Information Retrieval

Amin Nazari

Spring 2025

Chapter 7

Text Classification(Traditional Machine Learning)

Outline

- Machine Learning
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Classification algorithms
- Text Classification

Machine learning

- Instead of explicitly programming computers, machine learning attempts to force them to learn to act with appropriate accuracy based on input data.

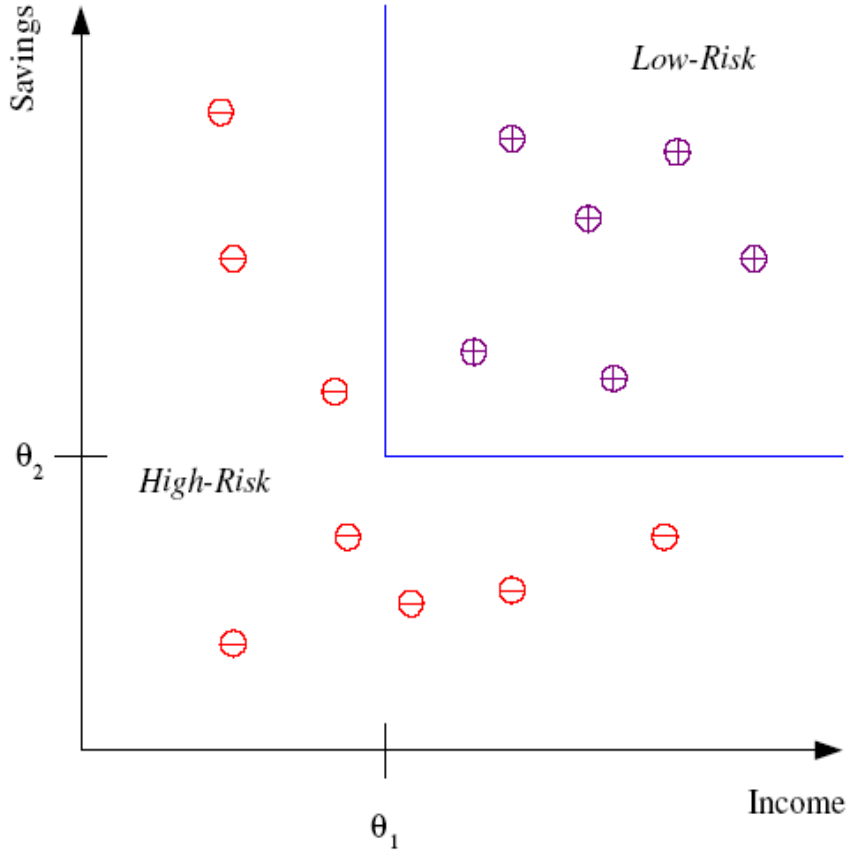
- ML is fundamentally data-driven. The core idea behind ML is that algorithms learn patterns, relationships, and structures in data without being explicitly programmed. In other words, instead of being given a set of rules to follow, the model uses data to "learn" and make predictions or decisions based on that information.

Supervised Learning

- **Supervised Learning** is a type of machine learning where the model is trained on **labeled data**. This means that for each input in the training set, the correct output (or label) is provided. The goal of supervised learning is for the model to learn a mapping from inputs (features) to outputs (labels) so that it can predict the output for new, unseen data.

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

Regression

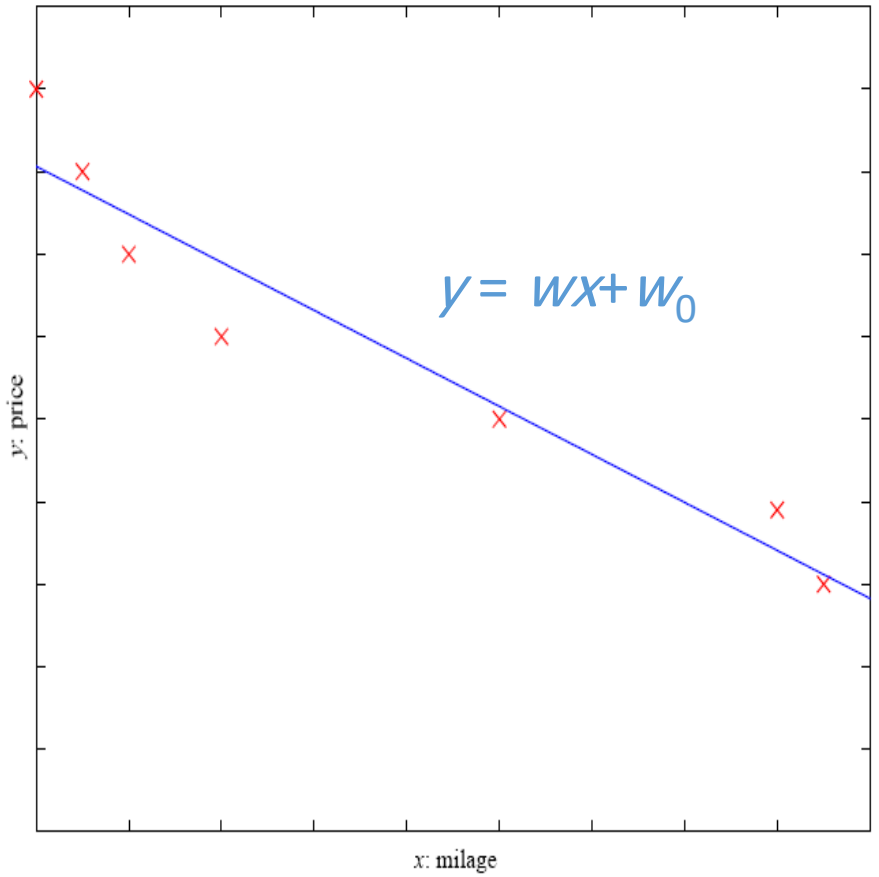
- Example: Price of a used car
- x : car attributes

  y : price

  $$y = g(x \mid \theta)$$

  $g(\ )$ model,

  θ parameters

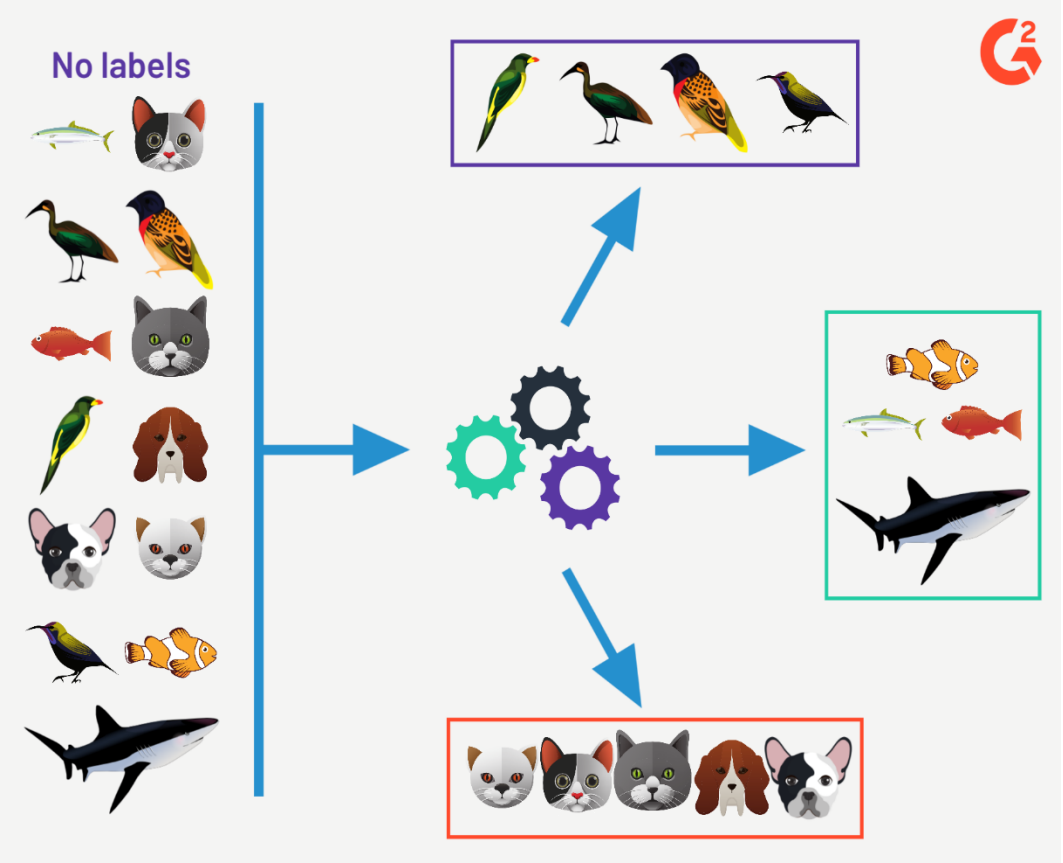


Unsupervised Learning

- **Unsupervised learning** is a type of machine learning where the model is trained on data without labeled responses. Instead of predicting a specific outcome, the algorithm identifies patterns, structures, or relationships within the data. Common techniques include clustering (grouping similar data points) and dimensionality reduction (simplifying data while preserving its structure).

Unsupervised Learning

- **Minimizing intra-cluster distance** (making points within a cluster close to each other).
- **Maximizing inter-cluster distance** (making clusters far apart from each other).

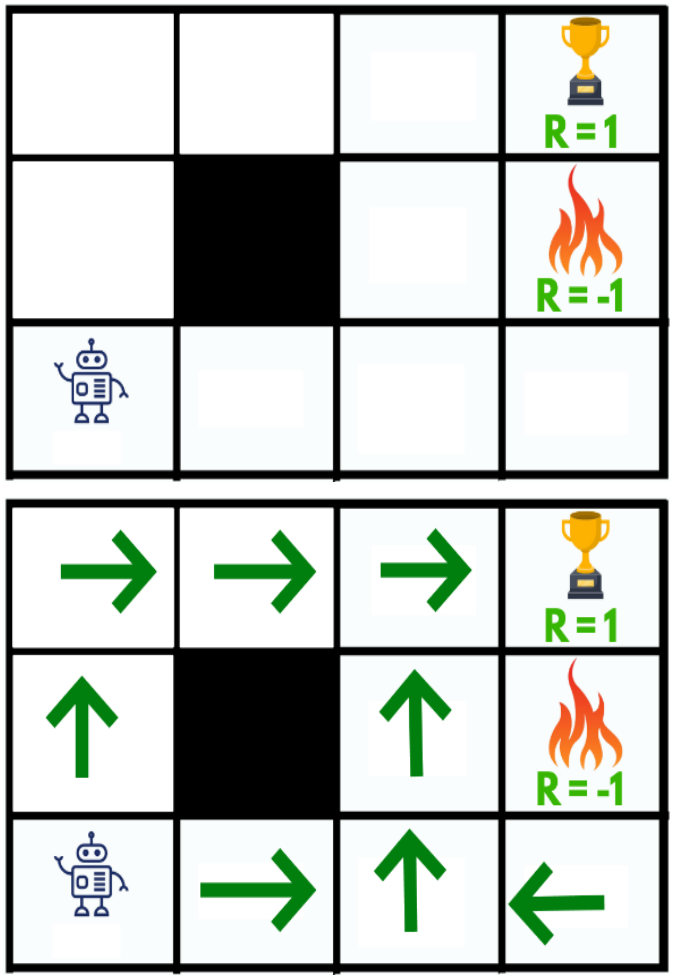


Reinforcement learning

- **Reinforcement learning** (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. The agent takes actions, receives feedback in the form of rewards or penalties, and adjusts its strategy to improve performance over time. It operates on a trial-and-error basis, guided by a policy that maps states to actions.

Reinforcement learning

- **Exploration and Interaction**: The agent explores the environment by taking actions and observing the outcomes (states and rewards).
- **Reward Feedback**: The environment provides immediate feedback in the form of rewards or penalties, guiding the agent's learning process.
- **Policy Improvement**: Over time, the agent refines its policy (strategy) to maximize cumulative rewards.



Similar Terms

- Machine Learning
- Data Science
- Data Mining

What is Classification?

- **Definition:**
  Classification is the task of predicting the label or category of a given input based on prior training.
- **Example:**
  Email →Spam or Not Spam
  Image →Cat, Dog, or Bird

Key Concepts

- **Features:** Input variables used for prediction.
- **Labels:** The categories or classes to predict.
- **Model:** A mapping function from features to labels.
- **Training vs Testing:** Learning phase vs evaluation phase.

Common Classification Algorithms

- Logistic Regression
- Naïve Bayes
- Decision Trees
- Support Vector Machines (SVM)
- K-Nearest Neighbors (KNN)
- Neural Networks
- Ensemble Methods (e.g., Random Forest, XGBoost)

Naïve Bayes

- Logistic Regression
- Naïve Bayes
- Decision Trees
- Support Vector Machines (SVM)
- K-Nearest Neighbors (KNN)
- Neural Networks
- Ensemble Methods (e.g., Random Forest, XGBoost)

Assumption of Naive Bayes

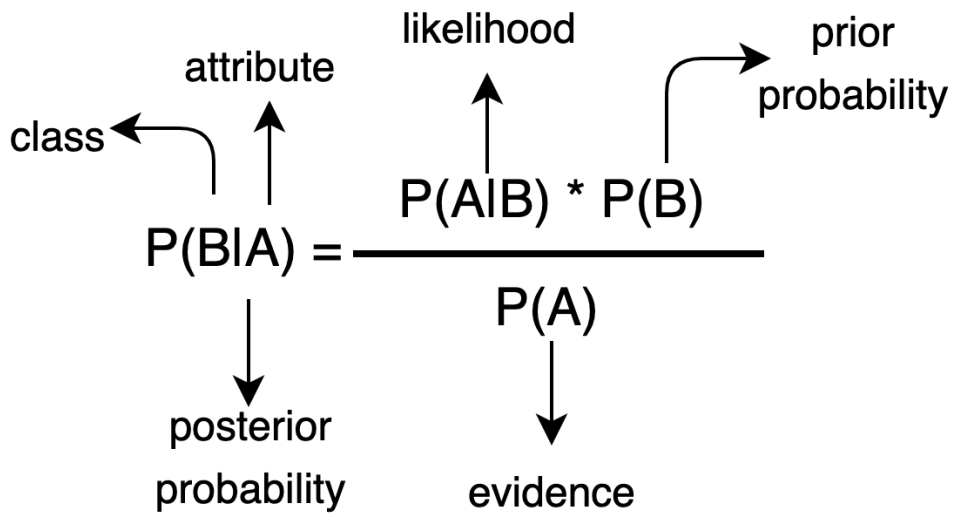
- **Feature independence:** This means that when we are trying to classify something, we assume that each feature (or piece of information) in the data does not affect any other feature.
- **Continuous features are normally distributed:** If a feature is continuous, then it is assumed to be normally distributed within each class.
- **Discrete features have multinomial distributions:** If a feature is discrete, then it is assumed to have a multinomial distribution within each class.
- **Features are equally important:** All features are assumed to contribute equally to the prediction of the class label.
- **No missing data:** The data should not contain any missing values.

Bayes Theorem Formula

- $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$
- $P(A \cap B) = P(A \mid B) * P(B)$
- $P(B \mid A) = \frac{P(B \cap A)}{P(A)}$
- $P(B \cap A) = P(B \mid A) * P(A)$
- $P(B \cap A) = P(A \cap B)$
- $P(B \mid A) * P(A) = P(A \mid B) * P(B)$

Bayes Theorem Formula

- $P(c \mid \mathrm{X}) = \frac{P(X|c) * P(c)}{P(X)}$
- $P(c \mid \mathrm{X}) \propto P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$



Multinomial Naïve Bayes

- It is used for discrete counts. This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

$$P(x_i \mid Class) = \frac{P(x_i \cap Class)}{P(Class)}$$

Multinomial Naïve Bayes

- $X = \{x_1, x_2, \dots, x_n\}$
- $P(Class \mid \mathrm{X}) = \frac{P(X|Class) * P(Class)}{P(X)}$
- $P(Class \mid \mathrm{X}) \propto P(x_1 \mid Class) \times P(x_2 \mid Class) \times \cdots \times P(x_n \mid Class) \times P(Class)$
- $P(x_i \mid Class) = \frac{P(x_i \cap Class)}{P(Class)}$

- $y = \begin{cases} No & P(Class = No|X) > P(Class = Yes|X) \\ Yes & else \end{cases}$

Example

- today = (Sunny, Hot, Normal, False)

Day	OUTLOOK	TEMPERATURE	WINDY	CLASS (PLAY = YES PLAY = NO)
1	Sunny	Hot	Weak	No
2	Overcast	Hot	Weak	Yes
3	Sunny	Hot	Strong	No
4	Rain	Cool	Strong	No
5	Rain	Cool	Weak	Yes
6	Rain	Mild	Weak	Yes
7	Overcast	Cool	Strong	Yes
8	Sunny	Mild	Weak	No
9	Sunny	Cool	Weak	Yes
10	Rain	Mild	Weak	Yes
11	Sunny	Mild	Strong	Yes
12	Overcast	Mild	Strong	Yes
13	Overcast	Hot	Weak	Yes
14	Rain	Mild	Strong	No

- P(No|today) = P(Outlook = Sunny|No) * P(Temperature = Hot|No) * P(Humidity = Normal|No) * P(Wind = False|No) * P(No)
- P(Yes|today) = P(Outlook = Sunny|Yes) * P(Temperature = Hot|Yes) * P(Humidity = Normal|Yes) * P(Wind = False|Yes) * P(Yes)

	Yes	No	P(Outlook|yes)	P(Outlook|No)
Overcast	4	0	4/9	0/5
Rain	3	2	3/9	2/5
Sunny	2	3	2/9	3/5
	Total yes = 9	Total no = 5		

	Yes	No	P(Temperature|yes)	P(Temperature |No)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
	Total yes = 9	Total no = 5		

	Yes	No	P(Windy |yes)	P(Windy |No)
Weak	6	2	6/9	2/5
Strong	3	3	3/9	3/5
	Total yes = 9	Total no = 5		

P(class=Yes) = 9/14 P(class=No) = 5/14

Gaussian Naïve Bayes

- It is used in classification and it assumes that the predictors/features take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution (follow a normal distribution).
- The parameters of the Gaussian are the mean and variance of the feature values. Since the way the values are present in the dataset changes, the formula for conditional probability changes to:

$$P(x_i \mid Class) = \frac{1}{\sqrt{2\pi\sigma_{Class}^2}} \exp\left(-\frac{(x_i - \mu_{Class})^2}{2\sigma_{Class}^2}\right)$$

Gaussian Naïve Bayes

- $X = \{x_1, x_2, \dots, x_n\}$
- $P(Class \mid \mathrm{X}) = \frac{P(X|Class) * P(Class)}{P(X)}$
- $P(Class \mid \mathrm{X}) \propto P(x_1 \mid Class) \times P(x_2 \mid Class) \times \cdots \times P(x_n \mid Class) \times P(Class)$
- $P(x_i \mid Class) = \frac{1}{\sqrt{2\pi\sigma_{Class}^2}} \exp(-\frac{(x_i - \mu_{Class})^2}{2\sigma_{Class}^2})$

- $y = \begin{cases} No & P(Class = No|X) > P(Class = Yes|X) \\ Yes & else \end{cases}$

Example (Quiz)

- X=(Refund=No,Married,Income=120K)
- For income variable when Evade = No:

  Sample mean:110

  Sample Variance: 2975
- For income variable when Evade = Yes:

  Sample mean: 95

  Sample Variance: 25

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Advantages of Naive Bayes

- **Fast and Efficient**
  - Works well with large datasets due to its simplicity.
  - Training and prediction times are very low compared to complex models like SVM or neural networks.
- **Works Well with High-Dimensional Data**
  - Performs well even when the number of features is large (e.g., text classification).
- **Good for Small Datasets**
  - Can produce reasonable results even with limited training data.
- **Handles Categorical and Numerical Data**
  - Different variants (Gaussian, Multinomial, Bernoulli) allow flexibility in handling different data types.
- **Robust to Irrelevant Features**
  - Since it assumes feature independence, irrelevant features do not heavily impact performance.
- **Low Risk of Overfitting**
  - Due to its simplicity, it generalizes well to unseen data.
- **Easy to Implement**
  - Requires minimal hyperparameter tuning compared to deep learning models.

Disadvantages of Naive Bayes

- **Strong Independence Assumption (Naivety)**
  - Assumes all features are independent, which is rarely true in real-world data.
  - Can lead to poor performance if features are correlated.
- **Zero-Frequency Problem**
  - If a category in the test data was not seen in training, it assigns a zero probability (solved using Laplace smoothing).
- **Not Ideal for Complex Relationships**
  - Struggles with capturing complex patterns compared to decision trees or neural networks.
- **Biased Probability Estimates**
  - Predicted probabilities are not always well-calibrated (better for ranking than exact probabilities).
- **Sensitive to Input Data Quality**
  - Requires good preprocessing (e.g., handling missing values, outliers).

Naive Bayes for text classification

- For handling unseen words use the add-one (Laplace) smoothing

$$P(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Naive Bayes for text classification

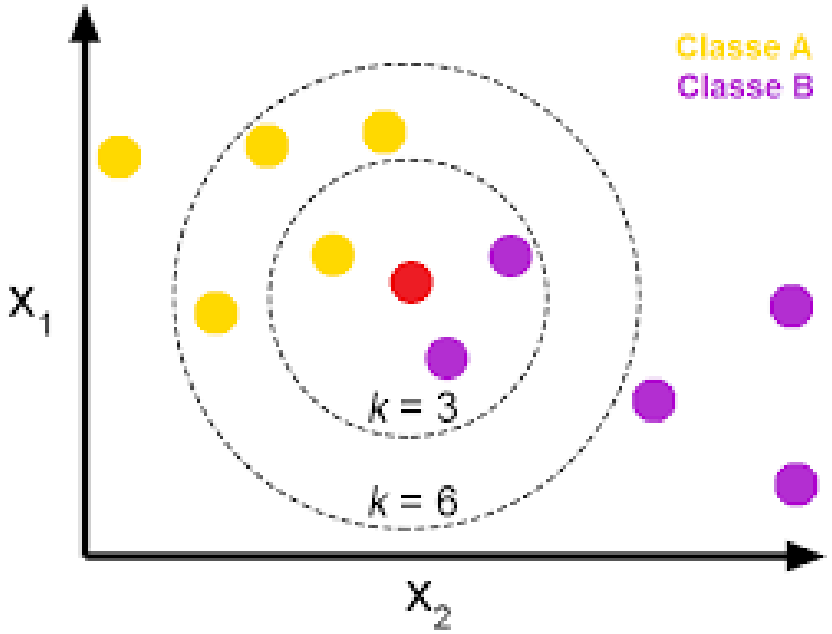
	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Naive Bayes as a Language Model

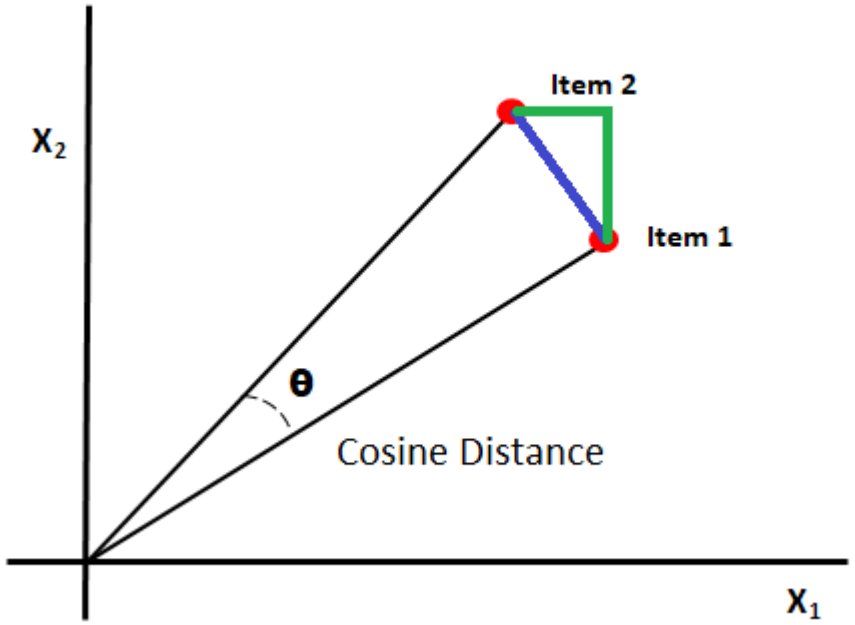
- Acts like a class-conditional unigram language model
- For each class ccc, learn a word distribution P(w|c)
- Classify new text by checking which model it "fits" best
- Treats language generation as probabilistic event

KNN

1.Euclidean Distance
2.Manhattan Distance
3.Hamming Distance
4.Cosine Similarity



KNN

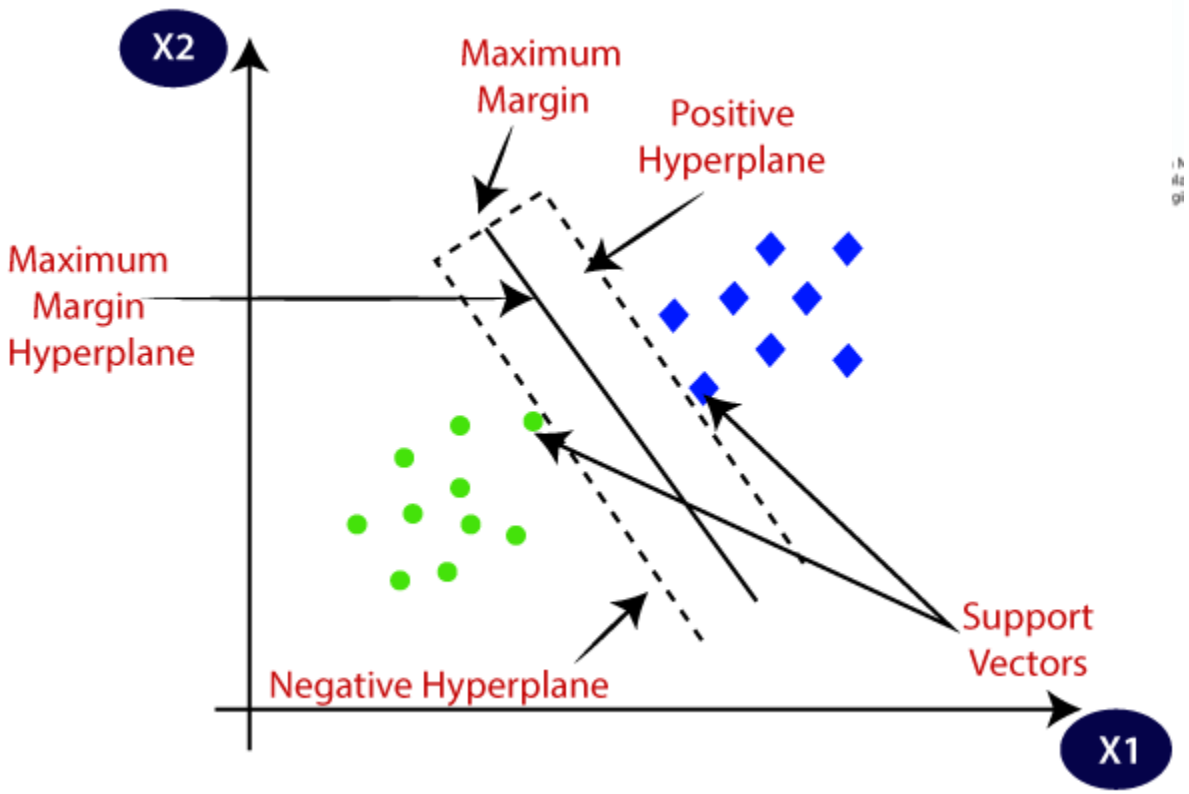


$$L^1 = |x_2 - x_1| + |y_2 - y_1|$$

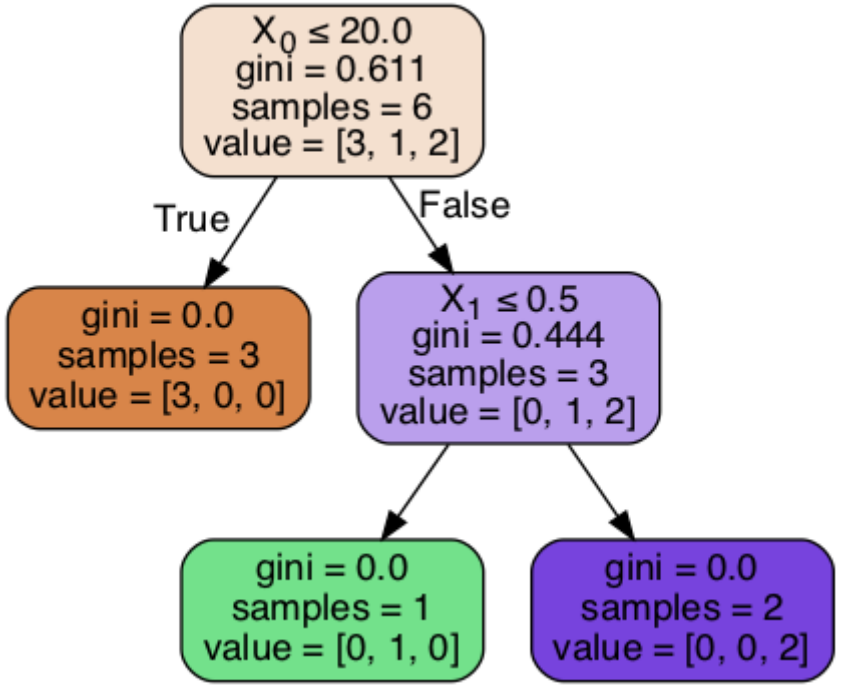
$$L^2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

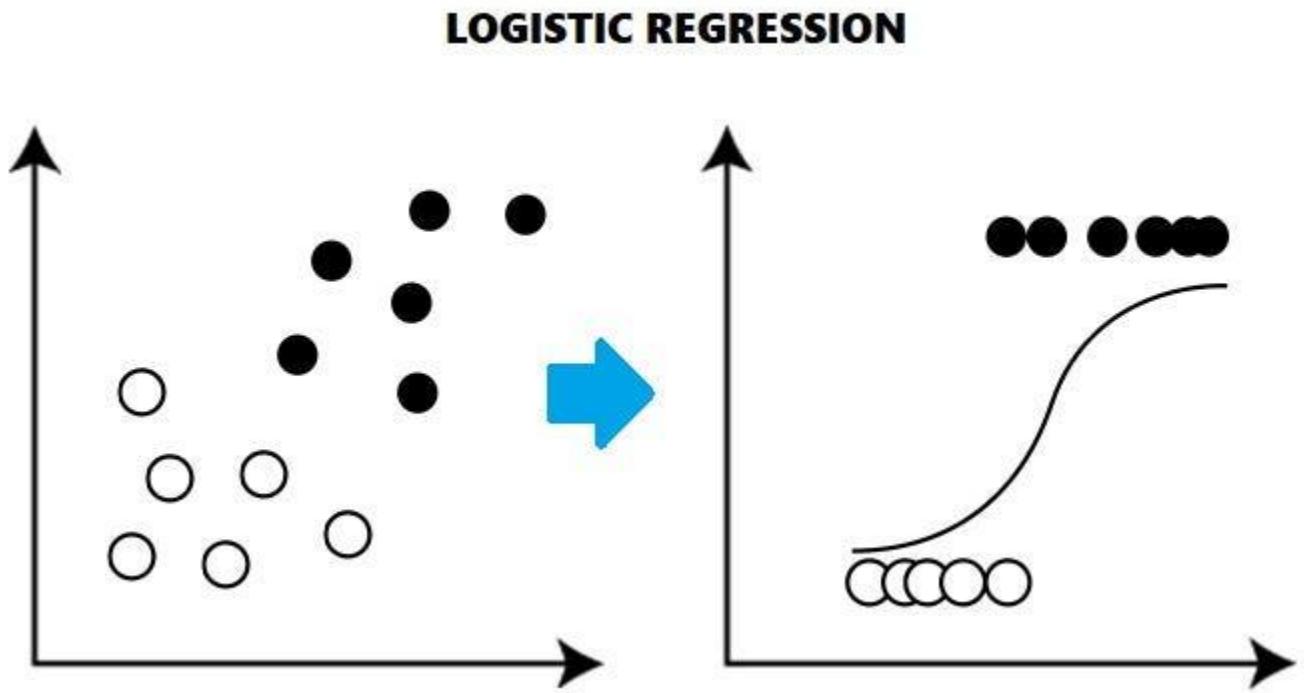
SVM


X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

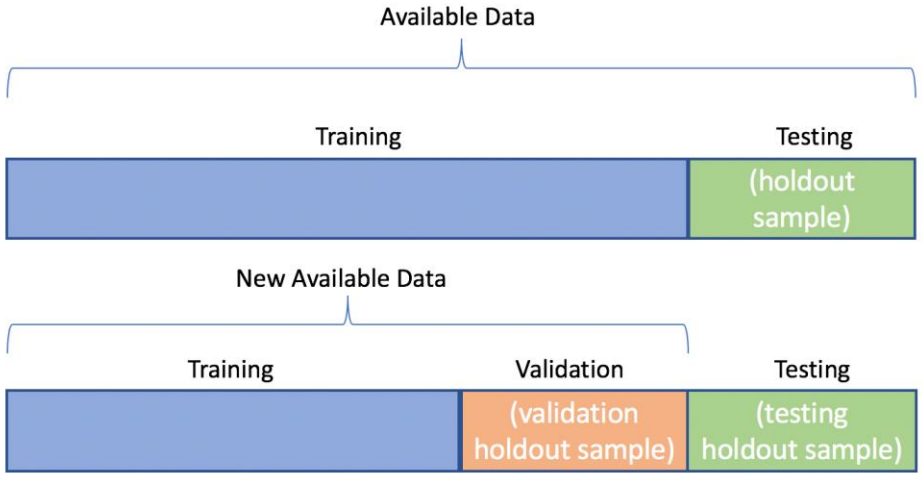
Decision Tree

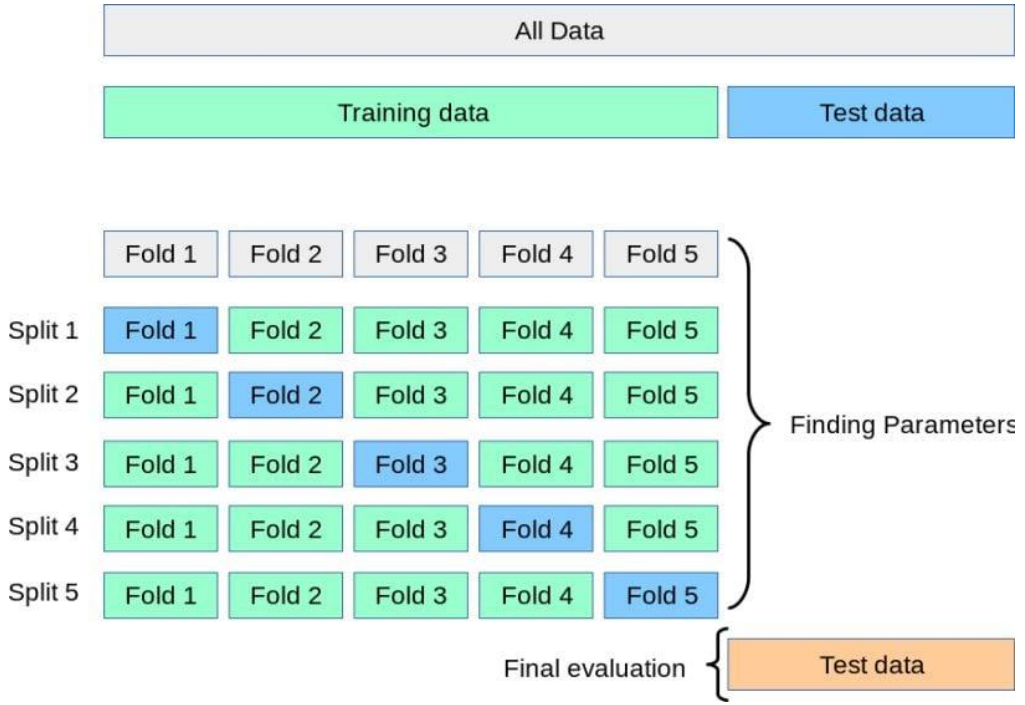


Logistic Regression

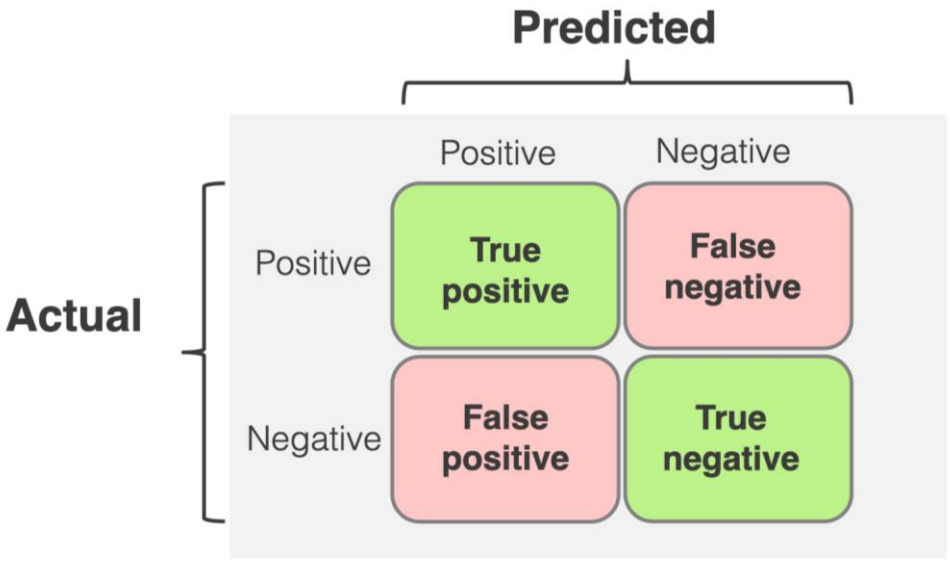


Train-Test Split-Cross Validation





Confusion matrix



$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

Confusion matrix

- **Dataset:**
  - Total patients: **1,000**
  - Positive cases (disease present): **10**
  - Negative cases (healthy): **990**

Confusion matrix

	Predicted Positive	Predicted Negative	Total
Actual Positive (Has disease)	5 (**TP**)	5 (**FN**)	10
Actual Negative (Healthy)	0 (**FP**)	990 (**TN**)	990

	Predicted Positive	Predicted Negative	Total
Actual Positive	9 (**TP**)	1 (**FN**)	10
Actual Negative	90 (**FP**)	900 (**TN**)	990

Bayes' theorem and classification

Given a 1% actual risk and a 20% model-predicted risk, the probability of a positive prediction being accurate is only 3.39%.

	Tests positive	Tests negative
At risk	198	2
Not at risk	50	750

$$\text{precision} = \frac{198}{198 + 50} = .798$$

$$P(\text{At Risk if Positive}) = \frac{P(\text{Positive if At Risk}) \times P(\text{At Risk})}{P(\text{Positive})}$$

$$P(\text{At Risk if Positive}) = \frac{.99 \times .01}{.248}$$

$$P(\text{At Risk if Positive}) = .0339$$

Python

- **Traditional text classification**