MACHINE LEARNING PROJECT

Project Title: Credit Card Fraud Detection

Problem Description

Fraud detection systems (FDS) mainly perform two tasks:

1. Real-time detection
2. Posterior detection

The second task manages the largest volume of transactions. It aims at predicting if a transaction is fraudulent based on its characteristics and the past transactions of the cardholder. Yet, in posterior detection, verification often takes days, so new payments on the card become available before a decision is taken.

Our motivation : Posterior fraud detection

Datasets Briefing

1) The Main dataset used for this project is taken from Kaggle. The dataset contains 307511 rows and 122 columns.
2) The other two Datasets used for this project is taken from GitHub. The dataset are encrypted and are used to train the model.

Datasets

Data Pre-processing

The dataset used for this project is highly imbalanced with only 0.17% of the data being fraudulent. The dataset is further trimmed to 307511 rows and 9 columns.

The 9 columns are:

```
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 9 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   SK_ID_CURR              307511 non-null  int64
 1   TARGET                  307511 non-null  int64
 2   AMT_INCOME_TOTAL        307511 non-null  float64
 3   AMT_CREDIT              307511 non-null  float64
 4   AMT_ANNUITY             307499 non-null  float64
 5   AMT_GOODS_PRICE         307233 non-null  float64
 6   NAME_INCOME_TYPE        307511 non-null  object
 7   HOUR_APPR_PROCESS_START 307511 non-null  int64
 8   ORGANIZATION_TYPE       307511 non-null  object
dtypes: float64(4), int64(3), object(2)
memory usage: 21.1+ MB
```

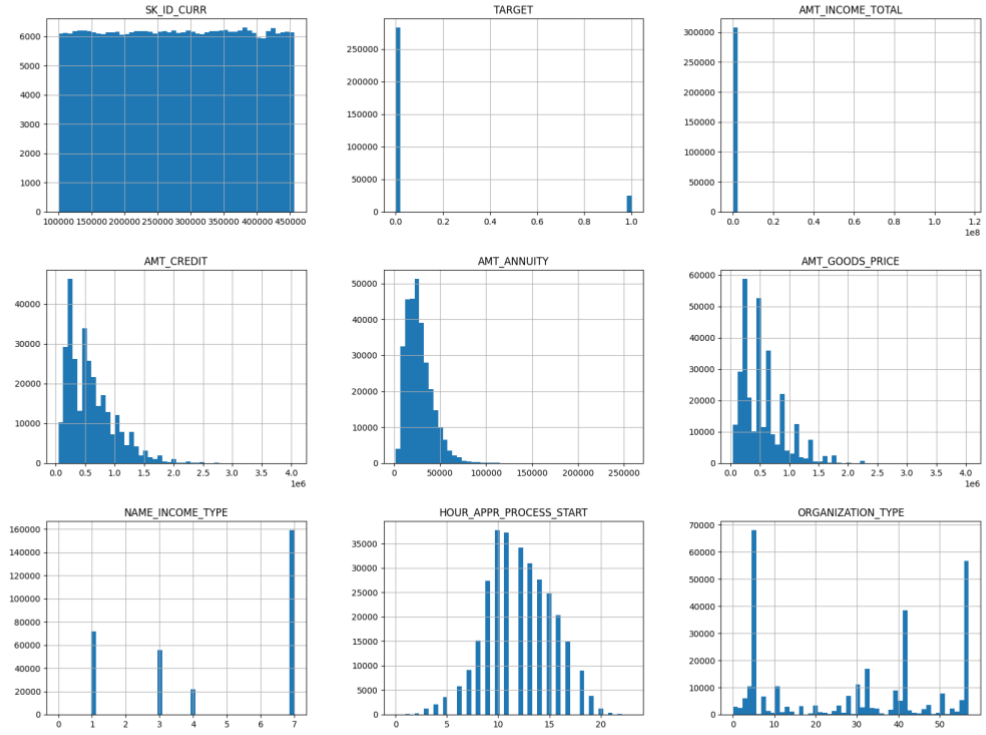
Keywords

• Posterior detection • highly imbalanced
• Random Forest Classifier •Logistic Regression
•K-Neighbours Classifier •Confusion Matrix

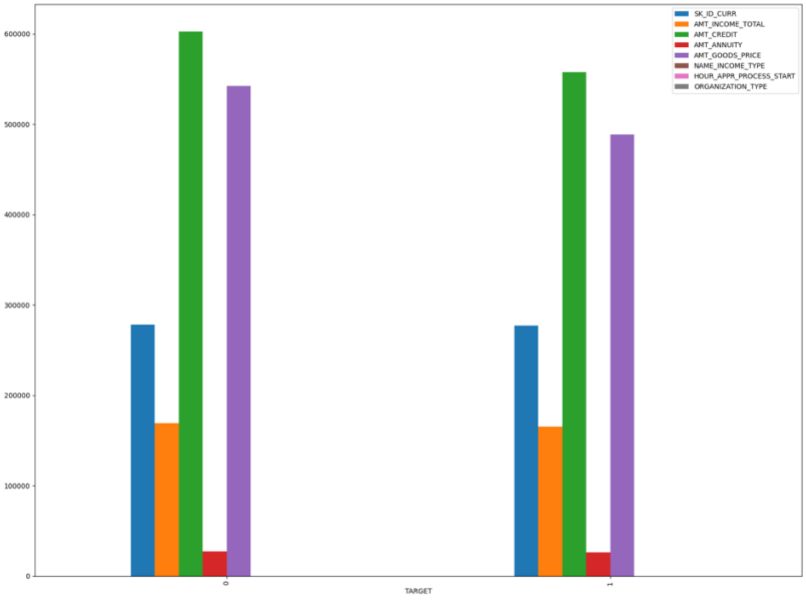
Methods used for Data Pre-processing

1) Correlation matrix for numerical data
2) One-hot encoding for categorical data
3) StandardScaler for numerical data
4) Label Encoding for categorical data

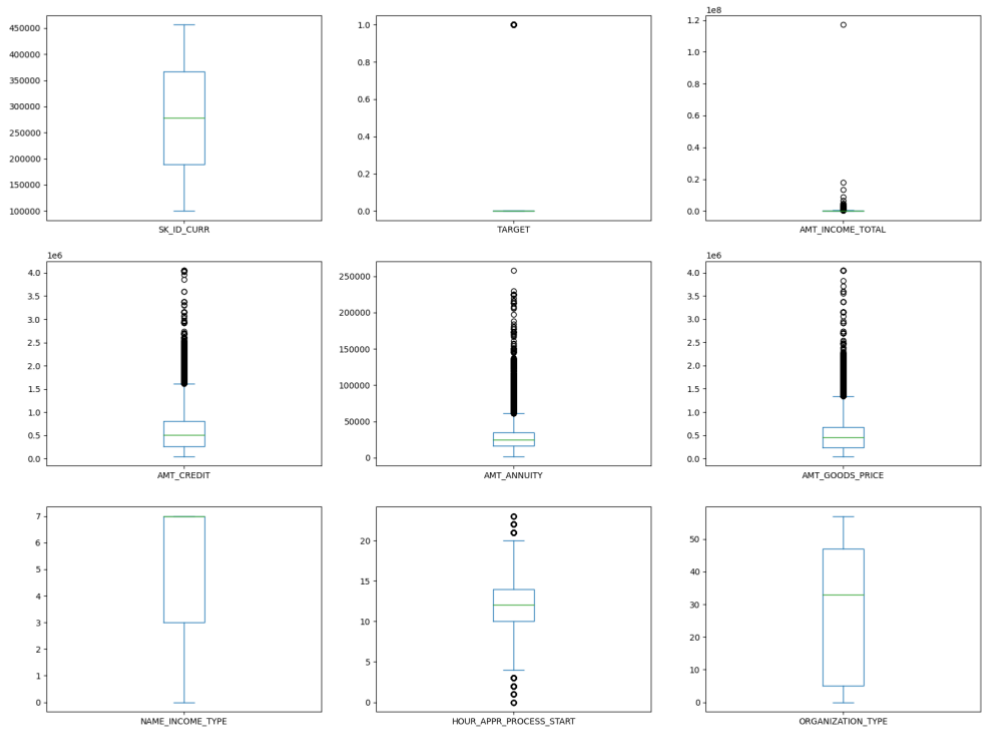
Statistical summary of all attributes



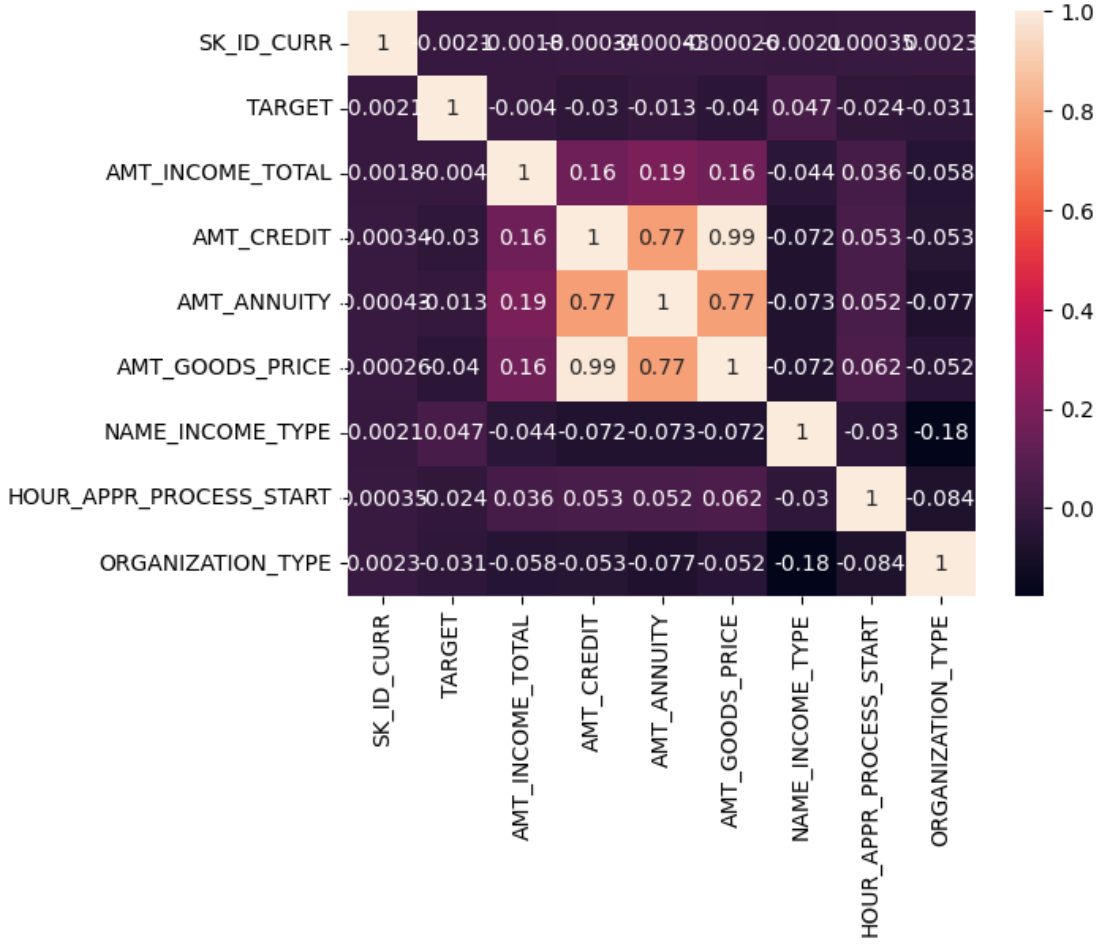
Data Visualization



Checking Outliners



Data Corelation



Python Packages used for:

- Data Pre-processing:
  - Pandas
  - Numpy
  - Matplotlib
  - Seaborn
  - Scikit-learn

- Data Visualization:
  - Matplotlib
  - Seaborn

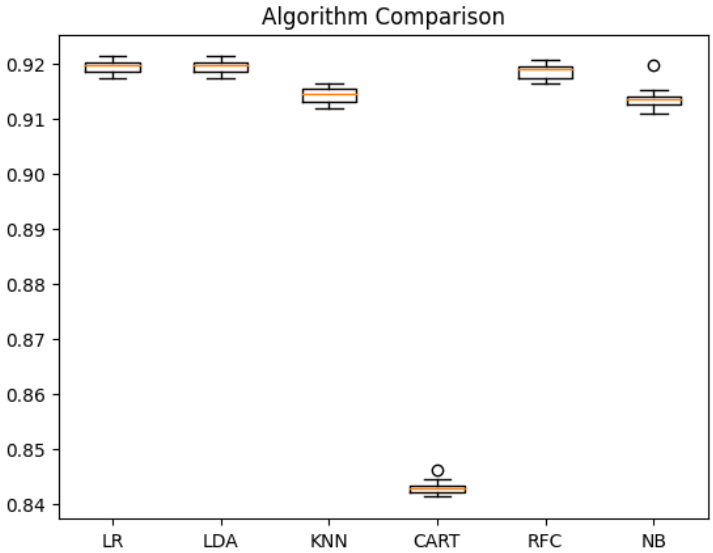
- Model Building:
  - Scikit-learn:
    - RandomForestClassifier
    - LogisticRegression
    - SVC, LinearSVC
    - train_test_split
    - LabelEncoder

- Model Evaluation:
  - Scikit-learn:
    - confusion_matrix
    - accuracy_score
    - f1_score
    - classification_report
    - roc_curve
    - roc_auc_score

Learning Algorithms

Out of all the supervised Machine Learning Algorithm. There were the 3 algorithms which performed well on our dataset.

1) Random Forest Classifier: Result mean 0.919268
2) Logistic Regression: Result mean 0.919261
3) K-Neighbours Classifier: Result mean 0.914260



Result and Conclusion

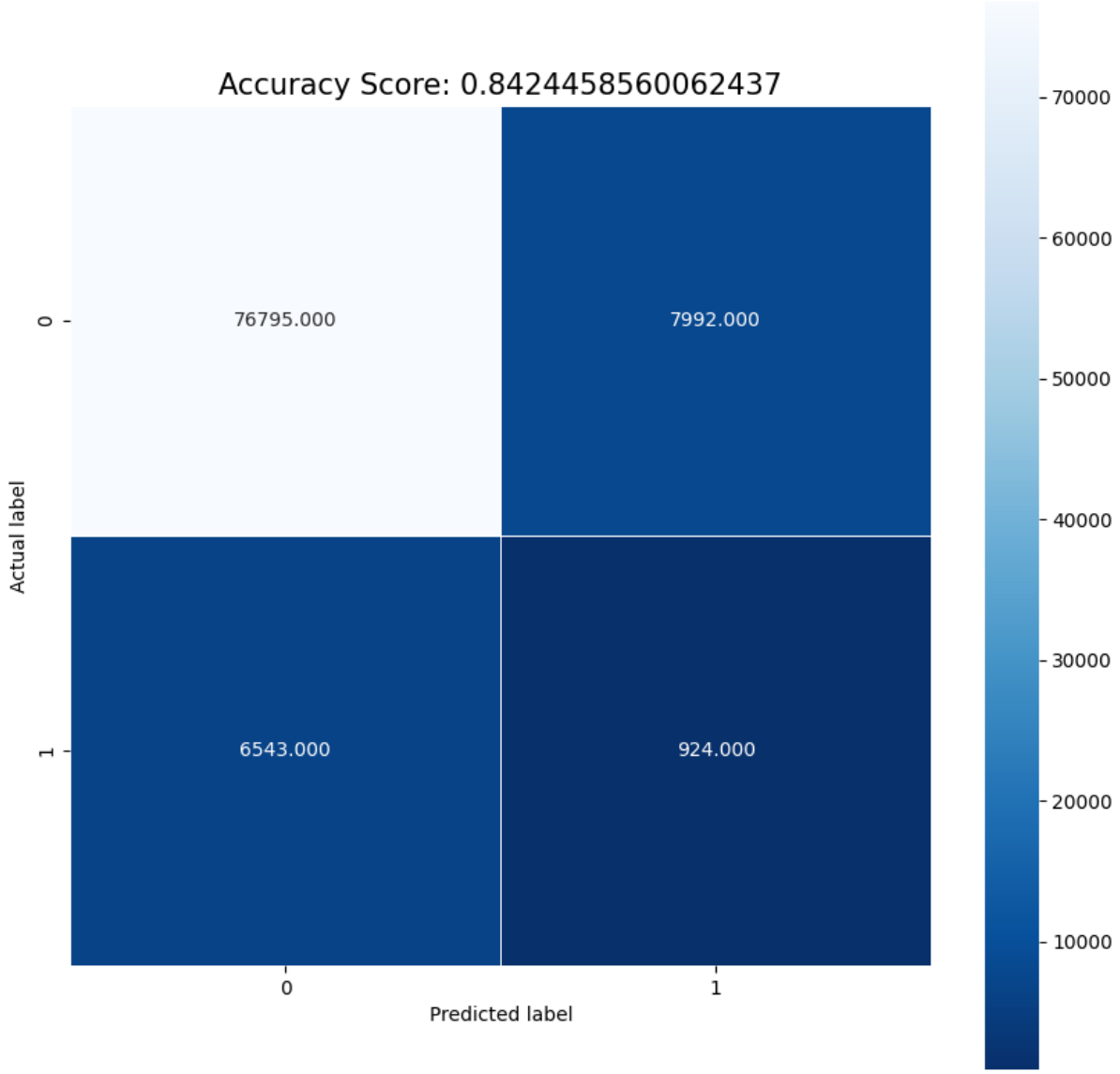
- Model Evaluation :

  Confusion Matrix

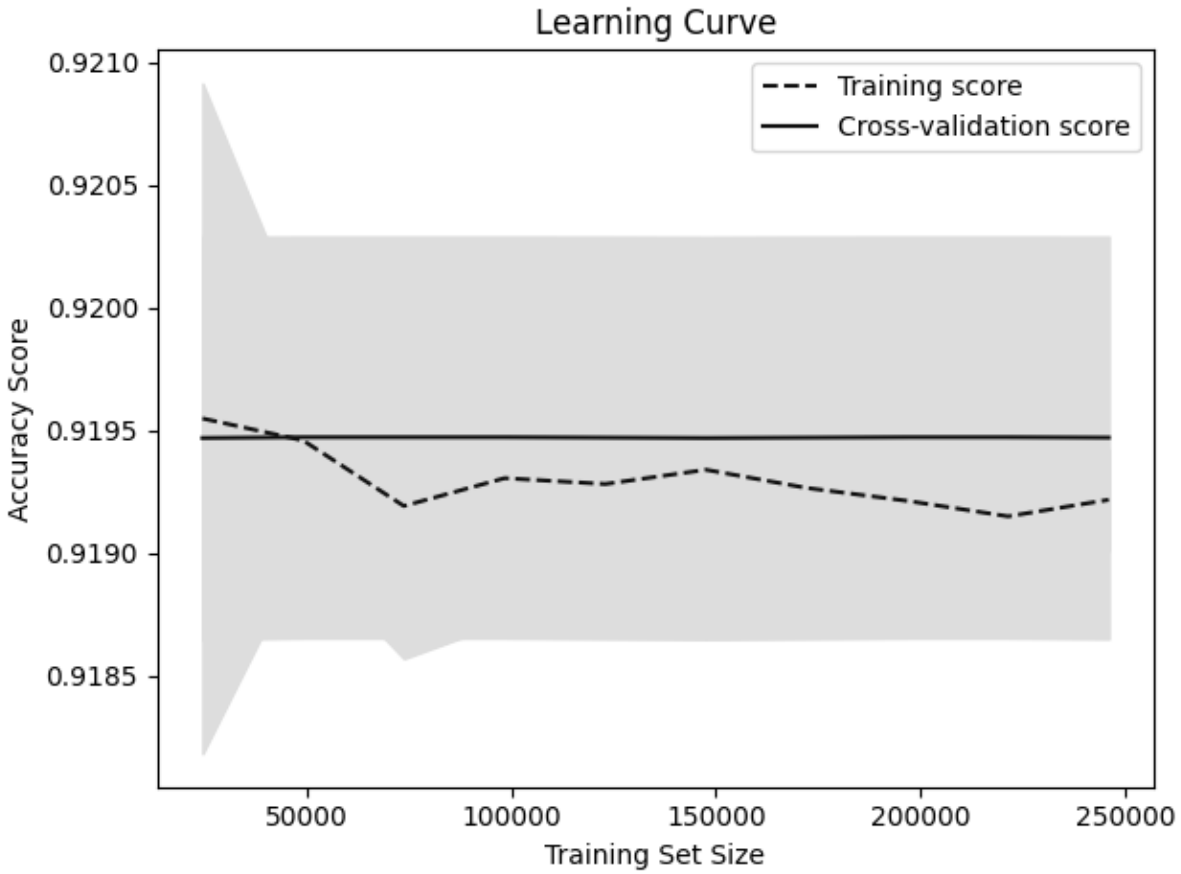
```
[[76795  7992]
 [ 6543   924]]
              precision    recall  f1-score   support

           0       0.92      0.91      0.91     84787
           1       0.10      0.12      0.11      7467

    accuracy                           0.84     92254
   macro avg       0.51      0.51      0.51     92254
weighted avg       0.86      0.84      0.85     92254
```



- Overfit Under Fit curve



Team Members

- Harshit Raj – AM.EN.U4CSE20349
- Patel Rajkumar – AM.EN.U4CSE20349
- Prashant Kumar N – AM.EN.U4CSE20352
- Rupesh Kumar Tailor – AM.EN.U4CSE20357