Proposal Towards Udacity Machine Learning Advanced Nanodegree

Project: Home Credit Default Analysis Using Machine Learning and Deep Learning

Monimoy Deb Purkayastha

1 Project Proposal

For financial institutes like banks giving loan to customers is a complicated process. Banks 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 and Deep 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.

I have chosen the project for my capstone for following reasons:

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 and process the data and train suing various machine learning/deep learning models and finally choose the model that gives the best performance which can be used for calculating the risk associated with the customer applying for loan.

The solution should maximize the ROC-AUC score for the test data given.

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
```

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. Analysis of data by trying different visualization techniques (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 and deep learning techniques to train the data
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

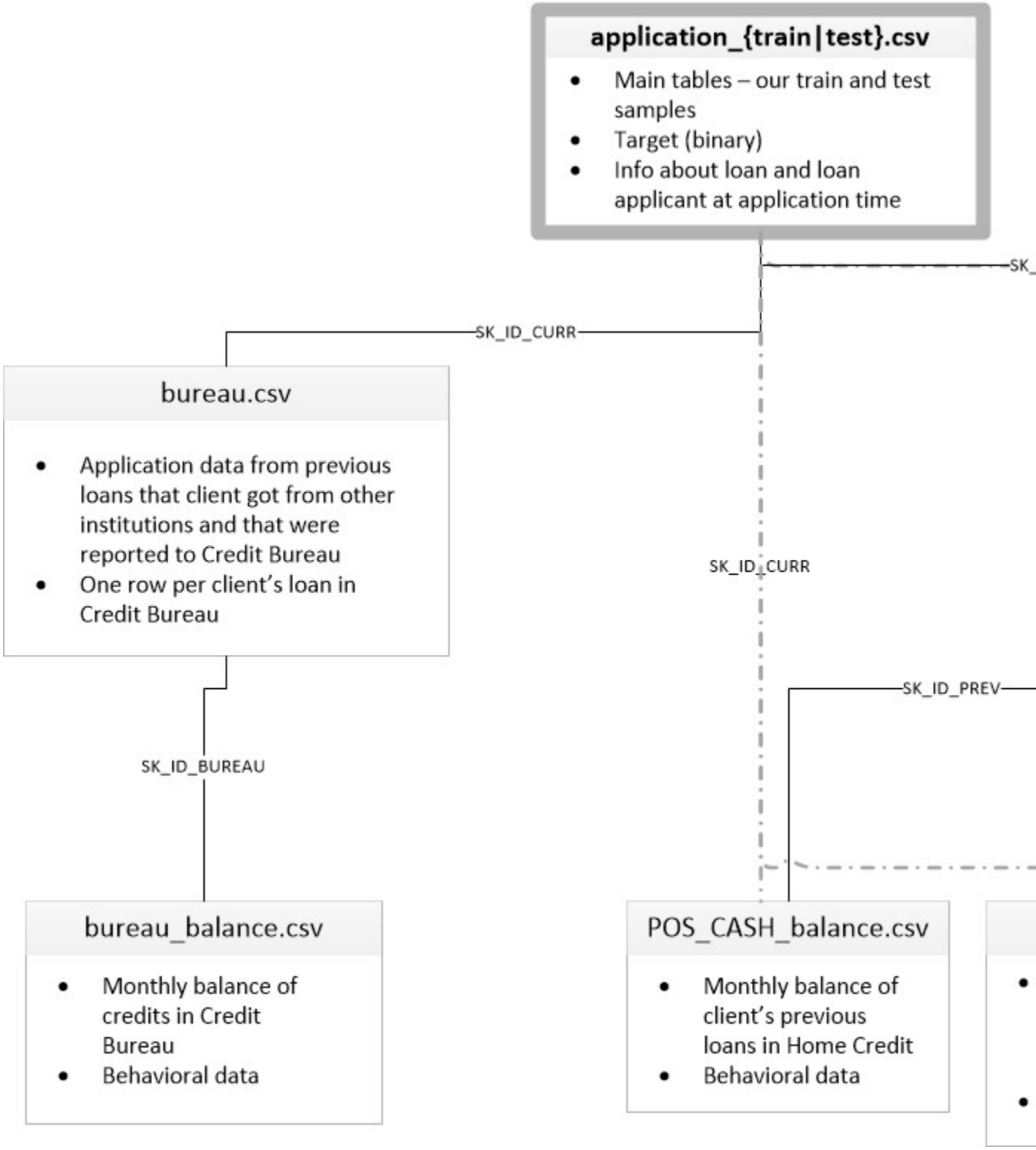
4 Datasets and Inputs

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.



.

4. Evaluation Metrics

This Kaggle competition is judged based on ROC-AUC score, so I have considered ROC-AUC score as the main metric for evaluating various models.
However, I have calculated Accuracy and F-score for completeness and for comparing models.

The definitions of these metrics are given below:

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.

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 plot, heatmap of correlation coefficients)

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

Next, apply various machine learning and deep 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:

Proposed Steps are:

- Load all the datasets (application_train.csv, application_test.csv, bureau.csv, bureau_balance.csv, POS_CASH_balance.csv, credit_card_balance.csv, previous_application.csv, installments_payments.csv) given into pandas data frame
- Consider data frame from dataset application_train.csv as master data for training and application_test.csv as master data for generating test data for Kaggle submission and same field transformations needs
- For each of type of CREDIT_ACTIVE (Active, Closed, Sold, Bad Debt) in bereau.csv group by SK_ID_CUR summarize into separate dataframes, also create another dataframe which count of CREDIT_ACTIVE group by SK_ID_CUR and merge application_train.csv and application_test.csv with these dataframes left join on SK_ID_CUR
- For each of type of NAME_CONTRACT_TYPE ((i.e. Cash Loans, Consumer Loans, Revolving Loans, XNA) in bereau.csv group by SK_ID_CUR summarize into separate dataframes, also create another dataframe which count of NAME_CONTRACT_TYPE group by SK_ID_CUR and merge application_train.csv and application_test.csv with these dataframes left join on SK_ID_CUR
- that installments_payments, POS_CASH_balance, credit_card_balance are mainly transactional purpose, may not be very important. I will not consider these for further analysis:

- After all these operations new frame dataframe application_bureau_loan_train_data for training, dataframe application_bureau_loan_test_data for testing

6.2: 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: Data Exploration:

- **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
- Find the statistics (mean, median, standard deviation) of fields in master data (CNT_CHIDREN, AMT_INCOME_TOAL,AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE) and interpret the results
- Find the statistics related to loan (mean, median, standard deviation) of fields of previous load data and interpret the results

6.5: Exploratory Visualization:

- Crosstab values and bar plot between CODE_GENDER vs TARGET and interpret the results
- Crosstab values and bar plot between NAME_CONTRACT_TYPE vs TARGET and interpret the results
- Crosstab values and bar plot between FLAG_CAR_OWN vs TARGET and interpret the results
- Calculate linear coefficients and heatmap between TARGET, AMT_CREDIT, Active, Closed, Bad_debut, Sold and interpret the results
- Calculate linear coefficients and heatmap between TARGET, AMT_INCOME_TOTAL,AMT_CREDIT,AMT_ANNUITY,AMT_GOODS_PRICE and interpret the plot
- Draw violin plot between DAYS_BIRTH, TARGET and interpret the plot
- Draw violin plot between EXT_SOURCE, TARGET and interpret the plot

6.5: Data Transformation:

- **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.

- **Normalizing Numerical Features:**Perform normalization on numerical features

- **One hot encoding:** For categorical features Categorical variables having more than two possible vlaues are encoded using the one-hot encoding scheme
- **Label Encoding**: Categorical variables having more than two possible are encoded using Label Encode to have values 0 and 1
- **Drop features:** Categorical Drop features which are not relevant

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 is used as criteria to evaluate the model. For completeness and comparing models, 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 best 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

6.9 Refinement

Further refine the parameters of best of six model with GridSearchCV

Try **Deep Neural Network** and calculate the scores and check if score is better than best of 6 models. In case **Deep Neural Network** scores are better, choose it as the final model

6.10: Prepare the data to submit to 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

I need to compare my results against the benchmark scores