CAMPUS RECRUITMENT PREDICTION

Neural Networks and Deep Learning (Assignment-2)

Student Name: Devi Samyuktha Chitturi Student ID: c0901961

DATASET OVERVIEW

In this project we are going to utilize the **Campus Recruitment** Dataset from Kaggle which consists of various features which might influence the Placement of Student in Jobs. The "Campus Placement Prediction" dataset encapsulates a comprehensive array of attributes aimed at predicting the outcome of candidate selection during campus placement processes. This dataset offers valuable insights into the factors influencing a candidate's success in securing placement opportunities within various academic institutions and corporate entities.

DATA LINK

https://www.kaggle.com/c/ml-with-python-course-project/data

It contains 215 records (rows) and 14 features (columns). This relatively small dataset size makes it manageable for analysis and modeling, allowing for efficient processing and quicker computation times. The structure of the dataset, comprising numerous features, provides a solid foundation for extracting insights and performing various machine learning tasks.

FEATURES

1. Gender (Categorical): Represents the gender identity of the candidate participating in the placement process.
2. Secondary Education Percentage (Numerical): Denotes the percentage score obtained by candidates in their secondary education.
3. Secondary Education Board (Categorical): Indicates the educational board associated with the candidate's secondary education.
4. Higher Secondary Education Percentage (Numerical): Reflects the percentage score attained by candidates in their higher secondary education.
5. Higher Secondary Education Board (Categorical): Identifies the educational board governing the candidate's higher secondary education.
6. Higher Secondary Education Stream (Categorical): Specifies the academic stream pursued by candidates during their higher secondary education.
7. Undergraduate Degree Percentage (Numerical): Signifies the percentage score achieved by candidates in their undergraduate degree program.
8. Undergraduate Degree Type (Categorical): Characterizes the type of undergraduate degree pursued by candidates.
9. Work Experience (Categorical): Indicates whether candidates possess prior work experience.
10. Employability Test Percentage (Numerical): Represents the percentage score obtained by candidates in employability tests.
11. MBA Percentage (Numerical): Indicates the percentage score attained by candidates in their Master of Business Administration (MBA) program.
12. Specialization (Categorical): Specifies the specialization area of candidates in their MBA program.
13. Placement Status (Categorical): Serves as the target variable, indicating whether candidates were placed or not during the campus placement process.

OBJECTIVE

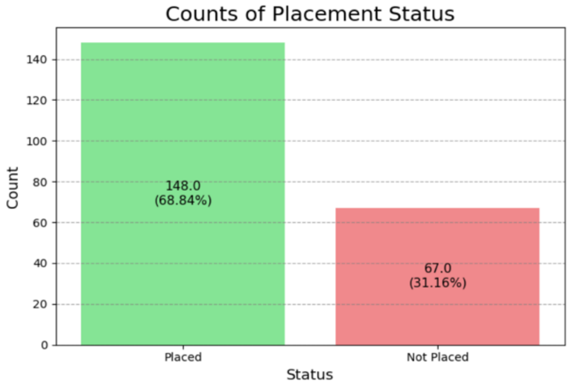
The primary objective of this dataset is to facilitate the development of robust predictive models that accurately discern the likelihood of candidate selection during campus placements. By leveraging a diverse array of candidate attributes, encompassing academic performance, gender, work experience, and specialization areas, this dataset enables the exploration of underlying patterns and predictive relationships that significantly influence placement outcomes.

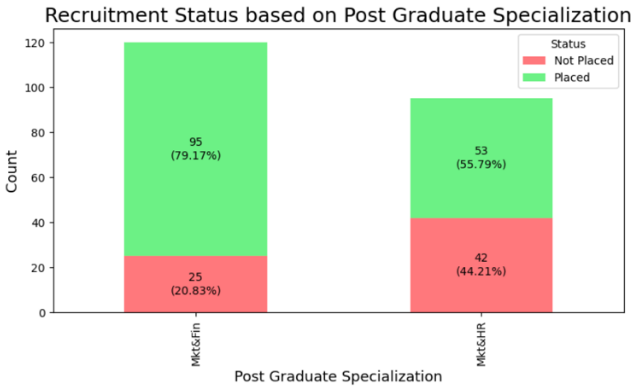
Predict Class label (y): status.

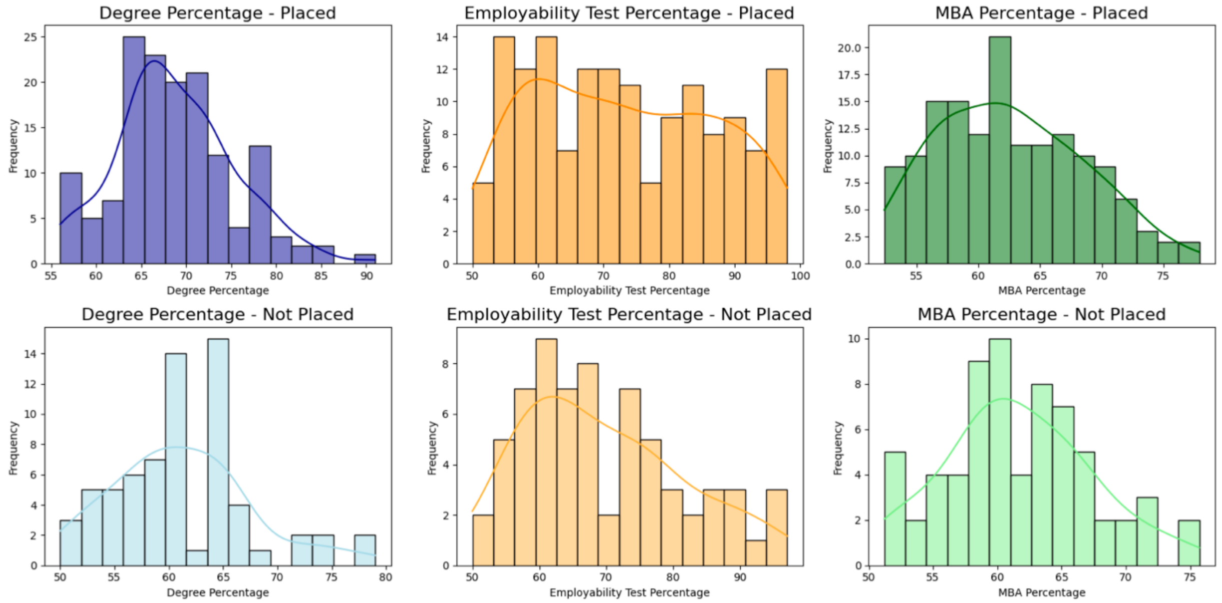
EXPLORATORY DATA ANALYIS

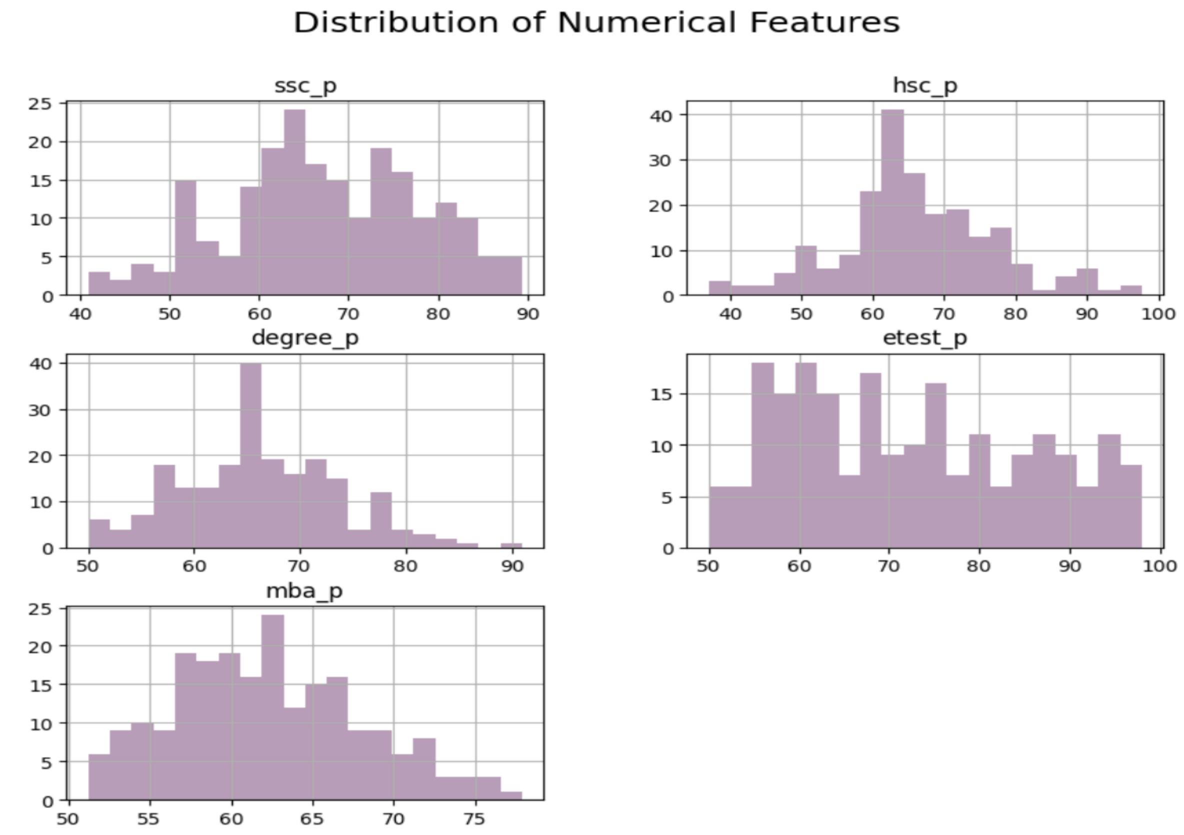
In this exploratory data analysis (EDA) of the **Campus Placement** dataset, several key aspects were examined to understand the data's structure and quality.

1. **Null Values**: The dataset consists of 215 records across 15 features. A check for null values revealed that all columns, except for the **salary** column, were complete, indicating that there are 67 missing entries in the salary field. These null values correspond to students who were not placed, highlighting the need to address this in further analysis.
2. **Duplicate Records**: The analysis confirmed that there are no duplicate records in the dataset, ensuring the integrity of the data.
3. **Placement Status**: A count of the placement status showed that out of 215 students, 148 were placed while 67 were not. This indicates a higher number of students successfully securing placements. However, the class distribution reveals an imbalance, suggesting the potential for using oversampling or under sampling techniques in subsequent modeling efforts.
4. **Outliers**: The examination of numerical features through histograms and boxplots indicated that there are few to no outliers present, suggesting a cleaner dataset conducive for modeling.









DATA PREPROCESSING AND HANDLING CLASS IMBALANCE

In this phase, we prepared the **Campus Placement** dataset for modeling by performing data preprocessing and addressing class imbalance.

1. **Data Cleaning**: We started by dropping unnecessary columns, specifically sl_no (serial number) and salary, as they do not contribute to the predictive modeling process.
2. **Feature and Target Variable Definition**:
   - The features (X) were defined by excluding the target variable status, which indicates placement outcomes.
   - The target variable (y) was encoded to a binary format, where Placed was mapped to 1 and Not Placed to 0. This encoding facilitates the use of machine learning algorithms.
3. **Feature Specification**: We categorized the features into different types:
   - **Numerical Columns**: Included percentage scores from various education levels (e.g., ssc_p, hsc_p, degree_p, etest_p, mba_p).
   - **Binary Columns**: Features with binary values, such as gender, ssc_b, and hsc_b, were identified.
   - **Categorical Columns**: Features with multiple categories included degree_t, specialisation, and workex.

4. **Shape Verification**: We printed the shapes of the feature and target variables to ensure they matched expectations:
   - Features shape: (215, 12)
   - Target shape: (215,)
5. **Pipeline Creation**: We established a preprocessing pipeline:
   - A numerical transformer was implemented using StandardScaler to standardize numerical features.
   - Binary features were processed using OneHotEncoder to encode them into binary values.
   - Multi-class categorical features were transformed using OneHotEncoder, dropping the first category to avoid dummy variable traps.
6. **Data Splitting**: The dataset was split into training and testing sets, with 80% allocated for training and 20% for testing.
7. **Transformation**: The preprocessing pipeline was fitted on the training data, transforming it for further analysis.
8. **Class Imbalance Handling**: To address class imbalance in the training set, we utilized SMOTE (Synthetic Minority Over-sampling Technique). This method oversampled the minority class to create a more balanced dataset, ensuring that the model training process is less biased towards the majority class.
9. **Testing Data Transformation**: Finally, the test data was transformed using the fitted pipeline to prevent data leakage.

This comprehensive preprocessing and imbalance handling strategy ensures that the dataset is well-prepared for effective model training and evaluation.

MODEL SELECTION AND TRAINING

In this phase, we evaluated and compared four different classification models to predict campus placement outcomes: **Logistic Regression**, **Support Vector Machine (SVM)**, **Decision Tree**, and **Random Forest**. Each model was trained on the balanced training dataset generated using SMOTE and evaluated on the test set using multiple metrics.

1. **Logistic Regression**:
   - **Accuracy**: 0.8372
   - **Precision**: 0.9
   - **Recall**: 0.871
   - **F1 Score**: 0.8852
   - Logistic Regression demonstrated solid performance, effectively capturing the relationship between student features and placement outcomes. Its interpretability makes it a good choice for understanding the influence of various factors on placement.
2. **Support Vector Machine (SVM)**:
   - **Accuracy**: 0.8605
   - **Precision**: 0.9032

- **Recall**: 0.9032
- **F1 Score**: 0.9032
- The SVM model outperformed all other models across all metrics, indicating its strong capability to distinguish between placed and not placed students. Its robustness in handling high-dimensional data and flexibility with different kernels makes it a preferred choice for this classification task.

3. **Decision Tree**:
   - **Accuracy**: 0.814
   - **Precision**: 0.871
   - **Recall**: 0.871
   - **F1 Score**: 0.871
   - While Decision Trees provided reasonable performance, they did not match the effectiveness of the top two models. Their interpretability is a strength, but they can be prone to overfitting, making them less favorable for this application.
4. **Random Forest**:
   - **Accuracy**: 0.7674
   - **Precision**: 0.8182
   - **Recall**: 0.871
   - **F1 Score**: 0.8438
   - The Random Forest model was eliminated from consideration due to its significantly lower accuracy and precision compared to the other models. Although it performed adequately in recall, its overall reliability was insufficient for the specific needs of this classification problem.

Conclusion of Model Comparison

Based on the evaluation metrics, the **Support Vector Machine (SVM)** emerged as the best model for predicting campus placements, demonstrating the highest overall performance in accuracy, precision, recall, and F1 score. This superior performance indicates its capability to effectively differentiate between students who secured placements and those who did not.

Logistic Regression followed closely, showing solid performance while remaining interpretable. On the other hand, **Decision Trees** provided reasonable results but lacked the performance levels of SVM and Logistic Regression. **Random Forest** was excluded from further consideration due to its notably lower effectiveness, which could compromise the reliability of predictions in this application.

In summary, SVM was selected as the optimal model due to its high predictive accuracy and reliability, while Logistic Regression served as a strong alternative. Decision Trees and Random Forest did not meet the required performance standards for this specific classification task.

HYPERPARAMETER TUNING AND CROSS-VALIDATION

In this section, we conducted hyperparameter tuning and cross-validation for three selected models: **Logistic Regression**, **Support Vector Machine (SVM)**, and **Decision Tree**.

Hyperparameter Tuning

Using **GridSearchCV**, we optimized the hyperparameters for each model:

- **Logistic Regression**:
  - Best Parameters: {'C': 0.1, 'solver': 'lbfgs'}
  - Best F1 Score: 0.8547
- **Support Vector Machine (SVM)**:
  - Best Parameters: {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
  - Best F1 Score: 0.8797
- **Decision Tree**:
  - Best Parameters: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 10}
  - Best F1 Score: 0.8277

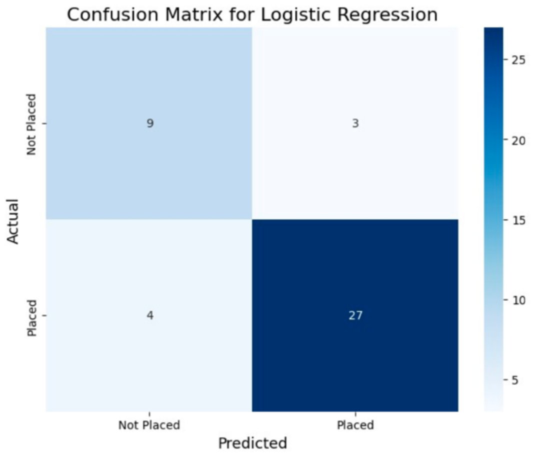
After tuning, Logistic Regression and SVM maintained their performance, while SVM achieved the highest F1 score, confirming its reliability. However, Decision Tree showed a decline in performance metrics, suggesting potential overfitting with the tuned parameters.

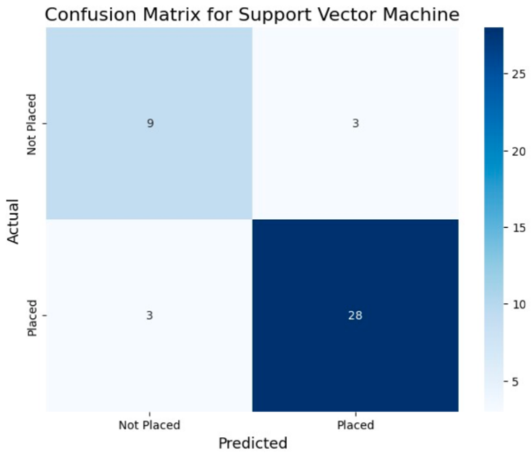
Cross-Validation

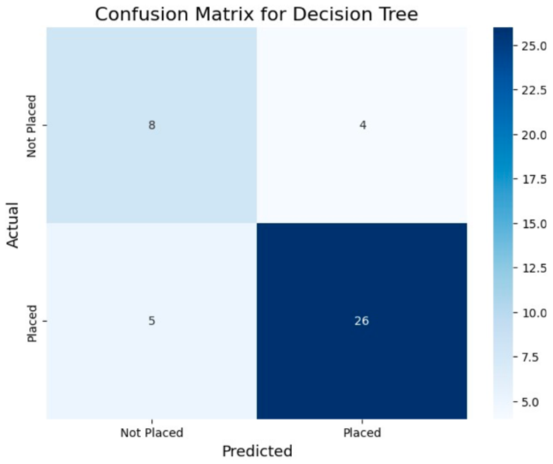
We further evaluated the models using 5-fold cross-validation to ensure robustness:

- **Logistic Regression**:
  - Mean F1 Score: 0.8547
  - Standard Deviation: 0.0318
- **Support Vector Machine (SVM)**:
  - Mean F1 Score: 0.8797
  - Standard Deviation: 0.0270
- **Decision Tree**:
  - Mean F1 Score: 0.8134
  - Standard Deviation: 0.0317

The **Support Vector Machine (SVM)** emerged as the best-performing model overall, demonstrating the highest accuracy (0.8605), F1 score (0.9032), and cross-validation mean F1 score (0.8797). This model's consistent performance validates its effectiveness in predicting placement outcomes.







VOTING CLASSIFIER SUMMARY

We implemented a **Voting Classifier** to combine the predictions of the best-performing models: **Logistic Regression**, **Support Vector Machine (SVM)**, and **Decision Tree**. The classifier used majority voting to aggregate predictions from these individual models.

Performance Metrics

The Voting Classifier achieved the following evaluation metrics on the test set:

- **Accuracy**: 0.8372
- **Precision**: 0.9000
- **Recall**: 0.8710
- **F1 Score**: 0.8852

Confusion Matrix

	Predicted Not Placed	**Predicted Placed**
Actual Not Placed	**9**	**3**
Actual Placed	**4**	**27**

ANALYSIS

The Voting Classifier demonstrated comparable performance to the **Logistic Regression** model, with the same accuracy. However, it outperformed Logistic Regression in terms of precision and F1 score, indicating improved handling of positive class predictions. Despite this, the **Support Vector Machine (SVM)** remains the top-performing model in this analysis, achieving the highest accuracy (86.05%) and F1 score (0.9032), making it the most effective choice for predicting placement outcomes in this dataset.

Finally, the trained Voting Classifier model was saved using **pickle** for future use.