Employee Performance and Retention Analysis using Python

1.Project Overview

This project focuses on analysing employee performance and retention patterns within an organisation. By leveraging HR data, the goal is to identify key factors influencing employee productivity, performance, and retention, and provide actionable insights for improving workforce efficiency, reducing turnover, and fostering a positive work environment. The project will employ data analysis techniques, including exploratory data analysis (EDA), trend analysis, and visualisation to help HR managers make data-driven decisions for workforce management.

2. Dataset Summary

- Rows: 17417
- Columns: 13
- Key Features : Employee Identifier (employee_id)
  Organizational Attributes (department, region)
  Demographic Features (education, gender, age)
  Recruitment Information (recruitment_channel)
  Training & Development (avg_training_score, no_of_trainings)
  Performance Metrics (KPIs_met_more_than_80, previous_year_rating)
  Recognition & Experience (length_of_service, awards_won)
- Missing Data: 771 values in education column & 1363 in previous_year_rating column.

3. Workforce Overview

Employees are mostly in the early to mid-career range (20–40 years).

Distribution across departments and regions varies, highlighting operational concentration risks.

Recruitment channels influence performance outcomes, indicating differences in candidate quality.

4. Performance Insights

High training scores, KPI achievement, and awards strongly correlate with better performance.

Consistent performers continue to excel year-over-year.

Departments with lower performance may need targeted interventions.

5. Retention Trends

High retention: Employees aged 30–40, award winners, and consistent KPI achievers.

Retention risk: Early-career employees (20–30), low KPI achievers, and employees in certain departments with shorter tenure.

Effective training and recognition directly improve retention.

6. Exploratory Data Analysis (EDA)

We began with data preparation and cleaning in Python:

- Data Loading: Imported the dataset using pandas.

- Initial Exploration: Used df.info() to check structure and .describe() for summary statistics.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17417 entries, 0 to 17416
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   employee_id           17417 non-null  int64  
 1   department            17417 non-null  object 
 2   region                17417 non-null  object 
 3   education             16646 non-null  object 
 4   gender                17417 non-null  object 
 5   recruitment_channel   17417 non-null  object 
 6   no_of_trainings       17417 non-null  int64  
 7   age                   17417 non-null  int64  
 8   previous_year_rating  16054 non-null  float64
 9   length_of_service     17417 non-null  int64  
 10  KPIs_met_more_than_80 17417 non-null  int64  
 11  awards_won            17417 non-null  int64  
 12  avg_training_score    17417 non-null  int64  
dtypes: float64(1), int64(7), object(5)
memory usage: 1.7+ MB
```

	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met
count	17417.000000	17417	17417	16646	17417	17417	17417.000000	17417.000000	16054.000000	17417.000000	
unique	NaN	9	34	3	2	3	NaN	NaN	NaN	NaN	
top	NaN	Sales & Marketing	region_2	Bachelors	m	other	NaN	NaN	NaN	NaN	
freq	NaN	5458	3918	11519	12314	9751	NaN	NaN	NaN	NaN	
mean	39083.491129	NaN	NaN	NaN	NaN	NaN	1.250732	34.807774	3.345459	5.801860	
std	22707.024087	NaN	NaN	NaN	NaN	NaN	0.595692	7.694046	1.265386	4.175533	
min	3.000000	NaN	NaN	NaN	NaN	NaN	1.000000	20.000000	1.000000	1.000000	
25%	19281.000000	NaN	NaN	NaN	NaN	NaN	1.000000	29.000000	3.000000	3.000000	
50%	39122.000000	NaN	NaN	NaN	NaN	NaN	1.000000	33.000000	3.000000	5.000000	
75%	58838.000000	NaN	NaN	NaN	NaN	NaN	1.000000	39.000000	4.000000	7.000000	
max	78295.000000	NaN	NaN	NaN	NaN	NaN	9.000000	60.000000	5.000000	34.000000	

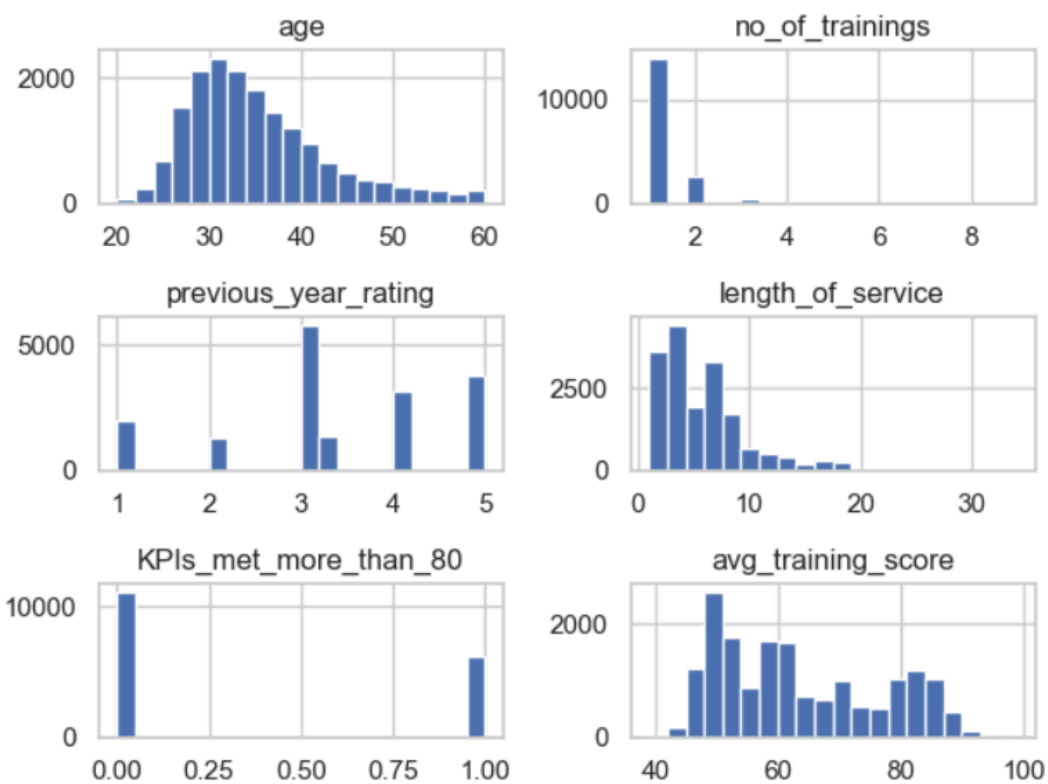
- Found the missing and duplicate values in the dataset using isnull() and duplicated() functions.

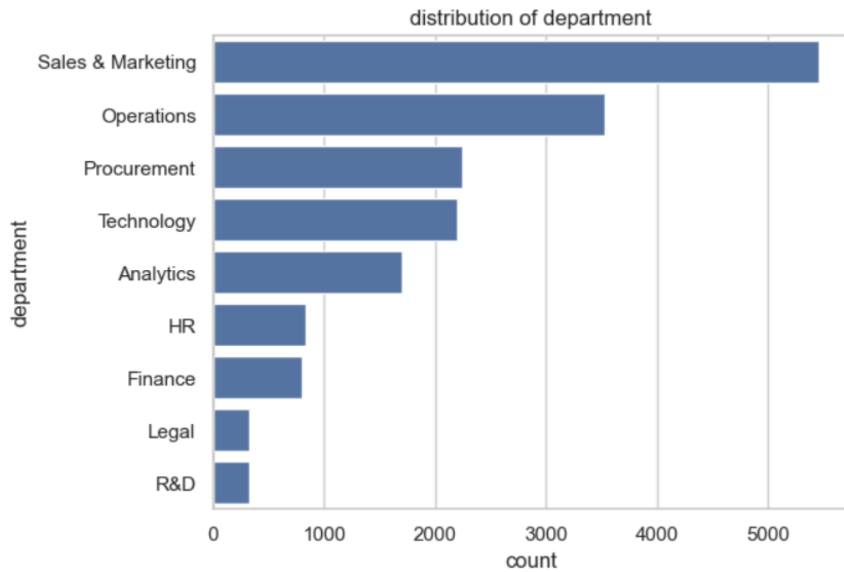
```
employee_id                0
department                 0
region                     0
education                771
gender                     0
recruitment_channel        0
no_of_trainings            0
age                        0
previous_year_rating    1363
length_of_service          0
KPIs_met_more_than_80      0
awards_won                 0
avg_training_score         0
dtype: int64
```

```
employees.duplicated().sum()
```

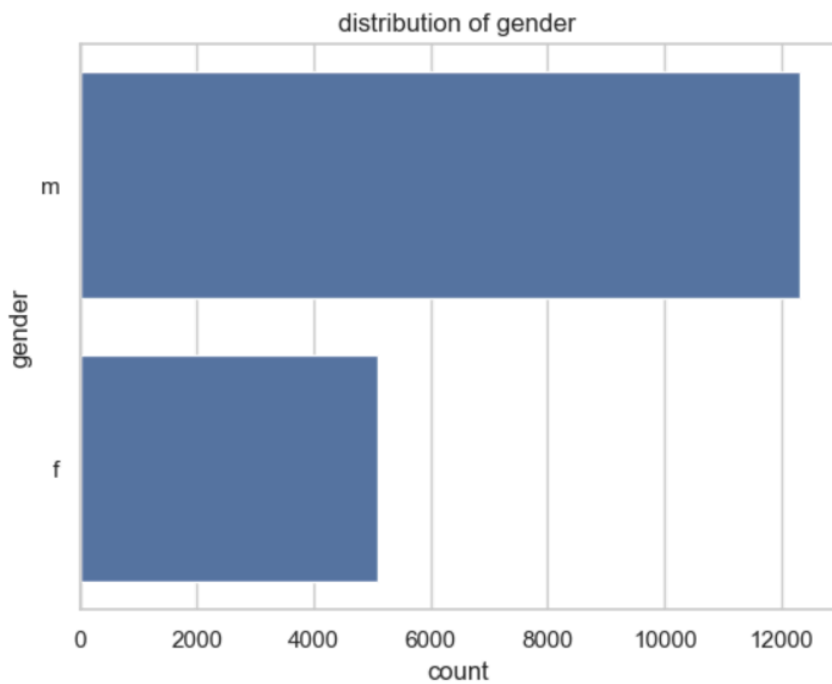
```
np.int64(2)
```

- Found the distribution of key numerical variables and categorical variables

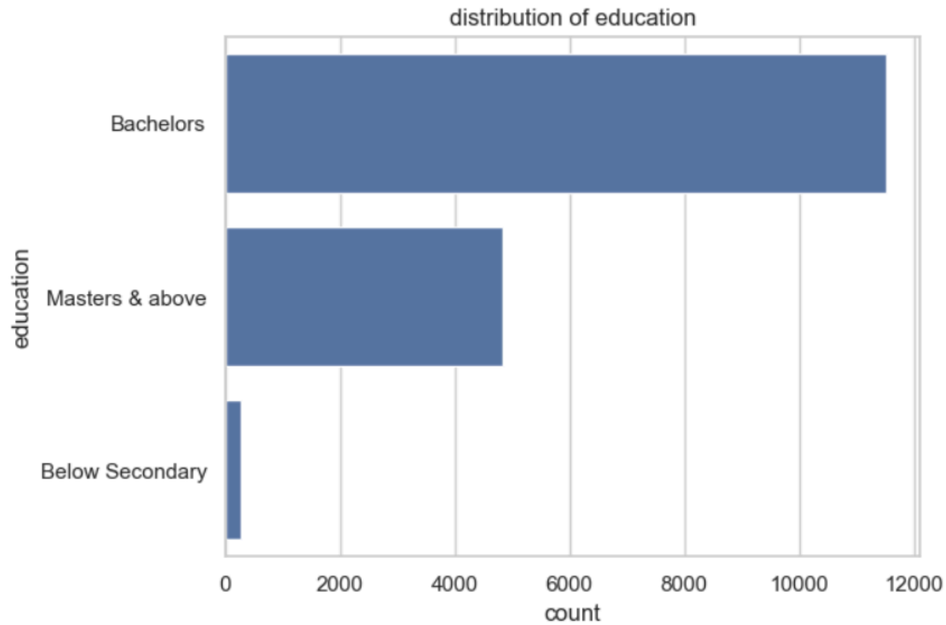




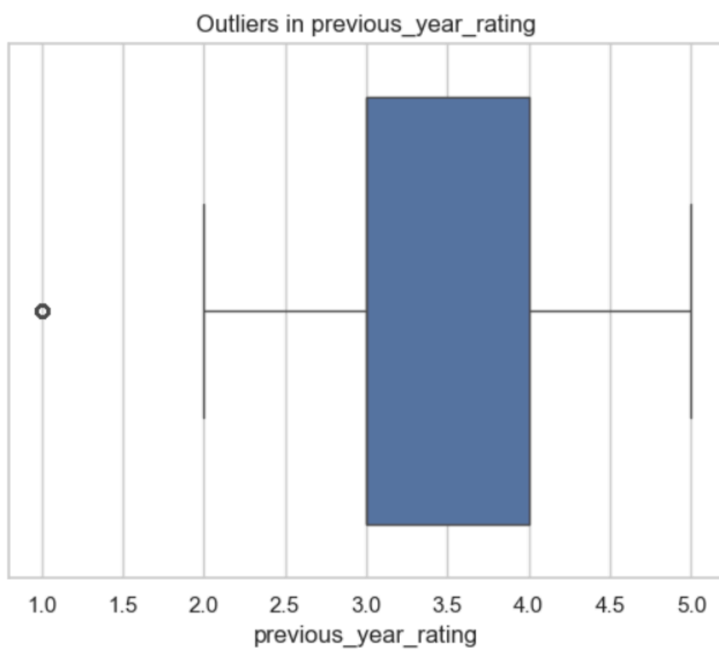

```
<Figure size 400x200 with 0 Axes>
```

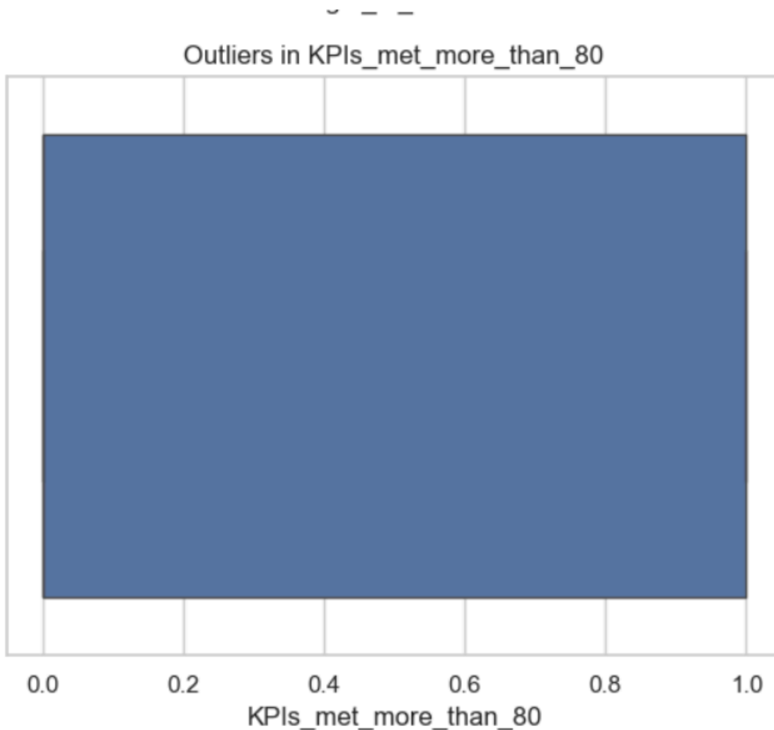



```
<Figure size 400x200 with 0 Axes>
```



- Outlier detection using seaborn library.





7. Data Preprocessing

- Handling of the missing values with fillna(), median(), & mode () functions

```
employees.isnull().sum()
```

```
employee_id             0
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
KPIs_met_more_than_80   0
awards_won              0
avg_training_score      0
dtype: int64
```

- Encoding Categorical variables and ensuring consistency in data formatting.

```python
from sklearn.preprocessing import LabelEncoder

label_encoders = {}

for col in cat_cols:
    le = LabelEncoder()
    employees[col + '_enc'] = le.fit_transform(employees[col])
    label_encoders[col] = le
```

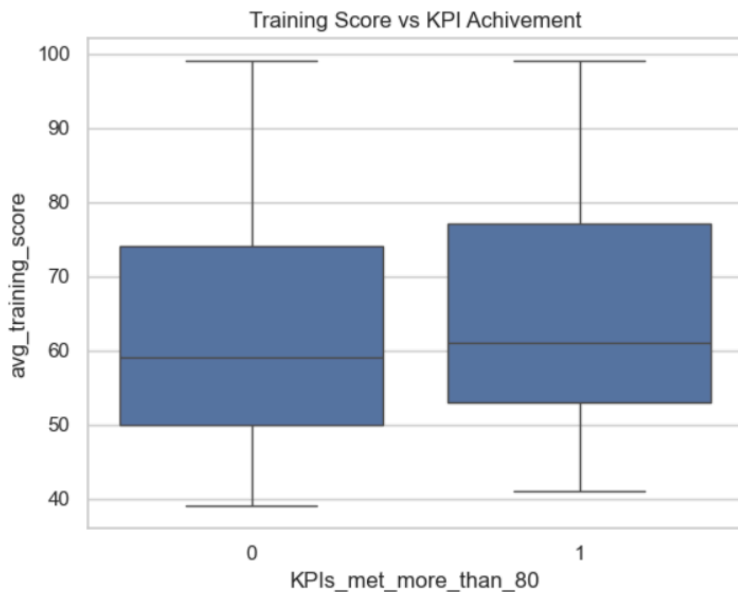
8. Key Metrics Analysis

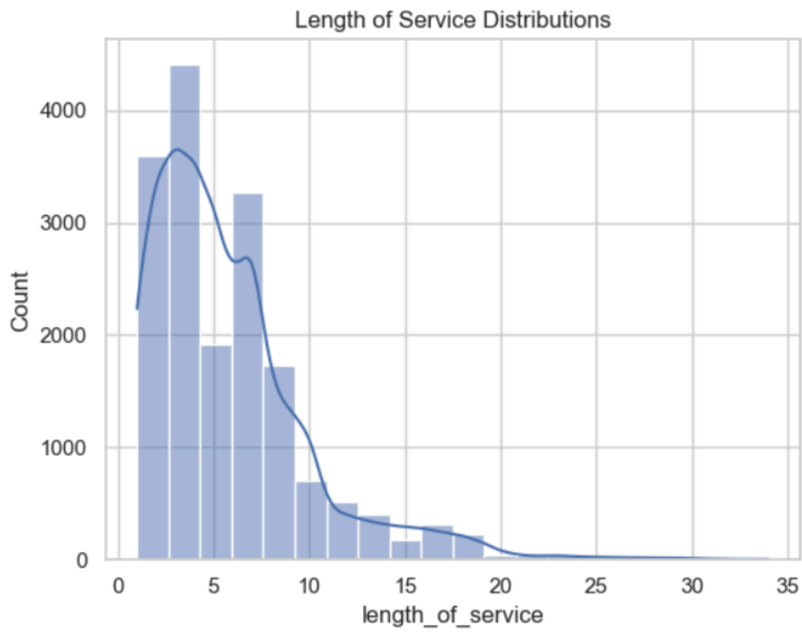
- Summarizing performance metrics by creating a performance_metrics table

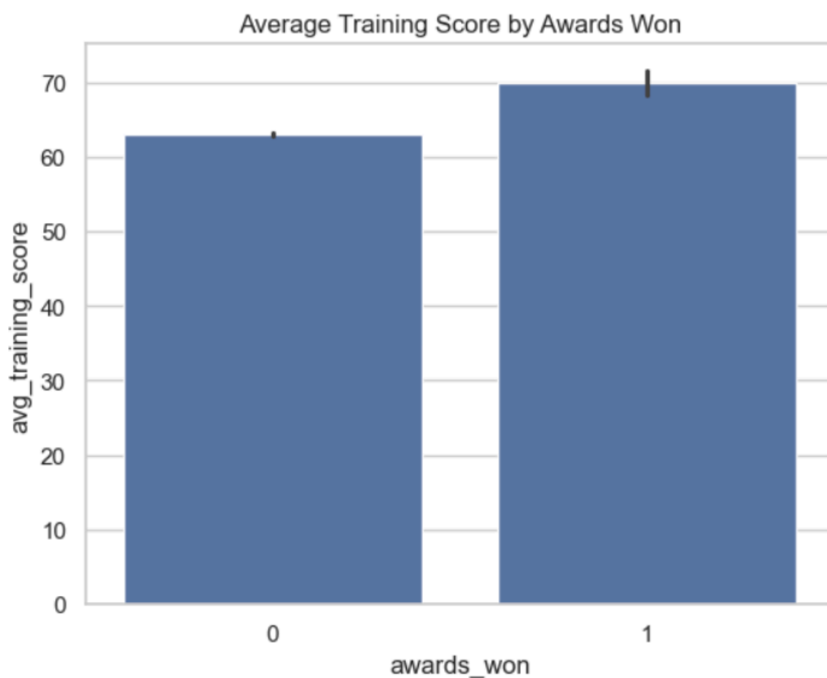
```python
performance_metrics = employees [['KPIs_met_more_than_80','previous_year_rating','avg_training_score','awards_won']]

performance_metrics.describe()
```

	KPIs_met_more_than_80	previous_year_rating	avg_training_score	awards_won
count	17417.000000	17417.000000	17417.000000	17417.000000
mean	0.358845	3.345459	63.176322	0.023368
std	0.479675	1.214862	13.418179	0.151074
min	0.000000	1.000000	39.000000	0.000000
25%	0.000000	3.000000	51.000000	0.000000
50%	0.000000	3.000000	60.000000	0.000000
75%	1.000000	4.000000	75.000000	0.000000
max	1.000000	5.000000	99.000000	1.000000

- Analysing key metrics such as KPI Achievements vs performance, awards impact on performance,length of service analysis

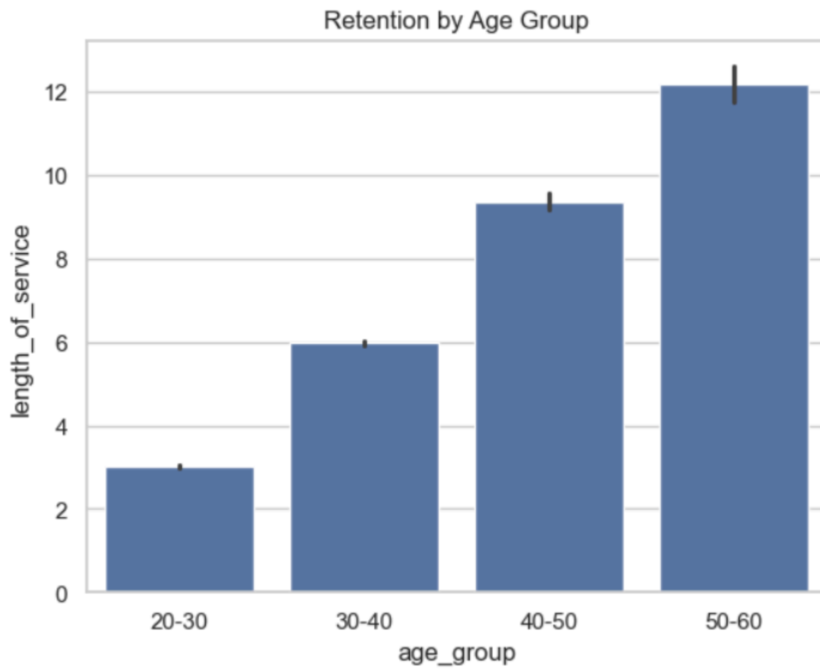

Training Score vs KPI Achivement
100
90
80
70
60
50
40
avg_training_score
0
1
KPIs_met_more_than_80



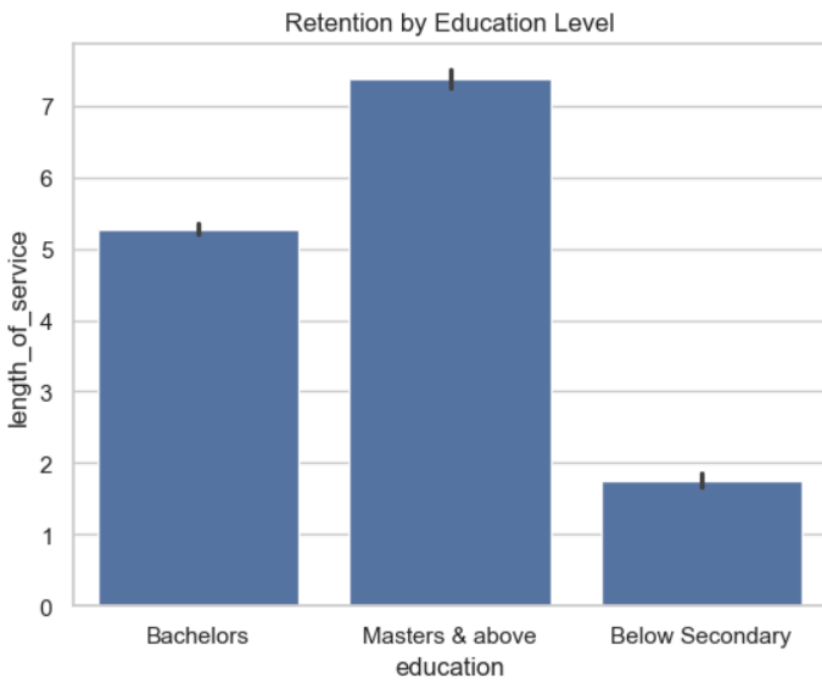


9. Retention Trends Analysis

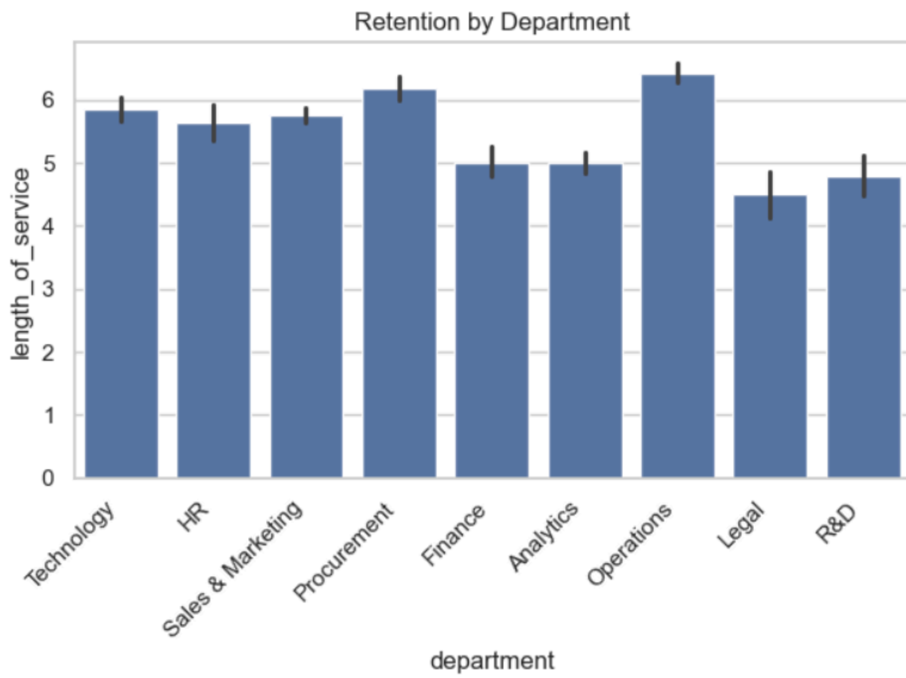
- Trend analysis assuming longer length of service = higher retention
- Analysing retention by age group, education, department, training impact on retention.



Retention by Age Group
length_of_service
0
2
4
6
8
10
12
20-30
30-40
40-50
50-60
age_group



Retention by Education Level
length_of_service
0
1
2
3
4
5
6
7
Bachelors
Masters & above
Below Secondary
education



10. Predictive Insights & Actionable Recommendation.

- Key Insights

Employees with higher training scores and KPI achievement show longer retention

Award-winning employees consistently outperform peers

Mid-career age groups (30–40) show the highest retention

Certain departments have shorter service lengths, indicating engagement gaps

Employees with lower previous year ratings are more likely to exit early

- HR recommendation

1. Increase targeted training programs for low-performing departments.

2. Introduce recognition programs to improve motivation and retention.

3. Focus retention strategies on early-career employees (20–30 age group).

4. Use KPI performance as an early indicator for engagement interventions.

5. Invest in continuous learning for employees with high potential but low ratings.

Summary

Conducted EDA, preprocessing, and metric analysis

Identified key drivers of performance and retention

Visualized departmental, demographic, and training trends

Provided clear, interpretable insights for HR leadership

Business Impact

Improved retention strategy alignment

Better performance management decisions

Data-driven workforce planning

Conclusion

Retention and performance are closely linked. By investing in skill development, recognition, and proactive performance management, the organization can enhance engagement, retain talent, and drive sustainable growth.