MODEL DEPLOYMENT ON FLASK

Name: MEMUDU Alimatou sadia

Batch Code: LISUM02

Date of submission: 07/08/2021

Submitted to:

Model Deployment stages

Stage1: Choosing a simple data

This is the data used for this project. The dataset contains several parameters which are considered important during the application for Masters Programs.
The parameters included are:

GRE Scores (out of 340)
TOEFL Scores (out of 120)
University Rating (out of 5)
Statement of Purpose and Letter of Recommendation Strength (out of 5)
Undergraduate GPA (out of 10)
Research Experience (either 0 or 1)
Chance of Admit (ranging from 0 to 1)

Stage2: Build and save a model using Flask

The model's goal is to predict the chance of a student to get admitted into a university. We used Pickle to serialize the model for future use in the admission_model.py file.

```
#Splitting the data
x = data.iloc[:, :-1].values
y = data.iloc[:, 7].values

# split dataset

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)

# Fitting linear regression Regression to the dataset
Lin_regressor = LinearRegression(normalize=True)
Lin_regressor.fit(X_train, y_train)

# To save the model to the disk (serialization) for future use
pickle.dump(Lin_regressor, open('admission_model.pkl', 'wb'))
```

Stage3: Deployment using Flask

Firstly, App.py is built, a flask app that used the deserialized model to accept new data and predict a student percentage to get admission.

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

app = Flask(__name__)

# Performing deserialization using pickle
model = pickle.load(open("admission_model.pkl", 'rb'))


@app.route('/')
def index():
    return render_template(
        'index.html',
        data=[{'UR': 'University Rating'}, {'UR': 1}, {'UR': 2}, {'UR': 3}, {'UR': 4}, {'UR': 5}],
        data1=[{'ReS': 'Research'}, {'ReS': 0}, {'ReS': 1}])

```

Now the function below accepts the data and return the predicted percentage

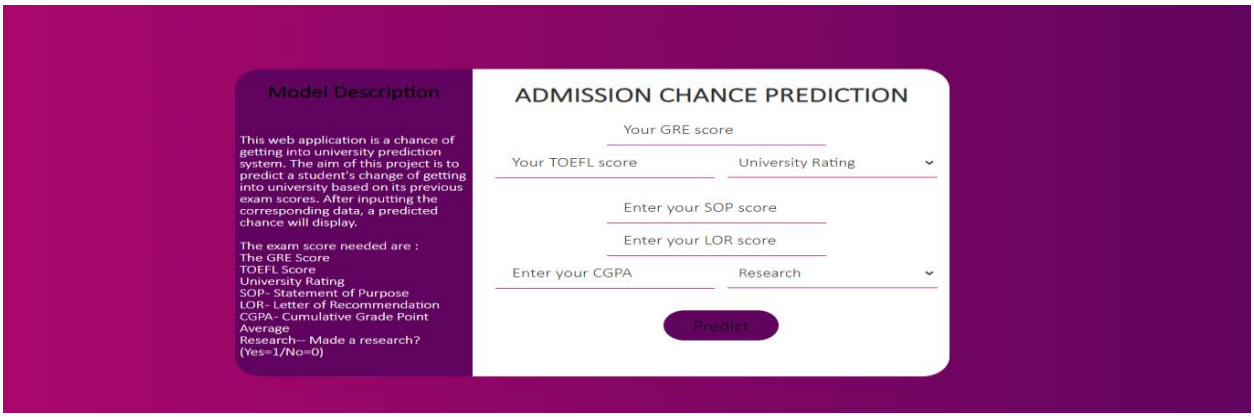
```python
@app.route("/predict", methods=['GET', 'POST'])
def predict():
    input_data = list(request.form.values())
    if int(input_data[0]) & int(input_data[1]) & input_data[3].isdigit() & input_data[4].isdigit() & input_data[5].isdigit() == True:
        pass
    else:
        print(ValueError)

    input_values = [x for x in input_data]
    arr_val = [np.array(input_values)]
    prediction = model.predict(arr_val)
    output = round(prediction[0], 2)*100
    return render_template('index.html', prediction_text=" The Chance of Getting into the University is {} %".format(output),
                           data=[{'UR': 'University Rating'}, {'UR': 1}, {'UR': 2}, {'UR': 3}, {'UR': 4}, {'UR': 5}],
                           data1=[{'ReS': 'Gender'}, {'ReS': 0}, {'ReS': 1}])


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

The index.html is a file that contains the structure of the web app design and AppStyle.css is used to beautify the web design.

Secondly, you have to write: “python app.py” in the terminal to run the flask application then a link will display.

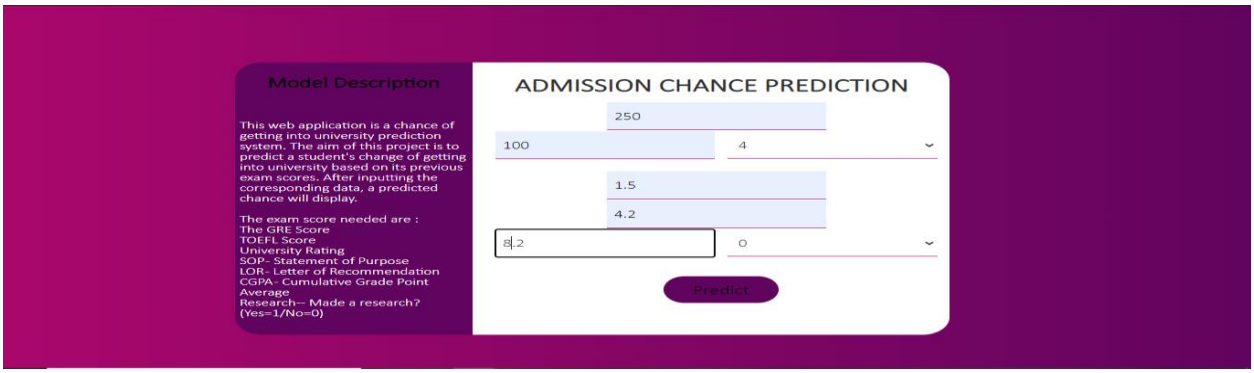
```
Terminal:  Local

PS C:\Users\Alimat sadia\my pyPrograms> cd "data science"
PS C:\Users\Alimat sadia\my pyPrograms\data science> cd "ADMISSION WEB APPLICATION"
PS C:\Users\Alimat sadia\my pyPrograms\data science\ADMISSION WEB APPLICATION> 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 stat
 * Debugger is active!
 * Debugger PIN: 603-954-664
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Thirdly, clicking on the link will direct you to the flask web application interface shown below.

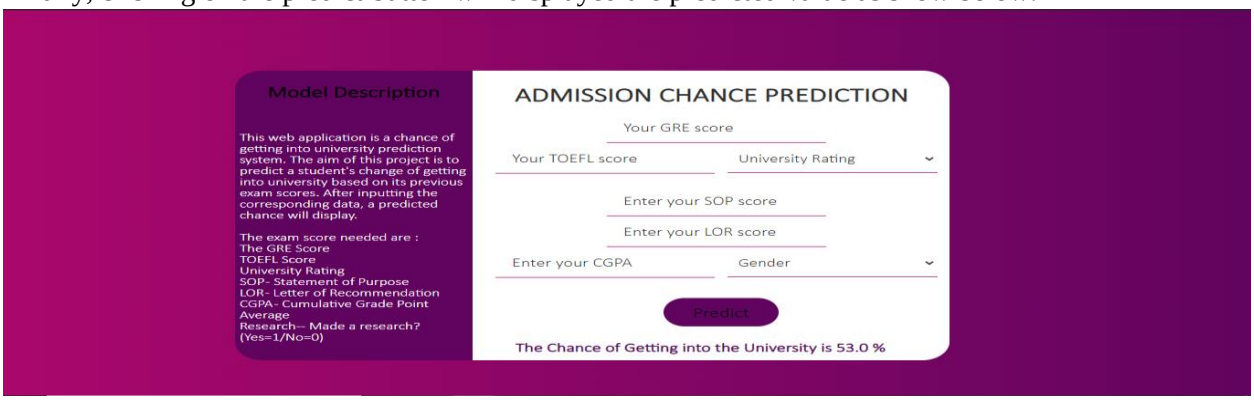


On this interface, a description of the web app's function is explained by the left and on the right there is the input section that collects the user's data.

This data will be fed into the deserialized model which will provide an output (percentage of change to get admitted) as illustrated below.



Finally, clicking on the predict button will displayed the predicted value as show below.



.