

# EMPLOYEE SALARY PREDICTION APP

## (A Machine Learning-Based Web Application)

**Presented By :**

Shagun

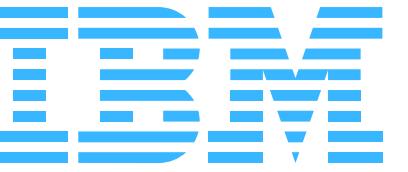
Roll no. : 2400330100325

B.Tech-CSE

Raj Kumar Goel Institute of Technology, AKTU

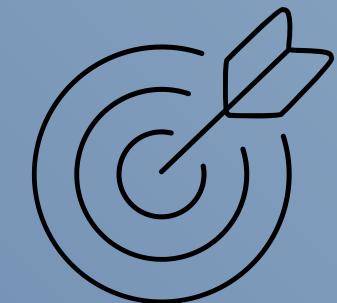
**Under the Guidance of :**

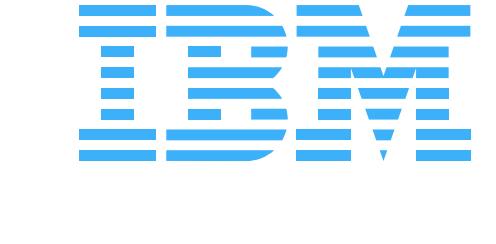
Ms. Naina Devi



# INTRODUCTION

- Employee salary depends on factors like experience , education , job role etc. 
- This project demonstrates prediction using Linear Regression. 
- Machine Learning helps in predicting salaries more precisely. 
- Traditional estimation is slow and not always accurate.





# OBJECTIVE



- To minimize manual effort and reduce errors in salary estimation
- To analyze the relationship between employee experience and salary using Linear Regression.
- To improve prediction accuracy by using structured data 

# SCOPE

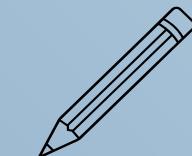
- The project focuses on predicting employee salaries based on multiple factors such as experience, education level, and job role.
- Uses Machine Learning (Linear Regression) for salary prediction.
- Helps organizations analyze pay structure and make data-driven salary decisions.
- Can be extended in the future using more algorithms (like Decision Tree, XGBoost) or additional features (like location, company size, etc.).



# LINEAR REGRESSION MODEL

- It is a statistical model that finds the best-fitting straight line (called a regression line) through a set of data points to predict future values.

**Mathematical Equation :**  $Y = mX + C$



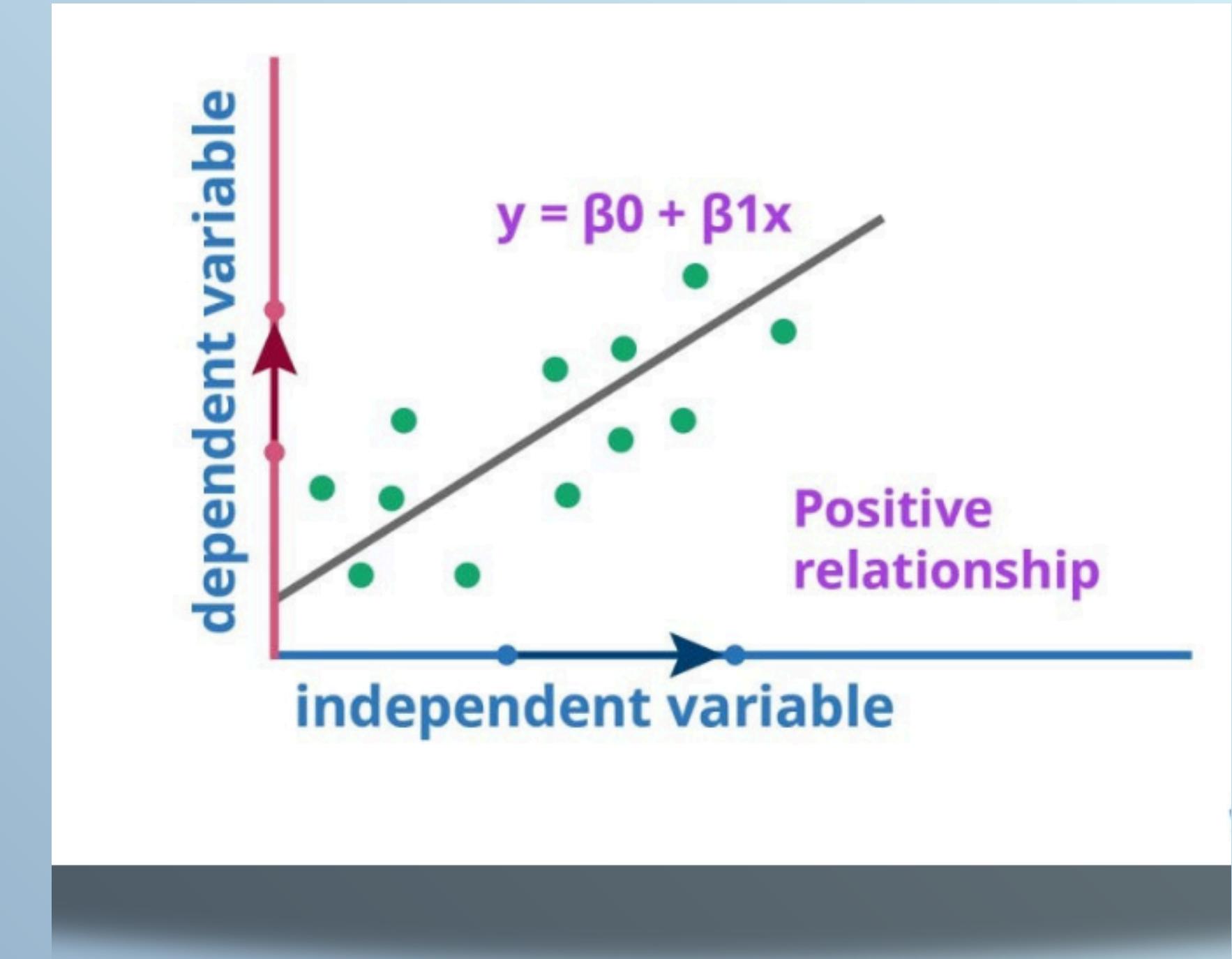
Where:

$Y \rightarrow$  Predicted value (e.g., Salary)

$X \rightarrow$  Input variable (e.g., Experience)

$m \rightarrow$  Slope of the line (shows how much  $Y$  changes when  $X$  changes)

$c \rightarrow$  Intercept (value of  $Y$  when  $X = 0$ )



# METHODOLOGY

Data Collection

Data Preprocessing

Model Building

Model Evaluation

Deployment

Import Dataset using Pandas

Convert categorial data into numeric using Label Encoding

Split dataset using `train_test_split()`

Store trained model and preprocessing tools as .pkl files

Using metrices ensure that model gives accurate prediction

Built a Streamlit (app.py) for real time salary prediction



# Software and Hardware Requirements



## Software

- Python 
- Libraries : pandas , numpy , xgboost  
scikit-learn , matplotlib , seaborn
- Development Tools :  
Jupyter Notebook / VS Code 
- Framework : Streamlit
- Dataset : salary\_Data.csv 

## Hardware

- Laptop/PC with 4GB+ RAM 
- Stable Internet connection 

# OUTPUT:

## 💡 Description :

- The application takes user inputs such as Age, Gender, Education Level, Job Title, and Years of Experience.
- On clicking the “Predict Salary” button, the model predicts the expected salary based on the trained dataset.
- The model uses Linear Regression to estimate salary values accurately.
- The output is displayed instantly on the screen with the predicted amount.

🧠 This demonstrates the working of the Employee Salary Prediction App.

The screenshot shows the "Employee Salary Prediction App" interface. At the top, there's a purple briefcase icon followed by the app name. Below that is a section titled "Employee Details" with the following fields:

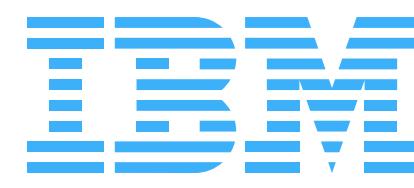
- Age: 35 (with minus and plus buttons to adjust)
- Gender: Female (with a dropdown arrow)
- Education Level: Master's (with a dropdown arrow)
- Job Title: Sales Executive (with a dropdown arrow)
- Years of Experience: 6.50 (with minus and plus buttons to adjust)

At the bottom of the form is a red-bordered "Predict Salary" button with a person icon. Below the button, a green bar displays the predicted salary: "Predicted Salary: ₹83,506.60" with a rupee sign icon.

# CONCLUSION

- The project successfully predicts employee salaries using Linear Regression.
- It demonstrates how Machine Learning can automate salary estimation accurately.
- The model helps organizations in fair decision-making and data-driven insights.
- The app provides instant predictions through a simple and user-friendly interface.





# Thank you .

*"The best way to predict the future is to create it."*

*— Peter Drucker*