Fraud Detection on Financial Data

Busra Gokmen
Computer Engineering
Marmara University
Istanbul, Turkey
Busragokmen67@gmail.com

Caner Dagdas
Electrical and Electronics Engineering
Marmara University
Istanbul, Turkey
Canerdagdas@hotmail.com

Ceyhun Vardar
Industrial Engineering
Marmara University
Istanbul, Turkey
Vardarceyhun13@gmail.com

Cem Gulec
Computer Engineering
Marmara University
Istanbul, Turkey
cem.ggulecc@gmail.com

Omer Faruk Caki
Computer Engineering
Marmara University
Istanbul, Turkey
omerfarukcaki@gmail.com

Abstract : In this project, we investigate the application of machine learning techniques to detect fraud activities in mobile payment transactions. The predictions are based on synthetic data provided from Edgar Lopez's Paysim: a financial mobile money simulator for fraud detection article.The data consist of several qualities and features related to the banking industry. Different techniques like multiple linear regression analysis, k nearest neighbours, naïve bayes and decision trees have been used to make the predictions. The predictions are then evaluated and compared in order to find those which provide the best performances. Our results show which method performed better in order to detect fraud activities in mobile payments.

Keywords : Fraud detection, Machine learning algorithms, Deep learning algorithms, predictive analysis

1. Introduction

The Association of Certified Fraud Examiners (ACFE) defines "fraud" as: the use of one's occupation for personal enrichment through the deliberate misuse or misapplication of the employing organization's resources or assets [5].

The basic motivation for committing fraud is to make money on false ground through illegal ways. This has far-reaching implications for the economy, the law, and even human moral standards[1].

With the advancement of modern technology, fraud is on the rise, resulting in billions of dollars being lost each year throughout the world[2].

Traditional fraud detection tools are unable to detect complex fraud techniques. Limiting oneself to an examination of cardholder behavior or to static fraud risk management guidelines has never deterred criminals from committing their crimes[6].In the real world, a highly precise system for detecting mobile payment fraud is required, as financial fraud results in financial loss[4].

The data used in this project was downloaded from Kaggle. It was uploaded to Kaggle by Edgar Lopez Rojas, a Kaggle.com user.

2. Related Work

There are many works related to our topic. Many related works can be found under a headline of fraud detection.

In the work of Dahee Choi and Kyungho Lee, they have used various clustering algorithms such as EM, K-Means, Farthestfirst, Xmeans, Densitybased and classification algorithms such as NaiveBayes, SVM, Logistic, and DecisionTree. They have used F1 score and ROC curve as evaluation metrics. Logistic regression performed slightly better than other classification algorithms and Farthestfirst clustering algorithm performed better than other clustering algorithms[4].

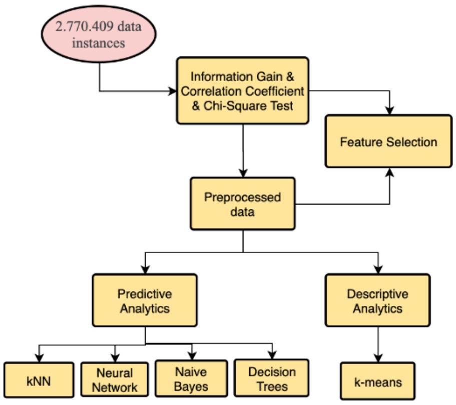
In the work of Imane Sadgali, Nawal Sael, and Fouzia Benabbou, they have used four different types of supervised learning techniques such as Decision Tree, Support Vector Machine, Random Forest, and K-Nearest Neighbor on Credit Card Transaction Data. They have used Accuracy and MSE as evaluation metrics. Support Vector Machine performed better than the other 3 algorithms[6].

According to Francis Effirim Botchey, Zhen Qin, and Kwesi Hughes Lartey's work, Gradient Boosted Decision Tree almost performed perfectly according to accuracy, precision, recall, and F1 scores with the same dataset as we used. Also Bernoulli Naive Bayes did perform well in the same dataset too. They compared Support Vector Machines, Gradient Boosted Decision Tree, and Naive Bayes algorithms in their paper[3].

3. Approach

Initially, our paysim synthetic data set consisted of 6,362,622 instances. We find that of the five types of transactions, fraud occurs only in two of them: 'transfer' where money is sent to a customer / fraudster and 'cash_out' where money is sent to a merchant who pays the customer / fraudster in cash. Remarkably, the number of fraudulent transfers almost equals the number of fraudulent cash_outs. So, fraud is committed by first transferring out funds to another account which subsequently cashes it out. The number of fraudulent transfers is 4097 and The number of fraudulent cash_outs is 4116. From this analysis, we know that fraud only occurs in ‘transfer' and 'cash_out's. Therefore, we only combine data of type fraudulent transformers and fraudulent cash out for analysis. As a result, we have a total of 2.770.409 data in our final dataset. Of these data, 8213 are fraud instances and 2,762,196 are non-fraud instances.

Diagram 1: Flow Diagram of Processes



3.1. Feature Selection

Considering the learning of models, not all variables in the data set are useful for building a model. Adding unnecessary variables reduces the model's ability to generalize and can also reduce the overall accuracy of a classifier. Also, adding more variables to a model increases the overall complexity of the model. To prevent this, we used various feature selection methods and features that only the model can learn in an optimized way.

3.1.1 Information Gain Method

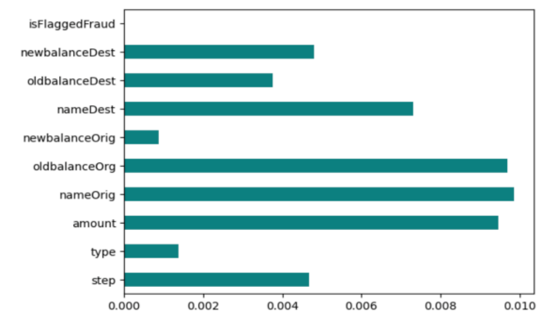
Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.

Information Gain, or IG for short, measures the reduction or surprise in entropy by dividing a data set according to a particular value of a random variable. A larger information gain indicates a lower group of entropy or groups of samples and is therefore less surprising. You may remember that information measures how surprising an event is in bits. Events with lower probability have more information, higher probability events have less information. Entropy measures how much information is in a random variable, or more specifically, its probability distribution. A curved distribution has a low entropy, while a distribution in which events have an equal probability has a greater entropy. We can think of the entropy of a dataset in terms of the probability distribution of observations in a dataset belonging to one class or another, eg. Two classes in case of binary classification dataset. For example, in a binary classification problem (two classes), we can calculate the entropy of the data sample as follows:

$$\text{Entropy} = -(p(0) * \log(P(0)) + p(1) * \log(P(1)))$$

After the implementation of the information gain feature selection method in our data set, we obtained the following importances distribution plot:

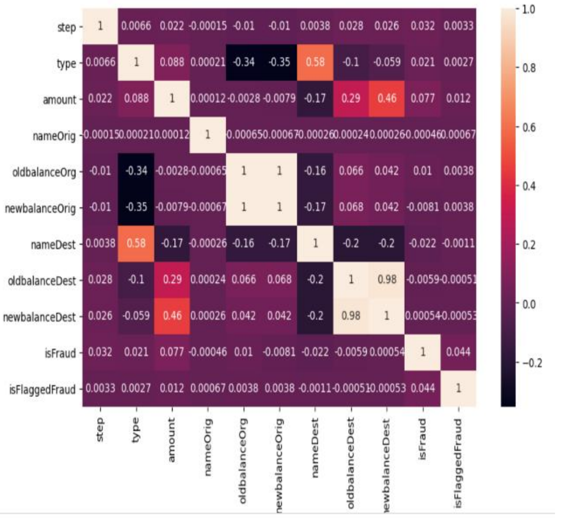
Graph 1: Information Gain Importance Table



3.1.2 Correlation Coefficient Method

Correlation is a measure of the linear relationship of 2 or more variables. Through correlation, we can predict one variable from the other. The logic behind using correlation for feature selection is that the good variables are highly correlated with the target. Furthermore, variables should be correlated with the target but should be uncorrelated among themselves. After the implementation of the correlation coefficient selection method in our data set, we obtained the following correlation distribution plot:

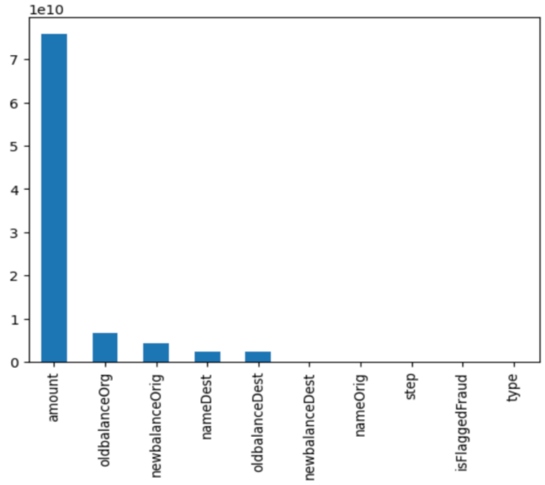
Graph 2: Correlation Coefficient Importance Table



3.1.3 Chi-square Test Method

Chi-square test is used for categorical features in a data set. We calculate the Chi-square between each feature and target and select the desired number of features with the best Chi-square scores. To correctly apply the chi-square to test the relationship between the various features in the dataset and the target variable, the following conditions must be met: variables must be categorical, sampled independently, and values must have an expected frequency. After the implementation of the correlation coefficient selection method in our data set, we obtained the following importance distribution plot:

Graph 3: Chi-Square Test Importance Table



3.2. Classification Models

In this section, we provide information about the models we use for classification with stratified 10-fold cross validation methods.

3.2.1 KNN

The working principle behind KNN is that all example dataset n (feature number) is placed in a size field and given an unknown instance, the algorithm determines the Euclidean distance between the instance and the k nearest instance. Different models with variable k values have been conducted and trained for further classification. The reason behind using KNN is that its structure is very simple. It is easy to use and is a good model for linearly separated classes.

3.2.2 Naive Bayes

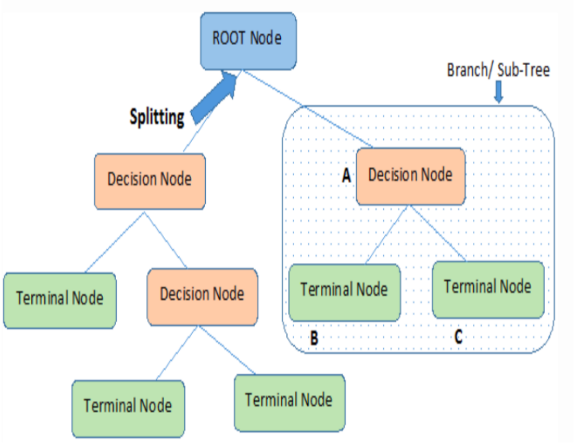
Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that a given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. Naive Bayes classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature. The reason behind using Naive Bayes algorithm is a fast, highly scalable algorithm. Also it can be used for binary classification.

3.2.3 Decision Tree

The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Diagram 2: Decision Tree Structure



3.2.4 Neural Network(Multi Layer Perceptron)

Neural networks are multilayered networks of neurons that are modeled as an imitation of how the brain works. Back propagation neural networks consist of an input neuron layer, which is the layer that takes input features that are hidden neuron layers and an output layer has a number of neurons equal to the number of each diagnostic category, in our classification task that is two. Every link neuron has weights, and these weights were fixed in the training phase by finding its gradient. The backpropagation algorithm used to find the local minimum of the error function. Our neural networks model for classification consists of 100 epochs with a batch size of 512. We used ADAM optimizer with 0.0001 learning rate. Relu is used as an activation function and output activation function is softmax.

The reason we use neural networks is, deep learning model can solve complex relationships between features and we have large data to obtain information.

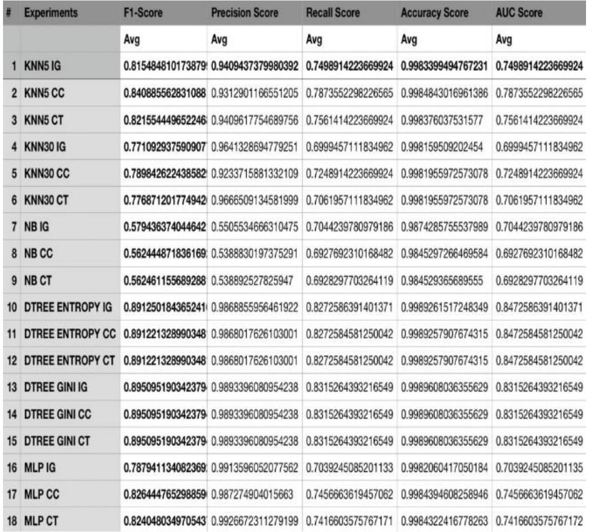
4. Experimental Setup

We dropped duplicate instances from the data we obtained based on different feature selection methods. We converted categorical data into numerical values using ordinal and one-hot encoding. Also we selected stratified 10-fold cross validation to split data to train and test the classification models mainly because we have an imbalanced data. With stratified 10-fold cross validation, every sample is used for training and testing thus generally returning more accurate results when compared to other statistical methods. We have chosen recall, precision, accuracy, f1 and AUC scores as our evaluation metrics. For each metric we have taken the selected class as the positive label, separating evaluation metrics between classes because of our imbalanced dataset. Also because of this imbalance, precision, recall and AUC are very useful as a measure of success between models. Recall score refers to the total number of positive samples that are correctly classified and precision refers to the percentage of the samples classified in the positive class that are correct. The f1 score is the weighted average of precision and recall. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

5. Experimental Results and Discussion

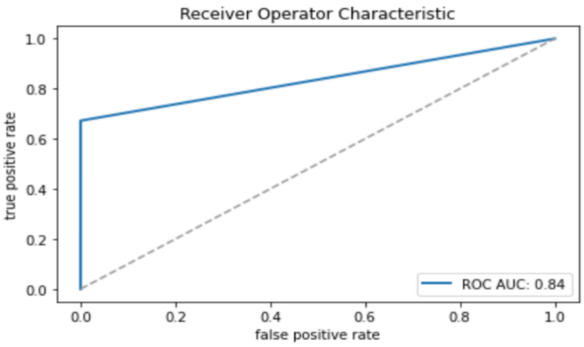
We trained 6 different models and we used information gain, correlation coefficient, chi-square test feature selection methods. We run our 18 different classifiers and run predictions on corresponding test-sets. And finally we evaluate models by using 5 metrics(Accuracy, P, R, F1, AUC) we determined above. In total, we run 18 different experiments in 10 folds. We used KNeighborsClassifier model with 5 neighbours, KNeighborsClassifier model with 30 neighbours, DecisionTreeClassifier model with gini, DecisionTreeClassifier model with entropy, GaussianNB model and MLP deep learning model of Sklearn. Following table shows the AUC, precision accuracy, recall and F1 evaluation metrics of the kNN, Naive Bayes, MLP and Decision Tree classifier models constructed for predictive analytics. Generally, the metrics of the models turned out to be high values. We performed a t-test using all evaluation metrics of the two best models. Our best model was the DecisionTreeClassifier model with entropy. This model has an average of 89% F1 score, 98% precision score, 82% recall score, 99% accuracy score and 84% AUC score. Our second model is the DecisionTreeClassifier model with gini. This model also has an average of 89% F1 score, 98% precision score, 83% recall score, 99% accuracy score and 83% AUC score.

Table 1: Results of Evaluation Metrics



#	Experiments	F1-Score	Precision Score	Recall Score	Accuracy Score	AUC Score
		Avg	Avg	Avg	Avg	Avg
1	KNN5 IG	0.815484810173879	0.9409437379980392	0.7498914223669924	0.9983399494767231	0.7498914223669924
2	KNN5 CC	0.840885562831088	0.9312901166551205	0.7873552298226565	0.9984843016961386	0.7873552298226565
3	KNN5 CT	0.821554449652246	0.9409617754689756	0.7561414223669924	0.998376037531577	0.7561414223669924
4	KNN30 IG	0.771092937590907	0.9641328694779251	0.6999457111834962	0.998159509202454	0.6999457111834962
5	KNN30 CC	0.789842622438582	0.9233715881332109	0.7248914223669924	0.9981955972573078	0.7248914223669924
6	KNN30 CT	0.776871201774942	0.9666509134581999	0.7061957111834962	0.9981955972573078	0.7061957111834962
7	NB IG	0.579436374044642	0.5505534666310475	0.7044239780979186	0.9874285755537989	0.7044239780979186
8	NB CC	0.562444871836169	0.5388830197375291	0.6927692310168482	0.9845297266469584	0.6927692310168482
9	NB CT	0.562461155689288	0.538892527825947	0.6928297703264119	0.984529365689555	0.6928297703264119
10	DTREE ENTROPY IG	0.891250184365241	0.9868855956461922	0.8272586391401371	0.9989261517248349	0.8472586391401371
11	DTREE ENTROPY CC	0.891221328990348	0.9868017626103001	0.8272584581250042	0.9989257907674315	0.8472584581250042
12	DTREE ENTROPY CT	0.891221328990348	0.9868017626103001	0.8272584581250042	0.9989257907674315	0.8472584581250042
13	DTREE GINI IG	0.895095190342379	0.9893396080954238	0.8315264393216549	0.9989608036355629	0.8315264393216549
14	DTREE GINI CC	0.895095190342379	0.9893396080954238	0.8315264393216549	0.9989608036355629	0.8315264393216549
15	DTREE GINI CT	0.895095190342379	0.9893396080954238	0.8315264393216549	0.9989608036355629	0.8315264393216549
16	MLP IG	0.787941134082369	0.9913596052077562	0.7039245085201133	0.9982060417050184	0.7039245085201135
17	MLP CC	0.826444765298859	0.987274904015663	0.7456663619457062	0.9984394608258946	0.7456663619457062
18	MLP CT	0.824048034970543	0.9926672311279199	0.7416603575767171	0.9984322416778263	0.7416603575767172

Graph 4 : ROC curve of Decision Tree (entropy) Classifier



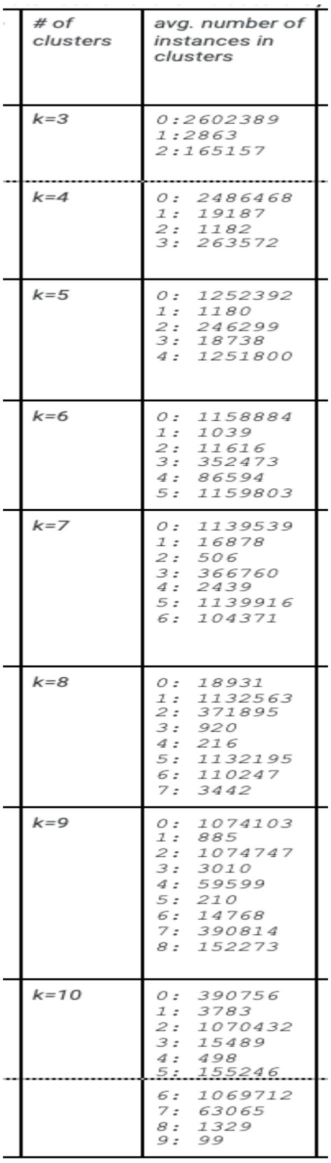
We compared two best models by using 5 different metrics(Accuracy, P, R, F1, AUC) according to T-Test. According to all score metric types there’s a significant difference between models.So we can conclude that DecisionTree-Entropy-IG is our best experiment.

Also, our decision tree classifier model performed much better than our deep learning model. In general, the Correlation coefficient and Chi-square test feature selection methods resulted in higher accuracy than the Information gain method.

We used a k-mean clustering algorithm to cluster Fraud Detection data into groups. Our aim was to observe how the instance groups were separated from each other and how close they were to each other. We tried k values by clustering from 3 to 10 and found the optimum k value.

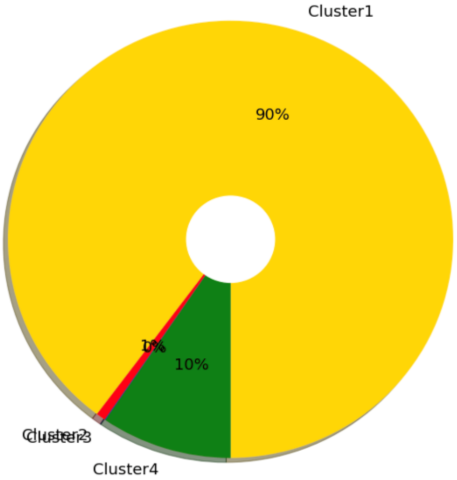
We tried the Elbow method to decide which k value is optimal. In this method, the optimum k value is determined at the point where the SSE transition values differ significantly according to the changing k values. Accordingly, we found the optimum point at the point where the k value is 4. This is the optimal number of clusters into which the data may be clustered and best model performance.

Table 2: Distribution of instances for different k values



# of clusters	avg. number of instances in clusters
k=3	0:2602389 1:2863 2:165157
k=4	0: 2486468 1: 19187 2: 1182 3: 263572
k=5	0: 1252392 1: 1180 2: 246299 3: 18738 4: 1251800
k=6	0: 1158884 1: 1039 2: 11616 3: 352473 4: 86594 5: 1159803
k=7	0: 1139539 1: 16878 2: 506 3: 366760 4: 2439 5: 1139916 6: 104371
k=8	0: 18931 1: 1132563 2: 371895 3: 920 4: 216 5: 1132195 6: 110247 7: 3442
k=9	0: 1074103 1: 885 2: 1074747 3: 3010 4: 59599 5: 210 6: 14768 7: 390814 8: 152273
k=10	0: 390756 1: 3783 2: 1070432 3: 15489 4: 498 5: 155246
	6: 1069712 7: 63065 8: 1329 9: 99

Graph 5: Pie chart of instance distribution with k=4



We look at the details of the k=4 model, which is 90% of all instances in cluster 1, other instances scattered in cluster 2, cluster 3, cluster 4.

6. Conclusions

Fraud is one of the major problems in the financial industry. It creates a big problem especially in payment and money transfer systems. In this fraud detection study, using different classification algorithms (MLP(Neural Network), kNN, Decision Tree and Naive Bayes) we made it possible for the developed models to detect whether the examples they have not seen before are fraudulent or not. The best among these models is the Decision tree, which we tried with the entropy parameter. This model has an F1 score of 89% and an AUC of 84%. In the descriptive analytics section, we used the k-means clustering algorithm to observe how the instances are grouped among each other.
We developed k-means clustering modeling using the k value in the range of 3-10. For each k-value, we measured the model's SSE: Sum of squared error, RI: Rand index, NMI: Normalized Mutual Information and Silhouette Value.
The model with k value of 4 turned out to be the best because the method showed the highest performance and was the optimum value. As a result, in the fraud detection study, we had a model that predicted 89% accurately whether the sample received was fraudulent or non-fraudulent. We also observed the difference of our data with different groups.

7. Acknowledgments

We would like to thank our advisor, Murat Can Ganiz, who guided us in our improvements throughout the term.

8.References

[1] Alexopoulos, P., Kafentzis, K., Benetou, X., Tagaris, T., & Georgolios, P. (2007, July). Towards a Generic Fraud Ontology in e-Government. In *ICE-B* (pp. 269-276).

[2] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical science*, *17*(3), 235-255.

[3] Botchey, F. E., Qin, Z., & Hughes-Lartey, K. (2020). Mobile Money Fraud Prediction—A Cross-Case Analysis on the Efficiency of Support Vector Machines, Gradient Boosted Decision Trees, and Naïve Bayes Algorithms. *Information*, *11*(8), 383.

[4] Choi, D., & Lee, K. (2017). Machine learning based approach to financial fraud detection process in mobile payment system. *IT CoNvergence PRActice (INPRA)*, *5*(4), 12-24.

[5] Investigating Fraudulent Acts,
2000 UNIVERSITY OF HOUSTON SYSTEM ADMINISTRATIVE MEMORANDUM,

[6] Sadgali, I., Nawal, S. A. E. L., & BENABBOU, F. (2019, October). Fraud detection in credit card transaction using machine learning techniques. In *2019 1st International Conference on Smart Systems and Data Science (ICSSD)* (pp. 1-4). IEEE.

APPENDIX

➔ Delivery 0

Project Name: Fraud Detection on Financial Data

Project description:
This project is designed for the financial services industry, where cyber fraud needs to be tackled, and is aimed at finding out whether transactions are fraudulent, through money transfers and a number of factors.

Dataset explanation:
Consists of 11 attributes and 2,722,362 instances in total. Attributes are as explained below:

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.

newbalanceDest: new balance recipient after the transaction.

isFraud: whether the fraud happened or not.

isFlaggedFraud: attempt to transfer more than 200.000 in a single transaction.

Aim of the project:

The financial services industry and the industries that involve financial transactions are suffering from fraud-related losses and damages. The number of fraudulent customers has reached a high level in recent years. The reason for this is the money stolen from banks. The shift to the digital space opens new channels for financial service delivery. It also created a rich environment for scammers.

In conjunction with this project, development and improvement can be made in the process of detecting real-time fraud attempts.

➔ **Delivery 1**

Project Explanation

The financial services industry and the industries that involve financial transactions are suffering from fraud-related losses and damages. The number of fraudulent customers has reached a high level in recent years. The reason for this is the money stolen from banks. The shift to the digital space opens new channels for financial service delivery. It also created a rich environment for scammers. As a consequence of this, the need for automatic systems which are able to detect and fight fraudsters has emerged.

Fraud detection is notably a challenging problem because;
Fraud strategies change in time, as well as customers' spending habits evolve. Few examples of frauds available, so it is hard to create a model of fraudulent behavior.
Not all frauds are reported or reported with a large delay.
Few transactions can be timely investigated.

If earlier criminals had to counterfeit client IDs, now getting a person's account password may be all that's needed to steal money. With fraudsters becoming more adept at finding and exploiting loopholes in systems, fraud management has turned painful for the banking and finance industry. Customer loyalty and conversions are affected by fraudsters.

In order to maintain customer loyalty and conversions, financial services firm's need to detect fraud correctly and rapidly. Machine Learning and Deep Learning systems can detect changing strategies of fraudness quickly and correctly as needed.

Why we use Machine Learning and Deep Learning to detect fraud has 4 main reasons: • Scalable
• Faster
• Efficient
• More accurate
There are 11 attributes in our data with approximately 6.3 million instances. Attributes are ;step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud.

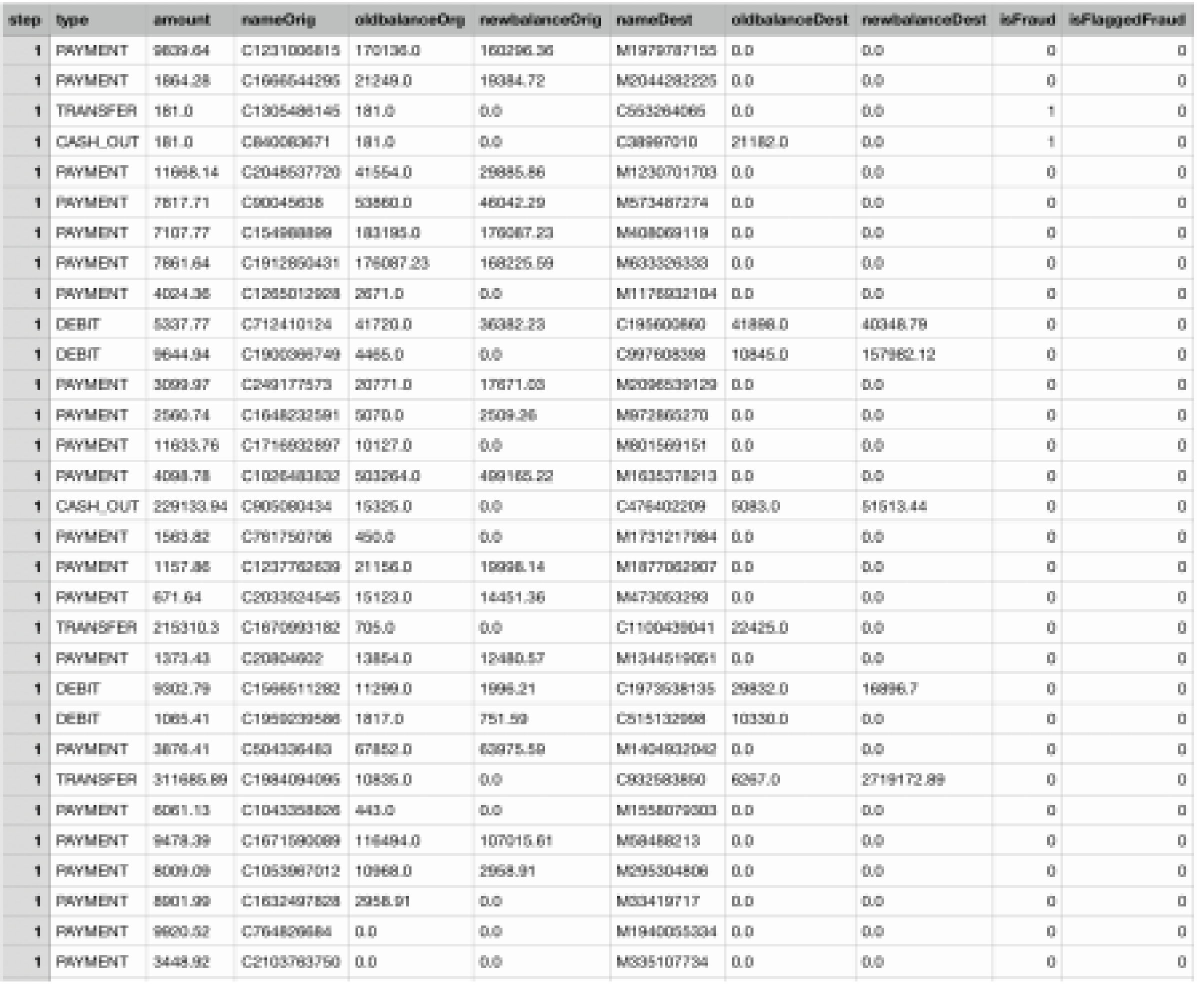
Our main purpose is to detect fraud activities in transactions between accounts.

Our project will take the financial sector one step further by identifying fraud, which is the bleeding wound of the financial sector, through machine learning by modeling in a way to detect fraud using the data set we have obtained.

Data Statistics

There are 11 columns and 6.362.622 rows in our dataset.

Explanation of Dataset Attributes:



step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
1	TRANSFER	181.0	C1305486145	181.0	0.0	C553264065	0.0	0.0	1	0
1	CASH_OUT	181.0	C840083671	181.0	0.0	C38997010	21182.0	0.0	1	0
1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0
1	PAYMENT	7817.71	C90045638	53860.0	46042.29	M573487274	0.0	0.0	0	0
1	PAYMENT	7107.77	C154988899	183195.0	176087.23	M408069119	0.0	0.0	0	0
1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0.0	0.0	0	0
1	PAYMENT	4024.36	C1265012928	2671.0	0.0	M1176932104	0.0	0.0	0	0
1	DEBIT	5337.77	C712410124	41720.0	36382.23	C195600860	41898.0	40348.79	0	0
1	DEBIT	9644.94	C1900366749	4465.0	0.0	C997608398	10845.0	157982.12	0	0
1	PAYMENT	3099.97	C249177573	20771.0	17671.03	M2096539129	0.0	0.0	0	0
1	PAYMENT	2560.74	C1648232591	5070.0	2509.26	M972865270	0.0	0.0	0	0
1	PAYMENT	11633.76	C1716932897	10127.0	0.0	M801569151	0.0	0.0	0	0
1	PAYMENT	4098.78	C1026483832	503264.0	499165.22	M1635378213	0.0	0.0	0	0
1	CASH_OUT	229133.94	C905080434	15325.0	0.0	C476402209	5083.0	51513.44	0	0
1	PAYMENT	1563.82	C761750706	450.0	0.0	M1731217984	0.0	0.0	0	0
1	PAYMENT	1157.86	C1237762639	21156.0	19998.14	M1877062907	0.0	0.0	0	0
1	PAYMENT	671.64	C2033524545	15123.0	14451.36	M473053293	0.0	0.0	0	0
1	TRANSFER	215310.3	C1670993182	705.0	0.0	C1100439041	22425.0	0.0	0	0
1	PAYMENT	1373.43	C20804602	13854.0	12480.57	M1344519051	0.0	0.0	0	0
1	DEBIT	9302.79	C1566511282	11299.0	1996.21	C1973538135	29832.0	16896.7	0	0
1	DEBIT	1065.41	C1959239586	1817.0	751.59	C515132998	10330.0	0.0	0	0
1	PAYMENT	3876.41	C504336483	67852.0	63975.59	M1404932042	0.0	0.0	0	0
1	TRANSFER	311685.89	C1984094095	10835.0	0.0	C932583850	6267.0	2719172.89	0	0
1	PAYMENT	6061.13	C1043358826	443.0	0.0	M1558079303	0.0	0.0	0	0
1	PAYMENT	9478.39	C1671590089	116494.0	107015.61	M58488213	0.0	0.0	0	0
1	PAYMENT	8009.09	C1053967012	10968.0	2958.91	M295304806	0.0	0.0	0	0
1	PAYMENT	8901.99	C1632497828	2958.91	0.0	M33419717	0.0	0.0	0	0
1	PAYMENT	9920.52	C764826684	0.0	0.0	M1940055334	0.0	0.0	0	0
1	PAYMENT	3448.92	C2103763750	0.0	0.0	M335107734	0.0	0.0	0	0

step (numeric): 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 (nominal): CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

amount (numeric): amount of the transaction in local currency.

nameOrig (nominal): customer who started the transaction

oldbalanceOrg (numeric): initial balance before the transaction

newbalanceOrig (numeric): new balance after the transaction

nameDest (nominal): customer who is the recipient of the transaction oldbalanceDest (numeric): initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).

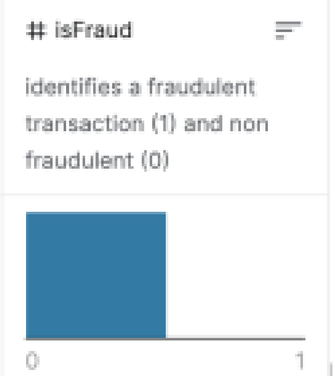
newbalanceDest (numeric): new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).

isFraud (boolean): 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 trying to empty the funds by transferring to another account and then cashing out of the system.

isFlaggedFraud (boolean): 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.

Target Attributes

- Our target attribute is isFraud column data.



isFraud

identifies a fraudulent transaction (1) and non fraudulent (0)

0, represents non fraudulent transaction
1, represents fraudulent transaction

- We will use step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFlaggedFraud parameters and target attribute(isFraud) for classification modeling.
    - After having these attributes preprocessed and ready to be used in order to classify an instance - which will the future input to be predicted, we are planning to try several different classification algorithms such as Decision Trees (Gain Ratio and Gini index), Naive Bayes, Neural Networks (will be experimented with different number of hidden layers) and Support Vector Machines (SVM). These algorithms may require different types of hyperparameters and the experimentation phase also will include hypertuning these parameters and see which one is the best fitting one for our goal.

➔ Delivery 2

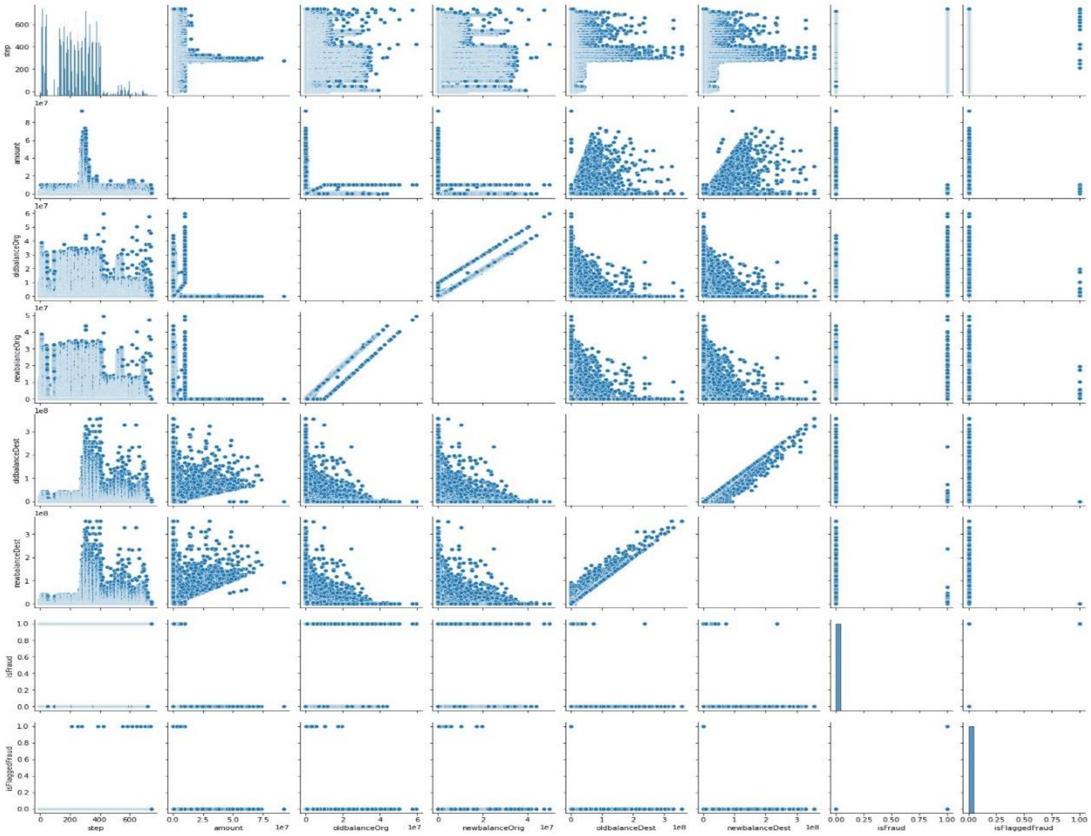
Feature Names	Description	Type	Missing Values(%)
step	maps a unit of time in the real world	Numeric	0
type	CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER	Nominal	0
amount	amount of the transaction in local currency	Numeric	0
nameOrig	customer who started the transaction	Nominal	0
oldbalanceOrg	initial balance before the transaction	Numeric	0
newbalanceOrig	new balance after the transaction	Numeric	0
nameDest	customer who is the recipient of the transaction	Nominal	0
oldbalanceDest	initial balance recipient before the transaction	Numeric	0
newbalanceDest	new balance recipient after the transaction	Numeric	0
isFraud	This is the transactions made by the fraudulent agents inside the simulation	Nominal	0
isFlaggedFraud	The business model aims to control massive transfers from one account to another and flags illegal attempts	Nominal	0

Feature Names	Average	Std.Dev.	Entropy	#of values
step	243.397	142.332	5.5276	6.362.622
type	-	-	1.3077	6.362.622
amount	179861.9	603858.2	15.4085	6.362.622
nameOrig	-	-	15.6639	6.362.622
oldbalanceOrg	833883.10	2888243	9.3923	6.362.622
newbalanceOrig	855116.66	2924049	7.0845	6.362.622
nameDest	-	-	14.0318	6.362.622
oldbalanceDest	1100701.66	3399180	9.3598	6.362.622
newbalanceDest	1224996.39	3674129	9.9373	6.362.622
isFraud	-	-	0.0098	6.362.622
isFlaggedFraud	-	-	0.000034	6.362.622

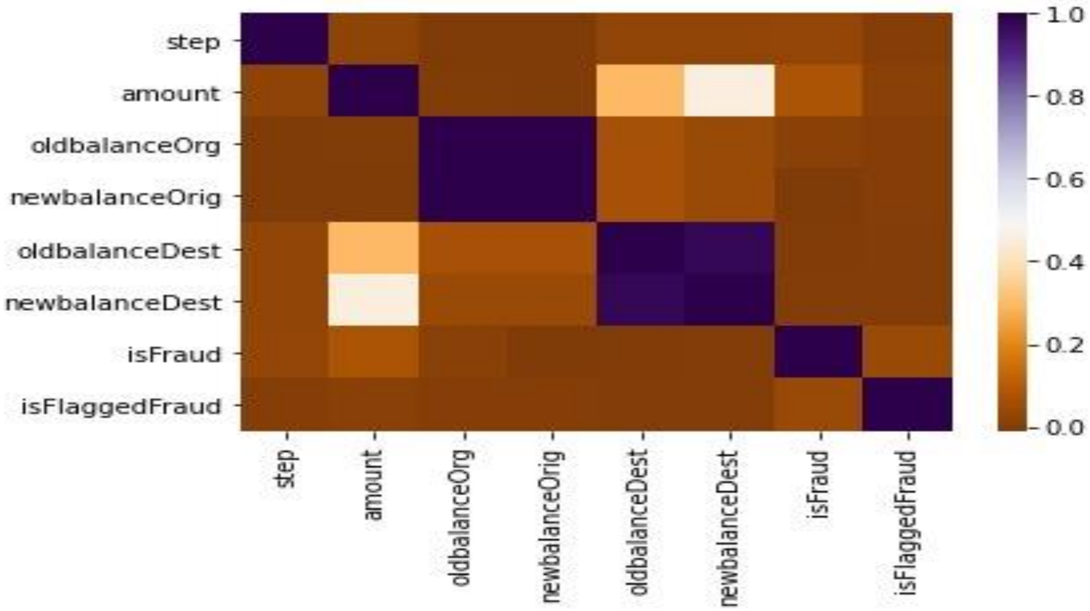
Feature Names	Min. Or Least Frequent	Max. Or Most Frequent
step	1	743
type	DEBIT	CASH_OUT
amount	0	92445516,64
nameOrig	C2066766136	C1999539787
oldbalanceOrg	0	59585040.37
newbalanceOrig	0	49585040,37
nameDest	M917557255	C1286084959
oldbalanceDest	0	356015889,92
newbalanceDest	0	356179278,92
isFraud	0	1
isFlaggedFraud	0	1

➔ Delivery 3

Scatter plot matrix

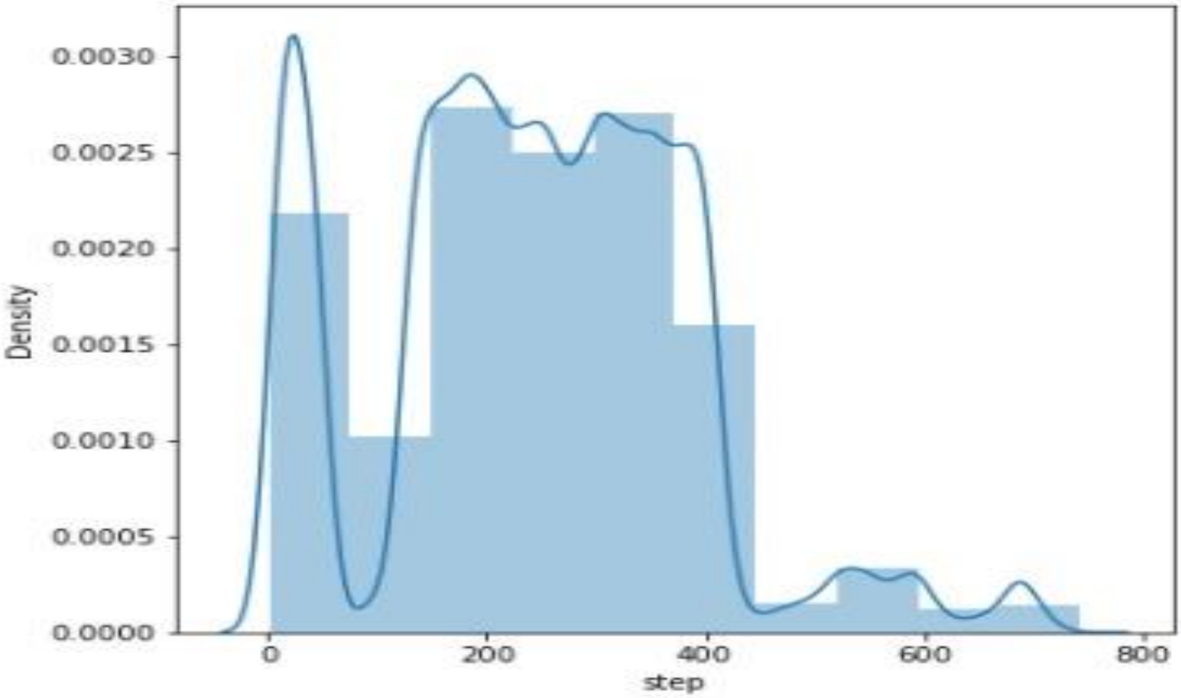


Heat map

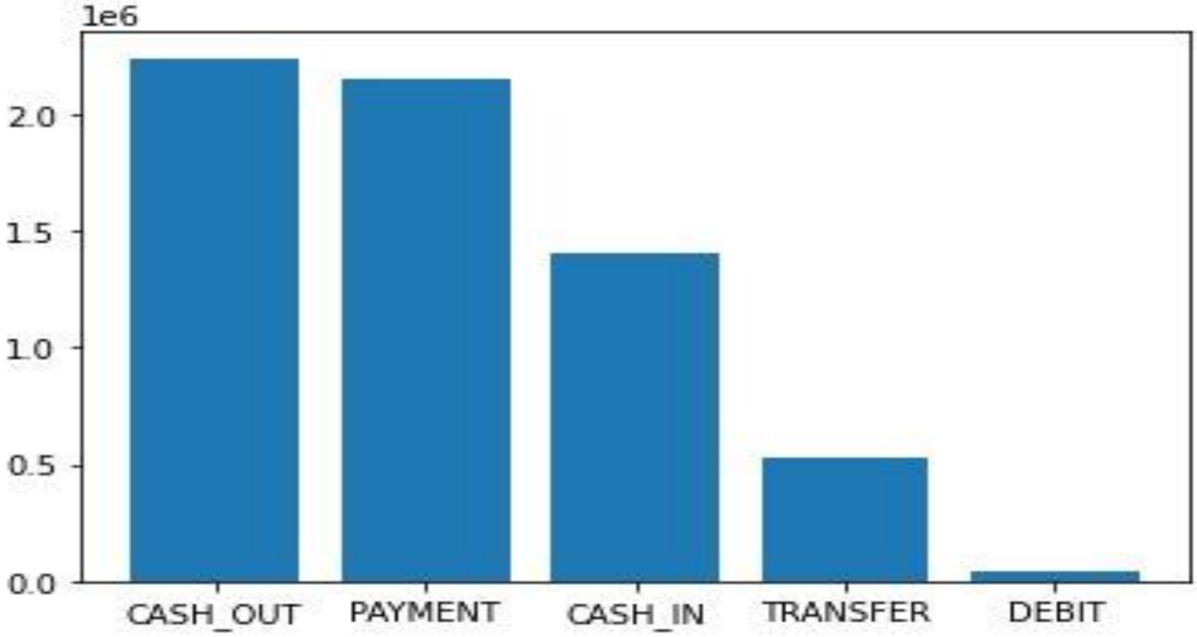


Charts of Attributes

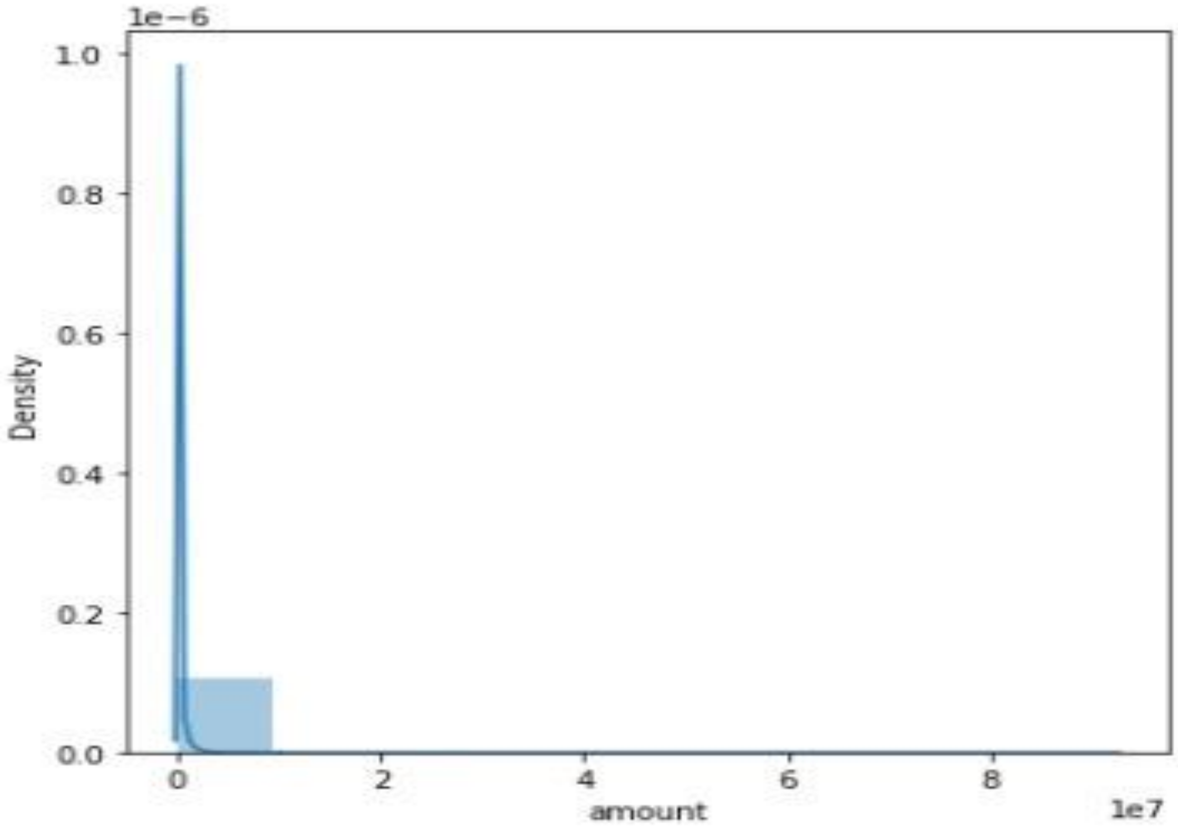
Step



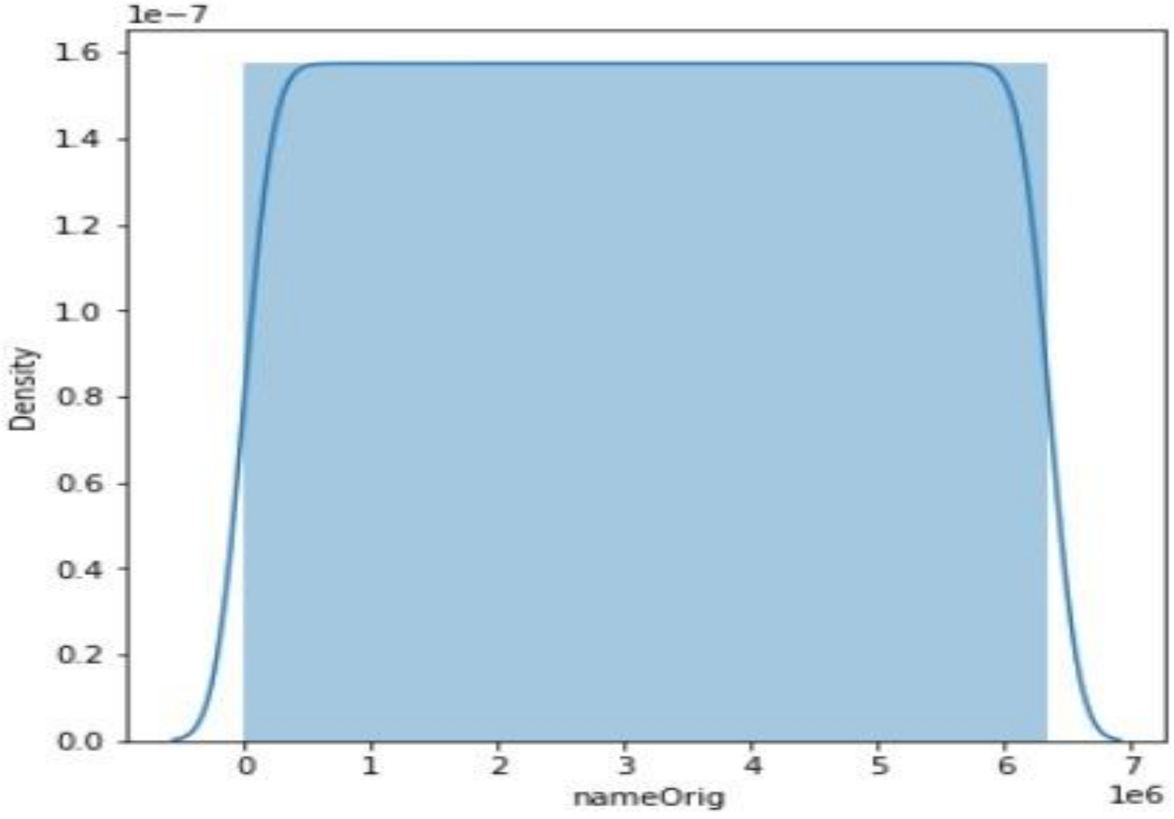
Type



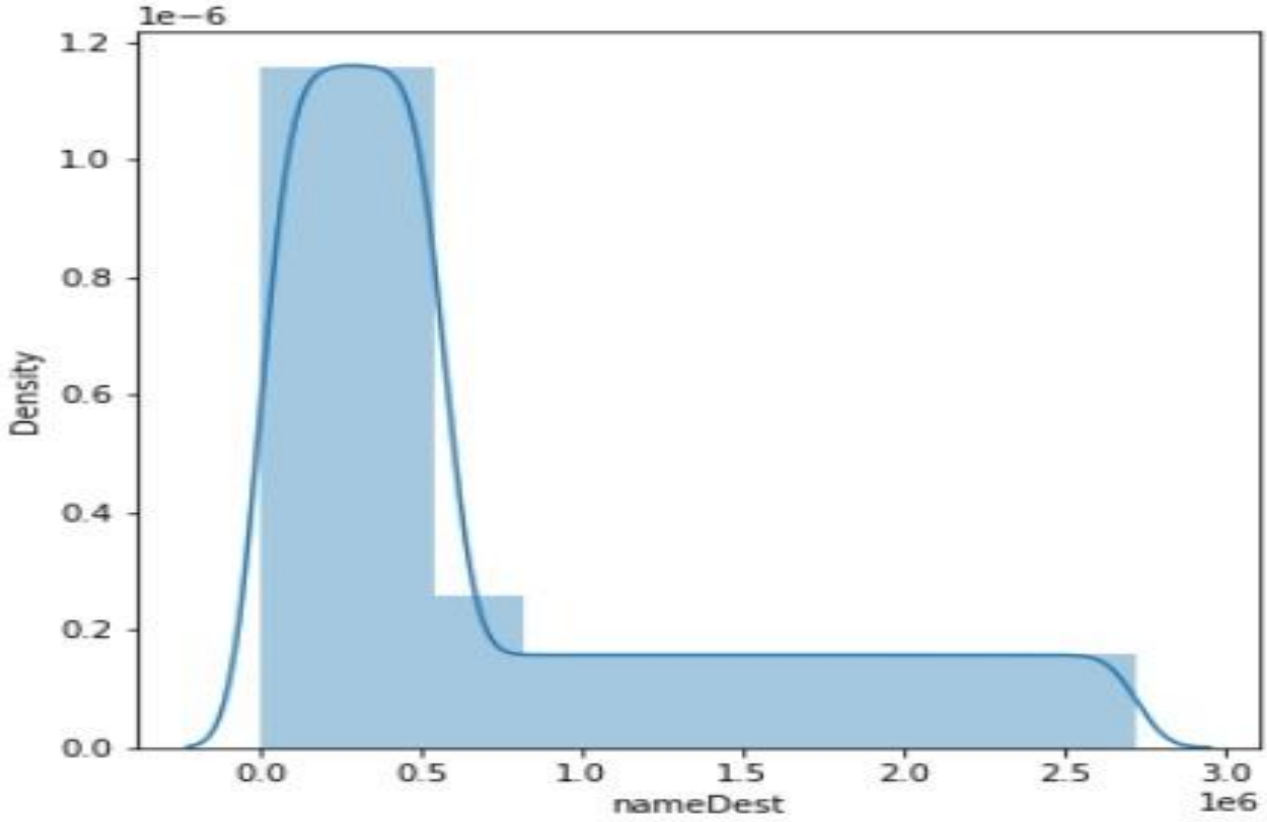
Amount



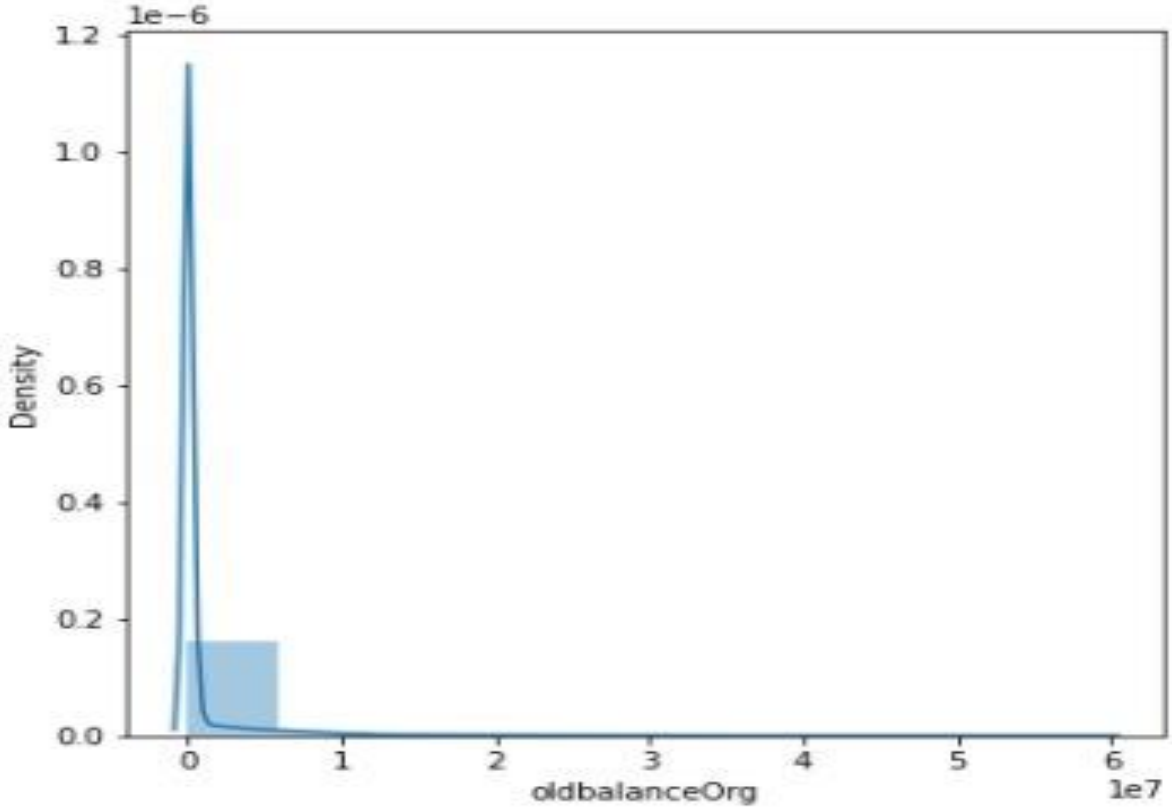
NameOrig



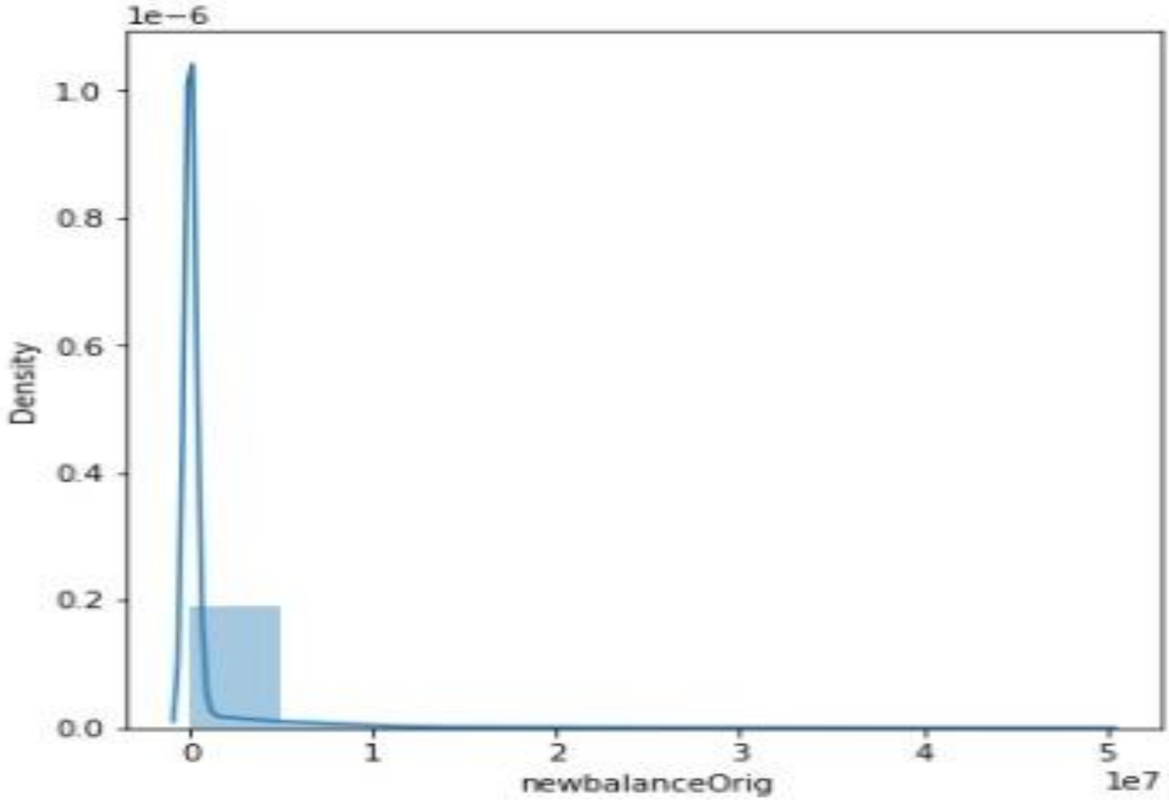
NameDest



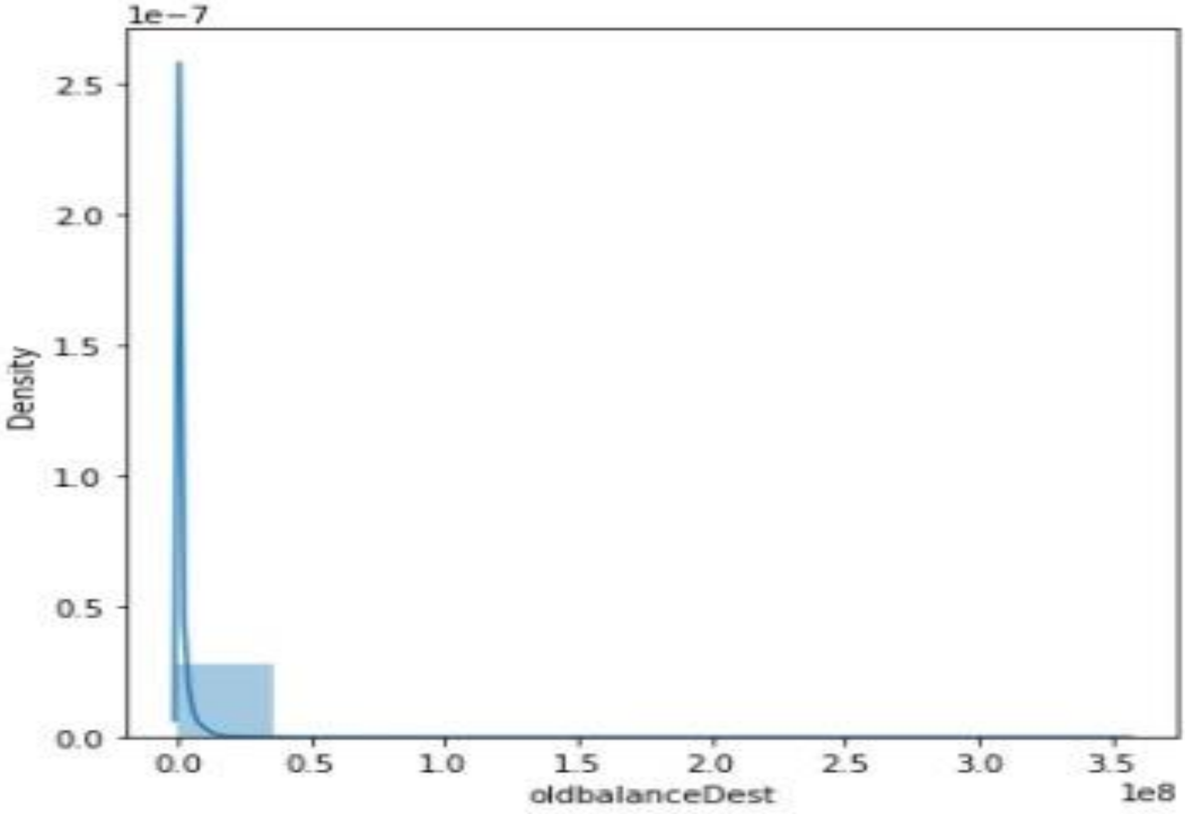
OldBalanceOrg



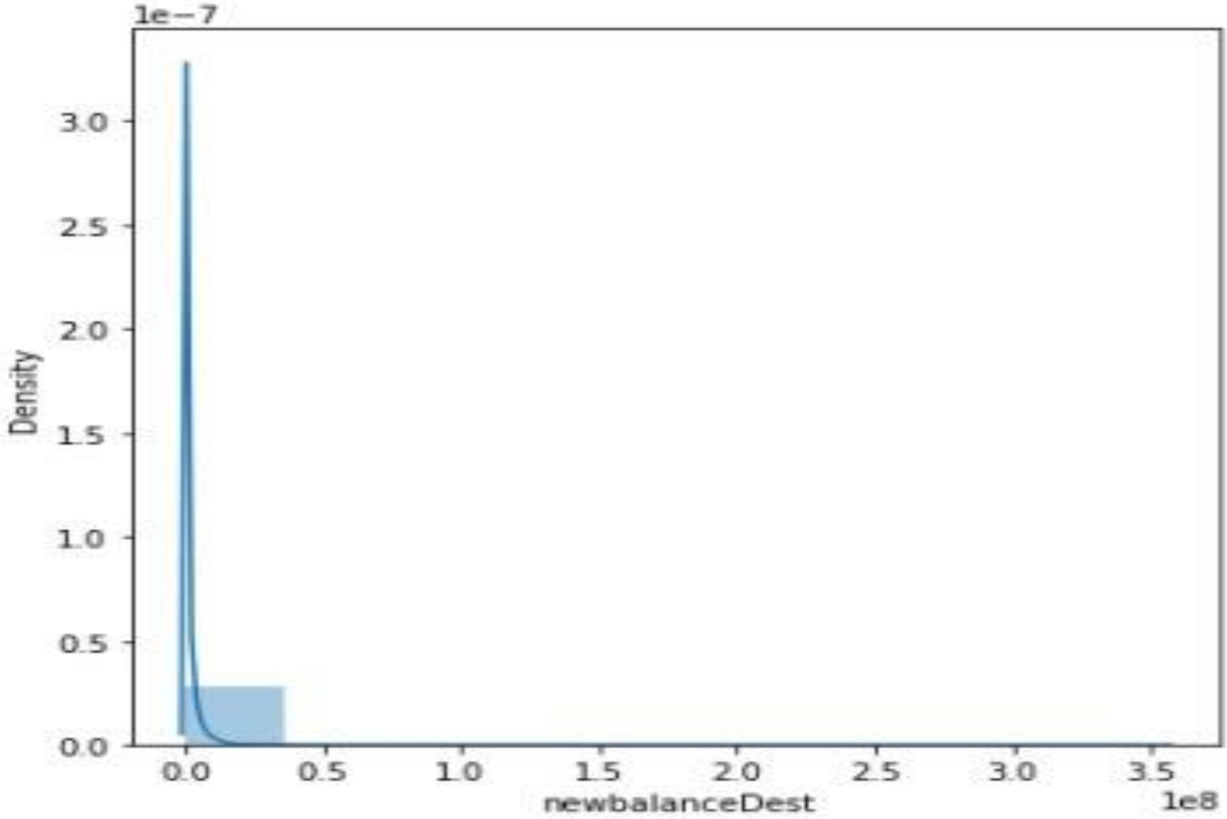
NewBalanceOrig



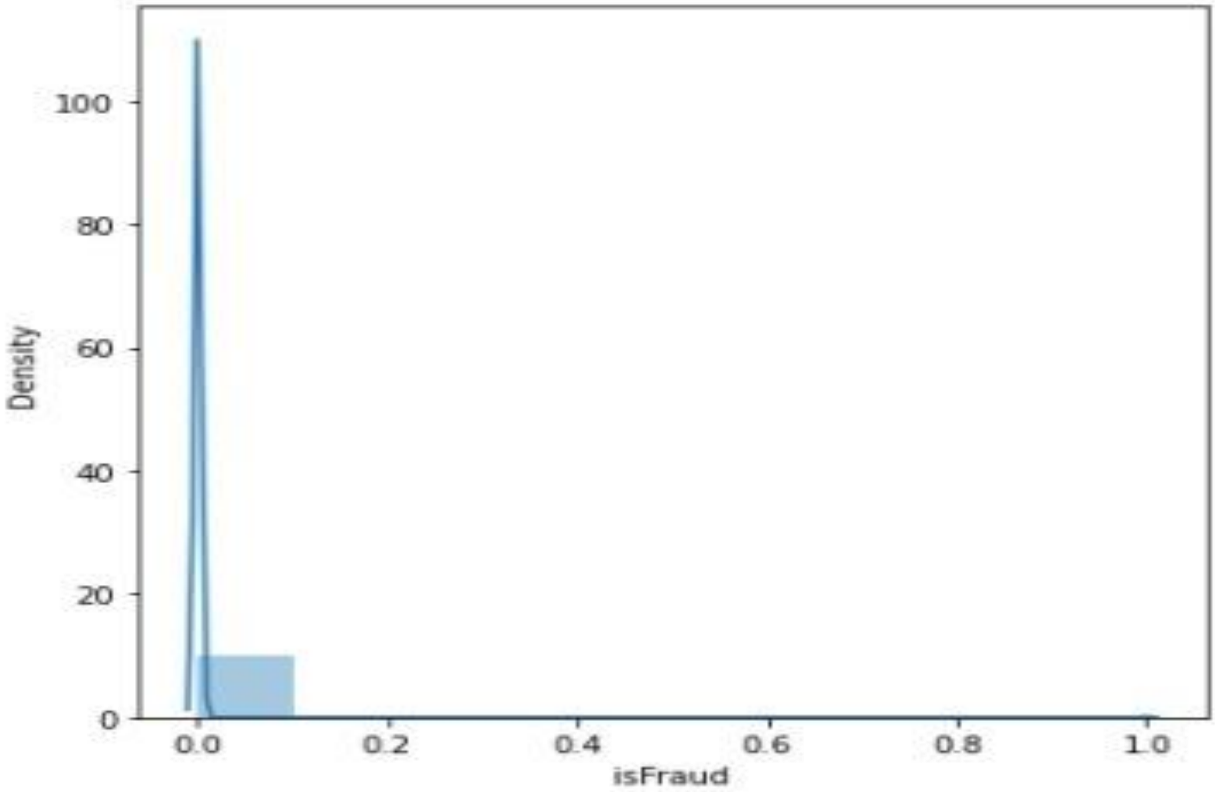
OldBalanceDest



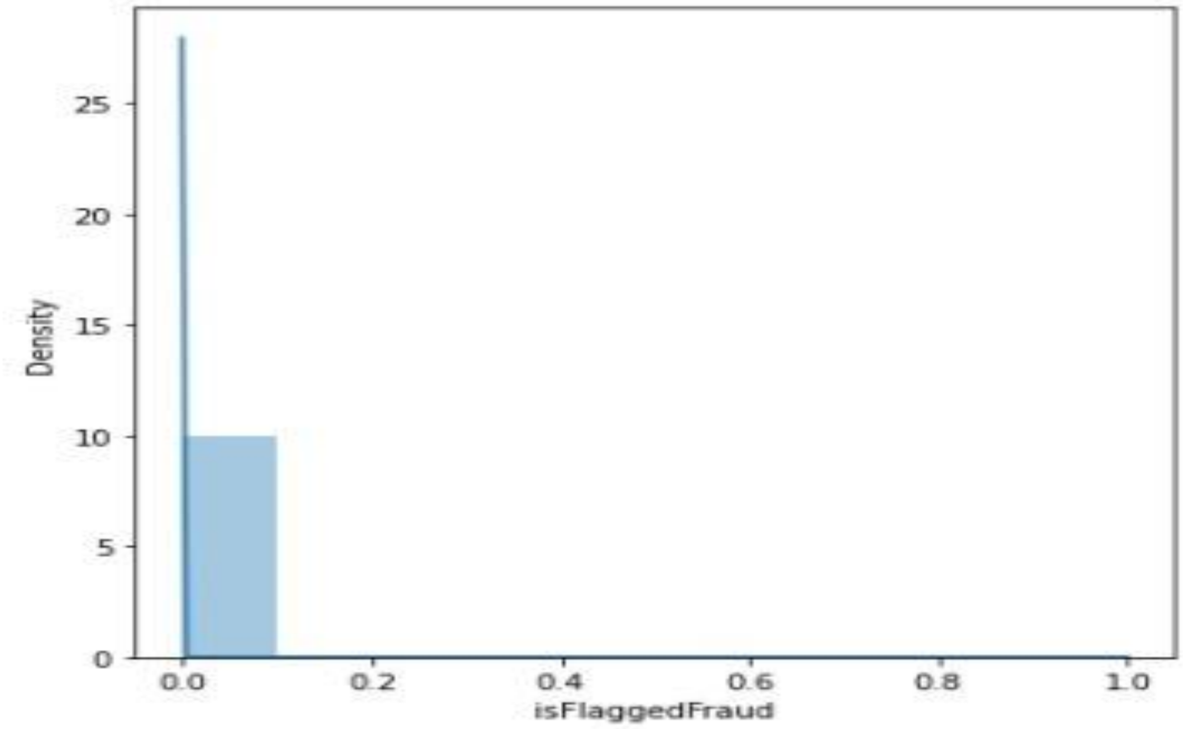
NewBalanceDest



IsFraud



IsFlaggedFraud



Results

- R values lower than 0.20 and near zero indicate no relationship or very weak relationship.
- Weak relationship between 0.20-0.39
- Medium level relationship between 0.40-0.59
- Strong relationship between 0.6-1.00

According to our scatter plot and correlation matrix, there is strong positive correlation between newBalanceOrig-oldBalanceOrg and newBalanceDest-oldBalanceDest. There are Medium level positive correlation between amount- oldBalanceDest and amount-newBalanceDest. Other relationships between attributes has weak positive or no correlation.

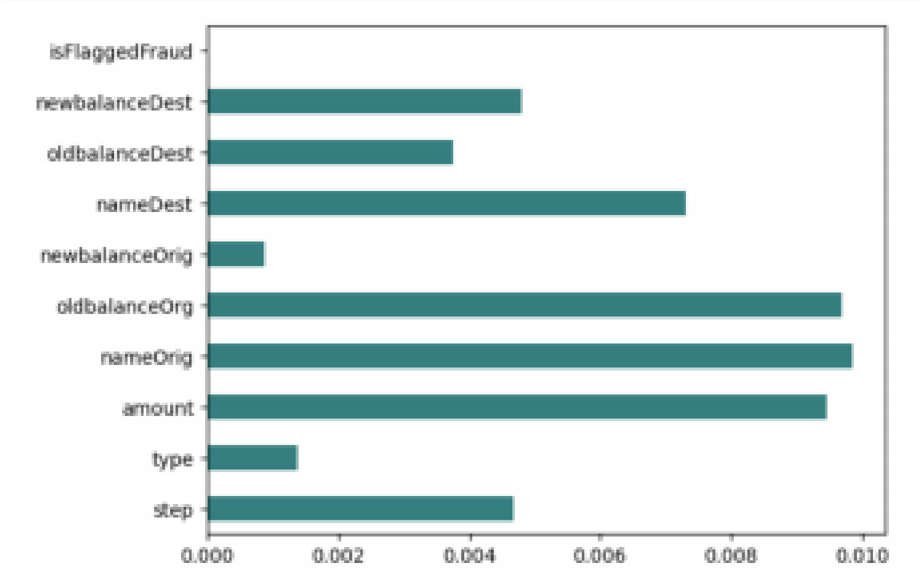
According to our attribute charts:

- Most of the transactions occurred between step 0 and 400. After 400 transaction counts has decreased significantly
- Most of the transactions has occurred as a “CASH_OUT, PAYMENT and CASH_IN” type. TRANSFER and DEBIT’s counts are lower than the other 3 type.
- More than 99% of the amount’s instances are lower than 100,000 $. That’s why Amount chart is right-skewed
- Most of the new and old balances of original and destination accounts’ values are lower than 200,000$. That’s why these charts are right skewed too.
- There are 6.3 million instances in our dataset and there are only 8213 fraud instances in our dataset. That’s why there is accumulation of non-fraud labeled data.
- In nameOrig and nameDest, we encoded the unique values to numbers. One unique value occurred at most 1 time in nameOrig chart and at most 6 times in nameDest chart.

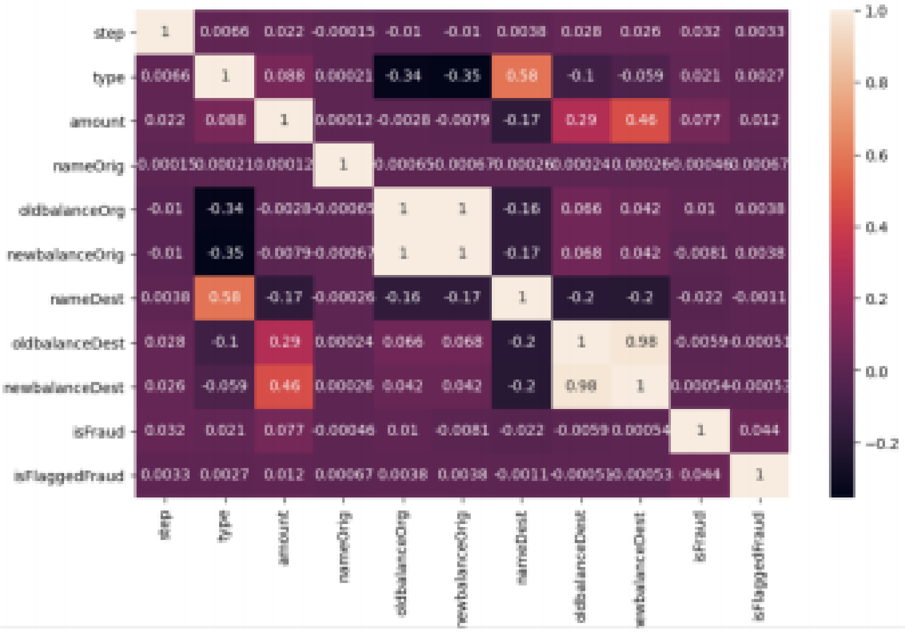
➔ Delivery 4

1- Feature Selection Methods

Information Gain Importance of Attributes Graphic:



Correlation Coefficient Importance of Attributes Graphic:



Chi-Square Test Importance of Attributes

Graphic:

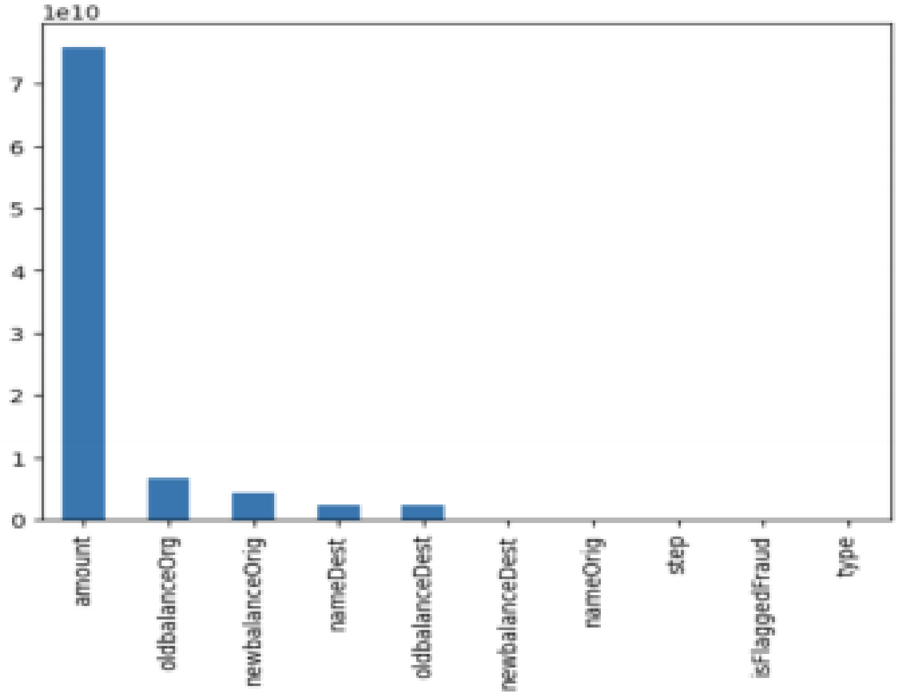
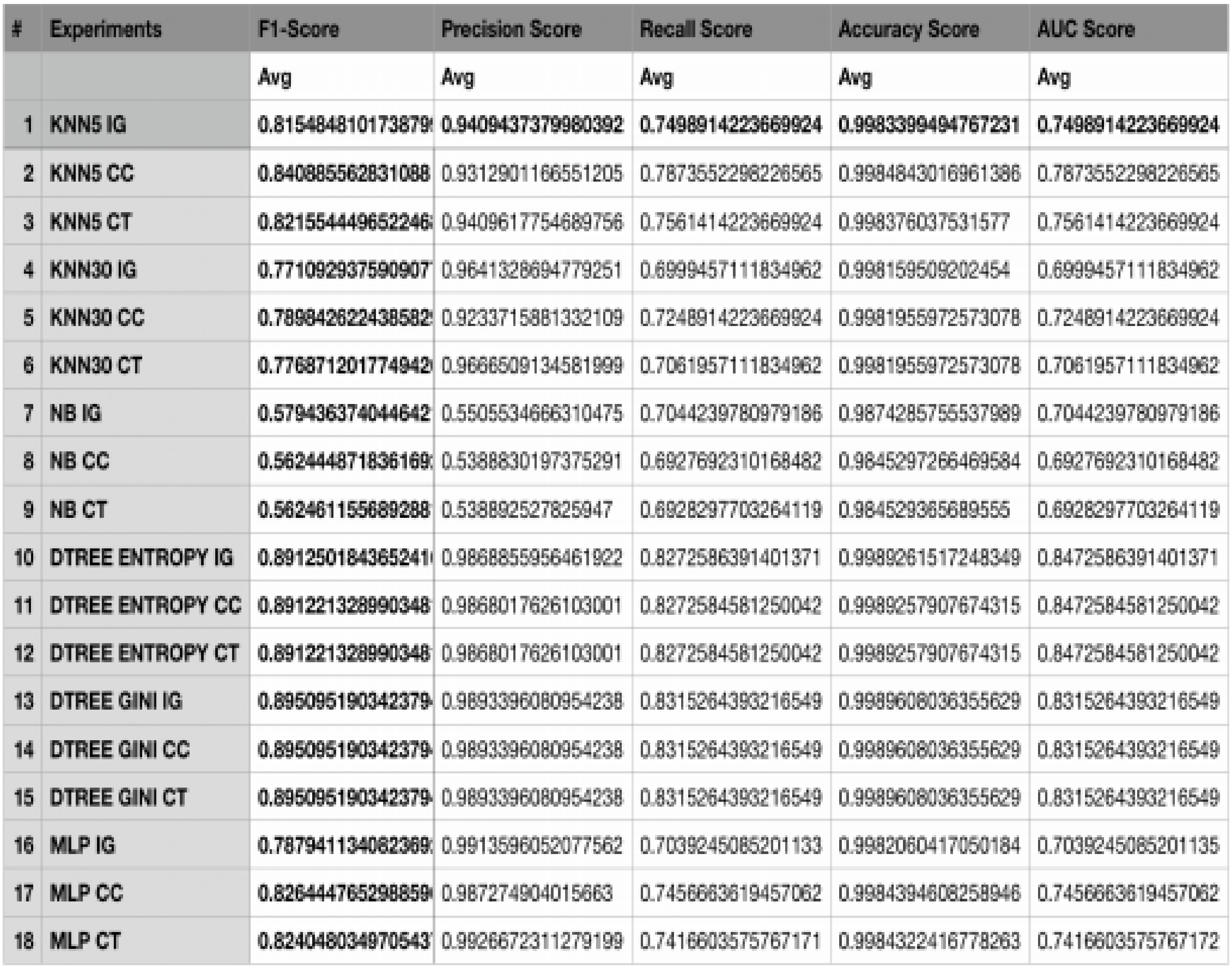


Table of Feature Extraction Values

Feature Names	Information Gain Importance	Correlation Coefficient Importances	Chi-Square Importances
step	4.68873089e-03	0.032	5.28059079e+05
type	1.38039937e-03	0.021	2.93663069e+0
amount	9.45706635e-03	0.077	7.58623617e+10
nameOrig	9.87103998e-03	-0.00046	1.44913424e+06
oldbalanceOrg	9.68981785e-03	0.01	6.56309030e+09
newbalanceOrig	8.84693967e-04	-0.0081	4.22377158e+09
nameDest	7.31022523e-03	-0.022	2.31449109e+09
oldbalanceDest	3.75719688e-03	-0.0059	2.31338860e+09
newbalanceDest	4.80100350e-03	0.00054	2.00946581e+07
isFlaggedFraud	1.67313483e-05	0.044	1.23792173e+04

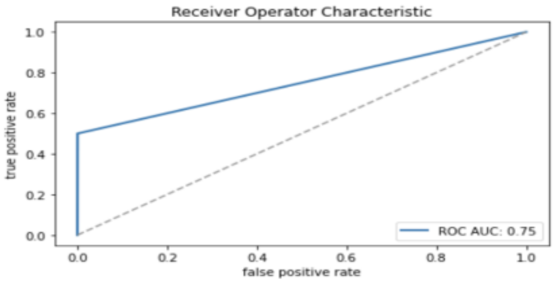
2- Classification Experiments



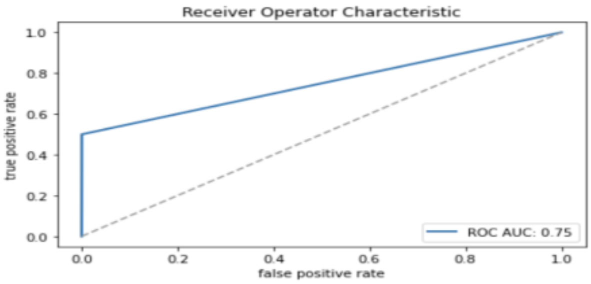
#	Experiments	F1-Score	Precision Score	Recall Score	Accuracy Score	AUC Score
		Avg	Avg	Avg	Avg	Avg
1	KNN5 IG	0.815484810173879	0.9409437379980392	0.7498914223669924	0.9983399494767231	0.7498914223669924
2	KNN5 CC	0.840885562831088	0.9312901166551205	0.7873552298226565	0.9984843016961386	0.7873552298226565
3	KNN5 CT	0.821554449652246	0.9409617754689756	0.7561414223669924	0.998376037531577	0.7561414223669924
4	KNN30 IG	0.771092937590907	0.9641328694779251	0.6999457111834962	0.998159509202454	0.6999457111834962
5	KNN30 CC	0.789842622438582	0.9233715881332109	0.7248914223669924	0.9981955972573078	0.7248914223669924
6	KNN30 CT	0.776871201774942	0.9666509134581999	0.7061957111834962	0.9981955972573078	0.7061957111834962
7	NB IG	0.579436374044642	0.5505534666310475	0.7044239780979186	0.9874285755537989	0.7044239780979186
8	NB CC	0.562444871836169	0.5388830197375291	0.6927692310168482	0.9845297266469584	0.6927692310168482
9	NB CT	0.562461155689288	0.538892527825947	0.6928297703264119	0.984529365689555	0.6928297703264119
10	DTREE ENTROPY IG	0.891250184365241	0.9868855956461922	0.8272586391401371	0.9989261517248349	0.8472586391401371
11	DTREE ENTROPY CC	0.891221328990348	0.9868017626103001	0.8272584581250042	0.9989257907674315	0.8472584581250042
12	DTREE ENTROPY CT	0.891221328990348	0.9868017626103001	0.8272584581250042	0.9989257907674315	0.8472584581250042
13	DTREE GINI IG	0.895095190342379	0.9893396080954238	0.8315264393216549	0.9989608036355629	0.8315264393216549
14	DTREE GINI CC	0.895095190342379	0.9893396080954238	0.8315264393216549	0.9989608036355629	0.8315264393216549
15	DTREE GINI CT	0.895095190342379	0.9893396080954238	0.8315264393216549	0.9989608036355629	0.8315264393216549
16	MLP IG	0.787941134082369	0.9913596052077562	0.7039245085201133	0.9982060417050184	0.7039245085201135
17	MLP CC	0.826444765298859	0.987274904015663	0.7456663619457062	0.9984394608258946	0.7456663619457062
18	MLP CT	0.824048034970543	0.9926672311279199	0.7416603575767171	0.9984322416778263	0.7416603575767172

3- ROC Curves

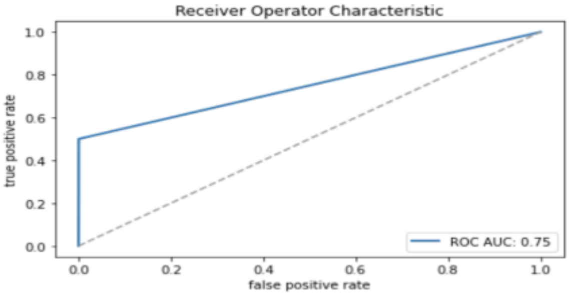
KNN5-IG



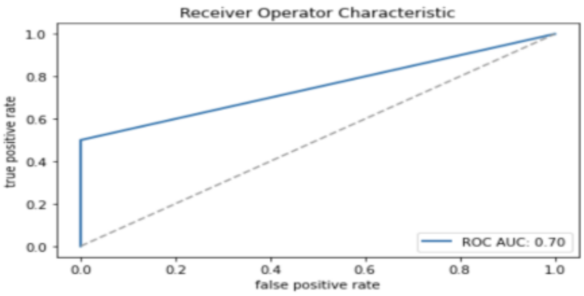
KNN5-CT



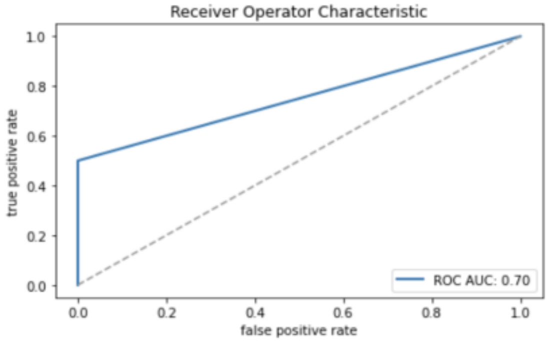
KNN5-CC



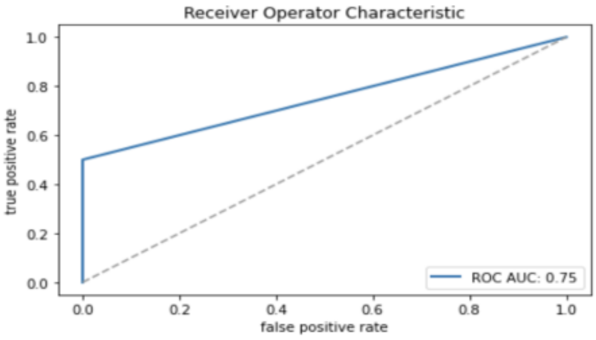
KNN30-IG



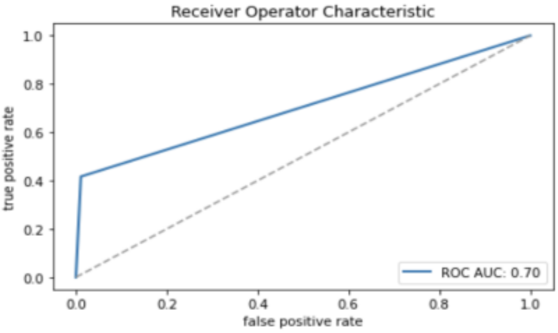
KNN30-CC



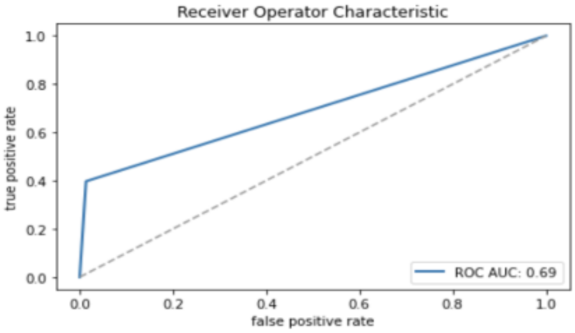
KNN30-CT



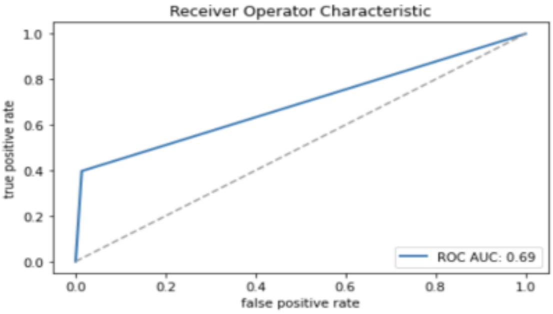
NG-IG



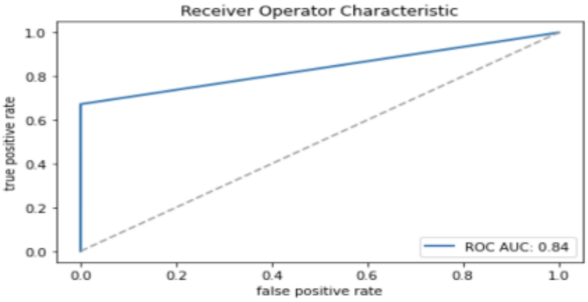
NG-CC



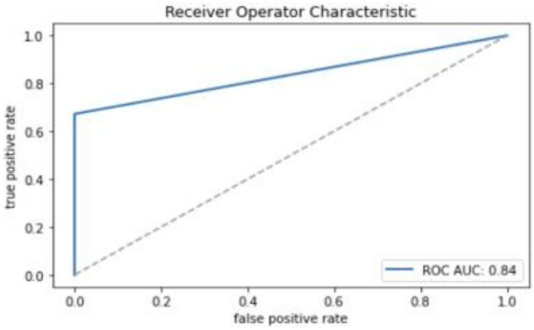
NG-CT



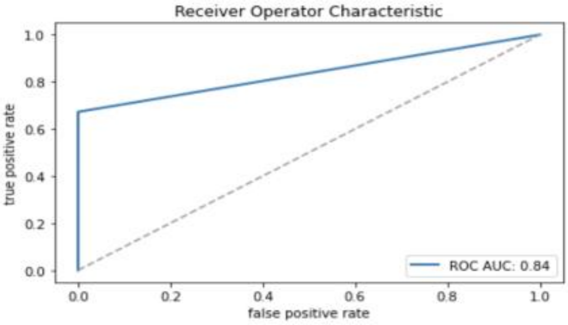
DTREE-ENTROPY-IG



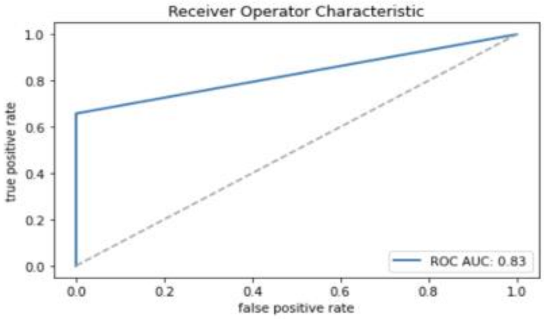
DTREE-ENTROPY-CC



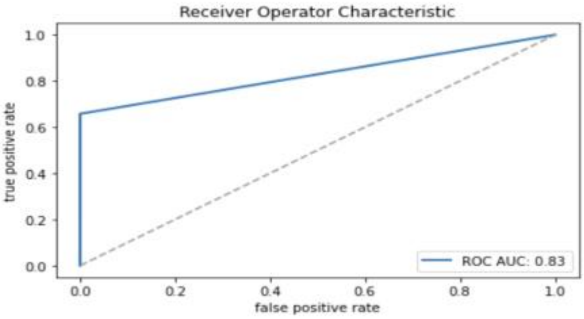
DTREE-ENTROPY-CT



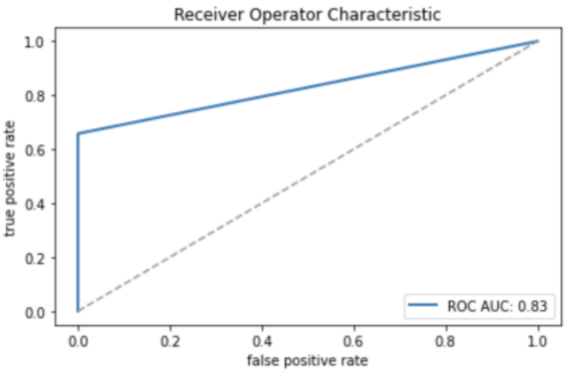
DTREE-GINI-IG



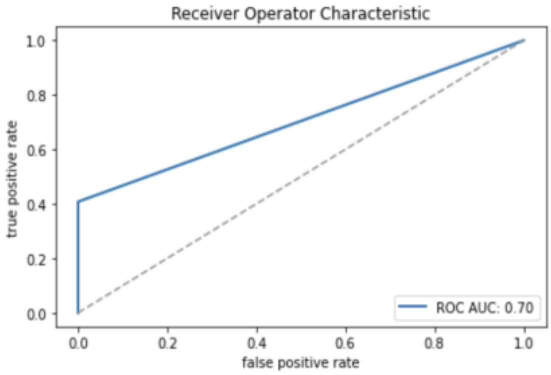
DTREE-GINI-CC



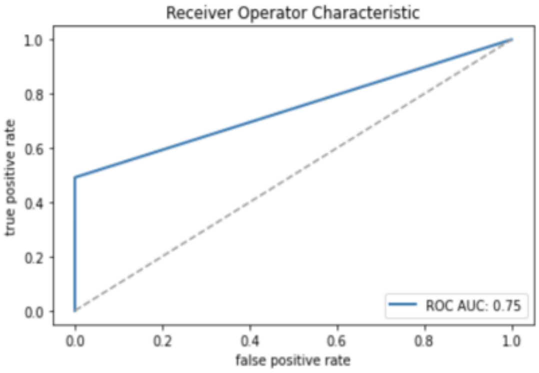
DTREE-GINI-CT



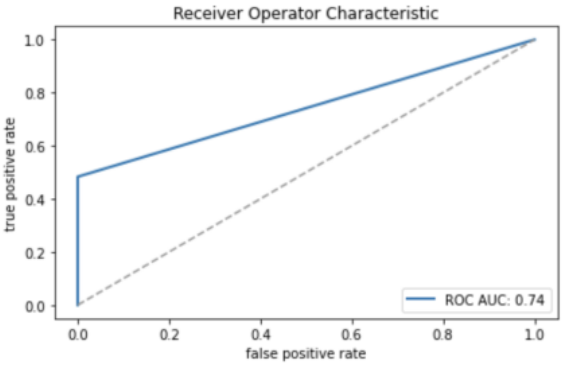
MLP-IG



MLP-CC

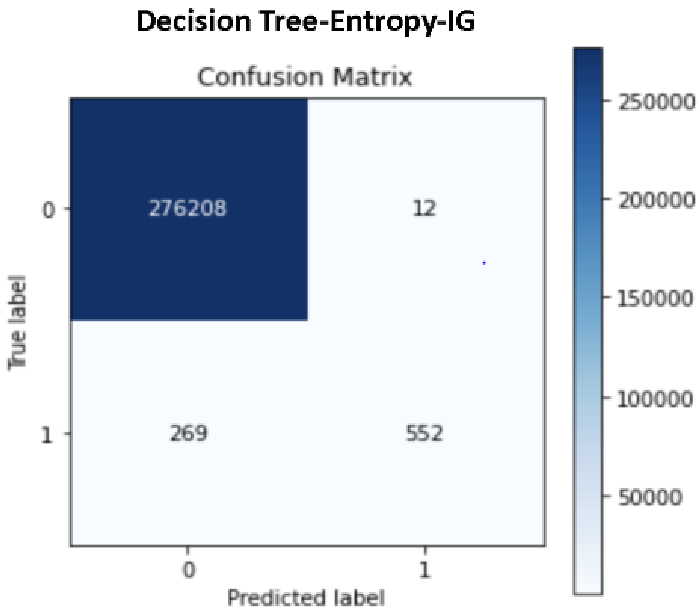


MLP-CT



4- Confusion Matrices

Confusion matrix of the best fold performance of our best performing model:



5 - T-Test

• **Accuracy T_Test** between Decision Tree-Gini-IG & Decision Tree-Entropy-IG: T Value = [7.2228622946768155], P Value = [5.091482790930968e-11] p-value<=0.05 so there's a significant difference between models.

• **Precision T_Test** between Decision Tree-Gini-IG & Decision Tree-Entropy-IG: T Value = [6.129260759394118], P Value = [8.83001671780903e-08]

p-value<=0.05 so there's a significant difference between models.

• **Recall T_Test** between Decision Tree-Gini-IG & Decision Tree-Entropy-IG: T Value = [6.018732092097544], P Value = [1.758104772875413e-07] p-value<=0.05 so there's a significant difference between models.

• **F1 T_Test** between Decision Tree-Gini-IG & Decision Tree-Entropy-IG: T Value = [7.5616320046041725], P Value = [3.9745984281580604e-12]

p-value<=0.05 so there's a significant difference between models.

- **AUC T_Test** between Decision Tree-Gini-IG & Decision Tree-Entropy-IG:

T Value = [6.018732092097659], P Value = [1.758104772875413e-07] p-value<=0.05 so there's a significant difference between models.

6 – Methods

For this iteration of the project, we first determined 3 different feature subsets:

1. Feature selection with feature_selection method of sklearn. Scoring function we chose Information Gain.
2. Feature selection with correlation matrix.
3. Feature selection with the chi2 method of sklearn. Scoring function we chose the Chi-Square Test.

And we determined 6 different classifying models:

1. KNeighborsClassifier model of Sklearn with 5 neighbours
2. KNeighborsClassifier model of Sklearn with 30 neighbours
3. DecisionTreeClassifier model of Sklearn with gini
4. DecisionTreeClassifier model of Sklearn with entropy
5. GaussianNB model of Sklearn.
6. MLP deep learning model of Sklearn

And finally we determined 5 metrics to calculate t-test on:

1. accuracy_score metric of Sklearn.
2. precision_score metric of Sklearn.
3. recall_score metric of Sklearn.
4. f1_score metric of Sklearn.
5. auc_score metric of Sklearn.

7 – Implementation

We used 10 fold cross-validation technique for all classification models. In every fold, we made feature elimination with 3 different methods. Also we used an extra deep learning model(MLP) as a classifier for our 2.770.409 data points. We trained 6 different models and we used information gain, correlation coefficient, chi-square test feature selection methods. We run our 18 different classifiers and run predictions on corresponding test-sets. And finally we evaluate models by using 5 metrics(Accuracy, P, R, F1, AUC) we determined above. In total, we run 18 different experiments in 10 folds.

8 – Conclusion

For t-test, we chose our top two experiments according to F1-Score and

ROC-AUC metric : **1.** Decision Tree-Entropy-IG

- F1-Score: 0.89
- ROC-AUC metric**:** 0.84

2. Decision Tree-Gini-IG

- F1-Score: 0.89
- ROC-AUC metric: 0.83

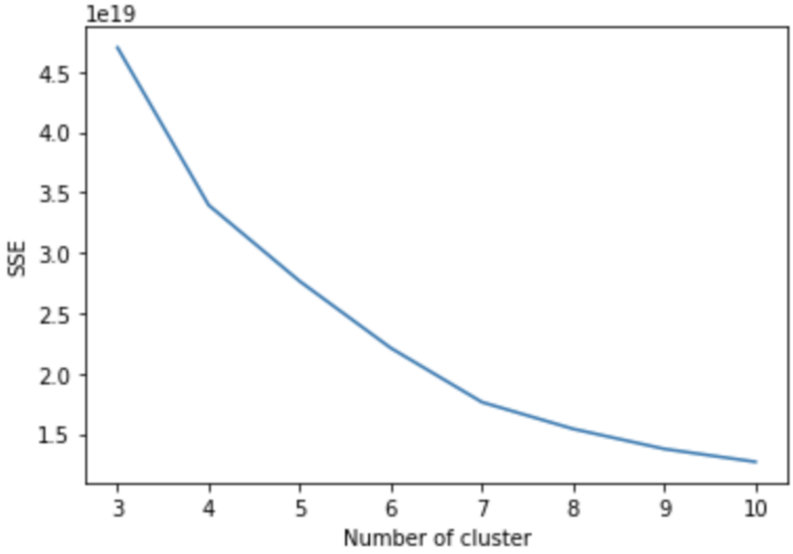
Then run scipy’s t.cdf() method on their 10-Fold CV scores. We compared them by using 5 different metrics(Accuracy, P, R, F1, AUC) according to **T-Test.**

According to all score metric types there’s a significant difference between models. So we can concluded that, **Decision Tree-Entropy-IG** is our best experiment.

Also, our decision tree classifier model performed much better than our deep learning model. In general, the Correlation coefficient and Chi-square test feature selection methods resulted in higher accuracy than the Information gain method.

➔ Delivery 5

Figure 1: The Elbow Plot Showing the Optimal k=4



```
Max silhouette coeficient for df is 0.7021559575716491 for 3 cluster
```

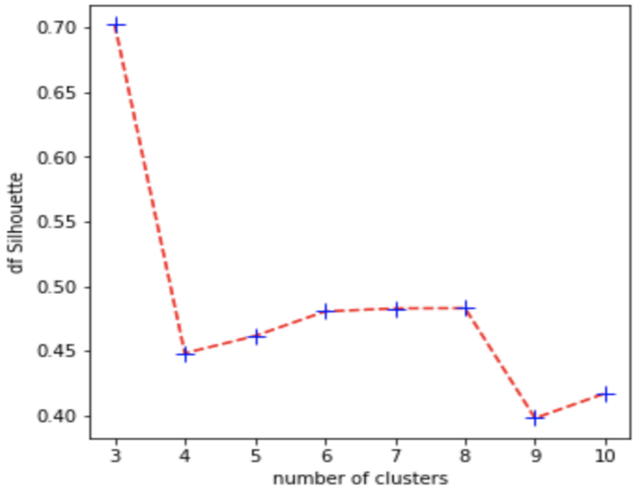


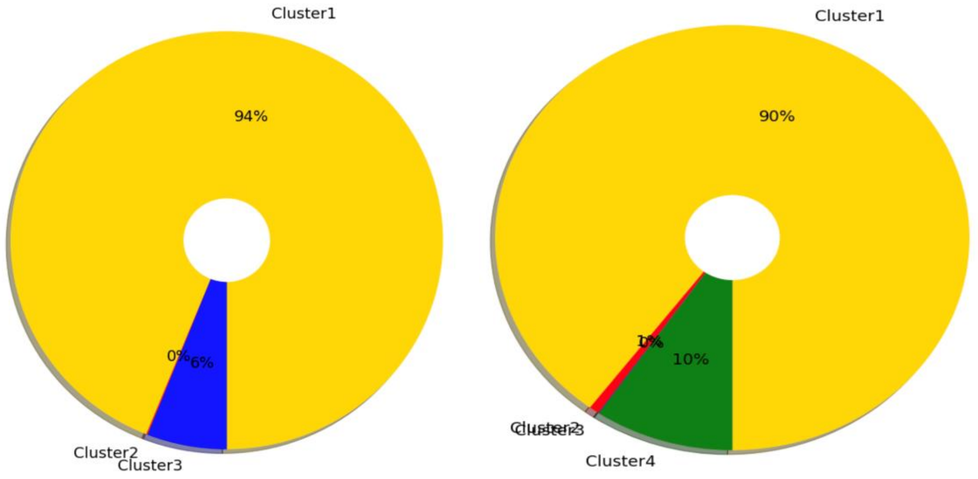
Figure 2: Silhouette coefficient calculation with different cluster numbers

1- Table listing features and their values

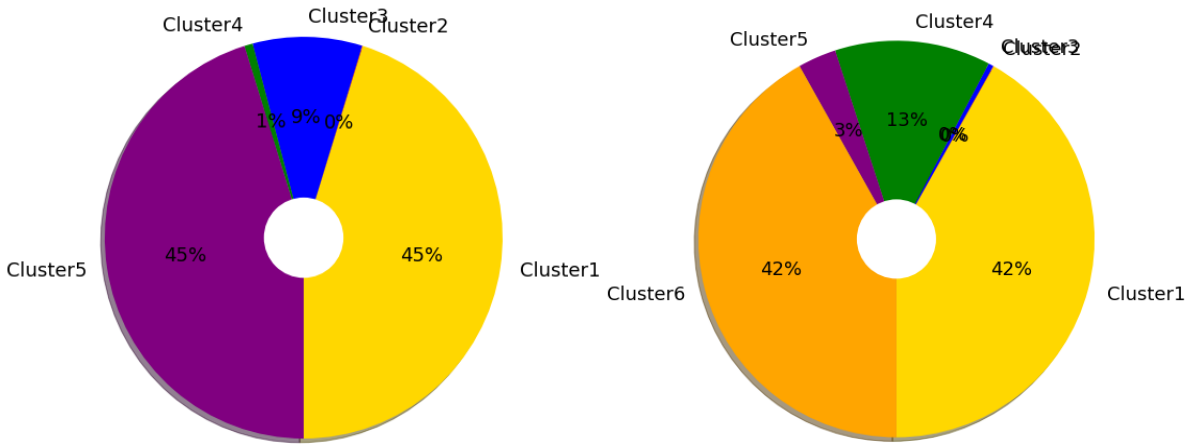
At the below table, features and their clusters by k-means algorithm (k=4) are listed.

#	feature name	description	type	overall avg./ mode	cluster 1	cluster 2	cluster 3	cluster 4
	step	maps a unit of time in the real world	numeric	243.397	1117466	463432	10474462	142065
	amount	amount of the transaction in local currency	numeric	179861.9	2648758	9141	111705	805
	nameOrig	customer who started the transaction	nominal	C1999539787	692293	690546	699864	687706
	oldbalanceOrg	initial balance before the transaction	numeric	833883.1	2657750	985	69	111605
	newbalanceOrig	new balance after the transaction	numeric	855113.6	2707346	842	59	62162
	nameDest	customer who is the recipient of the transaction	nominal	C1286084959	689529	690622	697683	692575
	oldbalanceDest	initial balance recipient before the transaction	numeric	1100701.66	2488963	19153	1099	261194
	newbalanceDest	new balance recipient after the transaction	numeric	1224996.39	2481277	19568	1350	268214

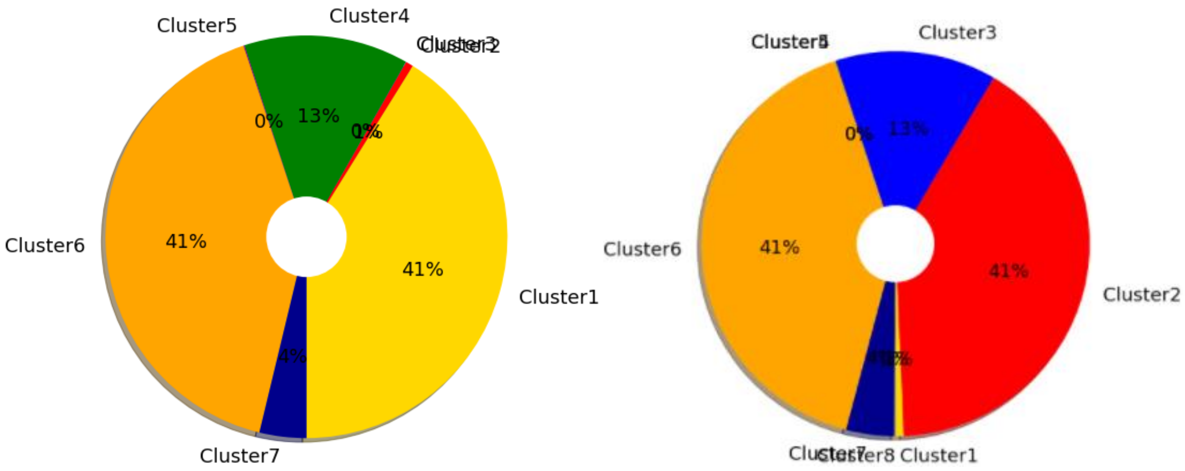
2- Pie charts showing the instance distributions of each cluster as percentages



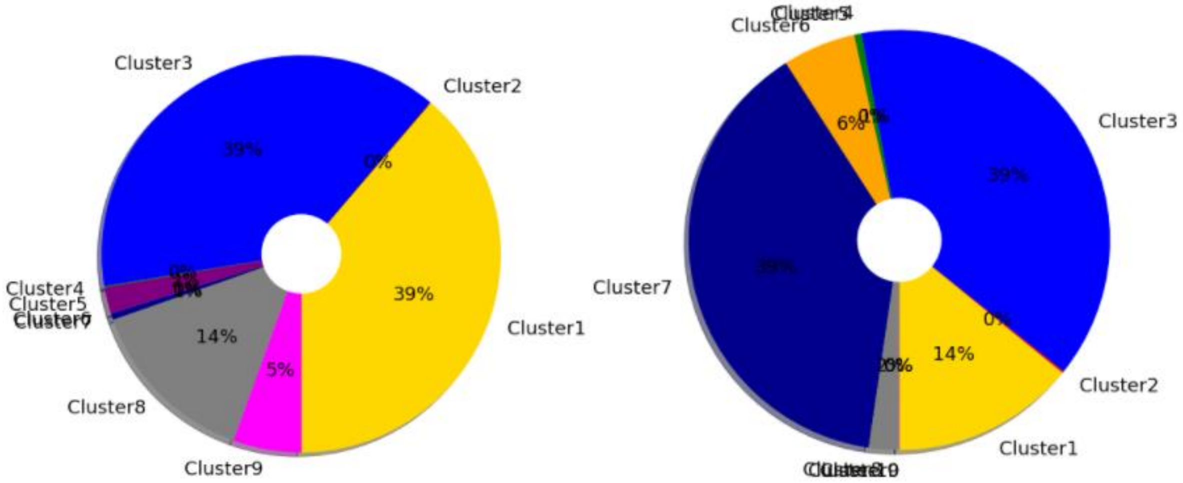
Graph 1 and 2: Pie chart of instance distribution when k=3 and k=4



Graph 3 and 4: Pie chart of instance distribution when k=5 and k=6



Graph 5 and 6: Pie chart of instance distribution when k=7 and k=8



Graph 7 and 8: Pie chart of instance distribution when k=9 and k=10

3- Table for evaluation of clustering experiments

At the below table, instances and their clusters by k-means algorithm with different k values are listed.

Experiment	# of clusters	avg. number of instances in clusters	std.dev.	SSE	NMI	Silhouette Value	RI
1	k=3	0:2602389 1:2863 2:165157	0.47	4.7	0.00035	0.70	-0.0032
2	k=4	0: 2486468 1: 19187 2: 1182 3: 263572	0.88	3.39	0.000177	0.44	-0.00209
3	k=5	0: 1252392 1: 1180 2: 246299 3: 18738 4: 1251800	1.90	2.76	6.61e-05	0.46	-0.000292
4	k=6	0: 1158884 1: 1039 2: 11616 3: 352473 4: 86594 5: 1159803	2.30	2.21	0.00011	0.48	-0.000414
5	k=7	0: 1139539 1: 16878 2: 506 3: 366760 4: 2439 5: 1139916 6: 104371	2.36	1.76	0.000127	0.48	-0.000440
6	k=8	0: 18931 1: 1132563 2: 371895 3: 920	1.54	1.96	0.000131	0.48	-0.000447

		4: 216 5: 1132195 6: 110247 7: 3442					
7	k=9	0: 1074103 1: 885 2: 1074747 3: 3010 4: 59599 5: 210 6: 14768 7: 390814 8: 152273	2.66	1.37	0.000153	0.39	–0.000486
8	k=10	0: 390756 1: 3783 2: 1070432 3: 15489 4: 498 5: 155246 6: 1069712 7: 63065 8: 1329 9: 99	2.33	1.26	0.0001615	0.41	–0.000493

4. Our Inferences and Results

In this section, we analyzed our dataset with K-Means clustering. We are testing the clustering algorithm with different k(3,10) values then we draw a plot for k(3,10) values to SSE values. While using the k-means clustering algorithm, we used the Elbow method to select an optimum k number. The elbow method is very simple and common; experiments with different values of k and takes the sum of squared errors. The plot has an elbow-like shape, where the break is the shape of an elbow, this place indicates the optimum k to be selected. The higher the value of k, the closer the cluster centers are to each other. After a point, the development of the model will decrease and will be the most optimum value for the elbow point and the k value.
According to the plot, the optimum k value is 4 according to the k-means clustering model. We obtained 8 experiments by changing the number of clusters between 3 and 10. We saw that each time we increased the number of clusters, the number of standard deviations increased, but the SSE value decreased. We see that the NMI value varies between 0.00015-0.00030 for all data according to the changing k values. In a sense, NMI tells us how much the uncertainty about class labels decreases when we know the cluster labels.

So, we can say that the correlation between the values is low. Rand index values are quite low. This shows that the success of the model is not high. The silhouette score ranges from -1 to 1. If the score is 1, the cluster is dense and well separated from the other clusters. A value close to 0 represents clusters that overlap with samples very close to the decision boundary of neighboring clusters. In our trials, we obtained the highest silhouette score (0.70) at k=3. In other words, the experiment in which the data is most consistently distributed to the clusters is the experiment where the k value is 3. The lowest score is the experiment with 0.39 and k=9.