Financial Fraud Detection with Machine Learning.

Solomon OKORO

github.com/soulsuv

OBJECTIVES

- Analyse a bank’s loan data and perform detailed Exploratory Data Analysis for missing values treatment & outlier removal.
- Perform Feature Engineering for the variables to be used in the Machine Learning Models.
- Predict the tendency of their clients to default on loans with different Machine Learning Models and compare the Models’ performance.
- Tool used: Python for visualisation and machine learning.

DATA SUMMARY

Dataset has **105,471** rows and **771** columns.

The target variable that indicates if a client has defaulted on a loan is termed as “loss”.

A loss value of 0 indicates there has been no default, while a value of 1 or greater denotes default, with potentially higher values suggesting a more severe default.

All other 770 columns of data represent the different characteristics of the clients relating to loan records.

EDA and Pre-processing

- Null values in each column ranged from 0% to 18% of the rows.
- Deleting all rows with Null values reduced the rows from 105,471 to 51,940 (~45% reduction). This is above the 10% threshold for dropping Null Values.

- Following the analysis of the numeric data, deployed imputation instead to fill Null values with the median as the data was mostly skewed.

- After filling numeric values, no nulls were remaining.

- Converting Target Column to Binary (0,1)

```
data.loc[data['loss'] >= 1, 'loss'] = 1
```

Also dropped the id column before proceeding with Outlier Detection.

Outlier Detection & Scaling

- Outlier detection and removal
  - Using Box-plot method for outlier removal led to removal of over 65% of the Data.
  - Instead, performed Z-score method on the following columns:

```
['f337', 'f466', 'f482', 'f492', 'f531', 'f576',
'f585']
```

  - Followed by Box-plot method on the following two columns as they contained many outliers:

```
['f576', 'f585']
```

- Final dataset: 71,070 rows * 770 columns

- Scaling the dataset (except the Target Column)

`RobustScaler()` : for columns still with large outliers.

`StandardScaler()` : for other numeric data.

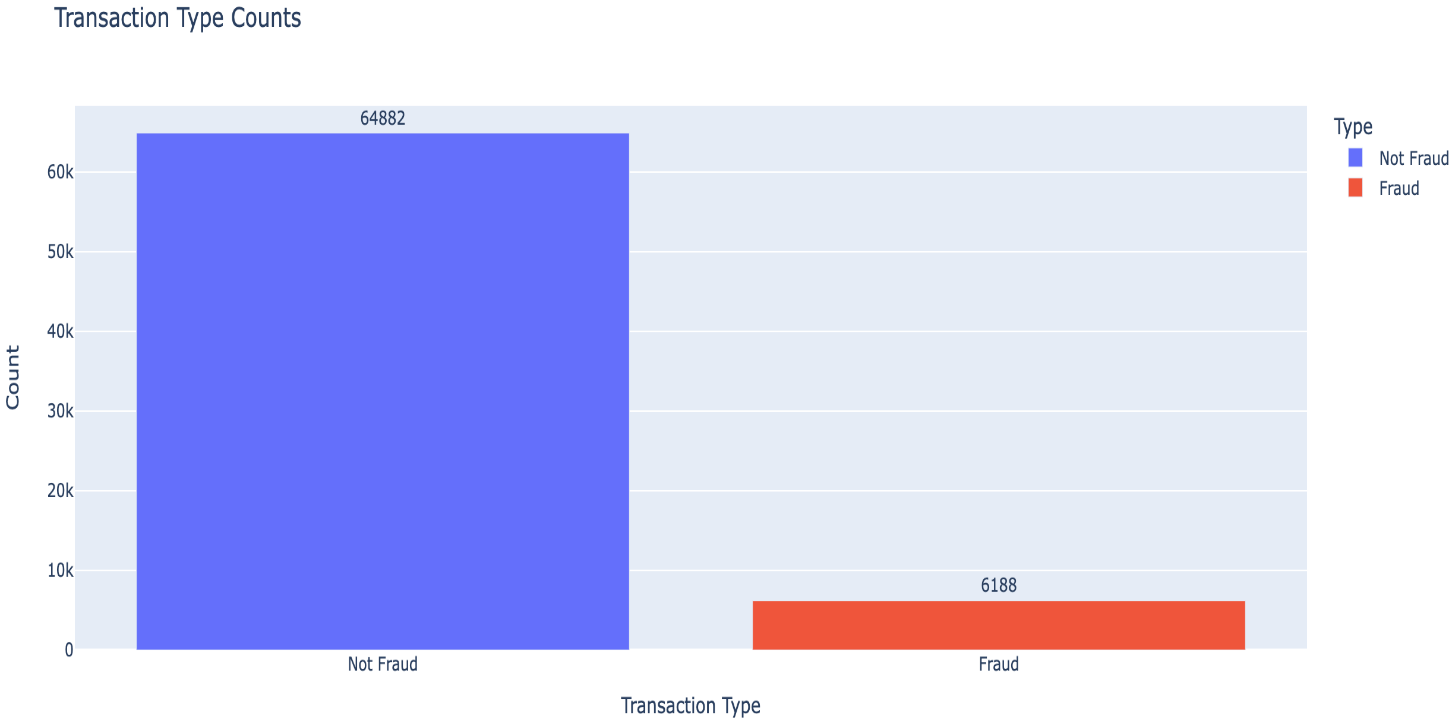
Feature Engineering – Principal Component Analysis (PCA)

Performed Hyper Parameter Tuning to determine the number of components to be used in the PCA for a 95% Variance.

```
pca = PCA()

# Fit PCA on the training data 'X_train'
pca.fit(X_train)

# Calculate the cumulative explained variance ratio from the training
data
cumulative_variance_ratio = np.cumsum(pca.explained_variance_ratio_)

# Determine the number of components needed to retain the desired
threshold of variance
threshold_variance = 0.95
n_components = np.where(cumulative_variance_ratio >=
threshold_variance)[0][0] + 1
```

```
Number of components to retain 95.0% variance: 137
```

Transaction Classes



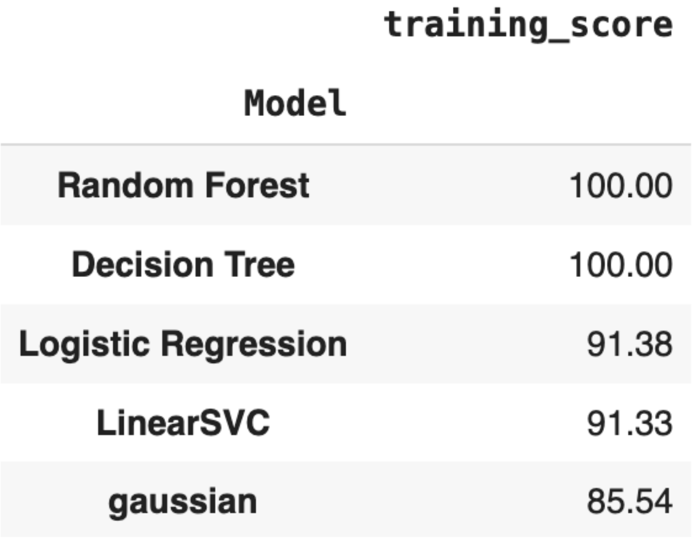
- Following the pre-processing, the Fraud classes are observed in the chart with most of the transactions reported as legitimate transactions.

- It is therefore important for the ML models to correctly predict the Fraud cases. Only the Accuracy of the models would not be a good judge of model performance.

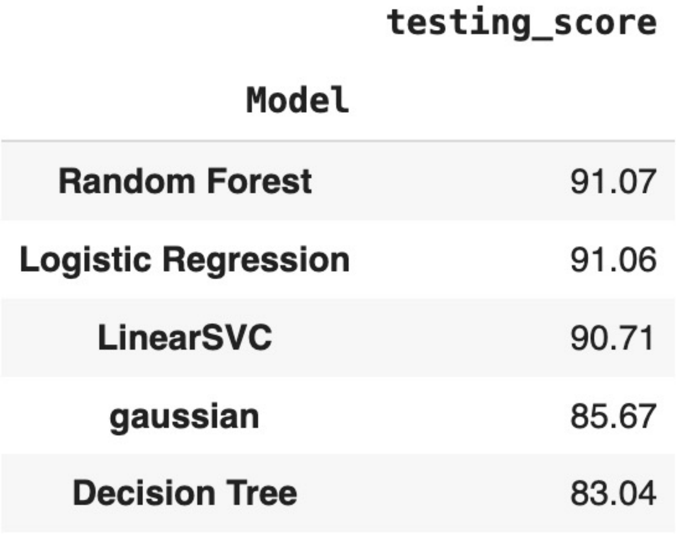
Machine Learning (ML models)

Machine Learning Models to be Deployed:

- Logistic Regression
- Decision Tree
- Random Forest
- GaussianNB
- LinearSVC



	training_score
Model	
Random Forest	100.00
Decision Tree	100.00
Logistic Regression	91.38
LinearSVC	91.33
gaussian	85.54



	testing_score
Model	
Random Forest	91.07
Logistic Regression	91.06
LinearSVC	90.71
gaussian	85.67
Decision Tree	83.04

The models generally perform better in training but in testing/predicting, most are unable to correctly classify the transactions.

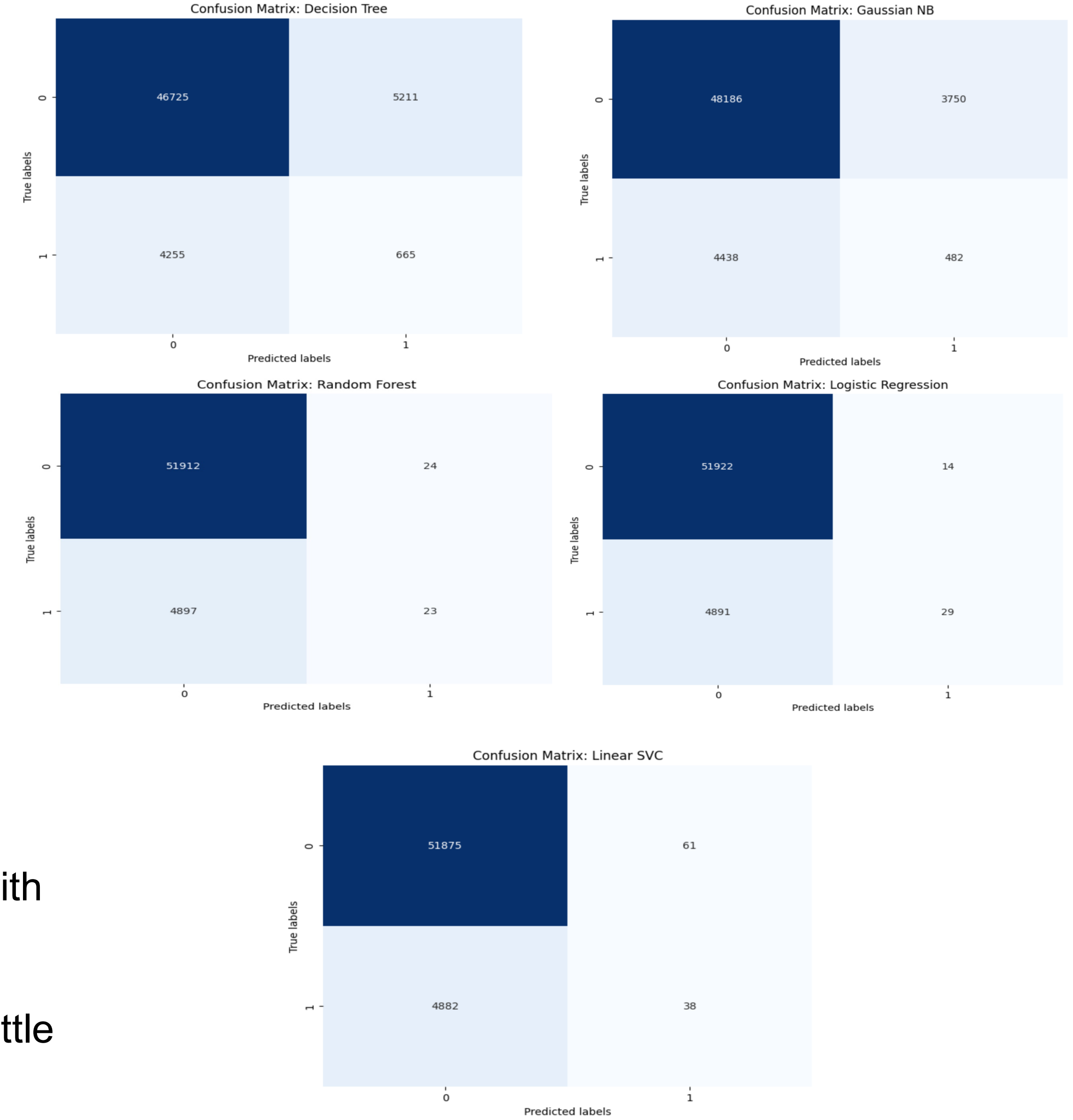
ML Models Performance – Cross Validation

Models	Average CV score	Previous Training Accuracy
Decision Tree	0.83	1.00 ⬇
Random Forest	0.91	1.00 ⬇
Logistic Regression	0.91	0.91 =
Gaussian NB	0.85	0.85 =
Linear SVC	0.91	0.91 =

From cross validation, the accuracy of the top models dropped with increasing misclassification of transactions.

For the other models, the accuracy stayed about the same with little improvements in classification.

Confusion Matrix - Training



CONCLUSION

Predictive Reliability: The Model Performance as indicated by the Confusion Matrix shows that a large degree of misclassification of fraudulent transactions occurs in the training data after cross validation.

The overfitting challenge of the top performing models in training is also to blame for the poor performance over the test data.

More work would need to be done on Feature Engineering and Hyper Parameter tuning to get better prediction of Fraudulent transactions by the ML models.