Machine Learning

K Nearest Neighbors

K Nearest Neighbors

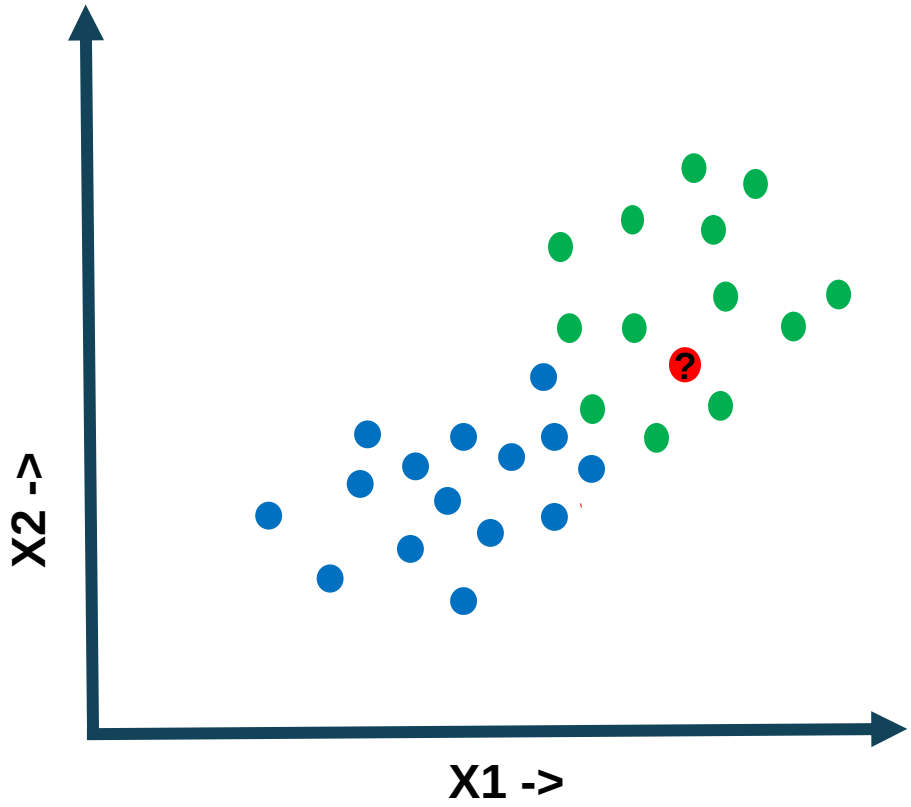
- A supervised learning method
- Though it is a supervised learning method, it is a ‘lazy learner’, i.e. does not construct a model using training data
- Classification is determined based on a majority vote of the nearest neighbors of each point
- Suitable for classification where items in a class tend to be fairly homogenous on the values of attributes
- Not suitable if the data is too noisy or the target classes do not have clear demarcation in terms of attribute values

K Nearest Neighbors

- Nonparametric model: distribution-free tests because no assumption of the data needing to follow a specific distribution
- Commonly used for classification
- Can also be used for regression

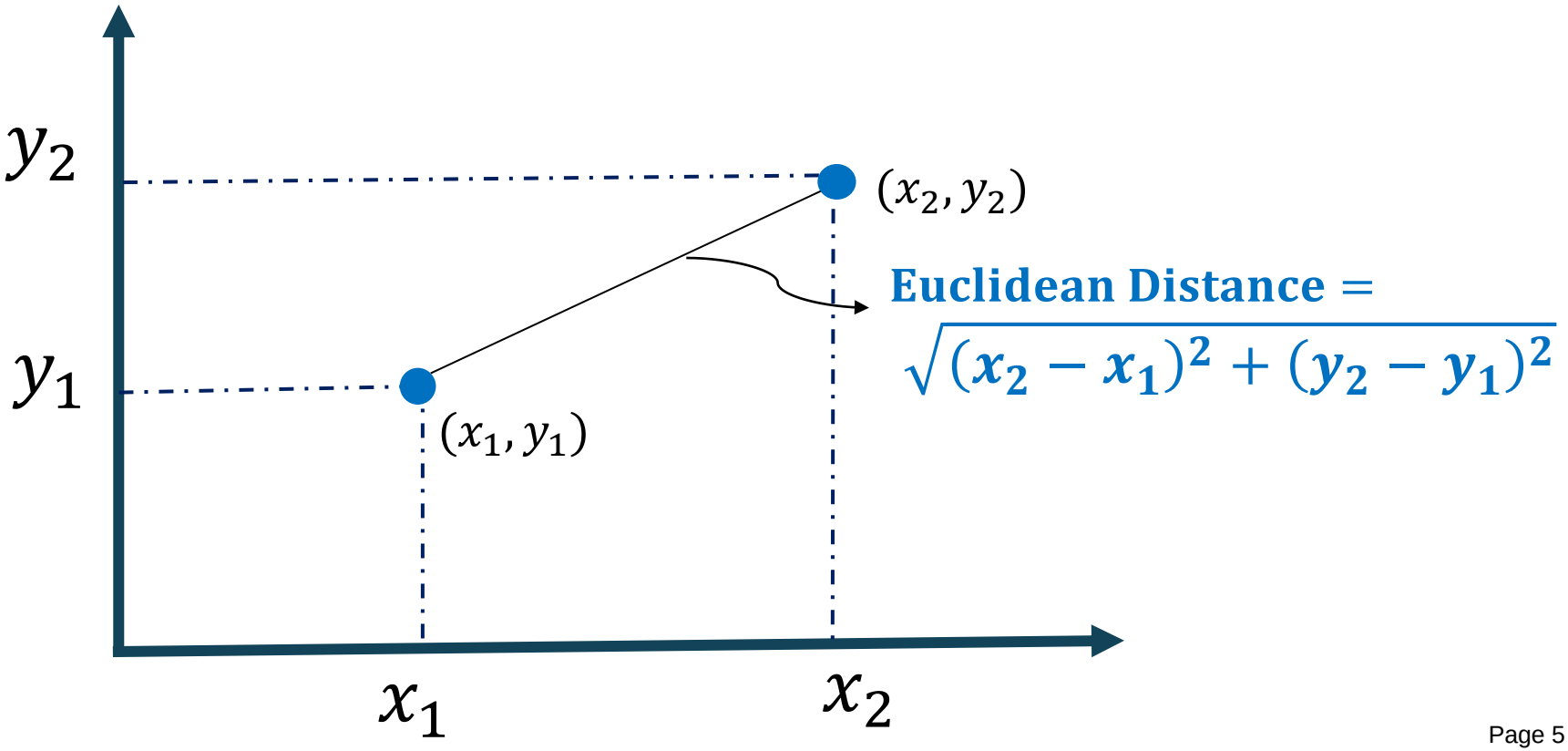
K Nearest Neighbors

- New data point is assigned a class which has the most data points in the nearest neighbors of the point



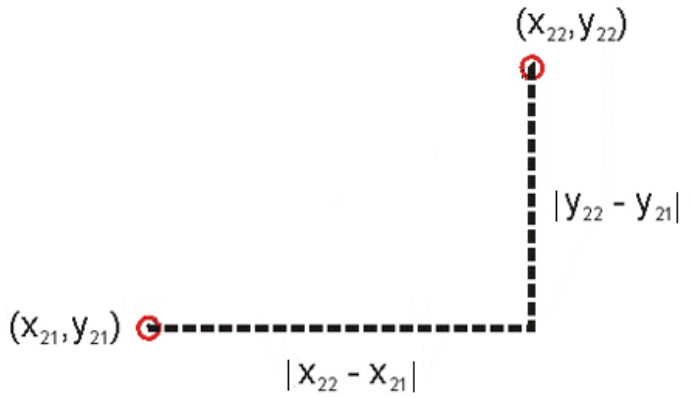
K Nearest Neighbors

- Nearest neighbors are calculated based on shortest distance.
- Most commonly used distance measure is Euclidean distance (default)



K Nearest Neighbors

- Manhattan / Taxi Distance: Also called L1 norm. It is the sum of the differences in each dimension.



- Mahalanobis distance – takes into account the covariance between attributes

- Matching distance (for Boolean data) = number of non-equal dimensions / number of dimensions

K Nearest Neighbors

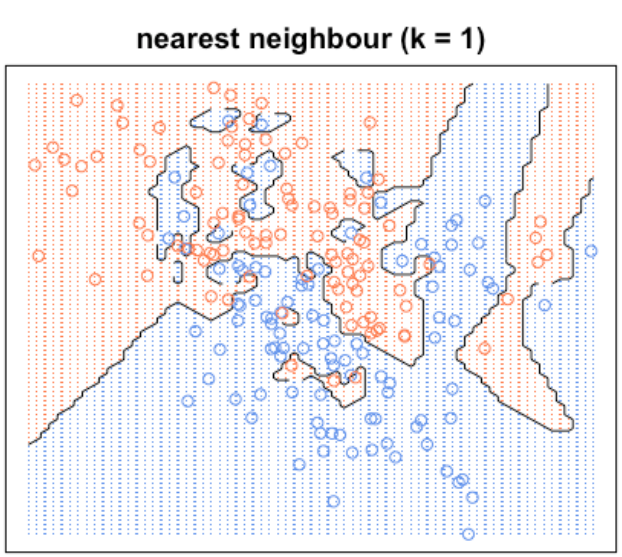
- To ensure all the dimensions have similar scale, it is necessary to normalize the data. Common way to normalize data is
  - Z-score standardization using formula $$z_i = \frac{x_i - \bar{x}}{s}$$
  - Min-max scaler
    - X_std = (X - min) / (max - min)

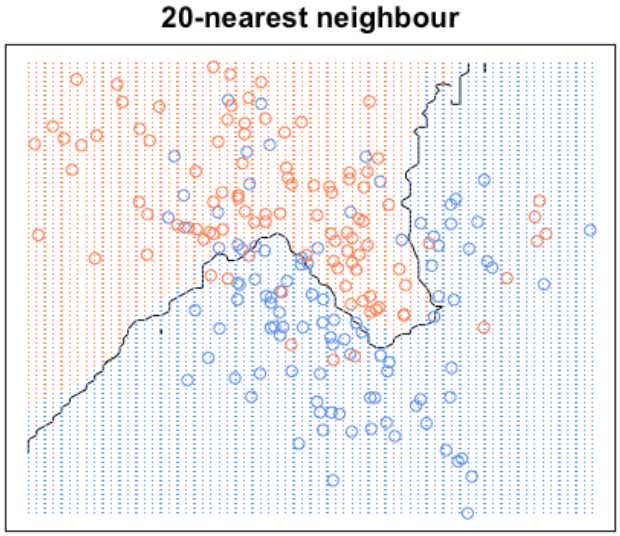
K Nearest Neighbors

- Advantages -
  - Makes no assumptions about distributions of data
  - Easy to understand and implement
  - Not impacted by outliers
  - Able to segregate classes using non-linear boundary

- Dis-advantages -
  - Determining the optimal value of K is a challenge
  - Not effective when the class distributions have large overlap
  - Does not output any models. Calculates distances for every new point. Hence very computation intensive

Selecting value of K





- K is a hyperparameter must be picked in order to get the best possible fit for the data set
- K controls the shape of the decision boundary we talked about earlier
- A small value of k means that noise will have a higher influence on the result

K Nearest Neighbors

- Hands-on exercise

Confusion Matrix

		Predicted		
		A	B	C
Actual	A	15	0	0
	B	0	19	1
	C	0	0	15

- Classification accuracy = correct predictions / total predictions
- Precision is the proportion of the predicted positive cases that were correct.
  - Precision of C = 15 / (15+1)
- Recall is the proportion of positive cases that were correctly identified
  - Recall for B = 19 / (19+1)
- F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Confusion Matrix

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \qquad \textbf{Precision} = \frac{TP}{TP + FP}$$

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

- Note that in binary classification, recall of the positive class is also known as “sensitivity”; recall of the negative class is “specificity”.
- High recall, low precision: This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.
- Low recall, high precision: This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)

Categorical Variables

- Categorical variables must be converted into numbers for **all ML algorithms**
- This can be done using **Label encoder** (or by assigning correctly sequenced numbers to Ordinal data) or **Dummy variables** (covered as part of regression)
- Label encoder are especially useful for **Tree** based algorithms

Thank you

- Prashant Koparkar