Report on Campus Recruitment Process Analysis

Prepared By

Anand Vinoy

Student ID: C0902471

Professor:

Ishant Gupta

Course: Neural Network and Deep Learning

1. Introduction

The objective of this project is to predict whether a student will be successfully recruited in campus placements based on various academic and demographic factors available in the dataset. Accurate predictions in this area can assist educational institutions in identifying and supporting students who may require additional help to secure placements, ultimately enhancing the institution's reputation and attractiveness to future students.

2. Dataset Description and Preprocessing

Dataset Overview:
The dataset consists of multiple columns, including academic scores, gender, specializations, and other features that could influence campus placement outcomes. The target variable is a binary indicator of whether a student was placed or not.

Link to the dataset: Campus Recruitment Prediction (Kaggle)

Some of the Important Columns from the dataset includes:

A. Gender
B. Degree - The type of undergraduate degree the student pursued
C. Specialization - The student's specialization during their undergraduate degree (e.g., 'MKT' for Marketing, 'FIN' for Finance)
D. Status: ‘Placed’ or ‘Not Placed’
E. ssc_p - Secondary Education - Percentage of Marks
F. workex - Work Experience

Preprocessing Steps:

- **Data Cleaning:**
    - Unnecessary columns such as sl_no, ssc_b, and hsc_b were removed as they did not contribute meaningfully to the placement prediction task.
    - From our first look, we can see that most of the data is in the form of either numerical or categorical data.
    - Missing values were checked for each feature. The column for salary had missing values. So they were replaced by calculating the mean of the entire row and replaced the missing values with the mean value.

- **Outliers**: Outliers are data points that differ significantly from other observations and may affect the performance of machine learning models, leading to inaccurate predictions or biased results

    1. **Q1 (First Quartile)** is the value below which 25% of the data points fall. It is the median of the lower half of the dataset.
    2. **Q3 (Third Quartile)** is the value below which 75% of the data points fall. It is the median of the upper half of the dataset.
    3. The **Interquartile Range (IQR)** is the difference between the third quartile (Q3) and the first quartile (Q1). It measures the spread of the middle 50% of the data and is used as a basis to identify potential outliers. IQR is an important measure because it focuses on the "central" part of the distribution and excludes extreme values that may not be representative of the majority of the data.
    4. **Outliers below the lower bound:** Any data point smaller than Q1−1.5×IQRQ1 - 1.5 \times IQRQ1−1.5×IQR is considered a lower outlier.
    5. **Outliers above the upper bound:** Any data point larger than Q3+1.5×IQRQ3 + 1.5 \times IQRQ3+1.5×IQR is considered an upper outlier.

- **Encoding Categorical Variables:**

    - Gender and other categorical columns and Status column (Target variable) were encoded as numerical variables to make them compatible with machine learning algorithms.

    Columns Encoded:

    The following categorical columns were label-encoded:

    - **gender**: Gender of the student (e.g., "Male" → 0, "Female" → 1).
    - **workex**: Whether the student has prior work experience (e.g., "Yes" → 1, "No" → 0).
    - **specialisation**: Type of specialisation in the MBA program (e.g., "Mkt & HR" → 0, "Mkt & Fin" → 1).
    - **status**: Placement status of the student (e.g., "Placed" → 1, "Not Placed" → 0).

- **Splitting the Data:**
    - The data was split into a training set (70%) and a test set (30%) using a stratified approach to ensure both classes in the target variable were represented proportionally.
    - The dataset was split using the **train_test_split** function from Scikit-learn.
    - **70%** of the data was used for training, and **30%** was reserved for testing the model's performance.
    - The **random_state=42** ensures that the data split is consistent each time the code is run, which is important for reproducibility.
- **Correlation Matrix Interpretation**
    - The correlation matrix revealed the following insights:
        1. Strong Positive Correlations: ssc_p with status: 0.61, indicating that higher secondary education performance is positively related to placement status. hsc_p with status: 0.49, suggesting that students with better higher secondary education results have a higher likelihood of being placed. degree_p with status: 0.48, also showing that undergraduate performance correlates with placement success.
        2. Moderate Positive Correlations: The ssc_p, hsc_p, and degree_p variables show moderate correlations with each other, suggesting that a student's performance across different educational stages tends to be consistent.
        3. Weak Correlations: The variables etest_p, mba_p, and salary exhibit weak correlations with placement status, suggesting that these factors may not significantly influence whether a student gets placed.
- **Data Visualization**
  To better highlight the connections in the data, a number of visualizations were made:

  A. Box Plot: The ssc_p variable's box plot versus the placement status efficiently showed how secondary education percentages were distributed across placed and pupils who were not placed. It emphasized the existence of anomalies and showed that Students that were placed often had greater percentages of secondary education than to the people who were left out.

B. A scatter plot: A degree_p versus pay scatter plot that is colored according to placement status was supplied graphic representations of the relationship between wage offers and undergraduate achievement. It showed that students who were placed were offered greater salaries, supporting the significance of achieving high academic standing.

3. Model Selection

Models Used:
To create a robust classification model, we experimented with several machine learning algorithms, each selected for their unique strengths:

1. **Logistic Regression:**
   Logistic Regression is a linear model useful for binary classification tasks and provides interpretability of feature coefficients.
2. **Random Forest Classifier:**
   The Random Forest algorithm was chosen for its ability to handle non-linear relationships and its robustness to overfitting through ensemble learning.
3. **Support Vector Classifier (SVC):**
   SVC was used for its effectiveness in high-dimensional spaces and capability to work well with clear margin separation.
4. **K-Nearest Neighbors (KNN):**
   KNN is a simple, non-parametric method that was used as a baseline for comparison with other models.
5. **Decision Tree Classifier (Decision Tree Classifier)**
6. **Naive Bayes:**
   Naive Bayes was selected due to its simplicity and efficiency in handling categorical data, which is relevant given the nature of some features in this dataset.

4. Model Training

Confusion Matrix

A thorough analysis of the model's predictions in relation to the actual results was given by the confusion matrix.
Interpretation of the Confusion Matrix

True Positives (TP): The proportion of pupils who were placed based on accurate predictions.
True Negatives (TN): The proportion of pupils who were accurately projected to be placed elsewhere.
False Positives (FP): The quantity of pupils whose placement was mis predicted.

False Negatives (FN): The number of pupils that were wrongly predicted to not be placed.

5. Model Evaluation

Evaluation Metrics Used: The models were evaluated using key classification metrics to assess performance on the test set:

- **Accuracy:** Measures the overall correctness of the model.
- **Precision, Recall, and F1-Score:** Provide insight into the model's performance with respect to false positives and false negatives, particularly valuable in binary classification tasks.
- **Confusion Matrix:** A visual representation of true positives, false positives, true negatives, and false negatives to gain further insights into model performance.

Performance Comparison: The following summarizes the performance of each model on the test set:

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.79	0.87	0.85	0.86
Random Forest	0.87	0.91	0.91	0.91
Decision Tree	0.80	0.91	0.83	0.87
Voting Classifier	0.87	0.91	0.91	0.91

Model Performance

- **Logistic Regression**: Logistic Regression performed reasonably well with high precision, indicating that when it predicted a student would be placed, it was correct 87% of the time. However, its recall is slightly lower, meaning it missed a few

students who were placed. The F1-score is a balanced measure of precision and recall and is relatively good at 0.86.

- **Random Forest:** The Random Forest model achieved the highest accuracy and performed equally well in terms of precision, recall, and F1-score. It correctly predicted the placement status for 91% of the students. It showed a balanced ability to identify both placed and non-placed students, making it the best-performing model in this comparison.
- **Decision Tree:** The Decision Tree model showed high precision (91%) but had a slightly lower recall (83%). It struggled a bit in identifying all the students who were placed, resulting in a slightly lower recall compared to Random Forest. However, it still performed well with an overall accuracy of 0.80.
- **Voting Classifier:** The Voting Classifier, which combines the strengths of multiple models (including Random Forest), performed similarly to Random Forest. It achieved the same accuracy, precision, recall, and F1-score, making it a robust choice for this task.

Random Forest and Voting Classifier emerged as the best models, both achieving an accuracy of 0.87 and consistently high scores across all evaluation metrics (precision, recall, and F1-score).

Logistic Regression and **Decision Tree** performed slightly worse in terms of accuracy and recall but still showed solid results, particularly in precision.

The **Voting Classifier**'s ability to combine different models' strengths allows it to achieve performance on par with the Random Forest model, which is ideal for improving predictions without overfitting.

Overall, Random Forest and Voting Classifier are recommended for the placement prediction task due to their higher and balanced performance across all metrics.

Conclusion

Based on a number of academic and demographic variables, we created a classification model in this project to forecast if a student will be placed effectively during campus recruiting. Key characteristics of the dataset included job experience, specialization, and academic achievement (secondary and upper secondary education rates, undergraduate

performance), all of which are crucial in predicting placement results.

To get the data ready for machine learning models, preparation procedures included resolving outliers, eliminating unnecessary columns, fixing missing values, and encoding categorical variables. Insightful patterns, such a favorable association between academic achievement and placement status, were discovered using the exploratory data analysis and visualization tools.

To determine which machine learning model performed best for the job, a number of models were trained and assessed. K-Nearest Neighbors, Random Forest, Decision Tree, Support Vector Classifier, Naive Bayes, and Logistic Regression were among the models that were put to the test. Accuracy, precision, recall, F1-score, and the confusion matrix were among the measures used to evaluate each model's performance.

With an accuracy of 87% and excellent precision, recall, and F1-score values, the Random Forest and Voting Classifier models were the best performers. With balanced performance across both classes (placed and not placed), these models showed that they could effectively predict placement outcomes.

Decision trees and logistic regression performed marginally worse, notably in recall, but they still produced good results, particularly in accuracy.