Deployment on Flask

Name: Raghavi Gururajan

Batch code: LISUM11:30

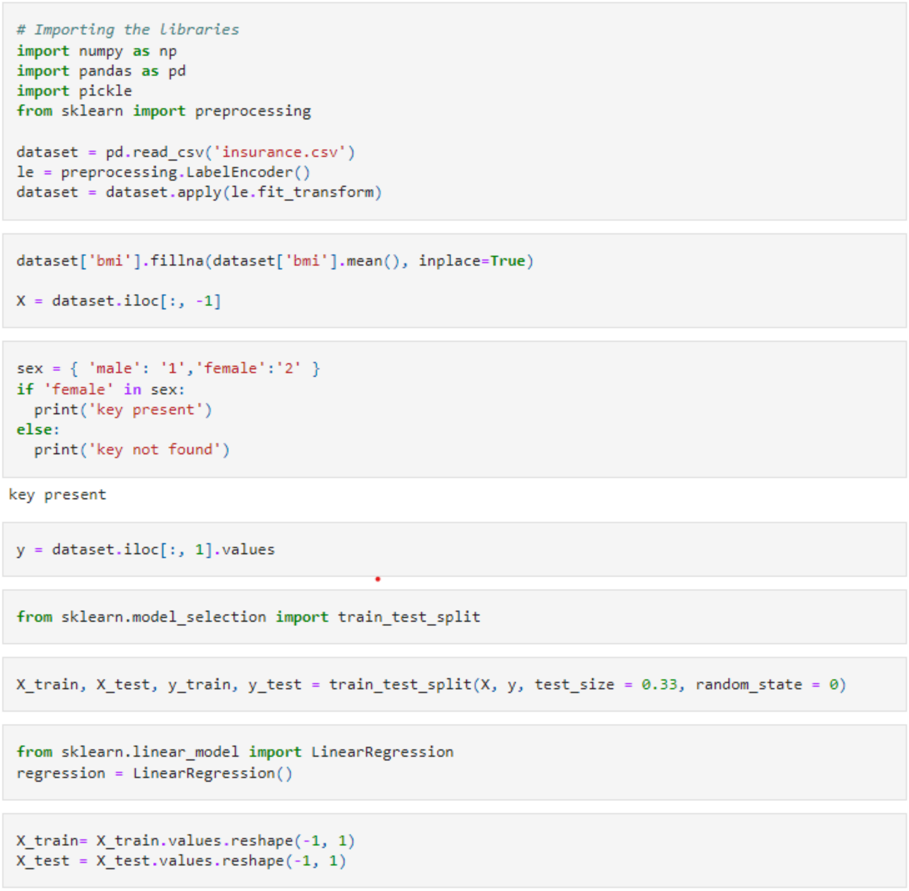
Submission date:25/8/2022

Submitted to: Data Glacier

Deployment procedure

Step 1:

Develop ML model: As a part of Data analysis Internship, I have used Linear Regression for the medical insurance personal dataset



```
# Importing the libraries
import numpy as np
import pandas as pd
import pickle
from sklearn import preprocessing

dataset = pd.read_csv('insurance.csv')
le = preprocessing.LabelEncoder()
dataset = dataset.apply(le.fit_transform)
```

```
dataset['bmi'].fillna(dataset['bmi'].mean(), inplace=True)

X = dataset.iloc[:, -1]
```

```
sex = { 'male': '1','female':'2' }
if 'female' in sex:
  print('key present')
else:
  print('key not found')
```

```
key present
```

```
y = dataset.iloc[:, 1].values
```

```
from sklearn.model_selection import train_test_split
```

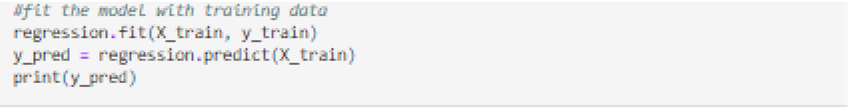
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
```

```
from sklearn.linear_model import LinearRegression
regression = LinearRegression()
```

```
X_train= X_train.values.reshape(-1, 1)
X_test = X_test.values.reshape(-1, 1)
```

Step2:

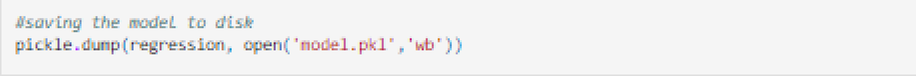
Fitted the model with the training dataset



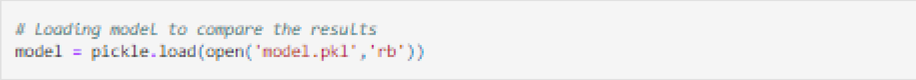
```
#fit the model with training data
regression.fit(X_train, y_train)
y_pred = regression.predict(X_train)
print(y_pred)
```

Step3:

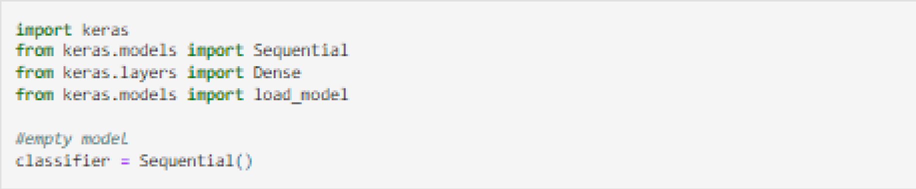
Saved and loaded the model to the disk to compare the results



```
#saving the model to disk
pickle.dump(regression, open('model.pkl','wb'))
```

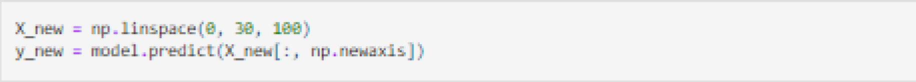


```
# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
```



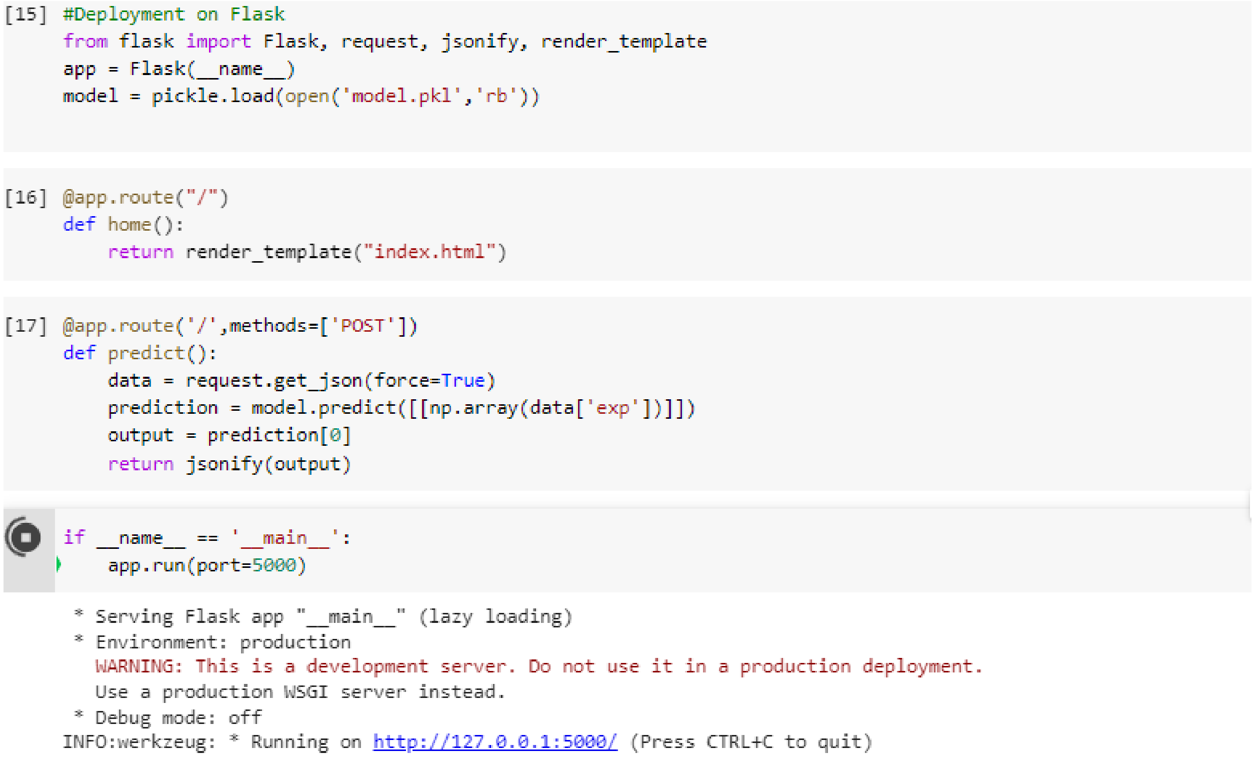
```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.models import load_model

#empty model
classifier = Sequential()
```



```
X_new = np.linspace(0, 30, 100)
y_new = model.predict(X_new[:, np.newaxis])
```

Step4:   Deploy the model



```
[15] #Deployment on Flask
     from flask import Flask, request, jsonify, render_template
     app = Flask(__name__)
     model = pickle.load(open('model.pkl','rb'))

[16] @app.route("/")
     def home():
         return render_template("index.html")

[17] @app.route('/',methods=['POST'])
     def predict():
         data = request.get_json(force=True)
         prediction = model.predict([[np.array(data['exp'])]])
         output = prediction[0]
         return jsonify(output)

     if __name__ == '__main__':
         app.run(port=5000)

      * Serving Flask app "__main__" (lazy loading)
      * Environment: production
        WARNING: This is a development server. Do not use it in a production deployment.
        Use a production WSGI server instead.
      * Debug mode: off
     INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```