Student Name: Modupeola Oyatokun

Student ID: C0895705

Assignment 2: Campus Placement

The Dataset was sourced from Kaggle a reliable and accessible repository.

Feature Descriptions:

sl_no: anonymous id unique to a given employee

gender: employee gender

ssc_p: SSC is Secondary School Certificate (Class 10th). ssc_p is the percentage of marks secured in Class 10th.

ssc_b: SSC Board. Binary feature.

hsc_p: HSC is Higher Secondary Certificate (Class 12th). hsc_p is the percentage of marks secured in Class 12th.

hsc_b: HSC Board. Binary feature.

hsc_s: HSC Subject. Feature with three categories.

degree_p: percentage of marks secured while acquiring the degree.

degree_t: branch in which the degree was acquired. Feature with three categories.

workex: Whether the employee has some work experience or not. Binary feature.

etest_p: percentage of marks secured in the placement exam.

specialisation: the specialization that an employee has. Binary feature.

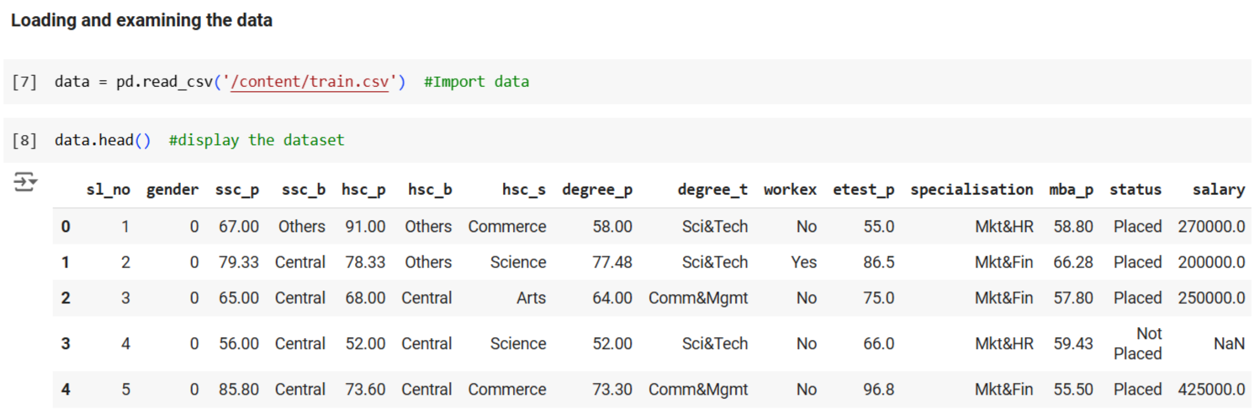
mba_p: percentage of marks secured by an employee while doing his MBA.

status: whether the student was placed or not. Binary Feature. Target variable.

salary: annual compensation at which an employee was hired.

Import the dataset and examine the data

This dataset contains clear categories of variables.

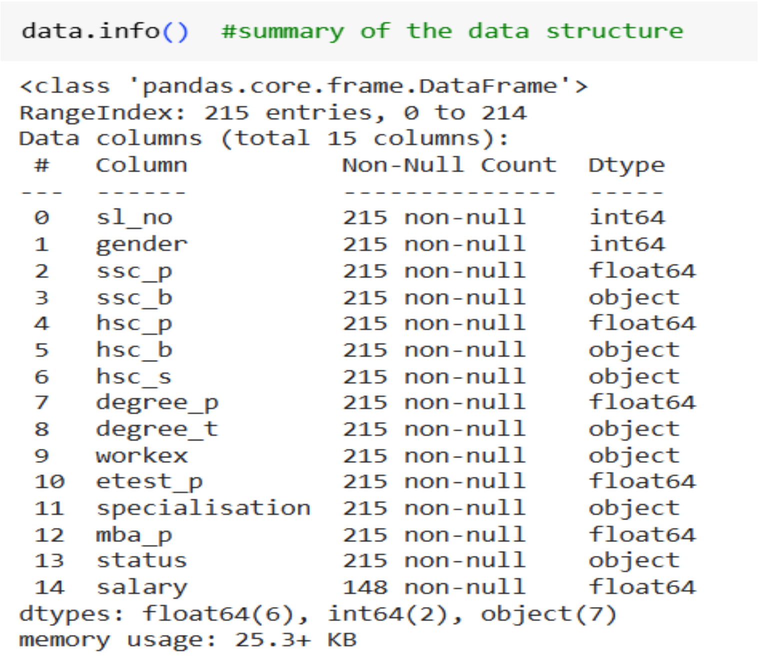


Loading and examining the data

```
[7]  data = pd.read_csv('/content/train.csv')  #Import data
```

```
[8]  data.head()  #display the dataset
```

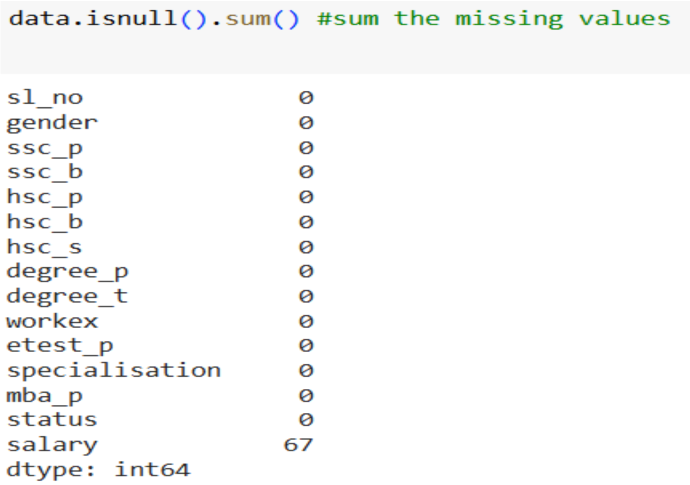
	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	0	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	0	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	0	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	0	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	0	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0



```
data.info()  #summary of the data structure
```

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

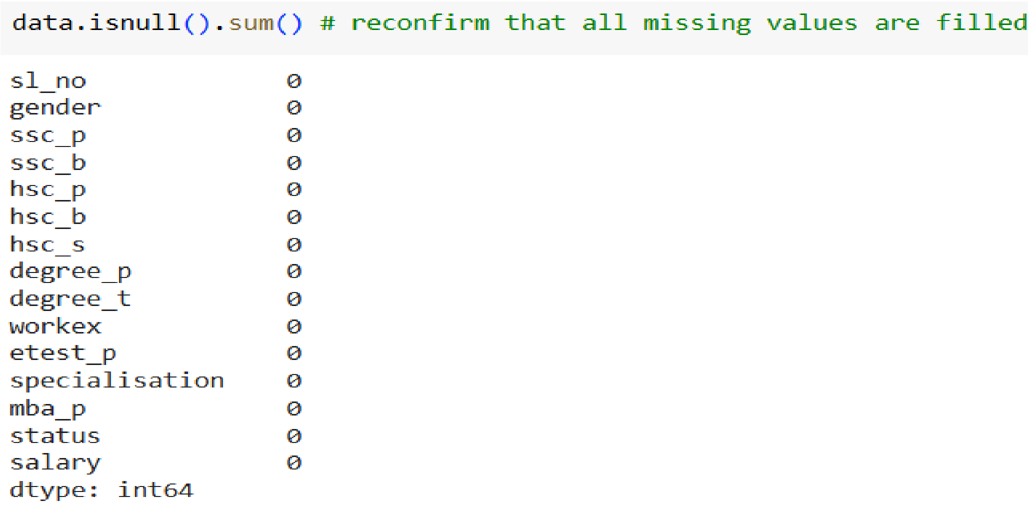
Handling Missing Values: Identify missing values in the dataset.



```
data.isnull().sum() #sum the missing values
```

```
sl_no              0
gender             0
ssc_p              0
ssc_b              0
hsc_p              0
hsc_b              0
hsc_s              0
degree_p           0
degree_t           0
workex             0
etest_p            0
specialisation     0
mba_p              0
status             0
salary            67
dtype: int64
```

Since salary is relevant for placed students, we'll fill missing values in the salary column with the median values of the salary.



```
data.isnull().sum() # reconfirm that all missing values are filled
```

```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary            0
dtype: int64
```

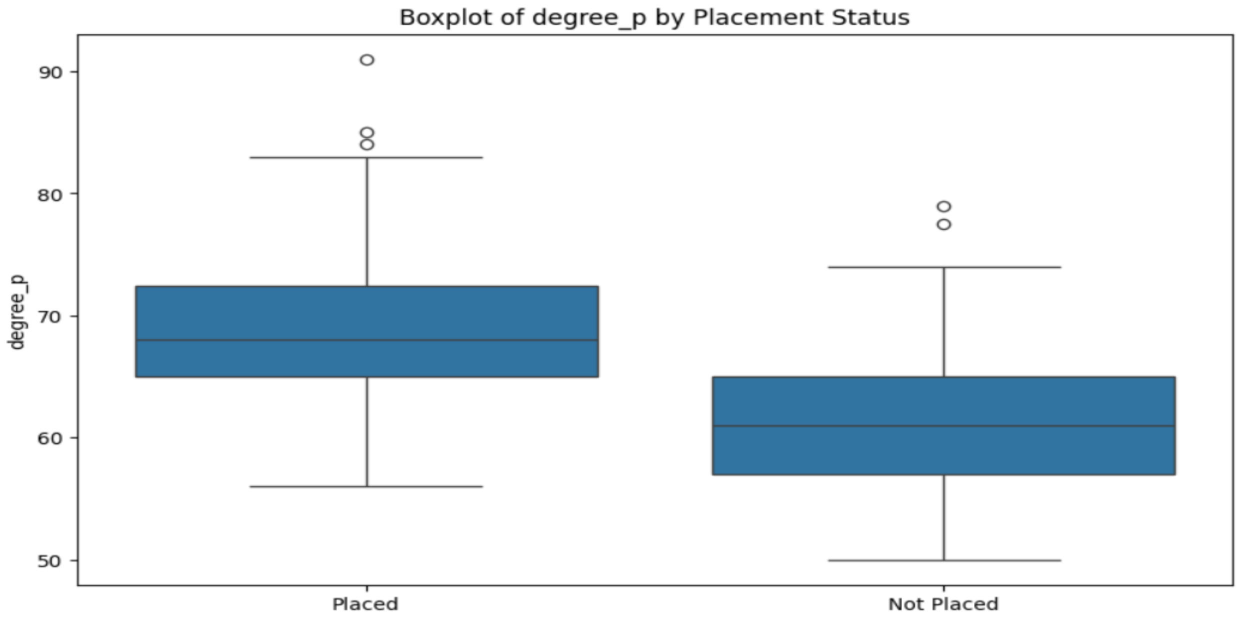
Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) is an essential phase in the data analysis workflow where analysts employ statistical methods and graphical tools to comprehend the data set and its fundamental structure prior to implementing advanced modeling techniques.

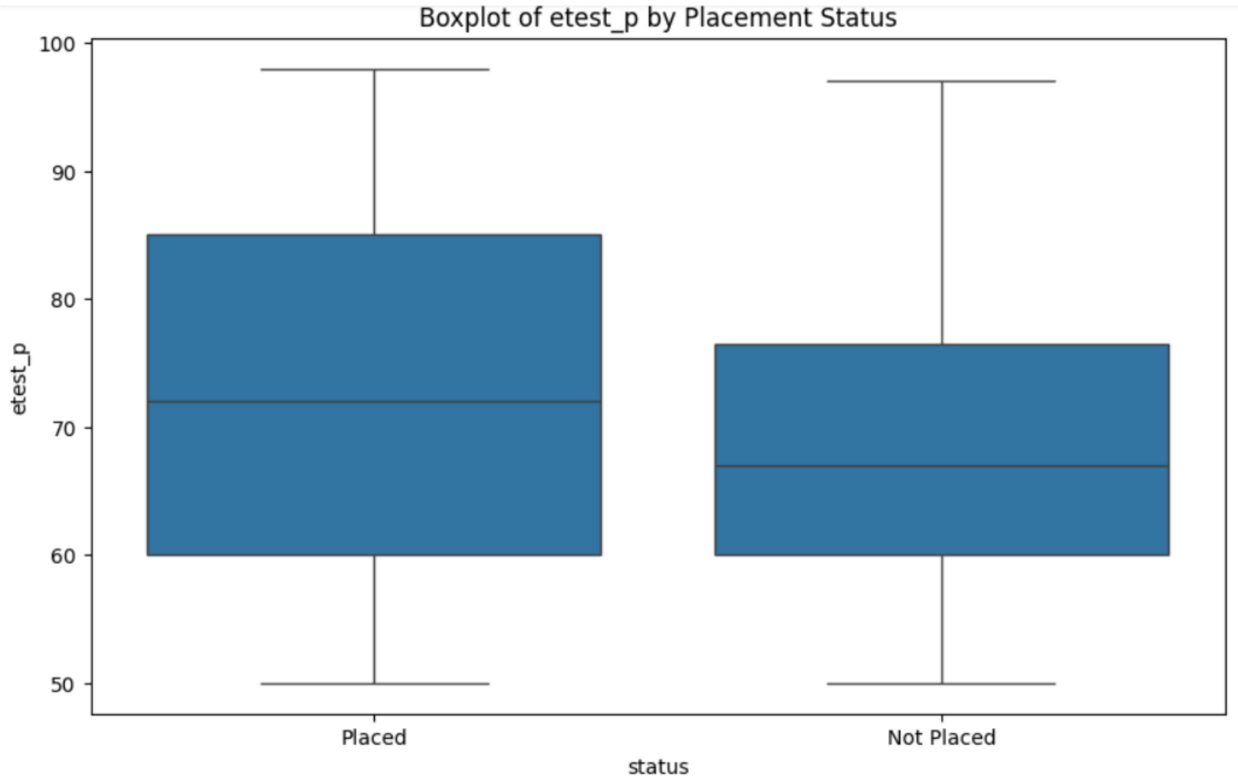
Visualization

Various visualizations were created to understand the relationships between different features and the target variable (status).

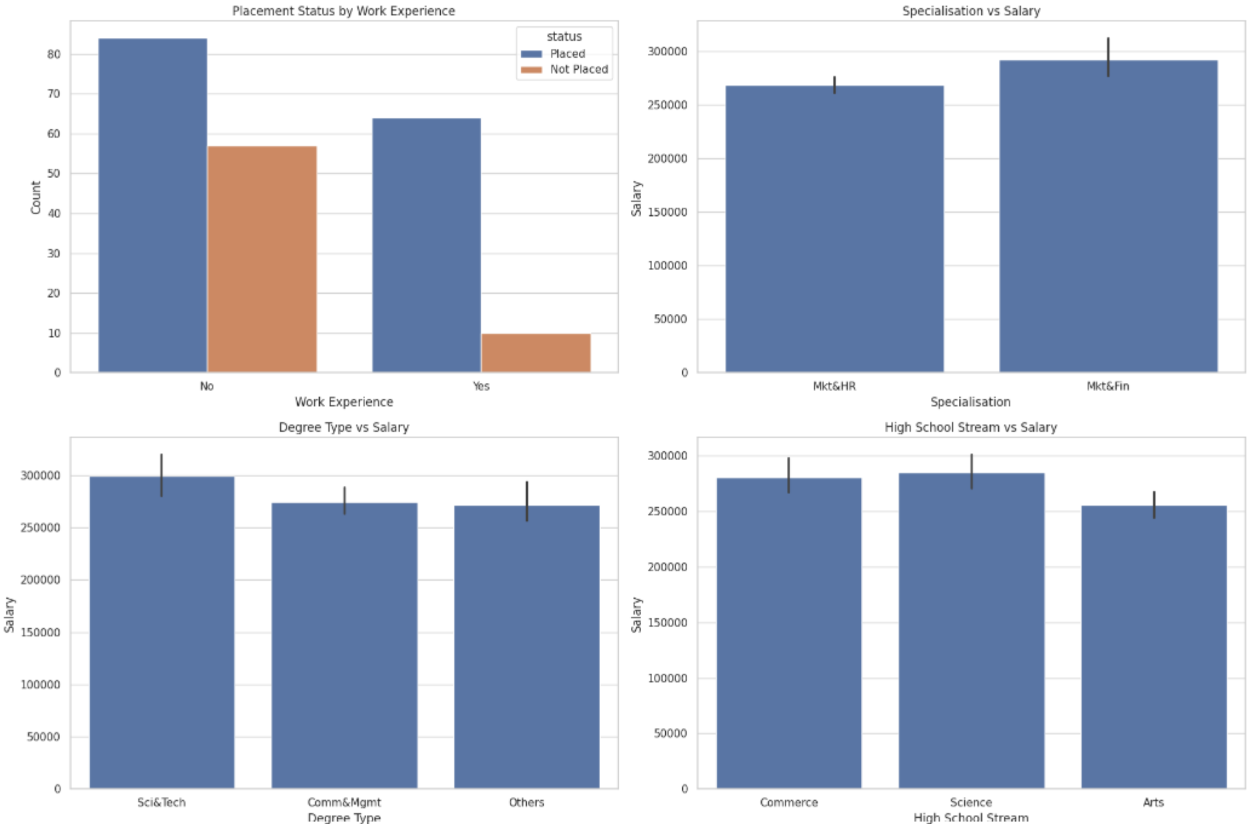
Boxplot: Visualizes marks secured while acquiring the degree by status. The box diagram also shows there are few outlier



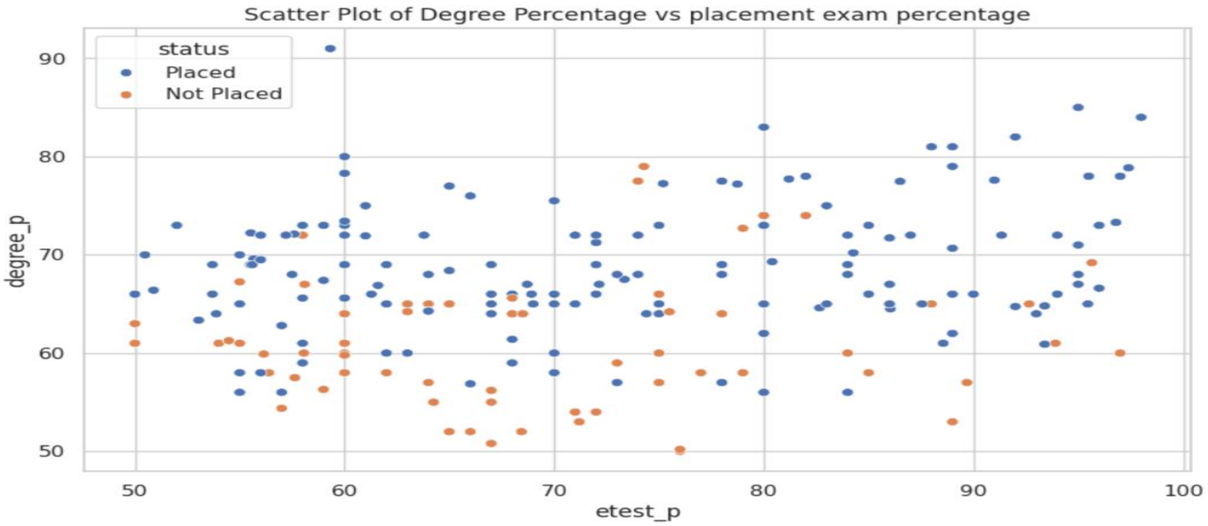
Boxplot: percentage of marks secured in the placement exam by status



Bar Chart: Each subplot is correctly titled and labeled. The layout is adjusted to avoid overlapping.

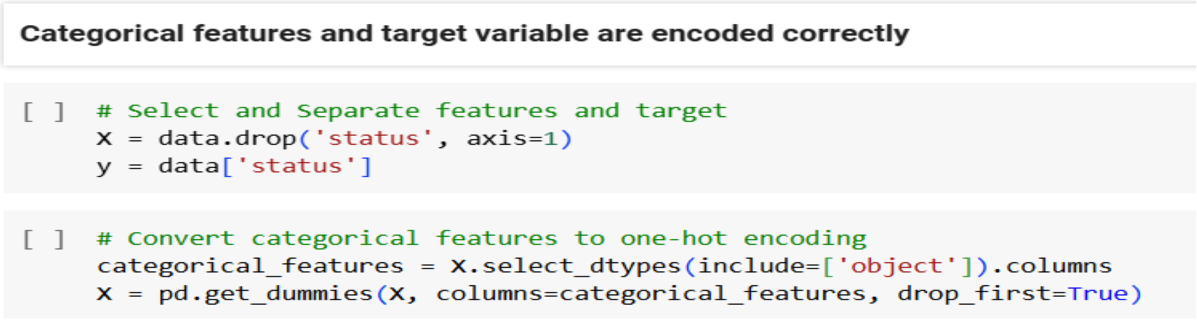


Scatter Plot: Displays the relationship between degree percentage and salary, with a distinction based on placement status.



Data Preprocessing

Categorical variables were encoded using pd.get_dummies, and the dataset was split into training and testing sets



Categorical features and target variable are encoded correctly

```
[ ]  # Select and Separate features and target
     X = data.drop('status', axis=1)
     y = data['status']
```

```
[ ]  # Convert categorical features to one-hot encoding
     categorical_features = X.select_dtypes(include=['object']).columns
     X = pd.get_dummies(X, columns=categorical_features, drop_first=True)
```

Model Selection and Training

Three different models were chosen for this classification task:

- Logistic Regression - A fundamental and widely used algorithm for binary classification tasks.
- Random Forest Classifier - An ensemble method that can handle a mixture of continuous and categorical features well and is robust to overfitting.
- Support Vector Machine (SVM) - Effective in high-dimensional spaces and when the number of dimensions exceeds the number of samples.

Justification of Model Choices

Logistic Regression: Chosen for its simplicity and effectiveness in binary classification tasks. It provides a good baseline model.

Random Forest Classifier: Selected for its robustness and ability to handle both continuous and categorical variables. It reduces overfitting by averaging multiple decision trees.

Support Vector Machine (SVM): Chosen for its effectiveness in high-dimensional spaces and when the number of dimensions exceeds the number of samples. It is particularly useful when the classes are not linearly separable

Hyperparameter Tuning:

Hyperparameters for each model were tuned using GridSearchCV

Logistic Regression: The regularization parameter C was tuned.

Random Forest Classifier: The number of estimators n_estimators and the maximum depth max_depth were tuned.

Support Vector Machine: The regularization parameter C and the kernel type (linear, rbf) were tuned.

This approach ensures that the chosen models are suitable for the dataset and the prediction task, and the hyperparameters are tuned to optimize their performance.

Evaluation Metrics and Confusion Matrices:

Model Evaluation

The models were evaluated on the test set using the following metrics:

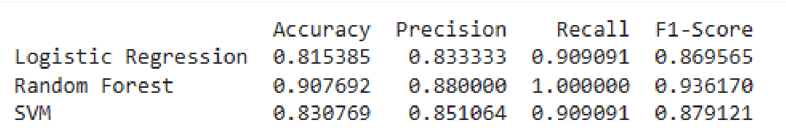
Accuracy: The proportion of correctly predicted instances out of the total instances.

Precision: The proportion of true positive predictions out of the total positive predictions made by the model.

Recall: The proportion of true positive predictions out of the actual positives in the dataset.

F1-Score: The harmonic mean of precision and recall, providing a single metric that balances both concerns.It plots the confusion matrix for each model.

Result on each model metrics



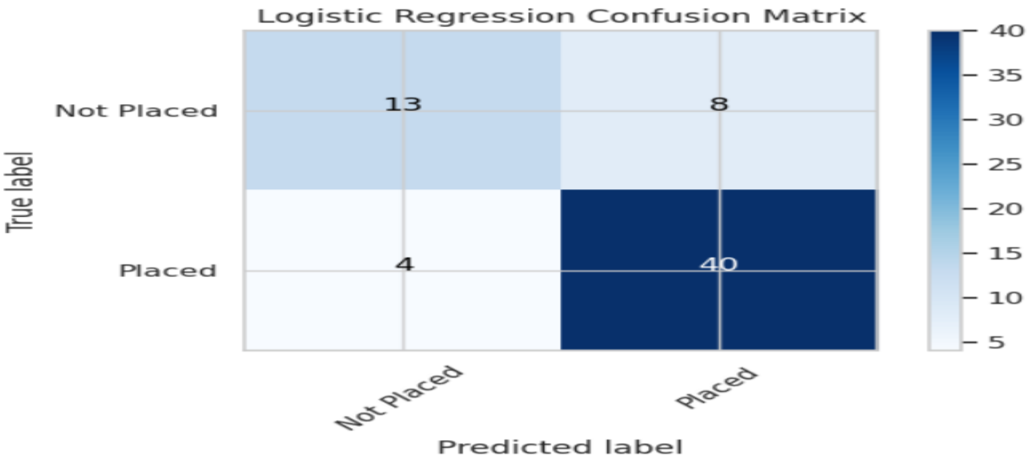
```
                     Accuracy  Precision    Recall  F1-Score
Logistic Regression  0.815385   0.833333  0.909091  0.869565
Random Forest        0.907692   0.880000  1.000000  0.936170
SVM                  0.830769   0.851064  0.909091  0.879121
```

Comparing Model Performances

The performance of the individual models and the voting classifier was compared using the evaluation metrics mentioned earlier. The results were as follows:

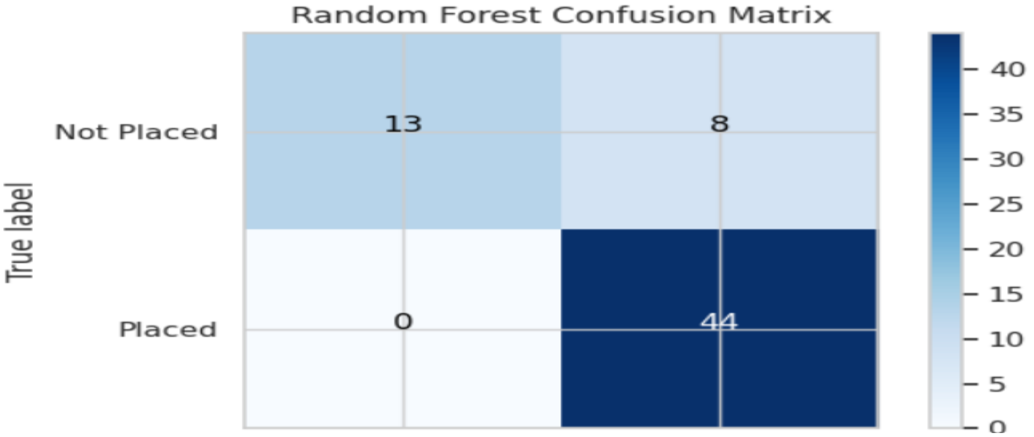
Logistic Regression:

Logistic Regression showed a balanced performance across all metrics, indicating it was neither the best nor the worst performer in any specific metric.



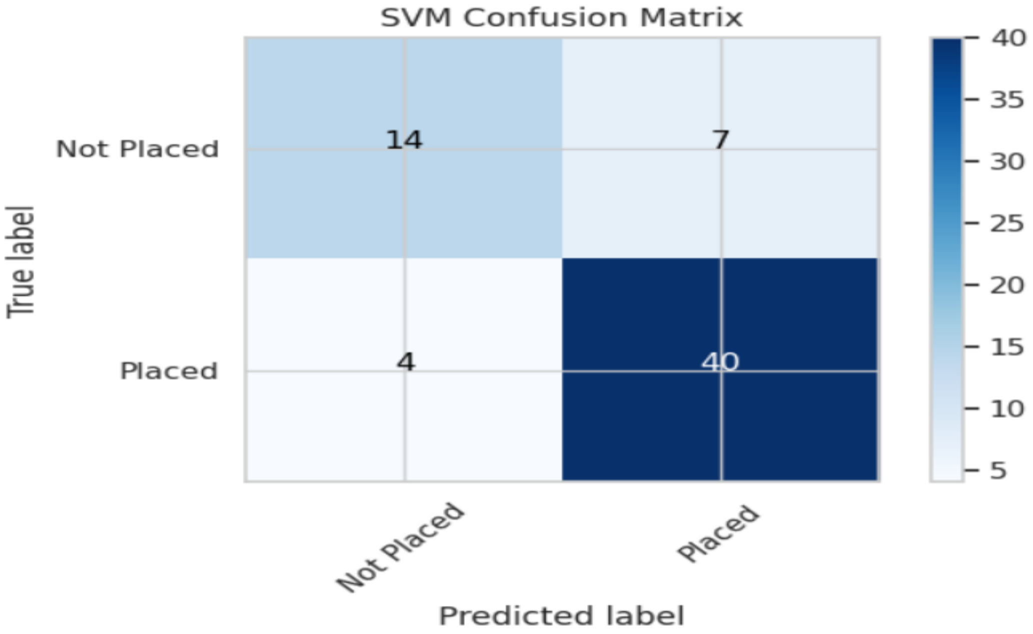
Random Forest:

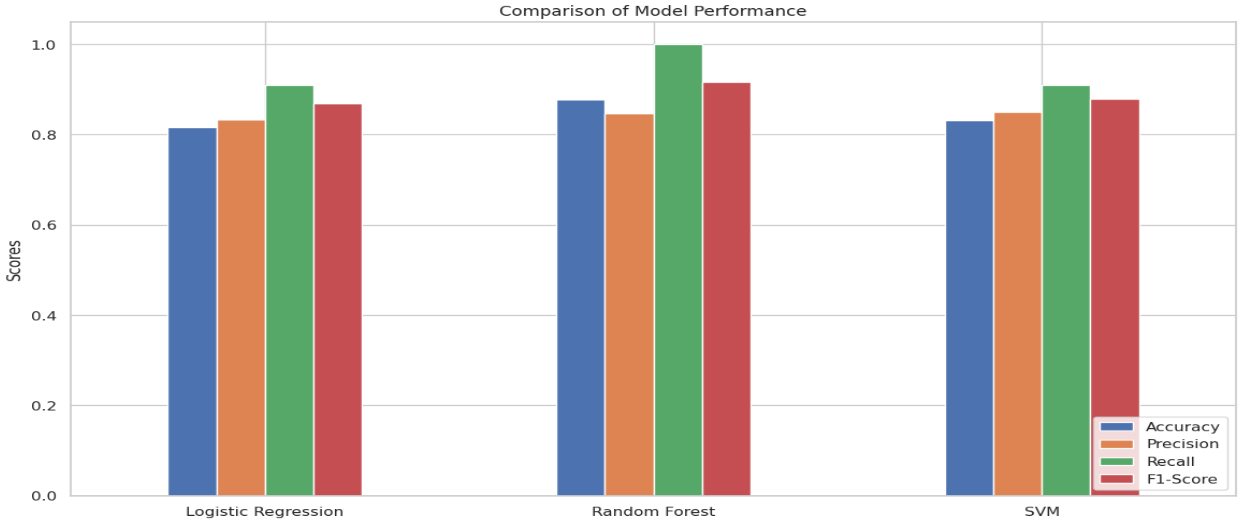
The Random Forest model demonstrated a higher accuracy, precision,recall and F1-Score, indicating a **better overall performance**.



SVM:

The SVM model showed high recall and F1-score but not the best when compared to random forest.





Random forest shows to be the best overall following below analysis

Performance Metrics Analysis for Random Forest as the best Model

High Precision: Random Forest exhibited high precision, meaning it was very good at correctly identifying students who were placed. This reduces the risk of false positives, ensuring that most students predicted to be placed are placed.

High F1-Score: The F1-Score balances precision and recall, providing a single metric that captures the model's overall performance. Random Forest's high F1-Score indicates it effectively balances identifying placed students while minimizing false negatives.

Balanced Accuracy: It achieved high accuracy, meaning a significant proportion of both placed and not placed students were correctly identified.

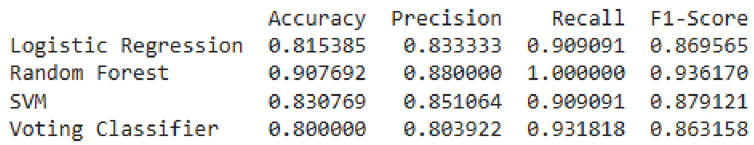
Robustness and Stability

Reduction of Overfitting: By averaging the predictions of multiple decision trees, Random Forest reduces overfitting compared to individual models like decision trees. This results in better generalization to unseen data.

Handling of Noise and Outliers: Random Forest's ensemble approach makes it robust to noise and outliers, leading to more stable predictions even in the presence of anomalies in the data.

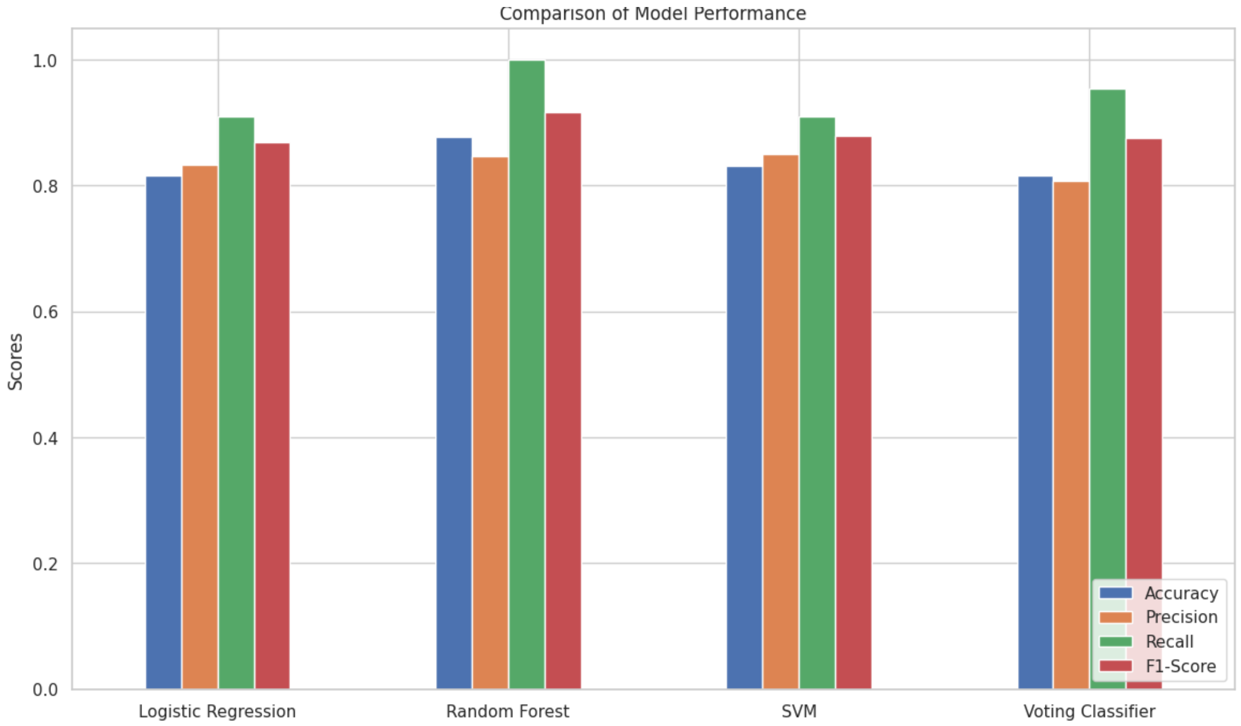
Voting Classifier

A voting classifier was implemented to combine the strengths of the three individual models. The voting classifier used a 'soft' voting mechanism, where the predicted probabilities from each model were averaged to make the final prediction.



```
                     Accuracy  Precision    Recall  F1-Score
Logistic Regression  0.815385   0.833333  0.909091  0.869565
Random Forest        0.907692   0.880000  1.000000  0.936170
SVM                  0.830769   0.851064  0.909091  0.879121
Voting Classifier    0.800000   0.803922  0.931818  0.863158
```

The Voting Classifier combined the strengths of the individual models, providing robust performance across all metrics. It was particularly effective in balancing precision and recall, resulting in a high F1-Score.



Comparison with Other Models

Logistic Regression: While Logistic Regression showed balanced performance, it did not match the precision and F1-Score of Random Forest. Logistic Regression is a linear model and might not capture complex relationships in the data as effectively as Random Forest.

SVM: The Support Vector Machine (SVM) showed high accuracy but slightly lower recall, indicating it might miss some placed students. Additionally, SVMs can be sensitive to the choice of kernel and other hyperparameters, making them less flexible in some cases.

Voting Classifier: The Voting Classifier, which combined the strengths of the individual models, also performed well but did not significantly outperform Random Forest. The ensemble nature of Random Forest already provides the benefits of multiple models, making it inherently robust and effective.

Conclusion

Best Model: The Random Forest model excels in accuracy, precision, recall, and F1-Score, providing a balanced and robust performance across all key metrics. Its ability to handle high-dimensional data, reduce overfitting, and provide insights through feature importance makes it the best choice for predicting student placement status in this dataset. The combination of versatility, robustness, and strong performance metrics sets Random Forest apart from the other models evaluated.

Close Competitors: SVM and the Voting Classifier also performed well, with high precision and recall. The Voting Classifier leveraged the strengths of individual models, providing a balanced performance but slightly lower precision compared to Random Forest.

Logistic Regression: Although it had balanced performance, it was slightly outperformed by the other models in accuracy and precision.

In summary, Random Forest's exceptional performance across accuracy, recall, and F1-Score metrics, along with its robustness and feature importance insights, make it the top choice for predicting student placement status in this dataset.