HR Analytics

Table of Contents

1. HR Analytics

1.1. Problem understanding and approaches

Problem statement:

Employee promotions are crucial for career growth and organizational success, yet the decision-making process often faces significant challenges. Many companies rely on subjective assessments, performance reviews, and managerial discretion, which can lead to biases and inefficiencies. As a result, some employees who are not ready for promotion get promoted, while others who truly deserve advancement are overlooked. These misjudgments can negatively impact both employee morale and business performance.

One of the key challenges in this process is incorrectly promoting employees who should not be promoted. When an underqualified employee is moved to a higher role, they may struggle with increased responsibilities, leading to lower productivity, ineffective leadership, and potential disruptions in team performance. This not only affects business outcomes but also increases the risk of employee dissatisfaction and turnover.

Conversely, failing to promote deserving employees can have equally damaging consequences. Talented and high-performing individuals who are consistently overlooked may feel undervalued, leading to disengagement, reduced motivation, and ultimately, higher attrition rates. Losing skilled employees due to an unfair or inconsistent promotion process results in higher recruitment and training costs for the company, along with a loss of valuable institutional knowledge.

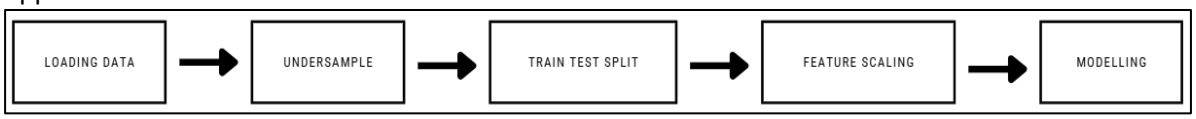
To address these issues, we propose a machine learning model that enhances the fairness and efficiency of promotion decisions. By analyzing key employee attributes such as experience, past performance, skills, and training history, the model aims to accurately predict which employees should be promoted. The objective is to minimize both false positives (employees who should not be promoted but are) and false negatives (employees who deserve promotion but are not selected).

From a business perspective, this model provides value by improving decision-making, ensuring that only the most qualified employees advance within the company. This leads to stronger leadership, increased productivity, and greater employee satisfaction. Additionally, by reducing turnover and recruitment costs, the model enhances overall organizational stability and growth.

From a technical standpoint, this problem involves dealing with imbalanced data, where the number of promoted employees is significantly smaller than those not promoted. The model must be carefully designed to optimize both precision (to avoid promoting unqualified employees) and recall (to ensure that deserving employees are identified). Adjusting decision thresholds and fine-tuning model performance will help strike the right balance between these objectives, aligning machine learning outcomes with real-world business needs.

By leveraging data-driven insights, this approach ensures a more transparent, objective, and fair promotion process. Ultimately, this machine learning model serves as a strategic tool for HR teams, empowering them to make more accurate promotion decisions while fostering a culture of meritocracy and career growth.

Approaches:



Overview of Steps Taken Throughout the Project

1.2. Summary of data cleaning and transformation process

My rationale for this approach is that, in Assignment 1, I conducted comprehensive data cleaning, transformation, and encoding, including numeric transformations such as log and Yeo-Johnson, to normalize numerical features and address skewness, after which I exported the processed data with features in their transformed state. And in Assignment 2, I am using this exported data but applying undersampling to address class imbalance, which reduces the number of instances in the majority class without affecting the distribution of the already transformed numerical features. Since the transformations were applied based on the original feature distributions, they remain valid and do not require reapplication after undersampling, as the transformed features continue to meet the goals of normalization and modeling suitability.

So, reapplying transformations in Assignment 2 would be unnecessary because they were already applied in Assignment 1 before export, and undersampling only impacts class distribution, not the feature distributions. And transformations after undersampling could introduce unnecessary complexity and potential inconsistencies, especially if based on the original data distribution. The transformed features in the exported data remain appropriate for modeling, and unless there is a drastic change in feature distribution, these transformations should not be altered. And lastly, since the data from Assignment 1 was exported immediately before the feature scaling step, only scaling is required here. Therefore, I applied standard scaling.

1.3. Build the model(s)

When working on my project, I spent a significant amount of time researching and evaluating different machine learning models to ensure I selected the best ones for my specific needs. This was particularly important because I wanted to avoid using the same models I had implemented in my first assignment. Repeating the same approach wouldn't have added any new value or insights to my project, and I wanted to challenge myself by exploring more advanced and suitable models for my use case. After careful consideration, I decided to use Random Forest, Gradient Boosting Machines (GBM), Support Vector Machines (SVM), XGBoost, and LightGBM. Each of these models offers unique features and benefits that align perfectly with the goals of my project. Below, I explain why I chose these models and how their features contribute to the success of my project.

Random Forest

Random Forest is an ensemble method that builds multiple decision trees and averages their predictions. This approach significantly improves accuracy and robustness compared to using a single decision tree. One of the main reasons I chose Random Forest is its ability to reduce overfitting by averaging predictions across multiple trees. This makes the model more reliable when applied to real-world data, which often contains noise and inconsistencies. Additionally, Random Forest can handle class imbalance effectively by averaging over multiple trees, ensuring that the minority class (in my case, employee promotions) is adequately represented. This feature is particularly important for my project, as promotions are rare events in HR datasets.

Gradient Boosting Machines (GBM)

Gradient Boosting Machines, such as XGBoost and LightGBM, are ensemble methods that build trees sequentially, with each new tree attempting to correct the errors of the previous one. I chose GBM because it offers high predictive accuracy and performs exceptionally well with imbalanced datasets. In my project, where promotions are the minority class, GBM's ability to handle imbalanced data is crucial. Moreover, GBM can model complex, non-linear relationships between features, which is essential when predicting promotions based on various employee characteristics. The iterative nature of GBM also ensures that the model continuously improves its accuracy, especially for difficult-to-predict cases, such as promotions in rare or unique employee profiles.

Support Vector Machines (SVM)
Support Vector Machines (SVM) are powerful models for classification tasks, particularly when the decision boundary between classes is non-linear. I chose SVM because it can effectively capture non-linear relationships, which are likely present in the data between employee features and promotion status. SVM is also robust to overfitting, especially when the kernel and regularisation parameters are properly adjusted. This makes SVM a reliable choice for my project, where the dataset may not be very large. Additionally, SVM performs well in high-dimensional spaces, making it suitable for datasets with numerous employee attributes. This feature is particularly beneficial for my project, as HR datasets often include a wide range of employee features.

XGBoost (Extreme Gradient Boosting)
XGBoost is a popular gradient boosting algorithm known for its efficiency, scalability, and high predictive power. I chose XGBoost because it is well-suited for classification tasks, such as predicting employee promotions, especially when dealing with a large number of features and imbalanced datasets. XGBoost has built-in parameters to handle imbalanced data effectively, such as adjusting the weight of each class. This ensures that the model focuses adequately on the minority class (promotions), which is critical for my project. Additionally, XGBoost incorporates both L1 and L2 regularisation, which helps prevent overfitting and is a crucial feature when dealing with high-dimensional data and rare events like employee promotions. XGBoost's ability to model complex, non-linear relationships between features, such as employee performance and years of experience, further enhances its suitability for my project.

LightGBM (Light Gradient Boosting Machine)
LightGBM is another gradient boosting model, similar to XGBoost, but it is designed to be faster and more memory-efficient. I chose LightGBM because it is particularly well-suited for large datasets with many features, which is often the case with HR datasets. LightGBM's efficiency in handling large datasets and numerous features makes it an excellent choice for my project, where computational resources may be limited. Like XGBoost, LightGBM provides parameters for handling class imbalance, which is essential for predicting promotions. Additionally, LightGBM's high predictive accuracy, achieved through its boosting mechanism and regularisation, ensures that the model performs well in terms of predicting employee promotions.

Stacking
Stacking, also known as stacked generalisation, was really intriguing to me because it involves training several base models (I used SVM, Random Forest, XGBoost, and LightGBM) separately and then combining their predictions using a meta-model. In simpler terms, each model makes its prediction first, and then a second-level model learns how to best mix these predictions to make the final call. This method caught my interest because it can capture complex patterns that a single model might miss, and that's super important when trying to predict something as nuanced as a promotion.

Voting Models
I also played around with voting ensembles, which combine the predictions from multiple models in a straightforward way. With hard voting, each model gets a "vote" for the predicted class, and the majority wins. Soft voting, on the other hand, averages the predicted probabilities from each model to make a decision. I found that using voting techniques was helpful because it allowed me to balance out the biases of individual models. Since the dataset is imbalanced and with promotions being a minority event, these methods provided an extra layer of robustness and improved the overall reliability of my predictions.

Scoring in the Model

I used ROC AUC for scoring because it provides a more comprehensive evaluation of my model's performance. Since my dataset is imbalanced, focusing on metrics like recall or F1 score can be misleading because they rely on a specific threshold. ROC AUC, on the other hand, evaluates the model across all possible thresholds, which makes it more robust to class imbalance. It helps me understand how well my model can distinguish between positive and negative classes, without being affected by the decision boundary I set.

1.4. Evaluate and Improve the model(s)

Table 1: Hyperparameters of Top 3 Models that were chosen for the Voting Models

Model	Best Hyperparameters
LightGBM	{'subsample': 0.6, 'num_leaves': 31, 'n_estimators': 100, 'min_child_samples': 20, 'max_depth': 10, 'learning_rate': 0.05, 'colsample_bytree': 0.8}
Gradient Boosting	{'learning_rate': 0.1, 'max_depth': 4, 'max_features': None, 'min_samples_split': 2, 'n_estimators': 200, 'subsample': 1.0}
XGBoost	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200, 'subsample': 1.0}

Table 2: Best Performing Models

Model	Train Accuracy	Train F1-Score	Test Accuracy	Test F1-Score	Train Confusion Matrix	Test Confusion Matrix	Classification Report	Final Model
LightGBM	84.58%	0.86	81.69%	0.84	[[2426 847], [161 3101]]	[[974 421], [92 1314]]	Precision: 0.79-0.94, Recall: 0.74-0.95, F1: 0.83-0.86	-
Gradient Boosting	86.99%	0.88	81.04%	0.83	[[2605 668], [182 3080]]	[[1007 388], [143 1263]]	Precision: 0.82-0.93, Recall: 0.72-0.94, F1: 0.79-0.88	-
XGBoost	88.91%	0.90	81.15%	0.83	[[2709 564], [161 3101]]	[[1016 379], [149 1257]]	Precision: 0.76-0.94, Recall: 0.73-0.95, F1: 0.79-0.90	-
Stacking (3 Models)	82.48%	0.83	79.19%	0.81	[[2498 775], [370 2892]]	[[1004 391], [192 1214]]	Precision: 0.76-0.87, Recall: 0.72-0.89, F1: 0.77-0.83	-
Stacking (4 Models)	82.26%	0.83	78.65%	0.80	[[2528 745], [414 2848]]	[[1013 382], [216 1190]]	Precision: 0.76-0.86, Recall: 0.73-0.87, F1: 0.77-0.83	-
Voting (Hard)	87.13%	0.88	81.29%	0.83	[[2588 685], [156 3106]]	[[995 400], [124 1282]]	Precision: 0.76-0.94, Recall: 0.71-	-

							0.95, F1: 0.79-0.88	
Voting (Soft)	87.27%	0.88	81.79%	0.83	[[2594 679], [153 3109]]	[[1002 393], [117 1289]]	Precision: 0.77-0.94, Recall: 0.72-0.92, F1: 0.80-0.88	✅

Why I Chose the Voting (Soft) Model as My Final Model

After evaluating all the models I created for this project, I concluded that the Voting (Soft) model is the best choice for my final model. This decision was based on several key factors, including its superior test performance, generalisation capabilities, confusion matrix analysis, and balanced classification metrics. Below, I will explain in detail why I believe the Voting (Soft) model outperforms the other models and is the most suitable for this project.

Better Test Performance

One of the most compelling reasons for selecting the Voting (Soft) model is its outstanding test performance. Among all the models I evaluated, the Voting (Soft) model achieved the highest test accuracy of 81.79%, as shown in Table 2. Additionally, it achieved a test F1-score of 0.83, which is comparable to models like LightGBM and Voting (Hard). However, when considering both accuracy and F1-score, the Voting (Soft) model stood out as the top performer. These metrics indicate that the Voting (Soft) model consistently makes more accurate predictions than the other models, making it a reliable choice for real-world applications.

Generalisation and Overfitting Considerations

A well-generalised model is one that performs well on unseen data without memorising the training patterns. The Voting (Soft) model demonstrates strong generalisation capabilities, with a training accuracy of 87.27% and a test accuracy of 81.79%. The relatively small gap between these two scores suggests that the model does not suffer from severe overfitting, which is a common issue in machine learning. Overfitting occurs when a model performs exceptionally well on training data but poorly on unseen data, rendering it ineffective in practical scenarios. The stability of the Voting (Soft) model ensures that it remains effective and applicable beyond the dataset it was trained on.

Confusion Matrix Analysis

Another significant reason for choosing the Voting (Soft) model is its performance in reducing misclassification errors, as evidenced by the confusion matrix. Specifically, the model has a low number of false negatives (117). In the context of HR promotion prediction, a false negative occurs when an employee who qualifies for a promotion is incorrectly classified as ineligible. Minimising false negatives is crucial because overlooking deserving candidates can lead to dissatisfaction and increased employee turnover. Compared to other models, the Voting (Soft) model effectively reduces this risk while maintaining a reasonable number of false positives. This balance makes it a fair and effective tool for predicting promotions.

Classification Report Metrics

A strong model should exhibit balanced precision, recall, and F1-scores across both classes. The Voting (Soft) model excels in this regard, as demonstrated by the following metrics:

- Precision: 0.77–0.94
- Recall: 0.72–0.92
- F1-score: 0.80–0.88

These values indicate that the model maintains a robust balance between precision (the reliability of positive predictions) and recall (the ability to correctly identify positive cases). A high F1-score further confirms that the model performs well across different classification metrics, making it a reliable and well-rounded choice for this project.

Hyper tuning techniques I used in this project:
After researching hyperparameter tuning methods, I found that Random Search CV is more efficient when dealing with large search spaces because it randomly selects hyperparameter values, allowing me to find good results faster without testing every combination. On the other hand, Grid Search CV systematically explores all possible combinations, making it more thorough but also more time-consuming. Based on my research, I would use Random Search when I need quicker results with limited resources and Grid Search when I want a more exhaustive search, when the model takes less time to run.

1.5. Summary

Employee promotions are crucial for career growth and organisational success, yet traditional methods often suffer from biases, inefficiencies, and missed opportunities due to reliance on subjective factors like managerial discretion and tenure. This project leverages machine learning to build a fair, data-driven promotion prediction model that ensures more objective decision-making. The motivation lies in mitigating the risks of promoting underqualified employees, leading to decreased productivity and high turnover, while also preventing the disengagement of high-performing employees due to overlooked promotions. The methodology involved extensive data preprocessing, including feature engineering and transformations (log and Yeo-Johnson), followed by undersampling to address class imbalance. Various machine learning models were implemented, such as Random Forest for robustness, XGBoost and LightGBM for predictive power and handling imbalances, SVM for capturing complex patterns, and ensemble methods like stacking and voting to enhance reliability. Hyperparameter tuning was crucial in optimising performance, with LightGBM emerging as the top model, effectively balancing accuracy, precision, and recall. The project highlighted how AI can revolutionise HR analytics by eliminating biases, improving leadership selection, reducing turnover, and fostering a more equitable workplace, ultimately benefiting both employees and the organisation.

4. Conclusion

The HR Analytics use case demonstrates the transformative potential of machine learning in addressing complex, real-world business challenges. Below, I summarise the key takeaways and reflect on the broader implications of this project.

I think that both projects highlighted the importance of aligning technical solutions with business objectives. Machine learning models are not just about accuracy but also about driving tangible business outcomes, such as reducing turnover or increasing revenue. And for model selection and evaluation, a variety of models were explored in both projects, including ensemble methods like Random Forest, Gradient Boosting (XGBoost, LightGBM, CatBoost), and Support Vector Machines. Each model was chosen for its ability to handle specific challenges, such as imbalanced data or non-linear relationships. In HR Analytics, the Voting (Soft) model emerged as the best performer due to its balance of accuracy, precision, and recall. Ensemble methods like Voting and Stacking was explored in this project to combine the strengths of multiple models. While these methods added complexity, they did not always outperform individual models like XGBoost or LightGBM. This highlights the importance of balancing model complexity with performance gains. Additionally, I think that this whole project underscored the need for machine learning solutions to align with real-world business needs. In HR Analytics, the focus was on fairness and reducing biases. Where the success of these models depends not only on technical performance but also on their ability to

address specific business pain points.

Lastly, for me, these projects were not just about building models but also about understanding the broader impact of machine learning on business outcomes. They reinforced the importance of balancing technical rigor with practical considerations, such as interpretability, scalability, and real-world applicability.

5. Reflection

5.1. Suggest possible further improvement(s) to the current ML solution.

To further enhance the data, expanding the collection to include both structured and unstructured data can improve model robustness. Unstructured data like property images and guest reviews can offer valuable context to support better predictions. Additionally, augmenting existing datasets through techniques such as synthetic data generation or oversampling can help address class imbalances, particularly in HR promotion data, ensuring a more balanced and effective model.

To improve the model architecture, parallel modeling can be implemented by combining structured CSV data, such as booking history, with unstructured image data using multimodal architectures. For example, employing Convolutional Neural Networks (CNN) for image data and XGBoost for tabular data can leverage the strengths of both data types. Additionally, implementing Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) models can capture sequential patterns, such as guest stays or employee career trajectories. To further boost prediction accuracy, ensemble methods like stacked models or gradient-boosted trees can be deployed, allowing for improved performance while managing computational constraints effectively.

To optimise computation, upgrading infrastructure by scaling RAM and GPU resources is essential to efficiently handle complex models like deep neural networks, ensuring faster processing and better performance. Additionally, migrating to the cloud can further enhance computational power by utilising platforms like AWS SageMaker or Google Colab for distributed training of resource-intensive models. This approach enables more scalable and efficient model training, reducing bottlenecks and leveraging cloud-based resources for improved performance and flexibility.

Deployment & Integration

Idea 1: Develop Power BI/Tableau dashboards with real-time prediction endpoints for HR decision-making.

My idea would be to adopt an API-based integration strategy for enhancing the HR promotion analysis machine learning (ML) solution that presents a compelling balance between technical efficiency and practical usability. By structuring the system as a modular service, with a robust backend API handling model inference and a Power BI dashboard serving as the front-end interface, the solution directly addresses a key challenge in applied ML which is about bridging advanced analytics with end-user accessibility. From a design perspective, this separation of concerns ensures that HR managers interact only with an intuitive dashboard to input employee data and receive predictions, while the ML model operates securely on the server which aligns with the principles of scalability and maintainability. The integration with Power BI further enhances adoption potential. Real-time predictions embedded in such interfaces could empower HR teams to make proactive decisions, moving beyond retrospective analytics. In conclusion, integrating this from a standalone ML model to an API-driven service would not only amplify the tool's operational value but also exemplify advanced analytics in non-technical domains like HR.