🧾 Step-by-Step Execution Procedure: Diabetes Prediction with ML

Step 1: Data Acquisition & Overview

- Load the original dataset containing **100,000 rows and 9 columns**.
- Features include: `gender`, `age`, `hypertension`, `heart disease`, `smoking history`, `BMI`, `HbA1c_level`, `blood_glucose_level`, and `diabetes`.

Step 2: Exploratory Data Analysis (EDA)

1. **Understand Class Distribution**
    - Diabetes = 8.5%, Non-diabetes = 91.5%
2. **Correlation Matrix**
    - Strong predictors: `HbA1c_level`, `blood_glucose_level`
3. **Cross-tabulations**
    - Hypertension, heart disease, and smoking history correlate with diabetes
4. **Outlier/Noise Check**
    - Filter out 911 patients with `age < 1` (none have diabetes)

Step 3: Data Preprocessing

1. **Label Encoding**
    - Convert categorical variables (`gender`, `smoking_history`) into numerical labels
2. **Handle Missing Data**

- Convert "No Info" in smoking_history to NaN
- Apply **KNN imputation** after scaling numeric features

3. **Remove Irrelevant Rows**
   - Drop patients with age < 1 year
4. **Split Dataset**
   - Use train_test_split (80% train, 20% test) → Training: 79,272 | Testing: 19,818
5. **Address Class Imbalance**
   - Apply **SMOTE** to balance training data (50/50 diabetes and non-diabetes)
   - After SMOTE: Training set = 144,936 rows
6. **Feature Scaling**
   - Use StandardScaler on numerical columns: age, bmi, HbA1c_level, blood_glucose_level
   - Apply same scaling to both train and test sets

Step 4: Model Selection

Choose four models for binary classification:

1. **Random Forest Classifier**
   - Tree-based, non-linear, interpretable via feature importance
2. **Logistic Regression**
   - Linear, interpretable baseline model
3. **Support Vector Classifier (SVC)**
   - RBF kernel, suited for high-dimensional, non-linear patterns
4. **XGBoost Classifier**

- Gradient boosting, highly accurate, handles complex feature interactions

---

Step 5: Model Training

- Train all models on the **SMOTE-balanced training set**
- Use default hyperparameters
- For models requiring it (Logistic Regression, SVC), ensure features are scaled

---

Step 6: Model Evaluation

Evaluate each model using the following metrics:

- **Accuracy** – Overall correctness
- **Precision** – Minimize false positives
- **Recall** – Minimize false negatives (critical for healthcare)
- **F1-Score** – Balance of precision and recall
- **ROC-AUC** – Robust to class imbalance, reflects model's ability to distinguish classes

---

Step 7: Results Analysis

Model	Set	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	Train	0.999317	0.999696	0.998937	0.999317	0.999996
Random Forest	Test	0.965385	0.858765	0.713023	0.779137	0.966835
Logistic Regression	Train	0.88982	0.889793	0.889855	0.889824	0.964767
Logistic Regression	Test	0.889595	0.428237	0.863288	0.572489	0.96092
SVM	Train	0.890752	0.888372	0.893815	0.891085	0.964641
SVM	Test	0.889293	0.427825	0.868002	0.573152	0.960863
XGBoost	Train	0.982109	0.998061	0.966095	0.981818	0.99853
XGBoost	Test	0.969523	0.913076	0.711844	0.8	0.977717