Learning:

Types of Learning:

There are also some types of machine learning algorithms that are used in very specific use-cases, but three main methods are used today.

1. **Supervised Learning**

Supervised learning is one of the most basic types of machine learning. In this type, the machine learning algorithm is trained on labelled data. Even though the data needs to be labelled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances.

In supervised learning, the ML algorithm is given a small training dataset to work with. This training dataset is a smaller part of the bigger dataset and serves to give the algorithm a basic idea of the problem, solution, and data points to be dealt with. The training dataset is also very similar to the final dataset in its characteristics and provides the algorithm with the labelled parameters required for the problem.

The algorithm then finds relationships between the parameters given, essentially establishing a cause and effect relationship between the variables in the dataset. At the end of the training, the algorithm has an idea of how the data works and the relationship between the input and the output.

This solution is then deployed for use with the final dataset, which it learns from in the same way as the training dataset. This means that supervised machine learning algorithms will continue to improve even after being deployed, discovering new patterns and relationships as it trains itself on new data.

2. **Unsupervised Learning**

Unsupervised machine learning holds the advantage of being able to work with unlabeled data. This means that human labor is not required to make the dataset machine-readable, allowing much larger datasets to be worked on by the program.

In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings.

The creation of these hidden structures is what makes unsupervised learning algorithms versatile. Instead of a defined and set problem statement, unsupervised learning algorithms can adapt to the data by dynamically changing hidden structures. This offers more post-deployment development than supervised learning algorithms.

3. Reinforcement Learning

Reinforcement learning directly takes inspiration from how human beings learn from data in their lives. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favorable outputs are encouraged or 'reinforced', and non-favorable outputs are discouraged or 'punished'.

Based on the psychological concept of conditioning, reinforcement learning works by putting the algorithm in a work environment with an interpreter and a reward system. In every iteration of the algorithm, the output result is given to the interpreter, which decides whether the outcome is favorable or not.

In case of the program finding the correct solution, the interpreter reinforces the solution by providing a reward to the algorithm. If the outcome is not favorable, the algorithm is forced to reiterate until it finds a better result. In most cases, the reward system is directly tied to the effectiveness of the result.

In typical reinforcement learning use-cases, such as finding the shortest route between two points on a map, the solution is not an absolute value. Instead, it takes on a score of effectiveness, expressed in a percentage value. The higher this percentage value is, the more reward is given to the algorithm. Thus, the program is trained to give the best possible solution for the best possible reward.

Well defined learning problems:

Well Posed Learning Problem – A computer program is said to learn from experience E in context to some task T and some performance measure P, if its performance on T, as was measured by P, upgrades with experience E.

Any problem can be segregated as well-posed learning problem if it has three traits –

Task

Performance Measure

Experience

Certain examples that efficiently defines the well-posed learning problem are –

1. To better filter emails as spam or not

Task – Classifying emails as spam or not

Performance Measure – The fraction of emails accurately classified as spam or not spam

Experience – Observing you label emails as spam or not spam

2. A checkers learning problem

Task – Playing checkers game

Performance Measure – percent of games won against opposer

Experience – playing implementation games against itself

3. Handwriting Recognition Problem

Task – Acknowledging handwritten words within portrayal

Performance Measure – percent of words accurately classified

Experience – a directory of handwritten words with given classifications

4. A Robot Driving Problem

Task – driving on public four-lane highways using sight scanners

Performance Measure – average distance progressed before a fallacy

Experience – order of images and steering instructions noted down while observing a human driver

5. Fruit Prediction Problem

Task – forecasting different fruits for recognition

Performance Measure – able to predict maximum variety of fruits

Experience – training machine with the largest datasets of fruits images

6. Face Recognition Problem

Task – predicting different types of faces

Performance Measure – able to predict maximum types of faces

Experience – training machine with maximum amount of datasets of different face images

7. Automatic Translation of documents

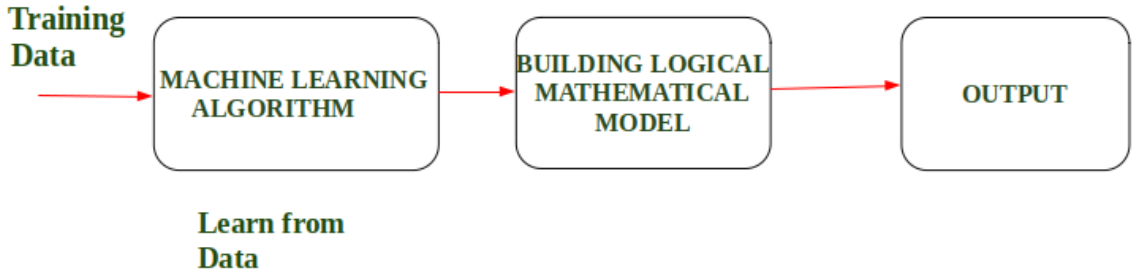
Task – translating one type of language used in a document to other language

Performance Measure – able to convert one language to other efficiently

Experience – training machine with a large dataset of different types of languages

Designing a Learning System:

According to Arthur Samuel “Machine Learning enables a Machine to Automatically learn from Data, improve performance from an Experience and predict things without explicitly programmed.”



In Simple Words, When we fed the Training Data to Machine Learning Algorithm, this algorithm will produce a mathematical model and with the help of the mathematical model, the machine will make a prediction and take a decision without being explicitly programmed. Also, during training data, the more machine will work with it the more it will get experience and the more efficient result is produced.

Example : In Driverless Car, the training data is fed to Algorithm like how to Drive Car in Highway, Busy and Narrow Street with factors like speed limit, parking, stop at signal etc. After that, a Logical and Mathematical model is created on the basis of that and after that, the car will work according to the logical model. Also, the more data the data is fed the more efficient output is produced.

Designing a Learning System in Machine Learning :

According to Tom Mitchell, “A computer program is said to be learning from experience (E), with respect to some task (T). Thus, the performance measure (P) is the performance at task T, which is measured by P, and it improves with experience E.”

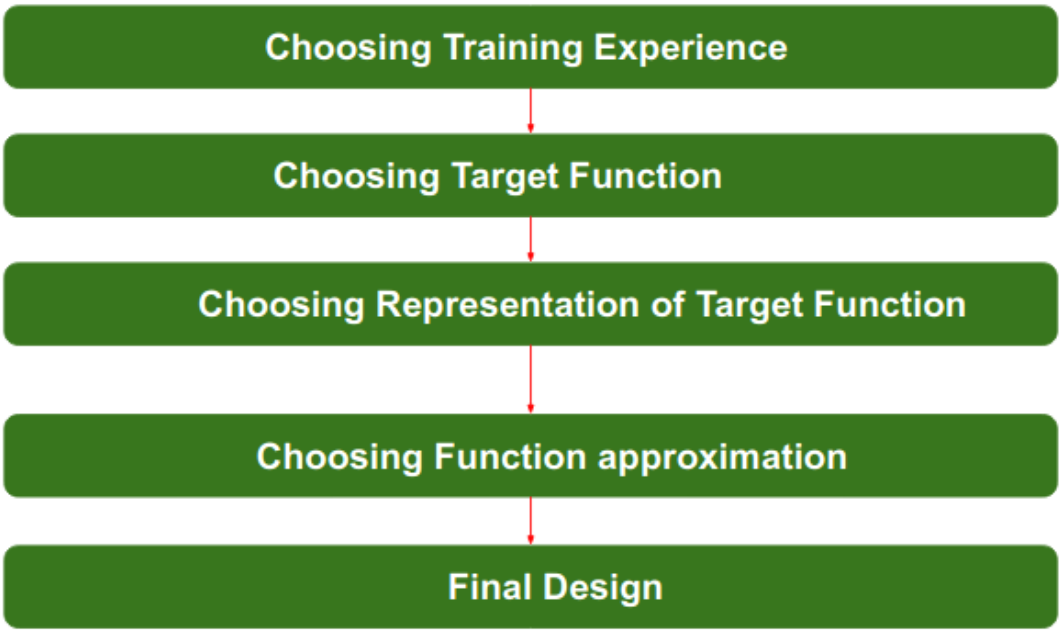
Example: In Spam E-Mail detection,

Task, T: To classify mails into Spam or Not Spam.

Performance measure, P: Total percent of mails being correctly classified as being “Spam” or “Not Spam”.

Experience, E: Set of Mails with label “Spam”

Steps for Designing Learning System are:



Step 1) Choosing the Training Experience: The very important and first task is to choose the training data or training experience which will be fed to the Machine Learning Algorithm. It is important to note that the data or experience that we fed to the algorithm must have a significant impact on the Success or Failure of the Model. So Training data or experience should be chosen wisely.

Below are the attributes which will impact on Success and Failure of Data:

The training experience will be able to provide direct or indirect feedback regarding choices. For example: While Playing chess the training data will provide feedback to itself like instead of this move if this is chosen the chances of success increases.

Second important attribute is the degree to which the learner will control the sequences of training examples. For example: when training data is fed to the machine then at that time accuracy is very less but when it gains experience while playing again and again with itself or opponent the machine algorithm will get feedback and control the chess game accordingly.

Third important attribute is how it will represent the distribution of examples over which performance will be measured. For example, a Machine learning algorithm will get experience while going through a number of different cases and different examples. Thus, Machine Learning Algorithm will get more and more experience by passing through more and more examples and hence its performance will increase.

Step 2- Choosing target function: The next important step is choosing the target function. It means according to the knowledge fed to the algorithm the machine learning will choose NextMove function which will describe what type of legal moves should be taken. For

example : While playing chess with the opponent, when opponent will play then the machine learning algorithm will decide what be the number of possible legal moves taken in order to get success.

Step 3- Choosing Representation for Target function: When the machine algorithm will know all the possible legal moves the next step is to choose the optimized move using any representation i.e. using linear Equations, Hierarchical Graph Representation, Tabular form etc. The NextMove function will move the Target move like out of these move which will provide more success rate. For Example : while playing chess machine have 4 possible moves, so the machine will choose that optimized move which will provide success to it.

Step 4- Choosing Function Approximation Algorithm: An optimized move cannot be chosen just with the training data. The training data had to go through with set of example and through these examples the training data will approximates which steps are chosen and after that machine will provide feedback on it. For Example : When a training data of Playing chess is fed to algorithm so at that time it is not machine algorithm will fail or get success and again from that failure or success it will measure while next move what step should be chosen and what is its success rate.

Step 5- Final Design: The final design is created at last when system goes from number of examples , failures and success , correct and incorrect decision and what will be the next step etc. Example: DeepBlue is an intelligent computer which is ML-based won chess game against the chess expert Garry Kasparov, and it became the first computer which had beaten a human chess expert.

History of ML:

The term Machine Learning (ML) was first used by Arthur Samuel, one of the pioneers of Artificial Intelligence at IBM, in 1959. The name came from researchers who observed computers recognizing patterns and developed the theory that computers could learn without being programmed to perform specific tasks.

Introduction of Machine Learning Approaches:

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches:

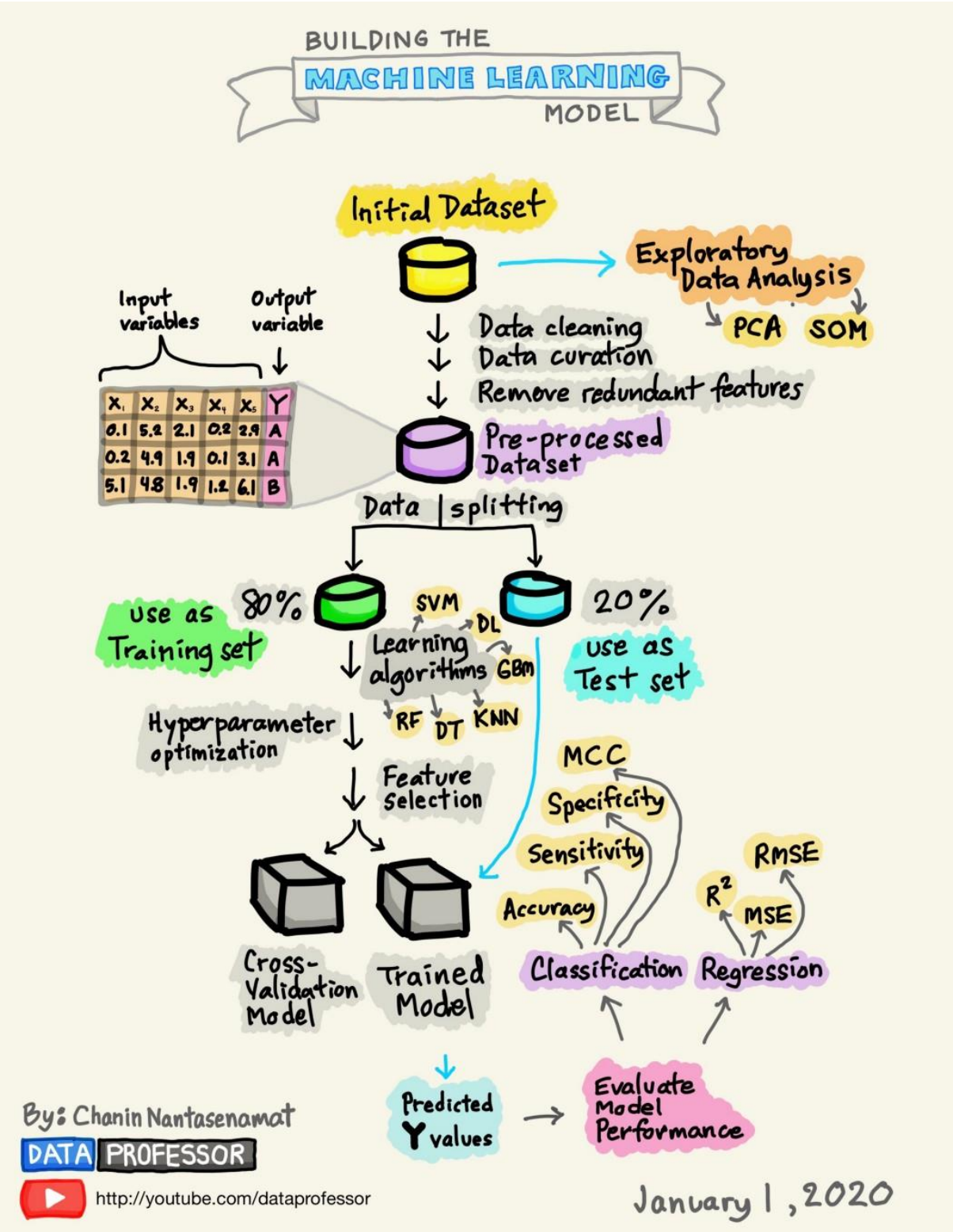
supervised learning,

unsupervised learning,

semi-supervised learning and

reinforcement learning.

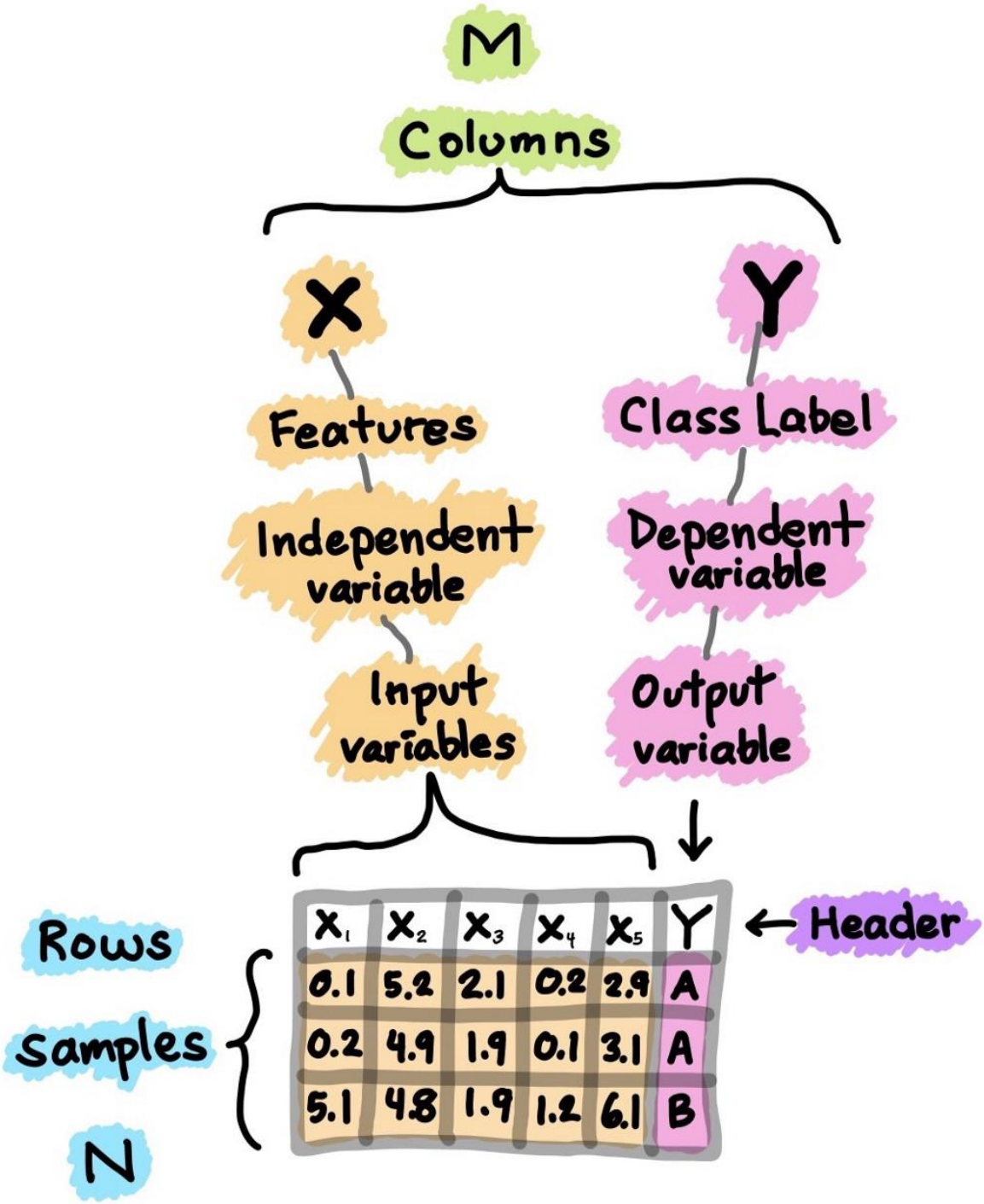
Introduction to Model Building:



Dataset

A dataset is the starting point in your journey of building the machine learning model. Simply put, the dataset is essentially an M×N matrix where M represents the columns (features) and N the rows (samples).

Columns can be broken down to X and Y. Firstly, X is synonymous with several similar terms such as features, independent variables and input variables. Secondly, Y is also synonymous with several terms namely class label, dependent variable and output variable.



It should be noted that a dataset that can be used for *supervised learning* (can perform either regression or classification) would contain both X and Y whereas a dataset that can be used for *unsupervised learning* will only have X.

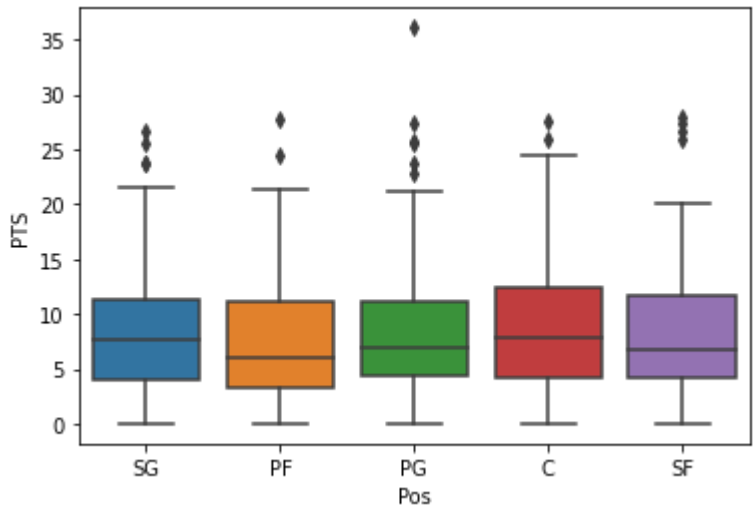
Moreover, if Y contains quantitative values then the dataset (comprising of X and Y) can be used for *regression* tasks whereas if Y contains qualitative values then the dataset (comprising of X and Y) can be used for *classification tasks*.

Exploratory Data Analysis

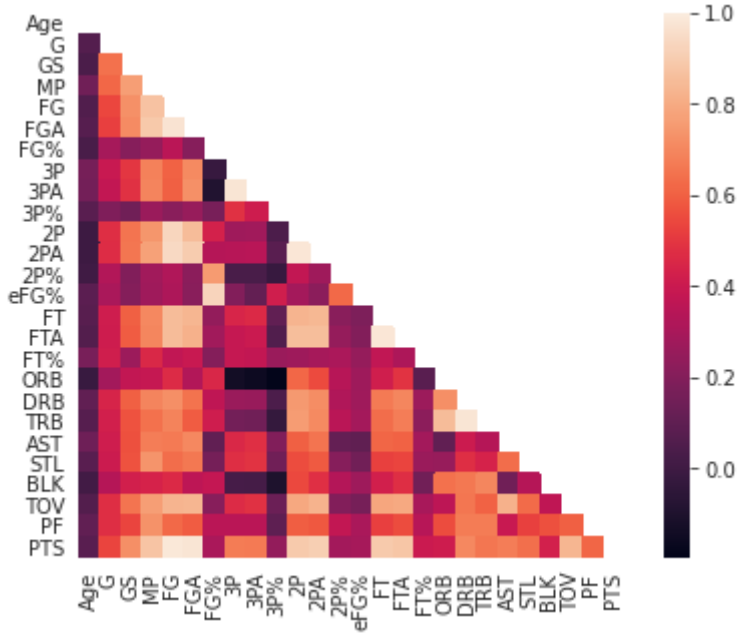
Exploratory data analysis (EDA) is performed in order to gain a preliminary understanding and allow us to get acquainted with the dataset. In a typical data science project, one of the first things that I would do is *"eyeballing the data"* by performing EDA so as to gain a better understanding of the data.

Three major EDA approaches that I normally use includes:

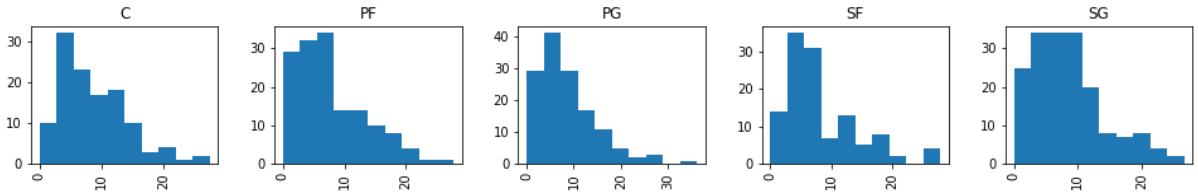
- Descriptive statistics — Mean, median, mode, standard deviation
- Data visualisations — Heat maps (discerning feature intra-correlation), box plot (visualize group differences), scatter plots (visualize correlations between features), principal component analysis (visualize distribution of clusters presented in the dataset), etc.
- Data shaping — Pivoting data, grouping data, filtering data, etc.



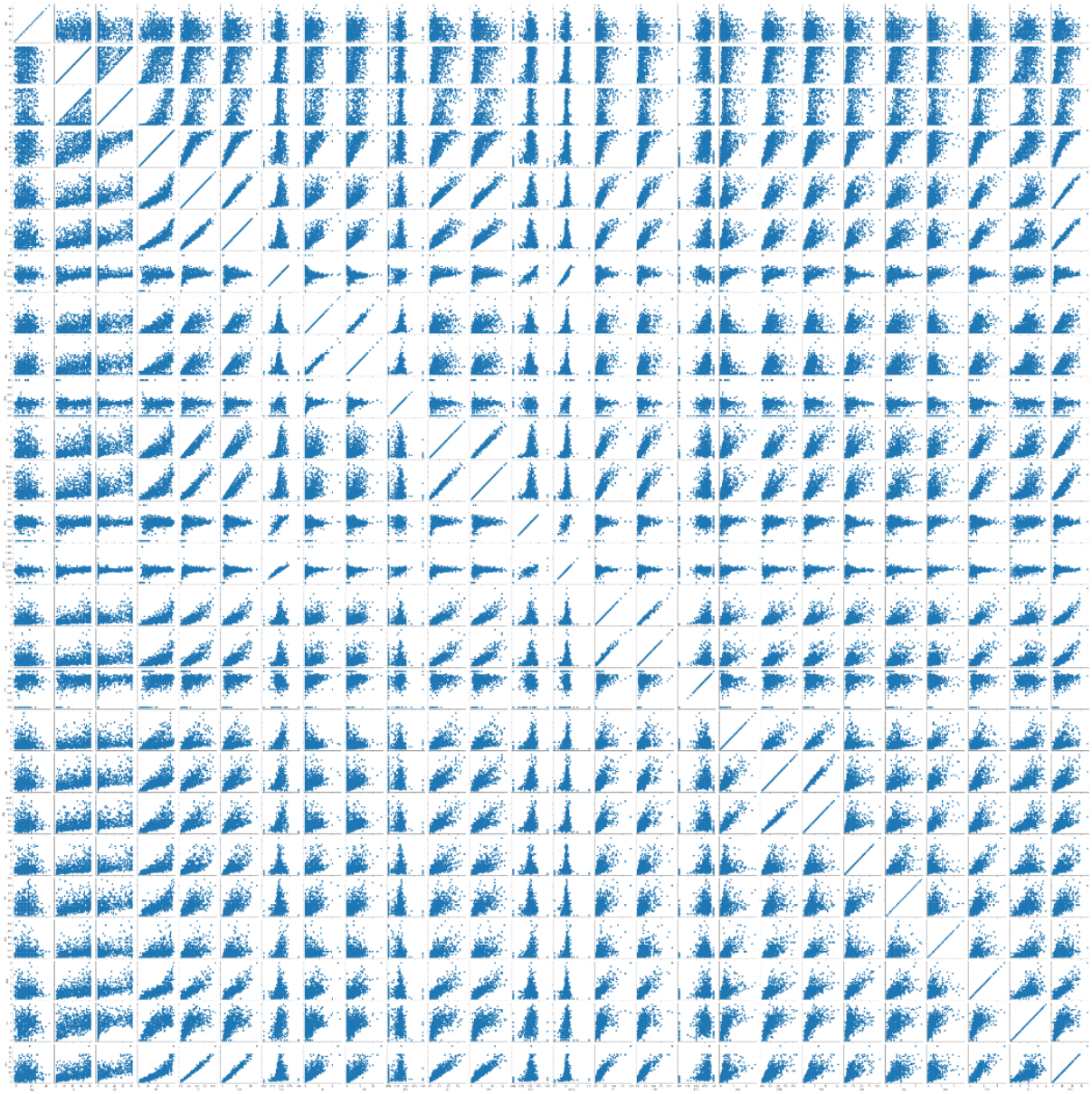
Example box plot of NBA player stats data. Plot obtained from the Jupyter notebook on Data Professor GitHub.



Example correlation heatmap of NBA player stats data. Plot obtained from the Jupyter notebook on Data Professor GitHub.



Example histogram plot of NBA player stats data. Plot obtained from the Jupyter notebook on Data Professor GitHub.



Example scatter plot of NBA player stats data. Plot obtained from the Jupyter notebook on Data Professor GitHub.

For more step-by-step tutorial on performing these exploratory data analysis in Python, please check out the video I made on the Data Professor YouTube channel.

Data Pre-Processing

Data pre-processing (also known as data cleaning, data wrangling or data munging) is the process by which the data is subjected to various checks and scrutiny in order to remedy issues of missing values, spelling errors, normalizing/standardizing values such that they are comparable, transforming data (e.g. logarithmic transformation), etc.

"Garbage in, Garbage out."

— George Fuechsel

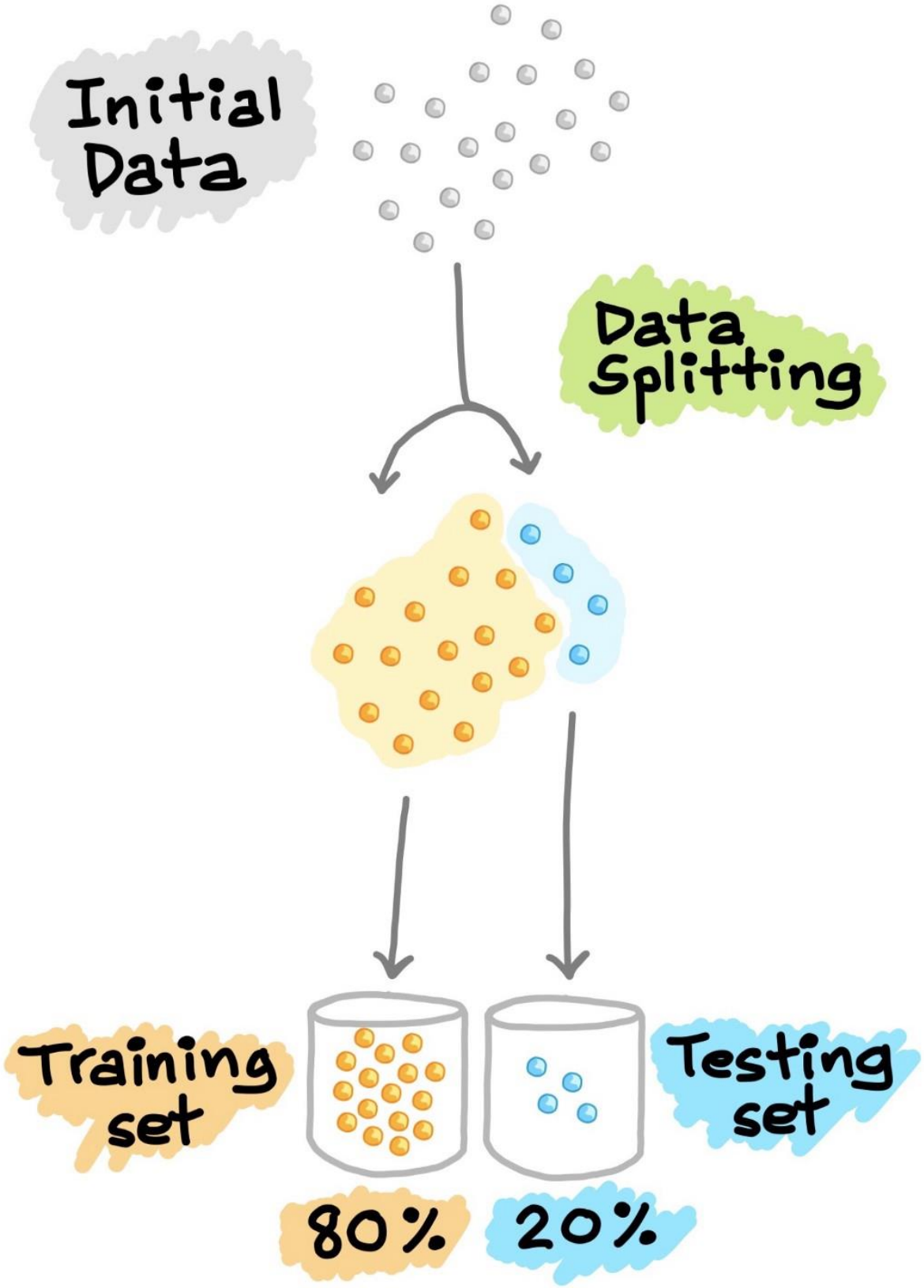
As the above quote suggests, the quality of data is going to exert a big impact on the quality of the generated model. Therefore, to achieve the highest model quality, significant effort should be spent in the data pre-processing phase. It is said that data pre-processing could easily account for 80% of the time spent on data science projects while the actual model building phase and subsequent post-model analysis account for the remaining 20%.

Data Splitting

Train-Test Split

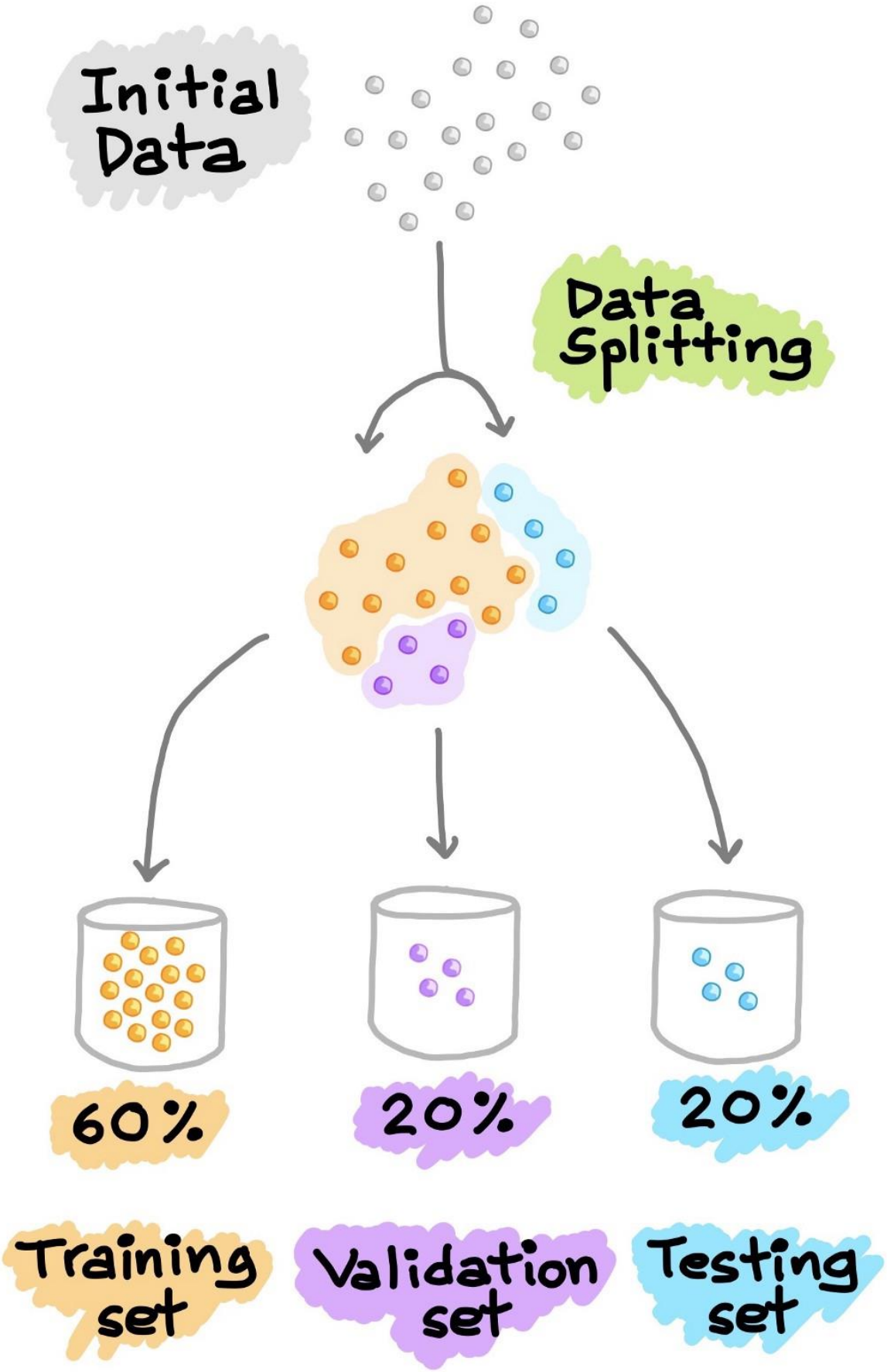
In the development of machine learning models, it is desirable that the trained model perform well on new, unseen data. In order to simulate the new, unseen data, the available data is subjected to *data splitting* whereby it is split to 2 portions (sometimes referred to as the *train-test split*). Particularly, the first portion is the larger data subset that is used as the *training set* (such as accounting for 80% of the original data) and the second is normally a smaller subset and used as the *testing set* (the remaining 20% of the data). It should be noted that such data split is performed once.

Next, the training set is used to build a predictive model and such *trained model* is then applied on the testing set (*i.e.* serving as the new, unseen data) to make predictions. Selection of the best model is made on the basis of the model's performance on the testing set and in efforts to obtain the best possible model, hyperparameter optimization may also be performed.



Train-Validation-Test Split

Another common approach for *data splitting* is to split the data to 3 portions: (1) training set, (2) validation set and (3) testing set. Similar to what was explained above, the training set is used to build a predictive model and is also evaluated on the *validation set* whereby predictions are made, model tuning can be made (e.g. hyperparameter optimization) and selection of the best performing model based on results of the validation set. As we can see, similar to what was performed above to the test set, here we do the same procedure on the validation set instead. Notice that the *testing set* is not involved in any of the model building and preparation. Thus, the testing set can truly act as the new, unseen data. A more in-depth treatment of this topic is provided by Google's Machine Learning Crash Course.

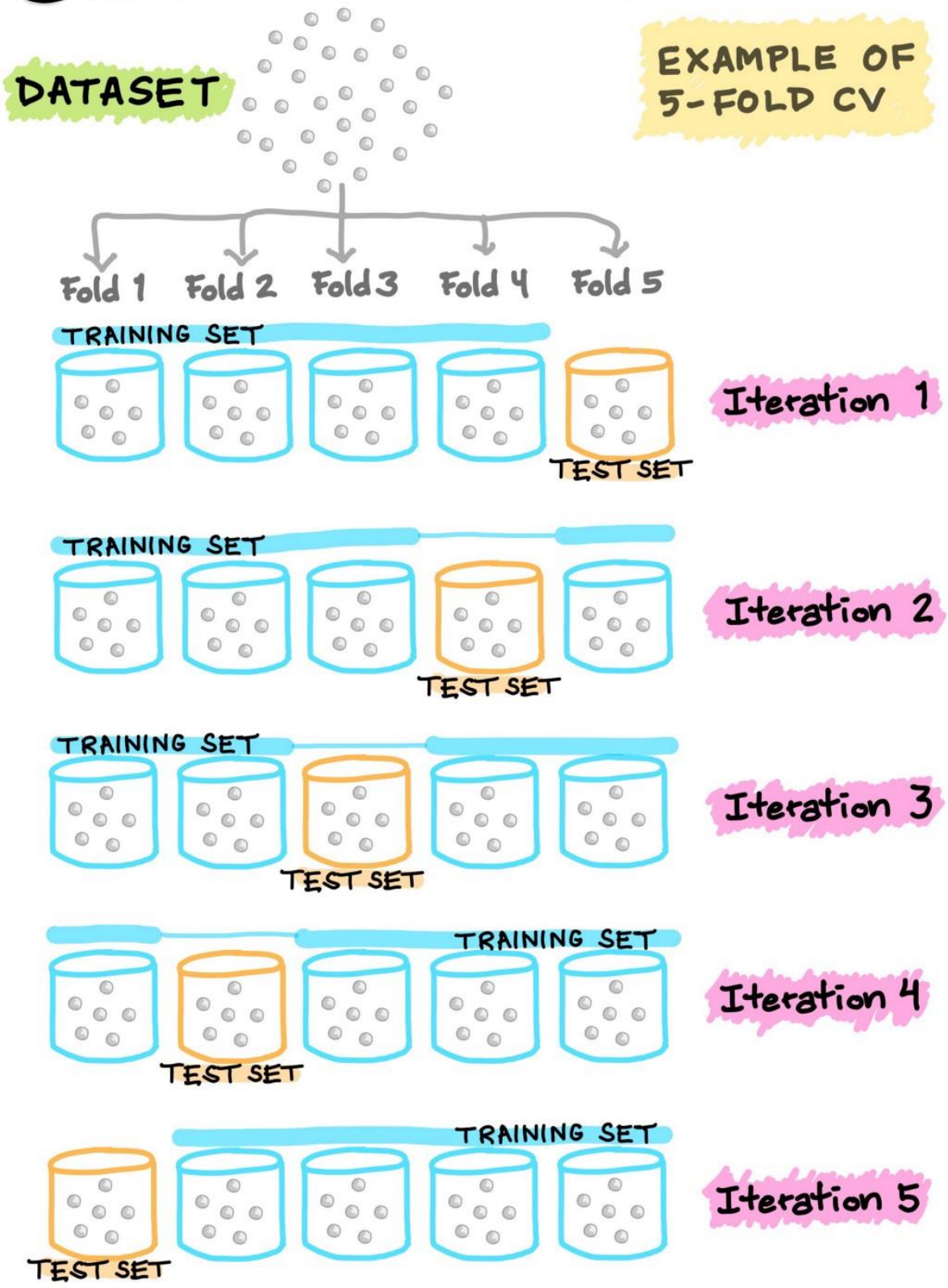

Initial Data
Data Splitting
60%
20%
20%
Training set
Validation set
Testing set

Cross-Validation

In order to make the most economical use of the available data, an *N-fold cross-validation (CV)* is normally used whereby the dataset is partitioned to *N* folds (*i.e.* commonly 5-fold or 10-fold CV are used). In such *N*-fold CV, one of the fold is left out as the testing data while the remaining folds are used as the training data for model building.

For example, in a 5-fold CV, 1 fold is left out and used as the testing data while the remaining 4 folds are pooled together and used as the training data for model building. The trained model is then applied on the aforementioned left-out fold (*i.e.* the test data). This process is carried out iteratively until all folds had a chance to be left out as the testing data. As a result, we will have built 5 models (i.e. where each of the 5 folds have been left out as the testing set) where each of the 5 models contain associated performance metrics (which we will discuss soon in the forthcoming section). Finally, the metric values are based on the average performance computed from the 5 models.

CROSS-VALIDATION



In situations when *N* is equal to the number of data samples, we call this *leave-one-out cross-validation*. In this type of CV, each data sample represents a fold. For example, if *N* is equal to 30 then there are 30 folds (1 sample per fold). As in any other *N*-fold CV, 1 fold is left out as the testing set while the remaining 29 folds are used to build the model. Next, the built model is applied to make prediction on the left-out fold. As before, this process is performed iteratively for a total of 30 times; and the average performance from the 30 models are computed and used as the CV performance metric.

Model Building

Now, comes the fun part where we finally get to use the meticulously prepared data for model building. Depending on the data type (qualitative or quantitative) of the target variable (commonly referred to as the Y variable) we are either going to be building a classification (if Y is qualitative) or regression (if Y is quantitative) model.

Learning Algorithms

Machine learning algorithms could be broadly categorised to one of three types:

1. *Supervised learning* — is a machine learning task that establishes the mathematical relationship between input X and output Y variables. Such X, Y pair constitutes the labeled data that are used for model building in an effort to learn how to predict the output from the input.

2. *Unsupervised learning* — is a machine learning task that makes use of only the input X variables. Such X variables are unlabeled data that the learning algorithm uses in modeling the inherent structure of the data.

3. *Reinforcement learning* — is a machine learning task that decides on the next course of action and it does this by learning through trial and error in an effort to maximize the reward.

Sensitivity Analysis:

Sensitivity analysis is a popular feature selection approach employed to identify the important features in a dataset. In sensitivity analysis, each input feature is perturbed one-at-a-time and the response of the machine learning model is examined to determine the feature's rank.

Sensitivity analysis is a useful tool for deep learning developers as well as users such as clinicians. It extends their toolbox, enabling and improving interpretability of segmentation models. Enhancing our understanding of neural networks through sensitivity analysis also assists in decision making.

Underfitting and Overfitting:

Underfitting:

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data. (It's just like trying to fit undersized pants!) Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with fewer non-linear data. In such cases, the rules of the machine learning model are too easy and flexible to be applied to such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

In a nutshell, Underfitting refers to a model that can neither performs well on the training data nor generalize to new data.

Reasons for Underfitting:

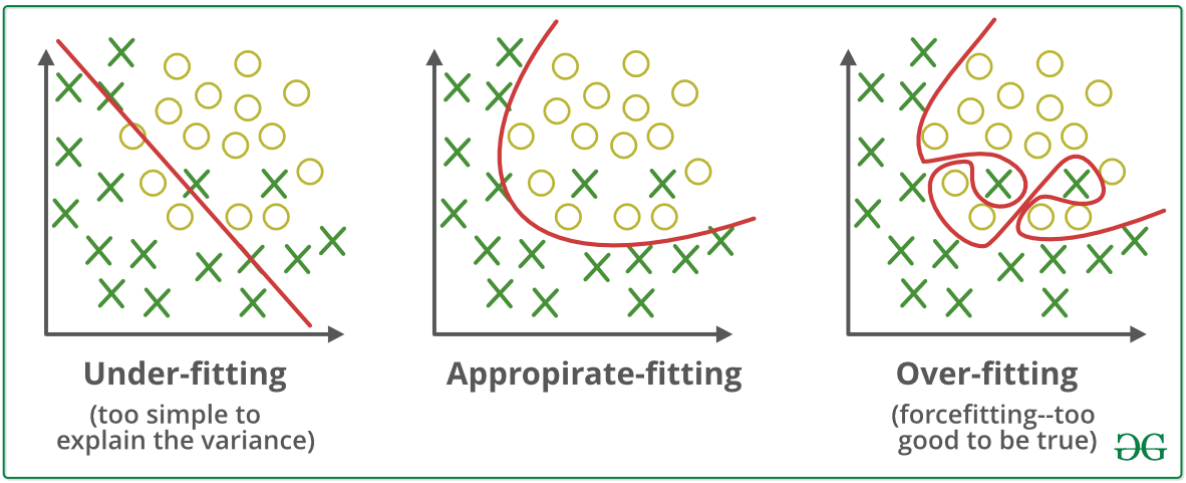
- High bias and low variance
- The size of the training dataset used is not enough.
- The model is too simple.
- Training data is not cleaned and also contains noise in it.
- Techniques to reduce underfitting:

Increase model complexity

Increase the number of features, performing feature engineering

Remove noise from the data.

Increase the number of epochs or increase the duration of training to get better results.



Overfitting: A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

Reasons for Overfitting are as follows:

- High variance and low bias
- The model is too complex
- The size of the training data
- Techniques to reduce overfitting:

Increase training data.

Reduce model complexity.

Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).

Ridge Regularization and Lasso Regularization

Use dropout for neural networks to tackle overfitting.

Bias and Variance:

Bias: Assumptions made by a model to make a function easier to learn. It is actually the error rate of the training data. When the error rate has a high value, we call it High Bias and when the error rate has a low value, we call it low Bias.

Variance: The difference between the error rate of training data and testing data is called variance. If the difference is high then it's called high variance and when the difference of errors is low then it's called low variance. Usually, we want to make a low variance for generalized our model.

Concept Learning Task

Find – S Algorithms:

Introduction :

The find-S algorithm is a basic concept learning algorithm in machine learning. The find-S algorithm finds the most specific hypothesis that fits all the positive examples. We have to note here that the algorithm considers only those positive training example. The find-S algorithm starts with the most specific hypothesis and generalizes this hypothesis each time it fails to classify an observed positive training data. Hence, the Find-S algorithm moves from the most specific hypothesis to the most general hypothesis.

Important Representation :

- ? indicates that any value is acceptable for the attribute.
- specify a single required value (e.g., Cold) for the attribute.
- ϕindicates that no value is acceptable.
- The most general hypothesis is represented by: {?, ?, ?, ?, ?, ?}
- The most specific hypothesis is represented by: {ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}

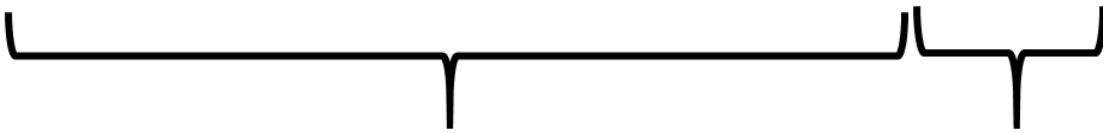
Steps Involved In Find-S :

a. Start with the most specific hypothesis.
b. h = {ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}
c. Take the next example and if it is negative, then no changes occur to the hypothesis.
d. If the example is positive and we find that our initial hypothesis is too specific then we update our current hypothesis to a general condition.
e. Keep repeating the above steps till all the training examples are complete.
f. After we have completed all the training examples we will have the final hypothesis when can use to classify the new examples.

Example :

Consider the following data set having the data about which particular seeds are poisonous.

EXAMPLE	COLOR	TOUGHNESS	FUNGUS	APPEARANCE	POISONOUS
1.	GREEN	HARD	NO	WRINKELD	YES
2.	GREEN	HARD	YES	SMOOTH	NO
3.	BROWN	SOFT	NO	WRINKLED	NO
4.	ORANGE	HARD	NO	WRINKLED	YES
5.	GREEN	SOFT	YES	SMOOTH	YES
6.	GREEN	HARD	YES	WRINKLED	YES
7.	ORANGE	HARD	NO	WRINKLED	YES



ATTRIBUTES ON WHICH THE CONCEPT DEPENDS ON **CONCEPT**

First, we consider the hypothesis to be a more specific hypothesis. Hence, our hypothesis would be :

h = {ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}

Consider example 1 :

The data in example 1 is { GREEN, HARD, NO, WRINKLED }. We see that our initial hypothesis is more specific and we have to generalize it for this example. Hence, the hypothesis becomes :

h = { GREEN, HARD, NO, WRINKLED }

Consider example 2 :

Here we see that this example has a negative outcome. Hence we neglect this example and our hypothesis remains the same.

h = { GREEN, HARD, NO, WRINKLED }

Consider example 3 :

Here we see that this example has a negative outcome. Hence we neglect this example and our hypothesis remains the same.

h = { GREEN, HARD, NO, WRINKLED }

Consider example 4 :

The data present in example 4 is { ORANGE, HARD, NO, WRINKLED }. We compare every single attribute with the initial data and if any mismatch is found we replace that particular attribute with a general case (" ? "). After doing the process the hypothesis becomes :

h = { ?, HARD, NO, WRINKLED }

Consider example 5 :

The data present in example 5 is { GREEN, SOFT, YES, SMOOTH }. We compare every single attribute with the initial data and if any mismatch is found we replace that particular attribute with a general case (" ? "). After doing the process the hypothesis becomes :

h = { ?, ?, ?, ? }

Since we have reached a point where all the attributes in our hypothesis have the general condition, example 6 and example 7 would result in the same hypothesizes with all general attributes.

h = { ?, ?, ?, ? }

Hence, for the given data the final hypothesis would be :

Final Hyposthesis: h = { ?, ?, ?, ? }

Algorithm :

Machine-Learning-Course

1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x

   For each attribute constraint a, in h

      If the constraint a, is satisfied by x

      Then do nothing

      Else replace a, in h by the next more general constraint that is satisfied by x

4. Output hypothesis h

Version Space and Candidate Elimination Algorithm:

The candidate elimination algorithm incrementally builds the version space given a hypothesis space H and a set E of examples. The examples are added one by one; each example possibly shrinks the version space by removing the hypotheses that are inconsistent with the example. The candidate elimination algorithm does this by updating the general and specific boundary for each new example.

You can consider this as an extended form of Find-S algorithm.

Consider both positive and negative examples.

Actually, positive examples are used here as Find-S algorithm (Basically they are generalizing from the specification).

While the negative example is specified from generalize form.

Terms Used:

Concept learning: Concept learning is basically learning task of the machine (Learn by Train data)

General Hypothesis: Not Specifying features to learn the machine.

G = {‘?’, ‘?’,’?’,’?’…}: Number of attributes

Specific Hypothesis: Specifying features to learn machine (Specific feature)

S= {‘pi’,’pi’,’pi’…}: Number of pi depends on number of attributes.

Version Space: It is intermediate of general hypothesis and Specific hypothesis. It not only just written one hypothesis but a set of all possible hypothesis based on training data-set.

Machine-Learning-Course

Algorithm:

Step1: Load Data set

Step2: Initialize General Hypothesis and Specific Hypothesis.

Step3: For each training example

Step4: If example is positive example

```
if attribute_value == hypothesis_value:
```

Do nothing

else:

replace attribute value with '?' (Basically generalizing it)

Step5: If example is Negative example

Make generalize hypothesis more specific.

Example:

Consider the dataset given below:

Sky	Temperature	Humid	Wind	Water	Forest	Output
sunny	warm	normal	strong	warm	same	yes
sunny	warm	high	strong	warm	same	yes
rainy	cold	high	strong	warm	change	no
sunny	warm	high	strong	cool	change	yes

Algorithmic steps:

Initially : G = [[?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?],

[?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?]]

S = [Null, Null, Null, Null, Null, Null]

For instance 1 : <'sunny','warm','normal','strong','warm ','same'> and positive output.

G1 = G

S1 = ['sunny','warm','normal','strong','warm ','same']

For instance 2 : <'sunny','warm','high','strong','warm ','same'> and positive output.

G2 = G

S2 = ['sunny','warm',?,'strong','warm ','same']

For instance 3 : <'rainy','cold','high','strong','warm ','change'> and negative output.

G3 = [['sunny', ?, ?, ?, ?, ?], [?, 'warm', ?, ?, ?, ?], [?, ?, ?, ?, ?, ?],

[?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, 'same']]

S3 = S2

For instance 4 : <'sunny','warm','high','strong','cool','change'> and positive output.

G4 = G3

S4 = ['sunny','warm',?,'strong', ?, ?]

At last, by synchronizing the G4 and S4 algorithm produce the output.

Output :

G = [['sunny', ?, ?, ?, ?, ?], [?, 'warm', ?, ?, ?, ?]]

S = ['sunny','warm',?,'strong', ?, ?]

Inductive Bias:

Issues in Machine Learning:

Although machine learning is being used in every industry and helps organizations make more informed and data-driven choices that are more effective than classical methodologies, it still has so many problems that cannot be ignored. Here are some common issues in Machine Learning that professionals face to inculcate ML skills and create an application from scratch.

1**. Inadequate Training Data**

The major issue that comes while using machine learning algorithms is the lack of quality as well as quantity of data. Although data plays a vital role in the processing of machine learning algorithms, many data scientists claim that inadequate data, noisy data, and unclean data are extremely exhausting the machine learning algorithms. For example, a simple task requires thousands of sample data, and an advanced task such as speech or image recognition needs

millions of sample data examples. Further, data quality is also important for the algorithms to work ideally, but the absence of data quality is also found in Machine Learning applications. Data quality can be affected by some factors as follows:

Noisy Data- It is responsible for an inaccurate prediction that affects the decision as well as accuracy in classification tasks.

Incorrect data- It is also responsible for faulty programming and results obtained in machine learning models. Hence, incorrect data may affect the accuracy of the results also.

Generalizing of output data- Sometimes, it is also found that generalizing output data becomes complex, which results in comparatively poor future actions.

2. Poor quality of data

As we have discussed above, data plays a significant role in machine learning, and it must be of good quality as well. Noisy data, incomplete data, inaccurate data, and unclean data lead to less accuracy in classification and low-quality results. Hence, data quality can also be considered as a major common problem while processing machine learning algorithms.

3. Non-representative training data

To make sure our training model is generalized well or not, we have to ensure that sample training data must be representative of new cases that we need to generalize. The training data must cover all cases that are already occurred as well as occurring.

Further, if we are using non-representative training data in the model, it results in less accurate predictions. A machine learning model is said to be ideal if it predicts well for generalized cases and provides accurate decisions. If there is less training data, then there will be a sampling noise in the model, called the non-representative training set. It won't be accurate in predictions. To overcome this, it will be biased against one class or a group.

Hence, we should use representative data in training to protect against being biased and make accurate predictions without any drift.

4. Overfitting and Underfitting

Overfitting is one of the most common issues faced by Machine Learning engineers and data scientists. Whenever a machine learning model is trained with a huge amount of data, it starts capturing noise and inaccurate data into the training data set. It negatively affects the performance of the model. Let's understand with a simple example where we have a few training data sets such as 1000 mangoes, 1000 apples, 1000 bananas, and 5000 papayas. Then there is a considerable probability of identification of an apple as papaya because we have a massive amount of biased data in the training data set; hence prediction got negatively affected.

The main reason behind overfitting is using non-linear methods used in machine learning algorithms as they build non-realistic data models. We can overcome overfitting by using linear and parametric algorithms in the machine learning models.

Underfitting:

Underfitting is just the opposite of overfitting. Whenever a machine learning model is trained with fewer amounts of data, and as a result, it provides incomplete and inaccurate data and destroys the accuracy of the machine learning model.

Underfitting occurs when our model is too simple to understand the base structure of the data, just like an undersized pant. This generally happens when we have limited data into the data set, and we try to build a linear model with non-linear data. In such scenarios, the complexity of the model destroys, and rules of the machine learning model become too easy to be applied on this data set, and the model starts doing wrong predictions as well.

5. Monitoring and maintenance

As we know that generalized output data is mandatory for any machine learning model; hence, regular monitoring and maintenance become compulsory for the same. Different results for different actions require data change; hence editing of codes as well as resources for monitoring them also become necessary.

6. Getting bad recommendations

A machine learning model operates under a specific context which results in bad recommendations and concept drift in the model. Let's understand with an example where at a specific time customer is looking for some gadgets, but now customer requirement changed over time but still machine learning model showing same recommendations to the customer while customer expectation has been changed. This incident is called a Data Drift. It generally occurs when new data is introduced or interpretation of data changes. However, we can overcome this by regularly updating and monitoring data according to the expectations.

7. Lack of skilled resources

Although Machine Learning and Artificial Intelligence are continuously growing in the market, still these industries are fresher in comparison to others. The absence of skilled resources in the form of manpower is also an issue. Hence, we need manpower having in-depth knowledge of mathematics, science, and technologies for developing and managing scientific substances for machine learning.

8. Customer Segmentation

Customer segmentation is also an important issue while developing a machine learning algorithm. To identify the customers who paid for the recommendations shown by the model

and who don't even check them. Hence, an algorithm is necessary to recognize the customer behavior and trigger a relevant recommendation for the user based on past experience.

9. Process Complexity of Machine Learning

The machine learning process is very complex, which is also another major issue faced by machine learning engineers and data scientists. However, Machine Learning and Artificial Intelligence are very new technologies but are still in an experimental phase and continuously being changing over time. There is the majority of hits and trial experiments; hence the probability of error is higher than expected. Further, it also includes analyzing the data, removing data bias, training data, applying complex mathematical calculations, etc., making the procedure more complicated and quite tedious.

10. Data Bias

Data Biasing is also found a big challenge in Machine Learning. These errors exist when certain elements of the dataset are heavily weighted or need more importance than others. Biased data leads to inaccurate results, skewed outcomes, and other analytical errors. However, we can resolve this error by determining where data is actually biased in the dataset. Further, take necessary steps to reduce it.

Methods to remove Data Bias:

Research more for customer segmentation.

Be aware of your general use cases and potential outliers.

Combine inputs from multiple sources to ensure data diversity.

Include bias testing in the development process.

Analyze data regularly and keep tracking errors to resolve them easily.

Review the collected and annotated data.

Use multi-pass annotation such as sentiment analysis, content moderation, and intent recognition.

11. Lack of Explainability

This basically means the outputs cannot be easily comprehended as it is programmed in specific ways to deliver for certain conditions. Hence, a lack of explainability is also found in machine learning algorithms which reduce the credibility of the algorithms.

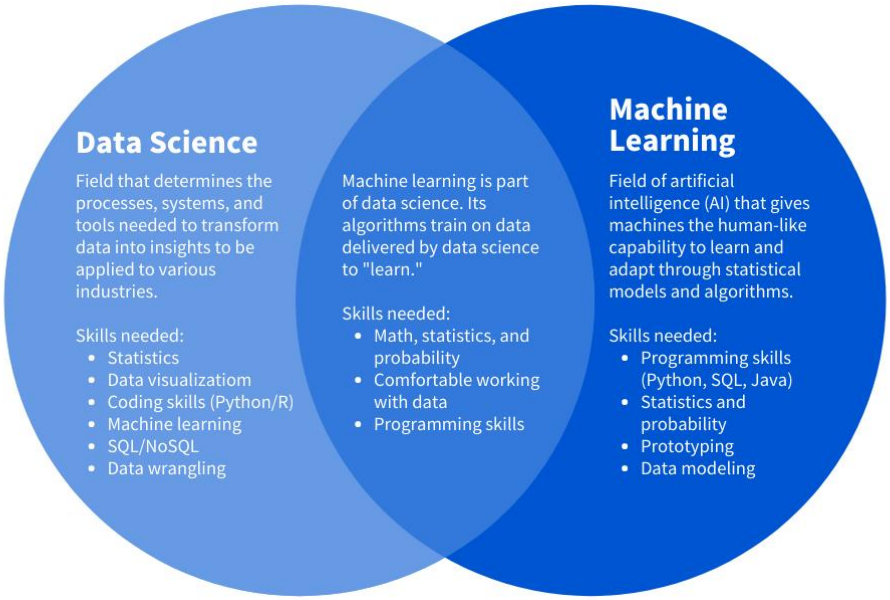
12. Slow implementations and results

This issue is also very commonly seen in machine learning models. However, machine learning models are highly efficient in producing accurate results but are time-consuming. Slow programming, excessive requirements' and overloaded data take more time to provide accurate results than expected. This needs continuous maintenance and monitoring of the model for delivering accurate results.

13. Irrelevant features

Although machine learning models are intended to give the best possible outcome, if we feed garbage data as input, then the result will also be garbage. Hence, we should use relevant features in our training sample. A machine learning model is said to be good if training data has a good set of features or less to no irrelevant features.

Data Science Vs Machine Learning:



S.no	Data Science	Machine Learning
1	Data Science is a field about processes and systems to extract data from structured and semi-structured data.	Machine Learning is a field of study that gives computers the capability to learn without being explicitly programmed.
2	Need the entire analytics universe.	Combination of Machine and Data Science.
3	Branch that deals with data.	Machines utilize data science techniques to learn about the data.
4	Data in Data Science maybe or maybe not evolved from a machine or mechanical process.	It uses various techniques like regression and supervised clustering.
5	Data Science as a broader term not only focuses on algorithms statistics but also takes care of the data processing.	But it is only focused on algorithm statistics.
6	It is a broad term for multiple disciplines.	It fits within data science.

7	Many operations of data science that is, data gathering, data cleaning, data manipulation, etc.	It is three types: Unsupervised learning, Reinforcement learning, Supervised learning.
8	Example: Netflix uses Data Science technology.	Example: Facebook uses Machine Learning technology.