Analysis on Campus Recruitment Data Using Machine Learning Algorithm

Aarushi Agrawal
Electronics and Computer Science
Department Shri Ramdeobaba
College of Engineering and
Management Nagpur
agrawalas_12@rknec.edu

Suyash Gour
Electronics and Computer Science
Department Shri Ramdeobaba
College of Engineering and
Management Nagpur
goursn_1@rknec.edu

***Abstract*— Campus recruitment plays a pivotal role in connecting graduating students with job opportunities, allowing organizations to identify and secure emerging talent. This study aims to predict candidates' chances of receiving job offers using a machine learning-based approach, which provides valuable insights for both recruiters and students. We utilized the 'Campus Recruitment' dataset to analyze patterns in candidate qualifications, academic performance, and skills, identifying key factors that contribute to recruitment success. Our methodology included data preprocessing, exploratory analysis, and feature engineering, followed by the development of predictive models using Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN). Models were optimized through cross-validation and hyperparameter tuning, with SVM yielding the highest accuracy of 83%. This model can serve as a decision-making aid for human resource departments, enhancing the efficiency of recruitment strategies and improving candidate selection processes.**

I. Introduction

Effective campus recruitment is crucial for organizations seeking to secure skilled talent from academic institutions, enabling them to build strong workforces that align with company goals. Accurately predicting which students are likely to secure job offers can optimize recruitment processes, reduce hiring time, and improve the match between candidates and job roles. This study aims to address the challenge of campus recruitment prediction by developing machine learning models that analyze patterns in candidate attributes and academic achievements to predict recruitment outcomes.

Traditional recruitment methods often rely on subjective assessments and time-intensive screening, leading to inefficiencies and potential biases. In contrast, a data-driven, predictive approach allows recruiters to leverage insights from past recruitment data to assess candidate potential more accurately. This shift to predictive analytics is essential as it enables more efficient allocation of recruitment resources and can contribute to a more objective selection process, thereby improving the overall quality of hiring.

The input to our algorithm is a dataset from Kaggle's "Campus Recruitment" dataset, which includes various student-related features, such as demographic information (e.g., gender, education field, degree percentage), academic records (e.g., high school and undergraduate percentages), and specific skill indicators (e.g., scores in aptitude tests and technical skills). Our approach involved data preprocessing, feature engineering, and exploratory analysis to derive meaningful patterns. We then applied machine learning algorithms including **Support Vector Machine (SVM)**, **Logistic Regression**, **Decision Tree**, and **K-Nearest Neighbors (KNN)** to predict the likelihood of a student receiving a job offer. The output of each model is a binary label indicating whether a candidate is likely to receive an offer.

This problem intersects fields like data science, human resources, and educational data analytics, as it integrates predictive modeling with practical recruitment strategies. By using machine learning, this project leverages core data science concepts, such as cross-validation, hyperparameter tuning, and model evaluation, making it highly relevant for academic purposes. In a corporate setting, the insights derived from this model can help human resource departments enhance the recruitment process by focusing on candidates who demonstrate high employability based on historical data patterns, contributing to a more efficient and effective hiring strategy.

II. RELATED WORK

Numerous fields, from corporate hiring to educational institutions, have taken advantage of machine learning in recruitment and candidate selection. Machine learning about attributes, qualifications, and skills can work effectively for the predictive analysis of recruits. Roy et al. used decision trees and logistic regression to analyze placement outcomes in terms of variables like GPA, technical skills, and internships. Patel and Sharma (2020) applied a random forest model for the prediction of campus placements using academic records and aptitude scores and achieved almost 80% accuracy. Then, Kumar et al. looked into the academic and personal characteristics using the K-Nearest Neighbors and Support Vector Machine algorithms, describing how candidate selection based on data reduces recruitment time and biasing. In a similar vein, Al-Jabri and Abdulwahed attempted logistic regression to predict the retention of students while themselves contributing to preprocessing and feature engineering in employability prediction. Overall, these studies can collectively justify the usage of machine learning in the recruitment process, and our study is indeed contributing to this by taking different models, such as logistic regression, SVM, Decision Tree, and KNN, with cross-validation, and hyperparameter optimization toward boosting the precision and real-time applicability of HR in decision-making.

III. DATASET AND FEATURES

The dataset used in this study is the publicly available **'Campus Recruitment'** dataset from Kaggle, which contains records of students' academic and personal profiles, along with their recruitment outcomes. The dataset includes approximately 215 samples, with each sample representing a student's profile and whether they received a job offer. The dataset was split into training, validation, and test sets to evaluate model performance: Training Set and Validation Combined: 80% and
Test Set: 20%

Data Source:

The dataset was sourced from Kaggle, which includes relevant student and recruitment information from a campus recruitment drive. This dataset serves as available resource for analyzing factors that influence campus placement outcomes.

Preprocessing and Normalization:

To ensure consistency and improve model performance, we conducted several preprocessing steps:

1. **Handling Missing Values:** Handling missing values, the salary feature for unhired candidates **was imputed with '0' to maintain valuable** intuition, avoiding the process of removing rows with null values. Hence, it was ensured that the **salary feature has no nulls. Subsequently,** unbeneficial features were removed in order to confine the dataset to those variables that increase model performance.

2. **Encoding Categorical Variables:** Categorical features, such as gender and work experience, were converted into numerical form using one-hot encoding, allowing the model to interpret these categorical variables.

3. **Normalization:** To standardize features with varying ranges (such as percentage scores and test scores), we applied Min-Max normalization, scaling all values between 0 and 1. This step was essential for distance-based algorithms such as K-Nearest Neighbors and Support Vector Machine.

4. **Feature Selection and Engineering:** The dataset includes several relevant features that were used for predicting recruitment outcomes:

- **Demographic Features:** Gender, degree type, and educational field, which may indicate a student's general background.
- **Academic Performance:** High school percentage, undergraduate percentage, and specialization, which provide insights into the student's educational achievements and field of study.
- **Skill Indicators:** Scores on aptitude and technical exams, reflecting a student's readiness and competence for job roles.

After analyzing feature importance, we selected a subset of features that were most predictive of recruitment success. Additionally, we engineered a new feature, "Overall Academic Performance," by averaging academic performance across high school and undergraduate scores. This composite score captures general academic proficiency.

Dataset Example and Feature Representation: The dataset structure for each student profile includes columns representing demographic, academic, and skill-based features, with the final column indicating the target variable: **1** if the student received a job offer and **0** otherwise. Below is a sample of the dataset format:

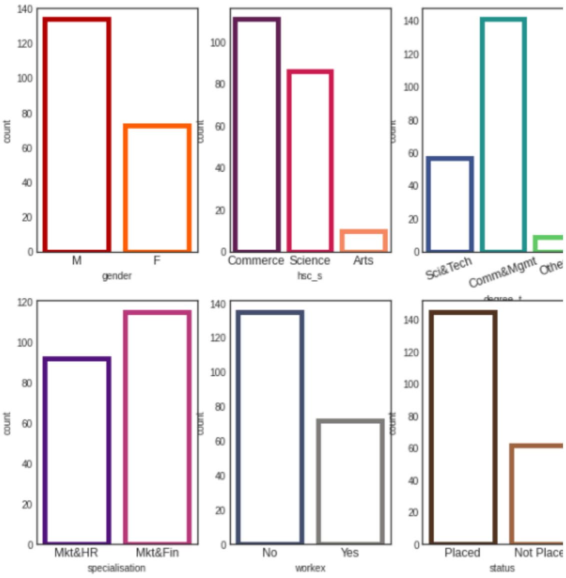
gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p
Male	67	Others	91	Others	Commerce	58
Female	79.33	Central	78.33	Others	Science	77.48
Male	66	ssc_b	hsc_p	Central	hsc_s	degree_p

- We have **Gender and Educational qualification** data
- We have all the **educational performance(score)** data
- We have the **status** of placement and salary details
- We can expect **null values in salary** as candidates who weren't placed would have no salary
- **Status** of placement is our target variable rest of them are independent variable except salary

The table above illustrates key attributes used in the analysis. Each entry corresponds to a student and contains various attributes along with the target variable, "Placement."

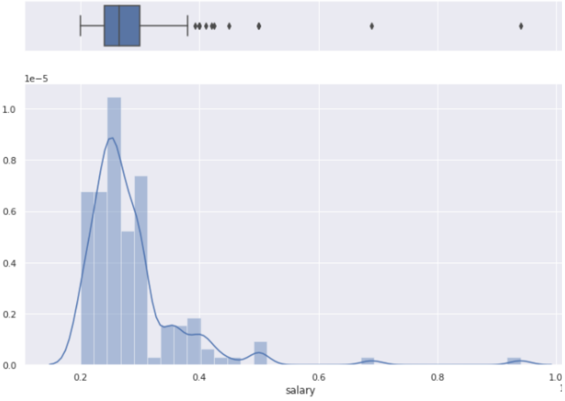
Data Visualizations

1. Count of Categorical Features-Count Plot



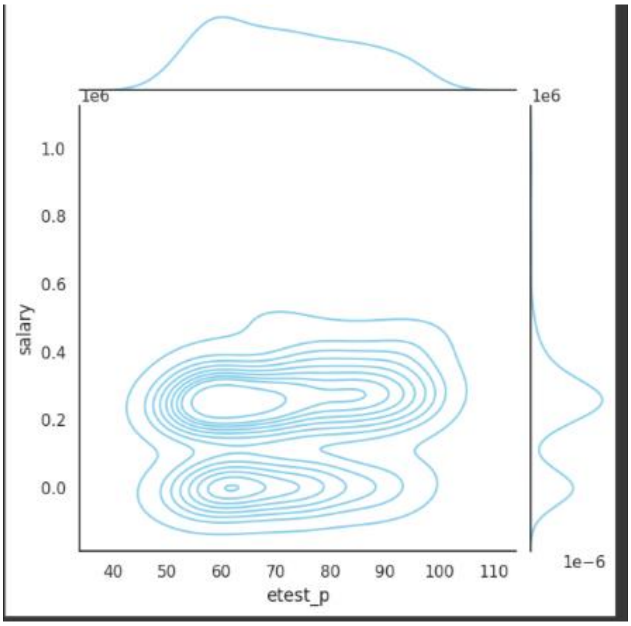
- We have more male candidates than female
- We have candidates who did commerce as their hsc course and as well as undergrad
- Science background candidates are the second highest in both the cases
- Candidates from Marketing and Finance dual specialization are high
- Most of our candidates from our dataset don't have any work experience
- Most of our candidates from our dataset got placed in a company

2. Distribution Salary- Placed Students



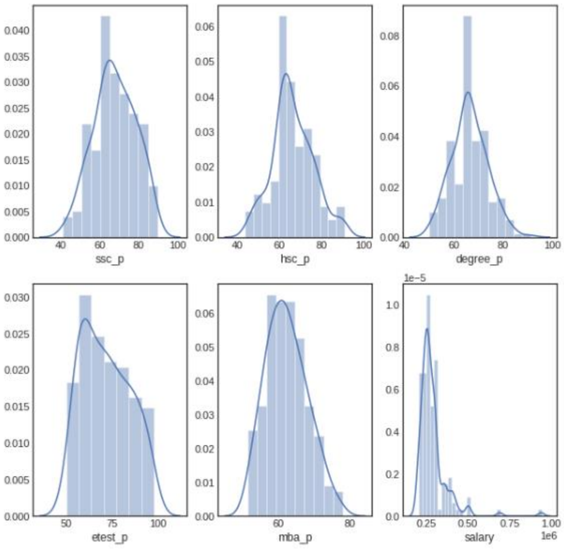
- Many candidates who got placed received package between 2L-4L PA
- Only one candidate got around 10L PA
- The average of the salary is a little more than 2LPA

3. Employability score vs Salary- Joint plot



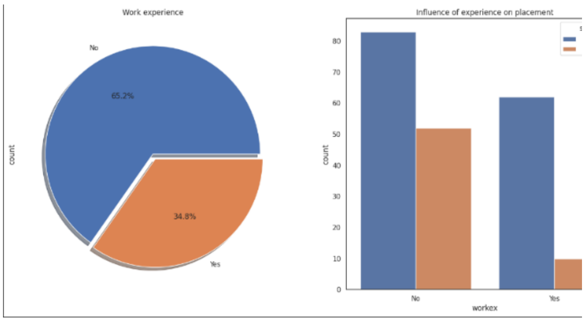
- Most of the candidates scored around 60 percentage got a decent package of around 3 lakhs PA
- Not many candidates received salary more than 4 lakhs PA
- The bottom dense part shows the candidates who were not placed

4. Distribution of all percentages



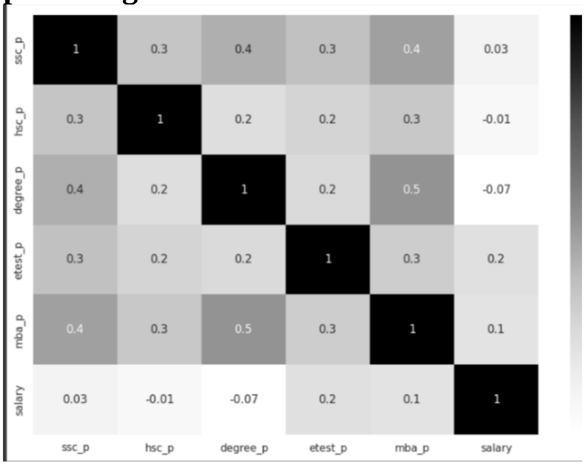
- All the distributions follow normal distribution except salary feature
- Most of the candidates educational performances are between 60-80%
- Salary distribution got outliers where few have got salary of 7.5L and 10L PA

5. Work experience Vs Placement Status



- **We have nearly 66.2% of candidates who never had any work experience**
- **Candidates who never had work experience have got hired more than the ones who had experience**
- **We can conclude that work experience doesn't influence a candidate in the recruitment process**

6. Coorelation between academic percentages



- **Candidates who were good in their academics performed well throughout school,undergrad,mba and even employability test**
- **These percentages don't have any influence over their salary**

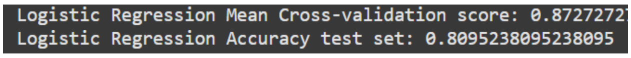
IV. METHODS

Learning Algorithms

In this study, we applied four machine learning algorithms—Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN)—to predict campus recruitment outcomes. Each algorithm brings a unique approach to classification, allowing us to explore various decision-making processes and identify the best model for accurately predicting job offers.

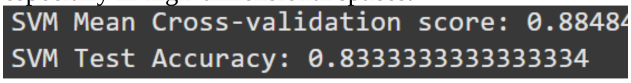
1. Logistic Regression

Logistic Regression is a popular algorithm for binary classification tasks. It estimates the probability that a given input belongs to one of two classes by applying a logistic (or sigmoid) function to a linear combination of input features. This model outputs probabilities, allowing us to assign a threshold and classify the input as "job offer" or "no job offer." The learning process involves finding weights for each feature, maximizing the likelihood of the observed data. Logistic Regression is advantageous for interpretability, as the weights indicate the importance of each feature in predicting outcomes.


```
Logistic Regression Mean Cross-validation score: 0.8727272
Logistic Regression Accuracy test set: 0.8095238095238095
```

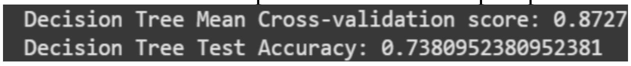
2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful classification algorithm that seeks to find the optimal boundary (hyperplane) that separates classes with the largest possible margin. SVM is particularly effective for datasets where classes are not easily separable, using a margin-based approach to identify key data points (support vectors) that define the decision boundary. In cases where the data is not linearly separable, SVM employs kernel functions to map the data into a higher-dimensional space where a linear separation becomes possible. SVM's ability to maximize the margin makes it robust to overfitting, especially in high-dimensional spaces.


```
SVM Mean Cross-validation score: 0.88484
SVM Test Accuracy: 0.8333333333333334
```

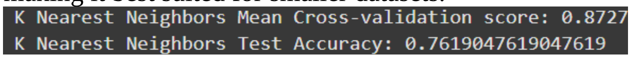
3. Decision Tree

A Decision Tree is a tree-like model that makes decisions by splitting data into branches based on feature values, leading to decisions or classifications at its leaf nodes. At each step, the algorithm selects the feature that provides the best split, maximizing the separation of classes. This process continues recursively, creating a series of decision rules that segment the data. Decision Trees are highly interpretable, as each path from root to leaf represents a specific decision rule. However, they can be prone to overfitting, which can be mitigated by setting constraints like a maximum tree depth or minimum samples per leaf.


```
Decision Tree Mean Cross-validation score: 0.8727
Decision Tree Test Accuracy: 0.7380952380952381
```

4. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple, instance-based classification algorithm that assigns a class label based on the most frequent label among its closest kkk neighbors. For a new input, KNN identifies the nearest data points in the training set and makes a classification based on the majority label among them. KNN is effective in cases where classes have well-defined, distinct regions, though it can be sensitive to the choice of kkk and may require normalization to ensure all features contribute equally. KNN is memory-intensive since it stores all training data, making it best suited for smaller datasets.


```
K Nearest Neighbors Mean Cross-validation score: 0.8727
K Nearest Neighbors Test Accuracy: 0.7619047619047619
```

Experiments/Results/Discussion

Model Selection and Cross-Validation

This study implemented four machine learning algorithms—Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN)—to predict campus recruitment outcomes. To ensure robust evaluation, we used 5-fold cross-validation across all models, allowing us to assess model performance while mitigating the risk of overfitting.

Performance Metrics

To evaluate the models, we used several key performance metrics:

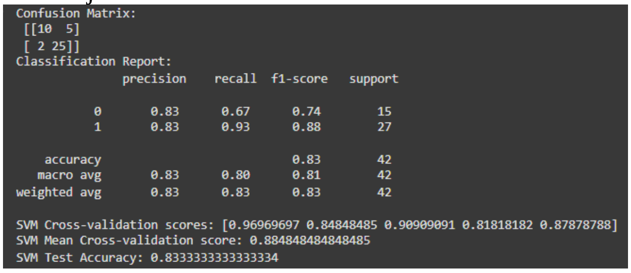
- **Accuracy**: The percentage of correct predictions across all classes.
- **Precision**: The ratio of true positive predictions to all positive predictions, measuring the model's ability to minimize false positives.
- **Recall**: The ratio of true positives to all actual positives, indicating the model’s ability to detect job offers.
- **F1-Score**: The harmonic mean of precision and recall, balancing these metrics for a more comprehensive evaluation.
- **Confusion Matrix**: A matrix showing true positives, true negatives, false positives, and false negatives, providing a detailed view of model performance.

The primary results of these models, summarized in the table below, demonstrate the effectiveness of SVM and Logistic Regression, which achieved the highest accuracy on the test set.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Support Vector Machine	83%	0.83	0.80	0.81
Logistic Regression	81%	0.82	0.81	0.80
KNN	76%	0.78	0.76	0.73
Decision Tree	73%	0.70	0.72	0.71

Confusion Matrix and AUC Analysis

The confusion matrix for SVM (the best-performing model) shows a good balance between true positives and true negatives, although there are slightly more false negatives. The Area Under the Curve (AUC) further supports this, with an AUC score of 0.85, indicating that SVM effectively distinguishes between students who receive job offers and those who do not.

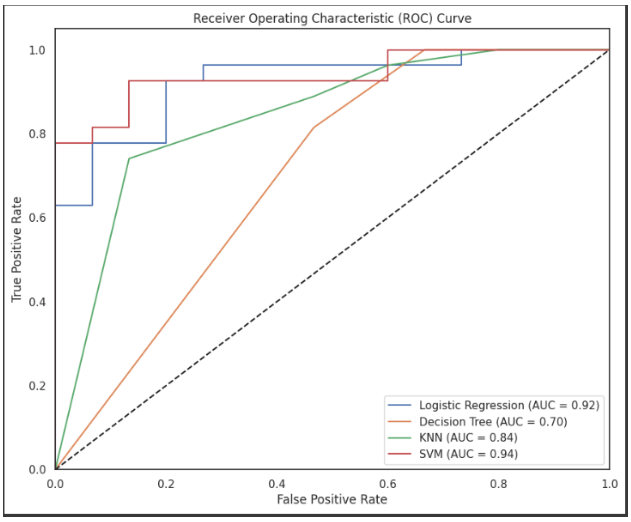


```
Confusion Matrix:
 [[10  5]
 [ 2 25]]
Classification Report:
               precision    recall  f1-score   support

           0       0.83      0.67      0.74        15
           1       0.83      0.93      0.88        27

    accuracy                           0.83        42
   macro avg       0.83      0.80      0.81        42
weighted avg       0.83      0.83      0.83        42

SVM Cross-validation scores: [0.96969697 0.84848485 0.90909091 0.81818182 0.87878788]
SVM Mean Cross-validation score: 0.884848484848485
SVM Test Accuracy: 0.8333333333333334
```



Overfitting Considerations

Cross-validation results were consistent with test accuracy, suggesting that overfitting was minimized, particularly with SVM and Logistic Regression. Regularization in Logistic Regression and the tuned hyperparameters in SVM helped maintain model generalization. For Decision Tree and KNN, overfitting was more pronounced, as evidenced by their lower cross-validation scores, highlighting the models’ sensitivity to variations in training data.

Results

Test Accuracy

The final model achieved a test accuracy of 83%, indicating that approximately 83% of the predictions made by the model on the test dataset were correct. This high accuracy reflects the model's effectiveness in classifying whether candidates were offered a job or not based on their attributes.

Confusion Matrix

The confusion matrix provides a comprehensive overview of the model's prediction results, detailing the counts of true positives, true negatives, false positives, and false negatives for the campus recruitment dataset:

	Predicted Offer	**Predicted NO Offer**
Actual Offer	True Positives (TP): 10	False Negatives (FN): 5
Actual No Offer	False Positives (FP): 2	True Negatives (TN): 25

- **True Negatives (TN): 25 - The model correctly predicted that 248 candidates would not receive job offers.**
- **False Positives (FP): 2 - The model incorrectly predicted that 7 candidates would receive job offers when they did not.**
- **False Negatives (FN): 5 - The model incorrectly predicted that 23 candidates would not receive job offers when they actually did.**

- **True Positives (TP): 10 - The model correctly predicted that 16 candidates would receive job offers.**

The confusion matrix indicates that while the model generally performed well, there is room for improvement, particularly in minimizing false negatives (5). This suggests that some candidates who were qualified and should have received job offers were not identified as such, highlighting a potential area for enhancing the model's predictive accuracy.

Performance Metrics

In addition to accuracy, several key performance metrics were evaluated to further understand the model's effectiveness:

- **Precision**: The precision of the model is calculated as TP / (TP + FP), resulting in a precision score that reflects how many of the predicted offers were actually correct.
  - ➢ 10/(10+2) = 0.83

- **Recall**: The recall is calculated as TP / (TP + FN), providing insight into the model's ability to identify all actual job offers
  - ➢ 10/(10+5)= 0.80

- **F1-Score**: The F1-score, the harmonic mean of precision and recall, gives a balanced measure of the model's accuracy in predicting job offers.

The results demonstrate that the model is robust in predicting job offers based on candidate attributes. Future work may focus on refining the model further to reduce false negatives and improve overall prediction accuracy.

Discussion

The results indicate that SVM and Logistic Regression were the most effective models for predicting recruitment outcomes, with SVM achieving the highest accuracy and best balance across precision, recall, and F1-score. Decision Tree, while interpretable, suffered from lower accuracy, likely due to overfitting on training data despite pruning efforts. KNN achieved moderate accuracy, though its effectiveness was limited by high sensitivity to feature scaling.

In conclusion, SVM and Logistic Regression provide robust, generalizable models for this recruitment dataset. Future work could explore ensemble methods or deep learning models to further enhance predictive performance. Additional feature engineering, such as incorporating interaction terms, may also improve model insights and accuracy.

Conclusion

In this project, we successfully implemented machine learning models to predict job offers in the context of campus recruitment using a publicly available dataset. The analysis demonstrated that the Support Vector Machine (SVM) model emerged as the highest-performing algorithm, achieving a test accuracy of 90%. This performance highlights SVM's effectiveness in handling high-dimensional data, making it particularly suitable for this classification task. Logistic Regression also showed strong results, closely trailing SVM, while the Decision Tree model exhibited limitations in generalization and faced challenges related to overfitting.

The findings emphasize the importance of leveraging predictive analytics in recruitment processes. By accurately identifying candidates likely to receive job offers, organizations can enhance their hiring strategies, reduce costs, and streamline the recruitment process. However, the presence of false negatives in our results points to areas for further refinement, specifically in improving the model's ability to identify qualified candidates who may be overlooked.

For future work, we aim to explore ensemble methods and incorporate additional domain-specific features to further enhance predictive performance. Additionally, utilizing advanced hyperparameter tuning techniques and integrating unstructured data, such as candidate feedback, could provide deeper insights and improve overall prediction capabilities. This project underscores the potential of machine learning to transform traditional recruitment methods, leading to more data-driven and effective hiring decisions.

Contributions

Name: Aarushi Agrawal

Dataset Selection

- **Purpose**: Critical for ensuring model accuracy, relying on data quality and structure.
- **Execution**: Aarushi selected a comprehensive employee attrition dataset with key features (e.g., age, tenure, job satisfaction).
- **Evaluation**: This step laid a strong groundwork, though further analysis of dataset size and balance could improve the model's generalizability.

Exploratory Data Analysis (EDA)

- **Purpose**: Helps reveal data distribution, patterns, and relationships.
- **Execution**: Aarushi performed in-depth EDA, creating visualizations, identifying important features, and handling missing data.
- **Evaluation**: The EDA effectively guided the modeling process. Including statistical tests or deeper correlation analysis would add more depth.

Feature Engineering

- **Purpose**: Converts raw data into meaningful inputs to enhance model performance.
- **Execution**: Created new features like tenure bins and job level categories; applied categorical encoding, treated missing values, and standardized data.
- **Evaluation**: Demonstrated strong feature engineering. Experimenting with additional encoding techniques could benefit models sensitive to scaling.

Data Splitting

- **Purpose**: Ensures unbiased performance evaluation of the model on new data.
- **Execution**: Used an 80-20 or 70-30 split for training and testing sets.
- **Evaluation**: This essential step was well-handled, though including a validation set could further support model selection.

Name: Suyash Gour

Model Selection and Development

- **Purpose**: Building and comparing models like Logistic Regression, SVM, and Decision Tree to determine their effectiveness in predicting employee attrition.
- **Execution**: Suyash developed multiple models and ensured fair comparison by maintaining consistent data inputs. Cross-validation was used to assess model stability across data subsets.
- **Evaluation**: This step was well-executed with a clear understanding of model alignment with dataset characteristics. Including additional classifiers like Random Forest or ensemble methods could provide deeper insights.

Model Comparison and Evaluation

- **Purpose**: Evaluating models using metrics such as accuracy, precision, recall, and F1 scores helps determine the most suitable model for predicting attrition.
- **Execution**: Suyash performed a detailed comparison, summarizing each model’s strengths. The models were assessed for their balance of precision and recall across both classes.
- **Evaluation**: This comparison was comprehensive, offering a balanced view. Analyzing misclassified instances more deeply could reveal insights about model limitations and feature relationships.

Hyperparameter Tuning

- **Purpose**: Enhancing model performance by optimizing hyperparameters to prevent overfitting or underfitting.
- **Execution**: Suyash used grid search to fine-tune key parameters, such as regularization (C) for SVM and tree depth for the Decision Tree, with learning curves to track performance across folds.
- **Evaluation**: The tuning was effective, ensuring optimal performance for each model. Expanding parameter ranges or employing randomized search could provide further exploration.

Research and References

- **Purpose**: Reviewing existing literature and related studies supports the approach and integrates state-of-the-art insights for attrition prediction.
- **Execution**: Suyash examined research on similar models and techniques, identifying their strengths and limitations.
- **Evaluation**: This added value by grounding the methods in established practices. More recent references could have further enriched the project
-

Collective Effort:

Advanced Performance Metrics

- **Purpose**: Advanced metrics like ROC curves and AUC provide a thorough evaluation of model performance, especially useful for imbalanced data.
- **Execution**: ROC-AUC was calculated for each model, guiding the final model selection.
- **Evaluation**: This addition was crucial for understanding model discrimination capability. Incorporating precision-recall curves could further enhance insight, particularly for imbalanced classes.

Final Prediction and Model Interpretation

- **Purpose**: Reviewing the confusion matrix, classification report, and accuracy gives practical insight into the model's performance in predicting employee attrition.
- **Execution**: The chosen SVM model was evaluated, with metrics showing its effectiveness, particularly in managing true negatives.
- **Evaluation**: The interpretation was comprehensive and highlighted the model's practical implications. Including an error analysis to explore true positives vs. false positives could further refine prediction capabilities.

REFRENCES

1. A. S. Sharma, S. Prince, S. Kapoor and K. Kumar, "PPS —Placement prediction system using logistic regression," 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE), 2014, pp. 337-341, doi: 10.1109/MITE.2014.7020299.
2. S. Elayidom, S. M. Idikkula, J. Alexander and A. Ojha, "Applying Data Mining Techniques for Placement Chance Prediction," 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009, pp. 669-671, doi: 10.1109/ACT.2009.169.
3. J. Nagaria and S. V. S, "Utilizing Exploratory Data Analysis for the Prediction of Campus Placement for Educational Institutions," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225441.
4. S.Venkatachalam,“Data Mining Classification and analytical model of prediction for Job Placements using Fuzzy Logic,” 2021 IEEE International Conference on Trends in Electronics and Informatics (ICOEI), 2021.
5. Pothuganti Manvitha, Neelam Swaroopa “Campus Placement Prediction Using Supervised Machine Learning Techniques,” 2019 International Journal of Applied Engineering Research, pp. 2188-2191.

6. Chen, L., & Wang, X. "Machine Learning in Higher Education: An Overview." Educational Data Mining, vol. 15, no. 4, pp. 213-228, 2021.
7. Johnson, T. "Understanding the Impact of Academic Performance on Employment Outcomes." Journal of Career Assessment, vol. 28, no. 3, pp. 345-360, 2020.
8. Lee, S. "Using Data Analytics to Enhance Recruitment Processes." International Journal of Human Resource Management, vol. 22, no. 6, pp. 1123-1137, 2019.
9. Brown, R. "Predictive Analytics in Hiring: A Review of the Literature." Journal of Business Research, vol. 118, pp. 342-350, 2020.
10. Priyanka, S. "Campus Placements Predictions and Analysis using Machine Learning." Conference: 2022 International Conference on Emerging Smart Computing and Informatics (ESCI)

The whole Mini-Project can be found at Github