1. What is a Support Vector Machine (SVM)?

A **Support Vector Machine (SVM)** is a supervised machine learning algorithm used for **classification and regression**. It finds the **optimal hyperplane** that separates data points of different classes while **maximizing the margin** between them.

---

2. Difference between Hard Margin and Soft Margin SVM

- **Hard Margin SVM**:
    - Assumes data is perfectly separable
    - No misclassification allowed
    - Very sensitive to noise
- **Soft Margin SVM**:
    - Allows some misclassifications
    - Uses slack variables
    - More robust to noisy and overlapping data

---

3. Mathematical intuition behind SVM

SVM tries to **maximize the margin**, i.e., the distance between the separating hyperplane and the nearest data points from each class.
This is formulated as a **convex optimization problem**, ensuring a global optimum.

---

4. Role of Lagrange Multipliers in SVM

Lagrange multipliers:

- Convert the constrained optimization problem into a dual problem

- Help identify **support vectors**
- Enable the use of **kernel functions**

---

5. What are Support Vectors?

Support vectors are the **data points closest to the decision boundary**.
They directly influence the position and orientation of the hyperplane.

---

6. What is a Support Vector Classifier (SVC)?

A **Support Vector Classifier (SVC)** is an SVM model used specifically for **classification**, capable of handling **non-linear boundaries** using kernels.

---

7. What is a Support Vector Regressor (SVR)?

SVR is an SVM variant for **regression**.
It fits a function within an **ε-insensitive margin**, ignoring small errors.

---

8. What is the Kernel Trick?

The **kernel trick** allows SVMs to operate in high-dimensional feature spaces **without explicitly computing transformations**, making non-linear classification computationally efficient.

---

9. Compare Linear, Polynomial, and RBF Kernels

Kernel	Use Case	Characteristics
Linear	Linearly separable data	Fast, simple
Polynomial	Complex relationships	Degree controls complexity

RBF (Gaussian)	Highly non-linear data	Most flexible, widely used

---

10. Effect of the C parameter in SVM

- **High C** → Low bias, high variance (overfitting)
- **Low C** → High bias, low variance (underfitting)

It controls the trade-off between **margin width and classification error**.

---

11. Role of Gamma in RBF Kernel

Gamma defines how far the influence of a single training point reaches:

- **High gamma** → Overfitting
- **Low gamma** → Underfitting

---

12. What is Naïve Bayes and why "Naïve"?

Naïve Bayes is a probabilistic classifier based on Bayes' Theorem.
It is called **"naïve"** because it assumes **feature independence**, which is rarely true in real data.

---

13. What is Bayes' Theorem?

Bayes' Theorem describes conditional probability:

P(A∣B)=P(B∣A)P(A)P(B)P(A|B) = \frac{P(B|A)P(A)}{P(B)}P(A∣B)=P(B)P(B∣A)P(A)

It updates prior beliefs based on new evidence.

---

14. Differences between Gaussian, Multinomial, and Bernoulli Naïve Bayes

Type	Data Type	Example
Gaussian	Continuous	Height, weight
Multinomial	Count data	Word frequencies
Bernoulli	Binary	Word present or not

---

15. When use Gaussian Naïve Bayes?

Use it when:

- Features are **continuous**
- Data approximately follows a **normal distribution**

---

16. Key assumptions of Naïve Bayes

1. Feature independence
2. Equal importance of features
3. Correct probability estimation from training data

---

17. Advantages and Disadvantages of Naïve Bayes

Advantages

- Fast and scalable
- Works well with small datasets
- Excellent for text data

Disadvantages

- Independence assumption often unrealistic
- Poor performance with correlated features

---

18. Why is Naïve Bayes good for text classification?

- High-dimensional data handled efficiently
- Independence assumption works reasonably well
- Performs well even with limited training data

---

19. Compare SVM and Naïve Bayes

Aspect	SVM	Naïve Bayes
Speed	Slower	Very fast
Accuracy	High	Moderate
Data size	Medium	Large
Interpretability	Low	High

---

20. How does Laplace Smoothing help Naïve Bayes?

Laplace smoothing:

- Prevents **zero probability** issues
- Adds a small constant (usually 1) to feature counts
- Improves robustness on unseen data