

NAME: Mazen Hawwa

BATCH: LISUM04

Submission Date: 18<sup>th</sup> October 2021

Submitted to: Data Glacier

Data selected for this assignment is a file for college admissions. It contains 3 features:

1. admit
  2. gpa
  3. gre
- admit gives the final decision of student admitted or not in binary, with 0 meaning not admitted and 1 meaning admitted.
  - gpa is the GPA score of the student
  - gre is the GRE test score of the student

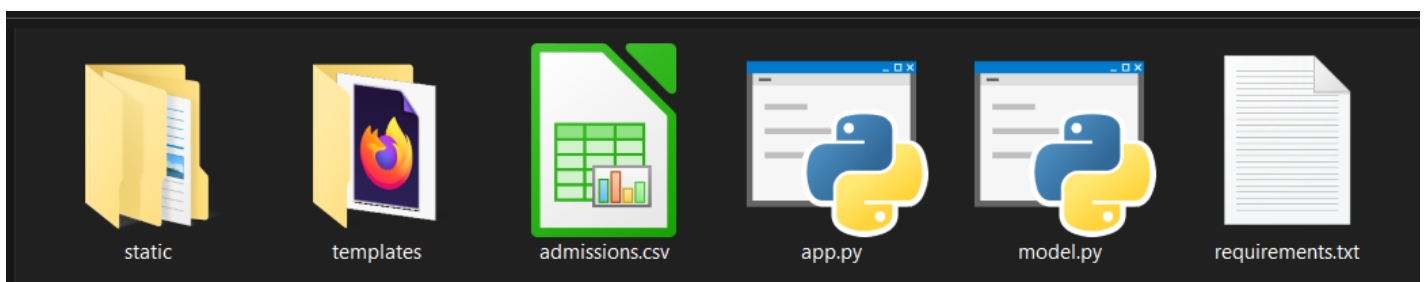
Spyder 5.05 IDE was used to edit the files

Logistic Regression is used to build a model which will predict student admission based on their GPA and GRE test scores.

Application will ask for 2 inputs, GPA and GRE test score, and will give a prediction of Admitted or Not Admitted. If non numeric values are entered or if values are numeric but out of range, then application will inform the user on the screen and ask for proper input values.

Below are screenshots of all major steps:

Preparing the files for the project:



preparing requirements.txt

```
Flask==2.0.2
gunicorn==20.1.0
itsdangerous==2.0.1
Jinja2==3.0.2
MarkupSafe==2.0.1
Werkzeug==2.0.2
numpy>=1.21.2
scipy>=1.7.1
scikit-learn>=1.0
pandas>=1.3.4
```

## index.html

### Changing number of inputs and their text labels

```
app.py X model.py X index.html X
<!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 College Admissions</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}" method="post">
      <input type="text" name="GPA" placeholder="GPA out of 4" required="required" type="number" />
      <input type="text" name="GRE" placeholder="GRE Test Score Out of 800" required="required" type="number"/>

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

    <br>
    <br>
    {{ prediction_text }}
  </div>
  
</body>
</html>
```

## model.py

### changing to logistic regression. Reading in data and using its features to make one text prediction

```
app.py X model.py X index.html X
1  # Importing the libraries
2  import numpy as np
3  import pandas as pd
4  import pickle
5  from sklearn.linear_model import LogisticRegression
6
7  dataset = pd.read_csv('admissions.csv')
8
9  X = dataset[['gpa', 'gre']]
10 y = dataset['admit']
11
12
13 regressor = LogisticRegression()
14
15 #Fitting model with training data
16 regressor.fit(X, y)
17
18 # Saving model to disk
19 pickle.dump(regressor, open('model.pkl', 'wb'))
20
21 # Loading model to compare the results
22 model = pickle.load(open('model.pkl', 'rb'))
23
24 prediction = 'Admitted' if model.predict([[2.9, 750]]) else 'Not Admitted'
25 print()
26 print(prediction)
27 print()
```

## Running model.py

We can see after running model.py that test prediction is working.

```
Python 3.9.6 (default, Jul 30 2021, 11:42:22) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

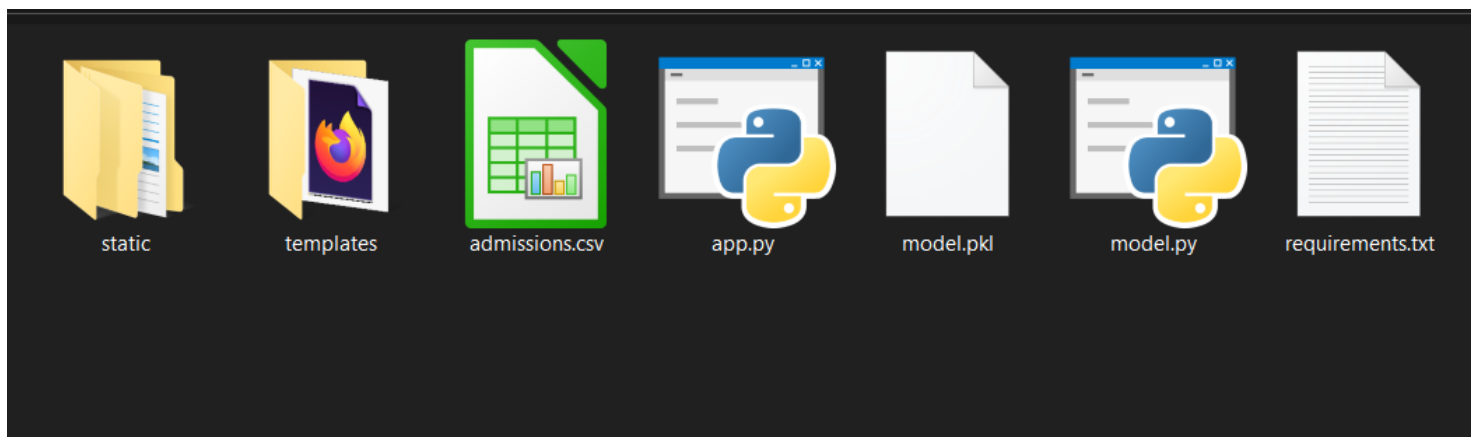
IPython 7.27.0 -- An enhanced Interactive Python.
Restarting kernel...

In [1]: runfile('C:/Users/xcomr/Documents/DATA SCIENCE/Data Science
Internship - Data Glacier/Internship Programme/Week 4/Week 4 Assignment
Submission/model.py', wdir='C:/Users/xcomr/Documents/DATA SCIENCE/Data
Science Internship - Data Glacier/Internship Programme/Week 4/Week 4
Assignment Submission')

Admitted

C:\Users\xcomr\miniconda3\lib\site-packages\sklearn\base.py:441:
UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
warnings.warn(
```

We also have a model.pkl file created



## preparing app.py for our new model and to handle new prediction format

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\xcom

...ta Science Internship - Data Glacier\Internship Programme\Week 4\Week 4 Assignment Submission\app.py

app.py* X model.py X index.html X

1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17     output = ''
18     value_out_of_range = 0
19     value_not_number = 0
20
21     #Get inputs
22     features = [x for x in request.form.values()]
23
24     #Test if inputs are numbers
25     try:
26         features[0] = float(features[0])
27     except ValueError:
28         output += 'You did not enter a number for GPA. \n'
29         value_not_number = 1
30
31     try:
32         features[1] = int(features[1])
33     except ValueError:
34         output += 'You did not enter a number for GRE Score. \n'
35         value_not_number = 1
36
37     if value_not_number:
38         output += 'Please make sure entered values are numbers.'
39         return render_template('index.html', prediction_text=output)
40
41
42
43     #test if inputs within range
44     if features[0] > 4.0 or features[0] < 0:
45         output += 'GPA out of range.\n'
46         value_out_of_range = 1
47
48     if features[1] > 800 or features[1] < 0:
49         output += 'GRE Score out of range.\n'
50         value_out_of_range = 1
51
52     if value_out_of_range:
53         output += 'Please make sure values are in range.\n'
54         return render_template('index.html', prediction_text=output)
55
56
57     #Output final result
58     else:
59         prediction = model.predict([features])
60         output = 'Admitted' if prediction else 'Not Admitted'
61         return render_template('index.html', prediction_text='Student is {}'.format(output))
62
63
64 if __name__ == "__main__":
65     app.run(debug=True)
```

## Running app.py

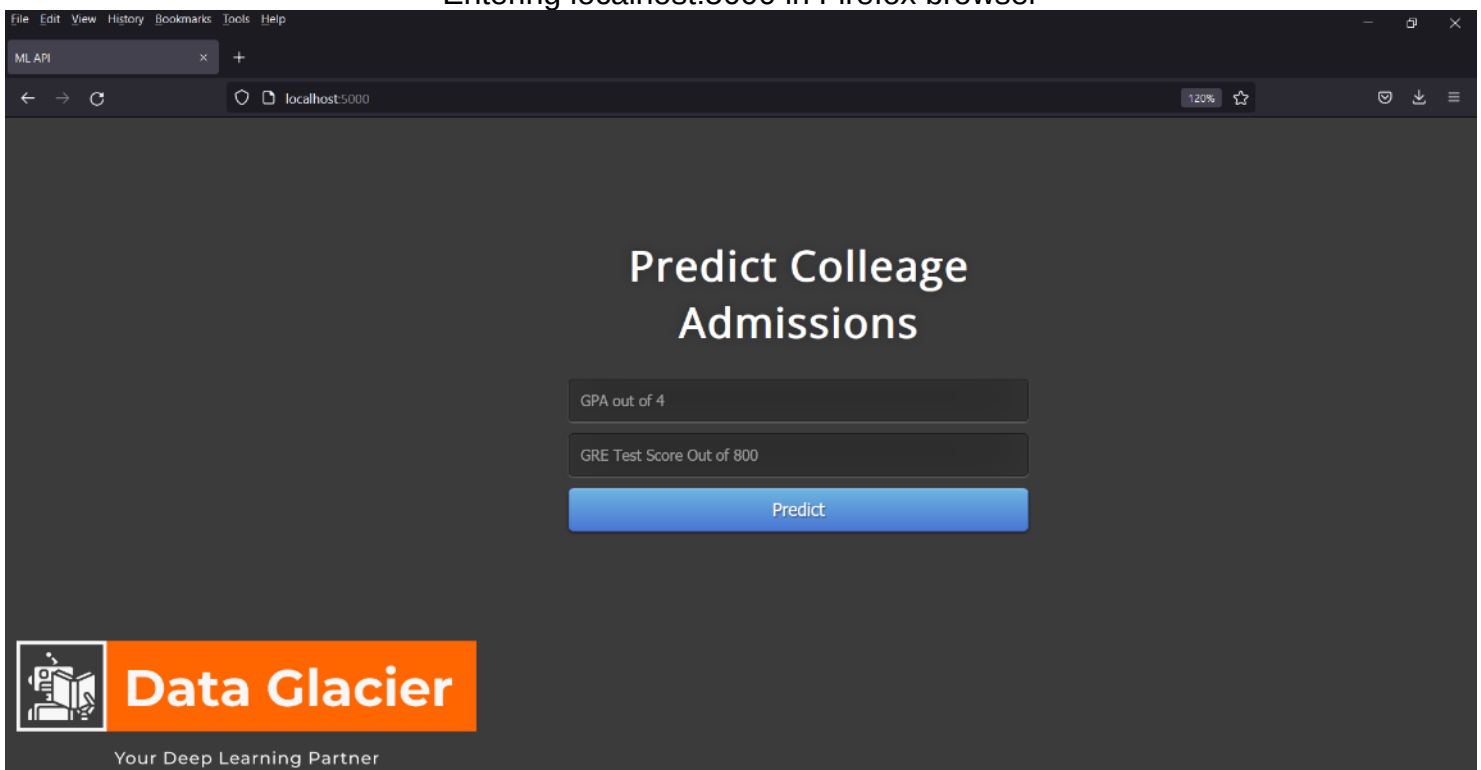
```
Python 3.9.6 (default, Jul 30 2021, 11:42:22) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.27.0 -- An enhanced Interactive Python.

Restarting kernel...

In [1]: runfile('C:/Users/xcomr/Documents/DATA SCIENCE/Data Science Internship -
Data Glacier/Internship Programme/Week 4/Week 4 Assignment Submission/app.py',
wdir='C:/Users/xcomr/Documents/DATA SCIENCE/Data Science Internship - Data Glacier/
Internship Programme/Week 4/Week 4 Assignment Submission')
* 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)
```

## Entering localhost:5000 in Firefox browser



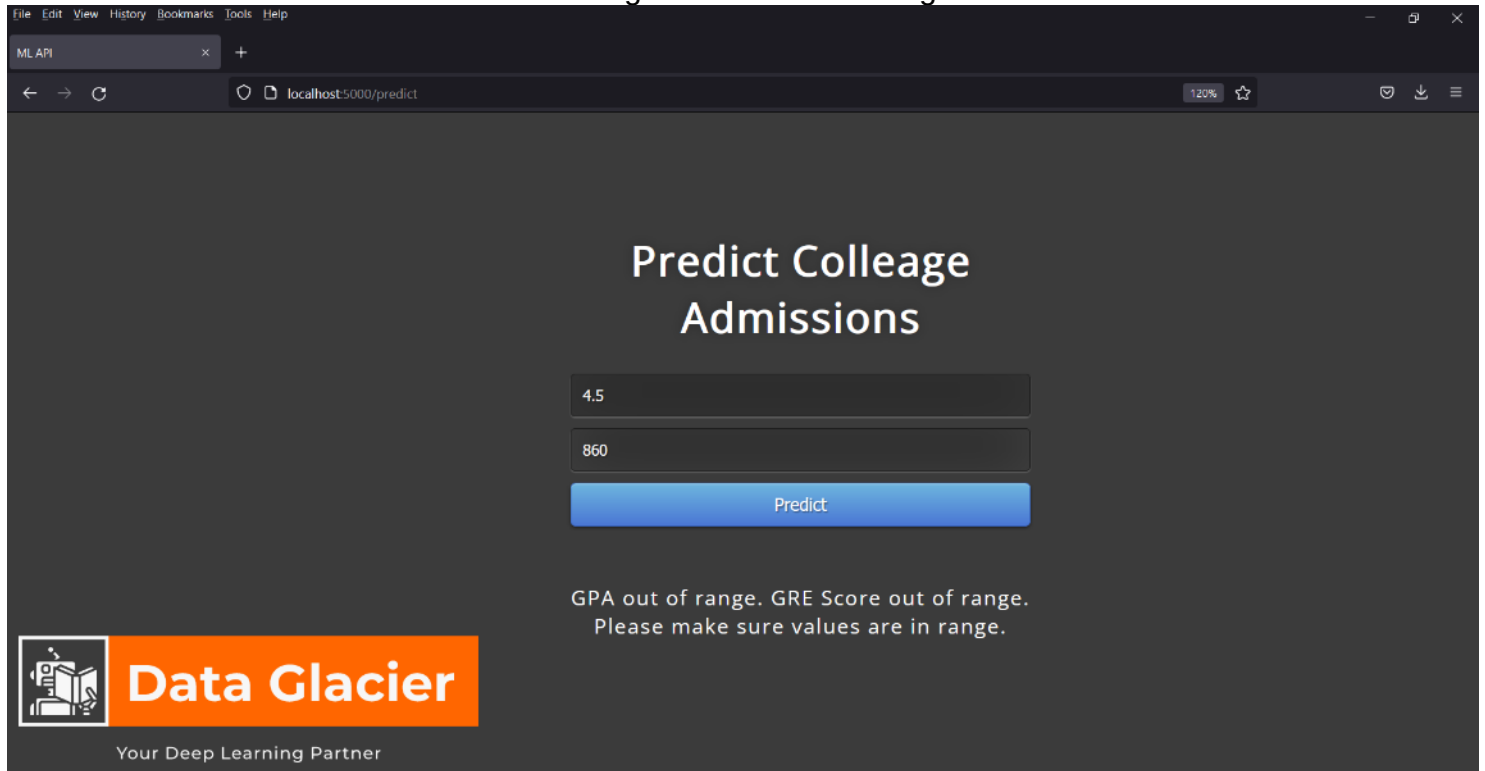
Entering good GPA and GRE score:

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/predict'. The page title is 'Predict College Admissions'. Below the title, there are two input fields: the first contains '3.4' and the second contains '690'. A blue 'Predict' button is positioned below these fields. The message 'Student is Admitted' is displayed in the center of the page. In the bottom left corner, there is a logo for 'Data Glacier' with the tagline 'Your Deep Learning Partner'.

Entering bad GPA and GRE score:

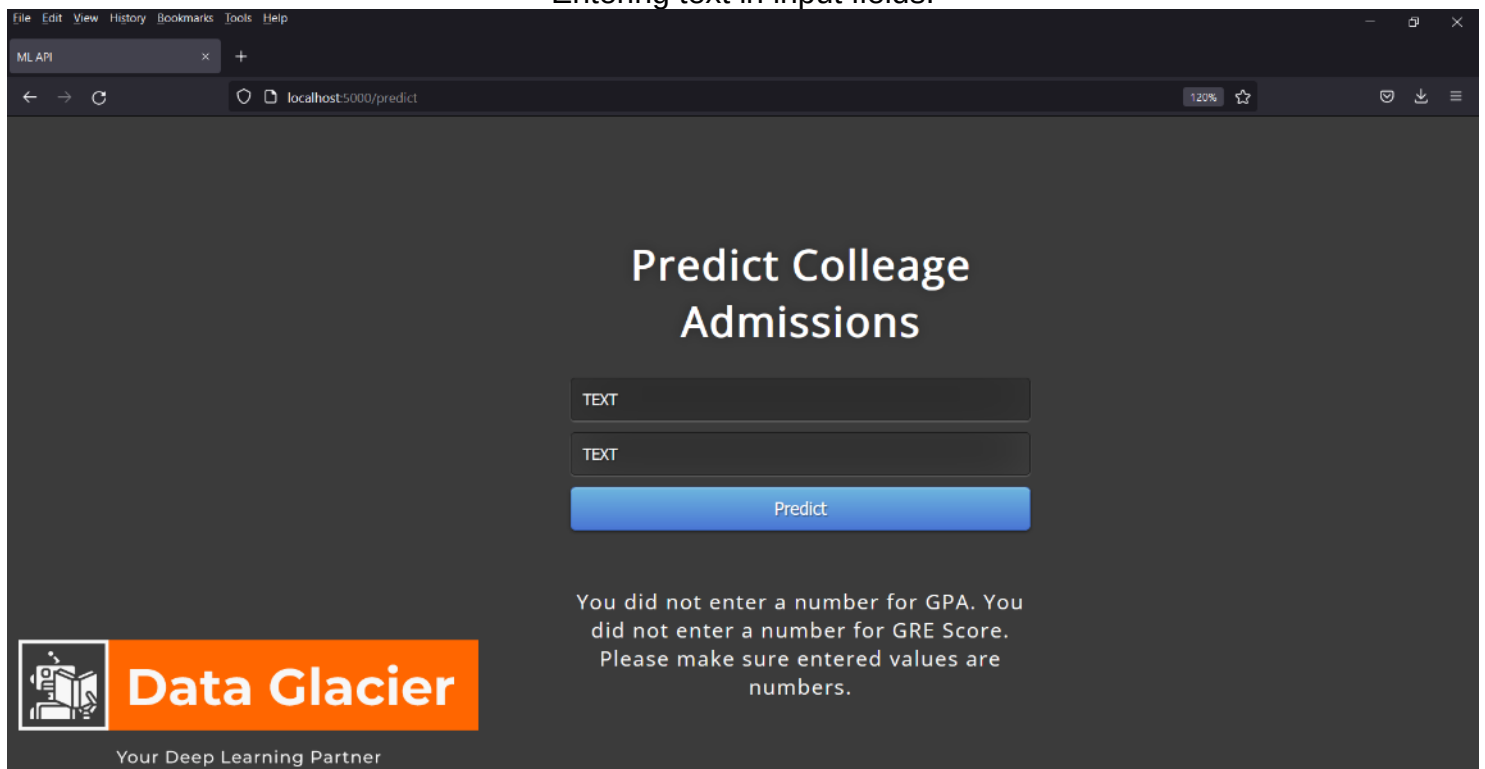
The screenshot shows the same web browser window as the previous one, but with different input values. The first input field now contains '2.5' and the second contains '500'. The 'Predict' button remains blue. The message 'Student is Not Admitted' is displayed in the center of the page. The 'Data Glacier' logo and tagline are still present in the bottom left corner.

Entering numbers out of range:



The screenshot shows a web browser window with the address bar at `localhost:5000/predict`. The page title is "Predict College Admissions". There are two input fields: the first contains "4.5" and the second contains "860". Below these is a blue "Predict" button. A message below the button reads: "GPA out of range. GRE Score out of range. Please make sure values are in range." In the bottom left corner, there is a logo for "Data Glacier" with the tagline "Your Deep Learning Partner".

Entering text in input fields:



The screenshot shows the same web browser window. The two input fields now contain the text "TEXT". The blue "Predict" button is still present. A new message below the button reads: "You did not enter a number for GPA. You did not enter a number for GRE Score. Please make sure entered values are numbers." The "Data Glacier" logo and tagline remain in the bottom left corner.