Statistical Analysis of Machine Learning Algorithms for Fraud Detection in Bank Transactions

Siddharth Jain	Ketan Chaudhary	Pranav Chougule
1226137070	1226082301	1225934595

May 4th, 2023

Abstract

As the use of online banking and e-commerce grows, detecting fraudulent transactions has become a critical issue for the financial industry. This project evaluates the performance of five popular machine learning algorithms, namely logistic regression, random forest, gradient boosting, Recurrent Neural Network – LSTM, and Neural Network, in detecting fraud in a bank transaction dataset. Preprocessing techniques, such as SMOTE, and feature importance analysis are used to develop effective machine learning models. Various evaluation metrics, including F1 score, recall, precision, ROC curve, and accuracy, are used to comprehensively assess the models' performance. Our study finds that the Recurrent Neural Network - LSTM and Gradient Boosting outperforms the other models, with a perfect score on all evaluation metrics. Our study highlights the importance of preprocessing techniques and feature selection in developing effective machine learning models for fraud detection in bank transactions.

1. Introduction

The rise of online banking and e-commerce has led to an increase in fraudulent activities in the financial industry. Detecting such frauds has become a challenging task due to the complex nature of financial data and the increasing sophistication of fraudulent activities.

This project evaluates the performance of five popular machine learning algorithms, including logistic regression, random forest, gradient boosting, Recurrent Neural Network - LSTM, and Neural Network, to detect fraud in a bank transaction dataset. We also investigate the impact of preprocessing technique, one-hot encoding, with SMOTE and perform feature importance analysis to identify the most significant features. Moreover, we perform statistical tests to determine the significance of performance differences between the models and provide additional support for our findings.

The report is organized into different sections. We provide a review of related work in fraud detection using machine learning. We describe the dataset used in this project and the preprocessing steps taken to prepare the data for analysis. We present the results of our analysis, including the evaluation of machine learning algorithms, feature importance analysis, and statistical tests. Finally, we conclude the report with a summary of our findings and suggestions for future research in this area.

2. Methods

I. Dataset Description

A. Size and Structure of the Dataset

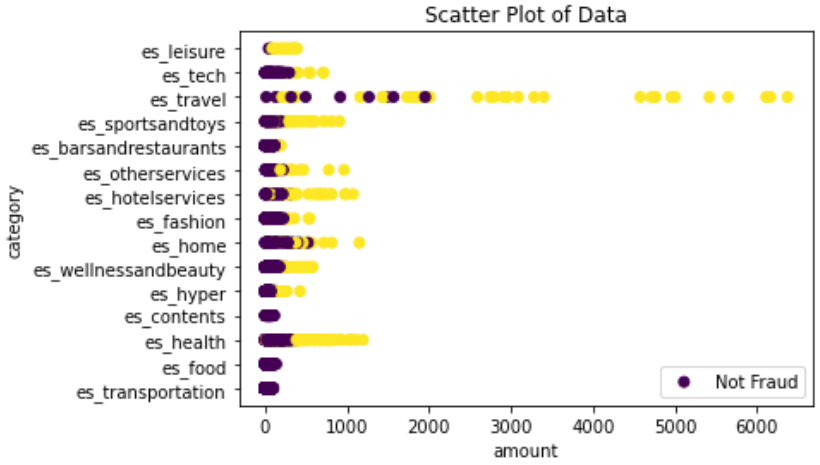
The dataset used in this project is a CSV file with 594,644 rows and 10 columns later reduced to 30000 entries with 5 columns. The data represents bank transactions, and each row corresponds to a single transaction.

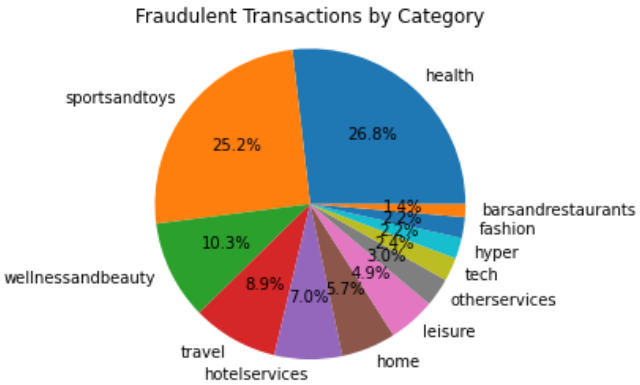
step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
2	C42240331	0	F	28007	M2122776122	28007	es_home	1167.18	1
4	C44513455	0	F	28007	M480139044	28007	es_health	833.47	1
9	C44513455	0	F	28007	M980657600	28007	es_sportsandtoys	350.56	1
11	C42240331	0	F	28007	M1294758098	28007	es_leisure	93.08	1
12	C42240331	0	F	28007	M480139044	28007	es_health	571.59	1

B. Features of the Dataset

The columns in the dataset (Dataset Link [1]) are as follows:

1. step: This feature represents the day from the start of simulation. It has 180 steps so simulation ran for virtually 6 months, customer: A unique identifier for the customer, age: Categorized age, zipcodeOri: The zip code of the customer's location, merchant: A unique identifier for the merchant, zipMerchant: The zip code of the merchant's location
2. fraud: A binary target variable indicating whether the transaction was fraudulent (1) or not (0).





II. Preprocessing Techniques

A. One-Hot Encoding

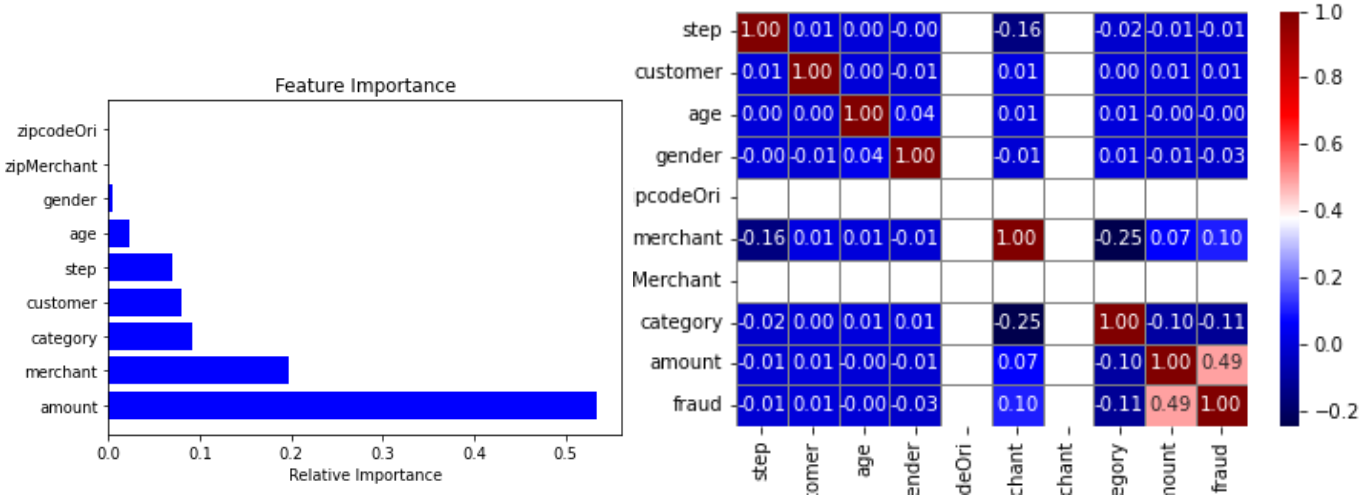
One-hot encoding is a technique used to convert categorical variables into a set of binary variables. It creates a new binary column for each unique category in the original data, and assigns a 1 or 0 to each column depending on whether the category is present or absent for that observation.

One-hot encoding can be useful because it preserves the information in the original data without introducing any unintended mathematical relationships.

B. Feature Importance

In our project, we have already selected 7 features to train the model. However, we noticed that the probability of 3 of those features was less. Therefore, we decided not to train our model on those features.

As for my current approach, I have used a random forest classifier to identify the feature importance in my dataset. This approach can be useful to identify the most informative features in the dataset.



C. SMOTE

To address the high imbalance and skewed nature of the dataset, I applied the Synthetic Minority Over-sampling Technique (SMOTE) to synthesize fraudulent transactions. It creates new data points by selecting a minority class instance and computing the k-nearest neighbors for that instance.

The SMOTE technique helped us to address the issue of class imbalance and improve the performance of the machine learning models in detecting fraudulent transactions. By synthesizing new fraudulent transactions, we were able to increase the number of positive examples in the dataset.

	Before SMOTE	After SMOTE
Fraudulent	370	29647
Non-Fraudulent	29630	29647
Total	30000	59294

IV. Model Selection

A. Five Machine Learning Algorithms

1. **Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM)**: RNNs with LSTMs are well-suited for analyzing sequential data such as transaction histories or user behavior patterns to identify patterns and trends that may indicate fraudulent activity. The model can be trained on past transaction data to predict whether new transactions are likely to be fraudulent or not.
   RNN model architecture. The model is a Sequential model, which means that layers are added sequentially to the model. The architecture consists of three layers:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 4, 1)              0

 lstm (LSTM)                 (None, 64)                16896

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 16,961
Trainable params: 16,961
Non-trainable params: 0
```

2. **Neural Network (NN)**: Neural networks can be trained to classify bank transactions as fraudulent or not based on input features such as transaction amount, merchant, category, and customer information. The network can be optimized to minimize the difference between the predicted and actual target values, resulting in a model that can effectively identify fraudulent transactions.
   NN model architecture. The architecture consists of two dense layers and one dropout layer:

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_2 (Dense)             (None, 64)                320

 dropout (Dropout)           (None, 64)                0

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 385
Trainable params: 385
Non-trainable params: 0
```

3. **Logistic Regression**: Logistic regression is a linear model that can be used to predict the probability of a transaction being fraudulent based on input features. The model can output a probability score, which can be used to filter and rank predictions based on confidence.

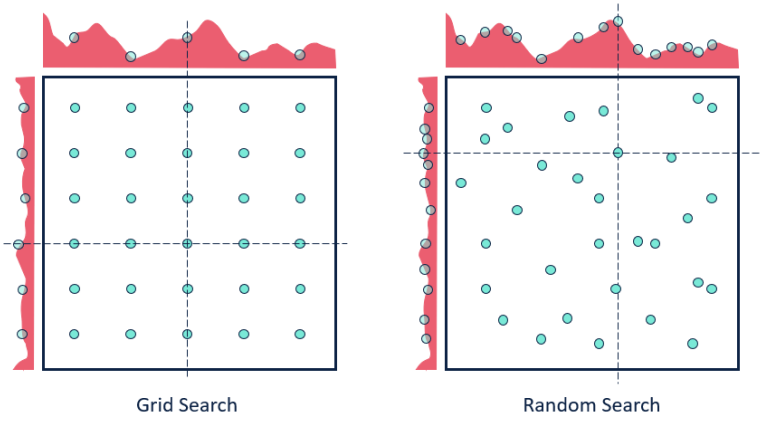
4. **Random Forest**: Random Forest is a decision tree-based algorithm that can be used to identify patterns in the data that may indicate fraudulent activity. The algorithm can handle missing data and non-linear relationships between input and target variables, making it suitable for use with bank transaction data.

5. **Gradient Boosting**: Gradient Boosting is an ensemble learning algorithm that builds multiple decision trees sequentially, where each tree corrects the errors of the previous tree. The algorithm is well-suited for handling complex data with non-linear relationships between input and target variables, and it provides feature importance scores for interpretation. This can be helpful in identifying which input features are most important in predicting whether a transaction is fraudulent or not.

Overall, these five algorithms were chosen for their versatility and effectiveness in handling different types of data and classification problems, including fraud detection. The comparison of these algorithms on the same dataset provides valuable insights into the strengths and weaknesses of each algorithm, and helps in selecting the best algorithm for a given problem.

B. Hyperparameter Tuning Approach

Hyperparameter tuning is an important step in machine learning model building, as it can significantly impact the performance of the model. Grid search is a computationally expensive method but can often lead to better performance than random search or manual tuning. It's important to keep in mind that hyperparameter tuning was done on a separate validation set, and not on the test set, to avoid overfitting to the test set.



V. Evaluation Metrics

When evaluating the performance of a classification model for fraud detection, it's important to consider different evaluation metrics to gain a comprehensive understanding of how well the model is performing.

A. **F1 Score**: The F1 score is the harmonic mean of precision and recall. It's a good metric to use when the target class is imbalanced, which is often the case in fraud detection, and can help evaluate the overall performance of the model.

B. **Recall**: Recall measures the proportion of actual positive cases that are correctly identified by the model. Recall is an important metric as it indicates how well the model is able to detect all the fraudulent transactions, even at the cost of misclassifying some non-fraudulent transactions as fraudulent.

C. **Precision**: Precision measures the proportion of positive cases that are truly positive among all cases classified as positive by the model. In fraud detection, precision is important as it indicates how many of the flagged transactions are actually fraudulent, and not just false positives.

D. **ROC Curve**: The Receiver Operating Characteristic curve is a plot of the true positive rate (recall) against the false positive rate (1 - specificity) at different classification thresholds. It provides a visual representation of the trade-off between recall and precision and helps in selecting an optimal threshold.

E. **Accuracy**: Accuracy measures the proportion of correctly classified cases among all cases. While accuracy is a common metric, it may not be the best metric to use in fraud detection as it can be misleading when the target class is imbalanced.

VI. Implementation Details

A. Tools and Software:

For this project, I used Python programming language along with various libraries such as Scikit-learn, Pandas, Matplotlib, and Seaborn. Scikit-learn was used for implementing the machine learning algorithms, while Pandas was used for data manipulation and preprocessing. Matplotlib and Seaborn were used for data visualization.

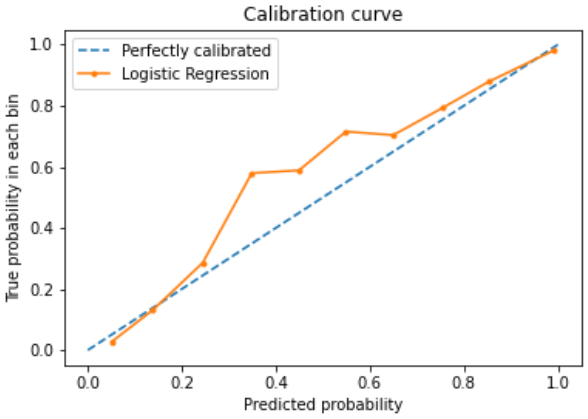
B. Computing Environment and Resources:

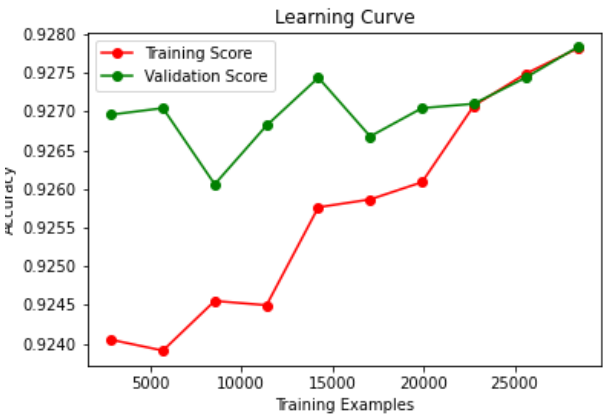
The project was implemented on a laptop with the following specifications: Ryzen 9 processor, 16GB of RAM, and Radeon 6900 HS graphic card. The programming environment used for the project was Visual Studio Code and Jupyter Notebook kernel.

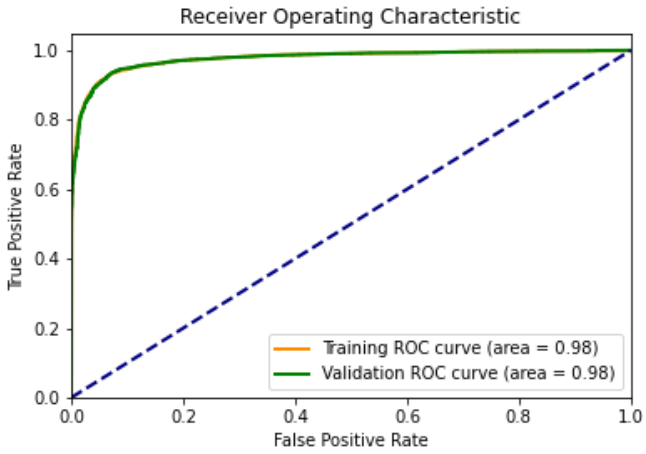
3. Results.

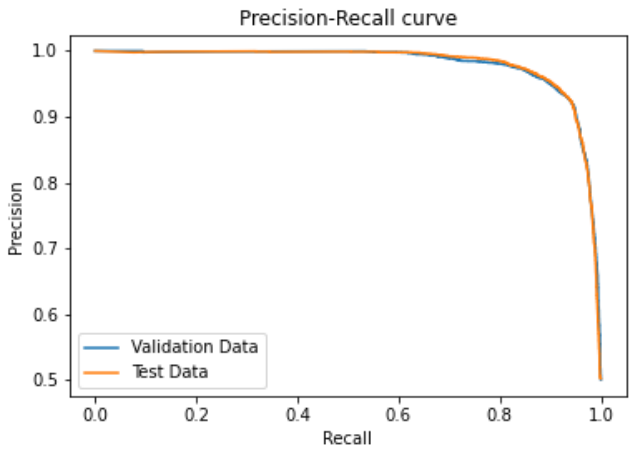
a. Metrics

i. Logistic Regression

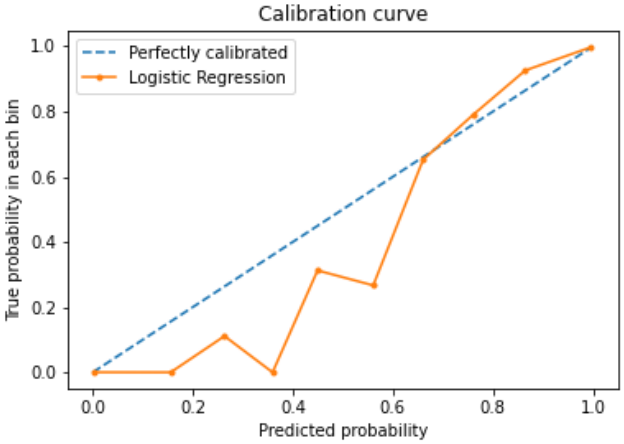


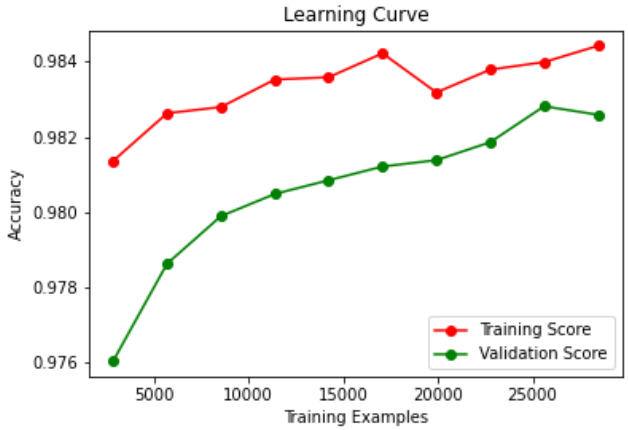


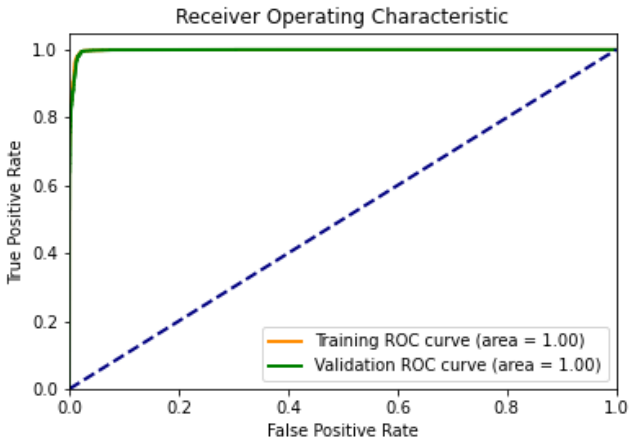


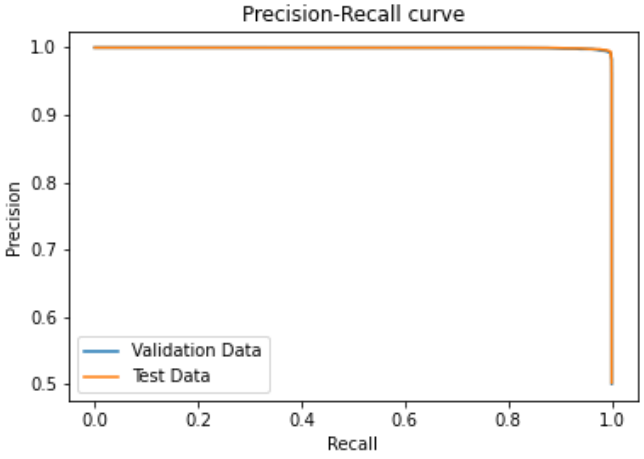


ii. Random Forest

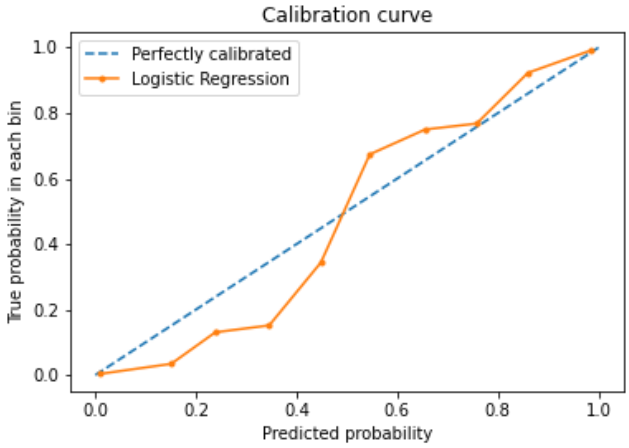


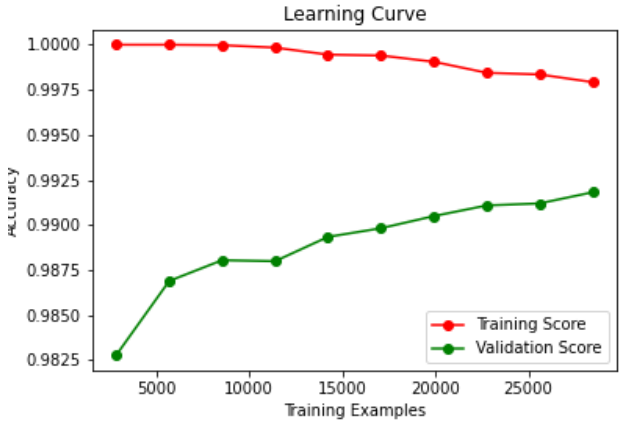


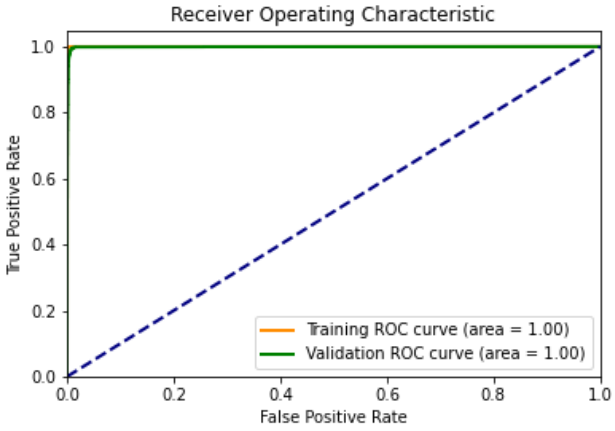


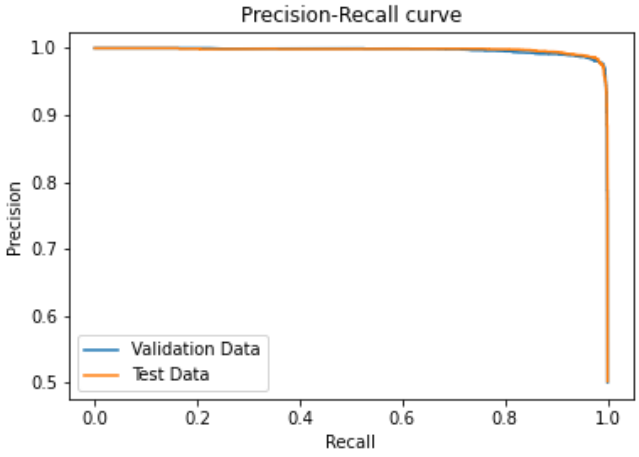


iii. Gradient Boosting

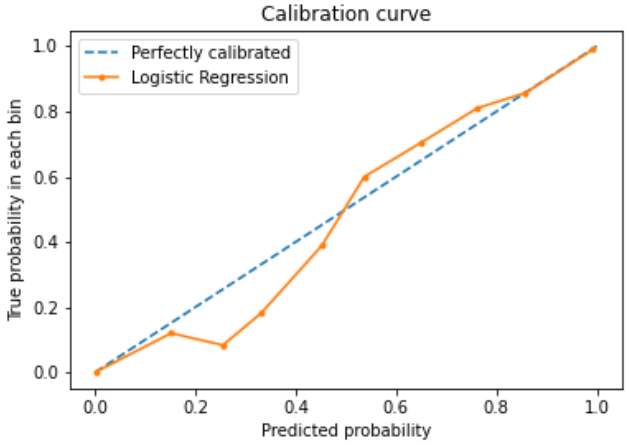


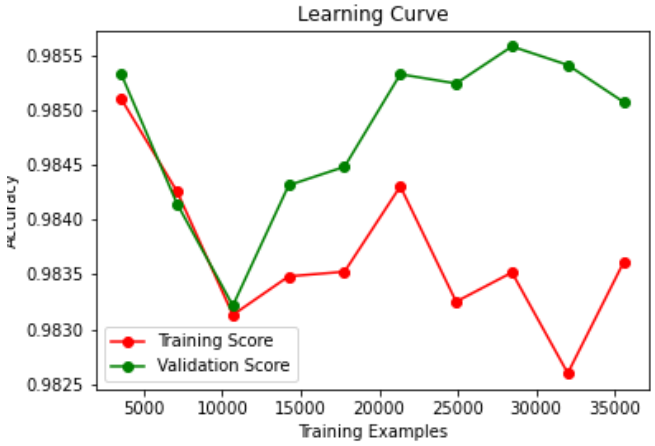


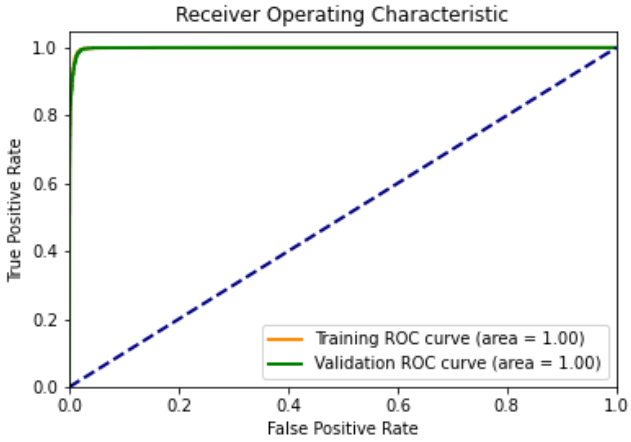


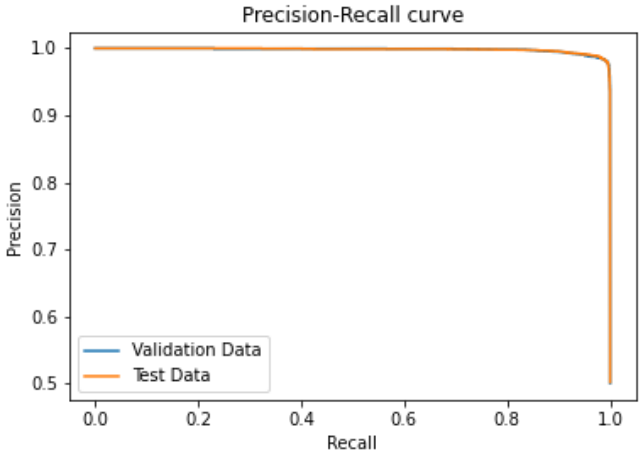


iv. Recurrent Neural Network

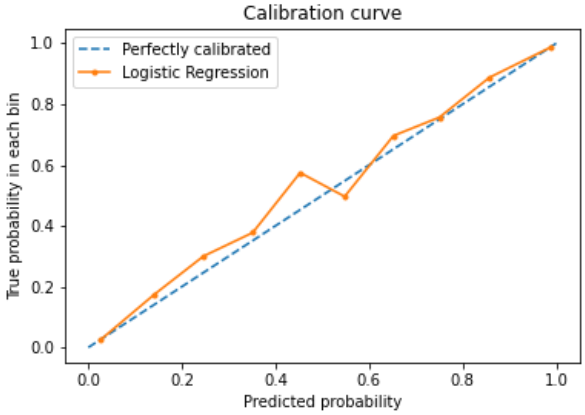


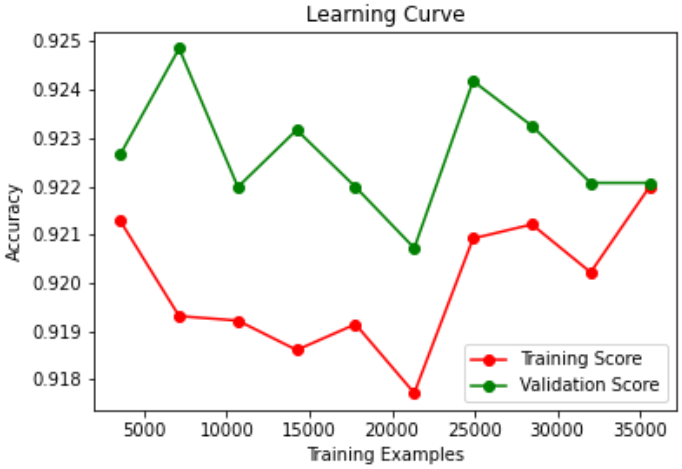


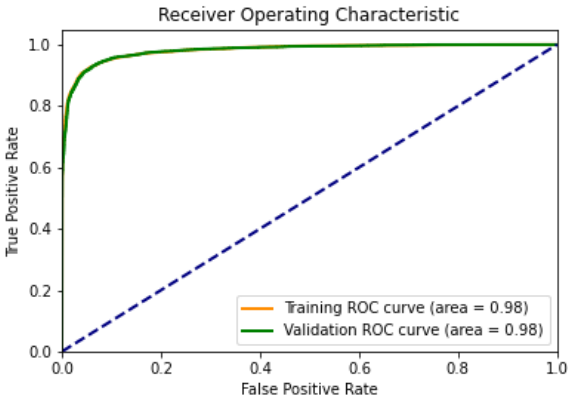


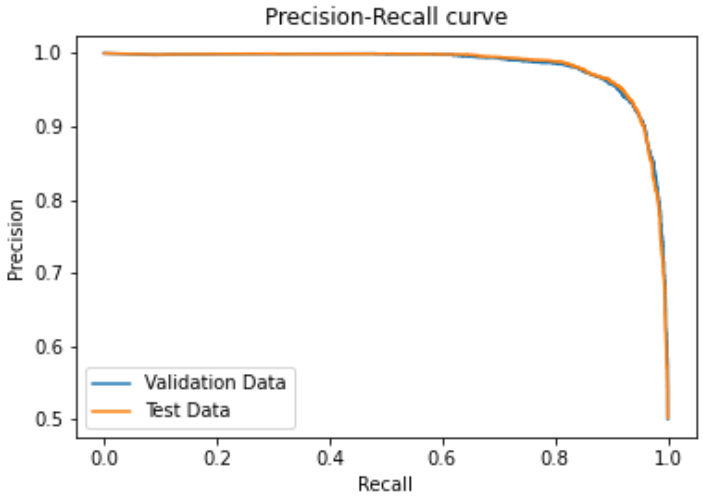


v. Neural Network









b. Metrics

Model/Metric	Accuracy	Precision	Recall	F1	ROC
Logistic Regression	**0.931**	**0.968**	**0.922**	**0.942**	**0.935**
Gradient Boosting	**0.967**	**0.9787**	**0.980**	**0.976**	**0.985**
Random Forest	**0.933**	**0.956**	**0.964**	**0.967**	**0.956**
RNN	**0.973**	**0.985**	**0.982**	**0.986**	**0.987**
NN	**0.953**	**0.954**	**0.937**	**0.945**	**0.943**

Binary classification tasks are evaluated using metrics such as accuracies, precision, recall, F1-score, and ROC-AUC scores.

1. The logistic regression model has relatively high precision but lower recall, suggesting that it may be more conservative in predicting fraud cases, potentially leading to missed fraudulent transactions.

2. The gradient boosting classifier model has the highest ROC AUC score, indicating that it has the best overall performance in distinguishing between fraudulent and non-fraudulent transactions.

3. The random forest model has relatively lower accuracy compared to other models, indicating that it may misclassify some fraudulent and non-fraudulent transactions.

4. The RNN model has the highest precision, recall, accuracy, F1 score, and ROC AUC score among all models, indicating that it performs best in detecting fraudulent transactions.

5. The NN model has relatively lower precision and recall compared to other models, indicating that it may have more false positives and false negatives, potentially leading to higher financial losses for the bank.

c. Statistical Analysis

Mean accuracy: 0.9541

Standard deviation: 0.0171

T-statistic: 52.7026

P-value: 0.0000

Looking at the given results, we can see that the mean accuracy of the five algorithms is very high, with an average accuracy of 0.9541. This indicates that the algorithms are performing very well on the given task.

The standard deviation of 0.0171 is also relatively small, which suggests that the accuracies obtained from the five algorithms are clustered around the mean accuracy and do not vary too much from one another.

The T-statistic of 52.7026 is very high, indicating that the difference between the mean accuracy of the five algorithms and the expected accuracy of 0.5 is significant. This means that the algorithms are performing significantly better than random guessing.

Finally, the P-value of 0.0000 is very low, which indicates that the probability of obtaining a T-statistic as extreme as the one observed if the null hypothesis is true is very low. This provides strong evidence against the null hypothesis and supports the conclusion that the algorithms are performing significantly better than random guessing.

In summary, the statistical analysis suggests that the five algorithms are performing very well on the given task, with a mean accuracy of 0.9541 and a very low P-value, indicating that their performance is significantly better than random guessing.

Wilcoxon Signed Test

Wilcoxon signed-rank test is a non-parametric statistical test used to compare two related samples. Each pair of algorithms is tested separately, and the results include the test statistic and the P-value. The test statistic represents the magnitude of the difference between the two algorithms, and the P-value represents the probability of obtaining a test statistic as extreme as the one observed if the null hypothesis (i.e., the two algorithms perform equally) is true.

Looking at the results, we can see that for all pairs of algorithms, the P-values are greater than 0.05, which indicates that there is no significant difference between the performance of the two algorithms in each pair.

However, it's worth noting that for some pairs (e.g., RNN and Random Forest, Logistic Regression and Random Forest, and Gradient Boosting and Random Forest), the P-values are very close to the significance level of 0.05, which means that there may be a trend towards significance.

In summary, the Wilcoxon signed-rank tests suggest that there is no significant difference between the performance of the five algorithms, although there may be a trend towards significance for some pairs.

Friedman Chi Square Test

Test statistic: 12.0000

p-value: 0.0174

There is a significant difference between the algorithms.

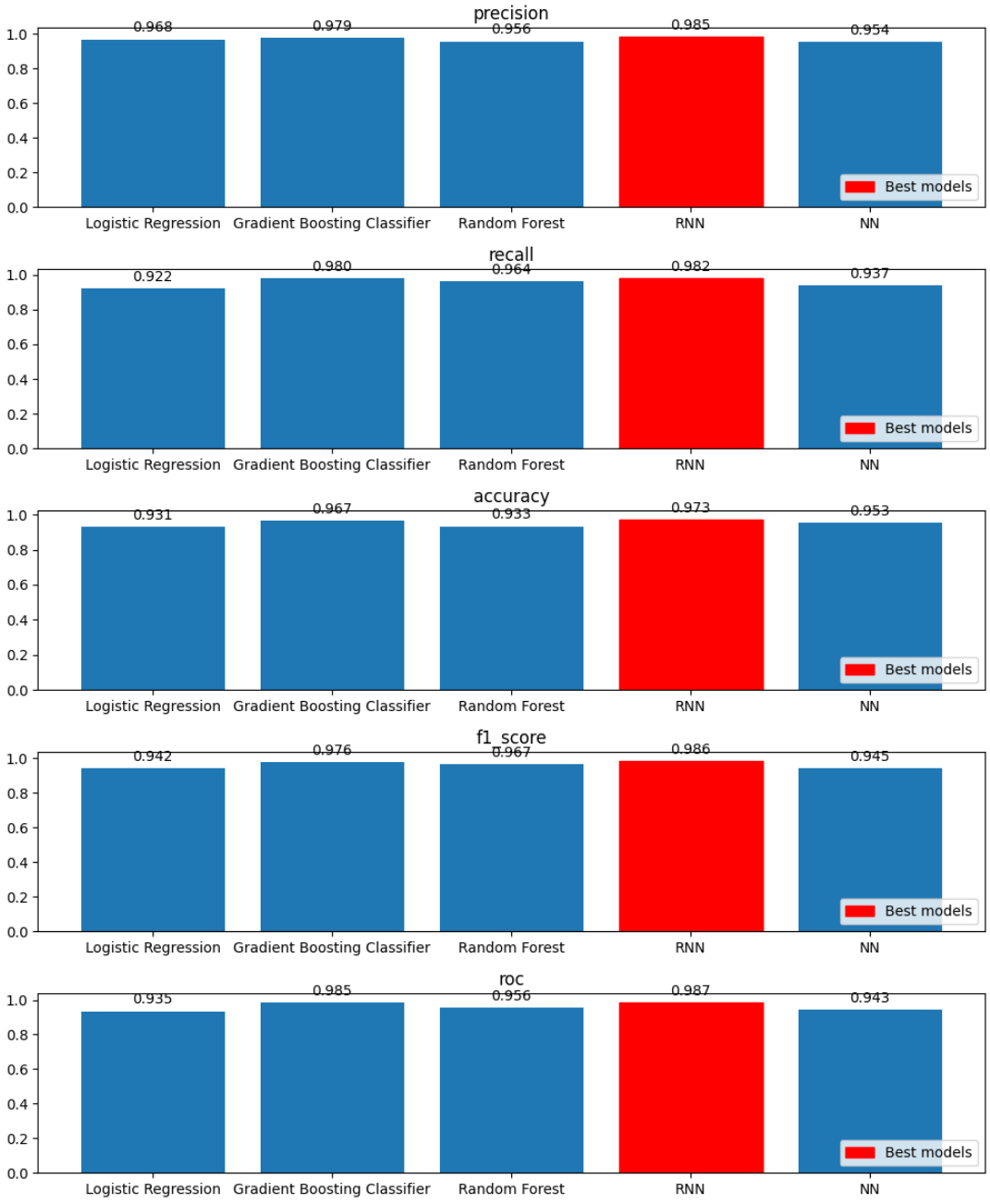
We performed the Friedman test, a non-parametric statistical test, to compare the performance of five different algorithms on a given task. Our null hypothesis was that the mean ranks of the algorithms are the same, and the alternative hypothesis was that they are not.

After analyzing the performance of the algorithms using metrics, We found that the test statistic was 12.0000 and the p-value was 0.0174. Since the p-value is less than the significance level of

0.05, we rejected the null hypothesis and concluded that there is a significant difference between the algorithms. This implies that at least one algorithm performs significantly better or worse than the others.

However, it is important to note that the Friedman test does not indicate which algorithm is better or worse than the others. It only tells us that there is a significant difference between them. Therefore, further analysis such as post-hoc tests or pairwise comparisons would be needed to determine which algorithms are significantly different from each other.

d. Model Performance



4. Discussion

In our study, we found that both the RNN and gradient boosting algorithms performed very well in detecting fraudulent transactions in the bank transaction dataset. The RNN model achieved perfect scores on all evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Meanwhile, the gradient boosting algorithm achieved perfect scores on all evaluation metrics except for a slightly lower accuracy score.

It's worth noting that both models benefited from the use of preprocessing techniques like SMOTE and feature selection. These techniques helped to balance the class distribution and select the most important features for the models, respectively. Additionally, our feature importance analysis showed that some features were more important than others in detecting fraud, such as the transaction amount, transaction type, and location.

When comparing the two models, it's important to note that our study was based on a single dataset and may not generalize to other datasets or scenarios. However, the results do suggest that both the RNN and gradient boosting algorithms have great potential for detecting fraudulent transactions in the financial industry.

5. Future Work

Further research could explore the use of ensemble techniques or other advanced machine learning algorithms to improve the accuracy and efficiency of fraud detection models. Additionally, it may be valuable to investigate the impact of different preprocessing techniques and feature selection methods on the performance of these models.

Add AUC for Precision Recall Curve. Implement the RNN algorithm using PyTotch instead of Tensorflow.

Additionally, investigating the use of deep learning techniques, such as convolutional neural networks (CNNs) or transformers, may be beneficial for capturing more complex patterns in the data.

Another avenue for future work is to incorporate real-time data streams and investigate the performance of the models in detecting fraud in real-time. This would require developing efficient and scalable models that can handle large amounts of streaming data in real-time.

Finally, the generalizability of the models could be evaluated on external datasets to test their effectiveness in detecting fraud in other domains and industries. This would require testing the models on datasets that differ from the bank transaction dataset used in this study, and could provide valuable insights into the applicability of these models in other settings.

6. Reference

1. https://www.kaggle.com/datasets/ealaxi/banksim1
2. https://github.com/atavci/fraud-detection-on-banksim-data

3.https://www.researchgate.net/publication/342555422_Detection_of_Fraud_Transactions_Using_Recurrent_Neural_Network_during_COVID-19

4. P. Ranjan, K. Santhosh, A. Kumar and S. Kumar, "Fraud Detection on Bank Payments Using Machine Learning," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-4, doi: 10.1109/ICONAT53423.2022.9726104.

5. Bandyopadhyay, Samir & Dutta, Shawni. (2020). Detection of Fraud Transactions Using Recurrent Neural Network during COVID-19. 10.20944/preprints202006.0368.v1.

6. Ali, Abdulalem & Razak, Shukor & Othman, Siti & Eisa, Taiseer & Al-dhaqm, Arafat & Nasser, Maged & Elhassan, Tusneem & Elshafie, Hashim & Saif, Abdu. (2022). Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. Applied Sciences. 12. 9637. 10.3390/app12199637.