

As you will see week to week, we will have a coding video in Python about how to get started.

And each one of these models, With KNN we are actually doing the most intensive coding that we will do throughout the program.

So sit back. Make sure to pause the video anytime to essentially try to simulate the code.

We don't really provide this because it's really good practice to find to do the coding yourself instead of copying, pasting everything.

So the video will be relatively short, as they will be in each of the weeks.

These shouldn't really top five or six minutes. But again, really understand what we're trying to do here.

Also, a lot of what we're going to be using is in sklearn.

So we will also there's a lot of documentation around what you're doing.

So if you feel a little bit lost you are more than welcome to kind of look up those resources, a pause to see what we're talking about and making sure that you're getting off on the right foot as a benefit.

The data that we're gonna be using this example is what is your midweek assignment.

So this should help you get off again on the right foot and off to success for week one.

So the first dataset that we're going to be looking at here is around irises, which are the types of flowers.

And what this dataset allows us to do is to look at classifying correctly those flowers.

So a few common packages that you will see a lot are NumPy, which allows us to do a lot of array manipulation, sklearn as in this case right now, we're seeing we're importing datasets.

Eventually, we'll be using sklearn to download packages about a lot of our models that we're gonna be using in the following weeks.

So really what we're doing here to get started is really just separating the data into our training and training variables.

I mean, just getting our labels on the dataset comes in a little bit different, doesn't come in to clean data frame from sklearn.

So essentially, I'm just loading the data here, referencing it pretty much for the most part here, separating the data from the actual label.

As you can see here, the independent variables, which is the data and then the target is what we're predicting, which are the classification of flowers.

So as I move down here a little bit, I'm going to be kind of splitting up my dataset into train and test.

It's good practice here to do it, do it manual.

You'll see some code in the following weeks that there's actually a package within sklearn that does it.

But again, really this week one is for a lot of you acclimating yourself to Python and just getting used to it to the more practice, the better.

So just separate separating into training and test.

We always want to do that because this will allow us to kind of model it off of a large percent of data.

Our dataset. And then later on validate that it works just as well when we introduce random data to it.

Sit down here. Now, what we're going to do so this is a little bit more elaborate, but it creates this really great visual for us to see what nearest neighbors is doing.

You'll see similar examples to this and other videos this week.

But essentially here we're building a loop here through our training data set to gather the data points to classify the different flowers, which are represented by zero one and two.

So this loop here is running through and then eventually classifying them as red, green or yellow, so that we have all of our data points and we can plot them on this multi axis set down here.

We can visually start to see again all the points and where we may have some troubles classifying because we see some points kind of blend into each other.

You can see right around here there's a lot of yellow and green that

run into each other.

But whereas red does have some close overlap with green, but not so much.

So this may be a target of where the model has a hard time predicting because there appears to be a lot of similarities between the values of one and two or green and yellow as referenced above.

So in your earlier notes, before you read this video, we talk about how nearest neighbor does calculation.

The neighbor, the nearest neighbor of nearness, represents distance.

So we need to calculate the distance between two arrays, which we do here, which is very helpful to have the NumPy packages.

And then here we build our nearest neighbors model here where essentially we can define that.

We have a train set, we have a dependent variable, which is our labels. Our test data set, a value of K.

So how many voters? And then our distance. And then we have this loop that will go down here.

That will eventually calculate the distance between the points so that we can classify them correctly in our model.

Because again, this is what the actual points are or we want to do here is model out and predict having a so that it can predict accurately what those points would be, especially when giving a random data set to see where those points go in.

Lastly, here we have our vote going on that will look at the distance are the closest points to each point that's being voted on.

And we will have a vote of whether it is zero one or two.

And then this final loop here, essentially, it just gives us our results here across the board and allows us to then also see what data points come into there.

Essentially, this will tell us which data point went through, what was the vote, what was the label.

So the vote is what the models predict. The label is going to be what it actually is.

And then what data points it considered. So as we can see here, we're

going to be looking at twelve points.

Zero. Eleven. Twelve is twelve. So we can see here the result of the vote.

So what the model predicted was one. It actually was one.

So you see the model does a fairly good job here, getting it all correct.

But we do see here on line eight that the vote was one.

But the label is two. So that looks like here in the first 12 results that we're looking at is our only in accurate point here.

So that would again, make this eleven out of twelve, which is a pretty good percentage to get started with.

But we're going to want to understand why that is. Especially with one and two.

That means that was green and yellow, which is where we thought we may have some issues here, because we do see there's a lot of blending here and a potential overlap that could give the models some tricks.

So this would be a really good start. Again, this was just the limit of the data.

So there's plenty more to go through. But we will have these videos to really give you a nice foundation also in the videos that we will have.

And also on the links will give you plenty of resources to look at other ways of accomplishing this problem, especially if you run into anything.

So there'll be more videos this week to come by to learn and see what we will be covering about nearest neighbors and different ways to use them towards business value.

But I will see you in the next video.