Assignment 2: Campus Placement

Submitted by: Amanjot Kaur Sohal

Student ID: (C0916838)

Introduction

Predicting student placement outcomes is a crucial task for educational institutions, as it helps in understanding the factors that contribute to employability and career success. With the increasing availability of student data, machine learning techniques offer powerful tools to identify patterns and make accurate predictions based on academic and personal attributes.

In this project, the goal is to build predictive models that can classify students based on their placement status (placed or not placed) using various features such as academic scores, specialization, work experience, and other relevant factors.

Data Description

The dataset consists of 215 entries, each representing a student's academic and professional background, including their placement status. There are 15 columns, which can be categorized into demographic information, academic performance, work experience, and placement-related data.

The columns include:

1. sl_no: A unique identifier for each entry.
2. gender: The gender of the student (encoded as an integer).
3. ssc_p: Secondary Education percentage (10th Grade).
4. ssc_b: Secondary Education board type (e.g., Central/State).
5. hsc_p: Higher Secondary Education percentage (12th Grade).
6. hsc_b: Higher Secondary Education board type.
7. hsc_s: Higher Secondary Education stream (e.g., Science, Commerce).
8. degree_p: Degree percentage (Undergraduate).
9. degree_t: Degree type (e.g., Science, Commerce).
10. workex: Whether the student has work experience.
11. etest_p: Employability test percentage.
12. specialisation: MBA specialisation (e.g., Marketing, Finance).
13. mba_p: MBA percentage.
14. status: Placement status (Placed or Not Placed).
15. salary: Salary offered to placed students (in non-null entries).

Out of the 15 columns, the data types vary:

- 6 columns are of type float64, representing percentages and salary.
- 2 columns are of type int64, representing identifiers and encoded demographic information.
- 7 columns are categorical, stored as object.

The dataset has missing values in the salary column, with 67 missing values out of 215 entries, as the salary is applicable only for students who were placed. The memory usage is approximately 25.3 KB.

Data Cleaning

To ensure the dataset is clean and ready for analysis, the following steps were taken:

Missing Values:

1. The only column with missing values was salary, with 67 missing entries. These missing values are due to students who were not placed, and therefore, have no salary associated.
2. The salary for students who were "Not Placed" was set to 0.
3. For students who were "Placed" but had a missing salary, the missing values were imputed with the mean salary of the placed students.

Encoding Categorical Features:

Categorical variables such as gender, ssc_b, hsc_b, hsc_s, degree_t, workex, specialisation, and status were encoded using LabelEncoder to convert them into numerical values, enabling easier processing for machine learning algorithms.

Duplicate Check:

A check was performed to ensure there were no duplicate rows in the dataset. The data was verified to be free of duplicates.

The dataset is now fully cleaned, with all missing values handled appropriately, and categorical variables encoded into numerical form. It is now ready for further analysis or modeling.

Outlier Detection and Removal

Outliers in the dataset can distort analysis and modeling, particularly in numeric features such as salary. I used the Interquartile Range (IQR) method to detect and remove outliers in the salary column.

1. Interquartile Range (IQR) is calculated as the difference between the third quartile (Q3) and the first quartile (Q1).
2. Any value below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR is considered an outlier.

This process ensures that extreme salary values that could unduly influence the model are excluded, allowing for a more robust analysis.

Handling Class Imbalance with SMOTE

The target variable status (whether a student is placed or not) may have an imbalance in classes, which can negatively affect model performance. For instance, if the number of "Placed" students is significantly larger than "Not Placed" students, the model might become biased toward the majority class. To mitigate this issue, we used SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic examples for the minority class to balance the dataset.

This ensures that our model is exposed to an equal number of examples from both classes, preventing bias toward the majority class. SMOTE oversamples the minority class by creating synthetic data points based on existing samples, thus balancing the number of "Placed" and "Not Placed" students in the dataset.

Feature Extraction

To enhance the dataset and provide more meaningful inputs for modeling, two new features were engineered:

Average Academic Score:

We created a composite academic performance indicator by averaging the scores from Secondary School Certificate (ssc_p), Higher Secondary Certificate (hsc_p), and Degree (degree_p).

This feature provides a more holistic view of a student's academic capabilities across multiple levels of education.

Work Experience Score:

To account for the impact of work experience on employability, we introduced the work_experience_score. This feature adjusts the Employment Test Percentage (etest_p) by a factor of 1.2 for students who have prior work experience, making it more reflective of their practical knowledge.

These derived features help encapsulate both academic performance and work experience in a simplified manner, improving the model's ability to interpret these critical factors.

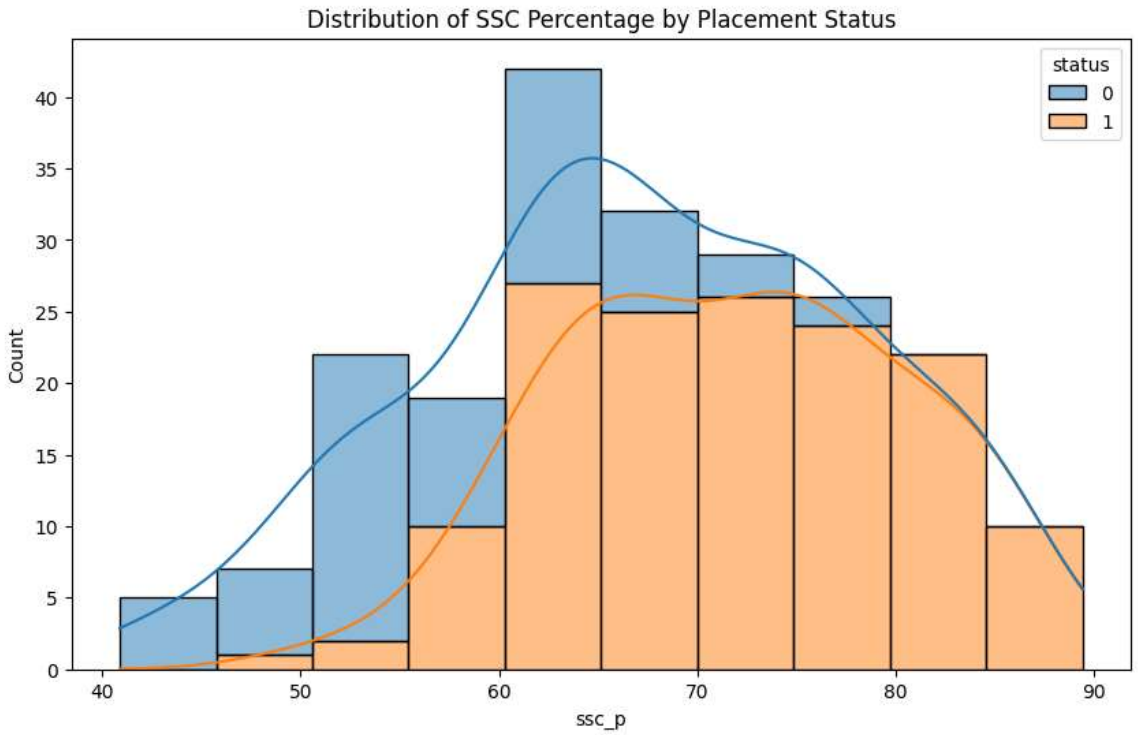
Data Visualization

SSC Percentage vs Placement Status

This graph represents the distribution of Secondary School Certificate (SSC) percentage scores, broken down by placement status. Two categories are shown:

Status 0 (blue) represents individuals who were not placed.

Status 1 (orange) represents those who were placed.



Key observations:

A higher percentage of students with SSC scores between 60-80% were placed (orange bars) compared to those not placed (blue bars). Moreover, students with SSC scores below 60% were more likely not to be placed. Also, there is some overlap around the 60-70% range where both placed and non-placed students are relatively close in numbers.

The density curves show that the distribution of non-placed students (blue curve) peaks around 60%, while the placed students (orange curve) peak around 70%. In summary, students with higher SSC percentages are generally more likely to be placed, with placement rates increasing as SSC percentage increases.

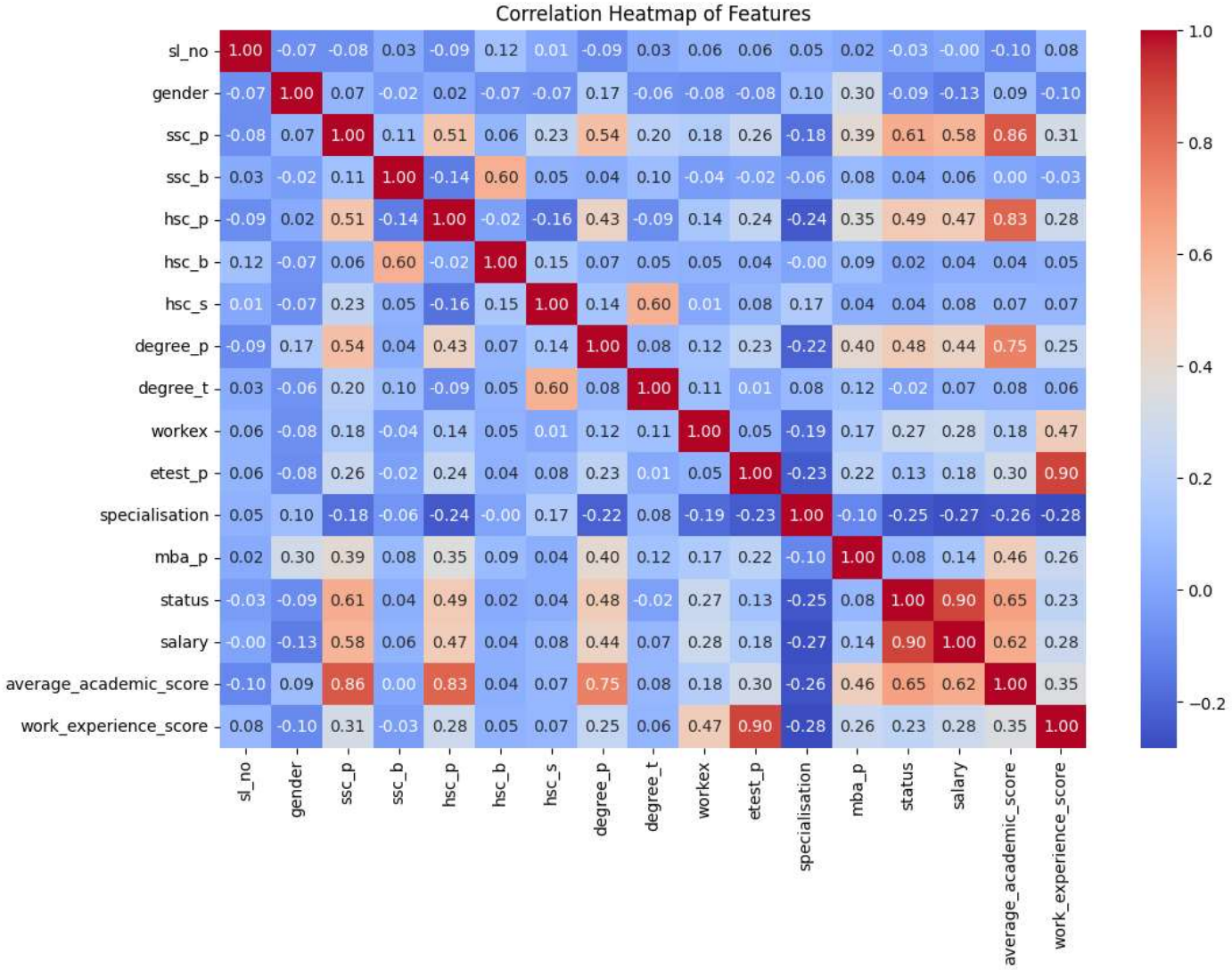
Correlation Heatmap

This image is a correlation heatmap that visualizes the relationships between various features in a dataset. Each cell represents the correlation coefficient between two variables, ranging from -1 to 1:

1 indicates a perfect positive correlation.

-1 indicates a perfect negative correlation.

0 indicates no correlation.



Key Observations:

Academic performance (represented by various scores like SSC, HSC, degree percentage) is a strong predictor of both placement and salary. Moreover, Work experience has a positive, though moderate, impact on placement and salary. There are some inter-correlations between educational metrics (e.g., SSC and HSC percentages are moderately correlated).

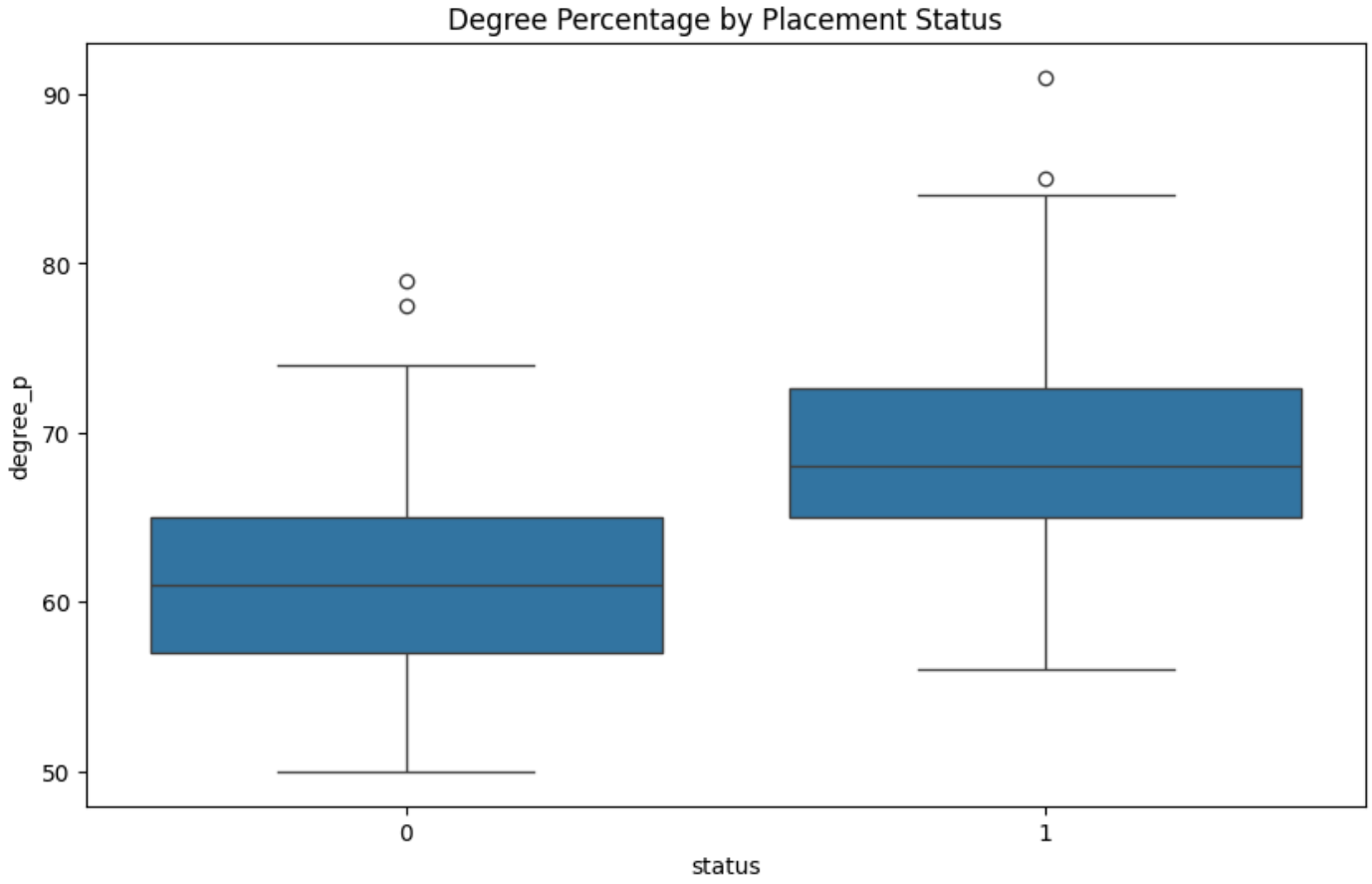
This heatmap is useful for understanding which factors are most strongly related to student success in terms of placement and salary outcomes.

Degree Percentages vs Placement Status

This boxplot illustrates the distribution of degree percentages (degree_p) categorized by placement status (status), where:

0 represents students who were not placed.

1 represents students who were placed.

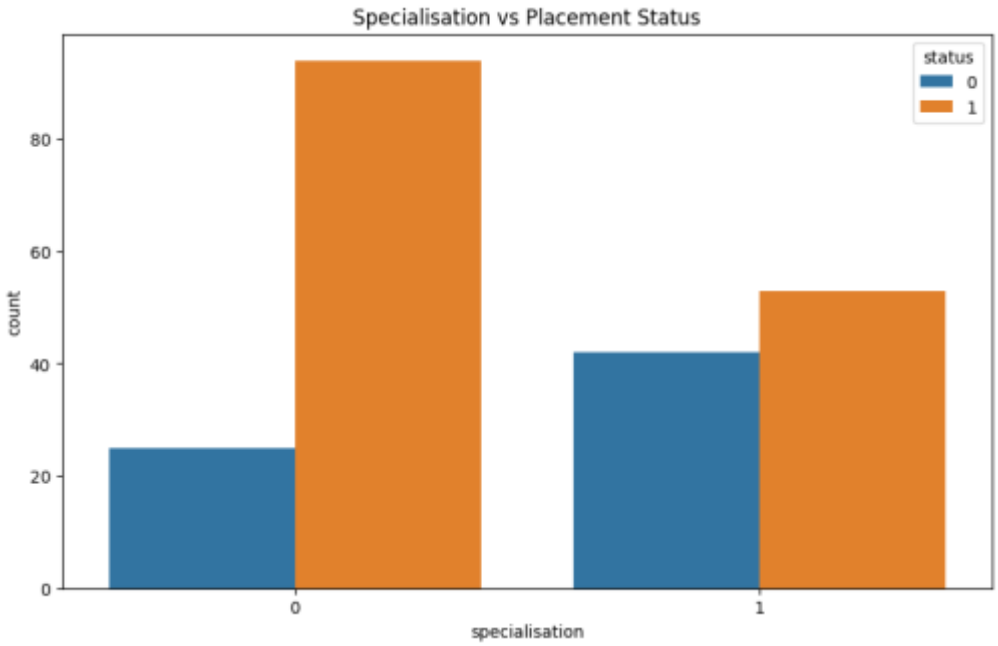


Key Observations:

Higher degree percentages are associated with a greater likelihood of placement. Students who achieved a higher percentage in their degree are more likely to be placed, and the data for placed students is more concentrated around higher percentages compared to the wider spread of non-placed students.

Specialisation vs Placement Status

This bar chart displays the relationship between "Specialisation" and "Placement Status." Here's how to interpret the chart:



Key Observation

Specialisation 0 has a much higher placement rate compared to specialisation 1, whereas specialisation 1 has a more balanced distribution between placed and not placed students.

Feature Selection

Using Correlation Analysis

In the initial phase of feature selection, a correlation heatmap was used to identify highly correlated features. The rationale behind this is to reduce multicollinearity, which can negatively impact model performance. Features that are highly correlated with each other carry redundant information, so it is advisable to remove them. Based on the correlation heatmap, the following features were dropped:

ssc_p (Secondary School Percentage)

hsc_p (Higher Secondary School Percentage)

degree_p (Degree Percentage)

etest_p (Employability Test Percentage)

By dropping these highly correlated features, the dimensionality of the dataset is reduced, and the model can potentially improve its performance by focusing on the most important, independent features.

Using Random Forest Classifier

A Random Forest Classifier was employed, with the target variable being the placement status (status), where 1 represents a student who was placed, and 0 represents a student who was not placed. The remaining features were used as predictors to train the model.

The dataset was split into training and testing sets with the following settings:

- Test size: 30% of the data was allocated to the testing set, while 70% was used for training.
- Random state: A value of 42 was used to ensure reproducibility.

After fitting the Random Forest model, the importance of each feature was assessed to identify which features contribute the most to predicting the placement status. This step helps in understanding the most influential factors that impact student placements. The Top 10 Feature Importances were visualized using a horizontal bar chart. Some features, including salary, average_academic_score, work_experience_score, etc. contribute significantly more than others to the model's predictions.

Based on the feature importance scores from the Random Forest model, further feature selection was conducted. Features with lower importance were dropped to simplify the model and potentially improve its performance by focusing on the most relevant features. The following features were dropped based on their lower contributions:

- ssc_b (Secondary School Board)
- hsc_s (Higher Secondary School Specialization)
- degree_t (Type of Degree)

Model Building

Data Splitting

After selecting the most important features, the dataset was split into training and testing sets to build and evaluate the machine learning model. The target variable for this classification problem is the placement status (`status`), where:

1. 1: Represents students who were placed.
2. 0: Represents students who were not placed.

The features were separated from the target variable to create a model that predicts whether a student will be placed based on the input features. The dataset was split into two sets:

- Training Set: 70% of the data was allocated for training the model. This allows the model to learn the relationships and patterns in the data.
- Testing Set: 30% of the data was reserved for testing. This set is used to evaluate the model's performance on unseen data and ensure it generalizes well.

Stratified Sampling was applied during the split, meaning the class distribution (placement status) in both the training and testing sets was kept similar to the original dataset. This is particularly important when dealing with imbalanced datasets, as it ensures that the model is exposed to a balanced distribution of the target classes during training and testing.

This ensures that the model has a sufficient amount of data to learn from while also providing enough testing data to evaluate its performance effectively.

Preprocessing the Data

Before fitting the machine learning models, the numerical features in the dataset were standardized. Standardization helps to ensure that each feature contributes equally to the model by transforming them to have a mean of 0 and a standard deviation of 1. This is particularly important for models like Support Vector Machines (SVM) and Logistic Regression, which are sensitive to the scale of the input data. The following numerical features were identified and standardized:

- mba_p: MBA percentage.
- salary: Expected salary.
- average_academic_score: Average score across academic evaluations.
- work_experience_score: Score reflecting prior work experience.

A ColumnTransformer was applied to standardize these features while leaving the rest of the dataset unchanged. This ensured that only the numerical features were scaled, and other categorical or ordinal features remained intact.

Model Training

Several classification models were employed to predict whether a student would be placed based on their features. The models were evaluated to determine which algorithm performs best for this classification task.

a) Random Forest Classifier

Algorithm: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification.

Parameters:

- n_estimators=100: The model constructs 100 trees.
- random_state=42: Ensures reproducibility.

b) Support Vector Classifier (SVM)

Algorithm: SVM is a powerful classification algorithm that finds a hyperplane that best separates the classes.

Parameters:

- probability=True: Allows for probability estimates from the model.
- random_state=42: Ensures reproducibility.

c) Logistic Regression

Algorithm: Logistic Regression is a simple yet effective classification algorithm that models the probability of a binary outcome based on input features.

Parameters:

- random_state=42: Ensures reproducibility.

d) Ensemble Model with Voting Classifier

To improve prediction accuracy, a Voting Classifier was used. Voting classifiers combine the predictions from multiple models to make a final prediction. This method is beneficial as it takes advantage of the strengths of different models, reducing the chances of error due to the weaknesses of any single model.

Soft Voting was applied, which uses the predicted probabilities from each classifier to make the final decision. This approach allows for more nuanced decisions compared to hard voting, which simply takes the majority class.

Model Evaluation

After training several machine learning models (Random Forest, Support Vector Machine (SVM), Logistic Regression, and Voting Classifier), their performance was evaluated using various metrics, including accuracy, precision, recall, and F1 score. The results of these metrics, along with the confusion matrix and classification report for each model, are presented below.

Evaluation Metrics:

1. Accuracy: Proportion of correctly predicted instances out of the total.
2. Precision: The number of true positive results divided by the total number of positive predictions (precision = TP / (TP + FP)).
3. Recall: The number of true positive results divided by the total number of relevant instances (recall = TP / (TP + FN)).
4. F1 Score: The harmonic mean of precision and recall, balancing both metrics.

Each model's performance was evaluated on the test set (`y_test` and `y_pred`).

a) Random Forest Classifier

- Accuracy: 1.0
- Precision: 1.0
- Recall: 1.0
- F1 Score: 1.0

The Random Forest model achieved perfect accuracy, correctly classifying all instances in the test set. Both precision and recall are 1.0, indicating that the model perfectly classified all students who were placed (`status=1`) and not placed (`status=0`).

b) Support Vector Machine (SVM)

- Accuracy: 0.69
- Precision: 0.69
- Recall: 1.0
- F1 Score: 0.82

The SVM model had an accuracy of 69.2%, indicating that it failed to classify the negative class (`status=0`). Precision for class `1` (students placed) is 0.69, while recall for class `1` is 1.0, which means that while the model captured all students placed, it mistakenly predicted all the `status=0` students as `status=1`.

c) Logistic Regression

- Accuracy: 1.0
- Precision: 1.0
- Recall: 1.0
- F1 Score: 1.0

Logistic Regression also achieved perfect accuracy, similar to the Random Forest model. The model correctly classified all students for both `status=0` and `status=1`, achieving perfect precision and recall.

d) Voting Classifier (Soft Voting)

- Accuracy: 1.0
- Precision: 1.0
- Recall: 1.0
- F1 Score: 1.0

The Voting Classifier, which combines the predictions from Random Forest, SVM, and Logistic Regression, also achieved perfect accuracy. Similar to Random Forest and Logistic Regression, this ensemble method perfectly classified all test instances, producing flawless precision, recall, and F1 scores.

Model Comparison

Random Forest, Logistic Regression, and Voting Classifier all achieved perfect accuracy, indicating that these models are well-suited for the dataset and classification task.

SVM, while performing well for class `1` (students placed), struggled to classify students in class `0` (students not placed). This resulted in lower accuracy and precision for the model overall.

The Voting Classifier (soft voting) demonstrated the benefits of ensemble learning by achieving the same level of performance as Random Forest and Logistic Regression, but potentially offering more robustness by combining multiple models.

The final recommendation would be to either use the Voting Classifier or Random Forest for future predictions, as both demonstrated superior performance across all evaluation metrics.

Conclusion

In this study, several machine learning models were developed and evaluated to predict the placement status of students based on various academic and personal features. After preprocessing the data, models like Random Forest, Support Vector Machine (SVM), Logistic Regression, and an ensemble Voting Classifier were trained and tested.

The key findings are as follows:

1. Random Forest, Logistic Regression, and the Voting Classifier achieved perfect performance with an accuracy, precision, recall, and F1 score of 1.0. These models correctly classified all students in the test dataset, making them highly reliable for this classification task.
2. The SVM model underperformed compared to the other models, with an accuracy of 69.2%, largely due to its inability to correctly classify students who were not placed (`status=0`).

Although the Voting Classifier offers the advantage of combining the predictions of three strong models (Random Forest, Logistic Regression, and SVM) and is recommended for its robustness and reliability in classification tasks, Random Forest and Logistic Regression could be used as an alternative to the Voting Classifier if computational efficiency is a concern.

In conclusion, the ensemble approach with the Voting Classifier is recommended for future predictions, ensuring a balanced and accurate classification of student placement status. These models will help educational institutions make informed decisions to improve student placement strategies.