

## Elective1----MCEL2012A : Data preparation and Analysis

### Unit I: Data Gathering and Preparation

# 1 .Defining A Data Science Problem

According to Cameron Warren, in his Towards Data Science article **Don't Do Data Science, Solve Business Problems**, “...*the number one most important skill for a Data Scientist above any technical expertise — [is] the ability to clearly evaluate and define a problem.*”

As a data scientist you will routinely discover or be presented with problems to solve. Your initial objective should be to determine if your problem is in fact a Data Science problem — and if so, what kind. There is great value in being able to translate a business idea or question into a clearly formulated problem statement. And being able to effectively communicate whether or not that problem can be solved by applying appropriate Machine Learning algorithms.

### Is it a Data Science problem?

A true data science problem may:

- Categorize or group data
- Identify patterns
- Identify anomalies
- Show correlations
- Predict outcomes

A good data science problem should be **specific and conclusive**. For example:

- *As personal wealth increases, how do key health markers change?*
- *Where in California do most people with heart disease live?*

Conversely, a **vague and unmeasurable** problem may not be a good fit for a data science solution. For example:

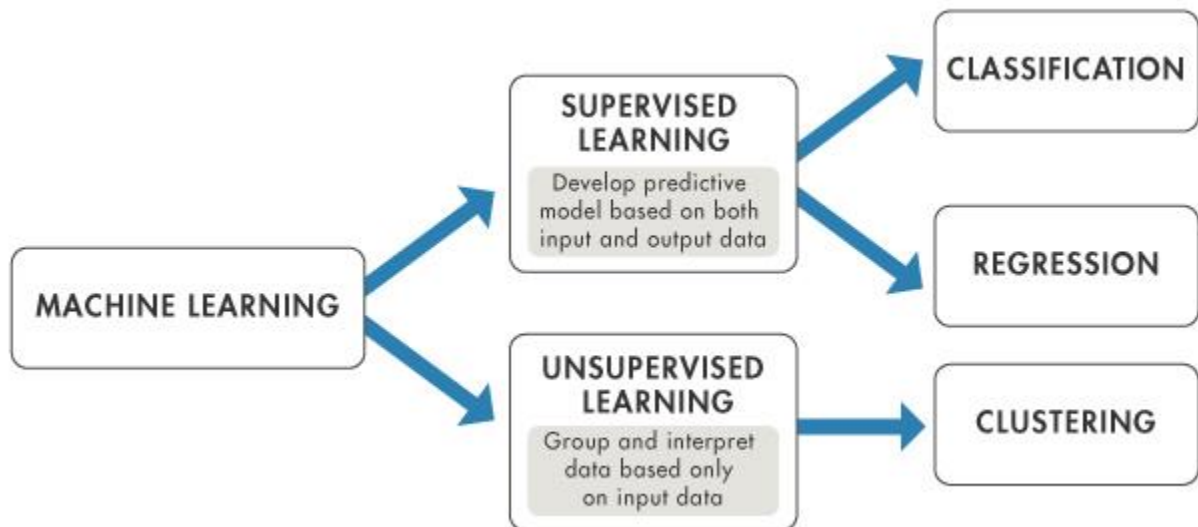
- *What is the link between finances and health?*
- *Are people in California healthier?*

### What type of Data Science problem is it?

Once you've decided that your problem is a good candidate for Data Science, you'll need to determine the type of problem you're working with. This step is necessary in order to know which type of Machine Learning algorithms can be effectively applied.

Machine Learning problems generally fall into one of two buckets:

- **Supervised** — *predicts future outputs based on labeled input & output data*
- **Unsupervised** — *finds hidden patterns or groupings in unlabeled input data*



*\*There is a third bucket (**Reinforcement Learning**) which is not covered in this post, but you can read about it [here](#).*

**Supervised** learning can be broken down into two additional buckets:

- **Classification** — *predicts discrete categorical response (ex: benign or malignant)*
- **Regression** — *predicts continuous numerical response (ex: \$200,000 home price or 5% probability of rain)*

## What are the use cases for each type of Machine Learning problem?

- **Unsupervised** (mainly considered “Clustering”) — market segmentation, political polling, retail recommendation systems, and more
- **Classification** — medical imaging, natural language processing, and image recognition, and more
- **Regression** — weather forecasting, voter turnout, and home sale pricing, and more

---

**Pro Tip:** *By using conditional logic to convert a continuous numerical response into a discrete categorical response, a Regression problem can be turned into a Classification problem! For example:*

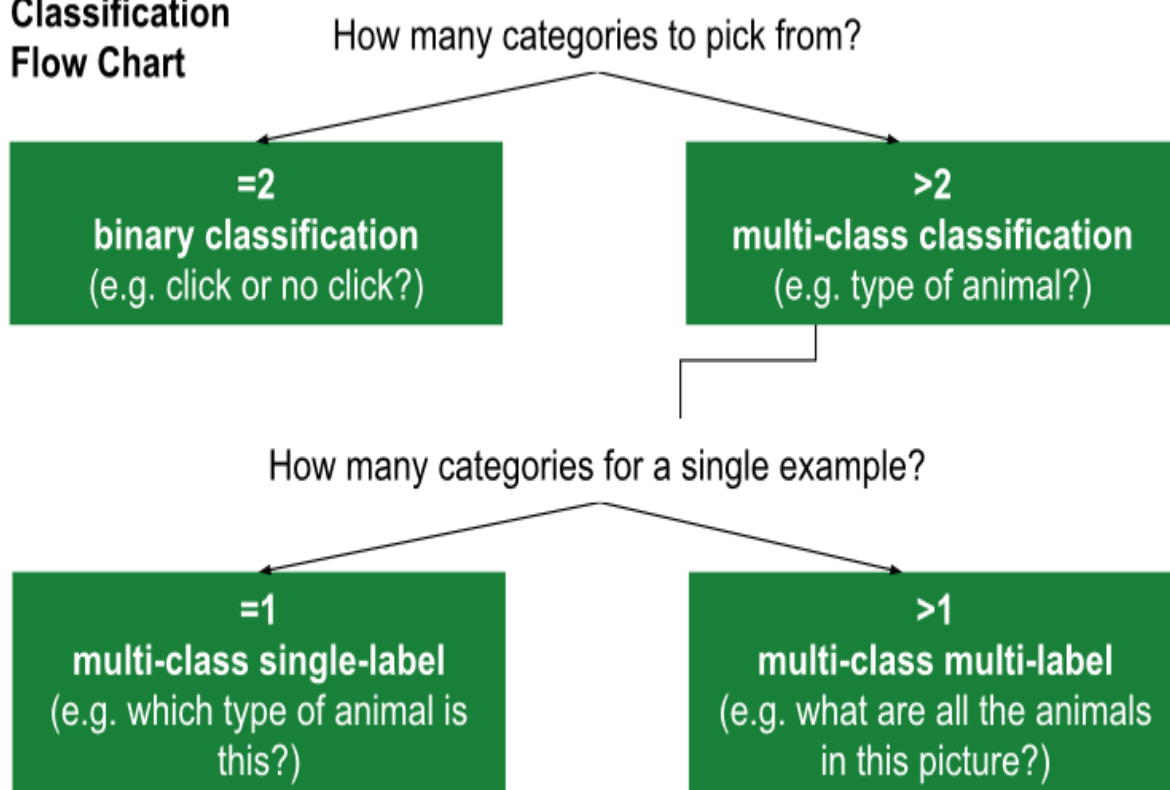
- **Problem:** *Estimate the probability that someone will vote.*
- **Regression response:** *60% probability*
- **Classification response:** *Yes (if Regression response is greater than 50%), No (if Regression response is less than 50%)*

---

## Getting granular on the subtype for your problem

Before honing in on your final problem definition, you'll want to get very specific about the Machine Learning subtype for your problem. Getting clear on the terminology will inform your decisions about which algorithms to choose. The diagrams below illustrate an example workflow for deciding on Classification subtypes (based on classes) and Regression subtypes (based on numerical values).

### Classification Flow Chart



## Regression Flow Chart

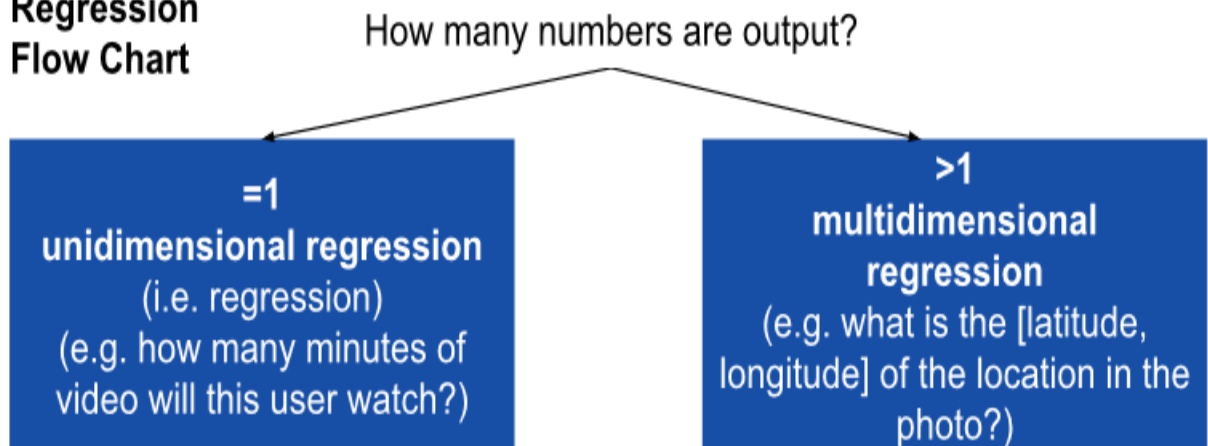


Image per [Google Developers](#)

## Finalizing the problem statement

Once you've determined the specific problem type, you should be able to clearly articulate a refined problem statement, including what the model will predict. For example:

*This is a multi-class classification problem, which predicts whether a medical image will be in one of three classes — {benign, malignant, inconclusive}.*

You should also be able to express a desired outcome or intended usage for the model prediction. For example:

*The ideal outcome is to provide healthcare providers with an immediate notification when a prediction is malignant or inconclusive.*

## Conclusion

A good data science problem will [aim to make decisions, not just predictions](#). Keep this objective in mind as you contemplate each problem you are faced with. In the example above, some action might be taken to reduce the number of *inconclusive* predictions, thereby avoiding the need for subsequent rounds of testing, or delaying needed treatment. Ultimately, the predictions from your model should empower your stakeholders to make informed decisions — and take action!

Data science and statistics are not magic. They won't magically fix all of a company's problems. However, they are useful tools to help companies make more accurate decisions and automate repetitive work and choices that teams need to make," [writes Seattle Data Guy](#), a data-driven consulting agency.

The questions that can be answered with the help of **data science fall under following categories:**

- **Identifying themes in large data sets:** Which server in my server farm needs maintenance the most?
- **Identifying anomalies in large data sets:** Is this combination of purchases different from what this customer has ordered in the past?
- **Predicting the likelihood of something happening:** How likely is this user to click on my video?
- **Showing how things are connected to one another:** What is the topic of this online article?
- **Categorizing individual data points:** Is this an image of a cat or a mouse?

Of course, this is by no means a complete list of all questions that data science can answer. Even if it were, data science is evolving at such a rapid pace that it would most likely be completely outdated within a year or two from its publication.

Now that we've established the types of questions that can be reasonably expected to be answered with the help of data science, it's time to lay down the steps most data scientists would take when approaching a new data science problem.

## The 5 steps on how to approach a new data science problem

### Step 1: Define the problem

First, it's necessary to **accurately define the data problem** that is to be solved. The problem should be **clear, concise, and measurable**. Many companies are too vague when defining data problems, which makes it difficult or even impossible for **data scientists to translate them into machine code**.

Here are some **basic characteristics of a well-defined data problem:**

- The solution to the problem is likely to have enough positive impact to justify the effort.
- Enough data is available in a usable format.
- Stakeholders are interested in applying data science to solve the problem.

## Step 2: Decide on an approach

There are many data science algorithms that can be applied to data, and they can be roughly grouped into the following families:

- **Two-class classification:** useful for any question that has just two possible answers.
- **Multi-class classification:** answers a question that has multiple possible answers.
- **Anomaly detection:** identifies data points that are not normal.
- **Regression:** gives a real-valued answer and is useful when looking for a number instead of a class or category.
- **Multi-class classification as regression:** useful for questions that occur as rankings or comparisons.
- **Two-class classification as regression:** useful for binary classification problems that can also be reformulated as regression.
- **Clustering:** answer questions about how data is organized by seeking to separate out a data set into intuitive chunks.
- **Dimensionality reduction:** reduces the number of random variables under consideration by obtaining a set of principal variables.
- **Reinforcement learning algorithms:** focus on taking action in an environment so as to maximize some notion of cumulative reward.

## Step 3: Collect data

With the problem clearly defined and a suitable approach selected, it's time to collect data. All **collected data should be organized in a log along with collection dates and other helpful metadata.**

It's important to understand that collected data is seldom ready for analysis right away. Most data scientists spend much of their time on **data cleaning**, which includes **removing missing values, identifying duplicate records, and correcting incorrect values.**

## Step 4: Analyze data

The next step after **data collection and cleanup is data analysis.** At this stage, there's a certain chance that the selected data science approach won't work. This is to be expected and accounted for. Generally, it's recommended **to start with trying all the basic machine learning approaches** as they have fewer parameters to alter.

There are many excellent [open source data science libraries](#) that can be used to **analyze data.** Most data science tools are written in Python, Java, or C++.

“Tempting as these cool toys are, for most applications the smart initial choice will be to pick a much simpler model, for example using [scikit-learn](#) and modeling techniques like simple logistic regression,”

– [advises Francine Bennett](#), the CEO and co-founder of Mastodon C.

## Step 5: Interpret results

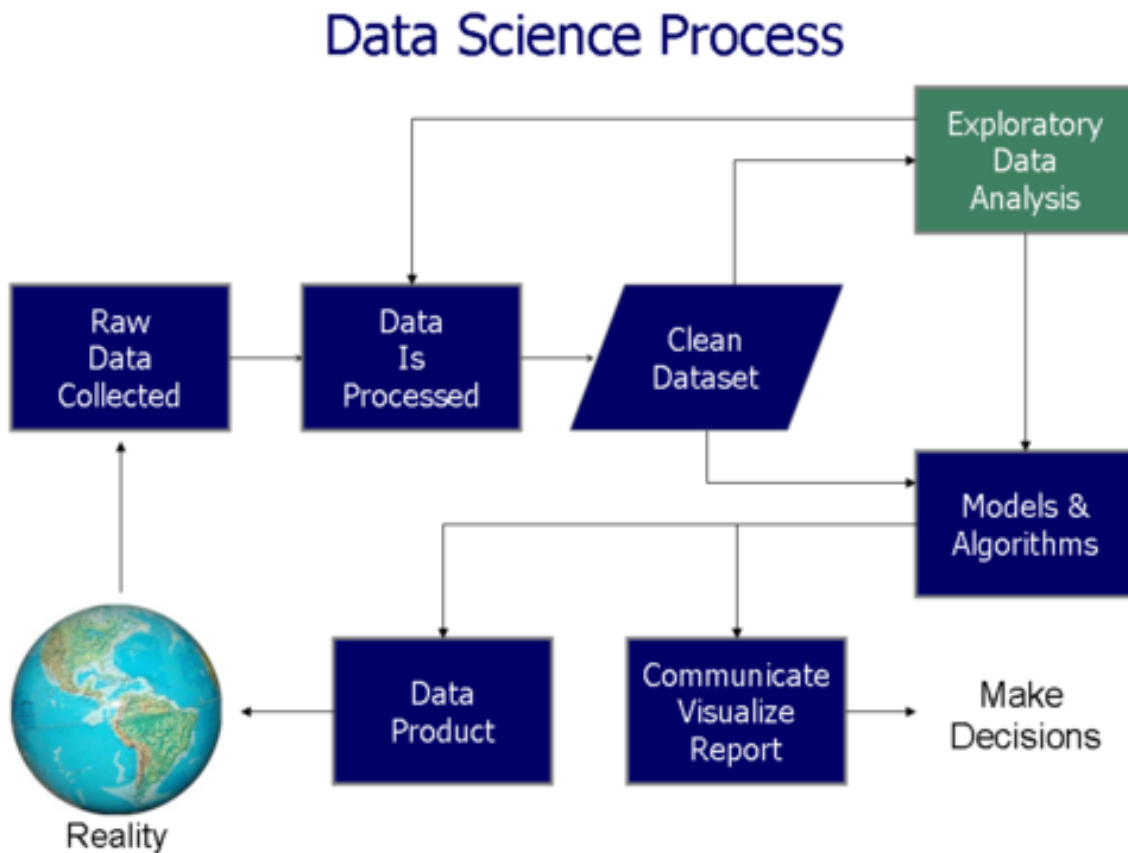
After data analysis, it's finally **time to interpret the results**. The most important thing to consider is **whether the original problem has been solved**. You might **discover that your model is working but producing subpar results**. One way how to deal with this is to add more data and keep retraining the model until satisfied with it.

### Conclusion

Most companies today are drowning in data. The global leaders are already using the data they generate to gain competitive advantage, and others are realizing that they must do the same or perish. While transforming an organization to become [data-driven](#) is no easy task, the reward is more than worth the effort.

The 5 steps on how to approach a new data science problem we've described in this article are meant to illustrate the general **problem-solving mindset** companies must adopt to successfully face the challenges of our current data-centric era.

### 3. Data science process



**Fig 1:** Data Science Process, credit: Wikipedia

## Step 1: Frame the problem

The first thing you have to do **before you solve a problem is to define exactly what it is**. You need to be able to translate data questions into something actionable.

You'll **often get ambiguous inputs from the people who have problems**. You'll have to develop the intuition to turn scarce inputs into actionable outputs—and to ask the questions that nobody else is asking.

Say **you're solving a problem for the VP Sales of your company**. You should **start by understanding their goals** and the **underlying why behind their data questions**. Before you can start thinking of solutions, you'll **want to work with them to clearly define the problem**.

A great way to do this is to ask the right questions.



You should then **figure out what the sales process looks** like, and who the customers are. You need as much context as possible for your numbers to become insights.

You should **ask questions** like the following:

1. Who are the customers?
2. Why are they buying our product?
3. How do we predict if a customer is going to buy our product?
4. What is different from segments who are performing well and those that are performing below expectations?
5. How much money will we lose if we don't actively sell the product to these groups?

In response to your questions, the **VP Sales might reveal that they want to understand why certain segments of customers have bought less than expected**. Their end goal might be to determine whether to continue to invest in these segments, or de-prioritize them. You'll want to tailor your analysis to that problem, and unearth insights that can support either conclusion.

It's important that at the end of this stage, you have all of the information and context you need to solve this problem.

## Step 2: Collect the raw data needed for your problem

Once you've defined the problem, you'll **need data to give you the insights needed to turn the problem around with a solution**. This part of the process involves thinking through what data you'll need and finding ways to get that data, whether it's **querying internal databases**, or **purchasing external datasets**.

You might find out that your company stores all of their sales data in a CRM or a customer relationship management software platform. You can export the CRM data in a CSV file for further analysis.

## Step 3: Process the data for analysis

Now that you have all of the raw data, **you'll need to process it before you can do any analysis**. Oftentimes, data can be quite messy, especially if it hasn't been well-maintained. You'll see errors that will corrupt your analysis: values set to null though they really are zero, duplicate values, and missing values. It's up to you to go through and check your data to make sure you'll get accurate insights.

You'll want to check for the following common errors:

1. Missing values, perhaps customers without an initial contact date

2. Corrupted values, such as invalid entries
3. Timezone differences, perhaps your database doesn't take into account the different timezones of your users
4. Date range errors, perhaps you'll have dates that makes no sense, such as data registered from before sales started

You'll need to look through aggregates of your file rows and columns and sample some test values to see if your values make sense. If you detect something that doesn't make sense, you'll need to remove that data or replace it with a default value. You'll need to use your intuition here: if a customer doesn't have an initial contact date, does it make sense to say that there was NO initial contact date? Or do you have to hunt down the VP Sales and ask if anybody has data on the customer's missing initial contact dates?

Once you're done working with those questions and cleaning your data, you'll **be ready for exploratory data analysis (EDA)**.

## Step 4: Explore the data

When your data is clean, you'll should start playing with it!

The difficulty here isn't coming up with ideas to test, it's coming up with ideas that are likely to turn into insights. You'll have a fixed deadline for your data science project (your VP Sales is probably waiting on your analysis eagerly!), so you'll have **to prioritize your questions**.

You'll have to look at some of the most interesting patterns that can help explain why sales are reduced for this group. You might notice that they don't tend to be very active on social media, with few of them having Twitter or Facebook accounts. You might also notice that most of them are older than your general audience. From that you can begin to trace patterns you can analyze more deeply.

## Step 5: Perform in-depth analysis

This step of the process is **where you're going to have to apply your statistical, mathematical and technological knowledge and leverage all of the data science tools** at your disposal to crunch the data and find every insight you can.

In this case, you might have **to create a predictive model that compares your underperforming group with your average customer**. You might **find out that the age and social media activity are significant factors in predicting who will buy the product**.

If you'd asked a lot of the right questions while framing your problem, you might realize that the company has been concentrating heavily on social media marketing efforts, with messaging that is aimed at younger audiences. You would know that certain demographics prefer being reached by telephone rather than by social media. You begin to see how the way the product has been has

been marketed is significantly affecting sales: maybe this problem group isn't a lost cause! A change in tactics from social media marketing to more in-person interactions could change everything for the better. This is something you'll have to flag to your VP Sales.

You can now combine all of those qualitative insights with data from your quantitative analysis to craft a story that moves people to action.

## Step 6: Communicate results of the analysis

It's important that the VP Sales understand why the **insights you've uncovered are important**. Ultimately, you've been called upon to create a solution throughout the data science process. **Proper communication** will mean the difference between **action** and **inaction** on your proposals.

You need to craft a compelling story here that ties your data with their knowledge. You start by **explaining the reasons behind the underperformance of the older demographic**. You tie that in with the answers your VP Sales gave you and the insights you've uncovered from the data. Then you move to concrete solutions that address the problem: we could shift some resources from social media to personal calls. You tie it all together into a narrative that solves the pain of your VP Sales: she now has clarity on how she can reclaim sales and hit her objectives.

She is now ready to act on your proposals.

Throughout the data science process, your day-to-day will vary significantly depending on where you are—and you will definitely receive tasks that fall outside of this standard process! You'll also often be juggling different projects all at once.

It's important to understand these steps if you want to systematically think about data science, and even more so if you're looking to start a career in data science.

# Data Science vs Business Intelligence, Explained

Defining the terms data science and business intelligence -- and the relationship between them -- has long been the subject of heated debate. Although these terms are related, failing to grasp the separate and distinct concepts behind them can have significant consequences.

For example, hundreds of thousands of data science (DS) and business intelligence (BI) jobs will open up in the next few years. The pool of candidates can seem impossibly small or surprisingly large depending on how relevant and useful you judge the required skills of each position to be. Can a BI expert successfully transition to a DS role? Is DS important to BI positions?

In 2021, business executives are going to evaluate billions of dollars of new projects. What gets the green light and what gets shelved will be affected by how executives and their teams understand and define the two terms. Project champions are quick to attach industry buzzwords to their projects to ride the latest trend, but are such projects just a rebranding of older ideas?

Figure 1 offers one perspective on the BI and DS landscape:

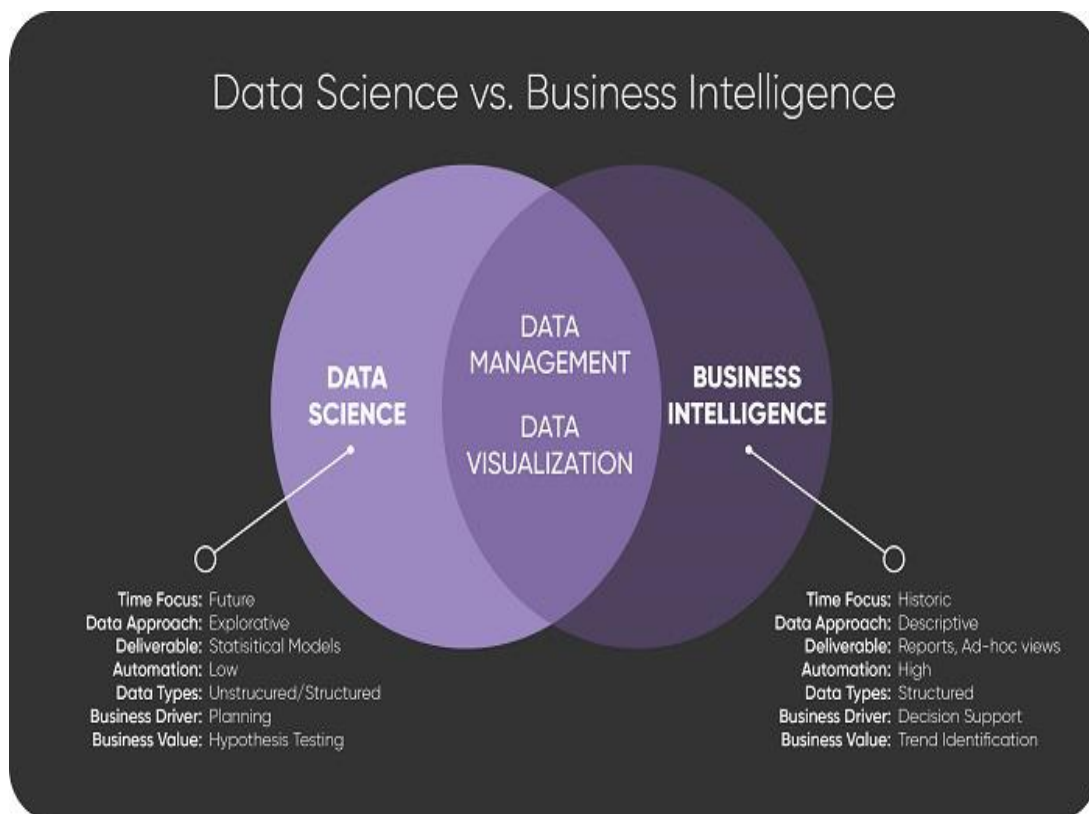


Fig. 1: The BI and data science landscape.

## The Foundations Haven't Changed

Data management and data visualization are still at the core of any effort to understand and plan a business. These involve technologies and processes to capture, clean, standardize, integrate, visualize, and secure data in a high-performing way. Excel is not enough. A pretty dashboard is not enough. You must commit long term to preserve data as an asset, and you need discipline to build and maintain data lake and data warehouse environments.

The crucial point is that any DS or BI initiative that does not have a solid data foundation will be unsustainable. Any processes that are built on manual, inconsistent processes will be slow, untrustworthy, and resource intensive. Eventually they need to mature with professional IT assistance, or they will fall apart under their own weight.

## Business Intelligence

Since the 1980s, nearly every company has tried to use computers and databases to manage and understand their historical data. However, here is the thing -- after almost 40 years, nobody has truly mastered it. Every year, companies add or replace software systems, and the IT department can rarely keep up, so enterprises end up constantly prioritizing projects to see which data gets attention and which data gets ignored (sorry, marketing department).

You will recognize business intelligence by its charts, dashboards, database diagrams, and data integration projects. It is expensive and frustrating -- but indispensable.

BI has a permanent advantage over DS because it has concrete data points; few, simple assumptions; self-explanatory metrics; and automated processes. Furthermore, BI will never go away. It will always be a work in progress because you will never stop changing your business or upgrading and replacing the source systems.

## Data Science

Looking in the rearview mirror of data is important and helpful, but it's limited and will never get you where you want to go. At some point you need to look ahead. BI needs to be accompanied by data science.

DS is a complicated, sophisticated form of planning and optimization. Examples include:

- Predicting in real time which product a customer is most likely to buy

- Forming a weighted network between business micro events and micro responses so that decisions can be made without human intervention, then updating that network with every outcome so that it learns as it acts
- Forecasting at the SKU level, by day, with every sale
- Identifying and predicting rare events, such as credit card fraud, and sending automatic notifications to customers and/or staff
- Creating clusters of customers based on dozens of attributes and behavior, then targeting them with custom messaging

Where traditional planning is done in discrete, human-directed sessions, DS techniques should result in planning and optimization steps that are embedded into software and run as part of automated processes. The model is trained using historical data, setting aside a subset of data to validate the accuracy of prediction. If the results are promising, then the model is deployed and monitored, often using BI reports.

Finding business sponsorship for DS projects is a challenge because the techniques are difficult to explain and visualize. The projects are difficult to manage and often involve unstructured or semistructured data, complex assumptions, statistical models, exploratory projects that come to dead ends, and limited or confusing visualizations.

As a result, pilot projects can be slow to start and frustrating to sustain. The results you achieve may be sporadic because of the unpredictability of your work. Predicting the future is never going to be simple.

Although BI projects are often completed by a single person, DS projects require extensive cooperation between employees who don't usually speak the same language, including data engineers, statisticians, business experts, and software developers. The competencies of each position require many years to master. Data scientists often have deep expertise in statistics but only elementary software development skills and limited business expertise. DS teams need to partner with IT and business departments to create truly integrated solutions.

## **5 Cloud Computing, Data Science and ML Trends in 2020–2022: The battle of giants**

### **Introduction**

I am going to dedicate a series of articles to the insights from the data collected in Kaggle's survey of 'State of Data Science and Machine Learning 2020' (<https://www.kaggle.com/c/kaggle-survey-2020>).

The survey covered a lot of diverse topics, and each of them deserves a separate post to discuss the respective trends.

*Notes:*

- Kaggle ([www.kaggle.com](https://www.kaggle.com)) is a global community made up of data scientists and machine learners from all over the world with a variety of skills and backgrounds. The community has around 3 million active members. Although it is not rigorously representative of the entire population of Data Science and ML professionals across the globe from the sociological perspective, it still constitutes the significant fraction of the practitioners and professionals in the field. Therefore, the results of the survey can really draw the projections of where the Data Science and AI/ML industry is likely to evolve in the next couple of years.
- You can check the repo per <https://github.com/gvyshnya/state-of-data-science-and-ml-2020> to see how every insight discussed in this post has been discovered.

## Battle of Giants

In this post, we are going to look at the popularity of cloud computing platforms and products among the data science and ML professionals participated in the survey. In particular, it will cover

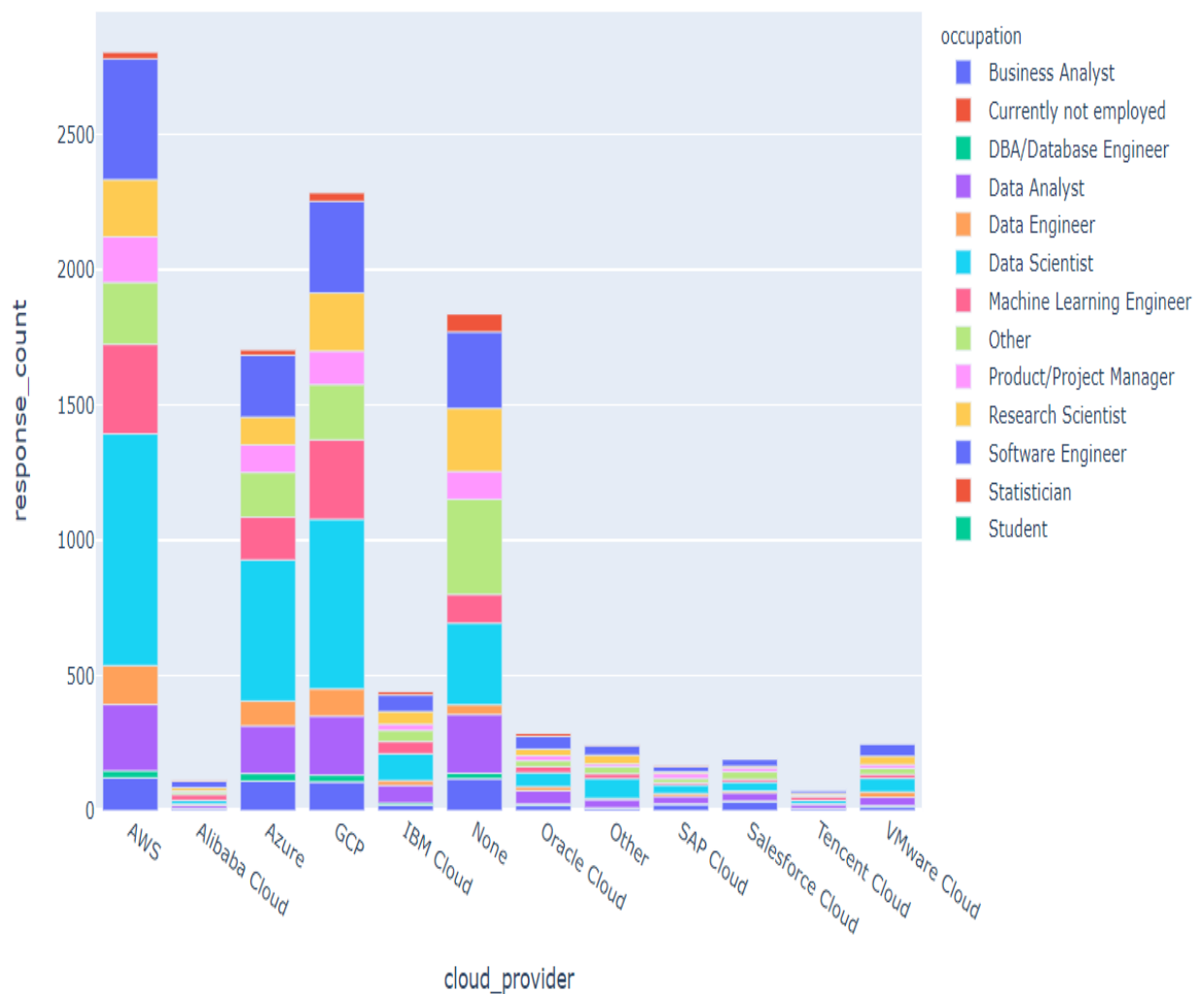
- Cloud Platforms usage
- Cloud Computing products usage
- Cloud ML products usage
- BigData platforms
- BI tools (mostly, the cloud-based ones)

The line of the narrative in this chapter will be often attached to the good news and opportunities for the top three cloud service providers in the market as follows

- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure Cloud (MS Azure)

*Note:*

- Survey organizers defined non-professionals as students, unemployed, and respondents that have never spent any money in the cloud. Everybody else is considered to be a professional





Also, it is notable that 'None' category slightly exceeds the size of MS Azure bar, and it means the market may not be saturated with the cloud service provider offerings.

We also see professionals with 3–5 years and 5–10 years of programming experience to be the largest group of users of the top 3 cloud service providers. More senior professionals (with 10+ years of experience) are less represented within the cloud service users on each of top 3 platforms (depending on the marketing priorities, special actions to educate such seniors could help to get the better spread of the cloud services).

As we can see, the majority of top three cloud service provider users fit into the roles below

- Data Scientists
- Software Engineers

The third position is held by

- ML Engineers (AWS, GCP)
- Data Analysts (MS Azure)

As noted earlier, 'Other' occupation group is too big itself, and it might be worth breaking it into more granular categories in the future surveys. As we see, 'Other' group takes a significant fraction of each cloud service platform users (although it is never seen in the top three list for any of the platforms).

In terms of the user occupation and programming experience, all of the top three Cloud Service Providers share the same trends below

- Data Scientists with 3–5 and 5–10 years of programming experience are the top user groups for AWS within the survey respondents
- In Software Engineer, ML Engineer, and Data Analyst groups, professionals with 3–5 and 5–10 years of experience predominate
- In Research Scientist, Data Engineer, DBA, Statistician and Other groups, professionals with 10+ years of experience are the biggest fraction of the users
- In Product/Project Management group, professionals with 5+ years of experience are the biggest fraction of the users
- In Business Analyst group, we see the users with 1–2 years of experience to dominate

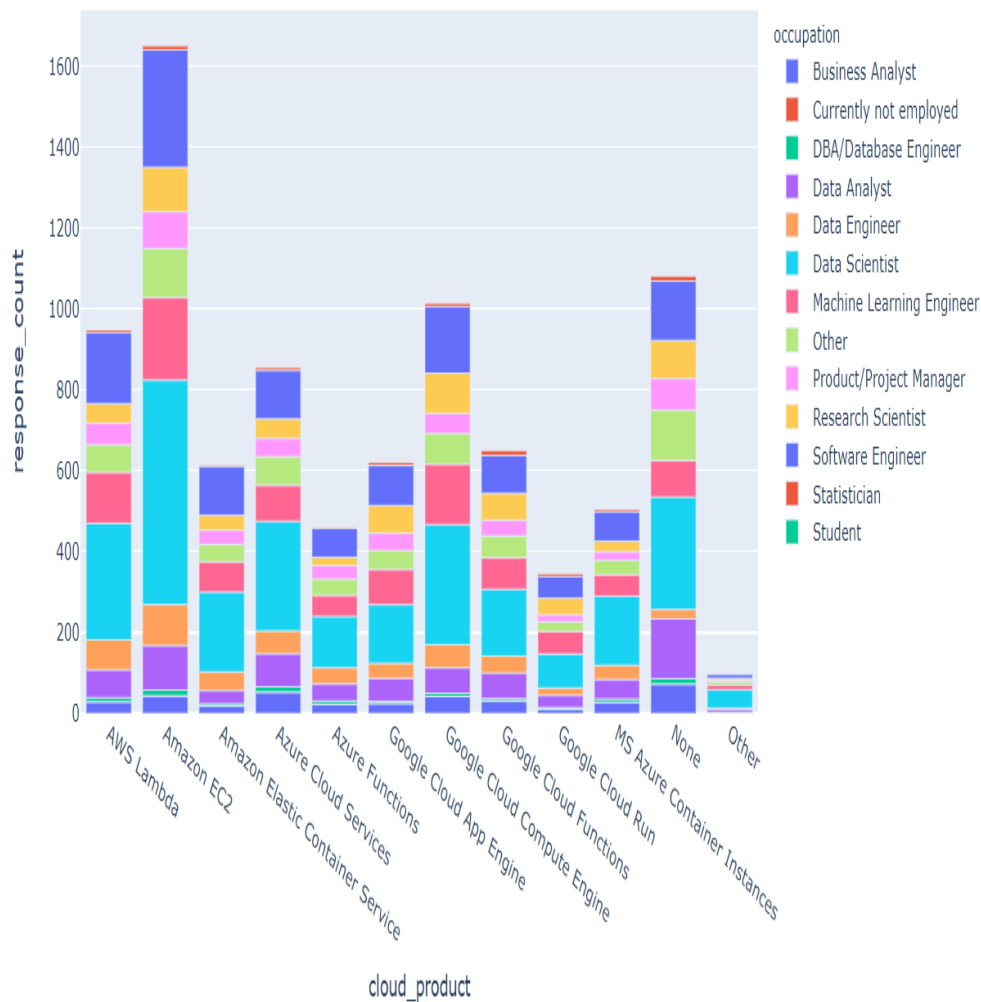
In terms of the organizational environments, the most Data Science and ML professionals using Cloud Services can be found in

- Organizations with 0–49 employees, having 1–2 workers dedicated to Data Science workloads
- Organizations with 10000+ employees, having 20+ workers dedicated to Data Science workloads

So we can conclude that AWS, GCP, and MS Azure tightly compete on the same types of the organizations/users.

## Usage of Cloud Computing Products

Kaggle 2020 Survey: Usage of Cloud Computing Products by Occupation



We find that

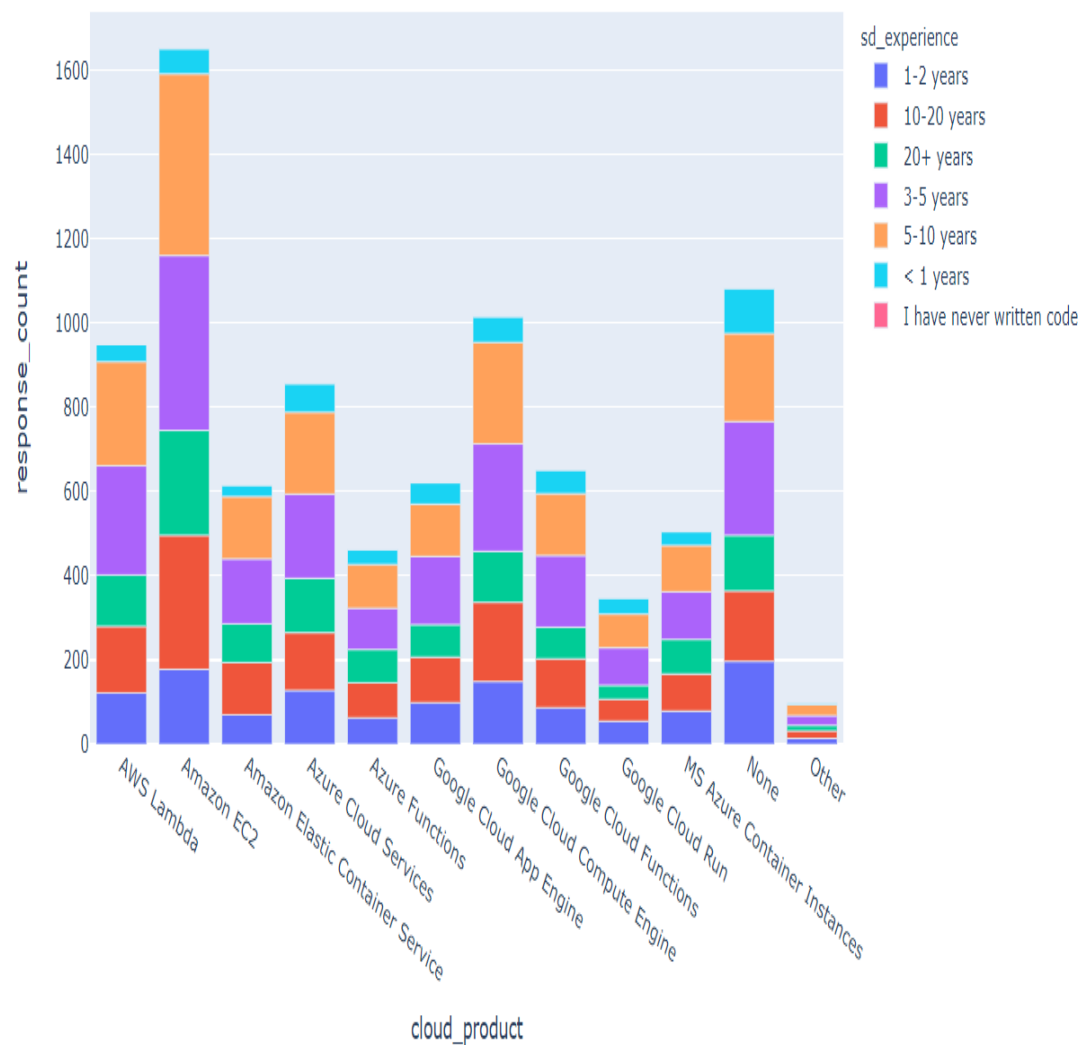
- in the segment of cloud computing engines, Amazon EC2 is more popular than its rivals from Google (Google Cloud Computing Engine) and MS Azure (Azure Cloud Services)
- in the segment of cloud functions, AWS Lambda is more popular than its rivals from Google (Google Cloud Functions) and MS Azure (Azure Functions)
- in the segment of cloud container runners, Amazon Elastic Container Service is more popular than its rivals from Google (Google Cloud Run) and MS Azure (MS Azure Container Instances)
- Google holds the second place in cloud computing engine and cloud function segments, and it is on the third place in the cloud container runner segment
- there is a huge pool of responses with 'None', and it is most likely to indicate the entire market of cloud computing applications is not saturated yet

In terms of user roles, the most users of every cloud computing product above hold the roles below (top to bottom)

- Data Scientists
- Software Engineers
- ML Engineers
- Data Analysts

## Usage of Cloud Computing Products by Programming Experience

Kaggle 2020 Survey: Usage of Cloud Computing Products by Programming Experience



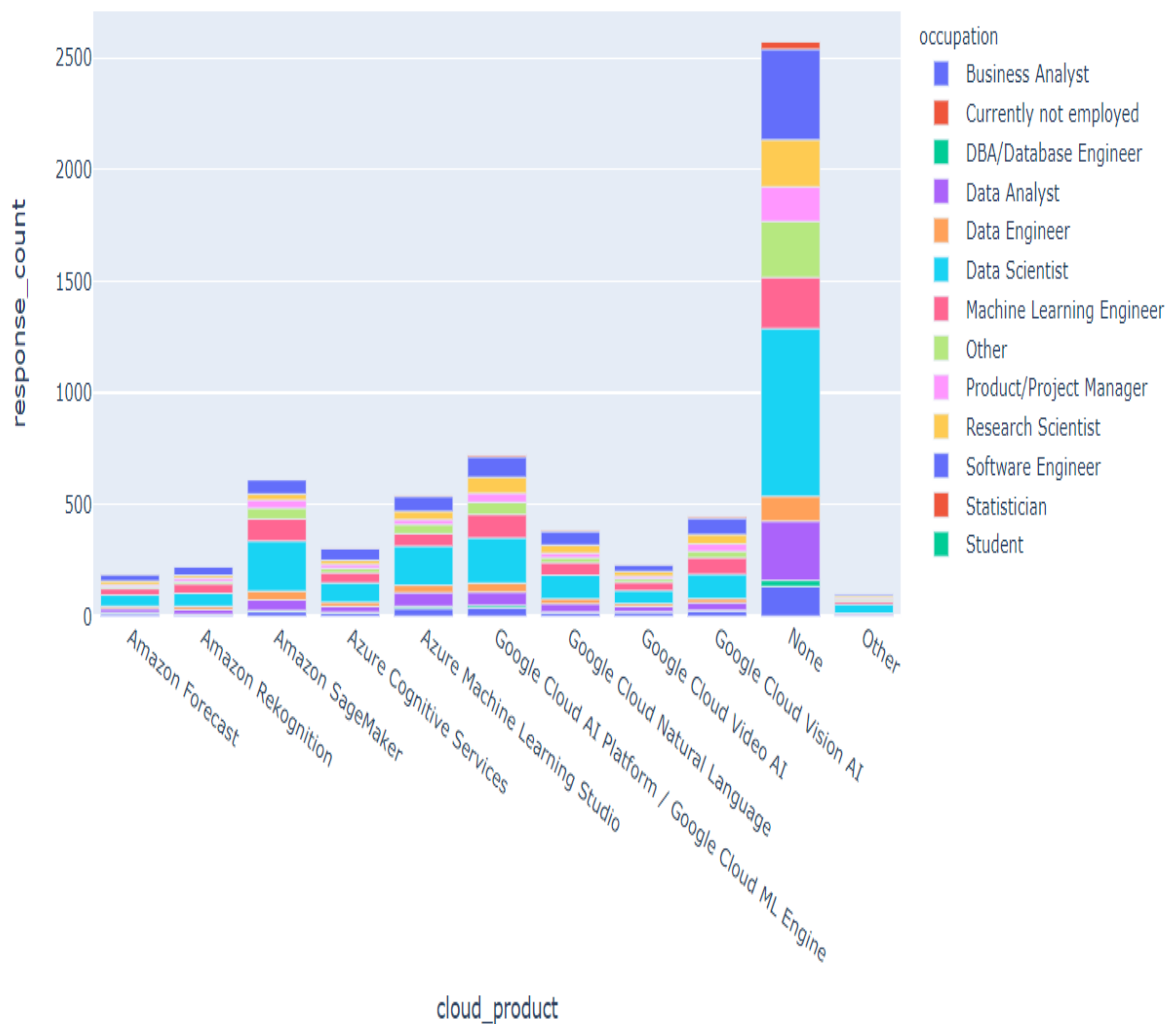
In addition to the insights above, we see that the top number of cloud computing product users fall into the following clusters in terms of their programming experience

- 5–10 years of experience
- 3–5 years of experience
- 10–20 years of experience

Juniors and super-seniors (20+ years of programming experience) seem to be less covered by the respective knowledge/skills.

## Usage of Cloud ML Products

## Kaggle 2020 Survey: Usage of Cloud ML Products by Occupation



We find that

- Google Cloud AI Platform / Google Cloud ML Engine leads the ML cloud products usage 'nomination
- the second and third best are Amazon SageMaker and Azure Machine Learning Studio, respectively
- Data Scientists are the top users of cloud ML products (for every product investigated)
- There is a huge chunk of responders who indicated they do not use cloud ML products at all — it indicates the market is under-saturated, and there is a good growth potential, subject to resolving the marketing and end-user barriers on the way

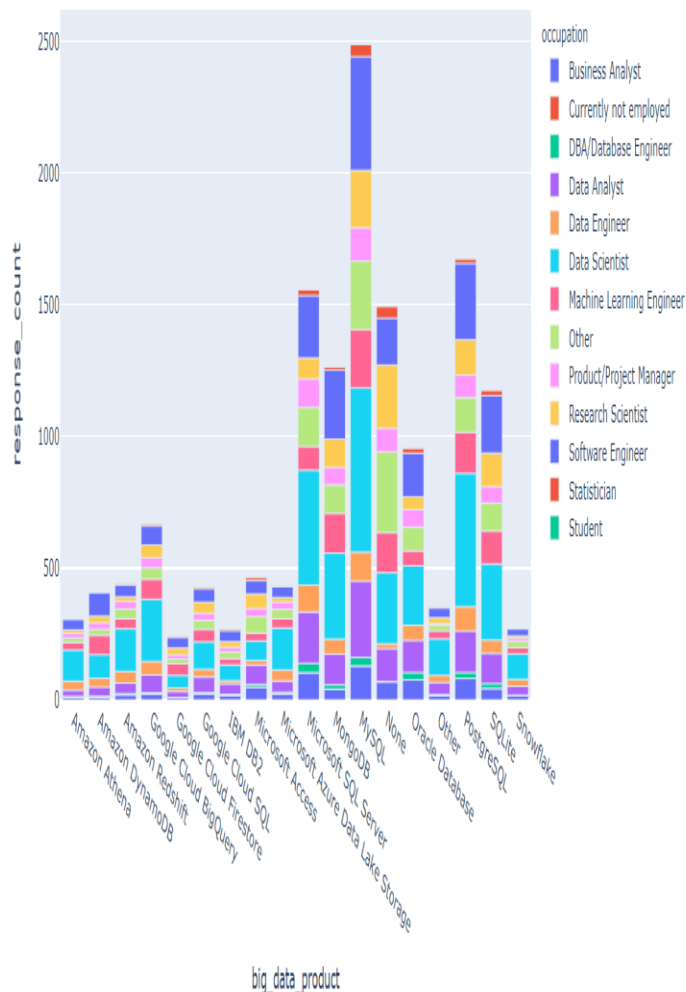
We find that the majority of organizations in every size category does not use any Cloud ML Products at the moment.

For the tiny fraction of those who use them, there are interesting insights as follows

- in small organizations (0–49 employees), Google Cloud AI Platform / Google Cloud ML Engine dominates
- in the middle-sized organizations (50–249 employees), Google Cloud AI Platform / Google Cloud ML Engine and Amazon SageMaker titly
- for companies of bigger size (250+ employees), the size of Data Science team is often correlated with the preferred Cloud ML Product (smaller teams sticks to Google Cloud AI Platform / Google Cloud ML Engine more, and Data Science teams with 20+ headcount are more inclined to use Amazon SageMaker )

## Usage of Big Data Products By Occupation

Kaggle 2020 Survey: Usage of Big Data Products by Occupation



We find that

- Overall top 3 list is constituted by three relational DBMS platforms (MySQL, PostgreSQL, MS SQL Server)
- MongoDB, a non-relational database platform, takes position 4 in the list
- Other relational DBMS platforms in the list (Oracle, IBM DB2, SQLite) are behind MongoDB
- In the segment of truly cloud-based Big Data products, Google BigQuery overruns its Amazon and MS Azure competitors ( Amazon Redshift, Amazon Athena, Amazon DynamoDB, and Microsoft Azure Data Lake Storage)
- Google Cloud SQL instances are still less popular then 'native' relational database instances for MySQL and PostgreSQL

- MS Access is still in use in the industry
- Data Scientists are the top users of each product in this list

### **Big Data Product Usage Patterns by User Occupation and Programming Experience**

We find that

- MySQL and PostgreSQL are the most popular database management platforms within each occupation
- MongoDB is quite popular with Software Engineers (although less popular than MySQL and PostgreSQL)

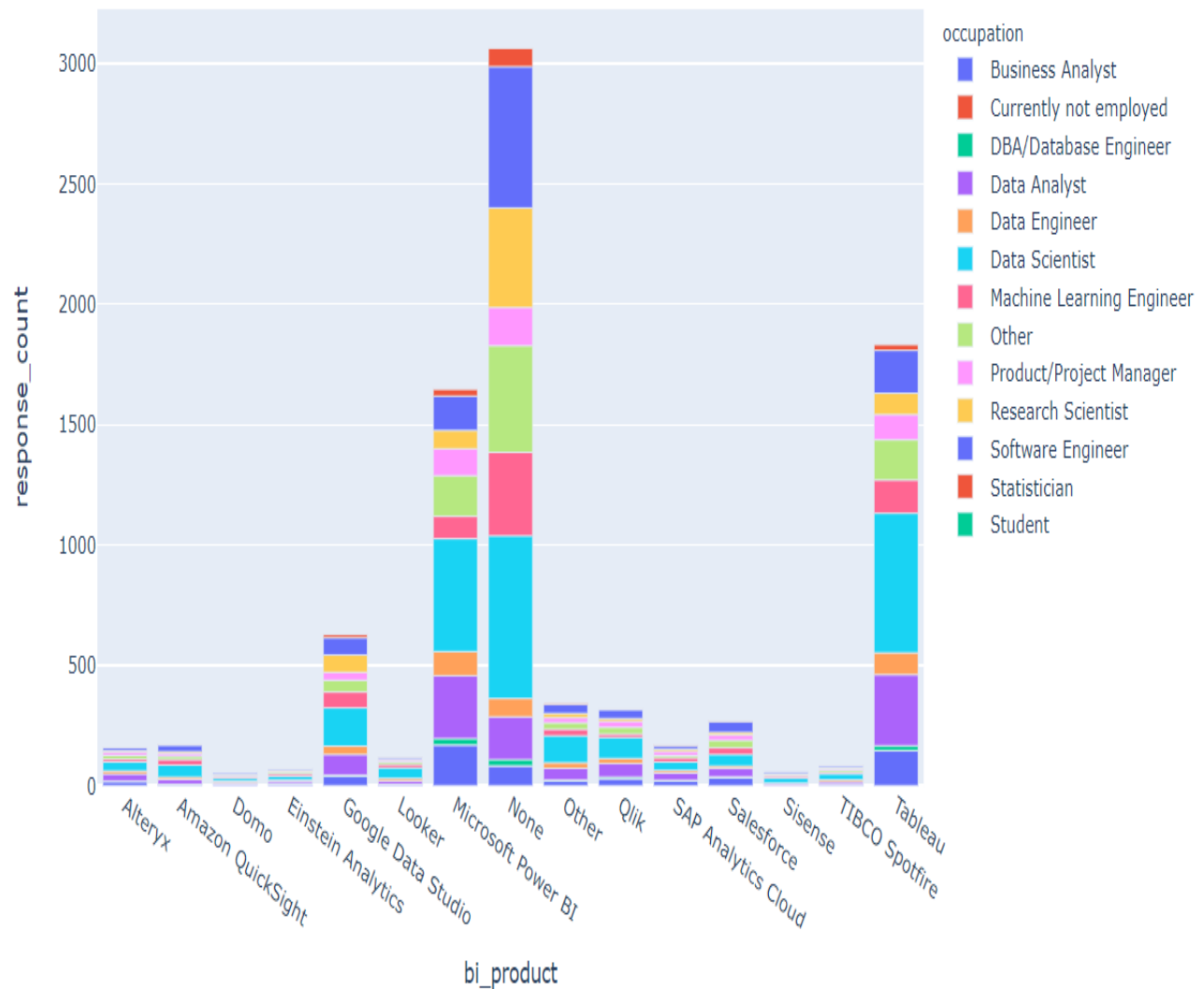
### **Big Data Product Usage Patterns by Organization Size and Data Science Capacity**

We find that

- Almost all of organizations except from the extra-large ones address their data management needs with MySQL, PostgreSQL, and MongoDB the most
- Extra-large organizations (with 10000+ employees) prefer to work with MySQL, MS SQL Server, Oracle, and PostgreSQL

## Usage of BI Tools

Kaggle 2020 Survey: Usage of BI Products by Occupation



We find that

- Tableue and MS Power BI outperforms other rivals significantly
- Google Data Studio becomes a challenger to the leading BI products above, occupying the third place in the list
- Data Scientists, Data Analysts, Research Scientists, and ML Engineers are the most frequent users of BI tools
- Huge fraction of the survey responders indicated they do not use BI tools at all



We find that

- AWS is popular among survey respondents in India and USA the most
- Brasil, Japan and UK go into tier 2 in terms of the number of respondents from these country who use AWS

We find that

- India is the top country where GCP is used
- USA takes the second place in the rank but it is significantly below India (unlike AWS, where India and USA were relatively on a par)
- Japan and Brazil are in the tier 2 in terms of the number of respondents from these country who use AWS
- GCP is less popular in UK, Canada and Australia vs. AWS
- GCP outperforms AWS in Turkey, Indonesia, and Russia

We find that

- Top country in terms of the number of MS Azure users is India (although MS Azure is well behind AWS and GCP there)
- USA holds the second place in the rank, and the number of MS Azure users is on a par with the number of GCP users in the US
- Brazil belongs to tier 2 in term of the number of MS Azure users
- In the majority of the countries (except the US), the number of MS Azure users is smaller then the number of GCP and AWS users

## Summary

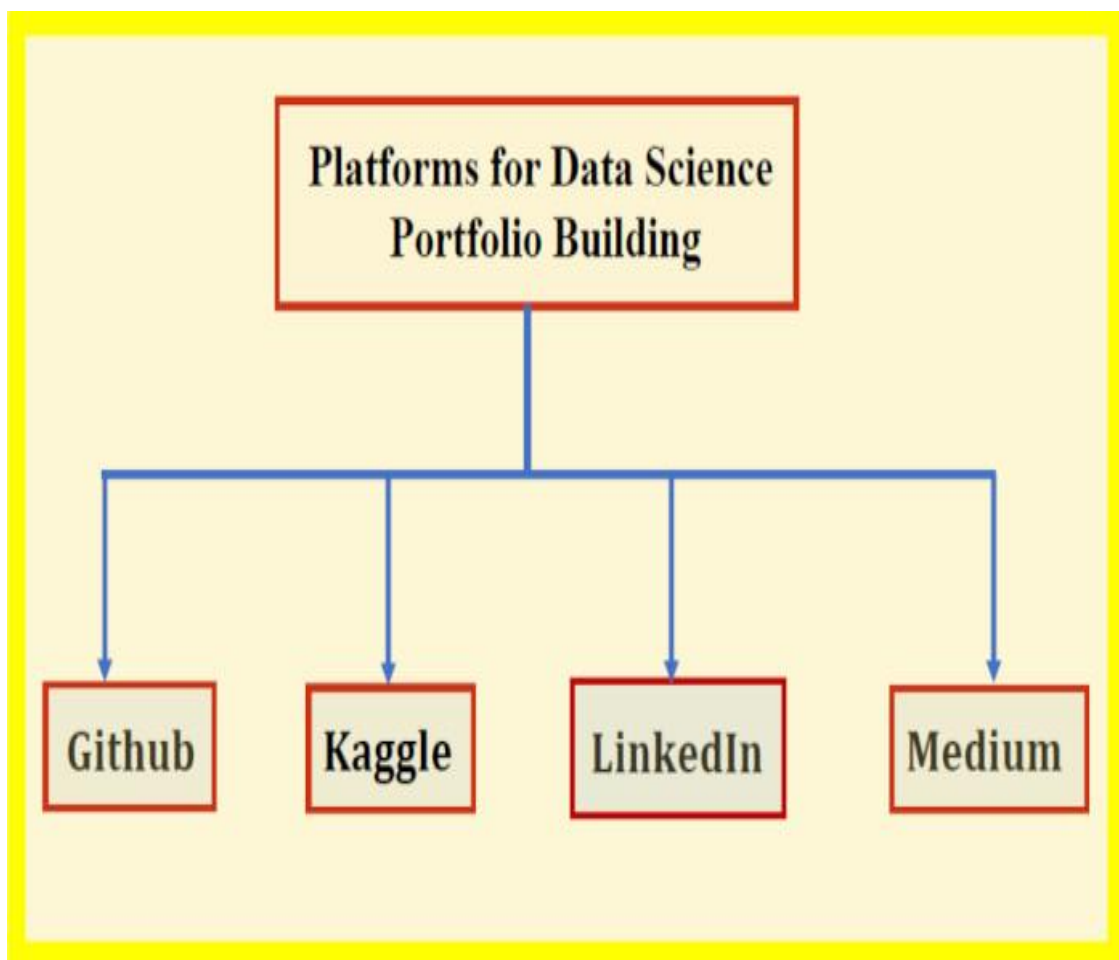
In this post, we reviewed the state of art with using Cloud computing platforms, products, and tools by the professionals in Data Science and ML industry. These are not just their preferences as of the end of 2020. These are cornerstones which are most likely to determine the trends for 2021–2022 as well.

The next couple of years will be crucial in the battle of Cloud Computing giants for minds, arms, and budgets in the Data Science and ML industry. Although AWS's position still looks stronger than other top rivals, the challenges from GCP could be the intrigues part of the market reshaping in the years to come. At the same time, MS Azure seems to keep its strong positions in

North America (while having little chances to penetrate other continents significantly vs. AWS and GCP).

However, we entered the age of global turbulence. 2021, the year under the Star of Kings, may expose us to unexpected surprises in every aspects of our lives.

## 6 Build a Data Science Portfolio that Stands Out Using These Platforms



*Platforms for data science portfolio building. Image by Benjamin O. Tayo.*

In the modern age of information technology, there is an enormous amount of [free resources for data science self-study](#). As a matter of fact, you can even design your own data science curriculum from the innumerable amount of available resources. While knowledge acquired from course work is essential to lay a good foundation in data science, you need to remember that data

science is a practical field. As such, hands-on skills are very important, especially if you are interested in working outside academia as a practicing data scientist.

This article will discuss 4 important platforms that will enable you **to build a portfolio to showcase your experience in data science**. A strong portfolio will give your employer an edge over the competition in attracting the best possible talent in the workforce. Keep in mind that employers interested in hiring you are going to ask you to provide evidence of completed data science projects. This famous quote from **Elon Musk** summarizes the mindset of employers in any technical discipline, including data science:

*“Generally, look for things that are evidence of exceptional ability. I don’t even care if somebody graduated from college or high school or whatever... Did they build some really impressive device? Win some really tough competition? Come up with some really great idea? Solve some really tough problem?”*

A **strong portfolio highlighting a list of completed projects, recognitions, and awards** will serve as evidence of your competence in data science.

Before delving into the topic of building a good data science portfolio, let’s first discuss 5 reasons why a data science portfolio is important.

## 5 Reasons why a data science portfolio is important

1. A portfolio helps you showcase your data science skills.
2. A portfolio enables you to network with other data science professionals and leaders in the field.
3. A portfolio is good for bookkeeping. You can use it to keep a record of your completed projects, including datasets, codes, and sample output files. That way, if you have to work on a similar project, you can always use code that has already been written, with only minor modifications.
4. By building a portfolio and networking with other data science professionals and leaders, you get exposed to technological changes in the field. Data science is a field that is ever-changing due to advances in technology. To keep up with the latest changes and developments in the field, it is important to join a network of data science professionals.
5. A portfolio increases your chances of getting a job. I’ve had numerous opportunities from LinkedIn, for instance, recruiters reaching out to me for job opportunities in data science.

Let’s now discuss 4 important platforms for creating a data science portfolio.

# Platforms for Building a Data Science Portfolio

## 1. GitHub

GitHub is a very useful platform for displaying your data science projects. As a data science aspirant, GitHub should serve as the first platform that you use as a repository of completed projects throughout your data science journey. These projects could include projects from weekly assignments or capstone projects. This platform enables you to share your code with other data scientists or data science aspirants. Employers interested in hiring you would check your GitHub portfolio to assess some of the projects you've completed. So, it's important for you to build a very strong and professional portfolio on GitHub.

To establish a GitHub portfolio, the first thing to do is create a GitHub account. Once your account is created, you may go ahead to edit your profile. When editing your profile, it's a good idea to add a short biography and a professional profile picture. You may find an example of a GitHub profile here: <https://github.com/bot13956>.

Now let's assume that you've completed an important data science project and you would like to create a GitHub repository for your project.

**Tips for creating a repository:** Make sure you choose a suitable title for your repository. Then include a README file to provide a synopsis of what your project is all about. Then you may upload your project files, including the dataset, Jupyter notebook, and sample outputs.

Here is an example of a GitHub repository for a machine learning project:

**Repository Name:** bot13956/ML\_Model\_for\_Predicting-Ships\_Crew\_Size

**Repository URL:** [https://github.com/bot13956/ML\\_Model\\_for\\_Predicting-Ships\\_Crew\\_Size](https://github.com/bot13956/ML_Model_for_Predicting-Ships_Crew_Size)

### README File:

```
ML_Model_for_Predicting-Ships_Crew_Size
```

```
Author: Benjamin O. Tayo
```

```
Date: 4/8/2019
```

```
We build a simple model using the cruise_ship_info.csv data set for predicting a ship's crew size. This project is organized as follows:
```

- (a) data preprocessing and variable selection;
- (b) basic regression model;
- (c) hyper-parameters tuning; and
- (d) techniques for dimensionality reduction.

```
cruise_ship_info.csv: dataset used for model building.
```

```
Ship_Crew_Size_ML_Model.ipynb: the Jupyter notebook containing code.
```

You can see from the sample README file that the file provides a good summary of what the project is all about, including goals and objectives, the dataset, and the Jupyter notebook file containing the code. When preparing a repository, always keep in mind that other users will have access to it since it is public, so you want to prepare it in such a way that it's easy to understand.

## **2. Kaggle**

[Kaggle](#) is the world's largest data science community with powerful tools and resources to help you achieve your data science goals. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges. On this platform, you can have access to [datasets](#), [courses](#), [notebooks](#), and [competitions](#). Again, as a beginner, you'll have to create an account, then set up your profile, including a profile picture and a short bio.

One of the primary purposes of joining Kaggle is to network with other data science professionals. It doesn't matter if you are new to data science or if you are a seasoned data scientist, you can find a suitable forum on Kaggle that would allow you to discover content and engage in discussion around topics that you're interested in. Your end goal should be to enter and participate in data science competitions launched on this platform. Because most competitions encourage teamwork, it is important to build a network with other data science aspirants who can serve as team members for Kaggle challenge competitions. As you participate in Kaggle competitions, you can showcase your completed projects, including your datasets, Jupyter notebooks, and project reports on your public profile.

## **3. LinkedIn**

LinkedIn is a very powerful platform for showcasing your skills and for networking with other data science professionals and organizations. LinkedIn is now one of the most famous platforms for posting data science jobs and for recruiting data scientists. I've actually got numerous data science interviews via LinkedIn.

Make sure your profile is up-to-date at all times. List your data science skill sets, as well as your experiences, including projects that you've completed. It would be worthwhile to also list awards and honors. You also want to let recruiters know that you are actively searching for a job. Also, on LinkedIn, you want to keep up-to-date by following data science influencers and publications such as [KDnuggets](#), [Towards Data Science](#), and [Towards AI](#). These companies post updates on interesting data science articles on various topics, including machine learning, deep learning, and artificial intelligence.

Find an example of my posts on LinkedIn from here: <https://www.linkedin.com/in/benjamin-o-tayo-ph-d-a2717511/detail/recent-activity/shares/>

## **4. Medium**

[Medium](#) is now considered one of the fastest-growing platforms for portfolio building and for networking. If you are interested in using this platform for portfolio building, the first step would be to create a Medium account. You can create a free account or a member account. With a free account, there are limitations on the number of member articles that you can actually access per month. A member account requires a monthly subscription fee of \$5 or \$50/year. Find out more about becoming a Medium member from here: <https://medium.com/membership>.

Once you've created an account, you can go ahead and create a profile. Make sure to include a professional picture and a short bio. Here is an example of a Medium profile: <https://medium.com/@benjaminobi>.

On Medium, a good way to network with other data science professionals is to become a follower. You can also follow specific Medium publications that are focused on data science. The 2 top data science publications are [Towards Data Science](#) and [Towards AI](#).

One of the best ways to enhance your portfolio on Medium is to become a Medium writer.

### **Why should you consider writing data science articles on Medium?**

Writing Medium articles has 5 main advantages:

1. It provides a means for you to showcase your knowledge and skills in data science.
2. It motivates you to work on challenging data science projects, thereby improving your data science skills.
3. It enables you to improve your communication skills. This is useful because it enables you to convey information in a way that the general public can understand.
4. Every article published on Medium is considered intellectual property, so you can add a medium article to your resume.
5. You can make money from your articles. By means of the [Medium Partner Program](#), anyone who publishes on Medium can make their articles eligible for earning money.

In summary, we've discussed 4 important platforms that could be used for building a data science portfolio. A portfolio is a very important way for you to showcase your skills and to network with other data science professionals. A good portfolio will not only help you keep up-to-date with new developments in the field, but it will help increase your visibility to potential recruiters.

# 9 How can you Convert a Business Problem into a Data Problem? A Successful Data Science Leader's Guide

- effectively translating business requirements to a data-driven solution is key to the success of your data science project
- Hear from a data science leader on his experience and thoughts on how to bridge this gap

## Introduction

How effectively can you convert a business problem into a data problem?

This question holds the key to unlocking the potential of your data science project. There is no one-size-fits-all approach here. This is a nontrivial effort with positive long-term results and hence deserves a great deal of focused collaboration across the product team, the data science team, and the engineering team.

Every leader knows that being able to measure progress is an invaluable aspect of any project. This understanding goes to an entirely different level when it comes to data science projects.

We discussed how to manage the different stakeholders in data science in [my previous article](#) (recap below). In this article, we are going to discuss the journey of translating the broad qualitative business requirements into tangible quantitative data-driven solutions.

One of the most tangible advantages of this approach, among many others, is that it establishes a common understanding of what ‘success’ means and how we can measure it. It also lays a framework for how progress will be tracked and communicated among the various internal and external stakeholders.

*This is the second article of a four-article series that discusses my learnings from developing data-driven products from scratch and deploying them in real-world environments where their performance influences the client's business/financial decisions. You can read articles one and three [here](#):*

## Table of Contents

- Quick Recap of Managing Different Data Science Stakeholders (Article #1)
- Bridging the Qualitative-to-Quantitative Gap in Data Science
- Is the Right Data Available with the Right Level of Granularity?
- Are We Asking the Right Questions?
- Repeatability and Reproducibility: Consistency in Labeled Data for Accurate AI Systems
- Active Learning for Efficient and More Accurate AI Systems
- Diverse Team Composition is Critical for Success

## Quick Recap of Managing Different Data Science Stakeholders (Article #1)

important to have this background before reading further as it is essentially the base on which this article will revolve.

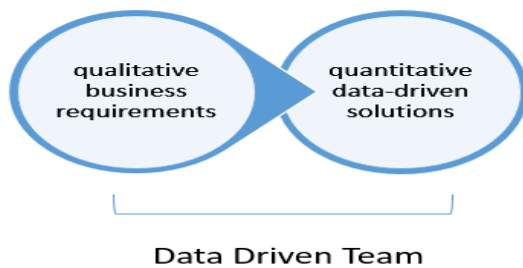
We discussed the three key stakeholders in a data-driven product ecosystem and how the data-science-delivery leader has to align them with each other. The three main stakeholders are:

- **The customer-facing team:** This team is tasked with the dual responsibility of ensuring that the internal teams act on customers' feedback/concerns in a timely manner and also of gauging the customers' unmet needs. When it comes to data-driven products, the customer-facing team, and through them the customers, have to be educated on the 'illusion of 100% accuracy' and 'continuous improvement process' which are unique to these data-driven products
- **The executive team:** It is critical to get the executive team's buy-in on the unique development, deployment and maintenance cycles of data-driven products. It is also important to help the executives distinguish between the low-stakes 'consumer-AI' image that the popular discourse has created versus the high-stakes ground reality of 'enterprise-AI' that the corporates will commonly face
- **The data science team:** The pace at which the data science field is evolving, there is always something new (and potentially fancier) to learn. While the core data science team may be tempted to periodically apply the newer technologies, the data science delivery leader has to regularly remind the data science team that the AI is only a part of the whole puzzle and that the 'appropriateness' of the technology matters more than its 'coolness'

With that background, let's dive into this article!



## Bridging the Qualitative-to-Quantitative Gap in Data Science



Consider the following mini-scenarios:

1. During a regular weekday lunch, as you are discussing how everybody's weekend was, one of your colleagues mentions she watched a particular movie that you have also been wanting to watch. To know her feedback on the movie, you ask her – "Hey, was the movie's direction up to the mark?"
2. You bump into a colleague in the hallway who you haven't seen for a couple of weeks. She mentions she just returned from a popular international destination vacation. To know more about the destination, you ask her – "Wow! Is it really as exotic as they show in the magazines?"
3. Your roommate got a new video game that he has been playing nonstop for a few hours. When he takes a break, you ask him – "Is the game really that cool?"

Did you find any of these questions 'artificial'? Do re-read the scenarios and take a few seconds to think through. Most of us would find these questions to be perfectly natural!

What would certainly be artificial though is asking questions like:

- 'Hey, was the movie direction 3.5-out-of-5?', or
- 'Is the vacation destination 8 on a scale of 1-to-10?', or
- 'Is the video game in the top 10 percentile of all the video games?'

In most scenarios, we express our asks in qualitative terms. This is true about business requirements as well.

Isn't it more likely that the initial client ask will be "Build us a landing page which is aesthetically pleasing yet informative" versus "we need a landing page which is rated at least 8.5-out-of-10 by 1000 random visitors to our website on visual-appearance, navigability and product-information parameters"?

On the other hand, systems are built and evaluated based on exact quantitative requirements. For example, the database query has to return in less than 30 milliseconds, the website has to fully load in less than 3 milliseconds on a typical 10mbps connection, and so on.



**This gap between qualitative business requirements and quantitative machine requirements is exacerbated when it comes to data-driven products.**

A typical business requirement for a data-driven product could be “develop an optimal digital marketing strategy to reach the likely target customer population”. Converting this to a quantifiable requirement has several non-trivial challenges. Some of these are:

- **How we define ‘optimal’:** Do we focus more on precision or more on recall? Do we focus more on accuracy (is the approached customer segment really our target customer segment or not)? Or do we focus more on efficiency (how quickly do we make a go/no-go decision once the customer segment is exposed to our algorithm)?
- **How do we actually evaluate if we have met the optimal criteria?** And if not, how much of a gap exists?

To define customers ‘similar’ to our target population, we need to agree on a set of N dimensions that will be used for computing this similarity:

- Patterns in the browsing history
- Patterns in e-shopping
- Patterns in user-provided meta-data, and so on. Or do we need to device a few other dimensions?

After that, we need to critically evaluate whether all the relevant data exists in an accessible format. If not, are there ways to infer at least parts of it?

### **Is the Right Data Available with the Right Level of Granularity?**

Consider a business scenario where a company has a chatbot that handles customer queries automatically. When the chatbot fails to resolve a customer query, the call is transferred to a human expert.

It is fair to assume that the cost of a human expert manning a call center is higher than an automated chatbot resolving the customer query. Thus, the business problem can be stated as: **Reduce the proportion of calls that reach a human expert.**

The first barrier to cross is often the **HiPPO Effect**.

Simply put, the HiPPO (Highest Paid Person's Opinion) effect states that the authority figure's suggestions are interpreted as the final truth, and promptly implemented, even if the findings from the data are contrary.

For instance, in the above example, the HiPPO might be that calls are getting diverted to human experts due to time-out issues related to network connectivity within the chatbot's workflow. A more prudent data-driven approach would be to list out all the possible reasons leading to call diversions, one of them being the connectivity issue.

Such a list can be derived from a combination of expert knowledge and some initial data log analysis. This step falls under, what we call, **the 'data-discovery' phase**.

The data-discovery phase, which is essentially an iterative process, **systematizes the use of insights from the data to guide the expert's intuition and to identify the next dimension of data to investigate.**

The data-discovery phase also identifies if there are any gaps in the 'ideal-data-needed' vs. 'actual-data-available'. For example, we may identify that the last interaction between the chatbot and the customer is not being stored in the database. This lack of data needs to be solved promptly by changing the data storage schema.

*Let's assume that this analysis of possible failure scenarios led to the following findings:*

1. The chatbot did not understand the intent
2. The chatbot is not able to establish a connection with the knowledge base
3. The chatbot is not able to retrieve relevant information from the knowledge base before the time-out
4. There is no relevant information in the knowledge-base, or
5. Unknown/non-replicable issues

Armed with this information, the next step would be to dig deeper. For example:

1. Is the intent not understood because the speech-to-text component failed or the text-to-intent mining component misfired?
2. Is the time-out occurring because the information is not stored in the right format (e.g., suboptimal inverted index)? or
3. Is the information not easily accessible (e.g., on an LRU cache vs in a network-call setup like ElasticSearch)? and so on

The findings from this step will help rank the problems in terms of their prevalence and also identify systemic issues. If the failure of the speech-to-text component is one of the prevalent problems, the speech-to-text vendor needs to be approached to identify if the speech inputs are not being captured/transferred as per the norms/best-practices or if the speech-to-text system needs more context for better predictions.

## Are We Asking the Right Questions?

Moving further along in this journey, translating qualitative data specific questions into quantitative model training strategies is also a nuanced topic, one that can have far-reaching consequences.

Continuing the conversation on speech-to-text issues, it may seem prudent to answer '*who is the caller?*'. At the surface level, it may seem synonymous to '*is the caller Miss Y?*'. But these two questions lead to totally different Machine Learning (ML) models.

The '*who is the caller?*' question leads to an N-class classification problem (where N is the number of possible callers), whereas '*is the caller Miss Y?*' leads to N binary-classifiers!

While all of this may seem complex and data science-led, we cannot underestimate the role of the domain expert. **While all errors are mathematically equal, some errors can be more damaging to the company's finances and reputation than others.**

Domain experts play a critical role in understanding the impact of these errors. Domain experts also help layout the best practices in the industry, understand customer expectations and adhere to regulatory requirements.

For example, even if the chatbot is 100% confident that the user has asked for a renewal of a relatively inexpensive service, the call may need to be routed to a human for regulatory compliance purposes depending on the nature of the service.

## Repeatability and Reproducibility: Consistency in Labeled Data for Accurate AI Systems

One of the final steps is to **have a relevant subset of data labeled by human experts in a consistent manner.**

At the vast scale of Big Data, we are talking about obtaining labels for hundreds of thousands of samples. This will need a huge team of human experts to provide the labels.

A more efficient way would be to sample the data in such a manner that only the most diverse set of samples are sent for labeling. One of the best ways to do this is to use **stratified sampling**. Domain experts will need to analyze which data dimensions get used for the stratification.

Consistency in human labels is trickier than it may seem at first. If the existing automated techniques for label generation are 100% accurate, then there is no need for training any newer machine learning algorithms. And hence, there is no need for human-labeled training samples (e.g., we do not need manual transcription of speech if speech-to-text systems are 100% accurate).



At the same time, if there is no subjectivity in human labeling, then it is just a matter of tabulating the list of steps that the human expert has followed and automating those steps. **Almost all practical machine learning systems need training because they are not able to adequately capture the various nuances that humans apply in coming to a particular decision.**

Thus, there will be a certain level of inherent subjectivity in the human labels that can't be done away with.

The goal, however, should be to design label-capturing systems that minimize avenues for 'extraneous' subjectivity.

For example, if we are training a machine learning system to predict emotion from speech, the human labels will be generated by playing the speech signals and asking the human labeler to provide the predominant emotion.

One way to minimize extraneous subjectivity is to provide a drop-down of the possible emotion label options instead of letting the human labeler enter his/her inputs in a free flow text format. Similarly, even before the first sample gets labeled, there should be a normalization exercise among the human experts where they agree on the interpretation of each label (e.g., what is the difference between 'sad' and 'angry').

An objective way to check the subjectivity is ‘repeatability and reproducibility (R&R)’.

**Repeatability** measures the impact of temporal context on human decisions. It is computed as follows:

- The same human expert is asked to label the same data sample at two different times
- The proportion of the times the expert agrees with themselves is called repeatability

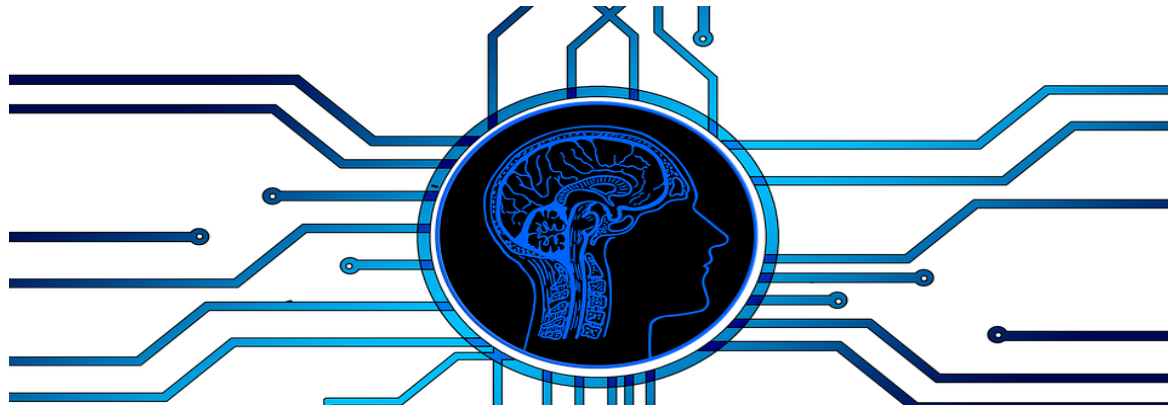
**Reproducibility** measures how consistently the labels can be replicated across experts. It is computed as follows:

- Two or more human experts are asked to label the same data samples in the same setting
- The proportion of the times the experts agree among themselves is called reproducibility

Conducting R&R evaluations on even a small scale of data can help identify process improvements as well as help gauge the complexity of the problem.

## Active Learning for Efficient and More Accurate AI Systems

Machine learning is typically ‘passive’. This means that the machine doesn’t proactively ask for human labels on samples where it is most confusing. Instead, the machines are trained on labeled samples that are fed to the training algorithms.



A relatively new branch of machine learning called **Active Learning** tries to address this. It does so by:

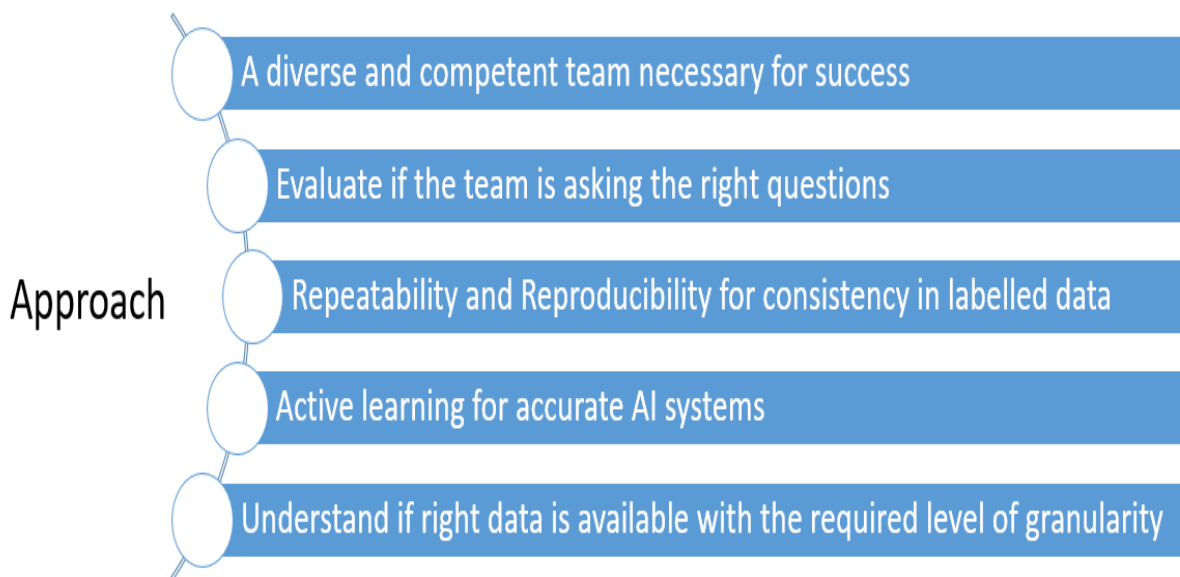
- First training a relatively simple model with limited human labels, and then
- Proactively highlighting only those samples where the model’s prediction confidence is below a certain threshold

The human labels are sought on priority for such ‘confusing samples’.

## Diverse Data Science Team Composition is Critical for Success

For all the pieces to come together, we need an “all-rounder” data science team:

- It is absolutely critical that the data science team has a **healthy mix of data scientists** who are trained to think in a data-driven manner. They should also be able to connect the problem-at-hand with established machine learning frameworks
- The team needs **Big Data engineers** who have expertise in data pipelining and automation. They should also understand, among other things, the various design factors that contribute to latency
- The team also needs **domain experts**. They can truly guide the rest of the members and the machine to interpret the data in ways consistent with the end customer’s needs



### Data formats

## How to read most commonly used file formats in Data Science (using Python)

### Introduction

If you have been part of the [data science](#) (or any data!) industry, you would know the challenge of working with different data types. Different formats, different compression, different parsing

on different systems – you could be quickly pulling your hair! Oh and I have not talked about the unstructured data or semi-structured data yet.

For any data scientist or data engineer, dealing with different formats can become a tedious task. In real-world, people rarely get neat tabular data. Thus, it is mandatory for any data scientist (or a data engineer) to be aware of different file formats, common challenges in handling them and the best / efficient ways to handle this data in real life.

This article provides common formats a data scientist or a data engineer must be aware of. I will first introduce you to different common file formats used in the industry. Later, we'll see how to read these file formats in [Python](#).

**P.S.** *In rest of this article, I will be referring to a data scientist, but the same applies to a data engineer or any data science professional.*

## Table of Contents

1. What is a file format?
2. Why should a [data scientist](#) understand different file formats?
3. Different file formats and how to read them in Python?
  1. Comma-separated values
  2. XLSX
  3. ZIP
  4. Plain Text (txt)
  5. JSON
  6. XML
  7. HTML
  8. Images
  9. Hierarchical Data Format
  10. PDF
  11. DOCX
  12. MP3
  13. MP4

### 1. What is a file format?

A file format is a standard way in which information is encoded for storage in a file. First, the file format specifies whether the file is a binary or ASCII file. Second, it shows how the information is organized. For example, comma-separated values (CSV) file format stores tabular data in plain text.



To identify a file format, you can usually look at the file extension to get an idea. For example, a file saved with name “Data” in “CSV” format will appear as “Data.csv”. By noticing “.csv” extension we can clearly identify that it is a “CSV” file and data is stored in a tabular format.

CSV					JSON					
	A	B	C	D						
1	ID	Gender	City	Monthly_I						
2	ID000002C	Female	Delhi	20000						
3	ID000004E	Male	Mumbai	35000						
4	ID000007H	Male	Panchkula	22500						
5	ID000008I	Male	Saharsa	35000						
6	ID000009J	Male	Bengaluru	100000						
7	ID000010K	Male	Bengaluru	45000						
8	ID000011L	Female	Sindhudurg	70000						
9	ID000012N	Male	Bengaluru	20000						
10	ID000013N	Male	Kochi	75000						
11	ID000014C	Female	Mumbai	30000						
12	ID000016C	Male	Mumbai	25000						
13	ID000018S	Female	Surat	25000						
14	ID000019T	Female	Pune	24000						
15	ID000021V	Male	Bhubanes	27000						
16	ID000022V	Female	Howrah	28000						

```
"Employee": [  
  {  
    "id": "1",  
    "Name": "Ankit",  
    "Sal": "1000",  
  },  
  {  
    "id": "2",  
    "Name": "Faizv",  
  }  
]
```

```
<?xml version="1.0"?>  
  
<contact-info>  
  
  <name>Ankit</name>  
  
  <company>Analytics Vidhya</company>  
  
  <phone>+9187654321</phone>  
  
</contact-info>
```

## 2. Why should a data scientist understand different file formats?

Usually, the files you will come across will depend on the application you are building. For example, in an image processing system, you need image files as input and output. So you will mostly see files in jpeg, gif or png format.

As a data scientist, you need to understand the underlying structure of various file formats, their advantages and dis-advantages. Unless you understand the underlying structure of the data, you will not be able to explore it. Also, at times you need to make decisions about how to store data.

Choosing the optimal file format for storing data can improve the performance of your models in data processing.

Now, we will look at the following file formats and how to read them in Python:

- Comma-separated values
- XLSX
- ZIP
- Plain Text (txt)
- JSON
- XML
- HTML
- Images
- Hierarchical Data Format
- PDF
- DOCX
- MP3
- MP4

### 3. Different file formats and how to read them in Python

#### 3.1 Comma-separated values

Comma-separated values file format falls under spreadsheet file format.

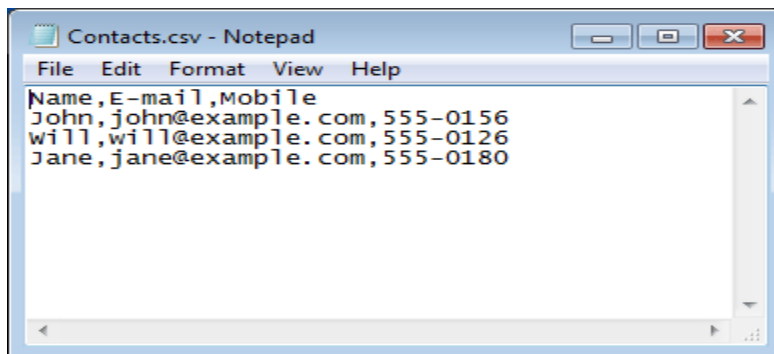
##### *What is Spreadsheet File Format?*

In spreadsheet file format, data is stored in cells. Each cell is organized in rows and columns. A column in the spreadsheet file can have different types. For example, a column can be of string type, a date type or an integer type. Some of the most popular spreadsheet file formats are Comma Separated Values ( CSV ), Microsoft Excel Spreadsheet ( xls ) and Microsoft Excel Open XML Spreadsheet ( xlsx ).

Each line in CSV file represents an observation or commonly called a record. Each record may contain one or more fields which are separated by a comma.

Sometimes you may come across files where fields are not separated by using a comma but they are separated using tab. This file format is known as TSV (Tab Separated Values) file format.

The below image shows a CSV file which is opened in Notepad.



#### Reading the data from CSV in Python

Let us look at how to read a CSV file in Python. For loading the data you can use the “pandas” library in python.

```
import pandas as pd

df = pd.read_csv("/home/Loan_Prediction/train.csv")
```

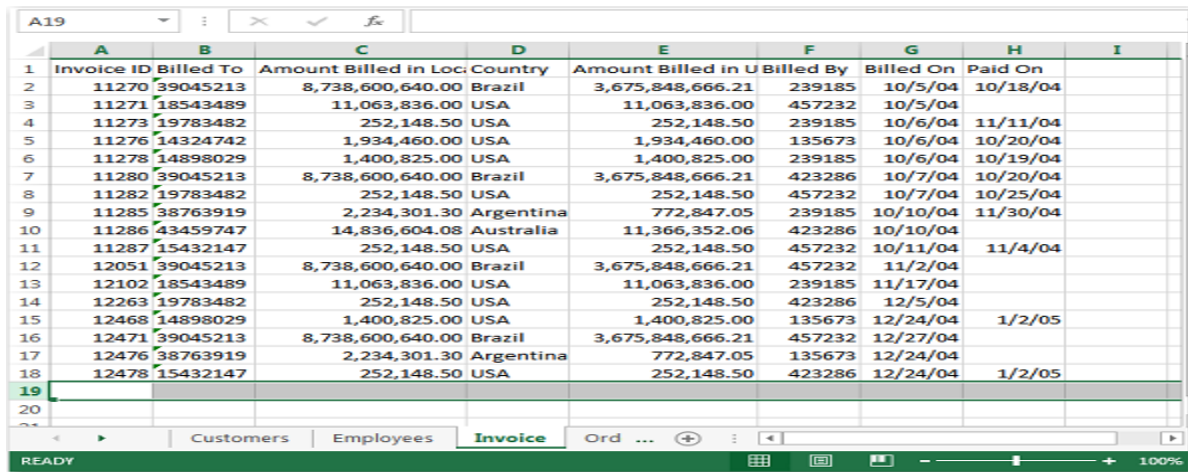
Above code will load the train.csv file in DataFrame df.

### 3.2 XLSX files

XLSX is a Microsoft Excel Open XML file format. It also comes under the Spreadsheet file format. It is an XML-based file format created by Microsoft Excel. The XLSX format was introduced with Microsoft Office 2007.

In XLSX data is organized under the cells and columns in a sheet. Each XLSX file may contain one or more sheets. So a workbook can contain multiple sheets.

The below image shows a “xlsx” file which is opened in Microsoft Excel.



	A	B	C	D	E	F	G	H	I
1	Invoice ID	Billed To	Amount Billed in Local	Country	Amount Billed in US	Billed By	Billed On	Paid On	
2	11270	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	239185	10/5/04	10/18/04	
3	11271	18543489	11,063,836.00	USA	11,063,836.00	457232	10/5/04		
4	11273	19783482	252,148.50	USA	252,148.50	239185	10/6/04	11/11/04	
5	11276	14324742	1,934,460.00	USA	1,934,460.00	135673	10/6/04	10/20/04	
6	11278	14898029	1,400,825.00	USA	1,400,825.00	239185	10/6/04	10/19/04	
7	11280	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	423286	10/7/04	10/20/04	
8	11282	19783482	252,148.50	USA	252,148.50	457232	10/7/04	10/25/04	
9	11285	38763919	2,234,301.30	Argentina	772,847.05	239185	10/10/04	11/30/04	
10	11286	43459747	14,836,604.08	Australia	11,366,352.06	423286	10/10/04		
11	11287	15432147	252,148.50	USA	252,148.50	457232	10/11/04	11/4/04	
12	12051	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	457232	11/2/04		
13	12102	18543489	11,063,836.00	USA	11,063,836.00	239185	11/17/04		
14	12263	19783482	252,148.50	USA	252,148.50	423286	12/5/04		
15	12468	14898029	1,400,825.00	USA	1,400,825.00	135673	12/24/04	1/2/05	
16	12471	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	457232	12/27/04		
17	12476	38763919	2,234,301.30	Argentina	772,847.05	135673	12/24/04		
18	12478	15432147	252,148.50	USA	252,148.50	423286	12/24/04	1/2/05	

In above image, you can see that there are multiple sheets present (bottom left) in this file, which are Customers, Employees, Invoice, Order. The image shows the data of only one sheet – “Invoice”.

### Reading the data from XLSX file

Let’s load the data from XLSX file and define the sheet name. For loading the data you can use the Pandas library in python.

```
import pandas as pd
```

```
df = pd.read_excel("/home/Loan_Prediction/train.xlsx", sheetname = "Invoice")
```

Above code will load the sheet “Invoice” from “train.xlsx” file in DataFrame df.

### 3.3 ZIP files

ZIP format is an archive file format.

## What is Archive File format?

In Archive file format, you create a file that contains multiple files along with metadata. An archive file format is used to collect multiple data files together into a single file. This is done for simply compressing the files to use less storage space.

There are many popular computer data archive format for creating archive files. Zip, RAR and Tar being the most popular archive file format for compressing the data.

So, a ZIP file format is a lossless compression format, which means that if you compress the multiple files using ZIP format you can fully recover the data after decompressing the ZIP file. ZIP file format uses many compression algorithms for compressing the documents. You can easily identify a ZIP file by the .zip extension.

## Reading a .ZIP file in Python

You can read a zip file by importing the “zipfile” package. Below is the python code which can read the “train.csv” file that is inside the “T.zip”.

```
import zipfile
archive = zipfile.ZipFile('T.zip', 'r')
df = archive.read('train.csv')
```

Here, I have discussed one of the famous archive format and how to open it in python. I am not mentioning other archive formats. If you want to read about different archive formats and their comparisons you can refer this [link](#).

## 3.4 Plain Text (txt) file format

In Plain Text file format, everything is written in plain text. Usually, this text is in unstructured form and there is no meta-data associated with it. The txt file format can easily be read by any program. But interpreting this is very difficult by a computer program.

Let's take a simple example of a text File.

The following example shows text file data that contain text:

```
"In my previous article, I introduced you to the basics of Apache Spark,
different data representations (RDD / DataFrame / Dataset) and basics of operations (Transformation and
Action). We even solved a machine learning problem from one of our past hackathons. In this article, I will
continue from the place I left in my previous article. I will focus on manipulating RDD in PySpark by applying
operations"
```

*(Transformation and Actions)."*

Suppose the above text written in a file called text.txt and you want to read this so you can refer the below code.

```
text_file = open("text.txt", "r")
lines = text_file.read()
```

### 3.5 JSON file format

JavaScript Object Notation(JSON) is a text-based open standard designed for exchanging the data over web. JSON format is used for transmitting structured data over the web. The JSON file format can be easily read in any programming language because it is language-independent data format.

Let's take an example of a JSON file

The following example shows how a typical JSON file stores information of employees.

```
{
  "Employee": [
    {
      "id": "1",
      "Name": "Ankit",
      "Sal": "1000",
    },
    {
      "id": "2",
      "Name": "Faizy",
      "Sal": "2000",
    }
  ]
}
```

### Reading a JSON file

Let's load the data from JSON file. For loading the data you can use the pandas library in python.

```
import pandas as pd

df = pd.read_json("/home/kunal/Downloads/Loan_Prediction/train.json")
```

### 3.6 XML file format

XML is also known as Extensible Markup Language. As the name suggests, it is a markup language. It has certain rules for encoding data. XML file format is a human-readable and machine-readable file format. XML is a self-descriptive language designed for sending information over the internet. XML is very similar to HTML, but has some differences. For example, XML does not use predefined tags as HTML.

Let's take the simple example of XML File format.

The following example shows an xml document that contains the information of an employee.

```
<?xml version="1.0"?>
<contact-info>

<name>Ankit</name>

<company>Anlytics Vidhya</company>

<phone>+9187654321</phone>

</contact-info>
```

The “<?xml version=“1.0”?>” is a XML declaration at the start of the file (it is optional). In this declaration, *version* specifies the XML version and *encoding* specifies the character encoding used in the document. <contact-info> is a tag in this document. Each XML-tag needs to be closed.

### Reading XML in python

For reading the data from XML file you can import xml.etree. ElementTree library.

Let's import an xml file called train and print its root tag.

```
import xml.etree.ElementTree as ET
tree = ET.parse('/home/sunilray/Desktop/2 sigma/train.xml')
root = tree.getroot()
print root.tag
```

### 3.7 HTML files

HTML stands for Hyper Text Markup Language. It is the standard markup language which is used for creating Web pages. HTML is used to describe structure of web pages using markup. HTML tags are same as XML but these are predefined. You can easily identify HTML document subsection on basis of tags such as <head> represent the heading of HTML document. <p> “paragraph” paragraph in HTML. HTML is not case sensitive.

The following example shows an HTML document.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body><h1>My First Heading</h1>
<p>My first paragraph.</p></body>
</html>
```

Each tag in HTML is enclosed under the angular bracket(<>). The <!DOCTYPE html> tag defines that document is in HTML format. <html> is the root tag of this document. The <head> element contains heading part of this document. The <title>, <body>, <h1>, <p> represent the title, body, heading and paragraph respectively in the HTML document.

### Reading the HTML file

For reading the HTML file, you can use BeautifulSoup library. Please refer to this tutorial, which will guide you how to parse HTML documents. [Beginner’s guide to Web Scraping in Python \(using BeautifulSoup\)](#)

### 3.8 Image files

Image files are probably the most fascinating file format used in data science. Any computer vision application is based on image processing. So it is necessary to know different image file formats.

Usual image files are 3-Dimensional, having RGB values. But, they can also be 2-Dimensional (grayscale) or 4-Dimensional (having intensity) – an Image consisting of pixels and meta-data associated with it.

Each image consists one or more frames of pixels. And each frame is made up of two-dimensional array of pixel values. Pixel values can be of any intensity. Meta-data associated with an image, can be an image type (.png) or pixel dimensions.

Let's take the example of an image by loading it.

```
from scipy import misc
f = misc.face()
misc.imshow('face.png', f) # uses the Image module (PIL)
import matplotlib.pyplot as plt
plt.imshow(f)
plt.show()
```



Now, let's check the type of this image and its shape.

```
type(f) , f.shape
numpy.ndarray, (768, 1024, 3)
```

If you want to read about image processing you can refer this article. This article will teach you image processing with an example – [Basics of Image Processing in Python](#)

### 3.9 Hierarchical Data Format (HDF)

In Hierarchical Data Format ( HDF ), you can store a large amount of data easily. It is not only used for storing high volumes or complex data but also used for storing small volumes or simple data.

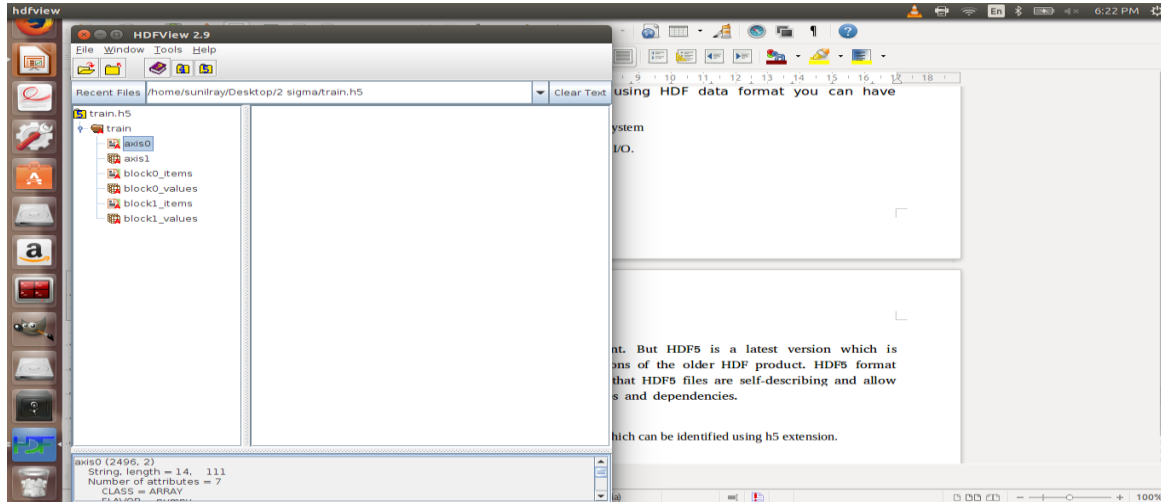
The advantages of using HDF are as mentioned below:

- It can be used in every size and type of system
- It has flexible, efficient storage and fast I/O.
- Many formats support HDF.

There are multiple HDF formats present. But, HDF5 is the latest version which is designed to address some of the limitations of the older HDF file formats. HDF5 format has some similarity with XML. Like XML, HDF5 files are self-describing and allow users to specify complex data relationships and dependencies.



Let's take the example of an HDF5 file format which can be identified using .h5 extension.



## Read the HDF5 file

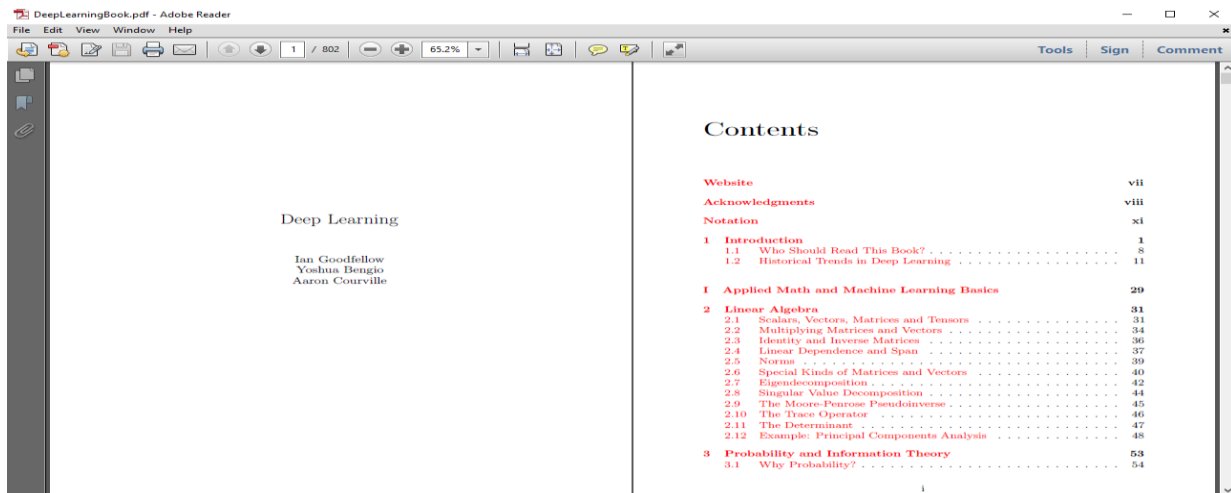
You can read the HDF file using pandas. Below is the python code can load the train.h5 data into the “t”.

```
t = pd.read_hdf('train.h5')
```

## 3.10 PDF file format

PDF (Portable Document Format) is an incredibly useful format used for interpretation and display of text documents along with incorporated graphics. A special feature of a PDF file is that it can be secured by a password.

Here's an example of a pdf file.



## Reading a PDF file

On the other hand, reading a PDF format through a program is a complex task. Although there exists a library which do a good job in parsing PDF file, one of them is PDFMiner. To read a PDF file through PDFMiner, you have to:

- Download PDFMiner and install it through the [website](#)
- Extract PDF file by the following code

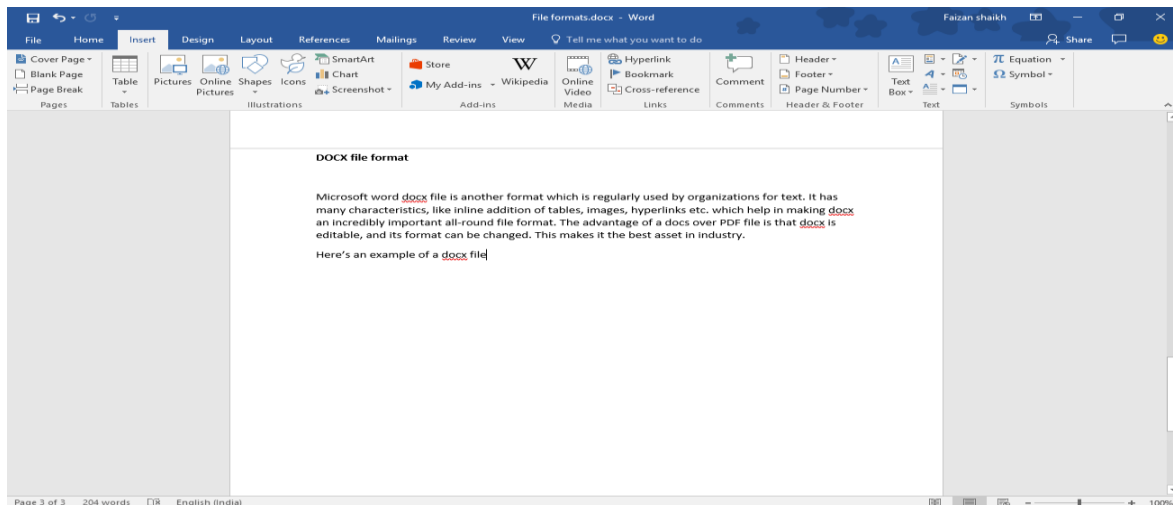
```
pdf2txt.py <pdf_file>.pdf
```

### 3.11 DOCX file format

Microsoft word docx file is another file format which is regularly used by organizations for text based data. It has many characteristics, like inline addition of tables, images, hyperlinks, etc. which helps in making docx an incredibly important file format.

The advantage of a docx file over a PDF file is that a docx file is editable. You can also change a docx file to any other format.

Here's an example of a docx file:



## Reading a docx file

Similar to PDF format, python has a community contributed library to parse a docx file. It is called python-docx2txt.

Installing this library is easy through pip by:

```
pip install docx2txt
```

To read a docx file in Python use the following code:

```
import docx2txt
text = docx2txt.process("file.docx")
```

## 3.12 MP3 file format

MP3 file format comes under the multimedia file formats. Multimedia file formats are similar to image file formats, but they happen to be one the most complex file formats.

In multimedia file formats, you can store variety of data such as text image, graphical, video and audio data. For example, A multimedia format can allow text to be stored as Rich Text Format (RTF) data rather than ASCII data which is a plain-text format.

MP3 is one of the most common audio coding formats for digital audio. A mp3 file format uses the MPEG-1 (Moving Picture Experts Group – 1) encoding format which is a standard for lossy compression of video and audio. In lossy compression, once you have compressed the original file, you cannot recover the original data.

A mp3 file format compresses the quality of audio by filtering out the audio which can not be heard by humans. MP3 compression commonly achieves 75 to 95% reduction in size, so it saves a lot of space.

## mp3 File Format Structure

A mp3 file is made up of several frames. A frame can be further divided into a header and data block. We call these sequence of frames an elementary stream.

A header in mp3 usually, identify the beginning of a valid frame and a data blocks contain the (compressed) audio information in terms of frequencies and amplitudes. If you want to know more about mp3 file structure you can refer this [link](#).

## Reading the multimedia files in python

For reading or manipulating the multimedia files in Python you can use a library called [PyMedia](#).

### 3.13 MP4 file format

MP4 file format is used to store videos and movies. It contains multiple images (called frames), which play in form of a video as per a specific time period. There are two methods for interpreting a mp4 file. One is a closed entity, in which the whole video is considered as a single entity. And other is mosaic of images, where each image in the video is considered as a different entity and these images are sampled from the video.

Here's is an example of mp4 video



## Reading an mp4 file

MP4 also has a community built library for reading and editing mp4 files, called MoviePy.

You can install the library from this [link](#). To read a mp4 video clip, in Python use the following code.

```
from moviepy.editor import VideoFileClip
clip = VideoFileClip('<video_file>.mp4')
```

You can then display this in jupyter notebook as below

```
ipython_display(clip)
```

## 12 Commonly used file formats in Data Science

### What is a File Format

File formats are designed to store specific types of information, such as **CSV**, **XLSX** etc. The file format also tells the computer how to display or process its content. Common file formats, such as **CSV**, **XLSX**, **ZIP**, **TXT** etc.

If you see your future as a data scientist so you must understand the different types of file format. Because data science is all about the data and it's processing and if you don't understand the file format so may be it's quite complicated for you. Thus, it is mandatory for you to be aware of different file formats.

### Different type of file formats:

**CSV:** the CSV is stand for Comma-separated values. as-well-as this name CSV file is use comma to separated values. In CSV file each line is a data record and Each record consists of one or more then one data fields, the field is separated by commas.

### Code: Python code to read csv file in pandas

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
import pandas as pd
df = pd.read_csv("file_path / file_name.csv")
print(df)
```

**XLSX:** The XLSX file is Microsoft Excel Open XML Format Spreadsheet file. This is used to store any type of data but it's mainly used to store financial data and to create mathematical models etc.

### **Code: Python code to read xlsx file in pandas**

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
import pandas as pd
df = pd.read_excel (r'file_path\\name.xlsx')
print (df)
```

### **Note:**

install xlrd before reading excel file in python for avoid the error. You can install xlrd using following command.

```
pip install xlrd
```

**ZIP:** ZIP files are used as data containers, they store one or more than one files in the compressed form. It is widely used in internet. After you download a ZIP file, you need to unpack its contents in order to use it.

### **Code: Python code to read zip file in pandas**

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
import pandas as pd
df = pd.read_csv(' File_Path \\ File_Name .zip')
print(df)
```

**TXT:** TXT files are useful for storing information in plain text with no special formatting beyond basic fonts and font styles. It is recognized by any text editing and other software programs.

### **Code: Python code to read txt file in pandas**

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
import pandas as pd
df = pd.read_csv('File_Path \\ File_Name .txt')
print(df)
```

**JSON:** JSON is stand for JavaScript Object Notation. JSON is a standard text-based format for representing structured data based on JavaScript object syntax

**Code: Python code to read json file in pandas**

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
import pandas as pd
df = pd.read_json('File_path \\ File_Name .json')
print(df)
```

**HTML:** HTML is stand for stands for Hyper Text Markup Language is use fore creating web pages. we can read html table in python pandas using read\_html() function.

**Code: Python code to read html file in pandas**

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
import pandas as pd
df = pd.read_html('File_Path \\File_Name.html')
print(df)
```

**Note:**

You need to install a package named “lxml & html5lib” which can handle the file with ‘.html’ extension.

```
pip install html5lib
pip install lxml
```

**PDF:** pdf stands for Portable Document Format (PDF) this file format is use when we need to save files that cannot be modified but still need to be easily available.

**Code: Python code to read pdf in pandas**

*filter\_none*

*edit*

*play\_arrow*

*brightness\_4*

```
pip install tabula-py
pip install pandas
df = tabula.read_pdf(file_path \\ file_name .pdf)
print(df)
```

**Note:**

You need to install a package named “tabula-py” which can handle the file with ‘.pdf’ extension.  
`pip install tabula-py`

**Data Formats for Data Science**

**<https://av.tib.eu/media/21234>**

**data parsing and transformation in data science**

**The transformation process starts with structuring the data into a single format, so it becomes compatible with the system in which it is copied and the other data available in it. Parsing is a process of analyzing data**



# structures and confirming the same with the rules of grammar.13-J

## What is data transformation: definition, benefits, and uses

Analyzing information requires structured and accessible data for best results. Data transformation enables organizations to alter the structure and format of raw data as needed. Learn how your enterprise can transform its data to perform analytics efficiently.

### What is data transformation?

Data transformation is the process of changing the format, structure, or values of data. For data analytics projects, data may be transformed at two stages of the data pipeline. Organizations that use on-premises data warehouses generally use an ETL ([extract, transform, load](#)) process, in which [data transformation is the middle step](#). Today, most organizations use cloud-based data warehouses, which can scale compute and storage resources with latency measured in seconds or minutes. The scalability of the cloud platform lets organizations skip preload transformations and load raw data into the data warehouse, then transform it at query time — a model called ELT ([extract, load, transform](#)).

Processes such as [data integration](#), [data migration](#), [data warehousing](#), and [data wrangling](#) all may involve data transformation.

Data transformation may be constructive (adding, copying, and replicating data), destructive (deleting fields and records), aesthetic (standardizing salutations or street names), or structural (renaming, moving, and combining columns in a database).

An enterprise can choose among a variety of [ETL tools](#) that automate the process of data transformation. Data analysts, data engineers, and data scientists also transform data using [scripting languages such as Python](#) or [domain-specific languages like SQL](#).

### Benefits and challenges of data transformation

Transforming data yields several benefits:

- Data is transformed to make it better-organized. Transformed data may be easier for both humans and computers to use.
- Properly formatted and validated data improves data quality and protects applications from potential landmines such as null values, unexpected duplicates, incorrect indexing, and incompatible formats.
- Data transformation facilitates compatibility between applications, systems, and types of data. Data used for multiple purposes may need to be transformed in different ways.

However, there are challenges to transforming data effectively:

- Data transformation can be expensive. The cost is dependent on the specific infrastructure, software, and tools used to process data. Expenses may include those related to licensing, computing resources, and hiring necessary personnel.
- Data transformation processes can be resource-intensive. Performing transformations in an on-premises data warehouse after loading, or transforming data before feeding it into applications, can create a computational burden that slows down other operations. If you use a cloud-based data warehouse, you can do the transformations after loading because the platform can scale up to meet demand.
- Lack of expertise and carelessness can introduce problems during transformation. Data analysts without appropriate subject matter expertise are less likely to notice typos or incorrect data because they are less familiar with the range of accurate and permissible values. For example, someone working on medical data who is unfamiliar with relevant terms might fail to flag disease names that should be mapped to a singular value or notice misspellings.
- Enterprises can perform transformations that don't suit their needs. A business might change information to a specific format for one application only to then revert the information back to its prior format for a different application.

## How to transform data

Data transformation can increase the efficiency of analytic and business processes and enable better data-driven decision-making. The first phase of data transformations should include things like data type conversion and flattening of hierarchical data. These operations shape data to increase compatibility with analytics systems. Data analysts and data scientists can implement further transformations additively as necessary as [individual layers of processing](#). Each layer of processing should be designed to perform a specific set of tasks that meet a known business or technical requirement.

Data transformation serves many functions within the data analytics stack.

### Extraction and parsing

In the modern ELT process, data ingestion begins with extracting information from a data source, followed by copying the data to its destination. Initial transformations are focused on shaping the format and structure of data to ensure its compatibility with both the destination system and the data already there. Parsing fields out of comma-delimited log data for loading to a relational database is an example of this type of data transformation.

### Translation and mapping

Some of the most basic data transformations involve the mapping and translation of data. For example, a column containing integers representing error codes can be mapped to the relevant error descriptions, making that column easier to understand and more useful for display in a customer-facing application.

Translation converts data from formats used in one system to formats appropriate for a different system. Even after parsing, web data might arrive in the form of hierarchical JSON or XML files, but need to be translated into row and column data for inclusion in a relational database.

### **Filtering, aggregation, and summarization**

Data transformation is often concerned with whittling data down and making it more manageable. Data may be consolidated by filtering out unnecessary fields, columns, and records. Omitted data might include numerical indexes in data intended for graphs and dashboards or records from business regions that aren't of interest in a particular study.

Data might also be aggregated or summarized. by, for instance, transforming a time series of customer transactions to hourly or daily sales counts.

BI tools can do this filtering and aggregation, but it can be more efficient to do the transformations before a reporting tool accesses the data.

### **Enrichment and imputation**

Data from different sources can be merged to create denormalized, enriched information. A customer's transactions can be rolled up into a grand total and added into a customer information table for quicker reference or for use by customer analytics systems. Long or freeform fields may be split into multiple columns, and missing values can be imputed or corrupted data replaced as a result of these kinds of transformations.

### **Indexing and ordering**

Data can be transformed so that it's ordered logically or to suit a data storage scheme. In relational database management systems, for example, creating indexes can improve performance or improve the management of relationships between different tables.

### **Anonymization and encryption**

Data containing personally identifiable information, or other information that could compromise privacy or security, should be anonymized before propagation. Encryption of private data is a requirement in many industries, and systems can perform encryption at multiple levels, from individual database cells to entire records or fields.

### **Modeling, typecasting, formatting, and renaming**

Finally, a whole set of transformations can reshape data without changing content. This includes casting and converting data types for compatibility, adjusting dates and times with offsets and format localization, and renaming schemas, tables, and columns for clarity.

## Refining the data transformation process

Before your enterprise can run analytics, and even before you transform the data, you must replicate it to a data warehouse architected for analytics. Most organizations today choose a cloud data warehouse, allowing them to take full advantage of ELT. Stitch can load all of your data to your [preferred data warehouse](#) in a raw state, ready for transformation. [Try Stitch](#) for free.

# 14 --6 Methods of Data Transformation in Data Mining

Data is currently one of the most important ingredients for success for any modern-day organization. With [data science](#) being rated among the most exciting fields to work, companies are hiring data scientists to make sense of their business data. These data professionals use a process called data mining to uncover hidden information from the company databases.

But, as most of this data is unstructured, it might be difficult to understand. It needs to be converted into a format that is easier to analyze. For this, the techies use data transformation tools.

In this article, we will learn about the different methods of **data transformation in data mining**. But first, let us see what data mining means.

### Table of Contents

- [What is Data Mining?](#)
- [Applications of Data Mining](#)
- [Data Transformation in Data Mining: The Processes](#)
  - [Data Smoothing](#)
  - [Data Aggregation](#)
  - [Discretization](#)
  - [Generalization](#)
  - [Attribute construction](#)
  - [Normalization](#)
- [Wrapping up](#)

## What is Data Mining?

[Data mining](#) is the method of analyzing data to determine patterns, correlations and anomalies in datasets. These datasets consist of data sourced from employee databases, financial information, vendor lists, client databases, network traffic and customer accounts. Using statistics, [machine learning \(ML\)](#) and [artificial intelligence \(AI\)](#), huge datasets can be explored manually or automatically.

Data mining helps companies develop better business strategies, enhance customer relationships, decrease costs and increase revenues.

In the data mining process, the business goal that is to be achieved using the data is determined first. Data is then collected from various sources and loaded into data warehouses, which is a repository of analytical data. Further, data is cleansed – missing data is added and duplicate data is removed. Sophisticated tools and mathematical models are used to find patterns within the data.

The results are compared with the business objectives to see whether it can be used for business operations. Based on the comparison, the data is deployed within the company. It is then presented using easy to understand graphs or tables.

## Applications of Data Mining

Data mining is used in several sectors:

- Multimedia companies use data mining to understand consumer behaviour and launch appropriate campaigns.
- Financial firms use it to understand market risks, detect financial frauds and get the best investment returns.
- In retail companies, data mining is used for understanding customer demands, their behaviour, forecast sales, and launch more targeted ad campaigns through data models.
- Manufacturing industries use data mining tools to manage their supply chain, improve quality assurance, and use machine data to predict machinery defects that help in the maintenance.
- Data mining is used to upgrade security systems, detect intrusions and malware. Data mining software can be used to analyze e-mails and filter out spam from your e-mail accounts.

## Data Transformation in Data Mining: The Processes

**Data transformation in data mining** is done for combining unstructured data with structured data to analyze it later. It is also important when the data is transferred to a new cloud [data warehouse](#). When the data is homogeneous and well-structured, it is easier to analyze and look for patterns.

For example, a company has acquired another firm and now has to consolidate all the business data. The smaller company may be using a different database than the parent firm. Also, the data in these databases may have unique IDs, keys and values. All this needs to be formatted so that all the records are similar and can be evaluated.

This is why data transformation methods are applied. And, they are described below:

### Data Smoothing

This method is used for removing the noise from a dataset. Noise is referred to as the distorted and meaningless data within a dataset. Smoothing uses algorithms to highlight the special features in the data. After removing noise, the process can detect any small changes to the data to detect special patterns.

Any data modification or trend can be identified by this method.

**Read:** [Data Mining Projects in India](#)

### **Data Aggregation**

Aggregation is the process of collecting data from a variety of sources and storing it in a single format. Here, data is collected, stored, analyzed and presented in a report or summary format. It helps in gathering more information about a particular data cluster. The method helps in collecting vast amounts of data.

This is a crucial step as accuracy and quantity of data is important for proper analysis. Companies collect data about their website visitors. This gives them an idea about customer demographics and behaviour metrics. This aggregated data assists them in designing personalized messages, offers and discounts.

### **Discretization**

This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels. This makes the data easier to study and analyze. If a continuous attribute is handled by a [data mining](#) task, then its discrete values can be replaced by constant quality attributes. This improves the efficiency of the task.

This method is also called data reduction mechanism as it transforms a large dataset into a set of categorical data. Discretization also uses [decision tree-based algorithms](#) to produce short, compact and accurate results when using discrete values.

### **Generalization**

In this process, low-level data attributes are transformed into high-level data attributes using concept hierarchies. This conversion from a lower level to a higher conceptual level is useful to get a clearer picture of the data. For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).

Data generalization can be divided into two approaches – data cube process (OLAP) and attribute oriented induction approach (AOI).

### **Attribute construction**

In the attribute construction method, new attributes are created from an existing set of attributes. For example, in a dataset of employee information, the attributes can be employee name, employee ID and address. These attributes can be used to construct another dataset that contains information about the employees who have joined in the year 2019 only.

This method of reconstruction makes mining more efficient and helps in creating new datasets quickly.

## Normalization

Also called data pre-processing, this is one of the crucial techniques for **data transformation in data mining**. Here, the data is transformed so that it falls under a given range. When attributes are on different ranges or scales, data modelling and mining can be difficult. Normalization helps in applying [data mining algorithms](#) and extracting data faster.

**The popular normalization methods are:**

- Min-max normalization
- Decimal scaling
- Z-score normalization

## Wrapping up

The techniques of **data transformation in data mining** are important for developing a usable dataset and performing operations, such as lookups, adding timestamps and including geolocation information. Companies use code scripts written in Python or SQL or cloud-based **ETL (extract, transform, load)** tools for data transformation.

# Data Preprocessing : Concepts

**Data is truly considered a resource in today's world. As per the World Economic Forum, by 2025 we will be generating about 463 exabytes of data globally per day! But is all this data fit enough to be used by machine learning algorithms? How do we decide that? In this article we will explore the topic of data preprocessing — transforming the data such that it becomes machine-readable...**

The aim of this article is to introduce the concepts that are used in data preprocessing, a major step in the Machine Learning Process. Let us start with defining what it is.

### ***What is Data Preprocessing?***

When we talk about data, we usually think of some large datasets with huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc..

Machines don't understand free text, image or video data as it is, they understand 1s and 0s. So it probably won't be good enough if we put on a slideshow of all our images and expect our machine learning model to get trained just by that!

In any Machine Learning process, Data Preprocessing is that step in which the data gets transformed, or *Encoded*, to bring it to such a state that now the machine can easily parse it. In other words, the *features* of the data can now be easily interpreted by the algorithm.

### ***Features in Machine Learning***

A dataset can be viewed as a collection of *data objects*, which are often also called as a records, points, vectors, patterns, events, cases, samples, observations, or entities.

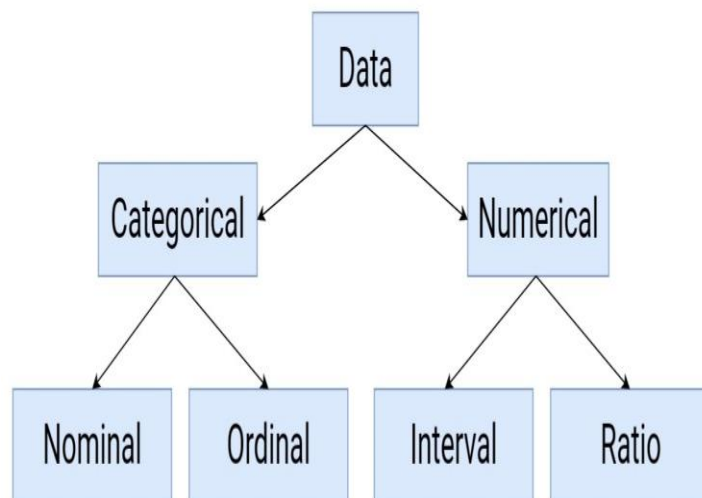
Data objects are described by a number of *features*, that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc..

Features are often called as variables, characteristics, fields, attributes, or dimensions.

As per [Wikipedia](#),

A feature is an individual measurable property or characteristic of a phenomenon being observed

For instance, color, mileage and power can be considered as features of a car. There are different types of features that we can come across when we deal with data.



### Statistical Data Types

Features can be:

- **Categorical** : Features whose values are taken from a defined set of values. For instance, days in a week : {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday} is a category because its value is always taken from this set. Another example could be the Boolean set : {True, False}



- **Numerical** : Features whose values are continuous or integer-valued. They are represented by numbers and possess most of the properties of numbers. For instance, number of steps you walk in a day, or the speed at which you are driving your car at.

Nominal	Ordinal	Interval	Ratio
Categorical variables without any implied order	Categorical variables with a natural implied order but the scale of difference is not defined	Numeric variables with a defined unit of measurement, so the differences between values are meaningful	Numeric variables with a defined unit of measurement but both differences and ratios are meaningful
Example : A new car model comes in these colors : Black, Blue, White, Silver	Example : Sizes of clothes has a natural order : Extra Small < Small < Medium < Large < Extra Large - But this does not mean Large - Medium = Medium - Small	Examples : Calendar Dates, Temperature in Celsius or Fahrenheit	Examples : Temperature in Kelvin, Monetary quantities, Counts, Age, Mass, Length, Electrical Current

## Feature Types

Now that we have gone over the basics, let us begin with the steps of Data Preprocessing. Remember, not all the steps are applicable for each problem, it is highly dependent on the data we are working with, so maybe only a few steps might be required with your dataset. Generally they are :

- Data Quality Assessment
- Feature Aggregation
- Feature Sampling
- Dimensionality Reduction

- Feature Encoding

## ***Data Quality Assessment***

Because data is often taken from multiple sources which are normally not too reliable and that too in different formats, more than half our time is consumed in dealing with data quality issues when working on a machine learning problem. It is simply unrealistic to expect that the data will be perfect. There may be problems due to human error, limitations of measuring devices, or flaws in the data collection process. Let's go over a few of them and methods to deal with them :

### **1. Missing values :**

It is very much usual to have missing values in your dataset. It may have happened during data collection, or maybe due to some data validation rule, but regardless missing values must be taken into consideration.

- Eliminate rows with missing data :  
Simple and sometimes effective strategy. Fails if many objects have missing values. If a feature has mostly missing values, then that feature itself can also be eliminated.
- Estimate missing values :  
If only a reasonable percentage of values are missing, then we can also run simple [interpolation methods](#) to fill in those values. However, most common method of dealing with missing values is by filling them in with the mean, median or mode value of the respective feature.

### **2. Inconsistent values :**

We know that data can contain inconsistent values. Most probably we have already faced this issue at some point. For instance, the 'Address' field contains the 'Phone number'. It may be due to human error or maybe the information was misread while being scanned from a handwritten form.

- It is therefore always advised to perform data assessment like knowing what the data type of the features should be and whether it is the same for all the data objects.

### **3. Duplicate values :**

A dataset may include data objects which are duplicates of one another. It may happen when say the same person submits a form more than once. The term deduplication is often used to refer to the process of dealing with duplicates.

- In most cases, the duplicates are removed so as to not give that particular data object an advantage or *bias*, when running machine learning algorithms.

## ***Feature Aggregation***

Feature Aggregations are performed so as to take the aggregated values in order to put the data in a better perspective. Think of transactional data, suppose we have day-to-day transactions of a

product from recording the daily sales of that product in various store locations over the year. Aggregating the transactions to single store-wide monthly or yearly transactions will help us reducing hundreds or potentially thousands of transactions that occur daily at a specific store, thereby reducing the number of data objects.

- This results in reduction of memory consumption and processing time
- Aggregations provide us with a high-level view of the data as the behaviour of groups or aggregates is more stable than individual data objects

Aggregation from Monthly to Yearly

### ***Feature Sampling***

Sampling is a very common method for selecting a subset of the dataset that we are analyzing. In most cases, working with the complete dataset can turn out to be too expensive considering the memory and time constraints. Using a sampling algorithm can help us reduce the size of the dataset to a point where we can use a better, but more *expensive*, machine learning algorithm.

The key principle here is that the sampling should be done in such a manner that the sample generated should have approximately the same properties as the original dataset, meaning that the sample is *representative*. This involves choosing the correct sample size and sampling strategy.

*Simple Random Sampling* dictates that there is an equal probability of selecting any particular entity. It has two main variations as well :

- **Sampling without Replacement** : As each item is selected, it is removed from the set of all the objects that form the total dataset.
- **Sampling with Replacement** : Items are not removed from the total dataset after getting selected. This means they can get selected more than once.

Data Sampling ( ISTOCK.COM/KKOLOSOV)

Although Simple Random Sampling provides two great sampling techniques, it can fail to output a representative sample when the dataset includes object types which vary drastically in ratio. This can cause problems when the sample needs to have a proper representation of all object types, for example, when we have an *imbalanced* dataset.

An Imbalanced dataset is one where the number of instances of a class(es) are significantly higher than another class(es), thus leading to an imbalance and creating rarer class(es).

It is critical that the rarer classes be adequately represented in the sample. In these cases, there is another sampling technique which we can use, called *Stratified Sampling*, which begins with predefined groups of objects. There are different versions of Stratified Sampling too, with the simplest version suggesting equal number of objects be drawn from all the groups even though the groups are of different sizes. For more on sampling check out this article by

[Team AV](#)

## **A Data Scientist's Guide to 8 Types of Sampling Techniques**

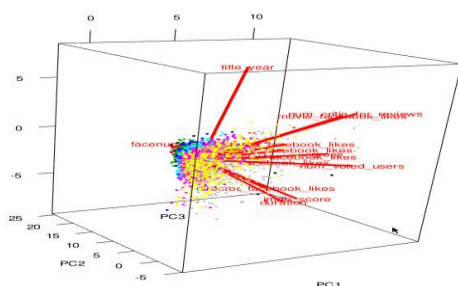
### ***Dimensionality Reduction***

Most real world datasets have a large number of features. For example, consider an image processing problem, we might have to deal with thousands of features, also called as *dimensions*. As the name suggests, dimensionality reduction aims to reduce the number of features - but not simply by selecting a sample of features from the *feature-set*, which is something else — Feature Subset Selection or simply Feature Selection.

Conceptually, *dimension* refers to the number of geometric planes the dataset lies in, which could be high so much so that it cannot be visualized with pen and paper. More the number of such planes, more is the complexity of the dataset.

### **The Curse of Dimensionality**

This refers to the phenomena that generally data analysis tasks become significantly harder as the dimensionality of the data increases. As the dimensionality increases, the number planes occupied by the data increases thus adding more and more sparsity to the data which is difficult to model and visualize.



Representation of components in different spaces

What dimension reduction essentially does is that it maps the dataset to a lower-dimensional space, which may very well be to a number of planes which can now be visualized, say 2D. The basic objective of techniques which are used for this purpose is to reduce the dimensionality of a dataset by creating new features which are a combination of the old features. In other words, the higher-dimensional feature-space is mapped to a lower-dimensional feature-space. [Principal Component Analysis](#) and [Singular Value Decomposition](#) are two widely accepted techniques.

A few major benefits of dimensionality reduction are :

- Data Analysis algorithms work better if the dimensionality of the dataset is lower. This is mainly because irrelevant features and noise have now been eliminated.
- The models which are built on top of lower dimensional data are more understandable and explainable.
- The data may now also get easier to visualize!  
Features can always be taken in pairs or triplets for visualization purposes, which makes more sense if the featureset is not that big.

### ***Feature Encoding***

As mentioned before, the whole purpose of data preprocessing is to *encode* the data in order to bring it to such a state that the machine now understands it.

Feature encoding is basically performing transformations on the data such that it can be easily accepted as input for machine learning algorithms while still retaining its original meaning.

There are some general norms or rules which are followed when performing feature encoding. For Continuous variables :

- **Nominal** : Any one-to-one mapping can be done which retains the meaning. For instance, a permutation of values like in [One-Hot Encoding](#).
- **Ordinal** : An order-preserving change of values. The notion of small, medium and large can be represented equally well with the help of a new function, that is,  $\langle \text{new\_value} = f(\text{old\_value}) \rangle$  - For example, {0, 1, 2} or maybe {1, 2, 3}.

	Name	Generation	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5
4	Octillery	Gen 2	0	1	0	0	0
5	Helioptile	Gen 6	0	0	0	0	0
6	Dialga	Gen 4	0	0	0	1	0
7	DeoxysDefense Forme	Gen 3	0	0	1	0	0
8	Rapidash	Gen 1	1	0	0	0	0
9	Swanna	Gen 5	0	0	0	0	1

One-hot encoding of the data

For Numeric variables:

- **Interval** : Simple mathematical transformation like using the equation  $\text{new\_value} = a * \text{old\_value} + b$ ,  $a$  and  $b$  being constants. For example, Fahrenheit and Celsius scales, which differ in their Zero values size of a unit, can be encoded in this manner.
- **Ratio** : These variables can be scaled to any particular measures, of course while still maintaining the meaning and ratio of their values. Simple mathematical transformations work in this case as well, like the transformation  $\text{new\_value} = a * \text{old\_value}$ . For, length can be measured in meters or feet, money can be taken in different currencies.

### ***Train / Validation / Test Split***

After feature encoding is done, our dataset is ready for the exciting machine learning algorithms! But before we start deciding the algorithm which should be used, it is always advised to split the dataset into 2 or sometimes 3 parts. Machine Learning algorithms, or any algorithm for that matter, has to be first trained on the data distribution available and then validated and tested, before it can be deployed to deal with real-world data.

**Training data** : This is the part on which your machine learning algorithms are actually trained to build a model. The model tries to *learn* the dataset and its various characteristics and intricacies, which also raises the issue of [Overfitting v/s Underfitting](#).

**Validation data** : This is the part of the dataset which is used to validate our various model fits. In simpler words, we use validation data to choose and improve our model hyperparameters. The model does not *learn* the validation set but uses it to get to a better state of hyperparameters.

**Test data** : This part of the dataset is used to test our model hypothesis. It is left untouched and unseen until the model and hyperparameters are decided, and only after that the model is applied on the test data to get an accurate measure of how it would perform when deployed on real-world data.



Data Split into parts

**Split Ratio** : Data is split as per a *split ratio* which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training purposes (usually the case) — For instance, training on textual data, image data, or video data usually involves thousands of features!

If the model has a lot of hyperparameters that can be tuned, then keeping a higher percentage of data for the validation set is advisable. Models with less number of hyperparameters are easy to tune and update, and so we can keep a smaller validation set.

Like many other things in Machine Learning, the split ratio is highly dependent on the problem we are trying to solve and must be decided after taking into account all the various details about the model and the dataset in hand.

### Scalability and real-time issues in data science

#### What are scalability issues?

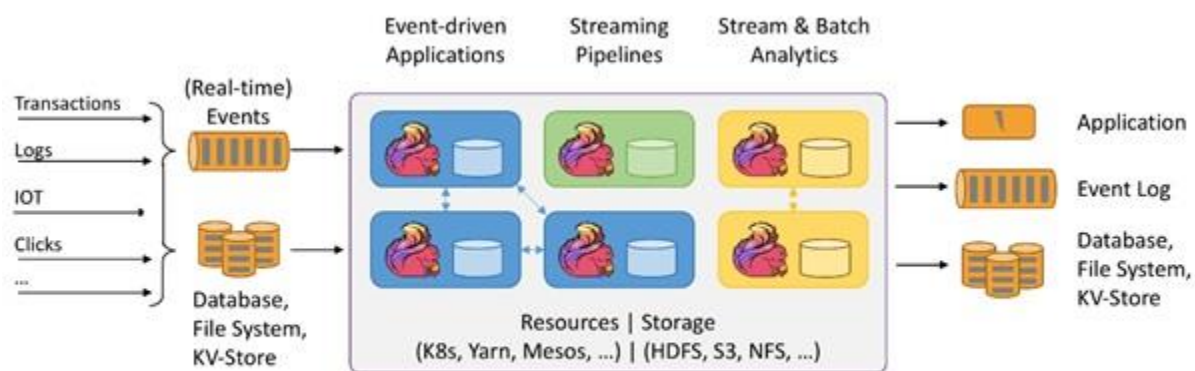
That's what, in the tech world is known as **scalability problems**. That is, the backend can't scale at the same speed the users pour into the application. The problem is that it's not only a problem of more users, but having users that interact more heavily with the site

#### What is scalability in data science?

The term '**scalability**' can be defined as the "capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth."

**Scalability** with respect to **data science** needs to reflect the hardware and software aspects, as well as the people and process

## What is Real-time data analytics? Top 5 Challenges and solutions



Real-time data analytics, one of the most crucial parts of [Big data analytics](#) today, is the most challenging part when it comes to the question of implementation for enterprises. Though real-time data analytics gives you a more in-depth insight into the business data by handling streaming data sources, there are multiple challenges associated with it. Because storing such a huge volume of data and analyzing them on a real-time basis is entirely a different ball game.

Additionally, real-time data processing often requires scalability, fault-tolerance, predictability, resiliency against stream imperfections and must be extensible. Before we dive into knowing what all challenges are there and their possible solution, let's discuss real-time analytics definitions.

### **What is Real-time data analytics?**

Real-time data analytics architecture allows analysis of data once the data becomes available. Hence, users can obtain insights immediately after the data enters their system. Thus it is all about the high availability of data and low response time. Furthermore, when analytics through batch data processing takes a longer time, like hours or even days, we can get instant insights using real-time analytics to yield results.

It saves both money and time for the business as the business can react without delay and prevent any unforeseen problems. Real-time analytics mainly follows lambda architecture or kappa architecture. However, one size may not fit all. Tools like Apache Flink or Spark streaming, Amazon Kinesis is a real-time analytics platform.

**Related post - [What is Real-time analytics and its benefits](#)**

### **Challenges of Real-time data analytics**

#### **Challenge #1 Real-time data analytics architecture**

Real-time data analytics architecture must be capable of processing data at high speed. But depending on the data source and type of data, the speed may vary from milliseconds to minutes, and the architecture should be built similarly. The second most important thing is that the architecture should handle the spike of data volume and be scaled up as and when required. Besides, the architecture should be able to capture real-time as well as offline analytics of data. At the same time, running real-time analytics and offline analytics may create conflicts. So, the designed architecture must be able to handle this conflict and address the fundamental architectural issues.



## **Challenge #2 Real-time is a contradictory term itself**

'Real-time' is a confusing as well contradictory term. It may be for instantaneous results for some stakeholders, and for others, it is okay to wait for several minutes. Thus, it is necessary to invest significant time and effort to clarify the requirement. Besides, it is necessary to bring your team at the same line so that they unanimously agree on the requirement of real-time, the type of data required for analysis, and the sources to be used for the data. This is the utmost required because unless the interpretations are clear, it may cause inconsistent requirements.

## **Challenge # 3 Understanding the need for internal process**

When an organization looks for or invest in real-time data analytics, there must be some internal motivation behind it. Ultimately it is to improve the internal process. However, when a team gets involved in real-time analytics, several tasks line up with it. Some of them are like –

- requirements gathering
- designing the solution's architecture,
- selecting the right technology stack,
- solving issues related to hardware and software

Thus to maximize the benefits of real-time data analytics, it is necessary to keep in mind that the whole process improves the internal process, and not the analysis is the ultimate goal.

## **Challenge #4 Change in an organization comes with many resistances**

Implementing real-time data analytics in an organization following traditional intelligence methods could sometimes be a huge challenge. Primarily the challenge comes from the end of existing employees. As real-time data analytics may open up new directions towards organizational goals and new opportunities, sometimes it seems like a disruption for the existing employees. Consequently, it results in resistance from the employee end towards the new change.

To avoid disruption, management should clarify the reasons for the shift to real-time analytics and the possible opportunities it brings, and convince the employees of these ideas. It is quite natural that many technical obstacles can come out during the change. Hence, appropriate training should be organized to build confidence among the employees. Furthermore, the organization must make sure that its employees understand the real-time data analytics system's benefits and prepare themselves to work with it successfully.

### **Challenge #5 New way of working must be implemented**

Real-time data analytics is a whole new model of working. In a traditional analytics system where organizations usually get their insights once in a week, real-time data analytics gives you insights every second. So, it is an entirely different approach to working and analysis. Similarly, the organization's work culture should be in line with this faster analysis method so that it can affect the business appropriately.

### **Final thought:**

Actionable metrics, like real-time data analytics, always help us to make better and smarter decisions. In real-time data analytics, action can be taken on data immediately so that data can be accessed within a few minutes after an event takes place. However, when you implement real-time analytics, you will face a variety of challenges. These challenges, as mentioned in the above sections, are not simple in reality. Moreover, they will not be solved unless proper action items are considered wisely.