Flask deployment report (Week 4)

Diego Martínez Echevarría

January 14, 2022

Abstract

For Week 4's assignment, I have deployed a toy model with Flask, as a web app. This document describes the process step by step with a technical, detailed view.

1   Environment setup

To facilitate replicability, the first thing to do is to setup an environment, so that others can also run the app in the intended conditions. To do so, we run the following shell commands:

```
$ python -m venv environment
$ source environment/bin/activate
$ pip install flask
$ pip freeze > requirements.txt
```

Command by command, this has the following effects:

1. A virtual environment of the name "environment" is created, with a the corresponding "environment" folder.

2. The bash script "activate" is executed. This script sets the newly created environment as the current one. Since it was just created, this environment is empty and hasn't got any modules installed.

3. We install Flask with pip.

4. We save the status of the environment (right now Flask and its dependencies) into "requirements.txt". This file can then be used by other user with `pip install requirements.txt` to replicate our setup.

1

Once we've executed those commands, the requirement file looks something like this:

```
click==8.0.3
Flask==2.0.2
itsdangerous==2.0.1
Jinja2==3.0.3
MarkupSafe==2.0.1
Werkzeug==2.0.2
```

While coding this assignment other python modules have been used, so in the end this was how our requirement list looked:

```
UPDATE REQUIREMENTS:TXT
```

2 The Dataset

We have selected a relatively simple dataset from Kaggle about crabs. It contains fields describing the physiology of a crab, using fields such as Sex, Length, Diameter, etc.

Since the focus of this assignment we have only kept the most basic features, ignoring those like "Viscera Weight" to keep the web app on the cleaner side.

The dataset has already been curated, so the main focus is on selecting the right attributes and filtering the sex as "Male" or "Female", just to not confuse users as what "I" is.

```
import pandas as pd

df = pd.read_csv('crab_age.csv').dropna()
df = df[df["Sex"].isin(("M", "F"))]

# Process csv
fields = ["Sex", "Length", "Diameter", "Height", "Weight"]
X = pd.get_dummies(data=df[fields])
y = df["Age"]
```

To fit the model, the whole dataset has been divided in predictors (X) and predictions (y), since for this toy model we are not considering training and testing batches.

3 The Model

The "Model" is a Linear Regression, simple as it is. This is the code used to define and train it (X and y are defined in the previous section):

```python
import pickle
from sklearn.linear_model import LinearRegression

MODEL_PATH = "model.pkl"

# Build model, fit model
model = LinearRegression()
model.fit(X, y)

# Pickle the object
with open(MODEL_PATH,'wb') as savefile:
    pickle.dump(model, savefile)
```

As it is stated, we import the model, fit it with the dataset (see previous section) and we save it in a file.