**INDEX**

**TITLE PAGENO**

**CHAPTER 1: INTRODUCTION 1**

* 1. Overview
  2. Purpose

**CHAPTER 2: LITERATURE SURVEY 2**

2.1 Existing problem

2.2 Proposed solution

**CHAPTER 3: THEORITICAL ANALYSIS 3**

3.1 Block Diagram

3.2 Hardware /Software designing

**CHAPTER 4: EXPERIMENTAL INVESTIGATIONS 10**

**CHAPTER 5: FLOWCHART 11**

**CHAPTER 6: RESULT 12**

**CHAPTER 7: ADVANTAGES AND DISADVANTAGES 14**

**CHAPTER 8: APPLICATIONS 15**

**CHAPTER 9: CONCLUSION 16**

**CHAPTER 10:FUTURE WORK 17**

**CHAPTER 11:BIBILOGRAPHY 18**

**APPENDIX 19**

A.Source code

**CHAPTER-1**

**INTRODUCTION**

* 1. **OVERVIEW:**

Generally as the students do not have much idea about the procedures, requirements and details of the universities, so they seek help from the education consultancy firms to help them successfully secure admission in the universities which are best suitable for their profile, for this they have to invest huge amount of money as consultancy fees. Apart from these the education consultancy firms there are few websites and blogs that guide the students on the admission procedures. The drawback of the currently available resources is that they are very limited and also they are not truly dependable taking into consideration of their accuracy and reliability. The aim of this research is to develop a system using machine learning algorithms.

* 1. **PURPOSE:**

The purpose of our project helps the students to identify the chances of their application to an university being accepted. Also it will help them in identifying the universities which are best suitable for their profile. 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 preparing to get a better idea.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 EXISTING PROBLEM:**

There are many projects and various studies dealing with university admission predictions but they do not give accurate results. Even though websites were there,they do not give the accurate results which causes ambiguity.

**2.2 PROPOSED SOLUTION:**

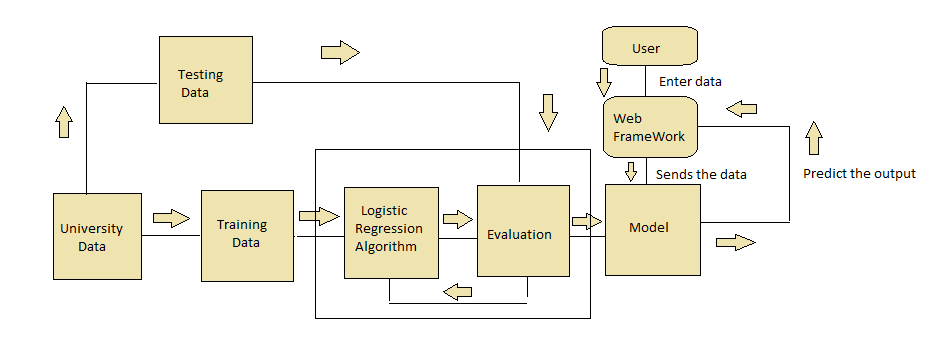
Our proposed work deals with the usage of machine learning algorithms that is logistic algorithm and developing an website which predicts admission of the student with the profile's they entered.The predicted output gives them a fair idea about their admission chances in a particular type of university.

**CHAPTER-3**

**THEORITICAL ANALYSIS**

**3.1 BLOCK DIAGRAM :**

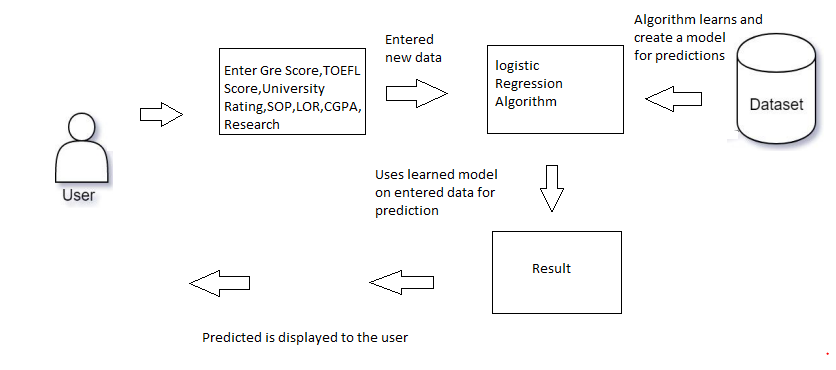
A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks.They are heavily used in engineering in hardware design, software design and process flow diagrams .Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation.



**HARDWARE/SOFTWARE DESIGNING:**

**3.2.1 SYSTEM ARCHITECTURE:**

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. Our system architecture comprises of categorical view of the whole processes taking place in the web page. At first the user visits our website and enters the required inputs namely GRE, TOEFL scores, University ratings, CGPA etc. The data is then sent to the logistic regression model which gives the highest accuracy. Then the output is displayed on the interface as the university allotted to the student based on the prediction using machine learning algorithms.

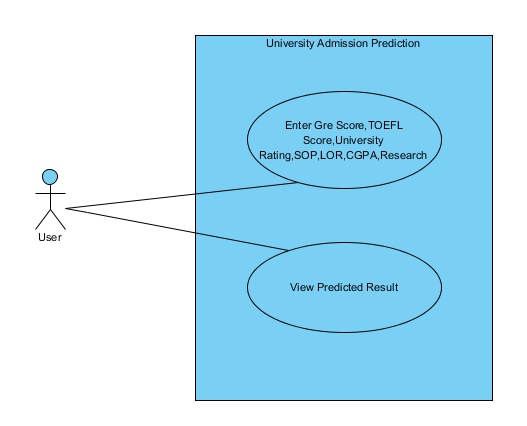


.

**3.2.1 USE CASE DIAGRAM:**

A use-case model describes a system's functional requirements in terms of use cases. It is a model of the system's intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs. It is a very powerful planning instrument, the use-case model is generally used in all phases of the development cycle by all team members. An effective use case diagram can help your team discuss and represent:

Scenarios in which your system or application interacts with people, organizations, or external systems goals that your system or application helps those entities (known as actors) achieve the scope of your system.

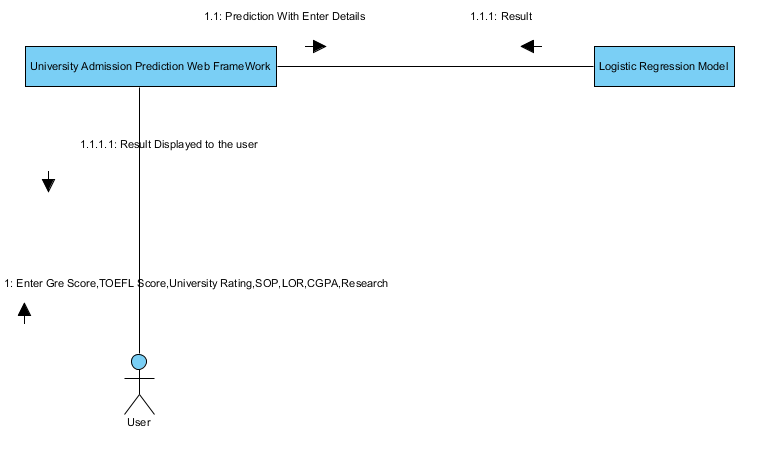


According to the use case diagram depicted above, user enters the GRE,TOEFL scores, he also enters the rating of the university he wants to join, also enters the SOP, LOR, CGPA and the required profiles. Here, we use the machine learning algorithms to classify the data and display the predicted results.

**3.2.2 COLLABORATION DIAGRAM:**

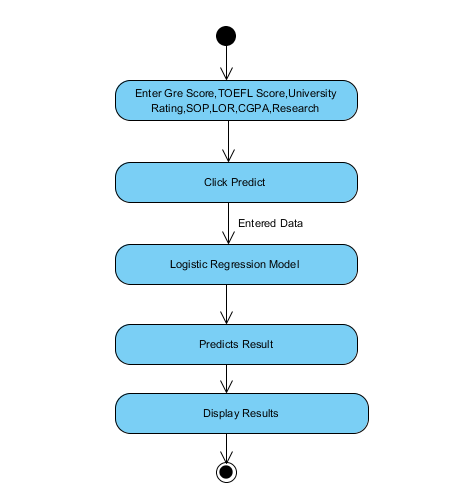
UML Collaboration diagrams (interaction diagrams) illustrate the relationship and interaction between software objects. They require use cases, system operation contracts, and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects (instances). A diagram is created for each system operation that relates to the current development cycle (iteration). The collaboration diagrams represent a combination of information taken from class, the sequence, use case diagrams describing both the static structure and dynamic behavior of a system.  
The below collaboration diagram represents a University Admission Prediction. The flow of communication in the above diagram is given by,  
1. A user opens the webpage and enters required details such as scores of GRE, TOEFL, ratings of universities, SOP, LOR, CGPA etc..  
2. The data is preprocessed and sent to the logistic regression model.

3. The output is predicted by using this model and result is displayed to the user.



**3.2.3 ACTIVITY DIAGRAM:**

In UML, an activity diagram is used to display the sequence of activities. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. Activity diagrams can be drawn in the initial stages of development to help both developers and clients to analyze business workflow processes and gain a shared understanding of what is going on in the system. At this stage they provide a useful vehicle for discussion, helping developers, clients and users to visualize the system functionality. The ability of activity diagrams to represent activities that can be carried out in parallel is particularly useful in high-level business modelling, as drawing the diagrams can help to identify potential for parallel processing, even where activities are currently carried out sequentially. Once the system use cases have been identified, activity diagrams can be used to illustrate the steps involved in achieving a use case goal, showing the activities and the order in which they take place. Finally, when development has reached a stage where classes have been identified together with their attributes and operations, activity diagrams are a useful means of describing how the operations work, particularly when these are based on complex algorithms. The activity diagram includes in UML specification consists following elements: activity, action, edge, branching and merging, forking and joining, initial and final nodes.

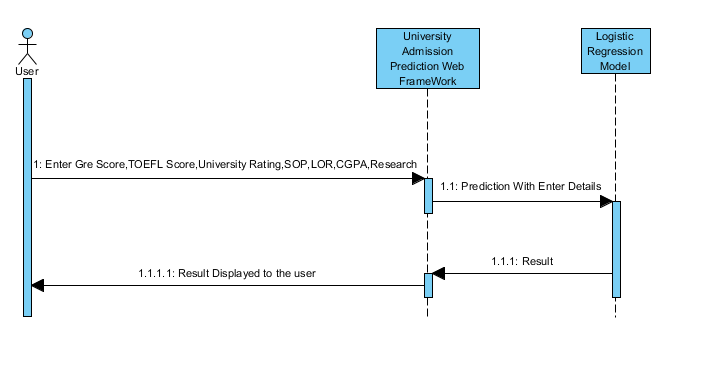


Our activity diagram University Admission Prediction can be evaluated as follows:

The first step consists of opening of an website, in which the user needs to enter the GRE,TOEFL scores, he also enters the rating of the university he wants to join, also enters the SOP, LOR, CGPA and the required profiles.. Then the user clicks on prediction ,In the further step the data is processed by the Logistic Regression machine learning model. As soon as the data is processed, the results are predicted with greater accuracy and required result is displayed to the user.

**3.2.4 SEQUENCE DIAGRAM:**

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram. Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications. The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top down, the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, the object instances that the messages are sent to.



The above sequence diagram specifies about the university admission prediction.From the diagram,when the user enters the GRE,TOEFL scores and required profiles . Then the data is pre-processed by using the logistic regression model and the process of prediction takes place. The predicted result is now in return sent to the user through the website.

**3.2.5. SOFTWARE DETAILS**

* Python
* Anaconda
* Html
* Css

**3.2.6 HARDWARE DETAILS**

* CPU : Intel i3 processor
* RAM : 4GB
* Operating system: Windows 7 or newer, 64-bit macOS 10.9+, or Linux. 
* System architecture: 64-bit x86, 32-bit x86 with Windows or Linux.

**CHAPTER-4**

**EXPERIMENTAL INVESTIGATIONS**

In an attempt to empirically validate the performance of our proposed model, we used the university prediction dataset in our work in order to layout the needed features, train our models and test their performance. In order to get more consistent and discipline results it is essential to pre-process the dataset. For this purpose, we implemented a number of common pre-processing techniques used in machine learning. In our experiment we performed Naïve Bayes, Logistic Regression, Random Forest and Decision tree algorithms. When we compared the accuracy results obtained from the respective algorithms, they are as follows:

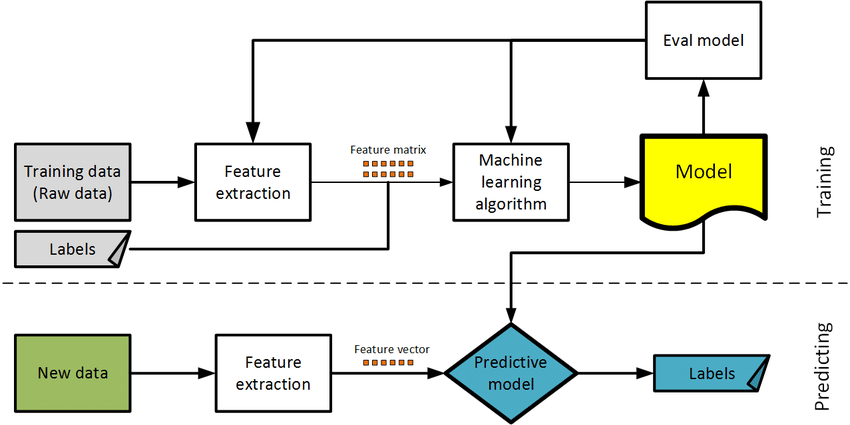
Naive bayes model gives the accuracy of 66% , Logistic Regression model gives the accuracy of 91%,Random forest gives the accuracy of 87%,Decision tree gives the accuracy of 77%.Out of all of these the best and highest accuracy is given by the Logistic regression model, so it is considered in predicting our results.

**CHAPTER-5**

**FLOWCHART**

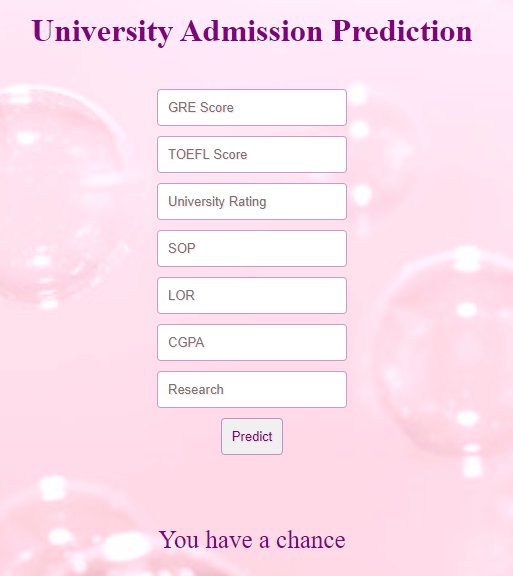
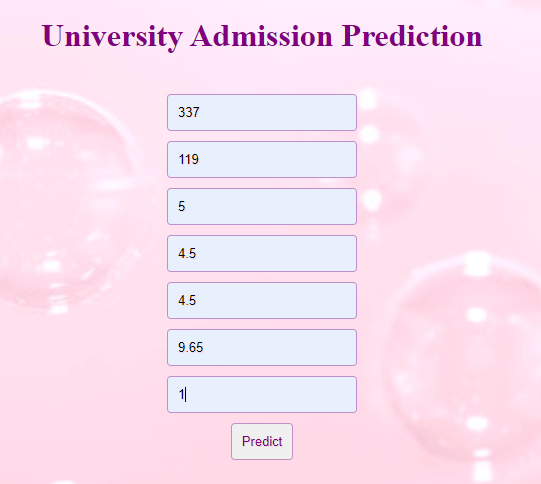
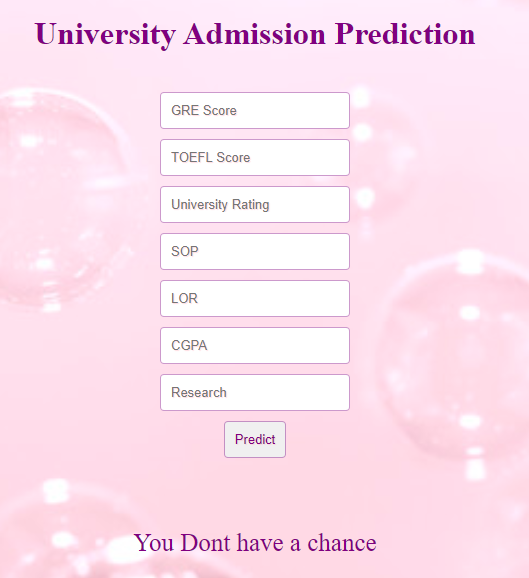
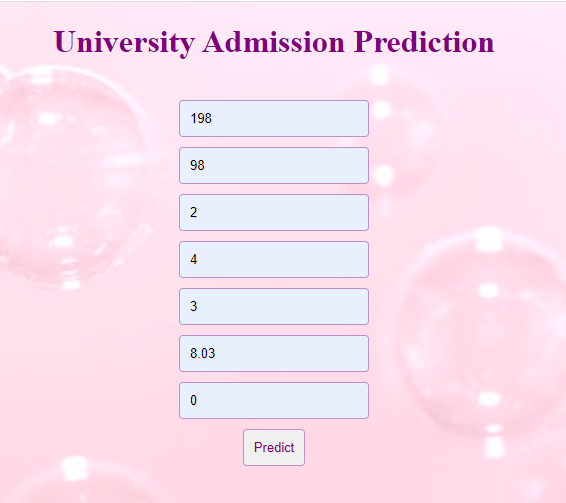
The flowchart for our university prediction algorithm using machine learning is given as:

1. **Gathering data**: The raw data is gathered
2. **Data pre-processing**: Here, the dataset is preprocessed by using the machine learning algorithms and some data visualization techniques.
3. **Researching the model that will be best for the type of data**: In this stage,we use many types of machine learning classification algorithms and use the one model which gives more accuracy.
4. **Training and testing the model:** All the features are extracted from the dataset and the data is further divided into training and testing data efficiently for best prediction results.
5. **Evaluation:** The trained and tested data is evaluated and the accuracy is calculated for the best performance of the model. In this way our prediction algorithm for university admission works.
6. **New data:** The new data is taken and given to the predictived model and it will return the predicted output and display to the user



**CHAPTER-6**

**RESULTS**



**CHAPTER 7**

**ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES:**

The advantages of our prediction algorithm can be included as follows:

1) **More Diverse Data**:  
Data is the fuel which drives Machine Learning. We can improve our analysis and predictions significantly by having more data.There are some new data sources like Yocket.in, Gradcafe.com, Facebook Groups, etc. The Data gathered from Facebook Groups is in structured form and can be used after some cleaning.

2**) Authenticity**:  
The data that I have used is self reported data. If input data contains any unused values they can be removed for authencity. If we can get authentic data, we can come up with more accurate predictions.

3**) Diversity: Demographic-Bias**  
We might look for data from more diverse sources like Chinese, German or Korean websites. This will help us see the big picture. It will also enable to compare the strengths, aspirations and priorities of International Students based on demography. We can compare the strength from GRE scores, TOEFL score, CGPA, etc. We can study the top priorities to shortlist a University like location, reputation, ranking, etc. by observing the popular Universities in each country. We can gauge the aspirations by comparing the Universities applied and the universities admitted.

**DISADVANTAGE:**

The disadvantages of the prediction algorithm are few and they can be listed as:

Our research has certain limitations. There was not an access to a dedicated student data set, and the study relies on public data sources. In addition, both data sets were small, having less than thousand records.. A research that has access to more comprehensive data may offer more conclusive results.

**CHAPTER-8**

**APPLICATIONS**

The applications include:

**Data Analysis**:

Here, in data analysis we find the missing data , redundancy in data, look for the unstructured and inconsistent data.

**Data Visualization**:

We used many data visualization and summarization techniques to understand each and every part of the data.

**Prediction of datasets**:

Many universities admissions are predicted by using the GRE, TOEFL scores which can be further used for university records.

**Future usage of data**:

Once predicted, the data can be used for future references for estimating statistics of the students joined in those universities.

**CHAPTER 9**

**CONCLUSION**

In this project, we predicted a student’s likelihood of receiving admission to an institution of their choice.We programmed the machine learning algorithms in the open source language Python, to analyze the data. We have taken the dataset, performed the preprocessing on data, data visualization and checked for any missing values, performed feature scaling, split the data into train and test and finally we built an model. We also created an website by using HTML and Python.

**CHAPTER 10**

**FUTURE SCOPE**

Another area that future research can improve is the variety of the machine learning methods. This research used Logistic Regression, Decision Trees, and the Naïve Bayes classification. Other methods, such as clustering and artificial neural networks can be used to have a better understanding of the importance of method selection.The other area that can be improved is the process of feature creation. Also, there is another future scope which includes collection of details of the user college details, gives the better prediction results.

**.CHAPTER 11**

**BIBILOGRAPHY**

[https://towardsdatascience.com/data-visualization-for-machine-learning-and-data-science-a45178970be7](https://towardsdatascience.com/data-visualization-for-machine-learning-and-data-science-a45178970be7" \t "https://mail.google.com/mail/u/0/" \l "inbox/_blank)

[https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf](https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf" \t "https://mail.google.com/mail/u/0/" \l "inbox/_blank)

[https://thesmartbridge.com/documents/spsaimldocs/Machinelearning.pdf](https://thesmartbridge.com/documents/spsaimldocs/Machinelearning.pdf" \t "https://mail.google.com/mail/u/0/" \l "inbox/_blank)

<https://scikit-learn.org/stable/>

[https://thesmartbridge.com/documents/spsaimldocs/FlaskML.pdf](https://thesmartbridge.com/documents/spsaimldocs/FlaskML.pdf" \t "https://mail.google.com/mail/u/0/" \l "inbox/_blank)

<https://www.ets.org/s/gre/pdf/concordance_information.pdf>

<https://www.ideals.illinois.edu/handle/2142/92866>

<https://www.ideals.illinois.edu/handle/2142/92866>

<https://studyabroad.careers360.com/articles/gre-eligibility-criteria>

**APPENDIX**

**Source code**

**UniversityAdmissionPrediction.py**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df=pd.read\_csv('Admission\_Predict\_Ver1.1.csv')

df.drop(labels='Serial No.',inplace=True,axis=1) #Dropped S.No column because it has no purpose for us

df.apply(lambda x: sum(x.isnull()),axis=0)

df.apply(lambda x: sum(x==0),axis=0)

x=df.iloc[:,:7].values

y=df.iloc[:,7:8].values

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=0)

from sklearn.preprocessing import MinMaxScaler

sc= MinMaxScaler()

x\_train =sc.fit\_transform(x\_train)

x\_test = sc.fit\_transform(x\_test)

y\_train=(y\_train>0.5)

y\_test=(y\_test>0.5)

from sklearn.linear\_model.logistic import LogisticRegression

cls1 =LogisticRegression(random\_state =0)

lr=cls1.fit(x\_train, y\_train)

y\_pred =lr.predict(x\_test)

y\_pred

import pickle

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

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

**app.py**

import numpy as np

from flask import Flask, request, jsonify, render\_template

import pickle

app = Flask(\_\_name\_\_)

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

@app.route('/')

def home():

return render\_template('index.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('index.html', prediction\_text='You Dont have a chance')

else:

return render\_template('index.html', prediction\_text='You have a chance')

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

**style.css**

body

{

background-image:url('a.jpg');

background-repeat:no-repeat;

background-size:cover;

background-position:center;

height:100%;

text-align:center;

}

h1

{

color:purple;

}

input,button {

margin-bottom: 10px;

border: none;

outline: none;

padding: 10px;

font-size: 13px;

color: purple;

text-shadow: 1px 1px 1px rgba(255,182,193,0.5);

border: 1px solid rgba(128,0,128,0.4);

border-radius: 4px;

}

p{

color:purple;

font-size:25px;

}

**index.html**

<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

<meta charset="UTF-8">

<title>University Admission Prediction</title>

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

<style>

.login{

top: 20%;

}

</style>

</head>

<body>

<div class="login">

<h1>University Admission Prediction</h1>

<br>

<!-- Main Input For Receiving Query to our ML -->

<form action="{{ url\_for('y\_predict')}}"method="post">

<input type="text" name="GRE Score" placeholder="GRE Score" required="required" /><br>

<input type="text" name="TOEFL Score" placeholder="TOEFL Score" required="required" /><br>

<input type="text" name="University Rating" placeholder="University Rating" required="required" /><br>

<input type="text" name="SOP" placeholder="SOP" required="required" /><br>

<input type="text" name="LOR" placeholder="LOR" required="required" /><br>

<input type="text" name="CGPA" placeholder="CGPA" required="required" /><br>

<input type="text" name="Research" placeholder="Research" required="required" /><br>

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

</form>

<br>

<br>

<p>

{{ prediction\_text }}

<p>

</div>

</body>

</html>