in this lecture i'm going to discuss a form of supervised learning called nearest neighbor learning it's a very simple approach that you should be familiar with in nearest neighbor learning we have a training set just as we have for other supervised learning techniques giving an input and an output that is a label for that input for n different pairs we take that training set and we feed it into our learner but all our learner does is store that training set inside it inside the predictor and return a predictor that can take a new x and label it by simply pursuing this one idea choose the label y of the nearest training data point so we're going to look for xj in the training set whose feature vector is close to the feature vector of what we're labeling we'll take all the so we have to compute this distance between phi of x j for each of the xjs in the training set and find the one with the lowest distance that's the j that reaches the min here and that's the lit the j we use to to choose a label so we find that x in the training data that's as close to the x we're labeling and return its label okay this is an embodiment of the idea that similar inputs should generally get similar labels and here we're bodying similar as meaning this distance so what is this distance it's just the norm of the difference in the vectors that's the distance all right so this predictor just has to go compute that distance for each of the training data the distance from x find the closest one and return that y so it corresponds to you see an image you think of all the images you've seen before and think of the most similar one and say oh that must be one of those whatever that was labeled as to look at this visually we're going to imagine the feature space in which these feature vectors live for the low dimensional case where we might have two features and say we have five training points with these labels so the pluses are plus labeled training points in the feature space and the minuses are minus labeled feat examples in the features represented in the feature space so we're going to ask the question if a new data point was to come along say right down here what should we label it well this one's a little tricky because it's sort of equally close to these so it might be right on a decision boundary but if it was a little further down here we're going to label it plus and if it was a little further up here we're going to label it minus in a nearest neighbor system so what are the decision boundaries in the nearest neighbor system they fall sort of between these points the boundaries too so for instance between these two points you could draw the segment and make a perpendicular bisector and that line would be a piece of the decision boundaries for this predictor it would only stop being a decision boundary when it got close to a different data point besides these two and so i've drawn i've done that for this diagram and you can see that the decision boundaries that emerge are pretty complicated looking just for five data points so i have these perpendicular bisectors of the segments and where they come together they denote regions that are all going to be labeled the same so any x in this region is going to be labeled minus and any x in this region will be labeled plus well this kind of diagram that divides up a space into regions by picking sort of centers and taking everything that's closest to a given center as its region has a name it's called a voronoi diagram no doubt after a guy named voronoi and so i've written out what i just said it's a partition of the space into regions where the regions are each defined by a center so we have a set of region centers and the center a given center contributes its region of all the points that are closest to it we're um also uh going to look at a clustering algorithm that learns of voronoi diagram so when you think about k-means you should also think about that it is learning exactly this sort of model of the input the last thing i want to talk about is just what's good and what's bad about this technique so one thing we can notice is a pretty complicated learned predictor here not some so you know our linear techniques just learn a line here this is not just a line so definitely one of the advantages is we can represent a very complicated predictor and we do it instantly once we've read the training data we're done learning so it's a very fast learner and represents a complex predictor the problem should be fairly obvious is that this argument has to be computed for each prediction we have to compute a distance to x to phi of x for each point in the data set so this ranges over the entire training data that can be very large training data so the the principal disadvantage is that prediction is very slow and there are things you can do to remedy that um certainly there like all the techniques we're studying we're just studying the basic core of a whole literature about that technique and you can imagine doing things like removing a redundant data points if i had another plus up here it wouldn't be changing the decision boundaries between plus and minus i only really care about the boundaries between plus and minus so i could do some things to pre-process and i can speed up this computation in various ways but it is still going to be a slow predictor and to my mind as someone who studies more the logical and linguistic end of ai it's disturbing to me that it doesn't there's no sense in which it's finding or learning a pattern implicitly it is but there's nothing that could possibly be explicated um whereas the linear learner seems to to be simplifying the data by finding some pattern here the learned hypothesis is just as complex as the data so all that said it's definitely a very simple and powerful technique that you should be familiar with and it has its place