Human Resource Management: Predicting Employee Promotions Using Machine Learning

Final Project Report

1. Introduction

1.1. Project overviews

In the competitive landscape of modern business, efficient human resource management is crucial for maintaining a productive and satisfied workforce. This project focuses on leveraging machine learning techniques to predict employee promotions, enabling HR departments to make data-driven decisions and ensure fair and merit-based advancement within the organization.

1.2. Objectives

The primary goal of this project is to develop a predictive model that accurately identifies employees who are likely to be promoted. By analyzing various employee attributes and historical data, the model aims to assist HR in recognizing potential candidates for promotion, improving employee satisfaction, and optimizing talent management.

2. Project Initialization and Planning Phase

2.1. Define Problem Statement

In large organizations, identifying and promoting the right employees at the right time is a complex and critical task. Human Resource (HR) departments often rely on subjective evaluations and historical practices to make promotion decisions, which can lead to biases, inefficiencies, and dissatisfaction among employees. The challenge is to create a data-driven, objective, and efficient method to predict employee promotions, ensuring fairness and meritocracy in career advancement.

2.2. Project Proposal (Proposed Solution)

The proposed solution involves using machine learning algorithms to analyze employee data and predict promotion eligibility based on various factors such as performance metrics, tenure, skills, and feedback. The model will be trained on historical data and validated to ensure accuracy and reliability.

Key Features

The following key features were considered in developing the predictive model for employee promotions:

- **Demographic Information:**

Age: Employee's age, which may influence career stage and promotion readiness.

Education Level: Highest degree attained, impacting qualifications and skill sets.

Employment History:

Department: The specific department or unit the employee belongs to.

Role/Position: Current job title or role, which may affect promotion likelihood.

Tenure: Length of time the employee has been with the organization.

- **Performance Metrics:**

Annual Performance Ratings: Evaluations of employee performance over time.

Achievements and Awards: Recognitions received for outstanding work.

- **Training and Development:**

Training Programs Attended: Number and type of professional development programs completed.

2.3. Initial Project Planning

The initial project planning phase outlines the key steps and milestones necessary for the successful execution of the project, "Human Resource Management: Predicting Employee Promotions Using Machine Learning." This phase is crucial in setting a clear roadmap and ensuring that all necessary preparations are in place. Below are the detailed plans for each step of the project:

Data Collection and Preprocessing

- Understanding & Loading Data: Gain a comprehensive understanding of the dataset structure and contents. Load the data into the working environment for analysis.
- Exploratory Data Analysis (EDA): Perform EDA to uncover patterns, trends, and relationships within the data. Visualize data distributions and correlations.
- Handling Null Values: Identify and address missing values in the dataset using appropriate imputation techniques or removing records if necessary.
- Handling Outliers: Detect and manage outliers that may skew the model by applying methods such as z-score, IQR, or transformation techniques.
- Handling Categorical Values: Encode categorical variables into numerical format using techniques like one-hot encoding or label encoding.

Model Building

- Training the Model: Train multiple machine learning models, including Decision Tree, Random Forest, XGBoost, and KNN, using the preprocessed data.
- Comparing Models: Compare the trained models based on performance metrics such as accuracy, precision, recall, and F1 score to identify the most effective model.
- Evaluating and Saving the Model: Evaluate the best-performing model using the test dataset and save the model for future use.
- Model Optimization: Fine-tune hyperparameters using techniques like Grid Search and Randomized Search to optimize model performance.

Web Integration and Deployment

- Building HTML Pages: Develop the front-end interface of the web application, including pages for home, about, prediction input, and results.
- Local Deployment: Deploy the web application locally to test its functionality and ensure smooth integration with the predictive model.

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

Data Collection Plan :

- Extract data from internal HR databases containing employee details, performance metrics, and promotion records.
- Prioritize datasets with comprehensive demographic information, including department, education level, and length of service.

Data Sources :

- Source Name: Kaggle Dataset
- Description: The dataset comprises various employee attributes such as department, education, training history, performance ratings, and promotion status.
- Location/URL: Kaggle HR Analytics Dataset
- Format: CSV
- Size:- 3.7 MB
- Access Permissions: Public

3.2. Data Quality Report

- Missing values in the ‘education’ and ‘previous year rating’ columns
- Categorical data in the dataset.
- Negative Data in the Dataset
- Imbalanced Data

3.3. Data Exploration and Preprocessing

Data Overview:

Dimensions: 54808 rows × 14 columns

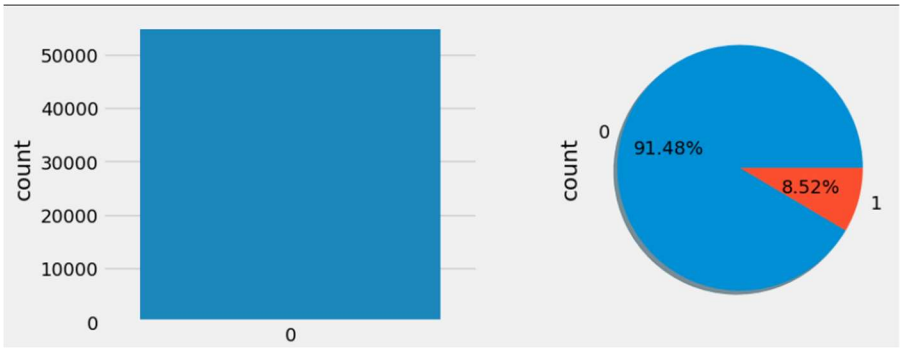
Descriptive statistics:

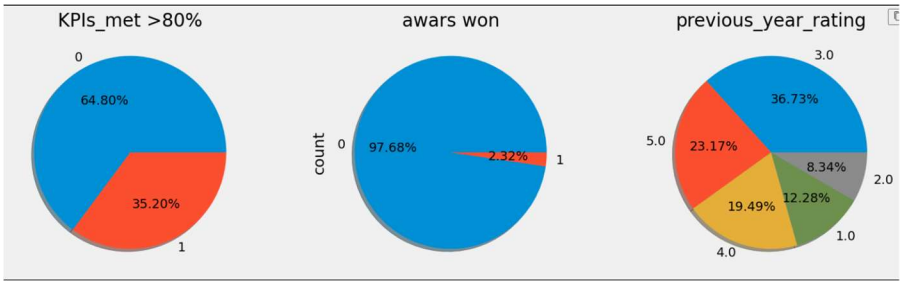
```
#Descriptive Analysis'
df.describe(include='all')
```

✓ 0.0s Python

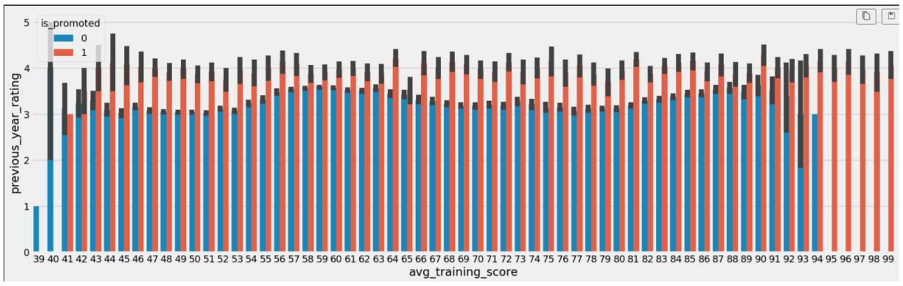
	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score
count	54808.000000	54808	54808	52399	54808	54808	54808.000000	54808.000000	50684.000000	54808.000000	54808.000000	54808.000000	54808.000000
unique	NaN	9	34	3	2	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	Sales & Marketing	region_2	Bachelor's	m	other	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	16840	12343	36669	38496	30446	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	39195.830627	NaN	NaN	NaN	NaN	NaN	1.253011	34.803915	3.329256	5.865512	0.351974	0.023172	63.386750
std	22586.581449	NaN	NaN	NaN	NaN	NaN	0.609264	7.660169	1.259993	4.265094	0.477590	0.150450	13.371559
min	1.000000	NaN	NaN	NaN	NaN	NaN	1.000000	20.000000	1.000000	1.000000	0.000000	0.000000	39.000000
25%	19669.750000	NaN	NaN	NaN	NaN	NaN	1.000000	29.000000	3.000000	3.000000	0.000000	0.000000	51.000000
50%	39225.500000	NaN	NaN	NaN	NaN	NaN	1.000000	33.000000	3.000000	5.000000	0.000000	0.000000	60.000000
75%	58730.500000	NaN	NaN	NaN	NaN	NaN	1.000000	39.000000	4.000000	7.000000	1.000000	0.000000	76.000000
max	78298.000000	NaN	NaN	NaN	NaN	NaN	10.000000	60.000000	5.000000	37.000000	1.000000	1.000000	99.000000

Univariate Analysis :

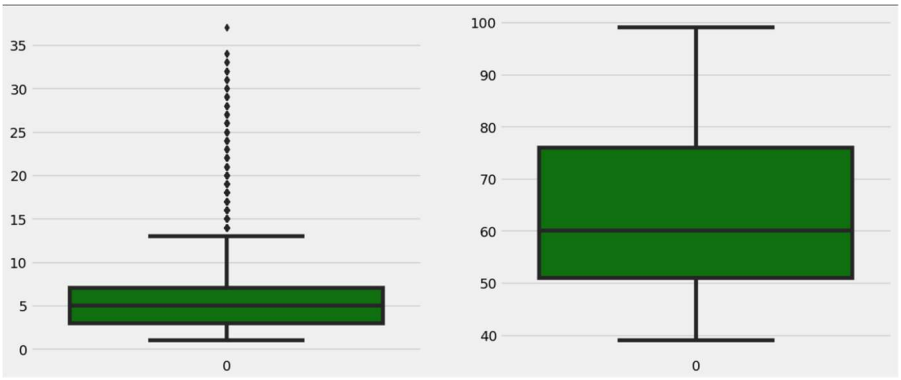




Multivariate Analysis:



Outliers and Anomalies:



Loading Data :

```
#Read data
df=pd.read_csv(r"\Human Resource Management Predicting Employee Promotions Using Machine Learning\5.Project Executable Files\Dataset\emp_promotion.csv")
```
0.1s Python

```
print('Shape of the data = {}'.format(df.shape))
```
0.0s Python

```
Shape of the data = (54808, 14)
```

```
df.head(5)
```
0.0s Python

	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	is_promoted
0	65438	Sales & Marketing	region_7	Master's & above	f	sourcing	1	35	5.0	8	1	0	49	0
1	65141	Operations	region_22	Bachelor's	m	other	1	30	5.0	4	0	0	60	0
2	7513	Sales & Marketing	region_19	Bachelor's	m	sourcing	1	34	3.0	7	0	0	50	0
3	2542	Sales & Marketing	region_23	Bachelor's	m	other	2	39	1.0	10	0	0	50	0
4	48945	Technology	region_26	Bachelor's	m	other	1	45	3.0	2	0	0	73	0

Handling Missing Data:

```
#Replacing Nan with Mode
print(df['education'].value_counts())
df['education']=df['education'].fillna(df['education'].mode()[0])
```

```
print(df['previous_year_rating'].value_counts())
df['previous_year_rating']=df['previous_year_rating'].fillna(df['previous_year_rating'].mode()[0])
```

Data Transformation :

```
#Handling Categorical Values
df['education']=df['education'].replace(("Below Secondary","Bachelor's","Master's & above"),(1,2,3))
lb=LabelEncoder()
df['department']=lb.fit_transform(df['department'])
```

4. Model Development Phase

4.1. Feature Selection Report:

Feature	Description	Selected (Yes/No)	Reasoning
Employee_id	Unique identifier for each employee.	No	Unique identifier, not useful for analysis.
Department	The division or section where the employee works.	Yes	Useful for understanding departmental differences.
Region	The geographical area where the employee is located.	No	Not relevant to predicting promotion of employee in this case.

Education	The highest level of education attained by the employee.	Yes	Correlates with performance and career progression.
Gender	Gender of employee.	No	Not relevant to predicting promotion of employee in this case.
Recruitment_channel	The source through which the employee was hired.	No	Not relevant to predicting promotion of employee in this case.
No_of_trainings	The number of training programs attended by the employee.	Yes	Affects performance and development.
Age	Age of employee.	Yes	Provides insights into experience and career stages.
Previosu_year_rating	The performance rating of the employee from the previous year.	Yes	Strong predictor of future performance.
Length_of_service	The number of years the employee has worked at the company.	Yes	Indicates tenure and potential turnover.
Awards_won	Whether the employee has won any awards.	Yes	Reflects achievements and performance.

4.2. Model Selection Report

Now our data is cleaned and it's time to build the model. We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance. To evaluate the performance confusion matrix and classification report is used

Model	Description	Hyperparameters	Performance Metric (e.g., Accuracy, F1 Score)
DecisionTreeClassifier	Splits data into branches to form decision nodes, aiming for homogeneous subsets.	Random_state=42	Accuracy :- 93.913% F1 score :- 94%
RandomForestClassifier	Constructs multiple decision trees and combines their outputs to improve accuracy and reduce overfitting.	Random_state=42 N_estimators=100	Accuracy :- 95.595% F1 score :- 96%
KNeighborsClassifier	Classifies data points based on the majority class among its k-nearest neighbors.	N_neighbors=5	Accuracy :- 90.573% F1 score :- 91%
GradientBoostingClassifier	Builds an ensemble of weak decision trees, improving accuracy by focusing on misclassified samples.	Random_state=42	Accuracy :- 85.856% F1 score :- 87%

4.3. Initial Model Training Code, Model Validation and Evaluation Report

Training code :

```
def DecisionTree(x_train,x_test,y_train,y_test):
    dt=DTC(random_state=42)
    dt.fit(x_train,y_train)
    ypred=dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print('Confusion_matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification Report')
    print(classification_report(y_test,ypred))

DecisionTree(x_train,x_test,y_train,y_test)
```

✓ 0.1s

```
def randomforest(x_train,x_test,y_train,y_test):
    rf=RandomForestClassifier(random_state=42,n_estimators=100)
    rf.fit(x_train,y_train)
    ypred=rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion_matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification Report')
    print(classification_report(y_test,ypred))

randomforest(x_train,x_test,y_train,y_test)
```

✓ 4.7s

```
def knn(x_train,x_test,y_train,y_test):
    kn=KNeighborsClassifier(n_neighbors=5)
    kn.fit(x_train,y_train)
    ypred=kn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion_matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification Report')
    print(classification_report(y_test,ypred))

knn(x_train,x_test,y_train,y_test)
```

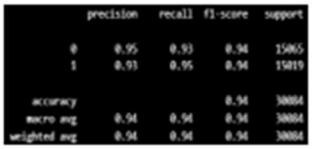
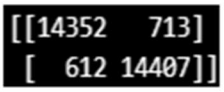
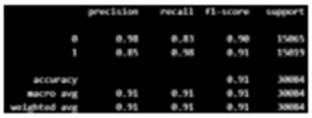
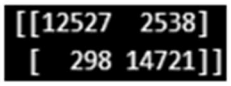
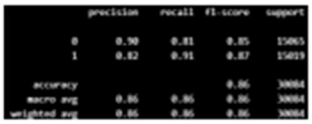
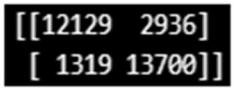
✓ 1.5s

```
def xgboost(x_train,x_test,y_train,y_test):
    xg=GradientBoostingClassifier(random_state=42)
    xg.fit(x_train,y_train)
    ypred=xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion_matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification Report')
    print(classification_report(y_test,ypred))

xgboost(x_train,x_test,y_train,y_test)
```

✓ 4.5s

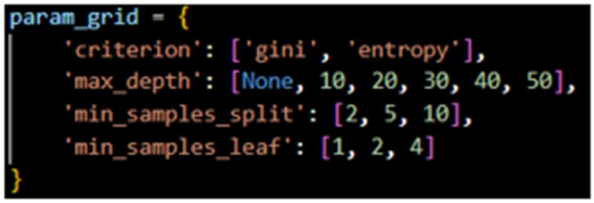
Model Validation and Evaluation Report:

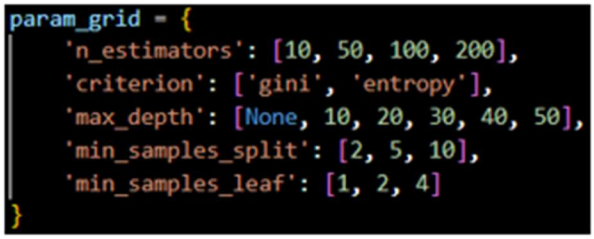
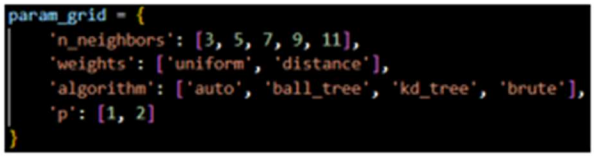
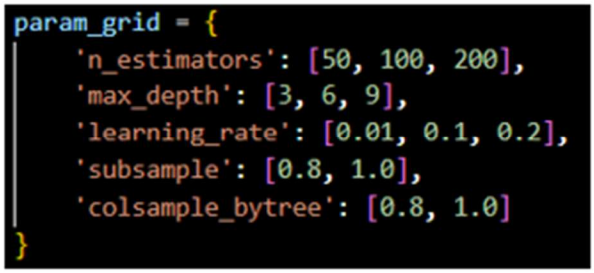
Model	Classification Report	Accuracy	Confusion Matrix
DecisionTree Classifier		93.913%	
RandomForestClassifier		95.595%	
KNeighborsClassifier		90.573%	
GradientBoostingClassifier		85.856%	

5. Model Optimization and Tuning Phase

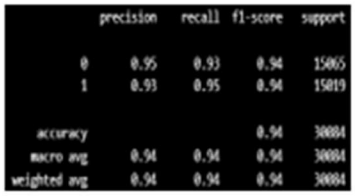
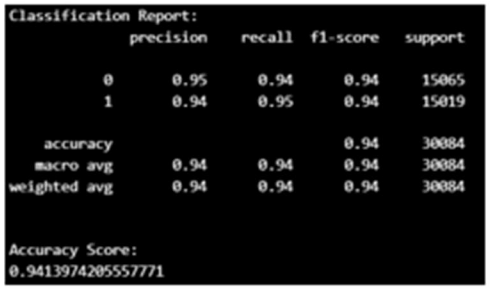
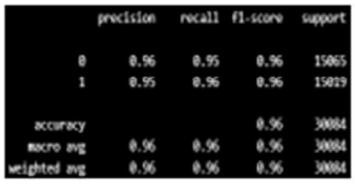
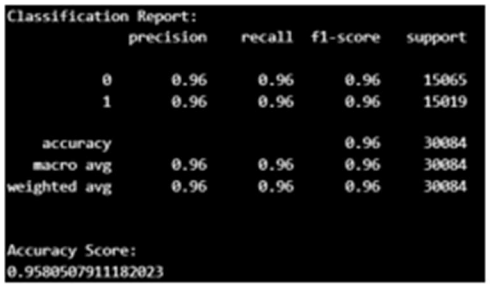
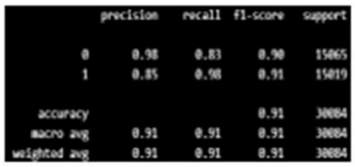
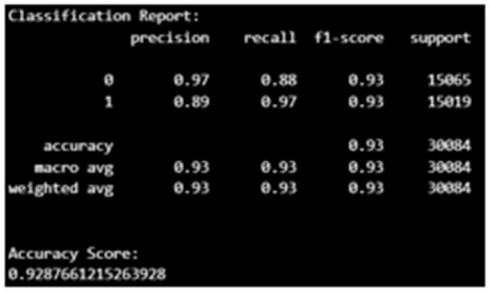
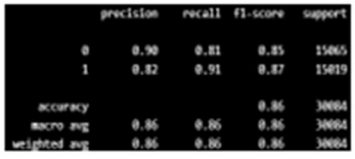
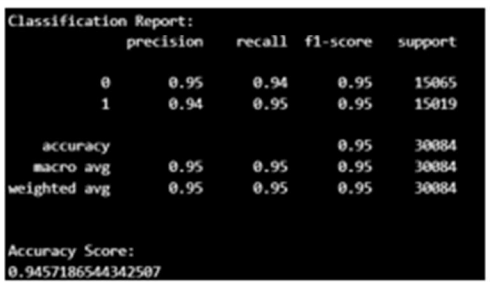
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

5.1. Hyperparameter Tuning Documentation

Model	Tuned Hyperparameters	Optimal Values
DecisionTreeClassifier		{'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5} Accuracy Score: 0.941

RandomForest Classifier		{'criterion': 'entropy', 'max_depth': 40, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200} Accuracy Score: 0.958
KNeighborsClassifier		{'algorithm': 'ball_tree', 'n_neighbors': 3, 'p': 1, 'weights': 'distance'} Accuracy Score: 0.928
GradientBoostingClassifier		{'colsample_bytree': 1.0, 'learning_rate': 0.2, 'max_depth': 9, 'n_estimators': 200, 'subsample': 0.8} Accuracy Score: 0.945

5.2. Performance Metrics Comparison Report

Model	Baseline Metric	Optimized Metric
DecisionTreeClassifier		
RandomForestClassifier		
KNeighborsClassifier		
GradientBoostingClassifier		

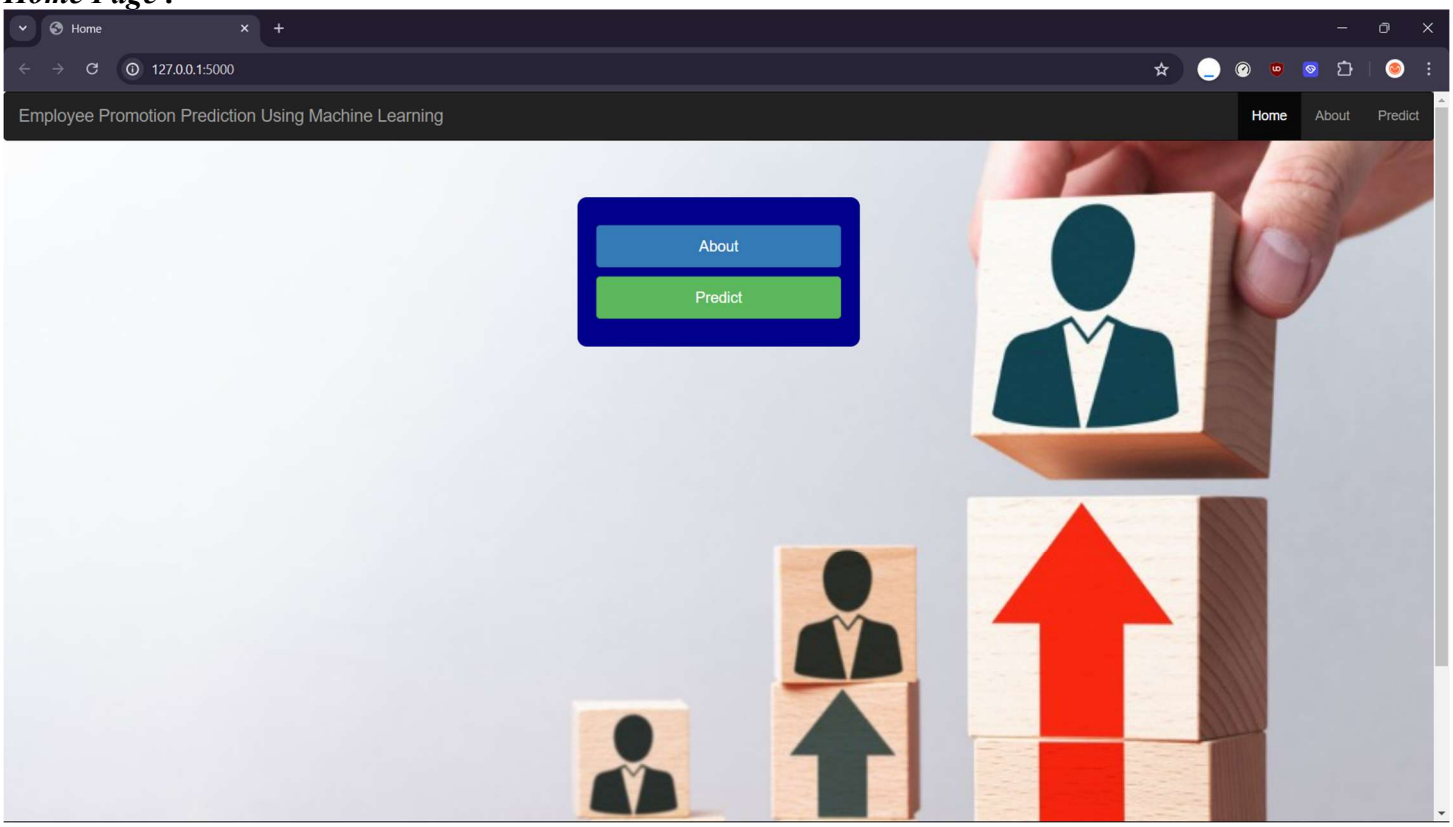
5.3. Final Model Selection Justification

Final Model	Reasoning
RandomForestClassifier	The RandomForestClassifier was chosen as the final optimized model due to its impressive performance, achieving 95.8% accuracy. Its robustness to overfitting, ability to handle large datasets, and high accuracy make it ideal for the task. The ensemble approach, which involves averaging multiple decision trees, enhances stability and generalization to unseen data. Additionally, random forests efficiently manage datasets with numerous features and provide insights into feature importance. This model's balance of accuracy, robustness, and interpretability, along with its versatility, confirms its suitability for the problem at hand.

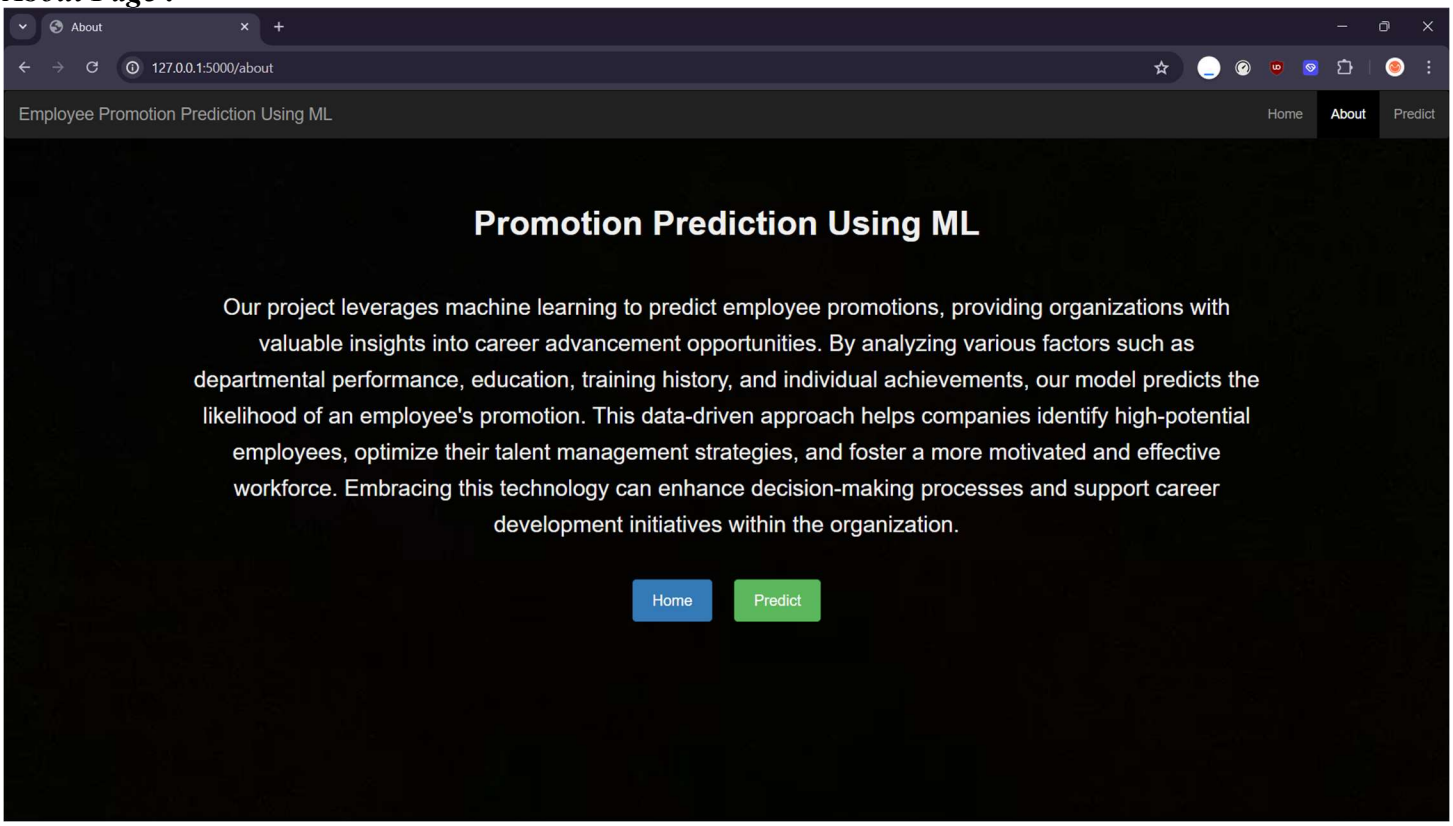
6. Results

6.1. Output Screenshots

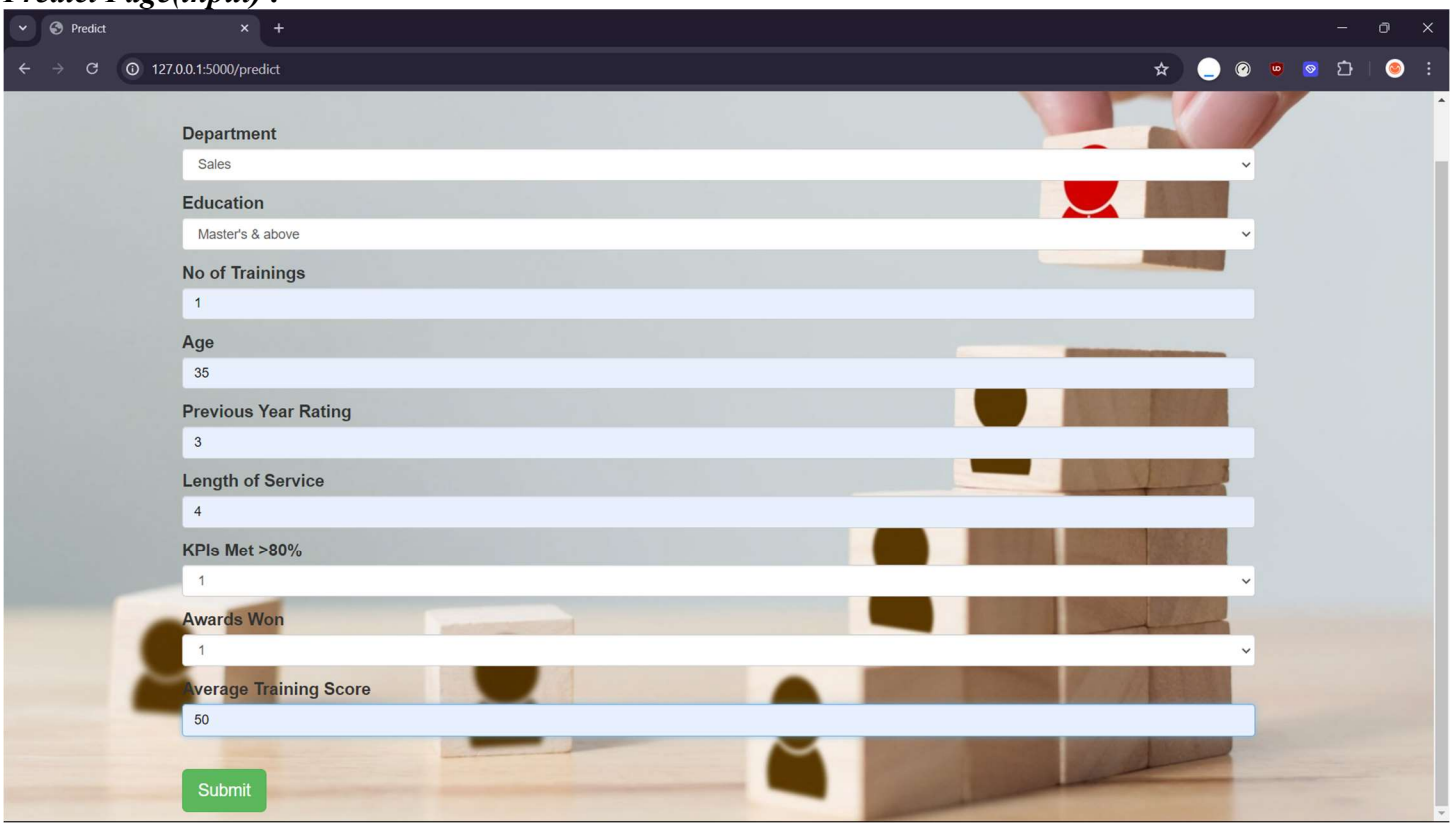
Home Page :



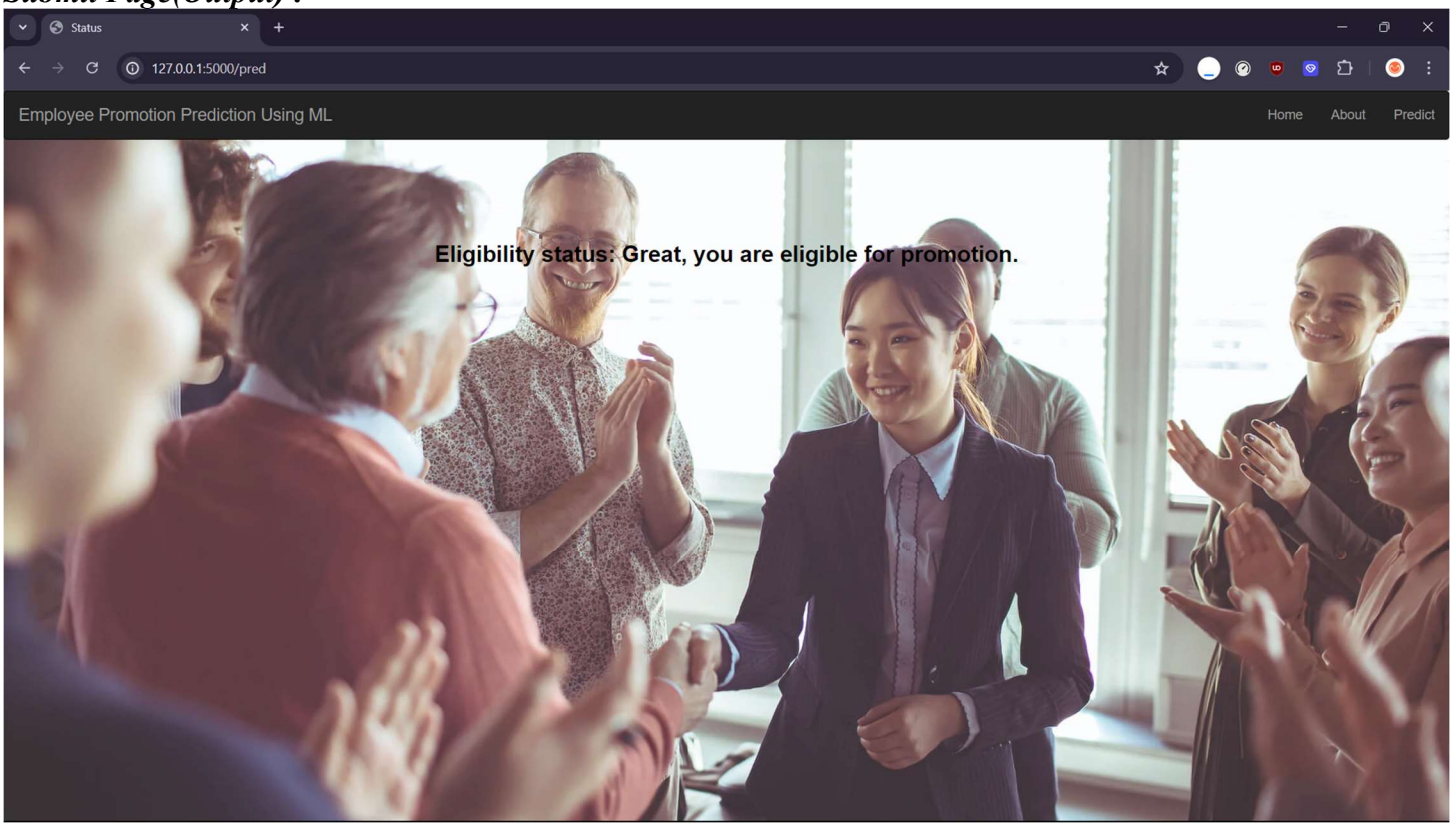
About Page :



Predict Page(input) :



Submit Page(Output) :



7. Advantages & Disadvantages:

Advantages:

- Enhanced Accuracy:
  - Advantage: Machine learning algorithms can process vast amounts of employee data and uncover complex patterns, resulting in highly accurate predictions of promotion likelihood. For example, the RandomForestClassifier achieved 95.8% accuracy in this project.
- Data-Driven Decision Making:
  - Advantage: By relying on historical data and statistical models, machine learning promotes objective and fair decision-making, reducing biases that might affect human judgment in promotion decisions.
- Efficiency:
  - Advantage: The model automates the promotion prediction process, saving time for HR professionals and enabling quicker identification of potential candidates for advancement.
- Scalability:
  - Advantage: Machine learning models can handle large datasets and adapt to growing employee data, making them suitable for organizations of varying sizes.
- Insights into Employee Performance:
  - Advantage: The model provides valuable insights into which factors are most influential in promotion decisions, helping HR departments to understand and manage their talent more effectively.

Disadvantages:

- Dependence on Data Quality:
    - Disadvantage: The accuracy of the model is contingent on the quality and completeness of the employee data. Inaccurate or missing data can lead to unreliable predictions.
- Complexity of Implementation:
    - Disadvantage: Developing, training, and tuning machine learning models can be complex and may require specialized knowledge and resources that might not be readily available.
- Interpretability Issues:
    - Disadvantage: Although RandomForestClassifier offers some feature importance insights, the overall decision-making process of the model can be opaque, making it difficult for HR professionals to fully understand how predictions are made.
- Cost and Resources:
    - Disadvantage: Implementing machine learning solutions involves costs related to data processing, computational resources, and potentially hiring skilled personnel for model development and maintenance
- Risk of Overfitting:
    - Disadvantage: The model might become too tailored to the training data, potentially leading to poor performance on new, unseen data. This requires careful monitoring and regular updates to ensure continued accuracy.
- Ethical and Privacy Concerns:
    - Disadvantage: Handling personal and sensitive employee data raises ethical and privacy issues. It is essential to ensure compliance with data protection regulations and address any concerns about data misuse.

8. Conclusion

This project successfully developed a machine learning model to predict employee promotions, offering a data-driven solution to streamline HR processes. By accurately forecasting promotion eligibility, the model enhances fairness and transparency in the promotion process, improving employee satisfaction and retention.

9. Future Scope

• Expanded Features:

Integrate additional data sources like employee engagement and real-time updates for a more comprehensive prediction model.

• Model Enhancement:

Explore advanced algorithms and refine hyperparameters to improve accuracy and capture complex patterns.

• Increased Interpretability:

Use techniques like SHAP or LIME to make model predictions more understandable and transparent.

• HR Integration:

Embed the model into HR systems for actionable insights and automate alerts for potential

promotion candidates.

- Ethical Considerations:

Monitor and mitigate biases, and ensure strong data privacy measures to maintain fairness and compliance.

- Employee Development:

Create personalized development plans based on model insights and establish feedback mechanisms to refine predictions.

- Broader Applications:

Apply the model to other HR functions like succession planning and compare promotion practices across departments.

10. Appendix

10.1. Source Code

10.2. GitHub & Project Demo Link