MINI PROJECT

Step 1 — Load and Prepare the Dataset

1. Import the built-in **Breast Cancer dataset** from `sklearn.datasets`.
2. Store features (`X`) and labels (`y`).
3. Split the dataset into **training (70%)** and **testing (30%)** using `train_test_split()` with stratification (to keep class balance).
4. Standardize the data using **StandardScaler()** to make features have zero mean and unit variance (important for SVM, Perceptron, and PCA).

Step 2 — Implement Simple Linear Regression (Least Squares Method)

1. Select one input feature (for example, *mean radius*).
2. Compute:

   m=∑(X−X¯)(Y−Y¯)∑(X−X¯)2andc=Y¯−mX¯m = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sum (X - \bar{X})^2} \quad\text{and}\quad c = \bar{Y} - m\bar{X}m=∑(X−X¯)2∑(X−X¯)(Y−Y¯)andc=Y¯−mX¯

   where `m` is slope and `c` is intercept.

3. Predict output using Y^=mX+c\hat{Y} = mX + cY^=mX+c.
4. Convert predictions into binary classes:
   If Y^≥0.5⇒1\hat{Y} ≥ 0.5 \Rightarrow 1Y^≥0.5⇒1, else 000.
5. Compute **Accuracy** and **Classification Report**.

Step 3 — Train a Single-Layer Perceptron

1. Use the **Perceptron** algorithm from `sklearn.linear_model`.
2. Fit the model on **standardized full features**.
3. Predict the test data.
4. Compute accuracy and classification report.
5. Visualize decision boundary using 2 features (e.g., mean radius and mean texture).

Step 4 — Train an SVM Classifier

1. Use **Support Vector Machine (SVC)** from `sklearn.svm`.
2. Perform **GridSearchCV** to find best hyperparameters (`C` and `gamma`).
3. Train the best SVM model on training data.
4. Predict on test data and compute metrics.
5. Visualize decision boundary and **support vectors** using 2 features.

Step 5 — Apply PCA (Principal Component Analysis)

1. Perform **PCA** on standardized features to reduce dimensions.
   - Extract top **2 principal components** for 2D view.
   - Extract top **3 components** for 3D view.
2. Plot:
   - 2D scatter of PC1 vs PC2.
   - 3D scatter of PC1, PC2, PC3.
3. Train a **Perceptron** on top-2 PCA features and evaluate accuracy.

Step 6 — Train Logistic Regression (Baseline Classifier)

1. Train **Logistic Regression** on full standardized features.
2. Predict the test set.
3. Evaluate accuracy and classification report.
4. Compare with other models.

Step 7 — Plot ROC Curves

1. For models that produce probabilities (SVM, Logistic Regression), get prediction probabilities.
2. Use `roc_curve()` and `auc()` to compute:
   - True Positive Rate (TPR)
   - False Positive Rate (FPR)
   - Area Under Curve (AUC)
3. Plot all ROC curves in one graph for comparison.

Step 8 — Compare All Models

1. Collect accuracies of all models:
   - Linear Regression (1 feature)
   - Perceptron (full)
   - SVM (tuned)
   - Perceptron (PCA)
   - Logistic Regression
2. Display them in a **bar chart** and a summary **table**.

Step 9 — Interpret Results

1. Observe that **Perceptron, SVM, and Logistic Regression** perform best (~98% accuracy).

2. **Linear Regression (thresholded)** performs weaker since it's not designed for classification.
3. **PCA-based Perceptron** performs slightly lower but is useful for visualization and dimensionality reduction.

- **Manual Least Squares (Simple Linear Regression)**
  - Implemented manually using one feature.
  - Used to demonstrate regression and threshold-based classification.
  - **Accuracy:** `0.8538`
- **Perceptron (Full Features)**
  - Uses all 30 breast cancer features after standard scaling.
  - **Accuracy:** `0.9825`
- **Support Vector Machine (SVM)**
  - Kernel: `rbf`
  - Tuned using `GridSearchCV` over `C` and `gamma`.
  - **Best Params:** `C=10, gamma=0.01`
  - **Accuracy:** `0.9825`
- **PCA + Perceptron**
  - PCA reduced dataset to 2D and 3D for visualization.
  - Then a Perceptron trained on 2D PCA-transformed data.
  - **Accuracy:** `0.9357`
- **Logistic Regression (Full Features)**
  - Baseline classifier using all features.
  - **Accuracy:** `0.9883`