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:
    - Handled missing values.
    - Converted categorical data (e.g., job roles) into numerical values using encoding techniques.
    - Checked for anomalies and outliers in salary distribution.
3. Data Analysis & Visualization:
    - Used Pandas for data manipulation.
    - Plotted heatmaps and bar charts using Matplotlib and Seaborn to understand salary trends.
    - Identified correlations between years of service and salary.

CODE

```
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)

# 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))
plt.title('Salary Distribution by Position (Violin Plot)')
plt.xticks(rotation=45)
plt.show()

# Step 4: Correlation Analysis

# Correlation matrix for numerical variables (Salary and Years of Service)
correlation_matrix = employee_data[['Salary', 'Years_of_Service']].corr()
```

```
print("\nCorrelation Matrix between Salary and Years of Service:")
print(correlation_matrix)

# Heatmap for correlation matrix
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Step 5: Linear Regression to predict Salary based on Years of Service and Position

# Define features (X) and target (y)
X = employee_data.drop(columns=['Salary', 'Employee_ID'])  # Drop target and ID column
y = employee_data['Salary']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 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)
```

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.