Fraud detection

June 1, 2020

1 Shopee Fraud Detection

This is problem presented by Shopee in 2019 during a competition https://www.kaggle.com/c/ungrd-rd2-auo/overview

Shopee provided 4 datasets, containing order information, device used by users, credit card used by users and bank account used by users. Our job is to find out the fake orders where buyer and seller are either directly or indirectly linked.

Objective: detect fake orders where buyer and seller are either directly or indirectly linked

1.0.1 Library

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import networkx as nx
```

1.0.2 Data

taken from: https://www.kaggle.com/c/ungrd-rd2-auo

[2]:
```
bank_account = pd.read_csv('data/bank_accounts.csv')
```

```
/Users/lingjie/opt/anaconda3/lib/python3.7/site-
packages/IPython/core/interactiveshell.py:3063: DtypeWarning: Columns (1) have
mixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

[3]:
```
#detect the mixed type
for account in bank_account['bank_account']:
    try:
        int(account)
    except:
        print(account)
```

```
029-992-19-99339-4
029-992-19-99339-4
029-992-19-99339-4
```

[4]:
```
#correct a unmatched entry in bank account
bank_account.loc[bank_account['bank_account'] == '029-992-19-99339-4',␣
↪'bank_account'] = '02999219993394'
bank_account['bank_account'] = bank_account['bank_account'].astype('int64')
```

[5]:
```
credit_card = pd.read_csv('data/credit_cards.csv')
```

[6]:
```
device_info = pd.read_csv('data/devices.csv')
```

[7]:
```
orders_record = pd.read_csv('data/orders.csv')
```

1.0.3 Checking if there's sign of frauds

[8]:
```
#bank account
bank_account['userid'].unique().shape, bank_account['bank_account'].unique().
↪shape
#since we have more unique bank account than user id, some users use the same␣
↪bank account
```

[8]:
```
((255495,), (328371,))
```

[9]:
```
#credit card
credit_card['userid'].unique().shape, credit_card['credit_card'].unique().shape
#since user id > credit card, some users use same credit card
```

[9]:
```
((22099,), (37367,))
```

[10]:
```
#device info
device_info['userid'].unique().shape, device_info['device'].unique().shape
#since device > userid, some users log in from multiple places
```

[10]:
```
((481519,), (1363287,))
```

1.0.4 Objective: Fraud detection

We want to detect fraud transactions, meaning: 1. buyer and seller shares either same bank account AND/OR credit card AND/OR device info 2. buyer and seller are indirectly linked through a third person or more

We can solve both cases using a network

1.0.5 Networkx

We will be building networks to investigate the relationships

[11]:
```
#create bank account network
bank_account_G = nx.from_pandas_edgelist(bank_account, 'userid', 'bank_account')
nx.set_node_attributes(bank_account_G, ['bank_account'], 'bank_account')
```

[12]:
```
#create credit card network
credit_card_G = nx.from_pandas_edgelist(credit_card, 'userid', 'credit_card')
nx.set_node_attributes(credit_card_G, ['credit_card'], 'credit_card')
```

[13]:
```
#create device info network
device_info_G = nx.from_pandas_edgelist(device_info,'userid','device')
nx.set_node_attributes(device_info_G, ['device_info'], 'device_info')
```

[14]:
```
#overall network
shopee_G = nx.compose(bank_account_G,
                      nx.compose(credit_card_G, device_info_G))
```

[15]:
```
#delete other networks to save space
del bank_account_G
del credit_card_G
del device_info_G
```

Now that we have the network ready, we can use the network to investigate the relationship between different buyer and sellers. For example, this buyer and seller share the same bank account

[16]:
```
nx.shortest_path(shopee_G, 221232712, 66353306)
```

[16]:
```
[221232712, 8300298809, 66353306]
```

[17]:
```
shopee_G.nodes[8300298809]
```

[17]:
```
{'bank_account': ['bank_account']}
```

Since there is more unique users than connected components, we can estimate the total number of fruadsin this dataset This is an estimation because different bank account/ credit card/ device info all serve as nodes

[18]:
```
num_components = nx.number_connected_components(shopee_G)
num_components
```

[18]:
```
447703
```

[19]:
```
total_components = (orders_record['buyer_userid'] +␣
 ↪orders_record['seller_userid']).unique().size
total_components
```

[19]:
```
574987
```

[20]:
```
total_components - num_components
```

[20]: 127284

1.0.6 Find out the frauds!

[21]:
```
def check_frauds(buyer_id, seller_id):
    try:
        path = nx.shortest_path(shopee_G, buyer_id, seller_id)
        return 1, path
    except:
        return 0, None
```

[22]:
```
#let's try with a small sample for testing first
orders_record.head(5).apply(lambda row: check_frauds(row[1],row[2]), axis=1)
```

[22]:
```
0    (0, None)
1    (0, None)
2    (0, None)
3    (0, None)
4    (0, None)
dtype: object
```

[23]:
```
#now let's apply to the whole dataset
orders_record['is_fraud'] = orders_record.apply(lambda row:␣
 ↪check_frauds(row[1],row[2]), axis=1)
```

[25]:
```
orders_record['fraud_method'] = orders_record['is_fraud'].apply(lambda x: x[1])
orders_record['is_fraud'] = orders_record['is_fraud'].apply(lambda x: x[0])
```

[26]:
```
#now we have the fraud orders ready
orders_record.loc[orders_record['is_fraud'] == 1,:].head()
```

[26]:

	orderid	buyer_userid	seller_userid	is_fraud	fraud_method
1649	1954198318	221232712	66353306	1	[221232712, 8300298809, 66353306]
2679	1955598428	35545436	70763052	1	[35545436, /3TLpeou8xXsNxpACFFKr34Kqqwxiu5Hi1k…
3545	1954515646	32834366	188151804	1	[32834366, 1KNEOFRIZaFcFx5+S+bOxyWuWBbITxnfoM7…
5938	1953728724	168491444	158559422	1	[168491444, yf7AHm3O97XAQwQuSmyoaxcaFSSAZcVCxm…
8393	1955955178	235599454	51098362	1	[235599454, 9120282009, 51098362]

1.0.7 Investigate the frauds

[27]:
```
fraud_orders = orders_record.loc[orders_record['is_fraud'] == 1,:]
```

[1]:
```
# for entry in fraud_orders.iloc[:,-1].values:
#     print('buyer: {}, seller: {}, connection: {}'.
 ↪format(entry[0],entry[-1],entry[1:-1]))
```

[29]:
```
shopee_G.nodes[17318002]
```

[29]:
```
{'bank_account': ['bank_account'],
 'credit_card': ['credit_card'],
 'device_info': ['device_info']}
```

we see that there are many interesting ways people attempt frauds:

the simplest way is creating two account but share the same bank account/ credit card/ device

the more complex way is (for example between buyer: 234217326, seller: 39287026) where multiple bank account, credit card and devices were used

1.0.8 For submission

[30]:
```
submission = orders_record.loc[:,['orderid','is_fraud']]
```

[31]:
```
submission.to_csv('submission.csv', index=False)
```

Our result has achieved perfect score

sadly the Leaderboard is closed and our result is not reflected there