1. Logistic Regression

```
Input: Training data X, labels Y, learning rate α, epochs EInitialize weights W and bias b
For epoch = 1 to E
   For each training example (x, y)
      z = W · x + b
      y_hat = 1 / (1 + e^(-z))
      dw = (y_hat - y) * x
      db = (y_hat - y)
      W = W - α * dw
      b = b - α * db
Return W, b
```

---

2. K-Nearest Neighbors (KNN)

```
Input: Training data X, labels Y, test sample x_test, k
For each x in X
   distance = compute_distance(x, x_test)
Sort distances in ascending order
Select first k samples
Get labels of selected samples
y_pred = most_frequent_label(labels)
Return y_pred
```

---

3. Naive Bayes

```
Input: Training data X, labels Y, test sample x_test

For each class c in Y
   prior[c] = count(c) / total_samples
   likelihood[c] = 1
   For each feature i
      likelihood[c] *= P(x_test[i] | c)
   posterior[c] = prior[c] * likelihood[c]
y_pred = class with maximum posterior
Return y_pred
```

---

4. Decision Tree

```
Input: Training data X, labels Y

If all labels are same
   Return label
If no features left
   Return majority_label
Select best_feature using information_gain
Create node with best_feature
For each value v of best_feature
   Subset = samples where best_feature == v
   Child = build_tree(Subset)
   Add Child to node
Return node
```