**RBS Developer Challenge: Machine Learning using Watson Studio**

“Featured Prediction – Loan credit default risk”

Problem statement:

Can you predict how capable each applicant is of repaying a loan?

Bank “ABC” is challenged with loan defaulters. It is struggling with taking a decision on whether the client is capable to repay the loan or not.

Bank is using basic statistical models to understand the pattern in the data collected in the last few years.

Bank needs help in exploring the data to the fullest potential .Doing so will ensure that clients capable of repayment are not rejected and that loans will empower their clients to be successful.

In this challenge, you will help this bank by predicting the probability that a member will default.

Data Information:

Two files will be provided:

1. test\_indessa.csv
2. train\_indessa.csv

Github link - <https://github.com/IBMDevConnect/RBS2018 /> (hackdata)

(Sample data parameters )

| **Variable** | **Description** |
| --- | --- |
| member\_id | unique ID assigned to each member |
| loan\_amnt | loan amount ($) applied by the member |
| funded\_amnt | loan amount ($) sanctioned by the bank |
| funded\_amnt\_inv | loan amount ($) sanctioned by the investors |
| term | term of loan (in months) |
| batch\_enrolled | batch numbers allotted to members |
| int\_rate | interest rate (%) on loan |
| grade | grade assigned by the bank |
| sub\_grade | grade assigned by the bank |
| emp\_title | job / Employer title of member |
| emp\_length | employment length, where 0 means less than one year and 10 means ten or more years |
| home\_ownership | status of home ownership |
| annual\_inc | annual income ($) reported by the member |
| verification\_status | status of income verified by the bank |
| pymnt\_plan | indicates if any payment plan has started against loan |
| desc | loan description provided by member |
| purpose | purpose of loan |
| title | loan title provided by member |
| zip\_code | first three digits of area zipcode of member |
| addr\_state | living state of member |
| dti | ratio of member's total monthly debt repayment excluding mortgage divided by self reported monthly income |
| delinq\_2yrs | number of 30+ days delinquency in past 2 years |
| inq\_last\_6mths | number of inquiries in last 6 months |
| mths\_since\_last\_delinq | number of months since last delinq |
| mths\_since\_last\_record | number of months since last public record |
| open\_acc | number of open credit line in member's credit line |
| pub\_rec | number of derogatory public records |
| revol\_bal | total credit revolving balance |
| revol\_util | amount of credit a member is using relative to revol\_bal |
| total\_acc | total number of credit lines available in members credit line |
| initial\_list\_status | unique listing status of the loan - W(Waiting), F(Forwarded) |
| total\_rec\_int | interest received till date |
| total\_rec\_late\_fee | Late fee received till date |
| recoveries | post charge off gross recovery |
| collection\_recovery\_fee | post charge off collection fee |
| collections\_12\_mths\_ex\_med | number of collections in last 12 months excluding medical collections |
| mths\_since\_last\_major\_derog | months since most recent 90 day or worse rating |
| application\_type | indicates when the member is an individual or joint |
| verification\_status\_joint | indicates if the joint members income was verified by the bank |
| last\_week\_pay | indicates how long (in weeks) a member has paid EMI after batch enrolled |
| acc\_now\_delinq | number of accounts on which the member is delinquent |
| tot\_coll\_amt | total collection amount ever owed |
| tot\_cur\_bal | total current balance of all accounts |
| total\_rev\_hi\_lim | total revolving credit limit |
| loan\_status | status of loan amount, 1 = Defaulter, 0 = Non Defaulters |

**Evaluation:**

* Predicted value
* Feature engineering approach
* AUC-ROC score

**Files to upload:**

1. Source code - python notebook ( ipynb) or any other language based source code
2. Provide “requirements” file to understand the dependent libraries (pip freeze > requirements.txt” in case of python)
3. Home Loan defaulter submission template, Annexure A

**Where to upload:**

Link to be provided by last week of Sept 2018

**Pre-requisite:**

* Registration to IBM cloud ([https://bluemix.net](https://bluemix.net/))

**Skills:**

* Financial domain knowledge
* Statistical programming and basic ML concepts
* Awareness of various ML libraries , frameworks and tools
* Language awareness – python or any other programming language

**Timeline: ( 3 weeks )**

* ML hands on workshop and challenge announcement
  + One day
* Challenge
  + 3 weeks (Sept 28th last date for submission)
* Evaluation
  + 1 week (first week of Oct 2018)

**Collaboration channel:**

* **https://rbshackathon2018.slack.com/**

**Resources:**

* https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction

• https://www.kaggle.com/ogrellier/home-credit-hyperopt-optimization

• https://www.kaggle.com/astrus/entity-embedding-neural-network-keras-lb-0-748

• https://www.kaggle.com/jsaguiar/updated-0-792-lb-lightgbm-with-simple-features

• https://www.kaggle.com/willkoehrsen/automated-model-tuning

• https://www.kaggle.com/willkoehrsen/automated-feature-engineering-basics

**Annexure A**

**Home Loan Defaulter:**

**Team Name : BSquad**

**Team members**: (i) Ramkumar Azhaguramanujam

(ii) Suresh Manoharan

**Solution approach** : We did try two different approaches (using different feature variable selection) and also using different tool set.

**Approach 1:**

Code : Jupyter Notebook [homecreditV2.9.ipynb]

Note : This is done locally and a plain jupyter notebook, runs locally and doesn’t need cloud account

Below are the final features considered after one hot encoding

['loan\_amnt' 'term' 'int\_rate' 'emp\_length' 'dti' 'delinq\_2yrs'

'inq\_last\_6mths' 'open\_acc' 'pub\_rec' 'revol\_util' 'total\_acc'

'collections\_12\_mths\_ex\_med' 'acc\_now\_delinq' 'tot\_coll\_amt'

'tot\_cur\_bal' 'total\_rev\_hi\_lim' 'log\_annual\_inc' 'log\_revol\_bal'

'sub\_grade\_A2' 'sub\_grade\_A3' 'sub\_grade\_A4' 'sub\_grade\_A5'

'sub\_grade\_B1' 'sub\_grade\_B2' 'sub\_grade\_B3' 'sub\_grade\_B4'

'sub\_grade\_B5' 'sub\_grade\_C1' 'sub\_grade\_C2' 'sub\_grade\_C3'

'sub\_grade\_C4' 'sub\_grade\_C5' 'sub\_grade\_D1' 'sub\_grade\_D2'

'sub\_grade\_D3' 'sub\_grade\_D4' 'sub\_grade\_D5' 'sub\_grade\_E1'

'sub\_grade\_E2' 'sub\_grade\_E3' 'sub\_grade\_E4' 'sub\_grade\_E5'

'sub\_grade\_F1' 'sub\_grade\_F2' 'sub\_grade\_F3' 'sub\_grade\_F4'

'sub\_grade\_F5' 'sub\_grade\_G1' 'sub\_grade\_G2' 'sub\_grade\_G3'

'sub\_grade\_G4' 'sub\_grade\_G5' 'home\_ownership\_OTHER' 'home\_ownership\_OWN'

'home\_ownership\_RENT' 'verification\_status\_Source Verified'

'verification\_status\_Verified' 'purpose\_credit\_card'

'purpose\_debt\_consolidation' 'purpose\_educational'

'purpose\_home\_improvement' 'purpose\_house' 'purpose\_major\_purchase'

'purpose\_medical' 'purpose\_moving' 'purpose\_other'

'purpose\_renewable\_energy' 'purpose\_small\_business' 'purpose\_vacation'

'purpose\_wedding' 'initial\_list\_status\_w' 'application\_type\_JOINT']

Categorical variables

* sub\_grade
* home\_ownership
* purpose
* verification\_status
* initial\_list\_status
* application\_type

For numeric features missing values are imputed via Mean

**Before tuning**

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Hyper parameter | Cross Validation ROC AUC | Test ROC AUC |
| RandomForest | model\_\_n\_estimators | 0.7921396257848778 | 0.5045410923860684 |
| Logistic regression with SGD | model\_\_alpha  model\_\_penalty | 0.7446858716596837 | 0.5003908855129359 |
| Light GBM | max\_depth  learning\_rate  num\_leaves  n\_estimators | 0.8455 | 0.640 |
| Logistic Regression | C | 0.7852806250616281 | 0.7852806250616281 |

**After tuning**

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Hyper parameter | Confusion Matrix | Test ROC AUC |
| RandomForest | bootstrap: True  class\_weight:None  criterion: gini  max\_depth: None  max\_features: 0.7  max\_leaf\_nodes: None min\_impurity\_decrease:0.0  min\_impurity\_split:None  min\_samples\_leaf:25  min\_samples\_split:2  min\_weight\_fraction\_leaf:0.0  n\_estimators:60  min\_samples\_split: 2 3 10 | [  [78056 3234]  [15654 9542]  ] | 0.8306171157063302 |
| Logistic regression with SGD | alpha=0.0001  average=False  class\_weight=None  epsilon=0.1  eta0=0.0  fit\_intercept=True  l1\_ratio=0.15  learning\_rate='optimal'  loss='log'  max\_iter=None  n\_iter=1000  n\_jobs=-1  penalty='l2'  power\_t=0.5  random\_state=None  shuffle=True  tol=None | [  [77102 4188]  [19045 6151]  ] | 0.7623 |
| Light GBM | learning\_rate: 0.1  num\_leaves: 63  boosting\_type : gbdt  objective : binary | [  [81290 0]  [25196 0]  ] | 0.5913080723789288 |

**Results:**

1. AUC ROC score : 0.8306 (Best ROC AUC validation test score from Random Forest algorithm )
2. Predicted output for test set : prediction\_test.csv

**What I Learnt :**

Data cleanup and filling missing values using different strategies (For eg Mean/Median)

Encoding of Categorical values (Nominal and Ordinal)

Feature Engineering

Determining correlation between predictor and dependent variable

Different classification Algorithms

About over fitting /under fitting, confusion matix(TP,TN,FP and FN)

Model optimization/tuning via hyper parameters

**What ‘s next**

Focus more on feature engineering, Learn about all the algorithms and idenify it best use. Also need to learn more on deep learning algorithms and corresponding frameworks/tools