Homework 2 - IEEE Fraud Detection

For all parts below, answer all parts as shown in the Google document for Homework 2. Be sure to include both code that justifies your answer as well as text to answer the questions. We also ask that code be commented to make it easier to follow.

Part 1 - Fraudulent vs Non-Fraudulent Transaction

In [86]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings


train_trans= pd.read_csv("train_transaction.csv")
train_ident = pd.read_csv("train_identity.csv")
```

In [2]:
```
print(train_trans.head())
print(len(train_trans))
```

```
   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  ... 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

  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 rows x 394 columns]
590540
```

In [3]:
```
print(train_ident.head())
print(len(train_ident))
```

```
   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

   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

  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 rows x 41 columns]
144233
```

In [73]:
```
# Outer join on TransactionID
merged_required = pd.merge(train_trans, train_ident, on='TransactionID
', how='outer')
```

In [74]:
```
print(merged_required.head())
print(merged_required.shape)
```

```
   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]
(590540, 434)
```

In [75]:
```
isFradTrans = merged_required[merged_required['isFraud'] == 1]
nonFraudTrans = merged_required[merged_required['isFraud'] == 0]

isFradTrans.head()
```

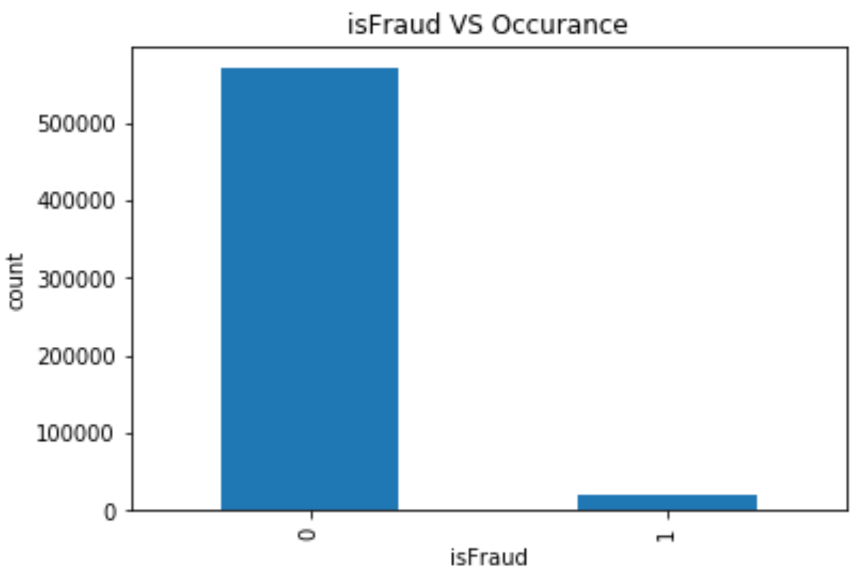
Out[75]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3
203	2987203	1	89760	445.000	W	18268	583.0	150.0
240	2987240	1	90193	37.098	C	13413	103.0	185.0
243	2987243	1	90246	37.098	C	13413	103.0	185.0
245	2987245	1	90295	37.098	C	13413	103.0	185.0
288	2987288	1	90986	155.521	C	16578	545.0	185.0

5 rows × 434 columns

In [76]:
```
import matplotlib.pyplot as plt
import seaborn as sns
merged_required['isFraud'].value_counts().plot(kind='bar')
plt.xlabel('isFraud')
plt.ylabel('count')
plt.title('isFraud VS Occurance')
merged_required['isFraud'].value_counts()
```

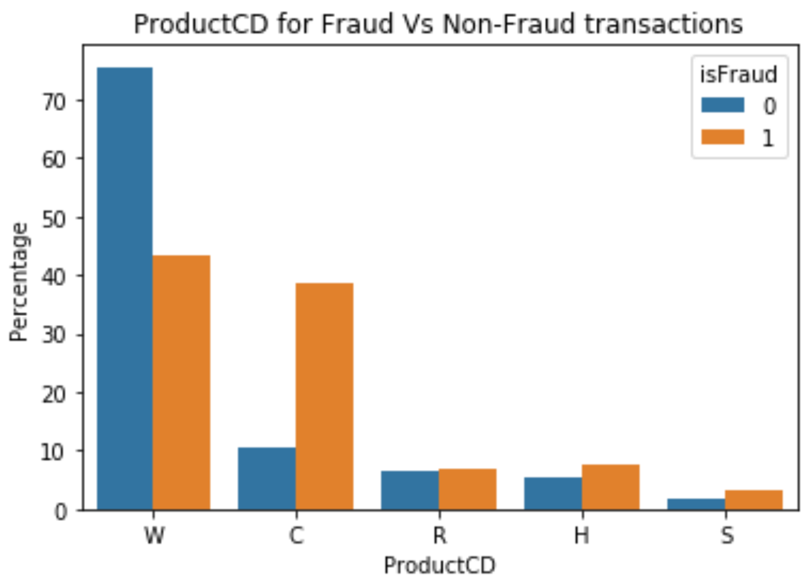
Out[76]:
```
0    569877
1     20663
Name: isFraud, dtype: int64
```



Most of the transactions are Non-Fraud. Almost 97% of transactions are Non-Fraud and remaining 3% are Fraud transactions.

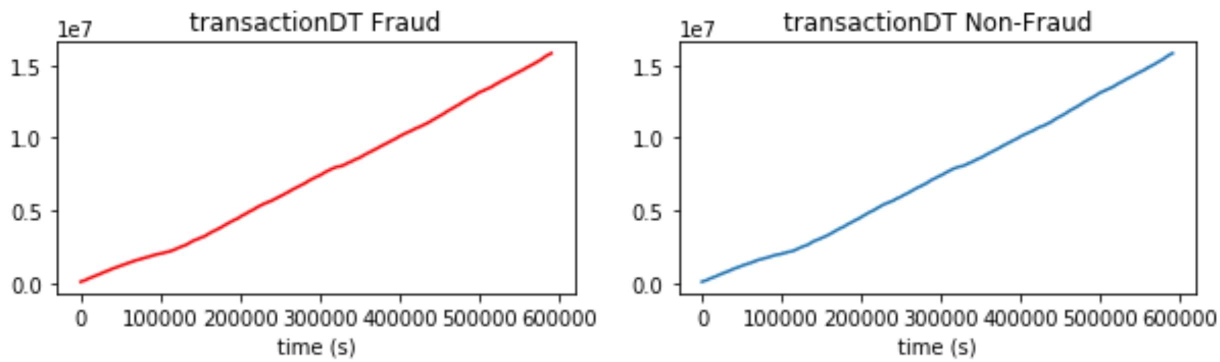
In [77]:
```
# Reference: https://seaborn.pydata.org/generated/seaborn.barplot.html

merge_plot = (merged_required["ProductCD"].groupby(merged_required["isFraud"]).value_counts(normalize=True).mul(100).rename("Percentage")
 .reset_index())
sns.barplot(x="ProductCD", y="Percentage", hue="isFraud", data=merge_plot)
plt.title('ProductCD for Fraud Vs Non-Fraud transactions')
```

Out[77]:
```
Text(0.5, 1.0, 'ProductCD for Fraud Vs Non-Fraud transactions')
```



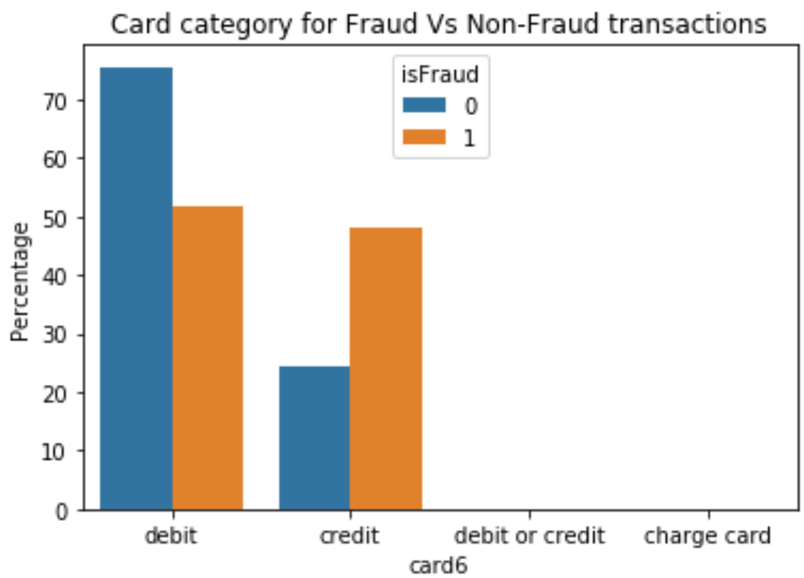
The above graph for ProductCD is plotted based on percentage of relative frequency. Although there're more fraud transactions for ProductCD 'W', from the graph we can observe that for Product Label 'C', there're more fraud transactions(40%) compared to Non-Fraud (10%).

In [78]:
```
# plot for transactionDT for Fraud Vs Non-Fraud
plt.figure(figsize=(10,5))
plt.subplot(2,2,1)
plt.plot(isFradTrans['TransactionDT'], 'r-')
plt.xlabel('time (s)')
plt.title('transactionDT Fraud')
plt.subplot(2,2,2)
plt.title('transactionDT Non-Fraud')
plt.plot(nonFraudTrans['TransactionDT'])
plt.xlabel('time (s)')
plt.show()
```



In [79]:
```
# Reference: https://seaborn.pydata.org/generated/seaborn.barplot.html
merge_plot = (merged_required["card6"].groupby(merged_required["isFrau
d"]).value_counts(normalize=True).mul(100).rename("Percentage")
 .reset_index())
sns.barplot(x="card6", y="Percentage", hue="isFraud", data=merge_plot)

plt.title('Card category for Fraud Vs Non-Fraud transactions')
```

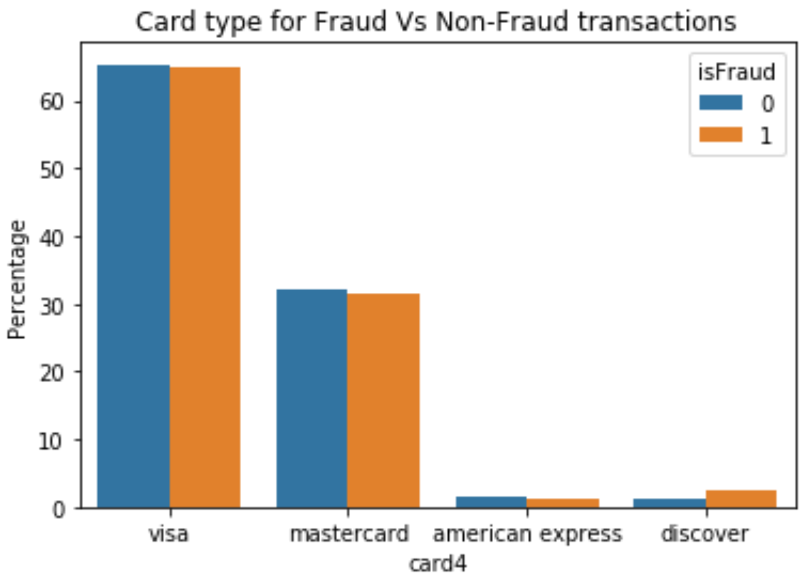
Out[79]: Text(0.5, 1.0, 'Card category for Fraud Vs Non-Fraud transactions')



The above graph for card6 is plotted based on percentage of relative frequency. Although there're more fraud transactions for debit card, comparing to total number of people who use credit card we can consider that there're more fraud transactions happening in credit card.

In [80]:
```
merge_plot = (merged_required["card4"].groupby(merged_required["isFrau
d"]).value_counts(normalize=True).mul(100).rename("Percentage")
 .reset_index())
sns.barplot(x="card4", y="Percentage", hue="isFraud", data=merge_plot)

plt.title('Card type for Fraud Vs Non-Fraud transactions')
```

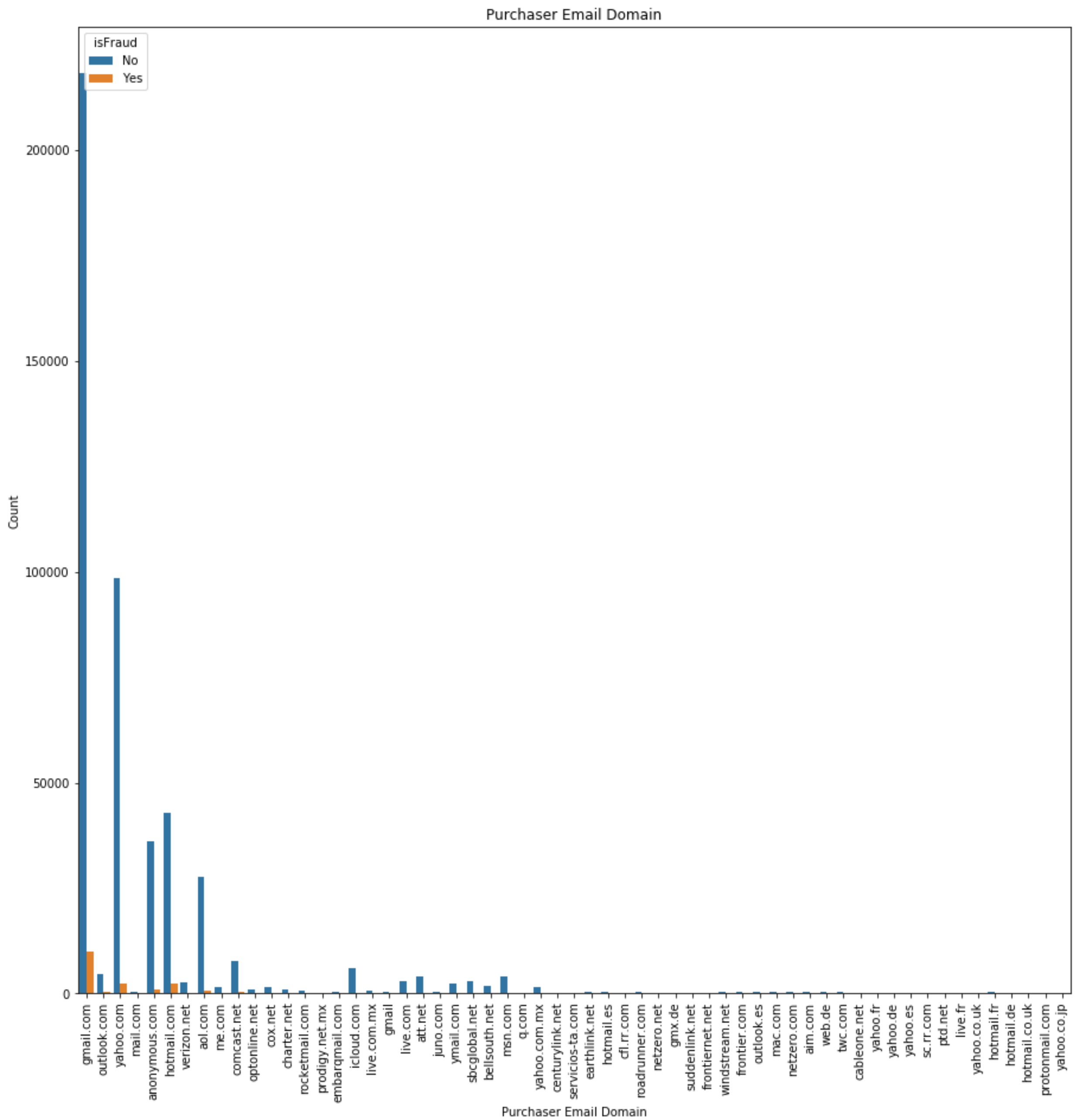
Out[80]:
```
Text(0.5, 1.0, 'Card type for Fraud Vs Non-Fraud transactions')
```



Percentage of fraud transactions (60%) is higher for the transactions payed by Visa card compared to mastercard.

In [81]:
```
# Reference: https://seaborn.pydata.org/generated/seaborn.countplot.ht
ml
fig = plt.figure(figsize=(15,15))
plot1=sns.countplot(x='P_emaildomain', data=merged_required,hue='isFra
ud')
plot1.set_title('Purchaser Email Domain')
plot1.set_xticklabels(plot1.get_xticklabels(),rotation=90)
plot1.set_xlabel("Purchaser Email Domain")
plot1.set_ylabel("Count")
plot1.legend(title='isFraud', labels=['No','Yes'])
```

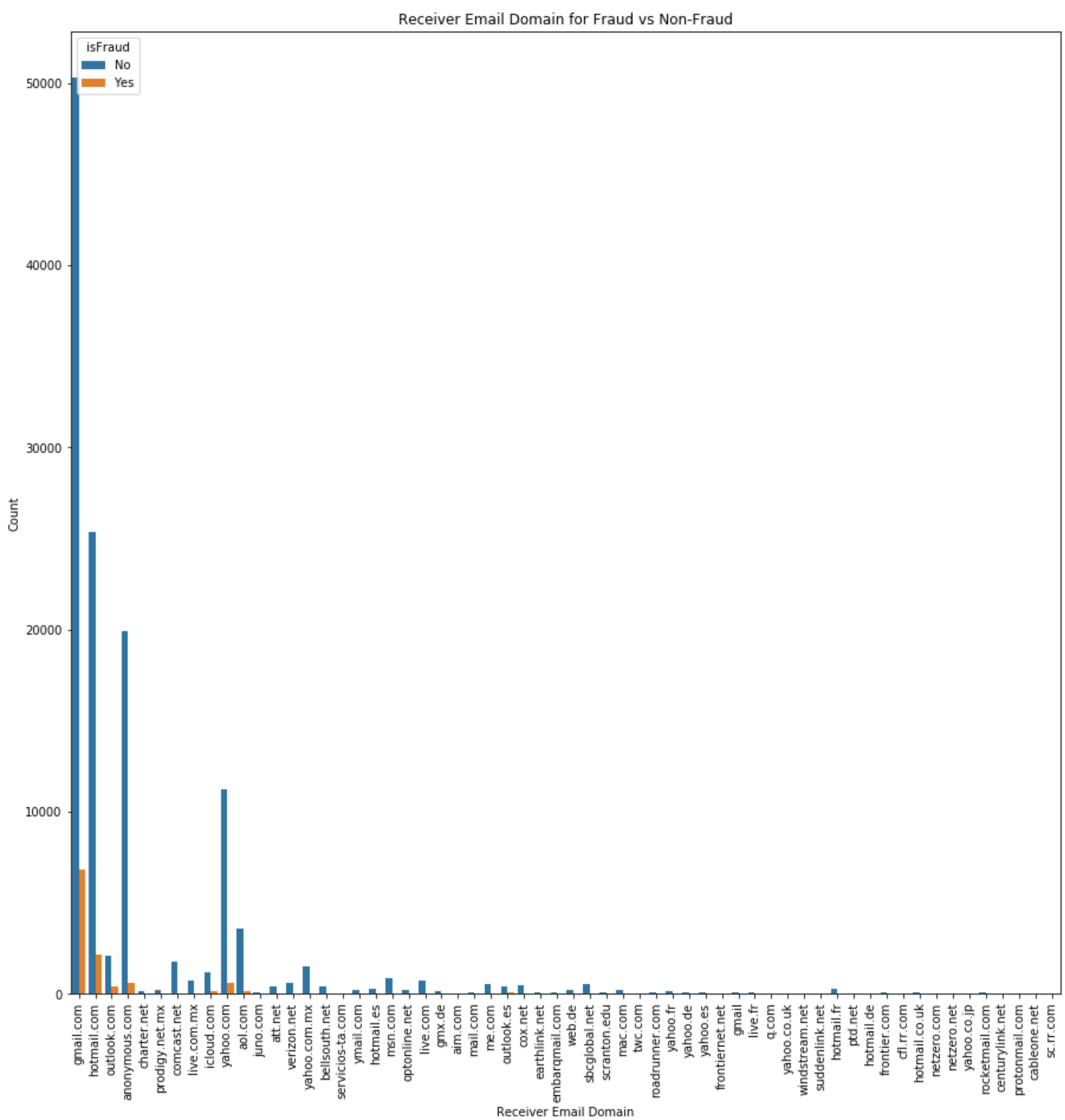
Out[81]: <matplotlib.legend.Legend at 0x1a25fadd30>



Most of the fraud trasactions for purchaser_email_domain and receiver_email_domain are from gmail.com. There's not much interesting insight in this, as lot of people use gmail.com domain. From the above and below graph we can see that very less percentage of frauds happening comapred to non-fraud transactions for gmail.com domain.

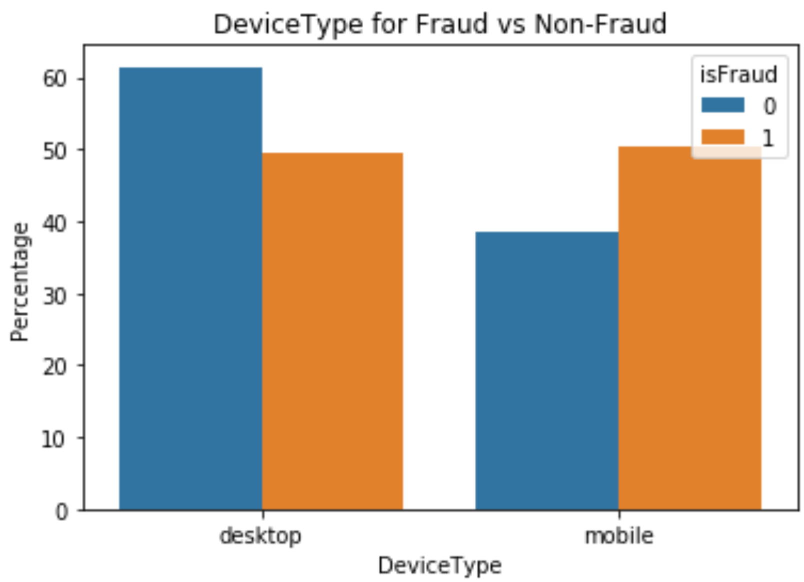
In [82]:
```
# Reference: https://seaborn.pydata.org/generated/seaborn.countplot.ht
ml
fig = plt.figure(figsize=(15,15))
plot1=sns.countplot(x='R_emaildomain', data=merged_required,hue='isFra
ud')
plot1.set_title('Receiver Email Domain for Fraud vs Non-Fraud')
plot1.set_xticklabels(plot1.get_xticklabels(),rotation=90)
plot1.set_xlabel("Receiver Email Domain")
plot1.set_ylabel("Count")
plot1.legend(title='isFraud', labels=['No','Yes'])
```

Out[82]: <matplotlib.legend.Legend at 0x1a2644ae48>



In [83]:
```
# Reference: https://seaborn.pydata.org/generated/seaborn.barplot.html
merge_plot = (merged_required["DeviceType"].groupby(merged_required["isFraud"]).value_counts(normalize=True).mul(100).rename("Percentage")
 .reset_index())
sns.barplot(x="DeviceType", y="Percentage", hue="isFraud", data=merge_plot)
plt.title('DeviceType for Fraud vs Non-Fraud')
```

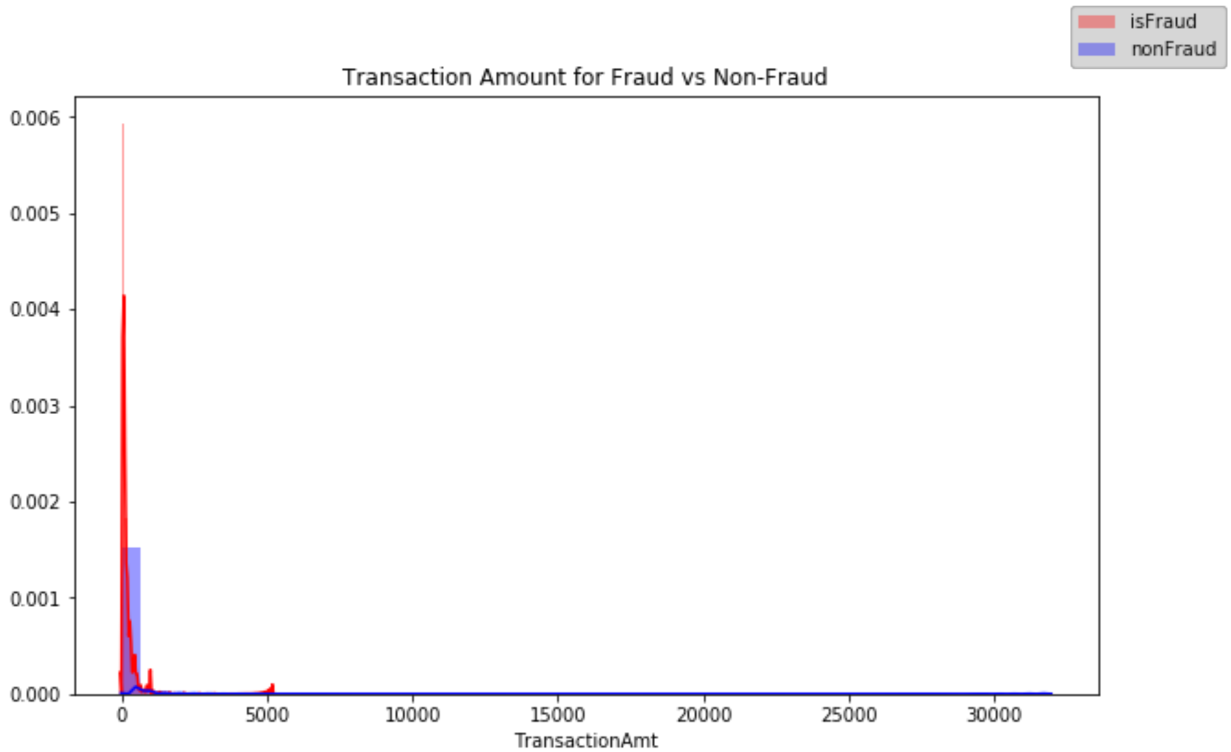
Out[83]: Text(0.5, 1.0, 'DeviceType for Fraud vs Non-Fraud')



The above graph for DeviceType is plotted based on percentage of relative frequency. Although there're equal percentage of fraud transactions for both mobile and desktop, comparing to total number of people who use mobile we can consider that there're more fraud transactions happening to the people who use mobile device-type.

In [84]:
```
fig = plt.figure(figsize=(10,6))
sns.distplot(isFradTrans["TransactionAmt"] , color="red", label="Fraud
")
sns.distplot(nonFraudTrans["TransactionAmt"] , color="blue", label="No
nFraud")
fig.legend(labels=['isFraud','nonFraud'])
plt.title('Transaction Amount for Fraud vs Non-Fraud')
```

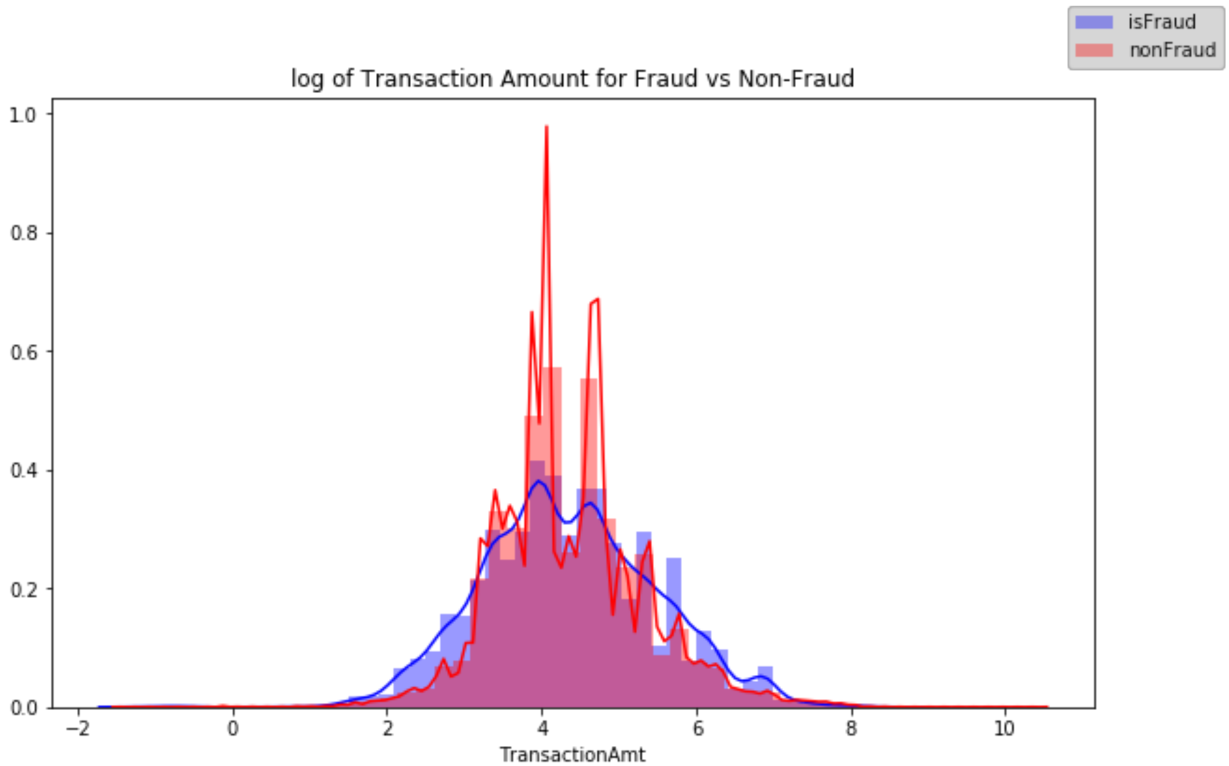
Out[84]:
```
Text(0.5, 1.0, 'Transaction Amount for Fraud vs Non-Fraud')
```



From the above graph, we can infer that cheap products transactions leads to more fraud. As we can see that most of fraud transactions happened for the transaction amount ranging between $200-300$

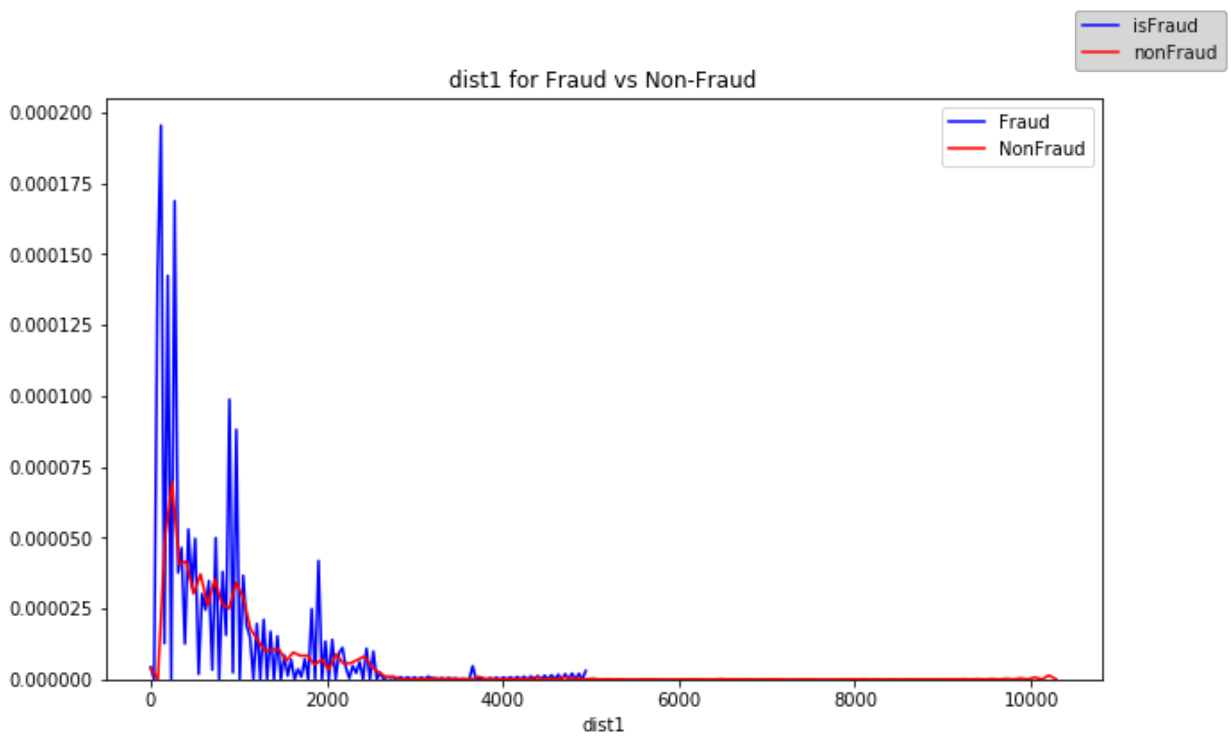
In [85]:
```
fig = plt.figure(figsize=(10,6))
sns.distplot(np.log(isFradTrans["TransactionAmt"]) , color="blue", lab
el="Fraud")
sns.distplot(np.log(nonFraudTrans["TransactionAmt"]) , color="red", la
bel="NonFraud")
fig.legend(labels=['isFraud','nonFraud'])
plt.title('log of Transaction Amount for Fraud vs Non-Fraud')
```

Out[85]: Text(0.5, 1.0, 'log of Transaction Amount for Fraud vs Non-Fraud')



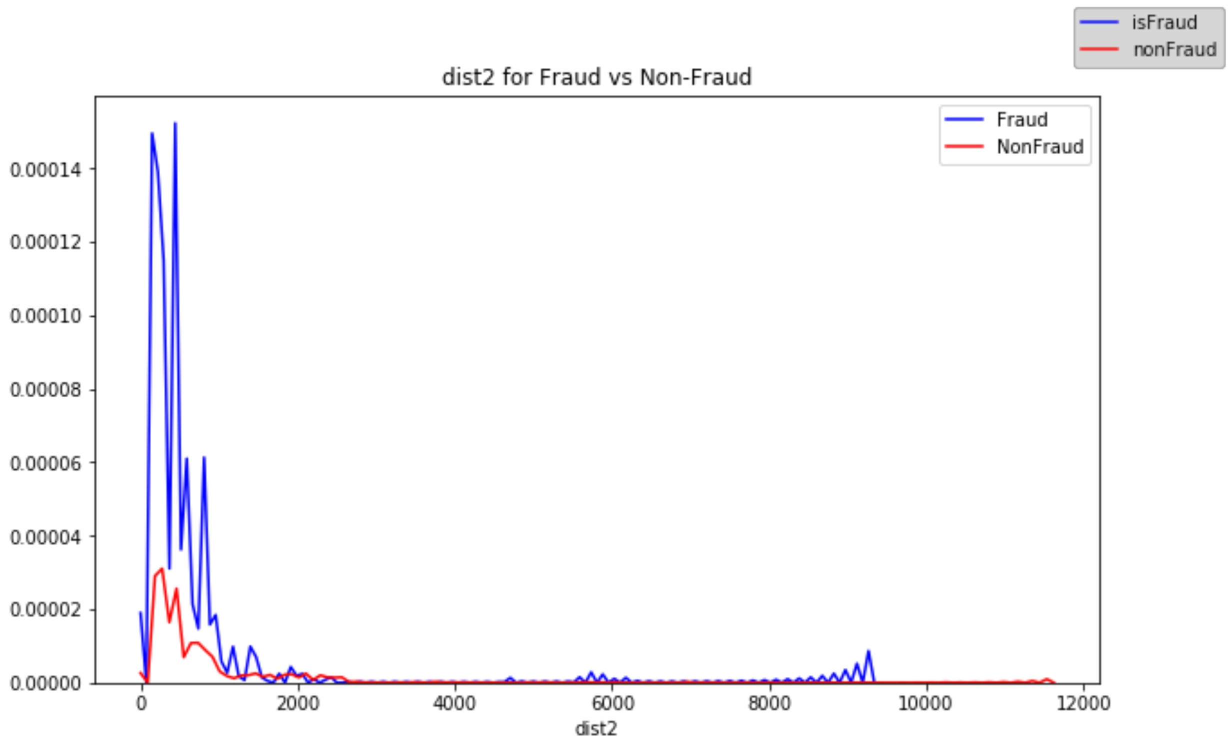
In [99]:
```
warnings.filterwarnings('ignore')
isFradTrans['dist1'].fillna(np.nanmedian(merged_required['dist1']), in
place=True)
isFradTrans['dist2'].fillna(np.nanmedian(merged_required['dist2']), in
place=True)
nonFraudTrans['dist1'].fillna(np.nanmedian(merged_required['dist1']),
inplace=True)
nonFraudTrans['dist2'].fillna(np.nanmedian(merged_required['dist2']),
inplace=True)
fig = plt.figure(figsize=(10,6))
sns.distplot(isFradTrans["dist1"] , color="blue", hist= False,label="F
raud")
sns.distplot(nonFraudTrans["dist1"] , color="red", hist=False,label="N
onFraud")
fig.legend(labels=['isFraud','nonFraud'])
plt.title('dist1 for Fraud vs Non-Fraud')
```

Out[99]: Text(0.5, 1.0, 'dist1 for Fraud vs Non-Fraud')



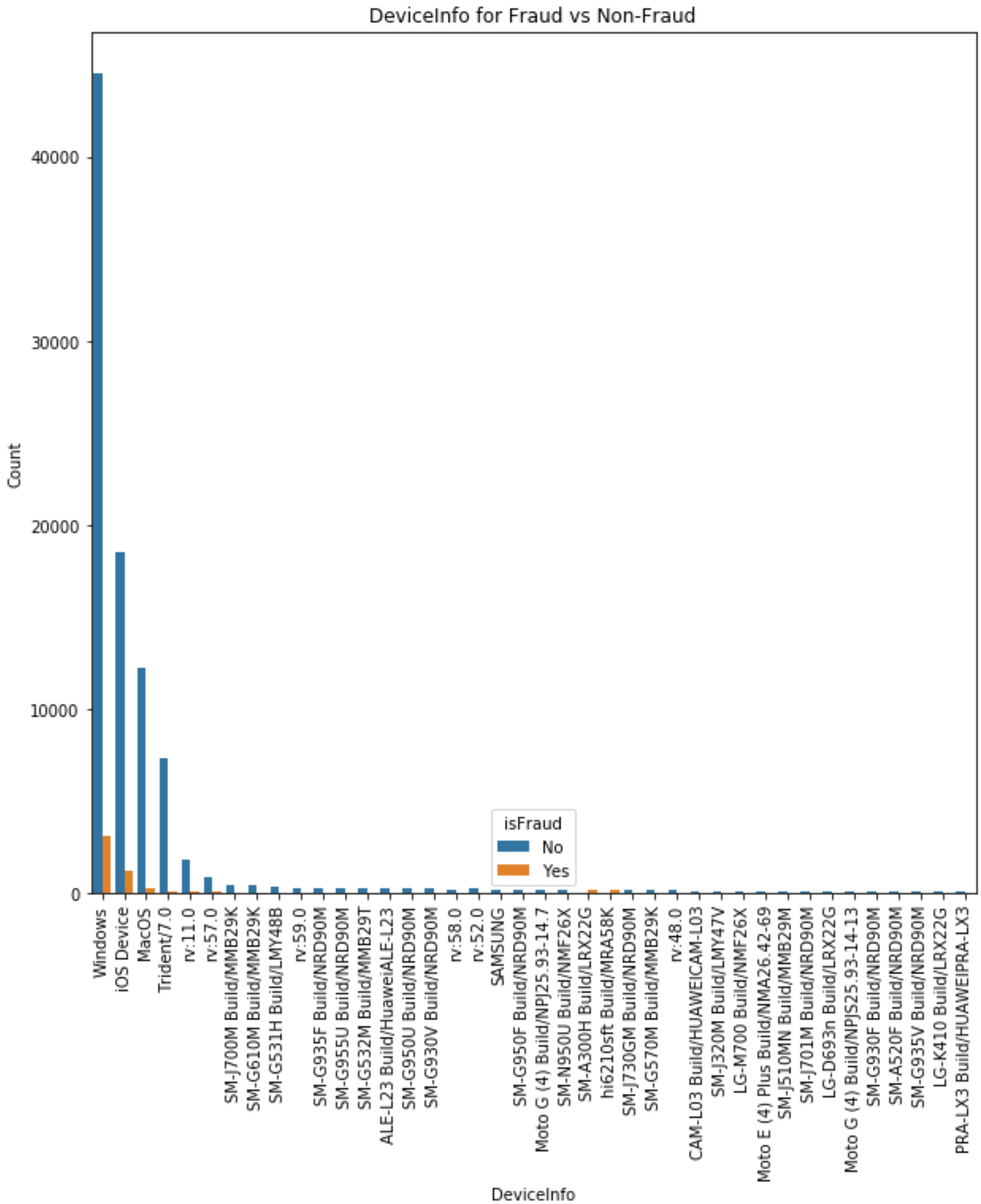
In [100]:
```
fig = plt.figure(figsize=(10,6))
sns.distplot(isFradTrans["dist2"] , color="blue", hist = False,label="
Fraud")
sns.distplot(nonFraudTrans["dist2"] , color="red", hist = False,label=
"NonFraud")
fig.legend(labels=['isFraud','nonFraud'])
plt.title('dist2 for Fraud vs Non-Fraud')
```

Out[100]: Text(0.5, 1.0, 'dist2 for Fraud vs Non-Fraud')



In [91]:
```
# As the number of labels are large, considering only top 40
# Reference: https://seaborn.pydata.org/generated/seaborn.countplot.ht
ml
fig = plt.figure(figsize=(10,10))
plot1=sns.countplot(x='DeviceInfo', data=merged_required,order = merge
d_required['DeviceInfo'].value_counts().iloc[:40].index,hue='isFraud')
plot1.set_title('DeviceInfo for Fraud vs Non-Fraud')
plot1.set_xticklabels(plot1.get_xticklabels(),rotation=90)
plot1.set_xlabel("DeviceInfo")
plot1.set_ylabel("Count")
plot1.legend(title='isFraud', labels=['No','Yes'])
```

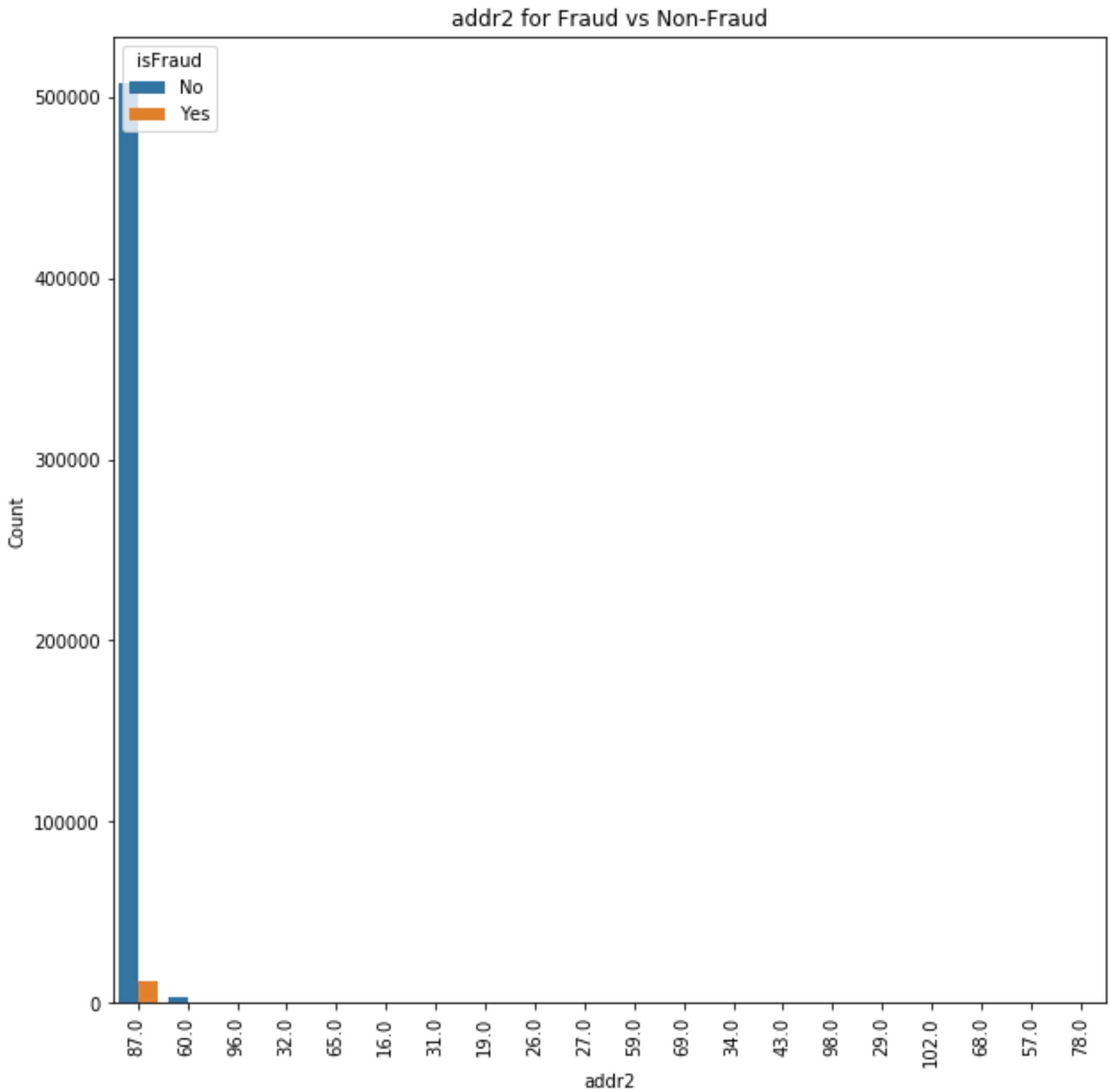
Out[91]: <matplotlib.legend.Legend at 0x1a27917320>



There's nothing much interesting about DeviceInfo. But comparitively, there're lot of windows users where fraud transactions are happening.

In [92]:
```
# though this is numerical data, lot of transactions is under 1 countr
y code. So considered countplot for top 20.
fig = plt.figure(figsize=(10,10))
plot1=sns.countplot(x='addr2', data=merged_required,order = merged_req
uired['addr2'].value_counts().iloc[:20].index,hue='isFraud')
plot1.set_title('addr2 for Fraud vs Non-Fraud')
plot1.set_xticklabels(plot1.get_xticklabels(),rotation=90)
plot1.set_xlabel("addr2")
plot1.set_ylabel("Count")
plot1.legend(title='isFraud', labels=['No','Yes'])
```

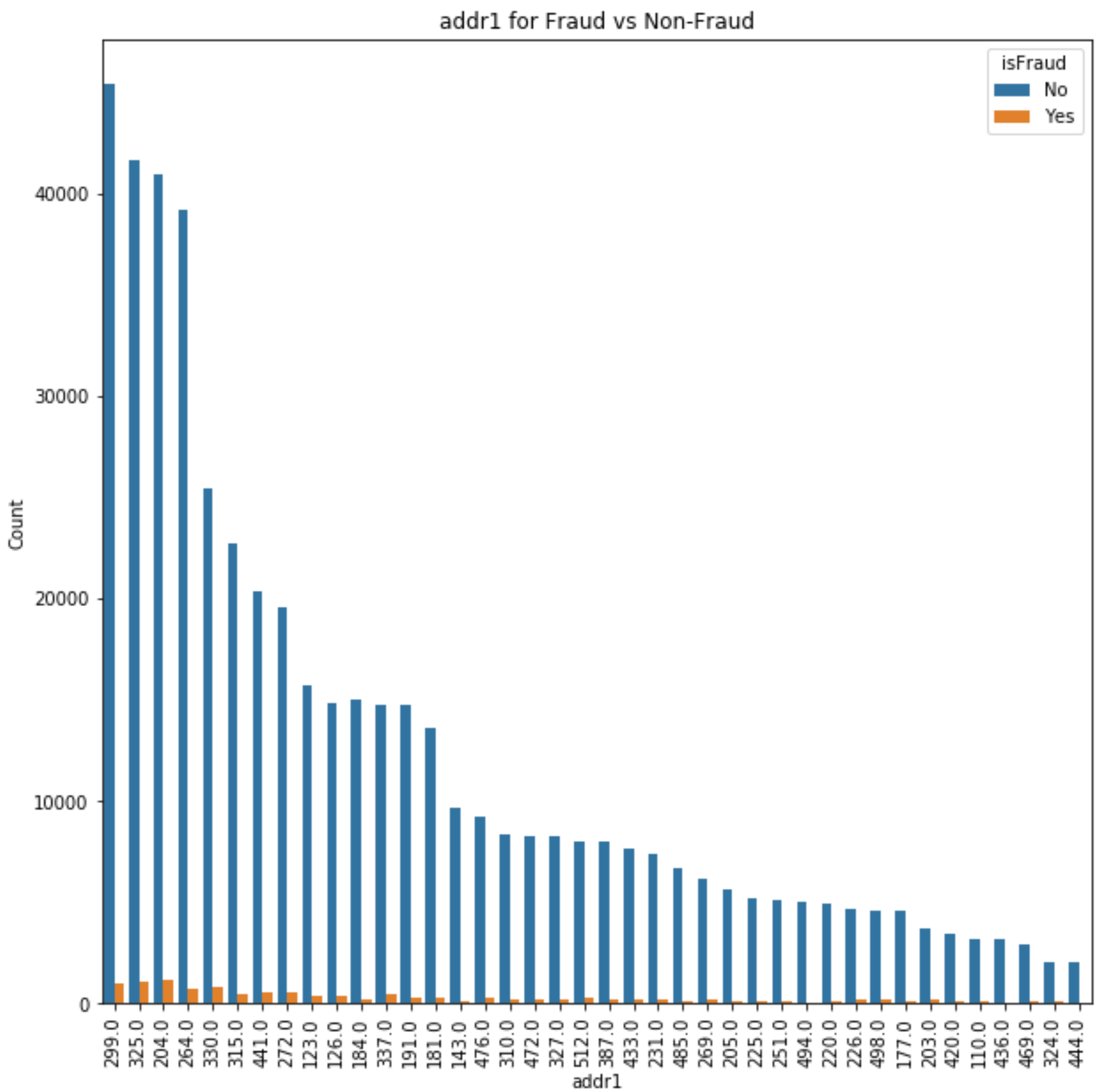
Out[92]: <matplotlib.legend.Legend at 0x1a27a4c7f0>



Most of the transactions are happening from country code 87. So all the fraud users from country code 87.

In [93]:
```
fig = plt.figure(figsize=(10,10))
plot1=sns.countplot(x='addr1', data=merged_required,order = merged_req
uired['addr1'].value_counts().iloc[:40].index,hue='isFraud')
plot1.set_title('addr1 for Fraud vs Non-Fraud')
plot1.set_xticklabels(plot1.get_xticklabels(),rotation=90)
plot1.set_xlabel("addr1")
plot1.set_ylabel("Count")
plot1.legend(title='isFraud', labels=['No','Yes'])
```

Out[93]: <matplotlib.legend.Legend at 0x1a29127ef0>



In [ ]:

Part 2 - Transaction Frequency

In [94]:
```
#Converting TransactionDT to timestamp. (Day, Hour)

merged_required['day'] = pd.to_datetime(merged_required['TransactionDT
'], unit='s').dt.day
merged_required['hours'] = pd.to_datetime(merged_required['Transaction
DT'], unit='s').dt.hour
```

In [21]:
```
print(merged_required.tail(15))
```

```
          TransactionID  isFraud  TransactionDT  TransactionAmt Produc
tCD  \
590525        3577525        0       15810866          57.950
W
590526        3577526        1       15810876         250.000
R
590527        3577527        0       15810883         189.950
W
590528        3577528        0       15810907         279.950
W
590529        3577529        0       15810912          73.838
C
590530        3577530        0       15810926         400.780
W
590531        3577531        0       15810935         400.000
R
590532        3577532        0       15811007         204.970
W
590533        3577533        0       15811029         107.950
W
590534        3577534        0       15811030          67.505
C
590535        3577535        0       15811047          49.000
W
590536        3577536        0       15811049          39.500
W
590537        3577537        0       15811079          30.950
W
590538        3577538        0       15811088         117.000
W
590539        3577539        0       15811131         279.950
W
```

```
        card1  card2  card3       card4  card5  ...      id_33  \
590525  11942  570.0  150.0        visa  226.0  ...        NaN
590526   1214  174.0  150.0        visa  226.0  ...    855x480
590527   6453  555.0  150.0        visa  226.0  ...        NaN
590528  15066  170.0  150.0  mastercard  102.0  ...        NaN
590529   5096  555.0  185.0  mastercard  137.0  ...        NaN
590530  15066  170.0  150.0  mastercard  102.0  ...        NaN
590531   6019  583.0  150.0        visa  226.0  ...  2560x1600
590532  12037  595.0  150.0  mastercard  224.0  ...        NaN
590533  13071  321.0  150.0        visa  226.0  ...        NaN
590534   5812  408.0  185.0  mastercard  224.0  ...        NaN
590535   6550    NaN  150.0        visa  226.0  ...        NaN
590536  10444  225.0  150.0  mastercard  224.0  ...        NaN
590537  12037  595.0  150.0  mastercard  224.0  ...        NaN
590538   7826  481.0  150.0  mastercard  224.0  ...        NaN
590539  15066  170.0  150.0  mastercard  102.0  ...        NaN

                id_34  id_35  id_36  id_37 id_38 DeviceType  \
590525            NaN    NaN    NaN    NaN   NaN        NaN
590526  match_status:2      T      F      T     F     mobile
590527            NaN    NaN    NaN    NaN   NaN        NaN
590528            NaN    NaN    NaN    NaN   NaN        NaN
590529            NaN      F      F      T     F     mobile
590530            NaN    NaN    NaN    NaN   NaN        NaN
590531  match_status:2      T      F      T     F    desktop
590532            NaN    NaN    NaN    NaN   NaN        NaN
590533            NaN    NaN    NaN    NaN   NaN        NaN
590534            NaN      F      F      T     F     mobile
590535            NaN    NaN    NaN    NaN   NaN        NaN
590536            NaN    NaN    NaN    NaN   NaN        NaN
590537            NaN    NaN    NaN    NaN   NaN        NaN
590538            NaN    NaN    NaN    NaN   NaN        NaN
590539            NaN    NaN    NaN    NaN   NaN        NaN

                               DeviceInfo  day  hours
590525                                NaN    2     23
590526                 A574BL Build/NMF26F    2     23
590527                                NaN    2     23
590528                                NaN    2     23
590529  Moto E (4) Plus Build/NMA26.42-152    2     23
590530                                NaN    2     23
590531                              MacOS    2     23
590532                                NaN    2     23
590533                                NaN    2     23
590534       RNE-L03 Build/HUAWEIRNE-L03    2     23
590535                                NaN    2     23
590536                                NaN    2     23
590537                                NaN    2     23
590538                                NaN    2     23
590539                                NaN    2     23
```

```
[15 rows x 436 columns]
```

In [22]:
```
merged_required['addr2'].value_counts()
```

Out[22]:
```
87.0     520481
60.0       3084
96.0        638
32.0         91
65.0         82
16.0         55
31.0         47
19.0         33
26.0         25
27.0         20
59.0         17
69.0         17
34.0         16
43.0         12
98.0         11
29.0         11
102.0        11
68.0         10
57.0         10
78.0          8
10.0          8
17.0          7
71.0          7
13.0          7
54.0          6
72.0          6
88.0          5
52.0          5
73.0          5
21.0          5
          ...
24.0          3
20.0          3
74.0          3
92.0          2
36.0          2
23.0          2
76.0          2
86.0          2
100.0         2
63.0          2
97.0          2
66.0          2
77.0          1
84.0          1
```

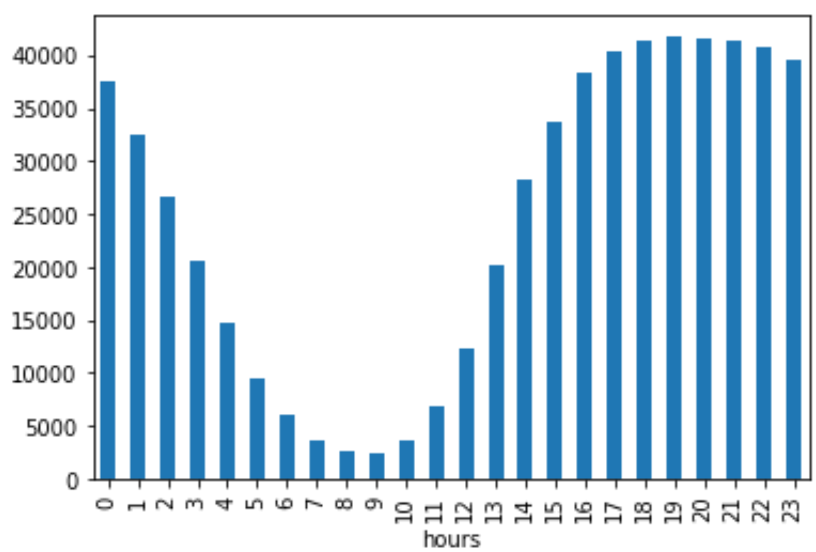
```
35.0          1
22.0          1
94.0          1
93.0          1
15.0          1
89.0          1
75.0          1
25.0          1
14.0          1
83.0          1
82.0          1
55.0          1
79.0          1
49.0          1
50.0          1
70.0          1
Name: addr2, Length: 74, dtype: int64
```

In [128]:

```
# Most frequent country code is 87.0 as per addr2 field. Plot graph fo
r hours vs frequency count.
plot1 = merged_required[merged_required['addr2']==87.0]['hours'].value
_counts().sort_index()
plot1.plot(kind='bar')
plt.xlabel('hours')
```

Out[128]:

```
Text(0.5, 0, 'hours')
```

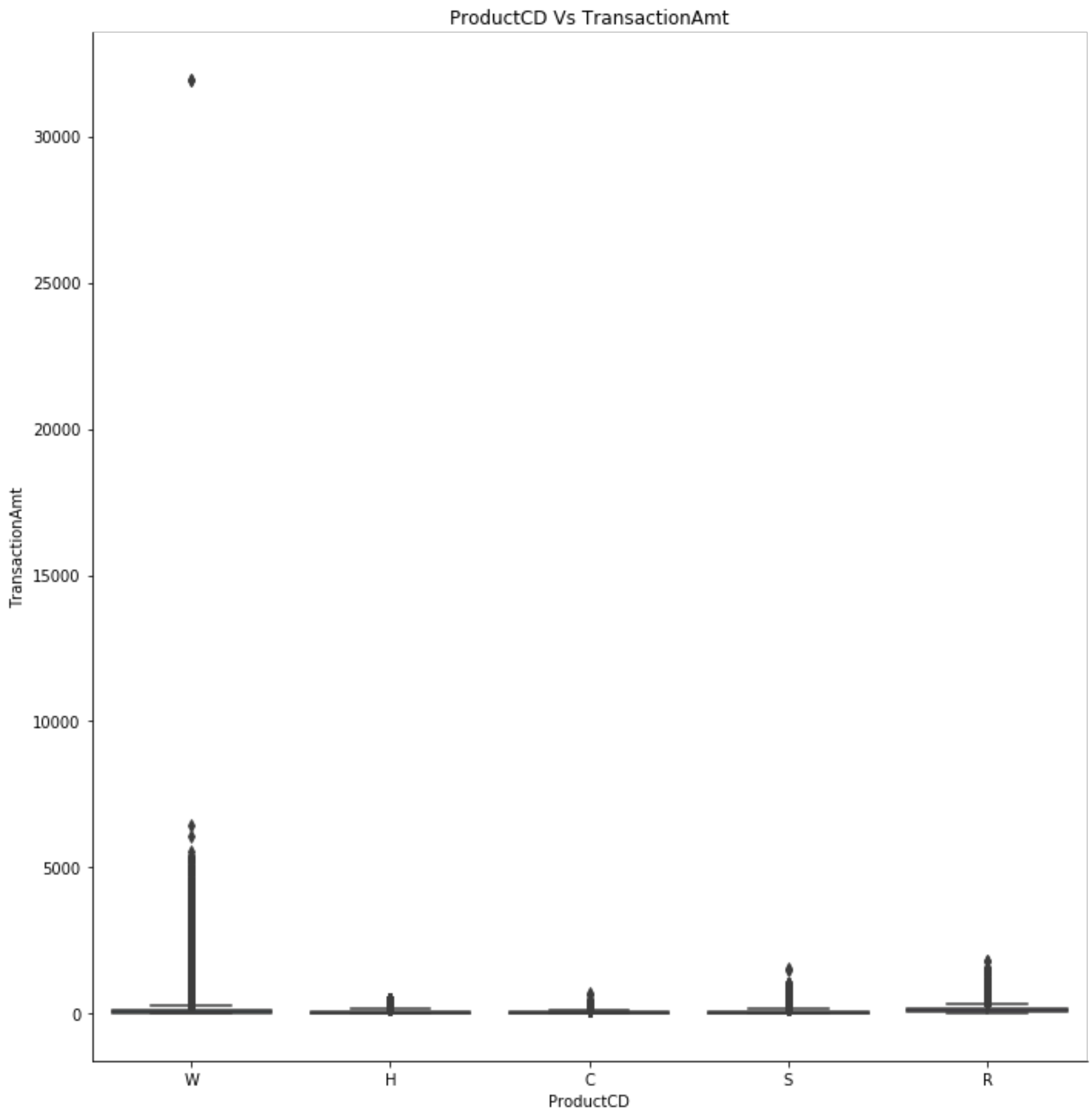


Most frequent country code in addr2 field is 87.0 which has 520481 entries. From the above graph, most of transactions are happening from evening to midnight(17th hour to 24rth hour). Waking hours might be around 10 in the morning as number of transactions started incresing from 10th hour. Sleeping hours can be around 3 to 9 as there're less number of transactions.

Part 3 - Product Code

In [96]:
```
# Reference: https://seaborn.pydata.org/generated/seaborn.catplot.html
sns.catplot(x="ProductCD", y="TransactionAmt", hue="ProductCD",
            kind="box", dodge=False, data=merged_required, height=10)
plt.title('ProductCD Vs TransactionAmt')
```

Out[96]:
```
Text(0.5, 1, 'ProductCD Vs TransactionAmt')
```



In [25]:
```
w_mean = merged_required[merged_required['ProductCD'] == 'W']['Transac
tionAmt'].mean()
print('For ProductCD W: ',w_mean)
h_mean = merged_required[merged_required['ProductCD'] == 'H']['Transac
tionAmt'].mean()
print('For ProductCD H: ', h_mean)
c_mean = merged_required[merged_required['ProductCD'] == 'C']['Transac
tionAmt'].mean()
print('For ProductCD C: ',c_mean)
s_mean = merged_required[merged_required['ProductCD'] == 'S']['Transac
tionAmt'].mean()
print('For ProductCD S: ',s_mean)
r_mean = merged_required[merged_required['ProductCD'] == 'R']['Transac
tionAmt'].mean()
print('For ProductCD R: ',r_mean)
```

```
For ProductCD W:  153.15855385223293
For ProductCD H:  73.17005813953489
For ProductCD C:  42.872353113733446
For ProductCD S:  60.269487444100434
For ProductCD R:  168.30618849306347
```

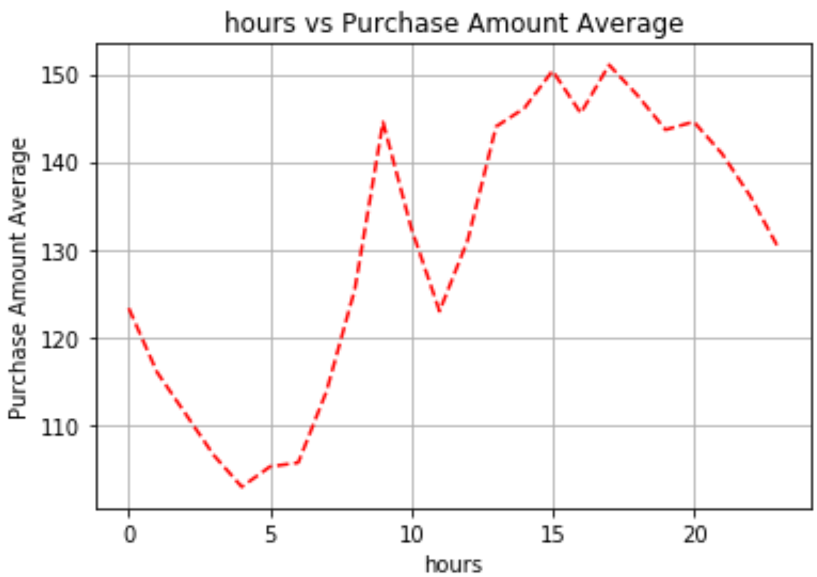
From the above bar plot and calculation of mean, ProductCD 'R' and 'W' has most expensive products followed by H, S and C.

Part 4 - Correlation Coefficient

In [101]:
```
np.corrcoef(merged_required['TransactionAmt'], merged_required['hours'
])
```

Out[101]:
```
array([[1.        , 0.04453236],
       [0.04453236, 1.        ]])
```

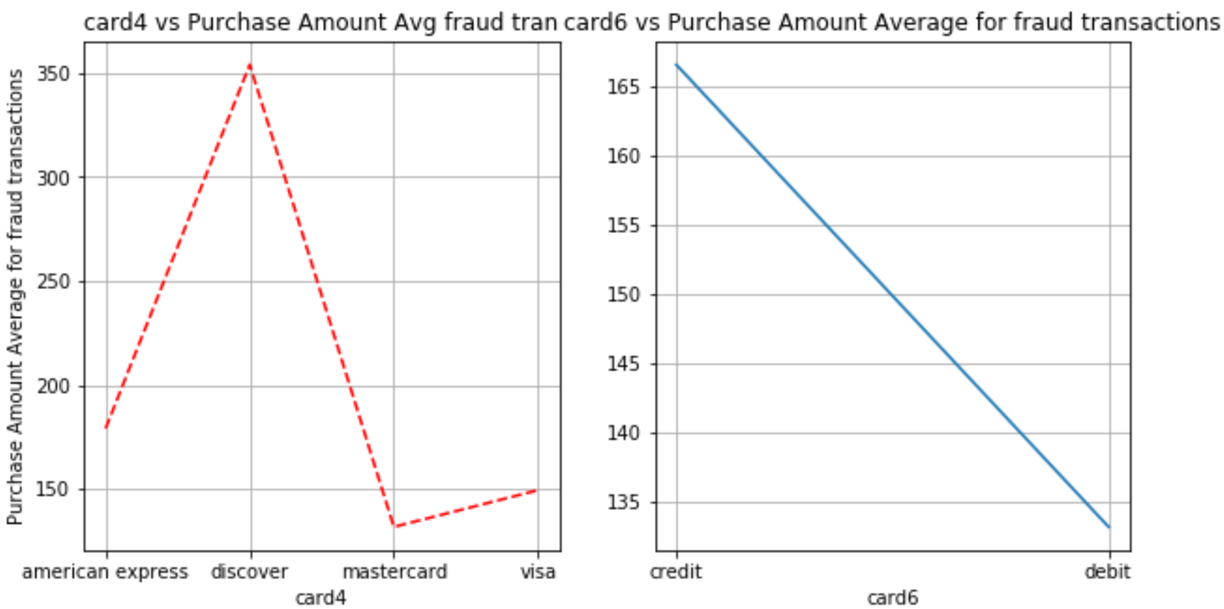
In [102]:
```
result_mean = merged_required['TransactionAmt'].groupby(merged_require
d['hours']).mean()
plt.plot(result_mean,'r--')
plt.xlabel('hours')
plt.ylabel('Purchase Amount Average')
plt.grid()
plt.title('hours vs Purchase Amount Average')
plt.show()
```



From the above plot, purchase amount is higher in the interval 14 hours to 17 hours and purchase amount is less between 3 to 7 hours. We can also assume those can be sleeping hours where purchase amount is less. Correlation coefficient for transaction amount and hours is 0.4453

Part 5 - Interesting Plot

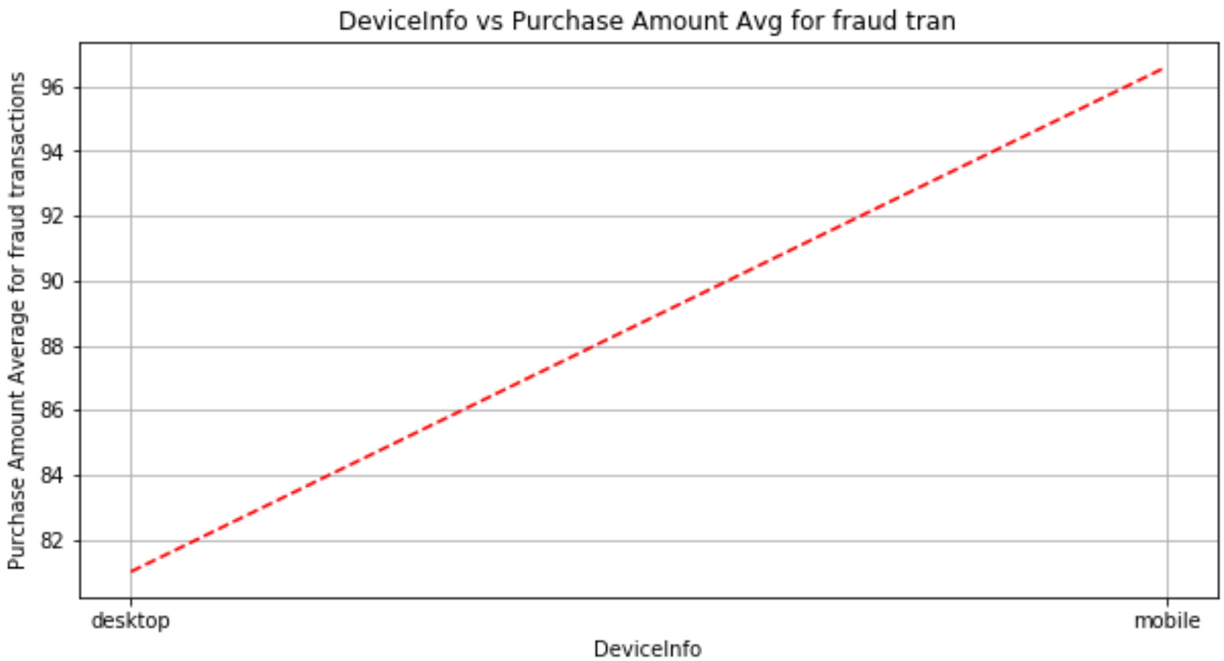
In [103]:
```
plt.figure(figsize=(10,5))
card4_mean = merged_required[merged_required['isFraud'] == 1]['TransactionAmt'].groupby(merged_required['card4']).mean()
card6_mean = merged_required[merged_required['isFraud'] == 1]['TransactionAmt'].groupby(merged_required['card6']).mean()
plt.subplot(1, 2, 1)
plt.plot(card4_mean,'r--')
plt.xlabel('card4')
plt.ylabel('Purchase Amount Average for fraud transactions')
plt.grid()
plt.title('card4 vs Purchase Amount Avg fraud tran')
plt.subplot(1, 2, 2)
plt.plot(card6_mean)
plt.xlabel('card6')
plt.grid()
plt.title('card6 vs Purchase Amount Average for fraud transactions')
plt.show()
```



Purchase amount average for fraud transactions is higher for discover card4 type and credit card of card6 type.

In [104]:
```
plt.figure(figsize=(10,5))
devicetype_mean = merged_required[merged_required['isFraud'] == 1]['Tr
ansactionAmt'].groupby(merged_required['DeviceType']).mean()
plt.plot(devicetype_mean,'r--')
plt.xlabel('DeviceInfo')
plt.ylabel('Purchase Amount Average for fraud transactions')
plt.grid()
plt.title('DeviceInfo vs Purchase Amount Avg for fraud tran')
```

Out[104]:
```
Text(0.5, 1.0, 'DeviceInfo vs Purchase Amount Avg for fraud tran')
```



Purchase amount average for fraud transactions is higher for mobile device type compared to transactions done by desktop.

Part 6 - Prediction Model

1. Done outer join to merge on transaction id, so that 4,00,000 entries are not missed.
2. Observed number of non-null values in all available columns.
3. Dropped columns which have more than 80 % of NAN values. 'id_23','id_27'
4. NAN Handling of Numerical Data: Replaced NAN's with median value of each column.
5. NAN Handling of Categorical Data: Replaced NAN's with an alphabet for each column.
6. Conversion of Categorical to Numerical: For features like: 'M1','M2','M3', 'M4', 'M5', 'M6','M7','M8','M9','ProductCD','card4','card6','DeviceType','id_12','id_15','id_16','id_28','id_29','id_35', : Label encoding is applied.
7. Split the training data to train and test with 80-20 ratio.
8. Baseline Model: Linear Regression with RMSE 0.163. Trained the model with train data and calculated RMSE with test data.
9. Applied Random Forest Regression
10. Applied XgBoost Regression

In [105]:
```
merged_required.info(verbose=True, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 590540 entries, 0 to 590539
Data columns (total 436 columns):
TransactionID     590540 non-null int64
isFraud           590540 non-null int64
TransactionDT     590540 non-null int64
TransactionAmt    590540 non-null float64
ProductCD         590540 non-null object
card1             590540 non-null int64
card2             581607 non-null float64
card3             588975 non-null float64
card4             588963 non-null object
card5             586281 non-null float64
card6             588969 non-null object
addr1             524834 non-null float64
addr2             524834 non-null float64
dist1             238269 non-null float64
dist2             37627 non-null float64
P_emaildomain     496084 non-null object
R_emaildomain     137291 non-null object
C1                590540 non-null float64
C2                590540 non-null float64
C3                590540 non-null float64
C4                590540 non-null float64
C5                590540 non-null float64
C6                590540 non-null float64
C7                590540 non-null float64
C8                590540 non-null float64
C9                590540 non-null float64
```

```
C10              590540 non-null float64
C11              590540 non-null float64
C12              590540 non-null float64
C13              590540 non-null float64
C14              590540 non-null float64
D1               589271 non-null float64
D2               309743 non-null float64
D3               327662 non-null float64
D4               421618 non-null float64
D5               280699 non-null float64
D6               73187 non-null float64
D7               38917 non-null float64
D8               74926 non-null float64
D9               74926 non-null float64
D10              514518 non-null float64
D11              311253 non-null float64
D12              64717 non-null float64
D13              61952 non-null float64
D14              62187 non-null float64
D15              501427 non-null float64
M1               319440 non-null object
M2               319440 non-null object
M3               319440 non-null object
M4               309096 non-null object
M5               240058 non-null object
M6               421180 non-null object
M7               244275 non-null object
M8               244288 non-null object
M9               244288 non-null object
V1               311253 non-null float64
V2               311253 non-null float64
V3               311253 non-null float64
V4               311253 non-null float64
V5               311253 non-null float64
V6               311253 non-null float64
V7               311253 non-null float64
V8               311253 non-null float64
V9               311253 non-null float64
V10              311253 non-null float64
V11              311253 non-null float64
V12              514467 non-null float64
V13              514467 non-null float64
V14              514467 non-null float64
V15              514467 non-null float64
V16              514467 non-null float64
V17              514467 non-null float64
V18              514467 non-null float64
V19              514467 non-null float64
V20              514467 non-null float64
V21              514467 non-null float64
```

```
V22                514467 non-null float64
V23                514467 non-null float64
V24                514467 non-null float64
V25                514467 non-null float64
V26                514467 non-null float64
V27                514467 non-null float64
V28                514467 non-null float64
V29                514467 non-null float64
V30                514467 non-null float64
V31                514467 non-null float64
V32                514467 non-null float64
V33                514467 non-null float64
V34                514467 non-null float64
V35                421571 non-null float64
V36                421571 non-null float64
V37                421571 non-null float64
V38                421571 non-null float64
V39                421571 non-null float64
V40                421571 non-null float64
V41                421571 non-null float64
V42                421571 non-null float64
V43                421571 non-null float64
V44                421571 non-null float64
V45                421571 non-null float64
V46                421571 non-null float64
V47                421571 non-null float64
V48                421571 non-null float64
V49                421571 non-null float64
V50                421571 non-null float64
V51                421571 non-null float64
V52                421571 non-null float64
V53                513444 non-null float64
V54                513444 non-null float64
V55                513444 non-null float64
V56                513444 non-null float64
V57                513444 non-null float64
V58                513444 non-null float64
V59                513444 non-null float64
V60                513444 non-null float64
V61                513444 non-null float64
V62                513444 non-null float64
V63                513444 non-null float64
V64                513444 non-null float64
V65                513444 non-null float64
V66                513444 non-null float64
V67                513444 non-null float64
V68                513444 non-null float64
V69                513444 non-null float64
V70                513444 non-null float64
V71                513444 non-null float64
```

```
V72                 513444 non-null float64
V73                 513444 non-null float64
V74                 513444 non-null float64
V75                 501376 non-null float64
V76                 501376 non-null float64
V77                 501376 non-null float64
V78                 501376 non-null float64
V79                 501376 non-null float64
V80                 501376 non-null float64
V81                 501376 non-null float64
V82                 501376 non-null float64
V83                 501376 non-null float64
V84                 501376 non-null float64
V85                 501376 non-null float64
V86                 501376 non-null float64
V87                 501376 non-null float64
V88                 501376 non-null float64
V89                 501376 non-null float64
V90                 501376 non-null float64
V91                 501376 non-null float64
V92                 501376 non-null float64
V93                 501376 non-null float64
V94                 501376 non-null float64
V95                 590226 non-null float64
V96                 590226 non-null float64
V97                 590226 non-null float64
V98                 590226 non-null float64
V99                 590226 non-null float64
V100                590226 non-null float64
V101                590226 non-null float64
V102                590226 non-null float64
V103                590226 non-null float64
V104                590226 non-null float64
V105                590226 non-null float64
V106                590226 non-null float64
V107                590226 non-null float64
V108                590226 non-null float64
V109                590226 non-null float64
V110                590226 non-null float64
V111                590226 non-null float64
V112                590226 non-null float64
V113                590226 non-null float64
V114                590226 non-null float64
V115                590226 non-null float64
V116                590226 non-null float64
V117                590226 non-null float64
V118                590226 non-null float64
V119                590226 non-null float64
V120                590226 non-null float64
V121                590226 non-null float64
```

```
V122              590226 non-null float64
V123              590226 non-null float64
V124              590226 non-null float64
V125              590226 non-null float64
V126              590226 non-null float64
V127              590226 non-null float64
V128              590226 non-null float64
V129              590226 non-null float64
V130              590226 non-null float64
V131              590226 non-null float64
V132              590226 non-null float64
V133              590226 non-null float64
V134              590226 non-null float64
V135              590226 non-null float64
V136              590226 non-null float64
V137              590226 non-null float64
V138              81945 non-null float64
V139              81945 non-null float64
V140              81945 non-null float64
V141              81945 non-null float64
V142              81945 non-null float64
V143              81951 non-null float64
V144              81951 non-null float64
V145              81951 non-null float64
V146              81945 non-null float64
V147              81945 non-null float64
V148              81945 non-null float64
V149              81945 non-null float64
V150              81951 non-null float64
V151              81951 non-null float64
V152              81951 non-null float64
V153              81945 non-null float64
V154              81945 non-null float64
V155              81945 non-null float64
V156              81945 non-null float64
V157              81945 non-null float64
V158              81945 non-null float64
V159              81951 non-null float64
V160              81951 non-null float64
V161              81945 non-null float64
V162              81945 non-null float64
V163              81945 non-null float64
V164              81951 non-null float64
V165              81951 non-null float64
V166              81951 non-null float64
V167              139631 non-null float64
V168              139631 non-null float64
V169              139819 non-null float64
V170              139819 non-null float64
V171              139819 non-null float64
```

```
V172               139631 non-null float64
V173               139631 non-null float64
V174               139819 non-null float64
V175               139819 non-null float64
V176               139631 non-null float64
V177               139631 non-null float64
V178               139631 non-null float64
V179               139631 non-null float64
V180               139819 non-null float64
V181               139631 non-null float64
V182               139631 non-null float64
V183               139631 non-null float64
V184               139819 non-null float64
V185               139819 non-null float64
V186               139631 non-null float64
V187               139631 non-null float64
V188               139819 non-null float64
V189               139819 non-null float64
V190               139631 non-null float64
V191               139631 non-null float64
V192               139631 non-null float64
V193               139631 non-null float64
V194               139819 non-null float64
V195               139819 non-null float64
V196               139631 non-null float64
V197               139819 non-null float64
V198               139819 non-null float64
V199               139631 non-null float64
V200               139819 non-null float64
V201               139819 non-null float64
V202               139631 non-null float64
V203               139631 non-null float64
V204               139631 non-null float64
V205               139631 non-null float64
V206               139631 non-null float64
V207               139631 non-null float64
V208               139819 non-null float64
V209               139819 non-null float64
V210               139819 non-null float64
V211               139631 non-null float64
V212               139631 non-null float64
V213               139631 non-null float64
V214               139631 non-null float64
V215               139631 non-null float64
V216               139631 non-null float64
V217               130430 non-null float64
V218               130430 non-null float64
V219               130430 non-null float64
V220               141416 non-null float64
V221               141416 non-null float64
```

```
V222             141416 non-null float64
V223             130430 non-null float64
V224             130430 non-null float64
V225             130430 non-null float64
V226             130430 non-null float64
V227             141416 non-null float64
V228             130430 non-null float64
V229             130430 non-null float64
V230             130430 non-null float64
V231             130430 non-null float64
V232             130430 non-null float64
V233             130430 non-null float64
V234             141416 non-null float64
V235             130430 non-null float64
V236             130430 non-null float64
V237             130430 non-null float64
V238             141416 non-null float64
V239             141416 non-null float64
V240             130430 non-null float64
V241             130430 non-null float64
V242             130430 non-null float64
V243             130430 non-null float64
V244             130430 non-null float64
V245             141416 non-null float64
V246             130430 non-null float64
V247             130430 non-null float64
V248             130430 non-null float64
V249             130430 non-null float64
V250             141416 non-null float64
V251             141416 non-null float64
V252             130430 non-null float64
V253             130430 non-null float64
V254             130430 non-null float64
V255             141416 non-null float64
V256             141416 non-null float64
V257             130430 non-null float64
V258             130430 non-null float64
V259             141416 non-null float64
V260             130430 non-null float64
V261             130430 non-null float64
V262             130430 non-null float64
V263             130430 non-null float64
V264             130430 non-null float64
V265             130430 non-null float64
V266             130430 non-null float64
V267             130430 non-null float64
V268             130430 non-null float64
V269             130430 non-null float64
V270             141416 non-null float64
V271             141416 non-null float64
```

```
V272             141416 non-null float64
V273             130430 non-null float64
V274             130430 non-null float64
V275             130430 non-null float64
V276             130430 non-null float64
V277             130430 non-null float64
V278             130430 non-null float64
V279             590528 non-null float64
V280             590528 non-null float64
V281             589271 non-null float64
V282             589271 non-null float64
V283             589271 non-null float64
V284             590528 non-null float64
V285             590528 non-null float64
V286             590528 non-null float64
V287             590528 non-null float64
V288             589271 non-null float64
V289             589271 non-null float64
V290             590528 non-null float64
V291             590528 non-null float64
V292             590528 non-null float64
V293             590528 non-null float64
V294             590528 non-null float64
V295             590528 non-null float64
V296             589271 non-null float64
V297             590528 non-null float64
V298             590528 non-null float64
V299             590528 non-null float64
V300             589271 non-null float64
V301             589271 non-null float64
V302             590528 non-null float64
V303             590528 non-null float64
V304             590528 non-null float64
V305             590528 non-null float64
V306             590528 non-null float64
V307             590528 non-null float64
V308             590528 non-null float64
V309             590528 non-null float64
V310             590528 non-null float64
V311             590528 non-null float64
V312             590528 non-null float64
V313             589271 non-null float64
V314             589271 non-null float64
V315             589271 non-null float64
V316             590528 non-null float64
V317             590528 non-null float64
V318             590528 non-null float64
V319             590528 non-null float64
V320             590528 non-null float64
V321             590528 non-null float64
```

```
V322               82351 non-null float64
V323               82351 non-null float64
V324               82351 non-null float64
V325               82351 non-null float64
V326               82351 non-null float64
V327               82351 non-null float64
V328               82351 non-null float64
V329               82351 non-null float64
V330               82351 non-null float64
V331               82351 non-null float64
V332               82351 non-null float64
V333               82351 non-null float64
V334               82351 non-null float64
V335               82351 non-null float64
V336               82351 non-null float64
V337               82351 non-null float64
V338               82351 non-null float64
V339               82351 non-null float64
id_01              144233 non-null float64
id_02              140872 non-null float64
id_03              66324 non-null float64
id_04              66324 non-null float64
id_05              136865 non-null float64
id_06              136865 non-null float64
id_07              5155 non-null float64
id_08              5155 non-null float64
id_09              74926 non-null float64
id_10              74926 non-null float64
id_11              140978 non-null float64
id_12              144233 non-null object
id_13              127320 non-null float64
id_14              80044 non-null float64
id_15              140985 non-null object
id_16              129340 non-null object
id_17              139369 non-null float64
id_18              45113 non-null float64
id_19              139318 non-null float64
id_20              139261 non-null float64
id_21              5159 non-null float64
id_22              5169 non-null float64
id_23              5169 non-null object
id_24              4747 non-null float64
id_25              5132 non-null float64
id_26              5163 non-null float64
id_27              5169 non-null object
id_28              140978 non-null object
id_29              140978 non-null object
id_30              77565 non-null object
id_31              140282 non-null object
id_32              77586 non-null float64
```

```
id_33               73289 non-null object
id_34               77805 non-null object
id_35               140985 non-null object
id_36               140985 non-null object
id_37               140985 non-null object
id_38               140985 non-null object
DeviceType          140810 non-null object
DeviceInfo          118666 non-null object
day                 590540 non-null int64
hours               590540 non-null int64
dtypes: float64(399), int64(6), object(31)
memory usage: 1.9+ GB
```

In [106]:
```
merged_required.drop(['id_23','id_27','id_30','id_31','id_33','id_34']
, axis = 1, inplace=True)
```

In [107]:
```
dtype_groups = merged_required.columns.to_series().groupby(merged_requ
ired.dtypes).groups
dtype_groups
```

Out[107]:
```
{dtype('int64'): Index(['TransactionID', 'isFraud', 'TransactionDT',
'card1', 'day', 'hours'], dtype='object'),
 dtype('float64'): Index(['TransactionAmt', 'card2', 'card3', 'card5
', 'addr1', 'addr2', 'dist1',
        'dist2', 'C1', 'C2',
        ...
        'id_17', 'id_18', 'id_19', 'id_20', 'id_21', 'id_22', 'id_24
', 'id_25',
        'id_26', 'id_32'],
       dtype='object', length=399),
 dtype('O'): Index(['ProductCD', 'card4', 'card6', 'P_emaildomain',
'R_emaildomain', 'M1',
        'M2', 'M3', 'M4', 'M5', 'M6', 'M7', 'M8', 'M9', 'id_12', 'id
_15',
        'id_16', 'id_28', 'id_29', 'id_35', 'id_36', 'id_37', 'id_38
',
        'DeviceType', 'DeviceInfo'],
       dtype='object')}
```

In [108]:
```
for k, v in dtype_groups.items():
    if (k.name == 'int64' or k.name == 'float64'):
        for eachVal in v:
            merged_required[eachVal].fillna(np.nanmedian(merged_requir
ed[eachVal]), inplace=True)
```

In [109]:
```
merged_required["ProductCD"].fillna("T", inplace = True)
merged_required["card4"].fillna("U", inplace = True)
merged_required["card6"].fillna("V", inplace = True)
merged_required["P_emaildomain"].fillna("B", inplace = True)
merged_required["R_emaildomain"].fillna("X", inplace = True)
merged_required["DeviceInfo"].fillna("Y", inplace = True)
merged_required["DeviceType"].fillna("Z", inplace = True)
merged_required["M1"].fillna("A", inplace = True)
merged_required["M2"].fillna("A", inplace = True)
merged_required["M3"].fillna("A", inplace = True)
merged_required["M4"].fillna("A", inplace = True)
merged_required["M5"].fillna("A", inplace = True)
merged_required["M6"].fillna("A", inplace = True)
merged_required["M7"].fillna("A", inplace = True)
merged_required["M8"].fillna("A", inplace = True)
merged_required["M9"].fillna("A", inplace = True)


merged_required["id_12"].fillna("D", inplace = True)
merged_required["id_15"].fillna("E", inplace = True)
merged_required["id_16"].fillna("D", inplace = True)
merged_required["id_28"].fillna("I", inplace = True)
merged_required["id_29"].fillna("J", inplace = True)
merged_required["id_35"].fillna("A", inplace = True)
merged_required["id_36"].fillna("A", inplace = True)
merged_required["id_37"].fillna("A", inplace = True)
merged_required["id_38"].fillna("A", inplace = True)


print(merged_required.head(5))
```

```
   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_29  id_32  id_35  id_
36  id_37  \
0  361.0  150.0    discover  142.0  ...         J   24.0      A
A      A
1  404.0  150.0  mastercard  102.0  ...         J   24.0      A
A      A
2  490.0  150.0        visa  166.0  ...         J   24.0      A
A      A
3  567.0  150.0  mastercard  117.0  ...         J   24.0      A
A      A
4  514.0  150.0  mastercard  102.0  ...  NotFound   32.0      T
F      T

  id_38 DeviceType                     DeviceInfo  day  hours
0     A          Z                              Y    2      0
1     A          Z                              Y    2      0
2     A          Z                              Y    2      0
3     A          Z                              Y    2      0
4     T     mobile  SAMSUNG SM-G892A Build/NRD90M    2      0

[5 rows x 430 columns]
```

In [110]:
```
#References: https://scikit-learn.org/stable/modules/generated/sklearn
.preprocessing.LabelEncoder.html
from sklearn.preprocessing import LabelEncoder

label_encod = LabelEncoder()
for each_col in ['M1','M2','M3', 'M4', 'M5', 'M6', 'M7', 'M8', 'M9','P
roductCD','card4','card6','DeviceType',
                'id_12','id_15','id_16','id_28','id_29','id_35','id_36
','id_37','id_38']:
    each_col_unique = np.unique(merged_required[each_col])
    each_col_labels = label_encod.fit_transform(merged_required[each_c
ol])
    merged_required[each_col] = each_col_labels
merged_required.head(5)
```

Out[110]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	ca
0	2987000	0	86400	68.5	4	13926	361.0	150.0	
1	2987001	0	86401	29.0	4	2755	404.0	150.0	
2	2987002	0	86469	59.0	4	4663	490.0	150.0	
3	2987003	0	86499	50.0	4	18132	567.0	150.0	
4	2987004	0	86506	50.0	1	4497	514.0	150.0	

5 rows × 430 columns

In [111]:
```
merged_required.head(5)
```

Out[111]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	ca
0	2987000	0	86400	68.5	4	13926	361.0	150.0	
1	2987001	0	86401	29.0	4	2755	404.0	150.0	
2	2987002	0	86469	59.0	4	4663	490.0	150.0	
3	2987003	0	86499	50.0	4	18132	567.0	150.0	
4	2987004	0	86506	50.0	1	4497	514.0	150.0	

5 rows × 430 columns

In [112]:
```
# References: https://scikit-learn.org/stable/modules/generated/sklear
n.feature_extraction.FeatureHasher.html
from sklearn.feature_extraction import FeatureHasher

fh = FeatureHasher(n_features=6, input_type='string')
hashed_features = fh.fit_transform(merged_required['P_emaildomain'])
hashed_features = hashed_features.toarray()
merged_required = pd.concat([merged_required, pd.DataFrame(hashed_feat
ures,columns = ['ped1','ped2','ped3','ped4','ped5','ped6']
          )],axis=1)
merged_required.head(5)
```

Out[112]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	ca
0	2987000	0	86400	68.5	4	13926	361.0	150.0	
1	2987001	0	86401	29.0	4	2755	404.0	150.0	
2	2987002	0	86469	59.0	4	4663	490.0	150.0	
3	2987003	0	86499	50.0	4	18132	567.0	150.0	
4	2987004	0	86506	50.0	1	4497	514.0	150.0	

5 rows × 436 columns

In [113]:
```
hashed_features = fh.fit_transform(merged_required['R_emaildomain'])
hashed_features = hashed_features.toarray()
merged_required = pd.concat([merged_required, pd.DataFrame(hashed_feat
ures,columns = ['red1','red2','red3','red4','red5','red6']
          )],axis=1)
merged_required.head(5)
```

Out[113]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	ca
0	2987000	0	86400	68.5	4	13926	361.0	150.0	
1	2987001	0	86401	29.0	4	2755	404.0	150.0	
2	2987002	0	86469	59.0	4	4663	490.0	150.0	
3	2987003	0	86499	50.0	4	18132	567.0	150.0	
4	2987004	0	86506	50.0	1	4497	514.0	150.0	

5 rows × 442 columns

In [114]:
```
hashed_features = fh.fit_transform(merged_required['DeviceInfo'])
hashed_features = hashed_features.toarray()
merged_required = pd.concat([merged_required, pd.DataFrame(hashed_feat
ures,columns = ['devInfo1','devInfo2','devInfo3','devInfo4','devInfo5'
,'devInfo6']
             )],axis=1)
merged_required.head(5)
```

Out[114]:

	TransactionID	isFraud	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	ca
0	2987000	0	86400	68.5	4	13926	361.0	150.0	
1	2987001	0	86401	29.0	4	2755	404.0	150.0	
2	2987002	0	86469	59.0	4	4663	490.0	150.0	
3	2987003	0	86499	50.0	4	18132	567.0	150.0	
4	2987004	0	86506	50.0	1	4497	514.0	150.0	

5 rows × 448 columns

In [115]:
```
y_train = merged_required[['isFraud']]
```

In [116]:
```
merged_required.drop(['P_emaildomain','R_emaildomain','DeviceInfo','Tr
ansactionDT','isFraud'], axis = 1, inplace=True)
```

In [117]:
```
merged_required.shape
```

Out[117]: (590540, 443)

In [118]:
```
# References: https://scikit-learn.org/stable/modules/generated/sklear
n.model_selection.train_test_split.html
from sklearn.model_selection import train_test_split
xTrain, xTest, yTrain, yTest = train_test_split(merged_required, y_tra
in, test_size = 0.2, random_state = 0)
```

In [119]:
```
# baseline model - linear regression
from sklearn.linear_model import LinearRegression
from sklearn import metrics

model = LinearRegression().fit(xTrain, yTrain)
y_prediction = model.predict(xTest)
RMSE = np.sqrt(metrics.mean_squared_error(yTest, y_prediction))
RMSE
```

Out[119]: 0.1620350459752219

In [ ]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.R
andomForestRegressor.html
from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor(n_estimators = 100, random_state = 42
, max_depth = 22,n_jobs = -1)
rf_model.fit(xTrain, yTrain)
y_pred = rf_model.predict(xTest)
```

In [ ]:
```
# References: https://xgboost.readthedocs.io/en/latest/
import xgboost as xgb
xgb_model = xgb.XGBRegressor(
                 learning_rate=0.06,
                 max_depth=20,
                 colsample_bytree=0.4,
                 n_estimators=100,
                 subsample=0.6,
                 n_jobs = -1)
xgb_model.fit(xTrain,yTrain)
y_pred = xgb_model.predict(xTest)
```

In [ ]:
```
from sklearn.metrics import roc_auc_score
roc_auc_score(yTest, y_pred)
```

ROC score: 0.9498469076100613

In [120]:
```
test_trans= pd.read_csv("test_transaction.csv")
test_ident = pd.read_csv("test_identity.csv")

merged_test = pd.merge(test_trans, test_ident, on='TransactionID', how
='outer')
```

In [121]:
```
merged_test['day'] = pd.to_datetime(merged_test['TransactionDT'], unit
='s').dt.day
merged_test['hours'] = pd.to_datetime(merged_test['TransactionDT'], un
it='s').dt.hour
```

In [122]:
```
merged_test.drop(['id_23','id_27','id_30','id_31','id_33','id_34'], axis = 1, inplace=True)
group = merged_test.columns.to_series().groupby(merged_test.dtypes).groups
for k, v in group.items():
    if (k.name == 'int64' or k.name == 'float64'):
        for eachVal in v:
            merged_test[eachVal].fillna(np.nanmedian(merged_test[eachVal]), inplace=True)
```

In [123]:
```
merged_test["ProductCD"].fillna("T", inplace = True)
merged_test["card4"].fillna("U", inplace = True)
merged_test["card6"].fillna("V", inplace = True)
merged_test["P_emaildomain"].fillna("B", inplace = True)
merged_test["R_emaildomain"].fillna("X", inplace = True)
merged_test["DeviceInfo"].fillna("Y", inplace = True)
merged_test["DeviceType"].fillna("Z", inplace = True)
merged_test["M1"].fillna("A", inplace = True)
merged_test["M2"].fillna("A", inplace = True)
merged_test["M3"].fillna("A", inplace = True)
merged_test["M4"].fillna("A", inplace = True)
merged_test["M5"].fillna("A", inplace = True)
merged_test["M6"].fillna("A", inplace = True)
merged_test["M7"].fillna("A", inplace = True)
merged_test["M8"].fillna("A", inplace = True)
merged_test["M9"].fillna("A", inplace = True)


merged_test["id_12"].fillna("D", inplace = True)
merged_test["id_15"].fillna("E", inplace = True)
merged_test["id_16"].fillna("D", inplace = True)
merged_test["id_28"].fillna("I", inplace = True)
merged_test["id_29"].fillna("J", inplace = True)
merged_test["id_35"].fillna("A", inplace = True)
merged_test["id_36"].fillna("A", inplace = True)
merged_test["id_37"].fillna("A", inplace = True)
merged_test["id_38"].fillna("A", inplace = True)


from sklearn.preprocessing import LabelEncoder

label_encod = LabelEncoder()
for each_col in ['M1','M2','M3', 'M4', 'M5', 'M6', 'M7', 'M8', 'M9','P
roductCD','card4','card6','DeviceType',
                'id_12','id_15','id_16','id_28','id_29','id_35','id_36
','id_37','id_38']:
    each_col_unique = np.unique(merged_test[each_col])
    each_col_labels = label_encod.fit_transform(merged_test[each_col])
    merged_test[each_col] = each_col_labels
merged_test.head(5)
```

Out[123]:

	TransactionID	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	card4	car
0	3663549	18403224	31.95	4	10409	111.0	150.0	4	226
1	3663550	18403263	49.00	4	4272	111.0	150.0	4	226
2	3663551	18403310	171.00	4	4476	574.0	150.0	4	226
3	3663552	18403310	284.95	4	10989	360.0	150.0	4	166
4	3663553	18403317	67.95	4	18018	452.0	150.0	3	117

5 rows × 429 columns

In [124]:
```
from sklearn.feature_extraction import FeatureHasher

fh = FeatureHasher(n_features=6, input_type='string')
hashed_features = fh.fit_transform(merged_test['P_emaildomain'])
hashed_features = hashed_features.toarray()
merged_test = pd.concat([merged_test, pd.DataFrame(hashed_features,col
umns = ['ped1','ped2','ped3','ped4','ped5','ped6']
          )],axis=1)
#merged_test.head(5)

hashed_features = fh.fit_transform(merged_test['R_emaildomain'])
hashed_features = hashed_features.toarray()
merged_test = pd.concat([merged_test, pd.DataFrame(hashed_features,col
umns = ['red1','red2','red3','red4','red5','red6']
          )],axis=1)
#merged_required.head(5)

hashed_features = fh.fit_transform(merged_test['DeviceInfo'])
hashed_features = hashed_features.toarray()
merged_test = pd.concat([merged_test, pd.DataFrame(hashed_features,col
umns = ['devInfo1','devInfo2','devInfo3','devInfo4','devInfo5','devInf
o6']
          )],axis=1)

merged_test.head(5)
```

Out[124]:

	TransactionID	TransactionDT	TransactionAmt	ProductCD	card1	card2	card3	card4	car
0	3663549	18403224	31.95	4	10409	111.0	150.0	4	226
1	3663550	18403263	49.00	4	4272	111.0	150.0	4	226
2	3663551	18403310	171.00	4	4476	574.0	150.0	4	226
3	3663552	18403310	284.95	4	10989	360.0	150.0	4	166
4	3663553	18403317	67.95	4	18018	452.0	150.0	3	117

5 rows × 447 columns

In [125]:
```
merged_test.drop(['P_emaildomain','R_emaildomain','DeviceInfo','Transa
ctionDT'], axis = 1, inplace=True)
```

In [126]:
```
merged_test.head(5)
```

Out[126]:

	TransactionID	TransactionAmt	ProductCD	card1	card2	card3	card4	card5	card6	addr
0	3663549	31.95	4	10409	111.0	150.0	4	226.0	3	170.
1	3663550	49.00	4	4272	111.0	150.0	4	226.0	3	299.
2	3663551	171.00	4	4476	574.0	150.0	4	226.0	3	472.
3	3663552	284.95	4	10989	360.0	150.0	4	166.0	3	205.
4	3663553	67.95	4	18018	452.0	150.0	3	117.0	3	264.

5 rows × 443 columns

In [127]:
```
#Baseline model using Linear Regression
y_prediction_test = model.predict(merged_test)
```

In [ ]:
```
# Using Xgboost Regressor
y_prediction_test = xgb_model.predict(merged_test)
```

In [ ]:
```
y_predict_test_df = pd.DataFrame(y_prediction_test, columns=['isFraud'
])
y_predict_test_df.head(5)
```

In [ ]:
```
final_df= pd.DataFrame(data={'TransactionID':merged_test['TransactionI
D'],'isFraud':y_predict_test_df['isFraud']})
```

In [ ]:
```
final_df.to_csv('submissionxg1.csv', index=False)
```

Part 7 - Final Result

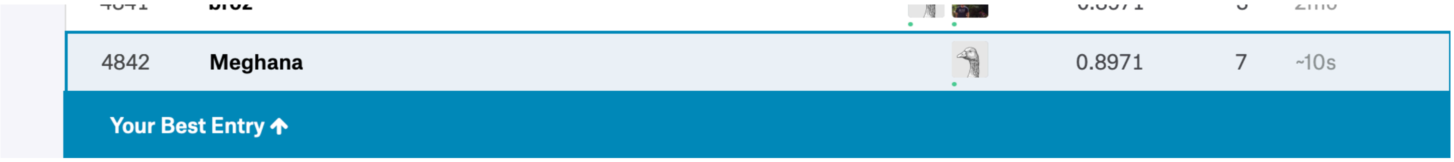
Report the rank, score, number of entries, for your highest rank. Include a snapshot of your best score on the leaderboard as confirmation. Be sure to provide a link to your Kaggle profile. Make sure to include a screenshot of your ranking. Make sure your profile includes your face and affiliation with SBU.

Kaggle Link: [https://www.kaggle.com/meghanavemulapalli (https://www.kaggle.com/meghanavemulapalli)](https://www.kaggle.com/meghanavemulapalli)

Highest Rank: 4842

Score: 0.8971

Number of entries: 7



In [ ]: