Machine Learning(22AIM52)
Introduction

Module I

Course Outcomes

- Understand the fundamental principles of Machine Learning and its applications.
- Apply logistic regression as a classification algorithm to demonstrate the proficiency in modelling binary and multi-class classification problems.
- Analyze the decision tree algorithm, its strengths and weaknesses in modeling complex decision boundaries and handling categorical and numerical data.
- Evaluate ML model performance (supervised and unsupervised algorithms) using advanced metrics such as precision, recall, F1 score, and ROC-AUC curve for effectiveness and robustness.
- Create effective classification models using supervised and unsupervised algorithms.
- Synthesize ethical considerations in Machine Learning (Algorithms), proposing strategies to address fairness, accountability, and transparency in model development

Syllabus

- **MODULE-1 Introduction to Machine Learning:**
- **Understanding Machine Learning:** Definition and Types of Machine Learning-Application of Machine Learning- Machine Learning Algorithms: Supervised, Unsupervised, and Semi-Supervised Learning Algorithms. Machine Learning Models-**Model Evaluation Metrics:** Confusion Matrix, Precision, Recall,F1 Score -ROC Curve and AUC-ROC. **Advanced Techniques:** Feature Scaling and Normalization – Encoding Categorical Variables-Train-test Split and Cross-validation.

- **MODULE-2 Supervised Learning Regression and Classification Algorithms**
- **Regression:** Introduction to Regression- Regression Models- Linear Regression, Polynomial Regression-. **Decision Trees:** Introduction to Decision Trees-Tree Construction, Splitting Criteria, and Pruning-Handling Missing Values and Categorical Features- Gini Index-ID3-CART
- **MODULE-3 Similarity based Models**
- **k-Nearest Neighbors (k-NN):** Introduction to k-Nearest Neighbors Algorithm-Distance Metrics and Choosing 'k' -k-NN for Classification. Logistic Regression- Multiclass Classification with Logistic Regression

- **MODULE-4 Probabilistic based Models**
- **Naive Bayes:** Introduction to Naive Bayes Classifier-Bayes' Theorem and Conditional Probability-Gaussian, Multinomial, and Bernoulli Naive Bayes. Bayesian Belief Network-EM algorithm
- **MODULE-5 Unsupervised Algorithms**
- Introduction to Rule Based learning and Association rules- Apriori Algorithm, FP-Growth. Introduction to unsupervised Algorithms- types of clustering algorithms- k-means clustering algorithms-DBSCAN clustering algorithm.

Module 1 - Syllabus

MODULE-1 Introduction to Machine Learning:

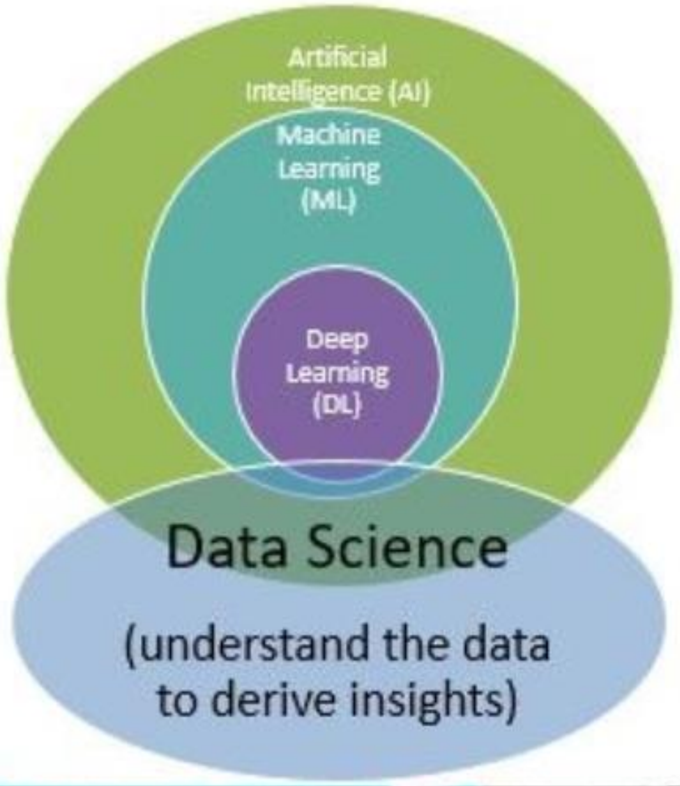
Understanding Machine Learning: Definition and Types of Machine Learning- Application of Machine Learning- Machine Learning Algorithms: Supervised, Unsupervised, and Semi-Supervised Learning Algorithms. Machine Learning Models

Model Evaluation Metrics: Confusion Matrix, Precision, Recall, F1 Score -ROC Curve and AUC-ROC.

Advanced Techniques: Feature Scaling and Normalization -Encoding Categorical Variables-Train-test Split and Cross-validation

AI Vs ML Vs DL

Artificial Intelligence (AI) vs Machine Learning vs Deep Learning vs Data Science


Artificial Intelligence (AI)
Machine Learning (ML)
Deep Learning (DL)
Data Science
(understand the data to derive insights)

- <u>Artificial Intelligence (AI):</u>
  - ➢ Broad concept of machines being able to carry out "smart" tasks
- <u>Machine Learning (ML):</u>
  - ➢ The use of statistical tools that help computers "learn" from data
  - ➢ Relies heavily on hand-crafted features
- <u>Deep Learning (DL):</u>
  - ➢ Subset of Machine Learning (ML)
  - ➢ Driven primarily by Neural Networks

What is Machine Learning

- The concept of learning in an ML system

Learning=Improving with experience at some task

- Improve over task T,

-With respect to performance measure, P

-Based on Experience, E.

Introduction to Machine Learning

- In algorithm based solution data and program are fed into the computer, program gets executed and the desired output is produced.
- In ML based solution data and target labels(output) is fed into the system, system learns from this data and build the model(program).

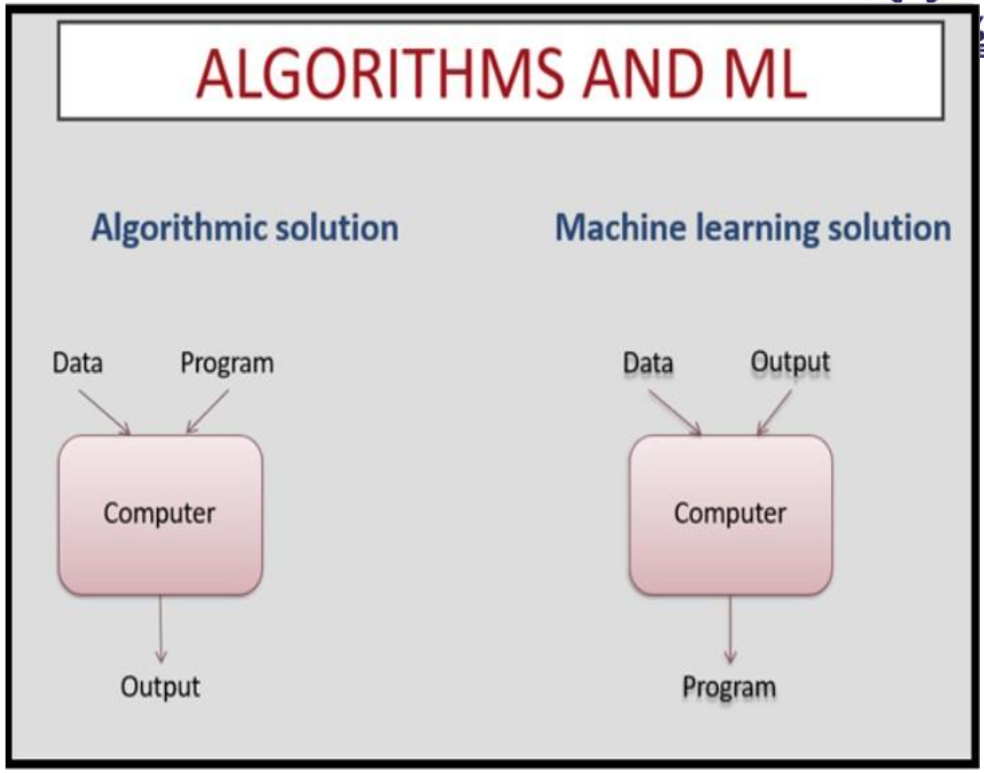


Figure 1: Diagram representing difference between Algorithmic and ML based solution

Definition of Machine Learning

- Machine Learning explores algorithms that can
  - learn from data / build a model from data
  - use the model for prediction, decision making or solving some tasks
- **Machine learning (ML) is** defined as a discipline of artificial intelligence (AI) that provides machines the ability to automatically learn from data and past experiences to identify patterns and make predictions with minimal human intervention.

Terminologies in Machine Learning

- **Data set** : Dataset is a collection of various types of data stored in a digital format.
- **Features:** A feature is one column of the data in your input set. feature is input;
  - The key identified properties based on which we would like to predict the result of our problem statement.
- **Labels** are also known as **tags**, which are used to give an identification to a piece of data and tell some information about that element.

-A **label** is the thing we're predicting

-Eg: Price of stock

- Labels are also referred to as the final output for a prediction. For example, as in the below image, we have labels such as a cat and dog, etc.
- label is output. This applies to both classification and regression problems.

Terminologies in Machine Learning

- **Training data** is the initial dataset you use to teach a machine learning application to recognize patterns or perform to your criteria.
- **Testing or validation data** is used to evaluate your model's accuracy.
- An **Algorithm** is a set of rules that a machine follows to achieve a particular goal.
  - An algorithm can be considered as a recipe that defines the inputs, the output and all the steps needed to get from the inputs to the output.
- **Model:** System that has been trained to recognize certain type of patterns.
  - A model defines the relationship between features and label.
  - Train the model over set of data by providing algorithm to learn from the data.

- **Regression vs. classification**

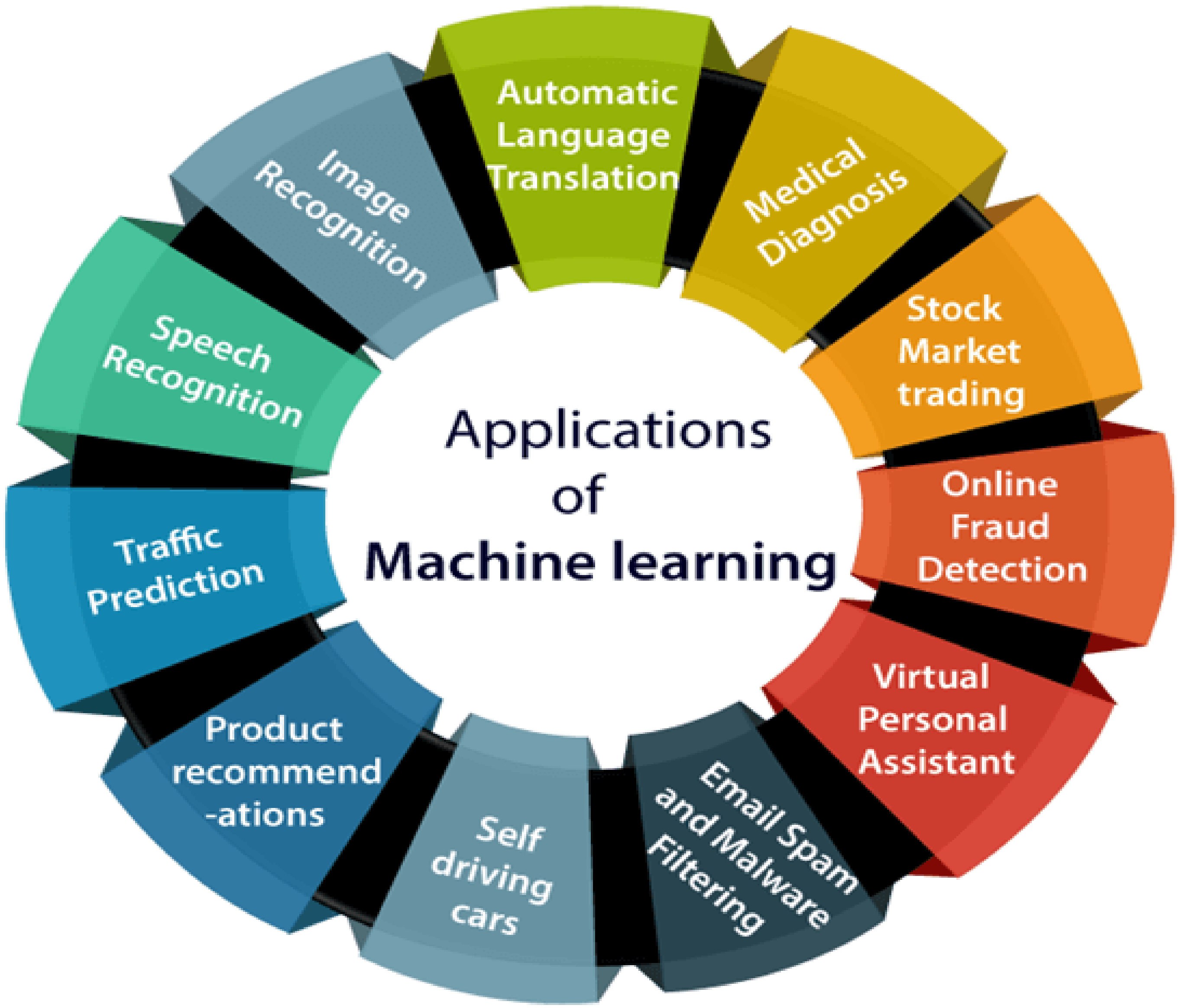
  -A **regression** model predicts continuous values.

- -Eg:What is the value of a house in Bangalore?(price of a house)

  -A **classification** model predicts discrete values.

  Eg: Is a given email message spam or not spam?

Applications of Machine learning



1. Image Recognition:

- It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:
- Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.
- It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

- While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**."
- At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction

- If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.
- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:
  - **Real Time location** of the vehicle form Google Map app and sensors
  - **Average time has taken** on past days at the same time.
- Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

- Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user.
- Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.
- Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.
- As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning

5.Self-driving cars:

- One of the most exciting applications of machine learning is self-driving cars.
- Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car.
- It is using unsupervised learning method to train the car models to detect people and objects while driving.

- **6. Email Spam and Malware Filtering:**

- Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning.
- Some machine learning algorithms such as **Multi-Layer Perceptron**, **Decision tree**, and **Naïve Bayes classifier** are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

- We have various virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.
- These virtual assistants use machine learning algorithms as an important part.
- These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

- Machine learning is making our online transaction safe and secure by detecting **fraud transaction**.
- Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts**, **fake ids**, and **steal money** in the middle of a transaction.
- So to detect this, **ML** helps us by checking whether it is a genuine transaction or a fraud transaction.
- A machine learning model is trained on a dataset of historical data, including both fraudulent and non-fraudulent transactions. The model learns to recognize patterns and deviations from normal behavior.

9. Stock Market trading

- Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning algorithm is used for the prediction of stock market trends.
- Machine learning algorithms such as regression, classifier, and support vector machine (SVM) help predict the stock market.

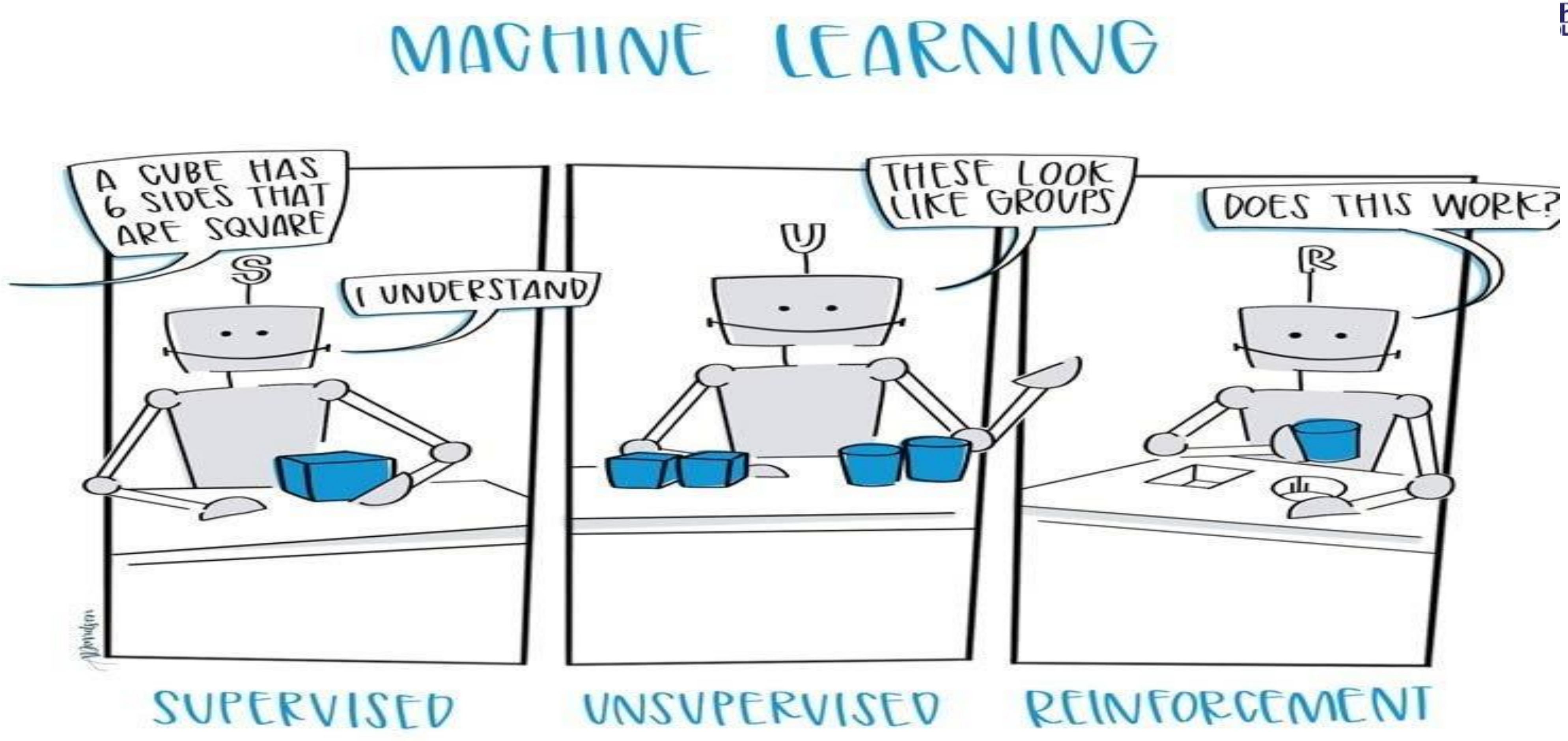
10. Medical Diagnosis:

- In medical science, machine learning is used for *diseases diagnoses*. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.

11. Automatic Language Translation:

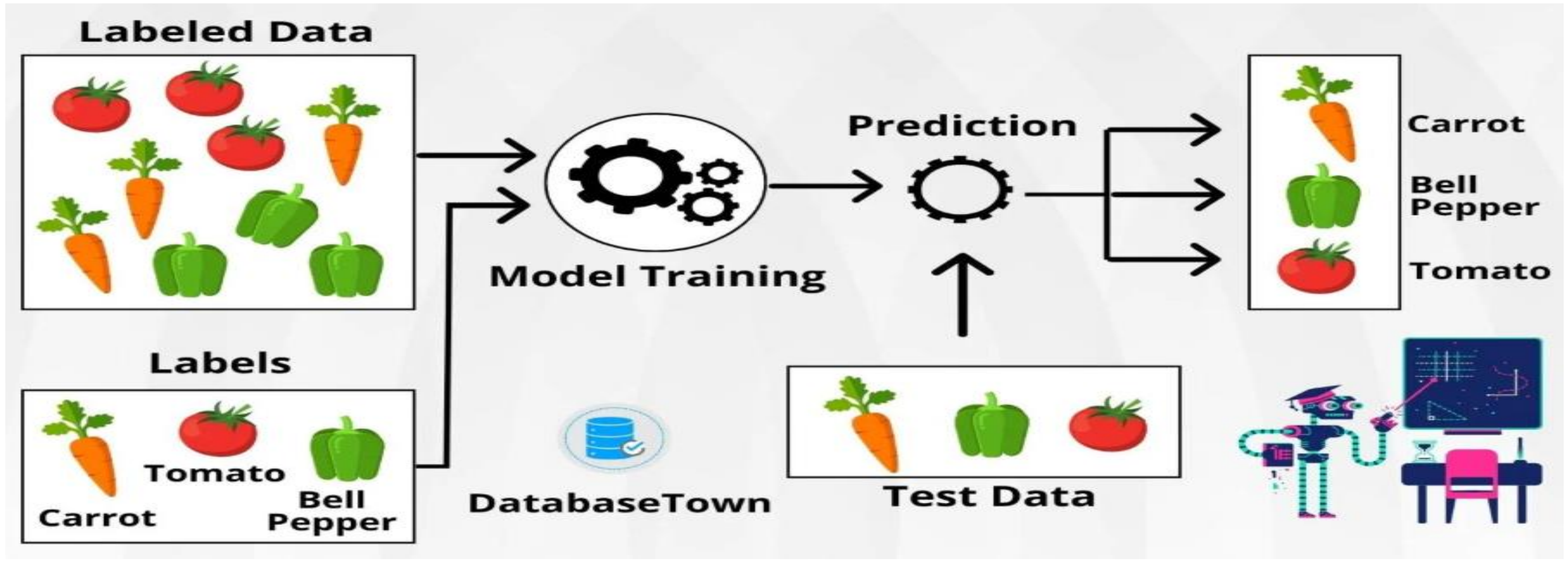
- Machine learning helps us by converting the text into our known languages.
- **Google's GNMT (**Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation. It is based on natural language processing and deep learning, and can translate both written and spoken languages.
- Training: The system learns how to translate by analyzing examples of texts in multiple languages and their translations.
- Translation: When a new text is inputted, the system uses what it's learned to generate a translation.
- Refining: Some additional adjustments may be added to refine the results

Types of Machine Learning



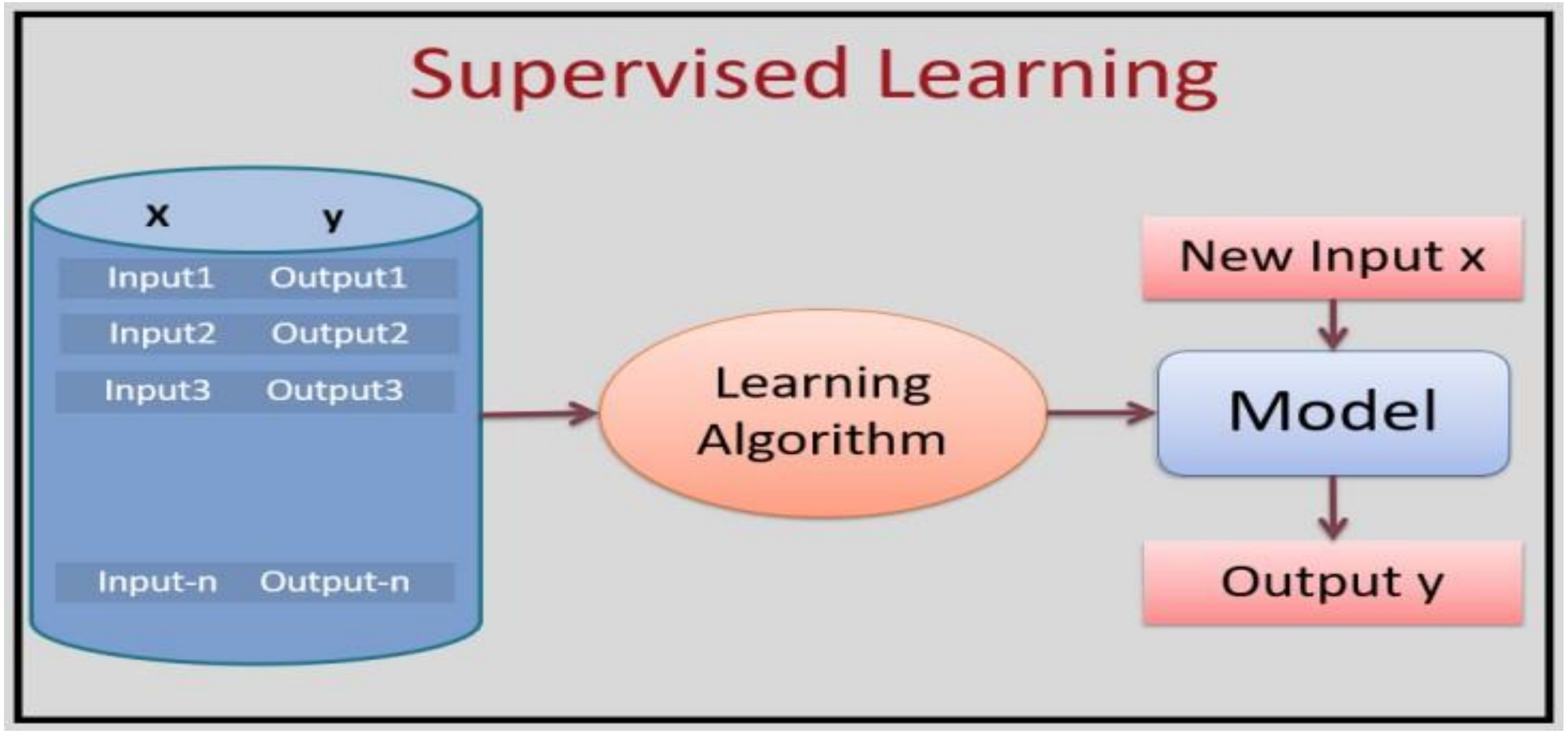
Supervised Learning

- Focuses on training models to make predictions or decisions based on ***labeled training data.***
- It involves a learning process where the model ***learns from known examples*** to predict or classify unseen or future instances accurately.
- ***Classification and Regression problems*** are Supervised learning problems.



Supervised Learning

- Supervised learning algorithms are used when the **output is classified or labeled.**
- These algorithms learn from the past data that is inputted, called training data, runs its analysis and uses this analysis to predict future events of any new data within the known classifications.
- The accurate prediction of test data requires large data to have a sufficient understanding of the patterns.
- The algorithm can be trained further by comparing the training outputs to actual ones and using the errors for modification of the algorithms.


Supervised Learning
x
y
Input1
Output1
Input2
Output2
Input3
Output3
Input-n
Output-n
Learning Algorithm
New Input x
Model
Output y

Advantages of Supervised Learning

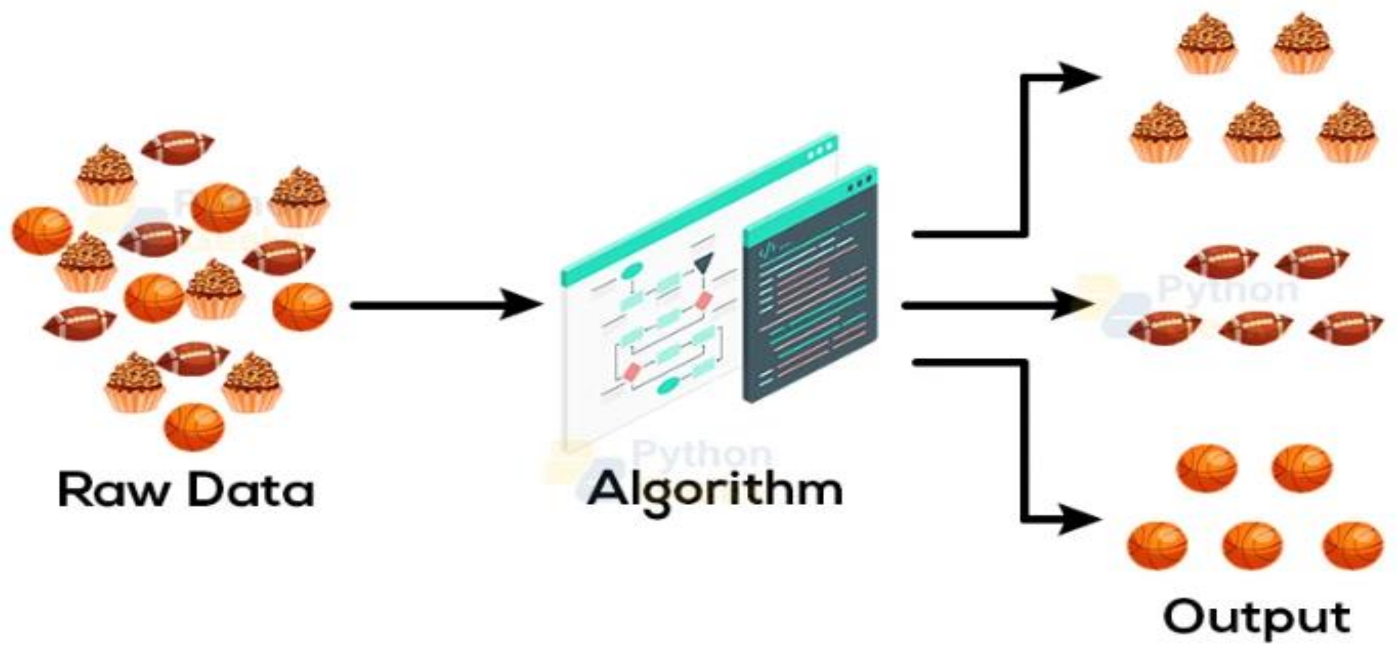
- Since supervised learning works with the labeled dataset, we can have an *exact idea about the classification* of objects and other variables that we feed in as input.
- These algorithms are helpful in predicting the output on the basis of *prior experience and trained data.*

Disadvantages of Supervised Learning

- These algorithms are not able to solve complex tasks due to a lack of data.
- It may predict the *wrong output if the test data is different from the training data* or the training data has some noise.
- It requires *lots of computational time to train* the algorithm and a huge load of trained data.

Unsupervised Machine Learning

- In unsupervised machine learning, we train the machine using the unlabeled or untrained dataset, and the machine predicts the output without any supervision of such data.



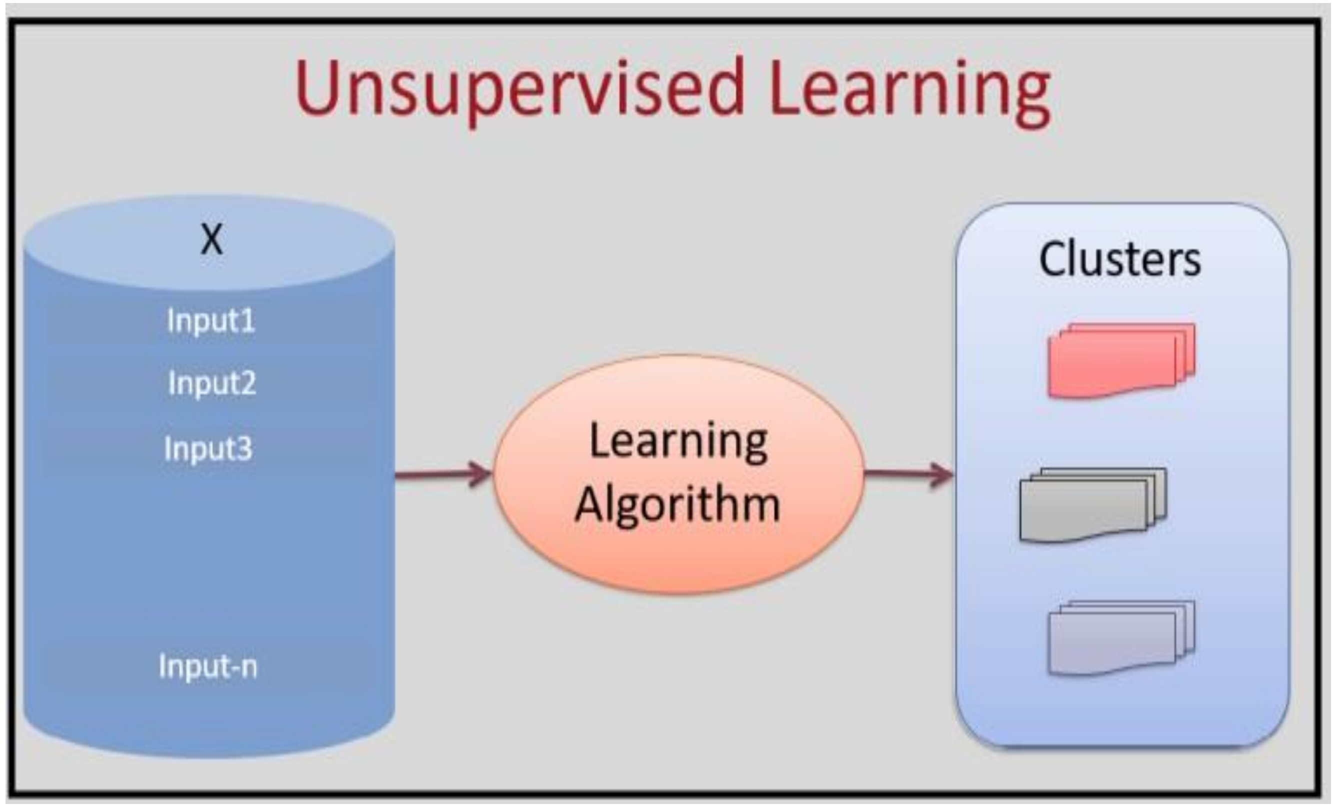
Advantages of Unsupervised Learning

- We can use these algorithms for complicated tasks as compared to the supervised ones because these algorithms work on unlabeled datasets and *do not require large datasets*.
- Unsupervised algorithms are *preferable for various tasks* as getting the unlabeled dataset is easier as compared to the labeled dataset for the training of these algorithms.

Disadvantages of Unsupervised Learning

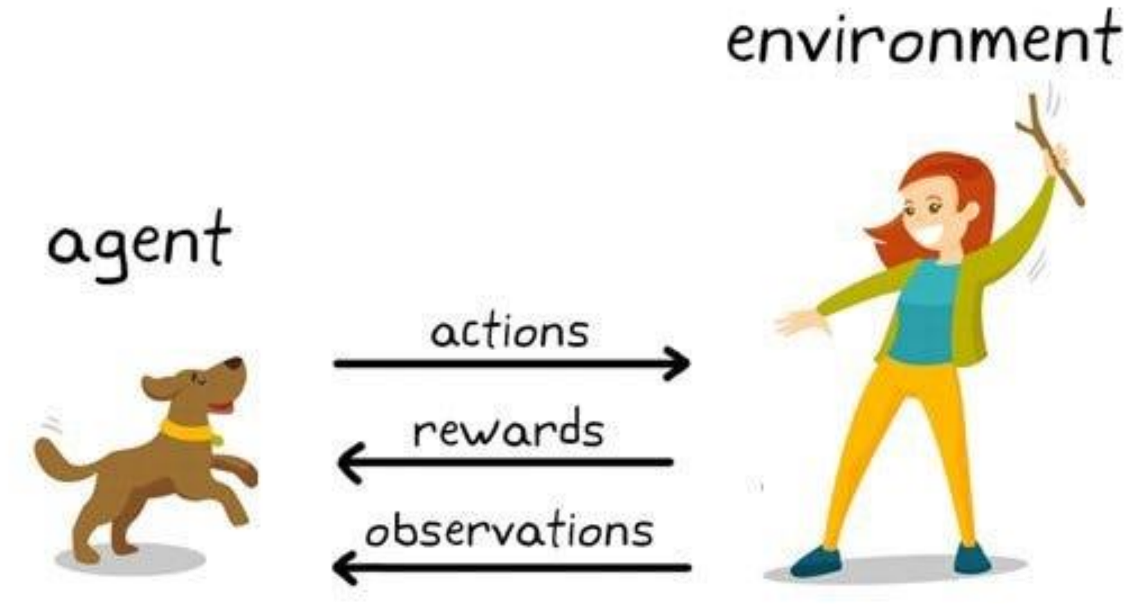
- An unsupervised algorithm may result *in less accurate outputs* as the dataset is not labeled, and we have not trained the algorithms with the exact output beforehand.
- Working with Unsupervised learning is more *difficult as compared to other types* as it works with the unlabelled dataset that does not map with the output precisely.

- The goal of unsupervised learning is to **find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format**.



Reinforcement Learning

- Reinforcement learning trains a machine to ***take suitable actions and maximize a reward in a particular situation.***
- It uses an ***agent and environment*** to produce actions and rewards.



Reinforcement Learning

- Reinforcement learning operates on a **feedback-based process**, in which an Artificial Intelligence agent automatically explores its surroundings by *hit and trial methods*.
- It takes action, *learns from experiences*, and improves its performance. The algorithm **rewards** such agents for **each good action** and **punishes** it for each **bad action**.
- Hence, the reinforcement learning agent aims to **maximize** the rewards and **minimize** the punishments.
- In reinforcement learning, the algorithm does not require any labeled data like supervised learning, and agents solely learn from their experiences.

Advantages of Reinforcement Learning

- This type of learning assists us in solving *complex real-world problems* which are difficult to be solved by general techniques that we use conventionally.
- The learning model of Reinforcement Learning is similar to the learning process of human beings; therefore, we can look for the most accurate results.
- This type of learning helps us in achieving *long-term results*.

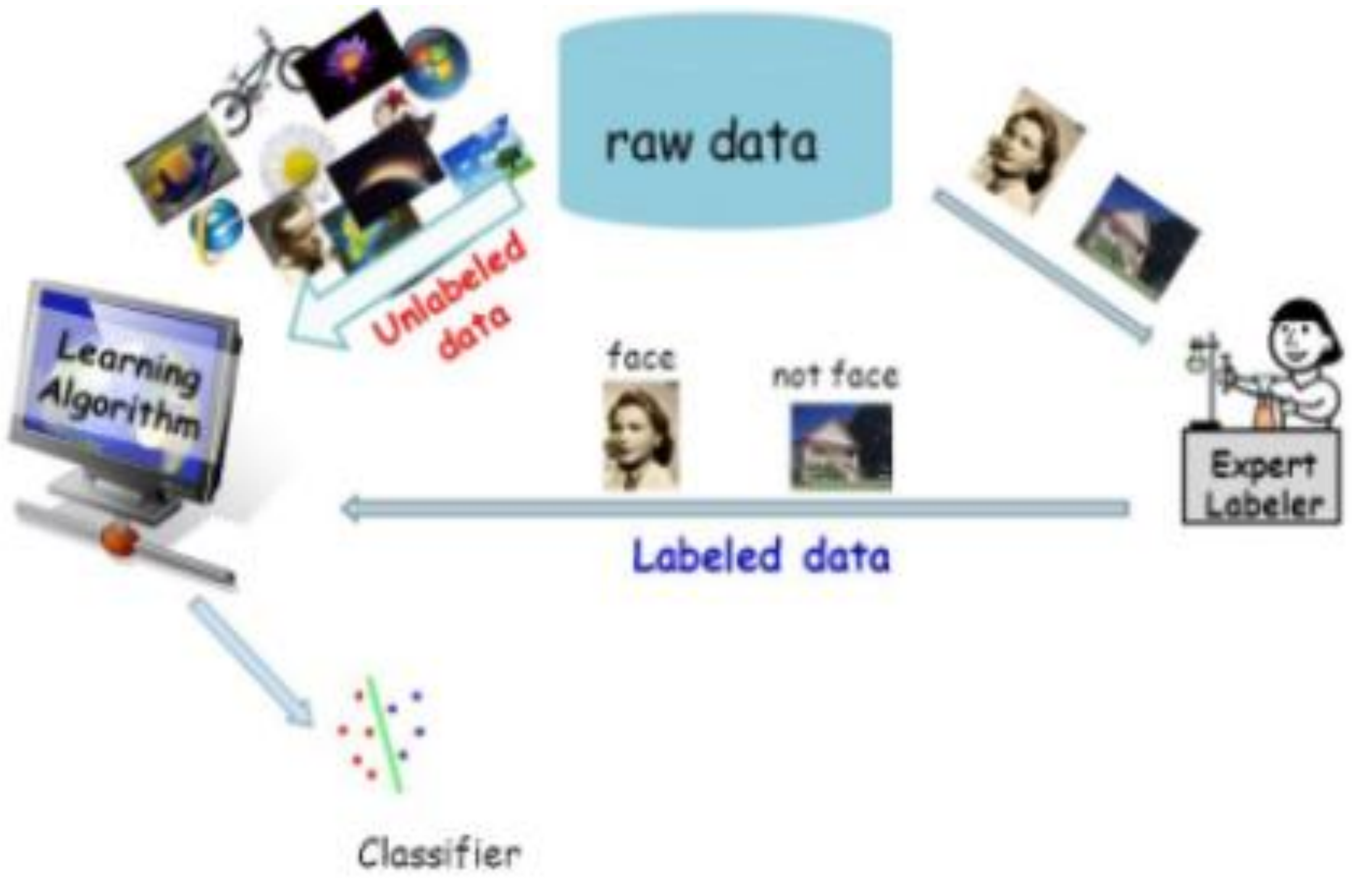
Disadvantages of Reinforcement Learning

- We generally do not prefer these algorithms for simple problems.
- These algorithms require *huge data and high computational powers*.
- Too much reinforcement learning can lead to an *overload of states* which can weaken the results defying the purpose of deploying them.

Semi-Supervised Learning

- Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms.
- It uses the combination of labeled and unlabeled datasets during the training period.
- The basic disadvantage of supervised learning is that it requires hand-labeling by ML specialists or data scientists, and it also requires a high cost to process.
- Unsupervised learning also has a limited spectrum for its applications.
- **To overcome these drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced**.

Semi-Supervised Learning



- In semi-supervised learning algorithm, training data is a combination of both labeled and unlabeled data.
- However, labeled data exists with a very small amount while it consists of a huge amount of unlabeled data.
- Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labeled data.

Semi Supervised Learning

Advantages of semi-supervised learning

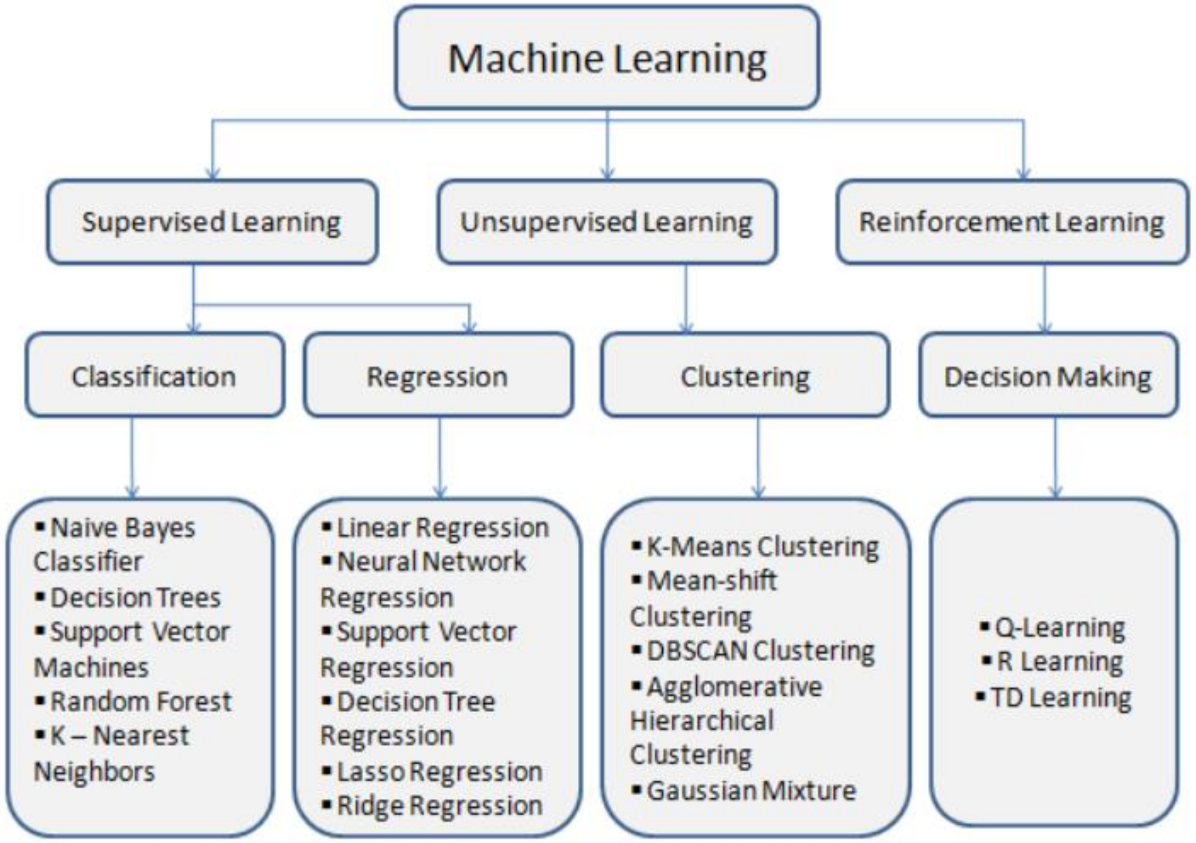
- It is quite simple and easy to understand the algorithm and does not encounter anomalies.
- This is *highly efficient in predicting the output* on the basis of input data.
- It overcomes the drawbacks of Supervised and Unsupervised Learning algorithms.

Disadvantages of semi-supervised learning

- Iteration results may not be stable and outputs may vary significantly.
- We cannot apply these algorithms to *network-level data due to its complexities*.
- The *accuracy rate* for this type of Learning is *low*.

Criteria	Supervised ML	Unsupervised ML	Reinforcement ML
Definition	Learns by using labelled data	Trained using unlabelled data without any guidance.	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No – predefined data
Type of problems	Regression and classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra supervision	No supervision	No supervision
Algorithms	Linear Regression, Logistic Regression, SVM, KNN etc.	K – Means, C – Means, Apriori	Q – Learning, SARSA
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of action
Application	Risk Evaluation, Forecast Sales	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

Types of Machine Learning Algorithms



Machine Learning Models and Performance Measures

Machine Learning models are programs that has been trained to find patterns within new data and make predictions.

These models are represented as a mathematical function that takes requests in the form of input data, makes predictions on input data, and then provides an output in response.

First, these models are trained over a set of data, and then they are provided an algorithm to reason over data, extract the pattern from feed data and learn from those data.

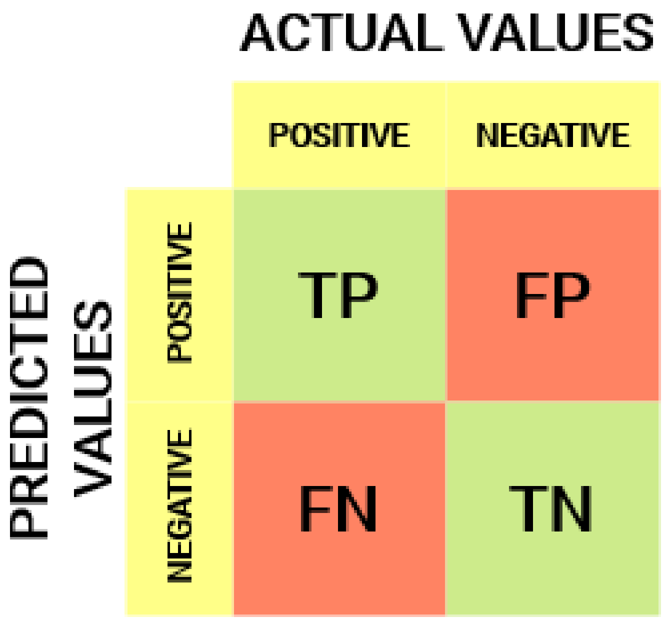
Once these models get trained, they can be used to predict the unseen dataset.

Model Evaluation Metrics: Confusion Matrix,

- Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values.
- A confusion matrix is defined as the table that is often used to describe the **performance of a classification model** on a *set of the test data* for which the *true values are known.*

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. or a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:



F

- Let’s take an example of a patient who has gone to a doctor with certain symptoms. Since it’s the season of Covid, let’s assume that he went with fever, cough, throat ache, and cold. These are symptoms that can occur during any seasonal changes too. Hence, it is tricky for the doctor to do the right diagnosis.

- ***True Positive (TP):***
- Let’s say the patient was actually suffering from Covid and on doing the required assessment, the doctor classified him as a Covid patient. This is called TP or True Positive.

- This is because the case is positive in real and at the same time the case was **classified correctly**. Now, the patient can be given appropriate treatment which means, the decision made by the doctor will have a positive effect on the patient and society.

- ***False Positive (FP):***
- Let’s say the patient was not suffering from Covid and he was only showing symptoms of seasonal flu but the doctor diagnosed him with Covid. This is called FP or False Positive.

- This is because the case was actually negative but was **falsely classified as positive**. Now, the patient will end up getting admitted to the hospital or home and will be given treatment for Covid.
- This is an unnecessary inconvenience for him and others as he will get unwanted treatment and quarantine. This is called **Type I Error**.

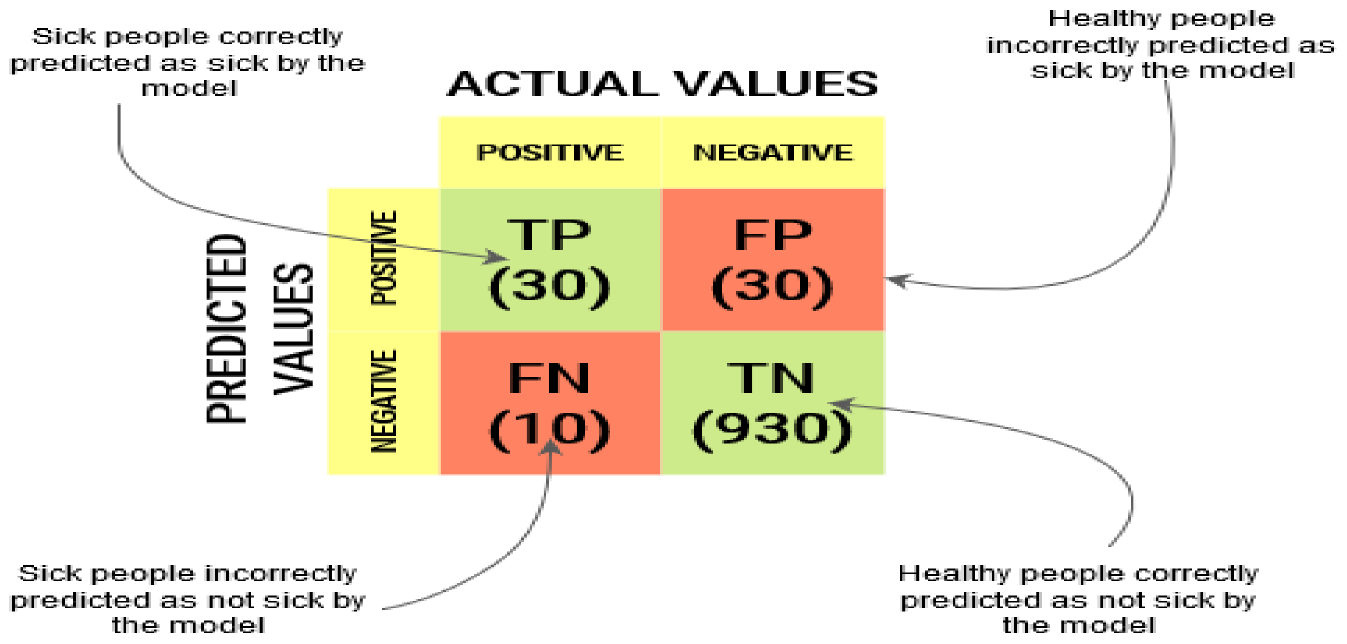
- ***True Negative (TN):***

- Let’s say the patient was not suffering from Covid and the doctor also gave him a clean chit. This is called TN or True Negative. This is because the case was **actually negative and was also classified as negative** which is the right thing to do. Now the patient will get treatment for his actual illness instead of taking Covid treatment.

- ***False Negative (FN):***

- Let’s say the patient was suffering from Covid and the doctor did not diagnose him with Covid. This is called FN or False Negative as the case was actually positive but was **falsely classified as negative**.

- Now the patient will not get the right treatment and also he will spread the disease to others. This is a highly dangerous situation in this example. This is also called **Type II Error.**


Sick people correctly predicted as sick by the model
ACTUAL VALUES
Healthy people incorrectly predicted as sick by the model
POSITIVE
NEGATIVE
PREDICTED VALUES
POSITIVE
NEGATIVE
TP (30)
FP (30)
FN (10)
TN (930)
Sick people incorrectly predicted as not sick by the model
Healthy people correctly predicted as not sick by the model

Accuracy:

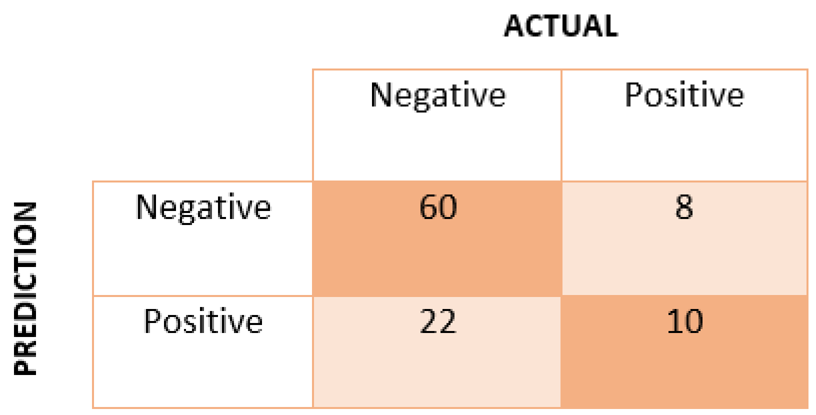
Accuracy = (TP + TN) / (TP + FP +TN + FN)

This term tells us how many right classifications were made out of all the classifications. In other words, how many TPs and TNs were done out of TP + TN + FP + FNs. It tells the ratio of "True"s to the sum of "True"s and "False"s.

- **Precision:**
  - Precision = TP / (TP + FP)
  - Out of all that positive predicted, how many are actually truly positive.
  - The precision value lies between 0 and 1.
- **Recall or Sensitivity:**
  - Recall = TP/ (TP + FN)
  - Out of all total positive cases, what percentage is predicted as positive.
- **F1-Score:**
  - F1 score = 2* (Precision * Recall) / (Precision + Recall)
  - F1- score is a weighted average of precision and recall.
  - It is the harmonic mean of precision and recall. It takes both false positive and false negatives into account. Therefore, it performs well on an imbalanced dataset.

Confusion Matrix-Example

- A machine learning model is trained to predict tumor in patients. The test dataset consists of 100 people.



		ACTUAL	
		Negative	Positive
PREDICTION	Negative	60	8
	Positive	22	10

- **True Positive (TP)** — model correctly predicts the positive class (prediction and actual both are positive). In the above example, **10 people** who have tumors are predicted positively by the model.
- **True Negative (TN)** — model correctly predicts the negative class (prediction and actual both are negative). In the above example, **60 people** who don’t have tumors are predicted negatively by the model.

NEW HORIZON
COLLEGE OF ENGINEERING

Python Code for Confusion Matrix

```python
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)
```

Example

- A model is used to predict whether patients have a disease based on some tests. After testing 100 patients, the results are as follows:
- True Positives (TP): 40
- True Negatives (TN): 30
- False Positives (FP): 10
- False Negatives (FN): 20
- **Question:** Construct the confusion matrix and calculate accuracy.
- Soln: Accuracy:70%

- A spam detection system analyzed 200 emails and produced the following results:
- 70 (correctly identified spam)
- 100 (correctly identified non-spam)
- 20 (non-spam incorrectly identified as spam)
- 10 (spam incorrectly identified as non-spam)
- **Question:** Create the confusion matrix and calculate accuracy, precision and recall.

Example

- ***Environmental scientists want to solve a two-class classification problem for predicting whether a population contains a specific genetic variant. Assuming the scientists use 500 samples for their data analysis. Assume the scientists predict that 350 test samples contain the genetic variant and 150 samples don't. If they determine the actual number of samples containing the variant is 305, the actual number of samples without the variant is 195.Draw the confusion matrix:***

1. **True Positives (TP)**: Samples correctly predicted to have the genetic variant.

2. **True Negatives (TN)**: Samples correctly predicted not to have the genetic variant.

3. **False Positives (FP)**: Samples incorrectly predicted to have the genetic variant.

4. **False Negatives (FN)**: Samples incorrectly predicted not to have the genetic variant.

AUC - ROC curve

- The ***Receiver Operator Characteristic (ROC)*** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR** against **FPR** at various threshold values and essentially **separates the ‘signal’ from the ‘noise’**.
- The ***Area Under the Curve (AUC)*** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.
- An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds.
- It is one of the most important evaluation metrics for checking any classification model’s performance. It is also written as AUROC (**Area Under the Receiver Operating Characteristics**)

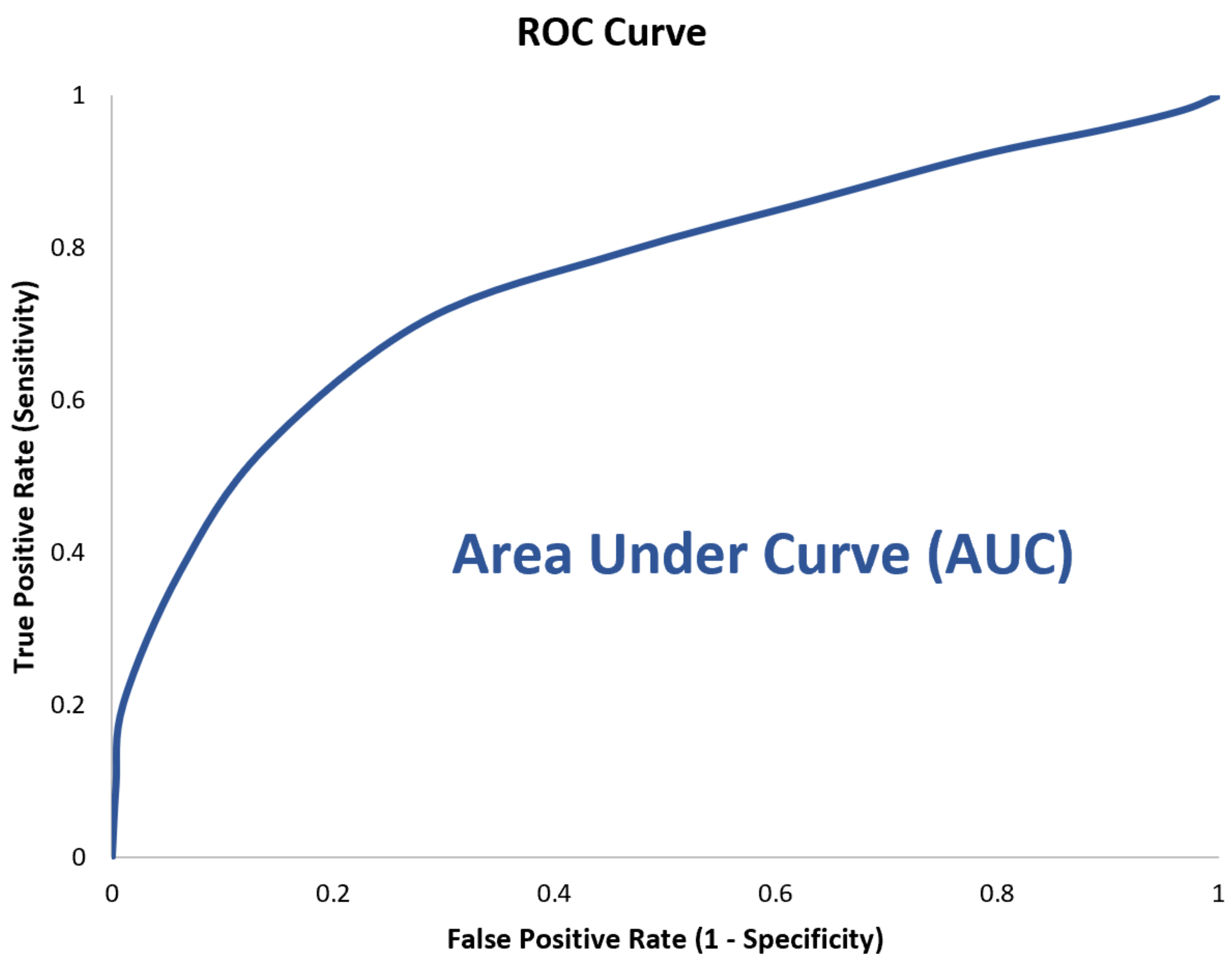
This curve plots two parameters:

- True Positive Rate
- False Positive Rate

ROC-AUC

- ROC AUC score shows how well the classifier distinguishes positive and negative classes. It can take values from 0 to 1.

- A higher ROC AUC indicates better performance. A perfect model would have an AUC of 1, while a random model would have an AUC of 0.5.



• True vs. False Positive rates:

- The ROC curve plots the True Positive rate (TPR) against the False Positive rate (FPR) at various classification thresholds.

- **TPR** (True Positive rate, also known as recall) shows the share of detected true positives.

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

- **FPR** (False Positive rate) shows the share of objects falsely assigned a positive class out of all objects of the negative class

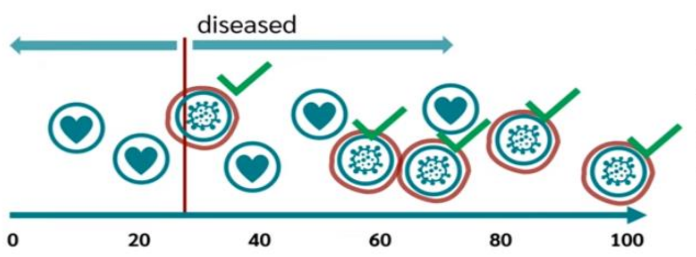
$$\text{FPR} = 1 - \text{Specificity}$$

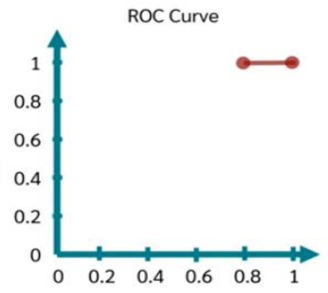
$$= \frac{FP}{TN + FP}$$

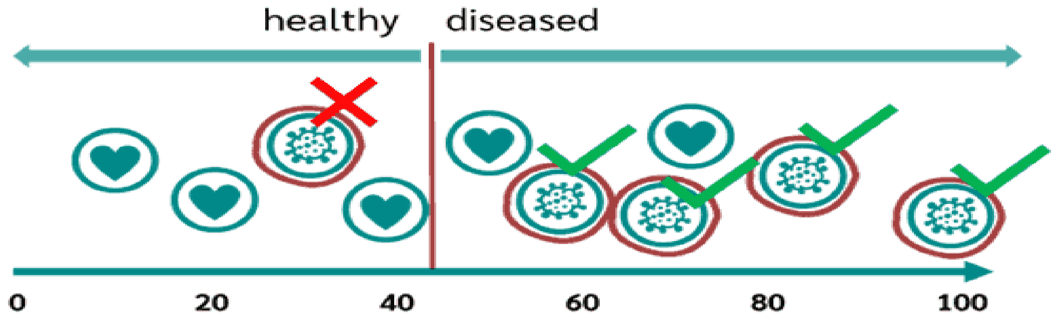
$$\text{Specificity} = \frac{TN}{TN + FP}$$

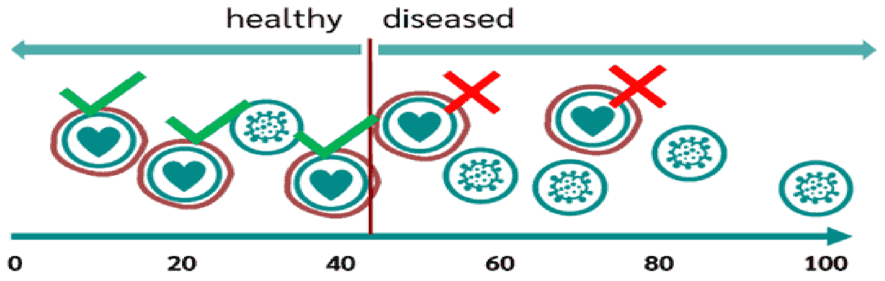
Example

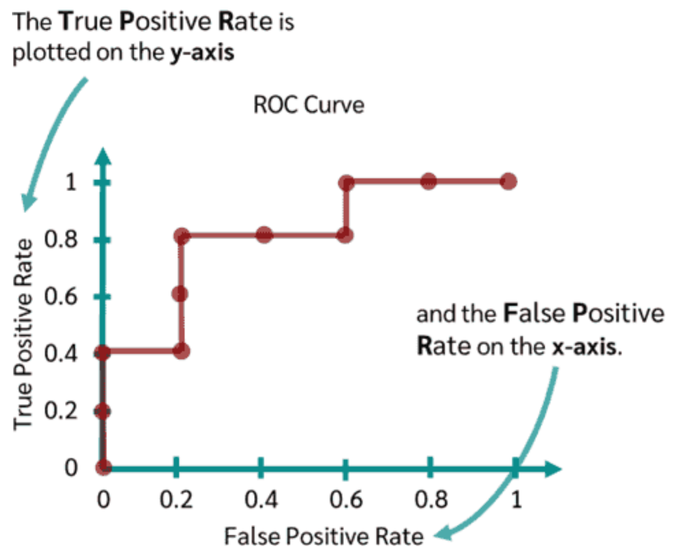
- If we choose the threshold value to be very small, i.e. pushing it all the way to the left, we correctly classify all 5 diseased individuals. Our True Positive Rate is thus 5 out of 5 i.e. 1.
- we also misclassify all 5 healthy persons as "diseased". Our False Positive Rate is therefore 5 out of 5, i.e. 1.



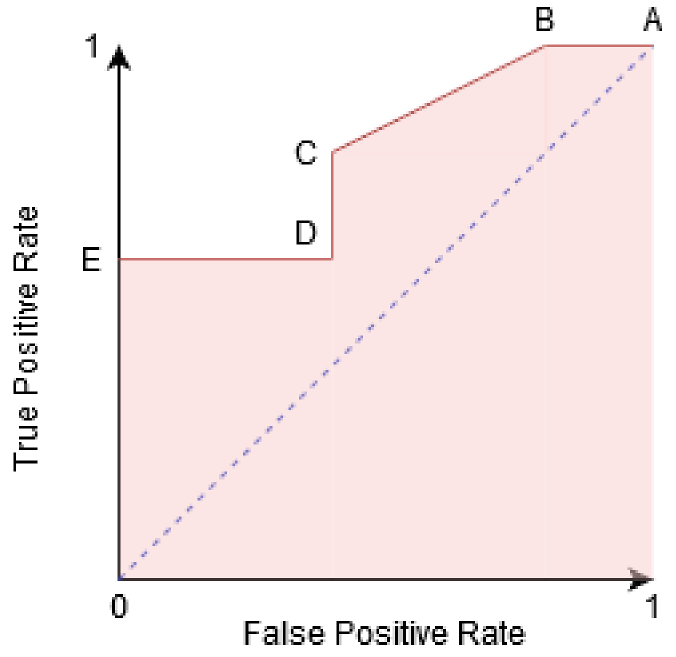






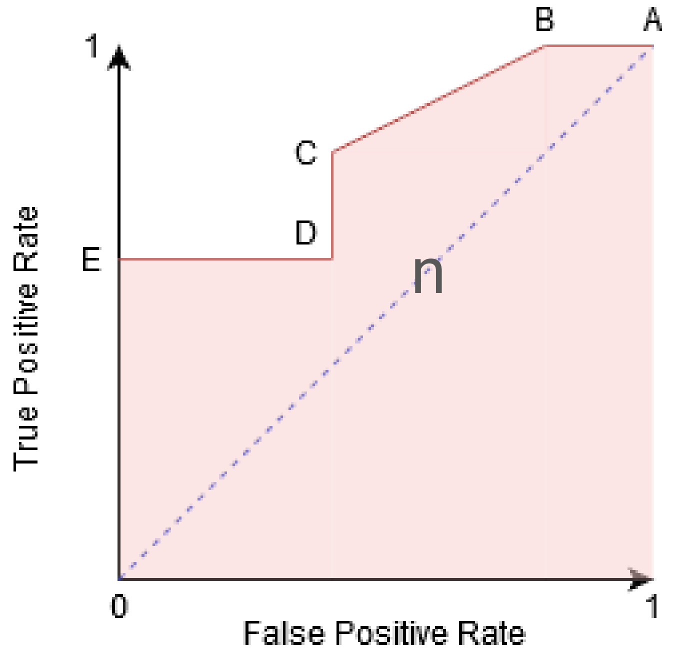


- Sensitivity and Specificity are inversely proportional to each other. when we increase Sensitivity, Specificity decreases, and vice versa.
- When we decrease the threshold, we get more positive values thus it increases the sensitivity and decreasing the specificity.
- Similarly, when we increase the threshold, we get more negative values thus we get higher specificity and lower sensitivity.



- Point A is where the Sensitivity is the highest and Specificity the lowest. This means all the Positive class points are classified correctly and all the Negative class points are classified incorrectly.

- Although Point B has the same Sensitivity as Point A, it has a higher Specificity. Meaning the number of incorrectly Negative class points is lower compared to the previous threshold. This indicates that this threshold is better than the previous one.



- Between points C and D, the Sensitivity at point C is higher than point D for the same Specificity. This means, for the same number of incorrectly classified Negative class points, the classifier predicted a higher number of Positive class points. Therefore, the threshold at point C is better than point D.

- Point E is where the Specificity becomes highest. Meaning there are no False Positives classified by the model.

Python Snippet

fpr, tpr, thresholds = roc_curve(y_test, y_prob)
Calculates the fpr, trpr and threshold values

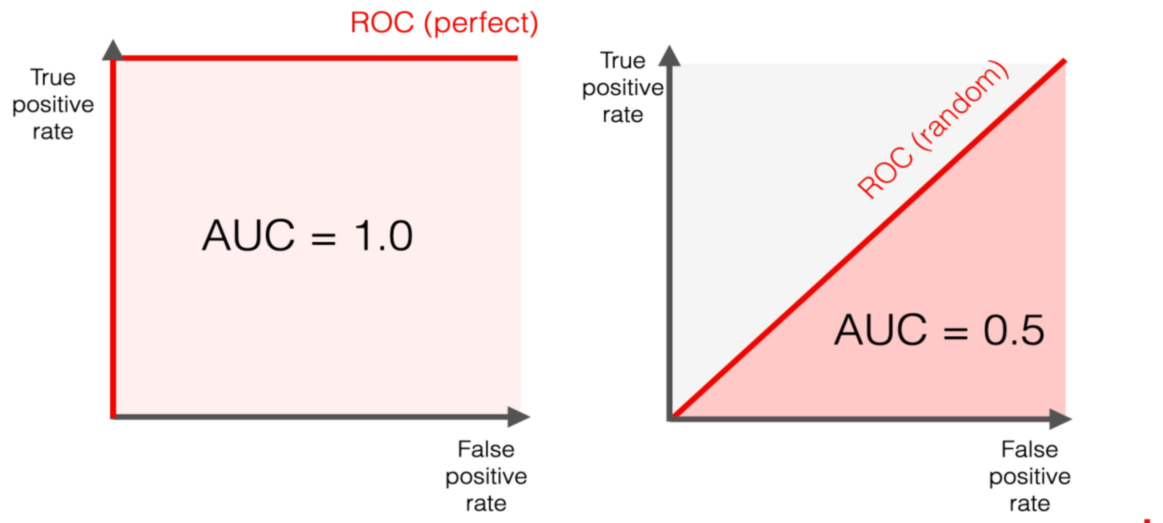
roc_auc = roc_auc_score(y_test, y_prob)

To print precision , recall and f1-score

```
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

AUC

- The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.
- The ROC AUC score can range from 0 to 1. A score of 0.5 indicates random guessing, and a score of 1 indicates perfect performance.
- A score slightly above 0.5 shows that a model has at least "some" predictive power. This is generally inadequate for any real applications.



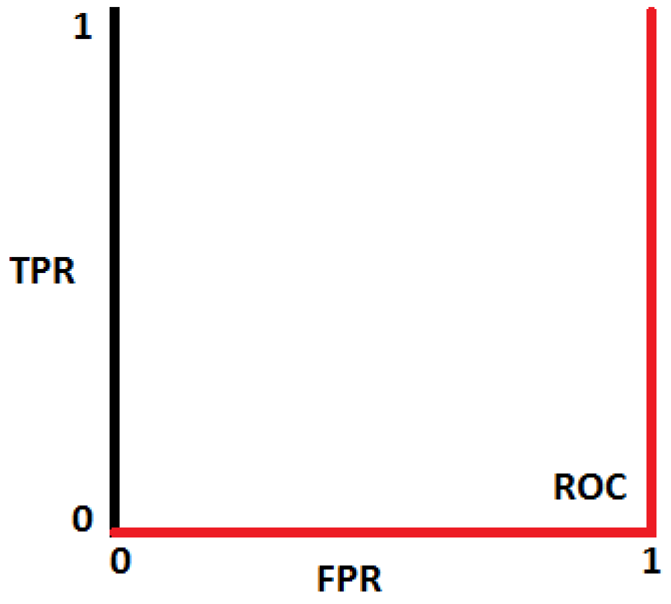
When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly.

AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.

When **0.5<AUC<1**, there is a high chance that the classifier will be able to distinguish the positive class values

When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points

- When AUC is approximately 0, the model is actually reciprocating the classes. It means the model is predicting a negative class as a positive class and vice versa.



What is a feature?

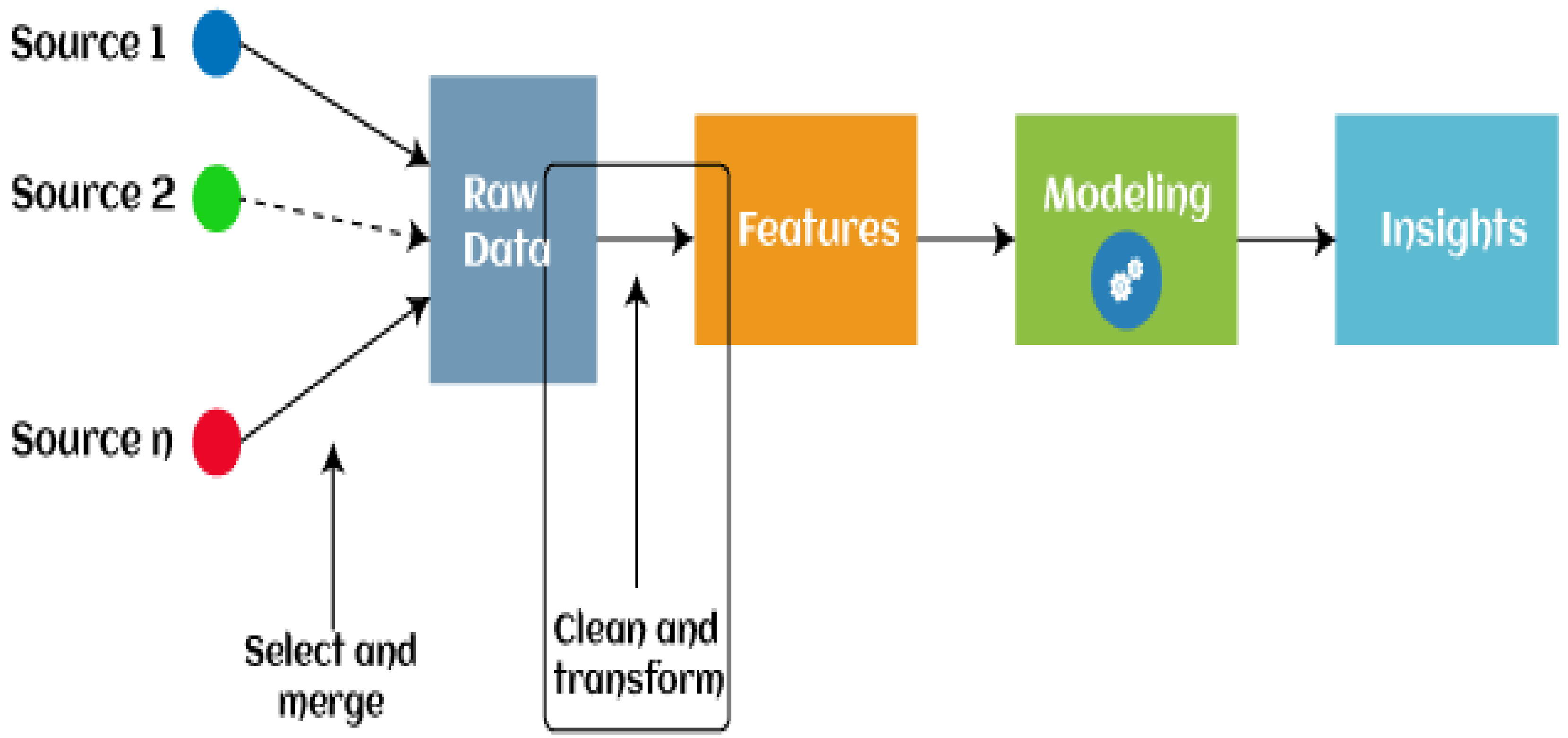
- Generally, all machine learning algorithms take input data to generate the output.
- The input data remains in a tabular form consisting of rows (instances or observations) and columns (variable or attributes), and these attributes are often known as **features**.
- The input variables that we give to our machine learning models are called features. Each column in our dataset constitutes a feature.
- Features are nothing but the **independent variables** in machine learning models. What is required to be learned in any specific machine learning problem is a set of these features (independent variables), coefficients of these features, and parameters for coming up with appropriate functions or models (also termed hyperparameters).

Example

- A model for predicting the risk of cardiac disease may have features such as the following:
  - Age
  - Gender
  - Weight
  - Whether the person smokes
  - Whether the person is suffering from diabetic disease, etc.

What is Feature Engineering?

- **Feature engineering is the pre-processing step of machine learning, which extracts features from raw data**.
- Feature engineering refers to a process of selecting and transforming variables when creating a predictive model using machine learning.
- It helps to *represent an underlying problem* to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data.
- The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.
- Eg: *size of the house*, *number of bedrooms, and location*, are the *predictor variables*. These are believed to determine the value of the property or house. *Actual prices* is the *outcome variable.*

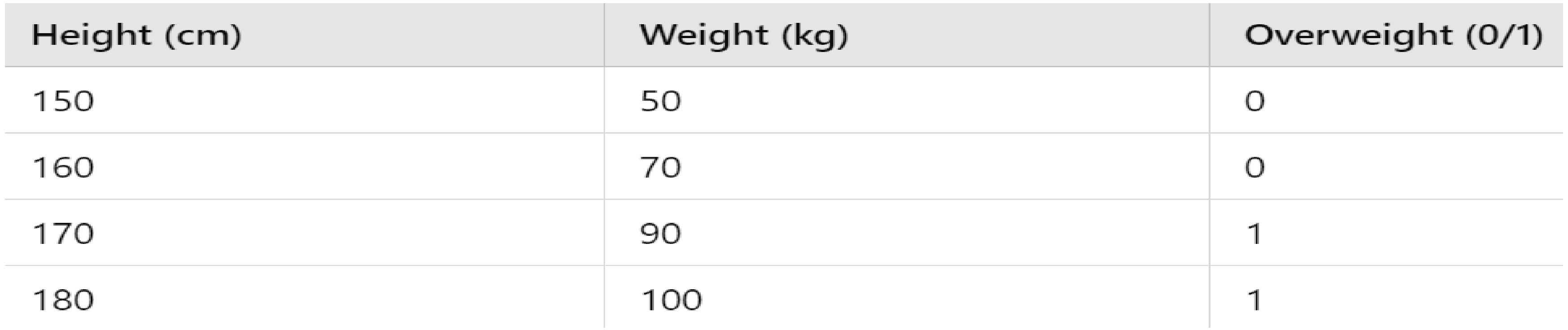

Source 1
Source 2
Source n
Raw Data
Features
Modeling
Insights
Select and merge
Clean and transform

Feature Scaling And Normalization

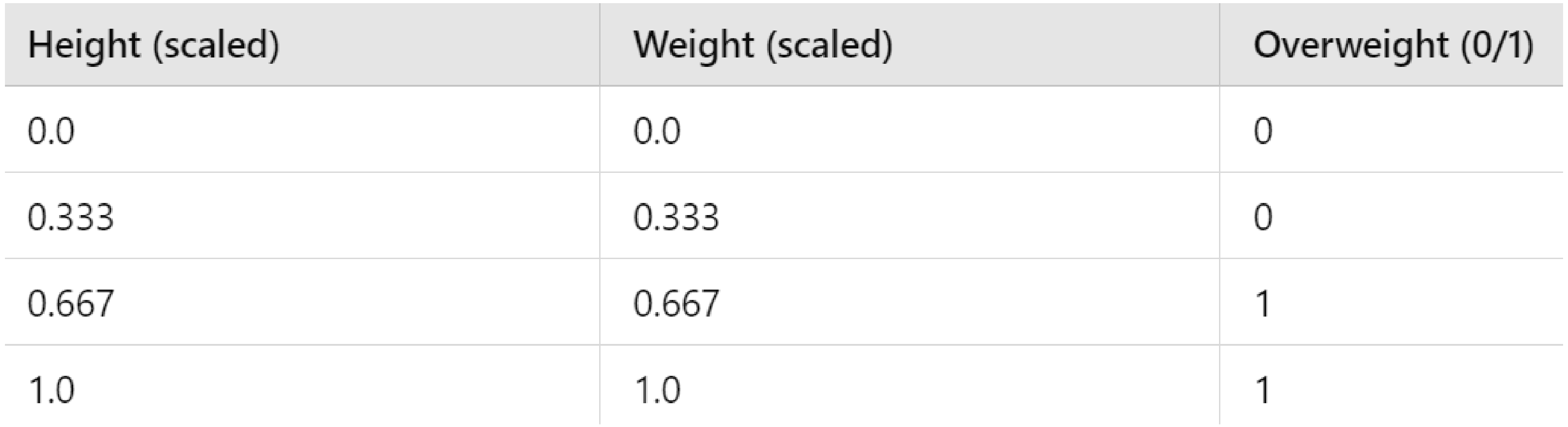
. Feature Scaling

- Feature Scaling is the process of transforming the features so that they have a similar scale. This is important in machine learning because the scale of the features can affect the performance of the model.
- Scaling guarantees that all features are on a comparable scale and have comparable ranges
- When features are on different scales, there is a risk that larger-scale features will dominate the model’s decisions, while smaller-scale features are neglected.

- Consider a simple example using two features: **height** (in cm) and **weight** (in kg) to predict whether a person is overweight.



Height (cm)	Weight (kg)	Overweight (0/1)
150	50	0
160	70	0
170	90	1
180	100	1



Height (scaled)	Weight (scaled)	Overweight (0/1)
0.0	0.0	0
0.333	0.333	0
0.667	0.667	1
1.0	1.0	1

Types of Feature Scaling

1. **Min-Max Scaling:** Rescaling the features to a specific range, such as between 0 and 1, by subtracting the minimum value and dividing by the range.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

X is the original value.

X_{min} is the minimum value of the feature.

X_{max} is the maximum value of the feature.

Disadvantages of Min-Max Scaling

Outlier Sensitivity: If there are extreme values in the dataset, the scaling can compress the majority of the data into a small range

Not Robust: If new data points fall outside the min-max range of the training data, they may be scaled improperly

Min-Max Scaling-Example

- A value is normalized as follows:
- y = (x – min) / (max – min)
- Where the minimum and maximum values pertain to the value x being normalized.
- For example, for a dataset, the min and max observable values as 30 and -10. We can then normalize any value, like 18.8, as follows:
- y = (x – min) / (max – min)
- y = (18.8 – (-10)) / (30 – (-10))
- y = 28.8 / 40
- y = 0.72

Python Code

- *from sklearn.preprocessing import MinMaxScaler*

- *# Initialize the MinMaxScaler*
- *scaler =* ***MinMaxScaler****()*

- *# Fit and transform the data*
- *scaled_data = scaler.fit_transform(df)*

- **Standard Scaling:** Rescaling the features to have a mean of 0 and a standard deviation of 1 by subtracting the mean and dividing by the standard deviation

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

- *# Initialize the StandardScaler scaler = StandardScaler()*
- *# Fit and transform the data scaled_data = scaler.fit_transform(df)*

Standard Scaling -Example

A value is standardized as follows:

y = (x – mean) / standard_deviation

Where the *mean* is calculated as:

mean = sum(x) / count(x)

And the *standard_deviation* is calculated as:

standard_deviation = sqrt(sum((x – mean)^2) / count(x))

We can estimate a mean of 10.0 and a standard deviation of about 5.0. Using these values, we can standardize the first value of 20.7 as follows:

y = (x – mean) / standard_deviation

y = (20.7 – 10) / 5

y = (10.7) / 5

y = 2.14

- *from sklearn.preprocessing import StandardScaler*

- *# Initialize the StandardScaler*

  scaler = StandardScaler()

- *# Fit and transform the data*
- *scaled_data = scaler.fit_transform(df)*

Robust Scaler

- **Robust Scaling:** Rescaling the features to be robust to outliers by dividing them by the interquartile range.

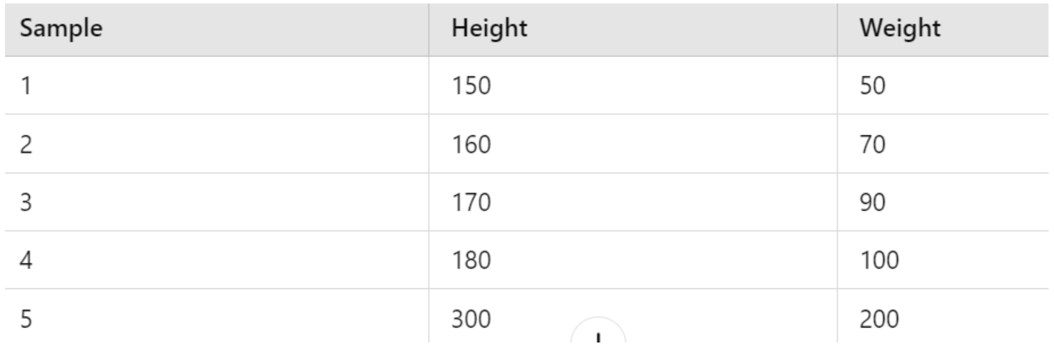
$$X_{scaled} = \frac{X - \text{median}(X)}{\text{IQR}(X)}$$

- Where:
- *Xis the original value.*
- *median(X)is the median of the feature values.*
- *IQR(X is the interquartile range, calculated as the difference between the 75th percentile and the 25th percentile.*

- Robust Scaler algorithms scale features that are robust to outliers.

- Where Q1 is the 1st quartile, and Q3 is the third quartile.

Example for Robust Scaler



Sample	Height	Weight
1	150	50
2	160	70
3	170	90
4	180	100
5	300	200

Height
median=170
25th percentile (Q1): 160
75th percentile (Q3): 180
QR=Q3−Q1=180−160=20
Weight:
median=90
25th percentile (Q1): 70
75th percentile (Q3): 100
IQR=Q3−Q1=100−70=30

Apply the Robust Scaling Formula
Height
Sample 1 (150): (150-170)/20 =-1.0
……….
Sample 5 (300):(300-170)/20 =6.5
Weight
Sample 1:50-90)/30 =-1.33
…………..
Sample 5:200-90)/30 =3.67

- *from sklearn.preprocessing import RobustScaler*

- *# Initialize the RobustScaler*
- *scaler = RobustScaler()*

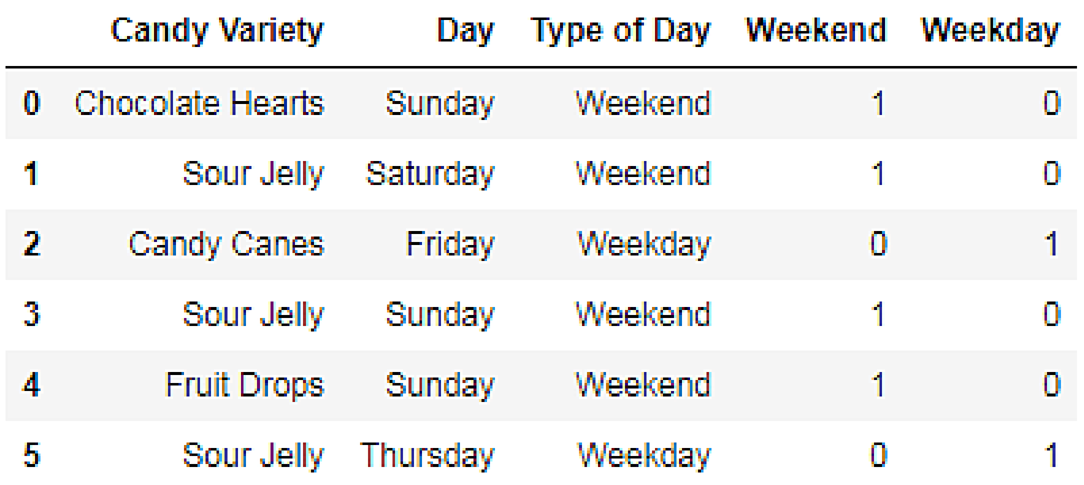
- *# Fit and transform the data*
- *scaled_data = scaler.fit_transform(df)*

Encoding Categorical Variables

- Encoding categorical variables is a crucial preprocessing step in machine learning when working with datasets that contain non-numeric (categorical) features.
- Many machine learning algorithms require numerical input, so categorical variables must be converted into a suitable format.
- 1. **Label Encoding:**
- Assigns a unique integer to each category.
- This method is simple but introduces an implicit ordinal relationship between categories, which might not always be desired.
- Example:For the categorical feature "Color" with values ["Red", "Blue", "Green"],
- Label Encoding might assign:
  - Red → 0
  - Blue → 1
  - Green → 2
- Best for ordinal variables (i.e., categories with a natural ordering, like ["Low", "Medium", "High"]).

One-Hot Encoding

- Converts each category into a new binary column. Each category gets its own column, and a row will have 1 for the present category and 0 for others.
- This method avoids introducing ordinal relationships.
- Example:For "Color" with values ["Red", "Blue", "Green"],
- One-Hot Encoding creates new columns:
- Red → [1, 0, 0]
- Blue → [0, 1, 0]
- Green → [0, 0, 1]
- Suitable for nominal (non-ordinal) categorical variables, especially in models that are sensitive to scale (e.g., linear models).



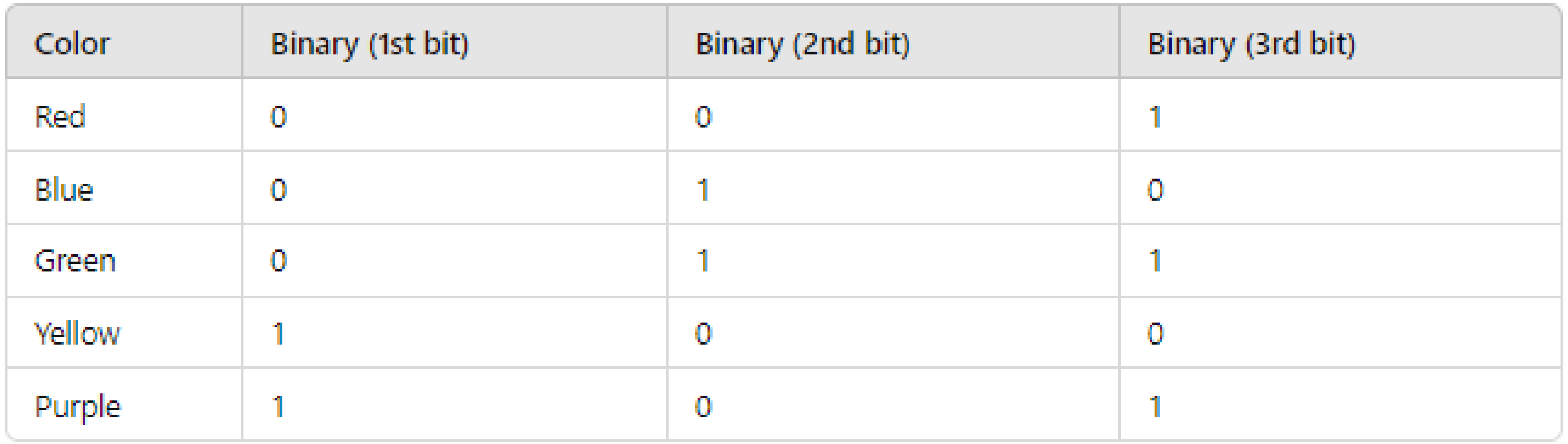
	Candy Variety	Day	Type of Day	Weekend	Weekday
0	Chocolate Hearts	Sunday	Weekend	1	0
1	Sour Jelly	Saturday	Weekend	1	0
2	Candy Canes	Friday	Weekday	0	1
3	Sour Jelly	Sunday	Weekend	1	0
4	Fruit Drops	Sunday	Weekend	1	0
5	Sour Jelly	Thursday	Weekday	0	1

.Binary Encoding

- Encodes categories as binary numbers and then splits them into columns.
- It is more memory-efficient than One-Hot Encoding when dealing with high cardinality (many categories).
- Step 1: Label Encoding :
- First, assign a unique integer to each category:
- Red → 1
- Blue → 2
- Green → 3
- Yellow → 4
- Purple → 5

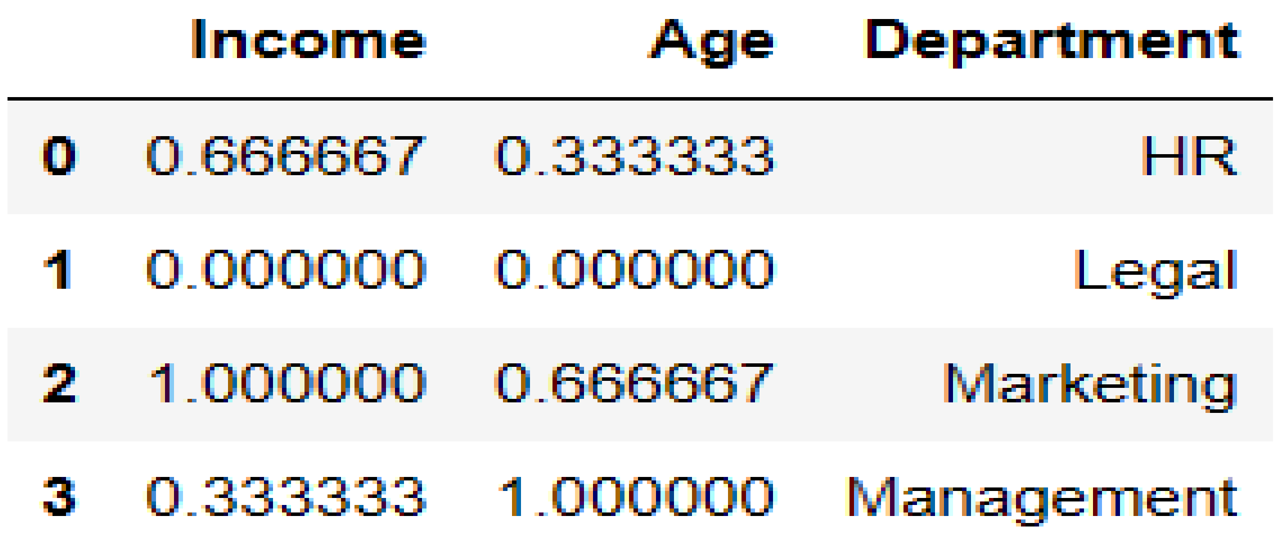
- Step 2: Convert to Binary
- Now, convert each integer into binary:
- Red (1) → 001
- Blue (2) → 010
- Green (3) → 011
- Yellow (4) → 100
- Purple (5) → 101

- Step 3: Split into Columns



Color	Binary (1st bit)	Binary (2nd bit)	Binary (3rd bit)
Red	0	0	1
Blue	0	1	0
Green	0	1	1
Yellow	1	0	0
Purple	1	0	1

- Now, instead of 5 columns (as you would get with One-Hot Encoding), you only need 3 columns to represent 5 categories.



	Income	Age	Department
0	0.666667	0.333333	HR
1	0.000000	0.000000	Legal
2	1.000000	0.666667	Marketing
3	0.333333	1.000000	Management

1. Feature Creation:

- Feature creation is finding the most **useful variables** to be used in a predictive model.
- The process is subjective, and it requires human creativity and intervention.
- The new features are created by **mixing existing features** using ***addition, subtraction, and ration,*** and these new features have great flexibility.
- **Types of Feature Creation:**

1. **Domain-Specific:** Creating new features based on **domain knowledge**, such as creating features based on business rules or industry standards.
2. **Data-Driven:** Creating new features by **observing patterns in the data**, such as calculating aggregations or creating interaction features.
3. **Synthetic:** Generating new features by combining existing features or synthesizing new data points.

2. Transformations:

- The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model.
- For example, it ensures that the model is flexible to take input of the variety of data; it ensures that all the variables are on the same scale, making the model easier to understand.
- It improves the model's accuracy and ensures that all the features are within the acceptable range to avoid any computational error.
- **Types of Feature Transformation:**

1. **Normalization:** Rescaling the features to have a similar range, such as between 0 and 1, to prevent some features from dominating others.
2. **Scaling:** Rescaling the features to have a similar scale, such as having a standard deviation of 1, to make sure the model considers all features equally.
3. **Encoding:** Transforming categorical features into a numerical representation. Examples are one-hot encoding and label encoding.
4. **Transformation:** Transforming the features using mathematical operations to change the distribution or scale of the features. Examples are logarithmic, square root, and reciprocal transformations.

3. Feature Extraction:

- Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data.
- The main aim of this step is to **reduce the volume of data** so that it can be easily used and managed for data modelling.
- Feature extraction methods include **cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA)**.
- **Types of Feature Extraction:**

1. **Dimensionality Reduction:** Reducing the number of features by transforming the data into a lower-dimensional space while retaining important information. Examples are PCA and t-SNE.
2. **Feature Combination:** Combining two or more existing features to create a new one. For example, the interaction between two features.
3. **Feature Aggregation:** Aggregating features to create a new one. For example, calculating the mean, sum, or count of a set of features.
4. **Feature Transformation:** Transforming existing features into a new representation. For example, log transformation of a feature with a skewed distribution.

4. Feature Selection:

- While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant.
- If we input the dataset with all these redundant and irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the model.
- Hence it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less important features, which is done with the help of feature selection in machine learning.
- **Types of Feature Selection:**

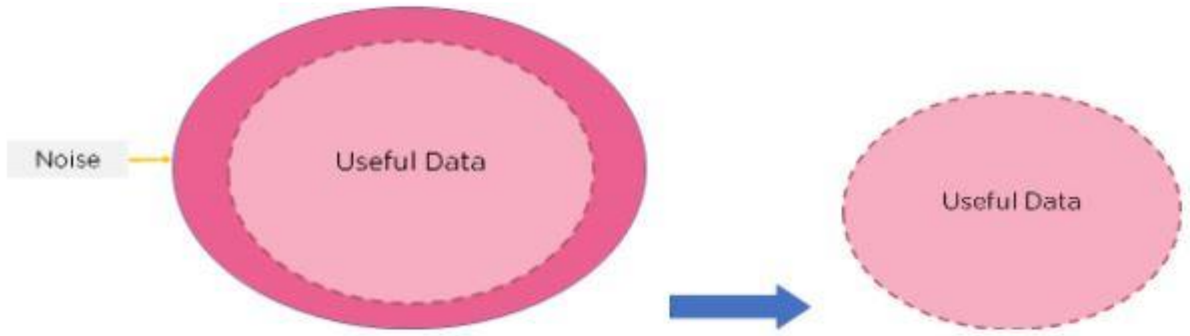
1. **Filter Method:** Based on the statistical measure of the relationship between the feature and the target variable. Features with a **high correlation** are selected.
2. **Wrapper Method:** In wrapper methods, we try to use a **subset of features** and train a model using them. Based on the inferences that we draw from the previous model, we decide **to add or remove features** from your subset.
3. **Embedded Method:** Embedded methods combine the qualities' of filter and wrapper methods. It's implemented by algorithms that have their own built-in feature selection methods.

Feature Selection

- To train an optimal model, we need to make sure that we use only the essential features.
- If we have too many features, the model can capture the unimportant patterns and learn from noise. The method of choosing **the important parameters** of our data is called Feature Selection.
- To train a model, we collect enormous quantities of data to help the machine learn better.
- If a good portion of the data collected is noise, then some of the columns of our dataset might not contribute significantly to the performance of our model.
- Further, having a lot of data can slow down the training process and cause the model to be slower. The model may also learn from this irrelevant data and be inaccurate.

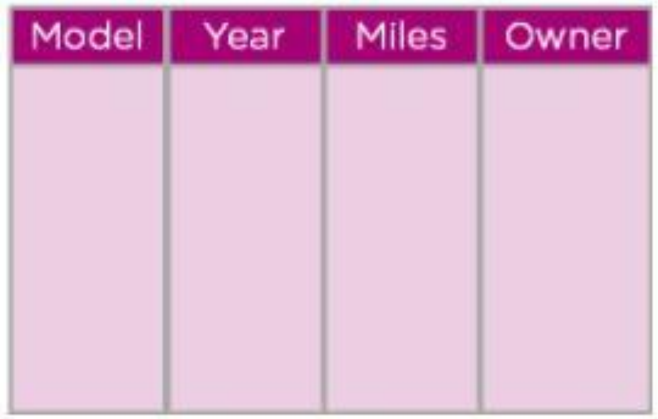
Feature Selection

- It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve.
- We do this by including or excluding important features without changing them. It helps in cutting down the noise in our data and reducing the size of our input data.

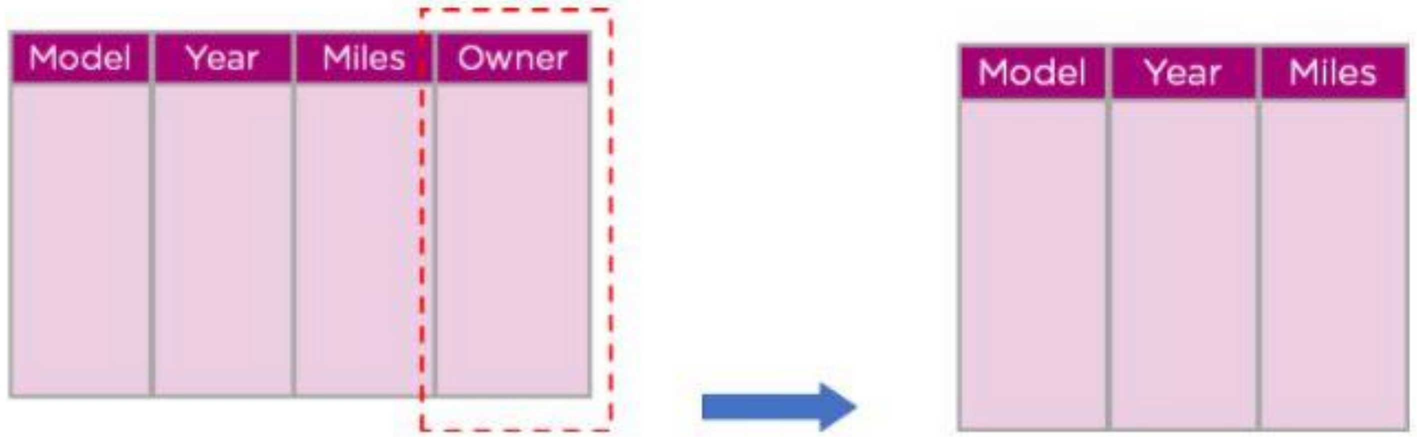


Example

- Consider a table which contains information on old cars. The model decides which cars must be crushed for spare parts.
- The model of the car, the year of manufacture, and the miles it has traveled are pretty important to find out if the car is old enough to be crushed or not.
- However, the name of the previous owner of the car does not decide if the car should be crushed or not.
- Hence we can drop the column.



Model	Year	Miles	Owner


Model
Year
Miles
Owner
Model
Year
Miles

Need for Feature Engineering in Machine Learning

- In machine learning, the performance of the model depends **on data pre-processing and data handling.**
- But if we create a model without pre-processing or data handling, then it may ***not give good accuracy***.
- Whereas, if we apply feature engineering on the same model, then the accuracy of the model is enhanced.
- Hence, feature engineering in machine learning improves the model's performance. Below are some points that explain the need for feature engineering:
  - Better features mean **flexibility.**
  - Better features mean **simpler models.**
  - Better features mean **better results.**

Steps in Feature Engineering

Data Preparation:

- In this step, raw data acquired from different resources are prepared to make it in a suitable format so that it can be used in the ML model.
- The data preparation may contain **cleaning of data, delivery, data augmentation, fusion, ingestion, or loading.**

Exploratory Analysis:

- Exploratory analysis or Exploratory data analysis (EDA) is an important step of features engineering, which is mainly used by data scientists.
- This step involves analysis, investing data set, and **summarization of the main characteristics** of data.
- Different data visualization techniques are used to better understand the manipulation of data sources
    - Understand data set variables and relationship between them
- to find the most **appropriate statistical technique for data analysis**-mean, std. deviation and Regression.
- **to select the best features** for the data.

Steps in Feature Engineering

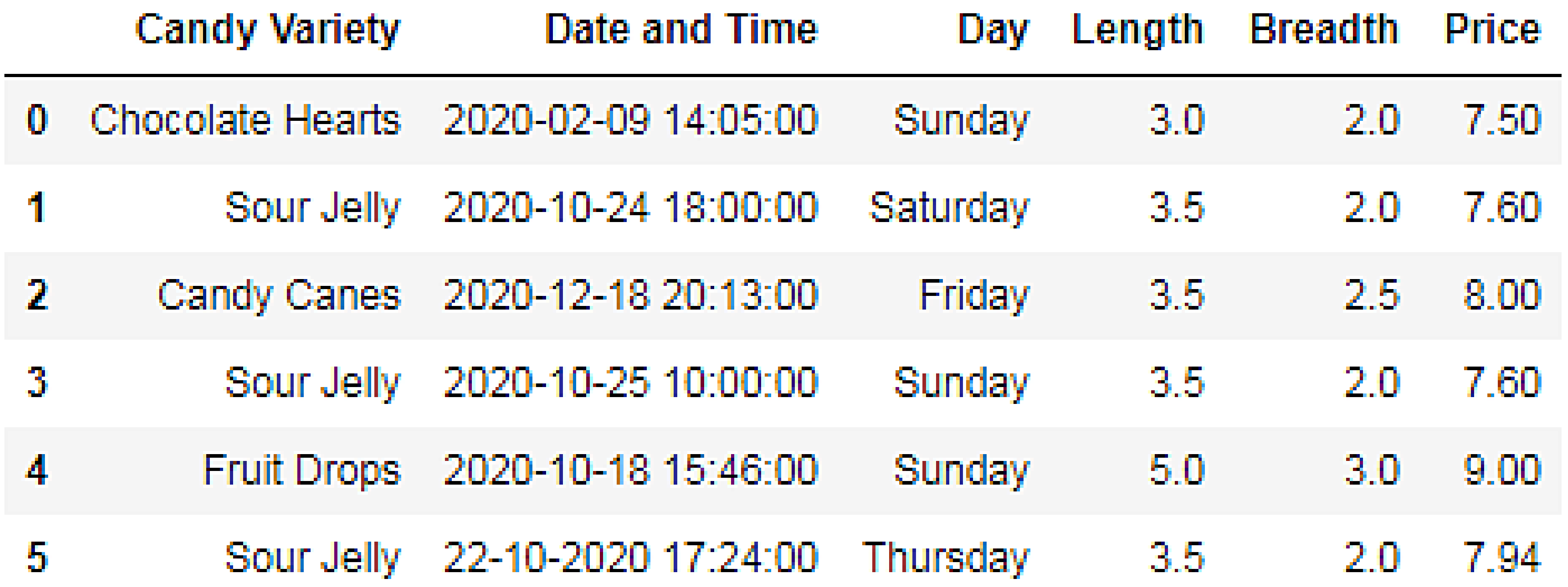
Benchmark:

- Benchmarking is a process of setting a standard baseline for accuracy to compare all the variables from this baseline.
- The benchmarking process is used to improve the predictability of the model and reduce the error rate.

Feature Engineering Techniques
1. Imputation

- Feature engineering deals with inappropriate data, missing values, human interruption, general errors, insufficient data sources, etc. Missing values within the dataset highly affect the performance of the algorithm, and to deal with them "Imputation" technique is used.
- **Imputation is responsible for handling irregularities within the dataset.**
- For example, removing the missing values from the complete row or complete column by a huge percentage of missing values. But at the same time, to maintain the data size, it is required to impute the missing data, which can be done as:
- For **numerical data imputation**, a default value can be imputed in a column, and **missing values** can be filled with **means or medians** of the columns.
- For **categorical data** imputation, missing values can be interchanged with the **maximum occurred value** in a column.

	Candy Variety	Date and Time	Day	Length	Breadth	Price
0	Chocolate Hearts	2020-02-09 14:05:00	Sunday	3.0	2.0	7.5
1	Sour Jelly	2020-10-24 18:00:00	Saturday	3.5	2.0	7.6
2	Candy Canes	2020-12-18 20:13:00	Friday	3.5	2.5	8.0
3	Sour Jelly	2020-10-25 10:00:00	Sunday	3.5	2.0	7.6
4	Fruit Drops	2020-10-18 15:46:00	Sunday	5.0	3.0	9.0
5	NaN	22-10-2020 17:24:00	Thursday	3.5	2.0	NaN



	Candy Variety	Date and Time	Day	Length	Breadth	Price
0	Chocolate Hearts	2020-02-09 14:05:00	Sunday	3.0	2.0	7.50
1	Sour Jelly	2020-10-24 18:00:00	Saturday	3.5	2.0	7.60
2	Candy Canes	2020-12-18 20:13:00	Friday	3.5	2.5	8.00
3	Sour Jelly	2020-10-25 10:00:00	Sunday	3.5	2.0	7.60
4	Fruit Drops	2020-10-18 15:46:00	Sunday	5.0	3.0	9.00
5	Sour Jelly	22-10-2020 17:24:00	Thursday	3.5	2.0	7.94

2. Handling Outliers

- Outliers are the deviated values or *data points* that are observed *too away from other data points* in such a way that they ***badly affect the performance of the model***.
- Outliers can be handled with this feature engineering technique. This technique first identifies the outliers and then remove them out.
- **Standard deviation** can be used to **identify** the outliers. For example, each value within a space has a definite to an average distance, but if a value is greater distant than a certain value, it can be considered as an outlier.
- **Z-score** can also be used to detect outliers.

3. Log transform

- Logarithm transformation or log transform is one of the commonly used mathematical techniques in machine learning.
- Log transform helps in handling the skewed data, and it makes the distribution more approximate to normal after transformation.
- It also reduces the effects of outliers on the data, as because of the normalization of magnitude differences, a model becomes much robust.

Note: Log transformation is only applicable for the positive values; else, it will give an error. To avoid this, we can add 1 to the data before transformation, which ensures transformation to be positive.

4. Binning

- In machine learning, overfitting is one of the main issues that degrade the performance of the model and which occurs due to a ***greater number of parameters and noisy data***. (model works fine with training data but poor performance on testing data)
- However, one of the popular techniques of feature engineering, **"binning",** can be used to *normalize the noisy data*. This process involves segmenting different features into bins.
- Binning is the process of transforming numerical variables into categorical counterparts.
- Binning improves accuracy of the predictive models by reducing the noise or non-linearity in the dataset. Finally, binning lets easy identification of outliers, invalid and missing values of numerical variables.

- #Numerical Binning Example
- Value   Bin
- 0-30   -> Low
- 31-70 -> Mid
- 71-100 -> High
- #Categorical Binning Example
- Value   Bin
- Spain -> Europe
- Italy -> Europe
- Chile -> South America
- Brazil -> South America

5. Feature Split

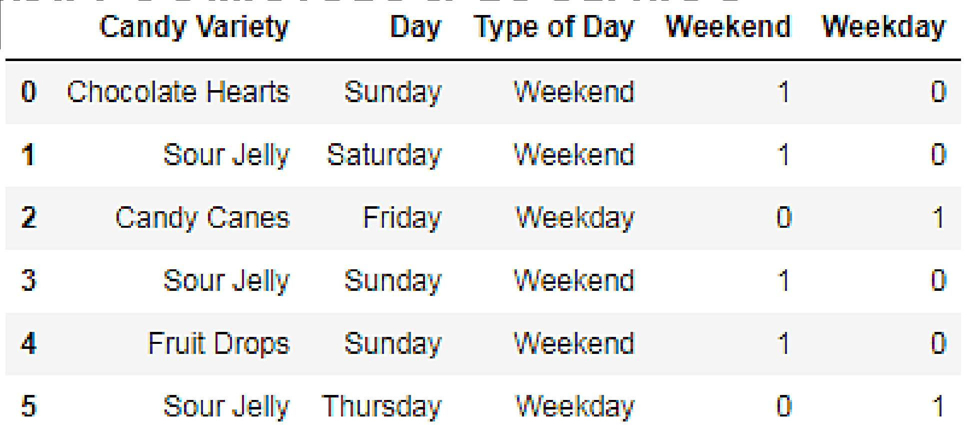
- As the name suggests, feature split is the process of splitting features intimately into two or more parts and performing to make new features.
- **This technique helps the algorithms to better understand and learn the patterns in the dataset.**
- The feature splitting process enables the new features to be clustered and binned, which results in extracting useful information and improving the performance of the data models.

- data.name
- 0 Luther N. Gonzalez
- 1   Charles M. Young
- 2   Terry Lawson
- 3   Kristen White
- 4   Thomas Logsdon
- #Extracting first names
- data.name.str.split(" ").map(lambda x: x[0])
- 0   Luther
- 1   Charles
- 2   Terry
- 3   Kristen
- 4   Thomas
- #Extracting last names
- data.name.str.split(" ").map(lambda x: x[-1])
- 0   Gonzalez
- 1   Young
- 2   Lawson
- 3   White
- 4   Logsdon

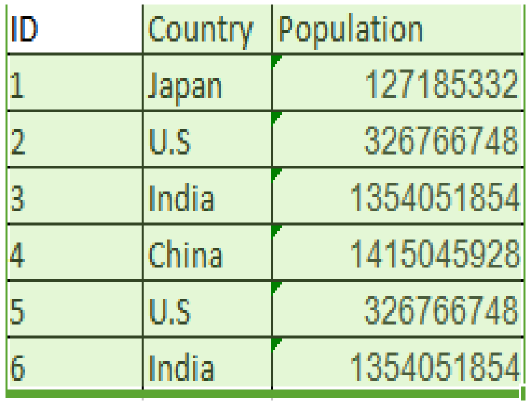
6. Encoding-Categorical variables
One hot encoding

- One hot encoding is the popular encoding technique in machine learning.
- It is a technique that ***converts the categorical data*** in a form so that they can be easily understood by machine learning algorithms and hence can make a good prediction.
- It enables group the of categorical data without losing any information.

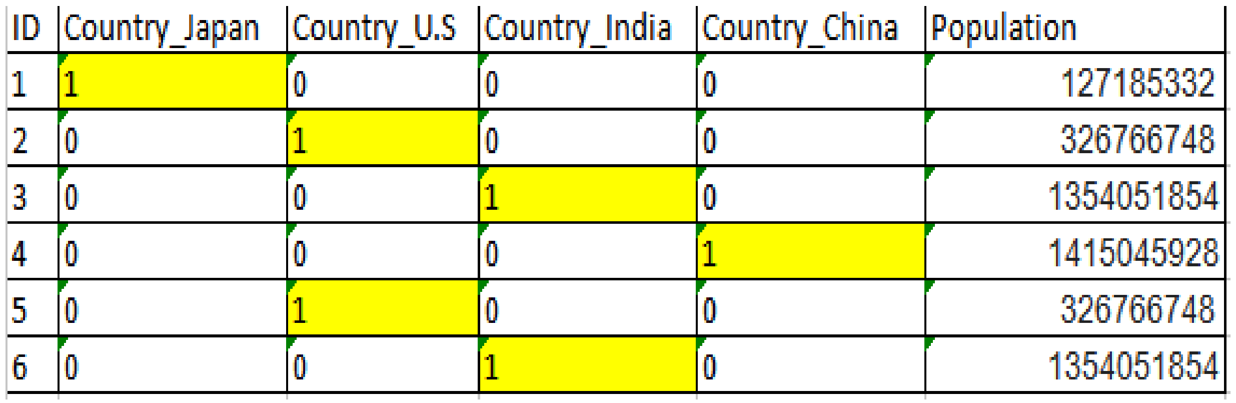
- Here, categorical values are converted into simple numerical 1’s and 0’s without the loss of information.
- As with other techniques, OHE has its own disadvantages and has to be used sparingly. It could result in a dramatic increase in the number of features and result in the creation of hig



	Candy Variety	Day	Type of Day	Weekend	Weekday
0	Chocolate Hearts	Sunday	Weekend	1	0
1	Sour Jelly	Saturday	Weekend	1	0
2	Candy Canes	Friday	Weekday	0	1
3	Sour Jelly	Sunday	Weekend	1	0
4	Fruit Drops	Sunday	Weekend	1	0
5	Sour Jelly	Thursday	Weekday	0	1

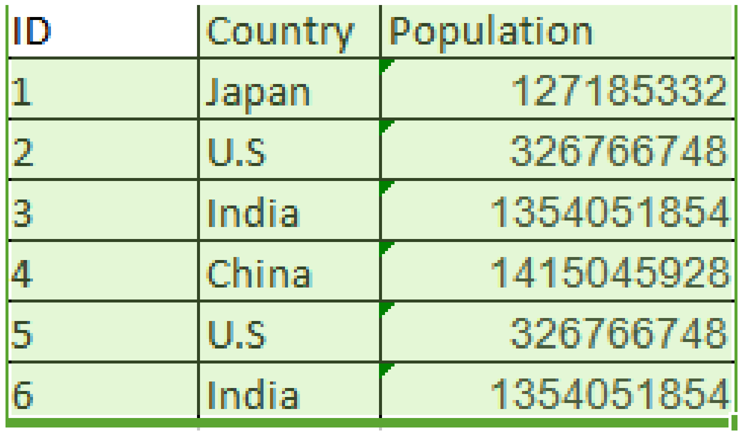


ID	Country	Population
1	Japan	127185332
2	U.S	326766748
3	India	1354051854
4	China	1415045928
5	U.S	326766748
6	India	1354051854

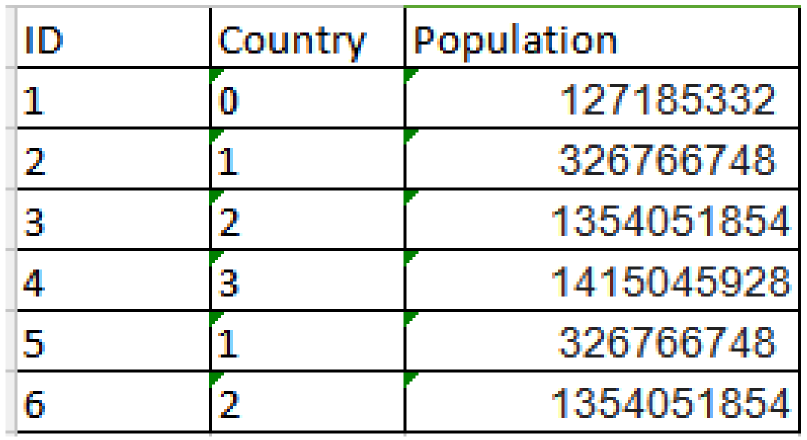


ID	Country_Japan	Country_U.S	Country_India	Country_China	Population
1	1	0	0	0	127185332
2	0	1	0	0	326766748
3	0	0	1	0	1354051854
4	0	0	0	1	1415045928
5	0	1	0	0	326766748
6	0	0	1	0	1354051854

- One-hot encoding can lead to the multiple number of columns .When the column contains the less number of variation in the categorical variables then we can label encode them such that variants of same kind gets encoded to the same number.



ID	Country	Population
1	Japan	127185332
2	U.S	326766748
3	India	1354051854
4	China	1415045928
5	U.S	326766748
6	India	1354051854



ID	Country	Population
1	0	127185332
2	1	326766748
3	2	1354051854
4	3	1415045928
5	1	326766748
6	2	1354051854

Curse of Dimensionality

- Curse of Dimensionality refers to a set of problems that arise when working with **high-dimensional data**.
- The dimension of a dataset corresponds to the **number of attributes/features** that exist in a dataset. A dataset with a large number of attributes, generally of the order of a hundred or more, is referred to as high dimensional data.
- Some of the difficulties that come with high dimensional data manifest during analyzing or visualizing the data to identify patterns, and some manifest while training machine learning models.
- The difficulties related to training machine learning models due to high dimensional data are referred to as the '***Curse of Dimensionality'.***
- In Machine Learning, a marginal increase in dimensionality also requires a large increase in the volume in the data in order to maintain the same level of performance. The curse of dimensionality is the by-product of a phenomenon which appears with high-dimensional data.

Cross Validation

- Cross-validation is a technique for evaluating a machine learning model and testing its performance.
- CV is commonly used in applied ML tasks.
- It helps to compare and select an appropriate model for the specific predictive modeling problem.
- CV is easy to understand, and easy to implement.
- All of this makes cross-validation a powerful tool for selecting the best model for a specific task.

Cross-validation Algorithm

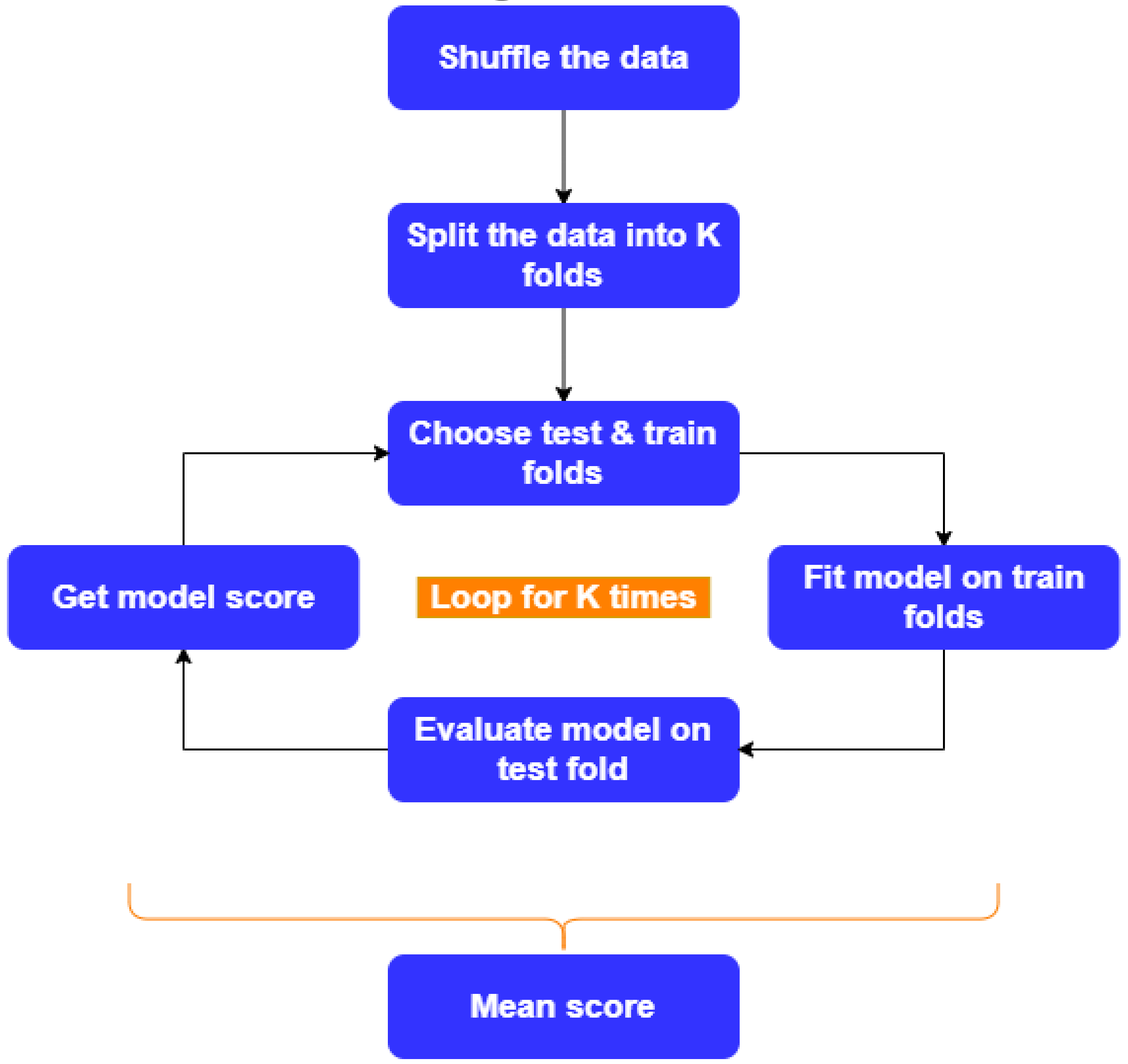
- Divide the dataset into two parts: one for training, the other for testing
- Train the model on the training set
- Validate the model on the test set
- Repeat 1-3 steps a couple of times. This number depends on the CV method that you are using

- Hold-out
- **K-folds**
- Leave-one-out
- Leave-p-out
- Stratified K-folds
- Repeated K-folds
- Nested K-folds
- Time series CV

Inner Working Process of Cross-Validation

- Shuffle the dataset in order to remove any kind of order
- Split the data into K number of folds. K= 5 or 10 will work for most of the cases.
- Now keep one fold for testing and remaining all the folds for training.
- Train(fit) the model on the train set and test(evaluate) it on the test set and note down the results for that split
- Now repeat this process for all the folds, every time choosing a separate fold as test data
- So for every iteration our model gets trained and tested on different sets of data
- At the end sum up the scores from each split and get the mean score

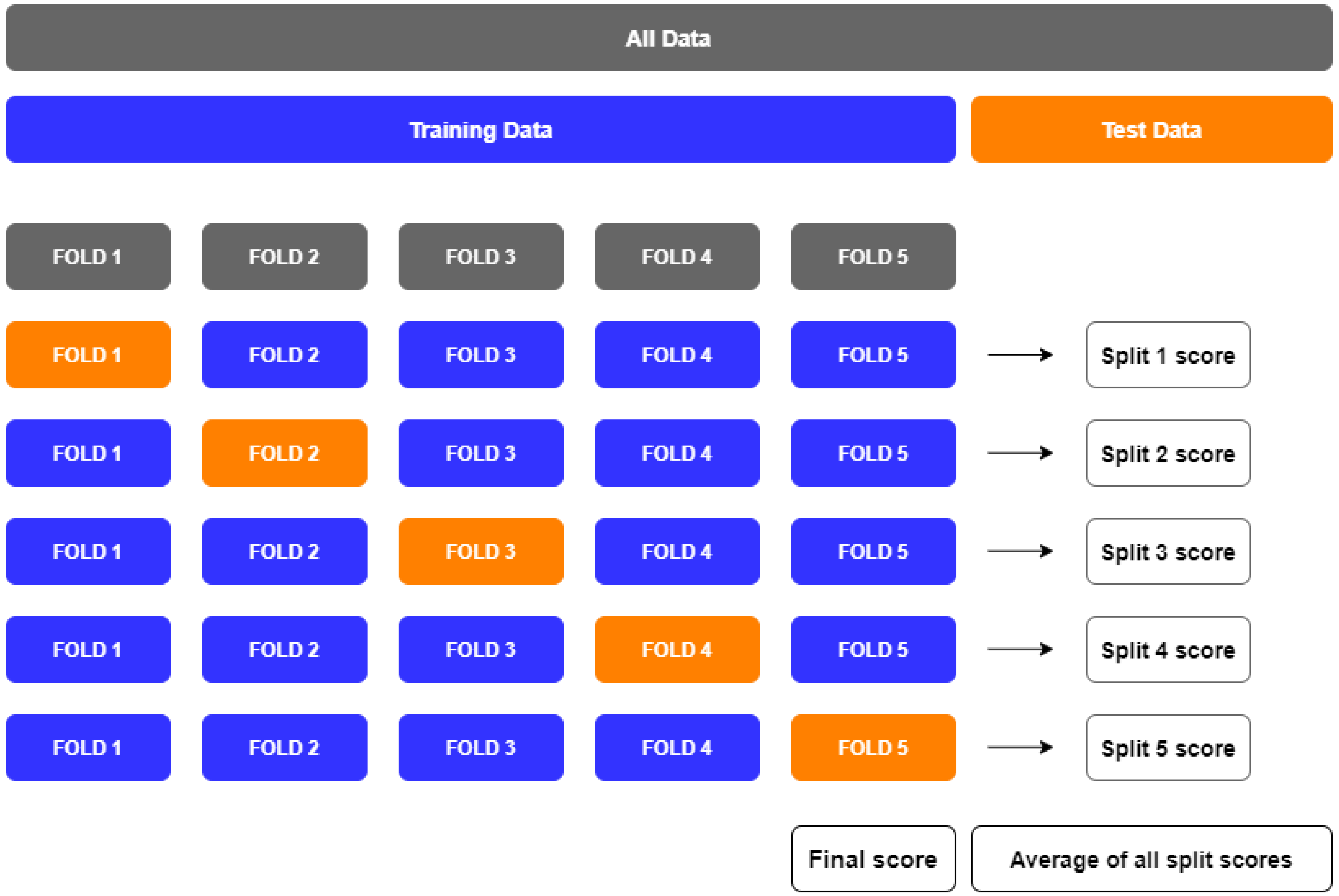
Inner Working of Cross Validation



K-fold cross validation

- In K Fold cross-validation input data is divided into 'K' number of folds, hence the name K Fold.
- Suppose we have divided data into 5 folds i.e. K=5.
- Now we have 5 sets of data to train and test our model.
- So the model will get trained and tested 5 times, but for every iteration we will use one fold as test data and rest all as training data.
- Note that for every iteration, data in training and test fold changes which adds to the effectiveness of this method.

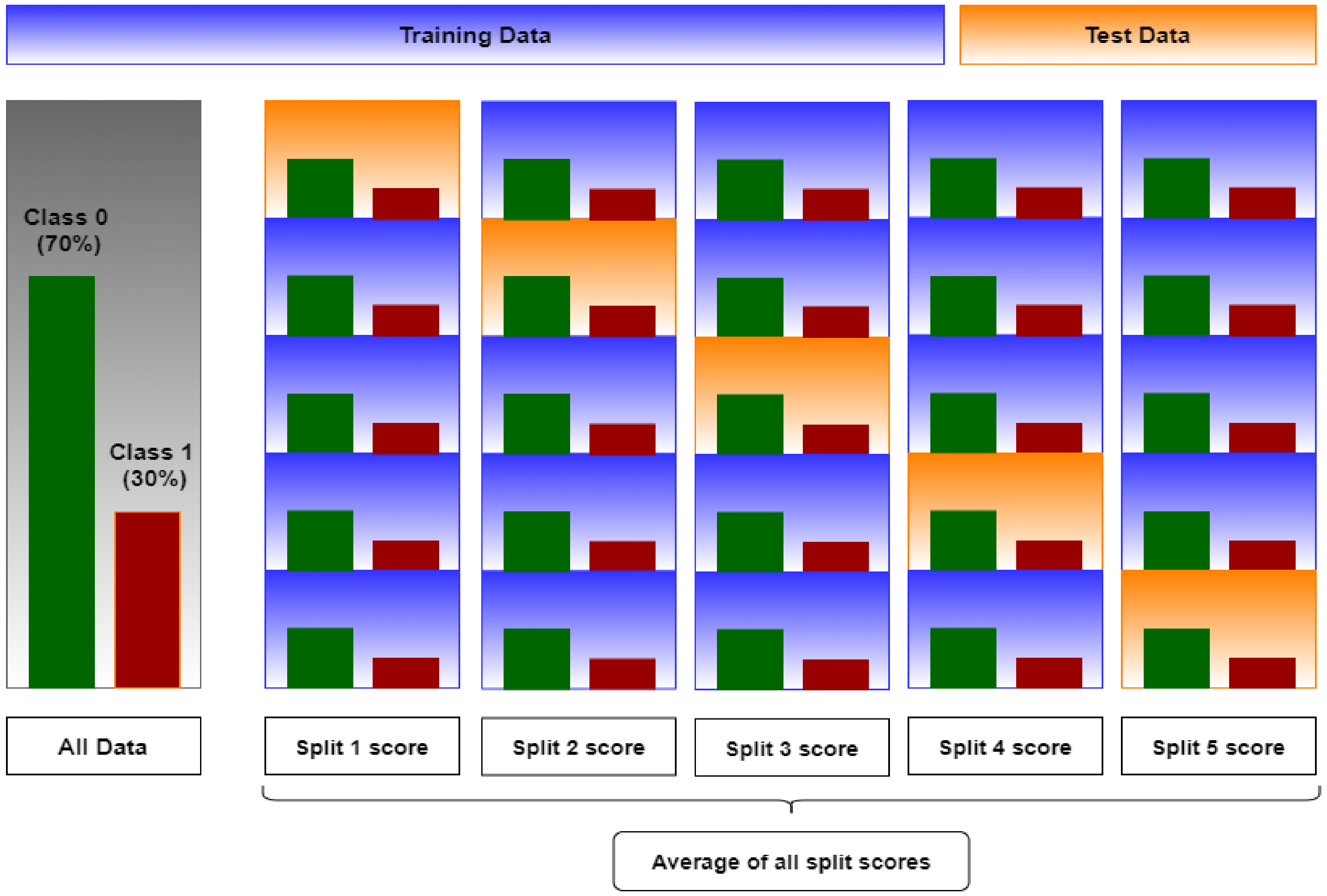
K-fold cross-validation



Stratified K Fold Cross Validation

- Stratified K Fold used when just **random shuffling and splitting the data is not sufficient.**
- In case of regression problem folds are selected so that the mean response value is approximately equal in all the folds.
- In the case of classification problems folds are selected to have the same proportion of class labels.
- Stratified K Fold is more useful in the case of classification problems, where it is very important to have the **same percentage of labels in every fold.**

Stratified K Fold Cross Validation



Advantages

- We end up using all the data for training and testing and this is very useful in case of small datasets
- It covers the variation of input data by validating the performance of the model on multiple folds
- Multiple folds also helps in case of unbalanced data
- Model performance analysis for every fold gives us more insights to fine tune the model
- Used for hyperparameter tuning

Train Test Split

- The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. The procedure involves taking a dataset and dividing it into two subsets.
- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.
- The procedure has one main configuration parameter, which is the size of the train and test sets. For example, a training set with the size of 0.67 (67 percent) means that the remainder percentage 0.33 (33 percent) is assigned to the test set.

- common split percentages include:
- Train: 80%, Test: 20%
- Train: 67%, Test: 33%
- Train: 50%, Test: 50%
- Code

```
# split into train test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

- **Repeatable Train-Test Splits**
- Rows are assigned to the train and test sets randomly.
- This is done to ensure that datasets are a representative sample (e.g. random sample) of the original dataset, which in turn, should be a representative sample of observations from the problem domain.
- This can be achieved by setting the "*random_state*" to an integer value

Code

```
# split into train test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=1)
```