Flask Deployment for Salaries of Economics Majors

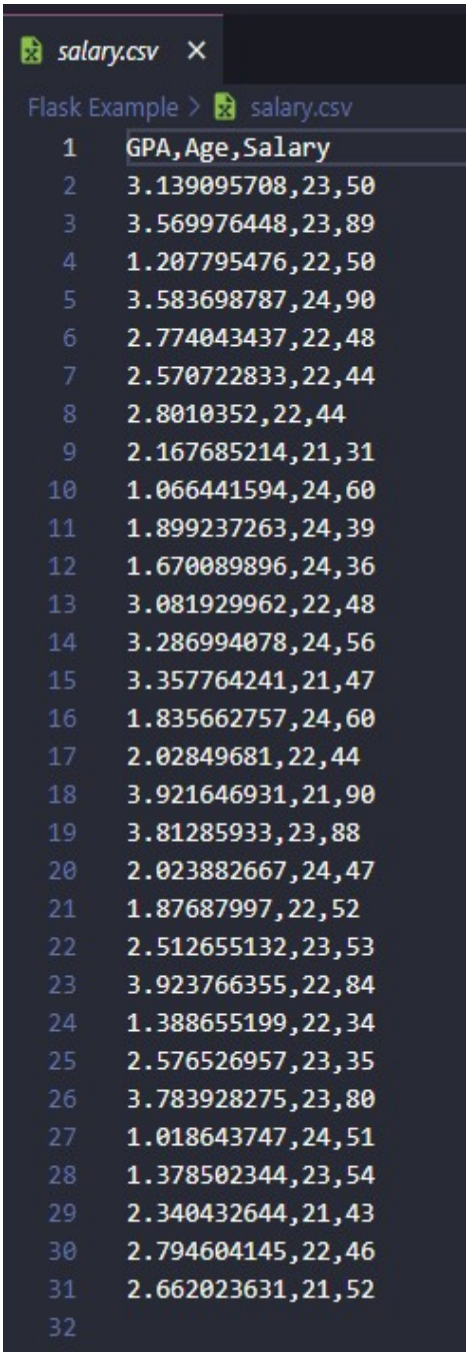
Name: Anthony Ghimpu

Batch Code: LISUM11

Submission Date: July 25, 2022

Submitted To: GitHub

Step 1: Download toy data set of Economic Salaries, ages, and GPAs. (Saved in GitHub folder)



Step 2: Create a machine learning model to predict salaries based on GPA and Age.

model.py

Flask Example > model.py > ...

```python
# Importing the libraries
import numpy as np
import pandas as pd
import pickle

dataset = pd.read_csv('salary.csv')

X = dataset.iloc[:, :2]

y = dataset.iloc[:, -1]

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with trainig data
regressor.fit(X, y)

```

Here, we are using a basic linear regression to predict the salary. First, we partition the data frame so we can fit the data to a linear regression. **X** being the first two columns of the data (GPA and Age), and **y** being the last column (salary).

Step 3: Dump contents of model file into a pickle file to sterilize the contents. (Optional – Testing the model before deployment)

```python
# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[4.0, 45]]))
```

You must create a pickle file within your directory before dumping. In this case, *model.pkl.*

Step 4: Create template files (downloaded from DG) which contains an HTML and CSS Files. Must make appropriate changes to each file.

```
model.py   index.html ×
Flask Example > templates > index.html > ...
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
  <h1>Predict Salary For Economics Majors</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
    <input type="text" name="GPA" placeholder="GPA" required="required" />
        <input type="text" name="Age" placeholder="Age" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
</html>
```

Here, the title and input titles must be changed in *index.html.*

Step 5: Create an *app.py* file and import appropriate libraries.

```
model.py   app.py ×
Flask Example > app.py > ...
1   import numpy as np
2   from flask import Flask, request,render_template
3   import pickle
```

Step 6: Open our pickle file with reading parameter. Create a home page with the use of our html file and make the app use route root. (‘/’)

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

@app.route('/')
def home():
    return render_template('index.html')
```

Step 7: We create a second route for the prediction part of the deployment, using methods=POST. Then we render our output using the render_template function.

```
@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [x for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = 1000*round(prediction[0], 2)

    return render_template('index.html', prediction_text='Salary Per Year Should Be ${}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)
```

The list for features is meant to grab the information the user gives within the page itself. We use that information to predict the actual salary based on the given information by turning the list into a numpy array.

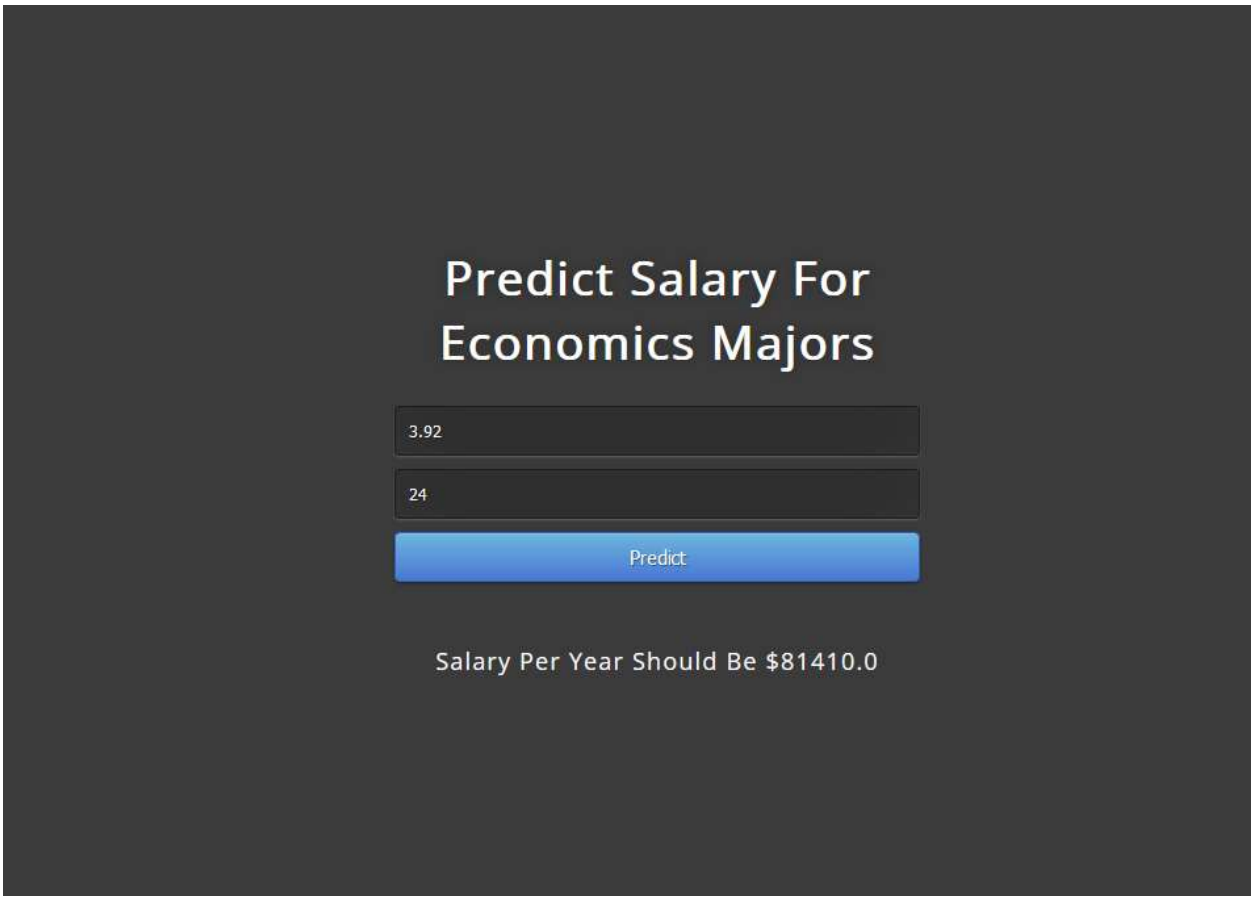
Step 8: Open CMD and run the app.py program.

```
(base) C:\Users\antho>cd flask

(base) C:\Users\antho\Flask>cd flask example

(base) C:\Users\antho\Flask\Flask Example>python app.py
 * Serving Flask app "app" (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: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 105-115-487
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Step 9: Copy and paste the local server link to browser. Run the model using sample data.


Predict Salary For
Economics Majors
3.92
24
Predict
Salary Per Year Should Be $81410.0