

Hello guys, hope you enjoyed the previous video.

As promised in the deliverables of the course, we will now implement one of the easiest machine learning algorithms. I hope you haven't forgotten about the os module of python we learnt in the earlier weeks.

Today we are going to build our first Image Classifier. In order to do so we'll implement a very simple machine learning algorithm named k-nn. K-nn stands for k-nearest neighbor and by now you would know that it is a supervised learning algorithm. We will use it for image classification. In plain English this algorithm goes as, "Tell me who your neighbors are, and I'll tell you who you are". It is similar to saying, if you live in a place where Gujjus live, its likely that you are a Gujju. I hope this doesn't reach Modi.

Let me make it more clear with an example.
Say we have a dataset consisting of images of dogs, cats and pandas.

<Show knn0.png>

Under the primary assumption that images with similar visual contents lie close together,
given a query image

<Show knn1.png>

we should be able to tell whether the query image is of a dog, a cat or a panda.

<Show knn2.png>

We do this by using the distance metric such as Manhatttan Distance or Euclidean Distance etc.

Let us see what happens if we choose $k=1$.

<Show k_1.png>

Let's see what happens if we choose $k=3$.

<Show k_3.png>

Note that the results vary depending on the value of k chosen. And since value of k is something that cannot be learnt but is to be chosen by the implementer, it is referred as the Hyper-parameter. One chooses the value of hyper-parameter depending on appropriateness of the value for the task in hand.

We will now implement a vanilla version of knn. Did you hear Vanilla? As in Chocolate, Strawberry, Vanilla. Yes, you did. You will see this term being used quite often in the world of machine learning. It is used to refer to the most basic form of anything. So when we say we will implement a vanilla version of k -nn, we mean the most basic version, most easy-peasy implementation of k -nn. Though you might find the code a bit complex, don't worry. Majority of it is a part of pre-processing and is a re-usable segment of code in any algorithm we'll implement in the future.

We have just touched up on what k -nn is at the surface level. There is a lot more to what k -nn is.

I would like you to run the code provided and make observation about what happens when we use:

1. even values for k
2. odd values for k

and then see Which one of it is preferable?

I would suggest you look into
what are the drawbacks of k -nearest neighbor
and
how Parameteric Learning ameliorates these drawbacks.

Go through the content provided to learn in detail about the performance measures used.

That brings us to the end of this course.

This course was intended to lay a foundation for what Machine Learning is.

If you liked the course and would love to learn more about Machine Learning and would love to implement more machine learning algorithms let us know about it.

Thanks for watching the video.