ASSIGNMENT # 02: BEE-14

CS-471 Machine Learning

Submission Deadline: 7th Nov 2025

Assignment Title: Implementation and Comparison of Classification Algorithms

Objective:
The objective of this assignment is to understand, implement, and compare a variety of **classification algorithms** on a **structured (tabular) dataset**.
You will:

- Implement **Logistic Regression**, **Support Vector Machine (SVM)**, and **Naive Bayes** classifiers **from scratch**.
- Use built-in implementations for **Decision Tree**, **Random Forest**, and **AdaBoost**.
- Preprocess data, train models, evaluate their performance, and analyse results.

Part1: Dataset Selection
Select a **tabular dataset** suitable for binary or multi-class classification (Tabular/image data).
Choose any dataset of your own choice from these sources:

https://archive.ics.uci.edu/

https://www.kaggle.com/datasets

Part 2: Data Preprocessing
Perform all necessary preprocessing before training models:

1. Load dataset and display sample rows.
2. Handle **missing values** (remove or impute).
3. Convert **categorical** features using Label Encoding or One-Hot Encoding.
4. Perform **feature scaling** (Standardization or Min–Max scaling).
5. Split dataset into **Training (80%)** and **Testing (20%)** sets.

(Document all preprocessing decisions in your notebook.)

Part 3: Model Implementation
A) Implement From Scratch

1. **Logistic Regression**
    - Use **gradient descent** for optimization.
    - Implement sigmoid activation, cost function, and parameter updates.
2. **Support Vector Machine (SVM)**
    - Implement linear SVM using **hinge loss** and **gradient descent** updates.
3. **Naive Bayes**
    - Implement **Gaussian Naive Bayes**:
        - Estimate class-wise mean, variance, and prior probabilities.
        - Compute posterior probabilities manually for prediction.

B) Use Built-in Implementations

4. Decision Tree Classifier
5. Random Forest Classifier
6. AdaBoost Classifier

Part 4: Model Evaluation

For each model:

- Predict on **training and testing** data and Compute:
  - Accuracy
  - Precision
  - Recall
  - F1-score
  - Confusion Matrix
- For binary classification: include **ROC–AUC** score and plot the ROC curve.

Also:

- Compare results in a **table**.
- Discuss any patterns (e.g., overfitting, bias-variance trade-off).

Part 5: Visualization

Include the following visualizations:

- Confusion Matrix (heatmap) for each model.
- Bar plot comparing accuracies of all models.
- For 2D datasets (like Iris: only select two features), show decision boundaries for from-scratch models.

Part 6: Report / Notebook

Submit a **Jupyter Notebook (.ipynb)** containing:

- Code with clear comments.
- Step-by-step explanations and outputs.
- Summary table of model results.
- **Conclusion section** discussing:
  - Which model performed best and why.
  - Strengths and weaknesses of your from-scratch models.

Deliverables

- Jupyter Notebook: Assignment_Classification_<YourName>.ipynb
- 2 page report summarizing key findings.

Note: Your submitted code should be neat and clean with proper comments added.