PROJECT

Report on

EMPLOYEE SALARY ANALYSIS

by

Udit Ranjan (202410116100229)
Suchita Singh (202410116100212)
Shyam Sundar (2024101161100207)
Shweta Patel (202410116100206)

Session:2024-2025 (II Semester)

Under the supervision of

Ms. Komal Salgotra
Assistant Professor

KIET Group of Institutions, Delhi-NCR, Ghaziabad

Department Of Computer Applications
KIET GROUP OF INSTITUTIONS, DELHI-NCR,
GHAZIABAD-201206
(2024- 2025)

TABLE OF CONTENTS

CHAPTER - 01

INTRODUCTION

Salaries play a pivotal role in employee satisfaction, retention, and overall organizational success. **Employee Salary Analysis** provides an in-depth exploration of salary distributions, identifying key factors that influence compensation across different departments and experience levels. This analysis helps organizations ensure fair pay practices, detect inconsistencies, and make data-driven decisions regarding employee compensation.

Key Objectives:

- **Understanding Salary Trends:** Analyze how salaries vary across different age groups, experience levels, and departments.
- **Data Cleaning & Preprocessing:** Handle missing values, detect and manage outliers, and encode categorical data to ensure accurate analysis.
- **Statistical Insights:** Compute salary distributions, measure central tendencies (mean, median, and standard deviation), and identify anomalies.
- **Visual Representation:** Use histograms, boxplots, and scatter plots to illustrate salary trends and disparities effectively.
- **Hypothesis Testing:** Compare salaries across departments to determine if significant differences exist using statistical tests.
- **Predictive Modeling:** Leverage linear regression to predict salaries based on experience, providing valuable insights into compensation growth over time.

This analysis not only highlights salary disparities but also aids in designing fair and competitive pay structures. By understanding the correlation between experience and salary, HR professionals and decision-makers can implement strategies to attract and retain top talent while ensuring transparency and equity in compensation policies.

With a blend of data analytics and visualization, **Employee Salary Analysis** transforms raw salary data into actionable insights, paving the way for informed and fair decision-making in workforce management.

CHAPTER - 02

METHODOLOGY

The **Employee Salary Analysis** follows a structured approach, integrating data preprocessing, statistical analysis, and machine learning techniques to extract meaningful insights. Below is a step-by-step breakdown of the methodology used:

1. Data Collection & Exploration

- Load the dataset from a CSV file.
- Inspect the dataset structure using df.info() and df.head() to understand the data types and key attributes.
- Identify missing values, outliers, and potential inconsistencies.

2. Data Preprocessing & Cleaning

- **Handling Missing Data:**
    - Replace missing values in numerical fields (e.g., salary and experience) with the median to maintain data integrity.
    - Drop records with missing categorical values to prevent inconsistencies in analysis.
- **Detecting & Handling Outliers:**
    - Use the **Interquartile Range (IQR)** method to identify and handle salary outliers.
    - Outliers are either removed or adjusted to maintain data quality.
- **Encoding Categorical Data:**
    - Convert categorical variables (e.g., department) into numerical values using **Label Encoding** for further analysis.

3. Exploratory Data Analysis (EDA)

- **Descriptive Statistics:**
    - Compute key statistics such as mean, median, standard deviation, and salary distribution percentiles.

- **Visualizing Salary Trends:**
  - **Histogram & KDE Plot:** Display salary distribution to understand common salary ranges.
  - **Boxplot Analysis:** Identify salary variations across different departments.
  - **Scatter Plot:** Explore the relationship between **experience and salary** to identify trends.

4. Hypothesis Testing

- Conduct a **T-test** to determine if there is a significant salary difference between two departments.
- If the **p-value < 0.05**, it indicates a significant difference in salaries; otherwise, no significant difference is found.

5. Correlation Analysis

- Compute the **correlation matrix** using a heatmap to analyze the relationship between numerical variables such as experience, age, and salary.
- Identify which factors have the strongest impact on salaries.

6. Predictive Modeling (Linear Regression)

- **Objective:** Predict employee salary based on years of experience.
- **Steps:**
  1. Split the dataset into **training (80%)** and **testing (20%)** subsets.
  2. Train a **Linear Regression Model** using experience as the independent variable and salary as the dependent variable.
  3. Predict salaries on the test dataset and evaluate model performance.
  4. Visualize the regression line to see how well experience predicts salary.

7. Interpretation & Decision-Making

- Summarize key findings from statistical tests, correlation analysis, and predictive modeling.
- Provide insights into salary trends, fairness across departments, and salary growth over time.
- Offer recommendations for HR professionals and management to optimize salary structures based on data-driven insights.

CHAPTER - 03

CODE

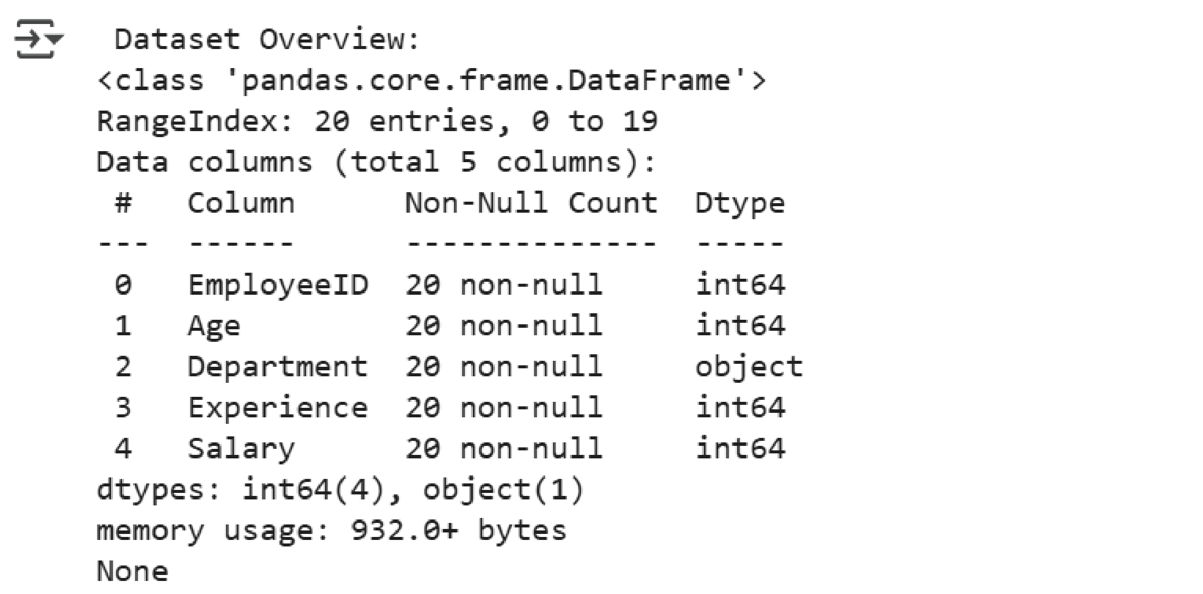
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import ttest_ind, pearsonr
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder

# Load the dataset
file_path = '/content/employee_data.csv'
df = pd.read_csv(file_path)

# Display basic info
print(" Dataset Overview:")
print(df.info())
print("\n First 5 Records:")
print(df.head())

# Handle missing values intelligently
df.loc[:, 'Salary'] = df['Salary'].fillna(df['Salary'].median())# Fill missing salaries with median
df.loc[:, 'Experience'] = df['Experience'].fillna(df['Experience'].median())# Fill missing experience
df.dropna(inplace=True)  # Drop rows where categorical fields are missing
```

```
# Detect Outliers using IQR method
Q1 = df['Salary'].quantile(0.25)
Q3 = df['Salary'].quantile(0.75)
IQR = Q3 - Q1
outliers = df[(df['Salary'] < (Q1 - 1.5 * IQR)) | (df['Salary'] > (Q3 + 1.5 * IQR))]
print(f"\n Outliers detected: {len(outliers)}")

# Salary Statistics
print("\n Salary Statistics:")
print(df['Salary'].describe())

# Salary Distribution Visualization
plt.figure(figsize=(8,5))
sns.histplot(df['Salary'], bins=20, kde=True, color='blue')
plt.title('Salary Distribution')
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.show()

# Boxplot for Outlier Detection
plt.figure(figsize=(6,4))
sns.boxplot(x=df['Salary'])
plt.title("Boxplot of Salaries (Outliers Detection)")
plt.show()

# Salary by Department
if 'Department' in df.columns:
    plt.figure(figsize=(10,5))
    sns.boxplot(x='Department', y='Salary', data=df)
    plt.xticks(rotation=45)
    plt.title('Salary Distribution by Department')
    plt.show()

# Salary by Experience
if 'Experience' in df.columns:
    plt.figure(figsize=(8,5))
    sns.scatterplot(x='Experience', y='Salary', data=df)
    plt.title('Salary vs Experience')
    plt.xlabel('Years of Experience')
    plt.ylabel('Salary')
    plt.show()

# Correlation Analysis (Exclude non-numeric columns)
numeric_df = df.select_dtypes(include=['number'])

if not numeric_df.empty:
    plt.figure(figsize=(6, 4))
    sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
    plt.title('Correlation Matrix')
    plt.show()
else:
    print("\nNo numeric columns found for correlation analysis.")

# Encode Categorical Data for Further Analysis
if 'Department' in df.columns:
    label_encoder = LabelEncoder()
    df['Department_encoded'] = label_encoder.fit_transform(df['Department'])
```

```
# Hypothesis Testing - Salary Difference Between Two Departments
if 'Department' in df.columns:
    unique_departments = df['Department'].unique()
    if len(unique_departments) >= 2:
        dept1 = df[df['Department'] == unique_departments[0]]['Salary']
        dept2 = df[df['Department'] == unique_departments[1]]['Salary']
        t_stat, p_value = ttest_ind(dept1, dept2, equal_var=False)
        print(f"\n Hypothesis Testing: Salary Difference between {unique_departments[0]} and {unique_departments[1]}")
        print(f"T-Statistic: {t_stat:.4f}, P-Value: {p_value:.4f}")
        if p_value < 0.05:
            print(" Significant Difference Found in Salaries.")
        else:
            print(" No Significant Difference Found in Salaries.")

# Regression Analysis: Predict Salary based on Experience
if 'Experience' in df.columns:
    X = df[['Experience']]
    y = df['Salary']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = LinearRegression()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Visualize Regression Line
    plt.figure(figsize=(8,5))
    plt.scatter(X_test, y_test, color='blue', label='Actual Salaries')
    plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted Salaries')
    plt.xlabel('Years of Experience')
    plt.ylabel('Salary')
    plt.title('Experience vs Salary (Regression Analysis)')
    plt.legend()
    plt.show()

    print(f"\n Regression Model: Predicting Salary from Experience")
    print(f"Intercept: {model.intercept_:.2f}")
    print(f"Coefficient (Experience Impact): {model.coef_[0]:.2f}")
```

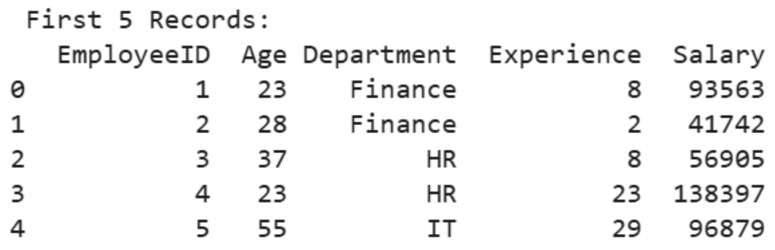
CHAPTER - 04

OUTCOMES



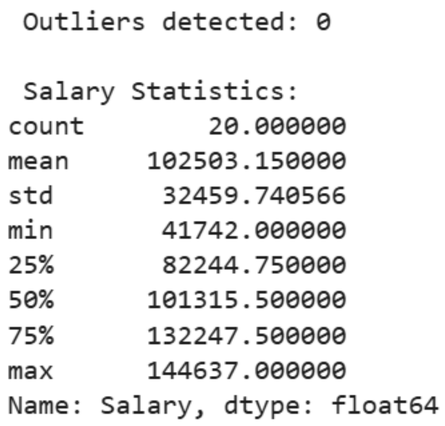
```
 Dataset Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   EmployeeID  20 non-null     int64 
 1   Age         20 non-null     int64 
 2   Department  20 non-null     object
 3   Experience  20 non-null     int64 
 4   Salary      20 non-null     int64 
dtypes: int64(4), object(1)
memory usage: 932.0+ bytes
None
```

Fig. 4.1



```
 First 5 Records:
   EmployeeID  Age Department  Experience  Salary
0           1   23    Finance           8   93563
1           2   28    Finance           2   41742
2           3   37         HR           8   56905
3           4   23         HR          23  138397
4           5   55         IT          29   96879
```

Fig. 4.2



```
 Outliers detected: 0

 Salary Statistics:
count        20.000000
mean     102503.150000
std       32459.740566
min       41742.000000
25%       82244.750000
50%      101315.500000
75%      132247.500000
max      144637.000000
Name: Salary, dtype: float64
```

Fig. 4.3

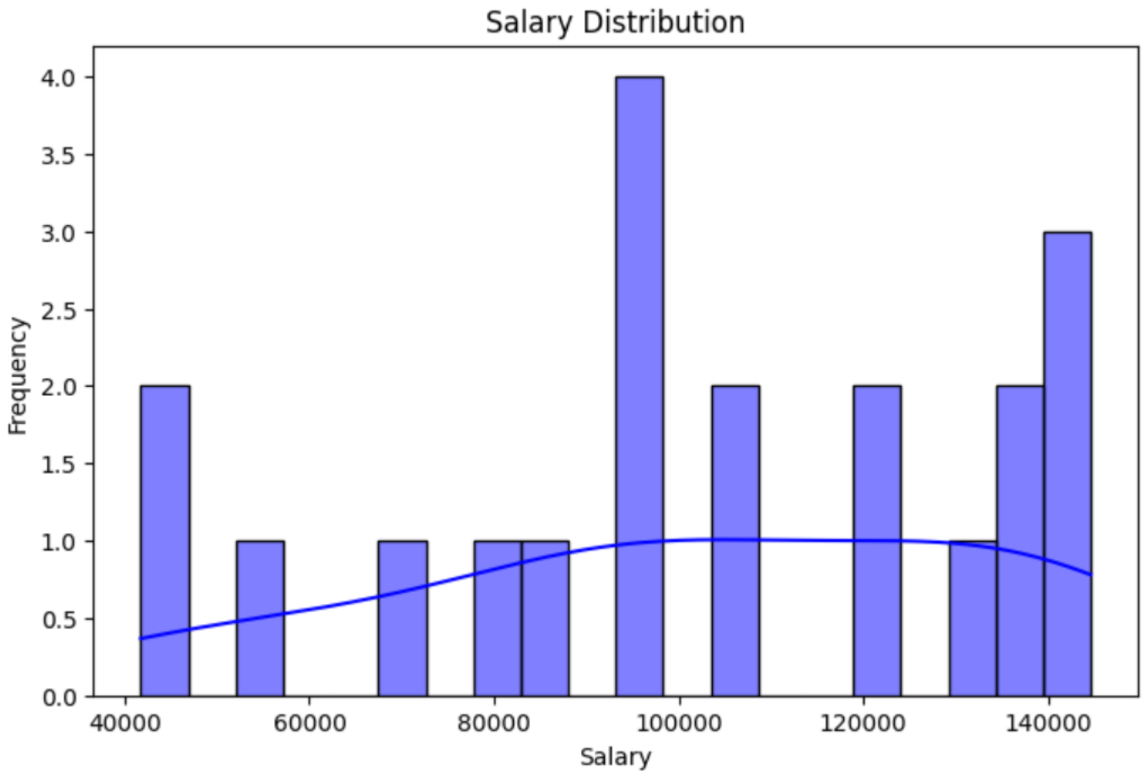


Fig. 4.4

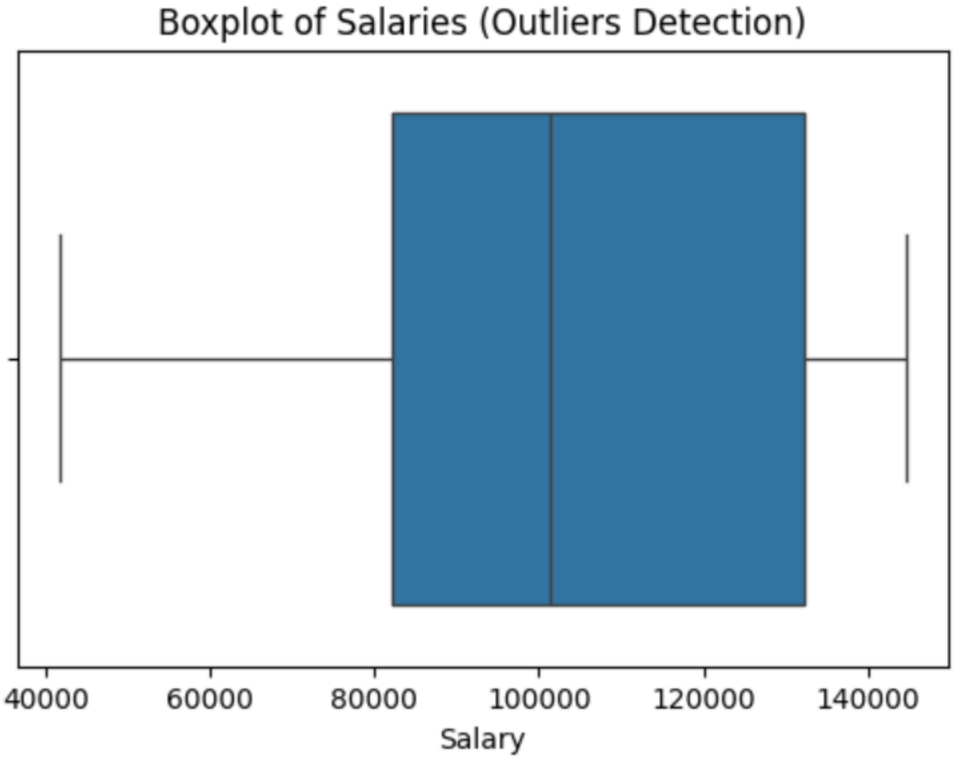

Boxplot of Salaries (Outliers Detection)
40000
60000
80000
100000
120000
140000
Salary

Fig. 4.5

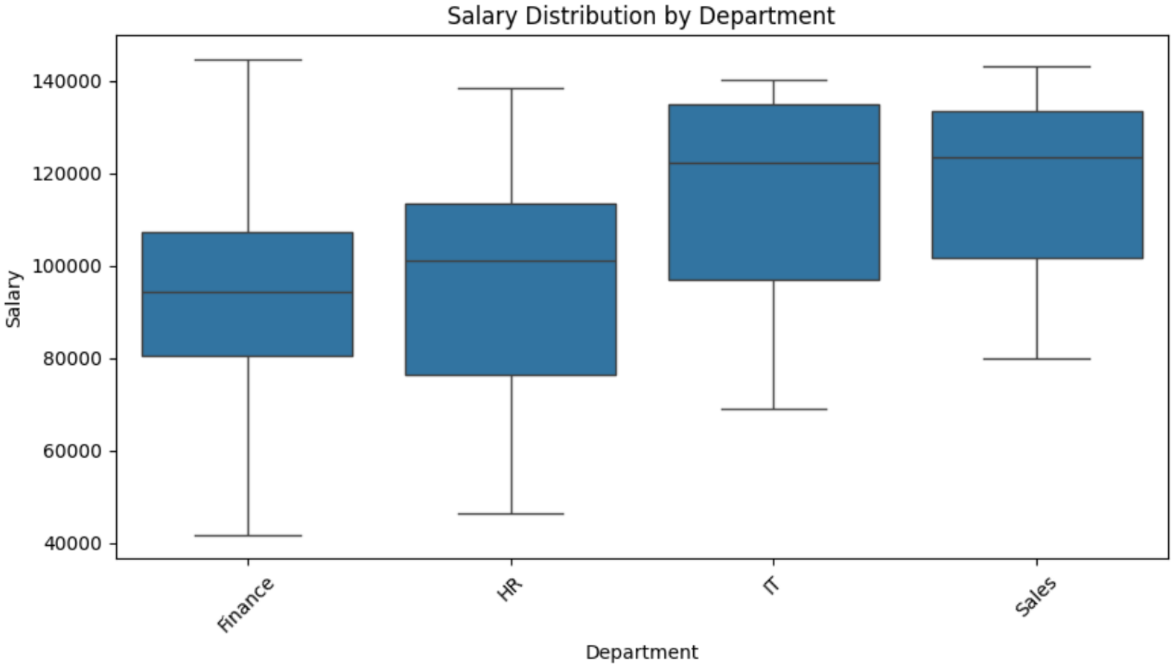

Salary Distribution by Department
140000
120000
100000
80000
60000
40000
Salary
Finance
HR
IT
Sales
Department

Fig. 4.6

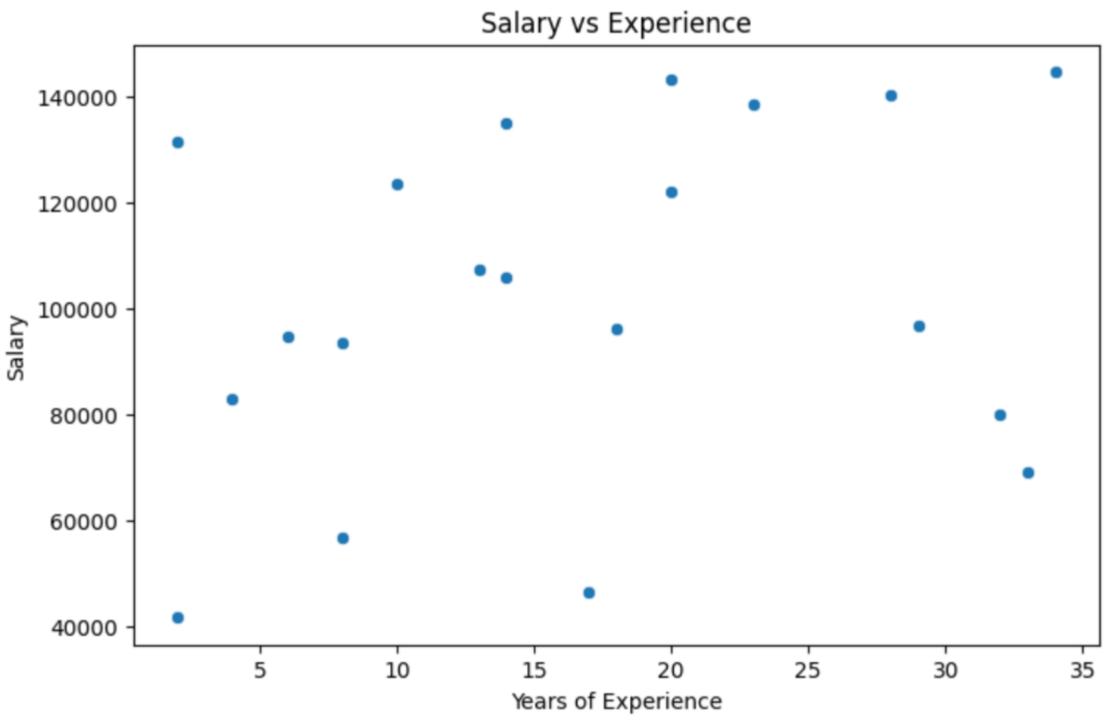

Salary vs Experience
Salary
140000
120000
100000
80000
60000
40000
5
10
15
20
25
30
35
Years of Experience

Fig. 4.7

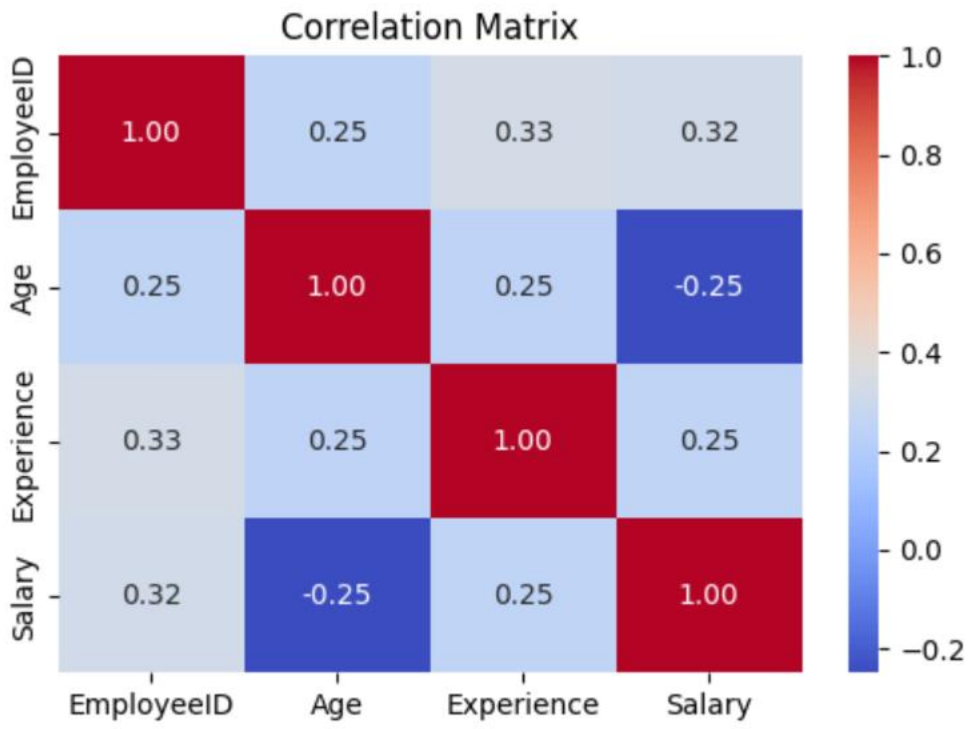
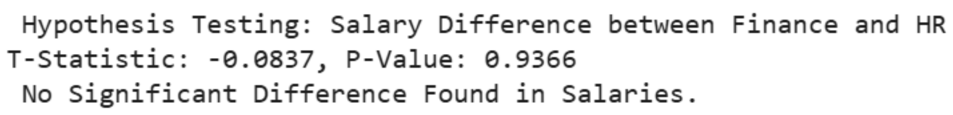

Correlation Matrix
EmployeeID
Age
Experience
Salary
1.00
0.25
0.33
0.32
0.25
1.00
0.25
-0.25
0.33
0.25
1.00
0.25
0.32
-0.25
0.25
1.00
EmployeeID
Age
Experience
Salary
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Fig. 4.8



```
 Hypothesis Testing: Salary Difference between Finance and HR
T-Statistic: -0.0837, P-Value: 0.9366
 No Significant Difference Found in Salaries.
```

Fig. 4.9

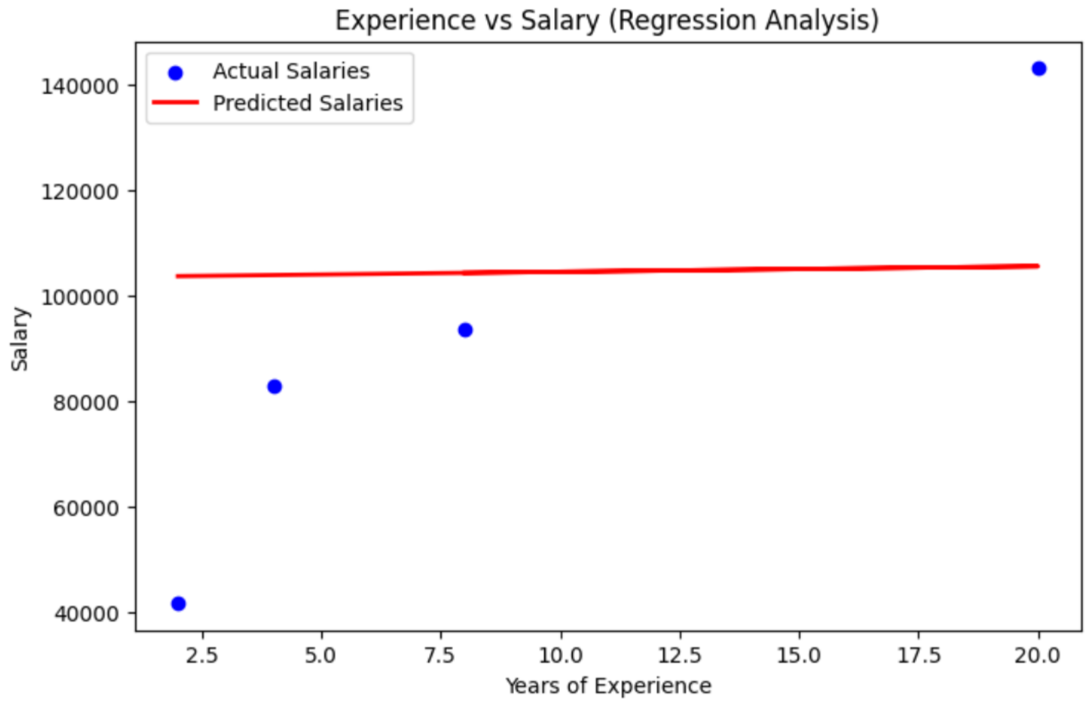
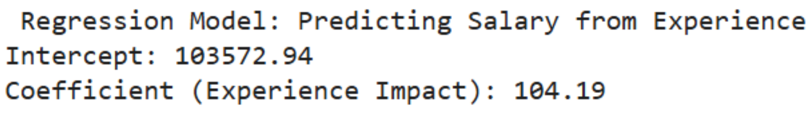


Fig. 4.10



```
 Regression Model: Predicting Salary from Experience
Intercept: 103572.94
Coefficient (Experience Impact): 104.19
```

Fig. 4.11