Lab 5: Creating a Model Service

In this lab, we'll use the churn prediction model trained on a smaller set of features.

Question 1

Install Pipenv
What's the version of pipenv you installed?
Use --version to find out

Question 2

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

Note: You don't need to train the model. This code is just for your reference.

And then saved with Pickle. Download them:

DictVectorizer
LogisticRegression
With wget:

```
PREFIX=https://raw.githubusercontent.com/fenago/mlbookcamp-code/master/course-zoomcamp/05-deployment/lab
wget $PREFIX/model1.bin
wget $PREFIX/dv.bin
```

Question 3

Let's use these models!

Write a script for loading these models with pickle
Score this customer:
{"contract": "two_year", "tenure": 12, "monthlycharges": 19.7}
What's the probability that this customer is churning?

If you're getting errors when unpickling the files, check their checksum:

$ md5sum model1.bin dv.bin
5868e129bfbb309ba60bf750263afab1 model1.bin
c49b69f8a5a3c560882ff5daa3c0ff4d dv.bin

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?

Docker
Install Docker. 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 image is based on python:3.8.12-slim and has a logistic regression model (a different one) as well a dictionary vectorizer inside.

This is how the Dockerfile for this image looks like:

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.

Note: You don't need to build this docker image, it's just for your reference.

Question 5

Now create your own Dockerfile based on the image I prepared.

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 image id for agrigorev/zoomcamp-model:3.8.12-slim?

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

$ docker some-command-for-building
Sending build context to Docker daemon 2.048kB
Step 1/N : FROM agrigorev/zoomcamp-model:3.8.12-slim
 ---> XXXXXXXXXXXX
Step 2/N : ....
You need this XXXXXXXXXXXX.

Alternatively, you can get this information when running docker images - it'll be in the "IMAGE ID" column. Submitting DIGEST (long string starting with "sha256") is also fine.

Question 6

Let's run your docker container!

After running it, score this customer:

url = "YOUR_URL"
customer = {"contract": "two_year", "tenure": 12, "monthlycharges": 10}
requests.post(url, json=customer).json()
What's the probability that this customer is churning?