Team Name: Vayam Coders

Team Leader Name: KARUNA SHANKAR PRAJAPATI, SHYAM SUNDAR PRAJAPATIf

Problem Statement: Manual classification of iris flowers based on petal/sepal dimensions is time-consuming and error-prone. We propose an automated ML solution

that predicts the species with high accuracy using minimal features.

Brief about the prototype

>This project demonstrates a machine learning solution for classifying iris flowers into three species (Setosa, Versicolor, Virginica) using four simple flower measurements (sepal length, sepal width, petal length, petal width).

>The model is built using a Support Vector Classifier (SVM) with a preprocessing pipeline (scaling + encoding).

>It achieves ~96% accuracy, showing reliable and efficient classification.

>The prototype is lightweight, reproducible, and extendable to other small-scale classification tasks.

>It serves as a quick yet complete end-to-end ML pipeline, from dataset to model training, evaluation, and saving for reuse.

Brief about the prototype

>This project demonstrates a machine learning solution for classifying iris flowers into three species (Setosa, Versicolor, Virginica) using four simple flower measurements (sepal length, sepal width, petal length, petal width).

>The model is built using a Support Vector Classifier (SVM) with a preprocessing pipeline (scaling + encoding).

>It achieves ~96% accuracy, showing reliable and efficient classification.

>The prototype is lightweight, reproducible, and extendable to other small-scale classification tasks.

>It serves as a quick yet complete end-to-end ML pipeline, from dataset to model training, evaluation, and saving for reuse.

Opportunity should be able to explain the following:

How different is it from other existing solutions?

Lightweight, quick to train, and easy to reproduce.

End-to-end pipeline (data → preprocessing → model → evaluation).

Uses minimal resources yet achieves ~96% accuracy.

How will it solve the problem?

Automates iris flower classification using four measurements.

Provides fast, consistent, and accurate predictions.

Adaptable to other small classification tasks.

USP of the proposed solution

Lightweight & reproducible on any laptop/Colab.

Educational, extensible, and scalable.

High accuracy with a simple approach.

List of features offered by the solution:

>**Data Loading and Preprocessing:** Handles data import and cleaning.

>**Feature Selection:** Selects relevant features for the model.

>**Model Training:** Trains ML models like SVM, Logistic Regression, or Random Forest.

>**Model Evaluation:** Measures performance using accuracy, confusion matrix, and classification report.

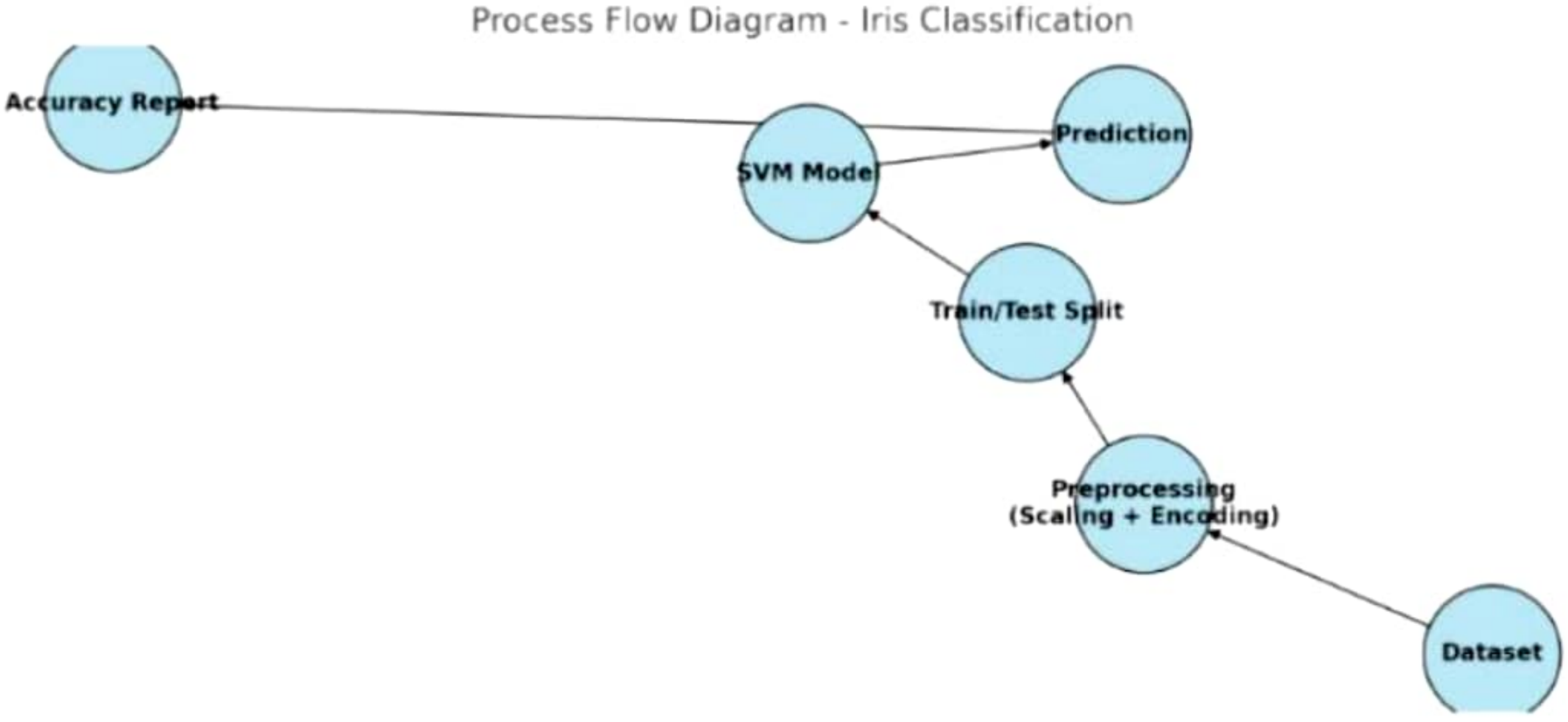
>**Prediction/Inference:** Predicts labels for new/unseen data.

>**Visualization:** Displays key metrics and graphs (optional).

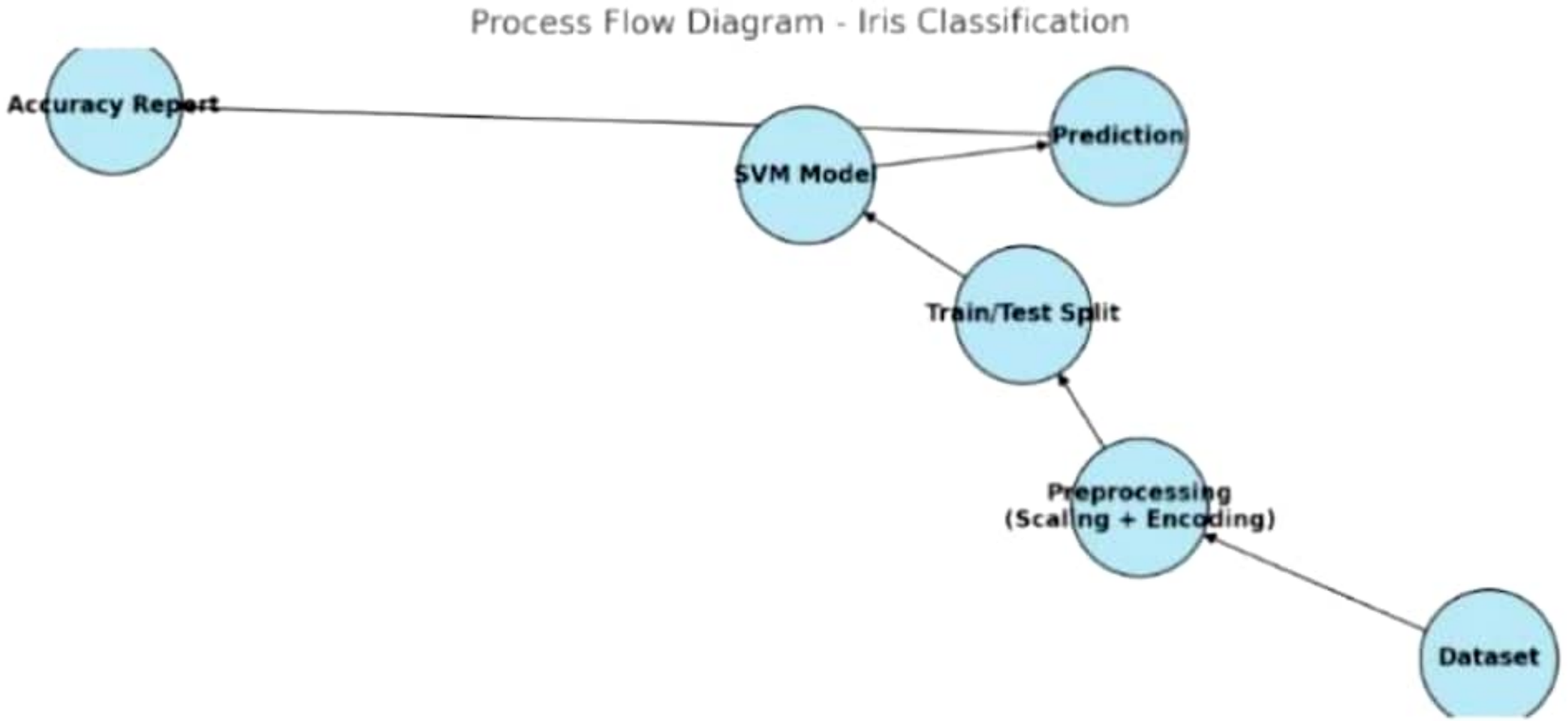
>**User Instructions:** Clear steps to run/train the model.

>**Reproducibility:** Fixed random seeds and environment details for consistent results.

Process flow diagram or Use-case diagram



Wireframes/Mock diagrams of the proposed solution (optional)



ML Pipeline Flowchart

Architecture diagram of the proposed solution

Dataset → Preprocessing (Scaling + Encoding) → Train/Test Split →
SVM Classifier → Predictions → Accuracy Report

Technologies to be used in the solution

Programming Language: Python

Libraries & Tools:

pandas → data handling & analysis

scikit-learn → preprocessing, model building (SVM), evaluation

joblib → model saving & loading

Platform: Google Colab / Jupyter Notebook

Visualization: matplotlib (optional, for plots like confusion matrix)

Estimated implementation cost (optional):

Development Cost: For a simple, classic ML project done solo: ₹0

Cloud Compute Cost: Free, if using Google Colab or Kaggle; otherwise, basic cloud VMs typically cost ~$0.05/hour

Add as per the requirements for the hackathon:

>Clean, documented source code (Python/Jupyter).

>Dataset or its source link. >null

>A brief PDF report: problem, dataset, model, results.

>Submit as per hackathon format (zip/GitHub link).