Data Storytelling

This step is to explore the dataset to discover certain trends.
The goal of the exploratory data analysis was to find out whether a certain demographic is more likely to get approved compared to others.
The dataset was truncated to create a dataset centered around gender and education. In order to compare the demographics perfectly, different histograms were used to explore these newly-formed dataset.

Plotting graphs for different features includes repetitive line of codes, so to overcome this scenario different plotting functions were defined. These functions were called as per the need of visualization.

★ Starting with the graph that depicts the dataset :-

```
#Percentages of Fraudlent and Non-Fraudulent transactions in data
## Here f-string is used to make string interpolation simple and it is faster than str.format
print(f'Percent of Non-Fraudulent Transactions = {round(data["Class"].value_counts()[0]/len(data) * 100,3)}%')
print(f'Percent of Non-Fraudulent Transactions = {round(data["Class"].value_counts()[1]/len(data) * 100,3)}%')
```

```
Percent of Non-Fraudulent Transactions = 99.827%
Percent of Non-Fraudulent Transactions = 0.173%
```

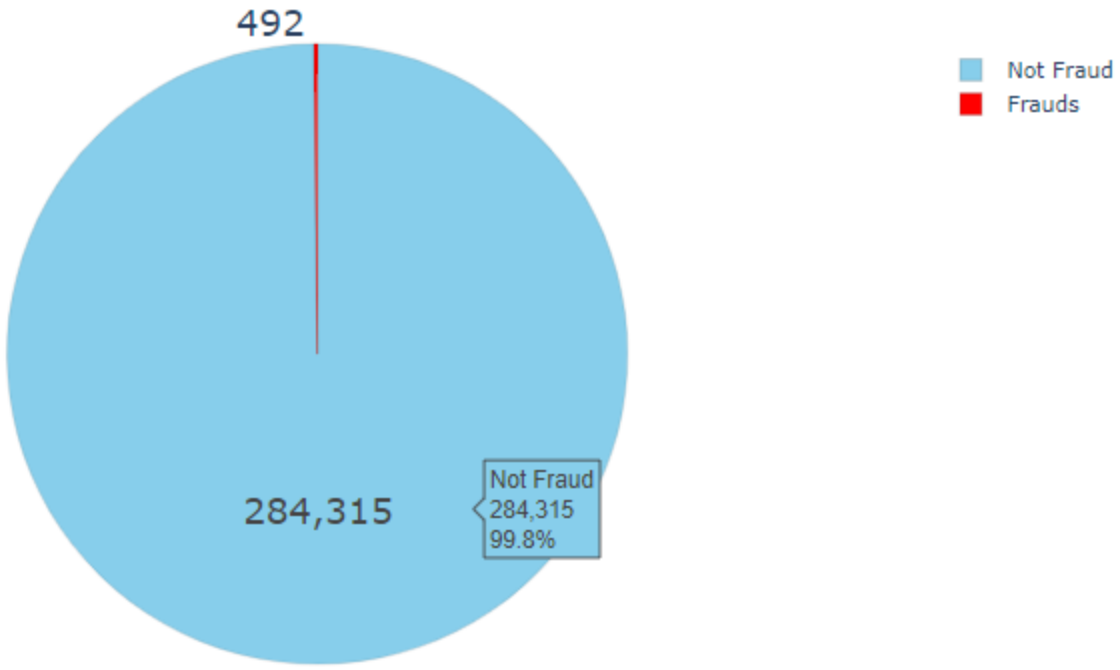
```
#plotting a pie chart for fraud and non-fraud transactions

fraud_or_not = data["Class"].value_counts().tolist()

labels = ['Not Fraud','Frauds']
values = [fraud_or_not[0], fraud_or_not[1]]
colors = ['skyblue', 'red']

trace = go.Pie(labels=labels, values=values, textinfo='value',
               textfont=dict(size=20),
               marker=dict(colors=colors, line=dict(color='#000000', width=0.1)))

plotly.offline.iplot([trace], filename='styled_pie_chart')
```

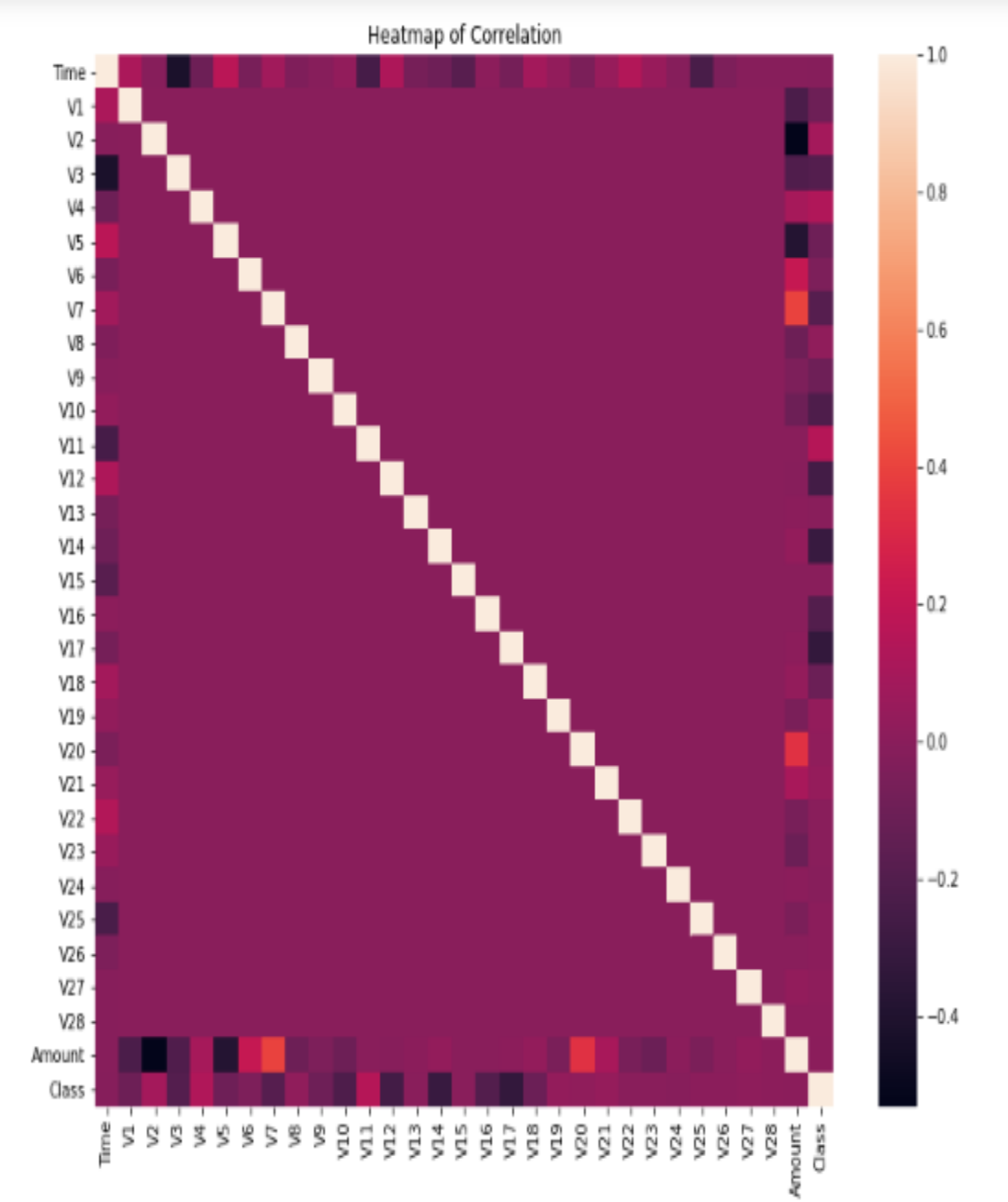


★ This dataset has 492 frauds out of 284,315 transactions. Thus, the dataset is highly unbalanced, the positive class (frauds) account for 0.173% of all transactions.

★ **Heat Map Correlation:-**

```
#heatmap
corr = data.corr()
plt.figure(figsize=(12,10))
heat = sns.heatmap(data=corr)
plt.title('Heatmap of Correlation')
```

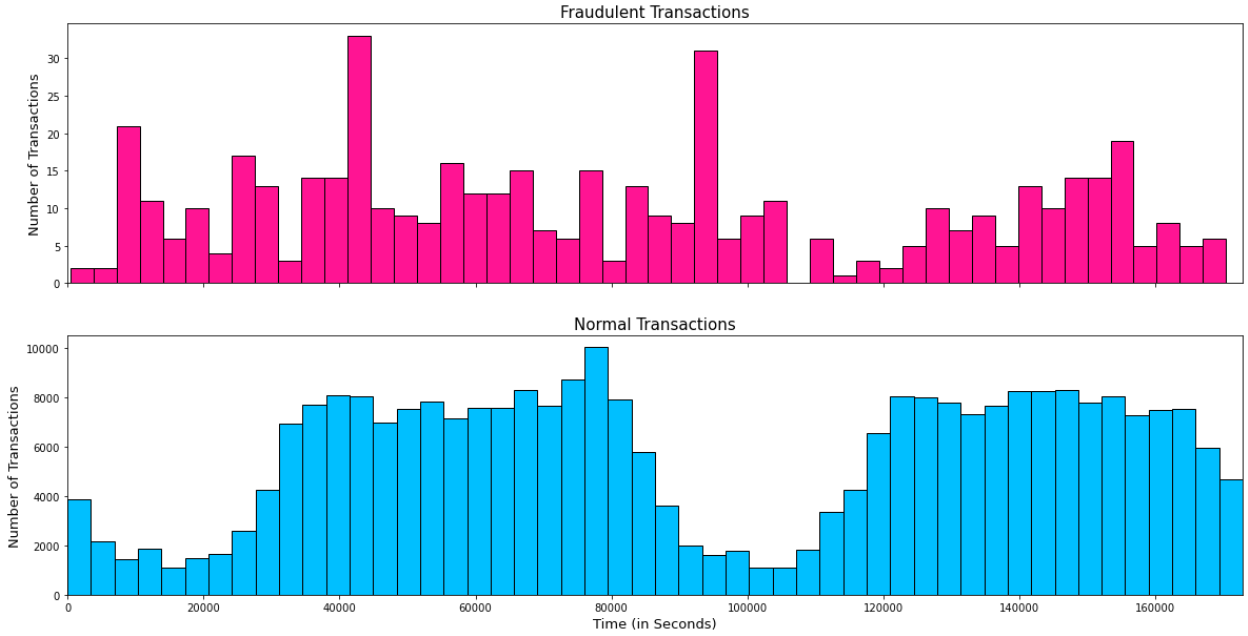
```
Text(0.5, 1.0, 'Heatmap of Correlation')
```



► In the HeatMap we can clearly see that most of the features do not correlate to other features but there are some features that either has a positive or a negative correlation with each other.

► For example “V2” and “V5” are highly negatively correlated with the feature called “Amount”. We also see some correlation with “V20” and “Amount”. This gives us a deeper understanding of the Data available to us.

★ **Analysis of Time Column:-**



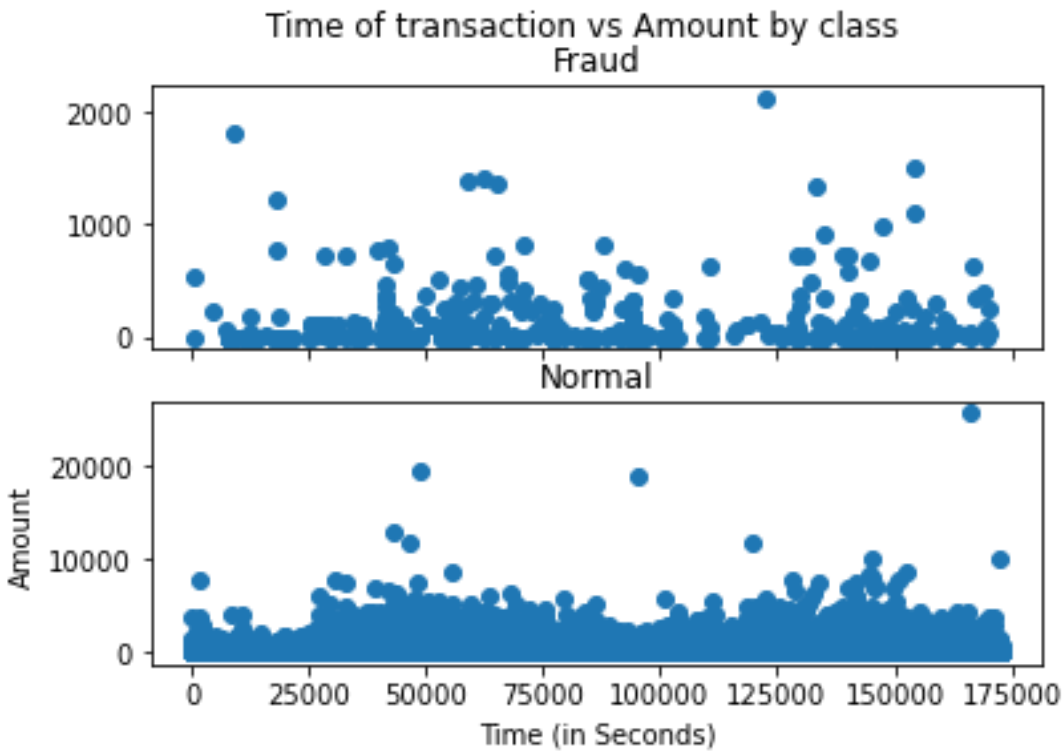
FRAUDULENT

► There are two peaks between 40000 seconds and 100000 seconds which were the maximum number of fraudulent transaction at any time.

NORMAL

► Normal transactions have not much to uncover except the fact that there were less transactions somewhere around 20000 seconds and 100000 seconds which is not very useful.

★ **Class and Amount vs Time:-**



FRAUDULENT

► There are much more outliers as compared to normal transactions.

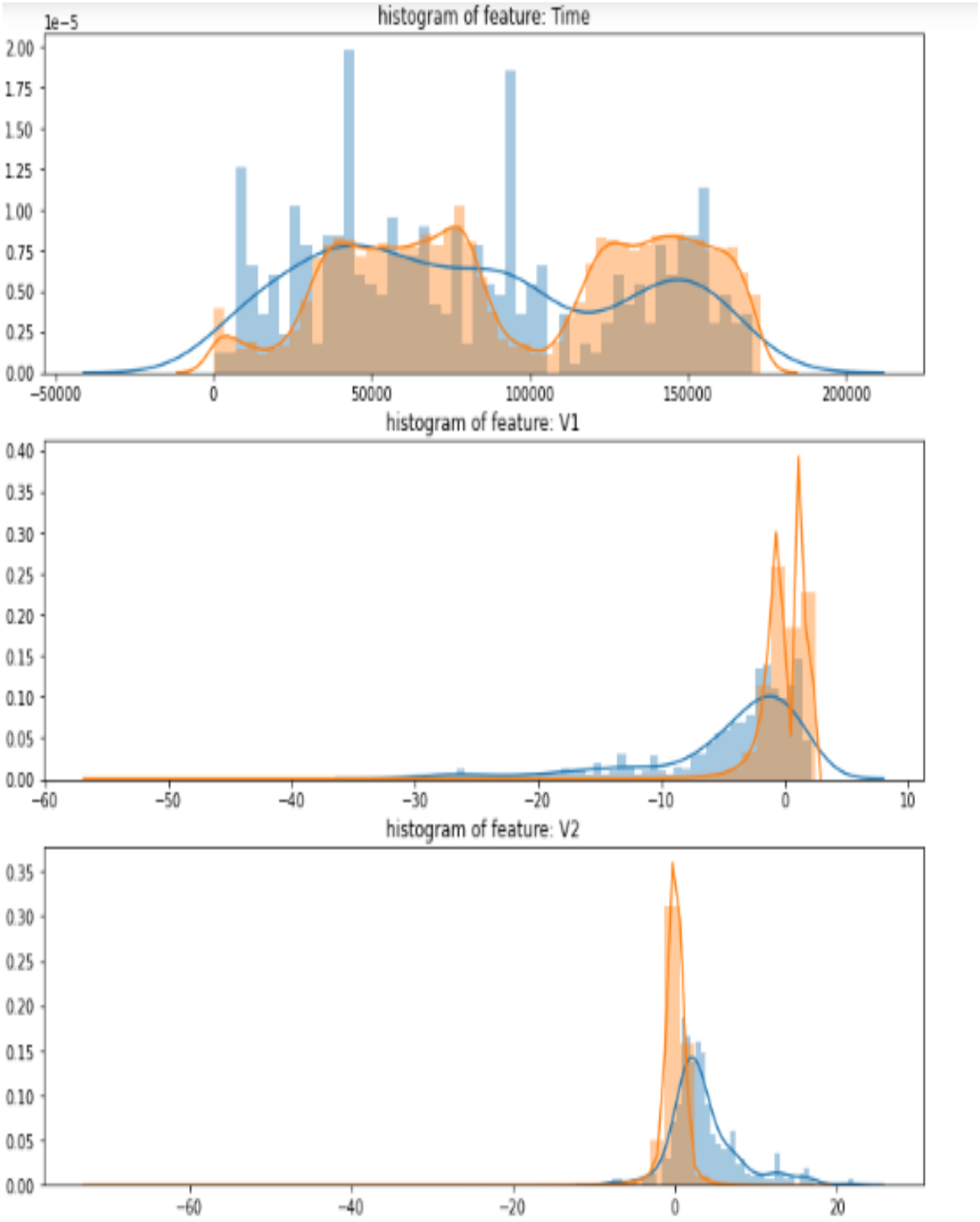
► The plot seems to not have any inherent pattern.

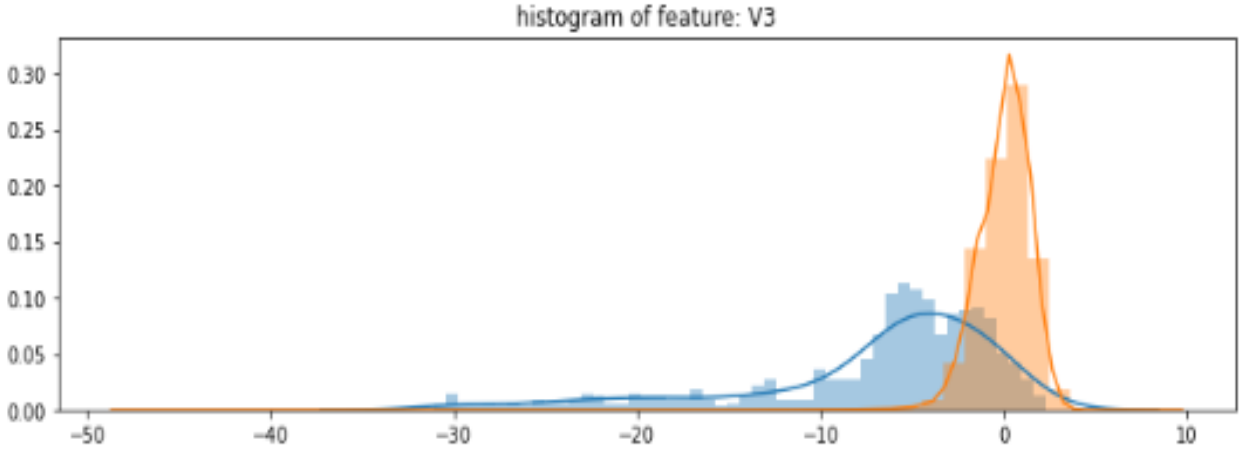
NORMAL

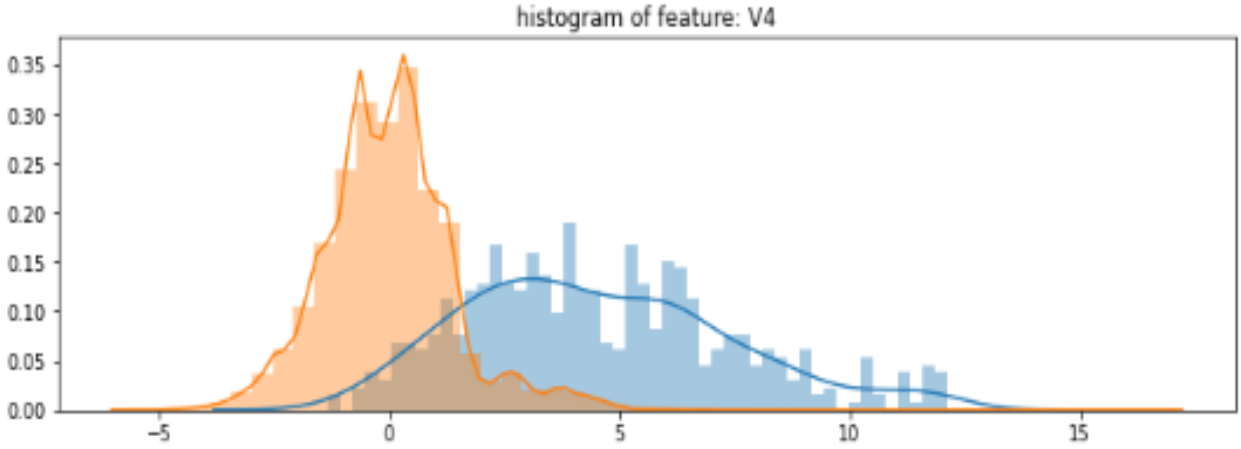
► There are a less number of outliers as compared to fraudulent transactions.

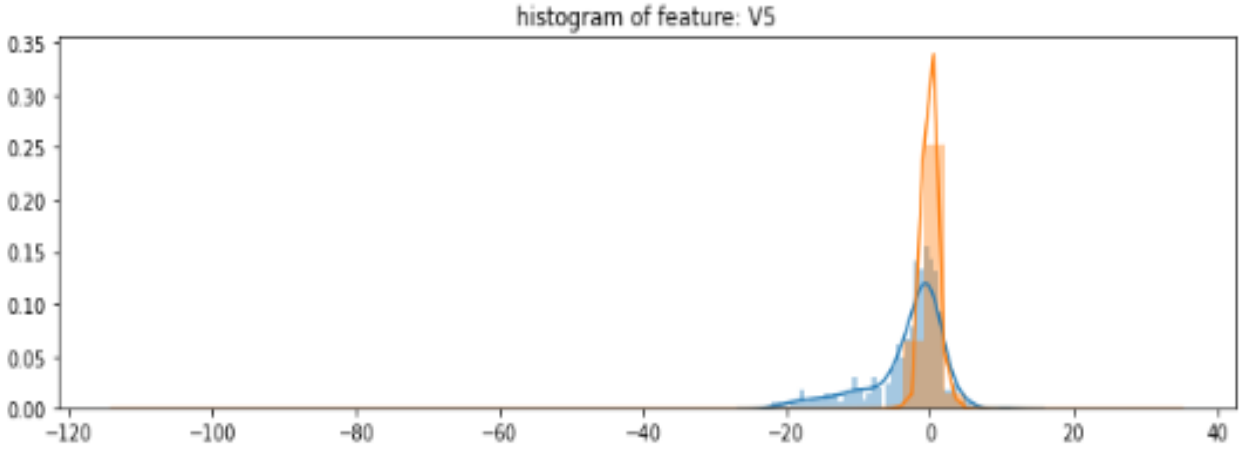
► There are a lot of transactions with amount less than 5000.

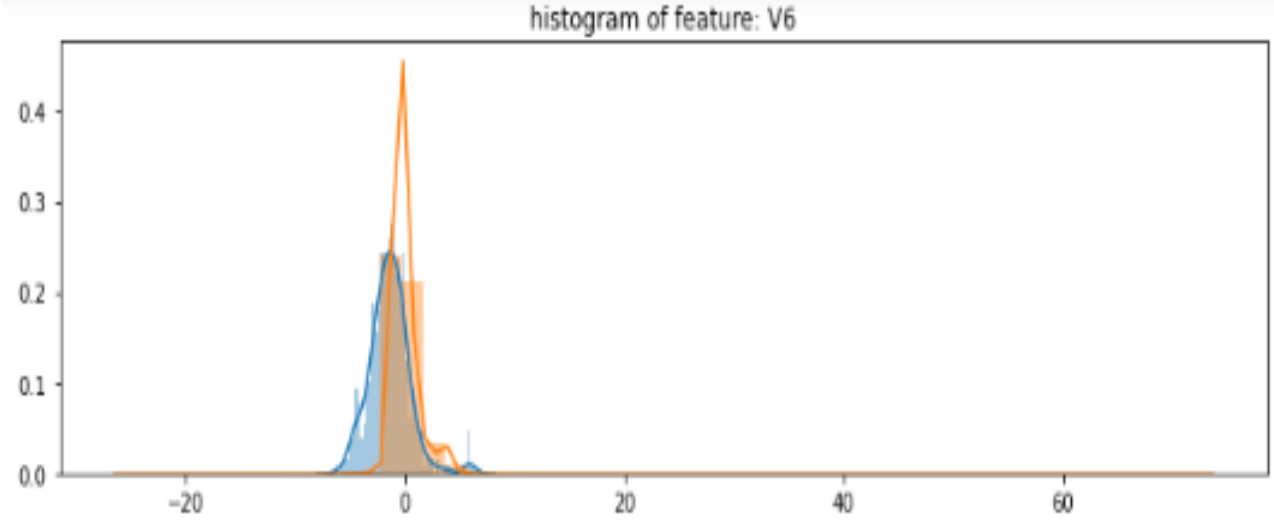
★ **Distribution of anomalous features:-**

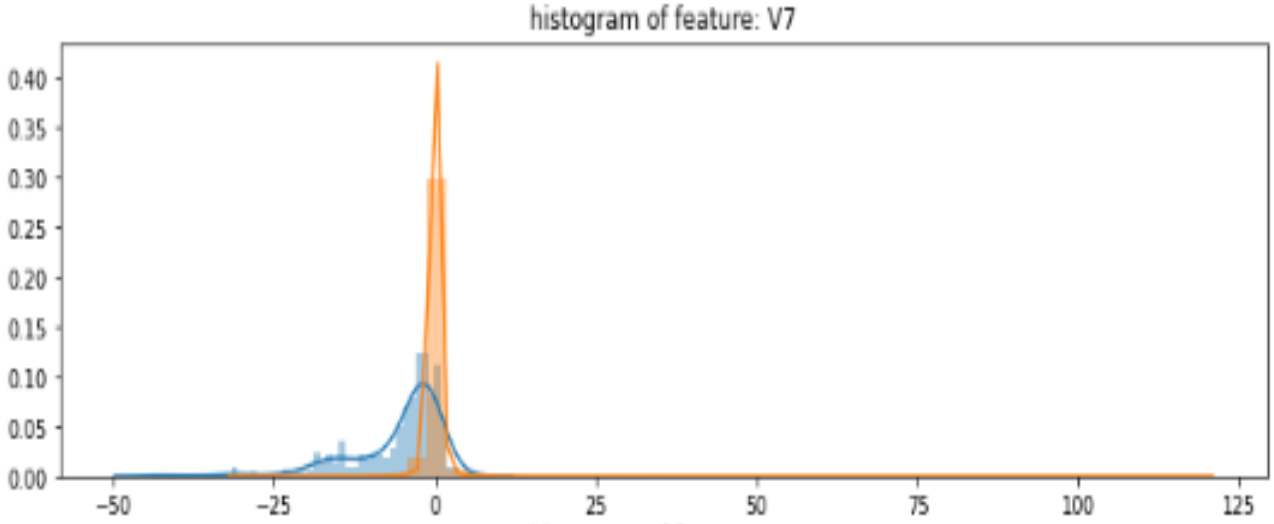


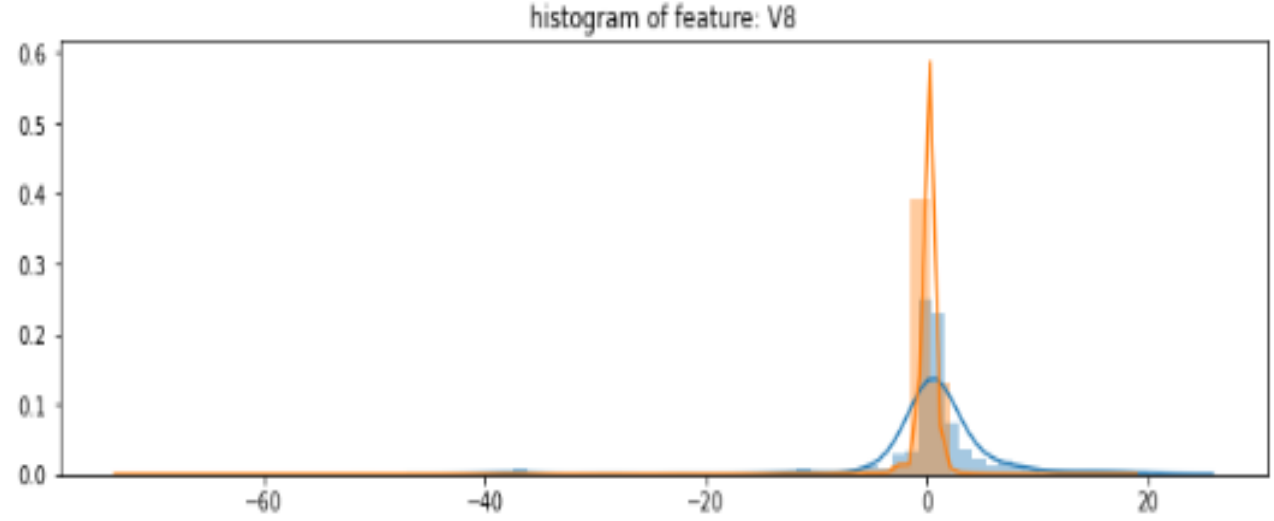

histogram of feature: V3
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−50
−40
−30
−20
−10
0
10

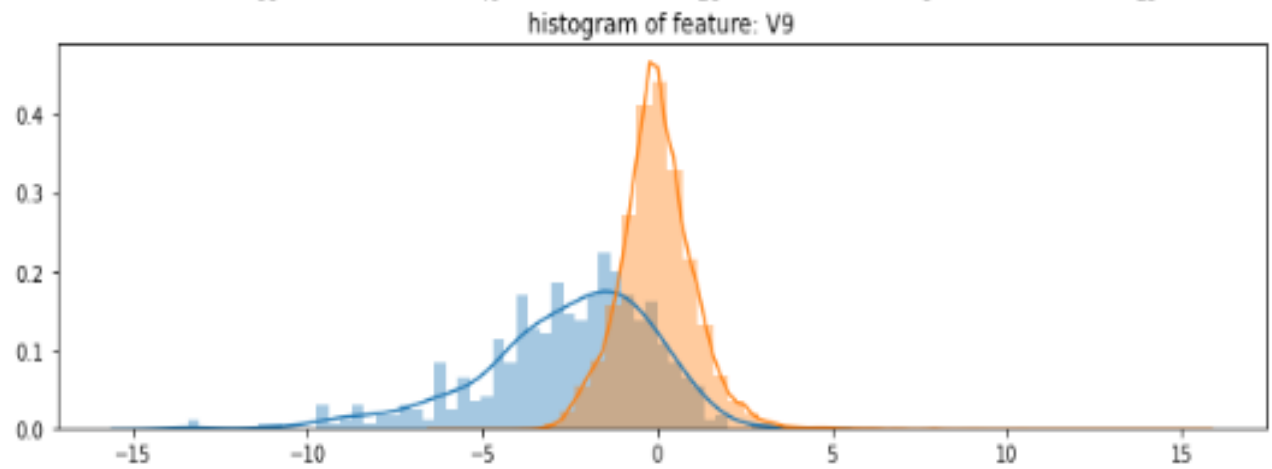

histogram of feature: V4
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−5
0
5
10
15

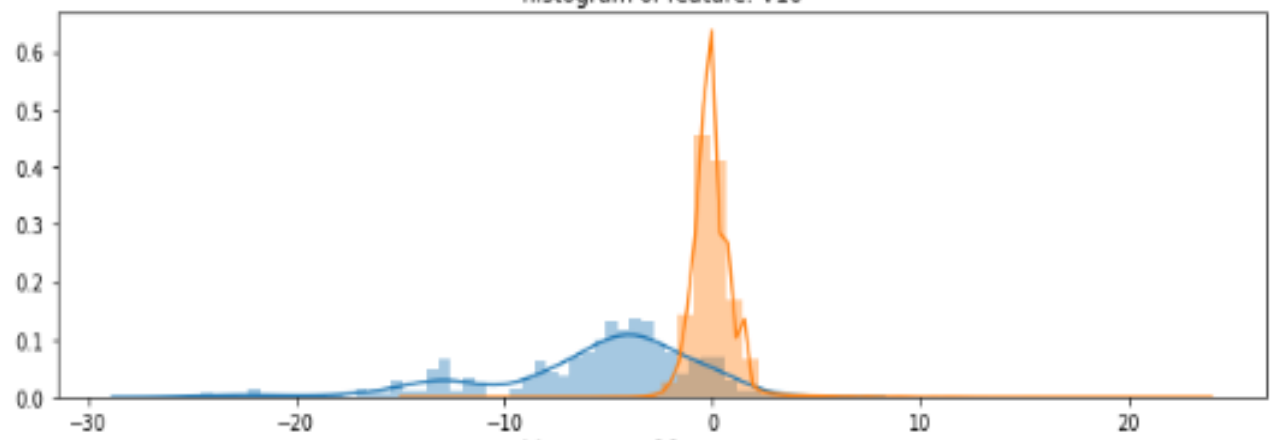

histogram of feature: V5
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−120
−100
−80
−60
−40
−20
0
20
40

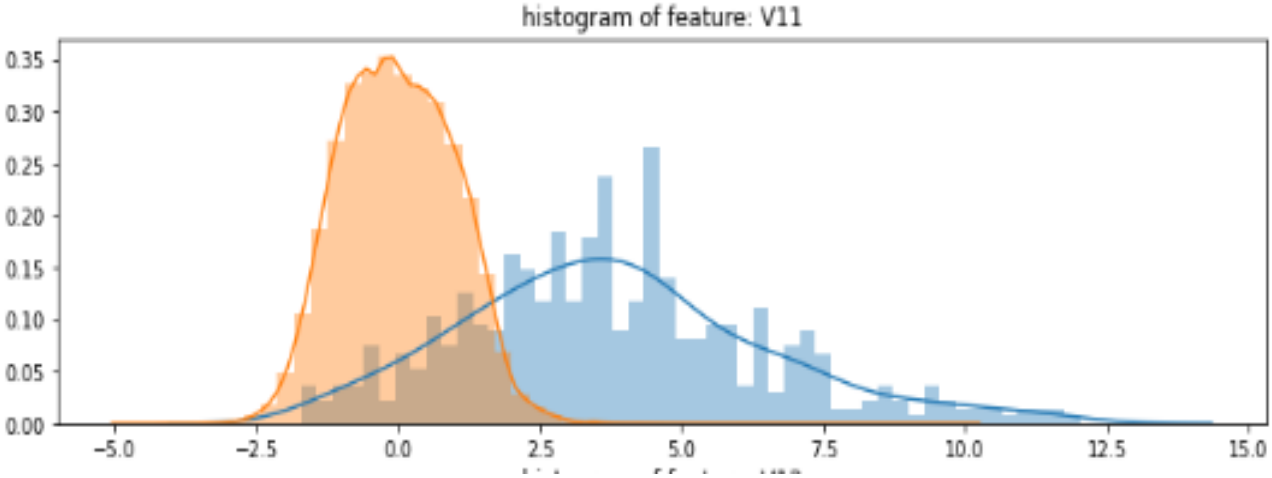

histogram of feature: V6
0.4
0.3
0.2
0.1
0.0
−20
0
20
40
60

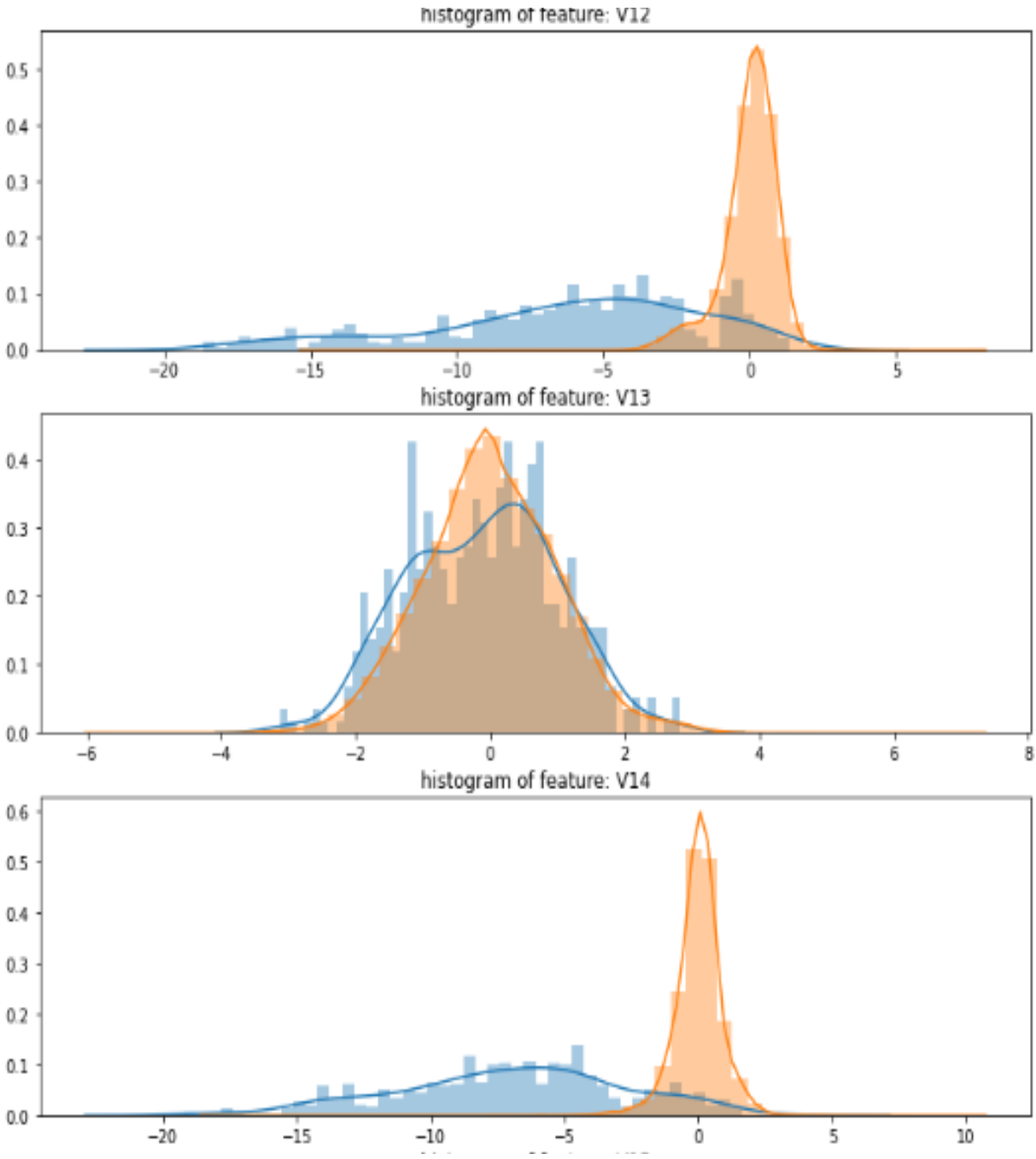

histogram of feature: V7
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−50
−25
0
25
50
75
100
125

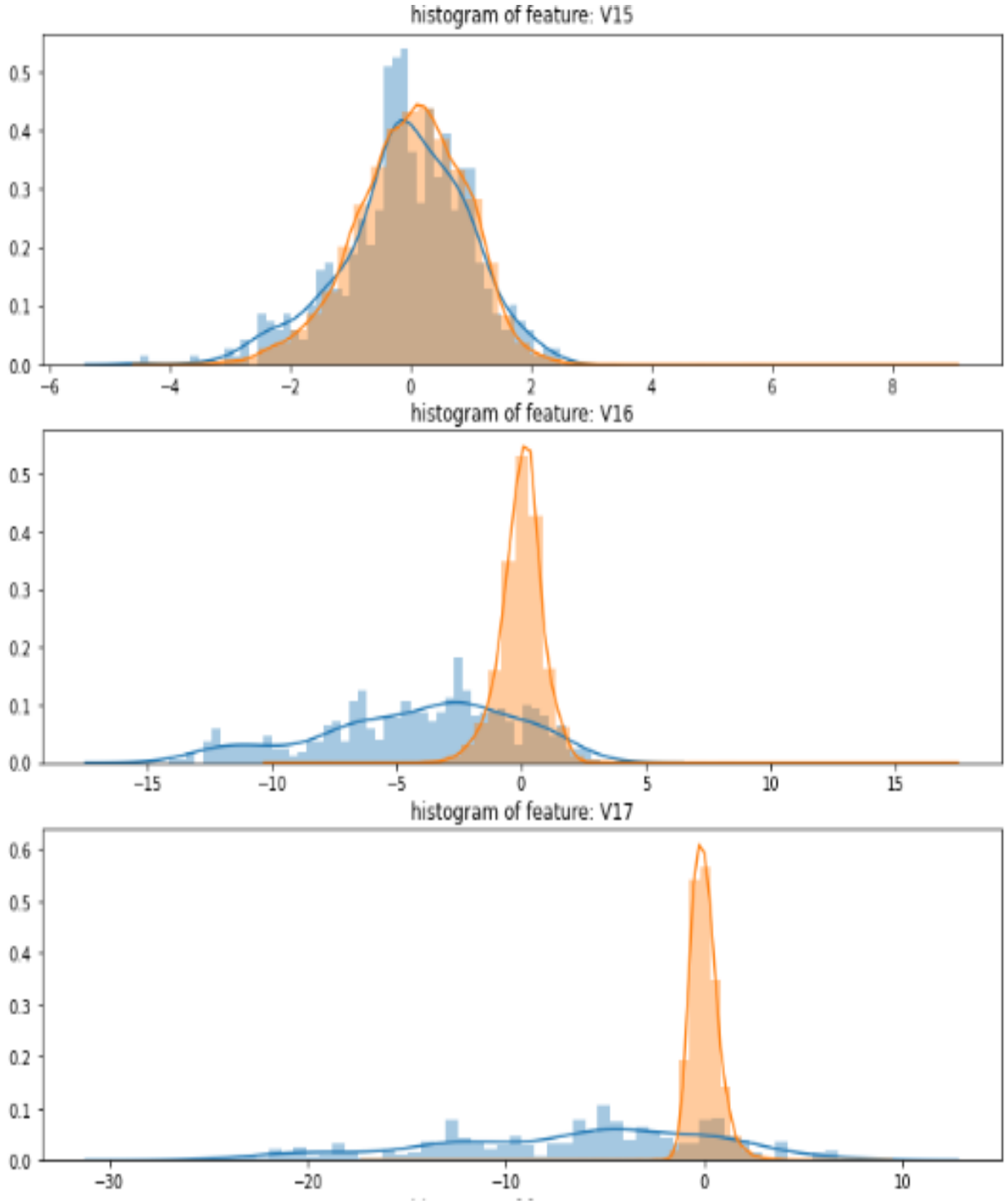

histogram of feature: V8
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−60
−40
−20
0
20

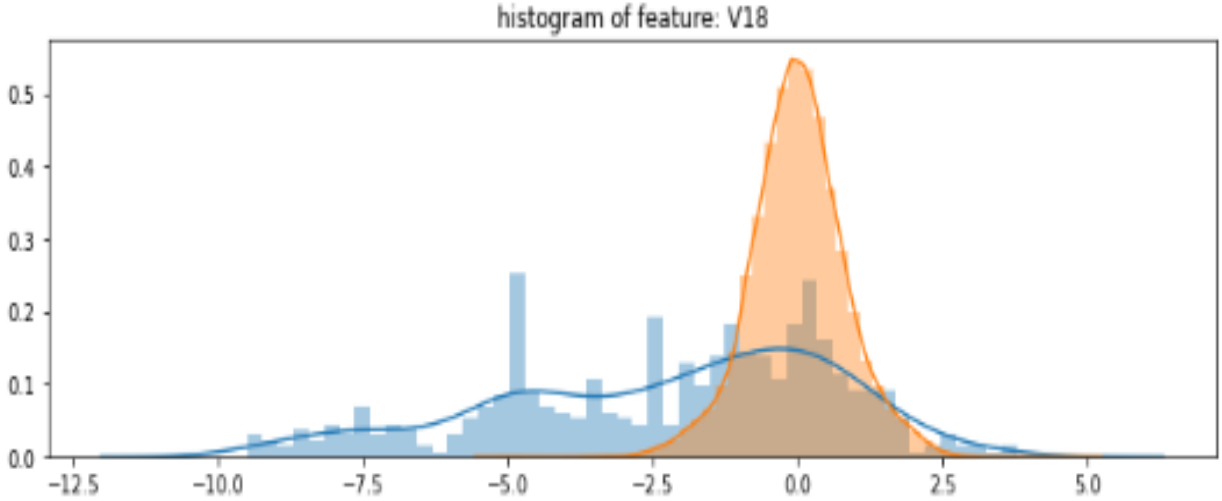

histogram of feature: V9
0.4
0.3
0.2
0.1
0.0
−15
−10
−5
0
5
10
15

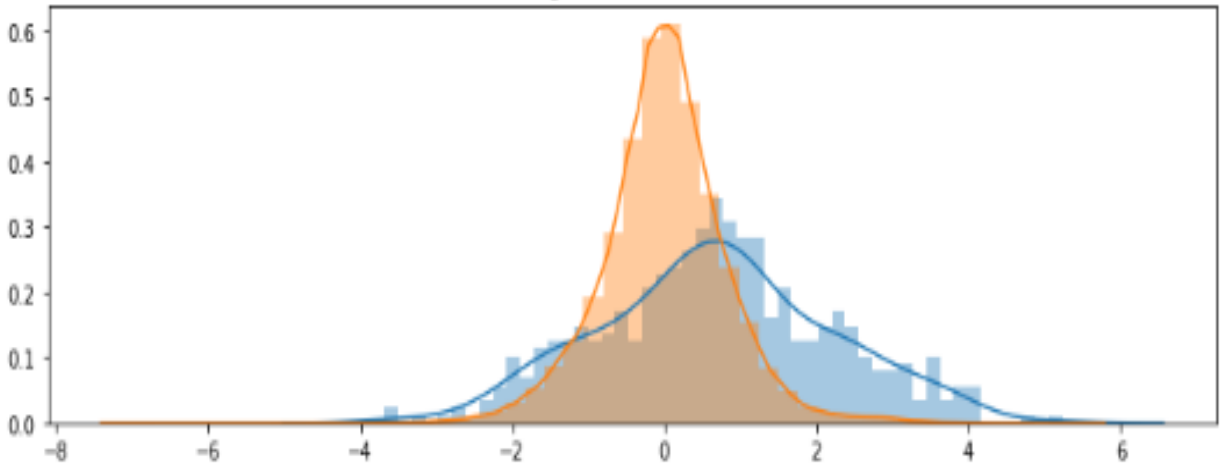

histogram of feature: V10
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−30
−20
−10
0
10
20

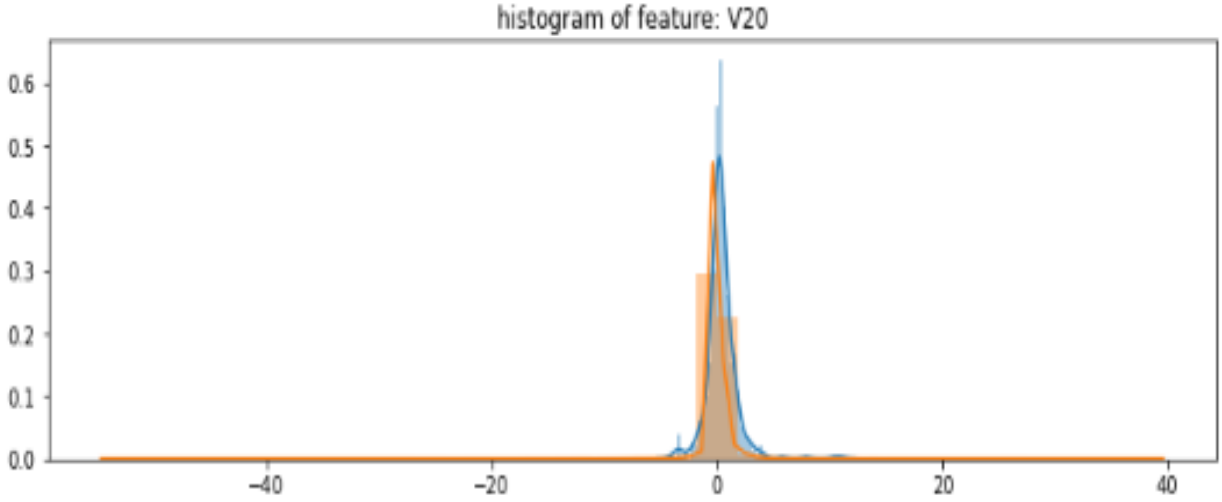

histogram of feature: V11
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

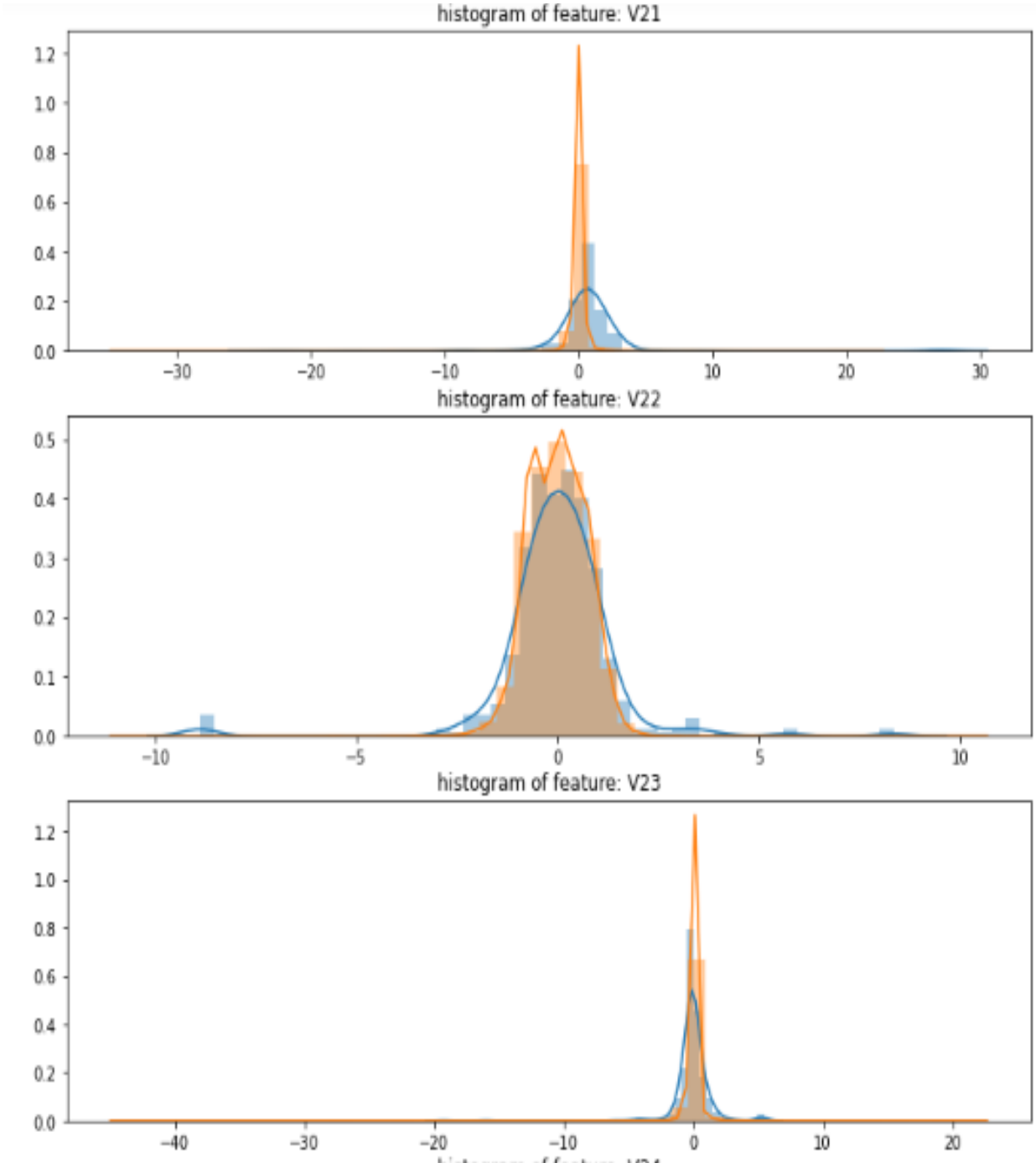

histogram of feature: V12
histogram of feature: V13
histogram of feature: V14

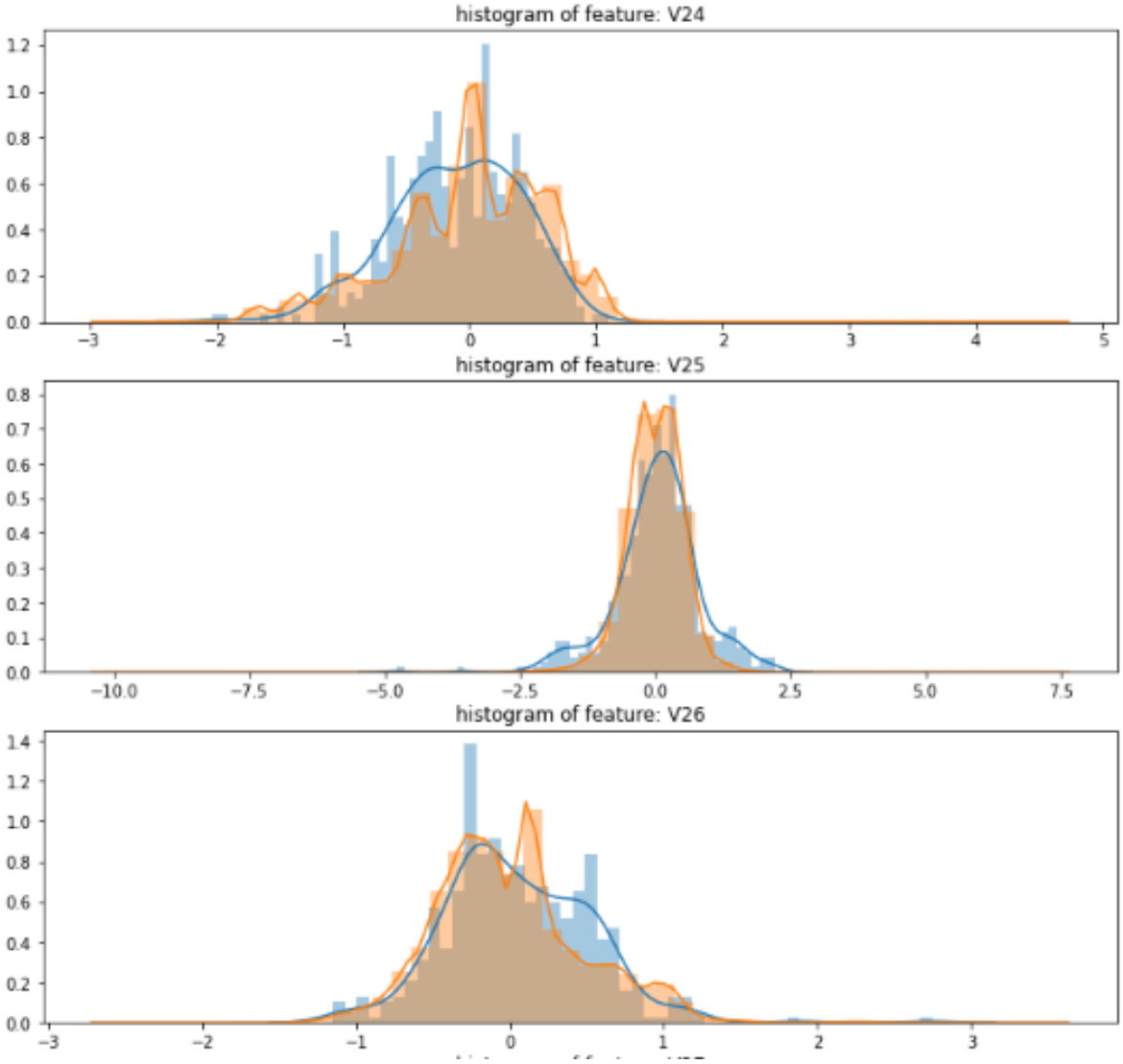

histogram of feature: V15
0.5
0.4
0.3
0.2
0.1
0.0
−6
−4
−2
0
2
4
6
8
histogram of feature: V16
0.5
0.4
0.3
0.2
0.1
0.0
−15
−10
−5
0
5
10
15
histogram of feature: V17
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−30
−20
−10
0
10

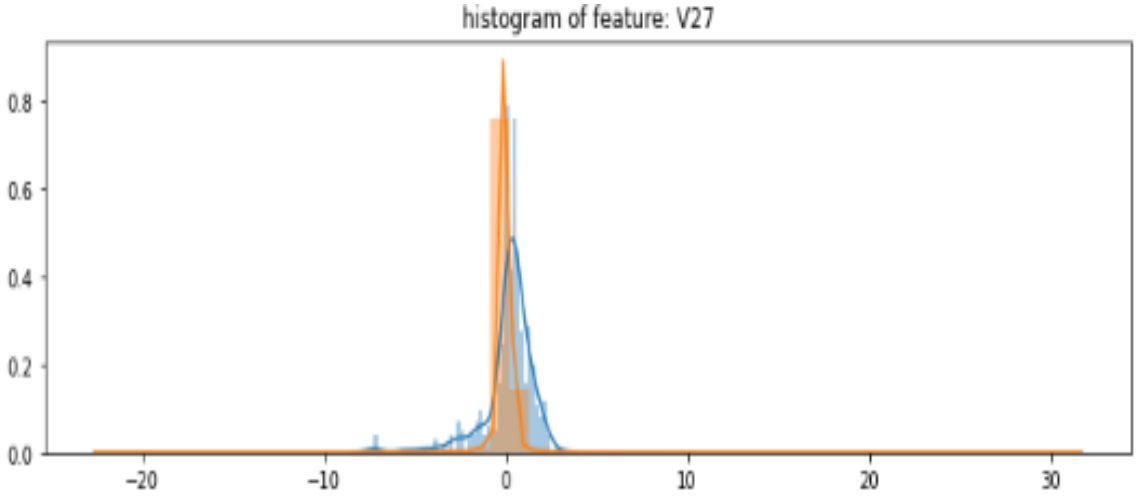

histogram of feature: V18
0.5
0.4
0.3
0.2
0.1
0.0
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0


histogram of feature: V19
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−8
−6
−4
−2
0
2
4
6

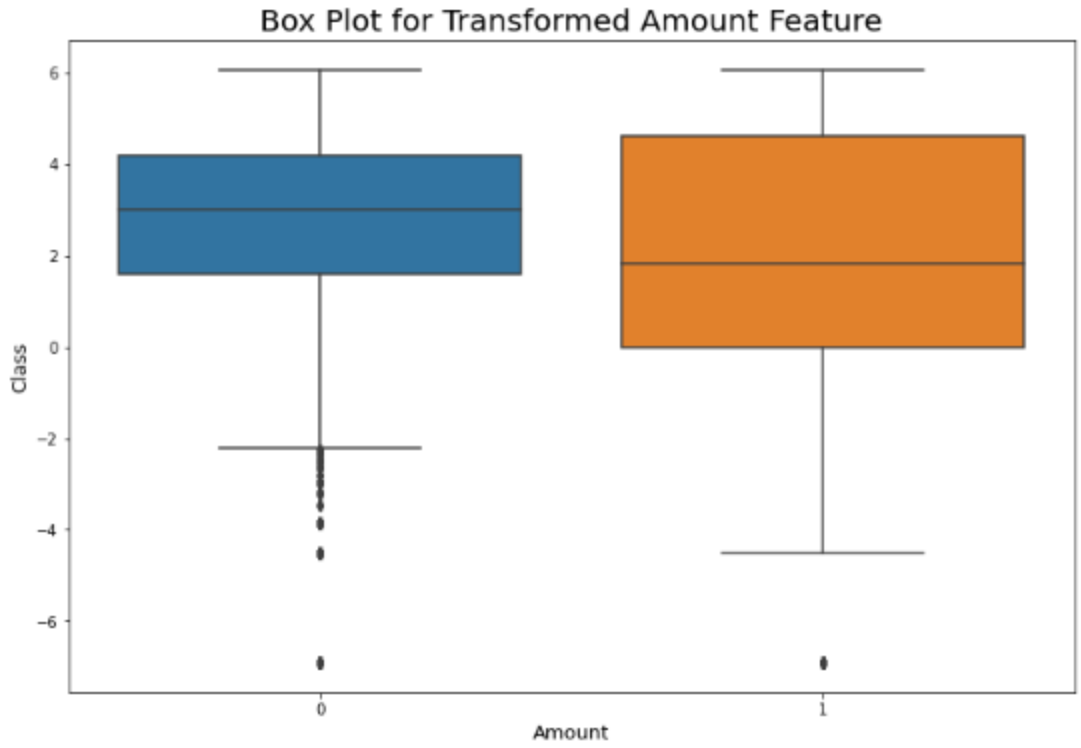

histogram of feature: V20
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−40
−20
0
20
40


histogram of feature: V21
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−30
−20
−10
0
10
20
30
histogram of feature: V22
0.5
0.4
0.3
0.2
0.1
0.0
−10
−5
0
5
10
histogram of feature: V23
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−40
−30
−20
−10
0
10
20


histogram of feature: V24
histogram of feature: V25
histogram of feature: V26

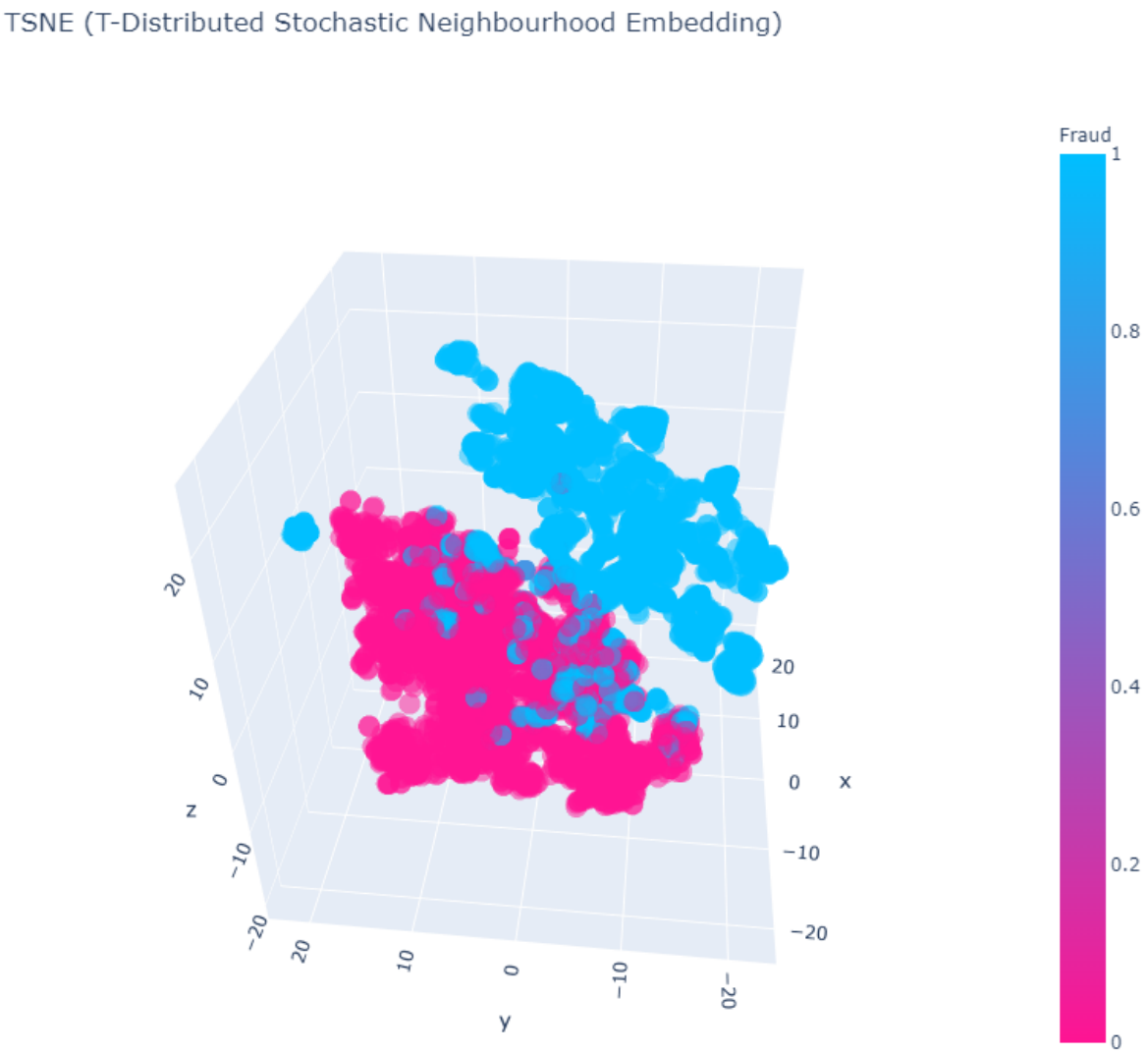

histogram of feature: V27

★ **BOX PLOT FOR TRANSFORMED AMOUNT FEATURE:-**



► After transforming a highly skewed amount feature.

★ **Dimensionality Reduction And Clustering:-**



► As expected, t-SNE is performing well, seperating both the classes. The clusters are not perfectly separated but it is able to separate almost 80% of the sample data which is cool.