ANSWERS TO HOMEWORK SESSION 5 -ML ZOOMCAMP

Question 1

Install Pipenv

What's the version of pipenv you installed?

Use --version to find out

```
PS C:\Users\dbazoneny\Desktop\virtualproject> pipenv --version
pipenv, version 2021.5.29
PS C:\Users\dbazoneny\Desktop\virtualproject>
```

Question 2

- Use Pipenv to install Scikit-Learn version 1.0
- What's the first hash for scikit-learn you get in Pipfile.lock?

```
},
"scikit-learn": {
    "hashes": [
        "sha256:038f4e9d6ef10e1f3fe82addc3a14735c299866eb10f2c77c090410904828312
```

Ans ; "sha256:038f4e9d6ef10e1f3fe82addc3a14735c299866eb10f2c77c090410904828312",

Models

We've prepared a dictionary vectorizer and a model.

They were trained (roughly) using this code:

```
features = ['tenure', 'monthlycharges', 'contract']
dicts = df[features].to_dict(orient='records')
```

```
dv = DictVectorizer(sparse=False)
X = dv.fit_transform(dicts)

model = LogisticRegression().fit(X, y)
```

And then saved with Pickle. Load them:

- DictVectorizer
- LogisticRegression

Question 3

Let's use these models!

- Write a script for loading these models
- Score this customer:

```
{"contract": "two_year", "tenure": 12, "monthlycharges": 19.7}
```

What's the probability that this customer is churning?

Ans ...Probability that the given Customer is churning is [0.11549581]

Question 4

Now let's serve this model as a web service

- Install Flask and Gunicorn (or waitress, if you're on Windows)
- Write Flask code for serving the model
- Now score this customer using `requests`:

```
url = "YOUR_URL"
customer = {"contract": "two_year", "tenure": 1, "monthlycharges": 10}
requests.post(url, json=customer).json()
```

What's the probability that this customer is churning?

Ans .. {'Churn': True, 'Churn Probability': 0.9988892771007961}

Docker

For this and the following quesition you'll need Docker. Install it. We will use it for the next two questions

For these questions, I prepared a base image: agrigorev/zoomcamp-model:3.8.12-slim. You'll need to use it (see Question 5 for an example).

This is what it already has:

FROM python:3.8.12-slim

WORKDIR /app

COPY ["model2.bin", "dv.bin", "./"]

I already built it and then pushed it to agrigorev/zoomcamp-model:3.8.12-slim

Question 5

Create your own Dockerfile based on this one

It should start like that:

FROM agrigorev/zoomcamp-model:3.8.12-slim

add your stuff here

Now complete it:

Install all the dependencies form the Pipenv file

Copy your Flask script

Run it with gunicorn

When you build your image, what's the digest for agrigorev/zoomcamp-model:3.8.12-slim?

Look at the first step of your build log. It should look something like that:

Step 1/3 : FROM python:3.8.12-slim

---> 2e56f6b0af69

Ans

```
PS C:\Users\dbazoneny\Downloads\ML_zoomcamp_session5\question_5> docker build .
[+] Building 6.4s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 167B                                                                                0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/agrigorev/zoomcamp-model:3.8.12-slim                                     0.7s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 447B                                                                                   0.0s
 => CACHED [1/3] FROM docker.io/agrigorev/zoomcamp-model:3.8.12-slim@sha256:1ee036b365452f8a1da0dbc3bf5e7dd0557cfd33f0e56b280  0.0s
 => [2/3] COPY . /app                                                                                               0.1s
 => [3/3] RUN pip install -r requirements.txt                                                                       5.1s
 => exporting to image                                                                                              0.3s
 => => exporting layers                                                                                             0.3s
 => => writing image sha256:caca4604624761e6bddb44ad70e7ac314c95f040b490934254583a21fb627474                        0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

CACHED [1/3] FROM docker.io/agrigorev/zoomcamp-model:3.8.12-slim@sha256:1ee036b365452f8a1da0dbc3bf5e7dd0557cfd33f0e56b280

Question 6

Let's run your docker container!

After running it, score the same customer:

url = "YOUR_URL"

customer = {"contract": "two_year", "tenure": 12, "monthlycharges": 10}

requests.post(url, json=customer).json()

Ans {'Churn': False, 'Churn Probability': 0.32940789808151005}

In [23]:
```
import requests
```

In [45]:
```
url = "http://localhost:9696/predict"
```

In [46]:
```
customer = {"contract": "two_year", "tenure": 12, "monthlycharges": 10}
```

In [47]:
```
requests.post(url, json=customer).json()
```

Out[47]:
```
{'Churn': False, 'Churn Probability': 0.32940789808151005}
```