Credit Card Fraud Detection

Aung Hein 06-Jun-2024

Executive Summary

Problem

How can we detect Credit Fraud ?

Can we detect Credit Fraud by Machine Learning?

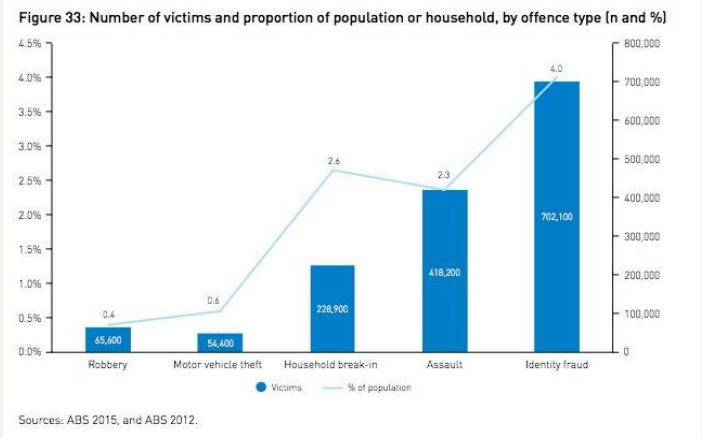
Goal

Introduction	Methodology	Results	Conclusion	Appendix
Introduction to problem,goal and dataset	Details explanation for library and analysis flow	Result outcome for each model and evluation	Summary about the finding	Others info and links

Introduction

- **Problem**

  Credit card fraud can occur when unauthorized users gain access to an individual's credit card information in order to make purchases, other transactions, or open new accounts. According to a 2021 annual report, about 50% of all Americans have experienced a fraudulent charge on their credit or debit cards, and more than one in three credit or debit card holders have experienced fraud multiple times. This amounts to 127 million people in the US that have been victims of credit card theft at least once.



- **Goals**

  Our Goals is to detect credit fraud by using machine learning algorithms, train model and turn parameter for performance.

- **Dataset**

  I used dataset from Kaggle which contain credit card transactions made by European cardholders in the year 2023. It comprises over 550,000 records, and the data has been anonymized to protect the cardholders' identities. The primary objective of this dataset is to facilitate the development of fraud detection algorithms and models to identify potentially fraudulent transactions.

  Key features :

  - id: Unique identifier for each transaction
  - V1-V28: Anonymized features representing various transaction attributes (e.g., time, location, etc.)
  - Amount: The transaction amount
  - Class: Binary label indicating whether the transaction is fraudulent (1) or not (0)

Methodology

- **Data Wrangling**

  Data was loaded and cleaned by using Pandas dataframe.

- **Exploratory data analysis**

  EDA was done by using Pandas, Numpy, Matplotlib, and Seaborn.

- **Standardization and Train/Test Split**

  Data been standardize by using StandardScaler function from sklearn.

  And data been split by using StratifiedShuffleSplit, StratifiedKFold to preserving the percentage of samples for each class.

- **Modeling**

  I been used LogisticRegression, KNeighborsClassifier and DecisionTreeClassifier from sklearn for this project.

- **Model evaluation**

  - I used accuracy_score, confusion_matrix, precision_score, recall_score, f1_score to evaluate the models.
  - And also models are validate by cross_val_predict.

- **Hyper parameter Tuning**

  Hyper parameter tuned by using gridsearchcv.

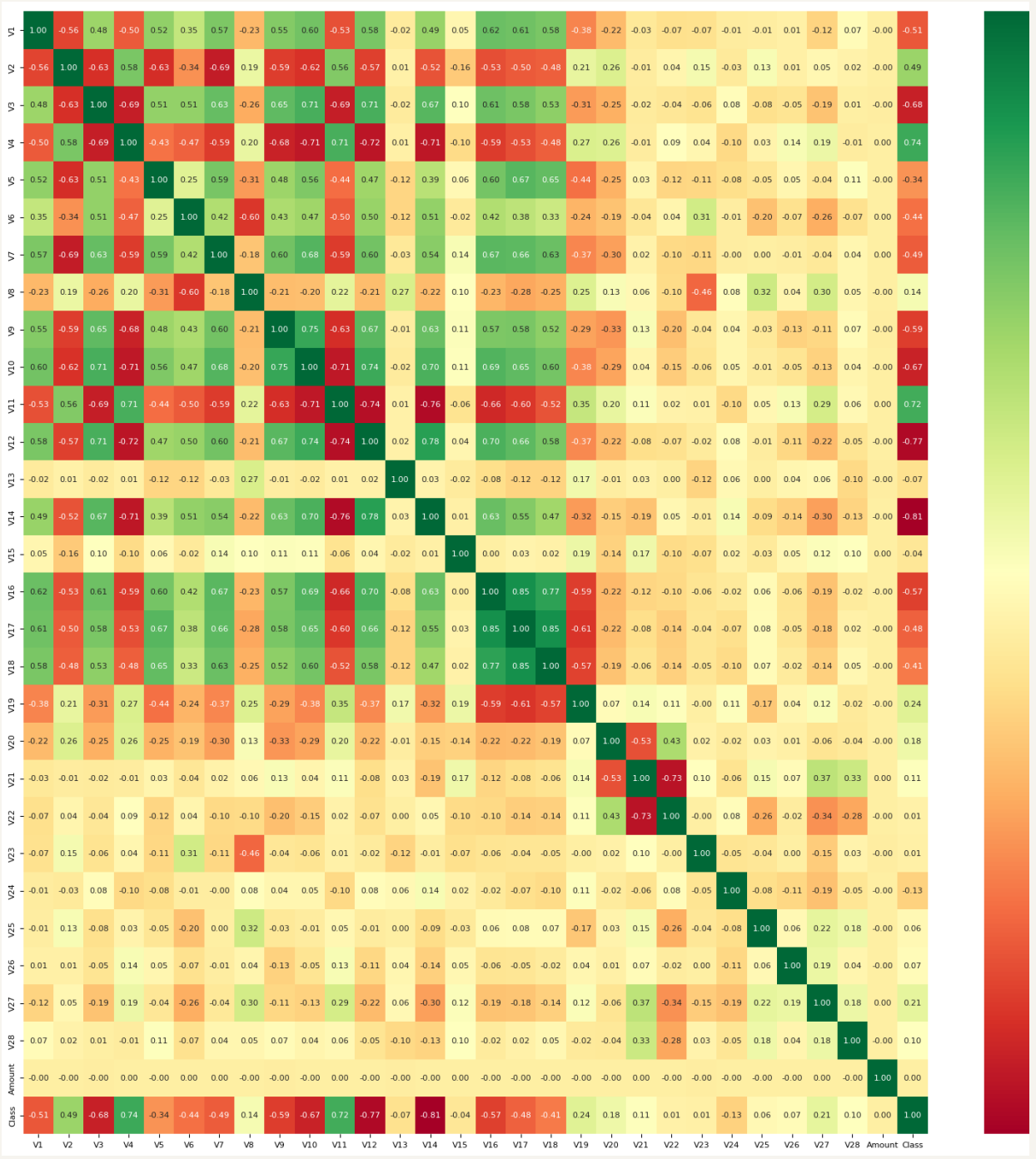
Data Preparation

- **Data Cleaning**
  - No null values was found
  - 1 duplicated columns are founds and removed
  - 'id' columns are removed
  - Data types are already numeric as float64 and target columns are int64
- **Data Standardization**
  - Dataset are standardize with Standard Scalar function from sklearn
- **Dataset Train/Test Split and Fold for cross validation**
  - Dataset's target sample contain 50% true (1) value and 50% (0) false value.
  - data been split by using StratifiedShuffleSplit, StratifiedKFold to preserving the percentage of samples for each class.

Exploratory data analysis

Correlation

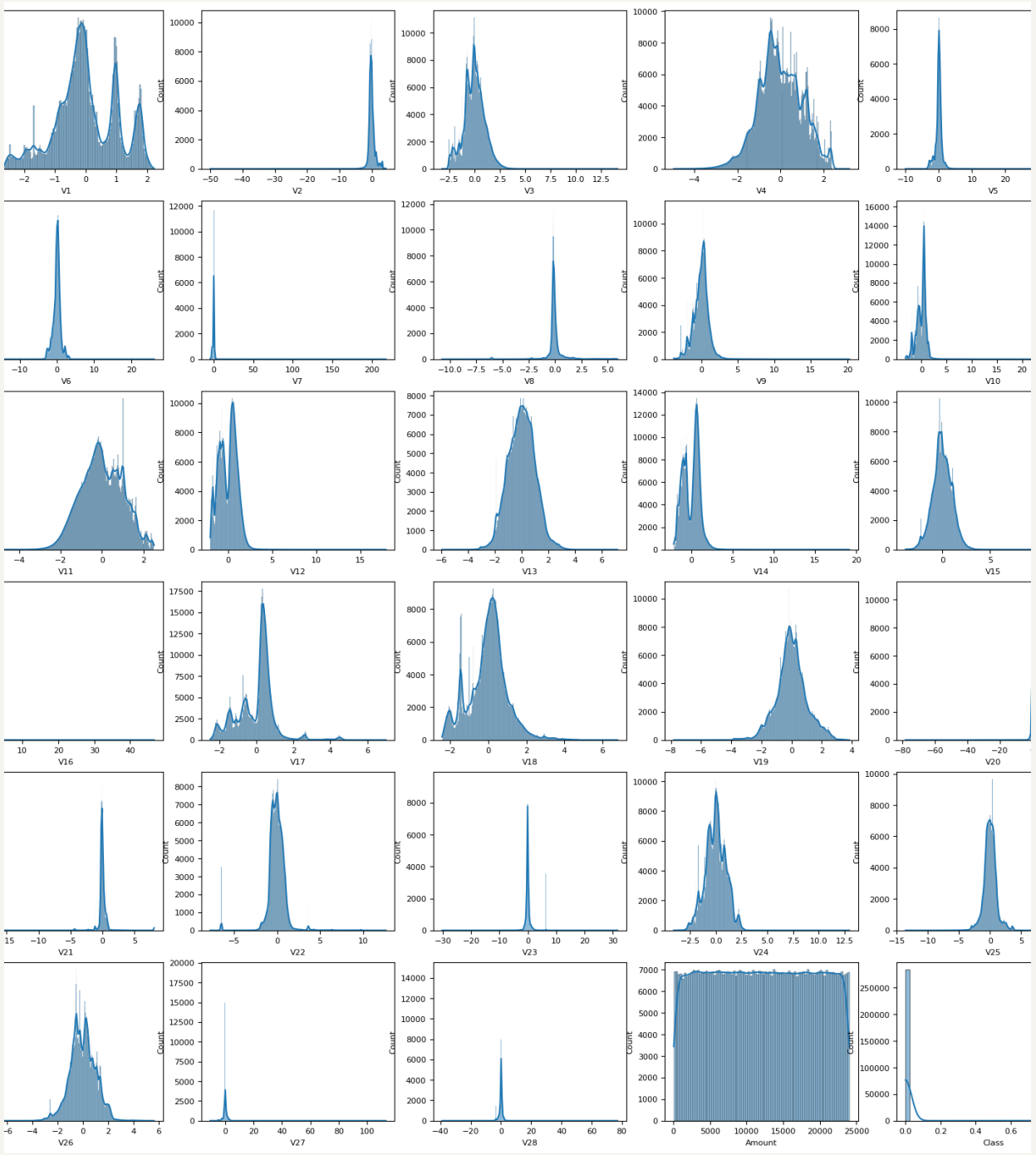
found a few features are strongly correlative to each others.



Exploratory data analysis

Distribution

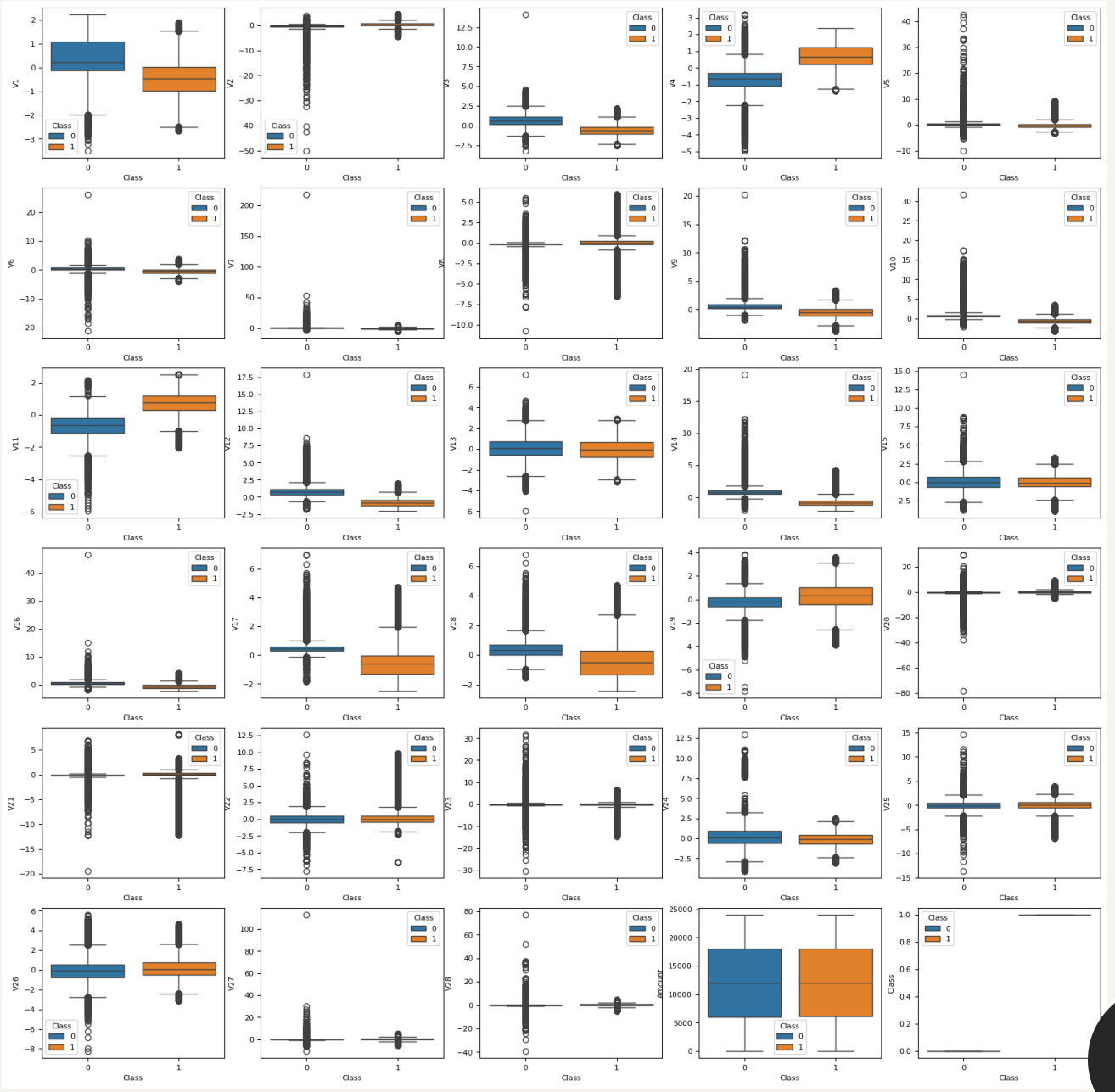
skewness and distribution are identified.



Exploratory data analysis

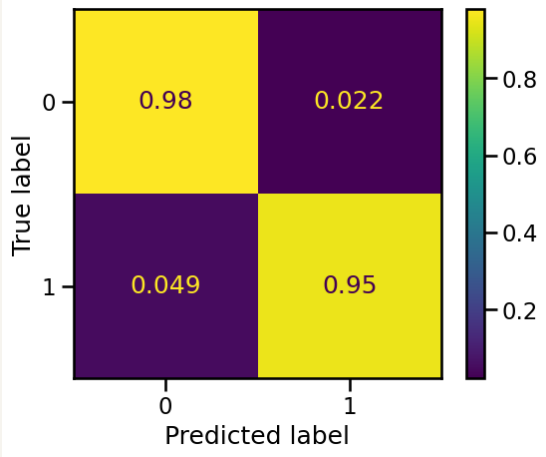
Outlier , mean and median

Outlier , mean and median values are identified



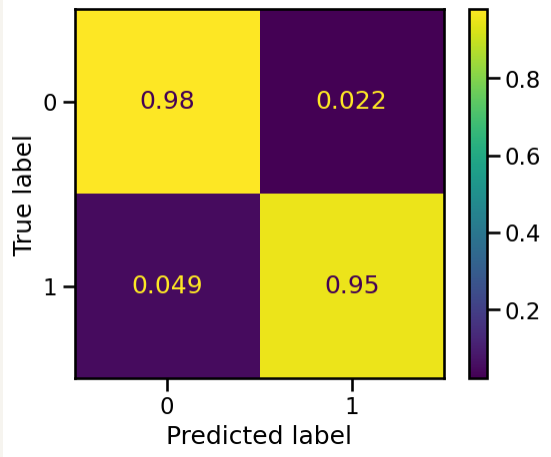
Logistic Regression

L2 penalty



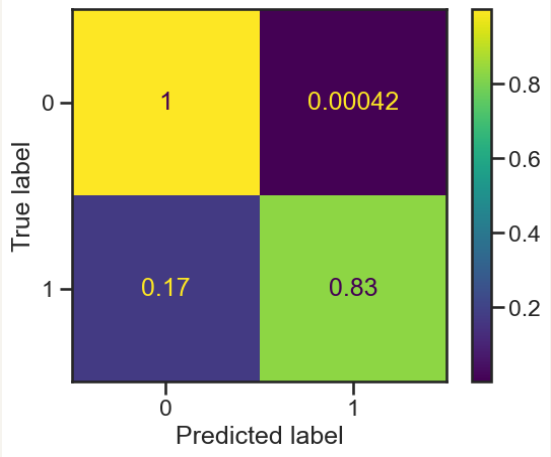
	0	1
accuracy	0.964546	0.964546
recall	0.978088	0.951005
precision	0.952297	0.977478
f1score	0.965020	0.964060

L1 penalty



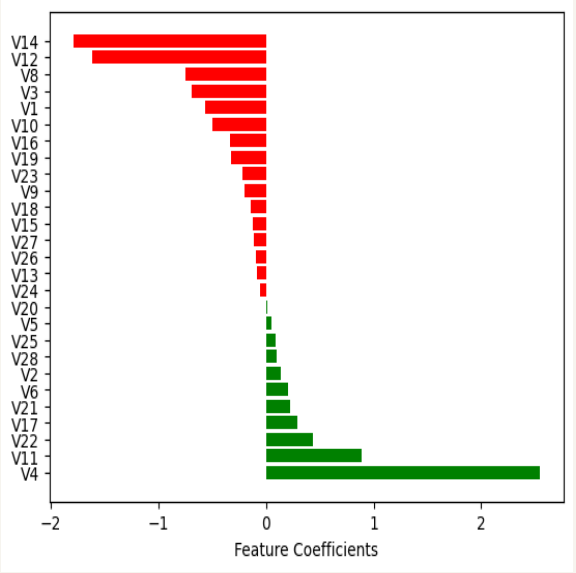
	0	1
accuracy	0.964520	0.964520
recall	0.978211	0.950829
precision	0.952140	0.977597
f1score	0.964999	0.964027

Elasticnet



	0	1
accuracy	0.914725	0.914725
recall	0.999578	0.829872
precision	0.854555	0.999492
f1score	0.921395	0.906818

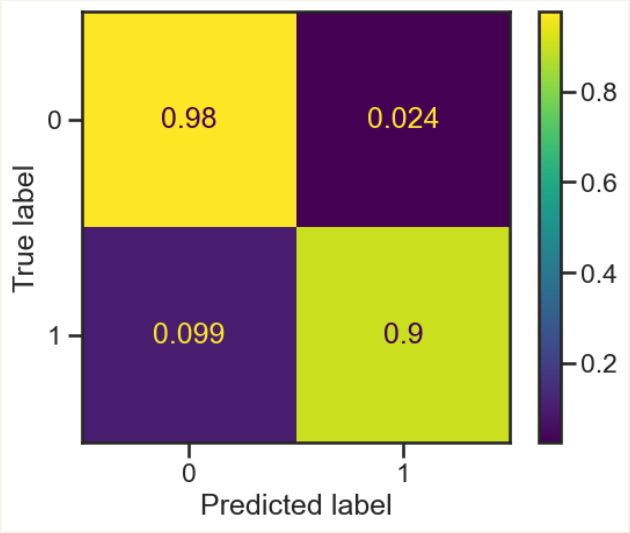
Feature coefficient

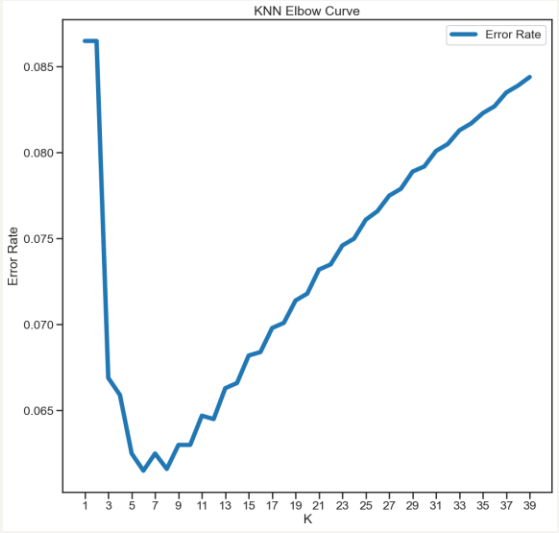


- Both L1 and L2 penalty got better accuray but recall for non-fraud value are lower than Elasticnet model which would impact on user experience.
- 'V7' and 'Amount' columns seen penalized by regulation which seen don't have much coefficient for target.

K Neighbors Classsifier

KNN model with all features

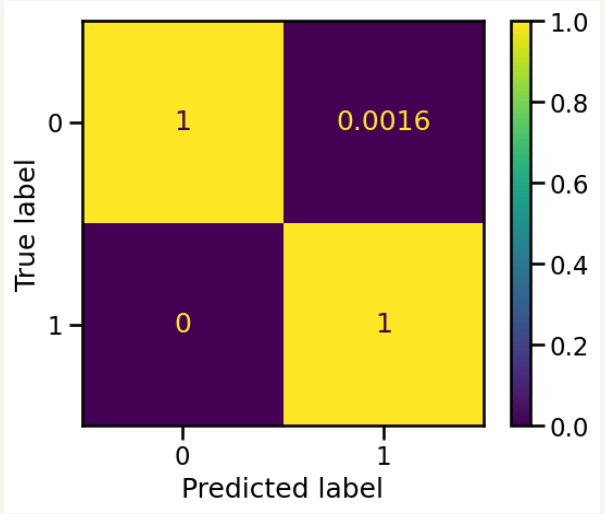




	0	1
accuracy	0.938545	0.938545
recall	0.976188	0.900902
precision	0.907841	0.974250
f1score	0.940775	0.936141

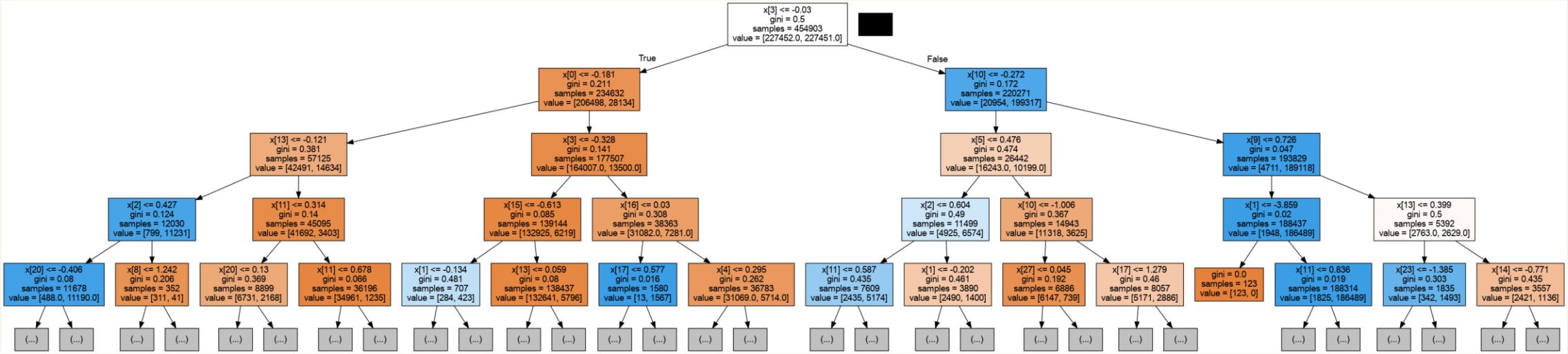
- KNN model with all features columns does not score much may be it overfit on outlier
- KNN elbow curve result show model fit best with k = 6 .
- KNN model predict well after remove 'V7' and 'Amount' columns.
- Cross validation with 4 fold by using StratifiedKFold result almost same.

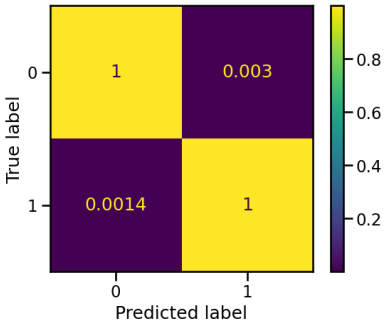
KNN model with two features removed



	0	1
accuracy	0.999117	0.999117
recall	0.998234	1.000000
precision	1.000000	0.998237
f1score	0.999116	0.999118

Decision Tree Classifier





	0	1
accuracy	0.997879	0.997879
recall	0.997200	0.998558
precision	0.998556	0.997204
f1score	0.997878	0.997881

- Hyper parameter are tuned by using gridsearchcv and best parameter for decision tree are max_features=6 and max_depth=36 for this project.
- cross validation still have good score with 99.8% and model work well even without removing two features ('V7' and 'Amount').

Conclusion

- Decision Tree Classifier predict almost 99.8% even with non corelated features 'V7' and 'Amount'.
- Feature Engineering are important and we can see impact of that in KNN model with same hyper parameter.
- KNN model score best 99.9%
- Logistic Regression does not work well for this credit fraud detection but it help much on feature selection.

Appendix

- DataSet Link : https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023/data
- Code Link : https://github.com/thonenyangal/Credit-Card-Fraud-Detection-Project.git