COMP 3608 Project

GitHub Repository

https://github.com/MadMoose02/COMP3608-Project

Project Title

Credit Card Fraud Detection

Group Name

VKS

Group Members

Name	**Student ID**
Virendra Narine	816031395
Keshan Moosai	816031326
Shaniah Baldeo	816031341

1 Introduction

1.1 Problem

In the realm of financial security, credit card fraud detection remains a critical challenge. The difficulty lies in identifying increasingly sophisticated fraudulent activities, often disguised as legitimate transactions. This challenge necessitates robust machine learning analysis and data science techniques to develop a comprehensive solution.

Key stakeholders in this scenario include financial institutions, credit card companies, and cardholders, each with distinct needs. Financial institutions require a reliable system to not only protect their customers' assets but also maintain trust in their services. By implementing a machine learning model, they can detect early onsets of fraudulent transactions to minimise financial losses. Contrariwise, credit card companies seek to minimise losses due to fraudulent activities and to uphold the integrity of their payment systems. A machine learning model can significantly reduce the number of fraudulent transactions, protecting their revenue streams and safeguarding the reputation of their brands. In addition, cardholders rely on secure transactions to safeguard their financial well-being and peace of mind. The proposed solution offers real-time transaction monitoring, allowing for immediate intervention in case of suspected fraud.

This focus on credit card fraud detection addresses impactful real-world concerns. By implementing a machine learning model, all stakeholders benefit: financial institutions experience reduced losses and maintain customer trust, credit card companies minimise fraud and protect their brand reputation, and cardholders enjoy increased security and peace of mind.

1.2 Solution

The solution path lies in developing and deploying advanced machine learning models trained on historical transaction data. Here, three prominent techniques are particularly well-suited for this task: Logistic Regression, Random Forests and Artificial Neural Networks.

1.2.1 Logistic Regression

This technique excels at modelling the probability of an event, fraudulent transaction in this case, based on a set of independent variables, which are the transaction details. Its interpretable nature allows us to understand the impact of each feature on the model's decision.

1.2.2 Artificial Neural Networks

These powerful models can learn complex, non-linear relationships within the data, potentially capturing subtle patterns indicative of fraudulent behaviour.

1.2.3 Random Forests

This ensemble method combines multiple decision trees, offering robustness to outliers and handling high-dimensional data effectively, which is often the case with transaction details.

1.2.4 Summary

Implementing a systematic approach that includes data preprocessing, model training, and evaluation ensures a robust fraud detection system. This solution directly addresses the needs of all stakeholders. Financial institutions benefit from real-time alerts on suspicious transactions, enabling them to take swift action and minimise financial losses. Credit card companies experience a significant reduction in fraudulent transactions, protecting their revenue streams and brand reputation. Lastly, cardholders enjoy increased security with real-time transaction monitoring, allowing for immediate intervention in case of suspected fraud, safeguarding them from financial losses and the inconvenience of disputed charges. An effective fraud detection system enhances the overall stability and integrity of the global financial ecosystem. By proactively identifying and mitigating fraudulent activities, we reinforce the resilience of digital payment infrastructure in the face of evolving threats, fostering trust and confidence in the financial framework.

1.3 Objective

Employ machine learning techniques to identify a model that maximises recall in detecting fraudulent from genuine credit card transactions. The recall score is the most representative performance metric to use when evaluating a model that deals with fraud detection because the number of false negatives needs to be minimised. As such, the recall score will be used to evaluate all of the machine learning models employed in this analysis. The formula for maximising recall is given:

$$x^{*} = argmax \frac{TP}{TP + FN}$$

2 Experimental Design

2.1 Dataset Preprocessing

The dataset [1] sourced from Kaggle comprises 1,000,000 data points encompassing 8 distinct features as seen in Figure 1.0. Among these features, 7 are designated for training purposes, while the remaining one serves as the target variable. The training features encompass parameters such as the distance of the transaction from the cardholder's home, distance from their last transaction, ratio to median purchases, repeat retailer status, chip utilisation, PIN usage and if the transaction is an online order. The target feature, denoted as 'fraud' assumes binary values, with 1 signifying fraudulent transactions and 0 representing genuine transactions.

```
<class 'pandas.core.frame.DataFrame'>
Index: 1000000 entries, 330111 to 585903
Data columns (total 8 columns):
 #   Column                          Non-Null Count    Dtype
---  ------                          --------------    -----
 0   distance_from_home              1000000 non-null  float64
 1   distance_from_last_transaction  1000000 non-null  float64
 2   ratio_to_median_purchase_price  1000000 non-null  float64
 3   repeat_retailer                 1000000 non-null  float64
 4   used_chip                       1000000 non-null  float64
 5   used_pin_number                 1000000 non-null  float64
 6   online_order                    1000000 non-null  float64
 7   fraud                           1000000 non-null  float64
dtypes: float64(8)
memory usage: 68.7 MB
```

Figure 1.0: Information statistics of the Pandas DataFrame containing the loaded dataset

Upon closer inspection of the data's features, it was noted that the continuously distributed features such as distance_from_home, distance_from_last_transaction and ratio_to_median_purchase_price had values exceeding the median value in the order of thousands as depicted in Figure 1.1 below. As such, a logarithmic transformation will be applied on the continuous features so that they are normalised and easier to work with, which is shown in Figure 1.2 below.

	distance_from_home	distance_from_last_transaction	ratio_to_median_purchase_price
count	1000000.000000	1000000.000000	1000000.000000
mean	26.628792	5.036519	1.824182
std	65.390784	25.843093	2.799589
min	0.004874	0.000118	0.004399
25%	3.878008	0.296671	0.475673
50%	9.967760	0.998650	0.997717
75%	25.743985	3.355748	2.096370
max	10632.723672	11851.104565	267.802942

Figure 1.1: Information description of the continuous features of the dataset before transformation

	distance_from_home	distance_from_last_transaction	ratio_to_median_purchase_price
count	1000000.000000	1000000.000000	1000000.000000
mean	2.489694	1.006892	0.826152
std	1.201064	0.973274	0.574956
min	0.004863	0.000118	0.004390
25%	1.584737	0.259800	0.389114
50%	2.394960	0.692472	0.692005
75%	3.286310	1.471496	1.130231
max	9.271786	9.380261	5.593979

Figure 1.2: Information description of the continuous features of the dataset after transformation

2.1.1 Cleaning of Dataset

Upon analysing the data, visualisations of the data was created in order to determine the outliers within the dataset.

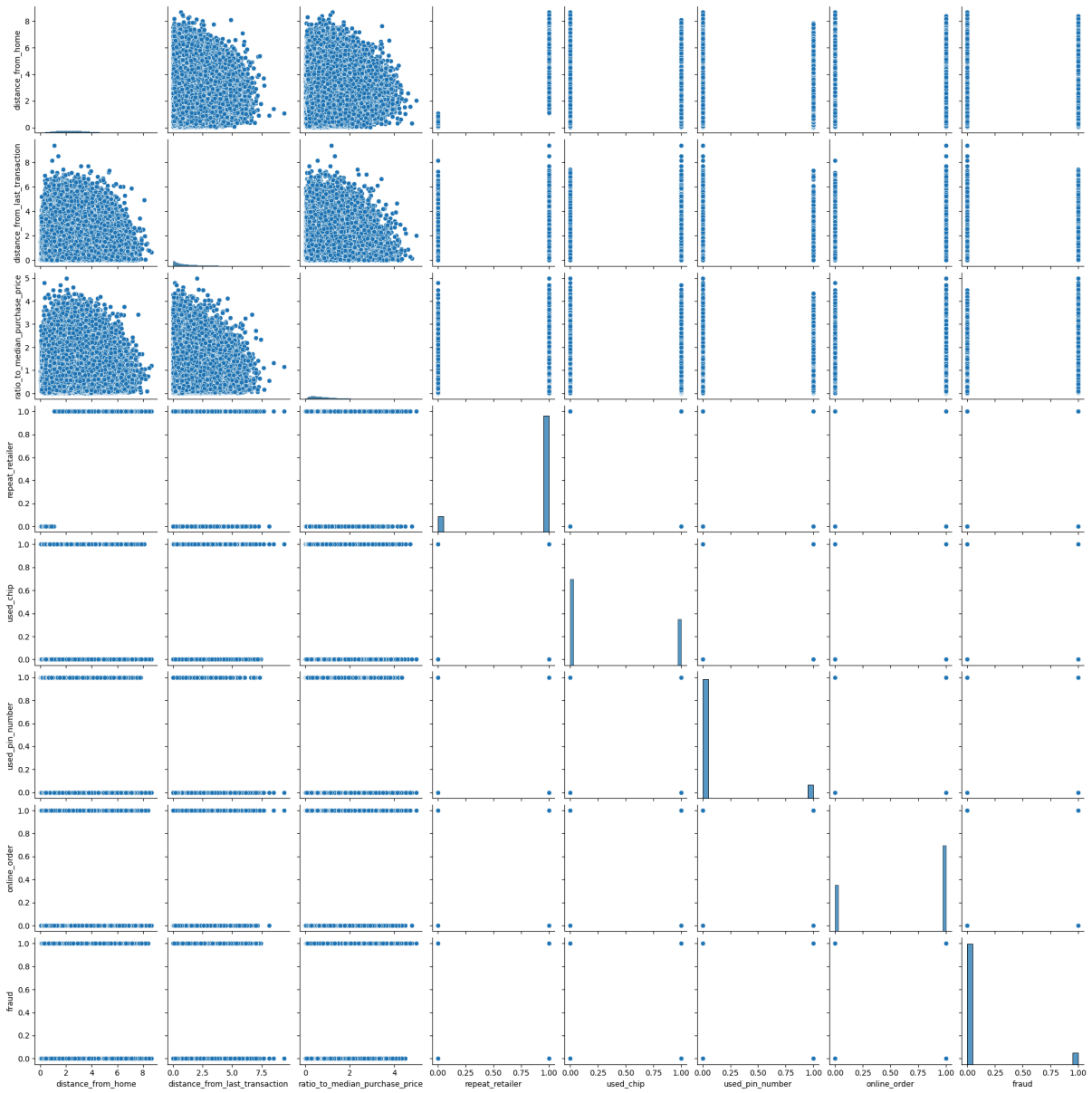

Figure 1.3: Pair plot showing distribution of features in dataset before removing outliers in the dataset

Figure 1.3 above shows the data to be very zoomed out and not evenly distributed along the axes of each subplot. This distribution of the data indicated the presence of outliers as they caused the subplots to become skewed in order to fit the outliers onto the subplot. The following pair plot shows the distributions of the attributes without the outliers.

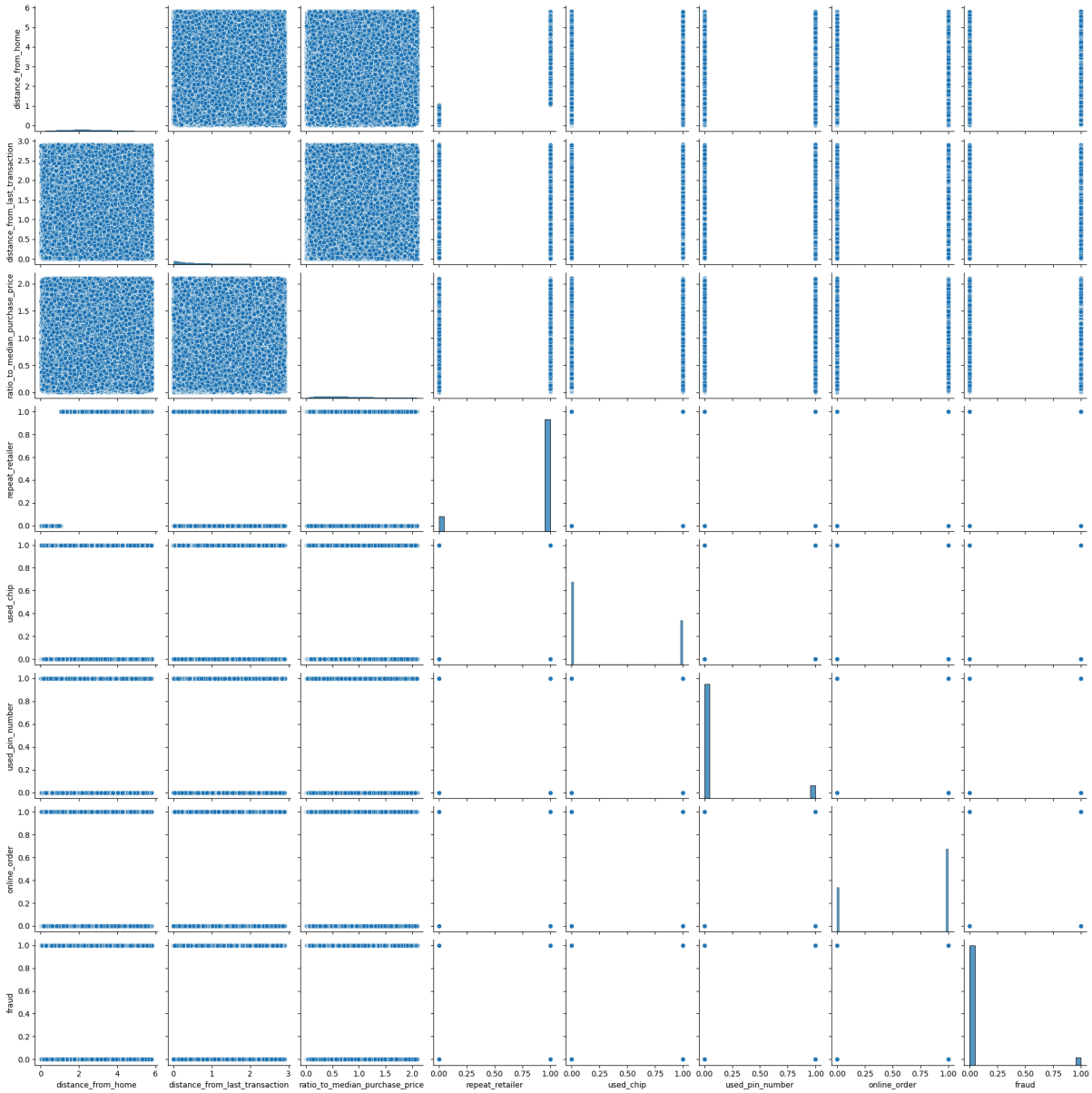

Figure 1.4: Pair plot showing distribution of features in dataset after removing outliers in the dataset

As shown in Figure 1.4, the absence of outliers allowed the data to be evenly distributed along the axes for each subplot.

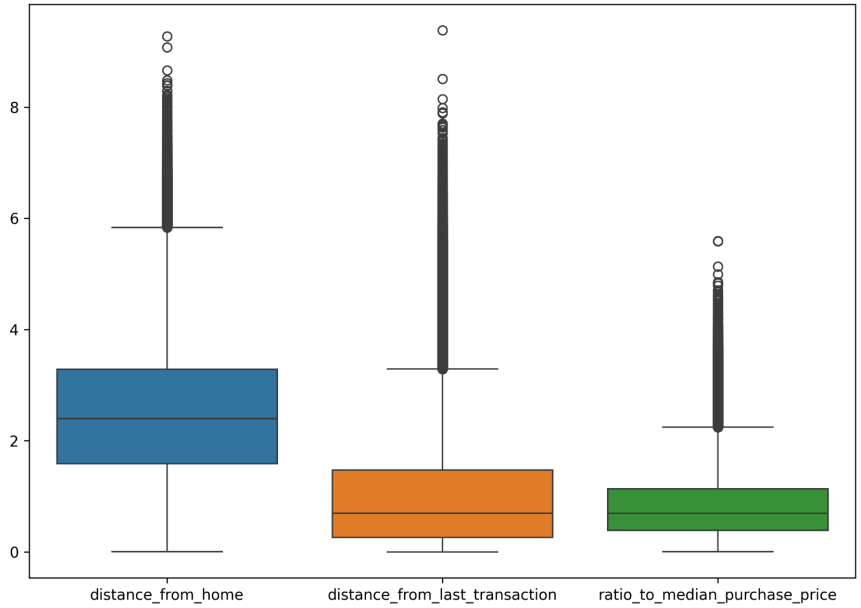

Figure 1.5: Box plot for the continuous features of the dataset with outliers

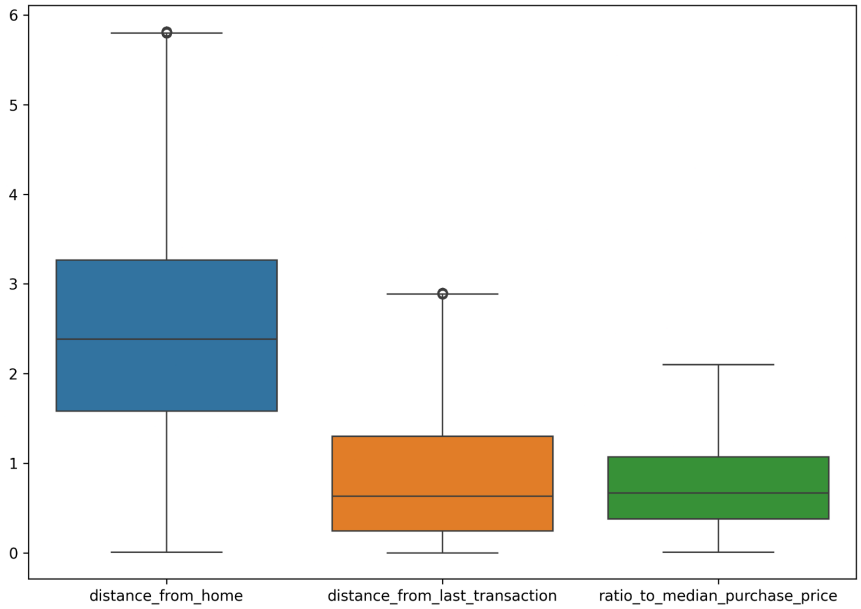

Figure 1.6: Box plot for the continuous features of the dataset after outliers have been removed

The method used to remove the outliers of these continuous features is the z-score normalisation with a threshold of 3. Any values with a computed z-score above 3 for any of the continuously distributed attributes were considered outliers and as a result, were removed from the dataset. The box plot in Figure 1.5 draws attention to the continuous features with outliers whereas Figure 1.6 illustrates the features after its removal.

2.1.2 Data Partitioning

Following the removal of outliers from the dataset, a total of 903,468 records remained. These data points were evenly distributed among each model for training, with partitions comprising 301,156 records each.

2.2 Logistic Regression

2.2.1 Base Model on Initial Partition 1

The model was trained and evaluated using partition 1 without any optimal hyperparameters to evaluate the model's recall score.

2.2.2 Grid Search Cross Validation

This optimization technique is employed to pinpoint the optimal hyperparameters that maximise the model's recall score. Initially, the user defines a parameter grid (see Figure 2.0) containing various values for each parameter: 'C' for regularisation amount, 'penalty' indicating regularisation strength, and 'solver' representing the optimizer adjusting parameters to refine predictions. Each unique combination undergoes testing against different subsets of the dataset to evaluate its performance in terms of recall. Ultimately, the combination of parameters yielding the highest recall is selected and returned.

```
param_grid = {
    'C': [0.1, 0.2, 0.3, 0.4],
    'penalty': ['l1', 'l2'],
    'solver': ['liblinear', 'saga']
}
```

Figure 2.0: Parameter grid used for Grid Search Cross Validation on the Logistic Regression model

2.2.3 Threshold Adjustment

The conventional threshold for a logistic regression model typically stands at 0.5. However, adjusting this threshold entails a trade-off between precision and recall. Hence, the threshold was carefully adjusted to minimise the amount of the false negatives, whilst maintaining a minimal impact on false positives. Through continuous trial and error, the optimal threshold was determined to be 0.4.

2.2.4 Tuned Logit with Grid Search Best Hyperparameters

The best hyperparameters discovered from the Grid Search Cross Validation are now applied to the Logistic Regression model and trained in accordance.

2.3 Artificial Neural Network

2.3.1 Base Network with 8 Neurons on Initial Partition 2

A simple neural network of one hidden layer containing 8 neurons was constructed and trained on partition 2 to evaluate its performance on recall.

2.3.2 Random Search Tuning

The employed optimization methodology entailed random search tuning, whereby a diverse set of neuron quantities is predetermined for utilisation by the random search tuner. This approach operates by randomly selecting neuron quantities from a predefined range (see Figure 2.1) and subsequently training a model on each specified quantity to ascertain its recall score. This iterative process is conducted based on an arbitrarily chosen number of trials set for the random search. Upon reaching the specified trial count, which was 6 trials in the context of this analysis, the neuron quantity yielding the highest recall score will then be used to tune the ANN to better classify the transactions in the dataset.

```
tf.keras.layers.Dense(
    hp.Choice('units',[32, 64, 128, 256, 512]),
    activation='relu'
)
```

Figure 2.1: Range of number of neurons used to construct the hidden layer during Random Search tuning of the ANN

2.3.3 K-fold Cross Validation

After determining the optimal amount of neurons to be used in the hidden layer, K-fold cross validation is performed on the neural network to determine its robustness of the model . This is done by testing the model on different portions of the data and averaging its performance.

2.3 Random Forest Classifier

2.3.1 Base Model on Initial Partition 3

The Random Forest classifier was initially trained and evaluated on partition 3 to determine how well it performs with regards to recall.

2.3.2 K-fold Cross Validation

Subsequently, K-fold cross validation was performed on the Random Forest classifier for partition 3. This procedure aimed to assess the model's recall performance across various train/test splits within partition 3.

2.3.3 Testing with Entire Dataset Reshuffled

To ensure that the performance of the Random Forest classifier was not skewed or biassed to partition 3 of the dataset, the entire cleaned dataset of over 900 thousand records was shuffled and used to train, as well as evaluate, the Random Forest classifier.

2.3.4 Base Model on Partitions 1 to 3 of Re-shuffled Dataset

Furthermore, following the shuffling of the cleaned 900 thousand records, the dataset was subdivided into three partitions to ascertain whether the model exhibited a preference for any particular subset. Subsequently, the Random Forest classifier underwent testing and evaluation on each partition to assess the recall score.

3 Related Work

Credit card fraud detection remains a critical challenge in the financial security landscape. Machine learning algorithms have emerged as a powerful tool to combat this issue, offering superior accuracy and adaptability compared to traditional rule-based systems. This section explores existing research on three prominent models utilised in our study: Logistic Regression, Artificial Neural Networks and Random Forests.

3.1 Logistic Regression

Logistic Regression offers a well-established approach for credit card fraud detection. Its interpretability allows for understanding the impact of individual features on fraud prediction, making it a valuable tool for building explainable models [2]. However, research also highlights potential limitations associated with LR, such as its sensitivity to outliers and potentially lower accuracy compared to more complex models [4].

3.2 Artificial Neural Network

Artificial Neural Networks (ANNs) provide a powerful alternative by learning complex, non-linear relationships within the data [2]. This capability allows them to potentially capture subtle patterns indicative of fraudulent activities that simpler models might miss. However, ANNs can be computationally expensive to train and may require careful parameter tuning to achieve optimal performance [4].

3.3 Random Forest Classification

Random Forest classifiers have gained significant traction due to their robustness and ability to handle high-dimensional data, a characteristic often encountered in credit card transactions [3]. Studies suggest that RF can achieve high accuracy in fraud detection, sometimes surpassing simpler models like Logistic Regression [3].

3.4 Conclusion

While these studies offer valuable insights into the effectiveness of individual models, a comparative analysis across various datasets and fraud detection scenarios remains crucial. Our research aims to contribute to this ongoing exploration by evaluating the performance of Logistic Regression, Artificial Neural Networks and Random Forests in our specific context. This evaluation will consider factors such as recall, computational efficiency, and the ability to adapt to evolving fraud patterns.

4 Experimental Results

4.1 Logistic Regression

4.1.1 Base Model

Table 1.1.1: Confusion matrix of base Logistic Regression model

	True Positive	True Negative
Predicted Positive	70036	751
Predicted Negative	2157	2345

Table 1.1.2: Classification report of base Logistic Regression model

	Precision	Recall	F1-Score	Support
Genuine	0.97	0.99	0.98	70787
Fraud	0.76	0.52	0.62	4502
Overall Accuracy	0.96			75289
Macro Average	0.86	0.76	0.80	75289

4.1.2 Grid Search Cross Validation

Best generalisation coefficient (C): 0.1
Best penalty type: L2
Best solver: liblinear
Expected Recall with best hyperparameters: 0.963

4.1.3 Tuned Model

Using the best hyperparameters discovered by the Grid Search Cross Validation, with a generalisation coefficient of 0.1, the L2 penalty and the liblinear solver, the following confusion matrix was generated.

Table 1.2.1: Confusion matrix of tuned Logistic Regression model

	True Positive	True Negative
Predicted Positive	69636	1151
Predicted Negative	1861	2641

Table 1.2.2: Classification report of tuned Logistic Regression model

	Precision	Recall	F1-Score	Support
Genuine	0.97	0.98	0.98	70787
Fraud	0.70	0.59	0.64	4502
Overall Accuracy	0.96			75289
Macro Average	0.84	0.79	0.81	75289

4.1.4 ROC Curve

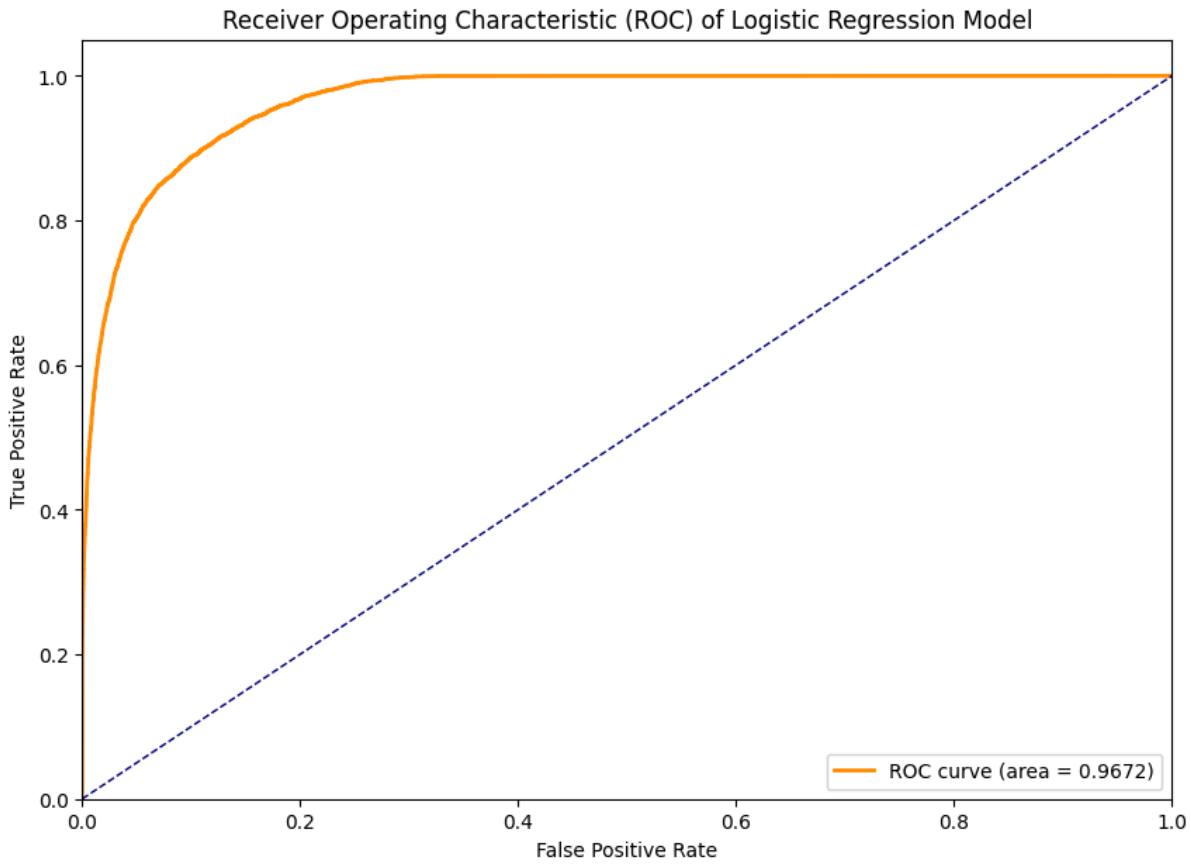

Figure 3.0: Graph showing Receiver Operating Characteristic (ROC) of the tuned Logistic Regression Model

4.2 Artificial Neural Network

4.2.1 Base Model

Aside from having 7 attributes, excluding the class label attribute, the neural network was created with a single hidden layer with an arbitrary number of neurons. In this case, the hidden layer was given 8 neurons to start off.

Table 2.1.1: Confusion matrix of base Artificial Neural Network

	True Positive	True Negative
Predicted Positive	70860	61
Predicted Negative	404	3964

The following classification report describes the detailed performance of the base Artificial Neural Network model.

Table 2.1.2: Classification report of base Artificial Neural Network

	Precision	Recall	F1-Score	Support
Genuine	0.99	0.99	1.00	70921
Fraud	0.98	0.91	0.94	4368
Overall Accuracy	0.99			75289
Macro Average	0.99	0.95	0.97	75289

4.2.2 Random Search

To find the most optimal number of neurons to use in the hidden layer of the ANN, the Random Search hyperparameter tuning method was used.

Table 2.1.3: Recall score obtained from various neuron configurations in the hidden layer of the ANN during Random Search tuning

Number of Neurons	Recall Score
32	0.964
64	0.971
128	0.966
256	0.958
512	0.976

4.2.3 K-fold Cross Validation

K-fold cross validation was performed on the ANN with 64 neurons in the hidden layer.

Table 2.1.4: Recall score obtained after each iteration of the k-fold cross validation on the ANN with 64 neurons in the hidden layer

Iteration Number	Recall Score
1	0.967
2	0.924
3	0.673
3	0.935
4	0.942
5	0.912
6	0.917
7	0.936
8	0.944
9	0.937

Overall Recall: 86.85 ± 11.56

4.2.4 Tuned Model

Table 2.2.1: Confusion matrix of tuned Artificial Neural Network model

	True Positive	True Negative
Predicted Positive	70036	751
Predicted Negative	2157	2345

Table 2.2.2: Classification report of tuned Artificial Neural Network model

	Precision	Recall	F1-Score	Support
Genuine	1.00	1.00	1.00	70921
Fraud	0.99	0.93	0.96	4368
Overall Accuracy	1.00			75289
Macro Average	0.99	0.97	0.98	75289

4.2.5 ROC Curve

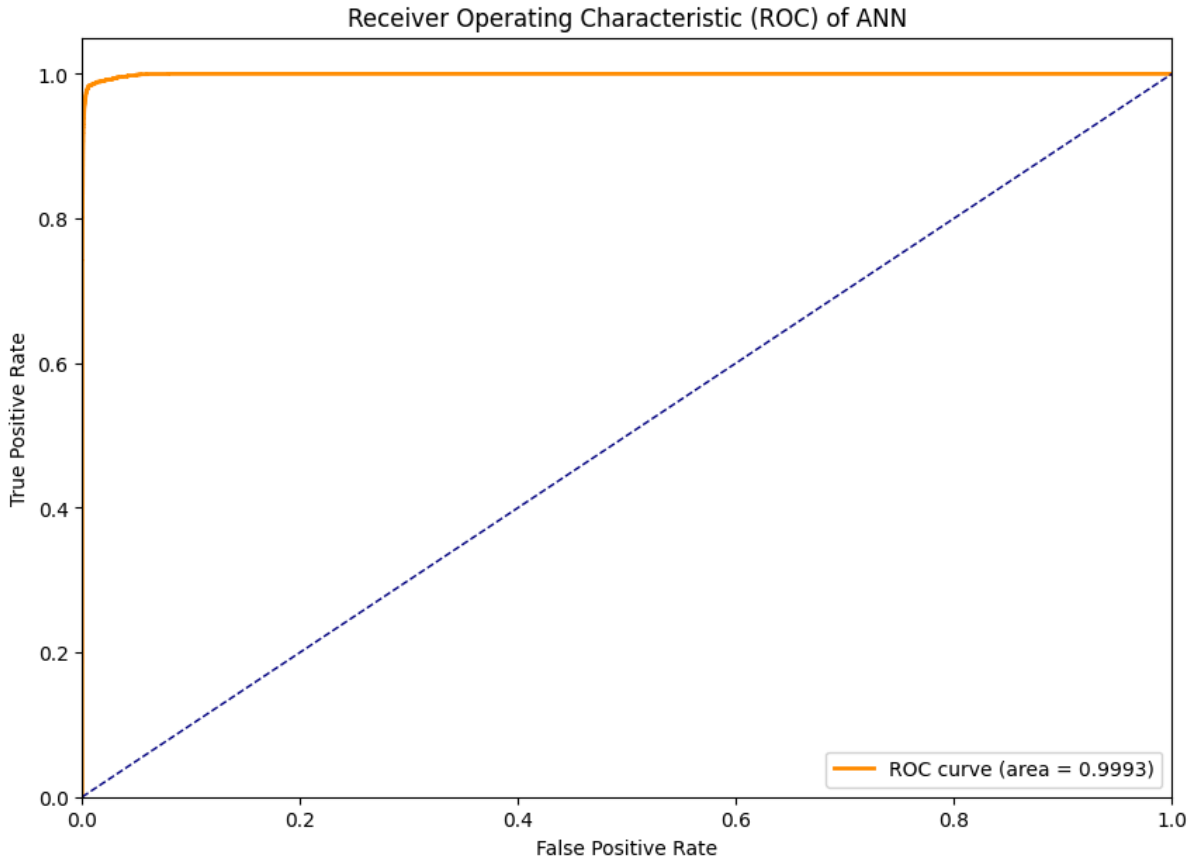

Figure 4.0: Graph showing Receiver Operating Characteristic (ROC) of the tuned Artificial Neural Network

4.3 Random Forest Classifier

4.3.1 Base Model on Initial Partition 3

Table 3.1.1: Confusion matrix of base Random Forest Classifier

	True Positive	True Negative
Predicted Positive	70837	0
Predicted Negative	1	4451

Table 3.1.2: Classification report of base Random Forest Classifier

	Precision	Recall	F1-Score	Support
Genuine	1.00	1.00	1.00	70837
Fraud	1.00	1.00	1.00	4452
Overall Accuracy	1.00			75289
Macro Average	1.0	1.0	1.0	75289

4.3.2 K-fold Cross Validation

Table 3.2.1: Recall Score per Iteration of the K-fold Cross Validation on the Random Forest classifier

Iteration Number	Recall Score
1	100.0
2	100.0
3	100.0
4	100.0
5	100.0
6	100.0

Overall Recall: 100.0 ± 0.0

4.3.3 Base Model on Re-shuffled Full Dataset

Table 3.3.1: Confusion matrix of base Random Forest Classifier on Re-shuffled Full Dataset

	True Positive	True Negative
Predicted Positive	255006	0
Predicted Negative	0	16035

Table 3.3.2: Classification report of base Random Forest Classifier on Re-shuffled Full Dataset

	Precision	Recall	F1-Score	Support
Genuine	1.00	1.00	1.00	255006
Fraud	1.00	1.00	1.00	16035
Overall Accuracy	1.00			271041
Macro Average	1.00	1.00	1.00	271041

4.3.4 Base Model on Partition 1 of Re-shuffled Dataset

Table 3.4.1: Confusion matrix of base Random Forest Classifier on Partition 1 of the Re-shuffled Full Dataset

	True Positive	True Negative
Predicted Positive	84969	0
Predicted Negative	0	5378

Table 3.4.2: Classification report of base Random Forest Classifier on Partition 1 of the Re-shuffled Full Dataset

	Precision	Recall	F1-Score	Support
Genuine	1.00	1.00	1.00	84969
Fraud	1.00	1.00	1.00	5378
Overall Accuracy	1.00			90347
Macro Average	1.00	1.00	1.00	90347

4.3.5 Base Model on Partition 2 of Re-shuffled Dataset

Table 3.5.1: Confusion matrix of base Random Forest Classifier on Partition 2 of the Re-shuffled Full Dataset

	True Positive	True Negative
Predicted Positive	85027	0
Predicted Negative	0	5319

Table 3.5.2: Classification report of base Random Forest Classifier on Partition 2 of the Re-shuffled Full Dataset

	Precision	Recall	F1-Score	Support
Genuine	1.00	1.00	1.00	85027
Fraud	1.00	1.00	1.00	5320
Overall Accuracy	1.00			90347
Macro Average	1.00	1.00	1.00	90347

4.3.6 Base Model on Partition 3 of Re-shuffled Dataset

Table 3.6.1: Confusion matrix of base Random Forest Classifier on Partition 3 of the Re-shuffled Full Dataset

	True Positive	True Negative
Predicted Positive	85090	0
Predicted Negative	2	5255

Table 3.6.2: Classification report of base Random Forest Classifier on Partition 3 of the Re-shuffled Full Dataset

	Precision	Recall	F1-Score	Support
Genuine	1.00	1.00	1.00	85090
Fraud	1.00	1.00	1.00	5257
Overall Accuracy	1.00			90347
Macro Average	1.00	1.00	1.00	90347

4.3.7 ROC Curve

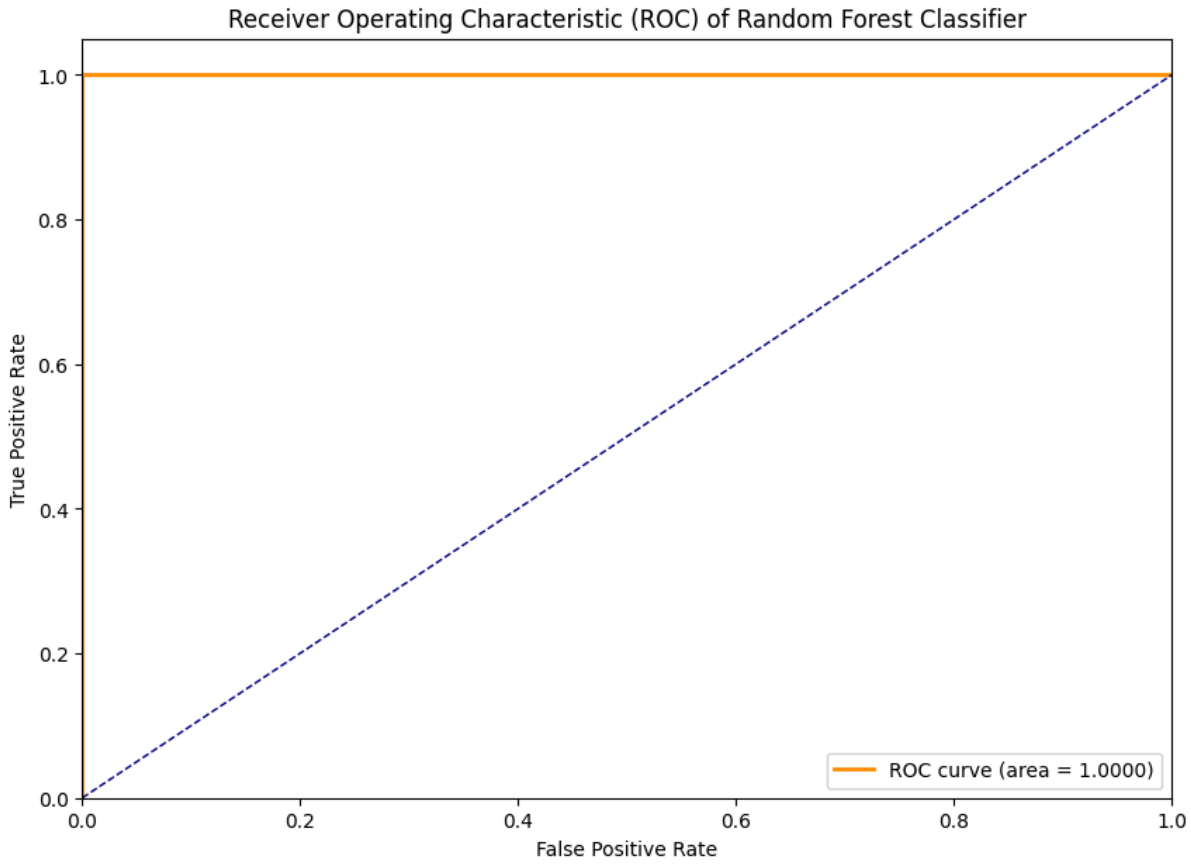

Figure 5.0: Graph showing Receiver Operating Characteristic (ROC) of the Random Forest Classifier

5 Discussion

The recall metric is used to evaluate the performance of machine learning models when dealing with fraud detection in transactional data because it is a measure of the ratio of the true positive cases out of the total true positive cases and false negative cases. As such, a high recall score is indicative of a model that is able to accurately classify the transactions in the dataset with minimal misclassification of transactions that were classified as genuine transactions but were actually fraudulent.

5.1 Logistic Regression

The base Logistic Regression model obtained a recall score of 0.76 with 2157 false negatives and 15 false positives out of 75 thousand test records. This indicates that the logistic regression model is fairly good at correctly identifying fraud. However, with the introduction of grid search cross validation and threshold tuning it can further improve the recall of the model. The grid search cross validation would combine the various hyperparameters defined in the parameter grid (see Figure 2.0) to find the most optimal configuration of the Logistic Regression model that maximises the recall score.

The most optimal hyperparameters were found to be C = 0.1, penalty = L2, solver = liblinear and an optimal decision threshold of 0.4. The combination of these parameters produced a recall score 0.79 with reduced false negative and false positive cases to 1861 and 1151 respectively.

5.2 Artificial Neural Network

The base Artificial Neural Network (ANN) was created with 8 neurons in the hidden layer with ReLU activation. The base neural network achieved a recall of 0.95, with a total of 404 false negatives and 61 false positives out of 75 thousand test records. These figures suggest that the ANN is effective at capturing the majority of positive instances and with some sensitivity tuning, can achieve much better results.

After performing a Random Search cross validation on the ANN, the most optimal number of neurons in the hidden layer was found to be 512 neurons at 0.98 recall. However, the second most optimal number of neurons was found to be 64 neurons at 0.97 recall. It was decided that for the purpose of efficiency and reducing computation and model complexity, the second best configuration was used in the tuned ANN model. After modifying the network to incorporate this finding, as well as performing k-fold cross validation on the model, the ANN achieved a recall of 0.97 with a total of 294 false negatives and 57 false positives out of 75 thousand test records.

5.3 Random Forest Classifier

The base random forest classifier that was tested using partition three of the initial dataset, produced a recall score of 1.0 with 1 false negative and 0 false positives out of the 75 thousand test records. This indicated the model's superior ability to correctly classify fraudulent credit card transactions from genuine ones in comparison to the previous models.

Furthermore, a k-fold cross validation was performed on partition three of the initial dataset. This continued to provide a recall of 1.0. However, to provide further validation of the model's performance, the model was trained and evaluated on the shuffled version of the entire dataset, once again achieving a perfect recall of 1.0. Moreover, the shuffled dataset was partitioned into three subsets for training and evaluation, the model maintained its flawless performance across each partition.

6 Conclusion

This research investigated the efficacy of machine learning models in detecting fraudulent credit card transactions. By implementing Logistic Regression, Random Forests, and Artificial Neural Networks, we aimed to identify a model that excelled in capturing fraudulent activities by maximising recall. Our evaluation process revealed that Random Forests outperformed the other models, achieving the highest recall value. This signifies its superior ability to identify a vast majority of fraudulent transactions within the dataset, minimising the number of false negatives. The successful implementation of a Random Forest classifier for credit card fraud detection may be the most successful detector of fraudulent transactions in financial frameworks.

7 Reflection

“Grid search cross validation in optimising the hyperparameters of the logistic regression model was one of the techniques that stood out most to me. I discovered that grid search cross validation is more robust compared to random search with regards to hyper parameter tuning. However, this method comes at cost because it is much more computationally intensive and time consuming due to its exhaustive exploration of hyperparameter combinations . Furthermore, I've come to appreciate the critical role of scaling data and removing outliers, as these practices can markedly enhance a model's performance.”

~V. Narine

“I learnt that there are some cases when balancing the dataset may not always be the most optimal approach to improve a machine learning model’s performance. In the context of this project, our models were able to perform better with the imbalance in our dataset as there was a severe underrepresentation of fraud samples in our transaction dataset. Alternative measures were taken to account for this imbalance such as outlier detection during data cleaning and rigorous hyperparameter tuning in each machine learning model.”

~ K. Moosai

“This project served as a catalyst for a significant shift in my perspective on the importance of data cleaning in machine learning. Initially, there was a misconception that the provided data was sufficiently clean, leading to prioritisation of other aspects of the analysis. However, the project unveiled the detrimental nature of data inconsistencies, demonstrating how even subtle outliers can significantly skew the results. This experience has instilled in me a profound appreciation for the meticulous attention required during data cleaning. Moving forward, I recognize data cleaning as the cornerstone of any robust machine learning project. I will ensure a more rigorous approach in future endeavours, acknowledging the profound impact it has on the validity and trustworthiness of the entire analytical process.”

~ S. Baldeo

8 References

[1] “Credit Card Fraud,” www.kaggle.com. https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud

[2] D. Alonge, “CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING (AI),” Feb. 2024, Accessed: May 01, 2024. [Online]. Available: https://www.researchgate.net/publication/377894904_CREDIT_CARD_FRAUD_DETECTION_USING_MACHINE_LEARNING_AI

[3] S. Das, R. Sulaiman, and U. Butt, “Comparative Analysis of Machine Learning Algorithms for Credit Card Fraud Detection,” Dec. 2023, Accessed: May 01, 2024. [Online]. Available: https://www.researchgate.net/publication/378746853_Comparative_Analysis_of_Machine_Learning_Algorithms_for_Credit_Card_Fraud_Detection

[4] U. Sam, “Credit Card Fraud Detection Using Machine Learning Algorithms,” Dec. 2023, doi: https://doi.org/10.13140/RG.2.2.14806.63044.