Planning Logic – Online Payments Fraud Detection using Machine Learning

This document briefly explains the sprint-wise planning logic of the Online Payments Fraud Detection project. The planning follows an Agile-based sprint model where each sprint focuses on a specific phase of the system.

Sprint 1: Data Collection and Data Preparation

```
df
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.0	0.00	0.0	0.0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.0	0.00	0.0	0.0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.0	0.00	1.0	0.0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.0	0.00	1.0	0.0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.0	0.00	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
56198	9	CASH_OUT	16024.60	C1088493558	442118.00	426093.40	C1084323592	5818.0	8074.67	0.0	0.0
56199	9	PAYMENT	20502.92	C410885495	3073.00	0.00	M1731153077	0.0	0.00	0.0	0.0
56200	9	CASH_IN	175858.36	C702220078	290164.69	466023.05	C65594254	24083.0	0.00	0.0	0.0
56201	9	PAYMENT	2955.89	C1632500548	466023.05	463067.17	M363811903	0.0	0.00	0.0	0.0
56202	9	PAYMEN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

56203 rows × 11 columns

Epic 1: Data Collection

USN1 – Gathering Data :
The dataset is collected from a simulated real-world online transaction dataset, which represents mobile money transactions.

USN2 – Loading Data :
The collected dataset is loaded into the system using Python libraries such as Pandas for further analysis.

Epic 2: Data Preparation

USN3 – Handling Missing Values :
The dataset is checked for missing or null values, and appropriate cleaning techniques are applied.

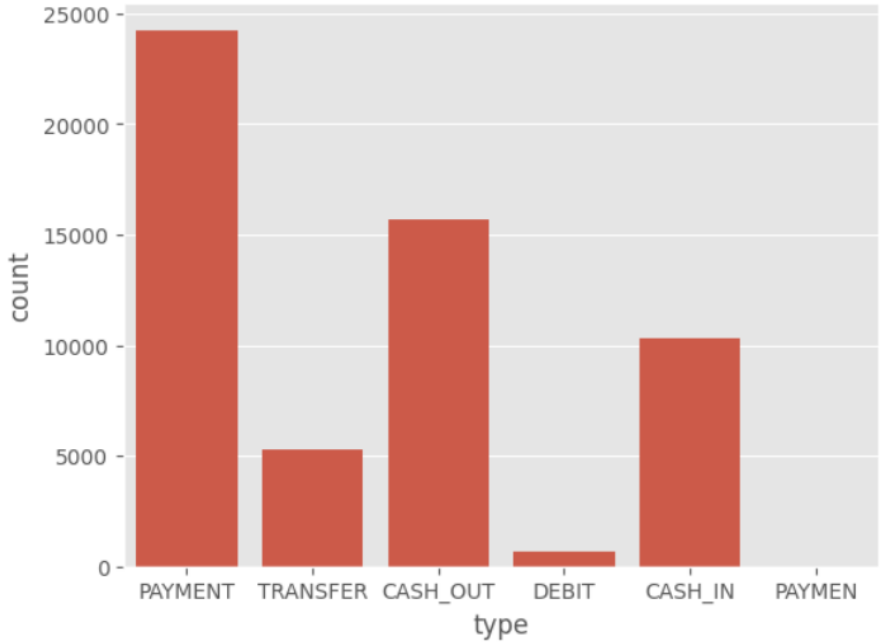
USN4 – Creating Fields :
Relevant features required for fraud detection such as transaction amount and balance fields are selected.

USN5 – Handling Data Inconsistencies :
Inconsistencies such as incorrect balance updates are handled to improve model learning.

Sprint 2: Data Visualization

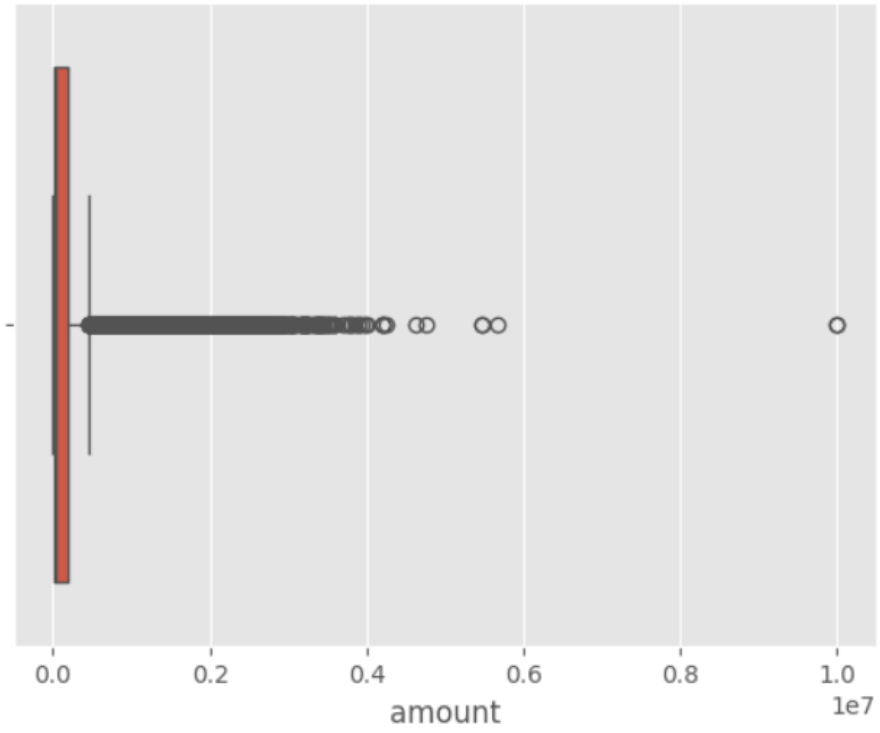
```
sns.countplot(data=df,x='type')
```

```
<Axes: xlabel='type', ylabel='count'>
```




```
sns.boxplot(data=df,x='amount')
```

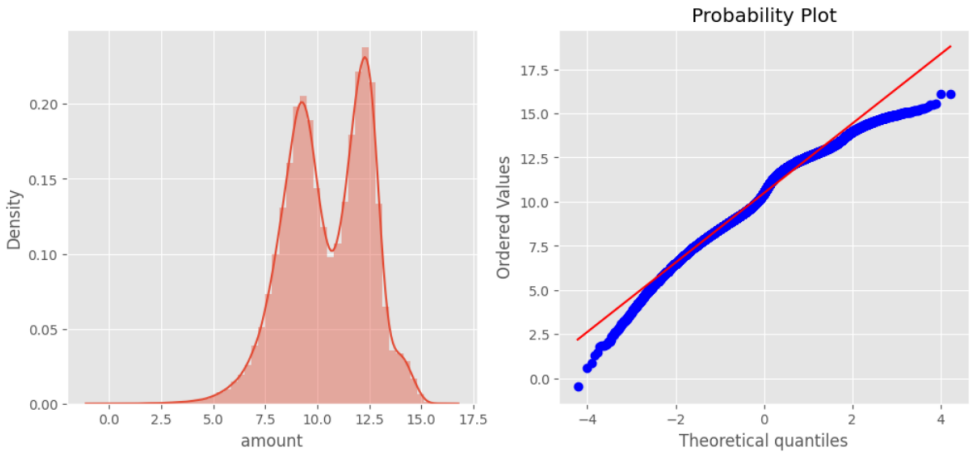
```
<Axes: xlabel='amount'>
```



```
sns.boxplot(data=df,x='isFraud',y='step')
```

```
<Axes: xlabel='isFraud', ylabel='step'>
```


```
transformationplot(np.log(df[df['amount'] > 0]['amount']))
```



Epic 3: Data Visualization

USN6 – Bar Chart Visualization :
Bar charts are used to analyze transaction types and fraud distribution.

USN7 – Pie Chart Visualization :
Pie charts help in understanding the proportion of fraudulent and legitimate transactions.

Sprint 3: Model Development and Evaluation

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test1, y_test_predict5)
print(cm)
```

```
[[   21    11]
 [    1 52069]]
```

```
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test1,y_test_predict5))
```

```
              precision    recall  f1-score   support

           0       0.78      0.58      0.67        12
           1       1.00      1.00      1.00     11229

    accuracy                           1.00     11241
   macro avg       0.89      0.79      0.83     11241
weighted avg       1.00      1.00      1.00     11241
```

In this sprint, multiple machine learning models such as Decision Tree, Random Forest, and XGBoost are trained and evaluated. XGBoost is selected due to its superior performance.

Sprint 4: Application Development and Deployment

The trained model is integrated into a Flask web application. The application allows users to input transaction details and receive fraud predictions in real time. The final system is deployed on Render cloud platform.

Online Payment Fraud Detection

4

CASH_OUT

10000

30000

0

0

0

Detect Fraud

Prediction Result

🚨 **Fraud Transaction**

Confidence: 99.55%

Check Another Transaction