Pre-processing: Features: Data description

Things to Do:

1. Meaning of features

2. Summary of needs to be done for each column to extract information

3. Datetime feature: Breakdown into day column, month column and year column

i.e, go through every single comments on the discussions on the explanatory variables in the dataset on Kaggle.

isFraud = Response variable we wish to predict from other explanatory variables

Source:

1. https://www.kaggle.com/c/ieee-fraud-detection/discussion/101203

2. https://www.kaggle.com/c/ieee-fraud-detection/discussion/101203#583227

- The data spans 6 months
  - So the maximum DT value is 15811131
- Also, seems like the training dataset and test data set have different number of rows
  - for Python(pandas), use -> result = pd.concat([data_identity, data_transactions], axis=1, join='inner')
  - We can merge the two files via this, if needs be.
  - Half-year data as training and the other half as testing.
    - However, fraud pattern might follows yearly pattern a lot.
- Furthermore, if you merge the train and test data sets and plot the data over transactionDT you can see that the data is split in time. I.e. that the test data is a continuation from the train data.
  - Test dataset contains payment transactions posterior to Train dataset. Public/private set was split by random though.
- Also these are all online transactions, so store address is not included!

Extensive discussion of the input variables that might be of use on Kaggle

1. TransactionDT: timedelta from a given reference datetime (not an actual timestamp)

- "TransactionDT first value is 86400, which corresponds to the number of seconds in a day (60 * 60 * 24 = 86400) so I think the unit is seconds. Using this, we know the data

spans 6 months, as the maximum value is 15811131, which would correspond to day 183."

2. TransactionAMT: transaction payment amount in USD

- There seems to be a link to three decimal places and a blank addr1 and addr2 field. Is it possible that these are foreign transactions and that, for example, the 75.887 in row 12 is the result of multiplying a foreign currency amount by an exchange rate?"
- So perhaps create a number of decimal places column
  - https://www.kaggle.com/code/yasagure/places-after-the-decimal-point-tell-us-a-lot/notebook --> Tells you how to create those columns in Python!
  - Decimals points are important so do not convert it to string, etc.

3. ProductCD: product code, the product for each transaction

- This can be both product or a service.

4. card1 - card6: payment card information, such as card type, card category, issue bank, country, etc.

5. addr: address

"both addresses are for purchaser

addr1 as billing region

addr2 as billing country"

5. dist: distance

"distances between (not limited) billing address, mailing address, zip code, IP address, phone area, etc."

6. P_ and (R__) emaildomain: purchaser and recipient email domain

- Certain transactions don't need recipient, so R_emaildomain is null.

7. C1-C14: counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.

- For example, counts of phone numbers, email addresses, names associated with the user & device, ip_addr, billing_addr, etc. Also these are for both purchaser and recipient, which doubles the number.

8. D1-D15: timedelta, such as days between previous transaction, etc.

- D1/D2 are about "how many days have passed from the first transaction", D3 is responsible for "how many days have passed from the previous transaction"

- Unit: days

- D1 - Difference between latest & most recent transaction in history (MR1)

- D2 - Difference between Most recent (MR1) & immediate previous transaction in history (MR2)

9. M1-M9: match, such as names on card and address, etc.

10. Vxxx: Vesta engineered rich features, including ranking, counting, and other entity relations.

- For example, how many times the payment card associated with a IP and email or address appeared in 24 hours time range, etc.

- All Vesta features were derived as numerical. some of them are count of orders within a clustering, a time-period or condition, so the value is finite and has ordering (or ranking). I wouldn't recommend to treat any of them as categorical. If any of them resulted in binary by chance, it maybe worth trying.

Use the code snippet to have a quick overview for all the features in the dataset.

```
for col, values in df_train.iteritems():
    num_uniques = values.nunique()
    print ('{name}: {num_unique}'.format(name=col,
num_unique=num_uniques))
    print (values.unique())
    print ('\n')
```

& top N values in each column

```
def df_column_unique_values(df, top_n = 5):
    for col_name, values in df.iteritems():
        col_value_counts = values.value_counts()
        print(f"{col_name} : {len(col_value_counts)}")
        col_value_count_list = [
            "'" + str(c) + "'" + ":" + str(n) for c, n in
sorted(
                col_value_counts.items(),
                key=lambda kv: kv[1],
                reverse=True
            )
        ]
        print(",
".join(col_value_count_list[:min(len(col_value_count_list),
top_n)]))
        # print ('\\n')
```

Card 1 to 3 and 5 = bankname, cardname, etc.

Mx is attribute of matching check, e.g. is phone areacode matched with billing zipcode, purchaser and recipient first/or last name match, etc.

Identity CSV file

- id_14
    - Clock or timezone related feature, I don't know, but kind of looks numerical
    - Since it's timezone, it can also be modelled as categorical
    - For example, It's interesting that it is always +/- 30 which means they are not running biz in ANZ.
- id-01 to id-20 are collected from Vesta, id-21 to id-30 are from sourceA and the rest are from source
- Do not make -100 values as NaN!
- id01 to id11 are numerical features for identity, such as device rating, ip_domain rating, proxy rating, etc. Also it recorded behavioral fingerprint like account login times/failed to login times, how long an account stayed on the page, etc.

In the dataset, not all cards or users are unique

- So some transactions are from the same card or from the same account.

Cxx and Dxx are features you can't generate from any other column in this dataset; they're generated from entity we couldn't provide (full card number, full name/address, email address, etc).

I wouldn't recommend removing them unless you can prove they're useless.

Categorical Features:

DeviceType

DeviceInfo

id_12 - id_38

Categorical Features:

ProductCD

card1 - card6

addr1, addr2

P_emaildomain

R_emaildomain

M1 - M9

Identity Table

Variables in this table are identity information – network connection information (IP, ISP, Proxy, etc) and digital signature (UA/browser/os/version, etc) associated with transactions.

They're collected by Vesta's fraud protection system and digital security partners.

(The field names are masked and pairwise dictionary will not be provided for privacy protection and contract agreement)