

Introduction To Machine Learning

Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans.

Artificial Intelligence

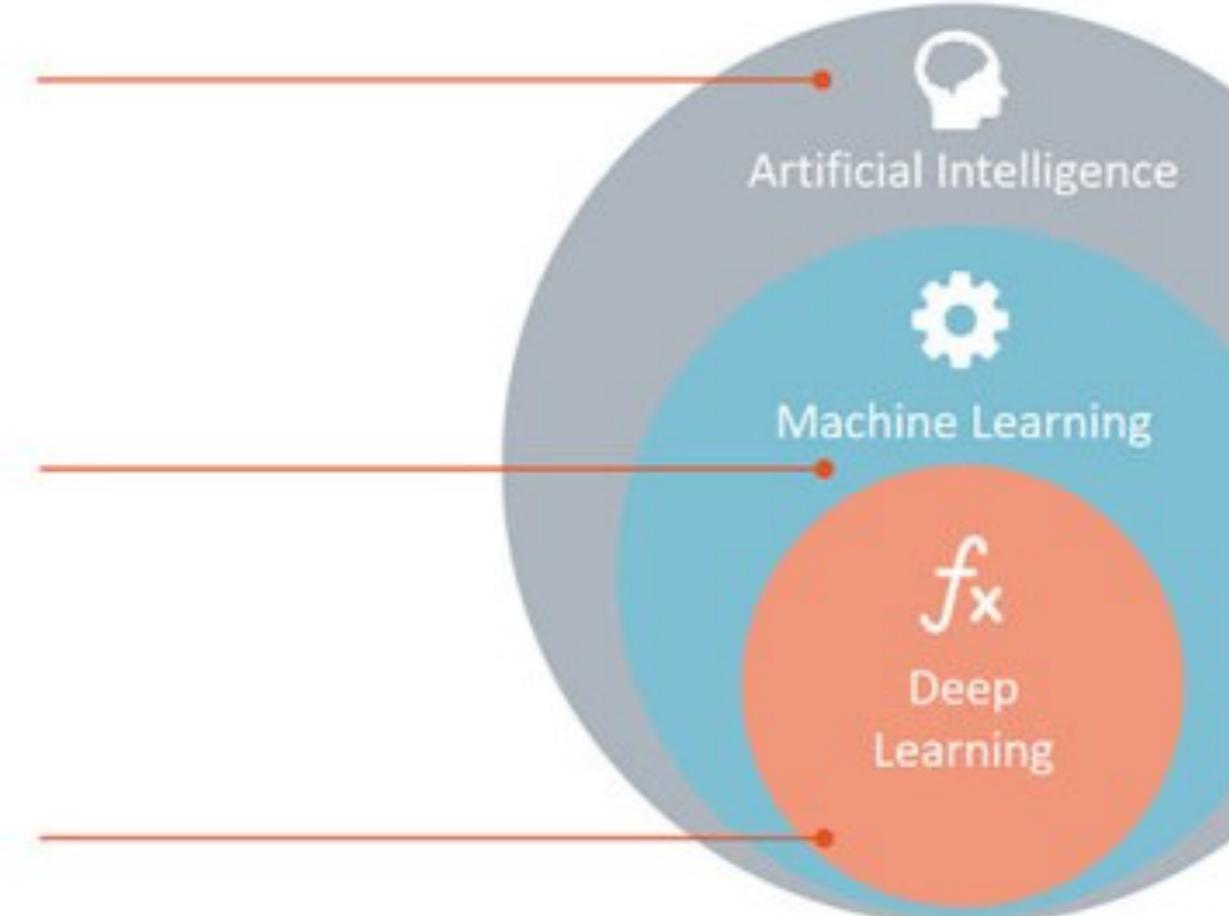
Any technique which enables computers to mimic human behavior.

Machine Learning

Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

Deep Learning

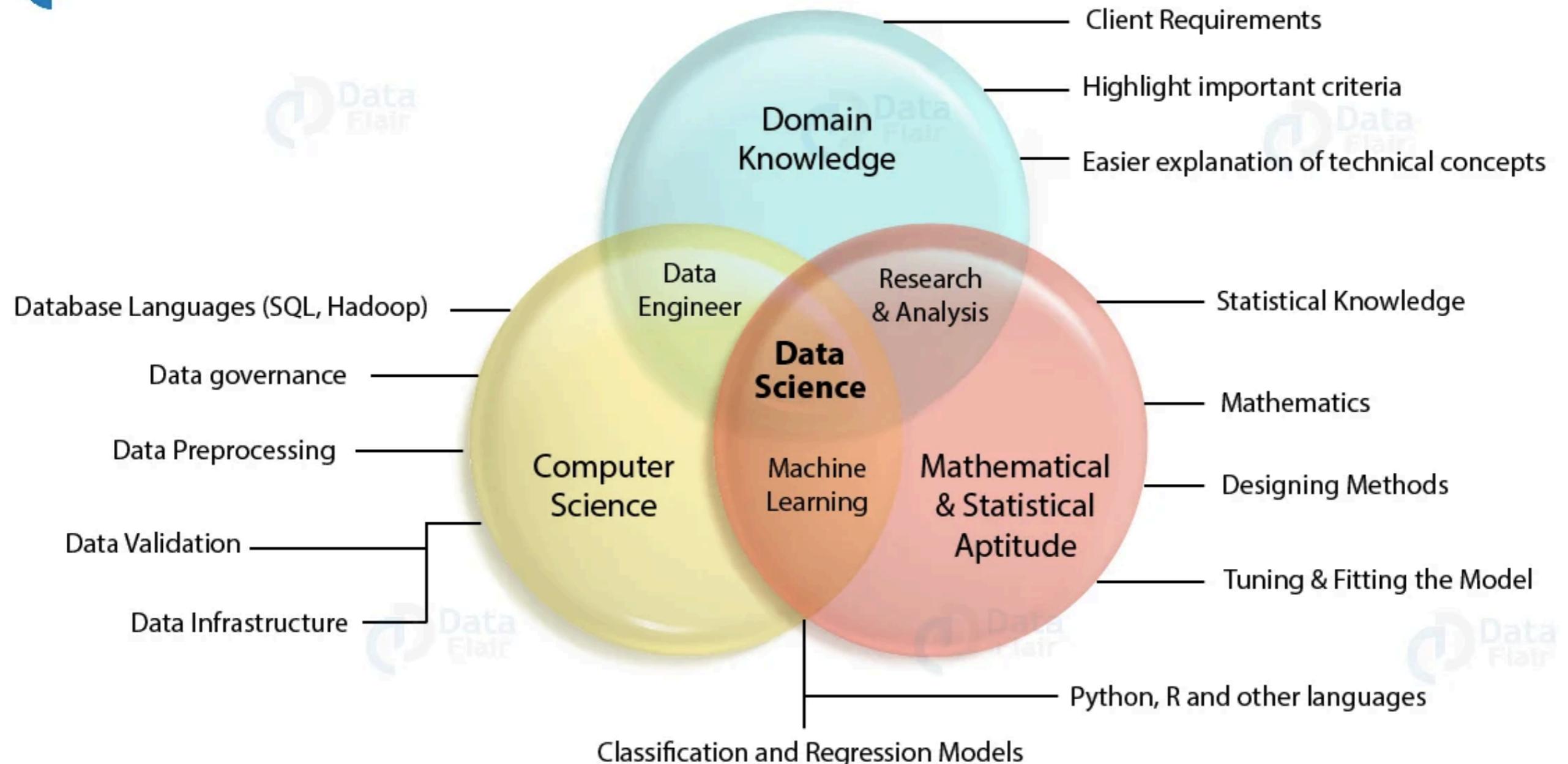
Subset of ML which make the computation of multi-layer neural networks feasible.



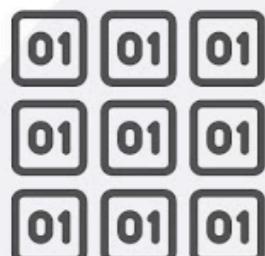
Data science is the study of data to extract meaningful insights for business.

It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyze large amounts of data.

This analysis helps data scientists to ask and answer questions like what happened, why it happened, what will happen, and what can be done with the results.



Structured data



Characteristics

Predefined data models
Easy to search
Text-based
Shows what's happening

Resides in

Relational databases
Data warehouses

Stored in

Rows and columns

Examples

Dates, phone numbers, social security numbers, customer names, transaction info

Unstructured data



Characteristics

No predefined data models
Difficult to search
Text, pdf, images, video
Shows the why

Resides in

Applications
Data warehouses and lakes

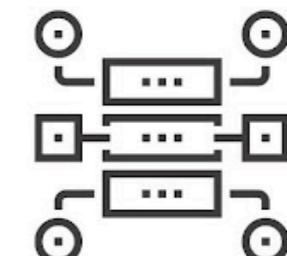
Stored in

Various forms

Examples

Documents, emails and messages, conversation transcripts, image files, open-ended survey answers

Semi-structured data



Characteristics

Loosely organized
Meta-level structure that can contain unstructured data

HTML, XML, JSON

Resides in

Relational databases
Tagged-text format

Stored in

Abstracts & figures

Examples

Server logs, tweets organized by hashtags, emails sorting by folders (inbox; sent; draft)

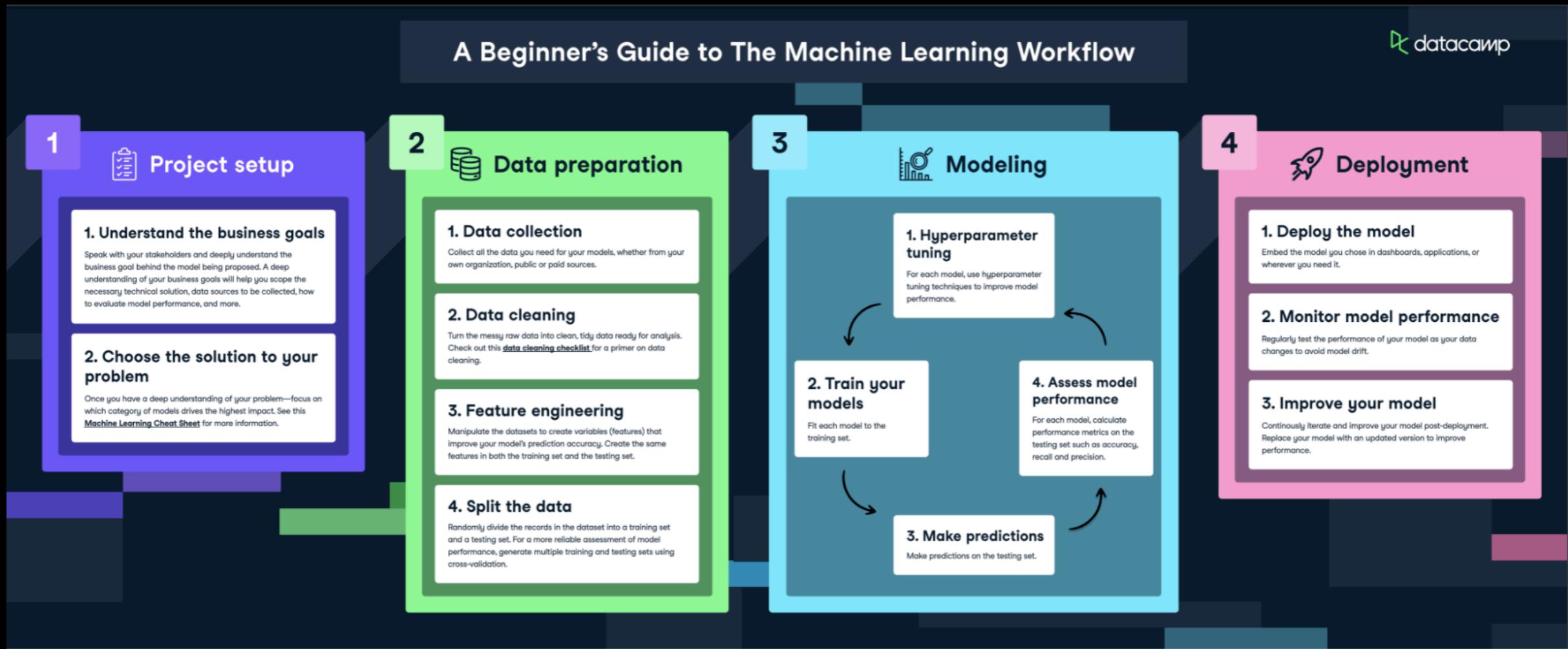
What is Machine Learning?

Machine Learning, often abbreviated as ML, is a subset of artificial intelligence (AI) that focuses on the development of computer algorithms that improve automatically through experience and by the use of data.

In simpler terms, machine learning enables computers to learn from data and make decisions or predictions without being explicitly programmed to do so.

- **Healthcare.** Algorithms are used to predict disease outbreaks, personalize patient treatment plans, and improve medical imaging accuracy.
- **Finance.** Machine learning is used for credit scoring, algorithmic trading, and fraud detection.
- **Retail.** Recommendation systems, supply chains, and customer service can all benefit from machine learning.

How Does Machine Learning Work?



Step 1: Data collection

Data is the lifeblood of machine learning - the quality and quantity of your data can directly impact your model's performance.

Data can be collected from various sources such as databases, text files, images, audio files, or even scraped from the web.

Step 2: Data preprocessing

Data preprocessing is a crucial step in the machine learning process.

It involves cleaning the data (removing duplicates, correcting errors), handling missing data (either by removing it or filling it in), and normalizing the data (scaling the data to a standard format).

Preprocessing improves the quality of your data and ensures that your machine learning model can interpret it correctly.

This step can significantly improve the accuracy of your model.

Step 3: Choosing the right model

Once the data is prepared, the next step is to choose a machine learning model.

There are many types of models to choose from, including linear regression, decision trees, and neural networks.

The choice of model depends on the nature of your data and the problem you're trying to solve.

Factors to consider when choosing a model include the size and type of your data, the complexity of the problem, and the computational resources available.

Step 4: Training the model

After choosing a model, the next step is to train it using the prepared data.

Training involves feeding the data into the model and allowing it to adjust its internal parameters to better predict the output.

During training, it's important to avoid overfitting (where the model performs well on the training data but poorly on new data) and underfitting (where the model performs poorly on both the training data and new data).

Step 5: Evaluating the model

Once the model is trained, it's important to evaluate its performance before deploying it. This involves testing the model on new data it hasn't seen during training.

Common metrics for evaluating a model's performance include accuracy (for classification problems), precision and recall (for binary classification problems), and mean squared error (for regression problems).

Step 6: Hyperparameter tuning and optimization

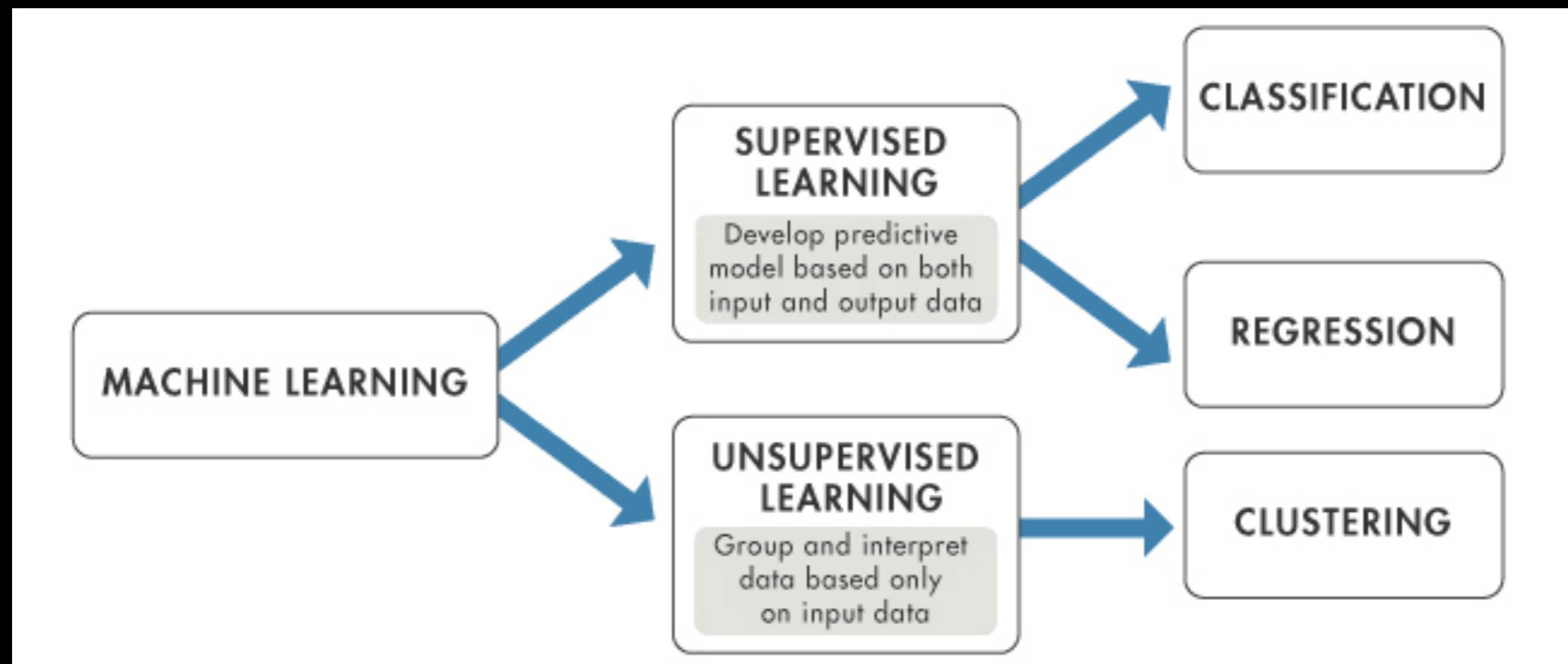
After evaluating the model, you may need to adjust its hyperparameters to improve its performance. This process is known as parameter tuning or hyperparameter optimization.

Step 7: Predictions and deployment

Once the model is trained and optimized, it's ready to make predictions on new data. This process involves feeding new data into the model and using the model's output for decision-making or further analysis.

Types of Machine Learning

Machine learning can be broadly classified into three types based on the nature of the learning system and the data available: supervised learning, unsupervised learning, and reinforcement learning. Let's delve into each of these:



Supervised learning

Supervised learning is the most common type of machine learning. In this approach, the model is trained on a labeled dataset.

In other words, the data is accompanied by a label that the model is trying to predict. This could be anything from a category label to a real-valued number.

The model learns a mapping between the input (features) and the output (label) during the training process. Once trained, the model can predict the output for new, unseen data.

Common examples of supervised learning algorithms include **linear regression** for regression problems and logistic regression, **decision trees**, and support vector machines for classification problems.

In practical terms, this could look like an image recognition process, wherein a dataset of images where each picture is labeled as "cat," "dog," etc., a supervised model can recognize and categorize new images accurately.

Supervised Machine Learning

4月28日(木) 24回目

Clean Data



Dataset

Unsupervised Machine Learning



Raw Data

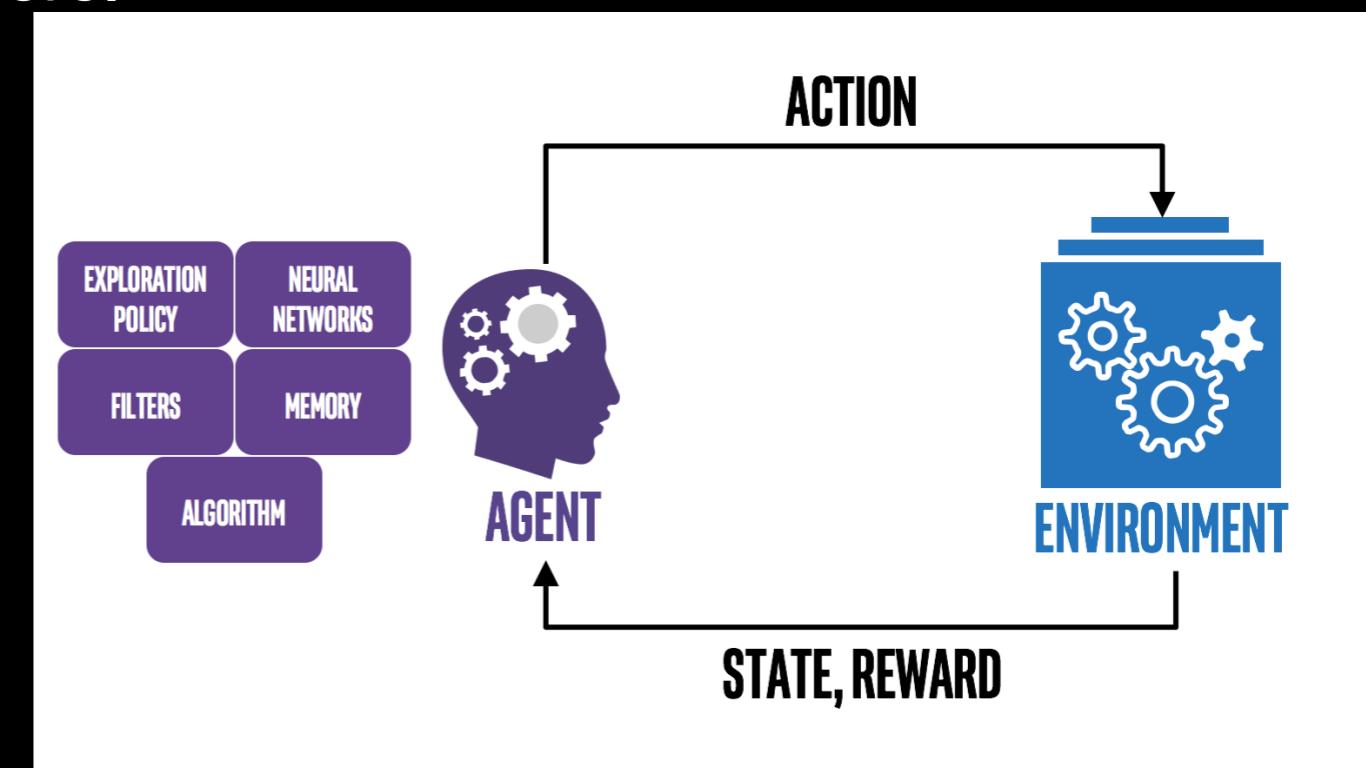
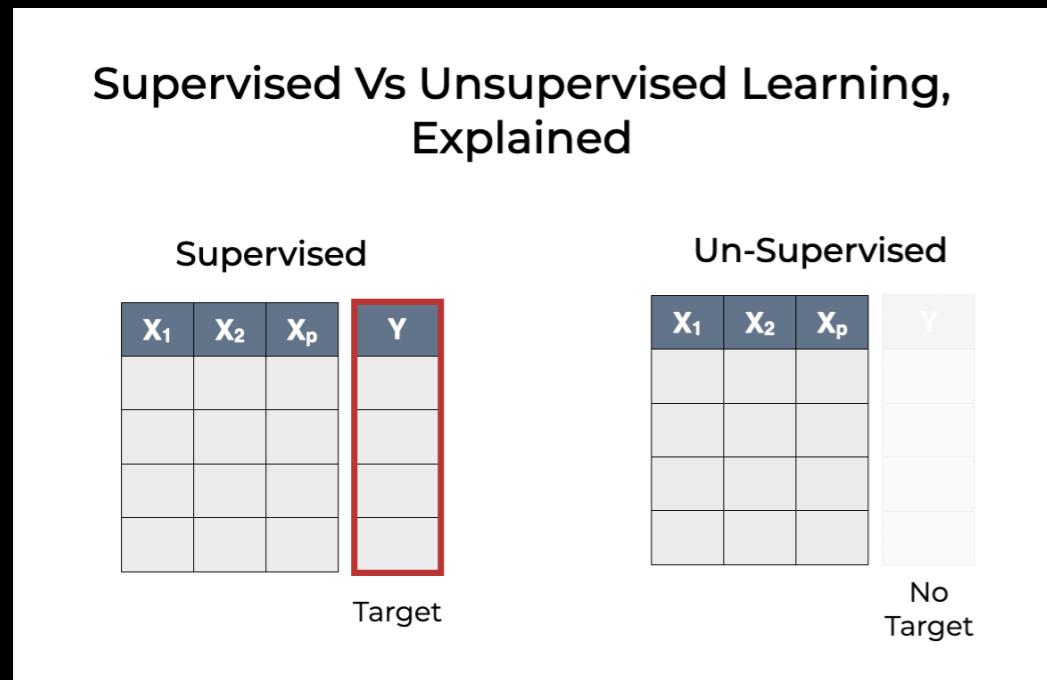
gifs.com

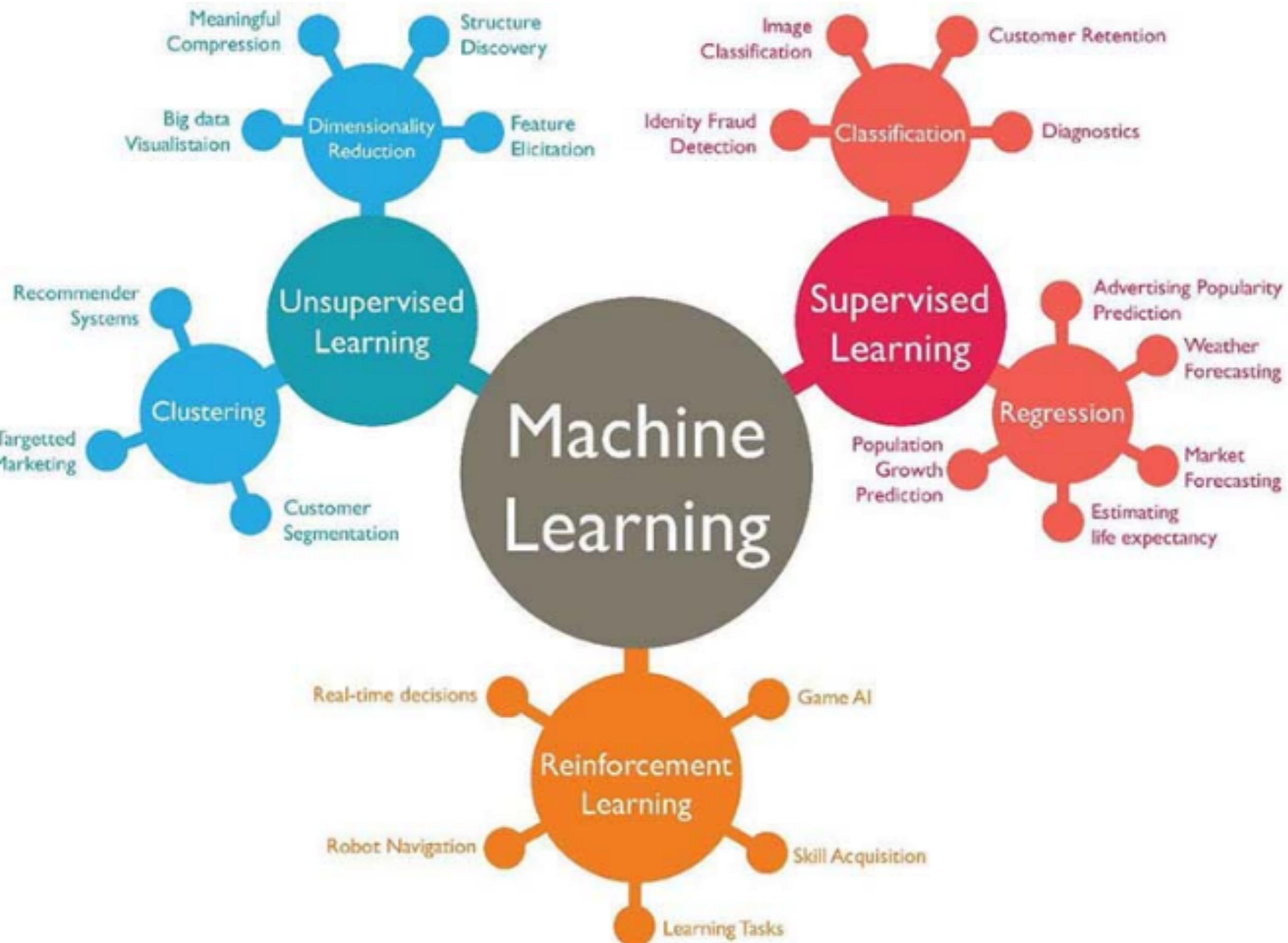
Reinforcement learning

Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with its environment. The agent is rewarded or penalized (with points) for the actions it takes, and its goal is to maximize the total reward.

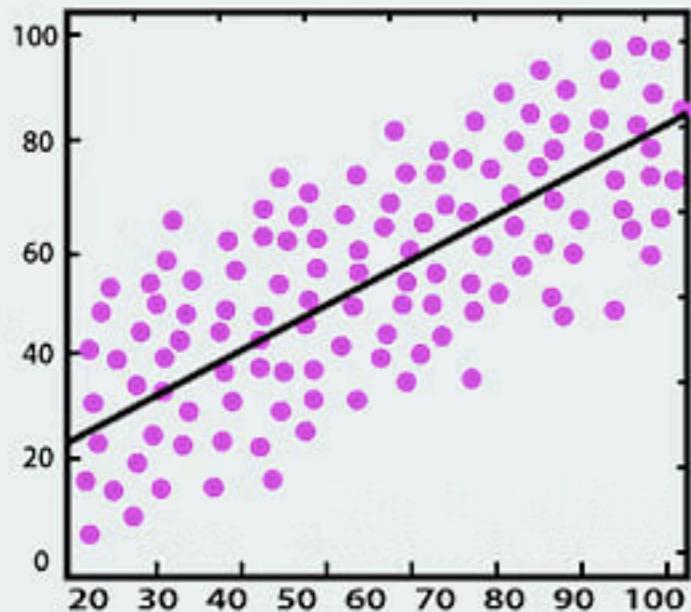
Unlike supervised and unsupervised learning, reinforcement learning is particularly suited to problems where the data is sequential, and the decision made at each step can affect future outcomes.

Common examples of reinforcement learning include game playing, robotics, resource management, and many more.



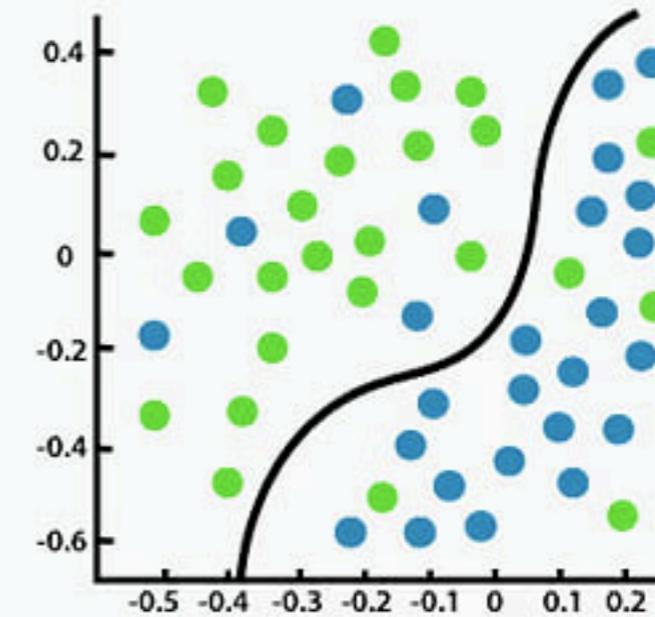


Supervised Learning: Regression vs. Classification



Regression

versus



Classification

Regression Algorithms

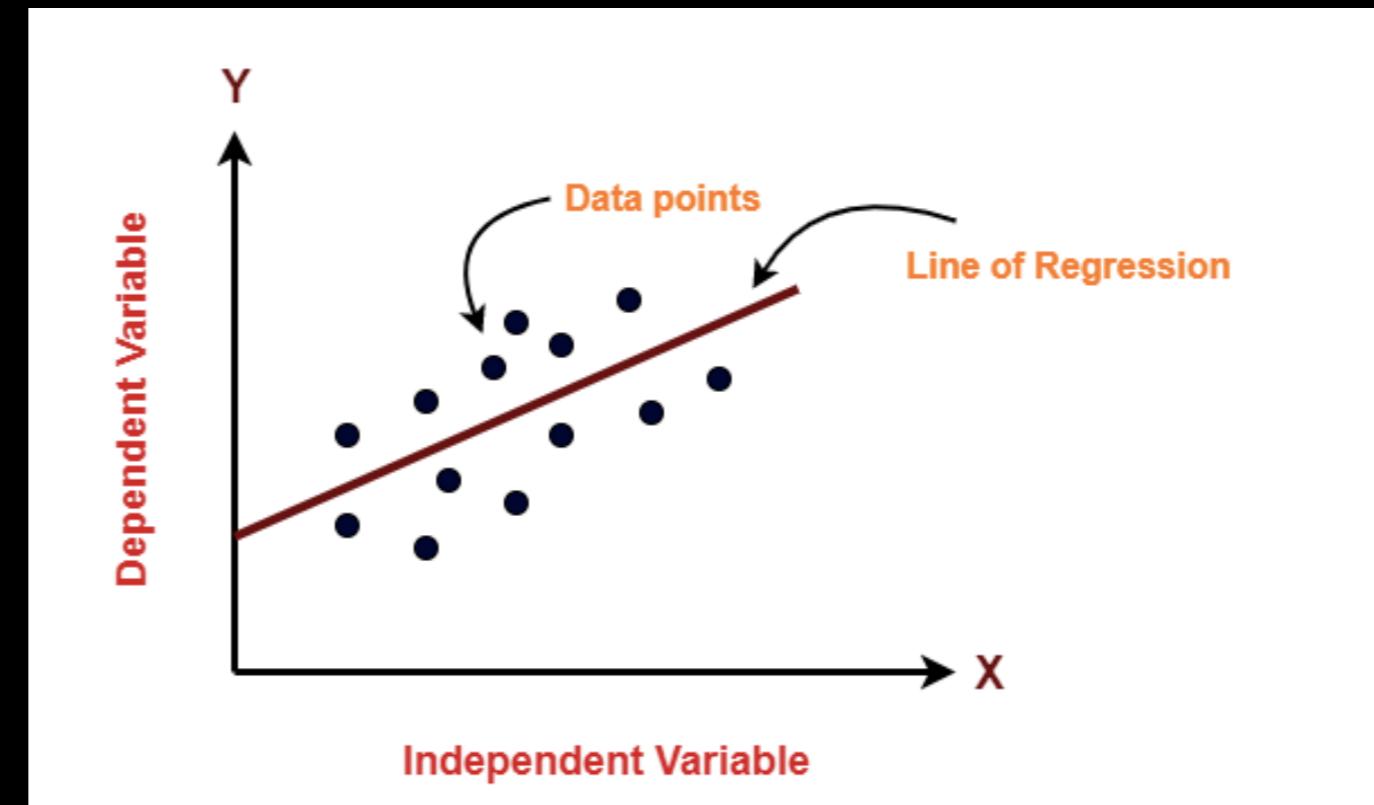
Regression is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values. It can also identify the distribution movement depending on the historical data.

- In a regression task, we are supposed to predict a continuous target variable using independent features.
- In the regression tasks, we are faced with generally two types of problems linear and non-linear regression.

Regression finds correlations between dependent(output) and independent (input) variables. Therefore, regression algorithms help predict continuous variables such as house prices, market trends, weather patterns, oil and gas prices (a critical task these days!), etc.

The Regression algorithm's task is finding the mapping function so we can map the input variable of "x" to the continuous output variable of "y."

- Linear Regression
- Decision Tree
- Support Vector Regression
- Lasso Regression
- Random Forest



Classification Algorithms

Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes i.e. discrete values. In classification, data is categorized under different labels according to some parameters given in the input and then the labels are predicted for the data.

- In a classification task, we are supposed to predict discrete target variables(class labels) using independent features.
- In the classification task, we are supposed to find a decision boundary that can separate the different classes in the target variable.

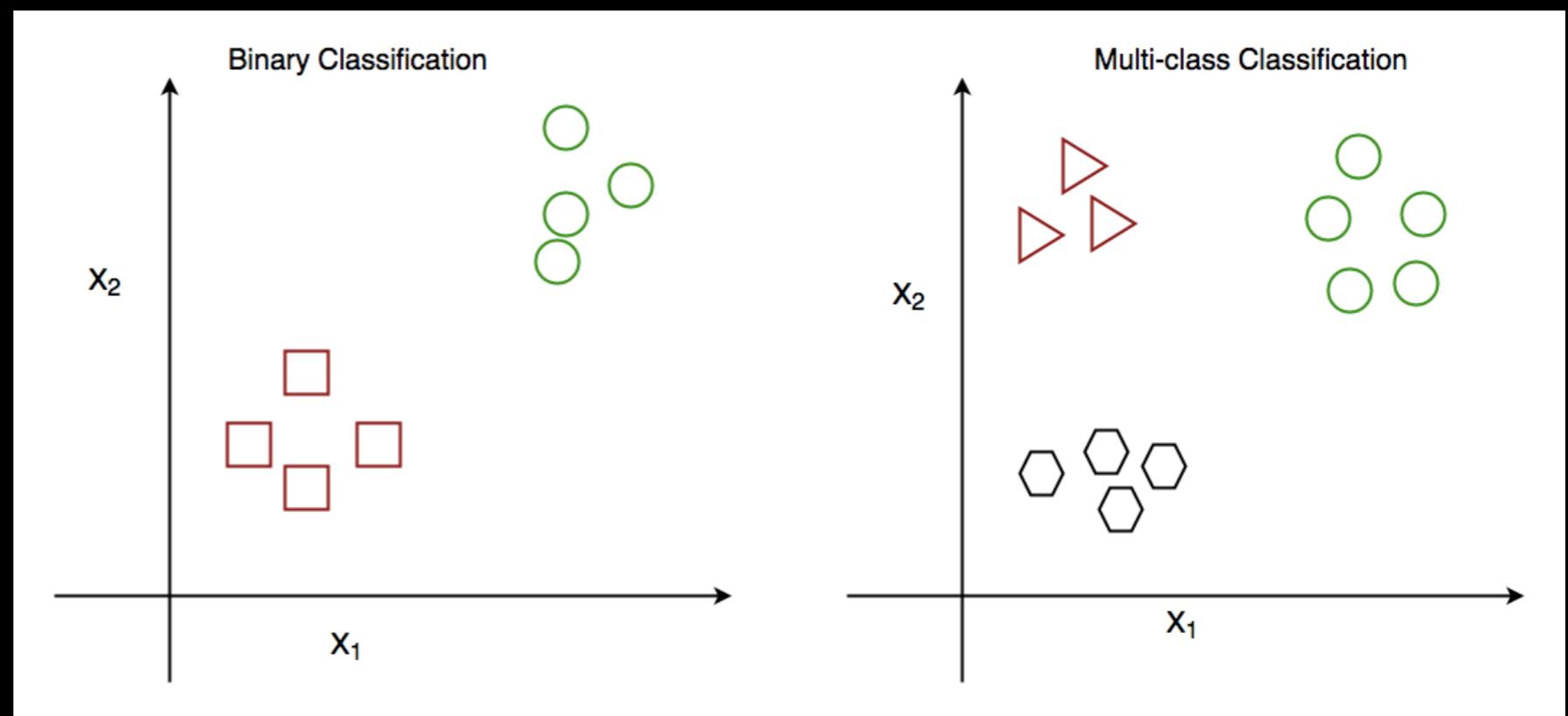
1. Logistic regression

2. Naive Bayes

3. K-Nearest Neighbors

4. Support Vector Machine

5. Decision Tree



Regression Algorithms	Classification Algorithms
The output variable must be either continuous nature or real value.	The output variable has to be a discrete value.
The regression algorithm's task is mapping input value (x) with continuous output variable (y).	The classification algorithm's task mapping the input value of x with the discrete output variable of y .
They are used with continuous data.	They are used with discrete data.
It attempt to find the best fit line, which predicts the output more accurately.	Classification tries to find the decision boundary, which divides the dataset into different classes.
Regression algorithms solve regression problems such as house price predictions and weather predictions.	Classification algorithms solve classification problems like identifying spam e-mails, spotting cancer cells, and speech recognition.
We can further divide Regression algorithms into Linear and Non-linear Regression.	We can further divide Classification algorithms into Binary Classifiers and Multi-class Classifiers.

Model Fitting in Machine Learning

Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained.

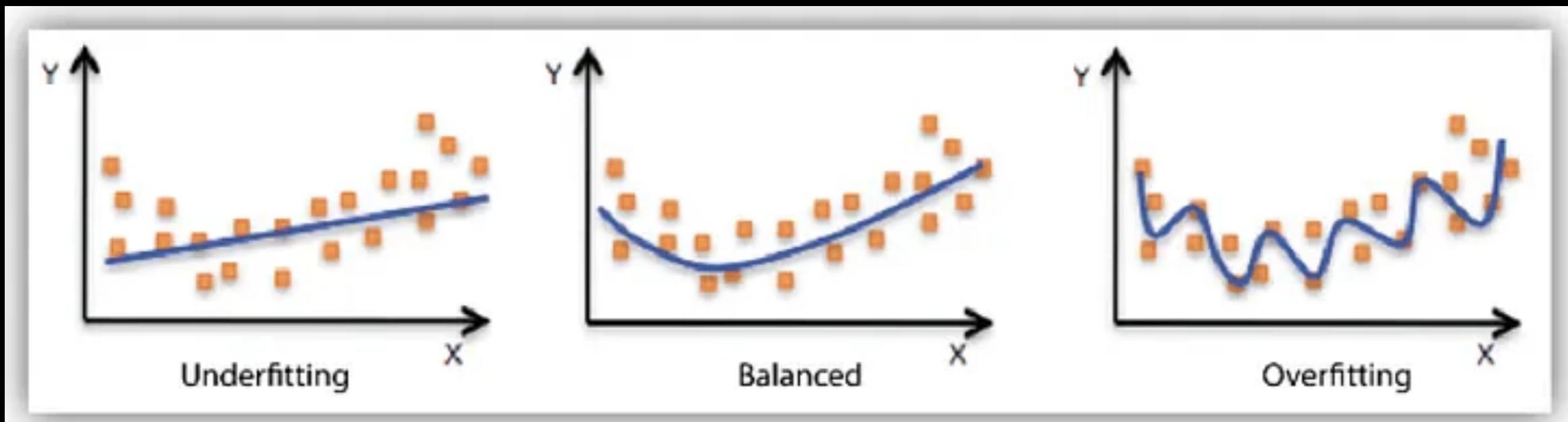
The generalization of a model to new data is ultimately what allows us to use machine learning algorithms every day to make predictions and classify data.

The cause of poor performance in machine learning models is either **overfitting** or **underfitting** the data.

A well-balanced model produces more accurate outcomes.

A model that is **overfitted** matches the data too closely.

A model that is **underfitted** doesn't match closely enough.



A machine learning algorithm is said to have underfitting when it is unable to capture the relationship between the input and output variables accurately.

It generates a high error rate on both the training set and unseen data. Hence, underfitting destroys the accuracy of our machine learning model.

It occurs when the data available to build a model is less or maybe when the model needs more training time and less regularization

How to avoid underfitting –

- Increase the duration of the training.
- Increasing the number of features by performing feature engineering.
- Remove noise from the data.
- Increase model complexity.

A brief about overfitting –

A machine learning algorithm is said to have overfitting when we see that the model performs well on the training data but does not perform well on the evaluation data. When this happens, the algorithm, unfortunately, cannot perform accurately against unseen data, defeating its purpose.

When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too many details and noise

How to avoid overfitting –

- Increase training data.
- Early stopping during the training phase.
- Ridge Regularization and Lasso Regularization.
- Feature reduction and dropouts.

A brief about a good-fit model –

Ideally, the case when the machine learning model makes the predictions with 0 error, is said to have a *good fit* on the data. This situation is achievable at a spot between overfitting and underfitting.

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> • High training error • Training error close to test error • High bias 	<ul style="list-style-type: none"> • Training error slightly lower than test error 	<ul style="list-style-type: none"> • Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	<ul style="list-style-type: none"> • Complexify model • Add more features • Train longer 		<ul style="list-style-type: none"> • Perform regularization • Get more data

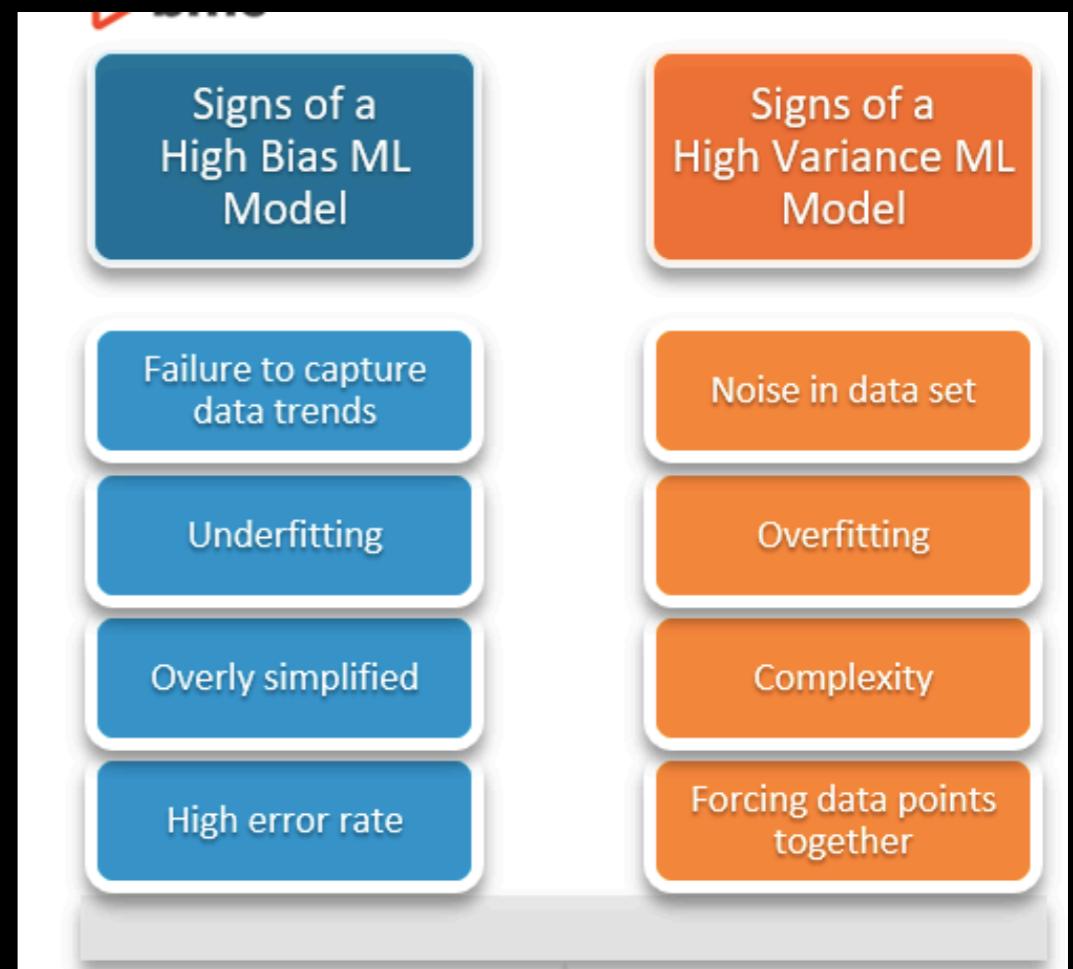
What is bias in machine learning?

Bias is a phenomenon that skews the result of an algorithm in favor or against an idea. Bias is considered a systematic error that occurs in the machine learning model itself due to incorrect assumptions in the ML process.

What is variance in machine learning?

Variance refers to the changes in the model when using different portions of the training data set.

- Models with high bias will have low variance.
- Models with high variance will have a low bias.



What is Feature Engineering

Feature engineering is a machine learning technique that leverages data to create new variables that aren't in the training set.

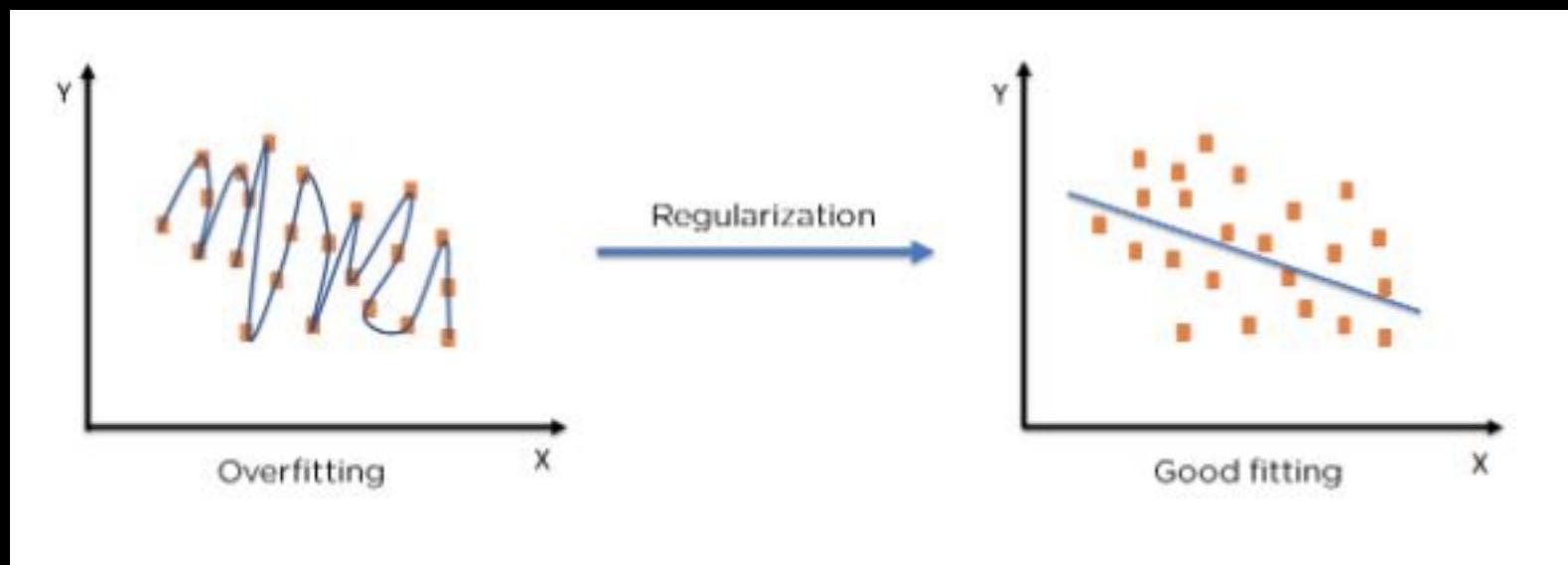
It can produce new features for both supervised and unsupervised learning, with the goal of **simplifying and speeding up data transformations** while also **enhancing model accuracy**.

Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.

[Link](#)

What is Regularization in Machine Learning?

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.



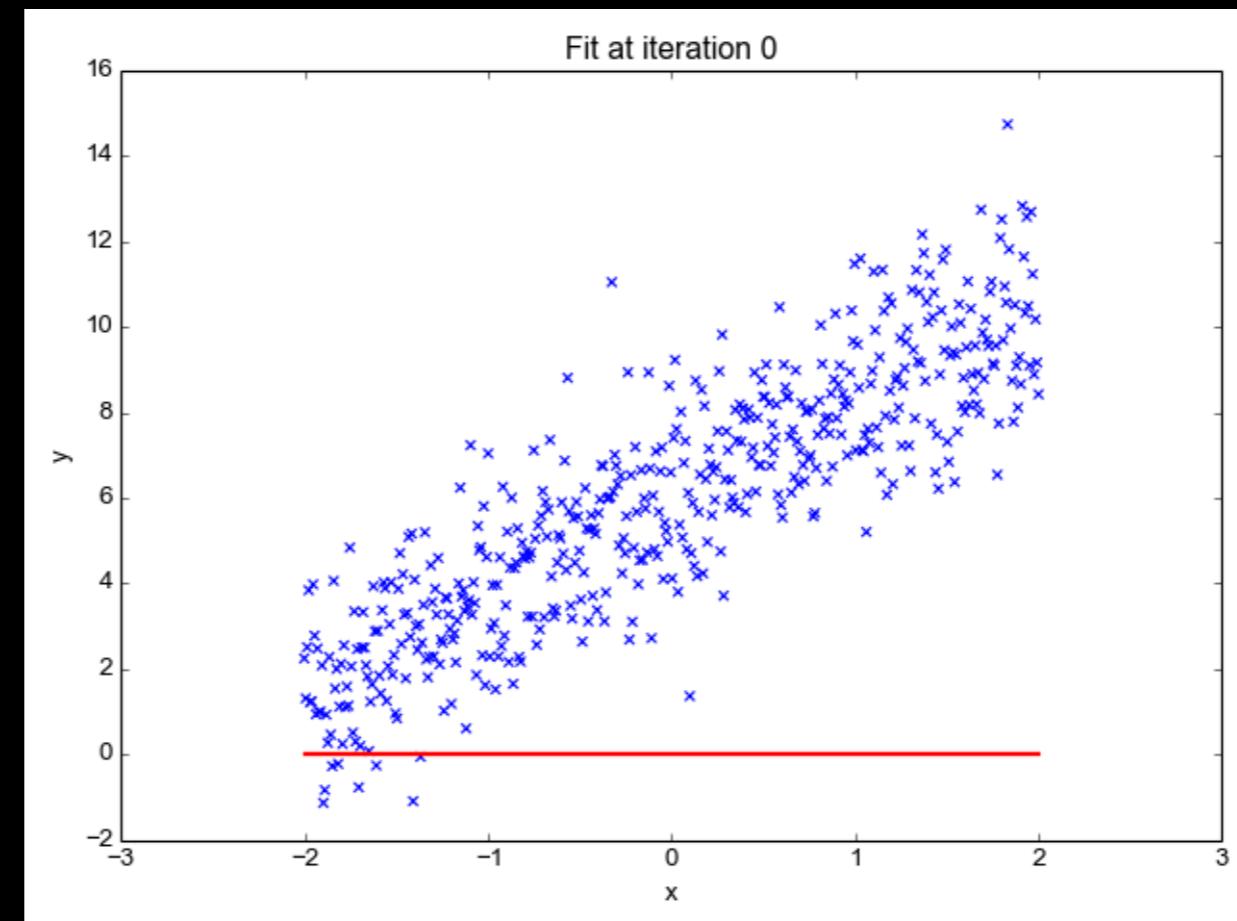
Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

[Link](#)

Linear regression

Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable(target) based on the given independent variable(s).

So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables. Hence, the name of this algorithm is Linear Regression. This best fit line is known as regression line and represented by a linear equation $Y= w *X + b$.



Pros:

- Linear Regression is simple to implement.
- Less complexity compared to other algorithms.
- Linear Regression may lead to over-fitting but it can be avoided using some dimensionality reduction techniques, regularization techniques, and cross-validation.

Cons:

- Outliers affect this algorithm badly.
- It over-simplifies real-world problems by assuming a linear relationship among the variables, hence not recommended for practical use-cases.

Implementation

- [Link](#)
- [Link](#)
- [Link](#)

Evaluation Metrics for Regression Model

Mean Absolute Error(MAE)

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

The diagram shows the formula for MAE: $MAE = \frac{1}{N} \sum |Y - \hat{Y}|$. Arrows point from the labels to the corresponding parts of the formula: 'Divide by total Number of Data Points' points to the fraction $\frac{1}{N}$; 'Actual Output' points to Y ; 'Predicted Output' points to \hat{Y} ; 'Sum Of' points to the summation symbol; and 'Absolute Value of residual' points to the absolute value expression $|Y - \hat{Y}|$.

Mean Squared Error(MSE)

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

$$MSE = \frac{1}{n} \sum \underbrace{\left(y - \hat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

Root Mean Squared Error(RMSE)

As RMSE is clear by the name itself, that it is a simple square root of mean squared error.

$$\text{RMSE} = \sqrt{\text{MSE}}$$
$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

R Squared (R²)

R² score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.

$$\textbf{R2 Squared} = 1 - \frac{\text{SSr}}{\text{SSm}}$$

SSr = Squared sum error of regression line

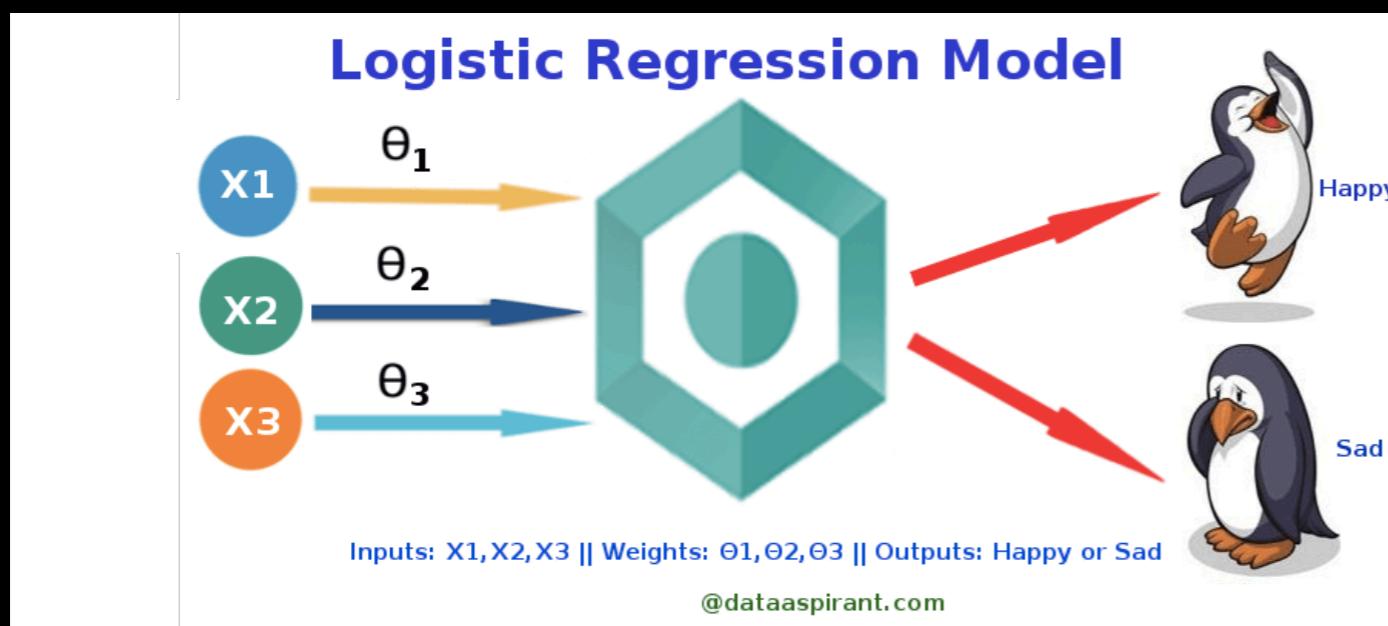
SSm = Squared sum error of mean line

Logistic Regression

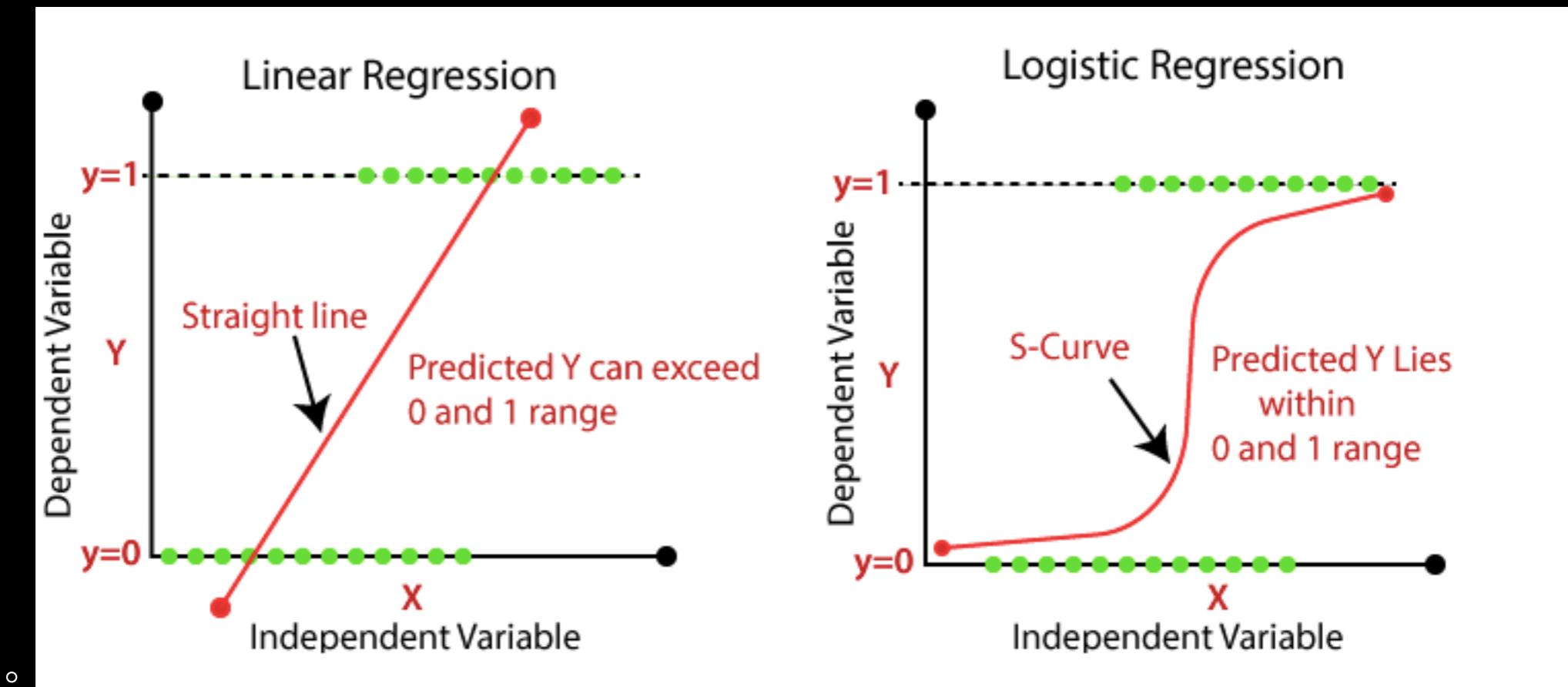
- Logistic regression is a statistical method used for binary classification problems, where the goal is to predict the probability of an event occurring based on input features.
- Key Concepts:
 - Dependent Variable: Binary outcome (e.g., Yes/No, 1/0)
 - Independent Variables: Input features used for prediction
 - Logistic Function: S-shaped curve used to model probabilities

For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

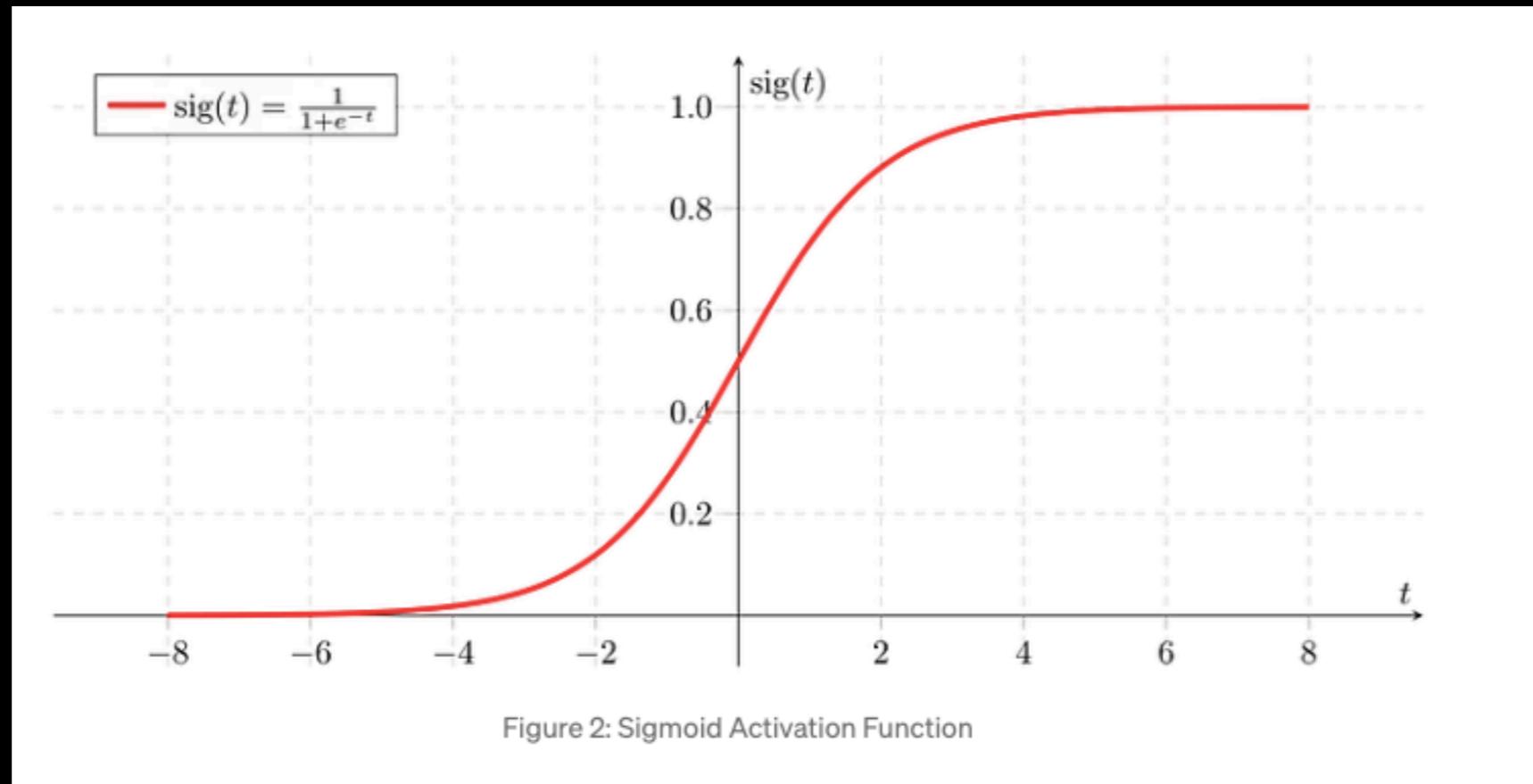


- Linear Regression:
 - Used for continuous target variables
 - Predicts a continuous value
 - Examples: predicting house prices, sales revenue
- Logistic Regression:
 - Used for binary classification
 - Predicts the probability of an event
 - Examples: spam email detection, disease diagnosis



- The logistic function (sigmoid function) is used to map any real-valued number to a value between 0 and 1.
- The S-form curve is called the Sigmoid function or the logistic function.
- **Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.

If 'Z' goes to infinity, Y(predicted) will become 1 and if 'Z' goes to negative infinity, Y(predicted) will become 0.



Here are some common terms involved in logistic regression:

- **Independent variables:** The input characteristics or predictor factors applied to the dependent variable's predictions.
- **Dependent variable:** The target variable in a logistic regression model, which we are trying to predict.
- **Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.

- **Odds:** It is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.

$$P(y = 1) = \sigma(z)$$

$$P(y = 0) = 1 - \sigma(z)$$

Applying natural log on odd. then log odd will be

$$\log \left[\frac{p(x)}{1-p(x)} \right] = z$$

$$\log \left[\frac{p(x)}{1-p(x)} \right] = w \cdot X + b$$

then the final logistic regression equation will be:

$$p(X; b, w) = \frac{e^{w \cdot X + b}}{1+e^{w \cdot X + b}} = \frac{1}{1+e^{-w \cdot X - b}}$$

- **Coefficient:** The logistic regression model's estimated parameters, show how the independent and dependent variables relate to one another.
- **Intercept:** A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.
- **Maximum likelihood estimation:** The method used to estimate the coefficients of the logistic regression model, which maximizes the likelihood of observing the data given the model.

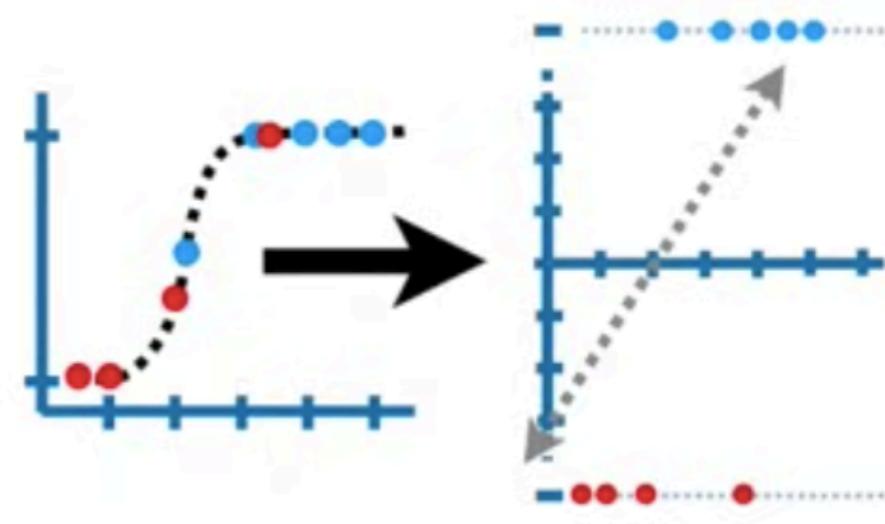
Maximum Likelihood Estimation

It is a method of determining the parameters (mean, standard deviation, etc) of normally distributed random sample data or a method of finding the best fitting PDF over the random sample data.

Maximum Likelihood Estimation (MLE) in logistic regression is a method to find the best-fitting parameters (coefficients) for a logistic regression model by maximizing the likelihood of the observed data. In simpler terms, it's about adjusting the model's settings to make it most likely to predict the actual outcomes based on the provided data.

[Link](#)

$$\log\left(\frac{p}{1-p}\right) = \text{log(odds)}$$



Exponentiate both sides...

$$\frac{p}{1-p} = e^{\text{log(odds)}}$$

Multiply both sides by $(1 - p)$...

$$p = (1 - p)e^{\text{log(odds)}}$$

Multiply $(1 - p)$ and $e^{\text{log(odds)}}$...

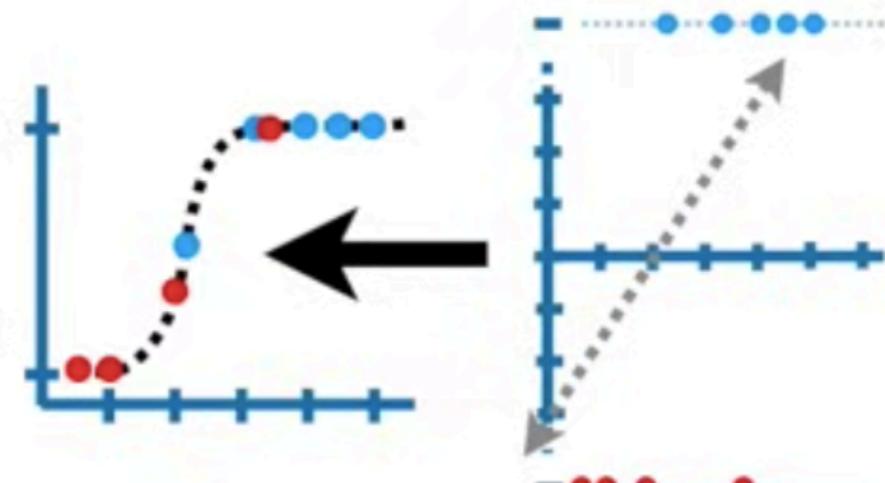
$$p = e^{\text{log(odds)}} - pe^{\text{log(odds)}}$$

Add $pe^{\text{log(odds)}}$ to both sides...

$$p + pe^{\text{log(odds)}} = e^{\text{log(odds)}}$$

Pull p out...

$$p(1 + e^{\text{log(odds)}}) = e^{\text{log(odds)}}$$



Divide both sides by $(1 + e^{\text{log(odds)}})$...

$$p = \frac{e^{\text{log(odds)}}}{1 + e^{\text{log(odds)}}}$$

Types of Logistic Regression

1. Binary Logistic Regression

The categorical response has only two possible outcomes.
Example: Spam or Not

2. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5

Decision Boundary

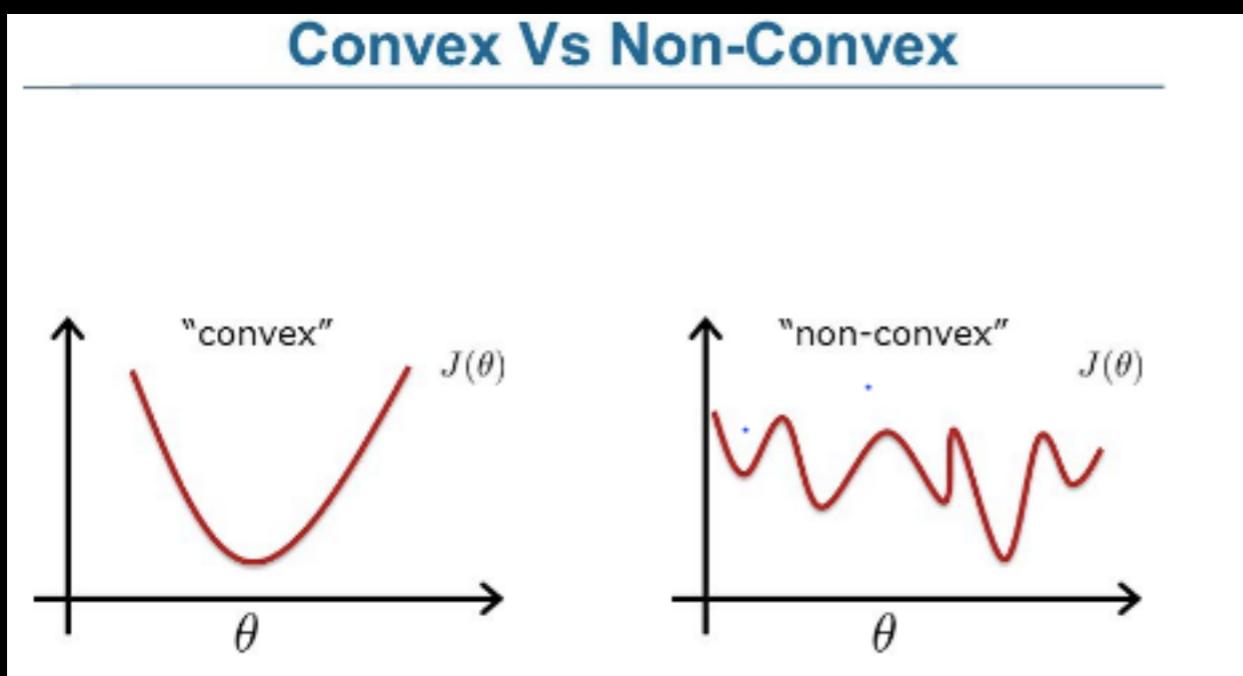
To predict which class a data belongs, a threshold can be set. Based upon this threshold, the obtained estimated probability is classified into classes.

Say, if $\text{predicted_value} \geq 0.5$, then classify email as spam else as not spam.

Decision boundary can be linear or non-linear. Polynomial order can be increased to get complex decision boundary.

Cost Function

Cost function or loss function is that function that describes how much the calculated value deviates from the actual value. Linear regression employs the least squared error as the cost function. But the least squared error function for logistic regression is non-convex. While performing gradient descent chances that we get stuck in a local minimum is more. So instead, we use log loss as the cost function.



The formula gives the cost function for the logistic regression.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

Gradient Descent

Gradient descent is an optimization algorithm used in machine learning, including logistic regression, to find the best set of parameters that minimize the cost function.

In the context of logistic regression, the cost function measures how well the model's predictions match the actual outcomes in the training data. The goal is to find parameter values that minimize this cost function.

The gradients are the vector of the 1st order derivative of the cost function.

By differentiating the cost function, we get the gradient descent expression

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y) x_j^{(i)}$$

Metrics to Evaluate your Classification Model to take the right decisions

Evaluation metrics are tied to machine learning tasks. There are different metrics for the tasks of classification and regression.

Using different metrics for performance evaluation, we should be able to improve our model's overall predictive power before we roll it out for production on unseen data.

Without doing a proper evaluation of the Machine Learning model by using different evaluation metrics, and only depending on accuracy, can lead to a problem when the respective model is deployed on unseen data and may end in poor predictions.

Confusion Matrix

A **Confusion matrix** is an $N \times N$ 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.

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

The predicted value is positive and its positive

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

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

The predicted value is Negative and its Negative

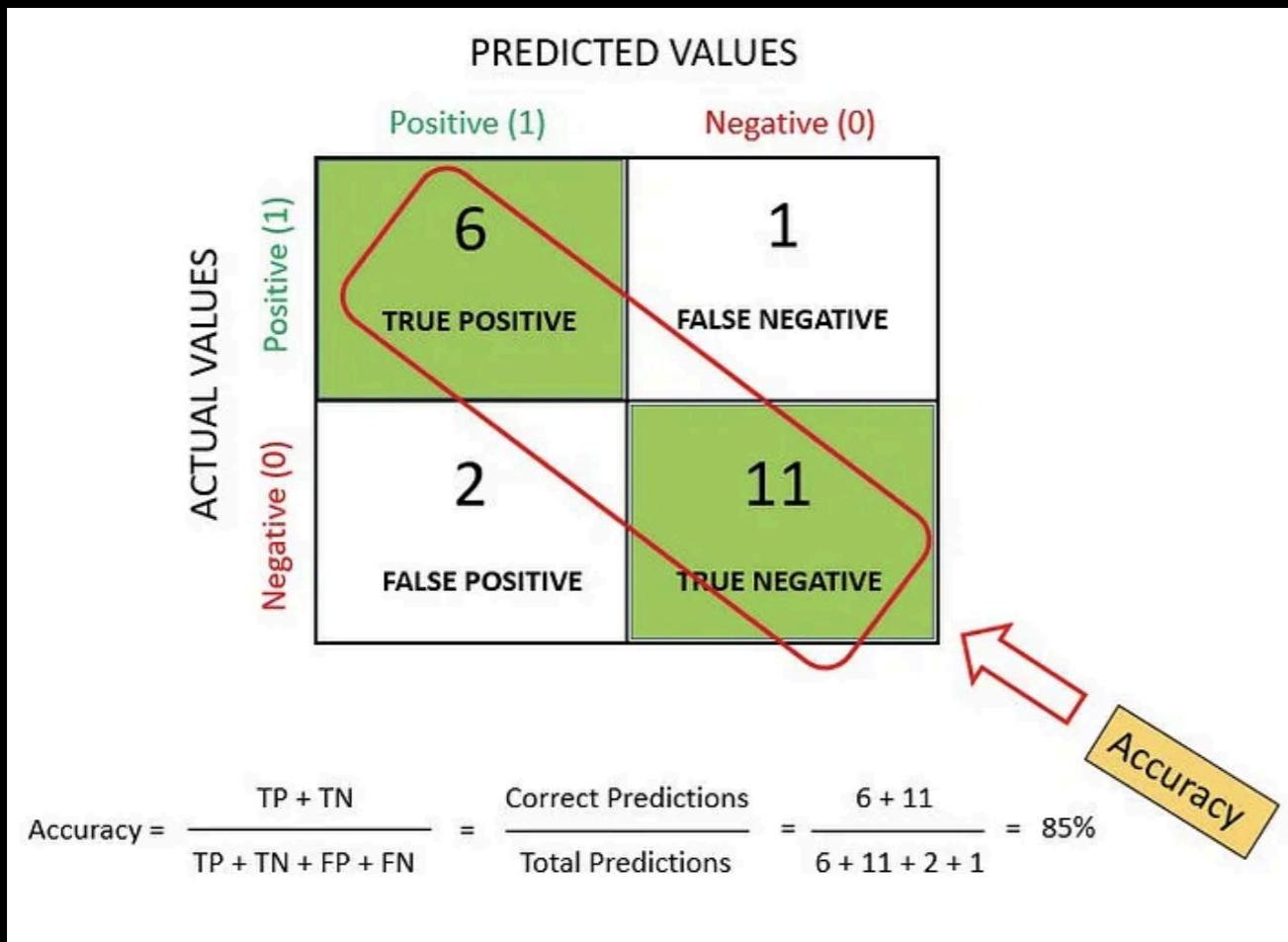
Understanding Confusion Matrix:

- **True Positives (TP)**: when the actual value is Positive and predicted is also Positive.
- **True negatives (TN)**: when the actual value is Negative and prediction is also Negative.
- **False positives (FP)**: When the actual is negative but prediction is Positive. Also known as the **Type 1 error**
- **False negatives (FN)**: When the actual is Positive but the prediction is Negative. Also known as the **Type 2 error**
- A good model is one which has high TP and TN rates, while low FP and FN rates.
- If you have an ***imbalanced dataset*** to work with, it's always better to use ***confusion matrix*** as your evaluation criteria for your machine learning model

Classification Measure

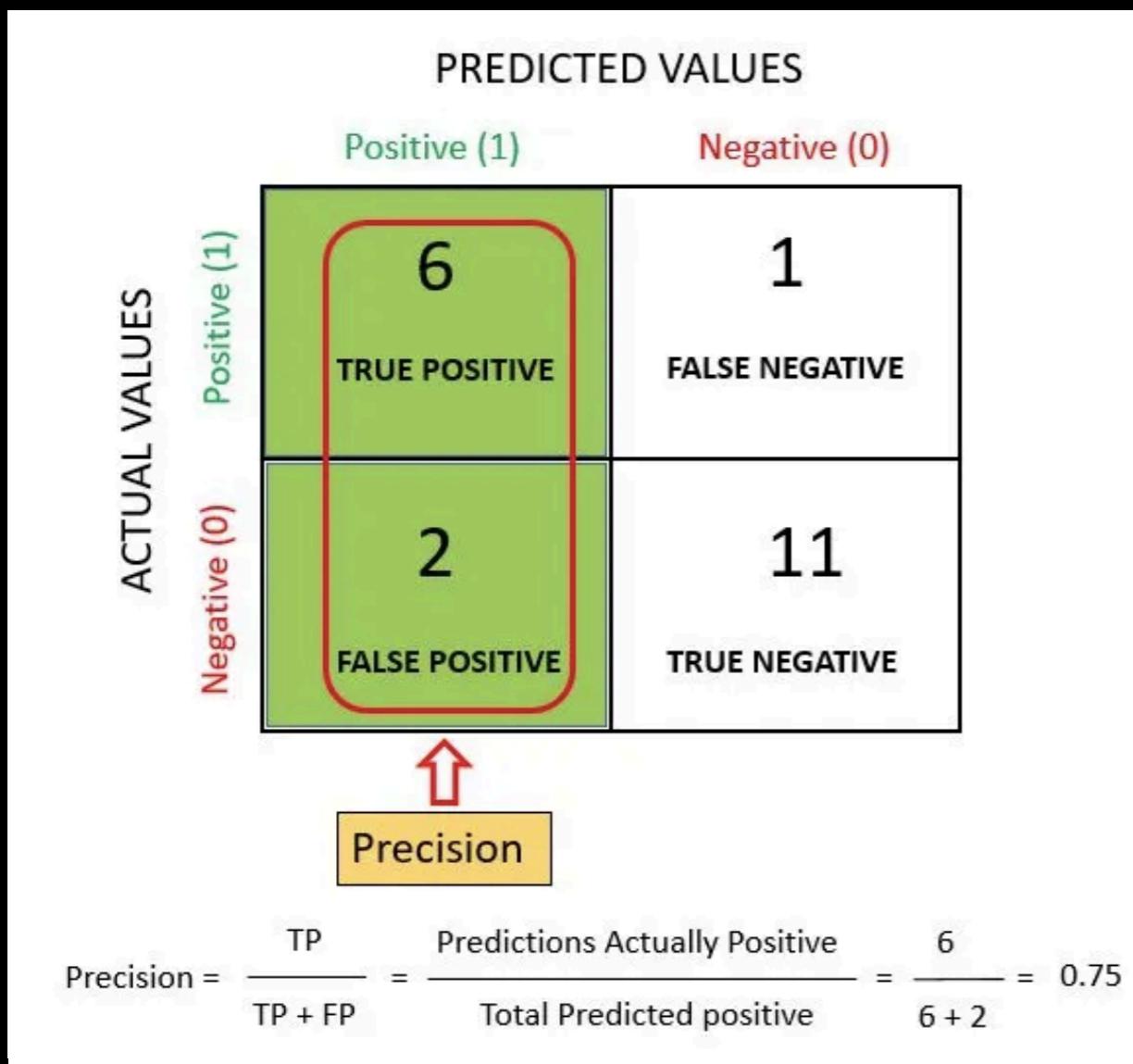
1. Accuracy

Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.



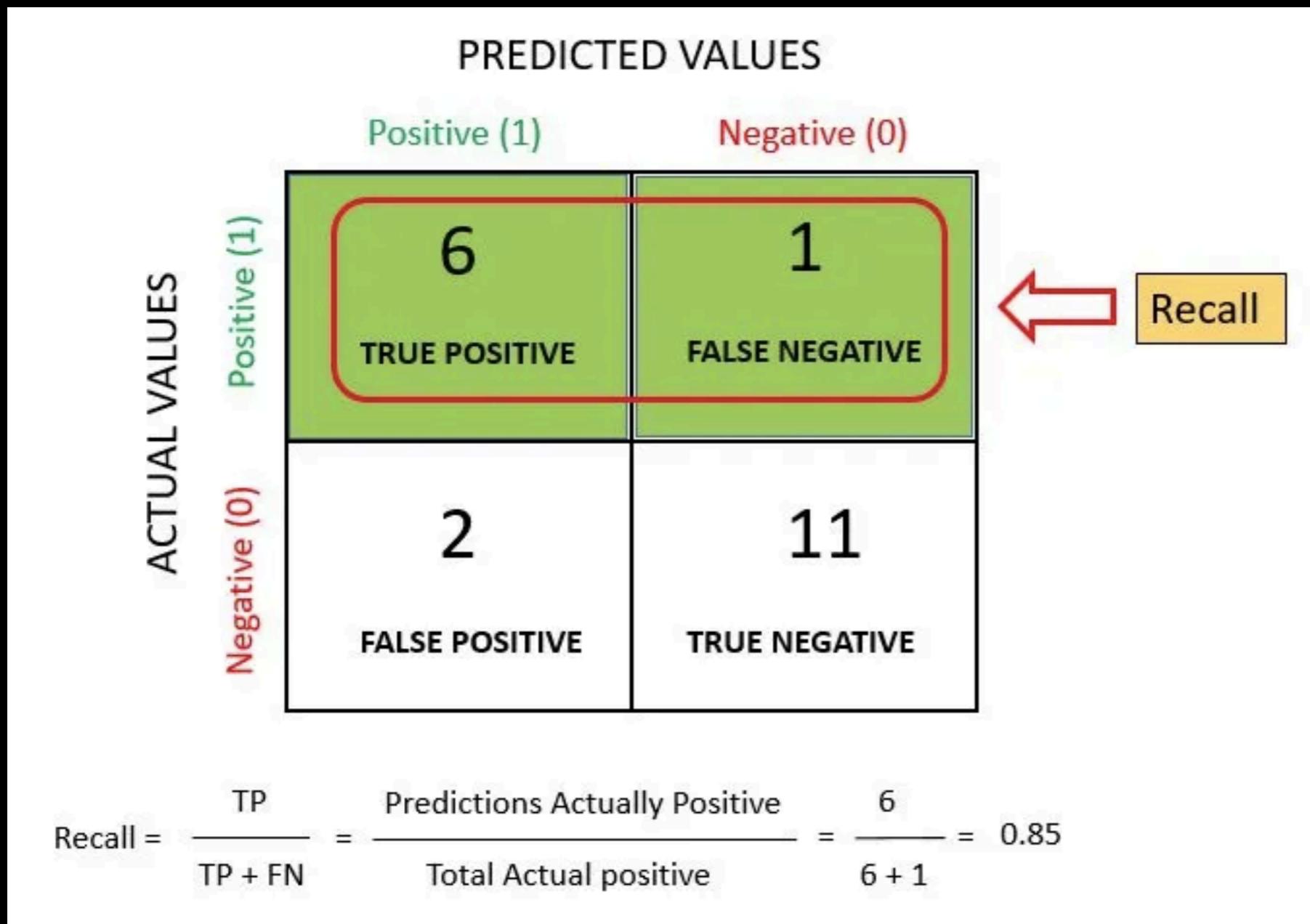
2. Precision

It is a measure of **correctness** that is achieved in **true prediction**. In simple words, it tells us how many predictions are **actually positive** out of all the **total positive predicted**.



3. Recall

It is a measure of **actual observations** which are predicted **correctly**, i.e. how many observations of positive class are actually predicted as positive. It is also known as **Sensitivity**. **Recall** is a valid choice of evaluation metric when we want to capture **as many positives** as possible.



4. F-measure / F1-Score

The **F1 score** is a number between **0 and 1** and is the **harmonic mean of precision and recall**. We use harmonic mean because it is not sensitive to extremely large values, unlike simple averages.

$$\text{F1-Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} = 2 * \frac{(0.85 * 0.75)}{(0.85 + 0.75)} = 0.79$$

Links

1. [Logistic Regression from scratch\(iris\).ipynb](#)
2. <https://colab.research.google.com/drive/1t3LmkEly8B9Oeg5CVaVG151w7Ks17krP?usp=sharing>
3. <https://medium.com/@martinpella/logistic-regression-from-scratch-in-python-124c5636b8ac>

Linear Regression

Logistic Regression

Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables.
Linear Regression is used for solving Regression problem.	Logistic regression is used for solving Classification problems.
In Linear regression, we predict the value of continuous variables.	In logistic Regression, we predict the values of categorical variables.
In linear regression, we find the best fit line, by which we can easily predict the output.	In Logistic Regression, we find the S-curve by which we can classify the samples.
Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for estimation of accuracy.
The output for Linear Regression must be a continuous value, such as price, age, etc.	The output of Logistic Regression must be a Categorical value such as 0 or 1, Yes or No, etc.
In Linear regression, it is required that relationship between dependent variable and independent variable must be linear.	In Logistic regression, it is not required to have the linear relationship between the dependent and independent variable.
In linear regression, there may be collinearity between the independent variables.	In logistic regression, there should not be collinearity between the independent variable.

Label Encoding

Label Encoding is a technique that is used to convert categorical columns into numerical ones so that they can be fitted by machine learning models which only take numerical data. It is an important pre-processing step in a machine-learning project.

Suppose we have a column *Height* in some dataset that has elements as Tall, Medium, and short. To convert this categorical column into a numerical column we will apply label encoding to this column.

After applying label encoding, the Height column is converted into a numerical column having elements 0,1, and 2 where 0 is the label for tall, 1 is the label for medium, and 2 is the label for short height.

Height	Height
Tall	0
Medium	1
Short	2

[LINK](#)

[LINK](#)

Other Classification-Models

1. Decision Tree Classification
2. Random Forest Classifier
3. K Nearest Neighbour Classification
4. SVM Classification
5. Naive Bayes Classification

REFERENCE

<https://www.geeksforgeeks.org/understanding-logistic-regression/>

<https://medium.com/swlh/decision-tree-classification-de64fc4d5aac>

<https://builtin.com/data-science/random-forest-algorithm>

<https://medium.com/swlh/random-forest-classification-and-its-implementation-d5d840dbead0>

<https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

<https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>

<https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>

<https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn>

<https://developers.google.com/machine-learning/crash-course/classification/accuracy>

<https://towardsdatascience.com/model-evaluation-techniques-for-classification-models-eac30092c38b>

<https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>