EMPLOYEE PROMOTION PREDICTION USING MACHINE LEARNING

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MINUKURI.ABHILASH REDDY	19UK1A05E5
SHAIKH .TAUSIF AHMED	19UK1A05H1
BABU NAVEEN	19UK1A05L4
AMBADIPELLI.AKHIL RAO	19UK1A05K3

Under the esteemed guidance of

Mrs. P.PREMALATHA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta,Warangal – 506005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005

CERTIFICATE

This is to certify that the UG Project Phase-1 entitled "**EMPLOYEE PROMOTION PREDICTION USING MACHINE LEARNING**" is being submitted by *M.ABHILASH* (H.NO:19UK1A05E5), *SK.TAUSIF AHMED* (H.NO:19UK1A05H1), *BABU.NAVEEN* (H.NO:19UK1A05L4*),* *A.AKHIL RAO* (H.NO:19UK1A05K3) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2022-23, is a record of work carried out by them under the guidance and supervision.

Project Guide
Mrs. P. Premalatha
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dr.P.PRASAD RAO, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to Dr.R.NAVEEN KUMAR, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, Mrs. P. Premalatha Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

CONTENT:

1.INTRODUCTION

Overview :

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.

Proposed System :

The client is facing a problem in identifying the right people for promotion. The company needs help in identifying the eligible candidates at a particular checkpoint so that they can expedite the entire promotion cycle. This problem can be solved by building a machine learning that automates the process of promoting an employee. we make use of employee datasets to build different classification ML models such as Decision tree, Random forest, KNN, and xgboost. The best model is selected and saved for integration with the flask application.

For better training results we make use of IBM to train the model to deploy the model on IBM.

2.LITERATURE SURVEY

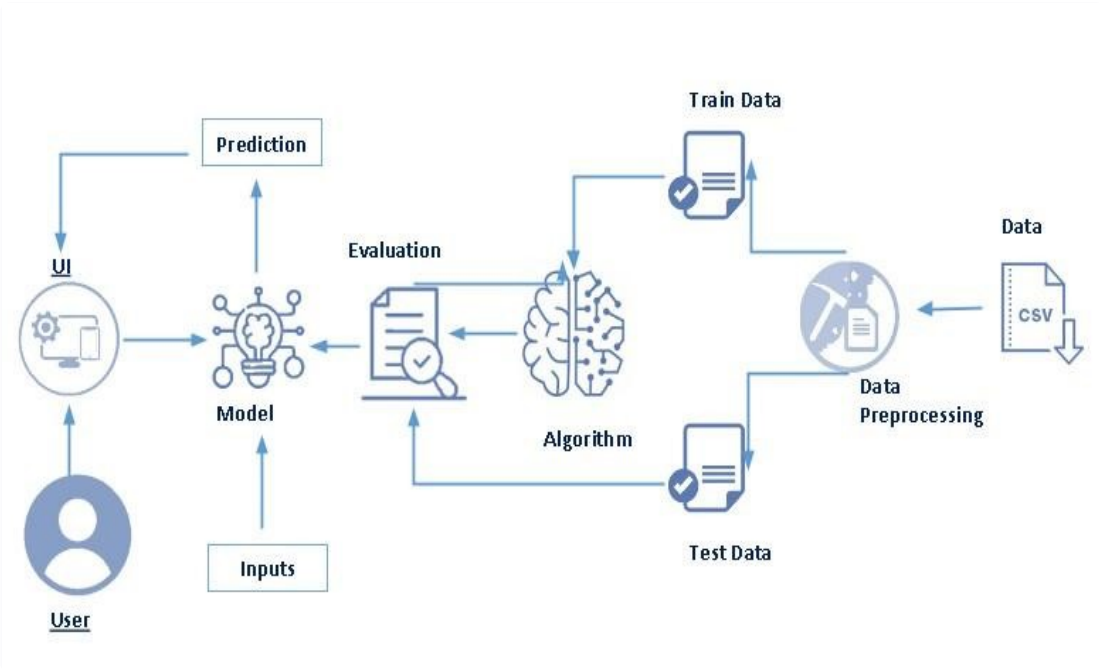
2.1Existing Problem

In [1], computer scientist was once quoted as spoken language, “You remove our prime twenty workers and that we [Microsoft] become a mediocre company”. This statement by computer scientist took our attention to 1 of the main issues of worker attrition at workplaces. worker attrition (turnover) causes a major price to any organization which can afterward impact its overall potency. As per CompData Surveys, over the past 5 years, total turnover has accumulated from fifteen.1 p.c to one8.5 percent. For any organization, finding a well trained and experienced worker may be a complicated task, however it's even additional complicated to interchange such workers. This not solely will increase the many Human Resource (HR) price, however additionally impacts the market price of a corporation. Despite these facts and ground reality, there's very little attention to the literature, that has been seeded to several misconceptions between time unit and workers. Therefore, the aim of this paper is to supply a framework for predicting the worker churn by analyzing the employee’s precise behaviors and attributes mistreatment classification techniques

3.THEORETICAL ANALYSIS

3.1 Block Diagram

Technical Architecture:



3.2Hardware/software Desgning

Hardware requriement

1. 2 GB ram or above
2. Dual core processor or above
3. Internet connection

Software requirements

1. Anaconda Navigator
2. Python Packages
3. IBM Watson Studio

4.EXPERIMENTAL INVESTIGATION

Project Objectives :

By the end of this project:

- You'll be able to understand the problem to classify if it is a regression or a classification kind of problem.
- You will be able to know how to pre-process/clean the data using different data preprocessing techniques.
- You will be able to analyze or get insights into data through visualization.
- Applying different algorithms according to the dataset and based on visualization.
- Have knowledge of data/capping techniques on outliers and some visualization concepts.
- You will be able to know how to build a web application using the Flask framework.

5.FLOWCHART

Project Flow :

- The user interacts with the UI to enter the input.
- Entered input is analyzed by the model which is integrated.
- Once the model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

- Data collection
  - Collect the dataset or create the dataset
- Visualizing and analyzing data
  - Univariate analysis
  - Multivariate analysis
  - Descriptive analysis
- Data pre-processing
  - Drop unwanted features
  - Checking for null values
  - Remove negative data
  - Handling outlier
  - Handling categorical data
  - Handling Imbalanced data
  - Splitting data into train and test
- Model building
  - Import the model building libraries
  - Initializing the model
  - Training and testing the model
  - Evaluating performance of the model
  - Save the model
- Application Building
  - Create an HTML file ○ Build python code

6 .CODE

Predicting Employee Promotion

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.

The client is facing a problem in identifying the right people for promotion. The company needs help in identifying the eligible candidates at a particular checkpoint so that they can expedite the entire promotion cycle.

Data Loading

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of csv file.

```
]: import numpy as np
   import pandas as pd
```

```
]: data=pd.read_csv('D:\Major project\Dataset\emp_promotion.csv')
```

```
] data.head()
]
```

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards
0	65438	Sales & Marketing	region_7	Master's & above	f	sourcing	1	35	5.0	8	1	
1	65141	Operations	region_22	Bachelor's	m	other	1	30	5.0	4	0	
2	7513	Sales & Marketing	region_19	Bachelor's	m	sourcing	1	34	3.0	7	0	
3	2542	Sales & Marketing	region_23	Bachelor's	m	other	2	39	1.0	10	0	
4	48945	Technology	region_26	Bachelor's	m	other	1	45	3.0	2	0	

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54808 entries, 0 to 54807
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   employee_id           54808 non-null  int64
 1   department            54808 non-null  object
 2   region                54808 non-null  object
 3   education             52399 non-null  object
 4   gender                54808 non-null  object
 5   recruitment_channel   54808 non-null  object
 6   no_of_trainings       54808 non-null  int64
 7   age                   54808 non-null  int64
 8   previous_year_rating  50684 non-null  float64
 9   length_of_service     54808 non-null  int64
```

Data Preprocessing

we need to clean the dataset properly in order to fetch good result,for this we need to follow the below steps.

1.Removing unnecessary columns.

2.Handling Null values and dealing with wrongly entered data.

For checking the null values, df.isnull() function is used.

```
data.isnull().any()
```

```
employee_id           False
department            False
region                False
education              True
gender                False
recruitment_channel   False
no_of_trainings       False
age                   False
previous_year_rating   True
length_of_service     False
KPIs_met >80%         False
awards_won?           False
avg_training_score    False
is_promoted           False
dtype: bool
```

we found that education column and previous year rating column has null values.

```
data.isnull().count()
```

```
employee_id             54808
department              54808
region                  54808
education               54808
gender                  54808
recruitment_channel     54808
no_of_trainings         54808
age                     54808
previous_year_rating    54808
length_of_service       54808
KPIs_met >80%           54808
awards_won?             54808
avg_training_score      54808
is_promoted             54808
dtype: int64
```

```
data['education'].unique()
```

```
array(["Master's & above", "Bachelor's", nan, 'Below Secondary'],
      dtype=object)
```

```
data['previous_year_rating'].unique()
```

```
array([ 5.,  3.,  1.,  4., nan,  2.])
```

```
data['department'].unique()
```

```
array(['Sales & Marketing', 'Operations', 'Technology', 'Analytics',
       'R&D', 'Procurement', 'Finance', 'HR', 'Legal'], dtype=object)
```

```
data['region'].unique()
```

```
data['region'].unique()
```

```
array(['region_7', 'region_22', 'region_19', 'region_23', 'region_26',
       'region_2', 'region_20', 'region_34', 'region_1', 'region_4',
       'region_29', 'region_31', 'region_15', 'region_14', 'region_11',
       'region_5', 'region_28', 'region_17', 'region_13', 'region_16',
       'region_25', 'region_10', 'region_27', 'region_30', 'region_12',
       'region_21', 'region_8', 'region_32', 'region_6', 'region_33',
       'region_24', 'region_3', 'region_9', 'region_18'], dtype=object)
```

```
data['region'].nunique()
```

```
34
```

```
data['recruitment_channel'].unique()
```

```
array(['sourcing', 'other', 'referred'], dtype=object)
```

```
data['recruitment_channel'].unique()
```

```
array(['sourcing', 'other', 'referred'], dtype=object)
```

```
data['age'].unique()
```

```
array([35, 30, 34, 39, 45, 31, 33, 28, 32, 49, 37, 38, 41, 27, 29, 26, 24,
       57, 40, 42, 23, 59, 44, 50, 56, 20, 25, 47, 36, 46, 60, 43, 22, 54,
       58, 48, 53, 55, 51, 52, 21], dtype=int64)
```

```
data['length_of_service'].unique()
```

```
array([ 8,  4,  7, 10,  2,  5,  6,  1,  3, 16,  9, 11, 26, 12, 17, 14, 13,
       19, 15, 23, 18, 20, 22, 25, 28, 24, 31, 21, 29, 30, 34, 27, 33, 32,
       37], dtype=int64)
```

```
print(data['education'].value_counts())
```

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

```
data.size
```

```
767312
```

```
data['education']=data['education'].fillna(data['education'].mode()[0])
```

```
data.isnull().any()
```

```
employee_id           False
department            False
region                False
education             False
gender                False
recruitment_channel   False
no_of_trainings       False
age                   False
previous_year_rating   True
length_of_service     False
KPIs_met >80%         False
awards_won?           False
avg_training_score    False
is_promoted           False
dtype: bool
```

```
print(data['previous_year_rating'].value_counts())
```

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

```
data['previous_year_rating'].fillna(data['previous_year_rating'].mean())
```

```
0         5.0
1         5.0
2         3.0
3         1.0
4         3.0
        ...
54803     3.0
54804     2.0
54805     5.0
54806     1.0
54807     1.0
Name: previous_year_rating, Length: 54808, dtype: float64
```

```
data.isnull().any()
```

```
employee_id           False
department            False
region                False
education             False
gender                False
recruitment_channel   False
no_of_trainings       False
age                   False
previous_year_rating   True
length_of_service     False
KPIs_met >80%         False
awards_won?           False
avg_training_score    False
is_promoted           False
dtype: bool
```

```
data['previous_year_rating']=data['previous_year_rating'].replace(np.nan,3.329256)
```

```
data.isnull().any()
```

```
employee_id           False
department            False
region                False
education             False
gender                False
recruitment_channel   False
no_of_trainings       False
age                   False
previous_year_rating  False
length_of_service     False
KPIs_met >80%         False
awards_won?           False
avg_training_score    False
is_promoted           False
dtype: bool
```

hese two features don't have
nd for previous year rating is 3.

Remove Negative Data

Employees with poor performance got promoted. It affects model performance. So, negative value should be removed.

Here list comprehension is used to find the negative data.

Negative data: Employees with no awards, previous year rating was 1.0, KPIs less than 80% and average training score is less than 60.

```
negative=data[(data['KPIs_met >80%']==0) & (data['awards_won?']==0) & (data['previous_year_rating']==1.0) &
        (data['is_promoted']==1) & (data['avg_training_score']<60)]
negative
```

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	av
31860	29663	Sales & Marketing	region_22	Bachelor's	m	referred	1	27	1.0	2	0	
51374	28327	Sales & Marketing	region_2	Bachelor's	m	sourcing	1	31	1.0	5	0	

Now, negative data is removed.

Drop Unwanted Features

We are building the model to predict the promotion of employees.

No organizations will promote their employees by gender, region, and recruitment channel. So, these features are removed from the dataset

```
data=data.drop(['gender','region','recruitment_channel'],axis=1)
```

```
data
```

	employee_id	department	education	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score	is_prom
0	65438	Sales & Marketing	Master's & above	1	35	5.0	8	1	0	49	
1	65141	Operations	Bachelor's	1	30	5.0	4	0	0	60	
2	7513	Sales & Marketing	Bachelor's	1	34	3.0	7	0	0	50	
3	2542	Sales & Marketing	Bachelor's	2	39	1.0	10	0	0	50	
4	48945	Technology	Bachelor's	1	45	3.0	2	0	0	73	
...	...	...	...	...	...	...	...	...	...	...	
54803	3030	Technology	Bachelor's	1	48	3.0	17	0	0	78	
54804	74592	Operations	Master's & above	1	37	2.0	6	0	0	56	

```
data.drop(index=[31860,51374],inplace=True)
```

```
data
```

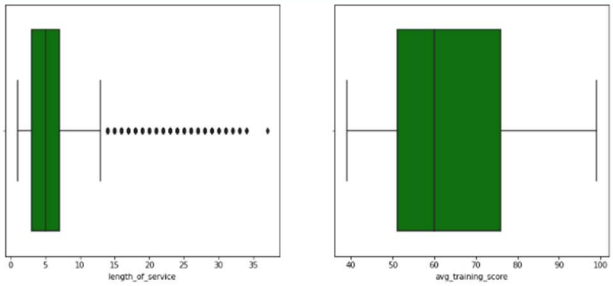
	employee_id	department	education	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score	is_prom
0	65438	Sales & Marketing	Master's & above	1	35	5.0	8	1	0	49	
1	65141	Operations	Bachelor's	1	30	5.0	4	0	0	60	
2	7513	Sales & Marketing	Bachelor's	1	34	3.0	7	0	0	50	
3	2542	Sales & Marketing	Bachelor's	2	39	1.0	10	0	0	50	
4	48945	Technology	Bachelor's	1	45	3.0	2	0	0	73	
...	...	...	...	...	...	...	...	...	...	...	
54803	3030	Technology	Bachelor's	1	48	3.0	17	0	0	78	
54804	74592	Operations	Master's & above	1	37	2.0	6	0	0	56	
54805	13918	Analytics	Bachelor's	1	27	5.0	3	1	0	79	
54806	13614	Sales & Marketing	Bachelor's	1	29	1.0	2	0	0	45	
54807	51526	HR	Bachelor's	1	27	1.0	5	0	0	49	

54806 rows × 11 columns

```
import seaborn as sns
import matplotlib.pyplot as plt
```

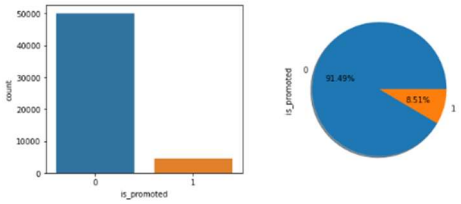
```
plt.figure(figsize=(14,6))
plt.subplot(121)
sns.boxplot(data['length_of_service'],color='g')
plt.subplot(122)
sns.boxplot(data['avg_training_score'],color='g')
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



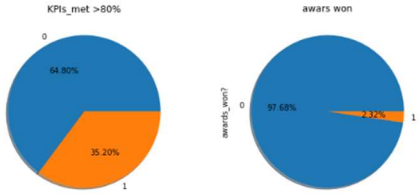

```
plt.figure(figsize=(10,4))
plt.subplot(121)
sns.countplot(data['is_promoted'])
plt.subplot(122)
data['is_promoted'].value_counts().plot(kind='pie',autopct = '%.2f%%',shadow=True)
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```




```
plt.figure(figsize=(16,10))
plt.subplot(231)
plt.axis('off')
plt.title('KPIs_met >80%')
data['KPIs_met >80%'].value_counts().plot(kind='pie',shadow=True,autopct='%.2f%%')
plt.subplot(232)
plt.title('awars won')
data['awards_won?'].value_counts().plot(kind='pie',shadow=True,autopct='%.2f%%')

plt.show()
```




```
data.describe()
```

	employee_id	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score	is_promoted
count	54806.000000	54806.000000	54806.000000	54806.000000	54806.000000	54806.000000	54806.000000	54806.000000	54806.000000
mean	39196.202879	1.253020	34.804127	3.329341	5.865599	0.351987	0.023173	63.386947	0.085137
std	22586.909147	0.609273	7.660219	1.211601	4.265138	0.477594	0.150453	13.371764	0.279088
min	1.000000	1.000000	20.000000	1.000000	1.000000	0.000000	0.000000	39.000000	0.000000
25%	19669.250000	1.000000	29.000000	3.000000	3.000000	0.000000	0.000000	51.000000	0.000000
50%	39226.500000	1.000000	33.000000	3.000000	5.000000	0.000000	0.000000	60.000000	0.000000
75%	58731.500000	1.000000	39.000000	4.000000	7.000000	1.000000	0.000000	76.000000	0.000000
max	78298.000000	10.000000	60.000000	5.000000	37.000000	1.000000	1.000000	99.000000	1.000000

Handling Outliers

With the help of boxplot, outliers are visualized (refer activity 3 univariate analysis). And here we are going to find upper bound and lower bound of Na_to_K feature with some mathematical formula.

-> To find upper bound we have to multiply IQR (Interquartile range) with 1.5 and add it with 3rd quantile. To find lower bound instead of adding, subtract it with 1st quantile. Take image attached below as your reference.

->If outliers are removed, we lose more data. It will impact model performance.

->Here removing outliers is impossible. So, the capping technique is used on outliers.

->Capping: Replacing the outliers with upper bound values.

```
print('q1:',q1)
print('q2:',q3)
print('IQR:',IQR)
print('Upperbound:',upperBound)
print('Lowerbound:',lowerBound)
print('skeweddata:',len(data[data['length_of_service']>upperBound]))
```

```
q1: 3.0
q2: 7.0
IQR: 4.0
Upperbound: 13.0
Lowerbound: 3.0
skeweddata: 3489
```

Here outerliers can't be removed.Employee with higher length of service s has higher promotion percentage. So, capping is done on this feature.

```
pd.crosstab(data['length_of_service']>upperBound,data['is_promoted'])
```

is_promoted	0	1
length_of_service		
False	46885	4432
True	3255	234

```
data['length_of_service']=[upperBound if x>upperBound else x for x in data['length_of_service']]
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 54806 entries, 0 to 54807
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   employee_id           54806 non-null  int64
 1   department            54806 non-null  object
 2   education             54806 non-null  object
 3   no_of_trainings       54806 non-null  int64
 4   age                   54806 non-null  int64
 5   previous_year_rating  54806 non-null  float64
 6   length_of_service     54806 non-null  float64
 7   KPIs_met >80%         54806 non-null  int64
 8   awards_won?           54806 non-null  int64
 9   avg_training_score    54806 non-null  int64
 10  is_promoted           54806 non-null  int64
dtypes: float64(2), int64(7), object(2)
memory usage: 5.0+ MB
```

```
data.select_dtypes('object').head(10)
```

	department	education
0	Sales & Marketing	Master's & above
1	Operations	Bachelor's
2	Sales & Marketing	Bachelor's
3	Sales & Marketing	Bachelor's
4	Technology	Bachelor's
5	Analytics	Bachelor's
6	Operations	Bachelor's
7	Operations	Master's & above
8	Analytics	Bachelor's
9	Sales & Marketing	Master's & above

```
data['education']=data['education'].replace(('Below secondary',"Bachelor's","Master's & above"),(1,2,3))
```

```
data
```

	employee_id	department	education	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score	is_pron
0	65438	Sales & Marketing	3	1	35	5.0	8.0	1	0	49	
1	65141	Operations	2	1	30	5.0	4.0	0	0	60	
2	7513	Sales & Marketing	2	1	34	3.0	7.0	0	0	50	
3	2542	Sales & Marketing	2	2	39	1.0	10.0	0	0	50	
4	48945	Technology	2	1	45	3.0	2.0	0	0	73	
...	...	...	...	...	...	...	...	...	...	...	
54803	3030	Technology	2	1	48	3.0	13.0	0	0	78	
54804	74592	Operations	3	1	37	2.0	6.0	0	0	56	
54805	13918	Analytics	2	1	27	5.0	3.0	1	0	79	
54806	13614	Sales & Marketing	2	1	29	1.0	2.0	0	0	45	
54807	51526	HR	2	1	27	1.0	5.0	0	0	49	

```
import warnings
warnings.filterwarnings('ignore')
```

lable encoder

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
data['department']=lb.fit_transform(data['department'])
```

```
data.head()
```

	employee_id	department	education	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score	is_promoted
0	65438	7	3	1	35	5.0	8.0	1	0	49	0
1	65141	4	2	1	30	5.0	4.0	0	0	60	0
2	7513	7	2	1	34	3.0	7.0	0	0	50	0
3	2542	7	2	2	39	1.0	10.0	0	0	50	0
4	48945	8	2	1	45	3.0	2.0	0	0	73	0

```
y=data['is_promoted']
```

```
x=data.drop('is_promoted',axis=1)
```

```
x.shape
```

```
(54806, 10)
```

```
y.shape
```

```
(54806,)
```

```
from sklearn.model_selection import train_test_split
```

7.FLASK APP:-

Pickle: Pickle is a module in Python used for serializing and de-serializing Python objects. Flask: Refer prior knowledge section mentioned above.

```
import pickle
from flask import Flask, render_template, request
```

Load the saved model. Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

```
model = pickle.load(open('model.pkl', 'rb'))

app = Flask(__name__)
```

Render HTML page:

```
@app.route('/')
def home():
    return render_template('home.html')


@app.route('/home')
def home1():
    return render_template('home.html')


@app.route('/about')
def about():
    return render_template('about.html')


@app.route('/predict')
def predict():
    return render_template('predict.html')
```

Retrieves the value from UI:

```
@app.route('/pred', methods=['POST'])
def pred():
    department = request.form['department']
    education = request.form['education']
    if education == '1':
        education = 1
    elif education == '2':
        education = 2
    else:
        education = 3
    no_of_trainings = request.form['no_of_trainings']
    age = request.form['age']
    previous_year_rating = request.form['previous_year_rating']
    length_of_service = request.form['length_of_service']
    KPIs = request.form['KPIs']
    if KPIs == '0':
        KPIs = 0
    else:
        KPIs = 1
    awards_won = request.form['awards_won']
    if awards_won == '0':
        awards_won = 0
    else:
        awards_won = 1
    avg_training_score = request.form['avg_training_score']
    total = [[department, education, no_of_trainings, age, float(previous_year_rating), float(length_of_service),
              KPIs, awards_won, avg_training_score]]
    prediction = model.predict(total)
    if prediction == 0:
        text = 'Sorry, you are not eligible for promotion'
    else:
```

Main Function:

```
if __name__ == '__main__':
    app.run(debug=True)
```

Run The Application :

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command
- Navigate to the localhost where you can view your web page.
- Click on the predict button from the top right corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

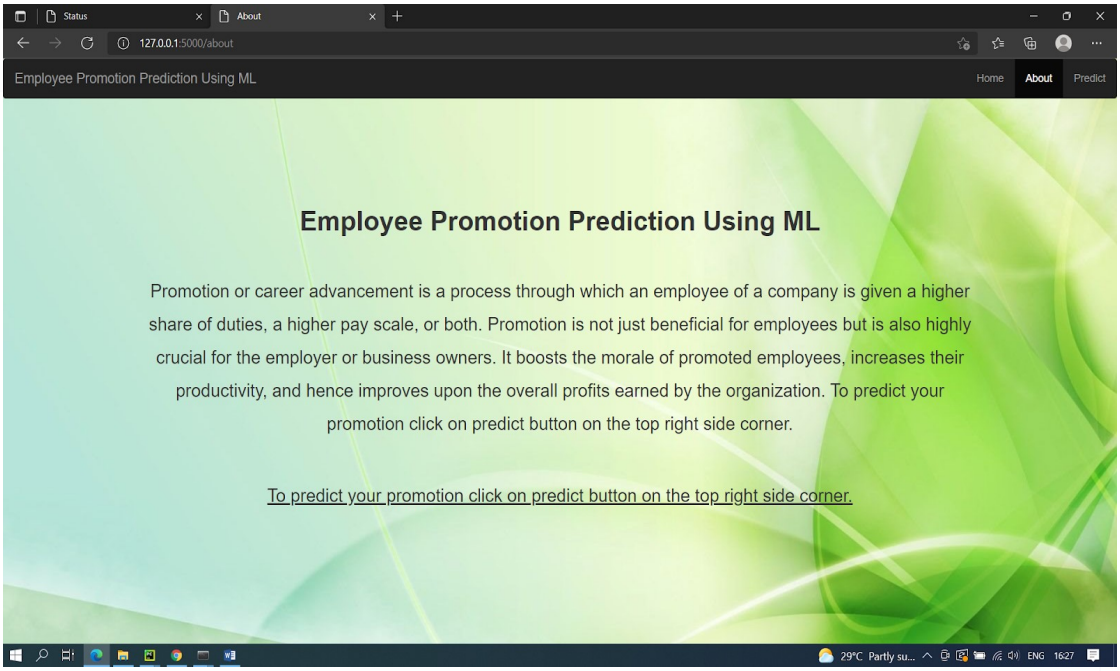
```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 135-972-108
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Now paste the URL on the browser, you will redirect to home.html page. Let’s look our home page

8. RESULT

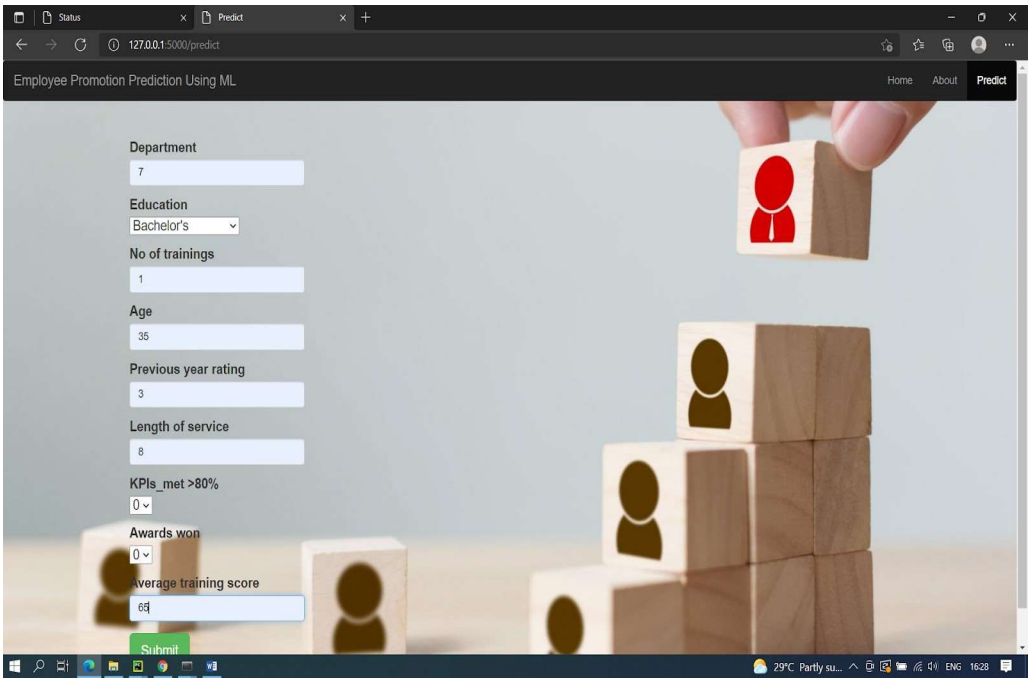
Home page:

To know about the project click on About button on right top corner. Now it will redirect to about.html page

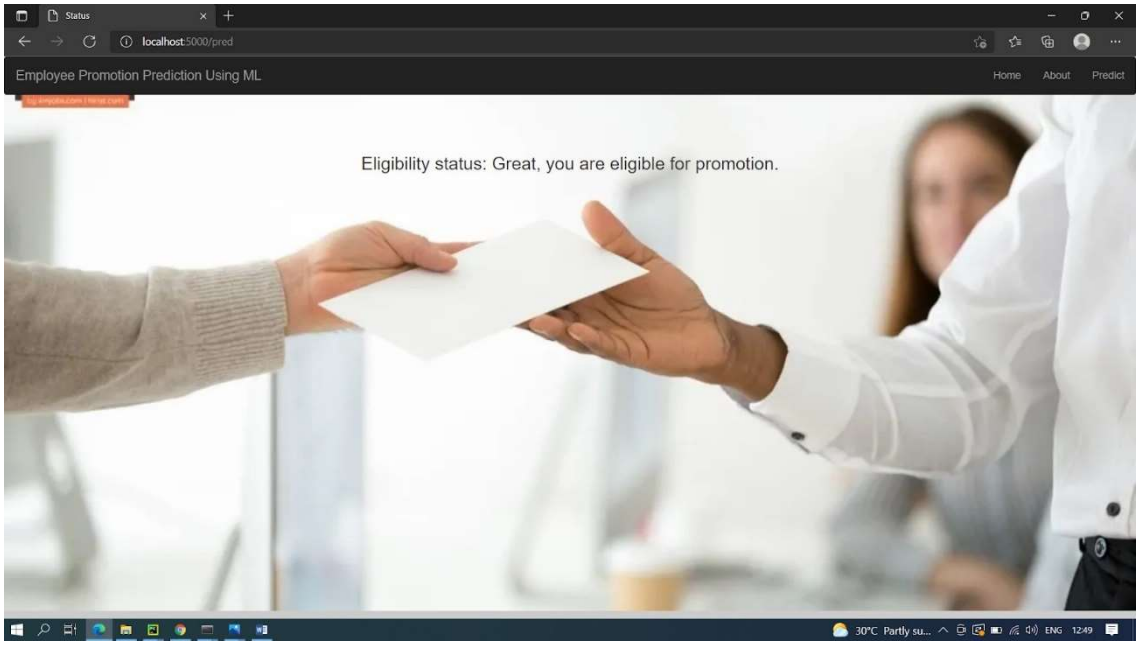


To predict your promotion click on predict button on right top corner. It will redirect to predict.html page. Now give your inputs and click on submit button. Output will be displayed in submit.html page

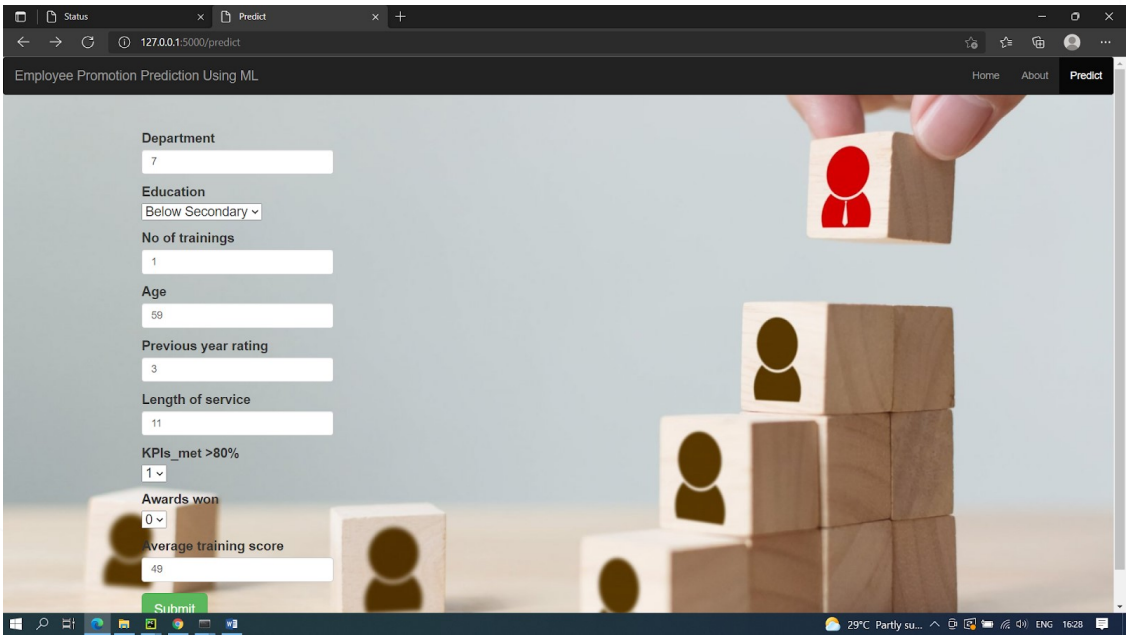
Input 1:



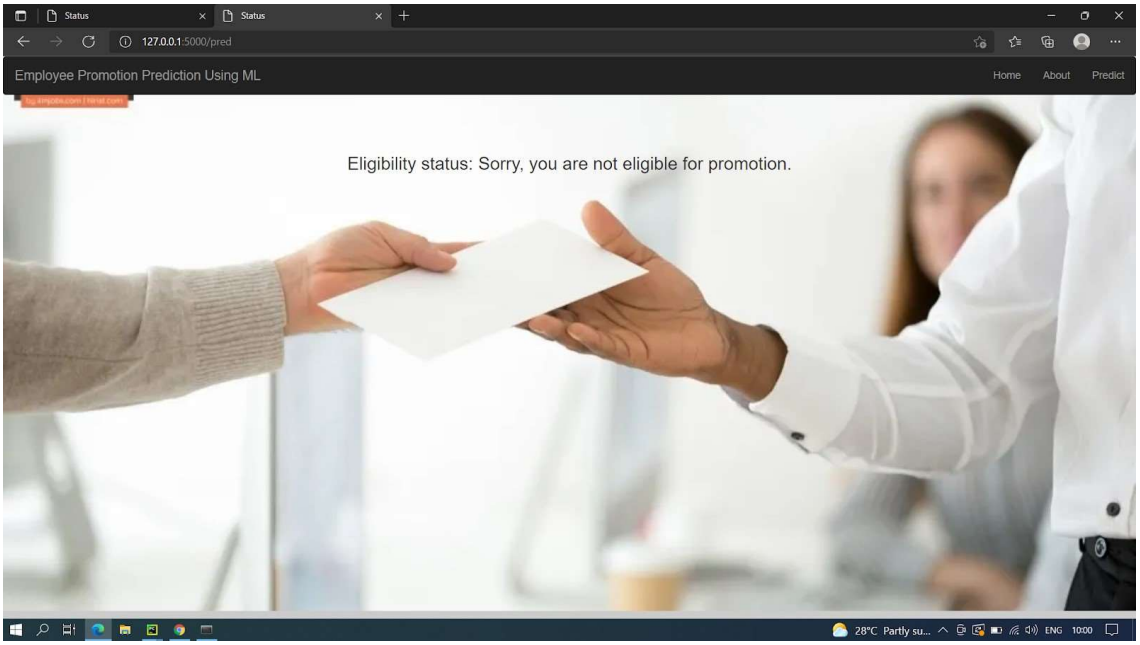
Output 1:



Input 2:



Output 2:



8. ADVANTAGES AND DISADVANTAGES

Advantages :

- Efficient program for Employee promotion prediction
- Accurate output is produced
- Will predict Employee promotion with extreme accuracy
- Relatively inexpensive and fast

Disadvantages :

- It will work in all condition but some condition it may not give correct output

9. CONCLUSION

Promotions have a favorable, significant and beneficial impact on employee work performance in human resources process. In this study a prediction model for employee promotion is proposed by using RF method.

10. FUTURE SCOPE

This program allows users to predict if the Employee is promoted or not. By the help of this prediction, we can use this program and we can ensure that the is promoted or not. It helps reduce the stress on user identifying important promoted employee easily.

11. BIBLIOGRAPHY

- https://towardsdatascience.com/will-your-employee-leave-a-machine-learning-model-8484c2a6663e
- https://medium.com/digitaladoption101/5-ways-predictive-analytics-can-contribute-toemployee-engagement-e37c0ea542a