1. Py

**Prompt:**

Create a Python script to compare Logistic Regression and SVM for a simple 1D binary classification problem using the CRISP-DM framework:

1. **Business Understanding**: Show differences in predictive behavior between Logistic Regression and SVM.
2. **Data Understanding**: Generate synthetic data (X: 0–1000) with binary labels (Y: 1 if 500 < X < 800, else 0).
3. **Data Preparation**: Split the data into training and testing sets (80-20 split).
4. **Modeling**: Train Logistic Regression and SVM models and predict on the test set.
5. **Evaluation**: Use scatter plots to compare predictions from both models against true labels.
6. **Deployment**: Present results with clear, side-by-side visualizations.

Use numpy, scikit-learn, and matplotlib for implementation.

一張含有 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

1. py

**Prompt:**

Create a Python script to compare Logistic Regression and SVM for a 1D binary classification task using the CRISP-DM framework:

1. **Business Understanding**: Compare the decision boundaries and probabilities of Logistic Regression and SVM.
2. **Data Understanding**: Generate 300 random values (X) between 0 and 1000, with binary labels (Y) based on the rule: 1 if 500 < X < 800, else 0.
3. **Data Preparation**: Split the dataset into 80% training and 20% testing.
4. **Modeling**: Train both models, predict probabilities and classes, and compute decision boundaries.
5. **Evaluation**: Visualize predictions, probability curves, and decision boundaries in two side-by-side plots.
6. **Deployment**: Use matplotlib to clearly present the results, including true label boundaries and decision thresholds.

一張含有 文字, 行, 繪圖, 軟體 的圖片

自動產生的描述

3D.py

**Prompt:**

Create a Python script to classify and visualize 3D data using a Linear Support Vector Classifier (LinearSVC), following the CRISP-DM framework:

1. **Business Understanding**  
   Demonstrate the ability of a linear SVM to classify data in 3D space by finding and visualizing the separating hyperplane for a synthetic dataset.
2. **Data Understanding**
   * Generate 600 random points (x1, x2) centered at (0, 0) with a variance of 10.
   * Compute the distance of each point from the origin and assign labels (Y):
     + Y=0Y=0 if the distance < 4.
     + Y=1Y=1 otherwise.
   * Define a third feature, x3, as a Gaussian function of x1 and x2.
3. **Data Preparation**
   * Combine x1, x2, and x3 into a feature matrix (X).
   * Ensure the data is properly labeled and ready for SVM training.
4. **Modeling**
   * Train a LinearSVC on the 3D data (X, Y).
   * Extract the model’s coefficients and intercept to define the separating hyperplane.
5. **Evaluation**
   * Create a 3D scatter plot:
     + Use different colors and markers for Y=0Y=0 and Y=1Y=1.
     + Plot the separating hyperplane as a translucent surface.
   * Ensure the plot includes proper axes labels, a title, and a legend for clarity.
6. **Deployment**
   * Visualize the results with matplotlib, showcasing both the data distribution and the SVM’s decision boundary.
   * Highlight the separation between the two classes in 3D space.