Home Credit Default Risk

Kornkanok Somkul
Shashi Bala

Project Overview and Flow Diagram

Data Exploration

Home Credit Default Risk

Define problem statement

Exploratory Data Analysis

Tool used: Python in Jupyter Notebook

python™

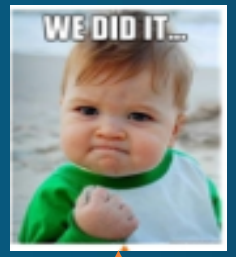
jupyter Notebook

Data Cleaning and Pre-processing

Model creation with new data set

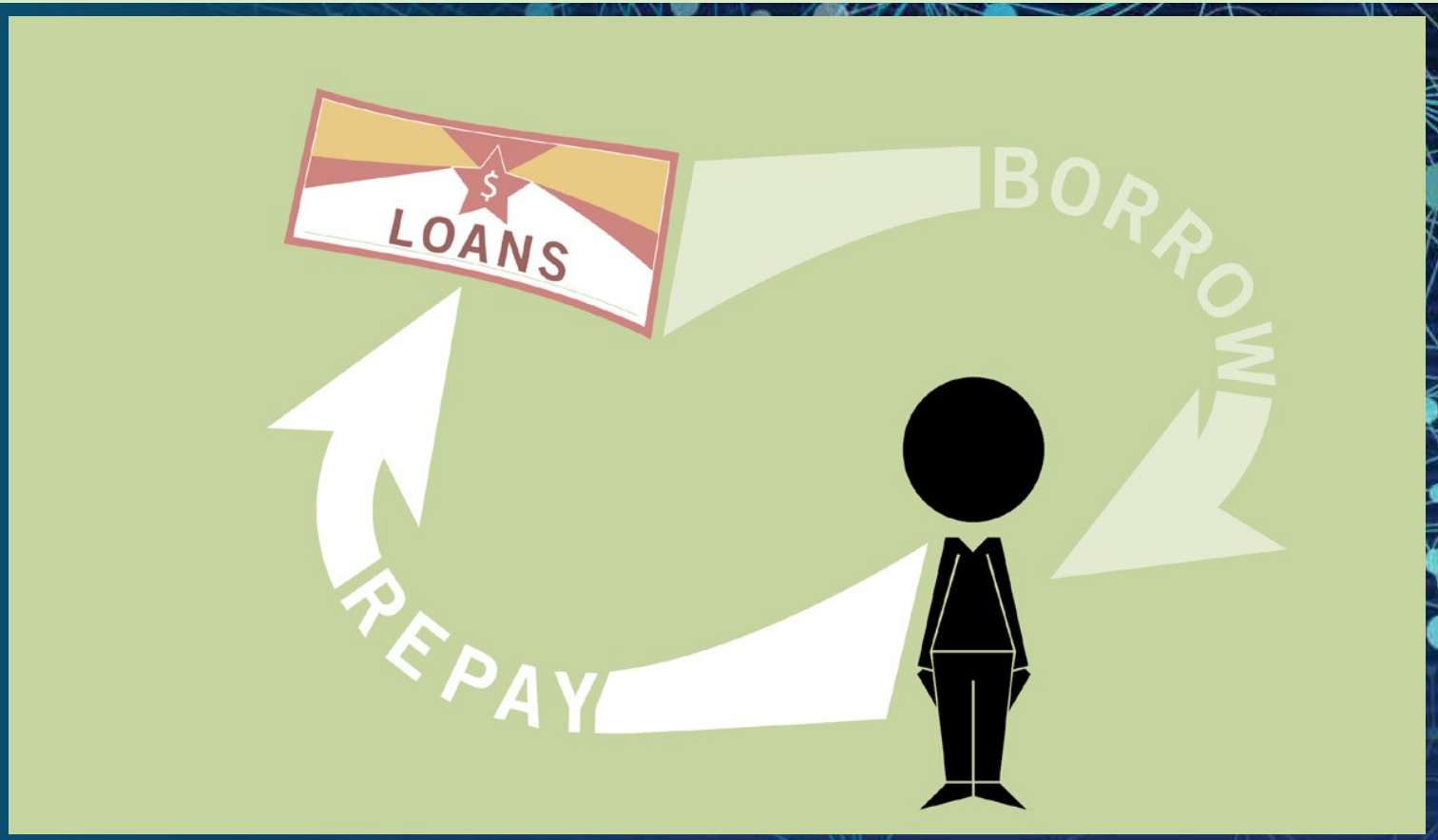
Quality check (AUC)

Result & conclusion



Problem Statement

A dataset of historical loans, along with clients' socioeconomic and financial information, our task is to build a model that can predict whether the client will repay or defaulting on a loan.



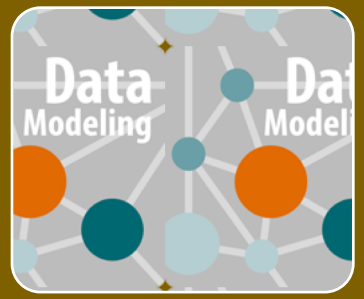
Project Analysis Objective

- Predict the clients' repayment abilities.
- Using various statistical and machine learning methods to make the best predictions.

Project Introduction

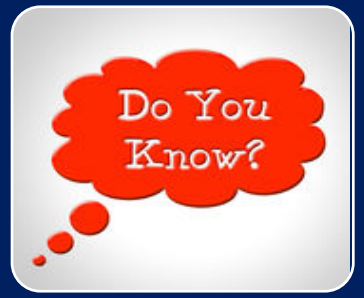
Data and Data Source

- Data taken from Kaggle
- Test data – 121 variables, 48,777 instances
- Train data – 122 variables, 307,511 instances
- Final outcome – categorical -> Target (0 or 1)



Modeling Techniques

- Random Forest



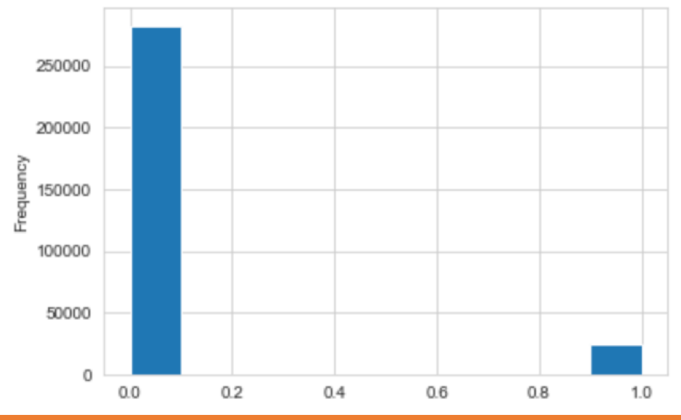
Terms and Business Significance Discussion

- What is home credit default risk?

Exploratory Data Analysis - 1

Target Variable

```
Train_data['TARGET'].plot.hist();
```




```
Train_data['TARGET'].value_counts()
```

```
0    282686
1     24825
Name: TARGET, dtype: int64
```

Missing Values in TRAIN dataset

```
missing_values = mis_values(Train_data)
```

```
Your selected dataframe has 122 columns.
There are 67 cols that have missing values.
```

	Missing Values	% of Total Missing Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4
NONLIVINGAPARTMENTS_AVG	213514	69.4
FONDKAPREMONT_MODE	210295	68.4
LIVINGAPARTMENTS_MODE	210199	68.4
LIVINGAPARTMENTS_MEDI	210199	68.4
LIVINGAPARTMENTS_AVG	210199	68.4
FLOORSMIN_MODE	208642	67.8
FLOORSMIN_MEDI	208642	67.8
FLOORSMIN_AVG	208642	67.8
YEARS_BUILD_MODE	204488	66.5
YEARS_BUILD_MEDI	204488	66.5
YEARS_BUILD_AVG	204488	66.5
OWN_CAR_AGE	202929	66.0
LANDAREA_AVG	182590	59.4
LANDAREA_MEDI	182590	59.4
LANDAREA_MODE	182590	59.4
BASEMENTAREA_MEDI	179943	58.5
BASEMENTAREA_AVG	179943	58.5
BASEMENTAREA_MODE	179943	58.5
EXT_SOURCE_1	173378	56.4
NONLIVINGAREA_MEDI	169682	55.2
NONLIVINGAREA_MODE	169682	55.2
NONLIVINGAREA_AVG	169682	55.2
ELEVATORS_MEDI	163891	53.3
ELEVATORS_MODE	163891	53.3

Missing Values in TEST dataset

```
missing_values_test = mis_values_test(Test_data)
```

```
Your selected dataframe has 121 columns.
There are 64 cols that have missing values.
```

	Missing Values	% of Total Missing Values
COMMONAREA_MODE	33495	68.7
COMMONAREA_MEDI	33495	68.7
COMMONAREA_AVG	33495	68.7
NONLIVINGAPARTMENTS_MEDI	33347	68.4
NONLIVINGAPARTMENTS_AVG	33347	68.4
NONLIVINGAPARTMENTS_MODE	33347	68.4
FONDKAPREMONT_MODE	32797	67.3
LIVINGAPARTMENTS_MODE	32780	67.2
LIVINGAPARTMENTS_MEDI	32780	67.2
LIVINGAPARTMENTS_AVG	32780	67.2
FLOORSMIN_MEDI	32466	66.6
FLOORSMIN_MODE	32466	66.6
FLOORSMIN_AVG	32466	66.6
OWN_CAR_AGE	32312	66.3
YEARS_BUILD_AVG	31818	65.3
YEARS_BUILD_MEDI	31818	65.3
YEARS_BUILD_MODE	31818	65.3
LANDAREA_MODE	28254	58.0
LANDAREA_AVG	28254	58.0
LANDAREA_MEDI	28254	58.0
BASEMENTAREA_MEDI	27641	56.7
BASEMENTAREA_AVG	27641	56.7
BASEMENTAREA_MODE	27641	56.7
NONLIVINGAREA_MEDI	26084	53.5
NONLIVINGAREA_MODE	26084	53.5
NONLIVINGAREA_AVG	26084	53.5
ELEVATORS_MEDI	25189	51.7
ELEVATORS_MODE	25189	51.7
ELEVATORS_AVG	25189	51.7

Exploratory Data Analysis - 2

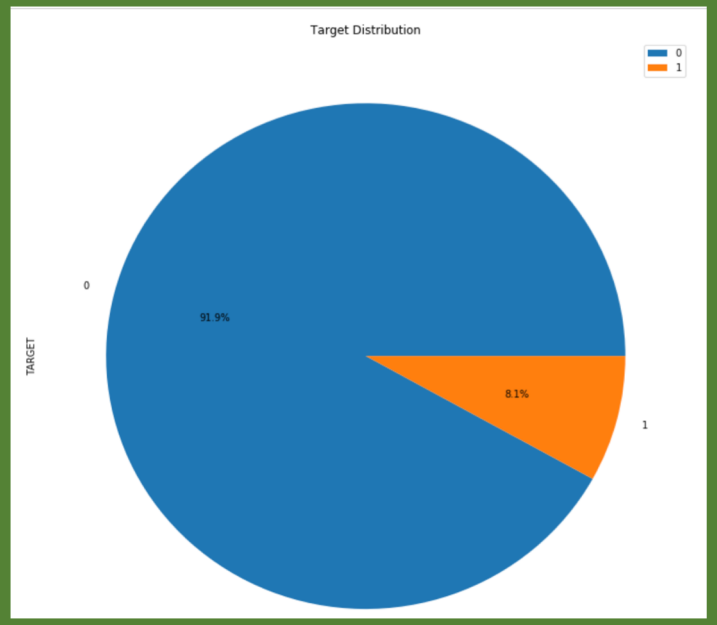
```
Train_data.dtypes.value_counts()
```

```
float64    65
int64      41
object     16
dtype: int64
```

```
Train_data.TARGET.value_counts(normalize=True)
```

```
0    0.919271
1    0.080729
Name: TARGET, dtype: float64
```

Target Distribution

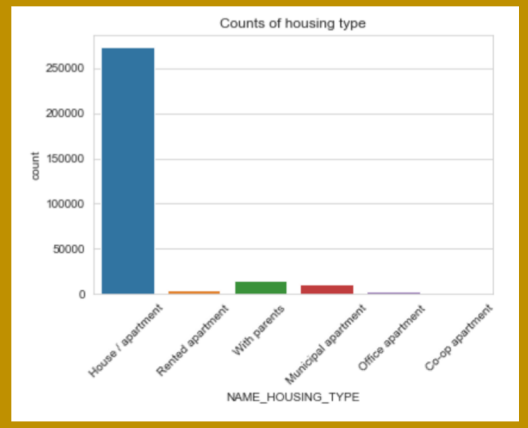


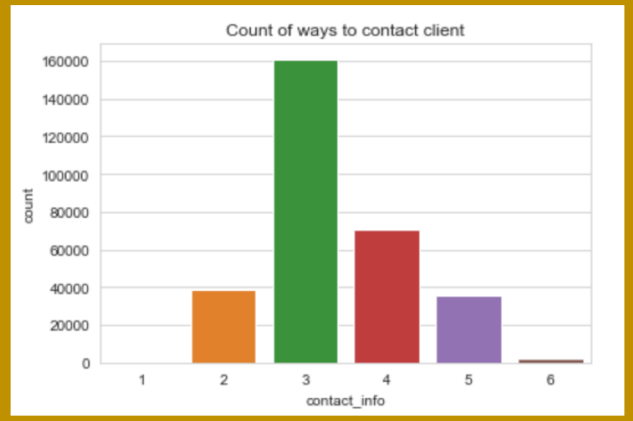
CODE_GENDER	F	M	XNA
TARGET			
0	188278	94404	4
1	14170	10655	0

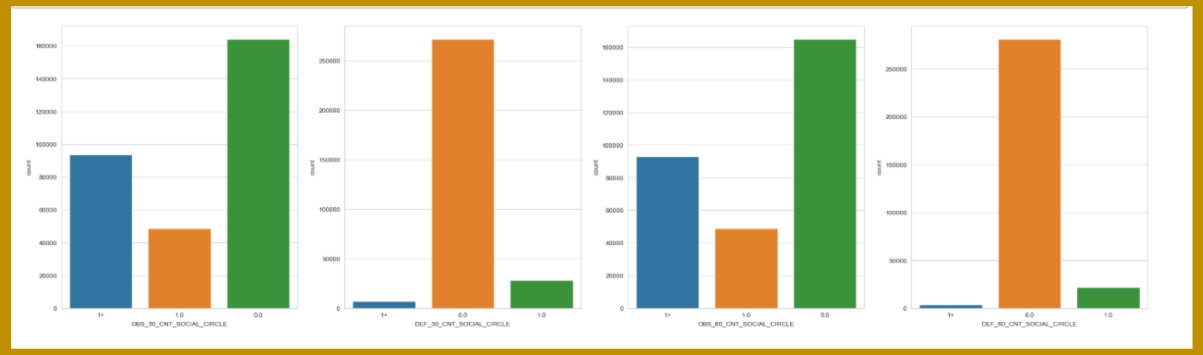
NAME_CONTRACT_TYPE	Cash loans	Revolving loans
TARGET		
0	0.829274	0.089997
1	0.075513	0.005216

Exploratory Data Analysis - 3

Histograms:

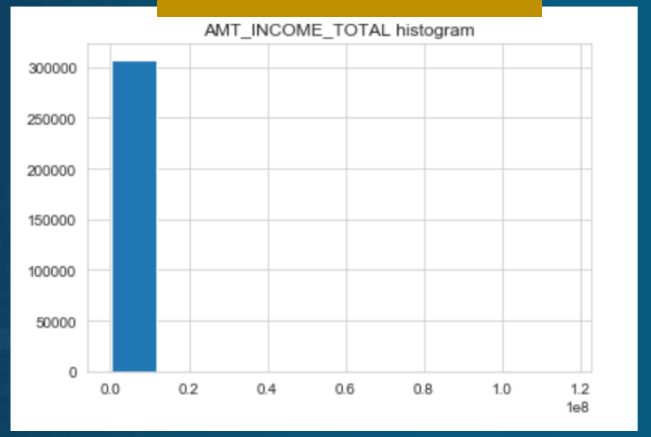




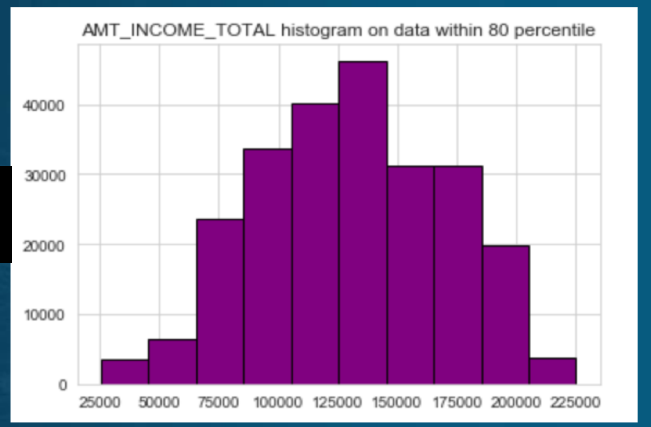


Total income

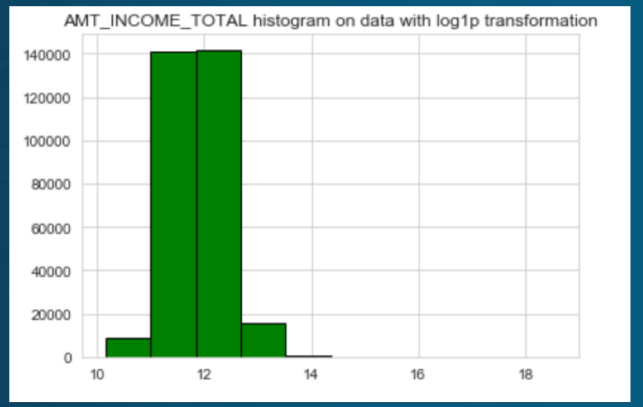
Original



80 percentile

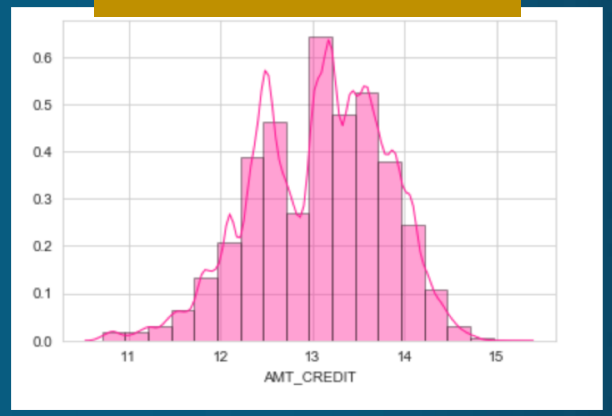


Log



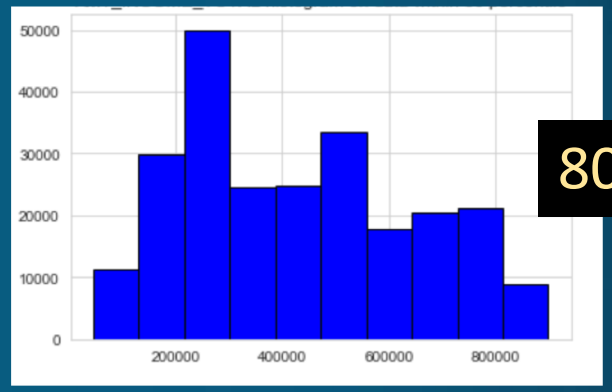
Exploratory Data Analysis - 4

Amount Credit

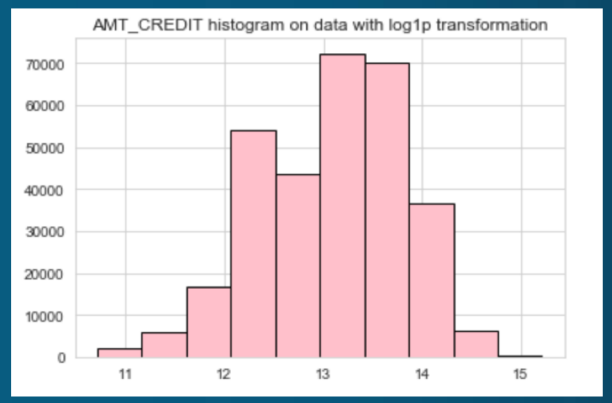


Original

	AMT_CREDIT		
	mean	median	count
TARGET			
0	602648.282002	517788.0	282686
1	557778.527674	497520.0	24825

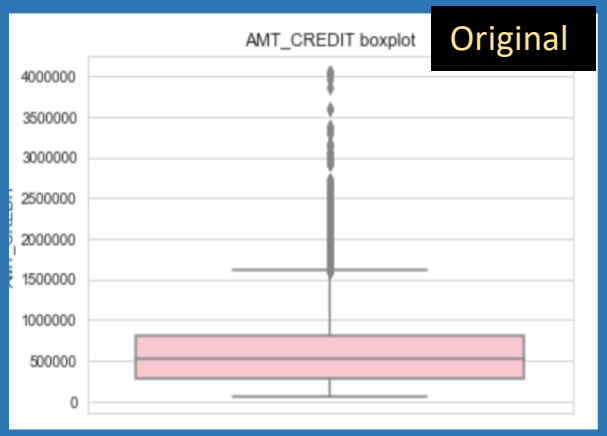


80 percentile

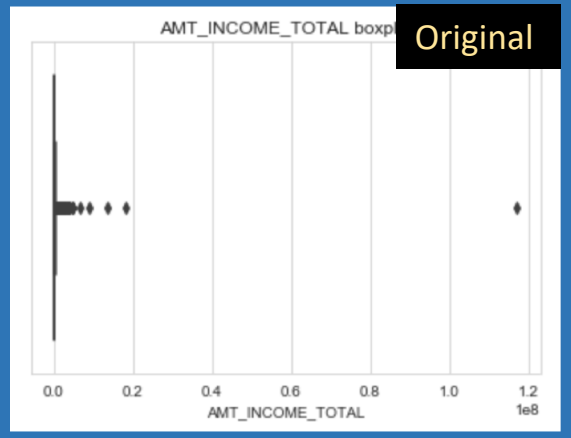


Log

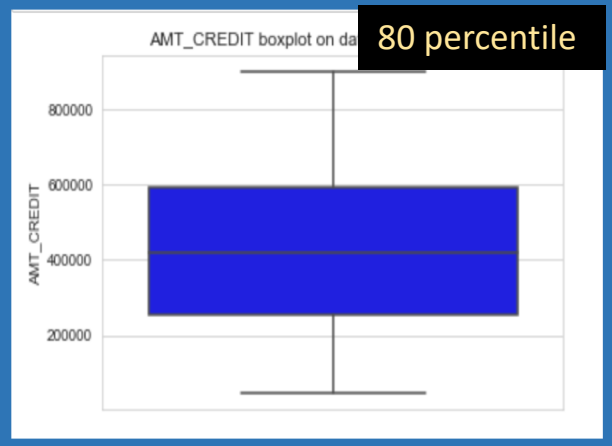
Box plots:



Original

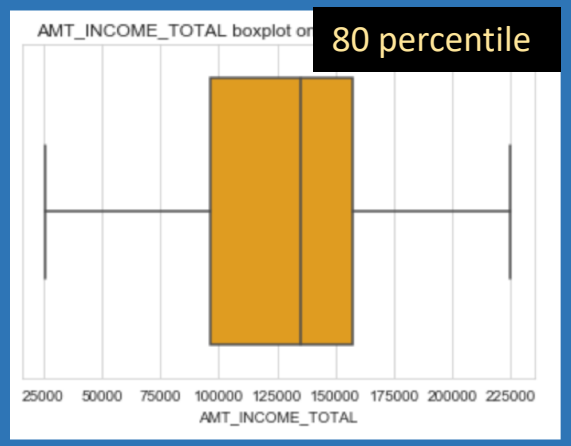


Original



80 percentile

Amount Credit

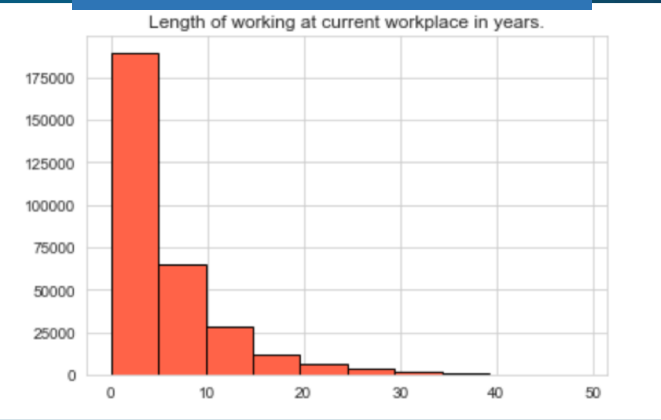


80 percentile

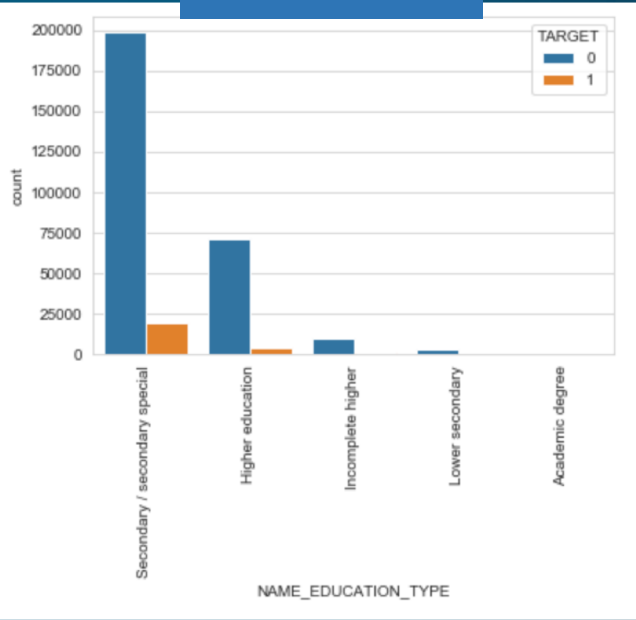
Amount Income

Exploratory Data Analysis - 5

Length of working

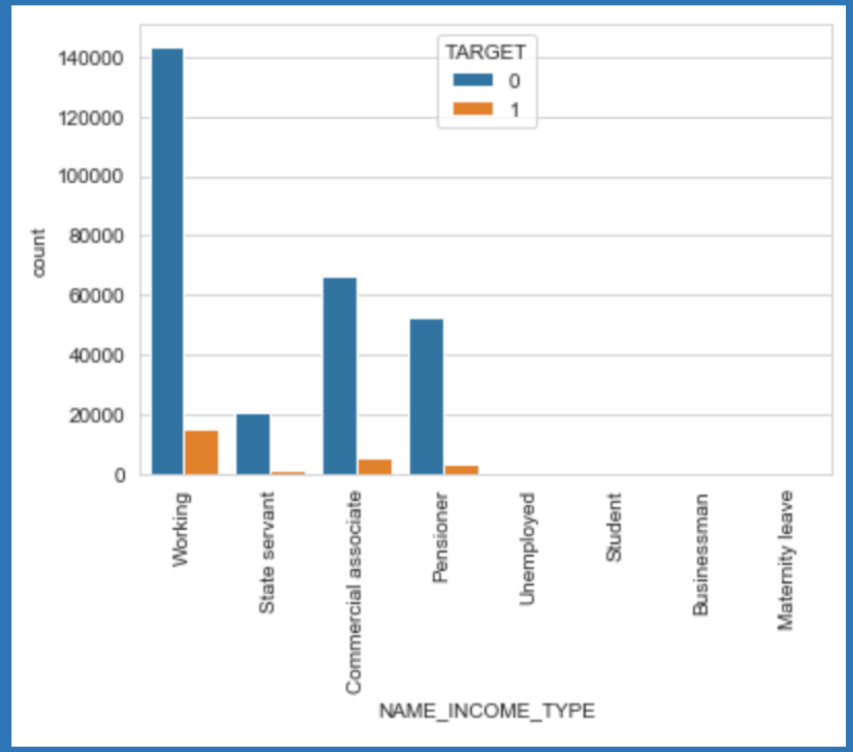


Education



Income Type

	years_employed				age
	mean	median	count	max	median
NAME_INCOME_TYPE					
Businessman	7.874795	6.280822	10	14.917808	47.498630
Commercial associate	5.833627	4.147945	71617	48.071233	39.780822
Maternity leave	7.224110	8.273973	5	10.306849	39.350685
Pensioner	0.002448	-0.000000	55362	35.246575	60.413699
State servant	9.454441	7.205479	21703	48.172603	40.693151
Student	6.835160	5.069863	18	21.931507	36.663014
Unemployed	0.000000	-0.000000	22	-0.000000	45.860274
Working	6.446909	4.427397	158774	49.073973	39.876712

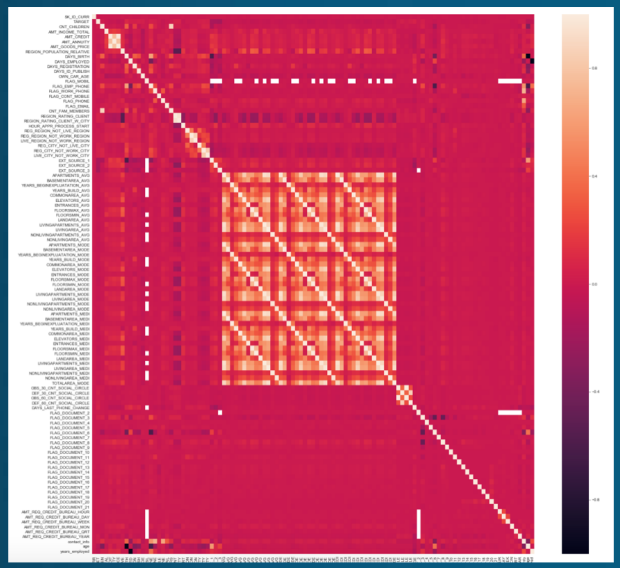


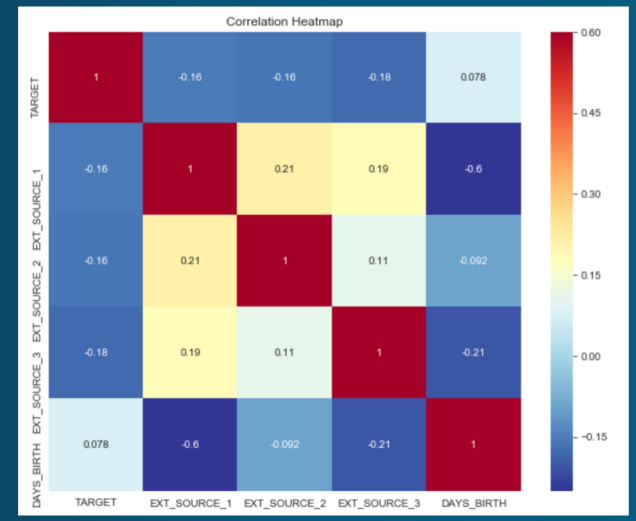
Exploratory Data Analysis - 6

Correlation (Top 15)

```
Most Positive Correlations:
 FLAG_WORK_PHONE                0.028524
LIVE_CITY_NOT_WORK_CITY         0.032518
OWN_CAR_AGE                     0.037612
DAYS_REGISTRATION               0.041975
FLAG_DOCUMENT_3                 0.044346
REG_CITY_NOT_LIVE_CITY          0.044395
FLAG_EMP_PHONE                  0.045982
DAYS_EMPLOYED                   0.046052
REG_CITY_NOT_WORK_CITY          0.050994
DAYS_ID_PUBLISH                 0.051457
DAYS_LAST_PHONE_CHANGE          0.055218
REGION_RATING_CLIENT            0.058899
REGION_RATING_CLIENT_W_CITY     0.060893
DAYS_BIRTH                      0.078239
TARGET                          1.000000
Name: TARGET, dtype: float64
```

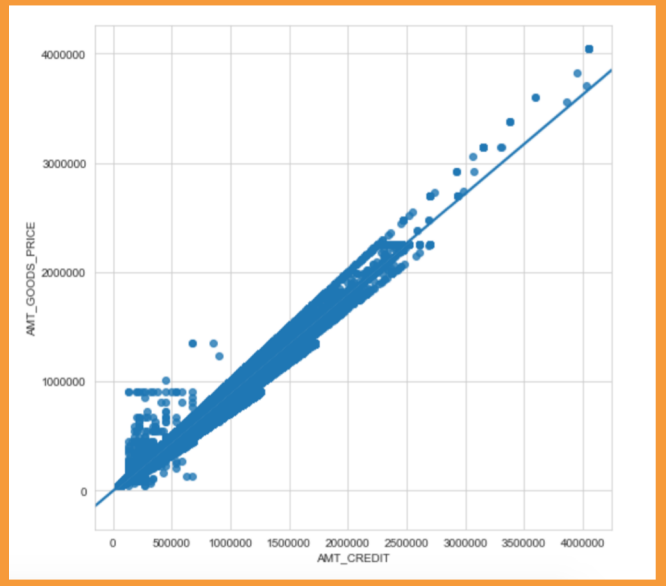
```
Most Negative Correlations:
 EXT_SOURCE_3                  -0.178919
EXT_SOURCE_2                   -0.160472
EXT_SOURCE_1                   -0.155317
age                            -0.078239
years_employed                 -0.046052
FLOORSMAX_AVG                  -0.044003
FLOORSMAX_MEDI                 -0.043768
FLOORSMAX_MODE                 -0.043226
AMT_GOODS_PRICE                -0.039645
REGION_POPULATION_RELATIVE     -0.037227
ELEVATORS_AVG                  -0.034199
ELEVATORS_MEDI                 -0.033863
FLOORSMIN_AVG                  -0.033614
FLOORSMIN_MEDI                 -0.033394
LIVINGAREA_AVG                 -0.032997
Name: TARGET, dtype: float64
```

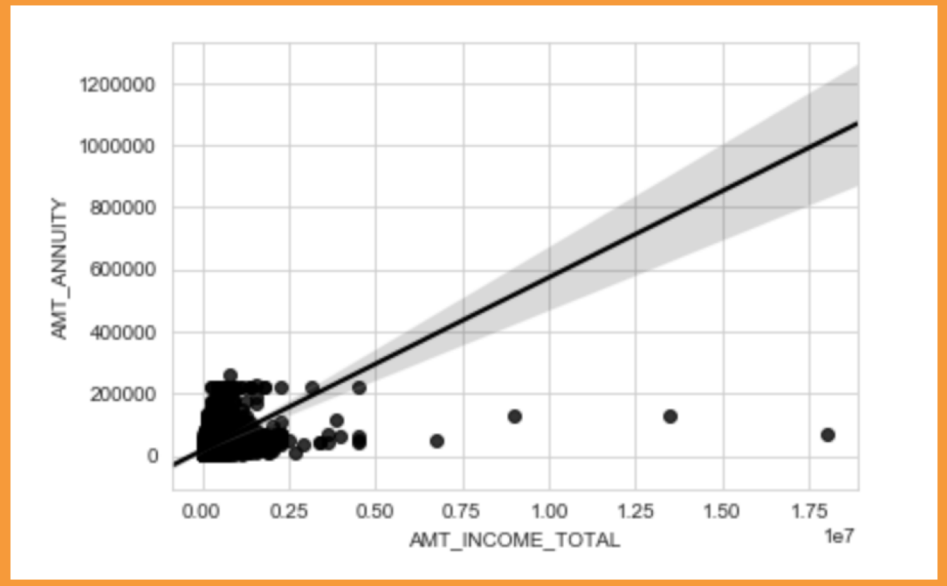


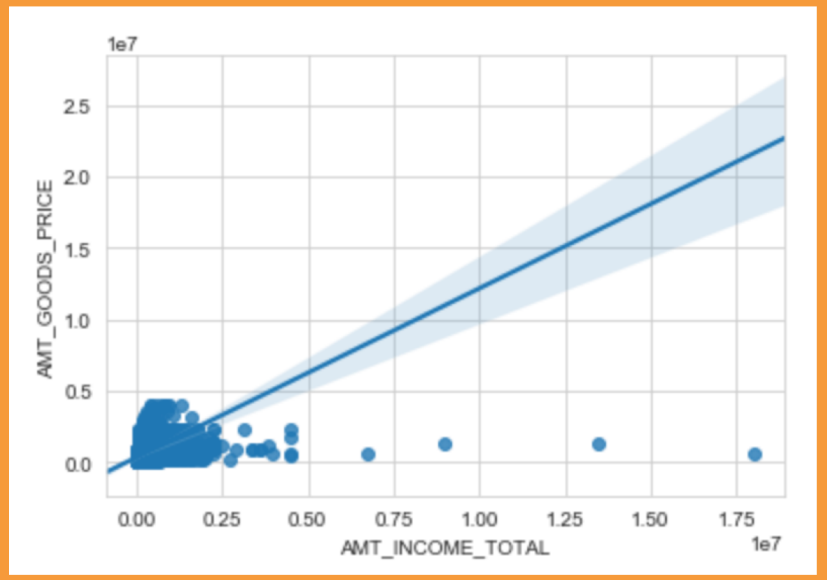


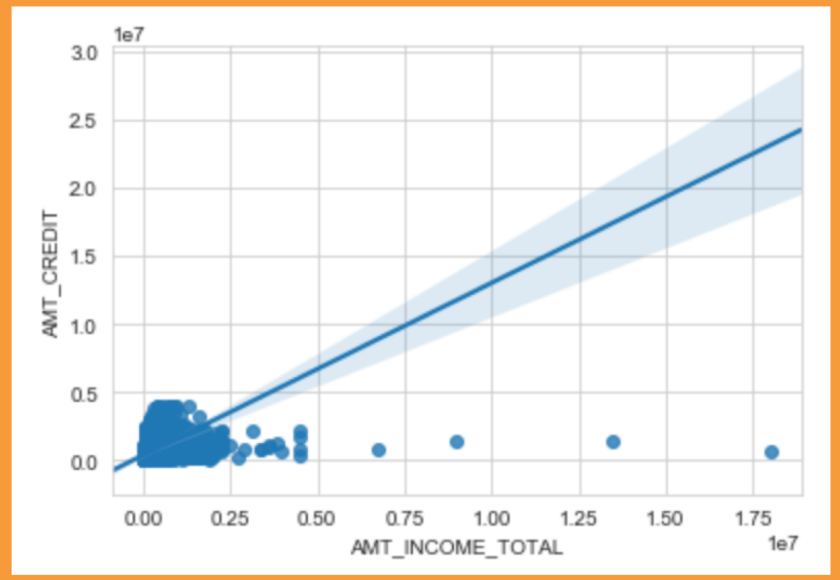
Exploratory Data Analysis - 7

Linear regression









Feature Engineering

Missing values (cut off = 40%)

```
# Identify missing values above threshold
train_missing = train_missing.index[train_missing > 0.4]
test_missing = test_missing.index[test_missing > 0.4]

all_missing = list(set(set(train_missing) | set(test_missing)))
print('There are %d columns with more than 40%% missing values' % len(all_missing))
```

```
There are 45 columns with more than 40% missing values
```

```
Train_data = Train_data.drop(columns = to_drop)
print('Training shape: ', Train_data.shape)
```

```
Training shape:  (307511, 203)
```

Feature Engineering - 2

Correlation (cut off = 0.8)

```
# Select columns with correlations above threshold
to_drop = [column for column in upper.columns if any(upper[column] > threshold)]

print('There are %d columns to remove.' % (len(to_drop)))
```

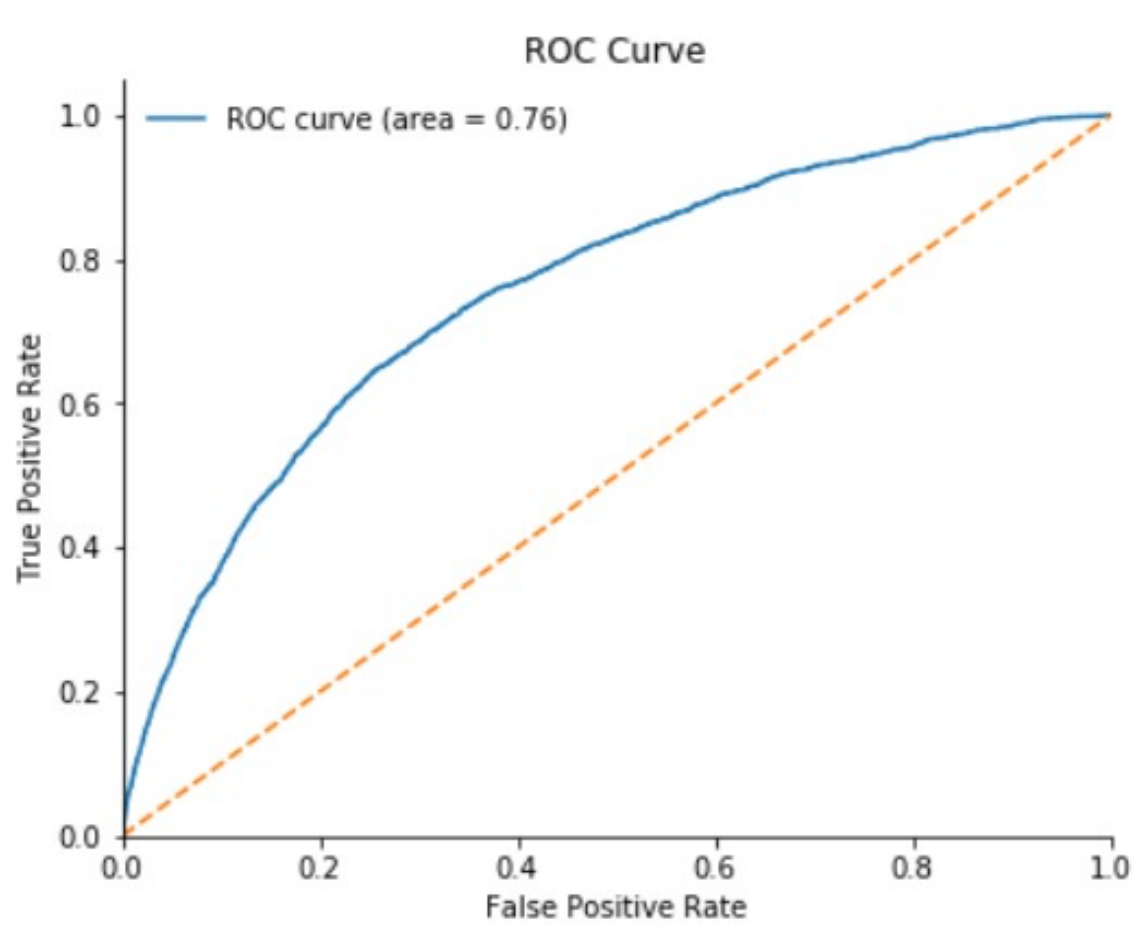
```
There are 49 columns to remove.
```

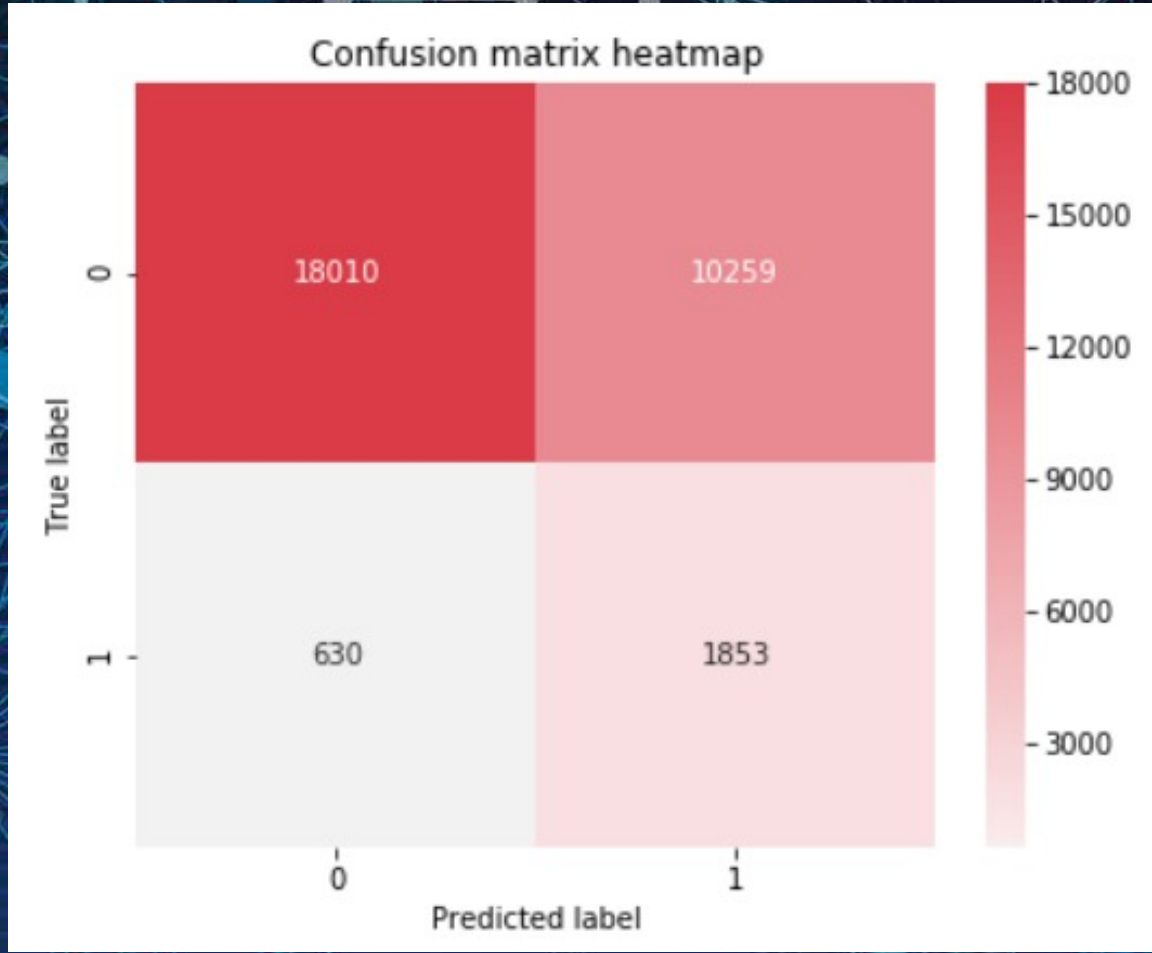
```
# Threshold for removing correlated variables
threshold = 0.8

# Absolute value correlation matrix
corr_matrix = Train_data.corr().abs()
corr_matrix.head()
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AM
SK_ID_CURR	1.000000	0.002108	0.001654	0.001216	0.000703	0.001129	0.001820	
TARGET	0.002108	1.000000	0.030896	0.021851	0.006148	0.019187	0.003982	
NAME_CONTRACT_TYPE	0.001654	0.030896	1.000000	0.004022	0.067177	0.029998	0.003531	
FLAG_OWN_CAR	0.001216	0.021851	0.004022	1.000000	0.002817	0.102023	0.083383	
FLAG_OWN_REALTY	0.000703	0.006148	0.067177	0.002817	1.000000	0.002366	0.002934	

5 rows × 252 columns

Modeling


ROC Curve
ROC curve (area = 0.76)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Confusion matrix heatmap
18010
10259
630
1853
True label
Predicted label
0
1
18000
15000
12000
9000
6000
3000

Result & Conclusion

We can predict the model with accuracy of 64%, which can help bank to know the threshold point of risk for home default loan.

We can estimate accuracy of the model using confusion metric and ROC curve, which area is 76%.

Information that could be directly used by bank:
The clients who are less likely to repay the loan, maybe they should be provided more guidance or financial planning tips. This does not mean that bank should not give loan to those clients having defaulted loans but it should be smart to take precautionary measures to help clients pay on time.

References

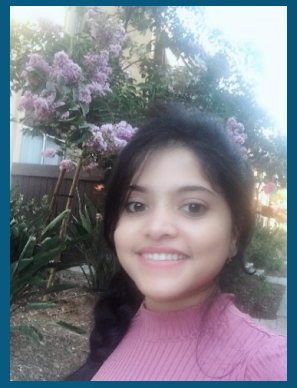
- Home Credit Group,2018. Home Credit Default Risk. https://www.kaggle.com/c/home-credit-default-risk
- Brownlee, J. 2017. How to Handle Missing Data with Python. https://machinelearningmastery.com/handle-missing-data-python/
- Nair, D. 2018. Part II: Manual Feature Engineering techniques for the Kaggle Home Credit Default Competition. https://medium.com/comet-ml/manual-feature-engineering-kaggle-home-credit-db1362d683c4
- Images:

https://goo.gl/images/7PmrLS

https://goo.gl/images/nxwBHC

https://goo.gl/images/MRgJps

Project Team



Shashi Bala
MS – Data Analytics
2018-19

Kornkanok Somkul
MS – Data Analytics
2018-19

Thank You