UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Machine Learning

Session 9 - T

Instance-based and Probabilistic Models

Degree in Applied Data Science

2024/2025

Instance-Based Learning

- Instance-based models are a class of machine learning algorithms that **make predictions based on similarities between instances**.
- Idea: **similar examples have similar labels**.
  - What is meant by similar?
    - Similar means "close" / **similar feature values**.
- **Algorithm:**
  - Given a new example X for which we want to predict the label Y;
  - Find the most similar training examples (closest);
  - Predict the label Y of X based on the labels of the most similar examples.

Instance-Based Learning

- **Questions:**
  - **How to determine similarity?**
  - **What similarity measures to use?**
  - **How does the model learn?**
  - **How many similar examples to consider?**
  - **How to resolve inconsistencies among the similar examples?**

Instance-Based Learning

- **Questions:**
  - **How to determine similarity?**
    - Similarity is determined using a similarity measure, which quantifies the closeness between instances in the feature space.
  - **What similarity measures to use?**
    - Common measures include Euclidean and Manhattan distances, cosine similarity, and Pearson correlation coefficient, depending on the data type and characteristics.
  - **How does the model learn?**
    - The model does not "learn" (lazy-learner), it stores the entire training dataset and makes predictions based on the similarities between new instances and the stored examples.

Instance-Based Learning

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- **Questions:**
  - **How many similar examples to consider?**
    - It is a parameter that needs to be tuned based on the specific characteristics of the dataset and the problem at hand.
  - **How to resolve inconsistencies among the similar examples?**
    - For classification problems, the predicted label is determined by majority voting.
    - For regression problems, the predicted label is determined by averaging (or weighted averaging) the target values of the similar examples.

Instance-Based Learning

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- **Advantages:**
  - **Flexibility:** Instance-based models can handle complex relationships and non-linearities in the data.
  - **No Model Training:** These models do not require an explicit training phase, making them easy to implement and update.
  - **Computational Efficiency:** While prediction time can be slow with large datasets, the training phase is typically fast since there's no explicit model training involved.
  - **Interpretable:** Predictions can often be explained by examining the closest instances in the training data.

Instance-Based Learning

- **Limitations:**
  - **Computational Complexity:** Predictions can be slow, especially with large datasets, as they involve calculating distances between the new instance and all training instances.
  - **Sensitivity to Noise:** Instance-based models can be sensitive to noisy or irrelevant features in the dataset.
  - **Memory Requirements:** Storing the entire training dataset may be impractical for very large datasets.

Instance-Based Learning

- **Best Practices:**
  - **Feature Scaling:** Scaling features is important to ensure that all dimensions contribute equally to the distance calculation.
  - **Cross-Validation:** Evaluate model performance using techniques like k-fold cross-validation to choose optimal hyperparameters (e.g. number of similar examples to use) and assess generalization performance.
  - **Handling Imbalanced Data:** Address class imbalances by adjusting the weighting of instances.

K-Nearest Neighbors (KNN)

- KNN is a simple instance-based learning algorithm used for both classification and regression tasks.

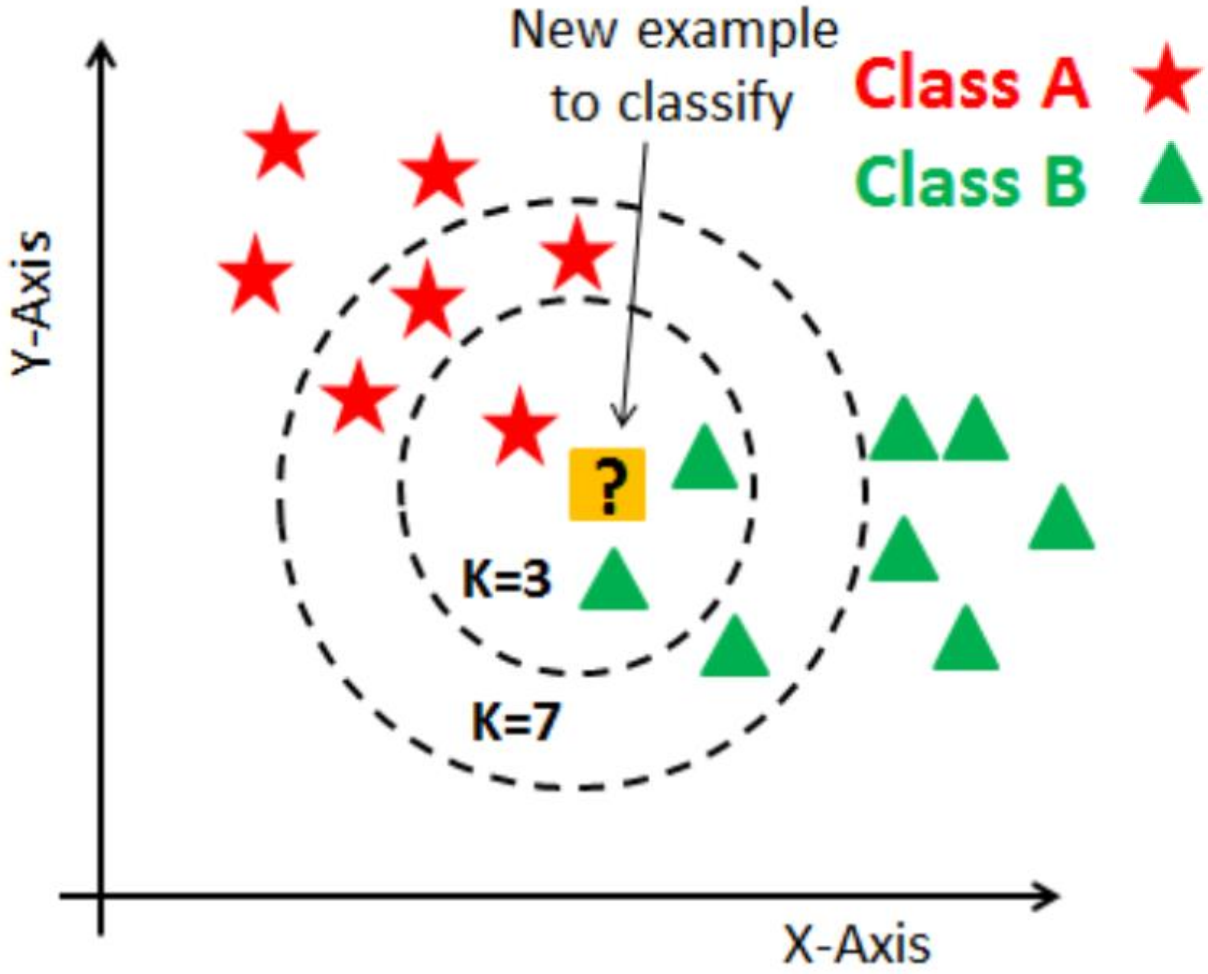
- In the training phase, it **stores the entire training dataset**;

- During prediction, it **computes the distance** between the input data points and all training examples using a distance metric (e.g. Euclidean distance).

- The algorithm identifies the K nearest neighbors to the input data points and uses their labels to make predictions. It uses **majority voting** for classification and **averaging** for regression.

K-Nearest Neighbors (KNN)

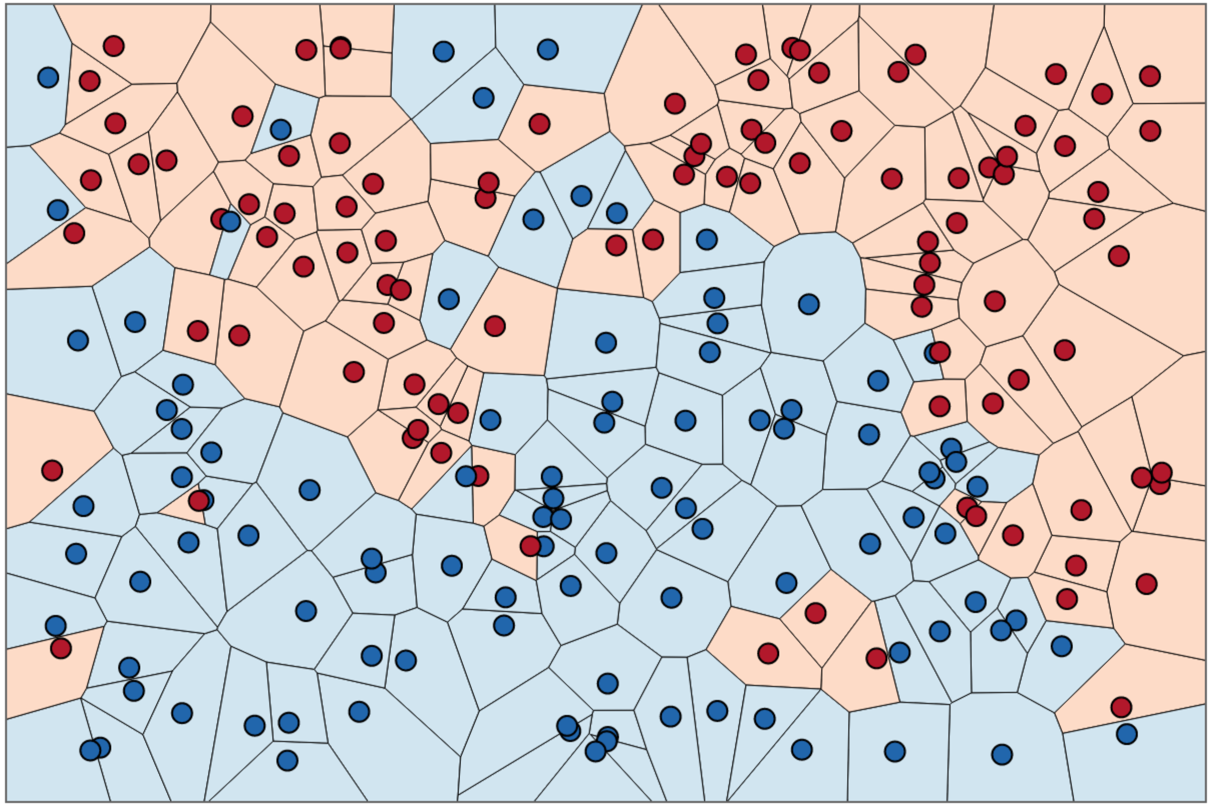
- How would you classify the new example?
  - With **k=3**?
  - With **k=7**?



KNN – Decision Boundaries

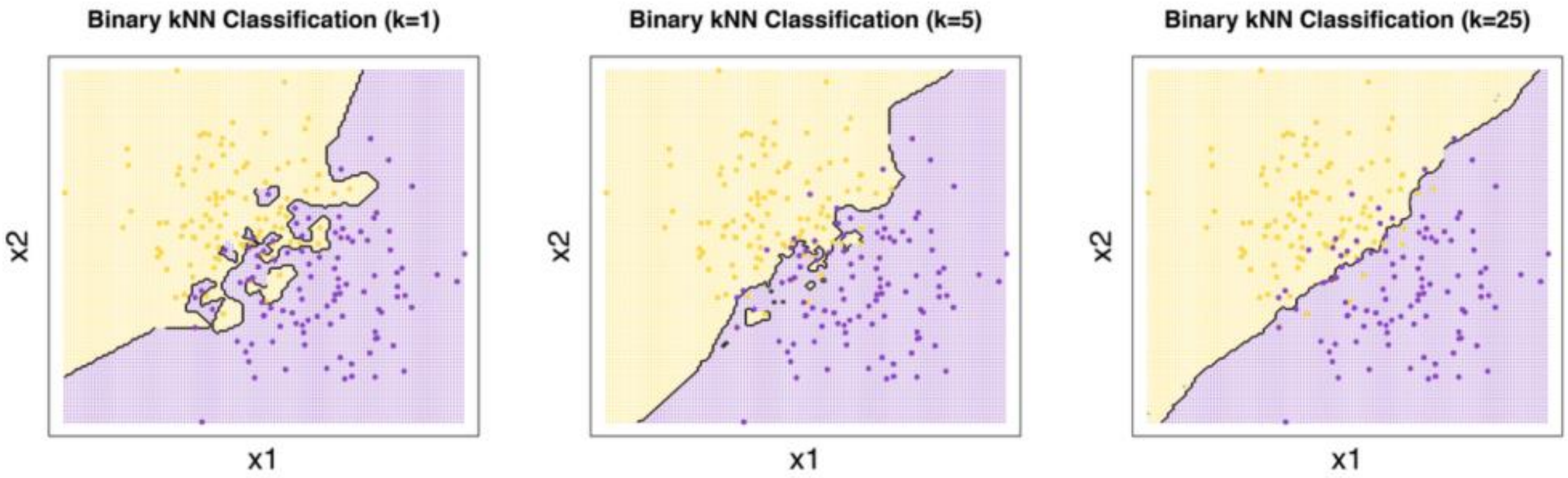
- The nearest neighbor algorithm does not directly calculate decision boundaries; however, they can be inferred from the the training data.

- **Voronoi diagram**:
  - Show how the input space is divided into classes
  - Each line is equidistant to points of different classes



KNN – Decision Boundaries

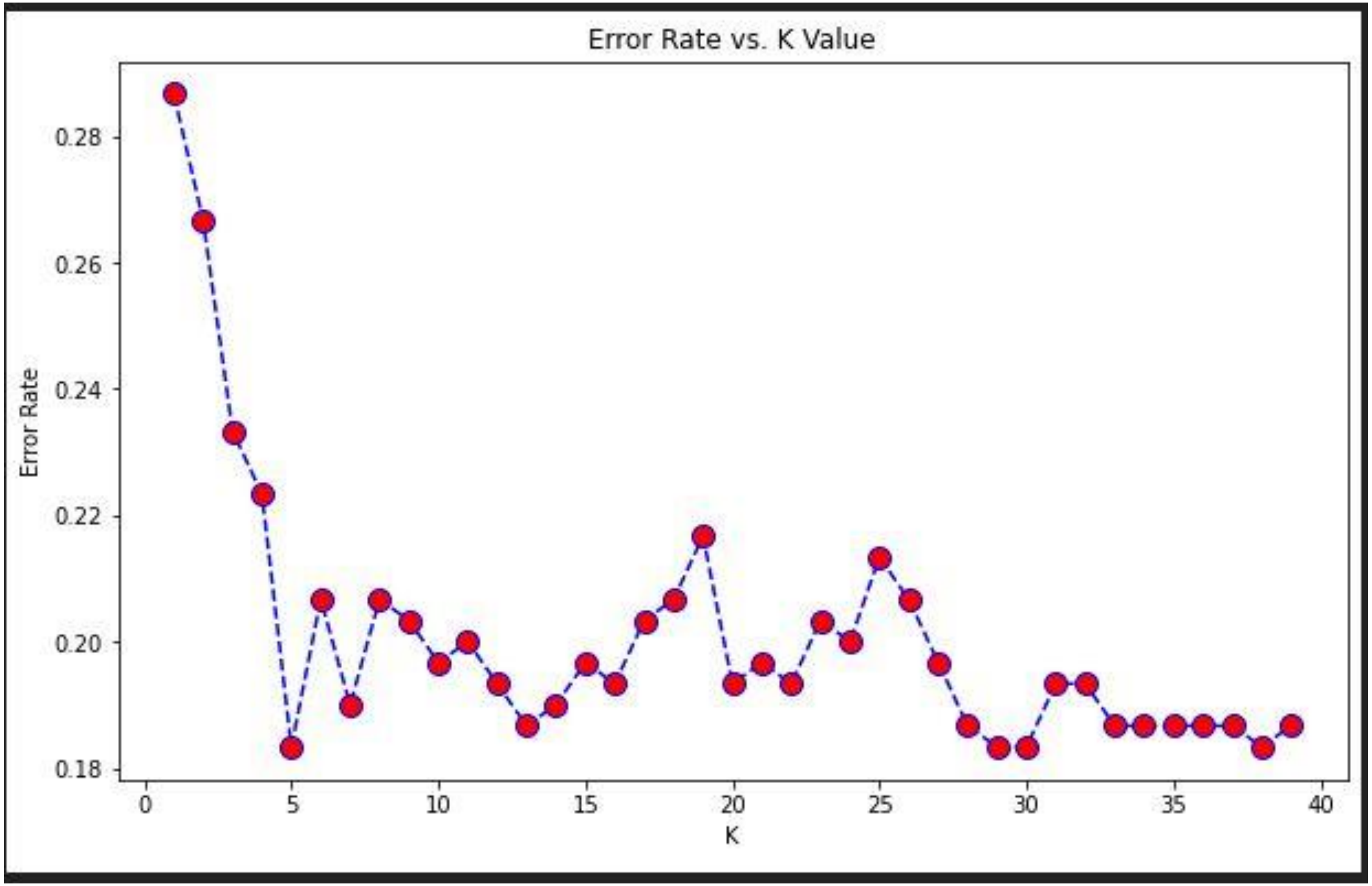
- **Impact of k**



KNN – How to Choose k?

- A **larger k can potentially improve performance**.
- However, setting k excessively large may involve considering samples that are **not true neighbors**, leading to decreased accuracy.
- Error estimation methods (like holdout and cross-validation) can help in finding the **optimal k**.
- It is common to use $k = \sqrt{n}$, where n is the number of training examples.

KNN – Decision Boundaries



Probabilistic Models

- Probabilistic models are mathematical frameworks used to represent **uncertainty**.

- These models aim to capture the underlying **probability distributions of the data**.

- **Naive Bayes** is a probabilistic classifier based on **Bayes' theorem**. It assumes that features are conditionally independent given the class label.

Bayes' Theorem

LIKELIHOOD
The probability of "B" being True, given "A" is True

PRIOR
The probability "A" being True. This is the knowledge.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

POSTERIOR
The probability of "A" being True, given "B" is True

MARGINALIZATION
The probability "B" being True.

Naive-Bayes

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Based on **conditional probabilities** (Bayes' Theorem);

- It calculates the probabilities associated with the belonging of an example to each possible class;

- **Assumptions** (which rarely occur in reality):
    - All features have the same importance;
    - Values from the various features occur independently.

Naive-Bayes

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Types of Naive Bayes:
  - **Multinomial Naive Bayes:** Suitable for classification with discrete features.
  - **Gaussian Naive Bayes:** Assumes features follow a normal distribution.
  - **Bernoulli Naive Bayes:** Works with binary features.

- Advantages:
  - **Simplicity:** Easy to understand and implement.
  - **Efficiency:** Computationally efficient, especially with high-dimensional data.
  - **Robustness:** Performs well even with small datasets and in the presence of irrelevant features.

- Limitations:
  - **Assumption of Independence:** The "naive" assumption of feature independence might not hold in real-world datasets, potentially leading to inaccurate classifications.
  - **Sensitive to Input Data Quality:** Naive Bayes can be sensitive to irrelevant features or features with high correlation, which may degrade classification performance.

Multinomial Naive-Bayes

- Step 1: **Get the data**

Outlook	Humidity	Wind	Run
Sunny	High	Weak	No
Overcast	High	Strong	No
Rainy	High	Weak	Yes
Rainy	Normal	Weak	No
Sunny	Normal	Weak	Yes
Sunny	High	Weak	Yes
Sunny	High	Weak	Yes
Rainy	Normal	Strong	No
Overcast	High	Weak	Yes
Sunny	High	Weak	Yes
Rainy	High	Weak	No
Overcast	Normal	Strong	No
Overcast	High	Weak	Yes
Sunny	High	Weak	Yes

Multinomial Naive-Bayes

- Step 2: **Convert data to a frequency tables**

Frequency Table		Run	
		Yes	No
Outlook	Sunny	5	1
	Overcast	2	2
	Rainy	1	3

Frequency Table		Run	
		Yes	No
Humidity	High	7	3
	Normal	1	3

Frequency Table		Run	
		Yes	No
Wind	Strong	0	3
	Weak	9	2

Multinomial Naive-Bayes

- Step 3: **Calculate the prior probability and likelihood**

Likelihood Table		Run		
		Yes	No	
Outlook	Sunny	5/8	1/6	6/14
	Overcast	2/8	2/6	4/14
	Rainy	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run		
		Yes	No	
Humidity	High	7/8	3/6	10/14
	Normal	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run		
		Yes	No	
Wind	Strong	0/9	3/5	3/14
	Weak	9/9	2/5	11/14
		9/14	5/14	

Multinomial Naive-Bayes

- Step 4: **Apply the Bayes' Theorem**

- Let’s say you want to focus on the likelihood that you go for a run given that it’s sunny outside.

P(Sunny|Yes) = 5/8 = 0.625

Likelihood Table		Run		
		Yes	No	
Outlook	Sunny	5/8	1/6	6/14
	Overcast	2/8	2/6	4/14
	Rainy	1/8	3/6	4/14
		8/14	6/14	

P(Sunny) = 6/14 = 0.428

P(Yes) = 8/14 = 0.571

- P(Yes|Sunny) = P(Sunny|Yes) * P(Yes) / P(Sunny) = 0.625 * 0.571 / 0.428 = 0.834

Multinomial Naive-Bayes

Caution with probabilities of 0. Generally a constant is added to all counts.

Likelihood Table		Run		
		Yes	No	
Outlook	Sunny	5/8	1/6	6/14
	Overcast	2/8	2/6	4/14
	Rainy	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run		
		Yes	No	
Humidity	High	7/8	3/6	10/14
	Normal	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run		
		Yes	No	
Wind	Strong	$\boxed{0/9}$	3/5	3/14
	Weak	9/9	2/5	11/14
		9/14	5/14	

- Outlook: Rainy
- Humidity: Normal
- Wind: Weak
- **Run: ?**

Bayes Theorem $\longrightarrow$ $$P(A \mid x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n \mid A)\, P(A)}{P(x_1, \ldots, x_n)}$$

We can drop the denominator from the formula while assuming feature independence

Naïve Bayes $\longrightarrow$ $$P(A \mid x_1, \ldots, x_n) = P(x_1 \mid A) \cdot P(x_2 \mid A) \cdot P(x_i \mid A)\, P(A)$$

- P(Yes|Rainy, Normal, Weak) = P(Rainy|Yes)*P(Normal|Yes)*P(Weak/Yes)*P(Yes)
  = 1/8 * 1/8 * 9/9 * 8/14 = 0.0089
- P(No|Rainy, Normal, Weak) = P(Rainy|No)*P(Normal|No)*P(Weak/No)*P(No)
  = 3/6 * 3/6 * 2/5 * 6/14 = 0.042

P(Yes) = 0.0089 / (0.0089 + 0.042) = 0.175

P(No) = 0.042/ (0.0089 + 0.042) = 0.825

Multinomial Naive-Bayes

Likelihood Table		Run: Yes	Run: No	
Outlook	Sunny	5/8	1/6	6/14
	Overcast	2/8	2/6	4/14
	Rainy	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run: Yes	Run: No	
Humidity	High	7/8	3/6	10/14
	Normal	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run: Yes	Run: No	
Wind	Strong	0/9	3/5	3/14
	Weak	9/9	2/5	11/14
		9/14	5/14	

Bayes Theorem $\longrightarrow$ $$P(A \mid x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n \mid A)\, P(A)}{P(x_1, \ldots, x_n)}$$

$\downarrow$

Naïve Bayes $\longrightarrow$ $$P(A \mid x_1, \ldots, x_n) = P(x_1 \mid A) \cdot P(x_2 \mid A) \cdot P(x_i \mid A)\, P(A)$$

- Outlook: Sunny
- Humidity: High
- Wind: Weak
- **Run: ?**

Multinomial Naive-Bayes

Likelihood Table		Run: Yes	Run: No	
Outlook	Sunny	5/8	1/6	6/14
	Overcast	2/8	2/6	4/14
	Rainy	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run: Yes	Run: No	
Humidity	High	7/8	3/6	10/14
	Normal	1/8	3/6	4/14
		8/14	6/14	

Likelihood Table		Run: Yes	Run: No	
Wind	Strong	0/9	3/5	3/14
	Weak	9/9	2/5	11/14
		9/14	5/14	

- Outlook: Sunny
- Humidity: High
- Wind: Weak
- **Run: ?**

Bayes Theorem ⟶ $$P(A \mid x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n \mid A)\, P(A)}{P(x_1, \ldots, x_n)}$$

We can drop the denominator from the formula while assuming feature independence

Naïve Bayes ⟶ $$P(A \mid x_1, \ldots, x_n) = P(x_1 \mid A) \cdot P(x_2 \mid A) \cdot P(x_i \mid A)\, P(A)$$

- P(Yes|Sunny, High, Weak) = P(Sunny|Yes)*P(High|Yes)*P(Weak/Yes)*P(Yes)
  = 5/8 * 7/8 * 9/9 * 8/14 = 0.3125
- P(No|Sunny, High, Weak) = P(Sunny|No)*P(High|No)*P(Weak/No)*P(No)
  = 1/6 * 3/6 * 2/5 * 6/14 = 0.0143

P(Yes) = 0.3125 / (0.3125 + 0.0143) = 0.956

P(No) = 0.0143 / (0.3125 + 0.0143) = 0.044

Resources

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Webb, G. I. (2011). Naïve Bayes. In Encyclopedia of Machine Learning (pp. 713–714). Springer US. https://doi.org/10.1007/978-0-387-30164-8_576

- Kramer, O. (2013). K-Nearest Neighbors. In Dimensionality Reduction with Unsupervised Nearest Neighbors (pp. 13–23). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38652-7_2

- https://www.youtube.com/watch?v=HVXime0nQeI

- https://www.youtube.com/watch?v=O2L2Uv9pdDA