Employee Performance Prediction User Guide

# 1. Project Overview

This project analyzes employee performance data, builds machine learning models to predict performance outcomes, and visualizes results for insights. It is structured for modularity, with separate folders for preprocessing, modeling, visualization, and documentation.

# 2. Project Folder Structure

Project Root/  
│  
├── Project Summary/   
│ ├── Requirement.txt <- Contains third-party data and supporting materials  
│ ├── EMPLOYEE PERFORMANCE ANALYSIS – PROJECT SUMMARY <- High-level summary of the project  
│  
├── data/ <- Contains all datasets  
│ ├── INX\_Future\_Inc\_Employee\_Performance\_CDS\_Project2\_Data\_V1.8.xls  
│ ├── preprocessed\_data.csv  
│  
├── src/ <- All source notebooks and scripts  
│ ├── Data Processing/  
│ │ ├── data\_processing.ipynb <- Data cleaning and preprocessing  
│ ├── models/  
│ │ ├── Train\_and predict\_model.ipynb <- Model training and evaluation  
│ └── visualization/  
│ └── visualization.ipynb <- EDA and visual reporting  
│  
└── references/ <- Contains manuals and documentation (including this user guide)

# 3. System Requirements

• Python version 3.8 or higher

• Recommended IDE: Jupyter Notebook / JupyterLab

• Install dependencies using:

* pip install pandas numpy matplotlib seaborn scikit-learn xgboost

# 4. Notebook 1: Data Processing

Location: src/Data Processing/data\_processing.ipynb

Purpose: Cleans, encodes, and prepares the employee dataset for modeling.

**Main Steps:**

1. Load raw data from /data/INX\_Future\_Inc\_Employee\_Performance\_CDS\_Project2\_Data\_V1.8.xls

2. Handle missing values and encode categorical features using LabelEncoder or OneHotEncoder.

3. Standardize features using StandardScaler.

4. Save the cleaned dataset as preprocessed\_data.csv in the /data directory.

* Output Files:

• /data/preprocessed\_data.csv - Cleaned dataset ready for modeling.

# 5. Notebook 2: Train and Predict Model

Location: src/models/Train\_and predict\_model.ipynb

Purpose: Trains and evaluates multiple machine learning models on the preprocessed data.

**Main Steps:**

1. Load preprocessed data from /data/preprocessed\_data.csv.

2. Split the dataset into training and testing sets using train\_test\_split.

3. Scale features using StandardScaler.

4. Train various models (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, etc.).

5. Evaluate models using accuracy, precision, recall, and F1-score.

6. Save the best-performing model for future prediction tasks.

# 6. Notebook 3: Visualization

Location: src/visualization/visualization.ipynb

Purpose: Conducts exploratory data analysis (EDA) and generates visual reports.

**Main Steps:**

1. Load the raw dataset from /data/.

2. Perform EDA using statistical plots and correlation heatmaps.

3. Identify key patterns, trends, and potential outliers.

4. Display all visualizations inline within the notebook.

# 7. Typical Workflow

1. Run visualization.ipynb to explore and interpret the data visually

2. Run data\_processing.ipynb to clean and prepare the data.

3. Run Train\_and predict\_model.ipynb to train and evaluate models.

4. Review all generated files under /data, /models, and /visualization outputs.

# 8. Expected Outputs

|  |  |  |
| --- | --- | --- |
| Notebook | Output File | Description |
| data\_processing | preprocessed\_data.csv | Cleaned dataset after preprocessing |
| Train\_and predict\_model | best\_model.pkl (optional) | Serialized trained model |
| visualization | inline plots | EDA results and graphical insights |