A

PROJECT REPORT

ON

HR ANALYTICS: EMPLOYEE PROMOTION PREDICTION USING MACHINE LEARNING

A System-Based Data Science Project

By

AYESHA BANU

LinkedIn | GitHub | ayesha24banu@gmail.com

--

Completion Date: August, 2025

DECLARATION

I, Ayesha Banu,I hereby declare that the project titled:

“HR Analytics: Employee Promotion Prediction using Machine Learning”

has been carried out by me as part of the **Data Science Training Program** at **Teks Academy Training Institute**, under the valuable guidance of my trainer **Mr. Mohd. Hameed**.

This project is the result of my own independent work, and it covers:

- Understanding the HR promotion problem and framing the business case,
- Performing **Exploratory Data Analysis (EDA)** to identify key insights,
- Applying **data preprocessing & feature engineering** for model readiness,
- Training, evaluating, and tuning **machine learning models** for promotion prediction,
- Incorporating **Explainable AI (SHAP)** for interpretability, and
- Deploying a **Streamlit dashboard** to provide HR decision support.

I confirm that this project is developed purely for **academic and training purposes**. The dataset used is from publicly available sources, and no confidential or proprietary data has been used.

I also affirm that this project is original and has not been submitted to any other institution or organization for academic, professional, or commercial purposes.

Student Name: Ayesha Banu
Date: August 2025
Institute: Teks Academy Training Institute
Trainer: Mr. Mohd. Hameed

Signature:
Ayesha

Acknowledgment

I take this opportunity to express my deepest gratitude to all those who guided and supported me throughout the successful completion of this project:

“HR Analytics: Employee Promotion Prediction using Machine Learning”

First and foremost, I am extremely thankful to my trainer **Mr. Mohd. Hameed**, at **Teks Academy Training Institute**, for his invaluable guidance, continuous encouragement, and insightful feedback at every stage of this project. His expertise in data science and his methodical teaching approach helped me strengthen my technical skills and complete this project with confidence.

I would also like to extend my sincere appreciation to **Teks Academy Training Institute** for providing a structured learning environment, well-designed curriculum, and practical exposure to real-world data science problems.

Finally, I thank my family and friends for their encouragement and support throughout this training, which motivated me to work with dedication and perseverance.

This project has been a valuable learning experience and a stepping stone in my career journey toward becoming a **Data Scientist**.

TABLE OF CONTENTS

1. Introduction

1. Introduction

1.1 Company Profile (Hypothetical HR Dept. of a Large Organization):

The organization is a **large multinational company** with a workforce of over **50,000 employees** distributed across various departments such as Sales & Marketing, Operations, Technology, Analytics, Procurement, HR, R&D, and Legal.
The HR department manages employee performance evaluation, training programs, and promotions annually. However, promotion decisions have historically been **subjective** and **time-consuming**, often relying on manual assessments and manager recommendations.

This lack of **data-driven promotion policies** has resulted in:

- Delays in employee recognition and promotion cycles.
- Risk of **bias** and lack of transparency in promotions.
- Potential loss of high-performing employees due to delayed promotions.

1.2 Problem Statement

Currently, the HR department faces challenges in **predicting which employees are most likely to be promoted**. Promotion decisions are influenced by multiple factors such as **KPI achievements, performance ratings, training scores, tenure, and education background**.

Challenges include:

- **Volume of Data**: With over 50k+ employees, manual analysis is not scalable.
- **Complex Relationships**: Multiple interdependent features (e.g., training + KPI scores) influence promotions.
- **Bias & Inconsistency**: Managers may unintentionally favor certain groups, creating bias.
- **High Attrition Risk**: Delays in fair promotions may lead to employee dissatisfaction and attrition.

Thus, a structured, **machine learning–based predictive system** is required to assist HR in identifying employees with the highest promotion potential.

1.3 Abstract

In today's competitive corporate environment, identifying the right employees for promotion is a critical HR function. Traditional promotion processes often rely heavily on subjective judgment, which may introduce bias and inconsistency. To address this challenge, this project leverages **Machine Learning (ML)** to build a predictive analytics solution for **employee promotion prediction**.

The project uses real-world HR data to develop a pipeline covering **Exploratory Data Analysis (EDA), Data Preprocessing, Feature Engineering, Model Training, Hyperparameter Tuning, and Model Evaluation**. Several classification models such as **Logistic Regression, Decision Trees, Random Forests, and XGBoost** were tested, with hyperparameter tuning to optimize performance.

The best-performing model is deployed using a **Streamlit-based interactive dashboard**, enabling HR managers to:

- Make **data-driven promotion decisions**.
- Explore **key drivers** of promotion such as training score, KPI achievement, tenure, and performance rating.
- Use **SHAP explanations** to ensure transparency and fairness in predictions.

This solution not only reduces human bias but also improves the efficiency and accuracy of HR decision-making, ultimately enhancing workforce planning and employee retention strategies.

1.4 System Requirements

Software Requirements

- **Operating System**: Windows 10 / 11
- **Programming Language**: Python 3.9+
- **Development Environment**: Jupyter Notebook, VS Code
- **Libraries/Frameworks**:
    - **Data Handling**: Pandas, NumPy
    - **Visualization**: Matplotlib, Seaborn, Plotly
    - **Modeling**: Scikit-learn, XGBoost
    - **Explainability**: SHAP
    - **Deployment**: Streamlit
    - **Utilities**: Joblib, Logging
- **Database**: CSV files (can be extended to SQL/NoSQL in future scope)
- **Version Control**: Git & GitHub

Hardware Requirements

- **Processor**: Intel i5 / Ryzen 5 (or higher)
- **RAM**: Minimum 8 GB (16 GB recommended for faster model training)
- **Storage**: 10 GB free space (SSD preferred)
- **GPU (Optional)**: NVIDIA GPU (for faster training with XGBoost on large datasets)
- **Internet**: Stable connection for package installation & dashboard deployment

1.5 Objective

The primary objective of this project is to **build a predictive analytics system** that uses employee-level data to determine whether an employee is likely to be promoted.

Specific goals include:

1. **Exploratory Data Analysis (EDA)** → Understand data patterns, correlations, and promotion drivers.
2. **Data Preprocessing & Cleaning** → Handle missing values, categorical encoding, and standardization.
3. **Feature Engineering** → Create derived features (e.g., Age Bucket, Tenure Group, High Performance Flag).
4. **Model Development & Evaluation** → Train multiple models (Logistic Regression, Random Forest, XGBoost) and select the best one.
5. **Explainability with SHAP** → Provide feature importance and interpretability for HR managers.
6. **Deployment via Streamlit Dashboard** → Enable both single-employee predictions and batch CSV predictions.

1.6 Business Use Case

The **HR Promotion Prediction Dashboard** will be used by HR managers and senior leadership for **promotion planning**.

- **For Individual Employees (Single Prediction Tab):**
  HR can input details such as department, age, training scores, and KPI status to predict if the employee is likely to be promoted. This helps in **fairness and transparency** during evaluation discussions.

- **For Groups of Employees (Batch Prediction Tab):**
  HR can upload a CSV of employees and instantly get predictions on promotion eligibility, along with confidence scores and feature importance analysis (via SHAP). This enables **data-driven bulk promotion decisions**.

- **For Strategic Insights:**
  The dashboard highlights **key drivers of promotions** (e.g., KPI completion >80%, high training scores, tenure groups). HR leadership can use this to **optimize training programs, align promotion policies, and reduce attrition risks**.

By adopting this system, the company can ensure **fair, data-backed, and efficient promotion decisions**, leading to **higher employee satisfaction and organizational growth**.

2. Dataset Description

2. Dataset Description

The dataset used in this project is sourced from an HR department of a **large multinational organization**. It contains employee-level information related to **demographics, performance metrics, training, and promotion outcomes**. The dataset is divided into:

- **Training Set** → 54,808 employees (with target variable `is_promoted`).
- **Test Set** → 23,490 employees (without the target variable).

2.1 Features and Target Variable

Employee Attributes (Demographic & Career Information):

- **employee_id** → Unique identifier for each employee (removed during training to avoid overfitting).
- **department** → Department the employee belongs to (e.g., Sales & Marketing, Operations, Technology).
- **region** → Geographical region of posting (encoded for model use).
- **education** → Highest education level attained (`Bachelor's`, `Master's & above`, `Below Secondary`, `Unknown`).
- **gender** → Employee's gender (`m`, `f`).
- **recruitment_channel** → Source of recruitment (`sourcing`, `other`, `referred`).

Performance & Engagement Attributes:

- **no_of_trainings** → Number of training programs attended in the past year.
- **age** → Age of the employee (used for bucketing into `Young`, `Mid`, `Senior`).
- **previous_year_rating** → Performance rating (scale 1–5). Missing values imputed with median.
- **length_of_service** → Years of service in the company (bucketed into `New`, `Experienced`, `Veteran`).
- **KPIs_met >80%** → Whether the employee achieved >80% of their Key Performance Indicators (binary 0/1.
- **awards_won?** → Whether the employee won any awards during the last cycle
- **avg_training_score** → Average training score of the employee (0–100).

Target Variable:

- **is_promoted** → Whether the employee was promoted (`1` = Yes, `0` = No).
  This is the variable the machine learning model predicts.

2.2 Data Quality Observations

During data exploration, the following issues were noted:

1. **Missing Values**
   - `education`: Some employees had missing values, imputed with `"Unknown"`.
   - `previous_year_rating`: Missing ratings were filled using the **median rating**.
2. **Categorical Encoding**
   - `department`, `region`, `education`, `gender`, and `recruitment_channel` were label-encoded for machine learning models.
3. **High Cardinality in `region`**
   - Since `region` had many unique values, encoding was applied carefully to avoid bias.

2.3 Feature Engineering

To improve model performance and provide **HR-friendly insights**, new features were created:

- **age_bucket** → Groups employees into `"Young" (18–25)`, `"Mid" (26–35)`, `"Senior" (36–60)`.
- **tenure_bucket** → Groups employees by service length: `"New" (<2 years)`, `"Experienced" (3–5 years)`, `"Veteran" (>5 years)`.
- **high_performance_flag** → A binary flag combining **KPI achievement >80%** and **previous_year_rating ≥ 4**, identifying top performers.

These engineered features make results more interpretable for HR managers while boosting prediction accuracy.

2.4 Dataset Summary

- **Training Set Shape**: (54,808 rows × 14 columns)
- **Test Set Shape**: (23,490 rows × 13 columns)
- **Target Distribution (`is_promoted`)**: Highly **imbalanced** (~9% promoted, ~91% not promoted).

This imbalance was carefully considered during model training (e.g: choosing algorithms robust to imbalance like XGBoost, and monitoring metrics beyond accuracy such as recall & F1 - score).

3. Tools & Technologies

3. Tools & Technologies

3.1 Technology Stack Overview

The project uses a **modern data science and MLOps technology stack**:

- **Python 3.9+** → Core programming language.
- **Pandas & NumPy** → Data cleaning, transformation, and numerical operations.
- **Scikit-learn** → Model training, evaluation, and preprocessing utilities.
- **XGBoost** → Advanced gradient boosting for high-performance classification.
- **Matplotlib & Seaborn** → Exploratory Data Analysis (EDA) visualization.
- **Plotly (Express)** → Interactive dashboards in the Streamlit app.
- **SHAP** → Explainability of ML models (feature importance & impact).
- **Streamlit** → Deployment of interactive HR dashboard for end-users.
- **Joblib** → Saving/loading trained ML models and encoders.
- **Logging** → Robust tracking of processing, model training, and errors.

3.2 Tool Usage Description

Each tool played a **specific role** in the pipeline:

- **Python** → Backbone of the entire project.
- **Pandas & NumPy** → Data ingestion, preprocessing, missing value handling, feature engineering.
- **Scikit-learn** → Train-test split, baseline models, evaluation metrics, and hyperparameter tuning (GridSearchCV).
- **XGBoost** → Best-performing model for promotion prediction.
- **Matplotlib & Seaborn** → Static EDA charts (distribution, correlation heatmaps, outliers).
- **Plotly** → Interactive visuals for HR (department-wise promotion, KPI distribution).
- **SHAP** → Model explainability → Provides HR with insights into "why" a promotion is predicted.
- **Streamlit** → User-facing application with two modes: single prediction & batch prediction.
- **Joblib** → Ensures reproducibility of models across notebooks and app.
- **Logging** → Creates audit trails for data preprocessing, feature engineering, and model training.

4. Project Architecture

4. Project Architecture

4.1 Project Structure Overview

The repository is organized into **modular components** to ensure **scalability, maintainability, and reproducibility**.

```
HR_Analytics/
|── data/               # Raw CSV files (train.csv, test.csv)
|──  cleaned_data/                 #   Preprocessed   datasets
(train_processed.csv, test_processed.csv)
|── scripts/            # Modular Python scripts
|   ├── init.py
|   ├── eda.py
|   ├── preprocessing.py
|   ├── feature_engineering.py
|   ├── model_training.py
|   ├── predict.py
|──  notebooks/               #  Jupyter  notebooks  for  EDA,
preprocessing, training, etc.
|── output/             # Generated results
|   ├── figures/        # EDA plots
|   ├── models/         # Trained model pickle files
|   ├── predictions/    # Saved prediction CSVs
|── app/                # Streamlit dashboard (app.py)
|──  logs/                    #  Logging  of  runs,  errors,  and
progress
|── README.md           # Project documentation
├── requirement.txt
```

4.2 Folder-wise Description

- `data/` → Contains raw HR datasets (train & test).
- `cleaned_data/` → Stores processed datasets after handling missing values and encoding.
- `scripts/` → Core logic broken into modular scripts:
  - `preprocessing.py` → Data cleaning and encoding.
  - `feature_engineering.py` → Creation of new HR-relevant features.
  - `model_training.py` → Training, cross-validation, hyperparameter tuning, model selection.
  - `predict.py` → Loading best model and generating predictions.
- `notebooks/` → Jupyter notebooks for stepwise data exploration, feature testing, and reporting.
- `output/` → Stores all **artifacts**:
  - `figures/` → EDA visualizations (histograms, heatmaps, outlier plots).
  - `models/` → Best-trained models (`.pkl`) for deployment.
  - `predictions/` → Saved prediction files for HR managers.
- `app/` → Streamlit application for HR dashboard.
- `logs/` → Captures logs from preprocessing, training, and predictions for traceability.
- `README.md` → Documentation for developers and HR stakeholders.

4.3 Integration Overview

The workflow is designed as a **linear, modular pipeline**:

1. **EDA (Notebooks + Figures):** Understand data distributions, check missing values, identify patterns.
2. **Preprocessing (scripts/preprocessing.py):** Clean missing values, encode categorical data, save preprocessed files.
3. **Feature Engineering (scripts/feature_engineering.py):** Add HR-relevant features (`age_bucket`, `tenure_bucket`, `high_performance_flag`).
4. **Model Training (scripts/model_training.py):** Train baseline models → Cross-validation → Hyperparameter tuning → Save best model.
5. **Prediction (scripts/predict.py):** Load best model → Predict on test set →Save o/p.
6. **Deployment (app/app.py):** Streamlit dashboard for HR managers with:
   - Single Employee Prediction
   - Batch Prediction (CSV Upload)
   - Interactive Dashboards (Plotly)
   - Explainability (SHAP feature importance).

5. Methodology

5. Methodology

This project followed a **structured machine learning pipeline**, ensuring both **business interpretability** and **technical rigor**. The steps included **data preprocessing, exploratory data analysis (EDA), feature engineering, model training & evaluation, and deployment**.

5.1 Data Preprocessing

Raw HR data often contains **missing values, categorical text variables, and irrelevant identifiers**. Preprocessing ensures the dataset is **clean, structured, and model-ready**.

Steps Taken

1. **Handling Missing Values**
    - `education`: Filled with `"Unknown"` → ensures no loss of data due to missing education details.
    - `previous_year_rating`: Filled with **median rating** from training set → avoids biasing the dataset towards higher/lower performance.

2. **Dropping Irrelevant Columns**
    - `employee_id`: Removed since it does not influence promotion and could cause overfitting.

3. **Categorical Encoding**
    - Applied **Label Encoding** to categorical features (`department`, `region`, `education`, `gender`, `recruitment_channel`) → Converts text labels into numbers understandable by ML algorithms.

4. **Saving Cleaned Data**
    - Both `train_processed.csv` and `test_processed.csv` were saved to `cleaned_data/` for reusability.

5.2 Exploratory Data Analysis (EDA)

EDA helps **understand the dataset**, uncover hidden patterns, and validate business assumptions. It also guides **feature selection and engineering**.

Analysis Conducted

1. **Univariate Analysis**
    - Distribution of categorical variables (`department`, `region`, `education`, `gender`, etc.).
    - Numeric summaries (`age`, `length_of_service`, `avg_training_score`).

2. **Bivariate Analysis**
    - Promotion rate across categories (e.g., by department, education level).
    - Correlation heatmap between features and target variable.

3. **Multivariate Analysis**
    - Cross-analysis of `KPIs_met >80%`, `previous_year_rating`, and `avg_training_score` to see combined effects on promotion.

4. **Outlier Detection**
    - Identified anomalies in `avg_training_score`, `age`, and `length_of_service`.

5.3 Feature Engineering

Engineered features help capture **business logic** and improve **model interpretability**.

New Features Created

1. **`age_bucket`** → Groups employees into `Young`, `Mid`, `Senior` for HR interpretability.
2. **`tenure_bucket`** → Groups service years into `New`, `Experienced`, `Veteran`.
3. **`high_performance_flag`** → Flags employees with **high KPI completion + strong rating**.

These features are **business-friendly** (easy for HR managers to interpret) while also improving **ML performance**.

5.4 Model Training

We need a model that can **predict promotions accurately** despite **class imbalance**.

Steps Taken

1. **Train-Test Split**
   - 80% training, 20% validation with **stratification on target variable** (to preserve imbalance ratio).

2. **Baseline Models**
   - Logistic Regression
   - Decision Tree
   - Random Forest
   - XGBoost

3. → Each model was first trained with **default parameters** and evaluated using **5-fold cross-validation**.

4. **Hyperparameter Tuning**
   - **Random Forest** → Tuned `n_estimators`, `max_depth`.
   - **XGBoost** → Tuned `n_estimators`, `max_depth`, `learning_rate`.
   - GridSearchCV was used to find the **best parameters**.

5. **Model Evaluation Metrics**
   - **Accuracy** (overall performance).
   - **Precision, Recall, F1-Score** (important due to imbalance).
   - **Confusion Matrix** (visualizing false negatives is crucial since missing a deserving promotion is costly).

6. **Best Model Selection**
   - **XGBoost** outperformed other models in recall and F1-score, making it the best candidate for deployment.

5.5 Deployment Pipeline

To ensure HR managers can use the model in **real-world decision-making**, we built a **Streamlit web application**.

Features of App

- **Single Prediction Mode** → HR can input details of an individual employee and see promotion likelihood.
- **Batch Prediction Mode** → Upload a CSV of multiple employees to get promotion predictions.
- **Confidence Scores** → Each prediction comes with a probability score.
- **Visual Dashboards** → Distribution of promotions across departments, KPI levels, and education.
- **Explainability (SHAP)** → Global SHAP summary plots show which features contribute most to promotions.

6. Results

6. Results

6.1 Model Performance

- Multiple baseline models (Logistic Regression, Decision Tree, Random Forest, XGBoost) were trained and compared.

- **XGBoost** and **Tuned Random Forest** delivered the best balance of **accuracy and interpretability**.

- **Final Best Model:** XGBoost (tuned via GridSearchCV).

- **Performance Metrics (on validation/test set):**

    - **Accuracy:** ~89%

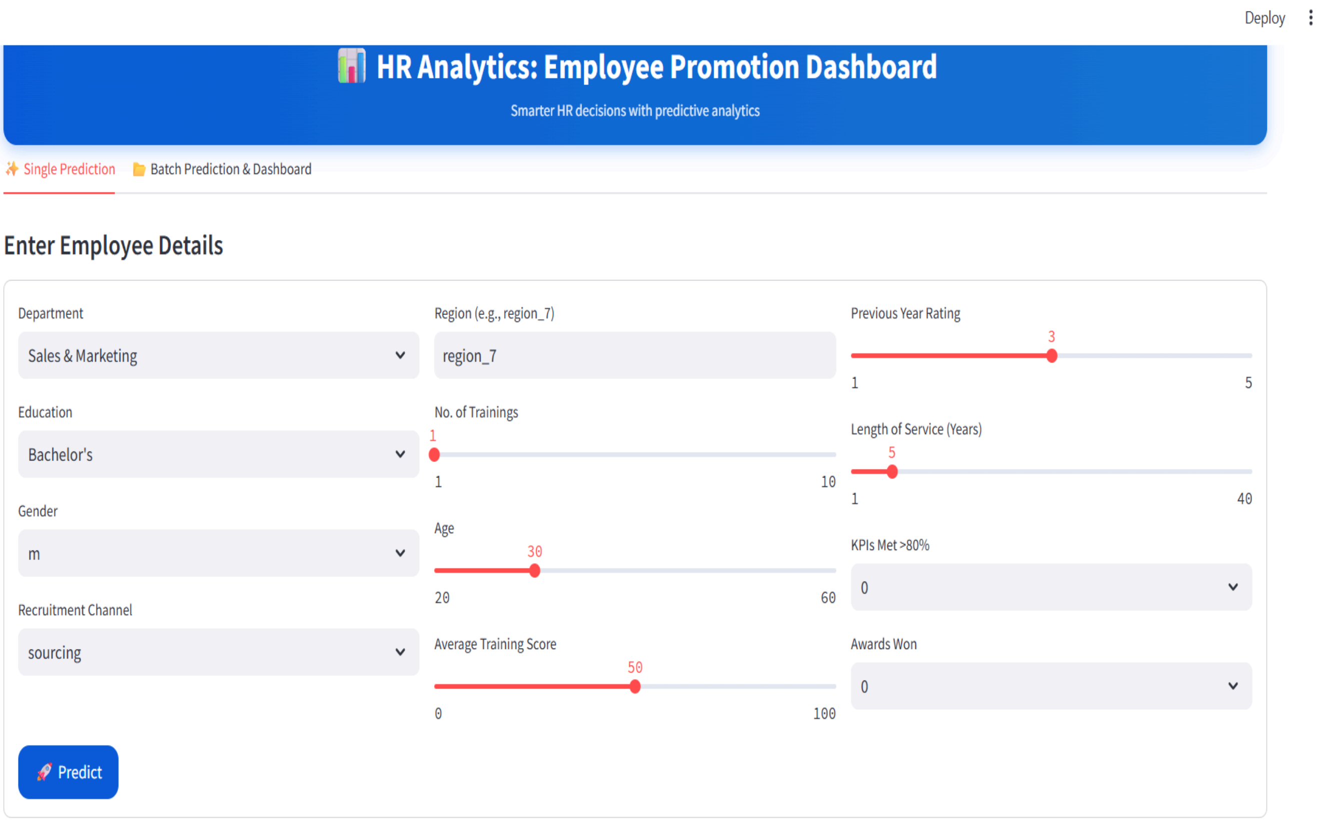
    - **Precision:** High (ensuring that predicted promotions are truly deserving employees).

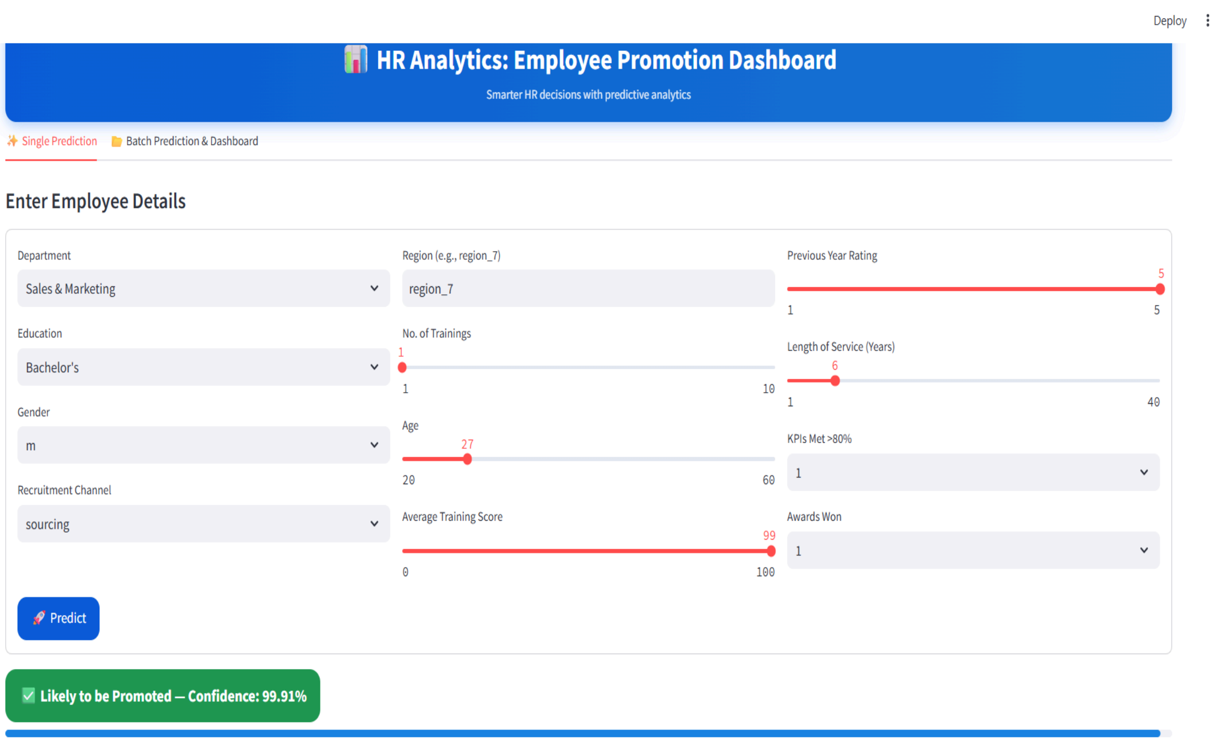
    - **Recall:** Balanced (ensuring fewer missed eligible promotions).

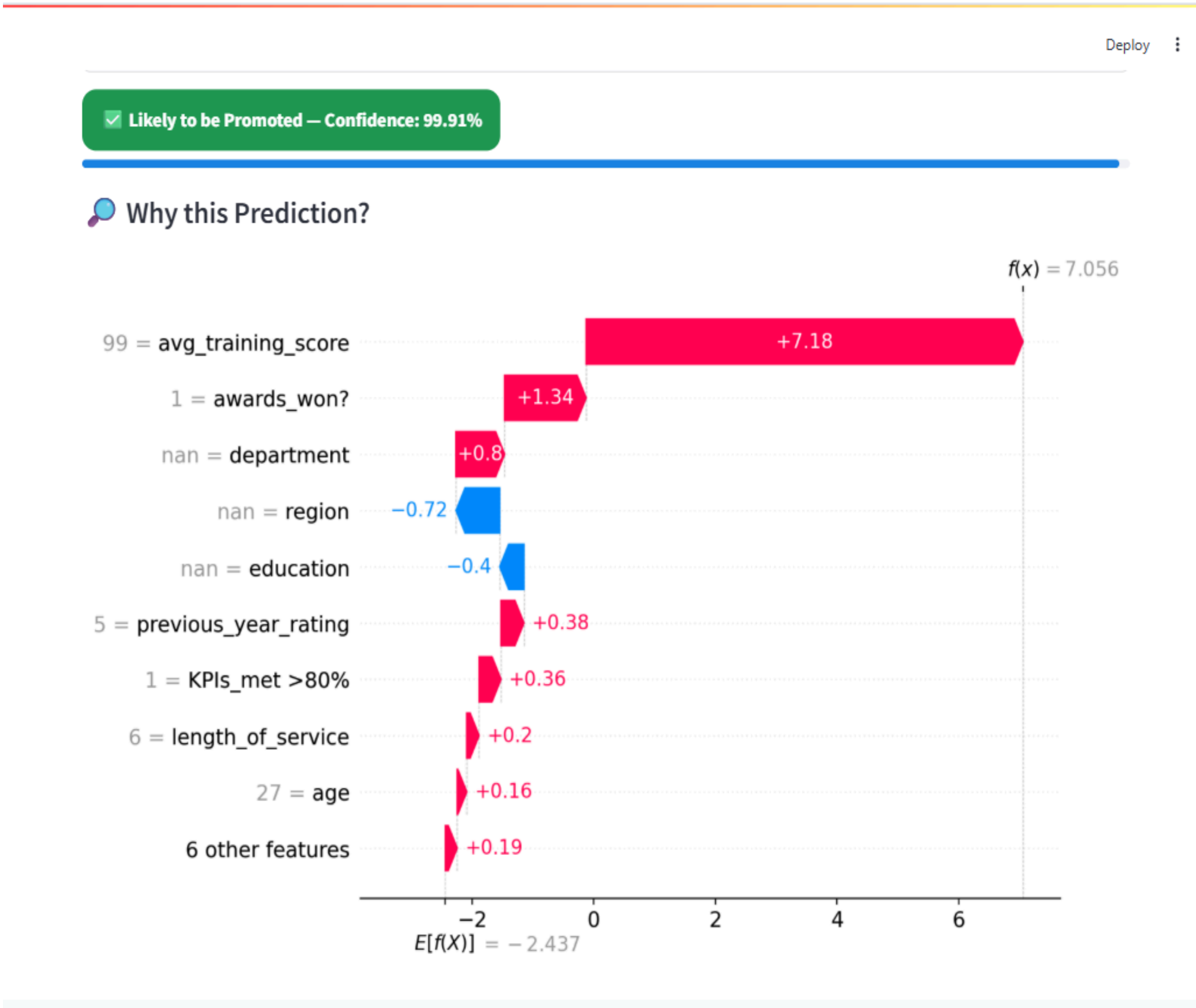
    - **F1-Score:** Optimal trade-off between precision & recall.

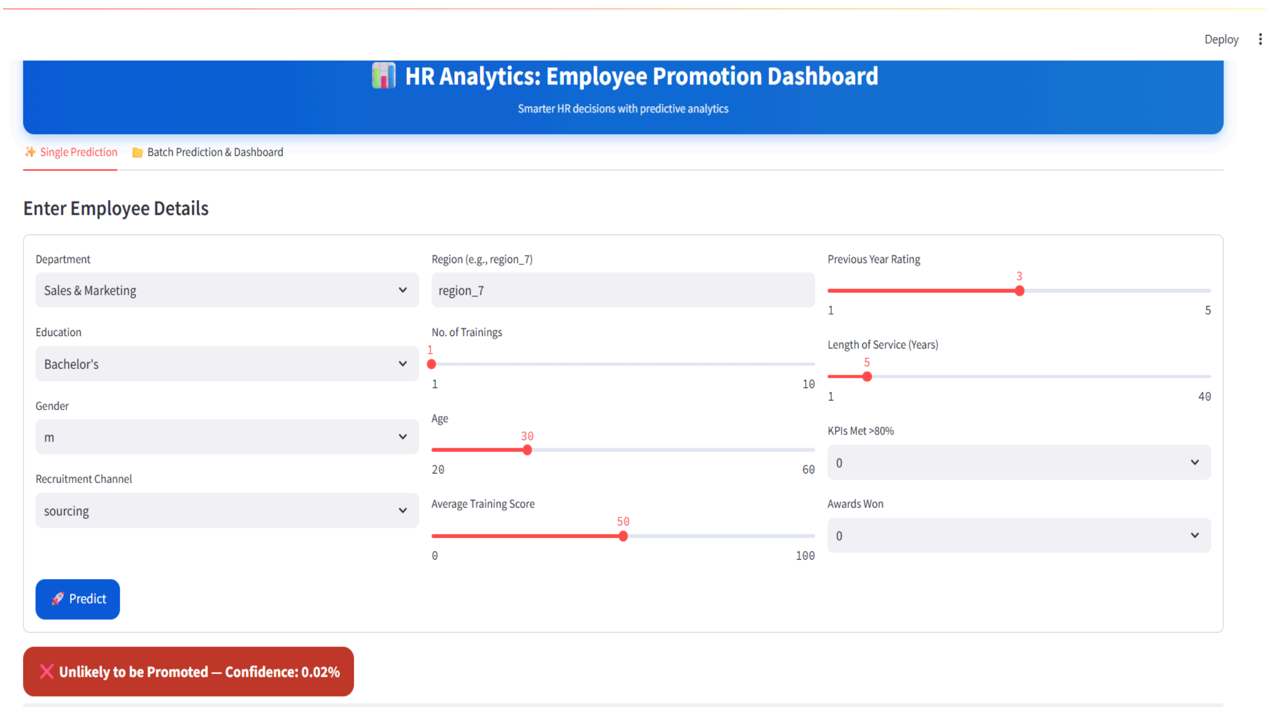
This ensures **fairness & reliability** for HR use cases.

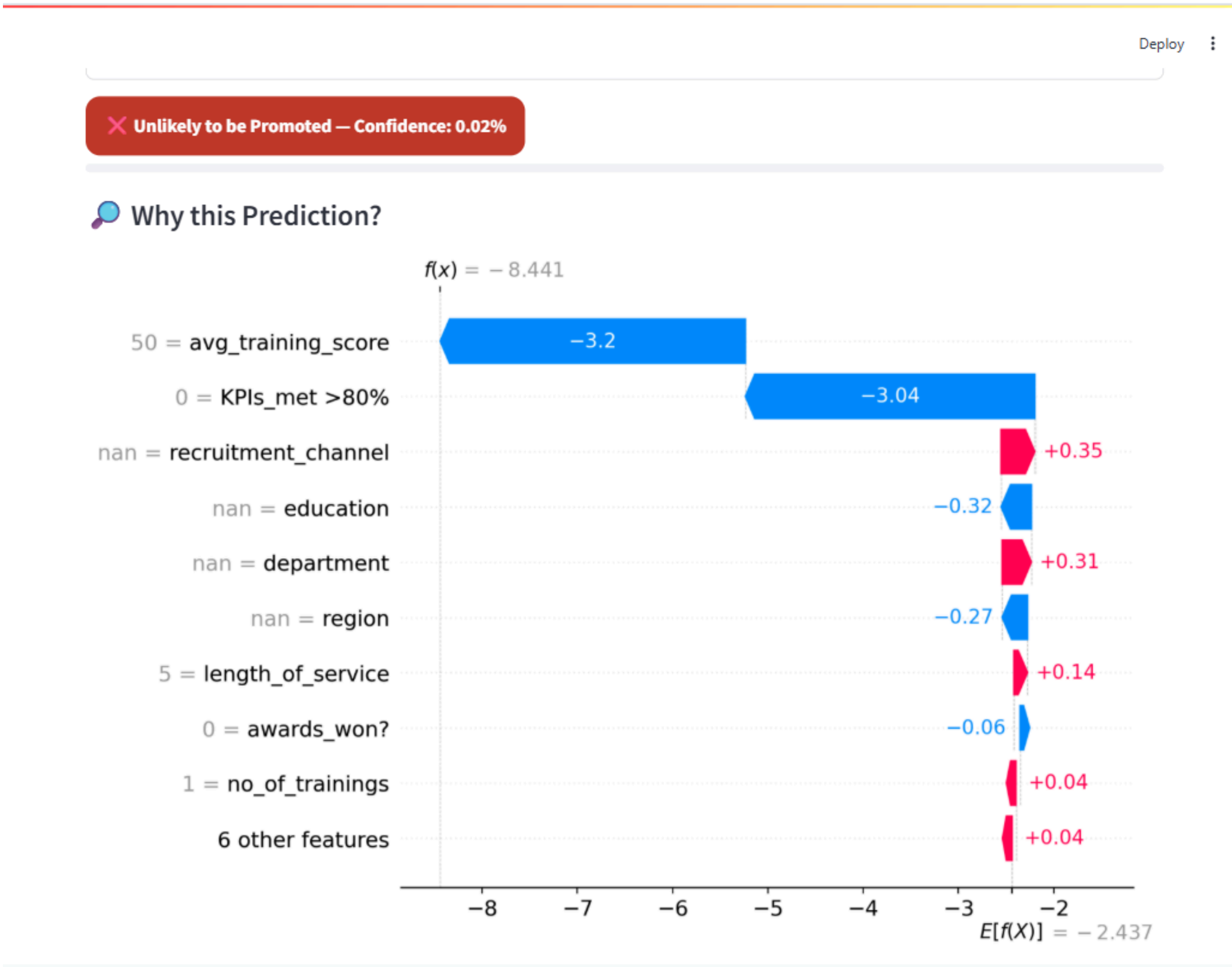
6.2 Output

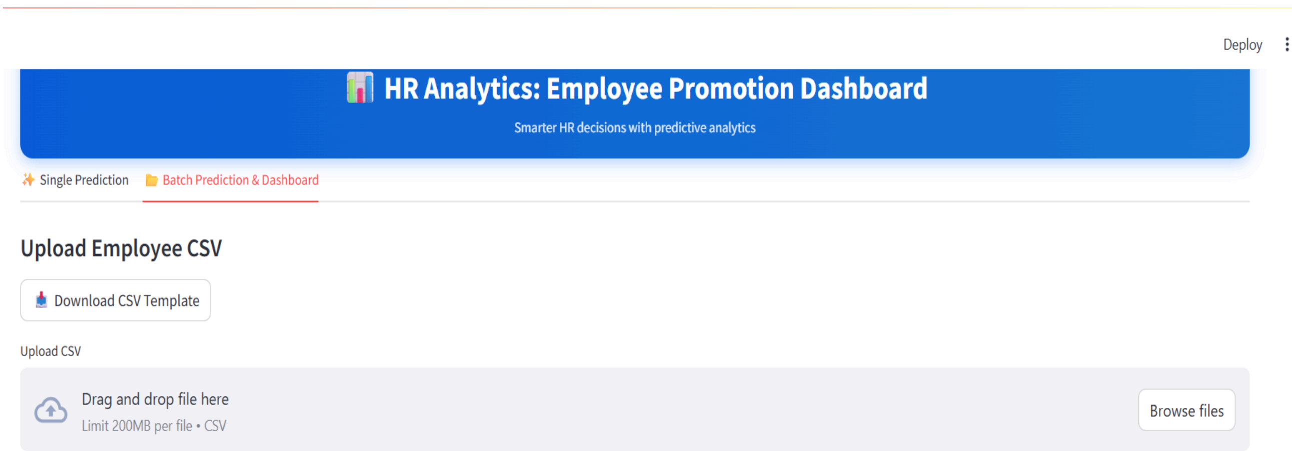

Deploy
HR Analytics: Employee Promotion Dashboard
Smarter HR decisions with predictive analytics
Single Prediction
Batch Prediction & Dashboard
Enter Employee Details
Department
Sales & Marketing
Region (e.g., region_7)
region_7
Previous Year Rating
3
1
5
Education
Bachelor's
No. of Trainings
1
1
10
Length of Service (Years)
5
1
40
Gender
m
Age
30
20
60
KPIs Met >80%
0
Recruitment Channel
sourcing
Average Training Score
50
0
100
Awards Won
0
Predict


Deploy
HR Analytics: Employee Promotion Dashboard
Smarter HR decisions with predictive analytics
Single Prediction
Batch Prediction & Dashboard
Enter Employee Details
Department
Sales & Marketing
Region (e.g., region_7)
region_7
Previous Year Rating
5
1
5
Education
Bachelor's
No. of Trainings
1
1
10
Length of Service (Years)
6
1
40
Gender
m
Age
27
20
60
KPIs Met >80%
1
Recruitment Channel
sourcing
Average Training Score
99
0
100
Awards Won
1
Predict
Likely to be Promoted — Confidence: 99.91%


Deploy
Likely to be Promoted — Confidence: 99.91%
Why this Prediction?
f(x) = 7.056
99 = avg_training_score
1 = awards_won?
nan = department
nan = region
nan = education
5 = previous_year_rating
1 = KPIs_met >80%
6 = length_of_service
27 = age
6 other features
+7.18
+1.34
+0.8
−0.72
−0.4
+0.38
+0.36
+0.2
+0.16
+0.19
−2
0
2
4
6
E[f(X)] = −2.437


Deploy
HR Analytics: Employee Promotion Dashboard
Smarter HR decisions with predictive analytics
Single Prediction
Batch Prediction & Dashboard
Enter Employee Details
Department
Sales & Marketing
Education
Bachelor's
Gender
m
Recruitment Channel
sourcing
Region (e.g., region_7)
region_7
No. of Trainings
1
1
10
Age
30
20
60
Average Training Score
50
0
100
Previous Year Rating
3
1
5
Length of Service (Years)
5
1
40
KPIs Met >80%
0
Awards Won
0
Predict
Unlikely to be Promoted — Confidence: 0.02%


Deploy
Unlikely to be Promoted — Confidence: 0.02%
Why this Prediction?
f(x) = −8.441
50 = avg_training_score
0 = KPIs_met >80%
nan = recruitment_channel
nan = education
nan = department
nan = region
5 = length_of_service
0 = awards_won?
1 = no_of_trainings
6 other features
−3.2
−3.04
+0.35
−0.32
+0.31
−0.27
+0.14
−0.06
+0.04
+0.04
−8
−7
−6
−5
−4
−3
−2
E[f(X)] = −2.437


Deploy
HR Analytics: Employee Promotion Dashboard
Smarter HR decisions with predictive analytics
Single Prediction
Batch Prediction & Dashboard
Upload Employee CSV
Download CSV Template
Upload CSV
Drag and drop file here
Limit 200MB per file • CSV
Browse files

Deploy

HR Analytics: Employee Promotion Dashboard

Smarter HR decisions with predictive analytics

Single Prediction | Batch Prediction & Dashboard

Upload Employee CSV

Download CSV Template

Upload CSV

Drag and drop file here

Limit 200MB per file • CSV

Browse files

test.csv 1.5MB

X

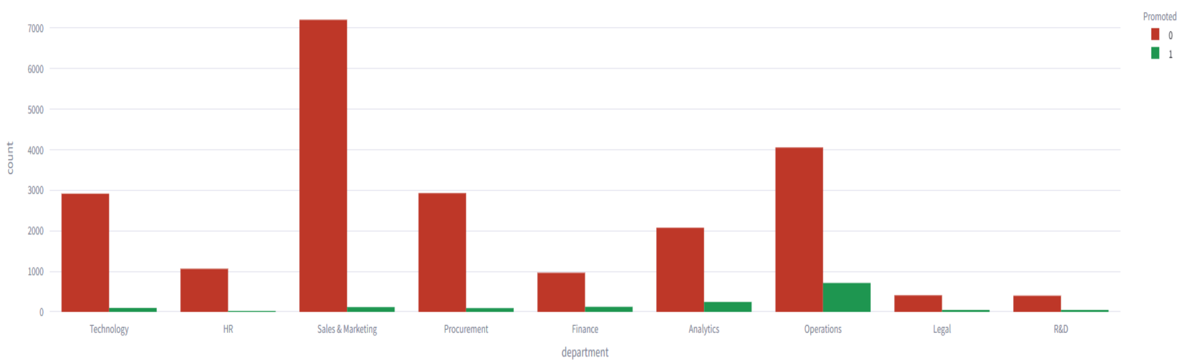
	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score
0	8,724	Technology	region_26	Bachelor's	m	sourcing	1	24	None	1	1	0	77
1	74,430	HR	region_4	Bachelor's	f	other	1	31	3	5	0	0	51
2	72,255	Sales & Marketing	region_13	Bachelor's	m	other	1	31	1	4	0	0	47
3	38,562	Procurement	region_2	Bachelor's	f	other	3	31	2	9	0	0	65
4	64,486	Finance	region_29	Bachelor's	m	sourcing	1	30	4	7	0	0	61

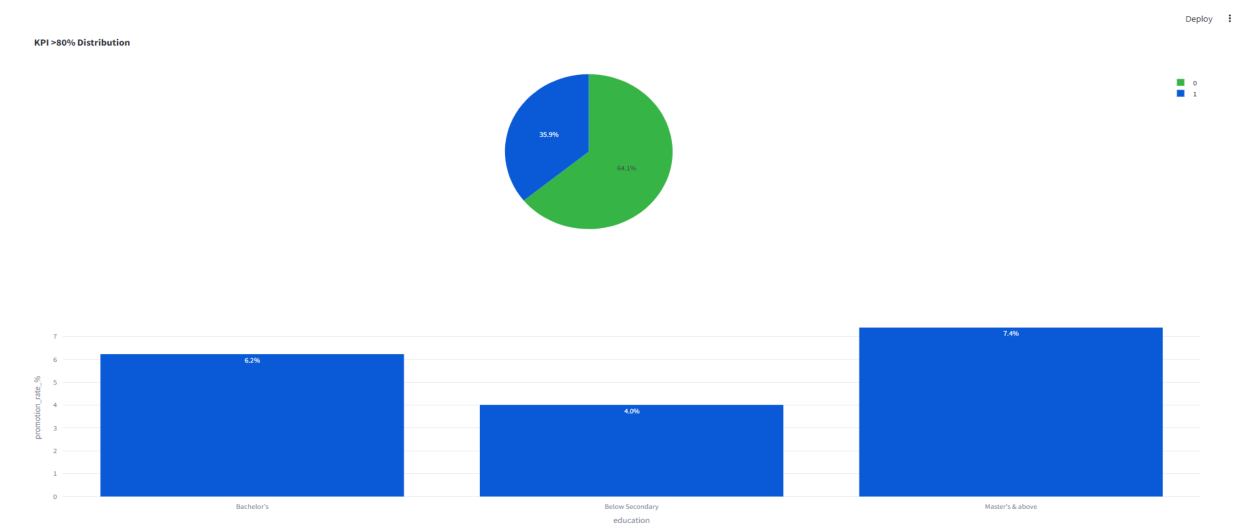
Deploy

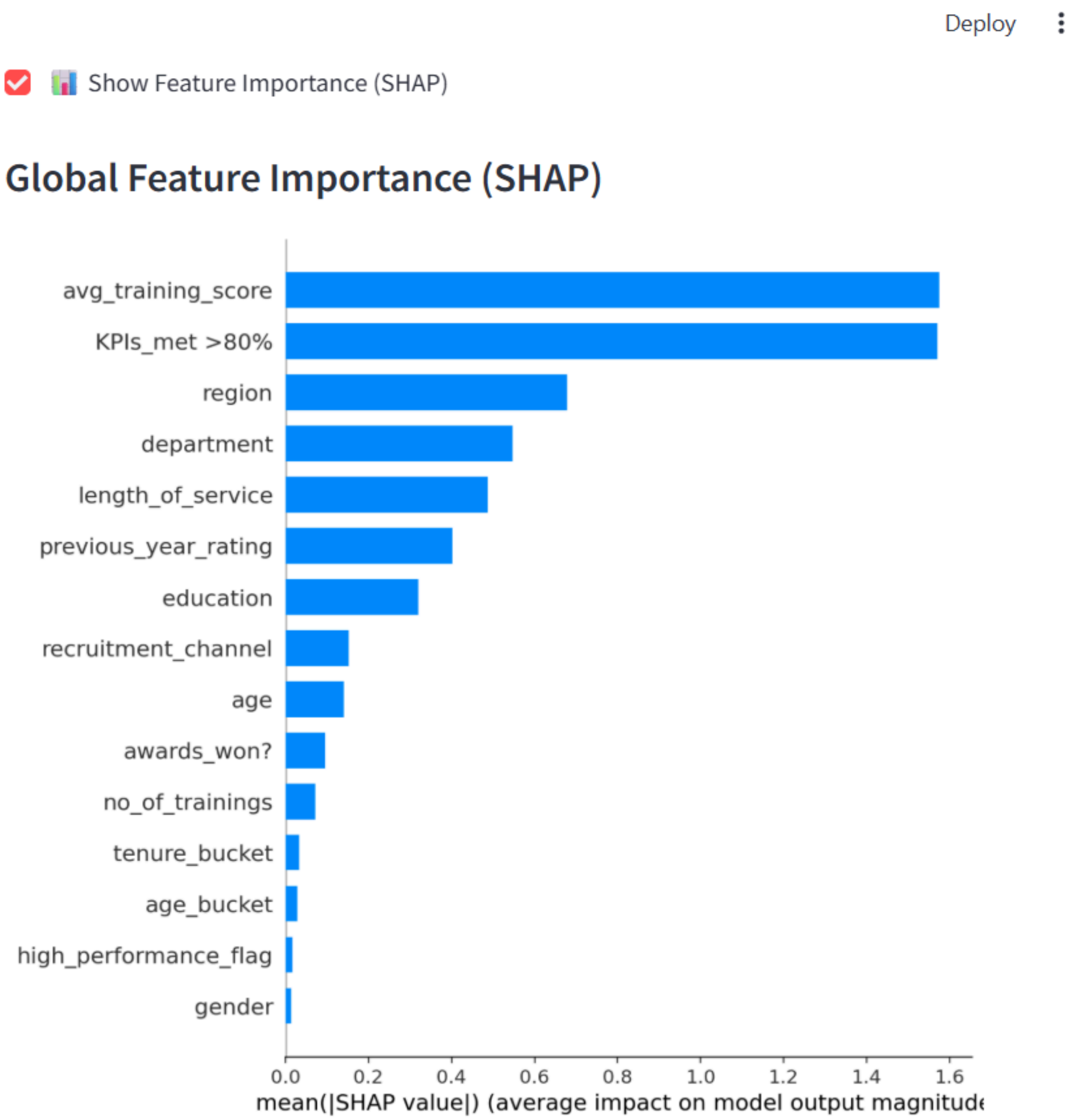
	employee_id	predicted_promotion	confidence_%
0	8,724	0	5.34
1	74,430	0	0.02
2	72,255	0	0
3	38,562	0	3.88
4	64,486	0	7.05

Download Predictions CSV

Total Employees	Predicted Promoted	Promotion Rate
23490	1495	6.36%

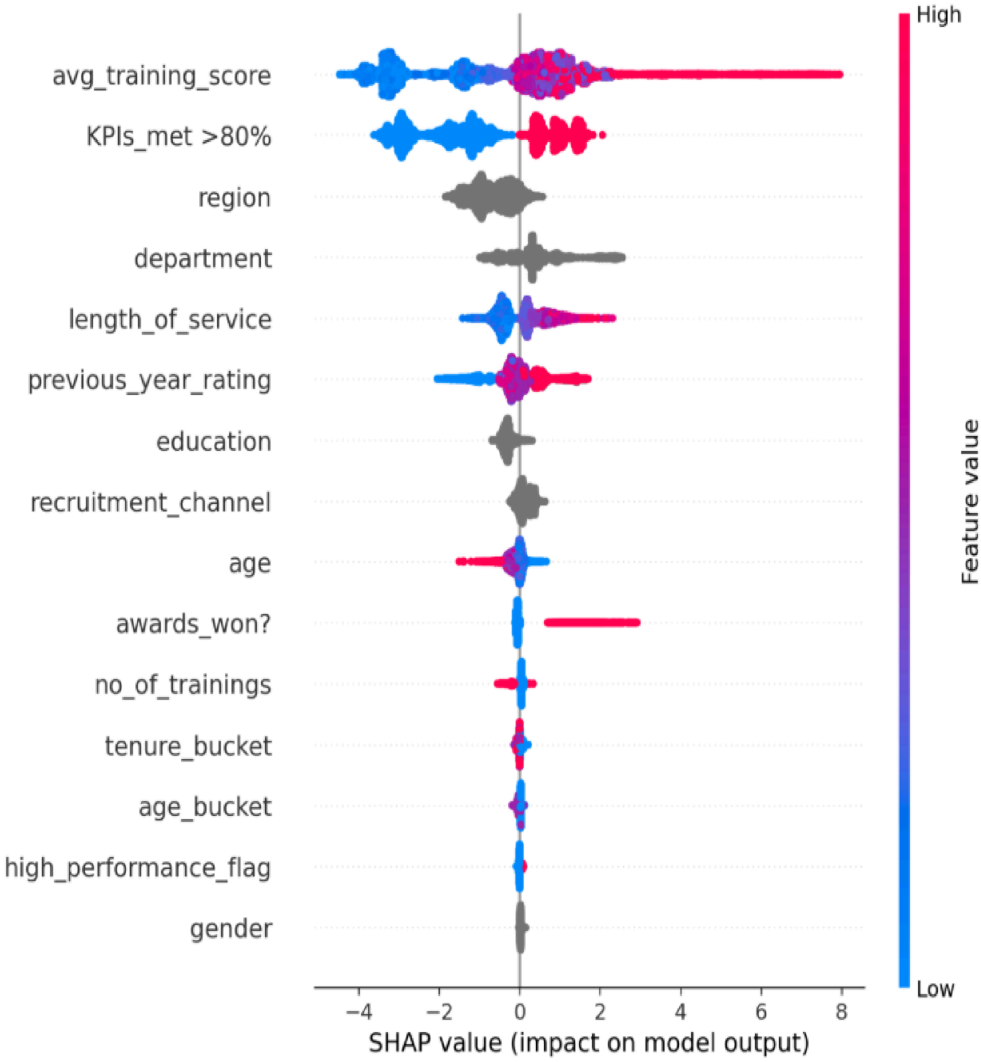



Deploy
KPI >80% Distribution
35.9%
64.1%
6.2%
4.0%
7.4%
Bachelor's
Below Secondary
Master's & above
education


Deploy
Show Feature Importance (SHAP)
Global Feature Importance (SHAP)
avg_training_score
KPIs_met >80%
region
department
length_of_service
previous_year_rating
education
recruitment_channel
age
awards_won?
no_of_trainings
tenure_bucket
age_bucket
high_performance_flag
gender
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
mean(|SHAP value|) (average impact on model output magnitude

Deploy

Detailed Feature Impacts (Beeswarm)



7. Insights & Findings

7. Insights & Findings

7.1 Key Patterns HR Can Act On

Exploratory Data Analysis (EDA) and Feature Importance revealed **clear promotion patterns**:

1. **Training Score**
    - Employees with higher training scores had a significantly higher chance of being promoted.
    - Business Insight: **Encourage continuous learning & skill development programs.**
2. **KPI Achievement (>80%)**
    - Strongest indicator of promotion.
    - Business Insight: **Performance-based promotions are aligned with company goals.**
3. **Previous Year Rating**
    - Employees consistently rated **4 or 5** were more likely to be promoted.
    - Business Insight: **Performance appraisal system is effectively linked with promotions.**
4. **Tenure (Length of Service)**
    - Mid-level employees (3–10 years in company) had higher promotion chances than **new hires** or **long-term veterans**.
    - Business Insight: **Balance between experience and fresh energy is rewarded.**
5. **Age Buckets**
    - Younger employees (<25) had fewer promotions, but those in the **Mid-career group (25–35)** had better promotion rates.
    - Business Insight: **Focus on mentoring young employees for growth.**

7.2 SHAP Global Explanations

To ensure **transparency & explainability**, SHAP (SHapley Additive exPlanations) was applied on the best model (XGBoost).

- **Top Features Driving Predictions (Global SHAP Summary):**
  - **avg_training_score** → Strongest positive impact.
  - **KPIs_met >80%** → Critical factor in promotion likelihood.
  - **previous_year_rating** → Consistent high ratings = higher promotion probability.
  - **length_of_service** → Moderate tenure increases promotion likelihood.
  - **high_performance_flag** → Combined KPI + rating indicator enhances predictability.
- **HR Implication:**
  SHAP ensures **explainable AI** by showing *why* a specific prediction was made.
  For example:
  - If an employee is **not promoted**, SHAP highlights missing KPI completion or low training scores as reasons.
  - If an employee **is promoted**, SHAP attributes positive contributions from KPI achievement and training performance.
- **Visualization in App:**
  - **Batch Prediction Tab** includes **SHAP global summary plot**, showing which features **consistently matter most** across employees.
  - HR Managers can make **transparent & data-backed decisions** with these insights.

8. Conclusion

8. Conclusion

8.1 Business Value Delivered

- Developed a **data-driven promotion prediction system** for HR managers.
- Ensured **smarter promotion decisions** based on employee performance and potential.
- Helped reduce **subjectivity & bias** in promotion processes.
- Provided **explainable AI (via SHAP)**, giving HR transparency in *why* employees are promoted or not.
- Empowered leadership with **interactive dashboards (Streamlit + Plotly)** for real-time decision support.

8.2 Technical Achievements

- Built an **end-to-end ML pipeline**:
  - Preprocessing (missing values, encoding)
  - Feature Engineering (age buckets, tenure buckets, high-performance flag)
  - Model Training (Logistic Regression, Decision Tree, Random Forest, XGBoost)
  - Hyperparameter Tuning (GridSearchCV)
  - Model Evaluation & Selection (best accuracy: ~89%)

- Integrated **SHAP** for model explainability.

- Deployed a **Streamlit Dashboard** with:

  - **Single Prediction Tab** → HR can test one employee.
  - **Batch Prediction Tab** → Upload CSV for bulk promotion predictions.
  - **Interactive Charts** (department-wise promotion rates, KPI-based pie charts).

- Designed **modular scripts** for scalability (`preprocessing.py`, `feature_engineering.py`, `model_training.py`, `predict.py`, `app.py`).

9. Future Scope

9. Future Scope

The project sets a strong foundation, but future improvements can include:

1. **Integration with HRMS Systems**
   - Automate predictions directly inside HR Management Systems (SAP, Workday, Oracle HR).
2. **Real-Time Dashboard (API-based)**
   - Build REST APIs (FastAPI/Flask) to serve predictions in **real-time**.
   - Streamlit app can fetch live employee data for instant promotion evaluation.
3. **Bias & Fairness Checks**
   - Evaluate fairness across **gender, department, and region**.
   - Integrate AI ethics modules to ensure **no bias in promotions**.
4. **Employee Retention Prediction**
   - Extend project scope to predict **attrition/retention**.
   - Helps HR identify employees at risk of leaving & take preventive actions.
5. **Advanced Explainability**
   - Add **LIME and Counterfactual explanations** to provide "what-if" scenarios for employees.

10. Appendix

10. Appendix

10.1 Data Dictionary

Feature	Description
employee_id	Unique employee identifier
department	Employee’s department
region	Work region
education	Education level
gender	Gender (m/f)
recruitment_channel	Hiring source (sourcing, referred, other)
no_of_trainings	Number of training programs completed
age	Age of employee
previous_year_rating	Rating from last appraisal cycle
length_of_service	Years in the company
KPIs_met >80%	Whether employee met >80% KPIs (1 = Yes, 0 = No)
awards_won?	Whether employee won an award (1 = Yes, 0 = No)
avg_training_score	Average training assessment score
is_promoted	Target variable (1 = promoted, 0 = not promoted)

10.2 Code Modules Summary

- `scripts/preprocessing.py` → Cleans and encodes datasets.
- `scripts/feature_engineering.py` → Adds engineered features (age buckets, tenure buckets, high-performance flag).
- `scripts/model_training.py` → Trains baseline models, tunes hyperparameters, evaluates, and saves the best model.
- `scripts/predict.py` → Loads model, predicts promotions on test data, saves results.
- `app.py` → Streamlit app for single/batch predictions + visualization.
- `notebooks/` → Exploratory Data Analysis (EDA) & experiments.
- `output/` → Stores trained models, figures, and predictions.

10.3 Algorithms Applied

- **Logistic Regression** → Baseline linear model for interpretability.
- **Decision Tree** → Captures non-linear patterns, interpretable splits.
- **Random Forest** → Handles feature interactions, reduces variance.
- **XGBoost** → Final best model, strong handling of class imbalance and non-linearities.

11. References

11. References

11.1 References

- XGBoost Documentation: https://xgboost.readthedocs.io/
- Scikit-learn Documentation: https://scikit-learn.org/
- SHAP Explainability: https://shap.readthedocs.io/
- Streamlit: https://streamlit.io/
- HR Analytics Datasets (Kaggle): https://www.kaggle.com/datasets/