Everybody welcome. Today's topic is Discriminant Analysis.

I will as usual go through this demonstration,

talk through each of these functions and then we'll talk about what the results mean.

As always, I recommend that you follow

along and run the code and maybe try some variations.

Now, the additional R package we'll be needing for today is MASS,

M-A-S-S all caps for the packages LDA,

which as you might expect stands for linear discriminant analysis and QDA for quadratic.

Now, the other packages are pretty usual,

although we will also use the package HE plots for the function boxM,

which is used to test equality of variances between groups.

This is a very famous dataset, almost trite.

Basically, there are four measurements on 150 flowers

from three subspecies of Iris: setosa,

versicolor, and virginica and we'll use it

to illustrate linear and quadratic discriminant analysis.

This dataset turns out to be included with R,

so let's load it up.

Here's what it looks like.

We have five variables in a dataset.

Three quantitative continuous,

sepal length, sepal width, petal length,

petal width, and one categorical species,

which is setosa,

virginica, and versicolor.

Now, let's make a plot.

I'm going to use ggpairs as usual.

One thing I'm going to do in addition to that is use AES, colour equals species.

This basically means colour-code by species.

Then make sure that you get a different colour for each species in all of the plots,

and also use Alpha 0.3,

which if for some cases, for example,

when you have these parallel density or over plot or density plots,

we can actually have three separate colours.

You can compare these species.

In this case, setosa is red,

versicolor is green, virginica is blue.

As we can see down here, although,

if we wanted to, we could also add a legend on the side.

Now, we can see there's equal number of each species,

this is because that's how the data were collected.

Then we can also look at the sample distribution

of some of these variables and how they separate some of these groups.

In particular, it seems just we're eyeballing

this petal length and petal width

might separate three species pretty well, particularly put together.

Anyway, how do we fit this?

Well, let's begin by assuming

the three species have equal variances and covariances among the variables,

and we can fit the linear discriminant analysis.

My question is, is that a good assumption?

Well, I leave it to you,

looking at these plots,

whether you think that these groups have the same group variance.

Here's what the output looks like.

Well, here's what the code looks like.

LDA and then the species as a function of all the other columns in the DataFrame,

we could have written them out one by one, we don't have to.

Then data equals Iris.

That's the usual interface used in R,

but it will give us prior probabilities of the groups.

These are just proportions of data.

Group means: setosa, versicolor,

virginica and then coefficients of linear discriminant.

Now, the output here is a bit different from the one used in the lecture.

This is because it's optimised for rapidly classifying large numbers of observations.

There is a mapping there.

You can think of it in terms of as you're trying to classify it into three groups,

and you have predicted probabilities for two of the groups.

You know the third one, so really you only need two discriminants,

which is why there are only two of them.

Then the third one is the default.

Now, I'm going to try to give you some intuition here,

which we do our best plotting in two dimensions.

We will focus on two variables,

sepal length and sepal width.

Now, these are not the best variables for this.

We can see that here.

The best ones are actually petal length and petal width.

But we're going to use these two to get some idea of

how this works and how the algorithm performance when challenged.

Here we have our fit,

we have our means,

and we have our discriminants.

We have three categories,

so we still have two discriminants,

but we only have two variables,

so only two group meets.

Now, the way I'm going to illustrate this is I'm going

to generate a grid of values that cover the data.

We've done this before when dealing with

confidence intervals or confidence regions, confidence ellipsoids.

Here, we have from

the minimal sepal width to the maximum sepal width a sequence of numbers.

Length, sepal length,

that's 200 long,

and the same thing with the width.

From the minimum width to the maximum width, 200 points.

They use expand grid to get every combination,

there should be 40,000 of them.

Then we'll make the predictions and we'll use predict.

Then we pass the fitted LDA, we put it up here.

We pass the new data,

which is the sepal length and sepal width combinations,

and then we extract the element class.

We extract element class and we're going to convert it to

a factor because that makes it colour-code.

That basically means it will be automatically colour-coded.

Now we're going to plot each of the two coordinates of x,

y, the sepal length and sepal width here.

We're going to colour them according to the predicted value.

Then we're going to label the axes appropriately.

We're going to use just a small dot as our plotting symbol.

Then finally, what we're going to do is we're going to plot the original data sets points,

and we're going to again colour-code them according to the species and we're going to

use a distinct character for it,

like a filled square.

This makes it look a bit better.

This is what these look like.

You can see that the boundaries for the predictions are linear.

There's this point where they all meet,

and then you have lines going out from that.

In each region, its own species predominates,

but you have to figure out where to split the greens from the reds,

and that's where you do it.

But now what we assume here is that the variances are equal,

so we need to check that.

Now we know how to do that.

We learned to do that last week.

We grab the iris dataset except for the species,

that's what the select function in deep layer does.

Then we run boxM,

that's the test for iris dollar sign species, so record it.

Basically that's equivalent to basically saying boxM,

iris without the species as the first argument and iris species as a second argument.

It'll give you the M-test for homogeneity of covariance matrices,

and this is what the result looks like.

Now, obviously this is a very small p-value,

so we confirmed that they are in fact very different,

and we could see that from the plots.

Now, one thing to keep in mind is that this does not mean that we can't use the LDA.

It may actually do a perfectly good job of separating the groups.

In fact, if you, for example,

if we had petal length and petal width here.

Sorry, I'm going to zoom in here on this frame a bit.

If we drew a separating line here and a separating line here,

we do a really good job.

In fact, it could be that LDA is perfectly good.

Now, the predictions of probabilities won't

be valid because they assume a specific model,

but in terms of predictions is fine.

Now, coming back to this question of what to do,

we can keep the LDA,

might be good enough,

but we can also consider quadratic discriminant analysis.

Let's begin by looking at the classification regions.

We use QDA to fit the model to the data, same as before.

Then we go through the same code as before.

Now this is what it looks like.

Now the boundaries are actually curved.

As you can see that it tries to get things a bit more vivid, more thorough.

Now one slightly counter-intuitive result is that if we zoom out,

so this is zooming out going from 2-10 and 0-6,

which is way outside the range of the data,

we in fact get these weird regions.

They are, in fact, not even contiguous.

This is because QDA allows for much more complicated structures.

But of course there's the usual lesson

that don't extrapolate beyond the range of the data.

Next, we can fit using all the predictors.

Let's do that. Here it is.

Now, one last thing I want to talk about in this section is comparing classifiers.

Now the simplest way to do that is something called a confusion matrix.

It's pretty straightforward. We just

cross-tabulate observed classes against the predicted.

Now turns out that R has a function called table,

and here it is.

You just give it the variable 1 and variable 2 and it'll cross-classify them.

You can give them names, it's completely optional.

I chose to call this truth and prediction.

I could have called them anything else or nothing at all.

We can see that the prediction is pretty good.

In fact, LDA and QDA for all four,

give pretty much the same result,

whereas QDA and LDA for sepal length and width only,

they actually mix up versicolor and virginica a bit.

Setosa is well-separated in any case.

In this case a two classifiers performed by equally well.

Now, this approach has

a disadvantage that the same data are used to train the classifier as to test it.

Now, this is probably something you'll learn more about in data mining.

But in some sense that's cheating because we can always fit

a model that fits any given pattern of points perfectly,

just by adding more and more parameters.

There're all sorts of technical terms for it: bias-variance, trade-off, overfitting.

But in this case,

what we're just going to do is we're going to talk a little bit

about how to actually account for that.

The approach we use is cross-validation.

The idea is that we split the data set up into a bunch

of groups and in fact it actually be a group of one,

so just split data set into data points.

Then each of the subsets of the data set takes a turn being predicted by all the others.

That we were not using the same data to predict as we do to evaluate,

and be clear predict using the model fitted based on the others.

Anyway, it turns out that this is actually built in to LDA.

In fact, we just take CV equals true,

and same thing with QDA.

Now we actually see that LDA performs a little bit better,

and because there is a slightly smaller error,

only slightly, could very well be a fluke.

But does go to the general point that

often simpler models perform

better once you take into account overfitting and cross-validation.

Same thing here, LDA is slightly better.

This concludes the introduction to classification and discriminant analysis demo.

As always, I encourage you to go through

this code and play around with them and look up for help.