EMPLOYEE PROMOTION PREDICTION MODEL INTRODUCTION TO MACHINE LEARNING AND MACHINE LEARNING 1

Nurbek Bektursyn
Petra Kralj Novak

Central European University
Quellenstrasse 51, 1100 Vienna, Austria
e-mail: Bektursyn_Nurbek@student.ceu.edu

ABSTRACT

The purpose of this research is to investigate the effectiveness of several machine learning models in forecasting employee promotions. The research applies models such as GradientBoostingClassifier, RandomForestClassifier, DecisionTreeClassifier, KNeighborsClassifier, and LogisticRegression on a large dataset that contains variables like age, department, education, awards won, and average training score. To ensure effectiveness, each model is thoroughly tuned using RandomizedSearchCV, with a special focus on AUROC values and F1 scores to evaluate discriminative power and model correctness.

The GradientBoostingClassifier is the most effective in prediction among all considered models. This discovery emphasizes the model's robustness in dealing with feature-rich and complicated datasets. In addition, the RandomForestClassifier is the second among the models, demonstrating the usefulness of ensemble approaches in predictive analytics.

This study helps advance the field of HR analytics by providing a data-driven method to improve the way decisions are made in employee promotions. The use of these models has the potential to change talent management by assuring objective and effective promotion processes.

1 INTRODUCTION

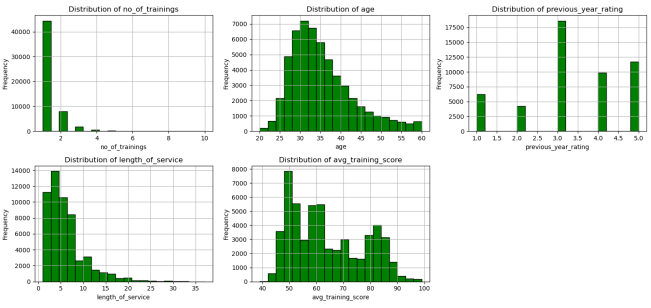
The research project addresses a fundamental issue in Human Resources (HR): predicting employee promotions. Being able to effectively estimate prospective employees for promotion is critical in the setting of corporate development and talent administration. This not only guarantees merit-based and fair promotion but also assists with workforce development and strategic planning. The goal of this project is to forecast promotion eligibility using machine learning methods that consider performance measures, historical data, and demographics while assessing the efficiency of different methods in the present scenario. Such tasks have traditionally been time ineffective and dependent on subjective evaluations, but current developments in HR analytics have evolved towards more advanced models, incorporating machine learning implementations. This project emphasizes training and tweaking a variety of algorithms for machine learning in order to achieve an appropriate combination of interpretability and accuracy. The research's inferences are meant to support HR initiatives and assist data-driven promotion decisions.

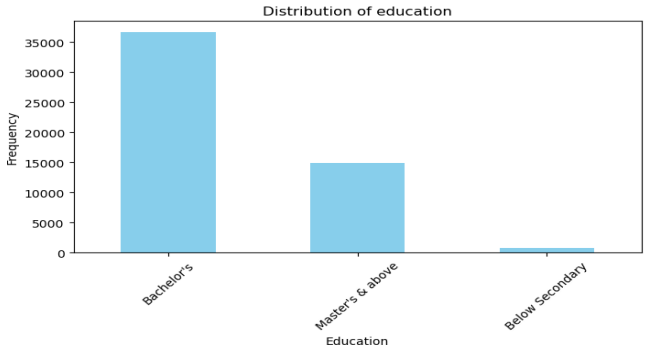
2 DATA

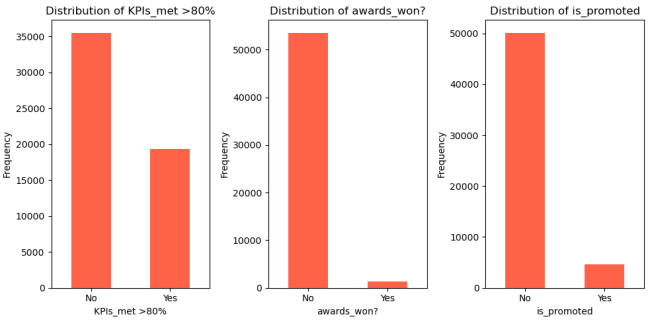
The dataset comes from a large multinational corporation (MNC) that wants to design a predictive model for determining employees who are qualified for promotion at a certain checkpoint, with the goal of speeding up the promotion cycle. The data acquisition process is not specified, but the dataset was developed as a part of a hackathon by Analytics Vidhya, the world's leading and India's largest Data Science community.

2.1 Data description

- **Number of instances:** 54808 rows, each of which represents an employee.
- **Number of attributes**: 14 attributes or columns.
- **Numerical attributes:** 9 numerical attributes,namely, 'employee_id', 'no_of_trainings', 'age','previous_year_rating','length_of_service','KPIs_met>80%','awards_won?','avg_training_score', and the target variable 'is_promoted'.
- **Nominal attributes:** 5 nominal attributes include 'region', 'education', 'gender,' and 'recruitment_channel'.
- **Target variable:** The column 'is_promoted' we wish to predict.
- **Target variable type:** Numerical (integer type), but in nature, it is categorical since it indicates if an employee is promoted (1) or not (0).
- **Distribution of the target variable:** Imbalanced, with about 91.483% of employees not promoted and 8.517% promoted.
- **Missing values:** In 2 attributes:
    - 'education': 2409
    - 'previous_year_rating': 4124







2.2 Data understanding

The target variable is_promoted, which is a binary variable that indicates whether or not an employee is promoted after the evaluation procedure. Since the transition to new roles takes some time, predicting this variable accurately ahead of time will allow the corporation to identify possible promotion candidates early on, facilitating the promotion cycle.

Summary Statistics for numerical attributes:

- **'employee_id'**: A unique identifier, which ranges from 1 to 78298.
- **'no_of_trainings'**: Employees have attended between 1 and 10 training sessions, with an average of approximately 1.25 training sessions per employee.
- **'age'**: The ages of employees range from 20 to 60, with an average age of 34.8 years.
- **'previous_year_rating'**: The ratings vary between 1 to 5, with a mean of around 3.33 suggesting a decent overall performance rating. This variable has missing values that must be addressed.
- **'length_of_service'**: Employees have worked for the organization from 1 to 37 years, with an average duration of 5.87 years.
- **'KPIs_met>80%'**: Approximately 35.2% of employees have met or exceeded 80% of their KPIs.
- **'awards_won?'**: Only roughly 2.3% of the employees have received awards
- **'avg_training_score'**: Training scores vary from 39 to 99, having a mean of around 63.39.
- **'is_promoted'**: This is the target variable, with about 8.5% of employees promoted.

2.3 Data preprocessing

In this stage, Python's *'pandas'* and *'sklearn.preprocessing'* libraries were used. Two features, namely 'education' and 'previous_year_rating,' had missing values. To deal with this, missing values in the 'education' variable were replaced with the mode value, which is "Bachelor's," using 'fillna().' In contrast, missing values in the 'previous_year_rating' variable were replaced with a 0 constant value, as some employees were first-year workers and had no prior rating. This was followed by the one-hot

encoding of categorical variables' department', 'region,' 'education,' 'gender,' and 'recruitment_channel' using OneHotEncoder with drop='first' to eliminate multicollinearity, resulting in a new data frame containing encoded variables. After resetting indices to match the main data frame with the encoded data frame, the two were merged, and the original categorical columns were removed, resulting in a transformed dataset suitable for machine learning applications. The same preprocessing procedures were used for the test dataset to ensure consistent data treatment for both sets.

3. MACHINE LEARNING METHODS USED

A number of machine learning methods were used to create a predictive model for employee promotions, each with its own set of strengths. The *'DecisionTreeClassifier'* provided a simple, interpretable model that was effective for preliminary evaluation and identifying the significance of features. The *'RandomForestClassifier'* was selected because of its resistance to overfitting and capacity to deal with imbalanced datasets, which makes it a good alternative for complicated data. With its straightforwardness and accuracy in detecting related scenarios, the *'KNeighborsClassifier'* offers a simple way to categorize. The *'GradientBoostingClassifier'* was picked for its accuracy in resolving errors in sequential learning as well as its broad application in optimizing different loss functions. *'LogisticRegression'* was added because of its efficiency in tasks involving binary classification, in addition to its capacity to generate probabilities for outcomes. This feature is especially valuable for threshold adjustment in decision-making processes.

3.1 Brief description of the methods used

DecisionTreeClassifier: A classification method based on non-parametric supervised learning.

Parameters:

- **max_depth:** responsible for the tree's depth. Deeper trees are capable of capturing more complicated patterns, but they are prone to overfitting.
- **min_samples_split:** The smallest amount of samples necessary for splitting an internal node, affecting the granularity of the tree.
- **min_samples_leaf:** the smallest amount of samples required for a leaf node, which is important to control overfitting by smoothing the model.

RandomForestClassifier: An ensemble method for improving classification accuracy and controlling overfitting by integrating numerous decision trees.

Parameters:

- **n_estimators:** Number of trees in the forest. Gdenerally, a greater count improves performance but increases the computational cost.
- **max_depth:** Sets the maximum depth of each tree. Deeper trees capture more information, but they are prone to overfitting.
- **min_samples_split:** The smallest amount of samples necessary for splitting an internal node, affecting the granularity of the tree.
- **min_samples_leaf:** The minimum amount of samples necessary at a leaf node. A smaller leaf size increases the likelihood of the model collecting noise in train data.

KNeighborsClassifier: A non-parametric method for classifying samples based on the class most commonly found among their 'k' nearest neighbors.

Parameters:

- **n_neighbors:** The number of neighbors to consider. A lower value makes the algorithm more vulnerable to noise, while a higher value smoothes the decision boundary.
- **weights:** This parameter specifies how votes are weighted. 'Uniform' grants equal importance to all neighbors, whereas 'distance' gives closer neighbors greater influence, which could lead to more complicated decision boundaries.
- **algorithm ['auto', 'ball_tree', 'kd_tree', 'brute']:** The algorithm used to find the closest neighbors. The computational effectiveness of each approach varies based on the dataset's dimensionality and size.
- **p [1, 2]:** The Minkowski metric's power parameter. '1' is the Manhattan distance, which is appropriate for spaces with high dimensions, whereas '2' is the Euclidean distance, which is appropriate for low-dimensional spaces.

GradientBoostingClassifier: A forward stage-wise additive model that constructs trees in an orderly manner, with each tree attempting to fix the flaws of its predecessors.

Parameters:

- **n_estimators:** The amount of boosting steps, which influences the complexity of the model and propensity for overfitting.
- **max_depth:** The maximum depth of the trees, which is critical for avoiding overfitting.
- **learning_rate:** Determines the step size for each iteration while balancing it with the amount of estimators.

LogisticRegression: A regression model that one or more independent variables to estimate the outcome of a categorical dependent variable

Parameters:

C: Controls the strength of regularization, which is critical for preventing overfitting.
solver: An optimization procedure that influences the computing efficiency of the model.

3.2 Brief description of the evaluation criteria

F-1 score:

Selected for its balance between recall and precision, which is critical in employment-related forecasts where both false negatives and false positives are relevant.

Area Under the Receiver Operating Characteristic (AUROC):

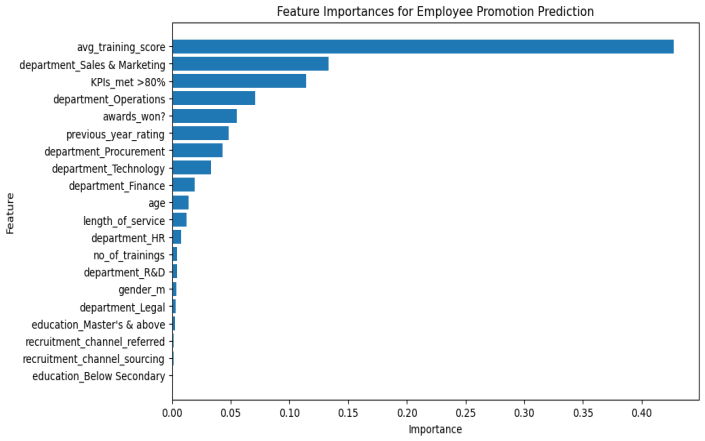
Gives an overall assessment of the model's capacity to differentiate between the two classes. A high AUROC implies that personnel are effectively differentiated as promotable or non-promotable.

These measures verify that the models are not only accurate but also have an important balance between recognizing true positives and minimizing false classifications, which is especially significant in delicate situations like employee promotions.

4. EXPERIMENTS

The abovementioned machine learning methods were evaluated to predict employee promotions, where the features (X) consisted of different employee attributes, excluding the region-related and 'employee_id' columns, and the target (y) was the 'is_promoted' column. Because the test set lacks the target variable, traditional measures such as accuracy, F1-score, and AUROC cannot be used to assess the model's performance. Therefore, the dataset was divided into training and validation sets, with 20% left aside for validation. Since models such as Logistic Regression and KNeighborsClassifier are prone to feature scaling, the features were standardized for these two models. The hyper-parameter tuning process, which was carried out using RandomizedSearchCV to investigate a range of parameter values methodically, was critical in decision-making. Having obtained the best parameters, F-1 scores, and AUROC for each model, I decided to use GradientBoostingClassifier for a test set because it had the highest F-1 (0.507) and AUROC (0.906) scores among all models, showing the best balance between recall and precision, as well as great ability in class separation. After using the GradientBoostingClassifier to the test set, with parameters {'n_estimators': 100, 'max_depth': 5, 'learning_rate': 0.1}, I got predicted classes or probabilities showing the possibility of each employee getting promoted. The most decisive factors for one being promoted turned out to be average training scores, being from the Sales and marketing department, and meeting KPI requirements. The results can be useful for HR analytics, including identifying prospective employees for promotion or refining and improving the predictive model.

5. VISUALIZATION



6. CONCLUSION

In this project, I implemented a set of machine learning models: DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier, GradientBoostingClassifier, and LogisticRegression. The data was rigorously prepared, with irrelevant information such as 'employee_id' and region-specific columns excluded. A substantial portion of this effort required hyperparameter tuning via RandomizedSearchCV, which identified optimal parameters for each model. The result was interesting. As expected, The GradientBoostingClassifier achieved a particularly high AUROC, indicating higher discriminative capability. However, the RandomForestClassifier performed admirably, lagging only slightly behind the GradientBoostingClassifier. This comparative analysis revealed the subtle distinctions among ensemble methods in dealing with prediction tasks. The results of this study can be applied to HR analytics. Predicting employee promotion in advance and accurately has the potential to speed up existing promotion cycles, making data-driven decisions. The benefits of machine learning methods compared to classical statistical methods are obvious. Machine learning models, especially ensemble methods, are able to capture complicated non-linear interactions and relationships more efficiently. Yet, these models come at the cost of intensive computations. Further work could include conducting A/B testing in a real setting to assess the model's influence on actual promotion cycles.