Applied Predictive Modeling

Lab 1, Part V Transcript: Principal Component Analysis in R

Now let's move on to principal component analysis. As I said before, this is one of the most useful techniques you'll learn in Module 1. Principal component analysis is commonly used in machine learning algorithms to reduce the dimensionality of the data. Basically, what it means is if you have a data with a large number of features, and you want to kind of pick a subset of features that have the most amount of information, principal component analysis is one of the methods that achieves that purpose. And principal component analysis usually finds linear relationships within the data. So if there are non-linear relationships within the data, those will not be found through principal component analysis. And the beauty about the principal component analysis is it gives you what are called the principal components in an order of how much variance each component kind of captures in the data.

So, as we go from principal component 1, 2, 3, you kind of see the reduction in the amount of variance that has been captured by each component. And all of these components are uncorrelated with each other, which is a very good property for a lot of machine learning algorithms or linear models that would require you don't, the feature should be uncorrelated. So the principal component analysis achieves both of those things, but the only drawback is once you do a principal component analysis and you get a new feature set, that feature set is a linear combination of all the existing features.

You cannot explain the outputs as easily as you are able to explain the previous outputs with the original data. So that's the only drawback. Interpretability is a big thing, you got to remember that this is gonna be difficult, with the principal component analysis. So let's go ahead and look at a few examples on how principal component analysis works. So here I'm gonna create, some random numbers.

I'm starting with a seed of 100, and then we have seen this before, x, y, z, are three vectors, numeric vectors that have 1000 observations. And I'm gonna create a matrix of 1000 rows and 3 columns as zoom x, y, z are my features that we are interested in, and this becomes our data set. So principal component analysis takes matrix as its input. So this prcomp function is available in the base R and if you pass a matrix, it gives you a list. So you can see, let's run it. So m_pca is a list, as you can see here. And if you look at the structure of m_pca, you see there is the standard

deviation, there is a rotation, there is a centering, there is a scaling, and then there is an x, which is the transformed coordinate dataset, basically.

So standard deviation is how much of the variance is explained, or the standard deviation is explained by each of the principal components rotation. These are the principal components themselves, principal component vectors. And the center, it is by default, these values are centered in principal component analysis. So you can check more details about principal component analysis by looking at the options for principal component analysis. The centering is true by default, if you don't want to do the centering, it's already done, then you can set it to false, but these were the way values used to center those vectors.

That's what it means because we have three features. These are the three values that were used to center those features. And then the scaling is another important. We'll go about the scaling little bit more in detail in a few minutes, but for now, if I do str, pca, the most important things to look at are the standard deviation and the rotation, which gives you the principal components and the variance captured by the principal components.

So if you want to come look at the variance, you just take the square of the standard deviation that's captured by these principal components. Here I created a new vector m_pca variance, which is a square of this. And then you can see right here, or I can print it out here to see, what it is.

You can see, this is the variance explained by each of the principal components. So that's one way to see how much of the variance each principal component is explaining. And if you want to look at the matrix, you can do m_pca$rotation, these are the principal components. So the way to read these principal components is if you look at, we know these are the three features that we have provided.

You could think of these are called sometimes loadings and the weights assigned to each of the original features. So you could think of this as principal component 1 is a linear combination of the original features using these weights. Basically, these are weights that are used to compute the new features. Similarly, PC2 and PC3, and by looking at these weights, sometimes you may be able to figure out which feature each principal component is focusing on.

It could be one single feature or multiple features that each principal component could focus on. And then if you want to, yeah, look at pca, m_pca$x. So this is, as I told you, this is the transformed data set. So we had a 1000 X 3 matrix that we passed after the principal component

analysis, you'll get another 1000 X 3 matrix if you use all principal components.

Another thing, the number of principal components will be equal to the number of features that you have provided. The only difference will be the amount of variance captured by each of the principal components. So here you can see, this will be another 1000 X 3 matrix, and this will be the new features with which you'll be doing your machine learning modeling and other modeling tasks.

Let's go to example 2 for now. So here I wanted to show you, I created a new feature called t, which is you can think of, it's like, it's x+y+z/ 3. So I'll create another matrix m1 which is using x, y, z and t. So this now will have 4 features, 1000 rows. So let's create this first. And then let's do the principal component analysis and see what happens.

As I mentioned before, if principal component analysis is very good at identifying linear dependencies between features. So here, because t is a linear function of x, y, and z, let's see how the variance capture by each principal component looks.

If we look at the variance captured by each principal component, let me do m1, underscore pca, underscore variance. So you can clearly see that the last principal component almost contributes nothing, it's like e to the power of -31. So what the principal component analysis was able to identify is that there are only three real features in this dataset. The fourth feature can be inferred from the first three, that's why it doesn't have any importance in the terms. It doesn't explain any additional variance in the data. Biplot is another useful function that can be used to plot principal component analysis results.

Basically, here I'm doing biplot , which is our principal component object. When you do that, what you see is a plot of points with principal component directions, basically. So within the transformed coordinates, how those different points are with respect to PC1 and PC2. So, and it will give you the variables and the loadings. So if you see there are two scales. So the principal component 1 has this scale, which is the transformed coordinates, and this is the weight of the rotation matrix, components used to weigh each of these variables. Principal component 2 has two scales, one scale here and another scale here where this is the scale where which the new points fall on principal component 2, and then these are the values for the loading. So the way to read this is if you look at m1_pca$rotation matrix, let's look at PC1. So PC1, you can see variable 1 has a loading of 0.62. So if I look at

PC1, which is this and this scale, you can see variable 1 is somewhere here, which is close to that 0.62-ish.

And then if you look at variable 2, variable 2 is somewhere here, which is somewhere around 0.3-ish, and then variable three is around 0.5. So if you see variable 3 is pretty close to 0.5, and then you have variable 4, again, which is close to 0.5. So this is how you read it. And if you go to principal component 2, which are these two scales for principal component two, you can clearly see only variable 3 has a positive component, and then variable 1, 2, and 4 are either negative or close to zero. So here you can see the same thing in the rotation metrics, variable 2 as close to 0.8 or variable 3 as close to 0.8. So this is what this value is, which is close to probably 0.8, if you look at it.

That's how you read these biplots. So this is one way to look at how the data points are scattered along these new principal component analysis, because we picked random data. You don't see much trend, but if there is trends in data, you may see clusters of data forming, where one of the clusters will be maybe somewhere here showing that principal component 1 high means you're capturing this cluster and other cluster may form somewhere here where it means that the principal component 2 is capturing this cluster when these values are high.

Okay, let's look at another example. So as I told you, principal component analysis captures linear transformations very well, but how about if the transformations are nonlinear? So here I created a new variable t with nonlinear transformation, e power x times y plus z. And I created another matrix, m2, similar to what we did in the previous example. And now let let's look at how the principal component analysis and the variance capture looks like. Here, you can clearly see none of the principal components are equal to zero.

Previously, we had m1, the last component was zero here, even though the final variable is a function of everything else, but since it's a nonlinear transformation, pca won't be able to identify it. So you'll see that it all contributes to me of the variance in the data here. And then you can do the same thing by looking at the biplot. And as I mentioned about this choices function in the biplot, if you don't want to plot principal component 1 and 2, but you want to plot principal component 2 and 3 this is how you would do it.

Choices 2 to 3 would plot the principal component 2 and 3 in this case and how the available loadings and the data are distributed. Okay, , and one

final example I want to go, I want to show you guys before going is, let's create another example where we have three normal, random normal variables. And then these are, when you create a random normal variable like this, it has means zero and variance one, and this random normal variable, let's create with higher variance. It has a standard deviation of four, variance of 16.

So when we create something like this, and then we pass the data, and then we do the principal component analysis. What you'll observe is when you look at m3_pca and the variance contributed by each component, one of the components will dominate big time with respect to standard deviation, because everything else has a standard deviation of one. This one has a standard deviation of four. So that's how, that's what you kind of see in the standard deviation captured. 4, 1, 1, close to 1, 1 kind of a thing. So that's why it is very important to scale the variables before you do principal component analysis, if not your principal component will point to the direction that has maximum variance.

To do the, to do remove, to remove this kind of behavior, we do the center and scaling that the principal components clearly capture the variation after adjusting for the scale and the standard deviation within their components. If not, higher variance components will show up as the most important principal components. And then like we have seen before, we can do the, what is the PCA standard deviation and what is the variance? Here, you can see the first component contributes 84% of the variation, because it's just the fact that it has such higher variance when compared to the other theory.

It seems like it's a very important thing, it's just the matter of fact that it has more variance associated with it. So it just illustrates the fact that you have to be careful when you're doing principal component analysis. And if somebody asks you, "Hey, how many components do you need to look at to capture 90% of the variance?" You can see here when we did this m3_pca_var times 100 divided by the sum, we are dividing by the sum of all the variance components and then dividing it here and then multiplying it by 100. So we get kind of variance contribution for each PC, each principal component. Here, we can see that the first two components contribute 90% or more of the variation.