HR Analytics – Project Execution Log

# 1. Project Overview

This project aims to predict employee attrition using IBM HR Analytics data.  
The complete workflow was implemented in Python scripts using VS Code instead of Jupyter, following professional coding structure and versioning practices.

# 2. Folder Structure

HR\_Analytics\_Attrition\_Project/  
├── data/ # Raw and cleaned CSV files  
├── scripts/ # Python scripts for each project phase  
├── models/ # Trained model .pkl files  
├── app/ # Streamlit app, visuals  
├── report/ # Final report and PDFs  
├── requirements.txt # Dependency list  
├── README.md # Project summary for GitHub

# 3. Step-by-Step Execution

## 3.1 Data Cleaning

Script: 01\_data\_cleaning.py  
- Dropped columns: EmployeeCount, StandardHours, Over18, EmployeeNumber  
- LabelEncoded: Attrition, Gender, OverTime  
- One-hot encoded remaining categorical columns  
- Saved to: data/cleaned\_hr\_data.csv

## 3.2 EDA Visuals

Script: 02\_eda\_visuals.py  
- Plots created:  
 • Attrition Count  
 • Age vs Attrition  
 • Monthly Income vs Attrition  
 • Correlation Heatmap  
- Saved to: visuals/\*.png

## 3.3 Model Building

Script: 03\_model\_building.py  
- Models trained: Logistic Regression, Decision Tree  
- Accuracy (LogReg): ~86%  
- Metrics: Accuracy, Confusion Matrix, Classification Report  
- Saved model: models/model.pkl

## 3.4 SHAP Explainability

Script: 04\_shap\_explainability.py  
- Used SHAP KernelExplainer with predict\_proba  
- Explained first 100 samples for performance  
- Generated SHAP summary plot for class 1 (Attrition)  
- Saved to: app/shap\_summary\_plot.png

# 4. Tools & Libraries

Python, pandas, scikit-learn, matplotlib, seaborn, SHAP, Streamlit (for frontend), VS Code.  
Project is ready for deployment and GitHub push.