Nearest Neighbors algorithms

Lluís Talavera, 2022

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

ETSEIB

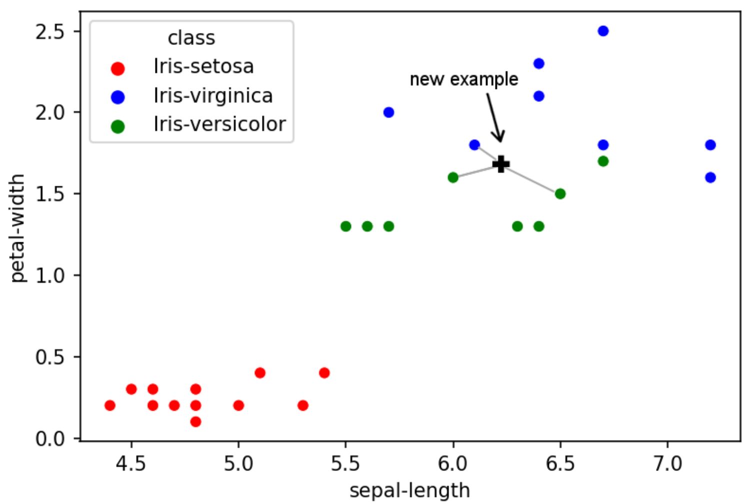
Nearest neighbors

Idea: given a new (unlabeled) example, return the label of the nearest example/s in the training data.

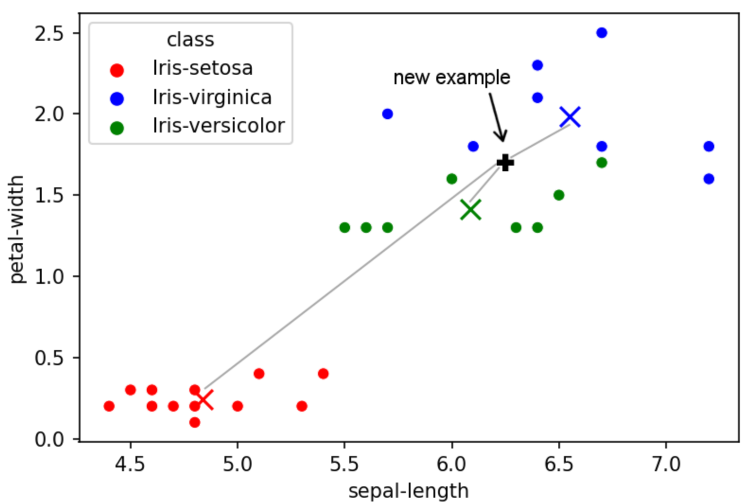
We can:

- summarize examples into centroids: **Nearest Centroid Classifier**
- use the k closest examples: **k-Nearest Neighbor (k-NN)**

3-NN



Nearest Centroid



Distance metrics

We need a metric to define *nearest*. In the following examples n is the number of features in the data:

Euclidean distance

$$d(x,y) = \sqrt{\sum_{1}^{n}(x_i-y_i)^2}$$

Manhattan distance

$$d(x,y) = \sum_{1}^{n} |(x_i-y_i)|$$

Minkowski distance

$$d(x,y) = (\sum_{1}^{n}(x_i-y_i)^p)^{\frac{1}{p}}$$

$p=1$ for Manhattan distance, $p=2$ for Euclidean distance

These metrics are for numerical features. There are others for binary and categorical data.

Nearest Centroid Classifier

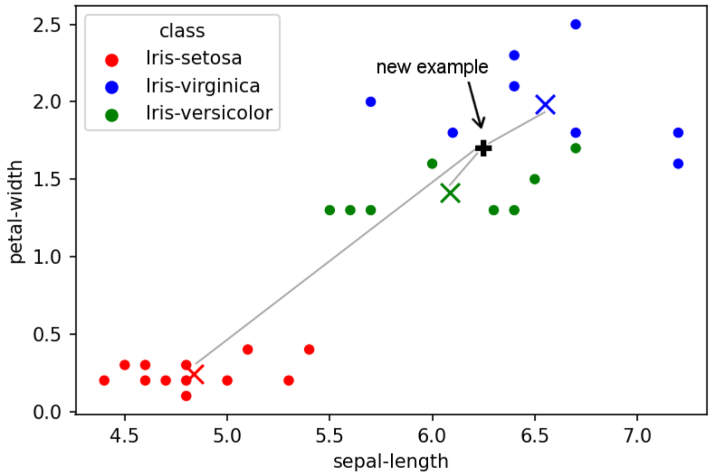
Each class C is represented by a *centroid*, a kind of "summary example" where the value of each feature f_i is the average of the values of f_i across all the examples of C.

Training

> Compute the centroids for each class.

Prediction

> Compute the distance of the new example to each of the centroids and return the class corresponding to the nearest centroid.



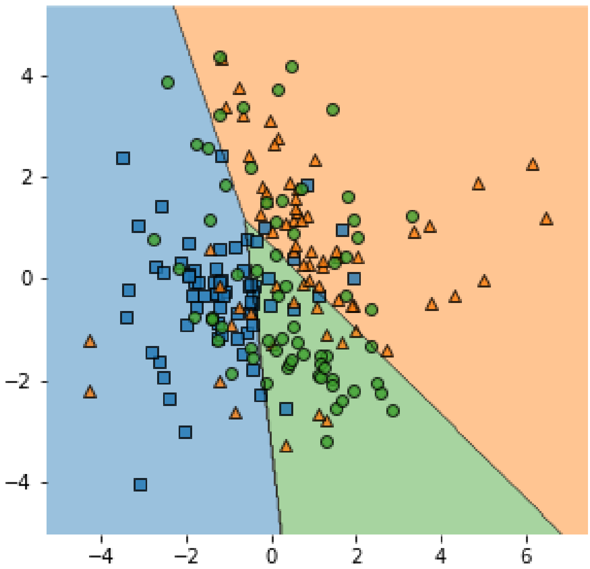
The nearest centroid is from the class *versicolor*

The prediction is *versicolor*

Nearest Centroid summary

- Too simple but a good choice to start experimenting with classification.
- Could be considered a parametric method (estimate the centroids).
- Performance can degrade for high dimensional data, the distance metric can become distorted.
- Feature with larger values will be dominant in the distance metric (e.g. income over age).
- It is a **linear classifier**.

Decision boundaries:



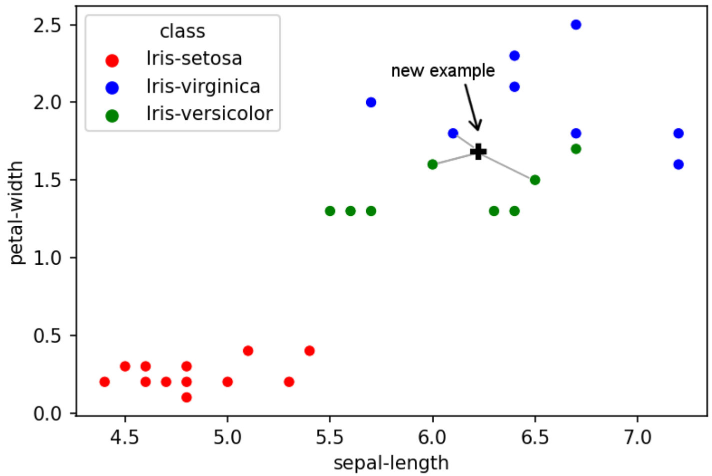
k-NN for classification

Training

Just store all the examples (**lazy learning** method)

Prediction

1. Compute the distance of the new example to each of the stored examples.
2. Select the k nearest examples (smaller distances).
3. Return the **most frequent class** among the previously selected examples (majority vote).

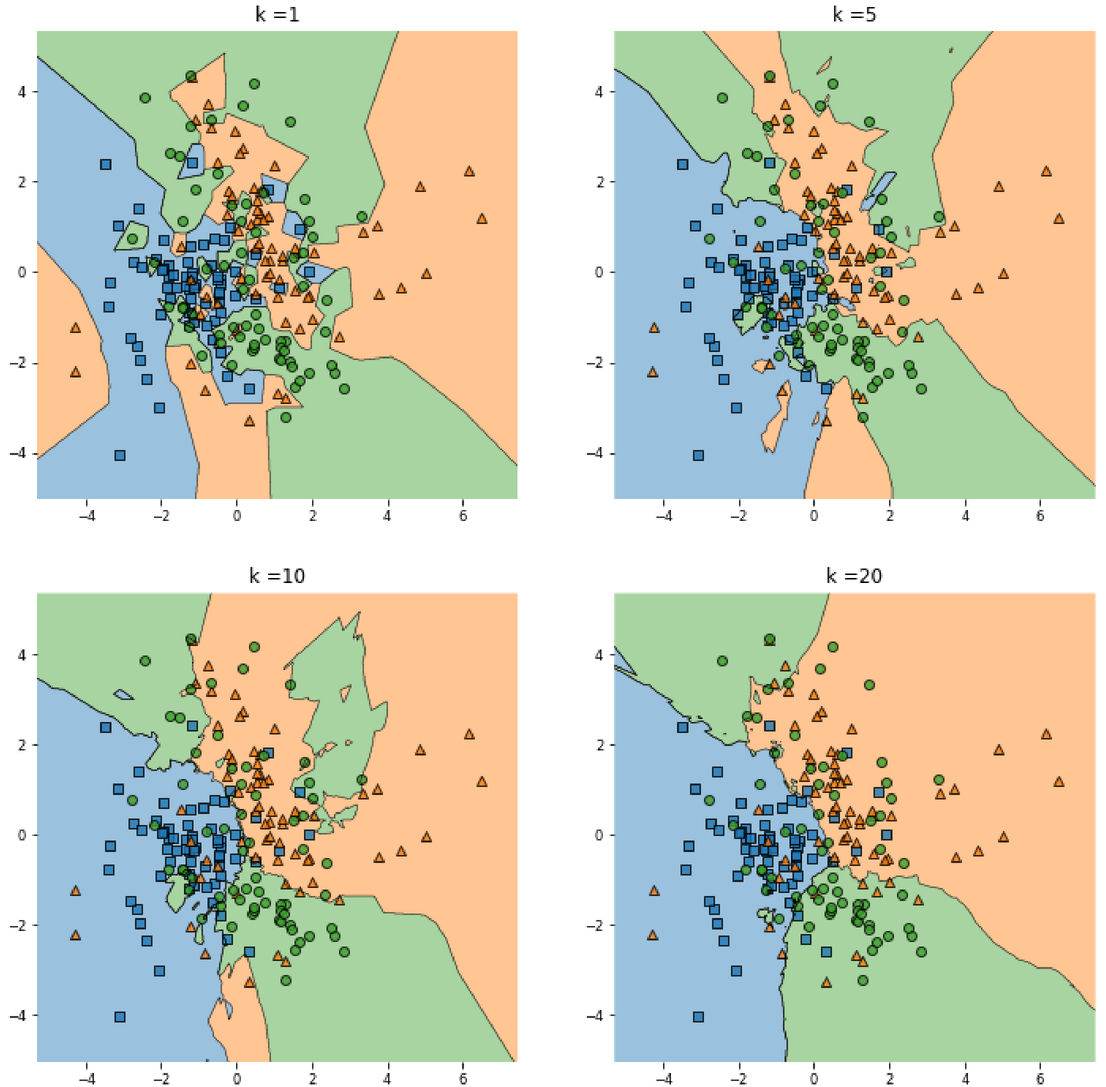


The 3 nearest examples: 2 *versicolor*, 1 *virginica*

The prediction is *versicolor*

If $k=1$, the prediction is *virginica*

k-NN decision boundaries



k-NN for regression

Almost the same algorithm but returning a continuous value.

Training

> Just store all the examples

Prediction

1. Compute the distance of the new example to each of the stored examples.
2. Select the k nearest examples (smaller distances).
3. Return the **mean of the target** among the previously selected examples.

k-NN summary

- It is a non-parametric technique.
- Can be used for both classification and regression.
- Common practice is to choose odd values of k in order to avoid ties.
- The best k depends on the data
- Performance can degrade for high dimensional data, the distance metric can become distorted.
- Feature with larger values will be dominant in the distance metric (e.g. income over age).
- With large datasets, it can be computationally expensive to retrieve the nearest neighbors. Tree data structures are commonly used to organize the examples and improve retrieval.
- Variant: weight points, close neighbors have greater influence

imports with sklearn

```python
from sklearn.neighbors import NearestCentroid
```

```python
from sklearn.neighbors import KNeighborsClassifier
```

```python
from sklearn.neighbors import KNeighborsRegressor
```