UNIT–2 : Regression, Bayesian Learning and Support Vector Machines (PYQ Solution)

AKTU · Machine Learning Techniques (BCDS062) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. What is a Bayes Optimal Classifier? (May 2026)

The Bayes optimal classifier assigns a new instance the class that has the highest posterior probability after combining the predictions of all hypotheses, each weighted by its posterior: v = argmax over vj of Σ P(vj | hi) P(hi | D). No other classifier using the same hypothesis space and prior knowledge can do better on average, so it is the best possible but costly benchmark.

Q2. Explain Bayesian Belief Networks. (May 2026)

A Bayesian belief network is a directed acyclic graph in which nodes are random variables and edges show conditional dependence. Each node stores a conditional probability table given its parents. The joint probability is P(x1… xn) = Π P(xi | parents(xi)). It models uncertain knowledge compactly and supports inference, e.g. in medical diagnosis.

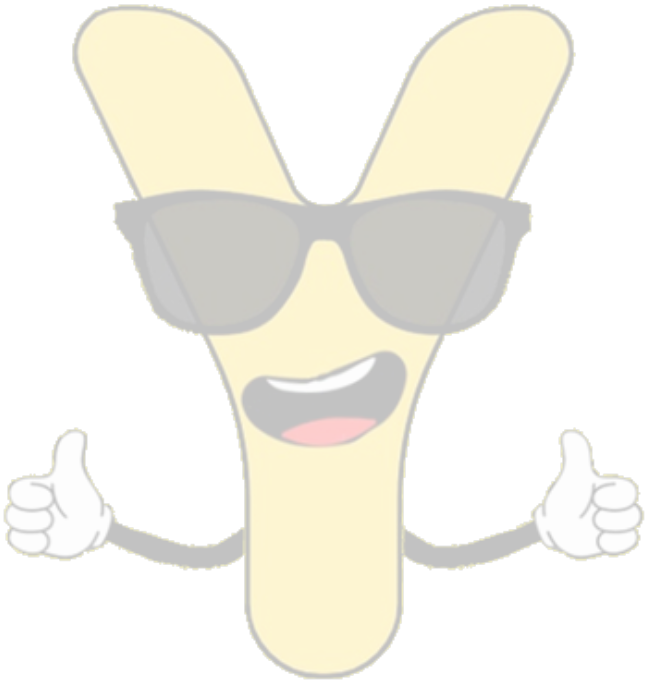
Q3. State any two applications of Logistic Regression. Explain the role of the sigmoid function. (Jan 2026, Jan 2025)

Applications: spam detection and disease diagnosis (also credit default prediction). Logistic regression uses the sigmoid $\sigma(z) = 1 / (1 + e^{-z})$ to squash the linear output z = wx + b into the range (0, 1), which is read as a probability. A threshold (usually 0.5) then gives the class, and the smooth curve makes gradient-based training possible.

Q4. Define Support Vector Machine. What are support vectors and the

significance of the decision surface? (Jan 2026, Jan 2025, Jan 2024)

An SVM is a supervised classifier that finds the hyperplane separating classes with the maximum margin. Support vectors are the training points closest to the hyperplane; they alone define its position, and removing other points changes nothing. The decision surface (w·x + b = 0) is the boundary that decides the class of a new point; a larger margin gives better generalisation.



SECTION B Answers (≈250 Words Each)

Q1. What are Linear and Logistic Regression? How do they differ in output and application? Describe Linear Regression and its assumptions. (May 2026, Jan 2026)

Linear regression models a continuous target as a linear function of inputs: y = β0 + β1x + ε. Coefficients are found by **least squares**, minimising Σ(yi − ŷi)². **Logistic regression** passes the linear combination through the sigmoid, P(y=1|x) = 1 / (1 + e^-(β0 + β1x)), to predict the probability of a class; it is fitted by maximum likelihood.

Assumptions of linear regression:

- **Linearity:** the relationship between X and y is linear.
- **Independence** of errors (no autocorrelation).
- **Homoscedasticity:** constant variance of errors.
- **Normality** of error terms.
- **No multicollinearity** among predictors (for multiple regression).

Basis	Linear	Logistic
Output	Continuous value	Probability, then class (0/1)
Function	Straight line	S-shaped sigmoid
Estimation	Least squares	Maximum likelihood
Loss	MSE	Log loss (cross-entropy)
Application	House price, sales	Spam, diagnosis, churn

Conclusion: linear regression is used for prediction of quantities; logistic regression for classification.

Q2. Differentiate between Simple and Multiple Linear Regression. How does including multiple features affect the model? (Jan 2025)

Simple linear regression uses one independent variable: y = β0 + β1x. **Multiple linear regression** uses several: y = β0 + β1x1 + β2x2 + … + βnxn.

Basis	Simple	Multiple
Predictors	One	Two or more
Geometry	Line	Plane/hyperplane
Example	Salary from experience	Price from area, rooms, location
Complexity	Low	Higher

Effect of multiple features:

- Usually **higher accuracy** because more information is used, and each coefficient shows the effect of one variable holding others fixed.
- Risk of **overfitting** if features are many compared with samples.
- **Multicollinearity** makes coefficients unstable.
- Needs feature scaling, selection and more data; use **adjusted R^2** to judge.

Conclusion: extra relevant features improve the fit, but irrelevant or correlated ones harm generalisation.

Q3. Explain the Naive Bayes classifier with an example. (Jan 2026)

Naive Bayes applies **Bayes' theorem**, $P(c \mid x) = P(x \mid c)\,P(c) / P(x)$, with the "naive" assumption that features are **conditionally independent** given the class: $P(x1 \ldots xn \mid c) = \Pi\, P(xi \mid c)$. The predicted class is argmax over c of $P(c)\,\Pi\, P(xi \mid c)$.

Example: training set of 10 mails, 4 spam and 6 not-spam. The word "offer" appears in 3 of the spam and 1 of the non-spam mails. For a mail containing "offer": spam score = (4/10) × (3/4) = 0.30; non-spam score = (6/10) × (1/6) = 0.10. Since 0.30 > 0.10, it is classified as **spam** (posterior 0.30/0.40 = 75%).

It is fast, works with little data and handles text well.

Q4. Discuss the properties and issues of Support Vector Machines. (Jan 2026)

Properties:

- Finds the **maximum-margin hyperplane**, which gives good generalisation.
- Depends only on **support vectors**, so the model is compact.
- **Kernel trick** (linear, polynomial, RBF) handles non-linear data.

- Convex optimisation gives a **global optimum**, with no local minima.
- Works well in high-dimensional spaces.

Issues:

- Training is slow on very large datasets (about $O(n^2)$ to $O(n^3)$).
- Choice of **kernel and parameters** (C, γ) is difficult and needs tuning.
- Sensitive to noise and overlapping classes; the soft margin (C) is a compromise.
- No direct probability output.
- Needs feature scaling; hard to interpret; binary by nature (multi-class needs one-vs-rest).

Q5. Explain Maximum Likelihood and Least Squared Error hypothesis with example. (Jan 2024)

The **Maximum Likelihood (ML) hypothesis** is $h_{ML} = \text{argmax}$ over h of $P(D \mid h)$, the hypothesis under which the observed data is most probable.

Assume the target value $d_i = f(x_i) + e_i$, where noise e_i is Gaussian with zero mean and variance σ^2. Then $P(D \mid h) = \Pi\ (1/\sqrt{(2\pi\sigma^2)})\ \exp(-(d_i - h(x_i))^2 / 2\sigma^2)$. Taking the log and dropping constants:

$h_{ML} = \text{argmax}\ \Sigma\ -(d_i - h(x_i))^2 = \text{argmin}\ \Sigma\ (d_i - h(x_i))^2$

So under Gaussian noise the **maximum likelihood hypothesis is the one that minimises the sum of squared errors**, which is the least-squares hypothesis.

Example: for points (1, 2), (2, 4), (3, 6) the line $h(x) = 2x$ gives squared error 0, hence maximum likelihood, while $h(x) = 1.5x$ gives error $0.25 + 1 + 2.25 = 3.5$.

SECTION C Answers (≈500 Words Each)

Q1. Describe the Naive Bayes classifier and discuss its assumptions, advantages and limitations. (May 2026)

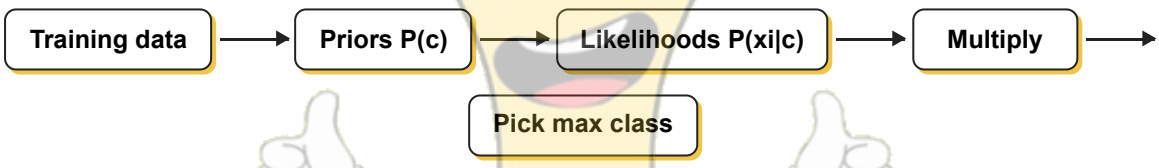
The **Naive Bayes classifier** is a probabilistic supervised classifier based on **Bayes' theorem**:

$P(c \mid x1\ldots xn) = P(c) \times P(x1\ldots xn \mid c) / P(x1\ldots xn)$

With the independence assumption, $P(x1\ldots xn \mid c) = \Pi\, P(xi \mid c)$, so the **MAP decision** is: class = argmax over c of $P(c)\, \Pi\, P(xi \mid c)$. The denominator is the same for all classes and is ignored.

Steps:

1. Compute the prior P(c) for each class from training data.
2. Compute the likelihood P(xi | c) of each feature value in each class (Gaussian for continuous features).
3. Multiply the prior by the likelihoods for the new instance.
4. Choose the class with the largest value.



Naive Bayes workflow

Assumptions:

- Features are **conditionally independent** given the class.
- All features contribute equally and are relevant.
- Training data represents the true distribution.

Advantages:

- Simple, very fast to train and predict.
- Needs little training data; scales to high dimensions.
- Works well for text classification and spam filtering.
- Handles missing values and multi-class naturally.

Limitations:

- Independence assumption rarely holds, which reduces accuracy.

- **Zero-frequency problem:** an unseen value gives probability 0 (fixed by Laplace smoothing).
- Probability estimates are often poorly calibrated.
- Cannot learn feature interactions.

Conclusion: despite its naive assumption, Naive Bayes is a strong, cheap baseline.

Q2. What is a Support Vector Machine, and how does it construct a hyperplane (decision boundary)? Categorise the popular kernels. (May 2026, Jan 2024)

An **SVM** is a supervised learning algorithm for classification and regression that finds the hyperplane separating classes with the **largest margin**.

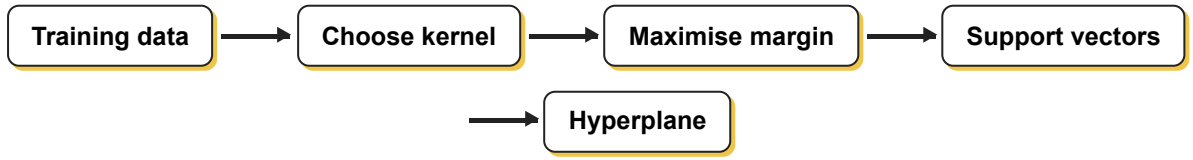
Hyperplane: $w \cdot x + b = 0$, where w is the normal vector. A point is class +1 if $w \cdot x + b \geq +1$ and class −1 if $w \cdot x + b \leq -1$. The two margin lines $w \cdot x + b = \pm 1$ pass through the **support vectors**, and the margin width is $2 / \|w\|$.

Construction: maximise the margin, i.e. solve
minimise $\frac{1}{2}\|w\|^2$ subject to $y_i(w \cdot x_i + b) \geq 1$ for all i.
This is a convex quadratic problem solved with **Lagrange multipliers** α_i; $w = \Sigma\, \alpha_i y_i x_i$, and only support vectors have $\alpha_i > 0$. For noisy data the **soft margin** adds slack variables ξ_i and a penalty C: minimise $\frac{1}{2}\|w\|^2 + C \Sigma\, \xi_i$.

Non-linear data: the **kernel trick** replaces $x \cdot x'$ with $K(x, x')$, implicitly mapping data to a higher-dimensional space where it is linearly separable.

Popular kernels:

Kernel	Formula	Use
Linear	$x \cdot x'$	Linearly separable, text
Polynomial	$(\gamma\, x \cdot x' + r)^d$	Curved boundaries, images
RBF (Gaussian)	$\exp(-\gamma\|x - x'\|^2)$	General-purpose non-linear
Sigmoid	$\tanh(\gamma\, x \cdot x' + r)$	Neural-network-like



SVM hyperplane construction

Conclusion: SVM gives a unique maximum-margin boundary defined by support vectors, and kernels extend it to non-linear problems.

Q3. Apply the Naive Bayes classifier to the weather data and predict Play Tennis for {Sunny, Cool, High, Strong}. (Jan 2025)

Data has 8 days: Play = Yes on days 2, 4, 5, 7 (4 days) and No on days 1, 3, 6, 8 (4 days).

Priors: P(Yes) = 4/8 = 0.5, P(No) = 4/8 = 0.5.

Likelihoods:

Feature value	Given Yes	Given No
Outlook = Sunny	0/4	2/4 (days 1, 6)
Temperature = Cool	1/4 (day 4)	1/4 (day 3)
Humidity = High	2/4 (days 2, 7)	1/4 (day 1)
Wind = Strong	2/4 (days 4, 7)	1/4 (day 6)

Posterior scores:

- Yes: 0.5 × 0 × 1/4 × 2/4 × 2/4 = **0**
- No: 0.5 × 2/4 × 1/4 × 1/4 × 1/4 = 0.5 × 0.5 × 0.015625 = **0.00391**

With Laplace smoothing (add 1 to each count and the number of possible values to the denominator: 3 for Outlook and Temperature, 2 for Humidity and Wind):

- Yes: 0.5 × (1/7) × (2/7) × (3/6) × (3/6) = 0.0051
- No: 0.5 × (3/7) × (2/7) × (2/6) × (2/6) = 0.0068

In both cases score(No) > score(Yes).

Answer: the person will **not play tennis** (Play = No).

Q4. Fit a linear regression model for (1, 1.5), (2, 3.0), (3, 4.5), (4, 6.0) and predict y for x = 5. (Jan 2025)

Here n = 4. $\bar{x}$ = (1+2+3+4)/4 = 2.5 and $\bar{y}$ = (1.5+3+4.5+6)/4 = 3.75.

x	y	$x - \bar{x}$	$y - \bar{y}$	$(x-\bar{x})(y-\bar{y})$	$(x-\bar{x})^2$
1	1.5	−1.5	−2.25	3.375	2.25
2	3.0	−0.5	−0.75	0.375	0.25
3	4.5	0.5	0.75	0.375	0.25
4	6.0	1.5	2.25	3.375	2.25
Sum				7.5	5.0

Slope $b1 = 7.5 / 5 =$ **1.5**.
Intercept $b0 = \bar{y} - b1\,\bar{x} = 3.75 - 1.5 \times 2.5 =$ **0**.

Model: $\hat{y} = 1.5x$. All points lie exactly on this line, so the error is zero.

Prediction: for $x = 5$, $\hat{y} = 1.5 \times 5 =$ **7.5**.

Q5. Explain the "Concept Learning" task giving an example. (Jan 2024)

Concept learning means inferring a Boolean-valued function (a concept) from training examples labelled positive or negative; it is a search through a hypothesis space for the hypothesis that fits the data.

Task elements: instance space X, target concept $c: X \rightarrow \{0, 1\}$, training examples D, hypothesis space H.

Example: EnjoySport. Attributes: Sky, AirTemp, Humidity, Wind, Water, Forecast. A hypothesis is a conjunction of constraints, with "?" (any value) and "∅" (no value).

- Example 1: (Sunny, Warm, Normal, Strong, Warm, Same) → Yes.
- Example 2: (Sunny, Warm, High, Strong, Warm, Same) → Yes.
- Example 3: (Rainy, Cold, High, Strong, Warm, Change) → No.

Find-S algorithm: start with the most specific hypothesis (∅, ∅, ∅, ∅, ∅, ∅). For each positive example, generalise each attribute that does not match to "?". After example 1: (Sunny, Warm, Normal, Strong, Warm, Same). After example 2: (Sunny, Warm, ?, Strong, Warm, Same). Negative examples are ignored.

Candidate Elimination keeps the **version space** between the specific boundary S and general boundary G, shrinking it with each example.

Conclusion: concept learning generalises from examples by searching from specific to general hypotheses.

Q6. Differentiate between the Naive Bayes classifier and Bayesian belief networks. Give an application of Bayesian belief networks. (Jan 2024)

Both are probabilistic graphical models based on Bayes' theorem, but they differ in how dependencies are handled.

Basis	Naive Bayes	Bayesian belief network
Independence	All features independent given class	Only conditional independence given parents is assumed
Structure	Fixed star: class → each feature	General DAG, learned or specified by experts
Dependencies	No feature interactions	Models dependencies among features
Joint probability	P(c) × product of P(xi given c)	Product of P(xi given parents of xi)
Complexity	Simple, fast	More complex to build and infer
Accuracy	Lower when features correlated	Usually higher
Purpose	Classification	Reasoning, diagnosis and classification



Naive Bayes structure: class points to features

A belief network holds a **conditional probability table (CPT)** in each node, and inference computes the posterior of any variable given observed evidence. Naive Bayes is a special case of a belief network.

Application: medical diagnosis. Nodes such as Smoking, Cancer, Bronchitis, Cough and X-ray result are linked: Smoking → Cancer, Smoking → Bronchitis, Cancer → X-ray, Bronchitis → Cough. Given symptoms (cough, abnormal X-ray), the network computes the probability of cancer. Other uses: fault diagnosis, spam filtering and risk assessment.

Conclusion: belief networks relax the naive assumption and give richer reasoning at higher cost.

