Name: Hamza Al Hajj Chehade

Batch Code: LISUM02

Submission Date: 15-08-2021

Submitted to: Data Glacier

DEPLOYMENT ON FLASK:

➢ Step1:

Develop the ML model: The competition goal is to predict the profit of startup profit on the bases of data provided which are on the bases of Research and Development Spend (R&D Spend), Administration Spend, Marketing Spend and State. We use multiple regression in this model because we have to predict profit (dependent variable) on bases of multiple field (independent variables) rather then one field just like we done in Simple Linear Regression. This model can help those people who want to invest in startup company by analyzing profit of the company.

```
#Importing the Libraries:
import numpy as np
import pandas as pd
import pickle



#Importing the Dataset:
df=pd.read_csv('C:/Users/mizoh/Desktop/Data Glacier/50_Startups.csv')
X=df.iloc[:,:3].values
y=df.iloc[:,-1].values
```

```
'''#Encoding the categorical variables:
from sklearn.preprocessing import LabelEncoder
labelencoder_X=LabelEncoder()
X[:,3]=labelencoder_X.fit_transform(X[:,3])'''

#Splitting the dataset into Training set and Test set:
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20)

#Linear Regression Model:
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(X_train,y_train)
y_pred=regressor.predict(X_test)
```

➢ Step2:

Save training model to a file using *pickle* library.

```
#Serialization (Convert python model into a file):
with open('model.pkl','wb') as model:
    pickle.dump(regressor,model)
```

➢ Step3:

```
from flask import Flask,render_template,request

app=Flask(__name__)
```

```
model=pickle.load(open('model.pkl','rb'))

@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
    int_features=[float(x) for x in request.form.values()]
    final_features=[np.array(int_features)]
    prediction=model.predict(final_features)

    output = round(prediction[0], 2)
    return render_template('index.html', prediction_text='Profit is
{}'.format(output))
if __name__ == "__main__":
    app.run(debug=True)
```

- Creating the instance of the *Flask()* and load the model.
- Bounded "/ " with the *predict()* in which predict method gets the data from
  The loaded model passed by the requester.
- *model.predict()* method takes input and convert them into numpy array and the results are stored into the variable output.
- Returning this object using *render_template*.

Step4:

Open the WEB app using the CMD Prompt by typing python app.py