Cool Kids Coding School

Problem Solving with Python
Lesson 10: Introduction to AI (KNN Algorithm)

1.0 What is AI?

Before we start talking about the KNN algorithm we need to answer a question first. What is AI, exactly?

In the broadest sense, AI refers to machines that can learn, reason, and act for themselves. They can make their own decisions when faced with new situations, in the same way that humans and animals can.

As it currently stands, the vast majority of the AI advancements and applications you hear about refer to a category of algorithms known as machine learning.

Machine-learning algorithms use statistics to find patterns in massive amounts of data. And data, here, encompasses a lot of things—numbers, words, images, clicks, what have you. If it can be digitally stored, it can be fed into a machine-learning algorithm.

Machine learning is the process that powers many of the services we use today—recommendation systems like those on Netflix, YouTube, and Spotify; search engines like Google and Baidu; social-media feeds like Facebook and Twitter; voice assistants like Siri and Alexa. The list goes on.

In all of these instances, each platform is collecting as much data about you as possible—what genres you like watching, what links you are clicking, which statuses you are reacting to—and using machine learning to make a highly educated guess about what you might want next. Or, in the case of a voice assistant, about which words match best with the funny sounds coming out of your mouth.

Frankly, this process is quite basic: find the pattern, apply the pattern. But it pretty much runs the world. That's in big part thanks to an invention in 1986.

```
The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised
machine learning algorithm that can be used to solve both classification and
regression problems.
```

A supervised machine learning algorithm (as opposed to an unsupervised machine learning algorithm) is one that relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data.

Supervised machine learning algorithms are used to solve classification or regression problems.

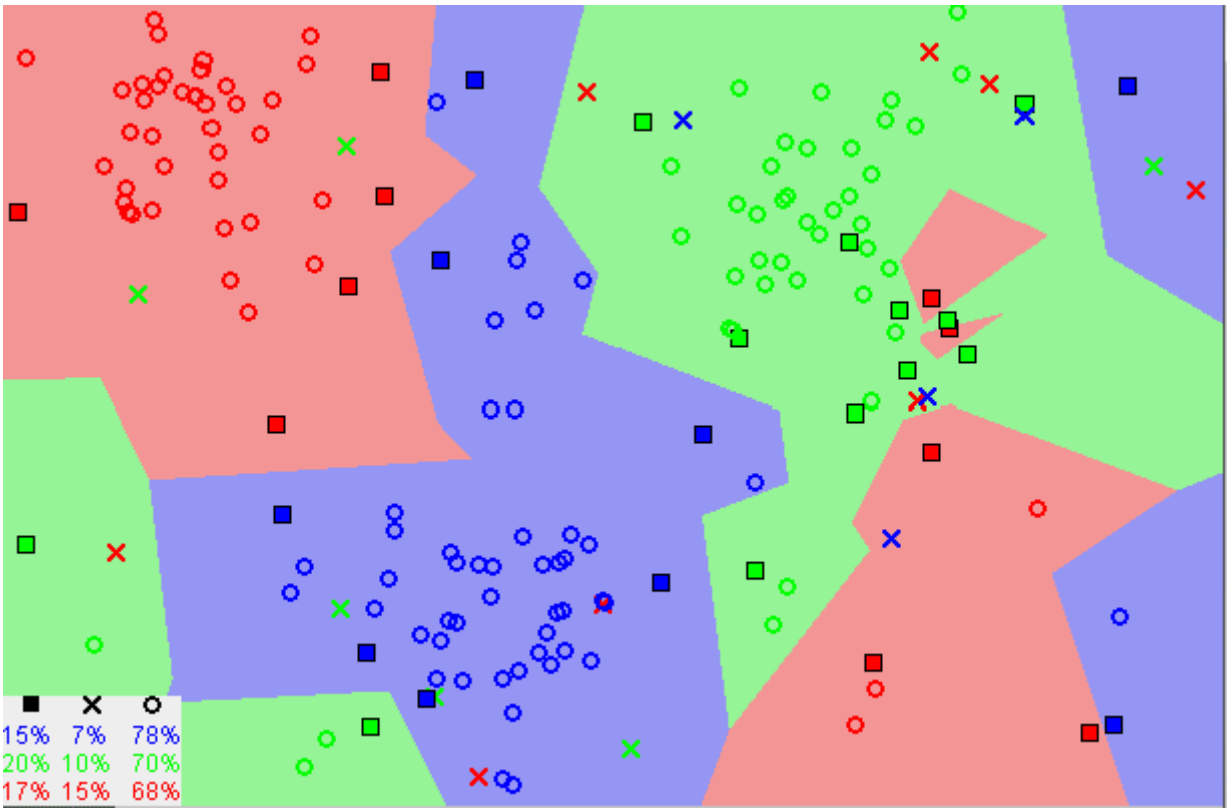
A classification problem has a discrete value as its output.

A regression problem has a real number (a number with a decimal point) as its output.

2.0 K-Nearest Neighbors

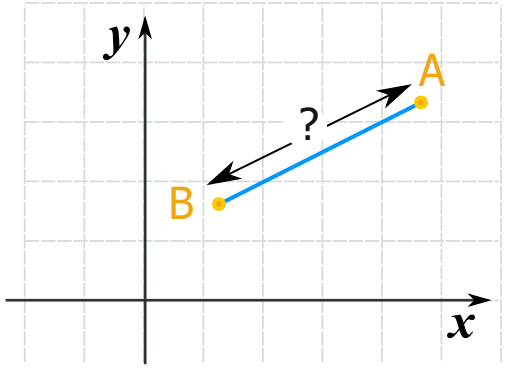
The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

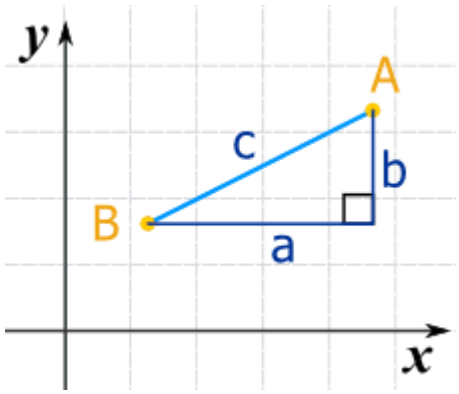
```
“Birds of a feather flock together.”
```

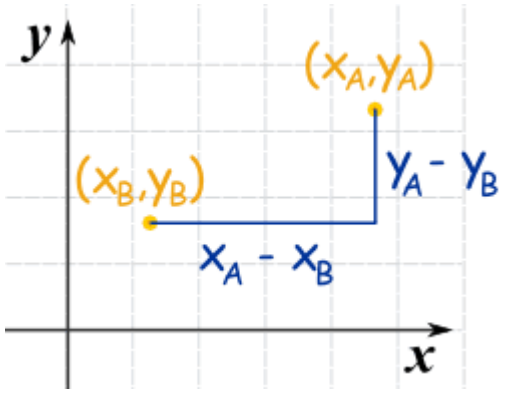


Notice in the image above that most of the time, similar data points are close to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

2.1 Prerequisite: Distance between two points







Distance between 2 points

3.0 The KNN Algorithm

- Load the data
- Initialize K to your chosen number of neighbors
- For each example in the data
    - Calculate the distance between the query example and the current example from the data.
    - Add the distance and the index of the example to an ordered collection
- Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
- Pick the first K entries from the sorted collection
- Get the labels of the selected K entries
- Return the mode of the K labels

4.0 The Data

For this exercise we are going to use a classic machine learning dataset.

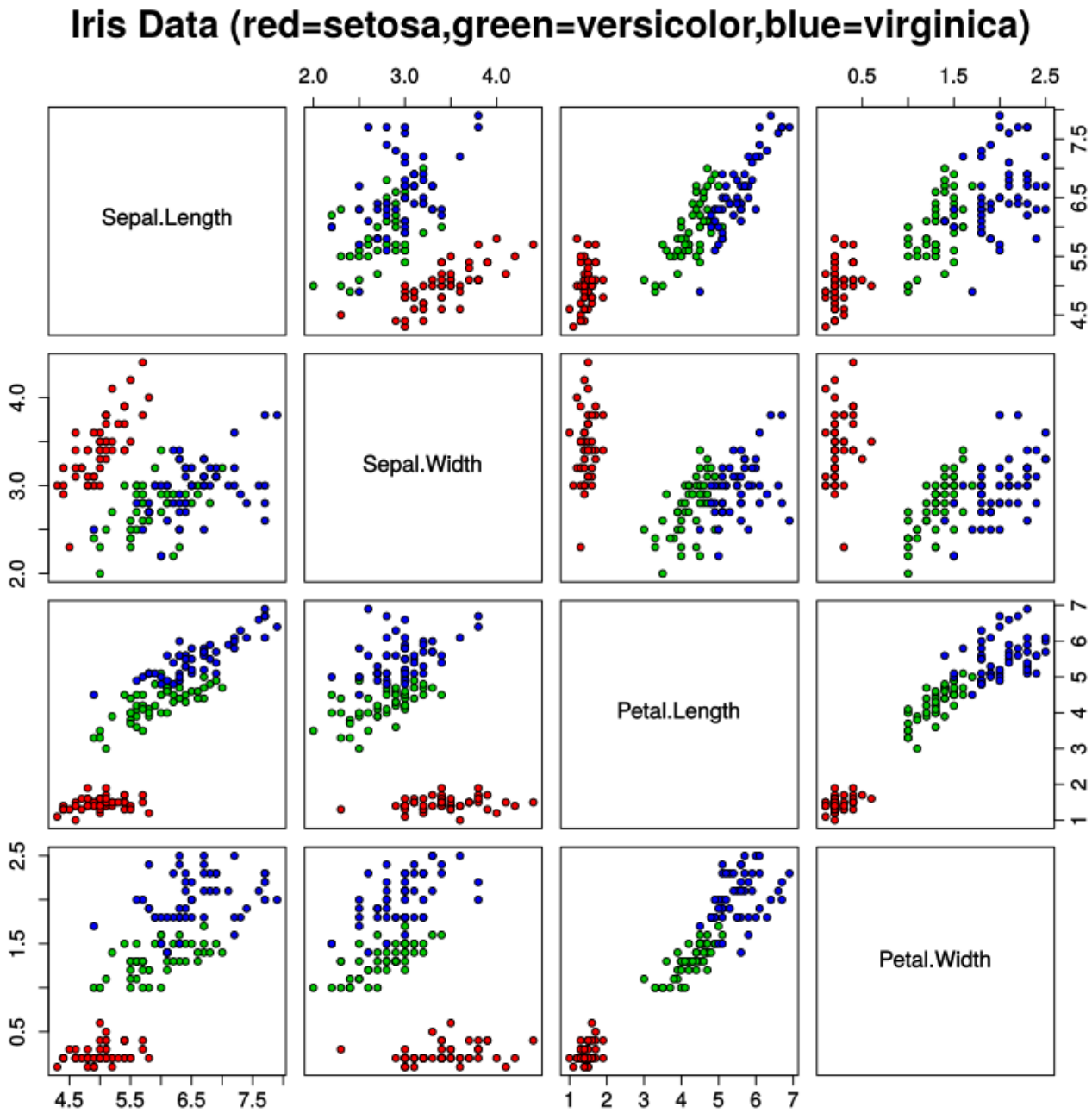
The Iris flower data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher.

The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

The dataset contains a set of 150 records under five attributes - petal length, petal width, sepal length, sepal width and species.

```
5.5,3.5,1.3,0.2,Iris-setosa
7.7,2.6,6.9,2.3,Iris-virginica
7.2,3.6,6.1,2.5,Iris-virginica
```

Looking at a scatterplot of the data we see there is some order.



Ok, the last unanswered part before we start coding is "How do we choose that magical k"? The answer is idk. Smaller k values in most cases is highly affected to noise in the dataset - this is called a model with a high variance or simply overfitted model. Bigger k values lead to bigger bias of the model meaning that it would ignore the training dataset. The general approach is to use $k=\sqrt{N}$, where N is the size of the training dataset. It's also useful to always keep this number odd - to ensure the majority during voting for classification.

Any Questions?

for any questions contact hw_help@coolkidscodingschool.com