DEPLOYMENT ON FLASK

Name: Jose Vicente Solorzano

Batch code: LISUM04

Submitted date: 21/10/21

Submitted to: Data Glacier

Introduction

The main goal of this assignment is to create a simple machine learning model and its deployment on Flask to make predictions in http interface.

Their document is going to be organized into two sections:

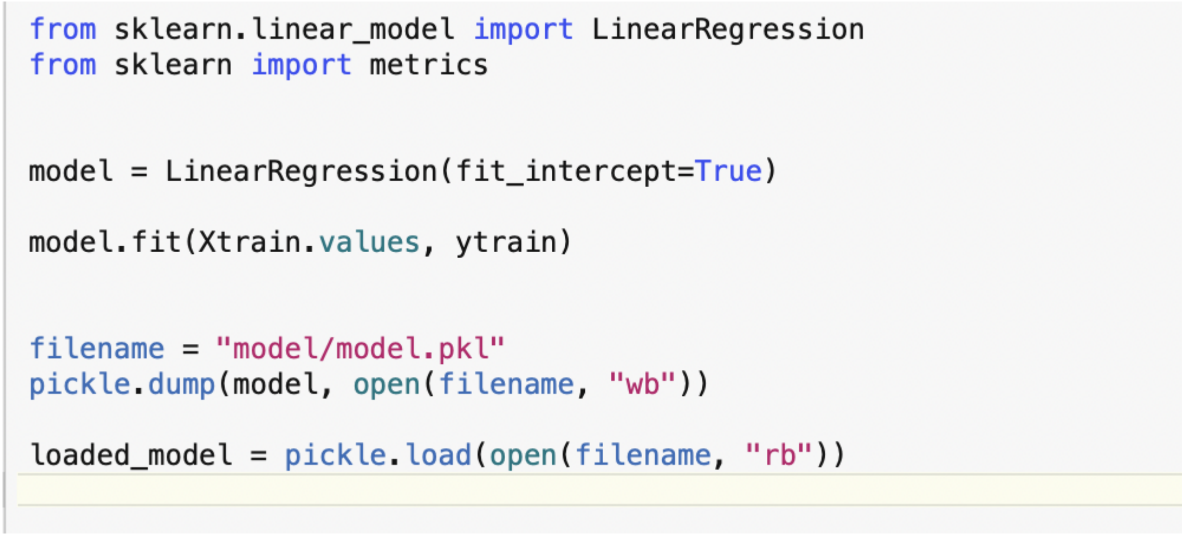
- Model
- Documentation, execution and deployment

Model

The model chose was LinearRegression. The idea was to forecast sales of a company, based on its investment on media (TV, radio and newspaper).

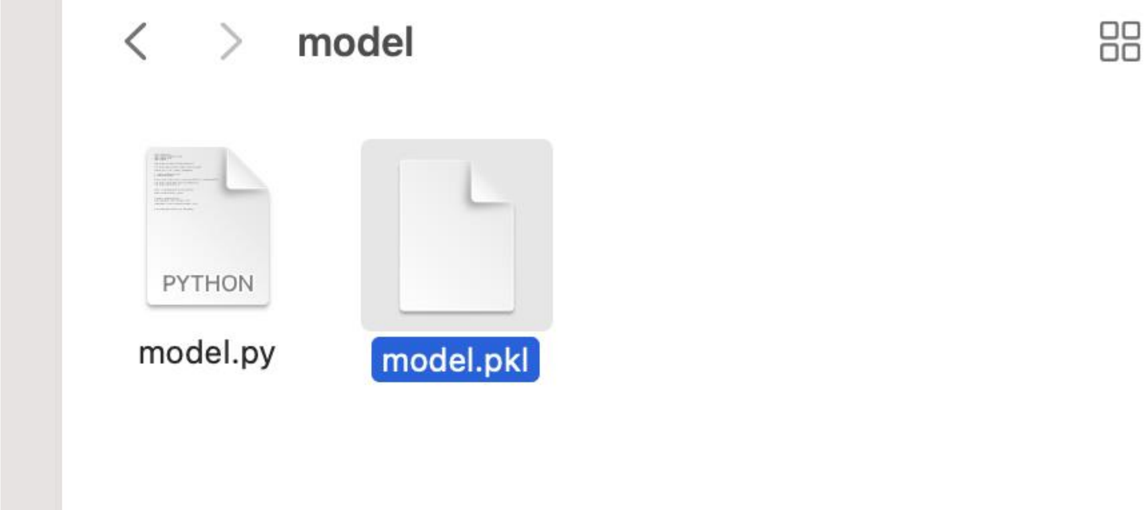
The R2 of the model is 0.9, so it is expected to give us accurate results.

Once we were ready with the model, it was time to serialize it and save it in our repository.



```
from sklearn.linear_model import LinearRegression
from sklearn import metrics


model = LinearRegression(fit_intercept=True)

model.fit(Xtrain.values, ytrain)


filename = "model/model.pkl"
pickle.dump(model, open(filename, "wb"))

loaded_model = pickle.load(open(filename, "rb"))
```

“model.pkl” (pickle extension) was saved on our repository “model/model.pkl”.



Documentation, Execution and Deployment

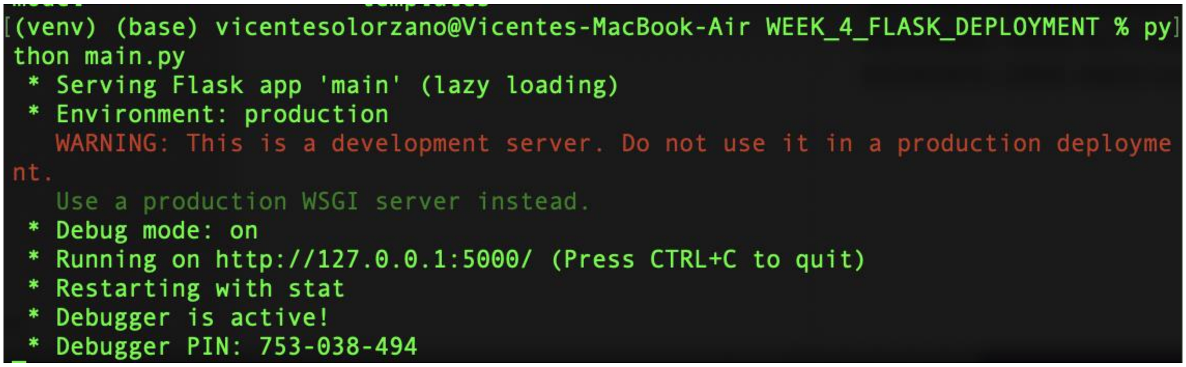
In our repository we will find some files:

- "main.py" script with flask constructor to execute the process.
- "model" file with the machine learning model document ("model.py") and its Pickle extension ("model.pkl").
- "static" with css files
- "templates" with html files

Aditionally, there are:

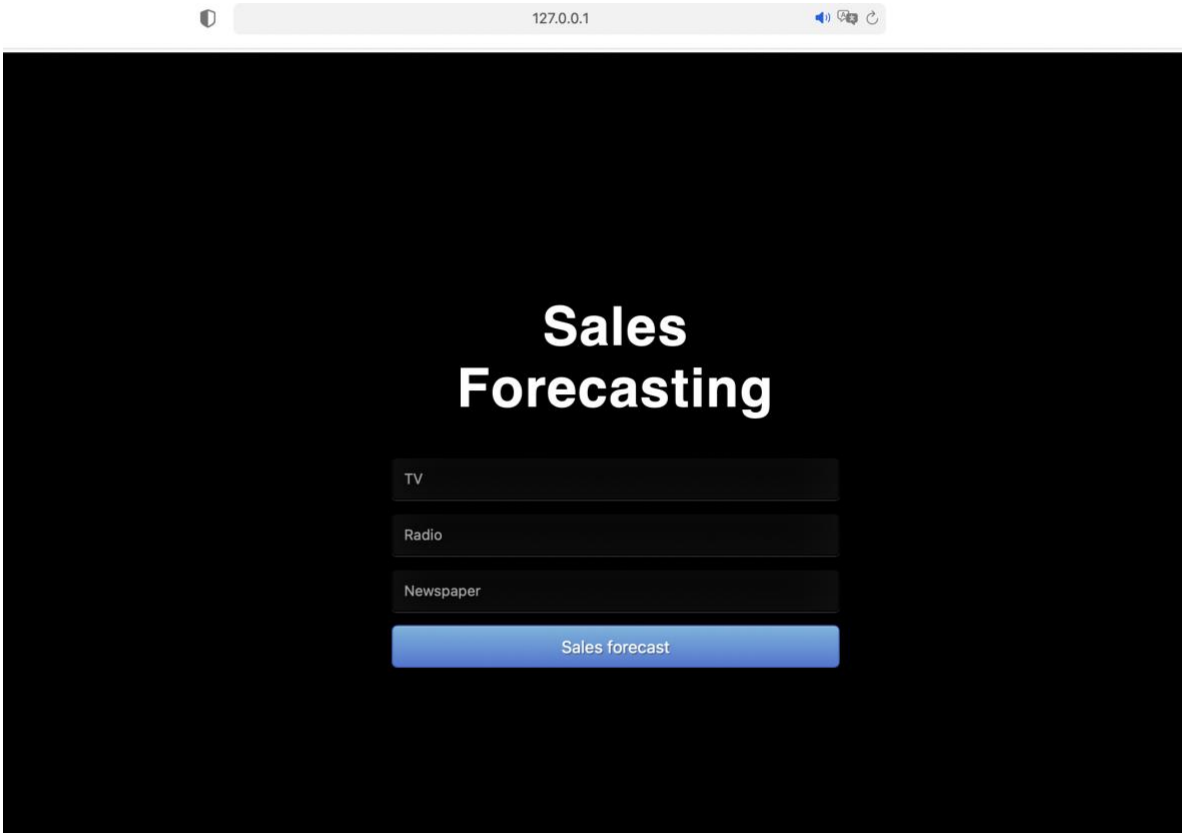
- "requirements.txt" to install libraries needed.
- "README.txt", a brief description of what we are doing.
- "venv" with the documentation of the virtual environment created.

Now, I am going to show the execution and deployment of the model on Flask. We execute our "main.py", script with flask constructor.

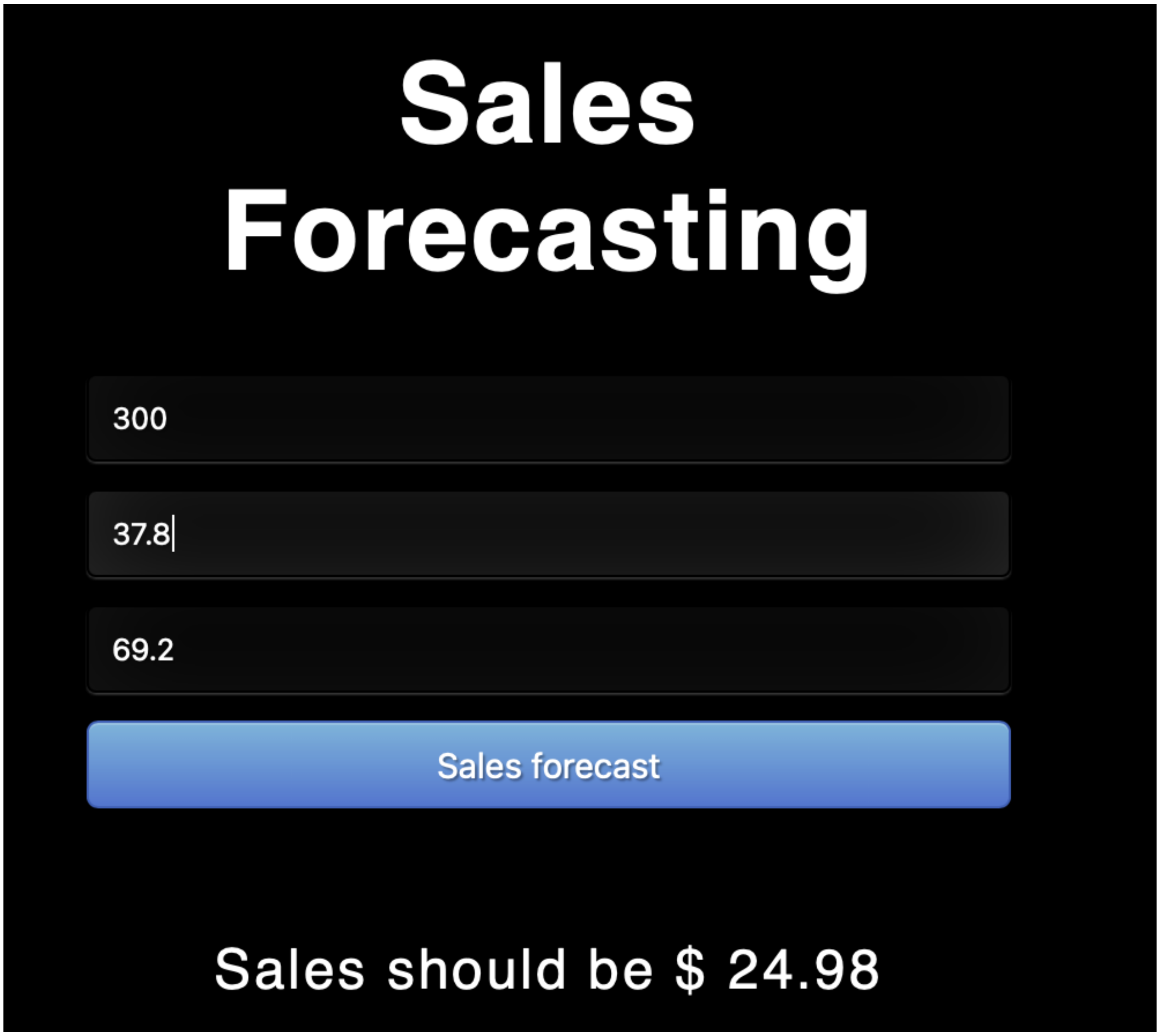


```
[(venv) (base) vicentesolorzano@Vicentes-MacBook-Air WEEK_4_FLASK_DEPLOYMENT % py]
thon main.py
 * Serving Flask app 'main' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 753-038-494
```

The link given by the terminal was http://127.0.0.1:5000/. The web interface looks like this:



Model deployment:



Sales Forecasting

300

37.8

69.2

Sales should be $ 24.98