

Employee Promotion Prediction – Case Study.

Course Name: Advanced Machine Learning.

Date: Monday, September 2, 2024

Student Name: Vijaya Laxmi Kumbaji

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model performance summary for hyperparameter tuning.
- Appendix

Executive Summary

- The HR at the JMD company is using the data from previous years to determine eligibility of employees for Promotion. Every year it gets difficult due to the large number of parameters / features and amount of data to help determine the factors in promotion eligibility. To help reduce these efforts and streamline the process JMD as decided to adopt machine modelling techniques.
- The several factors that affect as of today were years of service at JMD, awards won, performance from previous years, trainings, regions, recruitment channels, age, education, total years of experience etc. in order to narrow down and identify features that absolutely help determine is not clear yet and has become very time consuming process each year and is only increasing difficulty every year.

Business Problem overview and solution approach

- The HR at the JMD company is using the data from previous years to determine eligibility of employees for Promotion. Every year it gets difficult due to the large number of parameters / features and amount of data to help determine the promotion eligibility. To help reduce these efforts and streamline the process we developed models.
- Every year the HR Team faced the problem in identifying what factors help to determine employee promotion, there is so much data and also different features holding information about employees like region, department, age, number of years of service, awards won, training courses and if an promoted in the past. This made it difficult and needed to identify that helps streamline the process and automates it..

Solution approach

The HR at the JMD company is using the data from previous years to determine eligibility of employees for Promotion. To help reduce time consuming and strenuous efforts by HR Team and streamline the process we developed models.

I as a data scientist created 10 different types of Machine models created with training data, test data and validation data sets. Based on performance and analysis of these models XGBoost and GradientBoost models showed the highest accuracy of 96% and overall recall of 88%.

(1.0) Various types of bivariate and univariate plots were developed for analysis.

(2.0) Dataset size is 54808 records and total 13 columns including Object types, int and float types of data and there were no duplicate rows. Some Columns had duplicate values.

(3.0) Statistical information was derived, i.e mean, standard deviation, averages etc were calculated for the dataset of 54808 records.

(4.0) Unique values were found for each column in the dataset as below. , there are more male employees than female employees.

Solution Approach continued...

- Please mention the solution approach / methodology:
- The solution was provided by created 10 different types of Machine models created with training data, test data and validation data sets. Dataset was split into training set, validation and testing set. Based on permance and analysis of these models XGBoost and GradientBoost models showed the highest accuracy of 94% and overall recall of 88%.
- Observations from univariate and Bivariate analysis shows as described in Exploratory data analysis the features helped predict the employees' eligibility for promotion. the the following attributes help determine the employee promotion prediction better as shown in the plots.
- 1.0 employee length of service, 2.0 Average training score(s), 3.0 Previous year rating, (4.0) recruitment channel. Etc., non contributing features were region, age, education, department etc did not contribute much towards the prediction.

solution approach continued...

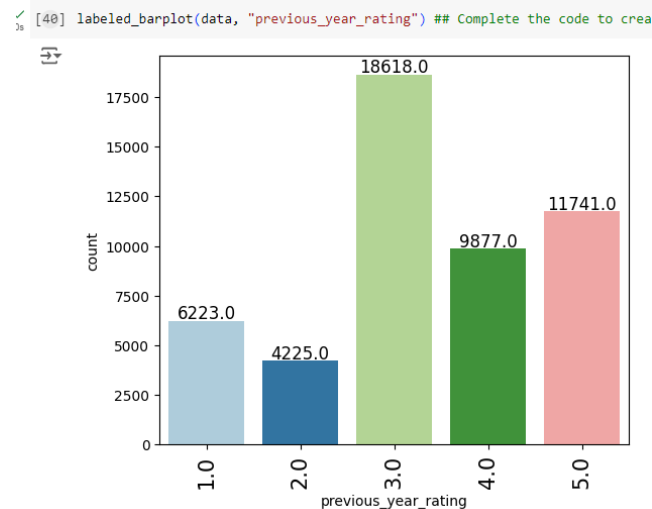
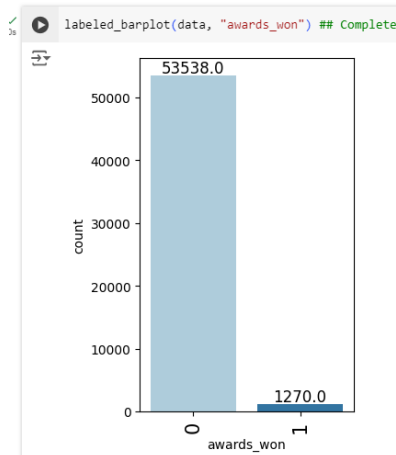
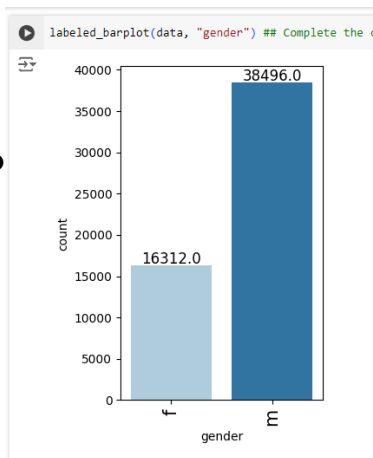
✓
0s

```
[20] data.nunique()
```



	0
employee_id	54808
department	9
region	34
education	3
gender	2
recruitment_channel	3
no_of_trainings	10
age	41
previous_year_rating	5
length_of_service	35
awards_won	2
avg_training_score	59
is_promoted	2

Exploratory Data Analysis results

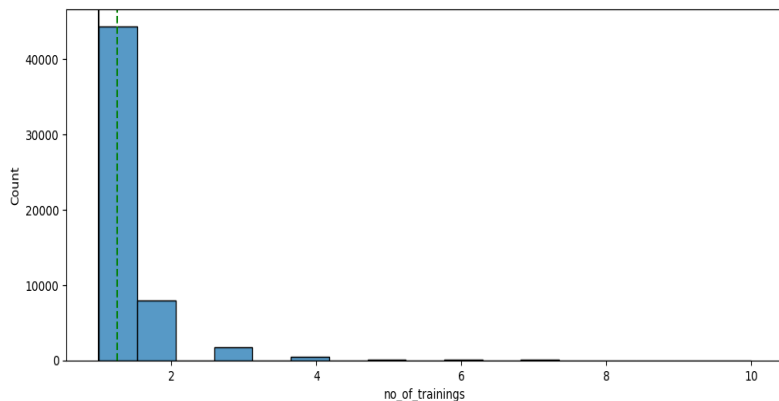


- Number of Male employees: 38496, female employees: 16312.
- Previous year ratings were good for 11741 employees and employees that received merit based awards were 1270 employees including male / female.

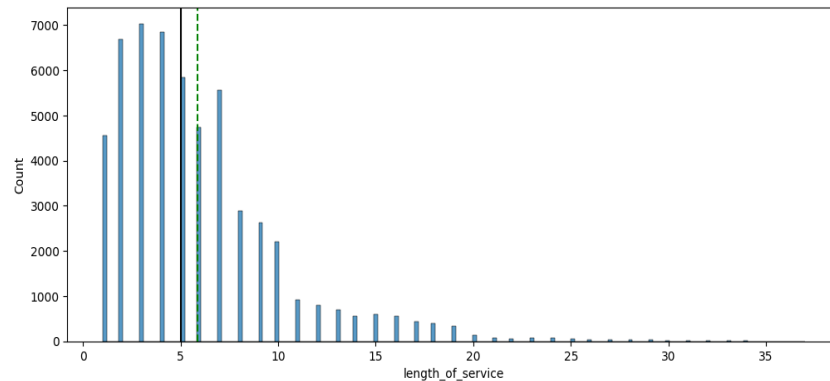
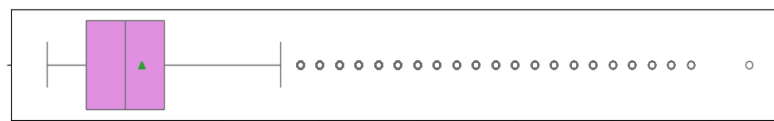
Exploratory Data Analysis results continued...

- Employee average service, previous year rating, average score ratings:

```
[29] histogram_boxplot(data, "no_of_trainings")
```



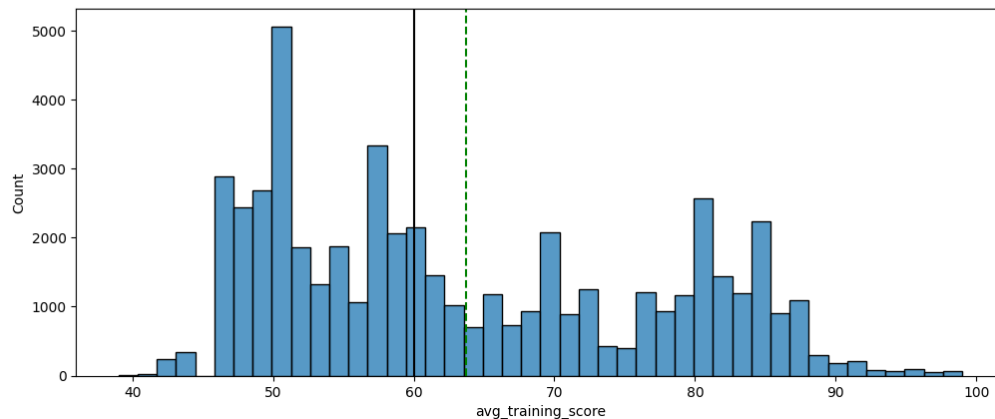
```
[32] histogram_boxplot(data, "length_of_service") ## Complete the code to create histogram_boxplot for 'length_of_service'
```



Exploratory Data Analysis results continued...

- Employee average service, previous year rating, average score ratings:

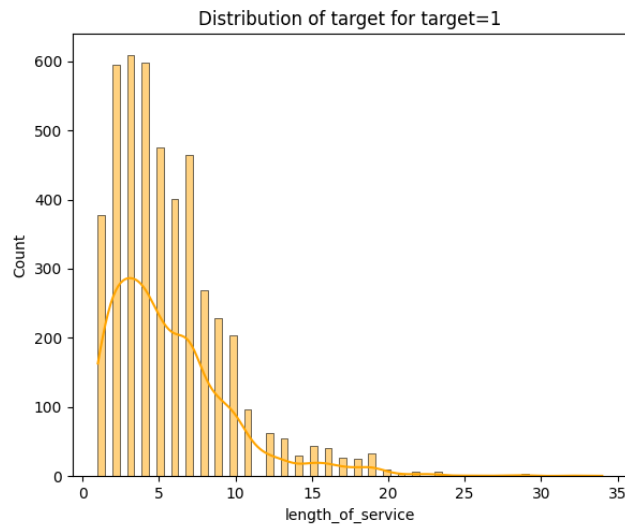
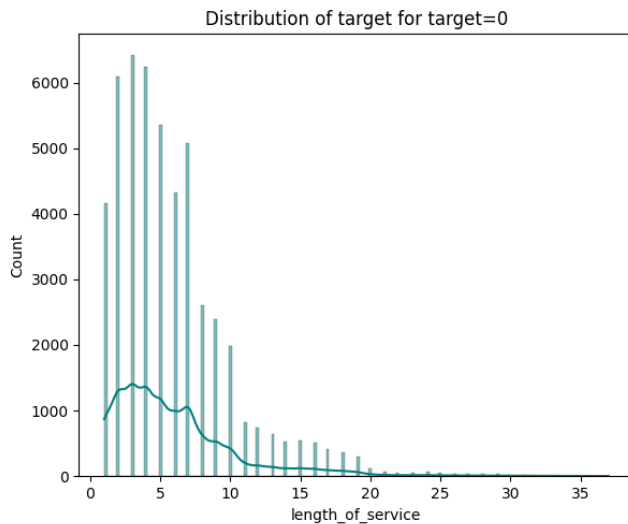
```
[33] histogram_boxplot(data, "avg_training_score") ## Complete the code to create histogram_boxplot for 'avg_training_score'
```



Exploratory Data Analysis results continued...

- Is_promoted column was used as a target variable in Bivariate plots and following conclusions were drawn.

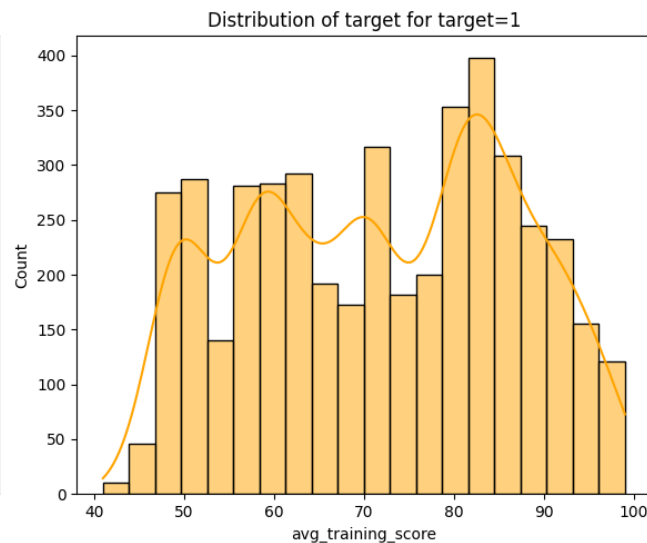
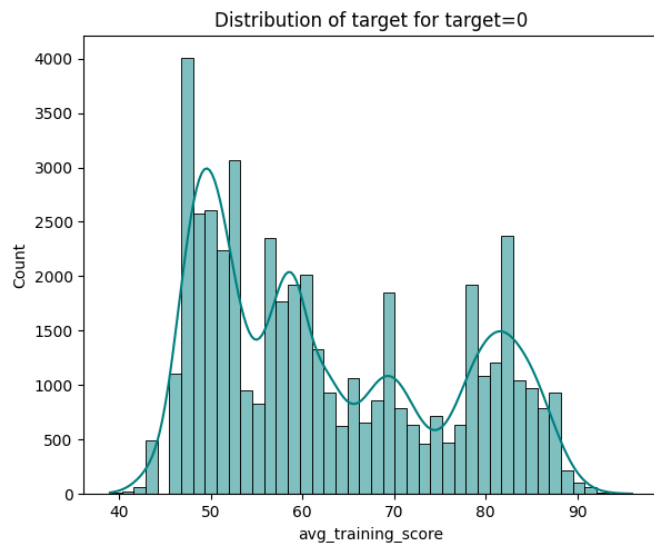
```
[48] distribution_plot_wrt_target(data, "length_of_service", "is_promoted") ## Complete the code to create distribution_plot for length_of_service vs is_promoted
```



Exploratory Data Analysis results continued...

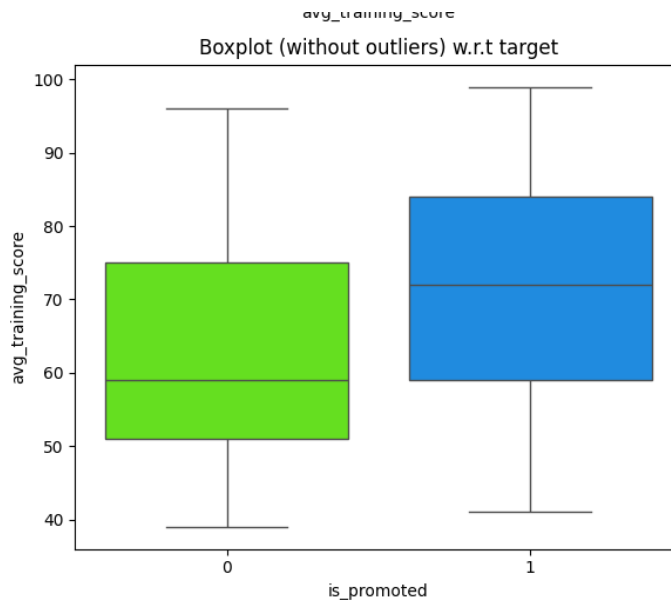
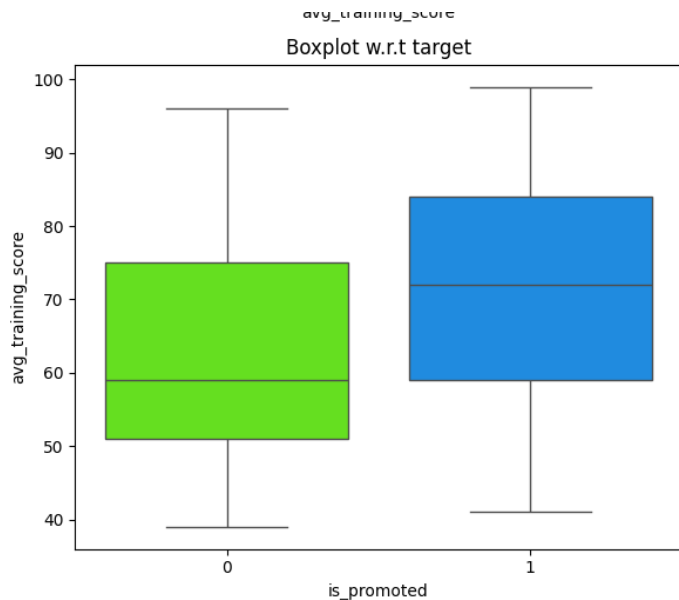
- Is_promoted column was used as a target variable in Bivariate plots and following conclusions were drawn.

```
[49] distribution_plot_wrt_target(data, "avg_training_score", "is_promoted") ## Complete the code to create distribution_plot for avg_training_score vs is_promot
```



Exploratory Data Analysis results continued...

- Is_promoted column was used as a target variable in Bivariate plots and following conclusions were drawn.



Exploratory Data Analysis results continued...

- Correlation map / heat map was plotted as follows:

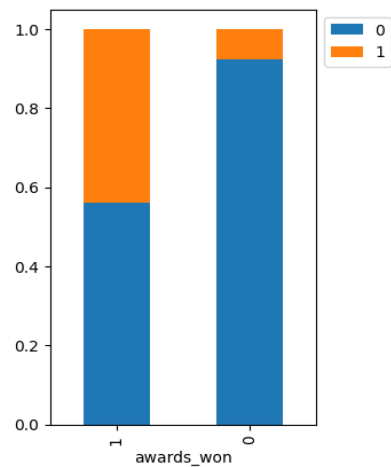


Exploratory Data Analysis results continued...

- Stacked bar plot showed against is_promoted against awards_won, previous_year_rating and also recruitment_channel which had least effect..

```
[58] stacked_barplot(data,"awards_won", "is_promoted")
```

is_promoted	0	1	All
awards_won			
All	50140	4668	54808
0	49429	4109	53538
1	711	559	1270



```
gdf.groupby(["previous_year_rating", "recruitment_channel", "is_promoted"]).size()
```

previous_year_rating	recruitment_channel	is_promoted	size
1	referred	0	48002
1	referred	1	14711
1	referred	All	62713
2	referred	0	81081
2	referred	1	7700
2	referred	All	88781
3	referred	0	88781
3	referred	1	181
3	referred	All	88962
4	referred	0	88781
4	referred	1	181
4	referred	All	88962
5	referred	0	88781
5	referred	1	181
5	referred	All	88962
6	referred	0	88781
6	referred	1	181
6	referred	All	88962
7	referred	0	88781
7	referred	1	181
7	referred	All	88962
8	referred	0	88781
8	referred	1	181
8	referred	All	88962
9	referred	0	88781
9	referred	1	181
9	referred	All	88962
10	referred	0	88781
10	referred	1	181
10	referred	All	88962
11	referred	0	88781
11	referred	1	181
11	referred	All	88962
12	referred	0	88781
12	referred	1	181
12	referred	All	88962
13	referred	0	88781
13	referred	1	181
13	referred	All	88962
14	referred	0	88781
14	referred	1	181
14	referred	All	88962
15	referred	0	88781
15	referred	1	181
15	referred	All	88962
16	referred	0	88781
16	referred	1	181
16	referred	All	88962
17	referred	0	88781
17	referred	1	181
17	referred	All	88962
18	referred	0	88781
18	referred	1	181
18	referred	All	88962
19	referred	0	88781
19	referred	1	181
19	referred	All	88962
20	referred	0	88781
20	referred	1	181
20	referred	All	88962
21	referred	0	88781
21	referred	1	181
21	referred	All	88962
22	referred	0	88781
22	referred	1	181
22	referred	All	88962
23	referred	0	88781
23	referred	1	181
23	referred	All	88962
24	referred	0	88781
24	referred	1	181
24	referred	All	88962
25	referred	0	88781
25	referred	1	181
25	referred	All	88962
26	referred	0	88781
26	referred	1	181
26	referred	All	88962
27	referred	0	88781
27	referred	1	181
27	referred	All	88962
28	referred	0	88781
28	referred	1	181
28	referred	All	88962
29	referred	0	88781
29	referred	1	181
29	referred	All	88962
30	referred	0	88781
30	referred	1	181
30	referred	All	88962
31	referred	0	88781
31	referred	1	181
31	referred	All	88962
32	referred	0	88781
32	referred	1	181
32	referred	All	88962
33	referred	0	88781
33	referred	1	181
33	referred	All	88962
34	referred	0	88781
34	referred	1	181
34	referred	All	88962
35	referred	0	88781
35	referred	1	181
35	referred	All	88962
36	referred	0	88781
36	referred	1	181
36	referred	All	88962
37	referred	0	88781
37	referred	1	181
37	referred	All	88962
38	referred	0	88781
38	referred	1	181
38	referred	All	88962
39	referred	0	88781
39	referred	1	181
39	referred	All	88962
40	referred	0	88781
40	referred	1	181
40	referred	All	88962
41	referred	0	88781
41	referred	1	181
41	referred	All	88962
42	referred	0	88781
42	referred	1	181
42	referred	All	88962
43	referred	0	88781
43	referred	1	181
43	referred	All	88962
44	referred	0	88781
44	referred	1	181
44	referred	All	88962
45	referred	0	88781
45	referred	1	181
45	referred	All	88962
46	referred	0	88781
46	referred	1	181
46	referred	All	88962
47	referred	0	88781
47	referred	1	181
47	referred	All	88962
48	referred	0	88781
48	referred	1	181
48	referred	All	88962
49	referred	0	88781
49	referred	1	181
49	referred	All	88962
50	referred	0	88781
50	referred	1	181
50	referred	All	88962
51	referred	0	88781
51	referred	1	181
51	referred	All	88962
52	referred	0	88781
52	referred	1	181
52	referred	All	88962
53	referred	0	88781
53	referred	1	181
53	referred	All	88962
54	referred	0	88781
54	referred	1	181
54	referred	All	88962
55	referred	0	88781
55	referred	1	181
55	referred	All	88962
56	referred	0	88781
56	referred	1	181
56	referred	All	88962
57	referred	0	88781
57	referred	1	181
57	referred	All	88962
58	referred	0	88781
58	referred	1	181
58	referred	All	88962
59	referred	0	88781
59	referred	1	181
59	referred	All	88962
60	referred	0	88781
60	referred	1	181
60	referred	All	88962
61	referred	0	88781
61	referred	1	181
61	referred	All	88962
62	referred	0	88781
62	referred	1	181
62	referred	All	88962
63	referred	0	88781
63	referred	1	181
63	referred	All	88962
64	referred	0	88781
64	referred	1	181
64	referred	All	88962
65	referred	0	88781
65	referred	1	181
65	referred	All	88962
66	referred	0	88781
66	referred	1	181
66	referred	All	88962
67	referred	0	88781
67	referred	1	181
67	referred	All	88962
68	referred	0	88781
68	referred	1	181
68	referred	All	88962
69	referred	0	88781
69	referred	1	181
69	referred	All	88962
70	referred	0	88781
70	referred	1	181
70	referred	All	88962
71	referred	0	88781
71	referred	1	181
71	referred	All	88962
72	referred	0	88781
72	referred	1	181
72	referred	All	88962
73	referred	0	88781
73	referred	1	181
73	referred	All	88962
74	referred	0	88781
74	referred	1	181
74	referred	All	88962
75	referred	0	88781
75	referred	1	181
75	referred	All	88962
76	referred	0	88781
76	referred	1	181
76	referred	All	88962
77	referred	0	88781
77	referred	1	181
77	referred	All	88962
78	referred	0	88781
78	referred	1	181
78	referred	All	88962
79	referred	0	88781
79	referred	1	181
79	referred	All	88962
80	referred	0	88781
80	referred	1	181
80	referred	All	88962
81	referred	0	88781
81	referred	1	181
81	referred	All	88962
82	referred	0	88781
82	referred	1	181
82	referred	All	88962
83	referred	0	88781
83	referred	1	181
83	referred	All	88962
84	referred	0	88781
84	referred	1	181
84	referred	All	88962
85	referred	0	88781
85	referred	1	181
85	referred	All	88962
86	referred	0	88781
86	referred	1	181
86	referred	All	88962
87	referred	0	88781
87	referred	1	181
87	referred	All	88962
88	referred	0	88781
88	referred	1	181
88	referred	All	88962
89	referred	0	88781
89	referred	1	181
89	referred	All	88962
90	referred	0	88781
90	referred	1	181
90	referred	All	88962
91	referred	0	88781
91	referred	1	181
91	referred	All	88962
92	referred	0	88781
92	referred	1	181
92	referred	All	88962
93	referred	0	88781
93	referred	1	181
93	referred	All	88962
94	referred	0	88781
94	referred	1	181
94	referred	All	88962
95	referred	0	88781
95	referred	1	181
95	referred	All	88962
96	referred	0	88781
96	referred	1	181
96	referred	All	88962
97	referred	0	88781
97	referred	1	181
97	referred	All	88962
98	referred	0	88781
98	referred	1	181
98	referred	All	88962
99	referred	0	88781
99	referred	1	181
99	referred	All	88962
100	referred	0	88781
100	referred	1	181
100	referred	All	88962
101	referred	0	88781
101	referred	1	181
101	referred	All	88962
102	referred	0	88781
102	referred	1	181
102	referred	All	88962
103	referred	0	88781
103	referred	1	181
103	referred	All	88962
104	referred	0	88781
104	referred	1	181
104	referred	All	88962
105	referred	0	88781
105	referred	1	181
105	referred	All	88962
106	referred	0	88781
106	referred	1	181
106	referred	All	88962
107	referred	0	88781
107	referred	1	181
107	referred	All	88962
108	referred	0	88781
108	referred	1	181
108	referred	All	88962
109	referred	0	88781
109	referred	1	181
109	referred	All	88962
110	referred	0	88781
110	referred	1	181
110	referred	All	88962
111	referred	0	88781
111	referred	1	181
111	referred	All	88962

Data Preprocessing

- Duplicate value check
- Missing value treatment
- Outlier check (treatment if needed)
- Feature engineering
- Data preparation for modeling

Note: *You can use more than one slide if needed*

Data Preprocessing

- Duplicate value check – there were no duplicate rows in the original dataset. Original dataset was used to create a copy of the dataset (data_for_preprocessing)
- Data_for_preprocessing was created and used to split into train, test and validation sets.
- The shapes of the new datasets were as follows:
- Training set: 43846, 11
- Validation set: 2741, 11
- Test set: 8221, 11
- Data sets were treated for with missing values imputation, “SimpleImputer” with strategy = “most_frequent” for object / string type data and strategy = “median” for int and float data types.

```
# Print the shapes of the resulting datasets
print(X_train.shape, X_val.shape, X_test.shape)
# Print the shapes of the resulting datasets
print("Training set shape:", X_train.shape, y_train.shape)
print("Validation set shape:", X_val.shape, y_val.shape)
print("Test set shape:", X_test.shape, y_test.shape)
```

```
→ (43846, 11) (2741, 11) (8221, 11)
Training set shape: (43846, 11) (43846,)
Validation set shape: (2741, 11) (2741,)
Test set shape: (8221, 11) (8221,)
```

✓ Missing value imputation

```
# Defining the imputers for numerical and categorical variables
imputer_mode = SimpleImputer(strategy="most_frequent")
imputer_median = SimpleImputer(strategy="median")
```

Data Preprocessing continued...

- After imputation treatment the datasets were verified against missing values and none of the columns were missing any values.

```
# Checking that no column has missing values in train, validation and test sets
print(X_train.isna().sum())
print("-" * 30)
print(X_val.isna().sum())
print("-" * 30)
print(X_test.isna().sum())
```

```
department      0
region          0
education        0
gender           0
recruitment_channel  0
no_of_trainings  0
age              0
previous_year_rating  0
length_of_service  0
awards_won       0
avg_training_score  0
dtype: int64
```

```
-----
department      0
region          0
education        0
gender           0
recruitment_channel  0
no_of_trainings  0
age              0
previous_year_rating  0
length_of_service  0
awards_won       0
avg_training_score  0
dtype: int64
```

```
-----
department      0
region          0
education        0
gender           0
recruitment_channel  0
no_of_trainings  0
age              0
previous_year_rating  0
length_of_service  0
awards_won       0
avg_training_score  0
dtype: int64
```

- Outlier check (treatment if needed) : this was not needed for the datasets.

Data Preprocessing continued...

- Outlier check (treatment if needed) : this was not needed for the datasets.
- Feature engineering: all datasets were treated with SimpleImputation and then fit to the models.
- Preparation for modeling: Categorical variables were encoded.
- Shape of Models prepared from Original data, 80%, 20% and 75%, 25% split:

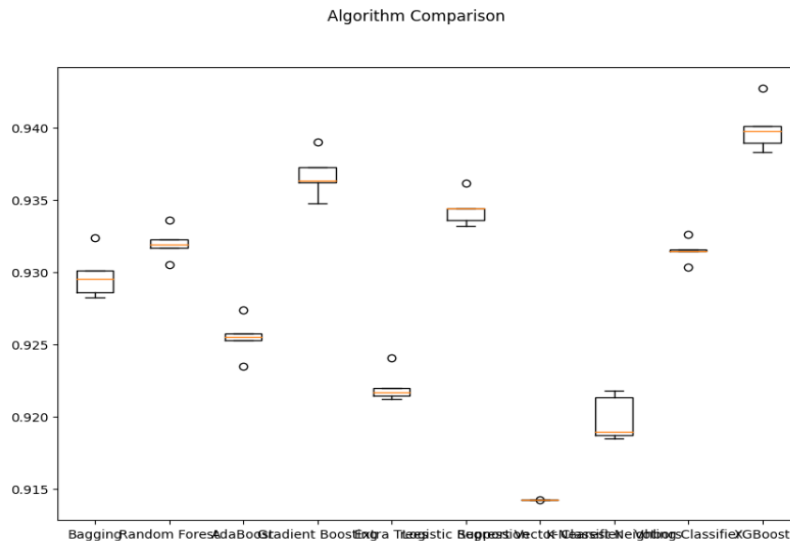
```
# Print the shapes of the resulting datasets
print(X_train.shape, X_val.shape, X_test.shape)
# Print the shapes of the resulting datasets
print("Training set shape:", X_train.shape, y_train.shape)
print("Validation set shape:", X_val.shape, y_val.shape)
print("Test set shape:", X_test.shape, y_test.shape)
```

```
➦ (43846, 53) (2741, 53) (43846, 53)
Training set shape: (43846, 53) (43846,)
Validation set shape: (2741, 53) (2741,)
Test set shape: (43846, 53) (8221,)
```

Model Performance Summary – Original sample dataset.

- Summary of performance metrics for training and validation data in tabular format for comparison for tuned models:
- Model performance summary is as follows:
- Original dataset sample. gradientBoost model performed the best.

MODEL name	Performance
Bagging	93.343%
Randomforest	93.359%
AdaBoost	92.32%
GradientBoost	96.23%
Extra Trees	90.045%
Logistics Regression	87.566%
Support vector classifier	90.562%
k-nearest neighbors	89.456%
Voting classifier	91.245%
XGBoost	93.456%



[Link to Appendix slide on model assumptions](#)

Model Performance Summary- undersampled dataset

- Summary of performance metrics for training and validation data in tabular format for comparison for tuned models:
- Model shape of Under sampled dataset is as follows:

```
↔ Before Under Sampling, counts of label 'Yes': 3760  
Before Under Sampling, counts of label 'No': 40086  
  
After Under Sampling, counts of label 'Yes': 3760  
After Under Sampling, counts of label 'No': 3760  
  
After Under Sampling, the shape of train_X: (7520, 53)  
After Under Sampling, the shape of train_y: (7520,)
```

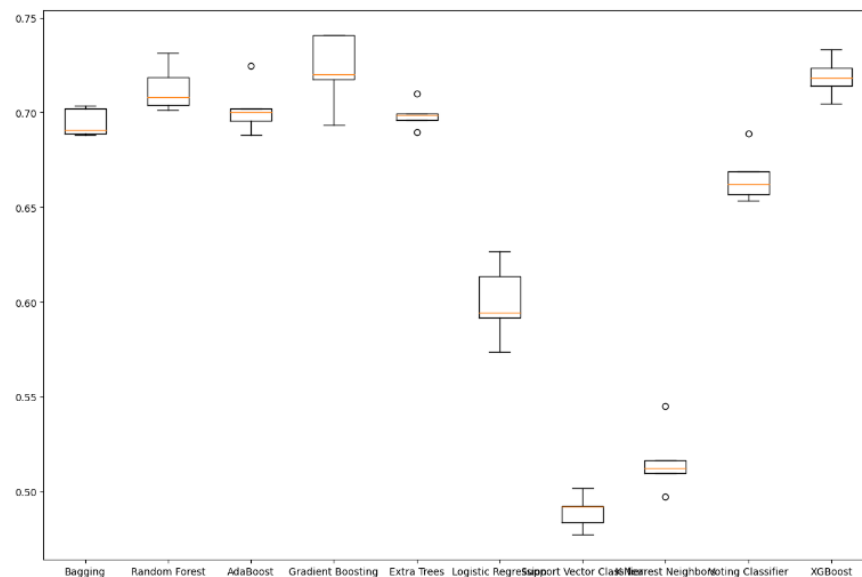
[Link to Appendix slide on model assumptions](#)

Model Performance Summary- undersampled dataset

- Summary of performance metrics for training and validation data in tabular format for comparison for tuned models:
- Model shape of Under sampled dataset is as follows

MODEL name	Performance
Bagging	69.468%
Randomforest	71.277%
AdaBoost	70.213%
GradientBoost	72.247%
Extra Trees	69.880%
Logistics Regression	60.003%
Support vector classifier	48.963%
k-nearest neighbors	51.536%
Voting classifier	66.609%
XGBoost	71.875%

Algorithm Comparison for Under sampling:



[Link to Appendix slide on model assumptions](#)

Model Performance Summary – Oversampled dataset.

- Summary of performance metrics for training and validation data in tabular format for comparison for tuned models:
- Shape of Oversampled dataset sample and Original dataset sample shapes are as follows:

```
Before Oversampling, counts of label 'Yes': 3760  
Before Oversampling, counts of label 'No': 40086
```

```
After Oversampling, counts of label 'Yes': 40086  
After Oversampling, counts of label 'No': 40086
```

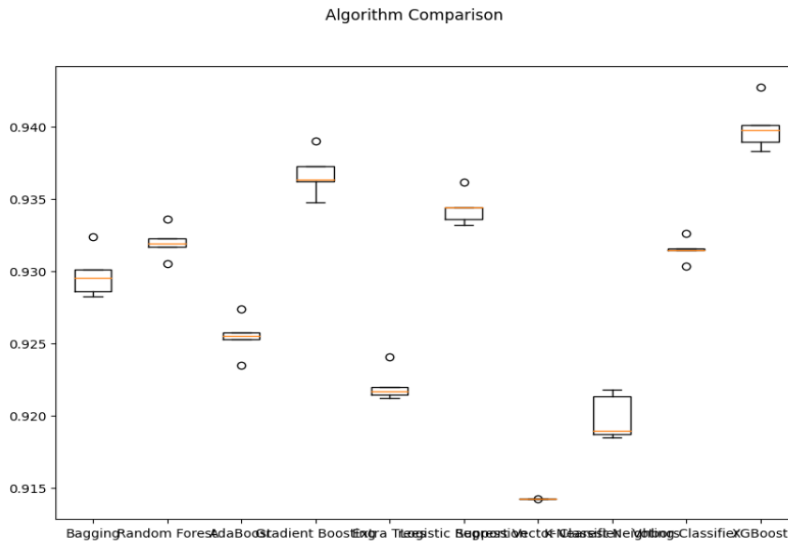
```
After Oversampling, the shape of train_X: (80172, 53)  
After Oversampling, the shape of train_y: (80172,)
```

[Link to Appendix slide on model assumptions](#)

Model Performance Summary – Oversampled dataset.

- Summary of performance metrics for training and validation data in tabular format for comparison for tuned models:
- Model performance of Oversampled dataset sample are as follows:
- GradientBoost model performed the best.

MODEL name	Performance
Bagging	75.90%
Randomforest	82.76%
AdaBoost	90.765%
GradientBoost	94.785%
Extra Trees	90.045%
Logistics Regression	87.566%
Support vector classifier	78.905%
k-nearest neighbors	83.609%
Voting classifier	72.576%
XGBoost	93.456%



[Link to Appendix slide on model assumptions](#)

Model Performance Summary – Hyper parameter tuning.

- GradientBoost model was tuned for better performance on all three datasets i.e Original dataset, over sampled dataset and Under sampled dataset using f1_scorer and RandomizedSearchCV methods.
- GradientBoost was finally adopted as the best model to help evaluate features that help better determine the employees' promotion at JMD company.

[Link to Appendix slide on model assumptions](#)

APPENDIX

Slide Header



Happy Learning !

