STUDENT ADMISSION PREDICTION

A Project Report submitted in the partial fulfillment of the

Requirements for the award of the degree

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

J. KOTESWARARAO	**(19471A05M0)**
D.HARISH	**(19471A05L0)**
SK. MOHAMMAD SEERAJ	**(19471A05O6)**

Under the esteemed guidance of

Mr. M. VENKAT RAO, M.Tech.

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE

AUTONOMOUS

Accredited by NAAC with A+ Grade and NBA under (Tier -1)
NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name “STUDENT ADMISSION PREDICTION” is a bonafide work done by the team J.KOTESWARARAO(19471A05M0), D.HARISH(19471A05L0), SK. MOHAMMAD SEERAJ(19471A05O6) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2022-2023.

PROJECT GUIDE

Mr. M.Venkat Rao M.Tech.
Asst. Professor

PROJECT CO-ORDINATOR

Dr. M. Sireesha, M.Tech.,Ph.D
Assoc. Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao M.Tech., Ph.D
Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao B.Sc,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. M. Sreenivasa Kumar M.Tech.,Ph.D(UK).,MISTE.,FIE(I),** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao M.Tech.,Ph.D,** HOD of CSE department and also to our guide **Mr. M.Venkat rao M.Tech,** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr. M. Sireesha M.Tech.,Ph.D,** Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

J. Koteswararao	**(19471A05M0)**
D. Harish	**(19471A05L0)**
SK. Mohammad Seeraj	**(19471A05O6)**

ABSTRACT

In response to the highly competitive job market at present times, an increased interest in graduate studies has arisen. This has not only burdened applicants but also led to an increased workload on admission faculty members of universities. Any chance of abridging the admission process impelled applicants and faculty workers to look for faster, efficient, and more accurate methods for predicting admissions. The goal approach of this paper is to implement and compare several supervised predictive analysis methods on a labeled dataset based on real applications from the universities. Regression, classification, and Ensemble methods are all the supervised methods that are to be employed for prediction. The dataset relies profoundly on the academic performance of the applicants during their undergrad years. The coefficient of determination, as well as precision and accuracy, are the measures used to compare the different models.

All predictive methods proved to show accurate results, however; certain methods proved to be more promising than others were. Predictions were obtained within short time frames, which in turn will cut down the time in the admission process by the chance of getting Admission.

NEC NARASARAOPETA
ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching,imbibing experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems.

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NEC NARASARAOPETA
ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature.

CO425.4: Design and Modularize the project.

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		✓											✓		
C425.2	✓		✓		✓								✓		
C425.3				✓		✓	✓	✓					✓		
C425.4			✓			✓	✓	✓					✓	✓	
C425.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C425.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation betweenCO's and PO's:

1. **Low level**
2. **Medium level**
3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a software for student admission prediction	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into different modules	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our three members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project can be handled easily and in future updates in our project can be done by predicting admissions of different colleges	PO4, PO7
CC4.2.8 CC4.2.	The design includes software components like model and python application.	PO5, PO6

INDEX

S.NO	CONTENTS	PAGE.NO
1	**Introduction**	**1**
	1.1 Introduction	**1**
	1.2 Existing System	**1**
	1.3 Proposed System	**1**
	1.4 System Requirements	**2**
2	**Literature Survey**	**3**
	2.1 Machine Learning	**3**
	2.2 Some Machine Learning Methods	**4**
	2.3 Applications of Machine Learning	**5**
	2.4 Implementation of Machine Learning Using Python	**5**
3	**System Analysis**	**9**
	3.1 Scope of the Project	**9**
	3.2 Analysis	**9**
	3.3 Data Pre-Processing	**10**
	3.4 Confusion Matrix	**12**
4	**Implementation Code**	**15**
5	**Screenshots**	**22**
6	**Test Cases**	**24**
7	**Future Enhancement**	**28**
8	**Conclusion**	**29**
9	**References**	**30**

LIST OF FIGURES

S.NO.	LIST OF FIGURES	PAGE NO
1	Fig:3.1 Dataset	9
2	Figure:3.2 Dataset Instructions	10
3	Fig:3.1 Data Pre-processing	12
4	Fig 3.1 Confusion Matrix	13
5	Fig:5.1 Anaconda prompt	22
6	Fig:5.2 Home page	22
7	Fig:5.3 Input values screen	23
8	Fig:6.1 Output screen for Invalid student data	24
9	Fig:6.2 Output screen for Invalid Student data	25
10	Fig:6.3 Output screen for valid Student data	26
11	Fig:6.4 Output screen for valid Student data	27

1. INTRODUCTION

1.1 Introduction

Machine Learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly.

This project mainly focuses on the how the get the admission chance of a student by taking the input values that can be predicted by using some related information. The scope of this project is the user can calculate the chance of getting admission based on some factors.

1.2 Existing System

Student may predict their next admission chance by some mathematical calculations on previous marks. It takes lot of time and it is very difficult to predict manually.

Disadvantages

1. Doesn’t generate accurate and efficient results
2. Computation time is very high.
3. Difficulty in calculation.

1.3 Proposed System

We proposed to develop a system which will help students to predict their Admission chance in higher studies purpose based on some attributes which are nothing but their previous marks data. So, this system can predict students Admission chance to get in the various universities within a short period of time and reduces manual difficulty. Proposed system uses Multiple

Random Forest Regression Algorithm. This algorithm provides the output by processing different attributes.

Advantages

1. Generates accurate and efficient results

2. Computation time is greatly reduced

1.4. System Requirements

1.4.1 Hardware Requirements

• Processor	:	intel®core™i7-7500UCPU@2.70gh
• Cache memory	:	4MB(Megabyte)
• RAM	:	8 gigabyte(GB)

1.4.2 Software Requirements

• Operating System	:	Windows 10 Home, 64 bit Operating System
• Coding Language	:	Python
• Python distribution	:	Anaconda, Jupyter Notebook

2. LITERATURE SURVEY

2.1 Machine Learning

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves.The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly.

Predicting Factors: Various studies have identified several predictors that help students make admissions decisions. For example, academic records such as GPA and standardized test scores are often used as primary predictors. Other factors such as extracurricular activities, personal statements, and letters of recommendation were also found to be significant predictors of admission.

Performance measures: The study used various performance measures to assess the accuracy and performance of the predictive models.

Data preprocessing: Data preprocessing is a critical step in developing accurate predictive models. The research used various techniques such as feature selection, normalization and coding to prepare the data for analysis.

Model interpretability: Model interpretability is an important factor in ensuring that predictions are transparent and understandable.

The research used a variety of techniques, including attribute, importance analysis and decision tree visualization, to explain patterns, and understand factors that influence admissions decisions.

By using machine learning algorithms and other statistical techniques to analyze this data, admission prediction models can be developed that are able to identify patterns and trends that can help predict which students are most likely to be admitted. This can be especially helpful for students who are applying to highly competitive schools, as it can help them make more informed decisions about where to apply and increase their chances of being admitted.

In general, literature on student admissions predictions shows that machine algorithms can effectively predict admissions outcomes. However, the accuracy and performance of the model may vary depending on the data and the problem to be solved. Additionally, interpretability and transparency are key factors to ensure reliable and trustworthy forecasts.

2.2 Some machine learning methods

Machine learning algorithms are often categorized as supervised and unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.
- The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning
- accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additionalresources.

- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 Applications of machine learning

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection

2.4 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability.Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference

between the SciPy library and the SciPy stack. The SciPy is one of the core packages that makeup the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit- learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self- verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used by individuals for their own projects.

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language Processing(NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc.

3. SYSTEM ANALYSIS

3.1 Scope of the project

This project mainly focuses on the how the get the admission chance of a student by taking the input values that can be predicted by using some related information. The scope of this project is the user can calculate the chance of getting admission based on some factors and then predict the Admission chance of a student.

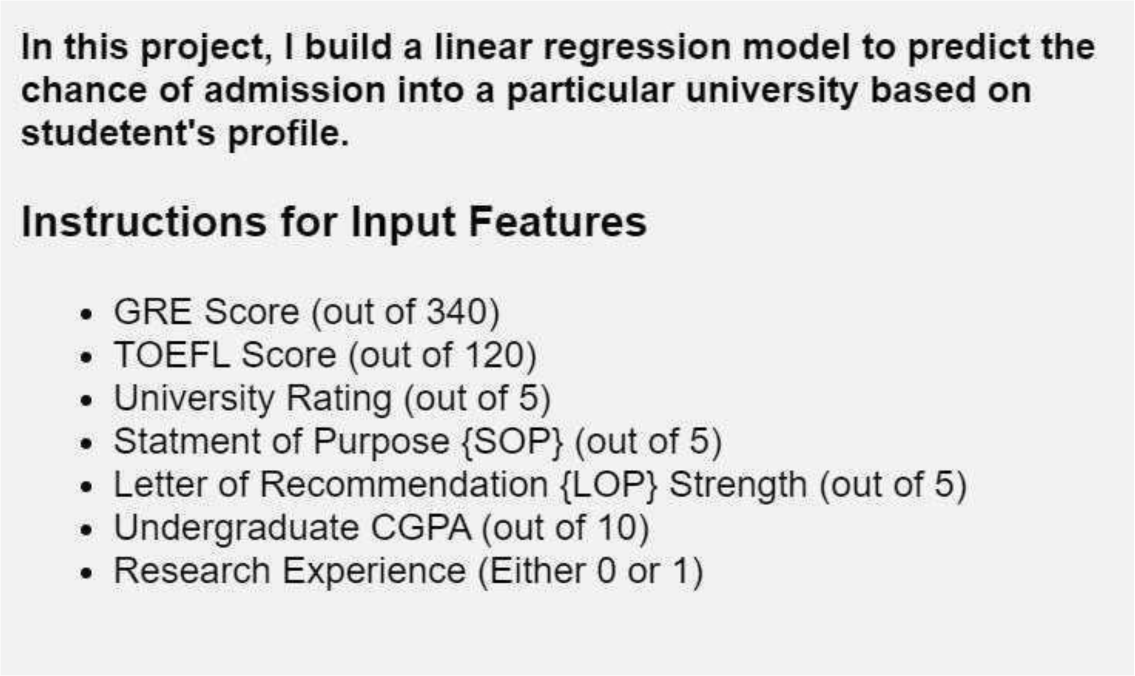
3.2 Analysis

The dataset used in this project is Students previous marks details dataset. The dataset contains

9 Attributes.

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit	
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65
...	...	...	...	...	...	...	...	...	...
395	396	324	110	3	3.5	3.5	9.04	1	0.82
396	397	325	107	3	3.0	3.5	9.11	1	0.84
397	398	330	116	4	5.0	4.5	9.45	1	0.91
398	399	312	103	3	3.5	4.0	8.78	0	0.67
399	400	333	117	4	3.2	4.0		1	

Fig:3.1 Dataset



In this project, I build a linear regression model to predict the chance of admission into a particular university based on studetent's profile.

Instructions for Input Features

- GRE Score (out of 340)
- TOEFL Score (out of 120)
- University Rating (out of 5)
- Statment of Purpose {SOP} (out of 5)
- Letter of Recommendation {LOP} Strength (out of 5)
- Undergraduate CGPA (out of 10)
- Research Experience (Either 0 or 1)

Figure:3.2 Dataset Instructions

3.3 Data Pre-processing:

Before feeding data to an algorithm we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest Regression algorithm it does not support null values. So that those null values have to be managed using raw data.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm[7]. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Preprocessing is the first step while creating the machine learning model. It is the process of converting raw dataset into cleaned dataset. Raw data contains noise, missing values, duplicate values which is not suitable for machine learning model. So, preprocessing is required for cleaning the data and making it suitable for machine learning model.

Data cleaning: This involves identifying and addressing missing data, addressing outliers, and resolving any inconsistencies or errors in the data

FeatureSelection : It is important to select the most relevant characteristics that are likely to have an impact on admission results. This may involve the use of techniques such as correlation analysis orfeature importance ranking.

Data Normalization: This involves scaling data so that it has a consistent range and distribution.This is important because different features can have different scales, and normalization can help ensurethat all features.

Data encoding: One of the most common techniques for categorical encoding is one-hot encoding or dummy variable encoding. In this technique, each category in a categorical variable is represented asa separate binary variable, with a value of 1 if the observation falls into that category, and 0 otherwise.

For example, if we have a categorical variable "Color" with categories "Red," "Blue," and "Green," we would create three binary variables, "Color_Red," "Color_Blue," and "Color_Green," and assign them values of 1 or 0 depending on the color of each observation. Overall, categorical encoding is an important step .preparing data for machine learning algorithms.

By transforming categorical variables into a numeric format, we can use them as inputs to algorithmssuch as linear regression, decision trees, and neural networks, allowing us to build accurate and effective predictive models.

Data splitting: Data splitting is a technique used in machine learning to divide a dataset into two or more subsets for the purpose of training and evaluating predictive models. The most common type ofdata splitting involves dividing the dataset into two subsets: a training set and a test set.

The training set is used to develop the predictive model. This involves training the model using various machine learning algorithms such as linear regression, decision trees, and neural networks. The goal of the training phase is to develop a model that accurately predicts the outcome variable based on the input variables.

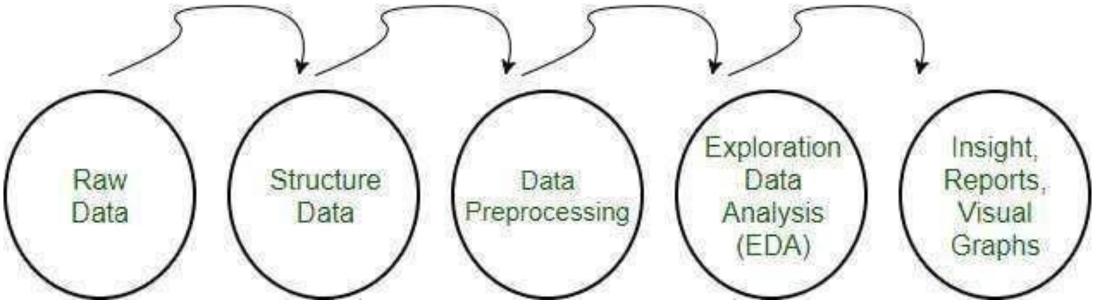

Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Fig:3.3 Data Pre-processing

3.3.1 Missing values

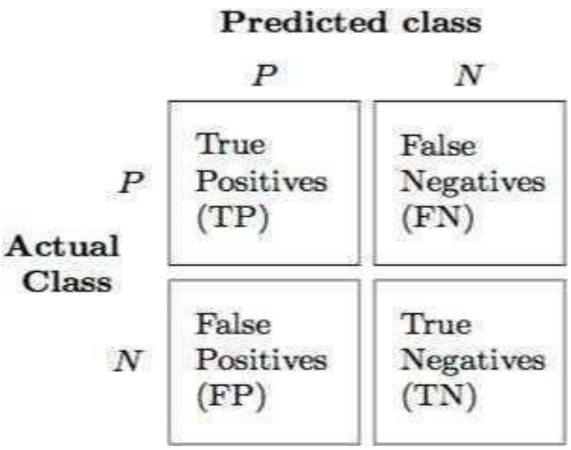
Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value NaN. So that pandas can detect the missing values. The fig1 below is a heatmap representing the missing values. In these graph missing values are present in ca, thal features. We have filled that missing values using the median of the features. After filling the missing values our heat map looks like below figure[8].

Handling missing values is important beacause Machine learning algorithms donot support missing values in the data. Some of the techniques to handle missing values are:

1. Replace with mean or median or mode value
2. Back-fill or forward-fill
3. Deleting row

3.4 CONFUSION MATRIX

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on aset of test data for which the true values are known.



		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Fig 3.4 Confusion Matrix

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly,a true negative (tn) is an outcome where the model correctly predicts the negative class.

A false positive (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the disease were identified as having the disease.

TPR = tp / (tp + fn)

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.

TNR = tn / (tn + fp) =1-FPR

Precision or positive predictive value (PPV)

If the test result is positive what is the probability that the patient actually has the disease. PPV

= tp / (tp + fp)

Negative predictive value (NPV)

If the test result is negative what is the probability that the patient does not have disease.

NPV = tn / (tn +fn)

Miss rate or false negative rate (FNR)

It is the proportion of the individuals with a known positive condition for which the test resultis negative.

FNR =fn / (fp +tn)

Fall-out or false positive rate (FPR)

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

FPR = fp/ (fp +tn)

False discovery rate (FDR)

It is the proportion of all the people identified as having the disease who do not have the disease.

FDR =fp / fp +tp

False omission rate (FOR)

It is the proportion of the individuals with a negative test result for which the true conditionis positive.

FOR =fn / (fn +tn)

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified.ACC

= (tp +tn) / (tp +tn +fp +fn)

F1 score

It is the harmonic mean of precision and sensitivityF1 =

2tp / (2tp+fp +fn)

4.IMPLEMENTATION CODE

Random Forest Regression and Flask:

```
import pandas as pd

import numpy as nm

import matplotlib.pyplot as plt

import seaborn as sns


admission=pd.read_csv('https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/Datasets/master/Regression/Admission_Predict.csv')

admission


admission.head()

admission.drop('Serial No.',axis=1,inplace=True)

admission.head(4)

admission.isnull().sum()

admission.info()

admission.describe()

admission.corr()


# Heatmap for correlation matrix

corr_matrix = admission.corr()

plt.figure(figsize=(9,9,))

sns.heatmap(corr_matrix, annot=True)

plt.show()
```

```
 import matplotlib.pyplot as plt
 import seaborn as sns
 #plot diagram for admission data
 sns.pairplot(admission)

 # heatmp for heads of admission
 plt.figure(figsize=(10,6,))
 sns.heatmap(admission.head(),annot=True,linewidths=0)
 plt.show()
 admission['GRE Score'].plot(figsize=(11,8))
 x = admission.iloc[:,[0,1,2,3,4,5,6]]
 x
 x.head(3)
import pandas as pd
 y=admission.iloc[:,[7]]
 y

 from sklearn.ensemble import RandomForestRegressor
 rnd_reg = RandomForestRegressor(n_estimators=20)
 rnd_reg
 admission.info()
 admission.corr()
 admission.shape
```

```
### Bulk data prediction
y_pred1 = rnd_reg.predict(x)
print('admitting chance:', y_pred1)
rnd_reg.predict([[230,400,1,6.0,7.5,9.4,0]])

import        pickle
import numpy as np
pickle.dump(rnd_reg,  open('rnd_reg.pkl','wb'))
rnd_model=pickle.load(open('rnd_reg.pkl','rb'))
rnd_model.predict([[320, 110, 1, 5, 5, 9, 1]])

from sklearn.metrics import accuracy_score
accuracy=rnd_reg.score(x,y_pred1)
accuracy
accuracy2=rnd_reg.score(x,y)
accuracy2

import pickle
pickle.dump(randomForest_model, open('randomForest_model.pkl','wb'))
pickle.load(open('randomForest_model.pkl','rb'))
model.predict([[320, 110, 1, 5, 5, 9, 1]])

accuracy = randomForest_model.score(X_train,y_train)
accuracy
```

Python App code:

```
import pandas as pd

import pickle

from flask import Flask, request, render_template

from sklearn.ensemble import RandomForestRegressor

app = Flask(__name__)

rnd_model=pickle.load(open('rnd_reg.pkl','rb'))


@app.route('/')

def index():

    return render_template('index.html')


@app.route('/predict', methods=['GET','post'])

def predict():

    GRE_Score = int(request.form['GRE Score'])

    TOEFL_Score = int(request.form['TOEFL Score'])

    University_Rating = int(request.form['University Rating'])

    SOP = float(request.form['SOP'])

    LOR = float(request.form['LOR'])

    CGPA = float(request.form['CGPA'])

    Research  = int(request.form['Research'])

    final_data = pd.DataFrame([[GRE_Score, TOEFL_Score, University_Rating, SOP, LOR, CGPA, Research]])

    predict = rnd_model.predict(final_data)

    output = predict[0]
    return render_template('index.html', prediction_text=output)


if___name___== "_main_":
    app.run(debug=True)
```

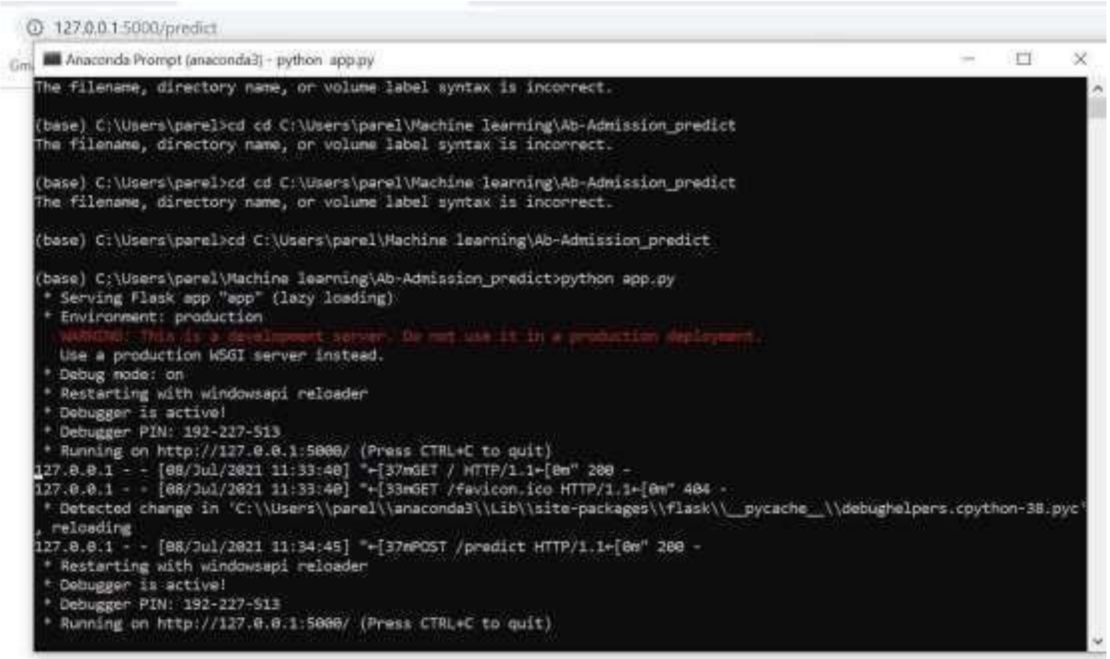
HTML code:

```
<!Doctype html>
<html>
<head>
 <style>
    label{
         width:100px;
         display:inline-block;
         }
    #form{
          border-radius:10px;
          background-color:black;
          color:white;
          width:280px;
          padding:4px;
          }
   #submit{
           border-radius:5px;
         position:relative;
         left:1px;
```

```
            height:35px;

           background-color:#FFB6C1;

           }

  </style>

</head>

<body>

<center>

 <div id="form">

    <form action="/predict" method="POST">

 <br> <label>TOEFL Score:</label><input type="number" style="width: 8em" name="GRE
Score" placeholder="TOEFL Score" required="required" min="0" max="120" ><br>


<br> <label> GRE Score:</label><input type="number" style="width: 8em" name="TOEFL
Score" placeholder="GRE Score" required="required" min="0" max="340">

<br>

  <br> <label>University Rating :</label><input type="number" style="width: 8em"
name="University Rating" placeholder="University Rating" required="required" min="1"
max="5"><br>

  <br><label>SOP:</label><input type="number" style="width: 8em" placeholder="SOP"
name="SOP" required="required" onkeypress="return check(event,value)" step="0.1"
min="1" max="5"><br>

  <br> <label> LOR:</label><input type="number" style="width: 8em" placeholder="LOR"
name="LOR" required="required" onkeypress="return check(event,value)" step="0.1"
min="1" max="5"><br>

        <br> <label> CGPA:</label><input type="number" style="width: 8em"
placeholder="CGPA" name="CGPA" required="required" onkeypress="return
check(event,value)" step="0.01" min="1" max="10"><br>
```

```
<br> <label> Research:</label><input type="number" style="width: 8em"
placeholder="Research" name="Research" required="required" min="0" max="1"><br>

  <br> <input type="submit" id="submit" style=width:118px;height:45px" value="Predict">
<br>

   </form>

 </div>

   <div class="row">

   {% if prediction_text %}

    <h4 style="text-align: center; color:red"> The Chance of getting Admission of a student is
=[{{prediction_text}}] </h4>

    {% endif %}

    </div>

<body>

</html>
```

5.SCREEN SHOTS



```
The filename, directory name, or volume label syntax is incorrect.

(base) C:\Users\parel>cd cd C:\Users\parel\Machine learning\Ab-Admission_predict
The filename, directory name, or volume label syntax is incorrect.

(base) C:\Users\parel>cd cd C:\Users\parel\Machine learning\Ab-Admission_predict
The filename, directory name, or volume label syntax is incorrect.

(base) C:\Users\parel>cd C:\Users\parel\Machine learning\Ab-Admission_predict

(base) C:\Users\parel\Machine learning\Ab-Admission_predict>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with windowsapi reloader
 * Debugger is active!
 * Debugger PIN: 192-227-513
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [08/Jul/2021 11:33:40] "←[37mGET / HTTP/1.1←[0m" 200 -
127.0.0.1 - - [08/Jul/2021 11:33:40] "←[33mGET /favicon.ico HTTP/1.1←[0m" 404 -
 * Detected change in 'C:\\Users\\parel\\anaconda3\\Lib\\site-packages\\flask\\__pycache__\\debughelpers.cpython-38.pyc'
, reloading
127.0.0.1 - - [08/Jul/2021 11:34:45] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
 * Restarting with windowsapi reloader
 * Debugger is active!
 * Debugger PIN: 192-227-513
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig:5.1 Anaconda prompt

OUTPUT HOME SCREEN PAGE:

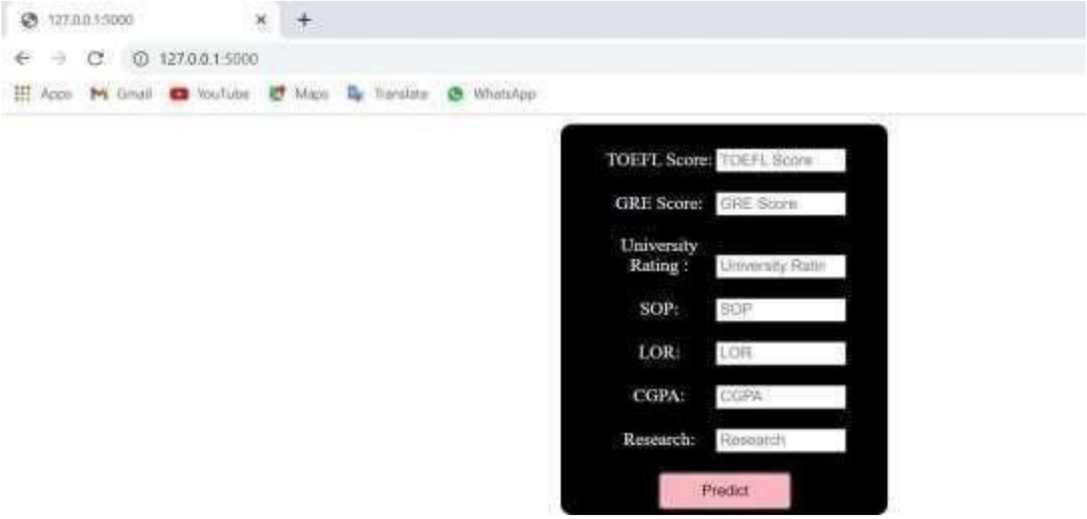
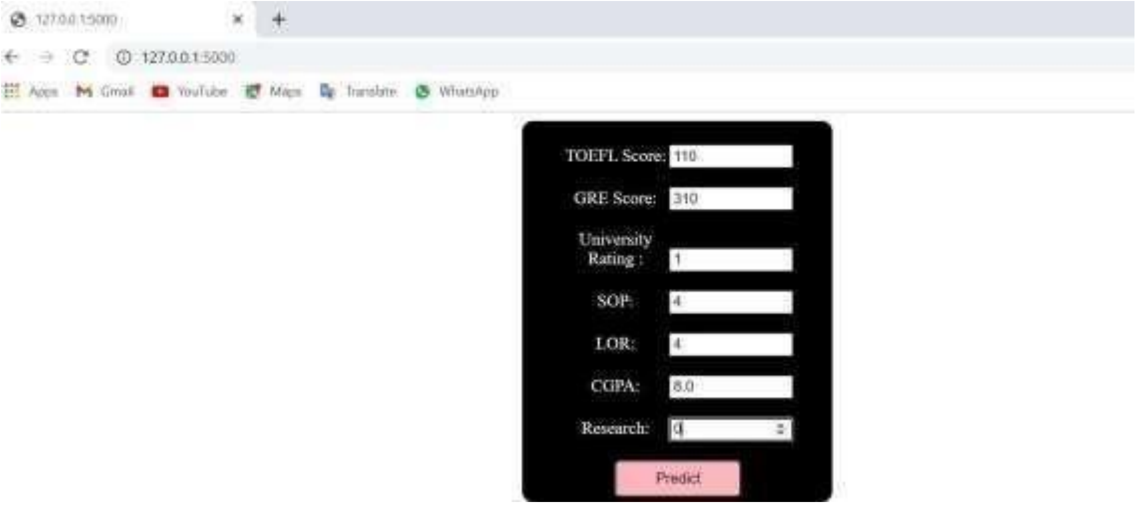


Fig:5.2 Home page

DATA INPUT PAGE:



TOEFL Score: 110

GRE Score: 310

University Rating : 1

SOP: 4

LOR: 4

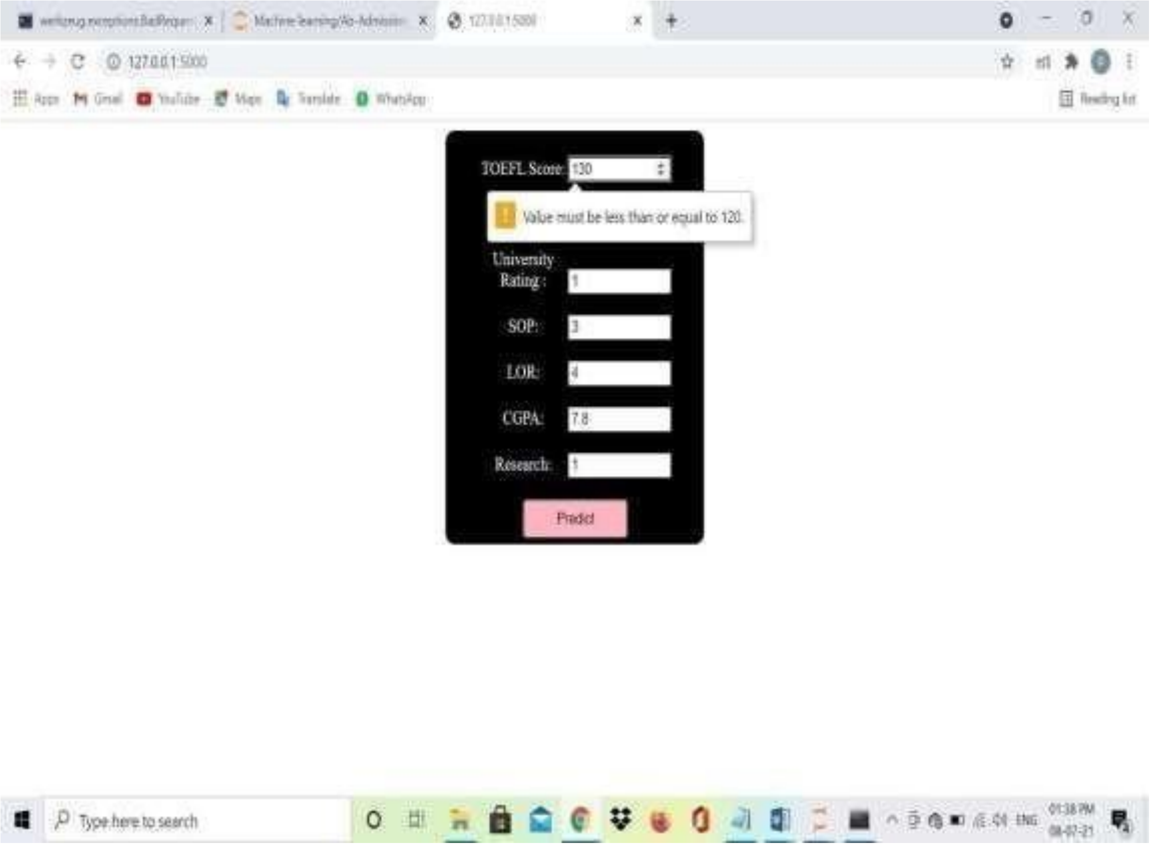
CGPA: 8.0

Research: 0

Fig:5.3 Input values screen

6.TEST CASES

INVALID INPUT 1:



TOEFL Score: 130

Value must be less than or equal to 120.

University Rating : 1

SOP: 3

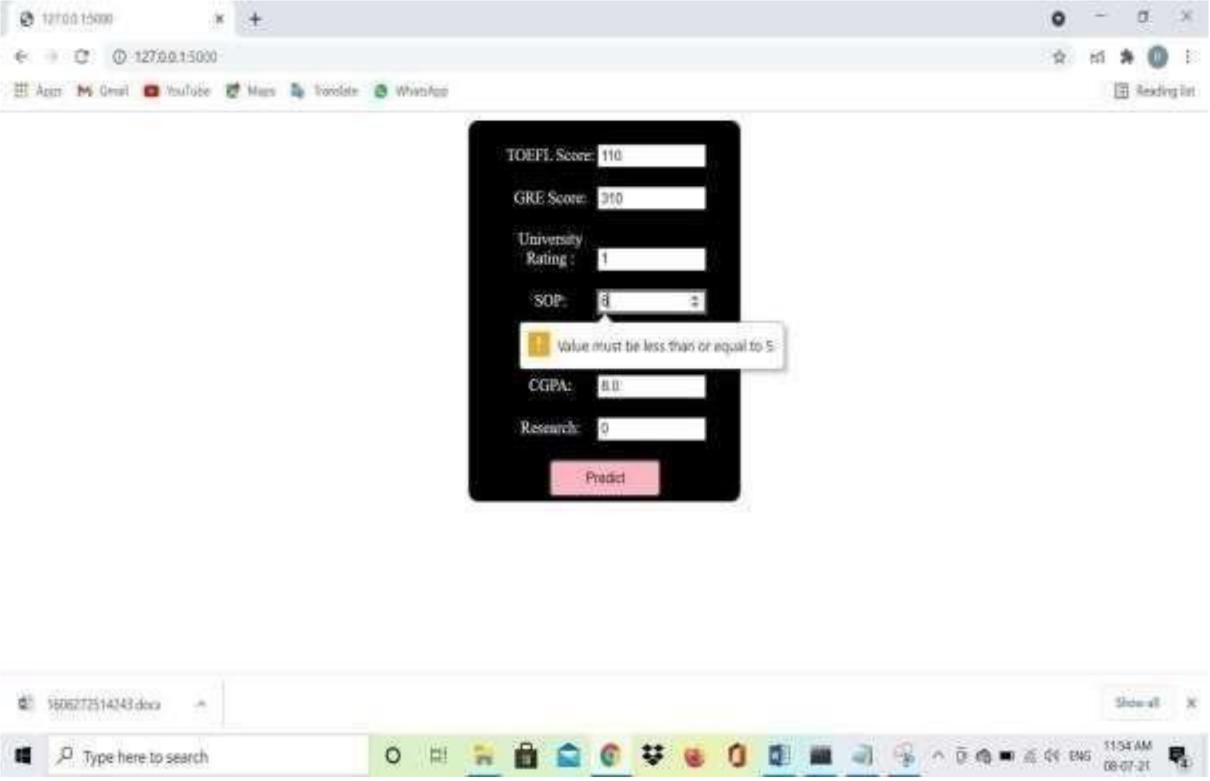
LOR: 4

CGPA: 7.8

Research: 1

Fig:6.1 Output screen for Invalid student data

INVALID INPUT 2:



TOEFL Score: 110

GRE Score: 310

University Rating : 1

SOP: 6

Value must be less than or equal to 5

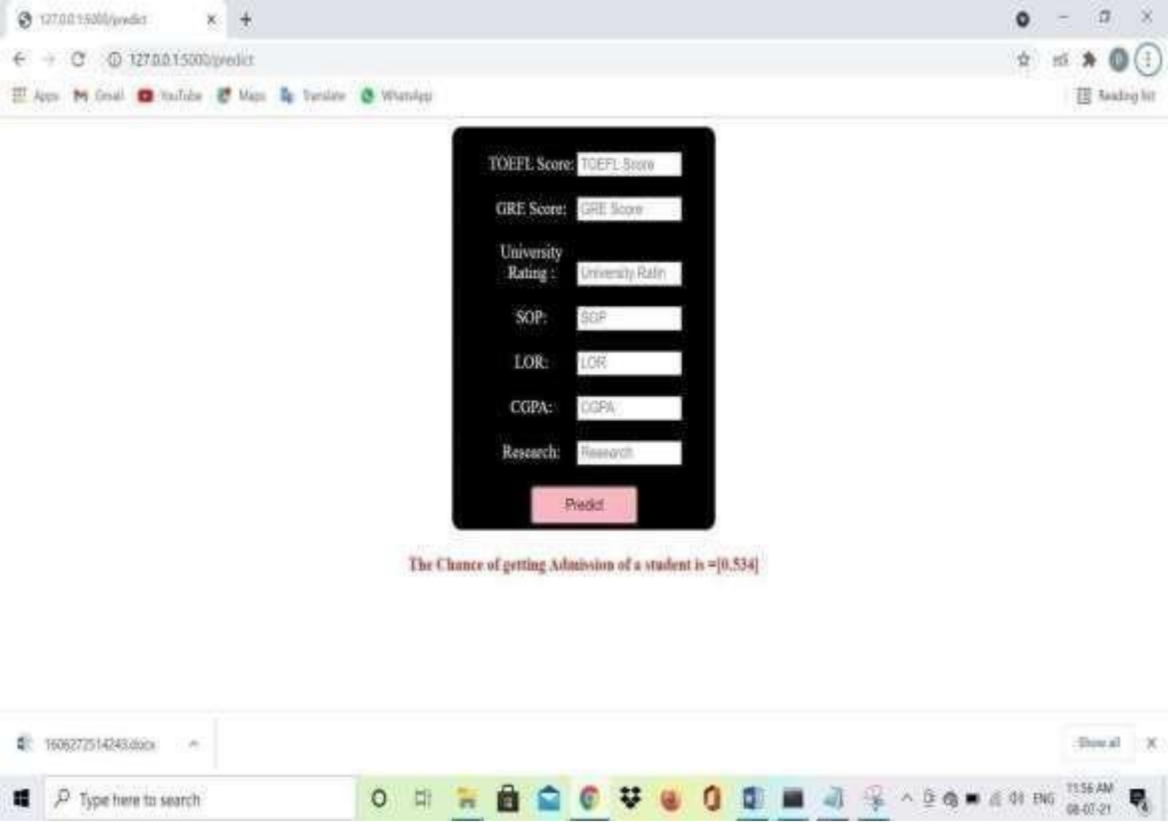
CGPA: 8.0

Research: 0

Predict

Fig:6.2 Output screen for Invalid Student data

VALID INPUT DATA 1:



TOEFL Score: TOEFL Score

GRE Score: GRE Score

University Rating : University Ratin

SOP: SOP

LOR: LOR

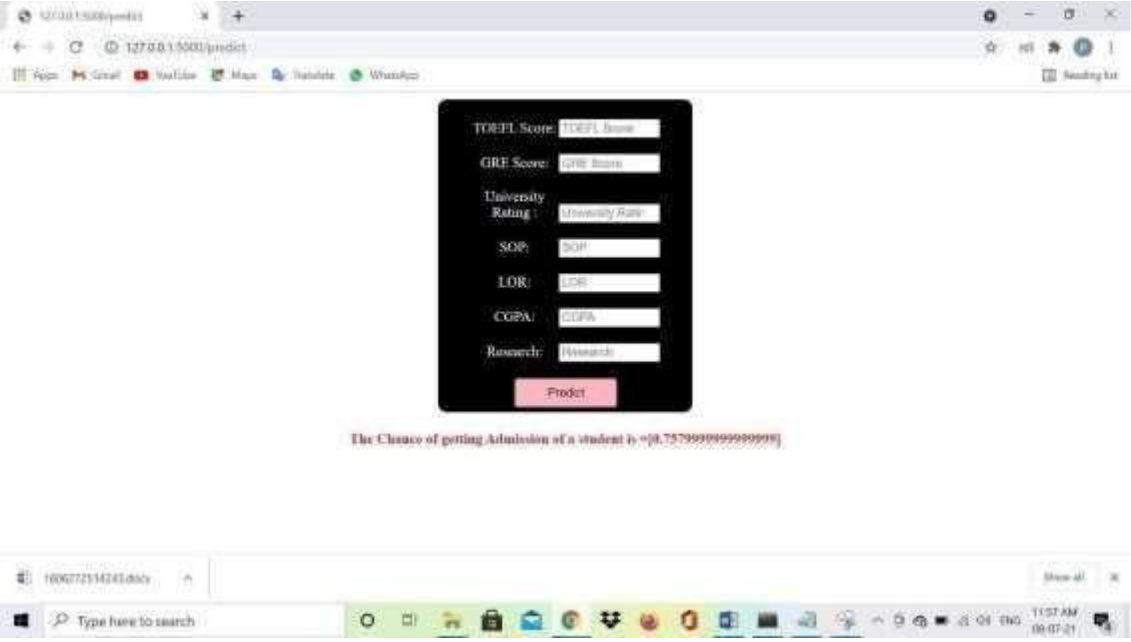
CGPA: CGPA

Research: Research

The Chance of getting Admission of a student is =[0.534]

Fig:6.3 Output screen for valid Student data

VALID INPUT DATA 2:



The Chance of getting Admission of a student is =[0.7579999999999999]

Fig:6.4 Output screen for valid Student data

7.FUTURE ENHANCEMENT

- We will update our project by providing some more attributes to a student like attendance percentage, health status etc.,
- By providing such attributes we can get more accuracy on predicting chance of getting student Admission for higher studies.
- We will Enhance that which college that particular student get the Admission by providing accurate information about universities and their minimum requirements.

8.CONCLUSION

- Finally ,Student able to predict their admission chance for higher studies based on their performance in the respective exams like GRE,TOEFL and under graduated marks as input values.

9.REFERENCES

[1] Chen, Y., Pan, C. C., Yang, G. K., & Bai, J. (2014, August). Intelligent decision system for accessing academic performance of candidates for early admission to university. In 2014 10th International Conference on Natural Computation (ICNC) (pp. 687-692). IEEE.

[2] Hasan, M., Ahmed, S., Abdullah, D. M., & Rahman, M. S. (2016, May). Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools. In 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV) (pp. 502-507). IEEE.

[3] Dietterich, T. G. (1997). Machine-learning research. AI magazine, 18(4),

[4] Yazdipour, S.,& Taherian, N.(2017, December). Data Driven Decision Supportto Fund Graduate Studies in Abroad Universities. In 2017 International Conference on Machine Learning and Data Science (MLDS) (pp. 44-50). IEEE.

[5] Waters, A., & Miikkulainen, R. (2014). Grade: Machine learning support for graduateadmissions. AI Magazine, 35(1), 64-64.

[6] Mohan S. Acharya, Asfia Armaan, Aneeta S Antony . A Comparison of Regression Models for Prediction of Graduate Admissions, 2019 IEEE International Conference onComputational Intelligence in Data Science

[7] Wickramaratna, J., Holden, S., & Buxton, B. (2001, July). Performancedegradation in boosting. In International Workshop on Multiple Classifier Systems (pp. 11-21). Springer, Berlin, Heidelberg.

[8] R. Dong, H. Wang, and Z. Yu. The module of prediction of college entrance examination aspiration. In 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, pages 1559–1562, May 2012.

[9] P. Goyal, T. Kukreja, A. Agarwal, and N. Khanna. Narrowing awareness gap by using e- learning tools for counselling university entrants. In 2015 International Conference on Advances in Computer Engineering and Applications, pages 847– 851, March 2015.

[10] Mahamudul Hasan, Shibbir Ahmed, Deen Md. Abdullah, and Md. Shamimur Rahman. Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools. In 5th International Conference on Informatics, Electronics and Vision (ICIEV), pages 502–507, 2016

STUDENT ADMISSION PREDICTION

J. Koteswararao [1], D. Harish[2], Sk. Mohammad Seeraj [3], M.Venakata Rao[4]

[1,2,3] Student, Dept. of Computer Science and Engineering, Narasaraopeta Engineering College, Narasaraopet.
[4] Assistant Professor, Dept. of Computer Science and Engineering, Narasaraopeta Engineering College, Narasaraopet.
kotichowdaryjammula58@gmail.com[1], harishpspk429@gmail.com[2], shaikseeraj33@gmail.com[3], venkatmarella670@gmail.com[4]

***Abstract* -** The student admissions prediction problem involves using data from past applicants to develop a model that can accurately predict whether a new applicant will be accepted or rejected by an educational institution. In this study, we will examine several factors that may influence admissions decisions, such as academic performance, standardized test scores, extracurricular activities, and demographic characteristics. We will use machine learning techniques to develop a predictive model and evaluate its performance on a set of test data. This research has important implications for educational institutions looking to optimize their admissions process, as well as prospective students looking to improve their chances of gaining admission.

I. Introduction

The Student Admission Prediction Project is a data-driven initiative to develop a predictive model to predict a student's chances of admission based on their academic profile and other relevant factors. The Student Admissions Forecasting Project uses data -driven insights to increase transparency and fairness in the college admissions process and make it easier to match qualified students with the right universities and colleges. Student admission prediction is a data-driven approach to predicting a student's likelihood of being admitted to a particular college or university based on their academic profile and other relevant factors. This approach involves collecting and analyzing large amounts of data on previous applicants and their admissions outcomes, as well as data on current students and their academic performance.[1]

II. Literature Review

- Predicting Factors: Various studies have identified several predictors that help students make admissions decisions. For example, academic records such as GPA and standardized test scores are often used as primary predictors. Other factors such as extracurricular activities, personal statements, and letters of recommendation were also found to be significant predictors of admission.[2]

- Performance measures: The study used various performance measures to assess the accuracy and performance of the predictive models.

- Data preprocessing: Data preprocessing is a critical step in developing accurate predictive models. The research used various techniques such as feature selection, normalization and coding to prepare the data for analysis.

- Model interpretability: Model interpretability is an important factor in ensuring that predictions are transparent and understandable.

The research used a variety of techniques, including attribute, importance analysis and decision tree visualization, to explain patterns, and understand factors that influence admissions decisions.

By using machine learning algorithms and other statistical techniques to analyze this data, admission prediction models can be developed that are able to identify patterns and trends that can help predict which students are most likely to be admitted. This can be especially helpful for students who are applying to highly competitive schools, as it can help them make more informed decisions about where to apply and increase their chances of being admitted.

In general, literature on student admissions predictions shows that machine learning algorithms can effectively predict admissions outcomes. However, the accuracy and performance of the model may vary depending on the data and the problem to be solved. Additionally, interpretability and transparency are key factors to ensure reliable and trustworthy forecasts.

III. EXISTING SYSTEM

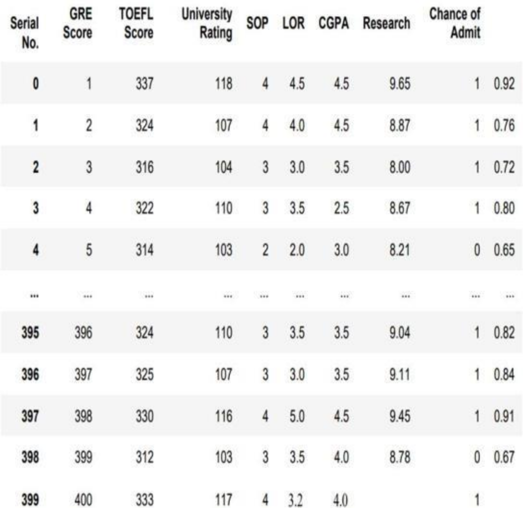
Students can predict their next admission chance by doing mathematical calculations on previous scores. It takes a long time and is difficult to plan by hand. Predicting admission chances based on previous scores using mathematical calculations can be a useful tool for students who are applying to colleges or universities. However, doing these calculations manually can be time-consuming and difficult to plan, especially for students who are applying to multiple schools or who are considering different academic programs or majors.[3]

IV proposed system

The proposed student admissions prediction system would use machine learning algorithms to analyze historical admissions data and identify patterns and correlations between various factors and admissions outcomes. The system will consider factors such as academic achievement, test scores, and other relevant factors that may affect a student's chances of admission. The system will use a multiple random forest regression algorithm. The algorithm provides output by processing different properties.

V. *Dataset and Visualization*

- The dataset includes features like GRE scores, TOEFL scores, undergraduate GPA, research experience, and admissions results.Once an appropriate data set is available, data visualization techniques can be used to explore the data and identify patterns and correlations between different characteristics and admission outcomes. This can be done using tools like scatterplots, histograms, and heatmaps. Data visualization can also be used to identify outliers or missing data that may need to be addressed before developing a predictive model.
- Data visualization is the process of representing complex data and information in a visual format that is easier to understand and interpret. This can include creating charts, graphs, maps, and other visual representations that help to communicate key trends and patterns in the data.



Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit	
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65
...	...	...	...	...	...	...	...	...	...
395	396	324	110	3	3.5	3.5	9.04	1	0.82
396	397	325	107	3	3.0	3.5	9.11	1	0.84
397	398	330	116	4	5.0	4.5	9.45	1	0.91
398	399	312	103	3	3.5	4.0	8.78	0	0.67
399	400	333	117	4	3.2	4.0		1	

Fig. 1. Dataset

VI. *Preprocessing*

- Pre-processing is an important step in developing an accurate and efficient student admission prediction model. Some key steps in pre-processing include:

1) Data cleaning: This involves identifying and addressing missing data, addressing outliers, and resolving any inconsistencies or errors in the data.

2) FeatureSelection : It is important to select the most relevant characteristics that are likely to have an impact on admission results. This may involve the use of techniques such as correlation analysis or feature importance ranking.

3) Data Normalization: This involves scaling data so that it has a consistent range and distribution.This is important because different features can have different scales, and normalization can help ensure that all features.

4) Data encoding: One of the most common techniques for categorical encoding is one-hot encoding or dummy variable encoding. In this technique, each category in a categorical variable is represented as a separate binary variable, with a value of 1 if the observation falls into that category, and 0 otherwise. For example, if we have a categorical variable "Color" with categories "Red," "Blue," and "Green," we would create three binary variables, "Color_Red," "Color_Blue," and "Color_Green," and assign them values of 1 or 0 depending on the color of each observation. Overall, categorical encoding is an important step in preparing data for machine learning algorithms. By transforming categorical variables into a numeric format, we can use them as inputs to algorithms such as linear regression, decision trees, and neural networks, allowing us to build accurate and effective predictive models.

5) Data splitting: Data splitting is a technique used in machine learning to divide a dataset into two or more subsets for the purpose of training and evaluating predictive models. The most common type of data splitting involves dividing the dataset into two subsets: a training set and a test set.

The training set is used to develop the predictive model. This involves training the model using various machine learning algorithms such as linear regression, decision trees, and neural networks. The goal of the training phase is to develop a model that accurately predicts the outcome variable based on the input variables.

In general, preprocessing is a critical step in developing an accurate and efficient student enrollment prediction model. By cleaning, selecting, normalizing, encoding and partitioning the data, we can ensure that the model is based on high quality data and can accurately predict the admission outcome of freshmen.

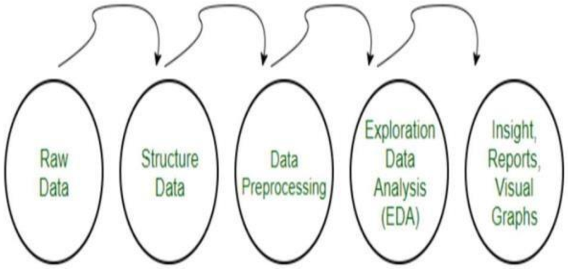

Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Fig. 2. Data Pre-processing

VII. System Analysis

This analysis includes identifying factors that influence admissions decisions, determining what data needs to be collected to make accurate predictions, and designing algorithms or models that can use that data to make predictions.

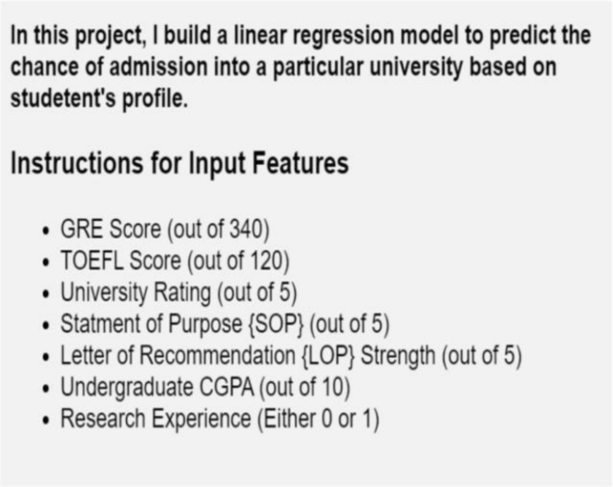
Here are the steps typically involved in an analysis of a student admissions forecasting system:

- ✓ Identifying Factors Affecting Admissions Decisions: The first step is to identify the factors that typically affect admissions decisions .
- ✓ Data Collection: Once the factors influencing admission decisions have been identified, the next step is to collect data on these factors. This data may be collected from a variety of sources, including student applications, test scores, transcripts, and other documents.
- ✓ Data cleaning and pre-processing: After data is collected, it may need to be cleaned and pre-processed to remove any errors or inconsistencies. This may involve identifying missing data points, correcting errors, and normalizing data.
- ✓ Designing Algorithms or Models: After preprocessing the data, the next step is to design algorithms or models that can use that data to make predictions. This may involve choosing an appropriate machine learning or statistical technique to analyze the data and then training a model using a subset of the data.
- ✓ Model Deployment: Once a model has been tested and validated, it can be deployed to predict student admission decisions.[4] [5][6]

In general, systematically analyzing student admissions predictions requires an in-depth understanding of the factors that influence admissions decisions, as well as the statistical and machine learning techniques needed to analyze the data and make accurate predictions . By following these steps, schools and colleges can improve their admissions process and ensure that they admit the most qualified and deserving students.

Statistical and machine learning techniques are needed to analyze the data and make accurate predictions. These techniques include data cleaning, feature engineering, model selection, and model evaluation. Data cleaning involves removing errors and inconsistencies from the data, while feature engineering involves transforming the data into a format suitable for analysis. Model selection involves choosing the best predictive model for the data, while model evaluation involves assessing the accuracy of the model.[7]

By following these steps, schools and colleges can improve their admissions process. For example, they can use predictive models to identify the most qualified and deserving students, and allocate resources more effectively. They can also use the models to identify the factors that are most important in the admissions process, and use this information to improve their admissions criteria.



In this project, I build a linear regression model to predict the chance of admission into a particular university based on studetent's profile.

Instructions for Input Features

- GRE Score (out of 340)
- TOEFL Score (out of 120)
- University Rating (out of 5)
- Statment of Purpose {SOP} (out of 5)
- Letter of Recommendation {LOP} Strength (out of 5)
- Undergraduate CGPA (out of 10)
- Research Experience (Either 0 or 1)

Fig. 3. Dataset Instructions

VIII. Methodology and Implementation

Ensemble methods are machine learning techniques that combine multiple weak classifiers to create a strong classifier. The goal of ensemble methods is to improve the accuracy of the model by reducing the errors of individual classifiers. In this case, the authors used four different ensemble methods: voting classifier, bagging, AdaBoost, and stacking.

The authors implemented a speech classifier on a dataset using previously implemented classifiers: decision trees, support vector machines (SVC), and logistic regression. The voice hyperparameter chosen was Hard Voice, which is used when SVC does not provide a probability measure.

The accuracy of the voice classifier was found to be 0.91, which is a good result. Logistic regression and SVC are powerful classifiers that use linear models to classify data. However, when comparing the accuracy of the speech classifier with the other ensemble methods, the accuracy increased. [8]

Bagging is an ensemble method that uses a classifier to randomly sample instances with replacement. The authors used the Bagging method from the MLXTEND library with four SVC classifiers, Random Forest, Gaussian Naive Bayes, and Metaclassifier Logistic Regression.

The stacking classifier is an advanced ensemble method that uses multiple models as base classifiers, and then trains a meta-classifier on the outputs of the base classifiers. After training and testing the stacking classifier on the test data, the accuracy of the classifier reached a value of 0.95, which is considered better in terms of accuracy compared to the other ensemble methods and all other classifiers.[9]

In summary, the authors used ensemble methods to improve the accuracy of the speech classifier. The stacking classifier was found to be the most accurate among all classifiers and ensemble methods. This highlights the effectiveness of using ensemble methods in machine learning to improve the accuracy of predictive models.

IX. *Result and Analysis*

When analyzing the results of a student admissions prediction system, it is important to use metrics such as accuracy and recall to assess the performance of the model used. Accuracy measures the proportion of correct predictions made by the model, while recall measures the proportion of true positives that the model correctly identified. By using these metrics, schools and colleges can determine how well the model is performing and make any necessary adjustments to improve its accuracy and efficiency.

Additionally, it is important to examine the factors identified by the model as being most important in predicting admission decisions. By understanding these key factors, schools and colleges can make better decisions about which students to admit and improve the admissions process overall. This information can also be used to identify any biases in the admissions process and make adjustments to ensure that all qualified students have an equal opportunity to be admitted.

Overall, analyzing the results of a student admissions prediction system involves a combination of statistical and machine learning techniques, as well as a thorough examination of the factors that influence admissions decisions. By using this approach, schools and colleges can improve their admissions process, admit the most qualified and deserving students, and ensure fairness and transparency in the admissions process.[10]

For example, if a model is used to predict whether a student will be admitted to a university, precision measures the proportion of students who were admitted out of all the students that the model predicted would be admitted. Precision is a useful metric when avoiding false positives is more important than identifying all positive cases.

When analyzing the results of a predictive model for student admissions, it is important to use these metrics to evaluate the performance of the model. In addition, it is important to examine the factors that the model identified as most important in predicting admissions decisions. This information can be used to improve the admissions process and make more informed decisions about which students to admit.

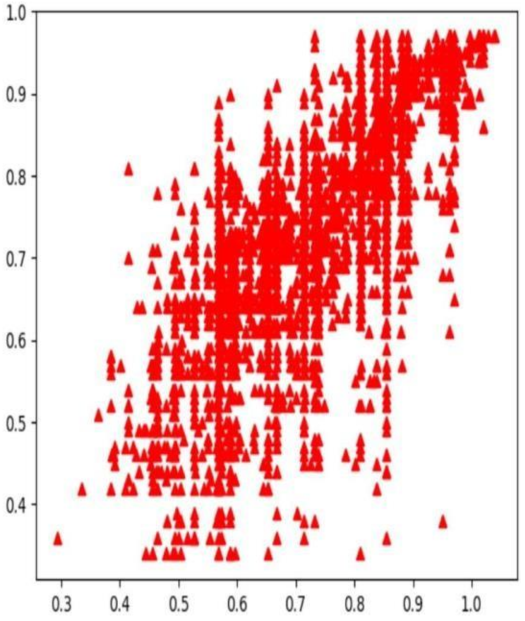


Fig. 4. Predicted values and tested values

X. *Conclusion*

The student admissions prediction problem involves using data from past applicants to develop a model that can accurately predict whether a new applicant will be accepted or rejected by an educational institution. In this study, we will examine several factors that may influence admissions decisions, such as academic performance, standardized test scores, extracurricular activities, and demographic characteristics.

This research has important implications for educational institutions looking to optimize their admissions process, as well as prospective students looking to improve their chances of gaining admission.

REFERENCES

[1] Chen, Y., Pan, C. C., Yang, G. K., & Bai, J. (2014, August). Intelligent decision system for accessing academic performance of candidates for early admission to university. In 2014 10th International Conference on Natural Computation (ICNC) (pp. 687-692). IEEE.

[2] Hasan, M., Ahmed, S., Abdullah, D. M., & Rahman, M. S. (2016, May). Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools. In 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV) (pp. 502-507). IEEE.

[3] Dietterich, T. G. (1997). Machine-learning research. AI magazine, 18(4),

[4] Yazdipour, S., & Taherian, N. (2017, December). Data Driven Decision Support to Fund Graduate Studies in Abroad Universities. In 2017 International Conference on Machine Learning and Data Science (MLDS) (pp. 44-50). IEEE.

[5] Waters, A., & Miikkulainen, R. (2014). Grade: Machine learning support for graduate admissions. AI Magazine, 35(1), 64-64.

[6] Mohan S. Acharya, Asfia Armaan, Aneeta S Antony . A Comparison of Regression Models for Prediction of Graduate Admissions, 2019 IEEE International Conference on Computational Intelligence in Data Science

[7] Wickramaratna, J., Holden, S., & Buxton, B. (2001, July). Performancedegradation in boosting. In International Workshop on Multiple Classifier Systems (pp. 11-21). Springer, Berlin, Heidelberg.

[8] R. Dong, H. Wang, and Z. Yu. The module of prediction of college entrance examination aspiration. In 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, pages 1559–1562, May 2012.

[9] P. Goyal, T. Kukreja, A. Agarwal, and N. Khanna. Narrowing awareness gap by using e-learning tools for counselling university entrants. In 2015 International Conference on Advances in Computer Engineering and Applications, pages 847–851, March 2015.

[10] Mahamudul Hasan, Shibbir Ahmed, Deen Md. Abdullah, and Md. Shamimur Rahman. Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools. In 5th International Conference on Informatics, Electronics and Vision (ICIEV), pages 502–507, 2016.

DB10

ORIGINALITY REPORT

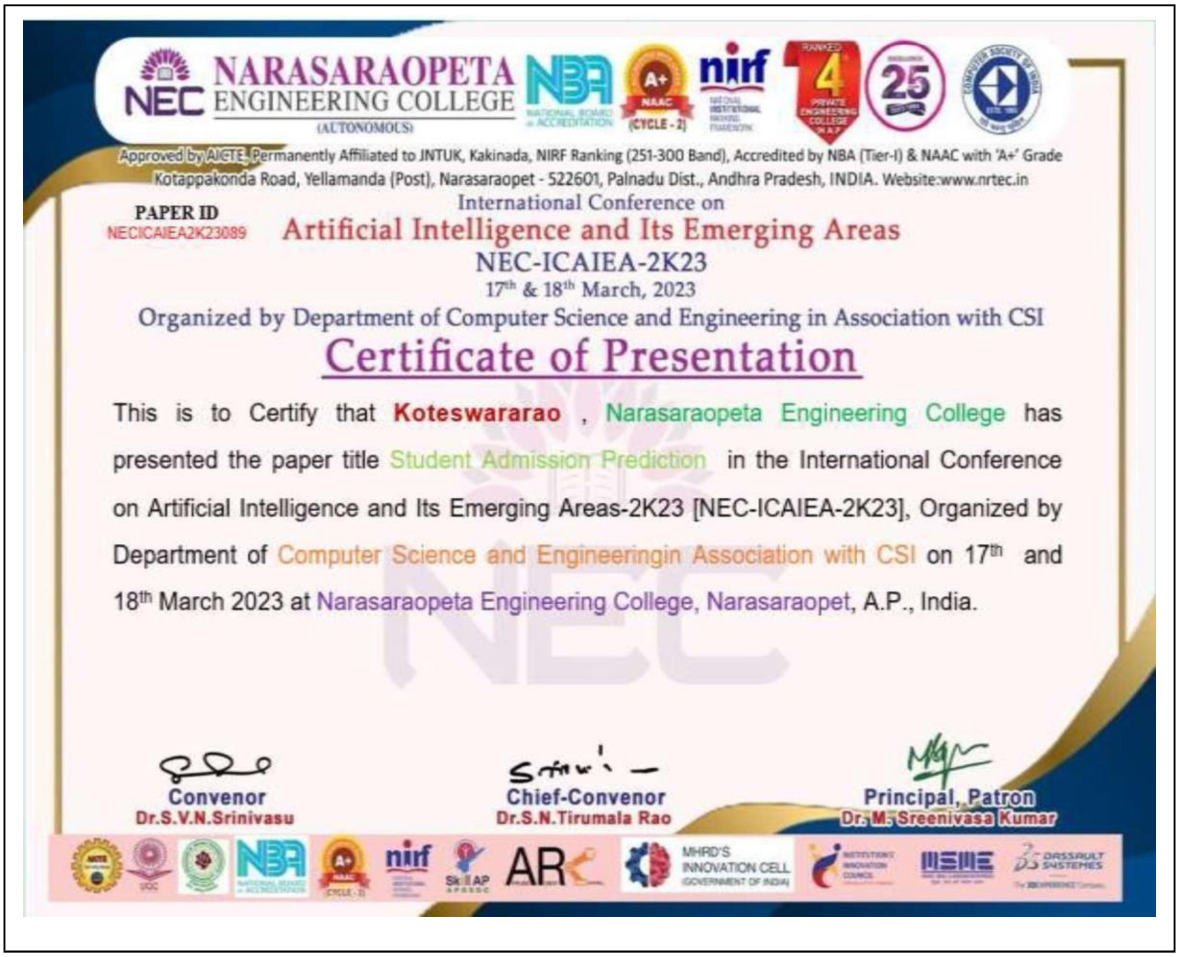
6 %	3%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Michael Onyema Edeh, Osamah Ibrahim Khalaf, Carlos Andrés Tavera, Sofiane Tayeb et al. "A Classification Algorithm-Based Hybrid Diabetes Prediction Model", Frontiers in Public Health, 2022 Publication	1 %
2	business-analytics-courses-in-hyd.blogspot.com Internet Source	1 %
3	Ab Shaqoor Nengroo, K.S. Kuppusamy. "Machine learning based heterogeneous web advertisements detection using a diverse feature set", Future Generation Computer Systems, 2018 Publication	1 %
4	www.ncbi.nlm.nih.gov Internet Source	1 %
5	Submitted to University of Essex Student Paper	1 %
6	Submitted to University of Northampton Student Paper	1 %

7	Submitted to Taylor’s Education Group Student Paper	1%
8	Submitted to University of Surrey Student Paper	1%
9	Submitted to Berlin School of Business and Innovation Student Paper	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		



NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23089

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023

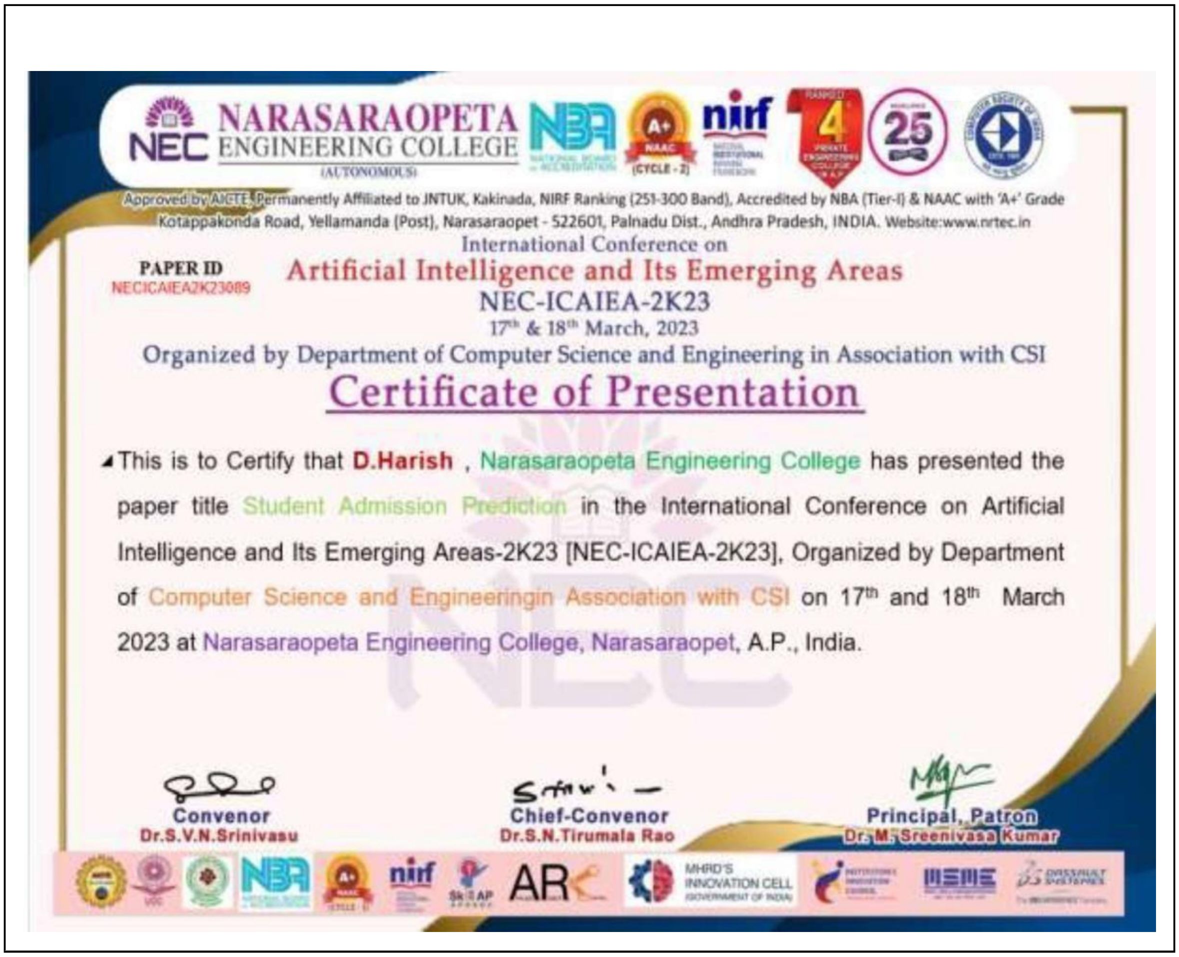
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Koteswararao** , Narasaraopeta Engineering College has presented the paper title Student Admission Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineering in Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor	**Chief-Convenor**	**Principal, Patron**
Dr.S.V.N.Srinivasu	**Dr.S.N.Tirumala Rao**	**Dr. M. Sreenivasa Kumar**

MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)



NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23089

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

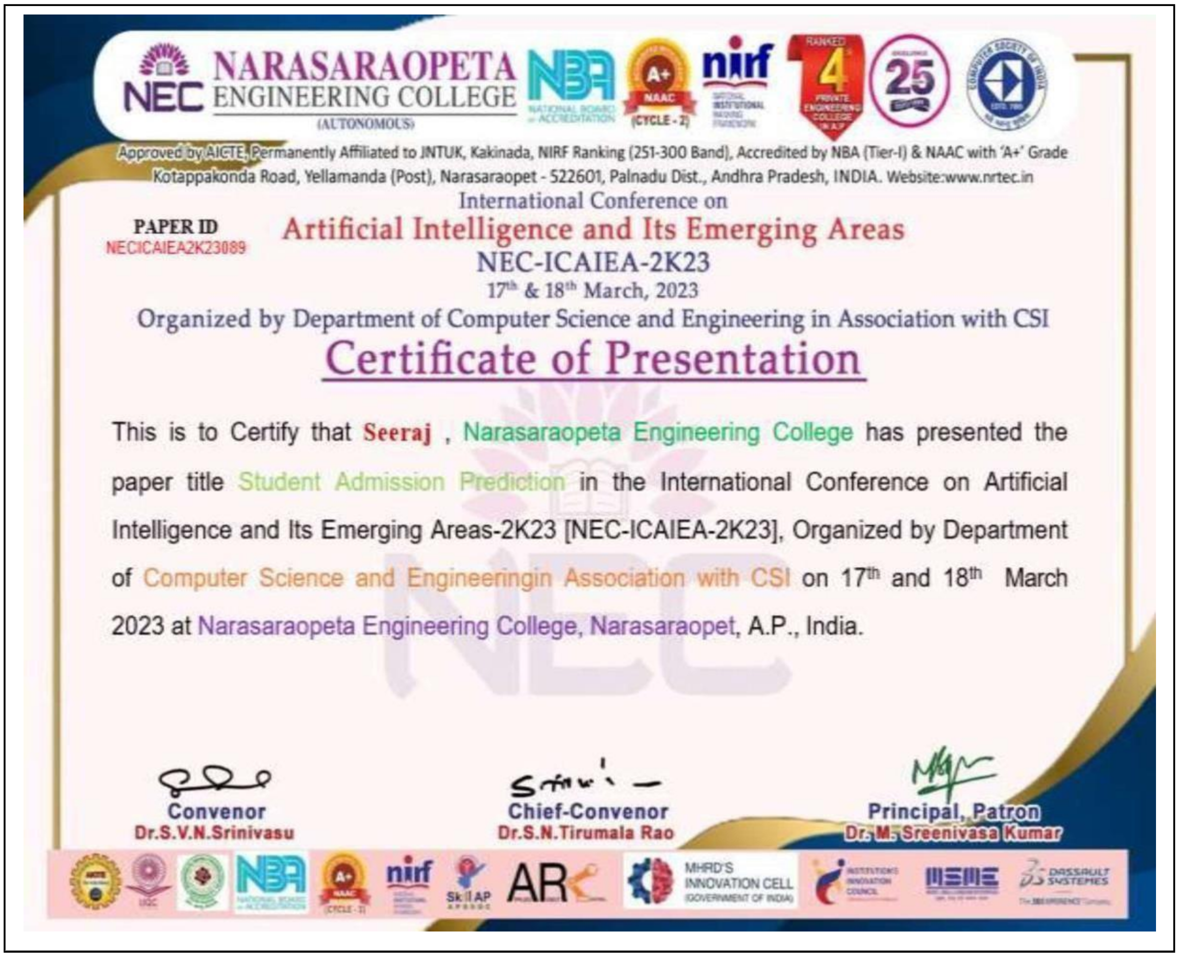
Certificate of Presentation

This is to Certify that **D.Harish** , Narasaraopeta Engineering College has presented the paper title Student Admission Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar



NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

PAPER ID
NECICAIEA2K23089

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

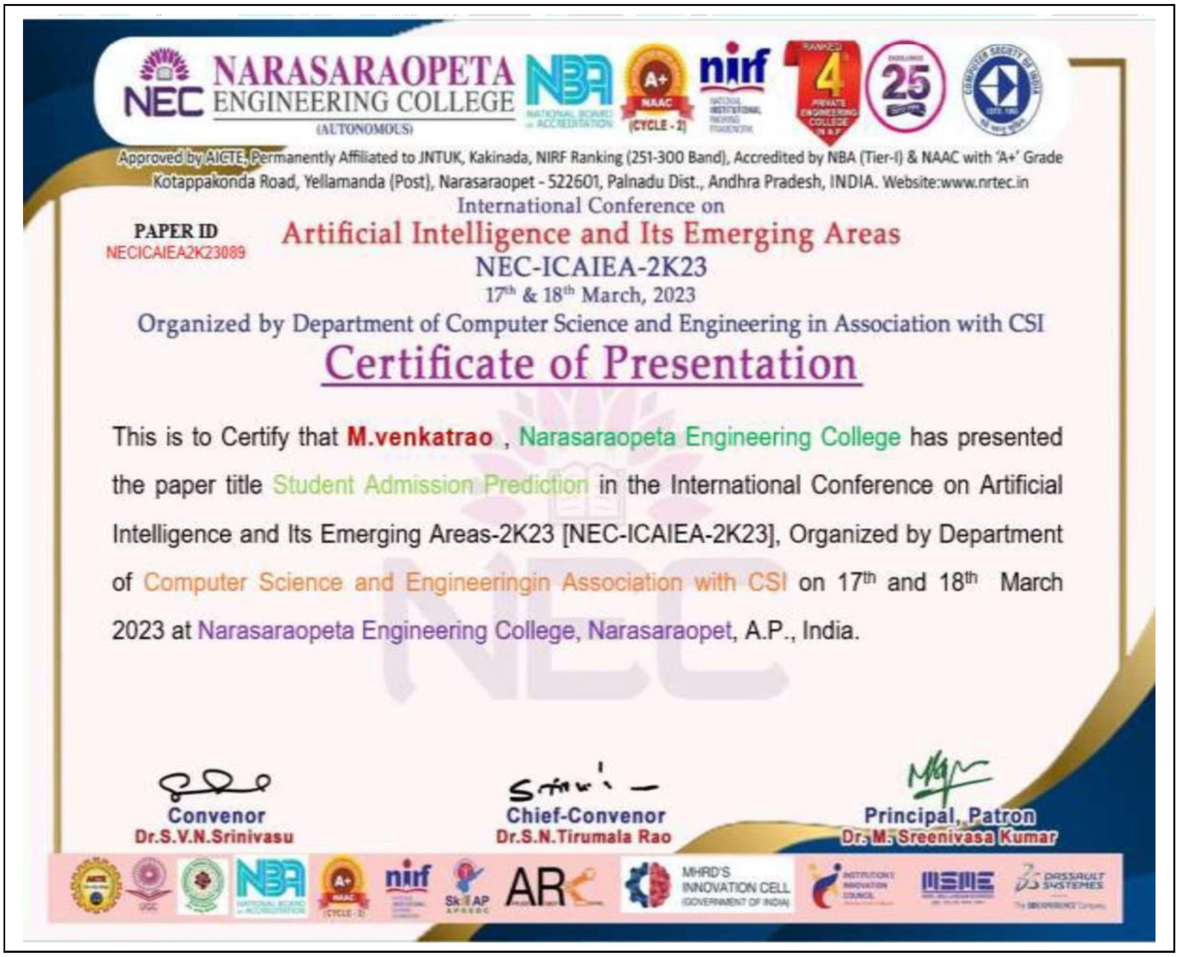
Certificate of Presentation

This is to Certify that **Seeraj** , Narasaraopeta Engineering College has presented the paper title Student Admission Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineering in Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar



NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23089

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **M.venkatrao** , Narasaraopeta Engineering College has presented the paper title Student Admission Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr.M.Sreenivasa Kumar