UNIVERSITY ADMISSION PREDICTION

Using Linear Regression Algorithm

Developed by: Andhe Madhuri, Haripriya Mydam, Bharath Kanukuntla, Arun Nagapuri
Smart Bridge-Remote Summer Internship Program

1. INTRODUCTION

Machine learning is a branch in computer science that studies the design of algorithms that can learn. Typical machine learning tasks are concept learning, function learning or predictive modeling, clustering and finding predictive patterns. These tasks are learned through available data that were observed through experiences or instructions. For example, Machine learning hopes that including the experience into its tasks will eventually improve the learning. The ultimate goal is to improve the learning in such a way that it becomes automatic, so that humans like ourselves don't need to interfere any more. Machine learning is a type of artificial intelligence in which the focus is on the development of computer programs that can learn from data and adapt to changes. There are many types of machine learning algorithms; they can be classified based on the desired output of the algorithm or the type of input available for training. As Big Data is the hottest trend in the tech industry at the moment, machine learning is incredibly powerful to make predictions or calculated suggestions based on large amounts of data. Some of the most common examples of machine learning are Netflix's algorithms to make movie suggestions based on movies you have watched in the past or Amazon's algorithms that recommend books based on books you have bought before. Learning Machine is a machine (a software, a web site, a mobile app, a robot) that performs a task, and that gets better and better as it performs it. In recent years, some learning machines made headlines. In 2016, AlphaGo won a match against one of the top Go players. AlphaGo is also a learning machine. It was first trained on a large set of recorded Go games between top players. Then it trained against itself. As it trained, its performance at Go increased, until it became better than a top Human player. First of all, It should be built a model of that data using a machine learning algorithm that can predict. The quality of a model boils down to the quality of the predictions we can make using it. It should be able to get good predictions through a good algorithm and the right model. It needs a machine learning algorithm that creates a model from data. It needs to deploy the model in a production environment where new data are ingested. Effect of each prediction is monitored and the feedback is ingested at once, leading to a new training, before the next prediction occurs.

There are huge number of phases in the prediction based on Machine Learning, and this prediction problem used most of them, Data collection is the first phase, by this phase data should be collected not usually a less data set, it should be huge data set according to the requirements one should collect or create the data for the prediction. Data Preprocessing is the second phase and this contain a lot of sub-phases for the processing of the data, it includes importing libraries, Data Visualization, Data Transformation, Feature Scaling, Splitting and Label Encoding. Data Splitting, in this phase the data is to be split into two as train_data and test_data for the training of the model. Then the Fourth phase is Model Training, Supervised learning allows for processing data with target attributes or labeled data.

These attributes are mapped in historical data before the training begins. And the last phase is Model evaluation and Testing and it is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That's the optimization of model parameters to achieve an algorithm's best performance.

Using the Machine Learning Algorithms we need to predict the admission chances in a particular university . Based on the student profiles and qualifications we need to predict whether they have chances to get the admission in an university. This blog will give them an overview of their admissions.

1.1 Overview

Students are often worried about their chances of admission in University. We need to create a model 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 preparing to get a better idea. Data Mining is one of the most motivating and vital area of research with the aim of extracting information from tremendous amount of accumulated data sets. The model has been built using data from student profiles to predict the chances of admissions. Three algorithms have been used to build the proposed model: Linear regression, RandomForest, Decision Tree. By using the algorithm a Flask model has been implemented and tested. The results has been discussed and a full comparison between algorithms was conducted. Linear Regression was selected as best algorithm based on accuracy.

1.2 Purpose

In this project, we make use of pandas, numpy, matplotlib, and seaborn libraries using the open-source, object-oriented programming language python 3.0 and packages such as scikit-learn to predict the chances of admission.

And in the end, to predict whether the student can get the admission or not using the predictions from multiple machine learning algorithms and withdrawing the conclusions.

2. LITERATURE SURVEY

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data. Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc. Regression is a data mining technique used to predict a range of numeric values (also called continuous values), given a particular dataset. For example, regression might be used to predict the future actions. The university admission prediction is particularly well suited to Regression Algorithm. In Regression , the independent variables are well trained to predict the dependent variable. And the model is evaluated.

2.1 Existing Problem

The previous models have less accuracy and the predictions are not accuarate whereas this model is constrained with the lot of advantages and with a higher accuracy than any other model already proposed. In this model we used Machine learning algorithm named Linear Regression which give an accuracy 80% of the predicted problem and there is an user friendly user interface to check the chances of admission for the students who are preparing for the admissions in universities.

2.2 Proposed Solution

Machine Learning (Linear Regression):

Linear Regression is one of the simplest and most common supervised machine learning algorithms that data scientists use for predictive modeling. We'll use Linear regression to build a model that predicts the chances of admissionsof students in an university. And also we have created an UI using the Flask for the admission prediction and this UI will allow the users to predict their chances of admission very easily and the User interface is user friendly not at least one complication in using the interface, and it can be used just by entering some necessary details into the UI in real time it'll give the predicted value like the chances of admission of students whether they get the admission or not.

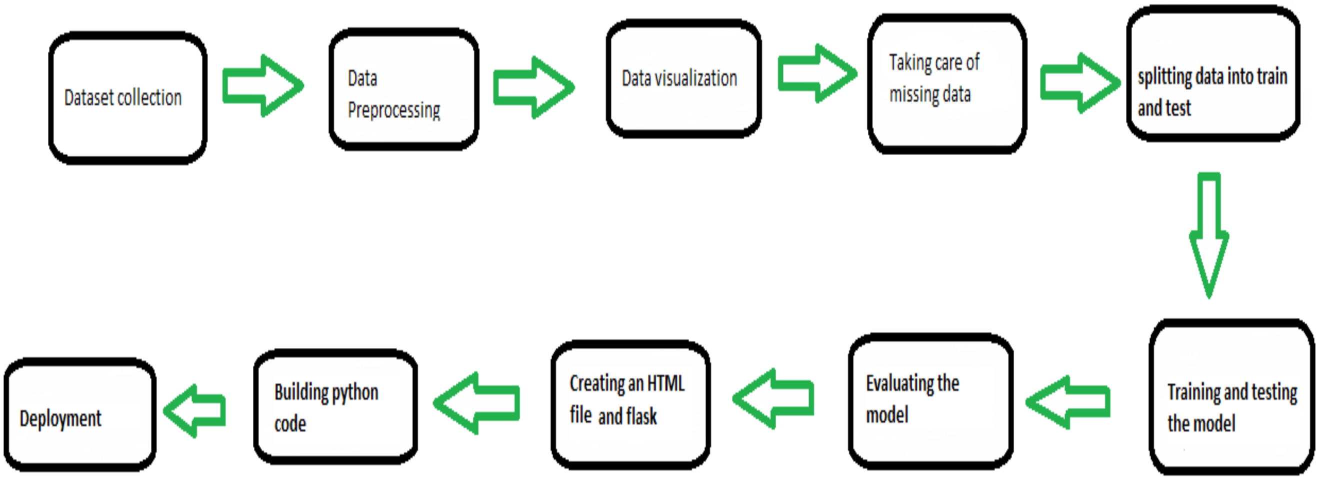
3. THEORETICAL ANALYSIS

While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is Linear Regression , it gives the output based on the independent variables accurately.

The peculiarity of this problem is collecting the student profiles real time and working with the prediction at the same time, so we developed an user interface for the students who'll be accesssing for the university admission prediction. There are several ways to check the Linear Regression model accuracy. Usually we use Root Mean Squared Error.We train Linear regression model by adding or removing the features to dataset, and see which one has the lowest RME- is the best one.

At first we got like lot of worst accuracies because we tried lot of algorithms for the best accurate algorithm , finally after all of that we tried the best suitable algorithm which gives the prediction accurately is Linear Regression Algorithm. And developed it to use as a real time prediction probelm for the admission prediction.

3.1 Block Diagram


Dataset collection
Data Preprocessing
Data visualization
Taking care of missing data
splitting data into train and test
Training and testing the model
Evaluating the model
Creating an HTML file and flask
Building python code
Deployment

3.2 Software Designing

● Jupyter Notebook Environment and Google Colab

● Spyder Ide

● Machine Learning Algorithms

● Python (pandas, numpy, matplotlib, seaborn, sklearn)

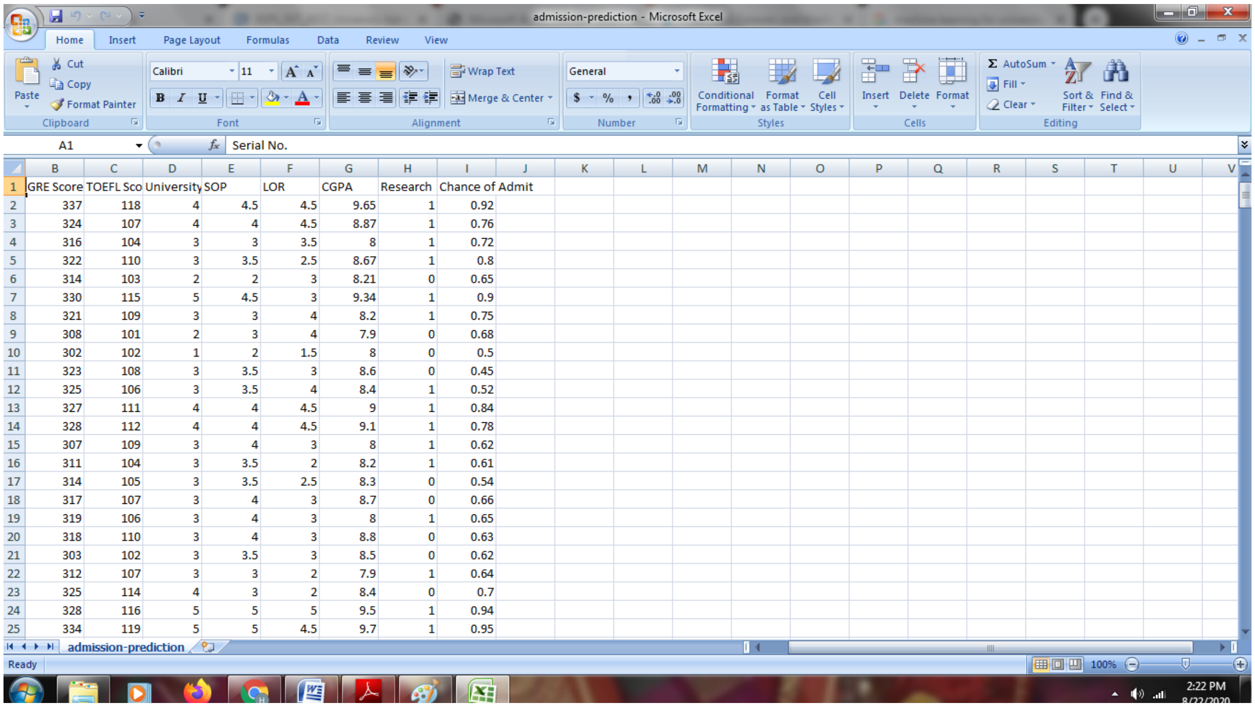
● HTML

● Flask

We developed this university admission prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Google Colab and Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

4. EXPERIMENTAL INVESTIGATION

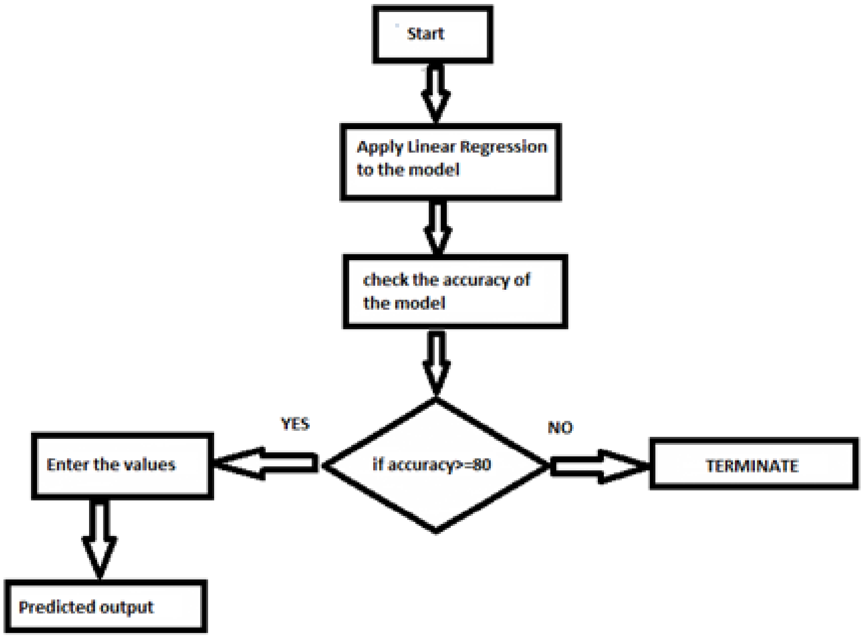
The dataset we used is derived from www.kaggle.com. It contains plenty of datasets which are real time. We choosed admission prediction dataset which contains 9 attributes and 400 rows. After that, the missing values are checked and the unwanted columns are deleted, finally we have retained to 8 attributes. Those are shown below.



A1 Serial No.

	B	C	D	E	F	G	H	I
1	GRE Score	TOEFL Sco	University	SOP	LOR	CGPA	Research	Chance of Admit
2	337	118	4	4.5	4.5	9.65	1	0.92
3	324	107	4	4	4.5	8.87	1	0.76
4	316	104	3	3	3.5	8	1	0.72
5	322	110	3	3.5	2.5	8.67	1	0.8
6	314	103	2	2	3	8.21	0	0.65
7	330	115	5	4.5	3	9.34	1	0.9
8	321	109	3	3	4	8.2	1	0.75
9	308	101	2	3	4	7.9	0	0.68
10	302	102	1	2	1.5	8	0	0.5
11	323	108	3	3.5	3	8.6	0	0.45
12	325	106	3	3.5	4	8.4	1	0.52
13	327	111	4	4	4.5	9	1	0.84
14	328	112	4	4	4.5	9.1	1	0.78
15	307	109	3	4	3	8	1	0.62
16	311	104	3	3.5	2	8.2	1	0.61
17	314	105	3	3.5	2.5	8.3	0	0.54
18	317	107	3	4	3	8.7	0	0.66
19	319	106	3	4	3	8	1	0.65
20	318	110	3	4	3	8.8	0	0.63
21	303	102	3	3.5	3	8.5	0	0.62
22	312	107	3	3	2	7.9	1	0.64
23	325	114	4	3	2	8.4	0	0.7
24	328	116	5	5	5	9.5	1	0.94
25	334	119	5	5	4.5	9.7	1	0.95

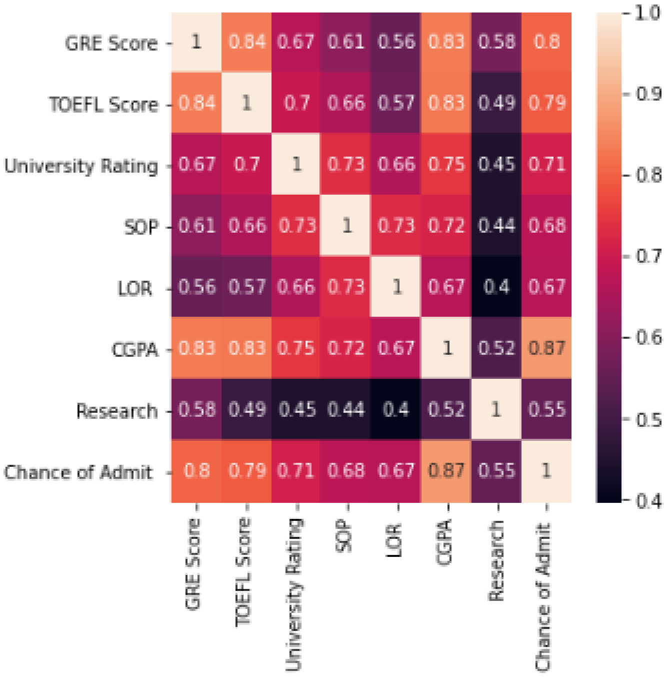
5. FLOWCHART


Start
Apply Linear Regression to the model
check the accuracy of the model
if accuracy>=80
YES
NO
Enter the values
TERMINATE
Predicted output

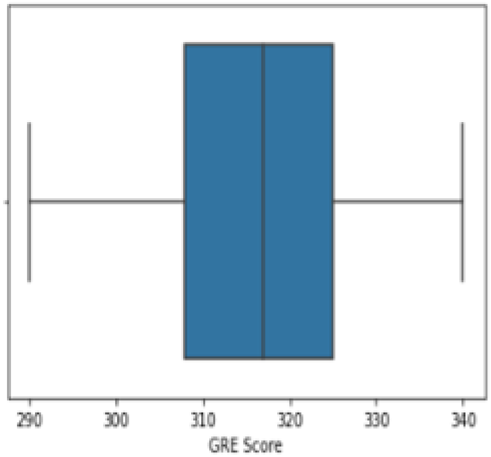
6. RESULT

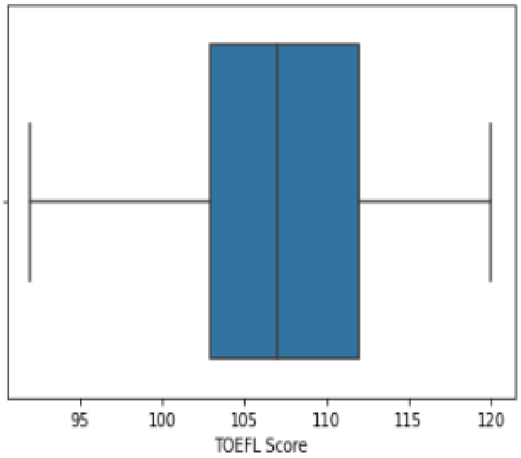
In this paper, the Linear Regression algorithm is used to predict its performance, and compared with another two machine learning methods namely the decision tree, the Random Forest. The obtained results are displayed in Table below. The results show that, the performance of Linear Regression have better performance than random forest and decision tree. The Linear Regression is best with an accuracy of 80% higher than RandomForest with an accuracy of 64%.

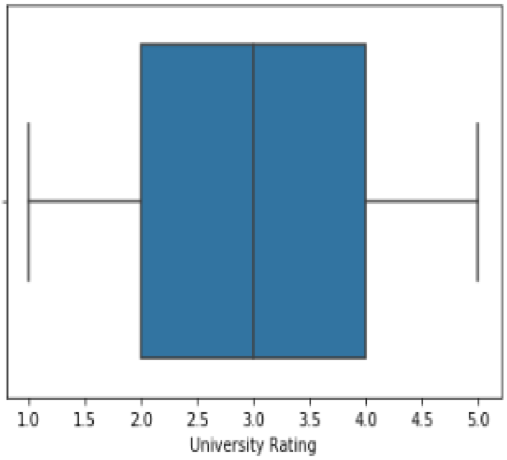
The given are the heatmap of the dataset represents the correlation between attributes and the boxplot of each attribute.

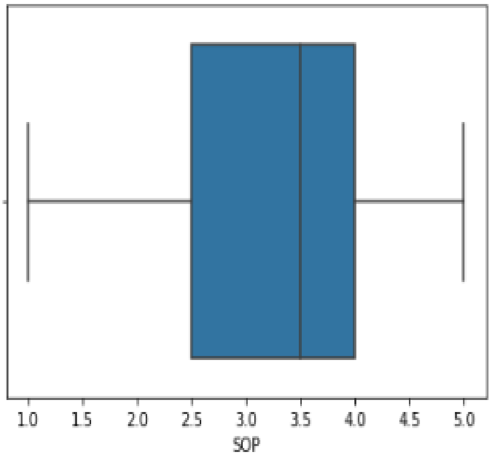


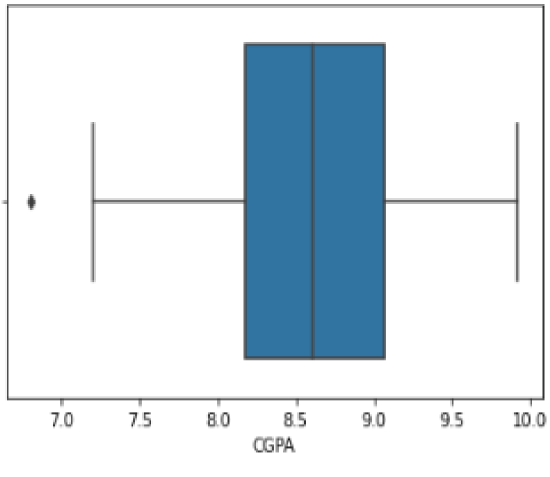
	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
GRE Score	1	0.84	0.67	0.61	0.56	0.83	0.58	0.8
TOEFL Score	0.84	1	0.7	0.66	0.57	0.83	0.49	0.79
University Rating	0.67	0.7	1	0.73	0.66	0.75	0.45	0.71
SOP	0.61	0.66	0.73	1	0.73	0.72	0.44	0.68
LOR	0.56	0.57	0.66	0.73	1	0.67	0.4	0.67
CGPA	0.83	0.83	0.75	0.72	0.67	1	0.52	0.87
Research	0.58	0.49	0.45	0.44	0.4	0.52	1	0.55
Chance of Admit	0.8	0.79	0.71	0.68	0.67	0.87	0.55	1

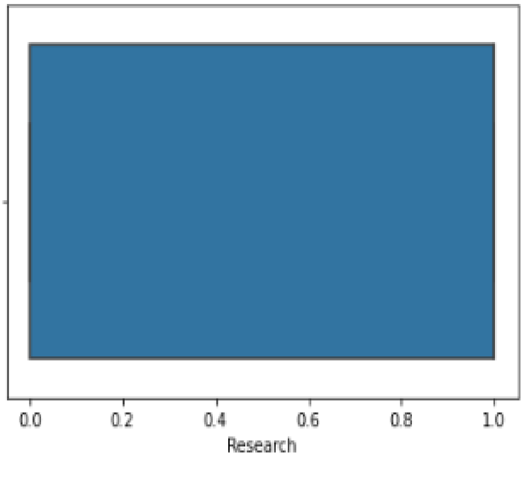

290
300
310
320
330
340
GRE Score

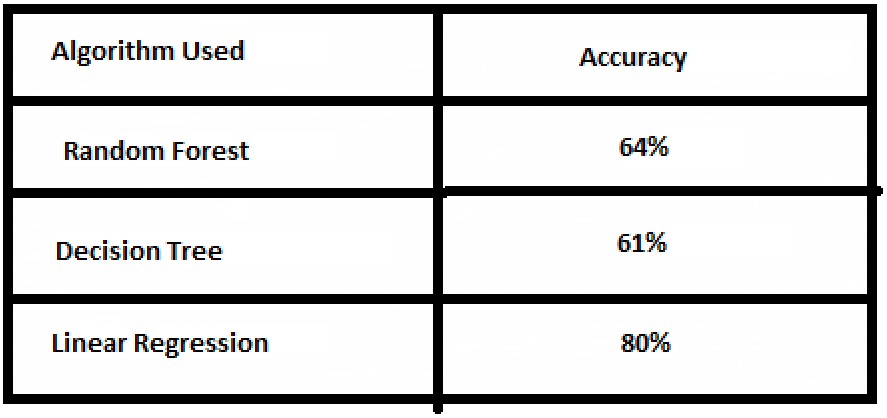

95
100
105
110
115
120
TOEFL Score


1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
University Rating


1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
SOP


7.0
7.5
8.0
8.5
9.0
9.5
10.0
CGPA


0.0
0.2
0.4
0.6
0.8
1.0
Research



Algorithm Used	Accuracy
Random Forest	64%
Decision Tree	61%
Linear Regression	80%

7. ADVANTAGES AND DISADVANTAGES

Advantages:

● For the students who are preparing to join in the universities, this will be the best model to know their chances of getting admissions.

● It is a feasible web blog, which can be accessed easily.

● The Linear Regression model gives 80% of accuracy for university admission prediction.

● It is composed using HTML and python for the web usage in real time.

● It can work in real time and predicted as soon as the necessary details for prediction are given to the model.

Disadvantages:

● It predicts the output within the range but not accurate value.

●It could not work anywhere like an web-application.

●Needs more than a single value for the prediction.

8. APPLICATIONS

●This model is mostly used for students who are trying to get the admissions in the universities based on their profiles and qualifications.

●Using this model we can also predict in different sectors like business, bank, salaries etc...

● As many students are preparing for their entries in universities without knowing whether they can make it or not, this blog will give a clear idea of it. So that they can try for another one.

9. CONCLUSION

In this paper, the Linear Regression algorithm is adopted to build a UI model for predicting university admissions and the results are compared with other two algorithms of RandomForest and Decision Tree. The model shows that Linear Regression performs best than the other two algorithms in the prediction of admissions. There is no definitive guide of which algorithms to be used. What may work on some datasets may not work on others. Therefore, always check the accuracy and predict with the dataset values.

10. FUTURE SCOPE

This Linear Regression model can also be used in the future predictions like weather forecast, job prediction, salary prediction etc. In future we can create a web application on this type of problems so that it can be accessed from everywhere with more users.

11. BIBLIOGRAPHY

● Adichie, J. N. (1967). Estimates of regression parameters based on rank test. Annals of Mathematical Statistics, 38, 894-904.

● Aia, M. A., Goldsmith, R. L., and Mooney, R. W. (1961). Predicting stoichiometric CaHP04 • 2H20.Indllstrial and Engineering Chemistry, 53, January, 55-57.

●Adcock, R. J. (1878). A problem in least squares. Analyst, 5, 53-54.

● Andrews, D. F. (1974). A robust method for multiple linear regression. Technometrics. 16. 523-531.

●Andrews, D. E, and D. Pregibon (1978). Finding the outliers that matter. Journal of the Royal Statistical Society, Series B, 4, 84-93.

●Barnett, V. D. (1967). A note on linear structural relationships when both residual variances arc known. Biometrika, 54, 670-672.

●Dobson. A. J. (1990). Introduction to Generalized Linear Models. London: Chapman and Hall.

●Marquardt, D. W. (1963). An algorithm for least squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics, 11,431-441.

●McCullagh, P., and J. A. NeIder (1989). Generalized Linear Models, 2nd ed. London: Chapman and Hall.

●Schemper. M. (1990). The explained variation in proportional hazards regression. Biometrika, 77,216-218.

APPENDIX

HTML:

```
<!DOCTYPE html>
<html>
<!--From https://codepen.io/frytyler/pen/EGdtg--><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="../static/css/style.css">    <style>
        .login {
            top: 20%;
        }
    </style>
</head><body>
    <h1>University Admission Prediction</h1>
    <div class="login">        <!-- Main Input For Receiving Query to our ML -->
        <form action="{{ url_for('y_pred')}}" method="post">
            <label for="grescore">GRE Score</label>
            <input type="text" name="grescore" placeholder="Enter GRE Score" id="grescore"
required="required" />            <label for="tofelscore">TOFEL Score</label>
            <input type="text" name="tofelscore" placeholder="Enter TOFEL Score"
id="tofelscore" required="required" />            <label for="universityrating">University
Rating</label>
            <input type="text" name="universityrating" placeholder="Enter University Rating"
id="universityrating" required="required" />            <label for="sop">SOP(Statement of
Purpose)</label>
            <input type="text" name="sop" placeholder="Enter SOP" id="sop"
required="required" />            <label for="lor">LOR(Letter of Recommendation)</label>
```

```
            <input type="text" name="lor" placeholder="Enter LOR" id="lor" required="required"
min="0" max="5" />            <label for="cgpa">CGPA</label>
            <input type="text" name="cgpa" placeholder="Enter CGPA" id="cgpa"
required="required" />            <label for="research">Research</label>
            <input type="text" name="research" placeholder="Research" id="research"
required="required" />            <button type="reset" class="btn btn-primary
btn-large">Clear</button>
            <button type="submit" class="btn btn-primary  btn-large">Predict</button>
<br>
            <br>Chances of getting admission in a university : {{ prediction_text }}
        </form>    </div></body></html>
```

CSS:

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn {
    display: inline-block;
    margin-top: 1rem;
    *display: inline;
    *zoom: 1;
    width: 40%;
    padding: 4px 10px 4px;
    margin-bottom: 0;
    font-size: 13px;
    line-height: 18px;
    color: #333333

;
    text-align: center;
    text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
    vertical-align: middle;
    background-color: #F5F5F5

;
    background-image: -moz-linear-gradient(top, #FFFFFF

, #E6E6E6

);
    background-image: -ms-linear-gradient(top, #FFFFFF
, #E6E6E6

);
    background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#FFFFFF

), to(#E6E6E6

));
    background-image: -webkit-linear-gradient(top, #FFFFFF

, #E6E6E6

);
    background-image: -o-linear-gradient(top, #FFFFFF

, #E6E6E6

);
    background-image: linear-gradient(top, #FFFFFF

, #E6E6E6

);
    background-repeat: repeat-x;
    filter: progid: dximagetransform.microsoft.gradient(startColorstr=#ffffff,
endColorstr=#e6e6e6, GradientType=0);
    border-color: #E6E6E6

 #E6E6E6

 #E6E6E6

;
    border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
    border: 1px solid #E6E6E6

;
    -webkit-border-radius: 4px;
    -moz-border-radius: 4px;
    border-radius: 4px;
    -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
    -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
    cursor: pointer;
    *margin-left: .3em;
}.btn:hover,
.btn:active,
.btn.active,
.btn.disabled,
.btn[disabled] {
    background-color: #E6E6E6

;
}.btn-large {
    padding: 9px 14px;
    font-size: 15px;
    line-height: normal;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}.btn:hover {
    color: #000000

;
    text-decoration: none;
    background-color: #E6E6E6

;
    background-position: 0 -15px;
    -webkit-transition: background-position 0.1s linear;
    -moz-transition: background-position 0.1s linear;
    -ms-transition: background-position 0.1s linear;
    -o-transition: background-position 0.1s linear;
    transition: background-position 0.1s linear;
}.btn-primary,
.btn-primary:hover {
    text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
    color: #FFFFFF

;
}.btn-primary.active {
    color: rgba(255, 255, 255, 0.75);
}.btn-primary {
    background-color: #4A77D4

;
    background-image: -moz-linear-gradient(top, #6EB6DE

, #4A77D4

);
```

```
    background-image: -ms-linear-gradient(top, #6EB6DE

, #4A77D4

);
    background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6EB6DE

), to(#4A77D4

));
    background-image: -webkit-linear-gradient(top, #6EB6DE

, #4A77D4

);
    background-image: -o-linear-gradient(top, #6EB6DE

, #4A77D4

);
    background-image: linear-gradient(top, #6EB6DE

, #4A77D4

);
    background-repeat: repeat-x;
    filter: progid: dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0);
    border: 1px solid #3762BC

;
    text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.4);
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5);
}.btn-primary:hover,
.btn-primary:active,
.btn-primary.active,
.btn-primary.disabled,
.btn-primary[disabled] {
    filter: none;
    background-color: #4A77D4

;
}.btn-block {
    width: 100%;
    display: block;
}* {
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    -ms-box-sizing: border-box;
    -o-box-sizing: border-box;
    box-sizing: border-box;
}* {
    box-sizing: border-box;
}body {
    background-repeat: no-repeat;
    width: 100%;
    height: 100%;
    font-family: 'Open Sans', sans-serif;
    color: #fff;
    font-size: 18px;
    text-align: center;
    letter-spacing: 1.2px;
    background: #F0F2F0

;
    /* fallback for old browsers */
    background: -webkit-linear-gradient(to right, #000C40

, #F0F2F0

);
    /* Chrome 10-25, Safari 5.1-6 */
    background: linear-gradient(to right, #000C40

, #F0F2F0

);
    /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */}.login {
    width: 35%;
    margin: 1rem auto;
}h1 {
    color: #fff;
    font-size: 4rem;
    text-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
    letter-spacing: 1px;
    text-align: center;
    padding: 0;
    margin-top: 10px;
    background: inherit;
    position: sticky;
    top: 0;
```

```
}input {
    text-align: center;
    width: 100%;
    margin-bottom: 1rem;
    background: rgba(0, 0, 0, 0.3);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: #fff;
    text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.3);
    border: 1px solid rgba(0, 0, 0, 0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.2), 0 1px 1px rgba(255, 255, 255,
0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}input:focus {
    box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.4), 0 1px 1px rgba(255, 255, 255,
0.2);
}@media screen and (max-width: 700px) {
    .login {
        width: 60%;
    }
    h1 {
        font-size: 2rem;
    }
}
```

APP.PY:

```
from flask import Flask, request, jsonify, render_template
import pickleapp = Flask(__name__)
model = pickle.load(open('Chance of Admit.pkl', 'rb'))@app.route('/')
def home():
    return render_template('base.html')@app.route('/y_pred',methods=['POST'])
```

```
def y_pred():
    '''
    For rendering results on HTML GUI
    '''
    grescore        =request.form['grescore']
    tofelscore      =request.form['tofelscore']
    universityrating =request.form['universityrating']
    sop             =request.form['sop']
    lor             =request.form['lor']
    cgpa            =request.form['cgpa']
    research        =request.form['research']

data=[[int(grescore),int(tofelscore),int(universityrating),float(sop),float(lor),float(cgpa),int(research)]]
    prediction  = model.predict(data)
    print(prediction)
    output=prediction[0][0]
    return render_template('base.html', prediction_text=output)if __name__ == "__main__":
    app.run(debug=True)
```