SYMBIOSIS
INSTITUTE OF GEOINFORMATICS
॥वसुधैव कुटुम्बकम्॥

CAMPUS PLACEMENT PREDICTION 2023

Mohammad Vohra

PRN: 22070243055

Course: DSSA

TABLE OF CONTENTS

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Sir Sahil Shah and Dr Vidya Patkar for their guidance and support throughout this project. Their expertise in the field of machine learning has been invaluable in the successful completion of this project.

Their insights, feedback, and suggestions were instrumental in shaping the direction and scope of the project. They provided continuous motivation and support, helping me to overcome various obstacles and challenges along the way.

I would also like to thank the faculty members of the Data Science Department for their continuous encouragement and support.

Finally, I would like to thank my friends and family for their unwavering support and encouragement throughout the project. Their constant motivation helped me to stay focused and complete the project successfully.

Once again, thank you to everyone who has helped me in this project. I am deeply grateful for your support and encouragement.

Mohammad Vohra

ABSTRACT

Because to a variety of "machine learning" approaches, software programmes may anticipate outcomes more accurately without having to be explicitly designed. The fundamental concept underlying algorithms is the development of algorithms and methods that can handle incoming data, predict outcomes through statistical analysis, and update findings when new data becomes available. So, precise measures can be made to accomplish the organization's goal, these models may be deployed in a variety of situations and taught in line with performance expectations. In this research, the placement of the students is predicted using their educational backgrounds. Results are delivered using the technique used to create a dataset for education libraries and its numerous components.

Using several components of a dataset acquired for academic institutions and the method used to develop a prediction model, solutions with high levels of precision are created. These findings might help them expand their recruiting department and thereby enhance their university as a whole.Developing a mechanism that can forecast a student's placement status across different universities is the goal of the project. by reviewing the placement status of previous student records. Institutions of higher learning kept records on their pupils to predict their placement in the future. Annual admittance fluctuates in direct proportion to the placement it offers its pupils. Every institution is eager to enhance its placement cell for this reason. This will always benefit the school as well as the pupils..

Machine learning

The amount of data accessible is growing daily, and this massive volume of unprocessed data must be carefully evaluated in order to provide outcomes that meet the current standards for being highly useful and exquisitely pure. It is accurate to claim that Machine Learning (ML) is evolving at a rapid rate, just as Artificial Intelligence (AI) has over the previous 20 years. ML is a significant pillar of the IT industry and, as a result, a significant, if mostly unnoticed, aspect of our lives. Data is incredibly important in current aspects, therefore as technology advances, so will the analysis and interpretation of data to provide effective outcomes.

Machine learning deals with both supervised and unsupervised task types, and frequently a classification-type problem acts as a source for information literacy. The main goal is to create a self-efficient system that can do computations and analysis on its own and deliver answers that are considerably more precise and perfect. In order to produce precise future forecasts, it both develops resources and applies regression. By using statistical and probabilistic algorithms, data may be turned into knowledge. Statistical inference is conceptually based on sampling distributions.

ML can take a variety of shapes. This study first examines a variety of ML applications as well as the types of data they use. The problem statement that serves as the study's main topic is formalised after that.

Problem Statement

Student placement is one of an academic institution's most important objectives. The placements a school gives its students are intrinsically related to the institution's reputation and yearly admissions. Every institution therefore makes a concerted effort to improve its employment division in order to progress the organization as a whole. The capacity of an institution to promote its students would be positively impacted by any help in this specific area. Both the institution and the students will always benefit from this.
The primary objective is to forecast, using the data provided in the dataset, whether or not the students will be selected for campus placements.

Approach: The traditional machine learning activities, including model building, model testing, feature engineering, data exploration, and data cleaning.

Proposed Solution

We will perform EDA to find the important relation between different attributes and will use a machine-learning algorithm to predict the placement status.

Data Requirements

The Campus Placement Prediction data is recorded many students information along with educational career description
The placement information for students on our campus makes up this data collection. It contains percentage and speciality from secondary and advanced secondary schools. Moreover, it covers the degree specialisation, kind, and pay offers to the placed students.

" Source: https://www.kaggle.com/benroshan/factors-affecting-campus-placement or https://www.kaggle.com/c/ml-with-python-course-project/data "

Attribute Information:

sl_no Serial Number

gender Gender- Male='M',Female='F'(male=0, female=1)

ssc_p Secondary Education percentage- 10th Grade

ssc_b Board of Education- Central/ Others

hsc_p Higher Secondary Education percentage- 12th Grade

hsc_b Board of Education- Central/ Others

hsc_s Specialization in Higher Secondary Education

degree_p Degree Percentage

degree_t Under Graduation(Degree type)- Field of degree education

workex Work Experience

etest_p Employability test percentage (conducted by college)

specialisation Post Graduation(MBA)- Specialization

mba_p MBA percentage

status Status of placement- Placed/Not placed

salary Salary offered by corporate to candidates

Tools Used

The programming language is Python that is used here, also we will use some other python-based libraries like for ml we will use Scikit-Learn library, for data manipulation we will use pandas, for numerical computation Numpy and also we have used psycopg2 to store data in postgres sql.

Data Gathering

Data source: https://www.kaggle.com/competitions/ml-with-python-course-project/data

Raw data Gathering

Before moving on with any operation, multiple sorts of validation must be performed on the imported data. Validations include ensuring that all of the columns have a standard deviation of zero and ensuring that no columns have any full missing data. They are necessary because the qualities that include them are useless. It will not be used to forecast a student's placement status.

For example, if an attribute has zero standard deviation, all of its values are the same and its mean is zero. This suggest that the characteristic will stay the same whether the individual is placed or not. According to this, it serves no use to take any property into consideration while operating if all of its values are absent.

Data Transformation

Data transformation is necessary before transferring the data to the database so that it may be transformed into a format that makes database insertion simple. The training data in this case has no missing values. If any values are missing, both the train set and the test set have those values filled in using the supported relevant data types.

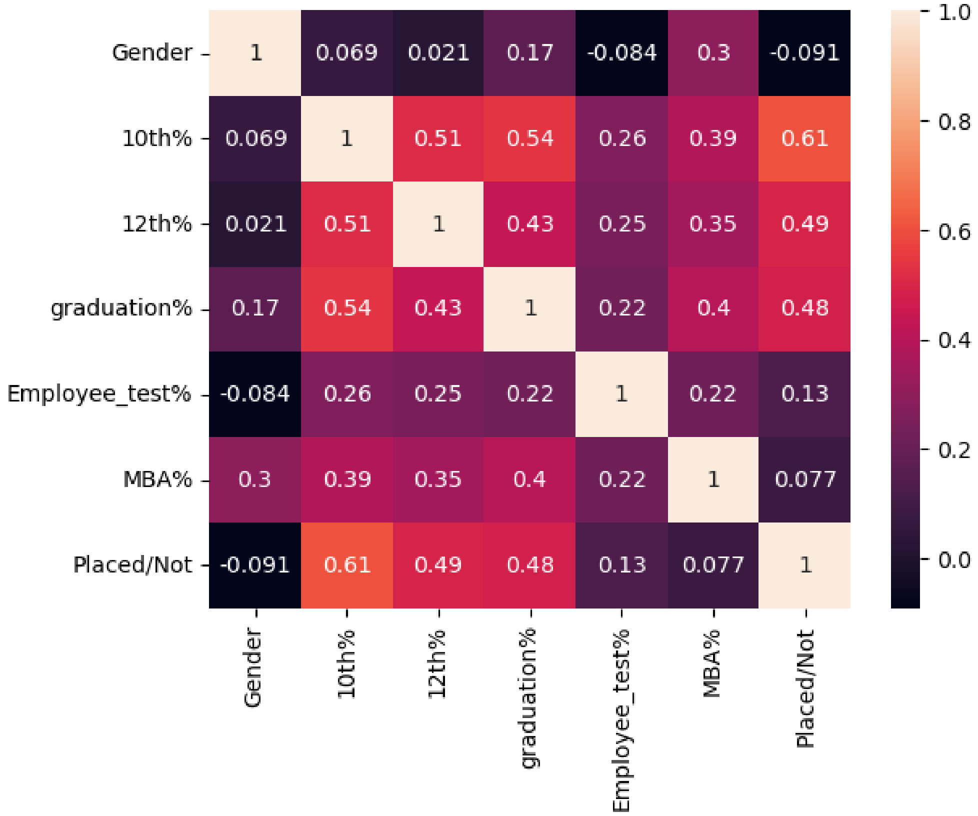
Data Preprocessing

All of the steps necessary before transmitting the data for building the model are completed during data preparation. Here, I removed all those columns who standard deviation is zero because they are not contributing anything and those columns also which are of no use in predicting the status of Placement. After that I separate label and features columns from the train and test data.

Assignng values:

```
#'Placed': 1, 'Not Placed': 0
df['Placed/Not'] = df['Placed/Not'].map({'Placed': 1, 'Not Placed': 0})
```

- **Correlation:**



Analysis From Correlation matrix

As we can see, MBA% is not that much correlated with our data we are going to drop it

```
# as we can see, MBA% is not that much correlated with our data we are going to drop it
df.drop("MBA%",axis=1,inplace=True)
```

Feature Encoding

I have converted the target column categorical values into numerical using label encoder.

```
#Encoding into numerical values
from sklearn import preprocessing
encoder = preprocessing.LabelEncoder()
for i in df.columns:
 if isinstance(df[i][0], str):
    df[i] = encoder.fit_transform(df[i])
```

Performance

The Campus Placement Prediction is dependent on machine-learning algorithms. We will train various ml algorithms and will find the best fitting algorithm for predicting the target. Our system performance will be based on the data we are going to feed to the algorithms. And the performance will depend on the finalized model. model training is also very important to improve the performance. Below algorithms are used for determining the best algorithm for our dataset:

Logistic Regression	: 79.06976744186046 %
Decision Tree	: 81.3953488372093 %
Random Forest	: 83.72093023255815 %
AdaBoost	: 81.3953488372093 %
Support Vector Machine	: 83.72093023255815 %
Bernoulli Naive Bayes	: 69.76744186046511 %
Naive Bayes	: 93.02325581395348 %

Application Compatibity

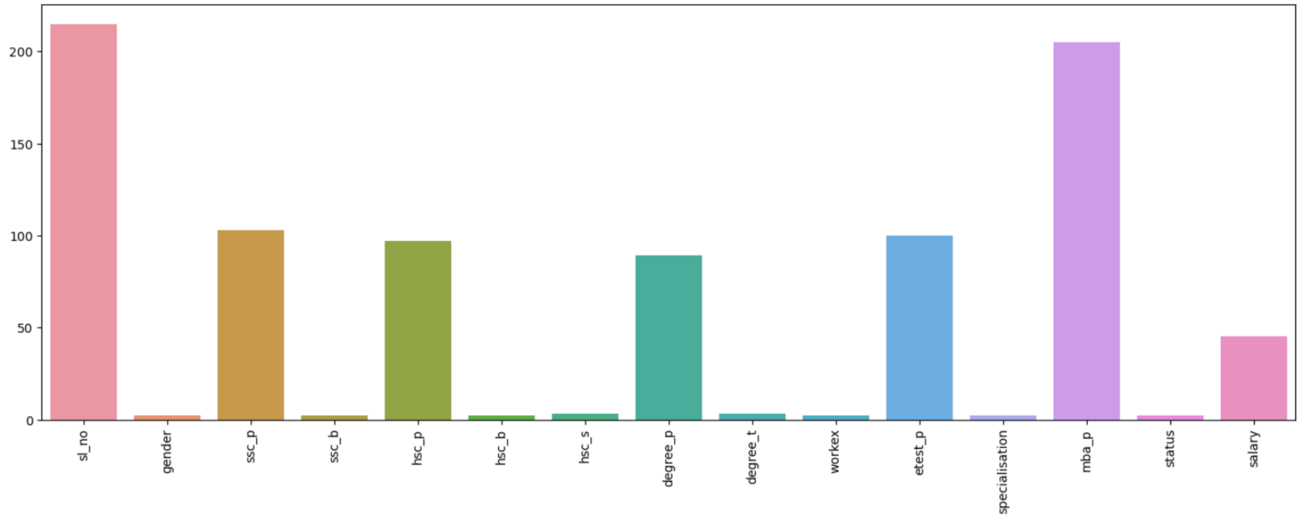
Python will serve as a gateway between the many parts of this project. Each element will have a certain duty to do, it is the responsibility of the Python that guarantee accurate information transmission.

VALUE COUNTS

Gender 0 64.651163 1 35.348837 Name: Gender, dtype: float64	10th_board Central 53.953488 Others 46.046512 Name: 10th_board, dtype: float64
12th_board Others 60.930233 Central 39.069767 Name: 12th_board, dtype: float64	12th_specialization Commerce 52.558140 Science 42.325581 Arts 5.116279 Name: 12th_specialization, dtype: floa t64
graduation_deg Comm&Mgmt 67.441860 Sci&Tech 27.441860 Others 5.116279 Name: graduation_deg, dtype: float64	Work_experience No 65.581395 Yes 34.418605 Name: Work_experience, dtype: float64
MBA_specialization Mkt&Fin 55.813953 Mkt&HR 44.186047 Name: MBA_specialization, dtype: float 64	Placed/Not Placed 68.837209 Not Placed 31.162791 Name: Placed/Not, dtype: float64

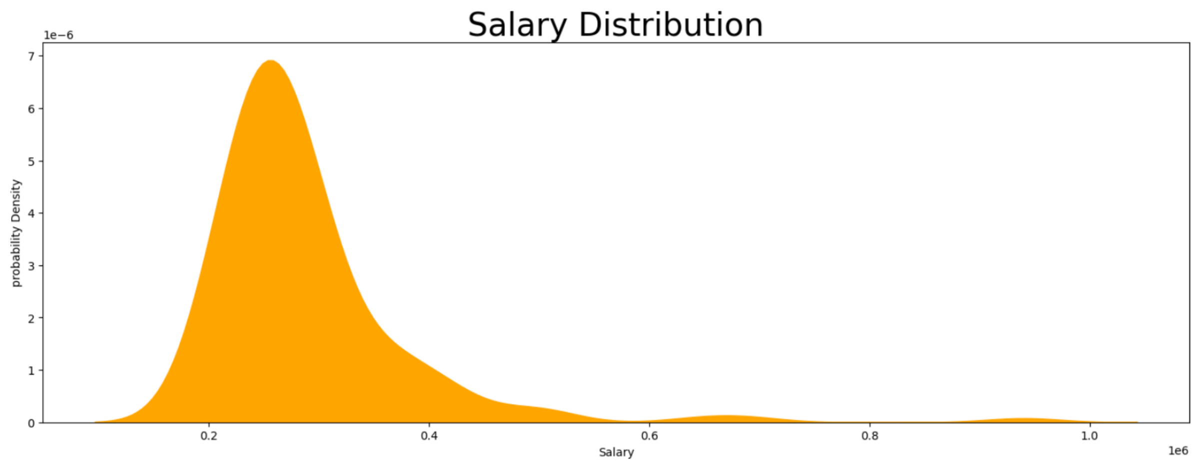
VISUALIZATIONS

Unique number of values in a particular feature

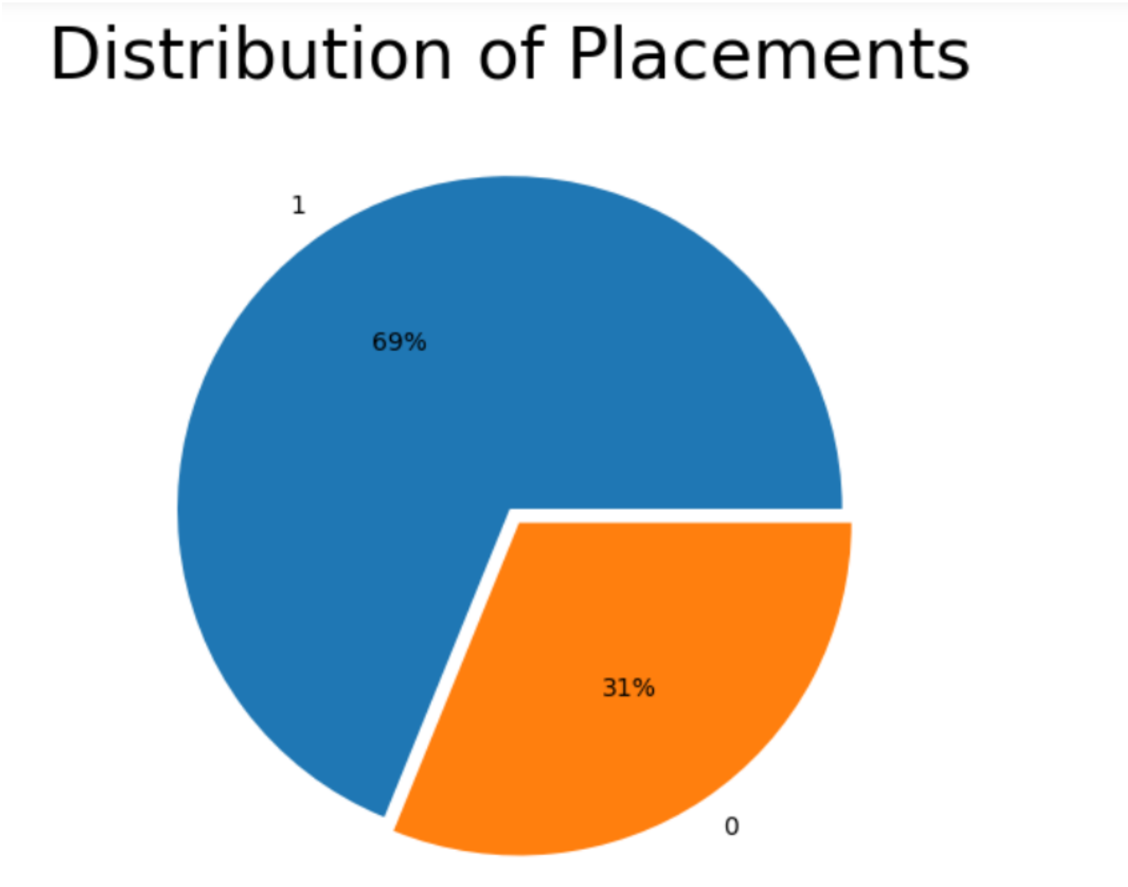


Non-Missing Values

Salary Distribution

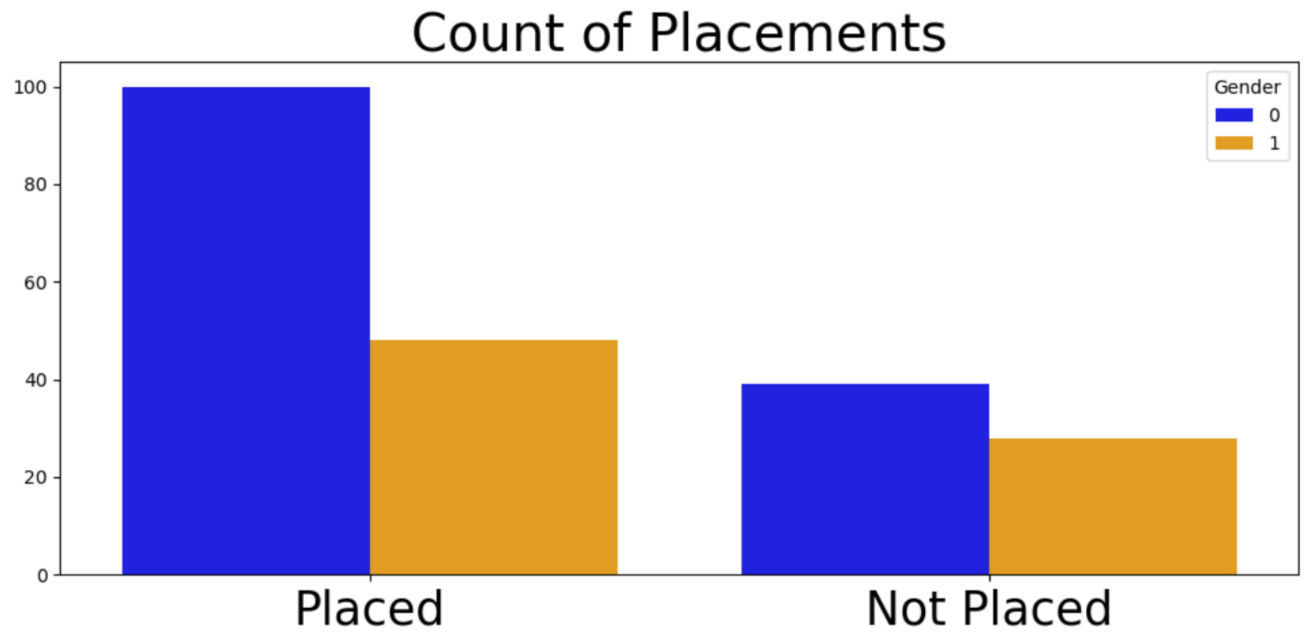


Distribution of Placements



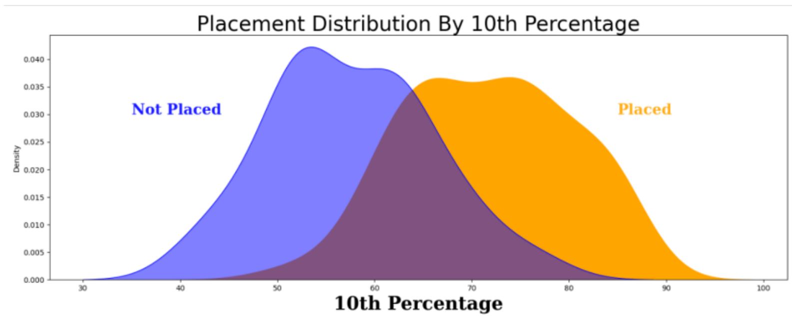

```
1    148
0     67
Name: Placed/Not, dtype: int64
```

Count of Placements

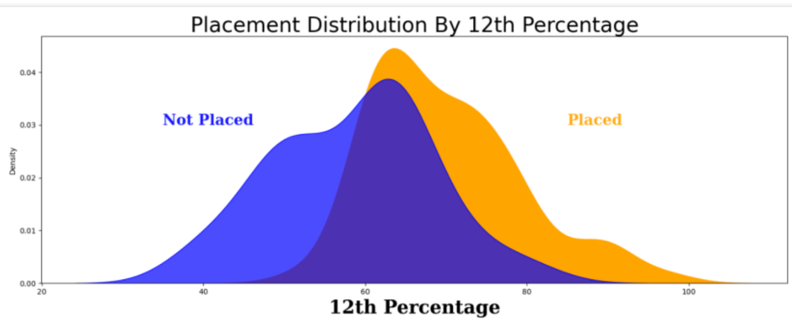


Placement Distribution by Various educational phases

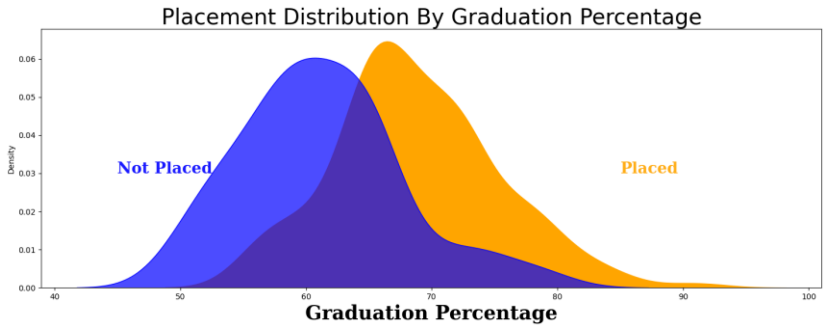
Placement Distribution By 10th Percentage



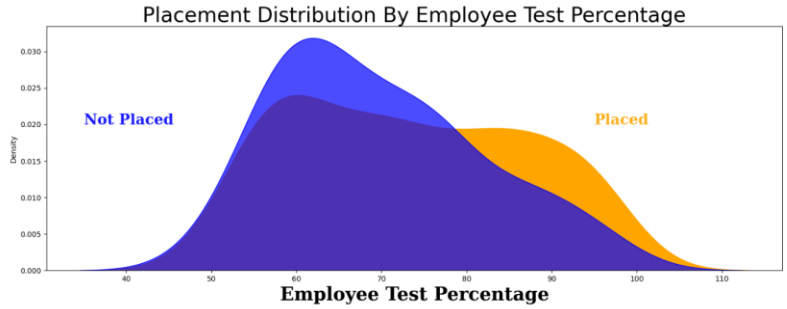
Placement Distribution By 12th Percentage



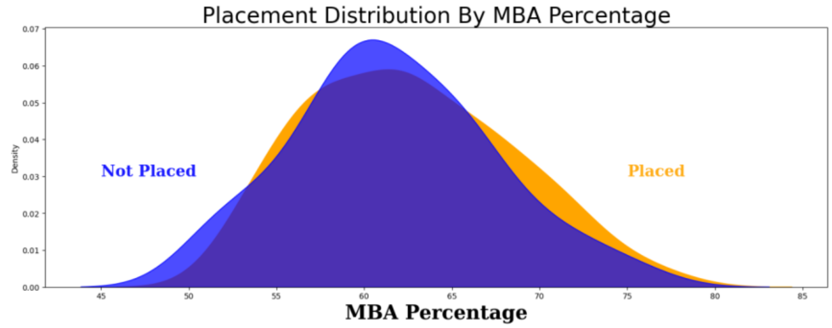
Placement Distribution By Graduation Percentage



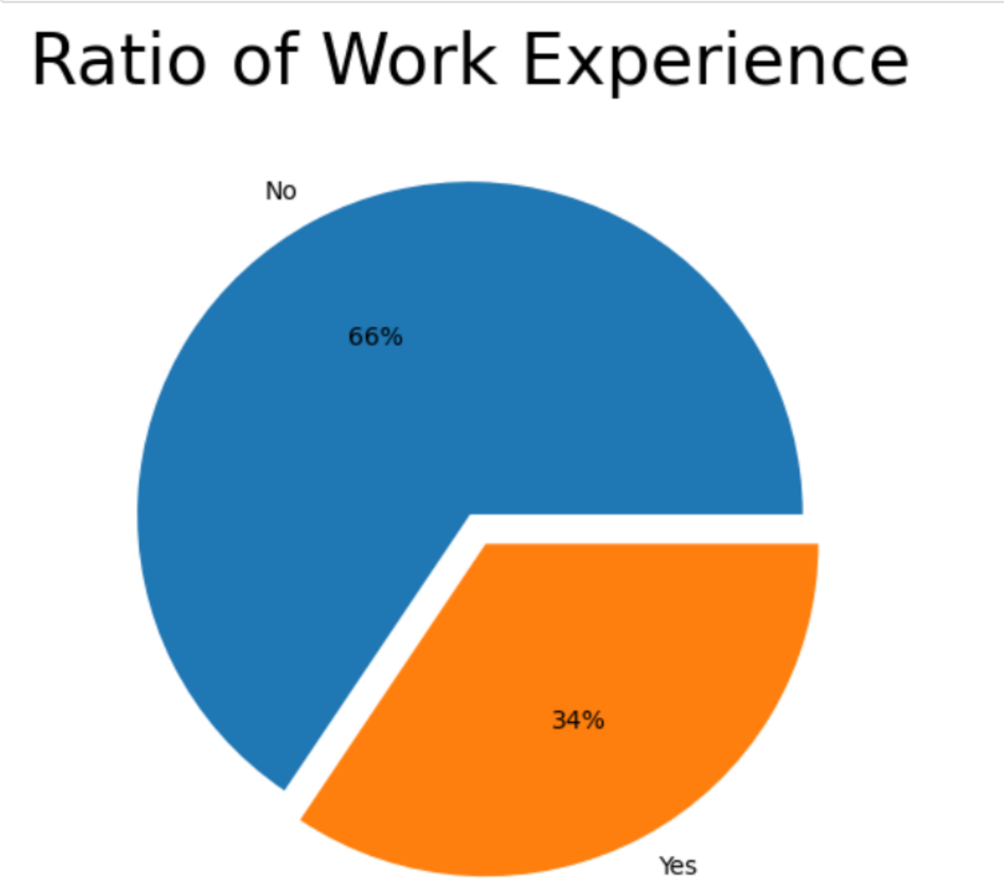
Placement Distribution By Employee Test Percentage



Placement Distribution By MBA Percentage



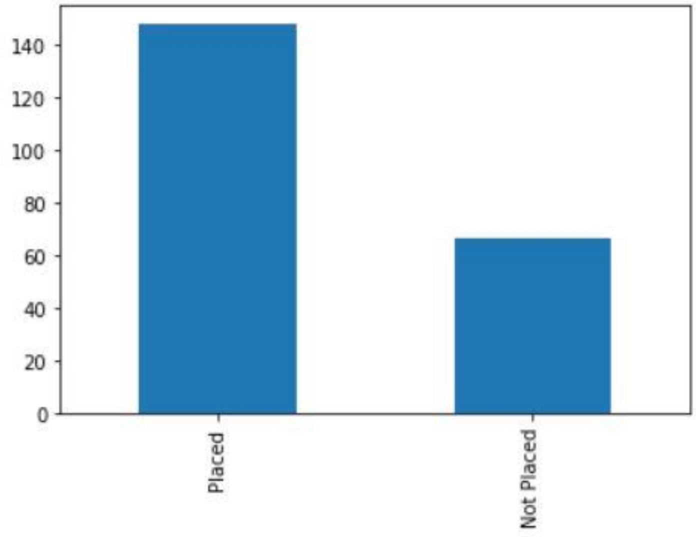
Ratio of Work Experience



- **Barplot of Placed and Not Placed**

```
train['status'].value_counts().plot(kind='bar')
```

```
<AxesSubplot:>
```



DATASET DESCRIPTION

The name, type, subsequent measurement, and a brief explanation of the variable are all provided. The sequence of the numbers along the database's rows matches the sequence of this listing.

Name	Data Type	Measurement
Gender	Integer	Gender of student
ssc_p	Float	Ssc percentage
ssc_b	Object	Ssc board
hsc_p	Float	Hsc percentage
Hsc_b	Object	Hsc board
Hsc_s	Object	Hsc stream
Degree_p	Float	Degree percentage
Degree_t	Object	Graduation stream
Workex	Object	Are they having any work experience
Etest_p	Float	Online test percentage
Specialisation	Object	Specialization the choosed in Mba
Mba_p	Float	Mba percentage
Status	Object	Are they placed or not placed. This the outcome column.

Data of students for predicting campus placement is collected from various institutions. This data about students contains their percentage at different educational level and which stream they choose in graudation and post graduation. The dataset looks like as follow:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	0	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	0	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	0	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	0	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	0	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	211	0	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0
211	212	0	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0
212	213	0	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0
213	214	1	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0
214	215	0	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN

As seen in Fig., the data collection includes a variety of data kinds, including integer, float, and object..

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           215 non-null    int64  
 1   gender          215 non-null    int64  
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object 
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object 
 6   hsc_s           215 non-null    object 
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object 
 9   workex          215 non-null    object 
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object 
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object 
 14  salary          148 non-null    float64
dtypes: float64(6), int64(2), object(7)
memory usage: 25.3+ KB
```

The raw data may contain a variety of underlying patterns that might potentially offer insights into the problem and in-depth knowledge about the topic of interest. Nonetheless, care should be taken when working with data since it may contain missing value, duplicated values, or other types of ambiguity that call for pre-processing. So, an information should be examined as completely as is practical.

The following table illustrates many statistically significant parameters for numerical properties, including average, variance, median, count of values, maximum value, etc.

```
train.describe()
```

	sl_no	gender	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	148.000000
mean	108.000000	0.353488	67.303395	66.333163	66.370186	72.100558	62.278186	288655.405405
std	62.209324	0.479168	10.827205	10.897509	7.358743	13.275956	5.833385	93457.452420
min	1.000000	0.000000	40.890000	37.000000	50.000000	50.000000	51.210000	200000.000000
25%	54.500000	0.000000	60.600000	60.900000	61.000000	60.000000	57.945000	240000.000000
50%	108.000000	0.000000	67.000000	65.000000	66.000000	71.000000	62.000000	265000.000000
75%	161.500000	1.000000	75.700000	73.000000	72.000000	83.500000	66.255000	300000.000000
max	215.000000	1.000000	89.400000	97.700000	91.000000	98.000000	77.890000	940000.000000

The preprocessing of this dataset includes studies on the independent factors, such as checking for missing values in each column and then substituting and replacing them with supported suitable data types in order to ensure that analysis and model fitting are accurate. The images above show a few of the visualisations made with Pandas tools, which include details on works by collecting for category column model values and variables count for numerical column variable counts. It depends on the highest and/or least value for the numerical column as well as their percentage values for the median when deciding which value to concentrate for the next inquiry activities and analysis. Throughout the model building process, data types from various parts are also used for label computation and encoding.

Implementation and Results

This section discusses the computer language, libraries, implementation platform, data modelling, and the findings and conclusions drawn from it.

Implementation Platform and Language

Python is a general-purpose, interpreted-high level language that is frequently used in modern times to solve domain issues rather than handle system complexity. For programming, it is also known as the "batteries included language." It has a number of libraries utilised for scientific research and inquiries as well as a number of libraries from other parties to facilitate effective issue solutions.

The Python libraries Numpy and Matplotlib have been utilised in this study for scientific computing and 2D visualisation, respectively. Together with this, the Python tools Pycopg2 and Pandas have been used for database insertion and data analysis, respectively. Jupyter Notebook has been widely used as a development environment.

Metrics for Data Modelling

- In order to determine classification accuracy, a classification model must first be used to predict each instance in a test dataset. The actual labels for those cases in the test set then are compared to the predictions. The percentage of accurately predicted cases in the testing set divided by the total number of predictions made on the testing set is then used to measure accuracy.

Area under the curve, or AUC, is a measure of how accurate a prediction is. So, before discussing ROC AUC score, ROC curve must first be defined.

It is a graph that illustrates how true positive rate (TPR) & false positive rate trade off (FPR). Fundamentally, we compute TPR and FPR for each threshold and present the results on a single chart.

Naturally, the better classifiers are those with top-left-side curves since they have greater TPR and lower FPR for each threshold.

MODEL BUILDING

After doing all kinds of preprocessing operations mention above the data set is passed into 7 models, Logistic Regression, Decision Tree, Random Forest, AdaBoost, Support Vector Machine, Bernoulli Naive Bayes, Naive Bayes. It was found that Naïve Bayes classifier performs best with the highest accuracy score equals 93.02%. So Naïve Bayes classifier performed well in this problem.

- **Logistic Regression**

To predict binary result based on several characteristics, a statistical method known as logistic regression is applied. In campus placement prediction, the result is a binary one: either a student will be put or not. Based on a variety of independent variables, including a student's academic standing, employment history, and other elements, logistic regression may be used to forecast the likelihood that the student would be placed. This approach is frequently utilised because it offers a straightforward, interpretable model that is simple to use and comprehend. It has also been demonstrated to be successful in a number of applications, including forecasting churn rate, creditworthiness, and medical diagnosis. As a result, the approach of logistic regression is appropriate for predicting campus placement.

- **Decision Tree**

A strong machine learning method called a decision tree may be used to predict outcomes accurately in a variety of situations, including campus location.

This algorithm works by analyzing a set of training data and identifying patterns and relationships between various variables. It then uses this information to make predictions about future outcomes based on new input data.

In the context of campus placement prediction, decision tree algorithms can be used to analyze a range of factors that are likely to influence whether a student is selected for a particular job or internship. These might include things like academic performance, extracurricular activities, work experience, and personal characteristics. By analyzing these variables, a decision tree algorithm can make accurate predictions about which students are most likely to be successful in the job market.

One of the key advantages of decision tree algorithms is that they are highly interpretable. This means that it is easy to understand how the algorithm is making its predictions, which makes it a valuable tool for businesses, educators, and other stakeholders who want to gain insights into the job market and improve their hiring practices. Additionally, decision tree algorithms are highly flexible and can be customized to suit a wide range of different scenarios and data types. This makes them a valuable tool for predicting campus placement outcomes, as well as other types of business and organizational decisions.

- **Random Forest**

A machine learning approach called Random Forest may be applied to both classification and regression applications. It works by building a multitude of decision trees and combining their outputs to make predictions. This makes it an effective tool for predicting campus placements, as it can consider multiple factors (such as academic performance, skills, internships, etc.) and weigh their importance in making predictions. Additionally, Random Forest can handle missing data and noisy features, making it more robust and reliable than other algorithms. Overall, Random Forest can be a useful tool for predicting campus placements with high accuracy and efficiency.

- **AdaBoost**

AdaBoost is a popular ensemble learning algorithm that combines several weak classifiers to create a more accurate prediction model. It can be used for various machine learning tasks, including campus placement prediction.

In campus placement prediction, AdaBoost can be used to predict the probability of a student getting placed in a particular company based on their academic performance, skill set, and other factors. By combining multiple weak classifiers, AdaBoost can create a more accurate prediction model that can take into account the complexity of the data.

Moreover, AdaBoost is a versatile algorithm that can work with different types of data and can handle both categorical and continuous variables. It is also robust against overfitting and can handle noisy data well.

Therefore, using AdaBoost for campus placement prediction can be a useful tool for universities and colleges to predict the likelihood of their students getting placed in different companies and help them prepare better for the job market.

- **Support Vector Machine**

Campus placement predictions may be made using the potent machine learning algorithm Support Vector Machine (SVM). The supervised learning method SVM may be used to categorise data into several groups.. In the case of campus placement prediction, the algorithm can be trained on the data of past campus placements, which includes factors such as students' academic performance, skills, and work experience.

SVM can be used in campus placement prediction because it is effective in handling large datasets with a high number of features. SVM can also handle both linear and non-linear data, which makes it a versatile algorithm for classification. Additionally, SVM is effective in handling noisy and incomplete data, which is common in campus placement datasets.

Moreover, SVM is a well-established algorithm that has been used in various applications, including image recognition, text classification, and finance. The algorithm has been proven to be accurate and efficient in these applications.

In conclusion, SVM is a suitable algorithm for campus placement prediction due to its ability to handle large datasets with high dimensions, non-linear data, noisy and incomplete data, and its proven accuracy in various applications.

- **Bernoulli Naive Bayes**

Bernoulli Naive Bayes is a classification algorithm that works by calculating the probability of a data point belonging to a certain class based on its features. This algorithm can be used for campus placement prediction by training it on historical data of students who have been placed or not placed in the past.

The algorithm can use features such as academic performance, skills, and other attributes to predict whether a student is likely to get placed or not. Since Bernoulli Naive Bayes works well with binary data, it is suitable for this task where the target variable (placement) is binary.

Using Bernoulli Naive Bayes for campus placement prediction can help colleges and universities to identify students who are likely to get placed and provide them with the necessary support and resources to improve their chances of getting hired. It can also help institutions to identify areas where they need to focus on improving the employability of their students.

In conclusion, Bernoulli Naive Bayes is a useful algorithm for campus placement prediction due to its ability to handle binary data and calculate the probability of a data point belonging to a certain class based on its features.

- **Naive Bayes**

A machine learning approach called Naive Bayes is used to categorise data according to likelihood. It may be used for different forms of data in addition to text categorization and spam filtering.

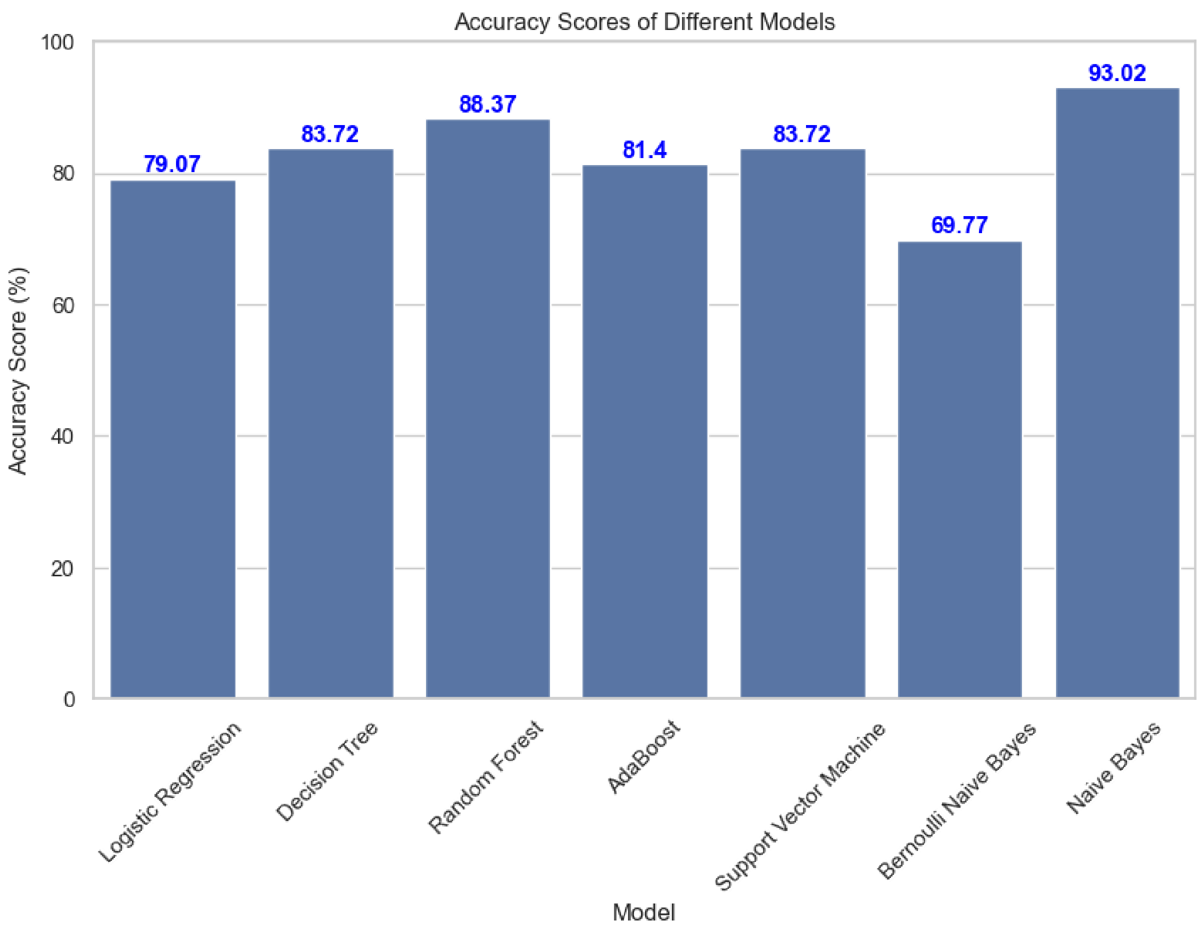
In the context of campus placement prediction, Naive Bayes can be used to analyze the factors that contribute to a student's success in getting a job. By looking at past data, such as the student's academic performance, extracurricular activities, and previous work experience, the algorithm can predict the likelihood of a student getting a job after graduation.

One of the advantages of Naive Bayes is that it is relatively simple to implement and requires less data to train the model than other machine learning algorithms. This makes it a good choice for applications where data is limited or the model needs to be updated frequently.

Ultimately, Naive Bayes can be a useful technique in campus placement prediction, giving insights into the elements that influence a student's performance and assisting both students and businesses in making decisions.

Model Accuracy:

```
Logistic Regression : 79.06976744186046 %
Decision Tree : 81.3953488372093 %
Random Forest : 83.72093023255815 %
AdaBoost : 81.3953488372093 %
Support Vector Machine : 83.72093023255815 %
Bernoulli Naive Bayes : 69.76744186046511 %
Naive Bayes : 93.02325581395348 %
```



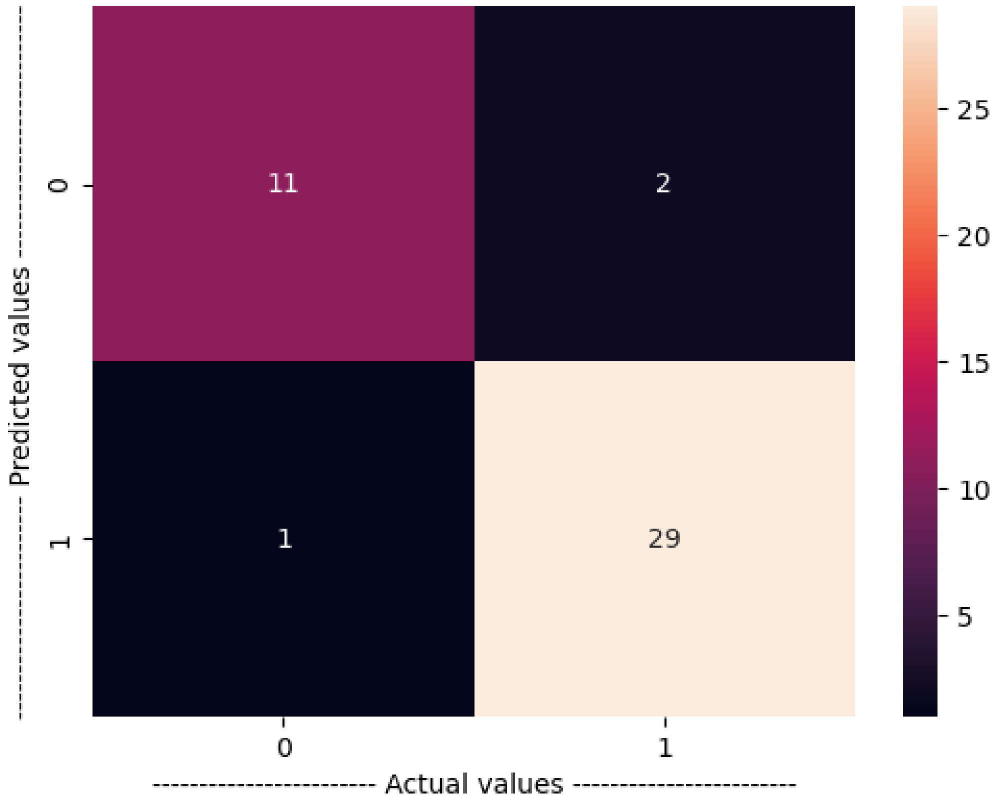
So from the above scores and scores visualizations we select the model with highest accuracy.

We select Naïve Bayes Algorithm.

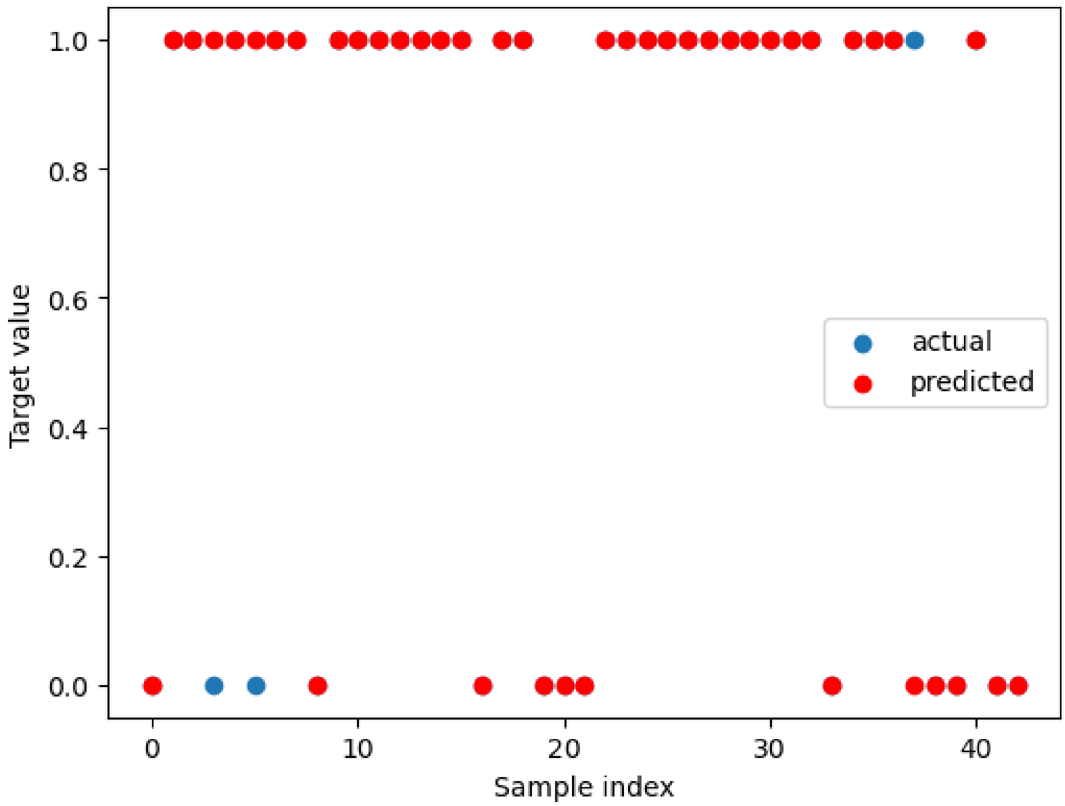
- **Classification Report**

	precision	recall	f1-score	support
0	0.92	0.85	0.88	13
1	0.94	0.97	0.95	30
accuracy			0.93	43
macro avg	0.93	0.91	0.92	43
weighted avg	0.93	0.93	0.93	43

- Confusion matrix : Naïve Bayes



- **Scatter plot : Actual VS Prediction**



Application Compatibity

Python will serve as a bridge between the many parts of this project. Each element will have a certain duty to do, it is the responsibility of both the Python to guarantee accurate information transmission.

Prediction

The user must provide the required information, such as gender, hsc p, hsc b, degree p, degree t, workex, etest p, specialty, etc. The output, which displays whether or not the student has been placed, is based on the user's input.

Conclusion

My project "The Campus Placement Prediction" will assist educational institutions in forecasting their students' placement status. It may assist them improve their institution as a whole by fortifying their placement department.

For predicting campus location, the machine learning method Naive Bayes is helpful. By analyzing past data such as academic performance, extracurricular activities, and previous work experience, Naive Bayes can predict the likelihood of a student getting a job after graduation. It is a simple algorithm that requires less data to train the model than other machine learning algorithms, making it a good choice for applications where data is limited. Using Naive Bayes for campus placement prediction can provide valuable insights into the factors that contribute to a student's success and guide the decision-making process for both students and employers. Overall, Naive Bayes is an effective tool for improving the placement rate of students in campus recruitment.

References

https://www.kaggle.com/c/ml-with-python-course-project/data