CHITKARA UNIVERSITY

FRAUD SLEUTH

ONLINE PAYMENT FRAUD DETECTION

CHITKARA UNIVERSITY

GROUP MEMBERS: -

2210990365- HARLEEN KAUR

2210990366- HARLEEN KAUR

2210990421-ISHIKA

2210990422- ISHIKA BEDI

CHITKARA UNIVERSITY

Introduction

1.1 BACKGROUND

CHITKARA UNIVERSITY

A corresponding increase in online payment fraud has coincided with the recent explosion in online transactions, which has been made possible by the ease of e-commerce platforms. Financial losses are incurred by both customers and businesses due to fraudsters taking advantage of weaknesses in payment systems. That's why it's critical to set up strong systems to identify and stop online payment fraud. The need for swift action to put in place strong safeguards against fraud, maintain the integrity of online commerce, and secure the interests of stakeholders is highlighted by the exponential development in digital transaction volume.

1.2 OBJECTIVE

CHITKARA UNIVERSITY

This project's main goals are diverse and include the following main objectives:

Create a Machine Learning Model: Creating and deploying a machine learning model that can accurately identify fraudulent activity in online payment transactions is the main goal. In order to identify trends suggestive of fraudulent activity, the model will be trained using transaction data from the past.

Attain High Accuracy: The project seeks to minimise false positives while achieving high accuracy rates in detecting fraudulent transactions. To do this, the model's performance metrics—precision, recall, and F1-score—must be optimised in order to balance false alarm rates and detection accuracy.

Implement Scalable Solution: The project's objective is to design a scalable solution that can be easily integrated into the current online payment systems, in addition to an accurate fraud detection model. This entails creating software components that are flexible and modular in order to handle different transactional complexity and data quantities

1.3 SIGNIFICANCE

CHITKARA UNIVERSITY

This initiative is important for a number of parties involved in the digital payment ecosystem, including as corporations, financial institutions, and consumers:

Consumer Protection: The project's goal is to protect consumers from identity theft and financial losses by identifying and stopping fraudulent transactions. This will increase their trust and confidence in online payment systems.

Business Reputation: Reducing the possibility of fraud is crucial for online companies and e-commerce platforms to preserve their good name and develop client loyalty. Businesses can show their dedication to the security and integrity of their customers by implementing efficient fraud detection systems.

Industry Integrity: The initiative advances the overarching objective of strengthening the security and integrity of the digital payment sector overall. The project contributes to strengthening the resistance of online commerce against fraudulent activities and cyber threats by detecting and resolving vulnerabilities in payment systems.

CHITKARA UNIVERSITY

Problem Definition and Requirements

2.1 PROBLEM DEFINITION

CHITKARA UNIVERSITY

The problem of machine learning-based online payment fraud detection is creating algorithms and systems that can reliably identify fraudulent transactions made through digital payment networks. This includes identifying many kinds of fraud, including account takeover, unapproved transactions, credentials theft, and other fraudulent activity. The difficulty is in quickly and accurately identifying fraudulent from valid transactions in the middle of the large number and complexity of digital transactions. Additionally, in order to stay up with the ever-evolving fraud strategies used by bad actors, the system needs to continuously adapt and enhance its detection capabilities. To reduce the risks and financial losses related to online payment fraud, the issue definition thus entails creating reliable machine learning models and integrating them into payment processing systems.

2.2 REQUIREMENTS

CHITKARA UNIVERSITY

 # Software Requirements:
Python: The project utilizes the Python programming language for its versatility, ease of use, and extensive libraries for data analysis and machine learning.
Libraries: Key libraries such as Scikit-learn, Pandas, NumPy, and Matplotlib are used for data manipulation, model development, and visualization.
IDE: Jupyter Notebook or any preferred Python Integrated Development Environment (IDE) is used for coding, experimentation, and collaboration.

 # Hardware Requirements:
The project can be executed on a standard laptop or desktop computer with sufficient RAM and processing power to handle data processing and model training tasks efficiently.

 # Dataset :
Datasetsource-https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection?select=onlinefraud.csv

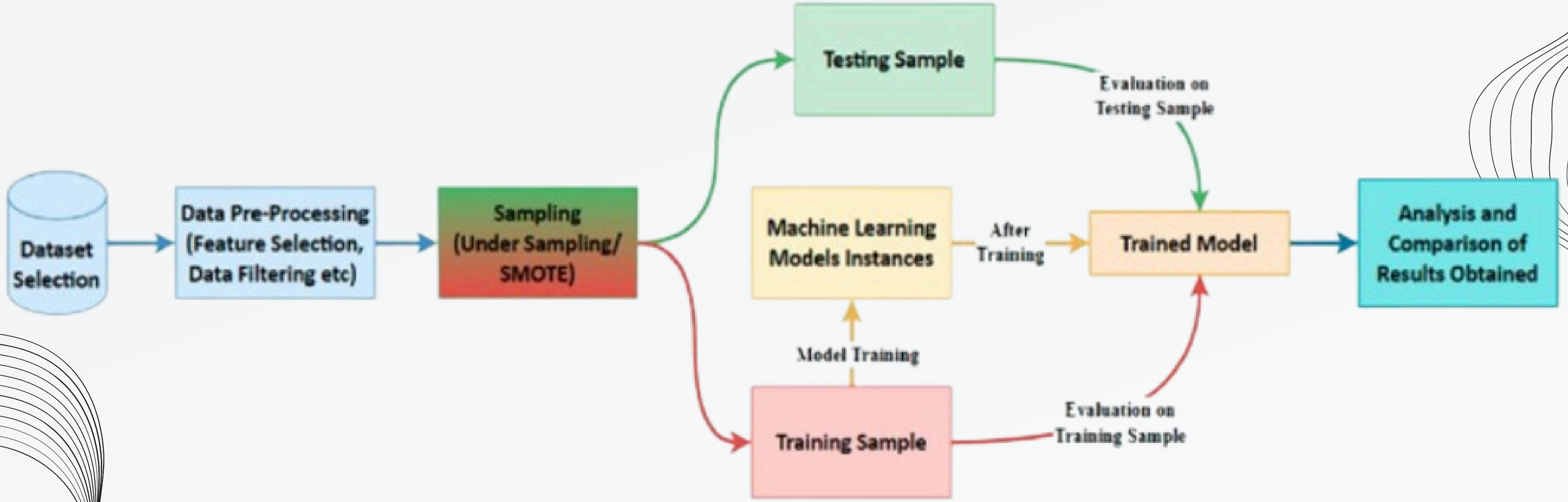
Feature	Description
step	tells about the unit of time
type	type of transaction done
amount	the total amount of transaction
nameOrg	account that starts the transaction
oldbalanceOrg	Balance of the account of sender before transaction
newbalanceOrg	Balance of the account of sender after transaction
nameDest	account that receives the transaction
oldbalanceDest	Balance of the account of receiver before transaction
newbalanceDest	Balance of the account of receiver after transaction
isFraud	The value to be predicted i.e. 0 or 1

CHITKARA UNIVERSITY

Proposed Methodology

3.1 FLOWCHART

CHITKARA UNIVERSITY



3.2 ALGORITHMS

CHITKARA UNIVERSITY

The algorithms used in the project are as follows :

```
models = {
    "LR": LogisticRegression(),
    "KNN": KNeighborsClassifier(),
    "DT": DecisionTreeClassifier(),
    "RF": RandomForestClassifier(),
    "XGB": XGBClassifier(),
    "Naive Bayes": GaussianNB(),
    "SVC": SVC()
}
```

CHITKARA UNIVERSITY

Screenshots

CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [ ]:
```
#ONLINE FRAUD DETECTION

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

df=pd.read_csv(r'C:\Users\kharl\Downloads\onlinefraud(1).csv')
```

In [143]:
```
df
```

Out[143]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	0	0
...	...	...	...	...	...	...	...	...	...	...	...
6362615	743	CASH_OUT	339682.13	C786484425	339682.13	0.00	C776919290	0.00	339682.13	1	0
6362616	743	TRANSFER	6311409.28	C1529008245	6311409.28	0.00	C1881841831	0.00	0.00	1	0
6362617	743	CASH_OUT	6311409.28	C1162922333	6311409.28	0.00	C1365125890	68488.84	6379898.11	1	0
6362618	743	TRANSFER	850002.52	C1685995037	850002.52	0.00	C2080388513	0.00	0.00	1	0
6362619	743	CASH_OUT	850002.52	C1280323807	850002.52	0.00	C873221189	6510099.11	7360101.63	1	0

6362620 rows × 11 columns

CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [147]:
```
df.shape #rows->rows,col->feature
```

Out[147]:
```
(6362620, 11)
```

In [148]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

In [149]:
```
df.describe()
```

Out[149]:

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03

CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

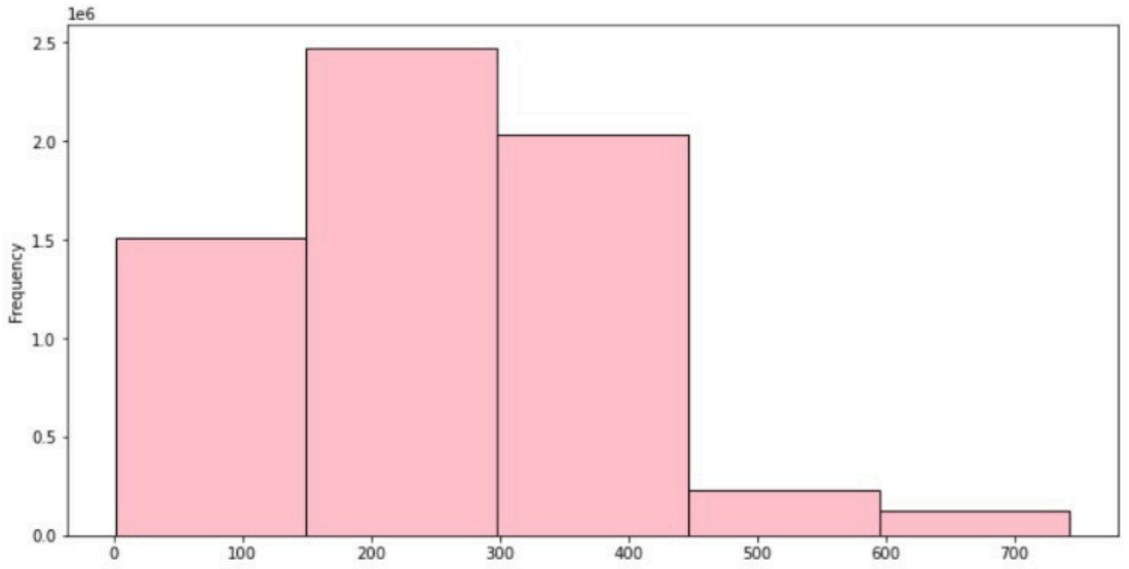
Python 3 (ipykernel)

Run Code

In [17]:
```
####EDA
```

In [161]:
```
import matplotlib.pyplot as plt
```

In [46]:
```
plt.xlabel('step')
df['step'].plot(kind='hist', bins=5, figsize=(12,6), facecolor='pink',edgecolor='black')
plt.show()
```



CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel)

Run Code

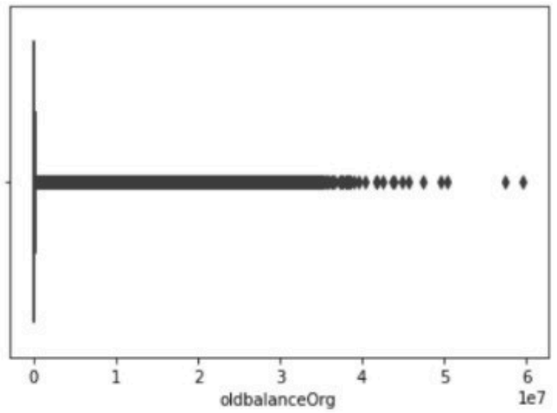
In [52]:
```
##OUTLIERS
```

In [168]:
```
sb.boxplot(df.oldbalanceOrg)
plt.show()
# plt.scatter(df['step'], df['amount'],c=['red'])

# plt.xlabel('step')
# plt.ylabel('amount')
# plt.grid(True)

# plt.show()
```

```
C:\Users\kharl\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: a minute ago (unsaved changes)

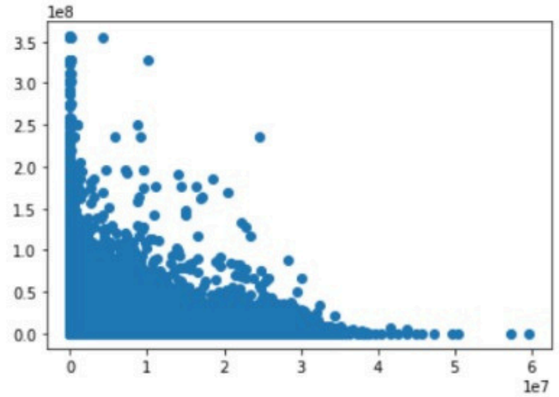
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [175]:
```
plt.scatter(df.oldbalanceOrg, df.newbalanceDest)
plt.show()
```



In [176]:
```
Q1 = np.percentile(df['amount'], 25,
                   interpolation = 'midpoint')

Q3 = np.percentile(df['amount'], 75,
                   interpolation = 'midpoint')
IQR = Q3 - Q1
```

In [177]:
```
IQR
```

Out[177]: 195331.935

In [178]:
```
upper = Q3 +1.5*IQR
lower = Q1 - 1.5*IQR
```

CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: 3 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

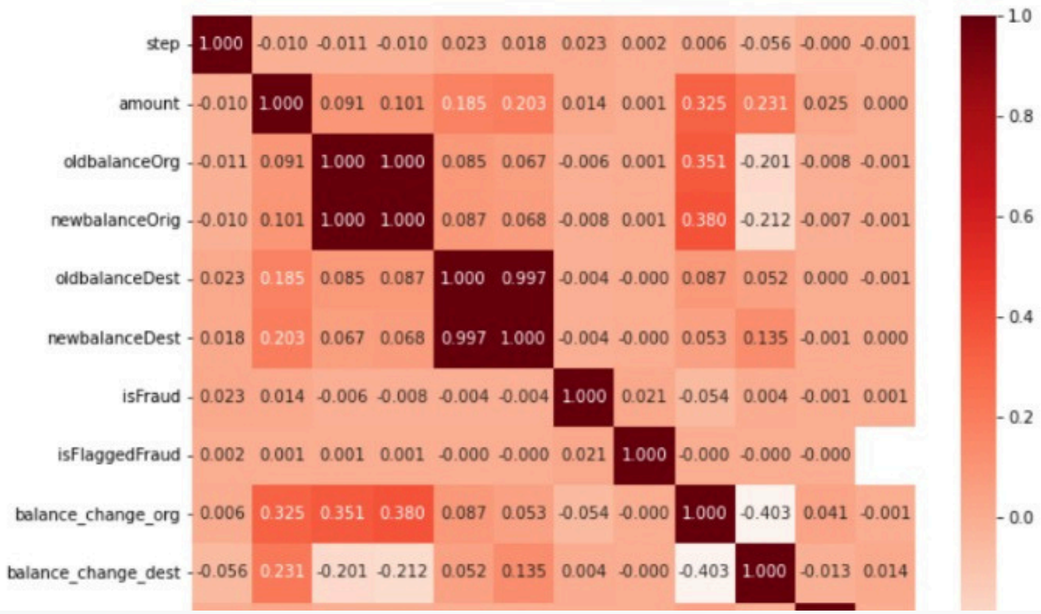
Not Trusted | Python 3 (ipykernel)

Run Code

In [224]:
```
# Assuming df is your DataFrame and you want to exclude non-numeric columns
numeric_df = df.select_dtypes(include=['float64', 'int64'])  # Select only numeric columns

# Compute correlation matrix on numeric data
corr = numeric_df.corr()

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=True, cmap='Reds', fmt=".3f")
plt.show()
```



CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: 3 minutes ago (autosaved)

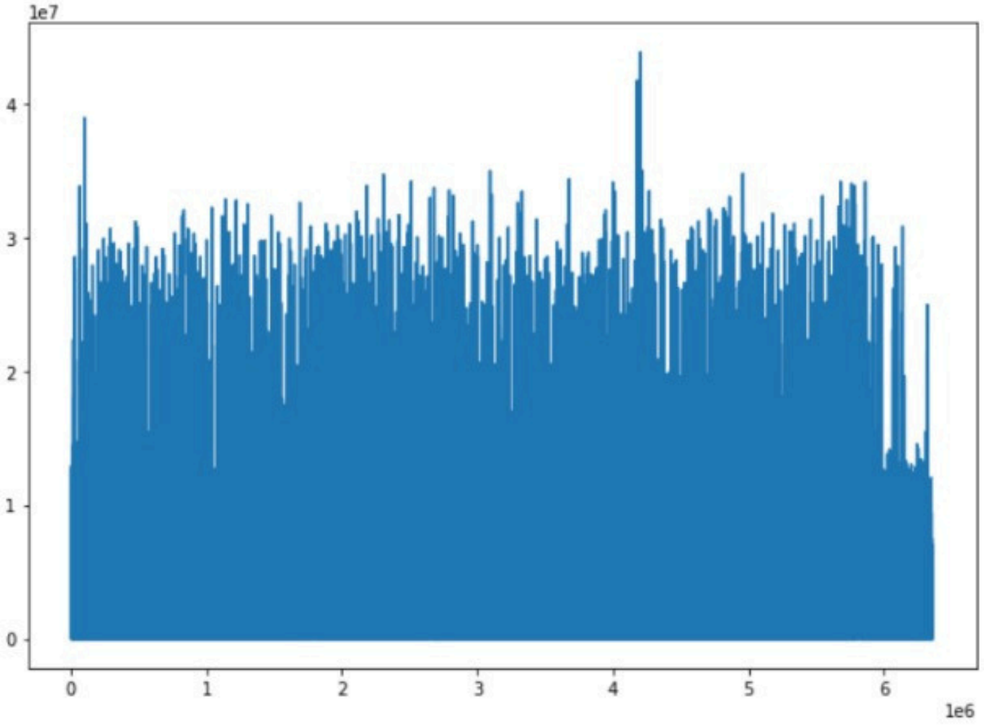
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [255]:
```
plt.figure(figsize = [10,7])
plt.plot(df.oldbalanceOrg)
plt.show()
```



In [256]:
```
df.nlargest(20, 'oldbalanceOrg')
```

CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

```
enable_categorical=False, eval_metric=None, feature_types=None,
gamma=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=None, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=None, max_leaves=None,
min_child_weight=None, missing=nan, monotone_constraints=None,
multi_strategy=None, n_estimators=None, n_jobs=None,
num_parallel_tree=None, random_state=None, ...)
```

In [316]:
```
# make predictions for test data
y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]
```

In [317]:
```
# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

```
Accuracy: 99.14%
```

In [318]:
```
rounded_y = np.round(y_pred)
unique_val = pd.DataFrame(y_test)
unique_val[0].unique()
```

Out[318]:
```
array([1, 0], dtype=int64)
```

In [319]:
```
from sklearn.metrics import multilabel_confusion_matrix
y_unique = unique_val[0].unique()
mcm = multilabel_confusion_matrix(y_test, y_pred, labels = y_unique)
mcm
```

Out[319]:
```
array([[[866,  11],
        [  4, 863]],
```

CHITKARA UNIVERSITY

jupyter Online Fraud Detection Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [314]:
```
for name, model in models.items():
    print(f'Training Model {name} \n--------------')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f'Training Accuracy: {accuracy_score(y_train, model.predict(X_train))}')
    print(f'Testing Accuracy: {accuracy_score(y_test, y_pred)}')
    print(f'Testing Confusion Matrix: \n{confusion_matrix(y_test, y_pred)}')
    print("Recall Score : ",recall_score(y_test, y_pred, pos_label='positive', average='micro'))
    print("Precision Score : ",precision_score(y_test, y_pred,  pos_label='positive',average='micro'))
    print(f"Testing F-1:", f1_score(y_test, y_pred, pos_label='positive', average='micro' )        )
    print(f"Testing F-Beta:", fbeta_score(y_test, y_pred, beta=0.5, pos_label='positive', average='micro'))
    print('-'*30)
```

```
  warnings.warn(
C:\Users\kharl\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1370: UserWarning: Note that pos_label (set to 'p
ositive') is ignored when average != 'binary' (got 'micro'). You may use labels=[pos_label] to specify a single positive class.
  warnings.warn(
C:\Users\kharl\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1370: UserWarning: Note that pos_label (set to 'p
ositive') is ignored when average != 'binary' (got 'micro'). You may use labels=[pos_label] to specify a single positive class.
  warnings.warn(

Training Accuracy: 1.0
Testing Accuracy: 0.9902522935779816
Testing Confusion Matrix:
[[863  14]
 [  3 864]]
Recall Score :  0.9902522935779816
Precision Score :  0.9902522935779816
Testing F-1: 0.9902522935779816
Testing F-Beta: 0.9902522935779816
------------------------------
Training Model XGB
--------------
```

CHITKARA UNIVERSITY

Result

CHITKARA UNIVERSITY

The results of the project are as follows:

Accuracy: 99.14%
Precision: 0.9936926605504587
Recall: 0.9936926605504587
F1-score: 0.9936926605504587
Training Accuracy: 0.9967020361342128
Testing Accuracy: 0.9936926605504587
Testing Confusion Matrix: [[871 6] [5 862]]
Testing F-Beta: 0.9936926605504587

a. Accuracy: 99.14%: Accuracy represents the overall correctness of the machine learning model in classifying transactions as either fraudulent or legitimate. It is calculated as the ratio of correctly predicted transactions (both true positives and true negatives) to the total number of transactions.

b. Precision: 0.993692660550458 7: Precision measures the proportion of true positive predictions among all positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives.

c. Recall: 0.993692660550458 7: Recall, also known as sensitivity or true positive rate, measures the proportion of actual fraudulent transactions that were correctly identified by the model. It is calculated as the ratio of true positives to the sum of true positives and false negatives.

d. F1-score: 0.993692660550458 7 The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. It considers both false positives and false negatives, making it particularly useful for evaluating models with imbalanced datasets.

e. Training Accuracy: 0.996702036134212 8
Training accuracy measures the proportion of correctly classified instances in the training dataset. It is calculated as the ratio of the number of correctly predicted instances to the total number of instances in the training set.

f. Testing Accuracy: 0.993692660550458 7
Description: Testing accuracy measures the proportion of correctly classified instances in the testing dataset. It is calculated as the ratio of the number of correctly predicted instances to the total number of instances in the testing set.

g. Testing Confusion Matrix:
A confusion matrix is a table that summarizes the performance of a classification model on a testing dataset. It presents the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

h. Testing F-Beta: 0.9936926605504587
Description: The F-beta score is the harmonic mean of precision and recall, weighted by the beta parameter. It provides a balance between precision and recall, with higher values indicating better performance.

Overall, these metrics collectively demonstrate the effectiveness and robustness of the machine learning model in accurately detecting and preventing online payment fraud. The high accuracy, precision, recall, and F1-score validate the success of the project in achieving its objectives and highlight the model's reliability in safeguarding digital payment systems against fraudulent activities.


Thank You!!