Task 7: Support Vector Machines (SVM)

☆ Task 7: Support Vector Machines (SVM) – Full Explanation

This task focuses on understanding and implementing **SVM**, a powerful supervised machine learning algorithm used for **classification**.

🎯 Objective

Use **Support Vector Machines (SVMs)** to perform:
Linear classification (data that is linearly separable)
Non-linear classification (data that needs curves to separate classes)
You will compare how SVM behaves with different **kernels** like:
Linear kernel
RBF (Radial Basis Function) kernel

🛠 Tools Required

You will use the following Python libraries:
Scikit-learn → for SVM models
NumPy → for dataset handling
Matplotlib → for plotting decision boundaries

📓 Hints / Mini Guide (Detailed Explanation)

1. Load and prepare a dataset for binary classification

You can use:

sklearn.datasets.make_blobs()

sklearn.datasets.make_circles()

Or any CSV dataset

You need two classes (example: Class 0 and Class 1).

Steps:

Split into **train/test**

Standardize data using StandardScaler

2. Train an SVM with linear and RBF kernel

Use:

```
from sklearn.svm import SVC


svm_linear =
SVC(kernel='linear')
svm_rbf = SVC(kernel='rbf')
Fit both models:
svm_linear.fit(X_train,
y_train)

svm_rbf.fit(X_train, y_train)


This compares how linear vs
non-linear boundaries behave.
```

3. Visualize the decision boundary using 2D data

For a 2D dataset (two features):

Create a mesh grid

Predict over the grid

Draw contours using Matplotlib

This helps you see how SVM separates the classes.

Example boundaries:

Linear SVM → straight line

RBF SVM → curved boundaries

4. Tune hyperparameters like C and gamma

Hyperparameters in SVM:

Parameter	Meaning
C	Controls misclassification penalty
Gamma (RBF only)	Controls the curve of decision boundary

Higher C → stricter margin

Higher Gamma → more complex boundary

Use **GridSearchCV** to find best values.

5. Use cross-validation to evaluate performance

Evaluate your model using:

K-fold cross-validation

Accuracy, precision, recall, F1-score

Example:

```
from sklearn.model_selection
import cross_val_score

scores =
cross_val_score(svm_rbf, X, y,
cv=5)

This reduces overfitting and
gives a reliable accuracy
estimate.
```

📌 What You Learn From This Task

✓ How SVM works

✓ Difference between linear and RBF kernel

✓ How to visualize classification boundaries

✓ How to tune SVM hyperparameters (C, gamma)

✓ How to evaluate an ML model properly

If you want, I can also give you:

✅ Ready-to-run Python code for this task
✅ PDF version
✅ A report-style explanation for assignment submission
Just tell me!