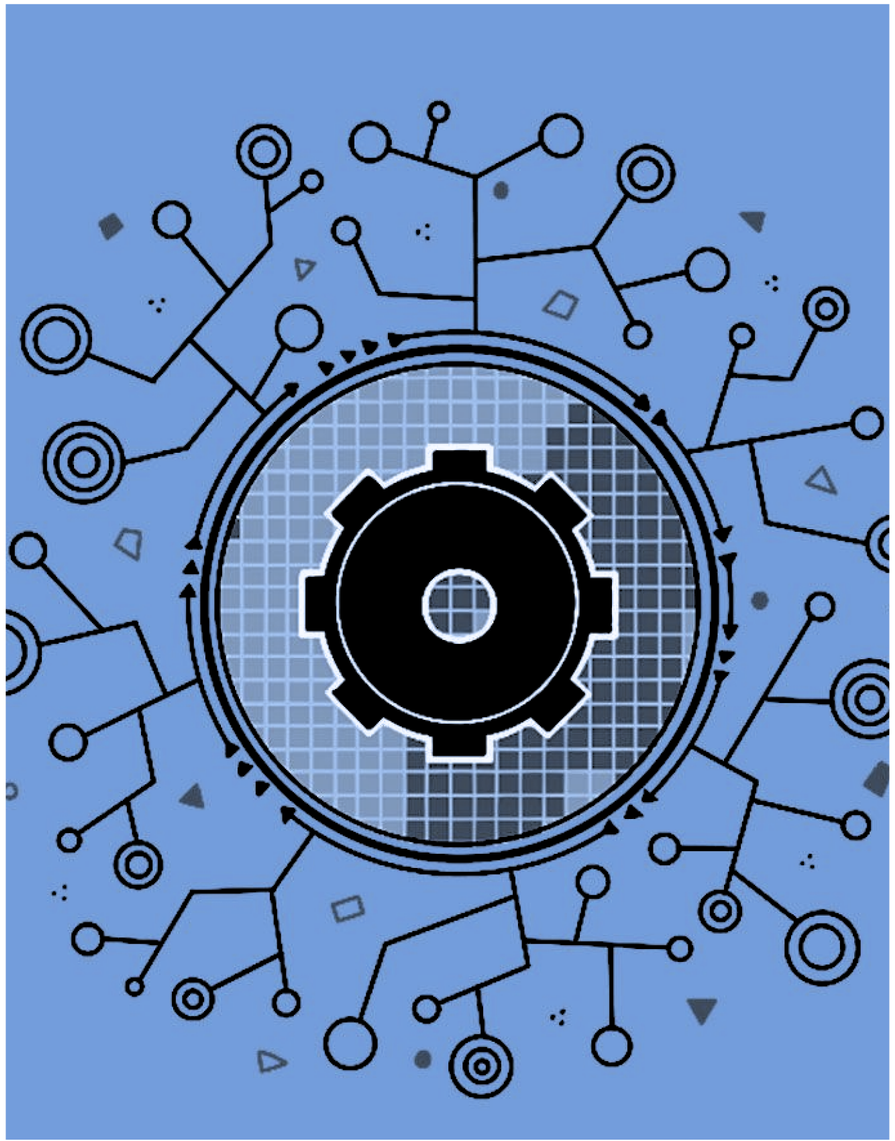
Navigating the Labyrinth of Credit Card Fraud: A Novel Perspective with Decision Tree Algorithms

Albert - 2501996045
Jordan Rubin Kurnia – 2501997426
Muhammad Raffi Hakim - 2502024312
Hanif Muhammad Sangga Buana - 2502049441



Background

- Credit cards are integral for modern financial transactions, offering convenience and enabling cashless transactions. Widespread use and the rise of online activities make credit cards susceptible to sophisticated fraud, leading to financial losses and eroding consumer trust.
- Enormous transaction volume, adaptive fraud techniques, and imbalanced occurrences pose challenges in detecting and preventing fraud. Opportunity exists to establish a system for detecting and remembering identities associated with fraudulent activities.

Solutions

Decision Trees

Decision tree algorithms, like RandomForest and AdaBoost, offer a solution for fraud detection.

RandomForest

Decision tree algorithms, like RandomForest and AdaBoost, offer a solution for fraud detection.

ADABOOSTER

AdaBoost, a versatile algorithm, shows strength when paired with strong algorithms, like Decision Trees, demonstrating speed and efficiency.

Related Works

- Decision Tree Algorithm Study (Tiwari et al.):
- Decision trees are popular for their ease of use and flexibility across different data types. RandomForest addresses instability and sensitivity of single trees by using ensembles, providing better computational efficiency. Two sources of randomness in RandomForest: bootstrapped samples and a random subset of data attributes for each tree, resulting in variance among trees.
- Improving Accuracy with Ensembles (Bagga & Others):
- RandomForest and AdaBoost are explored to enhance accuracy and overcome weaknesses of Decision-Tree algorithms. Ensembles combine multiple classifiers, improving overall performance. Iterative training with weighted classifiers ensures accurate predictions, especially for unusual behavior.
- Isolation Forest Algorithm (Cherif et al.):
- Cherif discusses the Isolation Forest algorithm, an anomaly detection method developed by Fei Tony Liu.Determines isolation of data by measuring how distant a data point is from others.Uses binary trees for faster anomaly detection, directly identifying anomalies without profiling regular instances.

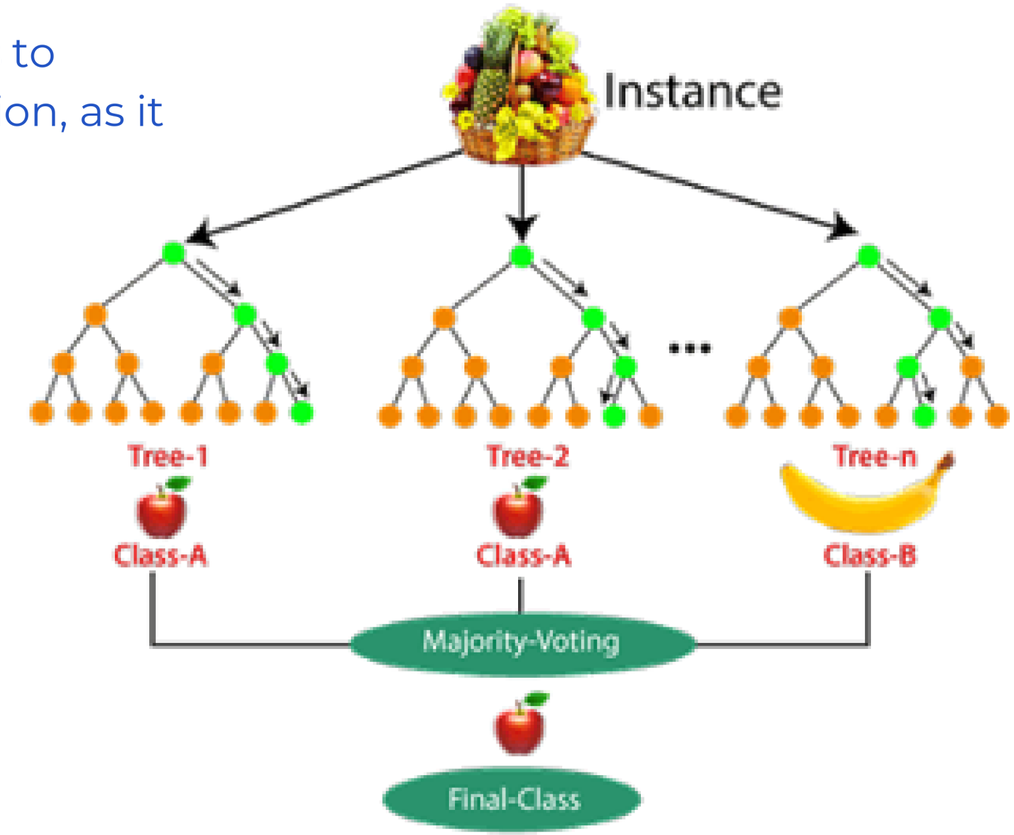
Methodology

- Data Selection & Preprocessing
  - Datasets used will be “Synthetic Financial Datasets For Fraud Detection” by Edgar Lopez-Rojaz that can be collected in Kaggle. The dataset contains ~6.36m variables with 11 attrbiutes for each label. The main attribute of said dataset is categorial as fraud/not fraud. There are problems
    - The datasets are simply huge, with each process that can take extremely long times.
    - Also said datasets were simply lopsided in terms of fraud classification. It risks overfitting issues.
  - Solutions include
    - Problems regarding the huge datasets can be resolved by shrinking the datasets into manageable size.
    - Overfitting can be resolved with Bootstrapping method, a resampling technique used to estimate the uncertainty or variability of a statistic or model parameter.
  - The final result of Datasets is a very manageable and balanced datasets with size of ~16.5k variables with balanced size of fraudulent & not-fraudlent sizes.

Data Transformation

Random Forest , which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

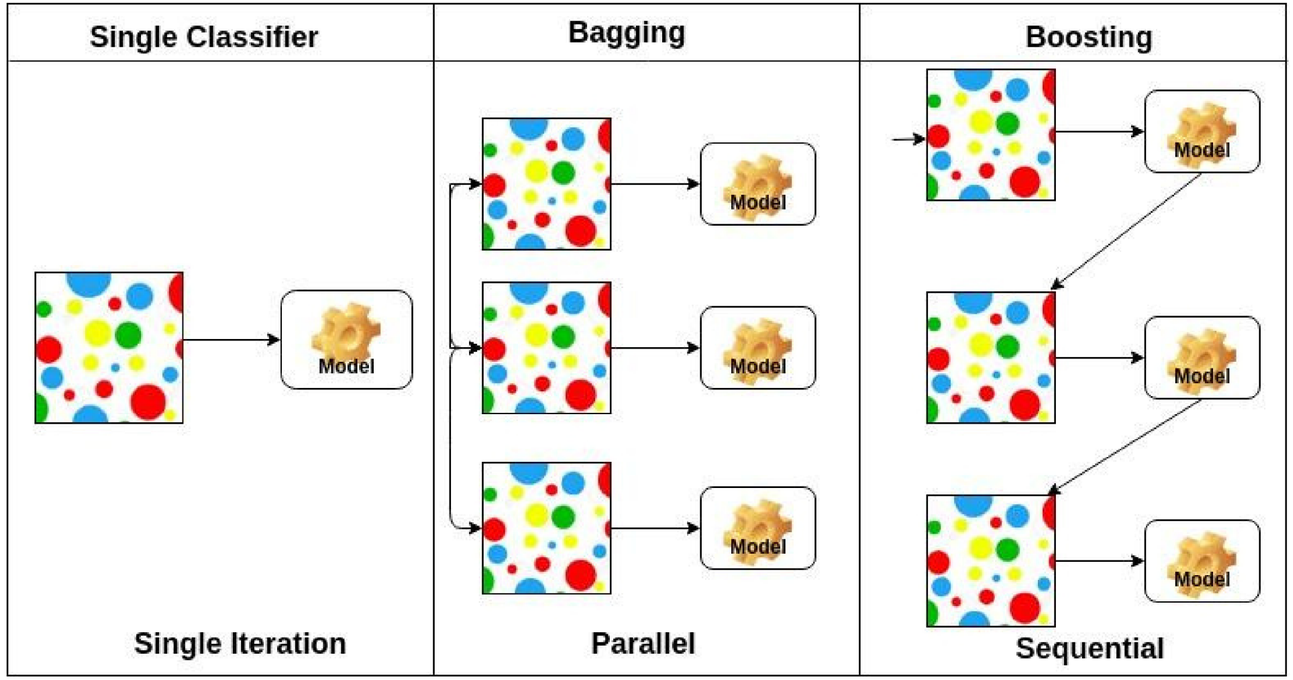
1. Subset Selection:
   - Randomly selects a subset of data points and features for each decision tree.
   - Takes "n" random records and "m" features from a dataset with "k" total records.
2. Decision Tree Construction:
   - Constructs individual decision trees for each sample.
3. Output Generation:
   - Each decision tree generates an output.
4. Final Result:
   - For classification, uses Majority Voting; for regression, employs Averaging.
   - Produces a final output based on the aggregated results.



Data Transformation

AdaBoost algorithm is a Boosting technique used as an Ensemble Method in Machine Learning. The weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

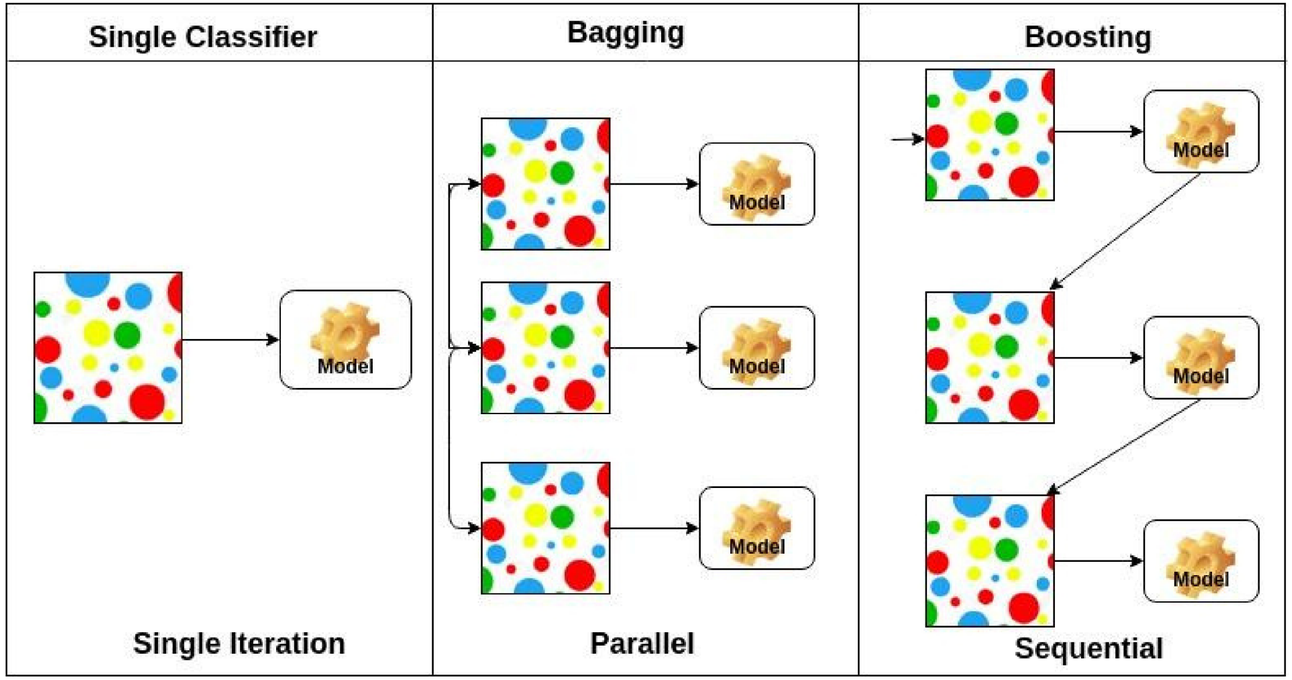
Step 1: Calculate sample Weight
$w(x_i, y_i) = \frac{1}{N}, \; i = 1, 2, \ldots . n$
Step 2: Sample Classification based ona vaialble attributes
Step3: Calculate the Influence towards the sample attribute category
$\frac{1}{2} \log \frac{1 - Total\ Error}{Total\ Error}$
Step 4: Calculate Total Error & Perfomance
$New\ sample\ weight = old\ weight * e^{\pm Amount\ of\ say\ (\alpha)}$
Step 5: Decrease Error by recalibrating the error/deviation
Step 6: Reclassify each value of dataset based on given classification
Step 7+: Reiterate all previous steps by recalculating sample weights and normalizing the value. How many iterations depends on how accurate we want.



Data Transformation & Mining

AdaBoost algorithm is a Boosting technique used as an Ensemble Method in Machine Learning. The weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

Step 1: Calculate sample Weight
$w(x_i, y_i) = \frac{1}{N}, \; i = 1, 2, \ldots . n$
Step 2: Sample Classification based ona vaialble attributes
Step3: Calculate the Influence towards the sample attribute category
$\frac{1}{2} \log \frac{1 - Total\ Error}{Total\ Error}$
Step 4: Calculate Total Error & Perfomance
$New\ sample\ weight\ =\ old\ weight\ *\ e^{\pm Amount\ of\ say\ (\alpha)}$
Step 5: Decrease Error by recalibrating the error/deviation
Step 6: Reclassify each value of dataset based on given classification
Step 7+: Reiterate all previous steps by recalculating sample weights and normalizing the value. How many iterations depends on how accurate we want.



Data Visualization

Integration of the code will use python 3.12.0 versions with several libraries installed and used for this experiment purposes

·**streamlit**: Streamlit is an open-source Python library that allows you to create custom web apps for machine learning and data science.
·**pandas**: Pandas is a powerful Python library for data analysis and manipulation.
·**Randomforest & adabooster**: respective algorithms used for algorithm purposes
·**StandardScaler**: Standardize features by removing the mean and scaling to unit variance to allows machine for an easier computation whilst maintaining the original value of the datasets
·**Train_test_split**: Splits the datasets for testing & validation purposes.

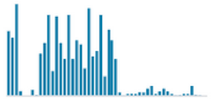
Datasets Used

Datasets that will be used will be “Synthetic Financial Datasets For Fraud Detection” by Edgar Lopez-Rojaz that can be collected in Kaggle. The dataset contains ~6.36m variables with 11 attrbiutes for each label.

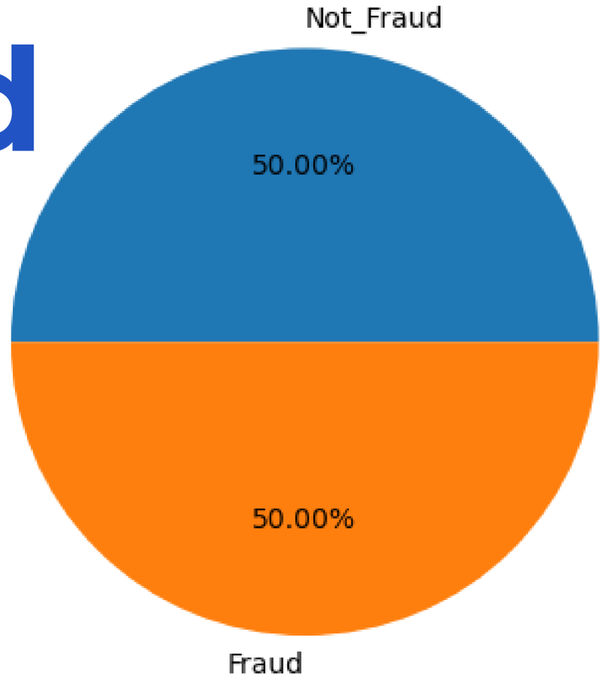
Datasets Used

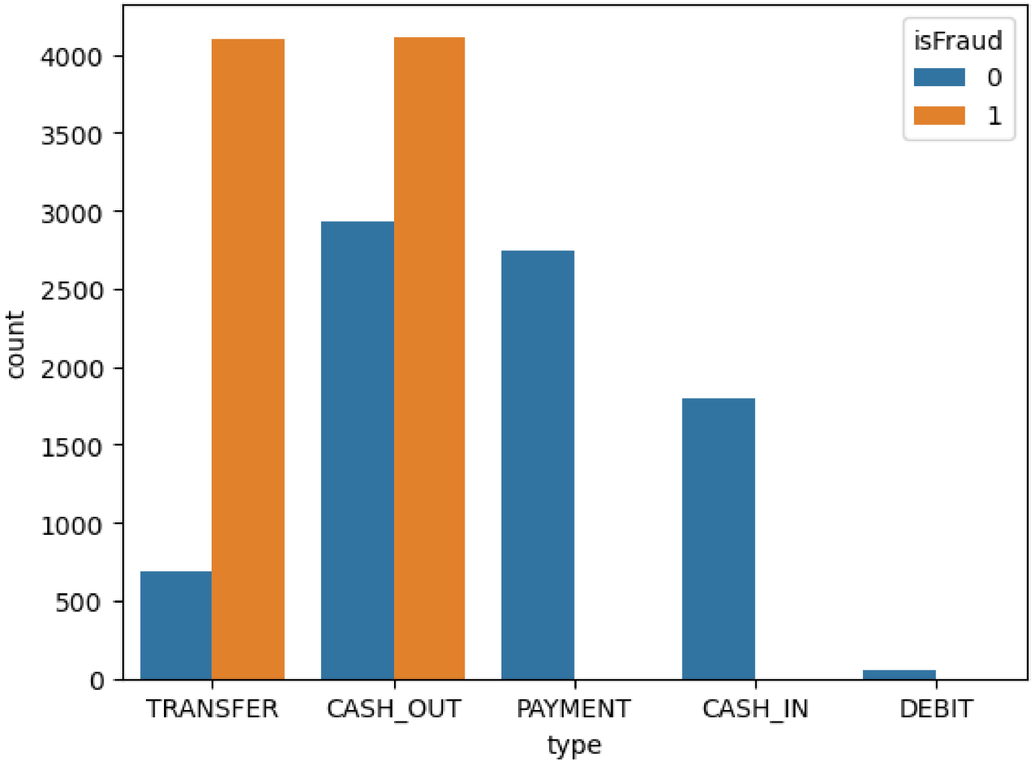
step - maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
type - CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
amount - amount of the transaction in local currency.
nameOrig - customer who started the transaction
oldbalanceOrg - initial balance before the transaction
newbalanceOrig - new balance after the transaction.
nameDest - customer who is the recipient of the transaction
oldbalanceDest - initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).
newbalanceDest - new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).
isFraud - This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers accounts and try to empty the funds by transferring to another account and then cashing out of the system.
isFlaggedFraud - The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

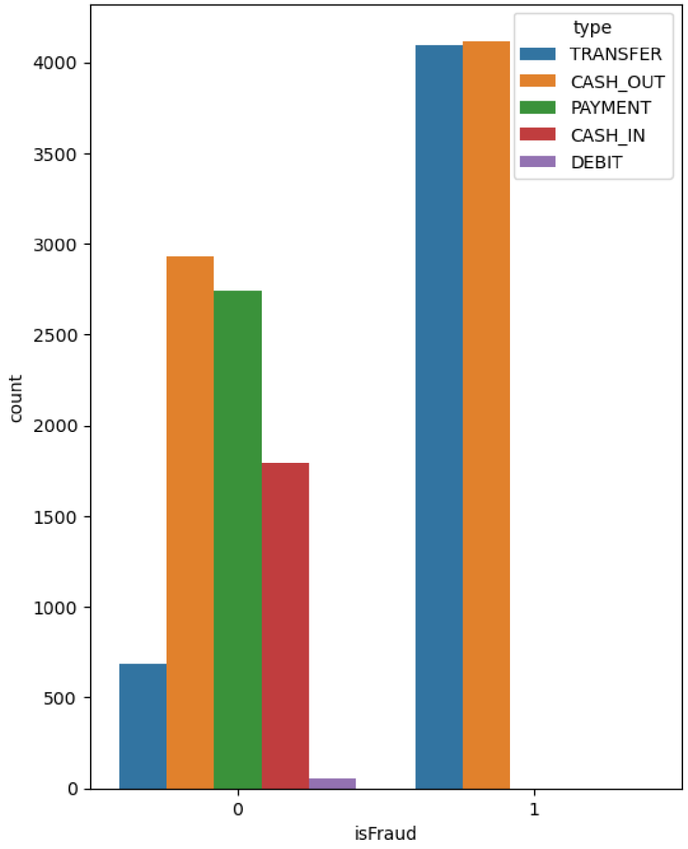
Datasets Used

# step	A type	# amount	A nameOrig	# oldbalanceOrg	# newbalanceOrig	A nameDest	# oldbalanceDest	# newbalanceDest	# isFraud
Maps a unit of time in the real world. In this case 1 step is 1 hour of time.	CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER	amount of the transaction in local currency	customer who started the transaction	initial balance before the transaction	customer's balance after the transaction.	recipient ID of the transaction.	initial recipient balance before the transaction.	recipient's balance after the transaction.	identifies a fraudulent transaction (1) and non fraudulent (0)
 1 743	CASH_OUT 35% PAYMENT 34% Other (1973625) 31%	0 92.4m	6353307 unique values	 0 59.6m	 0 49.6m	2722362 unique values	0 356m	0 356m	 0 1
1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0
1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0
1	TRANSFER	181.0	C1305486145	181.0	0.0	C553264065	0.0	0.0	1
1	CASH_OUT	181.0	C840083671	181.0	0.0	C38997010	21182.0	0.0	1
1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0
1	PAYMENT	7817.71	C90045638	53860.0	46042.29	M573487274	0.0	0.0	0
1	PAYMENT	7107.77	C154988899	183195.0	176087.23	M408069119	0.0	0.0	0
1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0.0	0.0	0
1	PAYMENT	4024.36	C1265012928	2671.0	0.0	M1176932104	0.0	0.0	0
1	DEBIT	5337.77	C712410124	41720.0	36382.23	C195600860	41898.0	40348.79	0
1	DEBIT	9644.94	C1900366749	4465.0	0.0	C997608398	10845.0	157982.12	0
1	PAYMENT	3099.97	C249177573	20771.0	17671.03	M2096539129	0.0	0.0	0
1	PAYMENT	2560.74	C1648232591	5070.0	2509.26	M972865270	0.0	0.0	0
1	PAYMENT	11633.76	C1716932897	10127.0	0.0	M801569151	0.0	0.0	0
1	PAYMENT	4098.78	C1026483832	503264.0	499165.22	M1635378213	0.0	0.0	0
1	CASH_OUT	229133.94	C905080434	15325.0	0.0	C476402209	5083.0	51513.44	0
1	PAYMENT	1563.82	C761750706	450.0	0.0	M1731217984	0.0	0.0	0
1	PAYMENT	1157.86	C1237762639	21156.0	19998.14	M1877062907	0.0	0.0	0
1	PAYMENT	671.64	C2033524545	15123.0	14451.36	M473053293	0.0	0.0	0
1	TRANSFER	215310.3	C1670993182	705.0	0.0	C1100439041	22425.0	0.0	0
1	PAYMENT	1373.43	C20804602	13854.0	12480.57	M1344519051	0.0	0.0	0
1	DEBIT	9302.79	C1566511282	11299.0	1996.21	C1973538135	29832.0	16896.7	0

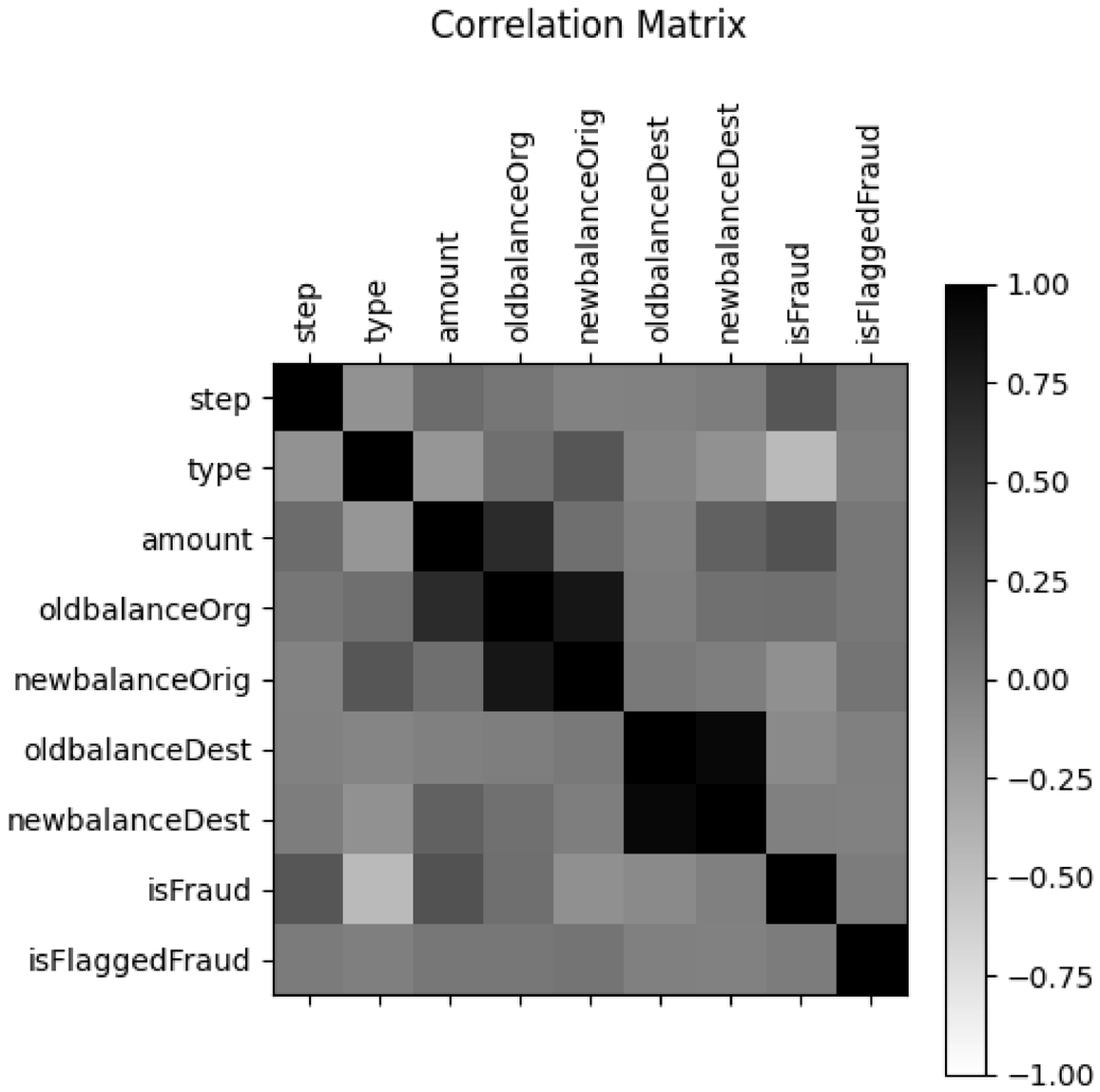
Datasets Used



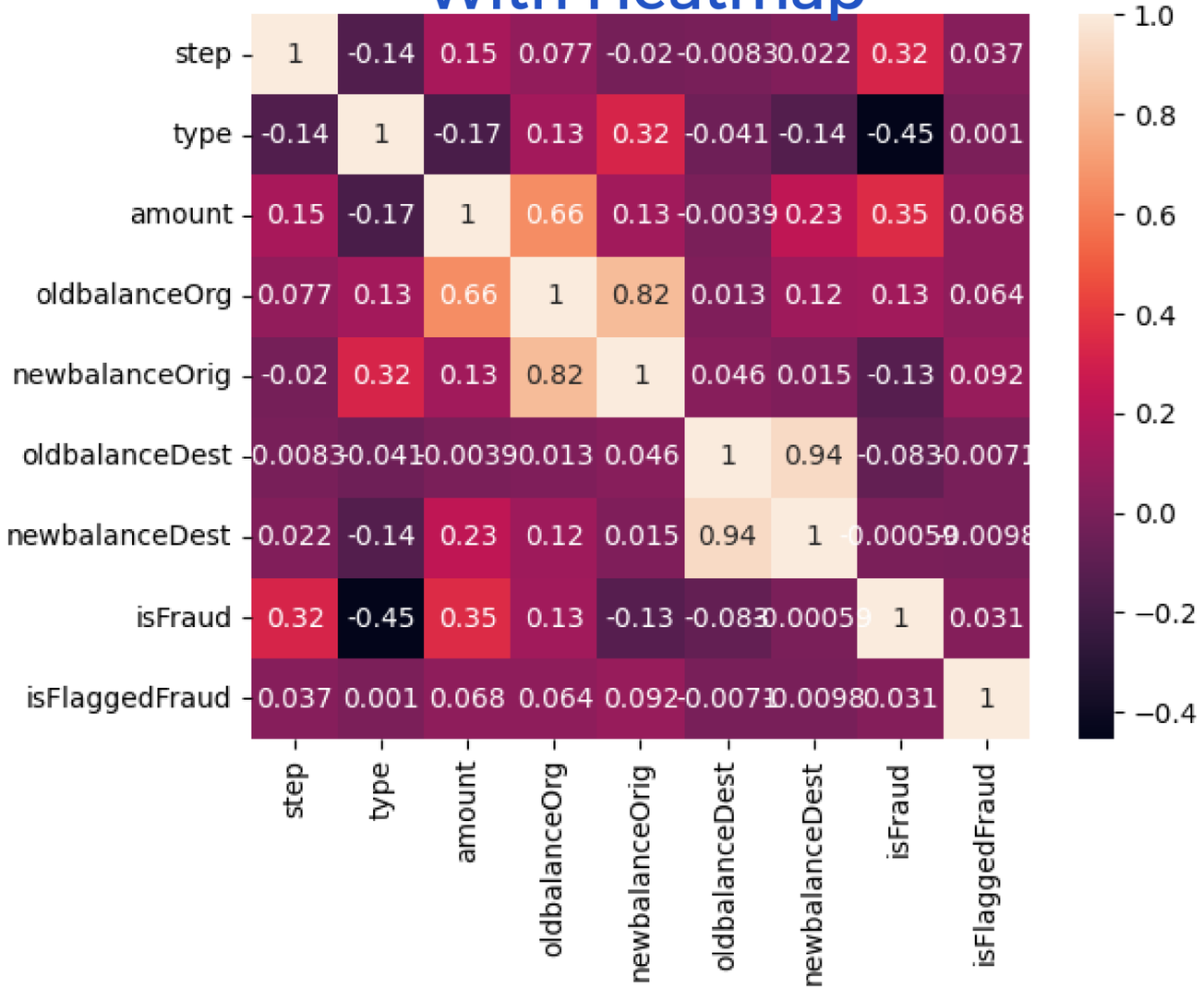




Datasets Correlation



With Heatmap



Dataset Visualization

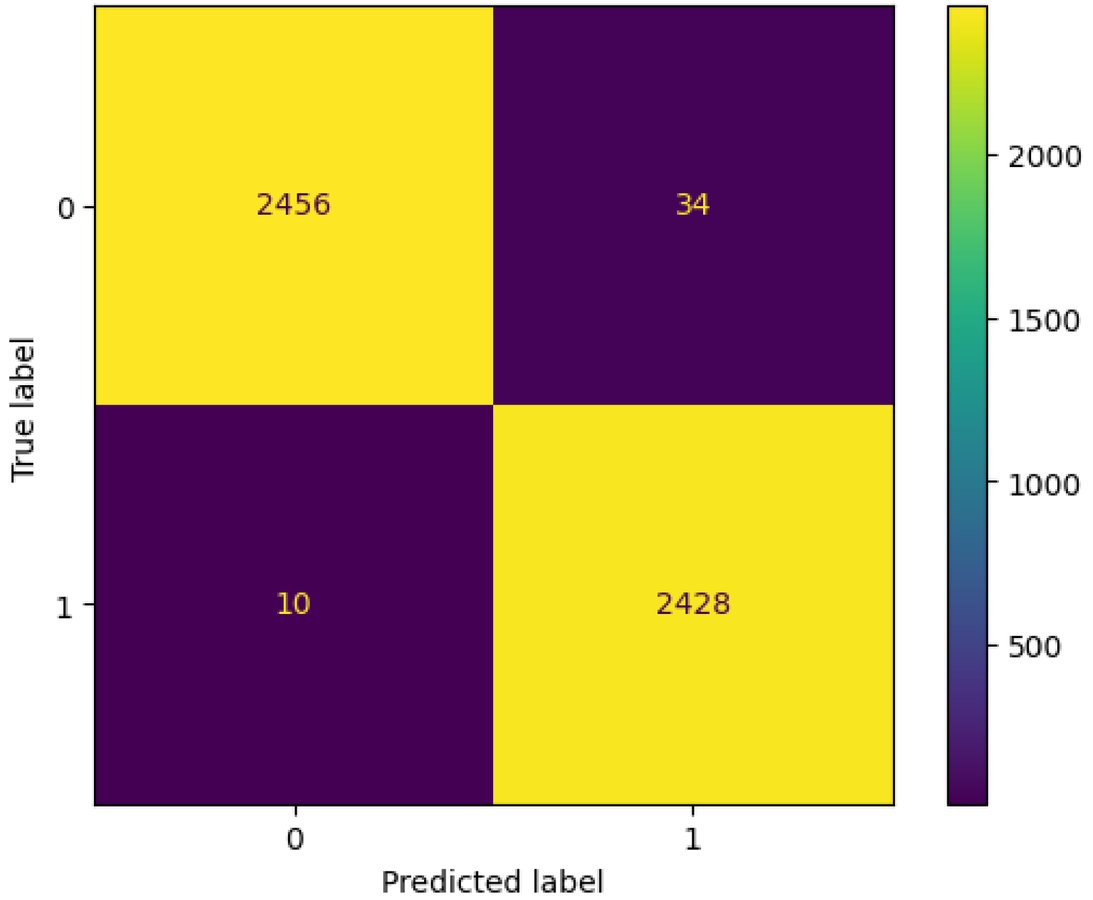
- Users must input various Data into machine before they can determine whenever said interaction constitutes as fraudulent action
-

step	type	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
16	PAYMENT	7103.96	20104.0	13000.04	0.00	0.00	0	0

- After inputting various necessary value for classification, user can choose which kind of algorithms that wanted to be used.
- After confirming the value & algorithms to be used, system first needs to check whevener inserted ID, both origin & destination were involved in fraudulent processes, if both were found to have fraudulent record the system immediately consider both as fraudulent and skips the process. Otherwise, the system will apply the trained model into assigned value to determine whenever the system were considered as fraudulent actions. Users will be notified whenever this transaction is considered as fraudulent and stores implicated IDs of both Origin & Destination for further classifications.
-

Result

Random Forest



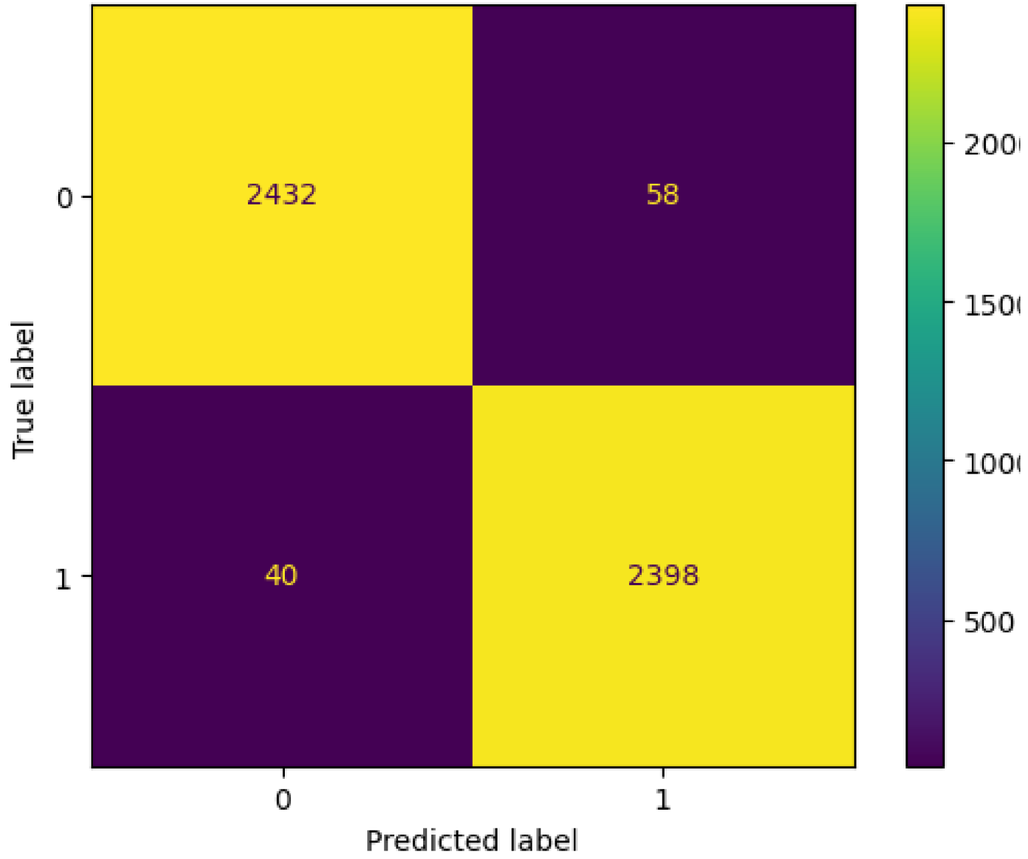
Result

Random Forest

```
[58]  print(classification_report(y_test,y_pred5))
      ✓ 0.0s

...                   precision    recall  f1-score   support

                0       1.00      0.99      0.99      2490
                1       0.99      1.00      0.99      2438

         accuracy                           0.99      4928
        macro avg       0.99      0.99      0.99      4928
     weighted avg       0.99      0.99      0.99      4928
```

Result

ADABoost Classifier



Result

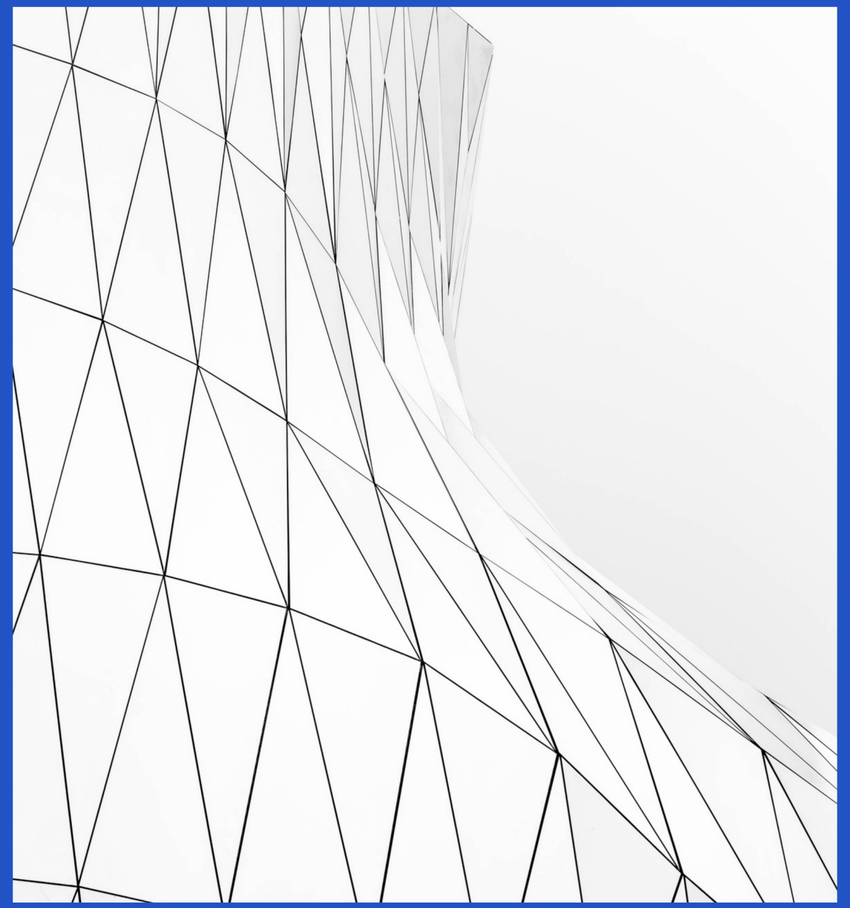
ADABoost Classifier

```
[62]  print(classification_report(y_test,y_pred6))
      ✓ 0.0s

...                   precision    recall  f1-score   support

                0       0.98      0.98      0.98      2490
                1       0.98      0.98      0.98      2438

         accuracy                           0.98      4928
        macro avg       0.98      0.98      0.98      4928
     weighted avg       0.98      0.98      0.98      4928
```

Result, Discussion, & Implications

- Based on this test, we have determined several findings, regarding about how

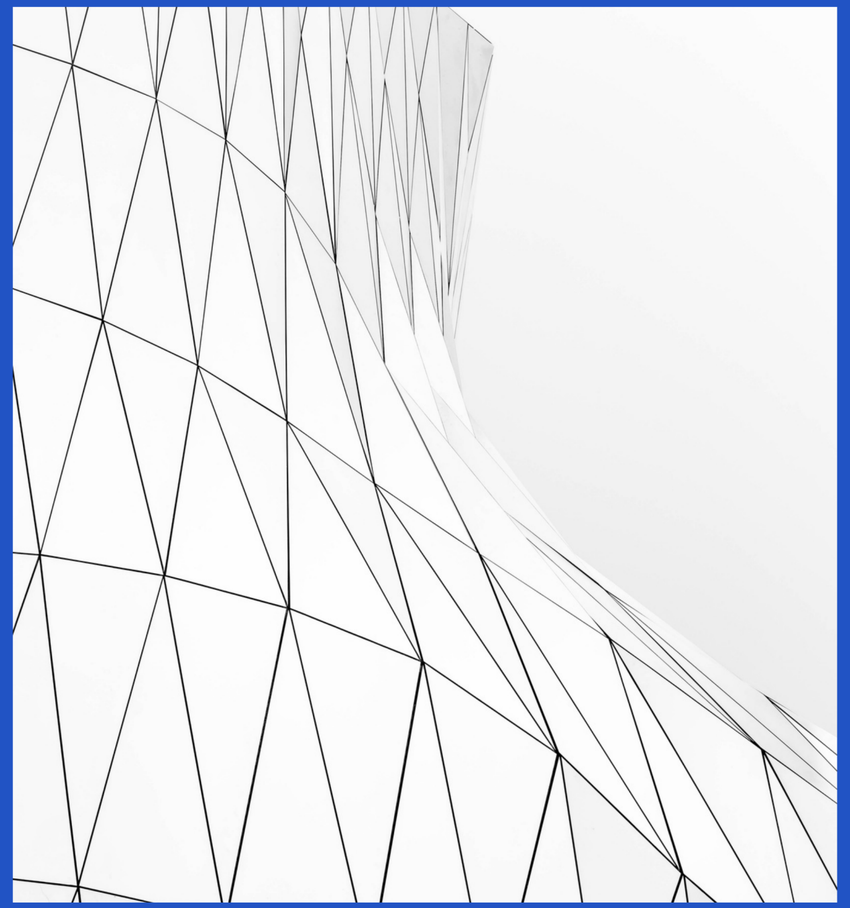
    - RandomForest & AdaBoosterbarely closes in eachother, however RandomForest is slightly better based than AdaBooster on the tests.
    - RandomForest Capability of handling overfitting or large data definitely helps with processing and classification
    - AdaBoost capability for learning from previous adjustment helps improve the calculation & classification and less prone to overfitting compared to other weak learners
- The Further Implication of this founding is that it`s generally possible to create a relatively simple system that can detect and recognize and also classify said transaction with reported value as fraudulent actions. Said system also demonstrates the capability & flexibility if Decision Tree algorithm as algorithm easy to apply and be taken advantage of in various scenarios.
-



Conclusion

- Through the implementation of AdaBoost and RandomForest, two powerful decision tree-based algorithms, we have demonstrated their efficacy in navigating the complex landscape of credit card fraud detection. Furthermore, our exploration into the capabilities of AdaBoost highlights its strength in learning from previous adjustments, enhancing both calculation and classification. Notably, AdaBoost proves to be less prone to overfitting compared to other weak learners, contributing to its reliability in the detection of fraudulent activities.
- The practical implication of our findings suggests the feasibility of developing a straightforward yet effective system for detecting, recognizing, and classifying fraudulent transactions.

- These insights pave the way for the development of more sophisticated and reliable fraud detection mechanisms, ensuring a safer and more secure future for credit card transactions globally.



Thank You