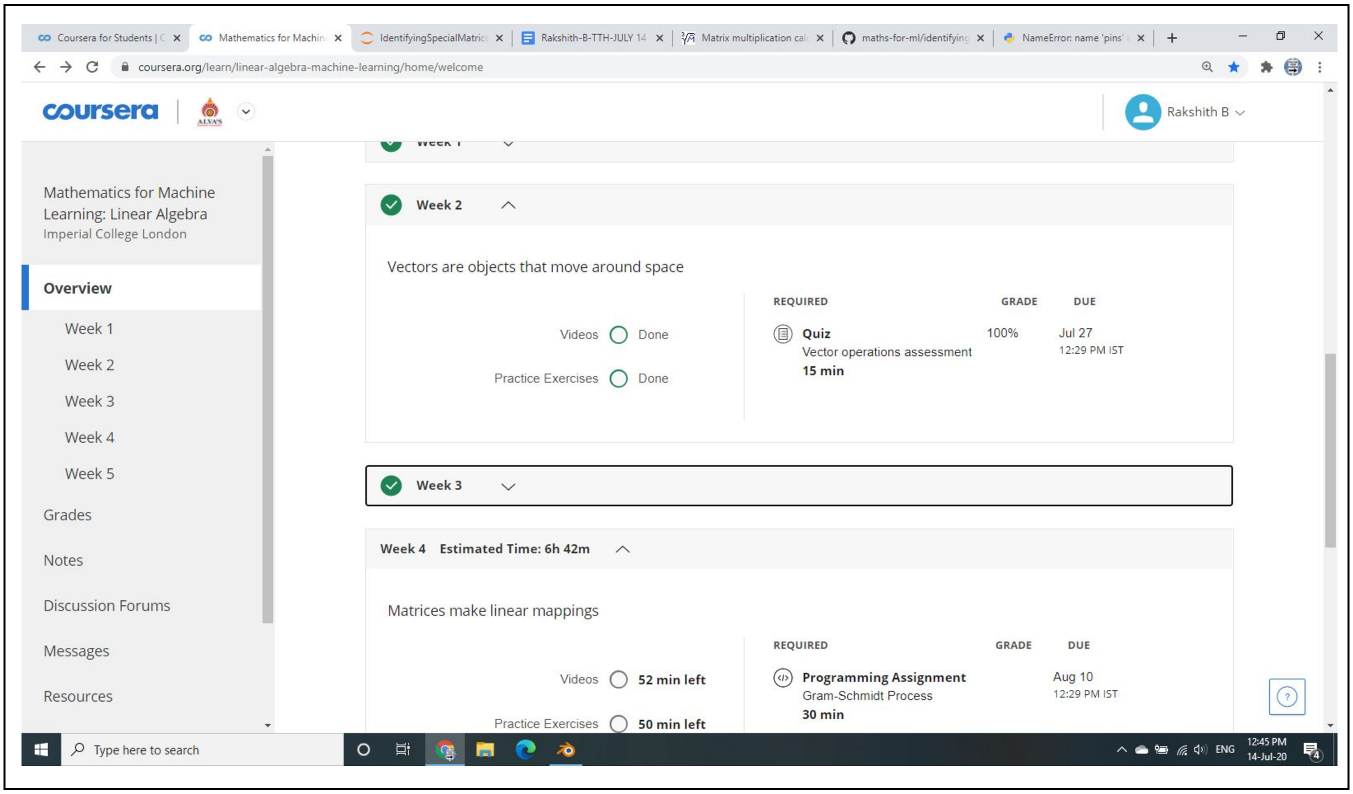
REPORT JULY 14

Date:	14 JULY 2020	Name:	Rakshith B
Course:	Coursera	USN:	4AL16EC409
Topic:	Mathematics for Machine Learning	Semester & Section:	6th SEM B
Github Repository:	Rakshith-B		

Image of the Session



So we've looked at the two main vector operations of addition and scaling by a number. Those are all the things we really need to be able to do to define what we mean by a vector, the mathematical properties that a vector has. Now, we can move on to define two things; the length of a vector, also called its size, and the dot product of a vector, also called it's inner scalar or projection product. The dot product is this huge and amazing concept in linear algebra, with huge numbers of implications. I will only be able to touch on a few parts here, but enjoy. It's one of the most beautiful parts of linear algebra. So when we define a vector, initially, say this guy r here, we did it without reference to any coordinate system. In fact, the geometric object, this thing r, just has two properties, its length and its direction that it's pointing that way. So irrespective of the coordinate system we decided to use, we want to know how to calculate these two properties of length and direction. If the coordinate system was constructed out of two unit vectors that are orthogonal to each other, like i here and j here in 2D, then we can say that r is equal to a times i, plus b times j. When I say unit about i and j, I mean that of length one, which people will often denote by putting a little hat over them like this. Then from Pythagoras, we can say that the length of r is given by the hypotenuse.

So what I mean by that is, if we draw a little triangle here, then we've got this length here is ai. So if we write the length being, with these two little vertical lines, it's just of length a, because it is of length one. This side here is bj, and that's of length b. So this side here is from Pythagoras, is just a square plus b squared, all square rooted, and that's the size of r. So we can write down r, quite often people will do this, write r down like this, just ignoring the i and j and writing it as a column vector. So r is equal to a-b. The size of r, we write down as being the square root of a squared plus b squared. Now, we've done this for two spatial directions defined by unit vectors i and j that are at right angles to each other. But this definition of the size of a vector is more general than that. It doesn't matter if the different components of the vector or dimensions in space like here, or even things have different fiscal units like length, and time, and price. We still define the size of a vector through the sums of the squares of its components. The next thing we're going to do is to find the dot product. One way among several, multiplying if you'd like two vectors together. If we have two vectors, r and s here, r here has components r_i, r_j, so r in the i direction, r in the j direction, and s has components s_i and s_j, then we define r dotted with s to be given by multiplying the i components together. So that's r_i times s_i, and adding the j components together, so that's r_j times s_j.

The dot product is just a number, a scalar number, about three, given by multiplying the components of the vector together in turn, and adding those up. So in this case, that would be three and two, for the rij, and minus one, and two for s. So if we do that, then we get a sum, the r.s is equal to minus three plus four, which gives us one. So r.s in this case, it's just one. Now, we need to prove some properties of the dot product. First, it's commutative.

What commutative means is that r.s is equal to s.r. It doesn't matter which way around we do it. It doesn't matter because when we put these numbers in here, if we interchange those, the rs and Ss, we get the same thing when we multiply minus one by three, it's the same as three times minus one. So, it doesn't matter which way round we do the dot product. s.r is equal to r.s, which means it's commutative. Second property we want to prove the dot product is distributive over addition. By which I mean that if I've got a third vector here now t, that r dotted with s plus t is equal to r dotted with s, plus r dotted with t.

I can multiply it out in that way. It probably feels mundane or obvious, but let's prove it in the general case. So let's say I've got some n-dimensional vector r, components r_1, r_2, all the way up to r_n, and s, is the same as components s_1, s_2, all the way up to s_n, and t has components t_1, t_2, all the way up to t_n. Then let's multiply it out. So if we take the left-hand side, r dotted with s plus t, that's going to be equal to r_1 times s_1 plus t_1.

We take the components. Then r_2, component r_2, times components s_2 plus t_2. Then all the dimensions in between, and then finally, r-n times s_n plus t_n. Then what we can do, is we can then sort that out. So we've got to multiply that out. So we've got r_1, s_1, plus r_1, t_1, plus r_2, s_2, plus r_2, t_2, plus all the ones in between r_n and s_n, plus r_n, t_n. Then we can collect it together. So we've got the r_1, s_1 times r_2, s_2, all the way to r_n, s-n. That's of course, just equal to r dotted with s. If we collect the r_t terms together, we've got r_1, t with t_1, r_2, t_2, all the ones in between r_n, t_n.

That's just r dotted with t. So we've demonstrated that this is in fact true, that you can pull out plus signs and dots in this way, which is called being distributed over addition. The third thing we're going to look at is what's called associativity. So that is, if we take a vector, a dot product, and we've got r dotted with some multiple of s, where a is just a number, it's just a scalar number. So we're multiplying s by a scalar. What we're going to say is that that is equal to a times r dotted with s. That means that it's associative over scalar multiplication. We can prove that quite easily, just in the 2D case. So if we say we've got r_1 times a s_1 plus r_2 times a s_2, that's the left-hand side, just for a two-dimensional vector. Then we can pull the a out. So we can take the a out of both of these, and then we've got r_1, s_1, plus r_2, s_2. That's just r.s, a time r.s. So this is in fact true. So we've got our three properties that the dot product is commutative. We can interchange it. Is distributed over addition, which is this expression, and its associative over scalar multiplication. We can just pull scalar numbers out. As an aside, sometimes you'll see people in physics and engineering write vectors in bold, numbers or scalars in normal font or they'll underline their vectors to easily distinguish them from things that have scalars. Whereas in math and computer science, people don't tend to do that. It's just the notation difference between different communities, and it's not anything fundamental to worry about. The last thing we need to do before we can move on is, draw out a link between the dot product and the length or modulus of a vector. If I take a vector and dot it with itself, so r dotted with r, what I get is just the sums of the squares of its components. So I get r_1 times r_1, plus r_2 times r_2, and all the others if there were all the others. So I get r-1 squared plus r_2 squared. Now that's quite interesting because that means if I take the dot product of a vector with itself, I get the square of its size of its modulus. So that equals r_1 plus r_2 squared, square rooted, all squared. So that's mod r squared. So if we want to get the size of a vector, we can do that just by dotting the vector with itself and taking the square root. That's really neat and really hopefully, quite satisfying.

Now, there's one last thing to talk about in the segment which is called projection.

Projection. And for that, we'll need to draw a triangle. So if I've got a vector R and another vector S. Now, if I take a little right-handed triangle, drop a little right-handed triangle down here where this angle's 90 degrees, then I can do the following. If I can say that if this angle here is theta, but cos theta is equal to, from sohcahtoa, is equal to the adjacent length here over the hypotenuse, that is, and this hypotenuse is the size of S. Now, if I compare that to the definition of the dot product, I can say that R dotted with, we'll have fun with colors, dotted with S is equal to mode R size of R, times the size of S, times cos theta. But the size of S times cos theta if i put S up here, just need to put my theta in there, cos S, cos theta is just the adjacent side, so that's just the adjacent site here in the triangle. So, the adjacent side here is just kind of the shadow, if I had a light coming down from here, it's the shadow of S on R.

That length there, it's kind of a shadow cast. If I had a light at 90 degrees to R shining down on S, and that's called the projection. So what that dot product gives us, is it gives us the projection here of S on to R times the size of R. And one thing to notice here is that if S was perpendicular to R, if S was pointing this way, it would have no shadow. That is if cos theta was 90 degrees that shadow would be no, the cos theta would be no here and I get no projection. So, the other thing the dot product gives us, is it gives us the size of our times some idea about the projection of S onto R. The shadow of S onto R. So, if I divide the dot product R.S by the length of R, just bring the R down here, I get mode S cos theta. I get that adjacent side, I get a number which is called, cause R.S is a number and the size of R is a number, and that's called the scalar projection.

And that's why the dot product is also called the projection product, because it takes the projection of one vector onto another. We just have to divide by the length of R, and if R happened to be a unit vector or one of the vectors we used to find the space of length one, then that would be of length one and our dot S would just be the scalar projection of S onto that all that vector defining the axes or whatever it was. Now, if I want to remember to encode something about R, which way R was going into the dot product or into the project book product could define something called the vector projection. And that's defined to be R.S over mode R dotted with itself.

So R.R mode R squared, that's R.S over R.R if you like because mode R squared is equal to R.R. And we multiply that by the vector R itself. So that is that's dot products just a number, these sizes are just a number, and R itself is a vector. So what we've done here is we've taken the scalar projection R.S over R, this guy that's how much S goes along R, and we've multiplied it by R divided by its length. So we've multiplied it by a vector going the direction of R but it's been normalized to have a length one. So that vector projection is a number, times a unit vector that goes the direction of R. So if R say was was some number of lengths, the vector that will be R divided by its size, say if that was a unit length vector I've just drawn there, and the vector projection would be that number S.R, that adjacent side, times a vector going in the unit length of R. So that's, if you like the scalar projection also encoded with something about the direction of R, just a unit vector going in the direction of R. So we've defined a scalar projection here, and we've defined a vector projection there. So good job. This was really the cool video for this week, we've done some real work here. We found the size of a vector and we defined the dot projection product. We've then found out some mathematical operations we can do with the dot product. This distributes over vector addition and is associative with scalar multiplication and is commutative. We then found that it finds the angle between two vectors, the extent to which they go in the same direction, or then it finds the projection of one vector onto another. That's kind of how one vector will collapse onto another, which is what we'll explore in the next two videos. So, good work now's a good time to pause, and try some examples, but put all this together and give it all a workout on a bit of a try before we move on.

Summary :

Let's just take a moment to think about what we've done in this module because you've worked quite hard. And if all this is completely new to you, you've had to think about a lot of new ideas. We've looked at vectors as being objects that describe where we are in space which could be a physical space, a space of data, or a parameter space of the parameters of a function. It doesn't really matter. It's just some space. Then we've defined vector addition and scaling a vector by a number, making it bigger or reversing its direction. Then we've gone on to find the magnitude or modulus of a vector, and the dot scalar and vector projection product. We've defined the basis of a vector space, its dimension, and the ideas of linear independence and linear combinations. We've used projections to look at one case of changes from one basis to another, for the case where the new basis is orthogonal. So we've done a lot of stuff with vectors and along the way, we've done a bit of thinking about how this will apply to working with data. So hopefully, that's been really useful. And you've enjoyed giving it all a workout in the exercises and activities. And I'll see you in the next modules, where we'll move on to think about the related idea of matrices.

Date	14 JULY 2020	Name:RAKSHITH B
Course	Salesforce	USN:4AL16EC409
Topic	Administration	Semester & Section:6 B

Install Apps and Packages in Your Trailhead Playground

Learning Objectives

After completing this unit, you'll be able to:

- Install an app or package in your Trailhead Playground.

What's an App?

You're probably comfortable with the idea of app stores. Whether you're downloading apps on your phone, tablet, computer, or other device, you have to download and install apps to make the most of your technology. Salesforce is the same way.

Salesforce has a community of partners that use the flexibility of the Salesforce platform to build amazing apps that anyone can use. These apps are available for installation on AppExchange (some for free, some at a cost).

What's a Package?

A package is a set of pre-created configurations and developments. At various points in your Trailhead learning journey, you may need to install a package in order to complete a challenge or work through the steps in a badge.

Packages allow us to load sample data, custom objects and fields, or just about anything else into your Trailhead Playground.

Install an App or Package in Your Trailhead Playground

To install an app or package in your Trailhead Playground, you just need the package ID provided in the Trailhead content. This is a long string that starts with `04t`—you'll know it when you see it.

First things first: Launch your Trailhead Playground by going to any hands-on challenge, scrolling to the bottom of the page, and clicking **Launch**. If you see a tab in your org labeled Install a Package, fantastic! Follow the steps in the Your Playground Has the Playground Starter App section below.

If not, click to launch the App Launcher, then click **Playground Starter** and keep reading. If you don't see the Playground Starter app, go ahead and skip to the Your Playground Doesn't Have the Playground Starter App section.

Your Playground Has the Playground Starter App

If your playground has the Playground Starter app, follow these steps to install an app or package.

1. Find the package ID starting with `04t` and copy it to your clipboard.
2. Click the **Install a Package** tab.
3. Paste the package ID into the field.
4. Click **Install**.

1. Select **Install for Admins Only**, then click **Install**.

1. If you see a pop-up asking you to approve third-party access, select **Yes, grant access to these third-party websites** and click **Continue**.

When your package or app is finished installing, you see a confirmation page and get an email to the address associated with your playground.

Your Playground Doesn't Have the Playground Starter App

If your playground doesn't have the Playground Starter app, never fear. Follow these steps to install a package. To do this, you'll need your username and password for your Trailhead Playground, as well as the package installation link. If you don't have your username and password on hand, go to the previous unit and get them.

Next, take a look at that installation link. Does it begin with *appexchange.salesforce.com*? If so, you're installing an app, not a package, and you can skip to the next section, Install an AppExchange App. Otherwise, you're installing a package, and you can keep reading.

1. Open a new private browsing window. In Chrome, click **File | New Incognito Window**. In Safari, click **File | New Private Window**. This ensures that you install the package in your playground, and not any other org you have open. We wouldn't want you to accidentally install a package or app in your production org.

2. Copy the package installation link and paste it into your private browsing window.
3. You'll be prompted to log in. Enter the username and password of your Trailhead Playground and click **Log In**.
4. Select **Install for Admins Only**, then click **Install**.
5. If you see a pop-up asking you to approve third-party access, select **Yes, grant access to these third-party websites** and click **Continue**.

When your package or app is finished installing, you'll see a confirmation page and get an email to the address associated with your playground.

Install an AppExchange App

1. Open a new private browsing window. In Chrome, click **File | New Incognito Window**. In Safari, click **File | New Private Window**.
2. Copy the AppExchange link and paste it into your private browsing window.
3. Click **Get It Now**.

1. Click **Log In**.

1. Click **Salesforce**.

1. You'll be prompted to log in. Enter the username and password of your Trailhead Playground and click **Log In**.
2. Click **Install in Production**. Your Trailhead Playground is a production instance for your personal use. You can make customizations in your playground without impacting anything else (in this case, your production org).
3. Select **I have read and agree to the terms and conditions**.
4. Click **Confirm** and **Install**.
5. If you're prompted to log in again, enter your Trailhead Playground username and password and click **Log In**.
6. Select **Install for Admins Only**, then click **Install**.
7. If you see a pop-up asking you to approve third-party access, select **Yes, grant access to these third-party websites** and click **Continue**.
8. Once the installation is complete, click **Done**.

See What Was Installed

Want to see what was included in the app or package you installed?

1. In your Trailhead Playground, click and select **Setup**.
2. From Setup, enter `Installed Packages` in the Quick Find box and select **Installed Packages**.
3. Click the app or package from the list.
4. Click **View Components**.

On this page, you'll find all of the components you installed.