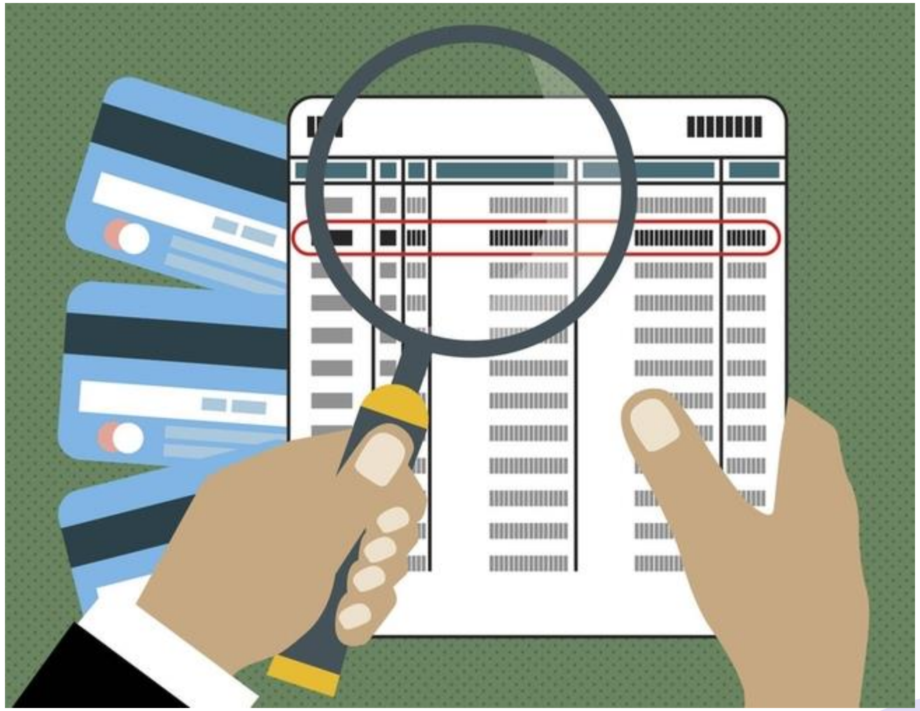
DEFAULT CREDIT CARD CLIENTS PREDICTION

CAPSTONE PROJECT PRESENTATION

Machine Learning Foundations Training

Dialog Data Science Academy



Dasun Kehelwala (DSA_0392)

2022-11-20

Contents

- ✓ Introduction (Problem Definition)
- ✓ Dataset
- ✓ Methodology (Solution Approach, Tools used)
- ✓ Results
- ✓ Conclusions
- ✓ Future Developments

Introduction

- **Problem Definition:-**

  Predicting credit card clients who will default on their next month payment.

- **Input Data :-**

  Prediction need to be done based on demographic characteristics, past spending and repayment patterns.

- **Target users :-**

  Helpful for banks which provide credit card facilities for Customers.

- **Business Value :-**

  Useful to manage credit risks.

- **Usability:-**

  Service need to accessed e through API and also by submitting batch input as csv file.

Dataset

- "Default of credit card clients Data Set" in UCI Machine learning repository
- Contains the default payment details in Taiwanese banking industry in year 2005
- Multivariate dataset with 24 attributes and 30,000 instances.
- Attributes of dataset was already converted to Real Integer values
- Class label indicates Default payment in Next Month?
    - Yes = 1→Positive Class
    - No = 0 →negative Class
- There is noticeable Class Imbalance in Dataset
- No null values present in dataset
- Duplicate values are observed
- Out of range values are observed

UCI Machine Learning Repository

Dataset : Column Details

Attribute	Description
ID	Identifier for data entry
X1 (LIMIT_BAL)	Amount of the given credit (NT dollar): Includes both the individual consumer credit and supplementary credit. →**Numerical**
X2 (SEX)	Gender (1 = male; 2 = female). →**Categorical variable mapped to integers**
X3 (EDUCATION)	Education Level (1 = graduate school; 2 = university; 3 = high school; 4 = others). →**Categorical variable mapped to integers**
X4 (MARRIAGE)	Marital status (1 = married; 2 = single; 3 = others). →**Categorical variable mapped to integers**
X5 (AGE)	Age (year) →**Numerical**
X6 - X11 (PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6)	History of past payment derived from past monthly payment records from April to September 2005. X6 = the repayment status in September; X7 = the repayment status in August; . . .; X11 = the repayment status in April (The measurement scale : -2: No consumption; -1 = pay duly; 0: The use of revolving credit; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above) → **Categorical variables mapped to integers, but have ordinal nature as per definition**
X12-X17 (BILL_AMT1 to BILL_AMT6	Amount of bill statement (NT dollar) from April to September 2005. X12 = amount of bill statement in September ; X13 = amount of bill statement in August; . . .; X17 = amount of bill statement in April . →**Numerical**
X18-X23 (PAY_AMT1)	Amount of previous payment (NT dollar) from April to September 2005. X18 = amount paid in September ; X19 = amount paid in August ; . . .; X23 = amount paid in April . →**Numerical**
Y (default payment next month)	Default payment (Yes = 1, No = 0) → **class label variable - Categorical variables mapped to integers**

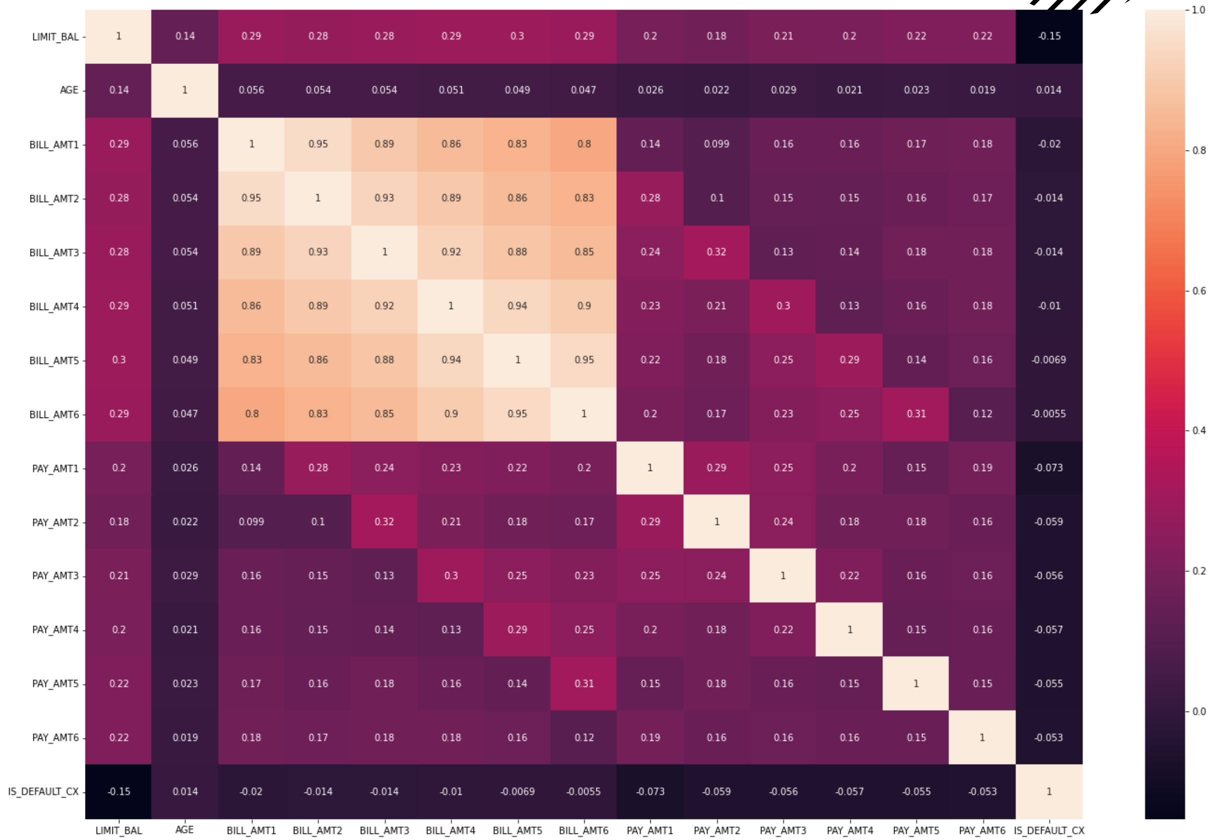
Independent Variables

Methodology

This machine learning challenge was approached as binary classification problem.

ID	Step	Tools Used
#1	Identifying and Loading Required Libraries	Jupyter Notebook, Google Collaboratory platform
#2	Loading Data and Viewing Basic Information About Dataset	Pandas, Matplotlib
#4	Data Preprocessing	Pandas, Numpy
#4	Exploratory Data Analysis	Matplotlib, Seaborn
#5	Feature Engineering, Feature Selection and Preparing for Machine Learning Model training	Sklearn.Preprocessing, Sklearn.Model_Selection, Imblearn
#6	Model Building and Evaluating	Sklearn (RandomForest,Logistic Regresssion), XGBoost, Sklearn.Metrics
#7	Hyperparameter Tuning and Selecting Best Model	Sklearn.Metrics
#8	Saving Best Model	Joblib.
#9	Developing Inference Flow (Future Step)	Joblib, Pandas
#10	Application Deployment (Future Step)	Flask, request , jsonify, json

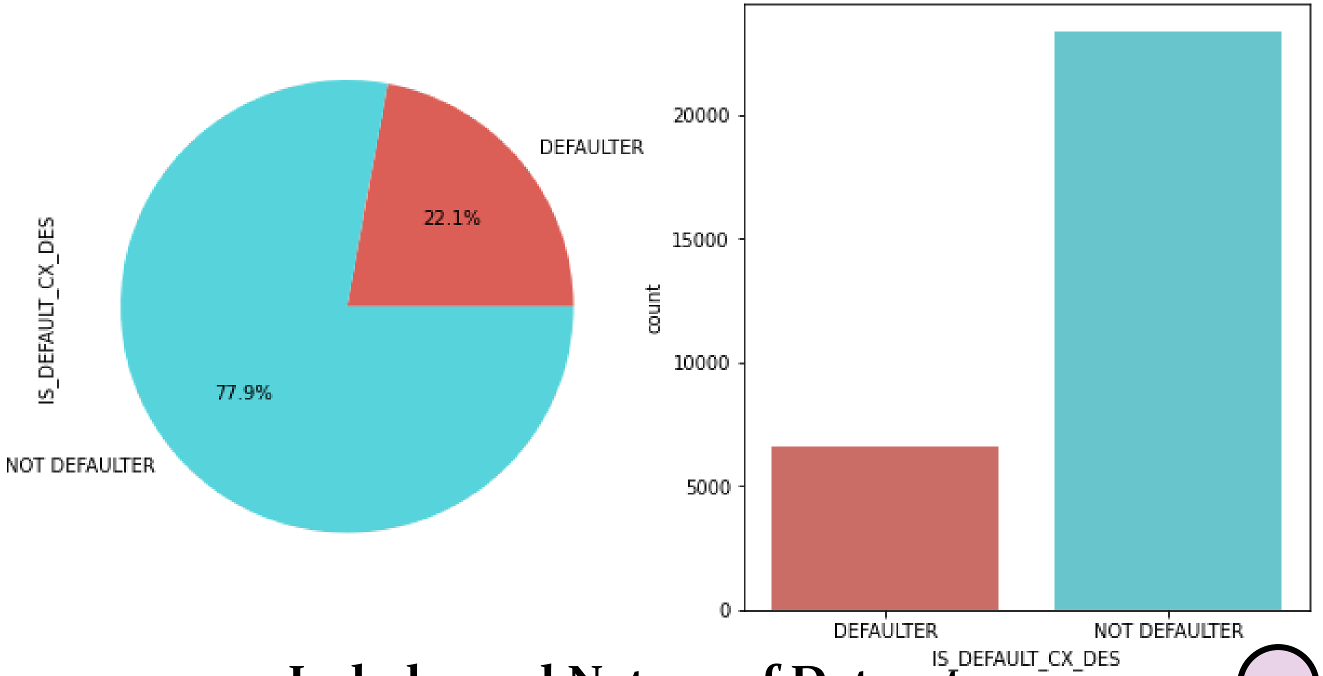
Methodology : Exploratory Data Analysis



- Numerical columns show very low negative correlation with Class Label
- Bill_AMT variables of Six months are highly correlated

Methodology : Handling Imbalance

- Using Random Over Sampling Technique to Oversample Minority Positive Class
  - Implemented using **Imblearn** Library
- Instead of Accuracy, Using Alternative Performance Evaluation Metrics
  - Precision Score
  - Recall Score
  - F1 Score
  - F1 score (Weighted)
  - ROC AUC Score



Imbalanced Nature of Dataset

Results

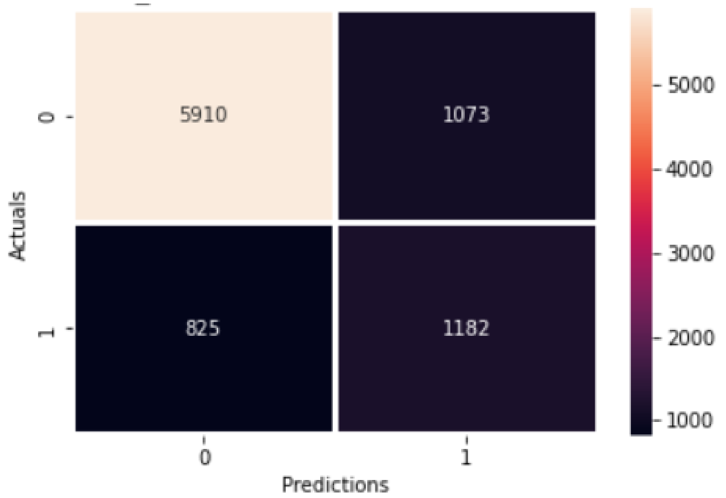
Output Models After Hyper Parameter Tuning

Model ID	Model	Resampeling method	Feature count	Accuracy score	Precision score	Recall score	F1 score	F1 score weighted	ROC AUC score
xgb_rovs_02	XGBClassifier(colsample_bytree=0.5, gamma=9, m...	Random Over Sampling	23	0.760734	0.473742	0.647235	0.547063	0.772604	0.790571
xgb_rovs_03	XGBClassifier(colsample_bytree=0.5, gamma=1, n...	Random Over Sampling	23	0.763404	0.477794	0.643249	0.548312	0.774668	0.788177
xgb_rovs_04	XGBClassifier(colsample_bytree=0.5, gamma=0.5,...	Random Over Sampling	23	0.763181	0.477424	0.642750	0.547887	0.774456	0.789252
xgb_rovs_05	XGBClassifier(colsample_bytree=0.5, gamma=0.5,...	Random Over Sampling	23	0.765740	0.481257	0.633284	0.546902	0.776146	0.787701
rf_rovs_02	(DecisionTreeClassifier(max_features='auto', r...	Random Over Sampling	23	0.809121	0.602972	0.424514	0.498246	0.796439	0.771253
rf_rovs_03	(DecisionTreeClassifier(max_features='auto', r...	Random Over Sampling	23	0.810011	0.603892	0.432985	0.504353	0.798066	0.774043
rf_rovs_04	(DecisionTreeClassifier(max_depth=10, max_feat...	Random Over Sampling	23	0.788877	0.524169	0.588939	0.554669	0.793111	0.785155
rf_rovs_05	(DecisionTreeClassifier(max_depth=10, max_feat...	Random Over Sampling	23	0.788877	0.524211	0.587942	0.554251	0.793049	0.785135

Best Model Observed : (Model ID = rf_rovs_04)

- RandomForestClassifier(max_depth=10, n_estimators=500, n_jobs=3,random_state=42)
- Resampling method : Random Over Sampling
- Feature count : 23

Confusion Matrix and Performance Scores for Best Model



Metric	Value
Accuracy Score	0.7889
Precision Score	0.5242
Recall Score	0.5889
F1 Score	0.5547
F1 score (Weighted)	0.7931
ROC AUC Score	0.7856

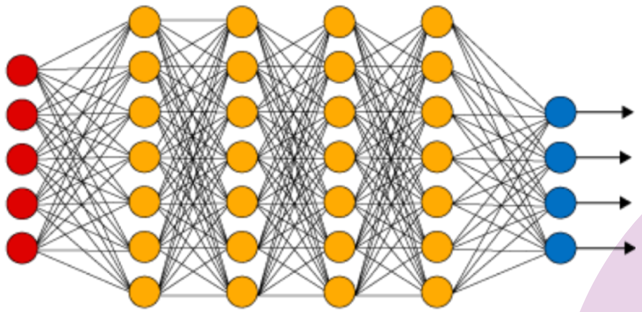
Conclusions

- Application requires to increase true positive count while maintaining false positives and false negatives at satisfactory level.
- Due to imbalanced nature of dataset, accuracy is high when algorithm classify majority of true negatives correctly.
- F1 score was used to benchmark model performance as measuring accuracy not serves intended purpose in this context
- Observed maximum F1 score (0.5547) can be considered as acceptable value in this context (F1 score > 0.5)
- In general, Models with >0.8 F1 scores greater are considered as good Models.
- Even though this model is usable, further finetuning is required to improve model performance further.

Future Developments

Further Improving Model Performance

- Further refining input feature set using feature engineering and feature selection techniques (Use one-hot encoding, Derive new features by removing correlations)
- Trying with more different combinations of features out of full feature set to find optimum feature combination
- Trying with different resampling technique like Synthetic Minority Oversampling Technique (SMOTE)
- Trying with more classifier types including Classifiers like Support Vector Machine and Deep Neural Networks (DNN)



Developing Inference Pipeline and Deploy Application

- Enabling convenient access to solution through API/GUI

Using explainable Machine Learning techniques

- Improving model explain ability and result interpretability

THANK YOU

Email : Dasun.Kehelwala@dialog.lk