CID: C2600591

Recruiter: Lindsey Boggess

Version: v20.01

Please read the instruction before running the code

First of all, thank you for providing the chance to take the challenge! This file can guild you how to run the code and the meaning of each file.

File name

conclusion.pdf : Conclusions of each questions without code.

transaction.ipynb : Source code of Data Science Challenge, please open it with jupyter notebook.

transaction.py : In case the ipynb cannot be opened, please run this file. To provide a better visualization, it is highly recommend to open transaction.ipynb instead of transaction.py.

transaction.pdf: PDF version of transaction.ipynb (with code run).

rfc.model : extracted Random Forest model.

Version of Python and packages

The following is the version of Python and packages. In order to run the code successfully, the version of Python and each packages should be equal or higher than the record below:

Python: 3.7.3

pandas: 0.24.2

numpy: 1.16.2

matplotlib: 3.0.3

interval: 1.0.0

scikit-learn: 0.22

How to run the ipynb file

First of all, please make sure all the packages are installed, and have the data ready. If you don't want to run the cell one by one, you can just use Cell->Run All to run all cells automatically. Then you will get the result of each cells and codes.

About rfc.model

If you want to run the model directly instead of training the model again from the beginning, you can use the following code to read the model.

```
import joblib
from sklearn.preprocessing import LabelEncoder
rfc = joblib.load('rfc.model')
```

Before running the model, please make sure the input data is in dataframe type(cdv file) and meet the requirement of the model.

You can run the code below to formalize the input data.

```
input_data=pd.read_csv('transactions.csv',encoding='utf-8',index_col=[0])
input_data['isCVVmatch']=input_data['cardCVV']==input_data['enteredCVV']
input_data=input_data[['isCVVmatch','cardPresent','merchantCountryCode','posConditionCode',
                'transactionAmount','posEntryMode','isFraud']]

input_data.dropna(inplace=True)
input_predict_x=input_data[['isCVVmatch','cardPresent','merchantCountryCode','posConditionCode',
                'transactionAmount','posEntryMode']]
input_predict_y=input_data['isFraud']
```

```
for feature in ['isCVVmatch','cardPresent','merchantCountryCode']:
    enc = LabelEncoder() #Classify non-numeric quantities to corresponding integers
    enc.fit(input_predict_x[feature])
    input_predict_x[feature] = enc.transform(input_predict_x[feature])

enc = LabelEncoder() #Classify non-numeric quantities to corresponding integers
enc.fit(input_predict_y)
input_predict_y = enc.transform(input_predict_y)
```

After cleaning the data, you can use the following code to make prediction.

```
predict=rfc.predict(input_predict_x)
```

If you don't have the csv file of dataset, then this code can help!

```
with open('transactions.txt') as f:
    data = json.loads("[" +
                      f.read().replace("}\n{", "},\n{") + "]")
input_data=pd.DataFrame(data)
input_data.to_csv('transactions.csv',encoding='utf-8')
```

Thank you for reading this document!