SVM & Naïve Bayes - Complete Solutions

Theoretical Questions

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

SVM is a supervised machine learning algorithm used for classification and regression. It finds the optimal hyperplane that maximizes the margin between different classes by identifying support vectors—data points closest to the decision boundary.

2. What is the difference between Hard Margin and Soft Margin SVM?

Hard Margin SVM: Assumes data is linearly separable; no misclassification allowed. Cannot handle outliers.

Soft Margin SVM: Allows some misclassification using the C parameter, making it robust to outliers and applicable to real-world data that isn't perfectly separable.

3. What is the mathematical intuition behind SVM?

SVM solves the optimization problem: minimize $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$ subject to $y_i(w^T x_i + b) \geq 1 - \xi_i$. It maximizes the margin while minimizing classification errors through constraint satisfaction.

4. What is the role of Lagrange Multipliers in SVM?

Lagrange multipliers convert the constrained optimization problem into an unconstrained one, enabling efficient computation of the optimal hyperplane without explicitly solving high-dimensional problems.

5. What are Support Vectors in SVM?

Support vectors are the training examples closest to the decision boundary. They define the margin and directly influence the position of the hyperplane. Removing non-support vectors doesn't change the model.

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

SVC is the classification variant of SVM that categorizes data into discrete classes by finding an optimal hyperplane that separates classes with maximum margin.

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

SVR is the regression variant of SVM that predicts continuous values. It uses an epsilon-insensitive loss function to fit a regression line within an epsilon band around the data.

8. What is the Kernel Trick in SVM?

The kernel trick maps data into higher-dimensional space implicitly without explicit transformation, enabling SVM to handle non-linear data efficiently using the dot product formula: $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$.

9. Compare Linear Kernel, Polynomial Kernel, and RBF Kernel

Aspect	Linear	Polynomial	RBF
Formula	$x_i \cdot x_j$	$(x_i \cdot x_j + c)^d$	$\exp(-\gamma \|x_i - x_j\|^2)$
Use Case	Linearly separable data	Moderate non-linearity	High non-linearity
Complexity	Low	Medium	Higher
Performance	Fast	Moderate	Good for complex patterns

10. What is the effect of the C parameter in SVM?

C controls the trade-off between maximizing margin and minimizing training error. **High C**: stricter penalty for misclassification, risk of overfitting. **Low C**: allows more misclassification, better generalization.

11. What is the role of the Gamma parameter in RBF Kernel SVM?

Gamma determines the influence of each training example. **High gamma**: only nearby points matter, causes overfitting. **Low gamma**: broader influence, smoother decision boundary.

12. What is the Naïve Bayes classifier, and why is it called "Naïve"?

Naïve Bayes is a probabilistic classifier based on Bayes' Theorem. It's called "Naïve" because it assumes all features are conditionally independent given the class label, which is unrealistic but simplifies computation.

13. What is Bayes' Theorem?

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Where $P(A|B)$ is the posterior probability, $P(B|A)$ is the likelihood, $P(A)$ is the prior, and $P(B)$ is the evidence.

14. Explain the differences between Gaussian, Multinomial, and Bernoulli Naïve Bayes

Type	Data Type	Distribution	Use Case
Gaussian	Continuous	Normal distribution	Numeric features (age, height)
Multinomial	Discrete counts	Multinomial	Text classification, word counts
Bernoulli	Binary/Boolean	Bernoulli	Binary features, presence/absence

15. When should you use Gaussian Naïve Bayes over other variants?

Use Gaussian Naïve Bayes when working with continuous numerical features that approximately follow a normal distribution, such as measurements, scores, or real-valued data.

16. What are the key assumptions made by Naïve Bayes?

1. Conditional independence of features given the class
2. All features are equally important
3. Feature values are independent
4. No missing data or specific value distribution required

17. What are the advantages and disadvantages of Naïve Bayes?

Advantages: Fast, simple, works well with small datasets, requires less training data, probabilistic predictions.

Disadvantages: Assumes conditional independence (unrealistic), struggles with highly dependent features, moderate accuracy on complex datasets, biased probability estimates.

18. Why is Naïve Bayes a good choice for text classification?

Naïve Bayes handles high-dimensional sparse data efficiently, is fast to train, naturally suited for word count features, and performs surprisingly well despite its simplicity assumption on text data.

19. Compare SVM and Naïve Bayes for classification tasks

Aspect	SVM	Naïve Bayes
Speed	Slower training	Fast training
Data requirement	Needs more data	Works with less data
Accuracy	Generally higher	Moderate
Probability	No probabilities	Probabilistic
Feature Independence	No assumption	Assumes independence
Scalability	Moderate	Highly scalable

20. How does Laplace Smoothing help in Naïve Bayes?

Laplace smoothing adds a small constant (typically 1) to feature counts to avoid zero probabilities when a feature-class combination doesn't appear in training data, preventing numerical instability and improving robustness.

Practical Implementation Guide

Questions 21-26: Theoretical Programming Tasks

Q21: Train SVM on Iris → Use `sklearn.svm.SVC` with default kernel, load iris dataset, fit model, evaluate with accuracy score.

Q22: SVM Linear vs RBF on Wine → Train two SVC models, compare accuracies using cross_val_score or train-test split.

Q23: SVR on Housing Data → Use `sklearn.svm.SVR`, train on housing dataset, evaluate using MSE with `mean_squared_error()`.

Q24: SVM with Polynomial Kernel → Visualize decision boundary using meshgrid, plot decision regions with 2D features.

Q25: Gaussian Naïve Bayes on Breast Cancer → Use `sklearn.naive_bayes.GaussianNB`, train, evaluate accuracy.

Q26: Multinomial Naïve Bayes for Text → Use `sklearn.naive_bayes.MultinomialNB`, apply TfidfVectorizer on 20Newsgroups dataset.

Questions 27-46: Practical Advanced Tasks

Q27: SVM with varying C values → Loop through C values, visualize decision boundaries using contour plots.

Q28: Bernoulli Naïve Bayes → Use `sklearn.naive_bayes.BernoulliNB` for binary feature datasets.

Q29: Feature Scaling Comparison → Scale features using StandardScaler, compare SVM accuracy with/without scaling.

Q30: Laplace Smoothing Effect → Train GaussianNB with/without Laplace smoothing (alpha parameter), compare predictions.

Q31: GridSearchCV Hyperparameter Tuning → Use GridSearchCV to optimize C, gamma, and kernel parameters for SVM.

Q32: Class Weighting for Imbalanced Data → Use `class_weight='balanced'` parameter in SVC to improve accuracy on imbalanced datasets.

Q33: Spam Detection → Build email spam classifier using Multinomial Naïve Bayes with TF-IDF features.

Q34: SVM vs Naïve Bayes Comparison → Train both on same dataset, compare accuracy metrics comprehensively.

Q35: Feature Selection for Naïve Bayes → Use SelectKBest or RFE, compare performance before and after selection.

Q36: Multi-class Strategies → Compare One-vs-Rest and One-vs-One strategies on Wine dataset using `decision_function_shape` parameter.

Q37: Multiple Kernels Comparison → Train SVM with Linear, Polynomial, RBF kernels on Breast Cancer dataset, compare accuracies.

Q38: Stratified K-Fold Cross-Validation → Use StratifiedKFold, compute average accuracy across folds for SVM.

Q39: Prior Probabilities Tuning → Train Naïve Bayes with different prior probabilities, compare performance.

Q40: Recursive Feature Elimination → Apply RFE with SVM, compare accuracy with all features.

Q41: Precision, Recall, F1-Score → Evaluate SVM using classification_report instead of single accuracy metric.

Q42: Log Loss for Naïve Bayes → Use `log_loss()` to evaluate Naïve Bayes probabilistic predictions.

Q43: Confusion Matrix Visualization → Create confusion matrix with seaborn heatmap for SVM predictions.

Q44: SVR with MAE → Evaluate Support Vector Regressor using MAE metric instead of MSE.

Q45: ROC-AUC Score for Naïve Bayes → Calculate and plot ROC curve, compute AUC score.

Q46: Precision-Recall Curve → Visualize precision-recall trade-off for SVM classifier.

Key Implementation Tips

1. Always normalize/scale features before SVM training
2. Use cross-validation for robust performance evaluation
3. For imbalanced datasets, use class weights or stratified splitting
4. Tune hyperparameters systematically using GridSearchCV or RandomizedSearchCV
5. For text data, use TfidfVectorizer or CountVectorizer before Naïve Bayes

6. Apply Laplace smoothing (alpha > 0) to handle zero probabilities
7. Choose kernel based on data characteristics: Linear for simple cases, RBF for complex non-linear patterns
8. Use probability calibration (CalibratedClassifierCV) if probability estimates are important
9. Compare multiple metrics (accuracy, precision, recall, F1, AUC) for comprehensive evaluation
10. Visualize decision boundaries for 2D datasets to understand model behavior