Contents

[**100 DAYS OF MACHINE LEARNING** 3](#_heading=h.gjdgxs)

[**Day 1:** 3](#_heading=h.30j0zll)

[∙](#_heading=h.1fob9te) What is Machine Learning 3

[**Day 2:** 3](#_heading=h.3znysh7)

[∙](#_heading=h.2et92p0) AI VS ML VS DL 3

[**Day 3:** 4](#_heading=h.tyjcwt)

[∙](#_heading=h.3dy6vkm) Types Of Machine Learning 4

[**Day 4:** 4](#_heading=h.1t3h5sf)

[∙](#_heading=h.4d34og8) Batch/Offline VS Online Machine Learning 4

[**Day 5:** 4](#_heading=h.2s8eyo1)

[∙](#_heading=h.17dp8vu) 1)Online Machine Learning 4

[**Day 6:** 5](#_heading=h.3rdcrjn)

[∙](#_heading=h.26in1rg) Instance based VS Model Based Learning 5

[**Day 7:** 5](#_heading=h.lnxbz9)

[∙](#_heading=h.35nkun2) Challenges in Machine Learning (Problems in Machine Learning) 5

[**Day 8:** 6](#_heading=h.1ksv4uv)

[∙](#_heading=h.44sinio) Application of Machine Learning 6

[**Day 9:** 6](#_heading=h.2jxsxqh)

[∙](#_heading=h.z337ya) Machine Learning Development Life Cycle (MLDLC) 6

[**Day 10:** 7](#_heading=h.3j2qqm3)

[∙](#_heading=h.1y810tw) Data Engineer / Data Analyst / Data Scientist / ML Engineer 7

[**Day 11:** 7](#_heading=h.4i7ojhp)

[∙](#_heading=h.2xcytpi) 1)What are Tensors 7

[**Day 12:** 8](#_heading=h.1ci93xb)

[∙](#_heading=h.3whwml4) Installing Anaconda 8

[**Day 13:** 8](#_heading=h.2bn6wsx)

[∙](#_heading=h.qsh70q) End to End Toy Project 8

[**Day 14:** 8](#_heading=h.3as4poj)

[∙](#_heading=h.1pxezwc) Framing The Problem (example : Netflix) 8

[**Day 15:** 8](#_heading=h.49x2ik5)

[∙](#_heading=h.2p2csry) Working with CSV files 8

[**Day 16:** 9](#_heading=h.147n2zr)

[∙](#_heading=h.3o7alnk) Working with JSON and SQL 9

[**Day 17:** 9](#_heading=h.23ckvvd)

[∙](#_heading=h.ihv636) Fetching Data From API 9

[**Day 18:** 9](#_heading=h.32hioqz)

[∙](#_heading=h.1hmsyys) Web Scrapping 9

[**Day 19:** 9](#_heading=h.41mghml)

[∙](#_heading=h.2grqrue) Understanding Your Data 9

[**Day 20:** 9](#_heading=h.vx1227)

[∙](#_heading=h.3fwokq0) EDA using univariable Analysis 9

[**Day 21:** 10](#_heading=h.1v1yuxt)

[∙](#_heading=h.4f1mdlm) EDA Using BIVARIATE AND MULTIVARIATE ANALYSYS 10

[**Day 22:** 10](#_heading=h.2u6wntf)

[∙](#_heading=h.19c6y18) Pandas Profiling 10

[**Day 23:** 10](#_heading=h.3tbugp1)

[∙](#_heading=h.28h4qwu) What is Feature Engineering 10

[**Day 24:** 10](#_heading=h.nmf14n)

[∙](#_heading=h.37m2jsg) Feature Scalling 10

[**Day 25:** 11](#_heading=h.1mrcu09)

[∙](#_heading=h.46r0co2) Feature Scaling – Normalization 11

[**Day 26:** 12](#_heading=h.2lwamvv)

[●](#_heading=h.111kx3o) Encoding Categorical Data(Variables) 12

[**Day 27:** 12](#_heading=h.3l18frh)

[∙](#_heading=h.206ipza) One Hot Encoding 12

[**Day 28:** 13](#_heading=h.4k668n3)

[∙](#_heading=h.2zbgiuw) Column Transformer 13

[**Day 29:** 13](#_heading=h.1egqt2p)

[∙](#_heading=h.3ygebqi) Learning Pipelines A-Z 13

[**Day 30:** 13](#_heading=h.2dlolyb)

[∙](#_heading=h.sqyw64) Function Transformer | Log Transform | Reciprocal Transform | Square Root Transform 13

# **100 DAYS OF MACHINE LEARNING**

## **Day 1:**

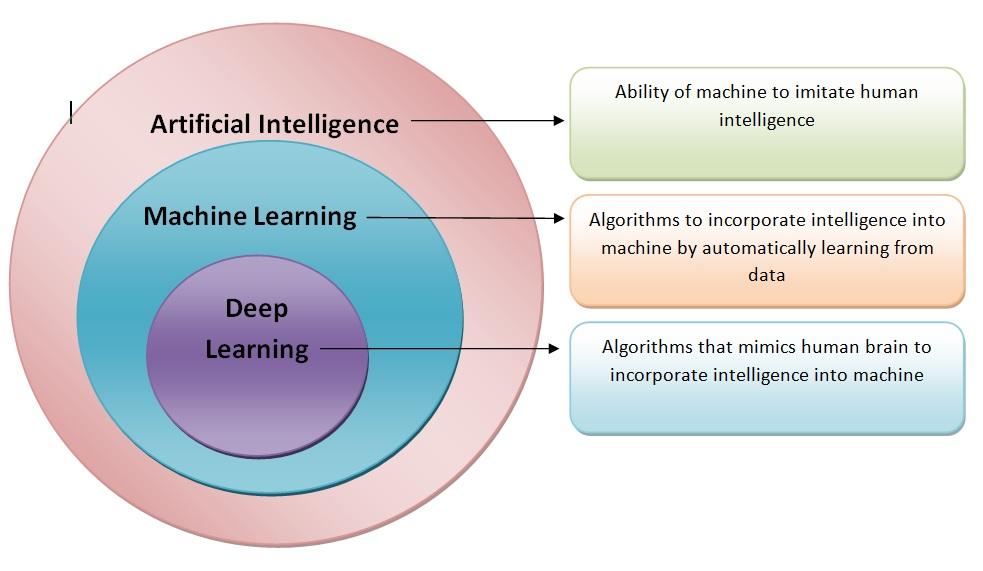
### **What is Machine Learning**

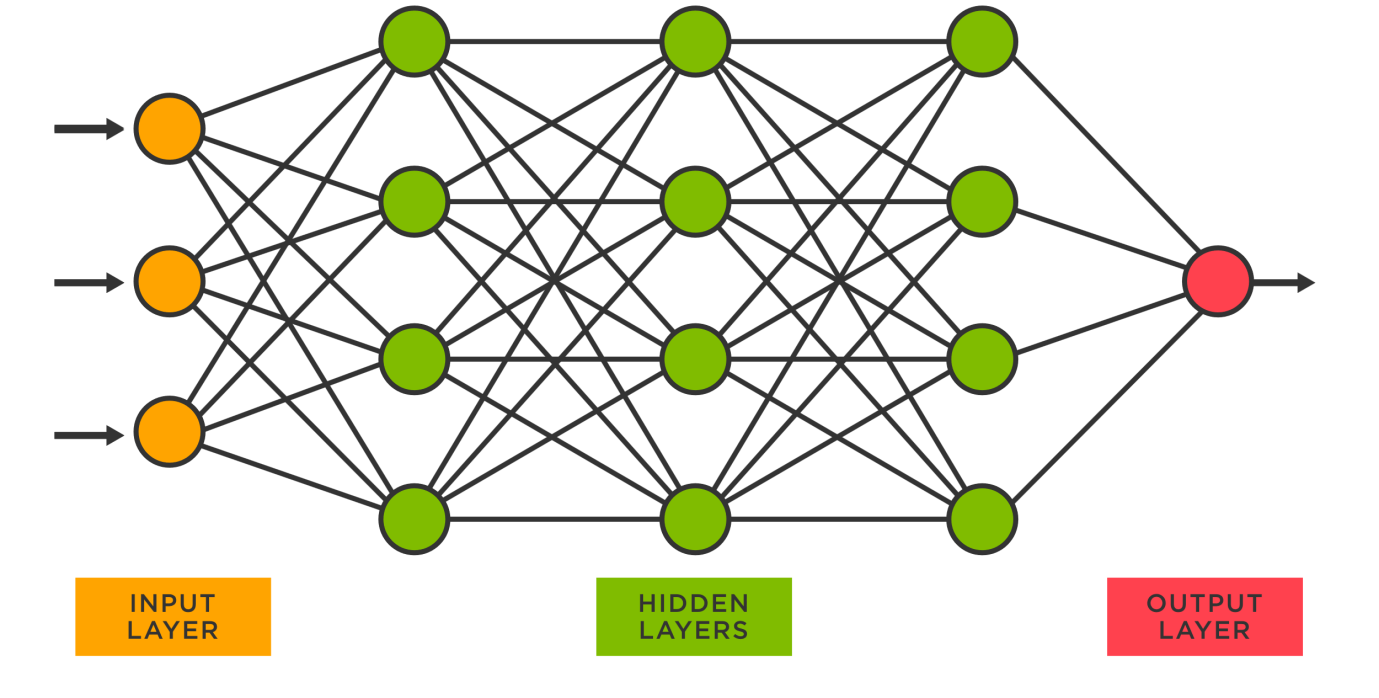
* [Resources](https://www.ibm.com/topics/machine-learning)
* [Day 1](https://drive.google.com/drive/folders/1cX8O-_yATTy3Xv_M1NIVuJlfZANdjaPy?usp=drive_link)

## **Day 2:**

### **AI VS ML VS DL**

* Pattern Recognition
* Fuzzy Logics (don’t know the features)
* Symbolic Logic is not applicable so ML introduced.
* Fully concern on DATA
* Why We Need Deep Learning
* Deep Learning is the Subset of Machine Learning
* Neural Network (inspired by biology neuron)
* Deep Learning is a Mathematical Model
* Machine Learning couldn’t solve some problems properly but deep Learning can Solve.
* [Recourses](https://www.geeksforgeeks.org/difference-between-artificial-intelligence-vs-machine-learning-vs-deep-learning/)
* [**DAY 2**](https://drive.google.com/drive/folders/1vBdlffbz6acIVRqM3hGuQYqiiNTED69Z?usp=drive_link)

****

****

## **Day 3:**

### **Types Of Machine Learning**

* Supervised Machine Learning
* 🡪Regression and Classification
* Some of the most popularly used supervised learning algorithms are:
* Linear Regression, Logistic Regression, Support Vector Machine, K Nearest Neighbor, Decision Tree, Random Forest, Naive Bayes
* Semi-supervised Learning (some datapoint labels) (mobile photo tags)
* Unsupervised Learning
* 🡪Clustering(grouping), Dimensionality Reduction (remove extra columns), Anomaly Detection (detect outliers) and Association (products association)
* Algorithms:
* K Means Clustering, Hierarchical Clustering, DBSCAN, Principal Component Analysis
* Reinforcement Learning (start learning from scratch) (self-driving car)
* Algorithms:
* Q-learning, Sarsa, Monte Carlo, Deep Q network
* [Types of Machine Learning | Simplilearn](https://www.simplilearn.com/tutorials/machine-learning-tutorial/types-of-machine-learning" \l ":~:text=There%20are%20primarily%20three%20types,%2C%20Unsupervised%2C%20and%20Reinforcement%20Learning.)
* [PPT](https://docs.google.com/presentation/d/1kFdrlDe_rcejEmew4oXwe-YYzk29qIoe/edit" \l "slide=id.p14)

## **Day 4:**

### **Batch/Offline VS Online Machine Learning**

* Batch Learning (conventional way of learning) (utilized all the data)
* Data🡪Model🡪test🡪server (drawback retrained model) (No incremental Learning)
* Tools for batch learning: Scikit learn, Tensor Flow, Porch etc
* [Online Vs. Offline Machine Learning | Qwak's Blog](https://www.qwak.com/post/online-vs-offline-machine-learning-whats-the-difference)

## **Day 5:**

### **Online Machine Learning**

* Train model in server
* Incremental Learning
* Start with some data 🡪model🡪test🡪server 🡨Dynamically trained with new data
* Examples: Chat bots, YouTube
* 2)When to use? problem nature change (Online learning), Cost effective (Online Learning), Faster Solution (Online Learning)
* 3)How to implement? Same as batch learning (using sklearn library)
* River library, Vowpal wabbit library
* 4)Set learning rate
* 5)Out of core Learning (data is so huge so could not load data in memory) (convert into small datasets and then one by one model and then evaluate)
* 6)Disadvantages: Tricky to use, Risky (depend on incoming data)
* Tools: Scikit multiflow, StreamDM
* Online Vs. Offline Machine Learning | Qwak's Blog

## **Day 6:**

### **Instance based VS Model Based Learning**

* Learning (Memorizing and Generalizing)
* Instance based learning (Raata based model)
* KNN
* Model Based model (Concept based model)
* Linear Regression etc
* [Blog](https://help.nextbrain.ai/en/article/4-instance-based-versus-model-based-learning-rhfr8i/" \l ":~:text=In%20summary%2C%20instance%2Dbased%20models,by%20creating%20a%20mathematical%20representation.)

## **Day 7:**

### **Challenges in Machine Learning (Problems in Machine Learning)**

* 1)Data Collection (fetch data from API, Web Scraping)
* 2)Insufficient data / Labelled Data
* 3)Non-Representative Data (Half gathering data)
* 4)Poor Quality Data
* 5)Irrelevant Features (No meaning in model)
* 6)Overfitting (if we train model on data then our model only knows about data if we give some data to model it didn’t show proper output) (instance-based model) (Problem)
* 7)Underfitting (opposite of Overfitting) (didn’t show good result in training data or test data)
* 8)Software Integration (To help the user and also a part of any software) (Important part is integrate the software) (Difficult)
* 9)Offline Learning / deployment (Difficult task)
* 10)Cost Involved
* [7 Major Challenges Faced By Machine Learning Professionals](https://www.geeksforgeeks.org/7-major-challenges-faced-by-machine-learning-professionals/)

## **Day 8:**

### **Application of Machine Learning**

* Chat bots, Recommendation system, Health etc
* Machine Learning helps in Business to Business
* 1)Retail -Amazon/Big Bazaar (Which Product Sales Most)
* 2)Banking and Finance (plans)
* 3)Transport (Driver app having maps)
* 4)Manufacturing -Tesla
* 5)Consumer Internet -Twitter
* [What are Machine Learning Applications? Top 10 Industry and Real-World Use Cases](https://emeritus.org/blog/machine-learning-what-are-machine-learning-applications/)
* [Applications of Machine Learning - Javatpoint](https://www.javatpoint.com/applications-of-machine-learning)

## **Day 9:**

### **Machine Learning Development Life Cycle (MLDLC)**

* Set of guidelines to develop the software of machine learning
* 1)Frame the Problem (Examine the problem and also know the cost of the problem and what the algorithm will be use to solve the problem)
* 2)Gathering Data
* 3)Data Preprocessing
* 4)Exploratory Data Analysis (Experiment on data and plots graph and analysis on each column)
* 5)Feature Engineering and Selection
* 6)Model Training, Evaluation and Selection
* 7)Model Deployment
* 8)Testing
* 9)Optimize
* [“](https://medium.com/@dancerworld60/the-machine-learning-development-life-cycle-mldc-a-comprehensive-guide-8f6ff35541f5" \l ":~:text=The%20Machine%20Learning%20Development%20Life%20Cycle%20(MLDC)%20is%20a%20framework,and%20scalable%20machine%20learning%20models)The Machine Learning Development Life Cycle (MLDC): A Comprehensive Guide” | by ajaymehta | Medium.
* [Life cycle of Machine Learning - Javatpoint](https://www.javatpoint.com/machine-learning-life-cycle)

## **Day 10:**

### Data Engineer / Data Analyst / Data Scientist / ML Engineer

* **1)Data Engineer:**
* Scrap data from the given sources.
* Handle database and data warehouses.
* Advance knowledge in DBMS.
* **2)Data Analyst:**
* Clean data and analyse data and make report.
* Creating data visualization
* Collaborating with teams
* Optimize the data.
* **3)Data Scientist:**
* A data scientist is someone who is better at statistical than any software engineer and better at software engineering than any statistician.
* Understand data and make something in future.
* Predictive Modelling.
* **4)ML Engineer:**
* Deploy model in the website.
* Scaling and optimize the model for production.
* Monitoring and maintenance of deployed models.
* [Data Analyst vs Data Engineer vs Data Scientist: Skills, Responsibilities, Salary](https://www.edureka.co/blog/data-analyst-vs-data-engineer-vs-data-scientist/)

## **Day 11:**

### 1)What are Tensors

* Tensor are basically a data structure.
* Container to store numbers.
* 2)OD Tensors/Scalar 🡪np.array(3)
* 3)1D Tensors/Vector 🡪np.array([1,2,3])
* 4)2D Tensors/Matrix 🡪np.array([[1,2,3],[4,5,6]])
* 5)ND Tensors🡪np.array([[[1,2,3],[4,5,6]], [[1,2,3],[4,5,6]]])
* [WTF is a Tensor?!? - KDnuggets](https://www.kdnuggets.com/2018/05/wtf-tensor.html)

## **Day 12:**

### [Installing Anaconda](https://www.anaconda.com/download)

* Jupyter Notebook
* Virtual Environment
* [Google Collaboratory](https://colab.research.google.com/)
* Kaggle

## **Day 13:**

### End to End Toy Project

* Steps:
* 0.Preprocess + EDA + Feature Selection
* 1.Extract input and output columns
* 2.Train test split
* 3.Scale the values
* 4.Train the model
* 5.Evaluate the model / model selection
* 6.Deploy the Model

## **Day 14:**

### Framing The Problem (example : Netflix)

* 1)Business Problem
* 2)Types of Problem
* 3)Current Solution
* 4)Metrics to measure
* 5)Online VS Batch
* 6)Check Assumption

## **Day 15:**

### Working with CSV files

* Gathering Data
* 1-CSV🡪initial journey (see code file)
* 2-JSON/SQL
* 3-Later learn fetch data through API
* 4-web scraping

## **Day 16:**

### Working with JSON and SQL

* See Code file from data
* Read Json file🡪read Json(‘’)
* Read data through URL just enter URL
* Working with SQL
* Read code from file

## **Day 17:**

### Fetching Data From API

* Work to connect two software’s to communicate each other.
* Learn about API’s

## **Day 18:**

### Web Scrapping

* Using beautifulsoup library
* Check code file for more

## **Day 19:**

### Understanding Your Data

* When you get data do following points
* 1)How big is the data? (df.shape)
* 2)How does the data look like? (df.head(),df.sample())
* 3)what is the data type of columns? (df.info())
* 4)Are there any missing values? (df.isnull().sum())
* 5)How does the data look mathematically? (df.describe())
* 6)Are there duplicate values? (df.duplicated.sum())
* 7)How is the correlation between columns? (df.corr)

## **Day 20:**

### EDA using univariable Analysis

* Single Variable Analysis
* Two types of data (Numerical or categorical)
* By using matplotlib and seaborn library we can analysis of singles columns.
* See code file for more.

## **Day 21:**

### EDA Using BIVARIATE AND MULTIVARIATE ANALYSYS

* First Good Practice of matplotlib and seaborn
* See code files from data.

## **Day 22:**

### Pandas Profiling

* This method all EDA work for you in the form of html document
* from pandas\_profiling import ProfileReport

prof = ProfileReport(df)

prof.to\_file(output\_file='output.html')

## **Day 23:**

### What is Feature Engineering

* Feature Engineering is the process of using domain knowledge to extract features from raw data. These features can be used to improve the performance of machine learning algorithms.
* 1)Feature Transformation (Missing values imputation, Handling categorical features, Outliers detection, feature scaling)
* 2)Feature Construction (make new feature from other features)
* 3)Feature selection (remove unnecessary features)
* 4)Feature Extraction

## **Day 24:**

### Feature Scalling

* Feature scalling is a technique to standarize the independent features present in the fixed range.
* We scale input features.
* Two Techniques or Types Standarization and Normalization.
* Formula of Standarization is Xi`=Xi-X~/sigma
* X~ denotes mean
* Sigma denotes standard devaition
* range value is -3 to +3
* if we apply standarization on data then the mean of data is equal to zero and standard devaion is equal to 1.
* Why we use Standarizaion.
* K Means (Use the Euclidean distance measure),K nearest Neighbours (Measure the distances),Principal Component analysis(Feature with maximum variance),Artificial neaural network (Apply gradient Descent),Gradient Descent.

## **Day 25:**

### Feature Scaling – Normalization

* Normalizaion is a techique often applied as part of data preparation for machine learning. The Goal of normalization is to change the value of numeric columns in the dataset to use a common scale,without distorting differences in the range of values or losing information.
* 1)MinMax Scaling (Mostly use)
* X`=Xi-Xmin/Xmax-Xmin
* Range is 0 to 1
* sckit learn class mixmaxscaler
* 2)Mean Normaliztion
* X`=Xi-Xmean/Xmax-Xmin
* No support in scikit learn
* 3)Max absolute
* X`=Xi/|Xmax|
* maxabsscaler class
* use when we have sparse data (zero zayda hoon)
* 4)Robust Scaling
* X`=Xi-Xmedian/IQR
* iQR → Inter quartile range
* if data has outliers this one is good.
* Normalization VS Standarization
* → if Feature scaling is required
* → Standard scaler most used
* → image processing main minmax scaler use hota hai.
* → if you sure that your data having minimum and maximum values then you use min max scaler.

## **Day 26:**

### Encoding Categorical Data(Variables)

* Categorical Data Two Types
* Nominal(state) and Ordinal(grades and evaluation)
* Ordinal Data (One hot encoding)
* for X values we apply ordinal encoding and for Y values we apply Sckit learn class is Label Encoder.
* Transform categorical column into numerical data.

## **Day 27:**

### One Hot Encoding

* Apply this method on nominal Data
* Multicolinearity
* Dummy Variable Trap
* Using Most Frequent Columns
* in pandas we do by using get\_dummies function
* Differnce between pandas get\_dummies function and sckit learn class one hot encoder

→ pandas don’t remember columns positions

→ Sckit learn Remember it.

## **Day 28:**

### Column Transformer

## **Day 29:**

### Learning Pipelines A-Z

* Pipelines chains together multiple steps so that the output of each step is used as to the next step.
* Pipelines make it easy to apply the same preprocessing to train the test.

## **Day 30:**

### Function Transformer | Log Transform | Reciprocal Transform | Square Root Transform

* Function Transformer:
* Log transform, reciprocal, square,custom
* How to find if data is normal?
* Sns.distplot, pd.skew() if answer is zero then data is normal,QQ plot
* QQ plot
* Log Transformer (right skewed data)
* Reciprocal Transform(1/x) (larger values small and small values larger)
* Square root
* Square (normally used for left skewed data)

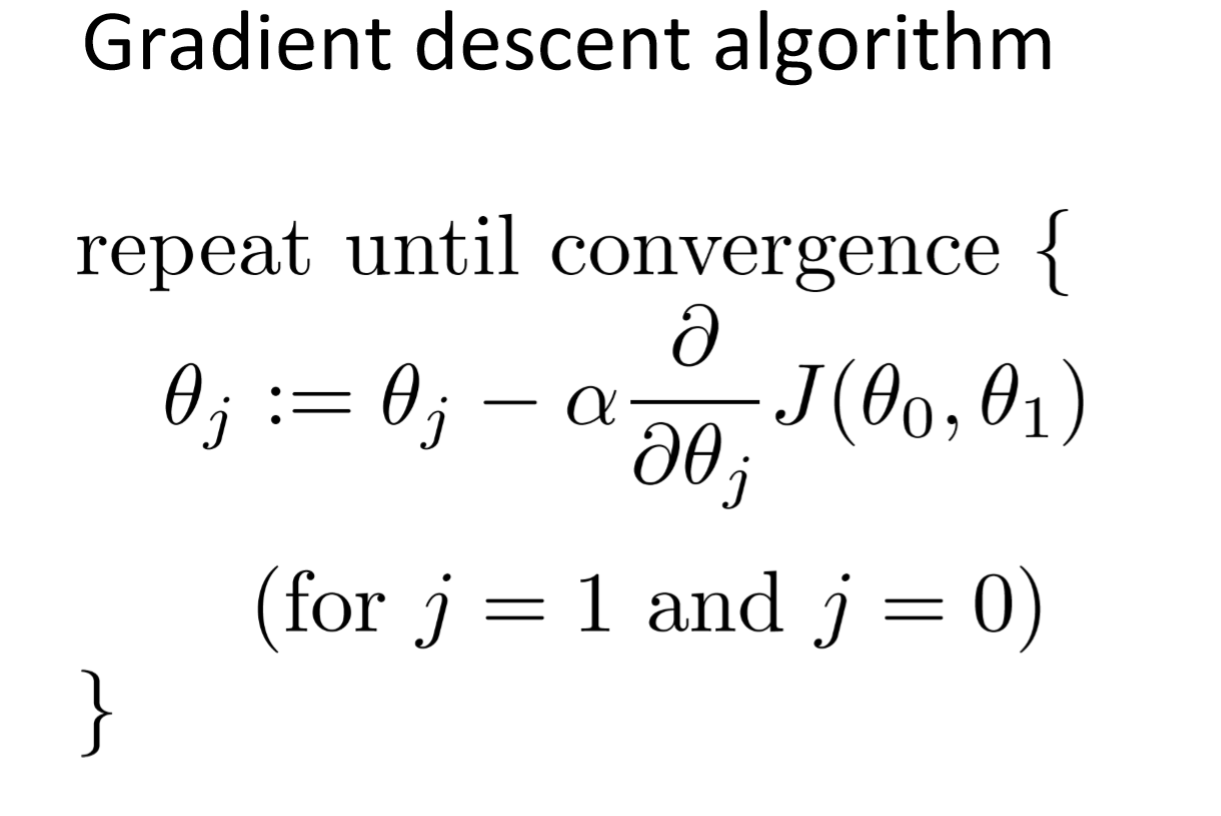
## **Day 31:**

### Power Transformer | Box - Cox Transform | Yeo - Johnson Transform

## **Day 51:**

### Gradient Descent From Scratch | End to End Gradient Descent | Gradient Descent Animation

* Iterative optimization algorithm to find local minimum of differential equation.
* Mathematical Formulation



* Cost Function

Measures performance of the model for given data.

Quantify error between predicted value and actual value

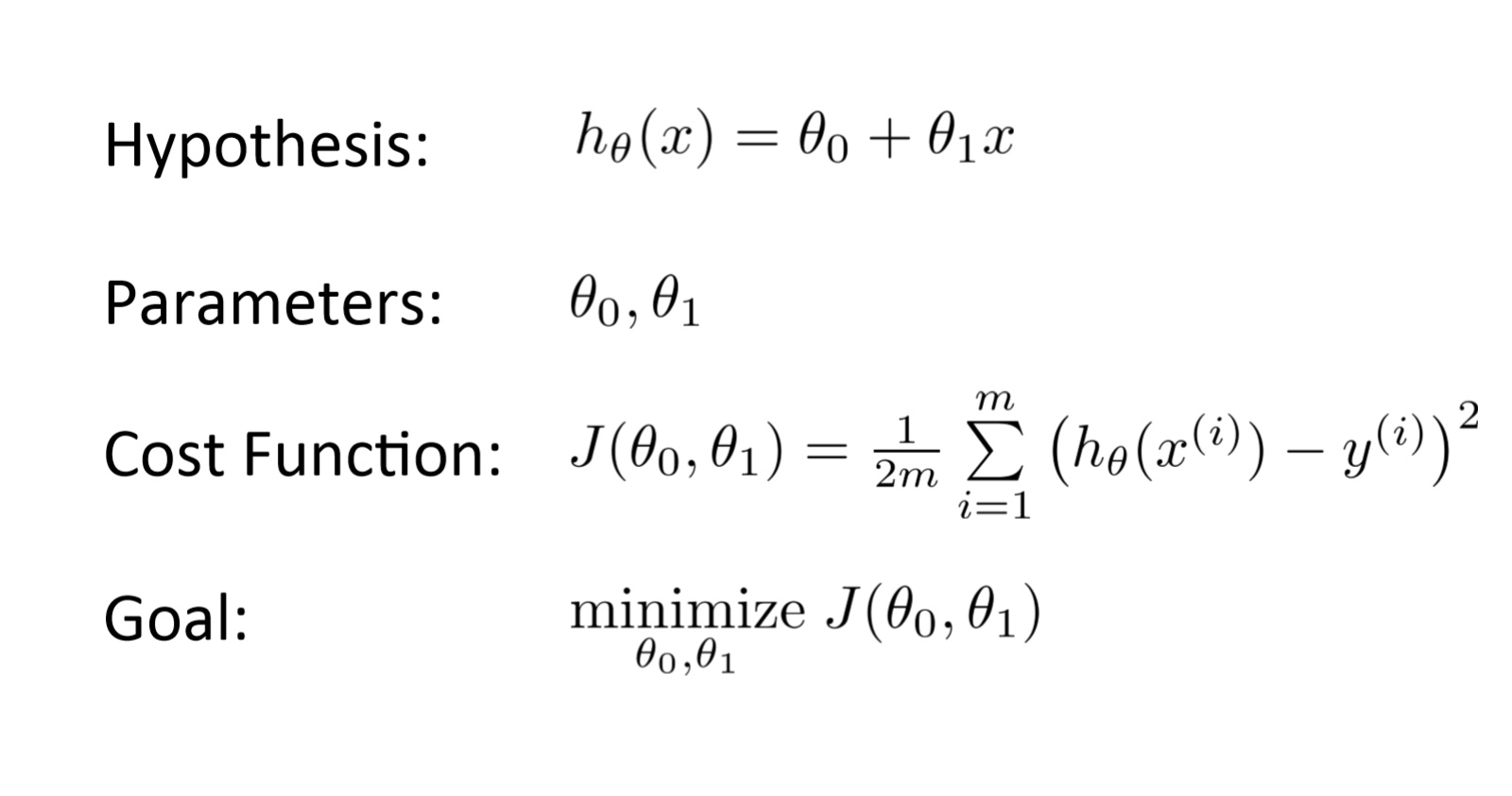
Goal to reduce the cost function.

find slope of line

if slope -ive the value of b increases

if slope +ive the value of b is decreases

\*When to Stop\* → (bnew-bold=0.0000001) close to zero →(limit iterations)



* Gradient Descent

Gradient descent is an optimization algorithm used in machine learning to minimize the cost function by iteratively adjusting parameters in the direction of the negative gradient, aiming to find the optimal set of parameters.

The cost function represents the discrepancy between the predicted output of the model and the actual output. The goal of gradient descent is to find the set of parameters that minimizes this discrepancy and improves the model’s performance.The Learning rate and hyperparameter determine step size in each iteration.

* Gradient Descent apply many machine learning algorithm Linear Regression,Logistic Regression, neural network and support vector machines.
* Example of Gradient Descent

Let’s say you are playing a game where the players are at the top of a mountain, and they are asked to reach the lowest point of the mountain. Additionally, they are blindfolded. So, what approach do you think would make you reach the lake?

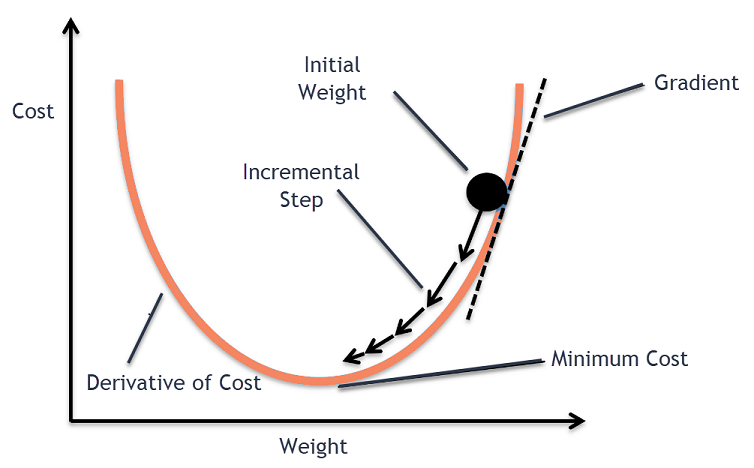
Take a moment to think about this before you read on.

The best way is to observe the ground and find where the land descends. From that position, take a step in the descending direction and iterate this process until we reach the lowest point.



To find the local minimum of a function using gradient descent, we must take steps proportional to the negative of the gradient (move away from the gradient) of the function at the current point. If we take steps proportional to the positive of the gradient (moving towards the gradient), we will approach a local maximum of the function, and the procedure is called Gradient Ascent.

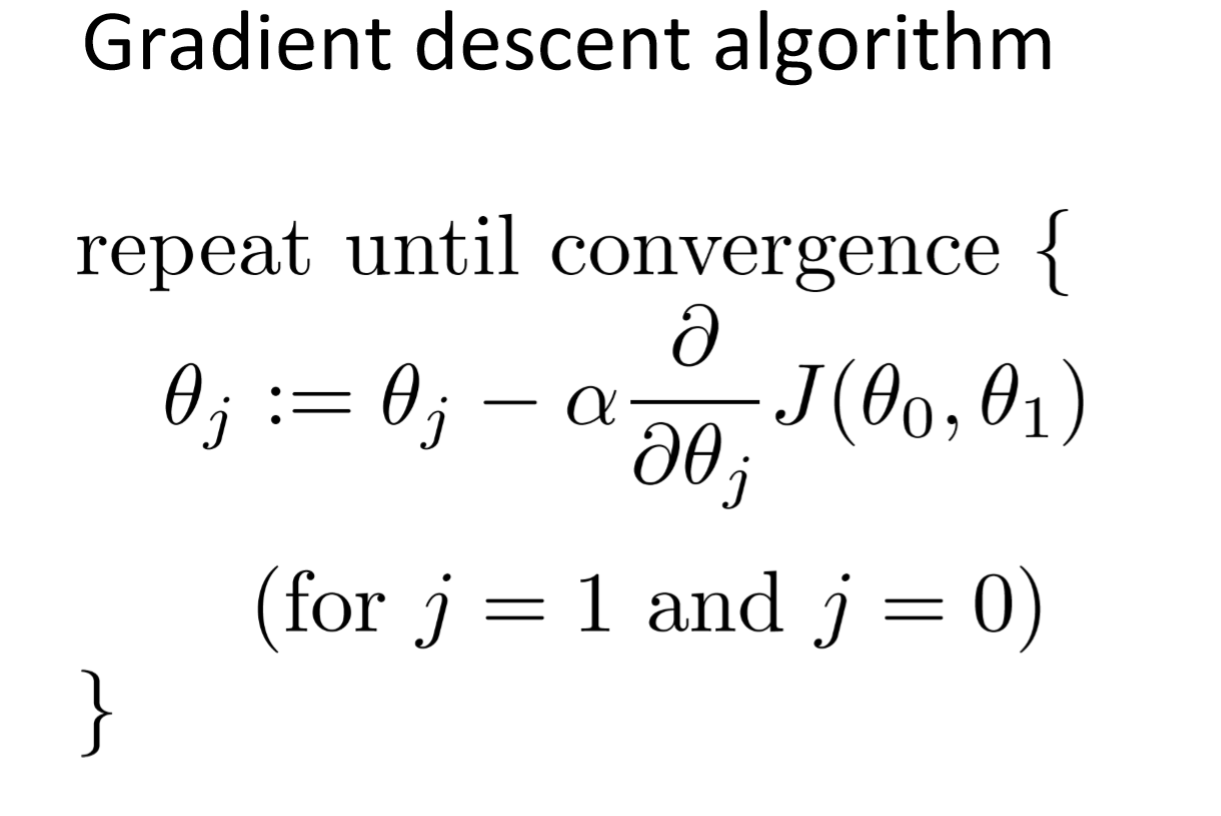
Gradient descent was originally proposed by CAUCHY in 1847. It is also known as steepest descent.



it performs two steps iteratively.

1-Compute the Gradient (the first order derivative of a function)

2-Make a step in the direction opposite to the gradient (opposite direction of slope increase from the current point aplatimes gradient at that point)



alpha is learning rate.

How Does Gradient Descent Work?

1. Gradient descent is an optimization algorithm used to minimize the cost function of a model.
2. The cost function measures how well the model fits the training data and is defined based on the difference between the predicted and actual values.
3. The gradient of the cost function is the derivative with respect to the model’s parameters and points in the direction of the steepest ascent.
4. The algorithm starts with an initial set of parameters and updates them in small steps to minimize the cost function.
5. In each iteration of the algorithm, the gradient of the cost function with respect to each parameter is computed.
6. The gradient tells us the direction of the steepest ascent, and by moving in the opposite direction, we can find the direction of the steepest descent.
7. The size of the step is controlled by the learning rate, which determines how quickly the algorithm moves towards the minimum.
8. The process is repeated until the cost function converges to a minimum, indicating that the model has reached the optimal set of parameters.
9. There are different variations of gradient descent, including batch gradient descent, stochastic gradient descent, and mini-batch gradient descent, each with its own advantages and limitations.
10. Efficient implementation of gradient descent is essential for achieving good performance in machine learning tasks. The choice of the learning rate and the number of iterations can significantly impact the performance of the algorithm.

* Types of Gradient Descent

Batch,mini batch,stochastic gradient descent

**Helpting Notes:**

1. Blog([Gradient Descent Algorithm in Machine Learning - Analytics Vidhya](https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/))
2. <https://drive.google.com/drive/folders/1h7xdnUFcACY8k0d-mS6yIgQYIILJSJwm?usp=drive_link>
3. Day-51 Resources(<https://drive.google.com/drive/folders/1o0nsSomuVXwDDyPxgtCmM_s9uhH25Z2H?usp=drive_link>)
4. <https://drive.google.com/file/d/1XhWVAXCHVTElA8VVUC-P3MZ7dm6r4XAu/view?usp=drive_link>

## **Day 59:**

### Logistic Regression

* dont apply logistic regression on non linear dataset.
* Perceptron Trick
* [Resources](https://drive.google.com/drive/folders/1gw34SEZKJUbTY5_fjdWlnDQAT6EB7RAt?usp=drive_link)

## **Day 61:**

### Decision Trees Geometric Intuition | Entropy | Gini impurity | Information Gain

* Nested if-else conditions
* Works on both Continuous variable (Regression) and Categorical Data (Classification).
* **Pseudo Code:**

1)Begin with your training dataset, which should have some features variables and classification or regression output.

2)Determine the “best feature” in the dataset to split the data on; more on how we define best feature” later

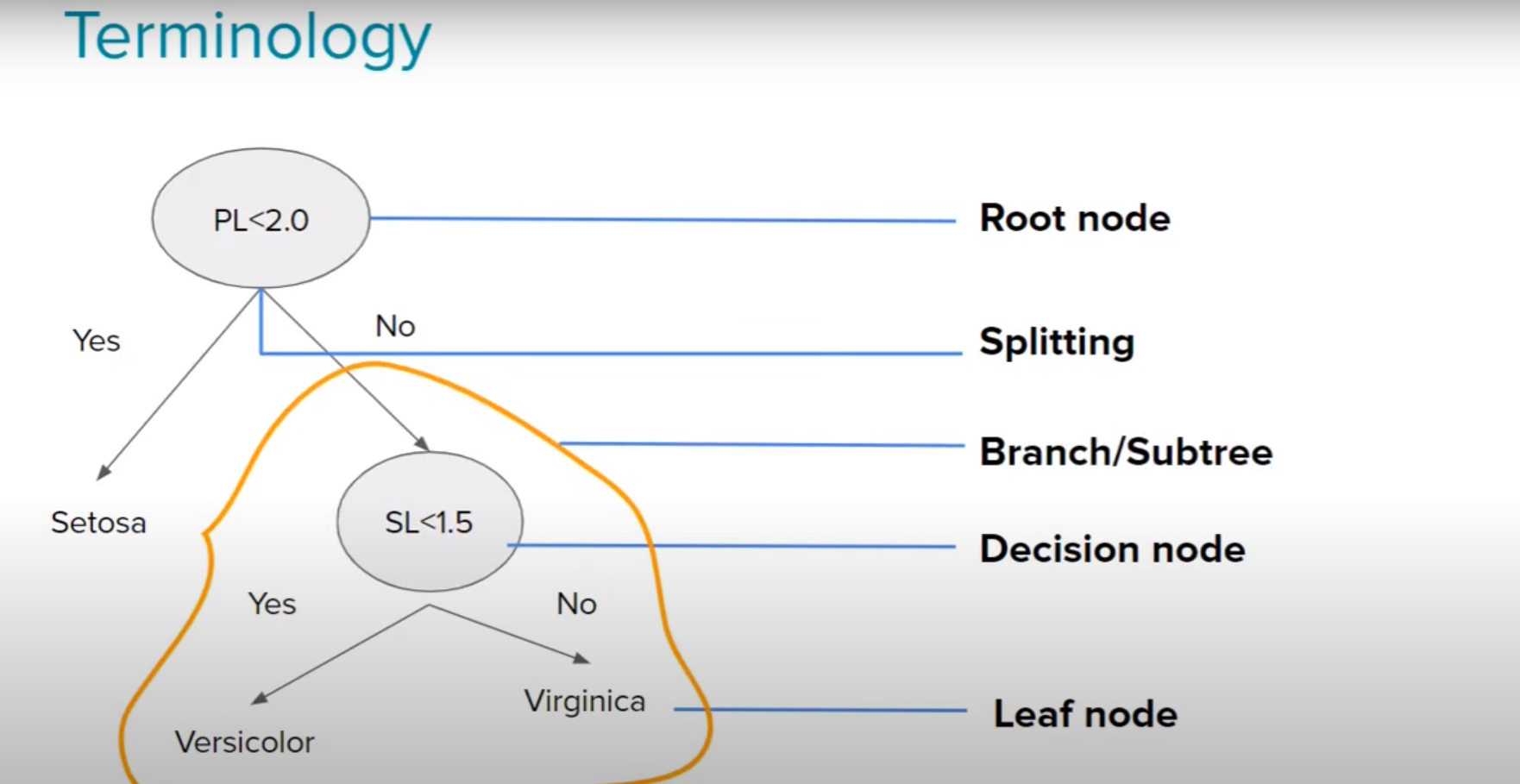
3)Split the data into subsets that the correct values for this best feature. This splitting basically defines a node on the tree i.e each node is a splitting point based on a certain feature form our data.

4)Recursively generate new tree nodes by using the subset of data creates from the step 3

* Conclusion:

🡪Programmatically speaking, Decision trees are nothing but a giant structure of nested if-else condition.

🡪Mathematically speaking, Decision trees use hyperplanes which run parallel any one of the axes to cut your coordinate system into hyper cuboids.



* How to decide which column should be considered as root node?
* How to select subsequent decision nodes?
* How to decide splitting criteria in case of numerical columns?
* **Advantages:**  
  🡪Intuitive and easy to understand

🡪Minimal data preparation is required

🡪The cost of using the tree for inference is logarithmic in the number of data points used to train the tree

* **Disadvantages:**

🡪Overfitting

🡪Prone to errors for imbalanced datasets.

* CART (Classification and Regression tree)
* **Entropy:**
  + In the most layman terms, Entropy is nothing but the measure of disorder. On you can also call it the measure of purity/impurity. Let’s see an example..
  + More Knowledge less entropy
  + How to calculate Entropy?
  + Mathematical Formula:
  + Where Pi is simply the frequentist probability of an element/class “I” in our data.
  + For example: if out data has only 2 class labels Yes and No.
  + E(D)=-PyesLog2(Pyes)-PnoLog2(Pno)
  + More uncertainty more is entropy
  + For a 2class problem the min entropy is 0 and the max is 1
  + For more than 2class the min entropy is 0 but the max can be greater than 1
  + Both log2 or loge can be used to calculate entropy.
  + In continuous variable ->Whichever is less peaked