**DATA SCIENCE**

* 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.
* The typical work day 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.

**BIG DATA:**

“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**.

**1.) 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.

**2.) 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.

**3.) 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.

**4.) 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 age. Is the information real, or is it false?

**5.) 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.

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.

*How ML is different from traditional programming?*

Traditionally in computation and processing data we would bring the data to the computer. You'd want to 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 pretty 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**.

## **Establishing Data Mining Goals**

The first step in data mining requires you to set up goals for the exercise. Obviously, you must identify the key questions that need to be answered. However, going beyond identifying the key questions are the concerns about the costs and benefits of the exercise. Furthermore, you must determine, in advance, the expected level of accuracy and usefulness of the results obtained from data mining. If money were no object, you could throw as many funds as necessary to get the answers required. However, the cost-benefit trade-off is always instrumental in determining the goals and scope of the data mining exercise. The level of accuracy expected from the results also influences the costs. High levels of accuracy from data mining would cost more and vice versa. Furthermore, beyond a certain level of accuracy, you do not gain much from the exercise, given the diminishing returns. Thus, the cost-benefit trade-offs for the desired level of accuracy are important considerations for data mining goals.

## **1.) Selecting Data**

The output of a data-mining exercise largely depends upon the quality of data being used. At times, data are readily available for further processing. For instance, retailers often possess large databases of customer purchases and demographics. On the other hand, data may not be readily available for data mining. In such cases, you must identify other sources of data or even plan new data collection initiatives, including surveys. **The type of data, its size, and frequency of collection have a direct bearing on the cost of data mining exercise.** Therefore, identifying the right kind of data needed for data mining that could answer the questions at reasonable costs is critical.

## **2.) Preprocessing Data**

Preprocessing data is an important step in data mining. Often **raw data are messy**, containing erroneous or irrelevant data. In addition, even with relevant data, information is sometimes missing. In the preprocessing stage, you identify the irrelevant attributes of data and expunge such attributes from further consideration. At the same time, identifying the erroneous aspects of the data set and flagging them as such is necessary. For instance, human error might lead to inadvertent merging or incorrect parsing of information between columns. Data should be subject to checks to ensure integrity. Lastly, you must develop a formal method of dealing with missing data and determine whether the data are missing randomly or systematically.

If the data were missing randomly, a simple set of solutions would suffice. However, when data are missing in a systematic way, you must determine the impact of missing data on the results. For instance, a particular subset of individuals in a large data set may have refused to disclose their income. Findings relying on an individual's income as input would exclude details of those individuals whose income was not reported. This would lead to systematic biases in the analysis. Therefore, you must consider in advance if observations or variables containing missing data be excluded from the entire analysis or parts of it.

## **3.) Transforming Data**

After the relevant attributes of data have been retained, the next step is to determine the appropriate format in which data must be stored. An important consideration in data mining is to reduce the number of attributes needed to explain the phenomena. This may require transforming data. **Data reduction algorithms**, such as **Principal Component Analysis** (demonstrated and explained later in the chapter), can reduce the number of attributes without a significant loss in information. In addition, variables may need to be transformed to help explain the phenomenon being studied. For instance, an individual's income may be recorded in the data set as wage income; income from other sources, such as rental properties; support payments from the government, and the like. Aggregating income from all sources will develop a representative indicator for the individual income.

Often you need to transform variables from one type to another. It may be prudent to transform the continuous variable for income into a categorical variable where each record in the database is identified as low, medium, and high-income individual. This could help capture the non-linearities in the underlying behaviors.

## **4.) Storing Data**

The transformed data must be stored in a format that makes it conducive for data mining. The data must be stored in a format that gives unrestricted and immediate read/write privileges to the data scientist. During data mining, new variables are created, which are written back to the original database, which is why the data storage scheme should facilitate efficiently reading from and writing to the database. It is also important to store data on servers or storage media that keeps the data secure and also prevents the data mining algorithm from unnecessarily searching for pieces of data scattered on different servers or storage media. Data safety and privacy should be a prime concern for storing data.

## **5.) Mining Data**

After data is appropriately processed, transformed, and stored, it is subject to data mining. This step **covers data analysis methods**, including **parametric** and **non**-**parametric** methods, and **machine-learning algorithms**. A good starting point for data mining is data **visualization**. Multidimensional views of the data using the advanced graphing capabilities of data mining software are very helpful in developing a preliminary understanding of the trends hidden in the data set.

## **6.) Evaluating Mining Results**

After results have been extracted from data mining, you do a formal evaluation of the results. Formal evaluation could include testing the predictive capabilities of the models on observed data to see how effective and efficient the algorithms have been in reproducing data. This is known as an **"in-sample forecast"**. In addition, the results are shared with the key stakeholders for feedback, which is then incorporated in the later iterations of data mining to improve the process.

Data mining and evaluating the results becomes an iterative process such that the analysts use better and improved algorithms to improve the quality of results generated in light of the feedback received from the key stakeholders.

**Difference between BD, ML, DL, AI, DS...**

In data science, there are many terms that are used interchangeably, so let's explore the most common ones.

1.) 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.

2.) **Machine learning** is a subset of AI that uses computer algorithms to analyze data and make intelligent decisions based on what it is 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.

3.) **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 layer-deep and are the reason deep learning algorithms become more efficient as the data sets increase in volume, as opposed to other machine learning algorithms that may plateau as data increases.

4.) **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.

5.) **Artificial Intelligence** is **the simulation of human intelligence processes by machines, especially computer systems**. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

Also, it refers to the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

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

# **Applications of Data Science**

1.) 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'll 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.

2.) Personal assistants like Siri on Apple devices use data science to devise answers to the infinite number of questions end users may ask.

3.) 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.

4.) Netflix collects and analyzes massive amounts of data from millions of users, including which shows people are watching at what time a 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.

**THE REPORT STRUCTURE:**

* 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

1. Cover page
2. Table of contents
3. Executive Summary
4. Introductory section
5. Methodology section
6. Results section
7. Discussion section
8. Conclusion section
9. References
10. Acknowledgment

* The report should present a thorough analysis of the data and communicate the project findings.

**LANGUAGE OF DATA SCIENCE**

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.

**1.) PYTHON:**

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.

**Uses,**

Python is useful for many situations, including data science, AI and machine learning, web development, and IoT devices like the Raspberry Pi.

**Companies,**

Large organizations that use Python heavily include IBM, Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify, and Reddit.

**Applications,**

1. Python is a high-level general-purpose programming language that can be applied to many different classes of problems.

2. 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.

3. For data science, you can use Python's scientific computing libraries such as Pandas, NumPy, SciPy, and Matplotlib.

4. For artificial intelligence, it has TensorFlow, PyTorch, Keras, and Scikit-learn.

5. Python can also be used for Natural Language Processing (NLP) using the Natural Language Toolkit (NLTK).

**2.) R LANGUAGE:**

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?

* 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.

**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.

**Companies,**

R is popular in academia but companies that use R include IBM, Google, Facebook, Microsoft, Bank of America, Ford, TechCrunch, Uber, and Trulia.

**Applications,**

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

**3.) SQL LANGUAGE:**

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!

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.

The SQL language is subdivided into several language elements, including clauses, expressions, predicates, queries, and statements.

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.

**OTHERS..........**.

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

**4.) 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.

**5.) 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 alongwith 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

**6.) 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.

**7**.) 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.

**8.) Julia** was designed at MIT for high-performance numerical analysis and computational science.

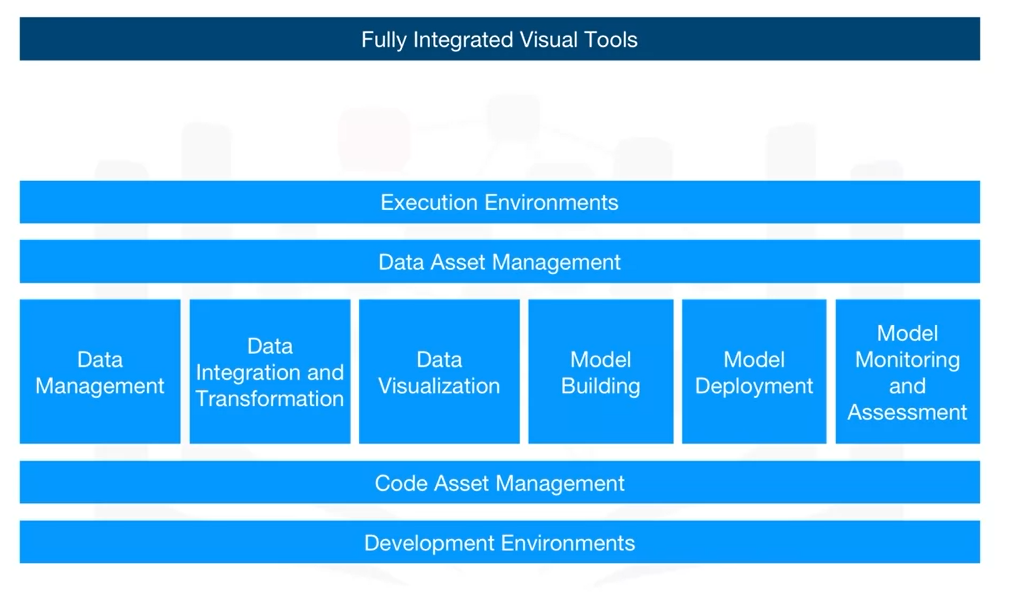
It provides speedy development like Python or R, while producing programs that run as fast as C or Fortran programs.

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.

**CATEGORIES OF DATA SCIENCE TOOLS**

1. **Data Management** is the process of persisting and retrieving data.
2. **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.
3. **Data Visualization** is part of an initial data exploration process, as well as being part of a final deliverable.
4. **Model Building** is the process of creating a machine learning or deep learning model using an appropriate algorithm with a lot of data.
5. **Model deployment** makes such a machine learning or deep learning model available to third-party applications.
6. **Model monitoring and assessment** ensures continuous performance quality checks on the deployed models. These checks are for accuracy, fairness, and adversarial robustness.
7. **Code asset management** uses versioning and other collaborative features to facilitate teamwork.
8. **Data asset management** brings the same versioning and collaborative components to data. Data asset management also supports replication, backup, and access right management.
9. **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.
10. **Execution environments** are tools where data preprocessing, model training, and deployment take place.
11. Finally, there is **fully integrated, visual tooling** available that covers all the previous tooling components, either partially or completely.



**OPEN SOURCE TOOLS FOR DATA SCIENCE STEPS MENTIONED ABOVE....**

**1. FOR DATA MANAGEMENT:**

The most widely used open source data management tools are

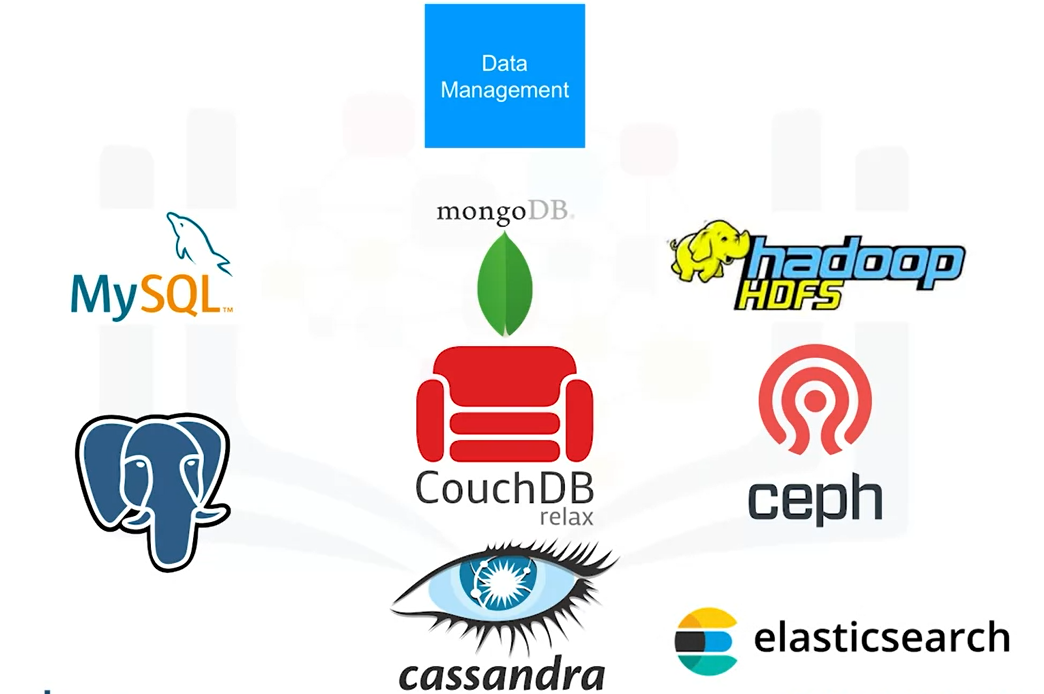
relational databases such as **MySQL** and **PostgreSQL**;

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.



**2. FOR DATA INTEGRATION AND TRANSFORMATION:**

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 themself 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:

1. **Apache AirFlow**, originally created by AirBNB;
2. **KubeFlow**, which enables you to execute data science pipelines on top of Kubernetes;
3. **Apache** **Kafka**, which originated from LinkedIn;
4. **Apache Nifi**, which delivers a very nice visual editor;
5. **Apache SparkSQL** (which enables you to use ANSI SQL and scales up to compute clusters of 1000s of nodes), and
6. **NodeRED**, which also provides a visual editor. NodeRED consumes so little in resources that it even runs on small devices like a Raspberry Pi.



**3.) FOR DATA VISUALIZATION:**

We have to distinguish between programming libraries where you need to use code and tools that contain a user interface.

Some of the tools used are

**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.



**4.) FOR MODEL DEPLOYMENT:**

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.js**



**5.) FOR MODEL MONITORING AND ASSESSMENT:**

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.

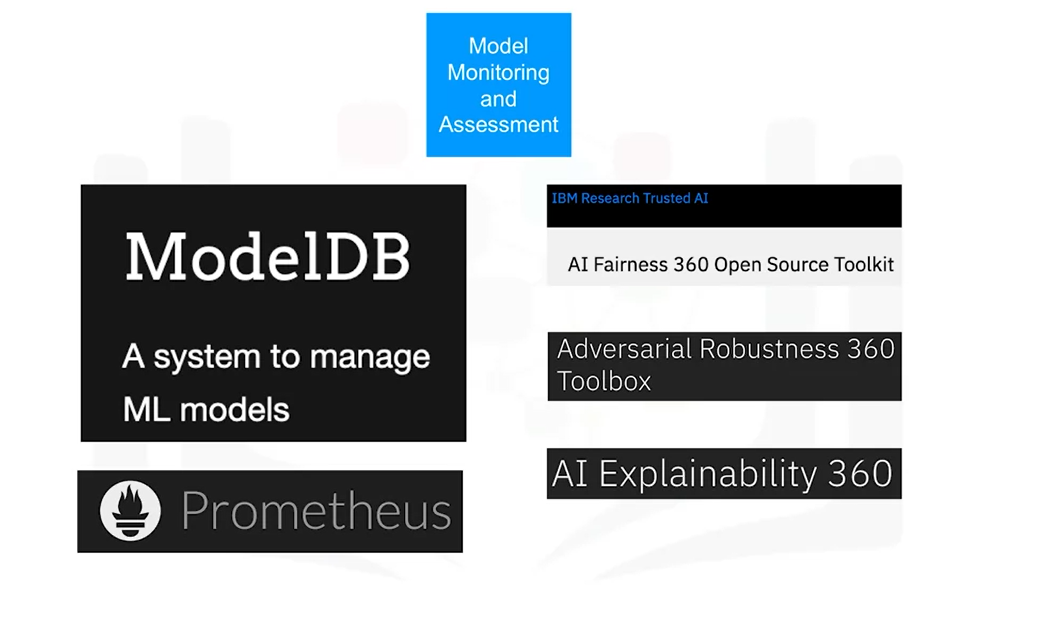
Machine learning models, especially neural-network-based deep learning models, can be subject to adversarial attacks, where an attacker tries to fool the model with manipulated data or by manipulating the model itself.

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 Toolki**t 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.



**6.) FOR CODE ASSET MANAGEMENT**

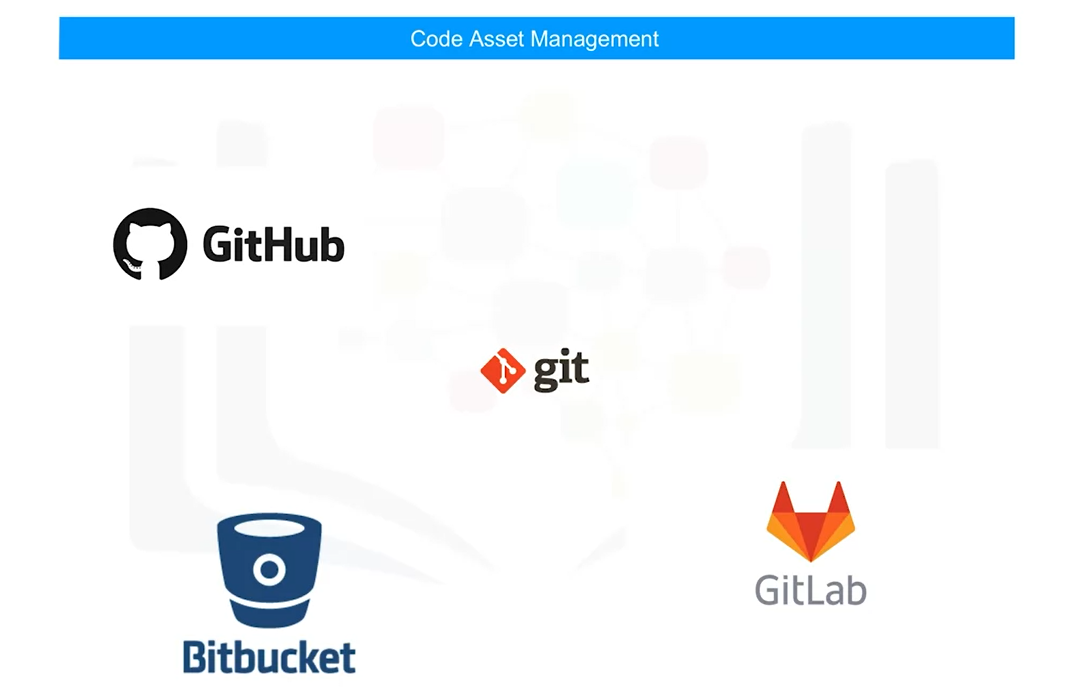
For code asset management – also referred to as version management or version control

– **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.**



**7.) FOR ASSET MANAGEMENT:**

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.



**8.) DEVELOPMENT ENVIRONMENTS:**

1. 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.

1. **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.

1. **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.

1. **Spyder** tries to mimic the behaviour 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.
2. 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.

1. 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.

1. 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.

**9.) FULLY INTEGRATED DATA SCIENCE TOOLS:**

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; these tasks include 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 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.

COMMERCIAL TOOLS AND CLOUD BASED TOOLS CAN BE VIEWED IN THE COURSERA VIDEOS.....

**LIBRARIES OF DATA SCIENCE..**

Libraries are a collection of functions and methods that enable you to perform a wide variety of actions without writing the code yourself.

Libraries usually contain built-in modules providing different functionalities that you can use directly; these are sometimes called “frameworks.” here are also extensive libraries, offering a broad range of facilities.

Python libraries:

1. Scientific Computing Libraries in Python
2. Visualization Libraries in Python
3. 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
4. Deep Learning Libraries in Python

**1.Scientific Computing Libraries in Python:**

**I.)** **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.

**II.) NumPy** libraries are based on arrays, enabling you to apply mathematical functions to these arrays. Pandas is actually built on top of NumPy

**2.Visualization Libraries in Python:**

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.

**I.)** 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.

**II.)** 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.

**3. High-Level Machine Learning and Deep Learning Libraries:**

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.

**4. Deep Learning Libraries in Python(low-level):**

**I.) 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.

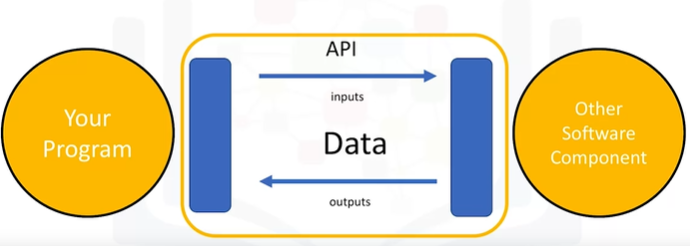
**II.) Pytorch** is used for experimentation, making it simple for researchers to test their ideas

**III.) 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.

**BASIC TERMINOLOGIES:**

**1. APPLICATION PROGRAMMING INTERFACES:**

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.



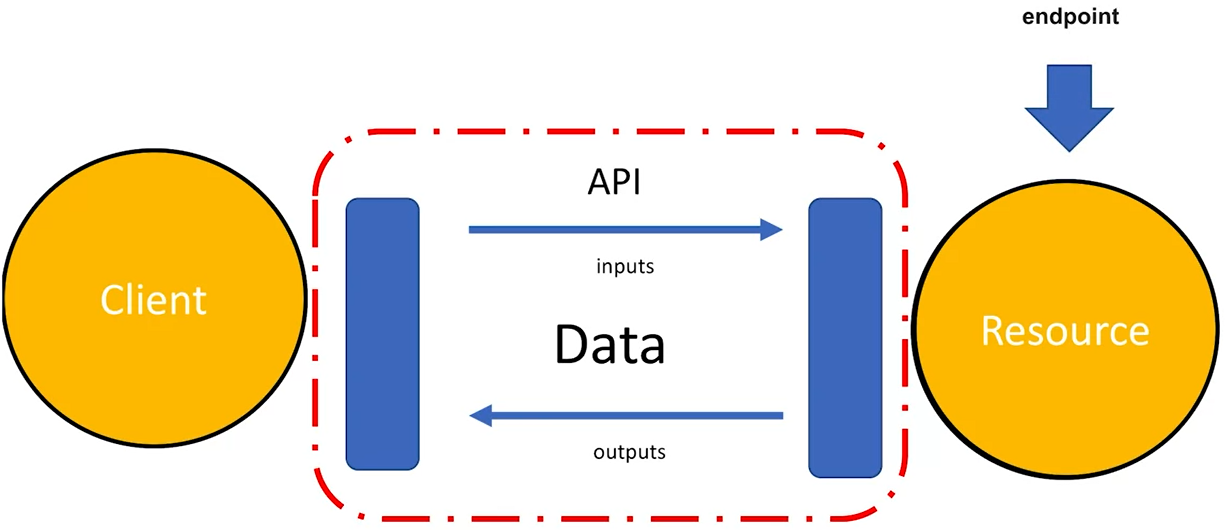
Consider the pandas library. Pandas is actually a set of software components, many of which are not even written in Python. You have some data. You have a set of software components. We use the pandas API to process the data by communicating with the other software components.

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.

**2.REST API’s:**

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.

Example:

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.

**3.) DATASET:**

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.

**I.)** 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.

**II.) 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.

**III.)** 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.

**4.) DATA OWNERSHIP:**

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.



**5.) DATA LICENSING:**

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.

**6.) 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.

**I.) SUPERVISED LEARNING:**

Here data is labelled and model is trained to make correct predictions.

**A.) REGRESSION**: Predict real numerical values.

Ex: home sales price, stock market price

**B.) CLASSIFICATION**: Classify things into categories.

Ex: email spam filters, image classification, fraud detection.

**II.) UNSUPERVISED LEARNING:**

Here data is not labelled and model tries to identify patterns without external help

Ex: Clustering and anomaly detection.

**III.) REINFORCEMENT LEARNING:**

Conceptually similar to human learning processes

Ex: A robot learning to walk; Chess, Go and other games of skill.

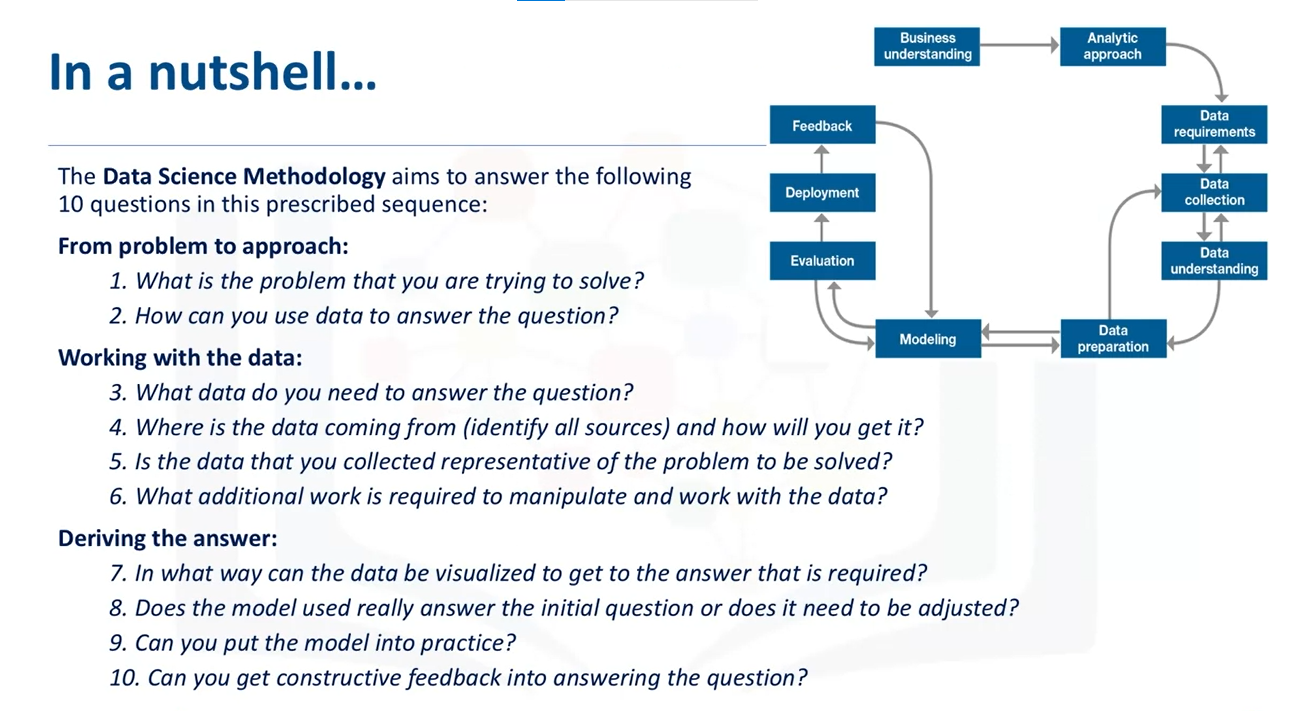
**IV.) DEEP LEARNING:**

Tries to loosely emulate how the human brain works. It requires typically very large datasets of labelled data and is compute intensive.

Built using frameworks such as Tensorflow, Pytorch, Keras or can even be built from scratch.

* APPLICATIONS:
  + Natural Language Processing
  + Image, audio and video analysis
  + Time series forecasting and much more

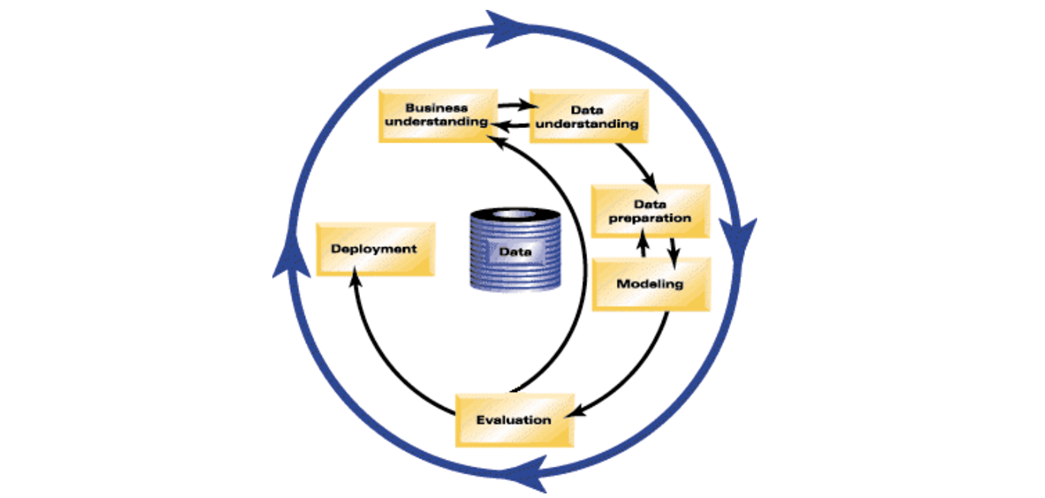
**DATA SCIENCE METHODOLOGY**



In data mining, the Cross Industry Process for Data Mining (CRISP-DM) methodology is widely used.

### What is **CRISP-DM**?

The CRISP-DM methodology is a process aimed at increasing the use of data mining over a wide variety of business applications and industries. The intent is to take case specific scenarios and general behaviors to make them domain neutral. CRISP-DM is comprised of six steps with an entity that has to implement in order to have a reasonable chance of success. The six steps are shown in the following diagram:



*Fig.1 CRISP-DM model,* [*IBM Knowledge Center, CRISP-DM Help Overview*](https://www.ibm.com/support/knowledgecenter/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html?utm_email=Email&utm_source=Nurture&utm_content=000026UJ&utm_term=10006555&utm_campaign=PLACEHOLDER&utm_id=SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DS0103EN-SkillsNetwork-20083987)

1. **Business Understanding** This stage is the most important because this is where the intention of the project is outlined. Foundational Methodology and CRISP-DM are aligned here. It requires communication and clarity. The difficulty here is that stakeholders have different objectives, biases, and modalities of relating information. They don’t all see the same things or in the same manner. Without clear, concise, and complete perspective of what the project goals are resources will be needlessly expended.
2. **Data Understanding** Data understanding relies on business understanding. Data is collected at this stage of the process. The understanding of what the business wants and needs will determine what data is collected, from what sources, and by what methods. CRISP-DM combines the stages of Data Requirements, Data Collection, and Data Understanding from the Foundational Methodology outline.
3. **Data Preparation** Once the data has been collected, it must be transformed into a useable subset unless it is determined that more data is needed. Once a dataset is chosen, it must then be checked for questionable, missing, or ambiguous cases. Data Preparation is common to CRISP-DM and Foundational Methodology.
4. **Modeling** Once prepared for use, the data must be expressed through whatever appropriate models, give meaningful insights, and hopefully new knowledge. This is the purpose of data mining: to create knowledge information that has meaning and utility. The use of models reveals patterns and structures within the data that provide insight into the features of interest. Models are selected on a portion of the data and adjustments are made if necessary. Model selection is an art and science. Both Foundational Methodology and CRISP-DM are required for the subsequent stage.
5. **Evaluation** The selected model must be tested. This is usually done by having a pre-selected test, set to run the trained model on. This will allow you to see the effectiveness of the model on a set it sees as new. Results from this are used to determine efficacy of the model and foreshadows its role in the next and final stage.
6. **Deployment** In the deployment step, the model is used on new data outside of the scope of the dataset and by new stakeholders. The new interactions at this phase might reveal the new variables and needs for the dataset and model. These new challenges could initiate revision of either business needs and actions, or the model and data, or both.

CRISP-DM is a highly flexible and cyclical model. Flexibility is required at each step along with communication to keep the project on track. At any of the six stages, it may be necessary to revisit an earlier stage and make changes. The key point of this process is that it’s cyclical; therefore, even at the finish you are having another business understanding encounter to discuss the viability after deployment. The journey continues.

Refer the jupyter notebooks for more details..

**Remaining...**

1. Github
2. Other IBM tools
3. Webscraping
4. Data analysis in Python Chi-square test
5. Data analysis in Python notebooks