1_Data_Loading_and_EDA (1)

December 12, 2025

0.1 1. Data Loading and Exploratory Data Analysis

[3]:
```
import boto3
import zipfile
import io
import pandas as pd
#import matplotlib.pyplot as plt
#import seaborn as sns
import numpy as np
```

0.1.1 1.1 Data Loading

We have stored the zip file containing the data in S3, we will load it and unzip it into the Jupyter Lab environment

[1]:
```
!aws s3 cp "s3://tfm-annagm/ieee-fraud-detection.zip" ./
```

```
/bin/bash: switchml: line 1: syntax error: unexpected end of file
/bin/bash: error importing function definition for `switchml'
/bin/bash: module: line 1: syntax error: unexpected end of file
/bin/bash: error importing function definition for `module'
download: s3://tfm-annagm/ieee-fraud-detection.zip to ./ieee-fraud-detection.zip
```

[4]:
```
zip_path = "./ieee-fraud-detection.zip"
extract_to = "./input_data"

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_to)
```

0.1.2 1.2 Exploratory Data Analysis

We have loaded all the files and we see two types of .csv for training: * `train_identity.csv` * `train_transaction.csv`

We will perform the Exploratory Data Analysis to both

[2]:
```
transaction_df = pd.read_csv("./input_data/train_transaction.csv")
transaction_df.head(10)
```

[2]:

```
   TransactionID  isFraud  TransactionDT  TransactionAmt ProductCD  card1  \
0        2987000        0          86400            68.5         W  13926
1        2987001        0          86401            29.0         W   2755
2        2987002        0          86469            59.0         W   4663
3        2987003        0          86499            50.0         W  18132
4        2987004        0          86506            50.0         H   4497
5        2987005        0          86510            49.0         W   5937
6        2987006        0          86522           159.0         W  12308
7        2987007        0          86529           422.5         W  12695
8        2987008        0          86535            15.0         H   2803
9        2987009        0          86536           117.0         W  17399

   card2  card3       card4  card5  … V330  V331  V332  V333  V334 V335  \
0    NaN  150.0    discover  142.0  …  NaN   NaN   NaN   NaN   NaN  NaN
1  404.0  150.0  mastercard  102.0  …  NaN   NaN   NaN   NaN   NaN  NaN
2  490.0  150.0        visa  166.0  …  NaN   NaN   NaN   NaN   NaN  NaN
3  567.0  150.0  mastercard  117.0  …  NaN   NaN   NaN   NaN   NaN  NaN
4  514.0  150.0  mastercard  102.0  …  0.0   0.0   0.0   0.0   0.0  0.0
5  555.0  150.0        visa  226.0  …  NaN   NaN   NaN   NaN   NaN  NaN
6  360.0  150.0        visa  166.0  …  NaN   NaN   NaN   NaN   NaN  NaN
7  490.0  150.0        visa  226.0  …  NaN   NaN   NaN   NaN   NaN  NaN
8  100.0  150.0        visa  226.0  …  0.0   0.0   0.0   0.0   0.0  0.0
9  111.0  150.0  mastercard  224.0  …  NaN   NaN   NaN   NaN   NaN  NaN

  V336  V337  V338  V339
0  NaN   NaN   NaN   NaN
1  NaN   NaN   NaN   NaN
2  NaN   NaN   NaN   NaN
3  NaN   NaN   NaN   NaN
4  0.0   0.0   0.0   0.0
5  NaN   NaN   NaN   NaN
6  NaN   NaN   NaN   NaN
7  NaN   NaN   NaN   NaN
8  0.0   0.0   0.0   0.0
9  NaN   NaN   NaN   NaN

[10 rows x 394 columns]
```

[3]:

```python
identity_df = pd.read_csv("./input_data/train_identity.csv")
identity_df.head(10)
```

[3]:

```
   TransactionID  id_01     id_02  id_03  id_04  id_05  id_06  id_07  id_08  \
0        2987004    0.0   70787.0    NaN    NaN    NaN    NaN    NaN    NaN
1        2987008   -5.0   98945.0    NaN    NaN    0.0   -5.0    NaN    NaN
2        2987010   -5.0  191631.0    0.0    0.0    0.0    0.0    NaN    NaN
3        2987011   -5.0  221832.0    NaN    NaN    0.0   -6.0    NaN    NaN
4        2987016    0.0    7460.0    0.0    0.0    1.0    0.0    NaN    NaN
```

```
5        2987017   -5.0   61141.0    3.0    0.0    3.0    0.0    NaN    NaN
6        2987022  -15.0       NaN    NaN    NaN    NaN    NaN    NaN    NaN
7        2987038    0.0   31964.0    0.0    0.0    0.0  -10.0    NaN    NaN
8        2987040  -10.0  116098.0    0.0    0.0    0.0    0.0    NaN    NaN
9        2987048   -5.0  257037.0    NaN    NaN    0.0    0.0    NaN    NaN

   id_09  …                id_31  id_32      id_33           id_34  id_35  \
0    NaN  …  samsung browser 6.2   32.0  2220x1080  match_status:2      T
1    NaN  …   mobile safari 11.0   32.0   1334x750  match_status:1      T
2    0.0  …          chrome 62.0    NaN        NaN             NaN      F
3    NaN  …          chrome 62.0    NaN        NaN             NaN      F
4    0.0  …          chrome 62.0   24.0   1280x800  match_status:2      T
5    3.0  …          chrome 62.0   24.0   1366x768  match_status:2      T
6    NaN  …                  NaN    NaN        NaN             NaN    NaN
7    0.0  …          chrome 62.0   32.0  1920x1080  match_status:2      T
8    0.0  …          chrome 62.0    NaN        NaN             NaN      F
9    NaN  …          chrome 62.0    NaN        NaN             NaN      F

  id_36 id_37  id_38  DeviceType                     DeviceInfo
0     F     T      T      mobile  SAMSUNG SM-G892A Build/NRD90M
1     F     F      T      mobile                     iOS Device
2     F     T      T     desktop                        Windows
3     F     T      T     desktop                            NaN
4     F     T      T     desktop                          MacOS
5     F     T      T     desktop                        Windows
6   NaN   NaN    NaN         NaN                            NaN
7     F     T      T      mobile                            NaN
8     F     T      T     desktop                        Windows
9     F     T      T     desktop                        Windows

[10 rows x 41 columns]
```

[4]:
```python
train_df = pd.merge(transaction_df, identity_df, on='TransactionID', how='left')
```

We display basic information on the dataset merged below. And we see how not all transactions have identity information

[22]:
```python
print("Shape of transaction data:", transaction_df.shape)
print("Shape of identity data:", identity_df.shape)
print("Shape after merge:", train_df.shape)
train_df.head()
```

```
Shape of transaction data: (590540, 394)
Shape of identity data: (144233, 41)
Shape after merge: (590540, 434)
```

[22]:
```
   TransactionID  isFraud  TransactionDT  TransactionAmt ProductCD  card1  \
0        2987000        0          86400            68.5         W  13926
```

```
1       2987001         0          86401            29.0          W   2755
2       2987002         0          86469            59.0          W   4663
3       2987003         0          86499            50.0          W  18132
4       2987004         0          86506            50.0          H   4497

   card2  card3       card4  card5  …                 id_31  id_32  \
0    NaN  150.0    discover  142.0  …                   NaN    NaN
1  404.0  150.0  mastercard  102.0  …                   NaN    NaN
2  490.0  150.0        visa  166.0  …                   NaN    NaN
3  567.0  150.0  mastercard  117.0  …                   NaN    NaN
4  514.0  150.0  mastercard  102.0  …  samsung browser 6.2   32.0

       id_33           id_34  id_35 id_36 id_37  id_38  DeviceType  \
0        NaN             NaN    NaN   NaN   NaN    NaN         NaN
1        NaN             NaN    NaN   NaN   NaN    NaN         NaN
2        NaN             NaN    NaN   NaN   NaN    NaN         NaN
3        NaN             NaN    NaN   NaN   NaN    NaN         NaN
4  2220x1080  match_status:2      T     F     T      T      mobile

                     DeviceInfo
0                           NaN
1                           NaN
2                           NaN
3                           NaN
4  SAMSUNG SM-G892A Build/NRD90M

[5 rows x 434 columns]
```

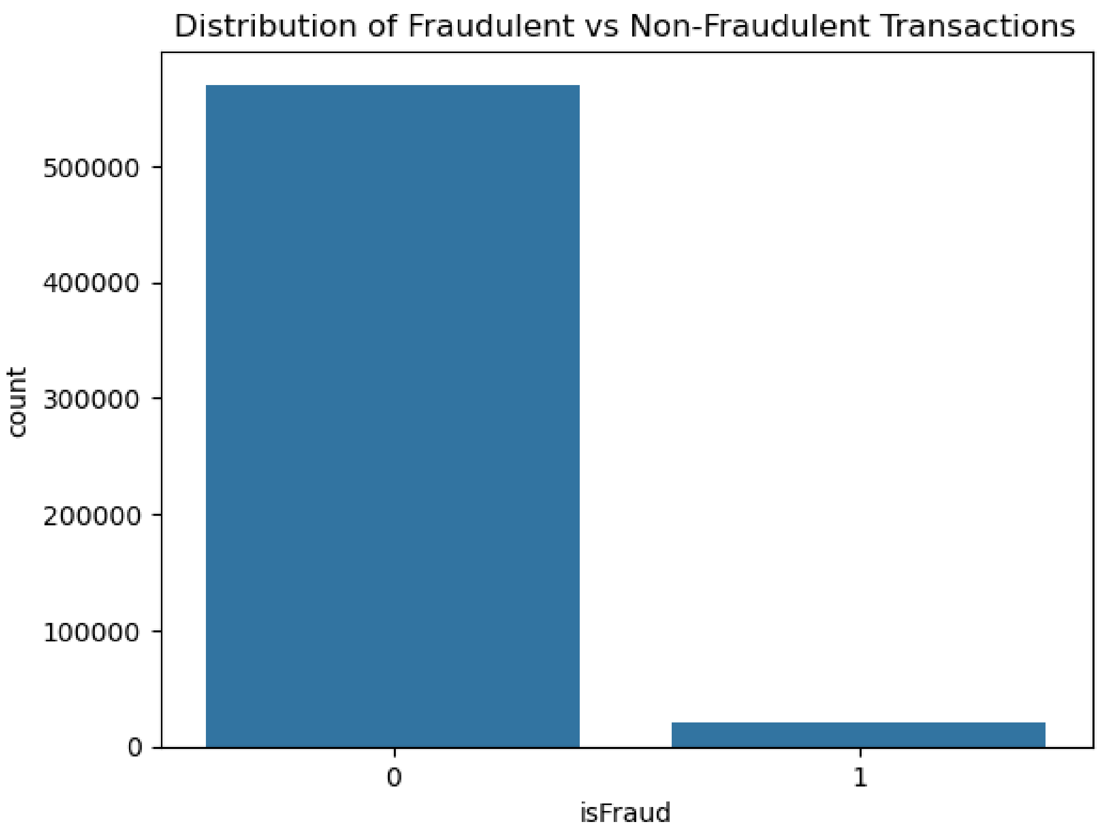
[28]:
```python
missing = train_df.isnull().sum() / len(train_df) * 100
missing = missing[missing > 0].sort_values(ascending=False)
print("Columns with missing values:")
print(missing.head(100))
```

```
Columns with missing values:
id_24    99.196159
id_25    99.130965
id_07    99.127070
id_08    99.127070
id_21    99.126393
           …
V236     77.913435
V237     77.913435
V240     77.913435
V249     77.913435
V252     77.913435
Length: 100, dtype: float64
```

Many columns have a large number of nulls, so we will need to take that into account. If it represents the data we will see in real transactions, it might be interesting to keep it as the model will need to learn from those nulls as well.

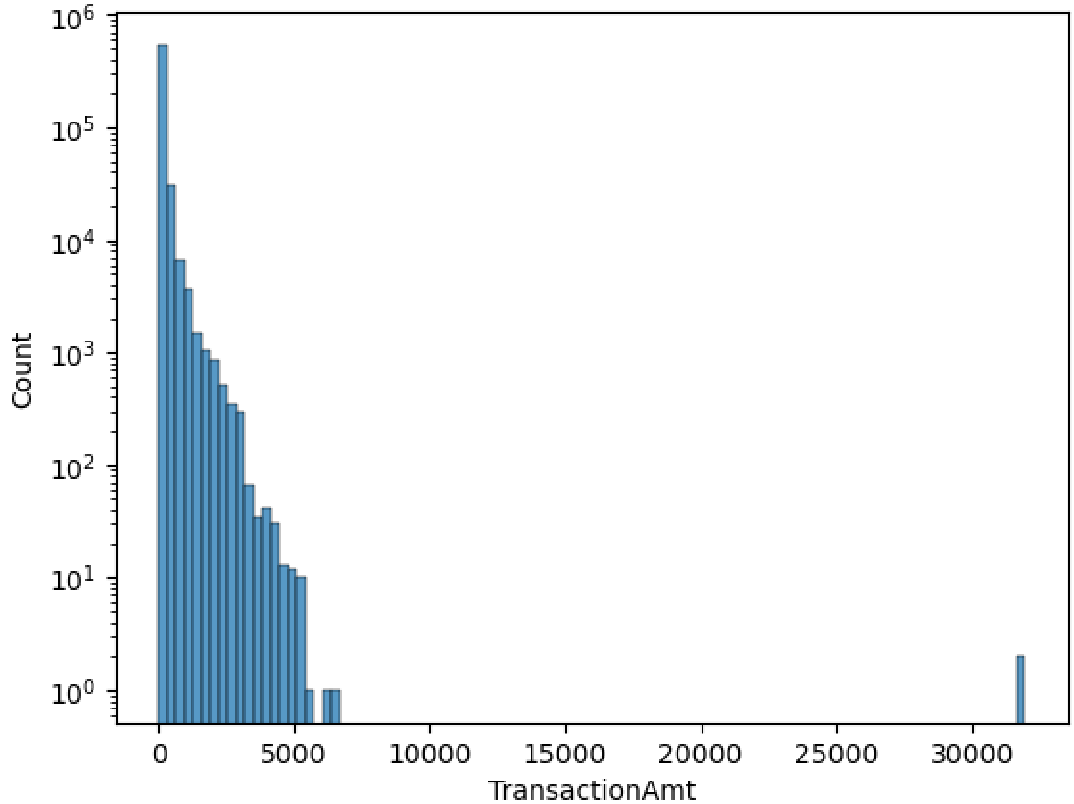
[33]:
```
fraud_ratio = train_df['isFraud'].value_counts(normalize=True) * 100
print(fraud_ratio)

sns.countplot(data=train_df, x='isFraud')
plt.title('Distribution of Fraudulent vs Non-Fraudulent Transactions')
plt.show()
```

```
isFraud
0    96.500999
1     3.499001
Name: proportion, dtype: float64
```



We observe that the proportion of non-fraudulent cases is only 3.5%, which represents the reality of cases that have this imbalance in the datasets.

[36]:
```
sns.histplot(train_df['TransactionAmt'], bins=100, log_scale=(False, True))
plt.yscale('log')
```

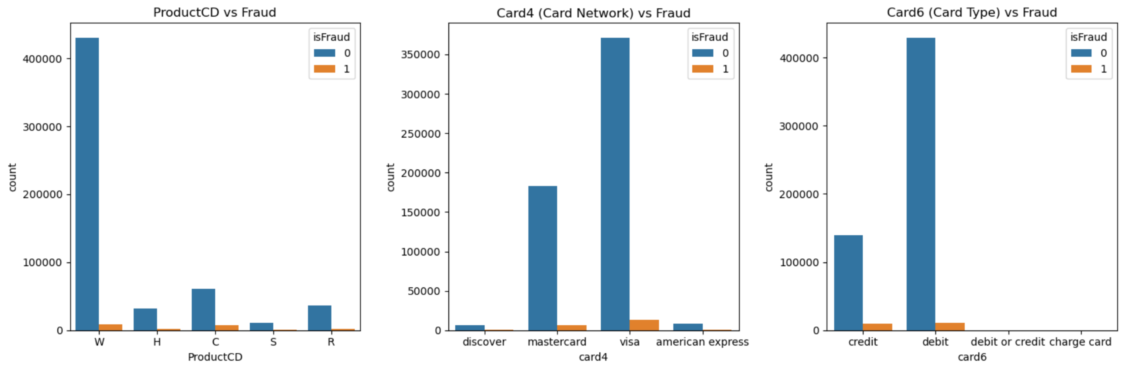


We can see that the transaction amounts are highly skewed, meaning that the majority of transactions are really small. This leads to a long right tail in the histogram. We observe it in logaritmic scale to be able to observe the large transactions as well.

[37]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=train_df, x='isFraud', y='TransactionAmt')
plt.yscale('log')
plt.title('Transaction Amount vs Fraud')
plt.show()
```

If we observe the Transaction Amount for fraudulent vs non-fraudulent cases we can see that fraudulent cases shows a slightly higher median and a wider interquartile range (IQR). This suggests that while most frauds happen at small amounts (likely to avoid detection), some frauds involve large transactions, creating the long upper tail.

[39]:
```
fig, axes = plt.subplots(1, 3, figsize=(15,5))

sns.countplot(data=train_df, x='ProductCD', hue='isFraud', ax=axes[0])
axes[0].set_title('ProductCD vs Fraud')

sns.countplot(data=train_df, x='card4', hue='isFraud', ax=axes[1])
axes[1].set_title('Card4 (Card Network) vs Fraud')

sns.countplot(data=train_df, x='card6', hue='isFraud', ax=axes[2])
axes[2].set_title('Card6 (Card Type) vs Fraud')

plt.tight_layout()
plt.show()
```

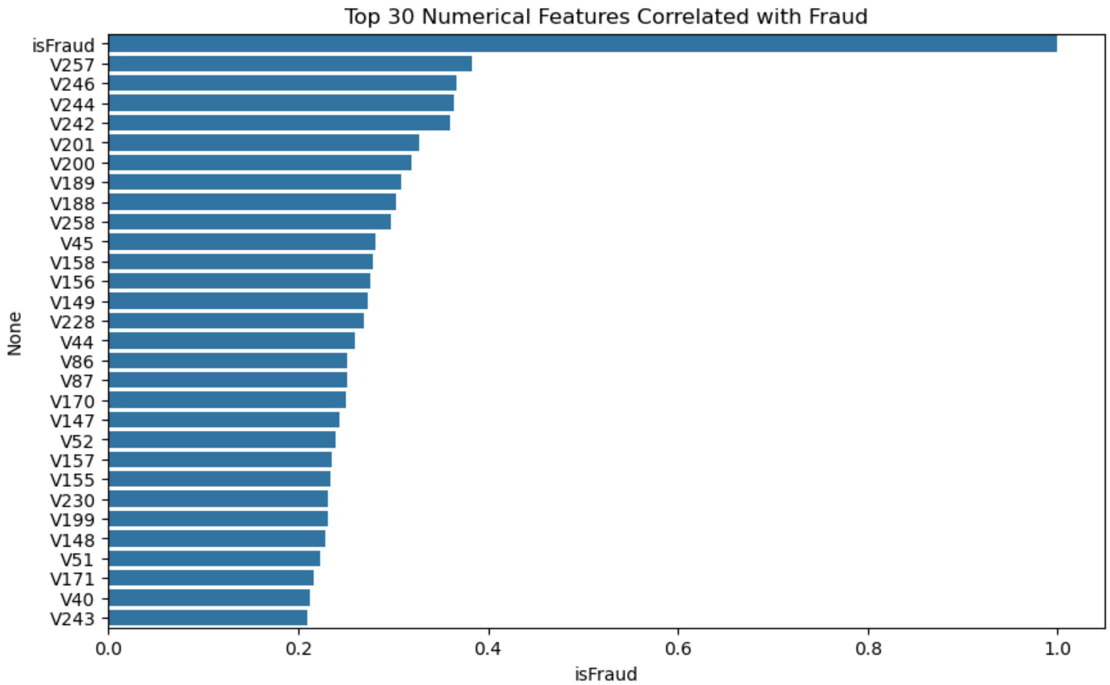


We also observe which products are being bought, seeing that product `W` has the largest presence in the dataset. We also observe most transactions are with `mastercard` or `visa` and most are `debit`.

[42]:
```
numeric_cols = train_df.select_dtypes(include=np.number).columns
corr = train_df[numeric_cols].corr()['isFraud'].sort_values(ascending=False)
corr.head(10)
```

[42]:
```
isFraud    1.000000
V257       0.383060
V246       0.366878
V244       0.364129
V242       0.360590
V201       0.328005
V200       0.318783
V189       0.308219
V188       0.303582
V258       0.297151
Name: isFraud, dtype: float64
```

[48]:
```
plt.figure(figsize=(10,6))
sns.barplot(x=corr.head(30), y=corr.head(30).index)
plt.title('Top 30 Numerical Features Correlated with Fraud')
plt.show()
```



We observe that there are multiple V columns that are highly correlated with isFraud