

ML Engineering Code Challenge

This code challenge is intended to test your skills in Machine Learning Engineering. We are looking to evaluate your knowledge in building machine learning models as well as the ability to deploy these models in production.

You are provided a synthetic dataset for a mobile payments application. In this dataset, you are provided the sender and recipient of a transaction as well as whether transactions are tagged as fraud or not fraud. Your task is to build a fraud detection API that can be called to predict whether or not a transaction is fraudulent.

You can download the dataset [here](#)

You are expected to build a REST API that predicts whether a given transaction is fraudulent or not. You are also to assume that the previous API calls are to be stored in order to engineer features relevant to finding fraud. The API calls will include the time step of the transaction, so you can assume that a transaction happens sequentially within the same time step.

For example, if I make the following transactions in the same time step:

nameOrig	nameDest	amount	type
anon	fake123	50,000.0	TRANSFER
anon	fake123	50,000.0	TRANSFER
anon	fake123	50,000.0	TRANSFER
anon	fake123	50,000.0	TRANSFER
anon	fake123	50,000.0	TRANSFER

The first transaction is unlikely to be fraudulent, since anon is initiating a normal transfer. However, multiple successive transfers of the same amount in the same hour is potentially fraudulent, since anon's account might have been taken over by a fraudster. On the first API call, your model is unlikely to classify the transaction as fraudulent. However, on the fifth call, it's likely that it will be tagged as fraudulent.

The REST API only has 1 endpoint **/is-fraud** that takes in a **POST** request:

The body is expected to receive the following fields(which are also the fields that can be found in your dataset):

Name	Type	Description
step	Integer	Maps a unit of time in the real world. In this case, 1 step is 1 hour of time.
type	String	CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER
amount	Decimal	amount of the transaction in local currency
nameOrig	String	customer who started the transaction
oldbalanceOrig	Decimal	initial balance before the transaction
newbalanceOrig	Decimal	customer's balance after the transaction.
nameDest	String	recipient ID of the transaction.
oldbalanceDest	Decimal	initial recipient balance before the transaction.
newbalanceDest	Decimal	recipient's balance after the transaction.

The following is a sample body when making a POST request to your

```
{
  "step":1,
  "type":"PAYMENT",
  "amount":9839.64,
  "nameOrig":"C1231006815",
  "oldbalanceOrig":170136.0,
  "newbalanceOrig":160296.36,
  "nameDest":"M1979787155",
  "oldbalanceDest":0.0,
  "newbalanceDest":0.0
}
```

Your API is expected to return a JSON object with a boolean field **isFraud**. You may find a sample response below:

```
{
  "isFraud": true
}
```

To help you in testing out your API when we evaluate it, this is a sample script in Python making an API call:

```
import requests
r = requests.post('https://example.com/is-fraud', json = {
    "step":1,
    "type":"PAYMENT",
    "amount":9839.64,
    "nameOrig":"C1231006815",
    "oldbalanceOrig":170136.0,
    "newbalanceOrig":160296.36,
    "nameDest":"M1979787155",
    "oldbalanceDest":0.0,
    "newbalanceDest":0.0
})
```

For your submission, we are expecting the following:

1. Deployed REST API

- a. As mentioned above, we would need an API that takes in a POST request for the /is-fraud url and returns a prediction on whether or not a transaction is fraudulent.
- b. Your REST API should be public for us to call the API and evaluate the accuracy of your model
- c. Given the nature of the data, your REST API will likely need to take into account previous transactions, so make sure it is able to take note of transactions from your training dataset as well as previous API calls.
- d. Please don't forget to share the URL of your deployed REST API once you're done

2. Model

- a. We are expecting a machine learning model that can correctly classify whether or not a transaction is fraudulent.

3. Copy of relevant files

- a. We would need to see all the code you used in to come up with the final output including but not limited to: data processing, feature engineering, model development, API source code.
- b. It is preferable if you make use of a git repository like Github to host your code.
- c. We are also expecting you to do a quick presentation of your model API and any interesting findings. Slides are completely optional, but please share your slides prior to the presentation if you plan to make use of slides.

Please submit your output via email to kenley+mleng@plentina.com

Good luck and we look forward to seeing your work.