Unit-III
Preparing to Model and Preprocessing

Computer Department
AVPTI Rajkot

Unit Outcomes

- Describe different types of Machine learning Activities
- Explain Data preprocessing

Topics

- Machine Learning activities
- Types of Data
- Data quality and remediation
- Data Pre-Processing

Preparing Model

- Data Collection
→ The quantity & quality of your data dictate how accurate our model is
→ The outcome of this step is generally a representation of data which we will use for training
→ Using pre-collected data, by way of datasets from Kaggle

- Data Preparation
→ Wrangle data and prepare it for training
→ Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)
→ Visualize data to help detect relevant relationships between variables
→ Split into training and evaluation sets

Continue…

- Choose a Model
  → Different algorithms are for different tasks; choose the right one

- Train the Model
  → The goal of training is to answer a question or make a prediction correctly as often as possible
  → Each iteration of process is a training step

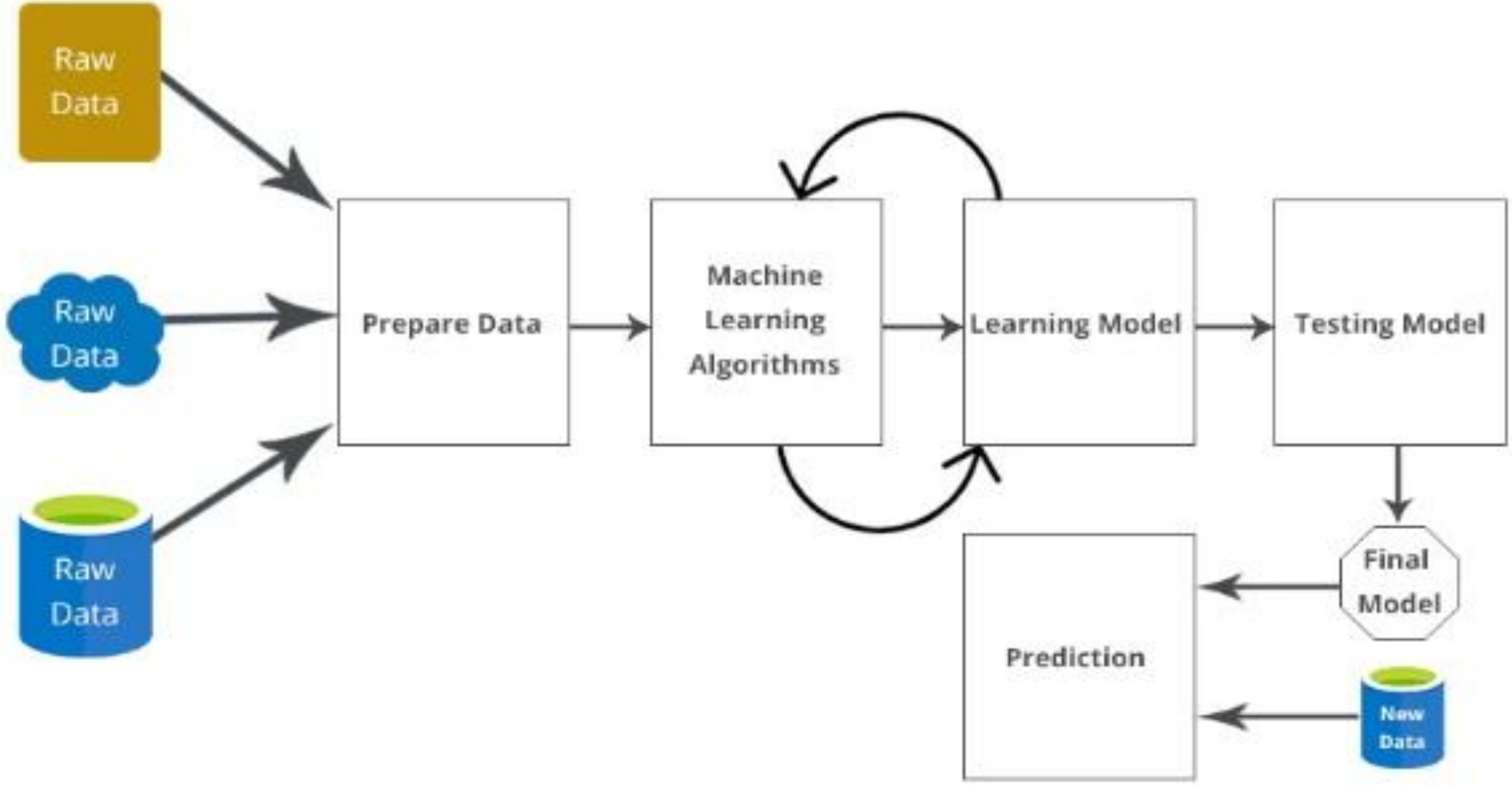
Continue…

- Evaluate the Model
  → Uses some metric or combination of metrics to "measure" objective performance of model
  → Test the model against previously unseen data

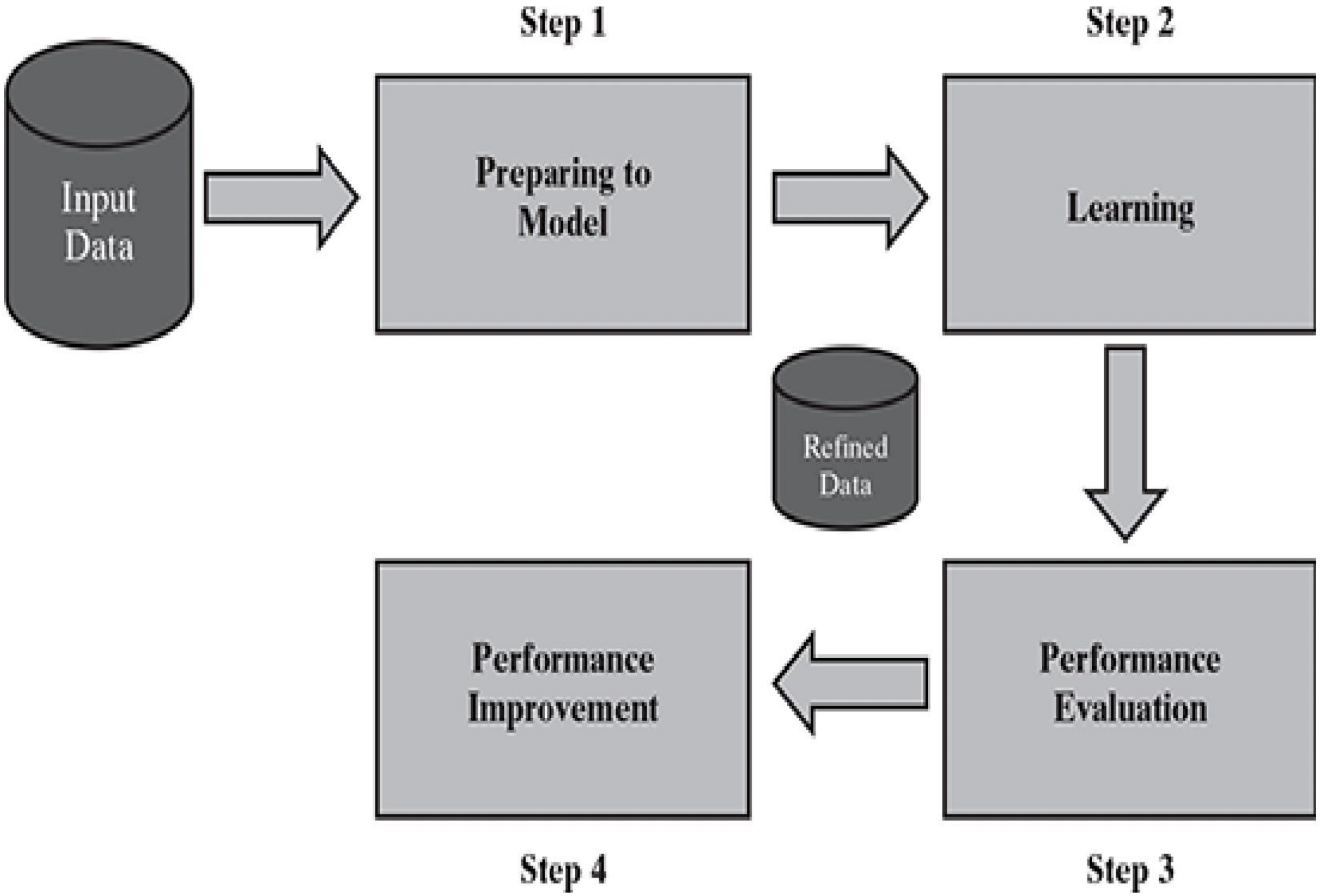
- Parameter Tuning
  → This step refers to *hyperparameter* tuning
  → Tune model parameters for improved performance
  → Simple model hyperparameters may include: number of training steps, learning rate etc.

- Make Predictions
  → Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world

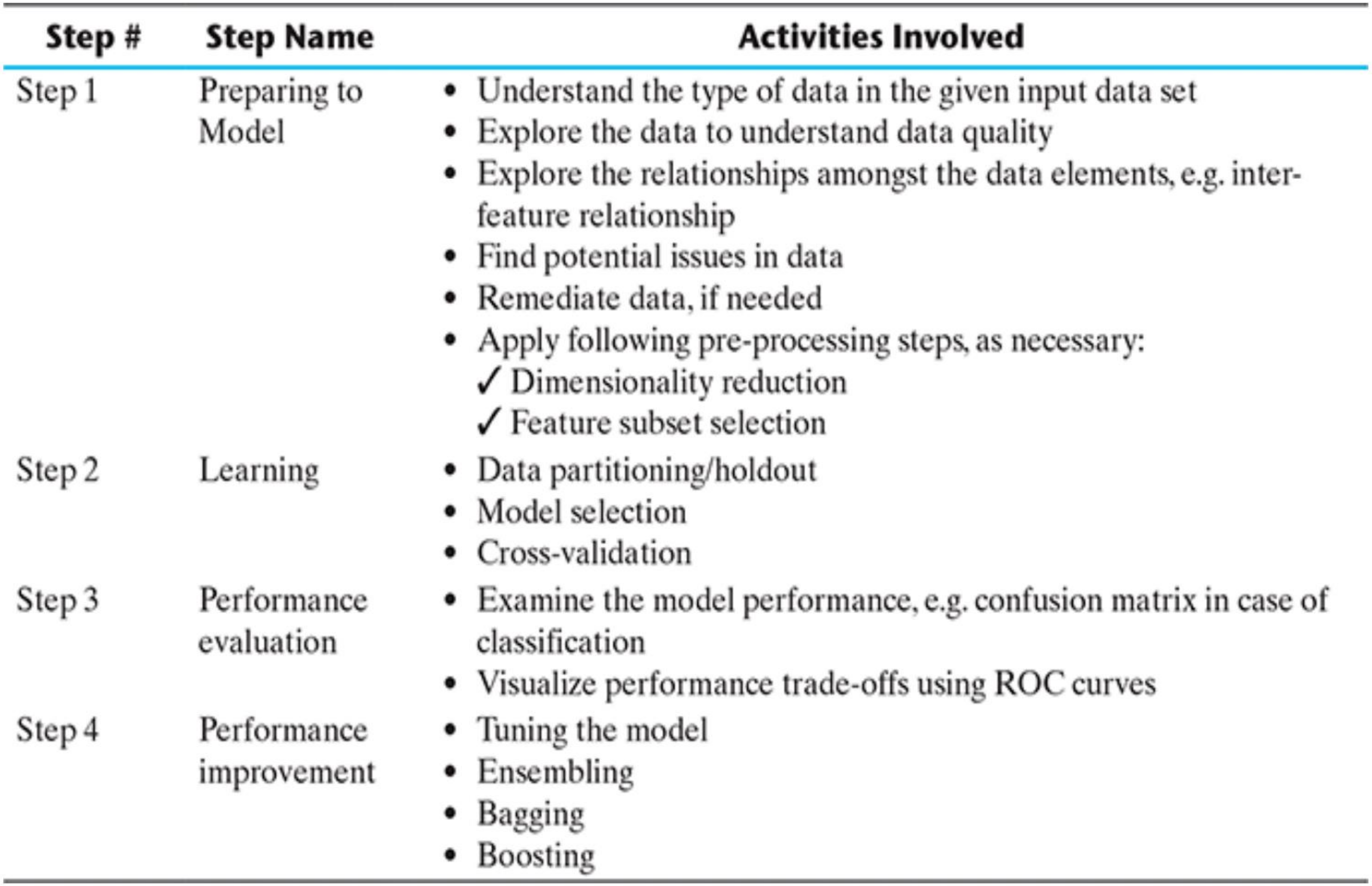
Continue…


Raw Data
Raw Data
Raw Data
Prepare Data
Machine Learning Algorithms
Learning Model
Testing Model
Final Model
Prediction
New Data

Process of Machine Learning

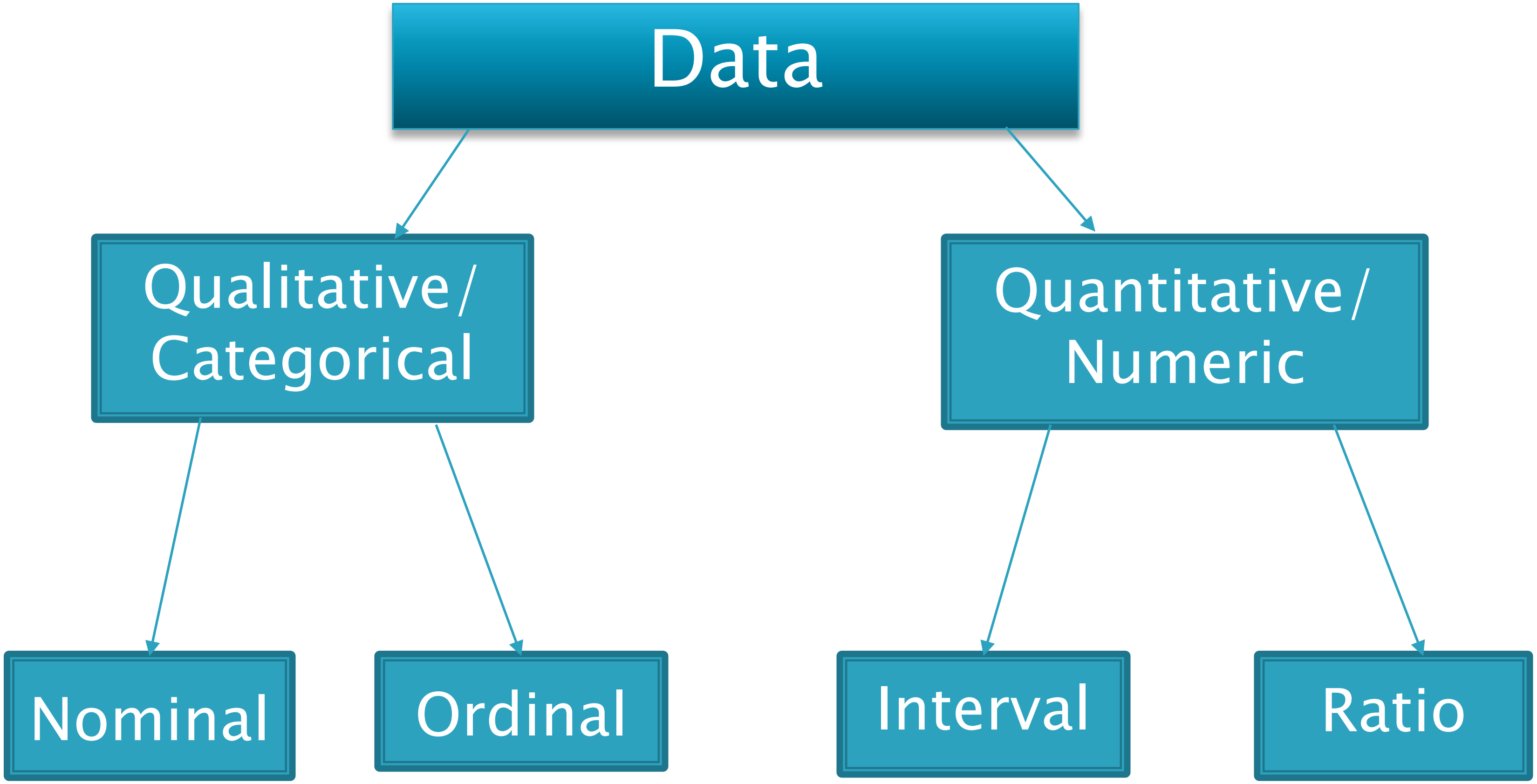


Continue…



Step #	Step Name	Activities Involved
Step 1	Preparing to Model	• Understand the type of data in the given input data set • Explore the data to understand data quality • Explore the relationships amongst the data elements, e.g. inter-feature relationship • Find potential issues in data • Remediate data, if needed • Apply following pre-processing steps, as necessary: ✓ Dimensionality reduction ✓ Feature subset selection
Step 2	Learning	• Data partitioning/holdout • Model selection • Cross-validation
Step 3	Performance evaluation	• Examine the model performance, e.g. confusion matrix in case of classification • Visualize performance trade-offs using ROC curves
Step 4	Performance improvement	• Tuning the model • Ensembling • Bagging • Boosting

Types of Data



Data

Qualitative/ Categorical

Quantitative/ Numeric

Nominal

Ordinal

Interval

Ratio

Continue…

- **Quantitative data is anything that can be counted or measured**; it refers to numerical data.
- Quantitative data can tell you “how many,” “how much,” or “how often”—for example, how many people attended last week’s webinar? How much revenue did the company make in 2019? How often does a certain customer group use online banking?
- My best friend is 5 feet and 7 inches tall
- My best friend lives twenty miles away from me

Continue…

- **Qualitative data is descriptive,** referring to things that can be observed but not measured—such as colors or emotions.
- qualitative data cannot be measured or counted. It’s descriptive, expressed in terms of language rather than numerical values.
- My best friend has curly brown hair
- My best friend drives a red car

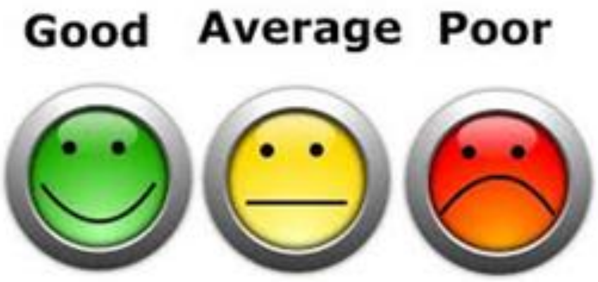
Nominal Data

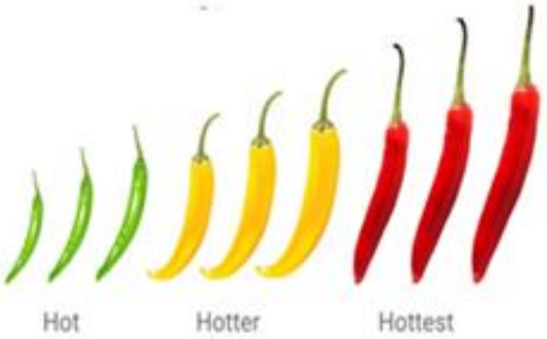
- This Is In Use To Express Names Or Labels Which Are Not Order Or Measurable.
- Nominal data is data that can be labelled or classified into mutually exclusive categories within a variable.
- These categories cannot be ordered in a meaningful way.
- It is a group of objects or ideas that can be collectively grouped on the basis of a particular characteristic—a qualitative property.

Fig: Gender (Female, Male),
An Example Of Nominal Data Type

Ordinal Data

- classifies data while introducing an order, or ranking. However, there is no clearly defined interval between these categories.
- We use ordinal data to observe customer feedback, satisfaction, economic status, education level, etc. Such data only shows the sequences and cannot be used for statistical analysis. We cannot perform arithmetical tasks on ordinal data.





Interval data

- interval data are a **numerical data** type. In other words, it’s a level of measurement that involves data that’s **naturally quantitative** (is usually measured in numbers).
- Interval data is classified and ordered by intervals, which specify that the distance between each value is equal. The distances are, therefore, important.
- Temperature in Fahrenheit or Celsius (–20, –10, 0, +10, +20, etc.)
- Times of the day (1pm, 2pm, 3pm, 4pm, etc.)
- Income level on a continuous scale ($10K, $20K, $30K, $40K, and so on)
- IQ scores (100, 110, 120, 130, 140, etc.)

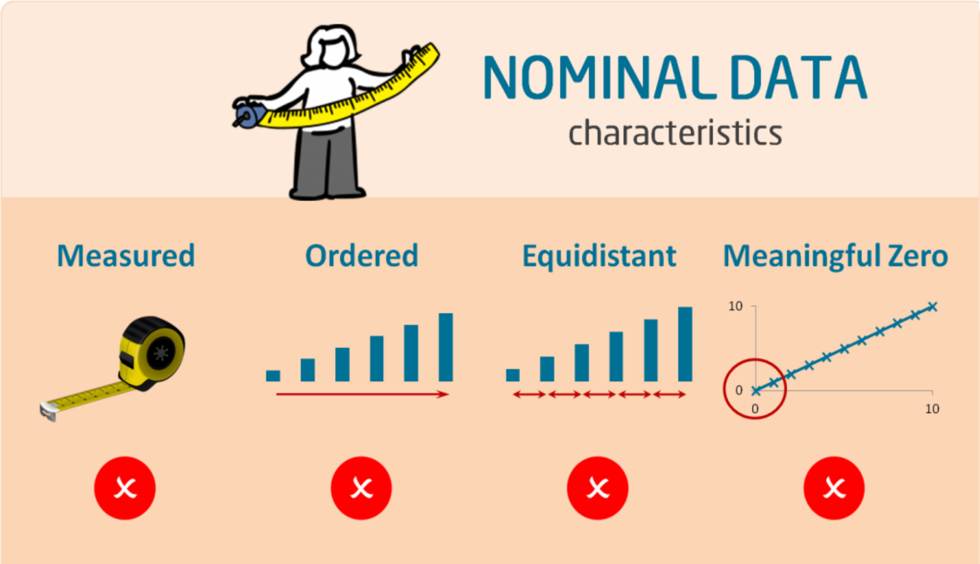
Ratio data

- Ratio data is a form of qualitative data. Like interval data, variables in ratio data are placed at equal distances. Also, this scale features a true zero (which means that the zero has a meaning).
- The ratio data has a true zero, which denotes an absence of a variable.
- Question: How much is your family's monthly income?
- Possible answers: $0-$5000, $5000-$10,000, $10,000-$15,000, $15,000 or more.
- *The distance between the intervals is equal, i.e., $5000.There is also a true zero. Also, the answer can not be negative, i.e., $ -20*

Continue…

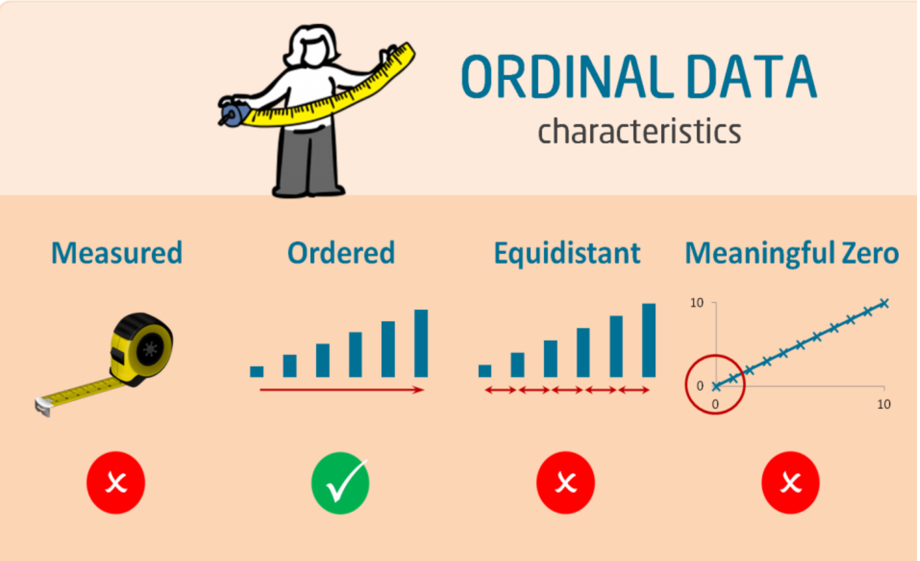
Level	Characteristics of the values
Nominal data	Categories
Ordinal data	Categories, rank/order
Interval data	Categories, rank/order, equal spacing
Ratio data	Categories, rank/order, equal spacing, true zero

Continue…



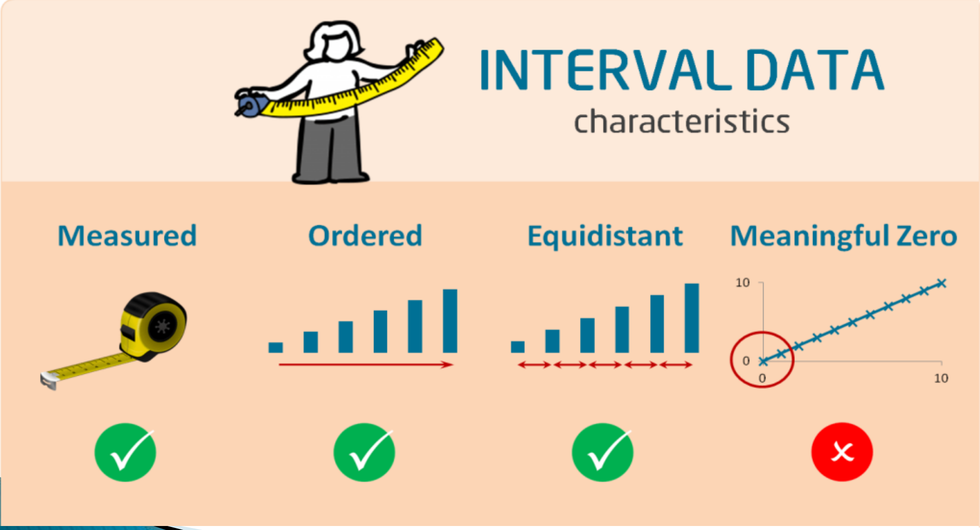
NOMINAL DATA
characteristics

Measured	Ordered	Equidistant	Meaningful Zero
✗	✗	✗	✗



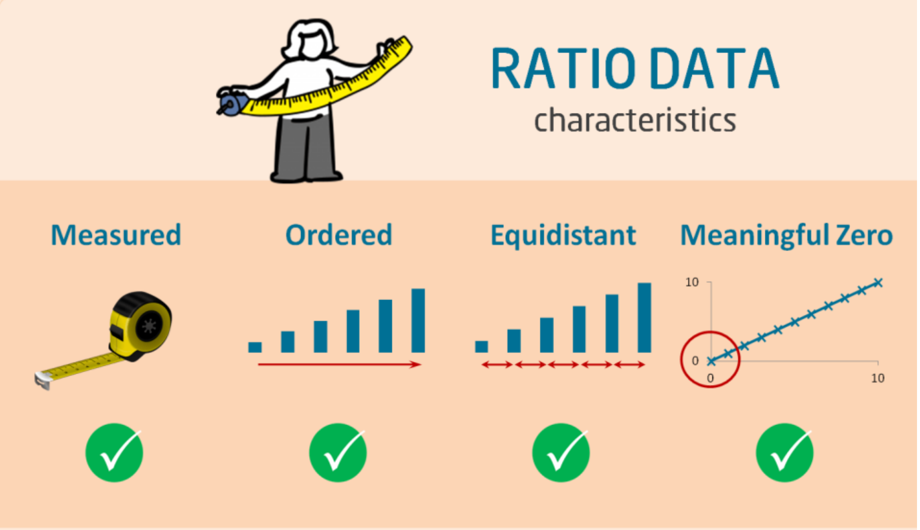
ORDINAL DATA
characteristics

Measured	Ordered	Equidistant	Meaningful Zero
✗	✓	✗	✗



INTERVAL DATA
characteristics

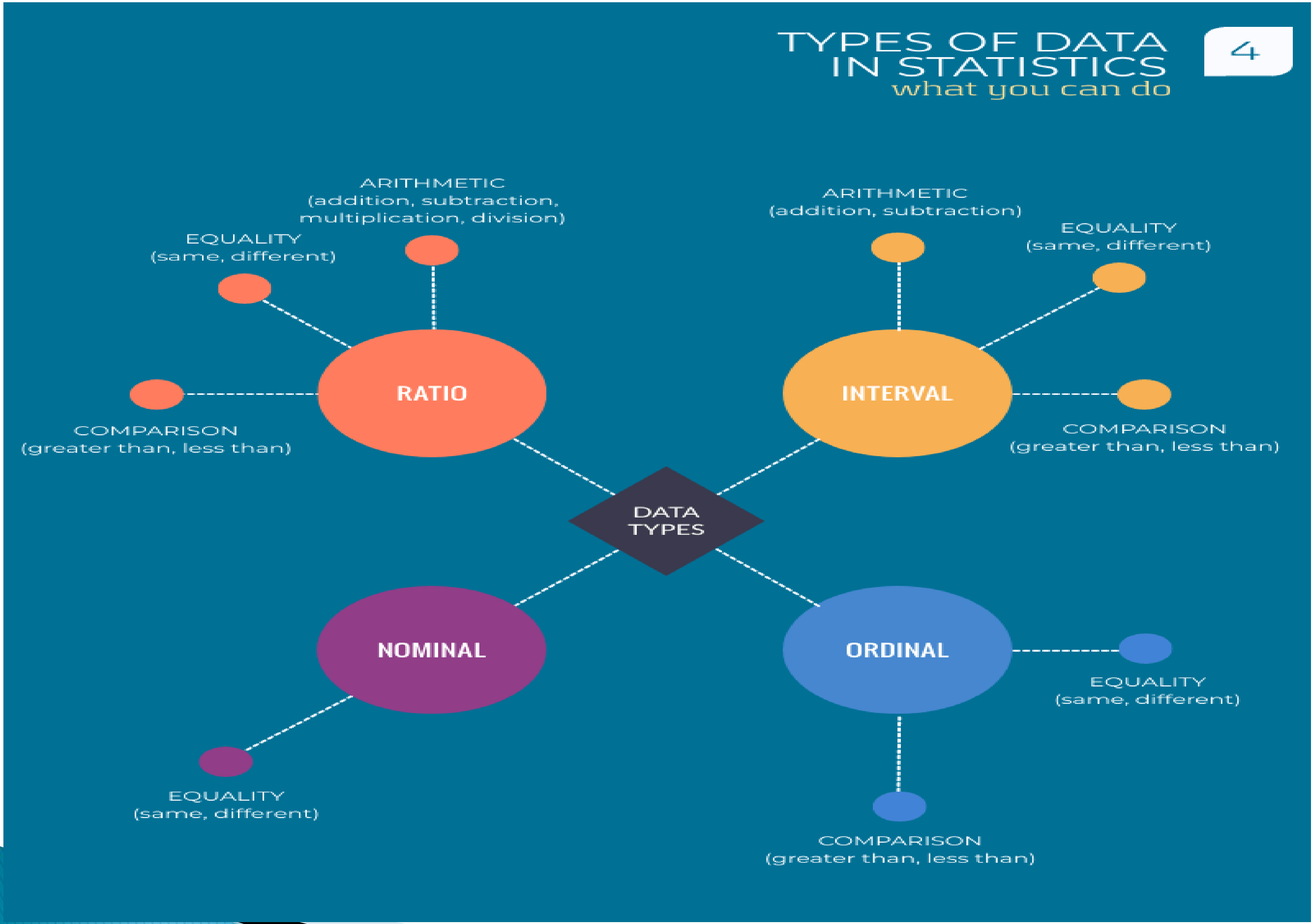
Measured	Ordered	Equidistant	Meaningful Zero
✓	✓	✓	✗



RATIO DATA
characteristics

Measured	Ordered	Equidistant	Meaningful Zero
✓	✓	✓	✓

Continue…


TYPES OF DATA
IN STATISTICS
what you can do
4
ARITHMETIC
(addition, subtraction, multiplication, division)
EQUALITY
(same, different)
COMPARISON
(greater than, less than)
RATIO
ARITHMETIC
(addition, subtraction)
EQUALITY
(same, different)
INTERVAL
COMPARISON
(greater than, less than)
DATA TYPES
NOMINAL
EQUALITY
(same, different)
ORDINAL
EQUALITY
(same, different)
COMPARISON
(greater than, less than)

Data quality

- Data quality is the measure of how well suited a data set is to serve its specific purpose.
- Data quality tells us how reliable a particular set of data is and whether or not it will be good enough for a user to employ in decision-making.
- Data quality measures the condition of data, relying on factors such as how useful it is to the specific purpose, completeness, accuracy, timeliness (e.g., is it up to date?), consistency, validity, and uniqueness.

Data Quality Issues in ML

Issues :

- Imbalanced Data
- Class Overlap
- Lack of Data
- Inconsistent Data
- Irrelevant Data
- Redundant Data
- Noisy Data
- Missing Data
- Ambiguous data
- Inaccurate data
- Too much data

Reasons :

Data entry errors,Human errors, data collection error, Sample set selection error, data transformation errors, data storage, and accessibility issues, inconsistent data formats.

Data remediation

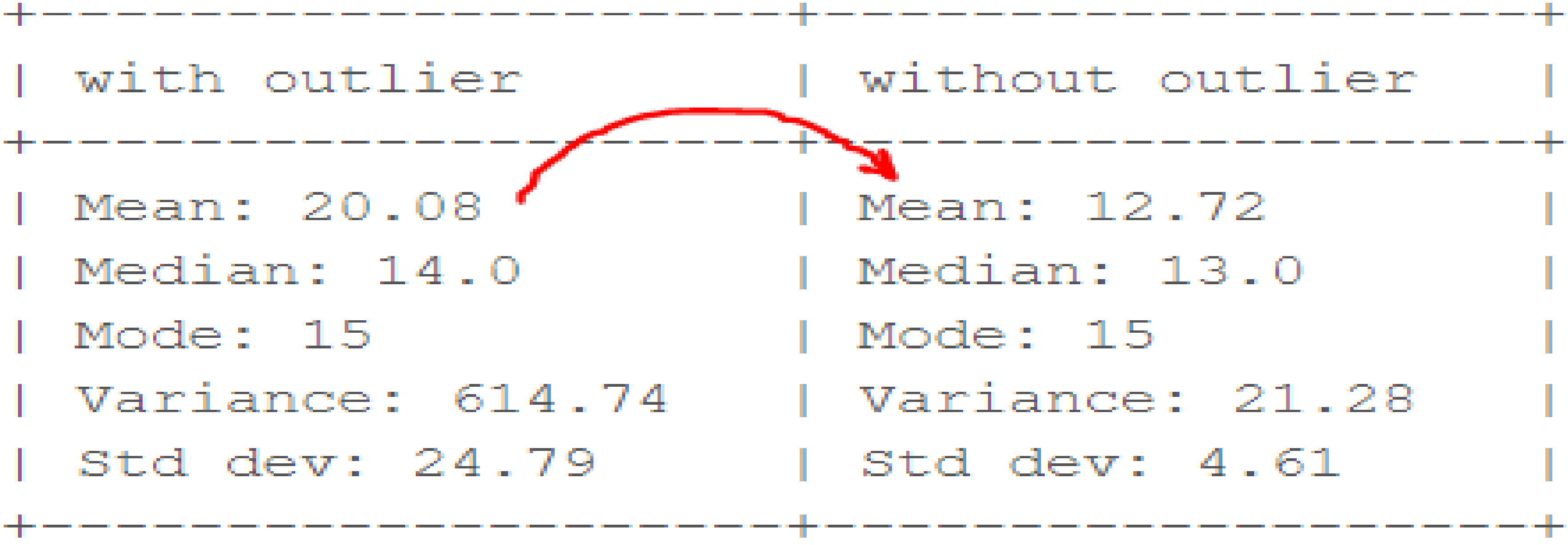
- “remediation” derives from the word “remedy,” which is to correct a mistake.
- Data remediation process typically involves replacing, modifying, cleansing or deleting any “dirty” data.
- Data remediation is the process of cleansing, organizing and migrating data so that it’s properly protected and best serves its intended purpose.
- Data remediation refers to the process of identifying and correcting errors, inconsistencies, and inaccuracies in data. This can include tasks such as removing duplicate records, standardizing format and data types, and filling in missing values.

Handling outliers

- In statistics, we call the data points significantly different from the rest of the dataset outliers.
- In other words, an outlier contains a value that is inconsistent or doesn’t comply with the general behavior.
- Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.
- An outlier may occur due to the variability in the data, or due to experimental error, human error, a measurement error , an input error.

Continue…

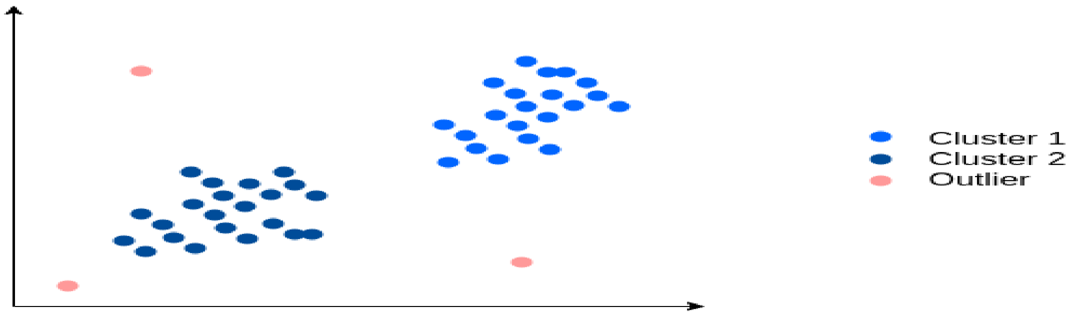
- Consider a small dataset, sample= [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]. By looking at it, one can quickly say ‘101’ is an outlier that is much larger than the other values.



```
+---------------------+---------------------+
| with outlier        | without outlier     |
+---------------------+---------------------+
| Mean: 20.08         | Mean: 12.72         |
| Median: 14.0        | Median: 13.0        |
| Mode: 15            | Mode: 15            |
| Variance: 614.74    | Variance: 21.28     |
| Std dev: 24.79      | Std dev: 4.61       |
+---------------------+---------------------+
```

Continue…

- Detecting and handling outlier values in the dataset is a critical issue in machine learning.
- As the supervised learning algorithms learn the patterns in the dataset, training with noisy datasets results in models with low prediction power.
- Some algorithms, such as KNN, are more sensitive to outliers.



Method to handle Outliers

- There are several ways to handle an outlier value. Depending on the cause and density, we can select an appropriate method to address the outlier values.
- **Remove outliers:**
  In some cases, it may be appropriate to simply remove the observations that contain outliers.
- This can be particularly useful if you have a large number of observations and the outliers are not true representatives of the underlying population.

Continue…

- **Impute outliers**:
  In this case, outliers are simply considered as missing values.
- You can employ various imputation techniques for missing values, such as mean, median, mode, nearest neighbor, etc., to impute the values for outliers.
- **Capping**
- setting a limit for the feature and set the value of all the outliers exceeding the limit to the value of the limit.

Handling missing values

- Missing data is defined as the values or data that is not stored (or not present) for some variable/s in the given dataset.
- In the dataset, the blank shows the missing values.
- In Pandas, usually, missing values are represented by **NaN**. It stands for **Not a Number**.
- The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values.
- Incomplete data can bias the results of the machine learning models and/or reduce the accuracy of the model.

Why the data could be missing.

- Past data might get corrupted due to improper maintenance.
- There might be a failure in recording the values due to human error.
- The user has not provided the values intentionally.
- incomplete data entry
- equipment malfunctions
- lost files etc.

Types of missing value

Figure 1 - Different Types of Missing Values in Datasets

Missing Completely At Random (MCAR)

Missing At Random (MAR)

Missing Not At Random (MNAR)

Method to handle missing data

- **Deleting the Missing value**
  - Deleting the entire row
  - Deleting the entire column
- **Imputing the Missing Value**
  - Replacing with an arbitrary value
  - Replacing with the mean,mode,median,previous value, next value
- **Estimate missing values**

Based on types of missing data you can use the above method to handle the those missing data.

Data Pre-Processing

- Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model.
- It is the first and crucial step while creating a machine learning model.
- When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

Dimensionality reduction

- Dimensionality reduction is the process of reducing the number of features (or dimensions) in a dataset while retaining as much information as possible.
- This can be done for a variety of reasons, such as to reduce the complexity of a model, to improve the performance of a learning algorithm, or to make it easier to visualize the data.
- It is a process of transforming high-dimensional data into a lower-dimensional space that still preserves the essence of the original data.

Continue…

- In machine learning, high-dimensional data refers to data with a large number of features or variables.
- It is a common problem in machine learning, where the performance of the model deteriorates as the number of features increases.
- This is because the complexity of the model increases with the number of features, and it becomes more difficult to find a good solution.
- In addition, high-dimensional data can also lead to overfitting, where the model fits the training data too closely and does not generalize well to new data.

Continue…

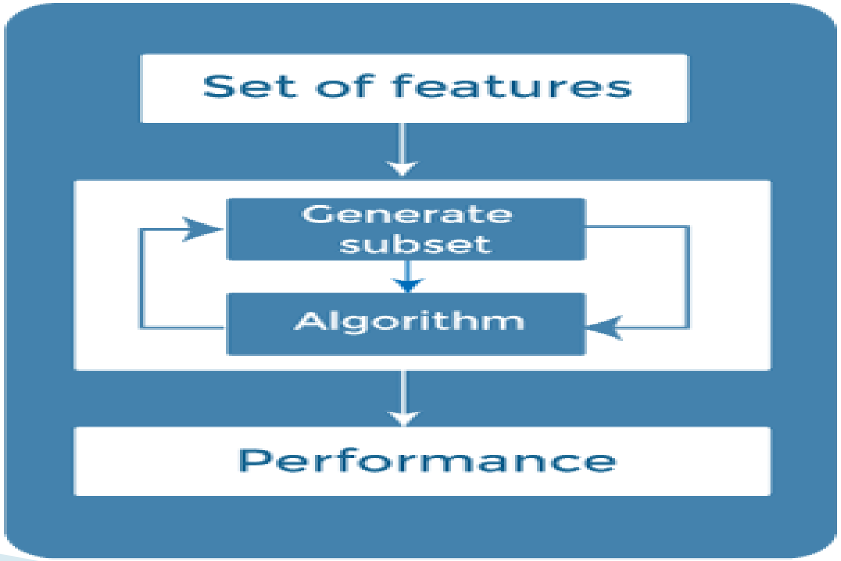
- Dimensionality reduction can help to mitigate these problems by reducing the complexity of the model and improving its generalization performance.
- There are two main approaches to dimensionality reduction: feature selection /feature subset selection and feature extraction.

Feature Subset Selection

- Feature selection is a way of selecting the subset of the most relevant features from the original features set by removing the redundant, irrelevant, or noisy features.
- Feature selection involves selecting a subset of the original features that are most relevant to the problem at hand.
- The goal is to reduce the dimensionality of the dataset while retaining the most important features.
- It usually involves three ways:
- Wrapper
- Filter
- Embedded

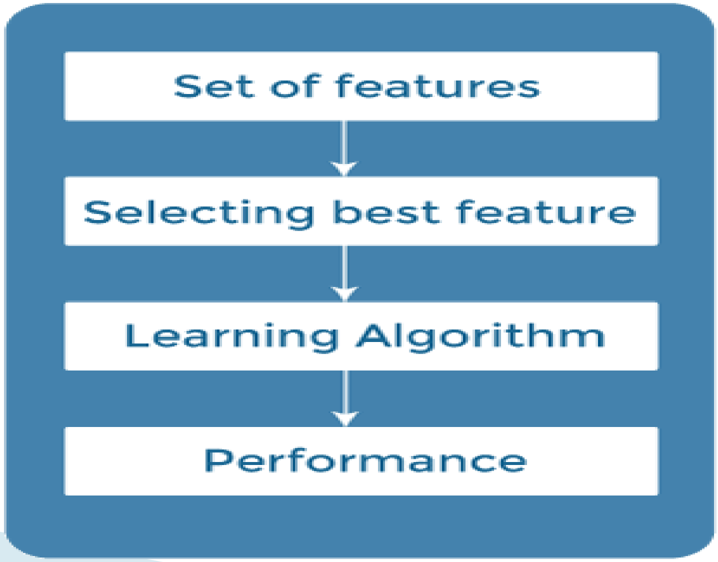
Wrapper Methods

- In wrapper methodology, selection of features is done by considering it as a search problem, in which different combinations are made, evaluated, and compared with other combinations. It trains the algorithm by using the subset of features iteratively.



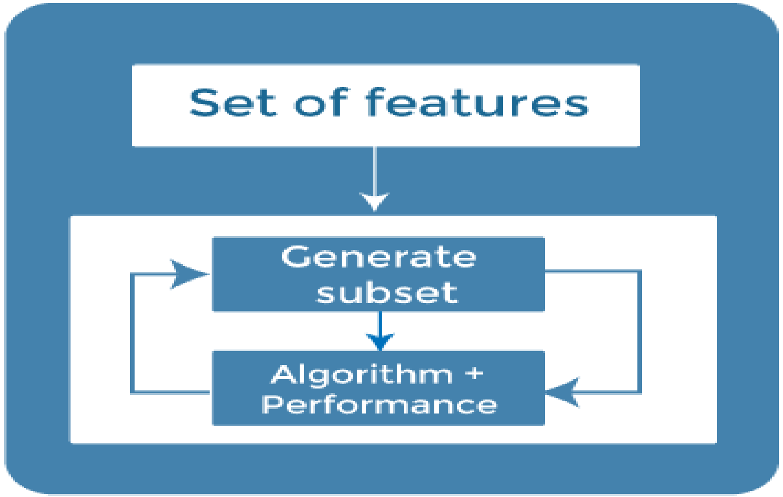
Filter Methods

- In Filter Method, features are selected on the basis of statistics measures. This method does not depend on the learning algorithm and chooses the features as a pre-processing step.
- The filter method filters out the irrelevant feature and redundant columns from the model by using different metrics through ranking.
- The advantage of using filter methods is that it needs low computational time and does not overfit the data.



Embedded Methods

- These methods are also iterative, which evaluates each iteration, and optimally finds the most important features that contribute the most to training in a particular iteration.
- Embedded methods combined the advantages of both filter and wrapper methods by considering the interaction of features along with low computational cost.
- These are fast processing methods similar to the filter method but more accurate than the filter method.
- Combination of above methods is known as a hybrid method.



Feature extraction.

- Feature extraction involves creating new features by combining or transforming the original features.
- The goal is to create a set of features that captures the essence of the original data in a lower-dimensional space.
- Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions.
- This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

Continue…

- There are several methods for feature extraction, including principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (t-SNE).
- PCA is a popular technique that projects the original features onto a lower-dimensional space while preserving as much of the variance as possible.

Data Partition-*k*-fold cross validation

- K-fold cross-validation is a data partitioning technique which splits an entire dataset into k groups.
- Then, we train and test k different models using different combinations of the groups of data we just partitioned, and use the results from these k models to check the model’s overall performance.
- We will use k-folds to describe a number of groups we decide to partition the data, so in an example of 20 rows, we can split them into 2 folds with 10 rows each, 4 folds with 5 rows each, or 10 folds with 2 rows each.

Continue…

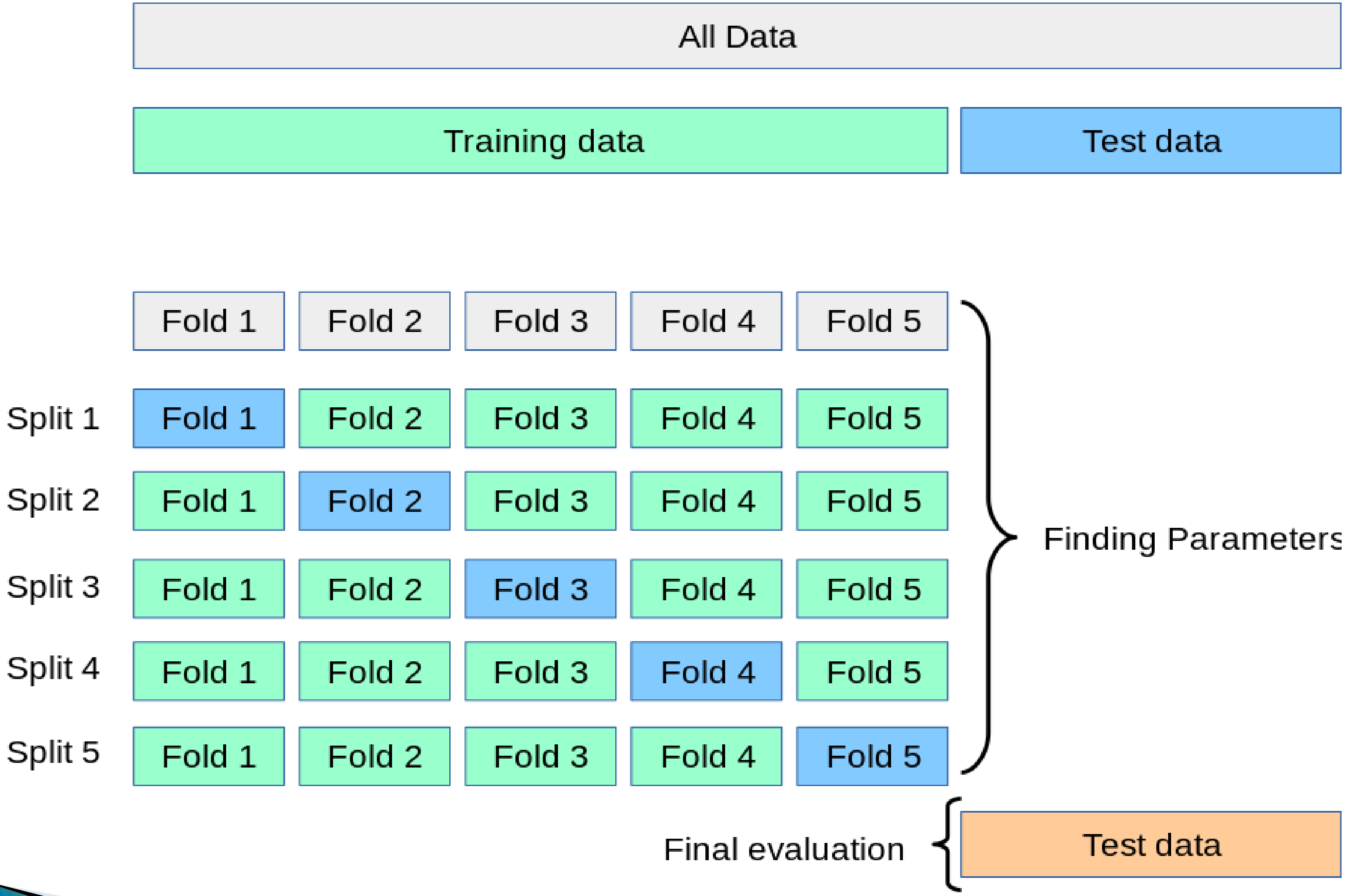
- It is a ***data partitioning strategy*** so that you can effectively use your dataset to build ***a more generalized model***.
- The main intention of doing any kind of machine learning is to develop a more generalized model which can perform well on ***unseen data***.
- One can build a perfect model on the training data with 100% accuracy or 0 error, but it may fail to generalize for unseen data. So, it is not a good model.

How it works

- The entire dataset is randomly split into equally-sized, independent
- k-folds, without reusing any of the rows in another fold.
- We use k-1folds for model training, and once that model is complete, we test it using the remaining 1 fold to obtain a score of the model's performance.
- We repeat this process k times, so we have k number of models and scores for each.
- Lastly, we take the mean of the k number

  of scores to evaluate the model's performance.

Continue...



All Data

Training data | Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

Final evaluation { Test data

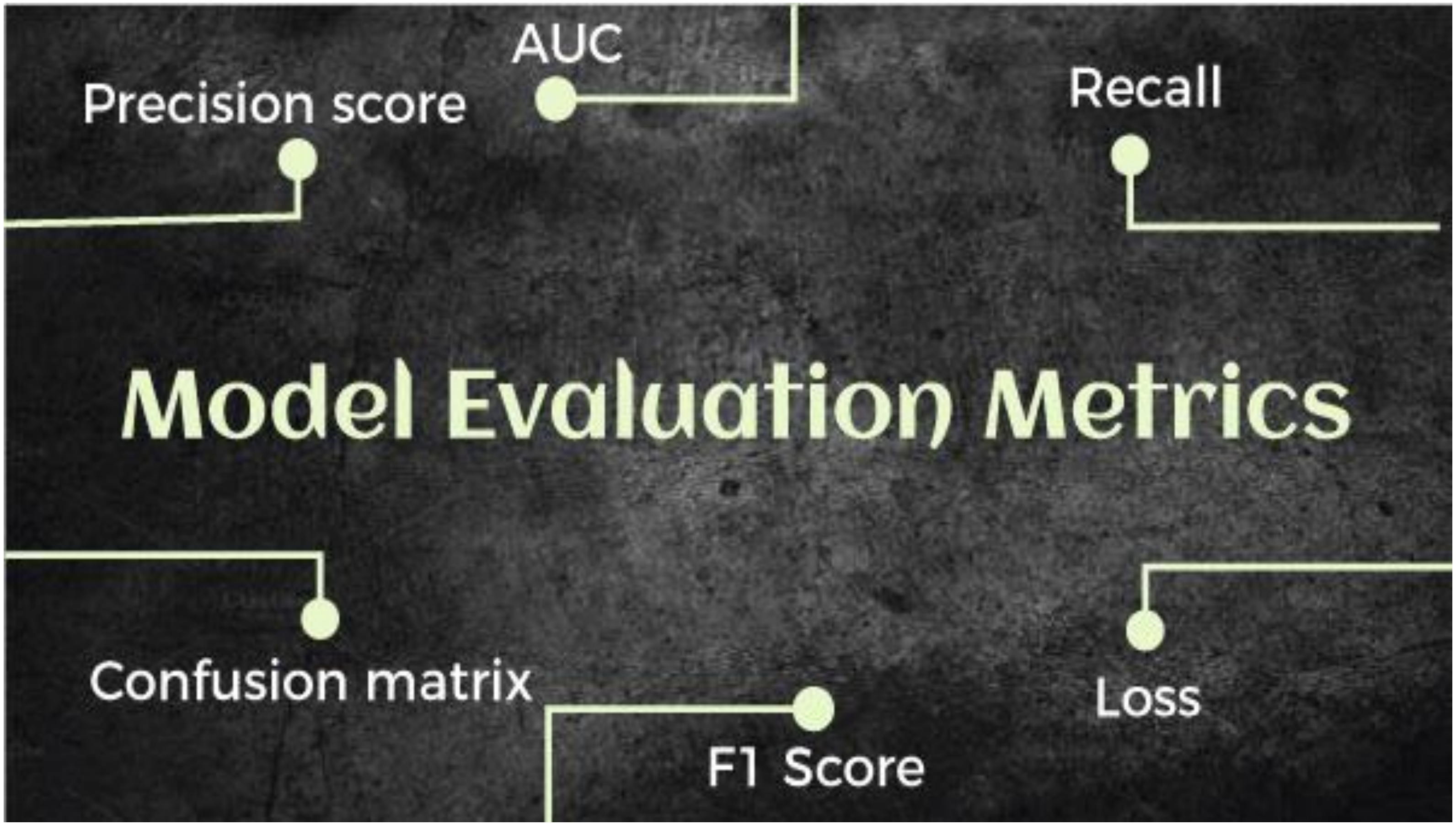
Advantages

- The main purpose of cross validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data.
- By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model’s generalization performance, i.e., its ability to perform well on new, unseen data.

Performance Evaluation

- Evaluating the performance of a Machine learning model is one of the important steps while building an effective ML model.
- *To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics.*
- These performance metrics help us understand how well our model has performed for the given data.
- In this way, we can improve the model's performance by tuning the hyper-parameters.
- Each ML model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset.

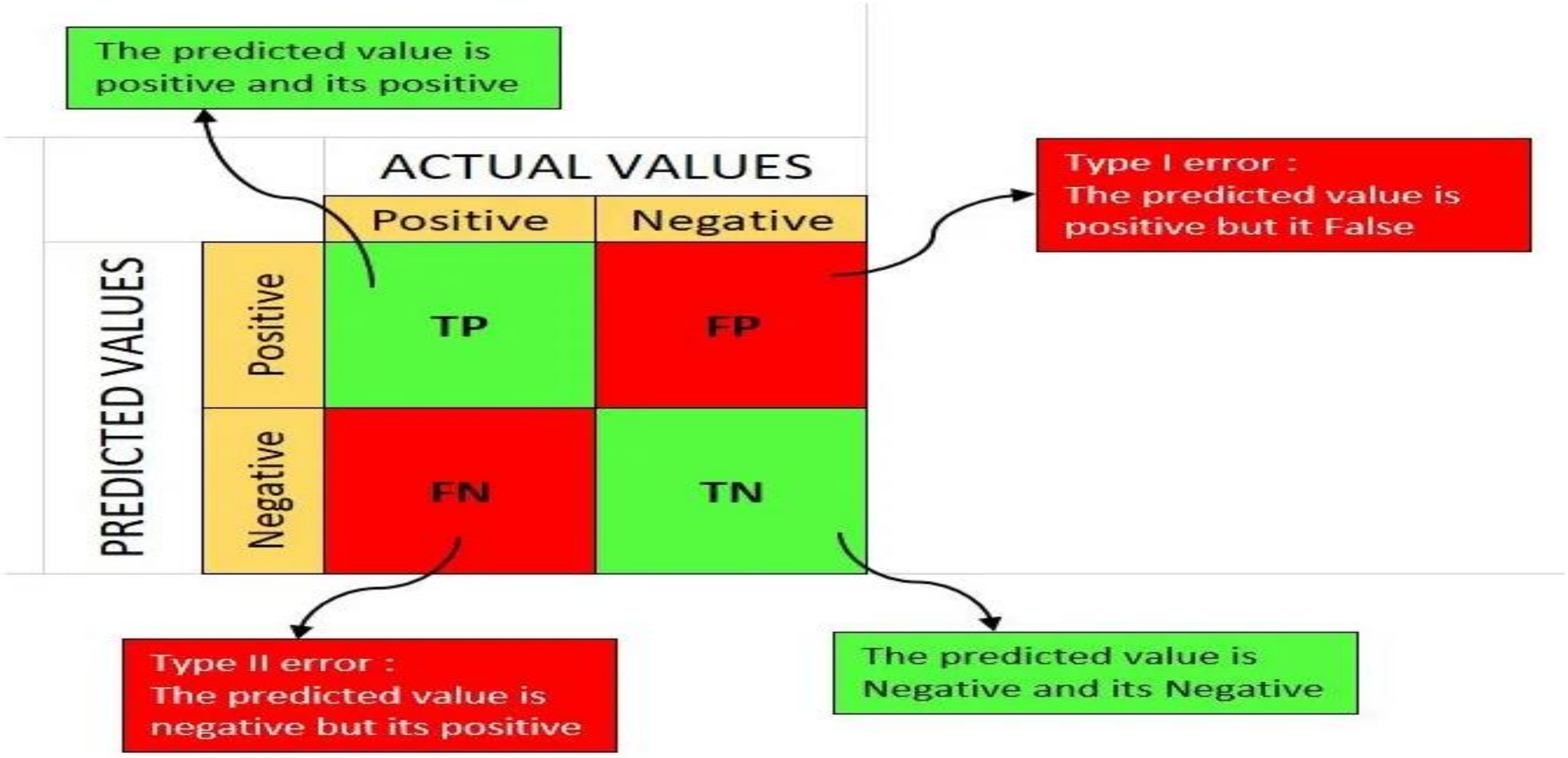
Continue…


AUC
Precision score
Recall
Model Evaluation Metrics
Confusion matrix
Loss
F1 Score

Confusion matrix

- A **Confusion matrix** is an ***N x N matrix*** used for evaluating the **performance of a classification model**, where **N** is the number of ***target classes***.
- The matrix compares the actual target values with those predicted by the machine learning model.
- It summarizes the performance of a machine learning model on a set of test data.
- It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance.
- The matrix displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the model on the test data.
- For binary classification, the matrix will be of a 2X2 table, For multi-class classification, the matrix shape will be equal to the number of classes i.e for n classes it will be nXn.

Continue…



The predicted value is positive and its positive

Type I error :
The predicted value is positive but it False

		ACTUAL VALUES	
		Positive	Negative
PREDICTED VALUES	Positive	TP	FP
	Negative	FN	TN

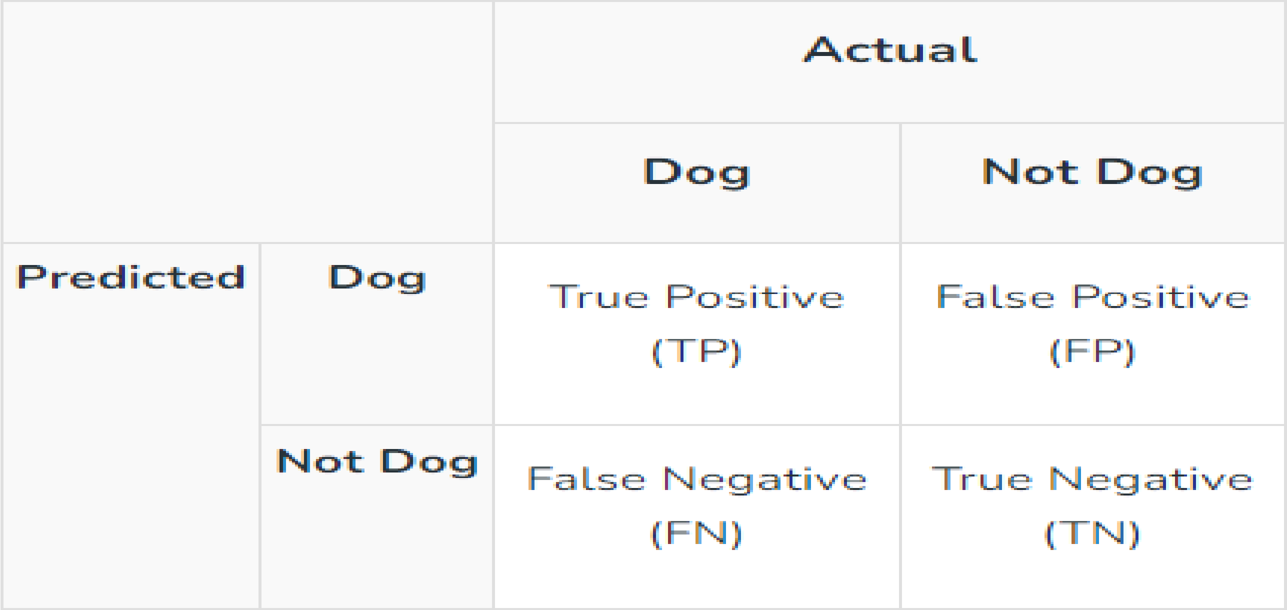
Type II error :
The predicted value is negative but its positive

The predicted value is Negative and its Negative

- The target variable has two values: **Positive** or **Negative**
- The **columns** represent the **actual values** of the target variable
- The **rows** represent the **predicted values** of the target variable

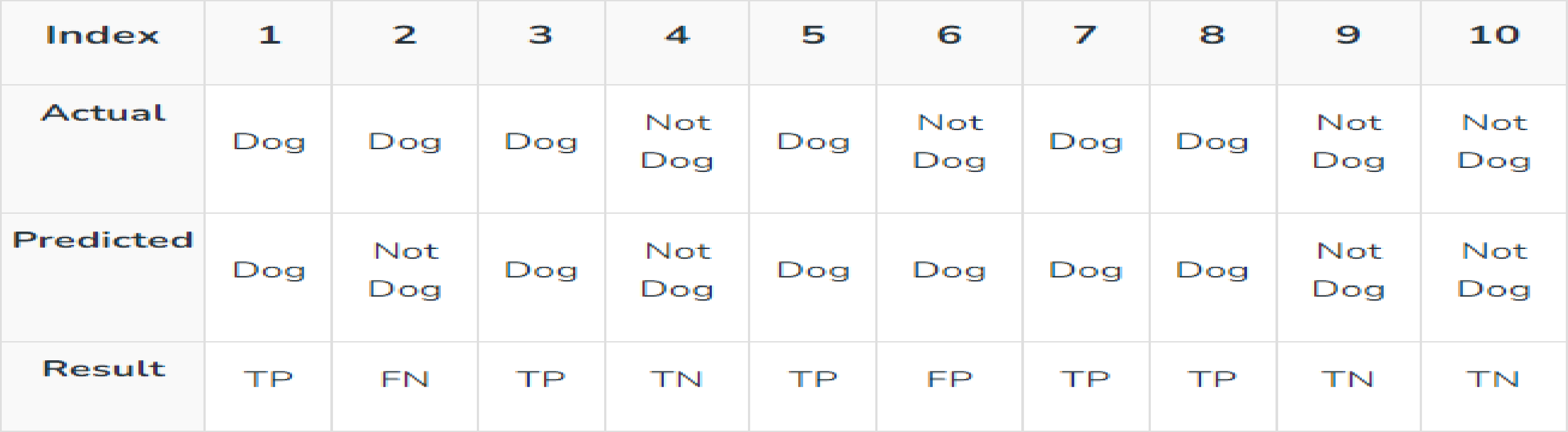
Continue…

- A 2X2 Confusion matrix is shown below for the image reorganization having a Dog image or Not Dog image.



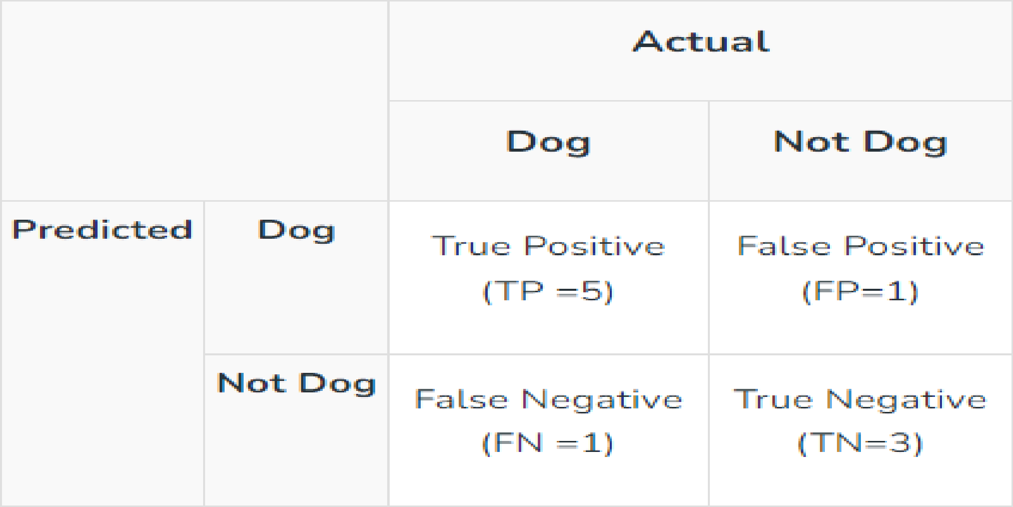
		Actual	
		Dog	Not Dog
Predicted	Dog	True Positive (TP)	False Positive (FP)
	Not Dog	False Negative (FN)	True Negative (TN)

Continue…



Index	1	2	3	4	5	6	7	8	9	10
Actual	Dog	Dog	Dog	Not Dog	Dog	Not Dog	Dog	Dog	Not Dog	Not Dog
Predicted	Dog	Not Dog	Dog	Not Dog	Dog	Dog	Dog	Dog	Not Dog	Not Dog
Result	TP	FN	TP	TN	TP	FP	TP	TP	TN	TN

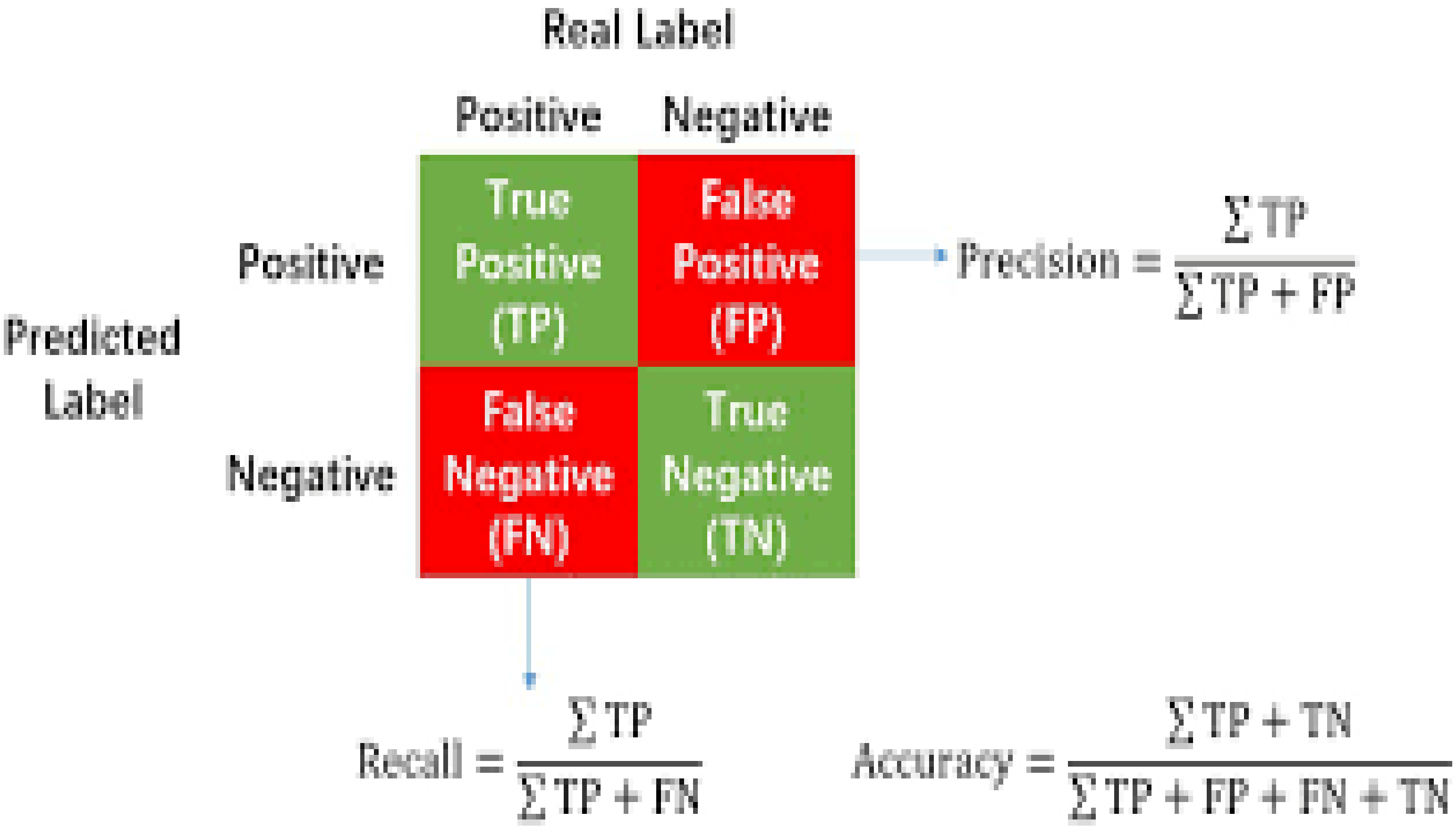
- •Actual Dog Counts = 6
- •Actual Not Dog Counts = 4
- •True Positive Counts = 5
- •False Positive Counts = 1
- •True Negative Counts = 3
- •False Negative Counts = 1



		Actual	
		Dog	Not Dog
Predicted	Dog	True Positive (TP =5)	False Positive (FP=1)
	Not Dog	False Negative (FN =1)	True Negative (TN=3)

Confusion Matrix

Continue…



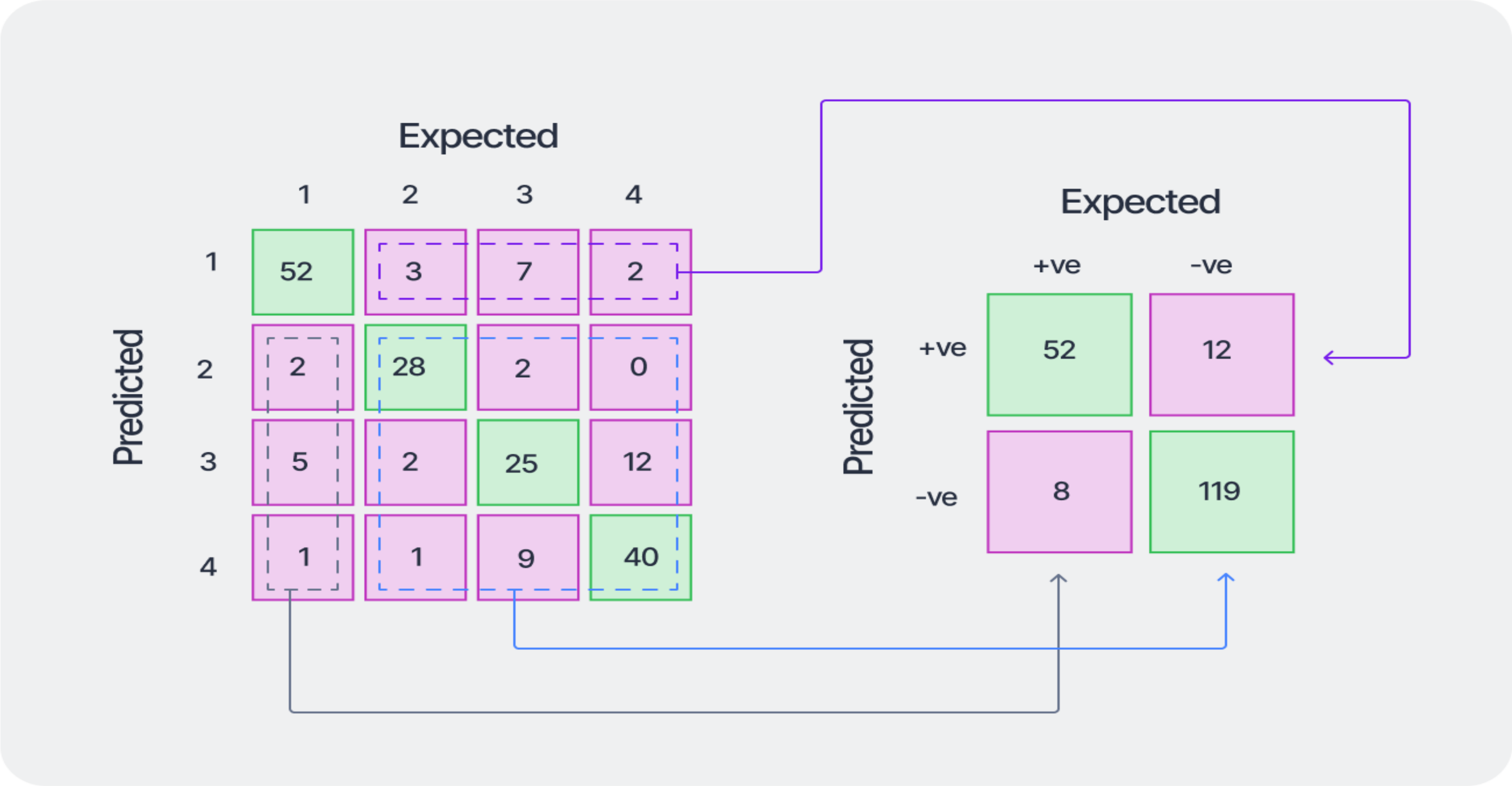
		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{\sum TP}{\sum TP + FP}$$

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN}$$

$$\text{Accuracy} = \frac{\sum TP + TN}{\sum TP + FP + FN + TN}$$

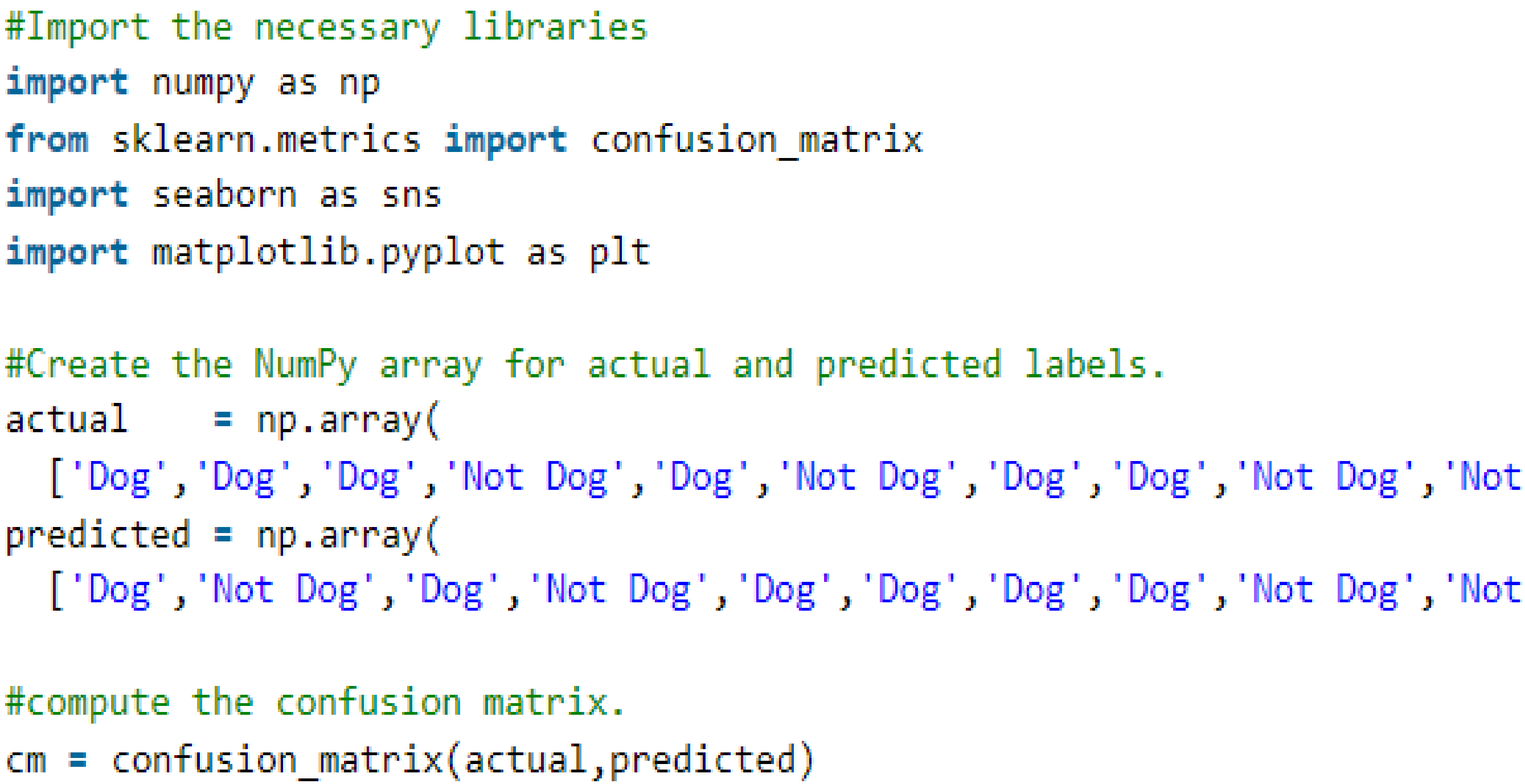
Continue...



Predicted \ Expected	1	2	3	4
1	52	3	7	2
2	2	28	2	0
3	5	2	25	12
4	1	1	9	40

Predicted \ Expected	+ve	-ve
+ve	52	12
-ve	8	119

Continue…



```
#Import the necessary libraries
import numpy as np
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

#Create the NumPy array for actual and predicted labels.
actual    = np.array(
  ['Dog','Dog','Dog','Not Dog','Dog','Not Dog','Dog','Dog','Not Dog','Not
predicted = np.array(
  ['Dog','Not Dog','Dog','Not Dog','Dog','Dog','Dog','Dog','Not Dog','Not

#compute the confusion matrix.
cm = confusion_matrix(actual,predicted)
```

Performance Improvement: Ensemble

- Ensemble learning helps improve machine learning results by combining several models. Ensembles can give you a boost in accuracy on your dataset.
- Basically, ensemble models consist of several individually trained supervised learning models and their results are merged in various ways to achieve better predictive performance compared to a single model.

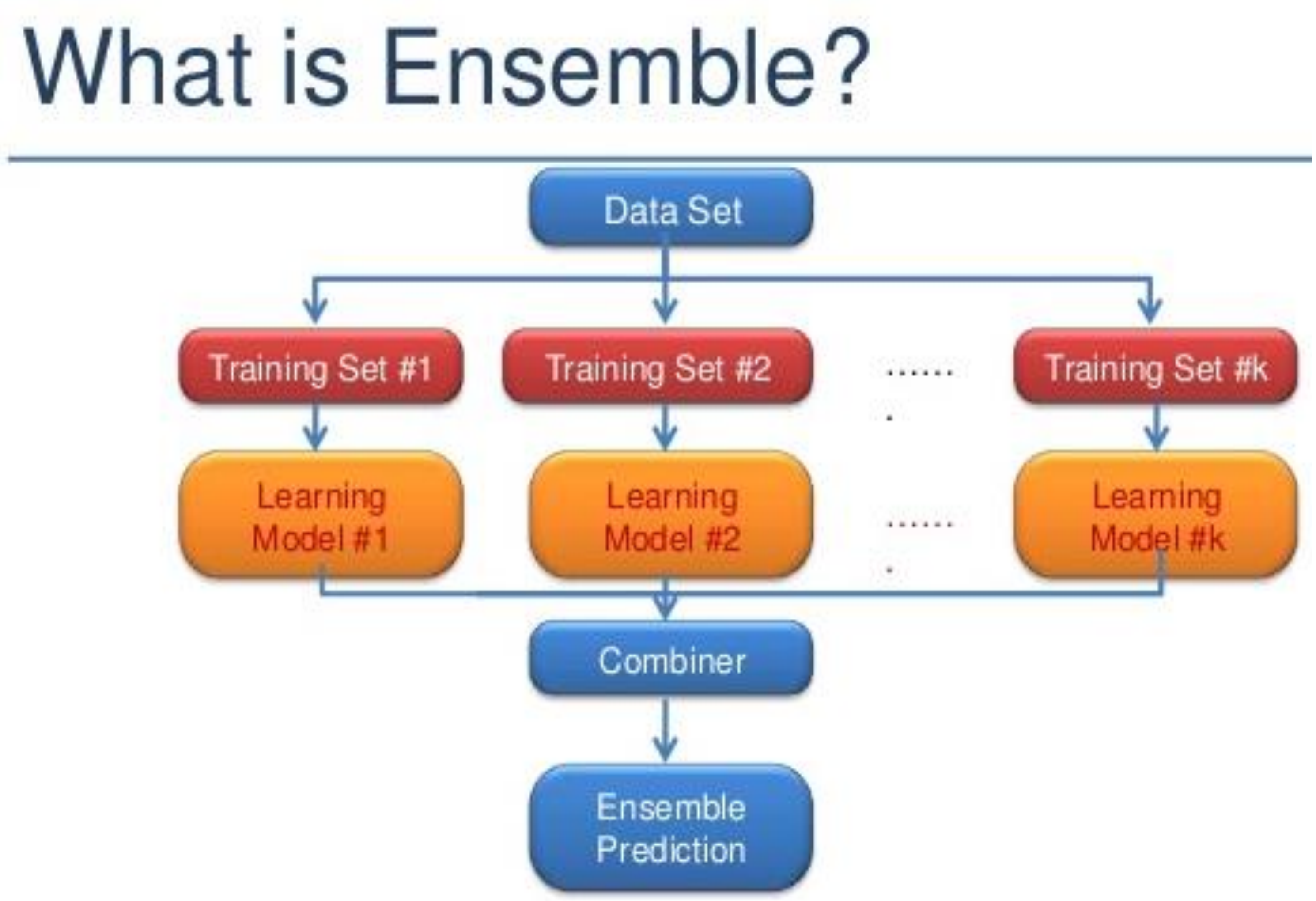
Continue…

- When you want to purchase a new car, will you walk up to the first car shop and purchase one based on the advice of the dealer? It’s highly unlikely.
- You would likely browser a few web portals where people have posted their reviews and compare different car models, checking for their features and prices. You will also probably ask your friends and colleagues for their opinion. In short, you wouldn’t directly reach a conclusion, but will instead make a decision considering the opinions of other people as well.
- Ensemble models in machine learning operate on a similar idea. They combine the decisions from multiple models to improve the overall performance.

Continue…

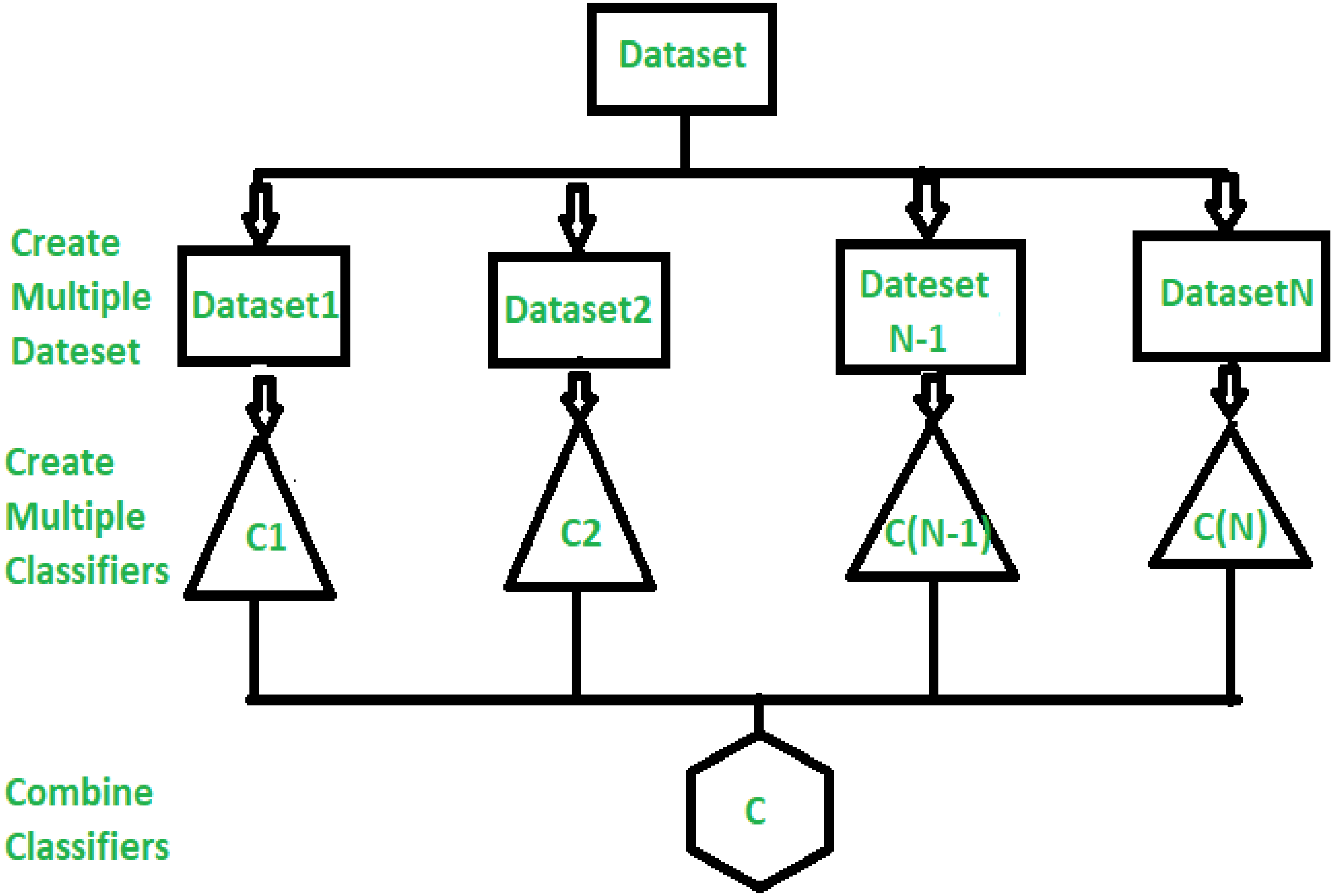
- Ensemble learning is a machine learning technique that enhances accuracy and resilience in forecasting by merging predictions from multiple models.
- It aims to mitigate errors or biases that may exist in individual models by leveraging the collective intelligence of the ensemble.
- The underlying concept behind ensemble learning is to combine the outputs of diverse models to create a more precise prediction.
- By considering multiple perspectives and utilizing the strengths of different models, ensemble learning improves the overall performance of the learning system.
- Diverse group of people are likely to make better decisions as compared to individuals. Similar is true for a diverse set of models in comparison to single models. This diversification in Machine Learning is achieved by a technique called Ensemble Learning.

Continue…



What is Ensemble?

Continue…



Dataset

Create
Multiple
Dateset

Dataset1

Dataset2

Dateset
N-1

DatasetN

Create
Multiple
Classifiers

C1

C2

C(N-1)

C(N)

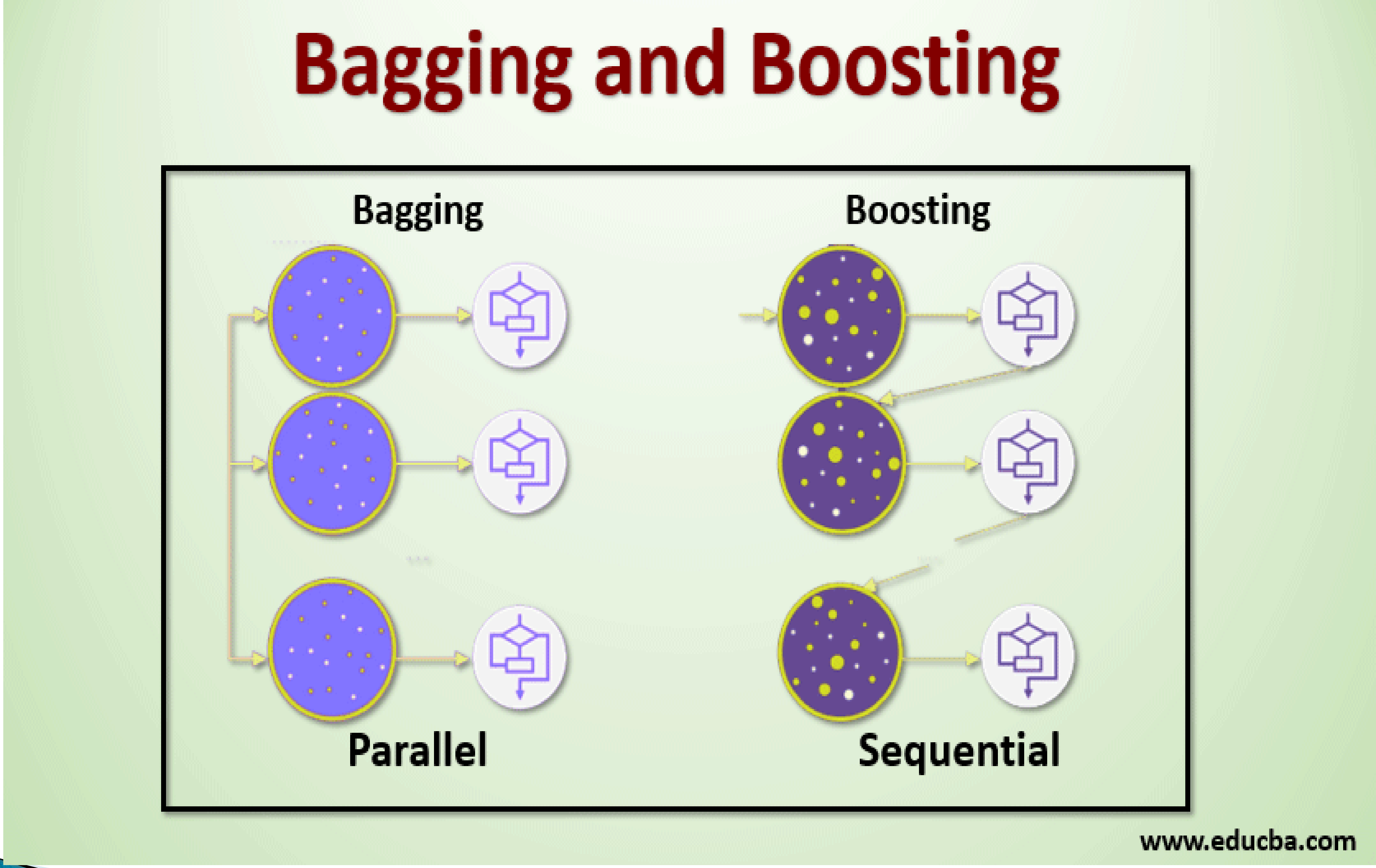
Combine
Classifiers

C

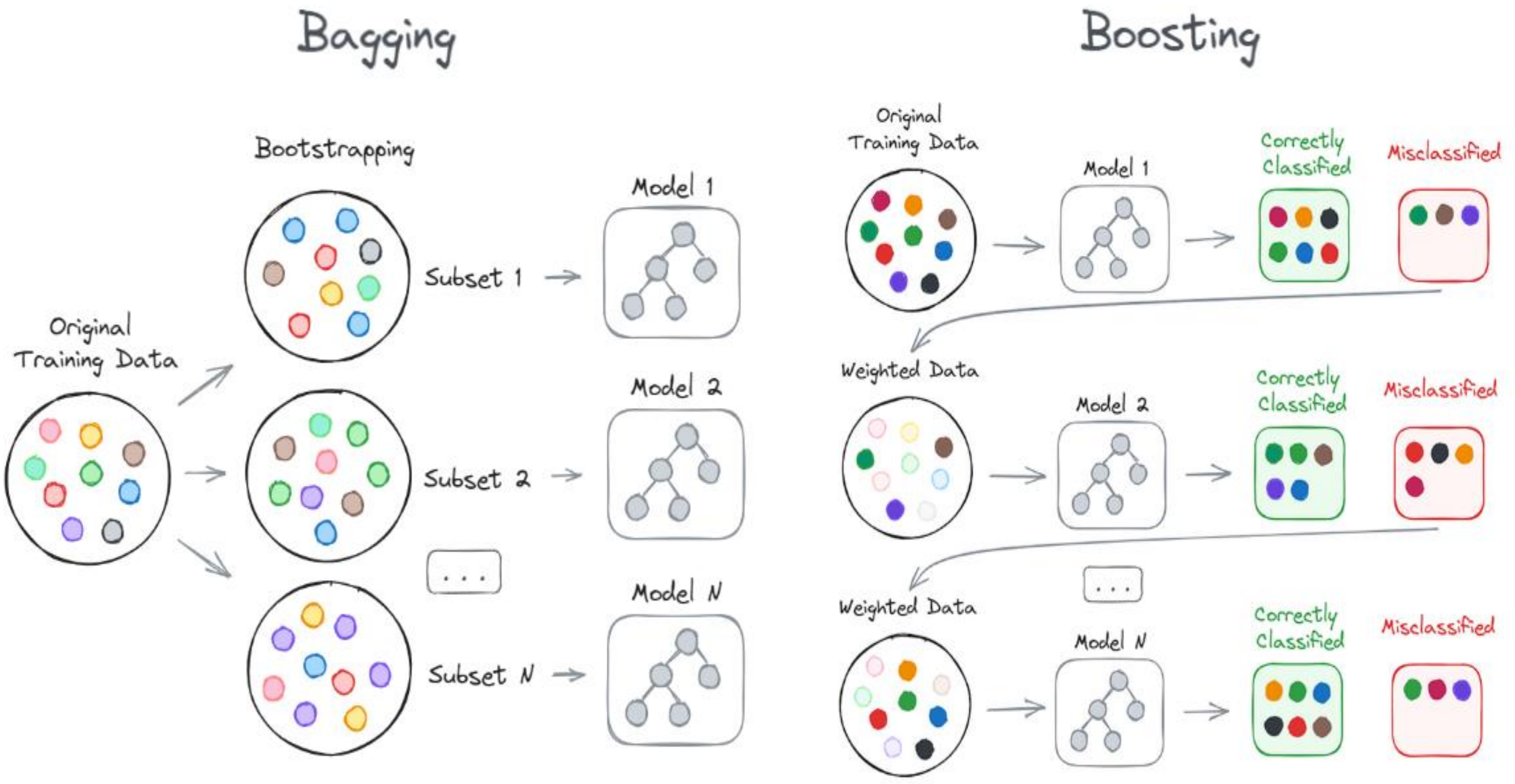
Ensemble Methods

- **Bagging**. Building multiple models (typically of the same type) from different subsamples of the training dataset.
- **Boosting**. Building multiple models (typically of the same type) each of which learns to fix the prediction errors of a prior model in the sequence of models.
- **Voting**. Building multiple models (typically of differing types) and simple statistics (like calculating the mean) are used to combine predictions.

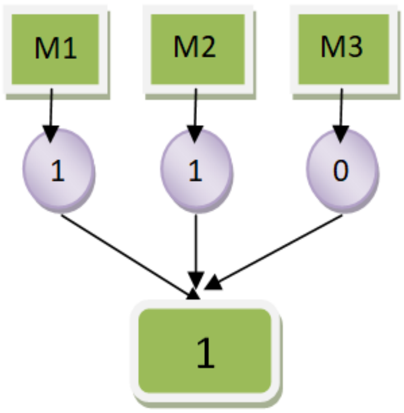
Continue…


Bagging and Boosting
Bagging
Boosting
Parallel
Sequential
www.educba.com

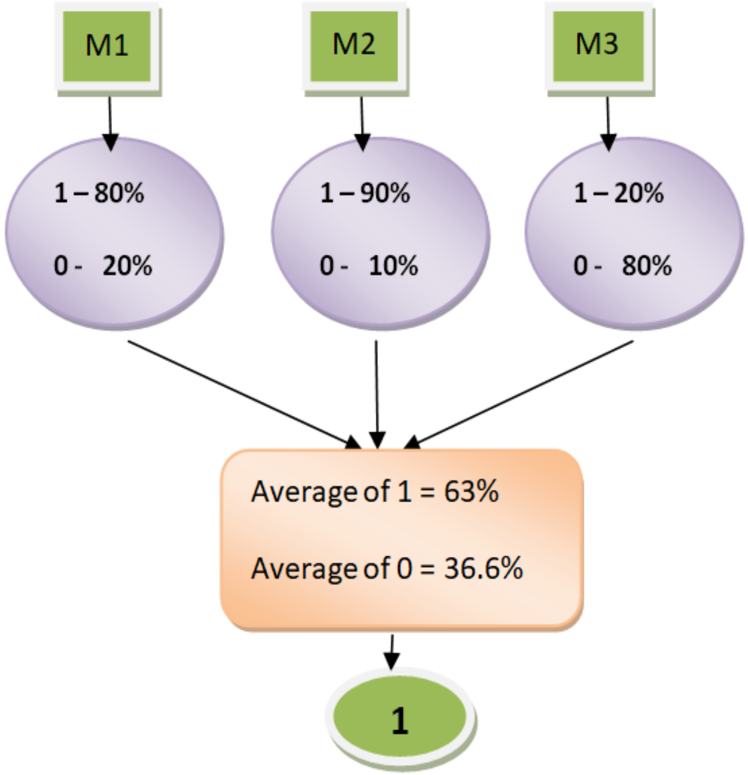
Continue…


Bagging
Boosting
Bootstrapping
Original Training Data
Subset 1
Subset 2
Subset N
...
Model 1
Model 2
Model N
Original Training Data
Model 1
Correctly Classified
Misclassified
Weighted Data
Model 2
Correctly Classified
Misclassified
Weighted Data
...
Model N
Correctly Classified
Misclassified

Continue…



Hard Voting



Soft Voting

Happy Learning