LES FACULTÉS
DE L'UNIVERSITÉ
CATHOLIQUE DE LILLE

Inference & Estimation theory

# DECISION TREES & MODELS SELECTION

**Baptiste Mokas**

baptiste.mokas@gmail.com

weeki.io/dynamical-système

linktr.ee/baptistemokas

+33 7 69 08 54 19

# KNOWLEDGE TREE

## Part 1: How could you choose your model ? Let's dive in a decision tree

1.1 The decision tree

*1.2 Number of dependent variables (DV):*

- *One DV => Univariate*
- *Multiple DVs => Multivariate*

*1.3 Nature of the dependent variable (DV):*

- *Quantitative*
- *Qualitative (categorical)*

*1.4 Number of independent variables (IV):*

- *Quantitative:*
- *Continuous*
- *Ordinal*
- *Qualitative (categorical)*

*1.5 Type of question:*

- *Difference*
- *Relationship between two variables*

*1.6 Independence of groups or samples:*

- *Are the groups or samples independent or paired?*
- *Number of groups or samples*
- *Knowledge of the distribution: Is the analysis within the framework of parametric or non-parametric statistics?*

## Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation

2.1 Model Evaluation Metrics

- Guideline
- F1-Score and ROC Curve
- Accuracy, Precision, and Recall
- Confusion Matrix

2.2 Understanding Model Complexity

- Bias, Variance, and Model Complexity
- The Bias-Variance Decomposition
- Example: Bias-Variance Tradeoff

2.3 Deeper Dive into Model Complexity

- Optimism of the Training Error Rate
- Estimates of In-Sample Prediction Error
- The Effective Number of Parameters

2.4 Cross-Validation

- K-Fold Cross-Validation
- Leave-One-Out Cross-Validation (LOOCV)
- The Wrong and Right Way to Do Cross-validation
- Does Cross-Validation Really Work?

## Part 3: Model Selection Techniques and Insights

3.1 Model Selection Techniques

- Akaike Information Criterion (AIC)
- Bayesian Information Criterion (BIC) and The Bayesian Approach
- Minimum Description Length
- Vapnik-Chervonenkis Dimension
- Cross-Validation for Model Selection

3.2 Additional Insights in Model Selection

- General Presentation and Definition
- Distances Between Models
- Criteria for Model Selection

## Part 4: Advanced Techniques in Model Selection

4.1 Regularization Techniques

- L1 and L2 Regularization
- Ridge and Lasso Regression
- Elastic Net

4.2 Feature Selection

- Selecting Relevant Features
- Forward and Backward Selection
- Recursive Feature Elimination (RFE)

4.3 Ensemble Methods

- Bagging and Boosting
- Random Forests
- Model Stacking
- Probability of Preferring One Model Over Another
- Model Selection Algorithm
- Example Treated by Software

# OUR STORY LINE

## Probability and Statistics

| | | | | | |
|---|---|---|---|---|---|
| STEP -1 _ PROGRAM INTRODUCTION<br>STEP 0 _ FOUNDATIONS<br>STEP 1 _ THEORY OF SYSTEMS | STEP 2 _ STOCHASTIC DYNAMICS & PROBABILITY<br>STEP 3 _ DATA OBSERVATION | STEP 4 _ INFERENCE & ESTIMATION THEORY | STEP 5 _ LINEAR MODEL EXAMPLES | STEP 6 _ OTHER MODEL EXAMPLES | STEP 7 _ NON LINEAR MODELS |

# OUR STORY LINE

## Probability and Statistics

### STEP -1 _ PROGRAM INTRODUCTION
- -1 - PROGRAM INTRODUCTION

### STEP 0 _ FOUNDATIONS
- 0.1 - ELEMENTS OF CALCULUS & TOOLS
- 0.2 - EPISTEMOLOGY & THEORY OF KNOWLEDGE

### STEP 1 _ THEORY OF SYSTEMS
- 1.1 - DYNAMICAL SYSTEMS
- 1.2 - COMPLEX ADAPTIVE SYSTEMS

### STEP 2 _ STOCHASTIC DYNAMICS & PROBABILITY
- 2.1 - MEASURE THEORY
- 2.2 - PROBABILITY THEORY
- 2.3 - USUAL PROBABILITY DISTRIBUTIONS
- 2.4 - ASYMPTOTIC STATISTICS
- 2.5 - STOCHASTIC PROCESS & TIME SERIES
- 2.6 - INFORMATION GEOMETRY

### STEP 3 _ DATA OBSERVATION
- 3.1 - DESCRIPTIVE STATISTICS & DATAVIZUALISATION
- 3.2 - EXPLORATORY DATA ANALYSIS

### STEP 4 _ INFERENCE & ESTIMATION THEORY
- 4.1 - PARAMETERS ESTIMATIONS & LEARNING
- 4.2 - EXPERIMENTAL DESIGN & HYPOTHESIS TESTING
- 4.4 - DECISION TREES & MODEL SELECTION
- 4.5 - BAYESIAN INFERENCES

### STEP 5 _ LINEAR MODEL EXAMPLES
- 5.1 - SIMPLE LINEAR REGRESSION
- 5.2 - MULTIPLE LINEAR REGRESSION
- 5.3 - OTHER REGRESSIONS MODELS

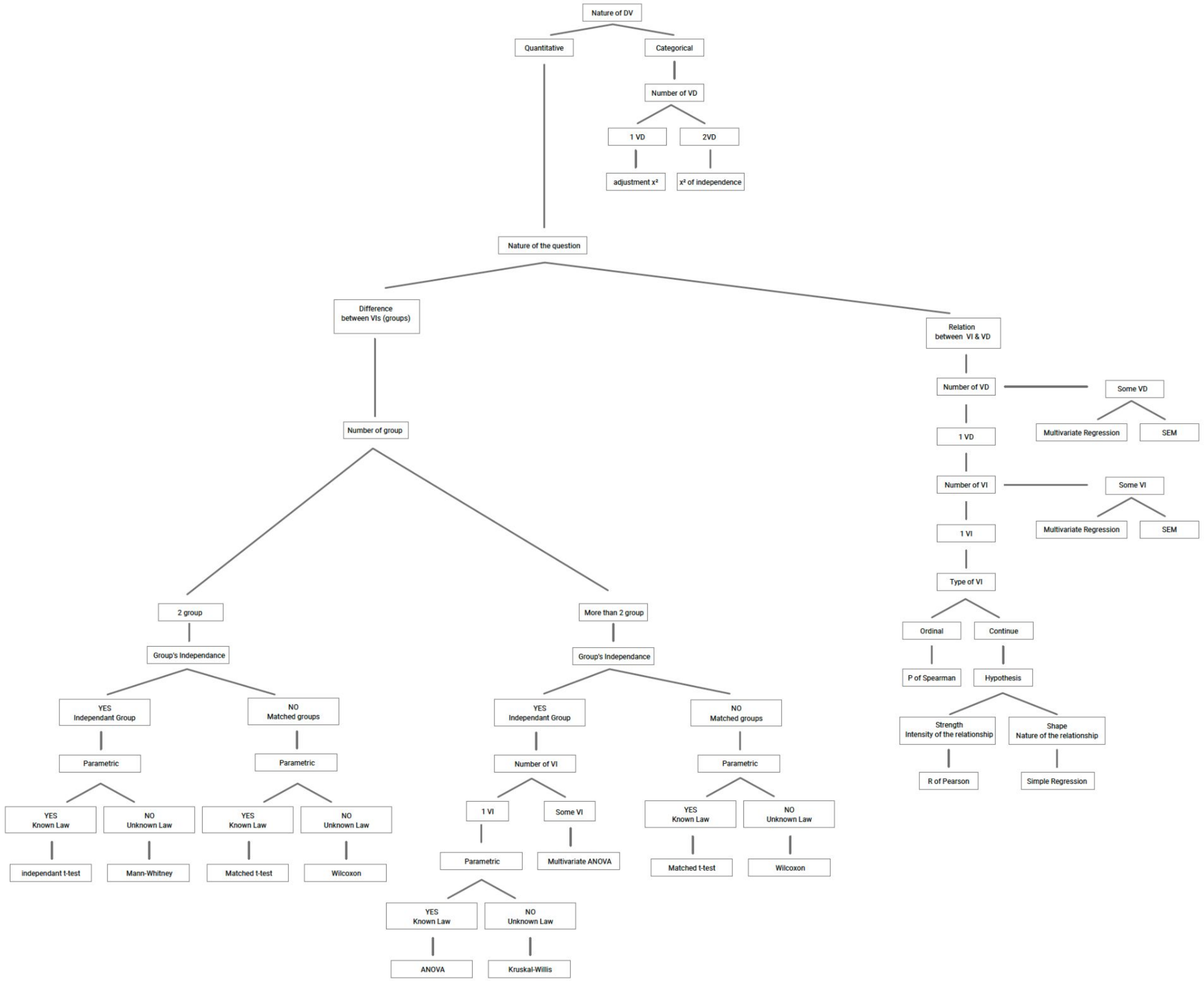
### STEP 6 _ OTHER MODEL EXAMPLES
- 6.1 - USUAL UNIVARIATE TESTING
- 6.2 - USUAL MULTIVARIATE TESTING
- 6.3 - NON PARAMETRIC STATISTICS

### STEP 7 _ NON LINEAR MODELS
- 7.1 - PROBABILISTIC GRAPHICAL MODELS
- 7.2 - PERCOLATION THEORY
- 7.3 - SPATIAL STATISTICS
- 7.4 - EXTREM VALUE THEORY
- 7.5 - AGENT BASED MODELING
- 7.6 - NETWORK DYNAMICS

# Decision Tree

**Part 1: How could you choose your model ? Let's dive in a decision tree**

<u>1.1 The decision tree</u>



# Guideline

Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation

2.1 Model Evaluation Metrics

Split the Data:
- Divide your dataset into two or more subsets, typically a training set and a testing set. In some cases, a validation set is also used for hyperparameter tuning.

Select Evaluation Metrics:
- Choose appropriate evaluation metrics based on the nature of your problem. Common metrics include accuracy, precision, recall, F1 Score, mean squared error, etc. Select metrics that align with your model's objectives.

Model Training:
- Train your statistical model on the training set using the chosen algorithm and parameters.

Model Testing:
- Evaluate the model's performance on the testing set using the selected evaluation metrics. This helps you assess how well your model generalizes to new, unseen data.

Cross-Validation (Optional):
- In cases where you have limited data, use techniques like k-fold cross-validation to assess model stability and generalization.

Analyze and Interpret Results:
- Examine the evaluation metrics and consider what they mean in the context of your problem. Identify areas where the model is performing well and where it needs improvement.

Visualize the Results:
- Use visualization techniques like confusion matrices, ROC curves, and precision-recall curves to gain insights into your model's performance.

Adjust Model and Features:
- If your model's performance is unsatisfactory, consider adjusting model hyperparameters, feature selection, or engineering to improve results.

Evaluate Against Baselines:

- Compare your model's performance against baseline models or simple heuristic approaches. This can provide context for your results.

Consider Business and Domain Context:

- Take into account the broader context of your problem. Understand how model performance aligns with business objectives and the domain you're working in.

Handling Imbalanced Data:

- If your dataset is imbalanced, choose metrics like precision, recall, F1 Score, or area under the precision-recall curve (AUC-PR) to assess model performance more effectively.

Iterate and Improve:

- Depending on the results, iterate the model development process, making necessary changes to improve model performance.

Validation Set and Hyperparameter Tuning:

- Use a validation set for hyperparameter tuning to optimize your model. However, do not over-tune to the point where the model doesn't generalize well.

Consider Real-World Impact:

- Think about the real-world implications of your model's performance, such as the cost of false positives or false negatives in your specific application.

Documentation and Reporting:

- Document your evaluation process, results, and any actions taken to improve the model. Clear reporting helps communicate findings to stakeholders.

# F1-Score and ROC Curve

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.1 Model Evaluation Metrics

The F1Score is a commonly used metric in binary classification that provides a balanced measure of both precision and recall. It is particularly useful when there is an imbalance between the two classes (i.e., when one class is much more frequent than the other). The F1 Score is the harmonic mean of precision and recall and combines both metrics into a single value.
Mathematics:

The F1 Score is calculated as follows:

$$F1\ \text{Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:
- Precision is the ratio of true positive predictions to the total number of positive predictions.
- Recall is the ratio of true positive predictions to the total number of actual positive instances.

Explanation:
- The F1 Score is a measure that balances the trade-off between precision and recall. It gives equal weight to both precision and recall, and it's particularly useful when there is an uneven class distribution. This is important because in imbalanced datasets, simply optimizing for accuracy might not be the best approach.
- The harmonic mean is used because it gives more weight to lower values. Therefore, the F1 Score tends to be higher when both precision and recall are balanced and lower when they are not.
- A high F1 Score indicates a good balance between precision and recall, meaning that the classifier is making accurate positive predictions (precision) while also capturing most of the actual positives (recall). Conversely, a low F1 Score suggests that the classifier is biased towards either precision or recall, depending on the direction of the imbalance.

# Accuracy, Precision, and Recall

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.1 Model Evaluation Metrics

Accuracy:

Definition: Accuracy is a measure of how many of the total predictions made by a classification model are correct. It provides an overall assessment of the model's correctness.

Mathematics:

Accuracy is calculated as the ratio of correctly predicted instances to the total number of instances:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Precision:

Definition: Precision is a measure of how many of the predicted positive instances are actually correct. It is particularly relevant in cases where false positives are costly or misleading.

Mathematics:

Precision is calculated as the ratio of true positive predictions to the total number of positive predictions made by the model:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall (Sensitivity):

Definition: Recall, also known as Sensitivity or True Positive Rate, measures how many of the actual positive instances were correctly predicted by the model. It is important in scenarios where missing actual positives is costly.

Mathematics:

Recall is calculated as the ratio of true positive predictions to the total number of actual positive instances:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

# Confusion Matrix

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.1 Model Evaluation Metrics

A confusion matrix is a common tool for evaluating the performance of a classification model, particularly in binary classification problems. It provides a summary of the model's predictions and actual outcomes, breaking them down into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These elements are used to calculate various evaluation metrics.

Here's an explanation of the confusion matrix and its components:

True Positives (TP):

- These are cases where the model correctly predicted the positive class. In other words, the model correctly identified instances that belong to the positive class.

True Negatives (TN):

- These are cases where the model correctly predicted the negative class. The model correctly identified instances that do not belong to the positive class.

False Positives (FP):

- These are cases where the model incorrectly predicted the positive class when the actual class was negative. This is also known as a Type I error.

False Negatives (FN):

- These are cases where the model incorrectly predicted the negative class when the actual class was positive. This is also known as a Type II error.

# Confusion Matrix

Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation

2.1 Model Evaluation Metrics

The confusion matrix is used to calculate various evaluation metrics, including:

Accuracy:
- The proportion of all predictions that are correct (TP and TN) out of the total predictions. Precision (Positive Predictive Value):
- The proportion of true positive predictions out of all positive predictions, giving a measure of how many of the predicted positive cases were correct.

Recall (Sensitivity, True Positive Rate):
- The proportion of true positive predictions out of all actual positive cases, indicating how well the model identifies positive cases.

F1Score:
- The harmonic mean of precision and recall, which provides a balanced measure of a model's performance.

Specificity (True Negative Rate):
- The proportion of true negative predictions out of all actual negative cases, measuring the model's ability to identify negative cases.

False Positive Rate:
- The proportion of false positive predictions out of all actual negative cases.

Area Under the ROC Curve (AUC-ROC):
- A graphical metric that measures a model's ability to discriminate between positive and negative cases.

Area Under the Precision-Recall Curve (AUC-PR):
- A metric that focuses on precision and recall and is especially useful for imbalanced datasets.

# Bias, Variance, and Model Complexity

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.2 Understanding Model Complexity

Bias (B):

Bias measures the error introduced by approximating a real-world problem, which may be complex, by a simplified model. In the context of linear regression, bias can be calculated as follows:

Given a linear model $Y = \beta_0 + \beta_1 X + \varepsilon$, where $\beta_0$ and $\beta_1$ are model parameters and $\varepsilon$ is the error term, bias can be calculated as the difference between the expected value of the model's predictions and the true underlying values:

$$B = E(\hat{Y}) - Y$$

Where:
- $B$ represents the bias.
- $E(\hat{Y})$ is the expected value of the model's predictions.
- $Y$ is the true target value.

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.2 Understanding Model Complexity

Variance (V):
Variance measures the model's sensitivity to small fluctuations in the training data. In the context of linear regression, variance can be calculated as:

$$V = E\left((\hat{Y} - E(\hat{Y}))^2\right)$$

Where:
- $V$ represents the variance.
- $\hat{Y}$ is the model's predictions.
- $E(\hat{Y})$ is the expected value of the model's predictions.

# Bias, Variance, and Model Complexity

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.2 Understanding Model Complexity

Model Complexity (C):

Model complexity refers to the number of parameters and the level of flexibility in the model. In the context of linear regression, model complexity can be quantified by the number of predictors or features $(p)$ and the degree of the polynomial.

One way to mathematically represent model complexity is through the number of parameters. In a linear regression model, the number of parameters $(k)$ can be directly related to the model complexity:

$$C = k$$

Where:

- $C$ represents model complexity.
- $k$ is the number of parameters (including the intercept) in the model.

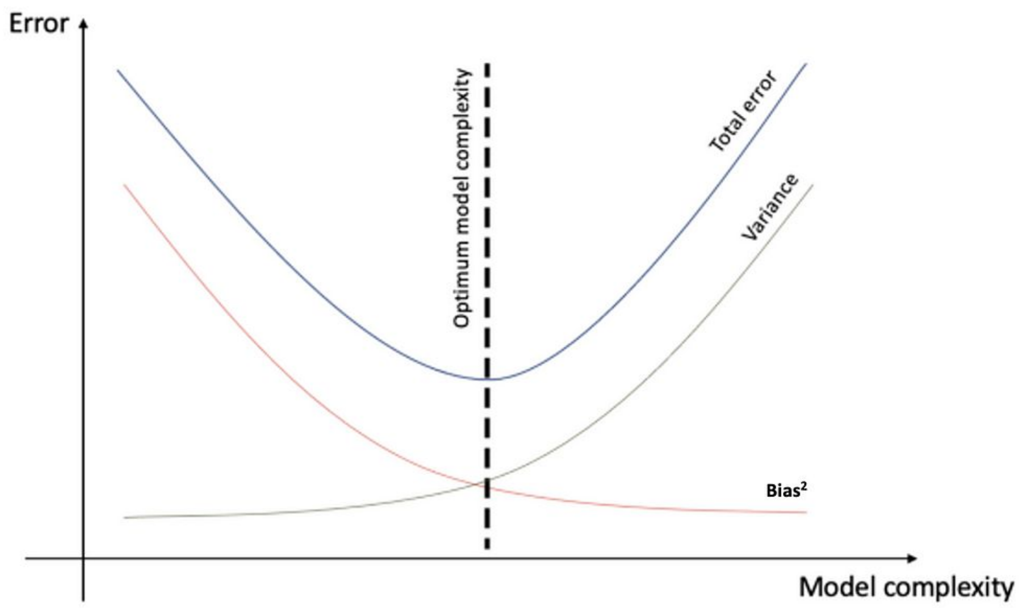
Understanding the trade-off between bias and variance is critical in model selection and regularization. A high-complexity model (high $k$ ) may have low bias but high variance, leading to overfitting. A low-complexity model (low $k$ ) may have high bias but low variance, leading to underfitting. The goal is to find an appropriate balance to minimize the expected prediction error.

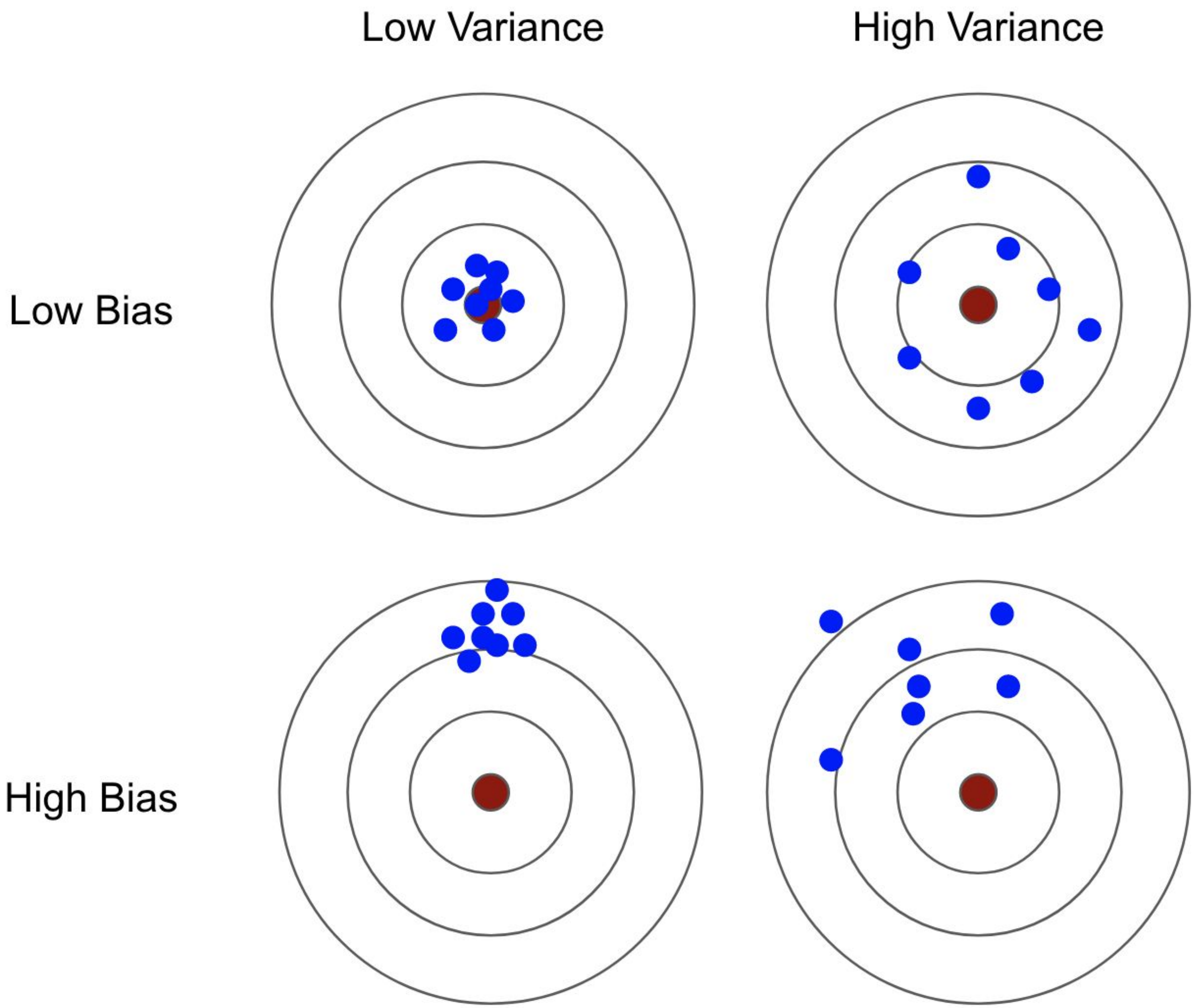
Regularization techniques, such as L1 and L2 regularization, can be used to control model complexity and strike a balance between bias and variance. These methods introduce penalty terms in the loss function to limit the impact of complex models.

# Example: Bias-Variance Tradeoff

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.2 Understanding Model Complexity

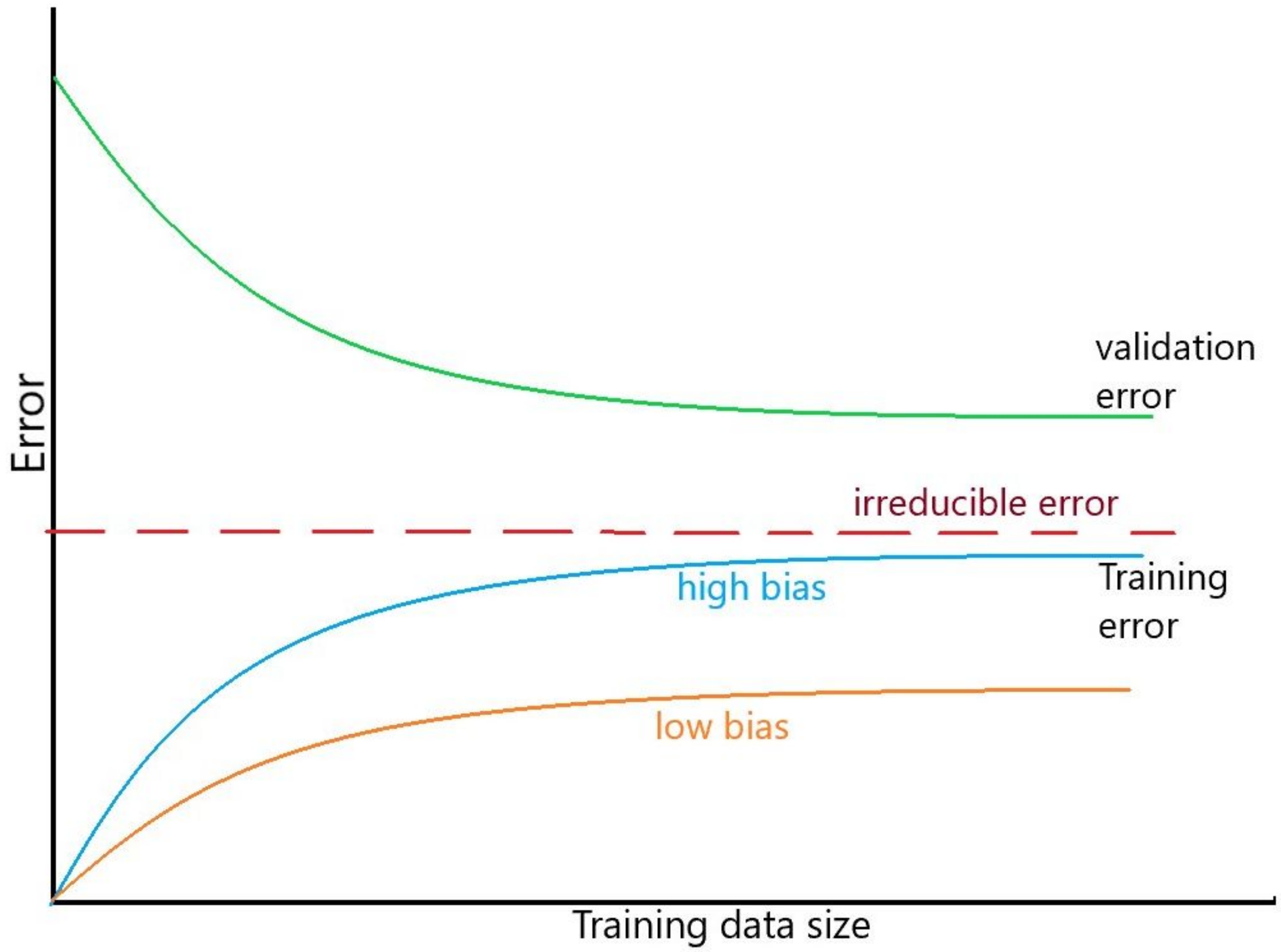


# Example: Bias-Variance Tradeoff

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

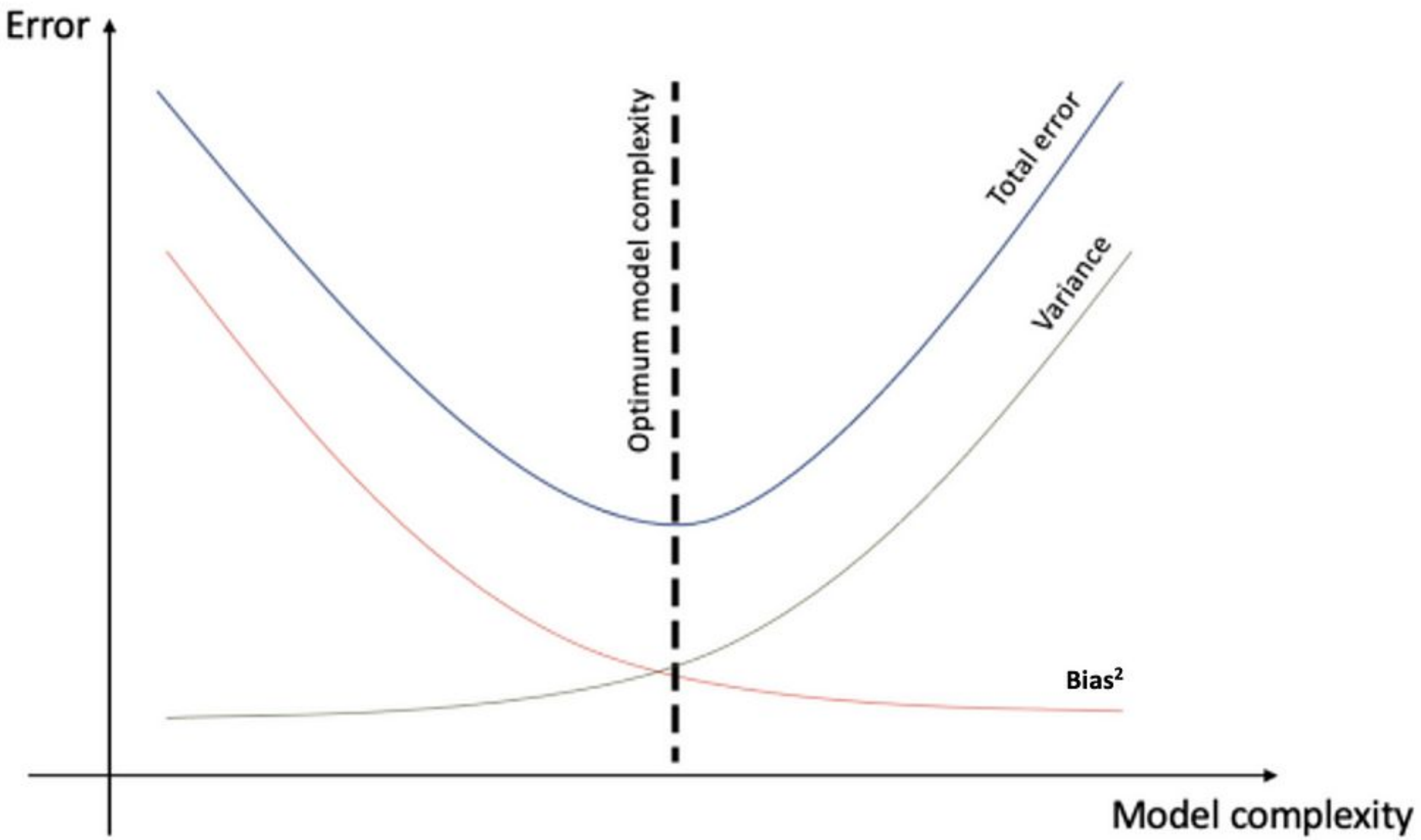
2.2 Understanding Model Complexity



# The Effective Number of Parameters

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.2 Understanding Model Complexity



# K-Fold Cross-Validation

**Part 2: Fundamentals of Model Evaluation, Model Complexity and Cross-Validation**

2.4 Cross-Validation

Cross-validation is a technique used in machine learning to assess how well a model will perform on an independent dataset. It helps prevent overfitting and provides a more accurate estimate of a model's performance.

In cross-validation, the dataset is split into multiple subsets, typically into two parts: a training set and a testing set. The model is trained on the training set and then evaluated on the testing set. This process is repeated multiple times, and each time, a different portion of the data is used as the testing set while the remaining data is used for training. The results are then averaged to provide a more reliable estimate of the model's performance. Common types of cross-validation include k-fold cross-validation, leave-one-out crossvalidation, and stratified cross-validation. Cross-validation helps ensure that the model generalizes well to unseen data and provides a robust assessment of its effectiveness.

# Akaike Information Criterion (AIC)

Part 3: Model Selection Techniques and Insights

3.1 Model Selection Techniques

The Akaike Information Criterion (AIC) is a statistical metric used for model selection and evaluation. It is particularly useful in the context of linear regression, generalized linear models, and other types of statistical models. AIC balances the trade-off between model fit and model complexity, helping to identify the model that best explains the data.

Mathematics of AIC:

In the context of model selection for a statistical model, AIC is calculated using the following formula:

$$AIC = -2\log(L) + 2k$$

Where:

$AIC$ is the Akaike Information Criterion.

- $\log(L)$ is the natural logarithm of the likelihood of the model.
- $k$ is the number of model parameters.

Explanation:

$-2 \setminus \log(\mathrm{L})$ : This term represents the model fit. It is proportional to the negative log-likelihood of the model. The likelihood measures how well the model explains the observed data. Models with a higher likelihood value fit the data better.

2k: This term penalizes the model for complexity. It is proportional to the number of model parameters $(k)$. Adding more parameters to a model increases complexity. AIC penalizes models with more parameters to avoid overfitting. The term $2k$ is a trade-off parameter that balances fit and complexity.

Interpretation:

Lower AIC values are better, as they indicate a good balance between model fit and complexity.

- When comparing different models, the model with the lowest AIC is typically preferred.
- AIC is a relative measure, so it is often used for comparing models rather than providing an absolute measure of goodness of fit.
- It is important to consider the context and domain-specific knowledge when interpreting AIC results.

# Bayesian Information Criterion (BIC) and The Bayesian Approach

**Part 3: Model Selection Techniques and Insights**

3.1 Model Selection Techniques

## Bayesian Information Criterion (BIC):

The Bayesian Information Criterion (BIC), also known as the Schwarz criterion, is another statistical metric used for model selection and evaluation. Like AIC, it balances the trade-off between model fit and complexity, but BIC places a stronger penalty on model complexity compared to AIC.

Mathematics of BIC:
In the context of model selection for a statistical model, BIC is calculated using the following formula:

$$BIC = -2\log(L) + k\log(n)$$

Where:
- $BIC$ is the Bayesian Information Criterion.
- $\log(L)$ is the natural logarithm of the likelihood of the model.
- $k$ is the number of model parameters.
- $n$ is the sample size (the number of data points).

## Explanation:

1. $-2\setminus\log(\mathrm{L})$ : This term represents the model fit, just like in the AIC formula. It is proportional to the negative log-likelihood of the model.
2. $\mathbf{k}\setminus\log(\mathbf{n})$ : This term penalizes model complexity more strongly than AIC. It scales with the number of model parameters $(k)$ and the natural logarithm of the sample size $(n)$. BIC places a heavier penalty on complex models, which can lead to the selection of simpler models.

## Interpretation:

- Lower BIC values are better, indicating a better balance between model fit and complexity.
- BIC is more conservative than AIC in terms of model complexity, which means it tends to favor simpler models.
- Similar to AIC, BIC is used for comparing models rather than providing an absolute measure of goodness of fit.

# L1 and L2 Regularization

**Part 4: Advanced Techniques in Model Selection**

4.1 Regularization Techniques

L1 and L2 Regularization, also known as Lasso (L1) and Ridge (L2) regularization, are techniques used to prevent overfitting and improve the performance of machine learning models, especially linear regression models. They add penalty terms to the model's loss function to encourage the model to have smaller parameter values. Here's the mathematics behind $L1$ and $L2$ regularization:

L1 Regularization (Lasso):
In L1 regularization, a penalty term is added to the loss function to encourage sparsity in the model. This means that it tends to set some of the model's parameter values to exactly zero, effectively performing feature selection.

The L1 regularization term is defined as:
$L1$ regularization term

$$= \lambda \sum_{i=1}^{p} |\theta_i|$$

Where:
- $\lambda$ is the regularization parameter, controlling the strength of the penalty.
- $p$ is the total number of model parameters.
- $\theta_i$ represents each parameter.

The overall loss function for L1-regularized linear regression is:
$\text{Loss} = \text{Mean Squared Error} + L1 \text{ regularization term}$

# L1 and L2 Regularization

**Part 4: Advanced Techniques in Model Selection**

4.1 Regularization Techniques

Lasso Regression (L1 Regularization):
Lasso regression, like Ridge, adds a regularization term to the OLS loss function, but in this case, it's an L1 regularization term. The L1 regularization term encourages sparsity in the model, effectively performing feature selection:

Loss Function for Lasso Regression:
$$\text{Loss} = \text{Mean Squared Error} + \lambda \sum_{i=1}^{p} |\theta_i|$$
Where:
- Loss is the loss function.
- Mean Squared Error is the OLS loss.
- $\lambda$ is the regularization parameter, controlling the strength of the penalty.
- $p$ is the total number of model parameters.
- $\theta_i$ represents each parameter.

Lasso regression aims to minimize this modified loss function, but it has the effect of setting some model parameters to exactly zero, performing feature selection.

# L1 and L2 Regularization

**Part 4: Advanced Techniques in Model Selection**

4.1 Regularization Techniques

L2 Regularization (Ridge):
In $L2$ regularization, a penalty term is added to the loss function to prevent large parameter values and encourage all parameters to be small but non-zero.

The $L2$ regularization term is defined as:
$L2$ regularization term

$$= \lambda \sum_{i=1}^{p} \theta_i^2$$

Where:
- $\lambda$ is the regularization parameter, controlling the strength of the penalty.
- $p$ is the total number of model parameters.
- $\theta_i$ represents each parameter.

The overall loss function for L2-regularized linear regression is:
$$\text{Loss} = \text{Mean Squared Error} + L2 \text{ regularization term}$$

# Ridge and Lasso Regression

**Part 4: Advanced Techniques in Model Selection**

4.1 Regularization Techniques

Ridge Regression (L2 Regularization):
Ridge regression adds an $L2$ regularization term to the ordinary least squares (OLS) loss function. The $L2$ regularization term encourages the model's parameter values to be small and prevents them from becoming too large:
Loss Function for Ridge Regression:

$$\text{Loss} = \text{Mean Squared Error} + \lambda \sum_{i=1}^{p} \theta_i^2$$

Where:
- Loss is the loss function.
- Mean Squared Error is the OLS loss.
- $\lambda$ is the regularization parameter, controlling the strength of the penalty.
- $p$ is the total number of model parameters.
- $\theta_i$ represents each parameter.

Ridge regression aims to minimize this modified loss function.

# L1 and L2 Regularization



Mathematics Explanation:
- The penalty terms in both L1 and L2 regularization are controlled by the hyperparameter $\lambda$. A larger $\lambda$ leads to a stronger penalty, encouraging smaller parameter values.
- In L1 regularization, the absolute values of the parameters are summed, promoting sparsity. It effectively encourages feature selection by setting some coefficients to zero.
- In $L2$ regularization, the squares of the parameters are summed, promoting all parameter values to be small but non-zero.
- The choice between L1 and L2 regularization depends on the problem and the specific trade-offs you want to make in the model's complexity and feature selection.

L1 regularization is useful for feature selection, while L2 regularization is good for preventing multicollinearity and obtaining a more balanced model with small but non-zero coefficients.

# KEYWORDS

- Inference and Estimation Theory
- Statistical Inference
- Probability in Inference
- Types of Statistical Inference
- Point Estimation
- Estimators
- Bias in Estimation
- Consistency in Estimation
- Efficiency in Estimation
- Maximum Likelihood Estimation (MLE)
- Interval Estimation
- Confidence Intervals
- Hypothesis Testing
- Null Hypothesis
- Alternative Hypothesis
- Significance Levels
- Alpha Level
- P-Values
- Parametric Hypothesis Tests
- Z-Tests
- T-Tests
- ANOVA
- Nonparametric Hypothesis Tests
- Wilcoxon Signed-Rank Test
- Mann-Whitney U Test
- Kruskal-Wallis Test
- Model Evaluation Metrics
- Accuracy
- Precision
- Recall
- F1-Score
- ROC Curve
- Confusion Matrix
- Overfitting
- Bias-Variance Trade-off
- Cross-Validation
- k-Fold Cross-Validation
- Leave-One-Out Cross-Validation (LOOCV)
- Model Selection Criteria
- Akaike Information Criterion (AIC)
- Bayesian Information Criterion (BIC)
- Regularization Techniques
- L1 Regularization
- L2 Regularization
- Ridge Regression
- Lasso Regression
- Elastic Net
- Feature Selection
- Forward Selection
- Backward Selection
- Recursive Feature Elimination (RFE)
- Ensemble Methods
- Bagging
- Boosting
- Random Forests
- Model Stacking
- Data-Driven Decisions
- Real-World Scenarios
- Inference and Estimation Application
- Statistical Model Selection
- Model Assessment
- Advanced Topics in Model Selection

Arbre de décision | Sélection de modèle | Régularité du modèle

# Use Case with python

## Use Case: Model Selection in Machine Learning

In the context of the course on Inference and Estimation Theory, covering topics on statistical inference, point estimation, interval estimation, hypothesis testing, model evaluation, model selection techniques, and advanced topics, let's explore a use case related to model selection in machine learning.

Description:

In this use case, we will focus on using model selection techniques to choose the best machine learning model for a classification problem. Model selection is crucial in machine learning to ensure that the selected model generalizes well to unseen data and performs optimally.

Key Components:

Introduction to Inference and Estimation Theory: Understanding the basics of statistical inference, point and interval estimation, hypothesis testing, and their relevance to machine learning model selection.

Model Evaluation Metrics: Utilizing accuracy, precision, recall, F1-score, ROC curve, and the confusion matrix as evaluation metrics to assess the performance of machine learning models.

Model Selection Techniques: Exploring techniques to prevent overfitting, including cross-validation, AIC, BIC, and cross-validation-based model selection. Understanding the bias-variance trade-off in model complexity.

Advanced Topics in Model Selection: Examining regularization techniques like L1 and L2 regularization, feature selection methods, and ensemble methods for improving model performance.

### Python Code Example (Model Selection in Machine Learning):

```python
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load the Iris dataset as an example
data = load_iris()
X, y = data.data, data.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Model selection: Compare Logistic Regression and Random Forest
models = {
    'Logistic Regression': LogisticRegression(max_iter=1000),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42)
}

results = {}
for model_name, model in models.items():
    # Perform k-fold cross-validation (e.g., 5-fold)
    scores = cross_val_score(model, X_train, y_train, cv=5)
    avg_accuracy = np.mean(scores)
    results[model_name] = avg_accuracy

# Select the best model based on cross-validation results
best_model = max(results, key=results.get)
print("Model Selection Results:")
for model_name, accuracy in results.items():
    print(f"{model_name}: {accuracy:.2f}")
print(f"Best Model: {best_model}")

# Train the best model on the full training set and evaluate on the test set
best_model_instance = models[best_model]
best_model_instance.fit(X_train, y_train)
y_pred = best_model_instance.predict(X_test)
test_accuracy = accuracy_score(y_test, y_pred)
print(f"Test Accuracy of {best_model}: {test_accuracy:.2f}")
```

In this code, we load the Iris dataset, split it into training and testing sets, and compare two machine learning models: Logistic Regression and Random Forest. We use 5-fold cross-validation to evaluate their performance and select the best model based on average accuracy. Finally, we train the best model on the full training set and evaluate it on the test set.

This use case illustrates how model selection techniques from the course on Inference and Estimation Theory can be applied in machine learning to choose the most suitable model for a specific task, ultimately leading to better predictive performance.

# REFERENCES


- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.
- Burnham, K. P., & Anderson, D. R. (2002). Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.
- Akaike, H. (1974). A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control.
- Schwarz, G. (1978). Estimating the Dimension of a Model. The Annals of Statistics.
- Ho, T. K. (1998). The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Breiman, L. (1996). Bagging Predictors. Machine Learning.
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences.
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software.
- Efron, B., & Tibshirani, R. (1997). Improvements on Cross-Validation: The 632+ Bootstrap Method. Journal of the American Statistical Association.
- Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society: Series B.
- Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.
- Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of the International Joint Conference on Artificial Intelligence.
- Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer.
- Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research.
- Arlot, S., & Celisse, A. (2010). A Survey of Cross-Validation Procedures for Model Selection. Statistics Surveys.
- Fawcett, T. (2006). An Introduction to ROC Analysis. Pattern Recognition Letters.
- Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning.
- Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. Journal of the Royal Statistical Society: Series B.

# TEXTE DE DESCRIPTION DU COURS

Dive into the world of "Inference and Estimation Theory" with this comprehensive course.

In the initial section, you'll explore the fundamental "Basics of Inference," including statistical inference, the role of probability in inference, and the various types of statistical inference. You'll then move on to "Point Estimation," where you'll delve into estimators, their key properties like bias, consistency, and efficiency, and the concept of Maximum Likelihood Estimation (MLE). Additionally, you'll gain expertise in "Interval Estimation," covering confidence intervals, their construction, and effective interpretation.

The course proceeds to "Hypothesis Testing and Model Evaluation," starting with the "Foundations of Hypothesis Testing." Here, you'll learn about null and alternative hypotheses, significance levels (alpha), and how to interpret p-values. The course demystifies "Parametric Hypothesis Tests," including Z-tests, T-tests, and ANOVA, and their applications. You'll also explore "Nonparametric Hypothesis Tests," unveiling tests like the Wilcoxon Signed-Rank Test, Mann-Whitney U Test, and Kruskal-Wallis Test, expanding your analytical toolkit. Furthermore, essential model evaluation metrics such as accuracy, precision, recall, F1-score, ROC curve, and the confusion matrix will be covered.

In the following section, you'll navigate the intricacies of "Model Selection Techniques." This includes understanding "Overfitting and Bias-Variance Trade-off," where you'll grasp the concept of overfitting and the balance between bias and variance in models. You'll also delve into "Cross-Validation," exploring techniques like k-Fold Cross-Validation and Leave-One-Out Cross-Validation (LOOCV) for model assessment and selection. Additionally, the course introduces "Model Selection Criteria," including the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and cross-validation-based methods.

The final segment takes you to the cutting edge with "Advanced Topics in Model Selection." Here, you'll explore "Regularization Techniques," including L1 and L2 regularization, Ridge and Lasso Regression, and Elastic Net. You'll also learn about "Feature Selection" and its techniques for selecting relevant features, such as forward and backward selection and Recursive Feature Elimination (RFE). Lastly, you'll uncover the power of "Ensemble Methods" with insights into bagging, boosting, random forests, and model stacking.

By the end of this course, you'll have a comprehensive understanding of inference and estimation theory, equipping you to apply these concepts in real-world scenarios and make data-driven decisions with confidence.