Proposal Towards Udactiy Machine Learning Advanced Nanodegree

Home Credit Default Analysis

Monimoy Deb Purkayastha (monimoyd@gmail.com)

1 Project Proposal

For financial institutes like banks giving loan to customers is a complicated process. Bank want to ensure that it gives loans to those customers who have low risk. If the customer defaults in repaying loans it will be a loss to Bank. That is why Banks perform extensive credit risk analysis before approving the loan to customer.

In this capstone project I have chosen Kaggle competition challenge “Home Credit Default Analysis” where I also participated in the competition. The URL for the competition is: https://www.kaggle.com/c/home-credit-default-risk

Home Credit is a global financial institute which provides loans to lender. Home Credit operates in 10 countries globally

Home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. In order to make sure this underserved population has a positive loan experience, Home Credit makes use of a variety of alternative data--including telco and transactional information--to predict their clients' repayment abilities.

While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful.

I have analysed the data provided by Home Credit data using statistical techniques. Next I have applied Machine Learning techniques to predict risk of each customer

2 Domain Background

This project is from Financial domain where financial institute Home Credit which is global company has given to Kagglers to competition challenge “Home Credit Default Analysis”

to predict their clients' repayment abilities. This is a important problem to solve as it will:
i. benefit the deserving customer to fulfil their dream of getting dream Home

ii. Home Credit to ensure that they are giving the loans to those customers who will repay the loan in time

iii. Myself as a participant to get the benefit of exposure to real life problem by applying knowledge gathered during Udacity Nanodegree

3 Problem Statement

The goal is to analyze the data provided by Home Credit and create a machine learning model that will give the risk associated with the customer applying for loan. The main steps involved are

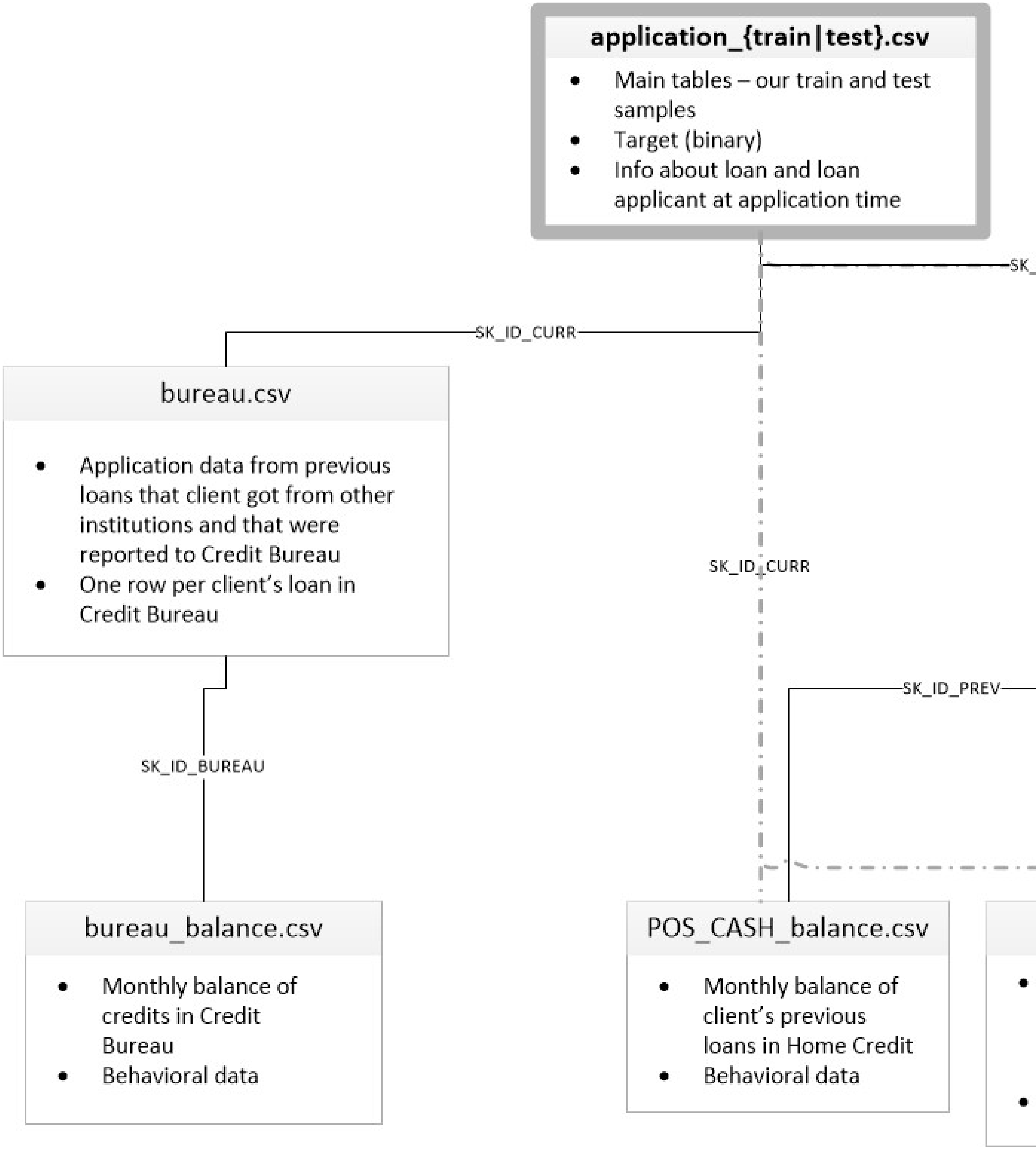
i. Process and merge datasets and create a new dataset which can be processed
ii. Fill up missing values in data
iii. Cleanup the data which is not needed
iv. Univariate analysis of data by trying different visualization graphs (cross tab bar graphs, violin graph, heatmap of linear correlation coefficients)
v. Perform transformation of data using log transformation, normalization, one-hot encoding
vi. Try different machine learning algorithms on transformed training dataframes
vii. Choose the best model which has best ROC-AUC score but which also satisfies all the requirements (like it should be capable of generating actual probability)
viii. Get the prediction probability on the model using test data and transform to the format expected by Kaggle
ix. Submit to Kaggle and get the submission score

4 Datasets

Dataset is provided in https://www.kaggle.com/c/home-credit-default-risk/data

- **application_{train|test}.csv**
  - This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET).
  - Static data for all applications. One row represents one loan in our data sample.
- **bureau.csv**
  - All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in our sample).

    - For every loan in our sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.
- **bureau_balance.csv**
    - Monthly balances of previous credits in Credit Bureau.
    - This table has one row for each month of history of every previous credit reported to Credit Bureau – i.e the table has (#loans in sample * # of relative previous credits * # of months where we have some history observable for the previous credits) rows.
- **POS_CASH_balance.csv**
    - Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
    - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credits * # of months in which we have some history observable for the previous credits) rows.
- **credit_card_balance.csv**
    - Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
    - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credit cards * # of months where we have some history observable for the previous credit card) rows.
- **previous_application.csv**
    - All previous applications for Home Credit loans of clients who have loans in our sample.
    - There is one row for each previous application related to loans in our data sample.
- **installments_payments.csv**
    - Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
    - There is a) one row for every payment that was made plus b) one row each for missed payment.
    - One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in our sample.
- **HomeCredit_columns_description.csv**
    - This file contains descriptions for the columns in the various data files.


application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
SK_ID_CURR
SK_ID_PREV
SK_ID_BUREAU
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data

.

4. Evaluation Metrics

ROC-AUC score: An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate. **True Positive Rate** (**TPR**) is a synonym for recall and is therefore defined as follows:

recall = (true positives)/(true positives + false negatives)

An ROC curve plots TPR vs. FPR at different classification thresholds..

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two dimensional area underneath the entire ROC curve. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

This Kaggle competition is judged based on ROC-AUC score, so I will be using this

Score for optimization.
However, I will also calculate Accuracy and F-score for completeness and for Comparing models

Accuracy: Accuracy is a common metric for binary classifiers. It takes into account both true positives and true negatives with equal weight.

accuracy = (true positives + true negatives)/dataset size

Precision:

precision = (true positives)/(true positives + false positive)

Recall:

recall = (true positives)/(true positives + false negatives)

F1-score:

F1-score = 2*(precision * recall)/(precision + recall)

Reference:

https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

5. Solution Statement

The solution should maximize the ROC-AUC score for the test data given. For getting solution first we need to pre process the data .

Next, a various analytical, statistical techniques are used to analyse and visualize the data (e.g. crosstab bar diagram, violin

Next, apply some data transformation techniques (like One hot encoding for categorical data, log transformation etc).

Next, apply various machine learning algorithms and train the data using these models and find the best model which maximizes ROC-AUC score and which is capable to generating actual probability score.

Once the model is chosen, the test data is predicted on the model and get the probability scores.

One probability scores are generated, it is merged with `SK_ID_CURR` .
For submission to Kaggle, For each `SK_ID_CURR` in the test set, we have to predict a probability for the `TARGET` variable. The file should contain a header and have the following format:

```
SK_ID_CURR,TARGET
100001,0.1
100005,0.9
100013,0.2
```

6. Project Design:

6.1 Preprocess the data:

In this step I have first all the loaded datasets given:

- application_train.csv into dataframe application_train_data
- application_test.csv into dataframe application_test_data
- bureau.csv into into dataframe bureau
- bureau_balance.csv into dataframe bureau_balance
- POS_CASH_balance.csv into dataframe POS_CASH_balance
- credit_card_balance.csv into dataframe credit_card_balance
- previous_application.csv into dataframe previous_application
- installments_payments.csv into dataframe installments_payments

Next, get separate dataframes from bureau.csv based on values in CREDIT_ACTIVE field which are:

- Active as active_bereau_credit dataframe
- Closed as closed_bereau_credit dataframe
- Sold as sold_bereau_credit dataframe

- Bad debt as bad_debt_bereau_credit dataframe

Next, get total count of each type of CREDIT_ACTIVE group by SK_ID_CUR and load into dataframe bureau_credit_count

Next merge dataframe application_train_data with each of the dataframes active_bereau_credit dataframe, closed_bereau_credit dataframe, sold_bereau_credit dataframe, bad_debt_bereau_credit dataframe by performing left join on field SK_ID_CURR and crea a new dataframe application_bureau_train_data

The above step do with application_train_data with the same set of bereau dataframes and create a new dataframe dataframe application_bureau_test_data

Similarly create separate dataframes for each contract type in previous_application (i.e. Cash Loans, Consumer Loans, Revolving Loans, XNA). Create another dataframe which will have count of number of each type of contract_type in previous_application group by SK_ID_CUR. With application_bureau_train_data merge these dataframes by performing left join on SK_ID_CUR and create new dataframe application_bureau_loan_train_data. Similarly, with application_bureau_test_data merge these dataframes by performing left join on SK_ID_CUR and create new dataframe application_bureau_loan_test_data

The dataframe application_bureau_loan_train_data will be used for further analysis. When any transformations or adding/deleting fields is done on application_bureau_loan_train_data, the same will be applied on application_bureau_test_data, as this is used for Kaggle public ranking

6.3: Missing Data Analysis¶

In this step, we first get which all columns have missing values and then calculate percentage of records which have missing values in each column.

Next find out all the columns whose type is string and fill value 'NA' for all the missing values For remaining misssing values which are numerics, fill value 0

6.4: Analyze the data:

i. Event Rate: First calculate the event rate by dividing number of 1s in TARGET by total number of records. If event rate is low, then the class is imbalanced

ii. Create a cross-tabulation bar plot for the following

NAME_CONTRACT_TYPE vs TARGET
CODE_GENDER vs TARGET
FLAG_OWN_CAR vs TARGET
CNT_CHILDREN vs TARGET
NAME_FAMILY_STATUS vs TARGET
NAME_HOUSING_TYPE vs TARGET
HOUSE_TYPE_MODE vs TARGET
TOTALAREA_MODE vs TARGET
FONDKAPREMONT_MODE vs TARGET

iii. Calculate correlation coefficients and visualize heatmap for the following

TARGET, AMT_INCOME_TOTAL,AMT_CREDIT,AMT_ANNUITY,AMT_GOODS_PRICE
TARGET,DAYS_BIRTH,DAYS_EMPLOYED, DAYS_REGISTRATION, DAYS_ID_PUBLISH, EXT_SOURCE
TARGET,AMT_CREDIT,CASH_LOANS, CONSUMER_LOANS,REVOLVING_LOANS,XNA
TARGET,AMT_CREDIT,Active, Bad_debt,Closed,Sold
TARGET, AMT_CREDIT,AMT_CREDIT_SUM_CLOSED,AMT_CREDIT_SUM_DEBT_CLOSED, AMT_CREDIT_MAX_OVERDUE_CLOSED,CNT_CREDIT_PROLONG_CLOSED, AMT_CREDIT_SUM_LIMIT_CLOSED,AMT_CREDIT_SUM_OVERDUE_CLOSED, AMT_ANNUITY_CLOSED
TARGET,AMT_CREDIT, AMT_CREDIT_SUM_ACTIVE,AMT_CREDIT_SUM_DEBT_ACTIVE, AMT_CREDIT_MAX_OVERDUE_ACTIVE,CNT_CREDIT_PROLONG_ACTIVE AMT_CREDIT_SUM_LIMIT_ACTIVE,AMT_CREDIT_SUM_OVERDUE_ACTIVE,AMT_ANNUITY_ACTIVE'

iii. Perform Violin Plot Visualization for the following:

DAYS_BIRTH vs TARGET
DAYS_EMPLOYED vs TARGET
DAYS_REGISTRATION vs TARGET
EXT_SOURCE vs TARGET
Active vs TARGET
CLOSED vs TARGET
BAD_DEBT vs TARGET

6.5: Data Transformation:

i. **Logarithmic Transformation**: For highly-skewed feature distributions such as AMT_INCOME_TOTAL', 'AMT_CREDIT', logarithmic transformation is done on the data so that the very large and very small values do not negatively affect the performance of a learning algorithm. Using a logarithmic transformation significantly reduces the range of values caused by outliers. Care must be taken when applying this transformation however: The logarithm of 0 is undefined, so we must translate the values by a small amount above 0 to apply the the logarithm successfully.

ii. **Normalizing Numerical Features**

In addition to performing transformations on features that are highly skewed, it is often good practice to perform some type of scaling on numerical features. Applying a scaling to the data does not change the shape of each feature's distribution (such as AMT_INCOME_TOTAL, AMT_CREDIT above); however, normalization ensures that each feature is treated equally when applying supervised learners. Note that once scaling is applied, observing the data in its raw form will no longer have the same original meaning, as exampled below.

iii.One hot encoding for categorical features Categorical variables having more than two possible vlaues are encoded using the one-hot encoding scheme. One-hot encoding creates a "dummy" variable for each possible category of each non-numeric feature. For example, assume someFeature has three possible entries: A, B, or C. We then encode this feature into someFeature_A, someFeature_B and someFeature_C.

iv. Label Encoding: Categorical variables having more than two possible are encoded using Label Encode to have values 0 and 1

v. Drop not relevant fields:

6.6: Cross Validation¶

Split the data into 70% for training and 30% for testing

6.7 Apply Supervised Machine Learning models

The following six supervised learning models that are currently available in scikit-learn are used to train the data:

i. Decision Trees

ii. Logistic Regression

iii. Gaussian Naive Bayes (GaussianNB)

iv. Gradient Boosting

v. XGB Boosting

vi. Random Forest

Once training is done , metrics ROC-AUC score, Accuracy score and F1-score are calculated for each model.

The model which best ROC-AUC score but satisfactory Accuracy score and also capable of generating actual probability using predict_proba will be chosen

6.8: Choose the model

Calculate ROC-AU score, Accuracy Score, F1 score, training time, prediction time. Choose the model which has best ROC-AU score but Accuracy score should be reasonable and model should be capable of generating actual probability using predict_proba

6.8: Prepare the data to submit to Kaggle

Following steps are for preparing data for submission to Kaggle

- Use the predict_proba to the chosen model to get the probabily scores
- Get SK_ID_CURR field of test_data and store as test_data_id
- Merge first column of probability and merge with test_data_id
- Save the dataframe to the CSV file as expected by Kaggle

6.9: Submit to Kaggle and get the submission score

7. Benchmark Model

The benchmark is done based on maximizing ROC-AUC score which can vary from 0 to 1. Kaggle has leadership board which has two kind of score:

i. Public Leadership Score: - The leadership is calculated with approximately 20% of test data

ii. Private Leadership Score: - The leadership is calculated with approximately 80% of test data

The highest score on Public Leadership is 0.81724 and Private Leadership is 0.80570

Below are the metrics results that I got when I applied various machine learning models on the test data

Metric	Decision Tree	Logistics	GaussianNB	XGBBoost	GradientBoost	RandomForest
ROC-AUC (test)	0.537319	0.502448	0.557352	0.501365	0.503695	0.500826
ROC-AUC (300 training)	1.0	0.521739	0.564746	0.5	0.498194	0.956521
Accuracy (test)	0.85338	0.920361	0.325134	0.920545	0.920480	0.920470
Accuracy (300 training)	0.85338	0.926666	0.306667	0.923333	0.92	0.993333
F1 score (test data)	0.149094	0.010771	0.164253	0.005697	0.015830	0.003531

Metric	Decision Tree	Logistics	GaussianNB	XGBBoost	GradientBoost	RandomForest
F1 score(300 training)	1.0	0.083333	0.161290	0.0	0.0	0.954545
Training Time	24.800899	23.82941	1.342295	117.141126	166.722213	28.703508
Prediction Time	0.171837	0.109350	0.623674	0.922175	0.567169	1.050263