# HR Analytics Promotion Prediction Project

Presented by: Kiran Teja

## 1. Project Overview

The main goal of this project is to find out whether an employee will be promoted or not. I used real HR data from a company and applied machine learning techniques to make this prediction.  
  
This can help HR teams take better promotion decisions — not just based on opinions, but based on real data. This makes the process more fair and efficient.

## 2. Approach & Method

**Data Loading**:  
- I got the data from a MySQL database using Python.  
  
 **Cleaning the Data**:  
- Some columns like education and previous year’s rating had missing values, so I filled them properly.  
- I changed text columns (like department, gender) into numbers using Label Encoding.  
- I removed the employee\_id column because it’s not useful for predictions.  
- Then I saved this cleaned data as cleaned3\_hr\_data.csv.  
  
 Understanding the Data (EDA):  
- I created graphs to understand how different things like training scores or number of trainings affect promotion chances.  
- I also used a heatmap to see how features are related to each other.  
- I grouped previous ratings into categories (like low, mid, high) to compare promotion chances.

## 3. Building the Prediction Model (Model Building)

I divided the data into two parts — 80% for training and 20% for testing.  
  
Then, I trained 3 different models:  
1. Logistic Regression – simple and easy to understand  
2. Random Forest – more powerful, handles imbalance better  
3. XGBoost – the best-performing model in this project  
  
I used a special setting in XGBoost (scale\_pos\_weight) to deal with the fact that promoted employees were very few compared to non-promoted ones.  
  
I saved the final model and encoding rules using joblib.  
  
Then, I created a Streamlit web app so others can use this model by entering employee details.

## 4. Model Evaluation

XGBoost (final model) gave these results:  
- Accuracy: 82%  
- Precision: 29%  
- Recall: 75%  
- F1-score: 42%  
- ROC AUC Score: 0.8992  
  
 It is very good at finding promotable employees (high recall).  
 But it also selects some wrong ones (low precision), so HR should review before finalizing.

## 5. Top Features

According to the XGBoost model, these are the top 5 factors that influenced promotions:  
1. KPIs\_met >80% – Whether employee met their key performance goals  
2. awards\_won? – If they won awards  
3. department – Which department they belong to  
4. avg\_training\_score – Their score in training programs  
5. previous\_year\_rating – How well they did last year

## 6. Findings & Conclusion

- XGBoost gave the best results compared to other models.  
- It’s good at catching employees who deserve promotion, but also makes some mistakes.  
- That’s why it should be used as a support tool, not the only decision-maker.  
- I successfully created a web app with Streamlit.

## 7. Tools & Technologies

- Python for coding  
- Pandas, Seaborn, Matplotlib for data and graphs  
- MySQL + SQLAlchemy to get the data  
- Scikit-learn + XGBoost to build models  
- Joblib to save models  
- Streamlit to create the user interface

## 8. Future Scope

- Add more features like employee feedback or manager’s review  
- Use better techniques to improve precision (like SMOTE or ensemble models)  
- Add a dashboard to show results visually  
- Allow the model to retrain itself when new data is added (continuous learning)