Fraud Detection Using ML

Yash Wadgave

Online Payment Fraud Transaction Detection

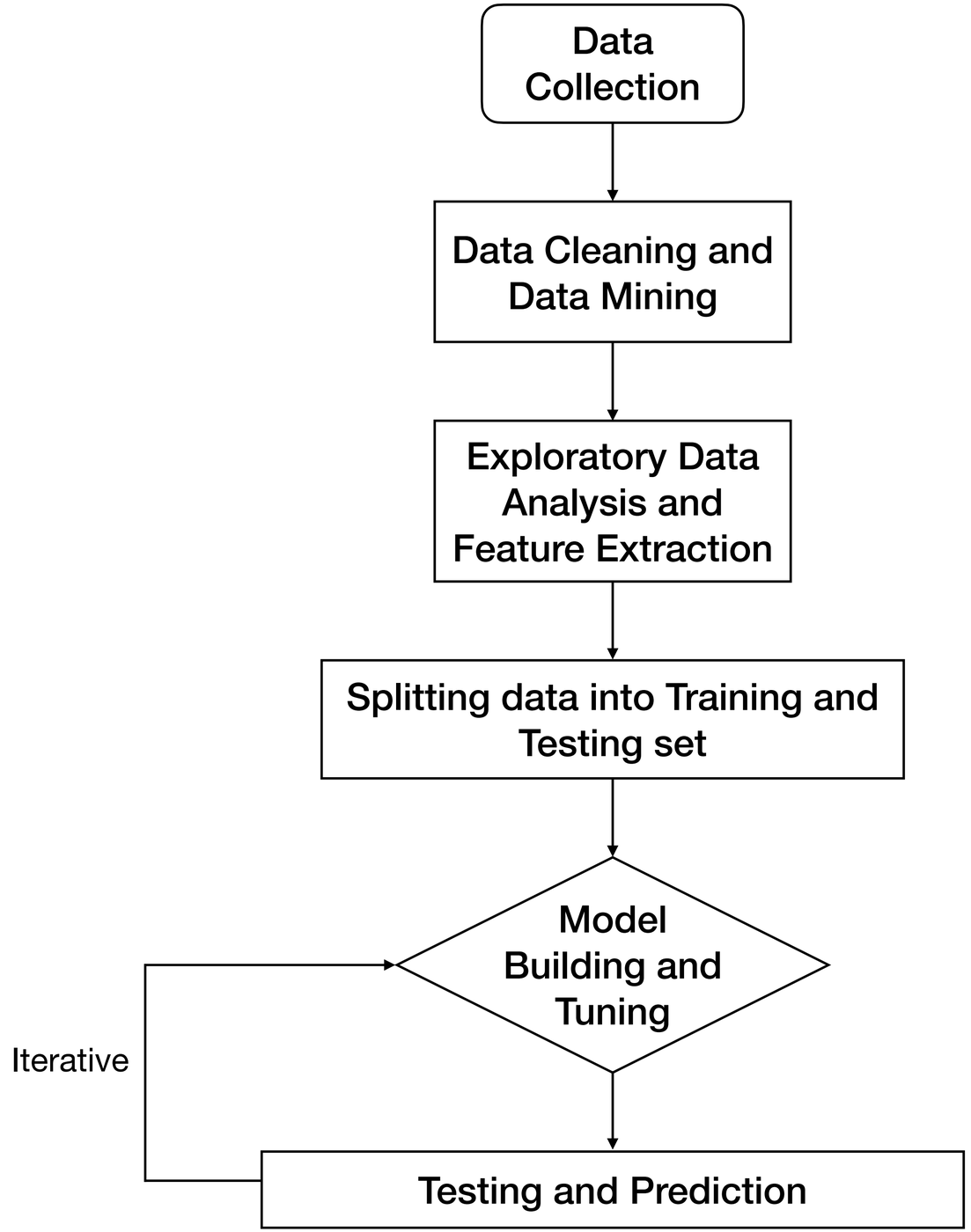
Develop a model for predicting online fraudulent transactions and use insights from the model to develop and actionable plan.

Data

Used a Fraudulent Transaction detection dataset from Kaggle which contained 6362620 rows and 11 columns.

Workflow

Fetched the data using pandas and performed data cleaning on it. Following through it extracted relevant features and performed Exploratory Data Analysis on cleaned data. Split the data into training and testing set and implement Machine Learning Algorithm on it. Then noted down all of its metrics and tested the model using the testing dataset.


Data Collection
Data Cleaning and Data Mining
Exploratory Data Analysis and Feature Extraction
Splitting data into Training and Testing set
Model Building and Tuning
Iterative
Testing and Prediction

Tools

- Python
- Pandas
- Numpy
- Matplotlib
- Seaborn
- Scikit Learn (Machine Learning Library)

Analysing Data

Cleaning data and analysing there datatypes and eliminating null values

In [2]:

```
data = pd.read_csv("PS_20174392719_1491204439457_log.csv")
data.head()
```

Out[2]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Analysing the data

In [3]:

```
data.info()
```

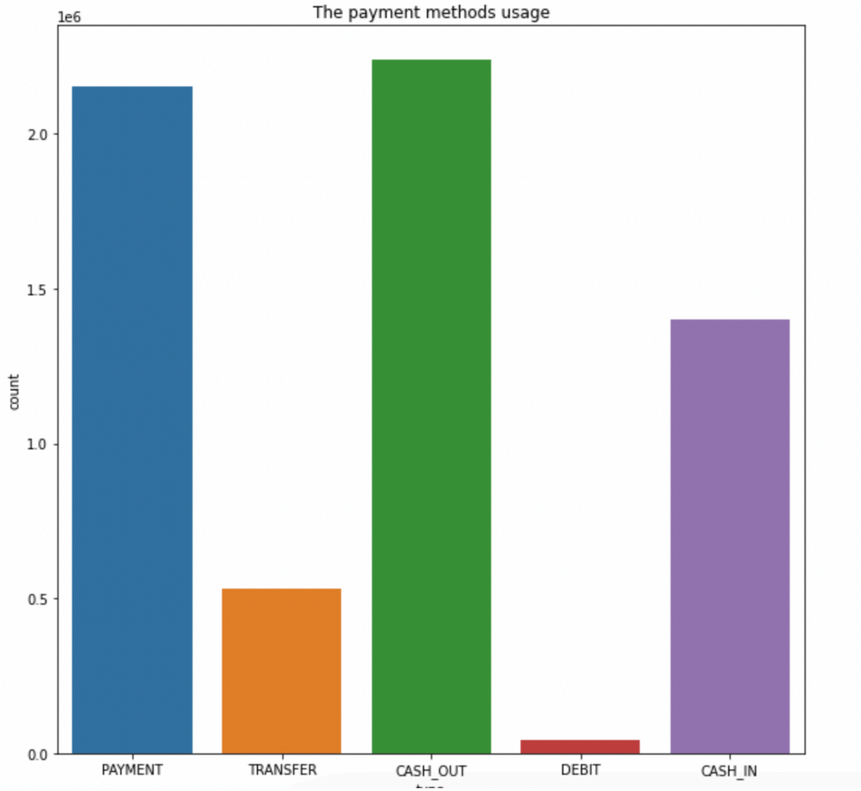
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   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
```

Exploratory Data Analysis

Analysing each feature in the dataset and plotting it with the target variable to better know the relation between them.

In [12]:
```
plt.figure(figsize=(10,10))
sn.countplot(x='type',data=data)
plt.title("The payment methods usage")
```

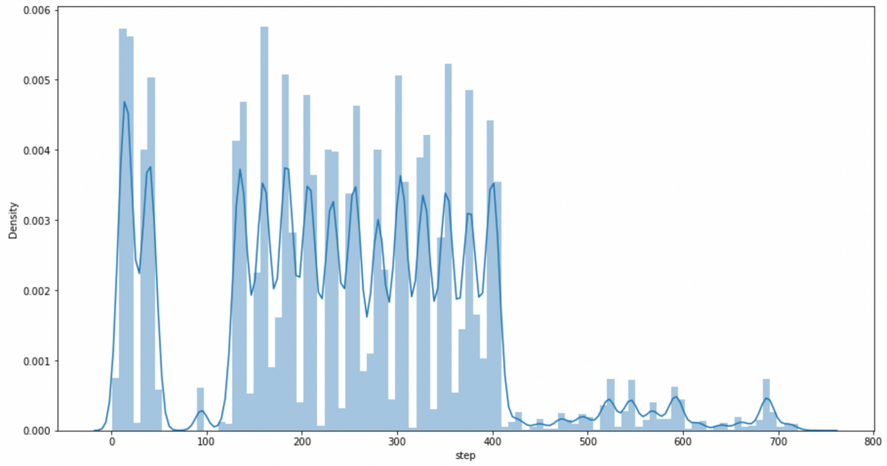
Out[12]:
```
Text(0.5, 1.0, 'The payment methods usage')
```



In [17]:
```
plt.figure(figsize=(15,8))
sn.distplot(data.step, bins=100)
```

```
/opt/homebrew/Caskroom/miniforge/base/envs/tensorflow/lib/python3.9/site-packages/seaborn/distributions.py:2619: Future
is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[17]:
```
<AxesSubplot:xlabel='step', ylabel='Density'>
```



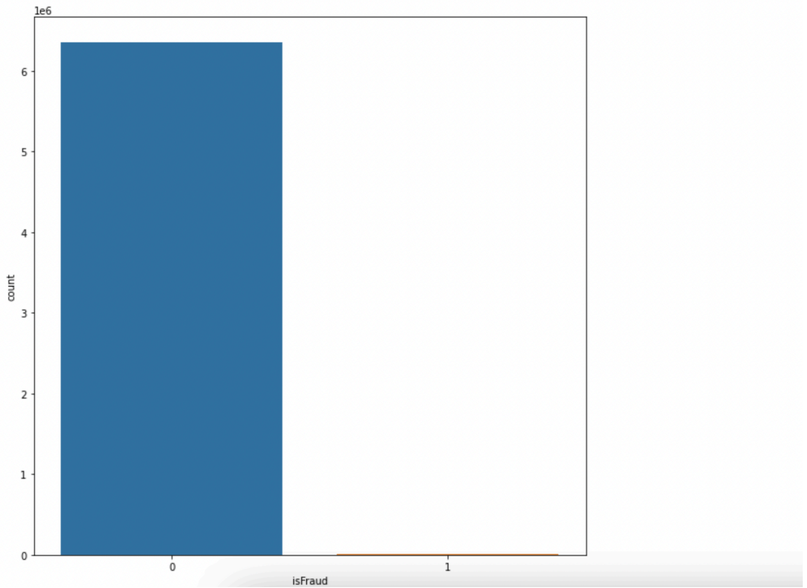
The dataset was highly imbalanced

Balanced the dataset so that the classifier does not get
biased towards the prediction and affect the accuracy of the and its predictions.

Counting the values of Fraudulent Transactions

In [13]:
```
plt.figure(figsize=(10,10))
sn.countplot(x='isFraud',data=data)
```

Out[13]:
```
<AxesSubplot:xlabel='isFraud', ylabel='count'>
```



This indicates the dataset is highly imbalanced

In [14]:
```
data.groupby('isFraud').count()
```

Out[14]:

isFraud	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest
0	6354407	6354407	6354407	6354407	6354407	6354407	6354407	6354407	6354407
1	8213	8213	8213	8213	8213	8213	8213	8213	8213

In [15]:
```
data.isFraud.value_counts()
```

Out[15]:
```
0    6354407
1       8213
Name: isFraud, dtype: int64
```

Balancing the Dataset

In [22]:
```
df_fraud = data[data['isFraud'] == 1]
df_fraud.shape
```

Out[22]:
```
(8213, 10)
```

In [23]:
```
df_notfraud = data[data['isFraud'] == 0]
df_notfraud.shape
```

Out[23]:
```
(6354407, 10)
```

In [24]:
```
df_notfraud_bal = df_notfraud.sample(df_fraud.shape[0])
df_notfraud_bal.shape
```

Out[24]:
```
(8213, 10)
```

In [25]:
```
df_balanced = pd.concat([df_fraud,df_notfraud_bal])
```

Model Building

- Implemented Random Forest Classifier algorithm on the dataset. Achieved an accuracy of 99.20%.

```
In [30]: from sklearn.ensemble import RandomForestClassifier

In [31]: model = RandomForestClassifier(n_estimators=100,n_jobs=3)

In [32]: model.fit(X_train,y_train)

Out[32]: RandomForestClassifier(n_jobs=3)

In [33]: model.score(X_test,y_test)

Out[33]: 0.9920876445526476
```

End Results

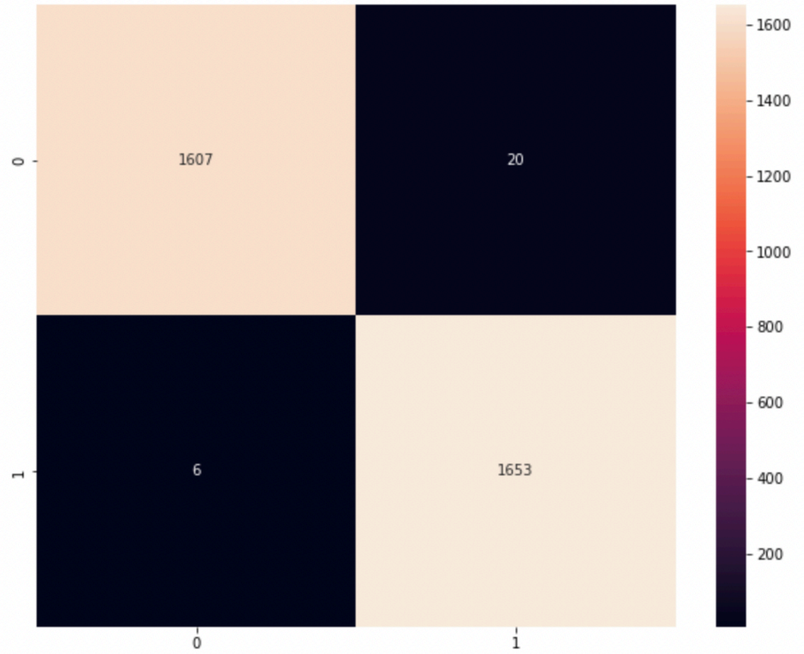
Plotted heatmap of the confusion matrix.

Printed the Classification Report.

```
In [41]: cm = confusion_matrix(y_test, y_pred)
```

```
In [49]: plt.figure(figsize=(10,8))
         sn.heatmap(cm,annot=True, fmt='d')
```

```
Out[49]: <AxesSubplot:>
```




```
In [50]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	1.00	0.99	0.99	1627
1	0.99	1.00	0.99	1659
accuracy			0.99	3286
macro avg	0.99	0.99	0.99	3286
weighted avg	0.99	0.99	0.99	3286

Motivation

Today majority of out transactions are online, we are exposed to fraudulent transaction. Machine Learning helps mitigating those risk by detecting fraudulent transaction and eliminating it.

Thank you