SUPPORT VECTOR MACHINE (SVM) — FULL COMBO NOTES

==

CHAPTER 1 — INTRODUCTION TO SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification, regression, and outlier detection. It finds the optimal separating boundary (hyperplane) between classes by maximizing the margin—the distance between the boundary and the nearest data points (support vectors).

SVM works extremely well with:

- High-dimensional data

- Complex, non-linear patterns

- Small to medium-sized datasets

Only the support vectors matter for defining the decision boundary.

CHAPTER 2 — GEOMETRIC INTUITION

Imagine two groups of points. SVM tries to draw a line that separates them with the largest possible distance from both sides. This distance is called the margin.

Why margin matters:

- Larger margin → better generalization

- Smaller margin → overfitting risk

CHAPTER 3 — HARD MARGIN SVM

Hard margin assumes:

- Data is perfectly separable

- No overlapping points

- No noise

Constraint:

$y_i (w^T x_i + b) \geq 1$

Objective:

Minimize $(1/2) ||w||^2$

Used rarely because real-world data is noisy.

CHAPTER 4 — SOFT MARGIN SVM

Soft margin introduces slack variables ξ_i to allow misclassification.

Constraint:

$y_i (w^T x_i + b) \geq 1 - \xi_i$

Objective:

Minimize $(1/2) ||w||^2 + C \Sigma \xi_i$

Where:

- C = penalty parameter
- Large C → strict
- Small C → relaxed

CHAPTER 5 — ROLE OF C PARAMETER

C controls how strictly the model punishes misclassification.

Low C:

- Wide margin

- More misclassification allowed

- Underfitting possible

High C:

- Narrow margin

- Very strict

- Overfits noise

CHAPTER 6 — KERNEL TRICK

Many datasets are not linearly separable. Kernel trick transforms input data into a higher dimensional space where separation becomes possible.

But instead of transforming explicitly, kernel functions compute similarity directly.

Common kernels:

1. Linear: $K(x, x') = x \cdot x'$

2. Polynomial: $K(x, x') = (x \cdot x' + c)^d$

3. RBF (Gaussian): $K(x, x') = \exp(-\gamma \|x - x'\|^2)$

4. Sigmoid: $K(x, x') = \tanh(\alpha x \cdot x' + c)$

RBF is the most widely used.

CHAPTER 7 — GAMMA PARAMETER

Gamma defines how much influence a single training example has.

Low gamma:

- Influence spreads far

- Smooth boundaries
- Underfits

High gamma:

- Influence is short-range
- Highly curved boundaries
- Overfits

CHAPTER 8 — SVR (SUPPORT VECTOR REGRESSION)

SVR does regression using the same principles of SVM.

It introduces the epsilon-tube:

- Errors within tube are ignored
- Errors outside tube are penalized

SVR constraints:

$|y - (w^T x + b)| \le \varepsilon + \xi_i$

SVR objective:

Minimize $(1/2) \|w\|^2 + C \Sigma (\xi_i + \xi_i^*)$

CHAPTER 9 — ONE-HOT ENCODING & COLUMN TRANSFORMER

SVM requires numeric input. Categorical columns (e.g., 'day') must be converted using OneHotEncoder.

drop='first' removes the first category to avoid dummy variable trap.

ColumnTransformer allows applying transformations to specific columns while leaving the rest unchanged.

CHAPTER 10 — FIT(), TRANSFORM(), FIT_TRANSFORM()

Transformers (OneHotEncoder, StandardScaler):

- fit() learns rules

- transform() applies rules

- fit_transform() does both

Models (SVM, SVR, LogisticRegression):

- fit() trains model

- predict() makes predictions

CHAPTER 11 — GRIDSEARCHCV FOR HYPERPARAMETER TUNING

GridSearchCV tries combinations of hyperparameters to find the best model.

Example:

C: [0.1, 10, 100, 1000]

gamma: [1, 0.1, 0.01, 0.001]

kernel: ['rbf']

It uses cross-validation to pick the best C and gamma.

CHAPTER 12 — FULL SVM WORKFLOW

1. Load dataset

2. Split train/test

3. Encode categorical variables

4. Scale numeric variables

5. Choose kernel

6. Train SVM/SVR

7. Tune hyperparameters with GridSearchCV

8. Evaluate accuracy/performance

CHAPTER 13 — INTERVIEW QUESTIONS

1. What is SVM?

2. What is margin? Support vector?

3. Difference between hard and soft margin?

4. What is C? What is gamma?

5. What is the kernel trick?

6. Why use RBF kernel?

7. Explain SVR.

8. Why use OneHotEncoder?

9. Why drop='first'?

10. What is GridSearchCV?

END OF NOTES