[Module 1 - Defining Data Science](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/" \l "block-v1:IBM+DS0101EN+3T2020+type@chapter+block@e3f0d19e8fc743989c1029bf9801c8ac)

* Learn some key fundamentals in the field of data science.
* Hear from data science professionals on why they chose a career in data science.
* Hear from Professor Murtaza Haider on what skills make a data scientist successful.
* Read why data scientists are in high demand

## **Video: What is Data Science? (2:29)**

Data Science is a process, not an event. It is the process of using data to understand different things, to understand the world. For me is when you have a model or hypothesis of a problem, and you try to validate that hypothesis or model with your data.

Data science is the art of uncovering the insights and trends that are hiding behind data. It's when you translate data into a story. So use storytelling to generate insight. And with these insights, you can make strategic choices for a company or an institution.

Data science is a field about processes and systems to extract data from various forms of whether it is unstructured or structured form. Data science is the study of data. Like biological sciences is a study of biology, physical sciences, it's the study of physical reactions.

Data is real, data has real properties, and we need to study them if we're going to work on them. Data Science involves data and some science. The definition or the name came up in the 80s and 90s when some professors were looking into the statistics curriculum, and they thought it would be better to call it data science.

But what is Data Science? I'd see data science as one's attempt to work with data, to find answers to questions that they are exploring. In a nutshell, it's more about data than it is about science. If you have data, and you have curiosity, and you're working with data, and you're manipulating it, you're exploring it, the very exercise of going through analyzing data, trying to get some answers from it is data science.

Data science is relevant today because we have tons of data available. We used to worry about lack of data. Now we have a data deluge. In the past, we didn't have algorithms, now we have algorithms. In the past, the software was expensive, now it's open source and free. In the past, we couldn't store large amounts of data, now for a fraction of the cost, we can have gazillions of datasets for a very low cost.

So, the tools to work with data, the very availability of data, and the ability to store and analyze data, it's all cheap, it's all available, it's all ubiquitous, it's here. There's never been a better time to be a data scientist.

## **Fundamentals of Data Science (2:52)**

Everyone you ask will give you a slightly different description of what Data Science is, but most people agree that it has a significant data analysis component. Data analysis isn't new. What is new is the vast quantity of data available from massively varied sources: from log files, email, social media, sales data, patient information files, sports performance data, sensor data, security cameras, and many more besides. At the same time that there is more data available than ever, we have the computing power needed to make a useful analysis and reveal new knowledge. Data science can help organizations understand their environments, analyze existing issues, and reveal previously hidden opportunities.

Data scientists use data analysis to add to the knowledge of the organization by investigating data, exploring the best way to use it to provide value to the business. So, what is the process of data science? Many organizations will use data science to focus on a specific problem, and so it's essential to clarify the question that the organization wants answered. This first and most crucial step defines how the data science project progresses. Good data scientists are curious people who ask questions to clarify the business need. The next questions are: "what data do we need to solve the problem, and where will that data come from?". Data scientists can analyze structured and unstructured data from many sources, and depending on the nature of the problem, they can choose to analyze the data in different ways. Using multiple models to explore the data reveals patterns and outliers; sometimes, this will confirm what the organization suspects, but sometimes it will be completely new knowledge, leading the organization to a new approach. When the data has revealed its insights, the role of the data scientist becomes that of a storyteller, communicating the results to the project stakeholders. Data scientists can use powerful data visualization tools to help stakeholders understand the nature of the results, and the recommended action to take. Data Science is changing the way we work; it's changing the way we use data and it’s changing the way organisations understand the world.

## **Video: The Many Paths to Data Science (3:47)**

Data science didn't really exist when I was growing up. It's not something that I ever woke up and said, I want to be a data scientist when I grow up. No, it didn't exist. I didn't know I would be working in data science.

>> When I grew up, there isn't that field called data science. And I think it's really new.

>> Data science didn't exist until 2009, 2011. Someone like DJ Patil or Andrew Gelman coined the term. Before that, there was statistics. And I didn't want to be any of those. I wanted to be in business. And then I found data science a heck of a lot more interesting.

>> I studied statistics, that's how I started. I went through many different stages in my life where I wanted to be a singer and then a doctor. And then I realized that I was good at math. So I chose an area that was focused on quantitative analysis. And from then I do think that I wanted to work with data. Not necessarily data science as it's known today.

>> The first time that I had contact with data science, when I was my first year as a mechanical engineering. And strategic consulting firms, they use data science to make decisions. So it was my first contact with data science.

>> I had a complicated problem that I needed to solve, and the usual techniques that we had at the time couldn't help with that problem.

>> I graduated with a math degree in the worst possible time, right after the economic crisis, and you actually had to be useful to get a job. So I went and got a degree in statistics. And then I worked enough jobs that were called data scientist that I suddenly became one.

>> My undergraduate degree was in business, and I majored in politics, philosophy, and economics. And then I did a master's in business analytics at New York University at the Stern School of Business. When I left my undergrad, the first company I joined, it turned out that they were analyzing electronic point of sale data for retail manufacturers. And what we were doing was data science. But we only really started using that term much later. In fact, I'd say four or five years ago is when we started calling it analytics and data science.

>> I had several options for my internship here in Canada. And one of the options was to work with data science. I used to work with project development. But I think that was a good choice. And then I start my internship with data science.

>> I'm a civil engineer by training, so all engineers work with data. I would say the conventional use of data science in my life started with transportation research. I started building large models trying to forecast traffic on streets, trying to determine congestion and greenhouse gas emissions or tailpipe emissions. So I think that's where my start was. And I started building these models when I was a graduate student at the University of Toronto. Started working with very large data sets, looking at household samples of, say, 150,000 households from half a million trips. And that, too, I'm speaking from mid 90s when this was supposed to be a very large data set, but not in today's terms. But that's how I started. I continued working with it. And then I moved to McGill University where I was a professor of transportation engineering. And I built even bigger data models that involved data and analytics. And so I would say, yes, transportation research brought me to data science.

## **Advice for New Data Scientists (2:50)**

My advice to an aspiring data scientist is to be curious, extremely argumentative and judgmental. Curiosity is absolute must. If you're not curious, you would not know what to do with the data. Judgmental because if you do not have preconceived notions about things you wouldn't know where to begin with. Argumentative because if you can argument and if you can plead a case, at least you can start somewhere and then you learn from data and then you modify your assumptions and hypotheses and your data would help you learn. And you might start at the wrong point. You may say that I thought I believed this, but now with data I know this.

So, this allows you a learning process. So, curiosity being able to take a position, strong position, and then moving forward with it. The other thing that the data scientist would need is some comfort and flexibility with analytics platforms: some software, some computing platform, but that's secondary. The most important thing is curiosity and the ability to take positions. Once you have done that, once you've analyzed, then you've got some answers. And that's the last thing that a data scientist need, and that is the ability to tell a story. That once you have your analytics, once you have your tabulations, now you should be able to tell a great story from it. Because if you don't tell a great story from it, your findings will remain hidden, remain buried, nobody would know. But your rise to prominence is pretty much relying on your ability to tell great stories.

A starting point would be to see what is your competitive advantage. Do you want to be a data scientist in any field or a specific field? Because, let's say you want to be a data scientist and work for an IT firm or a web-based or Internet based firm, then you need a different set of skills. And if you want to be a data scientist in the health industry, then you need different sets of skills. So figure out first what you're interested, and what is your competitive advantage. Your competitive advantage is not necessarily going to be your analytical skills. Your competitive advantage is your understanding of some aspect of life where you exceed beyond others in understanding that. Maybe it's film, maybe it's retail, maybe it's health, maybe it's computers. Once you've figured out where your expertise lies, then you start acquiring analytical skills. What platforms to learn and those platforms, those tools would be specific to the industry that you're interested in. And then once you have got some proficiency in the tools, the next thing would be to apply your skills to real problems, and then tell the rest of the world what you can do with it.



[Module 1 - Summary](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@sequential+block@c7ad65ffd2ba4036a5ceae764d14997c)

In this module, you have learned:

* Data science is the study of large quantities of data, which can reveal insights that help organizations make strategic choices.
* There are many paths to a career in data science; most, but not all, involve a little math, a little science, and a lot of curiosity about data.
* New data scientists need to be curious, judgemental and argumentative.
* Why data science is considered the sexiest job in the 21st century, paying high salaries for skilled workers.

[Module 2 - What Data Scientists Do](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@chapter+block@1bb47f8741ce4722b70fe9aa10f67423).

* Learn how organizations are using data science to solve problems.
* Learn about some key concepts, tools and algorithms used in data science shared by data science professionals.
* Hear from Professor Murtaza Haider on how the cloud has expanded the role of the data scientist.

### A Day in the Life of a Data Scientist (3:45)

I've built a recommendation engine before as part of a large organization and worked through all types of engineers and accounting for different parts of the problem. It's one of the ones I'm most happy with because ultimately, I came up with a very simple solution that was easy to understand from all levels, from the executives to the engineers and developers. Ultimately, it was just as efficient as something really complex, and they could have spent a lot more time on. Back in the university, we have a problem that we wanted to predict algae blooms. This algae blooms could cause a rise in toxicity of the water and it could cause problems through the water treatment company. We couldn't like predict with our chemical engineering background.

So we use artificial neural networks to predict when these blooms will occur. So the water treatment companies could better handle this problem. In Toronto, the public transit is operated by Toronto Transit Commission. We call them TTC. It's one of the largest transit authorities in the region, in North America. And one day they contacted me and said, "We have a problem." And I said, "Okay, what's the problem?" They said, "Well, we have complaints data, and we would like to analyze it, and we need your help." I said, "Fine I would be very happy to help." So I said, "How many complaints do you have?" They said, "A few." I said, "How many?" Maybe half a million. I said, "Well, let's start working with it." So I got the data and I started analyzing it. So, basically, they have done a great job of keeping some data in tabular format that was unstructured data. And in that case, tabular data was when the complaint arrived, who received it, what was the type of the complaint, was it resolved, whose fault was it. And the unstructured part of it was the exchange of e-mails and faxes. So, imagine looking at how half a million exchanges of e-mails and trying to get some answers from it. So I started working with it. The first thing I wanted to know is why would people complain and is there a pattern or is there some days when there are more complaints than others?

And I had looked at the data and I analyzed it in all different formats, and I couldn't find the impetus for complaints being higher on a certain day and lower on others. And it continued for maybe a month or so. And then, one day I was getting off the bus in Toronto, and I was still thinking about it. And I stepped out without looking on the ground, and I stepped into a puddle, puddle of water. And now, I was sort of ankle deep into water, and it was just one foot wet and the other dry. And I was extremely annoyed. And I was walking back and then it hit me, and I said, "Well, wait a second. Today it rained unexpectedly, and I wasn't prepared for it. That's why I'm wet, and I wasn't looking forward." What if there was a relationship between extreme weather and the type of complaints TTC receives?

So I went to the environment Canada's website, and I got data on rain and precipitation, wind and the light. And there, I found something very interesting. The 10 most excessive days for complaints. The 10 days where people complain the most were the days when the weather was bad. It was unexpected rain, an extreme drop in temperature, too much snow, very windy day. So I went back to the TTC's executives and I said, "I've got good news and bad news." And the good news is, I know why people would complain excessively on certain days. I know the reason for it. The bad news is, there's nothing you can do about it.

## **Old Problems, New Problems, Data Science Solutions (3:56)**

Organizations can leverage the almost unlimited amount of data now available to them in a growing number of ways. However, all organizations ultimately use data science for the same reason—to discover optimum solutions to existing problems. Let’s take a look at three examples of data science providing innovative solutions for old problems. In transport, Uber collects real-time user data to discover how many drivers are available, if more are needed, and if they should allow a surge charge to attract more drivers. Uber uses data to put the right number of drivers in the right place, at the right time, for a cost the rider is willing to pay. In a different transport related data science effort, the Toronto Transportation Commission has made great strides in solving an old problem with traffic flows, restructuring those flows in and around the city.

Using data science tools and analysis, they have: Gathered data to better understand streetcar operations, and identify areas for interventions Analyzed customer complaints data Used probe data to better understand traffic performance on main routes Created a team to better capitalize on big data for both planning, operations and evaluation By focusing on peak hour clearances and identifying the most congested routes, monthly hours lost for commuters due to traffic congestion dropped from 4.75 hrs. in 2010 to 3 hrs. in mid-2014.

In facing issues in our environment, data science can also play a proactive role. Freshwater lakes supply a variety of human and ecological needs, such as providing drinking water and producing food. But lakes across the world are threatened by increasing incidences of harmful cyanobacterial blooms. There are many projects and studies to solve this long-existing dilemma. In the US, a team of scientists from research centers stretching from Maine to South Carolina is developing and deploying high-tech tools to explore cyanobacteria in lakes across the east coast.

The team is using robotic boats, buoys, and camera-equipped drones to measure physical, chemical, and biological data in lakes where cyanobacteria are detected, collecting large volumes of data related to the lakes and the development of the harmful blooms. The project is also building new algorithmic models to assess the findings. The information collected will lead to better predictions of when and where cyanobacterial blooms take place, enabling proactive approaches to protect public health in recreational lakes and in those that supply drinking water.

Such interdisciplinary training prepares the next generation of scientists to address societal issues with the proper modernized data science tools. It takes gathering a lot of data, cleaning and preparing it, and then analyzing it to gain the insight needed to develop better solutions for today's enterprises.

How do you get a better solution that is efficient? You must: Identify the problem and establish a clear understanding of it. Gather the data for analysis. Identify the right tools to use. Develop a data strategy. Case studies are also helpful in customizing a potential solution. Once these conditions exist and available data is extracted, you can develop a machine learning model. It will take time for an organization to refine best practices for data strategy using data science, but the benefits are worth it.

## **Data Science Topics and Algorithms (3:53)**

I really enjoy regression. I'd say regression was maybe one of the first concepts that I, that really helped me understand data so I enjoy regression. I really like data visualization. I think it's a key element for people to get across their message to people that don't understand that well what data science is.

Artificial neural networks. I'm really passionate about neural networks because we have a lot to learn with nature so when we are trying to mimic our, our brain I think that we can do some applications with this behavior with this biological behavior in algorithms.

Data visualization with R. I love to do this.

Nearest neighbor. It's the simplest but it just gets the best results so many more times than some overblown, overworked algorithm that's just as likely to overfit as it is to make a good fit. So structured data is more like tabular data things that you’re familiar with in Microsoft Excel format. You've got rows and columns and that's called structured data. Unstructured data is basically data that is coming from mostly from web where it's not tabular. It is not, it's not in rows and columns. It's text. It's sometimes it's video and audio, so you would have to deploy more sophisticated algorithms to extract data. And in fact, a lot of times we take unstructured data and spend a great deal of time and effort to get some structure out of it and then analyze it. So if you have something which fits nicely into tables and columns and rows, go head. That's your structured data. But if you see if it's a weblog or if you're trying to get information out of webpages and you've got a gazillion web pages, that's unstructured data that would require a little bit more effort to get information out of it. There are thousands of books written on regression and millions of lectures delivered on regression. And I always feel that they don’t do a good job of explaining regression because they get into data and models and statistical distributions. Let's forget about it. Let me explain regression in the simplest possible terms. If you have ever taken a cab ride, a taxi ride, you understand regression. Here is how it works.

The moment you sit in a cab ride, in a cab, you see that there's a fixed amount there. It says $2.50. You, rather the cab, moves or you get off. This is what you owe to the driver the moment you step into a cab. That's a constant. You have to pay that amount if you have stepped into a cab. Then as it starts moving for every meter or hundred meters the fare increases by certain amount. So there's a... there's a fraction, there's a relationship between distance and the amount you would pay above and beyond that constant. And if you're not moving and you're stuck in traffic, then every additional minute you have to pay more. So as the minutes increase, your fare increases. As the distance increases, your fare increases. And while all this is happening you've already paid a base fare which is the constant. This is what regression is.

Regression tells you what the base fare is and what is the relationship between time and the fare you have paid, and the distance you have traveled and the fare you've paid. Because in the absence of knowing those relationships, and just knowing how much people traveled for and how much they paid, regression allows you to compute that constant that you didn't know. That it's $2.50, and it would compute the relationship between the fare and and the distance and the fare and the time. That is regression.

## **Cloud for Data Science (3:22)**

Cloud is -- it's a godsend for data scientists, primarily because you're able to take the, or you take your data, take your information and put it in the cloud, put it in the central storage system. It allows you to bypass the physical limitations of the computers and the systems you're using and it allows you to deploy the analytics and storage capacities of advanced machines that do not necessarily have to be your machine or your company's machine. And cloud allows you not just to store large amounts of data on servers somewhere in California or in Nevada, but it also allows you to deploy very advanced computing algorithms and the ability to do high-performance computing using machines that are not yours. So, and think of it as you have some information you can't store it so you send it to storage space, let's call it cloud, and the algorithms that you need to use, you don't have them with you but then on the cloud you have those, those algorithms available. So what you do is you deploy those algorithms on very large data sets. And you're able to do it even though your own systems, your own machines, your own computing environments were not allowing you to do so. So cloud is beautiful. And the other thing that cloud is beautiful for is that it allows multiple entities to work with same data at the same time. So you can be working with the same data that your colleagues in say, Germany, and another team in India, and another team in in Ghana. They are collectively working and they are able to do so because the information and the algorithms and the tools and the answers and the results-- whatever they needed—is available at a central place which we call cloud. So cloud is beautiful.

Using the cloud enables you to get instant access to open source technologies like Apache Spark without the need to install and configure them locally. Using the cloud also gives you access to the most up-to-date tools and libraries without the worry of maintaining them and ensuring that they are up-to-date. The cloud is accessible from everywhere and in every time zone. You can use cloud-based technologies from your laptop, from your tablet, and even from your phone, enabling collaboration more easily than ever before. Multiple collaborators or teams can access the data simultaneously, working together on producing a solution. Some big tech companies offer cloud platforms allowing you to become familiar with cloud-based technologies in a pre-built environment. IBM offers the IBM Cloud, Amazon offers Amazon Web Services or AWS, and Google offers Google Cloud Platform. IBM also provides skills, Network labs, or SN labs to learners. Register to any of the learning portals on the IBM developer skills Network where you have access to tools like Jupyter notebooks and Spark clusters so you can create your own data science project and develop solutions. With practice and familiarity you will discover how the cloud dramatically enhances productivity for data scientists.

## **What Makes Someone a Data Scientist (**Chapter 1 Pg. 12-15)

Now that you know what is in the book, it is time to put down some definitions. Despite their ubiquitous use, consensus evades the notions Of big data and data science. The question, Who is a data scientist? is very much alive and being contested by individuals, some of whom are merely interested in protecting their discipline or academic turfs. In this section, I attempt to address these controversies and explain Why a narrowly construed definition Of either big data Or data science will result in excluding hundreds Of thousands of individuals who have recently turned to the emerging field.

Everybody loves a data scientist, wrote Simon Rogers (2012) in the Guardian. Mr. Rogers also traced the newfound love for number crunching to a quote by Google's Hal Varian, who declared that the sexy job in the next ten years Will be statisticians.

Whereas Hal Varian named statisticians sexy, it is widely believed that what he really meant were data scientists. This raises several important questions:

* What is data science?
* How does it differ from statistics?
* What makes someone a data scientist?

In the times of big data, a question as simple as, What is data science? can result in many answers. In some cases, the diversity Of opinion on these answers borders on hostility.

I define data scientist as someone Who finds solutions to problems by analyzing big or small data using appropriate tools and then tells stories to communicate her findings to the relevant stakeholders. I do not use the data size as a restrictive clause. A data below a certain arbitrary threshold does not make one less Of a data scientist. Nor is my definition Of a data scientist restricted to particular analytic tools, such as machine learning. As long as one has a curious mind, fluency in analytics, and the ability to communicate the findings, I consider the person a data scientist.

I define data science as something that data scientists do. Years ago, as an engineering student at the University Of Toronto I was stuck With the question: What is engineering? I wrote my master's thesis on forecasting housing prices and my doctoral dissertation on forecasting homebuilders' choices related to What they build, when they build, and where they build new housing. In the civil engineering department, Others were working on designing buildings, bridges, tunnels, and worrying about the stability Of slopes. My work, and that of my supervisor, was not your traditional garden-variety engineering. Obviously, I was repeatedly asked by others whether my research was indeed engineering.

When I shared these concerns with my doctoral supervisor, Professor Eric Miller, he had a laugh. Dr. Miller spent a lifetime researching urban land use and transportation, and had earlier earned a doctorate from MIT. Engineering is what engineers do, he responded. Over the next 1 7 years, I realized the wisdom in his statement. You first become an engineer by obtaining a degree and then registering with the local professional body that regulates the engineering profession. Now you are an engineer. You can dig tunnels; write software codes; design components of an iPhone or a supersonic jet. You are an engineer. And when you are leading the global response to financial crisis in your role as the chief economist Of the International Monetary Fund (IMF), as Dr. Raghuram Rajan did, you are an engineer.

Professor Raghuram Rajan did his first degree in electrical engineering from the Indian Institute Of Technology. He pursued economics in graduate studies, later became a professor at a prestigious university, and eventually landed at the IMF. He is currently serving as the 23rd Governor of the Reserve Bank of India. Could someone argue that his intellectual prowess is rooted only in his training as an economist and that the fundamentals he learned as an engineering student played no role in developing his problem-solving abilities?

Professor Rajan is an engineer. So are Xi Jinping, the President of the People's Republic of China, and Alexis Tsipras, the Greek prime Minister who is forcing the world to rethink the fundamentals of global economics. They might not be designing new circuitry, distillation equipment, or bridges, but they are helping build better societies and economies and there can be no better definition of engineering and engineers—that is, individuals dedicated to building better economies and societies.

So briefly, I would argue that data science is what data scientists do.

Others have much different definitions. In September 2015, a co-panelist at a meetup organized by in Toronto confined data science to machine learning. There you have it. If you are not using the black boxes that make up machine learning, as per some experts in the field, you are not a data scientist. Even if you were to discover the cure to a disease threatening the lives Of millions, turf-protecting colleagues Will exclude you from the data science club.

Dr. Vincent Granville (2014), an author On data science, offers certain thresholds to meet to be a data scientist.2-9 On pages 8 and 9 in Developing Analytic Talent Dr. Granville describes the new data science professor as a non-tenured instructor at a non-traditional university, Who publishes research results in online blogs, does not waste time writing grants, works from home, and earns more money than the traditional tenured professors. Suffice it to say that the thriving academic community Of data scientists might disagree with Dr. Granville.

Dr. Granville uses restrictions on data size and methods to define what data science is. He defines a data scientist as one who can easily process a So-million-row data set in a couple of hours, and who distrusts (statistical) models. He distinguishes data science from statistics. Yet he lists algebra, calculus, and training in probability and statistics as necessary background to understand data science (page 4).

Some believe that big data is merely about crossing a certain threshold on data size or the number of observations, or is about the use of a particular tool, such as Hadoop. Such arbitrary thresholds on data size are problematic because with innovation, even regular computers and off-the-shelf software have begun to manipulate very large data sets. Stata, a commonly used software by data scientists and statisticians, announced that one could now process between 2 billion to 24.4 billion rows using its desktop solutions. If Hadoop is the password to the big data club, Stata's ability to process 24.4 billion rows, under certain limitations, has just gatecrashed that big data party.

It is important to realize that one who tries to set arbitrary thresholds to exclude others is likely to run into inconsistencies. The goal should be to define data science in a more exclusive, discipline- and platform- size-free context where data-centric problem solving and the ability to weave strong narratives take center Stage.

Given the controversy, I would rather consult others to see how they describe a data scientist. Why don't we again consult the Chief Data Scientist of the United States? Recall Dr. patil told the Guardian newspapRr in 2012 that a "data scientist is that unique blend of skills that can both unlock the insights of data and tell a fantastic story via the data.' What is admirable about Dr. Patil's definition is that it is inclusive of individuals Of various academic backgrounds and training, and does not restrict the definition Of a data scientist to a particular tool or subject it to a certain arbitrary minimum threshold Of data size.

The other key ingredient for a successful data scientist is a behavioural trait: curiosity. A data scientist has to be one with a very curious mind, willing to spend significant time and effort to explore her hunches. In journalism, the editors call it having the nose for news. Not all reporters know where the news lies. Only those Who have the nose for news get the Story. Curiosity is equally important for data scientists as it is for journalists.

Rachel Schutt is the Chief Data Scientist at News Corp. She teaches a data science course at Columbia University. She is also the author Of an excellent book, Doing Data Science. In an interview With the New York Times, Dr. Schutt defined a data scientist as someone Who is part computer scientist, part software engineer, and part statistician (Miller, 2013). But that's the definition Of an average data scientist. The best, she contended, tend to be really curious people, thinkers who ask good questions and are O.K. dealing with unstructured situations and trying to find structure in them.

## **Module 2 Summary:**

* The typical workday for a Data Scientist varies depending on what type of project they are working on.
* Many algorithms are used to bring out insights from data.
* Accessing algorithms, tools, and data through the Cloud enables Data Scientists to stay up-to-date and collaborate easily.

[Module 3 - Big Data and Data Mining](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@chapter+block@c84404f7f0234038b64b5820ced52fc9):

* Learn about the 5 Vs of Big Data.
* Learn about how Hadoop and other tools are handling the demands of big data.
* Hear from Norman White, Professor at New York University on data science and big data.
* Learn about data mining and the steps that comprise the process of mining a given data set.

### Foundations of Big Data (5:21)

In this digital world, everyone leaves a trace. From our travel habits to our workouts and entertainment, the increasing number of internet connected devices that we interact with on a daily basis record vast amounts of data about us.

There’s even a name for it: Big Data. Ernst and Young offers the following definition: “Big Data refers to the dynamic, large and disparate volumes of data being created by people, tools, and machines. It requires new, innovative, and scalable technology to collect, host, and analytically process the vast amount of data gathered in order to derive real-time business insights that relate to consumers, risk, profit, performance, productivity management, and enhanced shareholder value.”

There is no one definition of Big Data, but there are certain elements that are common across the different definitions, such as velocity, volume, variety, veracity, and value. These are the V's of Big Data.

Velocity is the speed at which data accumulates. Data is being generated extremely fast, in a process that never stops. Near or real-time streaming, local, and cloud-based technologies can process information very quickly.

Volume is the scale of the data, or the increase in the amount of data stored. Drivers of volume are the increase in data sources, higher resolution sensors, and scalable infrastructure.

Variety is the diversity of the data. Structured data fits neatly into rows and columns, in relational databases while unstructured data is not organized in a pre-defined way, like Tweets, blog posts, pictures, numbers, and video. Variety also reflects that data comes from different sources, machines, people, and processes, both internal and external to organizations. Drivers are mobile technologies, social media, wearable technologies, geo technologies, video, and many, many more.

Veracity is the quality and origin of data, and its conformity to facts and accuracy. Attributes include consistency, completeness, integrity, and ambiguity. Drivers include cost and the need for traceability. With the large amount of data available, the debate rages on about the accuracy of data in the digital era. Is the information real, or is it false?

Value is our ability and need to turn data into value. Value isn't just profit. It may have medical or social benefits, as well as customer, employee, or personal satisfaction. The main reason that people invest time to understand Big Data is to derive value from it.

Let's look at some examples of the V's in action.

Velocity: Every 60 seconds, hours of footage are uploaded to YouTube which is generating data. Think about how quickly data accumulates over hours, days, and years.

Volume: The world population is approximately seven billion people and the vast majority are now using digital devices; mobile phones, desktop and laptop computers, wearable devices, and so on. These devices all generate, capture, and store data -- approximately 2.5 quintillion bytes every day. That's the equivalent of 10 million Blu-ray DVD's.

Variety: Let's think about the different types of data; text, pictures, film, sound, health data from wearable devices, and many different types of data from devices connected to the Internet of Things.

Veracity: 80% of data is considered to be unstructured and we must devise ways to produce reliable and accurate insights. The data must be categorized, analyzed, and visualized.

Data Scientists today derive insights from Big Data and cope with the challenges that these massive data sets present. The scale of the data being collected means that it’s not feasible to use conventional data analysis tools. However, alternative tools that leverage distributed computing power can overcome this problem. Tools such as Apache Spark, Hadoop and its ecosystem provide ways to extract, load, analyze, and process the data across distributed compute resources, providing new insights and knowledge.

This gives organizations more ways to connect with their customers and enrich the services they offer. So next time you strap on your smartwatch, unlock your smartphone, or track your workout, remember your data is starting a journey that might take it all the way around the world, through big data analysis, and back to you.

## **How Big Data is Driving Digital Transformation (3:55)**

Digital Transformation affects business operations, updating existing processes and operations and creating new ones to harness the benefits of new technologies. This digital change integrates digital technology into all areas of an organization resulting in fundamental changes to how it operates and delivers value to customers. It is an organizational and cultural change driven by Data Science, and especially Big Data.

The availability of vast amounts of data, and the competitive advantage that analyzing it brings has triggered digital transformations throughout many industries. Netflix moved from being a postal DVD lending system to one of the world’s foremost video streaming providers, the Houston Rockets NBA team used data gathered by overhead cameras to analyze the most productive plays, and Lufthansa analyzed customer data to improve its service. Organizations all around us are changing to their very core.

Let’s take a look at an example, to see how Big Data can trigger a digital transformation, not just in one organization, but in an entire industry. In 2018, the Houston Rockets, a National Basketball Association, or NBA team, raised their game using Big Data. The Rockets were one of four NBA teams to install a video tracking system which mined raw data from games. They analyzed video tracking data to investigate which plays provided the best opportunities for high scores, and discovered something surprising. Data analysis revealed that the shots that provide the best opportunities for high scores are two-point dunks from inside the two-point zone, and three-point shots from outside the three-point line, not long-range two-point shots from inside it.

This discovery entirely changed the way the team approached each game, increasing the number of three-point shots attempted. In the 2017-18 season, the Rockets made more three-point shots than any other team in NBA history, and this was a major reason they won more games than any of their rivals. In basketball, Big Data changed the way teams try to win, transforming the approach to the game.

Digital transformation is not simply duplicating existing processes in digital form; the in-depth analysis of how the business operates helps organizations discover how to improve their processes and operations, and harness the benefits of integrating data science into their workflows. Most organizations realize that digital transformation will require fundamental changes to their approach towards data, employees, and customers, and it will affect their organizational culture. Digital transformation impacts every aspect of the organization, so it is handled by decision makers at the very top levels to ensure success. The support of the Chief Executive Officer is crucial to the digital transformation process, as is the support of the Chief Information Officer, and the emerging role of Chief Data Officer. But they also require support from the executives who control budgets, personnel decisions, and day-to-day priorities. This is a whole organization process. Everyone must support it for it to succeed. There is no doubt dealing with all the issues that arise in this effort requires a new mindset, but Digital Transformation is the way to succeed now and in the future.

## **What is Hadoop? (6:36)**

Traditionally in computation and processing data we would bring the data to the computer. You'd wanna program and you'd bring the data into the program. In a big data cluster what Larry Page and Sergey Brin came up with is very simple is they took the data and they sliced it into pieces and they distributed each and they replicated each piece or triplicated each piece and they would send it the pieces of these files to thousands of computers first it was hundreds but then now it's thousands now it's tens of thousands.

And then they would send the same program to all these computers in the cluster. And each computer would run the program on its little piece of the file and send the results back. The results would then be sorted and those results would then be redistributed back to another process. The first process is called a map or a mapper process and the second one was called a reduce process. Fairly simple concepts but turned out that you could do lots and lots of different kinds of handle lots and lots of different kinds of problems and very, very, very large data sets.

So the one thing that's nice about these big data clusters is they scale linearly. You had twice as many servers and you get twice the performance and you can handle twice the amount of data. So this was just broke a bottleneck for all the major social media companies. Yahoo then got on board. Yahoo hired someone named Doug Cutting who had been working on a clone or a copy of the Google big data architecture and now that's called Hadoop.

And if you google Hadoop you'll see that it's now a very popular term and there are many, many, many if you look at the big data ecology there are hundreds of thousands of companies out there that have some kind of footprint in the big data world. Most of the components of data science have been around for many, many, many decades. But they're all coming together now with some new nuances I guess. At the bottom of data science you see probability and statistics.

You see algebra, linear algebra you see programming and you see databases. They've all been here. But what's happened now is we now have the computational capabilities to apply some new techniques - machine learning. Where now we can take really large data sets and instead of taking a sample and trying to test some hypothesis we can take really, really large data sets and look for patterns. And so back off one level from hypothesis testing to finding patterns that maybe will generate hypotheses. Now this can bother some very traditional statisticians and gets them really annoyed sometimes that you know you're supposed to have a hypothesis that is not that is independent of the data and then you test it. So once some of these machine learning techniques started were really the only thing the only way you can analyze some of these really large social media data sets. So what we've seen is that the combination of traditional areas computer science probability, statistics, mathematics all coming together in this thing that we call Decision Sciences. Our department at Stern I'll give a little plug here we happen to have been very well situated among business schools because we're one of the few business schools that has a real statistics department with real PhD level statisticians in it.

We have an operations management department and an information systems department. So we have a wide range of computer scientists to statisticians, to operations researchers. And so we were perfectly positioned as a couple of other business schools were to jump on this bandwagon and say; okay this is Decision Sciences.

And Foster Provost who's in my department was the first director of the NYU Center for Data Science. Four years ago maybe five years ago. I mean, I feel this is one of those cases where you can just to Google and search for data science and see how often it occurred and you'll see almost nothing and then just a spike. The same thing you would see with big data about seven or eight years ago. So data science is a term I haven't heard of probably five years ago. The first question is what is it? And I think faculty and everybody is still trying to get their hands around exactly what is business analytics and what is data science.

We certainly know the components of it. But it's morphing and changing and growing. I mean the last three years deep learning has just been added into the mix. Neural networks have been around for 20 or 30 years. 20 years ago I would teach neural networks in a class and you really couldn't do very much with them. And now some researchers have come up with multi-layer neural networks in Toronto in particular the University of Toronto. And that technology is now rapidly expanding it's being used by Google, by Facebook, by lots of companies.

**Data Science Skills and Big Data (4:35)**

I'm Norman White, I'm a Clinical Faculty Member in the IOMS Department, Information, Operations and Management Science Department here at Stern. I've been here for a long time (laughs), since I got out of college, pretty much. I'm sort of a techy, geeky kind of person. I really like to play with technology in my spare time. I'm currently Faculty Director of the Stern Center for Research Computing, in which we have a private cloud that runs lots of different kinds of systems. Many of our faculty or PhD students who need specialized hardware and software will come to us, we'll spin up a machine for them, configure it, I'll help them and advise on them.

A lot of the data scientists, or virtually all the data scientists at Stern use our facilities. And their PhD students use them a lot. I have an undergraduate degree in Applied Physics and while I was an undergrad I took a number of economics courses, so I ended up deciding to go to business school, but I had, this was in the early days of computers and I had gotten interested in computers. I came to Stern, which was then NYU Business School downtown and they had a little computer center, and I decided that I was gonna learn two things while I was there.

One, I was gonna learn how to program. I had taken one programming course in college. And I was gonna learn how to touch type. I never did learn how to touch type (laughs). Or maybe I did but I've forgotten now, and back to two finger pecking. But I became a self taught programmer, and then I took a number of courses at IBM because I eventually came the director of the computer center while I was getting my PhD in Economics and Statistics at Stern.

In 1973, the school formed a department called Computer Applications and Information Systems and I was one of the first faculty members in the department and I've been here ever since (laughs). My typical Monday is, I usually get in around 11 o'clock and I do my email at home first, but I come in and I have two classes on Monday. I have a class on design and development of web based systems at six o'clock. Two o'clock, I have a dealing with data class. The class is based on Python notebooks, so we start with the basics of Unix and Linux, just to get the students used to that. We move onto some Python, some regular expressions, a lot of relational databases, some Python Pandas, which is sort of like R for Python, lets you do mathematical and statistical calculations in Python. And then I end up with big data, for which, as you probably know, I'm an evangelist. The students I have, weekly homeworks. I put them in teams and they have to do a big project at the end of the term, and they do some really cool things. Yes, in fact, the whole course is taught using Jupyter notebooks. Every student has their own virtual machine on Amazon Web Services, so we pre configure all the machines and they get a standard image that has all of the materials for the course either loaded on it or in a Jupyter notebook, there are the commands to download it or update the server with the right software. So everybody is in the same environment, it doesn't matter what kind of, whether they have a Mac or a Windows machine or how old it is, everybody can do everything in the class.

## **Data Scientists at New York University (4:13)**

Everybody knows how to program, at least a little bit. They all have a little bit of programming background at least, and some of them have a lot. Some of them are Masters of Science and Computer Science, some of them are MBA students who've come in from technical fields and programmed every day. And others are ones who maybe took a programming course in college four or five years ago but at least they can think computationally, which I think is the most important thing that they need.

Data science and business analytics have become very hot subjects in the last four or five years. We have new tools, we have new approaches, and we have lots and lots of data that traditional techniques just couldn't really store and handle. I think the word is out. I think at this point, at first, companies and employers understood the need, especially in certain fields.

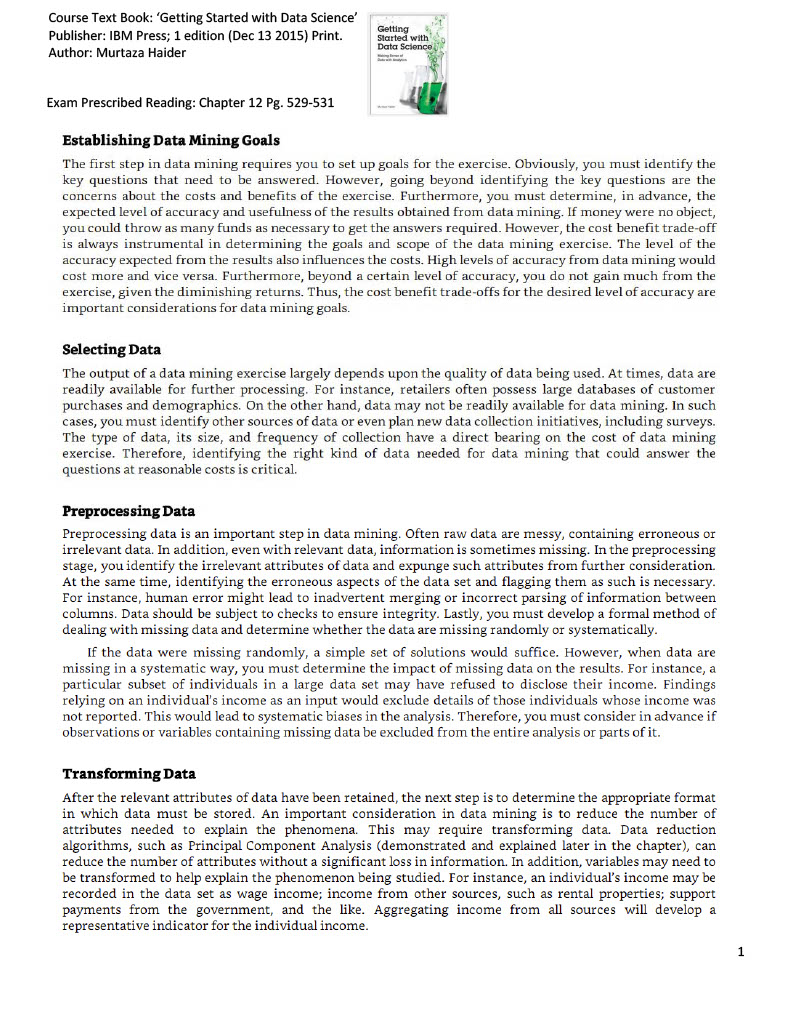
I can remember talking to a major bank three years ago about big data and there was one little group in the bank where one person had a little effort in putting a little cluster together. Now that same bank has five or six major big data clusters and they're putting all of their credit card data in it and they're grinding it upside down and sideways, using all sorts of data science kinds of techniques. Two years ago, or was it last year, I think, our undergraduate dealing with data course had 28 students in it. This year it has 140. So that means that the parents are now beginning to get the word, because one thing we understand with our undergrads is the parents who are paying very hefty tuitions, they, you know, they tell their sons and daughters, "You know, you should be an accountant," right? Or, "You should go into financial services, "or into marketing, 'cause this is where the money is." Now, they're getting the word that maybe you should take some more STEM classes in high school and be ready to go into data science or go into fields where analytics has become more and more important.

It depends on who you are (laughs). I have my own definition of big data. My definition of big data is data that is large enough and has enough volume and velocity that you cannot handle it with traditional database systems. Some of our statisticians think big data is something you can't fit on a thumb drive. Big data, to me, was started by Google. When Google tried to figure out how they were, when Larry Page and Sergey Brin wanted to, basically, figure out how to solve their page rank algorithm, there was nothing out there.

They were trying to store all of the web pages in the world, and there was no technology, there was no way to do this, and so they went out and developed this approach, which has now become, Hadoop has copied it, but this is where all these large, big data clusters are found. But big data has now also expanded into, how do you analyze?

There are new analytical techniques and statistical techniques for handling these really, really, really large data sets. We'll probably get to deep learning at some point along here.

## **Data Mining**



[Module 3 - Summary](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@sequential+block@d47ed57a4e4b456ca4704c31597dcc57):

How Big Data is defined by the Vs: Velocity, Volume, Variety, Veracity, and Value.

* How Hadoop and other tools, combined with distributed computing power,  are used to handle the demands of Big Data.
* What skills are required to analyse Big Data.
* About the process of Data Mining, and how it produces results.

[Module 4 - Deep Learning and Machine Learning](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@chapter+block@225a1945b14f4bf19de9c87ce51b84d9):

* Learn the difference between Machine Learning and Deep Learning.
* Learn about some of the many applications of Machine Learning.
* Learn about regression and what questions can be put to regression analysis.

## **What's the Difference? (4:11)**

In Data Science, there are many terms that are used interchangeably, so let’s explore the most common ones. The term Big Data refers to data sets that are so massive, so quickly built, and so varied that they defy traditional analysis methods such as you might perform with a relational database. The concurrent development of enormous compute power in distributed networks and new tools and techniques for data analysis means that organizations now have the power to analyse these vast data sets, and new knowledge and insights are becoming available to everyone.

Big data is often described in terms of five Vs - Velocity, Volume, Variety, Veracity, and Value. Data mining is the process of automatically searching and analyzing data, discovering previously unrevealed patterns. It involves preprocessing the data to prepare it and transforming it into an appropriate format. **Once this is done, insights and patterns are mined and extracted using various tools and** techniques ranging from simple data visualization tools to machine learning and statistical models. Machine learning is a subset of AI that uses computer algorithms to analyze data and make intelligent decisions based on what it has learned, without being explicitly programmed. Machine learning algorithms are trained with large sets of data and they learn from examples. They do not follow rules-based algorithms.

Machine learning is what enables machines to solve problems on their own and make accurate predictions using the provided data. Deep learning is a specialized subset of Machine Learning that uses layered neural networks to simulate human decision-making. Deep learning algorithms can label and categorize information and identify patterns.

It is what enables AI systems to continuously learn on the job, and improve the quality and accuracy of results by determining whether decisions were correct. Artificial neural networks, often referred to simply as neural networks, take inspiration from biological neural networks, although they work quite a bit differently. A neural network in AI is a collection of small computing units called neurons that take incoming data and learn to make decisions over time.

Neural networks are often layered deep and are the reason deep learning algorithms become more efficient as the datasets increase in volume, as opposed to other machine learning algorithms that may plateau as data increases. Now that you have a broad understanding of the differences between some key AI concepts, there is one more differentiation that is important to understand; that between artificial intelligence and data science.

Data science is the process and method for extracting knowledge and insights from large volumes of disparate data. It's an interdisciplinary field involving mathematics, statistical analysis, data visualization, machine learning, and more. It's what makes it possible for us to appropriate information, see patterns, find meaning from large volumes of data, and use it to make decisions that drive business.

Data Science can use many of the AI techniques to derive insight from data. For example, it could use machine learning algorithms and even deep learning models to extract meaning and draw inferences from data. There is some intersection between AI and data science, but one is not a subset of the other. Rather, data science is a broad term that encompasses the entire data processing methodology. While AI includes everything that allows computers to learn how to solve problems and make intelligent decisions. Both AI and Data Science can involve the use of big data, that is significantly large volumes of data.

## **Neural Networks and Deep Learning (6:42)**

It's a, I guess Computer Science's attempt to mimic a real, the neurons and how our brain actually functions. So 20, 30 years ago a neural network would have some inputs that would come in. They would be fed into different processing nodes that would then do some transformation on them and aggregate them or something, and then maybe go to another level of nodes and finally some output would come out. And I can remember training a neural network to recognize digits, handwritten digits and stuff.

So a neural network is trying to use a computer program that will mimic how neurons, how our brains use neurons to process things, brains to synapse, neurons to synapses and building these complex networks that can be trained. So a neural network starts out with some inputs and some outputs and you keep feeding these inputs in to try to see what kinds of transformations will get to these outputs, and you keep doing this over and over and over again in a way that this network should converge so these input, the transformations will eventually get these outputs.

The problem with neural networks was that even though the theory was there and they did work on small problems, like recognizing handwritten digits and things like that, they were computationally very intensive, and so they went out of favor. I stopped teaching them, well, probably 15 years ago. Then all of a sudden we started hearing about deep learning. I heard the term deep learning. This is another term that when did you first hear it? Fours years ago, five years ago? So I finally said, "What the hell is deep learning? It's really doing all this great stuff. What is it?" I Google it and I find this is neural networks on steroids. What they did was they just had more multiple layers of neural networks and they use lots and lots and lots of computing power to solve them. Just before this interview I had a young faculty member in the marketing department whose research is partially based on deep learning. She needs a computer that has a graphics processing unit in it because it takes an enormous amount of matrix and linear algebra calculations to actually do all of the mathematics that you need in neural networks, but they are now quite capable. We now have neural networks and deep learning that can recognize speech, can recognize people. If you're out there and getting your face recognized, I guarantee that NSA has a lot of work going on in neural networks. The University, right now, as Director of Research Computing, I have some small set of machines, down at our South Data Center, and I went in there last week and there were just piles and piles and piles of cardboard boxes all from Dell with a GPU on the side. Well, a GPU is a graphics processing unit. There is only one application in this University that needs 200 servers, each with graphics processing units in it, and each graphics processing unit has the equivalent of 600 cores of processing, so this is tens of thousands of processing cores. That is for deep learning. I guarantee.

Some of the first ones are speech recognition. Yann LeCun who teaches the deep learning class at NYU and is also the Head Data Scientist at Facebook, comes into class with a notebook, and it's a pretty thick notebook. It looks a little odd because it's like this. It's that thick because it has a couple of graphics processing units in it and then he will ask the class to start to speak to this thing and it will train while he's in class, he will train a neural network to recognize speech. So recognizing speech, recognizing people, images, classifying images, almost all of the traditional tasks that neural nets used to work on in little tiny things, now they can do really, really large things. It will learn, on it's own, the difference between a cat and a dog and different kinds of objects. It doesn't have to be taught. It doesn't, it just learns. That's why they call it deep learning, and if you hear, he plays this. If you hear how it recognizes speech and generates speech, it sounds like a baby learning to talk. You can just, you're like (babbles) All of sudden this stupid machine is talking to you and learned how to talk. That's cool.

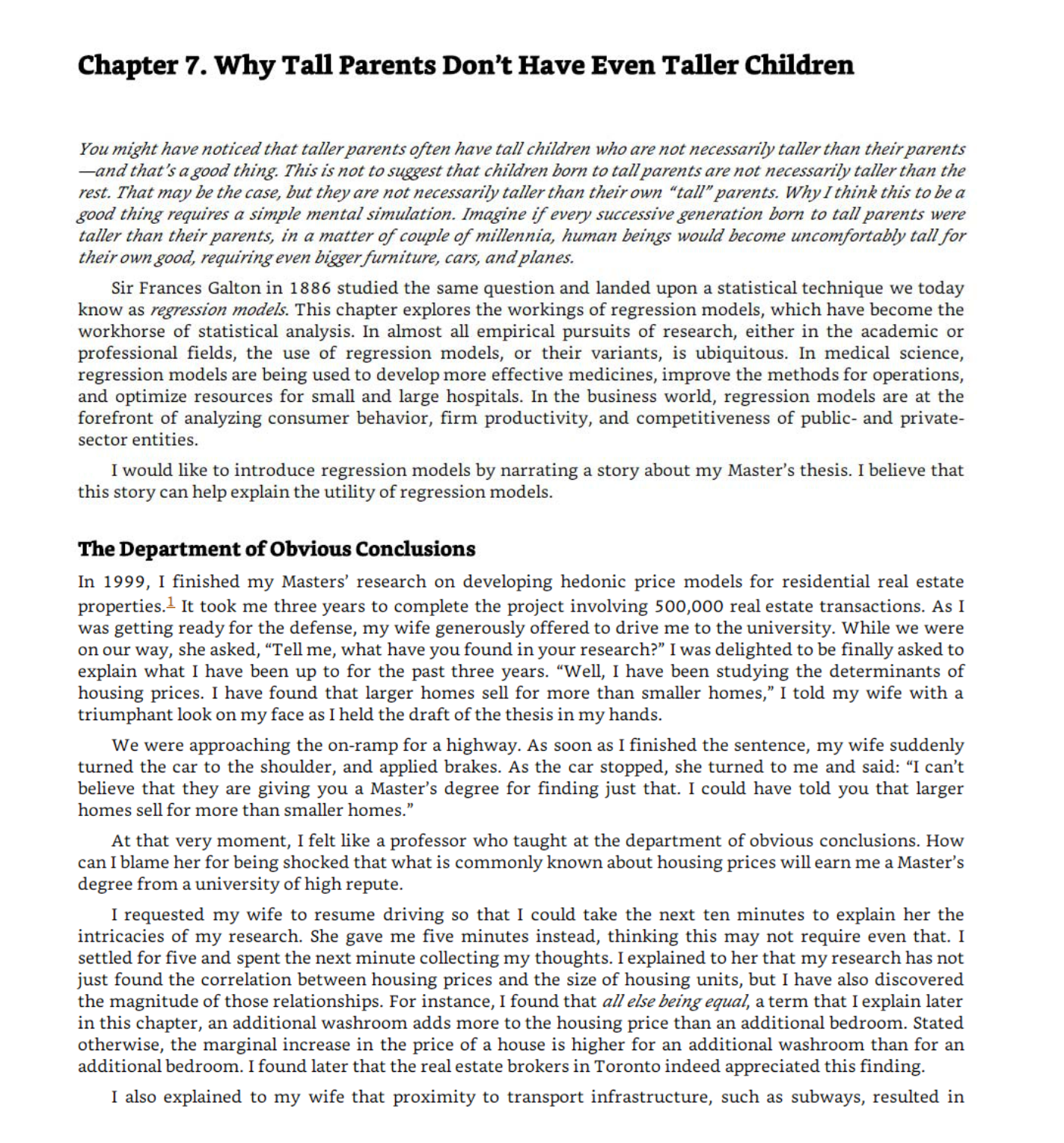
You need to learn some linear algebra. A lot of this stuff is based on matrix and linear algebra, so you need to know how to do, use linear algebra and do transformations. Now, on the other hand, there's now lots of packages out there that will do deep learning and they'll do all the linear algebra for you. But you should have some idea of what is happening underneath. Deep learning, in particular, needs really high powered computational power. So it's not something that you're going to go out and do on your notebook for, you could play with it, but if you really want to do it seriously you have to have some special computational resources.

## **Applications of Machine Learning (3:18)**

Everybody now deals with machine learning, but recommender systems are certainly one of the major applications, classifications, cluster analysis, trying to find some of the some of the marketing questions from 20 years ago, market basket analysis, what goods tend to be bought together. That was computationally a very difficult problem. I mean we are now doing that all the time with machine learning. So predictive analytics is another area of machine learning. We're using new techniques to predict things that statisticians don't particularly like. Decision Trees, Bayesian Analysis, Naive Bayes, lots of different techniques. The nice thing about them is that in packages like R now, you really have to understand how these techniques can be used and you don't have to know exactly how to do them but you have to understand what their meanings are.

Precision versus recall and the problems of over sampling and overfitting so you can, someone who knows a little bit about data science can apply these techniques, but they really need to know maybe not the details of the technique as much as how, what the trade-offs are. So some applications of machine learning in FinTech are probably a couple of different things I can talk about there. One of them is recommendations, right? So when you use Netflix, or you use Facebook, or a lot of different software services, the recommendations are served to you. Meaning, "Hey, you are a user, you have watched this show, so maybe you'd like to see this other show." Or, "You follow this person, so maybe you should follow that other person." It's actually kind of the same thing in FinTech. Because you've looked at, if you are an investment professional, and because you have looked at this investment idea, it might be really cool for you to look at this other investment idea, which is kind of similar. It is a similar kind of asset, or it is a similar kind of company, or it is a similar kind of technique for doing the investment. So we can apply recommendations using machine learning throughout a lot of different parts of FinTech. Another one that people talk about and is important, especially in the retail aspects of banking and finance, is fraud detection; trying to determine whether a charge that comes through a credit card is fraudulent or not, in real time, is a machine learning problem. You have to learn from all of the transactions that have happened previously. And build a model. And when the charge comes through, you have to compute all this stuff, and say, "Yeah, we think that's okay." Or, "Hmm, that's not so that's not so good, let's route it to our fraud people to check."

## **Regression**



[Module 4 - Summary](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@sequential+block@9c1af29539df4d3ba32f84c3b91b9845):

* Learned the differences between some common Data Science terms, including Deep Learning and Machine Learning.
* Deep Learning is a type of Machine Learning that simulates human decision-making using neural networks.
* Machine Learning has many applications, from recommender systems that provide relevant choices for customers on commercial websites, to detailed analysis of financial markets.
* How to use regression to analyze data.
* [Module 5 - Data Science in Business](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@chapter+block@d00d8c98429946a8b782e5f1c6b6017d)
* Learn about what companies need to do in order to start with data science.
* Learn about some of the qualities that differentiate data scientists from other professionals.
* Learn about some applications of data science.
* Learn about analytics and what important role data scientists play in this process.
* Learn about story-telling and the importance of an effective final deliverable.
* Learn about the main components of an effective final deliverable.
* Apply what you learned about data science to answer open-ended questions.
* Demonstrate your understanding of the readings to define what data science and data scientist mean.
* Demonstrate your understanding of the readings to answer a question about the final deliverable of data science project.

## **How Data Science is Saving Lives (4:37)**

Using Data Science techniques to understand and analyze the large data sets available today has a huge impact on human lives. It can provide targeted information to help healthcare professionals give the best treatment to patients, or help predict natural disasters so that people can prepare early, and much more besides. In healthcare, data scientists use predictive analytics developed from data mining, data modeling, statistics, and machine learning to find the best options for patients. This type of predictive analytics examines all known factors for a disease, including gene markers, associated conditions, and environmental factors. It then recommends appropriate tests, suitable trials, and any suggested treatments. Every individual physician has their own store of knowledge gained from their studies, interests, and experiences.

Data science systems that use predictive analytics ensure that all physicians can also access the latest information about the disease, tests, and treatment plans, tailored to their specific patient. With this type of system, every physician has access to the same knowledge, and the best options can be consistently offered, improving patient outcomes. For example, a study by the Boston Consulting Group and AdvaMedDx, an industry association of medical diagnostics companies, examined the barriers to the adoption of potentially lifesaving diagnostic tests for patients with a specific cancer and a particular gene marker. The study discovered that the biggest factor in the patient being offered a specific test was the patient’s oncologist, who may or may not have known about the test and its relationship to the gene marker.

By providing extra information through data science tools, physicians can be made aware of the most helpful tests and treatments for a specific patient. There are many opportunities to explore other ways to mine data, such as from electronic medical records for different types of medical research. Schools such as the NorthShore University HealthSystem in suburban Chicago, a leader in the implementation of Electronic Medical Records (EMR) systems, now offer guidance on data mining. It is the first healthcare provider in America to be awarded the highest level of EMR deployment for both inpatient and outpatient care.

This remarkable effort has generated much-anonymized data available for innovative analytics research. Developing more sophisticated big data analytics capabilities helps healthcare organizations move from basic descriptive analytics towards predictive insights, thanks to data science. In the field of Disaster Preparedness, the ability to save lives using Data Science tools has been under development for many years. The use of predictive analytics tools is improving and providing new data analysis in a multitude of ways, alerting populations to danger faster than ever before. Large, high-quality data sets can be used to predict the occurrence of numerous types of natural disasters, which can be the difference between life and death for thousands of people. Earthquakes, hurricanes & tornados, floods, and volcanic eruptions can be predicted with the help of data science.

Recent research at the University of Warwick in the UK used social media content such as photos and keywords to track the development of floods, hurricanes and other weather events. When added to the information recorded by scientists and weather stations, this type of data can be used to improve the predictions for localised weather events. Because the real benefit of this knowledge is so important, schools are starting to include this type of data science education in their curriculum. For instance, the University of Chicago Graham School offers a Master of Science course in Threat and Response Management. Data science tools enable organizations to analyse vast quantities of data from widely different sources, and present that information in a way that allows data scientists to gain new knowledge, in some cases, saving hundreds of lives.

## **How Companies Should Get Started in Data Science (2:52)**

At the end of the day, for businesses, they know one thing, that if they are unable to measure something, they are unable to improve it. And if they are unable to measure their costs, they are unable to reduce them. If they're unable to measure their profits, they are unable to increase them. So the first thing a company has to do is to start recording information, start capturing data, data about costs. And the differentiate it by labor costs and material cost, the cost to how much it cost to sell one product and the total cost. And then you look at the revenue, where's your revenue coming from? Is 80% of your revenue coming from 20% of your customers? Or is it the other way around? So first thing first, start capturing data. Once you have data, then you can apply algorithms and analytics to it. So the first thing to do would be to capture data. If you're not capturing it, start capturing it. If you're capturing it, archive it. Do not overwrite on your old data thinking you don't need it anymore. Data never gets old. Data is always relevant, even if it's 100 years old, 200 years old. It is relevant to you and and your firm and your success. So keep data, capture it, archive it, make sure nothing goes to waste. Make sure there's a consistency. So someone 20 years later trying to understand that data should be able to do so, so have proper documentation. Do it now. Put the best practices for data archiving in place the moment you start a business. And if you're already in business and you haven't done it, do it now.

>> Start measuring things. Too many companies haven't measured things properly for a decade and, then they decide they want data science. Data science inside a company is only going to be as valuable as the data collected. Garbage in, garbage out is a rule in any sort of analysis.

>> If something is not measured, it's very difficult to improve it or to change it. So the very first step is measurement. If companies have existing data, then they should start looking at it and cleaning it. If they don't have existing data, then they need to start collecting it.

>> I think to look for a team who love to work as a data scientist.

>> The first stop is to have employees that they are interested on data science. because if you don't have interest in your company, you will not have engagement.

>> Companies should remember that it's key to have a team. So it's not one data scientist, but a team of them, that each of them have strengths in different areas of data science.

## **Applications of Data Science (3:38)**

Data Science and Big Data are making an undeniable impact on businesses, changing day-to-day operations, financial analytics, and especially interactions with customers. It’s clear that businesses can gain enormous value from the insights Data Science can provide, but sometimes it’s hard to see exactly how, so let’s look at some examples.

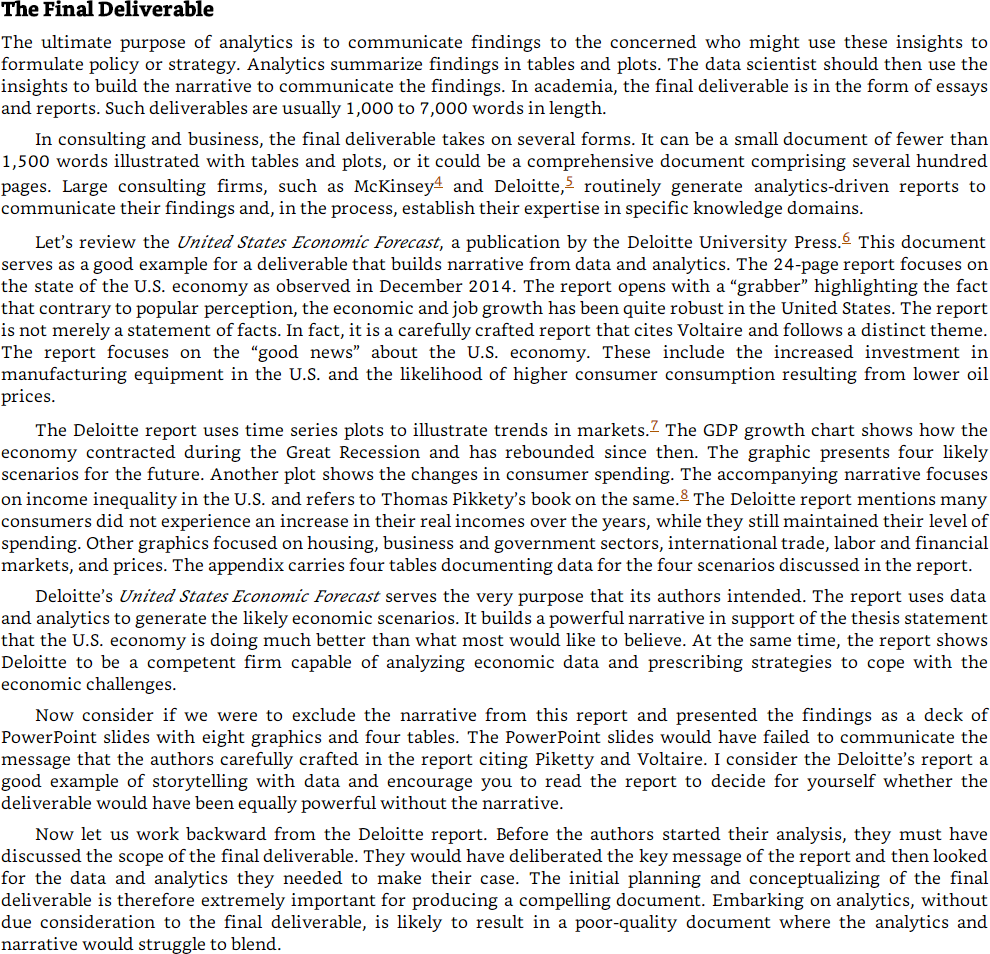
In this era of Big Data almost everyone generates masses of data every day, often without being aware of it. This digital trace reveals the patterns of our online lives. If you have ever searched for or bought a product on a site like Amazon, you will notice that it starts making recommendations related to your search. This type of system, known as a Recommendation Engine, is a common application of Data Science.

Companies like Amazon, Netflix, and Spotify use algorithms to make specific recommendations derived from customer preferences and historical behavior. Personal assistants like Siri on Apple devices use Data Science to devise answers to the infinite number of questions end users may ask. Google watches your every move in the world, your online shopping habits, and your social media. Then it analyzes that data to create recommendations for restaurants, bars, shops, and other attractions based on the data collected from your device and your current location. Wearable devices like Fitbits, Apple watches, and Android watches add information about your activity levels, sleep patterns, and heart rate to the data you generate. Now that we know how consumers generate data, let's take a look at how Data Science is impacting business.

In 2011, McKinsey and Company said that Data Science was going to become the key basis of competition, supporting new waves of productivity, growth, and innovation. In 2013, UPS announced that it was using data from customers, drivers, and vehicles in a new route guidance system aimed to save time, money, and fuel. Initiatives like this support the statement that Data Science will fundamentally change the way businesses compete and operate.

How does a firm gain a competitive advantage? Let’s take Netflix as an example. Netflix collects and analyzes massive amounts of data from millions of users, including which shows people are watching at what time of day, when people pause, rewind and fast-forward, and which shows, directors, and actors they search for. Netflix can be confident that a show will be a hit before filming even begins, by analyzing users’ preference for certain directors and acting talent and discovering which combinations people enjoy. Add this to the success of earlier versions of a show, and you have a hit. For example, Netflix knew many of its users had streamed the work of David Fincher. They also knew that films featuring Robin Wright had always done well, and that the British version of House of Cards was very successful. Netflix knew that significant numbers of people who liked Fincher also liked Wright. All this information combined to suggest that buying the series would be a good investment for the company. They were right, it was a huge hit. Thanks to Data Science, Netflix knows what people want before they do.

## **The Final Deliverable**



## **Module Summary**

* Data Science helps physicians provide the best treatment for their patients, and helps meteorologists predict the extent of local weather events, and can even help predict natural disasters like earthquakes and tornadoes.
* That companies can start on their data science journey by capturing data. Once they have data, they can begin analysing it.
* Some ways that data is generated by consumers.
* How businesses like Netflix, Amazon, UPs, Google, and Apple use the data generated by their consumers and employees.
* The purpose of the final deliverable of a Data Science project is to communicate new information and insights from the data analysis to key decision-makers.
* [Module 6 - Careers and Recruiting in Data Science](https://courses.edx.org/courses/course-v1:IBM+DS0101EN+3T2020/course/#block-v1:IBM+DS0101EN+3T2020+type@chapter+block@c288100f86f8465ba37376237ffbd7f7)
* Learn about what companies need to do in order to start with data science.
* Learn about some of the qualities that differentiate data scientists from other professionals.
* Learn about some applications of data science.
* Learn about analytics and what important role data scientists play in this process.
* Learn about story-telling and the importance of an effective final deliverable.
* Learn about the main components of an effective final deliverable.
* Apply what you learned about data science to answer open-ended questions.
* Demonstrate your understanding of the readings to define what data science and data scientist mean.
* Demonstrate your understanding of the readings to answer a question about the final deliverable of data science project.

## **How Can Someone Become a Data Scientist (5:19)**

A real data scientist, the high-end data scientists, are mostly PhDs. They often come out of physics, out of statistics, they have to have a computer science background, they have to have a math background, they have to know about databases and statistics and probability and all that stuff. However, if you're coming into a data science team, I think the first skills you need is you need to know how to program, at least have some computational thinking, so having taken a programing course, you need to know some algebra, at least up to analytics, geometry, and hopefully some calculus, some basic probability, some basic statistics, I mean really have to understand the difference and different statistical distributions, and database. I mean, one of the easiest places to start is relational databases, which stores lots and lots of our data so people can first walk before they can run by at least understanding about computers and databases and how we store things and if you understand relational databases nowadays you can still, just with that understanding, use big data clusters as if they were just a big relational database. You don't have to really have understand the whole MapReduce programming model.

But then, as you go further up in the field, then you have to know a lot of computer science theory and statistics, it's really, and probability, it's really the intersection of them that the high end data scientists, the PhD data scientists work with. I do a lot of self-learning. I think everybody these days, I mean, I learned about Hadoop all by myself, I read some articles, I watched some videos, I thought, I played, although I'm a builder, I'm a tinkerer, so if I wanna figure out how to do something, I build it. I mean, my first HPC cluster I heard about this term a Beowulf cluster, I mean, yeah, what the hell's that? So I looked it up and said, oh, it's just a bunch of computers hooked together with a TCP/IP network, that's pretty easy, so we get a grant from Citi Bank and we built a five thing cluster and I said, oh, well, that's HPC. I said, I had one of the first HPC clusters at the university, it was tiny but a lot of our researchers loved it because they could run stuff 40 and 50 times faster. So I think one of the ways you learn things is you do them, you have to do them, and these online learning platforms especially now that we have things like Ipython and Jupyter Notebooks and I guess Zeppelin means that you can actually go in and take some of these courses and you can do things right then and you can see them and feel them and play with them and, at that point, you know, you'll start to get your head around what is actually happening. Motivation is the key problem in all of these, is how to keep people motivated and I think the badge system that the, what was it, Big Data University has, is one of the ways is how do you get people to keep going through. But if they want to, they can. It's up to the individual to. So they have to understand what the goal is.

The place it can't sit is probably under the CIO, the Chief Information Officer. CIOs current chief information officers in many companies got there from an accounting background or a finance background, they're clueless. Sorry. But they really, it has to come out of the research side. So you'll find data scientists primarily in companies that have some research agenda, pharmaceuticals, finance, all of, any technology company. If you look at, we can't keep some of our PhD data scientists in our program, they are now at Facebook, they're at Linkedin, they're at Uber, they're at Lyft, because the demand out there for the PhD level data scientist is just unbelievable. They make large amounts of money and they're playing with problems that are really, really neat. How do you schedule the Uber cars? You have enormous amounts of data.

## **Recruiting for Data Science (7:32)**

When the companies are hiring people for a data science team, maybe a data scientist or an analyst, or a chief data scientist, the tendency would be to find the person who has all the skills, that they know the domain-specific knowledge. They're excellent in analyzing structured and unstructured data. And they're great at presenting and they've got great storytelling skills. So if you put all this together, you will realize you're looking for a unicorn. And your odds of finding a unicorn are pretty rare. I think what you need to do to is to see, given the pool of applicants you have, who has the most resonance with your firm's DNA. Because you can teach analytics skills, anyone can learn analytics skills if they dedicate time and effort to it. But what really matters is who's passionate about the kind of business that you do. Someone could be a great data scientist in the retail environment, but they may not be that excited about working in IT related firms or working with gigabytes of weblogs. But if someone is excited about those weblogs, if someone is excited about health-related data then they would be able to contribute to your productivity much more so. And I would say if I'm looking for someone, if I have to put together a data science team, I would first look for curiosity. Is that person curious about things not just for data science but anything like, are they curious about why this room is painted a certain way, why do the bookshelves have books, and what kinds of books? They have to have a certain degree of curiosity about everything that is in their vision, that they look at.

The second thing is do they have a sense of humor because, you see, you have to have a lighthearted about it. If someone is too serious about it, they probably would take it too seriously, and would not be able to look at the lighter elements. The third thing I think, and I think the last thing that I would look for if I had to have a hierarchy, the last thing I would look for are technical skills. I would go through the social skills, curiosity, and sense of humor. The ability to tell a story. The ability to know that there is a story there. And then once all is there then I would say, well, can you do the technical side of it? And if there is some hope or some sign of some technical skills, I would take them because I can train them in whatever skills they need. But I cannot teach curiosity. I cannot teach storytelling. I cannot certainly, instill sense of humor in anyone.

>> I think there's no hard and fast rule for hiring data scientists. I think it's going to be a case by case thing. I would say there has to be some sort of technical component, somebody should be able to work with and manipulate the data. They should be able to communicate what they find in the data. I find quite often nobody really cares about the r-square or the confidence interval.

So you have to be able to introduce those things and explain something in a compelling way. And they also have to find somebody who is relatable, because data science, it been typically new means that the person in that role has to make relationships and they have to work across different departments.

>> If these data scientist has a good mathematics and statistics background.>> They have to consider like problem solving abilities and analysis. The scientist needs to be good in analyzing problems.>> The persons they are hiring, they should love to play with data. And then they know how to play with the data visualization. They have analytical thinking.

>> When a company is hiring anyone to work on a data science team, they need to think about what role that person is going to take. Before a company begins, they need to understand what they want out of their data science team. And then they need to hire to begin it. As they grow a data science team, they need to understand whether they need engineers, architects, designers to work on visualization. Or whether they just need more people who can multiply large matrices.

>> From a skills point of view, let's focus on the technical skills and in that case, first thing would be what kind of a technical platform would you like to adopt? Let's say you want to work in a structured data environment and let's say you want to work in market research. Then the type of skills you need are slightly different than someone who would like to work in big data environments. If you want to work in the traditional market research data, structure data environment, your skills should be some statistical knowledge and some knowledge of basic statistical algorithms, maybe some machine learning algorithms. And these are the tools that you would like to develop. If you want to work in big data, then there's the other aspect of it and that is to be able to store data. So you start with the expertise in storing large amounts of data. And then you look into platforms that allow you to do that. The next step would be to be able to manipulate large amounts of data, and the final step would be to apply algorithms to those large sets of data. So it's a three-step process. But most likely it starts, most importantly, it starts with where you would like to be, in what field, in what domain. In terms of platforms, let's you want to be in the traditional predictive analytics environment, and you're not working with big data, then R or Stata, or Python would be your tools. If you're working mostly with unstructured data, then Python is most suitable than R. If you're working with big data, then Hadoop and Spark are the environments that you will be working with. So it all depends upon where you would like to be and what kind of work excites you and then you pick your tools. In addition to technical skills, the second aspect of the data science is to have the ability to communicate. The communication skills or presentation skills. I call them story telling skills, that is that you have your analysis done, now can you tell a great story from it? If you have a very large table, can you synthesize this and make it more appealing that when it goes on the screen, or is it part of a document that it just speaks? It sings the findings and the reader just gets it right there. So the ability to present your findings, either verbally, or in a presentation, or in a document. So those communication and presentation skills are equally important as the technical skills are. When you have a grading side, when you're presenting your results, imagine you're driving on a mountain and then there's a sharp turn. And you can't see what's beyond the turn. And then you make that turn and then suddenly, you see a tremendous valley in front of you. And this great sense of awe, that I didn't know that, right? So when you present your findings and you have this great finding and you communicate it well, this is what people feel because they were not expecting it. They were not aware of it, and then this great sense of happiness that now I know. And I didn't know this, now I know. And then it empowers them, it gives them ideas what they can do with this knowledge, this new insight. It's a great sense of joy. And you are able as a data scientist, you are able to share with your clients because you enabled it.

## **Careers in Data Science (2:51)**

The emergence of the Internet of Things and advances in distributed computing have brought vast amounts of data and the technological capability to analyze it. Now that we can extract useful insights and new knowledge, we need to know how to shape that data to focus on what to do with it and what it can do for us. Enter Data Science. Companies like LinkedIn, Glassdoor, Indeed, and Dice track employment trends which show a career in Data Science moving up the list of most promising jobs to become #1 since 2016. It remains one of the top three career choices for 2020. Dice noted that job postings are from companies in a wide variety of industries, not just tech. Global Industry Analysts, Inc, predicts that the Data Science Platform market will grow by 314.8 Billion US dollars by 2025, driven by a compounded growth of 38.2%.

McKinsey Global Institute warned of “huge talent shortages for data and analytics” by 2018. Forrester Research analyst Brandon Purcell said, in January of 2019, the demand for data scientists will only grow, as organizations increasingly rely on data-driven insights. We’re now well into that period, and recruiters are finding it difficult to fill the growing need for talented data scientists. What motivates someone going into a Data Science Career? For one thing, data science applies to almost any discipline. So, if you have the aptitude and desire to work with data, enjoy coding, have no problem learning math and statistics, and you are a good storyteller, then you can certainly enter a data science field and excel. For most people, this means acquiring some additional tools and skills and continuously learning about new tools and techniques in the field. The Women in Data Science initiative, spearheaded by the Stanford Institute for Computational and Mathematical Computing have committed to “inspire and educate data scientists worldwide, regardless of gender, and to support women in the field.”

When you are seeking a career in Data Science, you need to make sure your skill set matches the role you are targeting. You can tailor your skill set to the specific area you want to enter, adding missing skills via one of the many excellent online training resources. Then you will be prepared for a fascinating and rewarding career. So now is the time to move into this field when there are such diverse choices available **and education resources that make it a reality.**

## **High School Students and Data Science Careers (4:52)**

Learn how to program. Learn some math. Take a course in probability. Learn a little bit of statistics. And then, play. Build something, write something. I mean, when I say build, programming and building systems, building things isn't just physical, right?

You can build computer systems, statistical systems, whatever. But once you try to do something, then you'll know what tools you need, right? And you'll say, "Oh, oh my god, what? "There's this expression there, "what does an inner product mean? "What's that? "How do I, oh, okay, I can learn that." And then when they get to college, they will have a big jump on many of the other college students. And so when they get out of college, they'll have an even bigger jump, and then make a lot of money. And they'll be happy, too. This stuff is fun, right? It's fun.

If you're in high school and you're considering data science, I would say get familiar with data bases, start learning SQL, start thinking about, you know, computer science, if that's interesting. If you have a computer science course in your school, you know, take it, and that's a good part of being a data scientist. Beyond that there are probably ways to foster your creativity, right, your curiosity. If you like detective games, that's kind of cool, right. And if you like treasure hunts or whatever, right, if you're into that stuff, I think you'll, and you get the opportunity to do that stuff, that will help you as a data scientist because it's a really a good way to kind of make sure that you can be curious as you go about your daily life.

Encourage the curiosity, encourage the experimentation. It's kind of like science fairs, science fairs are great, they encourage that experimentation, that learning from, asking a question and answering it through a scientific method, but doing that with data sets rather than vinegar volcanoes. It's kind of the same thing, but learning from data and we're going through an election season right now, there's a lot of stuff in the news about polls and survey results and that's a great way to start a conversation and talk about how do the people who ran the polls, how do they know, how can they predict what's going to happen in the election. So that's another cool way to start a conversation about data science. I would say encourage the person who is interested in data science because to pursue to, because it's a great career and it is something that is definitely going to be in need in the future. It's one of those highly sought after knowledge professions that are really important to businesses around the world.

So being a data scientist and being able to help companies as they grow and learn how do to things more efficiently or how to do things smarter, there will always be a need for people like that. And data scientists are those people. I would say that I understand what you're talking about because I was never a great mathematics student as well. And I think there's actually a bunch of data scientists who are really successful and popular who are in the same boat. You know there's kind of arithmetic and math in school is not necessarily everybody's best subject. But when you combine it with, you know these aren't just hypothetical numbers, these aren't just, problem statements that you have no connection to. When you have a connection to the problem, it suddenly becomes much easier to use math to help understand it, I found. And so you know, knowing the people who will benefit from the math that you do I think is really cool.

## **The Report Structure**



## **Module Summary**

* The length and content of the final report will vary depending on the needs of the project.
* The structure of the final report for a Data Science project should include a cover page, table of contents, executive summary, detailed contents, acknowledgments, references, and appendices.
* The report should present a thorough analysis of the data and communicate the project findings.

## **Introduction to Tools for Data Science Course**

### Introduction to Tools for Data Science (3:14)

Welcome to the course! You've begun one of the most complete overviews on data science tooling that you’ll currently find on the internet. This doesn’t mean that we cover each and every tool, but later in the course we’ll introduce a comprehensive list of tasks a data scientist needs to perform and give you the top two or three open source and commercial tools available to complete them. We also explain how the tools overlap in functionality, what their pros and cons are, and how these tools can address the whole data science pipeline.

Let’s start with data. Data is obviously central to data scientists. In this course, we’ll show you how to manage, extract, transform, analyze, and visualize data. Now, you might be able to survive data science without programming skills if you use the right set of tools. However, we highly recommend getting familiar with programming and the related programming frameworks for data science. To help you along, we’ll introduce you to the most commonly used programming languages and frameworks available for data science. That said, there is a lot of automation available in the latest tooling that a data scientist can use.

In this course, we’ll explain how to make use of those tools to save time and uncover inspiration. Visual programming is available in many tools. In this course, you’ll learn how visual programming can be used to speed up development time and to help non-programmers enter the field of data science. Open source software is leading the field of data science, but its total costs of ownership, or "TCO," can be higher at times due to configuration, customization and maintenance costs. As a result, commercial software also has its place, especially since the new generation of commercial data science software leverages open source software and open standards. This makes it easy to migrate between tools and can reduce overall TCO.

In this course, we’ll introduce you to both open source and commercial software and point out their strengths and weaknesses for data science. We'll also show you ways that you can take advantage of their respective strengths. Finally, we'll show you how cloud computing can be used to further speed up and facilitate data scientists' work. We'll introduce you to the most commonly used and newly emerging cloud tools for data science. In addition to lectures, this course, has numerous labs to make you more familiar with the material and get hands-on experience. There are also multiple quizzes to test your learning. Nothing more to say than we’re glad to have you in the course and happy learning. In case you have trouble in any way, please don’t hesitate to contact us in the discussion forum. There's nothing left but to begin! We're very happy to have you with us as you start your data science journey. If you have any trouble with any of the course material, please don’t hesitate to contact us in the discussion forum.

## **Languages of Data Science (2:18)**

The Languages of Data Science.

For anyone just getting started on their data science journey, the range of technical options can be overwhelming. There is a dizzying amount of choice when it comes to programming languages. Each has it's own strengths and weaknesses and there is no one right answer to the question of which one you should learn first. The answer to that question depends largely on your needs, the problems you are trying to solve, and who you are solving them for.

Python, R, and SQL are the languages that we recommend you consider first and foremost. But there are so many others that have their own strengths and features. Scala, Java, C++, and Julia are some of the most popular. Javascript, PHP, Go, Ruby, and Visual Basic all have their own unique use cases as well. The language you choose to learn will depend on the things you need to accomplish and the problems you need to solve. It will also depend on what company you work for, what role you have, and the age of your existing application. We’ll explore the answers to this question as we dive into the popular languages in the data science industry.

There are many roles available for people who are interested in getting involved in data science. Business Analyst Database Engineer Data Analyst Data Engineer Data Scientist Research Scientist Software Engineer Statistician Product Manager Project Manager and many more. Let’s dive into what we will learn in Lesson 1. We will put most of our focus on the top three Data Science languages: Python, R, and SQL, which each have their own lessons. Then we will go on to highlight other noteworthy languages and what makes them special. Then we’ll finish with a short quiz to test your knowledge! Here's what we'll cover: Python R SQL Other Noteworthy Data Science Languages and Practice Quiz.

## **Introduction to Python (3:50)**

In this video, we will review the high-level features of the Python programming language. Python is a powerhouse language. It is by far the most popular programming language for data science. According to the 2019 Kaggle Data Science and Machine Learning Survey, 75% of the over 10,000 respondents from around the world reported that they use Python on a regular basis. Glassdoor reported that in 2019 more than 75% of data science positions listed included Python in their job descriptions. When asked which language an aspiring data scientist should learn first, most data scientists say Python. You are probably thinking, why on earth is Python so popular? Well, let’s start with the people who use Python. If you already know how to program, then Python is great for you because it uses clear, readable syntax. You can do many of the things you are used to doing in other programming languages but with Python you can do it with less code. If you want to learn to program, it’s also a great starter language because of the huge global community and wealth of documentation. In fact, several different surveys in 2019 found that over 80% of data professionals worldwide use Python. Python is useful for many situations, including data science, AI and machine learning, web development, and IoT devices like the Raspberry Pi. Large organizations that use Python heavily include IBM, Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify, and Reddit.

Python is a powerful general-purpose programming language that can do a lot of things. It is widely supported by a global community and shepherded by the Python Software Foundation. Python is a high-level general-purpose programming language that can be applied to many different classes of problems. It has a large, standard library that provides tools suited to many different tasks, including but not limited to databases, automation, web scraping, text processing, image processing, machine learning, and data analytics. For data science, you can use Python's scientific computing libraries such as Pandas, NumPy, SciPy, and Matplotlib.

For artificial intelligence, it has TensorFlow, PyTorch, Keras, and Scikit-learn. Python can also be used for Natural Language Processing (NLP) using the Natural Language Toolkit (NLTK). Another great selling point is the Python community, which has a well documented history of paving the way for diversity and inclusion efforts in the tech industry as a whole. The Python language has a code of conduct executed by the Python Software Foundation that seeks to ensure safety and inclusion for all, in both online and in person python communities. There are also communities like PyLadies that seek to create spaces for people interested in Python to learn in safe and inclusive environments. PyLadies is an international mentorship group with a focus on helping more women become active participants and leaders in the Python open source community.

## **Introduction to R Language (3:48)**

## In this video, we will give a brief overview of the R programming language. After our last video on Python, where we discussed its wide adoption, you might be wondering why on earth you should consider learning any other language. Well, according to the results of the 2019 Kaggle Data Science survey, which had over 10k respondents from around the world, learning up to three languages can increase your salary! And R has a lot to offer you. Like Python, R is free to use, but it's a GNU project -- instead of being open source, it's actually free software. So if Python is open source and R is free software, what’s the difference? Well, Both open source and free software commonly refer to the same set of licenses. Many open source projects use the GNU General Public License, for example. Both open source and free software support collaboration. In many cases (but not all), these terms can be used interchangeably.

The Open Source Initiative (OSI) champions open source while the Free Software Foundation (FSF) defines free software. Open source is more business focused, while free software is more focused on a set of values. Back to why you should learn R. Because this is a free software project, you can use the language in the same way that you contribute to open source, and it allows for public collaboration and private and commercial use. Plus, R is another language supported by a wide global community of people passionate about making it possible to use the language to solve big problems.

Who is R for? It's most often used by statisticians, mathematicians, and data miners for developing statistical software, graphing, and data analysis. The language’s array-oriented syntax makes it easier to translate from math to code, especially for someone with no or minimal programming background. According to Kaggle’s Data Science and Machine Learning Survey, most folks learn R when they're a few years into their data science career, but it remains a welcoming language to those who don’t have a software programming background. R is popular in academia but companies that use R include IBM, Google, Facebook, Microsoft, Bank of America, Ford, TechCrunch, Uber, and Trulia.

● R has become the world’s largest repository of statistical knowledge.

● As of 2018, R has more than 15,000 publicly released packages, making it possible to conduct complex exploratory data analysis.

● R integrates well with other computer languages, such as C++, Java, C, .Net, and Python.

● Common mathematical operations such as matrix multiplication work straight out of the box.

● R has stronger object-oriented programming facilities than most statistical computing languages. There are many ways to connect with other R users around the globe. Communities such as user!, WhyR?, SatRdays, and R-Ladies are all great to connect with. And you can also check out the R project website for R conferences and events.

## **Introduction to SQL (3:35)**

In this video, we'll take a high-level look at SQL. SQL is a bit different from the other languages we’ve covered so far. First off, it's formally pronounced “ess cue el,” although some people say “sequel.” While the acronym stands for “Structured Query Language,” many people do not consider SQL to be like other software development languages because it's a non-procedural language and its scope is limited to querying and managing data. While it is not a “data science” language per se, data scientists regularly use it because it's simple and powerful!

Another couple of neat facts about SQL: it's much older than Python and R, by about 20 years, having first appeared in 1974. And, SQL was developed at IBM! This language is useful in handling structured data; that is, the data incorporating relations among entities and variables. SQL was designed for managing data in relational databases. Here you can see a diagram showing the general structure of a relational database. A relational database is formed by collections of two-dimensional tables; for example, datasets and Microsoft Excel spreadsheets. Each of these tables is then formed by a fixed number of columns and any number of rows.

BUT! Even though SQL was originally developed for use with relational databases, because it's so pervasive and easy to use, SQL interfaces for many NoSQL and big data repositories have also been developed. The SQL language is subdivided into several language elements, including clauses, expressions, predicates, queries, and statements. So what makes SQL great? Knowing SQL will help you do many different jobs in data science, including business and data analyst, and it's a must in data engineering. When performing operations with SQL, you access the data directly. There's no need to copy it beforehand. This can speed up workflow executions considerably. SQL is the interpreter between you and the database. SQL is an American National Standards Institute, or "ANSI," standard, which means if you learn SQL and use it with one database, you will be able to easily apply that SQL knowledge to many other databases.

There are many different SQL databases available, including MySQL, IBM Db2, PostgreSQL, Apache OpenOffice Base, SQLite, Oracle, MariaDB, Microsoft SQL Server, and more. The syntax of the SQL you write might change a little bit based on the relational database management system you’re using. If you are looking to learn SQL you would be best served to focus on a specific relational database and then plug into the community for that specific platform. There are also many great introductory courses on SQL available!

## **Other Languages (6:35)**

So far, we’ve reviewed Python, R, and SQL. In this video, we will review some other languages that have compelling use cases for data science. Ok, so indisputably, Python, R, and SQL are the three most popular languages that data scientists use. But, there are many, many other languages that are worth your time when considering which language to use to solve a particular data science problem. Scala, Java, C++, and Julia are probably the most traditional data science languages on this slide. But JavaScript, PHP, Go, Ruby, Visual Basic, and others have all found their place in the data science community as well!

I won’t dive as deeply into each of these languages, but I'll mention some notable highlights. Java is a tried-and-true general-purpose object oriented programming language. It's been widely adopted in the enterprise space and is designed to be fast and scalable. Java applications are compiled to bytecode and run on the Java Virtual Machine, or "JVM." Some notable data science tools built with Java include Weka, for data mining; Java-ML, which is a machine learning library; Apache MLlib, which makes machine learning scalable; and Deeplearning4j, for deep learning.

Apache Hadoop is another Java-built application. It manages data processing and storage for big data applications running in clustered systems.

Scala is a general-purpose programming language that provides support for functional programming and a strong static type system. Many of the design decisions in the construction of the Scala language were made to address criticisms of Java. Scala is also interoperable with Java, as it runs on the JVM. The name "Scala" is a combination of "scalable" and "language." This language is designed to grow along with the demands of its users.

For data science, the most popular program built using Scala is Apache Spark. Spark is a fast and general-purpose cluster computing system. It provides APIs that make parallel jobs easy to write, and an optimized engine that supports general computation graphs. Spark includes Shark, which is a query engine; MLlib, for machine learning; GraphX, for graph processing; and Spark Streaming. Apache Spark was designed to be faster than Hadoop.

C++ is a general-purpose programming language. It is an extension of the C programming language, or "C with Classes.” C++ improves processing speed, enables system programming, and provides broader control over the software application. Many organizations that use Python or other high-level languages for data analysis and exploratory tasks still rely on C++ to develop programs that feed that data to customers in real-time.

For data science, a popular deep learning library for dataflow called TensorFlow was built with C++. But while C++ is the foundation of TensorFlow, it runs on a Python interface, so you don’t need to know C++ to use it. MongoDB, a NoSQL database for big data management, was built with C++. Caffe is a deep learning algorithm repository built with C++, with Python and MATLAB bindings. A core technology for the World Wide Web, JavaScript is a general-purpose language that extended beyond the browser with the creation of Node.js and other server-side approaches.

Javascript is NOT related to the Java language. For data science, the most popular implementation is undoubtedly TensorFlow.js. TensorFlow.js makes machine learning and deep learning possible in Node.js as well as in the browser.

TensorFlow.js was also adopted by other open source libraries, including brain.js and machinelearn.js. The R-js project is another great implementation of JavaScript for data science.

R-js has re-written linear algebra specifications from the R Language into Typescript. This re-write will provide a foundation for other projects to implement more powerful math base frameworks like Numpy and SciPy of Python.

Typescript is a superset of JavaScript. Julia was designed at MIT for high-performance numerical analysis and computational science. It provides speedy development like Python or R, while producing programs that run as fast as C or Fortran programs. Julia is compiled, which means that the code is executed directly on the processor as executable code; it calls C, Go, Java, MATLAB, R, Fortran, and Python libraries; and has refined parallelism. The Julia language is relatively new, having been written in 2012, but it has a lot of promise for future impact on the data science industry. JuliaDB is a particularly useful application of Julia for data science. It's a package for working with large persistent data sets. That's as far as we’ll dig into the many languages that are used to solve data science problems.

If you have experience with a particular language, I recommend you do a web search to see what might already be possible in terms of using it for data science. You might be surprised at the possibilities!

## **Categories of Data Science Tools (2:27)**

**Open source tools are available for various data science tasks.** In this video, we’ll have a look at the different data science tasks. In subsequent videos we’ll walk through the most commonly used open source tools for those tasks. The most important tools are covered throughout this course. Data Management is the process of persisting and retrieving data. Data Integration and Transformation, often referred to as Extract, Transform, and Load, or “ETL,” is the process of retrieving data from remote data management systems. Transforming data and loading it into a local data management system is also part of Data Integration and Transformation. Data Visualization is part of an initial data exploration process, as well as being part of a final deliverable. Model Building is the process of creating a machine learning or deep learning model using an appropriate algorithm with a lot of data. Model deployment makes such a machine learning or deep learning model available to third-party applications.

Model monitoring and assessment ensures continuous performance quality checks on the deployed models. These checks are for accuracy, fairness, and adversarial robustness. Code asset management uses versioning and other collaborative features to facilitate teamwork. Data asset management brings the same versioning and collaborative components to data. Data asset management also supports replication, backup, and access right management. Development environments, commonly known as Integrated Development Environments, or “IDEs”, are tools that help the data scientist to implement, execute, test, and deploy their work.

Execution environments are tools where data preprocessing, model training, and deployment take place. Finally, there is fully integrated, visual tooling available that covers all the previous tooling components, either partially or completely. This concludes this video. In the next video we’ll start looking at open source tools for data science tasks.

## **Open Source Tools for Data Science - Part 1 (7:20)**

In part one of this two-part series, we’ll cover data management, open source data integration, transformation, and visualization tools. The most widely used open source data management tools are relational databases such as **MySQL and PostgreSQL; NoSQL databases such as MongoDB Apache CouchDB, and Apache Cassandra;** and file-based tools such as the Hadoop File System or Cloud File systems like Ceph. Finally,Elasticsearch is mainly used for storing text data and creating a search index for fast document retrieval. The task of data integration and transformation in the classic data warehousing world is called ETL, which stands for “extract, transform, and load.” These days, data scientists often propose the term “ELT” – Extract, Load, Transform“ELT”, stressing the fact that data is dumped somewhere and the data engineer or data scientist themselves is responsible for data.

Another term for this process has now emerged: “data refinery and cleansing.” Here are the most widely used open source data integration and transformation tools: Apache AirFlow, originally created by AirBNB; KubeFlow, which enables you to execute data science pipelines on top of Kubernetes; Apache Kafka, which originated from LinkedIn; Apache Nifi, which delivers a very nice visual editor; Apache SparkSQL (which enables you to use ANSI SQL and scales up to compute clusters of 1000s of nodes), and NodeRED, which also provides a visual editor. NodeRED consumes so little in resources that it even runs on small devices like a Raspberry Pi.

We’ll now introduce the most widely used open source data visualization tools. We have to distinguish between programming libraries where you need to use code and tools that contain a user interface. The most popular libraries are covered in the next videos. A similar approach uses Hue, which can create visualizations from SQL queries. Kibana, a data exploration and visualization web application, is limited to Elasticsearch (the data provider). Finally, Apache Superset is a data exploration and visualization web application. Model deployment is extremely important. Once you’ve created a machine learning model capable of predicting some key aspects of the future, you should make that model consumable by other developers and turn it into an API. Apache PredictionIO currently only supports Apache Spark ML models for deployment, but support for all sorts of other libraries is on the roadmap.

Seldon is an interesting product since it supports nearly every framework, including TensorFlow, Apache SparkML, R, and scikit-learn. Seldon can run on top of Kubernetes and Redhat OpenShift. Another way to deploy SparkML models is by using MLeap.

Finally, TensorFlow can serve any of its models using the TensorFlow service. You can deploy to an embedded device like a Raspberry Pi or a smartphone using TensorFlow Lite, and even deploy to a web browser using TensorFlow dot JS. Model monitoring is another crucial step.

Once you’ve deployed a machine learning model, you need to keep track of its prediction performance as new data arrives in order to maintain outdated models. Following are some examples of model monitoring tools:

ModelDB is a machine model metadatabase where information about the models are stored and can be queried. It natively supports Apache Spark ML Pipelines and scikit-learn.

A generic, multi-purpose tool called Prometheus is also widely used for machine learning model monitoring, although it’s not specifically made for this purpose. Model performance is not exclusively measured through accuracy. Model bias against protected groups like gender or race is also important. The IBM AI Fairness 360 open source toolkit does exactly this. It detects and mitigates against bias in machine learning models. Machine learning models, especially neural-network-based deep learning models, can be subject to adversarial attacks, where an attacker tries to fool the model with manipulated data or by manipulating the model itself.

The IBM Adversarial Robustness 360 Toolbox can be used to detect vulnerability to adversarial attacks and help make the model more robust. Machine learning modes are often considered to be a black box that applies some mysterious “magic.” The IBM AI Explainability 360 Toolkit makes the machine learning process more understandable by finding similar examples within a dataset that can be presented to a user for manual comparison. The IBM AI Explainability 360 Toolkit can also illustrate training for a simpler machine learning model by explaining how different input variables affect the final decision of the model.

Options for code asset management tools have been greatly simplified: For code asset management – also referred to as version management or version control – Git is now the standard. Multiple services have emerged to support Git, with the most prominent being GitHub, which provides hosting for software development version management. The runner-up is definitely GitLab, which has the advantage of being a fully open source platform that you can host and manage yourself.

Another choice is Bitbucket. Data asset management, also known as data governance or data lineage, is another crucial part of enterprise grade data science. Data has to be versioned and annotated with metadata. Apache Atlas is a tool that supports this task. Another interesting project, ODPi Egeria, is managed through the Linux Foundation and is an open ecosystem. It offers a set of open APIs, types, and interchange protocols that metadata repositories use to share and exchange data. Finally, Kylo is an open source data lake management software platform that provides extensive support for a wide range of data asset management tasks. This concludes part one of this two-part series.

## **Open Source Tools for Data Science - Part 2 (5:15)**

In this video we’ll cover the development environment, execution environment, and fully integrated and visual tooling. One of the most popular current development environments that data scientists are using is “Jupyter.” Jupyter first emerged as a tool for interactive Python programming; it now supports more than a hundred different programming languages through “kernels.” Kernels shouldn’t be confused with operating system kernels. Jupyter kernels are encapsulating the different interactive interpreters for the different programming languages. A key property of Jupyter Notebooks is the ability to unify documentation, code, output from the code, shell commands, and visualizations into a single document.

JupyterLab is the next generation of Jupyter Notebooks and in the long term, will actually replace Jupyter Notebooks. The architectural changes being introduced in JupyterLab makes Jupyter more modern and modular. From a user’s perspective, the main difference introduced by JupyterLab is the ability to open different types of files, including Jupyter Notebooks, data, and terminals. You can then arrange these files on the canvas. Although Apache Zeppelin has been fully reimplemented, it’s inspired by Jupyter Notebooks and provides a similar experience. One key differentiator is the integrated plotting capability.

In Jupyter Notebooks, you are required to use external libraries; in Apache Zeppelin, and plotting doesn’t require coding. You can also extend these capabilities by using additional libraries. RStudio is one of the oldest development environments for statistics and data science, having been introduced in 2011. It exclusively runs R and all associated R libraries. However, Python development is possible and R is therefore tightly integrated into this tool to provide an optimal user experience. RStudio unifies programming, execution, debugging, remote data access, data exploration, and visualization into a single tool. Spyder tries to mimic the behavior of RStudio to bring its functionality to the Python world. Although Spyder does not have the same level of functionality as RStudio, data scientists do consider it an alternative, but in the Python world, Jupyter is used more frequently. This diagram shows how Spyder integrates code, documentation, visualizations, and other components into a single canvas.

Sometimes your data doesn’t fit into a single computer’s storage or main memory capacity. That’s where cluster execution environments come in. The well-known cluster-computing framework Apache Spark is among the most active Apache projects and is used across all industries, including in many Fortune 500 companies. The key property of Apache Spark is linear scalability. This means, if you double the number of servers in a cluster, you’ll also roughly double its performance.

After Apache Spark began to gain market share, Apache Flink was created. The key difference between Apache Spark and Apache Flink is that Apache Spark is a batch data processing engine, capable of processing huge amounts of data file by file. Apache Flink, on the other hand, is a stream processing image, with its main focus on processing real-time data streams. Although engine supports both data processing paradigms, Apache Spark is usually the choice in most use cases. One of the latest developments in the data science execution environments is called “Ray,” which has a clear focus on large-scale deep learning model training. Let’s look at open source tools for data scientists that are fully integrated and visual. With these tools, no programming knowledge is necessary. Most important tasks are supported by these tools; with tasks including data integration, transformation, data visualization, and model building.

KNIME originated at the University of Konstanz in 2004. As you can see, KNIME has a visual user interface with drag-and-drop capabilities. It also has built-in visualization capabilities. Knime can be extended by programming in R and Python, and has connectors to Apache Spark. Another example of this group of tools is Orange. It’s less flexible than KNIME, but easier to use.

In this section, you’ve learned about the most common data science tasks and which open source tools are relevant to those tasks. In the next video, we’ll describe some established commercial tools that you’ll encounter in your data science experience.

## **Commercial Tools for Data Science (5:59)**

We previously covered open source tools for data science. Now, let’s look at the commercial options you’ll find in many enterprise projects. Let’s revisit our overview of different tool categories. In data management, most of an enterprise’s relevant data is stored in an Oracle Database, Microsoft SQL Server, or IBM Db2. Although open source databases are gaining popularity, those three data management products are still considered the industry-standard. They won’t disappear in the near future. It’s not just about functionality.

Data is at the heart of every organization, and the availability of commercial supports plays a major role. Commercial supports are delivered directly from software vendors, influential partners, and support networks. When we focus on commercial data integration tools, we’re talking about “extract, transform, and load,” or “ETL” tools.

According to a Gartner Magic Quadrant, Informatica Powercenter and IBM InfoSphere DataStage are the leaders, followed by products from SAP, Oracle, SAS, Talend, and Microsoft. These tools support design and deployment of ETL data-processing pipelines through a graphical interface. They also provide connectors to most of the commercial and open source target information systems.

Finally, Watson Studio Desktop includes a component called Data Refinery, which enables the defining and execution of data integration processes in a spreadsheet style. In the commercial environment, data visualizations are utilizing business intelligence, or “BI”, tools. Their main focus is to create visually attractive and easy-to-understand reports and live dashboards. The most prominent commercial examples are: Tableau, Microsoft Power BI, and IBM Cognos Analytics.

Another type of visualization targets data scientists rather than regular users. A sample problem might be “How can different columns in a table relate to each other?” This type of functionality is contained in Watson Studio Desktop. If you want to build a machine learning model using a commercial tool, you should consider using a data mining product. The most prominent of these types of products are: SPSS Modeler and SAS Enterprise Miner. In addition, A version of SPSS Modeler is also available in Watson Studio Desktop, based on the cloud version of the tool.

We’ll talk more about cloud-based tools in the next video. In commercial software, model deployment is tightly integrated in the model building process. This diagram shows an example of the SPSS Collaboration and Deployment Services which are used to deploy any type of asset created by the SPSS software tools suite. Other vendors use the same type of process. Commercial software can also export models in an open format. For example, SPSS Modeler supports the exporting of models as Predictive Model Markup Language, or PMML, which can be read by many other commercial and open software packages. Model monitoring is a new discipline and there are currently no relevant commercial tools available. As a result, open source is the first choice. The same is true for code asset management. Open source with Git and GitHub is the effective standard. Data asset management, often called data governance or data lineage, is a crucial part of enterprise grade data science.

Data must be versioned and annotated using metadata. Vendors, including Informatica Enterprise Data Governance and IBM, provide tools for these specific tasks. The IBM InfoSphere Information Governance Catalog covers functions like data dictionary, which facilitates discovery of data assets. Each data asset is assigned to a data steward -- the data owner.

The data owner is responsible for that data asset and can be contacted. Data lineage is also covered; this enables a user to track back through the transformation steps followed in creating the data assets. The data lineage also includes a reference to the actual source data. Rules and policies can be added to reflect complex regulatory and business requirements for data privacy and retention. Watson Studio is a fully integrated development environment for data scientists. It’s usually consumed through the cloud, and we’ll cover more about it in a later lesson. There is also a desktop version available. Watson Studio Desktop combines Jupyter Notebooks with graphical tools to maximize data scientists’ performance. Watson Studio, together with Watson Open Scale, is a fully integrated tool covering the full data science life cycle and all the tasks we’ve discussed previously. We’ll talk more about both in the next lesson. but just keep in mind that they can be deployed in a local data center on top of Kubernetes or RedHat OpenShift. Another example of a fully integrated commercial tool is H2O Driverless AI, which covers the complete data science life cycle. In this lesson, you’ve learned how most common data science tasks are supported by commercial tools. In the next video, we’ll discover data science tools that are available exclusively on the cloud.

## **Cloud Based Tools for Data Science (8:08)**

Since we previously covered commercial tools for data science, let's look at the cloud-based tools you'll find in many enterprise projects. Take another look at the overview of different tool categories. Since cloud products are newer species, they follow the trend of having multiple tasks integrated into a single tool. This especially holds true for the Tasks marked Green in the diagram. Let's start with a fully integrated visual tools category, since these tools introduce a component where large scale execution of data science workflows happens in compute clusters. We've changed the title here, an added the word platform. These clusters are composed of multiple server machines transparently for the user in the background. Watson studio, together with Watson Open scale covers the complete development lifecycle for all data science, machine learning and AI tasks. Another example is Microsoft Azure Machine Learning. This is also a fully.

Cloud hosted offering supporting the complete development life cycle of all data science, machine learning and AI tasks. And finally, another example is H2O driverless AI which we've already introduced in the last video. Although it is a product that you download and install one click deployment is available for the common cloud service providers. Since operations and maintenance are not done by the cloud provider. As is the case with Watson Studio Open scale and Azure Machine Learning, this delivery model should not be confused with platform or software as a service PA S or SAS. In data management, with some exceptions, there are SAS versions of existing open source and commercial tools. Remember, SAS stands for software as a service. It means that the cloud provider operates the tool for you in the cloud. As an example, the cloud provider. Operates the product by backing up your data and configuration and installing updates. As mentioned, there is proprietary tooling which is only available as a cloud product. Sometimes it's only available from a single cloud provider.

One example of such a service is Amazon Web Services Dynamo DB, a no SQL database that allows storage and retrieval of data in a key value or a document store format. The most prominent document data structure is Jason. Another flavor of such a service is cloudant, which is a database as a service offering but under the hood. It is based on the open source. Apache couch DB. It has an advantage. Although complex operational tasks like updating backup restore and scaling are done by the cloud provider under the hood. This offering is compatible with couch DB, therefore the application can be migrated to another couch DB server without changing the application. And IBM offers DB 2 as a service as well. This is an example of a commercial database made available as a software as a service offering in the cloud, taking operational tasks away from the user. When it comes to commercial data integration tools, we talk not only about extract, transform and load or ETL tools, but also about extract, load and transform or Y Lt tools. This means the transformation steps are not done by a data integration team. But are pushed towards the domain of the data scientist or dated engineer. Two widely used commercial data integration tools are Informatica Cloud data integration and IBM's data refinery data refinery enables transformation of large amounts of raw data into consumable quality information in a spreadsheet like user interface. Data refinery is part of IBM Watson Studio. The market for cloud data visualization tools is huge and every major cloud vendor has one. An example of a smaller company's cloud-based data visualization tool is Datameer. IBM offers its famous Cognos business intelligence suite as a cloud solution as well. IBM data refinery also offers data exploration and visualization functionality in Watson Studio.

Again, these are just some examples of a rapidly changing and growing commercial ecosystem among a huge number of established and emerging vendors. In Watson Studio, an abundance of different visualizations can be used to better understand data. For example, this 3D bar chart enables you to visualize a target value on the vertical dimension, which is dependent on two other values on the horizontal dimensions. Coloring enables you to visualize a third dimension. Hierarchical change bundling enables you to visualize correlations and affiliations between entities. If sufficient, a classic barchart can do the job as well, whereas a 2D scatter plot with a heat map shows two dependent data fields, one on the Y axis and one as color intensity.A tree map shows distribution of subsets within a set. The famous pie chart does the same but in a non-hierarchical manner. And finally, a word cloud pops out significant terms in a document corpus model building can be done using a service such as Watson, machine learning, Watson.

Machine learning can train. And build models using various open source libraries. Google has a similar service on their cloud called AI platform training. Nearly every cloud provider has a solution for this task. Model deployment in commercial software is usually tightly integrated to the model building process. Here is an example of the SPSS collaboration and deployment services which can be used to deploy any type of asset created by the SPSS software tools suite. The same holds for other vendors. In addition, commercial software can export models in an open format as an example. SPSS Modeler supports exporting models as predictive model. Markup language or PMML, which can be read by numerous other commercial and open software packages. Watson machine learning can also be used to deploy a model and make it available to consumers using a rest interface. Amazon sage maker model monitor is an example of a cloud tool that continuously monitors deployed machine learning and deep learning models.

Again, every major cloud provider has a similar tooling. This is also the case for Watson Open scale, open scale, and Watson studio unify the landscape. Everything marked in green can be done using Watson Studio and Watson Open scale open scale will be covered in a later video. You've learned how the most common tasks in data science are supported by commercial cloud tools. Integration provides us the ability to use the same tools for multiple tasks in the next videos, we'll look at packages, APIs, datasets, and models for data science.

## **Libraries for Data Science (4:31)**

Libraries are a collection of functions and methods that enable you to perform a wide variety of actions without writing the code yourself. We will focus on Python libraries: Scientific Computing Libraries in Python Visualization Libraries in Python High-Level Machine Learning and Deep Learning Libraries – “High-level” simply means you don’t have to worry about details, although this makes it difficult to study or improve Deep Learning Libraries in Python Libraries used in other languages

Libraries usually contain built-in modules providing different functionalities that you can use directly; these are sometimes called “frameworks.” There are also extensive libraries, offering a broad range of facilities. Pandas offers data structures and tools for effective data cleaning, manipulation, and analysis. It provides tools to work with different types of data. The primary instrument of Pandas is a two-dimensional table consisting of columns and rows. This table is called a “DataFrame” and is designed to provide easy indexing so you can work with your data.

NumPy libraries are based on arrays, enabling you to apply mathematical functions to these arrays. Pandas is actually built on top of NumPy Data visualization methods are a great way to communicate with others and show the meaningful results of analysis. These libraries enable you to create graphs, charts and maps. The Matplotlib package is the most well-known library for data visualization, and it’s excellent for making graphs and plots. The graphs are also highly customizable.

Another high-level visualization library, Seaborn, is based on matplotlib. Seaborn makes it easy to generate plots like heat maps, time series, and violin plots. For machine learning, the Scikit-learn library contains tools for statistical modeling, including regression, classification, clustering and others. It is built on NumPy, SciPy, and matplotlib, and it’s relatively simple to get started. For this high-level approach, you define the model and specify the parameter types you would like to use. For deep learning, Keras enables you to build the standard deep learning model.

Like Scikit-learn, the high-level interface enables you to build models quickly and simply. It can function using graphics processing units (GPU), but for many deep learning cases a lower-level environment is required. TensorFlow is a low-level framework used in large scale production of deep learning models. It’s designed for production but can be unwieldy for experimentation.

Pytorch is used for experimentation, making it simple for researchers to test their ideas Apache Spark is a general-purpose cluster-computing framework that enables you to process data using compute clusters. This means that you process data in parallel, using multiple computers simultaneously. The Spark library has similar functionality as Pandas Numpy Scikit-learn Apache Spark data processing jobs can use Python R Scala, or SQL. There are many libraries for Scala, which is predominately used in data engineering but is also sometimes used in data science.

Let’s discuss some of the libraries that are complementary to Spark Vegas is a Scala library for statistical data visualizations. With Vegas, you can work with data files as well as Spark DataFrames. For deep learning, you can use BigDL.

R has built-in functionality for machine learning and data visualization, but there are also several complementary libraries: ggplot2 is a popular library for data visualization in R. You can also use libraries that enable you to interface with Keras and TensorFlow. R has been the de-facto standard for open source data science but it is now being superseded by Python.

## **Application Programming Interfaces (API) (4:30)**

In this video we will discuss Application Programming Interfaces, or APIs. Specifically, we will discuss: What an API is, API Libraries, REST APIs, including: Request and Response. An API lets two pieces of software talk to each other. For example you have your program, you have some data, you have other software components. You use the API to communicate with the other software components. You don’t have to know how the API works, you just need to know its inputs and outputs. Remember, the API only refers to the interface, or the part of the library that you see. The “library” refers to the whole thing. Consider the pandas library. Pandas is actually a set of software components, many of which are not even written in Python. You have some data. You have a set of software components. We use the pandas API to process the data by communicating with the other software components. There can be a single software component at the back end, but there can be a separate API for different languages. Consider TensorFlow, written in C++. There are separate APIs in Python, JavaScript, C++ Java, and Go. The API is simply the interface.

There are also multiple volunteer-developed APIs for TensorFlow; for example Julia, MATLAB, R, Scala, and many more. REST APIs are another popular type of API. They enable you to communicate using the internet, taking advantage of storage, greater data access, artificial intelligence algorithms, and many other resources. The RE stands for “Representational,” the S stands for “State,” the T stand for “Transfer.” In rest APIs, your program is called the “client.” The API communicates with a web service that you call through the internet. A set of rules governs Communication, Input or Request, and Output or Response. Here are some common API-related terms. You or your code can be thought of as a client. The web service is referred to as a resource. The client finds the service through an endpoint. The client sends the request to the resource and the response to the client. HTTP methods are a way of transmitting data over the internet We tell the REST APIs what to do by sending a request. The request is usually communicated through an HTTP message. The HTTP message usually contains a JSON file, which contains instructions for the operation that we would like the service to perform. This operation is transmitted to the web service over the internet. The service performs the operation. Similarly, the web service returns a response through an HTTP message, where the information is usually returned using a JSON file. This information is transmitted back to the client. The Watson Speech to Text API is an example of a REST API. This API converts speech to text. In the API call, you send a copy of the audio file to the API; this process is called a post request. The API then sends the text transcription of what the individual is saying. The API is making a get request.

The Watson Language-Translator API provides another example. You send the text you would like to translate into the API, the API translates the text and sends the translation back to you. In this case we translate English to Spanish. In this video, we’ve discussed what an API is, API Libraries, REST APIs, including Request and Response.

## **Data Sets - Powering Data Science (6:10)**

In this video we’ll discuss data sets: what they are, why they are important in data science, and where to find them. Let’s first loosely define what a data set is. A data set is a structured collection of data. Data embodies information that might be represented as text, numbers, or media such as images, audio, or video files. A data set that is structured as tabular data comprises a collection of rows, which in turn comprise columns that store the information.

One popular tabular data format is "comma separated values," or CSV. A CSV file is a delimited text file where each line represents a row and data values are separated by a comma. For example, imagine a data set of observations from a weather station. Each row represents an observation at a given time, while each column contains information about that particular observation, such as the temperature, humidity, and other weather conditions.

Hierarchical or network data structures are typically used to represent relationships between data. Hierarchical data is organized in a tree-like structure, whereas network data might be stored as a graph. For example, the connections between people on a social networking website are often represented in the form of a graph. A data set might also include raw data files, such as images or audio.

The MNIST dataset is popular for data science. It contains images of handwritten digits and is commonly used to train image processing systems. Traditionally, most data sets were considered to be private because they contain proprietary or confidential information such as customer data, pricing data, or other commercially sensitive information. These data sets are typically not shared publicly. Over time, more and more public and private entities such as scientific institutions, governments, organizations and even companies have started to make data sets available to the public as “open data," providing a wealth of information for free. For example, the United Nations and federal and municipal governments around the world have published many data sets on their websites, covering the economy, society, healthcare, transportation, environment, and much more. Access to these and other open data sets enable data scientists, researchers, analysts, and others to uncover previously unknown and potentially useful insights. They can create new applications for both commercial purposes and the public good. They can also carry out new research.

Open data has played a significant role in the growth of data science, machine learning, and artificial intelligence and has provided a way for practitioners to hone their skills on a wide variety of data sets. There are many open data sources on the internet. You can find a comprehensive list of open data portals from around the world on the Open Knowledge Foundation’s datacatalogs.org website. The United Nations, the European Union, and many other governmental and intergovernmental organizations maintain data repositories providing access to a wide range of information. On Kaggle, which is a popular data science online community, you can find and contribute data sets that might be of general interest. Last but not least, Google provides a search engine for data sets that might help you find the ones that have particular value for you. It’s important to recognize that open data distribution and use might be restricted, as defined by its licensing terms. In absence of a license for open data distribution, many data sets were shared in the past under open source software licenses. These licenses were not designed to cover the specific considerations related to the distribution and use of data sets.

To address the issue, the Linux Foundation created the Community Data License Agreement, or CDLA. Two licenses were initially created for sharing data: CDLA-Sharing and CDLA-Permissive. The CDLA-Sharing license grants you permission to use and modify the data. The license stipulates that if you publish your modified version of the data you must do so under the same license terms as the original data. The CDLA-Permissive license also grants you permission to use and modify the data. However, you are not required to share changes to the data. Note that neither license imposes any restrictions on results you might derive by using the data, which is important in data science. Let’s say, for example, that you are building a model that performs a prediction. If you are training the model using CDLA-licensed data sets, you are under no obligation to share the model, or to share it under a specific license if you do choose to share it.

In this video you’ve learned about open data sets, their role in data science, and where to find them. We’ve also introduced the Community Data License Agreement, which makes it easier to share open data. One important aspect that we didn’t cover in this video is data quality and accuracy, which might vary greatly depending on who collected and contributed the data set. While some open data sets might be good enough for personal use, they might not meet enterprise requirements due to the impact they might have on the business. In the next module, you will learn about the Data Asset eXchange, a curated open data repository.

## **Data Asset Exchange (3:05)**

Despite the growth of open data sets that are available to the public, it can still be difficult to discover data sets that are both high quality and have clearly defined license and usage terms. To help solve this challenge, IBM created the Data Asset eXchange, or "DAX,”, which we’ll introduce in this video. DAX provides a trusted source for finding open data sets that are ready for to use in enterprise applications. These data sets and which cover a wide variety of domains, including images, video, text, and audio.

Because DAX provides a high level of curation for data set quality, as well as licensing and usage terms, DAX data sets are typically easier to adopt, whether in research or commercial projects. Wherever possible, DAX aims to make data sets available under one of the variants of the CDLACommunity Data License Agreement, in order to foster data sharing and collaboration. DAX also provides a single place to access unique data sets, in particular from IBM Research projects. To make it easier for developers to get started with using the data sets, DAX also provides tutorials in the form of notebooks that walk through the basics of data cleaning, pre-processing, and exploratory analysis.

For some data sets, there are also notebooks illustrating how to perform more complex analysis, such as creating charts, statistical analysis, time-series analysis, training machine learning models, and integrating deep learning via using the Model Asset eXchange, (a project closely related to DAX and also available on the IBM Developer website). In this way, DAX helps developers to create end-to-end analytic and machine learning workflows and to consume open data and models with confidence under clearly defined license terms. Let’s say you’ve found a data set that might be of interest to you.

On the data set page you can download the compressed data set archive from cloud storage, explore the data set using Jupyter Notebooks, review the data set metadata, such as format, licensing terms and size, and preview some parts of the data set. Most data sets on DAX are complemented by one or more Jupyter Notebooks that you can use to perform data cleaning, pre-processing, and exploratory analysis. These notebooks run "as is"as is in Watson Studio, IBM’s Data Sciencedata science platform. Jupyter Notebooks and Watson Studio are covered later during in this course. In this video, you’ve learned about IBM’s open data repository, the Data Asset eXchange. In the hands-on lab you’ll have a chance to explore the repository.

## **Machine Learning Models (7:03)**

## In this video, we’ll introduce you to machine learning and deep learning models. Data contains a wealth of information that can be used to solve certain types of problems. Traditional data analysis approaches, such as a person manually inspecting the data or a specialized computer program that automates the human analysis, quickly reach their limits due to the amount of data to be analyzed or the complexity of the problem. Machine learning uses algorithms – also known as ”models” - to identify patterns in the data. The process by which the model learns these patterns from data is called “model training." Once a model is trained, it can then be used to make predictions. When the model is presented with new data, it tries to make predictions or decisions based on the patterns it has learned from past data.

Machine learning models can be divided into three basic classes: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is one of the most commonly used type of machine learning models. In supervised learning, a human provides input data and the correct outputs. The model tries to identify relationships and dependencies between the input data and the correct output. Generally speaking, supervised learning is used to solve regression and classification problems.

Let’s look at an example for each problem type: Regression models are used to predict a numeric, or “real," value. For example, given information about past home sales, such as geographic location, size, number of bedrooms, and sales price, you can train a model to predict the estimated sales price for other homes with similar characteristics.

Classification models are used to predict whether something belongs to a category, or “class." For example, given a set of emails along with a designation of whether or not they are considered spam, an algorithm can be trained to identify unsolicited emails.

In unsupervised learning, the data is not labelled by a human. The models must analyze the data and try to identify patterns and structure within the data based only on the characteristics of the data itself. Clustering and anomaly detection are two examples of this learning style.

Clustering models are used to divide each record of a data set into one of a small number of similar groups. An example of a clustering model could be providing purchase recommendations for an e-commerce store based on past shopping behavior and the content of a shopping basket. Anomaly detection identifies outliers in a data set, such as fraudulent credit card transactions or suspicious online log-in attempts. The third type of learning, reinforcement learning, is loosely based on the way human beings and other organisms learn. Think about a mouse in a maze. If the mouse gets to the end of the maze it gets a piece of cheese. This is the “reward” for completing a task. The mouse learns – through trial and error – how to get through the maze to get as much cheese as it can. In a similar way, a reinforcement learning model learns the best set of actions to take, given its current environment, in order to get the most reward over time.

This type of learning has recently been very successful in beating the best human players in games such as go, chess, and popular strategy video games.

Deep learning is a specialized type of machine learning. It refers to a general set of models and techniques that tries to loosely emulate the way the human brain solves a wide range of problems. It is commonly used to analyze natural language, both spoken and text, as well as images, audio, and video, to forecast time series data and much more. Deep learning has had a lot of recent success in these and other areas and is therefore becoming an increasingly popular and important tool for data science. Deep learning typically requires very large data sets of labeled data to train a model, is compute-intensive, and usually requires special purpose hardware to achieve acceptable training times. You can build a custom deep learning model from scratch or use pre-trained models from public model repositories.

Deep learning models are implemented using popular frameworks such as TensorFlow, PyTorch, and Keras. Deep learning frameworks typically provide a Python API, and many support other programming languages, such as C++ and JavaScript. You can download pre-trained state-of-the-art models from repositories that are commonly referred to as model "zoos." Popular model zoos include those provided by TensorFlow, PyTorch, Keras, and ONNX. Models are also published by academic and commercial research groups. While it is beyond the scope of this video to explain in detail how you would go about building a model, let’s briefly outline the high-level tasks using an example. Assume you want to enable an application to identify objects in images by training a deep learning model.

First, you collect and prepare data that will be used to train a model. Data preparation can be a time-consuming and labor-intensive process. In order to train a model to detect objects in images, you need to label the raw training data by, for example, drawing bounding boxes around objects and labeling them. Next, you build a model from scratch or select an existing model that might be well suited for the task from a public or private resource. You then train the model on your prepared data. During training, your model learns from the labeled data how to identify objects that are depicted in an image. Once training has commenced, you analyze the training results and repeat the process until the trained model performance meets your requirements.

When the trained model performs as desired, you deploy it to make it available to your applications. In this video, you’ve learned about machine learning and deep learning, what they are used for, and where to find open source models. In the next video, we’ll introduce you to the Model Asset eXchange, a curated collection of ready-to-use and customizable deep learning models.

## **The Model Asset Exchange (5:59)**

In this video, we will introduce you to the Model Asset eXchange on IBM Developer, a free open source resource for deep learning models. Throughout the video we will refer to the Model Asset eXchange as "MAX." In the previous video, we briefly outlined the high-level tasks you need to complete to train a model from scratch. Due to the amount of data, labor, time, and resources required to complete the tasks, time to value can be quite long. To reduce time to value, consider taking advantage of pre-trained models for certain types of problems. These pre-trained models can be ready to use right away, or they might take less time to train.

The Model Asset eXchange is a free open source repository for ready-to-use and customizable deep learning microservices. These microservices are configured to use pre-trained or custom-trainable state-of-the-art deep learning models to solve common business problems. These models have been reviewed, tested, and can be quickly deployed in local and cloud environments. All models in MAX are available under permissive open source licenses, making it easier to use them for personal and commercial purposes and reducing the risk of legal liabilities.

On MAX, you can find models for a variety of domains, including image, audio, video, and natural language analysis. This list includes a small selection. In the lab for this module, you’ll have a chance to explore those models. Let’s take a look at the components of a typical model-serving microservice. Each microservice includes the following components:

1. A pre-trained deep learning model.
2. Code that pre-processes the input before it is analyzed by the model and code that post-processes the model output.
3. A standardized public API that makes the services’ functionality available to applications.

The MAX model-serving microservices are built and distributed as open-source Docker images. Docker is a container platform that makes it easy to build applications and to deploy them in a development, test, or production environment. The Docker image source is published on GitHub and can be downloaded, customized as needed, and used in personal or commercial environments. You can deploy and run these images in a test or production environment using Kubernetes, an open-source system for automating deployment, scaling, and management of containerized applications in private, hybrid, or public clouds.

A popular enterprise-grade Kubernetes platform is Red Hat OpenShift, which is available on IBM Cloud, Google Cloud Platform, Amazon Web Services, and Microsoft Azure. The model-serving microservices expose a REST API that developers can use to incorporate deep learning into their applications and services. Because REST APIs can be consumed using any programming language, you can easily integrate these services into your existing ecosystem.

The API exposes a prediction endpoint and one or more metadata endpoints. This example shows the endpoints for the Object Detection microservice. The /model/predict endpoint takes an image as input and returns as a response a list of objects that were detected in the image, along with bounding box coordinates that identify where the detected object is located. Some prediction endpoints can also accept additional input parameters that impact the produced results, such as filters. This microservice exposes two metadata endpoints, /model/labels and /model/metadata. These endpoints provide information such as the objects that can be detected and the deep learning model used to derive the answer given the input. In the lab portion of this module, you will have a chance to explore and test these endpoints using a web browser.

Each endpoint accepts application-friendly inputs, such as an image in JPG, PNG, or GIF format, instead of a model-specific data structure. Each endpoint also generates application-friendly outputs, such as standardized JSON, which is a lightweight data-interchange format.

Let’s take a closer look at what happens when an application invokes the prediction endpoint. In this example, a user has selected an image in a web application, the prediction endpoint is invoked, and the image is uploaded. The microservice prepares the input image for processing, runs the deep learning model that identifies objects in the image, generates a response using the prediction results, and returns the result to the application. The application renders the results by drawing bounding boxes and labels. In this video, we’ve introduced the Model Asset eXchange, a free and open source repository for microservices that make deep learning functionality available to applications and services in local and cloud environments.

In the lab, you will have a chance to try a model-serving microservice, explore its API, and learn more about how you can leverage it from a web application and an Internet of Things application.

### Getting started with the Model Asset Exchange and the Data Asset Exchange

In this lab you will explore the Model Asset Exchange (MAX) and the Data Asset Exchange (DAX), which are two open source Data Science resources on IBM Developer.

Objective for Exercise 1:

* Find ready-to-use deep learning models on the Model Asset Exchange
* Locate resources that guide you through deployment in a local or cloud environment
* Explore the deep learning model-serving microservice API using your web browser
* Articulate how developers can consume those microservices

Objective for Exercise 2

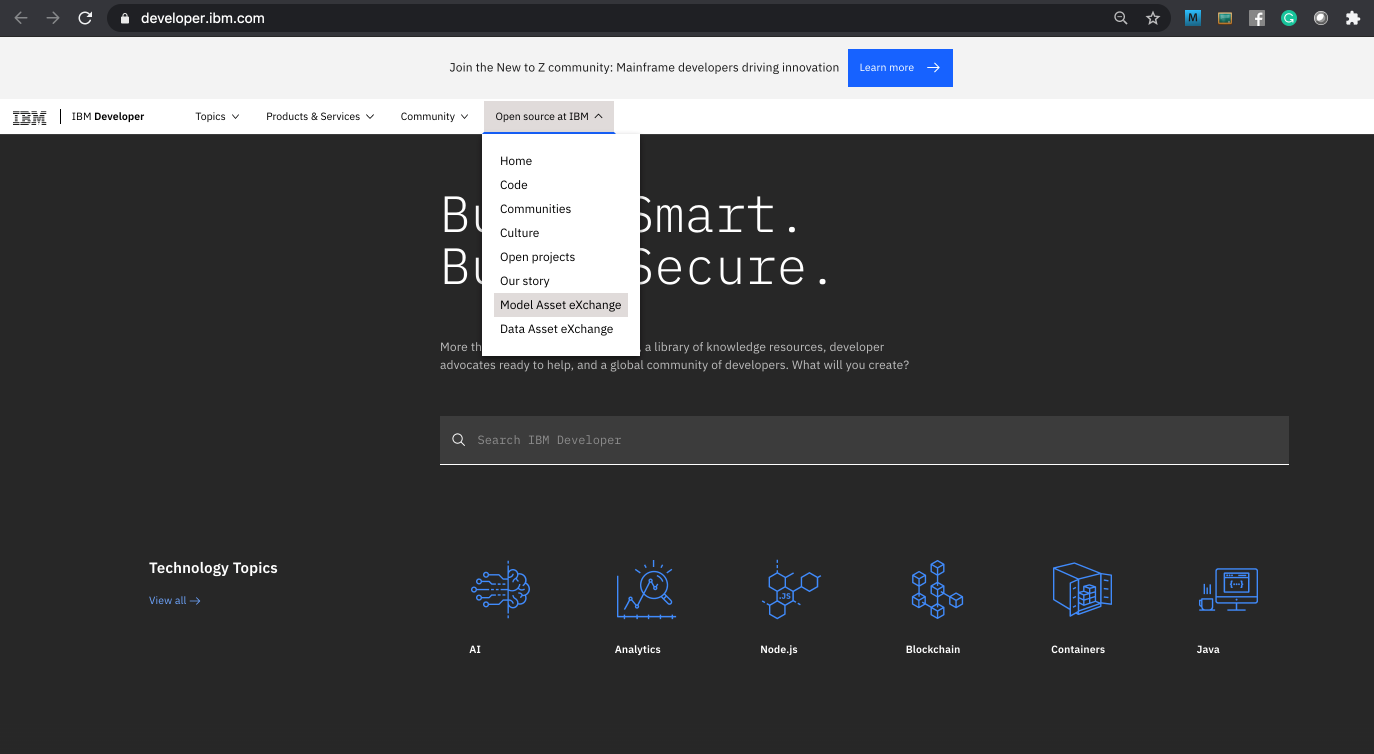
* Where to find open data sets on IBM Developer
* How to explore those data sets

It will take you approximately 30 minutes to complete the lab. Only a web browser is required to complete the tasks.

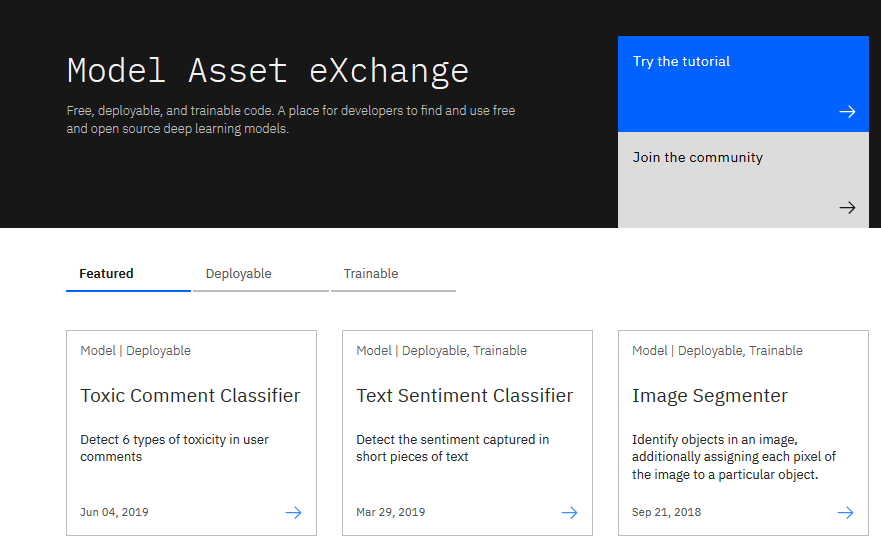
**Exercise 1** - Explore deep learning models

The Model Asset Exchange is a curated repository of open source deep learning models for a variety of domains, such as text, image, audio, and video processing.

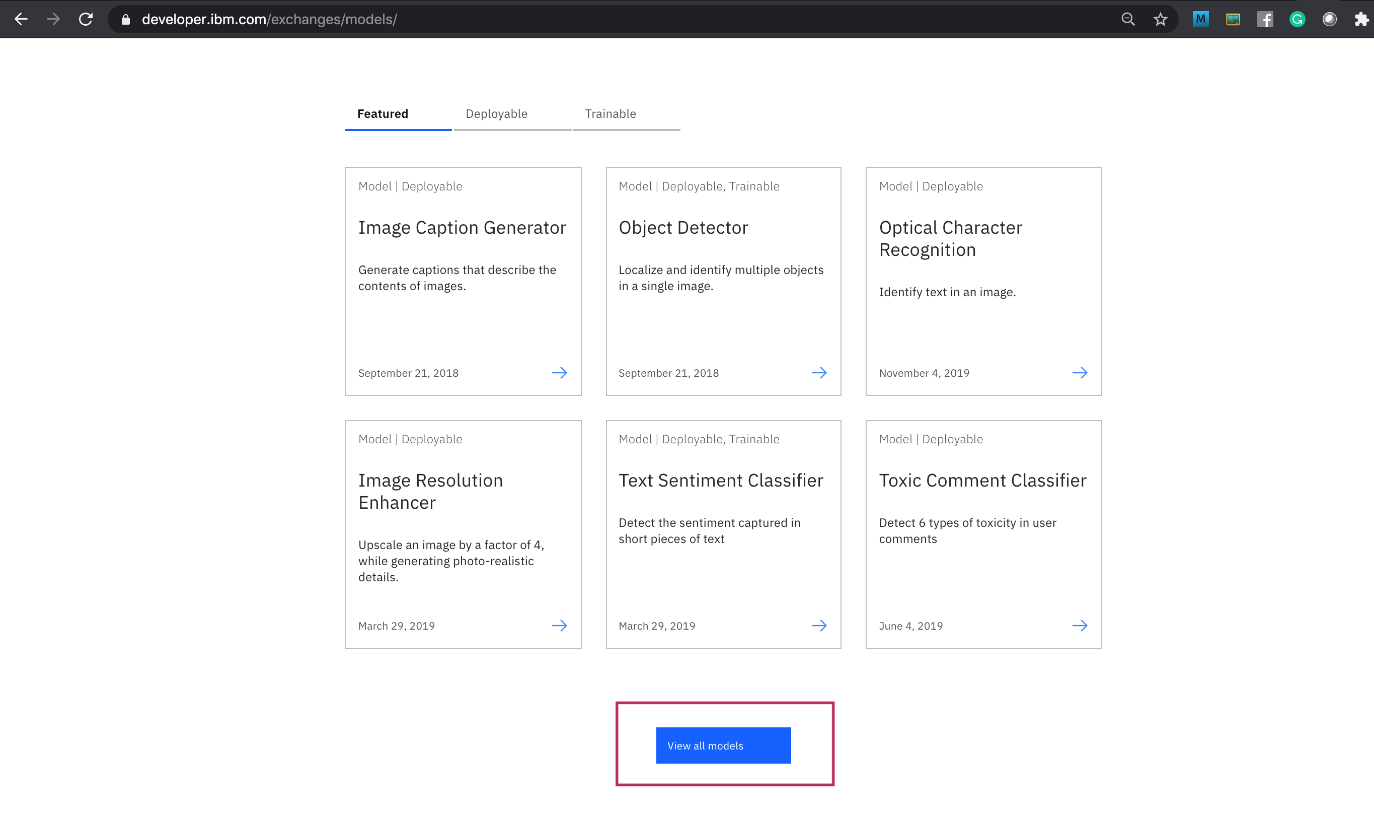
1. Open [https://developer.ibm.com/](https://developer.ibm.com/?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ&cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ) in your web browser.
2. From the main menu select **“Open Source at IBM” > “Model Asset eXchange”.**



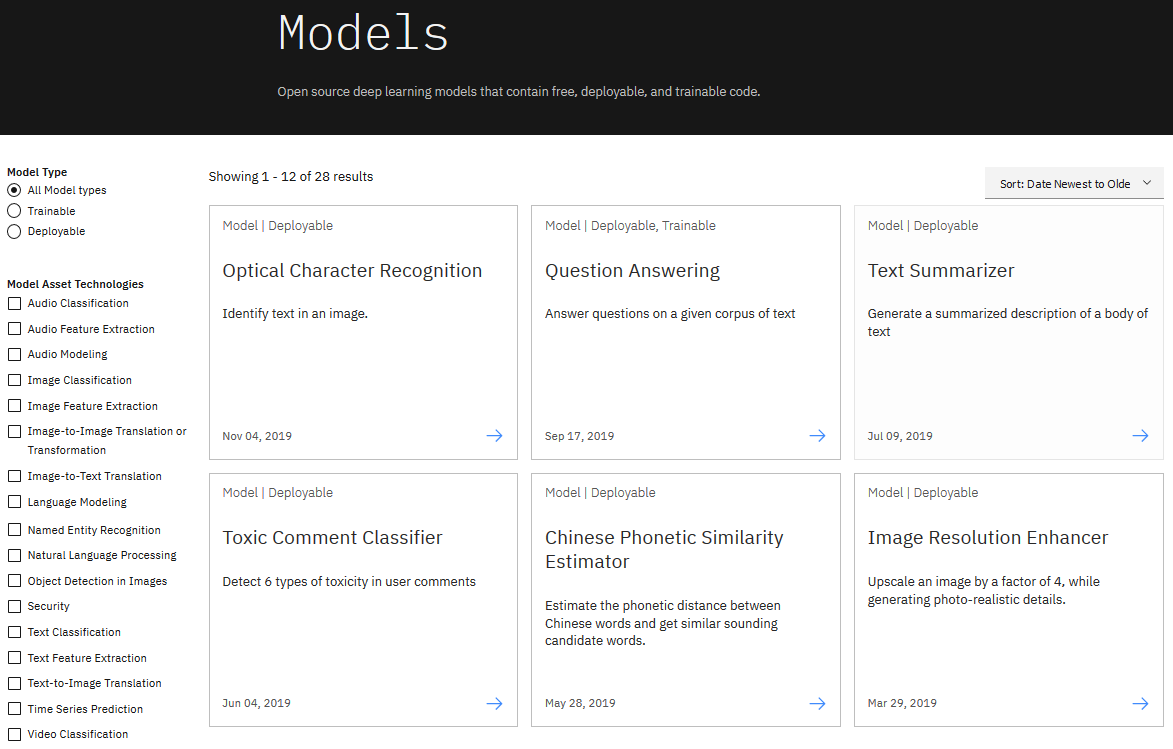
1. The MAX home page is displayed. In this introductory lab exercise, we are going to focus on a few MAX key features. More detailed information is available in the Learning Path, which covers common deployment and consumption scenarios.



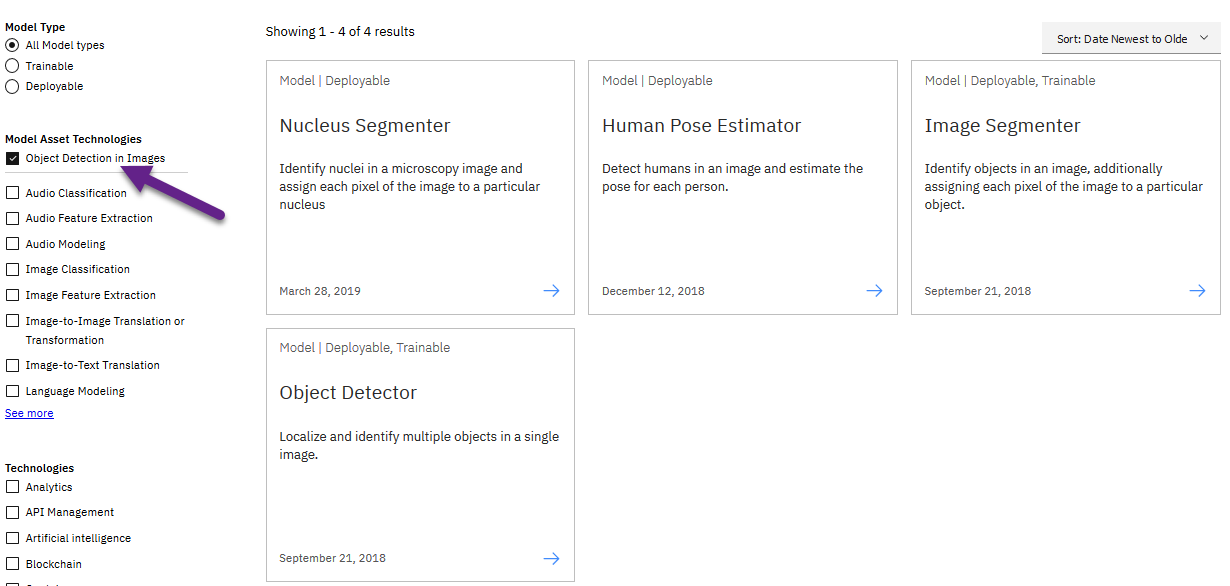
1. Scroll down the page and click on **View all Models**.



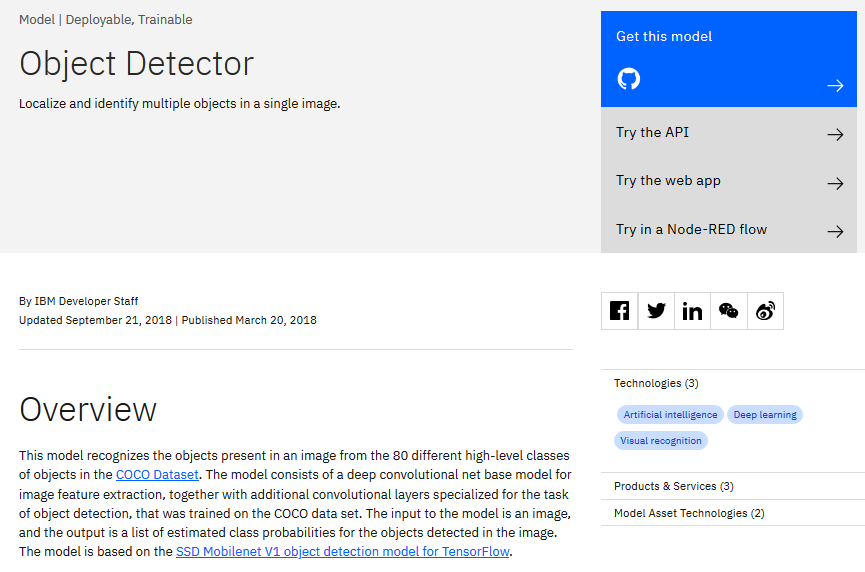
A page with all the models is displayed.



1. Select **Object Detection in Images** from the Model Asset Technologies list on the left side.

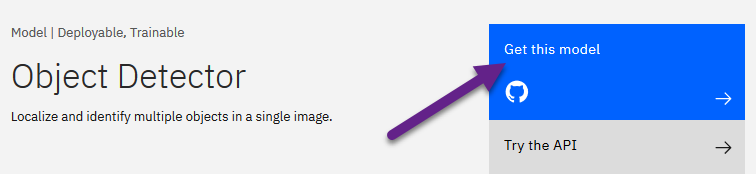


1. Four models should be displayed: Nucleus Segmenter, Human Pose Estimator, Image Segmenter, and Object Detector. Select the **Object Detector model.**

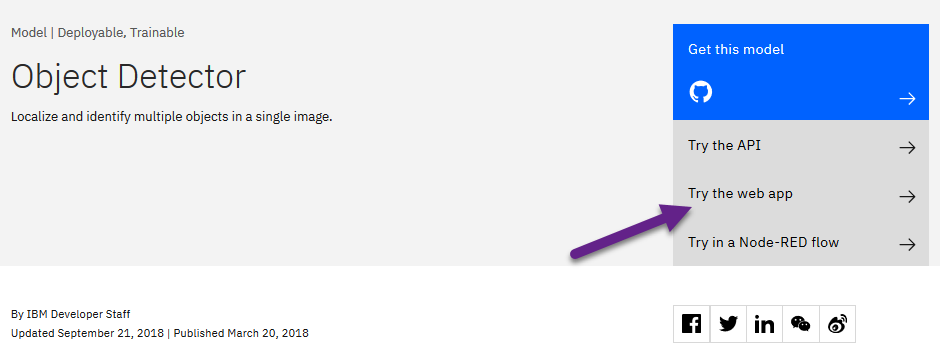


On the model page you can learn about the underlying technology, read up on related research, and explore deployment and consumption options.

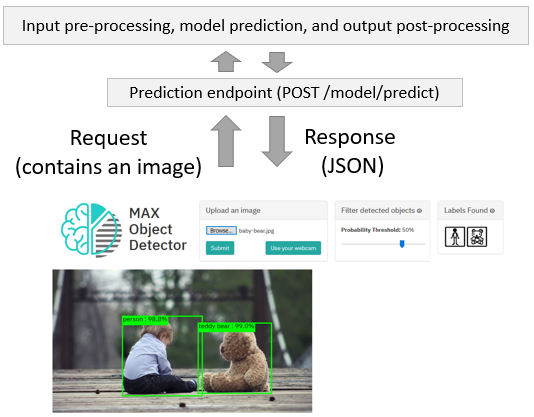
1. The model’s source is published on GitHub and can be downloaded and modified if desired. Click “Get this model” to open the repository in a new browser tab (or window) and view the files. You can use this once you are comfortable deploying models.



1. Close the GitHub window or tab
2. On the Object Detector page ([https://developer.ibm.com/exchanges/models/all/max-object-detector/](https://developer.ibm.com/exchanges/models/all/max-object-detector?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ)) on top right, choose **Try the web app**.



The demo web application uploads an image (or takes a picture using the web cam), sends a request to an Object Detector microservice and visualizes the response by drawing bounding boxes around detected objects and attaching a label.



Upload an image (or take a picture using the web camera) and inspect the visualized results.

1. Change the filter conditions (e.g. lower the probability threshold) and observe how they impact the visualized results. Note the web application caches the response and applies the filter on the cached data. This was done for two reasons:

* It eliminates the need to re-process the image using the deep learning microservice (which significantly lowers the application’s response time)
* It illustrates how client-side filtering can be applied to the results if the prediction endpoint doesn’t support filtering by the desired criteria.

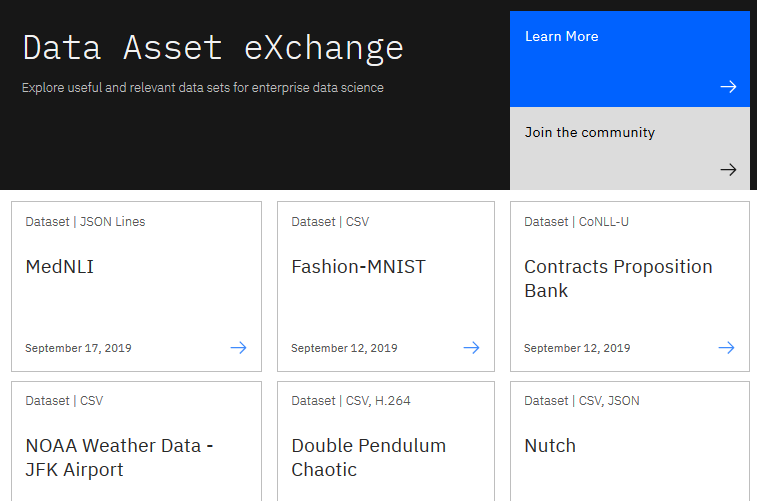
If you are interested in learning more about how the application was implemented, take a look at the code pattern: [https://developer.ibm.com/patterns/create-a-web-app-to-interact-with-objects-detected-using-machine-learning/](https://developer.ibm.com/patterns/create-a-web-app-to-interact-with-objects-detected-using-machine-learning?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ)

This concludes Exercise 1 of this lab, which introduced the Model Asset Exchange.

**Exercise 2:** Explore deep learning datasets

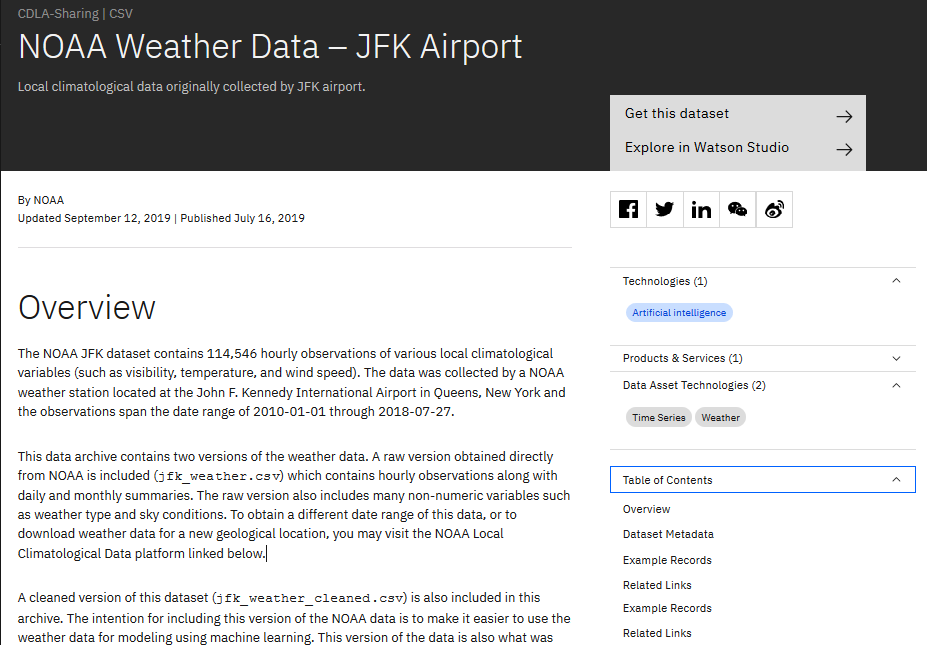
The Data Asset Exchange is a curated collection of open datasets from IBM Research and 3rd parties that you can use to train models.

1. Open [https://developer.ibm.com/](https://developer.ibm.com/?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ) in your web browser.
2. From the main menu select “Open Source at IBM” > “Data Asset eXchange”. The DAX homepage is displayed.



The collection includes datasets from the Debater project ([https://www.research.ibm.com/artificial-intelligence/project-debater/](https://www.research.ibm.com/artificial-intelligence/project-debater?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ)), datasets that can be used to train models to perform document layout analysis, natural language processing, time series analysis and more.

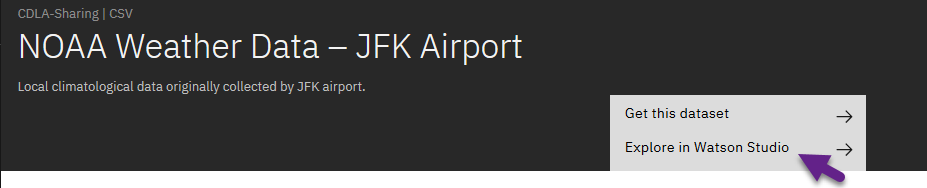
1. Open the NOAA Weather Data dataset ([https://developer.ibm.com/exchanges/data/all/jfk-weather-data/](https://developer.ibm.com/exchanges/data/all/jfk-weather-data?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ)), which contains data from a weather station at the John F. Kennedy Airport in New York spanning 8 years. This dataset was used to train the weather forecaster model on MAX (<https://developer.ibm.com/exchanges/models/all/max-weather-forecaster/>).



You can download the dataset using the “Get this dataset” link. Datasets are stored as compressed archives, which you can extract using any utility that supports the tar.gz format. If you are not familiar with this file format take a look at this short open source tutorial [https://opensource.com/article/17/7/how-unzip-targz-file](https://opensource.com/article/17/7/how-unzip-targz-file?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ).

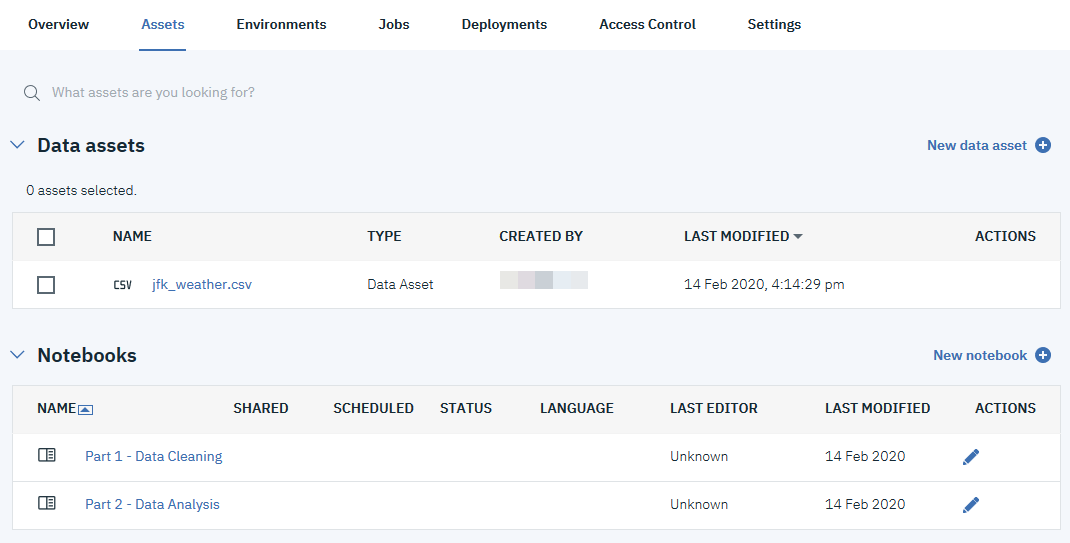
1. Inspect the dataset’s metadata.

This dataset is stored as tabular data and formatted as a comma separated value (CSV) file, which is a very popular basic data exchange format. The dataset was published under the data science friendly CDLA-Sharing license ([https://cdla.io/](https://cdla.io/?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork-20083975&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newsletter.M12345678&cvo_campaign=000026UJ)). The dataset contains time-series data and can be used to predict weather trends. 5. Most datasets are complemented by Python notebooks that you can use to explore, pre-process, and analyze the data. You can access the notebook (or notebooks) by clicking the “Explore in Watson Studio” or “Try the notebook” link:



The notebooks are hosted on Watson Studio, IBM’s Data Science platform. Later in this course you’ll learn more about Watson Studio, notebooks and how to run them.

1. [Optional] If you are already familiar with notebooks and Watson Studio feel free to open the link and import the project or notebook. The example below depicts the weather dataset project assets, which include the raw data file and two notebooks.



This concludes Exercise 2 of this lab, which introduced the Data Asset Exchange.

## **Overview of Git/GitHub (4:27)**

In this video, you’ll get an overview of Git and GitHub, which are popular environments among developers and data scientists for performing version control of source code files and projects and collaborating with others. You can’t talk about Git and GitHub without a basic understanding of what version control is. A version control system allows you to keep track of changes to your documents. This makes it easy for you to recover older versions of your document if you make a mistake, and it makes collaboration with others much easier.

Here is an example to illustrate how version control works. Let’s say you’ve got a shopping list and you want your roommates to confirm the things you need and add additional items. Without version control, you’ve got a big mess to clean up before you can go shopping. With version control, you know EXACTLY what you need after everyone has contributed their ideas. Git is free and open source software distributed under the GNU General Public License. Git is a distributed version control system, which means that users anywhere in the world can have a copy of your project on their own computer; when they’ve made changes, they can sync their version to a remote server to share it with you. Git isn’t the only version control system out there, but the distributed aspect is one of the main reasons it’s become one of the most common version control systems available. Version control systems are widely used for things involving code, but you can also version control images, documents, and any number of file types.

You can use Git without a web interface by using your command line interface, but GitHub is one of the most popular web-hosted services for Git repositories. Others include GitLab, BitBucket, and Beanstalk. There are a few basic terms that you will need to know before you can get started. The SSH protocol is a method for secure remote login from one computer to another. A repository contains your project folders that are set up for version control. A fork is a copy of a repository. A pull request is the way you request that someone reviews and approves your changes before they become final. A working directory contains the files and subdirectories on your computer that are associated with a Git repository.

There are a few basic Git commands that you will always use. When starting out with a new repository, you only need create it once: either locally, and then push to GitHub, or by cloning an existing repository by using the command "git init". "git add" moves changes from the working directory to the staging area. "git status" allows you to see the state of your working directory and the staged snapshot of your changes.

"git commit" takes your staged snapshot of changes and commits them to the project.

"git reset" undoes changes that you’ve made to the files in your working directory.

"git log" enables you to browse previous changes to a project.

"git branch" lets you create an isolated environment within your repository to make changes.

"git checkout" lets you see and change existing branches.

"git merge" lets you put everything back together again.

To learn how to use Git effectively and begin collaborating with data scientists around the world, you will need to learn the essential commands. Luckily for us, GitHub has amazing resources available to help you get started. Go to try.github.io to download the cheat sheets and run through the tutorials. In the following modules, we'll give you a crash course on setting up your local environment and getting started on a project.

## **GitHub - Getting Started (3:26)**

## In this video, you will learn how to create and merge a branch using the GitHub web interface. A branch is a snapshot of your repository to which you can make changes. It is a copy of the master branch and can be used to develop and test changes to the workflow before merging it back to the master branch. In Git and GitHub, there is a main branch.

The main branch which is called Master, is the one with deployable code and the official working version of your project. It is meant to be stable and it is always advisable never to push any code that is not tested to master. Many times, we want to make changes to the code and workflow in the master branch. That is when we create a copy of the Master branch.

Let’s call it Child Branch. We will then create a copy of the workflow to the child branch in the child branch, changes and experiments are done. We will build and make edits, test the changes and when we are satisfied with the changes we will merge it back to the master branch where we prepare the model for deployment. We can see that all of this is done outside of the main branch and until we merge, changes will not be made to the workflow before we branched. To ensure that changes done by one member, does not impede or affect the flow of work of other members, multiple branches can be created and merged appropriately to master after the workflow is properly tested and approved.

To create branches in GitHub, let’s look at this repository. There is currently one branch in the repository. I want to make some changes, but I don’t want to alter the master in case something goes wrong. We will create a branch. To do that, we will click the drop-down arrow and create a new branch. Let's name it - child branch and then we will click enter. The repository now has two branches, the Master and the Child branch. You can check this by selecting Child branch in the Branch selector drop-down list. Whatever was in the Master branch was copied to the child branch. But we can add files in the child branch without adding any files to the master branch.

To add a file, make sure Child branch is selected in the branch selector drop-down list. Click on create new file. In the space provided, name the file - we will name it testchild.py and then we will add a few lines of code. We will print the statement – Inside child branch. At the bottom of the screen, we will see a section called Commit new file. Commit messages are very important as it helps to keep track of the changes that were made. It is important to add a descriptive commit message so that other team members can understand it. Here we will add a commit message, Create testchild.py, then we will commit the new file. The file gets added to only the child branch. We can check this by going to the master branch by clicking ‘master’ from the Branch selector menu and here we can see that the new file is not added to the master branch. After we have created the new file, tested and made sure that is up to standards. We then want to merge the changes in the child branch to reflect in the master branch. To merge the changes, we will first have to create a pull request, also known as a PR. A pull request in simple terms is a way to notify other team members of your changes and edits and ask them for review so they can be pulled or merged into the master branch. Pull requests are the heart of collaboration on GitHub.

When you open a pull request, you’re proposing your changes and requesting that someone review and pull in your contribution and merge them into the target branch. Pull requests show the differences of the content from both branches. To open a pull request and see the differences between the branches, click on the Compare and pull request button. If you scroll down to the bottom of the screen, you will see something like this that shows you the difference between both branches. As you can see on the screen it shows that one file has changed and the file has two additions, which are the two lines we added to the file and 0 deletions. We will now create the pull request. Add the title and an optional comment for the pull request. Click Create Pull request to create the pull request. You can assign team members to review and approve pull requests. On the next page you will see this image. If you are okay with the changes, click on Merge pull request and click confirm. You will get a confirmation that the pull request has been successfully merged. You can now delete the branch if you no longer need to make any edits or add new information. Now, the child branch has completely merged with the Master branch. You can check the Master branch and we can now see it contains the testchild.py file. You should now be familiar with how to create and merge branches using the web interface.