Anti-Money Laundering Detection with Machine Learning

Team Members

- **Sophia Robles** – srobles2@hawk.iit.edu
- **Nicholas Simpkins** – nsimpkins@hawk.iit.edu
- **Harsh Patel** – hpatel100@hawk.iit.edu

Introduction and Motivation

Money laundering is a multi-trillion-dollar international underground market.

It is difficult for various financial institutions across the globe to properly identify ML within their systems.

This is why automated detection systems are important, monitor, flag and review possible ML transactions in real time.

The goal of this project is to develop ML models that can detect fraudulent transactions.

Dataset Overview and Research Goals

- **Source**: IBM AML Dataset (Kaggle)
- **Size**: ~13 million transactions, 1.1GB (combined)
- **Features**: Transaction metadata including bank codes, amounts, currencies, account info, and timestamps

- **Research Goals (As of now)**
  - **Classification**: Predict whether a transaction is laundering-related
  - **Anomaly Detection**: Flag novel suspicious activity without labels
  - **Modeling**: Detect abnormal transaction behavior over time

Dataset Features

Key Features:

- Timestamp
- Sender/Receiver Bank Codes
- Starting/Ending Accounts
- Amount Received/Paid
- Currencies
- Payment Format (i.e., USD, UKP, Bitcoin and other formats etc.)
- isLaundering (target variable)

Data Types:

(float64(2), int64(3), object(6))

Supervised Learning

01

Determine whether an exchange is fishy.

02

By using, Decision Trees, Random Forest, Bagging, XGBoost, SVC, Neural Networks.

03

Evaluation: Accuracy, Precision, Recall, F1, AUC-ROC etc.

Three Main datasets

01

Filtered Bitcoin dataset: 461347 rows × 11 columns

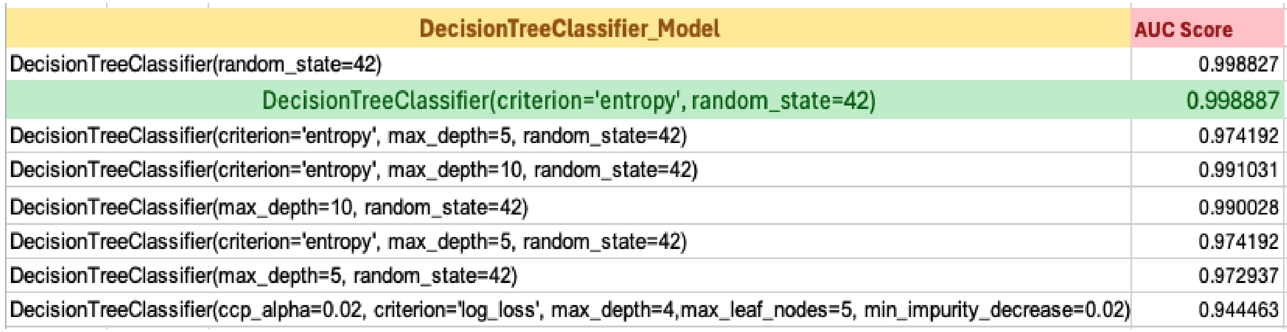
02

Filtered UK Pound dataset: 279255 rows × 11 columns

03

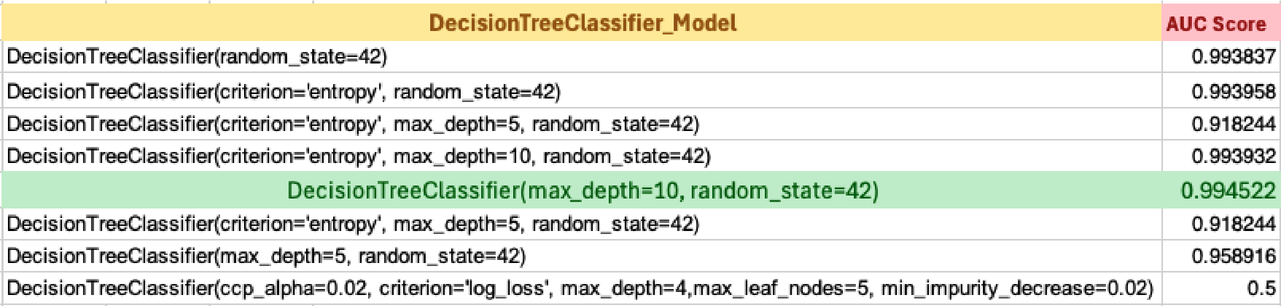
Filtered USA Dollar dataset: 300000 rows × 11 columns

Decision Tree -- USD



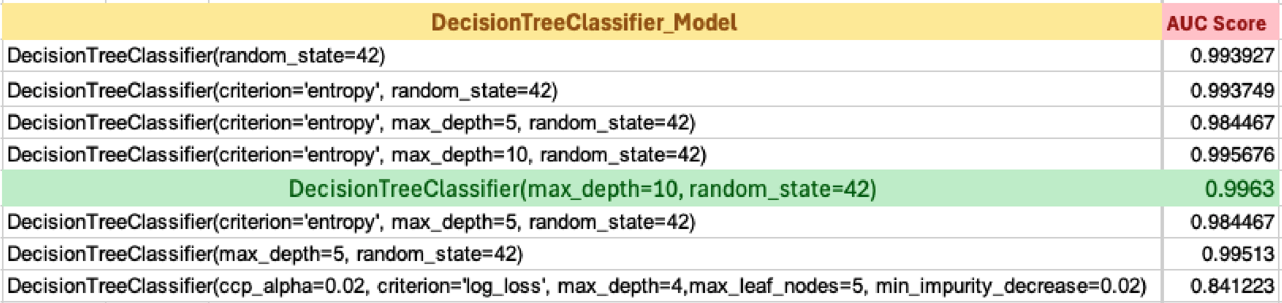
DecisionTreeClassifier_Model	AUC Score
DecisionTreeClassifier(random_state=42)	0.998827
DecisionTreeClassifier(criterion='entropy', random_state=42)	0.998887
DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)	0.974192
DecisionTreeClassifier(criterion='entropy', max_depth=10, random_state=42)	0.991031
DecisionTreeClassifier(max_depth=10, random_state=42)	0.990028
DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)	0.974192
DecisionTreeClassifier(max_depth=5, random_state=42)	0.972937
DecisionTreeClassifier(ccp_alpha=0.02, criterion='log_loss', max_depth=4,max_leaf_nodes=5, min_impurity_decrease=0.02)	0.944463

Decision Tree -- Bitcoin



DecisionTreeClassifier_Model	AUC Score
DecisionTreeClassifier(random_state=42)	0.993837
DecisionTreeClassifier(criterion='entropy', random_state=42)	0.993958
DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)	0.918244
DecisionTreeClassifier(criterion='entropy', max_depth=10, random_state=42)	0.993932
DecisionTreeClassifier(max_depth=10, random_state=42)	0.994522
DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)	0.918244
DecisionTreeClassifier(max_depth=5, random_state=42)	0.958916
DecisionTreeClassifier(ccp_alpha=0.02, criterion='log_loss', max_depth=4,max_leaf_nodes=5, min_impurity_decrease=0.02)	0.5

Decision Tree -- UKD



DecisionTreeClassifier_Model	AUC Score
DecisionTreeClassifier(random_state=42)	0.993927
DecisionTreeClassifier(criterion='entropy', random_state=42)	0.993749
DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)	0.984467
DecisionTreeClassifier(criterion='entropy', max_depth=10, random_state=42)	0.995676
DecisionTreeClassifier(max_depth=10, random_state=42)	0.9963
DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)	0.984467
DecisionTreeClassifier(max_depth=5, random_state=42)	0.99513
DecisionTreeClassifier(ccp_alpha=0.02, criterion='log_loss', max_depth=4,max_leaf_nodes=5, min_impurity_decrease=0.02)	0.841223

Logistic Regression – Bitcoin Results

STEP-1: **Total dataset**: 12002394 rows × 11 columns

STEP-2: **Filtered Bitcoin dataset**:
461347 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder**: 461347 rows × 38 cols

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
346010 rows × 37 cols
y-train: **(Imbalanced)**
346010 rows × 1 col
(i.e., 0: 345886, 1: 124)

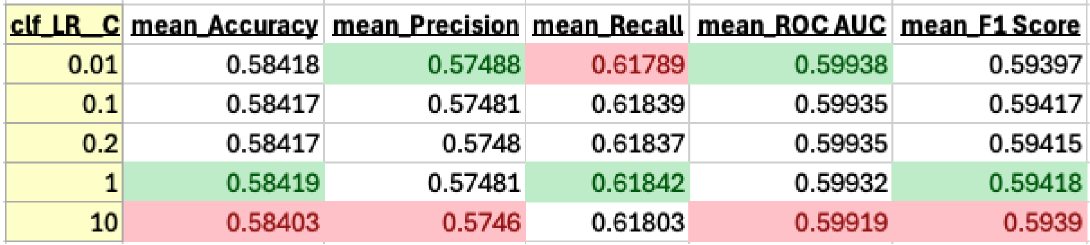
STEP-4 (b): **X_test:**
115337 rows × 37 cols
y-test:
115337 rows × 1 col
(i.e., 0: 115295, 1: 42)

STEP-5 (a): **Applying SMOTE**
X_train:
691767 rows × 37 cols
y_train: **(Balanced now)**
691767 rows × 1 col
(i.e., 0: 345886, 1: 345886)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple C values
- Multiple metrics (accuracy, recall, f1, etc.)



clf_LR__C	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.01	0.58418	0.57488	0.61789	0.59938	0.59397
0.1	0.58417	0.57481	0.61839	0.59935	0.59417
0.2	0.58417	0.5748	0.61837	0.59935	0.59415
1	0.58419	0.57481	0.61842	0.59932	0.59418
10	0.58403	0.5746	0.61803	0.59919	0.5939

Second Step:

Retrain the best model using the selected C.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC : {‘C’: 0.01}

[TEST DATA] The AUC-ROC Score: 0.6190

[TEST DATA] The accuracy Score: 0.5532

Logistic Regression – UK Pound Results

STEP-1: **Total dataset**: 12002394 rows × 11 columns

STEP-2: **Filtered UK Pound dataset**:
279255 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder** : 279255 rows × 42 columns

STEP-4: **75:25 Train:test split**

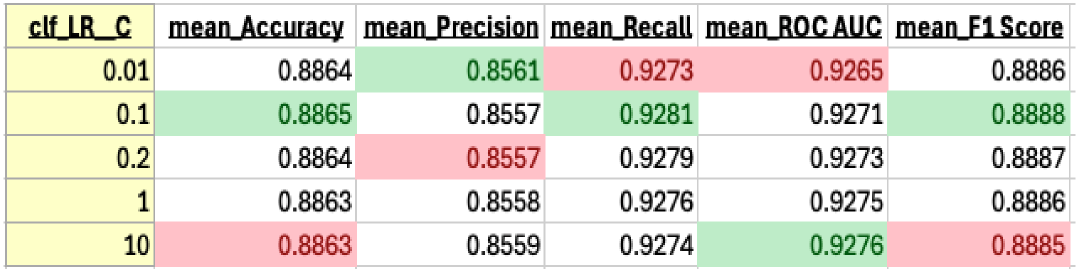
STEP-4 (a): **X_train:**
209441 rows × 41 cols
y-train: **(Imbalanced)**
209441 rows × 1 col
(i.e., 0: 209312, 1: 129)

STEP-4 (b): **X_test:**
69814 rows × 41 cols
y-test:
69814 rows × 1 col
(i.e., 0: 69771, 1: 43)

STEP-5 (a): **Applying SMOTE**
X_train:
418615 rows × 41 cols
y_train: **(Balanced now)**
418615 rows × 1 col
(i.e., 0: 209312, 1: 209303)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:
GridSearchCV on the training set only

- Multiple C values
- Multiple metrics (accuracy, recall, f1, etc.)



clf_LR_C	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.01	0.8864	0.8561	0.9273	0.9265	0.8886
0.1	0.8865	0.8557	0.9281	0.9271	0.8888
0.2	0.8864	0.8557	0.9279	0.9273	0.8887
1	0.8863	0.8558	0.9276	0.9275	0.8886
10	0.8863	0.8559	0.9274	0.9276	0.8885

Second Step:
Retrain the best model using the selected C.

Last Step:
Evaluate on X_test, y_test

The Best Parameters for AUC: {‘C’: 10}
[TEST DATA] The AUC-ROC Score: 0.8475
[TEST DATA] The accuracy Score: 0.8468

Logistic Regression – US Dollar Results

STEP-1: **Total dataset**: 12002394 rows × 11 columns

STEP-2: **Filtered USA Dollar dataset**:
300000 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder** : 300000 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
225000 rows × 41 cols
y-train: **(Imbalanced)**
225000 rows × 1 col
(i.e., 0: 224836, 1: 164)

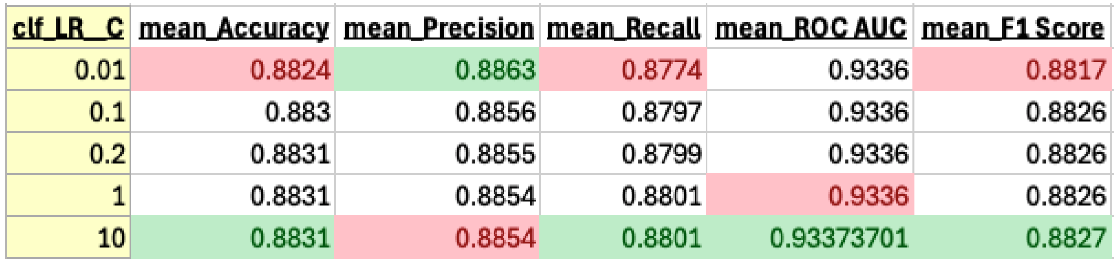
STEP-4 (b): **X_test:**
75000 rows × 41 cols
y-test:
75000 rows × 1 col
(i.e., 0: 74946, 1: 54)

STEP-5 (a): **Applying SMOTE**
X_train:
449672 rows × 41 cols
y_train: **(Balanced now)**
449672 rows × 1 col
(i.e., 0: 224836, 1: 224836)
Weights on y_train:
{0: 1.000, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple C values
- Multiple metrics (accuracy, recall, f1, etc.)



clf_LR__C	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.01	0.8824	0.8863	0.8774	0.9336	0.8817
0.1	0.883	0.8856	0.8797	0.9336	0.8826
0.2	0.8831	0.8855	0.8799	0.9336	0.8826
1	0.8831	0.8854	0.8801	0.9336	0.8826
10	0.8831	0.8854	0.8801	0.93373701	0.8827

Second Step:

Retrain the best model using the selected C.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC: {'C': 10}

[TEST DATA] The AUC-ROC Score: 0.9225
[TEST DATA] The accuracy Score: 0.8892

Random Forest – Bitcoin Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered Bitcoin dataset:**
461347 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder**: 461347 rows × 38 cols

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
346010 rows × 37 cols
y-train: <u>(Imbalanced)</u>
346010 rows × 1 col
(i.e., 0: 345886, 1: 124)

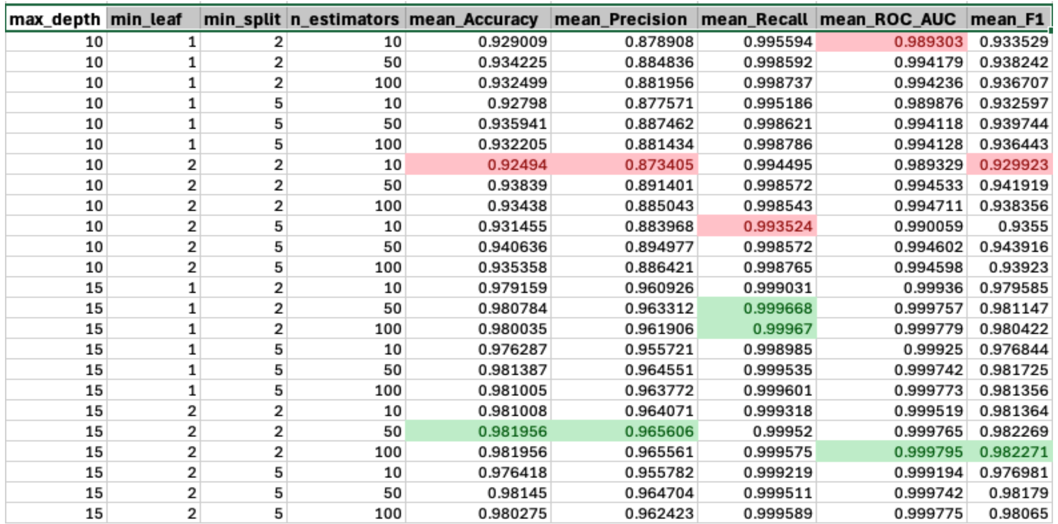
STEP-4 (b): **X_test:**
115337 rows × 37 cols
y-test:
115337 rows × 1 col
(i.e., 0: 115295, 1: 42)

STEP-5 (a): **Applying SMOTE**
X_train:
691767 rows × 37 cols
y_train: <u>(Balanced now)</u>
691767 rows × 1 col
(i.e., 0: 345886, 1: 345886)
Weights on y_train:
{0: 0.999, 1: 1.000}

<u>Initial Step:</u>

GridSearchCV on the training set only

- Multiple max_depth, min_leaf, min_split, n_estimator values
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



max_depth	min_leaf	min_split	n_estimators	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC_AUC	mean_F1
10	1	2	10	0.929009	0.878908	0.995594	0.989303	0.933529
10	1	2	50	0.934225	0.884836	0.998592	0.994179	0.938242
10	1	2	100	0.932499	0.881956	0.998737	0.994236	0.936707
10	1	5	10	0.92798	0.877571	0.995186	0.989876	0.932597
10	1	5	50	0.935941	0.887462	0.998621	0.994118	0.939744
10	1	5	100	0.932205	0.881434	0.998786	0.994128	0.936443
10	2	2	10	0.92494	0.873405	0.994495	0.989329	0.929923
10	2	2	50	0.93839	0.891401	0.998572	0.994533	0.941919
10	2	2	100	0.93438	0.885043	0.998543	0.994711	0.938356
10	2	5	10	0.931455	0.883968	0.993524	0.990059	0.9355
10	2	5	50	0.940636	0.894977	0.998572	0.994602	0.943916
10	2	5	100	0.935358	0.886421	0.998765	0.994598	0.93923
15	1	2	10	0.979159	0.960926	0.999031	0.99936	0.979585
15	1	2	50	0.980784	0.963312	0.999668	0.999757	0.981147
15	1	2	100	0.980035	0.961906	0.99967	0.999779	0.980422
15	1	5	10	0.976287	0.955721	0.998985	0.99925	0.976844
15	1	5	50	0.981387	0.964551	0.999535	0.999742	0.981725
15	1	5	100	0.981005	0.963772	0.999601	0.999773	0.981356
15	2	2	10	0.981008	0.964071	0.999318	0.999519	0.981364
15	2	2	50	0.981956	0.965606	0.99952	0.999765	0.982269
15	2	2	100	0.981956	0.965561	0.999575	0.999795	0.982271
15	2	5	10	0.976418	0.955782	0.999219	0.999194	0.976981
15	2	5	50	0.98145	0.964704	0.999511	0.999742	0.98179
15	2	5	100	0.980275	0.962423	0.999589	0.999775	0.98065

<u>Second Step</u>:

Retrain the best model using the selected best parameters.

<u>Last Step</u>:

Evaluate on X_test, y_test

The Best Parameters for AUC :

{max_depth': 15, min_samples_leaf': 2,
min_samples_split': 2, n_estimators': 100}

[TEST DATA] The AUC-ROC Score: 0.6596
[TEST DATA] The accuracy Score: 0.9565

Random Forest – UK Pound Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered UK Pound dataset:**
279255 rows × 11 columns

STEP-3: After applying **Normalization, Feature Engineering and OneHotEncoder** : 279255 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
209441 rows × 41 cols
y-train: (Imbalanced)
209441 rows × 1 col
(i.e., 0: 209312, 1: 129)

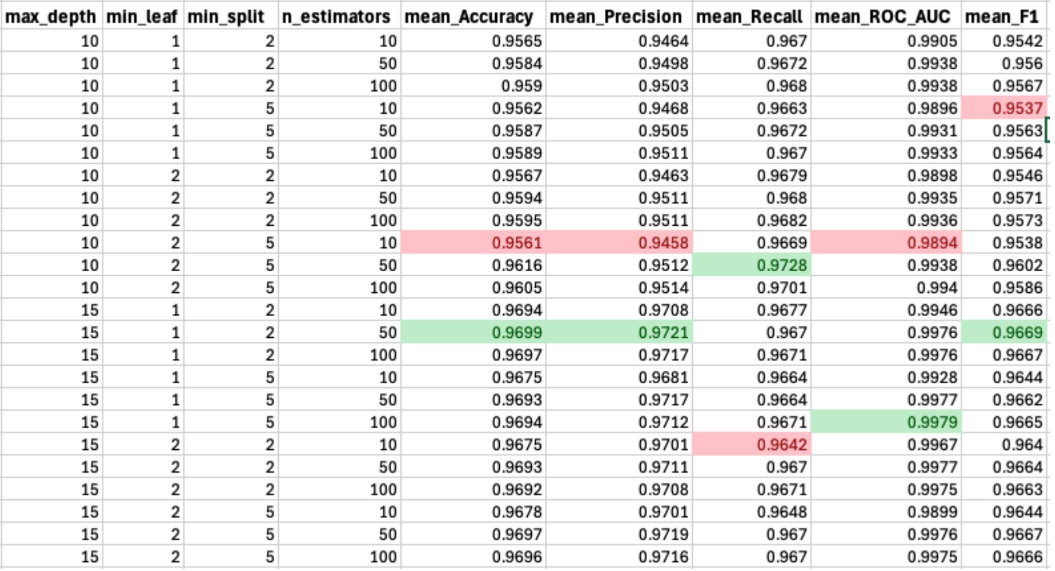
STEP-4 (b): **X_test:**
69814 rows × 41 cols
y-test:
69814 rows × 1 col
(i.e., 0: 69771, 1: 43)

STEP-5 (a): **Applying SMOTE**
X_train:
418615 rows × 41 cols
y_train: (Balanced now)
418615 rows × 1 col
(i.e., 0: 209312, 1: 209303)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple max_depth, min_leaf, min_split, n_estimator values
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



max_depth	min_leaf	min_split	n_estimators	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC_AUC	mean_F1
10	1	2	10	0.9565	0.9464	0.967	0.9905	0.9542
10	1	2	50	0.9584	0.9498	0.9672	0.9938	0.956
10	1	2	100	0.959	0.9503	0.968	0.9938	0.9567
10	1	5	10	0.9562	0.9468	0.9663	0.9896	0.9537
10	1	5	50	0.9587	0.9505	0.9672	0.9931	0.9563
10	1	5	100	0.9589	0.9511	0.967	0.9933	0.9564
10	2	2	10	0.9567	0.9463	0.9679	0.9898	0.9546
10	2	2	50	0.9594	0.9511	0.968	0.9935	0.9571
10	2	2	100	0.9595	0.9511	0.9682	0.9936	0.9573
10	2	5	10	0.9561	0.9458	0.9669	0.9894	0.9538
10	2	5	50	0.9616	0.9512	0.9728	0.9938	0.9602
10	2	5	100	0.9605	0.9514	0.9701	0.994	0.9586
15	1	2	10	0.9694	0.9708	0.9677	0.9946	0.9666
15	1	2	50	0.9699	0.9721	0.967	0.9976	0.9669
15	1	2	100	0.9697	0.9717	0.9671	0.9976	0.9667
15	1	5	10	0.9675	0.9681	0.9664	0.9928	0.9644
15	1	5	50	0.9693	0.9717	0.9664	0.9977	0.9662
15	1	5	100	0.9694	0.9712	0.9671	0.9979	0.9665
15	2	2	10	0.9675	0.9701	0.9642	0.9967	0.964
15	2	2	50	0.9693	0.9711	0.967	0.9977	0.9664
15	2	2	100	0.9692	0.9708	0.9671	0.9975	0.9663
15	2	5	10	0.9678	0.9701	0.9648	0.9899	0.9644
15	2	5	50	0.9697	0.9719	0.967	0.9976	0.9667
15	2	5	100	0.9696	0.9716	0.967	0.9975	0.9666

Second Step:

Retrain the best model using the selected best parameters.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC :
{'max_depth': 15, 'min_samples_leaf': 1,
'min_samples_split': 5, 'n_estimators': 100**}**

[TEST DATA] The AUC-ROC Score: 0.8678
[TEST DATA] The accuracy Score: 0.9733

Random Forest – USA Dollar Results

STEP-1: **Total dataset**: 12002394 rows × 11 columns

STEP-2: **Filtered USA Dollar dataset**:
300000 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder** : 300000 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
225000 rows × 41 cols
y-train: **(Imbalanced)**
225000 rows × 1 col
(i.e., 0: 224836, 1: 164)

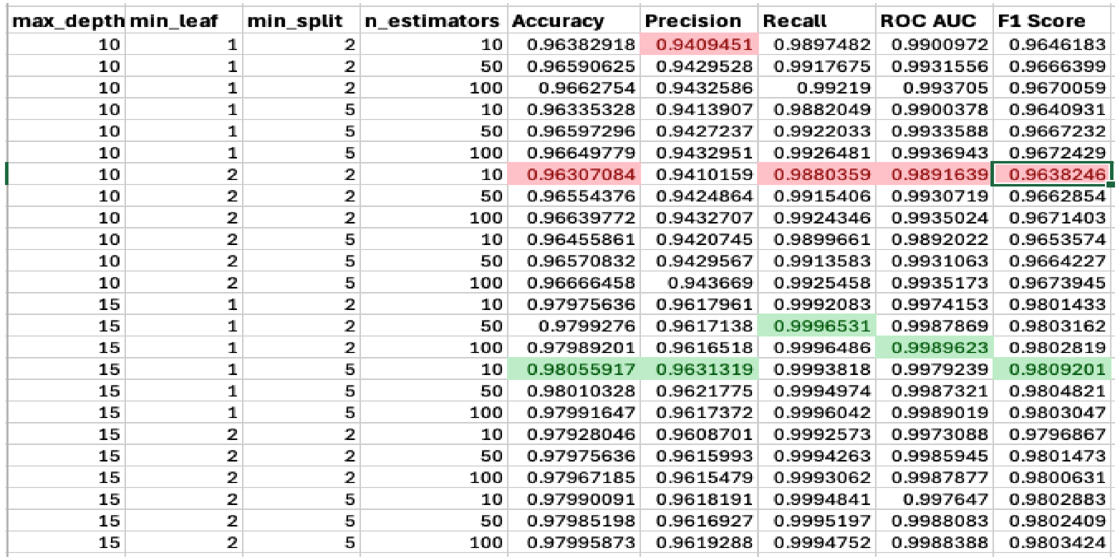
STEP-4 (b): **X_test:**
75000 rows × 41 cols
y-test:
75000 rows × 1 col
(i.e., 0: 74946, 1: 54)

STEP-5 (a): **Applying SMOTE**
X_train:
449672 rows × 41 cols
y_train: **(Balanced now)**
449672 rows × 1 col
(i.e., 0: 224836, 1: 224836)
Weights on y_train:
{0: 1.000, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple max_depth, min_leaf, min_split, n_estimator values
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



max_depth	min_leaf	min_split	n_estimators	Accuracy	Precision	Recall	ROC AUC	F1 Score
10	1	2	10	0.96382918	0.9409451	0.9897482	0.9900972	0.9646183
10	1	2	50	0.96590625	0.9429528	0.9917675	0.9931556	0.9666399
10	1	2	100	0.9662754	0.9432586	0.99219	0.993705	0.9670059
10	1	5	10	0.96335328	0.9413907	0.9882049	0.9900378	0.9640931
10	1	5	50	0.96597296	0.9427237	0.9922033	0.9933588	0.9667232
10	1	5	100	0.96649779	0.9432951	0.9926481	0.9936943	0.9672429
10	2	2	10	0.96307084	0.9410159	0.9880359	0.9891639	0.9638246
10	2	2	50	0.96554376	0.9424864	0.9915406	0.9930719	0.9662854
10	2	2	100	0.96639772	0.9432707	0.9924346	0.9935024	0.9671403
10	2	5	10	0.96455861	0.9420745	0.9899661	0.9892022	0.9653574
10	2	5	50	0.96570832	0.9429567	0.9913583	0.9931063	0.9664227
10	2	5	100	0.96666458	0.943669	0.9925458	0.9935173	0.9673945
15	1	2	10	0.97975636	0.9617961	0.9992083	0.9974153	0.9801433
15	1	2	50	0.9799276	0.9617138	0.9996531	0.9987869	0.9803162
15	1	2	100	0.97989201	0.9616518	0.9996486	0.9989623	0.9802819
15	1	5	10	0.98055917	0.9631319	0.9993818	0.9979239	0.9809201
15	1	5	50	0.98010328	0.9621775	0.9994974	0.9987321	0.9804821
15	1	5	100	0.97991647	0.9617372	0.9996042	0.9989019	0.9803047
15	2	2	10	0.97928046	0.9608701	0.9992573	0.9973088	0.9796867
15	2	2	50	0.97975636	0.9615993	0.9994263	0.9985945	0.9801473
15	2	2	100	0.97967185	0.9615479	0.9993062	0.9987877	0.9800631
15	2	5	10	0.97990091	0.9618191	0.9994841	0.997647	0.9802883
15	2	5	50	0.97985198	0.9616927	0.9995197	0.9988083	0.9802409
15	2	5	100	0.97995873	0.9619288	0.9994752	0.9988388	0.9803424

Second Step:

Retrain the best model using the selected best parameters.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC :

{'clf_RFC__max_depth': 15, 'clf_RFC__min_samples_leaf': 1,
'clf_RFC__min_samples_split': 2, 'clf_RFC__n_estimators': 100**}**

[TEST DATA] The AUC-ROC Score: 0.9231
[TEST DATA] The accuracy Score: 0.9689

Bagging (Using Naïve Bayes) – Bitcoin Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered Bitcoin dataset:**
461347 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder**: 461347 rows × 38 cols

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
346010 rows × 37 cols
y-train: __(Imbalanced)__
346010 rows × 1 col
(i.e., 0: 345886, 1: 124)

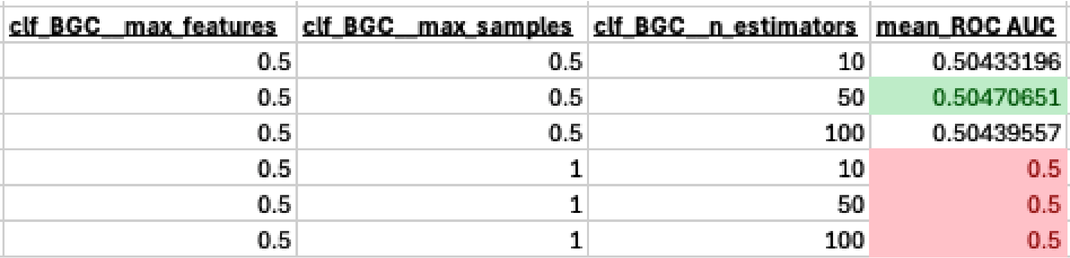
STEP-4 (b): **X_test:**
115337 rows × 37 cols
y-test:
115337 rows × 1 col
(i.e., 0: 115295, 1: 42)

STEP-5 (a): **Applying SMOTE**
X_train:
691767 rows × 37 cols
y_train: __(Balanced now)__
691767 rows × 1 col
(i.e., 0: 345886, 1: 345886)
Weights on y_train:
{0: 0.999, 1: 1.000}

__Initial Step:__

GridSearchCV on the training set only

- Multiple n_estimator, max_samples and max_feature values
- ROC scores.



clf_BGC__max_features	clf_BGC__max_samples	clf_BGC__n_estimators	mean_ROC AUC
0.5	0.5	10	0.50433196
0.5	0.5	50	0.50470651
0.5	0.5	100	0.50439557
0.5	1	10	0.5
0.5	1	50	0.5
0.5	1	100	0.5

__Second Step__:

Retrain the best model using the selected best parameters.

__Last Step__:

Evaluate on X_test, y_test

The Best Parameters for AUC :

{'clf_BGC__max_features': 0.5, 'clf_BGC__max_samples': 0.5, 'clf_BGC__n_estimators': 50}

[TEST DATA] The AUC-ROC Score: 0.5085
[TEST DATA] The accuracy Score: 0.0138

Bagging (Using Naïve Bayes) – UK Pound Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered UK Pound dataset:**
279255 rows × 11 columns

STEP-3: After applying **Normalization, Feature Engineering and OneHotEncoder** : 279255 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
209441 rows × 41 cols
y-train: **<u>(Imbalanced)</u>**
209441 rows × 1 col
(i.e., 0: 209312, 1: 129)

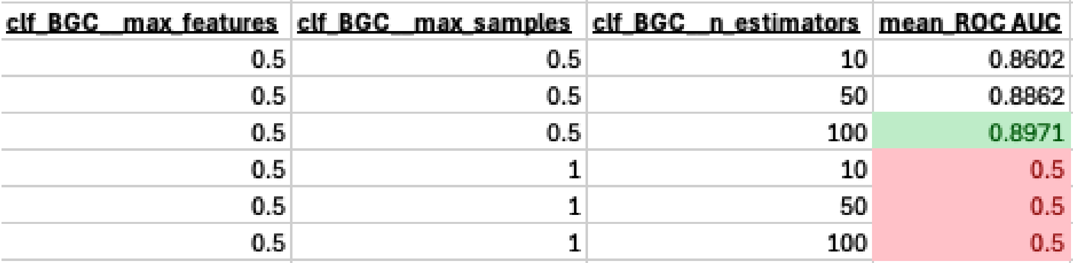
STEP-4 (b): **X_test:**
69814 rows × 41 cols
y-test:
69814 rows × 1 col
(i.e., 0: 69771, 1: 43)

STEP-5 (a): **Applying SMOTE**
X_train:
418615 rows × 41 cols
y_train: **<u>(Balanced now)</u>**
418615 rows × 1 col
(i.e., 0: 209312, 1: 209303)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple n_estimator, max_samples and max_feature values
- ROC scores.



clf_BGC__max_features	clf_BGC__max_samples	clf_BGC__n_estimators	mean_ROC AUC
0.5	0.5	10	0.8602
0.5	0.5	50	0.8862
0.5	0.5	100	0.8971
0.5	1	10	0.5
0.5	1	50	0.5
0.5	1	100	0.5

Second Step:
Retrain the best model using the selected best parameters.

Last Step:
Evaluate on X_test, y_test

The Best Parameters for AUC :
{'clf_BGC__max_features': 0.5, 'clf_BGC__max_samples': 0.5, 'clf_BGC__n_estimators': 50}

[TEST DATA] The AUC-ROC Score: 0.7893
[TEST DATA] The accuracy Score: 0.0186

Bagging (Using Naïve Bayes) – USA Dollar Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered USA Dollar dataset:**
300000 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder** : 300000 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
225000 rows × 41 cols
y-train: <u>(Imbalanced)</u>
225000 rows × 1 col
(i.e., 0: 224836, 1: 164)

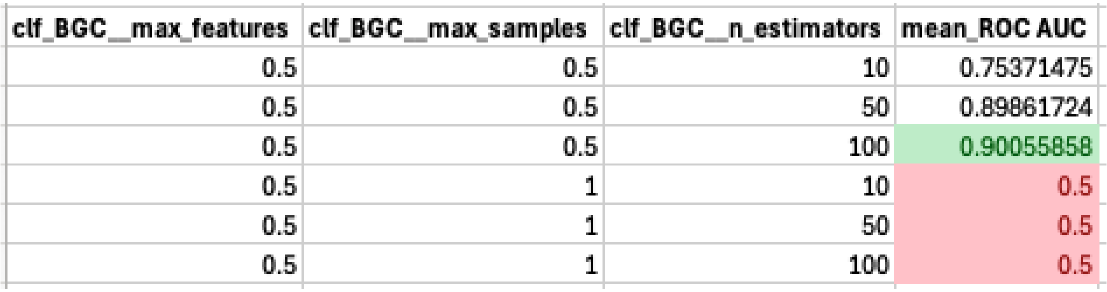
STEP-4 (b): **X_test:**
75000 rows × 41 cols
y-test:
75000 rows × 1 col
(i.e., 0: 74946, 1: 54)

STEP-5 (a): **Applying SMOTE**
X_train:
449672 rows × 41 cols
y_train: **<u>(Balanced now)</u>**
449672 rows × 1 col
(i.e., 0: 224836, 1: 224836)
Weights on y_train:
{0: 1.000, 1: 1.000}

<u>Initial Step:</u>

GridSearchCV on the training set only

- Multiple n_estimator, max_samples and max_feature values
- ROC scores.



clf_BGC__max_features	clf_BGC__max_samples	clf_BGC__n_estimators	mean_ROC AUC
0.5	0.5	10	0.75371475
0.5	0.5	50	0.89861724
0.5	0.5	100	0.90055858
0.5	1	10	0.5
0.5	1	50	0.5
0.5	1	100	0.5

<u>Second Step</u>:
Retrain the best model using the selected best parameters.

<u>Last Step</u>:
Evaluate on X_test, y_test

The Best Parameters for AUC :
{'clf_BGC__max_features': 0.5, 'clf_BGC__max_samples': 0.5, 'clf_BGC__n_estimators’: 100}

[TEST DATA] The AUC-ROC Score: 0.8860
[TEST DATA] The accuracy Score: 0.0222

Gradient Boosting Classifier – Bitcoin Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered Bitcoin dataset:**
461347 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder**: 461347 rows × 38 cols

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
346010 rows × 37 cols
y-train: **(Imbalanced)**
346010 rows × 1 col
(i.e., 0: 345886, 1: 124)

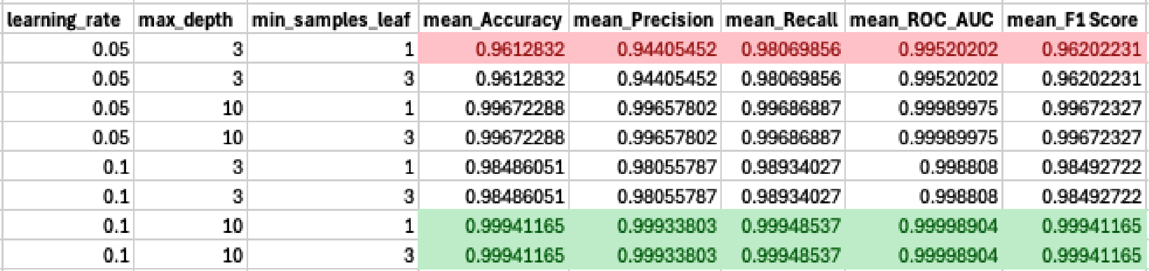
STEP-4 (b): **X_test:**
115337 rows × 37 cols
y-test:
115337 rows × 1 col
(i.e., 0: 115295, 1: 42)

STEP-5 (a): **Applying SMOTE**
X_train:
691767 rows × 37 cols
y_train: **(Balanced now)**
691767 rows × 1 col
(i.e., 0: 345886, 1: 345886)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple learning_rate, max_depth, and min_samples_leaf values.
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



learning_rate	max_depth	min_samples_leaf	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC_AUC	mean_F1 Score
0.05	3	1	0.9612832	0.94405452	0.98069856	0.99520202	0.96202231
0.05	3	3	0.9612832	0.94405452	0.98069856	0.99520202	0.96202231
0.05	10	1	0.99672288	0.99657802	0.99686887	0.99989975	0.99672327
0.05	10	3	0.99672288	0.99657802	0.99686887	0.99989975	0.99672327
0.1	3	1	0.98486051	0.98055787	0.98934027	0.998808	0.98492722
0.1	3	3	0.98486051	0.98055787	0.98934027	0.998808	0.98492722
0.1	10	1	0.99941165	0.99933803	0.99948537	0.99998904	0.99941165
0.1	10	3	0.99941165	0.99933803	0.99948537	0.99998904	0.99941165

Second Step:

Retrain the best model using the selected best parameters.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC :

{'clf_GBC__learning_rate': 0.1, 'clf_GBC__max_depth': 10, 'clf_GBC__min_samples_leaf': 1}

[TEST DATA] The AUC-ROC Score: 0.7544
[TEST DATA] The accuracy Score: 0.9991

Gradient Boosting Classifier – UK Pound Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered UK Pound dataset:**
279255 rows × 11 columns

STEP-3: After applying **Normalization, Feature Engineering and OneHotEncoder** : 279255 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
209441 rows × 41 cols
y-train: **(Imbalanced)**
209441 rows × 1 col
(i.e., 0: 209312, 1: 129)

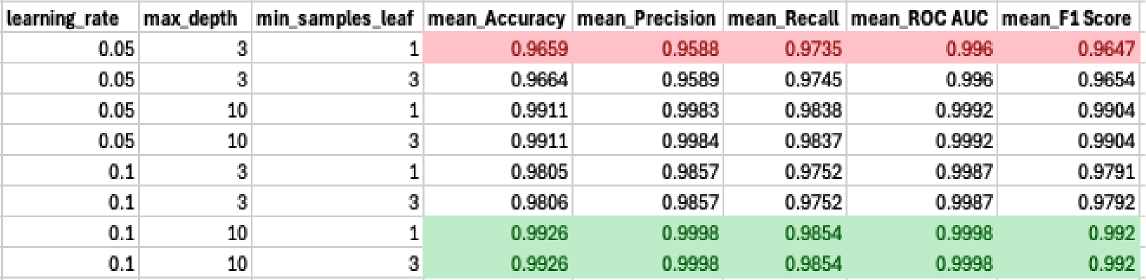
STEP-4 (b): **X_test:**
69814 rows × 41 cols
y-test:
69814 rows × 1 col
(i.e., 0: 69771, 1: 43)

STEP-5 (a): **Applying SMOTE**
X_train:
418615 rows × 41 cols
y_train: **(Balanced now)**
418615 rows × 1 col
(i.e., 0: 209312, 1: 209303)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple learning_rate, max_depth, and min_samples_leaf values.
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



learning_rate	max_depth	min_samples_leaf	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.05	3	1	0.9659	0.9588	0.9735	0.996	0.9647
0.05	3	3	0.9664	0.9589	0.9745	0.996	0.9654
0.05	10	1	0.9911	0.9983	0.9838	0.9992	0.9904
0.05	10	3	0.9911	0.9984	0.9837	0.9992	0.9904
0.1	3	1	0.9805	0.9857	0.9752	0.9987	0.9791
0.1	3	3	0.9806	0.9857	0.9752	0.9987	0.9792
0.1	10	1	0.9926	0.9998	0.9854	0.9998	0.992
0.1	10	3	0.9926	0.9998	0.9854	0.9998	0.992

Second Step:

Retrain the best model using the selected best parameters.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC :

{'clf_BGC__max_features': 0.1, 'clf_BGC__max_samples’: 10, 'clf_BGC__n_estimators’: 1}

[TEST DATA] The AUC-ROC Score: 0.8712
[TEST DATA] The accuracy Score: 0.9993

Gradient Boosting Classifier – USA Dollar Results

STEP-1: **Total dataset**: 12002394 rows × 11 columns

STEP-2: **Filtered USA Dollar dataset**:
300000 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder** : 300000 rows × 42 columns

STEP-4: **75:25 Train:test split**

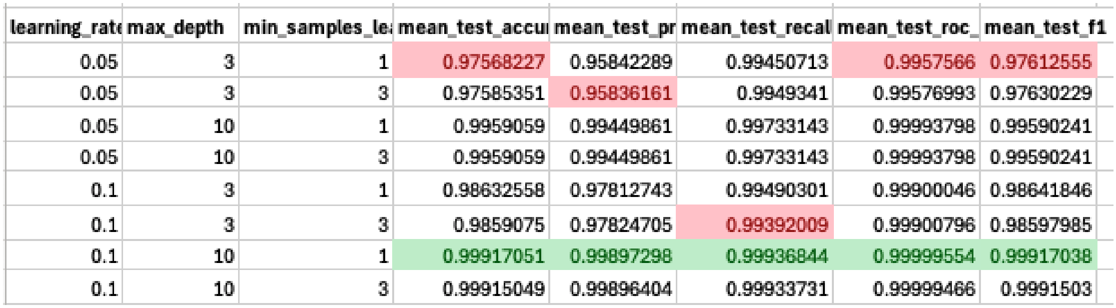
STEP-4 (a): **X_train:**
225000 rows × 41 cols
y-train: **(Imbalanced)**
225000 rows × 1 col
(i.e., 0: 224836, 1: 164)

STEP-4 (b): **X_test:**
75000 rows × 41 cols
y-test:
75000 rows × 1 col
(i.e., 0: 74946, 1: 54)

STEP-5 (a): **Applying SMOTE**
X_train:
449672 rows × 41 cols
y_train: **(Balanced now)**
449672 rows × 1 col
(i.e., 0: 224836, 1: 224836)
Weights on y_train:
{0: 1.000, 1: 1.000}

Initial Step:
GridSearchCV on the training set only

- Multiple learning_rate, max_depth, and min_samples_leaf values.
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



learning_rat	max_depth	min_samples_le	mean_test_accu	mean_test_pr	mean_test_recal	mean_test_roc_	mean_test_f1
0.05	3	1	0.97568227	0.95842289	0.99450713	0.9957566	0.97612555
0.05	3	3	0.97585351	0.95836161	0.9949341	0.99576993	0.97630229
0.05	10	1	0.9959059	0.99449861	0.99733143	0.99993798	0.99590241
0.05	10	3	0.9959059	0.99449861	0.99733143	0.99993798	0.99590241
0.1	3	1	0.98632558	0.97812743	0.99490301	0.99900046	0.98641846
0.1	3	3	0.9859075	0.97824705	0.99392009	0.99900796	0.98597985
0.1	10	1	0.99917051	0.99897298	0.99936844	0.99999554	0.99917038
0.1	10	3	0.99915049	0.99896404	0.99933731	0.99999466	0.9991503

Second Step:
Retrain the best model using the selected best parameters.

Last Step:
Evaluate on X_test, y_test

The Best Parameters for AUC :
{'clf_BGC__max_features': 0.1, 'clf_BGC__max_samples': 10, 'clf_BGC__n_estimators': 1}

[TEST DATA] The AUC-ROC Score: 0.8712
[TEST DATA] The accuracy Score: 0.9993

SVC – Bitcoin Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered Bitcoin dataset:**
461347 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder**: 461347 rows × 38 cols

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
346010 rows × 37 cols
y-train: **(Imbalanced)**
346010 rows × 1 col
(i.e., 0: 345886, 1: 124)

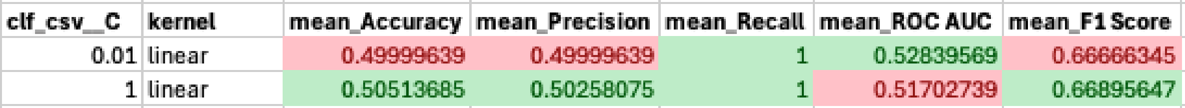
STEP-4 (b): **X_test:**
115337 rows × 37 cols
y-test:
115337 rows × 1 col
(i.e., 0: 115295, 1: 42)

STEP-5 (a): **Applying SMOTE**
X_train:
691767 rows × 37 cols
y_train: **(Balanced now)**
691767 rows × 1 col
(i.e., 0: 345886, 1: 345886)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple learning_rate, max_depth, and min_samples_leaf values.
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



clf_csv__C	kernel	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.01	linear	0.49999639	0.49999639	1	0.52839569	0.66666345
1	linear	0.50513685	0.50258075	1	0.51702739	0.66895647

Second Step:

Retrain the best model using the selected best parameters.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC :

{'clf_csv__C': 1, 'clf_csv__kernel': 'linear'}

[TEST DATA] The AUC-ROC Score: 0.5000
[TEST DATA] The accuracy Score: 0.0004

SVC – UK Pound Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered UK Pound dataset:**
279255 rows × 11 columns

STEP-3: After applying **Normalization, Feature Engineering and OneHotEncoder** : 279255 rows × 42 columns

STEP-4: **75:25 Train:test split**

STEP-4 (a): **X_train:**
209441 rows × 41 cols
y-train: (Imbalanced)
209441 rows × 1 col
(i.e., 0: 209312, 1: 129)

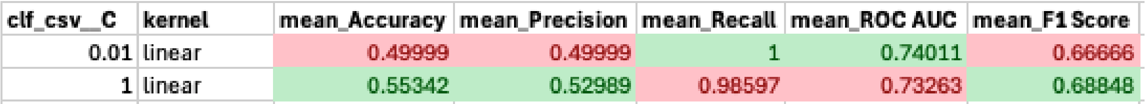
STEP-4 (b): **X_test:**
69814 rows × 41 cols
y-test:
69814 rows × 1 col
(i.e., 0: 69771, 1: 43)

STEP-5 (a): **Applying SMOTE**
X_train:
418615 rows × 41 cols
y_train: (Balanced now)
418615 rows × 1 col
(i.e., 0: 209312, 1: 209303)
Weights on y_train:
{0: 0.999, 1: 1.000}

Initial Step:

GridSearchCV on the training set only

- Multiple learning_rate, max_depth, and min_samples_leaf values.
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



clf_csv__C	kernel	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.01	linear	0.49999	0.49999	1	0.74011	0.66666
1	linear	0.55342	0.52989	0.98597	0.73263	0.68848

Second Step:

Retrain the best model using the selected best parameters.

Last Step:

Evaluate on X_test, y_test

The Best Parameters for AUC :
{'clf_csv__C': 1, 'clf_csv__kernel': 'linear'}

[TEST DATA] The AUC-ROC Score: 0.7153
[TEST DATA] The accuracy Score: 0.0396

SVC – USA Dollar Results

STEP-1: **Total dataset:** 12002394 rows × 11 columns

STEP-2: **Filtered USA Dollar dataset:**
300000 rows × 11 columns

STEP-3: Applying **Normalization, Feature Engineering and OneHotEncoder** : 300000 rows × 42 columns

STEP-4: **75:25 Train:test split**

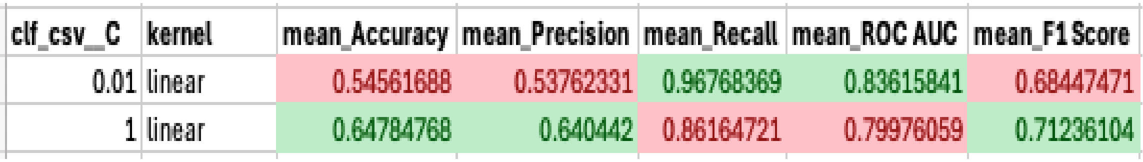
STEP-4 (a): **X_train:**
225000 rows × 41 cols
y-train: <u>(Imbalanced)</u>
225000 rows × 1 col
(i.e., 0: 224836, 1: 164)

STEP-4 (b): **X_test:**
75000 rows × 41 cols
y-test:
75000 rows × 1 col
(i.e., 0: 74946, 1: 54)

STEP-5 (a): **Applying SMOTE**
X_train:
449672 rows × 41 cols
y_train: <u>(Balanced now)</u>
449672 rows × 1 col
(i.e., 0: 224836, 1: 224836)
Weights on y_train:
{0: 1.000, 1: 1.000}

<u>Initial Step:</u>
GridSearchCV on the training set only

- Multiple learning_rate, max_depth, and min_samples_leaf values.
- Multiple metrics (accuracy, precision, recall, ROC_AUC, f1, etc.)



clf_csv__C	kernel	mean_Accuracy	mean_Precision	mean_Recall	mean_ROC AUC	mean_F1 Score
0.01	linear	0.54561688	0.53762331	0.96768369	0.83615841	0.68447471
1	linear	0.64784768	0.640442	0.86164721	0.79976059	0.71236104

<u>Second Step</u>:
Retrain the best model using the selected best parameters.

<u>Last Step</u>:
Evaluate on X_test, y_test

The Best Parameters for AUC :
{'clf_csv__C': 0.01, 'clf_csv__kernel': 'linear'}

[TEST DATA] The AUC-ROC Score: 0.8491
[TEST DATA] The accuracy Score: 0.7123

Unsupervised Learning

1

To detect anomalies without labeled data (i.e., that set off from predictable trends.)

2

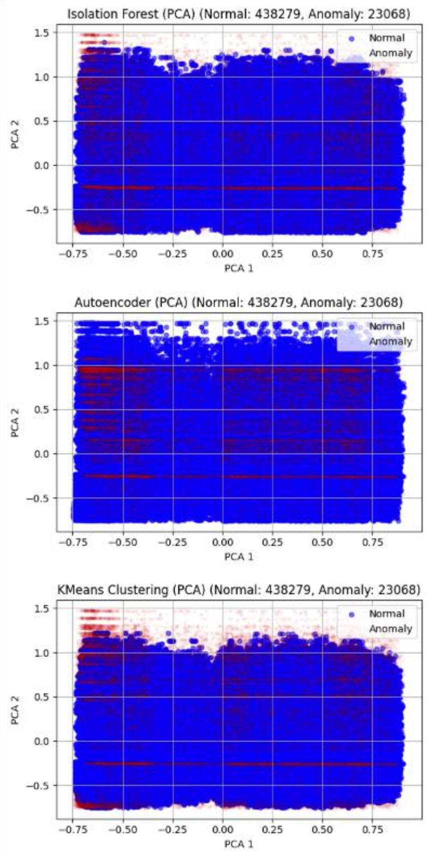
Isolation Forest, One-Class SVM, Autoencoders, and possible clustering.

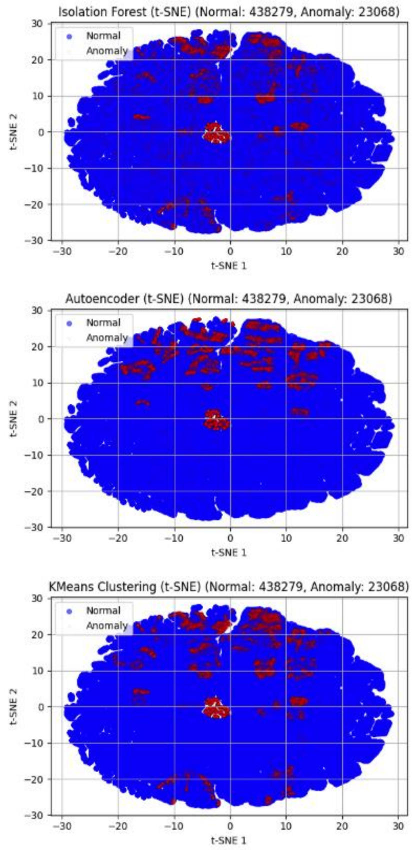
3

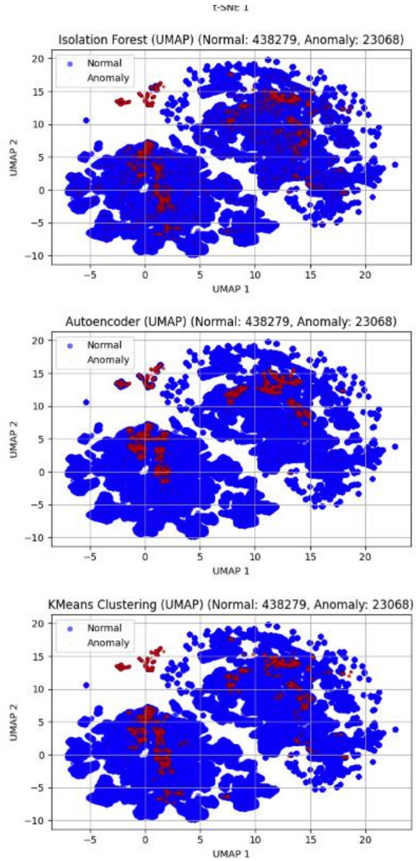
Evaluation: How most identified anomalies that set off from predictable trends.

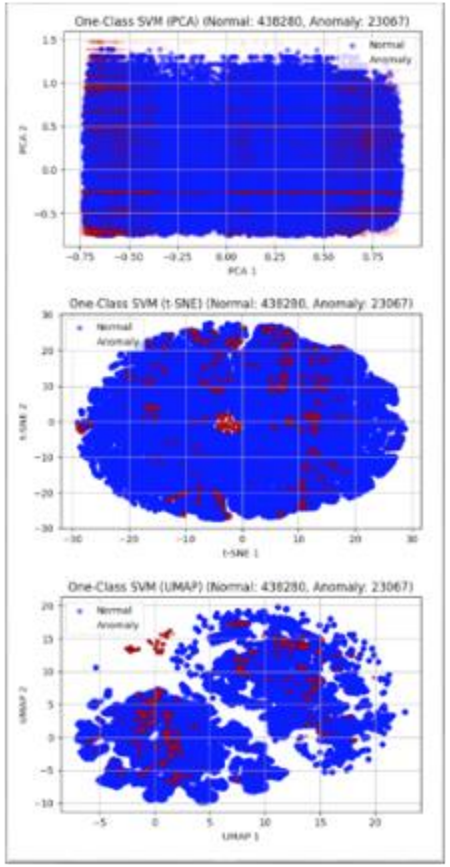
Isolation Forest, Autoencoders, K-means Clustering Results

Bitcoin (Normal: 438279, Anomaly: 23068)



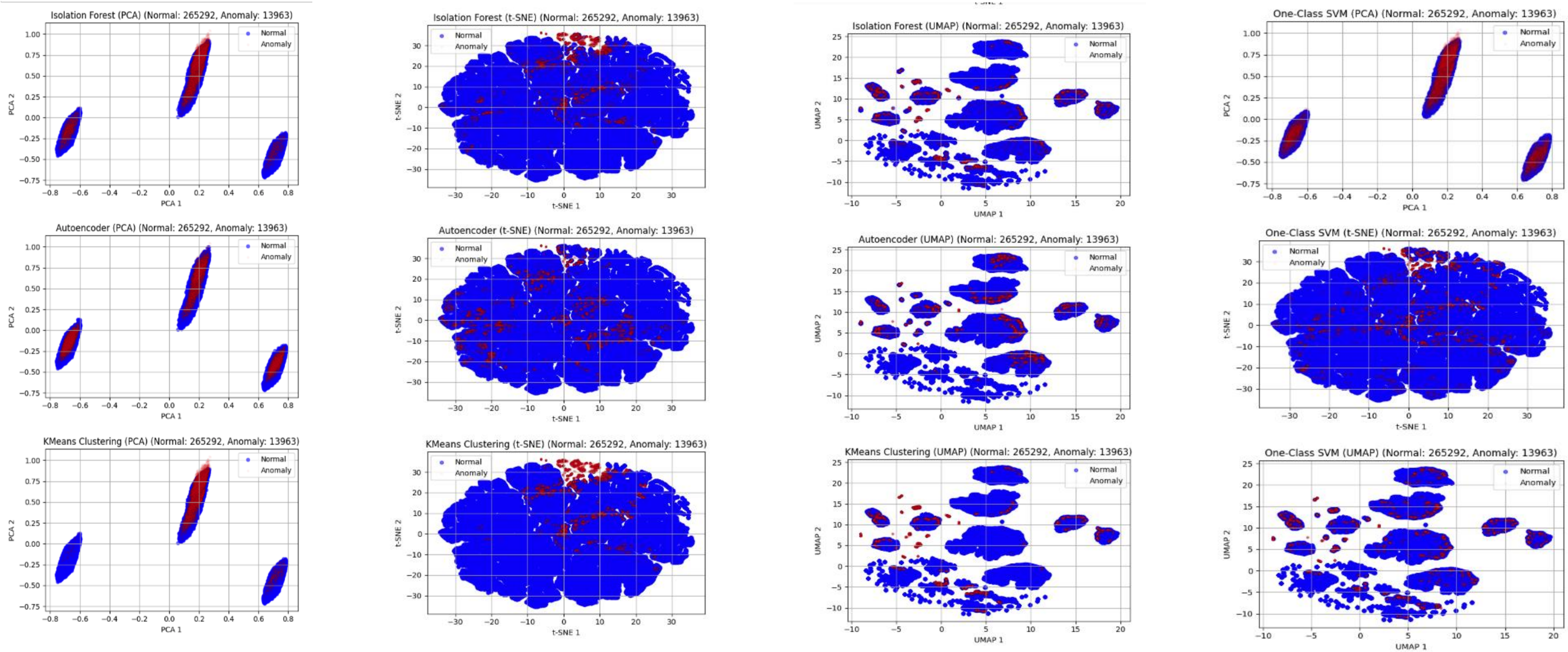


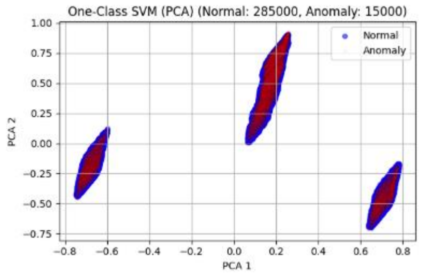


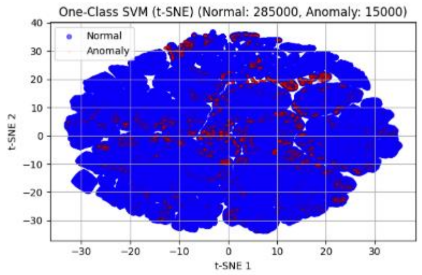


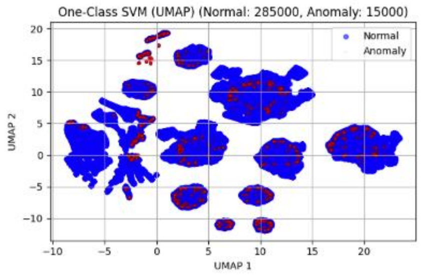
Isolation Forest, Autoencoders, K-means Clustering Results

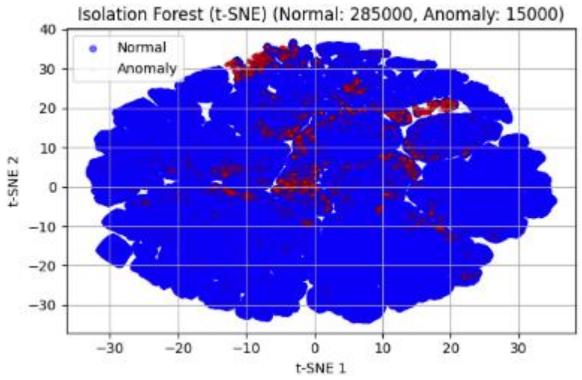
UK Pound (Normal: 265292, Anomaly: 13963)

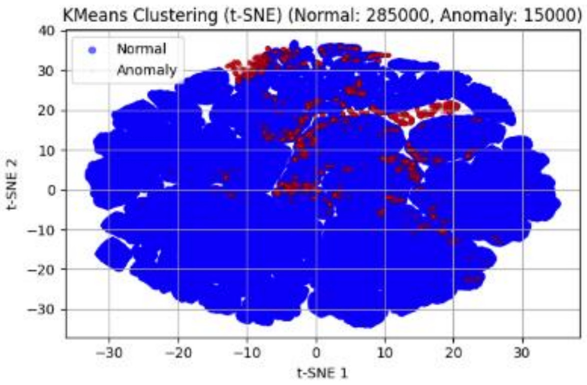


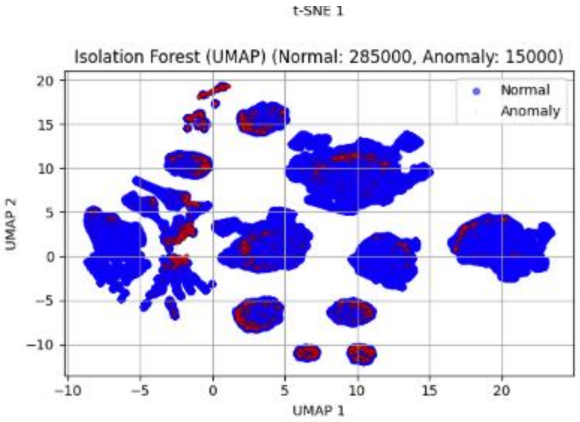


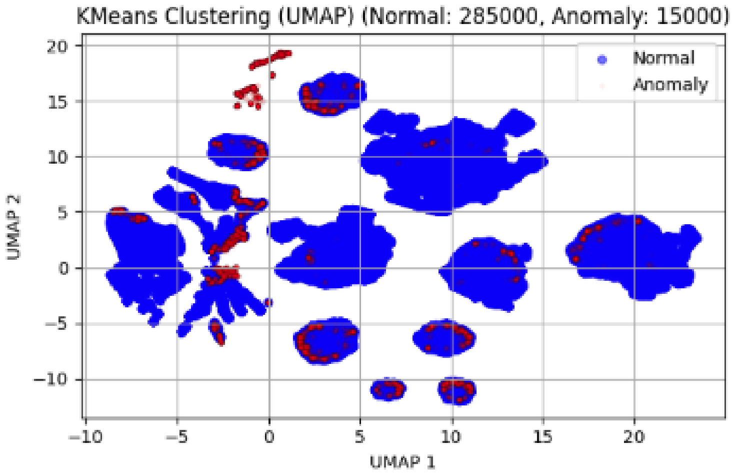












Isolation Forest, Autoencoders, K-means Clustering Results

US Dollar (Normal: 285000, Anomaly: 15000)

Future work – Most promising if huge dataset.

- **Recurrent Neural Networks (RNN) / LSTM (Long-Short term memory)**
  - **Objective:** Classify if transaction is fraud (1) or non-fraud (0).
    - STEP-1: Group by from bank, account number and investigate values by hourly/daily.
    - STEP-2: Normalized and OneHotEncoding into dataset.
    - STEP-3: Rolling window into 2 cols -- to sum up last last hour and 24 hours transaction or past N number of transactions (i.e., last N=25 transactions).
    - STEP-4: Split the dataset into n:100-n train:test split.
    - STEP-5: build an RNN model
    - STEP-6: Train on sequence to predict if the next transaction is fraud using the Long-term memory and feeding short-term memory.
    - STEP-7: RNN: Updated after each transaction, and having problem with long-term memory so we need to use LSTM in that case.
    - STEP-8: LSTM use gates (Forget → Input → output gates) for pattern input.
    - STEP-9: So, we can get if the next transaction is fraud or not.