Assignment: Predicting Loan Default Using PySpark MLlib

Objective
Build a scalable machine learning pipeline in PySpark MLlib to predict whether a customer will default on a loan.

Part 1 – Data Preparation

Dataset: Create or use a CSV file named `loan_data.csv` with the following columns:

- **customerID** (string)
- **age** (integer)
- **gender** (string: Male/Female)
- **income** (double)
- **loan_amount** (double)
- **loan_term** (integer: months)
- **credit_score** (double)
- **employment_status** (string: Employed, Unemployed, Self-employed, Student)
- **marital_status** (string: Single, Married, Divorced)
- **default** (string: Yes/No) – **target variable**

Tasks:

1. **Load the CSV** into a Spark DataFrame.
2. Display the schema using `printSchema()`.
3. Show the first 10 rows using `show()`.
4. Count the number of defaults vs. non-defaults using `groupBy().count()`.

Part 2 – Feature Engineering

1. **Handle Categorical Features:** Convert `gender`, `employment_status`, `marital_status` to numeric using `StringIndexer` and `OneHotEncoder`.
2. **Assemble Features:** Combine numerical and encoded categorical features using `VectorAssembler`.
3. **Split Dataset:** Divide data into **training (70%)** and **test (30%)** sets.

Part 3 – Model Training

1. Train a **Logistic Regression model** to predict loan default.
2. Train a **Random Forest Classifier** to improve accuracy.
3. Compare performance between models.

Part 4 – Model Evaluation

1. Use `BinaryClassificationEvaluator` to calculate **AUC**.
2. Print **precision, recall, accuracy** for each model.
3. Display the **confusion matrix**.

Part 5 – Bonus Tasks

1. Use `CrossValidator` or `TrainValidationSplit` for hyperparameter tuning.
2. Extract **feature importance** from the Random Forest model.
3. Save the trained model and load it back using `PipelineModel.load()`.