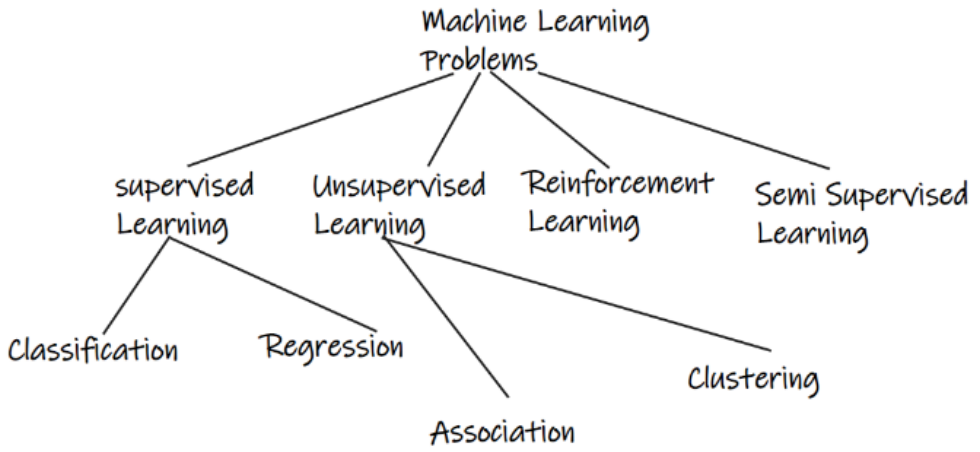
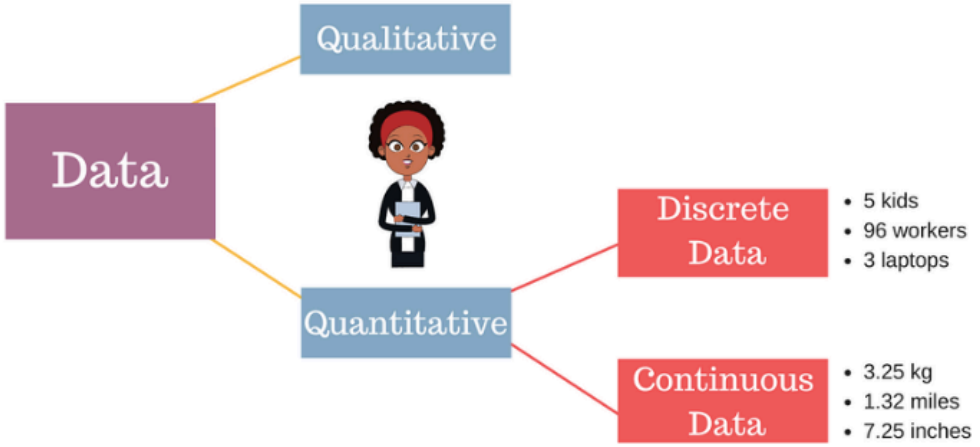
MACHINE LEARNING

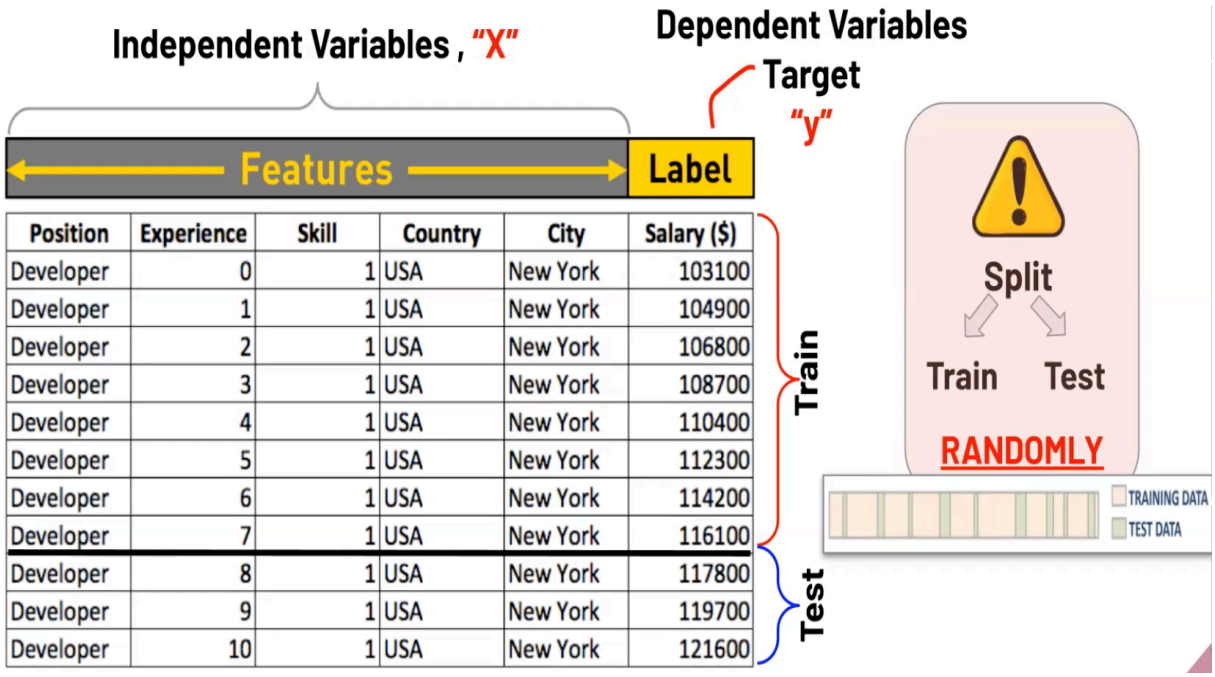




Supervised	Unsupervised
In a supervised learning model, input and output variables will be given.	In unsupervised learning model, only input data will be given
Supervised learning is a simpler method.	Unsupervised learning is computationally complex
Supervised learning model uses training data to learn a link between the input and the outputs.	Unsupervised learning does not use output data.
Highly accurate and trustworthy method.	Less accurate and trustworthy method.

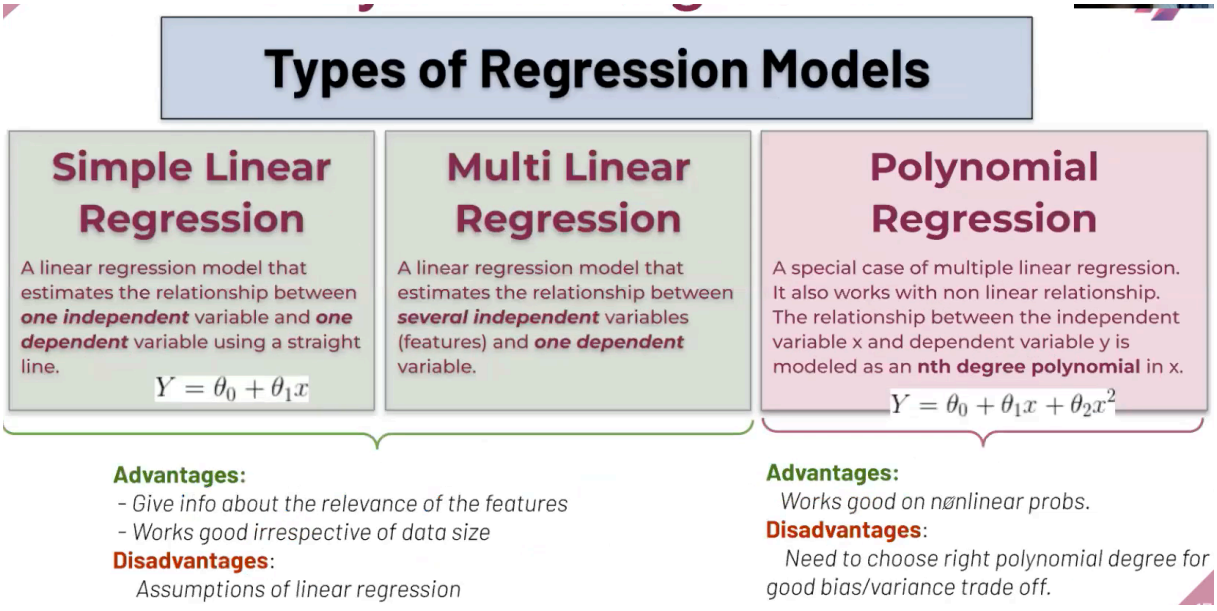
Note: In unsupervised learning, there is no tarining dataset.

Machine Learning Terminology



Position	Experience	Skill	Country	City	Salary ($)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

Supervised Learning



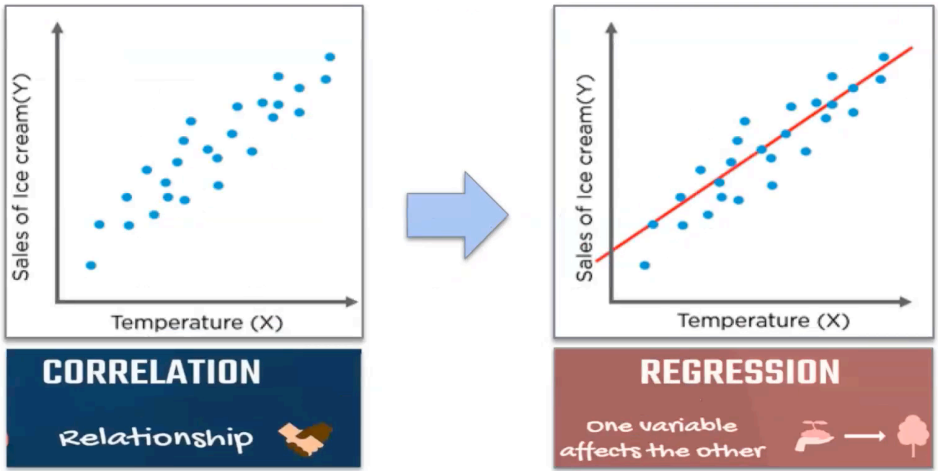
Advantages:
- *Give info about the relevance of the features*
- *Works good irrespective of data size*

Disadvantages:
Assumptions of linear regression

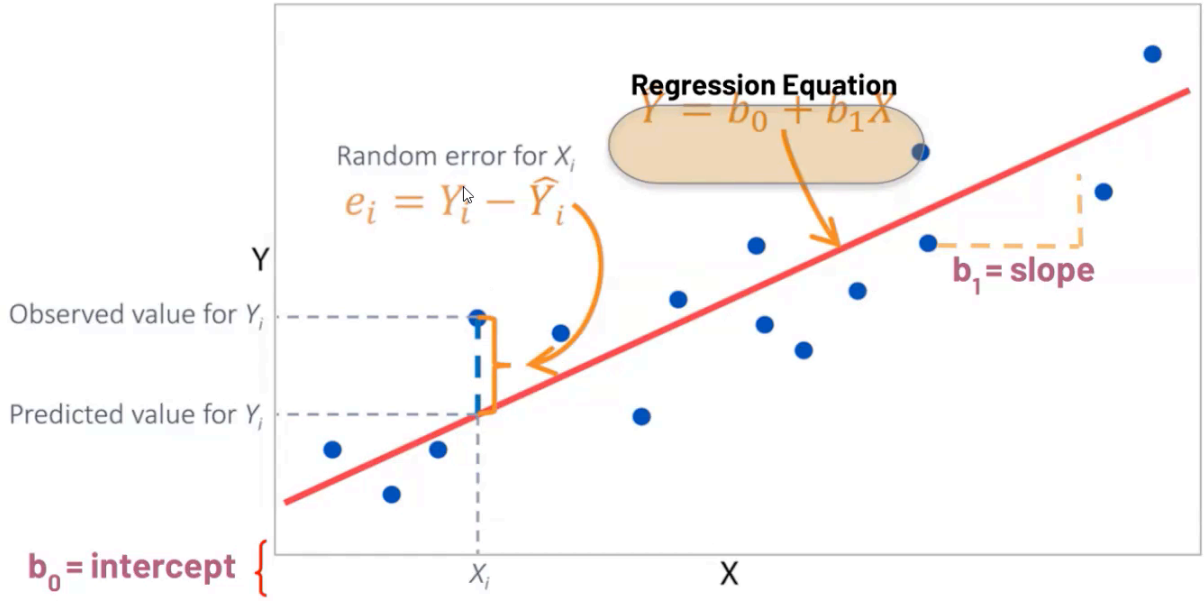
Advantages:
Works good on nonlinear probs.

Disadvantages:
Need to choose right polynomial degree for good bias/variance trade off.

Linear Regression Theory



- If there is no correlation, no regression.
- Linear regression y=bo+b1*x
- If there is one feature, it is called simple linear regression. If there are multi columns or features, it is called multi linear regression.
- Data must be continuous for linear regression.



Note: Gradient descent is an algorithm that finds the best fit line for given training dataset.

The Coefficient of Determination (R2)

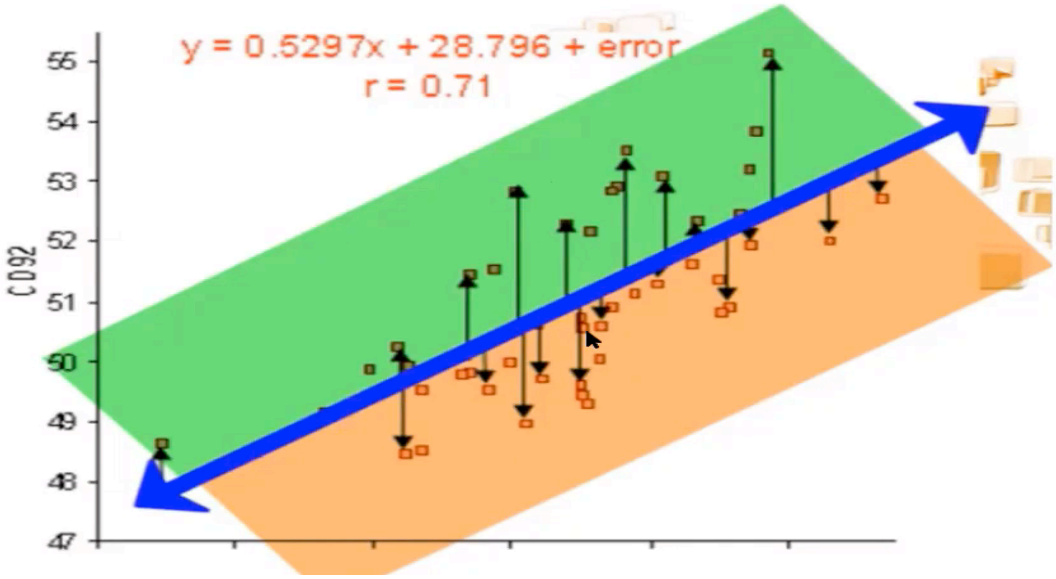
- An R2 of 0 means that the dependent variable cannot be predicted from the independent variable.
- An R2 of 1 means that the dependent variable can be predicted without error from the independent variable.-

- R2 score remains the most popular metric for evaluating linear regression models.

- Modelin başarısını değerlendirirken test data, error metric, ve R**2' e bakıyoruz.

Residuals (Errors)

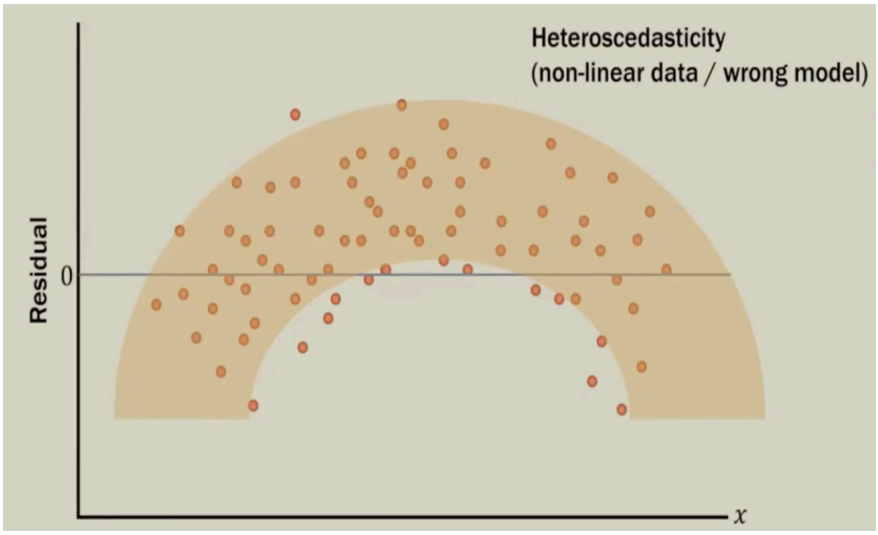
- A residual (error) is the vertical distance between a data point and the regression line.
- Sum and mean of all residuals equal zero.
- Residual = Observed value - predicted value
- Each data point has one residual:
  Positive: above the regression line
  Negative: below the line
  Zero: on the line



INTERVIEW QUESTION

Q: How to understand if the best line is drawn?
A: If sum of the residuals equals zero, the best line was drawn.

- Below shape is not suitable for linear regression.
- It is called as heteroscedasticity.

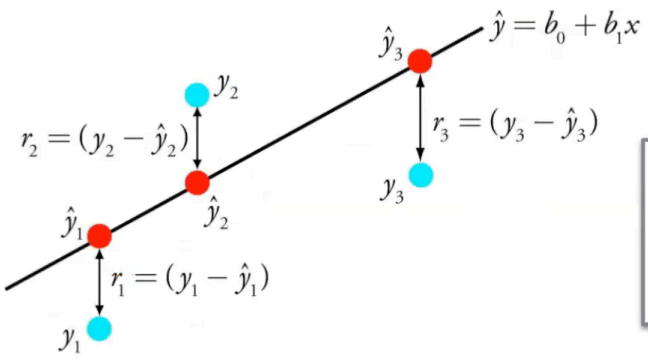


- Below data is suitable for linear regression.
- It is called as homoscedasticity.
- Sum of residuals equals zero.

Regression Error Metrics

- Mean Absolute Error (MAE)

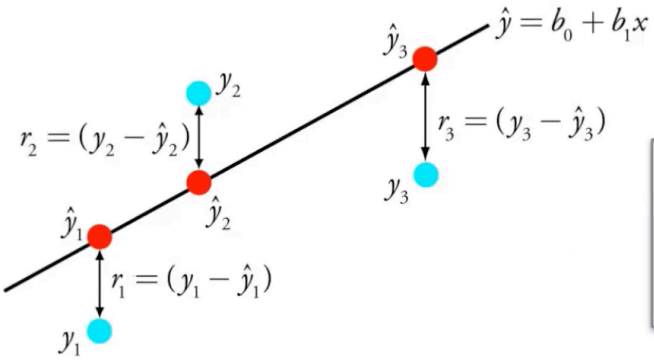
- MAE is the absolute differences between the actual target value and the value predicted by the model.
- Not penalizes errors.



$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Mean Squared Error (MSE)

- As it squares the differences, it penalizes even a small error which leads to over-estimation of how bad the model is.

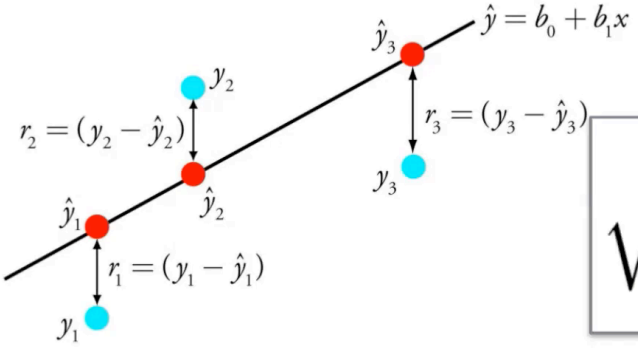


$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Root Mean Square Error (RMSE)

- The errors are first squared before averaging which poses a high penalty on large errors.

- RMSE is useful when large errors are undesired.



$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

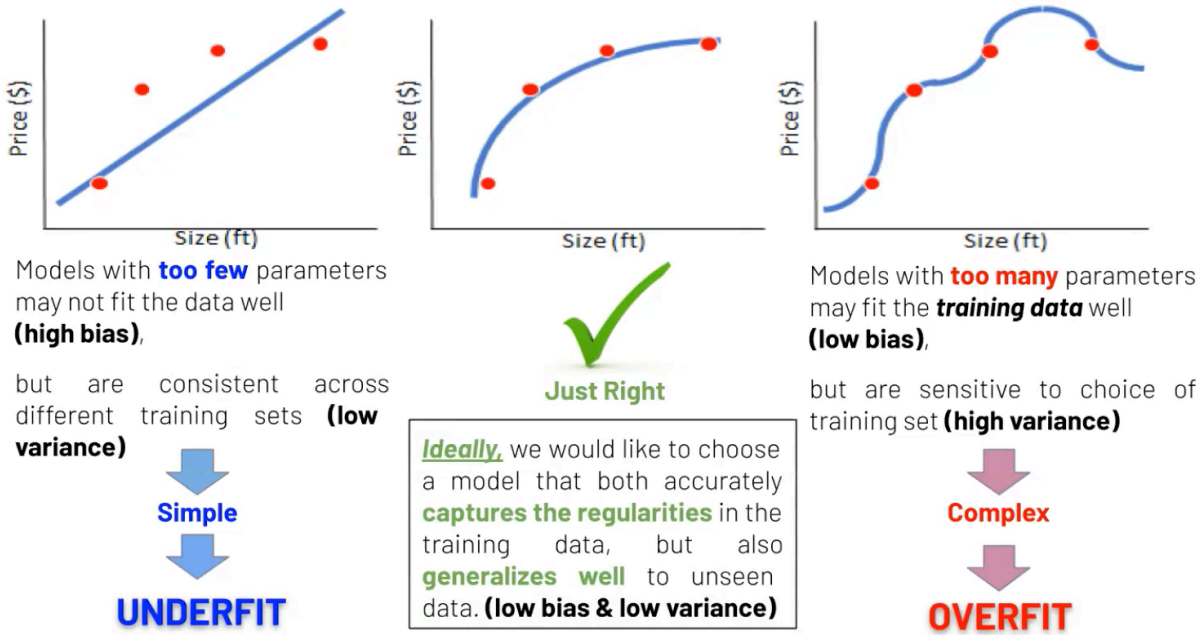
Note: If there are huge differences between mae and rmse, you have uneffective predictions.

INTERVIEW QUESTION

Q: How to evaluate linear regression models?

A: Have a look at R square, correlation, residuals, the metric you chose, residuals should has normal skewness, and residuals randomly should be located in the graph.

Introduction To Bias-Variance Trade-Off



High bias of a machine learning model is a condition where the output (y_pred) of the machine learning model is quite far off from the actual output.

The target function is estimated from training data by a machine learning algorithm, so we should expect the algorithm to have some variance. Ideally, it should not change too much from one training dataset to the next. ***Low variance*** suggests small changes to the estimate of the target function with changes to the training dataset.

Underfitting:

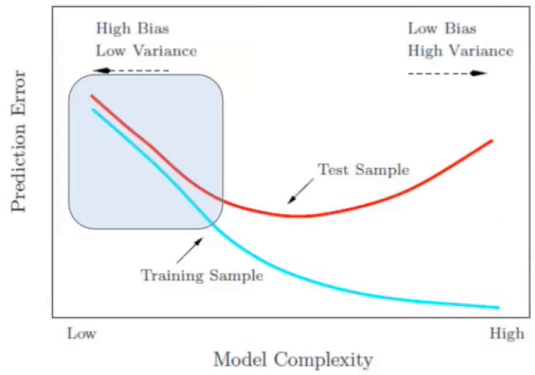
- High training error
- High test error

Overfitting:

- Low training error (high score)
- High test error (low score)
- We generally face overfitting

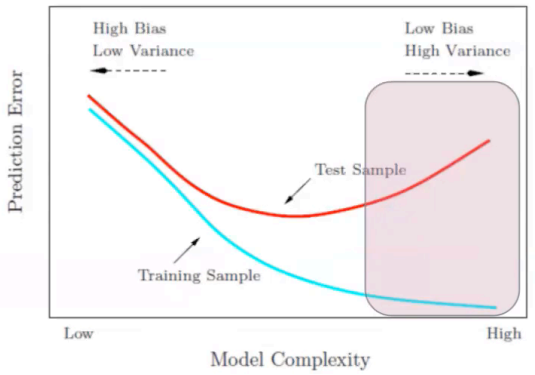
UNDERFITTING

High training error and **high** test error



OVERFITTING

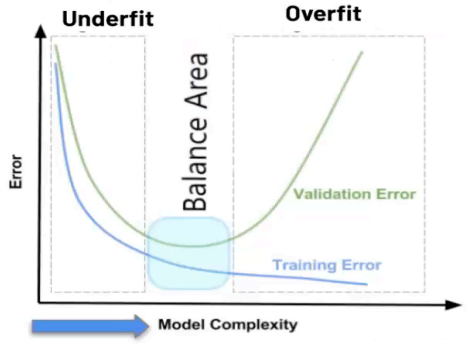
Low training error and **high** test error



How To Deal With

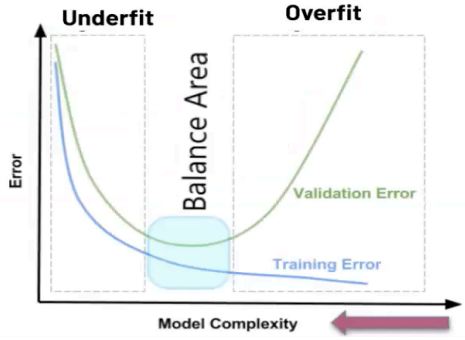
UNDERFITTING ?

- Find a **more complex** model



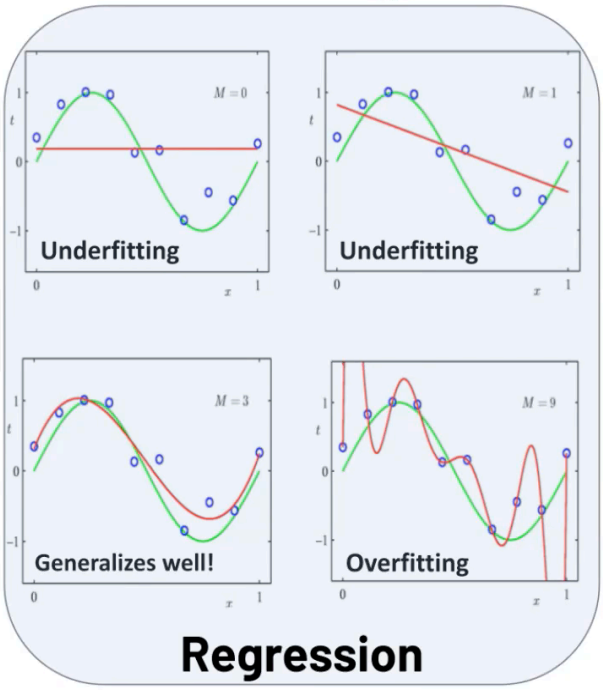
OVERFITTING ?

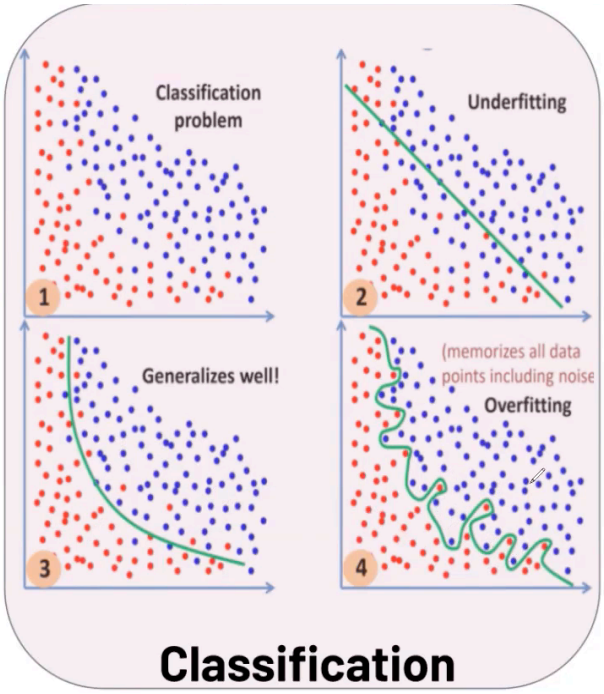
- Decrease the number of parameters
- More training data / **Cross Validation**
- **Regularization (Lasso&Ridge)**



Summary

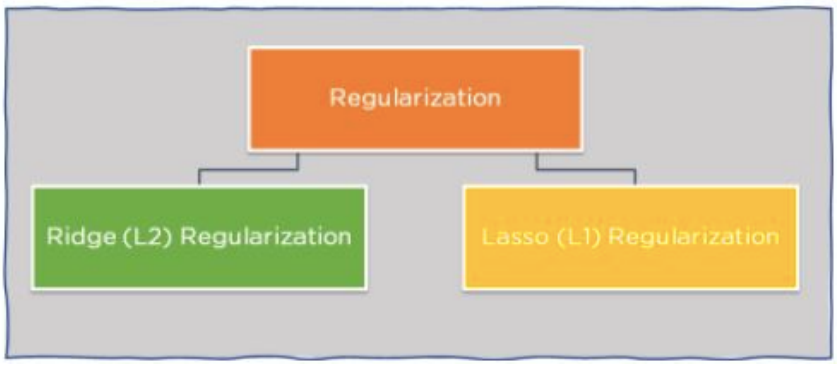
	MODEL				
	Bias	**Variance**	**Complexity**	**Flexibility**	**Generalizability**
Underfitting: An overly simple model	High	Low	Low	Low	High
Overfitting: Modeling the noise as well	Low	High	High	High	Low





Regularization

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.



Ridge (L2) Regression

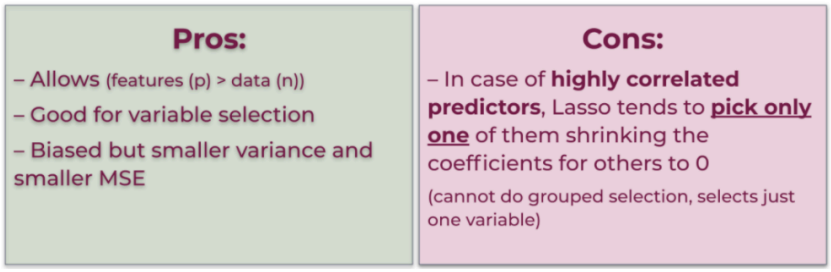
Ridge does not make feature selection. λ is very important for Ridge regression.

Pros:	Cons:
– Good for few data sets (if features (p) > data (n)) – Good for multicollinearity – Biased but smaller variance and smaller MSE	– Not suitable for variable selection or for assessing the relative importance of each predictor.

Lasso (L1)

Lasso regression has very much similarities with Ridge Regression. It's a regularization technique that reduces the complexity of a model by shrinking some of the coefficients to zero. It is used for feature selection. Feature selection is only done with lasso.

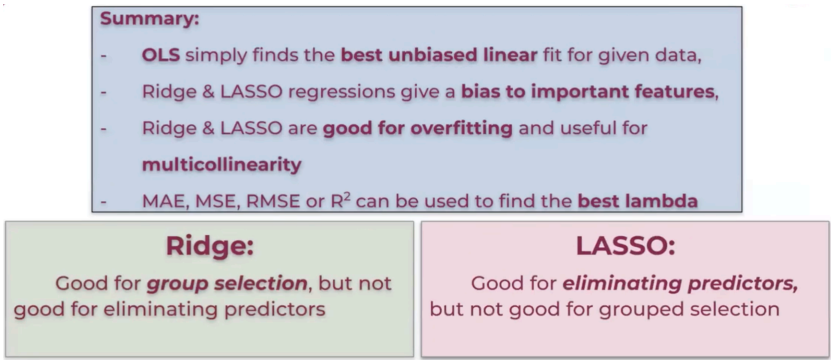
Note: LASSO diyor ki bu ikisi iş yapmıyor bunları çıkart, sadece şununla devam et. Ridge ise hepsinden faydalanmak istiyor. LASSO da multicollinearity varsa onları oyundan çıkartıyor.



Elastic-Net

Hyper-Parameter (Lambda): It is a method that we can manually change accuracy and metrics. We will use "alpha" as a hyperparameter name, instead of lambda.

Alpha: It is also known as the learning rate parameter which has to be set in a gradient descent to get the desired outcome from a machine learning model. Alpha is a set amount of change in the coefficients on each update.



Note: The main goal of Ridge and Lasso is to struggle with overfitting and multicollinearity.

Feature Scaling

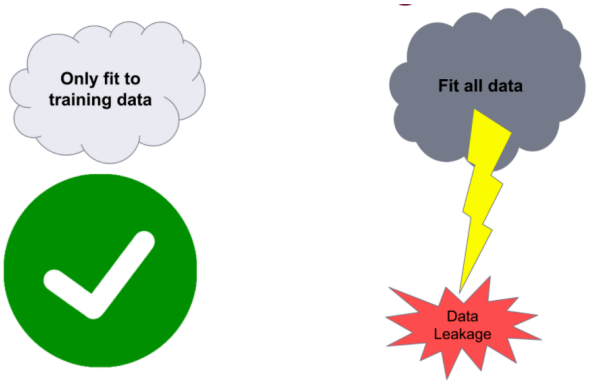
- If you train your model with scaled data, you must scale the unseen input before prediction.

- Only fit to training data.

- Feature scaling is a method used to normalize the range of independent Variables, inputs or features of data. In data processing, it is also known as data normalization and is generally

performed during the data preprocessing step. For example, if you have multiple independent variables like age, salary, and height; with their range as (18–100 Years), (25,000–75,000 Euros), and (1–2 Meters) respectively, feature scaling would help them all to be in the same range; for example, centered around 0 or in the range (0,1) depending on the scaling technique.

- The features will be re-scaled so that they'll have the properties of a standard normal distribution with mean=0, std=1.



- There are three main scalers;

1. **MinMaxScaler():** It transforms features by scaling each feature to a given range, which is generally [0,1], or [-1,-1] in case of negative values. This scaling algorithm works very well in cases where the standard deviation is very small, or in cases which don't have Gaussian distribution.

$$\frac{x_i - \min(x)}{\max(x) - \min(x)}$$

2. **RobustScaler():** Robust Scaler algorithms scale features that are robust (strong) to outliers.

$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)}$$

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

3. **StandardScaler():** Standardization can be helpful in cases where the data follows a Gaussian distribution (or Normal distribution). StandardScaler results in a distribution with a standard deviation equal to 1 and mean equal to 0.

Note: Data leakage happens when your training data contains information about the target, but similar data will not be available when the model is used for prediction.

Multicollinearity

- Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model. This means that an independent variable can be predicted from another independent variable in a regression model. Still, the model didn't recognize them as important predictors due to multicollinearity. That's why Ridge and Lasso are used to avoid multicollinearity.

- The most important matter of multicollinearity is to choose which features are the most important. Multicollinearity problems are genereally seen in linear regression.

Cross Validation

- Cross-validation is a statistical method used to evaluate the performance (or accuracy) of machine learning model which was trained on training dataset.

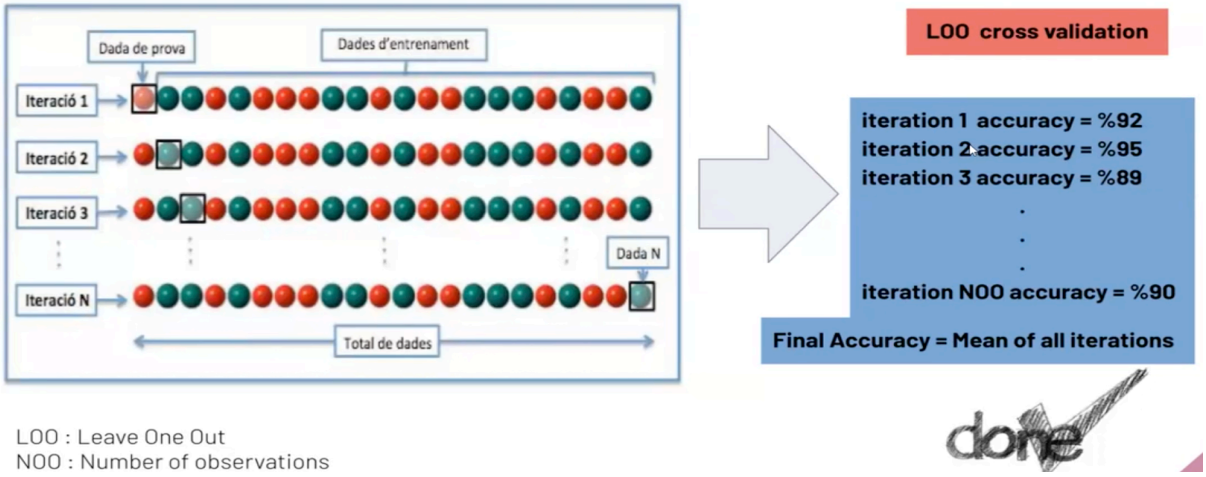
- Usually, k is equal to 3 or 5. It can be extended even to higher values like 10 or 15 but it becomes extremely computationally expensive and time-consuming.

```
print(cross_val_score(model, X_train, y_train, cv=5))
```

- The main goal is to overcome overfitting problem.

- There are several **cross validation techniques** such as:

1. K-Fold Cross Validation
2. Leave P-out Cross Validation
3. Leave One-out Cross Validation
4. Repeated Random Sub-sampling Method
5. Holdout Method



LOO : Leave One Out
NOO : Number of observations

Grid Search

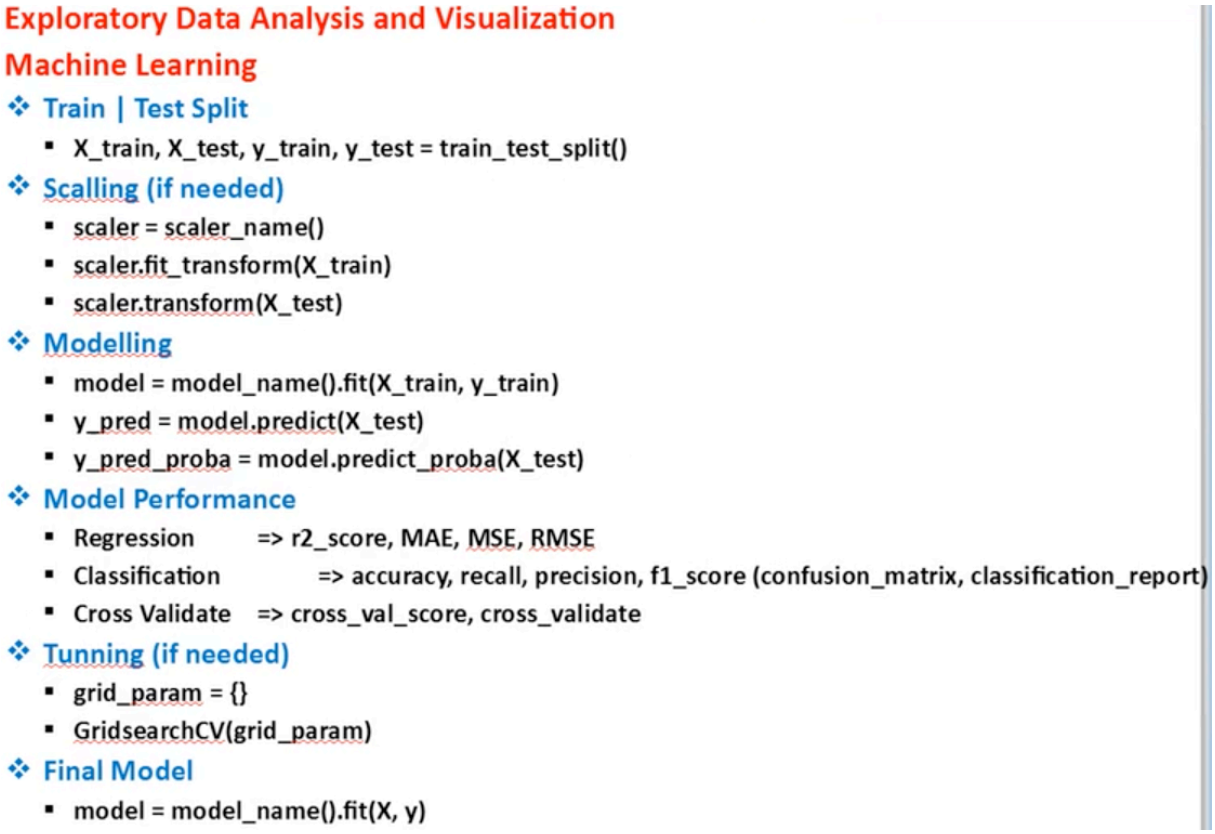
- Grid search method finds the hyperparameters that give the best scores in a model.

- Grid search is used for all models.

INTERVIEW QUESTION

Q: What is Regularization and why is it useful?
A: The main goal is to struggle with overfitting. It is useful because struggling improves model performance.

Machine Learning Flow Diagram



Note: Grid Search could be used for all models.

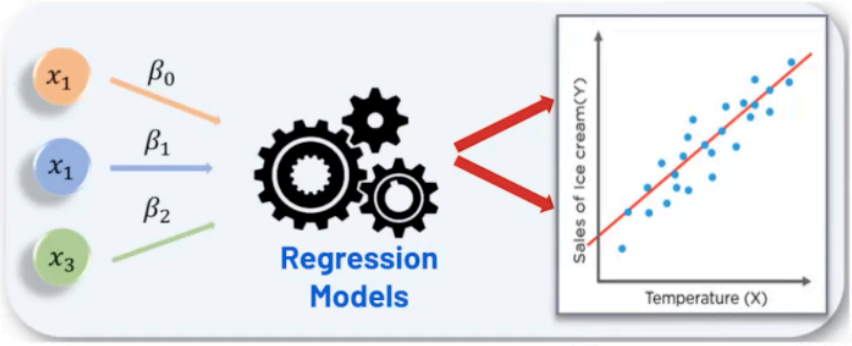
Logistic Regression Theory

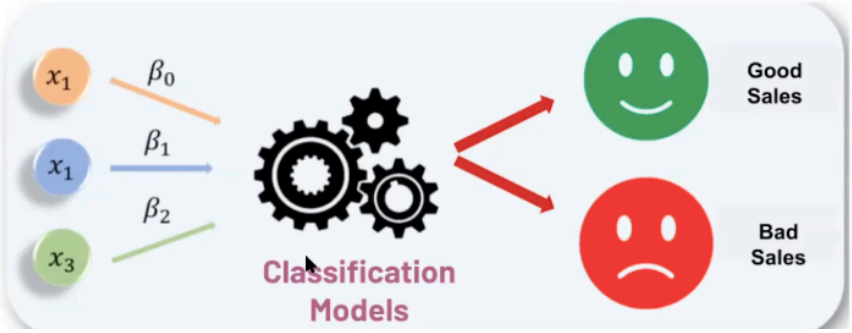
- Logistic regression is a method for classification.

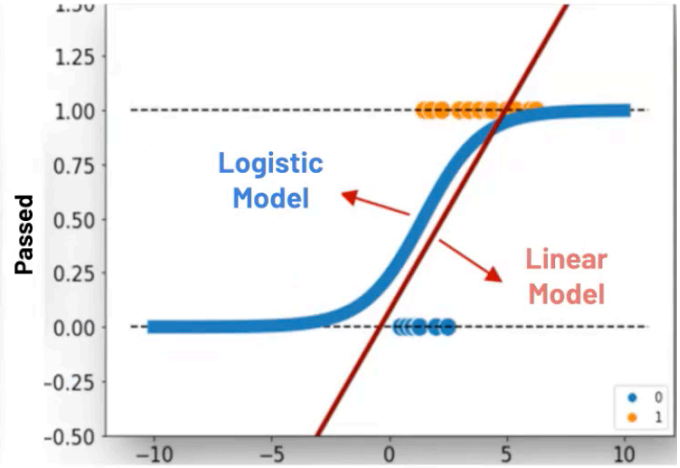
- For example, clinical trials (Disease Diagnosis), pregnancy test, cancer test, predicting the customer churn, and fraud detection.

- If there are two classess, it is called binary classification. Otherwise, it is called multiclass classification.

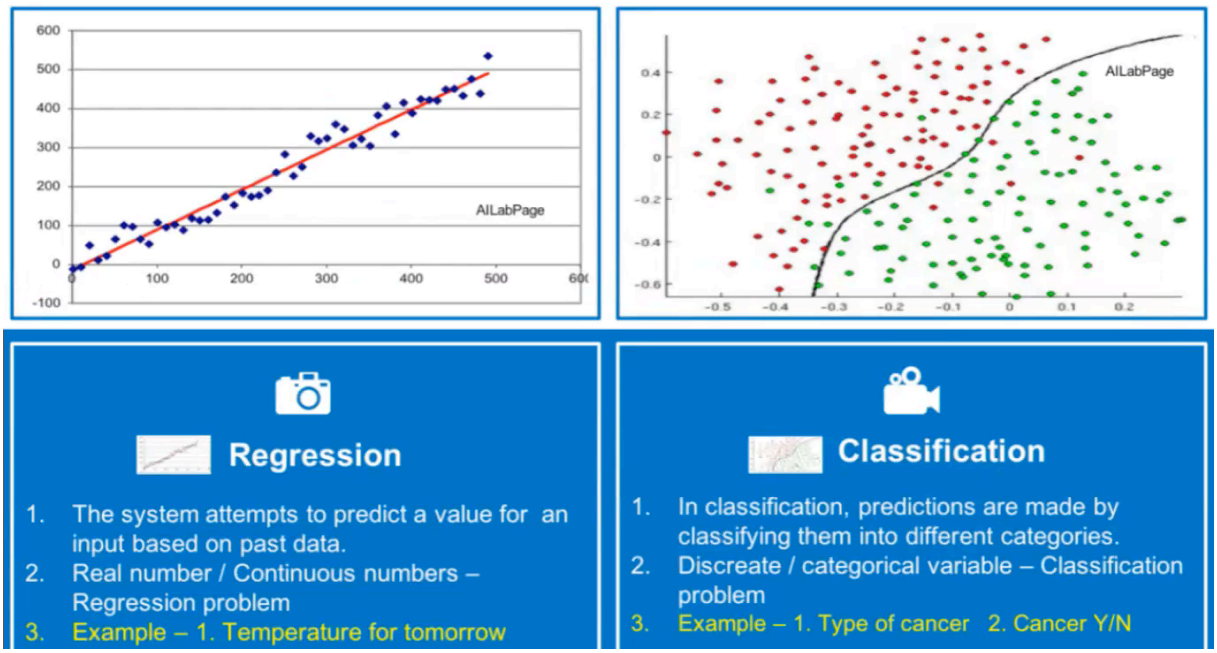
- In this model, Sigmoid function is used. It gives the probability between 0 and 1.


x_1
x_1
x_3
β_0
β_1
β_2
Regression Models
Sales of Ice cream(Y)
Temperature (X)


x_1
x_1
x_3
β_0
β_1
β_2
Classification Models
Good Sales
Bad Sales


Passed
1.25
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
−10
−5
0
5
10
Logistic Model
Linear Model
0
1

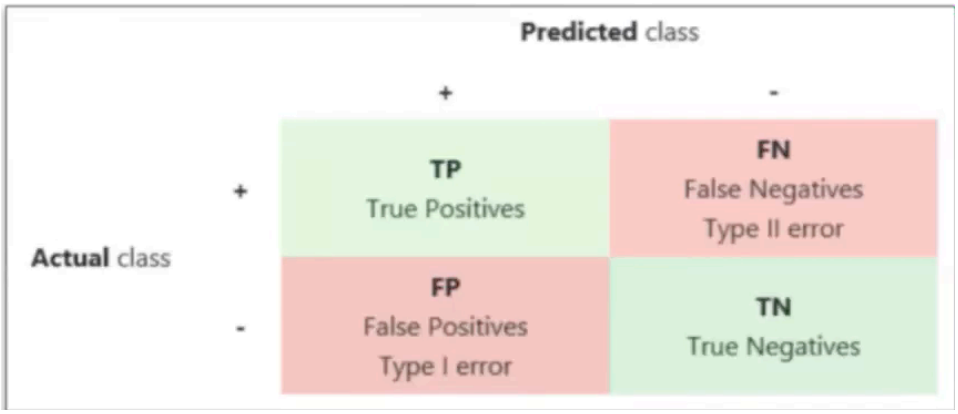
Regression vs Classification

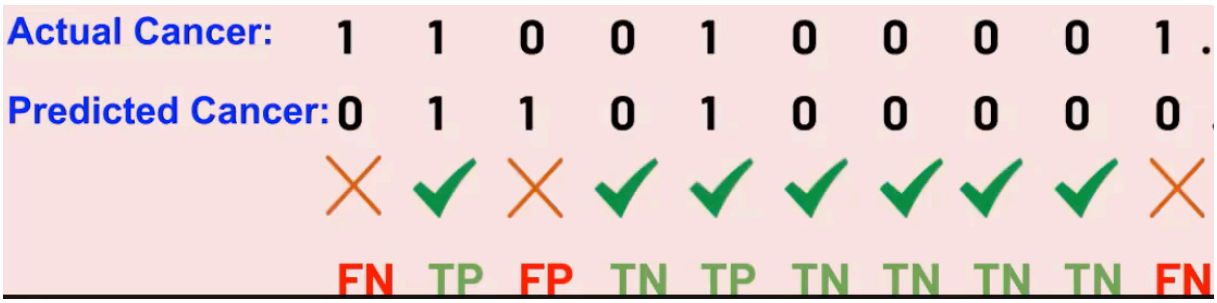


Evaluating Performance

Classification Error Metrics

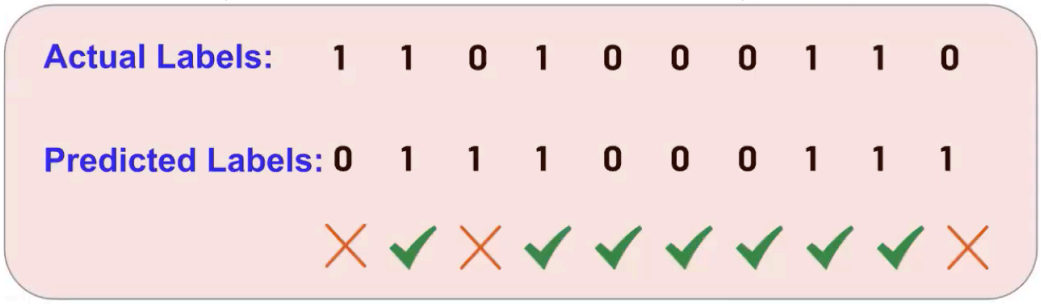
Confusion Matrix





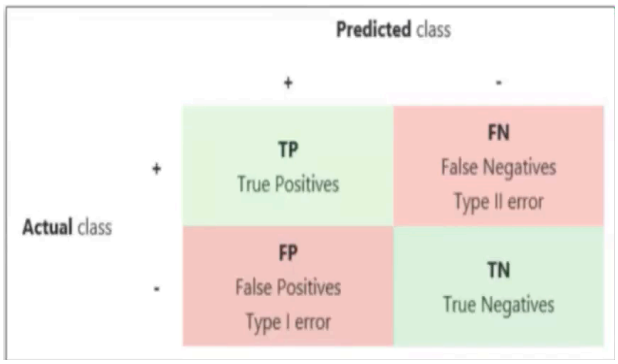
- We conclude four metrics (Accuracy, Recall, Precision, F1-Score) from confusion matrix.

Accuracy: The most basic and common metrics to evaluate our model.

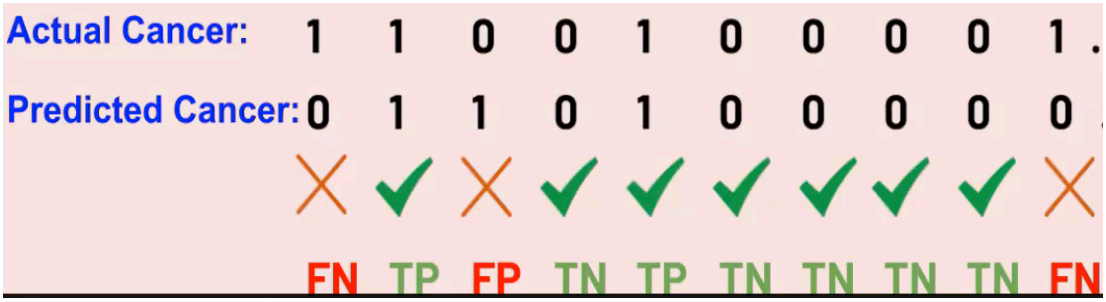


$$\frac{\text{All correctly predicted values (7)}}{\text{All predicted values (10)}} \times 100 \Rightarrow \text{Accuracy} = 70\ \%$$

Recall: Correctly predicted as positives compared to total number of positives. For instance, you have 10 red balls among 100 balls and you are trying 100 predictions. If you predicted all red balls during the prediction, Recall would equal 10/10=1

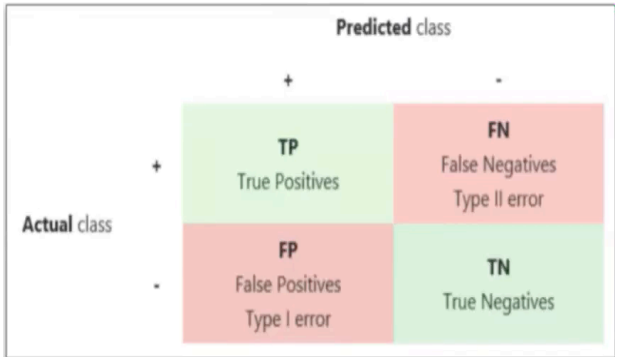


$$\text{Sensitivity} = \frac{TP}{TP+FN} = \frac{2}{2+2}$$

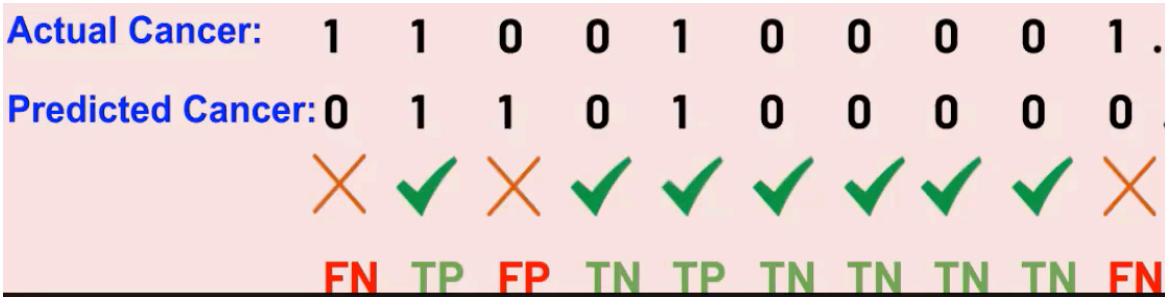


Note: Macro avg in classification report is calculated by (precision of 0 + precision of 1) / 2. Weighted avg is also calculated by (m of support*0 + n of support) / m+n. If data is balanced, accuracy (micro) could be choosen.

Precision: Correctly predicted as positives compared to total predicted of positives. For instance, you have 10 red balls among 100 balls and you are trying 100 predictions. If you predicted all red balls during 100 times prediction, Precision would equal 10/100=0.1. By the way, you made mistake predictions during the prediction, so you predicted as red balls actually not.



$$\text{Precision} = \frac{TP}{TP+FP} = \frac{2}{2+1}$$

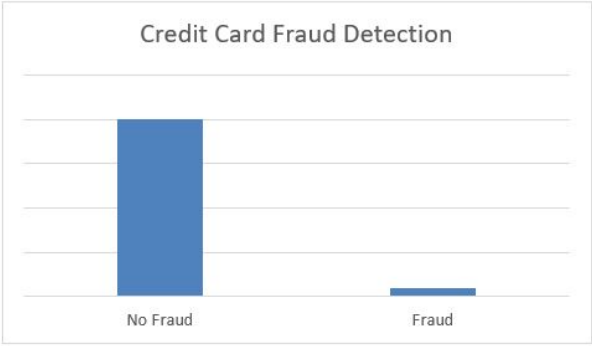


F1 Score: Harmonic mean of Precision and Recall. Mostly used for unbalanced distribution. Harmonic mean is used to avoid weighting of a parameter. For example, p=90, r=0; the result will be 2 for normal mean but it will be 0 for the harmonic mean.

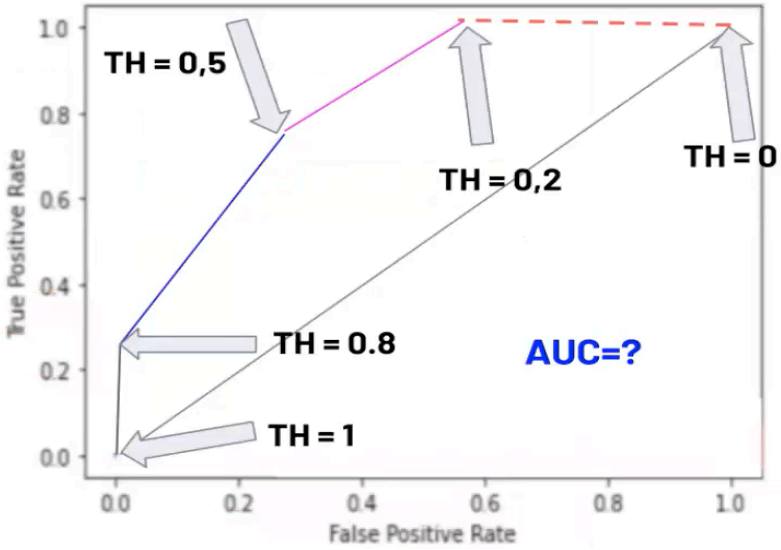
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Note: Imbalanced (unbalanced) data refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, stratify=y, random_state=42)



ROC/AUC (Receiver Operator Characteristics/Area Under Curve):



Note: In case of balanced datasets, we look at ROC/AUC, oppositely we look at precision recall curve for unbalanced datasets.

Imbalance Situation: If the model can not predict well with a sample trained data, it is called imbalanced situation. For example, the data has 10 times 0 class and 100 times 1 class, and then if the prediction is balanced, it is called balanced data. Oppositely, it is called unbalanced data.

Note: If scores are very imbalanced, then you must look at “macro avg” in unbalanced datasets.

KNN (K Nearest Neighbors) Theory

KNN is a simple, supervised classification. It is a memory-based approach.

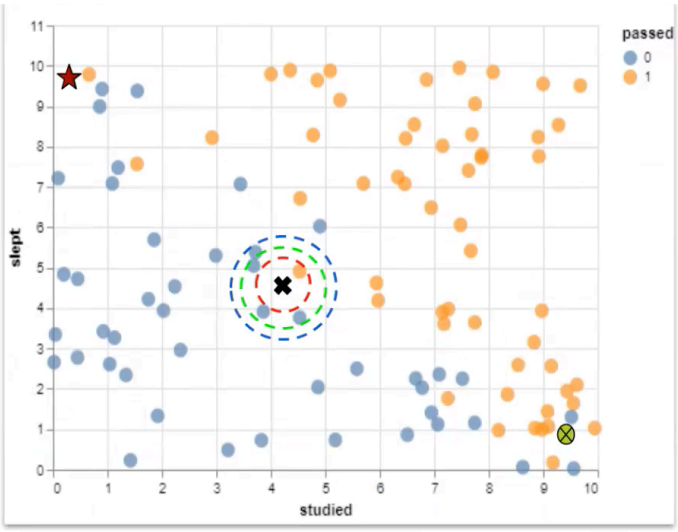
- Non-linear and non-parametric.

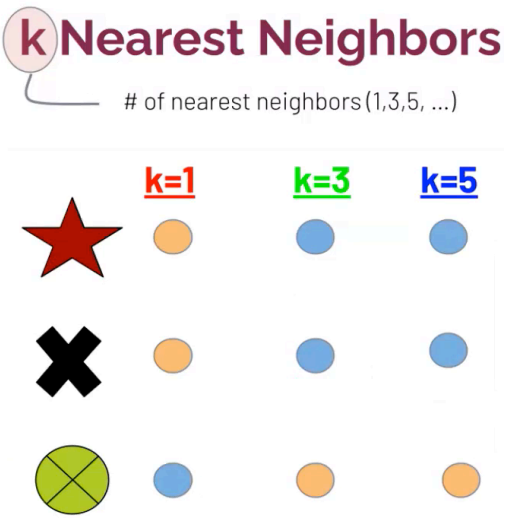
- It is used in fault detection.

- As k value increases, underfitting situation exists. Oppositely, as k value decreases, overfitting situation exists. So, chosing of k is so important.

- Scaling process has to be done.

- It is used in both classification and regression.





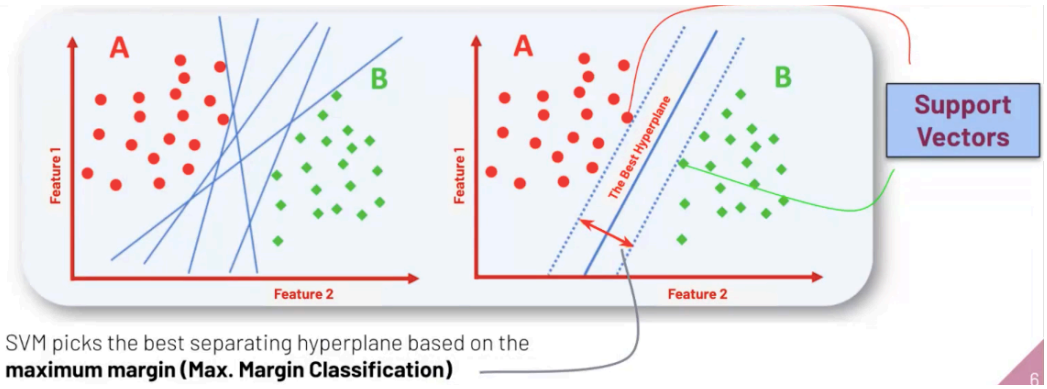
SVM (Support Vector Machine)

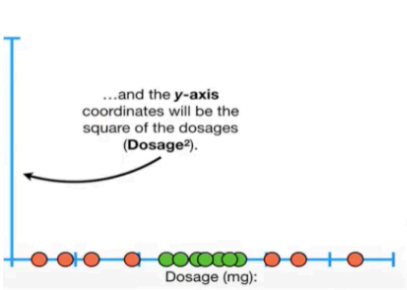
- SVM is widely used for classification algorithm.

- It is used in text classification, image recognition, and image-based gender detection.

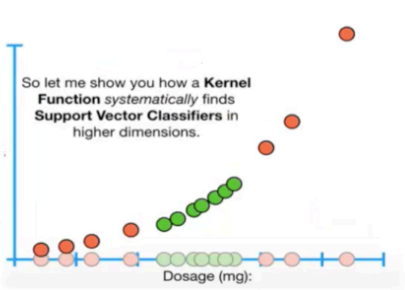
- Works great with computer vision problems.

- It can model the complex relationships.

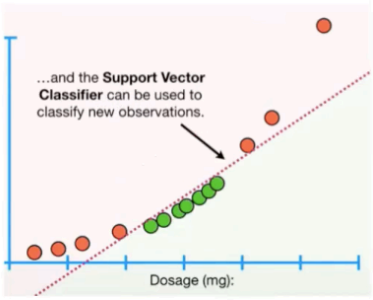




2. New *axis (y)*



3. 1D to 2D



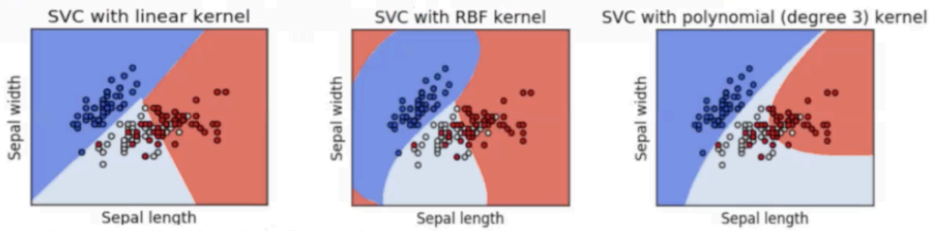
4. Hyperplane

SVM Algorithm:

Hyperparameters

"kernel" *parameter:* (default="rbf")
{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}

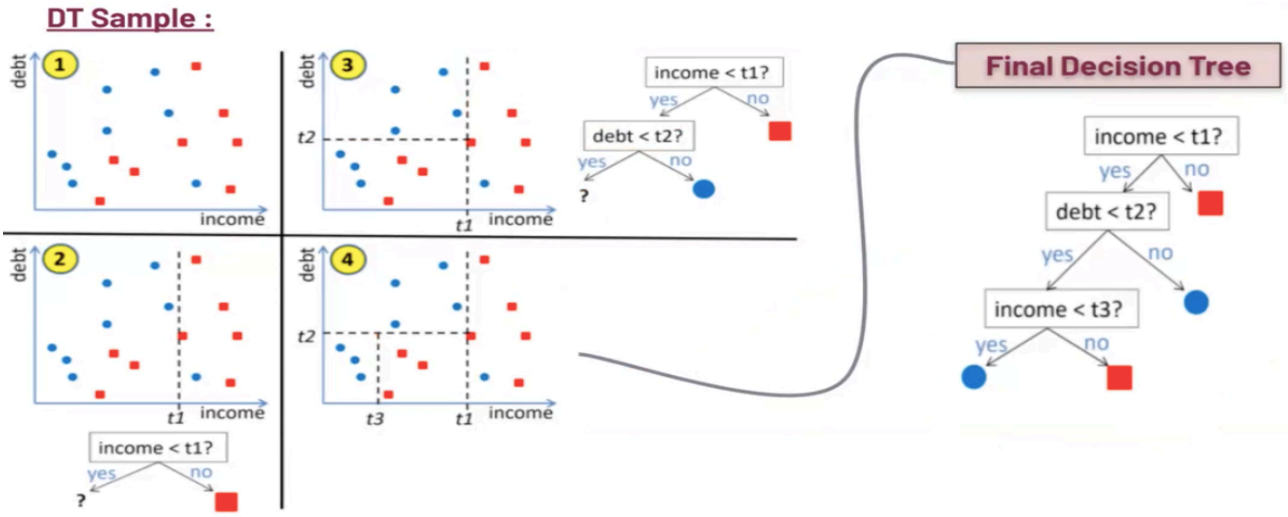
Specifies the kernel type to be used in the algorithm.



Decision Tree Theory

- It is considered as the most widely used and successful algorithms.

- Its usage areas are Medical Diagnosis, Text Classification, Credit Risk Analysis.



- We need some sort of a measure to decide which attribute to start splitting with. These are Gini Index and Entropy.

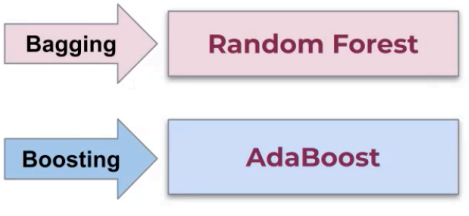
Gini Index (Gini Impurity Index): It is a criterion to minimize the probability of misclassification. It should be maximum.

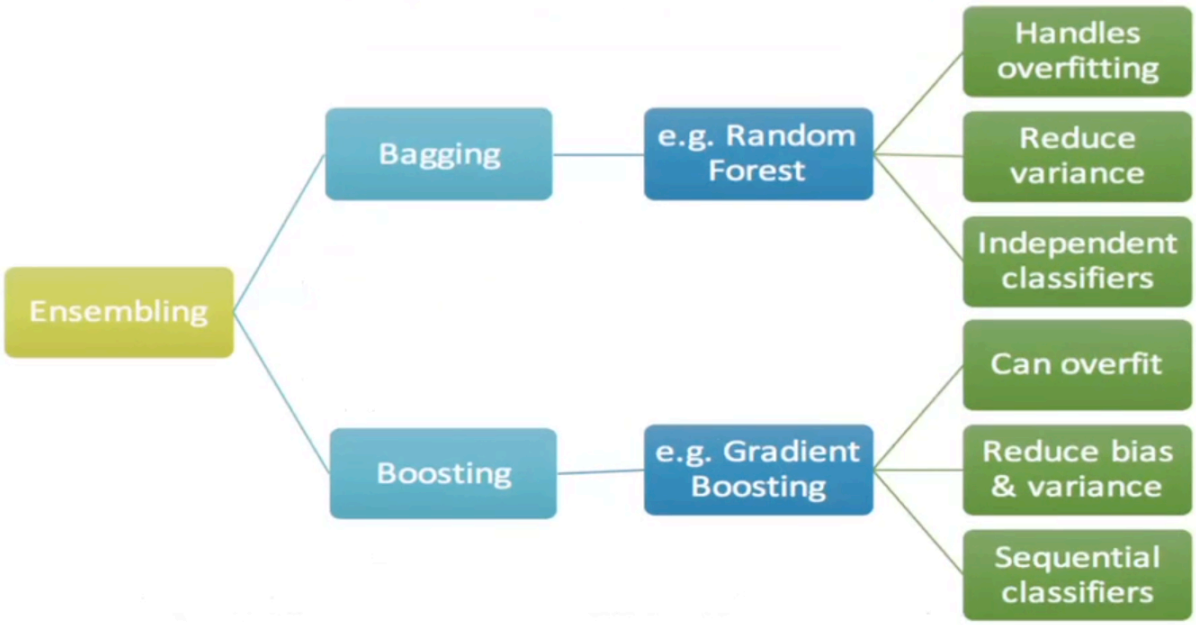
Entropy (Information Gain): It is based on a purity measure called Entropy. It is between 0 and 1. It should be minimum.

Ensemble Methods

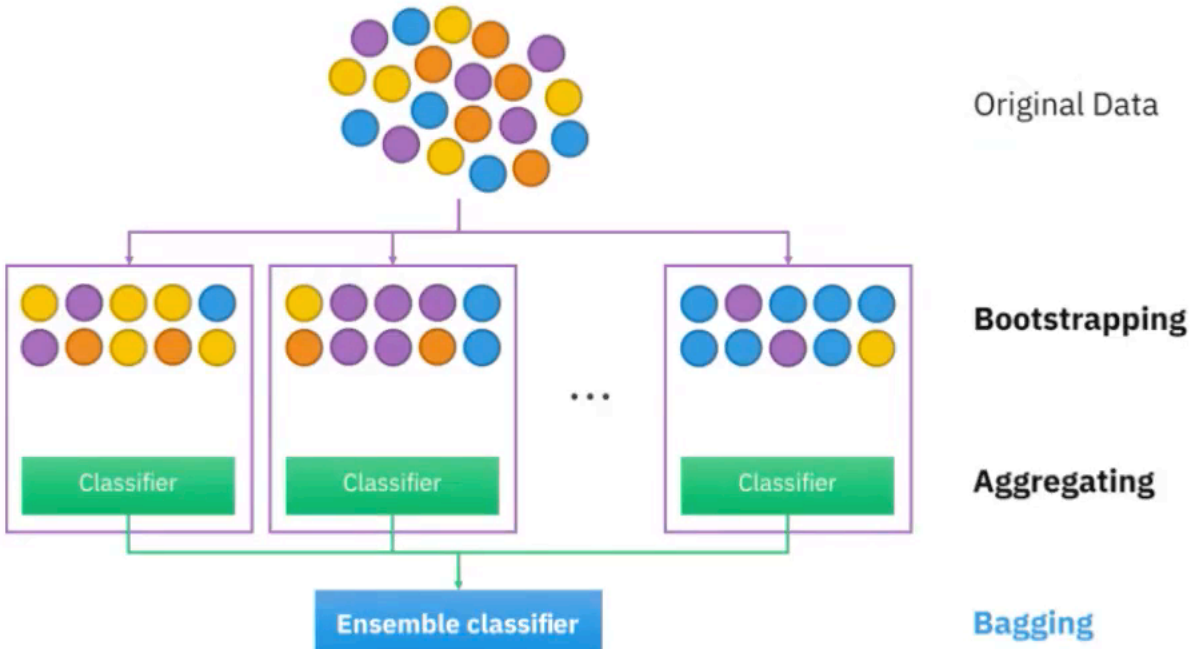
- It provides training of different models on the same dataset. Each model makes its own predict and a meta-model is formed from the combination of predictions.

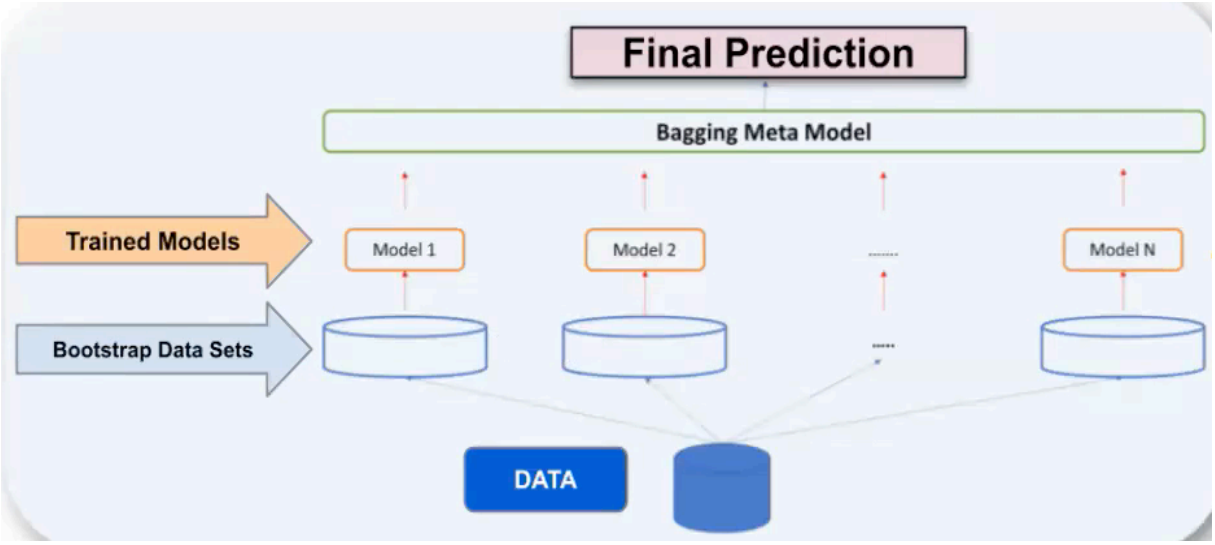
	Bagging	Boosting
Similarities	• Uses voting • Combines models of the same type	
Differences	Individual models are built separately	Each new model is influenced by the performance of those built previously
	Equal weight is given to all models	Weights a model's contribution by its performance





Bagging (Bootstrap AGGregatING): It can be defined Bootstraps as creating a new samle dataset using values from original dataset. The clue is that each subsample has the same size but different parts of the observations of the original dataset.





Random Forest Algorithm

- RF is an ensemble bagging algorithm composed of many individual trees.

- RF reduces the variance of individual trees by randomly selecting many features from the dataset, and averaging them.

- RF uses a random subset of features rather than using all features to grow trees.

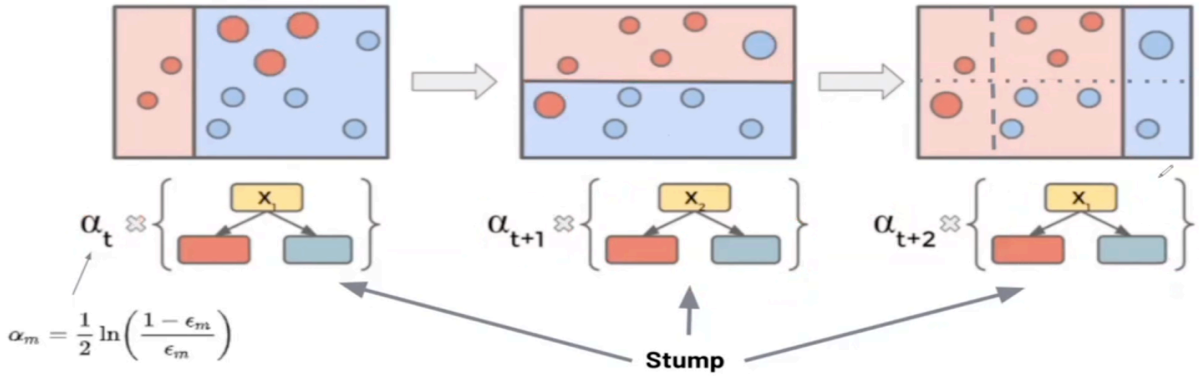
Pros:	Cons:
– Good for multi-class problems – Perfect for quick predictions – Ranking for feature importance	– Not good on smaller data sets – Training could be computationally expensive for a large number of trees

Boosting Algorithm

- It is a strong model building method with iterations, taking into account the errors of weak algorithms created in the same dataset.

- It combines the outputs from weak learner and creates a strong learner which eventually improves the prediction power of the model.

AdaBoost (Adaptive Boosting):



GBM (Gradient Boosting): Like Adaboost, the Gradient Boosting algorithm is based on correcting the errors of the previous model. However, this correction is realized by adding residual estimation errors to the model, not by changing the weights like AdaBoost.

XGBoost (Extreme Gradient Boosting): As the name suggest, XGBoost is from the same family as Gradient Boosting. However, there are some additional EXTREME capabilities of XGBoost that make it very prominent.

Pros:	Cons:
– Handles large sized datasets well – Perfect for quick predictions – Ranking for feature importance – Good model performance (wins most of the Kaggle competitions)	– Difficulty interpretation due to visualization difficulties – Harder to tune as there are too many hyperparameters

---

INTERVIEW QUESTION

Q: What are the differences between Bagging and Boosting?

Unsupervised Learning

Learning problems whose no dependent or target variable are called unsupervised learning. Two of the most common unsupervised learning problems are dimensionality reduction and clustering.

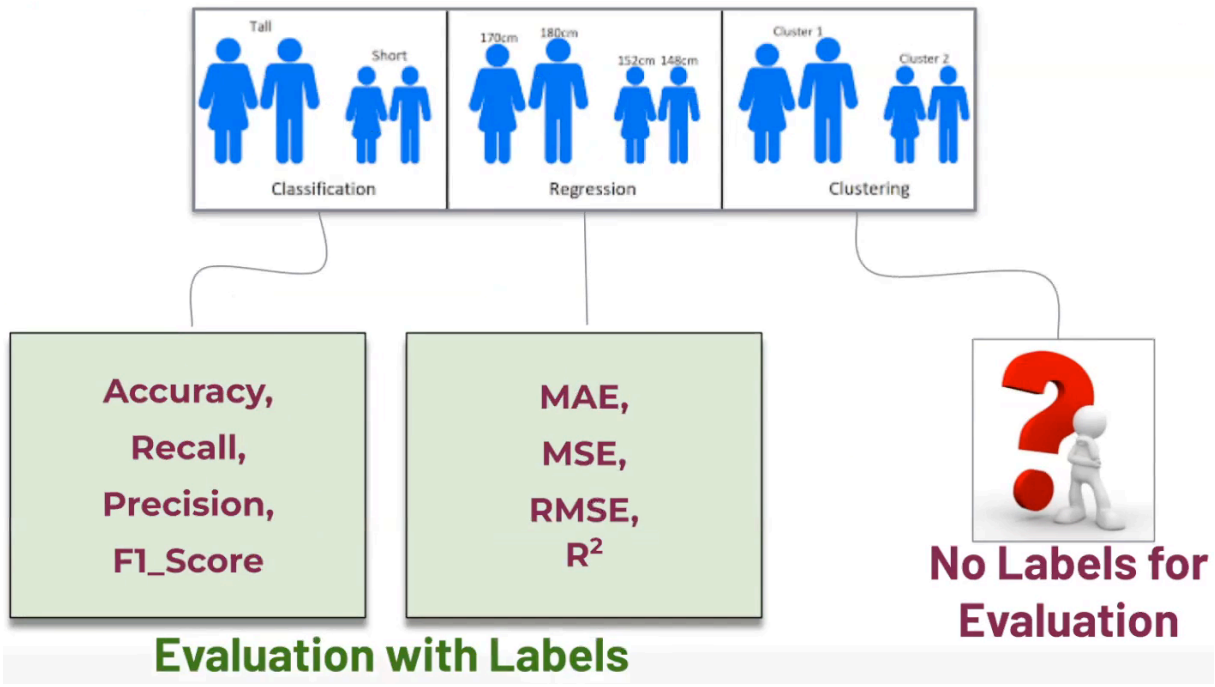
	Age	Cabin	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Ticket	Embarked	Survived
0	22.0	NaN	7.2500	Braund, Mr. Owen Harris	0	1	3	male	1	A/5 21171	S	0
1	38.0	C85	71.2833	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	2	1	female	1	PC 17599	C	1
2	26.0	NaN	7.9250	Heikkinen, Miss. Laina	0	3	3	female	0	STON/O2. 3101282	S	1
3	35.0	C123	53.1000	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	4	1	female	1	113803	S	1
4	35.0	NaN	8.0500	Allen, Mr. William Henry	0	5	3	male	0	373450	S	0
5	NaN	NaN	8.4583	Moran, Mr. James	0	6	3	male	0	330877	Q	0
6	54.0	E46	51.8625	McCarthy, Mr. Timothy J	0	7	1	male	0	17463	S	0
7	2.0	NaN	21.0750	Palsson, Master. Gosta Leonard	1	8	3	male	3	349909	S	0
8	27.0	NaN	11.1333	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	2	9	3	female	0	347742	S	1
9	14.0	NaN	30.0708	Nasser, Mrs. Nicholas (Adele Achem)	0	10	2	female	1	237736	C	1

No Dependant Feature

Clustering: In clustering, we group similar observations with an algorithm. These are called clustering algorithms.

- If the number of clusters is too large, the clusters will start to differ insignificantly.

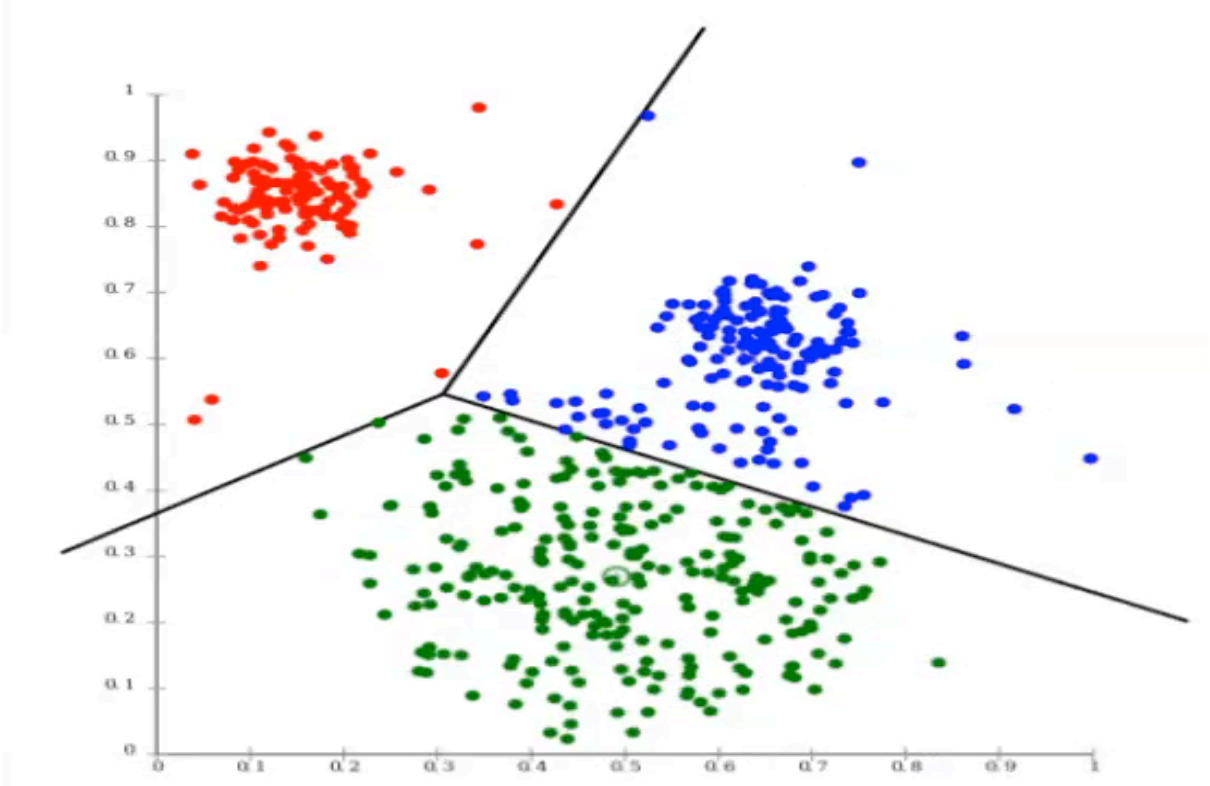
- If there are too few clusters, we won't be able to get too much information from them and the real differences will be hidden.



K-means Clustering Theory: It is widely used UNsupervised Learning classification algorithm.

- K represents the number of clusters.

- Each cluster consists of similar data within itself, but the clusters are not alike each other.



Clustering Evaluation

- Clustering Tendency

Hopkins test, a statistical test for randomness of a variable. If value below 0.5 after Hopkins command, the data is suitable for clustering. If not, it is not suitable.

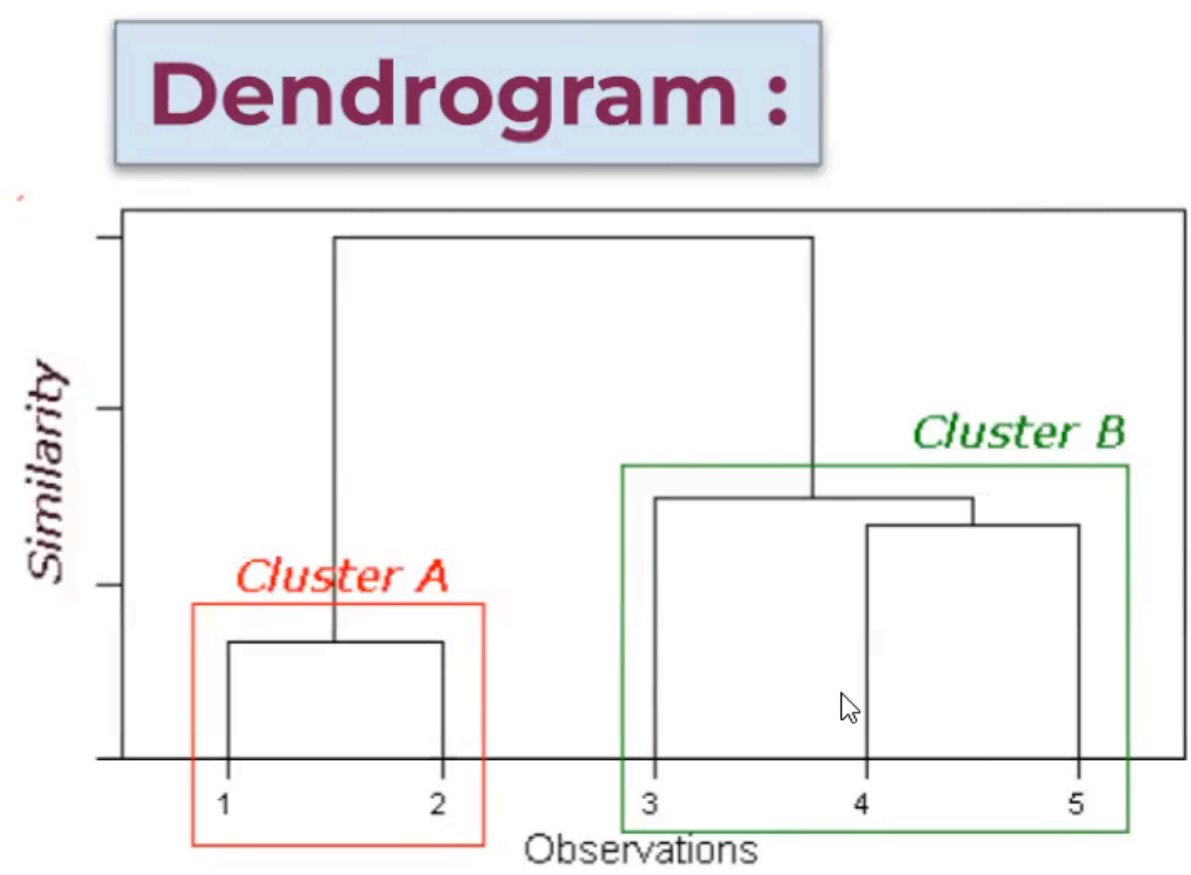
- Silhouette Coefficient

If the ground truth labels are not known, evaluation must be performed using the model itself.

- A higher Silhouette Coefficient score relates toa model with better defined clusters.

Hierarchical Clustering Theory

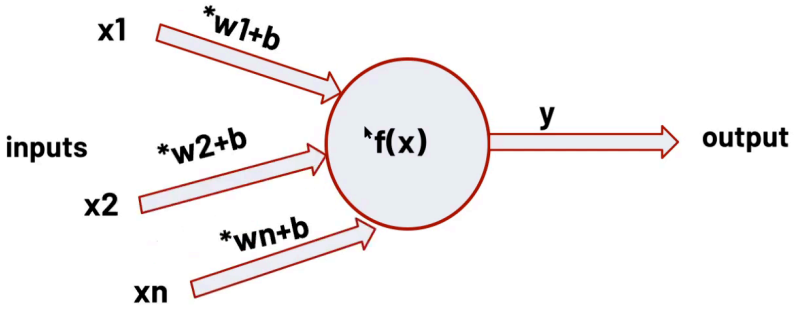
- Hierarchical Clustering is widely used Unsupervised Learning algorithm.

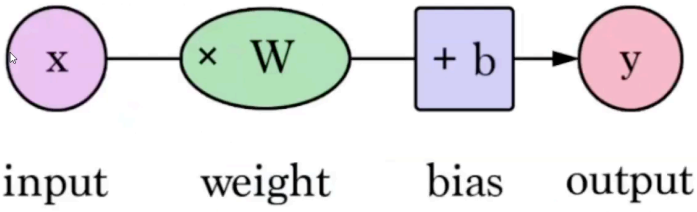


DEEP LEARNING

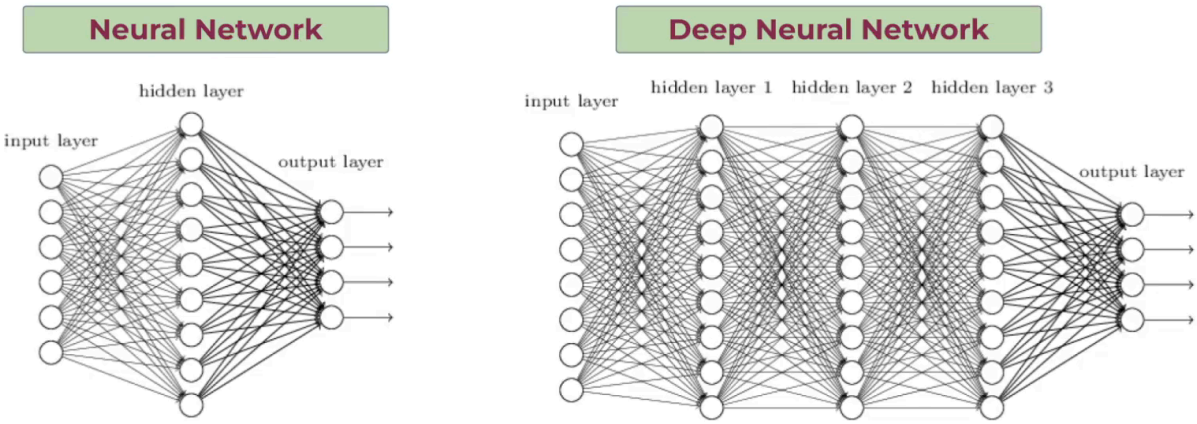
Perceptron Models

Generalization of the formula
y = x1*w1 + x2*w2 + ... + xn*wn + b



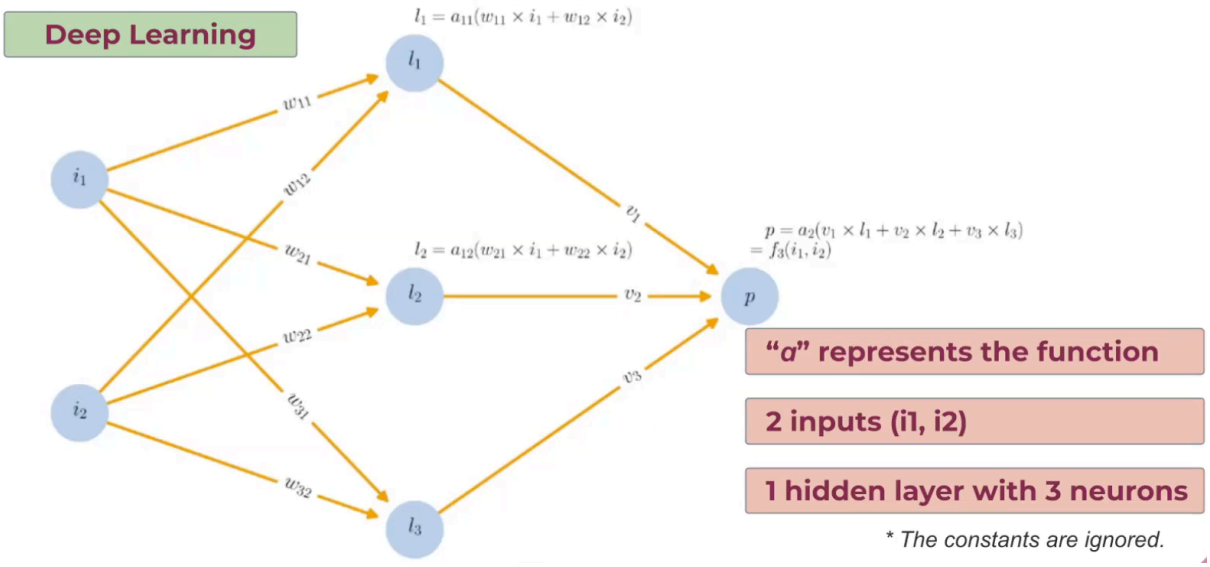


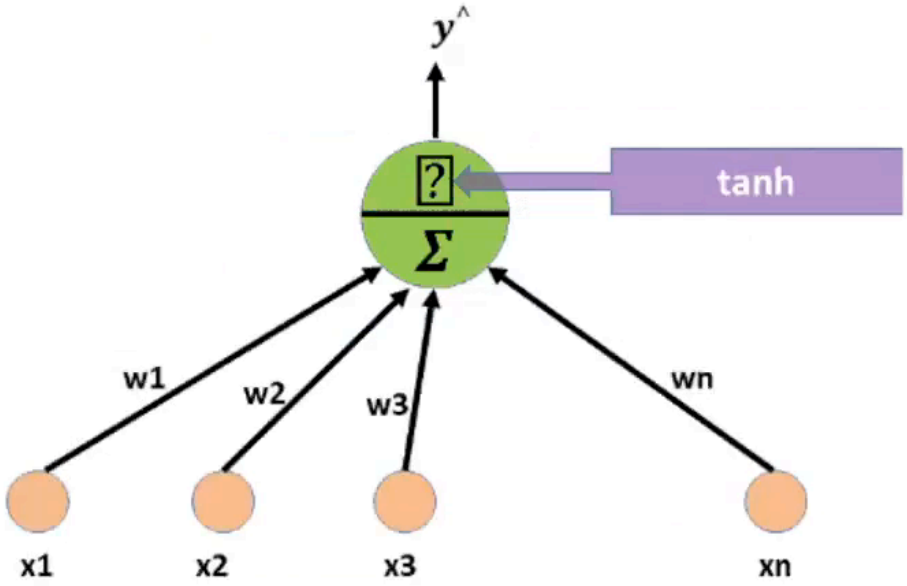
The formula is: $\hat{y} = \sum_{i=1}^{n} x_i w_i + b_i$



The word of “Deep” is related to hidden layers.

While two inputs give one result in the linear regression, deep learning produces 3 different results and then one result with activation functions.



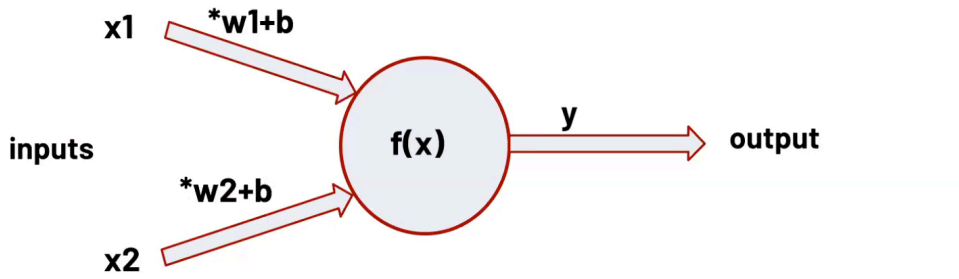


The main goal of the model is to find weights and bias values that cause the minimum errors.

Activation Functions

The activation function of a node defines the output of that node given an input or set of inputs.

Activation functions are decision making units of neural networks.



$$\hat{y} = \sum_{i=1}^{n} x_i w_i + b_i$$

The value of Y can be anything ranging from -inf to +inf. The neuron really doesn't know the bounds of the value.

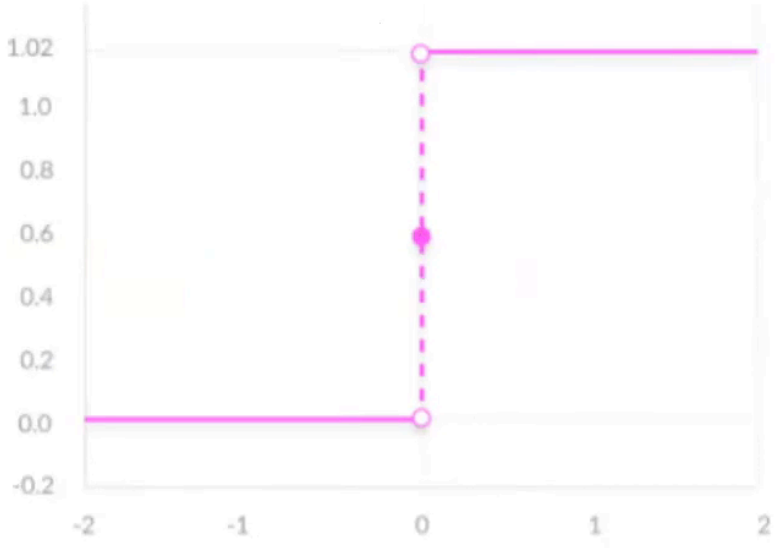
Role of the Activation function

- Activation functions decide if a neuron will be fired or not and strength of the signal outputted by the neuron.

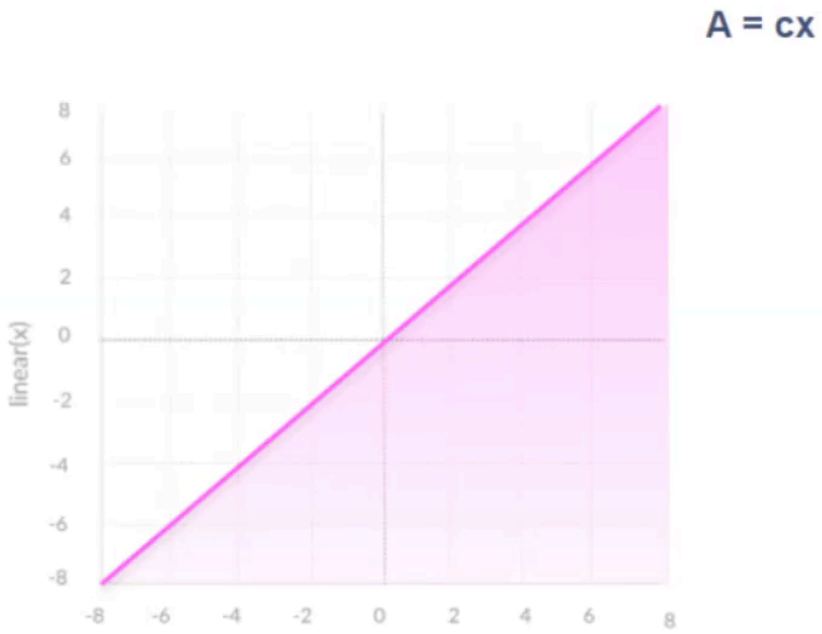
- Activation functions determine the deep learning model's accuracy and computational efficiency for training a model.

Types of Activation Functions

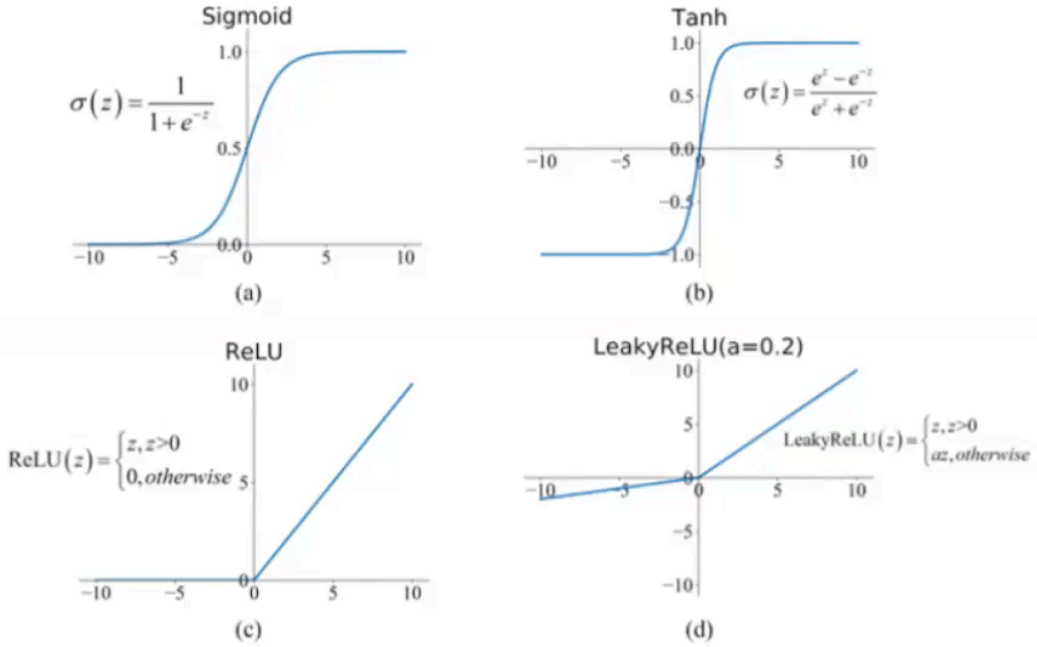
1. Binary Step Function: The function produces binary output. It produces 1 (or True) when input passes threshold limit whereas it produces 0 (or false) when input does not pass threshold. There is no multi classification.



2. Linear Activation Function: It takes inputs, multiplies by the weights for each neuron, and creates an output signal proportional to input. It is suitable for one hidden layer. Backpropagation (öğrenme) is not available.



3. Non-Linear Activation Function: Modern neural network models use non-linear activation functions. It allows backpropagation and multiple hidden layers.



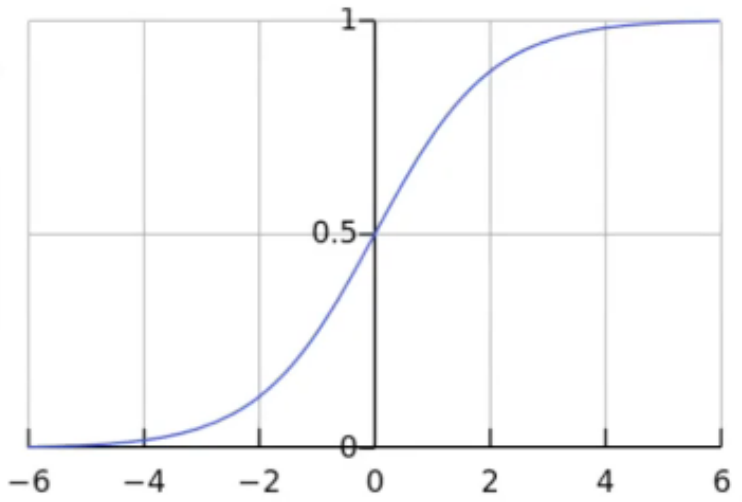
- Sigmoid Activation Function: It bounds output values between 0 and 1.

Pros

- Smooth gradient
- Clear predictions

Cons

- Vanishing gradient
- Outputs not zero centered
- Computationally expensive



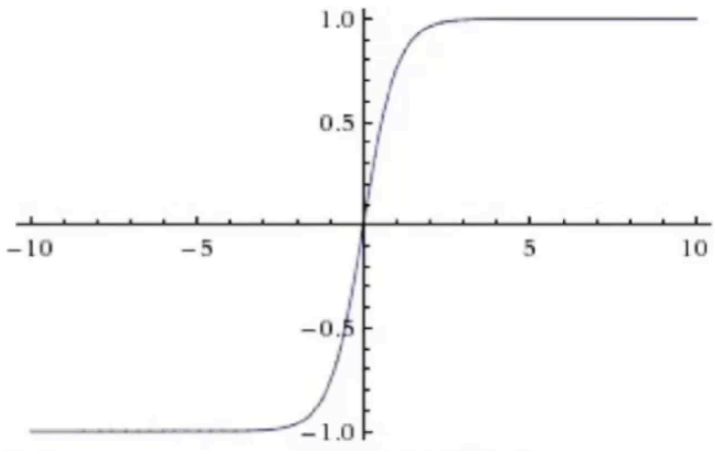
- TanH Activation Function: It bounds output values between -1 and 1.

Pros

- Some advantages with Sigmoid
- More efficient than Sigmoid
- Outputs are zero centered

Cons

- Vanishing gradient
- Computationally expensive



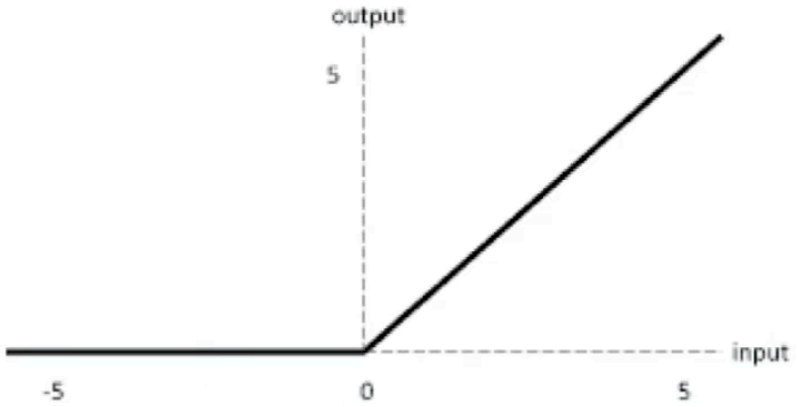
- ReLu Activation Function:

Pros

- Computationally efficient – allows the network to converge very quickly
- Non-linear – although it looks like a linear function, ReLu has a derivative function and allows for backpropagation

Cons

- The Dying ReLu problem - when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.



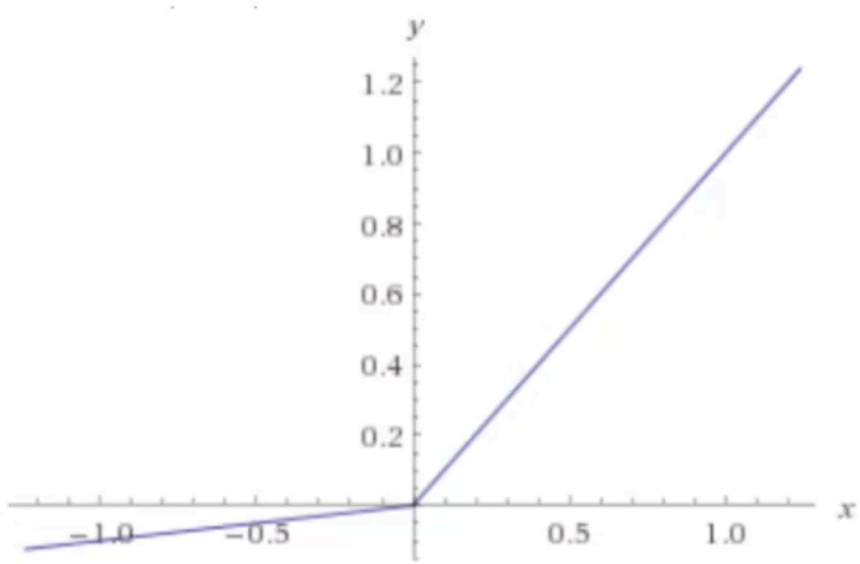
- Leakly ReLu Activation Function:

Pros

- Prevents dying ReLu problem

Cons

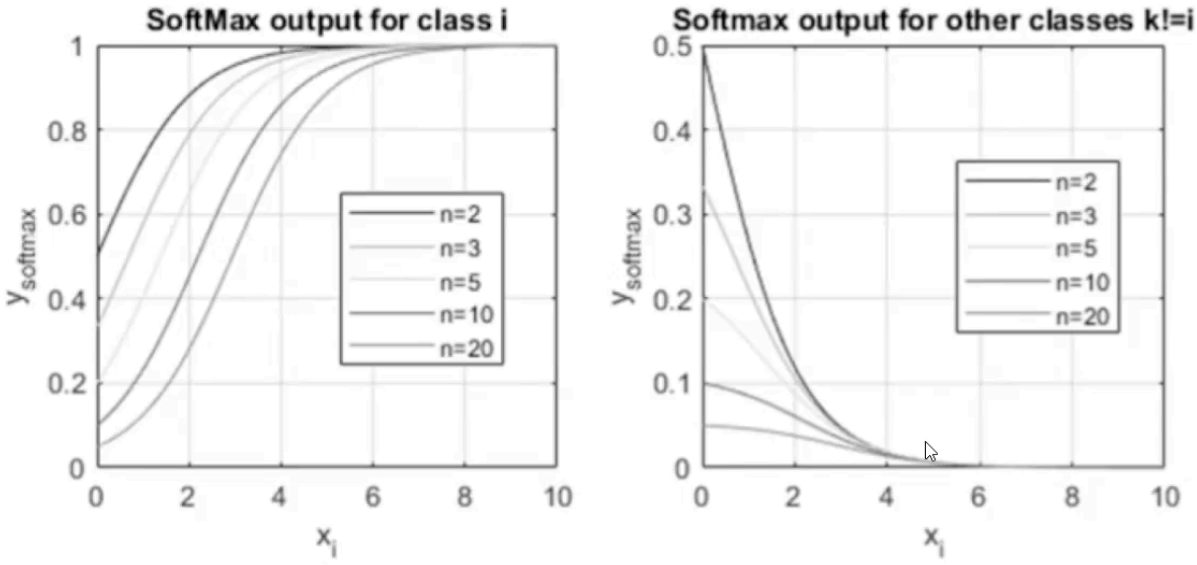
- Results not consistent



- Softmax Activation Function:

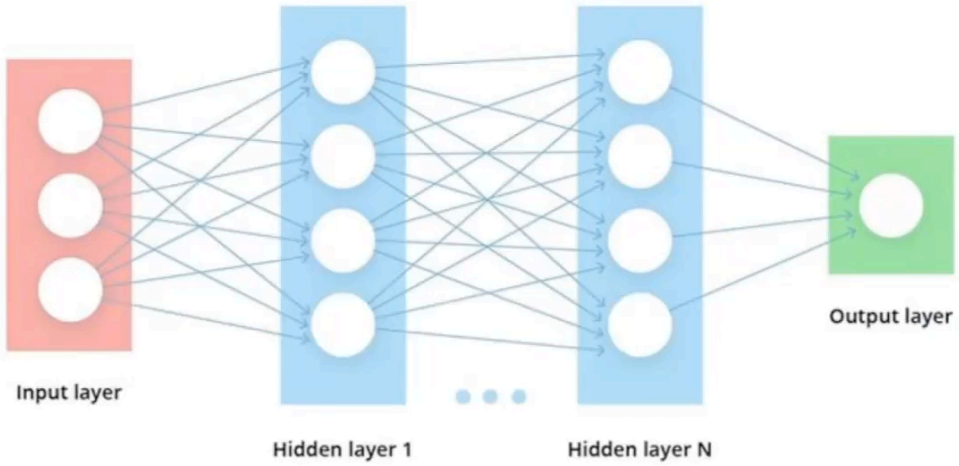
- Able to handle multiple classes.

- Useful for output neurons.

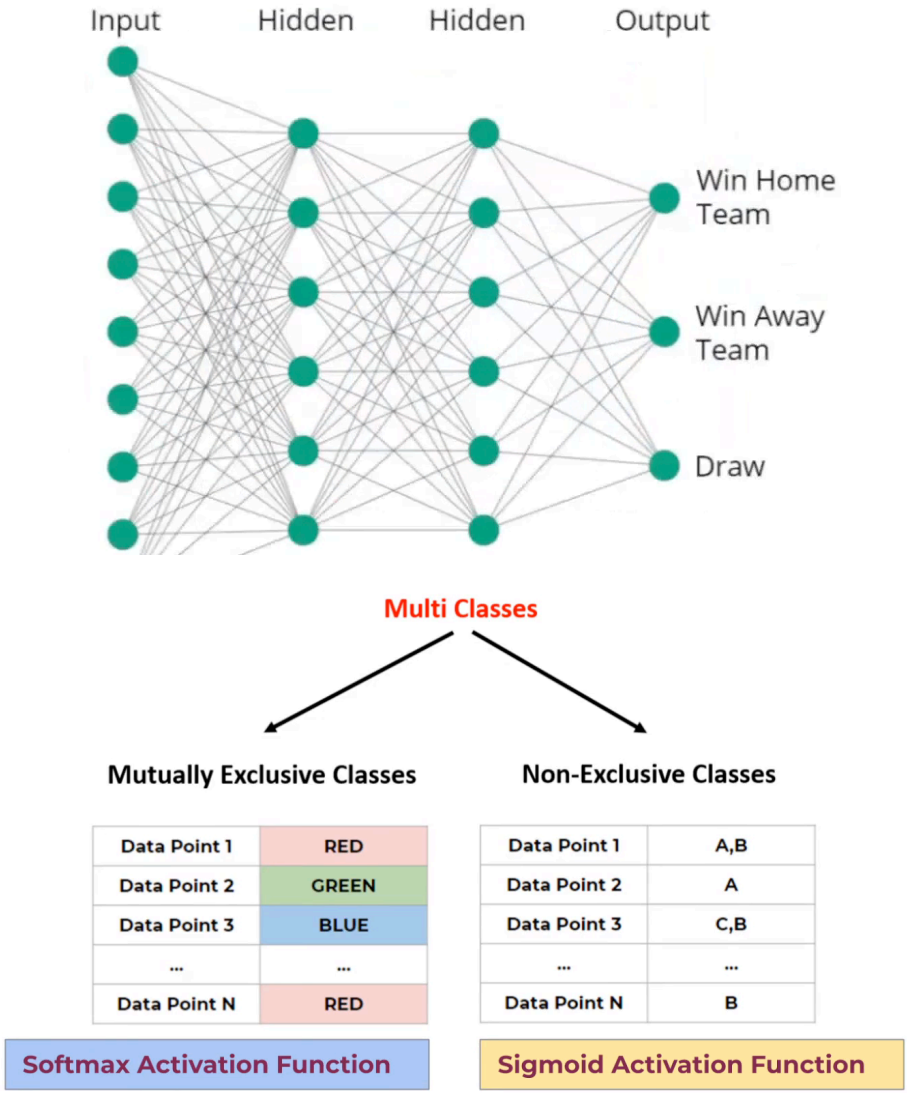


Multiclass Classification

Binary Classification



Multiclass Classification

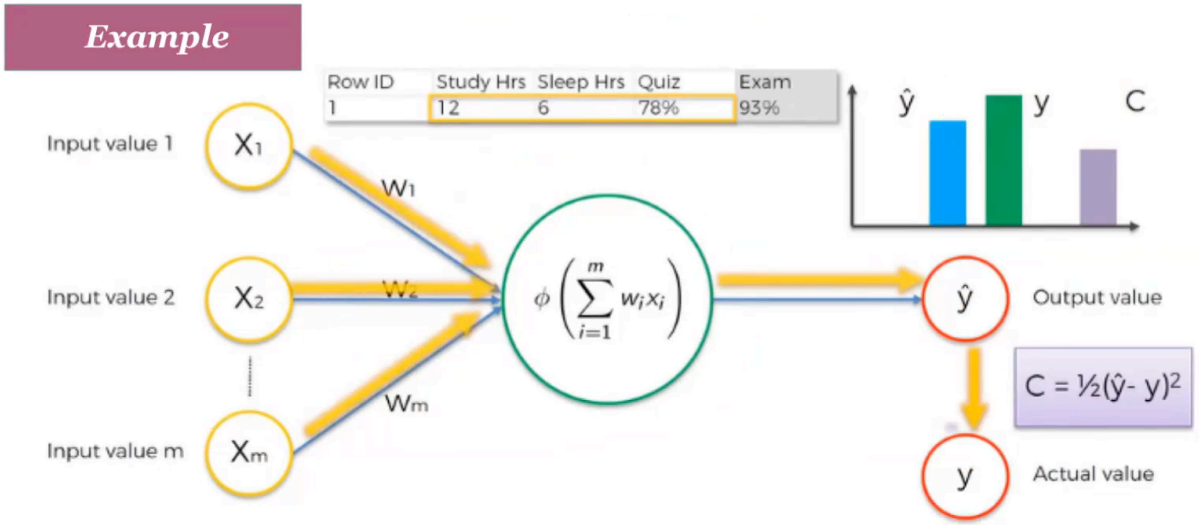


Cost Functions (Loss Function)

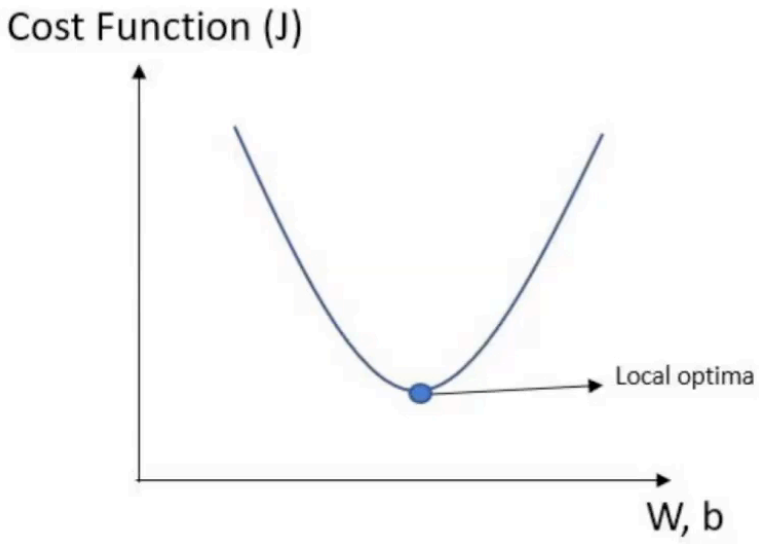
- Loss function is error value of each observation.

- Cost function is average of sum of loss errors. It refers to errors like RMSE, MSE, etc…

Cost func (J) = (Sum of Loss error for m examples) / m



- The objective of a ML model is to find parameters, weights or a structure that minimises the cost function.



Types of Cost Function

Regression Loss Function

1. Mean Squared Error Loss (favoruite)
2. Mean Squared Logarithmic Error Loss
3. Mean Absolute Error Loss

Binary Classification Loss Functions

1. Binary Cross-Entropy (favoruite)
2. Hinge Loss
3. Squared Hinge Loss

Multi-Class Classification Loss Functions

1. Multi-Class Cross-Entropy Loss (favoruite)
2. Sparse Multiclass Cross-Entropy Loss
3. Kullback Leibler Divergence Loss

Problem Type	Output Type	Final Activation Function	Loss Function
Regression	Numerical value	Linear	Mean Squared Error (MSE)
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple labels, multiple classes	Sigmoid	Binary Cross Entropy

In deep learning, training dataset must be trained multiple times (50, 500, 1000 etc…).

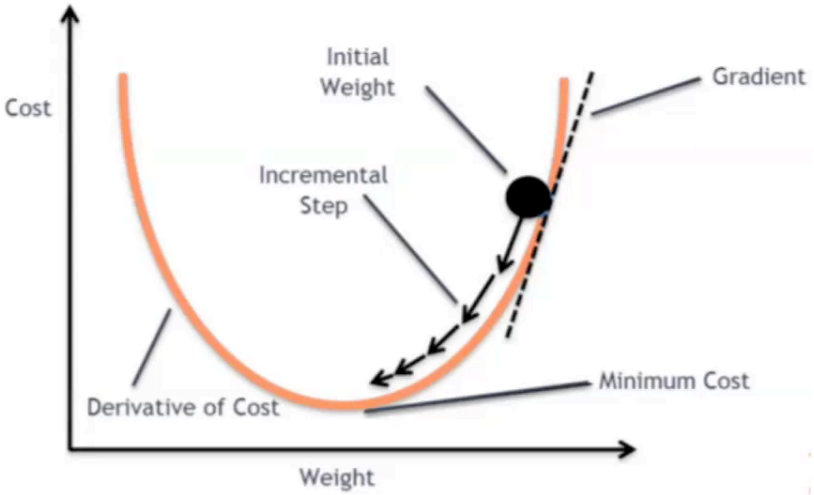
Epoch: One epoch means the entire dataset is passed forward and backward through the neural network once. In below example, dataset was trained for 400 times.

Batchsize: Batchsize refers to the number of training examples utilized in one iteration. It means that the weight is updated for every x observations. In below example, the weights is updated for every 128 rows.

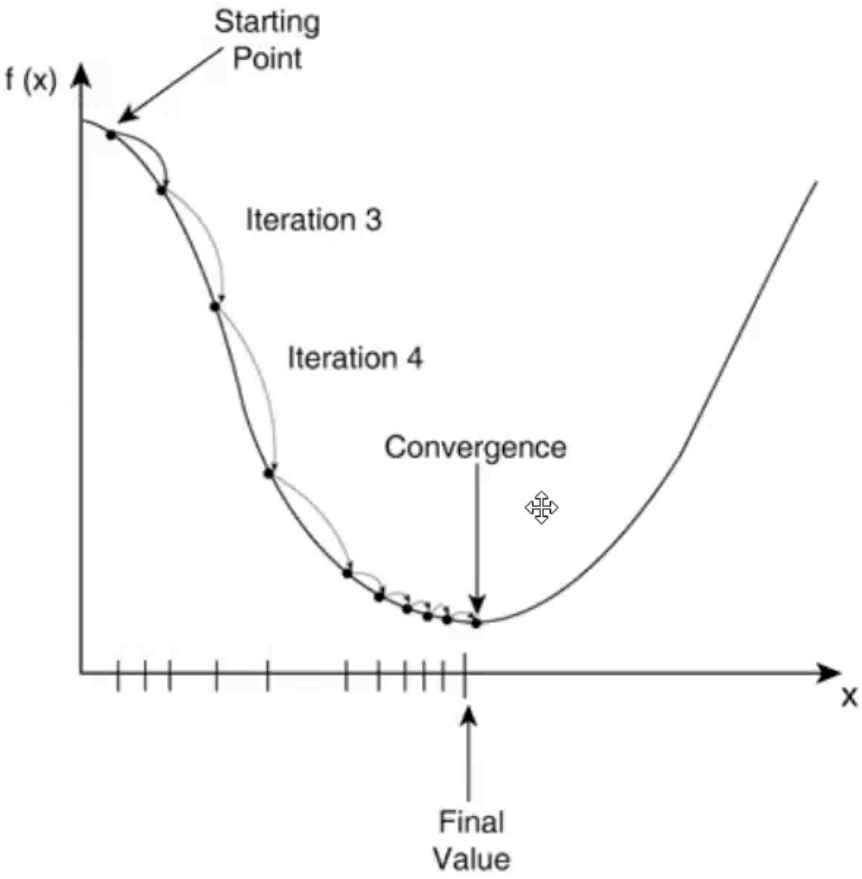
```
model.fit(x=X_train,y=y_train.values,
          validation_data=(X_test,y_test.values),
          batch_size=128,epochs=400)
```

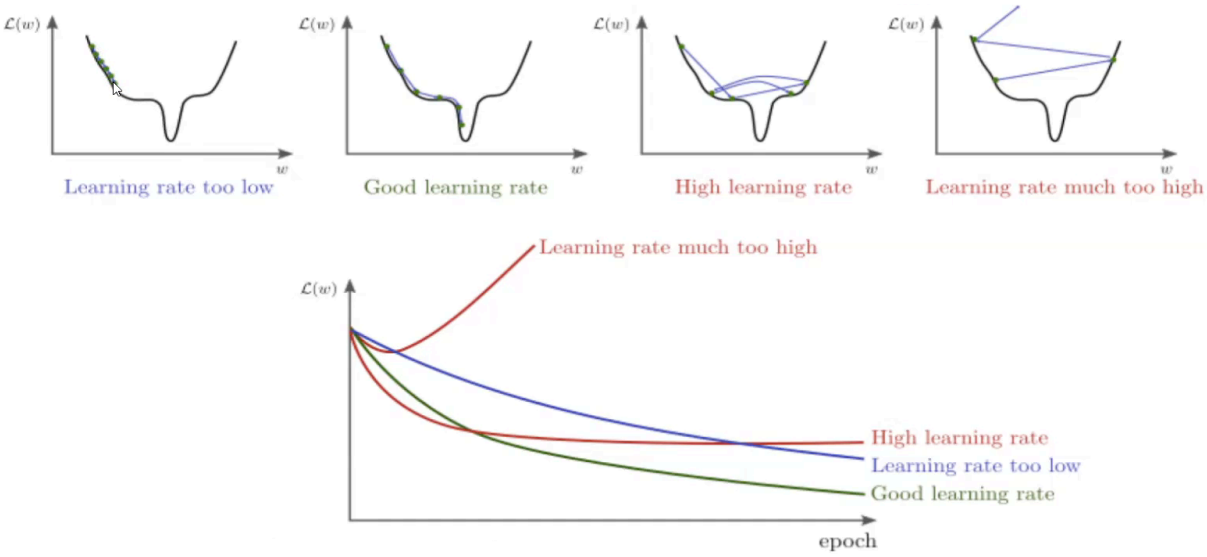
Gradient Descent

Gradient descent is an optimization algorithm used to minimize the cost function.



Step size = slope x learning rate





Gradient Descent (Variants)

Batch (Vanilla) Gradient Descent: Calculates error for each example.

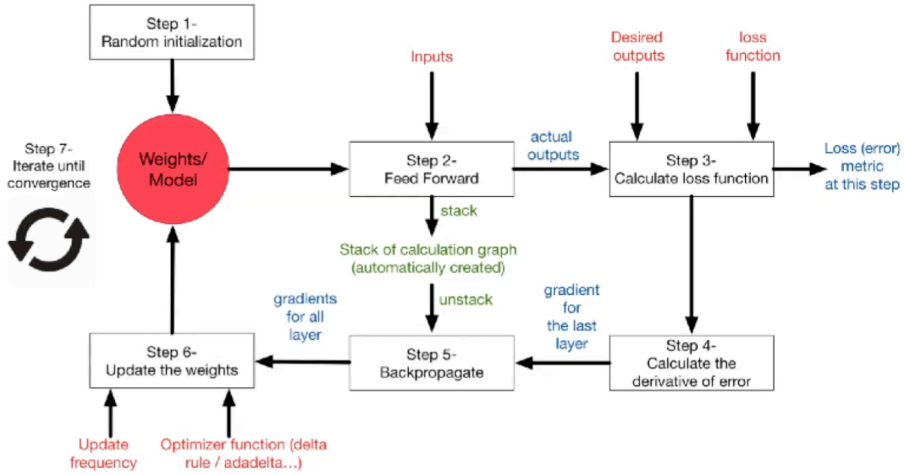
Stochastic Gradient Descent: Iterates over each example while updating the model.

Mini-Batch Gradient Descent: A combination of stochastic and batch gradient descents.

PARAMETERS	BATCH GD ALGORITHM	MINI BATCH ALGORITHM	STOCHASTIC GD ALGORITHM
ACCURACY	HIGH	MODERATE	LOW
TIME CONSUMING	MORE	MODERATE	LESS

Backpropagation

Error is calculated between the expected outputs and the outputs forward propagated from the network. The change in each parameter changes according to the effect of that parameter on the output value.



Keras vs TensorFlow

Keras is a high-level API (Application Programming Interface) that is built on top of TensorFlow. TensorFlow is also the most popular library used to build Deep Learning models.

RNN (Recurrent Neural Network)

- Recurrent Neural Networks (RNNs) are a class of neural networks helpful in modelling sequence data.
- RNNs produce predictive results in sequential data that other algorithms can't.
- RNNs allow you to model memory units to persist data and model short term dependencies.
- It is also used in time-series forecasting fort he identification of data correlations and patterns.

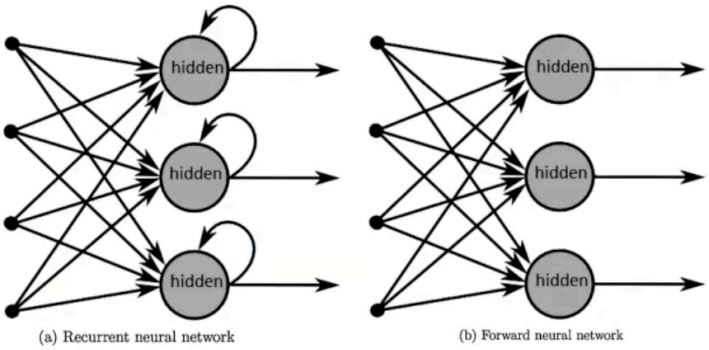
NOTE: Order of data is very very important.

When is RNN used?

Whenever there is a sequence of data and that temporal dynamics that connects the data is more important than the spatial content of each individual frame.

Use Cases: chat bots, music syntheses, machine translation, stock predictions, weather prediction

RNN vs Feed-Forward Neural Network



(a) Recurrent neural network (b) Forward neural network

	RNN	ANN
Data	Sequence	Tabular
Recurrent Connections	+	-
Memory	+	-

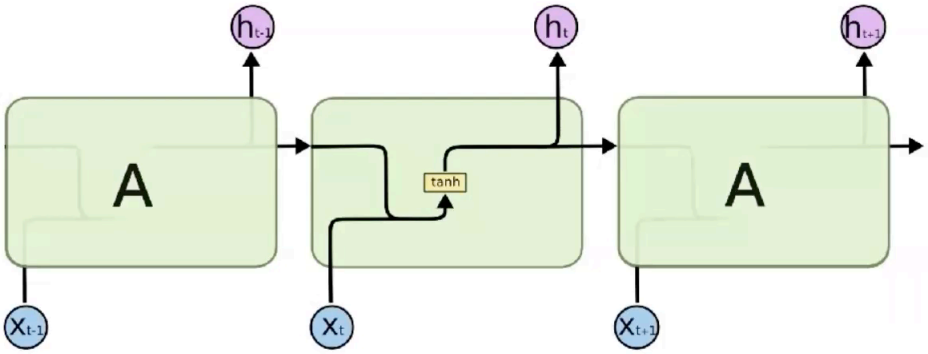
RNN is feeded by previous data.

Types of RNNs

Type	Describe	Explanation	Example
Many to Many	Sequence to Sequence	Given 5 previous words, predict the next 5	Machine Translation, Name Entity Recognition
Many to One	Sequence to Vector	Given 5 previous words, predict next word	Text Generation, Sentiment Analysis
One to Many	Vector to Sequence	Given 1 word, predict the next 5 words	Image Captioning, Music Generation

Many to one is the most common usage aim of RNN.

How does it work?

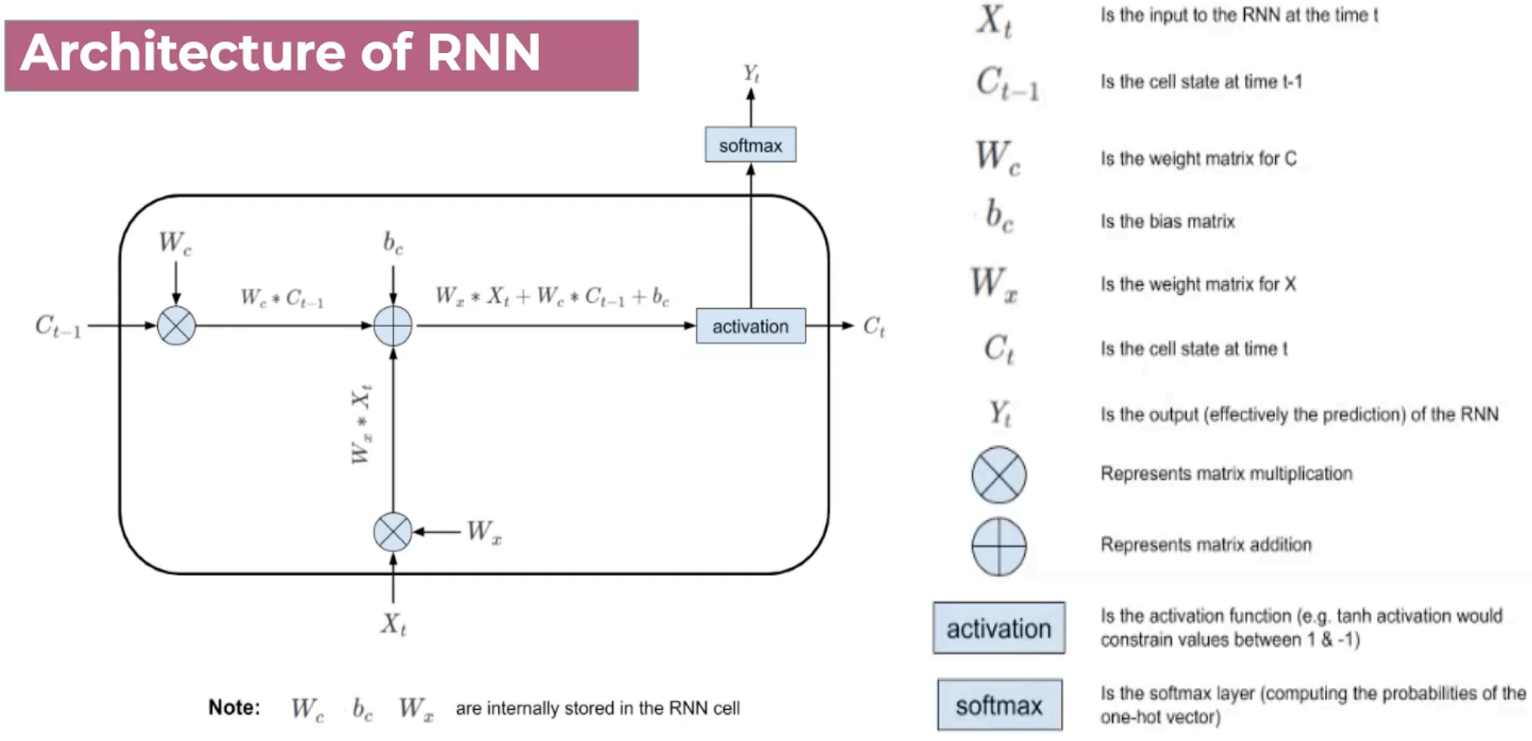


- The network is called 'recurrent' it performs the same operation in each activate square.
- The network computed the weights of the inputs and the previous output before to use an activation function.
- recurrent neuron is a function of all the inputs of the previous time steps. This is how the network build its own memory.
- The information from the previous time can propagate in future time.

h: hidden state

- The hidden state acts as the neural networks memory. It holds information on previous data the network has seen before.

Architecture of RNN



Pros of RNN

- RNN can model a collection of records (i.e. time collection) so that each pattern can be assumed to be dependent on previous ones.
- RNNs are even used with convolutional layers to extend the powerful pixel neighbourhood.

Cons of RNN

- Training an RNN is a completely tough task.
- Difficult to process longer sequences.
- Gradient exploding and vanishing problems.
- Short term memory.

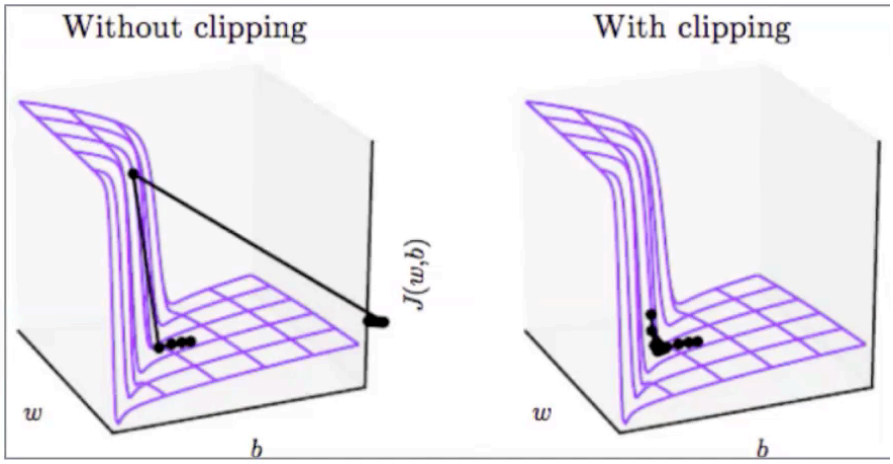
Exploding and Vanishing Graident

As the depth of model increases, the probability of exploiding (patlama) and vanishing (sönümlenme) gradient increases.

new weight = weight – learning rate * gradient

Exploding gradient: During backpropagation over time, the gradient can both fall and rise exponentially. These expanding gradients can be terrible for our networks since they can lead our parameters to grow so huge that our network just goes crazy!

Solution: Gradient clipping can be used to alleviate the problem of exploding gradients. This is just scaling or rescaling our gradient vectors when they hit a threshold or maximum value.



Vanishing gradient: Rate of change of error with respect to weight is propagated back to the time step network to minimize the cost function. Gradient in backward direction is calculated as product of derivatives of act. Func. Whose magnitude is high in the hidden layers to the right and less in the hidden layer in the left. So, when moving in the left, gradient decreases and hence gradient is vanished.

- Layers that get a small gradient update stops learning.
- Those are usually the earlier layers.
- RNNs can forget what it seen in longer sequences, thus having a short-term memory.

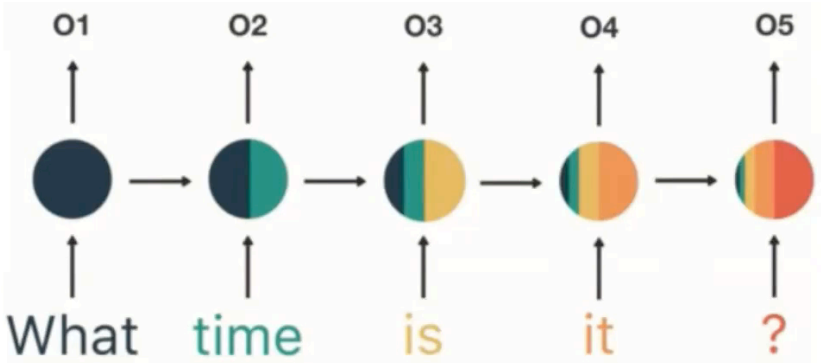
Solution:

- Weight initialization
- Batch normalization
    - Technique for training very deep neural networks
    - Faster training
    - Higher learning rate
    - Stabilizes the learning process
    - Reduces teh number of training epochs
    - After the activation function for s-shaped functions (sigmoid, tanh)
    - Before the activation function for relu, leakly relu

- Using activation function apart from “sigmoid” and “tahn” (ReLU)
- Using LSTM or GRU Networks

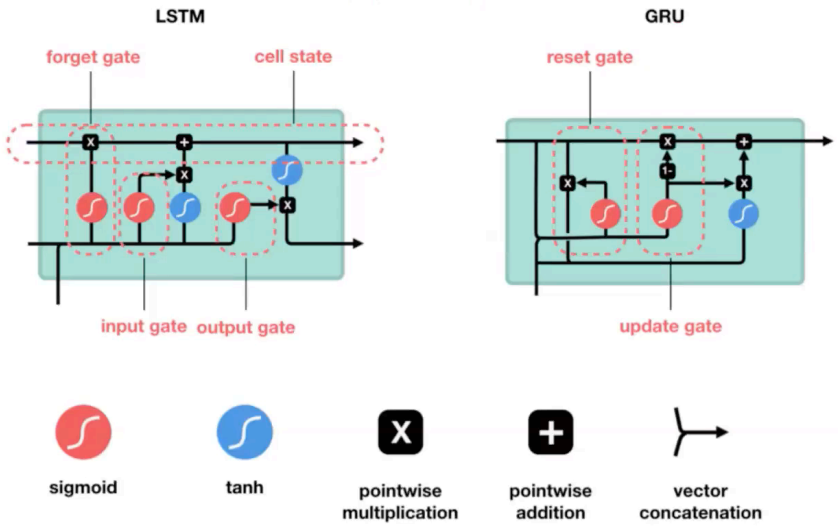
Note: Daha çok baştaki katmanlada görülür.

Short Term Memory



- Short term memory refers to forget initial inputs.
- As the RNN processes more steps, it has troubles retraining information from previous steps.
- The bigger gradient, the bigger adjustments
- Solution is LSTM & GRU

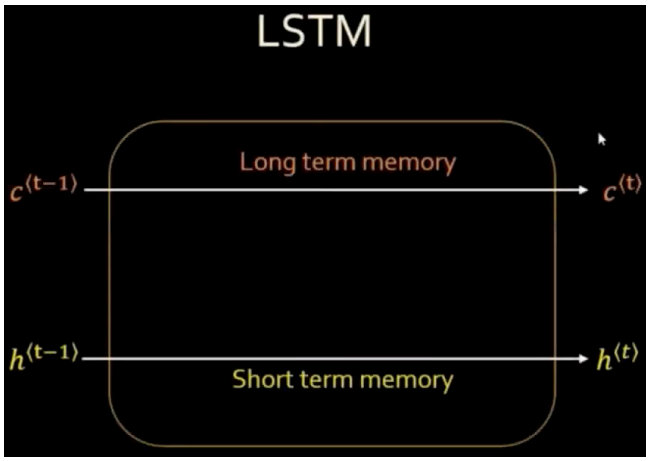
LSTM and GRU



LSTM	GRU
3 Gates: Input, output, forget	2 Gates: reset, update
More accurate on longer sequence, less efficient	More efficient computation wise. Getting more popular
Invented: 1995 - 1997	Invented: 2014

- They have internal mechanism called Gates that can regulate the flow of information.
- These Gates can learn which data in a sequence is important to keep or throw away.

LSTM (Long Short Term Memory)



Cell State: Carries relevant information throughout the processing of the sequence (memory).

Forget Gate: Decides what is relevant to keep from prior steps.

Input Gate: Decides what information is relevant to add from the current step.

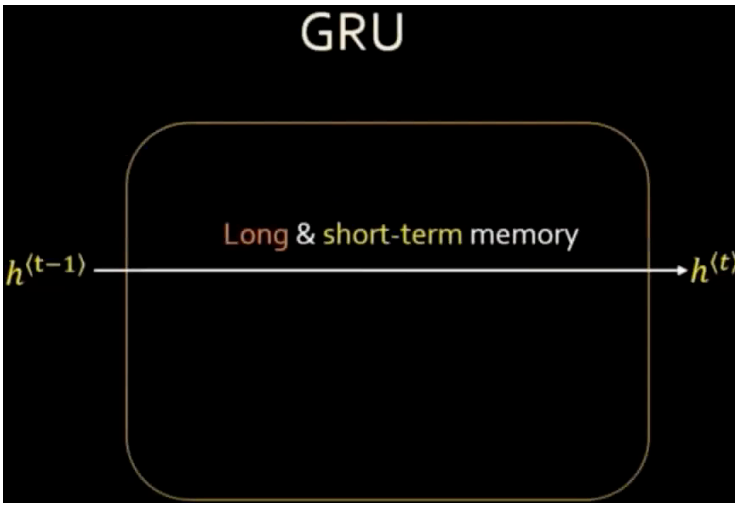
Outpt Gate: Determines what the next cell state should be.

Bidirectional LSTM

- Bidirectional LSTM train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence.

- This can provide additional context to the network and result in faster and even fuler learning on the problem.

GRU (Gated Recurrent Unit)



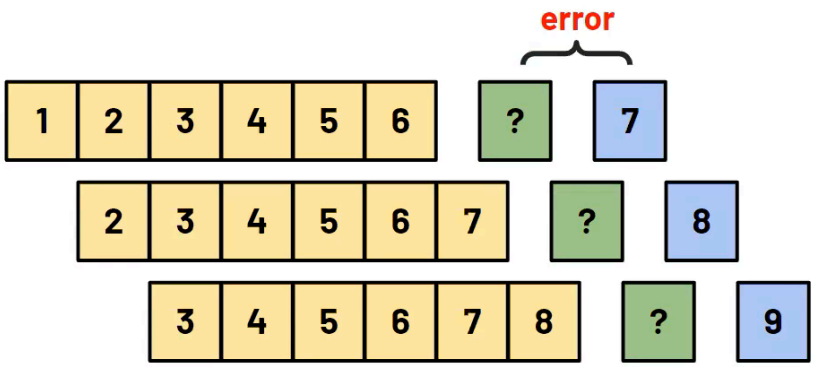
Hidden State: Carries relevant information throughout the processing of the sequence (memory).

Update Gate: Acts similar to the forget and input gate of an LSTM. It decides what information to throw away and what new information to add.

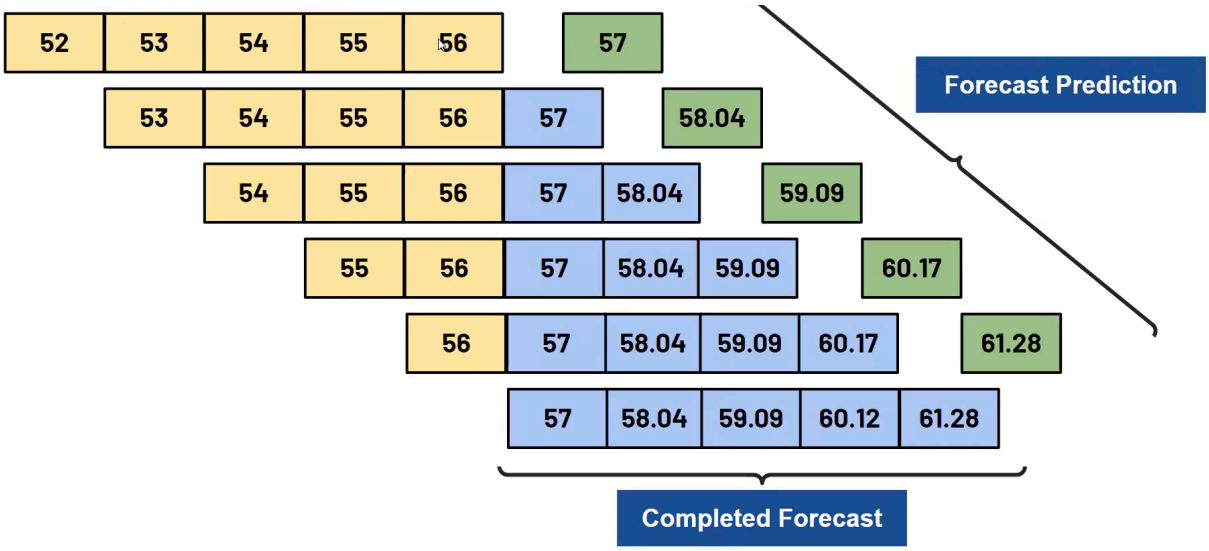
Reset Gate: Decides how much past information to forget.

RNN Training and Forecasting

Training



Forecasting



Overview

- RNNs are good for processing sequence data for predictions but suffers from short-term memory.

- LSTM and GRU were cerated as a method to mitigate short-term memory using mechanisms called gates.

- Gates and just neural networks that regulate the flow of information flowing through the sequence chain.

- LSTM and GRU are used in state of the art deep learning applications like speech recognition, speech synthesis, natural language understanding, etc.