Data Glacier

Your Deep Learning Partner

Model Deployment Flask

Predict acceptance chance of university admission

LISUM07 – Andersson Andreé Romero Deza

24-03-2022

Agenda

Background

Model

Flask API

Implementation

Background

In this project, a regression model is developed to predict the probability of being accepted for Graduate school.
- Data Source: https://www.kaggle.com/mohansacharya/graduate-admissions

- The dataset contains the following parameters:

- 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)

- Create Flask API.

Agenda

Background

Model

Flask API

Implementation

Model

```
admission_df.columns
```

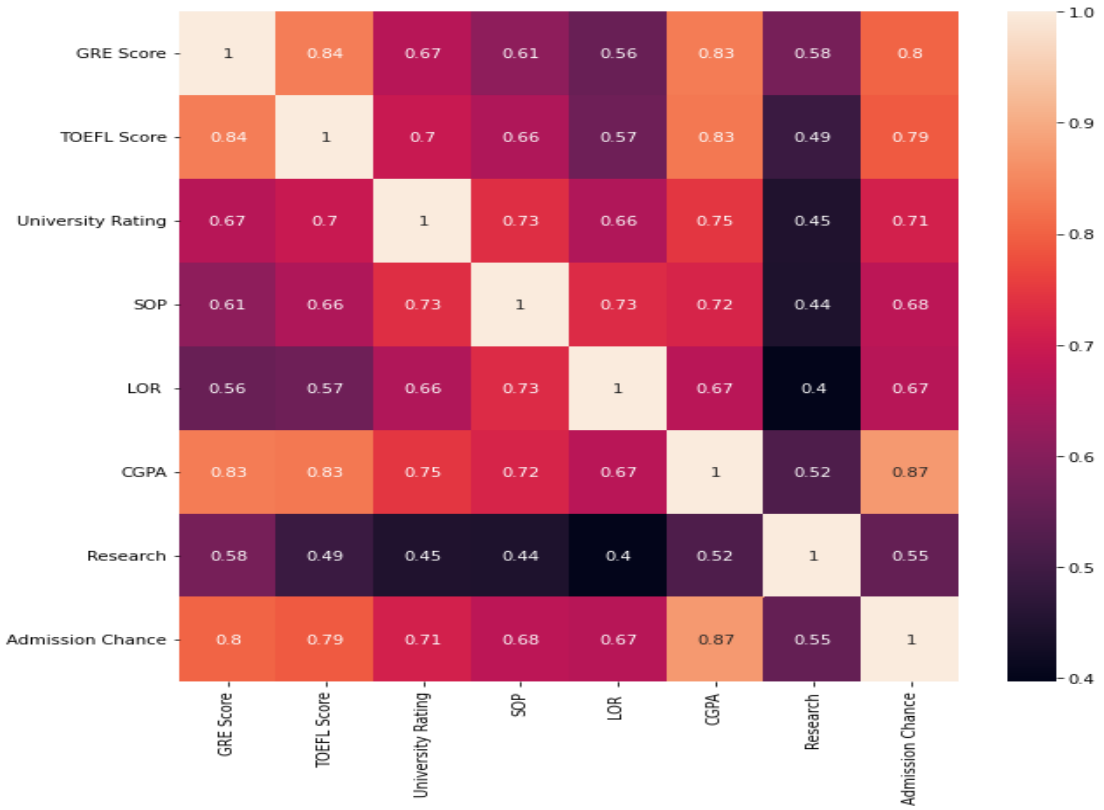
```
Index(['GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR ', 'CGPA',
       'Research', 'Admission Chance'],
      dtype='object')
```

```
X = admission_df[[ 'GRE Score', 'TOEFL Score','University Rating', 'CGPA']]
y = admission_df['Admission Chance']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression(fit_intercept = True)
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```



- A linear regression model was created.
- 'GRE Score', 'TOEFL Score', 'University Rating', 'CGPA' were selected due to their strong correlation.

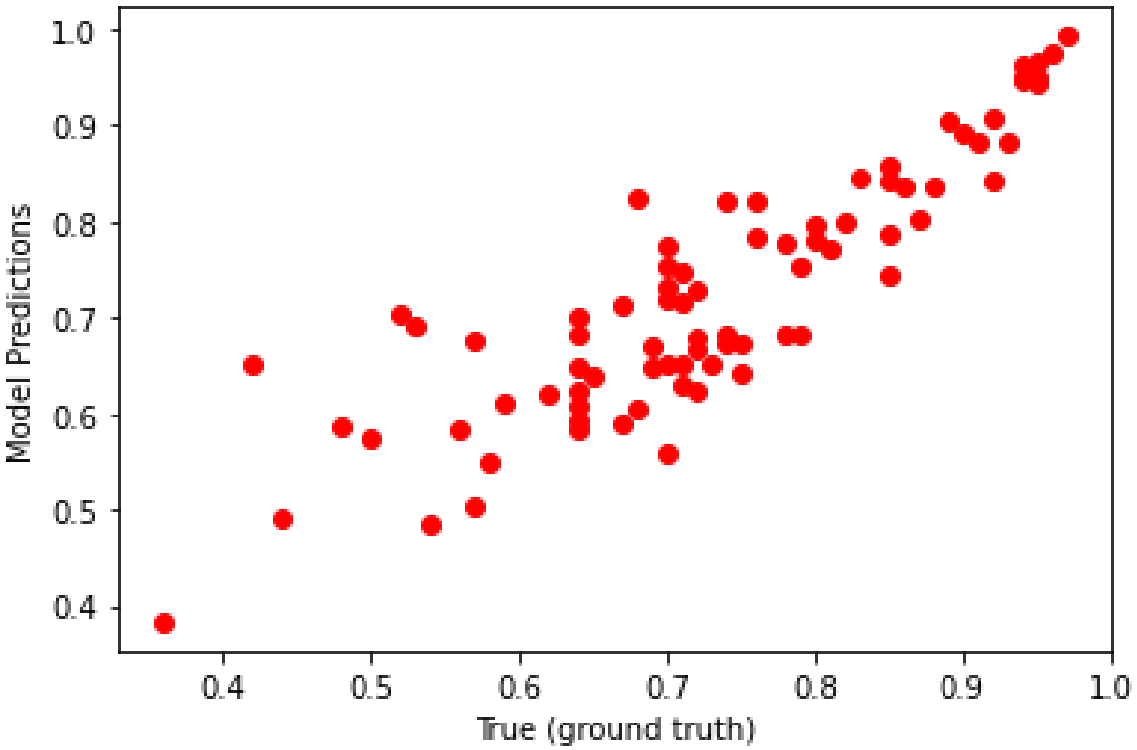
Model

```
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from math import sqrt

k = X_test.shape[1]
n = len(X_test)
RMSE = float(format(np.sqrt(mean_squared_error(y_test, y_predict)) , '.3f'))
MSE = mean_squared_error(y_test, y_predict)
MAE = mean_absolute_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)
adj_r2 = 1-(1-r2)*(n-1)/(n-k-1)
MAPE = np.mean( np.abs((y_test - y_predict) /y_test ) ) * 100

print('RMSE =',RMSE, '\nMSE =',MSE, '\nMAE =',MAE, '\nR2 =', r2,
      '\nAdjusted R2 =', adj_r2, '\nMean Absolute Percentage Error =', MAPE, '%')
```

```
RMSE = 0.067
MSE = 0.004502224607420019
MAE = 0.0506245957804314
R2 = 0.7638417284842514
Adjusted R2 = 0.7512466206700781
Mean Absolute Percentage Error = 7.771715973144319 %
```



- An R2 of 76% was obtained.
- Saved the model in a file 'finalized_model.pkl

```
import pickle
```

```
[ ] filename = 'finalized_model.pkl'
    pickle.dump(regressor, open(filename, 'wb'))
```

Agenda

Background

Model

Flask API

Implementation

Flask API - CODE

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

app = Flask(__name__)
model = pickle.load(open('finalized_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)
   if output < 0:
      return render_template('index.html',
         prediction_text = "Predicted is negative, not admitted")
   elif output >= 0:
      return render_template('index.html',
         prediction_text = 'Probability admission is: {}'.format(output))


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

- The loaded model is in pickle format.
- The prediction has an additional in case it is negative.

Flask API - HTML

```
<!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 University Admission</h1>
    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('predict')}}"method="post">
                                        <input type="text" name="GRE Scores" placeholder="GRE Scores (out of 340)"
required="required" />
   <input type="text" name="TOEFL Scores" placeholder="TOEFL Scores (out of 120)" required="required" />
                                        <input type="text" name="University Rating" placeholder="University Rating (out
of 5)" required="required" />
   <input type="text" name="CGPA" placeholder="Undergraduate GPA (out of 10)" required="required" />

      <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
   </form>
  <br>
  <br>
  {{ prediction_text }}
 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;"
alt="Company Logo"/>
</body>
</html>
```

- The HTML format used was provided by DATA GLACIER.

Agenda

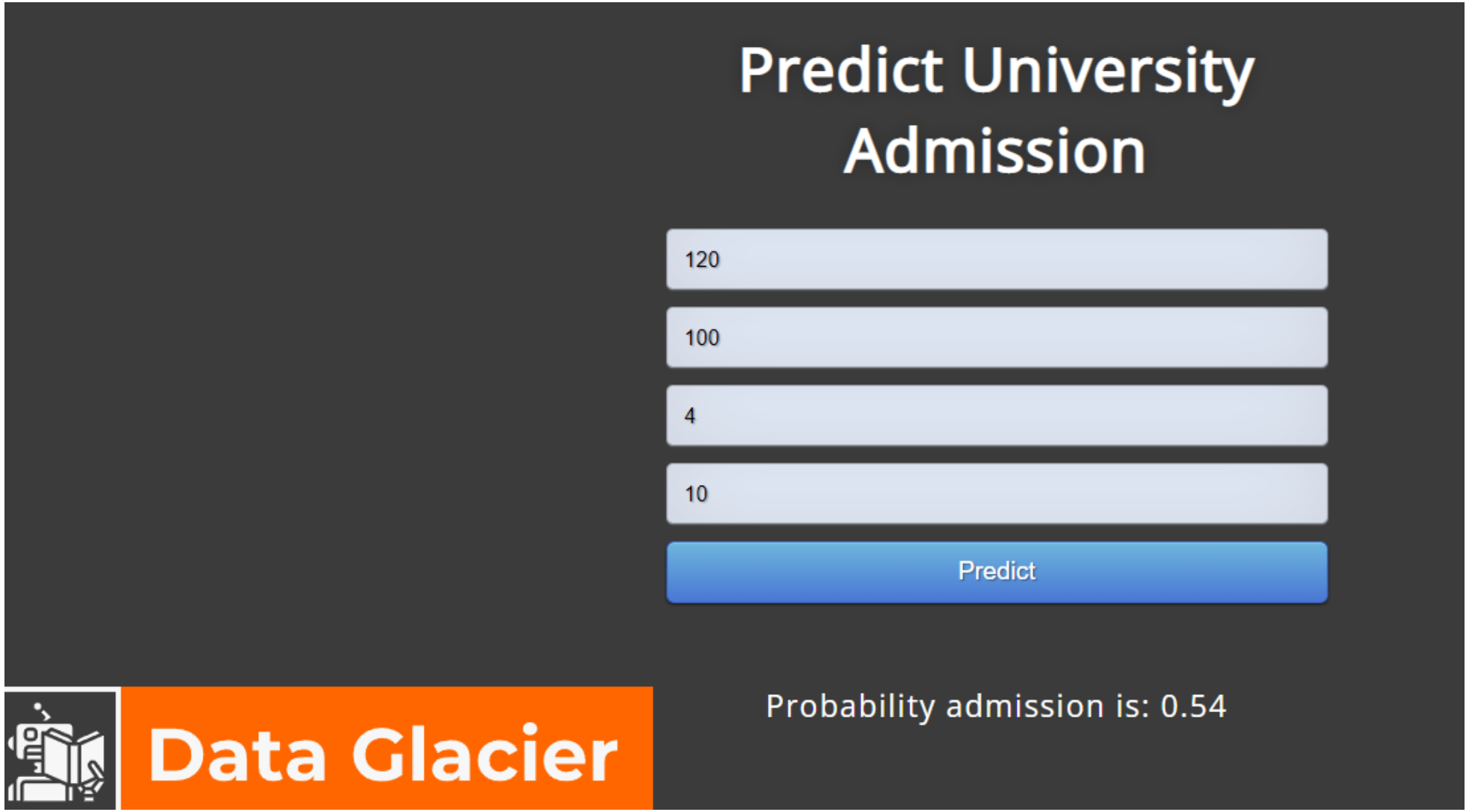
Background

Model

Flask API

Implementation

Implementation



- A web-app was developed where the machine learning model was implemented.

Data Glacier

Thank You