Machine Learning Fundamentals Summary

by Abdulkareem Alanazi

linkedin.com/in/akrm1

Motivation

- I had a great journey at SDAIA T5 bootcamp for data science and that was from 24/10/2021 to 13/1/2022, and one of the best things that happened in my life that I met class 21 in the bootcamp which has the best people I have ever known.
- After every section we were summarizing what we learned in that section in a simple way to improve our understanding, so after graduating from the bootcamp I decided to make a summary that focus on machine learning that cover all its concepts that we took.

Notes About Material

- The most of the concepts in this material has videos that attached with its page.
- Every text in this material with the light blue color it clickable.
- I assume that you have some background knowledge in statistics, linear algebra, calculus, and python.
- We need to understand what we will learn as concepts and how to apply, not as some steps that we will apply.
- The most of the problems we will face have common workflow, but every problem have the uniqueness characteristics that needs to be understand and fits what we learned to them.
- "The code is written, But the instructions must be given" by me XD.

Index

click on any section title to navigate to it

Index

Introduction

- Data Forms
- Data Types
- Data Science with Machine Learning
- Machine Learning

Introduction

(Data Forms)

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

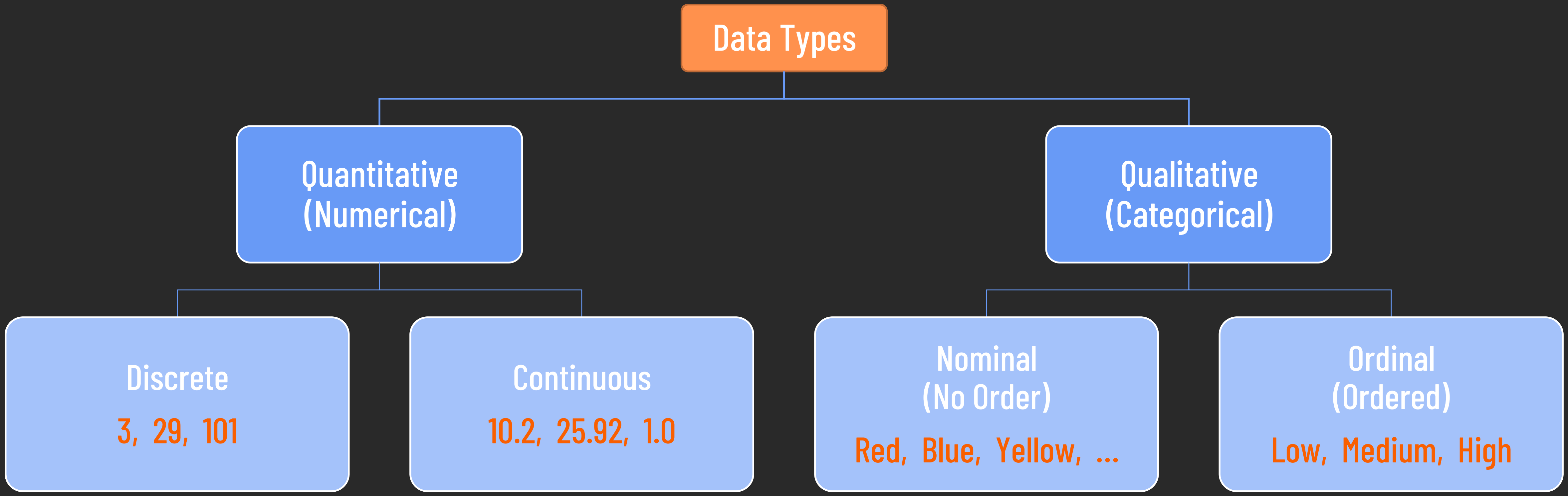
Examples

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

LEVITY

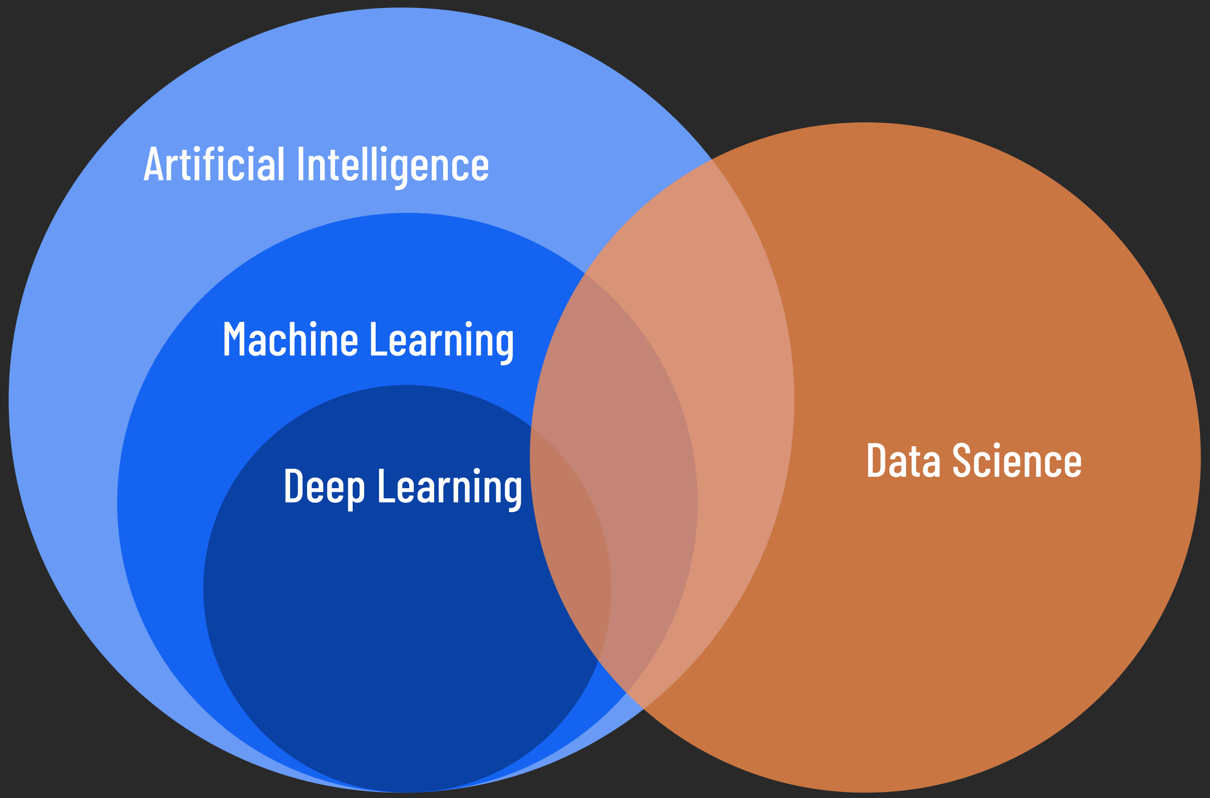
Introduction

(Data Types)



Introduction

(Data Science with Machine Learning)



Introduction

(Machine Learning)

- Machine Learning: is a field that solves specific problems by studying the patterns of the data and minimizing the error, and it is considered a sub-category of Artificial Intelligence.
  Another Definition: is an application of Artificial Intelligence where in the system gets the ability to automatically learn and improve based on experience.
- Some Terminologies we need to be familiar with:
  - Model: Which is the machine learning algorithm that used to solve the problem.
  - Features / Independent Variable (x): Which is the input variables that we feed to the model, noted by x.
  - Label / Target / Dependent Variable (y): Which is the variable we want to predict.
  - Observation: Which is the single piece of the information (row in the table).
- It is need to be there a dependency between the label and the features in the data so can be predict the label based on the features
- And if there is a dependency between two features that means the two feature give the same info, so one of them not needed
- The machine learning have two common sub-categories that we will focusing on:
  - Supervised Machine Learning: is the use of labeled data to train algorithms to classify data or predict outcomes
  - Unsupervised Machine Learning: is focusing on identify patterns in the data

Introduction

(Machine Learning)

Supervised Learning

Features

$$\underbrace{\boxed{X_1}\boxed{X_2}\boxed{X_{3..}}\boxed{X_n}}_{\text{Features}}\underbrace{\boxed{Y}}_{\text{Label}}$$

Label

Has a Label/Target

Unsupervised Learning

Features

$$\underbrace{\boxed{X_1}\boxed{X_2}\boxed{X_{3..}}\boxed{X_n}}_{\text{Features}}$$

Has not a Label/Target

Introduction

(Machine Learning)

- Machine Learning
  - Supervised (has label)
    - Regression (Continuous Label)
    - Classification (Categorical Label)
  - Unsupervised (hasn't label)
    - Clustering
    - Dimensionality Reduction

Dataset Operations

- Splitting The Dataset
- Features Engineering
- Features Selection
- Feature Extraction

Dataset Operations

(Splitting The Dataset)

- Before feeding the dataset into the model, it's need to split into:
  - Train: Which is the sub-dataset of the original dataset that train the model on it, to find patterns.
  - Validation: Which is the sub-dataset of the original dataset that used to validate the model performance during training.
    - Validation process gives information that helps us tune the model's hyperparameters and configurations.
    - The main idea of splitting the dataset into a validation set is to prevent our model from overfitting.
  - Test: Which is the sub-dataset of the original dataset that used to test the model after completing the training.
    - it answers the question of "How well does the model perform?"

Common split with small and normal data:

Train	Validation	Test
60%	20%	20%

Common split with big data:

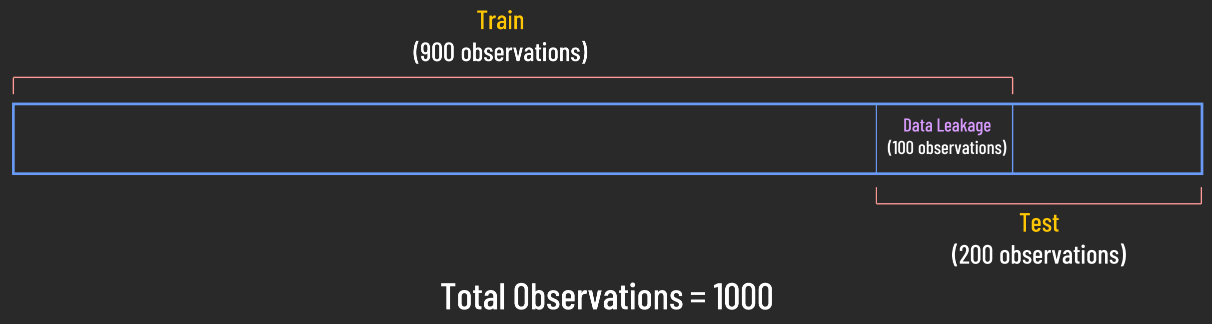
Train	Validation	Test
90%	5%	5%

Code Tutorial

Dataset Operations

(Splitting The Dataset)

- **Data Leakage:** it a concept means the model trained on a data that will used in validation or testing, so it will get good results on validation and testing, but it will show poor results in production(data that not seen before).
  - Example:



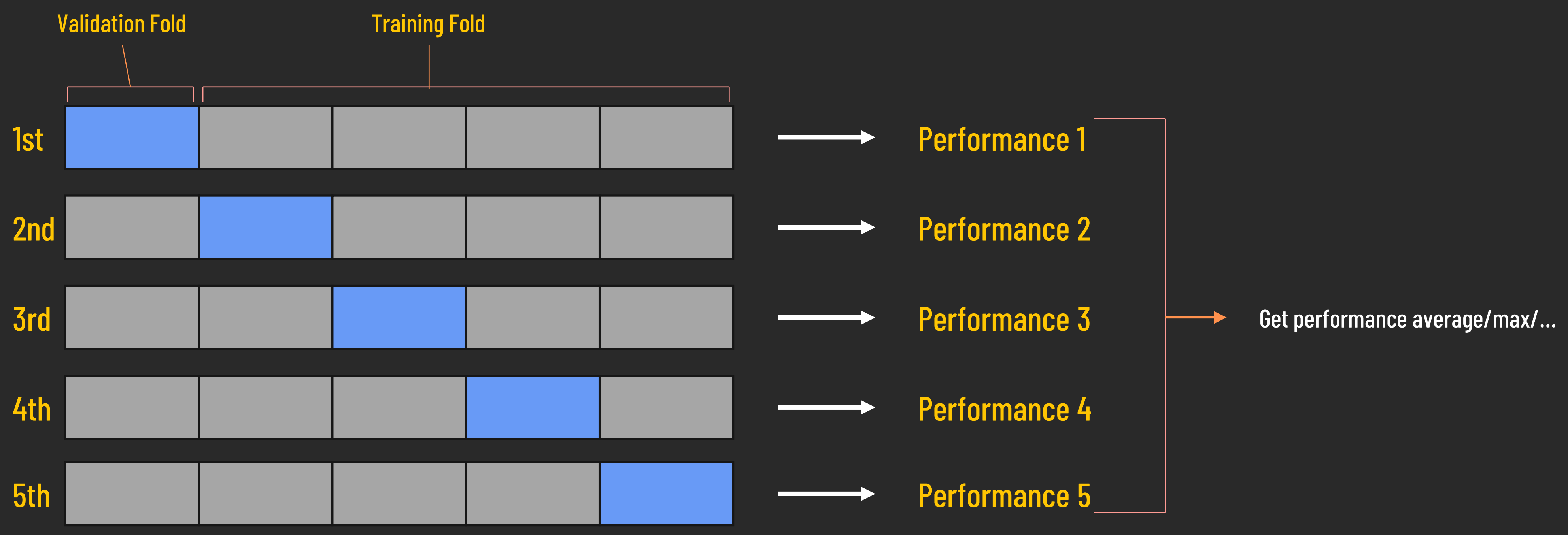
- **Cross Validation(CV):** it is a technique that is used to train and evaluate the model on a portion of the dataset, and it re-portioning several times and repeats the process to find the best model, the most common techniques (not limited):
  - **K-Fold CV:** the most technique used.
  - **Leave One Out Cross Validation(LOOCV):** it is an extreme case of k-fold where k equal to the number of observations, it is not commonly used.
  - **Stratified K-Fold CV:** it fix the k-fold problem with imbalance data.
  - **Time Series CV:** used with time series.

Data Leakage Cross Validation

Dataset Operations

(Splitting The Dataset)

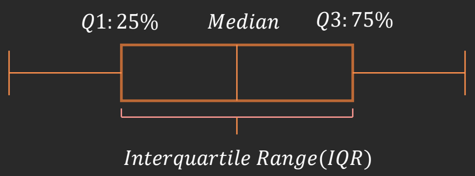
- **K-Fold CV:** Let's see how k-fold will be with k = 5



Dataset Operations

(Features Engineering)

- Features Engineering is a process of preparing the data for the model, these some of common techniques (not limited):
- **Imputation:** Which is handling the missing values by Replace it (mean/mode/median/max/min/...), or Remove it.
- **Handling Outliers:** Replace it (mean/mode/median/max range/min range/...), or Remove it.
  - Min Limit $= Q1 - 1.5 \times IQR$
  - Max Limit $= Q3 + 1.5 \times IQR$



- **One-Hot Encoding:** Convert categorical data into binary data, it used with categorical data that has no order.

color
red
blue
green

→

red	blue	green
1	0	0
0	1	0
0	0	1

- **Label / Integer Encoding:** Convert categorical data into numeric data, it used with categorical data that has an order.

risk
low
medium
high

→

risk
0
1
2

Imputation Handling Outliers Encoding

Dataset Operations

(Features Engineering)

- Scaling: is the process of converting all the selected features to the same scale(range).
  - Every numeric feature in the dataset has unit and magnitude, the model will give high importance to features that have high magnitude and low importance to features that have low magnitude.
  - Features scaling is used when the features magnitudes make a difference in the model base algorithm to find the solution.
  - Scaling will be necessary depends on fitting algorithms, there are three types of fitting algorithms (see its section for more details):
    - Gradient Descent Based Algorithms — Like: linear regression, logistic regression, ...
    - Distance-Based Algorithms — Like: KNN, K-means, SVM, ...
    - Tree-Based Algorithms — Like: Decision Tree, Random Forest, ...
  - Usually In Gradient Descent Based algorithms and Distance-Based algorithms scaling is required.
  - Usually In Tree-Based algorithms scaling is not required.
  - Scaling advantages:
    - Makes training faster.
    - Improve the performance.

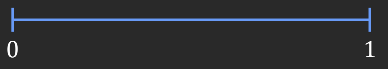
Scaling

Dataset Operations

(Features Engineering)

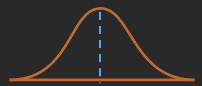
- **Scaling:** is the process of converting all the selected features to the same scale(range).
  - Some scaling techniques:
    - Normalization: is a scaling technique in which the values are rescaled between the range 0 to 1, It is also known as Min-Max scaling, useful when we don't know about the distribution.

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



    - Standardization: is a scaling technique in which the mean will be equal to zero and the standard deviation equal to one, useful when the feature distribution is normal or gaussian, Note that in this case the values are not restricted to a particular range.

$$X_{new} = \frac{X_i - \mu}{\sigma}$$


Normal Distribution

    - Robust Scalar: is one of the best scaling techniques when we have outliers present in our dataset. It scales the data accordingly to the interquartile range ($IQR = Q3 - Q1$).

$$X_{new} = \frac{X - X_{median}}{IQR}$$

    - Gaussian Transformation: When our dataset doesn't follow Gaussian/Normal distribution(Bell Curve) then we used Gaussian transformation.

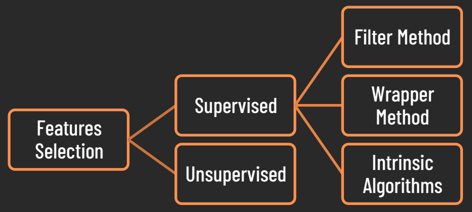
Standardization & Normalization

Dataset Operations

(Features Selection)

- Features Selection is the process of reducing the input variables to the model by using only relevant data and getting rid of noise in data, it consider as dimensionality reduction approach.

- One way to think about feature selection methods are in terms of supervised and unsupervised methods.
  - Supervised Methods: methods that remove irrelevant variables.
  - Unsupervised Methods: methods that remove redundant variables.



- Advantages:
  - It prevent learning from noise(help to solve overfitting).
  - improved accuracy.
  - reduce training time.

Dataset Operations

(Features Selection)

- Supervised Methods:
  - Filter Method: drop features based on its correlations with the target, this method is based on statistics, some techniques:
    - Pearson's: used between two numerical variables.
    - Spearman's: used between two numerical variables.
    - ANOVA: used between numerical variable and categorical variable.
    - Kendall's: used between numerical variable and categorical variable.
    - Chi Squared: used between two categorical variables.
    - Mutual Information: used between two categorical variables.

	Numerical	Categorical
Numerical	Pearson's Spearman's	ANOVA Kendall's
Categorical	ANOVA Kendall's	Chi Squared Mutual Information

  - Wrapper Method: generate subset of the features then try and test to find best subset that give best model, some techniques:
    - Recursive Feature Elimination(RFE): is a wrapper-style feature selection algorithm that also uses filter method or intrinsic algorithm internally, RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains.
    - Genetic Algorithms: One of the most advanced algorithms for feature selection, Genetic Algorithms (GA) are a mathematical model inspired by the famous Charles Darwin's idea of natural selection, it is a stochastic method for function optimization based on the mechanics of natural genetics and biological evolution.
  - Intrinsic Algorithms / Embedded Method: that perform automatic feature selection during training, some techniques:
    - Lasso Regularization: more about it in regression section.
    - Decision Trees: more about it in classification section.

Theory & Code Tutorial   Code Tutorial

Dataset Operations

(Features Extraction)

- Features Extraction is the process of extracting new input variables from original ones, it is also consider as dimensionality reduction approach.
- We can extract new input variables by using basic mathematical operations like (+ - * / ...)
  - Example: we have the height and weight of houses, we can extract new feature which is the area ($area = height \times weight$), and also we can extract another feature which is the perimeter ($perimeter = 2(height + weight)$)

Height	Weight
3	7
5	2
6	3

$\xrightarrow{Height \times Weight}$

Area
21
10
18

- But most of the time we will extract new input variables by using linear algebra operations like (Vectors Operations, Linear Combinations, Transformation, Matrix Decomposition, ...)
- Common Features Extraction techniques use linear algebra operations:
  - Singular Value Decomposition (SVD): more about it in dimensionality reduction section.
  - Principal Component Analysis (PCA): more about it in dimensionality reduction section.

Index

Model Fitting

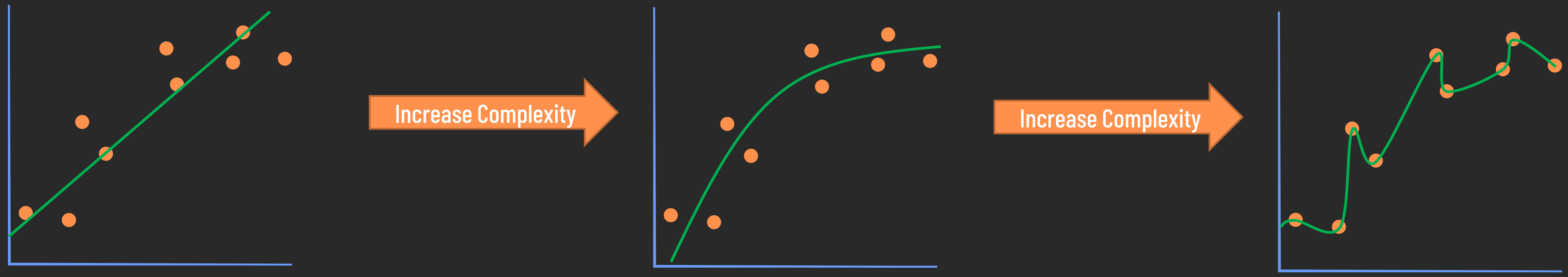
- Bias
- Variance
- Underfitting & Overfitting
- Bias-Variance Trade-Off
- Solutions

Visual Theory Visual Theory 2

Model Fitting

(Bias)

- Bias is the difference between the predicted values and the actual values.
  - Low Bias: means the difference is low, which means the predicted values is close to the actual values.
  - High Bias: means the difference is high, which means the predicted values is far away from actual values.
- When the complexity of the model increase, the bias decrease, Inverse relationship.
- More of the complexity we add the more the bias will decrease.
- Examples:
  - if we train a model on 1000 data points, and it gave us 96% accuracy, conclusion: that's mean our model have low bias.
  - if we train a model on 500 data points, and it gave us 54% accuracy, conclusion: that's mean our model have high bias.

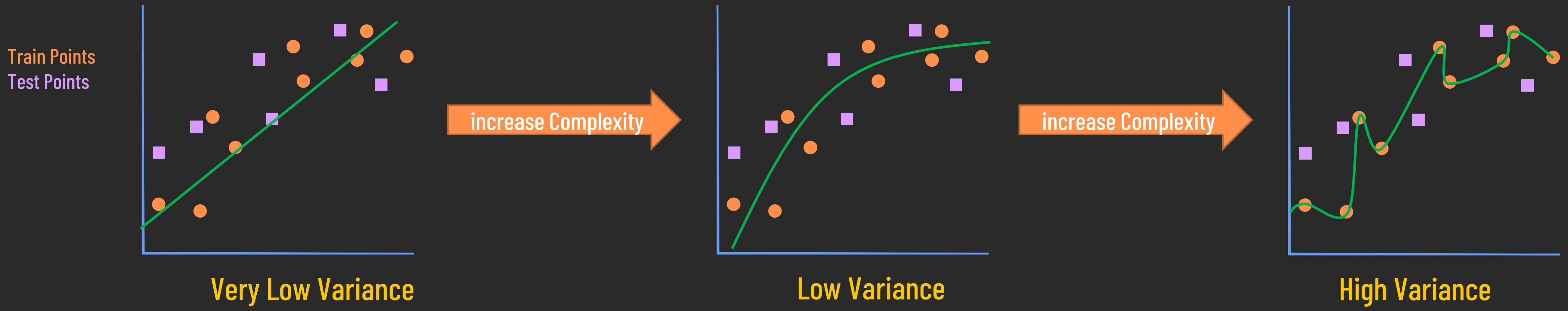


High Bias | Low Bias | Very Low Bias

Model Fitting

(Variance)

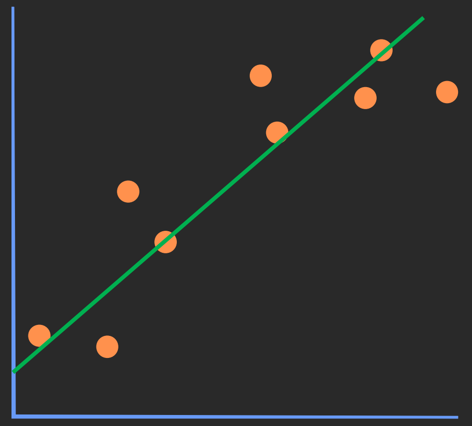
- Variance is the variability of model prediction for different datasets
  - Low Variance: means the model prediction have almost the same results with different datasets.
  - High Variance: means the model prediction have noticed difference between the results with different datasets.
- When the complexity of the model increase, the variance increase, positive relationship.
- The more of the complexity we remove the more the variance will decrease, and our model will be more generalized to different datasets.
- Examples:
  - if we train a model on 1000 data points, and it gave us 78% accuracy, then we test it with new 300 data points and it gave us 54% accuracy,
    conclusion: when we look at the accuracy of the same model at different data points have a big difference that means we have high variance.
  - if we train a model on 500 data points, and it gave us 63% accuracy, then we test it with new 150 data points and it gave us 65% accuracy,
    conclusion: when we look at the accuracy of the same model at different data points have a low difference that means we have low variance.



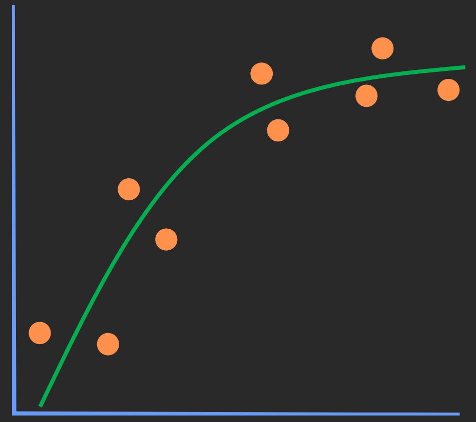
Model Fitting

(Underfitting & Overfitting)

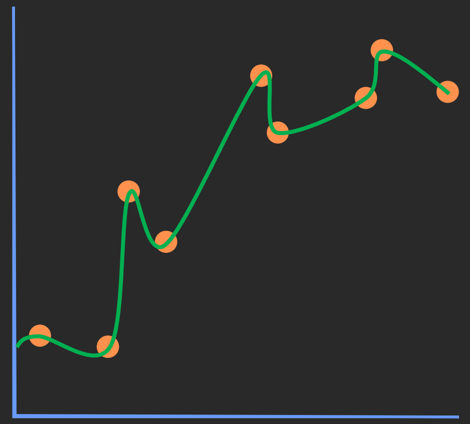
- There are two famous fitting problems in machine learning:
  - Underfitting: It happens when the model have high bias, which means the model have not trained enough.
  - Overfitting: It happens when the model have high variance, which means the model have trained so much on the training dataset, not generalized.



Underfitting
(High Bias)



Optimal Solution

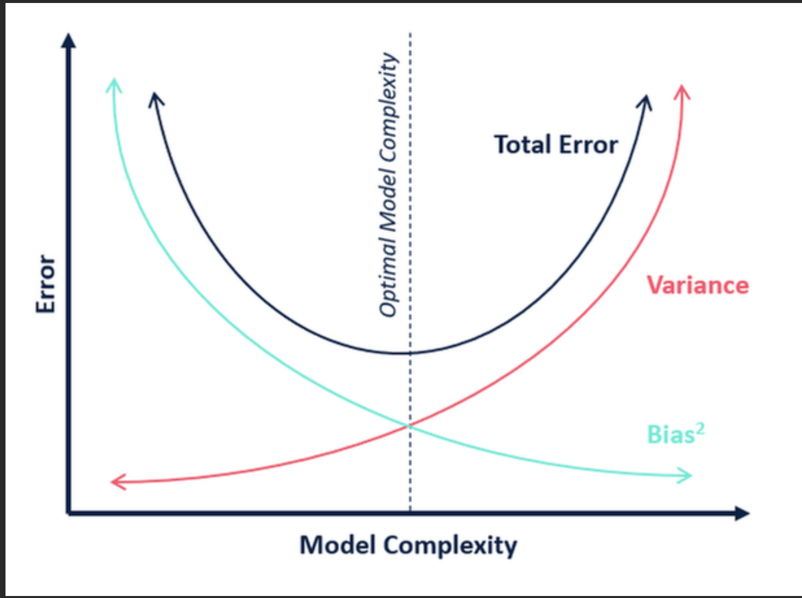


Overfitting
(High Variance)

Model Fitting

(Bias-Variance Trade-Off)

- The bias and variance have inverse relationship, when one of them increase the other one will decrease.
- Our goal always to make generalized model, to do so we need to find the optimal solution.
- The optimal solution is the balance of the variance and bias, it's where the underfitting and overfitting are avoided.



Model Fitting

(Solutions)

- These some techniques help to solve the underfitting and overfitting problems (not limited):
  - Cross Validation
  - Features Selection
  - Regularization
  - Dimensionality Reduction
  - Ensemble Techniques

Index

Supervised Learning

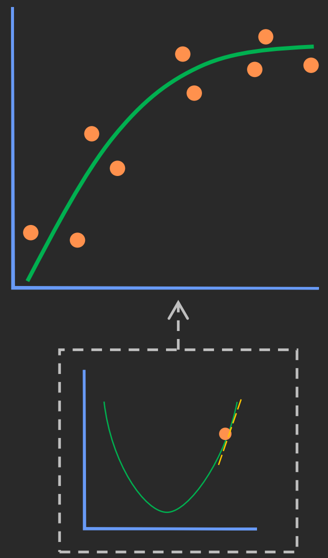
- Fitting Algorithms
- Regression
- Classification

Supervised Learning

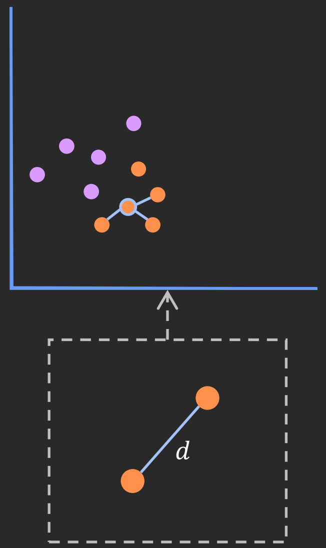
(Fitting Algorithms)

- Every machine learning algorithm is built based on another algorithm to train and fit the pattern of the data, there are three types of fitting algorithms:
  - Gradient Descent Based Algorithms: models use the gradient decent algorithm to find the optimal solution by finding the global minimum
  - Distance-Based Algorithms: models use distance to fit, there are three distances metrics
  - Tree-Based Algorithms: models use tree to fit

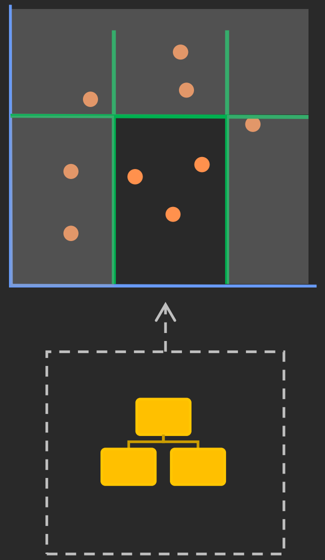
Use Gradient Decent



Use Distance



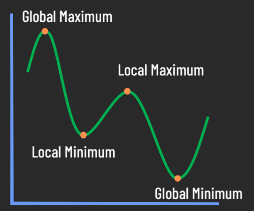
Use Tree



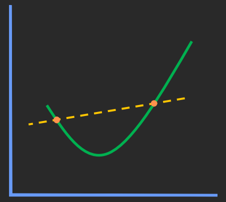
Supervised Learning

(Fitting Algorithms)

- Gradient Decent: is an optimization algorithm or cost function that finds the best values of the parameters(weights)
  - Optimization: is maximizing the objective or minimizing the cost, and here means minimizing the error/loss.
  - Global Minimum: is the point of the most minimum loss of the function, where is the best parameters are found.
  - Local vs Global Maximum/Minimum:



  - Draw any two points on the function and join them with a line, then if the line is always above the curve, then the function is called convex function.
  - Also we can called the function is convex if the local minimum is also the global minimum. Convexity

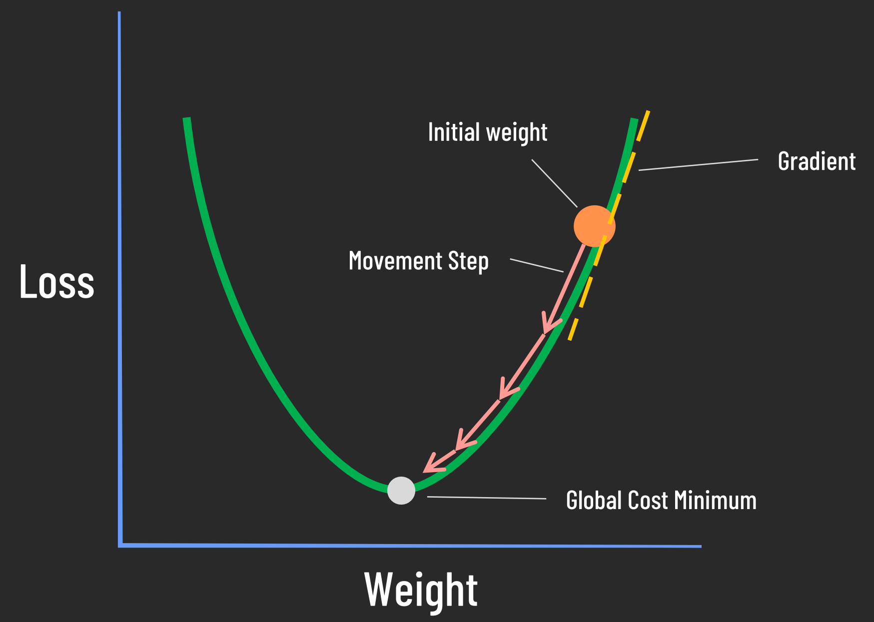


  - The goal of the gradient decent is to find the best parameters(weights) that minimize the loss function, and doing that by finding the global minimum.
  - Gradient decent finds the global minimum by searching, moving big steps if it's so far and baby step if it's so close.
  - The factor that responsible for the size of the step that gradient decent move is called learning rate

Supervised Learning

(Fitting Algorithms)

- Gradient Decent:
  - Algorithm Steps:
    1. Initialize weights with random values
    2. Calculate the gradient of the loss function with respect to the weights
    3. Update the weights in the direction of the optimal weights, using the learning rate
       - $New\ Weight = Old\ Weight\ - Learning\ Rate\ \times Gradient$
    4. Calculate the new loss function with the new weights using all the data
    5. Repeat steps 2, 3, 4 until the loss function reaches its minimum



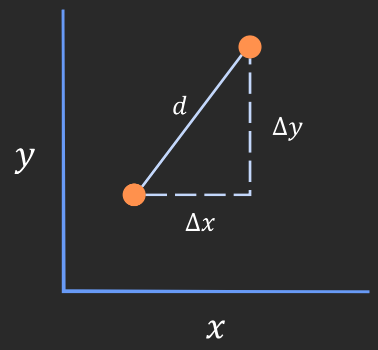
[Visual Theory](#) [Visual Theory & Code Tutorial](#)

Supervised Learning

(Fitting Algorithms)

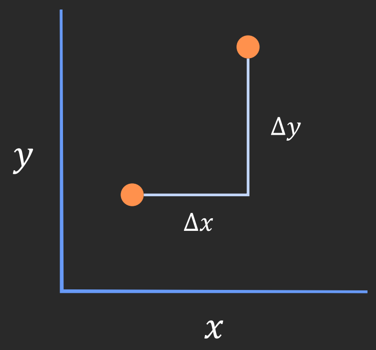
- Distance Measures:

Euclidean Distance



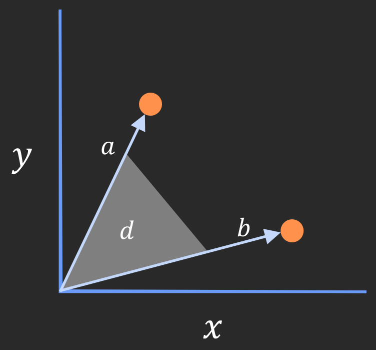
$$d = \sqrt{\Delta x^2 + \Delta y^2}$$

Manhattan Distance



$$d = |\Delta x| + |\Delta y|$$

Cosine Distance



$$d = \cos(\theta) = \frac{a \cdot b}{\|a\| \cdot \|b\|}$$

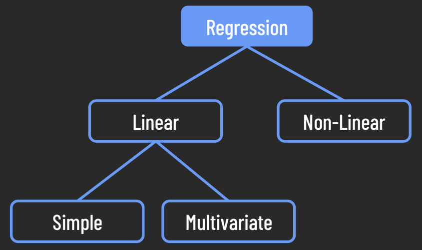
Supervised Learning

(Regression)

- Regression Analysis: consists of a set of machine learning methods that allow us to predict a continuous dependent variable (y) based on the value of one or multiple Independent variables (x).
- There are two types of Regression, Linear Regression and Non-Linear Regression, We will focus on Linear Regression.
- The Regression called Linear Regression when it has a linear relationship between its parameters.
- Linear Regression separate into to two categories:
  - Simple Regression: where is the problem have one independent variable.
  - Multivariate(Multiple Variables) Regression: where is the problem have two or more independent variables.

Y: *Dependent Variable*
x: *Independent Variable*
b: *Parameter*
ϵ: *Random Error*

$$Y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n + \epsilon$$



- Linear Regression Assumptions: this assumptions are needed to check if we want to apply linear regression
  - Linearity: there is need to be a linear relationship between the dependent and the independent variables.
  - No Multi-Collinearity: which means the independent variables should not be correlated.
  - Independence of Errors: the errors/residuals should have not any correlation with each other.
  - Normality of Error Distribution: which means the errors is need to be normally distributed.
  - Homoscedasticity: means the variance and errors needs to be constants .

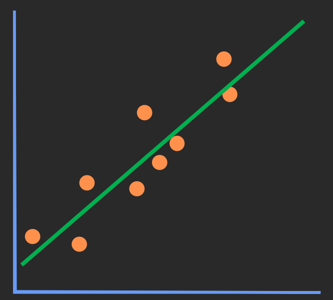
Linear Regression Assumptions
Regression (Intro)

Supervised Learning

(Regression)

- All Regression models are based on gradient descent, it used as cost function.
- Regression Models:

Linear Regression



$$Y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n + \epsilon$$

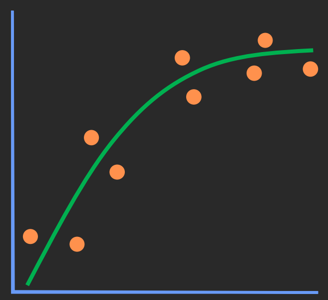
[Visual Theory]

[Theory]

[Code Tutorial (Simple)]

[Code Tutorial (Multivariate)]

Polynomial Regression



$$Y = b_0 + b_1x + b_2x^2 + \cdots + b_nx^n + \epsilon$$

[Theory]

[Theory 2]

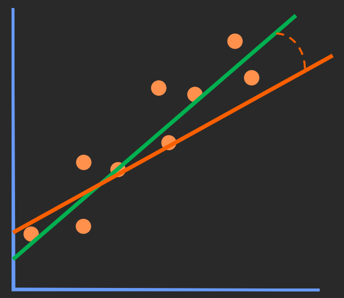
[Code Tutorial]

Supervised Learning

(Regression)

- **Regularization:** is techniques are used in the linear regression models to prevent underfitting and overfitting by adding penalty to the cost function.
- **Regularization Techniques:**

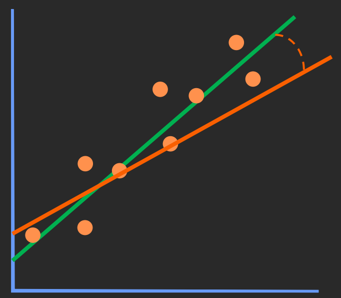
Ridge Regression (L2)



- Useful when we have many variables with relatively smaller data samples.
- It reduce the parameters close to zero, but not exactly zero.

$$\underbrace{Cost = Loss + \lambda \times \sum_{i=1}^{n} b_i^2}$$

Lasso Regression (L1)



- Preferred when we are fitting linear model with fewer variables.
- It reduce the parameters to exact zero.
- Helpful in features selection.

$$Cost = Loss + \lambda \times \sum_{i=1}^{n} |b_i|$$

Penalty

Visual Theory Theory & Code Tutorial Theory & Code Tutorial 2

Supervised Learning

(Regression)

- **Error Metric:** is a measurement of the overall performance, we use it to judge the model is good or bad.
- **Loss Function:** is a function that measures the loss/error of the model, it is used with the cost function in the training to optimize the model.
- **Error Metrics:**

  - $Mean\ Absolute\ Error\ (MAE) = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$
  - $Mean\ Squared\ Error\ (MSE) = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$
  - $Root\ MeanS\ quared\ Error\ (RMSE) = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$
  - $R\ Squared\ (R^2) = 1 - \frac{Sum\ Squared\ Error\ (SSE)}{Sum\ Squared\ Total\ (SST)} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}$

- Some error metrics are also considered as loss functions like MAE and MSE.

Error Metrics

Supervised Learning

(Classification)

- Classification algorithm is a supervised learning technique that is used to identify the class/category of new observations on the basis of training data
- The algorithm which implements the classification on a dataset is known as a classifier
- Based on learning the classification can be divided into two types of learners:
  - Lazy Learner: is Learner firstly stores the training dataset and wait until it receives the test dataset. In Lazy learner case, classification is done on the basis of the most related data stored in the training dataset, it takes less time in training but more time for predictions.
    Example: KNN algorithm.
  - Eager Learner: is Learner develop a classification model based on a training dataset before receiving a test dataset. Opposite to Lazy learners, Eager Learner takes more time in learning, and less time in prediction.
    Examples: Decision Trees, Naive Bayes
- Based on classification problem the classification can be divided into three types of classifications:
  - Binary classification: when the classification problem has only two possible outcomes, like: 1/0 True/False spam/not-spam ...
  - Multi-Class classification: when the classification problem has more than two outcomes, like: red/green/blue cat/dog/bird/bear/lion ...
  - Multi-Label classification: when the classification problem has more than label/target, for example: when predicting if is there a dog in the picture(True/False) and also is it there a cat in the picture(True/False), there is two labels here one for the dog and one for the cat
- There are two types of classification models:
  - Linear Models
  - Non-Linear Models

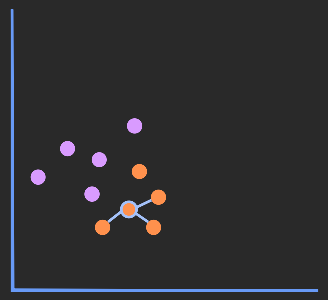
[Classification (Intro)]

Supervised Learning

(Classification)

- **Classification Models:**

K Nearest Neighbor (KNN)



Base Algorithm	Distance
Model Type	Non-Linear

About Model

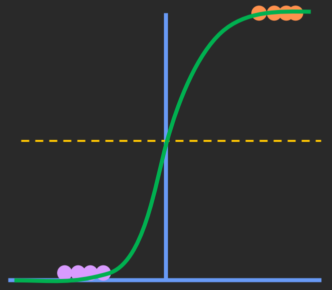
- KNN is based on features similarity.
- It classifies a data point based on how its neighbors are classified.
- k is a paramater that refers to the number of nearest neighbors to include in the majority voting process.
- odd value of k is selected to avoid confusion between two classes.

Visual Theory & Code Tutorial

Visual Theory & Code Tutorial 2

Code Tutorial

Logistic Regression



Base Algorithm	Gradient Descent
Model Type	Linear

About Model

- Logistic regression use sigmoid function to range the target between 0 and 1
- $sigmoid(z) = \frac{1}{1+e^{-z}}$
- Usually Logistic Regression used with binary classification problems.
- The middle line that cross the prediction line is called threshold, if the point above the threshold it will consider as 1, and if the point below the threshold it will consider as 0.

Visual Theory (Binary)

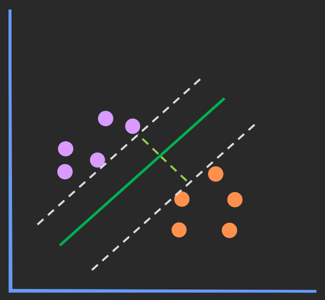
Visual Theory & Code Tutorial (Binary)

Visual Theory & Code Tutorial (Binary) 2

Code Tutorial (Multi-Class)

Sigmoid Explanation

Support Vector Machine (SVM)



Base Algorithm	Gradient Descent
Model Type	Linear

About Model

- Margin is the distance between the two boundaries (which is the green dashed line).
- Support Vectors are the points on the edge of the boundaries.
- There are two types of margin:
  - Hard Margin
  - Soft Margin
- When the data can not be separate linearly, kernel trick is used, kernals:(Poly, RBF, Linear, …).

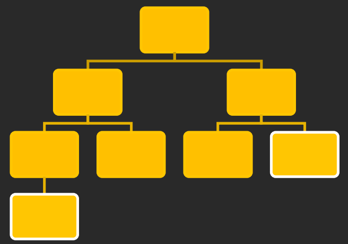
Visual Theory

Visual Theory 2

Visual Theory & Code Tutorial

Visual Theory & Code

Decision Tree



Base Algorithm	Tree
Model Type	Non-Linear

About Model

- Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations. It determines how a decision tree chooses to split data.

$$E = -\sum_{i=1}^{N} p_i \times log_2(p_i)$$

- Information Gain is a measure of how much information a feature provides about a class.

$$IG = E_{parent} - E_{children}$$

Visual Theory

Visual Theory 2

Visual Theory & Code Tutorial

Visual Theory & Code Tutorial 2

Supervised Learning

(Classification)

- **Evaluation:**
  - Confusion Matrix: is a performance measurement for the classification problem
  - We say True when the prediction is match the actual in one of the following two cases (False otherwise):
    - Predicted target class and the actual is the target class
    - Predicted class that not the target and the actual is any class that not the target
  - We say Positive when the prediction is the target class (Negative otherwise)

Binary Confusion Matrix

Predicted \ Actual	1	0
1	TP	FP
0	FN	TN

Multi-Class Confusion Matrix

Predicted \ Actual	A	B	C	D
A	TN	FN	TN	TN
B	FP	TP	FP	FP
C	TN	FN	TN	TN
D	TN	FN	TN	TN

Focusing on Class B

- = • True Positive(TP): when the prediction is the target class that we focusing on, and the actual is the target class
- = • True Negative(TN): when the prediction is not a target class, and the actual is also not a target class
- = • False Positive(FP) (type I error): when the prediction is the target class, and the actual is any class that not a target
- = • False Negative(FN) (type II error): when the actual is the target class, and the prediction is any class that not a target

Supervised Learning

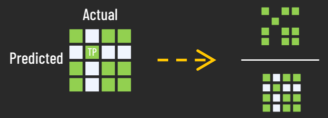
(Classification)

	Actual 1	Actual 0
Predicted 1	TP	FP
Predicted 0	FN	TN

Predicted \ Actual	A	B	C	D
A	TN	FN	TN	TN
B	FP	TP	FP	FP
C	TN	FN	TN	TN
D	TN	FN	TN	TN

- **Evaluation:**
  - **Performance Metrics:**
    - Accuracy: measures the overall true predictions for a specific class, good for balanced data.

$$\text{Accuracy} = \frac{\textit{true predictions}}{\textit{total predictions}} \qquad \text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$



    - Precision: measures how many specific class predictions are correct for that class, good with imbalanced data.

$$\text{Precision} = \frac{\textit{correct target class prediction}}{\textit{total target class predictions}} \qquad \text{Precision} = \frac{TP}{TP+FP}$$

    - Recall (Sensitivity): measures how many specific class actual did we correctly predict for that class, good with imbalanced data.

$$\text{Recall} = \frac{\textit{correct target class prediction}}{\textit{total target class actual}} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

    - F1: is a harmonic mean of precision and recall, it gives a combined idea about precision and recall metrics, good with imbalanced data.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

    - Precision and Recall have a trade-off relationship.
    - There are also Receiver Operating Characteristics (ROC) and Area Under Curve (AUC), these two measures are graphical way to know the diagnostic ability of a classifier, usually used with a binary classification problems. ROC & AUC

Visual Theory (Multi-Class)

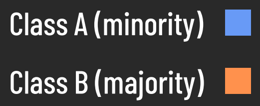
Visual Theory (Binary)

Visual Theory & Code Tutorial (Binary)

Supervised Learning

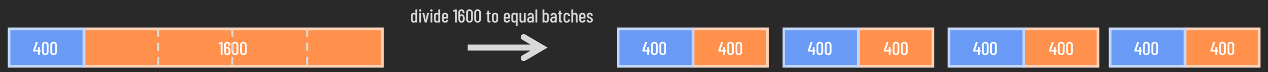
(Classification)

- Imbalanced Data:
  - Imbalanced Data: is a data with a significant difference in frequency of outcomes(classes)
  - It is better to use Precision or Recall or F1 as a performance metric with imbalanced data, or balancing the data with some techniques
  - Common imbalanced data handling techniques (not limited):
    - Collecting More Data
    - Oversampling: is increasing the minority class to be equal to the majority class



    - Undersampling: is decreasing the majority class to be equal to the minority class

    - Ensembling: divides majority class into batches equally to the minority class, and train every new sample on its own then do ensembling



    - Weighting Classes: assign high weight to minority class and low weight to majority class, the difference in weights will influence the classification of the classes during the training phase, the whole purpose is to penalize the misclassification made by the minority class

  - We can combine two or more techniques like oversampling and undersampling
  - There is no better technique, the best way is to understand your data and your problem and figure out how to deal with your imbalanced data

[Theory & Code Tutorial]   [Visual Theory & Code Tutorial]   [Code Tutorial]

Index

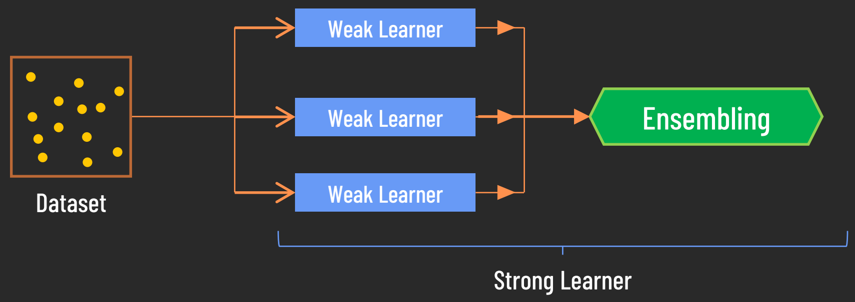
Enhancement

- Ensembling
- Hyperparameters Tuning

Enhancement

(Ensembling)

- Ensemble is a technique that combine individual models together to improve the stability and predictive power of the model
- The model that perform not so well by themselves either because they have a high bias or high variance called weak learner
- The main principle behind ensemble learning is to group weak learners together to form one strong learner that achieves better performance



- Ensemble learning models can be categorized into two types based on the choice of weak learners:
    - Homogeneous: all weak learners are from the same type, for example: all models are decision tree
    - Heterogeneous: different models are used
- There are two groups of ensemble methods:
    - Sequential Ensemble Methods: Where the models depend on each other
    - Parallel Ensemble Methods: Where the models do not depend on each other

Ensemble Visual Theory Ensemble Theory

[(Bagging & Boosting): Visual Theory & Code Tutorial]

[(Bagging & Boosting): Visual Theory & Code Tutorial 2]

[(Bagging & Boosting): Visual Theory]

[(Stacking): Visual Theory]

[(Stacking): Code Tutorial]

Enhancement

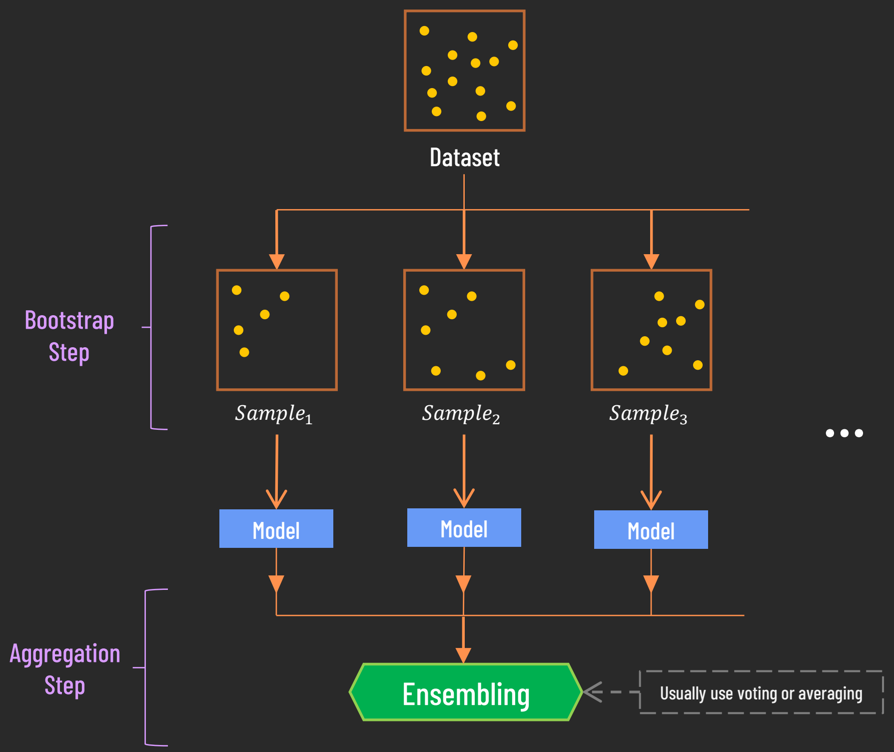
(Ensembling)

- Ensemble methods (not limited):
  - Majority Voting: every model makes a prediction then votes for each test instance and the final output prediction is the one that receives more votes
  - Weighted Voting: same as majority voting, but we can increase the importance of one or more models by assigns weights
  - Simple Averaging: take the average of the predictions of all models
  - Weighted Averaging: same as simple averaging, but we can increase the importance of one or more models by assigns weights
  - Bootstrap aggregation (Bagging): combine homogeneous models to reduce variance, its workflow as following:
    1) bootstrapping by resampling with replacement which is selecting a random samples of the original data with replacement, with replacement means every sample is selected it can be reselected again
    2) fit and predict for every selected sample with the model
    3) aggregate models by apply voting or averaging ensemble methods on the predictions of all models
  - Boosting: train homogeneous models sequentially to reduce bias and produce a strong learner, its workflow as following:
    1) The base learner takes all the distributions and assign equal weight or attention to each observation
    2) If there is any prediction error caused by first base learning algorithm, then we pay higher attention to observations having prediction error, then we apply the next base learning algorithm
    3) Iterate Step 2 till the limit of base learning algorithm is reached or higher accuracy is achieved
    4) Finally, it combines the outputs from weak learners and creates a strong learner which eventually improves the prediction power of the model
  - Stacking: is an ensemble machine learning algorithm that learns how to best combine the predictions from multiple well-performing machine learning models (heterogeneous)
- Usually voting used for classification and averaging used for regression.

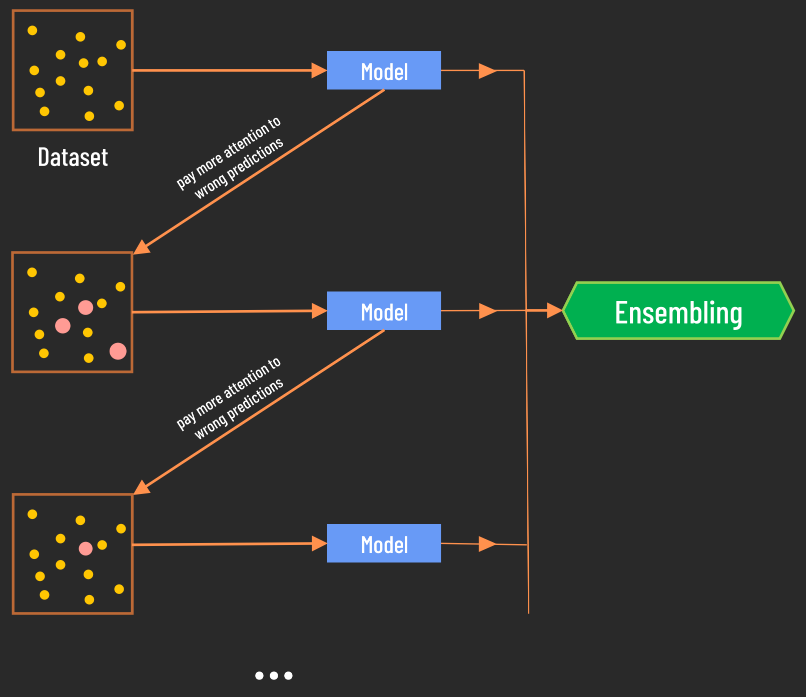
Enhancement

(Ensembling)

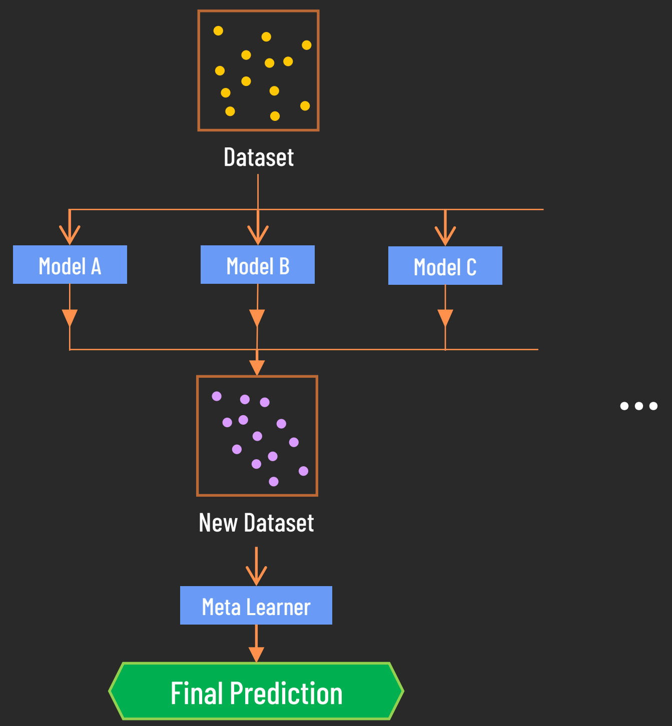
Bagging



Boosting



Stacking



Homogeneous   Parallel   Reduce Variance (solve overfitting)

Algorithms:
- Random Forest

Homogeneous   Sequential   Reduce Bias (solve underfitting)

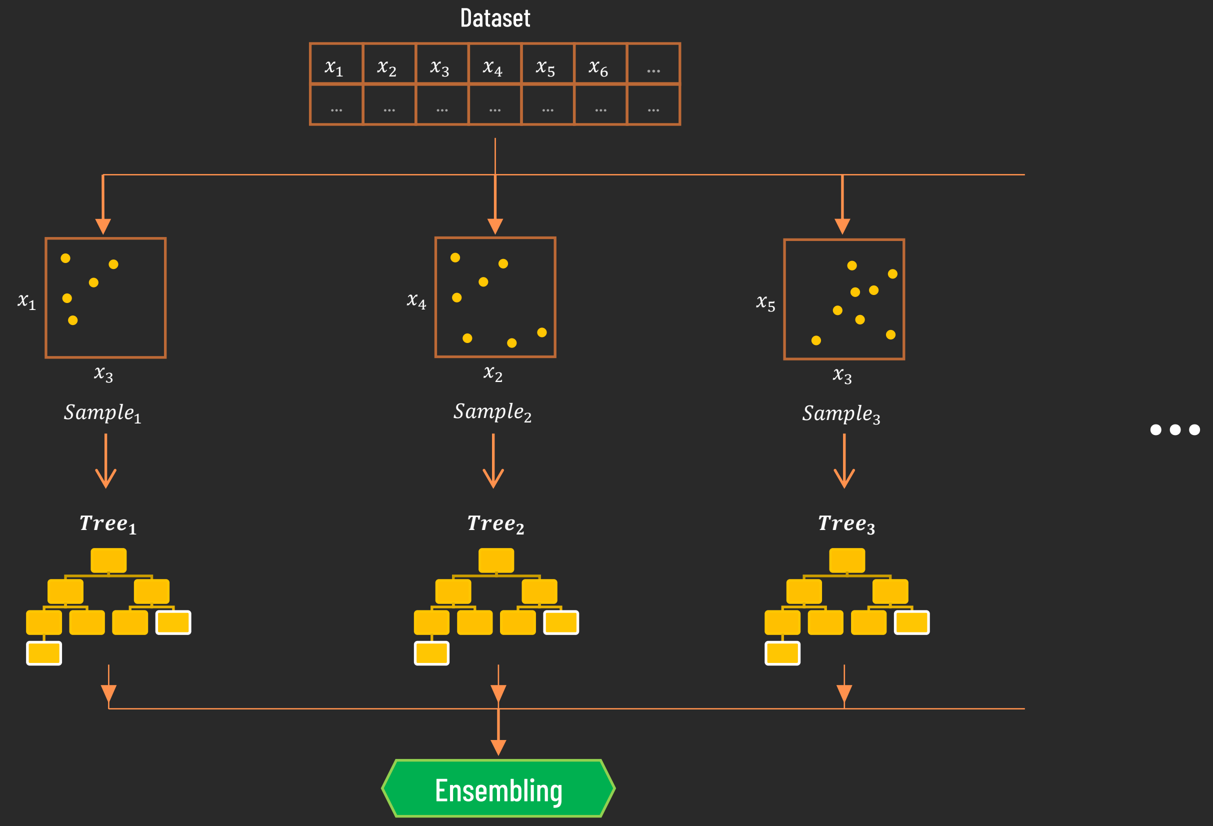
Algorithms:
- AdaBoost: Theory   Code Tutorial
- Gradient Boost: Theory   Code Tutorial
- XGBoost

Heterogeneous   Parallel

Enhancement

(Ensembling)

- Random Forest
  - Random Forest is the implementation of bagging that use decision tree as a weak learner with one additional difference
  - The difference is: in the bootstrap step it selects random features samples with replacement as well as observations



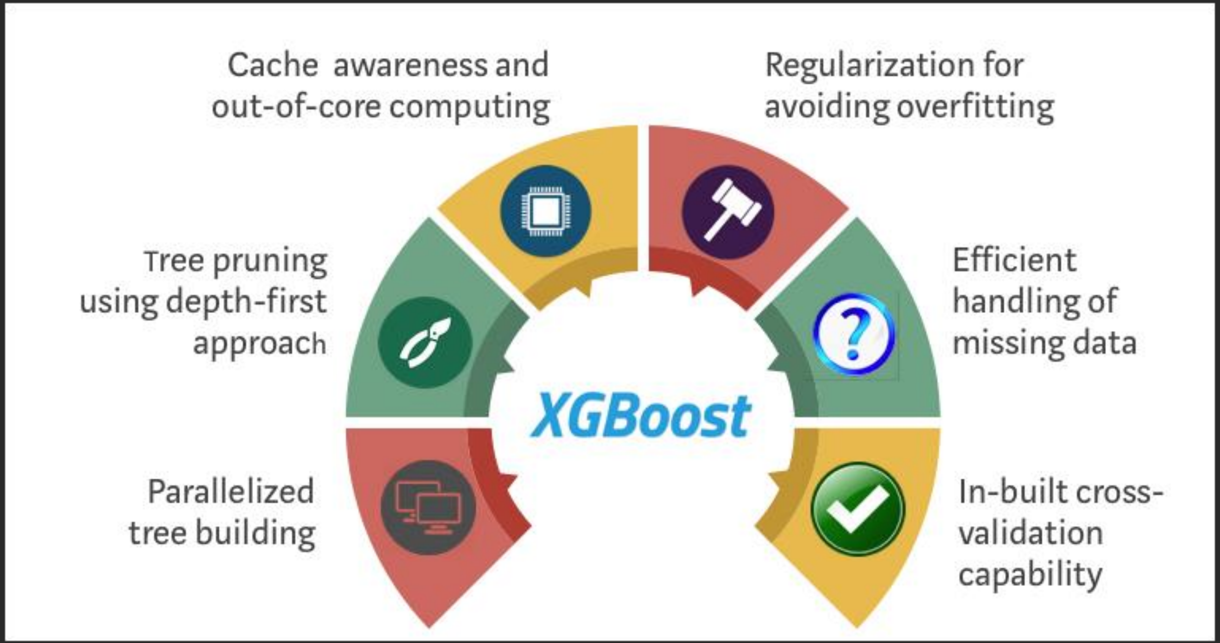
Visual Theory

Visual Theory & Code Tutorial

Enhancement

(Ensembling)

- XGBoost:
  - Stands for Extreme Gradient Boost

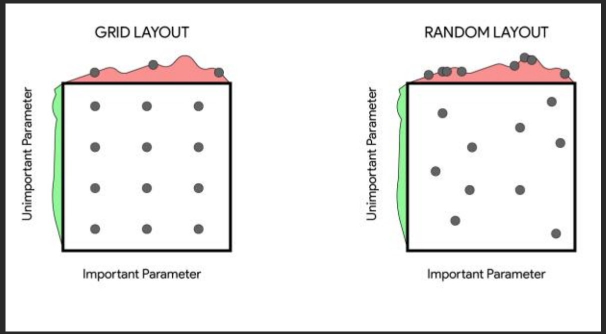


Overview & Code Tutorial   Visual Theory

Enhancement

(Hyperparameters Tuning)

- There are two types of parameters which are parameters and hyperparameters
- Parameters are the values learned during training from the historical data
- Hyperparameters are configuration variables that is external to the model, it is defined manually before the training
- Hyperparameters tuning is a technique that used to find the optimal hyperparameters for the model that give the best results
- Hyperparameters Tuning Techniques:
  - Grid Search: try all the combination of the hyperparameters set
  - Random Search: try random combination of the hyperparameters set



Theory   Theory & Code Tutorial   Code Tutorial

Index

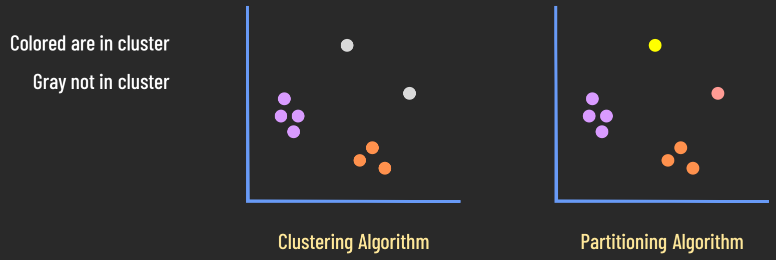
Unsupervised Learning

- Clustering
- Dimensionality Reduction

Unsupervised Learning

(Clustering)

- Clustering is the grouping of data points that are close or similar to each other to be in clusters(groups).
- Clustering Algorithm groups the similar data points into clusters(groups), but may there are data points not assigned into any cluster.
- Partitioning Algorithm is considered as clustering but in partitioning algorithm data points must be assigned to a cluster.



Unsupervised Learning

(Clustering)

- Clustering Models:

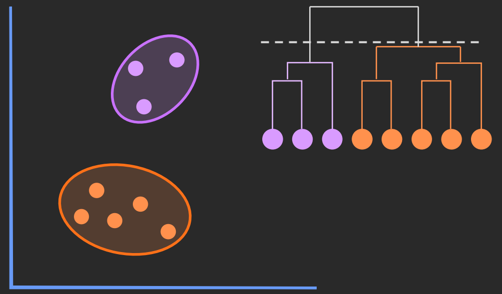
K Means

Partitioning Algorithm

[Visual Theory]

[Visual Theory & Code Tutorial]

Hierarchical Agglomerative Clustering (HAC)



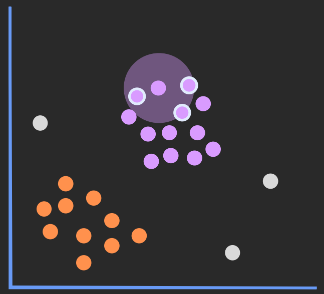
Partitioning Algorithm

[Theory]

[Visual Theory]

[Code Tutorial]

Density-based spatial clustering of applications with noise (DBSCAN)



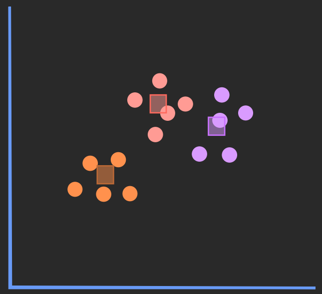
Clustering Algorithm

[Visual Theory]

[Theory & Code Tutorial]

[Code Tutorial]

Mean Shift



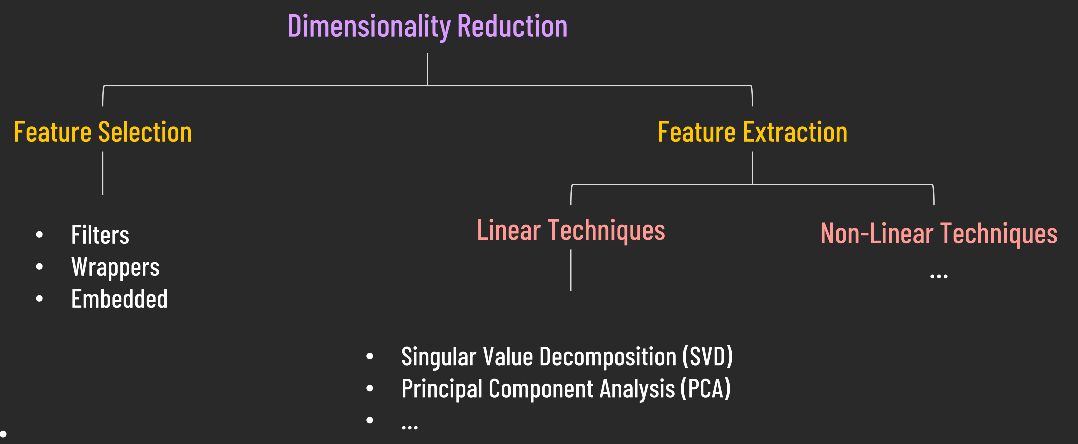
Partitioning Algorithm

[Theory & Code Tutorial]

Unsupervised Learning

(Dimensionality Reduction)

- Dimensionality Reduction refers to techniques for reducing the number of input variables in dataset.
- Dimensionality Reduction Advantages:
  - Less dimensions for a given dataset means less computation or training time
  - Redundancy is removed after removing similar entries from the dataset
  - Space required to store the data is reduced
  - Makes the data easy for plotting 2D and 3D plots
  - It helps to find out the most significant features and skip the rest
  - Leads to better human interpretations



- There is two approaches to reduce dimensions of the data, which are:
  - Feature Selection: see its section for more details.
  - Feature Extraction: It has linear techniques and non-linear techniques (see its section for more details)
- We will cover only two techniques in feature extraction, which are:
  - Singular Value Decomposition (SVD)
  - Principal Component Analysis (PCA)

Dimensionality Reduction Explanation (Intro)

Unsupervised Learning

(Dimensionality Reduction)

- **Singular Value Decomposition (SVD):**
  - SVD is a dimensionality reduction technique that applies matrix factorization by decomposing the original matrix into three matrices
  - SVD is used in different parts and applications in machine learning, and we will see later Latent Semantic Analysis(LSA) which it use the SVD for topic modeling
  - SVD decompose the original matrix into three matrices:
    - U : Left singular vectors
    - $\sum$: Diagonal matrix (all off-diagonal entries are 0) of singular values
    - V : Right singular vectors

$$A_{m \times n} = U_{m \times r} \textstyle\sum_{r \times r} {V^T}_{n \times r}$$

Left Singular Vectors (eigenvectors)

Singular Values (eigenvalues)

Right Singular Vectors (eigenvectors)

$$\underbrace{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}_{A_{m \times n}} = \underbrace{\begin{bmatrix} & \\ & \\ & \end{bmatrix}}_{U_{m \times r}} \times \underbrace{\begin{bmatrix} & \\ & \end{bmatrix}}_{\sum_{r \times r}} \times \underbrace{\begin{bmatrix} & & \\ & & \end{bmatrix}}_{{V^T}_{n \times r}}$$

[Visual Explanation 1]

[Mathematical Explanation]

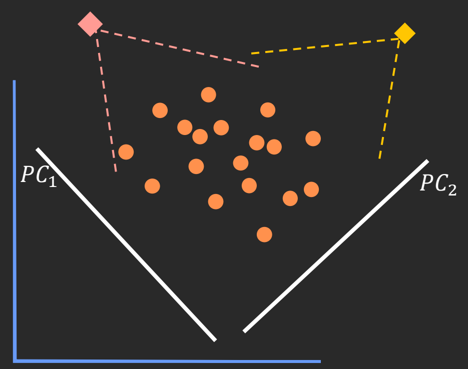
[Code Explanation]

[Visual Explanation 2]

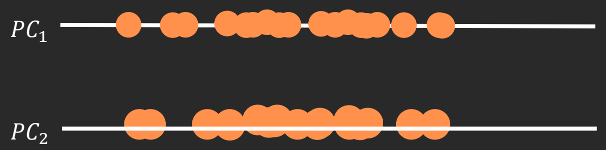
Unsupervised Learning

(Dimensionality Reduction)

- Principal Component Analysis (PCA):
  - PCA is a technique for reducing the dimensionality of dataset, increasing the interpretability but at the same time minimizing information loss
  - Each principal component is a linear combination of all variables, with a weight on each variable
  - PCA is looking for linear relationships between variables, if they're related in some different way, PCA won't help us
  - Standardization is necessary to do PCA in appropriate way, otherwise PCA will just pick out variables that are on larger scales (higher variance)
  - Principle components are orthogonal
  - The priority of principle components decrease as their numbers increase



After PCA



Visual & Code Explanation

Visual Explanation 1

Visual Explanation 2

Mathematical Explanation

Code Explanation

Time Series

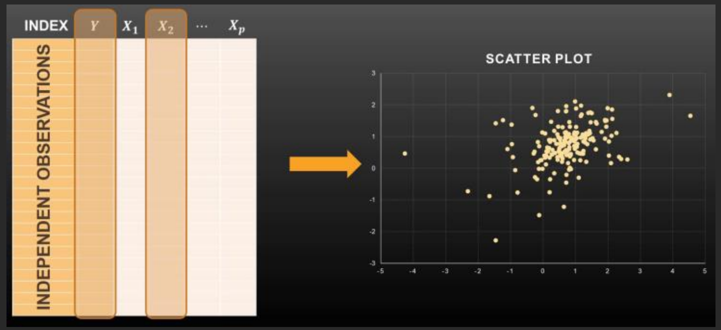
- Definitions
- Introduction
- Components
- Decomposition
- Stationarity
- ARIMA

Time Series

(Definitions)

- Time Series is a series of data points indexed in time order
- Time Series Analysis: is the use of methods and techniques for analyzing time series data in order to extract meaningful statistics and other characteristics of the data
- Time Series Forecasting: is the use of methods and techniques to predict events or future values through a sequence of time
- Time Series vs Cross-Sectional:
  - Time Series Data: is a set of ordered data values observed at points in time
  - Cross-Sectional Data: is a set of data values at a fixed point in time

Time Series



Cross-Sectional

What is Time Series? Time Series?

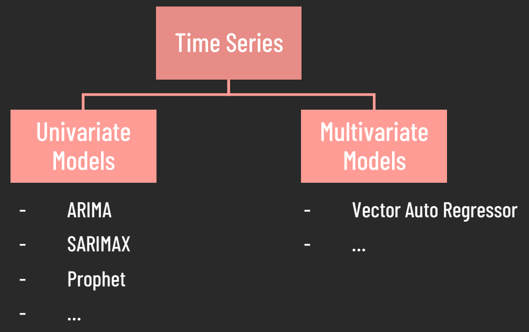
Time Series

(Introduction)

- Time Series data should be regular time intervals (monthly, weekly, annually, …)
  - such that: $t_2 - t_1 = t_3 - t_2 = \ … \ = t_n - t_{n-1}$
- White Noise: a series is called white noise if it purely random in nature, the mean is 0 and the variance is constant, which means the series has no pattern [White Noise]
- Lag: is shifting the target n times [Lag]

		Lag = 1	Lag = 2	Lag = 3
t	Y_t	Y_{t-1}	Y_{t-2}	Y_{t-3}
1	100	–	–	–
2	75	100	–	–
3	155	75	100	–
4	60	155	75	100
…	…	…	…	…

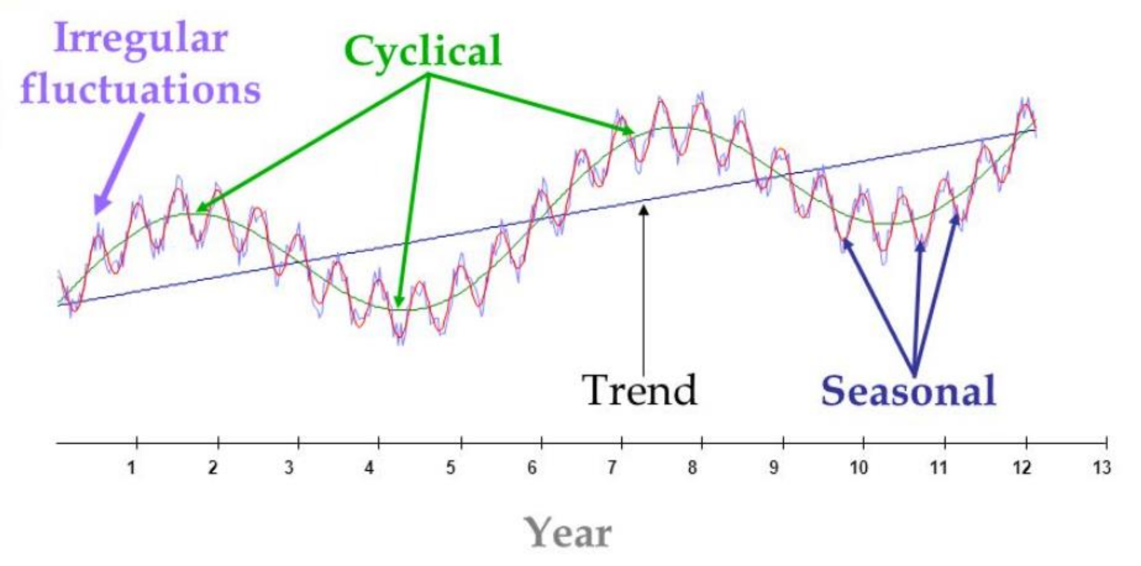
- There are two types of Time Series Models:
  - Univariate Models: predicting single target
  - Multivariate Models: predicting multiple targets



Time Series

(Components)

- Time Series data can exhibit a variety of patterns, so it is often helpful to split a time series into components, each representing an underlying pattern category
- Time Series is Composed of three main components: (there is a fourth component which is the cycle but we will not mention it here)
  - Trend: is a non-repeating long term change in the series over time
  - Seasonality: is the repeated behavior regularly(fixed intervals) that occurs over the short term within a year or less
  - Reminder (Noise/Irregularity): is the unexpected situations/events/scenarios and spikes in a short time span

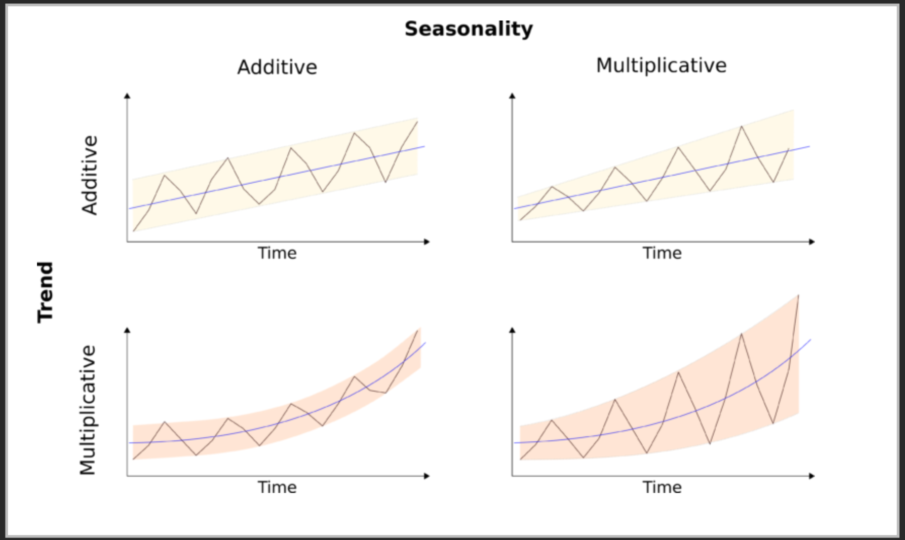


[Time Series Components]

Time Series

(Decomposition)

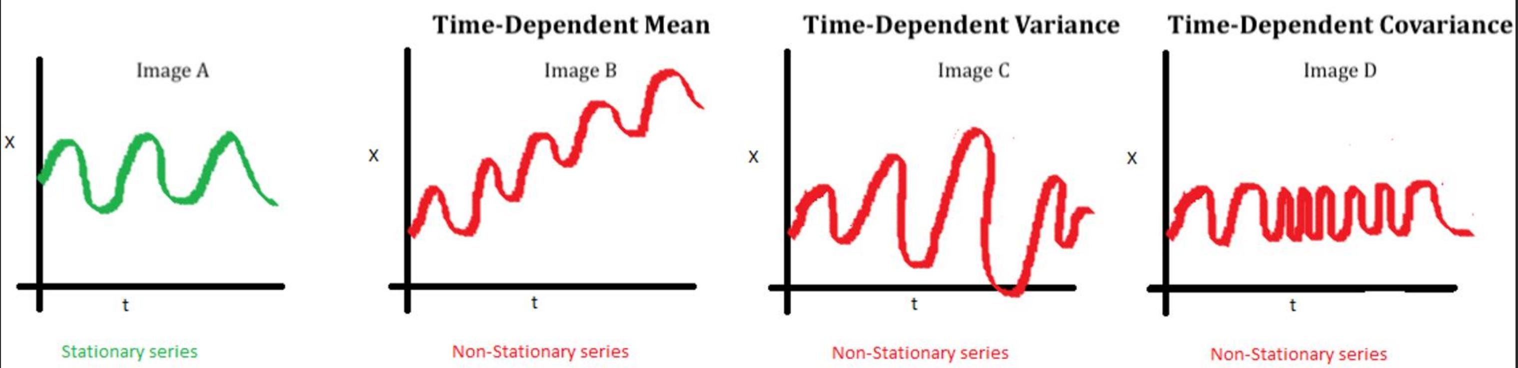
- Time Series Decomposition: is a statistical task that decompose time series into its components (trend(T), seasonality(S), and reminder(R))
- There are two types of decomposition:
  - Additive Decomposition: it decompose the time series into the sum of its components $Y_t = T_t + S_t + R_t$
  - Multiplicative Decomposition: it decompose the time series into the product of its components $Y_t = T_t \times S_t \times R_t$



Time Series

(Stationarity)

- Stationarity: is the consistency of the distributions in the time series
- the time series is called stationarity if the mean, variance, and covariance are constant



- Some time series models require the time series to be stationarity to apply (like: ARIMA)
- If the time series is non-stationarity we can convert it to stationarity by differencing

Differencing

t	Y_t	Y_{t-1}	$Y_t - Y_{t-1}$
1	5	–	–
2	10	5	5
3	15	10	5
4	20	15	5
…	…	…	…

Visual Theory Theory Code

Time Series

(ARIMA)

- Auto Regressive (AR): is a model that forecast a series based on the past target values   Auto Regressive
  - $Y_t = \omega + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3} + \ \dots + \phi_p Y_{t-p} + \epsilon_t$
  - p is the number of the target lags
- Moving Average (MA): is a model that forecast a series based on the past errors   Moving Average
  - $Y_t = \omega + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \theta_3 \epsilon_{t-3} + \ \dots + \theta_q \epsilon_{t-q} + \epsilon_t$
  - q is the number of the error lags
- The I term in ARIMA stands for integrated, which refers to the differencing to make the time series stationarity
- A series which is stationary after being differentiated once is said to be integrated of order(d) 1 and is denoted by $I(1)$
- To perform the ARIMA model the time series should be stationarity, and that why the I term is there for

- $ARIMA(p, d, q)$: p is for AR, d is for I, q is for MA   ARIMA
  - There are some different techniques to decide the values of p and q
    - Autocorrelation Function (ACF)
    - Partial Autocorrelation Function (PACF)
    - Minimum Information Criterion (MINIC)
    - Squared Canonical (SCAN)
    - Extended Sample Autocorrelation Function (ESACF)

AR vs MA   Code Tutorial   Code Tutorial 2

Index

Natural Language Processing (NLP)

- Introduction
- Text Preprocessing Techniques
- Text Formats
- Word Embedding
- Text Similarity Measures
- Sentiment Analysis
- Topic Modeling

Natural Language Processing (NLP)

(Introduction)

- NLP is a field that focus on how the machine deals with natural languages and text.
- Main NLP Topics:
  - Regular Expressions(regex): is a sequence of characters that specifies a search pattern in text, it used to handle text with some patterns. Code Tutorial
  - Text Preprocessing Techniques
  - Text Formats
  - Word Embedding
  - Text Similarity Measures
  - Applications of NLP:
    - Sentiment Analysis
    - Topic Modeling

Natural Language Processing Explanation (Intro)

Natural Language Processing (NLP)

(Text Preprocessing Techniques)

- Tokenization: splitting raw text into small indivisible units for processing.
  - These units can be: words, sentences, n-grams, other characters defined by regular expressions
- Clean Data:
  - Remove Capital Letters: convert all text to lower case.
  - Remove Punctuation: , ! $ ( ) * % @ …
  - Remove Numbers
  - Remove Stop Words: stop words are words that have very little semantic value (no meaning).
  - …
- Stemming: It is also known as the text standardization step where the words are stemmed or diminished to their root/base form
  - For example: words like 'programmer', 'programming, 'program' will be stemmed to 'program'.
  - The disadvantage of stemming is that it stems the words such that its root form loses the meaning or it is not diminished to a proper English word
- Lemmatization: It stems the word but makes sure that it does not lose its meaning
  - Lemmatization has a pre-defined dictionary that stores the context of words and checks the word in the dictionary while diminishing
  - Lemmatization will only be performed when the given word has a proper part of the speech tag associated with it
- Part of Speech Tagging: labels each word as part of speech(nouns, verbs, adjectives, …)
- Chunking:
  - Named Entity Recognition: Identifies and tags named entities in text (people, places, organizations, phone numbers, emails, …).
  - Compound Term Extraction: Extract compound terms that has a meaning to be single term.
- Misspellings
- …

Theory & Code Explanation

Natural Language Processing (NLP)

(Text Preprocessing Techniques)

Tokenization

"Hi Mr. Smith! I'm going to buy some vegetables" → ['Hi', 'Mr.' 'Smith', 'I', 'am', 'going', 'to', 'buy', 'some', 'vegetables']

Stemming

drive	driv
drives	driv
driver	driv
drivers	driv
driven	driv

Lemmatization

drive	drive
drives	drive
driver	drive
drivers	drive
driven	drive

Part of Speech Tagging

"I want an early upgrade" →

Pronouns	Verb	Determines	Adjectives	Noun
I	want	an	early	upgrade

Named Entity Recognition

"Sundar Pichai, the CEO of Google Inc. is walking in the streets of California." → "Sundar Pichai (Person), the CEO of Google Inc. (Organization) is walking in the streets of California (Location)."

Compound Term Extraction

"black eyed peas" → "black_eyed_peas"

Natural Language Processing (NLP)

(Text Formats)

- Bag of Words (BoW): simplified representation of text, where each document is recognized as a bag of its words, grammar and words order are disregarded.
- There is two common text formats used in NLP:
  - Corpus:
    - Corpus is a collection of texts(documents)
    - Document is a collection of paragraphs
    - Paragraph is a collection of sentences
    - Sentence is a collection of tokens
    - Token may be a word or a collection of words

  - Document-Term Matrix: it's a matrix its rows are documents and its columns are terms, we can consider it as an implementation of the Bag of Words concept. Explanation

	$term_1$	$term_2$	$term_3$	$term_n$
Doc_1	1	1	4	…
Doc_2	0	3	0	…
Doc_3	0	0	1	…
Doc_n	…	…	…	…

Document-Term Matrix

Natural Language Processing (NLP)

(Word Embedding)

Word Embedding is a method or technique of converting words in our vocabulary into vectors.

- Frequency or Statistical based Word Embedding approaches: Count Vectorizer & TF-IDF Code Tutorial
  - Count Vectorizer: It creates a document-term matrix that its cells represent how many each term appears in each document.
  - Term Frequency-Inverse Document Frequency (TF-IDF): It creates a document-term matrix that its cells represent the importance of each term in each document.

$$Term\ Frequency(TF) = \frac{\boldsymbol{Term}\ Count\ in\ Documents}{Total\ \boldsymbol{Terms}\ in\ Document}$$

$$\boldsymbol{Importance}\ of\ term\ in\ specific\ document = TF \times IDF$$

$$Inverse\ Document\ Frequency(IDF) = \log\left(\frac{Total\ Documents + 1}{Documents\ Containing\ the\ \boldsymbol{Term} + \mathbf{1}}\right)$$

  - N-Grams Vectorization: similar to the count vectorizer technique, the count vectorizer is a special case of N-Grams Vectorization where n = 1
- Prediction based Word Embedding approaches: More Advance
  - Word2Vec: if interested here are the explanations: Theory Code
  - Glove: if interested here are the explanations: Explanation 1 Explanation 2
  - …

Natural Language Processing (NLP)

(Text Similarity Measures)

- Word Similarity:
  - Levenshtein Distance: Minimum number of operations to get from one word to another, Levenshtein operations are: Theory & Code
    - Deletions: Delete a character.
    - Insertions: Insert a character.
    - Mutations: Change a character.

Levenshtein Distance Example:
kitten --> sitting

kitten	→	sitten	1 letter changed (Mutation)
sitten	→	sittin	1 letter changed (Mutation)
sittin	→	sitting	1 letter inserted (Insertion)

Levenshtein Distance = 3

- Document Similarity:
  - Cosine Similarity: is a way to quantify the similarity between documents, measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. Theory & Code

	$term_1$	$term_2$	$term_3$	$term_4$
Doc_1	1	1	1	0
Doc_2	1	1	0	1

Document-Term Matrix

$$Doc_1 = [1, 1, 1, 0]$$
$$Doc_2 = [1, 1, 0, 1]$$

Value between 0 and 1
- Close to 0 : no similarity
- Close to 1 : there is similarity

$$Similarity = \cos(\theta) = \frac{Doc_1 \cdot Doc_2}{||Doc_1|| \cdot ||Doc_2||} = 0.667$$

Natural Language Processing (NLP)

Corpus

...	*Document*	...
...	I like NLP	...
...	Learning machine learning is great	...
...	NLP is great	...

Count Vectorizer →

TF-IDF →

Document-Term Matrix

	I	*like*	*NLP*	*machine*	*learning*	*is*	*great*
Doc_1	1	1	1	0	0	0	0
Doc_2	0	0	0	1	2	1	1
Doc_3	0	0	1	0	0	1	1

	I	*like*	*NLP*	*machine*	*learning*	*is*	*great*
Doc_1	0.457	0.457	0.586	0.0	0.0	0.0	0.0
Doc_2	0.0	0.0	0.0	0.277	0.554	0.183	0.183
Doc_3	0.0	0.0	0.586	0.0	0.0	0.302	0.302

Apply Cosine Similarity for every two documents →

$$\cos(\theta) = \frac{Doc_A \cdot Doc_B}{\|Doc_A\| \cdot \|Doc_B\|}$$

Document Similarity

Value between 0 and 1

- Close to 0 : no similarity
- Close to 1 : there is similarity

Natural Language Processing (NLP)

(Sentiment Analysis)

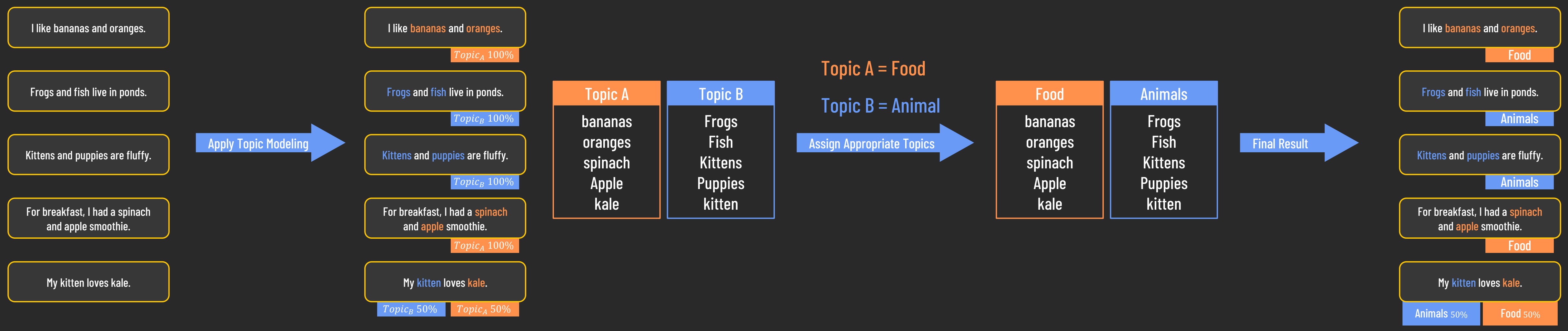
- Sentiment analysis is a technique in NLP used to detect some kind of attitudes in the text
- The simplest sentiment analysis task is the detection of the text if it positive or negative
- Python Package That Implement Simple Sentiment Analysis: TextBlob

Code Explanation   Theory Explanation   Visual & Code Explanation   Visual Theory Explanation

Natural Language Processing (NLP)

(Topic Modeling)

- The process of discovering topics that occur in a collection of documents.
- Techniques (not limited):
  - Latent Semantic Analysis (LSA)
  - Latent Dirichlet Allocation (LDA)



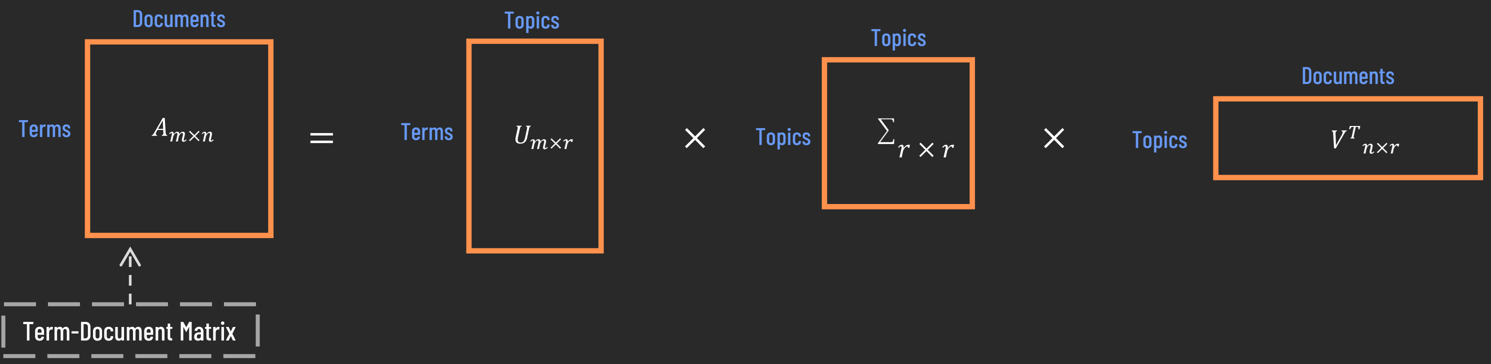
Topic Modeling (Intro)

Natural Language Processing (NLP)

(Topic Modeling)

Latent Semantic Analysis (LSA)

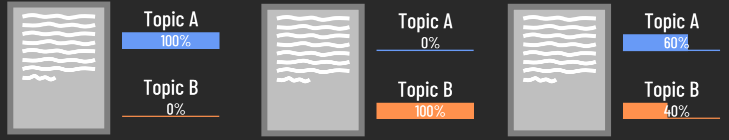
- It find the latent topics(hidden features) in every document by using the SVD(Singular Value Decomposition)
- Python Package: scikit-learn

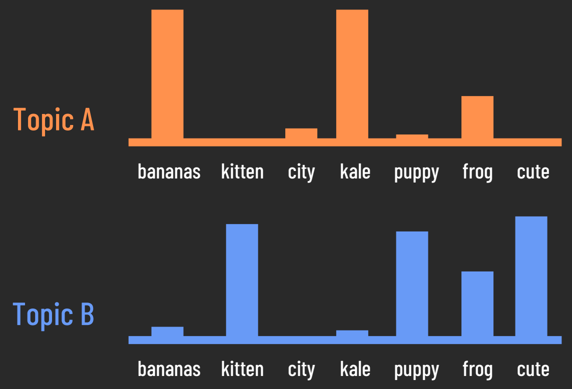


Code Explanation 1 Code Explanation 2 Theory Explanation Visual Theory Explanation

Latent Dirichlet Allocation (LDA)

- It find the latent topics(hidden features) in every document by assuming the following:
    - Every document consists of a distribution of topics
    - Every topic consists of a distribution of words
- Python Package: gensim





Code Explanation Theory Explanation Visual Theory Explanation 1 Visual Theory Explanation 2

Natural Language Processing (NLP)

BERT

Transformers

Bidirectional LSTM RNN, Encoders & Decoders, Attention Models

Text Preprocessing Level 3: Word Embeddings, Word2Vec

Understanding Recurrent Neural Network(RNN), LSTM, GRU

Get Understanding of Artificial Neural Network

Solve Machine Learning Use Cases

Text Preprocessing: Gensim, Word2Vec, AvgWord2Vec

Text Preprocessing Level 2: Bag of Words, TF-IDF, Uni-Grams, Bi-Grams

Text Preprocessing Level 1: Tokenization, Lemmatization, Stop Words

NLP Road Map by Krish Naik
(From Bottom to Top)

His Video

What Next?

Deep Learning

Happy Journey 😐