SYNTAX SOCIETY

Rakamin

The Promotion Paradox

Fixing Biased Talent Promotion Decisions Through Data-Driven HR Analytics

a project report by Syntax Society

Dataset

Phyton Code

Github

Interactive Dashboard

Our team

Keisya Nisrina Aulia
Data Scientist

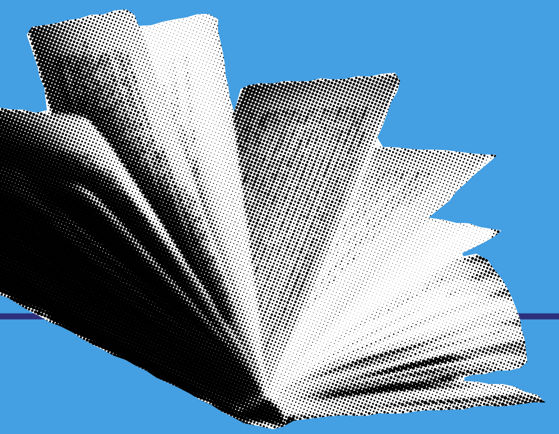
Febiansyah Annaufal Ahnaf Fauzi
Data Scientist

Bayu Maitra
Project Manager

Dian Ulhaq Qurrata A'yun
Data Analyst

Mashlahatul Husna
Data Engineer

Table of Content



Stage 0: Background.

The Value of Talent Promotion

Talent promotion plays a vital role in a company's long-term sustainability. It is the foundation of organizational growth because:

Cost efficient

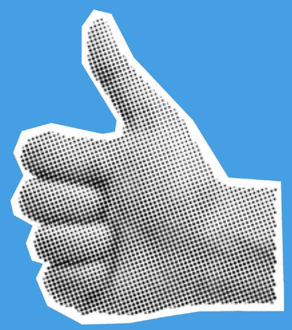
Cheaper than hiring externally, as it eliminates the need for recruitment, onboarding, and training processes.

Retention & Motivation

Drives productivity, motivation, retention, and long-term performance.

Trust

Fair and transparent promotions strengthen trust and organizational culture.



When done well, promotion systems create a culture of fairness and drive productivity across teams. That is why building a healthy, transparent promotion system is crucial for every HR division.

The Promotion Paradox

Many companies still struggle to implement fair and effective promotion systems, as shown in the research by Chartered Management Institute (CMI).

82%

Around 82% companies fail to select the right leaders

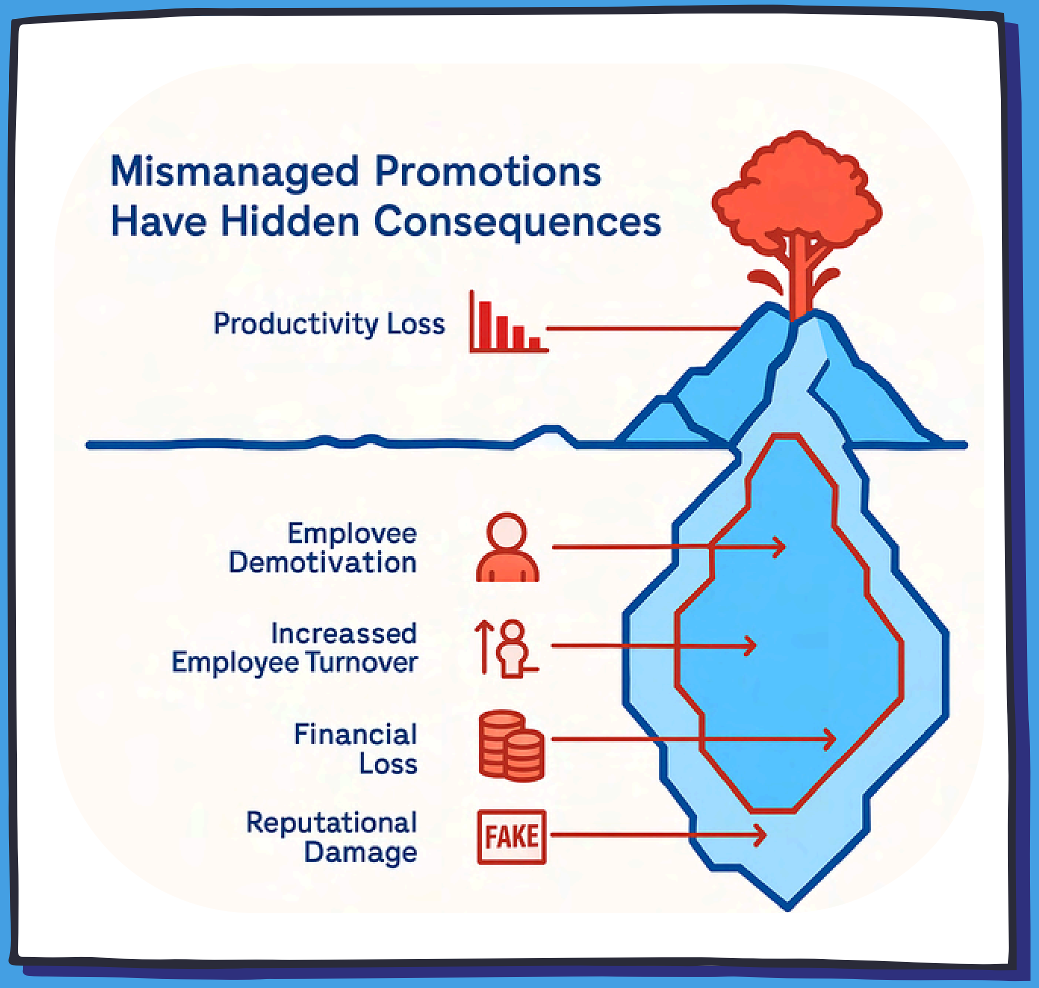
31%

Only 31% managers are engage in their workplace

29%

Only 29% employees trust their managers

WHY?



There are problems in the system. When the system fail, the wrong people rise.

This isn’t just happening globally.
It also occurs here in Indonesia,
including within Rakamin.

#1: Manual Data Record

Manual data records weaken data integrity, causing lost information, inconsistencies, and input errors. This makes HR decision-making biased, unreliable, and slow, often pushing companies to hire new people instead, which increases costs.

Feature	Outliers	Percentage
Age	5	0.5%
Years_at_Company	3	0.3%
Training_Hours	5	0.5%
TOTAL	13	1.3%

	Employee_ID	Age	Years_at_Company
35	EMP0036	22.0	24.0
59	EMP0061	23.0	29.0
93	EMP0095	23.0	29.0
95	EMP0097	26.0	27.0
115	EMP0117	23.0	27.0
135	EMP0137	24.0	26.0
162	EMP0164	24.0	28.0

Implausible values

- Age (Max = 200, Min = -3)
- Years_at_Company (Max = 100, Min = -5)
- Training_Hours (Max = 5000, Min = -50)
- Years_at_Company > Age = 13 Data
- Age - Years_at_Company < 18 = 212 Data

Outliers

- Age (Max = 200, Min = -3)
- Years_at_Company (Max = 100, Min = -5)
- Training_Hours (Max = 5000, Min = -50)

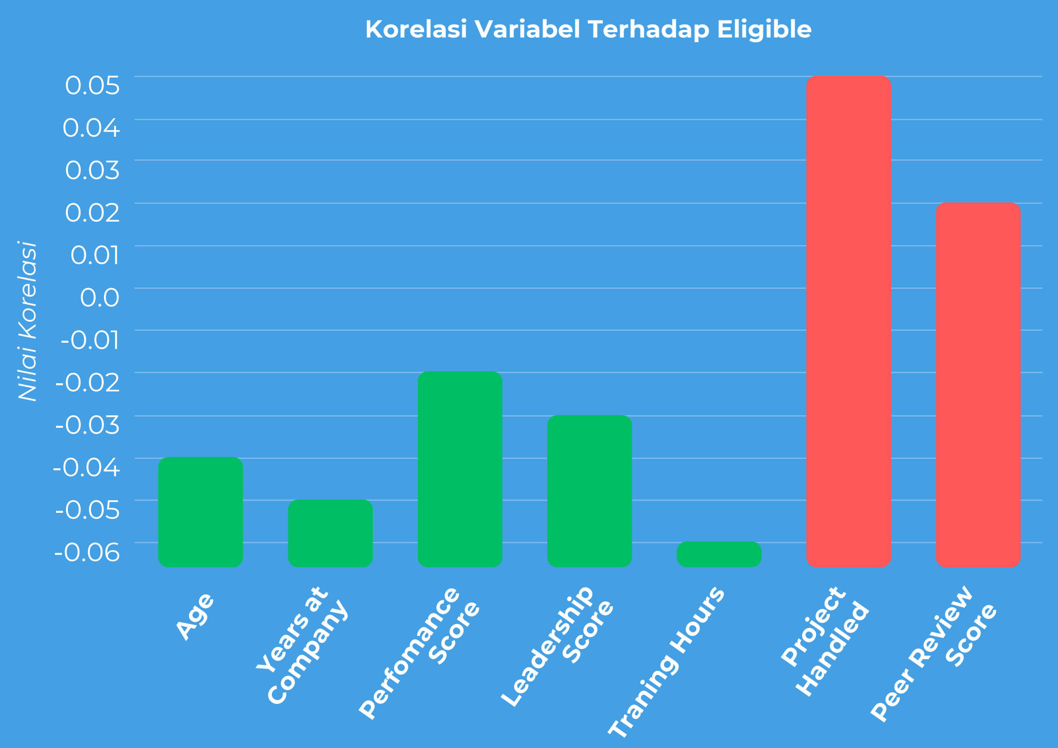
Missing values

- Age (50)
- Years_at_Company (49)
- Performance_Score (50)
- Leadership_Score (50)
- Training_Hours (50)
- Project_Handled (50)
- Peer_Review_Score (50)
- Current_Position_Level (50)
- Promotion_Eligible (50)
- TOTAL: 449

Source: Rakamin's Talent Promotion dataset

#2: Biased Promotion Criteria

Rakamin's dataset shows bias in defining promotion eligibility criteria by relying heavily on Peer_Review_Score and Projects_Handled, which leads to muddled outcomes. Furthermore, the promotion_eligible column also shows indications of bias, as there are several employees with high performance and competency scores who are still listed as not eligible, indicating a discrepancy between the actual data and the resulting promotion decisions.



Employee_ID	Projects_Handled	Peer_Review_Score	Promotion_Eligible
EMP0050	19	98	ya
EMP0324	19	94	tidak

Category	Eligible	Ineligible
Years_at_Company	28	67
Performance_Score	57	127
Leadership_Score	36	29
Training_Hours	18	35
Projects_Handled	24	67
Peer_Review_Score	9	31

Employee_ID	Performance_Score	Leadership_Score	Peer_Review_Score	Promotion_Eligible
EMP0939	1	12	4	ya
EMP0939	5	96	90	tidak

The number of eligible employees is 134, while the number of ineligible employees is 284.

Source: Rakamin's Talent Promotion dataset

#3: Hiring = More Cost

Hiring new employee instead of give employee promotion can lead to talent loss and costly employee turnover. **According to Eos Global Expansion, the Cost per Hire (CPH) can increase by 10–20%** of the employee's salary due to hidden expenses such as recruitment agency fees, background checks, extended HR working hours, and interview assessments.

According to Airswift, the average onboarding cost per new employee ranges from Rp16 million to Rp40 million, depending on role complexity and required expertise.

6 **Hiring new employee for Middle position**

1 **Hiring new employee for Senior position**

7 **Hiring new employee for Lead position**

Level	Hidden Expenses	Total
Middle	Rp1.900.000	Rp11.400.000
Senior	Rp3.750.000	Rp3.750.000
Lead	Rp4.500.000	Rp31.500.000

Loss
47jt

14.11%
Save cost

Source: *Eos Global* *Airswift* *Salary*

From bias to balance

We aim to shift from the current situation to an ideal state, but there's a gap we must bridge to get there.

CURRENT CONDITION

- Data collected but not analyzed. Insights stay hidden.
- Manual, biased, and error-prone process.
- HR struggles to spot and promote talent on time.

THE GAP

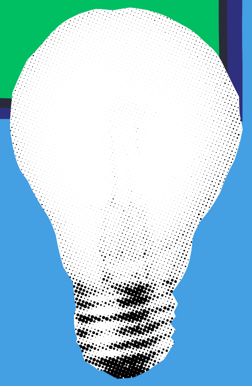
- Incomplete and inconsistent data.
- No predictive or decision-support system.

IDEAL CONDITION

- Data Integrity: Clean, standardized HR data
- Decision Support System: Automated, fair, and explainable decisions.

Goal & objectives

Building a dashboard for data-driven promotion system to be used by HR for fair and objective decisions.



1. Develop and test a machine learning based prediction model
2. Integrate the model into an interactive dashboard using endpoint API
3. Monitoring and evaluating the system's impact

Machine learning models

We will use five machine learning models to ensure accuracy and validity, allowing us to select the best-performing model for the final output.

Tree-Base

1. **Decision Tree**
   Simple baseline with clear, interpretable rules.
2. **Random Forest**
   Reduces overfitting and improves stability via ensembling
3. **XGBoost**
   Powerful boosting for complex patterns and imbalanced data

Linear Model

1. **Logistic Regression**
   Simple and interpretable baseline model
2. **SVM**
   Flexible classification and regression by finding the best hyperplane

Clustering

1. **K-Means**
   Fast and effective for round and separate clusters, but sensitive to outliers.
2. **GMM**
   Flexible soft-clustering for overlapping clusters, but computationally heavier.
3. **K-Medoids**
   Outlier-resistant and based on original data points, but slower on large datasets.

Feasibility, risk & mitigation

Resource feasibility

- Skilled team: Data Analyst, Data Scientist, Data Engineer
- Data is available

Technical feasibility

- The project utilizes free tools such as Google Colab and Looker Studio. No major investment is required

Business feasibility

- The project improves HR's time and operational efficiency
- Reduces subjectivity in promotion decisions and enhances transparency

Risk

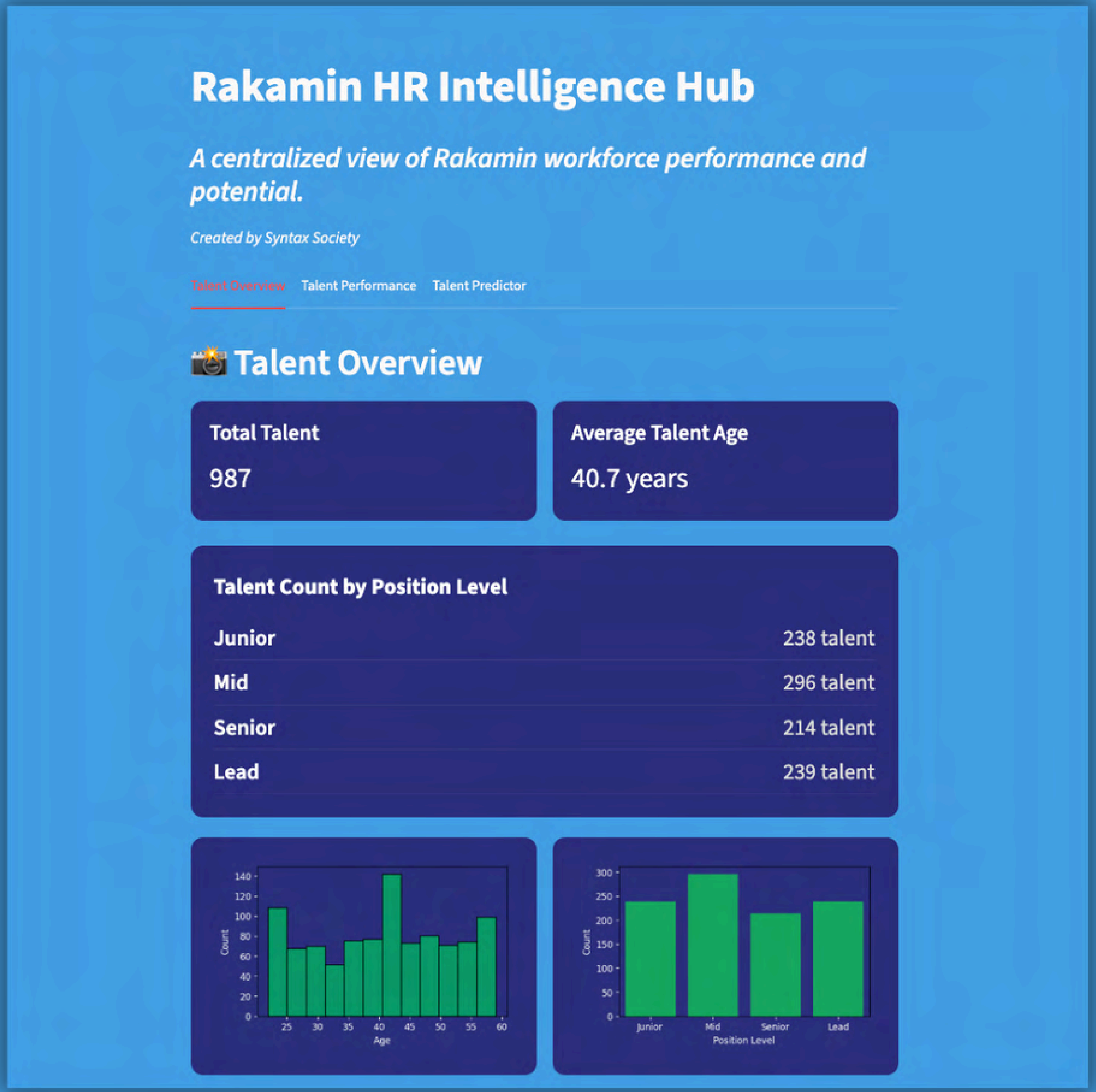
- Inconsistent Data Entry
- Stakeholders Lack Machine Learning Understanding
- High Development Cost
- Unintegrated Data Sources

Mitigation

- Implement data validation rules
- Use explainable AI and conduct training and awareness sessions to help HR and stakeholders
- Utilize free and open-source tools or libraries
- Standardize HR data structure

Output

A unified HR dashboard that delivers clear, real-time insights into employee performance, development progress, and promotion eligibility.



Maintenance

- Monitoring is done by regularly tracking the model's performance and data changes on a monthly basis.
- Maintenance is done by retraining, updating parameters, or replacing the model every 3–6 months.

Impact & Success metrics

COMPANY & HR DEPT

- The promotion process becomes fair, objective and transparent
- Decision-making becomes faster
- Reduces turn-over cost

EMPLOYEE

- More equitable promotion opportunities for employees
- Reduced risk of turnover among competent employees .

> 95% data is standardized and validated

>90% Machine learning model accuracy

24/7 Online dashboard with real-time information

[COST EFFICIENCY]

Workflow Diagram

CRISP-DM (Employee Career Advancement Prediction)

Business Understanding → Data Understanding → Data Preparation → Modeling → Evaluation → Deployment

TO DO LIST	PIC & DEADLINE	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
• PROJECT PROPOSAL • BUSINESS RESEARCH • WORKFLOW DIAGRAM	Project Manager & Data Analyst *1 - 7 Nov*	Business Understanding				
• EXPLORATORY DATA ANALYSIS (EDA) • DATA DICTIONARY MAPPING • INITIAL CORRELATION HEATMAP • CLEANED DATASET • FEATURE ENGINERING	Project Manager & Data Engineer *9 - 14 Nov*		Data Understanding & Preparation			
• MACHINE LEARNING MODELING • TRAINED MODELS + COMPARISON REPORT	Project Manager & Data Scientist *16 - 21 Nov*			Modeling		
• MODEL EVALUATION • EXPLAINABILITY REPORT (SHAP/LIME) • FAIRNESS ASSESSMENT REPORT	Project Manager & Data Scientist *23 - 28 Nov*				Evaluation	
• DEPLOYMENT / DASHBOARD / API • FINAL PROJECT REPORT & PRESENTATION • BUSINESS RECOMMENDATION	All Role *30 Nov - 5 Dec*					Deployment

Stage 1: EDA.

Data Characteristics

Here is the initial description of Rakamin's dataset. At a glance, everything appears fine. However, our EDA will later revealed several important findings.

Descriptive statistic

Statistic	Age	Years_at_Company	Performance_Score	Leadership_Score	Training_Hours	Projects_Handled	Peer_Review_Score	Promotion_Eligible
count	950.0	951.0	950.0	950.0	950.0	950.0	950.0	950.0
mean	40.94	14.968454	2.969474	50.117895	109.286316	10.016842	51.227368	0.294737
std	12.863864	9.319084	1.417977	28.668058	184.616517	5.666507	29.141002	0.456165
min	-3.0	-5.0	1.0	0.0	-50.0	0.0	0.0	0.0
25%	31.0	7.0	2.0	25.0	51.25	5.0	25.0	0.0
50%	41.0	15.0	3.0	51.0	102.5	10.0	51.0	0.0
75%	50.0	23.0	4.0	75.0	150.75	15.0	78.0	1.0
max	200.0	100.0	5.0	99.0	5000.0	19.0	99.0	1.0

Data Type

Employee_ID	object
Age	float64
Years_at_Company	float64
Performance_Score	float64
Leadership_Score	float64
Training_Hours	float64
Projects_Handled	float64
Peer_Review_Score	float64
Current_Position_Level	object
Promotion_Eligible	float64

Observation

- There are 1000 rows and 10 column
- There are 2 categorical data and 8 numeric data
- There are 449 missing data
- There are no duplicate data

Baseline

28% 72%

- Employees are eligible for promotion
- Employees are not eligible for promotion

Target Feature

`Promotion_Eligible`

"1" **Employees eligible** for promotion

"0" **Employees not eligible** for promotion

Source: Rakamin's Talent Promotion dataset

Data Anomalies

1. Years at company is negative and 2 data values are more than 50 years.

Employee_Id	Years_at_Company
EMP0291	60
EMP0624	-5
EMP0924	100

2. 1 employee have minus training and 4 employees have more than 500 hours of training.

Employee_Id	Training_Hours
EMP0345	-50
EMP0850	999
EMP0379	1000
EMP0394	2000
EMP0926	5000

3. There are 255 data showing workers who work under the legal age of 18 years.

Employee_Id	Age	Years_at_Company
EMP0001	24	21
EMP0018	26	15
EMP0014	36	24
EMP0017	38	27
EMP0004	39	24

4. There are 41 data that show that the length of service is higher than the employee's age.

Employee_Id	Age	Years_at_Company
EMP0049	-3	20
EMP0036	22	24
EMP0061	23	29
EMP0095	23	29
EMP0117	23	27
EMP0137	24	26
EMP0164	24	28
EMP0097	26	27

5. In the Age column, there are negative age, under 18 years, and 3 talents over 100 years.

Employee_Id	Age
EMP0049	-3
EMP0372	10
EMP0998	100
EMP0304	150
EMP0775	200

6. There are 449 missing value in 9/10 coloums

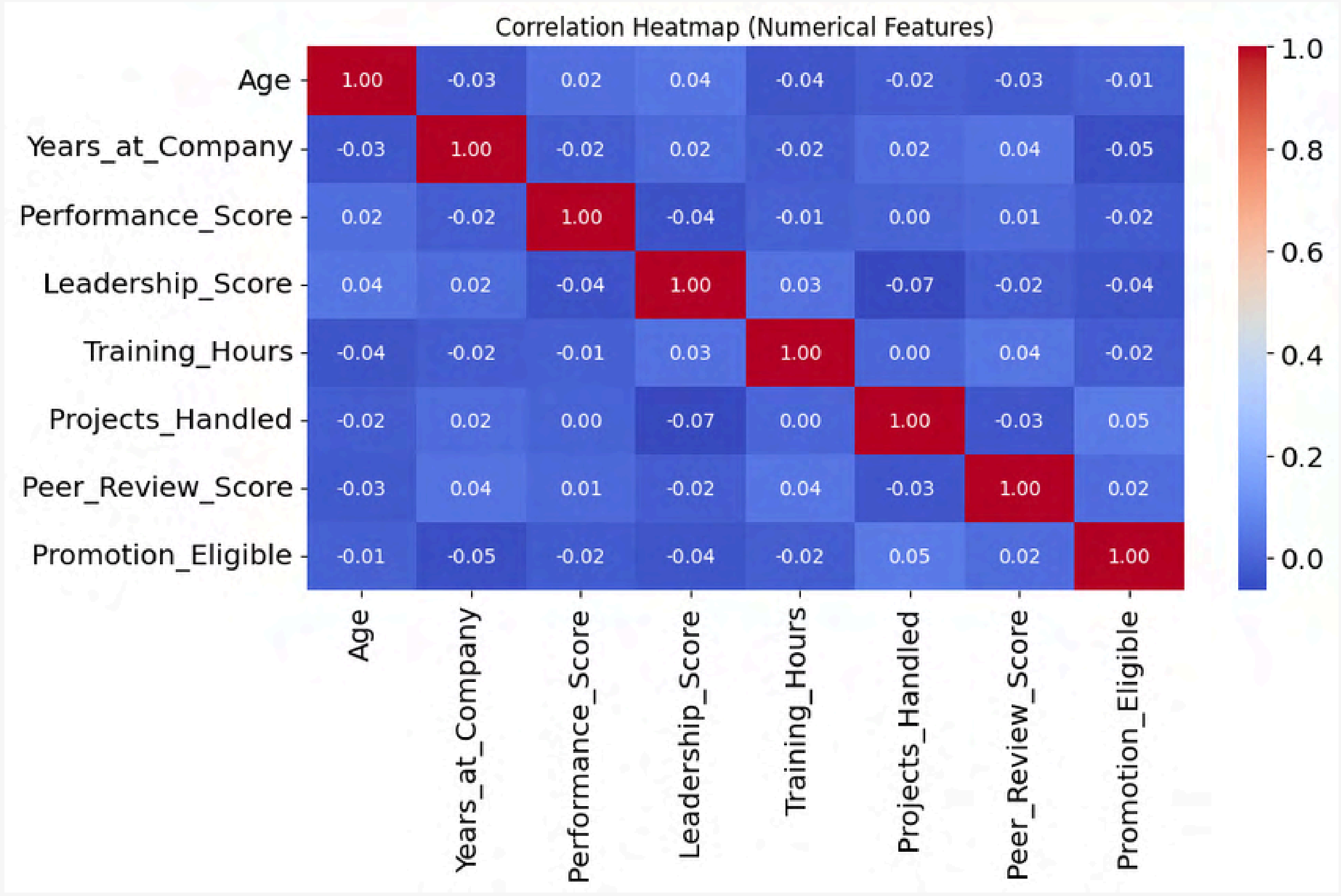
- Age (50)
- Years_at_Company (49)
- Performance_Score (50)
- Leadership_Score (50)
- Training_Hours (50)
- Project_Handled (50)
- Peer_Review_Score (50)
- Current_Position_Level (50)
- Promotion_Eligible (50)

Source: Rakamin's Talent Promotion dataset

Distribution & Correlation

Feature	Outliers	Percentage
Age	5	0.5%
Years_at_Company	3	0.3%
Training_Hours	5	0.5%
TOTAL	13	1.3%

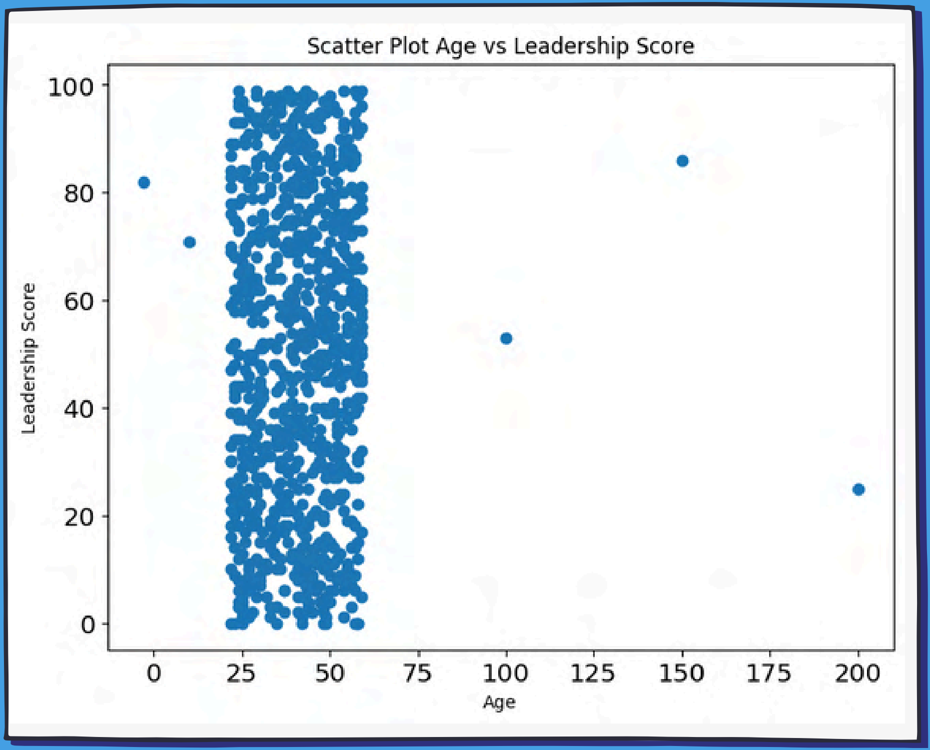
Approximately 1.3% of the observations are identified as outliers, causing the overall distribution to deviate from normality.



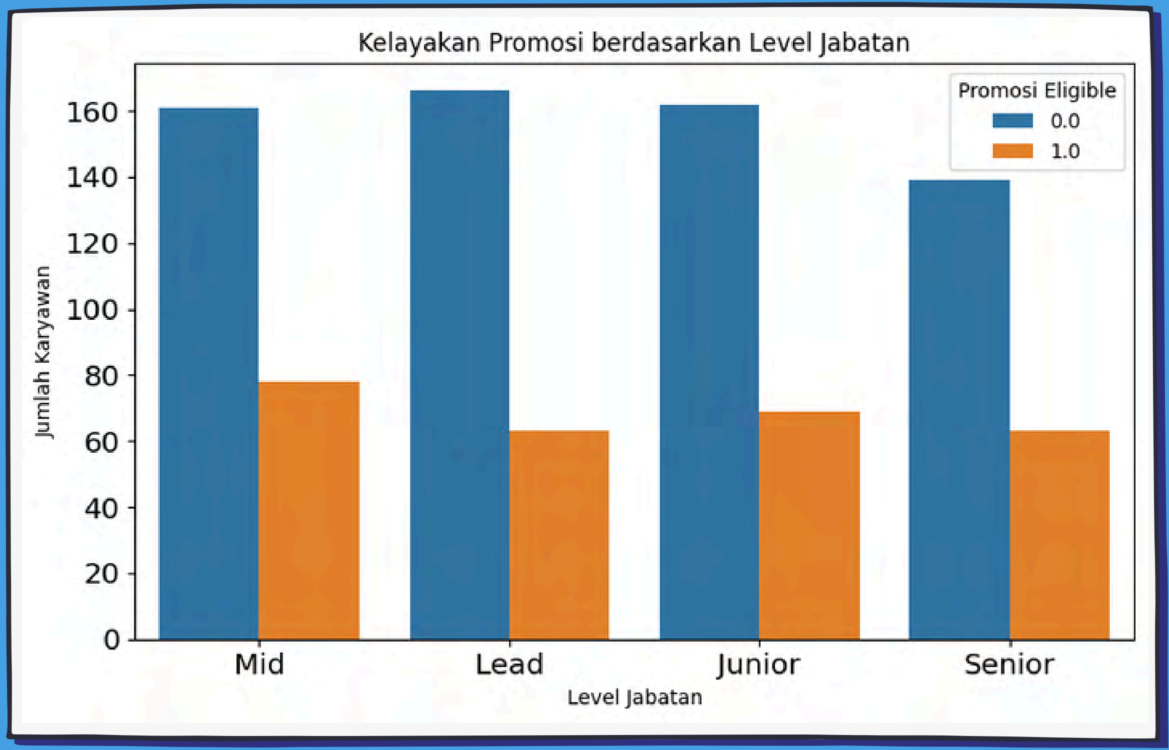
Projects_Handled and Peer_Review_Score have the highest correlation with the target (Promotion_Eligible), although neither of them shows a strong or statistically significant relationship.

Source: Rakamin's Talent Promotion dataset

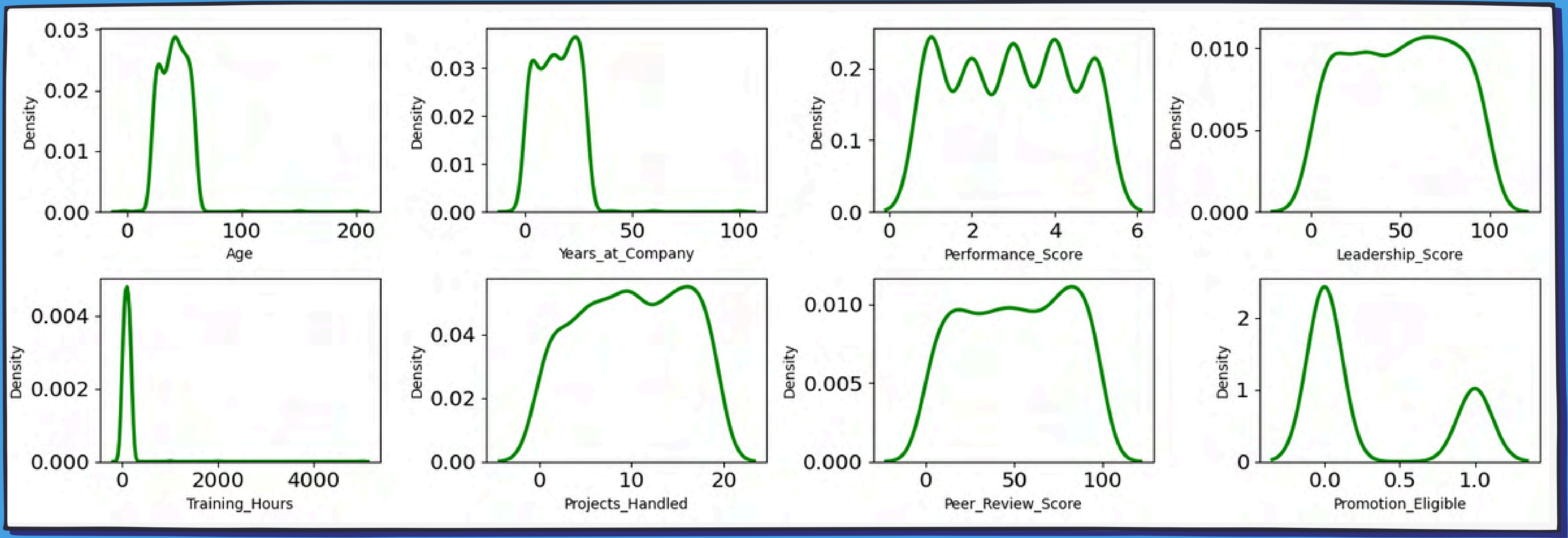
Distribution & correlation



The Leadership Score is mostly concentrated in the 25–60 age range, with wide variation, indicating that age is not a strong predictor. The chart also shows extreme outliers at unrealistic ages (100, 150, 200), suggesting data entry errors that require cleaning.



Despite the large number of employees at each level, the overall promotion eligibility rate (eligible = 1.0) is low, indicating significant barriers to the promotion process, regardless of current job level.

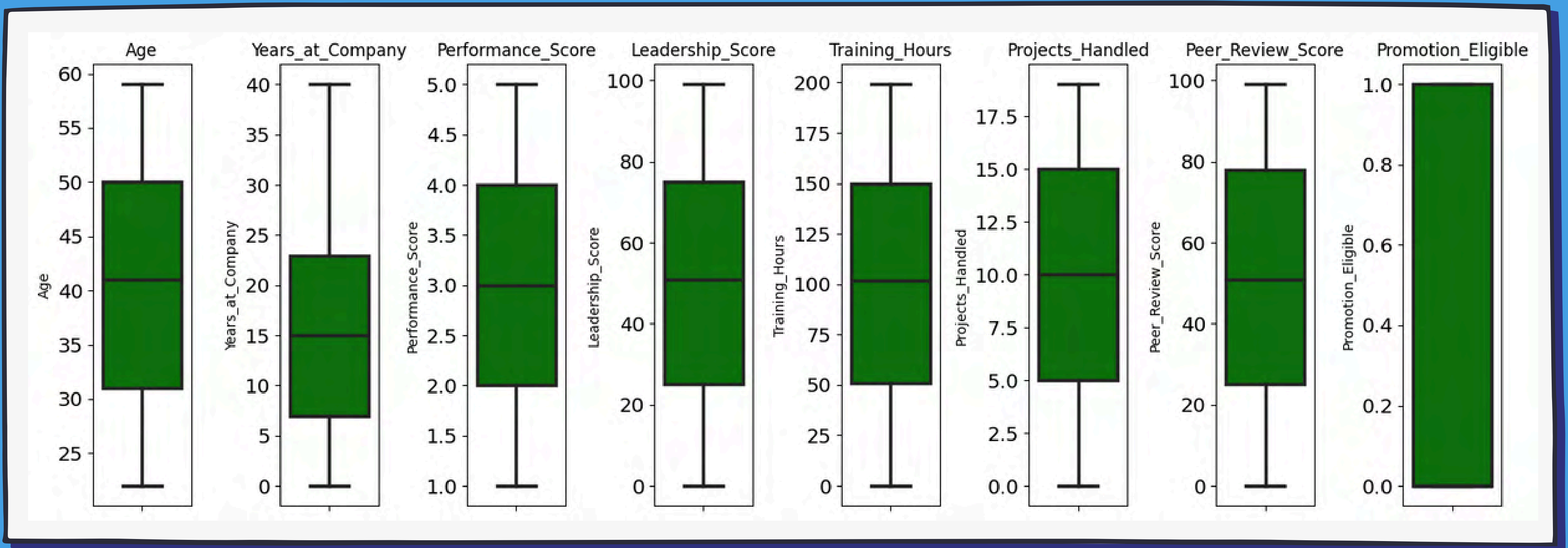


- The majority of employees have a working period of between 5 and 30 years.
- Shows a Multimodal distribution that is fairly regularly spaced between the values 1 to 6.
- Employee's leadership skills are average (50%)
- The distribution of most columns are relatively normal, except Training_Hours.
- Age range tends to be adult to senior (> 40)
- Has two clear peaks, one around 0 to 5 projects and another around 15 to 20 projects
- The distribution is quite even and wide, with the highest concentration 30-100
- 29% of employees are eligible for promotion

Source: Rakamin's Talent Promotion dataset

Data Cleaning

To fix missing values, we imputed all numeric features using the median and all categorical features using the mode, resulting in zero missing values across the dataset. This process also helped the overall distribution become more normal.



	Feature	Missing Sebelum	Missing Sesudah
0	Employee_ID	0	0
1	Age	48	0
2	Years_at_Company	49	0
3	Performance_Score	50	0
4	Leadership_Score	49	0
5	Training_Hours	49	0
6	Projects_Handled	50	0
7	Peer_Review_Score	50	0
8	Current_Position_Level	50	0
9	Promotion_Eligible	49	0
10	TOTAL	444	0

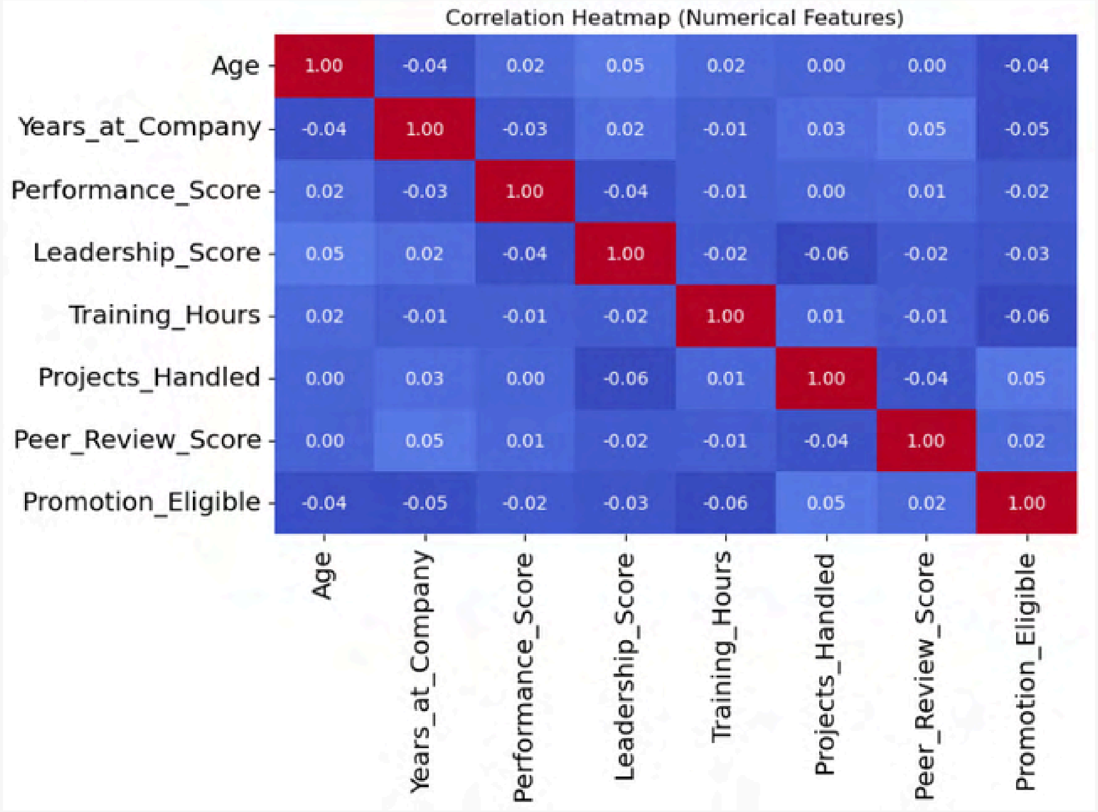
```
Data columns (total 9 columns):
 #   Column                  Non-Null Count
---  ------                  --------------
 0   Employee_ID             1000 non-null
 1   Age                     950 non-null
 2   Performance_Score       950 non-null
 3   Leadership_Score        950 non-null
 4   Training_Hours          950 non-null
 5   Projects_Handled        950 non-null
 6   Peer_Review_Score       950 non-null
 7   Current_Position_Level  950 non-null
 8   Promotion_Eligible      950 non-null
dtypes: float64(7), object(2)
```

We removed approximately 13 (1.3%) outliers of data in the Age and Training Hours columns so that the data distribution would be more normal and would not affect the model's accuracy. We also removed the Years at Company column because there were many errors, causing the column to be biased.

Source: Rakamin's Talent Promotion dataset

Feature Selection Strategy

Our feature selection strategy involved dropping features that did not contribute to prediction, those with unreliable data, and those that did not reflect employee performance.



Dropped

- **Employee_Id:** Has no effect
- **Age :** Not reflecting employee quality
- **Years_at_Company:** Lots of data are implausible (bigger than Age)
- **Current_Position_Level:** Does not reflect quality

Used

- **Performance_Score:** Employee performance indicators
- **Training_Score:** Employee motivation and knowledge indicators
- **Peer_Review_Score:** Indicators of teamwork and communication skills.
- **Project_Handled:** Indicators of work experience and company trust in employees.
- **Leadership_Score:** Leadership quality indicators.
- **Promotion_Eligible:** ML target

Source: Rakamin's Talent Promotion dataset

Feature Engineering

To reduce bias and improve the accuracy of employee assessment, we created two new features that are better at capturing performance-related signals.

New features

Leadership_Index

$$\frac{\text{(Leadership\_Score + Peer\_Review\_Score)}}{2}$$

These features can be a signal for leadership quality, as well as how much people value certain employee.

Perfomance_Index

(Performance_Score + Project_Handled + Training_Hours) / 3

These features shows talent's quality, experience, also knowledge and motivation. This can be an indicator for talent's potential.

Potential_Index

(0.4 x Training_Hours_scaled) + (0.4 x Peer_Review_Score_scaled) + (0.2 x Leadership_Score_scaled)

This feature can show employee's motivation, and also how people sees them in terms of quality.

Growth_Momentum

Projects_Handled_scaled / (Training_Hours_scaled + 1)

Captures the ratio of projects managed to training, indicating how quickly skills are applied into results.

Leadership_Influence

Peer_Review_Score_scaled / (Leadership_Score_scaled + 1)

This feature can show how people sees and trust the employee.

Performance_Consistency

Performance_Score x Projects_Handled_scaled

This feature might signal the employee's work quality plus consistency over period of times.

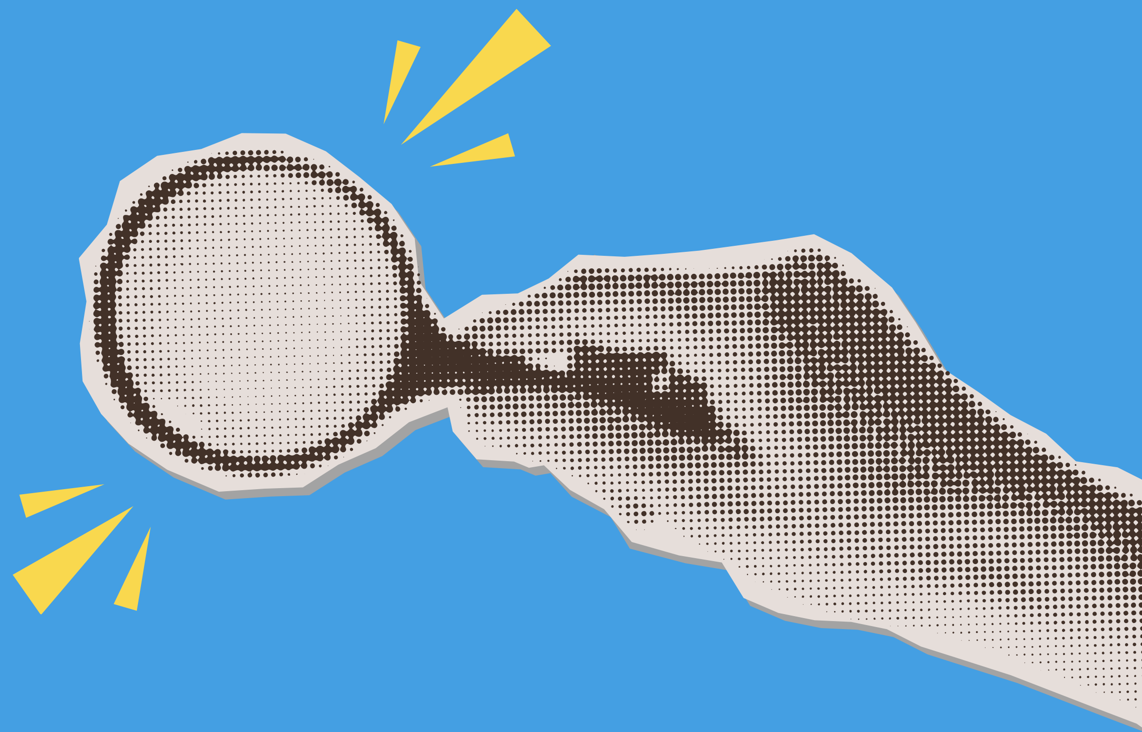
Data Preprocessing

Performance_Score
Performance_Consistency
Project_Handled
Leadership_Influence
Peer_Review_Score
Growth_Momentum
Leadership_Score
Perfomance_Index
Training_Score
Potential_Index
Leadership_Index

We applied standardization to all numeric features to keep the scaling consistent across the dataset. Since all features, including Project_Handled, showed sufficiently normal distributions, standardization was the most appropriate and stable preprocessing choice.

Promotion_Eligible

We kept the target variable in its original binary form because it already represents a clear classification outcome, and transforming it would not provide additional predictive value.

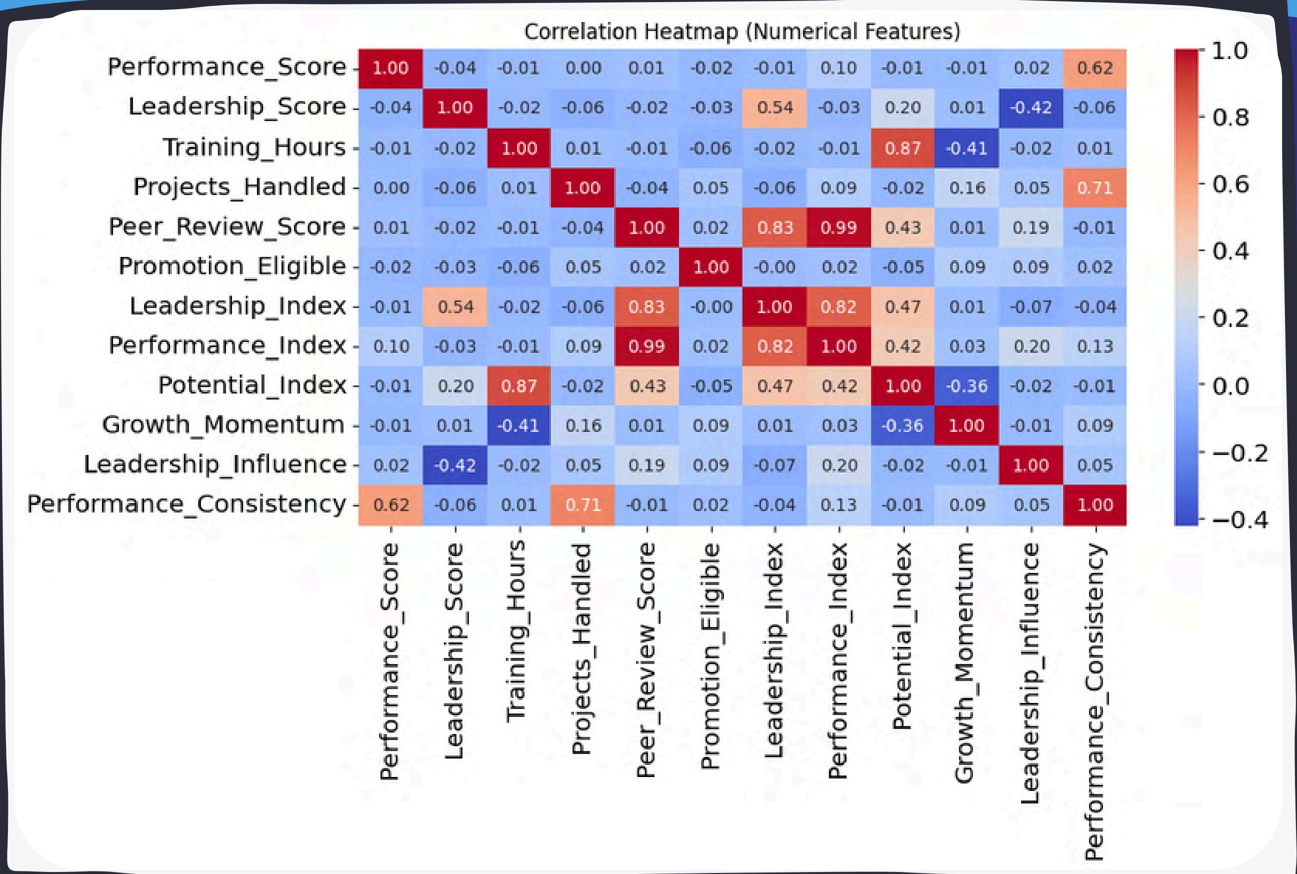


Source: Rakamin's Talent Promotion dataset

Data Preprocessing

Performance_Score	Leadership_Score	Training_Hours	Projects_Handled	Peer_Review_Score	Promotion_Eligible	Leadership_Index	Performance_Index	Potential_Index	Growth_Momentum	Leadership_Influence	Performance_Consistency
-0.701018	-0.220184	-1.035977	1.627892	0.349011	0.0	0.172393	0.499204	-0.805039	0.260405	-0.175344	0.357239
1.465699	-0.184340	-0.610027	0.178266	-1.591787	0.0	-1.443855	-1.437583	-1.281323	-0.135183	-0.363274	1.096978
1.465699	-1.116279	-0.077591	-0.727750	1.231192	0.0	0.420284	1.245916	0.233415	-0.280955	0.276325	0.009126
0.743460	0.030723	-1.497422	-0.727750	0.031426	0.0	0.043490	-0.002494	-1.296939	0.133730	-0.229885	-0.251958
0.743460	0.496693	-0.769758	1.084282	0.278437	0.0	0.509525	0.475869	-0.445875	0.023922	-0.244045	1.488605

Dataset has been cleaned and pre-processed, ready for ML



Correlation map with new features, ready for testing and tuning.

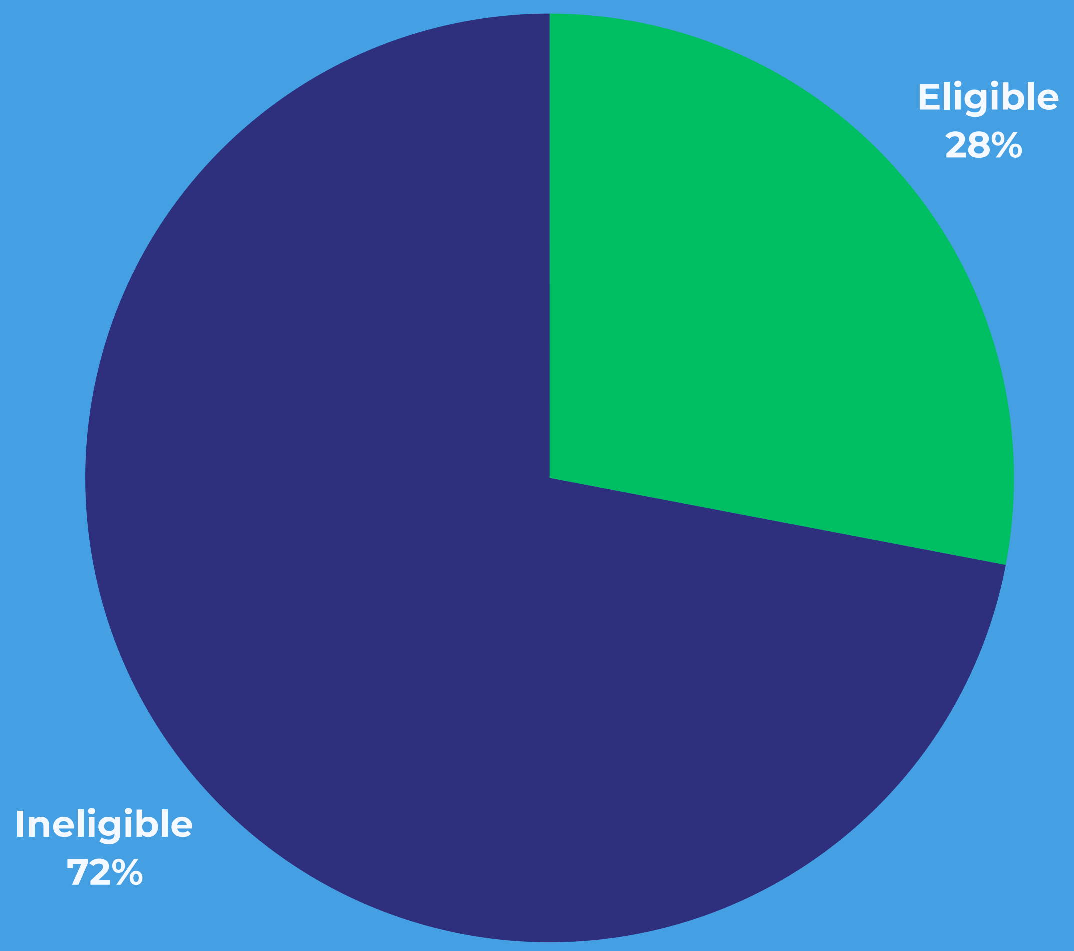
Source: Rakamin's Talent Promotion dataset

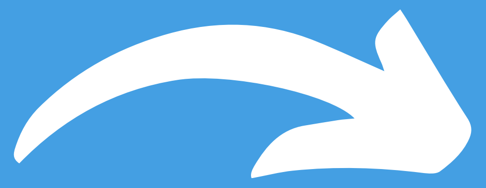
Imbalance Target

SMOTE helps balance the amount of data between Eligible and Ineligible employees.

Before

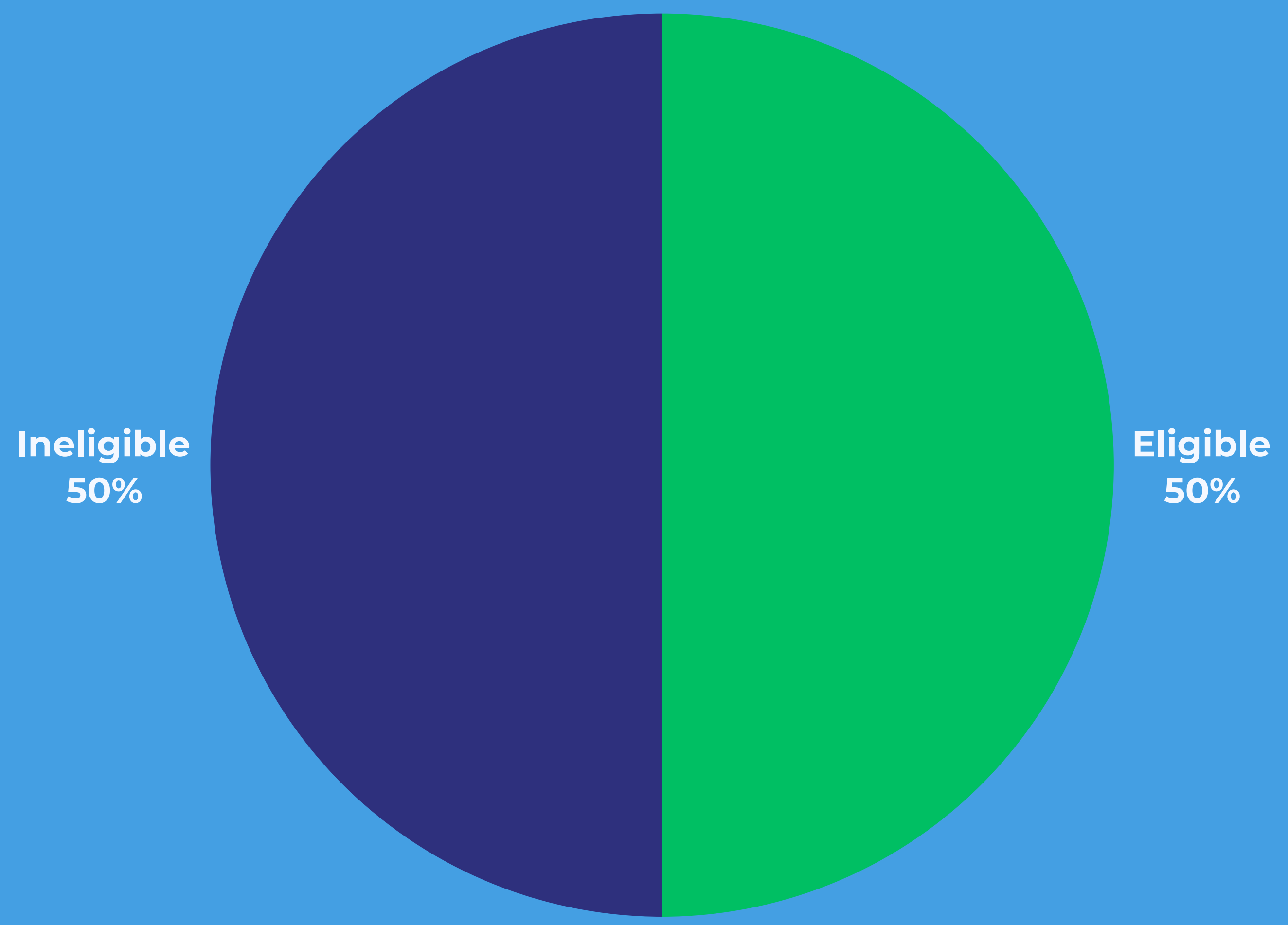
Proportion of promotion eligibility
Eligible vs Ineligible





After

Proportion of promotion eligibility
Eligible vs Ineligible



Stage 2: Modeling.

Evaluation Metrics

Unsupervised: Clustering

Silhouette Score
The dataset is continuous and overlapping so silhouette helps detect whether the model is forcing fake clusters or capturing real structure.

Davies–Bouldin Index (DBI)
It's sensitive to cluster overlap, exactly the case in human performance data. When silhouette is ambiguous, DBI offers a second opinion that often reveals hidden issues.

Supervised: Tree Based & Linier Model

ROC-AUC
ROC-AUC measures the model's ability to rank positive vs. negative cases across all thresholds. It is stable under class imbalance and allows fair comparison between tree-based and linear models.

F1-Score
The dataset is imbalanced. F1 balances precision and recall into a single score, preventing the model from achieving artificially high performance by predicting only the majority class.

Confusion Matrix
HR needs to see the number of false positives and false negatives clearly. The confusion matrix provides transparent insight into error types and is essential for decision-makers.

Recall
Promotion is a high-stakes positive class. Missing truly eligible employees (false negatives) is far more harmful than incorrectly flagging some non-eligible ones. Recall ensures the model captures as many genuinely promotable talents as possible.

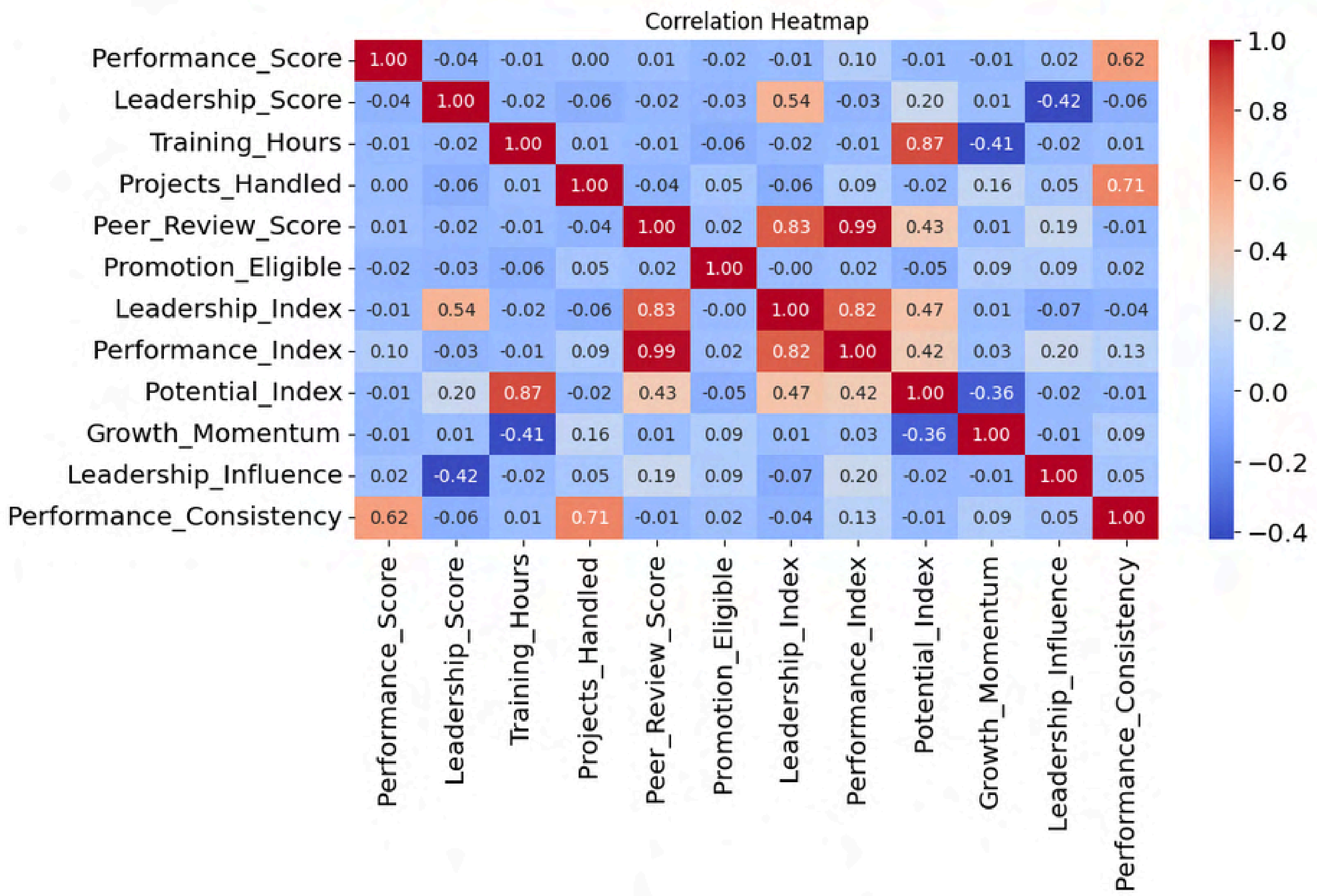
Precision
HR operational cost matters. When the model predicts "eligible," HR wants that prediction to be trustworthy. Precision reduces wasted resources on false promotion recommendations.

Tree-based.

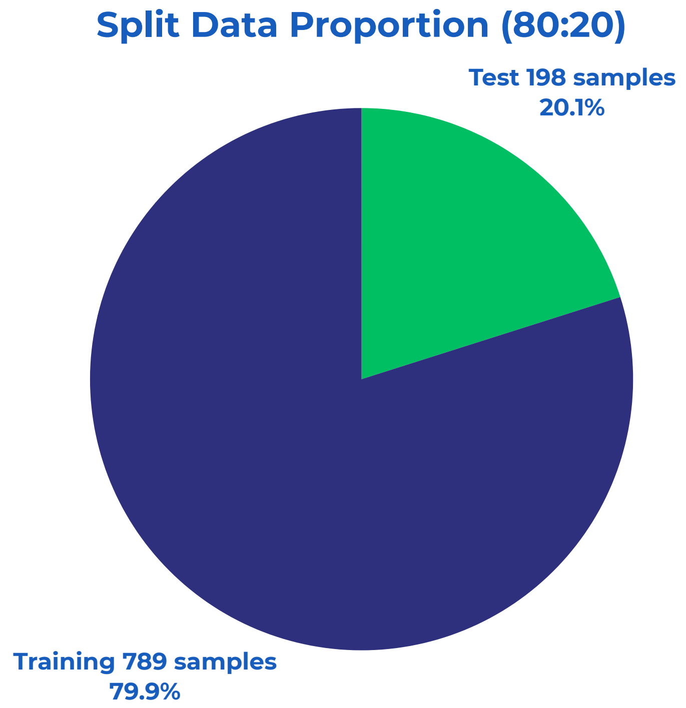
1. Models: (Decision Tree, Random Forest, XGBoost)
2. Evaluate features correlation
3. Split train-test data
4. Cross-validation
5. Fit data into model and train
6. Model evaluation (F1 Score, Accuracy, Precission, Recall, ROC-AUC, Confusion Metrics)
7. Hyperparameter tuning
8. Model re-evaluation

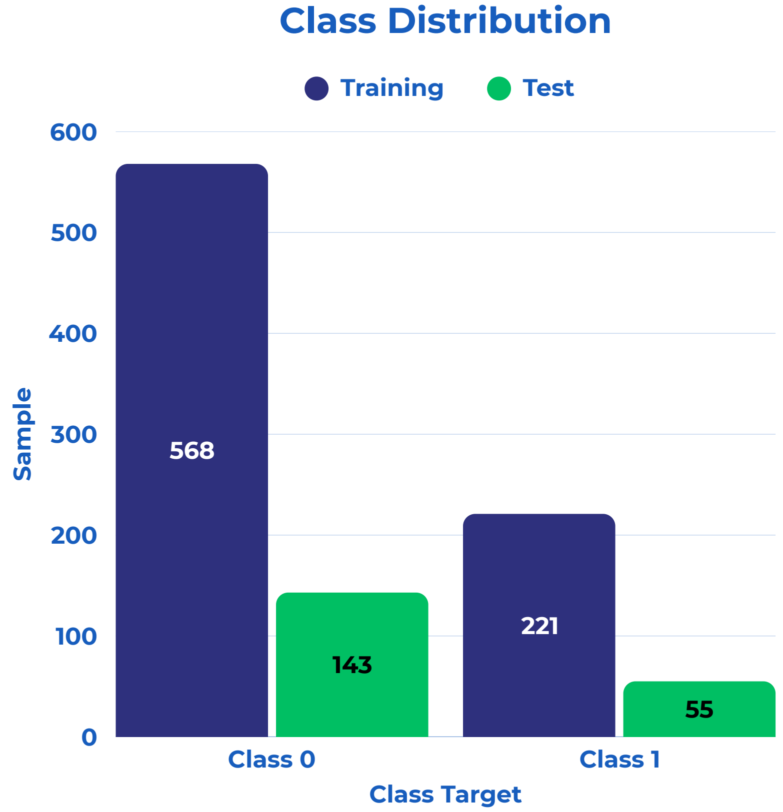
Correlation & Split Data

All Feature



All Feature





- Highly correlated features are retained, as tree-based models are not affected by multicollinearity.
- The data is split 80/20 to preserve representative training and evaluation.
- The class distribution is still proportional, and tree-based models can handle minor imbalance without resampling.

Decision Tree

Parameter
Decision Tree
criterion = "gini"
splitter = "best"
max_depth = None
min_samples_split = 2
min_samples_leaf = 1
class_weight = None

Evaluation metrics

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
100%	32%	100%	31%	100%	33%	100%	52%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	587	0
Actual 1	0	221

Test	Pred 0	Pred 1
Actual 0	103	40
Actual 1	37	18

- **Model Performance Differs Significantly Between Training and Test Data, Indicating Overfitting.**
- **Model at High Risk of Missing Eligible Employees**

Decision Tree

Evaluation metrics : Tuned Decision Tree

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
24%	17%	51%	27%	15%	13%	59%	48%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	535	33
Actual 1	187	34

Test	Pred 0	Pred 1
Actual 0	124	19
Actual 1	48	7

Best Parameter

Tuned Decision Tree
'class_weight': 'balanced'
'criterion': 'entropy'
'max_depth': 3
‘min_samples_lead’: 5
‘min_samples_split’: 2
cross validation = 10

- **The model demonstrates a strong balance of performance and stability across test data.**
- **The model successfully reduces the risk of missing qualified employees**

Random Forest

Parameter
Random Forest
n_estimators = 100
criterion = "gini"
max_depth = None
min_samples_split = 2
min_samples_leaf = 1
cross validation : 10

Evaluation metrics

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
100%	11%	100%	22%	100%	0.7%	100%	55%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	568	0
Actual 1	0	221

Test	Pred 0	Pred 1
Actual 0	129	14
Actual 1	51	4

The Random Forest model shows very high performance on the Train data but suffers from overfitting, as its performance drops on the Test data.

Random Forest

Evaluation metrics : Tuned Random Forest

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
65%	36%	54%	30%	83%	45%	86%	54%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	412	156
Actual 1	38	183

Test	Pred 0	Pred 1
Actual 0	85	58
Actual 1	30	25

Best Parameter

Tuned Random Forest
'n_estimators': 100
'model__max_depth': 5
'min_samples_split': 2
'min_samples_leaf': 1
'boostrap': 'False'
cross validation : 10

Demonstrated excellent and stable performance with an F1 score of 87%. Parameter tuning successfully addressed the previously observed overfitting.

XGBoost

Parameter
XGBoost
n_estimators = 100
learning_rate = 0.1
max_depth = 6
subsample = 1.0
colsample_bytree = 1.0
cross validation : 10

Evaluation metrics

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
100%	20%	100%	26%	100%	16%	100%	50%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	568	0
Actual 1	0	221

Test	Pred 0	Pred 1
Actual 0	118	25
Actual 1	46	9

The Model Delivers Very Powerful and Balanced Performance

XGBoost

Evaluation metrics : Tuned XGBoost

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
90%	20%	100%	26%	100%	16%	100%	50%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	568	0
Actual 1	0	221

Test	Pred 0	Pred 1
Actual 0	118	25
Actual 1	46	9

Tuning Parameter

Tuned XGBoost
‘subsample’: 0.7
‘max_depth’: 3
‘learning_rate’: 0.01
'gamma’: 0
'n_estimators’: 400
cross validation : 10

Tuning XGBoost does not improve performance, so the default parameters are sufficient for this data.

Other model

Model	Train Accuracy	Test Accuracy	Train Precision (macro)	Test Precision (macro)	Train Recall (macro)	Test Recall (macro)	Train F1 Score (macro)	Test F1 Score (macro)	ROC-AUC Train (ovr)	ROC-AUC Test (ovr)
Logistic Regression	0.997101	0.993266	0.994294	0.996063	0.994294	0.977778	0.994294	0.986660	0.997850	0.998413
SVM	0.973913	0.966330	0.927696	0.917504	0.980665	0.961905	0.951687	0.937877	0.999090	0.994621
Naive Bayes	0.868116	0.855219	0.758291	0.741062	0.886464	0.850794	0.795591	0.774065	0.955674	0.918783
Decision Tree	0.886957	0.875421	0.783938	0.769528	0.929558	0.899206	0.826183	0.807995	0.968674	0.932363
Random Forest	0.976812	0.932660	0.935025	0.858321	0.982369	0.896429	0.956740	0.875753	0.992797	0.977160
XGBoost	0.943478	0.909091	0.863377	0.811810	0.962778	0.928175	0.902603	0.852464	0.991631	0.984303

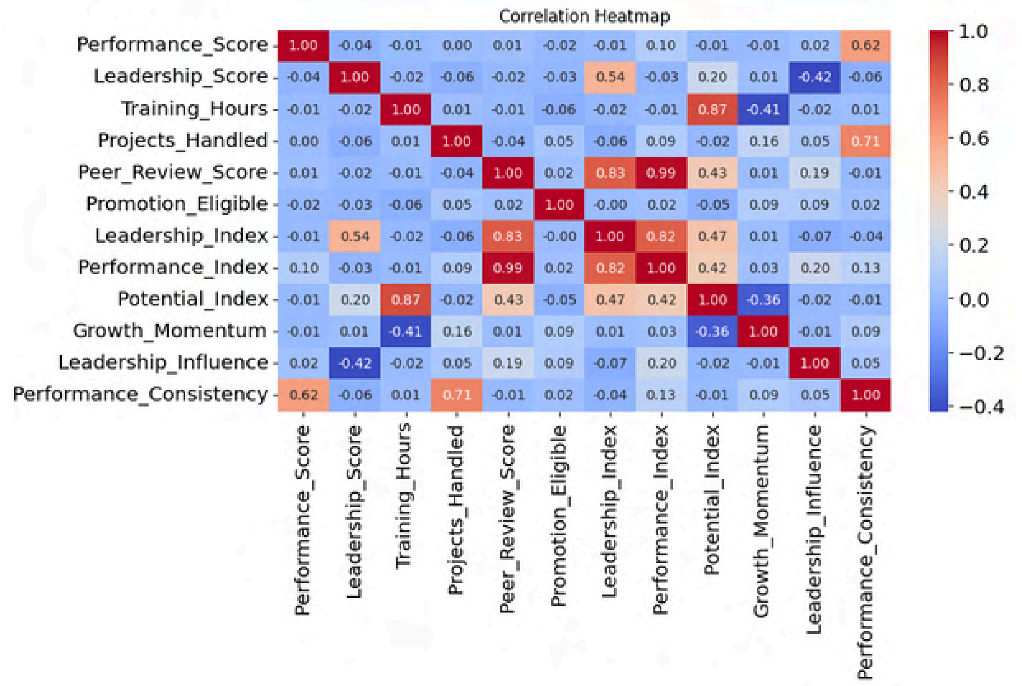
Gaussian Naive Bayes shows decent accuracy, but its performance stays limited because it can't model more complex feature patterns.

Linear model.

1. Linear model: Logistic Regression, SVM
2. Evaluate features correlation
3. Split train-test data and handle imbalance target with SMOTE
4. Cross-validation
5. Fit data into model and train
6. Model evaluation (F1 Score, Accuracy, Precission, Recall, ROC-AUC, Confusion Metrics)
7. Hyperparameter tuning
8. Model re-evaluation

Drop Features

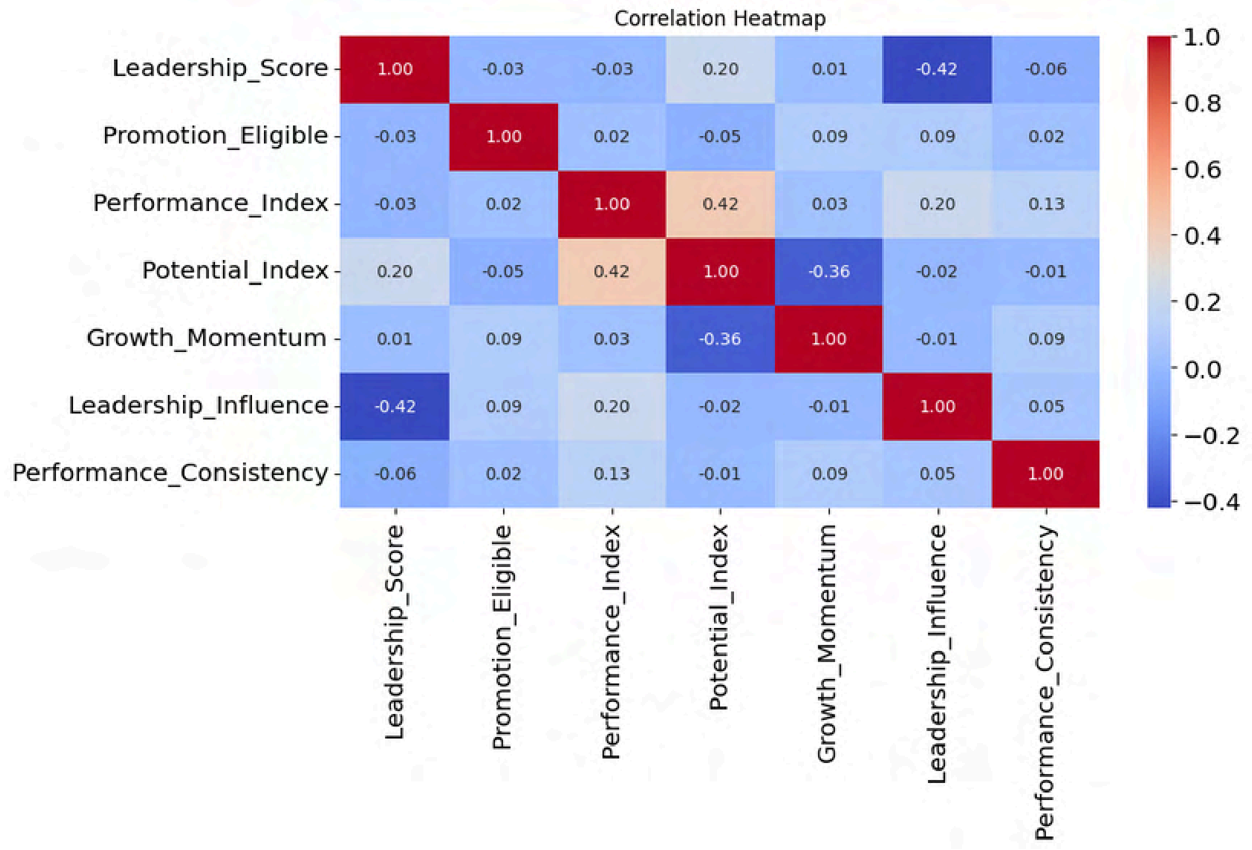
Full Features



Drop some features that have high correlation:

- Performance_Score
- Training_Hours
- Project__Handled
- Peer_Review_Score

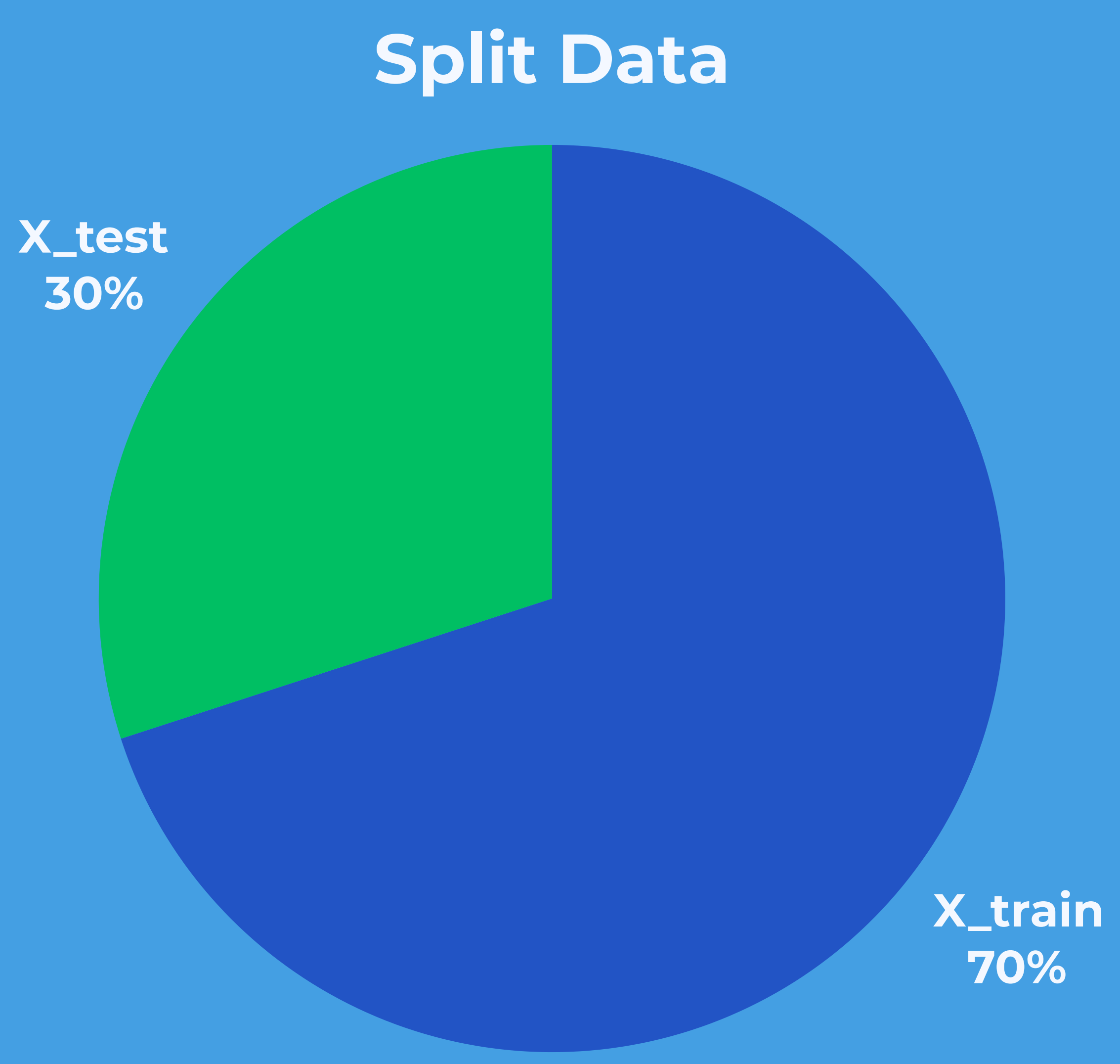
Selected Features

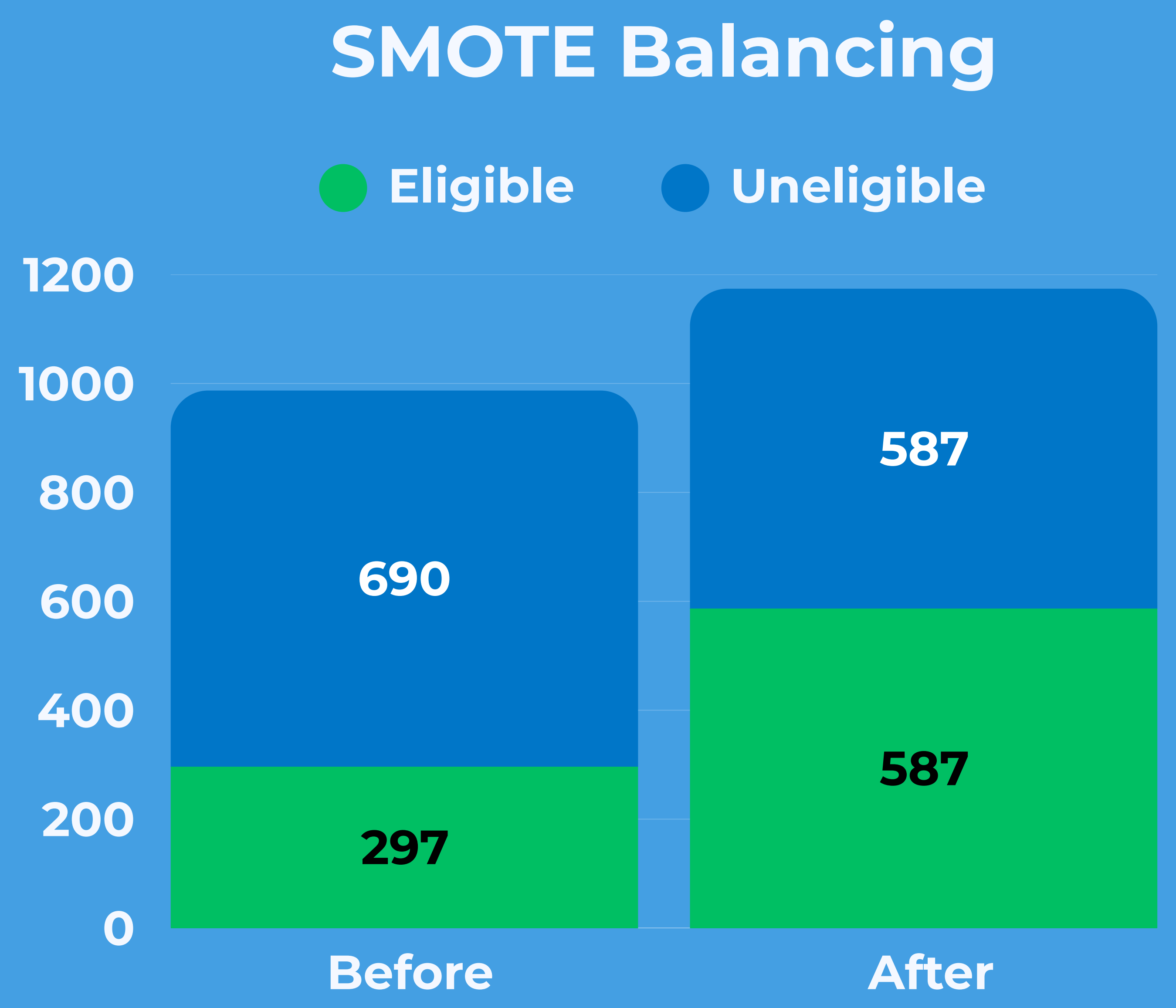


Cross-validation

```
skf = StratifiedKFold(
    n_splits=10,
    shuffle=True,
    random_state=42
)
```

Split Data & Balancing





- The data is split 70/30 to preserve representative training and evaluation.
- Handling class imbalance with SMOTE to prevent biases.

Logistic Regression

Parameter
Logistic Regression
penalty='l2'
dual=False
max_iter=100
multi_class='auto'
verbose=0
cross validation : 10

Evaluation metrics

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)
54%	51%	54%	52%	53%	50%	54%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	275	222
Actual 1	233	264

Test	Pred 0	Pred 1
Actual 0	137	77
Actual 1	45	38

Initial testing has relatively low recall. The model also has slight overfitting issues.

Logistic Regression

Evaluation metrics : Tuned Logistic Regression

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
38%	37%	30%	30%	52%	47%	56%	52%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	265	232
Actual 1	92	101

Test	Pred 0	Pred 1
Actual 0	125	89
Actual 1	44	39

Tuning Parameter

Tuned Logistic Regression
‘model_C’ : 100
‘model_class_weight’ : none
‘model_penalty’ : l2
'model_solver' : ‘lbgfs’
multi_class='auto'
cross validation : 10

After tuning, the overall train and score is slightly lower. However, the model did not overfit anymore.

SVM

Parameter
SVM
‘C’: 1.0
‘kernel’: ‘rbf
'max_iter’: 1
'shringking’: ‘True’
‘verbose’: ‘False’
cross validation : 10

Evaluation metrics

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)
73%	67%	66%	59%	83%	76%	66%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	278	219
Actual 1	79	418

Test	Pred 0	Pred 1
Actual 0	100	114
Actual 1	41	42

SVM did not show promising result at initial test. Overall, it has the lowest score compared to other linear model.

SVM

Evaluation metrics : Tuned SVM

F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)
47%	34%	34%	25%	73%	52%	61%	45%

Confusion metrics

Train	Pred 0	Pred 1
Actual 0	227	270
Actual 1	51	142

Test	Pred 0	Pred 1
Actual 0	87	127
Actual 1	40	43

Tuning Parameter

SVM
‘C’: 1.0
‘kernel’: ‘rbf
'max_iter’: 1
'shringking’: ‘True’
‘verbose’: ‘False’
cross validation : 10

Has a good score metrics but didn’t as great as Logistic Regression

Unsupervised.

1. Clustering model: K-Means, GMM, K-Medoids
2. Find correlation between features
3. Testing 2 different scaled datasets, full features and selected features
4. Testing different K-numbers using elbow method
5. Evaluate model using Silhouette Score and DBI Score
6. Visualisation using PCA and radar chart

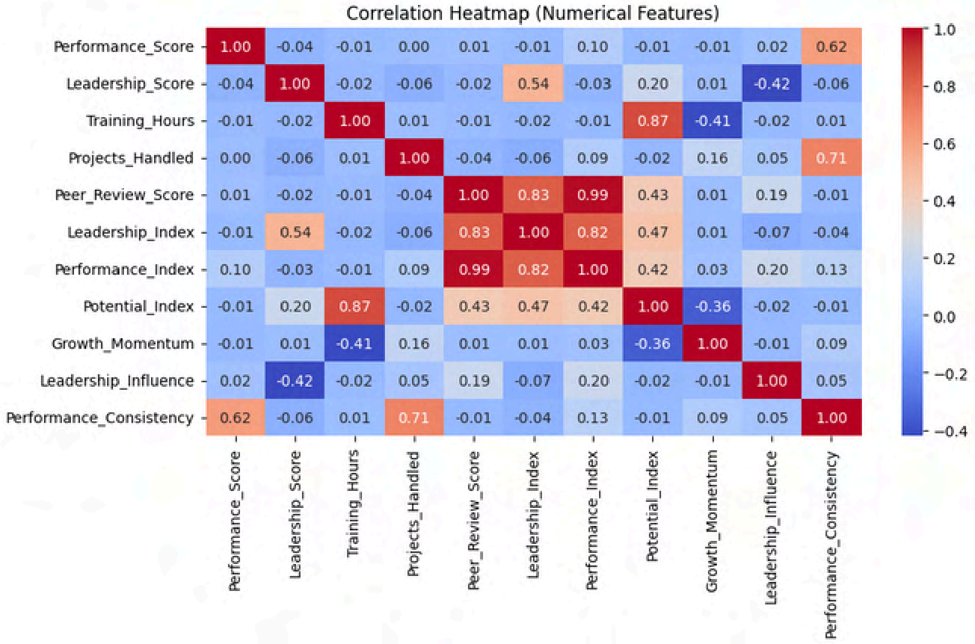
K-Means

In **df_scaled (full features)**, there are several features that has very high correlation.

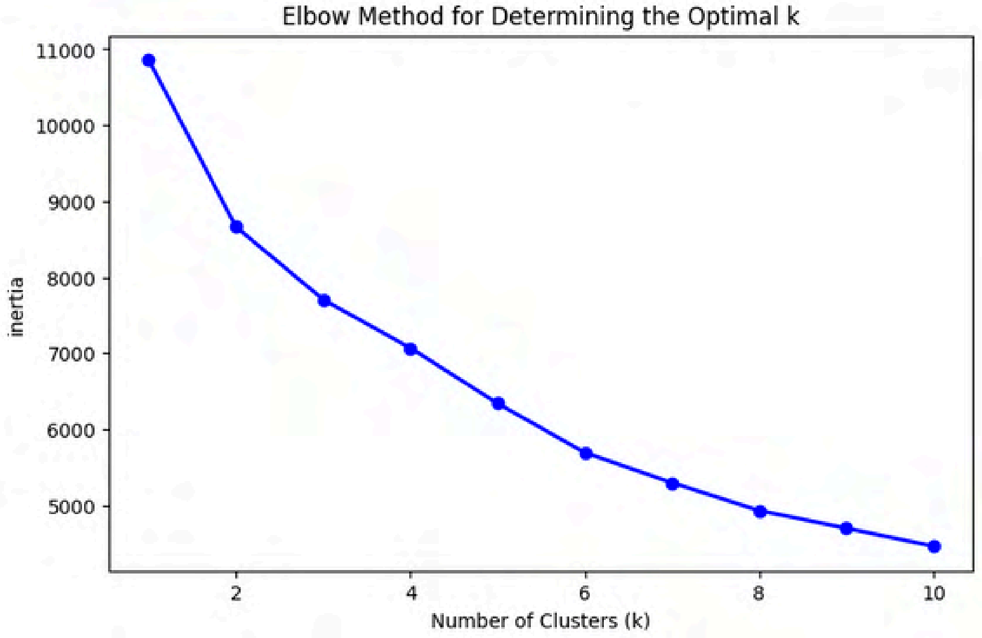
In **df_scaled2 (selected features)**, we drop these features:

- Performance_Score
- Leadership_Score
- Training_Hours
- Projects_Handled
- Growth_Momentum
- Peer_Review_Score
- Leadership_Influence
- Performance_Consistency

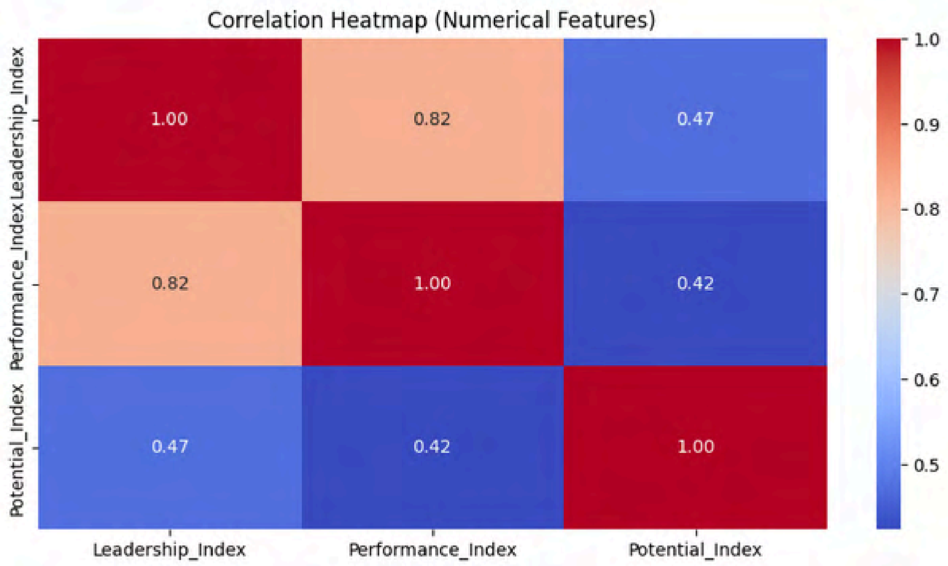
df_scaled - full features



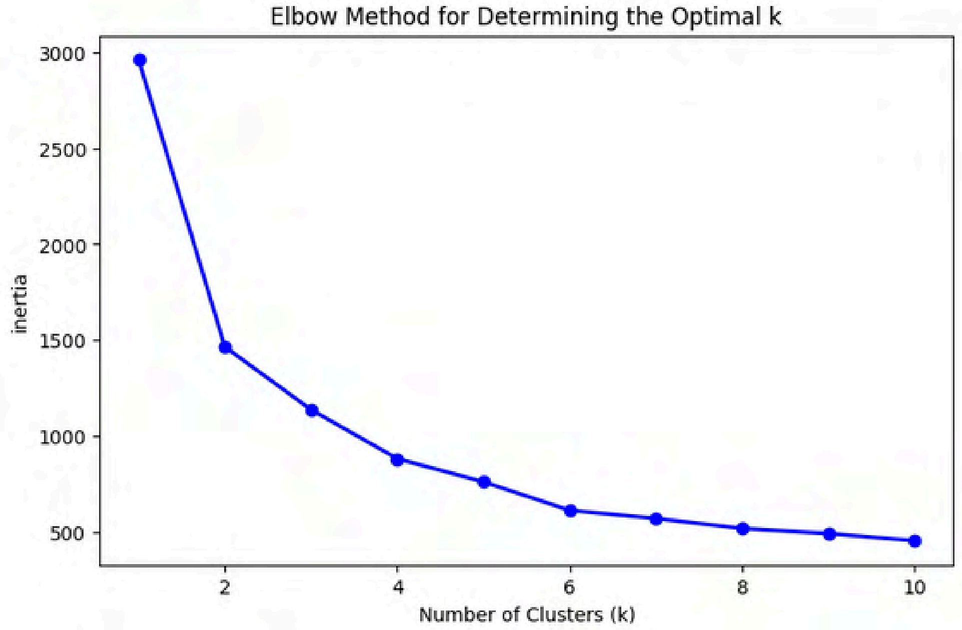
Elbow method evaluation using full features



df_scaled2 - selected features



Elbow method evaluation selected features



K-Means

Low Silhouette and DBI Score.

Based on the Silhouette & DBI score, the best dataset to be used is df_scaled2 (selected features) with K = 4. Thus, this dataset will be used to test other model as well.

df_scaled (full features)

k	Silhouette	DBI
2	0.202028	1.828185
3	0.187135	1.834187
4	0.194977	1.521013
5	0.197892	1.334550
6	0.182993	1.360843
7	0.170787	1.433211
8	0.165625	1.396923
9	0.167619	1.365354
10	0.169514	1.365004

df_scaled2 (selected features)

k	Silhouette	DBI
2	0.407040	0.931791
3	0.299720	1.180320
4	0.320992	1.000364
5	0.312454	1.033581
6	0.310237	0.965258
7	0.282720	1.039299
8	0.271177	1.063737
9	0.281358	1.049420
10	0.285985	1.054103

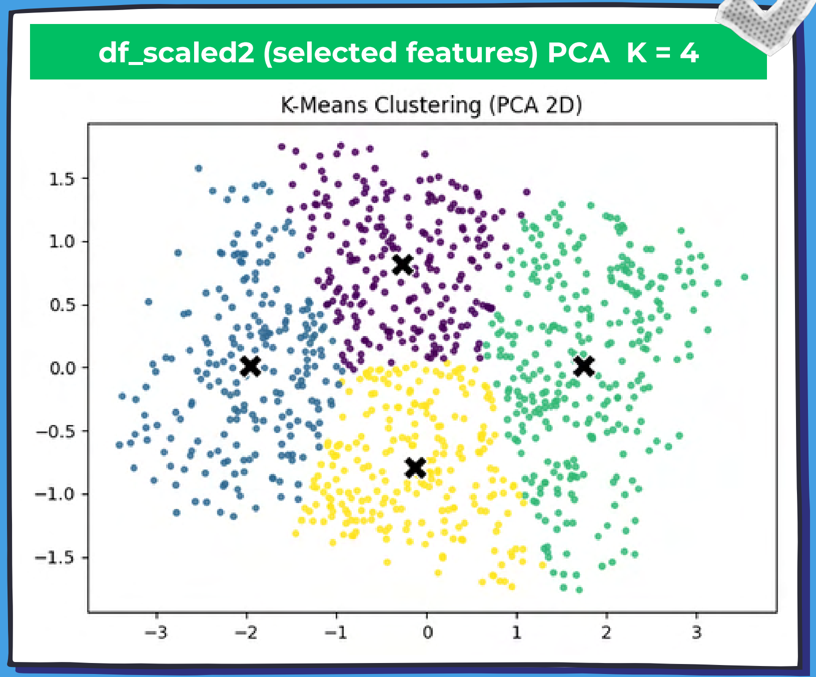
K-Means

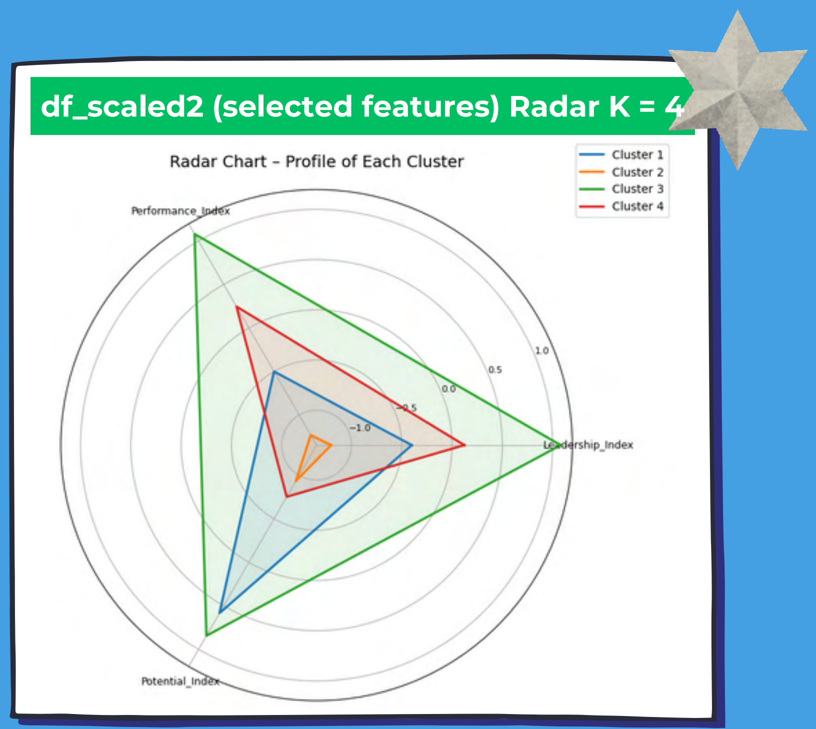
Model Interpretation

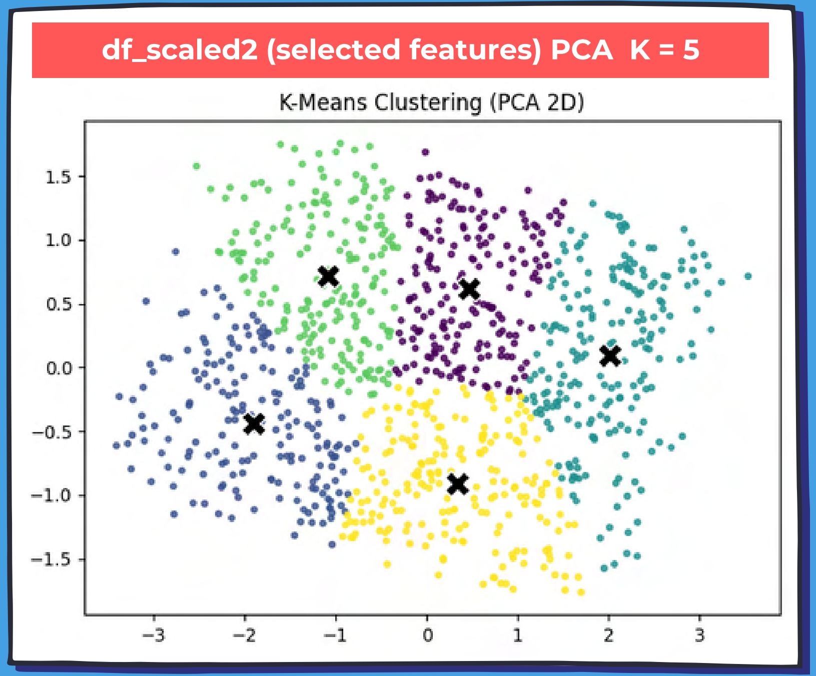
Clustering was performed using K-Means on standardized numerical features. PCA was applied for visual inspection, and radar charts were used to understand each cluster's directional behaviour across leadership, performance, potential, and consistency.

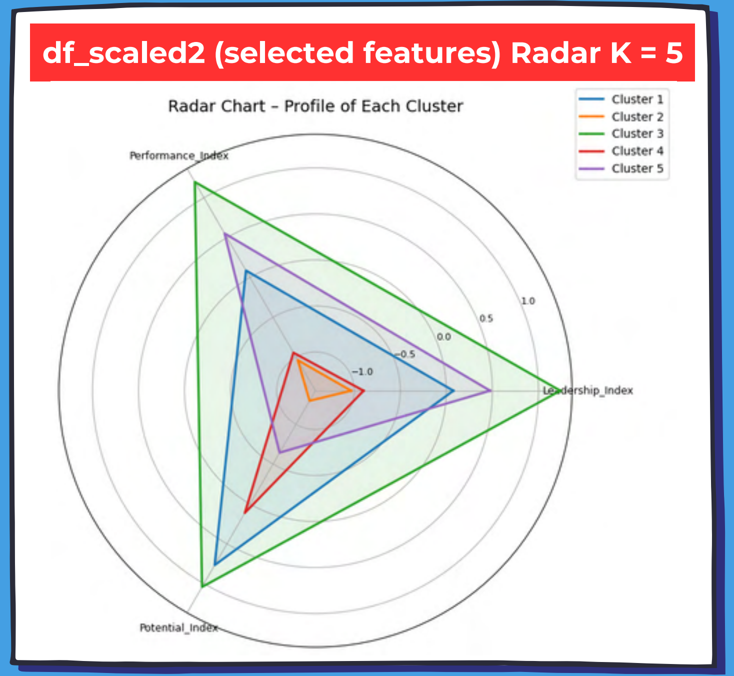
Key Observations

- PCA visuals show partially overlapping groups, a common pattern in HR datasets with continuous performance trait, but each cluster still forms a distinct center.
- Radar charts highlight clear separations in leadership quality, performance strength, potential level, and work consistency.
- Both K=4 and K=5 produce meaningful and interpretable segments; K=5 provides more granularity, while K=4 balances clarity and business usability.
- Silhouette and DBI metrics show slightly better results at K=5, but K=4 remains highly coherent and aligns well with practical HR personas.





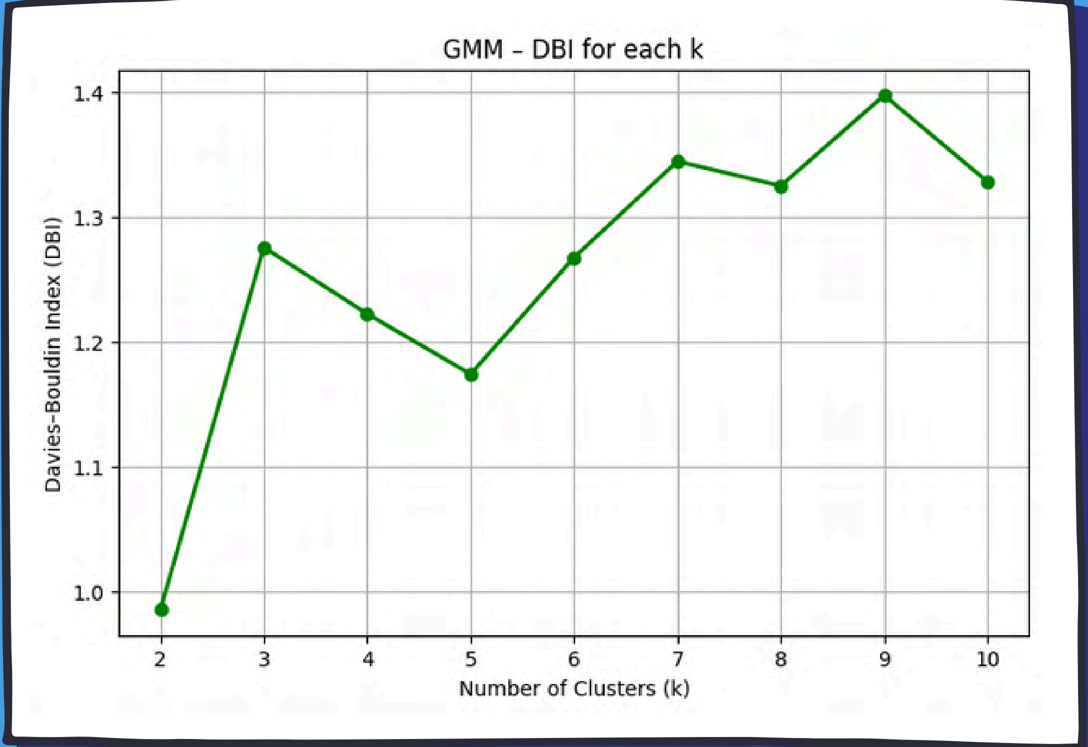




GMM

GMM produces less distinct clusters. Low silhouette and high DBI indicate that the cluster structure in this dataset is not suitable for GMM.

k	Silhouette	DBI
2	0.378726	0.985545
3	0.259586	1.275267
4	0.241541	1.222058
5	0.247479	1.173847
6	0.203150	1.267111
7	0.157913	1.344091
8	0.168369	1.324435
9	0.167572	1.396875
10	0.188466	1.327711



GMM

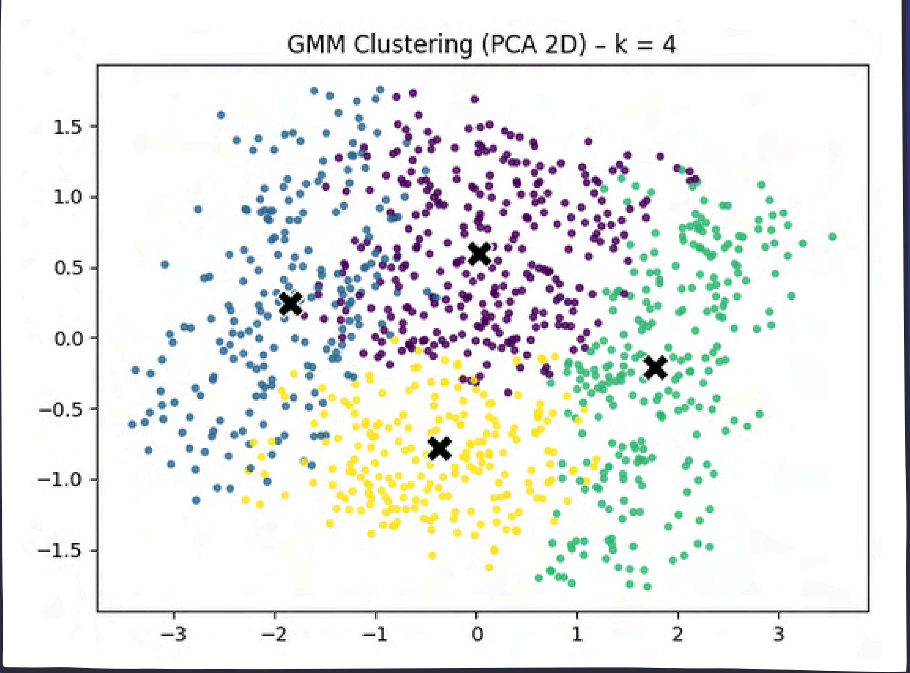
Model Interpretation

Overview: The Gaussian Mixture Model (GMM) was applied using probabilistic soft-clustering with full covariance matrices. It relaxes the spherical-cluster assumption of K-Means/K-Medoids and can model elliptical and overlapping cluster shapes

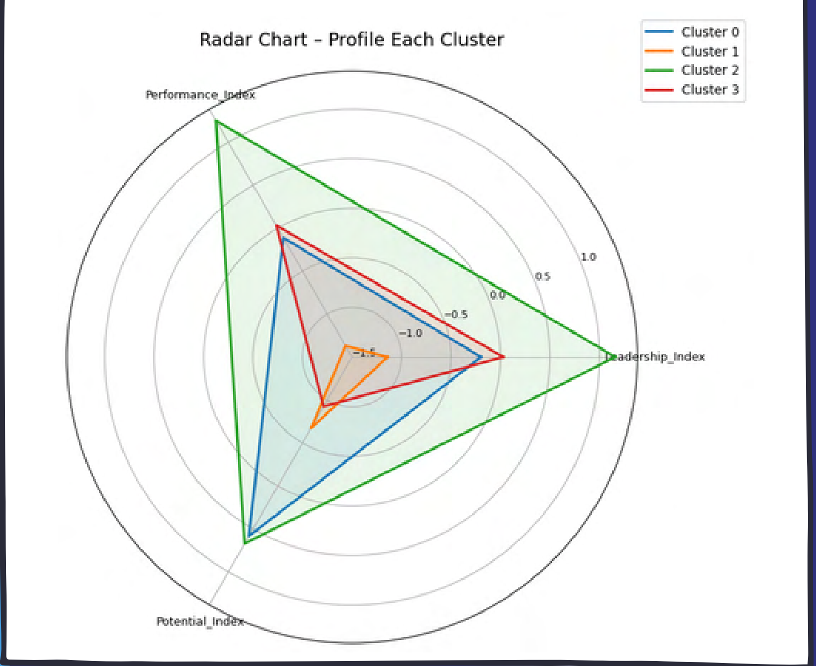
Key Observations

- The Silhouette Score is very low (−0.01–0.13) and the DBI is high (2.2–3.3), indicating poor cluster quality and significant overlap.
- PCA visualization (k=4 & k=5) shows clusters that are highly overlapping, with centroids close to the center of the data and cluster shapes that are barely formed.
- The radar chart shows almost identical patterns between clusters, indicating that there are no major distinguishing features and that the variation is merely noise.
- Overall, GMM fails to find a clear cluster structure and performs weaker than K-Means and K-Medoids; the dataset does not have a pattern suitable for probabilistic approaches such as GMM.

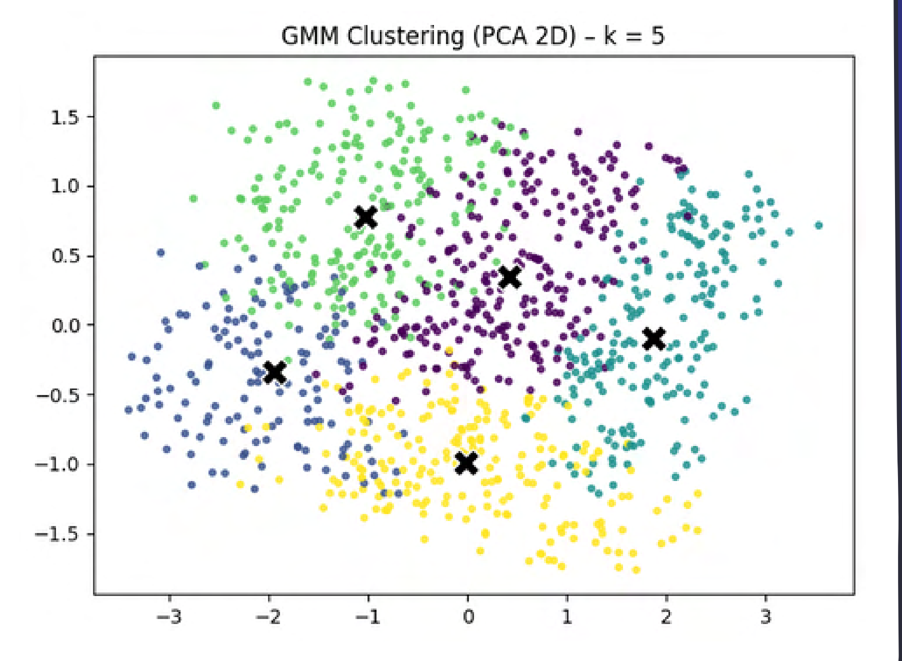
df_scaled2 (selected features) PCA K = 4



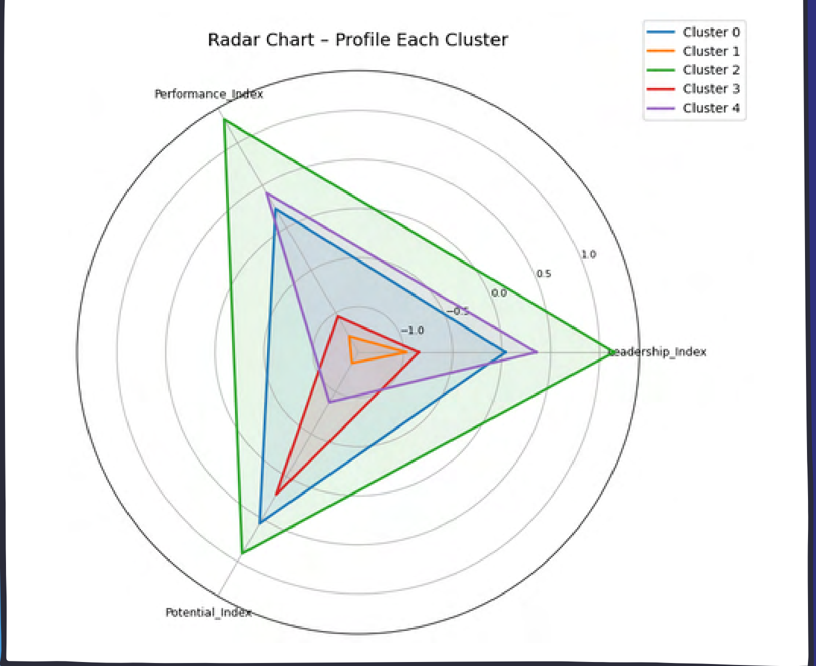
df_scaled2 (selected features) Radar K = 4



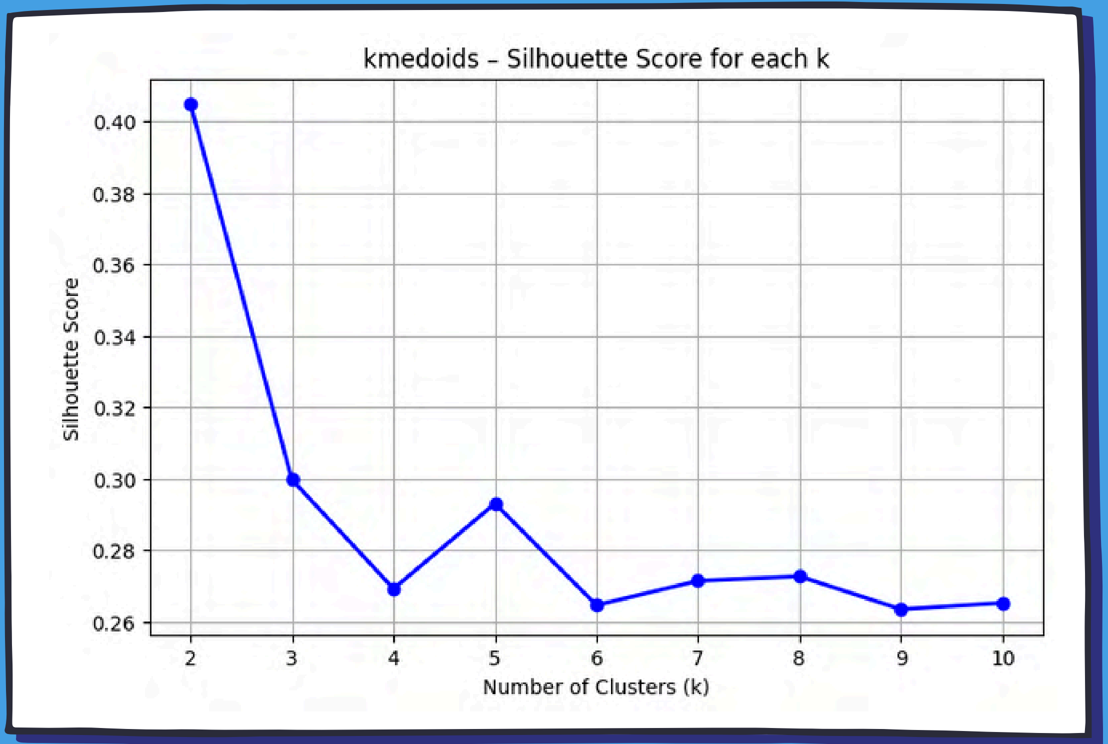
df_scaled2 (selected features) PCA K = 5



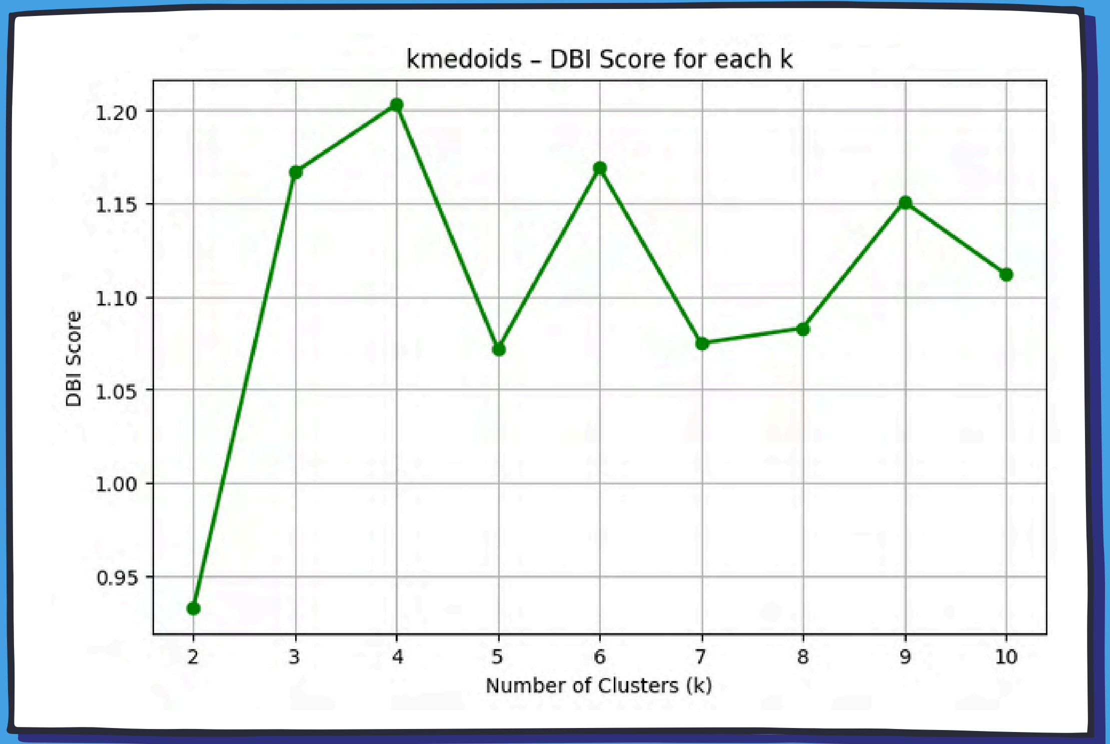
df_scaled2 (selected features) Radar K = 5



K-Medoids



k	Silhouette	DBI
2	0.404823	0.932629
3	0.299738	1.166699
4	0.269244	1.203232
5	0.293064	1.071740
6	0.264633	1.169160
7	0.271444	1.074950
8	0.272696	1.083045
9	0.263472	1.150987
10	0.265270	1.112114



Both Silhouette and DBI score did not show better results than K-Means

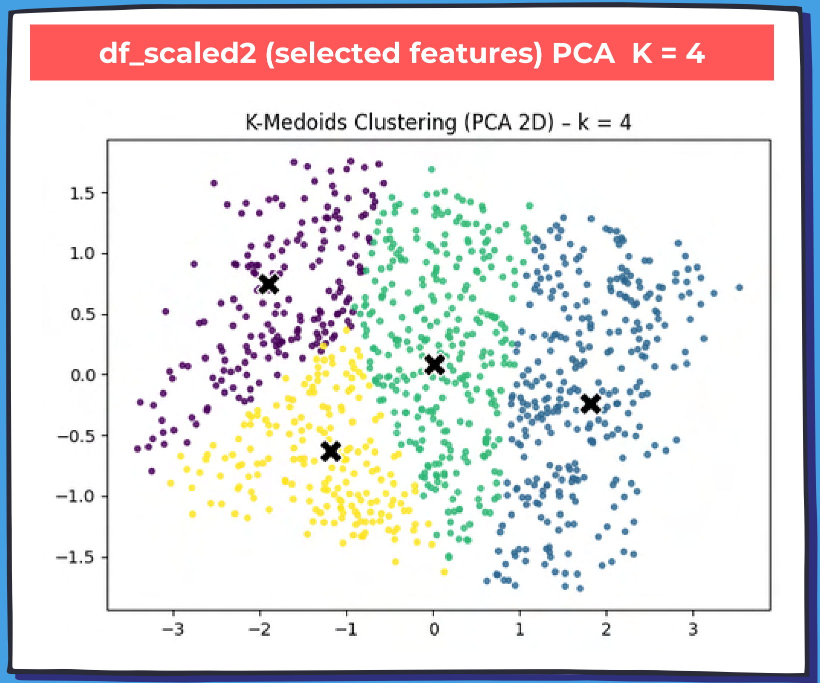
K-Medoids

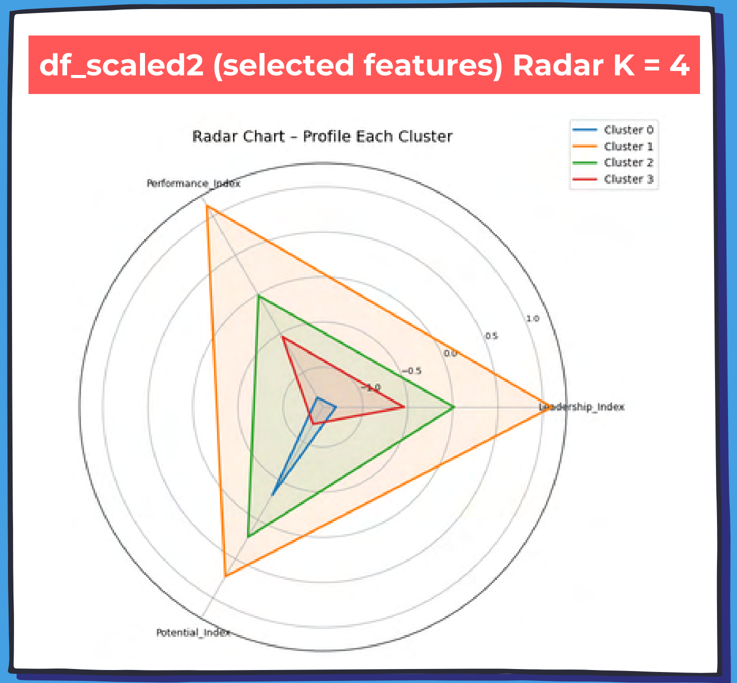
Model Interpretation

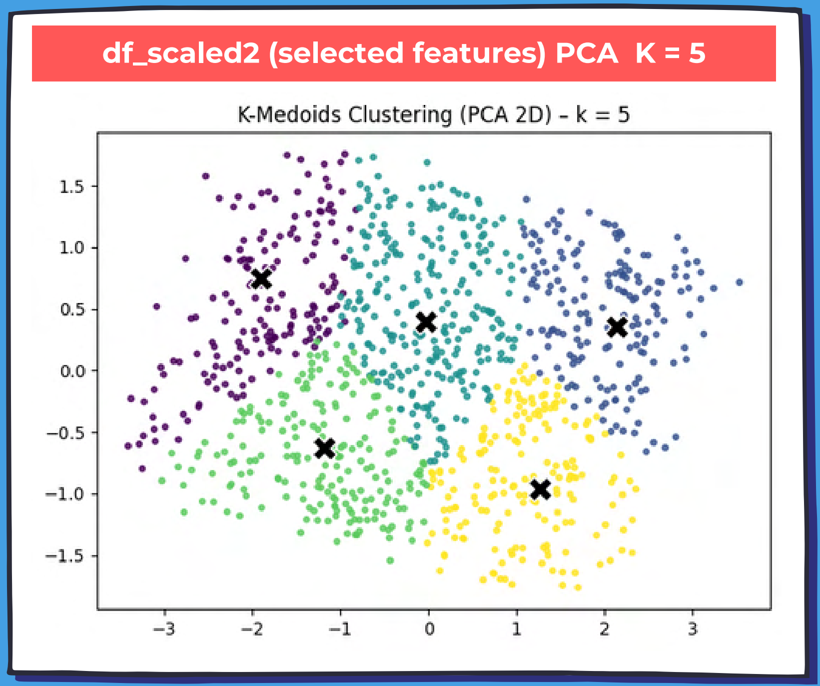
Overview: K-Medoids was applied using Euclidean distance with medoid-based centroid selection. While it is generally more robust to outliers than K-Means, the algorithm still assumes spherical cluster shapes and uses actual data points as centers, which affects how clusters form in HR datasets.

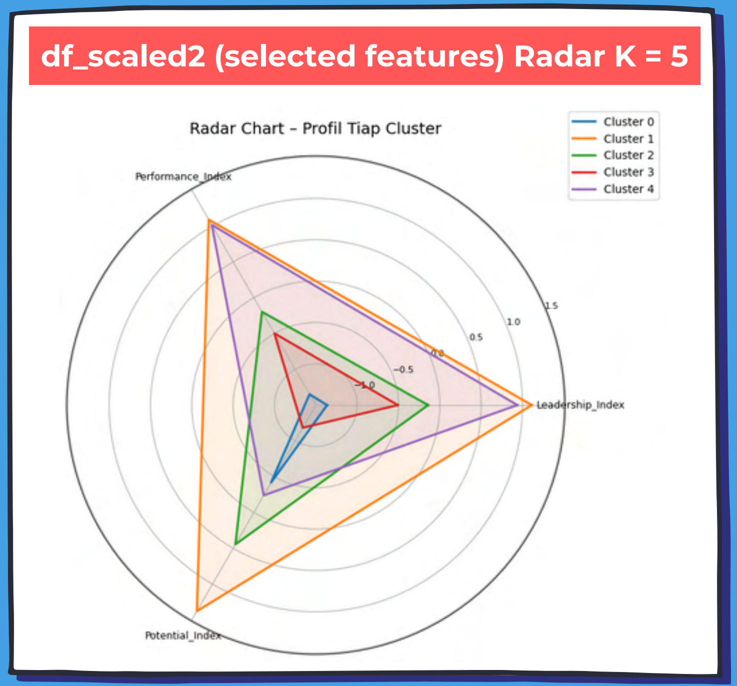
Key Observations

- Silhouette scores remain lower than K-Means (approximately 0.12–0.18), indicating weaker overall cluster separation.
- DBI values are relatively high (1.56–2.15), showing that cluster boundaries overlap heavily.
- PCA visualizations for both K=4 and K=5 show heavily blended regions, where clusters do not form clear or stable groupings.
- Radar charts reveal minimal differentiation across clusters—only subtle variations appear, and many are influenced by medoid selection rather than meaningful behavioral patterns.
- K-Medoids does not enhance clustering quality and provides less interpretable talent segments compared to K-Means.









Development process

#1 EDA

Found anomalies, missing values, and a biased Promotion_Eligible target.

#2 Data preprocessing

Cleaned the data, removed unreliable features, and created seven new engineered features.

#3 Classification models

Run tree-based and linear models (DT, RF, XGBoost, Logistic Regression, SVM). Accuracy stayed below 85% because the Promotion_Eligible label was noisy and biased.

Supervised learning in which we are using Promotion_Eligible as target feature.

#4 Rule-based calculation

Defined a threshold to label new eligibility (Eligible_New).

#5 Clustering models

Run K-Means, GMM, K-Medoids; selected K=4 based on DBI and Silhouette scores. Used for talent segmentation in the dashboard.

Unsupervised learning to find employee's characteristics.

#6 Dashboard integration

Embedded K-Means for talent mapping and rule-based for promotion probability

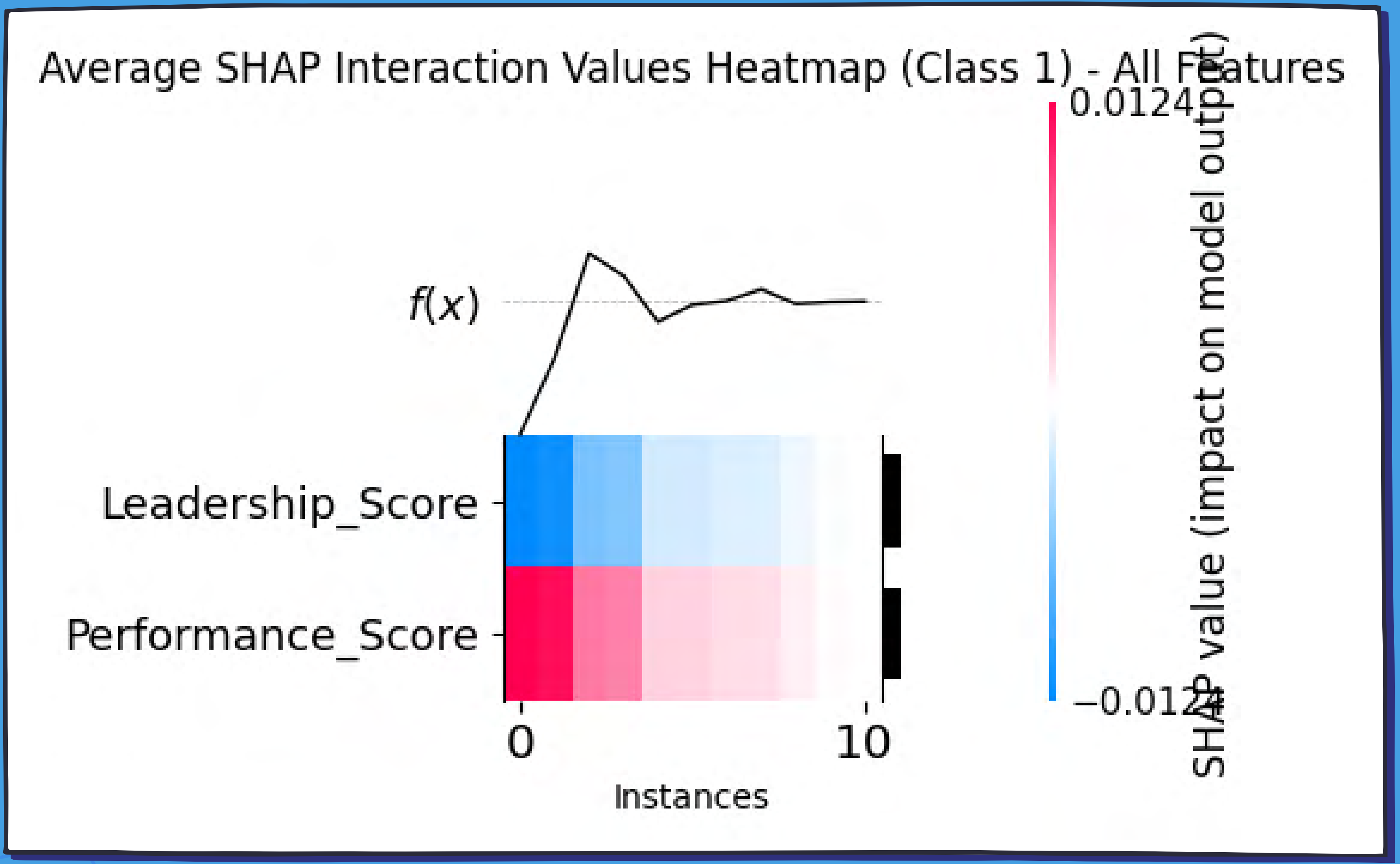
Stage 3: Evaluation.

Treebased model evaluation

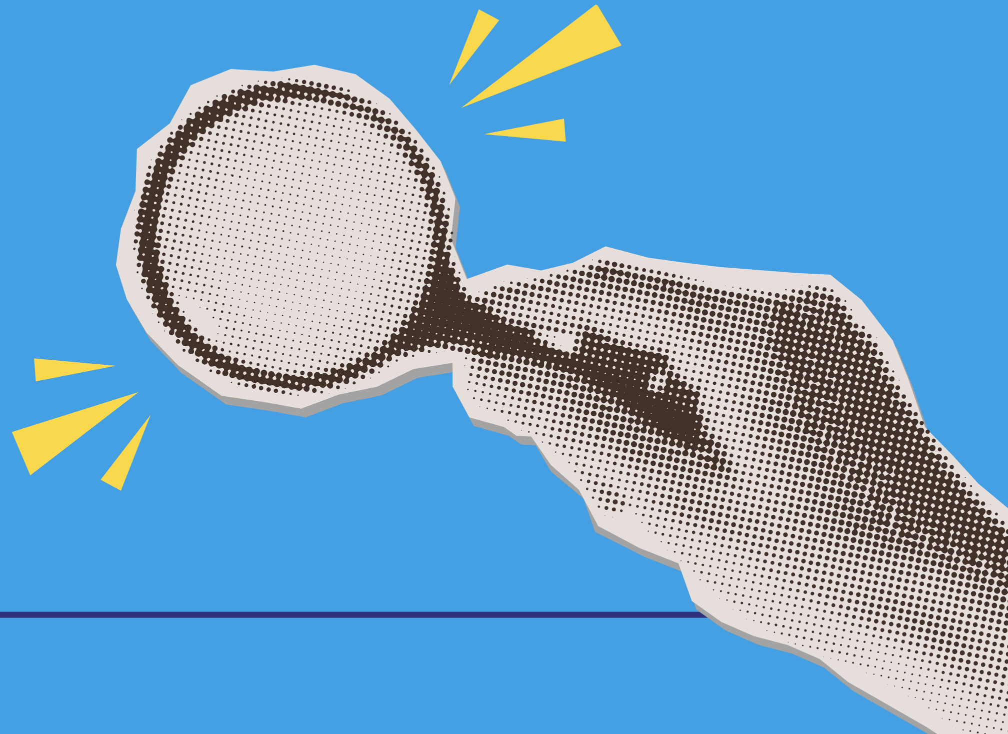
Model	F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)	Key Observation
Decision Tree	100%	75%	100%	83%	100%	68%	100%	83%	Extremely Overfit, Perfect Training Performance, Drastically Decreased Test
Decision Tree Tuned	82%	80%	78%	76%	89%	82%	96%	93%	Overfitting is handled well and Test performance is most balanced
Random Forest	100%	82%	100%	94%	100%	73%	100%	98%	Significant overfit, Perfect Training performance but Low Test Recall
Random Forest Tuned	95%	87%	93%	85%	98%	89%	99%	97%	Best Overall Test Performance
XGBoost	100%	90%	100%	93%	100%	88%	100%	99%	Strong Test Performance, Slight Overfit
XGBoost Tuned	90%	85%	86%	81%	96%	92%	99%	98%	Slightly underperforms compared to the untuned version.

The tuned Random Forest model was the overall best performer, showing the best test set performance and good balance, overcoming the overfitting issues seen in the untuned and early Decision Tree versions. The untuned XGBoost model also showed very strong test set performance.

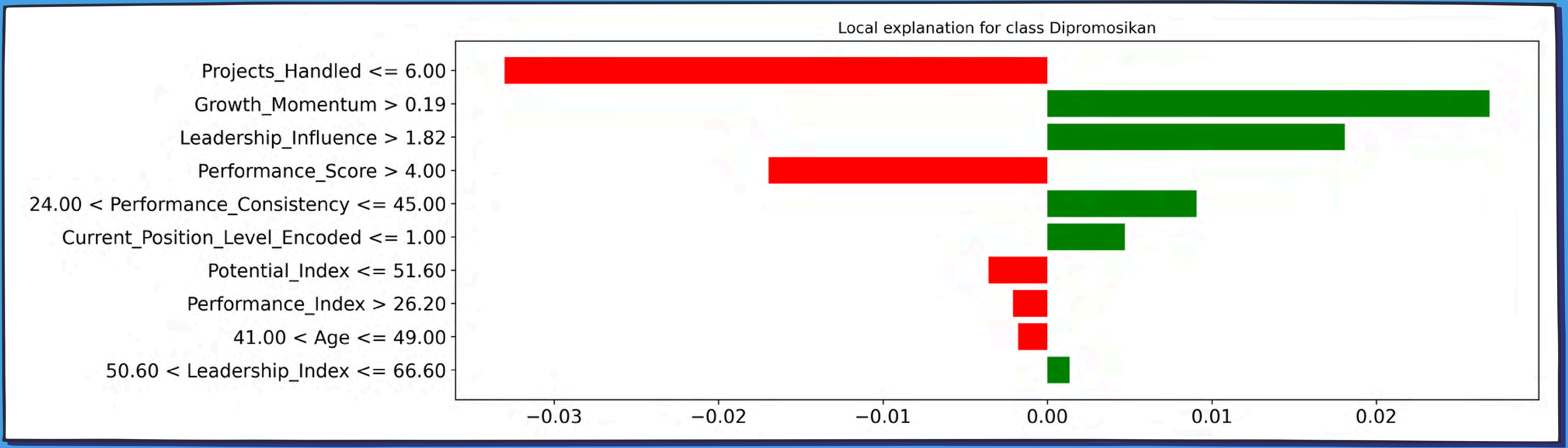
SHAP - Random Forest



The SHAP interaction heatmap shows that Performance_Score consistently pushes the Random Forest toward predicting Class 1 (red tones), while Leadership_Score generally pushes predictions downward (blue tones). This contrast means the model relies more on performance indicators than leadership signals when deciding Class 1, highlighting performance as the dominant driver in the prediction.



LIME - Random Forest



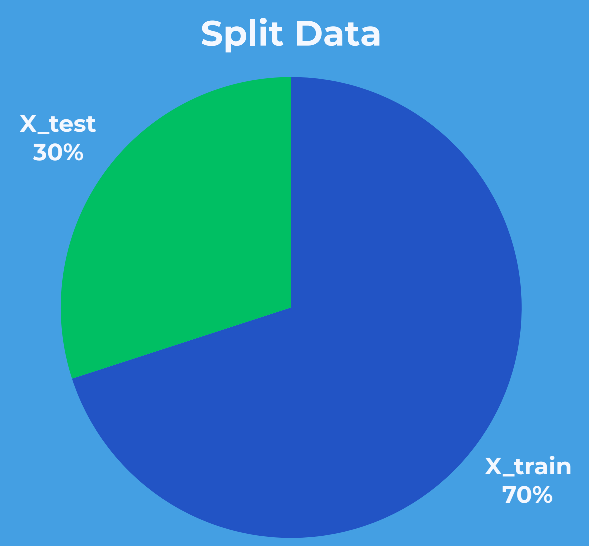
The LIME shows that for the selected employee, strong growth and leadership indicators support the promotion prediction, while low project count and inconsistent performance reduce it.

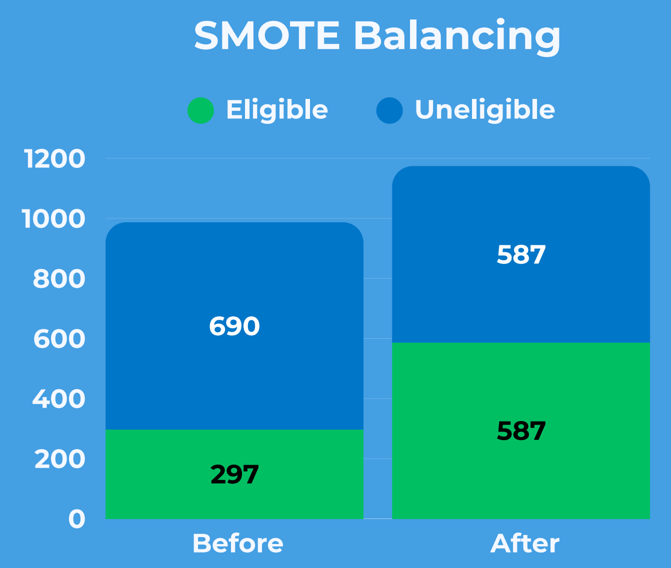
Linear model evaluation

Model	F1 (Train)	F1 (Test)	Precision (Train)	Precision (Test)	Recall (Train)	Recall (Test)	ROC-AUC (Train)	ROC-AUC (Test)	Key Observation
Logistic Regression	94%	95%	90%	91%	100%	100%	99%	99%	Cross-validation provided a slight, non-significant improvement.
Logistic Regression Tuned	99%	98%	99%	99%	99%	97%	99%	99%	Hyperparameter tuning improved the F1 a scores compared to the baseline.
SVM	93%	87%	87%	83%	91%	100%	99%	99%	Good performance, but not as good as Logistic Regression model.
SVM Tuned	95%	93%	92%	91%	98%	96%	99%	99%	Tuning the SVM model did not improve the model much.

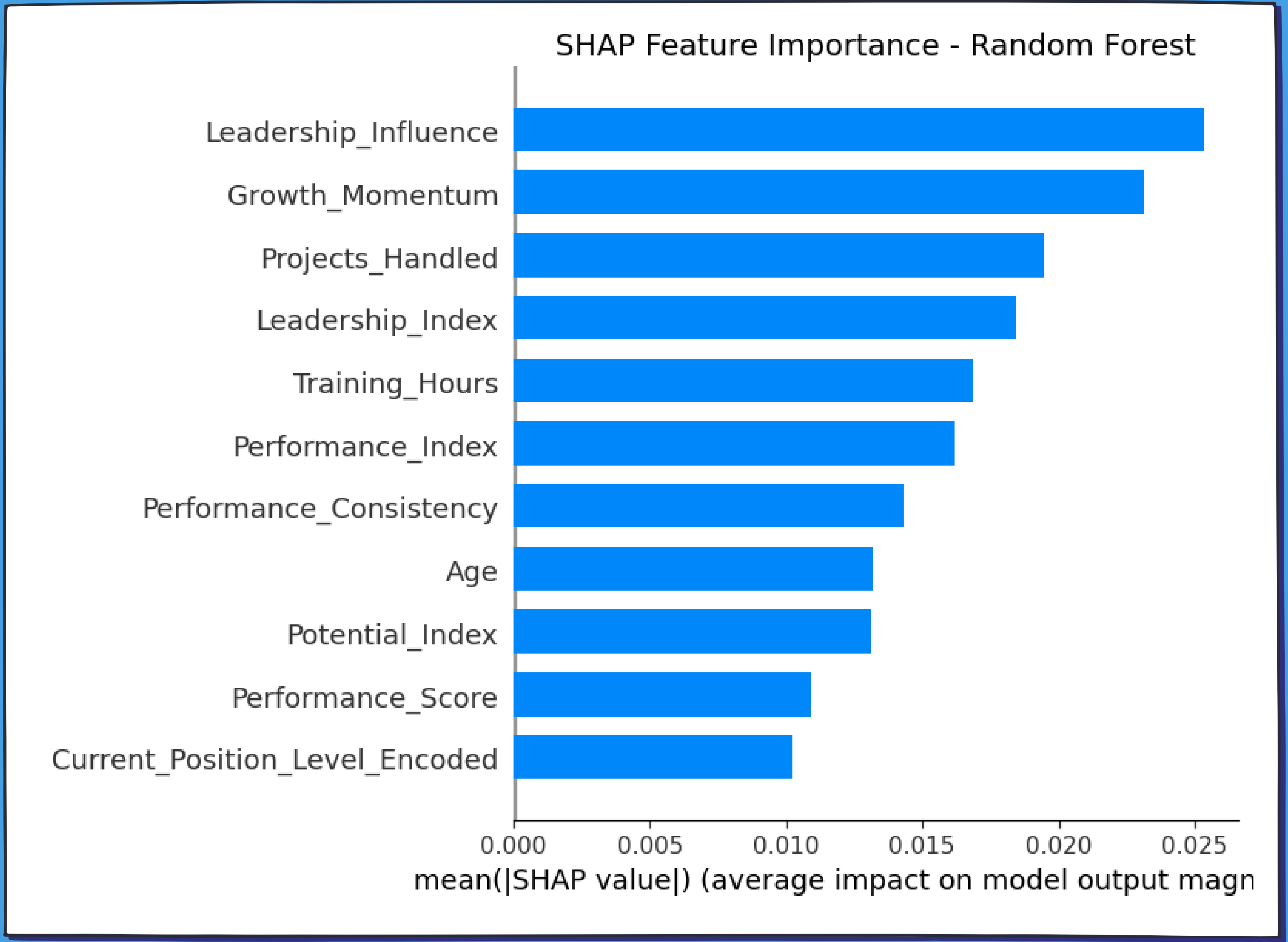
Drop some features that have high correlation:

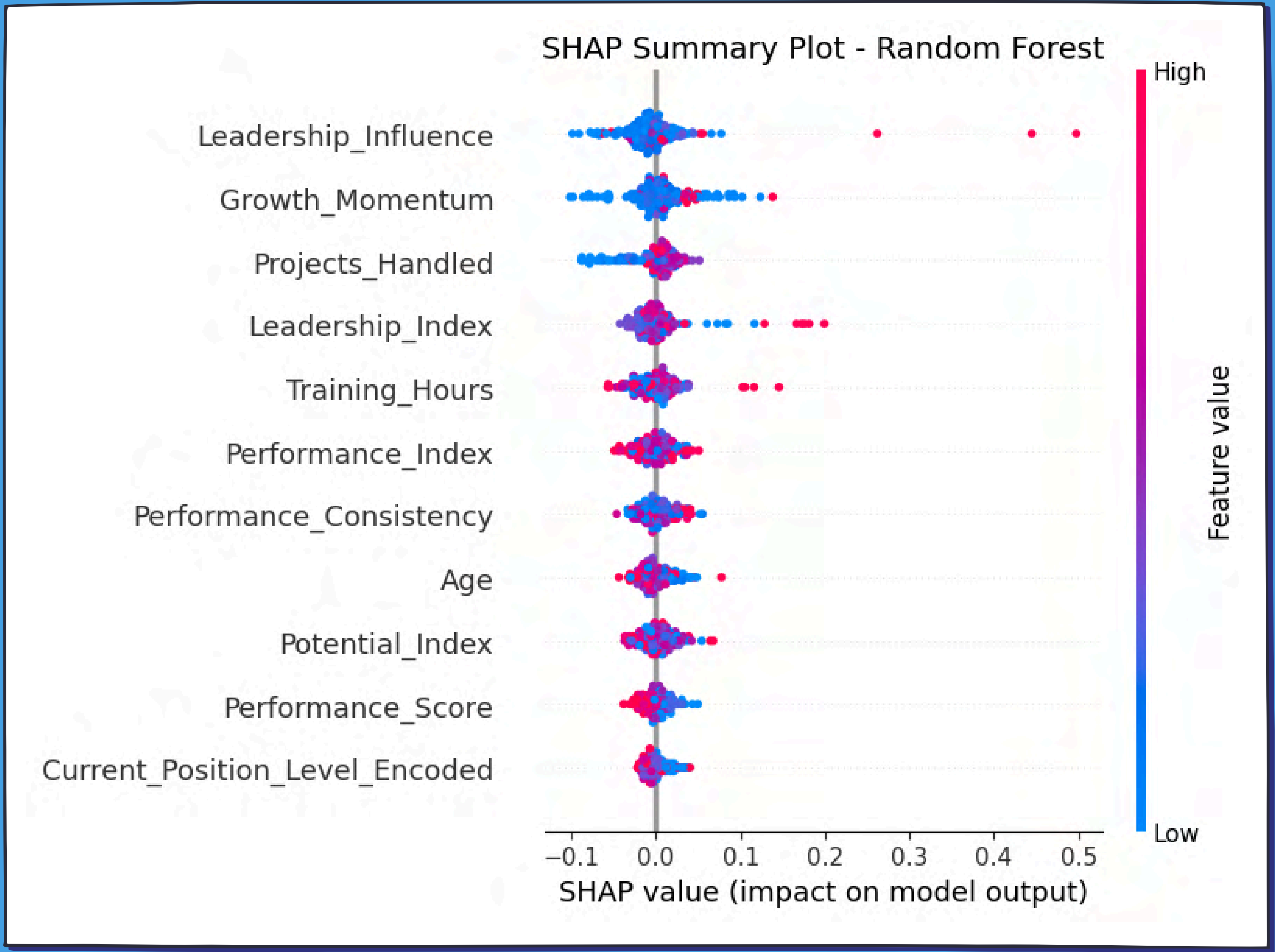
- Performance_Score
- Training_Hours
- Project__Handled
- Peer_Review_Score





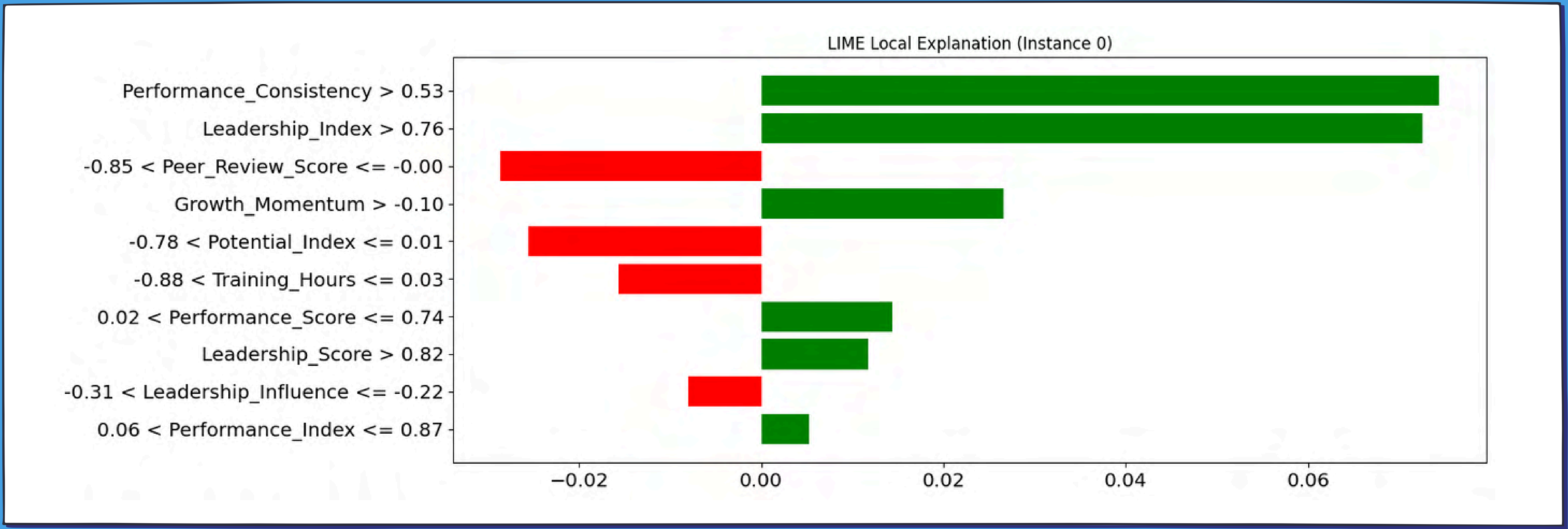
SHAP - Logistic Regression





The SHAP results show that leadership and growth-related features, especially Leadership Influence, Growth Momentum, and Projects Handled are the strongest global drivers of the promotion predictions.

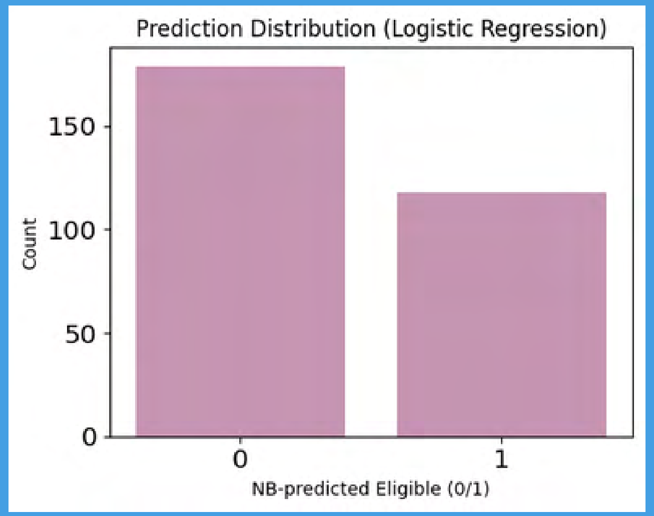
LIME - Logistic Regression

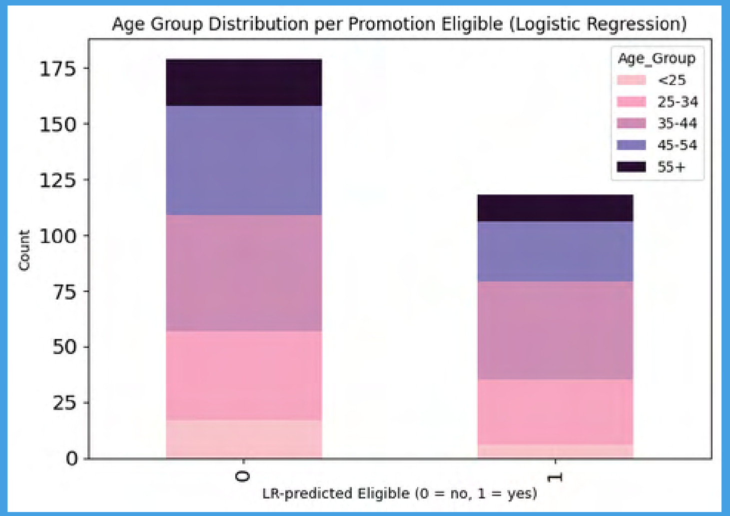


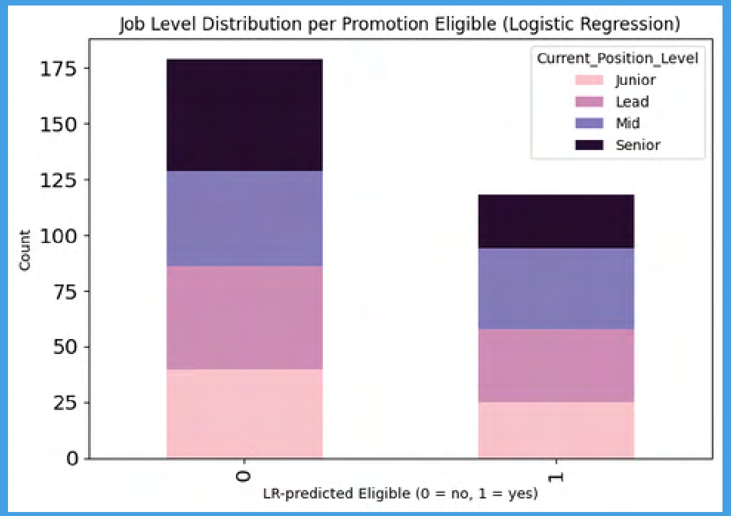
LIME shows which features influenced this specific prediction. Performance_Consistency, Leadership_Index, and Growth_Momentum push the prediction upward (green), while Peer_Review_Score, Potential_Index, and Training_Hours push it downward (red). This explains how the model reached its decision for this instance.

Classification bias analysis

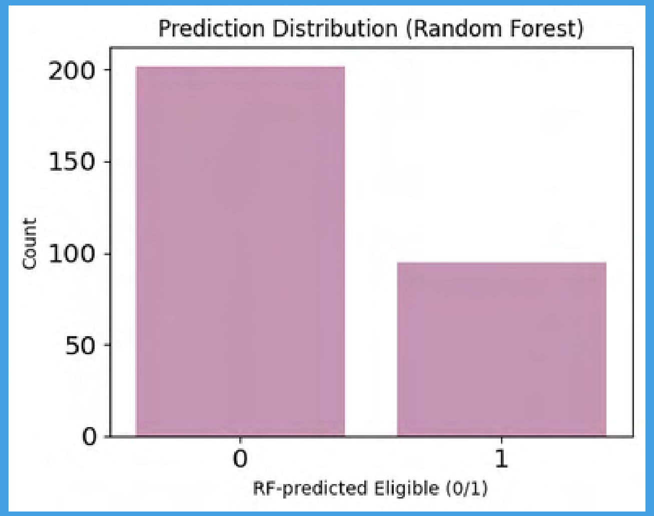
Logistic Regression

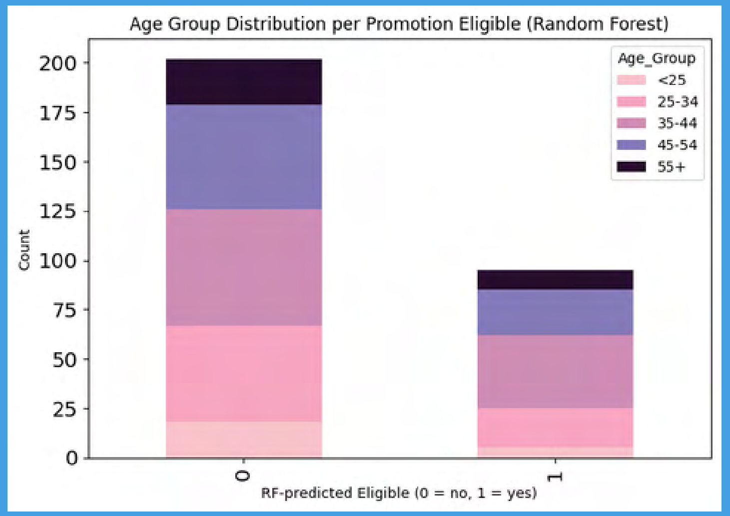


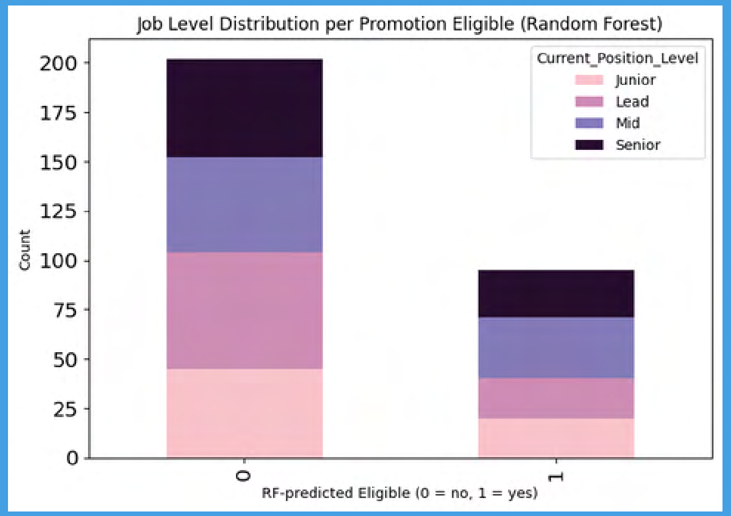




Random Forest



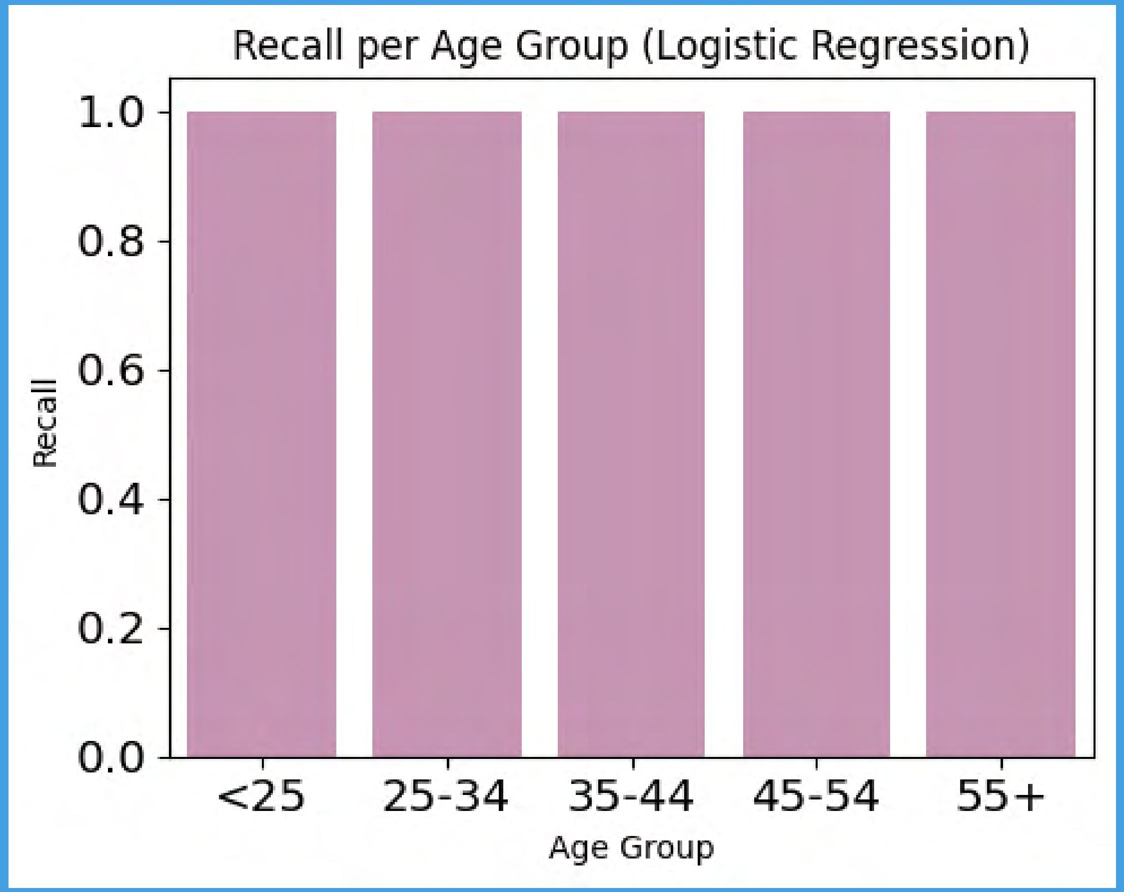




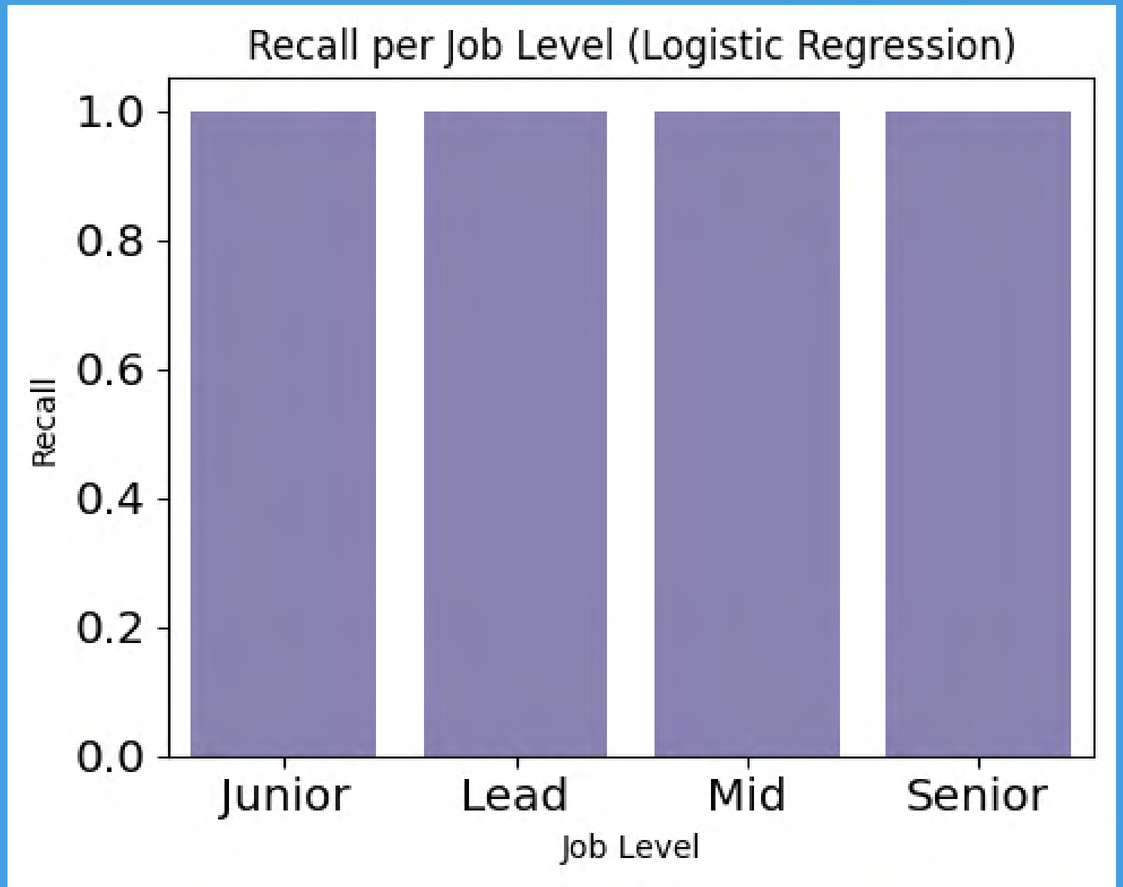
The Prediction Distribution diagram shows how many people are predicted to be “Ineligible” (0) and “Eligible” (1). The Age Group Distribution diagram shows which ages are more likely to be predicted for promotion or rejection. The Job Level Distribution diagram shows which job levels are most likely to be predicted for promotion.

Fairness Logistic Regression

Age



Job Level



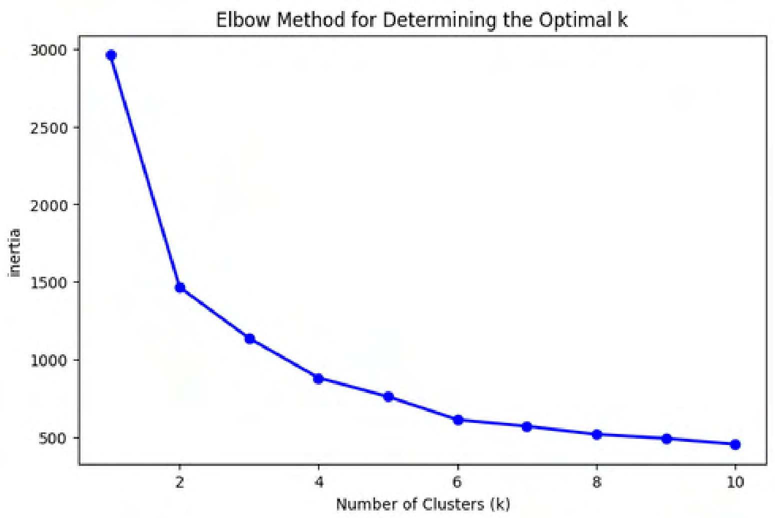
Testing Logistic Regeression fairness based on age group and job level using recall metric per group. The model successfully identified employees who were truly eligible for promotion in each group. No particular age group or job level was promoted more frequently than others.

Clustering model

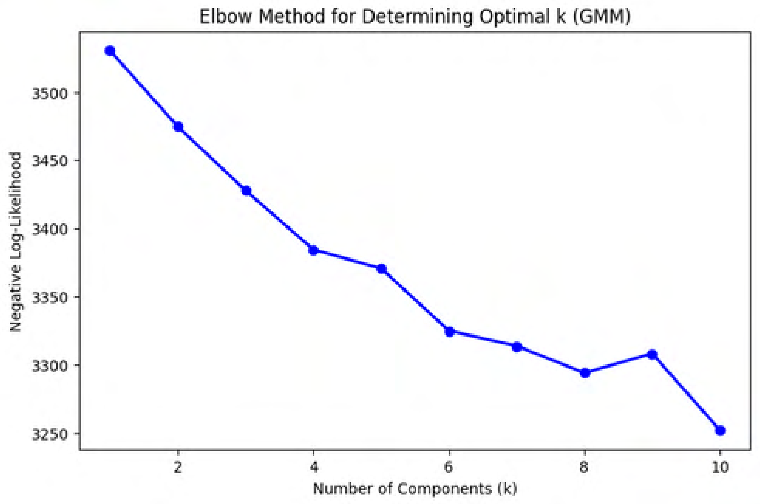
Criteria	K-Means	GMM	K-Medoids
Best K	K = 4	K = 5	K = 5
Silhouette Score	0.320	0.247	0.293
DBI Score	1.00	1.173	1.071
Key Observation	Clearest and most compact clusters.	Lowest performance, overlapping clusters.	Fairly good and stable but not as good as K-Means.

We tested three clustering algorithms: K-Means, GMM, and K-Medoids. K-Means with K = 4 won. It has the highest Silhouette Score (0.320) and best DBI (1.00), with clear, well-separated, and compact clusters. The elbow plots confirm K = 4 is optimal. This interpretability and cluster quality make K-Means perfect for segmenting employee talent profiles.

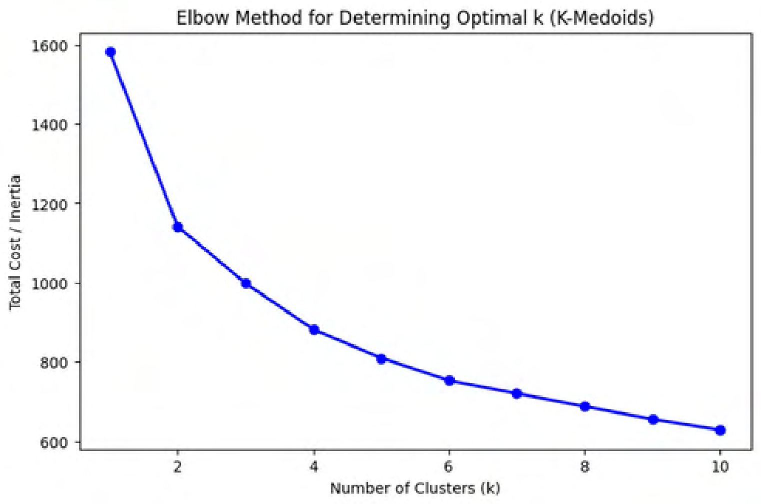
Elbow K-Means



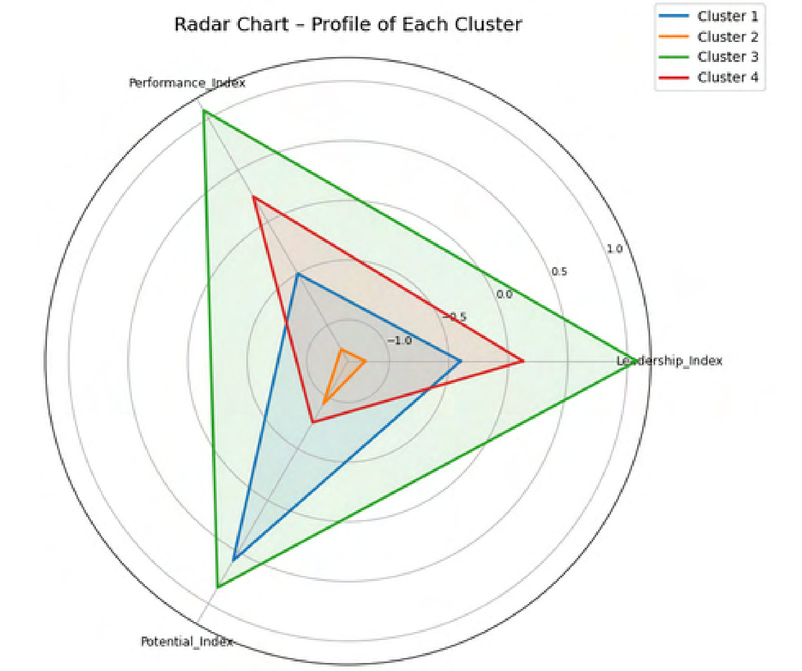
Elbow GMM



Elbow K-Medoids



Radar Chart K = 4



Clustering model interpretation

K-Means with K=4

Cluster 1: Under Developed With Potential

Technical: Moderate scores in Performance (-0.50) and Leadership (-0.39), but relatively higher scores in Potential (0.58). Stable and fairly consistent, but has yet to demonstrate strong execution skills.
Business: A reliable and stable operational force. With proper guidance, they can quickly develop to make a higher impact.

Cluster 2: At-Risk & Underpowered

Technical: Lowest values across all three indexes, which are performance (-1.23), leadership (-1.20), and potential (-0.93). Shows stagnation and lack of momentum.
Business: Requires priority attention. High risk of underperformance and disengagement, necessitating intervention, reskilling, or realignment.

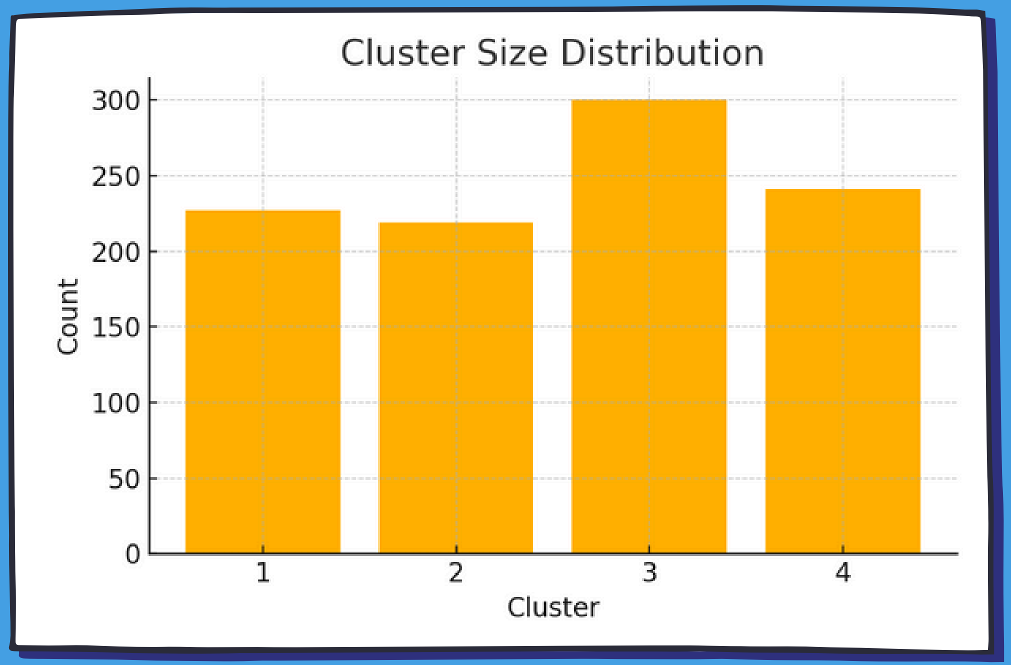
Cluster 3: All Around Top Performer

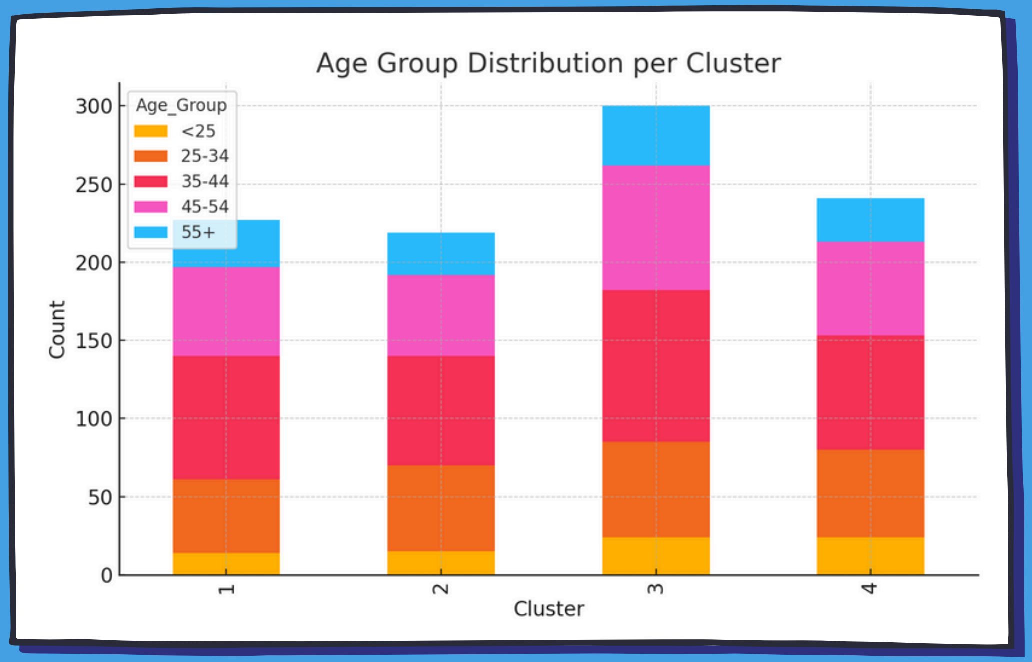
Technical: Achieves the highest scores in performance (1.08), leadership (1.08), and potential (0.84); highly adaptive and shows strong momentum.
Business: The best candidates for career acceleration and leadership pipeline. A strategic source of potential for the organization's future.

Cluster 4: Consistent Performer or Leader

Technical: Demonstrates consistent execution with average performance (0.24) and leadership (0.12), but low potential (-0.75). This means they are reliable in their day-to-day work but do not yet show strong signs of long-term growth.
Business: This is a strong operational core. It is important to retain as a specialist or stable role that supports the quality and sustainability of the organization.

Clustering bias analysis





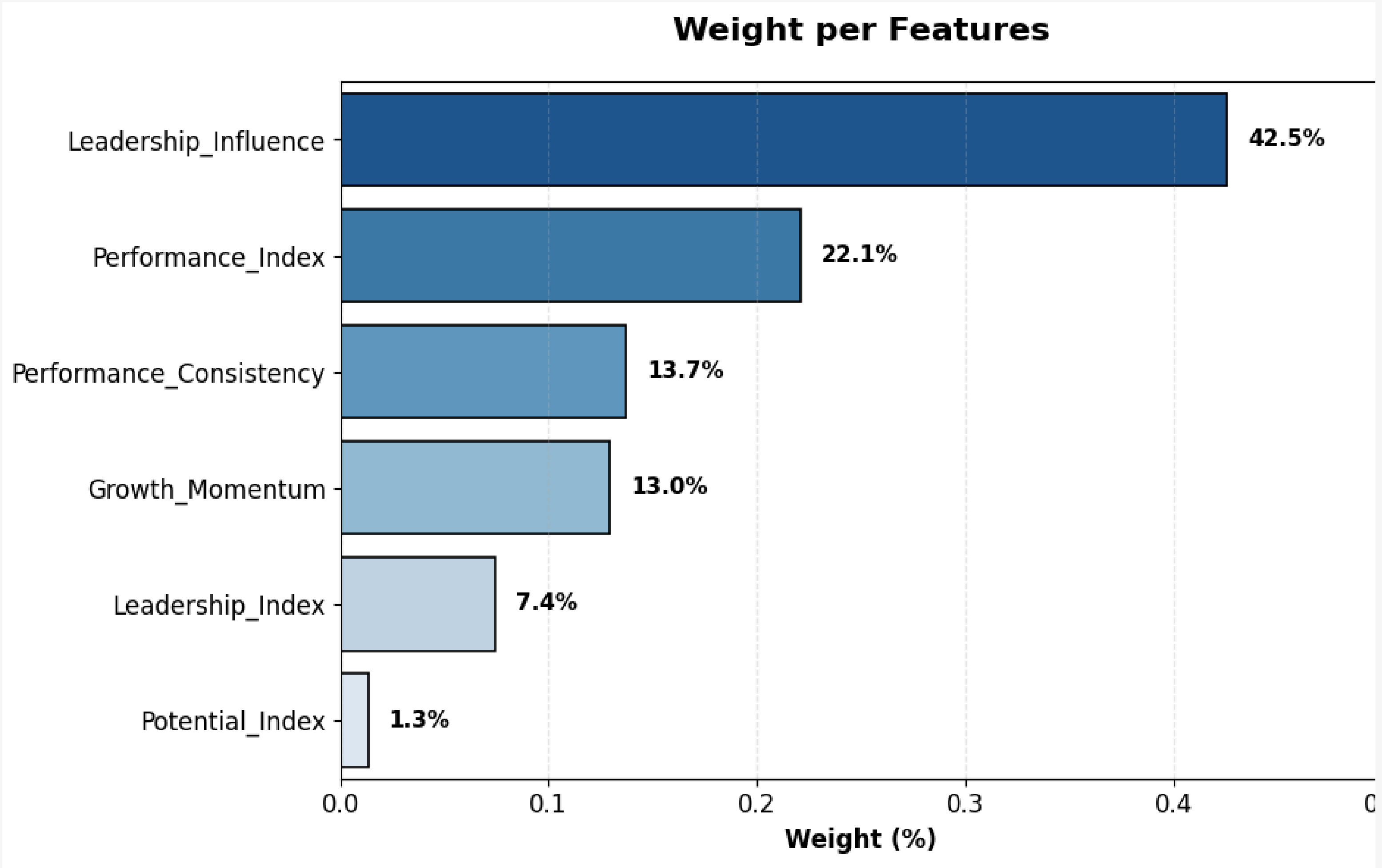
Cluster Size Distribution
All four clusters are relatively balanced, which indicates that K-Means did not overfit or collapse into disproportionately small groups. Balanced cluster sizes help ensure fairer interpretation and reduce the risk of segment-level bias due to under-representation.

Age Group Distribution Across Clusters
Age distribution across clusters appears fairly even, with each segment containing a healthy mix of early-career, mid-career, and senior employees. No cluster is dominated by a single age group, suggesting the clustering is not driven by age-related patterns and reducing the risk of age-driven representation bias.

Job Level Distribution per Cluster
All clusters contain employees across roles. While some segments lean slightly toward certain levels, no cluster is restricted to or overwhelmingly dominated by a single job tier. This indicates that the model is capturing behavioral and capability signals rather than simply replicating job-level hierarchy.

New feature for rule-based model

Weights are taken from normalized Logistic Regression coefficients to reflect the relative influence of each feature. Leadership Influence is the dominant factor, followed by performance and consistency. This approach makes promotion decisions data-driven. The results of this will later be used in the promotion eligibility calculator on our website.



Promotion_Score

$$Score = (w_1 \times X_1) + (w_2 \times X_2) + ... + (w_n \times X_n)$$

Eligible_New

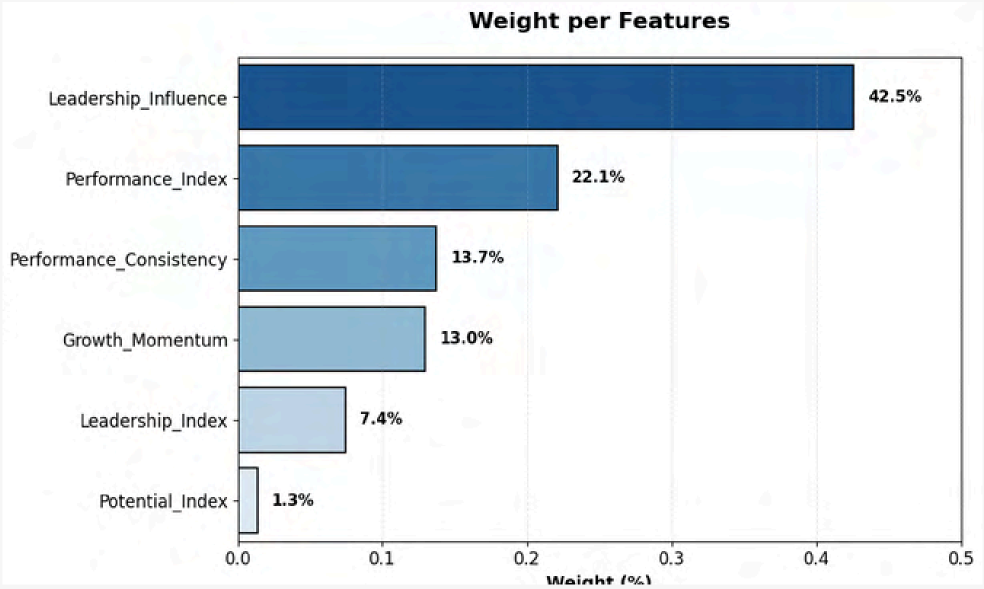
Promotion_Score
> 0.85

Rule-based model

We built a transparent rule-based system instead of a black-box model. It combines our features with weights, Leadership Influence (42.3%), Performance Index (22.1%), and others into a single Promotion Score. Scores above 0.85 indicate promotion readiness. Every recommendation is explainable and auditable by HR.

Promotion_Score

$$Score = (w_1 \times X_1) + (w_2 \times X_2) + ... + (w_n \times X_n)$$



Eligible_New

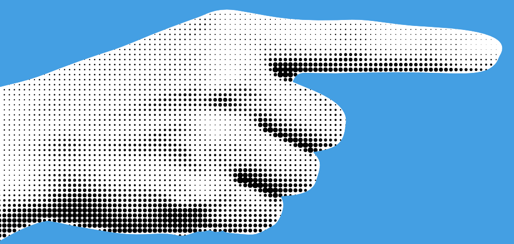
Promotion_Score > 0.85

	JUNIOR	LEAD	MID	SENIOR
1	42	32	42	32
0	196	207	254	182

Our rule-based threshold is backed by data. Industry standards show 6-8% is healthy, 8-12% is good, and tech companies average 14% promotion rates. Our system is tuned to match these realistic benchmarks, not arbitrary thresholds.

Source: taggd.in, pave.com, aihr.com

Final decision



K-Means model

K-Means (k = 4) with a Silhouette score of 0.32 and a DBI score of 1.00 provides the most stable segmentation and is easy for HR to understand for decision making.

Clustering gives the dashboard its intelligence by turning raw employee metrics into clear talent segments. Instead of scattered individual data points, HR sees structured groups with distinct strengths and risks. This makes the dashboard easier to interpret, supports targeted decisions, and gives every visualization a meaningful story about the organization’s talent landscape.

+

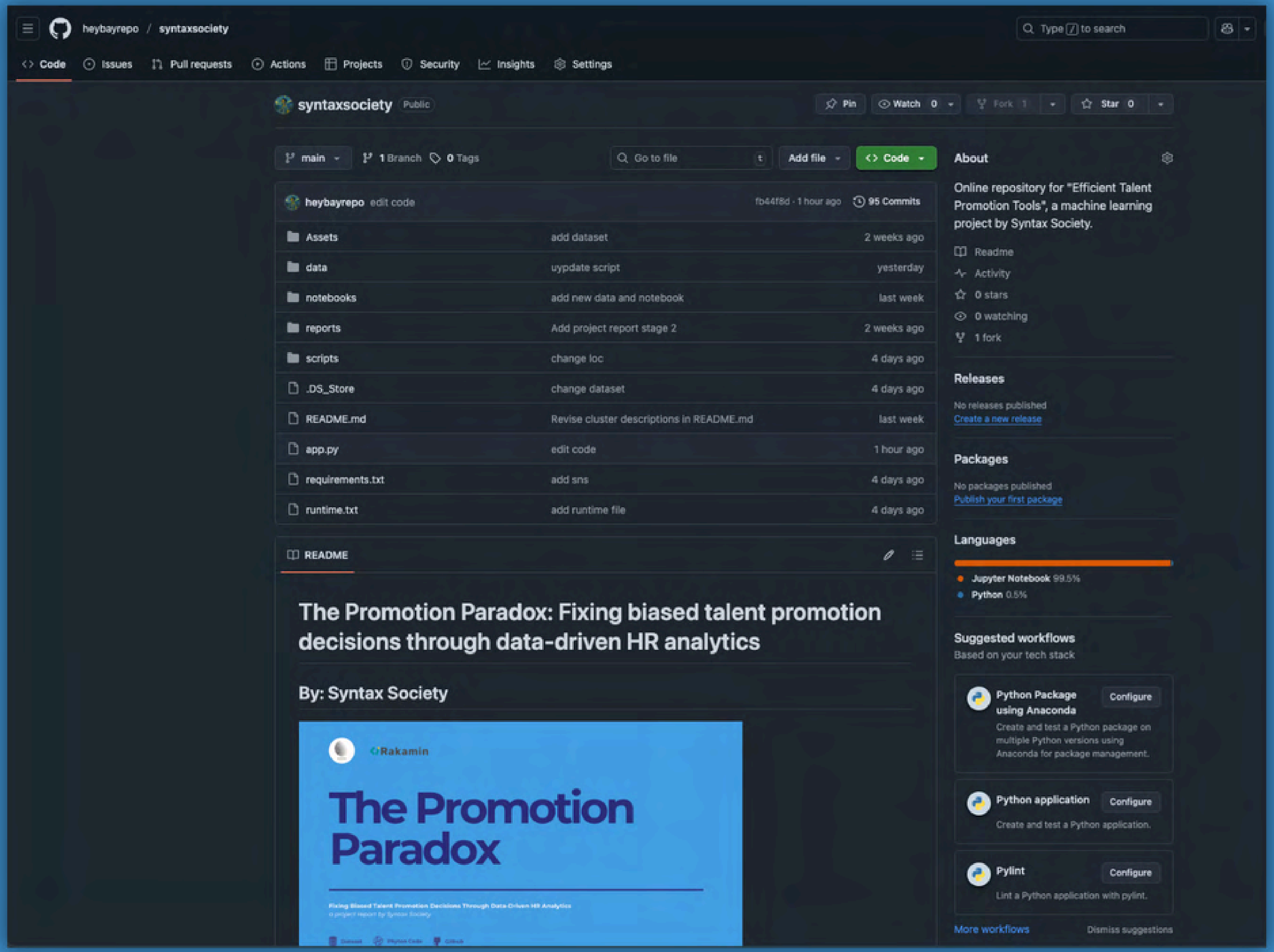
Rule-based

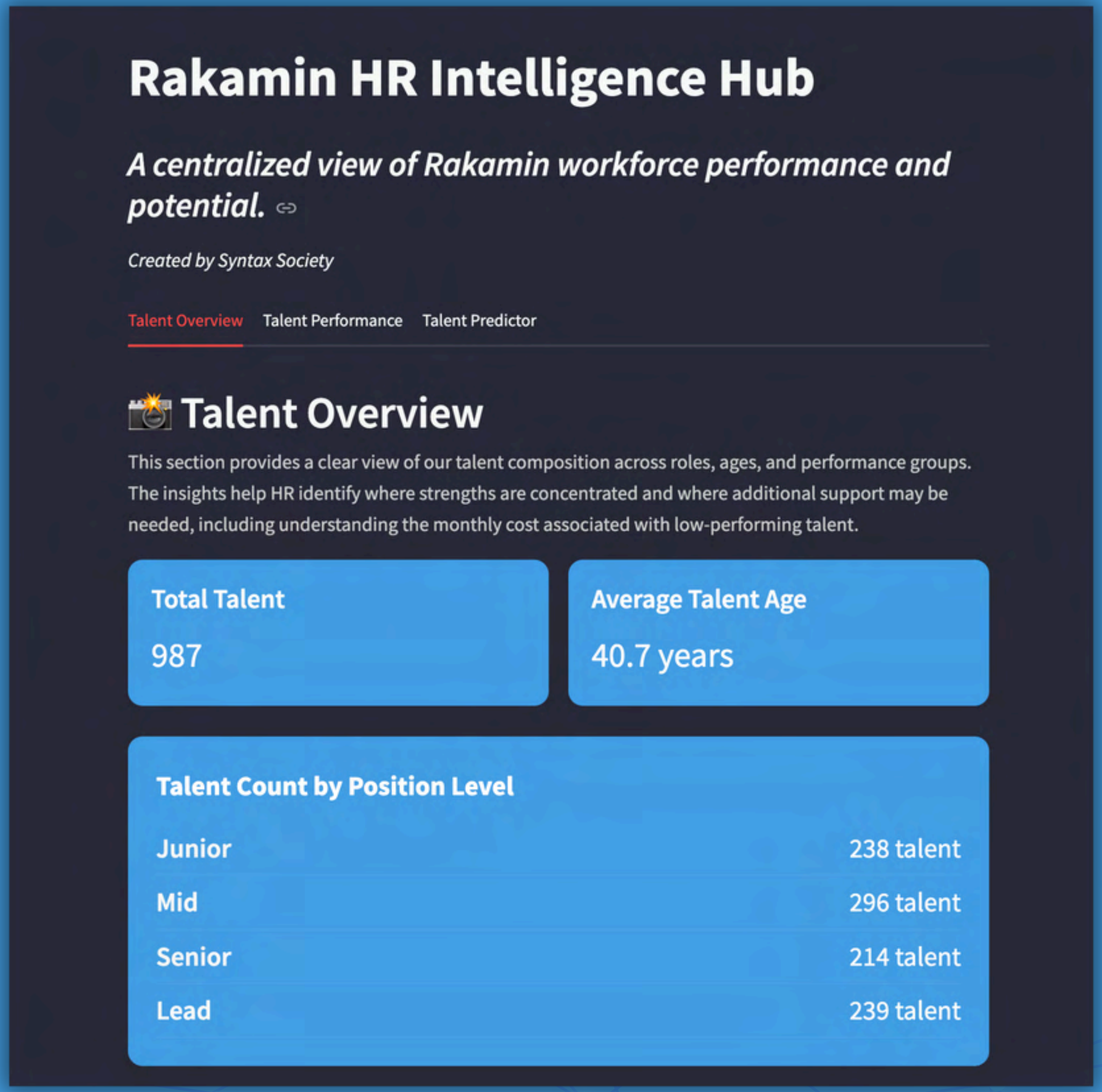
This rule-based method calculates a Eligible New using weighted feature values—Leadership Influence (0.425), Performance Index (0.221), Performance Consistency (0.137), Growth Momentum (0.130), Leadership Index (0.074), and Potential Index (0.013). The score directly reflects each factor’s importance, emphasizing leadership impact and core performance. This transparent, non–machine-learning approach provides consistent, easy-to-explain promotion decisions.

Stage 4: Deployment.

From Github to Streamlit

We built the dashboard on Streamlit with GitHub integration. Updates deploy automatically with each code push. The key design philosophy: keep it simple for non-technical users. Clean interface, clear metrics, intuitive workflows. And simplicity enables scalability: easier to maintain, modify, and expand as your needs grow. Check our prototype at
https://rakaminhrdashboard.streamlit.app/





Source: Streamlit

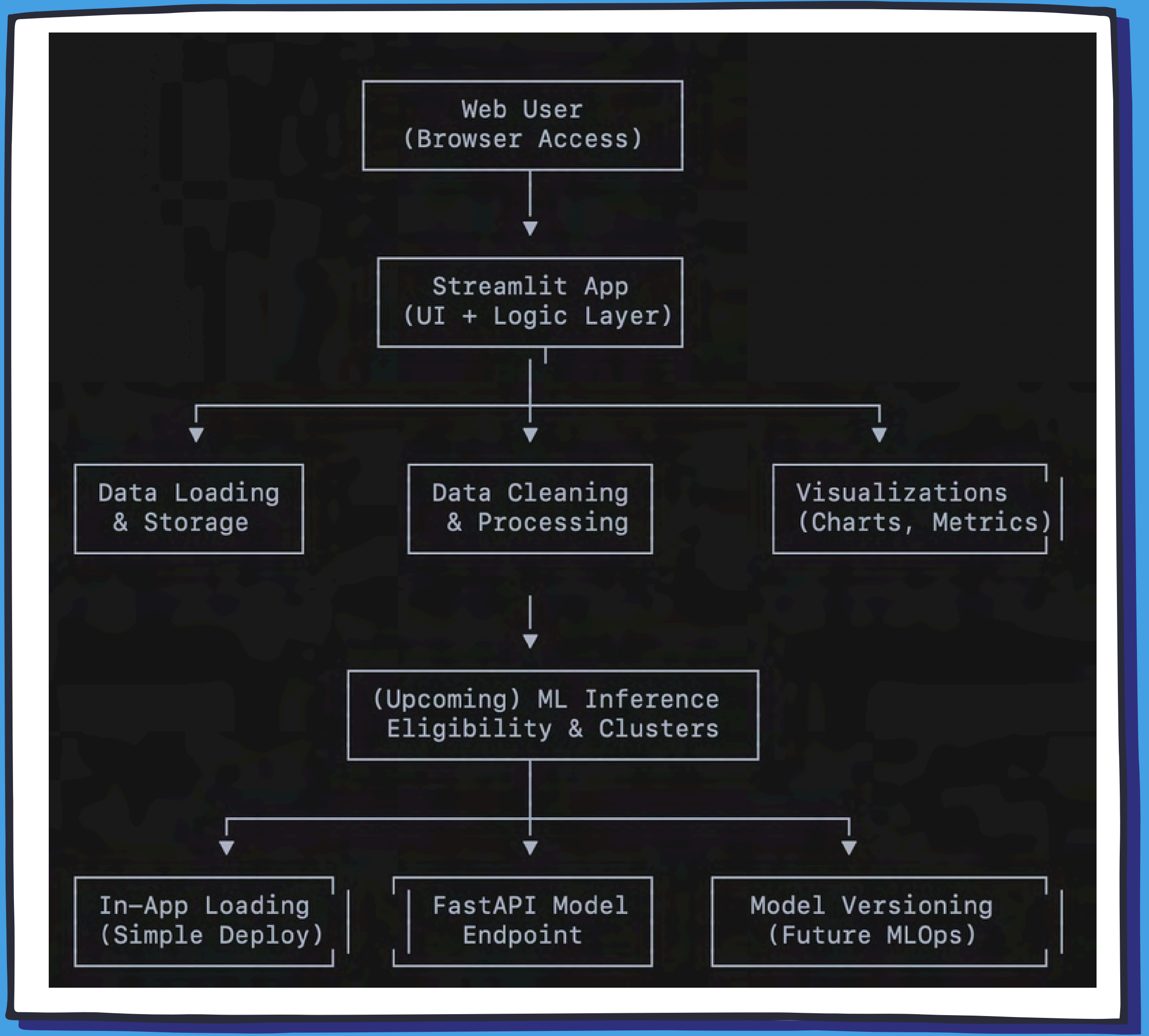
Model deployment

Deployment Architecture:

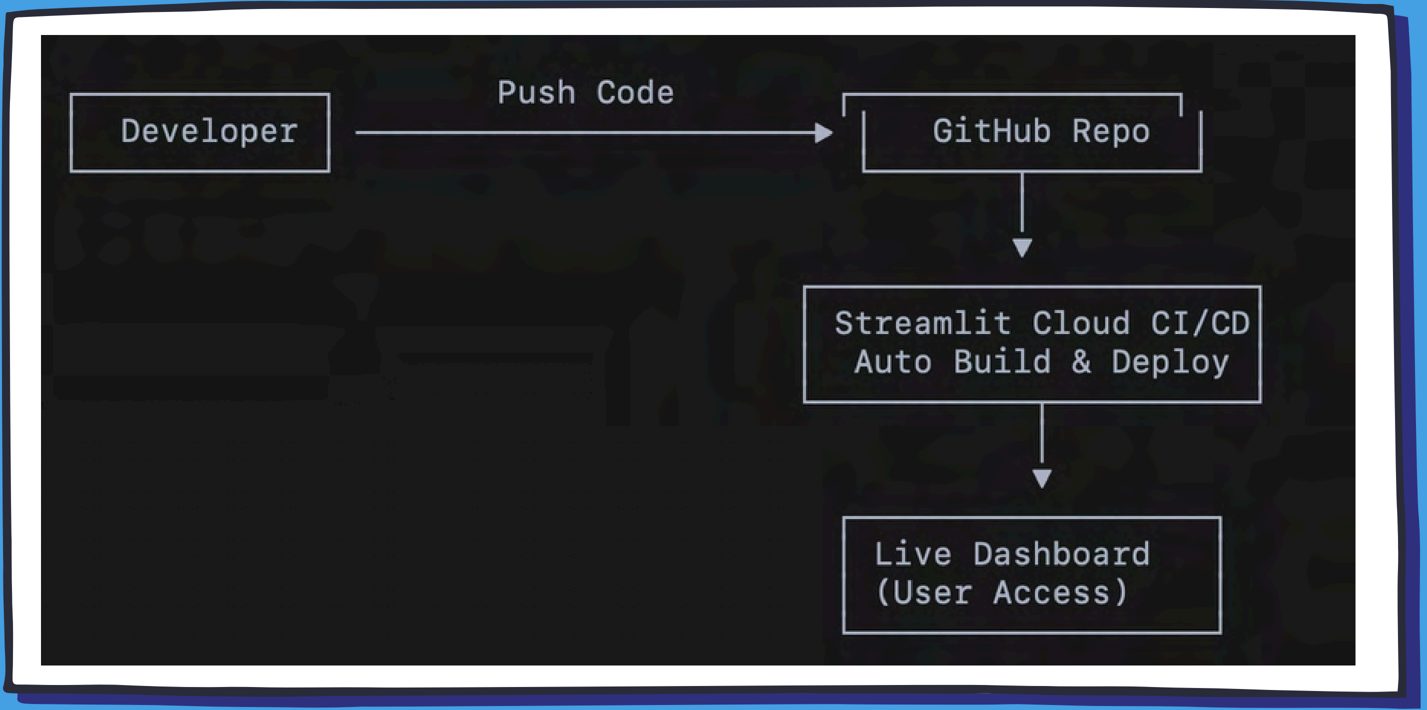
- The current deployment operates as a unified Streamlit application where the UI, data pipeline, and (upcoming) ML inference logic live in the same environment. This architecture is optimized for rapid iteration and low-maintenance deployment.

Key Components:

- Frontend: Streamlit interface for interactive exploration, filtering, and visualization.
- Backend: Python-based processing pipeline powered by Pandas, NumPy, and Seaborn.
- Hosting: Streamlit Cloud with automated build, environment setup, and deployment.
- Repository Backbone: requirements.txt ensures reproducible environments across development and production.



Model deployment




```
pip install -r requirements.txt
streamlit run rakaminhrdashboard_app.py
```

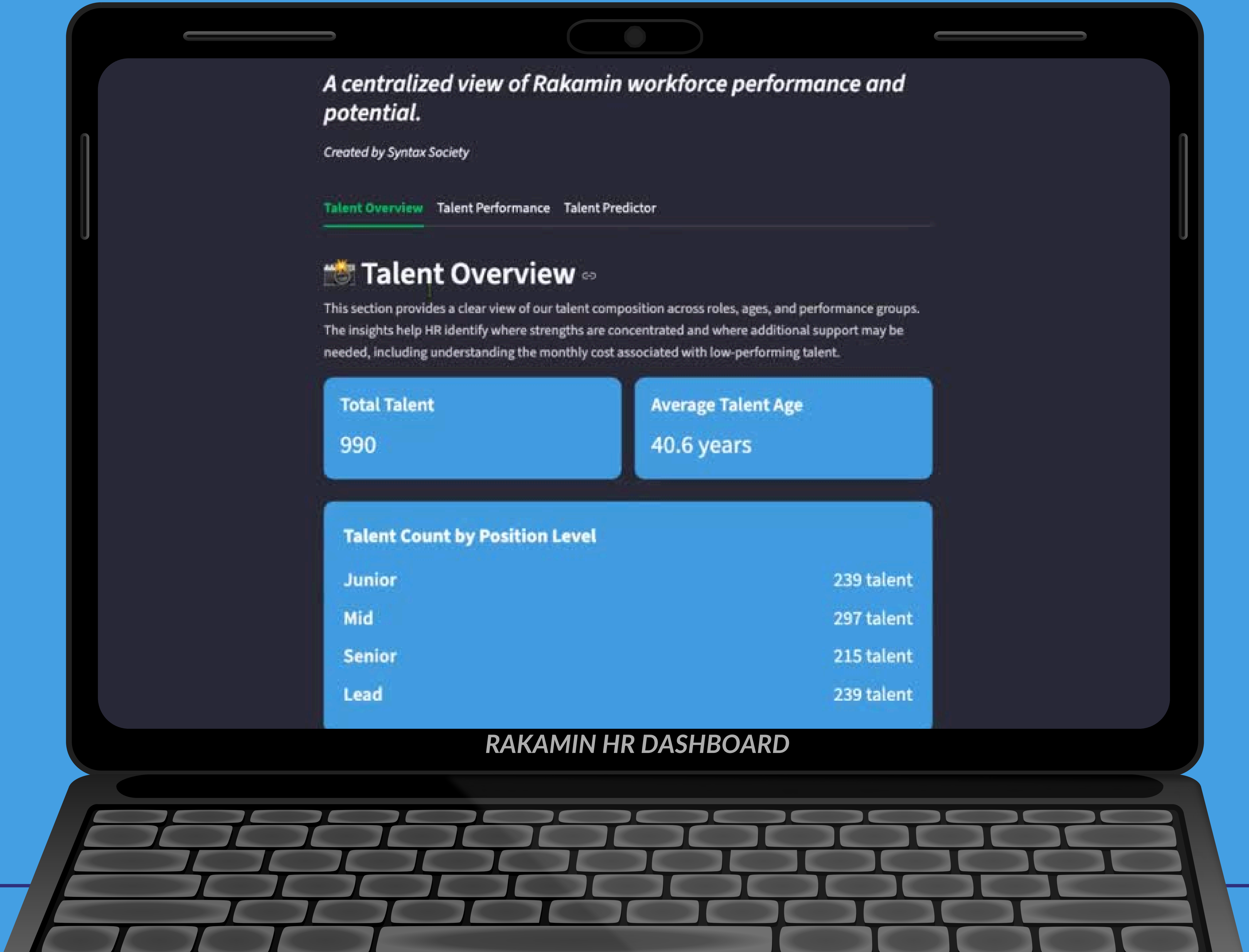
How the App Is Served:

- The GitHub repository triggers an automatic deployment when new commits are pushed.
- Streamlit Cloud rebuilds the environment and deploys the updated application.
- Users access the dashboard through the public Streamlit URL without needing any local installation.
- This serverless deployment approach minimizes infrastructure overhead and is ideal for experimentation, stakeholder demos, and low-latency analytical workflows.

Local Development:

- Developers can run the dashboard locally with a single command.

App features



Tab 1: Talent Overview. Shows a snapshot of the workforce: total talent, position-level distribution, and potential attrition indicators.

Tab 2: Talent Performance. Displays top and bottom performers across key indexes (performance, leadership, potential) with optional filters for deeper exploration.

Tab 3: Talent Predictor. ML-powered clustering that reveals each employee's group, traits, and risks. Users can input data manually or upload records via CSV.

Source: Streamlit

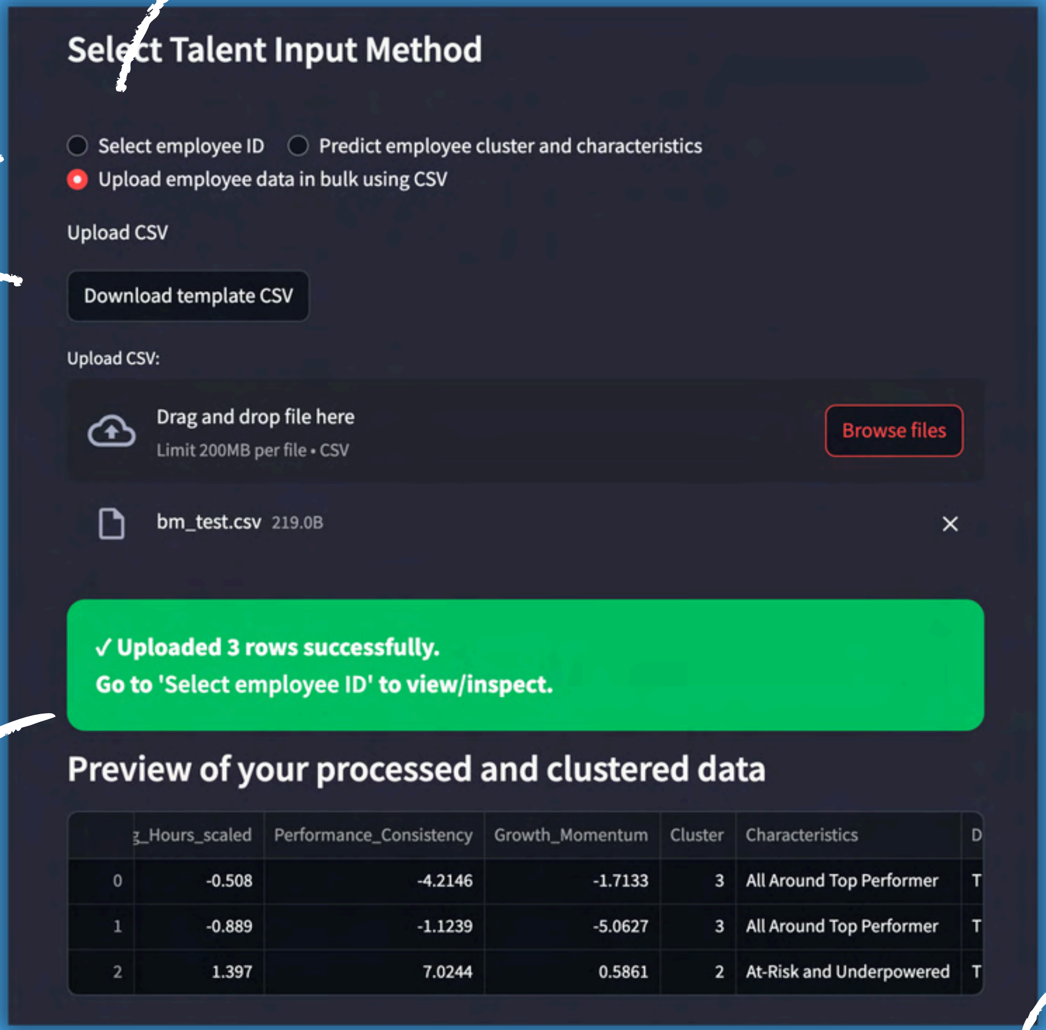
Flexibility & scalability

Select employee ID to see specific employee. We're not using name because it is not in the dataset.

HR can choose to upload their own dataset

Use the provided CSV template to ensure uniformity

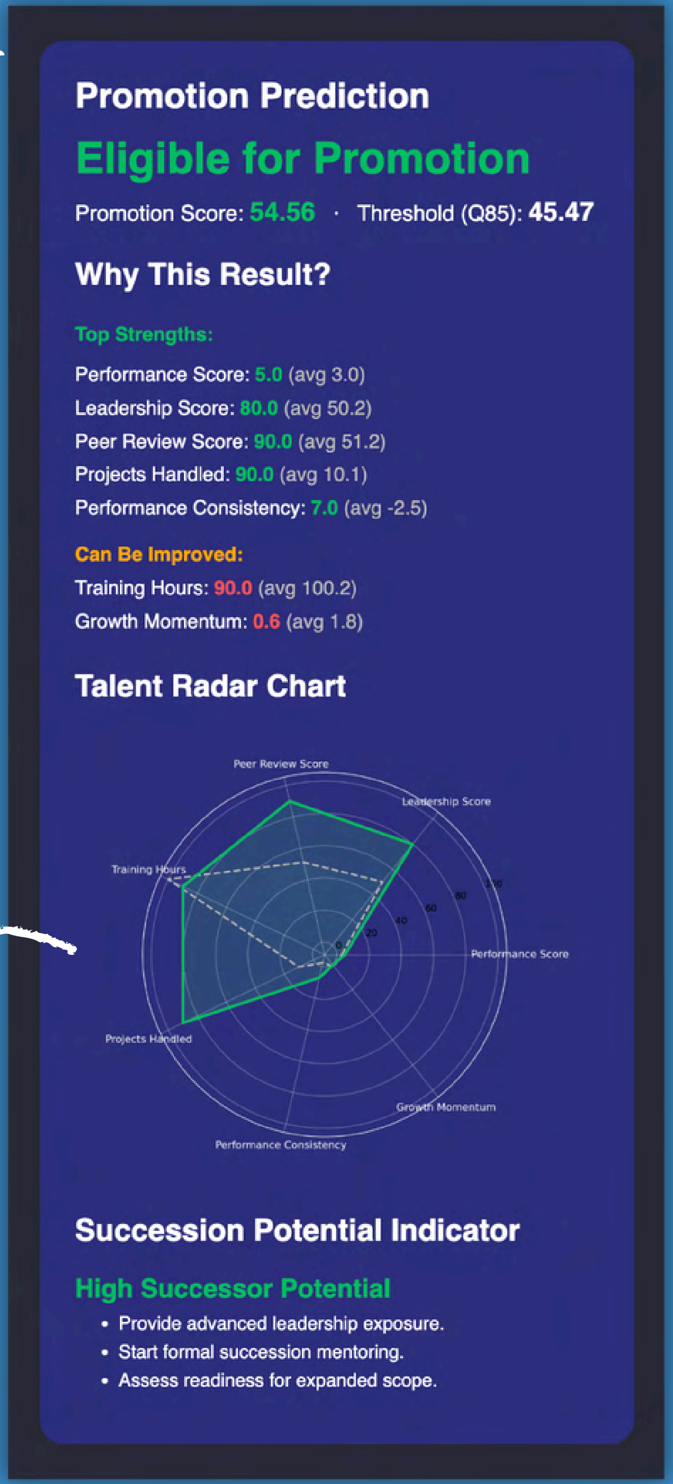
Data upload success notification



Browse uploaded data instantly on any device

Promotion eligibillity calculator can predict employee's readiness, strength and aspects that need to be improved.

Radar chart is shown for quick assessment of employee's quality.



Source: Streamlit

Model usage flow

#1 Talent data input

HR selects an employee or uploads a CSV file containing performance, leadership, and development metrics.

#2 Automated processing

The system applies feature engineering to compute key talent indexes such as performance, leadership, and potential.

#3 Talent clustering

Based on these indexes, the model groups employees into distinct talent clusters to highlight strengths, risks, and development profiles.

#4 Eligibility prediction

The rule-based model evaluates each employee and predicts their promotion readiness using combined performance–potential indicators.

#5 Insight delivery

The app displays clusters, key indexes, and promotion eligibility, along with recommended focus areas for HR decision-making.

#6 HR action & decision

HR uses these insights to identify promotable employees, develop top performers, and intervene early with high-risk talent.

Monitoring & retraining

We continuously monitor three things: shifts in employee data patterns (data drift), the health of the cluster structure, and how well the system's recommendations align with actual promotion decisions. When any metric crosses its threshold, it triggers a deeper analysis.

Monitoring strategy

Data drift

- Metric: Population Stability Index (PSI)
- PSI < 0.1 → Stable
- 0.1–0.25 → Run EDA
- > 0.25 → Retrain cluster

Cluster stability

- Metrics: Silhouette & DBI
- Silhouette decrease > 10% → EDA
- DBI increase > 15% → EDA
- Both degrade → Retrain model
- Cluster size shift > ±20% → Retrain

Rule drift

- Check % mismatch between rule output & actual promotions
- >10% mismatch → Review rules
- >20% mismatch → Update rules

Maintenance

Monthly: drift check
Quarterly: cluster & rule review
Semester: retrain if triggered
Annualy: audit system

Retraining Framework

- Refresh latest employee data
- Run EDA + drift analysis
- Refit clustering model
- Recompute Silhouette & DBI
- Update cluster profiles
- Review/update rule logic (if needed)
- Deploy + document changes

Business Impact & Recommendation.

Business impact

Talent management isn't just about fairness. It's about money. Every time you promote someone who isn't ready, you're paying a senior-level salary for below-par performance. Every time you pass over someone capable, you risk losing them.

⚠ High Risk Talent

These employees fall into lower performance ranges and may need targeted development, mentoring, or closer support.

Select ranking category: Low Performing

Filter by Position Level: All Levels

Employee_ID	Current_Position_Level	Performance_Index	Performance_Score	Cluster
EMP0010	Mid	1	1	1
EMP0178	Senior	1.3	1	1
EMP0974	Junior	1.8	1	1
EMP0910	Mid	2.1	1	1
EMP0810	Lead	2.6	3	1
EMP0241	Lead	2.7	1	1
EMP0939	Junior	2.7	1	1
EMP0251	Lead	2.8	2	1
EMP0656	Senior	2.8	4	1
EMP0829	Mid	3.1	3	1

Most of the workforce is in the Average Talent category, followed by High Performing Talent, and a fairly large proportion is Low Performing Talent at 219 employee which is the main cause of the estimated loss.

AVG salary based on Dealls!

Level	AVG Salary
Junior	Rp6.500.000
Middle	Rp11.500.000
Senior	Rp22.500.000
Lead	Rp27.500.000

Save

3.5 billions!

22%

Saving by not paying incompetent employees

42%

Improved accuracy in previously biased determinations of promotion eligibility.

Source: Dealls!

Business Recommendation

#1 Overhaul eligibility determination system.

Update it using reliable and reasonable metric calculations to minimize potential bias with online-based data collection.

#2 Conduct a rigorous performance review

Specifically for employees who fall into Cluster 2 (At-Risk & Underpowered) and make a decision between skill development or, at the worst, employment termination.

#3 Leverage the result of the Talent Clustering

Identify and map existing skills gaps. Focus development interventions on Clusters 1 and 2 to address the identified gaps. For Clusters 3 and 4, provide targeted training for retention.

Strategic roadmap

Next-Phase Opportunities

Expand predictive models to additional HR domains like attrition risk, promotion readiness

Scalability & Integration Roadmap

Develop API connections with HRIS, payroll, and attendance systems for real-time data flow

Future Capability Development

Create targeted upskilling programs aligned with model insights and skill gaps.

Thank You!

The Promotion Paradox:
Fixing biased talent promotion decisions through data-driven HR analytics

A project proposal by Syntax Society | Rakamin - Data Science batch 59