Employee Salary Analysis

A PROJECT REPORT
for
AI Project(AI101B)
Session (2024-25)

Submitted by

GAURAV VISHWAKARMA
(202410116100076)
KUMAR KASHYAP
(202410116100106)
KUNAL SINGH
(202410116100109)
LAVISH NEHRA
(202410116100111)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Mr. Apoorv Jain
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

TABLE OF CONTENTS

1. INTRODUCTION …….. 2

2. METHODOLOGY ……..….. 3

1. Approach
2. Algorithm Used

3. CODE …….……... 5

4. OUTPUTS SCREENSHOTS……..…….9

5. OUTPUT EXPLANATION ……..…….11

Introduction

Overview

Employee salary analysis plays a crucial role in workforce management, financial planning, and organizational decision-making. Companies aim to offer competitive salaries to attract and retain talent while ensuring financial sustainability. Traditional salary determination methods rely on experience, market trends, and subjective assessments. However, with the advancement of technology, machine learning models provide a more data-driven and accurate approach to salary prediction.

Importance of Salary Analysis

Employee salary analysis is essential for:

- **Ensuring Fair Compensation**: Helps companies set competitive wages based on industry standards.
- **Retaining Talent**: Providing appropriate salaries increases job satisfaction and reduces turnover.
- **Budget Allocation**: Assists in planning company expenses efficiently.
- **Performance Evaluation**: Helps in structuring salary increments based on experience and performance.

Scope of the Project

- **Data Collection and Cleaning**: Gathering a structured dataset and removing inconsistencies.
- **Feature Engineering**: Selecting the most relevant variables affecting salary.
- **Machine Learning Model**: Implementing a regression-based predictive model.
- **Performance Evaluation**: Assessing the model's accuracy using statistical metrics.
- **User Input Predictions**: Enabling real-time salary predictions based on user-provided details.

This project bridges the gap between traditional salary estimation methods and modern data-driven decision-making.

Methodology

A. Approach

The project follows a structured approach to ensure effective data processing and accurate salary predictions. The key steps involved are:

1. **Data Collection**: The dataset is sourced from a structured CSV file containing relevant employee salary data.
2. **Data Preprocessing**: This step involves handling missing values, converting categorical data into numerical format, and removing irrelevant features.
3. **Feature Selection**: We identify the most influential variables that contribute to salary determination. In this project, we use **Experience (years), Age, and Gender** as independent features.
4. **Data Splitting**: The dataset is split into **training (80%) and testing (20%)** subsets to ensure the model learns effectively.
5. **Model Training**: The **Linear Regression** model is trained using the training dataset.
6. **Model Evaluation**: The trained model is tested against unseen data, and performance metrics such as **Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared score (R^2)** are computed.
7. **Salary Prediction for Users**: The model allows users to input their experience, age, and gender to get a predicted salary estimate.
8. **Visualization**: Scatter plots, histograms, and regression lines are used to understand the correlation between salary and various factors.

B. Algorithm Used

Linear Regression

Linear Regression is a fundamental supervised learning algorithm used to establish a relationship between dependent and independent variables. The equation for multiple linear regression is:

Where:

- **b0** is the intercept
- **b1, b2, and b3** are coefficients learned during training
- **Experience, Age, and Gender** are input features used for prediction

The model minimizes the difference between actual and predicted salaries using **Least Squares Estimation**, making it an effective tool for salary analysis. It calculates the best-fit line that minimizes the sum of squared residuals between actual and predicted salary values. The evaluation metrics used include:

- **Mean Absolute Error (MAE)**: Measures the average absolute differences between predicted and actual salaries.
- **Mean Squared Error (MSE)**: Evaluates the squared differences between actual and predicted values, penalizing larger errors.
- **R-squared Score (R^2)**: Determines the proportion of salary variations explained by the independent variables.

Linear regression is chosen for this analysis due to its simplicity, interpretability, and efficiency in handling structured numerical data.

Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load dataset
df = pd.read_csv('/content/Employee_Salary_Dataset.csv')

# Display first few rows to understand the dataset structure
print("First few rows of the dataset:")
display(df.head())

# Check for missing values in each column
print("\nChecking for missing values in dataset:")
print(df.isnull().sum())

# Remove rows with missing values to avoid errors in model training
df.dropna(inplace=True)

# Remove the 'ID' column since it does not contribute to salary prediction
if 'ID' in df.columns:
    df.drop(columns=['ID'], inplace=True)
    print("\n'ID' column removed as it is not needed for prediction.")
```

```
# Convert 'Gender' into numerical format (Male = 0, Female = 1) for machine learning processing
df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})
print("\nGender column encoded (Male = 0, Female = 1)")

# Define the independent variables (features) and the dependent variable (target)
X = df[['Experience_Years', 'Age', 'Gender']]  # Features used for prediction
y = df['Salary']  # Target variable (Salary)

# Splitting dataset into 80% training data and 20% testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print("\nDataset split into training (80%) and testing (20%) sets.")

# Initialize the linear regression model
model = LinearRegression()
print("\nTraining the Linear Regression model...")
model.fit(X_train, y_train)
print("Model training completed.")

# Predict salaries on the test data
y_pred = model.predict(X_test)

# Evaluate the model performance
print("\nModel Evaluation:")
print("- Mean Absolute Error (MAE):", round(mean_absolute_error(y_test, y_pred), 2))  # Measures average absolute error in predictions
print("- Mean Squared Error (MSE):", round(mean_squared_error(y_test, y_pred), 2))  # Measures average squared difference between actual
print("- R² Score (Performance Indicator):", round(r2_score(y_test, y_pred), 4))  # Indicates how well the model explains the variation in target

# User input for salary prediction
```

```
print("\nEnter details to predict salary:")
experience = float(input("- Years of Experience: "))  # Take experience as input
age = float(input("- Age: "))  # Take age as input
gender = input("- Gender (Male/Female): ").strip()  # Take gender as input
gender = 0 if gender.lower() == 'male' else 1  # Convert gender into numerical format

# Convert user input into a dataframe with proper column names to match training data
user_data = pd.DataFrame([[experience, age, gender]], columns=X.columns)

# Predict salary based on user input
your_salary = model.predict(user_data)
print("\nPredicted Salary: ₹", format(round(your_salary[0], 2), ','))  # Display predicted salary with formatting
```

Outputs

First few rows of the dataset:

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000
3	4	2	22	Male	25000
4	5	1	17	Male	10000

```
Checking for missing values in dataset:
ID                  0
Experience_Years    0
Age                 0
Gender              0
Salary              0
dtype: int64

'ID' column removed as it is not needed for prediction.

Gender column encoded (Male = 0, Female = 1)

Dataset split into training (80%) and testing (20%) sets.

Training the Linear Regression model...
Model training completed.
```

```
Model Evaluation:
- Mean Absolute Error (MAE): 1158453.2
- Mean Squared Error (MSE): 2658843406389.61
- R² Score (Performance Indicator): -185.8912

Enter details to predict salary:
- Years of Experience: 5
- Age: 26
- Gender (Male/Female): Female

Predicted Salary: ₹ 547,792.33
```

OUTPUT EXPLANATION

The output of the model includes:

- **Dataset Overview**: Displays the first few rows of the dataset, showing information about employees, including their experience, age, gender, and salary.
- **Missing Value Check**: Identifies any missing values and removes null entries to ensure data consistency.
- **Feature Encoding**: Converts categorical variables like gender into numerical format for machine learning processing.
- **Model Training**: The Linear Regression model is trained using 80% of the dataset, learning the relationships between experience, age, gender, and salary.
- **Model Evaluation**: Displays performance metrics such as:
- **Mean Absolute Error (MAE)**: Represents the average deviation of predicted salaries from actual salaries.
- **Mean Squared Error (MSE)**: Highlights the squared differences between actual and predicted values, emphasizing larger errors.
- **R^2 Score**: Indicates how well the independent variables explain salary variations.
- **Salary Prediction**: Allows users to input their details (experience, age, gender) and receive a predicted salary estimate.
- **Visualization of Results**: Graphs such as scatter plots and histograms illustrate the relationship between salary and key variables.

The model's results confirm that **experience and age positively correlate with salary**, while gender influences salary trends. These insights help companies implement data-driven salary structures to ensure fair compensation and financial planning.