

---

# CAPSTONE PROJECT

## EMPLOYEE SALARY PREDICTION

### Presented By:

1. Mamilla Venkata Sai Nitheesh
2. Gandhi Institute of Technology and Management

# OUTLINE

- **Problem Statement**
- **System Development Approach (Technology Used)**
- **Algorithm & Deployment (Step by Step Procedure)**
- **Result**
- **Conclusion**
- **Future Scope**
- **References**

# PROBLEM STATEMENT

- Employee salary prediction is a significant challenge for HR departments in modern organizations. Traditional manual methods often lead to inconsistencies due to the influence of multiple variables such as:
  - Education
  - Years of experience
  - Job role
  - Location
- This project aims to apply **Machine Learning techniques** to predict employee salaries using historical data, allowing companies to offer fair, competitive, and **data-driven compensation packages**.

# SYSTEM APPROACH

- **Programming Language:** Python
- **Libraries Used:**
  - pandas, numpy – data processing
  - matplotlib, seaborn – data visualization
  - scikit-learn – machine learning
- **IDE/Tool:** Jupyter Notebook
- **Dataset:** Custom Employee Salary Dataset
- **Model Deployment:** joblib used to save the trained model in .pkl format
- **Visualization Techniques:**
  - Red bar plots for categorical data (like gender, department)
  - Correlation heatmaps

# ALGORITHM & DEPLOYMENT

- **Import Required Libraries**

(pandas, numpy, matplotlib, seaborn, sklearn, etc.)

- **Load Dataset**

Read the CSV or tabular data using pandas.

- **Data Cleaning**

- Handle missing values
- Remove outliers
- Normalize/standardize if required

- **Data Visualization**

- Red bar plots for features like gender, department, education
- Heatmaps for correlation analysis

- **Feature Encoding**

Convert categorical variables to numerical using techniques like Label Encoding or OneHotEncoding.

- **Split Dataset**

Divide the data into training and testing sets (e.g., 80/20 split).

- **Model Building**

Use **Linear Regression** from `sklearn.linear_model`.

- **Model Evaluation**

- Use metrics like **R<sup>2</sup> Score**, **Mean Squared Error (MSE)**
- Plot **Actual vs Predicted Salary**

- **Model Deployment**

Save the trained model using joblib for real-world integration

# RESULT

- **Model Performance:**
- **R<sup>2</sup> Score:** 0.84 (indicating good prediction strength)
- **MSE:** Low error value (indicating accurate predictions)
- **Visual Outputs:**
- Frequency plots of categorical features
- Correlation heatmap
- Actual vs Predicted Salary Plot
- **Github:** <https://github.com/NitheeshMVS/Salary-predictor.git>

# CONCLUSION

- The machine learning-based salary prediction model effectively estimates salaries based on key features. This tool can help:
- HR professionals to make objective decisions
- Eliminate bias in salary distribution
- Improve transparency and consistency
- **Challenges faced:**
- Handling missing values
- Outlier detection and correction
- Scaling of features
- The model is interpretable and extendable for future improvements.

## FUTURE SCOPE(OPTIONAL)

- Use **advanced algorithms** like:
- Random Forest
- XGBoost
- Develop a **web application** using Flask or Django
- **Integrate** with HRMS (Human Resource Management Systems)
- Extend functionality to:
- Predict promotions and bonuses
- Provide salary benchmarking dashboards
- Design an intuitive **UI for HR professionals**



# REFERENCES

- Scikit-learn Documentation
- Pandas and NumPy Docs
- “Hands-On Machine Learning” by Aurélien Géron
- Kaggle Employee Datasets
- GitHub Repositories
- Stack Overflow for troubleshooting



**THANK YOU**