Report on Placement Prediction using different ML techniques

Submitted to :

Dr. Vishal Gupta

Associate Professor

Department of Computer Science and Information Systems

CS F407 Artificial Intelligence

Prepared By:

Shubh Nema **2021A1PS2602P**

Harsh Nema **2021AAPS2714P**

Acknowledgement:

We are indebted to our professor Dr Vishal Gupta Sir for his constant guidance and support during the Artificial Intelligence course. We are also grateful to the Vice Chancellor, Director and administration of BITS Pilani for making it possible for us to pursue this course.

Abstract:

In this assignment, the use of machine learning (ML) techniques to forecast student placement in an Engineering institute is examined. We examined past student data and used machine learning techniques to create models that can predict appropriate placements for incoming students with accuracy. We evaluated each technique's performance through extensive testing and analysis on a real-world data with a somewhat different distribution as compared to the one on which our models were trained upon. Our results offer insightful information about the efficacy of machine learning techniques in predicting student placement, laying the groundwork for the adoption of data-driven decision-making procedures in learning environments.

Introduction and our approach:

The objective is to develop predictive models capable of forecasting optimal placements for students in educational settings by harnessing the power of predictive analytics to enhance students' job prospects. Inspired by research demonstrating the effectiveness of machine learning in foreseeing crucial student outcomes, our initiative aims to leverage existing machine models on datasets enriched with academic and extracurricular data.

Through meticulous data preprocessing, exploratory data analysis, and model selection, we strive not only to accurately predict placement outcomes but also to unearth actionable insights for proactive interventions. By optimizing model parameters and evaluating performance metrics, our approach ensures that the system not only predicts future job placements but also empowers educators and stakeholders with valuable insights to guide students towards successful career paths.

RESEARCH PAPERS REFERRED

- **Classification Model of Prediction for Placement of Students:** This research employs data mining techniques, specifically Naïve Bayes, Multilayer Perceptron, and J48 decision tree algorithms, to predict the placement of Master of Computer Applications (MCA) students. Analyzing diverse attributes from students' academic records, the study found that the Naïve Bayes algorithm outperformed others with an accuracy of 86.15%, quick model-building, and minimal errors. The research concludes that Naïve Bayes holds substantial potential to be applied on such datasets.

Limitations - Naïve Bayes assumes independence between features, which might not hold in real-world scenarios. In the context of job recommendations, skills or attributes may be correlated, and the independence assumption might oversimplify the model. Also, MLPs are prone to overfitting, especially when dealing with small datasets.Since the dataset we choose for our assignment is a relatively smaller one, overfitting might occur.

- **Collaborative Job Prediction based on NaIve Bayes Classifier using Python Platform**: The research paper introduces a collaborative filtering-based recommendation system for job portals, focusing on users' skill sets. Given the user's skill set, the algorithm employs Euclidean distance and Pearson coefficient algorithm to calculate a similarity index between different skills, and then ranks them using Naïve Bayes, and incorporates Bayesian weights. In this way, it tries to recommend jobs to the users. Testing reveals an average accuracy of 91.33% and 92.74% for division ratios of 0.2 and 0.25.

Limitations- The algorithm relies on Euclidean distance to calculate the similarity index between skill sets. Euclidean distance might not capture the semantic relationships between skills accurately, especially when dealing with high-dimensional and sparse data. The issue of the naive assumption of independence between the features might be incurred here as well since the skills are ranked using Naive Bayes only.

- **Model Construction Using ML for Prediction of Student Placement:**

Different ways (Support Vector Machine, Naive Bayes, Logistic Regression, Decision Tree, and XGBoost) to predict student placements are used. After analyzing their performance using metrics like accuracy, precision, recall, and F1-score, we found that Support Vector Machine (SVM) was the most accurate, achieving a success rate of 91%. This means SVM provided the best predictions compared to the other methods we tested.

Limitation: The paper does not explicitly discuss the rationale behind the selection of specific features for prediction. The choice of features is crucial, and the omission of relevant factors may impact the model's accuracy.

- **Prediction of Final Result and Placement of Students using Classification Algorithm:**

The research focuses on data mining algorithms to predict final result and placements of graduate student based on previous academic record. The study used classification trees to predict MCA results and student placements. The error rates for validating MCA result and

placement predictions were 38.46% and 45.38%, respectively. This suggests that anticipating final results and placements is possible before admitting students to the course.

Limitation: The hypotheses are based on the assumption of linear relationships between academic performance and placement outcomes. The real-world scenario may involve complex, non-linear relationships influenced by various factors like soft skills, extracurricular activities, and individual characteristics.

Assignment problem:

The task assigned involves preparing a comprehensive report on placement prediction using machine learning (ML) techniques for student placement. The objective is to develop predictive models capable of forecasting optimal placements for students in educational settings. This includes considering certain relevant attributes for placement, training the dataset for prediction, and subsequently testing the accuracy of different algorithms on separate datasets.

Implementation:

The implementation of the assignment involves a structured approach to developing and evaluating ML models for student placement prediction and broad idea about it is as follows:

1.**Data Collection, Analysis and Preprocessing:**

- We will start with out first step- Exploratory data analysis- which is to analyse the data and draw important insights from it, which will be used for the purpose of efficient model building. A snippet of the first five rows of the data is provided to check the structure of the data.

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

- We observed that the data contains 6 numerical columns and 2 categorical columns- namely gender and Branch. Gender takes Male or Female values, which there are 6 branches in total.

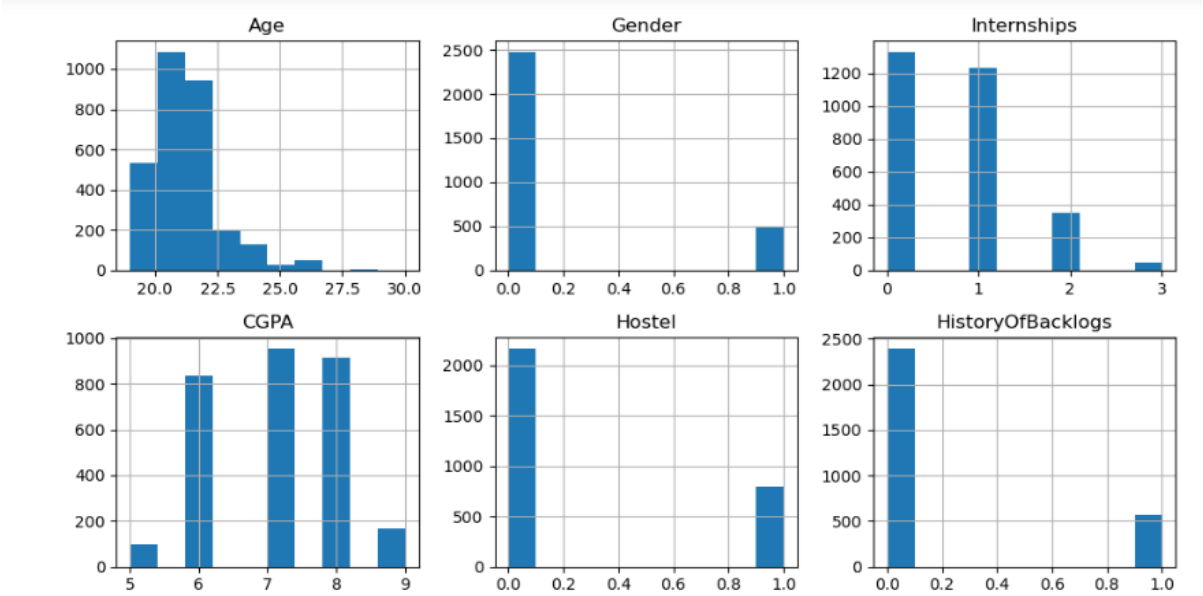
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 2966 non-null   int64
 1   Gender              2966 non-null   object
 2   Stream              2966 non-null   object
 3   Internships         2966 non-null   int64
 4   CGPA                2966 non-null   int64
 5   Hostel              2966 non-null   int64
 6   HistoryOfBacklogs   2966 non-null   int64
 7   PlacedOrNot         2966 non-null   int64
dtypes: int64(6), object(2)
memory usage: 185.5+ KB
```

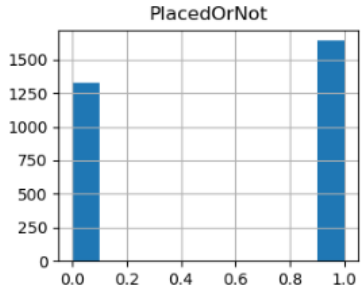
- Next, we observed that there are no null values in the dataset, hence we don't need any preprocessing to handle Nan values.

```
In [1065]: df.isnull().sum()

Out[1065]: Age                 0
           Gender              0
           Stream              0
           Internships         0
           CGPA                0
           Hostel              0
           HistoryOfBacklogs   0
           PlacedOrNot         0
           dtype: int64
```

The code `df.isnull().sum()` is used to count the number of missing values (NaNs) in each column of a DataFrame `df`.

As a part of our data analysis, we now try to see the distribution of various attributes.

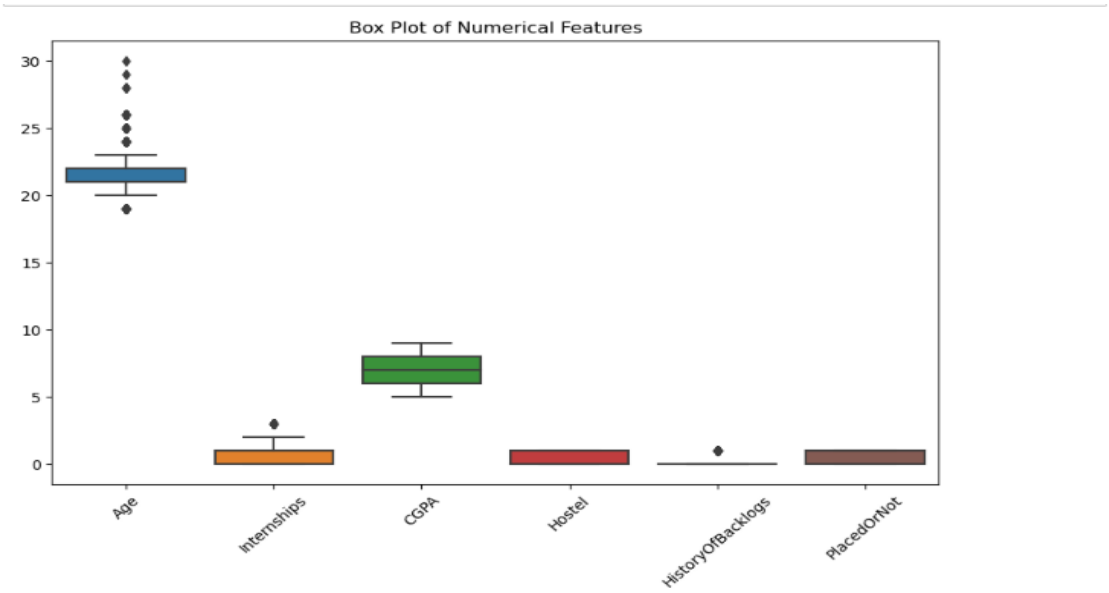




- In addition, we've included a snippet illustrating descriptive statistics for the numerical columns in the DataFrame.

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000
mean	21.485840	0.703641	7.073837	0.269049	0.192178	0.552596
std	1.324933	0.740197	0.967748	0.443540	0.394079	0.497310
min	19.000000	0.000000	5.000000	0.000000	0.000000	0.000000
25%	21.000000	0.000000	6.000000	0.000000	0.000000	0.000000
50%	21.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	22.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	30.000000	3.000000	9.000000	1.000000	1.000000	1.000000

- We've also included pictures of Box Plots to show how the data points are spread out in the dataset. Each box shows where most of the data points are, with a line inside showing the middle value. We saw that there are some outliers present for Age, internships, backlogs.

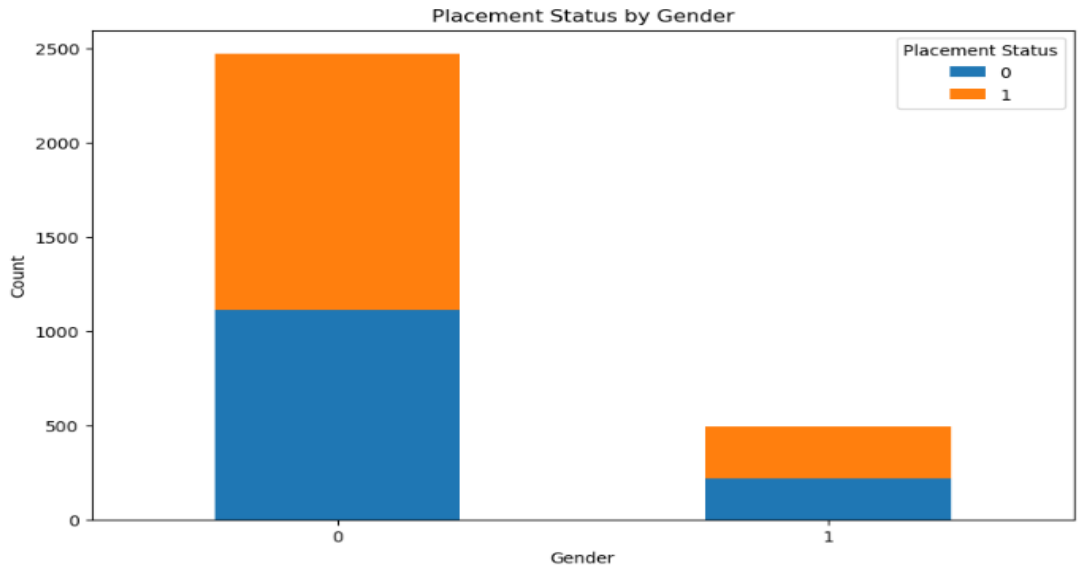


- We have chosen not to remove outliers from the dataset. This decision is based on the fact that our dataset reflects real-world scenarios where variations in student attributes, such as CGPA, internships, age, and backlogs, are expected and can be meaningful for placement prediction. Outliers in these attributes may represent exceptional cases or students with unique characteristics that remain relevant for predicting placements. By retaining these outliers, we aim to avoid biasing our understanding of the factors influencing placements and ensure that our predictions are as accurate as possible

  We analysed the Data Frame, grouping data by academic stream to calculate mean age, internships, CGPA, and placement rates. It then extracts unique stream values and examines placement counts per hostel. The results are compiled into a Data Frame named **stream_wise**.

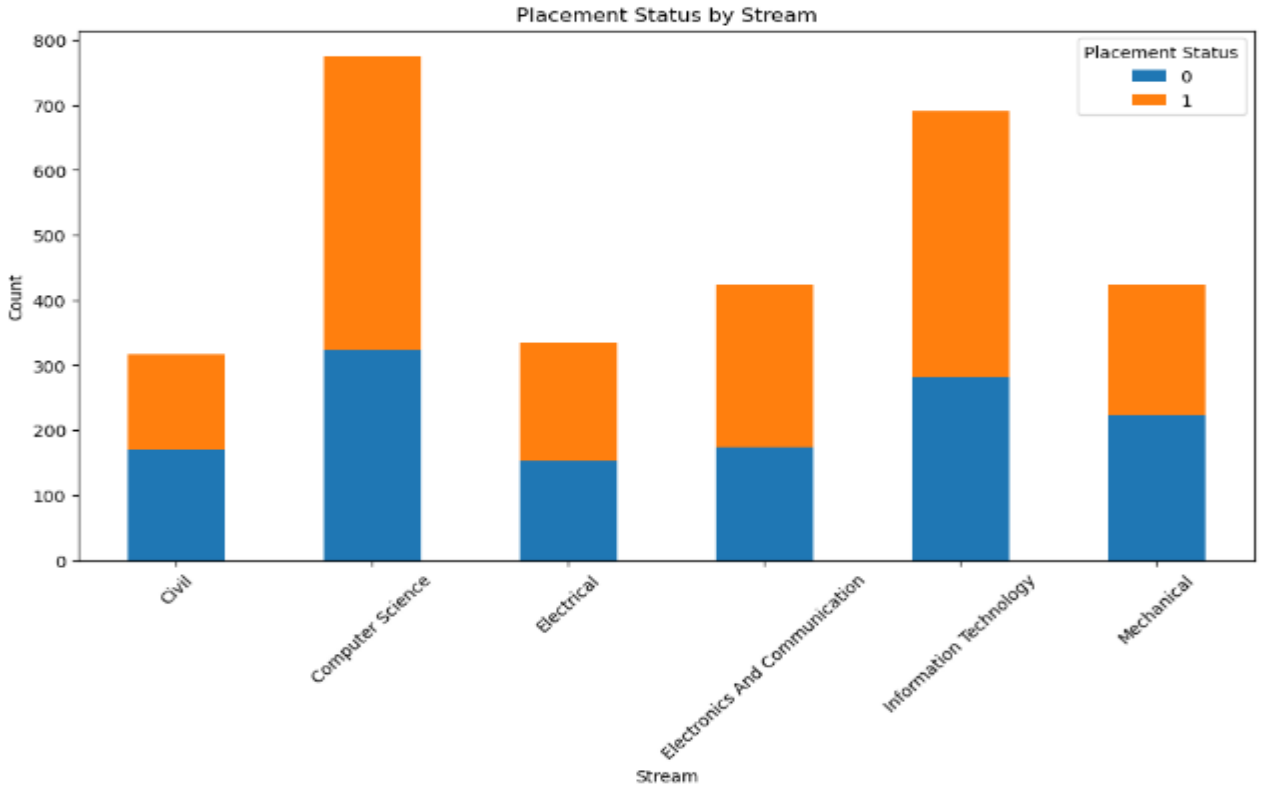
Stream	Age	Internships	CGPA	PlacedOrNot
Civil	21.441640	0.545741	7.094637	0.460568
Computer Science	21.559278	0.871134	7.039948	0.582474
Electrical	21.299401	0.607784	7.080838	0.541916
Electronics And Communication	21.410377	0.721698	7.125000	0.591981
Information Technology	21.539797	0.736614	7.073806	0.591896
Mechanical	21.518868	0.518868	7.063679	0.471698

- In the process of predicting student placement, we have considered all attributes to have equal weightage. These attributes include age, gender, stream of study, internship experience, hostel accommodation preference, and backlog status. However, it's important to note that each of these attributes has varying degrees of influence on the placement outcome, which are displayed individually after implementing the predictive model.

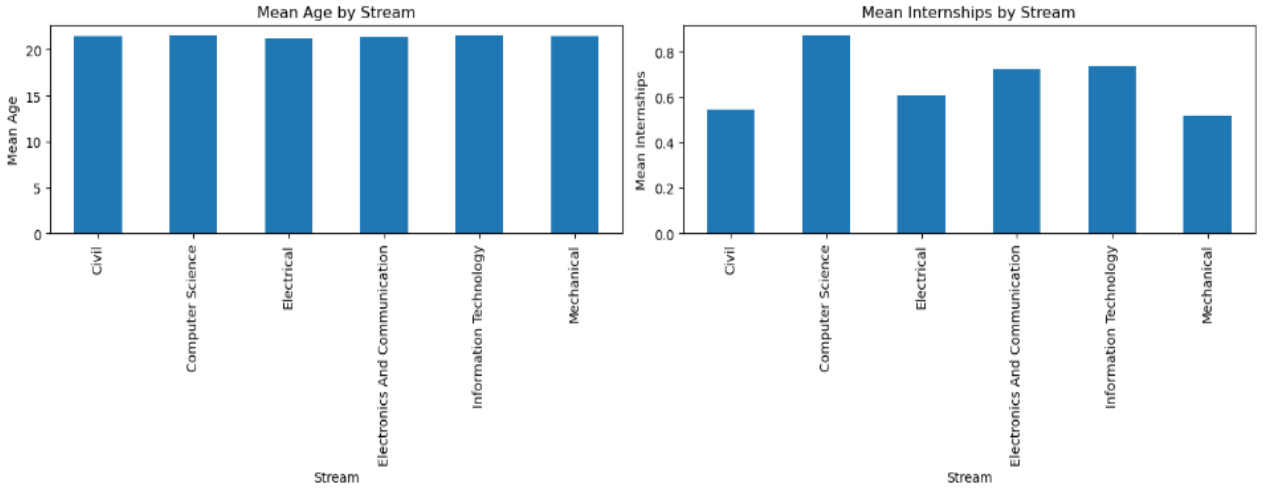


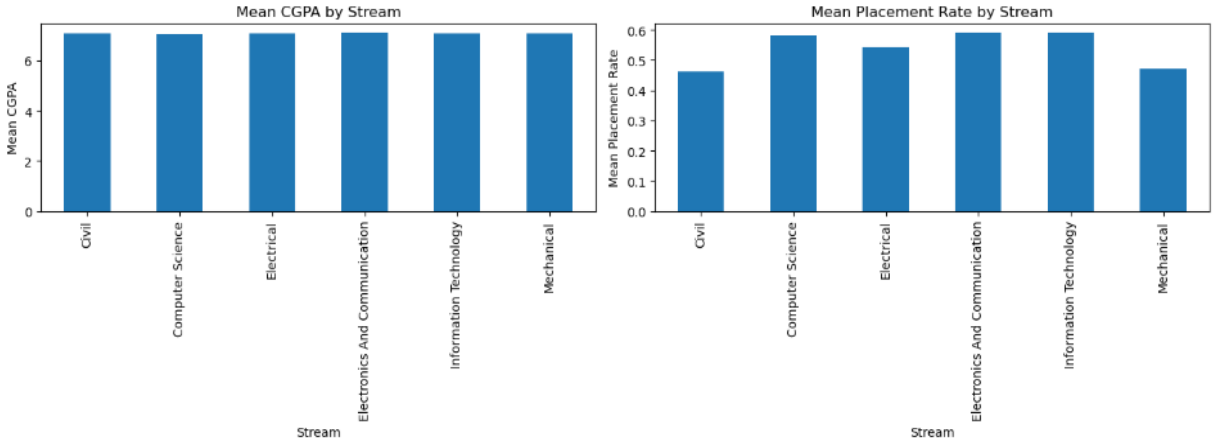
Here ‘0’ represent male candidates and ‘1’ represent female.

```
# Mapping gender to 0 and 1

gender_map = {'Male': 0, 'Female': 1}

df['Gender'] = df['Gender'].map(gender_map)
```



And based on the stream of study, histograms were generated to visualize the mean values of other attributes.



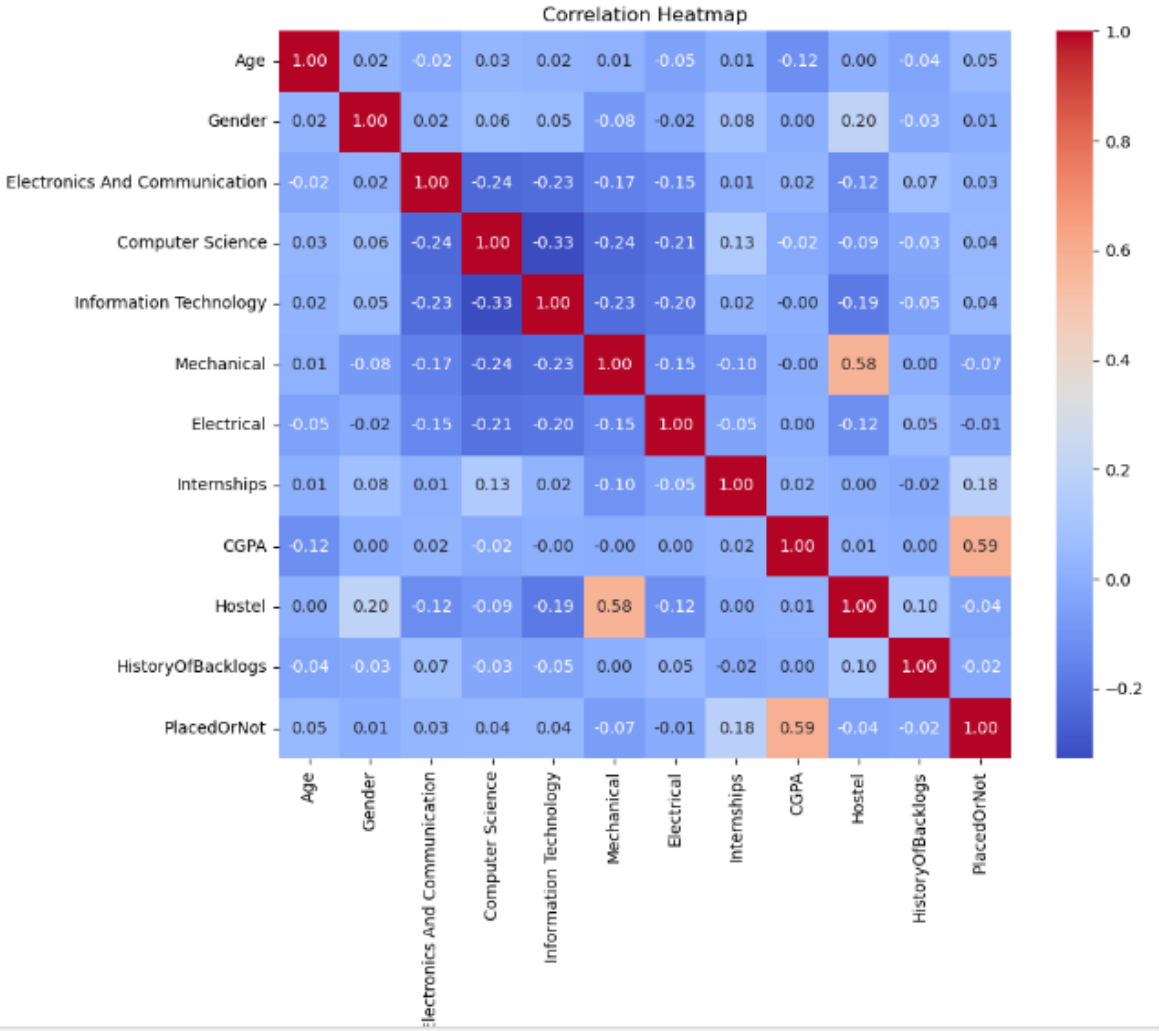


Since machine learning models are not comfortable with textual data, we need to convert the attribute “Stream”, which has 6 values, into numerical form by some way. For this, one-hot encoding was specifically applied to the 'Stream' column of the DataFrame **df**, representing different academic branches. Each branch was encoded as a binary column, where a value of 1 indicated its presence and 0 indicated absence. Notably, for the 'Civil' branch, all other branches were encoded as 0. This encoding scheme effectively prepares categorical data for subsequent analysis or machine learning tasks, ensuring accurate representation and model compatibility.

	Age	Gender	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot	Computer Science	Electrical	Electronics And Communication	Information Technology	Mechanical
0	22	0	1	8	1	1	1	0	0	1	0	0
1	21	1	0	7	1	1	1	1	0	0	0	0
2	22	1	1	6	0	0	1	0	0	0	1	0
3	21	0	0	8	0	1	1	0	0	0	1	0
4	22	0	0	8	1	0	1	0	0	0	0	1

- After conducting a feature selection analysis, we assessed the impact of removing less relevant features on model performance.
- Surprisingly, removing any feature, even those deemed less important, led to a noticeable reduction in model accuracy across several models.
- This observation indicates the crucial role of each feature in contributing to accurate predictions. Therefore, we have retained all features in our dataset, as their collective contribution is essential for maintaining optimal model performance, as confirmed by the feature selection process.

- We generated a correlation heatmap for the dataset, which visually depicts the relationships between different variables. Warm colors denote positive correlations, while cool colors signify negative correlations. This visualization is instrumental in identifying patterns, multicollinearity, and guiding feature selection decisions. We can clearly see that features like CGPA, internship has a high correlation with the student getting placed.



Next we performed Scaling using the Standard Scalar library of python. This was done because different features have different scales of measurements like age was around 20-23, while other numerical values were around 0-1 or so. Hence, to make sure that absolute values of features do not effect the way the model development is done, we use standard scalar .This transformation ensures that each feature has a mean of 0 and a standard deviation of 1, which can be useful for certain algorithms and helps in comparing features that have different scales.

2. **Model Development:**

- Now we move to our model development. From the analysis of the data and our prior knowledge, we selected various machine learning algorithms which we consider would be an effective choice. The algorithms are- Logistic Regression, Naïve bayes classifier, Random Forest classifier, support vector classifier, XGBoost classifier and K–nearest Neighbor classifier.

- For the purpose of model development, we need to split our data into training and testing part. The models will be trained on the training data and would be tested for accuracy on the test data. We used a split ratio of 80:20 for the same.

Next, we want to use grid search cross-validation to improve the model parameters for four distinct models: XGBoost, Random Forest, Support Vector Classifier (SVC), and Logistic Regression. We create a parameter grid for every model that lists the different hyperparameters that need to be evaluated. GridSearchCV is used to use 5-fold cross-validation to thoroughly search for the optimal combination of parameters. After the models are fitted to the training set, each model's optimal parameters and associated scores are printed. This methodology guarantees that the models are optimized for maximum efficacy, hence augmenting their precision in forecasting placement results.

(We did not used grid search-cv on the remaining two since from the literature, hyperparameter optimisation would not lead to a huge impact on their accuracies. Hence, we used them with default parameters.)

Its results are as follows:

```
Logistic Regression - Best Parameters: {'C': 10, 'penalty': 'l2'}
Logistic Regression - Best Score: 0.7588771929824562
Random Forest - Best Parameters: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}
Random Forest - Best Score: 0.8887141905396403
Support Vector Classifier - Best Parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
Support Vector Classifier - Best Score: 0.8726973129025094
XGBoost - Best Parameters: {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 100}
XGBoost - Best Score: 0.8882949145014434
```

3. **Model Training and Testing:**

- The ML models were trained on the training dataset to learn patterns and relationships between student attributes and placement outcomes.
- The trained models were then tested on the separate testing dataset to assess their predictive accuracy and generalization performance.

  After exploring the training data, such as through histogram analysis to understand the attributes' relationship with placement, we evaluated our predictive model on the remaining 20% of the dataset. Here are the outcomes of this testing phase, including accuracy metrics and the Classification Report.

- Accuracy by Logistic Regression : `0.782828282828282`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.77      0.76       271
           1       0.80      0.80      0.80       323

    accuracy                           0.78       594
   macro avg       0.78      0.78      0.78       594
weighted avg       0.78      0.78      0.78       594
```

- Accuracy by K-Nearest Neighbour : `0.8333333333333334`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.81      0.82       271
           1       0.84      0.85      0.85       323

    accuracy                           0.83       594
   macro avg       0.83      0.83      0.83       594
weighted avg       0.83      0.83      0.83       594
```

- Accuracy by Naïve Bayes : `0.7946127946127947`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.75      0.77       271
           1       0.80      0.83      0.81       323

    accuracy                           0.79       594
   macro avg       0.79      0.79      0.79       594
weighted avg       0.79      0.79      0.79       594
```

- Accuracy by Random Forest : `0.8804713804713805`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.90      0.87       271
           1       0.91      0.85      0.88       323

    accuracy                           0.88       594
   macro avg       0.88      0.88      0.88       594
weighted avg       0.88      0.88      0.88       594
```

- Accuracy by Support Vector : `0.8468013468013468`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.89      0.84       271
           1       0.90      0.81      0.85       323

    accuracy                           0.85       594
   macro avg       0.85      0.85      0.85       594
weighted avg       0.85      0.85      0.85       594
```

- Accuracy by XG- Boost : `0.8905723905723906`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.93      0.89       271
           1       0.94      0.85      0.89       323

    accuracy                           0.89       594
   macro avg       0.89      0.89      0.89       594
weighted avg       0.90      0.89      0.89       594


Confusion Matrix:
[[253  18]
 [ 47 276]]
```

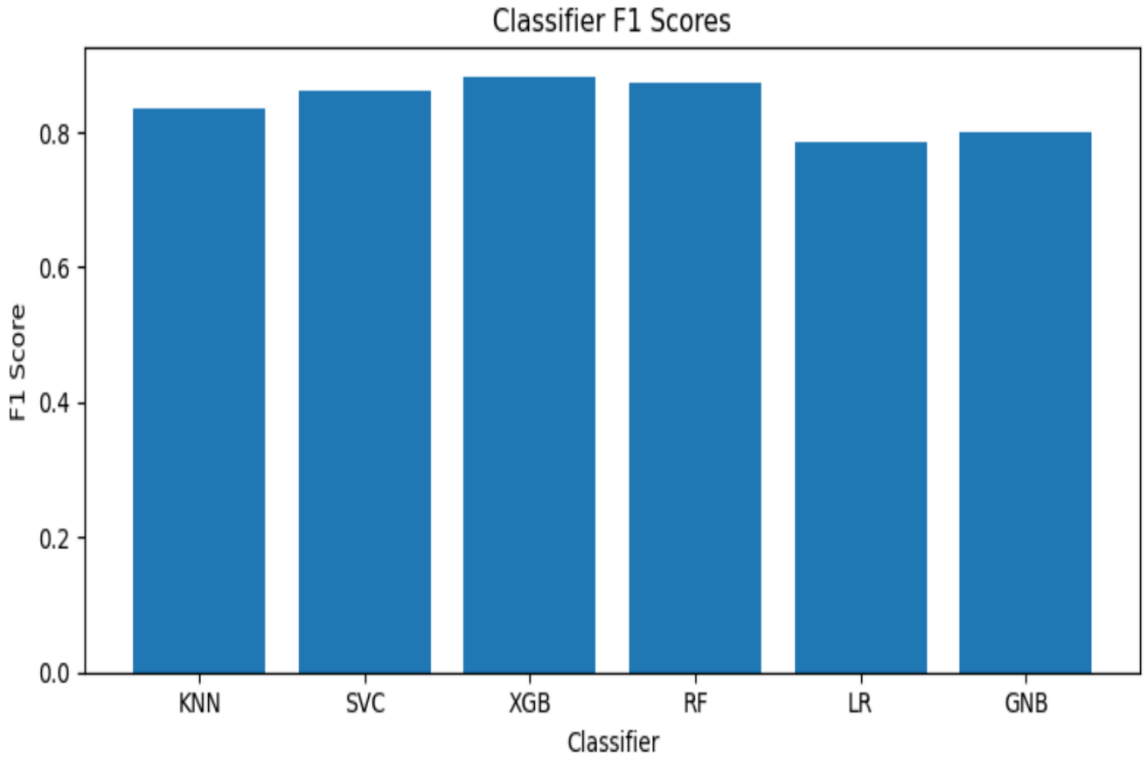
4. **Algorithm Comparison :**

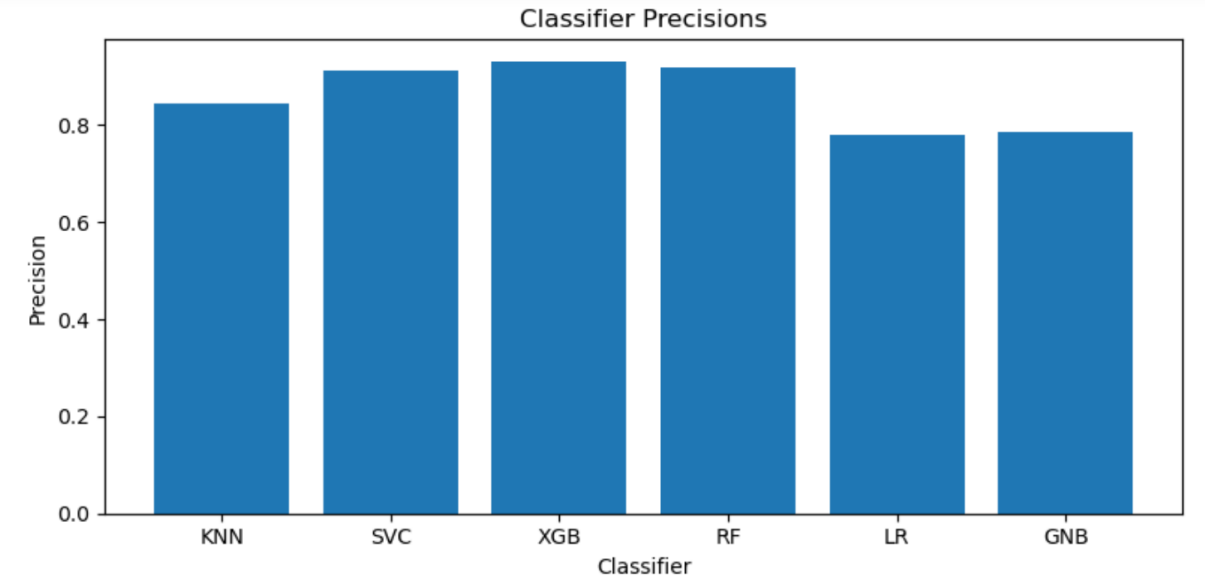
- The performance of different ML algorithms was compared using evaluation metrics such as accuracy, precision, recall, and F1-score.
- Each algorithm's strengths and weaknesses were identified based on its performance in predicting student placements.

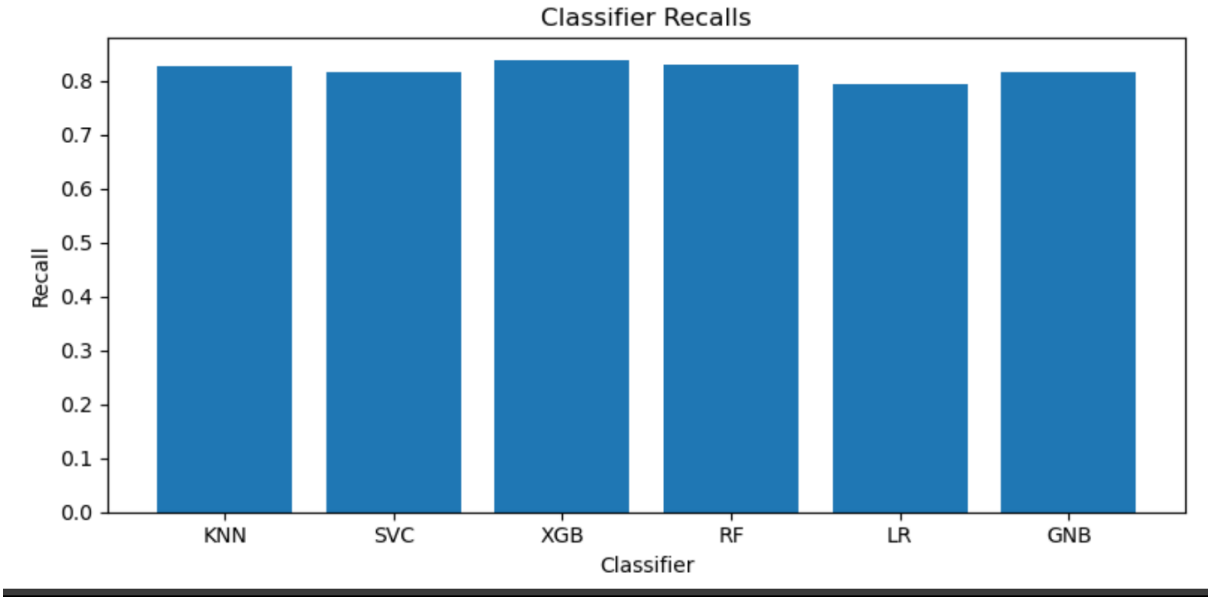
By applying Cross Validation on the following sets of models, we got the following result:

```
KNN Precision: 0.84495, Recall: 0.827943, F1 Score: 0.83636
SVC Precision: 0.91212, Recall: 0.816961, F1 Score: 0.86192
XGB Precision: 0.92953, Recall: 0.837095, F1 Score: 0.88089
RF Precision: 0.91902, Recall: 0.83099, F1 Score: 0.87279
LR Precision: 0.77904, Recall: 0.793776, F1 Score: 0.78634
GNB Precision: 0.78613, Recall: 0.81635, F1 Score: 0.80095
```

Its histogram is plotted as follows:







Conclusion:

In summary, XG Boost, Random Forest, and SVC had the greatest accuracy rates out of all the models tested. This implies that they were successful in predicting the goal variable—placement status in our case. These models performed well in identifying the underlying linkages and patterns in the dataset, which led to precise predictions about the placement of students. Their accomplishments demonstrate the effectiveness of sophisticated machine learning algorithms, such Random Forest and XG Boost, as well as the adaptability of conventional techniques, like SVC, in managing challenging prediction jobs. This emphasizes how crucial it is to use a variety of algorithms in order to optimize forecast performance and obtain thorough understanding of the variables affecting placement results.

Ensemble Model: Naive Bayes-Decision Trees Hybrid

As mentioned in the previous study, we have developed an ensemble model for classification that combines the decision-making power of Decision Trees with the probabilistic qualities of Naive Bayes. Utilizing this dataset to test the model yielded the following outcomes:

```
Ensemble Accuracy: 0.8804713804713805
```

We experimented with another model, which is a fusion of Logistic Regression (LR) and Naive Bayes (NB) to obtain he following result:

```
Accuracy: 0.8451178451178452
```

```
Although we had created the ensemble so as to get a higher accuracy, we
can clearly see that the simpler models like Random Forest classifier,
XGBoost etc are providing better accuracies. This might be due to the
Nature of the data on which the model is being trained.
```

Results and Analysis on the Test Dataset

Next, we'll assess the performance of our models on a completely new dataset. This dataset will feature a distribution different from the one our models were trained on. This evaluation will determine whether our models can generalize effectively. We'll utilize all the models trained on our original dataset to make predictions on this new data.

For this new dataset, we have conducted a comprehensive analysis similar to what was done for the training dataset. This includes visualizing histograms for different attributes, ensuring no missing values are present, grouping data by stream, generating box plots, performing one-hot encoding, and constructing a correlation heatmap. These analyses provide valuable insights into the distribution and relationships within the new dataset, facilitating further exploration and model preparation.
The visualizations for the new dataset are as follows:

- Descriptive statistics summarizing the central tendency

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot	Gender
count	500.00000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	21.49800	0.862000	7.044000	0.414000	0.370000	0.556000	0.300000
std	1.04699	0.827239	1.048935	0.493042	0.483288	0.497352	0.458717
min	19.00000	0.000000	4.000000	0.000000	0.000000	0.000000	0.000000
25%	21.00000	0.000000	6.000000	0.000000	0.000000	0.000000	0.000000
50%	21.50000	1.000000	7.000000	0.000000	0.000000	1.000000	0.000000
75%	22.00000	1.000000	8.000000	1.000000	1.000000	1.000000	1.000000
max	24.00000	4.000000	10.000000	1.000000	1.000000	1.000000	1.000000

- Verification that there are no missing values present in the dataset.

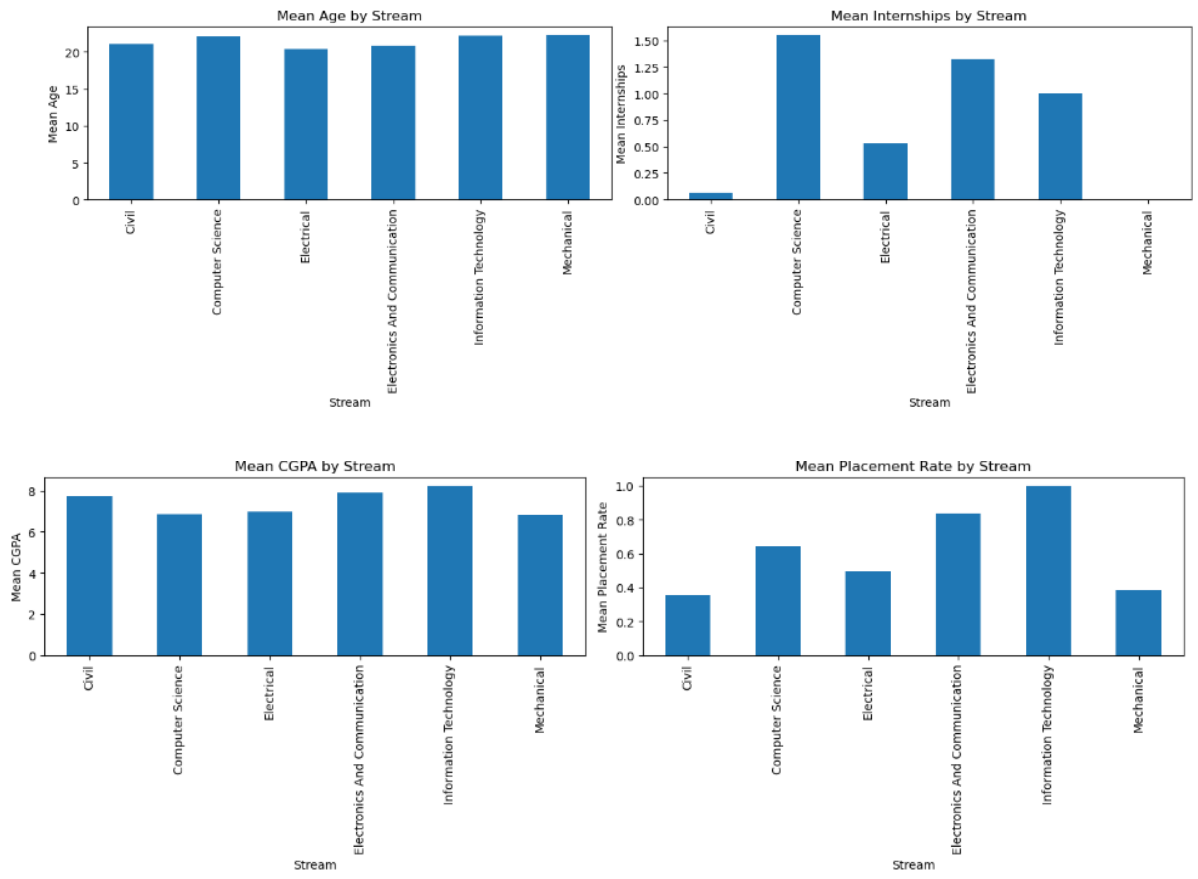
```
test_data_new.isnull().sum()
```

```
Age                 0
Internships         0
CGPA                0
Hostel              0
HistoryOfBacklogs   0
PlacedOrNot         0
Gender              0
Stream              0
dtype: int64
```

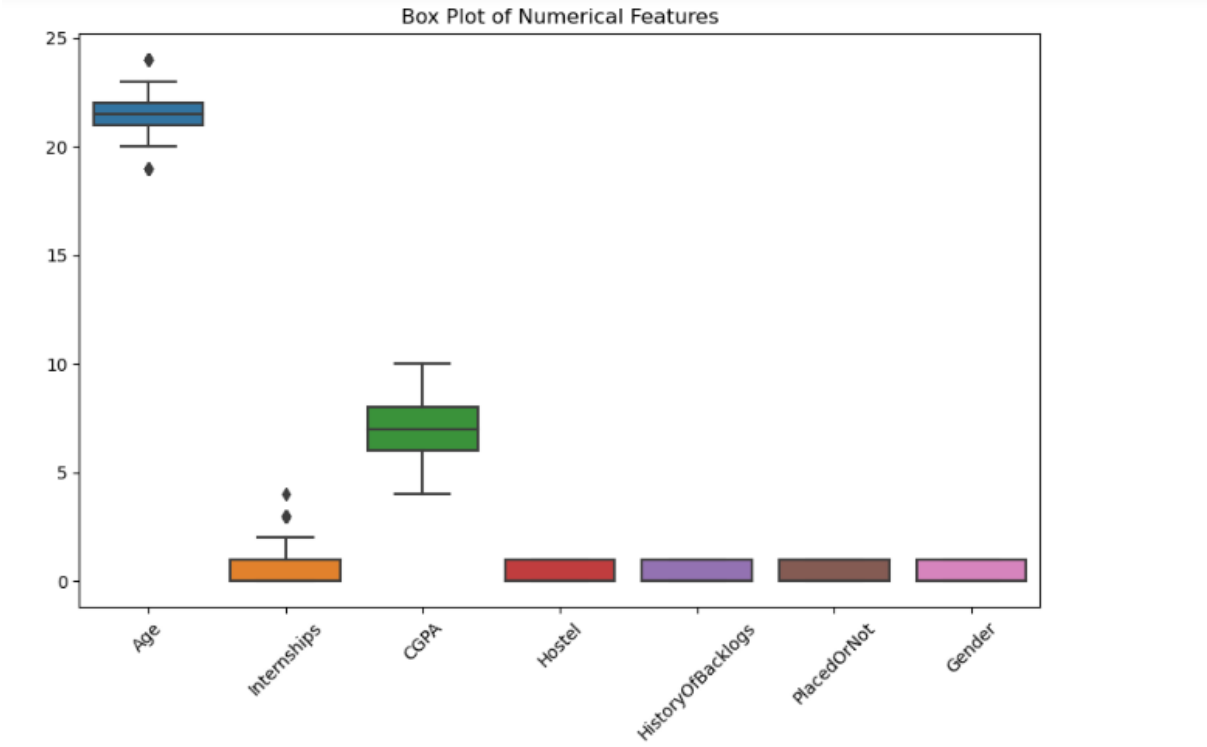
- This analysis provides insights into the average age, internships, CGPA, and placement rates across various academic streams.

Stream	Age	Internships	CGPA	PlacedOrNot
Civil	21.096774	0.064516	7.741935	0.354839
Computer Science	22.078534	1.549738	6.874346	0.643979
Electrical	20.408759	0.525547	6.978102	0.496350
Electronics And Communication	20.883721	1.325581	7.906977	0.837209
Information Technology	22.250000	1.000000	8.250000	1.000000
Mechanical	22.287234	0.000000	6.808511	0.382979

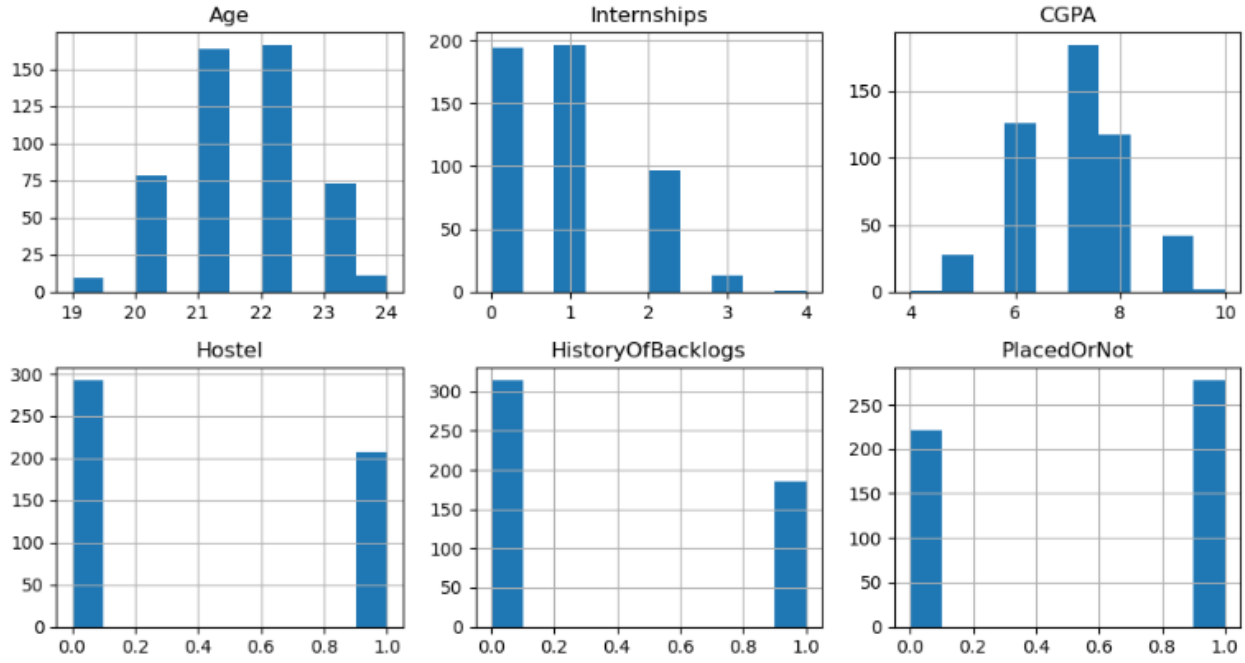
Snippet of subplot , each displaying a bar plot representing the mean values of different attributes grouped by 'Stream' in the testing dataset.

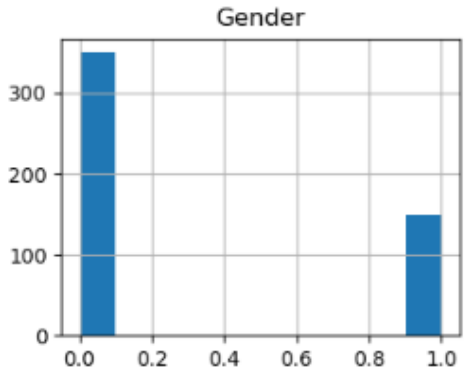


- Graphical depiction of the distribution of data across different categories or groups.



- Histograms: Visual representations of the distribution of different attributes.



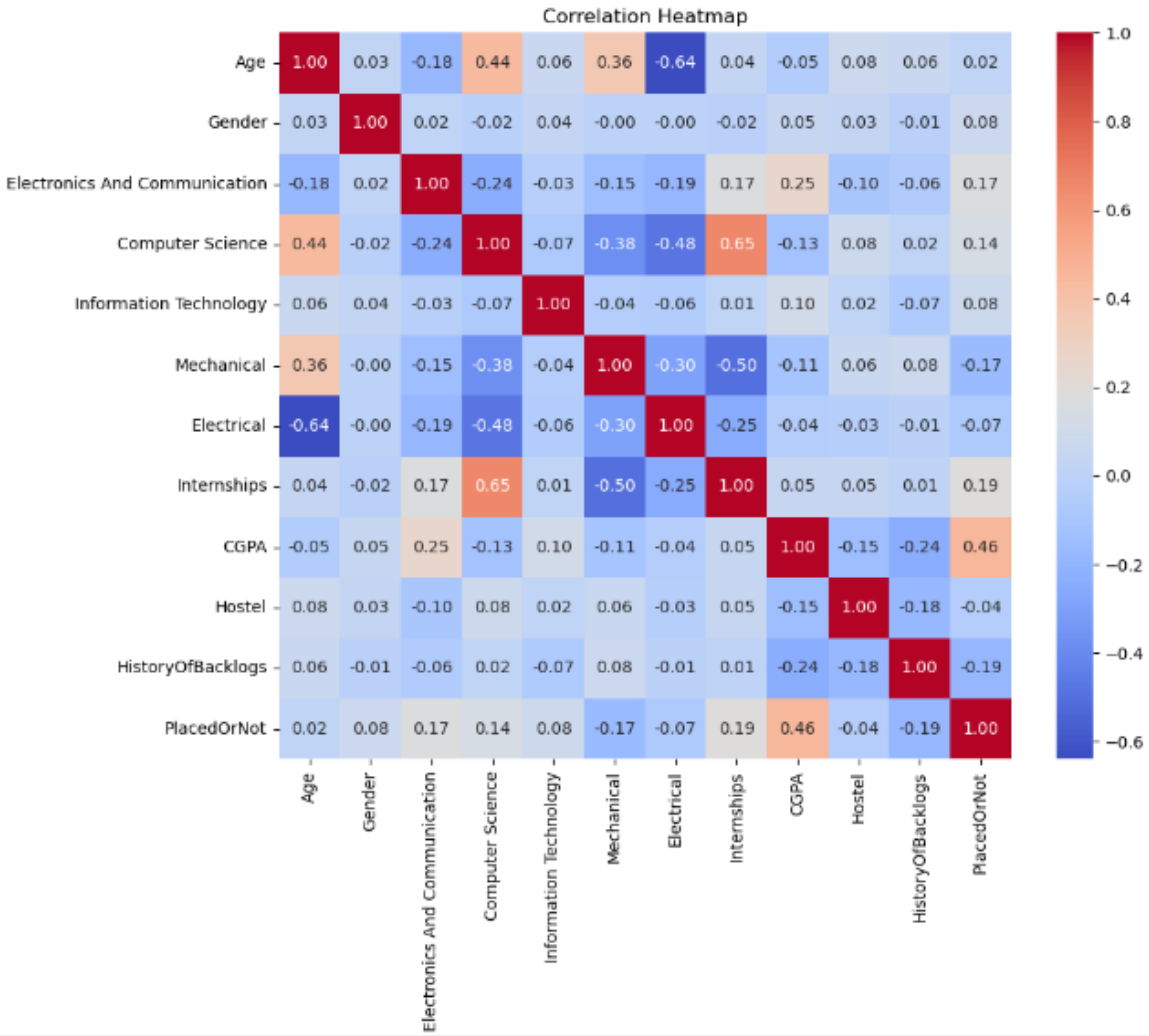


- One-Hot Encoding: Conversion of categorical variables into binary format for model compatibility.

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot	Gender	Computer Science	Electrical	Electronics And Communication	Information Technology	Mechanical
0	21	0	8	0	0	0	0	0	0	0	0	0
1	23	1	6	0	1	1	1	1	0	0	0	0
2	22	2	7	1	0	1	1	1	0	0	0	0
3	22	2	9	0	0	1	0	1	0	0	0	0
4	23	0	7	1	0	1	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
495	22	0	7	1	1	1	1	0	0	0	0	1
496	24	0	7	0	1	0	0	0	0	0	0	1
497	22	1	7	0	0	1	0	1	0	0	0	0
498	23	1	9	1	0	1	1	1	0	0	0	0
499	20	1	8	1	1	1	0	0	1	0	0	0

500 rows × 12 columns

- Correlation Heatmap: Visualization of pairwise correlations between variables



Finally, after conducting the aforementioned analyses and visualizations, we proceeded to test all the models trained on the original dataset on this new dataset. The outcomes of this evaluation are as follows:

- Accuracy by Logistic Regression : `0.714`

```
Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.71      0.69       222
           1       0.76      0.72      0.74       278

    accuracy                           0.71       500
   macro avg       0.71      0.71      0.71       500
weighted avg       0.72      0.71      0.71       500
```

- Accuracy by Naïve Bayes : `0.702`

Classification Report:

	precision	recall	f1-score	support
0	0.64	0.74	0.69	222
1	0.76	0.67	0.72	278
accuracy			0.70	500
macro avg	0.70	0.71	0.70	500
weighted avg	0.71	0.70	0.70	500

- Accuracy by K Nearest Neighbour : 0.646

Classification Report:

	precision	recall	f1-score	support
0	0.60	0.62	0.61	222
1	0.69	0.67	0.68	278
accuracy			0.65	500
macro avg	0.64	0.64	0.64	500
weighted avg	0.65	0.65	0.65	500

- Accuracy by Random Forest : 0.652

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.70	0.64	222
1	0.72	0.61	0.66	278
accuracy			0.65	500
macro avg	0.66	0.66	0.65	500
weighted avg	0.66	0.65	0.65	500

- Accuracy by SVR : 0.666

Classification Report:

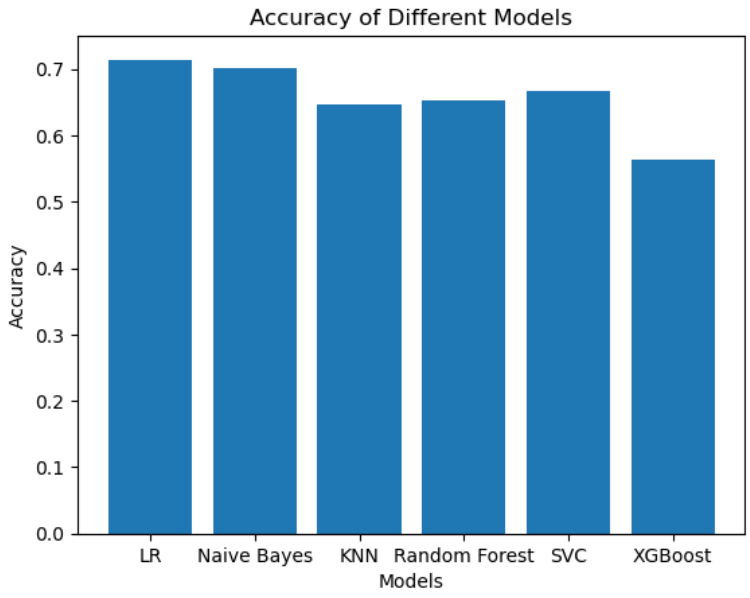
	precision	recall	f1-score	support
0	0.60	0.73	0.66	222
1	0.74	0.62	0.67	278
accuracy			0.67	500
macro avg	0.67	0.67	0.67	500
weighted avg	0.68	0.67	0.67	500

- Accuracy by XG Boost : 0.564

Classification Report:

	precision	recall	f1-score	support
0	0.50	0.95	0.66	222
1	0.87	0.26	0.39	278
accuracy			0.56	500
macro avg	0.69	0.60	0.53	500
weighted avg	0.71	0.56	0.51	500

Here is the graph illustrating the accuracy scores of different models :



Now testing the ensemble model on this dataset yielded the following result:

- `Ensemble Accuracy: 0.664`

We experimented with another model, which is a fusion of Logistic Regression (LR) and Naive Bayes (NB) to obtain he following result:

- `Ensemble Accuracy: 0.66`

Conclusion:

Thus, upon evaluation, our proposed ensemble models exhibited average performance on the given test dataset. Surprisingly, traditional models like Naive Bayes and Logistic Regression outperformed the ensemble model in this particular scenario. However, it's important to note that the performance of ensemble models can vary significantly depending on the distribution of both training and testing data. While they may not have excelled in this instance, it's conceivable that they could demonstrate superior performance on other datasets with different characteristics. Therefore, further experimentation and evaluation across diverse datasets are warranted to fully assess the effectiveness and versatility of ensemble models in placement prediction tasks.

FUTURE WORKS-

One of the possible additions that we would like to make in our project is the incorporation of time series analysis for the prediction of placements, taking into account the larger factors that impact it, like the economy, market situations, etc. Incorporating time-series analysis techniques will enable the examination of historical placement data over multiple years or semesters, providing insights into the dynamics of the job market, student preferences, and institutional performance. By organizing data into a time-series format and applying decomposition methods such as STL or Fourier transforms, underlying patterns and fluctuations in placement outcomes can be identified. Then, time-series forecasting models such as ARIMA (Autoregressive Integrated Moving Average), and SARIMA (Seasonal ARIMA) can be developed to predict future placement trends based on historical data

REFERENCES-

- https://www.kaggle.com/tejashvi14/engineering-placements-prediction

- Classification Model of Prediction for Placement of Students (researchgate.net)
- Collaborative job prediction based on Naïve Bayes Classifier using python platform | IEEE Conference Publication | IEEE Xplore
- https://www.researchgate.net/publication/361647065_Model_Construction_Using_ML_for_Prediction_of_Student_Placement
- https://www.researchgate.net/publication/258652670_Prediction_of_Final_Result_and_Placement_of_Students_using_Classification_Algorithm?enrichId=rgreq-5b2731b986aa2fc85cd8d5a68eac2082-XXX&enrichSource=Y292ZXJQYWdlOzI1ODY1MjY3MDtBUzozNTg1ODM5NDMzNTIzMjBAMTQ2MjUwNDQ4MjQ1Ng%3D%3D&el=1_x_3&_esc=publicationCoverPdf