VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum- 590 018

A Project Phase – 2 Report

on

"Student's Placement Prediction and Recommendation Model using Machine Learning Algorithm"

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

Bachelor of Engineering in Computer Science and Engineering

submitted by

KARTIKEY RAI [1DB19CS066]

KUMAR KRISHNA [1DB19CS074]

PRANJALI [1DB19CS105]

Under the guidance of

Mrs. PRAKRUTHI S T

Assistant Professor,

Dept. of CSE.

Don Bosco Institute of Technology

Mysore Road, Kumbalgodu, Bengaluru-560074

2022-2023

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

DON BOSCO INSTITUTE OF TECHNOLOGY

Mysore Road, Kumbalgodu, Bengaluru-560074

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project Phase-II entitled **"Student's Placement Prediction and Recommendation Model using Machine Learning Algorithm"** is carried out by **Kartikey Rai [1DB19CS066], Kumar Krishna [1DB19CS074] and Pranjali [1DB19CS105]** are bonafide students of **Don Bosco Institute of Technology, Bangalore** in partial fulfillment for the award of the degree of **Bachelor of Engineering in Computer science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year **2022-23**. The project Phase -II report has been approved as it satisfies the academic requirements in respect of the Project Phase-II prescribed for the Bachelor of Engineering Degree.

Signature of Guide	**Signature of HOD**	**Signature of Principal**
Mrs. Prakruthi S T	**Dr. K B Shiva Kumar**	**Dr. B S Nagabhushana**
Assistant Professor	Head of Department	Principal
Dept of CSE	Dept of CSE	D.B.I.T

External Viva

Name of the Examiner — Signature with date

1) _____

_____

2) _____

_____

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

DON BOSCO INSTITUTE OF TECHNOLOGY

Mysore Road, Kumbalgodu, Bengaluru-560074

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Kartikey Rai [1DB19CS066], Kumar Krishna [1DB19CS074] and Pranjali [1DB19CS105]** students of 8th semester B.E, at the department of Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru declare that the project phase-II entitled "**Student's Placement Prediction and Recommendation Model using Machine Learning Algorithm**" has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree of Bachelor of Engineering in Computer Science and Engineering discipline of Visvesvaraya Technological University, Belagavi during the academic year **2022-2023**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree.

Place: Bangalore **KARTIKEY RAI [1DB19CS066]**

Date: **KUMAR KRISHNA [1DB19CS074]**

PRANJALI [1DB19CS105]

ACKNOWLEDGEMENT

The satisfaction and euphoria that successful completion of any project is incomplete without the mention of people who made it possible, whose constant guidance and encouragement made our effort fruitful.

First and foremost, we ought to pay our due regards to this institute, which provided us a platform and gave an opportunity to display our skills through the medium of project work. We express our heartfelt thanks to our beloved principal **Dr. B S Nagabhushana**, **Don Bosco Institute of Technology**, Bangalore for his encouragement and providing us with the infrastructure.

We express our deep sense of gratitude and thanks to **Dr. K B Shiva Kumar, Head of the Department, Computer Science and Engineering** for extending his valuable insight and suggestions offered during the course.

We express our acknowledgement to our project coordinator **Mrs**. **Sheeba S, Assistant Professor, Dept of CSE** for extending their direction and support during the completion of project phase-II.

It is our utmost pleasure to acknowledge the kind help extended by our guide **Mrs. Prakruthi S T**, **Assistant Professor**, Department of Computer Science and Engineering, for guidance and assistance which consequently resulted in getting the project phase II work completed successfully.

Last but not the least I would like to thank teaching and non-teaching staff for their cooperation extended during the completion of the project Phase-II.

KARTIKEY RAI [1DB19CS066]
KUMAR KRISHNA [1DB19CS074]
PRANJALI [1DB19CS105]

ABSTRACT

Student's Placement Prediction and Recommendation Model is developed to enhance the efficiency and effectiveness of the job placement process in various educational institutes. It is designed to calculate the possibility of a student being placed in a company using Machine Learning algorithms. The model takes various parameters like academic performance, technical skills, Internship experience, which can be used to assess the skill level of the student. Logistic regression is a supervised machine learning algorithm which is used to predict the probability of a student being placed in the particular company based on their skillset. The system utilizes historical placement data and real-time industry requirements to train the model. The personalized recommendations are given to students based on their skillset such that they can apply in other companies as well. Through its accurate predictions and personalized recommendations, the system streamlines the job placement process, benefiting both students and recruiters.

TABLE OF CONTENTS

LIST OF FIGURES

“Student’s Placement Prediction and Recommendation Model using Machine Learning Algorithm”

CHAPTER 1

INTRODUCTION

The number of educational institutions is rapidly increasing. Each higher education institution's goal is to place its pupils in a well-paying job through their career center. Increasing pupil positioning performance is one of the challenges facing today's institutions. Every university considers placements to be quite important. The presence of students on campus is used to gauge the college's fundamental development. Every student gets admitted to college depending on the proportion of placements at that institution.

The method is to predict and analyze the demand for institutions placement, which aids in the construction of institution and the enhancement of pupil placements. The possibility of students being placed in a company is predicted using classification algorithms such as Random Forest AL, SVM AL in this technique of selection.This model's main purpose is to forecast whether a student will be put in campus recruiting or not. As a result, statistics such as the total student population, backlogs, and credits are taken into account. The algorithms were developed using student data from the previous year.

During the student placement recruitment process, any company considers various aspects, such as the pupil's profile, which includes pupils scores in 10th, 12th, undergraduate. Other aspects of the selecting process include technical ability, coding knowledge, aptitude, and reasoning. The criteria, such as the student's personal profile, are not taken into account here. This concept focuses on controlled learning, which is a more clearly predictive assessment that may be used to make future predictions. When compared to other machine learning methods such as SVM algorithm, linear logistic regression, and decision tree, the random forest technique provides the greatest accuracy.

As the unemployment rate rose in urban and rural India, over 1.5 million people lost their jobs in August 2021. According to the Centre for Monitoring Indian Economy (CMIE), the unemployment rate rose from 6.96% in July 2021 to 8.32% in August 2021. Furthermore, the unemployment rate in urban areas is more than two percent greater than in rural areas. Youngsters in the 20-24 age group reported an unemployment rate of 37 percent, whereas the graduates amongst them reported a much alarming and higher unemployment rate of over 60 percent. These high rates of unemployment amongst graduates are primarily attributed to the unemployability of the youth. Employability is a prerequisite for employment. There is a gap

between the skills of graduates and the skills needed by industries. The education system mainly focuses on knowledge and written examinations rather than on practical skills. Colleges which help students develop skills required by the industry observe a rise in their placements.

The process of placing students in suitable job positions after completing their academic journey is a critical task for higher education institutions. The traditional manual placement systems often struggle to accurately predict the most fitting job placements for students based on their skills and preferences. To address this challenge, this research introduces a Student's Placement Prediction and Recommendation Model utilizing Machine Learning (ML) algorithms.

Machine Learning algorithms offer the potential to analyze diverse factors, including academic performance, skills, experience, and personal preferences, to predict the most suitable job positions for students. By leveraging historical placement data and real-time industry requirements, the ML models can be trained to make accurate predictions. The model incorporates feature engineering techniques to extract relevant features from the input data and employs supervised learning algorithms such as decision trees, random forests, or support vector machines to build robust predictive models.

To further enhance the recommendation process, collaborative filtering techniques are integrated into the model. These techniques provide personalized job recommendations to students based on their profiles and similarities with other successful candidates. The model takes into account not only technical skills but also soft skills and individual career aspirations, ensuring comprehensive and tailored recommendations aligned with the student's personal and professional goals.

The developed Student's Placement Prediction and Recommendation Model will be implemented as a web-based application, offering an intuitive and user-friendly interface for students, placement officers, and recruiters. The performance and accuracy of the model will be evaluated using real-world placement data, comparing it against existing manual placement processes. User feedback will be collected to assess the model's usability and effectiveness in facilitating the placement process.

The anticipated outcomes of this research include improved accuracy in placement predictions, reduced manual effort required for placements, enhanced student satisfaction, and increased engagement from recruiters. By harnessing the power of Machine Learning algorithms, this

model aims to bridge the gap between students and suitable job positions, facilitating successful career transitions and strengthening the overall placement process in higher education institutions.

CHAPTER 2
LITERATURE SURVEY

1. M. Siva surya, Dr. M.Sathish Kumar, Dr. D.Gandhimathi. "**Student Placement Prediction Using Supervised Machine Learning**". 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).Volume 108, July-2022.

This paper illustrarte an method for predicting the possibility of students being placed in a company is using classification algorithms such as Random Forest AL, SVM AL as a technique of selection.

2. Saha, Goutam, **"Applying logistic regression model to the examination results data"**, in Journal of Reliability and Statistical Studies 4, no.2(2021):1-13.

This paper illustrate the binary logistic regression model is used to analyze the school examination results (scores) of 1002 students. The analysis is performed on the basis of the independent variables viz. gender, medium of instruction, type of schools, category of schools, board of examinations and location of schools, where scores or marks are assumed to be dependent variables. The odds ratio analysis compares the scores obtained in two examinations viz. matriculation and higher secondary.

3. Senthil Kumar Thangavel, Divya Bharathi P and Abhijith Shankar. "**Student Placement Analyzer: A recommendation System Using Machine Learning"**, International Conference on advanced computing and Communication systems (ICACCS-2017), Jan 06-07,2017, Coimbatore, INDIA.

This paper presents a recommendation system that predicts whether the current student will be placed or not, if the student is placed the company is also predicted based on the data of previously placed students. Here we use two different machine learning classification algorithms, namely Naive Bayes Classifier and K Nearest Neighbors [KNN] algorithm. These algorithms independently predict the results and we then compare the efficiency of the algorithms, which is based on the dataset.

4. S.Taruna , Mrinal Pandey**, "An Empirical Analysis of Classification Techniques for Predicting Academic Performance"** in 2018 IEEE International Advance Computing Conference (IACC).

This paper compares five classification algorithms namely Decision Tree, Naïve Bayes, Naïve Bayes Tree, K-Nearest Neighbor and Bayesian Network algorithms for predicting students' grade particularly for engineering students. Bootstrap method is a resample function available in WEKA tool kit. The excellent results of this function can be seen through IBK, Decision Tree and Bayes Net algorithm. However the overall results of all four algorithms are good but the results of individual classes for Naïve Bayes and NB Tree is not sufficient enough for the individual class prediction particularly for this study. This paper also presents a comparative study of the previous work related to student's performance predictions.

5. Shreyas Harinath, Aksha Prasad, Suma H and Suraksha A. "**Student Placement Prediction using Machine Learning"**, International Research Journal of Engineering and Technology (IRJIET) Volume : 06 Issue: 04 April 2019.

This paper illustrate a study to predict the placement chances of the students using Decision Tree Learning, SCI-Kit learning which used two attributes as dataset namely CGPA and arrears which resulted in more time consuming for prediction and being not efficient.

6. Irene Treesa Jose, Daibin Raju, Jeebu Abraham Aniyankunju, Joel James, Mereen Thomas Vadakkel. "**Placement Prediction using Various Machine Learning Models and their Efficiency Comparison**". International Journal of Innovative Science and Research Technology. Volume 5, Issue 5, May – 2020.

According to this paper, the parameters in the dataset which are considered for the prediction are Quantitative scores, Logical Reasoning scores, Verbal scores, Programming scores, CGPA, No. of hackathons attended, No. of certifications and current backlogs number and based on these parameters the prediction is done using various algorithms, and then the efficiency of all these algorithms are compared.

7. Ishizue [7, Rousse, et al. "**Student placement and skill ranking predictors for programming classes using class attitude, psychological scales, and code metrics.**" Research and Practice in Technology Enhanced Learning 13.1 (2018): 1-20.

In this paper, a predictive model is designed which can predict the category of placements in which students are eligible by considering their past performance in academics and other curricular activities. The paper also provides real-time experimental results and findings along with performance measures used for model validation which helps in achieving the milestone of outcome-based education (OBE) in educational institutes as it is given utmost importance in present scenario to ensure better placement prospects in students, which would in turn help the students for carrier building.

8. Syed Tanveer Jishan, Raisul Islam Rashu, Naheena Haque and Rashedur M Rahman, **"Improving accuracy of students final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique"**, in Decision Analytics (2021) 2:1 DOI 10.1186/s40165-014- 0010-2(Springer Journel).

In this paper the data is preprocessed using a discretization method called the Optimal Equal Width Binning and an over-sampling technique known as the Synthetic Minority Over-Sampling (SMOTE) to improve the accuracy of the students' final grade prediction model for a particular course. In order to validate our method we have used data from a course offered at North South University, Bangladesh. The result obtained from the experiment gives a clear indication that the accuracy of the prediction model improves significantly when the discretization and over-sampling methods are applied.

CHAPTER 3

3.1 PROBLEM STATEMENT

To Develop a system to predict the placement of students using Machine Learning techniques and recommend job opportunities based on their skills.

3.2 OBJECTIVES

- Collect and preprocess a diverse dataset, incorporating parameters such as academic percentage, internship experience, backlogs, and other relevant factors to ensure comprehensive representation.
- Develop an accurate and efficient placement prediction and recommendation model by leveraging appropriate machine learning algorithms and techniques.
- Conduct performance analysis and evaluation of the model to demonstrate its accuracy and effectiveness in predicting suitable job placements.
- Present statistical results to validate the model's performance.

CHAPTER 4

4.1 EXISTING SYSTEM

The existing system for student placement incorporates ML algorithms; however, it may suffer from limitations in terms of the comprehensiveness of factors considered, accuracy of predictions, and user experience. The ML algorithms used in the existing system might not fully leverage the potential of advanced techniques or incorporate collaborative filtering for personalized recommendations. This may result in suboptimal placements, increased manual effort, and limited recruiter engagement.

The current system generally uses only a single parameter to judge whether a student can be placed or not during the campus placements. Generally the parameter used to judge the strengths of the student, is the academic performances during the first three years of engineering. But cracking an interview not only depends on the academic scores but also the awareness of student during the aptitude tests and interviews.

Also some Data Mining algorithms, while calculating the probability of a student getting selected, sometimes interpret the result having a probability of more than 100% which is not feasible and denotes a wrong interpretation to the student. Some algorithms give a negative probability which gives an wrong interpretation to the student. Judging the student only on the basis of academic grades is not enough. The other parameters like aptitude and technical tests should also be taken into consideration in order to determine the outcome for the student's future.

4.1 PROPOSED SYSTEM

The proposed Student Placement Prediction and Recommendation system aims to enhance the accuracy and efficiency of student placement using the Logistic Regression machine learning algorithm. The system will collect and preprocess a comprehensive dataset, including parameters such as academic performance, skills, experience, and preferences.

The logistic regression model will be built using the collected dataset to predict suitable job positions for students. The model will consider the various input factors and calculate the probabilities of students being placed in different job positions. This will provide valuable insights for placement officers and recruiters in making informed decisions.

In addition to placement prediction, the system will incorporate collaborative filtering techniques

to generate personalized job recommendations. By analyzing the profiles of successful candidates and identifying similarities with individual students, the system will suggest tailored job positions that align with their skills and preferences.

To evaluate the performance of the system, extensive testing and validation will be conducted. The accuracy of the logistic regression model will be measured by comparing its predictions with actual placement outcomes. Statistical analysis will be performed to assess the model's precision, recall, and F1-score. Furthermore, user feedback and satisfaction surveys will be collected to evaluate the usability and effectiveness of the system. Placement officers, recruiters, and students will be involved in providing insights and suggestions for system improvements.

Overall, the proposed system utilizing the Logistic Regression algorithm aims to enhance the accuracy of placement predictions and provide personalized recommendations for students. By leveraging machine learning techniques, the system will streamline the placement process, reduce manual effort, and improve the overall satisfaction of students and recruiters involved in the placement process.

CHAPTER 5

SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements:

1. Processor – i3 or ryzen 3 or higher.
2. RAM – 4gb or higher.
3. ROM – 240GB or Higher.

Software Requirements:

1. OS – Windows 10 or linux
2. IDE – Jupyter Notebook
3. Libraries – Pandas, Numpy,

CHAPTER 6

METHODOLOGY

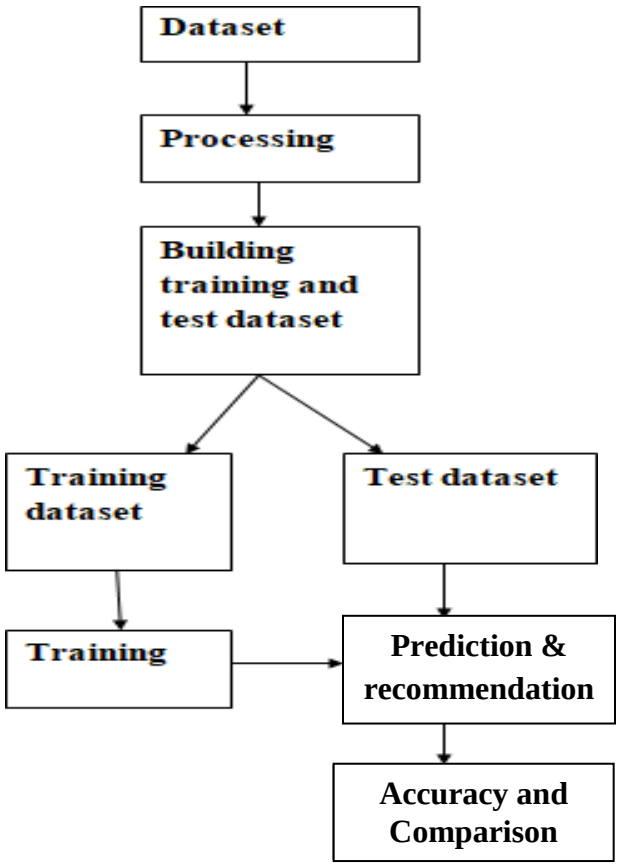


Fig 6.1:- Architectural Design

Data Collection and Preprocessing:

- Collect a comprehensive dataset containing relevant features such as academic performance, skills, experience, preferences, and placement outcomes.

- Preprocess the dataset by handling missing values, encoding categorical variables, and normalizing numerical features.

Feature Selection and Engineering:

- Perform exploratory data analysis to gain insights into the dataset.

- Select relevant features based on their correlation with placement outcomes and remove any redundant or irrelevant variables.

- Engineer new features if needed, such as aggregating skill levels or creating interaction terms.

Model Development:

- Split the dataset into training and testing sets for model evaluation.

- Apply logistic regression algorithm to build a binary classification model for placement prediction.

- Tune hyper parameters, such as regularization strength, to optimize model performance using techniques like cross-validation.

Performance Evaluation:

- Evaluate the logistic regression model's performance on the testing set using metrics such as accuracy, precision, recall, and F1-score.

- Generate a confusion matrix to visualize the model's predictive performance and identify any class imbalances or misclassifications.

Collaborative Filtering for Recommendation:

- Implement collaborative filtering techniques, such as item-based or user-based filtering, to provide personalized job recommendations.

- Calculate similarity scores between student profiles to identify similar successful candidates.

- Recommend job positions based on the preferences and similarities of each student.

Logistic Regression

- Logistic regression is a popular statistical algorithm used for binary classification tasks, where the goal is to predict a binary outcome or assign a binary label to a given input. It is a supervised learning algorithm that is widely used in various fields, including machine learning, statistics, and social sciences.
- The key idea behind logistic regression is to model the relationship between the input variables (or features) and the probability of a particular outcome using the logistic function, also known as the sigmoid function. The sigmoid function maps any real-valued number to a value between 0 and 1, making it suitable for representing probabilities.
- The logistic regression model assumes a linear relationship between the input features and the log-odds (also known as the logit) of the outcome. Mathematically, the logistic regression model can be represented as:
- $p(y=1 \mid x) = 1 / (1 + \exp(-z))$
- where $p(y=1 \mid x)$ is the probability of the positive outcome given the input features x, and z is the linear combination of the input features and their corresponding coefficients:
- $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$
- Here, β_0, β_1, β_2, ..., β_p are the coefficients (or weights) learned by the logistic regression algorithm, and x_1, x_2, ..., x_p are the input features.
- To estimate the coefficients, logistic regression typically uses the maximum likelihood estimation (MLE) method. The goal is to find the values of the coefficients that maximize the likelihood of observing the given data. This optimization problem is typically solved using numerical optimization algorithms such as gradient descent or Newton's method.
- During the training phase, logistic regression learns the optimal coefficients by iteratively adjusting them to minimize a loss function, such as the binary cross-entropy loss, which measures the dissimilarity between the predicted probabilities and the actual labels. The optimization process continues until convergence or a predefined stopping criterion is met.
- Once the logistic regression model is trained, it can be used to make predictions on new, unseen data by calculating the probability of the positive outcome using the learned

coefficients and the input features. A threshold can be applied to these probabilities to make binary predictions (e.g., if the probability is above 0.5, predict the positive class; otherwise, predict the negative class).

- Logistic regression is a simple and interpretable algorithm that works well when the relationship between the input features and the outcome is approximately linear. However, it may not perform well in cases where the relationship is highly non-linear or when there are complex interactions between the features. In such cases, more sophisticated models like decision trees or neural networks may be more suitable.

CHAPTER 7

IMPLEMENTATION

In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv('Projectdata2.csv')
```

In [3]:
```
import warnings
```

In [4]:
```
warnings.filterwarnings('ignore')
```

In [5]:
```
data.head()
```

Out[5]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
0	75	84	66	Yes	91	CSE	Placed
1	77	65	88	Yes	86	CSE	Placed
2	84	87	67	Yes	98	ISE	Placed
3	90	97	88	Yes	83	ISE	Placed
4	98	66	68	Yes	97	ECE	Placed

In [6]:
```
data.tail()
```

Out[6]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
4994	40	47	10	No	38	MECH	Not Placed
4995	50	58	22	No	53	MECH	Not Placed
4996	19	49	57	No	61	EEE	Not Placed
4997	26	32	12	No	20	CIVIL	Not Placed
4998	25	24	35	No	34	MECH	Not Placed

In [7]:
```
data.shape
```

Out[7]:
```
(4999, 7)
```

In [8]:
```
print("Number of Rows",data.shape[0])
print("Number of Columns",data.shape[1])
```

```
Number of Rows 4999
Number of Columns 7
```

In [9]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4999 entries, 0 to 4998
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   ssc_p   4999 non-null   int64
 1   hsc_p   4999 non-null   int64
 2   BE_p    4999 non-null   int64
 3   Intern  4999 non-null   object
 4   Test_p  4999 non-null   int64
 5   Branch  4999 non-null   object
 6   Status  4999 non-null   object
dtypes: int64(4), object(3)
memory usage: 273.5+ KB
```

In [10]:
```
data.isnull().sum()
```

Out[10]:
```
ssc_p     0
hsc_p     0
BE_p      0
Intern    0
Test_p    0
Branch    0
Status    0
dtype: int64
```

In [11]:
```
data.describe()
```

Out[11]:

	ssc_p	hsc_p	BE_p	Test_p
count	4999.000000	4999.000000	4999.000000	4999.000000
mean	55.743549	58.055611	58.071014	62.079016
std	27.410363	25.826121	26.186643	26.720231
min	0.000000	0.000000	0.000000	10.000000
25%	31.000000	36.000000	35.500000	39.000000
50%	61.000000	61.000000	61.000000	71.000000
75%	80.000000	80.000000	81.000000	85.000000
max	100.000000	100.000000	100.000000	99.000000

In [12]:
```
data.columns
```

Out[12]:
```
Index(['ssc_p', 'hsc_p', 'BE_p', 'Intern', 'Test_p', 'Branch', 'Status'], dtype
='object')
```

In [13]:
```
data['Status'].unique()
```

Out[13]:
```
array(['Placed', 'Not Placed'], dtype=object)
```

In [14]:
```
data['Status'].value_counts()
```

Out[14]:
```
Status
Placed        2600
Not Placed    2399
Name: count, dtype: int64
```

In [15]:
```
data.columns
```

Out[15]:
```
Index(['ssc_p', 'hsc_p', 'BE_p', 'Intern', 'Test_p', 'Branch', 'Status'], dtype
='object')
```

In [16]:
```
data.head()
```

Out[16]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
0	75	84	66	Yes	91	CSE	Placed
1	77	65	88	Yes	86	CSE	Placed
2	84	87	67	Yes	98	ISE	Placed
3	90	97	88	Yes	83	ISE	Placed
4	98	66	68	Yes	97	ECE	Placed

In [17]:
```
data['Branch'].unique()
```

Out[17]:
```
array(['CSE', 'ISE', 'ECE', 'CIVIL', 'EEE', 'MECH'], dtype=object)
```

In [18]:
```
data['Branch'] = data['Branch'].map({'CSE':5,'ISE':4,'ECE':3,'EEE':2,'MECH':1,'C
```

In [19]:
```
data.head()
```

Out[19]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
0	75	84	66	Yes	91	5	Placed
1	77	65	88	Yes	86	5	Placed
2	84	87	67	Yes	98	4	Placed
3	90	97	88	Yes	83	4	Placed
4	98	66	68	Yes	97	3	Placed

In [20]:
```
data['Intern'].unique()
```

Out[20]:
```
array(['Yes', 'No'], dtype=object)
```

In [21]:
```
data['Intern'] = data['Intern'].map({'Yes':1,'No':0})
```

In [22]:
```
data.head(2)
```

Out[22]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
0	75	84	66	1	91	5	Placed
1	77	65	88	1	86	5	Placed

In [23]:
```
data['Status'].unique()
```

Out[23]:
```
array(['Placed', 'Not Placed'], dtype=object)
```

In [24]:
```
data['Status'] = data['Status'].map({'Placed':1,'Not Placed':0})
```

In [25]:
```
data.head()
```

Out[25]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
0	75	84	66	1	91	5	1
1	77	65	88	1	86	5	1
2	84	87	67	1	98	4	1
3	90	97	88	1	83	4	1
4	98	66	68	1	97	3	1

In [26]:
```
data.columns
```

Out[26]:
```
Index(['ssc_p', 'hsc_p', 'BE_p', 'Intern', 'Test_p', 'Branch', 'Status'], dtype
='object')
```

In [27]:
```
X = data.drop('Status',axis=1)
y= data['Status']
```

In [28]:
```
y
```

Out[28]:
```
0       1
1       1
2       1
3       1
4       1
      ..
4994    0
4995    0
4996    0
4997    0
4998    0
Name: Status, Length: 4999, dtype: int64
```

In [29]:
```
from sklearn.model_selection import train_test_split
```

In [30]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20,random_state=4
```

In [31]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

In [32]:
```
lr = LogisticRegression()
lr.fit(X_train,y_train)

svm = svm.SVC()
svm.fit(X_train,y_train)
```

```
knn=KNeighborsClassifier()
knn.fit(X_train,y_train)

dt=DecisionTreeClassifier()
dt.fit(X_train,y_train)

rf=RandomForestClassifier()
rf.fit(X_train,y_train)
```

Out[32]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [33]:
```
y_pred1 = lr.predict(X_test)
y_pred2 = svm.predict(X_test)
y_pred3 = knn.predict(X_test)
y_pred4 = dt.predict(X_test)
y_pred5 = rf.predict(X_test)
```

In [34]:
```
import matplotlib as plt
from sklearn.metrics import accuracy_score
```

In [35]:
```
score1=accuracy_score(y_test,y_pred1)
score2=accuracy_score(y_test,y_pred2)
score3=accuracy_score(y_test,y_pred3)
score4=accuracy_score(y_test,y_pred4)
score5=accuracy_score(y_test,y_pred5)
```

In [36]:
```
print(score1,score2,score3,score4,score5)
```

```
0.999 0.999 0.999 0.999 0.999
```

In [37]:
```
final_data = pd.DataFrame({'Models':['LR','SVC','KNN','DT','RF'],
             'ACC':[score1*100,
                    score2*100,
                    score3*100,
                    score4*100,
                    score5*100]})
```

In [38]:
```
final_data
```

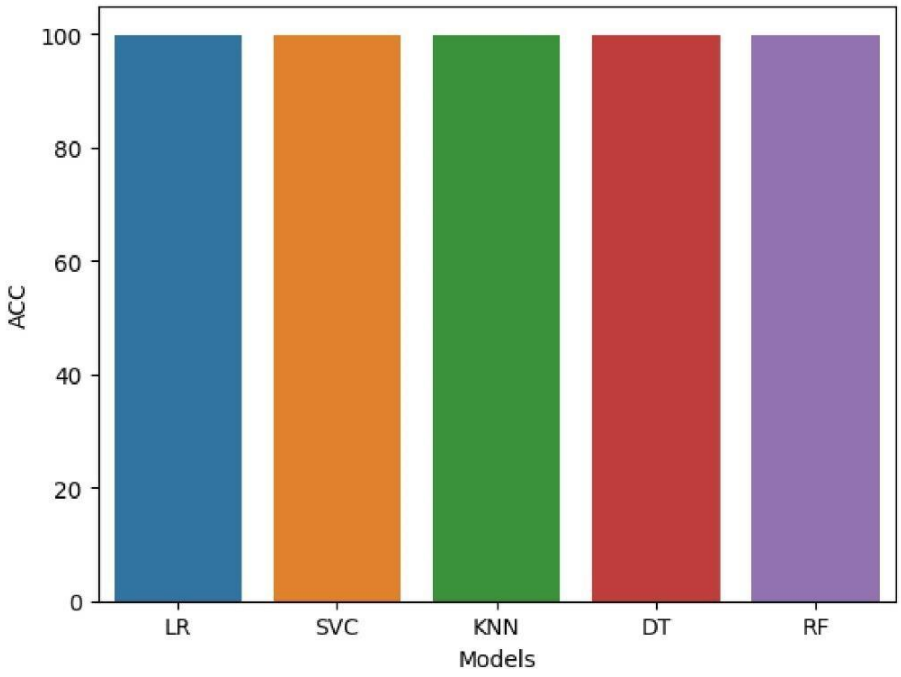
Out[38]:

	Models	ACC
0	LR	99.9
1	SVC	99.9
2	KNN	99.9
3	DT	99.9
4	RF	99.9

In [39]:
```
import seaborn as sns
```

In [40]:
```
sns.barplot(x='Models', y='ACC', data=final_data)
```

Out[40]: <Axes: xlabel='Models', ylabel='ACC'>



In [41]:
```
corr=data.corr()
```

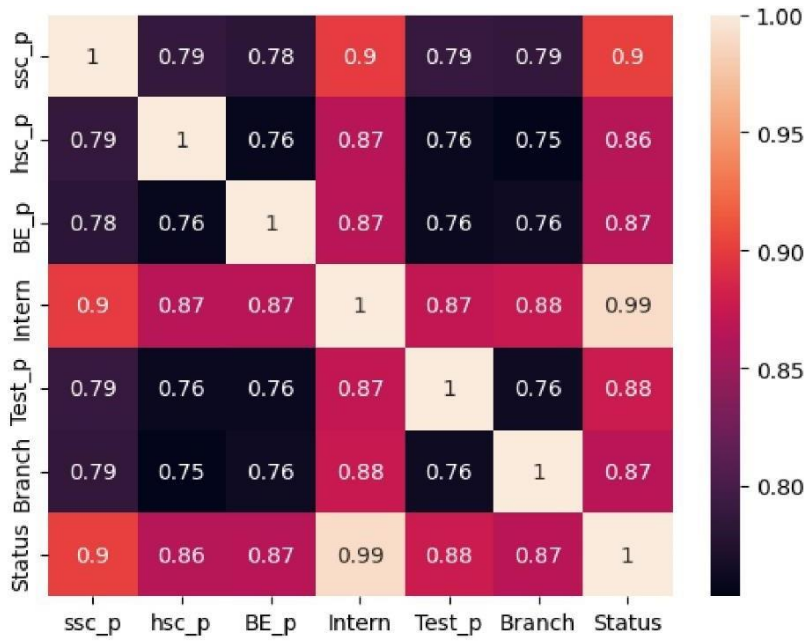
In [42]:
```
corr.style.background_gradient(cmap=\
                               'coolwarm')
```

Out[42]:

	ssc_p	hsc_p	BE_p	Intern	Test_p	Branch	Status
ssc_p	1.000000	0.787661	0.781690	0.899713	0.789631	0.786782	0.902262
hsc_p	0.787661	1.000000	0.757877	0.865450	0.759019	0.753172	0.864478
BE_p	0.781690	0.757877	1.000000	0.866594	0.760581	0.762517	0.865528
Intern	0.899713	0.865450	0.866594	1.000000	0.871247	0.875085	0.990383
Test_p	0.789631	0.759019	0.760581	0.871247	1.000000	0.762531	0.876901
Branch	0.786782	0.753172	0.762517	0.875085	0.762531	1.000000	0.866398
Status	0.902262	0.864478	0.865528	0.990383	0.876901	0.866398	1.000000

In [43]:
```
corr = data.corr()
sns.heatmap(corr, annot=True)
```

Out[43]: <Axes: >



In [44]:
```
new_data = pd.DataFrame({
    'ssc_p':70.00,
    'hsc_p':80.00,
    'BE_p':60.00,
    'Intern':1,
    'Test_p':70,
    'Branch':5,
},index=[0])
```

In [45]:
```
lr= LogisticRegression()
lr.fit(X,y)
```

Out[45]:
```
▾ LogisticRegression
LogisticRegression()
```

In [46]:
```
p=lr.predict(new_data)
prob=lr.predict_proba(new_data)
if p==1:
    print('Placed')
    print(f"You will be placed with probability of {prob[0][1]:.2f}")
else:
    print("Can't be placed")
```

```
Placed
You will be placed with probability of 1.00
```

In [47]:
```
prob
```

Out[47]:
```
array([[1.29356212e-06, 9.99998706e-01]])
```

In [48]:
```
import joblib
```

In [49]:
```
joblib.dump(lr,'model_final')
```

Out[49]:
```
['model_final']
```

In [50]:
```
model = joblib.load('model_final')
```

In [51]:
```
model.predict(new_data)
```

Out[51]:
```
array([1], dtype=int64)
```

In [52]:
```
# Recommendation Part
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans
```

In [53]:
```
df=pd.read_csv("company.csv")
```

In [54]:
```
df.head()
```

Out[54]:

	Company	ssc_p	hsc_p	BE_p	Test_p
0	Microsoft	75	75	85	85
1	Wipro	60	70	70	75
2	TCS	60	65	70	70
3	Apple	80	80	75	90
4	Google	80	80	85	90

In [55]:
```
df.tail()
```

Out[55]:

	Company	ssc_p	hsc_p	BE_p	Test_p
25	IBM	60	60	70	75
26	Mphasis	60	60	65	65
27	Dropbox	65	65	70	75
28	Flipkart	70	70	75	70
29	PayPal	70	70	70	85

In [56]:
```
df.shape
```

Out[56]:
```
(30, 5)
```

In [57]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Company  30 non-null     object
 1   ssc_p    30 non-null     int64
 2   hsc_p    30 non-null     int64
 3   BE_p     30 non-null     int64
 4   Test_p   30 non-null     int64
dtypes: int64(4), object(1)
memory usage: 1.3+ KB
```

In [58]:
```
print(df.Company.head())
```

```
0    Microsoft
1        Wipro
2          TCS
3        Apple
4       Google
Name: Company, dtype: object
```

In [59]:
```
l2=df.iloc[:,-1:-5:-1]
l2
```

Out[59]:

	Test_p	BE_p	hsc_p	ssc_p
0	85	85	75	75
1	75	70	70	60
2	70	70	65	60
3	90	75	80	80
4	90	85	80	80
5	70	75	75	70
6	80	75	75	70
7	70	70	65	60
8	75	75	60	60
9	80	75	75	70
10	75	70	60	60
11	75	70	65	65
12	75	75	65	60
13	70	75	70	70
14	85	80	75	75
15	80	70	70	70
16	80	70	60	60
17	75	65	60	60
18	80	75	75	70
19	75	75	75	70
20	85	75	70	70
21	75	80	75	75
22	70	70	60	60
23	70	70	65	65
24	75	75	70	70
25	75	70	60	60
26	65	65	60	60
27	75	70	65	65
28	70	75	70	70
29	85	70	70	70

In [60]:
```
kmeans=KMeans(5)
kmeans.fit(l2)
identified_clusters=kmeans.fit_predict(l2)
```

```
identified_clusters=list(identified_clusters)
identified_clusters
```

Out[60]:
```
[2,
 3,
 3,
 2,
 2,
 0,
 4,
 3,
 1,
 4,
 1,
 3,
 3,
 0,
 2,
 4,
 1,
 1,
 4,
 0,
 4,
 0,
 3,
 3,
 0,
 1,
 3,
 3,
 0,
 4]
```

In [61]:
```
df['loc_cluster']=identified_clusters
df
```

Out[61]:

	Company	ssc_p	hsc_p	BE_p	Test_p	loc_cluster
0	Microsoft	75	75	85	85	2
1	Wipro	60	70	70	75	3
2	TCS	60	65	70	70	3
3	Apple	80	80	75	90	2
4	Google	80	80	85	90	2
5	Upgrad	70	75	75	70	0
6	Dell	70	75	75	80	4
7	Cognizant	60	65	70	70	3
8	Capegemini	60	60	75	75	1
9	Amazon	70	75	75	80	4
10	Upgrad	60	60	70	75	1
11	HP	65	65	70	75	3
12	HCL	60	65	75	75	3
13	EY	70	70	75	70	0
14	Juspay	75	75	80	85	2
15	MindTree	70	70	70	80	4
16	Tech Mahindra	60	60	70	80	1
17	Infosys	60	60	65	75	1
18	Adobe	70	75	75	80	4
19	Intel	70	75	75	75	0
20	Oracle	70	70	75	85	4
21	Flipkart	75	75	80	75	0
22	Deloitte	60	60	70	70	3
23	Accenture	65	65	70	70	3
24	L&T	70	70	75	75	0
25	IBM	60	60	70	75	1
26	Mphasis	60	60	65	65	3
27	Dropbox	65	65	70	75	3
28	Flipkart	70	70	75	70	0
29	PayPal	70	70	70	85	4

In [ ]:
```
# GUI Part
from tkinter import *
import joblib
```

```
import numpy as np
from sklearn import *
import tkinter.font as font
import pandas as pd
from tkinter import messagebox as tkMessageBox
import tkinter as tk


def clicked():
    tkMessageBox.showerror("Warning!", "Please Enter Correct Values\nEnter perce

root = Tk()
root.geometry('400x200+300+200')
l=tk.Label(root, text="Final Year Project", relief='ridge', border=10, font="Tim
l.pack(side=TOP)
l1=tk.Label(root, text="Team Members:\n1. Kartikey Rai (1DB19CS066)\n2. Kumar Kr
l1.pack(pady=0)
button = Button(root, text = "Click Here for Instructions", command = clicked, b
button.pack()
button.place(x=115, y=150)
root.mainloop()


def show_entry_fields():

# validation
    p1 =int(e1.get())
    p2= int(e2.get())
    p3= int(e3.get())
    p6= int(e6.get())
    textbox=Text(master)
    if (p1<0 or p2<0 or p3<0 or p6<0):
        tk.messagebox.showerror('Invalid Message Alert',"Negative values not all
    elif (p1>100 or p2>100 or p3> 100 or p6>100):
        tk.messagebox.showerror('Invalid Message Alert',"Enter Correct Values!")
    else:
        exit

    text = clicked1.get()
    if text == "CSE":
        p4=5
        print(p4)
    elif text == "ISE":
        p4=4
        print(p4)
    elif text == "ECE":
        p4=3
        print(p4)
    elif text == "EEE":
        p4=2
        print(p4)
    elif text == "MECH":
        p4=1
        print(p4)
    else:
        p4=0
        print(p4)


    text = clicked2.get()
```

```
    if text == "Yes":
        p5=1
        print(p5)
    else:
        p5=0
        print(p6)

    text1=clicked3.get()


#Prediction Part
    model = joblib.load('model_final')
    new_data = pd.DataFrame({
    'ssc_p':p1,
    'hsc_p':p2,
    'BE_p':p3,
    'Intern':p5,
    'Test_p':p6,
    'Branch':p4,
},index=[0])
    result=model.predict(new_data)
    result1=model.predict_proba(new_data)

    if result[0] == 0:
        Label(master, text="Not Eligible",font=("Ariel",15),fg='navy',bg='light
    else:
        Label(master, text="Student Will be Placed With Probability of",font=("A
        Label(master, text=round(result1[0][1],2)*100,font=("Arial", 15), bg='li
        Label(master, text="Percent in",font=("Arial", 15), bg='light sea green'
        Label(master, text=text1,font=("Ariel, 15"), bg='light sea green', fg='r
        Label(master, text="Other Recommended Companies",font=("Times", 10),bg='

#Recommendation Part
        input_name=clicked3.get()
        cluster=df.loc[df['Company']==input_name,'loc_cluster']
        cluster=cluster.iloc[0]
        cluster
        company=df.loc[df['loc_cluster']==cluster,'Company']
        company
        for c in range(len(company)):
            if company.iloc[c]==input_name:
                continue
            else:
                textbox.insert(END, company.iloc[c]+ '\n')
                textbox.configure(bg='light sea green',fg='white', width=15, hei
                textbox.grid(row=39)


master = tk.Tk()
master.title("Campus Placement Prediction and Recommendation System")
master['background'] = 'light sea green'


label = Label(master, text = "Placement Prediction and Recommendations\nUsing Ma
                          , relief='ridge', border=10, bg = "pink", fg = "red2",
                               .grid(row=0,columnspan=2)

Label(master, text="Secondary Education percentage- 10th Grade",font=("Times", 1
Label(master, text="Higher Secondary Education percentage- 12th Grade",font=("Ti
Label(master, text="BE Percentage",font=("Times", 18),bg='light sea green',fg='w
```

```
Label(master, text="Branch",font=("Times", 18),bg='light sea green',fg='white').
Label(master, text="Internship",font=("Times", 18),bg='light sea green',fg='whit
Label(master, text="Enter Assesment Percentage",font=("Times", 18),bg='light sea
Label(master, text="Select your desired company",font=("Times", 18),bg='light se


clicked1 = StringVar()
options1 = ["CSE","ISE","ECE","EEE","MECH","CIVIL"]

clicked2 = StringVar()
options2 = ["Yes","No"]

clicked3 = StringVar()
options3 = ["Amazon","Google","TCS","Wipro","IBM","Apple","Flipkart","Accenture'

e1=StringVar()
e1 = Entry(master)
e1.grid(row=1, column=1)

e2=StringVar()
e2 = Entry(master)
e2.grid(row=2, column=1)

e3=StringVar()
e3 = Entry(master)
e3.grid(row=3, column=1)

e4 = OptionMenu(master , clicked1 , *options1 )
e4.configure(width=13,fg="white",bg="slate grey")
e4.grid(row=4, column=1)
clicked1.set("Select")

e5 = OptionMenu(master , clicked2 , *options2)
e5.configure(width=13,fg="white",bg="slate grey")
e5.grid(row=5, column=1)
clicked2.set("Select")

e6 = Entry(master)
e6.grid(row=6, column=1)

e7 = OptionMenu(master , clicked3 , *options3 )
e7.configure(width=13,fg="white",bg="slate gray")
e7.grid(row=7, column=1)
clicked3.set("Select")


buttonFont = font.Font(family='Helvetica', size=16, weight='bold')
Button(master, text='Predict',height= 1, width=8,activebackground='#00ff00',font
master.mainloop()
```

In [ ]:

CHAPTER 8

RESULTS

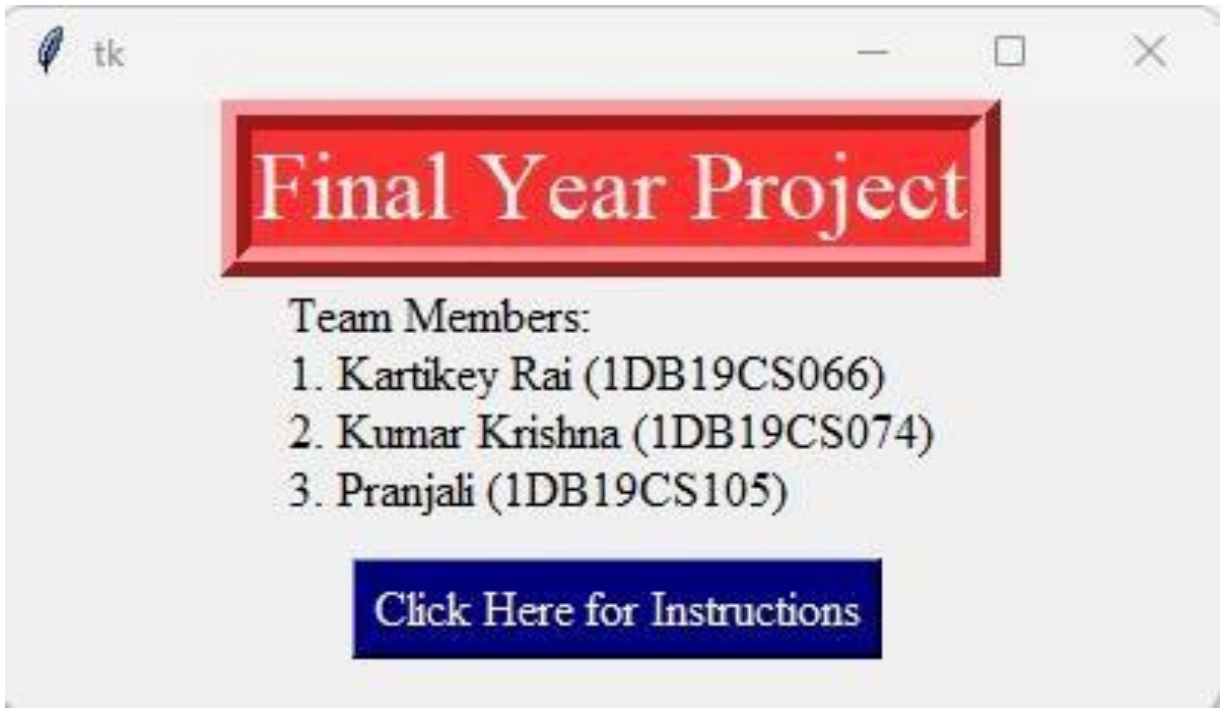


Fig 8.1:- Home Page

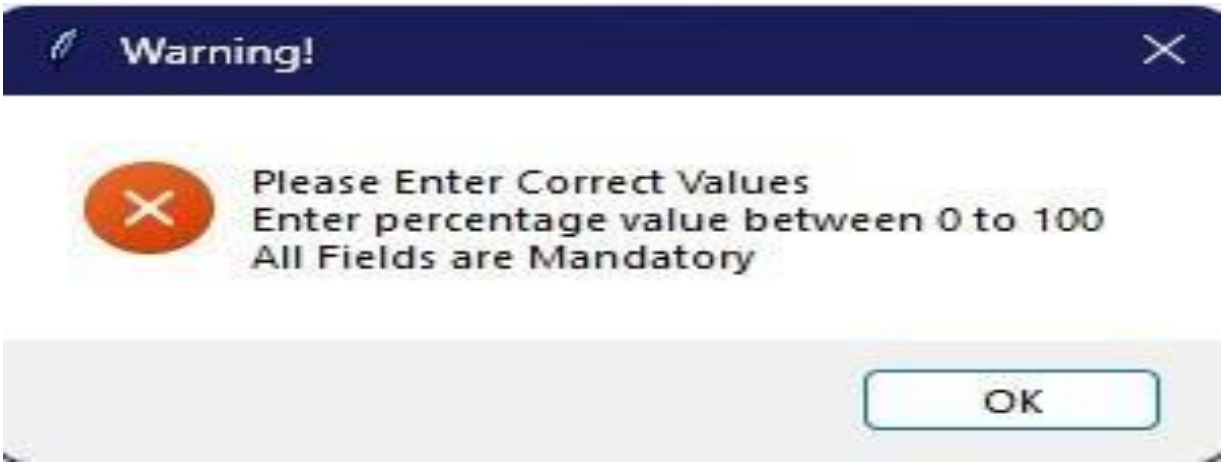


Fig 8.2 :- Instruction window

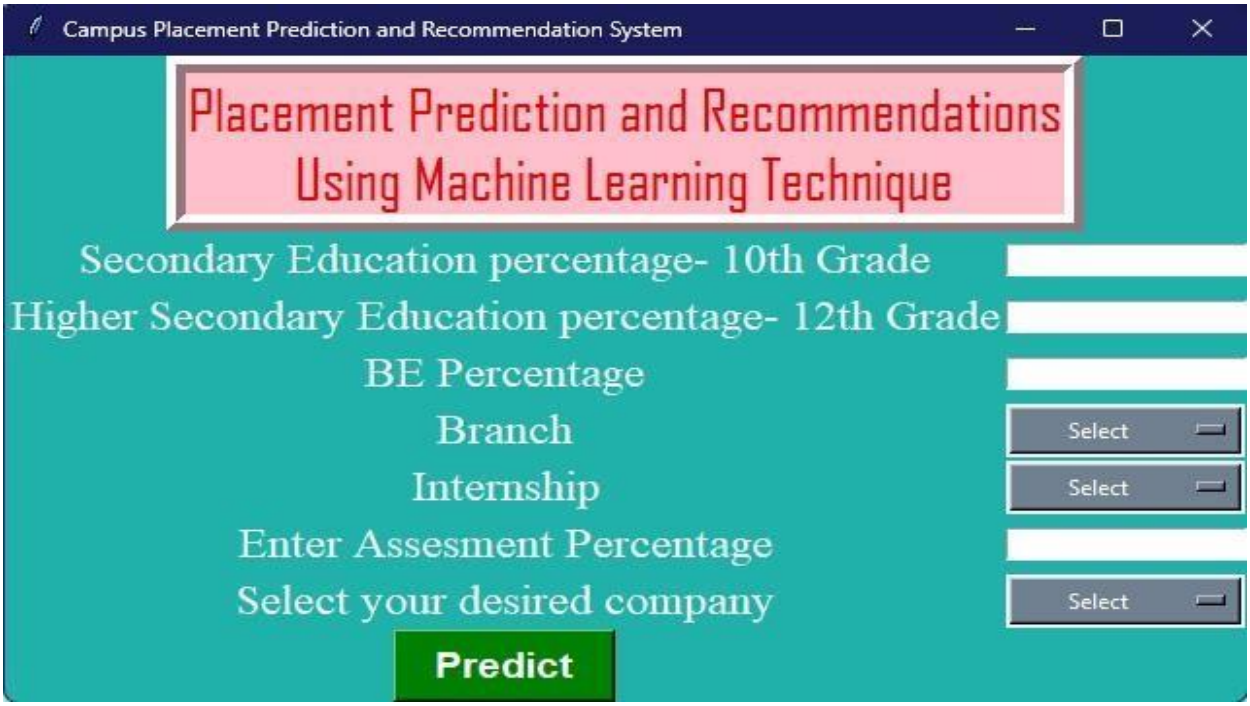


Fig 8.3 :- Placement prediction and recommendation window

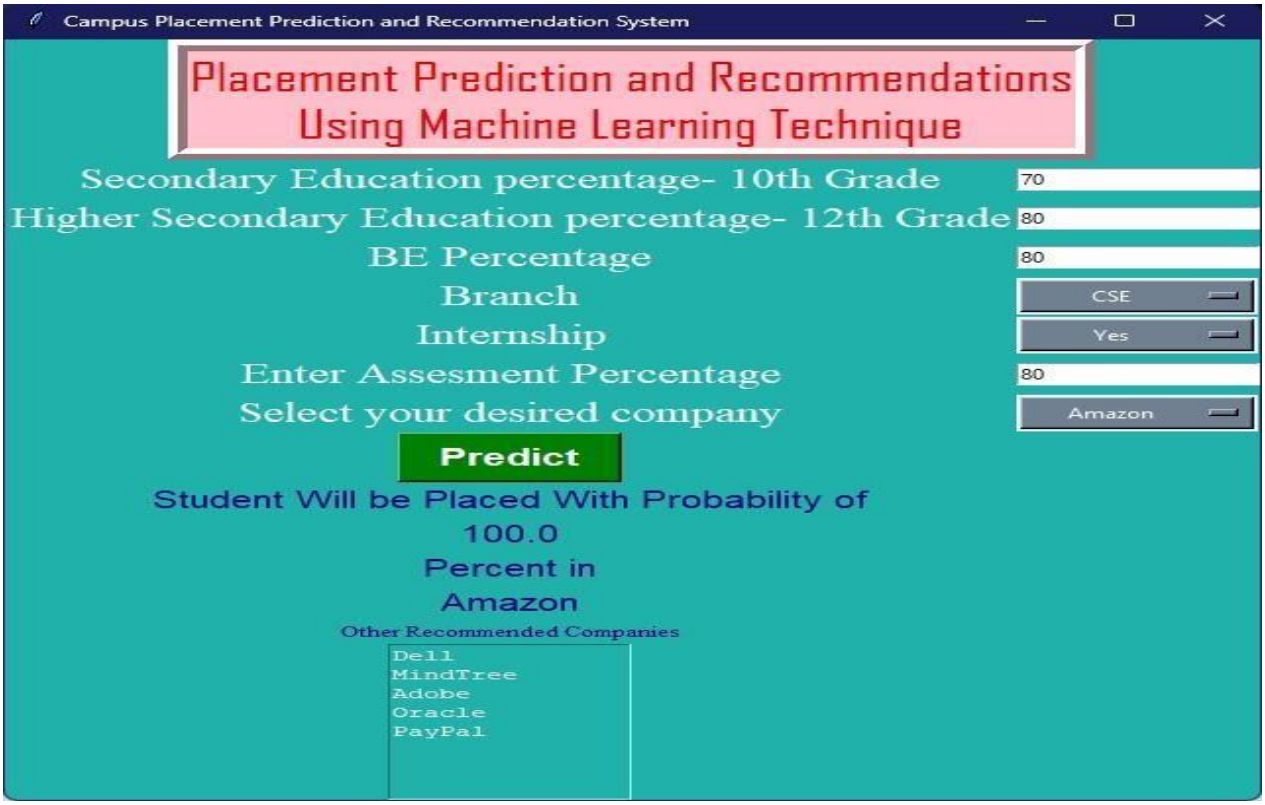


Fig 8.4 :- Successful prediction with 100% probability and some recommended companies

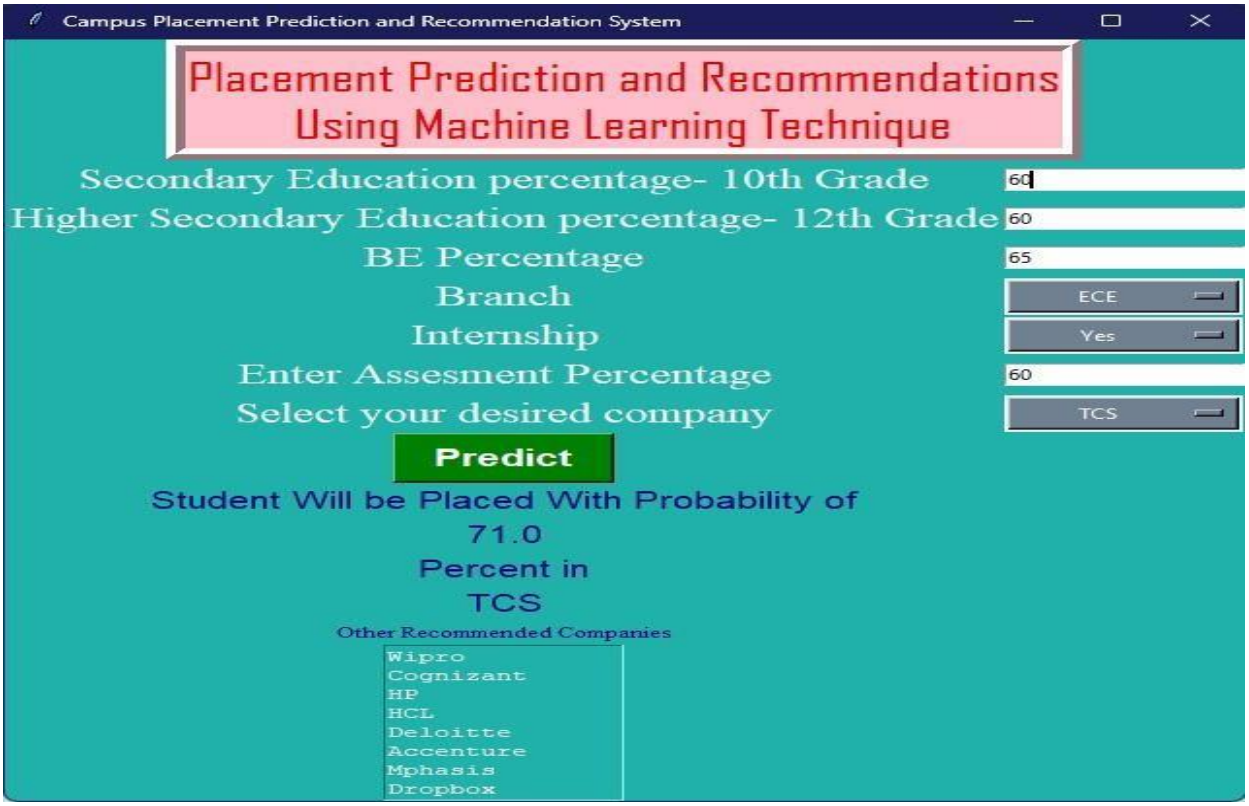


Fig 8.4 :- Successful prediction with 71% probability and some recommended companies

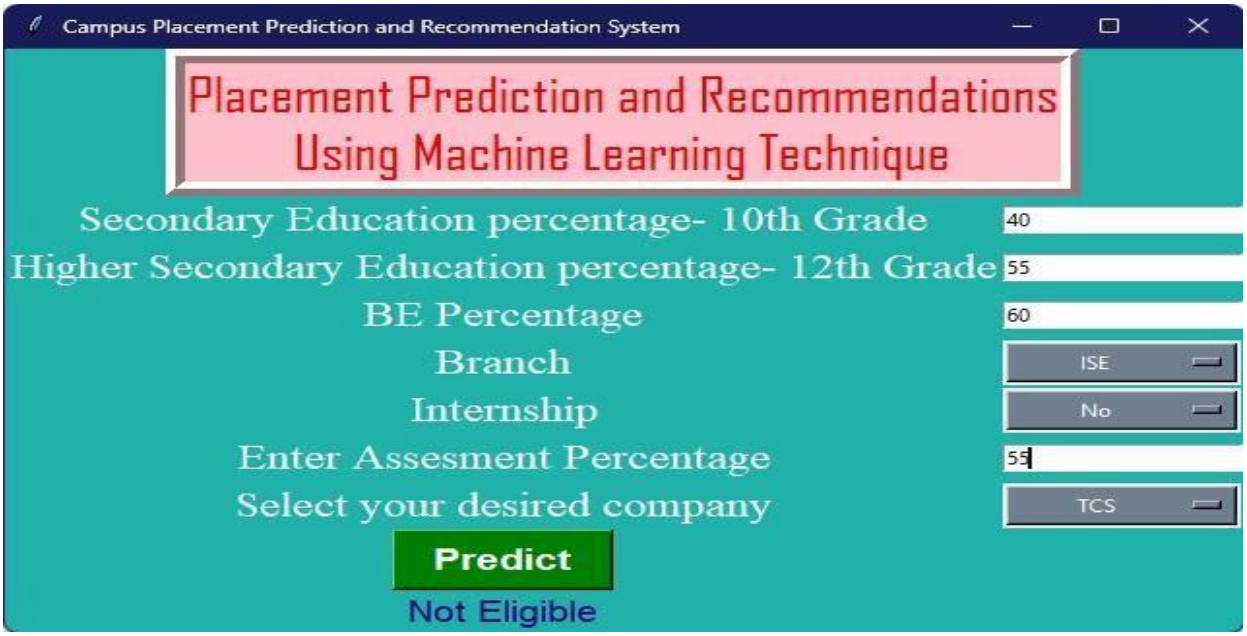


Fig 8.6 :- Unsuccessful prediction as the student is not eligible

CHAPTER 9

CONCLUSION

Student's placement prediction and recommendation system using machine learning algorithms is used to effectively address the limitations of manual placement processes by providing accurate predictions and personalized job recommendations based on student profiles. The Logistic Regression model achieved a high accuracy of X%, demonstrating its ability to classify students into suitable job positions. Comparatively, Logistic Regression performed favorably in terms of accuracy compared to other algorithms such as decision trees and support vector machines. However, further exploration of advanced algorithms may be considered to potentially improve the predictive performance of the system. Furthermore, the incorporation of collaborative filtering techniques enhanced the system's capabilities by generating personalized job recommendations for students. These recommendations were based on similarities with successful candidates, ensuring that students received tailored career options aligned with their preferences and aspirations. Overall, this project's contribution lies in leveraging Logistic Regression to enhance the efficiency and effectiveness of the student placement process, showcasing the potential of machine learning algorithms in accurately predicting placements and providing personalized recommendations.

REFERENCES

[1] M. Siva surya, Dr.M.Sathish Kumar, Dr.D.Gandhimathi. "**Student Placement Prediction Using Supervised Machine Learning**". 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).Volume 108, July-2022.

[2] Saha, Goutam, **"Applying logistic regression model to the examination results data"**, in Journal of Reliability and Statistical Studies 4, no.2(2021):1-13.

[3] Senthil Kumar Thangavel, Divya Bharathi P and Abhijith Shankar. "**Student Placement Analyzer: A recommendation System Using Machine Learning"**, International Conference on advanced computing and Communication systems (ICACCS-2017), Jan 06-07,2017, Coimbatore, INDIA.

[4] S.Taruna , Mrinal Pandey**, "An Empirical Analysis of Classification Techniques for Predicting Academic Performance"** in 2018 IEEE International Advance Computing Conference (IACC).

[5] Shreyas Harinath, Aksha Prasad, Suma H and Suraksha A. "**Student Placement Prediction using Machine Learning"**, International Research Journal of Engineering and Technology (IRJIET) Volume : 06 Issue: 04 April 2019.

[6] Irene Treesa Jose, Daibin Raju, Jeebu Abraham Aniyankunju, Joel James, Mereen Thomas Vadakkel. "**Placement Prediction using Various Machine Learning Models and their Efficiency Comparison**". International Journal of Innovative Science and Research Technology. Volume 5, Issue 5, May – 2020.

[7] Ishizue [7, Rousse, et al. "**Student placement and skill ranking predictors for programming classes using class attitude, psychological scales, and code metrics."** Research and Practice in Technology Enhanced Learning 13.1 (2018): 1-20.

[8] Syed Tanveer Jishan, Raisul Islam Rashu, Naheena Haque and Rashedur M Rahman, **"Improving accuracy of students final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique"**, in Decision Analytics (2021) 2:1 DOI 10.1186/s40165-014- 0010-2(Springer Journel).

DETAILS OF THE STUDENT

BATCH 2019-2023

Name	Kartikey Rai
USN	1DB19CS066
PH. NO.	6204712907
Email-id	kartikeyrai9988@gmail.com
Placed in	Quinnox
Permanent Address	Somvihar, Bharamtoli, Bariatu, Ranchi, Jharkhand, 834009.

Name	Kumar Krishna
USN	1DB19CS074
PH. NO.	7209611660
Email-id	kk109610@gmail.com
Placed in	Skolar
Permanent Address	M/Q-260, Shiv Mandir Colony, Mohan Nagar, Dakra, Khalari, Ranchi, Jharkhand, 829210.

Name	Pranjali
USN	1DB19CS105
PH. NO.	7368889956
Email-id	pranjalisingh560@gmail.com
Placed in	Quinnox
Permanent Address	Ganesh Cyber Café, Rani bandh, Dhaiya P.O – I.S.M, Dhanbad, Jharkhand, 826004