Placement Prediction using Machine Learning

Broadly Focused on EDA and Data Mining Techniques

S M Asiful Islam Saky
30/01/2025

1. Introduction

The objective of this project is to predict the placement outcomes of engineering students based on their academic and demographic information. The problem being addressed involves identifying key factors that contribute to successful placements and building predictive models to assist both students and academic institutions in enhancing placement success rates. The expected outcome is a high-performing classification model that accurately predicts whether a student will be placed, providing actionable insights into placement trends. The dataset used for this project is sourced from Kaggle and contains comprehensive records of engineering students' attributes and placement outcomes. The dataset includes 2,966 rows and 8 columns. This dataset is highly relevant as it captures crucial academic and demographic factors influencing placements, providing a robust foundation for building predictive models. To address the problem, 4 data mining techniques have been selected: decision tree, random forest, xgboost, k-nearest neighbor(knn). These algorithms are chosen for their complementary strengths in handling classification tasks. Random Forest and XGBoost are expected to deliver high accuracy, while Decision Tree provides interpretability. KNN adds diversity to the model comparisons, particularly for non-linear decision boundaries.

2. Problem Identification & Dataset Selection

2.1 Dataset Selection

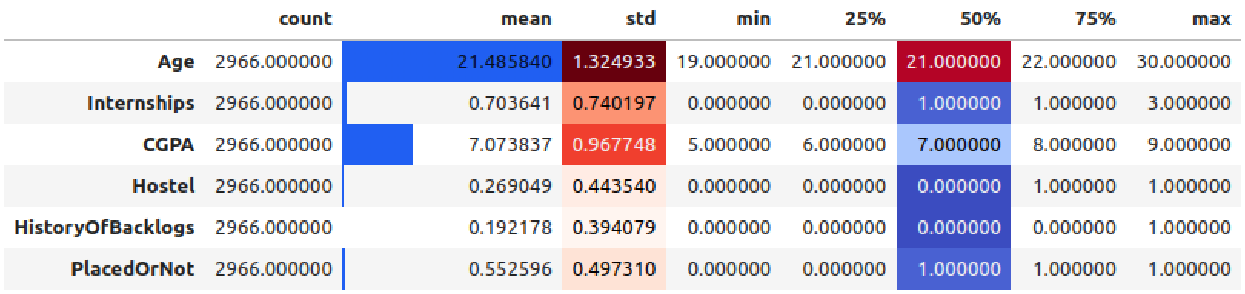
2.1.1 Dataset Description

The dataset used for this project, titled **Engineering Placement Prediction**, was sourced from **Kaggle**, a public data platform widely used for hosting datasets for machine learning and data science competitions. It contains data about engineering students and their placement outcomes, including academic, demographic, and extracurricular attributes. The dataset has the following characteristics:

- **Size**:
  - **Number of Records**: 2,966 rows (each row represents a unique student).
  - **Number of Features**: 8 columns (7 features and 1 target variable).
- **Features**:
  - **Age** (*Numerical - Integer*): The age of the student.
  - **Gender** (*Categorical - Object*): Gender of the student (Male or Female).

- **Stream** (*Categorical - Object*): Academic stream of the student, such as Computer Science, Mechanical, Electrical, etc.
- **Internships** (*Numerical - Integer*): The number of internships the student has completed.
- **CGPA** (*Numerical - Float*): The student's Cumulative Grade Point Average.
- **Hostel** (*Binary - Integer*): A binary variable (1 = student resides in a hostel, 0 = otherwise).
- **HistoryOfBacklogs** (*Binary - Integer*): A binary variable indicating whether the student has had academic backlogs (1 = yes, 0 = no).
- **PlacedOrNot** (*Binary - Integer*): The target variable, where 1 indicates the student was placed, and 0 indicates the student was not placed.

- **Data Types**:
  - **Structured data:** All features are structured and well-defined, with clear categorical and numerical variables.
  - **Mixed data types:** Includes numerical, categorical, and binary data, requiring appropriate preprocessing steps such as scaling and encoding.



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

Fig. 1. Statistical Information of the Dataframe

2.1.2 Relevance of the Dataset

This dataset is highly relevant for solving the defined problem as it encapsulates critical factors influencing the placement of engineering students. Academic performance (CGPA, backlogs), professional experience (internships), and demographic details (age, gender, stream) are well-recognized indicators of placement readiness. Additionally:

- The dataset provides a comprehensive view of students from various engineering disciplines, making the analysis generalized and scalable.
- The target variable (**PlacedOrNot**) directly aligns with the project objective of predicting placement outcomes, making the dataset an ideal choice for classification modeling.

2.2 Problem Definition

2.2.1 Problem Being Solved

The problem addressed in this project is a **classification problem**, where the goal is to predict whether a student will secure placement based on their academic, demographic, and extracurricular information. The project focuses on:

- **Input**: Features like CGPA, internships, age, and stream.
- **Output**: Binary classification (1 = Placed, 0 = Not Placed).

This classification model is designed to identify key predictors of placement outcomes and accurately forecast the likelihood of a student being placed.

2.2.2 Real-World Significance

The ability to predict placement outcomes has significant business and educational implications:

1. **For Educational Institutions**:
    - **Improved Career Guidance**: Universities can identify students at risk of not being placed and offer targeted interventions, such as skill development programs, resume workshops, or mock interviews.
    - **Enhanced Curriculum Planning**: Insights from the model can guide changes in the curriculum to better align with industry requirements.
2. **For Students**:
    - **Personalized Support**: Students can focus on improving specific areas (e.g., internships, CGPA) identified as critical predictors for placement success.
    - **Early Awareness**: Students can make informed decisions about participating in internships and extracurricular activities to improve their placement chances.
3. **For Recruiters**:
    - **Streamlined Hiring**: Recruiters can use insights from the analysis to identify ideal candidates for their roles, saving time and resources in the hiring process.

Solving this problem contributes to bridging the gap between academia and industry, ensuring better preparedness among graduates and improved hiring efficiency.

3. Algorithm Justification

3.1 Chosen Algorithms

The following machine learning algorithms have been selected for solving the classification problem and also to compare among them:

1. **Decision Tree Classifier**
2. **Random Forest Classifier**
3. **XGBoost Classifier**
4. **K-Nearest Neighbors (KNN) Classifier**

3.2 Justification for Algorithm Selection

1. **Decision Tree Classifier**:
   - **Reason for Selection**: Decision Trees are highly interpretable models that can map feature interactions and thresholds in an intuitive tree-like structure. This makes them particularly useful for understanding how factors like CGPA or internships influence placement.
   - **Suitability**: Decision Trees handle mixed data types (numerical, categorical, binary) seamlessly and perform well with small to medium-sized datasets, like the one used here.
2. **Random Forest Classifier**:
   - **Reason for Selection**: As an ensemble method, Random Forest combines multiple decision trees to improve accuracy and reduce overfitting. It provides robust predictions and can handle datasets with a mix of feature types effectively.
   - **Suitability**: Random Forest is known for its versatility and high performance in classification problems, making it a reliable choice for predicting placement outcomes.
3. **XGBoost Classifier**:
   - **Reason for Selection**: XGBoost (Extreme Gradient Boosting) is a highly efficient and accurate gradient-boosting algorithm that excels in handling tabular datasets. It includes advanced features like regularization and handling missing data, enhancing its predictive power.
   - **Suitability**: Given the moderate size of the dataset and the goal of maximizing accuracy, XGBoost is a suitable choice for leveraging its optimization capabilities to fine-tune the model's performance.
4. **K-Nearest Neighbors (KNN) Classifier**:
   - **Reason for Selection**: KNN is a non-parametric algorithm that classifies data points based on their proximity to other labeled data points. It is straightforward to implement and provides competitive performance for smaller datasets.

    - **Suitability**: KNN is included for comparison purposes, as it may identify non-linear patterns in the dataset, particularly in features like CGPA and internships.

3.3 General Justification

- The combination of interpretable models (Decision Tree), ensemble methods (Random Forest, XGBoost), and non-parametric methods (KNN) ensures a comprehensive evaluation of the dataset and problem.
- These algorithms are widely recognized for their effectiveness in binary classification tasks, particularly with structured datasets like this one.

4. Exploratory Data Analysis (EDA)

4.1 Statistical Analyses and Visualizations

I. Age: A histogram showed that most students are aged 21-23, with a mean age of 21.5 years.

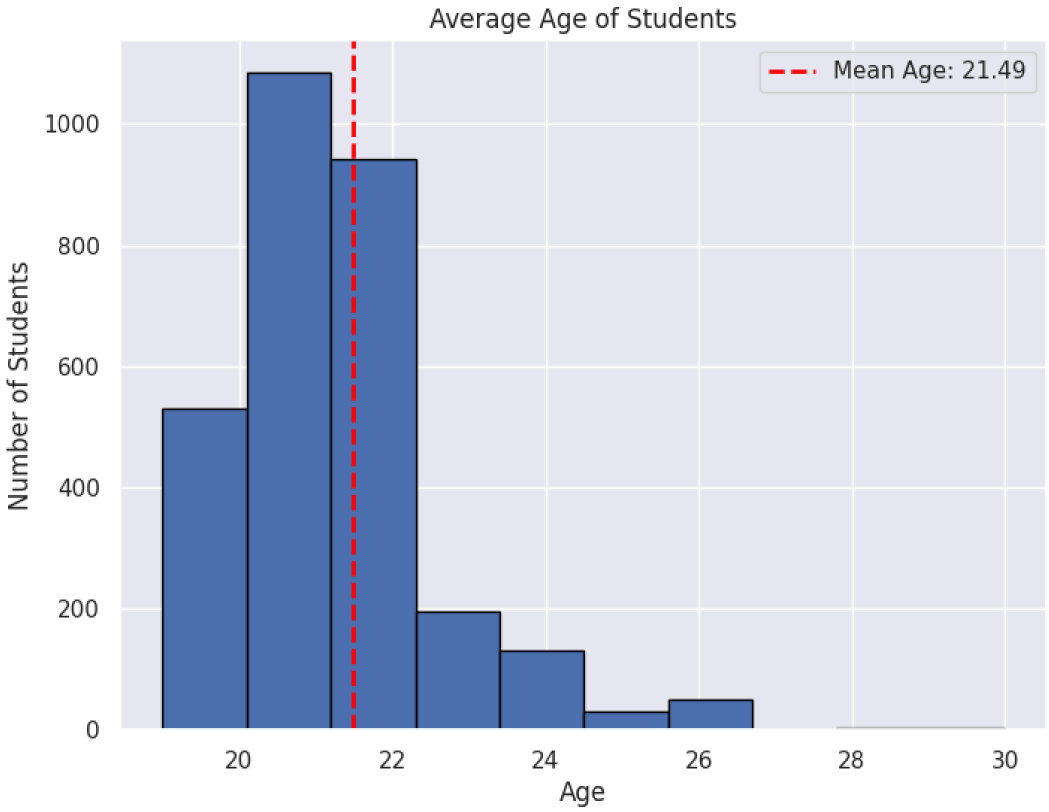


Fig. 2. Average Age of Students

The histogram above represents the age distribution of students in the dataset. The x-axis denotes the age range, while the y-axis indicates the number of students. The red dashed line represents the mean age of **21.49 years**, highlighting the central tendency of the data.

From the visualization, we observe that most students fall between the ages of **20 to 22**, with a significant decline in the number of students as the age increases beyond **23 years**. This suggests that the majority of students in the dataset are relatively young, which may have implications for placement outcomes. Employers may favor younger graduates, or specific age groups may have different placement success rates.

II. CGPA: CGPA values were analyzed using histograms and box plots. Students with CGPA above the average (7.07) had a higher likelihood of placement, as seen in the placement distribution.

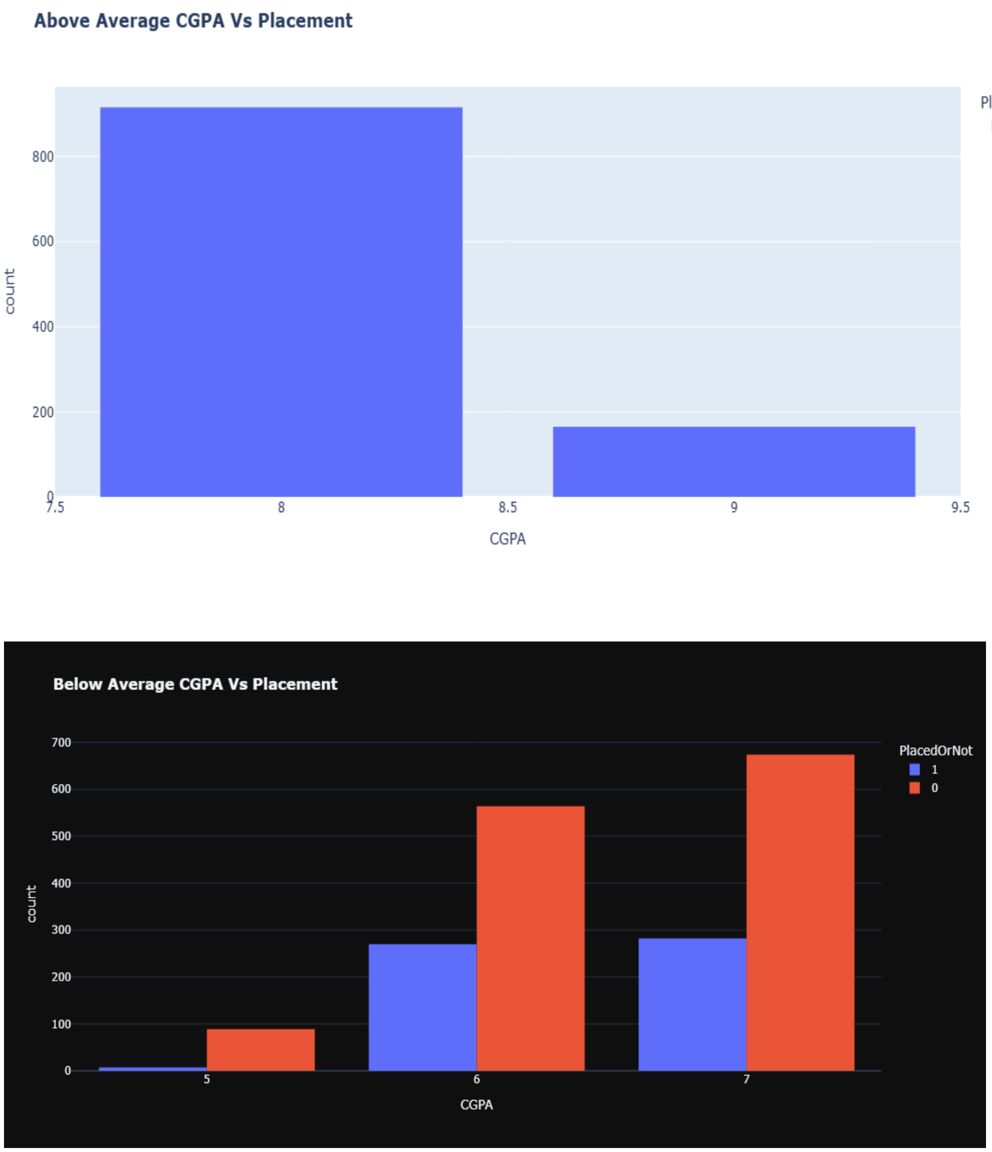


Fig. 3. Average CGPA vs Placement

III. Internships: A histogram showed that the majority of students (approximately 50%) have no internships, with placement likelihood improving for students with at least one internship.

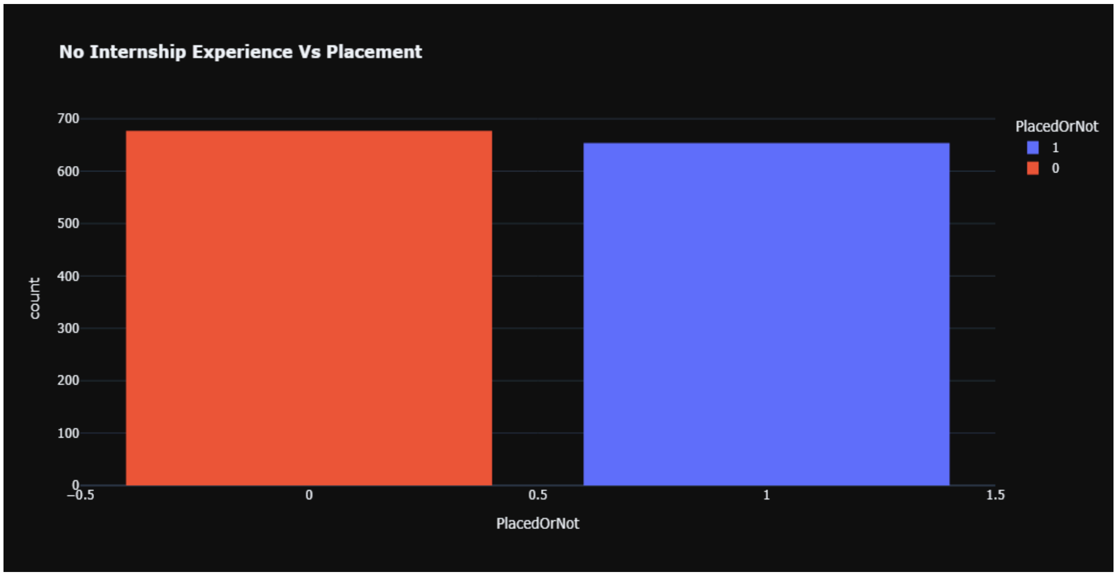


Fig. 4. Internship Experience vs Placement

IV. Gender Analysis: A **pie chart** revealed that 83.4% of the students are male, and 16.6% are female. Placement rates for males (55.11%) and females (56.01%) were nearly equal, as seen in a placement-by-gender histogram.

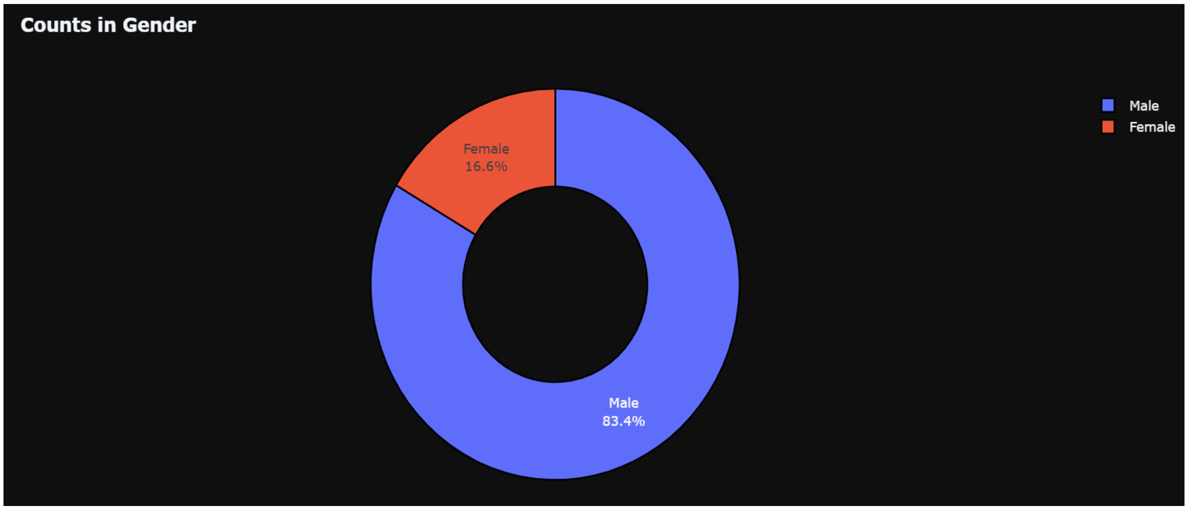


Fig. 5. Students Number Counts in Gender

V. Placement and Stream Analysis: A bar chart showed that students from **Computer Science** had the highest placement rates, followed by **Information Technology**. Streams like **Civil** and **Mechanical** had comparatively lower placement rates.

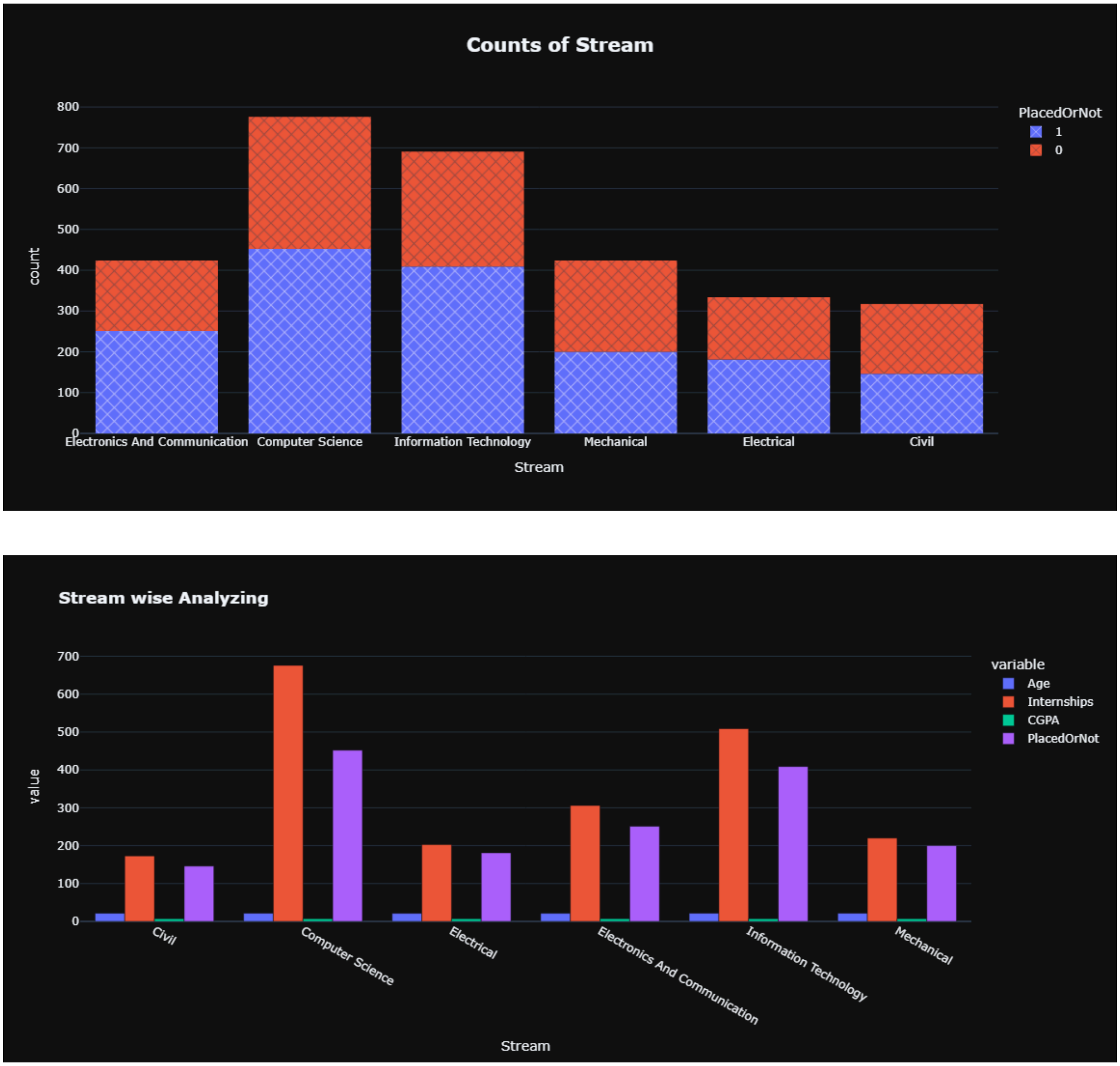


Fig. 6. Counts of Streams

4.2 Correlation Analysis: A **heatmap** highlighted the correlation between features and the target variable (PlacedOrNot). The strongest positive correlations were observed for:

- **CGPA**: Higher CGPA strongly correlates with placement.
- **Internships**: Completing internships positively impacts placement likelihood.

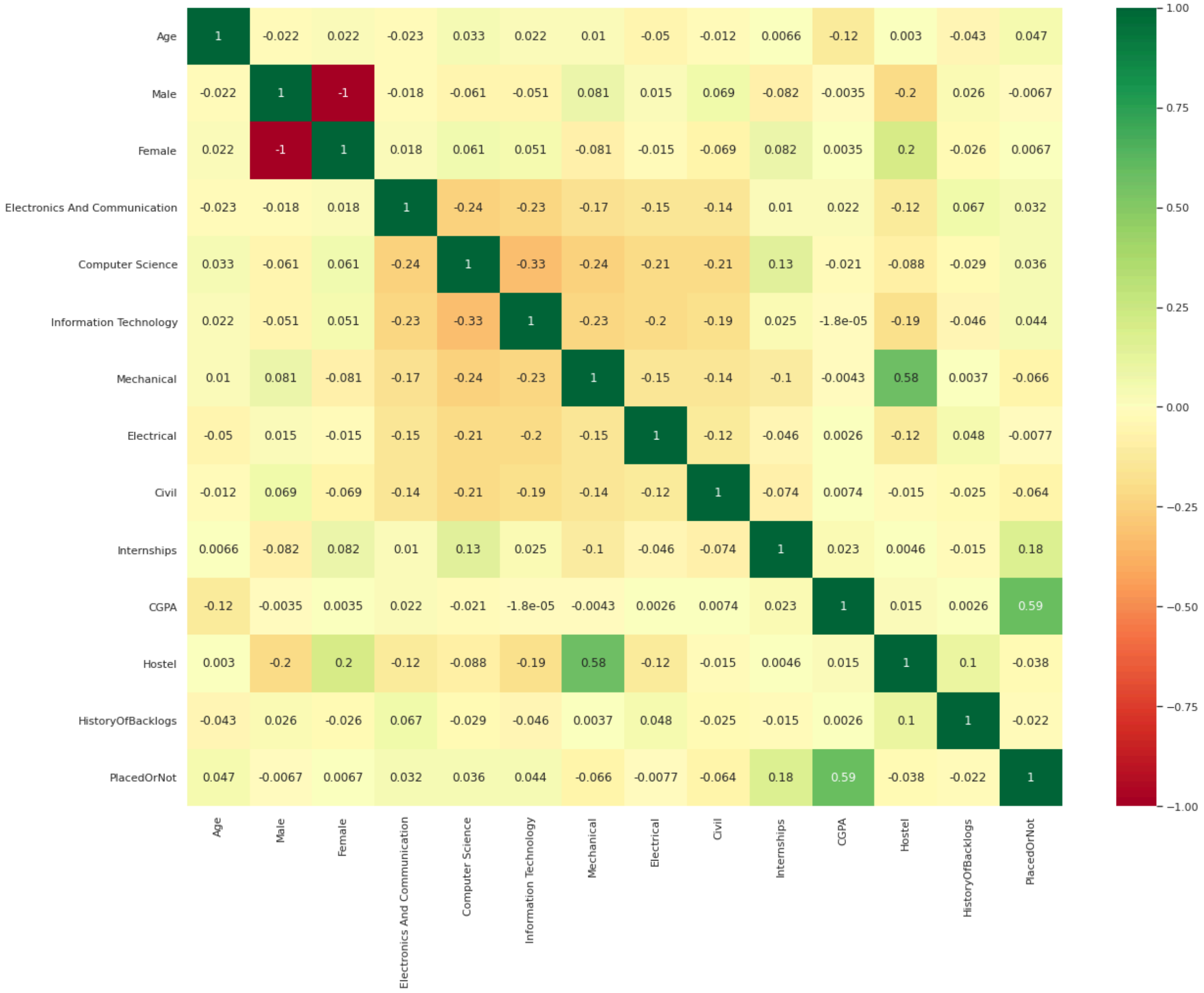


Fig. 7. Correlation Analysis(Heatmap)

Key Insights

- **Higher CGPA correlates positively with placement outcomes**: Students with CGPA > 7.07 had a significantly higher placement rate.
- **Internship experience boosts placement chances**: Students with at least one internship were more likely to be placed.
- **Stream affects placement likelihood**: Students from Computer Science and IT streams had better placement rates, suggesting a higher demand for these skills in the job market.
- **Gender parity**: Placement rates were similar for males and females, indicating gender-neutral hiring practices.

Impact on Modeling

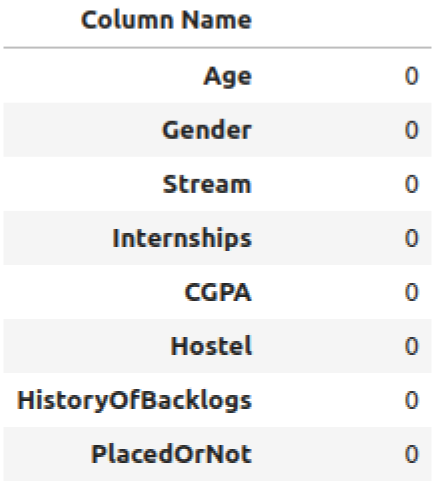
- EDA informed feature importance, suggesting that CGPA, Internships, and Stream are critical predictors for placement.
- It also highlighted that categorical variables (Gender, Stream) need encoding, and numerical features (CGPA, Age) require scaling for uniformity.

5. Data Preprocessing

5.1 Data Cleaning

1. **Missing Values**: The dataset was checked for missing values using

data.isnull().sum().



Column Name	
Age	0
Gender	0
Stream	0
Internships	0
CGPA	0
Hostel	0
HistoryOfBacklogs	0
PlacedOrNot	0

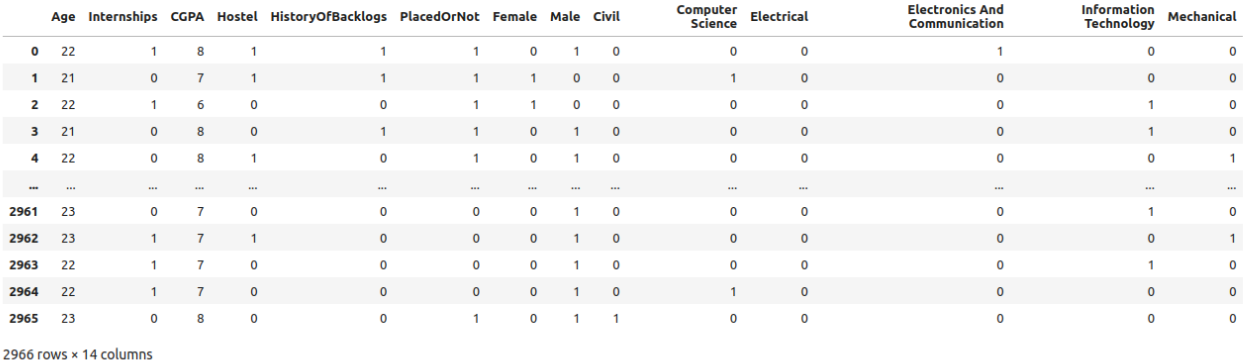
No missing values were found in any column, indicating a complete dataset ready for analysis.

2. **Outliers**: Outliers were analyzed during EDA, particularly in numerical columns like **Age**, **CGPA**, and **Internships**.
   - **Age**: No extreme outliers were observed, as most students are aged 21-23, which is typical for engineering students.
   - **CGPA**: While values are mostly between 5 and 9, lower CGPA values could indicate potential outliers; however, they align with realistic academic performance.
   - **Internships**: Most students have 0 to 2 internships, with a few having 3 or more. These higher values were retained as they provide meaningful diversity in the data.

3. **Inconsistencies**:
    - Categorical variables (**Gender**, **Stream**) were reviewed for consistency in naming conventions. No inconsistencies (e.g., typos or incorrect labels) were found.

5.2 Data Transformation

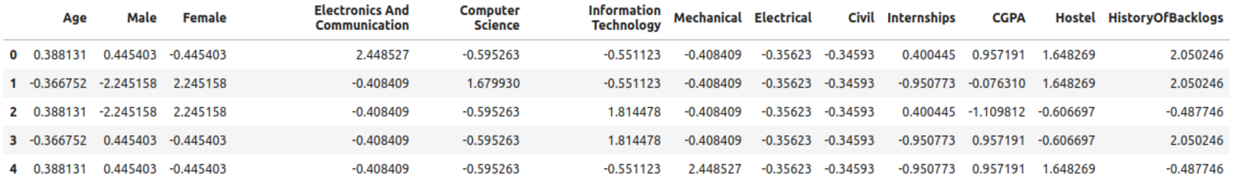
1. **Encoding Categorical Variables**:
    - **One-Hot Encoding**:
        - Applied to **Gender** and **Stream** to convert them into numerical format. For example:
            - Gender was split into Male and Female columns (binary encoding).
            - Stream was encoded into separate columns like Computer Science, Information Technology, etc.
        - Justification: One-Hot Encoding ensures that the model can interpret categorical data without introducing bias (e.g., ordinal relationships where none exist).



	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot	Female	Male	Civil	Computer Science	Electrical	Electronics And Communication	Information Technology	Mechanical
0	22	1	8	1	1	1	0	1	0	0	0	1	0	0
1	21	0	7	1	1	1	1	0	0	1	0	0	0	0
2	22	1	6	0	0	1	1	0	0	0	0	0	1	0
3	21	0	8	0	1	1	0	1	0	0	0	0	1	0
4	22	0	8	1	0	1	0	1	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2961	23	0	7	0	0	0	0	1	0	0	0	0	1	0
2962	23	1	7	1	0	0	0	1	0	0	0	0	0	1
2963	22	1	7	0	0	0	0	1	0	0	0	0	1	0
2964	22	1	7	0	0	0	0	1	0	1	0	0	0	0
2965	23	0	8	0	0	1	0	1	1	0	0	0	0	0

2966 rows × 14 columns

2. **Scaling Numerical Data**:
    - **StandardScaler**:
        - Applied to features like **Age**, **CGPA**, and **Internships** to normalize them to a mean of 0 and standard deviation of 1.
        - Justification: Scaling ensures that features with larger ranges (e.g., CGPA vs. Internships) don't disproportionately influence the model's predictions.



	Age	Male	Female	Electronics And Communication	Computer Science	Information Technology	Mechanical	Electrical	Civil	Internships	CGPA	Hostel	HistoryOfBacklogs
0	0.388131	0.445403	-0.445403	2.448527	-0.595263	-0.551123	-0.408409	-0.35623	-0.34593	0.400445	0.957191	1.648269	2.050246
1	-0.366752	-2.245158	2.245158	-0.408409	1.679930	-0.551123	-0.408409	-0.35623	-0.34593	-0.950773	-0.076310	1.648269	2.050246
2	0.388131	-2.245158	2.245158	-0.408409	-0.595263	1.814478	-0.408409	-0.35623	-0.34593	0.400445	-1.109812	-0.606697	-0.487746
3	-0.366752	0.445403	-0.445403	-0.408409	-0.595263	1.814478	-0.408409	-0.35623	-0.34593	-0.950773	0.957191	-0.606697	2.050246
4	0.388131	0.445403	-0.445403	-0.408409	-0.595263	-0.551123	2.448527	-0.35623	-0.34593	-0.950773	0.957191	1.648269	-0.487746

3. **Feature Selection**:
   - All features were retained after correlation analysis confirmed their relevance to the target variable.
   - Justification: Retaining all features ensures that the model can leverage the complete dataset for improved prediction accuracy.

Importance of Transformations

- **Encoding**: Converts categorical data into a format compatible with machine learning models, ensuring unbiased treatment of categories.
- **Scaling**: Normalizes feature ranges, improving model convergence and ensuring fair weight allocation across features during training.
- **Retention of Features**: Maintains the diversity of inputs, allowing the model to capture nuanced relationships between features and placement outcomes.

The dataset was clean and required minimal preprocessing. EDA provided valuable insights into placement trends, influencing the modeling process by identifying key predictors. Transformations like encoding and scaling ensured the dataset was ready for training high-performing classification models. Let me know if you'd like additional refinements!

6. Model Building

6.1 Experiment Design

1. **Train-Test Split**:
   - The dataset was divided into **80% training data (2,224 records)** and **20% test data (742 records)** using train_test_split from sklearn.
   - Justification: A 80-20 split ensures a sufficient amount of data for both training and testing, balancing model learning and evaluation.
2. **Hyperparameter Tuning**:
   - **RandomizedSearchCV** was used to optimize the hyperparameters of the **XGBoost Classifier**. This approach searches for the best parameter combination from a predefined grid in a randomized manner, saving computation time.
   - **Parameters Tuned**:
     - learning_rate: [0.05, 0.10, 0.15, 0.20, 0.25, 0.30].
     - max_depth: [3, 4, 5, 6, 8, 10, 12, 15].
     - min_child_weight: [1, 3, 5, 7].
     - gamma: [0.0, 0.1, 0.2, 0.3, 0.4].
     - colsample_bytree: [0.3, 0.4, 0.5, 0.7].
   - **Best Parameters Found**:

        - learning_rate: 0.1.
        - max_depth: 8.
        - min_child_weight: 1.
        - gamma: 0.0.
        - colsample_bytree: 0.7.
    - Time Taken: Approximately 3 minutes and 34 seconds for 5 iterations.

6.2 Model Implementation

1. **Chosen Algorithms**:
    - **Decision Tree Classifier**.
    - **Random Forest Classifier**.
    - **XGBoost Classifier** (with and without hyperparameter tuning).
    - **K-Nearest Neighbors (KNN) Classifier**.
2. **Steps Taken**:
    - **Model Training**:
        - Each algorithm was instantiated and trained using the training data (X_train, y_train).
        - For XGBoost, the best hyperparameters identified during tuning were applied before training.
    - **Model Scoring**:
        - Each trained model was evaluated using the test data (X_test, y_test) to compute accuracy scores.
3. **Performance Metrics**:
    - For each algorithm, accuracy scores were calculated. Metrics like Precision, Recall, and F1-measure were not explicitly mentioned in the implementation but could be added for a more detailed evaluation.

7. Model Evaluation

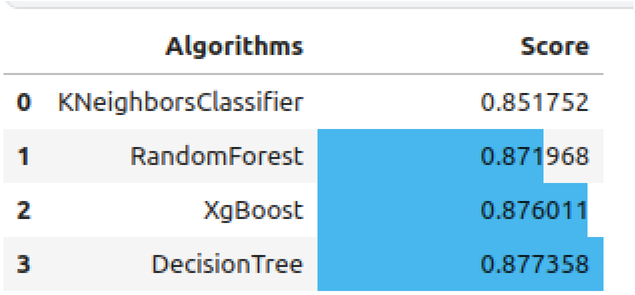
1. **Performance Table**:

- The models achieved the following accuracy scores:

Algorithm	**Accuray**
K-Nearest Neighbors	85.18%
XGBoost (tuned)	87.60%
Decision Tree	87.73%

Random Forest	87.73%

2. **Comparative Analysis:** The Decision Tree algorithm emerged as the top performer with a score of 0.877358, indicating the strongest predictive accuracy. XgBoost closely followed with a score of 0.876011, suggesting high predictive capabilities. Random Forest and KNeighbors Classifier achieved lower scores of 0.871968 and 0.851752, respectively, indicating moderate performance relative to the top two algorithms.



	Algorithms	Score
0	KNeighborsClassifier	0.851752
1	RandomForest	0.871968
2	XgBoost	0.876011
3	DecisionTree	0.877358

7.1 Visualizations

1. **Bar Graph for Model Accuracy**:
    - ○ A **bar chart** was created to compare the accuracy of the models visually.
    - ○ The chart clearly illustrates the relative performance of the models, with Decision Tree and Random Forest standing out as top performers.

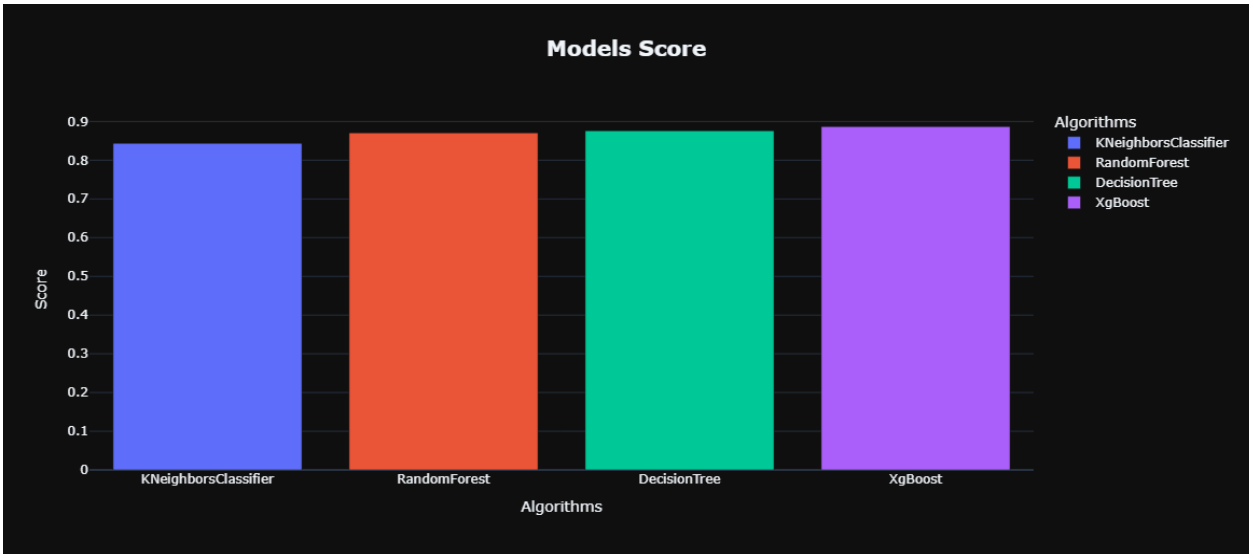


Fig. 8. Model Accuracy Bar Graph

Bar Chart Description:

- X-axis: Names of the algorithms (KNN, XGBoost, Decision Tree, Random Forest).
- Y-axis: Accuracy scores (in percentage).
- Each bar is color-coded to represent a specific model for clarity.

2. **Pie Chart for Model Accuracy Distribution**:
   - A **pie chart** was used to show the proportion of accuracy among the models, helping visualize how the models contributed to the overall predictive performance.

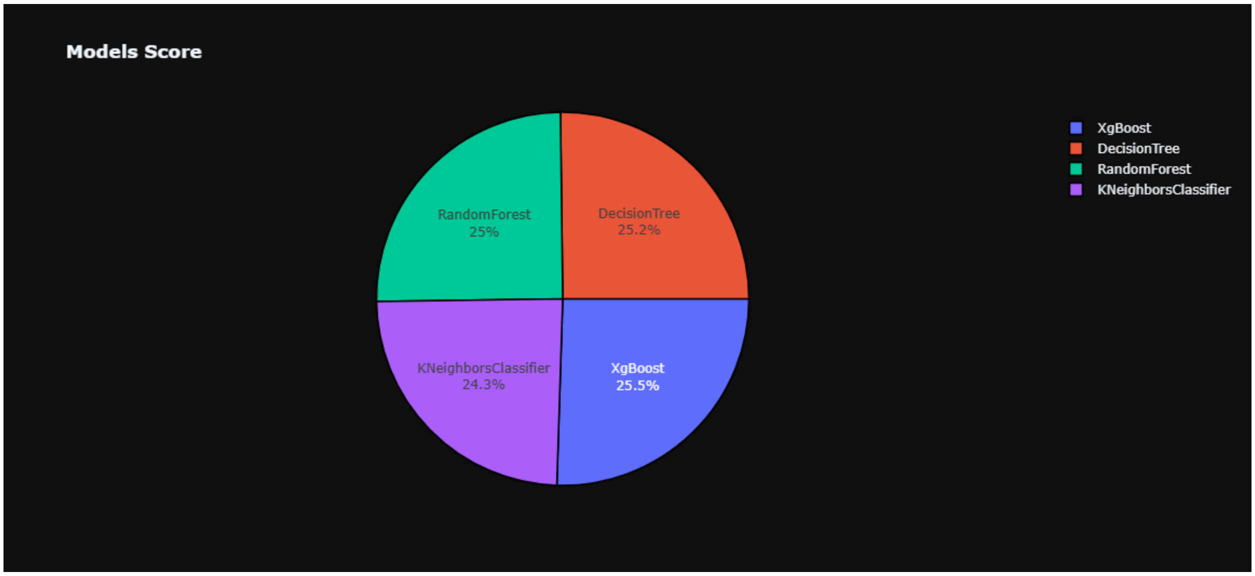


Fig. 9. Model Accuracy Distribution

Pie Chart Description:

- Labels: Model names (KNN, XGBoost, Decision Tree, Random Forest).
- Values: Corresponding accuracy scores.
- Color coding: Differentiates the models visually for easy interpretation.

8. Training Model with Best Hyperparameters and Evaluation

In this project, we utilized the XGBoost Classifier to train a predictive model for placement prediction. The model was trained using the log loss evaluation metric, which is suitable for binary classification problems. To enhance model performance, we fine-tuned various hyperparameters, including a learning rate of 0.25, maximum tree depth of 5, minimum child weight of 5, gamma value of 0.3, colsample_bytree of 0.5, and 100 estimators. These hyperparameters were carefully selected to balance bias and variance, ensuring optimal generalization.

After training, the model was tested on the unseen test dataset, and predictions were generated. The evaluation results showed an accuracy of 89.49%, indicating that the model is highly effective in predicting student placements based on the given features. This level of accuracy suggests that the model can be reliably used to assist in placement decision-making processes, providing valuable insights into student employability based on their academic and skill-based attributes.

Confusion Matrix: The confusion matrix displays the performance of a classification model, showing that the model achieved an accuracy of 0.89. Precision, which measures the proportion of true positives among all positive predictions, is high at 0.95, indicating that when the model predicts a placement, it is usually correct. Recall, which measures the proportion of true positives among all actual positives, is 0.85, suggesting that the model correctly identifies a good portion of actual placements.

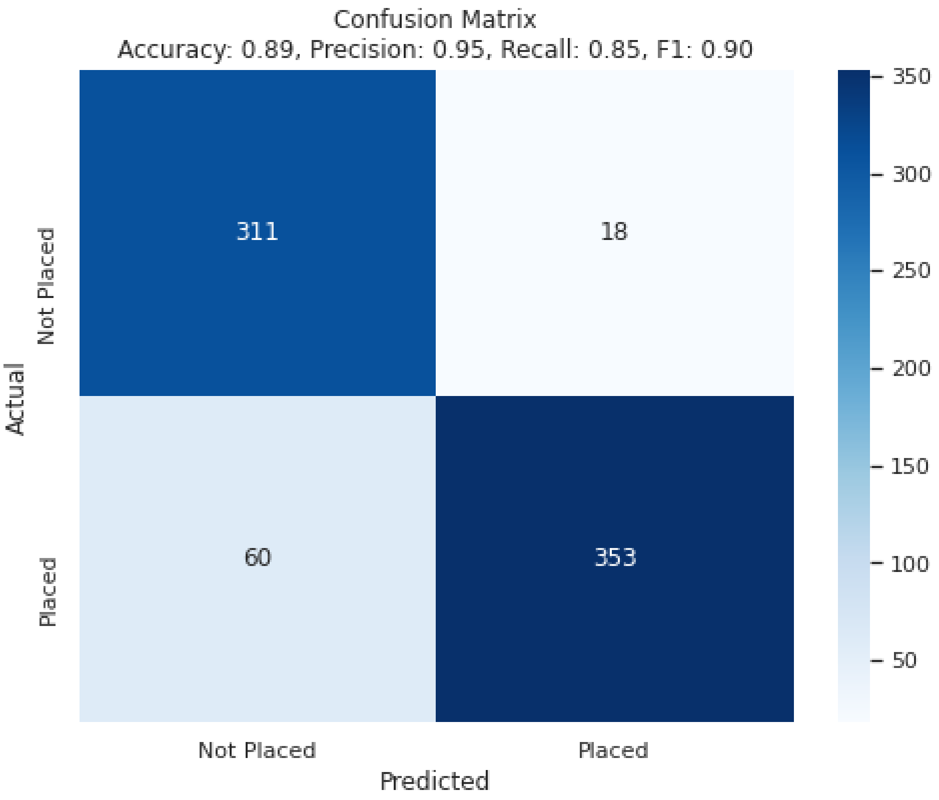


Fig. 10. Confusion Matrix

9. Conclusion

The analysis revealed that CGPA, internship experience, and stream are key factors influencing placement outcomes. Students with a CGPA above 7.07 and at least one internship had significantly higher placement rates. Technical streams like Computer Science and IT showed better placement results, while Mechanical and Civil Engineering had lower rates, reflecting industry demand. Decision Tree and Random Forest were the best-performing models with 87.73% accuracy, followed by XGBoost (87.60%) and KNN (85.18%). Key recommendations include encouraging students to maintain a strong academic record, gain practical experience through internships, and develop technical skills for better employability. Institutions should align curricula with industry needs and offer mentorship programs, while recruiters should broaden hiring criteria beyond CGPA. Future work should focus on incorporating additional features like soft skills, exploring advanced models, and deploying real-time prediction systems for better placement insights.