Tabular Classification & Pipeline

Goal:

Assess hands-on skills in data ingestion, cleaning, feature engineering, model building, evaluation, and pipeline design using Python (pandas, scikit-learn).

Dataset

Use the “Adult Income” dataset from the UCI Machine Learning Repository:

- **Data:** https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data
- **Description:** https://archive.ics.uci.edu/ml/datasets/Adult

Task: Predict whether an individual’s income exceeds $50K/year (binary classification).

Assignment Details

A. Data Ingestion & Cleaning

- Load the CSV into a pandas DataFrame.
- Assign appropriate column names using the “adult.names” file.
- Handle missing or “?” entries: decide whether to impute, drop, or otherwise transform.
- Provide summary statistics and basic visualizations of key features.

B. Exploratory Data Analysis (EDA)

- Analyze class imbalance.
- Visualize the relationship between at least two categorical features and the target.
- Plot distributions (histogram or boxplot) for two numerical features, segmented by the income class.

C. Feature Engineering & Preprocessing

- Encode categorical variables (e.g., one-hot or ordinal encoding).
- Scale numerical features if needed.
- Create at least two new features (e.g., combining hours-per-week and education, or binning age).
- Build a scikit-learn Pipeline that bundles preprocessing and model training.

D. Model Training & Hyperparameter Tuning

- Train two different classifiers (e.g., Logistic Regression, Random Forest, XGBoost) within your pipeline.
- Use 5-fold cross-validation and grid search or randomized search to tune key hyperparameters.
- Report performance on a held-out test set using accuracy, ROC AUC, and F1-score.

E. Model Interpretation

- For your best model, extract feature importances or coefficients.
- Use SHAP or permutation importance to identify the top five drivers of high income.
- Provide brief commentary (2–3 sentences) on any surprising insights.

F. (Optional Bonus) Pipeline Export

- Demonstrate how to serialize your pipeline (e.g., with `joblib` or `pickle`).
- Include a short snippet showing how to load the pipeline and make a prediction on a new sample.

Deliverables

1. **Jupyter Notebook** (or Python script) containing all code, visualizations, and commentary.
2. **README.md** with:
   - Environment setup instructions (e.g. requirements.txt).
   - How to run your code and interpret results.
3. **Results Summary** (in the notebook or separate PDF):
   - Data-cleaning decisions.
   - Final model performance metrics.
   - Key feature-importance insights.

Submit everything as a single GitHub repository (public or private link).