Final Project Report Template

Human resource management employee promotion prediction

1. Introduction

1.1. Project overview

Human resource management (HRM) employee promotion prediction involves using data analytics and machine learning techniques to forecast which employees are likely to be promoted within an organization. This process includes collecting and analyzing various types of employee data, such as demographics, job-related information, performance metrics, and behavioral patterns. The goal is to identify key factors that contribute to promotions and use these insights to make informed, objective decisions about employee career advancement. By leveraging predictive models, organizations can enhance their talent management strategies, reduce biases in promotion decisions, improve employee satisfaction and retention, and ensure that high-potential employees are recognized and developed effectively. This approach not only supports fair and transparent promotion practices but also aligns with broader organizational goals of optimizing workforce planning and productivity.

1.2 Objectives

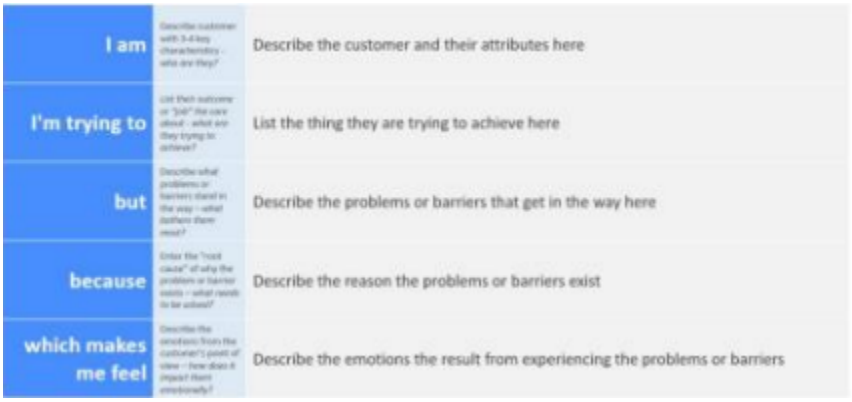
The primary objectives of human resource management (HRM) employee promotion prediction include enhancing decision-making processes, improving employee satisfaction and retention, and fostering a fair and transparent work environment. By accurately predicting which employees are likely to be promoted, organizations can ensure that their promotion processes are based on data-driven insights, reducing biases and subjective judgments. This not only helps in identifying high-potential employees but also aligns talent management strategies with organizational goals. Moreover, effective promotion prediction can lead to better workforce planning, ensuring that key positions are filled by qualified and prepared individuals, ultimately

contributing to the overall efficiency and productivity of the organization.

2.Project initialization and Planning Phase

2.1 Define Problem Statement

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for your customers' challenges.Throughout the process,you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makesme feel
HR facing challenges in identifying top performers	HR	Select best performers	efficiently analyze employee data	Taking more time	Facing challanges

It became difficult to select for a seeks to proactively identify and nurture high-performing employees to prevent attrition.	competitive industry management	Select best employees	It became tough to select the best one	Select from a flock of people	Vexation

2.2 Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective,

defined scope, and a concise problemstatement, the proposedsolution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	Develop a modelto predict employee promotions using historical data for enhanced HR decision-making.
Scope	Collect, preprocess data,engineer features, trainmodels, deploy for real-time predictions with ethical considerations.
Problem Statement	
Description	Utilize historical data to forecast employee promotions for enhanced organizational decision support.
Impact	Improve fairness and efficiency in promotion decisions, leading to a motivated and stable workforce.

Proposed Solution	
Approach	Employ machine learning algorithms on historical data to predict employee promotion likelihood accurately.
Key Features	Data preprocessing, featureengineering, model selection, validation, deployment integration for real-time promotion predictions

Resource Requirements

Resource Type	**Description**	**Specification/Allocation**

Hardware		
Computing Resources	CPU/GPU specifications, number of cores	e.g., 2 x NVIDIAV100 GPUs
Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models, and logs	e.g., 1 TB SSD
Software		
Frameworks	Python frameworks	e.g., Flask,Seaburn, Keras
Libraries	Additional libraries	e.g., scikit-learn, pandas, numpy
Development Environment	IDE, version control	e.g., Jupyter Notebook, Git
Data		
Data	Source, size,format	e.g., Kaggle dataset, 10,000 images

Project Initialization and Planning Phase

2.3 Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problemstatement, the proposedsolution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	Develop a modelto predict employee promotions using historicaldata for enhanced HR decision-making.
Scope	Collect, preprocess data,engineer features, trainmodels, deploy forreal-time predictions with ethical considerations.
Problem Statement	
Description	Utilize historical datato forecast employee promotions for enhancedorganizational decision support.
Impact	Improve fairness and efficiency in promotion decisions, leading to amotivated and stable workforce.
Proposed Solution	
Approach	Employ machine learning algorithms on historical data to predictemployee promotion likelihood accurately.
Key Features	Data preprocessing, feature engineering, modelselection, validation,deployment integration for real-time promotion

Resource Requirements

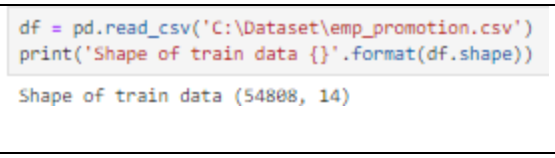
Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications,number of	e.g., 2 x NVIDIA V100 GPUs

Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models,and logs	e.g., 1 TBSSD
Software		
Frameworks	Python frameworks	e.g., Flask,Seaburn, Keras
Libraries	Additional libraries	e.g., scikit-learn, pandas,numpy
Development Environment	IDE, version control	e.g., JupyterNotebook, Git
Data		
Data	Source, size,format	e.g., Kaggle dataset, 10,000images

3.Data Collection and Preprocessing Phase

3.1 Data Exploration and Preprocessing Template

Identifies data sources, assessesquality issues like missing valuesand duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	Basic statistics, dimensions, and structure of the data.
Univariate Analysis	Exploration of individual variables (mean, median, mode, etc.).
Bivariate Analysis	Relationships between two variables (correlation, scatter plots).
Multivariate Analysis	Patterns and relationships involving multiple variables.
Outliers and Anomalies	Identification andtreatment of outliers.
Data Preprocessing Code Screenshots	
Loading Data	

```
df = pd.read_csv('C:\Dataset\emp_promotion.csv')
print('Shape of train data {}'.format(df.shape))
```

```
Shape of train data (54808, 14)
```

Handling Missing Data

```
df.isnull().sum()
```

```
department               0
education             2409
no_of_trainings          0
age                      0
previous_year_rating    4124
length_of_service        0
KPIs_met >80%            0
awards_won?              0
avg_training_score       0
is_promoted              0
dtype: int64
```

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

```
education
Bachelor's          36669
Master's & above    14925
Below Secondary       805
Name: count, dtype: int64
```

```
#Replacing nan with mode

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

```
previous_year_rating
3.0    18618
5.0    11741
4.0     9877
1.0     6223
2.0     4225
Name: count, dtype: int64
```

Data Transformation

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Select numerical columns
numerical_columns = ['no_of_trainings', 'age', 'previous_year_rating',
                     'length_of_service', 'KPIs_met >80%', 'awards_won?',
                     'avg_training_score']

# Standard scaling
scaler = StandardScaler()
data_standard_scaled = pd.read_csv('C:\Dataset\emp_promotion.csv')
data_standard_scaled[numerical_columns] = scaler.fit_transform(data_standard_scaled[numerical_columns])

# Min-Max normalization
min_max_scaler = MinMaxScaler()
data_min_max_scaled = pd.read_csv('C:\Dataset\emp_promotion.csv')
data_min_max_scaled[numerical_columns] = min_max_scaler.fit_transform(data_min_max_scaled[numerical_columns])

# Display the first few rows of the scaled and normalized data
print("Standard Scaled Data:\n", data_standard_scaled.head())
print("\nMin-Max Normalized Data:\n", data_min_max_scaled.head())
```

```
Standard Scaled Data:
    employee_id         department     region         education gender  \
0       65438  Sales & Marketing   region_7  Master's & above      f
1       65141         Operations  region_22        Bachelor's      m
2        7513  Sales & Marketing  region_19        Bachelor's      m
3        2542  Sales & Marketing  region_23        Bachelor's      m
4       48945         Technology  region_26        Bachelor's      m

  recruitment_channel  no_of_trainings       age  previous_year_rating  \
0           sourcing        -0.415276  0.025598              1.326008
1              other        -0.415276 -0.627135              1.326008
2           sourcing        -0.415276 -0.184948             -0.261318
3              other         1.226063  0.547785             -1.848645
4              other        -0.415276  1.331064             -0.261318

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0           0.500460       1.356878    -0.154018           -1.075931
1          -0.437395      -0.736986    -0.154018           -0.253282
2           0.265996      -0.736986    -0.154018           -1.001145
3           0.969387      -0.736986    -0.154018           -1.001145
4          -0.906322      -0.736986    -0.154018            0.718939

   is_promoted
0            0
1            0
2            0
3            0
4            0
```

```
Min-Max Normalized Data:
    employee_id         department     region         education gender  \
0         65438  Sales & Marketing   region_7  Master's & above      f
1         65141         Operations  region_22        Bachelor's      m
2          7513  Sales & Marketing  region_19        Bachelor's      m
3          2542  Sales & Marketing  region_23        Bachelor's      m
4         48945         Technology  region_26        Bachelor's      m

  recruitment_channel  no_of_trainings    age  previous_year_rating  \
0           sourcing         0.000000  0.375                   1.0
1              other         0.000000  0.250                   1.0
2           sourcing         0.000000  0.350                   0.5
3              other         0.111111  0.475                   0.0
4              other         0.000000  0.625                   0.5

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0           0.194444             1.0          0.0            0.166667
1           0.083333             0.0          0.0            0.350000
2           0.166667             0.0          0.0            0.183333
3           0.250000             0.0          0.0            0.183333
4           0.027778             0.0          0.0            0.566667

   is_promoted
0            0
1            0
2            0
3            0
4            0
```

Feature Engineering

```python
import pandas as pd

# Load the CSV file

data = pd.read_csv('C:\Dataset\emp_promotion.csv')

# Create a new feature 'is_high_performer'
data['is_high_performer'] = data.apply(lambda row: 1 if row['KPIs_met >80%'] == 1 and row['awards_won?'] == 1 else 0, axis=1)

# Create an 'age_group' feature
bins = [20, 30, 40, 50, 60]
labels = ['20-29', '30-39', '40-49', '50-59']
data['age_group'] = pd.cut(data['age'], bins=bins, labels=labels, right=False)

# Modify the 'education' feature to fewer categories
data['education'] = data['education'].replace({
    "Master's & above": "Postgraduate",
    "Bachelor's": "Undergraduate",
    "Below Secondary": "Secondary"
})

# Display the first few rows to verify changes
print(data.head())
```

```
   employee_id         department     region      education gender  \
0        65438  Sales & Marketing   region_7  Postgraduate      f
1        65141         Operations  region_22  Undergraduate      m
2         7513  Sales & Marketing  region_19  Undergraduate      m
3         2542  Sales & Marketing  region_23  Undergraduate      m
4        48945         Technology  region_26  Undergraduate      m

  recruitment_channel  no_of_trainings  age  previous_year_rating  \
0           sourcing                1   35                   5.0
1              other                1   30                   5.0
2           sourcing                1   34                   3.0
3              other                2   39                   1.0
4              other                1   45                   3.0

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0                  8              1            0                  49
1                  4              0            0                  60
2                  7              0            0                  50
3                 10              0            0                  50
4                  2              0            0                  73

   is_promoted  is_high_performer age_group
0            0                  0     30-39
1            0                  0     30-39
2            0                  0     30-39
3            0                  0     30-39
4            0                  0     40-49
```

Save Processed Data	

3.2 Data QualityReport Template

The Data Quality ReportTemplate will summarizedata quality issues from the selected source,including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

Data Source	Data Quality Issue	Severity	Resolution Plan
Employee Records	There are some nullvalues	Moderate	By using is.nawe have resolvedthe solution
Employee Records	Missing valuesin key fields	High	Implement data imputation methods such as mean/mode imputation for numerical and categorical data, or use advancedtechniques likeKNN imputation.
Histoíical PíomotionData	Outlieís and anomalies in píomotion íecoíds (e.g., unusually íapid	High	Utilizing statistical techniques or subject-matterexpertise, locate and look into anomalies. Based

	píomotions)		validation with HR specialists and consistency checks, eliminate or fix anomalies. Use a strongoutlier detection algoíithms
Employee Suíveys	Duplicate suívey íesponses	Moderate	Make use of datadeduplication strategies, such as timestamp analysis anduniqueemployee identification, to find and eliminate duplicaterows.Make sure every worker only submits one survey response.
Skills and Ceítifications	Outdated oí unveíified skills and ceítification data	Moderate	Update your qualification and skill recordson a regular basis by using self-reporting tools and verifying your information with professional certifying organizations. To guarantee data accuracy, implement routine audits and

3.3 Data Collection Plan & Raw Data SourcesIdentification Template

Elevate your data strategy with the Data Collection plan and the Raw Data Sources report, ensuring meticulous data curation and integrity for informed decision-making in every analysis and decision-making endeavor.

Data Collection Plan Template

Section	Description

Project Overview	Brief overview of the machinelearning project and its objectives.
Data Collection Plan	Mention fromwhich sources the dataare going to be collected.
Raw Data Sources Identified	List the raw data sourceswith relevant details(as a short description).

Raw Data Sources Template

Source Name	**Description**	**Location/URL**	**Format**	**Size**	**Access Permissions**

Dataset 1	This dataset contains information about employee promotions. Each row represents an individual employee with various attributes that may influence their promotion status. Below is a description of each column in the dataset: 1. **employee_id**: A unique identifier for each employee. 2. **department**: The department where the employee works (e.g., Sales & Marketing, Operations, Technology). 3. **region**: The geographical region where the employee is	https://drive.google.com/file/d/1I4qAYPpk3pctlYScWqw0Du2JEYF-rY80/view?usp=drivesdk	CSV	3MB	Public

	the employee (e.g., Master's & above, Bachelor's). 1. **gender**: The gender of the employee (e.g., f for female, m for male). 2. **recruitment_channel**: The channel through which the employee was recruited (e.g., sourcing, other). 3. **no_of_trainings**: The number of training programs the employee has completed. 4. **age**: The age of the employee. 5. **previous_ye**				

	1. **length_of_service**: The number of years the employee has been with the company. 2. **KPIs_met >80%**: Indicates whether the employee met more than 80% of their Key Performance Indicators (KPIs) (1 for yes, 0 for no). 3. **awards_won?**: Indicates whether the employee has won any awards (1 for yes, 0 for no). 4. **avg_training_score**: The average score of the employee in training programs				

	This dataset can be used to analyze the factors influencing employee promotions and build predictive models to identify potential candidates for promotion based on their attributes.				

4. Model Development Phase Template

4.1 Feature Selection Report Template

In the forthcoming update, each featurewill be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected (Yes/No)	Reasoning

Years of Experience	Number of years the employee has worked in the company.	Yes	Experience and skill level are frequently correlated withone other. Performance ratings show potential for growth as well as present performance levels
Last Performanc e Rating	The employee's performance rating in the last appraisal	Yes	Performance ratings reflect current performance levels and opportunities for advancement.

Previous Promotions	Number of times the employee has been promoted previously	Yes	Previous promotions imply career advancement and possibilities for future promotions.
Awards	Number of awards received by the employee	Yes	Awards reflect recognition of extraordinary performance, which may be associated withpromotion opportunities.
Education	Highest education level attained by the employee.	Yes	Higher education levels could be a sign of aptitude and preparedness for more senior roles.

Training Score	Employee's score in training and development programs.	No	When weighed against other variables like experience and performance, training ratings may not have a direct correlation with promotability.

4.2 Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the futurethrough a screenshot. The modelvalidation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model TrainingCode:

Decision Tree:-

```
def decisionTree(x_train,x_test,y_train,y_test):
 dt=DecisionTreeClassifier()
 dt.fit(x_train,y_train)
 y_pred=dt.predict(x_test)
 print("DecisionTreeClassifier")
 print('Confusion matrix')
 print(confusion_matrix(y_test,y_pred))
 print('Classification report')
 print(classification_report(y_test,y_pred))
 return y_pred
```

Random Forest:-

```
def randomForest(x_train,x_test,y_train,y_test):
 rf=RandomForestClassifier()
 rf.fit(x_train,y_train)
 y_pred=rf.predict(x_test)
 print("RandomForestClassifier")
 print('Confusion matrix')
 print(confusion_matrix(y_test,y_pred))
 print('Classification report')
 print(classification_report(y_test,y_pred))
 return y_pred
```

KNN:-

```
from sklearn.neighbors import KNeighborsClassifier
def KNN(x_train,x_test,y_train,y_test):
 knn=KNeighborsClassifier()
 knn.fit(x_train,y_train)
 y_pred=knn.predict(x_test)
 print("KNeighboursClassifier")
 print('Confusion matrix')
 print(confusion_matrix(y_test,y_pred))
 print('Classification report')
 print(classification_report(y_test,y_pred))
 return y_pred
```

Xgboost:-

```
def xgboost(x_train,x_test,y_train,y_test):
 xg=GradientBoostingClassifier()
 xg.fit(x_train,y_train)
 y_pred=xg.predict(x_test)
 print("GradientBoostingClassifier")
 print('Confusion matrix')
 print(confusion_matrix(y_test,y_pred))
 print('Classification report')
 print(classification_report(y_test,y_pred))
 return y_pred
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decisi onTree	`classification_report(y_test,y_pred)` Classification report precision recall f1-score support 0 0.95 0.93 0.94 10035 1 0.93 0.95 0.94 10021 accuracy 0.94 20056 macro avg 0.94 0.94 0.94 20056 weighted avg 0.94 0.94 0.94 20056	93.73%	`confusion_matrix(y_test,y_pred)` array([[9289, 746], [510, 9511]], dtype=int64)

rand om Forest	`classification_report(y_test,y_pred)` Classification report precision recall f1-score support 0 0.95 0.95 0.95 10035 1 0.95 0.95 0.95 10021 accuracy 0.95 20056 macro avg 0.95 0.95 0.95 20056 weighted avg 0.95 0.95 0.95 20056	94.94%	`confusion_matrix(y_test,y_pred)` array([[9498, 537], [477, 9544]], dtype=int64)
KNN	`classification_report(y_test,y_pred)` Classification report precision recall f1-score support 0 0.96 0.82 0.89 10035 1 0.84 0.97 0.90 10021 accuracy 0.90 20056 macro avg 0.90 0.90 0.89 20056 weighted avg 0.90 0.90 0.89 20056	89.5%	`confusion_matrix(y_test,y_pred)` array([[8242, 1793], [308, 9713]], dtype=int64)

xgboost	`classification_report(y_test,y_pred)` Classification report precision recall f1-score support 0 0.88 0.84 0.86 10035 1 0.85 0.89 0.87 10021 accuracy 0.86 20056 macro avg 0.87 0.86 0.86 20056 weighted avg 0.87 0.86 0.86 20056	86.43%	`confusion_matrix(y_test,y_pred)` array([[8409, 1626], [1094, 8927]], dtype=int64)

4.3 Model Selection Report:

Model	Description	Hyperparameters	Performance Metric(e.g., Accuracy, F1 Score)

decision Tree	Designed for both regression and classification applications, a decision tree is a supervised machine learning technique. In order to create a decision tree model, it divides the datainto subsets according to theinput feature values.	criterion='entropy', max_depth=5, min_samples_split=10, min_samples_leaf=5,random_state=42	Accuracy Score=93.73%

randomForest	In order to improve prediction accuracy and robustness through a reduction in overfitting and an increase in model generalization ability, Random Forests, one of the key algorithms of	n_estimators=100, max_depth=5, random_state=42	Accuracy Score=94.94%
KNN	K-Nearest Neighbors (KNN) is a simple yet effective machine learning algorithm that predicts the output of a new instance by	n_neighbors=3, weights='uniform',algorithm='auto',leaf_size=10	Accuracy Score=89.52%
xgboost	XGBoost is a highly efficient and flexible	n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42	Accuracy Score=86.43%

	structured/tabular data. Several data scientists and machine learning practitioners use it exclusively because of its speed, accuracy, and		

5.Model Optimization and Tuning PhaseTemplate

5.1 Model Optimization and Tuning Phase

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

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

DecisionTr
 	
eeClassifier
	`dt = DecisionTreeClassifier(criterion='entropy',max_depth=5,min_samples_split=10,`
	`min_samples_leaf=5,random_state=42)`
	`dt.fit(x_train,y_train)`
	`y_pred_dt=dt.predict(x_test)`
		`accuracy_score(y_test,y_pred_dt)`
		`0.7149980055843638`

RandomForestClassifier	`rf=RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42)`
	`rf.fit(x_train,y_train)`
	`y_pred_rf=rf.predict(x_test)`
		`accuracy_score(y_test,y_pred_rf)`
		`0.7957718388512166`

KneighboursClassifier	`knn=KNeighborsClassifier(n_neighbors=3, weights='uniform',algorithm='auto',leaf_size=10)`
	`knn.fit(x_train,y_train)`
	`y_pred_kn=knn.predict(x_test)`
		`accuracy_score(y_test,y_pred_kn)`
		`0.9032708416433984`

GradientBoosting Classifier	`xg=GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)`
	`xg.fit(x_train,y_train)`
	`y_pred_xg=xg.predict(x_test)`
		`accuracy_score(y_test,y_pred_xg)`
		`0.864379736737136`

Performance MetricsComparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric

DecisionTreeClassifier	`classification_report(y_test,y_pred)`
	precision recall f1-score support
	0 0.95 0.93 0.94 10035
	1 0.93 0.95 0.94 10021
	accuracy 0.94 20056
	macro avg 0.94 0.94 0.94 20056
	weighted avg 0.94 0.94 0.94 20056
	`confusion_matrix(y_test,y_pred)`
	`array([[9289, 746],`
	`[ 510, 9511]], dtype=int64)`
		`y_pred_dt=decisionTree(x_train,x_test,y_train,y_test)`
		DecisionTreeClassifier
		Confusion matrix
		[[6474 3561]
		[2155 7866]]
		Classification report
		precision recall f1-score support
		0 0.75 0.65 0.69 10035
		1 0.69 0.78 0.73 10021
		accuracy 0.71 20056
		macro avg 0.72 0.72 0.71 20056
		weighted avg 0.72 0.71 0.71 20056

RandomForestClassifier	classification_report(y_test,y_pred) precision recall f1-score support 0 0.95 0.95 0.95 10035 1 0.95 0.95 0.95 10021 accuracy 0.95 20056 macro avg 0.95 0.95 0.95 20056 weighted avg 0.95 0.95 0.95 20056 confusion_matrix(y_test,y_pred) array([[9498, 537], [477, 9544]], dtype=int64)	y_pred_rf=randomForest(x_train,x_test,y_train,y_test) RandomForestClassifier Confusion matrix [[7317 2718] [1378 8643]] Classification report precision recall f1-score support 0 0.84 0.73 0.78 10035 1 0.76 0.86 0.81 10021 accuracy 0.80 20056 macro avg 0.80 0.80 0.79 20056 weighted avg 0.80 0.80 0.79 20056
Kneighbou rsClassifier	classification_report(y_test,y_pred) precision recall f1-score support 0 0.96 0.82 0.89 10035 1 0.84 0.97 0.90 10021 accuracy 0.90 20056 macro avg 0.90 0.90 0.89 20056 weighted avg 0.90 0.90 0.89 20056 confusion_matrix(y_test,y_pred) array([[8242, 1793], [308, 9713]], dtype=int64)	y_pred_kn=KNN(x_train,x_test,y_train,y_test) Confusion matrix [[8451 1584] [356 9665]] Classification report precision recall f1-score support 0 0.96 0.84 0.90 10035 1 0.86 0.96 0.91 10021 accuracy 0.90 20056 macro avg 0.91 0.90 0.90 20056 weighted avg 0.91 0.90 0.90 20056
GradientBoosting Classifier	classification_report(y_test,y_pred) precision recall f1-score support 0 0.88 0.84 0.86 10035 1 0.85 0.89 0.87 10021 accuracy 0.86 20056 macro avg 0.87 0.86 0.86 20056 weighted avg 0.87 0.86 0.86 20056 confusion_matrix(y_test,y_pred) array([[8409, 1626], [1094, 8927]], dtype=int64)	y_pred_xg=xgboost(x_train,x_test,y_train,y_test) Confusion matrix [[8409 1626] [1094 8927]] Classification report precision recall f1-score support 0 0.88 0.84 0.86 10035 1 0.85 0.89 0.87 10021 accuracy 0.86 20056 macro avg 0.87 0.86 0.86 20056 weighted avg 0.87 0.86 0.86 20056

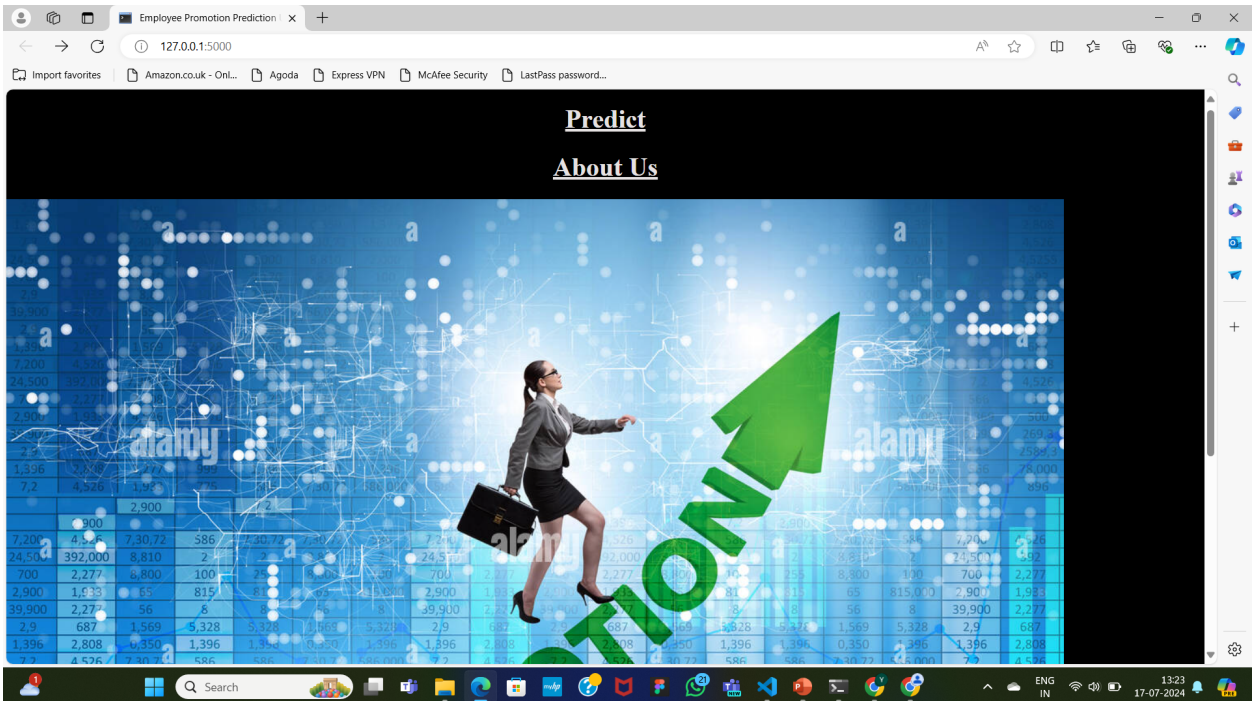
Final ModelSelection Justification (2 Marks):

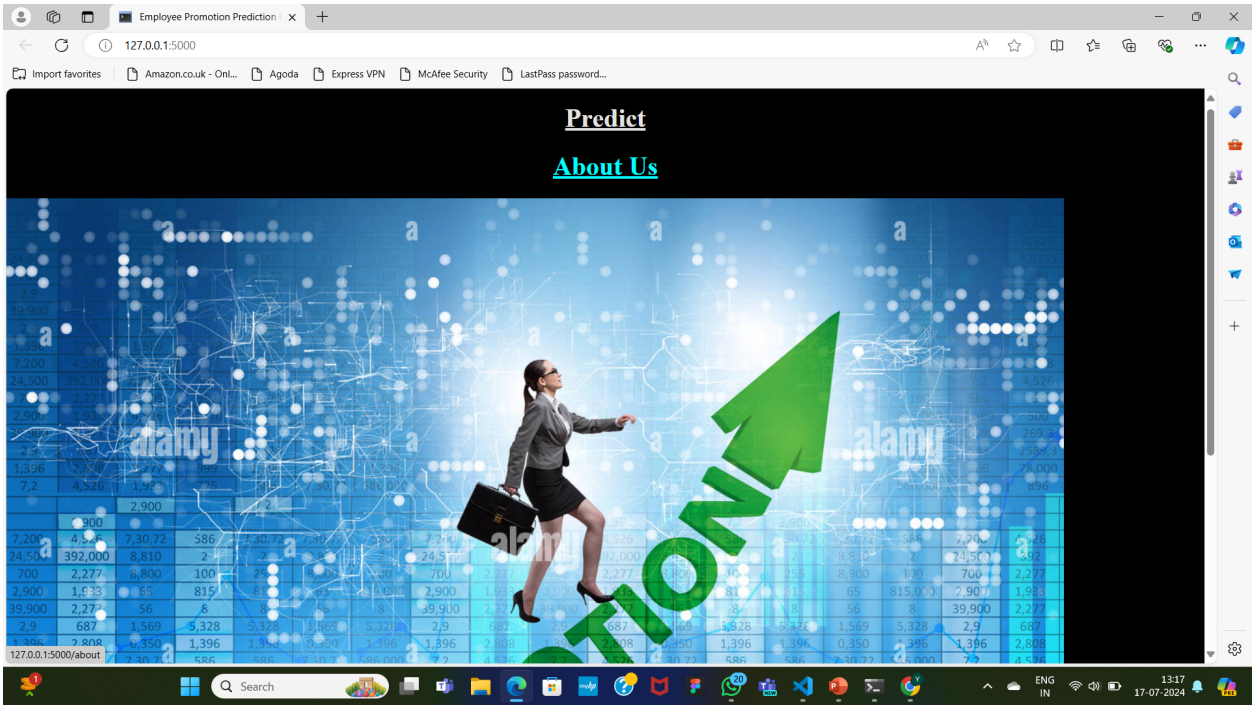
Final Model	Reasoning
RandomForest Classifier	The RandomForest Classifier was chosen because it performs well acrossa range of datasets and doesn't requirea lot of hyperparameter tweaking. By combining several decision trees— each trained on a different collection of attributes and observations itsuccessfully minimizes overfitting. This ensemble technique effectively manages noisy dataand outliers whileenhancing generalization.

6. Results

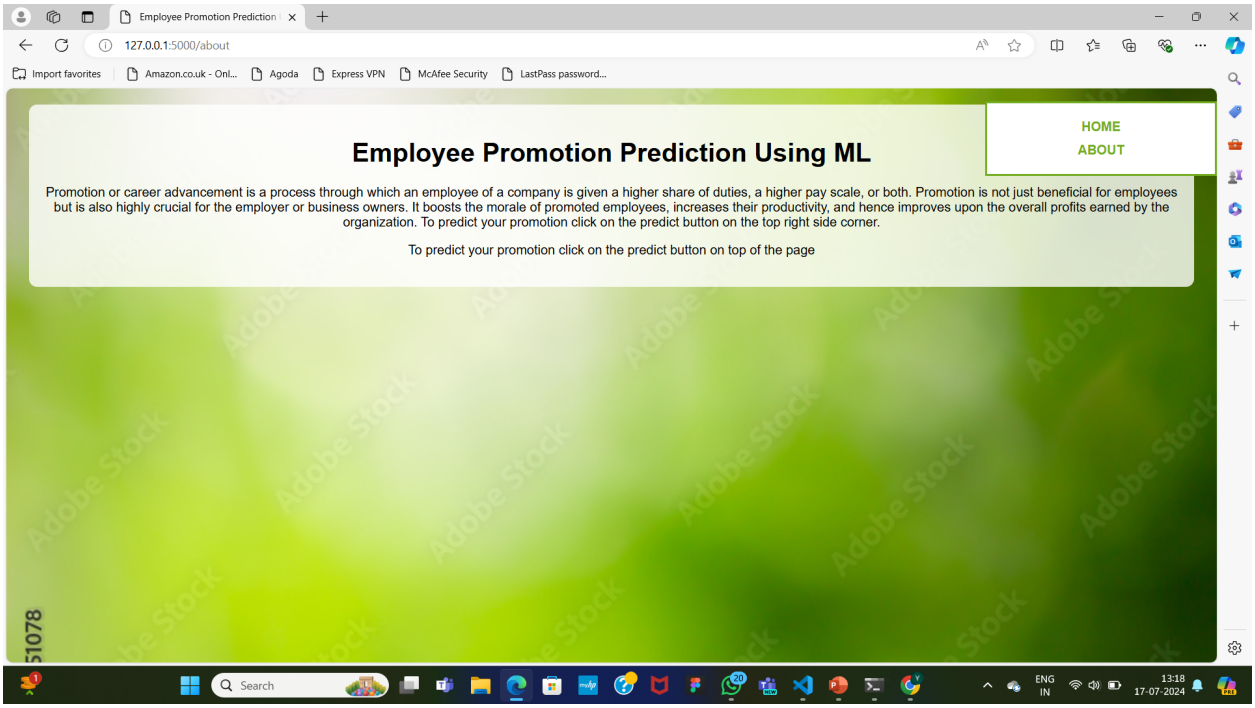
6.1. Output Screenshots

• Developed webpage: ➔ Start of the page:

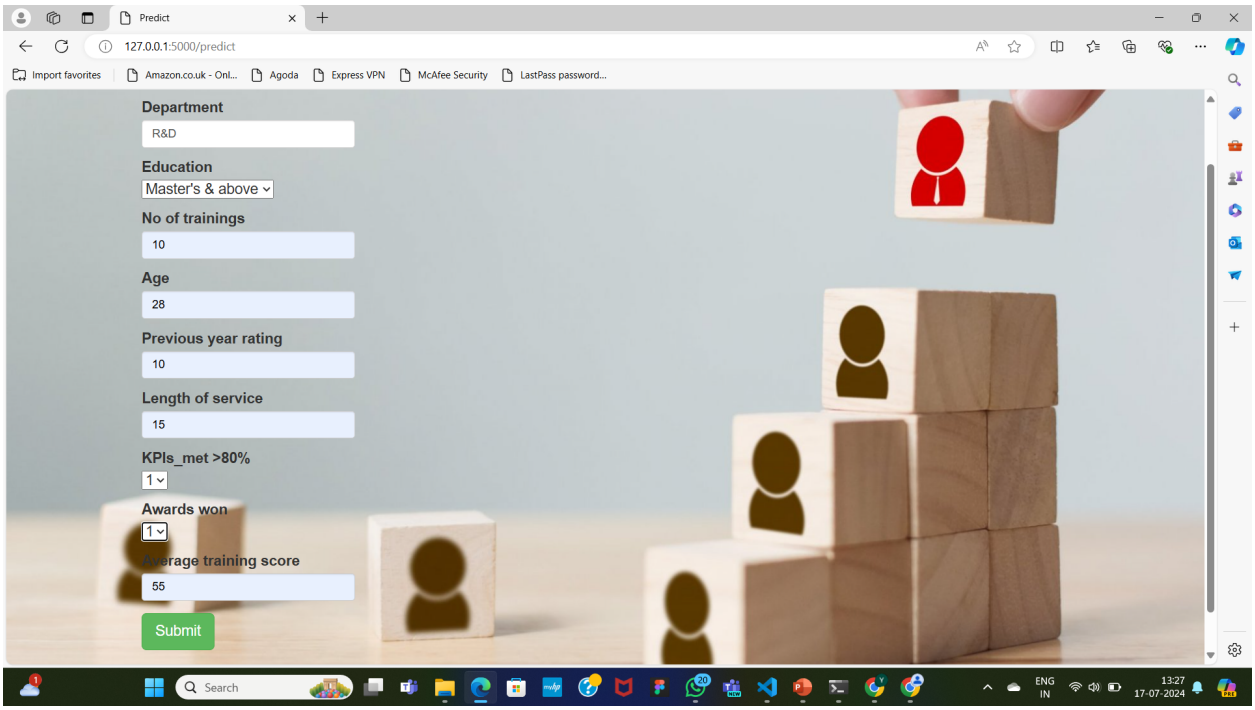



Employee Promotion Prediction
127.0.0.1:5000
Import favorites
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Predict
About Us
127.0.0.1:5000/about
Search
ENG
IN
13:17
17-07-2024

➔ In between:


Employee Promotion Prediction
127.0.0.1:5000/about
Import favorites
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
HOME
ABOUT
Employee Promotion Prediction Using ML
Promotion or career advancement is a process through which an employee of a company is given a higher share of duties, a higher pay scale, or both. Promotion is not just beneficial for employees but is also highly crucial for the employer or business owners. It boosts the morale of promoted employees, increases their productivity, and hence improves upon the overall profits earned by the organization. To predict your promotion click on the predict button on the top right side corner.
To predict your promotion click on the predict button on top of the page
Search
ENG
IN
13:18
17-07-2024

➔ For taking inputs: For categorical attributes, form provides options to select



➔ After filling the User's details, click on the ‘Predict’ button it will display the results.