Name: Xiaoqing (Lucy) Leng
Batch code: LISUM15
Submission date: 11/26/2022
Submitted to Data Glacier Intern Program

Run App

```
lucyleng@Lucys-Air Flask-Deployment % python3 app.py
/Users/lucyleng/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:329: U
serWarning: Trying to unpickle estimator LinearRegression from version 0.22.1 wh
en using version 1.0.2. This might lead to breaking code or invalid results. Use
 at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintain
ability-limitations
  warnings.warn(
 * Serving Flask app "app" (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 watchdog (fsevents)
/Users/lucyleng/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:329: U
serWarning: Trying to unpickle estimator LinearRegression from version 0.22.1 wh
en using version 1.0.2. This might lead to breaking code or invalid results. Use
 at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintain
ability-limitations
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 115-480-964
127.0.0.1 - - [26/Nov/2022 17:08:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Nov/2022 17:08:44] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [26/Nov/2022 17:08:44] "GET /static/images/Original.svg HTTP/1.1"
200 -
```

Linear regression model predicting GPA

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

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

#dataset['bed_room'].fillna(0, inplace=True)

#dataset['area'].fillna(dataset['area'].mean(), i

X = dataset.iloc[:, :3]

#Converting words to integer values
#def convert_to_int(word):
#    word_dict = {'one':1, 'two':2, 'three':3, 'f
#                 'nine':9, 'ten':10, 'eleven':11,
#    return word_dict[word]

#X['bed_room'] = X['bed_room'].apply(lambda x : c

y = dataset.iloc[:, -1]

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

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

# 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([[2, 2200, 5]]))
```

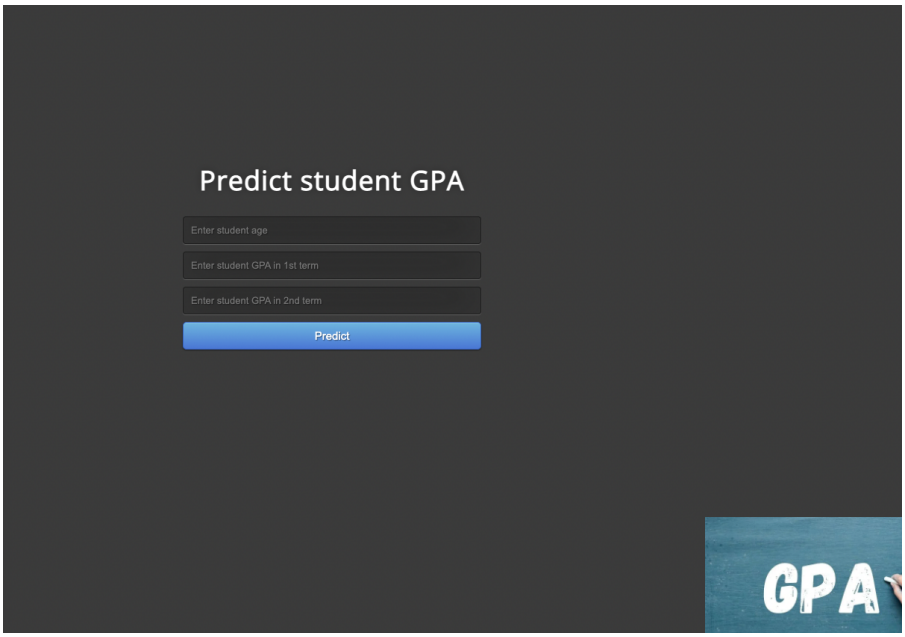
App.py deployment

```python
import numpy as np
from flask import Flask, request,render_template
import pickle

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():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(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='Current GPA is {}'.format(output))

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

HTML

```html
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Simple ML Model 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 student GPA</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="Age" placeholder="Enter student age" required="required" />
        <input type="text" name="GPA at first term" placeholder="Enter student GPA in 1st term" required="requ
        <input type="text" name="GPA at second term" placeholder="Enter student GPA in 2nd term" required="req

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

   <br>
   <br>
   {{ prediction_text }}

 </div>
 <img src="/static/images/GPA-Chalkboard.png" style="width: 400px;position: absolute;bottom: 10px;right: 10px;
</body>
</html>
```

Website page


Predict student GPA
Enter student age
Enter student GPA in 1st term
Enter student GPA in 2nd term
Predict
GPA