A Report on Campus Recruitment Process Analysis

Submitted By:

Nirmala Regmi

Student ID: c0903616

Submitted to :

Ishant Gupta

Subject: Neural Networks and Deep Learning

1. Introduction

This report presents an analysis of the campus recruitment process, aiming to identify key factors influencing student placement outcomes. By examining various educational metrics, it provide insights that can help educational institutions improve their recruitment strategies and student employability.

2. Dataset Overview

This dataset was taken from Kaggle from this link Campus Recruitment Prediction (Course Project) | Kaggle

The dataset includes several features that are crucial for understanding the relationship between student performance and placement status. The key variables are:

- **ssc_p**: Percentage in Secondary Education
- **hsc_p**: Percentage in Higher Secondary Education
- **degree_p**: Percentage in Undergraduate Degree
- **etest_p**: Entrance Test Score
- **mba_p**: MBA Percentage
- **salary**: Salary offered to placed students
- **status**: Placement status (1 for placed, 0 for not placed)

Initial Exploration

Upon initial exploration of the dataset, we observed the following:

- **Data Types**: The dataset consists of numerical and categorical data. Most features are numerical, while the status variable is categorical.
- **Missing Values**: There were some missing values, and we replace them with the mean. To handle missing data in the salary column, we applied mean imputation. First, the mean salary was calculated using the existing non-missing values. Then, all missing values in the column were replaced with this calculated mean. This approach helps to retain the entire dataset by replacing missing values with a reasonable estimate, reducing potential bias or data loss.

- **Outliers**: Some extreme values were noted in the ssc_p and salary columns. These outliers were evaluated, and it was decided to remove extreme outliers to prevent skewing the results

  Step-by-step Explanation:

1. **Calculate Q1 (First Quartile)**:
   ```
   Q1 = df['hsc_p'].quantile(0.25)
   ```
   This line computes the 25th percentile (also called the first quartile, Q1) of the hsc_p column. Q1 represents the value below which 25% of the data points lie.
2. **Calculate Q3 (Third Quartile)**:
   ```
   Q3 = df['hsc_p'].quantile(0.75)
   ```
   This calculates the 75th percentile (third quartile, Q3), representing the value below which 75% of the data points lie.
3. **Calculate the IQR (Interquartile Range)**:
   ```
   IQR = Q3 - Q1
   ```
   The interquartile range is the difference between the third quartile (Q3) and the first quartile (Q1). The IQR represents the range in which the middle 50% of the data lies and is used to define outliers.
4. **Define the Filter for Non-Outliers**:
   ```
   filter = (df['hsc_p'] >= Q1 - 1.5 * IQR) & (df['hsc_p'] <= Q3 + 1.5 * IQR)
   ```
   This line defines a filter for detecting outliers. Any data point that lies within 1.5 times the IQR below Q1 or above Q3 is considered non-outlier. Points that fall outside this range are considered outliers. The filter keeps only those data points that fall within this range.
5. **Apply the Filter to Remove Outliers**:
   ```
   df_filtered = df.loc[filter]
   ```
   This applies the filter to the original DataFrame (df), creating a new DataFrame (df_filtered) that contains only the non-outlier data points from the hsc_p column.

   Explanation:

   To detect and remove outliers in the hsc_p column, we employed the interquartile range (IQR) method. First, we calculated the first quartile (Q1) and the third quartile (Q3), representing the 25th and 75th percentiles of the data, respectively. The IQR, which measures the spread of the middle 50% of the data, was calculated as the difference between Q3 and Q1. We then defined a range that extends 1.5 times the IQR beyond Q1 and Q3 to identify outliers. Data points falling outside this range were considered outliers and were removed, resulting in a filtered dataset free from extreme values.

.

3. Correlation Analysis

To understand the relationships between different variables, we calculated the correlation coefficients using the Pearson method. The correlation matrix quantifies how strongly pairs of features are related.

Correlation Matrix Interpretation

The correlation matrix revealed the following insights:

- **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.
- **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.
- **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.

Visualization: Heatmap

A heatmap visualization of the correlation matrix clearly displayed these relationships, with shades indicating the strength and direction of the correlations. The strong positive correlations stood out prominently, guiding our focus toward the most influential educational metrics.

4. Visualizations

Several visualizations were created to further illustrate the relationships within the data:

A. Box Plot

The box plot for the ssc_p variable against the placement status effectively illustrated the distribution of secondary education percentages among placed and non-placed students. It highlighted the presence of outliers and indicated that placed students tended to have higher secondary education percentages compared to those who were not placed.

B. Scatter Plot

A scatter plot of degree_p versus salary, colored by placement status, provided visual insights into how undergraduate performance correlated with salary offers. It demonstrated that placed students received higher salary offers, reinforcing the importance of strong academic performance.

5. Model Selection and Evaluation

Various machine learning models were employed to predict placement status based on the identified features. The models included Logistic Regression, Decision Tree Classifier, Random Forest Classifier, and a Voting Classifier.

Confusion Matrix

The confusion matrix provided a detailed breakdown of the model's predictions compared to the actual outcomes.

Confusion Matrix Interpretation

- **True Positives (TP)**: The number of students correctly predicted to be placed.
- **True Negatives (TN)**: The number of students correctly predicted to not be placed.
- **False Positives (FP)**: The number of students incorrectly predicted to be placed.

- **False Negatives (FN)**: The number of students incorrectly predicted to not be placed.

Visualization

The confusion matrix visualization revealed how well the model performed in distinguishing between placed and not placed students. High values in the true positive and true negative cells indicate that the model effectively classified students, while lower values in the false positive and false negative cells suggest areas for improvement.

Evaluation Metrics

The models were evaluated based on several key metrics, including accuracy, precision, recall, and F1 score.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	80.95%	90.00%	84.38%	87.10%
Random Forest Classifier	83.33%	90.32%	87.50%	88.89%
Decision Tree Classifier	76.19%	89.29%	78.13%	83.33%
Voting Classifier	83.33%	87.88%	90.63%	89.23%

Model Accuracy

- **Random Forest Classifier**: Achieved the highest accuracy at **83.33%**, with a precision of **90.32%**, indicating its effectiveness in predicting placement status.
- **Voting Classifier**: Displayed similar accuracy (**83.33%**) but showed a stronger recall (**90.63%**), suggesting it was more effective at identifying students who were placed.
- **Logistic Regression**: Maintained a balanced performance with an accuracy of **80.95%**, indicating it is a reliable model for this type of analysis.
- **Decision Tree Classifier**: Exhibited the lowest accuracy at **76.19%**, suggesting it may be less suitable for this specific prediction task.

7. Summary of Findings

The analysis identified strong correlations between academic performance and placement outcomes. Specifically, higher percentages in secondary, higher secondary, and undergraduate education significantly influenced placement success.

Each model's effectiveness is evaluated using the following metrics:

1. **Accuracy**: This indicates how often the model's predictions are correct across all predictions (both positive and negative).
   - Random Forest and Voting Classifier have the highest accuracy (0.8333), meaning they make correct predictions 83.33% of the time.
   - Logistic Regression follows closely with an accuracy of 80.95%, while Decision Tree has the lowest at 76.19%.
2. **Precision**: Precision measures the proportion of true positive predictions out of all positive predictions (how many of the positive predictions were actually correct).
   - Random Forest and Logistic Regression have the highest precision (0.9032 and 0.9000, respectively), indicating these models are very good at minimizing false positives.
   - Decision Tree has the lowest precision (0.8929), but it's still relatively high.
3. **Recall**: Recall is the ability of the model to correctly identify all actual positive cases (true positives out of all actual positives).
   - Voting Classifier has the highest recall (0.9063), meaning it successfully captures the most actual positive cases.
   - Logistic Regression has the lowest recall (0.8438), meaning it may miss more positive cases compared to other models.
4. **F1 Score**: This is the harmonic mean of precision and recall. It balances both metrics, especially useful when there is an uneven class distribution.
   - The Voting Classifier has the highest F1 score (0.8923), indicating it strikes the best balance between precision and recall.

  - Decision Tree has the lowest F1 score (0.8333), reflecting that its overall performance isn't as strong as the other models.

Overall insights:

- **Voting Classifier** and **Random Forest** perform the best across all metrics, with Voting Classifier having the highest recall and F1 score, and Random Forest being highly precise.
- **Logistic Regression** performs well in terms of precision but has a slightly lower recall.
- **Decision Tree** underperforms compared to the other models across all metrics, with notably lower accuracy and recall.

If you're looking for the best balance of performance, **Voting Classifier** seems to be the most well-rounded option.

8. Conclusion and Recommendations

- **Focus on Education**: Institutions should prioritize enhancing students' performance in secondary and higher secondary education to improve placement rates.
- **Support Resources**: Providing resources for students to excel in entrance tests and undergraduate degrees can significantly impact their employability.
- **Continuous Monitoring**: Regular analysis of placement data will allow institutions to adapt their strategies effectively and enhance student outcomes.

By leveraging the insights from this analysis, educational institutions can develop targeted interventions to support student success in the campus recruitment process.