Model Deployment on Flask

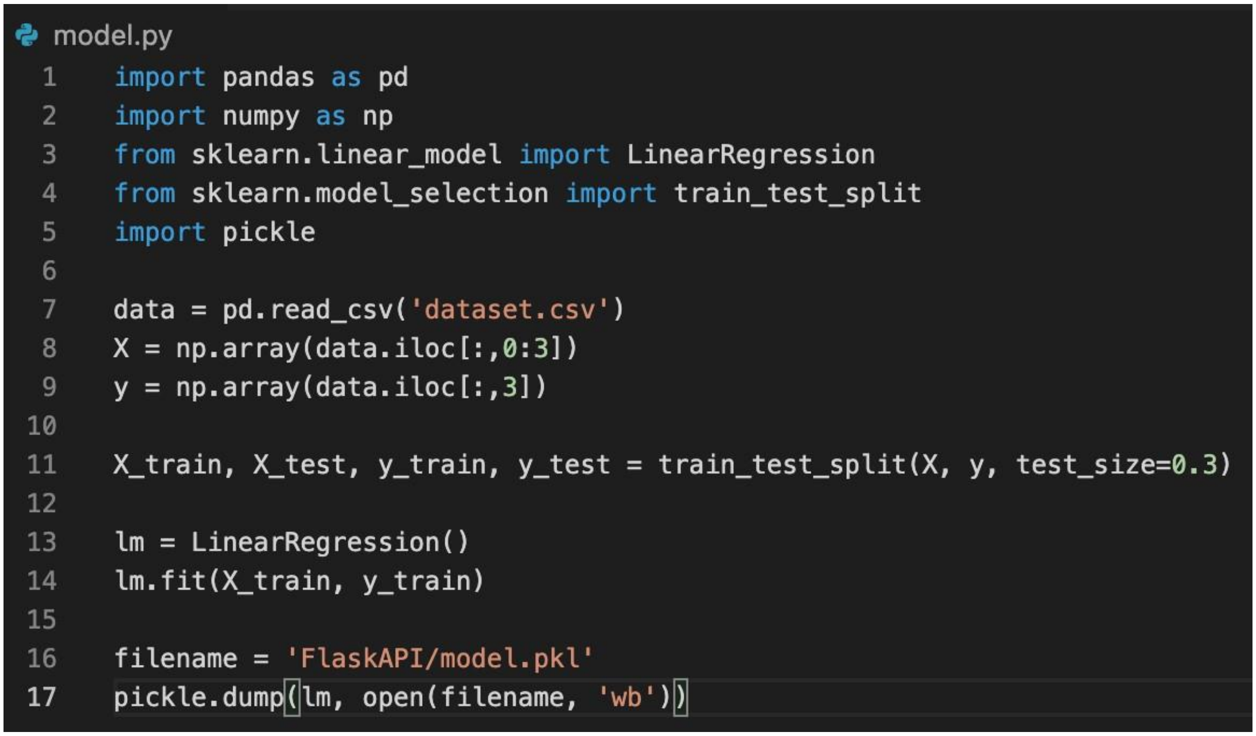
Parth Shah
LISUM01

The Dataset:

The dataset used for this deliverable can be found at the URL https://www.kaggle.com/mohansacharya/graduate-admissions.

The Model:

Below is the code used to train and serialize the model into a pickle file. Pickle was used as it is quite simple to use and understand.



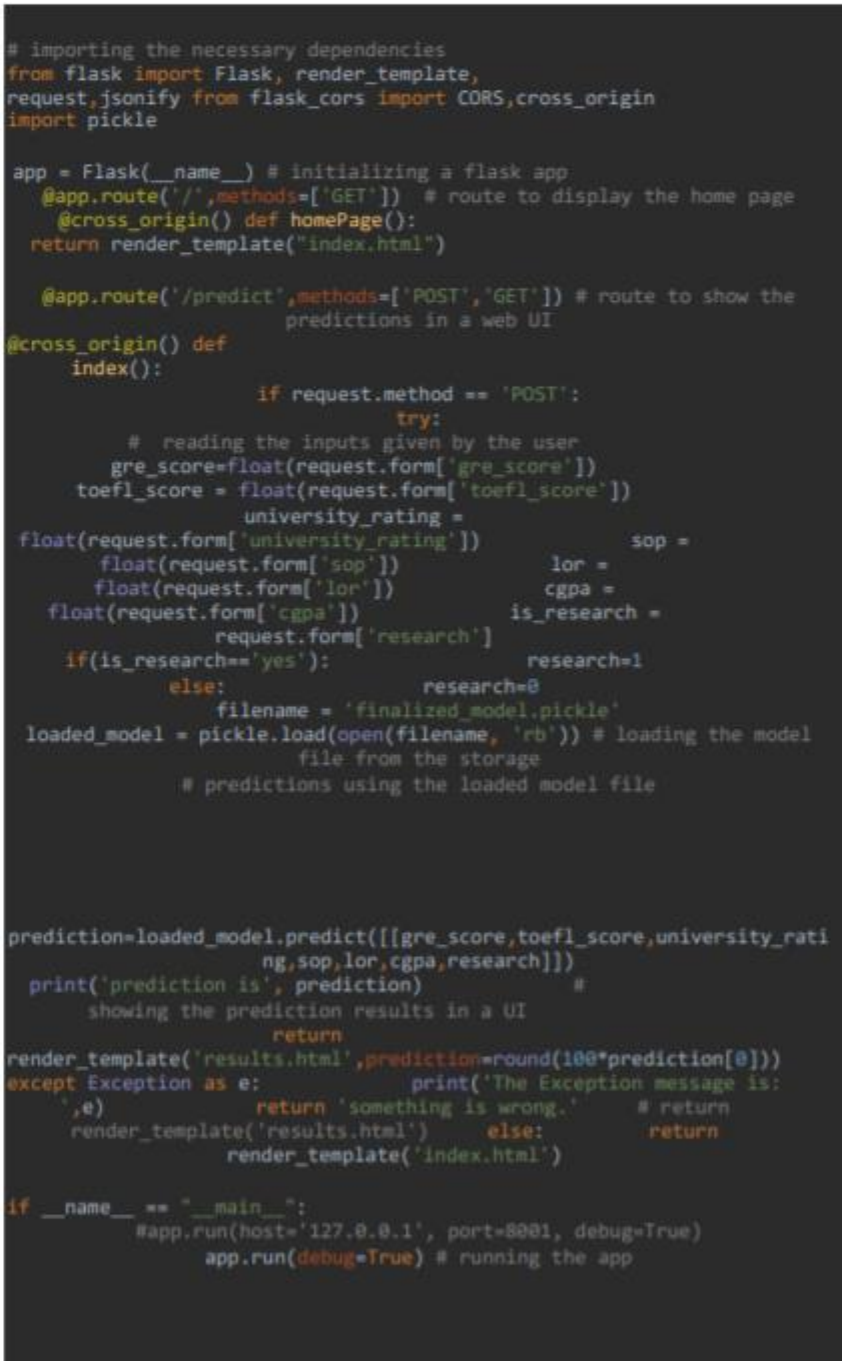
model.py

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pickle

data = pd.read_csv('dataset.csv')
X = np.array(data.iloc[:,0:3])
y = np.array(data.iloc[:,3])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

lm = LinearRegression()
lm.fit(X_train, y_train)

filename = 'FlaskAPI/model.pkl'
pickle.dump(lm, open(filename, 'wb'))
```

Model Deployment:

Once we have our model stored in the pickle file, we can now go on and deploy the model using Flask. It is in this step where we de-serialize the model back into a python object so that we can send some unseen data into our model, through our webpage, and predict an output.

The python script for the deployment of our model is shown below.

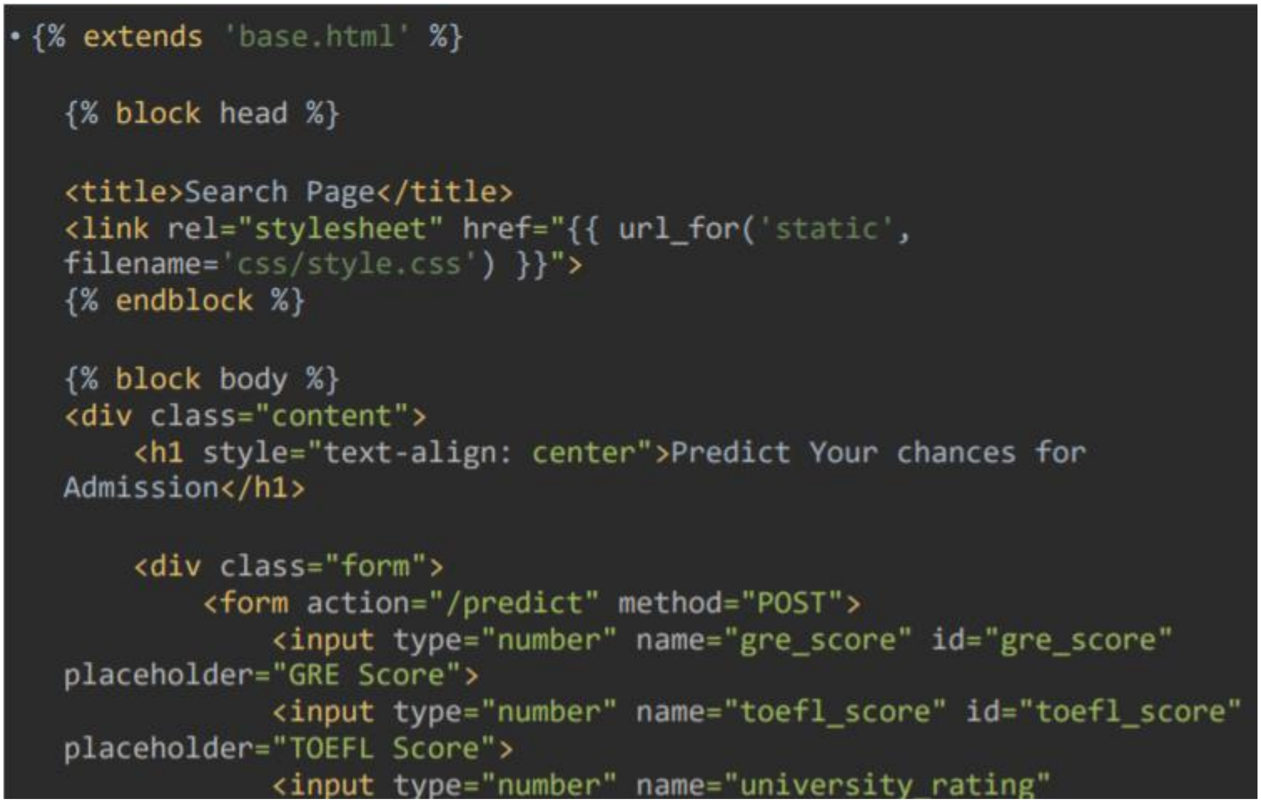


```
# importing the necessary dependencies
from flask import Flask, render_template,
request,jsonify from flask_cors import CORS,cross_origin
import pickle

 app = Flask(__name__) # initializing a flask app
    @app.route('/',methods=['GET'])  # route to display the home page
      @cross_origin() def homePage():
   return render_template("index.html")

    @app.route('/predict',methods=['POST','GET']) # route to show the
                               predictions in a web UI
@cross_origin() def
        index():
                              if request.method == 'POST':
                                               try:
            #  reading the inputs given by the user
           gre_score=float(request.form['gre_score'])
       toefl_score = float(request.form['toefl_score'])
                             university_rating =
 float(request.form['university_rating'])              sop =
          float(request.form['sop'])             lor =
         float(request.form['lor'])            cgpa =
     float(request.form['cgpa'])             is_research =
                        request.form['research']
       if(is_research=='yes'):                 research=1
                    else:                research=0
                        filename = 'finalized_model.pickle'
  loaded_model = pickle.load(open(filename, 'rb')) # loading the model
                                file from the storage
                     # predictions using the loaded model file




prediction=loaded_model.predict([[gre_score,toefl_score,university_rati
                              ng,sop,lor,cgpa,research]])
  print('prediction is', prediction)              #
       showing the prediction results in a UI
                               return
render_template('results.html',prediction=round(100*prediction[0]))
except Exception as e:            print('The Exception message is:
      ',e)             return 'something is wrong.'     # return
     render_template('results.html')     else:         return
                     render_template('index.html')

if __name__ == "__main__":
           #app.run(host='127.0.0.1', port=8001, debug=True)
                  app.run(debug=True) # running the app
```

<u>HTML Webpage:</u>

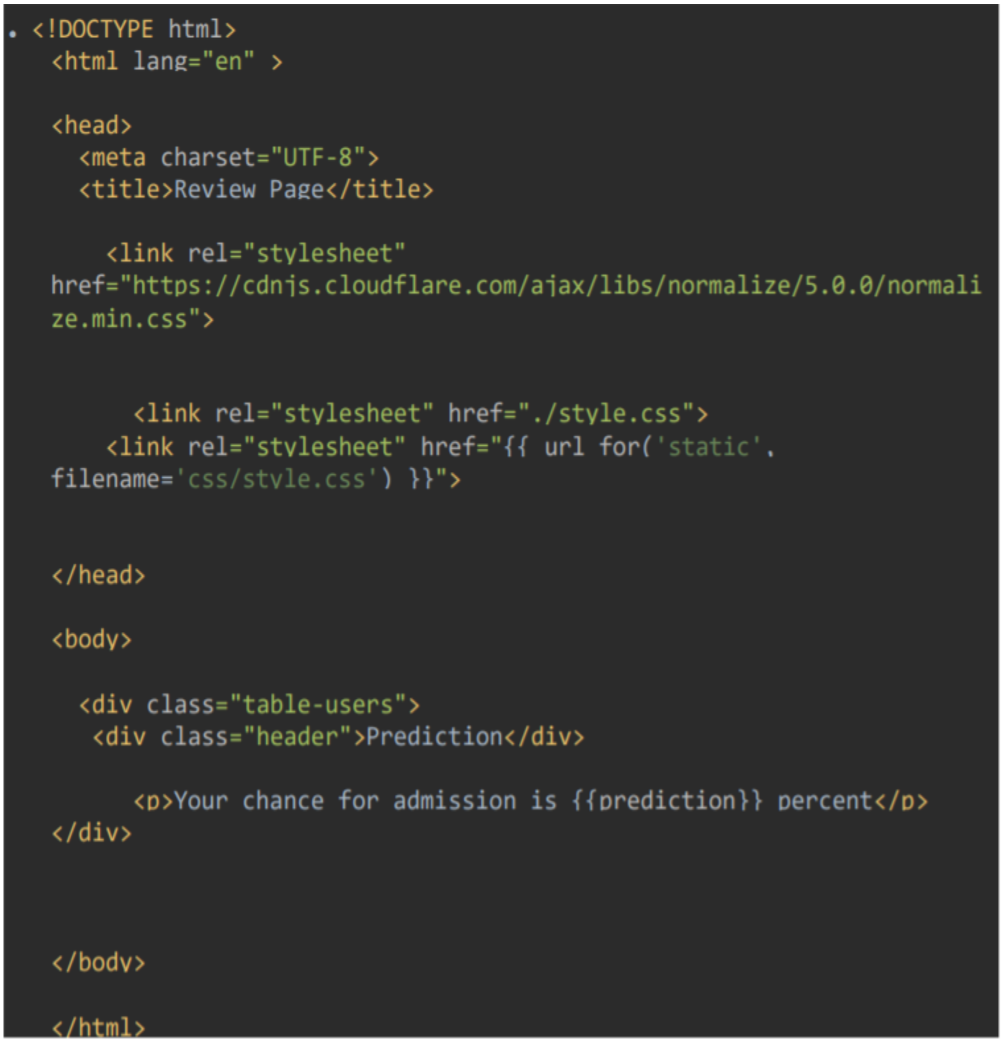
Now that our Flask app is ready, we needed an interface to interact with the user and get the data needed to perform predictions. For this, an HTML was used. Below is the HTML code written to generate the page.

Landing Page Source Code:



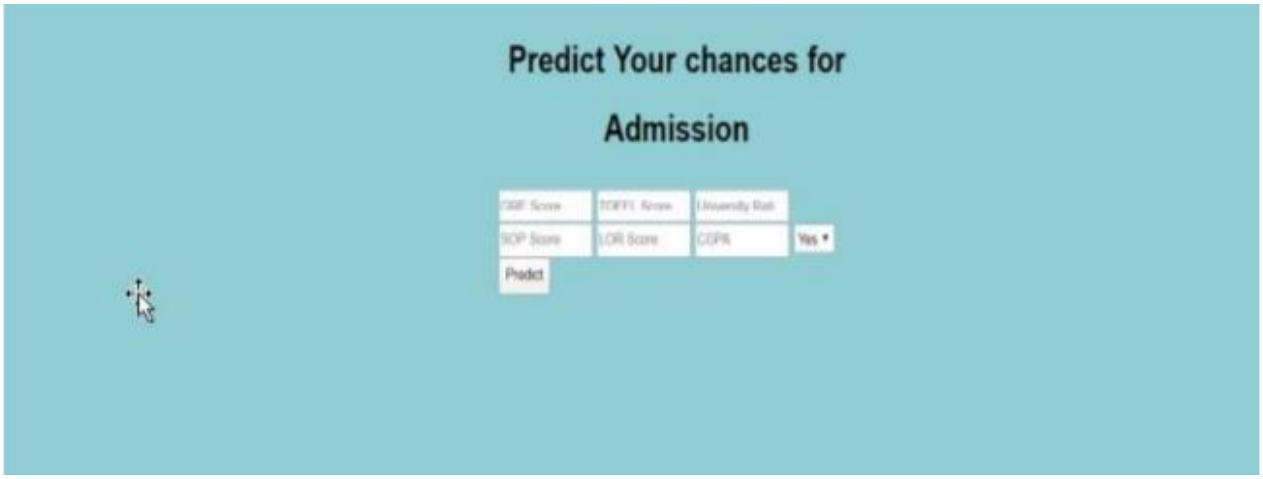
```
• {% extends 'base.html' %}

    {% block head %}

    <title>Search Page</title>
    <link rel="stylesheet" href="{{ url_for('static',
    filename='css/style.css') }}">
    {% endblock %}

    {% block body %}
    <div class="content">
        <h1 style="text-align: center">Predict Your chances for
    Admission</h1>

        <div class="form">
            <form action="/predict" method="POST">
                <input type="number" name="gre_score" id="gre_score"
    placeholder="GRE Score">
                <input type="number" name="toefl_score" id="toefl_score"
    placeholder="TOEFL Score">
                <input type="number" name="university_rating"
```

Output Page Source Code:



```
<!DOCTYPE html>
<html lang="en" >

<head>
  <meta charset="UTF-8">
  <title>Review Page</title>

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normali
ze.min.css">


      <link rel="stylesheet" href="./style.css">
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">


</head>

<body>

  <div class="table-users">
   <div class="header">Prediction</div>

      <p>Your chance for admission is {{prediction}} percent</p>
</div>



</body>

</html>
```

Output Page:



Predict Your chances for

Admission