Flask deployment report (Week 4)

Diego Martínez Echevarría

January 13, 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.

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 video game sales. It contains fields such as Company, Platform, Year (of release), Genre, etc. It has 16K+ entries and no missing values.