Analize the salary distribution of employees based on various factors and visualize the relationship between years of service and salary.

Name: Tarun Singh

Roll No: 202401100400198

_____

Introduction

Salary distribution analysis is a critical task in HR analytics. Understanding how different factors such as experience, job role, and department impact salaries can help organizations ensure fair compensation and improve employee satisfaction. In this report, we analyze salary trends using a dataset and visualize patterns between employees' years of service and their salary. The goal is to identify key insights that could be useful for decision-making in an organization.

_____

Methodology

1. Data Collection: We used an employee salary dataset containing job roles, years of experience, and salary information.

2. Data Preprocessing:

o Handled missing values.

o Converted categorical data (e.g., job roles) into numerical values using encoding techniques.

o Checked for anomalies and outliers in salary distribution.

3. Data Analysis & Visualization:

o Used Pandas for data manipulation.

o Plotted heatmaps and bar charts using Matplotlib and Seaborn to understand salary trends.

o Identified correlations between years of service and salary.

CODE

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Step 1: Create a sample dataset for employee salary analysis
data = {
    'Employee_ID': range(1, 21),
    'Position': ['Junior Developer', 'Senior Developer', 'Lead Developer', 'Junior Developer', 'Senior Developer',
                 'Lead Developer', 'Junior Developer', 'Senior Developer', 'Lead Developer', 'Junior Developer',
                 'Junior Developer', 'Senior Developer', 'Lead Developer', 'Junior Developer', 'Senior Developer',
                 'Lead Developer', 'Junior Developer', 'Senior Developer', 'Lead Developer', 'Junior Developer'],
    'Department': ['IT', 'IT', 'IT', 'HR', 'HR', 'HR', 'Sales', 'Sales', 'Sales', 'Finance',
                   'Finance', 'Finance', 'Marketing', 'Marketing', 'Marketing', 'Operations', 'Operations', 'Operations', 'Admin', 'Admin'],
    'Salary': [50000, 80000, 120000, 55000, 85000, 130000, 57000, 87000, 135000, 59000,
               51000, 81000, 125000, 53000, 82000, 128000, 55000, 84000, 140000, 58000],
    'Years_of_Service': [2, 5, 8, 3, 6, 9, 2, 5, 8, 3, 4, 7, 10, 2, 6, 9, 3, 7, 10, 4]
}

# Convert the data into a pandas DataFrame
employee_data = pd.DataFrame(data)

# Step 2: Data Preprocessing

# Handle missing values (if any)
employee_data.dropna(subset=['Salary'], inplace=True)
```

```python
employee_data.dropna(subset=['Salary'], inplace=True)

# Encode categorical columns ('Position' and 'Department') using one-hot encoding
employee_data = pd.get_dummies(employee_data, columns=['Position', 'Department'])

# Normalize salary using MinMaxScaler
scaler = MinMaxScaler()
employee_data['Salary'] = scaler.fit_transform(employee_data[['Salary']])

# Step 3: Exploratory Data Analysis (EDA)

# Descriptive statistics for salary
salary_stats = employee_data['Salary'].describe()
print("Descriptive Statistics for Salary:")
print(salary_stats)

# Average salary by position (before scaling)
avg_salary_by_position = pd.DataFrame(data).groupby('Position')['Salary'].mean()
print("\nAverage Salary by Position (Before Scaling):")
print(avg_salary_by_position)

# Visualizing the salary distribution by position using a boxplot
plt.figure(figsize=(10, 5))
sns.boxplot(x='Position', y='Salary', data=pd.DataFrame(data))
plt.title('Salary Distribution by Position')
plt.xticks(rotation=45)
plt.show()

# Violin plot to visualize the salary distribution
plt.figure(figsize=(10, 5))
sns.violinplot(x='Position', y='Salary', data=pd.DataFrame(data))
```

```
# Train linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
r_squared = model.score(X_test, y_test)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

print("\nModel Performance Metrics:")
print("R-squared:", r_squared)
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
```

```
Descriptive Statistics for Salary:
count    20.000000
mean      0.397222
std       0.357431
min       0.000000
25%       0.072222
50%       0.350000
75%       0.791667
max       1.000000
Name: Salary, dtype: float64

Average Salary by Position (Before Scaling):
Position
Junior Developer     54750.000000
Lead Developer      129666.666667
```

Output/Result

1. Salary Distribution Graph: The histogram visualizes how salaries are spread across different employees.

2. Correlation Heatmap: The heatmap highlights the correlation between years of service and salary, providing insights into career progression trends.

_____

References/Credits

- Python Libraries Used: Pandas, Matplotlib, Seaborn
- Guidance from AI MSE Course Materials

_____

Conclusion

The analysis provided insights into salary distribution and its relationship with years of service. The results indicate that experience generally plays a significant role in determining salary levels. Such studies help organizations in structuring compensation strategies effectively.