A

Project Report

On

IIIT ADMISSION PREDICTION USING MACHINE LEARNING

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, KADAPA

in partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

K.BHARGAVI	R171209
P.MOKSHITHA	R171195
P.PRASANNA LAKSHMI	R170502

Under the Guidance of

Mr.N.MOHANRAJU

Assistant Professor

Department Of Electronics and Communication Engineering

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

RGUKT, RK VALLEY

(RGUKT KADAPA is approved by UGC, AICTE, established in 2008, provide Education opportunities for rural people) Vempalli, Kadapa-516330

2019-2023

RGUKT, RK VALLEY

(RGUKT KADAPA is approved by UGC, AICTE, established in 2008, provide Education opportunities for rural people) Vempalli, Kadapa-516330

2019-2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "IIIT ADMISSION PREDICTION USING MACHINE LEARNING" a bonafide record of the project work done and submitted by

K.BHARGAVI	**R171209**
P.MOKSHITHA	**R171195**
P.PRASANNA LAKSHMI	**R170502**

for the partial fulfilment of the requirements for the award of B.Tech Degree in **ELECTRONICS AND COMMUNICATION ENGINEERING,** RGUKT , Kadapa.

GUIDE	**Head of the Department**
Mr.N.MohanRaju	M. Madhan Mohan
Assistant Professor	Assistant Professor,
	Head of Department of ECE

External Viva-Voice Exam Held on _____

INTERNAL EXAMINER *EXTERNAL EXAMINER*

DECLARATION

We hereby declare that the project report entitled “IIIT ADMISSION PREDICTION UDING MACHINE LEARNING” submitted to the Department of ELECTRONICS AND COMMUNICATION ENGINEERING in partial fulfilment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY**. This project is the result of our own effort and that it has not been submitted to any other University or Institution for the award of any degree or diploma other than specified above.

K.BHARGAVI	**R171209**
P.MOKSHITHA	**R171195**
P.PRASANNA LAKSHMI	**R170502**

ACKNOWLEDGEMENT

We are thankful to our guide Mr N.MOHANRAJU for his valuable guidance and encouragement. His helping attitude and suggestions have helped us in the successful completion of the project.

We would like to express our gratefulness and sincere thanks to **Prof. B. Madhan Mohan**, Head of the Department of ELECTRONICS AND COMMUNICATION ENGINEERING, for his kind help and encouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing our hearty thanks to our beloved Director **Prof. K. Sandhya Rani** for spending his valuable time with us to complete this project.

Successful completion of any project cannot be done without proper support and encouragement. We sincerely thanks to the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

K.BHARGAVI	**R171209**
P.MOKSHITHA	**R171195**
P.PRASANNA LAKSHMI	**R170502**

ABSTRACT

IIIT admission Prediction is very important and plays a key role in Andhra Pradesh students who just passed 10^{th} class. Predicting IIIT admission can be especially difficult because the students are not aware of admission requirements. For that reason, the main purpose of this research work is to provide a recommender system for early predicting IIIT admission.

Therefore, To predict chance of admission first we applied several Supervised Machine Learning algorithms namely Linear Regression, Support Vector Regression, Decision Tree Regression, and Random Forest Regression. Secondly we compared and evaluated algorithms used to create a predictive model based on various evaluation metrics. Lastly we determined the most important parameters that influence the chance of admission.

The experimental results showed that the Linear Regression is the most suitable Machine Learning algorithm for predicting IIIT admission

TABLES OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTML	Hyper Text Markup Language
CSS	Cascaded Style Sheets
ML	Machine Learning
MSR	Mean Squared Error
LR	Linear Regression
JS	Java Script

CHAPTER -1 INTRODUCTION

1.1 INTRODUCTION

The number of students applying for IIIT(RGUKT) has increasing year by year. This fact has motivated us to study the grades of students and the chance of admission for IIIT that can help students in predicting the possibility of getting admission in IIIT.

Machine Learning is a subset of Artificial Intelligence (AI) that enables computers to automatically improve through experience. In the area of education, the adoption of Machine Learning is also accelerating.

Using the machine learning algorithms it's easy to build the prediction model based on the previous year's admission data.

1.2 PURPOSE

The purpose of this document is to gather the requirements that are needed for implementing the IIIT ADMISSION PREDICTION. It also focuses on various key features, the product, product vision and scope, product overview. The main purpose of this model is to provide a Prediction system that predicts the percentage of chance of IIIT admission.

1.3 INTENDED AUDIENCE

The intended audience will be the **students** who are interested to join IIIT. They can know the percentage of chance of getting admission in IIIT.

1.4 PRODUCT VISION

Vision Statement:

The product vision is to develop a Prediction System which is user friendly and accurate enough to predict the percentage of chance of admission of a student into IIIT. By using this system students can early predict the admission.

1.5 TECHNOLOGIES

- Machine Learning for Prediction System
- HTML, CSS for UI
- Flask for backend

CHAPTER -2

2. LITERATURE SURVEY

2.1 REGRESSION

As we are predicting the percentage of chance of admission, the output (Dependent) variable is continuous variable. We have to use the **Regression** techniques in **Supervised Algorithms** to predict the continuous variable.

Types of Regression Algorithms used for Prediction:

Linear Regression (LR): It is the most important algorithm in the field of Machine Learning, especially supervised learning. It is a way to model a relationship between a dependent variable and one or more independent variables. It consists of finding a regression line straight line through the points.

- **Random Forest Regression (RFR):** it is an ensemble learning method that constructs a multitude of decision trees at training time and uses the average prediction of the individual trees to improve the prediction
- **Support Vector Regression (SVR)**: It is also a very popular Machine Learning technique used in both classification and regression. It is similar to Linear Regression with only a few minor differences. SVR allows defining how much error is acceptable in our predictive model and will find an appropriate line to fit the data
- **Decision Tree (DT)**: It is the most widely used classification and prediction technique. It is a tree structure, where each internal node with outgoing edges indicates a condition on an attribute, each branch is an outcome of the test, and each leaf terminal node represents a class label

2.2 EVALUATION METRICS

The main part of building a Machine Learning model is Evaluation of model. There are many methods of evaluation that can be used. While building our model we have used the below two metrics for the evaluation part.

R-Squared (R^2 or the coefficient of determination) :

R-squared is a statistical method that determines the goodness of fit.

It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.

The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.

It is also called a **coefficient of determination,** or **coefficient of multiple determination** for multiple regression.

It can be calculated from the below formula:

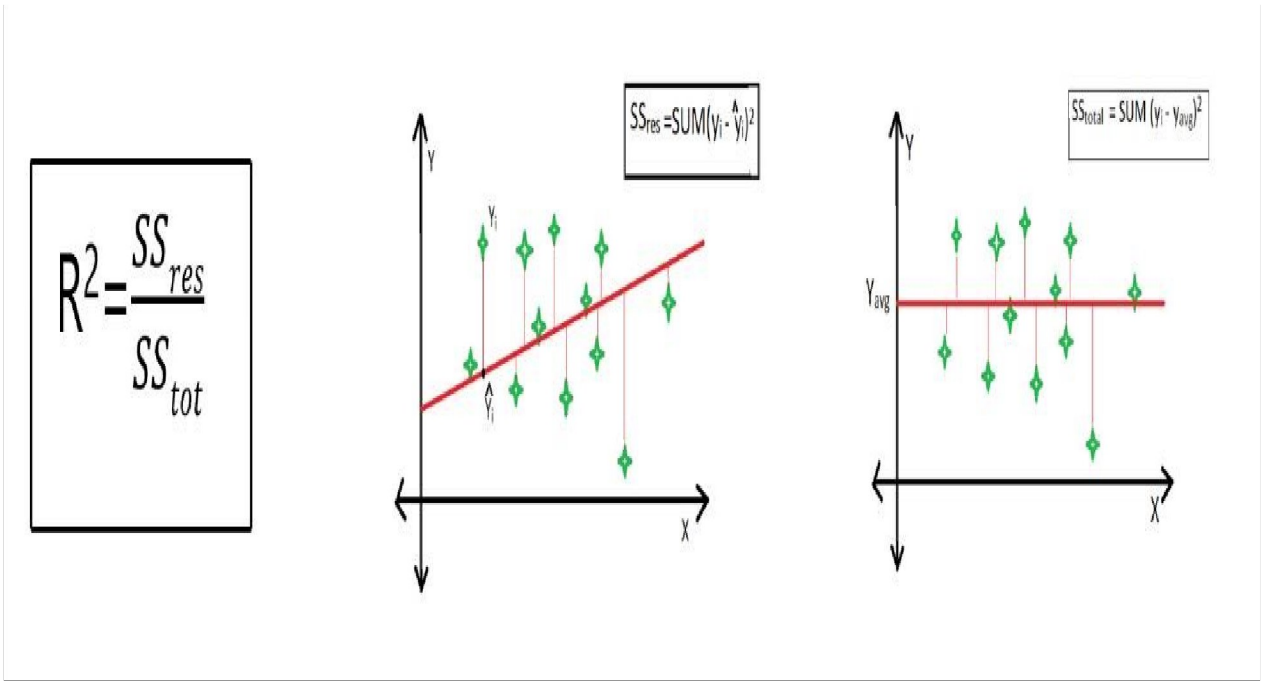

Fig 2.21 R squared coefficient of determination

Where,

SS_{res} = summation of squares of perpendicular distance between data points and the bestfitted line.

SS_{tot} = summation of squares of perpendicular distance between data points and the average lines.

Mean Square Error (MSE):

It is the arithmetic mean of the squares of the predictions between the model and the observations. This is the value to be minimized in the context of a single or multiple regressions. It measures the average of error squares i.e. the average squared difference between the estimated values and true values. It is a risk function, corresponding to the expected value of the squared error loss.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

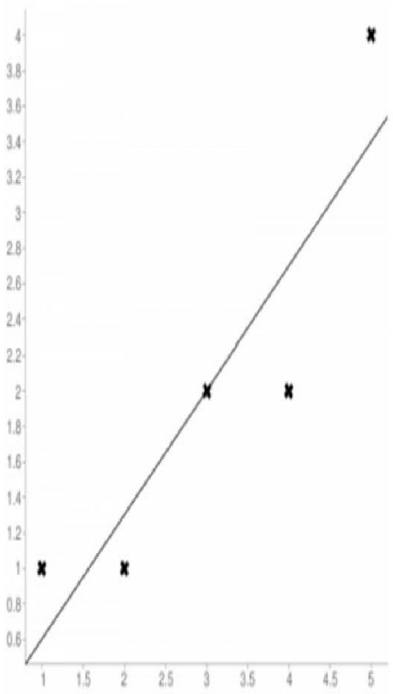

Fig.2.22 Mean square Error

n = Total no. of data points

Y_i = Actual Output Value

$\hat{Y}_i$ = Predicted Output Value

2.3 INITIAL RESULT OF ALGORITHMS

ALGORITHM	R2 (R SQUARED)	MEAN SQAURE ERROR
Support Vector Regression	0.8513	0.0792
Linear Regression	0.9417	0.0559
Random Forest Regression	0.9125	0.0492
Decision Tree Regression	0.8913	0.0743

Fig.2.33 Initial Result of Algorithm

From table, we can see that the Linear Regression suits best for our model.

CHAPTER -3

LINEAR REGRESSION

Linear regression is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model. When there is a single input variable (x), the method is referred to as **simple linear regression**. When there are **multiple input variables**, literature from statistics often refers to the method as multiple linear regression

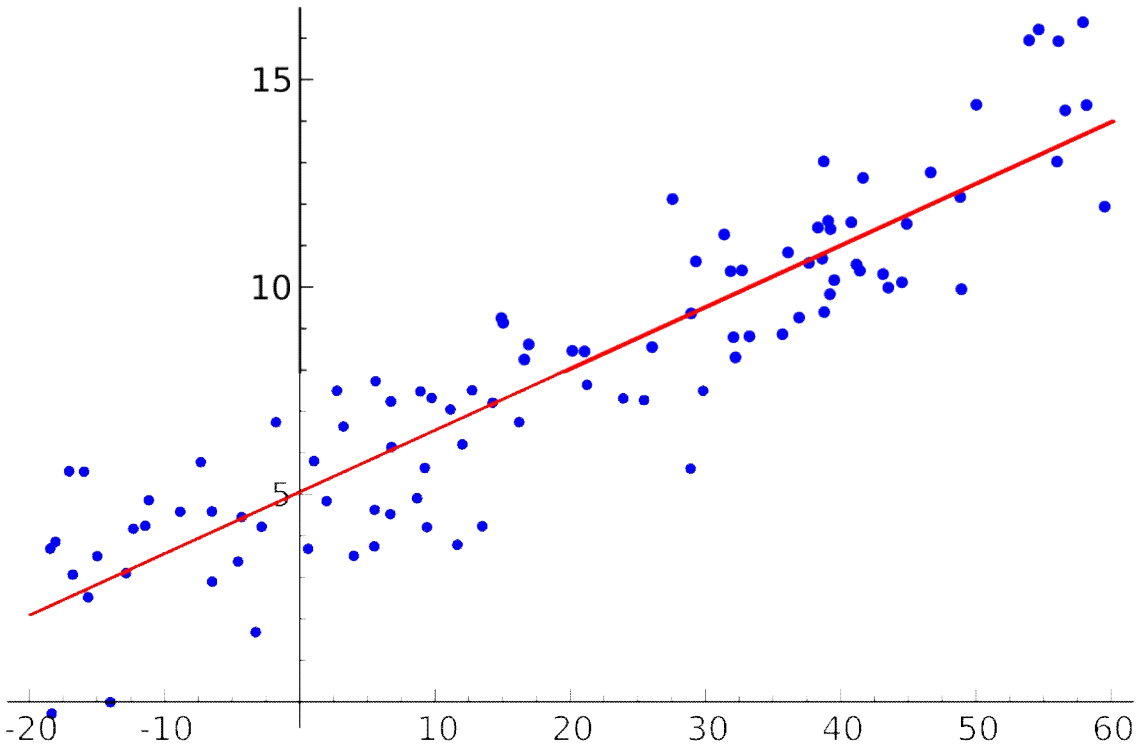

Fig.3 Linear Regression

3.1 HYPOTHESIS FUNCTION FOR LINEAR REGRESSION

$$y = \theta_1 + \theta_2.x$$

Fig.3.11 Hypothesis Function for Linear Regression

Where,

While training the model we are given :
X:input training data (one input variable) **y:**
labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x.

The model gets the best regression fit line by finding the best $\theta_1$and $\theta_2$values.

θ_1:intercept

θ_2:coefficient of x

Once we find the best θ_1 and $\theta_2$values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

3.2 FINDING THE BEST FIT LINE:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

3.3 COST FUNCTION:

Cost function is the calculation of error between predicted values and actual values, represented as a single real number.

- The different values for weights or coefficient of lines (θ_1, θ_2) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
- Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
- We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**

$$\text{Cost Function (J)} = \frac{1}{n}\sum_{i=0}^{n}(h_\theta(x^i) - y^i)^2$$

Fig.3.3 Cost Function

CHAPTER-4

IMPLEMENTATION

4.1 PROPOSED SYSTEM

- The students who completed tenth class have more curiosity and confusion whether they get admission in IIIT or not. At present there is no system available to help these students.
- So in this model we have proposed 'IIIT Admission Prediction System' which predicts the percentage of chance of getting admission into IIIT.

4.2 WORKING MODEL

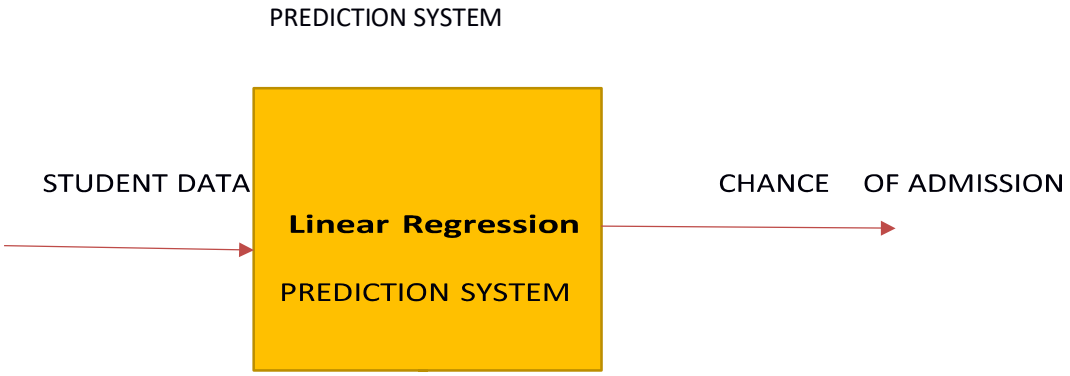

Fig 4.2 Working Model

This Proposed System is built in following steps

1. Data Collection
2. Data Pre-processing
3. Model Building

I used Jupyter notebook tool for this model building.

4.21 DATA COLLECTION

Initially I have collected the previous data related to admissions of IIIT and stored it in CSV file.

After importing proper modules access the dataset using read_csv method.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
d=pd.read_csv("//home//gangadhar//Downloads//admissions_data.csv")
```

```
df=pd.DataFrame(d)
```

```
df.head()
```

	Hall ticket (10th class)	10th GPA	Govt/Private School	Caste	Gender	Mandal	District	State	Chance of Admit
0	1720104115	10.0	Private	BC-B	Male	Kallur	Kurnool	Ap	0.83
1	1722136839	10.0	Govt school	BC-B	Male	Galiveedu	Kadapa	Ap	0.98
2	1718124014	9.7	Govt school	BC-B	Female	Madanapalli	Chittoor	Ap	0.93
3	1722136791	9.8	Govt school	SC	Male	Galiveedu	Kadapa	Ap	0.95
4	1722107642	9.8	Govt school	OC	Female	Vallur	Kadapa	Ap	0.90

4.22 DATA FEATURES

Given below table shows the data type of each attribute in our dataset

S.No	Attribute Name	Type
1	Hall Ticket Number	Quantitative
2	CGPA	Quantitative
3	Type of School	Qualitative
4	Caste	Qualitative
5	Gender	Qualitative
6	Mandal	Qualitative
7	District	Qualitative
8	State	Qualitative

PROPERTIES OF DATASET

```
df.describe()
```

	10th GPA	Govt/Private School	Caste	Gender	Chance of Admit
count	412.000000	412.000000	412.000000	412.000000	412.000000
mean	9.115777	0.349515	0.389320	0.525631	0.672549
std	0.925454	0.132997	0.227399	0.158126	0.327260
min	6.800000	0.000000	0.100000	0.340000	0.090000
25%	8.400000	0.400000	0.100000	0.340000	0.300000
50%	9.700000	0.400000	0.500000	0.660000	0.870000
75%	9.800000	0.400000	0.500000	0.660000	0.930000
max	10.000000	0.400000	0.800000	0.660000	0.990000

4.3 DATA PRE-PROCESSING

It represents one of the most crucial steps in all Machine Learning projects because it involves formatting data, Improving data quality, feature engineering, and labelling .

Firstly we need to process the data such that there will be no null values and duplicate values.

In the given data set there are so many parameters that are of type String. We need to convert this type into numeric values for further processing.

After doing this, add a new parameter called ‘Performance Score’ by adding caste, CGPA, school type and gender.

```
# Replacing Government schools with 0.4 and Private schools with 0 values

df['Govt/Private School']=df['Govt/Private School'].replace(['Govt school','Private'],[0.4,0])
```

```
# Replacing all castes with numerical values

df['Caste']=df['Caste'].replace(['OC','BC-A','BC-C','BC-D','BC-B','BC-E','SC','ST'],[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.
```

```
# Replacing Gender with numerical values

df['Gender']=df['Gender'].replace(['Male','Female'],[0.34,0.66])
```

```
df.head()
```

	Hall ticket (10th class)	10th GPA	Govt/Private School	Caste	Gender	Mandal	District	State	Chance of Admit
0	1720104115	10.0	0.0	0.5	0.34	Kallur	Kurnool	Ap	0.83
1	1722136839	10.0	0.4	0.5	0.34	Galiveedu	Kadapa	Ap	0.98
2	1718124014	9.7	0.4	0.5	0.66	Madanapalli	Chittoor	Ap	0.93
3	1722136791	9.8	0.4	0.7	0.34	Galiveedu	Kadapa	Ap	0.95
4	1722107642	9.8	0.4	0.1	0.66	Vallur	Kadapa	Ap	0.90

4.31 CALCULATING PERFORMANCE SCORE

Performance score is calculated to plot against chance of admit.

```
#Calculating performance score with major factors which affects chance of admit

df["Performance Score"] = df["Govt/Private School"] + df["Caste"] + df["Gender"] + df["10th GPA"]
```

```
df.head()
```

	Hall ticket (10th class)	10th GPA	Govt/Private School	Caste	Gender	Mandal	District	State	Chance of Admit	Performance Score
0	1720104115	10.0	0.0	0.5	0.34	Kallur	Kurnool	Ap	0.83	10.84
1	1722136839	10.0	0.4	0.5	0.34	Galiveedu	Kadapa	Ap	0.98	11.24
2	1718124014	9.7	0.4	0.5	0.66	Madanapalli	Chittoor	Ap	0.93	11.26
3	1722136791	9.8	0.4	0.7	0.34	Galiveedu	Kadapa	Ap	0.95	11.24
4	1722107642	9.8	0.4	0.1	0.66	Vallur	Kadapa	Ap	0.90	10.96

Finding Relationship between independent variables and dependent variable

```
#Relation between each parameters

df.corr()
```

	10th GPA	Govt/Private School	Caste	Gender	Chance of Admit	Performance Score
10th GPA	1.000000	-0.000630	-0.026136	0.007073	0.971466	0.950184
Govt/Private School	-0.000630	1.000000	0.011091	0.135724	0.097548	0.161434
Caste	-0.026136	0.011091	1.000000	-0.133113	0.008671	0.189551
Gender	0.007073	0.135724	-0.133113	1.000000	0.010696	0.157361
Chance of Admit	0.971466	0.097548	0.008671	0.010696	1.000000	0.945165
Performance Score	0.950184	0.161434	0.189551	0.157361	0.945165	1.000000

From the relationship table we can see that CGPA and Performance Score is highly related on Chance of Admit. But Performance Score is depended on cgpa,caste,school type and gender.

Now plot the relation between Performance Score and Chance of Admit.

4.32 GRAPH

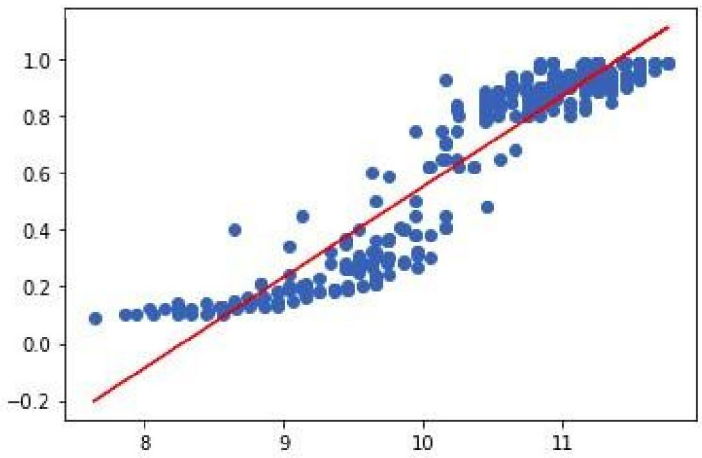

Fig. 4.32 Graph

4.33 MODEL BUILDING

As the final dataset is ready, now we can build our model.

Import the Linear Regression model from sklearn module.Divide the dataset into two parts such as Test data and Train Data.

Now train the test data using train data by fitting into Linear Regression Model.

```
from sklearn.linear_model import LinearRegression
```

```
LR=LinearRegression()
```

```
from sklearn.model_selection import train_test_split
```

```
x=df.loc[:,["10th GPA","Govt/Private School","Caste","Gender"]] # consedering 4 columns that chance of a
```

```
y=df["Chance of Admit"]
```

```
#Dividing dataset into train and test data

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=45)
```

```
#Fitting the data into Linear Regression Model

LR.fit(x_train,y_train)
y_pred=LR.predict(x_test)
```

Fig.

Now the predection model built sucessfully.

4.4 EVALUATION METRIC

Calculating R^2 score and Mean Square Error of the Model

```
#Calculating accuracy and error of model for this project

from sklearn.metrics import mean_absolute_error,r2_score
print("R2 score of the model is ",r2_score(y_pred,y_test))
print("mean_absolute_error  of the model is ",mean_absolute_error(y_pred,y_test))
```

```
R2 score of the model is  0.9417064974981846
mean_absolute_error  of the model is  0.05599397966866196
```

Fig.

We can see that our built model got **R2 Score** of **0.9417** which is quite good.**Mean Absolute Error** is **0.055.** Our prediction model is ready and now we can use it.

CHAPTER -5

5.SOURCE CODE

5.1 model.py

```
import pandas as pd
import numpy as np
import pickle
d=pd.read_csv("/Users//jayakrishna//Desktop//project//admission.csv")
df=pd.DataFrame(d)
df['Govt/Private School']=df['Govt/Private School'].replace(['Govt school','Private'],[0.4,0])
df['Caste']=df['Caste'].replace(['OC','BC-A','BC-C','BC-D','BC-B','BC-E','SC','ST'],
[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8])

df['Gender']=df['Gender'].replace(['Male','Female'],[0.34,0.66])
df ["Performance Score"] = df["Govt/Private School"] + df["Caste"] + df["Gender"] + df["10th GPA"]

from sklearn.linear_model import LinearRegression LR=LinearRegression()

from sklearn.model_selection import
train_test_spli t
x=df.loc[:,["10th GPA","Govt/Private School","Caste","Gender"]] y=df["Chance
of Admit"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=4
5) lr = LR.fit(x_train,y_train) pickle.dump(lr, open('admi.pkl', 'wb'))
```

5.2 app.py

```
from flask import Flask, render_template, request import
pickle import numpy as np model =
pickle.load(open('admi.pkl','rb')) app = Flask(__name__)
schoolDict={"Govt":0.4,"Private":0.0}
casteDict={"OC":0.1,"BC-A":0.2,"BC-B":0.3,"BC-D":0.4,
                    "BC-E":0.5,"SC":0.6,"ST":0.7}
genderDict={"Male":0.34,"Female":0.66}
```

```
@app.route('/')
def man():
 return render_template('home.html')
@app.route('/predict' , methods=['POST'])
def home():
   # getting details from html form
gpa = request.form['gpa']  gender =
request.form['gender']  school_type
= request.form['school']  caste =
request.form['caste']
arr = np.array([[float(gpa),schoolDict[school_type],casteDict[caste],
            genderDict[gender]]])

 pred = model.predict(arr)
 return
render_template('result.html',data=int(float(pred)*100)) if
__name__ == "__main__":
 app.run(debug=True)
```

5.3 home.html

```
<!DOCTYPE html>
<html>
<head>
<title>Registration Form</title>
<link rel="stylesheet" type="text/css" href="static/reg.css">
</head>
<body>
<div class="main">
<div class="register">
<h2 id="title">IIIT ADMISSION PREDICTION</h2>
 <form id="register" method="POST" , action="{{url_for('home')}}">
 <label>Name : </label>
 <br>
 <input type="text" name="name" id="name" placeholder="Enter Your Name" >
```

```
<br><br>
<label>Hall Ticket Number : </label>
 <br>
 <input type="text" name="ht" id="Hallticket" placeholder="Enter Your Hallticket number" >
 <br><br>
<label>CGPA : </label>
<br>
<input type="text" name="gpa" id="cgpa" placeholder="Enter 10th GPA" >
<br><br>
<label>Gender : </label>
<br>
    
<input type="radio" name="gender" id="male" value="Male">
 
<span id="male">Male</span>
     
<input type="radio" name="gender" id="female" value="Female">
  
 <span id="female">Female</span>
 <br><br>
<label>School Type : </label>
<br>
<select name="school" id="school_type">
<option value="Govt">Govt</option>
<option value="Private">Private</option>
</select>
<br><br>
<label>Caste : </label>
<br>
<select name="caste" id="Caste">
<option value="OC">OC</option>
<option value="BC-A">BC-A</option>
<option value="BC-B">BC-B</option>
```

```
<option value="BC-C">BC-C</option>
<option value="BC-D">BC-D</option>
<option value="BC-E">BC-E</option>
<option value="SC">SC</option>
<option value="ST">ST</option>
</select>
</form>
</div>
</div>
</body>
</html>
```

5.4 result.html

```
<!DOCTYPE html>
<html>
<head>
<title>Result</title>
<link rel="stylesheet" type="text/css" href="static/res.css">
</head>
<body>
<div class="result">
<center>
{%if data<0%}
<h1>Sorry</h1>
<h1>Unfortunately You Won't get Admission</h1>
{%else%}
{%if data>70%}
<h1>Hurray</h1>
{%endif%}
<h1>You have {{ data }}% Chance to get Admission </h2>
{%endif%}
<a href='/'>Go back to Home</a>
</center>
</div>
```

```
</body>
</html>
```

6.RESULT

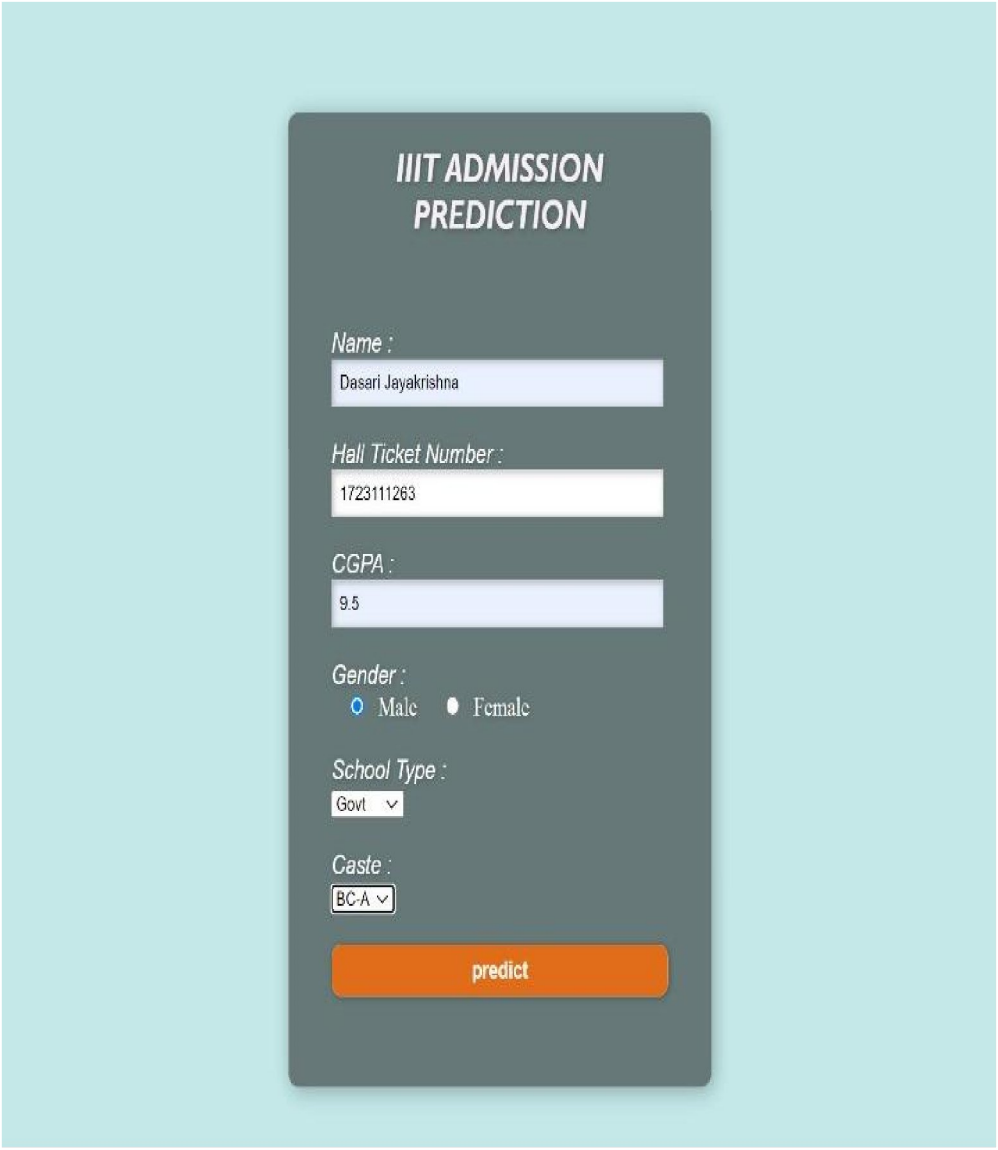

Fig. 6.11 Result

Here we are entering the details of a student and cgpa according to the 10^{th} standard.

By selecting the Gender, School Type and Caste it will predict the chance of admission based on our previous data which was stored in the csv file. BY calculating the performance score , divide it with the highest probability chance, we will get the percentage of chance to admit in the college.

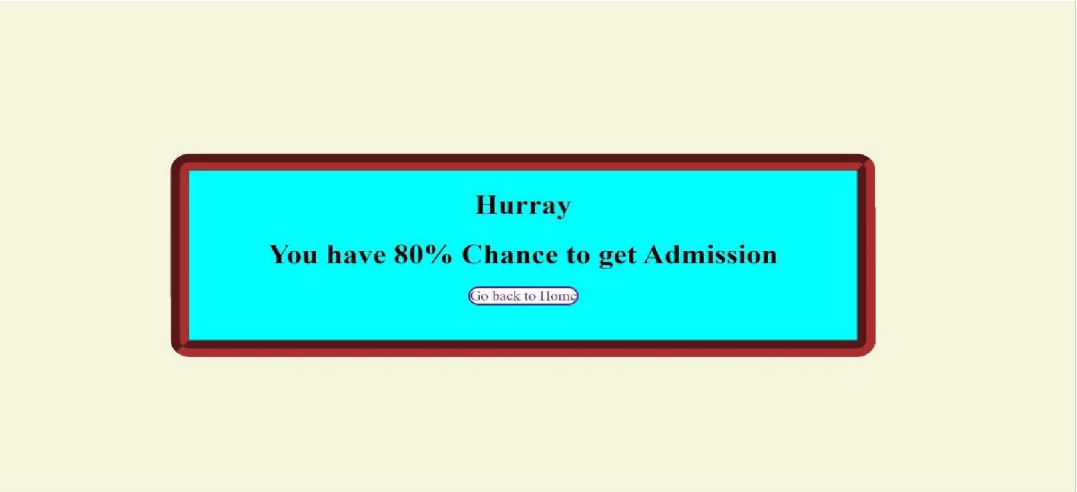

Fig.6.12 Result

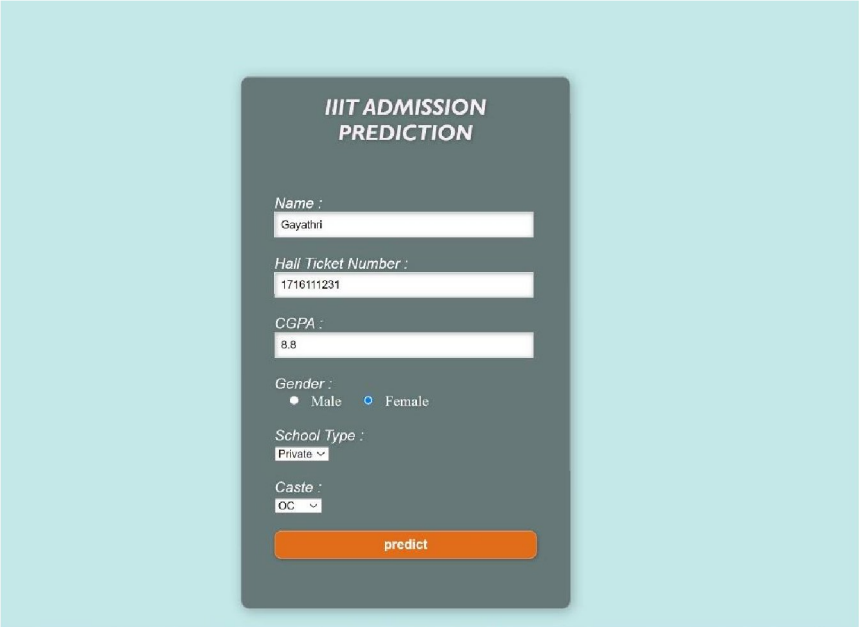

Fig. 6.13 Result

This is also the sample data given by the user to confirm the seat in IIIT, based on the CGPA, School Type, Gender, Caste. By clicking the predict button we will go into the new webpage which gives us the result.

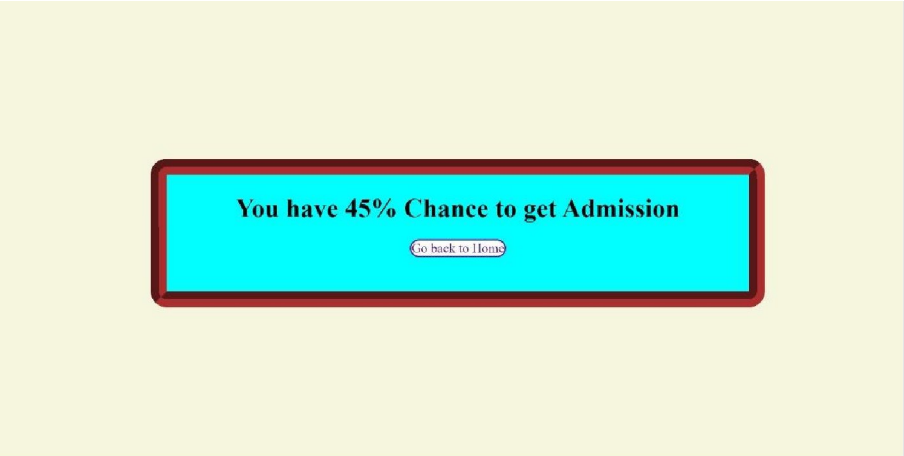

Fig.6.14 Result

6.2 CONCLUSION

In this project we used Linear Regression algorithm to provide percentage of chance of getting admission in IIIT to the students. Here we used previous year's date to train the model and testing.

Students may not have clear idea about selection process and attributes which effects admission more. Whenever student enter his details, the model gives the prediction based on similar previous data. So that students will know their chance and don't waste their time in waiting for the results. Students who have less chance they can proceed with other alternatives.

6.3 FUTURE ENHANCEMENTS

We will be considering option of special candidates like NCC,CAP,PH etc. There will be more priority for these special candidates, so automatically chance of admission for these candidates increases.We will also try to improve the interface more interactive i.e. queries section, help section etc.

6.4 REFERENCE

[1] GeeksForGeeks : https://www.geeksforgeeks.org/ml-linearregression/

[2] Scikit Learn : https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[3] **Javatpoint : https://www.javatpoint.com/machinelearning**