Data Science Intern at Data Glacier

Week 4: Deployment on Flask

Name: Chitra Chaudhari

Batch Code: LISUM17

Submission Date: 26 January 2023

Submitted to: Data Glacier

1. Introduction

In this project, we will deploy a machine learning model (Linear Regression) using the Flask Framework. As a demonstration, our model helps to predict Graduate admission.
We will first build a machine learning model for prediction, then create an API for the model, using Flask, the Python micro-framework for building web applications. This API allows us to utilize predictive capabilities through HTTP requests.

2. Data Information

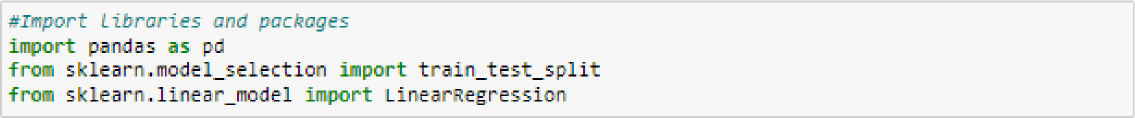
This dataset is created for the prediction of Graduate Admissions from an Indian perspective. This dataset was built with the purpose of helping students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their chances at a particular university. The dataset contains several parameters considered necessary during the application for Masters's Programs.

The parameters included are :

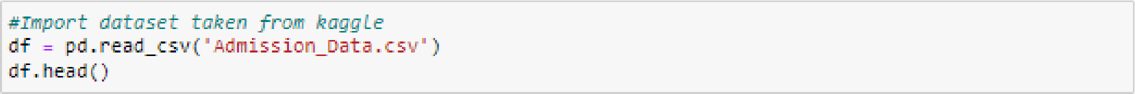
1. GRE Scores (out of 340)
2. TOEFL Scores (out of 120)
3. University Rating (out of 5)
4. Statement of Purpose and Letter of Recommendation Strength (out of 5)
5. Undergraduate GPA (out of 10)
6. Research Experience (either 0 or 1)
7. Chance of Admit (ranging from 0 to 1)

3. Building A Model

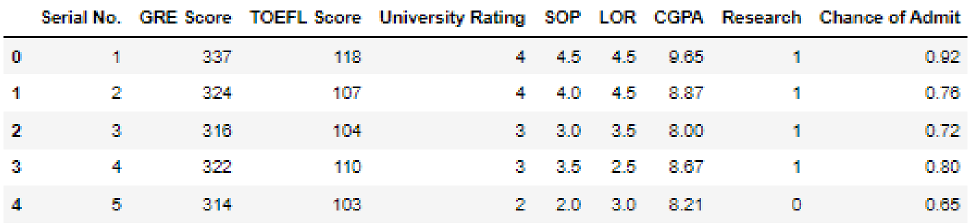
3.1 Importing Require Libraries and Dataset.



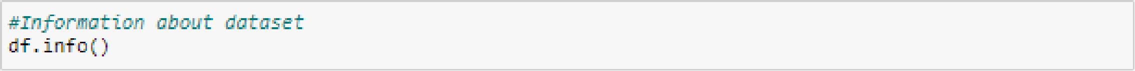
```
#Import libraries and packages
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```



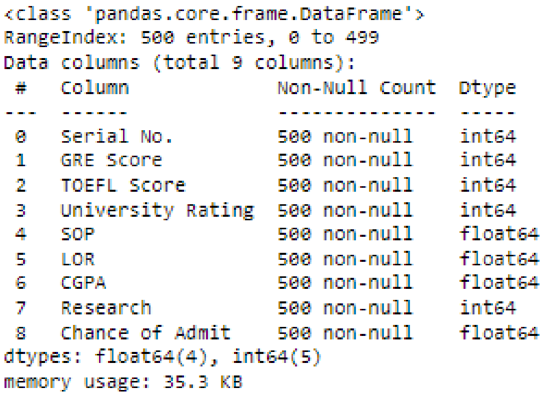
```
#Import dataset taken from kaggle
df = pd.read_csv('Admission_Data.csv')
df.head()
```



	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65



```
#Information about dataset
df.info()
```

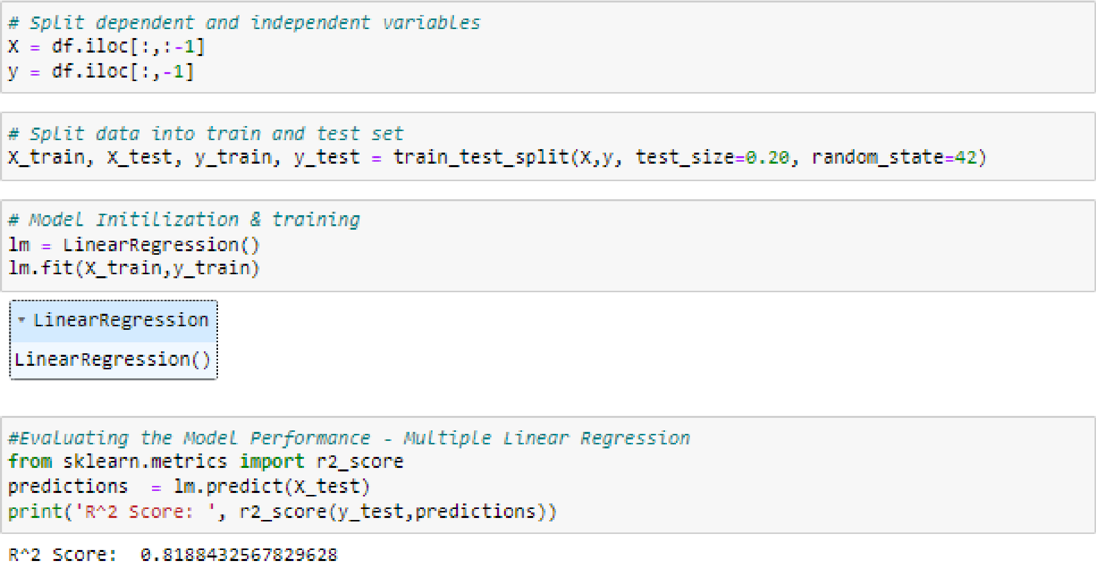


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial No.         500 non-null    int64  
 1   GRE Score          500 non-null    int64  
 2   TOEFL Score        500 non-null    int64  
 3   University Rating  500 non-null    int64  
 4   SOP                500 non-null    float64
 5   LOR                500 non-null    float64
 6   CGPA               500 non-null    float64
 7   Research           500 non-null    int64  
 8   Chance of Admit    500 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 35.3 KB
```

- There are a total of 8 useful features in this dataset, and each of the features has 500 records.
- There are no null or missing values in this dataset.

3.2 Build Model

We will split data into 80% for the training set and the remaining 20% for the test set. Here we implement a machine learning model to predict the applicant's chance of admission. For this purpose, we will use a Multiple linear regression model using scikit-learn. After importing and initializing the LinearRegression model we fit it into the training dataset.



```
# Split dependent and independent variables
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
```

```
# Split data into train and test set
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.20, random_state=42)
```

```
# Model Initilization & training
lm = LinearRegression()
lm.fit(X_train,y_train)
```

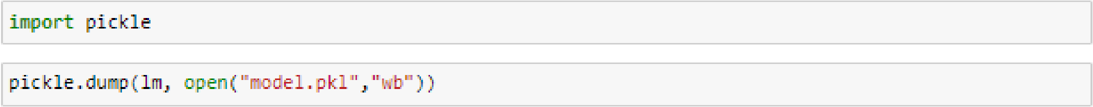
```
▾ LinearRegression
LinearRegression()
```

```
#Evaluating the Model Performance - Multiple Linear Regression
from sklearn.metrics import r2_score
predictions  = lm.predict(X_test)
print('R^2 Score: ', r2_score(y_test,predictions))
```

```
R^2 Score:  0.8188432567829628
```

3.3 Save the Model

We saved our model using pickle.



```
import pickle
```

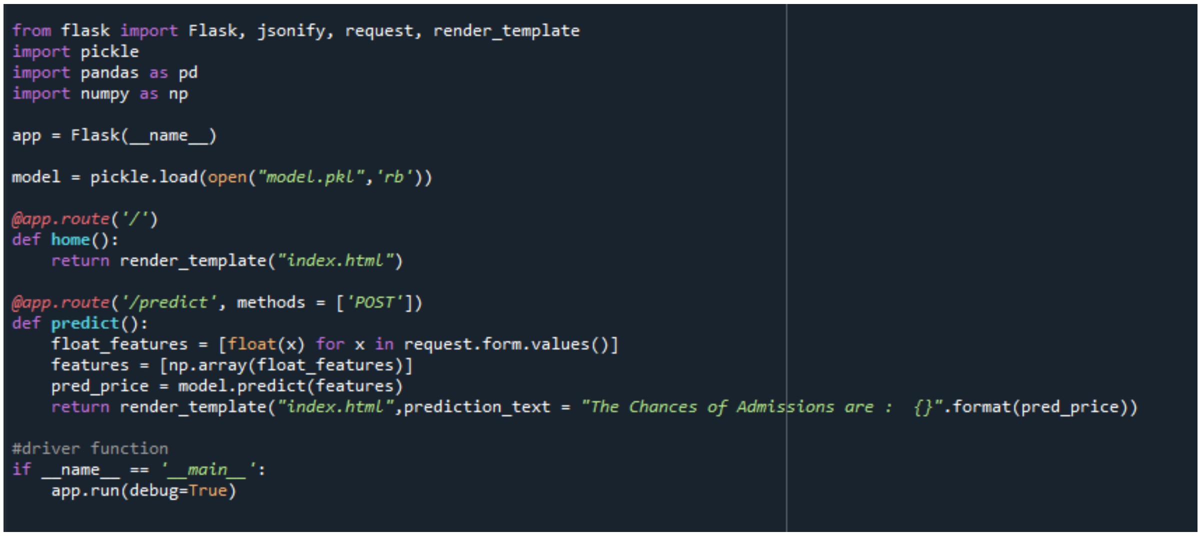
```
pickle.dump(lm, open("model.pkl","wb"))
```

4. Turning Model into Web Application

Now we will develop a web application that consists of a simple web page with a form field that lets us enter a few applicants’ scores. After submitting the features to the web application, it will tell us the applicants' chances of admission.

4.1 App.py

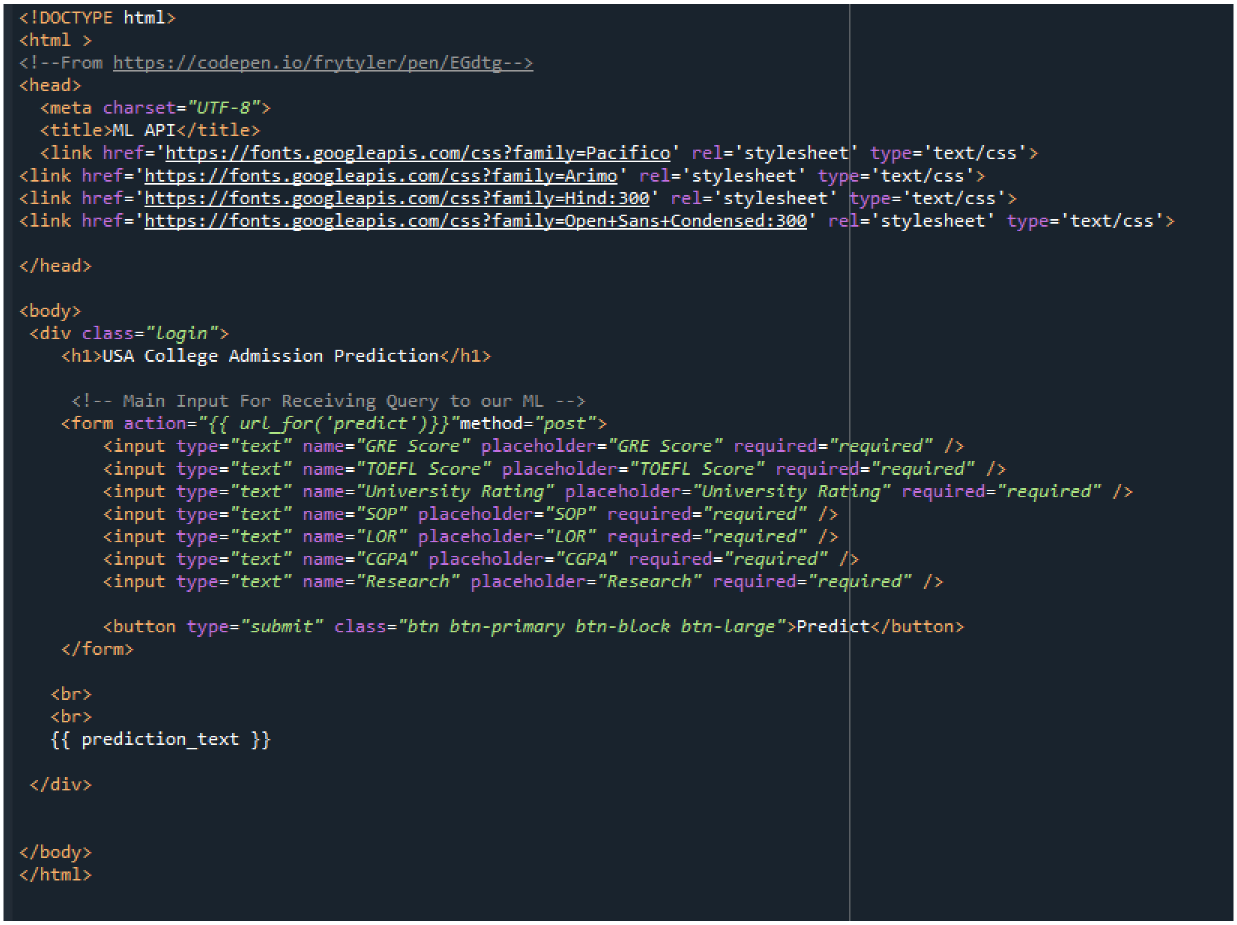
The app.py file contains the main code that will be executed by the Python interpreter to run the Flask web application, it uses t pickle to import the ML model for prediction.



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

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():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    pred_price = model.predict(features)
    return render_template("index.html",prediction_text = "The Chances of Admissions are :  {}".format(pred_price))

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

4.2 Index.html

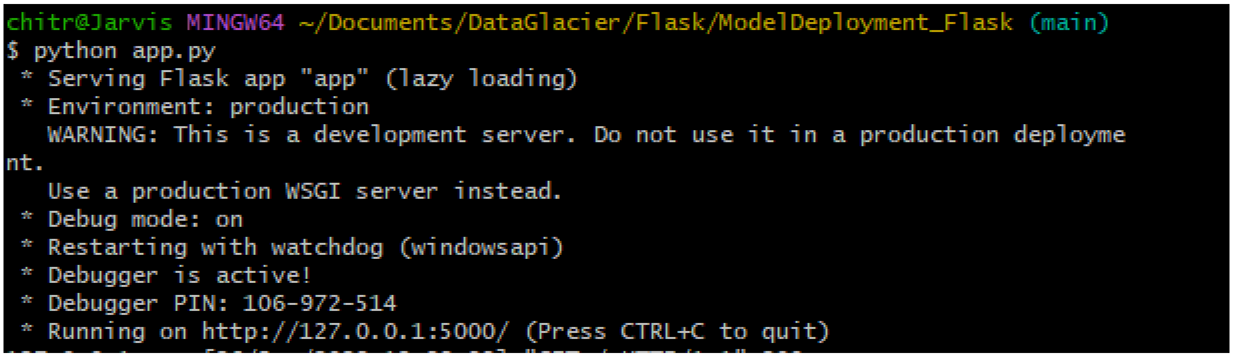
The following are the contents of the index.html file that will render a text form where a user can enter the required attributes for prediction.



```html
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<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'>

</head>

<body>
 <div class="login">
    <h1>USA College Admission Prediction</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="GRE Score" placeholder="GRE Score" required="required" />
        <input type="text" name="TOEFL Score" placeholder="TOEFL Score" required="required" />
        <input type="text" name="University Rating" placeholder="University Rating" required="required" />
        <input type="text" name="SOP" placeholder="SOP" required="required" />
        <input type="text" name="LOR" placeholder="LOR" required="required" />
        <input type="text" name="CGPA" placeholder="CGPA" required="required" />
        <input type="text" name="Research" placeholder="Research" required="required" />

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

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

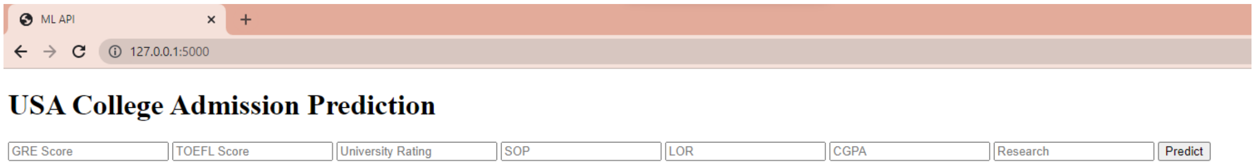
4.3 Running Procedure

Once we have done all of the above, we can start running the API by executing the command from the Terminal:



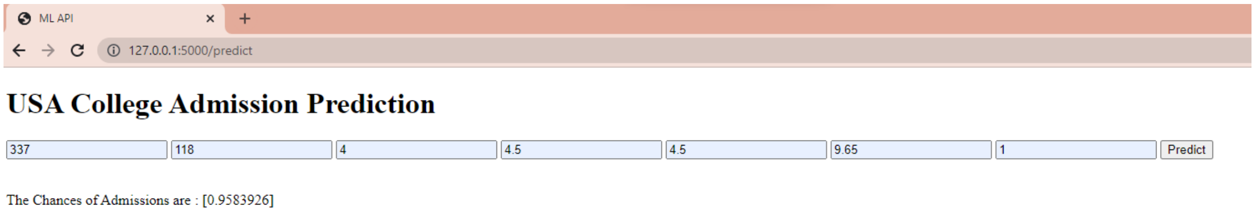
```
chitr@Jarvis MINGW64 ~/Documents/DataGlacier/Flask/ModelDeployment_Flask (main)
$ python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 106-972-514
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Now we could open a web browser and navigate to http://127.0.0.1:5000/, we should see a simple website like shown below



USA College Admission Prediction

Now we need to input the different parameters. After entering the input click the predict button and can see the result for chances of admission..



USA College Admission Prediction

337 | 118 | 4 | 4.5 | 4.5 | 9.65 | 1

The Chances of Admissions are : [0.9583926]

Citation

Mohan S Acharya, Asfia Armaan, Aneeta S Antony: A Comparison of Regression Models for Prediction of Graduate Admissions, IEEE International Conference on Computational Intelligence in Data Science 2019