4.463e-03  9.322e-02  -0.048
NAME_INCOME_TYPECommercial associate           1.048e+01  3.017e+02   0.035
```

NAME_INCOME_TYPEMaternity leave	1.388e+01	3.017e+02	0.046
NAME_INCOME_TYPEPensioner	-2.327e-01	3.692e+02	-0.001
NAME_INCOME_TYPEState servant	1.049e+01	3.017e+02	0.035
NAME_INCOME_TYPEStudent	-5.952e-01	3.327e+02	-0.002
NAME_INCOME_TYPEUnemployed	2.973e+00	3.692e+02	0.008
NAME_INCOME_TYPEWorking	1.059e+01	3.017e+02	0.035
NAME_EDUCATION_TYPEHigher education	1.070e+01	6.165e+01	0.174
NAME_EDUCATION_TYPEIncomplete higher	1.077e+01	6.165e+01	0.175
NAME_EDUCATION_TYPELower secondary	1.102e+01	6.165e+01	0.179
NAME_EDUCATION_TYPESecondary / secondary special	1.094e+01	6.165e+01	0.177
NAME_FAMILY_STATUSMarried	-1.322e-01	3.046e-02	-4.338
NAME_FAMILY_STATUSSeparated	4.853e-02	4.526e-02	1.072
NAME_FAMILY_STATUSSingle / not married	-5.443e-02	3.566e-02	-1.526
NAME_FAMILY_STATUSUnknown	-1.013e+01	5.354e+02	-0.019
NAME_FAMILY_STATUSWidow	-8.499e-02	5.350e-02	-1.588
NAME_HOUSING_TYPEHouse / apartment	1.033e-01	1.711e-01	0.604
NAME_HOUSING_TYPEMunicipal apartment	1.752e-01	1.777e-01	0.986
NAME_HOUSING_TYPEOffice apartment	-8.875e-02	2.041e-01	-0.435
NAME_HOUSING_TYPERented apartment	2.583e-01	1.812e-01	1.426
NAME_HOUSING_TYPEWith parents	1.321e-01	1.745e-01	0.757
REGION_POPULATION_RELATIVE	2.760e+00	9.695e-01	2.847
DAYS_BIRTH	-2.790e-05	3.250e-06	-8.586
DAYS_REGISTRATION	-1.109e-05	2.966e-06	-3.739
DAYS_ID_PUBLISH	-5.185e-05	6.677e-06	-7.765
OWN_CAR_AGE	2.103e-02	6.190e-03	3.397
FLAG_EMP_PHONE1	1.052e+01	2.505e+02	0.042
FLAG_WORK_PHONE1	1.934e-01	2.486e-02	7.780
FLAG_PHONE1	-9.585e-02	2.316e-02	-4.138
FLAG_EMAIL1	1.567e-02	4.307e-02	0.364
OCCUPATION_TYPECleaning staff	2.609e-01	9.389e-02	2.779
OCCUPATION_TYPECooking staff	1.601e-01	8.858e-02	1.808
OCCUPATION_TYPECore staff	1.505e-02	7.541e-02	0.200
OCCUPATION_TYPEDrivers	2.396e-01	7.884e-02	3.039
OCCUPATION_TYPEHigh skill tech staff	7.016e-02	8.442e-02	0.831
OCCUPATION_TYPEHR staff	4.174e-01	2.266e-01	1.842
OCCUPATION_TYPEIT staff	-4.723e-02	2.921e-01	-0.162
OCCUPATION_TYPELaborers	2.054e-01	6.998e-02	2.935
OCCUPATION_TYPELow-skill Laborers	3.284e-01	1.074e-01	3.057
OCCUPATION_TYPEManagers	3.775e-02	7.961e-02	0.474
OCCUPATION_TYPEMedicine staff	3.623e-02	9.891e-02	0.366
OCCUPATION_TYPEPrivate service staff	1.441e-02	1.282e-01	0.112
OCCUPATION_TYPERealty agents	-9.831e-02	2.163e-01	-0.454
OCCUPATION_TYPESales staff	1.299e-01	7.143e-02	1.819

OCCUPATION_TYPESecretaries	1.644e-01	1.620e-01	1.015
OCCUPATION_TYPESecurity staff	2.441e-01	9.652e-02	2.529
OCCUPATION_TYPEUnemployed	1.297e-01	7.021e-02	1.847
OCCUPATION_TYPEWaiters/barmen staff	2.721e-01	1.328e-01	2.048
CNT_FAM_MEMBERS	NA	NA	NA
REGION_RATING_CLIENT2	-3.482e-01	1.622e-01	-2.147
REGION_RATING_CLIENT3	-2.490e-01	1.638e-01	-1.520
REGION_RATING_CLIENT_W_CITY2	4.999e-01	1.548e-01	3.229
REGION_RATING_CLIENT_W_CITY3	5.553e-01	1.580e-01	3.514
WEEKDAY_APPR_PROCESS_STARTMONDAY	-8.573e-02	3.285e-02	-2.610
WEEKDAY_APPR_PROCESS_STARTSATURDAY	-6.681e-02	3.652e-02	-1.830
WEEKDAY_APPR_PROCESS_STARTSUNDAY	-1.110e-01	4.741e-02	-2.341
WEEKDAY_APPR_PROCESS_STARTTHURSDAY	-3.796e-02	3.257e-02	-1.166
WEEKDAY_APPR_PROCESS_STARTTUESDAY	2.076e-02	3.171e-02	0.655
WEEKDAY_APPR_PROCESS_STARTWEDNESDAY	-9.614e-03	3.221e-02	-0.298
HOUR_APPR_PROCESS_START	6.945e-04	3.068e-03	0.226
REG_REGION_NOT_WORK_REGION1	-7.786e-02	4.614e-02	-1.688
REG_CITY_NOT_LIVE_CITY1	1.861e-01	4.670e-02	3.986
REG_CITY_NOT_WORK_CITY1	-2.391e-03	5.219e-02	-0.046
LIVE_CITY_NOT_WORK_CITY1	3.323e-02	5.054e-02	0.658
ORGANIZATION_TYPEAgriculture	-2.935e-01	2.628e-01	-1.117
ORGANIZATION_TYPEBank	-4.943e-01	2.730e-01	-1.811
ORGANIZATION_TYPEBusiness Entity Type 1	-3.538e-01	2.532e-01	-1.397
ORGANIZATION_TYPEBusiness Entity Type 2	-3.117e-01	2.489e-01	-1.252
ORGANIZATION_TYPEBusiness Entity Type 3	-1.771e-01	2.445e-01	-0.724
ORGANIZATION_TYPECleaning	1.500e-02	3.579e-01	0.042
ORGANIZATION_TYPEConstruction	-5.584e-02	2.508e-01	-0.223
ORGANIZATION_TYPECulture	-7.263e-02	3.625e-01	-0.200
ORGANIZATION_TYPEElectricity	-3.759e-01	3.034e-01	-1.239
ORGANIZATION_TYPEEmergency	-3.099e-01	3.398e-01	-0.912
ORGANIZATION_TYPEGovernment	-3.314e-01	2.498e-01	-1.326
ORGANIZATION_TYPEHotel	-4.523e-01	2.984e-01	-1.516
ORGANIZATION_TYPEHousing	-4.194e-01	2.639e-01	-1.589
ORGANIZATION_TYPEIndustry: type 1	-1.198e-01	2.810e-01	-0.426
ORGANIZATION_TYPEIndustry: type 10	-2.031e-01	5.569e-01	-0.365
ORGANIZATION_TYPEIndustry: type 11	-2.816e-01	2.621e-01	-1.074
ORGANIZATION_TYPEIndustry: type 12	-8.838e-01	4.611e-01	-1.917
ORGANIZATION_TYPEIndustry: type 13	-2.529e-01	5.698e-01	-0.444
ORGANIZATION_TYPEIndustry: type 2	-5.986e-01	3.482e-01	-1.719
ORGANIZATION_TYPEIndustry: type 3	-1.463e-01	2.570e-01	-0.570
ORGANIZATION_TYPEIndustry: type 4	-1.903e-01	2.867e-01	-0.664
ORGANIZATION_TYPEIndustry: type 5	-4.307e-01	3.197e-01	-1.347
ORGANIZATION_TYPEIndustry: type 6	-1.044e+00	7.728e-01	-1.351

ORGANIZATION_TYPEIndustry: type 7	-3.597e-01	2.816e-01	-1.277
ORGANIZATION_TYPEIndustry: type 8	-4.583e-01	1.077e+00	-0.425
ORGANIZATION_TYPEIndustry: type 9	-6.141e-01	2.666e-01	-2.303
ORGANIZATION_TYPEInsurance	-4.141e-01	3.641e-01	-1.137
ORGANIZATION_TYPEKindergarten	-2.604e-01	2.525e-01	-1.031
ORGANIZATION_TYPELegal Services	2.940e-01	3.974e-01	0.740
ORGANIZATION_TYPEMedicine	-2.872e-01	2.525e-01	-1.137
ORGANIZATION_TYPEMilitary	-7.424e-01	2.776e-01	-2.675
ORGANIZATION_TYPEMobile	-1.855e-01	3.784e-01	-0.490
ORGANIZATION_TYPEOther	-3.066e-01	2.474e-01	-1.239
ORGANIZATION_TYPEPolice	-6.397e-01	2.813e-01	-2.274
ORGANIZATION_TYPEPostal	-1.588e-01	2.658e-01	-0.597
ORGANIZATION_TYPERealtor	5.412e-01	3.431e-01	1.577
ORGANIZATION_TYPEReligion	2.584e-01	5.467e-01	0.473
ORGANIZATION_TYPERestaurant	-6.858e-02	2.644e-01	-0.259
ORGANIZATION_TYPESchool	-4.187e-01	2.517e-01	-1.663
ORGANIZATION_TYPESecurity	-3.232e-01	2.646e-01	-1.222
ORGANIZATION_TYPESecurity Ministries	-5.352e-01	2.844e-01	-1.882
ORGANIZATION_TYPESelf-employed	-9.029e-02	2.451e-01	-0.368
ORGANIZATION_TYPEServices	-1.618e-01	2.822e-01	-0.573
ORGANIZATION_TYPETelecom	-2.906e-01	3.303e-01	-0.880
ORGANIZATION_TYPETrade: type 1	-7.032e-02	3.425e-01	-0.205
ORGANIZATION_TYPETrade: type 2	-6.417e-01	2.758e-01	-2.327
ORGANIZATION_TYPETrade: type 3	-1.041e-01	2.561e-01	-0.407
ORGANIZATION_TYPETrade: type 4	-1.505e+00	1.072e+00	-1.404
ORGANIZATION_TYPETrade: type 5	-1.123e+01	9.562e+01	-0.117
ORGANIZATION_TYPETrade: type 6	-5.597e-01	3.575e-01	-1.565
ORGANIZATION_TYPETrade: type 7	-1.348e-01	2.500e-01	-0.539
ORGANIZATION_TYPETransport: type 1	-1.631e+00	7.618e-01	-2.141
ORGANIZATION_TYPETransport: type 2	-3.182e-01	2.689e-01	-1.183
ORGANIZATION_TYPETransport: type 3	5.851e-01	2.706e-01	2.162
ORGANIZATION_TYPETransport: type 4	-2.007e-01	2.535e-01	-0.792
ORGANIZATION_TYPEUnemployed	2.101e+01	3.287e+02	0.064
ORGANIZATION_TYPEUniversity	-4.568e-01	3.006e-01	-1.519
EXT_SOURCE_1	-6.169e-01	4.050e-02	-15.232
EXT_SOURCE_2	-2.015e+00	4.851e-02	-41.526
EXT_SOURCE_3	-2.149e+00	4.754e-02	-45.197
APARTMENTS_MEDI	-4.082e-03	1.154e-01	-0.035
YEARS_BUILD_MEDI	-1.507e-01	9.518e-02	-1.584
COMMONAREA_MEDI	1.065e-01	1.393e-01	0.765
ELEVATORS_MEDI	-6.328e-02	9.660e-02	-0.655
ENTRANCES_MEDI	-1.401e-01	1.055e-01	-1.328
FLOORSMAX_MEDI	-1.572e-01	1.021e-01	-1.539

```
FLOORSMIN_MEDI                          8.244e-02  6.845e-02   1.204
LIVINGAPARTMENTS_MEDI                   3.676e-02  1.100e-01   0.334
LIVINGAREA_MEDI                         8.516e-02  1.124e-01   0.758
NONLIVINGAPARTMENTS_MEDI               -3.982e-01  2.281e-01  -1.746
OBS_30_CNT_SOCIAL_CIRCLE                5.179e-02  7.085e-02   0.731
DEF_30_CNT_SOCIAL_CIRCLE                1.572e-01  3.875e-02   4.057
OBS_60_CNT_SOCIAL_CIRCLE               -5.786e-02  7.158e-02  -0.808
DEF_60_CNT_SOCIAL_CIRCLE                5.402e-02  4.556e-02   1.186
DAYS_LAST_PHONE_CHANGE                 -5.456e-05  1.283e-05  -4.254
FLAG_DOCUMENT_31                        2.466e-01  5.526e-02   4.462
FLAG_DOCUMENT_61                        1.624e-01  7.099e-02   2.288
FLAG_DOCUMENT_81                       -3.124e-02  6.795e-02  -0.460
AMT_REQ_CREDIT_BUREAU_HOUR             -6.843e-02  1.260e-01  -0.543
AMT_REQ_CREDIT_BUREAU_DAY               1.206e-01  8.787e-02   1.373
AMT_REQ_CREDIT_BUREAU_WEEK             -5.026e-02  5.126e-02  -0.980
AMT_REQ_CREDIT_BUREAU_MON              -1.705e-02  1.372e-02  -1.242
AMT_REQ_CREDIT_BUREAU_QRT              -2.840e-02  1.626e-02  -1.747
AMT_REQ_CREDIT_BUREAU_YEAR              1.196e-02  5.361e-03   2.230
House_Attribute_Low_Variance           -1.686e-02  6.295e-03  -2.678
                                       Pr(>|z|)
(Intercept)                            0.935477
NAME_CONTRACT_TYPERevolving loans      0.235919
CODE_GENDERM                            < 2e-16 ***
FLAG_OWN_CARY                           < 2e-16 ***
FLAG_OWN_REALTYY                       0.360203
CNT_CHILDREN                           0.059103 .
AMT_INCOME_TOTAL                       0.806348
AMT_CREDIT                              < 2e-16 ***
AMT_ANNUITY                             < 2e-16 ***
AMT_GOODS_PRICE                         < 2e-16 ***
NAME_TYPE_SUITEFamily                  0.963938
NAME_TYPE_SUITEGroup of people         0.511867
NAME_TYPE_SUITEOther_A                 0.740244
NAME_TYPE_SUITEOther_B                 0.688993
NAME_TYPE_SUITESpouse, partner         0.313820
NAME_TYPE_SUITEUnaccompanied           0.961813
NAME_INCOME_TYPECommercial associate   0.972286
NAME_INCOME_TYPEMaternity leave        0.963305
NAME_INCOME_TYPEPensioner              0.999497
NAME_INCOME_TYPEState servant          0.972258
NAME_INCOME_TYPEStudent                0.998572
NAME_INCOME_TYPEUnemployed             0.993574
NAME_INCOME_TYPEWorking                0.971994
```

```
NAME_EDUCATION_TYPEHigher education              0.862192
NAME_EDUCATION_TYPEIncomplete higher             0.861346
NAME_EDUCATION_TYPELower secondary               0.858148
NAME_EDUCATION_TYPESecondary / secondary special 0.859204
NAME_FAMILY_STATUSMarried                        1.44e-05 ***
NAME_FAMILY_STATUSSeparated                      0.283623
NAME_FAMILY_STATUSSingle / not married           0.126921
NAME_FAMILY_STATUSUnknown                        0.984904
NAME_FAMILY_STATUSWidow                          0.112199
NAME_HOUSING_TYPEHouse / apartment               0.546017
NAME_HOUSING_TYPEMunicipal apartment             0.323923
NAME_HOUSING_TYPEOffice apartment                0.663664
NAME_HOUSING_TYPERented apartment                0.153906
NAME_HOUSING_TYPEWith parents                    0.449076
REGION_POPULATION_RELATIVE                       0.004417 **
DAYS_BIRTH                                        < 2e-16 ***
DAYS_REGISTRATION                                0.000184 ***
DAYS_ID_PUBLISH                                  8.16e-15 ***
OWN_CAR_AGE                                      0.000681 ***
FLAG_EMP_PHONE1                                  0.966487
FLAG_WORK_PHONE1                                 7.25e-15 ***
FLAG_PHONE1                                      3.50e-05 ***
FLAG_EMAIL1                                      0.715954
OCCUPATION_TYPECleaning staff                    0.005453 **
OCCUPATION_TYPECooking staff                     0.070658 .
OCCUPATION_TYPECore staff                        0.841827
OCCUPATION_TYPEDrivers                           0.002376 **
OCCUPATION_TYPEHigh skill tech staff             0.405917
OCCUPATION_TYPEHR staff                          0.065507 .
OCCUPATION_TYPEIT staff                          0.871555
OCCUPATION_TYPELaborers                          0.003337 **
OCCUPATION_TYPELow-skill Laborers                0.002234 **
OCCUPATION_TYPEManagers                          0.635358
OCCUPATION_TYPEMedicine staff                    0.714150
OCCUPATION_TYPEPrivate service staff             0.910532
OCCUPATION_TYPERealty agents                     0.649500
OCCUPATION_TYPESales staff                       0.068905 .
OCCUPATION_TYPESecretaries                       0.309986
OCCUPATION_TYPESecurity staff                    0.011424 *
OCCUPATION_TYPEUnemployed                        0.064803 .
OCCUPATION_TYPEWaiters/barmen staff              0.040528 *
CNT_FAM_MEMBERS                                        NA
REGION_RATING_CLIENT2                            0.031799 *
```

```
REGION_RATING_CLIENT3                          0.128516
REGION_RATING_CLIENT_W_CITY2                   0.001241 **
REGION_RATING_CLIENT_W_CITY3                   0.000442 ***
WEEKDAY_APPR_PROCESS_STARTMONDAY               0.009064 **
WEEKDAY_APPR_PROCESS_STARTSATURDAY             0.067308 .
WEEKDAY_APPR_PROCESS_STARTSUNDAY               0.019231 *
WEEKDAY_APPR_PROCESS_STARTTHURSDAY             0.243762
WEEKDAY_APPR_PROCESS_STARTTUESDAY              0.512630
WEEKDAY_APPR_PROCESS_STARTWEDNESDAY            0.765336
HOUR_APPR_PROCESS_START                        0.820895
REG_REGION_NOT_WORK_REGION1                    0.091505 .
REG_CITY_NOT_LIVE_CITY1                        6.73e-05 ***
REG_CITY_NOT_WORK_CITY1                        0.963456
LIVE_CITY_NOT_WORK_CITY1                       0.510802
ORGANIZATION_TYPEAgriculture                   0.264040
ORGANIZATION_TYPEBank                          0.070149 .
ORGANIZATION_TYPEBusiness Entity Type 1        0.162385
ORGANIZATION_TYPEBusiness Entity Type 2        0.210576
ORGANIZATION_TYPEBusiness Entity Type 3        0.468992
ORGANIZATION_TYPECleaning                      0.966556
ORGANIZATION_TYPEConstruction                  0.823794
ORGANIZATION_TYPECulture                       0.841194
ORGANIZATION_TYPEElectricity                   0.215395
ORGANIZATION_TYPEEmergency                     0.361776
ORGANIZATION_TYPEGovernment                    0.184684
ORGANIZATION_TYPEHotel                         0.129554
ORGANIZATION_TYPEHousing                       0.111960
ORGANIZATION_TYPEIndustry: type 1              0.669789
ORGANIZATION_TYPEIndustry: type 10             0.715333
ORGANIZATION_TYPEIndustry: type 11             0.282710
ORGANIZATION_TYPEIndustry: type 12             0.055278 .
ORGANIZATION_TYPEIndustry: type 13             0.657130
ORGANIZATION_TYPEIndustry: type 2              0.085575 .
ORGANIZATION_TYPEIndustry: type 3              0.568993
ORGANIZATION_TYPEIndustry: type 4              0.506831
ORGANIZATION_TYPEIndustry: type 5              0.177871
ORGANIZATION_TYPEIndustry: type 6              0.176597
ORGANIZATION_TYPEIndustry: type 7              0.201478
ORGANIZATION_TYPEIndustry: type 8              0.670608
ORGANIZATION_TYPEIndustry: type 9              0.021263 *
ORGANIZATION_TYPEInsurance                     0.255464
ORGANIZATION_TYPEKindergarten                  0.302459
ORGANIZATION_TYPELegal Services                0.459509
```

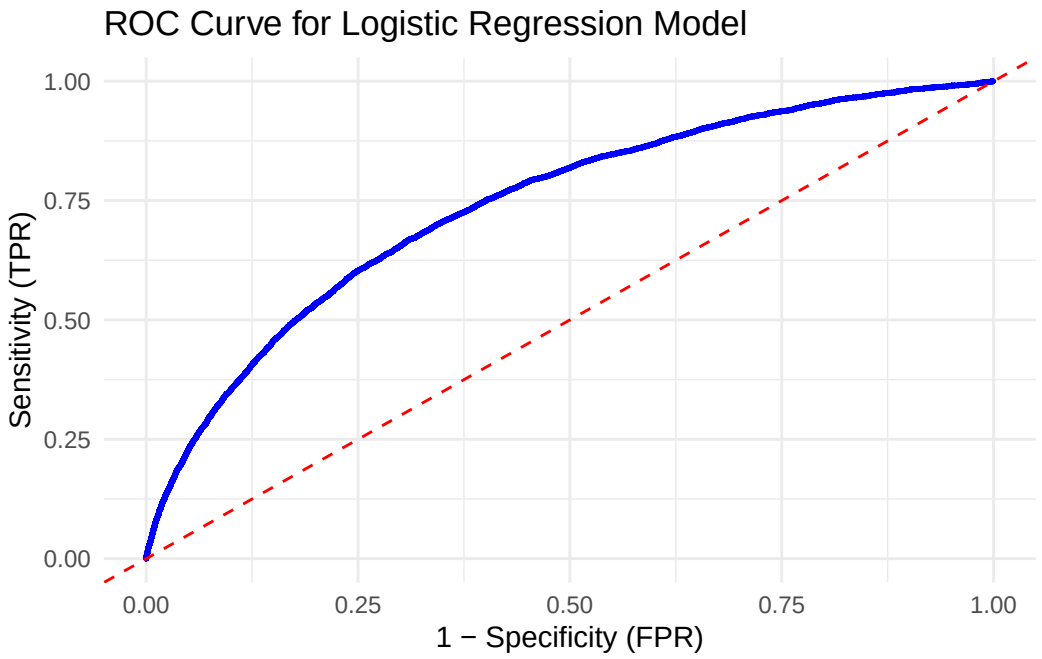
ORGANIZATION_TYPEMedicine	0.255454	
ORGANIZATION_TYPEMilitary	0.007478	**
ORGANIZATION_TYPEMobile	0.623972	
ORGANIZATION_TYPEOther	0.215188	
ORGANIZATION_TYPEPolice	0.022973	*
ORGANIZATION_TYPEPostal	0.550271	
ORGANIZATION_TYPERealtor	0.114768	
ORGANIZATION_TYPEReligion	0.636550	
ORGANIZATION_TYPERestaurant	0.795332	
ORGANIZATION_TYPESchool	0.096217	.
ORGANIZATION_TYPESecurity	0.221891	
ORGANIZATION_TYPESecurity Ministries	0.059868	.
ORGANIZATION_TYPESelf-employed	0.712593	
ORGANIZATION_TYPEServices	0.566395	
ORGANIZATION_TYPETelecom	0.379055	
ORGANIZATION_TYPETrade: type 1	0.837323	
ORGANIZATION_TYPETrade: type 2	0.019972	*
ORGANIZATION_TYPETrade: type 3	0.684347	
ORGANIZATION_TYPETrade: type 4	0.160273	
ORGANIZATION_TYPETrade: type 5	0.906499	
ORGANIZATION_TYPETrade: type 6	0.117467	
ORGANIZATION_TYPETrade: type 7	0.589750	
ORGANIZATION_TYPETransport: type 1	0.032257	*
ORGANIZATION_TYPETransport: type 2	0.236671	
ORGANIZATION_TYPETransport: type 3	0.030586	*
ORGANIZATION_TYPETransport: type 4	0.428435	
ORGANIZATION_TYPEUnemployed	0.949029	
ORGANIZATION_TYPEUniversity	0.128648	
EXT_SOURCE_1	< 2e-16	***
EXT_SOURCE_2	< 2e-16	***
EXT_SOURCE_3	< 2e-16	***
APARTMENTS_MEDI	0.971790	
YEARS_BUILD_MEDI	0.113291	
COMMONAREA_MEDI	0.444559	
ELEVATORS_MEDI	0.512398	
ENTRANCES_MEDI	0.184335	
FLOORSMAX_MEDI	0.123833	
FLOORSMIN_MEDI	0.228457	
LIVINGAPARTMENTS_MEDI	0.738227	
LIVINGAREA_MEDI	0.448476	
NONLIVINGAPARTMENTS_MEDI	0.080860	.
OBS_30_CNT_SOCIAL_CIRCLE	0.464746	
DEF_30_CNT_SOCIAL_CIRCLE	4.97e-05	***

```
OBS_60_CNT_SOCIAL_CIRCLE                    0.418945
DEF_60_CNT_SOCIAL_CIRCLE                    0.235814
DAYS_LAST_PHONE_CHANGE                      2.10e-05 ***
FLAG_DOCUMENT_31                            8.10e-06 ***
FLAG_DOCUMENT_61                            0.022158 *
FLAG_DOCUMENT_81                            0.645695
AMT_REQ_CREDIT_BUREAU_HOUR                  0.587073
AMT_REQ_CREDIT_BUREAU_DAY                   0.169888
AMT_REQ_CREDIT_BUREAU_WEEK                  0.326845
AMT_REQ_CREDIT_BUREAU_MON                   0.214147
AMT_REQ_CREDIT_BUREAU_QRT                   0.080677 .
AMT_REQ_CREDIT_BUREAU_YEAR                  0.025743 *
House_Attribute_Low_Variance                0.007411 **
---
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: 83686  on 166280  degrees of freedom
AIC: 84014

Number of Fisher Scoring iterations: 12
```

```r
# Predictions

## Predictions on the test data (probabilities)
test_pred_probs <- predict(logistic, newdata = test_data, type = "response")

## Convert probabilities to binary prediction outcomes using threshold = 0.5

test_predictions <- ifelse(test_pred_probs > 0.5, 1, 0)


# Evaluation

## Calculate ROC-AUC
roc_obj <- roc(test_data$TARGET, test_pred_probs)
roc_auc <- auc(roc_obj)
cat("ROC-AUC:", round(roc_auc, 4), "\n")
```

```
ROC-AUC: 0.7391
```

```r
## Plot ROC Curve
roc_curve <- data.frame(
  FPR = 1 - roc_obj$specificities,
  TPR = roc_obj$sensitivities
)

ggplot(roc_curve, aes(x = FPR, y = TPR)) +
  geom_line(color = "blue", size = 1) +
  geom_abline(linetype = "dashed", color = "red") +
  labs(title = "ROC Curve for Logistic Regression Model",
       x = "1 - Specificity (FPR)",
       y = "Sensitivity (TPR)") +
  theme_minimal()
```




```r
## Calculate accuracy
accuracy <- mean(test_predictions == test_data$TARGET)
cat("Accuracy:", round(accuracy, 4), "\n")
```

```
Accuracy: 0.919
```

The model shows high accuracy (91.91%) but this might be inflated due to an imbalanced dataset where non-defaults dominate. The AUC value of 73.91% suggests the model has ac-

ceptable discriminatory power in distinguishing between default and non-default cases, though there's room for improvement.

XGBoost

Define Model

```
library(tidymodels)
library(dials)
library(doParallel)

# Adjust for class imbalance: calculate scale_pos_weight
scale_pos_weight <- sum(train_data$TARGET == 0) / sum(train_data$TARGET == 1)
print(scale_pos_weight)
```

```
[1] 11.35481
```

```
# Set scale_pos_weight as an option
options(scale_pos_weight = scale_pos_weight)

# Define XGBoost model
xgb_model <- boost_tree(
  trees = tune(),
  tree_depth = tune(),
  learn_rate = tune(),
  mtry = tune(),
  min_n = tune(),
  sample_size = tune(),
  loss_reduction = tune()
) %>%
  set_engine("xgboost",
             early_stopping_rounds = 50,
             scale_pos_weight = getOption("scale_pos_weight"),
             tree_method = "hist",
             max_bin = 256,
             nthread = 4,
             verbosity = 1 ) %>%
  set_mode("classification")
```

Define Recipe

```
# Define Recipe
recipe <- recipe(TARGET ~ ., data = train_data) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_novel(all_nominal_predictors()) %>%
  step_other(all_nominal_predictors(), threshold = 0.05)
```

Define Grid search parameter tuning and CV function

```
# Define hyperparameter tuning grid
xgb_grid <- grid_random(
  trees(range = c(50, 100)),         # number of trees in each model
  tree_depth(range = c(3, 6)),
  learn_rate(range = c(0.01, 0.2)),
  finalize(mtry(), train_data),
  min_n(range = c(5, 20)),
  sample_size = sample_prop(c(0.7, 1)),  # prop. of data in each node
  loss_reduction(range = c(0, 1)),
  size = 50                          # 50 tree models
)

# Define 5-fold cross-validation
cv_folds <- vfold_cv(train_data, v = 5, strata = TARGET)
```

Train the Model

```
# Register parallel processing
library(doParallel)
num_cores <- detectCores()
num_cores
```

```
[1] 8
```

```r
registerDoParallel(cores = 4)


# Define the workflow
wf <- workflow() %>%
  add_model(xgb_model) %>%
  add_recipe(recipe)


# Tune the model

library(yardstick)

metrics <- yardstick::metric_set(yardstick::roc_auc, yardstick::mn_log_loss, yardstick::accu

set.seed(123)
xgb_tune_model <- tune_grid(
  wf,
  resamples = cv_folds,
  grid = xgb_grid,
  metrics = metrics
)

# Stop parallel processing
stopImplicitCluster()


#show errors
#show_notes(xgb_tune_model)
#collect_notes(xgb_tune_model)
```

Evaluate Model

Identify best hyperparameters and finalize Workflow

```r
# Collect fold-specific metrics
train_metrics <- xgb_tune_model %>%
  collect_metrics()

cat("Model Training Performance:\n")
```

Model Training Performance:

```
print(train_metrics)
```

```
# A tibble: 150 x 13
    mtry trees min_n tree_depth learn_rate loss_reduction sample_size .metric
   <int> <int> <int>      <int>      <dbl>          <dbl>       <dbl> <chr>
 1    32    69    14          3       1.06           1.90       0.942 accuracy
 2    32    69    14          3       1.06           1.90       0.942 mn_log_lo~
 3    32    69    14          3       1.06           1.90       0.942 roc_auc
 4    28    60     9          3       1.07           1.72       0.784 accuracy
 5    28    60     9          3       1.07           1.72       0.784 mn_log_lo~
 6    28    60     9          3       1.07           1.72       0.784 roc_auc
 7    26    58     7          3       1.07           1.98       0.924 accuracy
 8    26    58     7          3       1.07           1.98       0.924 mn_log_lo~
 9    26    58     7          3       1.07           1.98       0.924 roc_auc
10    15    69    19          3       1.07           5.45       0.958 accuracy
# i 140 more rows
# i 5 more variables: .estimator <chr>, mean <dbl>, n <int>, std_err <dbl>,
#   .config <chr>
```

```
# Extract the best parameters using a single metric (e.g., `roc_auc`)
best_params <- xgb_tune_model %>%
  select_best(metric = "roc_auc")

best_params
```

```
# A tibble: 1 x 8
   mtry trees min_n tree_depth learn_rate loss_reduction sample_size .config
  <int> <int> <int>      <int>      <dbl>          <dbl>       <dbl> <chr>
1    15    69    19          3       1.07           5.45       0.958 Preprocess~
```

```
# Finalize the workflow with the best parameters
final_wf <- wf %>%
  finalize_workflow(best_params)

# Fit the final model on the training data
final_fit <- final_wf %>%
  fit(data = train_data)

final_fit
```

```
== Workflow [trained] ==========================================================
Preprocessor: Recipe
Model: boost_tree()

-- Preprocessor ----------------------------------------------------------------
5 Recipe Steps

* step_dummy()
* step_zv()
* step_normalize()
* step_novel()
* step_other()

-- Model -----------------------------------------------------------------------
##### xgb.Booster
raw: 54 Kb
call:
  xgboost::xgb.train(params = list(eta = 1.06821538767645, max_depth = 3L,
    gamma = 5.45186158492749, colsample_bytree = 1, colsample_bynode = 0.0914634146341463,
    min_child_weight = 19L, subsample = 0.957812811527401), data = x$data,
    nrounds = 69L, watchlist = x$watchlist, verbose = 0, early_stopping_rounds = 50,
    scale_pos_weight = 11.354809976247, tree_method = "hist",
    max_bin = 256, nthread = 4, verbosity = 1, objective = "binary:logistic")
params (as set within xgb.train):
  eta = "1.06821538767645", max_depth = "3", gamma = "5.45186158492749", colsample_bytree = "
xgb.attributes:
  best_iteration, best_msg, best_ntreelimit, best_score, niter
callbacks:
  cb.evaluation.log()
  cb.early.stop(stopping_rounds = early_stopping_rounds, maximize = maximize,
    verbose = verbose)
# of features: 164
niter: 51
best_iteration : 1
best_ntreelimit : 1
best_score : 0.285393
best_msg : [1]  training-logloss:0.285393
nfeatures : 164
evaluation_log:
    iter training_logloss
       1        0.2853930
       2        0.2973932
---
```

```
50        0.3665277
51        0.3663267
```

Evaluate on Train Data

```
# Make predictions on the training data
train_predictions <- final_fit %>%
  predict(new_data = train_data, type = "prob")

# Combine data frames: actual target values from train data and predictions (from train predi
train_results <- bind_cols(train_data, train_predictions)


# Create class predictions based on threshold = 0.5
train_results <- train_results %>%
  mutate(pred_class = ifelse(.pred_1 >= 0.5, "1", "0"))

# Convert `pred_class` to factor
train_results <- train_results %>%
  mutate(pred_class = factor(pred_class, levels = levels(train_data$TARGET)))


# Evaluate on the training set

# ROC AUC
roc_auc_train <- yardstick::roc_auc(train_results, truth = TARGET, .pred_1)

# Log Loss
log_loss_train <- yardstick::mn_log_loss(train_results, truth = TARGET, .pred_1)

# Accuracy
accuracy_train <- yardstick::accuracy(train_results, truth = TARGET, estimate = pred_class)



# Error (1 - Accuracy)
accuracy_value <- accuracy(train_results, truth = TARGET, estimate = pred_class)
error_train <- 1 - accuracy_value$.estimate

#  AUCPR (Area Under Precision-Recall Curve)
aupr_train <- pr_auc(train_results, truth = TARGET, .pred_1)
```

```r
# All train metrics
train_metrics <- data.frame(
  Metric = c("ROC AUC", "Log Loss", "Accuracy", "Error", "AUCPR"),
  Value = c(
    roc_auc_train$.estimate,
    log_loss_train$.estimate,
    accuracy_value$.estimate,
    error_train,
    aupr_train$.estimate
  )
)

train_metrics
```

```
    Metric      Value
1  ROC AUC 0.50000000
2 Log Loss 2.04846364
3 Accuracy 0.91905986
4    Error 0.08094014
5    AUCPR 0.95952993
```

Evaluate on Test Data

```r
# Make predictions on the test data
test_predictions <- final_fit %>%
  predict(new_data = test_data, type = "prob")

head(test_predictions)
```

```
# A tibble: 6 x 2
  .pred_0 .pred_1
    <dbl>   <dbl>
1   0.891   0.109
2   0.891   0.109
3   0.891   0.109
4   0.891   0.109
5   0.891   0.109
6   0.891   0.109
```

```r
# Combine data frames: actual target values from test data and predictions (from test predict

test_results <- bind_cols(test_data, test_predictions)

head(test_results)
```

```
# A tibble: 6 x 65
  TARGET NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY
  <fct>  <chr>              <chr>       <chr>        <chr>
1 0      Cash loans         F           N            Y
2 0      Cash loans         M           N            Y
3 0      Cash loans         F           N            Y
4 0      Revolving loans    M           N            Y
5 0      Cash loans         F           N            Y
6 0      Revolving loans    M           Y            Y
# i 60 more variables: CNT_CHILDREN <dbl>, AMT_INCOME_TOTAL <dbl>,
#   AMT_CREDIT <dbl>, AMT_ANNUITY <dbl>, AMT_GOODS_PRICE <dbl>,
#   NAME_TYPE_SUITE <chr>, NAME_INCOME_TYPE <chr>, NAME_EDUCATION_TYPE <chr>,
#   NAME_FAMILY_STATUS <chr>, NAME_HOUSING_TYPE <chr>,
#   REGION_POPULATION_RELATIVE <dbl>, DAYS_BIRTH <dbl>,
#   DAYS_REGISTRATION <dbl>, DAYS_ID_PUBLISH <dbl>, OWN_CAR_AGE <dbl>,
#   FLAG_EMP_PHONE <fct>, FLAG_WORK_PHONE <fct>, FLAG_PHONE <fct>, ...
```

```r
# Evaluate on the test set

library(yardstick)

#test_metrics <- yardstick::metrics(test_results, truth = TARGET, .pred_1) %>%
  #filter(.metric %in% c("roc_auc", "mn_log_loss", "accuracy"))


# Create class predictions based on threshold = 0.5

test_results <- test_results %>%
   mutate(pred_class = ifelse(.pred_1 >= 0.5, "1", "0"))

# Convert `pred_class` to be a factor
test_results <- test_results %>%
  mutate(pred_class = factor(pred_class, levels = levels(test_results$TARGET)))

# Convert `TARGET` and `pred_class` to factors with the same levels
```

```r
test_results <- test_results %>%
  mutate(
    TARGET = factor(TARGET, levels = c("0", "1")),
    pred_class = factor(pred_class, levels = c("0", "1"))
  )



# ROC: uses class predicted probabilities

roc_auc_result <- yardstick::roc_auc(test_results, truth = TARGET, .pred_1)


# Log Loss: uses class predicted probabilities
log_loss_result <- yardstick::mn_log_loss(test_results, truth = TARGET, .pred_1)

# Accuracy: uses factored class prediction
accuracy_result <- yardstick::accuracy(test_results, truth = TARGET, estimate = pred_class)

# Error
error_test <- yardstick::metrics(test_results, truth = TARGET, estimate = pred_class) %>%
  filter(.metric == "accuracy") %>%
  mutate(.estimate = 1 - .estimate)

# AUCPR
aupr_test <- yardstick::pr_auc(test_results, truth = TARGET, .pred_1)

# MAP (Mean Average Precision) - equivalent to precision at different recall thresholds

map_test <- yardstick::average_precision(test_results, truth = TARGET, .pred_1)


# All test Metrics
test_metrics <- data.frame(
  Metric = c("ROC AUC", "Log Loss", "Accuracy", "Error", "AUCPR", "MAP"),
  Value = c(
    roc_auc_result$.estimate,
    log_loss_result$.estimate,
    accuracy_result$.estimate,
    error_test$.estimate,
    aupr_test$.estimate,
    map_test$.estimate
```

```
  )
)

test_metrics
```

```
    Metric      Value
1  ROC AUC 0.50000000
2 Log Loss 2.04848198
3 Accuracy 0.91906858
4    Error 0.08093142
5    AUCPR 0.95953429
6      MAP 0.91906858
```

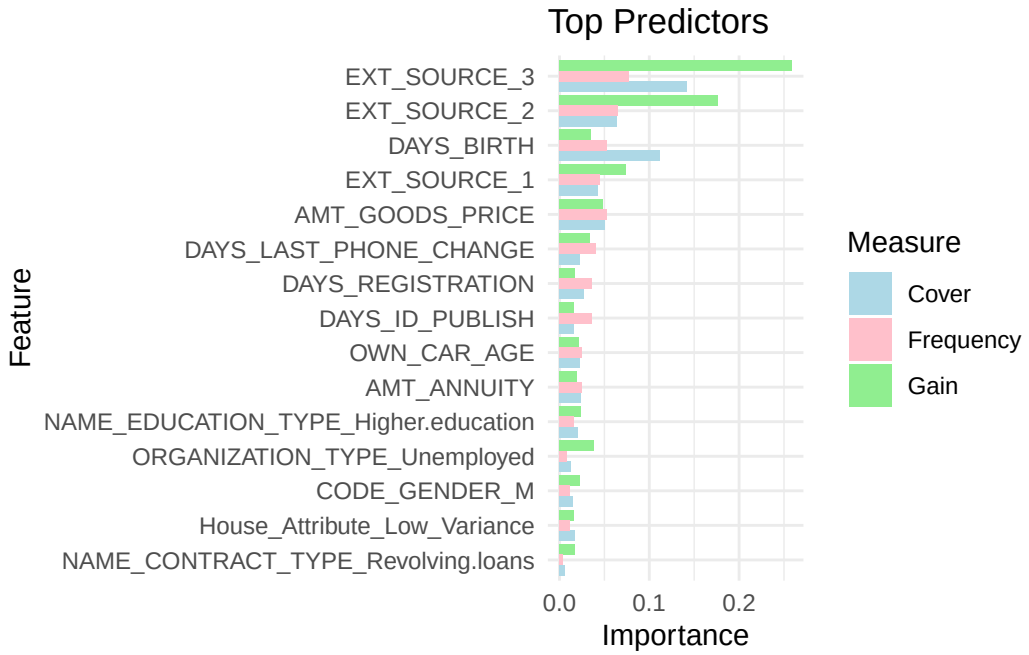
Feature Importance

```
xgb_booster <- extract_fit_engine(final_fit)

xgb_importance <- xgboost::xgb.importance(model = xgb_booster)

head(xgb_importance)
```

```
                       Feature       Gain      Cover   Frequency
1:                 EXT_SOURCE_3 0.25871787 0.14168321 0.077551020
2:                 EXT_SOURCE_2 0.17660937 0.06412984 0.065306122
3:                 EXT_SOURCE_1 0.07449437 0.04275258 0.044897959
4:              AMT_GOODS_PRICE 0.04857556 0.05077497 0.053061224
5: ORGANIZATION_TYPE_Unemployed 0.03849605 0.01231684 0.008163265
6:                   DAYS_BIRTH 0.03542029 0.11242377 0.053061224
```

```
# Convert the xgb_importance data to a data frame
xgb_importance_df <- as.data.frame(xgb_importance)

# Select the top 15 important features based on Gain
top_features <- xgb_importance_df %>%
  arrange(desc(Gain)) %>%
  head(15)

# Reshape the data for ggplot2 (long format)
```

```r
importance_long <- top_features %>%
  select(Feature, Gain, Cover, Frequency) %>%
  tidyr::pivot_longer(cols = c("Gain", "Cover", "Frequency"),
                      names_to = "Measure",
                      values_to = "Importance")

# Plot
ggplot(importance_long, aes(x = reorder(Feature, Importance), y = Importance, fill = Measure)
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +
  labs(title = "Top Predictors",
       x = "Feature",
       y = "Importance") +
  scale_fill_manual(values = c("lightblue", "pink", "lightgreen")) +
  theme_minimal()
```



The plot illustrates the top predictors based on three metrics: Gain, Cover, and Frequency. Variables such as External sources, DAYS_BIRTH, AMT_GOODS_PRICE, DAYS_REGISTRATION, OWN_CAR_AGE, DAYS_LAST_PHONE_CHANGE are the most significant drivers of the model's predictions. The Gain (green bars) indicates the relative contribution of each feature to the model's accuracy. The Cover (blue bars) represents the proportion of observations that utilize the feature, with EXT_SOURCE_3 and DAYS_BIRTH covering a large share of the dataset. The Frequency (pink bars) shows

how often the feature is used in the model's decision trees, suggesting that external source features are critical predictors for loan default.

Results

Make Predictions with the Test Application Data

```
# Ensure factors and the levels in `test_clean for all predictors match those in the training

te_clean$REGION_RATING_CLIENT_W_CITY <- factor(te_clean$REGION_RATING_CLIENT_W_CITY,
  levels = levels(train_data$REGION_RATING_CLIENT_W_CITY)
)

# remove any remaining NA's( just 1 was left)
sum(is.na(te_clean$REGION_RATING_CLIENT_W_CITY))
```

```
[1] 1
```

```
te_clean <- te_clean[!is.na(te_clean$REGION_RATING_CLIENT_W_CITY), ]


# Predict on test data


boost_pred <- predict(final_fit, new_data = te_clean, type = "prob")
```

Format the predictions into an acceptable format for Kaggle

```
# Convert predictions to a dataframe

boost_pred <- as.data.frame(boost_pred)

head(boost_pred)
```

```
    .pred_0   .pred_1
1 0.8912528 0.1087472
```

```
2 0.8912528 0.1087472
3 0.8912528 0.1087472
4 0.8912528 0.1087472
5 0.8912528 0.1087472
6 0.8912528 0.1087472
```

```
boost_pred1 <- boost_pred %>%
  select(".pred_0") %>% # Select only the zero column [ prob. of no default]
  dplyr::pull() # Pull this column out and convert to vector.



# Update kaggle_submission with the filtered predictions

kaggle_submission <- te_clean %>%
  select(SK_ID_CURR) %>% # Select the ID
  mutate(SK_ID_CURR = as.integer(SK_ID_CURR), # Convert SK_ID_CURR to an integer per Kaggle's
         TARGET = boost_pred1) # Kaggle wants Target to be a column in the dataset, so create


head(kaggle_submission) # Check the first few predictions
```

```
# A tibble: 6 x 2
  SK_ID_CURR TARGET
       <int>  <dbl>
1     100001  0.891
2     100005  0.891
3     100013  0.891
4     100042  0.891
5     100057  0.891
6     100065  0.891
```

```
# Specify where you want to export the submissions

setwd("C:/Users/nikit/Downloads/Capstone Project")

# Write the predictions to a csv file.

write.csv(kaggle_submission, "C:/Users/nikit/Downloads/Capstone Project/kaggle_submission.csv

ks <- read_csv("C:/Users/nikit/Downloads/Capstone Project/kaggle_submission.csv")
```

```r
head(ks)
```

```
# A tibble: 6 x 2
  SK_ID_CURR TARGET
       <dbl>  <dbl>
1     100001  0.891
2     100005  0.891
3     100013  0.891
4     100042  0.891
5     100057  0.891
6     100065  0.891
```

```r
str(ks)
```

```
spc_tbl_ [37,739 x 2] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ SK_ID_CURR: num [1:37739] 1e+05 1e+05 1e+05 1e+05 1e+05 ...
 $ TARGET    : num [1:37739] 0.891 0.891 0.891 0.891 0.891 ...
 - attr(*, "spec")=
  .. cols(
  ..   SK_ID_CURR = col_double(),
  ..   TARGET = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
```

```r
ks <- read_csv("C:/Users/nikit/Downloads/Capstone Project/kaggle_submission.csv",
               col_types = cols(
                 SK_ID_CURR = col_integer(),
                 TARGET = col_double()
               ))

ks <- read_csv("C:/Users/nikit/Downloads/Capstone Project/kaggle_submission.csv", show_col_ty
```

Final Interpretations and Conclusion

The XGBoost model achieves an accuracy of 91.9% on the training set and 91% on the test set, but its ROC AUC of 0.5 on both sets indicates poor class discrimination, effectively performing no better than random guessing. However, its high AUC-PR scores (0.9595 for training and 0.95 for testing) suggest better handling of the minority class compared to the decision tree model, which showed high accuracy but a complete failure to identify minority

instances. Unlike the decision tree, which struggled with the minority class (with a TPR of 0 and F1 score of 0), XGBoost performs better in this regard, though there's still room for improvement. When compared to the Naive Bayes model, which had a relatively faster training time (207.415 sec) and good overall efficiency but was also weak in classifying the minority class, XGBoost shows a better balance between accuracy and handling imbalanced data. Despite its low log loss (2.048) and relatively low error (8%), further optimization could improve the discriminatory power of XGBoost.