DAILY ASSESSMENT FORMAT

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **13/07/2020** | **Name:** | **Nayanashree k s** |
| **Course:** | **Coursera** | **USN:** | **4AL16EC042** |
| **Topic:** | **Mathematics of machine learning-Linear algebra** | **Semester & Section:** | **8 A** |
| **Github Repository:** | **Nayana\_online** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| **Report**  When formalizing intuitive concepts, a common approach is to construct a set of objects (symbols) and a set of rules to manipulate these objects. This is known as an algebra. Linear algebra is the study of vectors and certain algebra rules to manipulate vectors. The vectors many of us know from school are called “geometric vectors”, which are usually denoted by a small arrow above the letter, e.g., −→x and −→y . In this book, we discuss more general concepts of vectors and use a bold letter to represent them, e.g., x and y. In general, vectors are special objects that can be added together and multiplied by scalars to produce another object of the same kind. From an abstract mathematical viewpoint, any object that satisfies these two properties can be considered a vector. Here are some examples of such vector objects: 1. Geometric vectors. This example of a vector may be familiar from high school mathematics and physics. Geometric vectors – see Figure 2.1(a) – are directed segments, which can be drawn (at least in two dimensions).  Two geometric vectors → x, → y can be added, such that → x+ → y = → z is another geometric vector. Furthermore, multiplication by a scalar λ → x, λ ∈ R, is also a geometric vector. In fact, it is the original vector scaled by λ. Therefore, geometric vectors are instances of the vector concepts introduced previously. Interpreting vectors as geometric vectors enables us to use our intuitions about direction and magnitude to reason about mathematical operations. 2. Polynomials are also vectors; see Figure 2.1(b): Two polynomials can be added together, which results in another polynomial; and they can be multiplied by a scalar λ ∈ R, and the result is a polynomial as well. Therefore, polynomials are (rather unusual) instances of vectors.  Note that polynomials are very different from geometric vectors. While geometric vectors are concrete “drawings”, polynomials are abstract concepts. However, they are both vectors in the sense previously described. 3. Audio signals are vectors. Audio signals are represented as a series of numbers. We can add audio signals together, and their sum is a new audio signal. If we scale an audio signal, we also obtain an audio signal. Therefore, audio signals are a type of vector, too. 4. Elements of Rn (tuples of n real numbers) are vectors. Rn is more abstract than polynomials, and it is the concept we focus on in this book. For instance, a =   1 2 3   ∈ R 3 (2.1) is an example of a triplet of numbers. Adding two vectors a, b ∈ Rn component-wise results in another vector: a + b = c ∈ Rn |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **13/7/2020** | **Name:** | **Nayanashree K S** | |
| **Course:** | **Salesforce** | **USN:** | **4al16ec042** | |
| **Topic:** | **Platform development basics** | **Semester & Section:** | **8 A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Report** Meet the Salesforce Platform At Salesforce, we group our services by clouds. There’s Sales Cloud for CRM, Service Cloud for customer support, and a handful of other clouds that help companies support their business functions. And while each of these clouds serves a unique purpose, there’s one thing they all have in common: the power of the Salesforce platform.  What is the Salesforce platform, exactly?  Like any platform, the Salesforce platform is a group of technologies that supports the development of other technologies on top of it. What makes it unique is that the platform supports not only all the Salesforce clouds, but it also supports custom functionality built by our customers and partners. This functionality ranges from simple page layouts to full-scale applications.  If you’re here today, we’re assuming you know a bit about software development. Throughout this module, we’re going to give you an overview of development on the Salesforce platform. We talk about some of the pillars of Salesforce development and how they work together to create a robust system. We even touch on some common questions that developers new to the platform run into as they get started.  Before we continue, let’s make sure we’re on the same page. If you’re brand new to Salesforce and you haven’t completed the [Salesforce Platform Basics module](https://trailhead.salesforce.com/modules/starting_force_com), we suggest you do that before you keep reading.  Once you’re done with that, you’re ready to get started! Platform Building Blocks As we mentioned, the platform not only forms the foundation of core Salesforce products like Sales Cloud and Service Cloud, but it also lets you build your own functionality. Building your own functionality can mean customizing existing Salesforce offerings or it can mean building something from scratch.  Let’s focus on that latter part and talk about what the Salesforce platform offers developers.  Our core platform lets you develop custom data models and applications for desktop and mobile. And with the platform behind your development, you can build robust systems at a rapid pace.  And then there’s the Heroku platform. Heroku gives developers the power to build highly scalable web apps and back-end services using Python, Ruby, Go, and more. It also provides database tools to sync seamlessly with data from Salesforce.  And then there’s the host of Salesforce APIs. These let developers integrate and connect all their enterprise data, networks, and identity information.  And then there’s the Mobile SDK. The Mobile SDK is a suite of technologies that lets you build native, HTML5, and hybrid apps that have the same reliability and security as the Salesforce app.  And then... wait. Let’s stop for a second.  The problem with the platform and all its parts is that listing them out takes a really long time. And just talking about them doesn’t help you understand everything they do. Let’s take a different approach and talk about what we can do with the platform. Or, more precisely, what we can build with it. The DreamHouse App Let’s float a scenario. Throughout the rest of this module, we use this scenario to explore the many exciting tools and technologies that the Salesforce platform provides.  You’re a developer for DreamHouse Realty, a company that aggregates real estate listings to better connect homebuyers and real estate agents. Your boss asks you to build a new system to track real estate listings. Your internal employees will use it to track and communicate about properties. Your partner real estate brokers will use it to access information about customers. And your customers will view properties and contact brokers for viewings.  DreamHouse Realty logo  Building an app like this one from scratch isn’t an easy thing to do. Taking on this project in real life can involve a long, complicated list of functional requirements and the implementation of special integrations for your company’s business data. Working by yourself, it can take you months to get something out the door.  But before your stress builds and you melt into a puddle of existential dread, remember: You’ve got the platform. And building complex business applications at a breakneck pace is what the platform’s all about.  We’re going to show you a fully functional version of the DreamHouse app so you can get a feel for how it was built. As we move through, we discuss important Salesforce development concepts using the app to guide us. Install the DreamHouse App To follow along and practice the steps in this module, you need to install the DreamHouse package in your Trailhead Playground. Follow the instructions here to launch a playground and install the package. You also use this package and playground when it’s time to complete the hands-on challenge. | | | |