

University Admit Eligibility Predictor

V.MOHAN KUMAR

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

Category: Machine Learning

Project Description:

Introduction: The world markets are developing rapidly and continuously looking for the best knowledge and experience among people. Young workers who want to stand out in their jobs are always looking for higher degrees that can help them in improving their skills and knowledge. As a result, the number of students applying for graduate studies has increased in the last decade .so we are going to help student and university in predicting the possibility of accepting students submitting each year and provide the needed resources.

- Graduate Record Exam1 (GRE) score. The score will be out of 340 points.
- Test of English as a Foreigner Language2 (TOEFL) score, which will be out of 120 points.
- University Rating (Uni.Rating) that indicates the Bachelor University ranking among the other universities.
- Statement of purpose (SOP) which is a document written to show the candidate's life, ambitious and the motivations for the chosen degree/ university. The score will be out of 5 points.
- Letter of Recommendation Strength (LOR) which verifies the candidate professional experience, builds credibility, boosts confidence and ensures your competency. The score is out of 5 points
- Undergraduate GPA (CGPA) out of 10
- Research Experience that can support the application, such as publishing research papers in conferences, working as research assistant with university professor (either 0 or 1). One dependent variable can be predicted which is chance of admission, that is according to the input given will be ranging from 0 to 1.

Purpose of the Project:

Students are often worried about their chances of admission to University. The aim of this project is to help students in shortlisting universities with their profiles. The predicted

output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be

we are going to help student and university in predicting the possibility of accepting students submitting each year and provide the needed resources.

2. LITERATURE SURVEY

2.1 Existing Problem

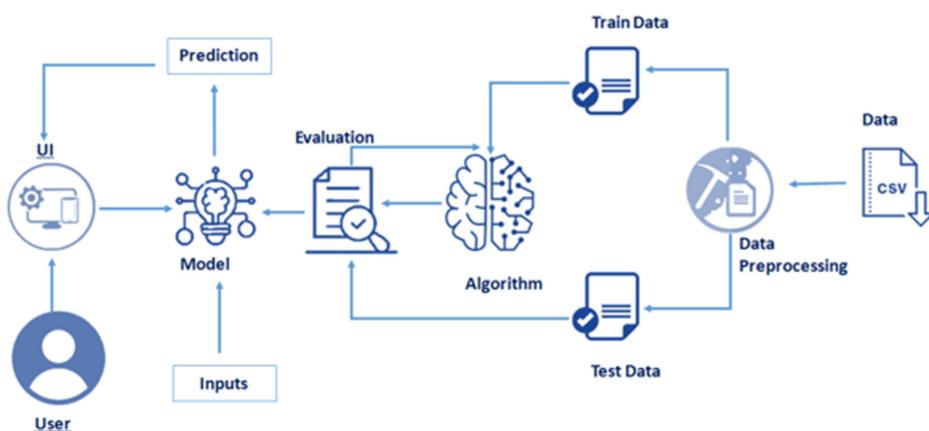
It is a time consuming process many students apply to many universities without even knowing whether they are eligible for the particular University or not. the application form also cost them students end up missing the good University in spite they think they will get admission in top university so they will not consider the lower rank University but the students are not eligible for the top University .as they try to get into a top university they will miss a good University's

2.2 Proposed system

This model will help them to predict whether they are eligible for admission by taking consideration of GRE score ,TOEFL score, University rating, SOP, LOR and CGPA each University has a set of scores to eligible a student for admission this model will help them to check whether they are eligible and they will get a idea of University that they are eligible

3. THEORETICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware/Software Designing

*Hardware requirements

- 2 GB ram or above
- Dual core processor or above
- Internet connection

*Software requirements

- Anaconda Navigator
- Python packages

4. EXPERIMENTAL INVESTIGATION

The main objective of this research is to use algorithms to identify eligibility of student In this study :

- Regression and Classification Problems
- To grab insights from data through visualization.
- Applying different algorithms according
- Evaluation metrics

5. FLOWCHART

- User interacts with the UI (User Interface) to enter Data
- The entered data is analyzed by the model which is integrated
- Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

- Data Collection.
- Collect the dataset or Create the dataset
- Data Pre-processing.
- Import the Libraries.
- Importing the dataset.
- Checking for Null Values.
- Data Visualization.
- Taking care of Missing Data.
- Label encoding.
- One Hot Encoding.
- Feature Scaling.
- Splitting Data into Train and Test.
- Model Building
- Training and testing the model
- Evaluation of Model

- Application Building
- Create an HTML file
- Build a Python Code

6.RESULT

UNIVERSITY ADMISSION PREDICTION SYSTEM

Enter your details and get probability of your admission

Enter GRE Score

Enter TOEFL Score

Select University no
 1
 2
 3
 4
 5

Enter SOP

Enter LOR

Enter CGPA

Research
 Research
 NO Research

RESULT

Predicting Chance of Admission

A Machine Learning Web App using Flask.

Prediction : [You have a chance of getting admission](#) 



7. ADVANTAGES AND DISADVANTAGES

Advantages :

Time saving process and gives student clarity of what are the universities they are eligible for. in spite they think they will get admission in top university so they will not consider the lower rank University but the students are not eligible for the top University .as they try to get into a top university they will miss a good University's . This give them increase in stress , anxiety . This model will help them to get a clarity

8.APPLICATION

University Admit Eligibility Predictor

9.CONCLUSION

In this paper, machine learning models were performed to predict the opportunity of a student to get admitted to a university program. The machine learning models included are Regression and Classification Problems ,To grab insights from data through visualization ,Applying different algorithms according , Evaluation metrics .As for the future work, more models can be conducted on more datasets to learn the model that gives the best performance.

10. FUTURE SCOPE

Today almost everybody above the age of 12 years has smartphones with them, and so we can incorporate these solutions into an android app or ios app. Also it can be incorporated into a multiple university website and these app and website will be highly beneficial for a large section of students

11. BIBLIOGRAPHY

dataset used for this project was obtained from Kaggle.

<https://www.kaggle.com/rishal005/admission-predict>

M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, “Multi-split Optimized Bagging Ensemble Model Selection for Multi-class Educational Data Mining,” Appl. Intell., vol. 50, pp. 4506 – 4528, 2020.

M. S. Acharya, A. Armaan, and A. S. Antony, “A Comparison of Regression Models for prediction of Graduate Admissions,” Kaggle, 2018.

University Admission prediction using Machine Learning Kruthika CS1*, Apeksha B2 , Chinmaya GR3 , Madhumathi JB4 , Veena MR5 1Department of CS&E.BIET, Davanagere. Karnataka, India

Moore, J. S. 1998. An expert system approach to graduate school admission decisions and academic performance prediction Omega 26(5): 659 – 670.

12. APPENDIX

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
data=pd.read_csv(r'C:\Users\Fazil\Downloads\archive\Admission_Predict.csv')
data.head()
data.tail()
data.drop(['Serial No.'],axis=1,inplace=True)
data.head()
data.describe()
data.info()
data=data.rename(columns = {'chance of admit' : 'chance of admit'})
data.corr()
plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True,cmap="RdYlGn")
data.isnull().any()
sns.pairplot(data=data,hue='Research')
sns.scatterplot(x= 'University Rating',y= 'CGPA',data=data,color='blue',s=100)
category = ['GRE Score','TOEFL Score','University Rating','SOP','LOR
','CGPA','Research','Chance of Admit ']
color = ['yellowgreen','gold','lightskyblue','pink','red','purple','orange','grey']
```

```

start = True

for i in np.arange(4):

    fig = plt.figure(figsize=(14,8))

    plt.subplot2grid((4,2), (i,0))

    data[category[2*i]].hist(color=color[2*i],bins=10)

    plt.title(category[2*i])

    plt.subplot2grid((4,2), (i,1))

    data[category[2*i+1]].hist(color=color[2*i+1],bins=1)

    plt.title(category[2*i+1])

    plt.subplots_adjust(hspace = 0.7, wspace = 0.2)

    plt.show()

print('Mean CGPA Score is :',int(data['CGPA'].mean()))

print('Mean GRE Score is :',int(data['GRE Score'].mean()))

print('Mean TOEFL Score is :',int(data['TOEFL Score'].mean()))

x=data.iloc[:,0:-1].values

x

y=data.iloc[:,7:].values

y

from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler()

x=sc.fit_transform(x)

x

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, random_state = 10)

import sklearn from sklearn.linear_model import LinearRegression

cls =LinearRegression()

lr=cls.fit(x_train, y_train)

y_pred =lr.predict(x_test)

```

```

y_pred

from sklearn.metrics import r2_score

print("\nAccuracy score: %f" %(r2_score(y_test,y_pred) * 100))

import pickle

pickle.dump(lr,open('university.pkl','wb'))

model=pickle.load(open('university.pkl','rb'))

import numpy as np

from flask import Flask, request, jsonify, render_template

import pickle

app = Flask(__name__)

model = pickle.load(open('university.pkl', 'rb'))

@app.route('/')

def home():

    return render_template('Demo2.html')

@app.route('/y_predict',methods=['POST'])

def y_predict():

    """
    """

```

For rendering results on HTML GUI

""

```

#min max scaling

min1=[290.0, 92.0, 1.0, 1.0, 1.0, 6.8, 0.0]

max1=[340.0, 120.0, 5.0, 5.0, 5.0, 9.92, 1.0]

k= [float(x) for x in request.form.values()]

p=[]

for i in range(7):

    l=(k[i]-min1[i])/(max1[i]-min1[i])

    p.append(l)

prediction = model.predict([p])

```

```
print(prediction)

output=prediction[0]

if(output==False):

    return render_template('noChance.html', prediction_text='You Dont have a chance of
getting admission')

else:

    return render_template('chance.html', prediction_text='You have a chance of getting
admission')

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