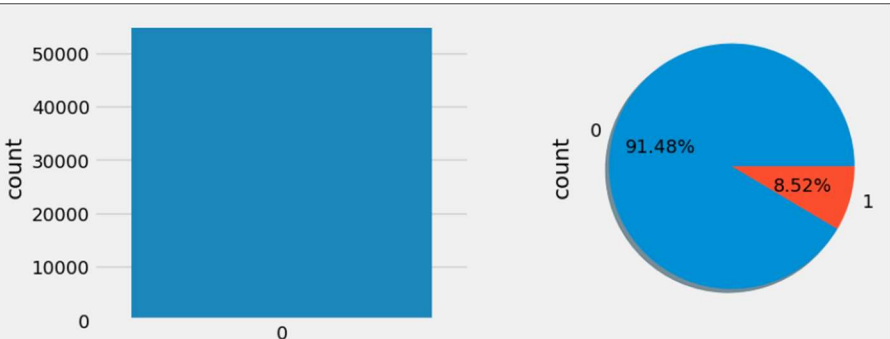


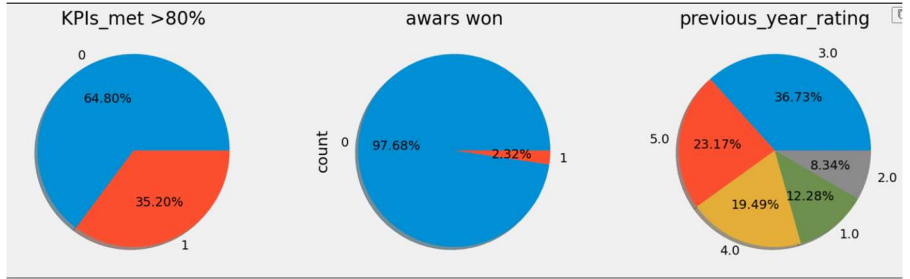
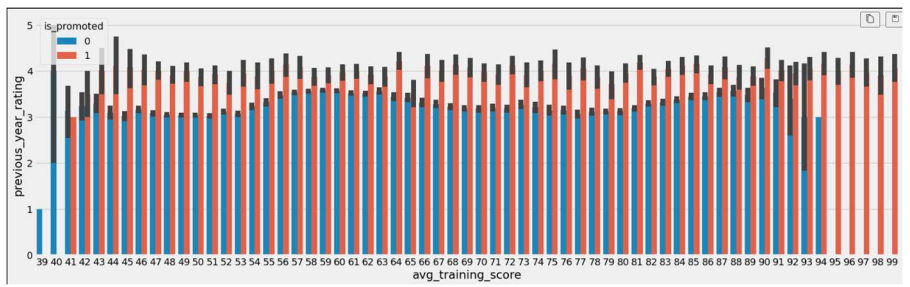
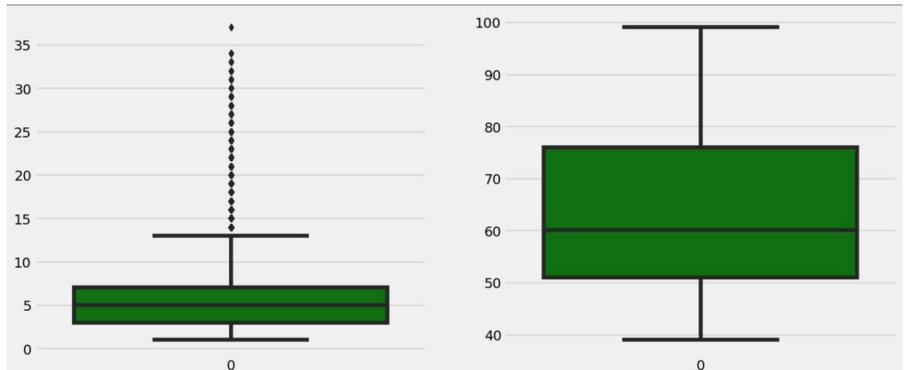
Data Collection and Preprocessing Phase

Date	9 July 2024
Team ID	XXXXXX
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	<pre>df.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 10 yrs_in_current_role 54808 non-null int64 11 awards_won? 54808 non-null int64 12 avg_training_score 54808 non-null int64 13 is_promoted 54808 non-null int64 dtypes: float64(1), int64(8), object(5) memory usage: 5.9+ MB</pre>
Univariate Analysis	 <p>The figure displays univariate analysis results. On the left, a bar chart shows the count for category 0 is approximately 50,000. On the right, a pie chart shows the distribution: 91.48% for category 0 (blue) and 8.52% for category 1 (red).</p>

	
Multivariate Analysis	
Outliers and Anomalies	
Data Preprocessing Code Screenshots	
Loading Data	<pre>#Load data df=pd.read_csv("../Human Resource Management Predicting Employee Promotions Using Machine Learning/5.Project Executable Files/Dataset/emp_promotion.csv") ✓ 0.1s Python print('Shape of the data = {}'.format(df.shape)) ✓ 0.0s Python Shape of the data = (54089, 14)</pre>
Handling Missing Data	<pre>#Replacing Nan with Mode print(df['education'].value_counts()) df['education']=df['education'].fillna(df['education'].mode()[0]) print(df['previous_year_rating'].value_counts()) df['previous_year_rating']=df['previous_year_rating'].fillna(df['previous_year_rating'].mode()[0])</pre>
Data Transformation	<pre>#Handling Categorical Values df['education']=df['education'].replace(("Below Secondary","Bachelor's","Master's & above"),(1,2,3)) lb=LabelEncoder() df['department']=lb.fit_transform(df['department'])</pre>