Credit Fraud Catcher

Team 18: Yaren Dogan, Trevor McSharry, Makayla McKinney

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Introduction

Credit fraud is a rampant integrity concern in digital transactions and poses heavy financial risk to consumers and institutions alike. It is nearly impossible to completely secure a system to prevent any fraudulent transactions -- Therefore, this paper postulates that predicting and flagging likely fraudulent actions can serve as a more realistic option for online retailers and money businesses to ensure integrity in transactions.

Artificial Intelligence and Machine Learning are broad terms. This paper implements a Random Forest algorithm as a baseline model to understand where and when fraud occurs. RF was chosen as a benchmark due to its robust nature and versatile learning method.

Methodology

Imported the Credit Card Fraud Data dataset from Kaggle as a csv and turned it into a Pandas DataFrame called data.
Ran some analysis on the data with `data.describe()` and data.info()

- The information we extracted and found necessary for our use case included the data type of each feature in the dataset and the number of rows.
  - We realized the 'is_fraud' feature consisted of objects, either '0' for a valid transaction or a '1' for a fraudulent transaction.
    - We tried to convert the object type into integers; however, we soon realized that there were values other than '0' or '1'
    - We cleaned up that feature so that the only values would be either a '0' or '1', and then we converted the object type into an integer type
    - This reduced our row number from 14446 to 14444
  - Additionally, we dropped the 'merchant' and 'trans_num' features because we deemed these two features as unnecessary for our analysis.

Performed groupby to see how many fraudulent transactions happened in each state in the dataset, and obtained the ratio of 'Normal Transactions' to 'Fraudulent Transactions'

Next, we wanted to do Quantitative Feature Analysis to predict how accurate this data is at predicting fraud. Quatitative features include: ["amt", "city_pop", "distance_miles", "age", "hour"]

- Created the 'age' feature of the person by formatting the 'dob' and 'trans_date_trans_num' features and subtracting them from each other
- Created the 'hour' feature by extracting the hour from the 'trans_date_trans_num' feature.
- Created the 'distance_mile' feature by using the 'lat', 'long', 'merch_lat', and 'merch_long' feature. This feature captures how far away the transaction was made from the location the credit card was issued to the user.
  - We used the Haversine function to find this distance in miles.

From this point forward, we created:

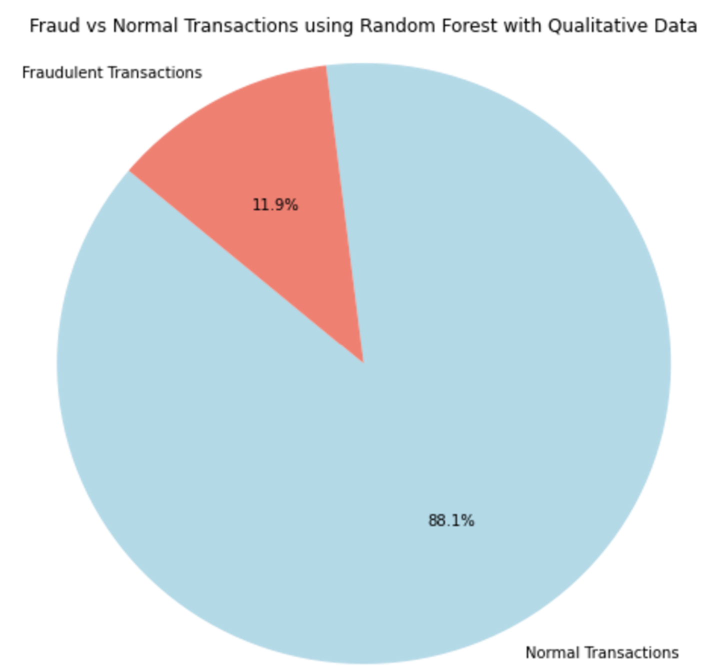
- Correlation Heatmap for the Quantitative Data, which includes transaction amount, city population, distance in miles, hour of day, and age.
- PCA Visualization of Credit Card Data with these Quantitative Features that explains around 80% of the variability (84%)
- Elbow Plot
- PCA Visualization of Credit Card data with K-Means Clusters

We then performed Random Forest, Decision Tree, and Logistic Regression, and the Ada Boosting algortithm to determine which one of these algorithms would have the highest accuracy.

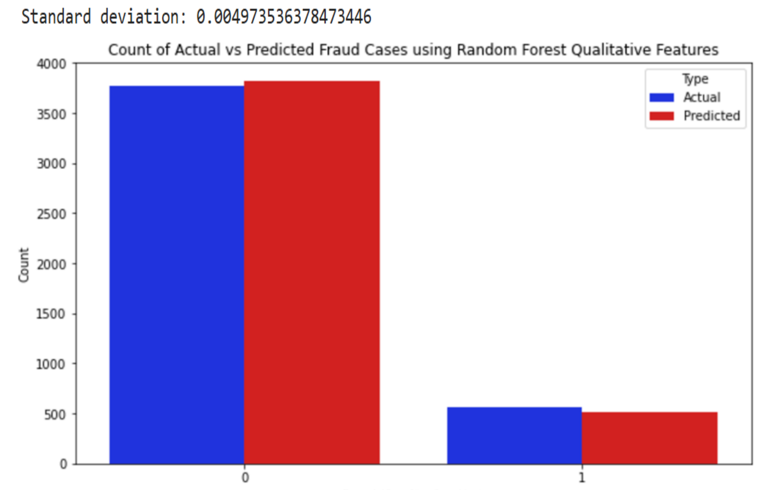
We wanted to also do some Categorical Feature Analysis to predict how accurate this data is at predicting fraud. Categorical features include: ["category", "state", "professions"]

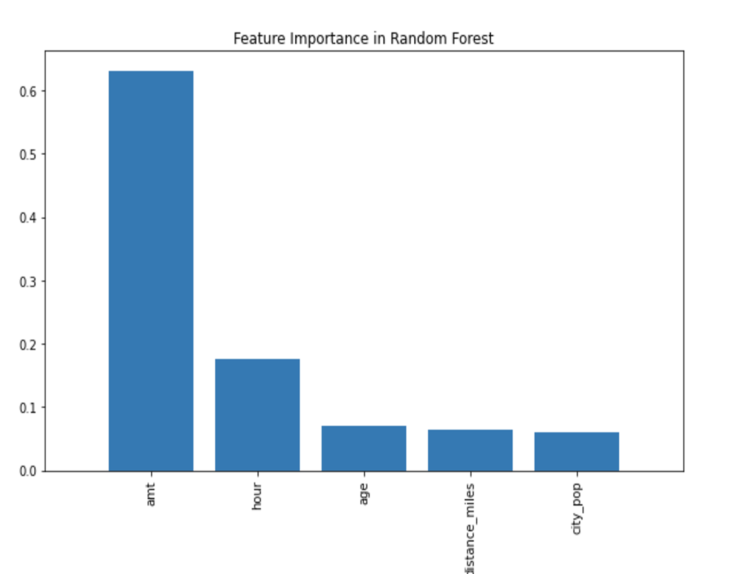
- Created 'professions' feature by manually grouping the different jobs into a larger category.
- We did the same analysis for the Categorical Data as we did for the Quantitative Data

Results and Analysis

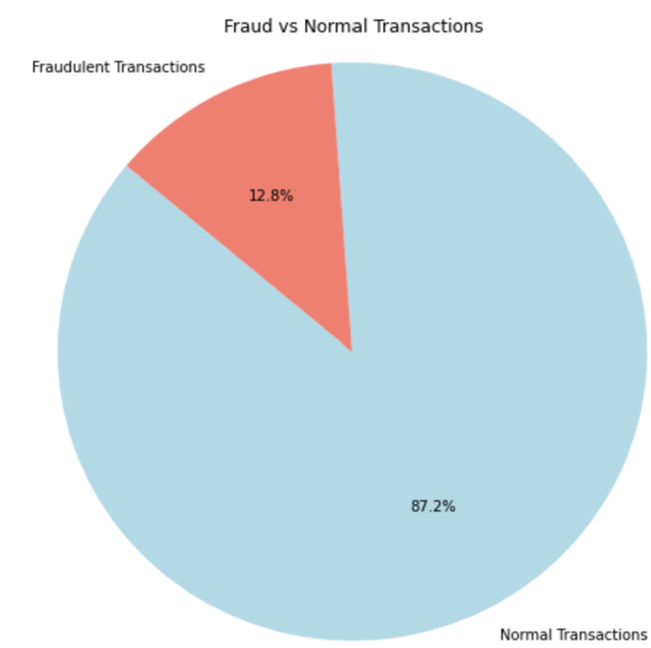


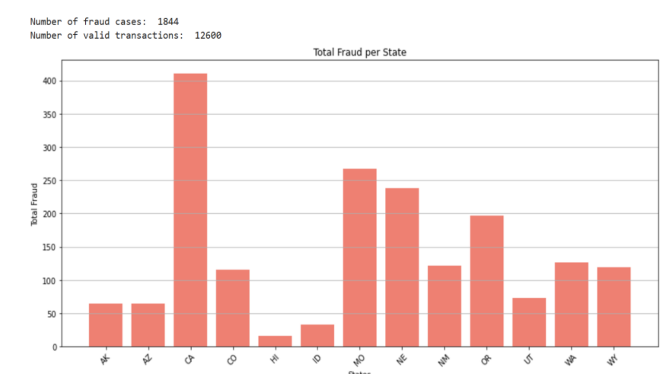

```
Accuracy for Random Forest for Quantitative Data: 0.9670050761421319
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      3772
           1       0.91      0.83      0.87       562

    accuracy                           0.97      4334
   macro avg       0.94      0.91      0.92      4334
weighted avg       0.97      0.97      0.97      4334

K-Folds with a Random Forest for Quantitative Features
Cross-validation scores: [0.96818811 0.97233748 0.96680498 0.9626556  0.96814404 0.966759
 0.96537396 0.966759   0.95844875 0.97091413 0.96260388 0.97506925
 0.96260388 0.9598338  0.96398892 0.97229917 0.97645429 0.966759
 0.95844875 0.96814404]
Mean score: 0.9666295023428851
Standard deviation: 0.004973536378473446
```

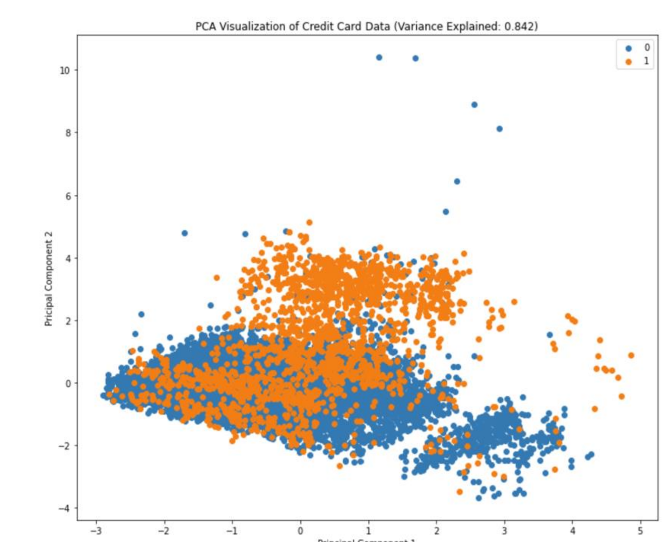


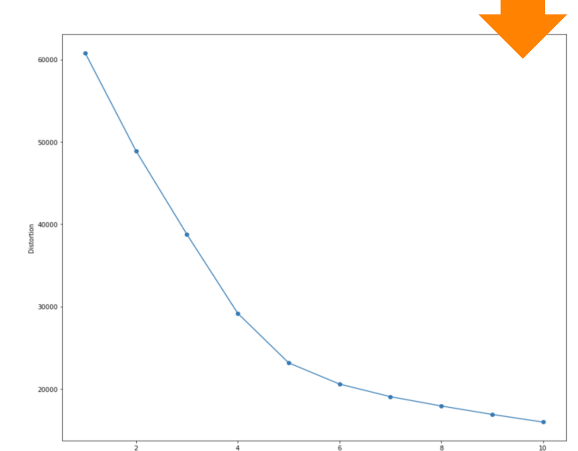


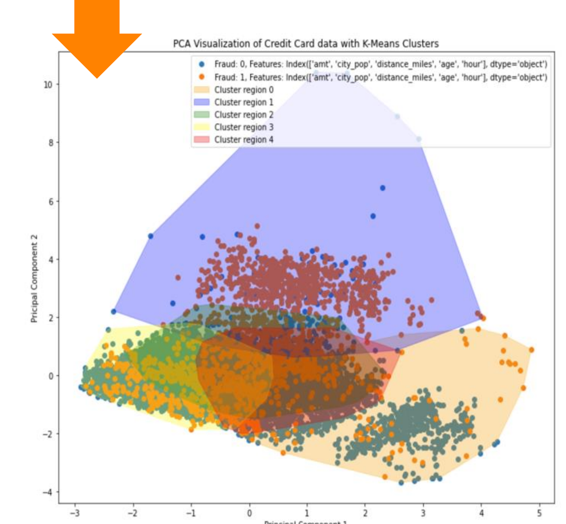
- Decision Tree Accuracy for Quantitative Features: .946
- Logistic Regression Accuracy for Quantitative Features: .919
- Ada Boosting Accuracy (Decision Tree) for Quantitative Features: .955

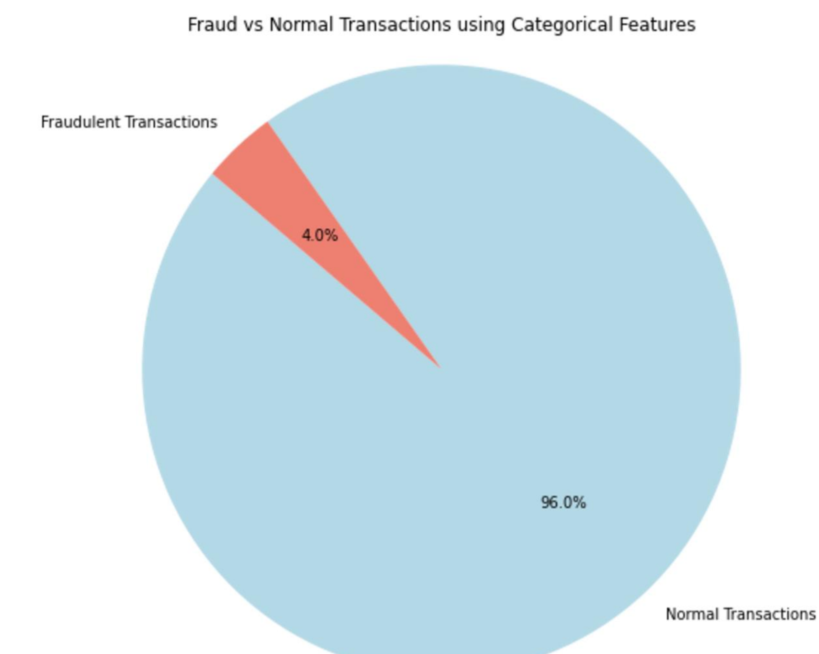




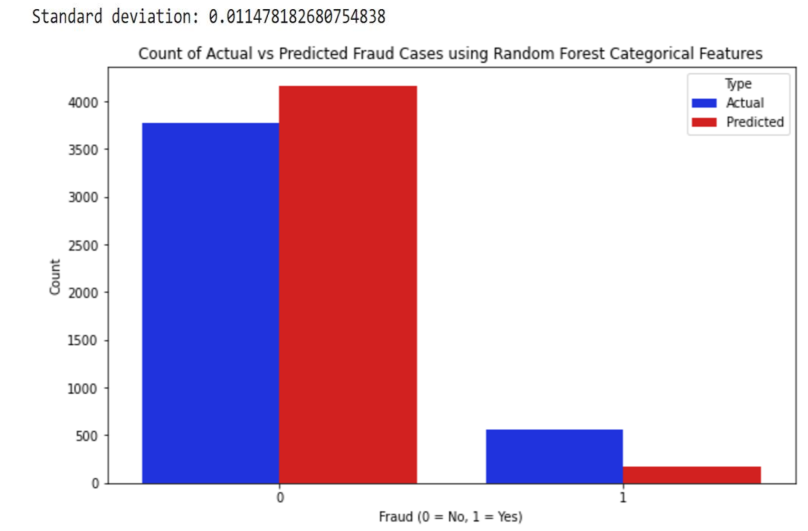


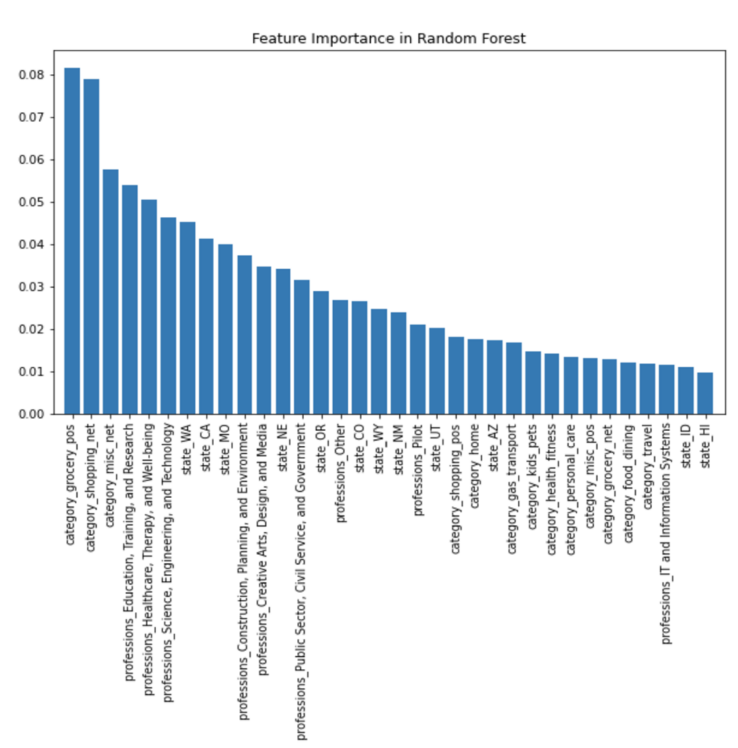




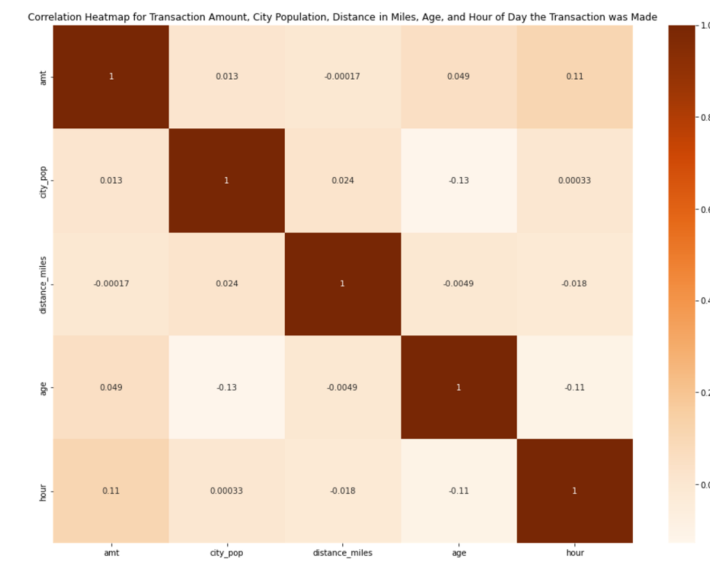


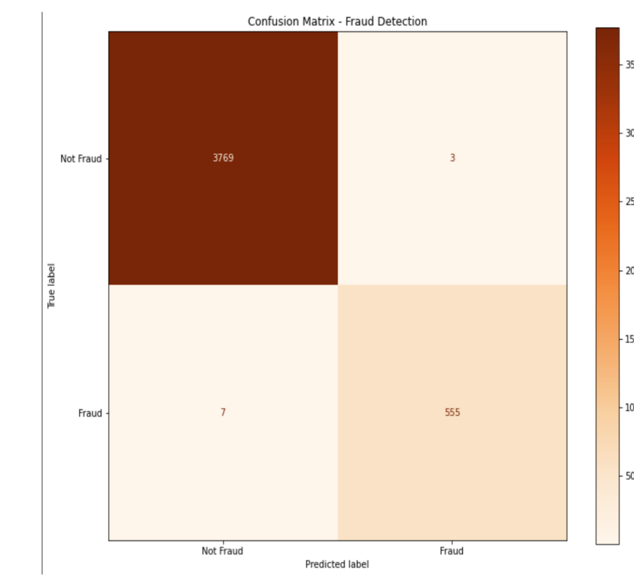

```
Random Forest for Catagorical Features
Accuracy: 0.8615597600369174
              precision    recall  f1-score   support

           0       0.88      0.97      0.92      3772
           1       0.39      0.12      0.18       562

    accuracy                           0.86      4334
   macro avg       0.64      0.55      0.55      4334
weighted avg       0.82      0.86      0.83      4334

K-Folds for Random Forest for Catagorical Features
Cross-validation scores: [0.87275242 0.88381743 0.86168741 0.85753804 0.88919668 0.86149584
 0.88781163 0.85872576 0.85041551 0.87534626 0.86980609 0.88365651
 0.86149584 0.86149584 0.88781163 0.87811634 0.87257618 0.87257618
 0.88504155 0.87950139]
Mean score: 0.8725432274724811
Standard deviation: 0.011478182680754838
```





- Decision Tree Accuracy for Categorical Features: .870
- Logistic Regression Accuracy for Categorical Features: .871
- Ada Boosting Accuracy (Decision Tree) for Quantitative Features: .871





Conclusion and Lessons Learned

To detect credit fraud, our group employed several different machine learning techniques. Our first implementation included a Random Forest Classifier, which was able to achieve 98% accuracy in detecting fraudulent transactions. We received feedback to implement K-Fold Cross Validation and Ada Boosting. The Boosting helped make our algorithm a stronger learner by boosting the training strength on classifications the algorithm got wrong. Implementing K-Fold Cross Validation did not result in an accuracy improvement.

References

1. https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost
2. https://www.kaggle.com/datasets/neharoychoudhury/credit-card-fraud-data/data
3. https://www.kaggle.com/code/stevenrockward/credit-card-transaction-fraud-prediction/notebook
4. https://stackoverflow.com/questions/63624633/pandas-info-not-showing-all-columns-and-datatypes
5. https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points

Acknowledgments

No Acknowledgments