Credit Card Fraud Detection

By Armaan Das

Background

Credit card fraud occurs when an unauthorized person gains access to your information and uses it to make purchases. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items wrongfully. In this report, I will be discussing Credit Card Fraud Detection as a binary classification task and use various machine learning models to see how accurate they are in detecting whether a transaction is a normal payment or a fraud.

Problem Formulation

Input 30 PCA transformed anonymous transaction details

Output Binary class label indicating if the transaction is fraud (1) or legitimate (0).

Dataset *Kaggle Credit Card Fraud Detection dataset* which contains 284,807 transactions. This dataset is highly unbalanced with only 0.172% of all transactions labeled as fraud.

Data Split

- Training set - 70%
- Validation set - 15%

- Testing set - 15%

Approach

Baseline

The baseline model that I will be using is logistic regression without regularization so no hyperparameter is used. This will provide a simple reference point so that I can compare other complex models.

Algorithms

- Logistic Regression
- k-NN
- Random Forest

HyperParameters

- Logistic Regression - Regularization Strength
- k-NN - Number of neighbours (k) with Minkowski Distance
- Random Forest - Number of trees (n_estimators) and max_depth of trees.

I will be using **Grid Search** for **hyperparameter tuning** which will try all combinations of hyperparameter values to give the best one.

Evaluation Metric

Measure of Success The main goal of this project is to accurately predict fraud transactions while also minimizing false positives where real transactions are predicted to be a fraud.

I use the following metrics to evaluate the models:

1. **Precision** = $\frac{True\ positives}{True\ positives + False\ Positives}$
2. **Recall** = $\frac{True\ positives}{True\ positives + False\ Negatives}$
3. **F1 Score** = $\frac{Precision*Recall}{Precision + Recall}$

Result

After splitting the data into Training set, Validation set and Test set, the fraud cases represent only 0.17% of the total data. Since the dataset is so imbalanced, the following evaluation metrics (Precision, Recall and F1-Score) are very important to showcase the model's performance.

Baseline Logistic Regression (without regularization):

Precision: 0.0671

Recall: 0.8784

F1-Score: 0.1248

We can notice that even though this model has high recall (87.8%), it has very low precision (6.7%). This means that this model can correctly predict most fraud transactions but it also has very high false positives (real transactions are predicted to be a fraud).

Logistic Regression (with Regularization):

After hyperparameter tuning, the best regularization strength for this model was found to be **C = 7.09073.** Using this parameter:

Precision: 0.0672

Recall: 0.8784

F1-Score: 0.1249

We can notice this model has a bit better result than the baseline but these improvements are negligible. This suggests that logistic regression is not optimal for handling highly unbalanced dataset even with hyperparameter tuning.

k-NN:

After hyperparameter tuning, the best number of neighbors for this model was found to be **k = 5.** Using this 5 neighbours with Minkowski distance, we get:

Precision: 0.9500

Recall: 0.7703

F1-Score: 0.8507

We can notice that this model has very high precision (95%) while also having significantly high recall (77%). This makes this model have a high F1-score (85%) suggesting that k-NN is able to predict most fraud cases while also keeping the false-positives low.

Random Forest:

After hyperparameter tuning, the best **max_depth** was found to be "**None**" and the best number of trees was found to be **n_estimators = 137.** Using these parameters, we get:

Precision: 0.9630

Recall: 0.7027

F1-Score: 0.8125

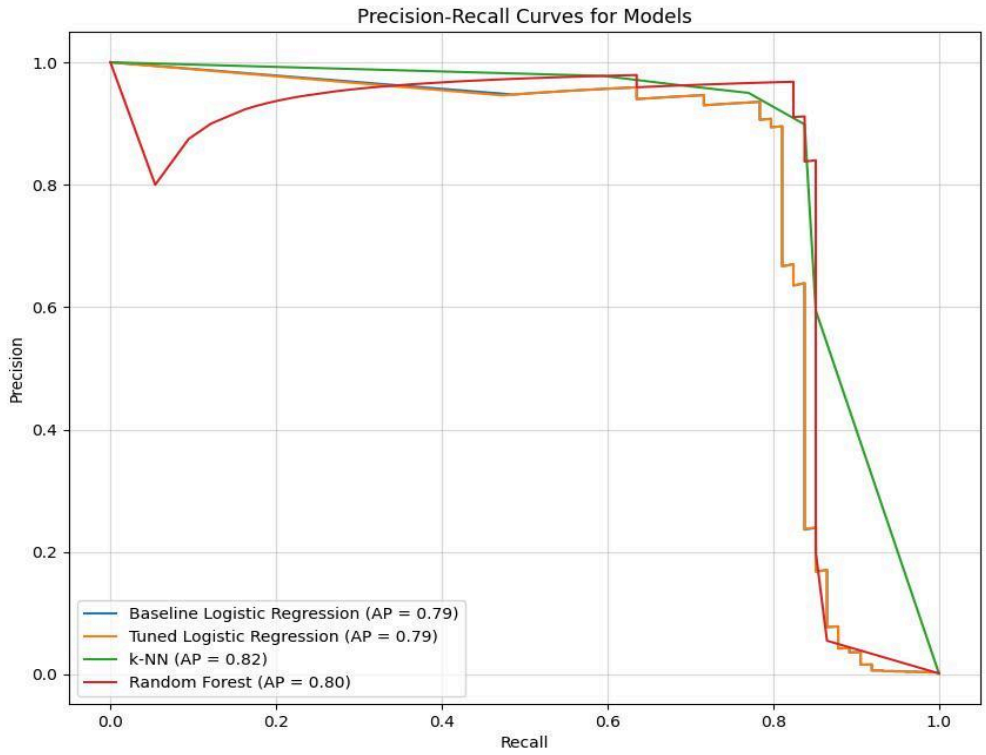
We can notice that this model has comparable performance to k-NN with higher precision (96%) but lower recall (70.3%). This makes this model have a lower F1-score (81%) than k-NN suggesting that most of the transactions predicted to be fraud turned out to be fraud cases, but it missed to predict some of the fraud cases.

Plots

The PR curve

The Precision-Recall (PR) curve below provides a visual representation of each model's balance between precision and recall.

The k-NN model showed the best balance of precision and recall, as its area under the PR curve is AP = 0.82.

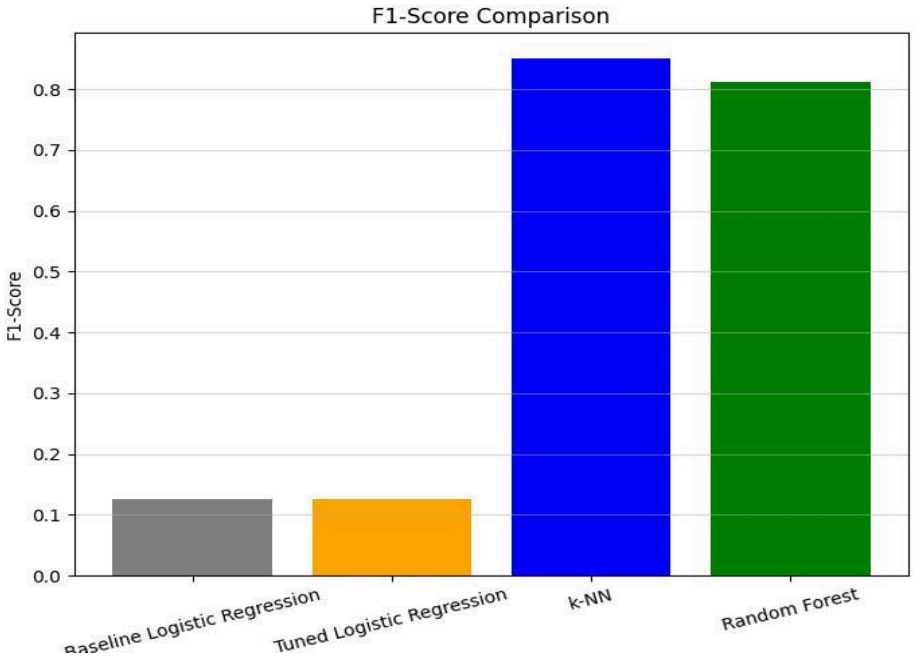


The F1-Score comparison

The bar chart below compares F1-Scores of all models, including the baseline logistic regression showing their performance.

k-NN has the highest F1-Score (0.85), and Random forest follows closely with a slightly lower F1-Score (0.81).

This comparison shows the far better performance of k-NN and random forest over logistic regression for this classification problem.



Conclusion

Compared to the baseline logistic regression, both k-NN and Random Forest models showed significantly high improvements.

- The precision went from 6.7% to 95% (approx.) for both models.
- The recall went down from 88% to 77% in k-NN and 70% in Random Forest but this decrease made precision and F1-score for both these models very high.

The results from the above study shows that k-NN and random forest models showed significantly better performance than logistic regression, with k-NN achieving the highest F1-Score by effectively balancing precision and recall. Random Forest showed better precision, making it better for scenarios where minimizing false positives is more important.

In our case, where we want a balance between predicting fraud transactions and having a low false positives rate, **k-NN is the most suitable model**.

References

- https://www.geeksforgeeks.org/machine-learning-algorithms/
- https://www.v7labs.com/blog/train-validation-test-set
- https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
- https://www.v7labs.com/blog/f1-score-guide
- https://scikit-learn.org/stable/supervised_learning.html
- https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://www.w3schools.com/python/matplotlib_bars.asp
- https://stackoverflow.com/questions/40758562/can-anyone-explain-me-standardscaler