Loan Payment Prediction Analysis

By Jiaqi Luo

ANALYSIS OUTLINE

1. Raw data filtering & cleaning
2. Feature encoding & transformation & selection
3. Baseline Model Selection(GradientBoosting & Logistic Regression)
4. Gradient Boosting(GBM) model parameters tuning
5. Assessment data prediction
6. Future work & Improvements

Data Cleaning & Filtering

- Find all missing value representations in the raw data source. For example, ‘-’ and ‘N\A’. Use null value to replace them all.

- Check columns that have over half of the missing values. Such as

`'DIF_INCOME_LAST_180DAY_R', 'DIF_RESPERIOD_LAST_180DAY_R', 'DIF_PERIOD_LAST_180DAY_R',ect`

- Check columns that have the same value over the whole dataset: none.

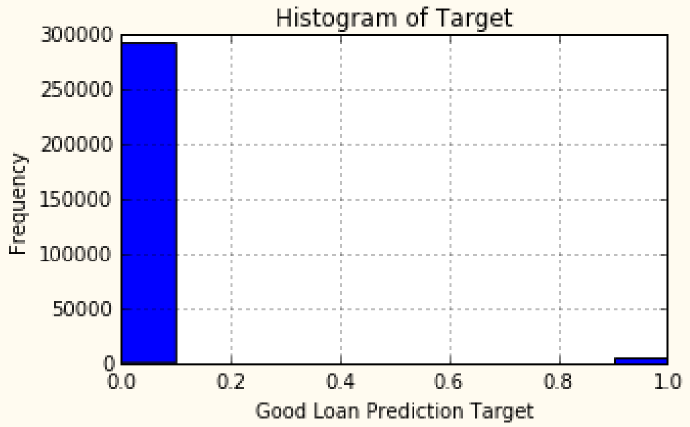
- Drop the above columns with DDATE and ID2 columns. Since in this dataset, I have no idea how to transform the DATE data into a useful feature. If we know the data collection date, we can transform it into how long the customer participated in the loan borrowing system. Also, we don’t need ID columns for training.

- 3. Check the distribution for all features

- 4. Output a csv file for later usage to reduce IO reading time

Potential Problem: Target Data Distribution

- From the TARGET column of raw data, we know that few values are 1.
- Target 1 Percent: 1.812%
- Target 0 Percent: 98.188%
- This means, if we predict 1 for every customer, our prediction accuracy will be 98.188%. Therefore, we should have a model performs better than this accuracy and use AUC number and confusion matrix to evaluate the model.

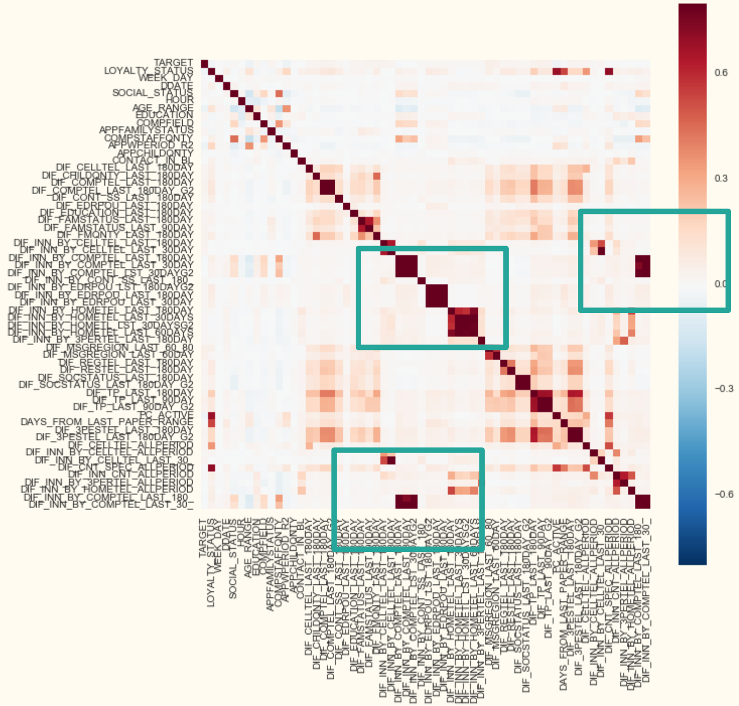


Object Data Numerical & On-Hot Encoding

- For this dataset , most of the features that python read in are considered as Object. For example, “1-3”, “>1” columns.
- Use package LabelEncoder to encode this levels into numerical value (1,2,3,4) by its number range size.
- Categorical features such as [‘SOCIAL_STATUS’, ‘EDUCATION’] are on-hot encoded into dummy variables: SOCIAL_STATUS_1, SOCIAL_STATUS_2,...

Feature Correlation Check

- Check features interactions & correlations. Some of them are highly correlated; thus, drop one feature in a highly correlated pair.



Feature Selection: RFE & KBest Algorithms

- Recursive Feature Elimination(RFE) algorithm was used to select important features among all. 60 out of 80 (including dummy variables) were selected.

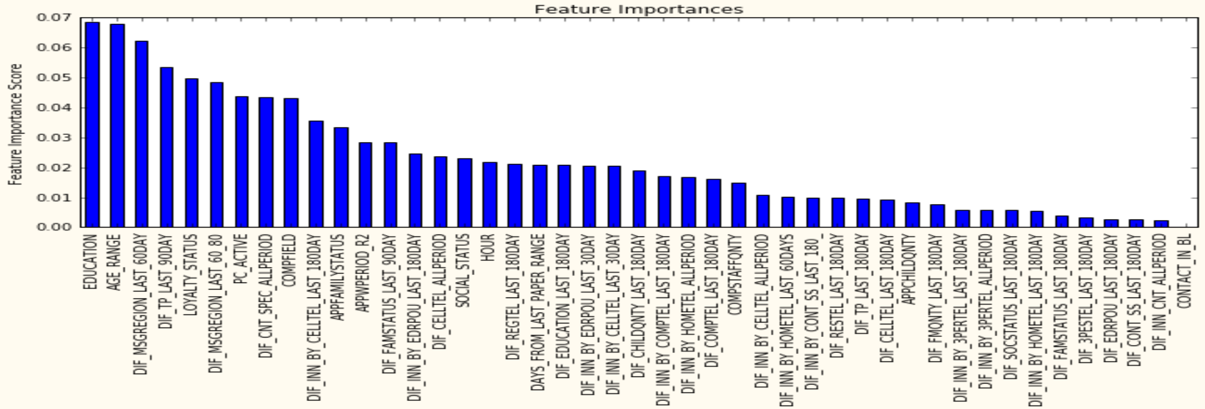
- Univariate selection (Kbest), which uses a set of statistical test to find significant features, were also performed to choose 60 features.

Develop Gradient Boosting Machine(GBM) Baseline Model – Use RFE Features

Baseline model performed on RFE selected features & the feature importance rank

- Accuracy : 0.9897
- AUC Score (Train): 0.798859

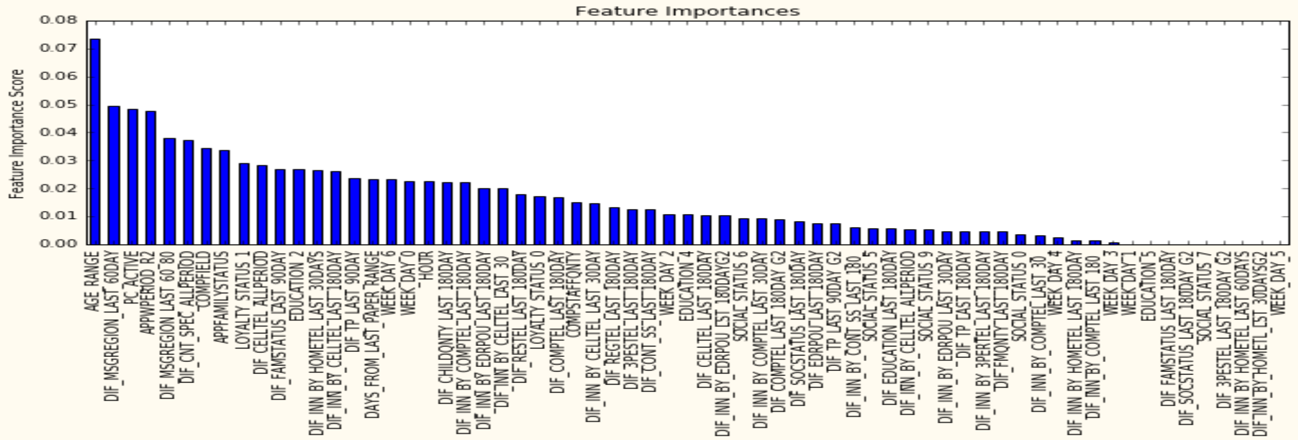
- CV Score : Mean - 0.7708912 | Std - 0.01233622 | Min - 0.7505894 | Max - 0.7859253rmed on RFE selected features.



GBM Baseline Model – Use Kbest Features

Baseline model performed on KBest selected features & the feature importance rank

- Accuracy : 0.9897
- AUC Score (Train): 0.808143
- CV Score : Mean - 0.7838879 | Std - 0.007363273 | Min - 0.7745355 | Max - 0.7954566



GBM Baseline Model -- Use All Features

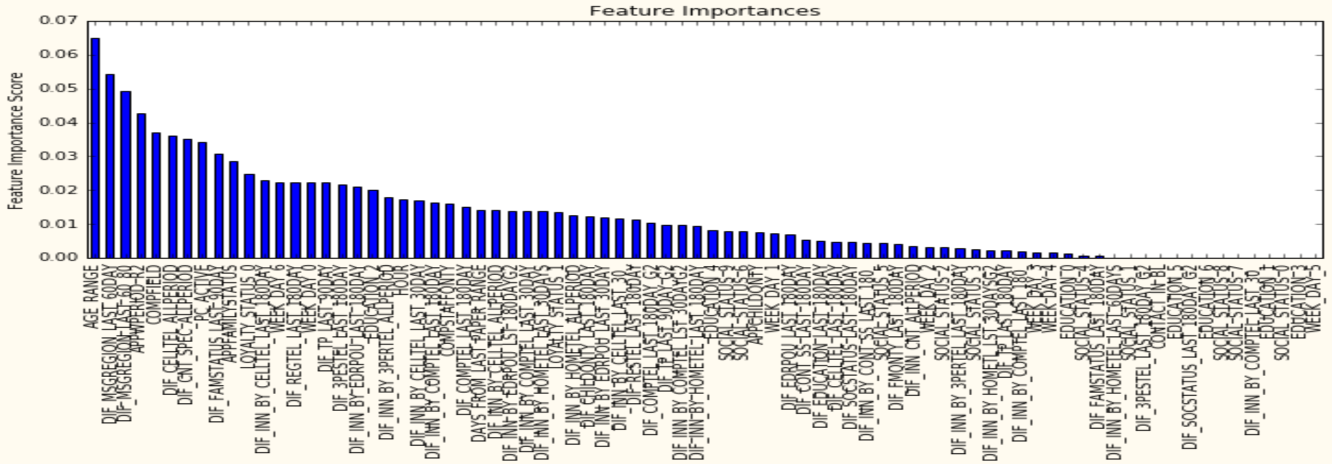
Baseline model performed on ALL features & feature importance rank

Accuracy : 0.9898

AUC Score (Train): 0.810849

All features: Highest AUC Score

CV Score : Mean - 0.7837371 | Std - 0.006790539 | Min - 0.7748344 | Max - 0.7951596



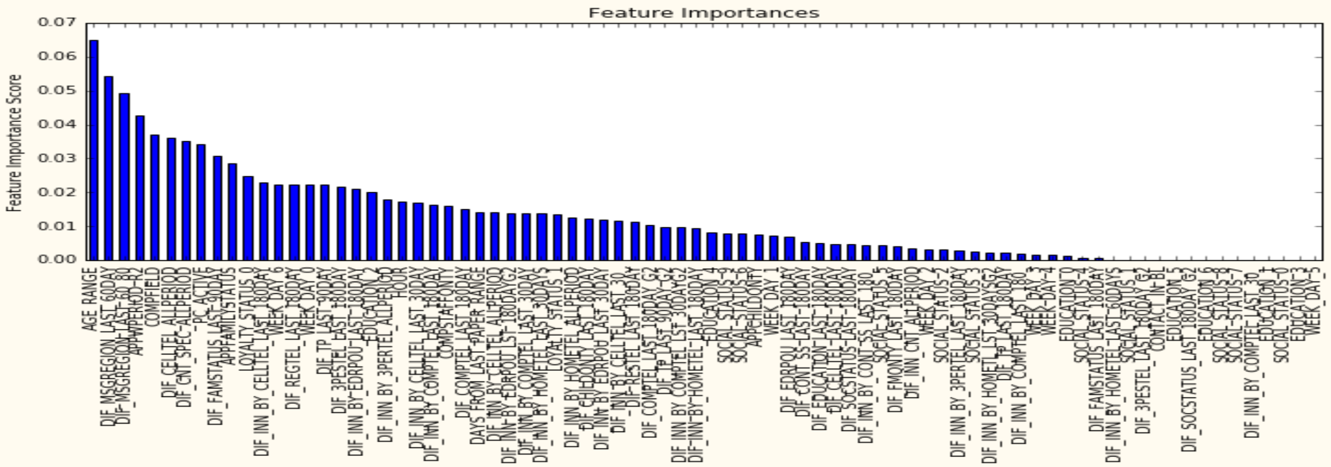
Logistic Regression Baseline Model -- All Features

Model Report

Accuracy : 0.989632605816

AUC Score (Train): 0.763690752416

CV Score : Mean – 0.75759



Since Gradient Boosting (GBM) baseline model performs better than Logistic Regression, we will use GDM to tune the parameters.

GBM Parameter Tuning

- **Step 1, tune the number of estimators**

- Parameters to begin with:

min_samples_split = 500 : ~usually 0.5-1% of total values (will be tuned later)

min_samples_leaf = 50 : small first, preventing over fitting (will be tuned later)

max_depth = 8 : since high number of observations and predictors, choose relatively high value (will be tuned later).

max_features = 'sqrt' : general thumb rule to start with (will be tuned later)

subsample = 0.8 : typically used value (will be tuned later)

learning_rate =0.1: fast at beginning

GBM Parameter Tuning

- **Step 2, tune tree specific parameters as following one by one**
- min_samples_split
- min_samples_leaf
- max_depth
- max_features
- **Step 3, tune subsample size and learning rate.**
- As the learning rate decreases, the subsample size should increase as the times it decreases

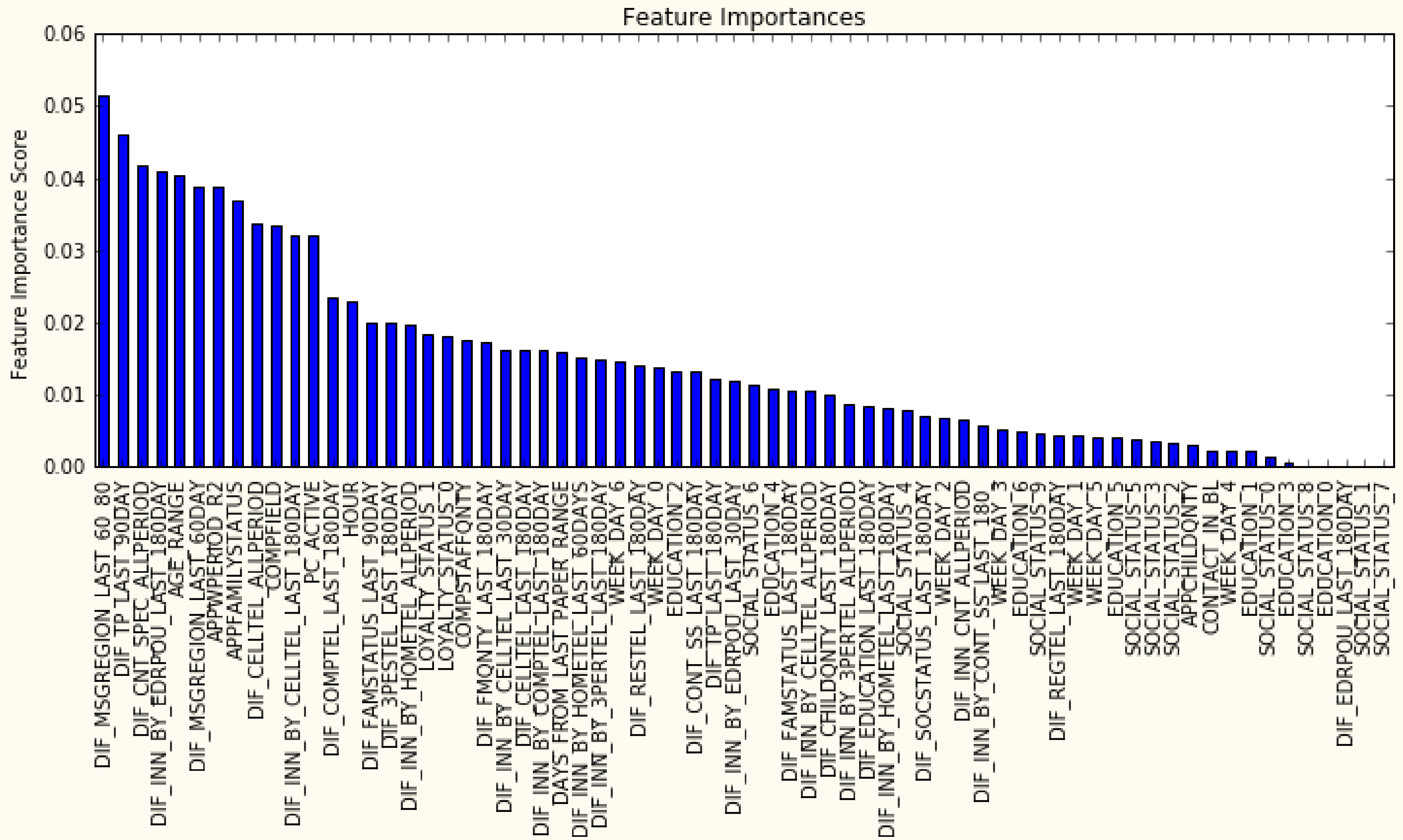
```
CV Score : Mean - 0.7885568 | Std - 0.008759326 | Min - 0.7768986 | Max - 0.8012914
```

GBM Final Tuned Model Performance

model_tuned = GradientBoostingClassifier(learning_rate=0.005, n_estimators=1200, max_depth=7, min_samples_split=1600, min_samples_leaf=50, subsample=0.75, random_state=10, max_features=24)

Model Report

- **Accuracy : 0.9896**
- **AUC Score (Train): 0.839501**
- **CV Score : Mean - 0.7863468 | Std - 0.006260485 | Min - 0.7757783 | Max - 0.7941119**


Feature Importances
Feature Importance Score
0.06
0.05
0.04
0.03
0.02
0.01
0.00
DIF_MSGREGION_LAST_60_80
DIF_TP_LAST_90DAY
DIF_CNT_SPEC_ALLPERIOD
DIF_INN_BY_EDRPOU_LAST_180DAY
AGE_RANGE
DIF_MSGREGION_LAST_60DAY
APPWPERIOD_R2
APPFAMILYSTATUS
DIF_CELLTEL_ALLPERIOD
COMPFIELD
DIF_INN_BY_CELLTEL_LAST_180DAY
PC_ACTIVE
DIF_COMPTEL_LAST_180DAY
HOUR
DIF_FAMSTATUS_LAST_90DAY
DIF_3PESTEL_LAST_180DAY
DIF_INN_BY_HOMETEL_ALLPERIOD
LOYALTY_STATUS_1
LOYALTY_STATUS_0
COMPSTAFFQNTY
DIF_FMQNTY_LAST_180DAY
DIF_INN_BY_CELLTEL_LAST_30DAY
DIF_CELLTEL_LAST_180DAY
DIF_INN_BY_COMPTEL_LAST_180DAY
DAYS_FROM_LAST_PAPER_RANGE
DIF_INN_BY_HOMETEL_LAST_60DAYS
DIF_INN_BY_3PERTEL_LAST_180DAY
WEEK_DAY_6
DIF_RESTEL_LAST_180DAY
WEEK_DAY_0
EDUCATION_2
DIF_CONT_SS_LAST_180DAY
DIF_TP_LAST_180DAY
DIF_INN_BY_EDRPOU_LAST_30DAY
SOCIAL_STATUS_6
EDUCATION_4
DIF_FAMSTATUS_LAST_180DAY
DIF_INN_BY_CELLTEL_ALLPERIOD
DIF_CHILDQNTY_LAST_180DAY
DIF_INN_BY_3PERTEL_ALLPERIOD
DIF_EDUCATION_LAST_180DAY
DIF_INN_BY_HOMETEL_LAST_180DAY
SOCIAL_STATUS_4
DIF_SOCSTATUS_LAST_180DAY
WEEK_DAY_2
DIF_INN_CNT_ALLPERIOD
DIF_INN_BY_CONT_SS_LAST_180
WEEK_DAY_3
EDUCATION_6
SOCIAL_STATUS_9
DIF_REGTEL_LAST_180DAY
WEEK_DAY_1
WEEK_DAY_5
EDUCATION_5
SOCIAL_STATUS_5
SOCIAL_STATUS_3
SOCIAL_STATUS_2
APPCHILDQNTY
CONTACT_IN_BL
WEEK_DAY_4
EDUCATION_1
SOCIAL_STATUS_0
EDUCATION_3
SOCIAL_STATUS_8
EDUCATION_0
DIF_EDRPOU_LAST_180DAY
SOCIAL_STATUS_1
SOCIAL_STATUS_7

GBM Final Tuned Model on Test Dataset (20% of Development Data)

Model Report

- **Accuracy : 0.99**
- **AUC Score (Train): 0.884125**
- **CV Score : Mean - 0.7777595 | Std - 0.02181214 | Min - 0.7457264 | Max - 0.8081906**

Predict the Assessment Dataset by the Tuned Model

- The assessment dataset was cleaned and encoded in the same way as the training dataset.
- Result is shown is the attached csv files
- All code written in python and in the attachment Jupyter Notebook file

Problems & Future Work & Improvements

- For the training dataset, the highest AUC we can reach from this model is 0.839.
- However, All customer was predicted as Target 0 in assessment dataset. In reality, we can assess the customer by the prediction probability or based on probability distribution, etc.
- Also we can improve the model by using higher quality data (more useful features instead of can’t be interpretable ones). Then we can perform feature transformations such as how long a customer borrow the money, how many times the customer paid back and so on.
- Tune the GBDT parameters on larger and higher-quality dataset
- Try other models such as XGBOOST, Random Forest,Nueron network ect.