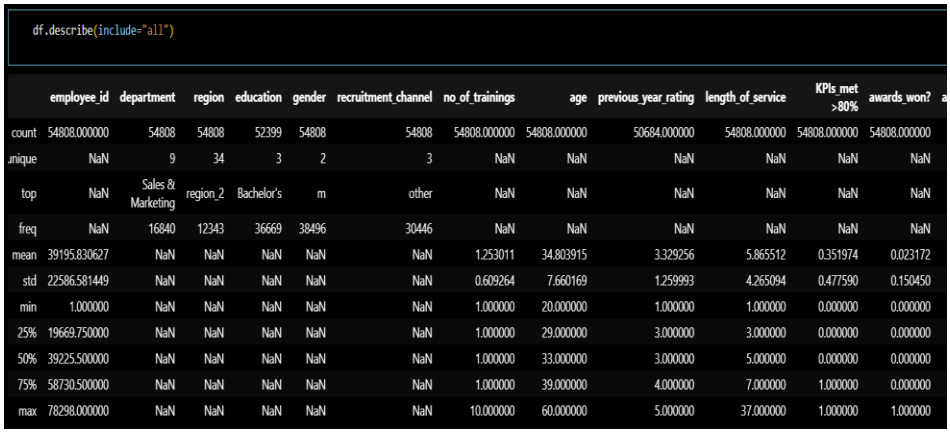
Data Collection and Preprocessing Phase

Date	9 July 2024
Team ID	SWTID1720104839
Project Title	**Human Resource Management: Predicting Employee Promotions Using Machine Learning**
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

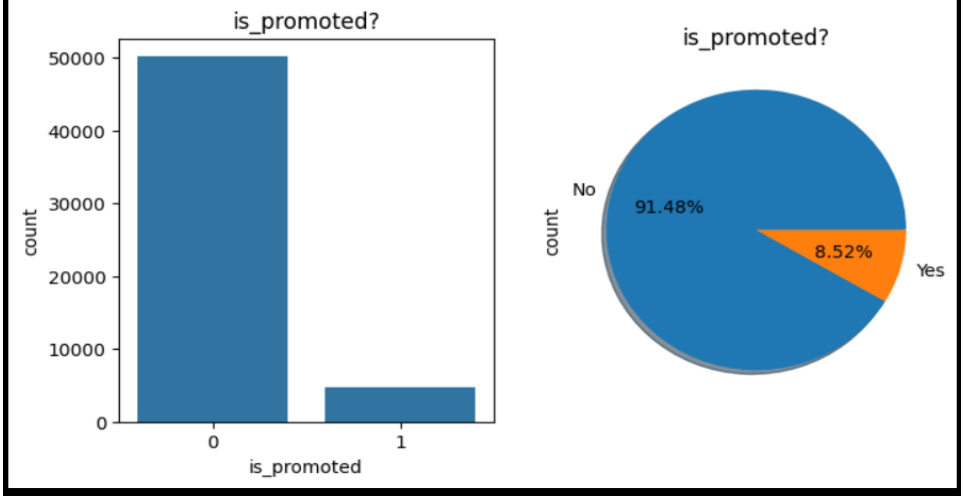
Section	Description
Data Overview	

```
df.describe(include="all")
```

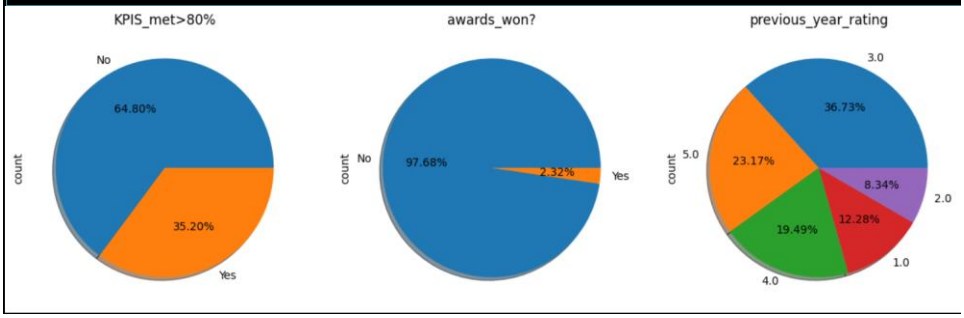
	employee_id	department	region	education	gender	recruitment_channel	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	a
count	54808.000000	54808	54808	52399	54808	54808	54808.000000	54808.000000	50684.000000	54808.000000	54808.000000	54808.000000	
unique	NaN	9	34	3	2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

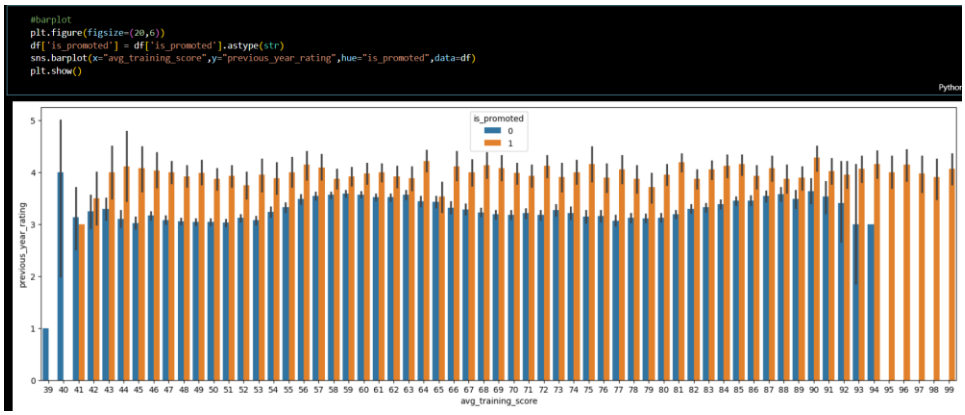
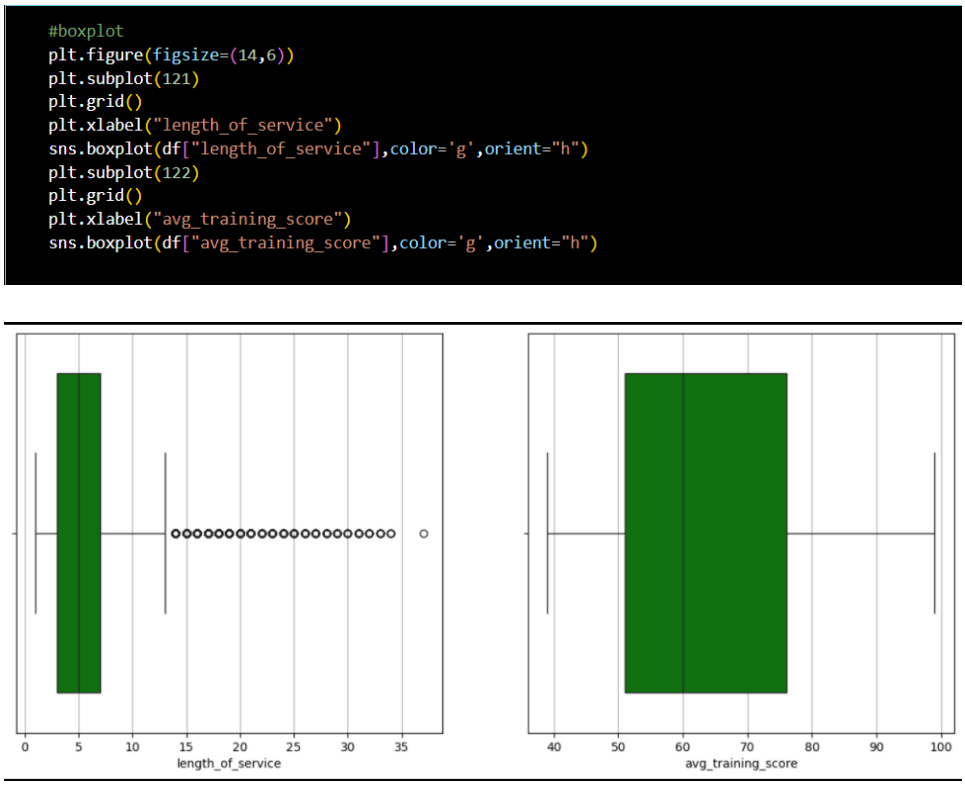
Univariate Analysis

```
plt.figure(figsize=(8,4))
plt.subplot(121)
#countplot
sns.countplot(x="is_promoted",data=df)
plt.title("is_promoted?")
plt.subplot(122)
#pieplot
plt.title("is_promoted?")
val_count=df["is_promoted"].value_counts()
val_count.plot(kind="pie",autopct='%.2f%%',shadow=True,labels=["No","Yes"])
plt.show()
```

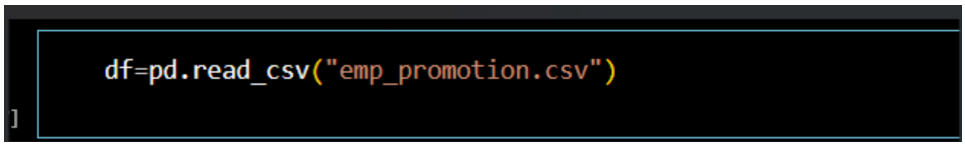



```
plt.figure(figsize=(15,15))
plt.subplot(131)
plt.title("KPIS_met>80%")
df["KPIs_met >80%"].value_counts().plot(kind="pie",shadow=True,autopct="%.2f%%",labels=["No","Yes"])
plt.subplot(132)
plt.title("awards_won?")
df["awards_won?"].value_counts().plot(kind="pie",shadow=True,autopct="%.2f%%",labels=["No","Yes"])
plt.subplot(133)
plt.title("previous_year_rating")
df["previous_year_rating"].value_counts().plot(kind="pie",shadow=True,autopct="%.2f%%")
plt.show()
```



Multivariate Analysis	
Outliers and Anomalies	

Data Preprocessing Code Screenshots

Loading Data	

<table>
<tr>
<td>Handling Missing Data</td>
<td>

Drop unwanted features

+ Code + Markdown

```
df=df.drop(["employee_id","gender","region","recruitment_channel"],axis=1)
```

```
df.head()
```

	department	education	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_score	is_promoted
0	Sales & Marketing	Master's & above	1	35	5.0	8	1	0	49	0
1	Operations	Bachelor's	1	30	5.0	4	0	0	60	0
2	Sales & Marketing	Bachelor's	1	34	3.0	7	0	0	50	0
3	Sales & Marketing	Bachelor's	2	39	1.0	10	0	0	50	0
4	Technology	Bachelor's	1	45	3.0	2	0	0	73	0

Checking for null values

```
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
```

```
#handling null values
df["education"]=df["education"].fillna(df["education"].mode()[0])
df["previous_year_rating"]=df["previous_year_rating"].fillna(df["previous_year_rating"].mode()[0])
```

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

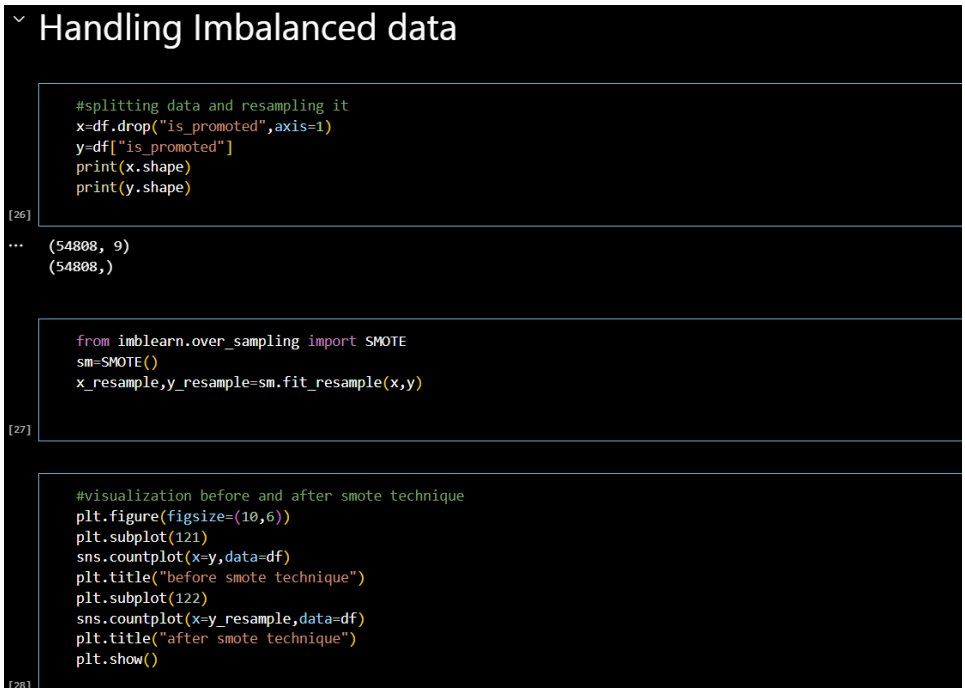
```
department            0
education             0
no_of_trainings       0
age                   0
previous_year_rating  0
length_of_service     0
KPIs_met >80%         0
awards_won?           0
avg_training_score    0
is_promoted           0
dtype: int64
```

</td>
</tr>
<tr>
<td>Data Transformation</td>
<td>

```
le=LabelEncoder()
df["department"]=le.fit_transform(df["department"])
df["education"]=le.fit_transform(df["education"])
df.head()
```

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

</td>
</tr>
</table>

Handling imbalanced data	
Spiltting data