Model Deployment using Flask

Name: Aishwarya

Batch code: LISUM27

Submission date:
11272023

Submitted to: Data Glacier

Overview

- Deploying basic machine learning model
- Learn how to use Flask to deploy a machine learning model into production

Abstract

This project has been written for the beginners of model deployment. With a simple linear regression example, a model was created on Spyder using Flask.

Table of Contents:

- What is model deployment?
- What is Flask?
- Installing Flask on your Machine
- Setting up the Project WorkFlow
- Build Machine Learning Model
- Spyder usage
- Save the Model
- Connect the Webpage with the Model
- Working of the Deployed Model

What is Model Deployment?

The process of integrating a machine learning model into an already-existing production environment in order to use data to inform actionable business choices is known as deployment. We make the model we have developed into a product in this way. We also make the product available to the user side simultaneously.

What is Flask?

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies, and several common framework-related tools.

Installing Flask on your Machine

Installing Flask is simple and straightforward. I generally use pip installed.

```
# If you are using pip
$ pip install flask

# For Linux
$ sudo apt-get install python3-flask
```

```
# Living on the edge
$ pip install -U https://github.com/pallets/flask/archive/master.tar.gz
```

Setting up the Project WorkFlow

- Model Building
- Save the model and setup app
- Webpage Template
- Predict class and send results

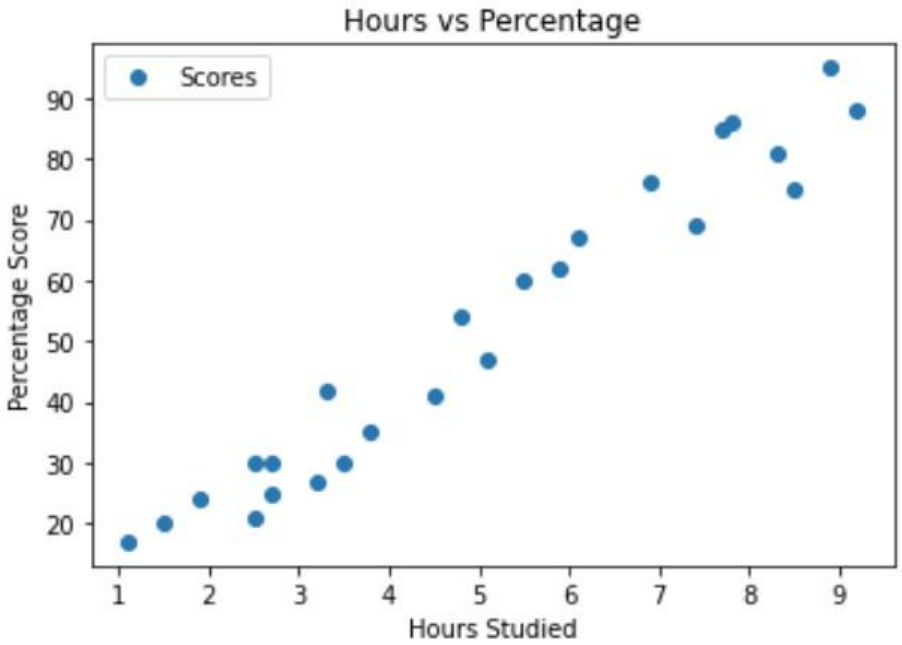
Build Machine Learning Model

In [3]:
```
scores.head()
```

Out[3]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [4]:
```
scores.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



```
In [6]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

In [7]: from sklearn.linear_model import LinearRegression
        regressor = LinearRegression()
        regressor.fit(X_train, y_train)

Out[7]: LinearRegression()
```

To retrieve the intercept and For retrieving the slope (coefficient of x):

```
In [8]: print(regressor.intercept_)

2.018160041434683

In [9]: print(regressor.coef_)

[9.91065648]
```

Making Predictions: Now that we have trained our algorithm, it's time to make some predictions.

```
In [10]: y_pred = regressor.predict(X_test)

In [11]: df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
         df
```

Out[11]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

Testing and Proof

```
In [13]: my_score = 5

In [14]: y_array = np.asarray(my_score)

In [15]: regressor.predict(y_array.reshape(-1,1))
Out[15]: array([51.57144244])

In [16]: (5 * 9.91065648) + 2.018160041434683
Out[16]: 51.571442441434684
```

References:

- https://towardsdatascience.com/how-to-easily-deploy-machine-learning-models-using-flask-b95af8fe34d4
- https://medium.datadriveninvestor.com/deploy-your-machine-learning-model-using-flask-made-easy-now-635d2f12c50c