Name: Maria Contractor

Batch Code: LISUM22

Submission Date: June 28, 2023

Submitted to: Data Glacier

Deployment on Flask

Instructions:

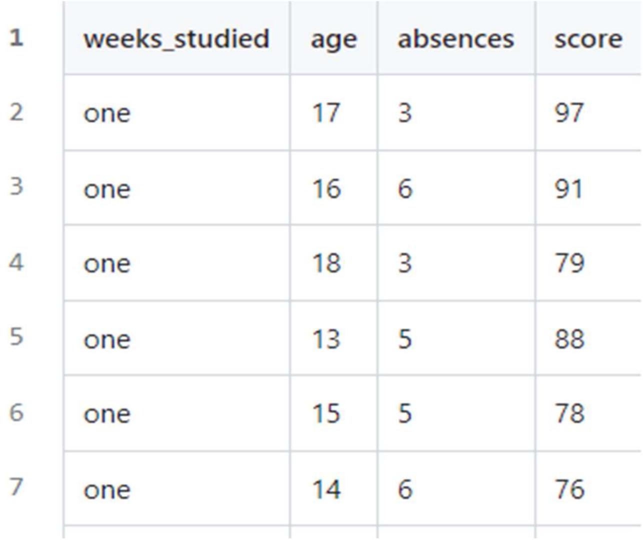
1. Select any toy data (simple data).

2. Save the model

3. Deploy the model on flask

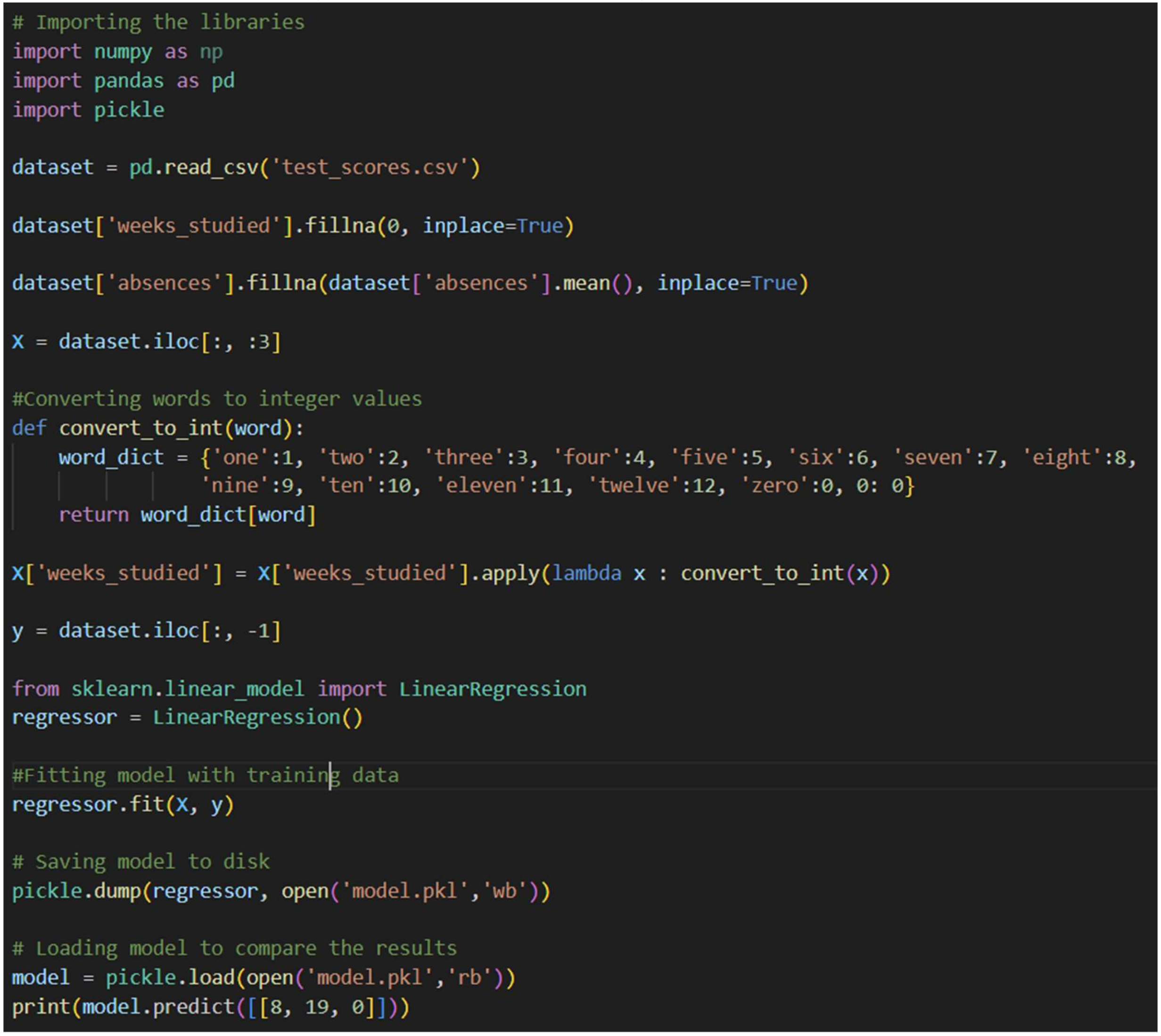
Steps

1. Created a dataset to predict student test scores based on the number of weeks studied, age, and absences



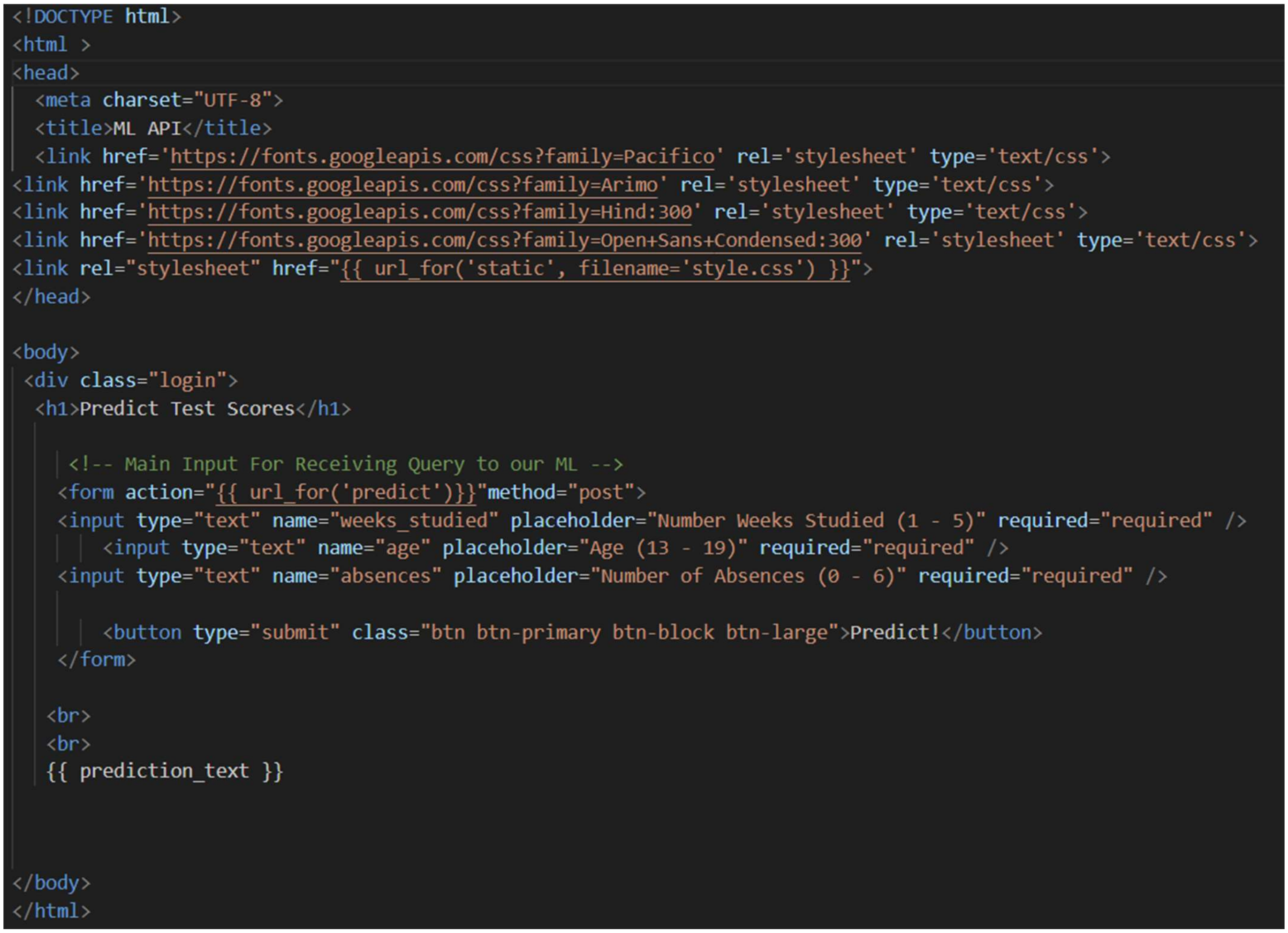
1	weeks_studied	age	absences	score
2	one	17	3	97
3	one	16	6	91
4	one	18	3	79
5	one	13	5	88
6	one	15	5	78
7	one	14	6	76

2. Coded a machine learning model file called model.py with selected data



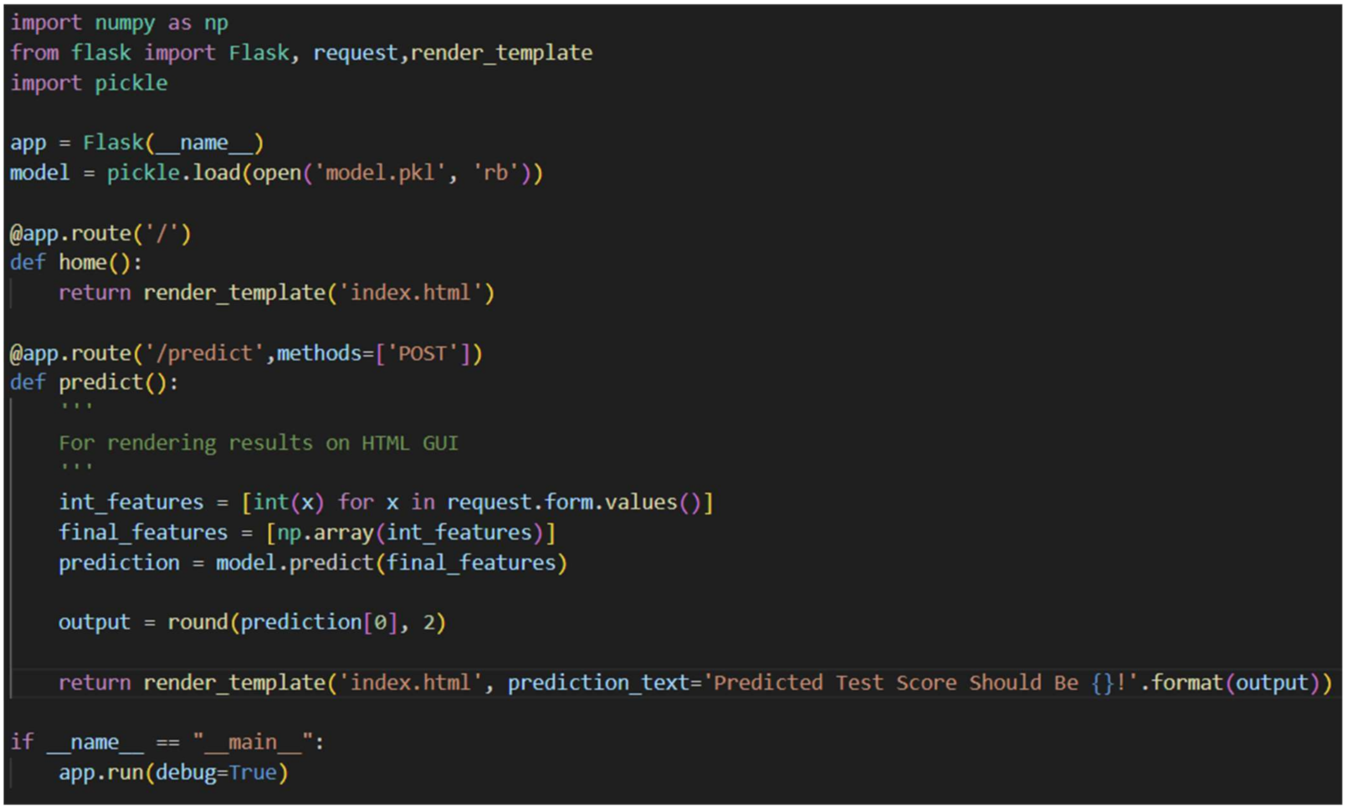
```python
# Importing the libraries
import numpy as np
import pandas as pd
import pickle

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

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

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

X = dataset.iloc[:, :3]

#Converting words to integer values
def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
                'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]

X['weeks_studied'] = X['weeks_studied'].apply(lambda x : convert_to_int(x))

y = dataset.iloc[:, -1]

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with training 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([[8, 19, 0]]))
```

3. Used an html template called index.html to create a website format



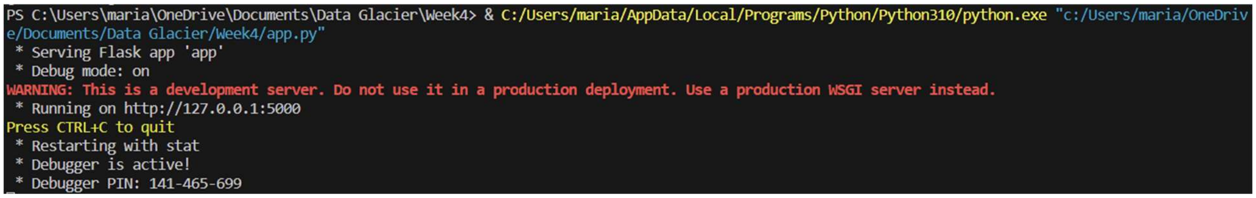
```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' 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='style.css') }}">
</head>

<body>
 <div class="login">
  <h1>Predict Test Scores</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
    <input type="text" name="weeks_studied" placeholder="Number Weeks Studied (1 - 5)" required="required" />
        <input type="text" name="age" placeholder="Age (13 - 19)" required="required" />
    <input type="text" name="absences" placeholder="Number of Absences (0 - 6)" required="required" />

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

   <br>
   <br>
   {{ prediction_text }}



</body>
</html>
```

4. Created app.py which holds the code to predict the test scores



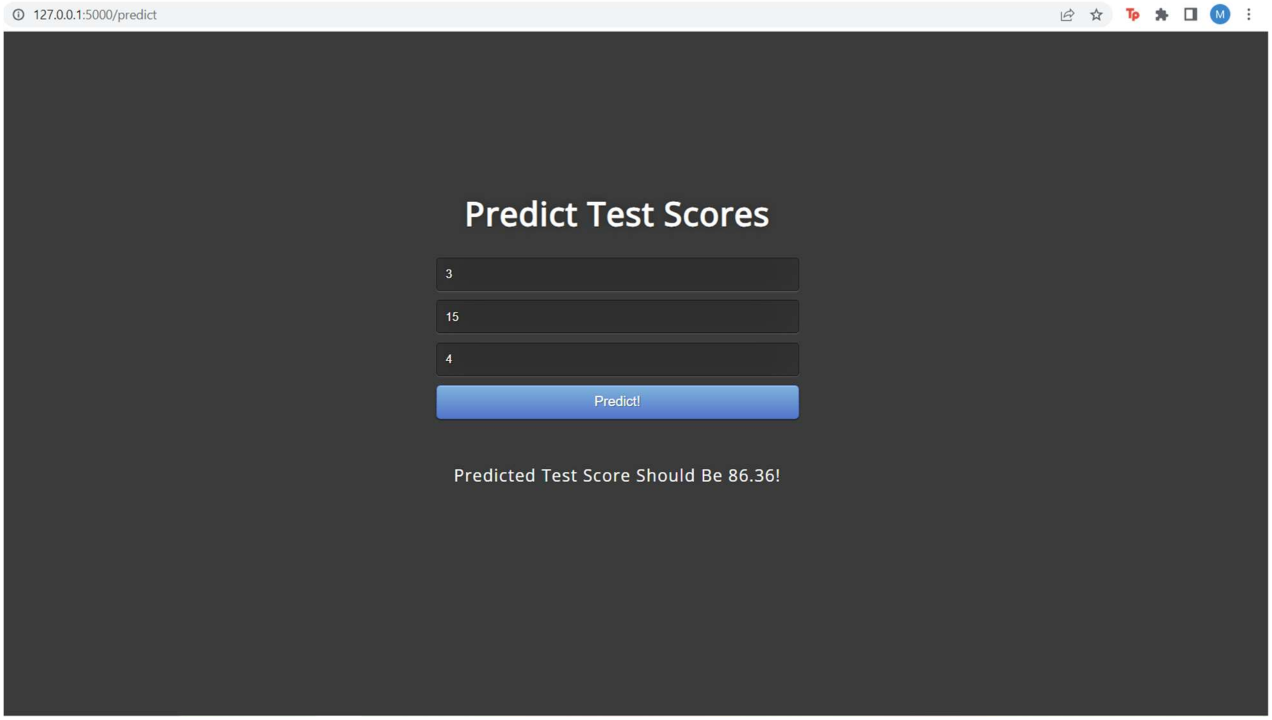
```python
import numpy as np
from flask import Flask, request,render_template
import pickle

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='Predicted Test Score Should Be {}!'.format(output))

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

5. After running app.py, the terminal creates a website that includes all the information to predict scores



```
PS C:\Users\maria\OneDrive\Documents\Data Glacier\Week4> & C:/Users/maria/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/maria/OneDriv
e/Documents/Data Glacier/Week4/app.py"
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 141-465-699
```

6. When clicking on the website, we produce our results, where we can predict test scores based on number of weeks studied, age, and absences



127.0.0.1:5000/predict

Predict Test Scores

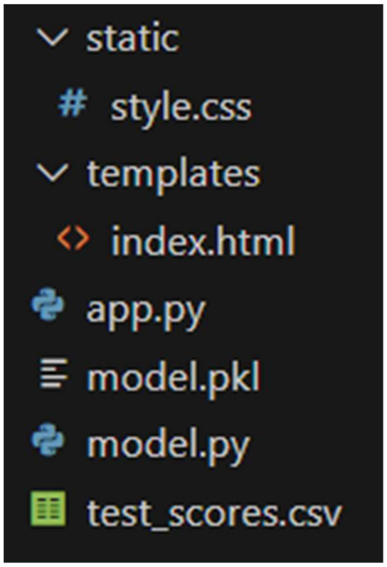
3

15

4

Predicted Test Score Should Be 86.36!

Files Included:



```
∨ static
    # style.css
∨ templates
    <> index.html
  app.py
  model.pkl
  model.py
  test_scores.csv
```