DIC Project Report

AWS Fraud Detection

Abstract:

The issue of fraudulent activities has grown significantly in importance for consumers and businesses as the e-commerce sector continues to expand quickly. Traditional rule-based fraud detection techniques have proved inefficient at detecting the constantly evolving fraudsters' tactics. Machine learning algorithms have emerged as a viable method for accurately detecting and preventing fraud in e-commerce transactions in recent years. This project report presents an extensive study on the application of machine learning algorithms for fraud detection in Amazon's transaction data. The major objective is to create a robust fraud detection system that can consistently identify fraudulent transactions while minimizing false negatives, guaranteeing the security and dependability of online transactions. As a part of Phase 3, we have built a data product from our Phase 2 models.

The dataset used in this project is from Amazon Web Services (AWS) Fraud Detector Samples. It contains transactions for the years 2022 and 2023, sourced from Github Repository.

Problem Statement:

Choosing the best algorithm that could accurately detect potential fraudulent transactions

- We selected this problem statement due to the significant number of undetected fraudulent transactions and the substantial losses they cause.
- Accurately detecting fraudulent transactions can tackle fraudulent activities most effectively. It can minimize the financial losses associated with fraudulent activities significantly. Manual reviewing of the transactions is expensive, time-consuming, and labor-intensive. Fraud incidents can sabotage the trust, causing customers to move to competitors.

Algorithm:

Six classifier algorithms that were used on our data to create visualizations for the results are as follows:

1. Logistic Regression,
2. Naive Bayes Classifier,
3. Support Vector Machine (SVM) Classifier,
4. Random Forest Classifier,
5. XGBoost Classifier,
6. and AdaBoost Classifier

These algorithms are used to help detect fraudulent transactions with the given data. To evaluate the results of each model we implemented the AUC-ROC curve and accuracy. Also, the dataset is divided into train and test where the train is used for training the model and the test is used for evaluating the model that is trained.

Steps to run the Notebook and Web Application:

In this, we used Python 3.6 version from our Jupyter Notebook to Streamlit web application. The following are the steps to execute the code.

1. Install **Python 3.6 [IMPORTANT]**
2. Install the libraries from the **requirements.txt** file using pip along with Jupyter Notebook

To run the web application, run the following command from the project folder. "**streamlit run webapp.py**"

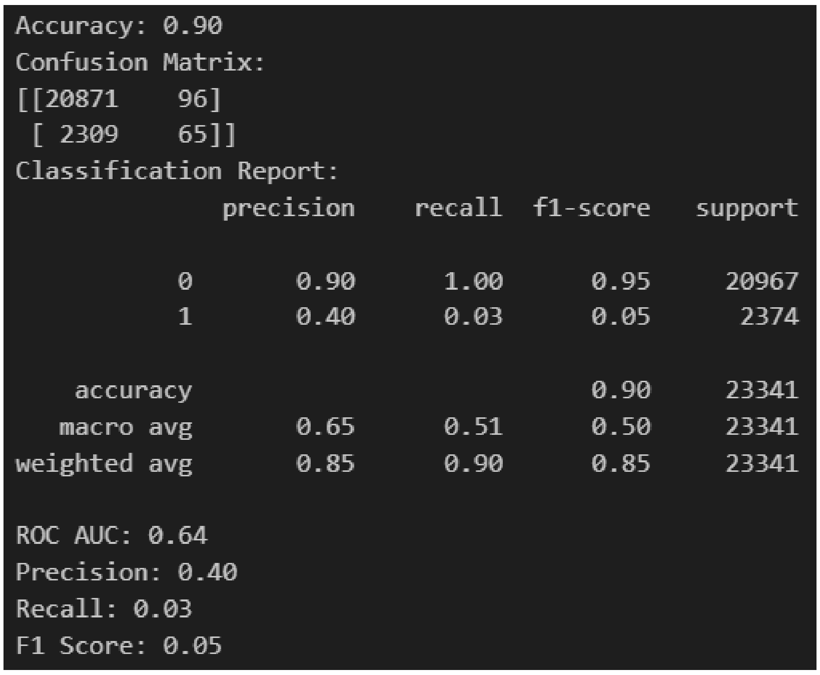
Model Tuning and Evaluation:

<u>Logistic Regression</u>

<u>1. Model Building & Tuning</u>
As we are performing fraud detection a classification on the dataset a simple logistic regression is performed. As it is evident that the accuracy shown by this model is 90 which means this model performs really well on this dataset (see Fig 1).

To attain this accuracy, we fine-tuned our model by changing the attributes solver and max_iterations. In this process we were able to choose the best values for the following attributes, they are 1000 for max_iterations and lbfgs for the solver. If anything over 1000 iterations the model is overfitting. Accuracy cannot be used as the only metric to evaluate our model. As we can see the values of precision, Recall, and F1 score, we can say that our model is still underperforming (see Fig.1). This Means still our model is still not up to the mark as we expected.

<u>2. Model Performance Insights</u>
The value of AUC we get from our model is 0.64 which is slightly higher than the threshold value (i.e., 0.5). This is evident from the graph shown below (see Fig.2) as you can see the model is performing moderately. With all these evaluation results we feel that maybe this is not the best model for our dataset.



```
Accuracy: 0.90
Confusion Matrix:
[[20871    96]
 [ 2309    65]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95     20967
           1       0.40      0.03      0.05      2374

    accuracy                           0.90     23341
   macro avg       0.65      0.51      0.50     23341
weighted avg       0.85      0.90      0.85     23341

ROC AUC: 0.64
Precision: 0.40
Recall: 0.03
F1 Score: 0.05
```

Fig 1. Evaluation Metrics

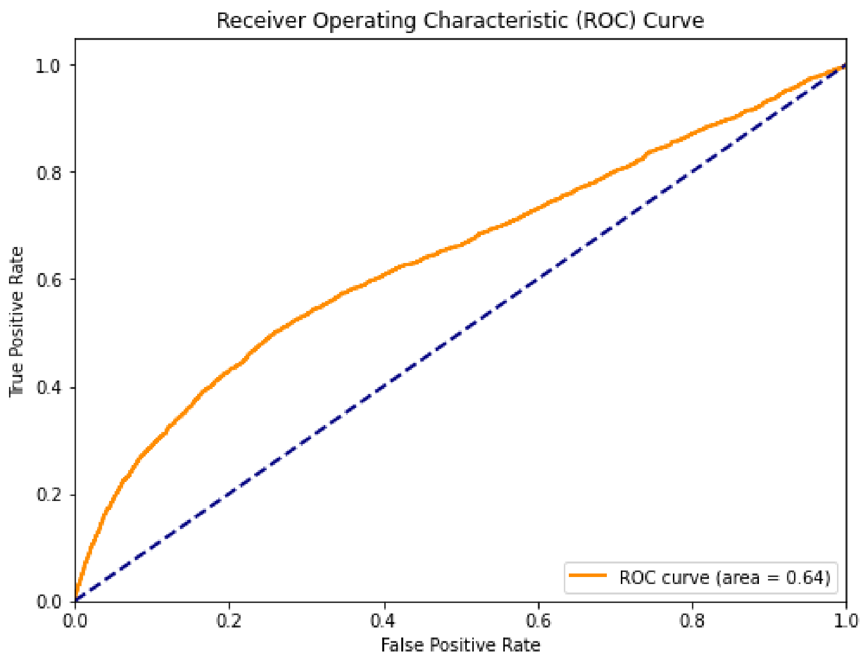


Fig 2. ROC Curve Plot

Naive Bayes Classifier

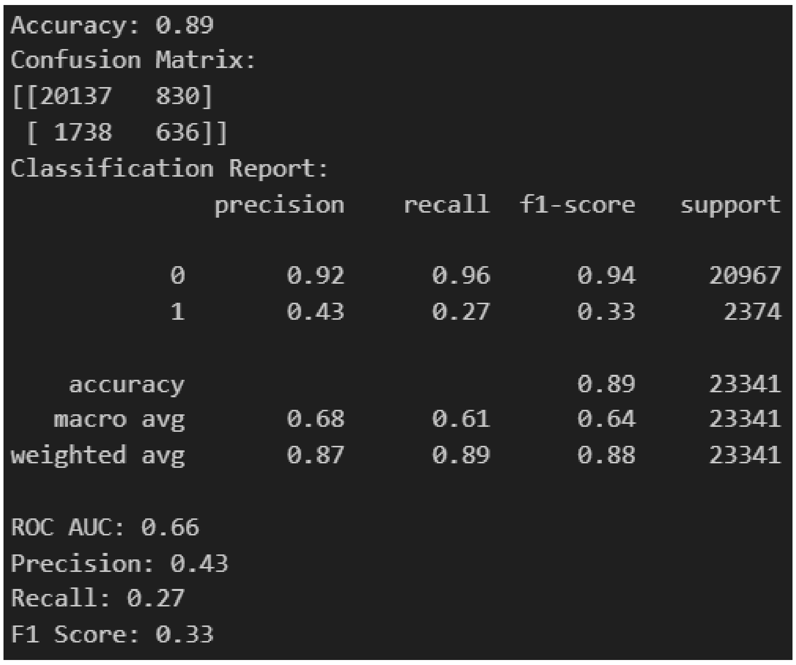
1. Model Building & Tuning

The accuracy for this algorithm is 89% which performs almost the same as the logistic regression. The model shows really good accuracy which was attained because of the fine tuning. We adjusted the smoothing parameter to improve the performance. Statistical tests were done to select only the relevant features. Recall and Precision were very low, but we were able to improve them by tuning the threshold. Naive Bayes is less prone to overfitting because of its simplicity, but overfitting was still monitored.

Even after all the fine-tuning, precision and recall for class 1 is still low which indicates that there are many false negatives still. So, the model is still not performing well.

2. Model Performance Insights

AUC of 0.66 shows moderate discriminatory ability which is slightly above the random chance values of 0.5. The ROC curve (Fig. 4) shows that the model doesn't achieve optimal separation between the classes. The accuracy is 89, but the low precision (0.43) and recall (0.27) suggest the model is not effectively identifying positive instances, which is also shown in the low F1 Score (0.33). Despite the high accuracy of the model, other metrics imply the model may not be the best fit for the dataset.



```
Accuracy: 0.89
Confusion Matrix:
[[20137   830]
 [ 1738   636]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94     20967
           1       0.43      0.27      0.33      2374

    accuracy                           0.89     23341
   macro avg       0.68      0.61      0.64     23341
weighted avg       0.87      0.89      0.88     23341

ROC AUC: 0.66
Precision: 0.43
Recall: 0.27
F1 Score: 0.33
```

Fig 3. Evaluation Metrics

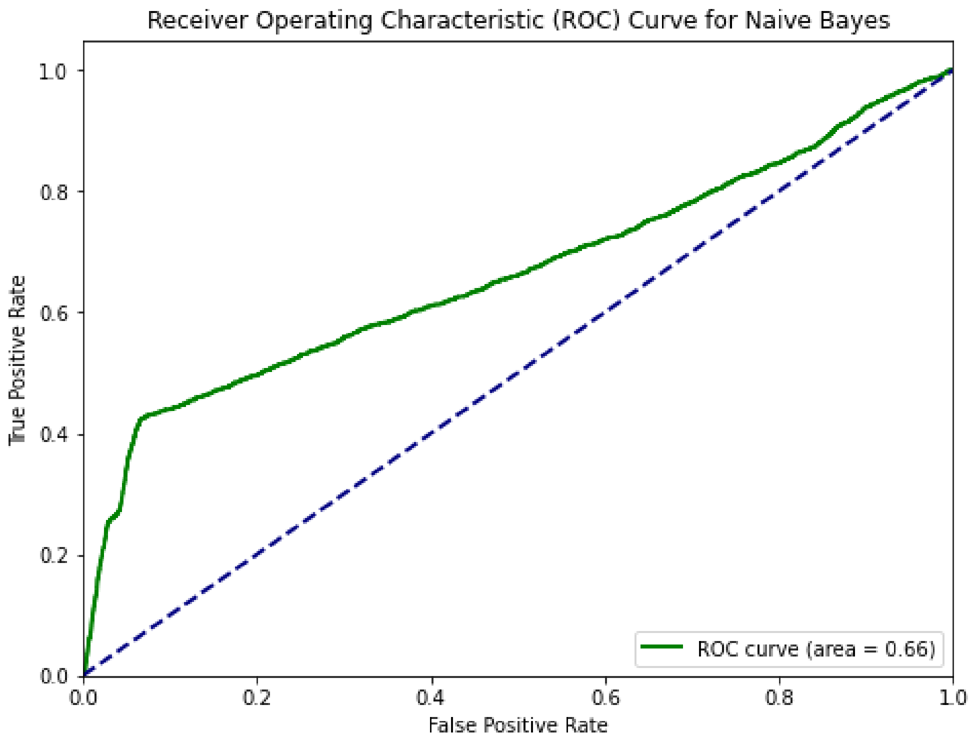


Fig 4. ROC Curve Plot

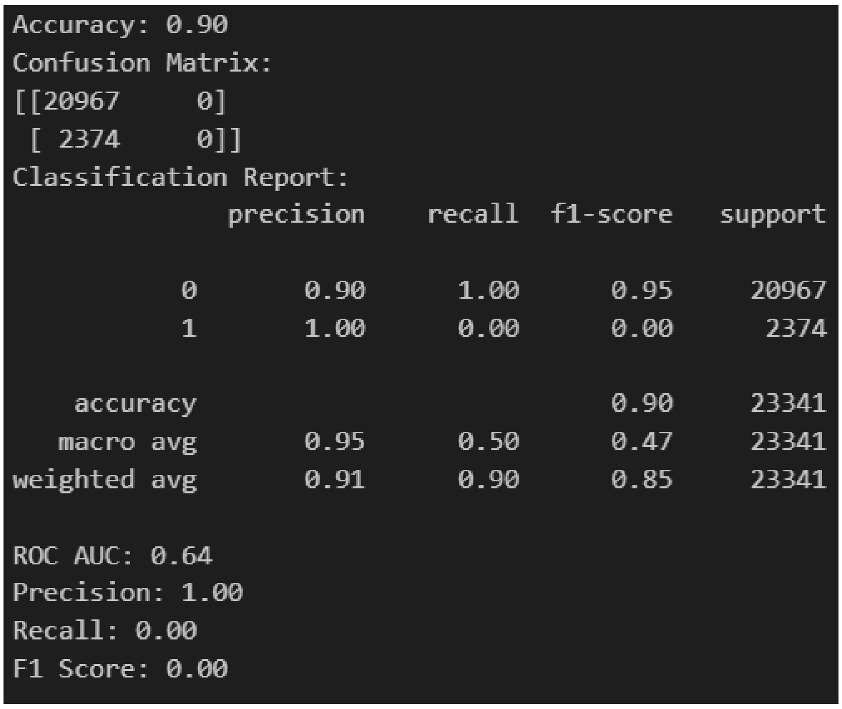
Support Vector Machine Classifier:

1. Model Building & Tuning

The accuracy shown for this model is 90 which is the same as logistic regression. This shows that the model performs well on the dataset. Some of the fine we did were adjusting the C parameter which helped to balance between bias and variance. Ensured that all of the features were on a similar scale. Identified the most suitable set of parameters and made sure the model performed well on data that had not yet been observed by using optimum cross-validation. Fine-tuning the decision threshold helped adjust the trade-off between precision and recall.

2. Model Performance Insights

The SVM model shows high accuracy (0.90), but there are noticeable imbalances shown by the evaluation metric. With a precision and recall of 0.90 and 1.00 for the negative class, the model accurately predicts the negative class. However, a recall of 0 for the positive class shows that no positive cases were found. The model fails to properly classify the positive cases is shown with F1 score of 0.00. The AUC of 0.64 suggests the model is slightly better than random at distinguishing between classes, but class-specific metrics show a serious problem with the model's sensitivity to the positive class.



```
Accuracy: 0.90
Confusion Matrix:
[[20967     0]
 [ 2374     0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95     20967
           1       1.00      0.00      0.00      2374

    accuracy                           0.90     23341
   macro avg       0.95      0.50      0.47     23341
weighted avg       0.91      0.90      0.85     23341

ROC AUC: 0.64
Precision: 1.00
Recall: 0.00
F1 Score: 0.00
```

Fig 5. Evaluation Metrics

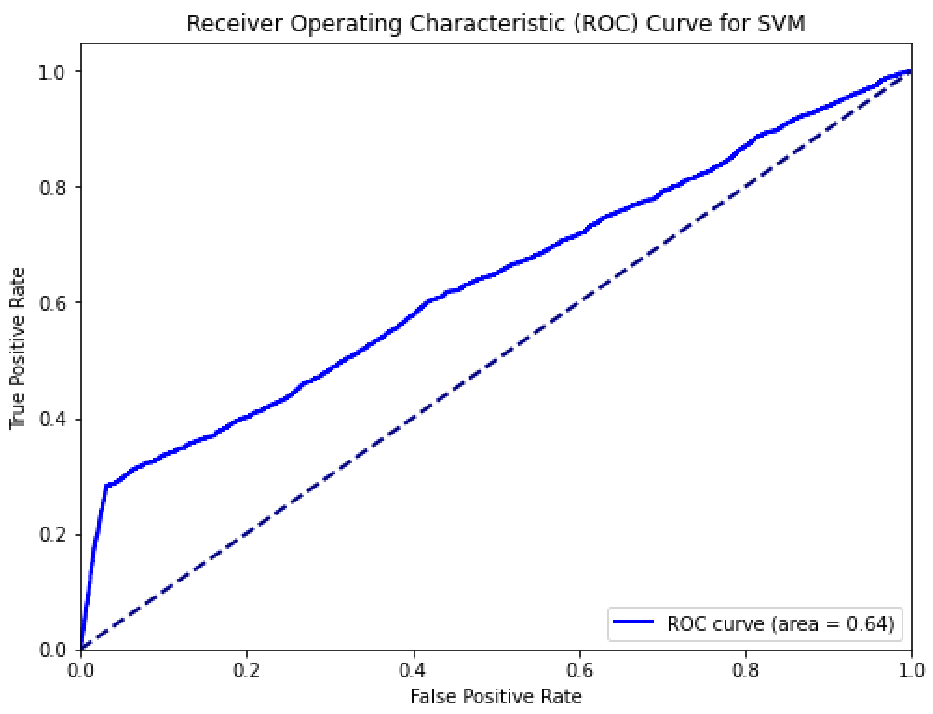


Fig 6. ROC Curve Plot

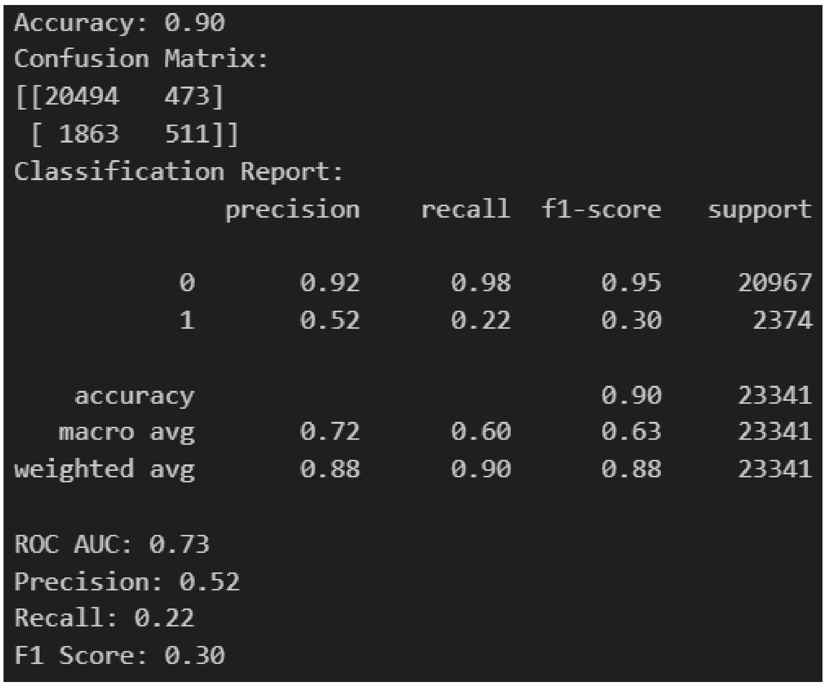
Random Forest Classifier:

1. Model Building & Tuning

The accuracy shown for this model is 90 which is the same as logistic regression and SVM. This shows that the model performs well on the dataset. Some of the tuning we did was modifying the depth of the trees to prevent overfitting. Increased the number of trees in the forest which helped improve the model's stability and accuracy. Used cross-validation to determine the effectiveness of the model on unseen data.

2. Model Performance Insights

The accuracy of 90 implies that the overall predictions are correct. However, the model shows a noticeable variation in the model's performance between classes. The precision for the negative class is high (0.92) and the recall is even higher (0.98) which is because of the strong F1 score (0.95). On the other hand, the positive class has a moderate precision (0.52), but the recall is very low (0.22) resulting in a poor F1 score (0.30). This shows the model does a good job are classifying the majority, but struggles with detecting the minority class. The ROC AUC (0.73) indicated that it can fairly distinguish between the classes. The model still needs refining to increase its sensitivity towards the positive class. Low recall and F1 score for positive class show that the model's ability to detect real positives is constrained.



```
Accuracy: 0.90
Confusion Matrix:
[[20494   473]
 [ 1863   511]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     20967
           1       0.52      0.22      0.30      2374

    accuracy                           0.90     23341
   macro avg       0.72      0.60      0.63     23341
weighted avg       0.88      0.90      0.88     23341

ROC AUC: 0.73
Precision: 0.52
Recall: 0.22
F1 Score: 0.30
```

Fig 7. Evaluation Metrics

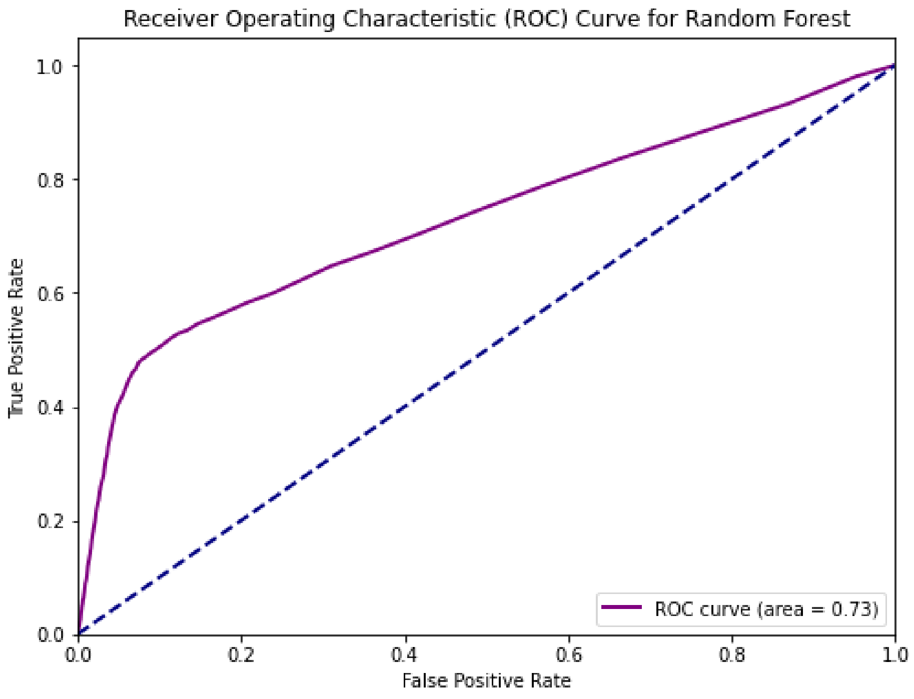


Fig 8. ROC Curve Plot

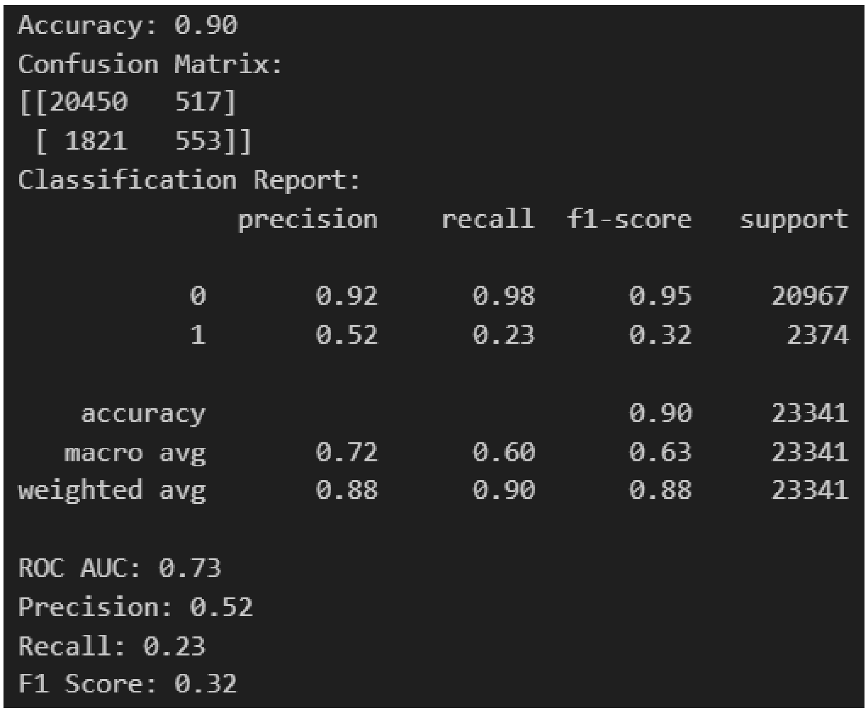
<u>XGBoost Classifier:</u>

<u>1. Model Building & Tuning</u>

The accuracy shown for this model is 90 which is the same as logistic regression, SVM, and Random Forest. This shows that the model performs well on the dataset. Some of the tuning we did was adjusting the eta value which governs each tree's input to the ensemble. Reducing it made the learning process simpler resulting in a robust model. Adjusted the tress ensuring the balance between performance and fitting. Stopping the training once the validation metric stops improving preventing overfitting. Adjusted parameters such as max depth of trees and subsample to prevent overfitting.

2. Model Performance Insights

The accuracy for this model is 90. It has a strong recall (0.98) and robust precision (0.92). The F1 score for the negative class is high (0.95). The precision (0.52) and recall (0.23) for the positive class are seen to be quite low resulting in a lower F1 score (0.32). The model performs well in predicting the negative class but underperforms significantly in identifying instances of the positive class. The ROC AUC (0.73) shows that it has a decent ability to distinguish between the classes, but the positive class performance needs improvement.



```
Accuracy: 0.90
Confusion Matrix:
[[20450   517]
 [ 1821   553]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     20967
           1       0.52      0.23      0.32      2374

    accuracy                           0.90     23341
   macro avg       0.72      0.60      0.63     23341
weighted avg       0.88      0.90      0.88     23341

ROC AUC: 0.73
Precision: 0.52
Recall: 0.23
F1 Score: 0.32
```

Fig 9. Evaluation Metrics

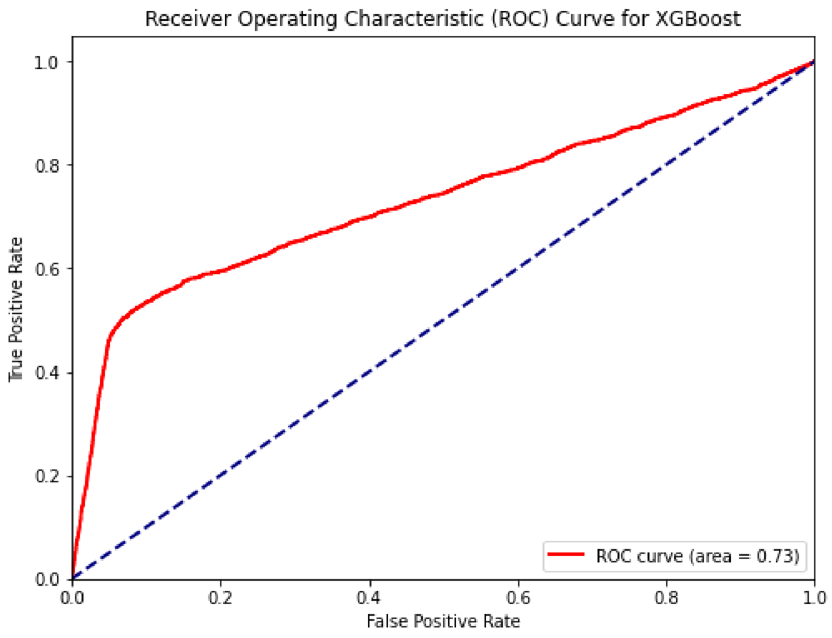


Fig 10. ROC Curve Plot

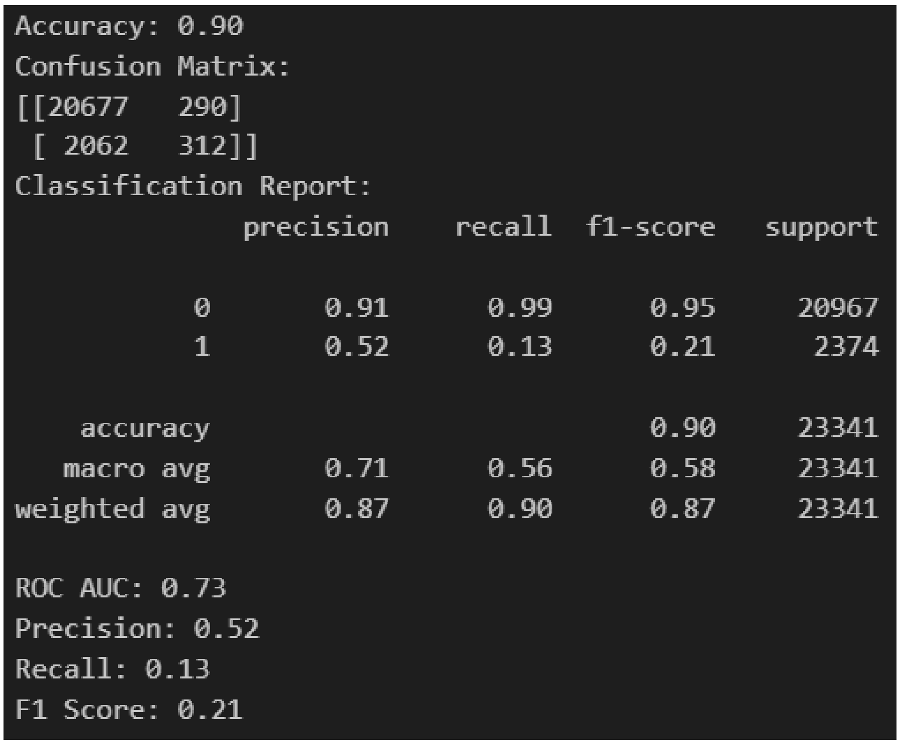
<u>AdaBoost Classifier:</u>

<u>1. Model Building & Tuning</u>

The accuracy shown for this model is 90 which is the same as logistic regression, SVM, Random Forest, and XG Boost. This shows that the model performs well on the dataset. Some of the tuning we did was adjusting the maximum number of weak learners in the boosting process and the learning rate for better generalization and improved overall performance of the model. Used cross-validation to find the best hyperparameters and to make sure the model is generalizing well.

<u>2. Model Performance Insights</u>

The accuracy attained for this model is 90 with a precision of 0.91 and a recall of 0.99, resulting in an F1 score of 0.95. The model performs extremely well in the negative class. With a precision of 0.52 and a noticeably low recall of 0.13, it underperforms on the positive class resulting in an F1 score of 0.21. This considerable difference in the model's performance indicates that it has trouble accurately recognizing the minority class correctly. The poor recall and F1 score for the positive class show that the model is not adequately capturing the true positive cases, although the ROC AUC of 0.73 shows a decent potential of the model to differentiate between the classes. The model's sensitivity to the positive class still needs improvement.



```
Accuracy: 0.90
Confusion Matrix:
[[20677   290]
 [ 2062   312]]
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.99      0.95     20967
           1       0.52      0.13      0.21      2374

    accuracy                           0.90     23341
   macro avg       0.71      0.56      0.58     23341
weighted avg       0.87      0.90      0.87     23341

ROC AUC: 0.73
Precision: 0.52
Recall: 0.13
F1 Score: 0.21
```

Fig 11. Evaluation Metrics

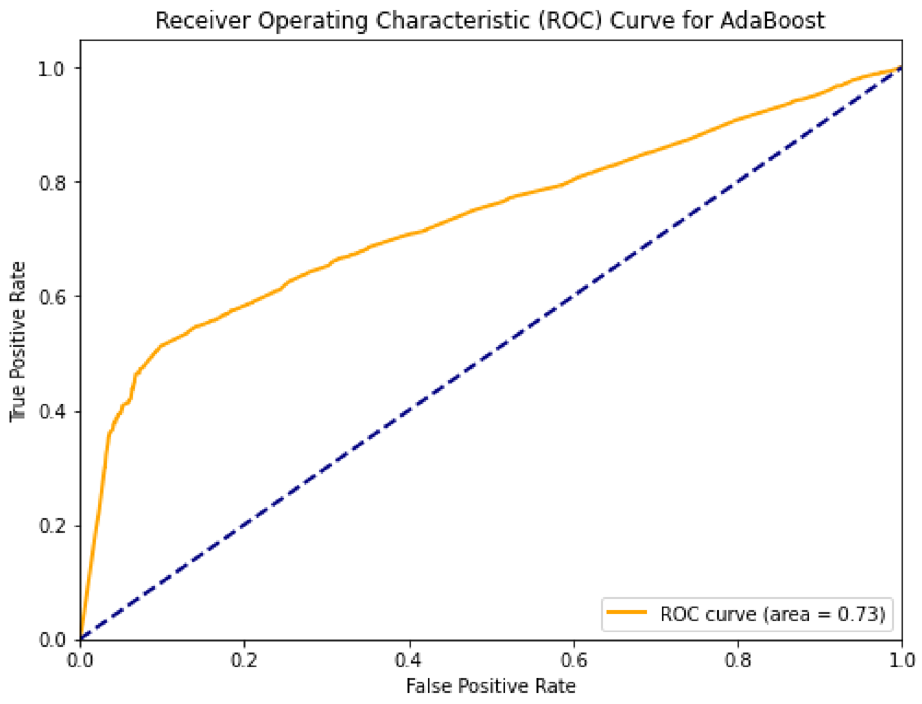


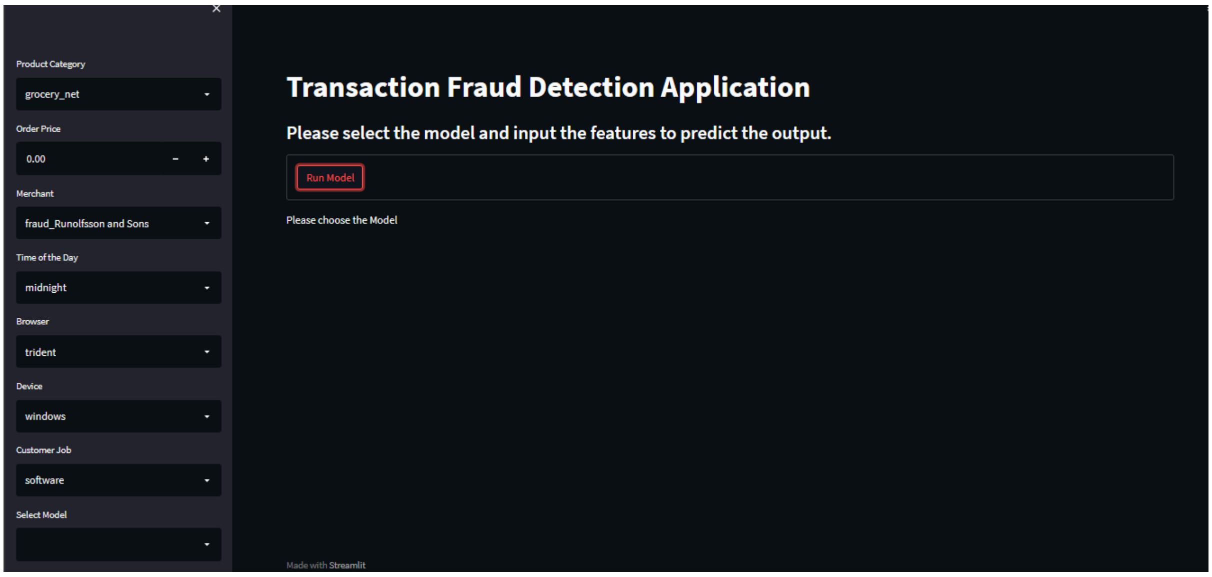
Fig 12. ROC Curve Plot

Web-Application Overview:

We were able to build this user-friendly web application with the help of Streamlit which is an open-source Python framework. This was possible with the help of Phase 2 models, allowing users to interact with all the models to gain insight into the best algorithm that could accurately detect potentially fraudulent transactions.

Feature Selection: Users can select from various features to specify transaction details, including Product Category, Order Price, Merchant, Time of the Day, Browser, Device, and Customer Job.
Model Selection: Users can select from available six algorithms for fraud detection.

Once the user selects the model and inputs the features, the application will evaluate and display the output showing if the selected transaction is termed a Fraud Transaction or not. This web application is designed to provide a quick and efficient assessment, aiding users in identifying and mitigating potential fraud in transactions. Figure 13 shows the User Interface of the web application.


Product Category
grocery_net
Order Price
0.00
Merchant
fraud_Runolfsson and Sons
Time of the Day
midnight
Browser
trident
Device
windows
Customer Job
software
Select Model

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

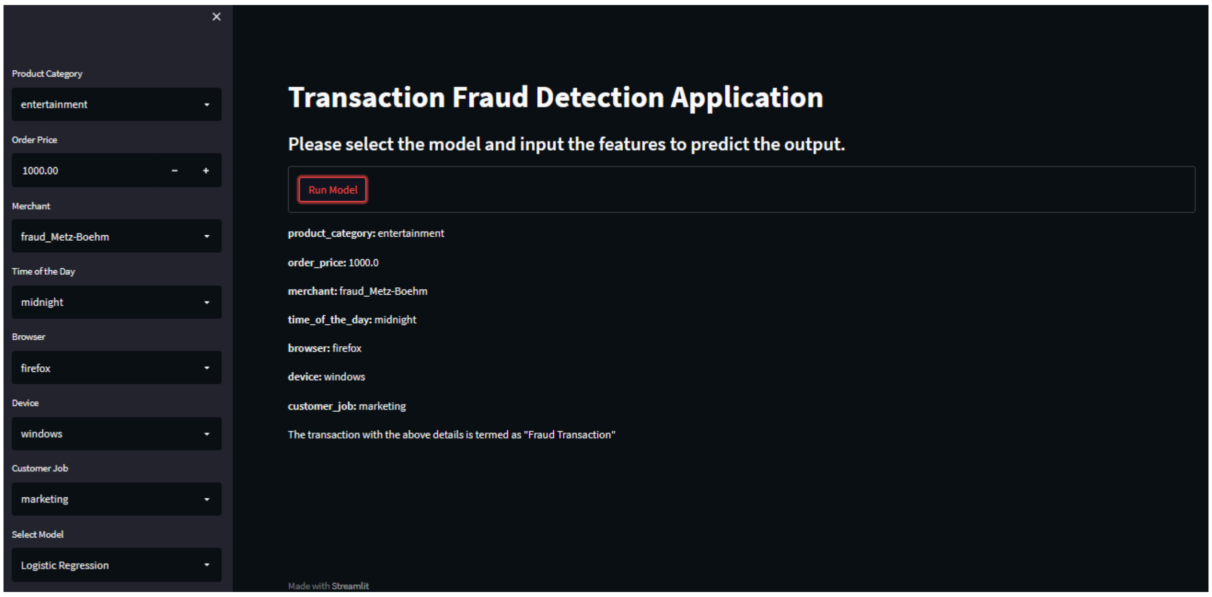
Please choose the Model

Fig 13. Transaction Fraud Detection Application.

Fraud Detection Application: Model and Feature Specification:

<u>Logistic Regression</u>

<u>*1. Fraud Transaction*</u>


Product Category
entertainment
Order Price
1000.00
Merchant
fraud_Metz-Boehm
Time of the Day
midnight
Browser
firefox
Device
windows
Customer Job
marketing
Select Model
Logistic Regression

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: entertainment

order_price: 1000.0

merchant: fraud_Metz-Boehm

time_of_the_day: midnight

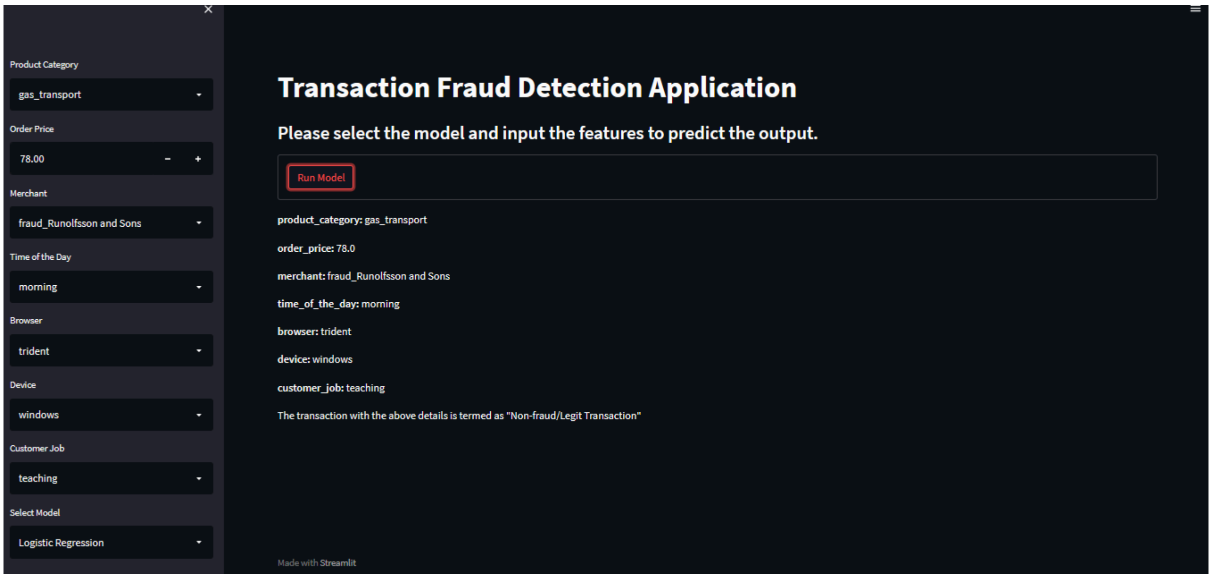
browser: firefox

device: windows

customer_job: marketing

The transaction with the above details is termed as "Fraud Transaction"

<u>*2. Non Fraud/ Legit Transaction*</u>



Product Category: gas_transport
Order Price: 78.00
Merchant: fraud_Runolfsson and Sons
Time of the Day: morning
Browser: trident
Device: windows
Customer Job: teaching
Select Model: Logistic Regression

Transaction Fraud Detection Application

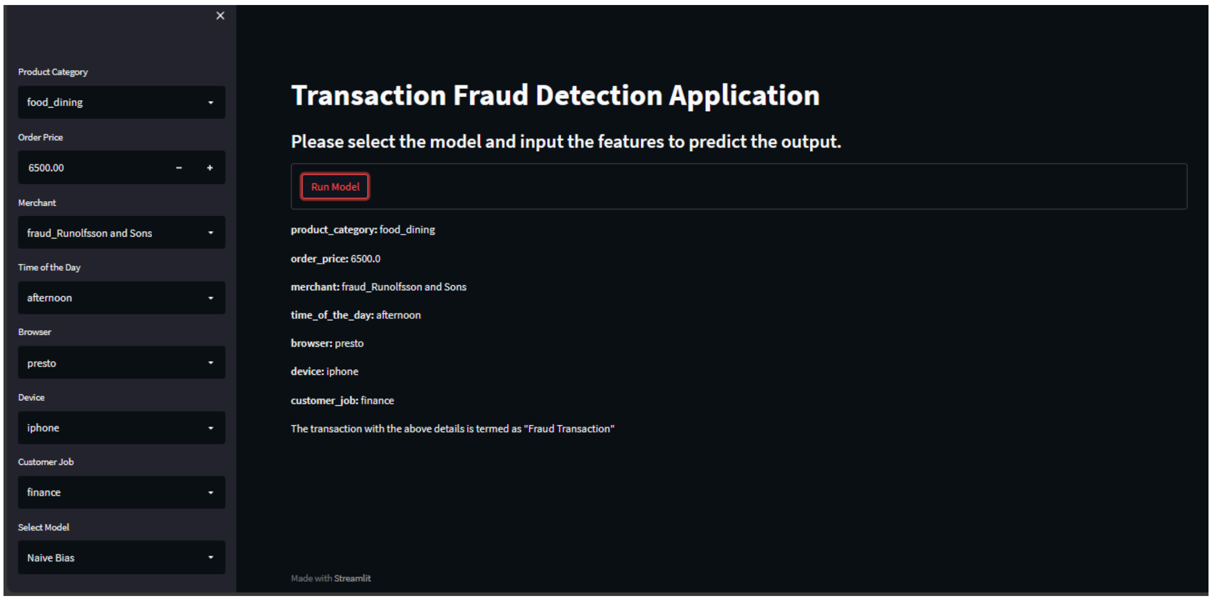
Please select the model and input the features to predict the output.

product_category: gas_transport
order_price: 78.0
merchant: fraud_Runolfsson and Sons
time_of_the_day: morning
browser: trident
device: windows
customer_job: teaching

The transaction with the above details is termed as "Non-fraud/Legit Transaction"

<u>Naive Bayes</u>

<u>*1. Fraud Transaction*</u>



Product Category: food_dining
Order Price: 6500.00
Merchant: fraud_Runolfsson and Sons
Time of the Day: afternoon
Browser: presto
Device: iphone
Customer Job: finance
Select Model: Naive Bias

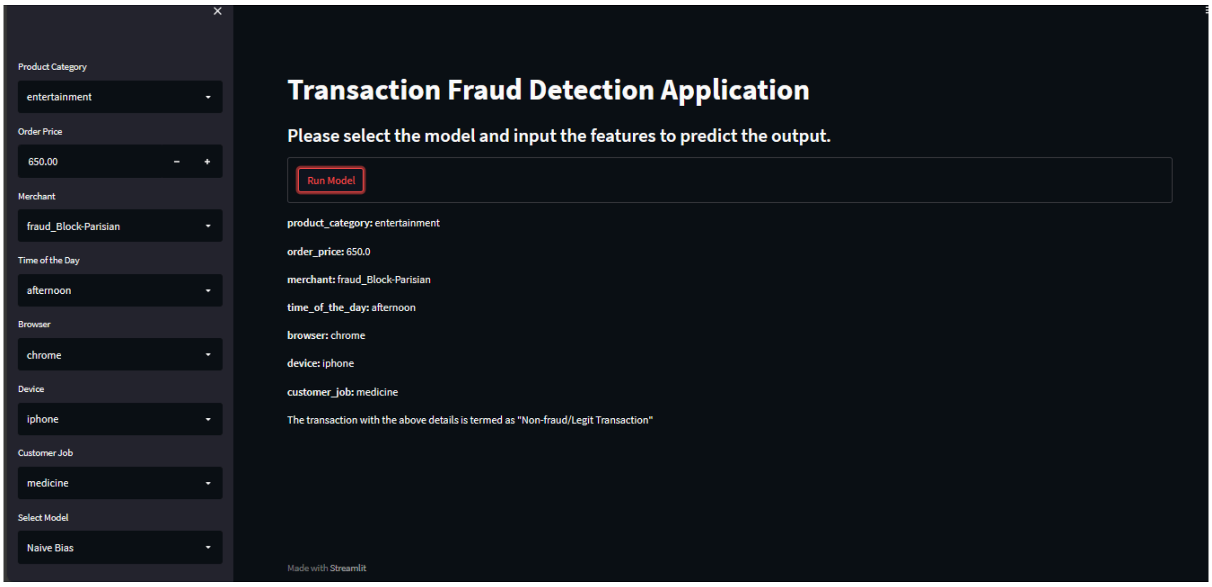
Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: food_dining
order_price: 6500.0
merchant: fraud_Runolfsson and Sons
time_of_the_day: afternoon
browser: presto
device: iphone
customer_job: finance

The transaction with the above details is termed as "Fraud Transaction"

<u>*2. Non Fraud/ Legit Transaction*</u>


Product Category
entertainment
Order Price
650.00
Merchant
fraud_Block-Parisian
Time of the Day
afternoon
Browser
chrome
Device
iphone
Customer Job
medicine
Select Model
Naive Bias

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

Run Model

product_category: entertainment

order_price: 650.0

merchant: fraud_Block-Parisian

time_of_the_day: afternoon

browser: chrome

device: iphone

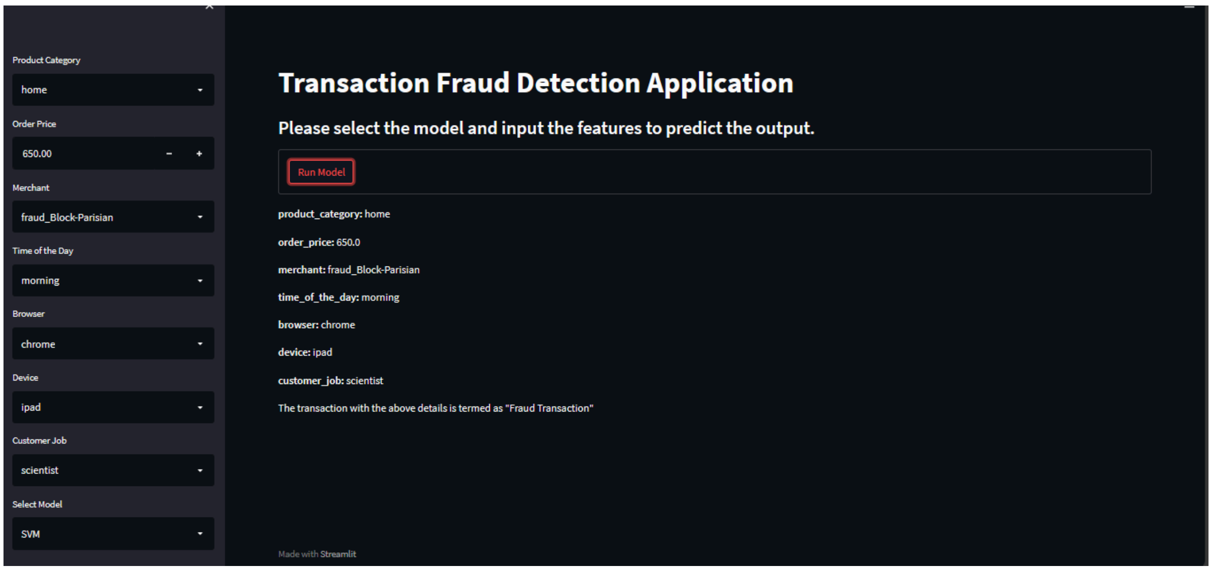
customer_job: medicine

The transaction with the above details is termed as "Non-fraud/Legit Transaction"

Made with Streamlit

<u>SVM</u>

<u>1. Fraud Transaction</u>


Product Category
home
Order Price
650.00
Merchant
fraud_Block-Parisian
Time of the Day
morning
Browser
chrome
Device
ipad
Customer Job
scientist
Select Model
SVM

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

Run Model

product_category: home

order_price: 650.0

merchant: fraud_Block-Parisian

time_of_the_day: morning

browser: chrome

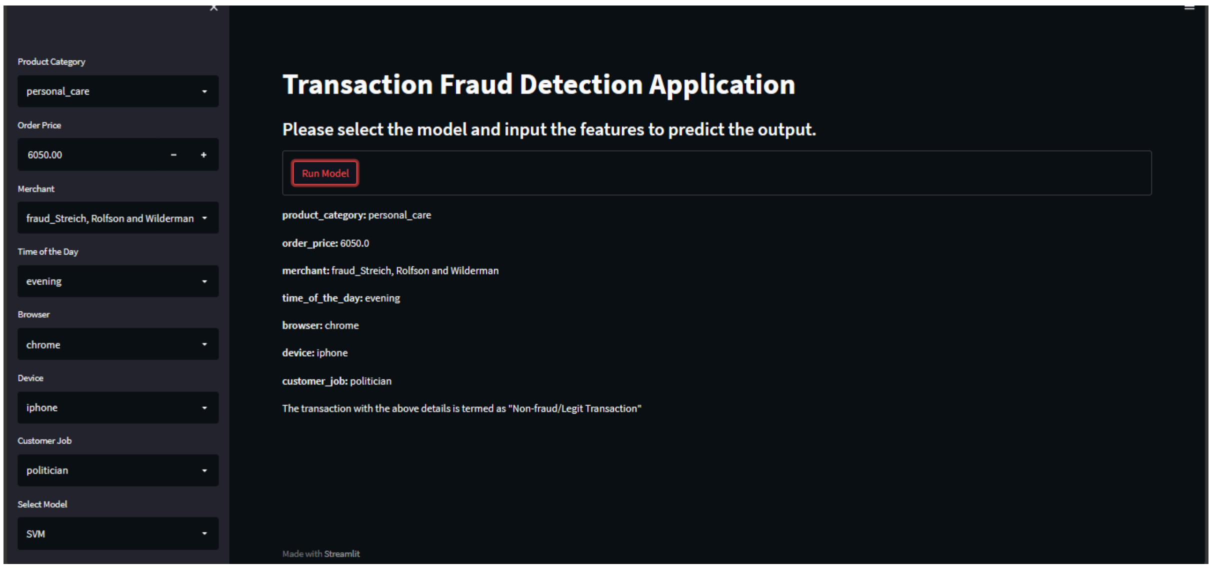
device: ipad

customer_job: scientist

The transaction with the above details is termed as "Fraud Transaction"

Made with Streamlit

<u>2. Non Fraud/ Legit Transaction</u>



Product Category

personal_care

Order Price

6050.00

Merchant

fraud_Streich, Rolfson and Wilderman

Time of the Day

evening

Browser

chrome

Device

iphone

Customer Job

politician

Select Model

SVM

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

Run Model

product_category: personal_care

order_price: 6050.0

merchant: fraud_Streich, Rolfson and Wilderman

time_of_the_day: evening

browser: chrome

device: iphone

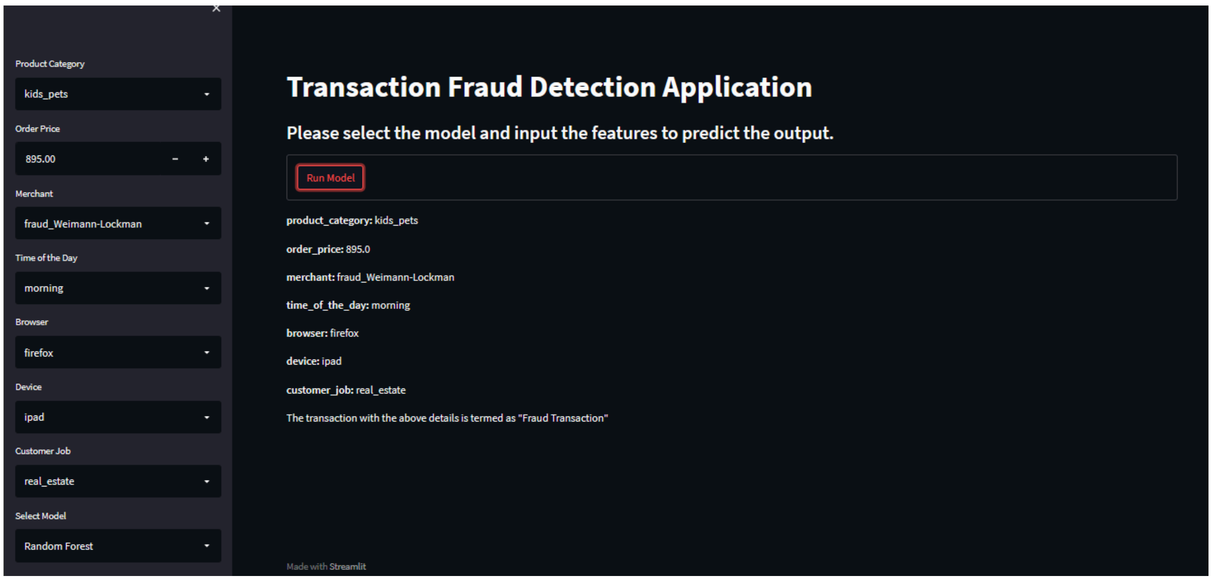
customer_job: politician

The transaction with the above details is termed as "Non-fraud/Legit Transaction"

Made with Streamlit

Random Forest

1. Fraud Transaction



Product Category

kids_pets

Order Price

895.00

Merchant

fraud_Weimann-Lockman

Time of the Day

morning

Browser

firefox

Device

ipad

Customer Job

real_estate

Select Model

Random Forest

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

Run Model

product_category: kids_pets

order_price: 895.0

merchant: fraud_Weimann-Lockman

time_of_the_day: morning

browser: firefox

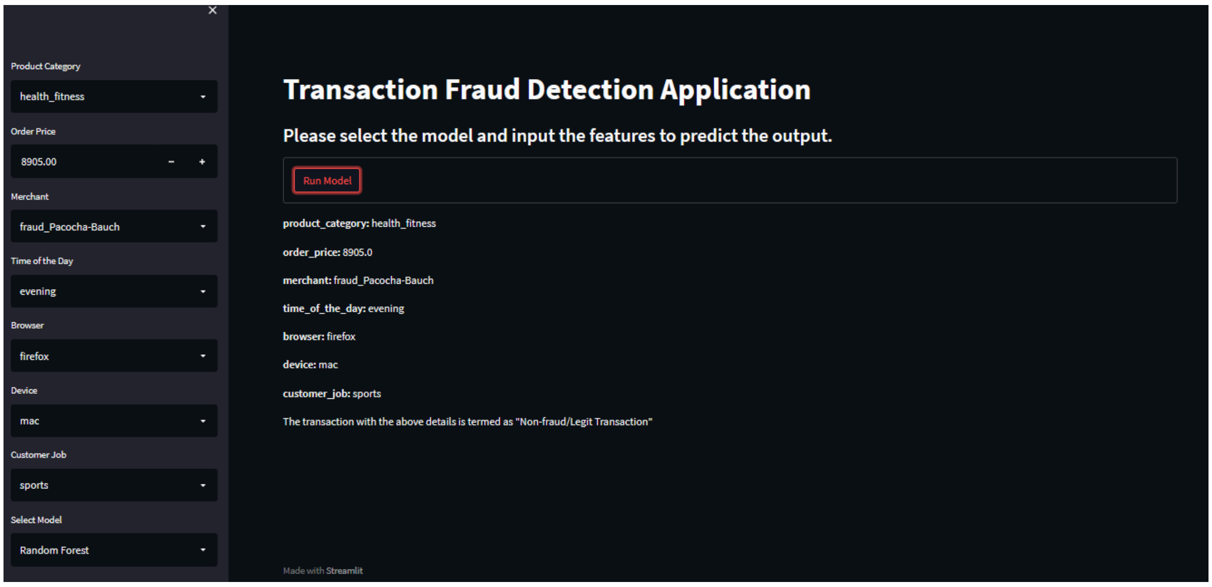
device: ipad

customer_job: real_estate

The transaction with the above details is termed as "Fraud Transaction"

Made with Streamlit

2. Non Fraud/ Legit Transaction



Product Category: health_fitness
Order Price: 8905.00
Merchant: fraud_Pacocha-Bauch
Time of the Day: evening
Browser: firefox
Device: mac
Customer Job: sports
Select Model: Random Forest

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: health_fitness

order_price: 8905.0

merchant: fraud_Pacocha-Bauch

time_of_the_day: evening

browser: firefox

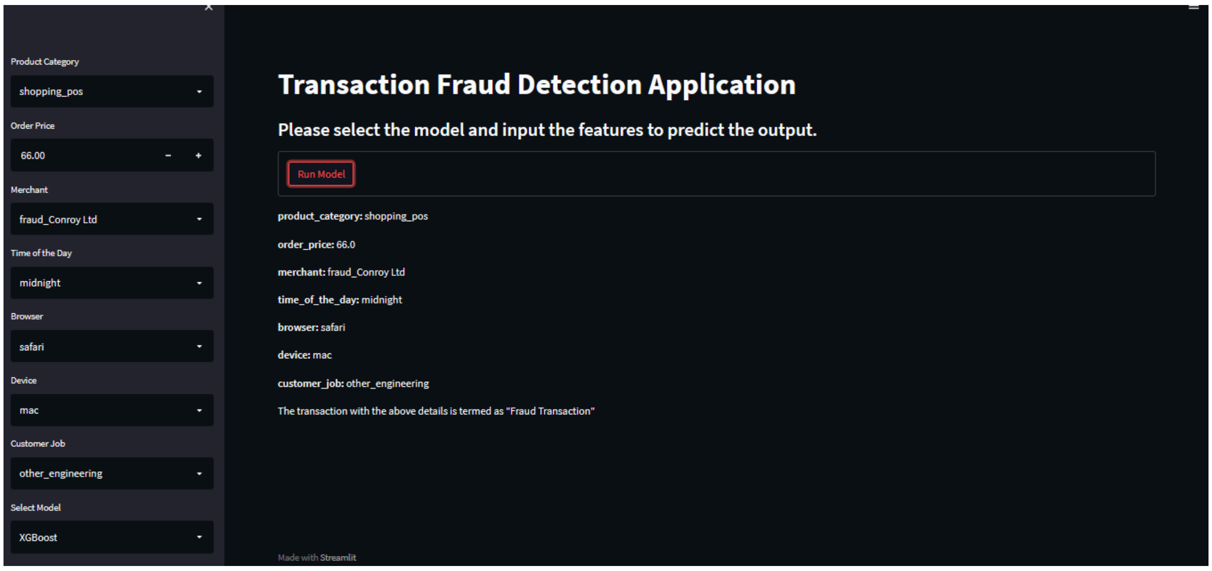
device: mac

customer_job: sports

The transaction with the above details is termed as "Non-fraud/Legit Transaction"

XGBoost

1. Fraud Transaction



Product Category: shopping_pos
Order Price: 66.00
Merchant: fraud_Conroy Ltd
Time of the Day: midnight
Browser: safari
Device: mac
Customer Job: other_engineering
Select Model: XGBoost

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: shopping_pos

order_price: 66.0

merchant: fraud_Conroy Ltd

time_of_the_day: midnight

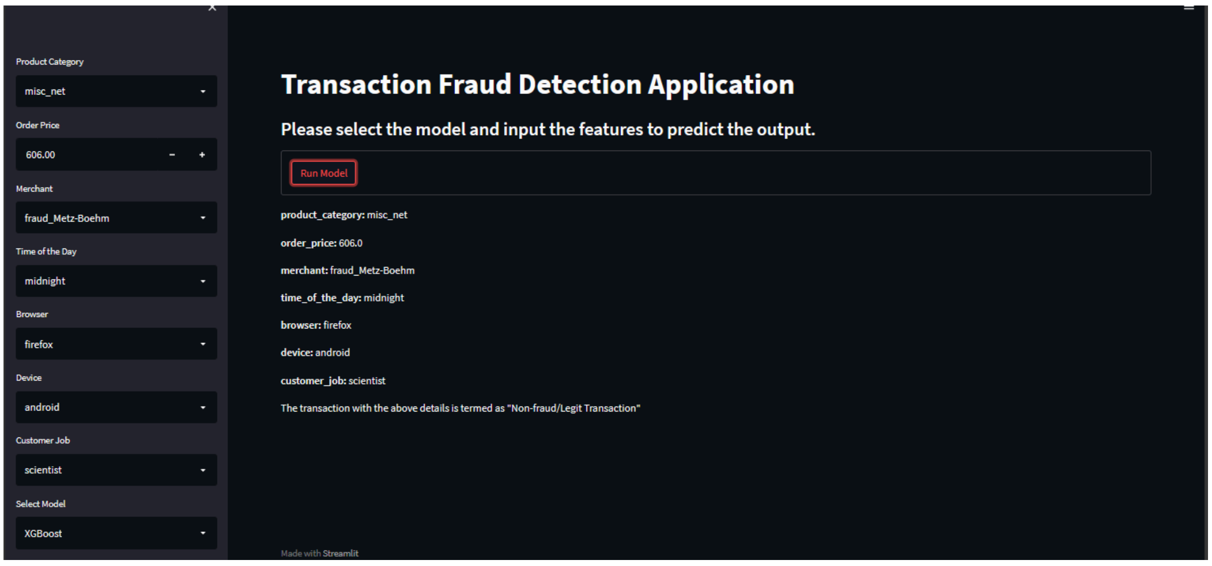
browser: safari

device: mac

customer_job: other_engineering

The transaction with the above details is termed as "Fraud Transaction"

2. Non Fraud/ Legit Transaction



Product Category: misc_net
Order Price: 606.00
Merchant: fraud_Metz-Boehm
Time of the Day: midnight
Browser: firefox
Device: android
Customer Job: scientist
Select Model: XGBoost

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: misc_net

order_price: 606.0

merchant: fraud_Metz-Boehm

time_of_the_day: midnight

browser: firefox

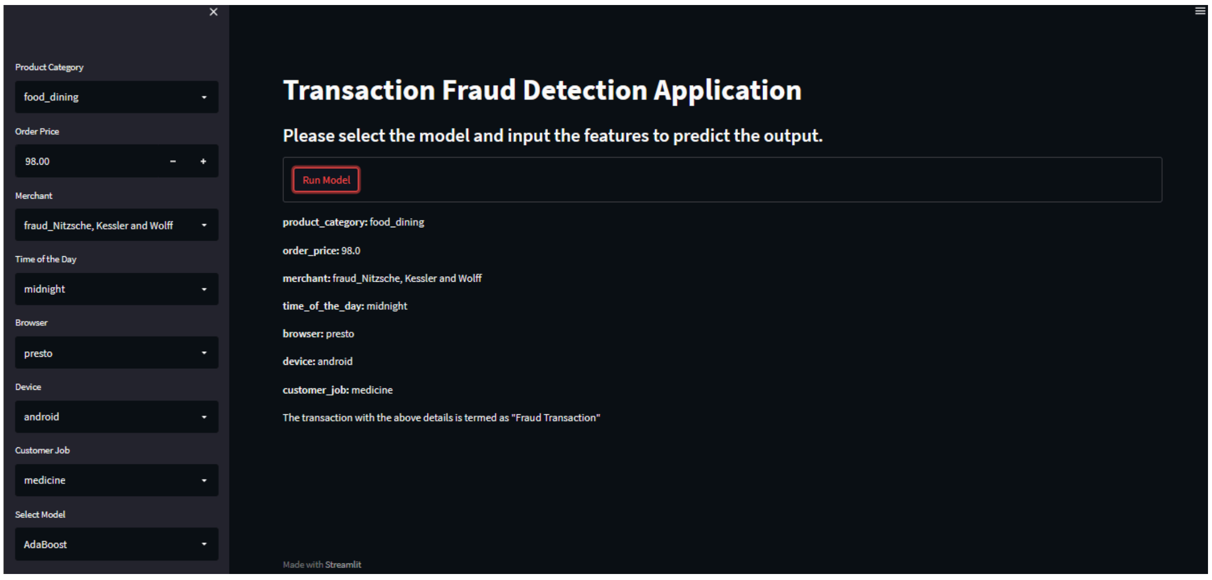
device: android

customer_job: scientist

The transaction with the above details is termed as "Non-fraud/Legit Transaction"

<u>AdaBoost</u>

1. Fraud Transaction



Product Category: food_dining
Order Price: 98.00
Merchant: fraud_Nitzsche, Kessler and Wolff
Time of the Day: midnight
Browser: presto
Device: android
Customer Job: medicine
Select Model: AdaBoost

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: food_dining

order_price: 98.0

merchant: fraud_Nitzsche, Kessler and Wolff

time_of_the_day: midnight

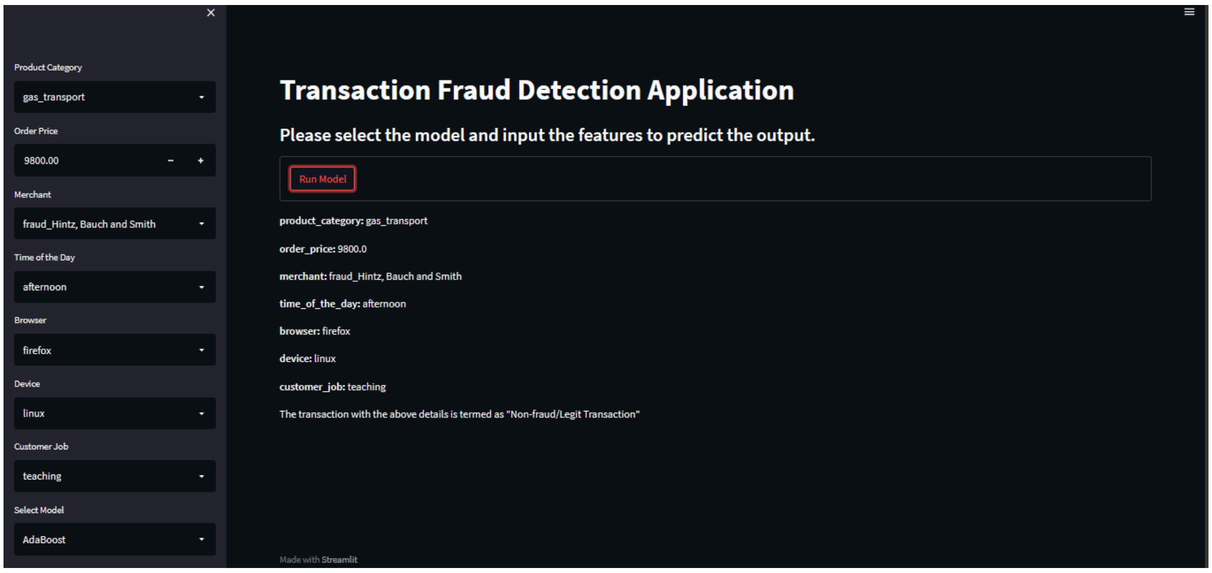
browser: presto

device: android

customer_job: medicine

The transaction with the above details is termed as "Fraud Transaction"

2. Non Fraud/ Legit Transaction


Product Category: gas_transport

Order Price: 9800.00

Merchant: fraud_Hintz, Bauch and Smith

Time of the Day: afternoon

Browser: firefox

Device: linux

Customer Job: teaching

Select Model: AdaBoost

Transaction Fraud Detection Application

Please select the model and input the features to predict the output.

product_category: gas_transport

order_price: 9800.0

merchant: fraud_Hintz, Bauch and Smith

time_of_the_day: afternoon

browser: firefox

device: linux

customer_job: teaching

The transaction with the above details is termed as "Non-fraud/Legit Transaction"

Recommendations:

Users are recommended to use the Transaction Fraud Detection Application for proactive fraud prevention, adjusting security measures in real-time according to the insights obtained. Teaching the users about fraud-detecting methods will help them make appropriate choices. The efficiency and accuracy of the detection can be enhanced by adding more prediction models and automating it. Blockchain integration can make the transactions even more secure. Customizable features catered to particular industry demands will greatly increase the application's usefulness and performance. Anti-fraud workflows can be improved by integration with business enterprise systems. These recommendations would assist users in detecting fraud and make the application more useful across many industries in the future.

Conclusion:

Considering the F1 Score, Precision, and Recall for Class 1, the Random Forest Classifier and XG Boost models seem to have better performance compared to the other models. The choice between the two depends on the specific requirements of the Amazon fraud detection dataset application. If precision and recall are equally important, consider the balance achieved by the Random Forest Classifier. If we want a slight improvement in recall at the cost of precision, XG Boost might be a better choice.

The Transaction Fraud Detection Application presents a robust tool designed to streamline the process of identifying potentially fraudulent transactions. The users can make sensible choices quickly because of the application's clear presentation of predicted results. The application demonstrates how machine learning algorithms can enhance security in online transactions. Tools such as these are essential to mitigate the number of fraudulent transactions as e-commerce keeps expanding. In the future, the application's adaptability and scalability will be important in ensuring its continued use in the evolving world of online transactions.

References:

[1] C. O'Neill and R. Schutt. Doing Data Science., O'Reilly. 2013.

[2] J. VanderPlas, Python Data Science Handbook., O'Reilly. 2016
[3] AWS Fraud Transaction Dataset
https://github.com/aws-samples/aws-fraud-detector-samples/blob/master/data/transaction_data_100K_full.csv
[4] XGBoost https://xgboost.readthedocs.io/en/stable/python/python_intro.html
[5] AdaBoost
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
[6] Streamlit
https://docs.streamlit.io/