Job Placement Dataset

Authors: Jason Zhang, Qi Li, Divya Iyer, Ariana Lozner

Abstract

This research paper compares the performance of several well-known machine learning algorithms in predicting job placement outcomes based on candidate data such as education, work experience, sex, and academic performance. The algorithms considered are k-Nearest-Neighbors, Decision Trees, Logistic Regression, and Neural Networks. Results of the study demonstrate that with sufficient feature selection and tuning, all models perform well in predicting job placement outcomes, with logistic regression achieving the highest accuracy of 93%. The findings suggest that it can be effective for both job seekers and employers to leverage machine learning algorithms as a tool to improve job placement outcomes. In summary, the goal of this paper is to find the most effective machine learning model to predict a candidate's placement status and highlight the most important features that determine whether an individual gets selected for a position. These insights will help employers identify the most important attributes of a candidate, and allow candidates to identify areas to improve.

Background

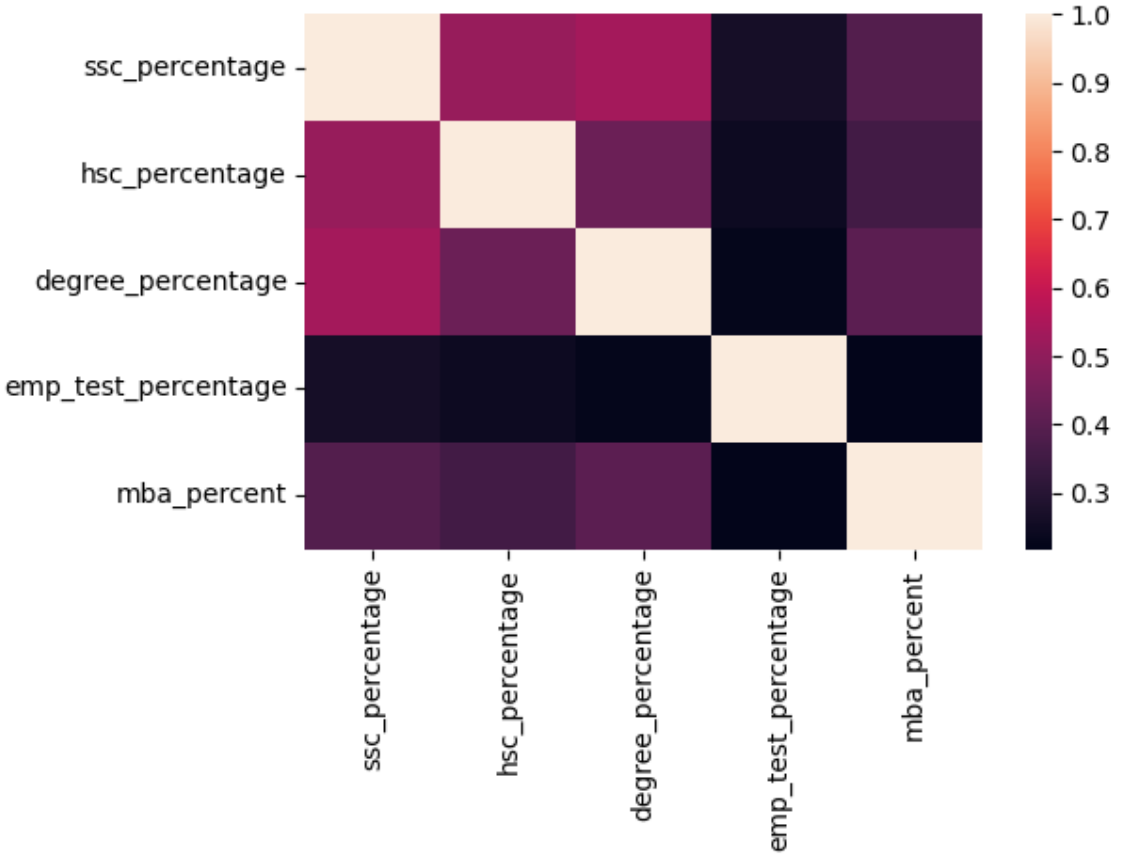
In the competitive job market of today, the recruitment process has become increasingly important for organizations worldwide to attract top talent to meet the demands of expanding economies. Traditional methods of recruitment and selection may not always be the most effective; job postings, referrals, and campus interviews are time-consuming and expensive. There exists a great opportunity to apply modern techniques of Data Science and Machine Learning to optimize this recruitment process and improve the quality of hires.

This dataset contains several attributes which describe a candidate's educational background and work experience. The goal of the project is to utilize these attributes to predict whether a candidate will receive a job offer. There are 12 features, compromising categorical and continuous variables, and 1 target variable which is a candidate's placement status. This dataset is significant because it provides a basis for exploring various questions relating to job placement, including the driving effect that sex, academic performance, work experience, and other variables may have.

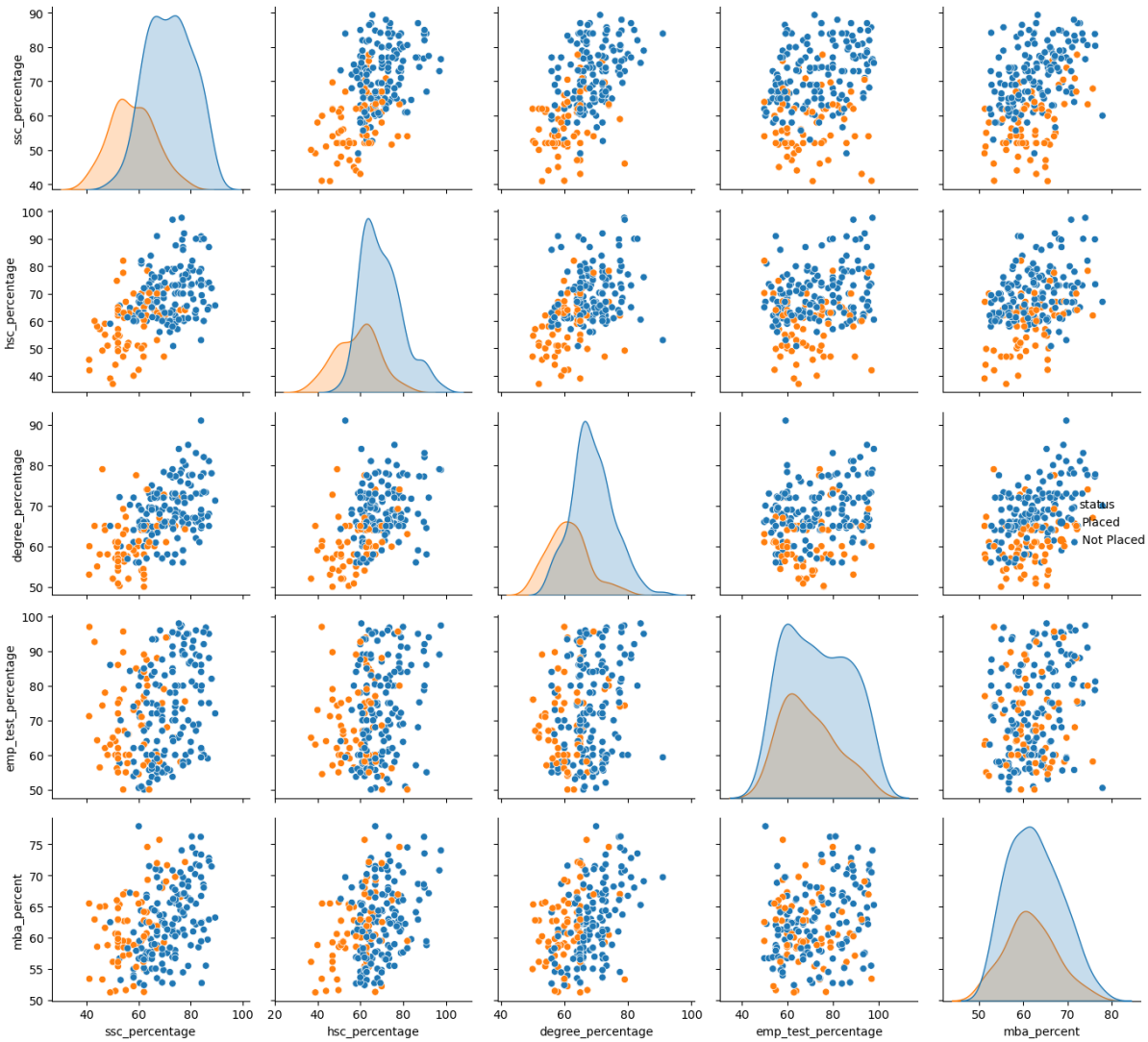
Because this dataset is relatively new (published in January 2023), there is not much prior research publicly available. However, there exist prior attempts to perform classification on this data, which utilized the Logistic Regression technique to predict whether candidates are placed. A highly rated notebook on Kaggle performed with an accuracy of 77.3% on the test set.

Data Analysis

The dataset contains a total of 215 records, with 12 features and 1 target. Of the 12 features, 7 are categorical variables (gender, ssc_board, hsc_board, hsc_subject, undergrad_degree, work_experience, and specialization) and 5 are continuous variables (ssc_percentage, hsc_percentage, degree_percentage, emp_test_percentage, and mba_percent). Taken together, the features provide a fairly comprehensive view on a candidate's academic performance through high school and undergraduate studies, graduate education, work experience, and job fit. Various techniques such as Confusion Matrices, Pairplots, and Histograms were employed to visualize the data. A summary of the findings are included below.

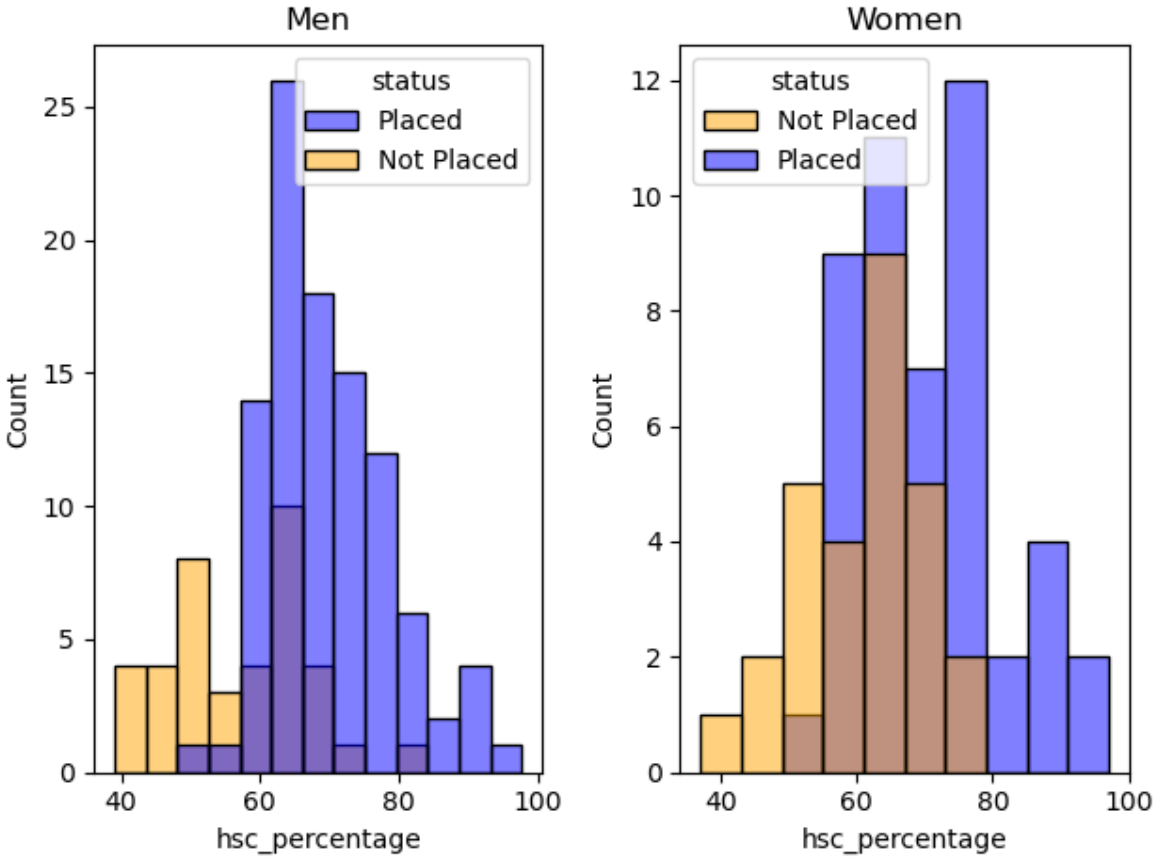


The heatmap shows that in the data, academic performance features are fairly correlated (ssc_percentage, hsc_percentage, degree_percentage) while the aptitude test performance (emp_test_percentage) is not highly correlated with any other feature.



Key: blue = Placed, orange = Not Placed

The pair plots reveal that in the data, candidates who are successfully placed tend to have better performance in academic measures as well as the aptitude test. However, the one feature that appears least useful in separating those who are Placed is “mba_percent”.



The two histograms demonstrate that in the data, a greater proportion of women who score average/higher than average in high school exams are “Not Placed” when compared to men.

Methods

Feature Selection

For kNN, the Select K Best algorithm was employed for feature selection. This selected 10 best features to use for the final model (gender, ssc_percentage, ssc_board, hsc_percentage, degree_percentage, work_experience, emp_test_percentage, specialisation, hsc_subject_Arts, undergrad_degree_Others).

The recursive feature elimination CV method was used for both the Decision Tree and Logistic Regression models. The corresponding model was used as the base estimator in each recursive feature elimination task in order to select the features of highest contribution unique to each respective model.

Feature selection was not combined with the Neural Network because the dataset is already on the smaller side, so it was decided that giving the model all 12 features would be most beneficial given this context.

Models Employed

Because there is no single model that will perform better than the rest in every single use case, four algorithms were selected that have varying strengths and weaknesses. This makes it possible to observe which set of strengths and weaknesses are most appropriate for the specific dataset in this study. The advantages and disadvantages of each model are listed below.

- kNN
    - Pros:
        - Training time is fast since the algorithm simply stores the instances.
    - Cons:
        - Classification is slow as all attributes contribute to the distance calculations leading to irrelevant attributes having a large impact
- Decision Tree
    - Pros:
        - Less preprocessing required, handles missing values easily, places most important features at the root easily providing insight into the data
        - Resulting models are easily understandable especially for those without a technical background.
    - Cons:
        - Possibility of high complexity and overfitting if many branches are used
        - Training time is slow, small changes in data may lead to a large change in the resulting decision tree (high variance).
- Logistic Regression
    - Pros:
        - Logistic Regression is well known for high performance in binary classification tasks
        - Logistic regression works well on smaller datasets and is very fast at classifying unknown records
    - Cons:
        - Cannot learn complex relationships because it constructs linear boundaries
        - Limited by the assumption of linearity between dependent and independent variables
- Neural Network
    - Pros:
        - NN's can learn complex patterns and underlying features that would otherwise be difficult or impossible to produce through manual feature engineering
        - NN's work well with nonlinear data and can generalize better than traditional algorithms
    - Cons:

- The dataset is on the smaller side, NN's generally require larger datasets than traditional ML algorithms to achieve high performance
- NN's are generally a black box where their decision-making process is not easily interpretable

Hyperparameter Tuning/Cross Validation Methods

For all models, grid-search CV methods were employed to find optimal hyperparameters. This strategy performed an exhaustive search of all possible combinations of input hyperparameters, while evaluating each candidate model on a k-fold Cross Validation to avoid overfitting and provide a more accurate measure of true model performance. For each model, the hyperparameters with the greatest influence on performance were tested over a wide range of values to select the best candidates. For example, the Decision Tree's "max_depth" parameter was searched over the range 5-30 and the Logistic Regression's "solvers" parameter was tested for "newton-cg", "lbfgs", and "liblinear".

Analysis

Each of the models (Knn, decision tree, logistic regression, and neural network) that were studied on this dataset all were split into a training and testing set. The test set was split off from and comprised 20% of the training set. Hyperparameter tuning and feature exclusion were employed by these models to get the most optimal accuracy.

For the k-Nearest Neighbors algorithm, the most optimal pairing of parameters is k=10 and distance_metric=Manhattan. The selected features were gender, ssc_percentage, ssc_board, hsc_percentage, degree_percentage, work_experience, emp_test_percentage', specialization, hsc_subject_Arts, and undergrad_degree_Others.
This produced an accuracy of approximately 86.14% on the training set. The validation accuracy resulting from 10 fold cross validation was approximately 90.2% and the mean accuracy of the validation set was approximately 87.69%. The accuracy for the final model with the selected features and parameters was 84%. The final metrics for candidates not places were 0.9 for precision, 0.6 for recall, and 0.72 for f1 score. For placed candidates the metrics were 0.82 for precision, 0.96 for recall, and 0.89 for f1 score. Therefore, the high recall for placed candidates and lower recall for candidates not placed indicates that this model is very good at detecting placed candidates but not as good as predicting candidates who have not been placed. Thus, Knn is able to successfully detect 96% of all positives (placed candidates). However, the precision is lower for placed candidates than candidates who have not been placed indicating a higher chance for false positives or identifying a candidate as having been placed when in reality they have not been placed.

For the decision tree algorithm, after running recursive feature elimination, the features deemed most important that were used in the final model were ssc_percentage, hsc_percentage, degree_percentage,emp_test_percentage, specialization, mba_percent, hsc_subject_Commerce,hsc_subject_Science,undergrad_degree_Comm&Mgmt,

undergrad_degree_otherst, undergrad_degree_Sci&Tech potentially indicating that degrees and test scores play a role in determining job placement. Grid search was used to determine hyperparameter tuning and based on this ccp_alpha=0.01, max_depth=10, max_features='auto', random_state=1024 ended up producing the best results. With feature selection and hyperparameter tuning the decision tree model was able to produce a final accuracy of 84%. The model performed substantially better than the original decision tree (without hyperparameter tuning and feature selection) model in correctly detecting candidates who were not placed with a recall for unplaced candidates going from 50% in the original model to 80%. The recall for successfully identifying all candidates who were placed was a little worse dropping down to 86% instead of 96% as in the original model. The final scores for unplaced candidates were 0.75 for precision, 0.80 for recall, and 0.77 for f-score. For placed candidates it was 0.89, for precision, 0.86 for recall, and 0.87 for f-score.

For logistic regression, after performing feature elimination, the top 5 most important features (ssc_percentage, hsc_percentage, degree_percentage, work_experience, mba_percent) were kept. Hyperparameter tuning was implemented to determine the best parameters which ended up having a C value =100, penalty=12, and solver=newton-cg. The logistic regression model with these hyperparameters and features had a high accuracy of 93%, greater than any of the other models implemented. It also had a precision score of 0.93, a recall of 0.96, and an f1-score of 0.95 for placed candidates. And a precision score of 0.93, a recall of 0.87, and an f1-score of 0.90 for unplaced candidates. This indicates a slightly higher accuracy for identifying all the placed candidates than detecting unplaced candidates.

For the neural network the best accuracy was achieved with batch size 10 and max epochs of 300 and using Adagrad and 32 neurons in the hidden layer. The final accuracy was 84.62%

In summary, all these models had an accuracy above 80% with logistic regression performing at the highest accuracy at 93%. It also had best precision, recall, and f1 score across all the other models tested. Overall, all models scored better at detecting placed candidates than unplaced candidates.

Model	Not placed Precision	Not placed recall	Not placed F1	Placed Precision	Placed Recall	Placed F1	Accuracy
KNN	0.90	0.60	0.72	0.82	0.96	0.89	0.84
Decision Tree	0.75	0.80	0.77	0.89	0.86	0.87	0.84
LogisticRegression	0.93	0.87	0.90	0.93	0.96	0.95	0.93
Neural Network							0.8462

Conclusions

In conclusion, this paper compared the performance of several machine learning algorithms in predicting job placement outcomes. The study found that logistic regression was the most performant model, achieving an accuracy of 93% in predicting job placement outcomes. Logistic regression is a popular algorithm for binary classification problems, such as job placement outcomes, because it uses a sigmoid function to effectively map the input features to the probability of job placement success. Additionally, logistic regression is a simple and interpretable algorithm, which makes it easy to understand and implement in practice.

While the other algorithms also performed well, logistic regression demonstrated better overall performance in terms of accuracy and computation time. The simplicity of the algorithm and its ability to handle categorical data also contributed to its superior performance.

The study's findings suggest that logistic regression can be a useful tool for recruiters to predict job placement outcomes based on candidate data. It can help organizations to make informed decisions and improve their hiring processes by identifying the most suitable candidates for specific job roles. Future research could explore the effectiveness of other machine learning algorithms or ensemble methods. Additionally, the study could be expanded to include additional factors that may influence job placement outcomes, such as personality traits, soft skills, and cultural fit.

Author Contributions

Jason Zhang

- Logistic Regression and Neural Network

Qi Li

- Decision Tree

Divya Iyer

- kNN

Ariana Lozner

- Research Paper

References

ahsan81. (2021). Job Placement Dataset. Kaggle. Retrieved April 10, 2023, from https://www.kaggle.com/datasets/ahsan81/job-placement-dataset

jjz17. (n.d.). Job-Placement. GitHub. Retrieved April 10, 2023, from https://github.com/jjz17/Job-Placement