Configuration Manual

MSc Research Project
Data Analytics

Tanmay Shivhare
Student ID: 23189517

School of Computing
National College of Ireland

Supervisor: Dr. Anu Sahni

National College of Ireland MSc

Project Submission Sheet School of

Computing

Student Name: Mr Tanmay Shivhare

Student ID: 23189517

Programme: MSc in Data Analytics **Year:** 2024

Module: Research Project

Lecturer: Dr. Anu Sahni

Submission Due Date: 12/12/2024

Project Title: How to Utilize Bank Statements as a New Credit Scoring Method

Word Count: 1200 **Page Count:** 10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Tanmay Shivhare

Date: 12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Tanmay Shivhare
23189517

1 Introduction

This manual provides an overview of the configuration options for the Python script that analyzes bank account data. The script assumes the data is stored in an Excel file named "bank.xlsx".

2 System Specification

2.1 Hardware Configuration

Below are the required system specifications to execute the code:

- **Processor:** Intel Core i3
- **System Memory:** 250gb SSD Hard Disk
- **RAM:** 8GB

2.2 Software Configuration

The software requirements are discussed below:

- **Windows Edition:** Windows 11
- **Integrated Development Environment:** Jupyter Notebook
- **Scripting Language:** Python 3 +
- **Storage:** Local System Storage

2.3 Python Libraries

The Python libraries are installed in the Jupyter Notebook environment using the "pip install Library_Name" command.
Below are the libraries used for thesis implementation:

- **pandas:** Powerful Python library for data manipulation and analysis.
- **matplotlib.pyplot:** Popular Python library for creating static, animated, and interactive visualizations.
- **seaborn:** Built on top of matplotlib, providing high-level statistical data visualization for Python.
- **missingno:** Python library for visualizing missing data in pandas DataFrames.
- **plotly.express:** Python library for creating interactive visualizations in a concise and declarative way.
- **scikit-learn:** Open-source machine learning library for Python that supports various algorithms, classification, regression, clustering, and more.

3 Project Development

After installing the required python libraries, code is ready for execution.

Installation:

These libraries can be installed using pip:

pip install pandas matplotlib seaborn missingno plotly scikit-learn

Configuration Options

There are minimal configuration options within this script. However, some sections can be modified based on your specific data:

3.1 Input File:

- The script currently reads data from the file "bank.xlsx". Change the filename within the line:

Python

df = pd.read_excel('bank.xlsx')

3.2 Dropping Columns:

- The script drops the column "CHQ.NO." You can modify the line:

Python

df= df.drop(['CHQ.NO.'], axis=1)

to remove different columns.

Account Selection for Visualization (Optional):

- The script currently performs visualizations on specific accounts (ac1 to ac10) defined within the script. You can comment out these sections and define your own account numbers for analysis.

Feature Engineering (Optional):

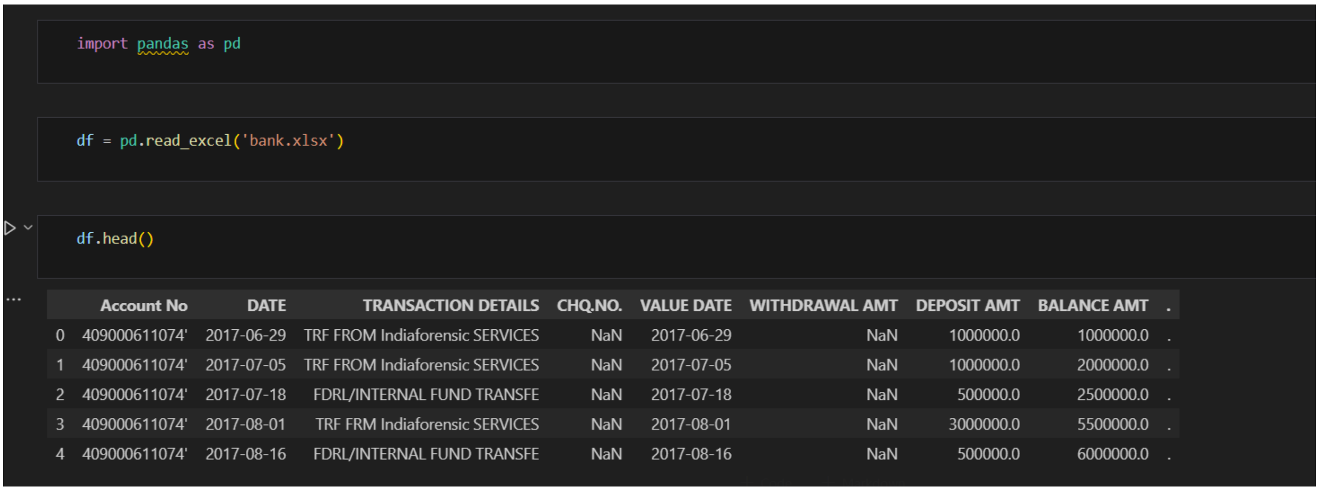
- The script currently defines a credit score based on average balance, total deposits, and total withdrawals. You can modify this section to create your own credit score formula within the credit_metrics function.

Machine Learning Model (Optional):

- The script demonstrates two machine learning models: Linear Regression and Random Forest

Regressor. You can modify these sections to explore different models or adjust hyperparameters (e.g., number of estimators for Random Forest).

3.3 Data Extraction:



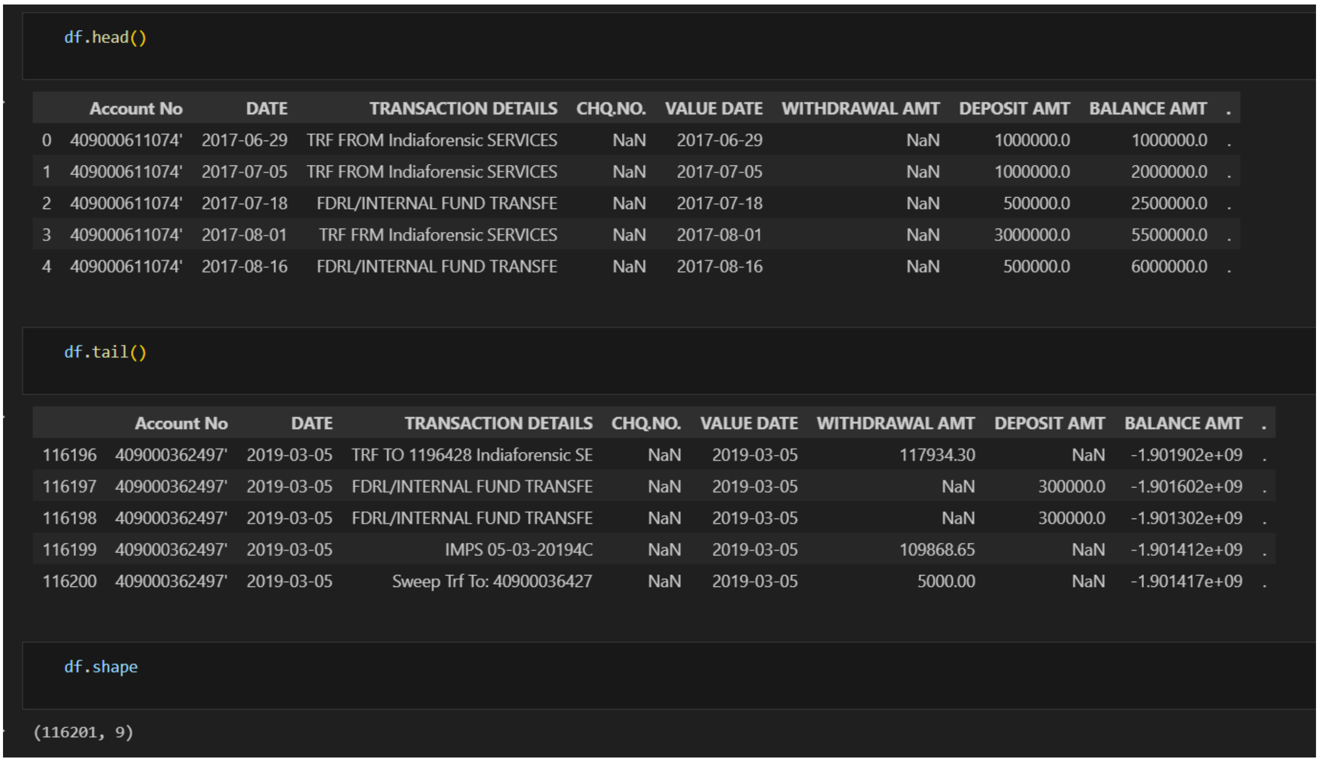
```
import pandas as pd
```

```
df = pd.read_excel('bank.xlsx')
```

```
df.head()
```

	Account No	DATE	TRANSACTION DETAILS	CHQ.NO.	VALUE DATE	WITHDRAWAL AMT	DEPOSIT AMT	BALANCE AMT	.
0	409000611074'	2017-06-29	TRF FROM Indiaforensic SERVICES	NaN	2017-06-29	NaN	1000000.0	1000000.0	.
1	409000611074'	2017-07-05	TRF FROM Indiaforensic SERVICES	NaN	2017-07-05	NaN	1000000.0	2000000.0	.
2	409000611074'	2017-07-18	FDRL/INTERNAL FUND TRANSFE	NaN	2017-07-18	NaN	500000.0	2500000.0	.
3	409000611074'	2017-08-01	TRF FRM Indiaforensic SERVICES	NaN	2017-08-01	NaN	3000000.0	5500000.0	.
4	409000611074'	2017-08-16	FDRL/INTERNAL FUND TRANSFE	NaN	2017-08-16	NaN	500000.0	6000000.0	.

DataFrame head shown below:



```
df.head()
```

	Account No	DATE	TRANSACTION DETAILS	CHQ.NO.	VALUE DATE	WITHDRAWAL AMT	DEPOSIT AMT	BALANCE AMT	.
0	409000611074'	2017-06-29	TRF FROM Indiaforensic SERVICES	NaN	2017-06-29	NaN	1000000.0	1000000.0	.
1	409000611074'	2017-07-05	TRF FROM Indiaforensic SERVICES	NaN	2017-07-05	NaN	1000000.0	2000000.0	.
2	409000611074'	2017-07-18	FDRL/INTERNAL FUND TRANSFE	NaN	2017-07-18	NaN	500000.0	2500000.0	.
3	409000611074'	2017-08-01	TRF FRM Indiaforensic SERVICES	NaN	2017-08-01	NaN	3000000.0	5500000.0	.
4	409000611074'	2017-08-16	FDRL/INTERNAL FUND TRANSFE	NaN	2017-08-16	NaN	500000.0	6000000.0	.

```
df.tail()
```

	Account No	DATE	TRANSACTION DETAILS	CHQ.NO.	VALUE DATE	WITHDRAWAL AMT	DEPOSIT AMT	BALANCE AMT	.
116196	409000362497'	2019-03-05	TRF TO 1196428 Indiaforensic SE	NaN	2019-03-05	117934.30	NaN	-1.901902e+09	.
116197	409000362497'	2019-03-05	FDRL/INTERNAL FUND TRANSFE	NaN	2019-03-05	NaN	300000.0	-1.901602e+09	.
116198	409000362497'	2019-03-05	FDRL/INTERNAL FUND TRANSFE	NaN	2019-03-05	NaN	300000.0	-1.901302e+09	.
116199	409000362497'	2019-03-05	IMPS 05-03-20194C	NaN	2019-03-05	109868.65	NaN	-1.901412e+09	.
116200	409000362497'	2019-03-05	Sweep Trf To: 40900036427	NaN	2019-03-05	5000.00	NaN	-1.901417e+09	.

```
df.shape
```

```
(116201, 9)
```

3.4 Preprocessing:

Here are the key preprocessing steps identified in the provided Python script:

1. Data Loading and Initial Inspection:

- Reading the Excel file: The script uses pd.read_excel() to load the data from the "bank.xlsx" file into a pandas DataFrame.

- Initial exploration: The df.head(), df.tail(), df.shape, df.info(), and df.describe() methods are used to get a basic understanding of the data.

2. Data Cleaning and Handling Missing Values:

- Dropping unnecessary columns: The df.drop() method is used to remove the "CHQ.NO." column, which is not relevant for the analysis.

- Handling missing values: The df.dropna() method is used to remove rows with missing values. However, this is a simple approach and might not be suitable for all scenarios. More sophisticated techniques like imputation or modeling-based approaches could be considered.

3. Data Type Conversion and Formatting:

- Date conversion: The pd.to_datetime() method is used to convert the "DATE" column to a datetime format.

- Numeric conversion: The pd.to_numeric() method is used to convert the "WITHDRAWAL AMT", "DEPOSIT AMT", and "BALANCE AMT" columns to numeric data types.

4. Feature Engineering:

- Grouping and aggregation: The df.groupby() and aggregation functions (count, mean, sum) are used to calculate metrics like total transactions, average balance, total deposits, and total withdrawals for each account.

- Creating a new feature: A "credit_score" feature is created based on a formula involving average balance, total deposits, and total withdrawals.

5. Data Preparation for Machine Learning:

- Splitting data: The train_test_split() function is used to split the data into training and testing sets.

- Feature scaling (implicit): While not explicitly mentioned, the nature of the features (numeric values) might not require explicit scaling. However, depending on the specific model and data distribution, scaling might be beneficial.

Additional Considerations:

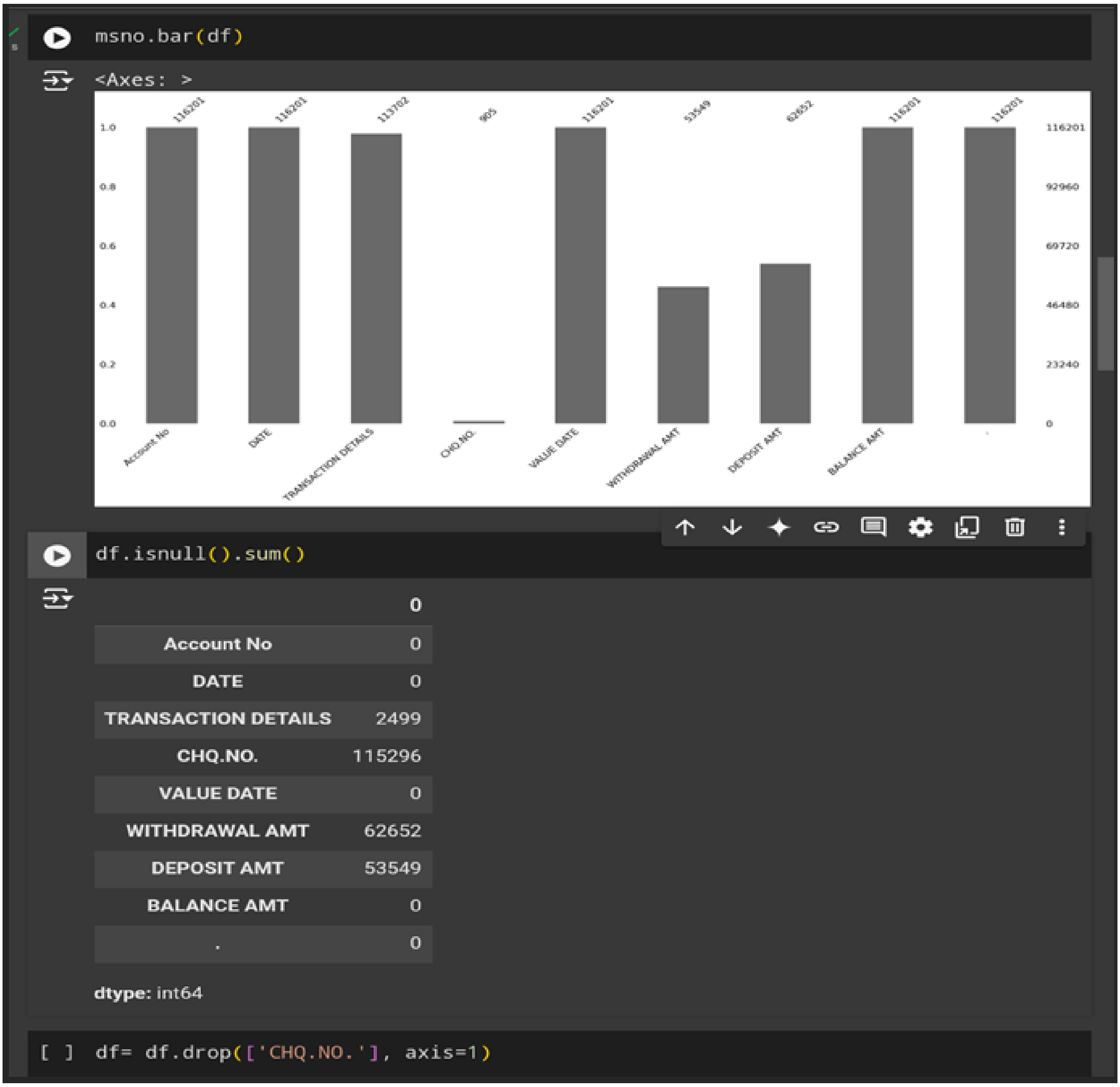
- Outlier Detection and Handling: The script does not explicitly address outlier detection and handling. Depending on the data, outliers might need to be identified and handled appropriately (e.g., by capping, flooring, or removing them).

- Feature Selection: The script uses all available features for the machine learning models. Feature selection techniques could be used to identify the most relevant features and improve model performance.

- Model Evaluation: The script evaluates the models using the score() method. More comprehensive evaluation metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared could be used for a deeper analysis.

3.5 Data Visualization

Stock data visualization:



```
msno.bar(df)
```

```
<Axes: >
```

```
df.isnull().sum()
```

	0
Account No	0
DATE	0
TRANSACTION DETAILS	2499
CHQ.NO.	115296
VALUE DATE	0
WITHDRAWAL AMT	62652
DEPOSIT AMT	53549
BALANCE AMT	0
.	0

dtype: int64

```
[ ] df= df.drop(['CHQ.NO.'], axis=1)
```

Figure 1. Mnso bar

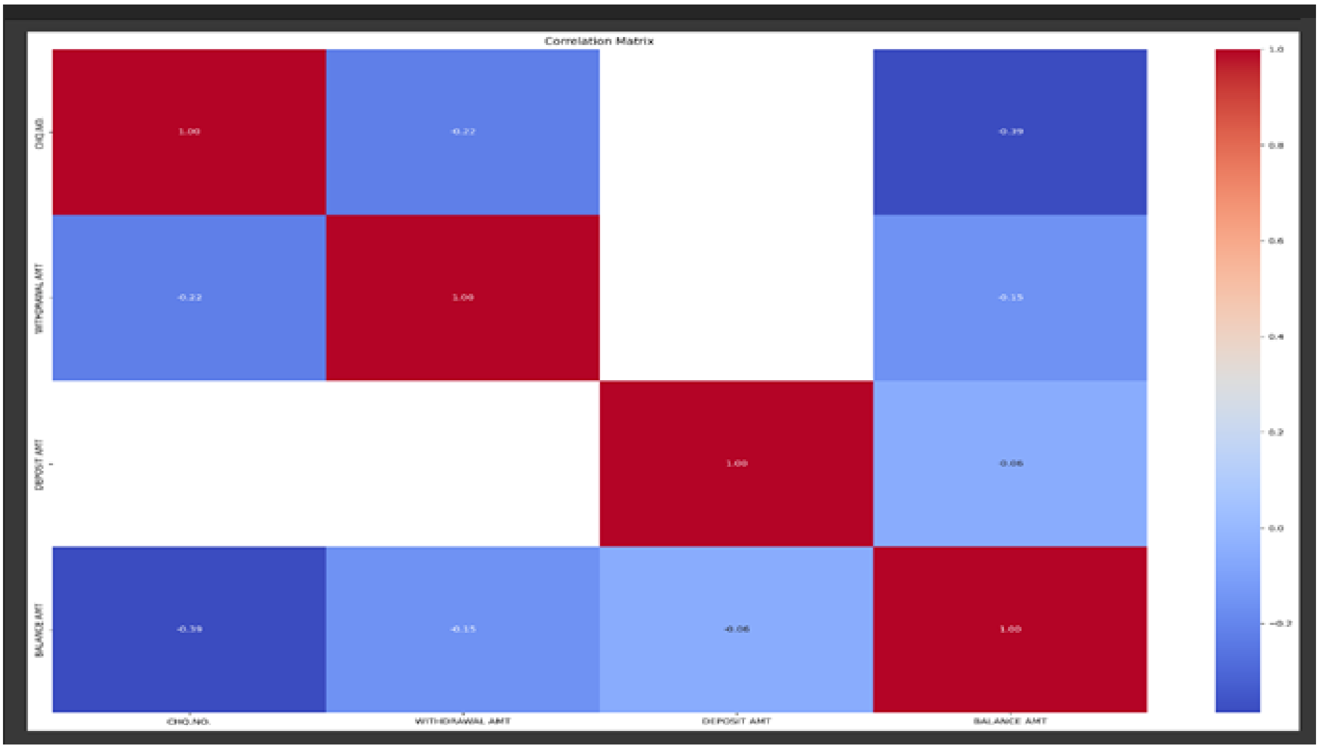
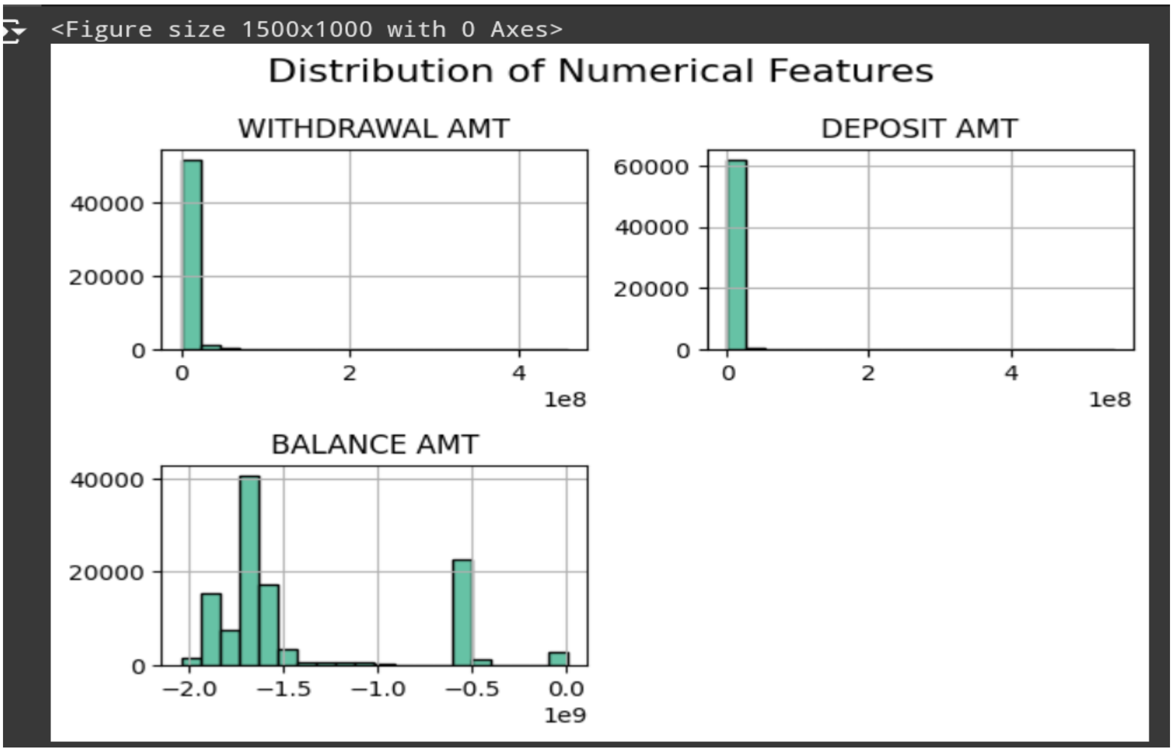


Figure 2. correlation matrix for numerical variable



```
<Figure size 1500x1000 with 0 Axes>
```

Figure 3. Distribution of Numerical Features

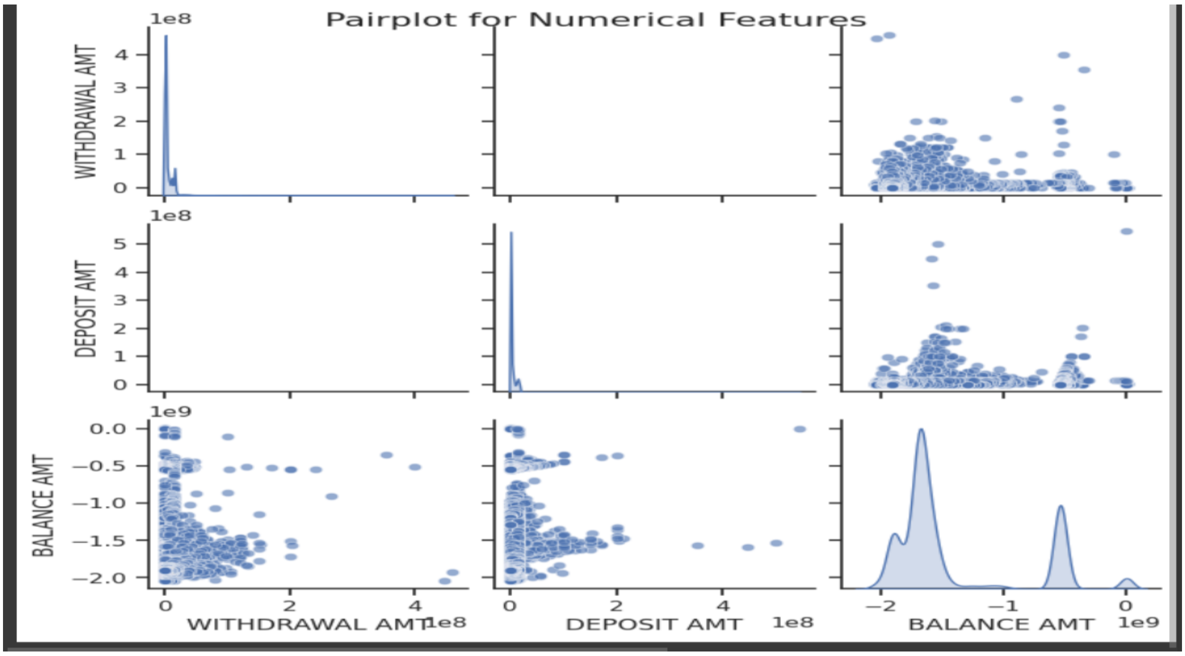


Figure 4. Pairplot for Numerical Features

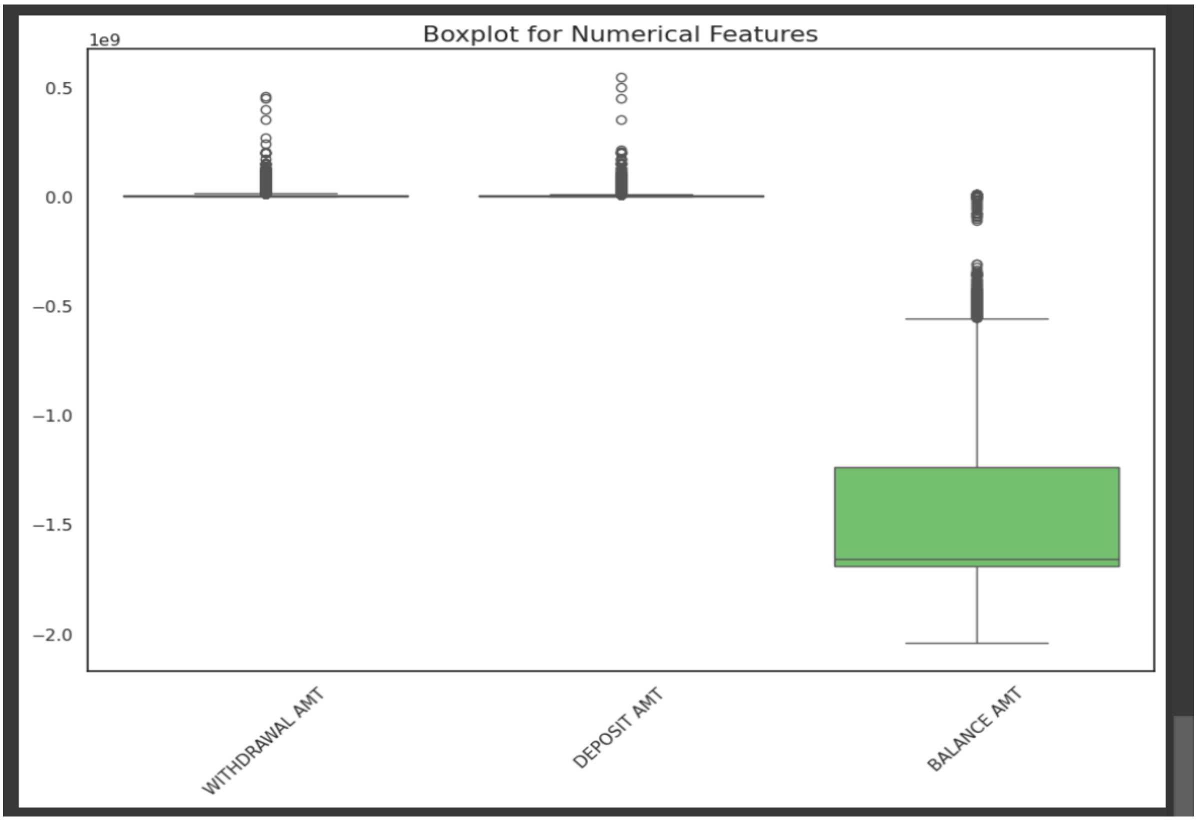
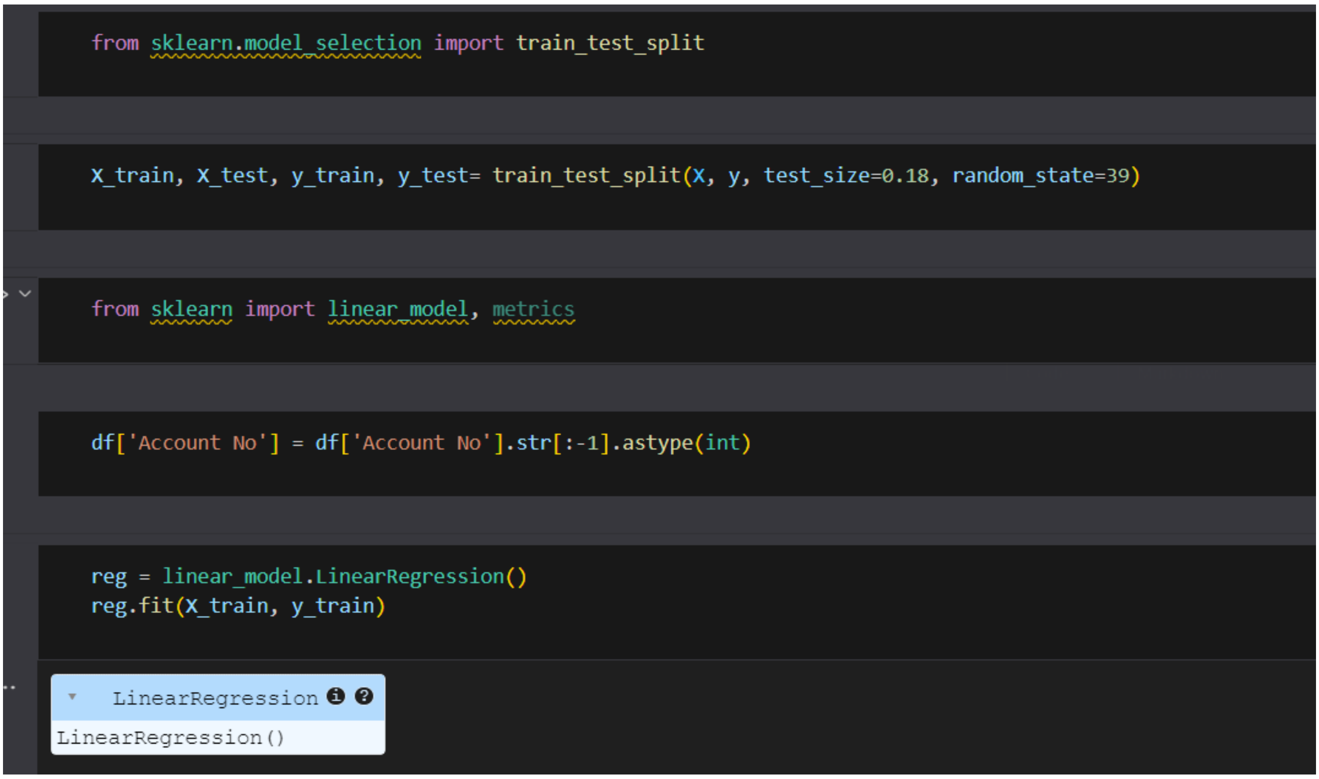


Figure 5. Boxplot for Numerical Features

3.6 Machine Learning Model:



```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.18, random_state=39)
```

```
from sklearn import linear_model, metrics
```

```
df['Account No'] = df['Account No'].str[:-1].astype(int)
```

```
reg = linear_model.LinearRegression()
reg.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

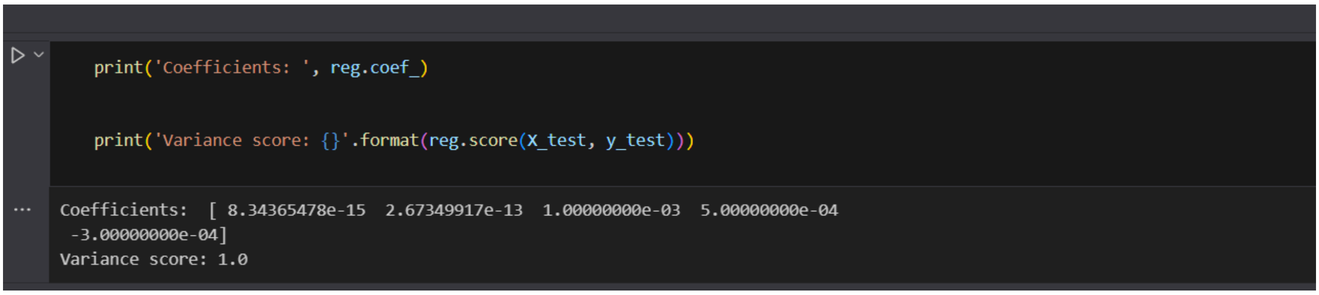
- Model training: Information about the training process, including the chosen model (e.g., Linear Regression, Random Forest Regressor), hyperparameters, and training time.
- Model evaluation: Model performance metrics like R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE) on the training and testing sets.
- Model coefficients: The coefficients of the linear regression model or the feature importance scores from the random forest model.

3.7 Credit Score Predictions:

- Predicted credit scores for each account based on the trained model.

Specific Outputs for the Provided Account Numbers:

- Visualizations of transactions, deposits, withdrawals, and balance over time for each account.
- Summary statistics for each account.
- Potential anomalies or unusual patterns in the transaction data.



```
print('Coefficients: ', reg.coef_)


print('Variance score: {}'.format(reg.score(X_test, y_test)))
```

```
Coefficients:  [ 8.34365478e-15  2.67349917e-13  1.00000000e-03  5.00000000e-04
 -3.00000000e-04]
Variance score: 1.0
```