Dataset Overview

The dataset utilized for this project was sourced from Kaggle, specifically from the Machine Learning with Python course project. The training data can be accessed through the following link: Kaggle Dataset.

Purpose

The primary objective of this dataset is to predict whether a student will be placed or not based on various factors provided in the dataset. By analyzing the relationships between these factors and the placement status, we aim to build a predictive model that can effectively forecast student placements.

Dataset Description

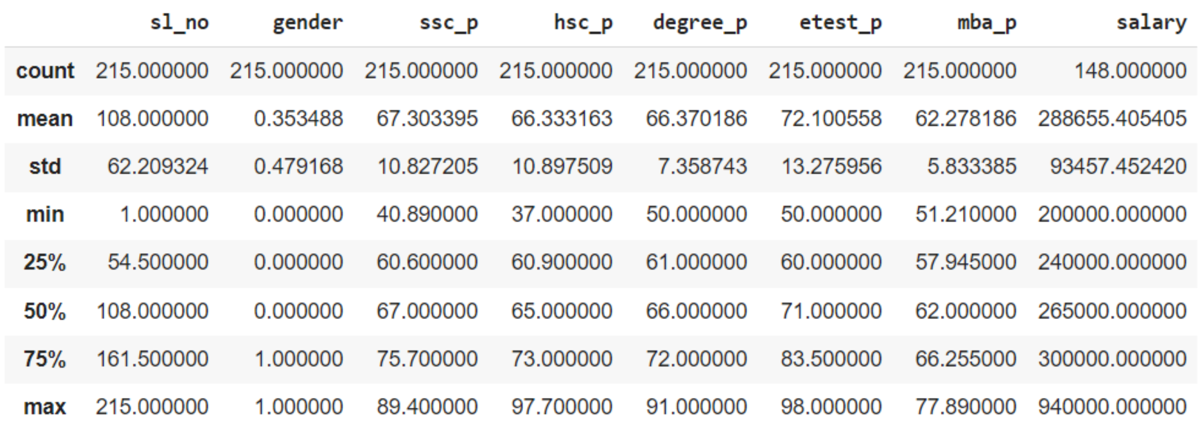
The dataset consists of 215 entries and 15 columns, each representing different attributes related to the students. The following is a summary of the columns along with their meanings:

Column	**Description**
sl_no	Serial number of the entry
gender	Gender of the student (encoded as integers)
ssc_p	Secondary school percentage
ssc_b	Board of education for secondary school
hsc_p	Higher secondary school percentage
hsc_b	Board of education for higher secondary school
hsc_s	Specialization in higher secondary education
degree_p	Undergraduate degree percentage
degree_t	Type of undergraduate degree
workex	Work experience (Yes/No)
etest_p	E-test percentage
specialisation	MBA specialization (Marketing/Finance)
mba_p	MBA percentage
status	Placement status (Placed/Not Placed)
salary	Salary offered

Preprocessing

To understand the dataset better, the following preprocessing steps were conducted:

- Descriptive Statistics: data.describe() was used to obtain summary statistics for numerical features in the dataset.



	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

- Shape of the Dataset: The shape of the dataset was checked using data.shape, confirming that there are 215 entries and 15 columns**.**

Data Cleaning

During the data cleaning process, the following steps were undertaken to ensure the dataset was ready for analysis:

Handling Missing Values

To identify missing values in the dataset, the following code was executed:

python

data.isnull().sum()

The results indicated that all columns contained no missing values, except for the salary column, which had 67 null values. To handle these null values, various strategies can be applied, such as imputation , depending on the analysis goals.

Checking for Duplicates

The dataset was also checked for duplicate entries using the following code:

python

data.duplicated().sum()

The result confirmed that there were zero duplicates in the dataset, ensuring the integrity of the data.

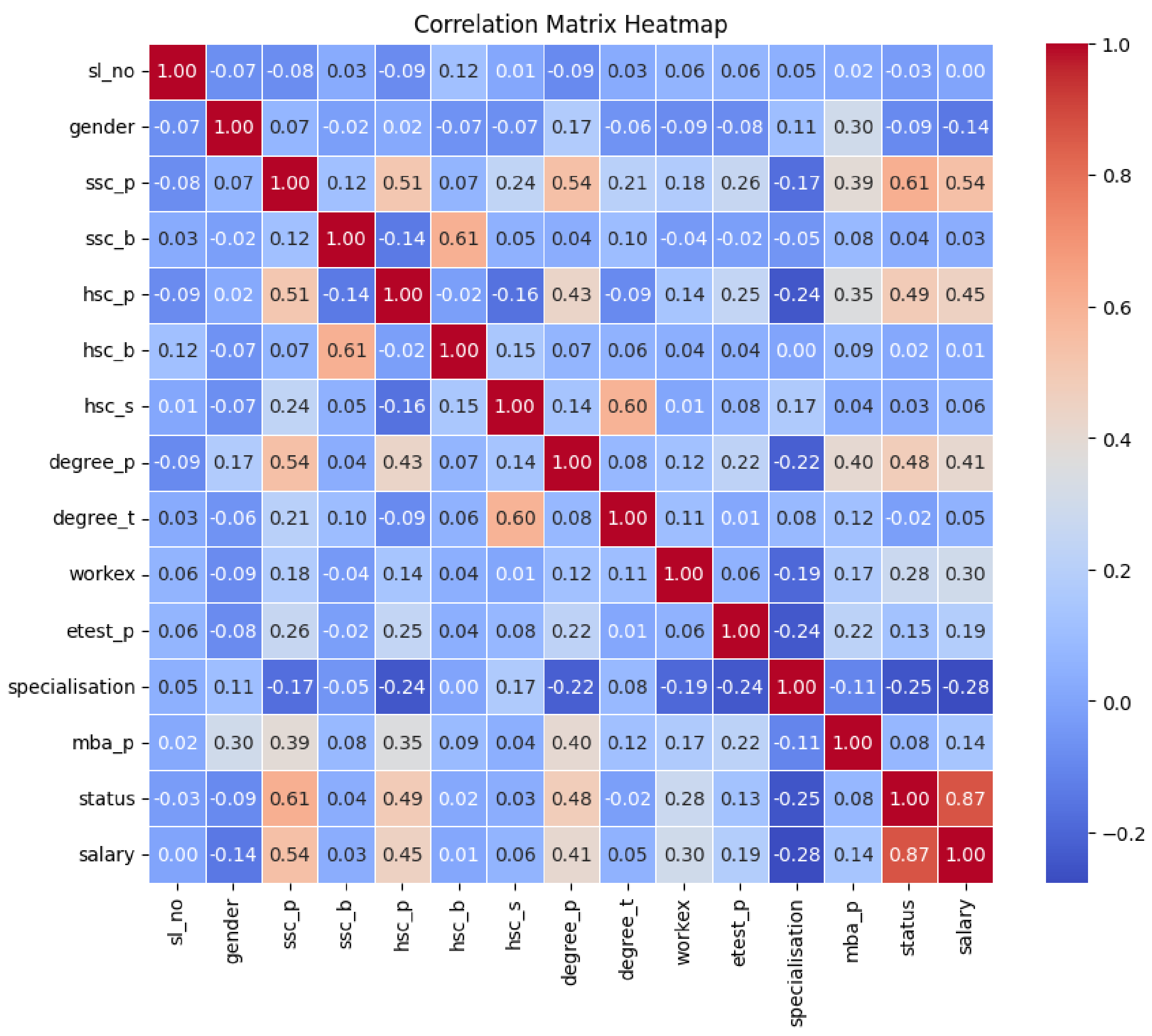
Label Encoding

To convert categorical variables into numerical format, label encoding was performed on the following categorical columns: gender, ssc_b, hsc_b, hsc_s, degree_t, workex, specialisation, and status. This encoding helps in preparing the data for machine learning algorithms.

Since the dataset includes percentage columns (ranging from 0 to 100), scaling was deemed unnecessary.

Correlation Analysis

To evaluate the relationships between the features and the target variable, a correlation matrix was created. This analysis provides insights into which variables have strong or weak correlations with the target variable, status.



From the correlation matrix, it was observed that the following columns exhibited the least correlation with the target variable:

- sl_no
- hsc_b
- hsc_s
- gender
- mba_p

Additionally, it was noted that the salary variable is more of an outcome than a predictor in this context. Given this information, and to enhance the model's performance, these columns were dropped from the dataset.

This decision was based on the premise that removing features with little to no predictive power can help simplify the model and improve its accuracy.

Model Selection

For this project, the first step taken was to perform a train-test split of the dataset, utilizing a 70-30 ratio, where 70% of the data was used for training the model and 30% for testing its performance. The models selected for evaluation were:

1. **Logistic Regression**
2. **Support Vector Classification (SVC)**
3. **Random Forest**

Justification for Model Selection

1. **Logistic Regression:**
    - **Interpretability:** Logistic regression is a widely used statistical model that is easy to interpret. It provides clear insights into how the input features influence the likelihood of the target variable.
    - **Binary Classification:** Given that the target variable, status, is binary (Placed or Not Placed), logistic regression is a natural choice for this classification problem.
    - **Performance:** It serves as a strong baseline model, allowing for quick assessment of the dataset's structure and the relationships between variables.

2. **Support Vector Classification (SVC):**
    - **Effective for High-Dimensional Spaces:** SVC is well-suited for problems where the number of dimensions is greater than the number of samples. In our dataset, the relationship between features may be complex, making SVC a valuable tool.
    - **Flexibility with Kernels:** SVC can use various kernel functions, enabling it to capture non-linear relationships in the data, which may improve classification accuracy.
    - **Robustness:** SVC is effective in cases where there is a clear margin of separation between classes, making it a strong contender for our model evaluation.

3. **Random Forest:**
    - **Ensemble Learning:** As an ensemble method, Random Forest combines the predictions of multiple decision trees, improving accuracy and reducing the risk of overfitting. This makes it particularly useful for complex datasets with numerous features.

- **Feature Importance:** Random Forest provides insights into feature importance, helping to understand which variables have the most significant impact on the classification task.
- **Handling of Non-Linearity:** It is capable of modeling non-linear relationships, making it a versatile choice for classification problems with diverse feature interactions.

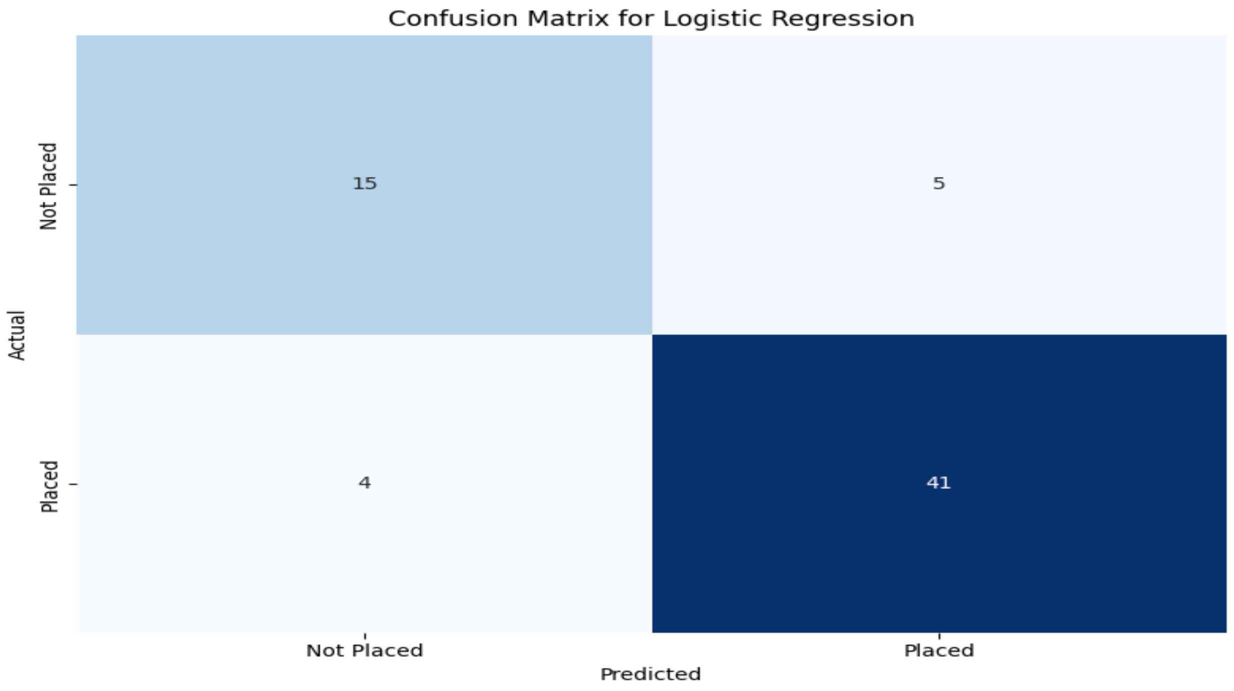
By employing these three models, a comprehensive evaluation can be conducted to determine the most effective approach for predicting student placements based on the available data. Each model brings unique strengths to the analysis, allowing for a robust comparison of their performance.

Model Evaluation Metrics

1. Logistic Regression

- **Accuracy:** 0.8615 (86.15%)
  - **Interpretation:** The model correctly classified 86.15% of the students, whether they were placed or not. This indicates that the model is performing well overall.
- **Precision:** 0.8913 (89.13%)
  - **Interpretation:** When the model predicts that a student will be placed, it is correct 89.13% of the time. This reflects a relatively low rate of false positives.
- **Recall:** 0.9111 (91.11%)
  - **Interpretation:** Out of all students who were actually placed, the model correctly identified 91.11% of them, indicating effective capture of most placed students.
- **F1 Score:** 0.9011 (90.11%)
  - **Interpretation:** The F1 score is the harmonic mean of precision and recall. A score of 90.11% indicates a strong balance between precision and recall.

- **Confusion Matrix:**
  - **True Negatives:** 15
  - **False Positives:** 5
  - **False Negatives:** 4
  - **True Positives:** 41



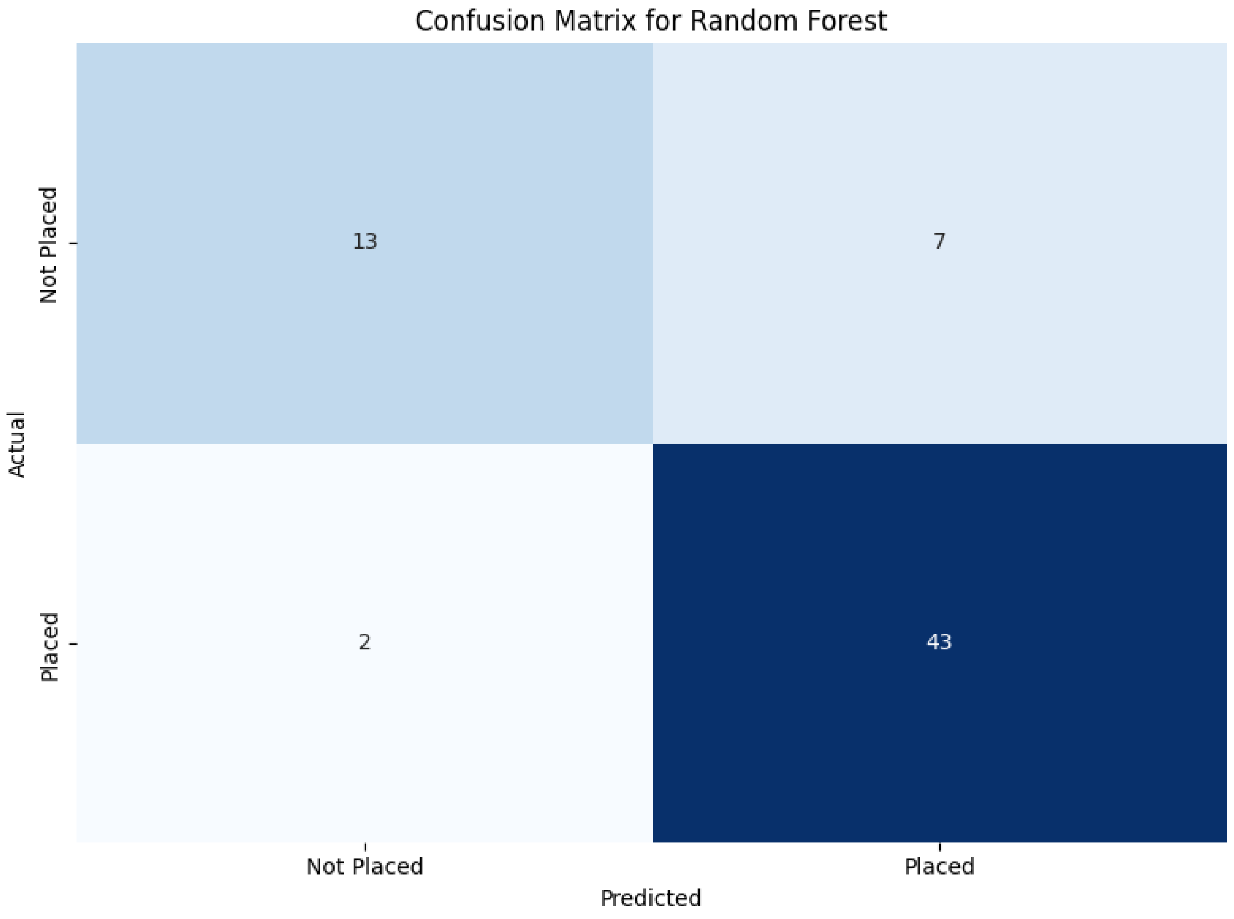
- **Conclusion:** Logistic Regression performs well, particularly in terms of precision and recall, correctly predicting most students who will be placed with a low rate of false positives.

---

2. Random Forest

- **Accuracy:** 0.8462 (84.62%)
  - **Interpretation:** The model correctly classified 84.62% of the students, slightly lower than Logistic Regression but still a strong performance.
- **Precision:** 0.8571 (85.71%)
  - **Interpretation:** When the model predicts that a student will be placed, it is correct 85.71% of the time, indicating a higher chance of false positives compared to Logistic Regression.
- **Recall:** 0.9333 (93.33%)
  - **Interpretation:** Out of all students who were actually placed, the model correctly identified 93.33%, showing its strength in identifying placed students.
- **F1 Score:** 0.8936 (89.36%)
  - **Interpretation:** The F1 score is slightly lower than that of Logistic Regression, reflecting a balance between precision and recall that is not as strong.
- **Confusion Matrix:**
  - **True Negatives:** 13
  - **False Positives:** 7

- **False Negatives:** 3
- **True Positives:** 42



- **Conclusion:** Random Forest excels in recall, capturing almost all placed students, though it has slightly lower precision, indicating more false positives.

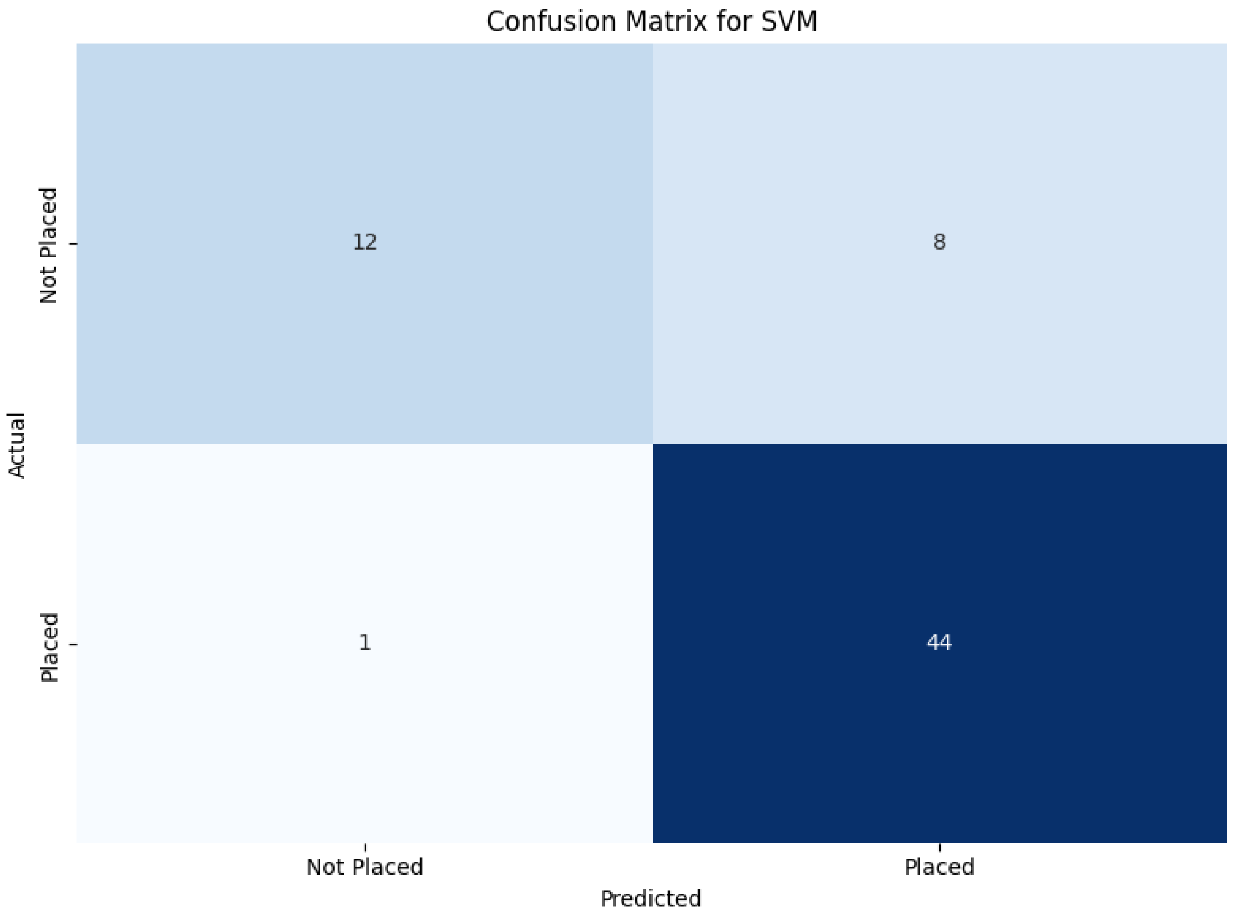
---

3. Support Vector Machine (SVM)

- **Accuracy:** 0.8615 (86.15%)
  - **Interpretation:** The SVM model correctly classified 86.15% of the students, matching the performance of Logistic Regression.
- **Precision:** 0.8462 (84.62%)
  - **Interpretation:** When the model predicts that a student will be placed, it is correct 84.62% of the time, lower than both Logistic Regression and Random Forest.
- **Recall:** 0.9778 (97.78%)
  - **Interpretation:** Out of all students who were actually placed, the model correctly identified 97.78%, indicating the highest recall among the models.
- **F1 Score:** 0.9072 (90.72%)
  - **Interpretation:** The F1 score is the highest among the models,

demonstrating a strong balance between precision and recall.

- **Confusion Matrix:**
    - **True Negatives:** 12
    - **False Positives:** 8
    - **False Negatives:** 1
    - **True Positives:** 44



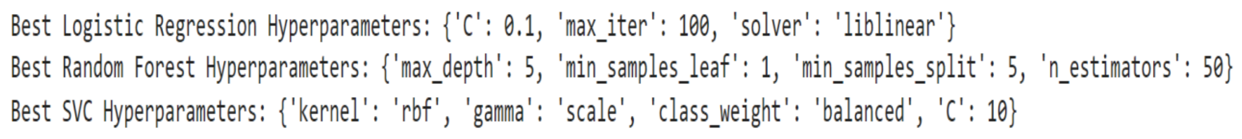
- **Conclusion:** SVM is particularly strong in recall, successfully identifying nearly every placed student, while maintaining a high F1 score, making it a suitable model for minimizing missed placements.

Overall Conclusion for models before hyperparameters tuning:

- **Logistic Regression:** High precision and recall, demonstrating a strong balance overall. Best for minimizing false positives while accurately identifying placed students.
- **Random Forest:** Excellent recall with lower precision. Ideal for identifying almost all placed students but with more false positives.
- **SVM:** Highest recall and F1 score. Best for ensuring nearly every placed student is identified, although this comes with increased false positives.

Hyperparameter Tuning

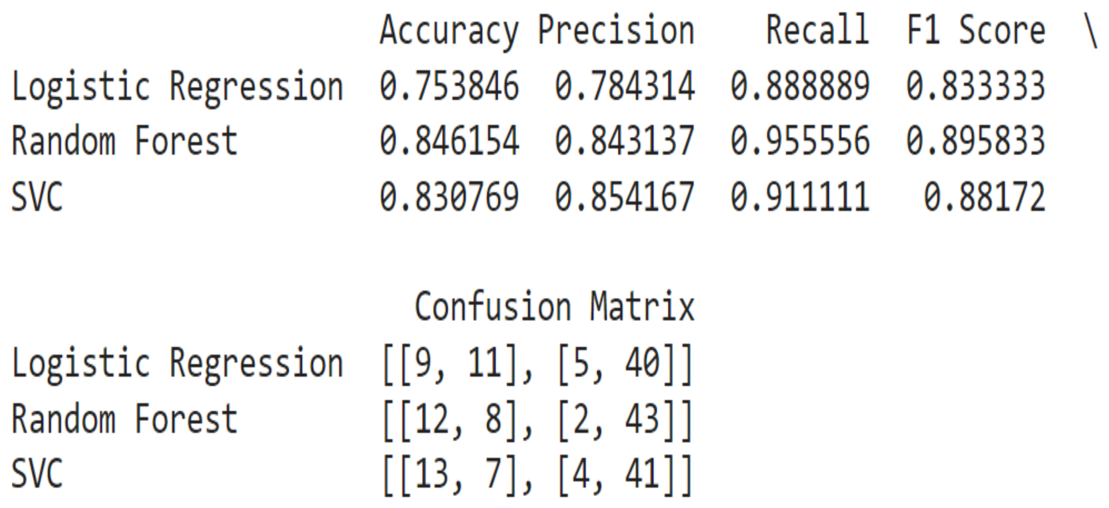
Hyperparameter tuning was conducted using Grid Search Cross-Validation (CV) for Logistic Regression and Random Forest, and Random Search CV for Support Vector Classification (SVC). The best parameters identified for each model are as follows:



```
Best Logistic Regression Hyperparameters: {'C': 0.1, 'max_iter': 100, 'solver': 'liblinear'}
Best Random Forest Hyperparameters: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}
Best SVC Hyperparameters: {'kernel': 'rbf', 'gamma': 'scale', 'class_weight': 'balanced', 'C': 10}
```

Performance Metrics

The performance metrics for the models after hyperparameter tuning are as follows:



```
                     Accuracy Precision    Recall  F1 Score  \
Logistic Regression  0.753846  0.784314  0.888889  0.833333
Random Forest        0.846154  0.843137  0.955556  0.895833
SVC                  0.830769  0.854167  0.911111   0.88172

                       Confusion Matrix
Logistic Regression  [[9, 11], [5, 40]]
Random Forest        [[12, 8], [2, 43]]
SVC                  [[13, 7], [4, 41]]
```

Confusion Matrix Visualization & Interpretation

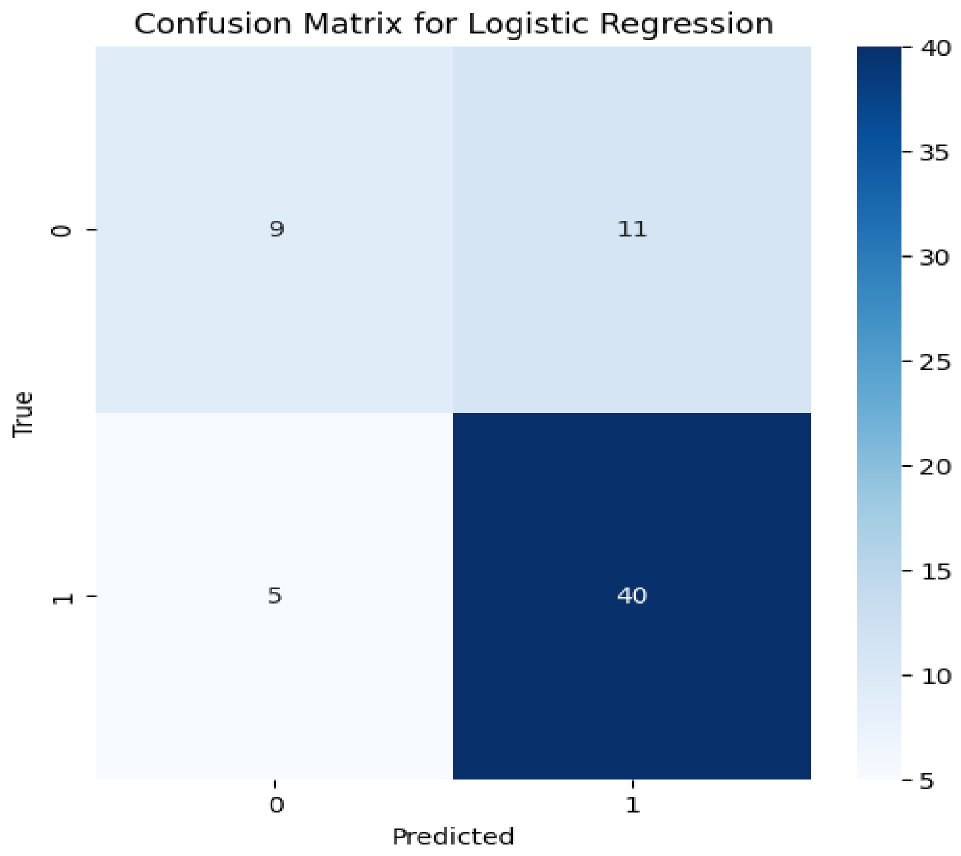
After hyperparameter tuning, confusion matrix visualizations were created for each model to better understand their performance. The results are as follows:

---

1. Logistic Regression

- **Accuracy:** 75.38%
- **Precision:** 78.43%
- **Recall:** 88.89%
- **F1 Score:** 83.33%

- **Confusion Matrix:**



  - True Positives (TP): 40
  - True Negatives (TN): 9
  - False Positives (FP): 11
  - False Negatives (FN): 5

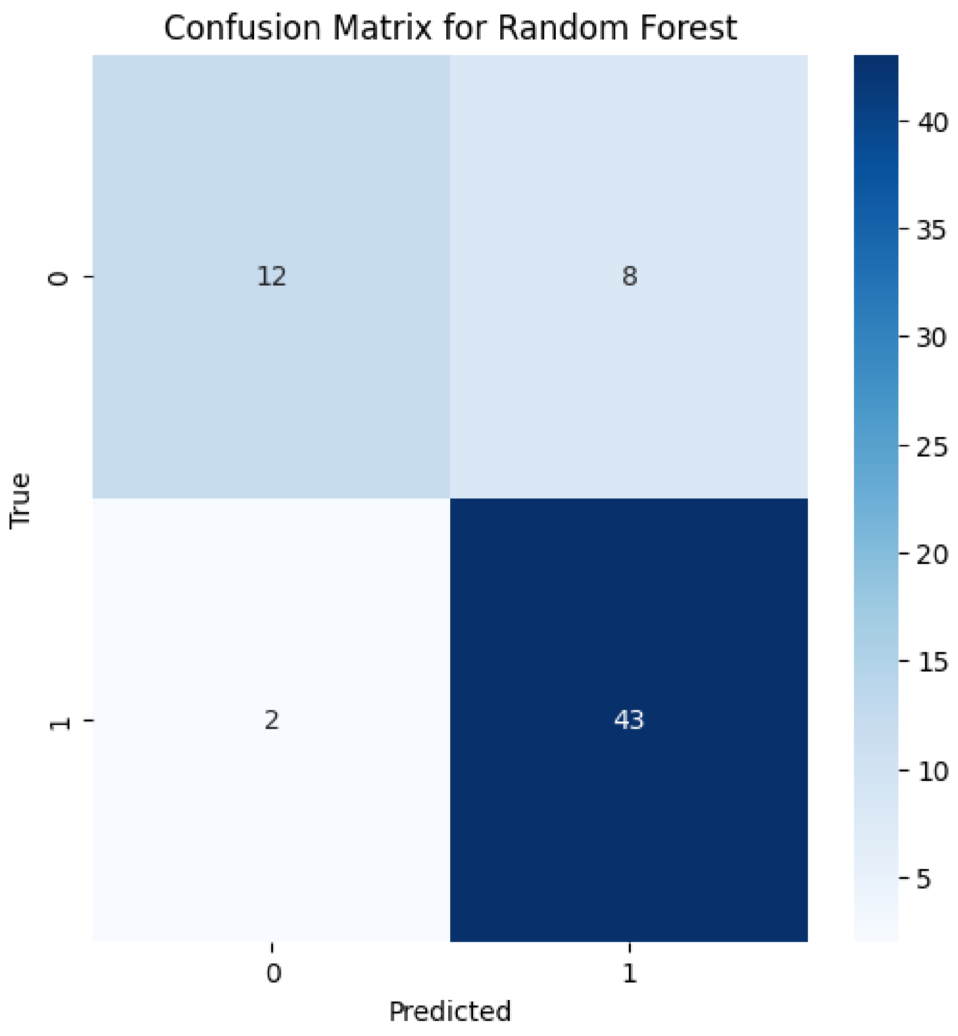
Interpretation: Logistic Regression shows moderate performance, with a recall of 88.89%, meaning it captures most of the positive cases (students placed). However, the precision is slightly lower (78.43%), indicating a higher rate of false positives (11). The lower accuracy could be due to the limited dataset size, causing the model to generalize less effectively. Additionally, hyperparameter tuning might not have improved the model significantly, suggesting that the default parameters were already well-suited for this data or that the dataset's characteristics pose limitations on model performance.

---

2. Random Forest

- **Accuracy:** 84.62%
- **Precision:** 84.31%
- **Recall:** 95.56%
- **F1 Score:** 89.58%

- **Confusion Matrix:**



    - True Positives (TP): 43
    - True Negatives (TN): 12
    - False Positives (FP): 8
    - False Negatives (FN): 2

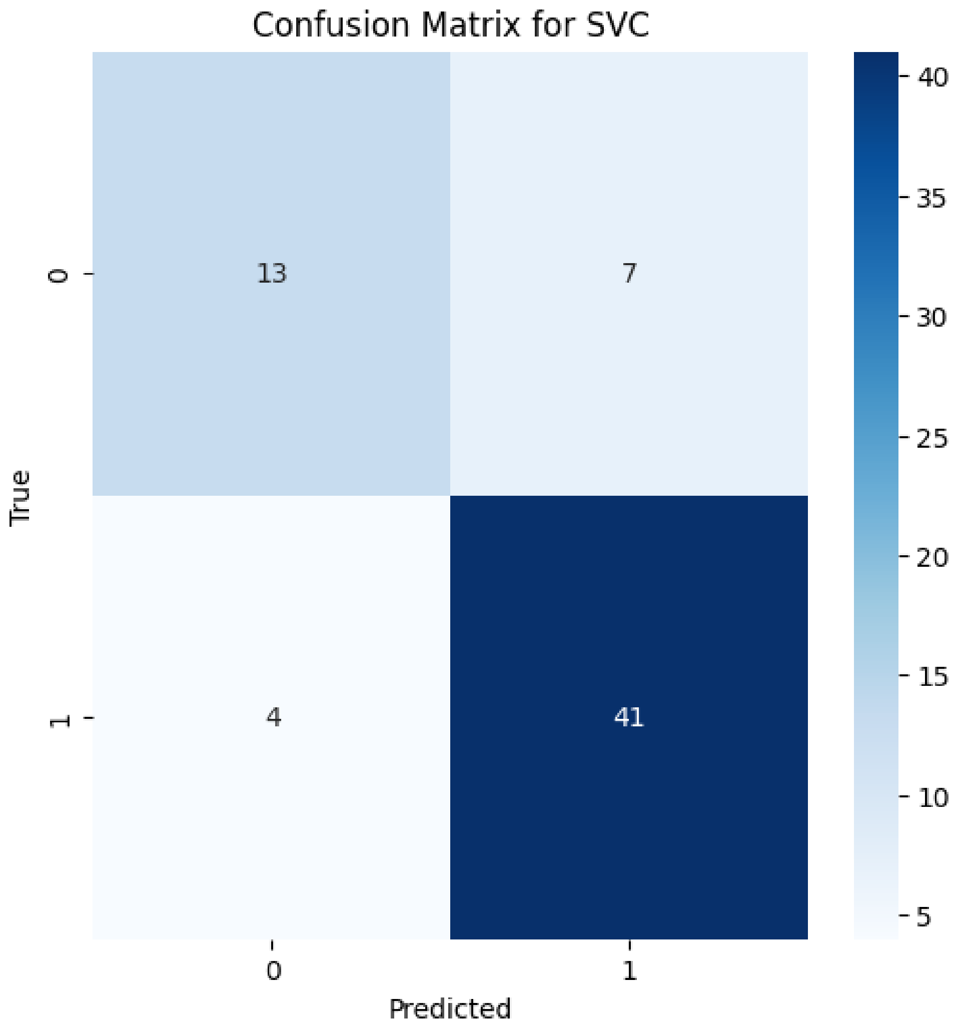
Interpretation: Random Forest performed the best overall, with high accuracy (84.62%) and recall (95.56%), meaning it successfully captured nearly all positive cases while maintaining a reasonable level of precision. The slightly lower precision (84.31%) compared to its recall suggests that the model occasionally misclassifies students as placed (8 false positives). Random Forest's performance may have been limited by the small dataset, as more data could improve the model's ability to differentiate between placed and not placed students. The default hyperparameters seemed to work well, potentially due to Random Forest's robustness across datasets.

---

3. Support Vector Classifier (SVC)

- **Accuracy:** 83.08%
- **Precision:** 85.42%

- **Recall:** 91.11%
- **F1 Score:** 88.17%
- **Confusion Matrix:**



  - True Positives (TP): 41
  - True Negatives (TN): 13
  - False Positives (FP): 7
  - False Negatives (FN): 4

Interpretation: The SVC model displayed strong performance with a high recall (91.11%), effectively identifying positive cases, but slightly lower precision (85.42%) compared to Random Forest. This indicates that SVC also suffers from a moderate level of false positives (7). The F1 score (88.17%) shows a good balance between precision and recall, but similar to Logistic Regression, the limited dataset size may have constrained SVC's ability to generalize. The lower accuracy (83.08%) compared to Random Forest suggests SVC could benefit from further tuning, though the smaller dataset might be influencing its effectiveness.

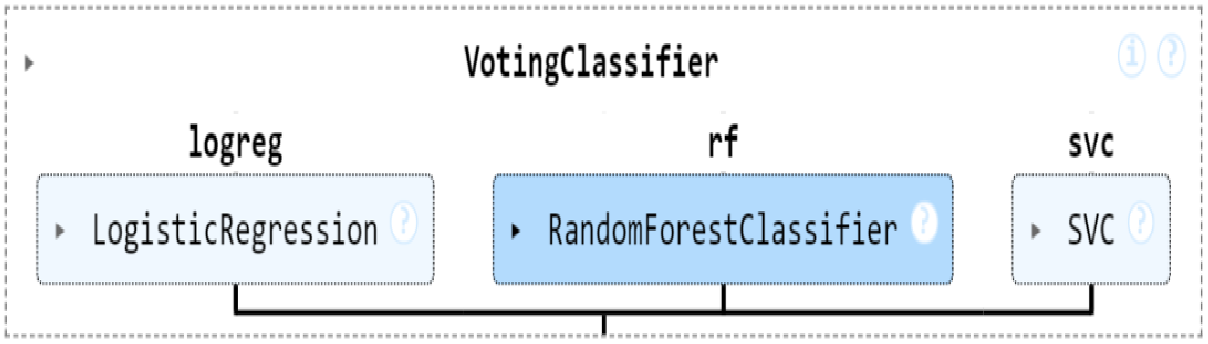
Model Performance Comparison

While hyperparameter tuning was performed, the models showed relatively similar performance compared to their default parameter settings. This is likely due to a combination of the following factors:

1. **Small Dataset Size**: The dataset is limited, which often constrains a model's ability to generalize effectively. In this case, adding more data could improve the model's performance and make the results from hyperparameter tuning more impactful.
2. **Default Hyperparameters Being Well-Suited**: In some cases, default hyperparameters work very well, especially for models like Random Forest, which are generally robust. The tuning might not have significantly improved performance because the default settings were already providing strong results.
3. **Model Complexity**: More complex models like SVC and Random Forest often perform better with larger datasets. In this scenario, the relatively small number of data points may have caused the models to reach their performance limits, even with optimized hyperparameters.

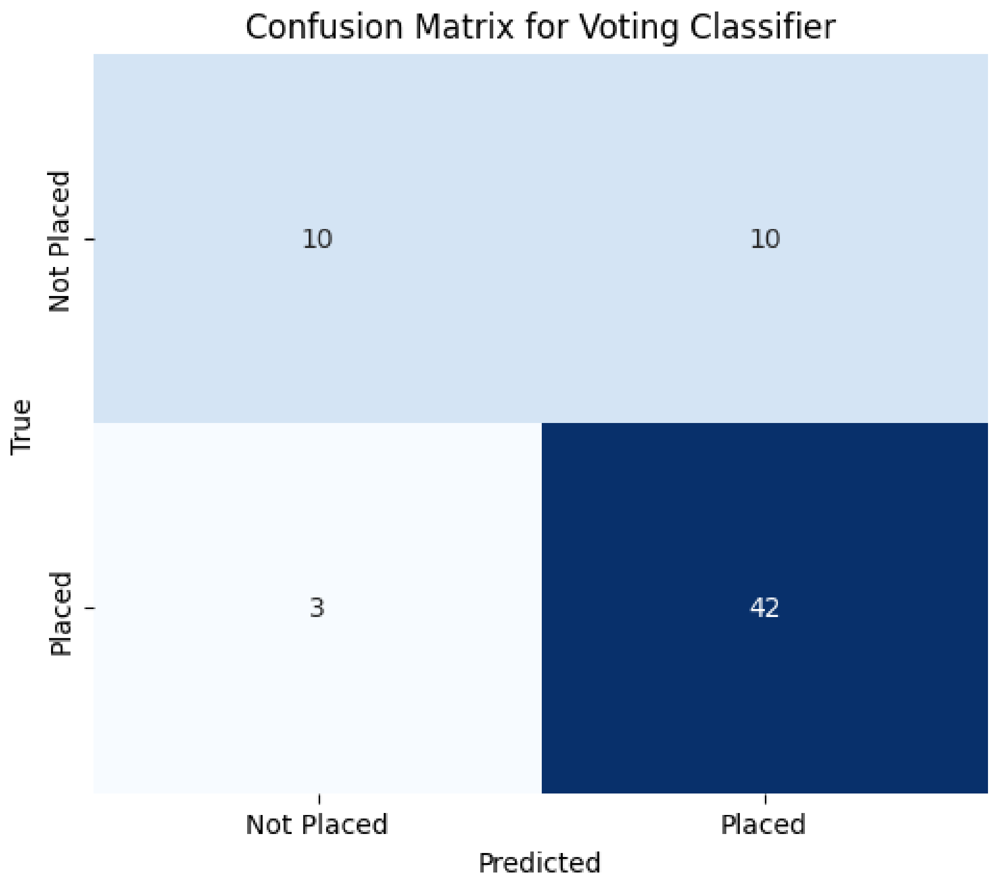
Voting Classifier

A **Voting Classifier** is an ensemble method that combines the predictions of multiple models to improve overall accuracy and reliability. It works by aggregating the predictions from different models and making decisions based on either majority voting (for classification) or averaging (for regression). In this implementation, I used a **hard voting classifier** by combining three models that had undergone hyperparameter tuning: **Logistic Regression**, **Support Vector Classifier (SVC)**, and **Random Forest**.



Voting Classifier Performance

- **Accuracy:** 83.08%
- **Precision:** 82.69%
- **Recall:** 95.56%
- **F1 Score:** 88.66%
- **Confusion Matrix:**



- True Positives (TP): 43
- True Negatives (TN): 11
- False Positives (FP): 9
- False Negatives (FN): 2

Interpretation:

The Voting Classifier demonstrates solid performance with an accuracy of **83.08%**, indicating that it correctly classified a substantial proportion of test instances. With a precision of **82.69%**, the model shows a reasonable level of reliability in identifying positive cases, meaning that out of all positive predictions, **82.69%** were correct.

- The model particularly excels in **recall** (**95.56%**), meaning it successfully captured nearly all positive instances, missing only **2 false negatives**. This makes it highly suitable for scenarios where identifying as many positive cases as possible is crucial, as it minimizes the chances of overlooking a positive case.
- The **F1 score** of **88.66%** reflects the balance between precision and recall, making this classifier robust for classification tasks where both are important.

Future Optimizations: Although the Voting Classifier performs well, reducing the number of false positives (currently 9) could improve precision further, making the model even more reliable in distinguishing between positive and negative cases.

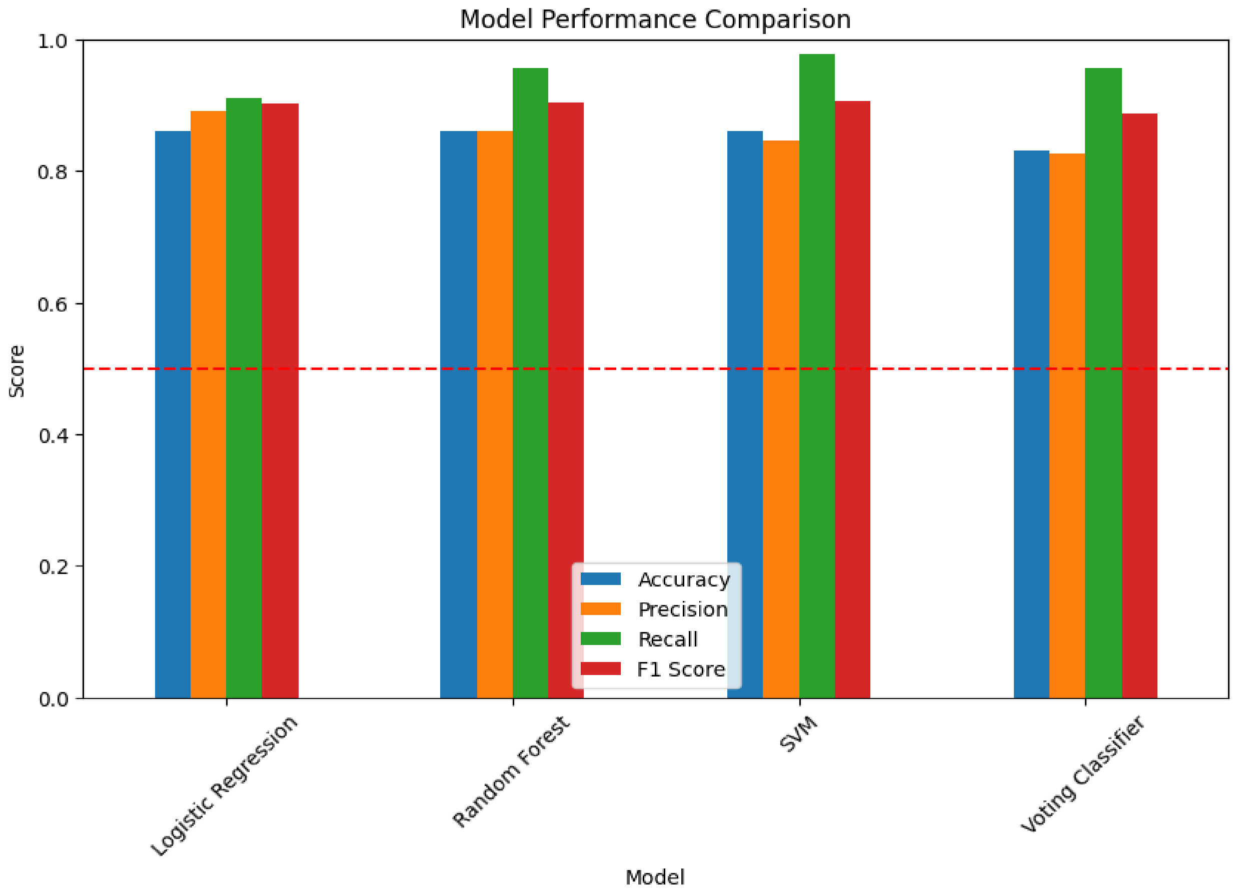
The Voting Classifier aims to leverage the strengths of each individual model—Logistic Regression for its simplicity and interpretability, SVC for its effectiveness in high-dimensional data, and Random Forest for its ability to handle non-linearity and complex relationships in the data.

Model Performance Comparison

To evaluate and compare the performance of different models, I gathered key metrics such as **Accuracy**, **Precision**, **Recall**, and **F1 Score** for each model, including Logistic Regression, Random Forest, SVC, and the Voting Classifier.

The results were organized into a table, and then I visualized the performance using a bar chart to highlight how each model performed across the various metrics.

- The **Accuracy** metric reflects how often the model correctly classified instances, showing the overall effectiveness.
- **Precision** indicates how well the model performed in predicting positive cases, while minimizing false positives.
- **Recall** measures the model's ability to identify all actual positive cases.
- The **F1 Score** provides a harmonic mean of precision and recall, offering a single metric that balances the two.



To assess the performance of each classification model, four key metrics—**Accuracy**,

Precision, **Recall**, and **F1 Score**—were evaluated for **Logistic Regression**, **Random Forest**, **Support Vector Machine (SVM)**, and the **Voting Classifier**. The confusion matrices were also generated to provide further insight into the models' performance. The results were as follows:

- **Logistic Regression** achieved an **Accuracy** of 86.15%, with a high **Precision** of 89.13% and a **Recall** of 91.11%, resulting in an **F1 Score** of 90.11%. Its confusion matrix revealed 15 true negatives and 41 true positives, but also showed 5 false positives and 4 false negatives.
- **Random Forest** produced a similar **Accuracy** of 86.15%, with slightly lower **Precision** at 86.00%, but a higher **Recall** of 95.56%, leading to an **F1 Score** of 90.53%. The confusion matrix showed 13 true negatives, 43 true positives, 7 false positives, and 2 false negatives, indicating strong performance in minimizing false negatives.
- **SVM** matched the **Accuracy** of 86.15%, with **Precision** at 84.62% and the highest **Recall** at 97.78%, resulting in an **F1 Score** of 90.72%. It had 12 true negatives, 44 true positives, 8 false positives, and 1 false negative, making it particularly effective at identifying positive instances.
- The **Voting Classifier** demonstrated solid performance with an **Accuracy** of 83.08%, **Precision** of 82.69%, and **Recall** of 95.56%, yielding an **F1 Score** of 88.66%. Its confusion matrix included 11 true negatives, 43 true positives, 9 false positives, and 2 false negatives.

These metrics were plotted on a bar graph for better visual comparison, showing how each model performed across the different evaluation criteria. The graph revealed that while **Random Forest** and **SVM** performed similarly overall, **SVM** excelled slightly in recall, and **Random Forest** achieved a better balance between precision and recall. The **Voting Classifier** performed reasonably well, but with slightly lower precision compared to the other models.

This analysis provides a clear picture of each model's strengths and areas for improvement, aiding in the selection of the most effective model based on the task's requirements.

Final Overall Interpretation

The performance comparison of the four models—**Logistic Regression**, **Random Forest**, **SVM**, and the **Voting Classifier**—demonstrates that all models are highly effective in classifying the dataset, with accuracies ranging between 83.08% and 86.15%. Each model has its own strengths, and the choice of the final model will depend on the specific objectives of the task.

- **Logistic Regression** shows the highest **precision** (89.13%) and a strong **F1 score** (90.11%), making it an ideal choice when it is crucial to minimize false positives. Its high precision indicates that when Logistic Regression predicts a positive class, it is more likely to be correct than the other models.
- **Random Forest** achieves the same **accuracy** (86.15%) but shines in **recall**

- (95.56%), meaning it effectively captures nearly all positive cases, making it a strong candidate for tasks where identifying all positives is critical. It balances precision and recall well, reflected in its high **F1 score** (90.53%).
- **SVM** matches the **accuracy** of Logistic Regression and Random Forest (86.15%) and has the highest **recall** (97.78%), ensuring that almost every positive case is detected. However, its **precision** (84.62%) is slightly lower, meaning it is more prone to false positives. Nonetheless, its **F1 score** (90.72%) indicates excellent overall performance.
- **Voting Classifier**, while combining the strengths of multiple models, performs slightly lower with an **accuracy** of 83.08%. It maintains a strong **recall** (95.56%) but has the lowest **precision** (82.69%), leading to a slightly reduced **F1 score** (88.66%) compared to the other models.

Conclusion

Overall, **Logistic Regression** is the best-performing model in terms of balancing precision and recall, making it a reliable choice for situations where minimizing false positives is important. **Random Forest** and **SVM** provide excellent recall, making them ideal for use cases where capturing all positive instances is paramount, such as in medical diagnoses or fraud detection. **Voting Classifier** offers a reasonable trade-off but may not be the optimal choice if precision is crucial.

In conclusion, depending on the specific objectives of the task—whether to prioritize precision or recall—either **Logistic Regression** or **Random Forest**/ **SVM** would be the preferred final models.