5705.25 - Hagfræðilig læring H25

November 22, 2025

Contents

1 Introduction

2 Statistical Learning

Statistical learning is concerned with understanding and estimating the relationship between a target variable Y and one or more predictors $X = (X_1, X_2, \ldots, X_p)$. We typically assume the model:

$$Y = f(X) + \epsilon$$

where $f(X)$ is the systematic part of the relationship, and ϵ is random noise with mean zero.

2.1 Supervised Learning

In supervised learning, both the predictors X and the response Y are observed, and the goal is to estimate the mapping $f : X \mapsto Y$.

2.1.1 Types of Supervised Learning

Depending on the type of Y, we distinguish between:

Type of Y	Task	Example
Continuous (quantitative)	Regression	Predicting sales, temperature, price
Discrete (qualitative)	Classification	Predicting spam vs. non-spam, disease vs. healthy

2.1.2 Example: Predicting Sales

Suppose we want to predict sales Y using advertising budgets in three media channels:

$$X_1 = \text{TV}, \quad X_2 = \text{Radio}, \quad X_3 = \text{Newspaper}.$$

We assume

$$Y = f(X_1, X_2, X_3) + \epsilon$$

and start with a linear model:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3.$$

2.1.3 Estimating f Using Squared Loss

To estimate f, we minimize the squared loss:

$$L(y_i, \hat{f}(x_i)) = (y_i - \hat{f}(x_i))^2,$$

and the total (mean) squared error:

$$\text{MSE}(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2.$$

The optimal parameters β_j are found by minimizing the MSE, leading to the ordinary least squares (OLS) solution:

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T y.$$

2.2 Unsupervised Learning

In unsupervised learning, only the predictors X are observed; there is no corresponding response variable Y. The goal is to discover structure or patterns within the data.

2.2.1 Clustering

A common example is **clustering**, where observations are grouped based on similarity among predictors $X_1, X_2, \ldots$. For instance, clustering points (x_1, x_2) may reveal natural groupings or clusters within a dataset.

2.2.2 Outcomes

In contrast to supervised learning:

- Continuous outcomes correspond to **regression**.
- Discrete outcomes correspond to **classification**.

2.3 Least Squares Regression

In least squares regression, we assume a model such as:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2,$$

which can capture some curvature or nonlinearity.

While more flexible models can better fit training data, they also introduce new issues such as **overfitting**. We can measure performance using the Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2.$$

The error can be decomposed into:

- **Training error**: computed on the same data used to fit the model.
- **Test (validation) error**: computed on unseen data.

A model that fits the training data too closely may not generalize well to new data this is overfitting. The aim is to find a balance between underfitting and overfitting.

As a general rule of thumb, more flexible methods tend to perform better than inflexible ones, but they must be validated using separate **validation** and **test** sets.

2.4 Bias–Variance Tradeoff

The expected prediction error for a given x_0 can be decomposed as:

$$\mathbb{E}\big[(Y - \hat{f}(x_0))^2\big] = \big[\text{Bias}(\hat{f}(x_0))\big]^2 + \text{Var}(\hat{f}(x_0)) + \sigma_\epsilon^2,$$

where:

- $\sigma_\epsilon^2 = \text{Var}(\epsilon)$ is the irreducible error,
- $\text{Bias}(\hat{f}(x_0)) = \mathbb{E}[\hat{f}(x_0)] - f(x_0)$,
- $\text{Var}(\hat{f}(x_0))$ is the variability of the estimate.

We cannot reduce the irreducible error, but we can trade off bias and variance. Accuracy depends on finding the right balance:

- High bias $\Rightarrow$ underfitting.
- High variance $\Rightarrow$ overfitting.

Interpretability often decreases as model flexibility increases.

2.5 Classification

In classification, the goal is to predict a qualitative response Y taking values in a finite set of classes $\{1, 2, \ldots, K\}$.

2.5.1 K-Nearest Neighbours (KNN)

Given a new observation x_0, KNN identifies the K training points closest to x_0 (denoted $\mathcal{N}_0$) and estimates:

$$\Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(Y_i = j),$$

where $I(\cdot)$ is the indicator function.

The predicted class is:

$$\hat{y}_0 = \arg\max_j \Pr(Y = j \mid X = x_0).$$

2.5.2 Classification Error

The classification error rate is:

$$\text{Error} = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(Y_i \neq \hat{Y}_i).$$

2.5.3 Bayes Classifier

The Bayes classifier assigns each observation to the most probable class:

$$\hat{y}_{\text{Bayes}} = \arg\max_j \Pr(Y = j \mid X = x),$$

which theoretically minimizes the classification error. In practice, methods like KNN approximate this boundary, sometimes sacrificing perfect classification to reduce noise.

2.5.4 Key Idea

A fundamental rule to remember is:

$$Y = f(X) + \epsilon.$$

This holds true for both regression and classification settings the goal of statistical learning is to estimate f as accurately as possible.

3 Regression

Regression analysis is a fundamental statistical technique used to model and understand the relationship between a response variable Y and one or more explanatory variables $X_1, X_2, \ldots, X_p$. The goal is to estimate a function $f(X)$ that captures the systematic relationship between X and Y, while accounting for random variation (error).

3.1 Additive Error Models

We often assume an additive model of the form:

$$Y = f(X) + \varepsilon,$$

where ε is a random error term with

$$\mathbb{E}[\varepsilon] = 0, \quad \text{and} \quad \text{Var}(\varepsilon) = \sigma^2.$$

The term ε captures the **irreducible error**—random variation that cannot be explained by the model, even with perfect knowledge of f.

A common goal is to estimate the **regression function**:

$$f(X) = \mathbb{E}[Y|X].$$

This represents the expected value of Y for a given value of X.

3.2 Estimating $f(X)$

Given a set of training data $(x_1, y_1), \ldots, (x_n, y_n)$, we want to construct an estimate $\hat{f}(X)$ that minimizes the expected prediction error:

$$\mathbb{E}\left[(Y - \hat{f}(X))^2\right].$$

3.2.1 K-Nearest Neighbour (KNN) Regression

A simple non-parametric estimator is the **K-nearest neighbours (KNN)** regression:

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_K(x_0)} y_i,$$

where $\mathcal{N}_K(x_0)$ is the set of the K closest training points to x_0.

Example: To predict $f(4)$, we take the average of the Y_i values for the K data points whose X_i values are closest to 4:

$$\hat{f}(4) = \text{avg}\{Y_i : X_i \in \mathcal{N}_K(4)\}.$$

Advantages and Limitations:

- Performs well when the number of predictors p is small and sample size n is large.
- Performs poorly when p is large due to the **curse of dimensionality**—as p increases, points become sparse, and distances between them lose meaning.

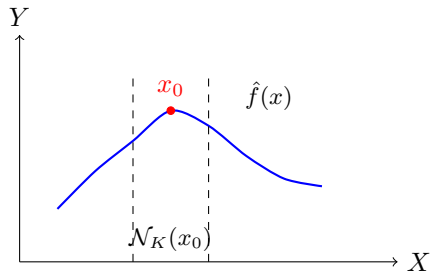


Figure: KNN regression—averaging points within a neighbourhood around x_0.

3.3 Parametric Models and Linear Regression

To overcome the limitations of non-parametric methods like KNN, we often use a **parametric model** that assumes a specific functional form for $f(X)$:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Even though this form is rarely exactly true, it provides a simple and interpretable approximation.

3.3.1 Simple Linear Regression

For a single predictor variable:

$$Y = \beta_0 + \beta_1 X + \varepsilon.$$

The goal is to estimate β_0 and β_1 using the training data, such that the **residual sum of squares (RSS)** is minimized:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2.$$

Setting the partial derivatives of RSS with respect to β_0 and β_1 equal to zero yields the least-squares estimates.

Population Regression Line Example:

$$Y = 2 + 3X + \varepsilon \quad \Rightarrow \quad f(X) = 2 + 3X.$$

Here, $\beta_0 = 2$ and $\beta_1 = 3$ are unbiased estimates of the true relationship between X and Y.

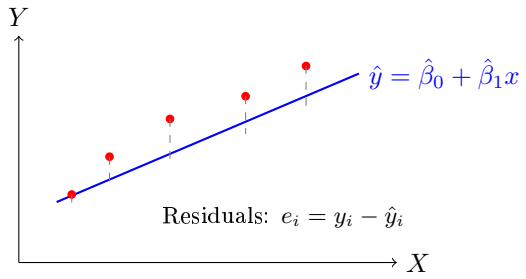


Figure: Simple linear regression with fitted line and residuals.

3.3.2 Standard Errors and Confidence Intervals

The variance of the error term is denoted $\sigma^2 = \text{Var}(\varepsilon)$. The estimated variance from data is:

$$\widehat{\sigma}^2 = \frac{\text{RSS}}{n-2}.$$

The **standard error** of $\hat{\beta}_1$ measures the uncertainty of the slope estimate. A 95% confidence interval for β_1 is:

$$[\hat{\beta}_1 - 2 \cdot SE(\hat{\beta}_1),\ \hat{\beta}_1 + 2 \cdot SE(\hat{\beta}_1)].$$

3.4 Hypothesis Testing

We often test whether there is a significant relationship between X and Y:

$$H_0 : \beta_1 = 0 \quad \text{(no relationship)}$$
$$H_a : \beta_1 \neq 0 \quad \text{(some relationship)}$$

The test statistic is:

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)},$$

which follows a t-distribution with $n-2$ degrees of freedom under H_0.

The **p-value** is the probability of observing a $|t|$ as large as the one obtained, assuming H_0 is true. A small p-value (typically < 0.05) leads to rejection of H_0.

3.5 Assessing Model Accuracy

3.5.1 Residual Standard Error (RSE)

The RSE provides an estimate of the standard deviation of the residuals:

$$RSE = \sqrt{\frac{RSS}{n-2}}.$$

3.5.2 The R^2 Statistic

The R^2 statistic measures the proportion of variability in Y explained by the model:

$$R^2 = 1 - \frac{RSS}{TSS},$$

where

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2.$$

Here, TSS measures the total variance in Y, and RSS measures the remaining variability after regression.

Properties:

- R^2 typically lies between 0 and 1.
- It can be negative for models that fit worse than a horizontal line at $\bar{y}$.
- R^2 is independent of the units of Y.

3.6 Bias–Variance Tradeoff

When fitting regression models, we balance two sources of error:

$$\text{Expected Test Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}.$$

Parametric models typically have higher bias but lower variance, while non-parametric models like KNN have lower bias but higher variance.

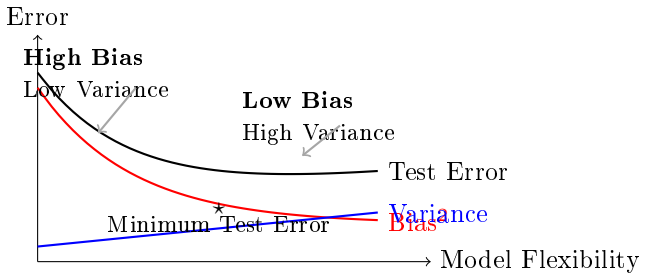


Figure: Bias–variance tradeoff showing how total test error depends on model flexibility.

3.7 Software Note: R and Quarto

When working with R in VSCode, you can compile regression analyses and visualizations within `.qmd` (Quarto) files, which support both code and markdown text for reproducible reports.

4 Multivariate Regression

Multivariate (multiple) linear regression models the relationship between a response variable Y and multiple predictors $X_1, X_2, \ldots, X_p$ using an additive model:

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \varepsilon. \tag{1}$$

The goal is to estimate coefficients β_i that minimize the Residual Sum of Squares (RSS):

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{2}$$

4.1 Design Matrix and Hat Matrix

In matrix notation, the model can be written:

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{3}$$

The fitted values are given by:

$$\hat{\mathbf{y}} = H\mathbf{y}, \quad \text{where } H = X(X^T X)^{-1} X^T. \tag{4}$$

The matrix H is the **hat matrix**, projecting the data onto the regression hyperplane.

4.2 Correlation and Visualization

To assess relationships between predictors and the response, we often compute the sample correlation:

$$\text{Cor}(X, Y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}. \tag{5}$$

Heatmaps of correlation matrices are helpful for identifying linear associations.

4.3 Inference: t-statistics and p-values

To test whether individual predictors have a relationship with Y, we test:

$$H_0 : \beta_j = 0, \qquad H_a : \beta_j \neq 0. \tag{6}$$

We compute the t-statistic:

$$t_j = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}. \tag{7}$$

A small p-value indicates evidence against H_0.

4.4 F-statistic for Overall Significance

The F-test evaluates whether the model provides a better fit than a constant-only model:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0. \tag{8}$$

Large F values suggest the model explains significant variation.

4.5 Variable Selection

When many predictors are available, we must decide which ones to include. Adding more variables always reduces RSS and always increases R^2, so we need principled methods.

4.5.1 Forward Selection

1. Start with the intercept-only model.
2. Add the predictor that most improves the model (largest reduction in RSS or improvement in F-statistic).
3. Continue adding predictors until no significant improvement occurs.

4.5.2 Backward Selection

1. Start with the full model.
2. Remove the predictor with the highest p-value (least significant).
3. Continue removing predictors until all remaining variables are significant.

4.5.3 Stepwise (Mixed) Selection

A combination of forward and backward selection:

- Add the best candidate predictor.
- Then check whether any included predictors should be removed.

This continues until no improvement is possible.

4.5.4 Model Selection Rationale

Variable selection helps:

- Improve interpretability.
- Reduce variance of coefficient estimates.
- Avoid multicollinearity.
- Prevent overfitting.

P-values alone are not sufficient; the overall model validity should be checked using the F-statistic.

4.6 Model Fit: RSE and R^2

Residual Standard Error (RSE) estimates the standard deviation of the error term:

$$RSE = \sqrt{\frac{RSS}{n-p-1}}. \tag{9}$$

R^2 measures the fraction of variance explained:

$$R^2 = 1 - \frac{RSS}{TSS} = \mathrm{Cor}(Y, \hat{Y})^2. \tag{10}$$

Note: R^2 always increases when adding predictors.

4.7 Prediction and Uncertainty

Predictions come with two types of uncertainty:

- **Confidence intervals**: uncertainty in the mean response.
- **Prediction intervals**: includes irreducible error; always wider.

Example in R:

```
model <- lm(y ~ x1 + x2)
predict(model, newdata, interval = "confidence")
```

4.8 Interaction Terms

Including interactions allows the effect of one predictor to depend on another:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon. \tag{11}$$

4.9 Polynomial Regression

A polynomial model remains linear in the parameters:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_d X^d + \varepsilon. \tag{12}$$

4.10 Residuals and Nonlinearity

Examining residuals helps diagnose model misfit. If residuals show curvature, a polynomial or nonlinear transformation may be appropriate.

4.11 Correlated Error Terms

Assuming errors are independent is critical. Correlated errors lead to underestimation of standard errors.

4.12 Heteroskedasticity

If error variance is not constant:

$$\text{Var}(\varepsilon_i) = \sigma_i^2 \neq \sigma^2, \tag{13}$$

then confidence intervals and standard errors become unreliable.

4.13 Outliers and Leverage Points

Outliers have unusual Y values; leverage points have unusual X values. Leverage for observation i:

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum (x_i - \bar{x})^2}. \tag{14}$$

High-leverage points can strongly affect regression lines.

4.14 Collinearity

When predictors are highly correlated, it becomes difficult to distinguish their individual effects. Consequences include:

- Inflated standard errors
- Small t-statistics
- Unstable coefficient estimates

The Variance Inflation Factor (VIF) detects multicollinearity.

5 Classification

Classification problems involve predicting a qualitative (categorical) response variable Y based on one or more predictor variables X.

5.1 Quantitative vs. Qualitative Outcomes

- **Quantitative outcomes:** $Y \in \mathbb{R}$. Typically modeled with regression, minimizing a loss function such as least squares.
- **Qualitative outcomes:** $Y \in \{1, 2, \ldots, K\}$. These are modeled using classification methods like logistic regression or generative models.

5.2 Feature Vector and Notation

Let

$$\mathbf{X} = (X_1, X_2, \ldots, X_p)^T$$

be the feature vector. For example, in the **IRIS dataset**, the features are:

$X_1 = \text{Sepal Length}, \quad X_2 = \text{Sepal Width}, \quad X_3 = \text{Petal Length}, \quad X_4 = \text{Petal Width}$

and the response $Y \in \{\text{setosa, versicolor, virginica}\}$.

The predicted class is:

$$g(\mathbf{x}) = \arg\max_k P(Y = k \mid \mathbf{X} = \mathbf{x})$$

5.3 Logistic Regression (Binary Case)

The probability of class 1 is modeled as:

$$P(Y = 1 \mid X = x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

The log-odds (logit) transformation gives a linear relationship:

$$\log \frac{P(x)}{1 - P(x)} = \beta_0 + \beta_1 x$$

$$\begin{cases} \beta_1 > 0 & \text{increasing } x \text{ increases } P(x) \\ \beta_1 < 0 & \text{increasing } x \text{ decreases } P(x) \end{cases}$$

5.4 Maximum Likelihood Estimation (MLE)

The likelihood function is:

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} P(x_i)^{y_i} \left[1 - P(x_i)\right]^{1 - y_i}$$

MLE maximizes the likelihood (or equivalently, minimizes the negative log-likelihood):

$$\hat{\beta}_0, \hat{\beta}_1 = \arg\max_{\beta_0, \beta_1} L(\beta_0, \beta_1)$$

5.5 Multinomial Logistic Regression

For $K > 2$ classes, probabilities are modeled using the **softmax function**:

$$P(Y = k \mid X = x) = \frac{e^{\beta_{k0} + \beta_k^T x}}{\sum_{j=1}^{K} e^{\beta_{j0} + \beta_j^T x}}$$

The log-odds relative to a baseline class K are:

$$\log \frac{P(Y = k \mid X = x)}{P(Y = K \mid X = x)} = \beta_{k0} + \beta_k^T x$$

5.6 Generative Models

Generative models estimate:

$$P(X \mid Y = k) \quad \text{and} \quad P(Y = k)$$

Then, using Bayes' theorem:

$$P(Y = k \mid X = x) = \frac{P(X = x \mid Y = k) P(Y = k)}{\sum_{j=1}^{K} P(X = x \mid Y = j) P(Y = j)}$$

5.7 Additive Models and Least Squares

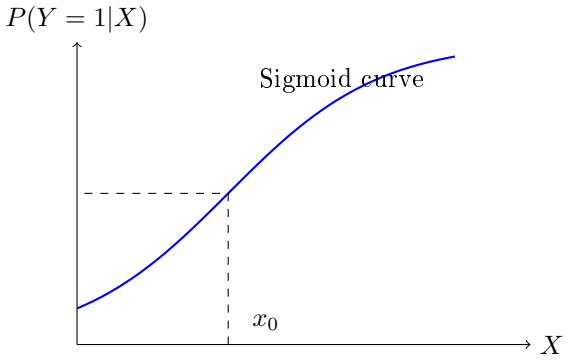
For continuous responses:

$$Y = f_\theta(X) + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

MLE reduces to minimizing the sum of squared errors (RSS):

$$\hat{\theta} = \arg\min_{\theta} \sum_{i} (Y_i - f_\theta(X_i))^2$$

5.8 Visualizing Logistic Regression Decision Boundary



5.9 Generative Models and Advanced Classification

Generative models estimate the joint distribution $P(X, Y)$ and then use Bayes' theorem to compute class probabilities:

$$P(Y = k \mid X = x) = \frac{P(Y = k)P(X = x \mid Y = k)}{P(X = x)}$$

Here:

- $P(Y = k)$ is the prior probability of class k
- $P(X = x \mid Y = k)$ is the class-conditional density
- $P(X = x) = \sum_{i=1}^{K} P(Y = i)P(X = x \mid Y = i)$ by the law of total probability

5.9.1 Linear Discriminant Analysis (LDA)

LDA assumes that:

- Each class k has a Gaussian distribution with mean μ_k and a **common covariance matrix** Σ
- The decision boundary is linear in x

The discriminant function is:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

where π_k is the prior probability of class k.

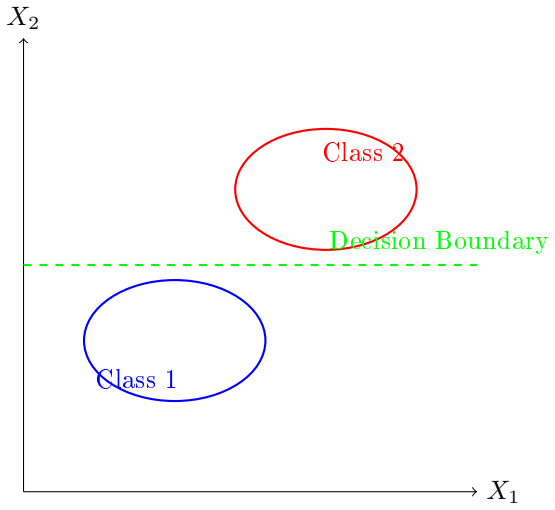
The predicted class is:

$$\hat{y} = \arg\max_k \delta_k(x)$$

We estimate unknown parameters from the data:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i, \quad \hat{\Sigma} = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

Decision Boundary Visualization (2 Classes, 2 Features)



5.9.2 Quadratic Discriminant Analysis (QDA)

QDA relaxes the LDA assumption of common covariance matrices:

$$\Sigma_k \neq \Sigma_j \quad \text{for some } k \neq j$$

The discriminant function becomes quadratic:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k$$

5.9.3 Naive Bayes Classifier

Naive Bayes assumes that features are conditionally independent given the class:

$$P(X_1, X_2, \ldots, X_p \mid Y = k) = \prod_{j=1}^{p} P(X_j \mid Y = k)$$

This allows simple estimation using Gaussian distributions for quantitative features or categorical probabilities for discrete features.

5.9.4 K-Nearest Neighbors (KNN)

- Non-parametric and flexible
- Assigns a class based on the majority vote of the K nearest neighbors in feature space
- More data generally improves performance

5.9.5 Bias-Variance Trade-off and Model Comparison

- LDA: Linear boundaries, low variance, higher bias
- QDA: Quadratic boundaries, higher variance, lower bias
- Logistic Regression: Linear boundary, sensitive to collinearity
- KNN: Highly flexible, low bias, potentially high variance

5.9.6 Performance Metrics

- **Accuracy:** Overall correct classification
- **Sensitivity / Recall:** True positive rate
- **Specificity:** True negative rate
- **ROC curve:** Trade-off between sensitivity and false positive rate

5.9.7 Other Notes

- Poisson regression for count data: $n \sim \text{Poisson}(\mu)$ with $\log(\mu) = X\beta$
- More data allows more flexible models (e.g., QDA, KNN) to perform well without overfitting