# Introduction for Machine Learning

# NumPy:

Numpy is one of the fundamental packages for sceintific computing in Python. It contains functionality for multidimensional arrays, high level mathmetical functions such as linear algebra operations and fourier transform(The Fourier transform decomposes a function into its constituent frequencies), and pseudorandom number generation.

* Scikit-learn takes in data in the form of NumPy arrays.

# SciPy:

SciPy is a collection of functions for sceintific computing in Python. It provides advanced linear algebra.

* scikit-learn draws from SciPy’s collection of functions for implementin its algorithms.
* The most important part of SciPy for us is scipy.sparse: this provides spars matrics, which are another representation that is used for data in scikit-learn

## Imoportant point: Saprse matrics are used whenever we want to store a 2D array that contains mostly zeros.

# Sparse martrics:

**Sparse matrices** are memory efficient data structures that enable us store large **matrices** with very few non-zero elements **aka sparse matrices**. In addition to efficient storage, **sparse matrix** data structure also allows us to perform complex **matrix** computations.

# CSR or The compressed sparse row

The compressed sparse row (**CSR**) or compressed row storage (CRS) or Yale **format** represents a matrix M by three (one-dimensional) arrays, that respectively contain nonzero values, the extents of rows, and column indices.

thinly dispersed or scattered.

## Sparse means: scattered.

# COO or The coordinate list:

Coordinate (of the same rank; equal) list (**COO**) **COO** stores a list of (row, column, value) tuples. Ideally, the entries are sorted first by row index and then by column index, to improve random access times. This is another **format** that is good for incremental matrix construction.

# Matplotlib

matplot lib is the primary scientific library in Python. it provides functions for making publication-quality visualizations such as line charts, histograms,scatter polts.

## Important note: Bunch object is very similar to dictionary, it contains keys and values.

## Important note: The individual items are called *samples* in machine learning, and their properties are called *features*.

## Important note: The *shape* of the data array is the number of *samples* multiplied by the number of *features*. This is a convention in scikit-learn, and your data will always be assumed to be in this shape.

## Important note: We cannot use the date we used to build the model to evaluate it. Why? Because our model can always simple remember the whole training set, this remembering does not indicate to us whether it will also perform well on new data.

## Important note: Using a test set containing 25% of the data is good rule of thumb.

## Important note: In scikit-learn, data is usually denoted with a captial X, while labels are denoted by a lower case y. Why? We use X because the data is a two-dimensional array (a matrix) and a lowercase y because the target is a one-dimensional array(a vector).

# What is random\_state parameter?

To make sure that we will get the same output if we run the same function several times, we provide the pseudorandom number generator with a fixed seed using the random\_state parameter. This will make the outcome deterministic, so this line will alwayse have the same outcome.

## Pseudo Random Number Generator(PRNG) refers to an algorithm that uses mathematical formulas to produce sequences of random numbers. PRNGs generate a sequence of numbers approximating the properties of random numbers. A PRNG starts from an arbitrary starting state using a seed state.

## Important Point: A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator. For a seed to be used in a pseudorandom number generator, it does not need to be random

# What is the first step before buiding a machine learning model?

Inspect the data with scatter plot.

## Create a dataframe from data in X\_train.

## label the column using the strings in the original data set.

## Create a scatter matrix from the dataframe, color by y\_train

# What is the Estimator classes?

All machine learning models in scikit-learn are implemented in their own classes, which are called **Estimator classes**.

## Important Point: The K-nearest neighbors classification algorithm is implemented in the KneighborsClassifier class in the neighbors module.

## Important Point: Before we can use the model, we need to instantiate the class into an object. This is when we will set any parameters of the model.

## Important Point: Scikit-learn always expects two-dimensional arrays for the data.

# Name a situation that the test set will comes in, why?

In evaluation the method, because we know the right items are in the test set. So, we can make a prediction for each iris in the test data and compare it against its label.

## Important Point. what lable means in the current example? THE KNOWN SPECIES.

# How can we measure how well the model works?

By computing the accuracy, which is the fraction of folwers for which the right species was predicted.

## Important Point. So!! Calculating the accuracy means: the fraction from the right labels(here the right species).

# When we use supervised learning?

Whenever we want to predict a certain outcome from a given input, and we have examples of input/output pairs.

## the goal is making predictions for new, never-before-seen data.

## Important Point: supervised learning often requires human effort to build the training set, but afterward automates and often speeds up an otherwise laborious task.

# What is Unsupervised algorithms?

In unsupervised learning, only the input data is known, and no known output data is given to the algorithm.

## Important Point: In binary classification ‘positive’ is the object of the study.

## Important Point: The goal of regression in tasks is to predict a continuous number, or a floating-point number in programming terms (or real number in mathematical terms).

## Important Point: If a model is able to make accurate predictions on unseen data, we say it is able to generalized from the training set to test set.

## The only measure of whether an algorithm will perform well on the new data is the evaluation on the test set.

# What overfitting means?

Overfitting occures when you fit the model too closely to the particularities of the training set and obtain a model that works well on the training set but is not able to generalize to new data.

## Choosing too simple a model is called underfitting.

# When we can import and instantiate the class?

When we can set parameters, like the number of neighbors to use.

# K-Nearest Neighbors:

## Arguably the simplest learning algorithm.

## building the model consists only of the storing the training dataset.

## To make a predicition for a new data point, the algorithm finds the closest data point in the training dataset-its “nearest neighbors”.

## In the simplest version, the K-NN algorithm only considers exactly one nearest neighbor, which is the closest training data point to the point we want to make a prediction for.

## the prediction is then simply *the known output* for this training point

## Important Point: *The prediction* of the *one-nearest-neighbor algorithm* is the *label* of that point.

## Important Point: Where the K-nearest neighbors algorithm came from? Instead of conidering only the closest neighbor, we can consider an arbitrary number, k, of neighbors.

## Important Point:When we considering more than one neighbor, we use voting to assign a label. This means for each test points, we count how many neighbors belong to class 0 and how many neighbors belong to class 1. We then assign the class that is more frequent: in other words, the majority class among the k-nearest neighbors.

## Important Point: Forge is an example of a synthetic two-class classification dataset, which has two features.

# What is mglearn?

Mglearn is a library of utility functions we wrote for the book of Introduction to ML.

# How we can apply K-nearest-neighbors algorithm using scikit-leanrn?

1. Split our data into train and test set so we can evaluate generalization performance.
2. We import and instantiate the class. **This is when we can set parameters, like the number of neighbors to use.**
3. We fit the classifier using the training set. (For KneighborsClassifier this means storing the datasets, so we can compute neighbors during prediction.
4. We call the **predict method**, which for every data point in the data set, this will compute its nearest neighbors in the training set and finds the most common class among them.
5. To evaluate how well our model generalizes, we can call the **score method**

with the test together with the test data together.

## Important Point: What we can figure out from the results? We can see the model is x% accurate, meaning the model predicted the class correctly for x% of the samples in the test dataset.

# What is decision boundry?

When we color the plane according to the class that would be assigned to a point in that region, this lets us veiw the decision boundry, which is the divide between where the algorithm assigns class 0 versus where it assigns class 1.

## Important Point: Using few neighbors corresponds to high model complexity.

## Importan Point: The prediction using a single neighbor is just the target value of the nearest neighbor.

## When using multiple nearest neighbors, the prediction is the average, or mean, of the relevant neighbore.