Capstone 2 Project :
Credit Card Approval Prediction

Dev Joshi[1]
(1) Spring Board Bootcamp

Problem Statement

The personal information and data submitted by credit-card applicants can be used to decide creditworthiness of the applicants.

The dataset source is :

https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction.

- Machine learning approaches can be applied to automate the approval of credit-card applications
- In this project, we build an automatic credit card approval predictor using machine learning techniques

Datasets

Two datasets :

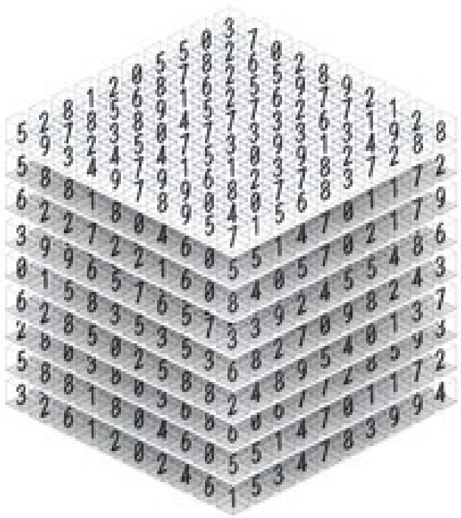
(i) one with the application records, and
(ii) another with the credit-card records.

- The Application record dataset: 438557 rows and 18 columns.
- The credit record dataset:1048575 rows and 3 columns.

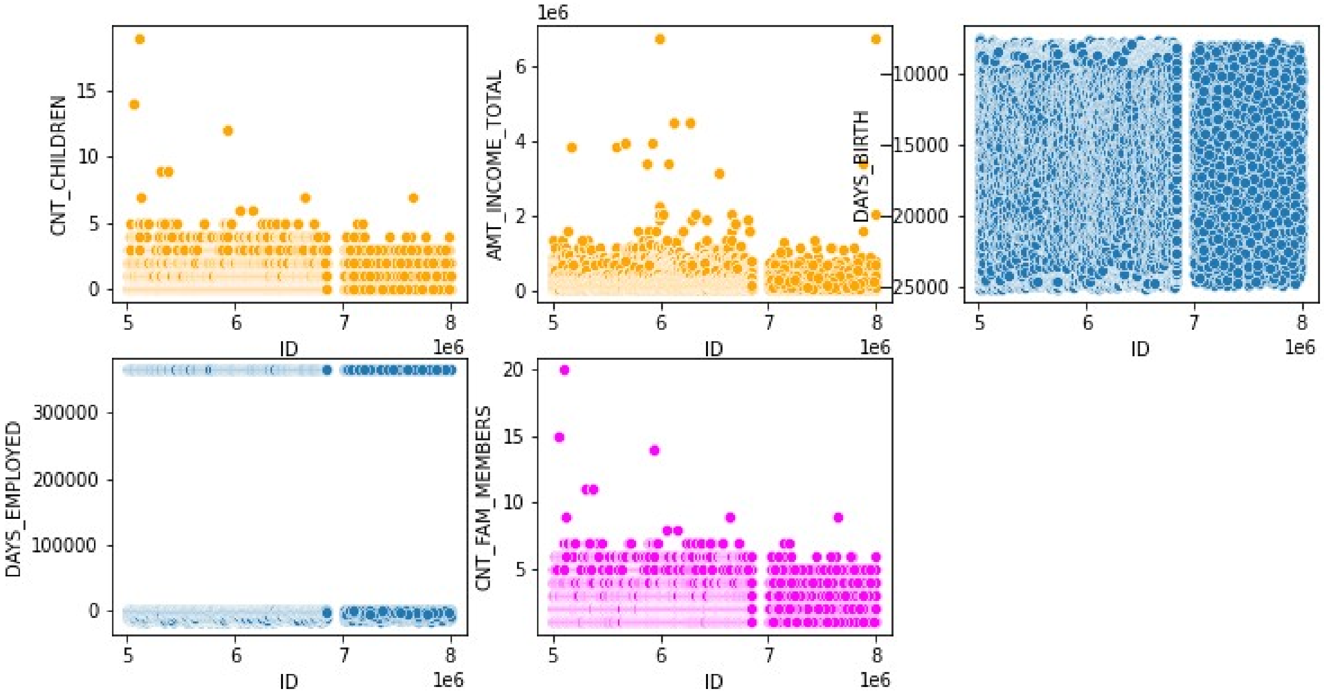
- Both the datasets have a common column name (‘ID’) connecting both the datasets.

- The length of intersection of two datasets is found to be 36457.

- The data-types in the application record are converted from the non-numeric data (object data-type) to numeric data using labelencoder.

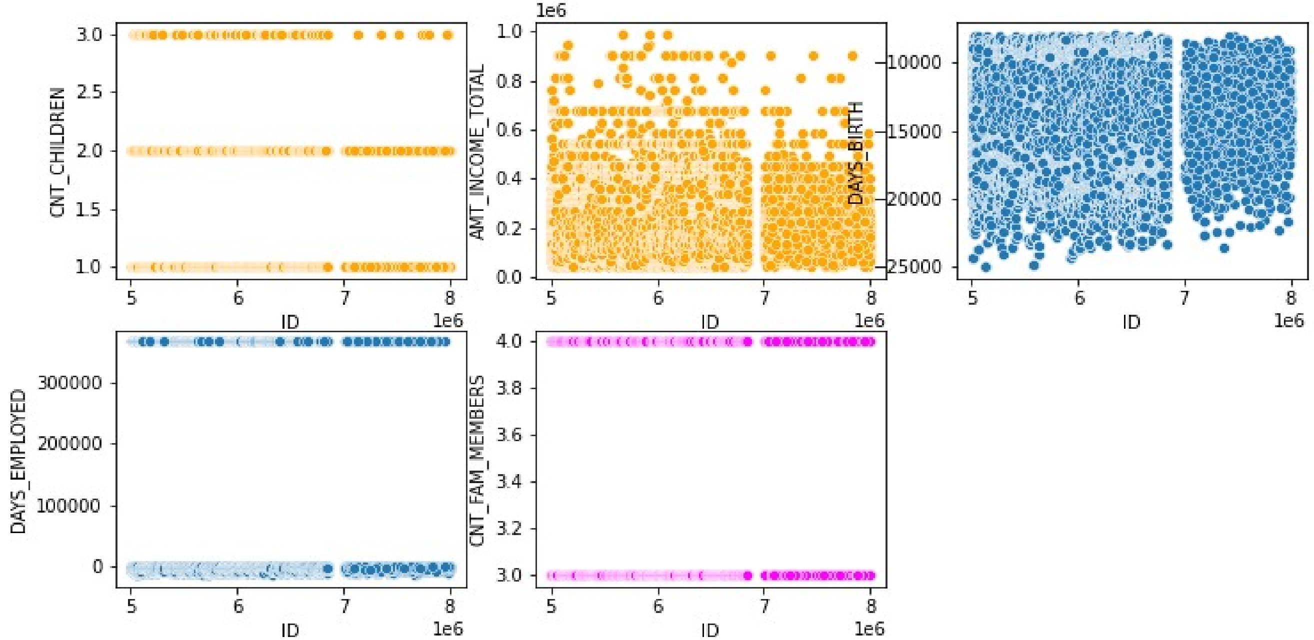


Features



- Few variables in the application record data-set checked if there are outliers
- Plot shows outliers in ‘cnt_children’, ‘amt_income_total’, ‘cnt_fam_members’

Features



- Removed the outliers in the above columns of the application record data-set and remade the plot

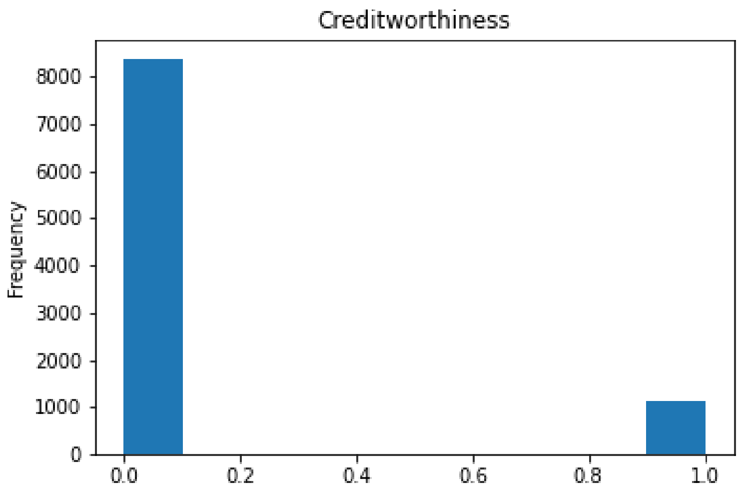
Credit-Record Dataset

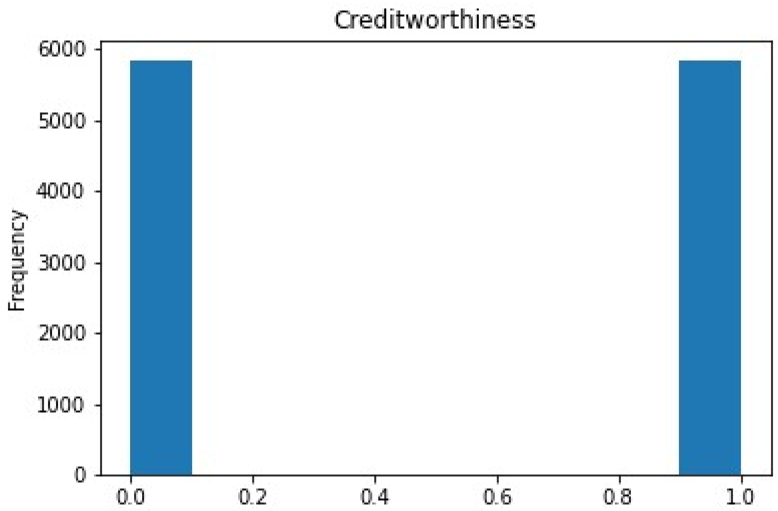
The credit-record dataset records the creditworthiness of a consumer into eight categories :

- 0 : 1 – 29 days past due
- 1 : 30 – 59 days past due
- 2 : 60 – 89 days over due
- 3 : 90 – 119 days over due
- 4 : 120 – 149 days over due
- 5 : Overdue or bad debts, write-offs for more than 150 days
- C : paid off that month
- X : no loan for the month

We regrouped these categories in only two : 0 (creditworthiness – 0. C, X) and 1 (no creditworthiness (1, 2, 3, 4 , 5).

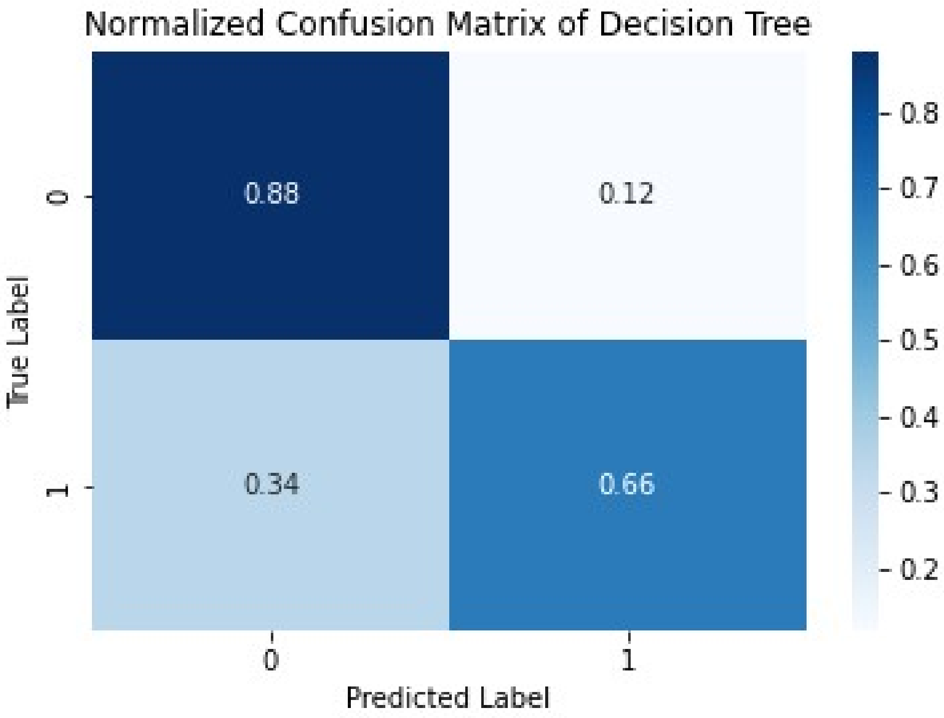
SMOTE Algorithm





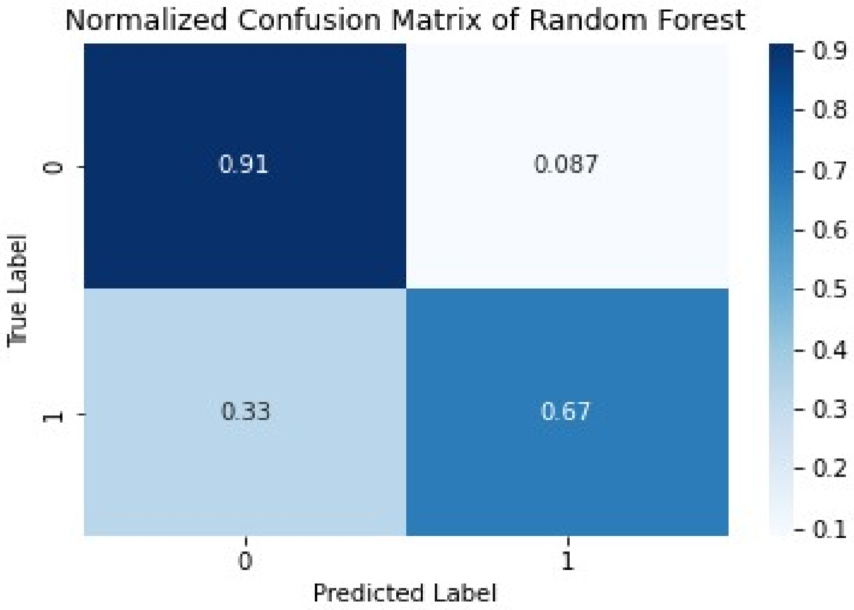
- The ‘status’ – which is the target ‘variable’ – has 87.97 % as 0 (creditworthiness) and 12.03 % as 1 (no credit-worthiness) (image in the left)
- The creditworthy category has larger population than the non-creditworthy. It suffers oversampling.
- We divided the data into those that will be used to train the model and those that will be used to predict the approval : 70 % for training and 30 % testing.
- To correct the over-sampling, we applied SMOTE (Synthetic Minority Over-sampling Technique) algorithm to generate the second category of data (non-creditworthiness) (image in the right)

CONFUSION Matrix



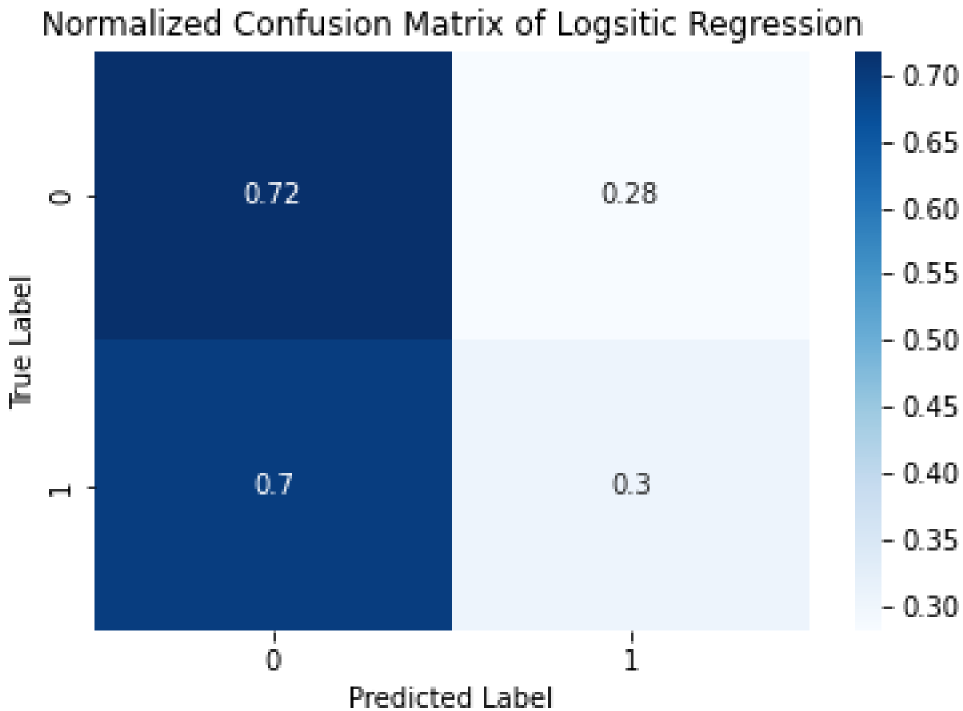
- We applied machine-learning models to the processed data.
- We began with the Decision Tree Classifier.
- The confusion matrix above shows it has high True Negative and True Positive.
- The accuracy score from this confusion matrix is 0.77.

CONFUSION Matrix



- Next, we fitted the Random Forest model on the training set of the data and use the model to make prediction using the test data.
- The accuracy score from this confusion matrix is 0.79.
- The Random Forest model was run with three variables : n_estimators = 150, max_depth = 16 and min_samples_lead = 12.
- We employed GridSearchCV to get the optimum values of these parameters and found n_estimators = 300, max_depth = 20 and min_samples_leaf =9.
- We refitted the model using these parameters and got the accuracy to be 0.80 – similar to previous value.

CONFUSION Matrix



- We then repeated the process of plotting the confusion matrix with the Logistic Regression model.
- We got the accuracy of this model to be 0.511.

Conclusions

- We built a machine learning-based classifier that predicts if a credit card application will get approved or not, based on the information provided in the application.
- While building this credit card approval predictor, we learned about common preprocessing steps such as label encoding, and handling outliers.
- We implemented three different machine learning models, optimized the hyper-parameters for one, and evaluated the performance using the accuracy score.
- Based on the accuracy score, we found the Random Forest Model to be most accurate.
- We have used python's machine learning libraries to implement machine learning algorithms. In the future, we can investigate to estimate the tangible benefits of the predictions of these machine learning models.

Thank You !