Artificial Intelligence: Lab 3

Supervised Machine Learning : k-Nearest Neighbors (k-NN)

Supervised learning is where you have input variables X and an output variable Y and you use an algorithm to learn the mapping function f from the input to the output. $Y = f(X)$ The goal is to approximate the mapping function so well that when you have new input data x that you can predict the output variables Y for that data.

It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

The k-Nearest Neighbors algorithm (k-NN for short) is a very simple technique. The entire training dataset is stored. When a prediction is required, the k-most similar records to a new record from the training dataset are then located. From these neighbors, a summarized prediction is made. Once the neighbors are discovered, the summary prediction can be made by returning the most common outcome (for classification problems) or taking the average (for regression problems).

1 Example 2. k-NN for a Real Dataset

We will illustrate our first supervised learning algorithm using it for breast cancer prediction. For this purpose, we will load a related datset which is included in the standard datasets of `sklearn`.

1. Import the modules `numpy`, `pandas`, `matplotlib` and `seaborn`.
2. Load the dataset. For this purpose, execute the following instructions :

```
from sklearn.datasets import load_breast_cancer
breast_cancer = load_breast_cancer()
print(breast_cancer.DESCR)
```

3. In the sequel, we will not consider all the dataset. We will explain concepts and k-NN algorithm using only two columns (together with predicted class). Execute the following instructions :

```
X = pa.DataFrame(breast_cancer.data, columns=breast_cancer.feature_names)
X = X[['mean area', 'mean compactness']]
y = pa.Categorical.from_codes(breast_cancer.target,
                              breast_cancer.target_names)
y = pa.get_dummies(y, drop_first=True)
```

4. Split the dataset into training and test subsets.
5. Train a k-NN model on the training data.
6. Use your model to predict the classes of the test data.
7. Print the confusion matrix and compute the accuracy of the trained model.