**🔷 Overview of the Lecture**

The lecture explains how supervised machine learning tasks generally work and dives deeper into **Naive Bayes**, focusing on **tokenization** and **vectorization**, which are crucial for handling **text data**.

**🔶 1. General Steps in Supervised Machine Learning**

Whether you're doing **classification** or **regression**, the basic structure of any supervised ML task follows these steps:

**✅ Step 1: Preprocessing the Data**

* Clean and organize raw data.
* Convert it into a form that the model can work with.

**✅ Step 2: Define Variables**

* **Dependent variable**: What you want to predict (e.g., spam or not spam).
* **Independent variables**: Features used to make the prediction (e.g., words in an email).
* Feature selection might be needed to choose only the important features.

**✅ Step 3: Split the Dataset**

* Divide your dataset into:
  + **Training set** – to teach the model.
  + **Test set** – to check how well the model learned.
* These sets should be independent of each other.

**✅ Step 4: Train the Model**

* The model learns from the training data.

**✅ Step 5: Test & Evaluate the Model**

* Use metrics like:
  + **Accuracy**
  + **Precision**
  + **Recall**
  + **F1 Score**

These tell you how reliable and effective your model is.

**🔶 2. Special Considerations for Naive Bayes**

Naive Bayes is **especially popular in text classification tasks** like spam detection, sentiment analysis, etc.

But before using Naive Bayes, you need two important pre-processing steps:

**🔷 A. Tokenization**

* **What is it?**  
  Splitting text into smaller pieces called **tokens**.
* **Example:**  
  Text: "You win a prize"  
  Tokens: ["You", "win", "a", "prize"]
* Each token (usually a word) becomes a feature for the model.

**🔷 B. Vectorization**

* **What is it?**  
  Converting tokens (text) into **numbers**, because ML models can’t understand plain text.
* **Common method:**  
  **Bag of Words** – represent text as an array of numbers:
  + Each position in the array corresponds to a token.
  + The value is how often the token appears in the document.

**🔶 3. Naive Bayes Training Process**

Once tokenization and vectorization are done:

**✅ During Training:**

* The algorithm learns the **probability** of each token appearing in each class.
* Example:
  + The word **"win"** might often appear in spam emails.
  + Naive Bayes calculates:
    - P("win" | spam)
    - P("win" | not spam)
* Then it does the same for other words like **"prize"**.

**✅ Naive Assumption (Why "Naive"?)**

* It assumes **all tokens are independent** of each other.
* So, it multiplies their probabilities directly, which simplifies the math.

**🔶 4. Prediction Phase**

When a **new text** comes in (like a new email), the model:

1. **Tokenizes** the text.
2. **Vectorizes** it into numbers.
3. **Uses the learned probabilities** to predict which class the new text most likely belongs to (e.g., spam or not spam).

**🔶 5. Summary and Key Takeaways**

* All supervised ML tasks follow a general process: preprocessing → variable definition → data split → training → evaluation.
* For **Naive Bayes with text**, the **key challenge** is converting human language (text) into machine-readable input.
* That’s why **tokenization** and **vectorization** are essential steps.
* Naive Bayes:
  + Uses **probabilities** of words given a class.
  + Assumes **word independence** (naive assumption).
  + Is **fast** and **effective** for text classification.

**🧠 Tip to Remember:**

**Tokenization** = Splitting text into words  
**Vectorization** = Turning words into numbers  
**Naive Bayes** = Uses probabilities of these numbers (tokens) to classify data