

Machine Learning

Prepared by Dr. Yasser Abdelhamid

DS 620
Machine Learning

Lecture 5

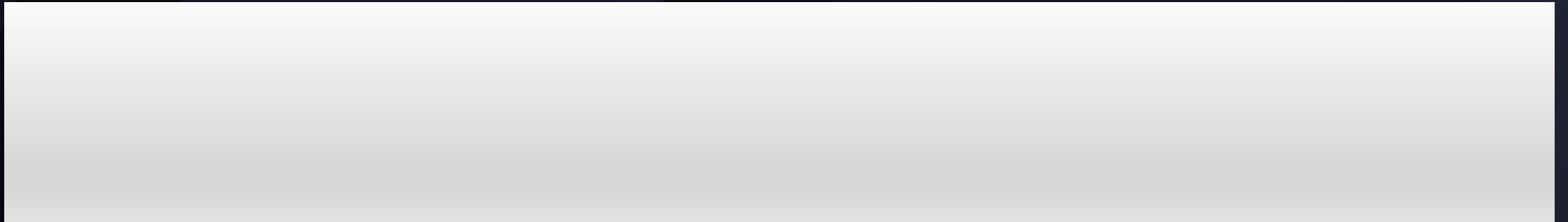




OUTLINE

- ◉ Classification
- ◉ Classification algorithms
- ◉ K-Nearest neighbors (KNN)
- ◉ Distance measures

CLASSIFICATION



CLASSIFICATION

- ❖ Classification is a fundamental task in machine learning that involves categorizing or assigning predefined labels to input data based on their features.
- ❖ It aims to build models that can learn patterns and relationships from labeled training data and use that knowledge to predict the labels of new, unseen data points.

CLASSIFICATION

- ❖ In classification, the data is typically represented as a set of features or attributes, which are numerical or categorical variables that describe the characteristics of each data point.
- ❖ The labels, also known as classes or categories, represent the different target outcomes we want to predict.

CLASSIFICATION

- ❖ The goal of a classification algorithm is to automatically learn the decision boundaries or decision rules that separate different classes in the feature space.

CLASSIFICATION

- ❖ Supervised learning is the most common approach in classification, where the algorithm learns from a labeled training dataset.
- ❖ During the training phase, the algorithm analyzes the relationship between the input features and their corresponding labels to create a model that can generalize to new, unseen data.

CLASSIFICATION ALGORITHMS

❖ There are various algorithms used for classification, each with its own strengths, assumptions, and performance characteristics. Some popular classification algorithms include:

- **K-Nearest Neighbors (KNN)**
- **Decision Trees**
- **Naive Bayes**
- **Logistic Regression**
- **Random Forest**
- **Support Vector Machines (SVM)**

LOGISTIC REGRESSION

- ❖ A linear model that uses a logistic function to estimate the probability of each class.
- ❖ It is widely used for binary classification, but can also be extended to handle multiclass problems.

DECISION TREES

- ❖ Tree-based models that recursively split the feature space based on the values of different features.
- ❖ Decision trees are intuitive and can handle both numerical and categorical data.

RANDOM FOREST

- ❖ An ensemble of decision trees where each tree is trained on a random subset of the data.
- ❖ Random Forest combines the predictions of multiple trees to make a final classification decision.

SUPPORT VECTOR MACHINES (SVM)

- ❖ A powerful algorithm that finds an optimal hyperplane to separate the classes while maximizing the margin between them.

K-NEAREST NEIGHBORS (KNN)

- ❖ A non-parametric algorithm that classifies new data points based on the majority vote of their k nearest neighbors in the feature space.

NAIVE BAYES

- ❖ A probabilistic algorithm that applies Bayes' theorem with strong independence assumptions between features.
- ❖ It is computationally efficient and works well with high-dimensional data.

K-NEAREST NEIGHBORS (KNN)

K-NEAREST NEIGHBORS

❖ KNN is an example of Instance-based learning algorithms, compared to model-based algorithms.

K-NEAREST NEIGHBORS

❖ It doesn't learn a discriminative function from the training data but memorizes the training dataset instead.

K-NEAREST NEIGHBORS

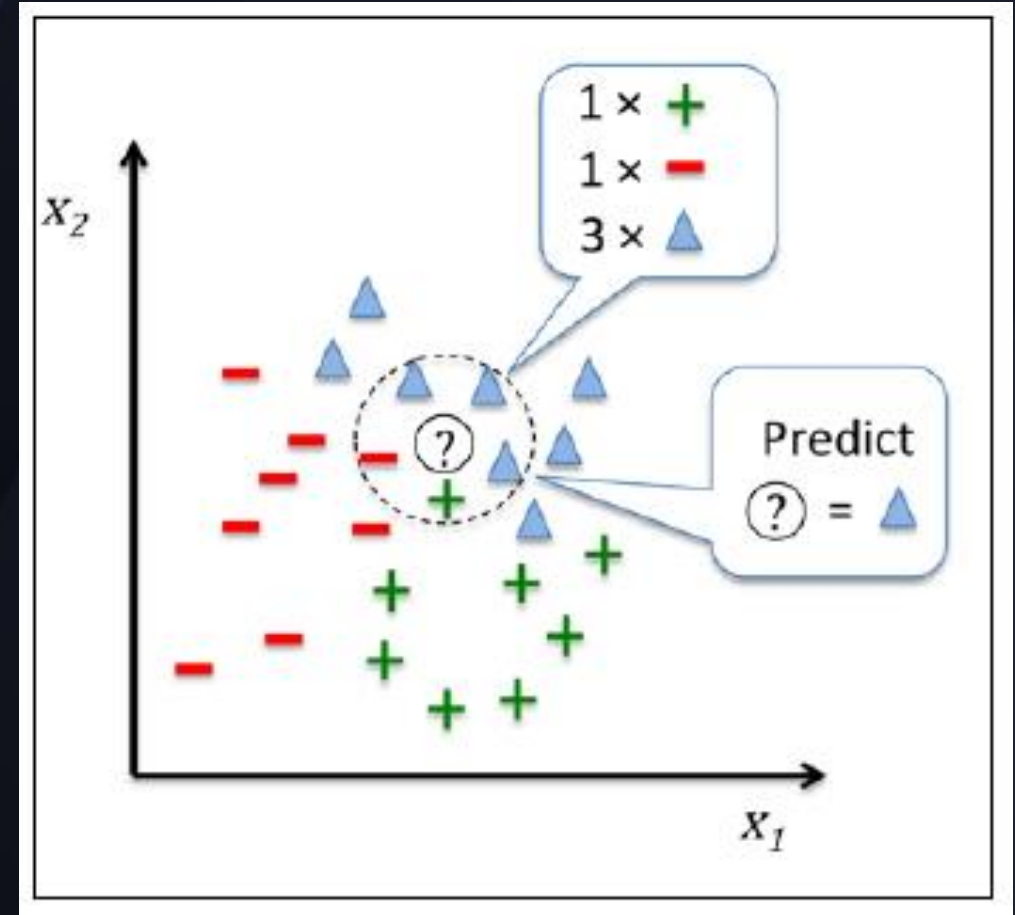
❖ KNN is a non-parametric model. Where in parametric models, we estimate parameters from the training dataset to learn a function that can classify new data points without requiring the original training dataset anymore.

KNN ALGORITHM

1. **Choose the number of k** : that is the number of nearest instances to the sample being classified.
2. **Choose a distance metric**: that is the method to be used to measure the distance between the sample and each of the k nearest instances selected in the previous step.
3. **Find the k nearest neighbours** of the sample that we want to classify.
4. **Assign the sample the class label** that has the majority vote.

EXAMPLE

- ❖ Based on the chosen distance metric, the KNN algorithm finds the k samples in the training dataset that are closest (most similar) to the point that we want to classify.
- ❖ The class label of the new data point is then determined by a **majority vote** among its k nearest neighbours.



ADVANTAGES OF KNN ALGORITHM

SIMPLICITY AND INTUITIVENESS

- ❖ KNN is a simple and easy-to-understand algorithm.
- ❖ Its underlying principle is straightforward:
 - for a given data point, it classifies it based on the labels of its k nearest neighbors in the feature space.
- ❖ This simplicity makes KNN a popular choice for introductory machine learning and classification tasks.

NO TRAINING PHASE

- ❖ Unlike many other machine learning algorithms, KNN does not require an explicit training phase.
- ❖ The algorithm stores the training data in memory, and the classification is performed at runtime based on the proximity of the test data to the training instances.
- ❖ This makes KNN particularly useful when dealing with streaming data or situations where new training instances are frequently added.

NON-PARAMETRIC NATURE

- ❖ KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution.
- ❖ It can effectively handle complex decision boundaries and adapt to different types of data without imposing specific constraints.
- ❖ This flexibility makes KNN suitable for both linear and non-linear classification tasks.

ABILITY TO HANDLE MULTICLASS PROBLEMS

- ❖ KNN can handle multiclass classification problems naturally.
- ❖ By considering the labels of the k nearest neighbors, it determines the majority class among them and assigns the corresponding label to the test data point.
- ❖ This makes KNN versatile and applicable to a wide range of classification scenarios.

ROBUSTNESS TO OUTLIERS

- ❖ KNN is relatively robust to outliers in the data.
- ❖ Outliers have a limited impact on the classification decision since the algorithm focuses on the local neighborhood of data points.
- ❖ Therefore, KNN can handle noisy or unclean datasets without significant degradation in performance.

ADAPTABILITY TO CHANGING DATA

- ❖ KNN can adapt to changes in the underlying data distribution.
- ❖ As new data points are added or existing ones are modified, the neighbors of each data point may change, potentially altering the classification decision.
- ❖ This adaptability makes KNN suitable for dynamic or evolving environments.

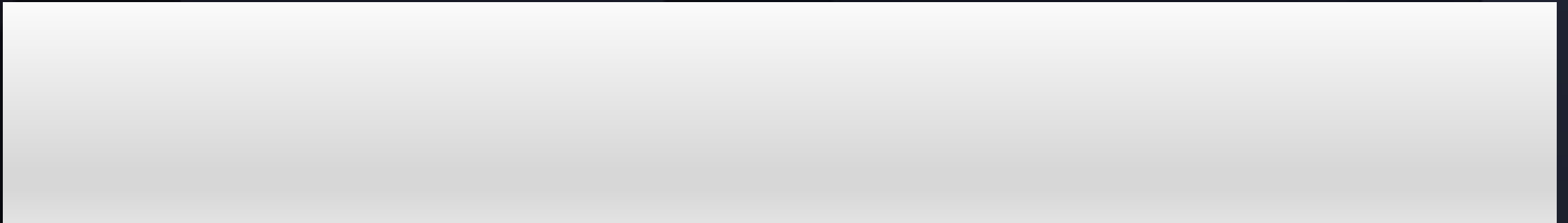
NO MODEL ASSUMPTIONS

- ❖ KNN does not assume any specific model structure.
- ❖ It does not require assumptions about linearity, normality, or other distributional properties of the data.
- ❖ This makes KNN a versatile choice when dealing with data that may not conform to predefined assumptions.

EASY HANDLING OF MISSING DATA

- ❖ KNN can handle missing values in the data effectively.
- ❖ It uses the available features to calculate distances and identify the nearest neighbors, even if some attributes have missing values.
- ❖ This can be advantageous in situations where missing data is common or when imputation techniques are not desirable.

KNN DISADVANTAGES



COMPUTATIONAL COMPLEXITY

- ❖ KNN can be computationally expensive, especially when dealing with large datasets.
- ❖ As the number of training instances grows, the algorithm needs to calculate distances between the test instance and all training instances.
- ❖ This can result in slower prediction times and increased memory requirements.

DEPENDENCE ON DISTANCE METRIC

- ❖ The choice of distance metric plays a crucial role in KNN.
- ❖ Different distance metrics, such as Euclidean distance or Manhattan distance, may yield different results and can have a significant impact on the algorithm's performance.
- ❖ Selecting an appropriate distance metric often requires domain knowledge and experimentation.

CURSE OF DIMENSIONALITY

- ❖ KNN's performance can degrade in high-dimensional feature spaces. This is known as the curse of dimensionality.
- ❖ As the number of dimensions increases, the feature space becomes increasingly sparse, making it difficult to find meaningful nearest neighbors.
- ❖ The presence of irrelevant or noisy features can further exacerbate this issue.

IMBALANCED CLASS DISTRIBUTION

- ❖ KNN can struggle with imbalanced class distributions.
- ❖ If one class significantly outweighs the others, the nearest neighbors of a test instance may be dominated by that majority class, leading to biased predictions.
- ❖ Techniques like resampling or using weighted distances can be employed to address this issue.

NEED FOR OPTIMAL VALUE OF K

- ❖ The choice of the value of k , the number of nearest neighbors, can impact the performance of KNN.
- ❖ A small value of k may result in overfitting and increased sensitivity to noise, while a large value of k may lead to oversmoothing and reduced ability to capture local patterns.
- ❖ Finding the optimal value of k often requires experimentation and model selection techniques.

STORAGE AND MEMORY REQUIREMENTS

- ❖ KNN requires storing the entire training dataset in memory to calculate distances during prediction.
- ❖ As the dataset grows larger, it can consume significant memory resources.
- ❖ Additionally, the need to access the entire dataset during prediction can limit the scalability of the algorithm.

LACK OF INTERPRETABILITY

- ❖ KNN is often considered a "black box" algorithm, providing little insight into the underlying decision-making process.
- ❖ It does not provide explicit explanations or feature importance measures, which can be important in some applications where interpretability is crucial.

DISTANCE MEASURES

MANHATTAN DISTANCE

- ❖ Suppose that we have two points, ***a*** with feature set (x_1, y_1) , and ***b*** with feature set (x_2, y_2) .
- ❖ Manhattan distance = $|x_1 - x_2| + |y_1 - y_2|$

EXAMPLE

❖ Let's have the following dataset about Book rating:

Customer	Book 1	Book 2
C1	5	5
C2	2	5
C3	1	4

❖ We would like to recommend a book to a new customer C4 who has given rates 4,2 to Book1 and Book2 respectively.

EXAMPLE

❖ Using Manhattan's distance, the distance between C4 and the rest of customers is as follows:

Customer	Book 1	Book 2
C1	5	5
C2	2	5
C3	1	4

Customer	Distance
C1	$ 5-4 + 5-2 = 1+3 = 4$
C2	$ 2-4 + 5-2 = 2+3 = 5$
C3	$ 1-4 + 4-2 = 3+2 = 5$

EUCLIDEAN DISTANCE

❖ Suppose that we have two points, ***a*** with feature set (x_1, y_1) , and ***b*** with feature set (x_2, y_2) .

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

❖ Using the book rating dataset, the distance between C1, and C4 is calculated as follows:

$$\sqrt{(5 - 2)^2 + (5 - 4)^2} = \sqrt{3^2 + 1^2} = \sqrt{10} = 3.16$$

EXAMPLE:

- ❖ We have a dataset classified into two classes (+,-), and we need to classify a new instance **x** that has the attributes 9.1, and 11.0.
- ❖ First: calculate the distance between the instance **x**, and the other instances.

$$d(x,y) = \sqrt{(9.1 - 0.8)^2 + (11.0 - 6.3)^2} = \mathbf{4.7}$$

$$d(x,y) = \sqrt{(9.1 - 1.4)^2 + (11.0 - 8.1)^2} = \mathbf{9.6}$$

⋮

$$d(x,y) = \sqrt{(9.1 - 19.6)^2 + (11.0 - 11.1)^2} = \mathbf{10.5}$$

- ❖ Second: select the nearest 5 instances to **x**
- ❖ Third: Count the instances according to their class, **3 (+), 2 (-)**
- ❖ Since we have more votes to (+) class than (-) class, we classify **x** as class (+)

Attribute 1	Attribute 2	Class
0.8	6.3	-
1.4	8.1	-
2.1	7.4	-
2.6	14.3	+
6.8	12.6	-
8.8	9.8	+
9.2	11.6	-
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	-
14.0	19.9	-
14.2	18.5	-
15.6	17.4	-
15.8	12.2	-
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	-
19.6	11.1	+



SUMMARY

- ◉ Classification
- ◉ Classification algorithms
- ◉ K-Nearest neighbors (KNN)
- ◉ Distance measures



Thank You

midocean.edu.km

