UNIT -I

UNIT - 1 Syllabus

- Machine Learning: What and Why?
- Types of Machine Learning
- Supervised Learning
- Unsupervised Learning
- Reinforcement learning
- The Curse of dimensionality
- Over fitting and under fitting
- Linear regression
- Bias and Variance tradeoff
- Testing – cross validation
- Regularization
- Learning Curve
- Classification
- Error and noise
- Parametric vs. non-parametric models
- Linear Algebra for machine learning

Learning Algorithms

We are drowning in information and starving for knowledge. — John Naisbitt

We in the era of big data.

- Ex: There are about 1 trillion web pages1;
- One hour of video is uploaded to YouTube every second

This deluge of data, calls for automated methods of data analysis called Machine Learning.

Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty

Machine learning is programming computers to optimize a performance criterion using example data or past experience.

Learning is used when:

- Human expertise does not exist (navigating on Mars),
- Humans are unable to explain their expertise (speech recognition)
- Solution changes in time (routing on a computer network)
- Solution needs to be adapted to particular cases (user biometrics)

What is Machine Learning?

Definition by Samuel – in 1959

- A field of study that gives computers the ability to learn without being explicitly programmed.

Machine Learning

- Study of algorithms that
- improve their performance
- at some task
- with experience

Role of Statistics: Inference from a sample

Role of Computer science: Efficient algorithms to

- Solve the optimization problem
- Representing and evaluating the model for inference

Common Terminologies in ML

Vector Feature – It is n dimensional vector of numerical features that represent some object.

Samples – They are the items to process

Feature Space – It refers to the collections of features that are used to characterize your data

Labeled Data – It is the data with known classification results.

Growth of Machine Learning

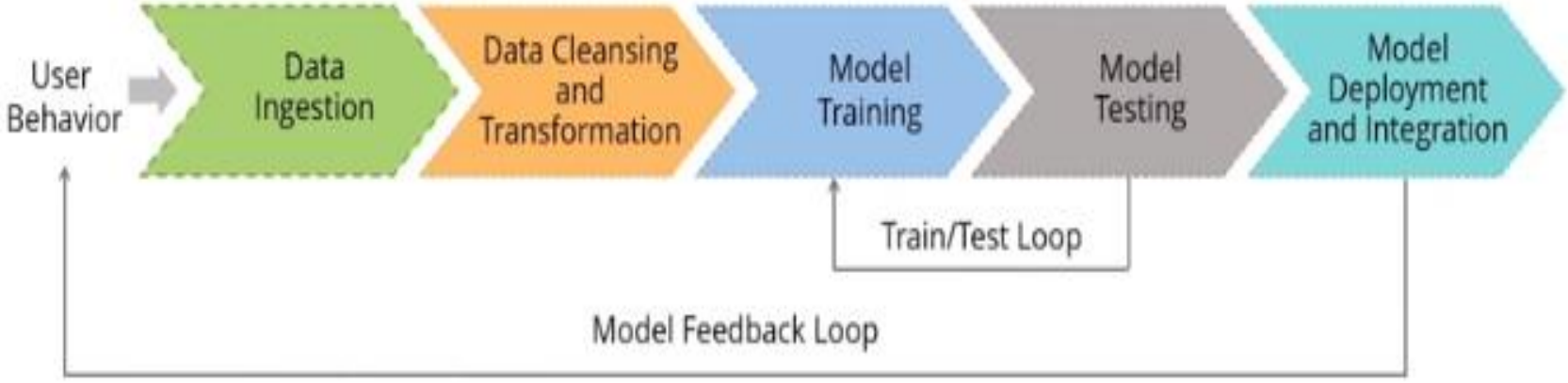
Machine learning is preferred approach to

- Speech recognition, Natural language processing
- Computer vision
- Medical outcomes analysis
- Robot control
- Computational biology

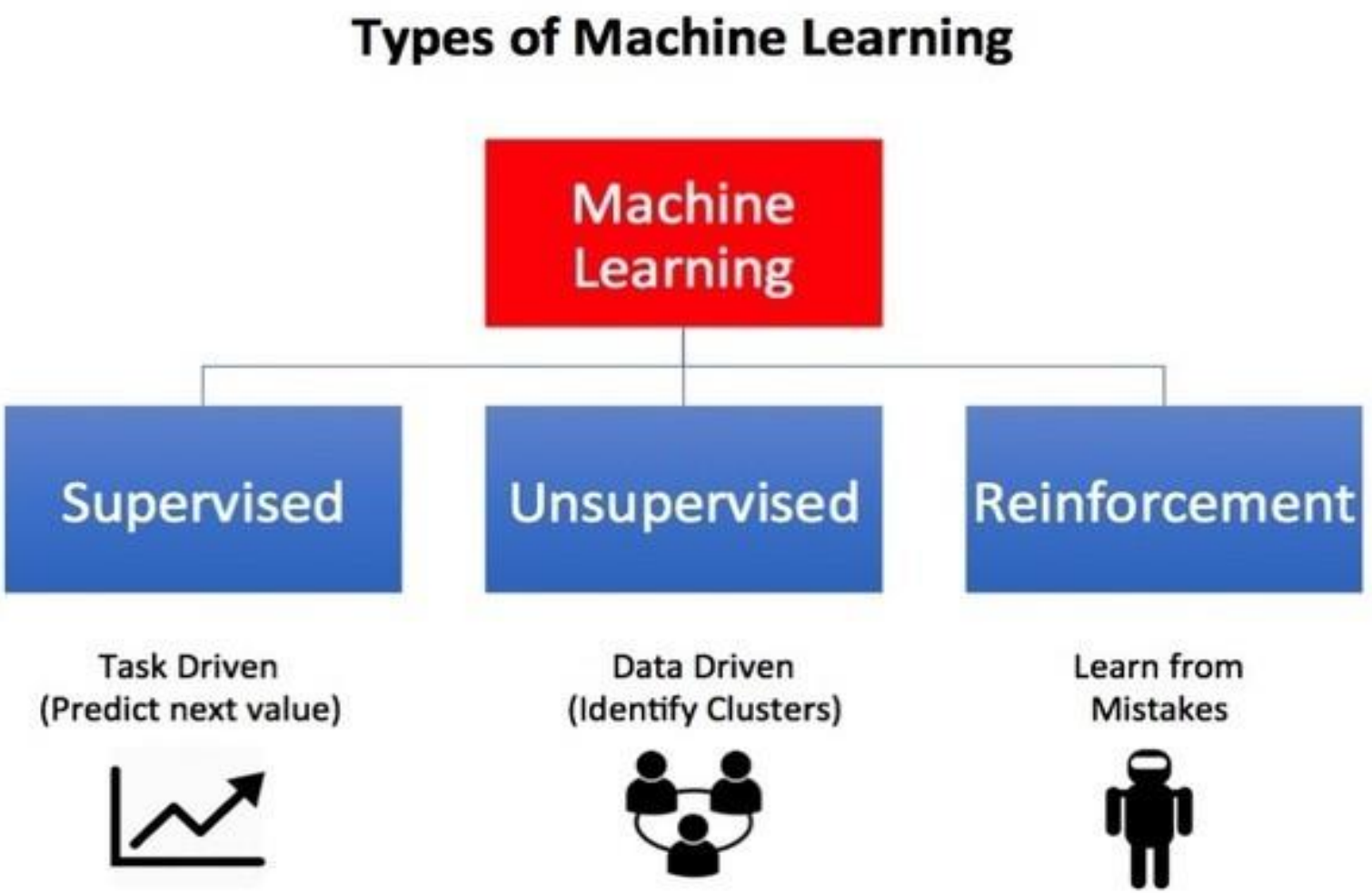
This trend is accelerating

- Improved machine learning algorithms
- Improved data capture, networking, faster computers
- Software too complex to write by hand
- New sensors / IO devices
- Demand for self-customization to user, environment
- It turns out to be difficult to extract knowledge from human experts→*failure of expert systems in the 1980's.*

Steps of General Machine Learning Pipeline



Types of Machine Learning



Supervised Learning/ predictive

It is similar to teaching a child with the use of flash cards.

Supervised Learning

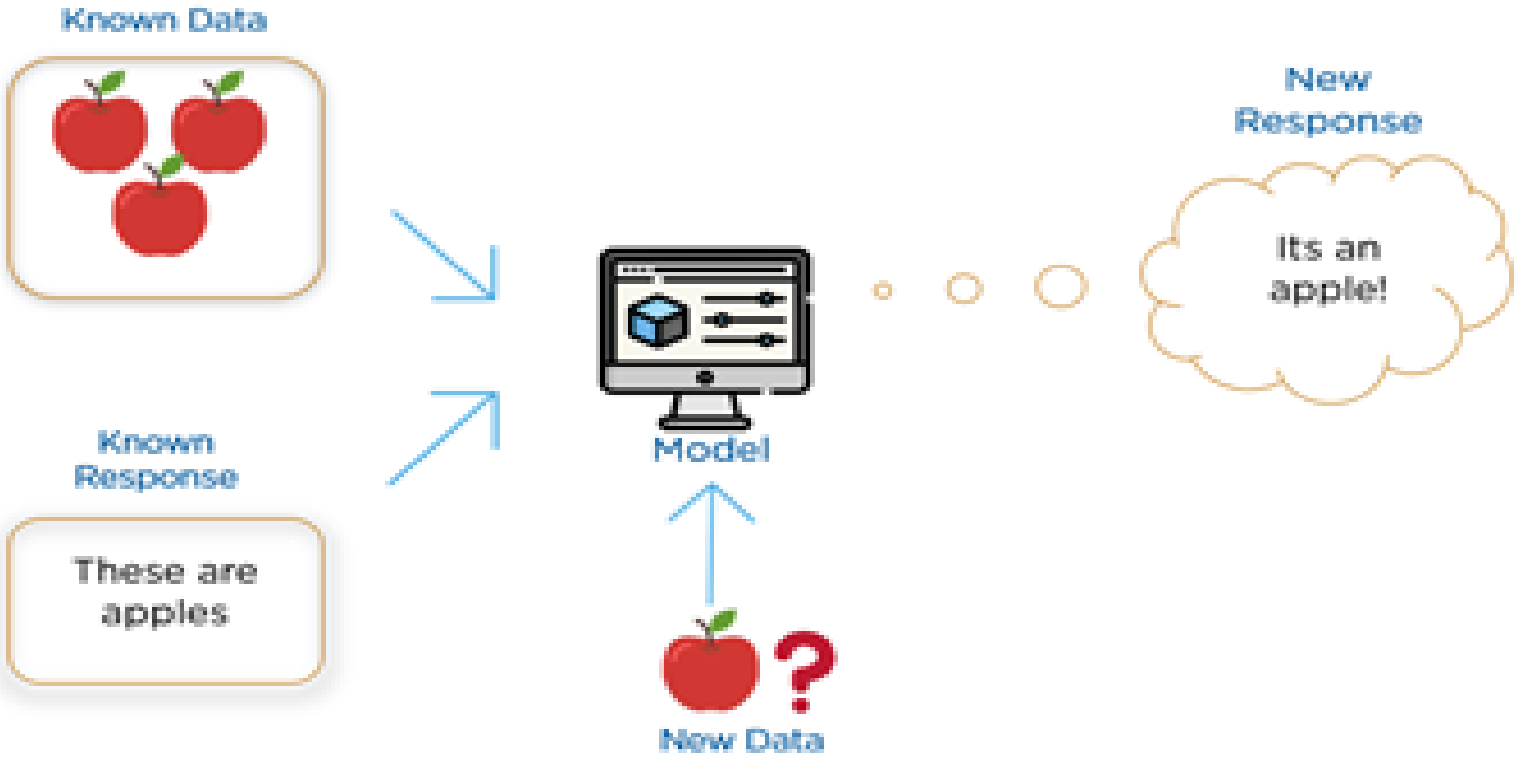
we feed a learning algorithm with example-label pairs one by one. (Traiing Data Set)

- D = {(xi, yi)}Ni where D is called the **training set**, and N is the number of training examples.
- Xi – Features / Attributes
- Yi – response Variable
- **Yi may be categorical** or **nominal**

allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not (Test Data Set)

Over time, the algorithm will learn to approximate the exact nature of the relationship between examples and their labels.

Supervised Learning



Types of Supervised learning:

- Classification
- Regression

Classification - is used when the output variable is categorical i.e. with 2 or more classes.

- Ex: yes or no, male or female, true or false, etc.

Regression - is used when the output variable is a real or continuous value. i.e., a change in one variable is associated with a change in the other variable.

- Ex: salary based on work experience, Price of car based on features etc

Real-Life Applications

Risk Assessment - Assess the risk in financial services or insurance domains in order to minimize the risk portfolio of the companies.

Image Classification – Ex : Facebook can recognize your friend in a picture from an album of tagged photos

Fraud Detection - Identify whether the transactions made by the user are authentic or not.

Visual Recognition - The ability of a machine learning model to identify objects, places, people, actions, and images.

Unsupervised Learning

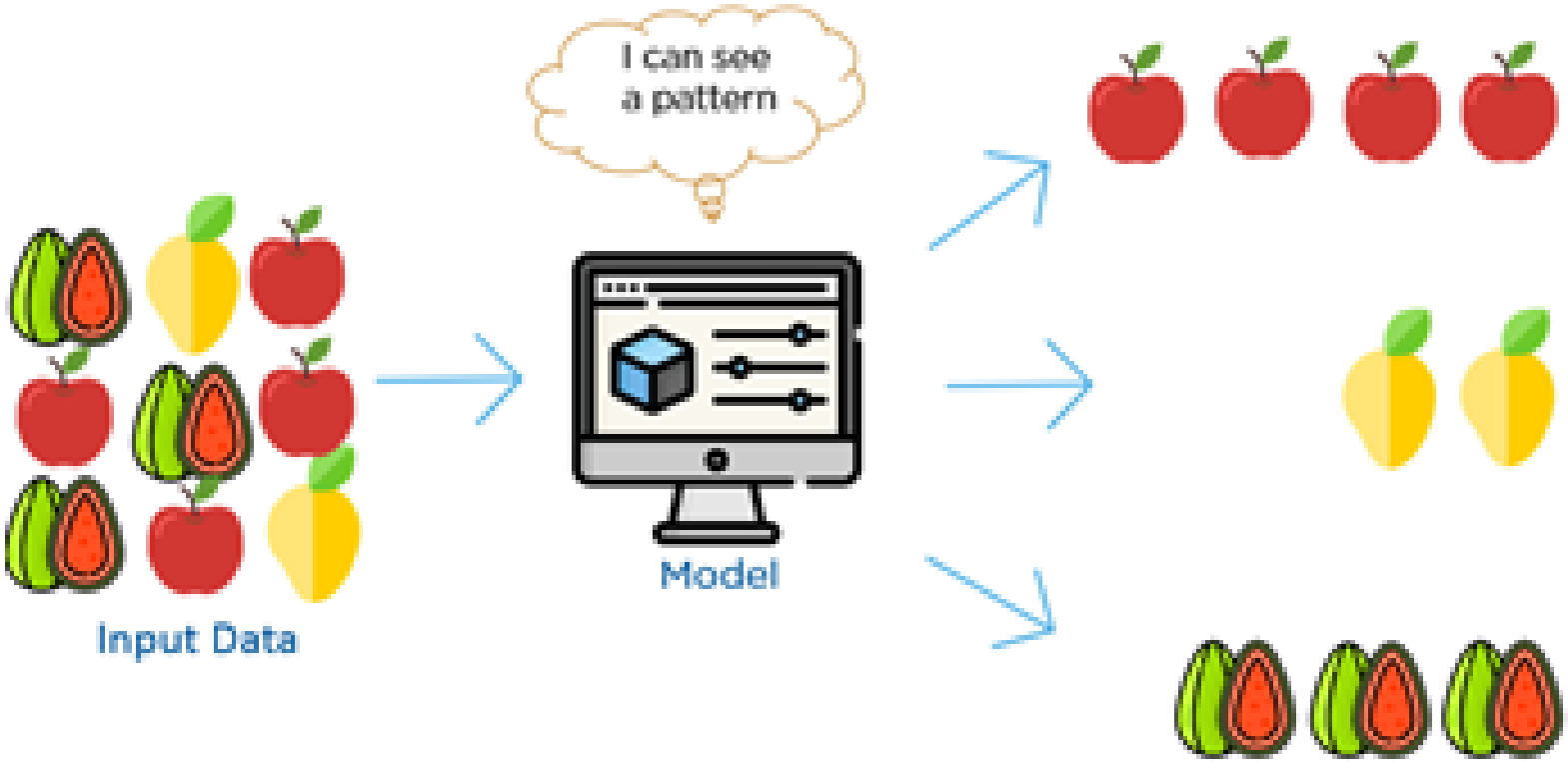
It is opposite of supervised learning.

It features has no labels.

The algorithm would be fed with lot of data and given the tools to understand the properties of the data.

From there, it can learn to group, cluster, and/or organize the data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data.

Unsupervised Learning



Types of Unsupervised learning :

- Clustering
- Association

Clustering - is the method of dividing the objects into clusters that are similar between them and are dissimilar to the objects belonging to another cluster.

- Ex: In Identification of Cancer Cells. It divides the cancerous and non-cancerous data sets into different groups.

Association is a rule-based machine learning to discover the probability of the co-occurrence of items in a collection.

- Ex: finding out which products were purchased together.

Real-Life Applications

Market Basket Analysis – Ex: if you buy a certain group of items, you are less or more likely to buy another group of items.

Semantic Clustering – Ex: People post their queries on websites in their own ways. Semantic clustering groups all these responses with the same meaning in a cluster

Delivery Store Optimization - predict the demand and keep up with supply.

Identifying Accident Prone Areas - based on the intensity of accidents it can grouped as accident prone area or not

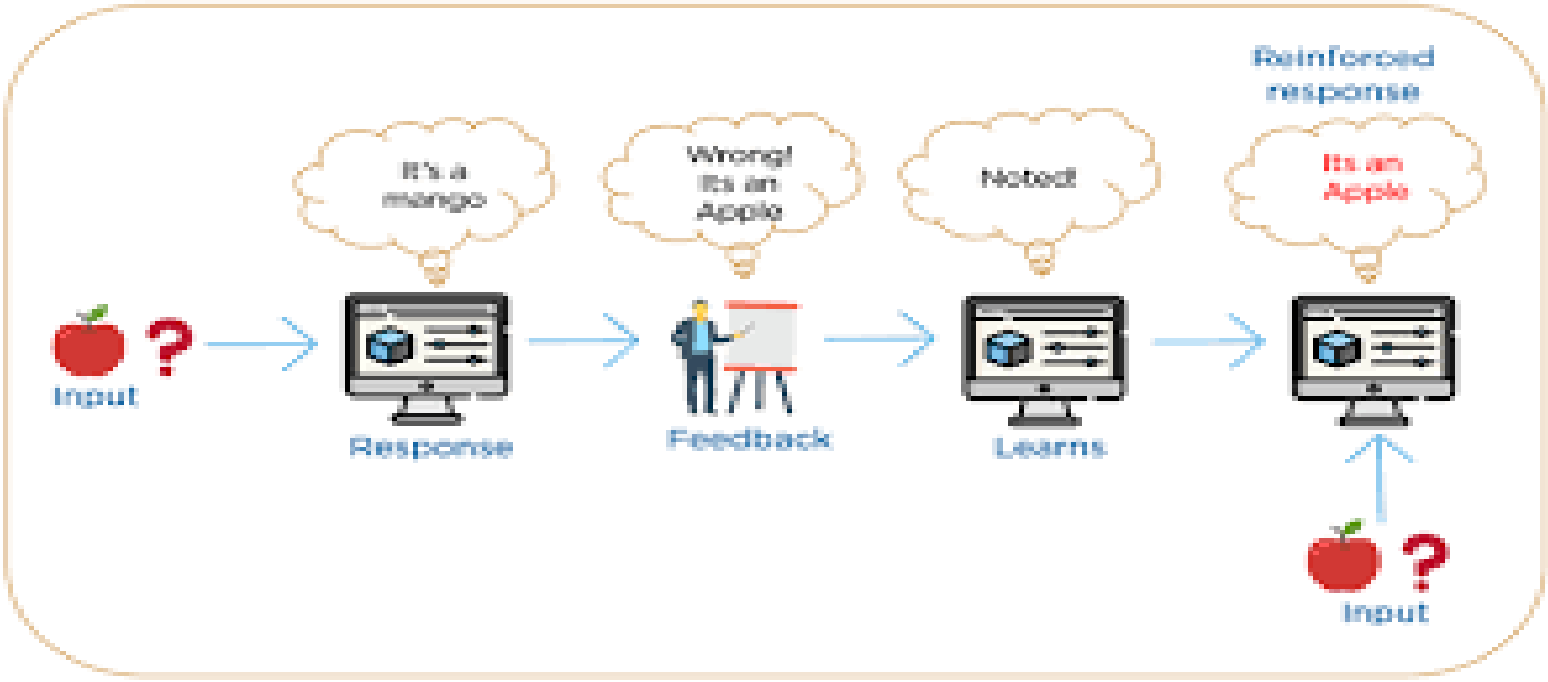
Difference Between Supervised and Unsupervised Learning

Supervised Learning	Unsupervised Learning
It uses known and labeled data as input	It uses unlabeled data as input
It has a feedback mechanism	It has no feedback mechanism
The most commonly used supervised learning algorithms are: • Decision tree • Logistic regression • Support vector machine	The most commonly used unsupervised learning algorithms are: • K-means clustering • Hierarchical clustering • Apriori algorithm

Reinforcement Learning

reinforcement learning s learning from mistakes

Reinforcement learning is behavior driven



Reinforcement Learning

Video Games

Industrial Simulation (Robotic Applications)

Resource Management

Curse of Dimensionality

CURSE OF DIMENSIONALITY

The dimension of a dataset corresponds to the number of attributes/features that exist in a dataset.

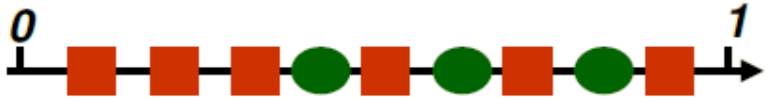
A dataset with a large number of attributes, generally of the order of a hundred or more, is referred to as high dimensional data.

The difficulties related to training machine learning models due to high dimensional data is referred to as ‘Curse of Dimensionality

It refers to a set of problems that arise when working with high-dimensional data

Curse of Dimensionality: Number of Samples

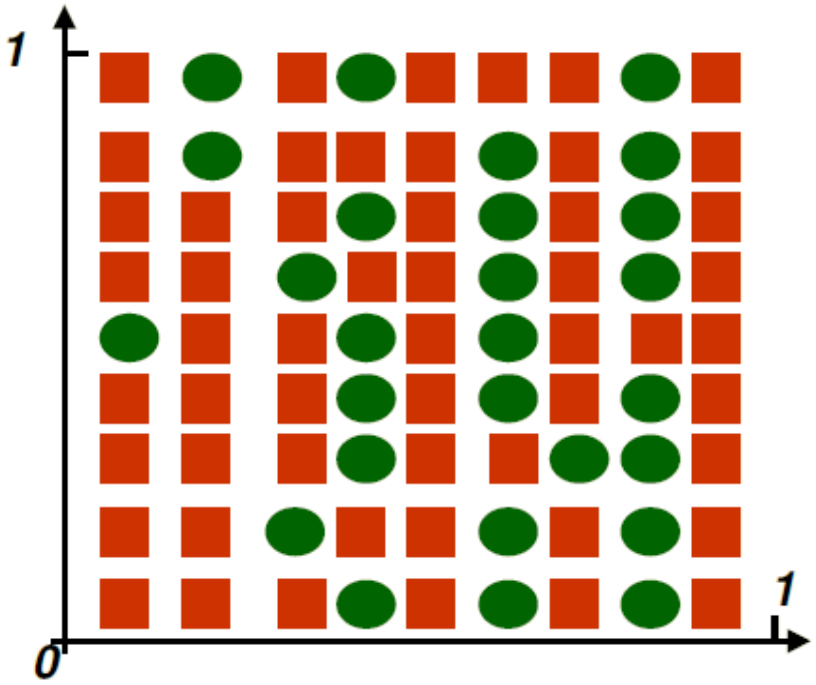
- Suppose we want to use the nearest neighbor approach with ***k*** = 1 (***1NN***)
- Suppose we start with only one feature



- This feature is not discriminative, i.e. it does not separate the classes well
- We decide to use 2 features. For the 1NN method to work well, need a lot of samples, i.e. samples have to be dense
- To maintain the same density as in 1D (9 samples per unit length), how many samples do we need?

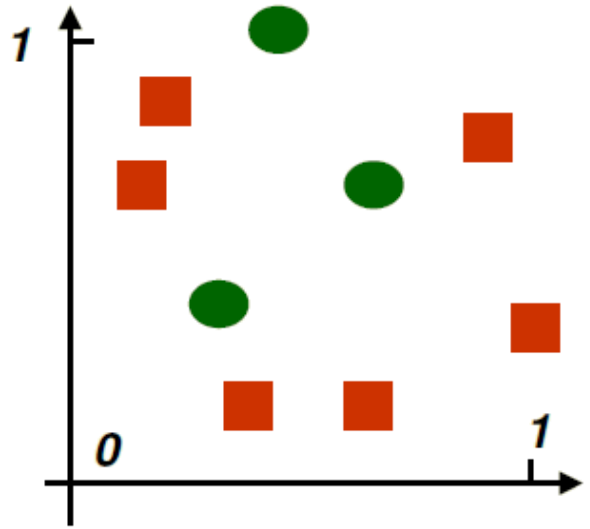
Curse of Dimensionality: Number of Samples

- We need 9^2 samples to maintain the same density as in *1D*



Curse of Dimensionality: Number of Samples

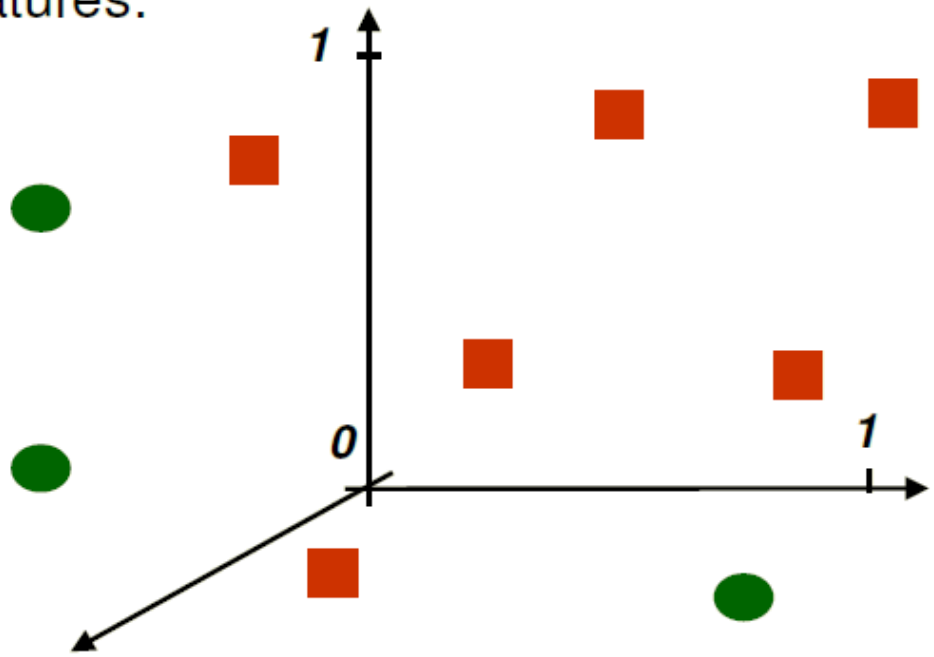
- Of course, when we go from 1 feature to 2, no one gives us more samples, we still have 9



- This is way too sparse for ***1NN*** to work well

Curse of Dimensionality: Number of Samples

- Things go from bad to worse if we decide to use 3 features:



- If ***9*** was dense enough in 1D, in 3D we need $9^3 = 729$ samples!

Curse of Dimensionality: Number of Samples

- In general, if ***n*** samples is dense enough in ***1D***
- Then in ***d*** dimensions we need n^d samples!
- And n^d grows really really fast as a function of ***d***
- Common pitfall:
  - If we can't solve a problem with a few features, adding more features seems like a good idea
  - However the number of samples usually stays the same
  - The method with more features is likely to perform worse instead of expected better

Fit fall

- Data sparsity is one of the facets of the curse of dimensionality.
- Training a model with sparse data could lead to high-variance or overfitting condition.
- This is because while training the model, the model has learnt from the frequently occurring combinations of the attributes and can predict the outcome accurately.
- In real-time when less frequently occurring combinations are fed to the model, it may not predict the outcome accurately.

Bias and Variance

Objective

Bias

Variance

Overfitting

Underfitting

Best Fitting

Mean Squared Error (MSE)

Mean Square error:

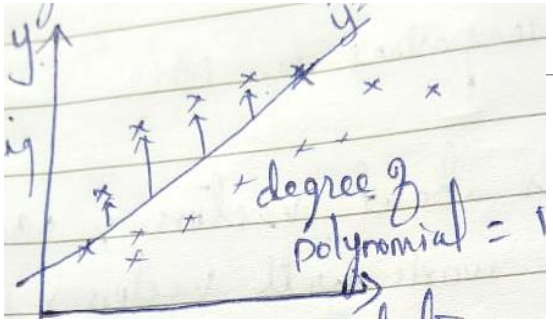
$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y_i})^2$$

Bias: Error of Training Data Set

Variance : Error of Test Data Set

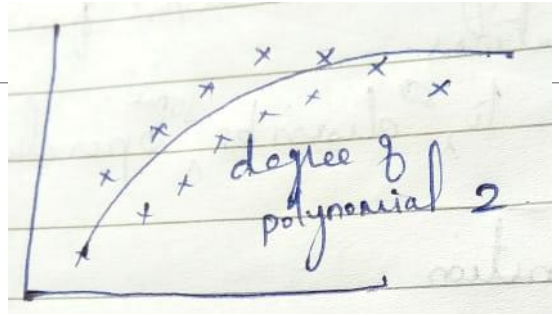
Polynomial Regression with different degree is used for understanding

Under fitting



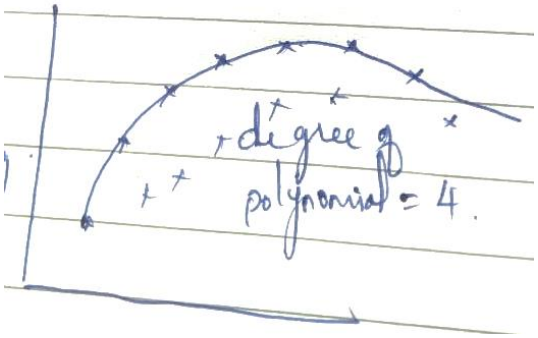
Equation: y = b0+ b1X
Training Error: High
Test Error : High

Best fitting



- Equation: y = b0+ b1X +b2X2
- Training Error: Less
- Test Error : Less

Over fitting



- Equation: y = b0+ b1X +b2X2 +b3X3+b4X4
- Training Error: Less
- Test Error : High

Exercise: Classification problem

Model 1:

- Training Error : 1%
- Test Error : 20%

Model 1:

- Training Error : 25%
- Test Error : 26%

Model 1:

- Training Error : 10%
- Test Error : 11%

Bias –variance

Bias is the algorithm's tendency to consistently learn the wrong thing by not taking into account all the information in the data (**underfitting**).

Variance is the algorithm's tendency to learn random things irrespective of the real signal by fitting highly flexible models that follow the error/noise in the data too closely (**overfitting**).

Learning Curve

Graph that compares the performance of a model on training and testing data over a varying number of training instances.

Performance improve as the number of training points increases.

When we separate training and testing sets and graph them individually

- We can get an idea of how well the model can generalize to new data

Learning curve allows us to verify when a model has learning as much as it can about the data

Learning Curve

When it occurs

- The performances on the training and testing sets reach a plateau
- There is a consistent gap between the two error rates

The key is to find the sweet spot that minimizes bias and variance by finding the right level of model complexity

Of course with more data any model can improve, and different models may be optimal.

Types of learning curves

Bad Learning Curve: High Bias

- When training and testing errors converge and are high
  - No matter how much data we feed the model, the model cannot represent the underlying relationship and has high systematic errors
  - Poor fit
  - Poor generalization

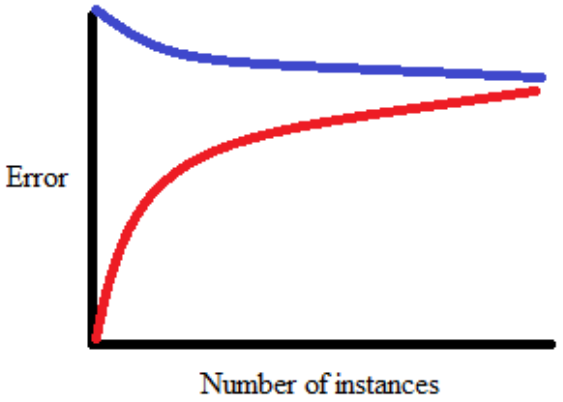


Image #1 (high bias)

Bad Learning Curve: High Variance

- When there is a large gap between the errors
  - Require data to improve
  - Can simplify the model with fewer or less complex features

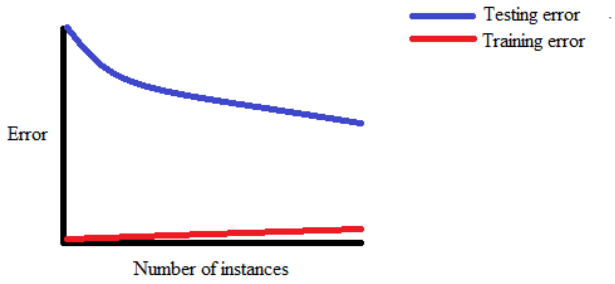


Image #3 (high variance)

Ideal Learning Curve

- Model that generalizes to new data
- Testing and training learning curves converge at similar values
- Smaller the gap, the better our model generalizes

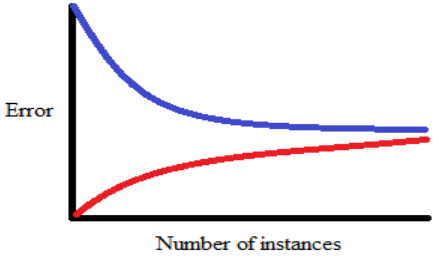
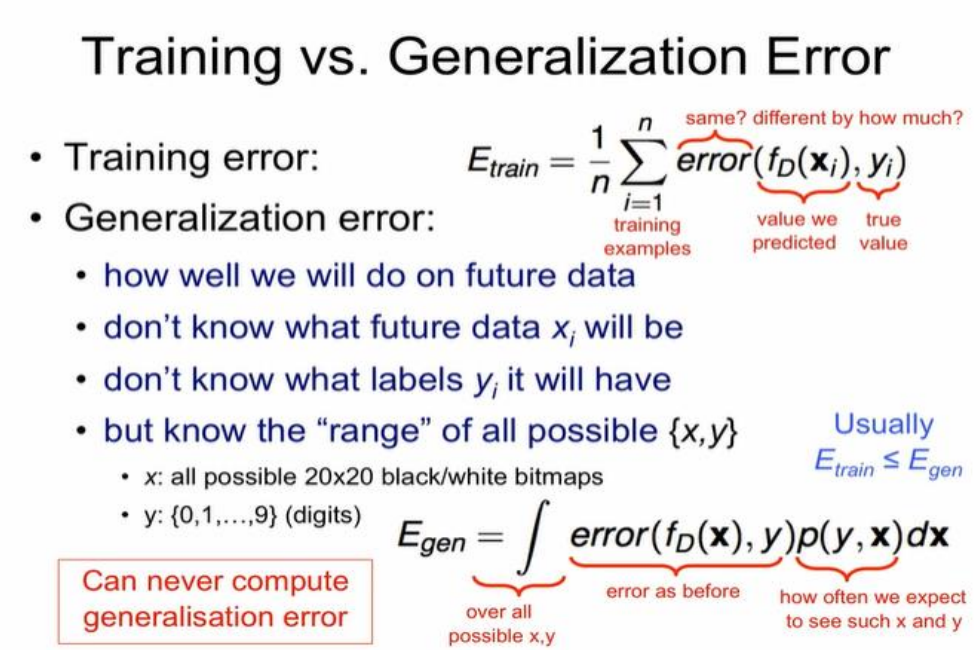


Image #2 (ideal)

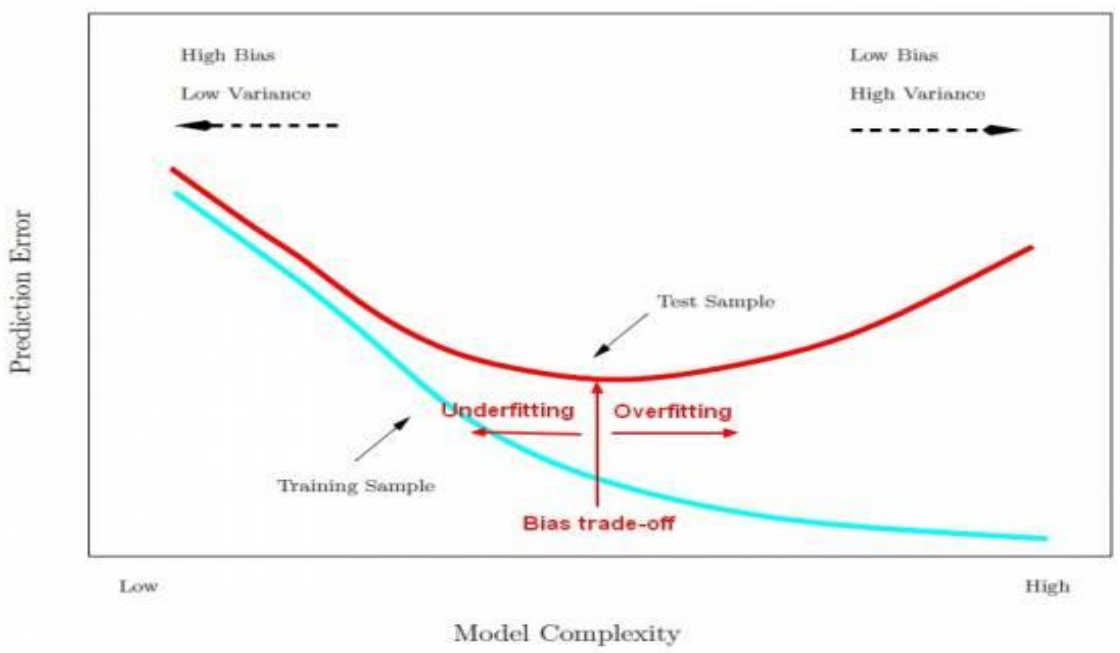
	Low *Training* Error	High *Training* Error
Low *Testing* Error	The model is learning!	Probably some error in your code. Or you've created a *psychic* AI.
High *Testing* Error	OVERFITTING	The model is not learning.

Generalization Error and Noise

Generalization error is a measure of how accurately an algorithm is able to predict outcome values for previously unseen data



Bias -Variance Tradeoff



Linear regression

Linear Regression

Linear Regression is a supervised machine learning algorithm.

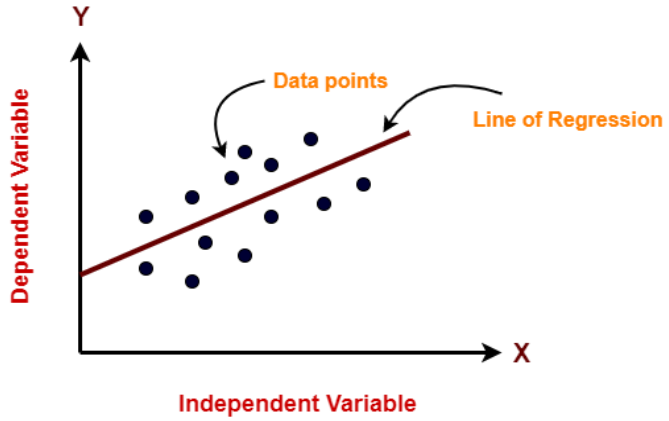
It tries to find out the best linear relationship that describes the data you have.

It assumes that there exists a linear relationship between a dependent variable and independent variable(s).

The value of the dependent variable of a linear regression model is a continuous value i.e. real numbers.

Representing Linear Regression Model-

Linear regression model represents the linear relationship between a dependent variable and independent variable(s) via a sloped straight line.

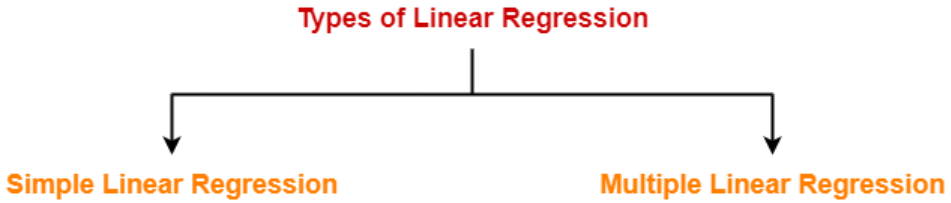


The sloped straight line representing the linear relationship that fits the given data best is called as a regression line.

It is also called as best fit line.

Types of Linear Regression

Based on the number of independent variables, there are two types of linear regression



Simple Linear Regression

In simple linear regression, the dependent variable depends only on a single independent variable.

For simple linear regression, the form of the model is-

$$Y = \beta 0 + \beta 1X$$

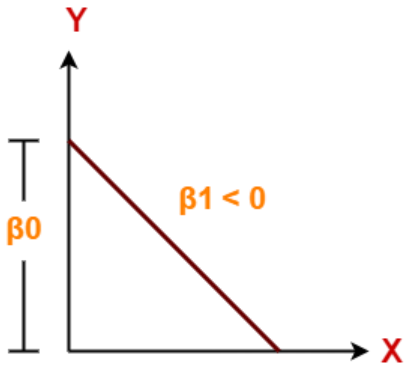
Here,

- Y is a dependent variable.
- X is an independent variable.
- β0 and β1 are the regression coefficients.
- β0 is the intercept or the bias that fixes the offset to a line.
- β1 is the slope or weight that specifies the factor by which X has an impact on Y.

Case-01: β1 < 0

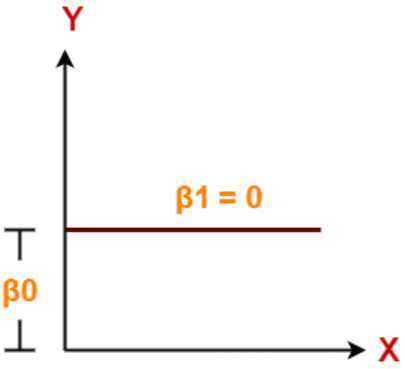
It indicates that variable X has negative impact on Y.

If X increases, Y will decrease and vice-versa.



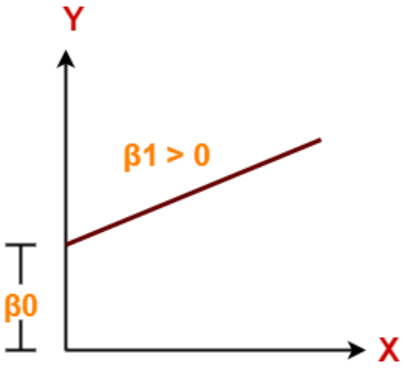
Case-02: β1 = 0

- It indicates that variable X has no impact on Y.
- If X changes, there will be no change in Y.



Case-03: β1 > 0

- It indicates that variable X has positive impact on Y.
- If X increases, Y will increase and vice-versa.



Multiple Linear Regression

In multiple linear regression, the dependent variable depends on more than one independent variables.

For multiple linear regression, the form of the model is

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \beta 3X3 + \ldots\ldots + \beta nXn$$

Here,

- Y is a dependent variable.
- X1, X2, …., Xn are independent variables.
- β0, β1,…, βn are the regression coefficients.
- βj (1<=j<=n) is the slope or weight that specifies the factor by which Xj has an impact on Y

Cross Validation

Cross Validation

The purpose of cross validation is to assess how your prediction model performs with an unknown dataset.

Method of estimating expected prediction error

Helps selecting the **best fit** model

Help ensuring model is **not over fit**

Types of Cross Validation

Exhaustive Cross Validation

- testing the machine on all possible ways by dividing the original sample into training and validation sets

Non Exhaustive Cross Validation

- do not work on all possible permutations and combinations.

Exhaustive Cross Validation

1. Leave-p-out Cross Validation (LpO CV)

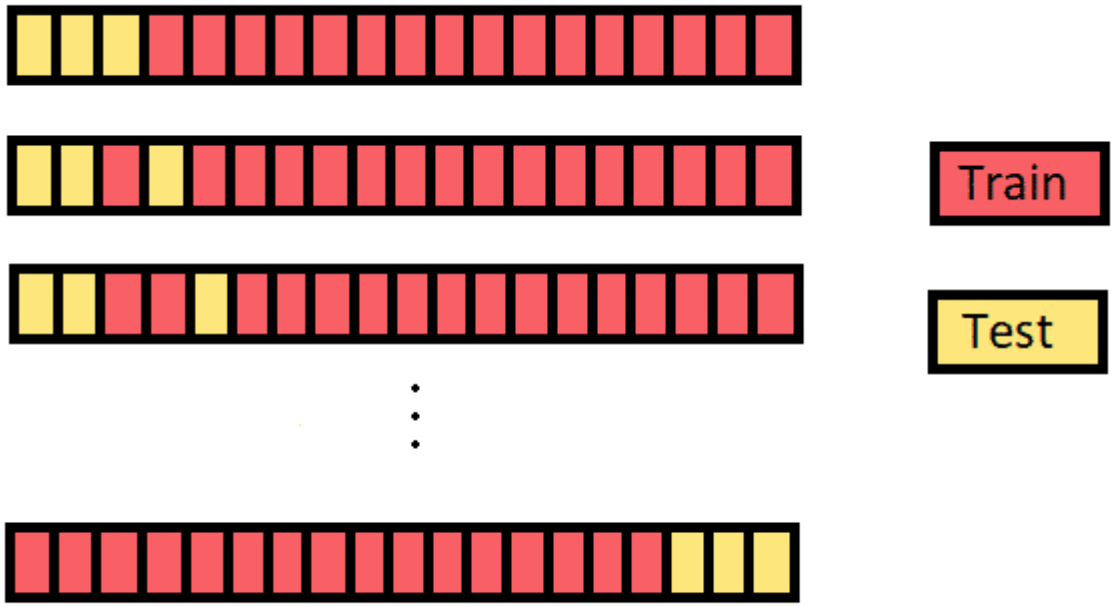
Treat the ‘p’ observations as validating set and the remaining as training sets

repeat this process for all the possible combinations of p from the original dataset

average the accuracies from all these iterations

Exhaustive Cross Validation

1. Leave-p-out Cross Validation (LpO CV)

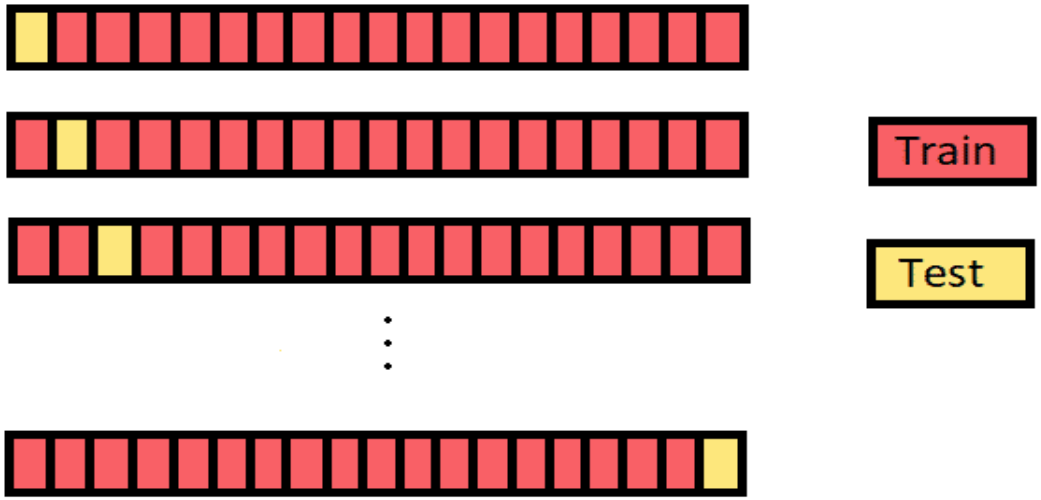


2. Leave-one-out Cross Validation (LOOCV)

similar to the LpO CV except for the fact that ‘p’ = 1.

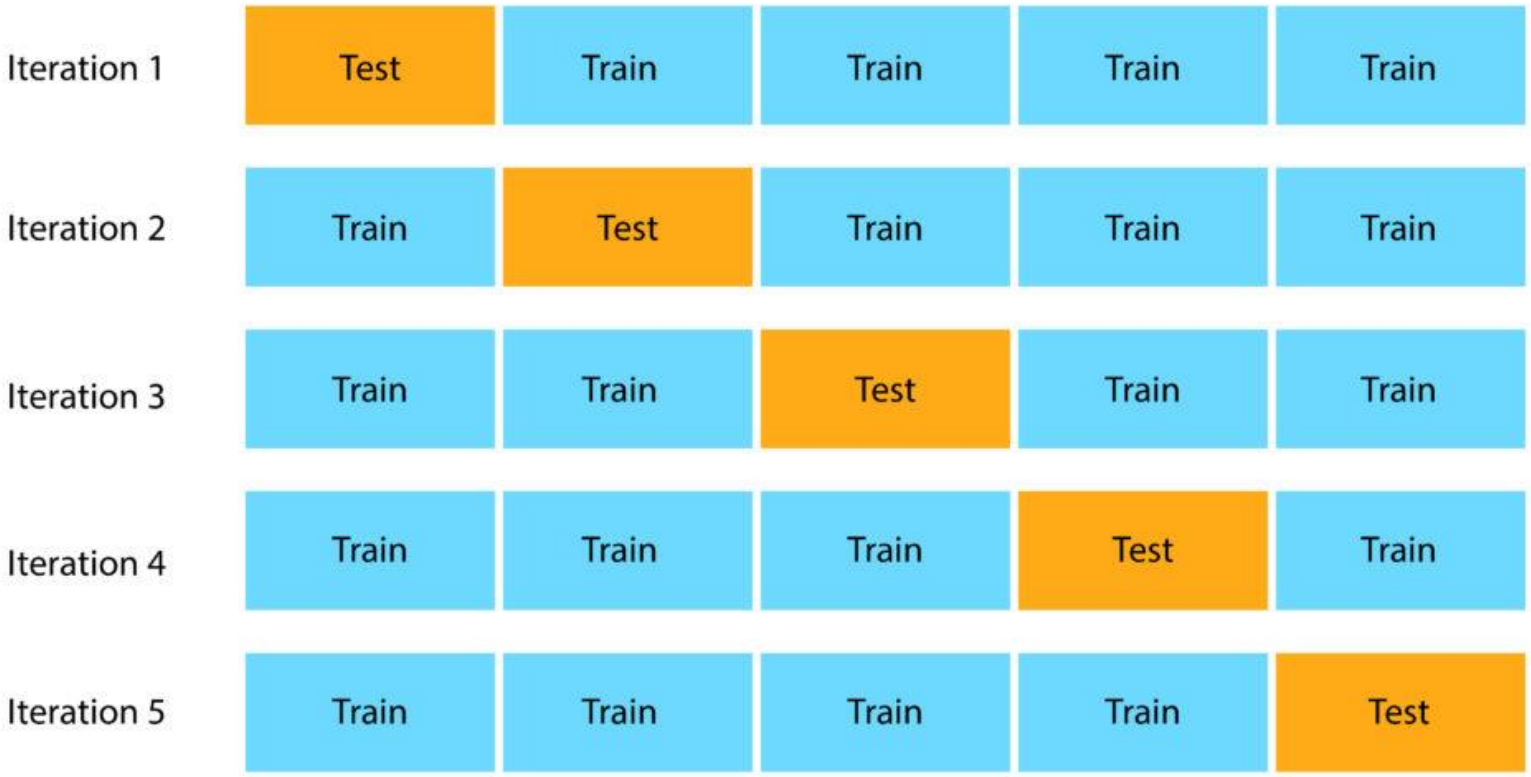
repeat this process for all the possible combinations of p from the original dataset

average the accuracies from all these iterations



Non Exhaustive Cross Validation

1. K-fold Cross Validation



Randomly split entire dataset into k number of folds (subsets)

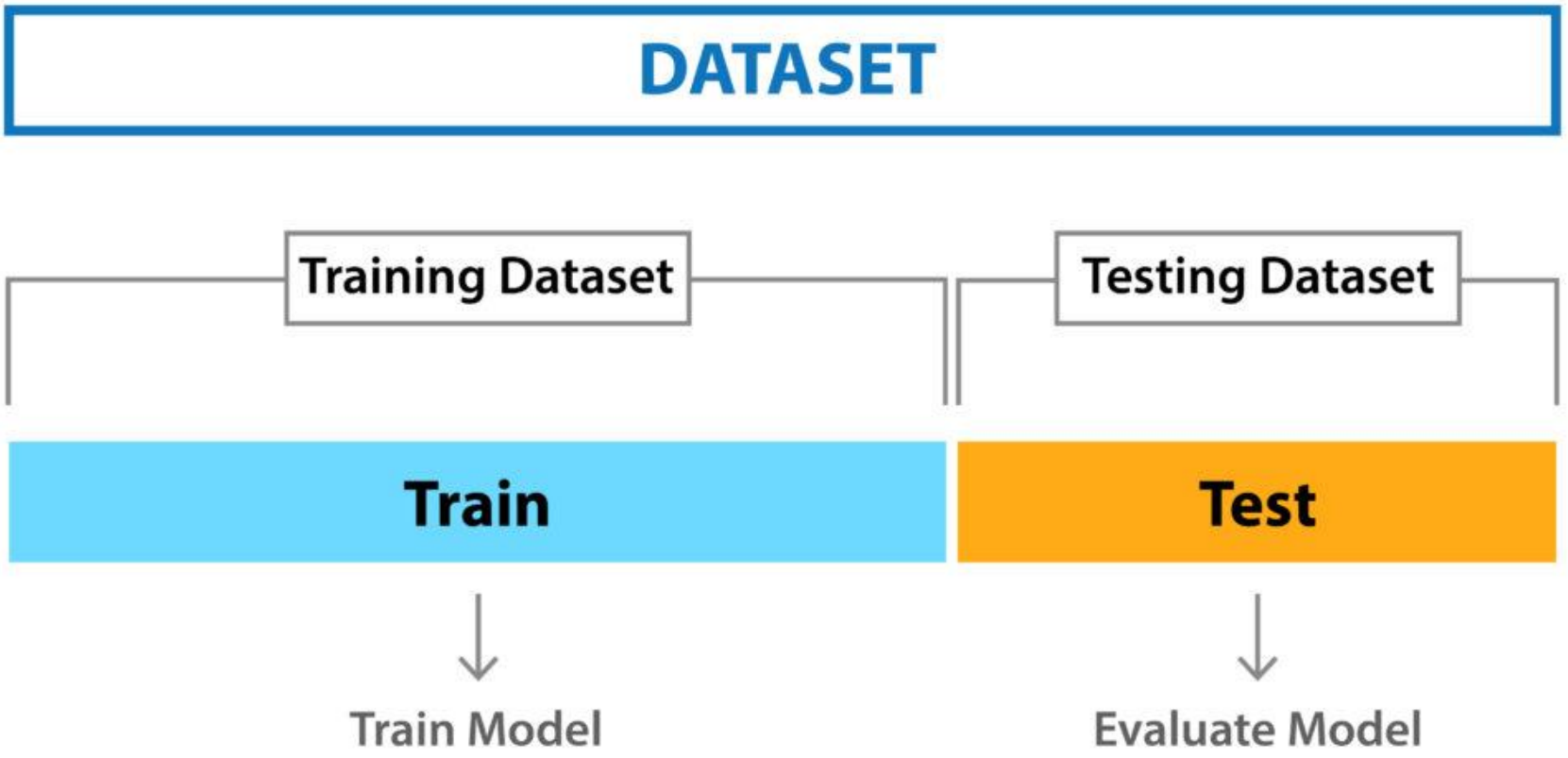
For each fold of dataset, build model on k – 1 folds of the dataset. Then, test the model to check the effectiveness for kth fold

Repeat this until each of the k-folds has served as the test set

The average of the k recorded accuracy is called the cross-validation accuracy and will serve as performance metric for the model.

Non Exhaustive Cross Validation

2. Holdout Method



Non Exhaustive Cross Validation

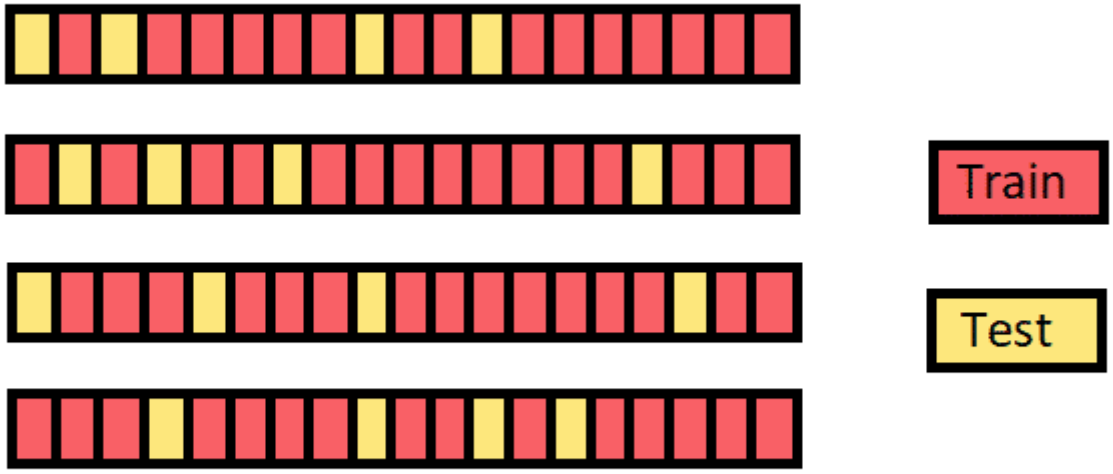
3. Monte Carlo Cross Validation

Monte Carlo validation splits the data randomly into train and test set, and this process is repeated multiple times. The results are averaged over all splits.

The disadvantage of this method is that some observations may never be chosen, whereas some might be selected multiple times.

Non Exhaustive Cross Validation

3. Monte Carlo Cross Validation



Regularization

What is Overfitting

The training data contains information about the regularities in the mapping from input to output. But it also contains **sampling error**.

There will be accidental regularities because of the particular training cases that were choosen.

When we fit the model, It cannot tell which regularities are real and which are caused by sampling error.

So it fits both kinds of regularity. If the model is very flexible it can model the sampling error really well.

This means the model will not generalize well to unseen data

regularize means to make things regular or acceptable

Regularization refers to a set of different techniques that lower the complexity of a neural network model during training, and thus prevent the overfitting

Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.”

Overfitting revisited: regularization

A regularizer is an additional criteria to the loss function to make sure that we don't overfit

It's called a regularizer since it tries to keep the parameters more normal/regular

It is a bias on the model forces the learning to prefer certain types of weights over others

$$\text{argmin}_{w,b} \sum_{i=1}^{n} loss(yy') + \lambda \; regularizer(w,b)$$

Regularization is implemented by adding a “penalty” term to the best fit derived from the trained data, to achieve a lesser variance with the tested data and also restricts the influence of predictor variables over the output variable by compressing their coefficients.

Regularization Techniques

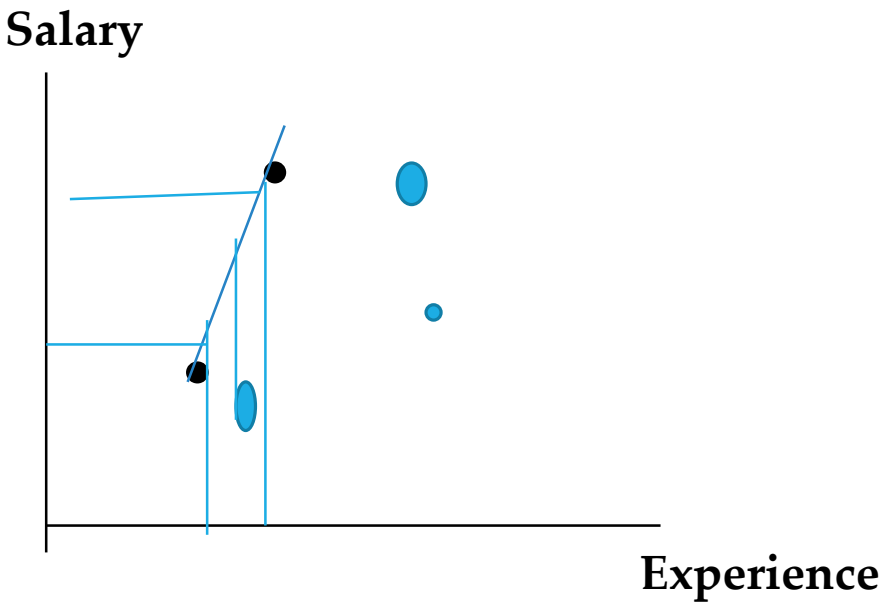
L2 Regualrizartion / Ridge Regularization

L1 Regualrizartion / Lasso Regularization

Dropout

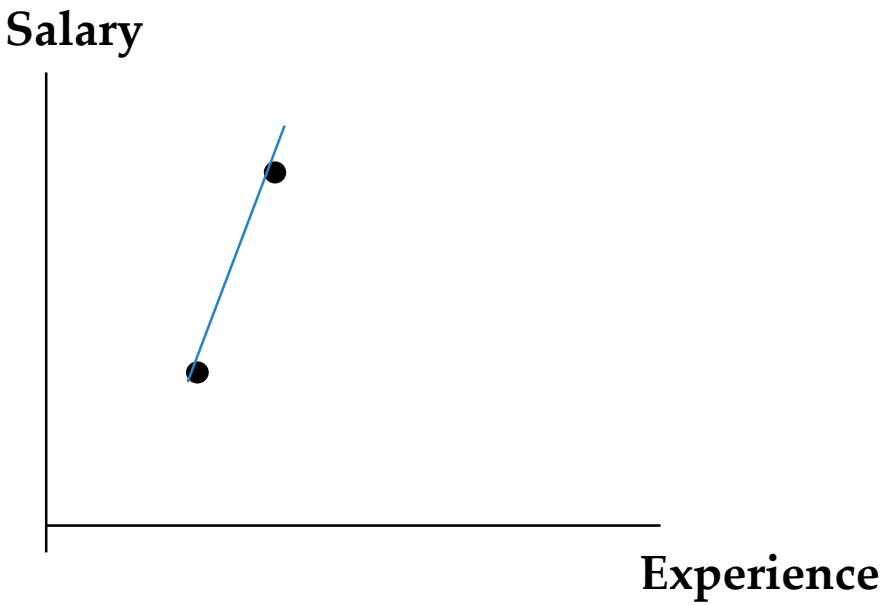
Early Stopping

Ridge Regularization



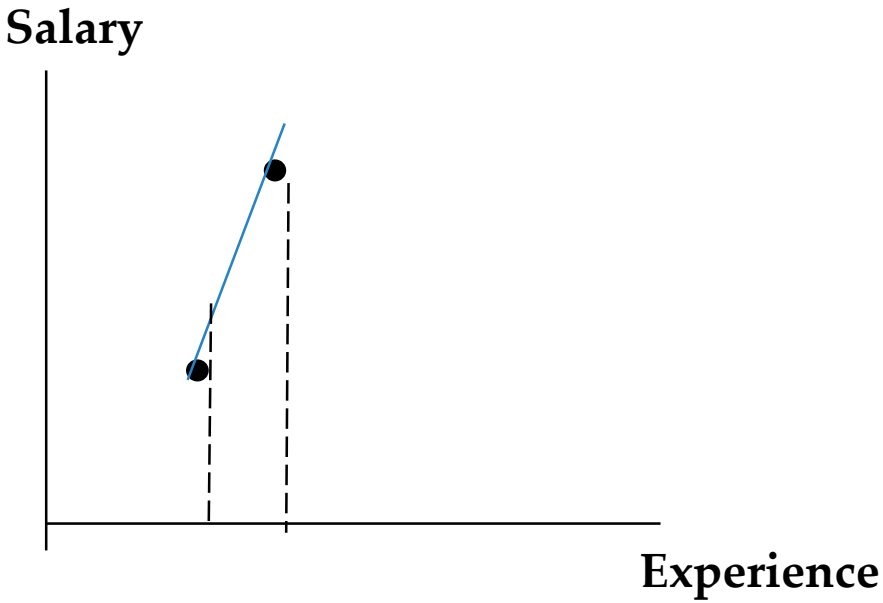
- Sum of Residues $=\sum_{i=1}^{n}(y-\overline{y})2$

Ridge Regularization



- Sum of Residues $=\sum_{i=1}^{n}(y-\overline{y})2$ + (Lamda)*(slope)2
- Where lamda is the learning rate.

Steep slope



- Sum of Residues $=\sum_{i=1}^{n}(y-\overline{y})2$ + (Lamda)*(slope)2
- Where lamda is the learning rate.

Assume lamda =1

Slope = 1.3

Then cost = 0 + 1(1.3)2

= 1.69

Assume lamda =1

Slope = 1.1

Then cost = 0 + 1(1.1)2

= 1.21

- For multiple features
- Sum of Residues $=\sum_{i=1}^{n}(y-y)2$ + (Lamda)*((m1)2+(m2)2+(m3)2)

- Lamda will take value greater than 0 and any +ve value

Lasso Regression - Least Absolute Shrinkage and Selection Operator

This help in feature selection too

Lamda * | m1+m2+m3…|

- Objective = RSS + α * (sum of absolute value of coefficients)

Instead of taking the square of the coefficients, magnitudes are taken into account.

This regularization (L1) lead to zero coefficients

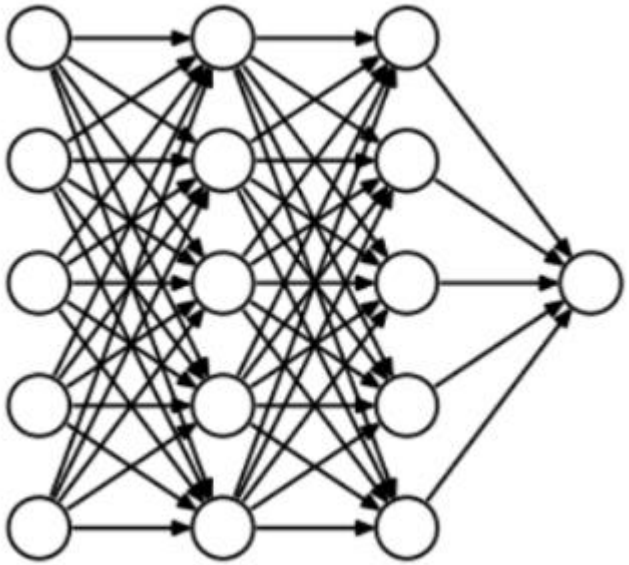
- i.e. some of the features are completely neglected for the evaluation of output.
- **So Lasso regression not only helps in reducing over-fitting but it can help us in feature selection.**

Dropout

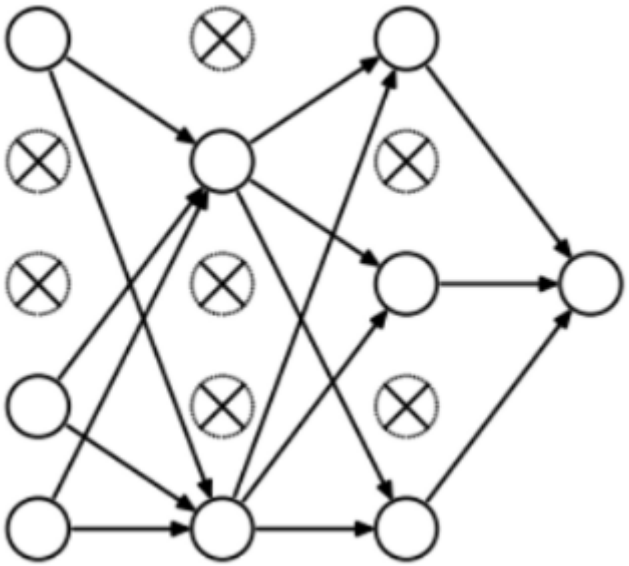
This is the one of the most interesting types of regularization techniques.

It also produces very good results and is consequently the most frequently used regularization technique in the field of deep learning

To understand dropout, let's say our neural network structure



At every iteration, it randomly selects some nodes and removes them along with all of their incoming and outgoing connections as shown below



So each iteration has a different set of nodes and this results in a different set of outputs. **It can also be thought of as an ensemble technique in machine learning.**

Ensemble models usually perform better than a single model as they capture more randomness. Similarly, dropout also performs better than a normal neural network model

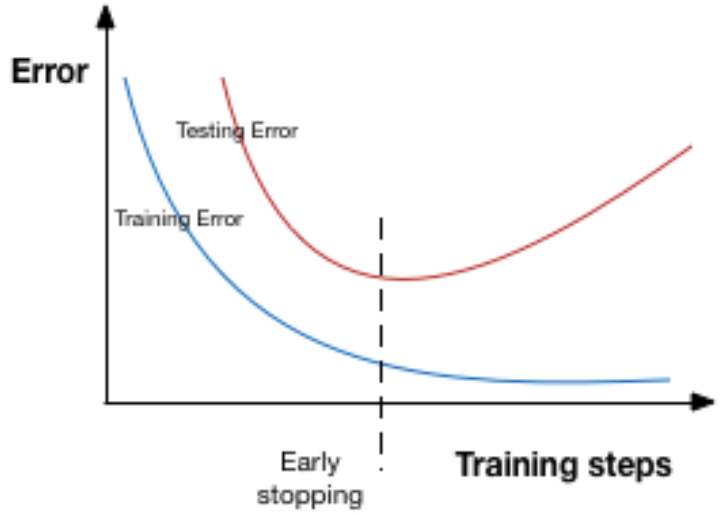
Due to these reasons, dropout is usually preferred when we have a large neural network structure in order to introduce more randomness.

This probability of choosing how many nodes should be dropped is the hyperparameter of the dropout function.

Early stopping

Early stopping is a kind of cross-validation strategy where we keep one part of the training set as the validation set.

When we see that the performance on the validation set is getting worse, we immediately stop the training on the model. This is known as early stopping



In the above image, we will stop training at the dotted line since after that our model will start overfitting on the training data

Classification

Classification

- A machine learning task that deals with identifying the class to which an instance belongs is called classification.
- **Classifier:** A classifier performs classification
- Classification is a supervised learning method. Supervised learning maps labelled data to known output.
- Classification can be applied to dataset whose features are labelled.

Classification: Labelled data

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

labels

A sample labelled dataset with two labels (classes) : benign and malignant

Classification: Labelled data

Three Type of Classification Tasks

YAHOO! JAPAN

Binary Classification

- Spam
- Not spam

Multiclass Classification

- Dog
- Cat
- Horse
- Fish
- Bird
- ...

Multi-label Classification

- Dog
- Cat
- Horse
- Fish
- Bird
- ...

Classification: Unlabeled data

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5
10	47	3	23	115	0.653	3.947	NBA011	4
11	44	3	8	88	0.285	5.083	NBA010	6.1
12	34	2	9	40	0.374	0.266	NBA003	1.6

unlabeled

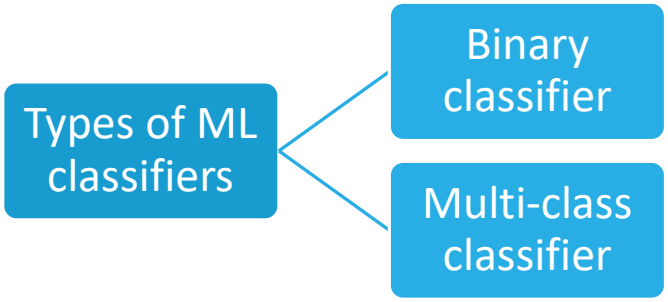
A sample labelled dataset no labels but with continuous values (DebtIncomeRatio)

Classification

Formula:

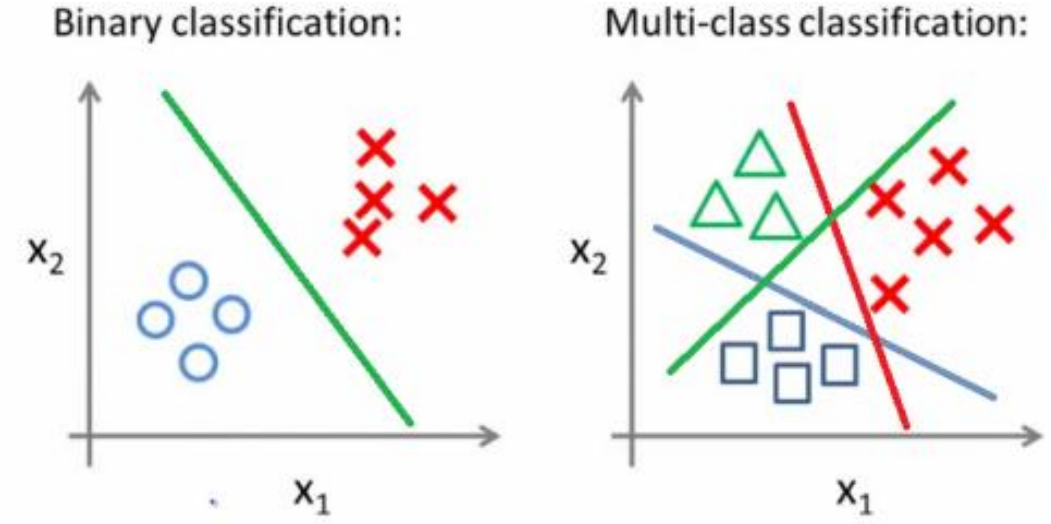
- In classification algorithm, a discrete output function(y) is mapped to input variable(x).
- y=f(x), where y = categorical output
- The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

Types of Classifiers



Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary Classifier. Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

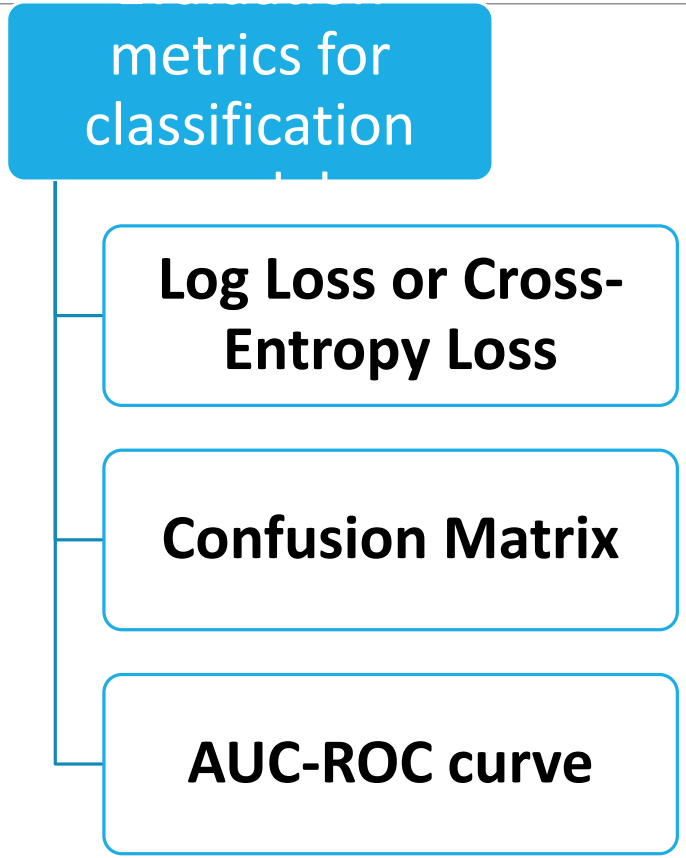
Multi-class Classifier: If a classification problem has more than two outcomes, then it is called as Multi-class Classifier. Examples: Classifications of types of crops, Classification of types of music.



Types of ML classification algorithms

Types of ML classification algorithms	**Linear Models**	•Logistic Regression •Support Vector Machines
	•Non-linear Models	•K-Nearest Neighbours •Kernel SVM •Naïve Bayes •Decision Tree Classification •Random Forest Classification

Evaluating a classification model



Uses of classification algorithm

- Email Spam Detection
- Speech Recognition
- Identifications of Cancer tumor cells.
- Drugs Classification
- Biometric Identification, etc.

Parametric vs Non-parametric models

Parametric vs Non-parametric models

- Parametric model: The model which has fixed number of parameters is a Parametric model.

- Non - Parametric model: The model whose parameter grows with the amount of training data is non-parametric model.

Benefits of Parametric model

- Simpler: These methods are easier to understand and interpret results.
- Faster: Parametric models are very fast to learn from data.
- Less training Data: They do not require as much training data and can work well even if the fit to the data is not perfect.

Limitations of Parametric model

- Highly Constrained: By choosing a functional form these methods are highly constrained to the specified form.
- Limited Complexity: The methods are more suited to simpler problems.
- Poor Fit: In practice the methods are unlikely to match the underlying mapping function.

Examples of Parametric model

- Linear Regression
- Linear Support Vector Machines
- Logistic Regression
- Naive Bayes
- Perceptron

Benefits of Non-Parametric model

- Flexibility: Capable of fitting a large number of functional forms.
- Power: No assumptions (or weak assumptions) about the underlying function.
- Performance: Can result in higher performance models for prediction.

Limitations of Non-Parametric model

- More data: Require a lot more training data to estimate the mapping function.
- Slower: A lot slower to train as they often have far more parameters to train.
- Overfitting: More of a risk to overfit the training data and it is harder to

explain why specific predictions are made.

Examples of Non-Parametric model

- Decision Trees
- K-Nearest Neighbor
- Support Vector Machines with Gaussian Kernels
- Artificial Neural Networks

Parametric vs Non-parametric models

	Parametric	Non-parametric
Assumed distribution	Normal	Any
Assumed variance	Homogeneous	Homogenous and Heterogeneous
Typical data	Ratio or Interval	Ordinal or Nominal
Data set relationships	Independent	Any
Usual central measure	Mean	Median
Benefits	Can draw more conclusions	Simplicity; Less affected by outliers

Linear Algebra for Machine Learning

Data Representation

How can we represent data in way that computer can understand

(images, text, user preferences etc)

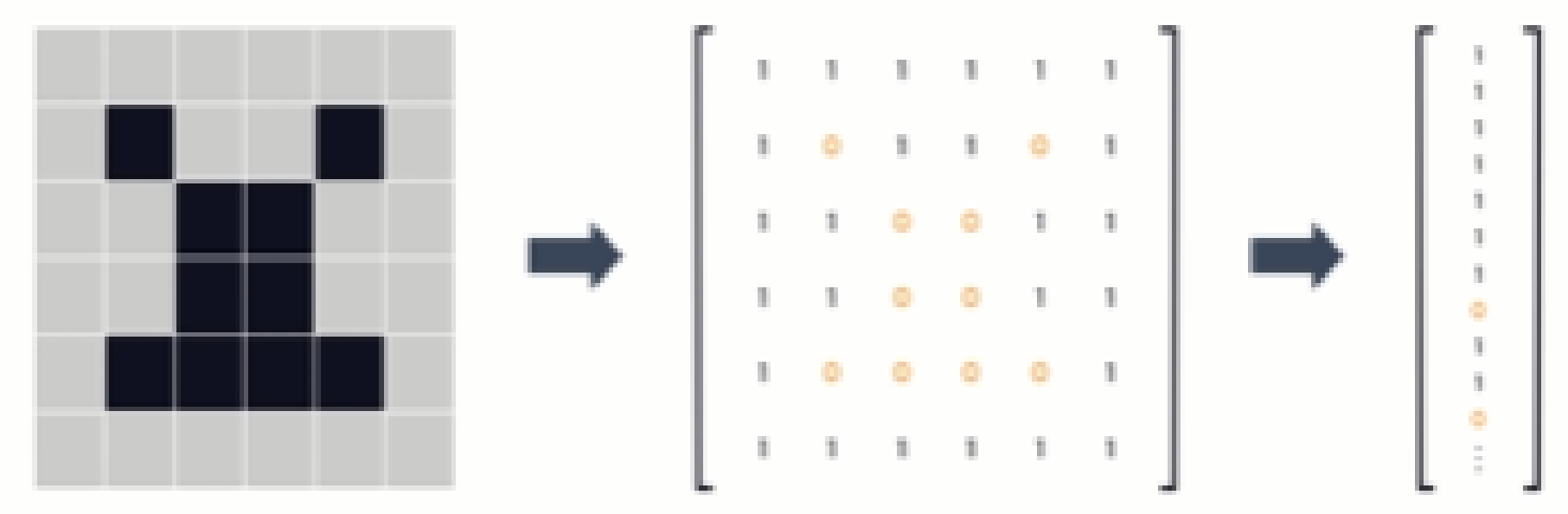
Organize information into vector . A Vector is a 1-dimensional array of numbers It has both a magnitude and a direction

A feature vector is a vector whose entries represent the “features” of some object.

A vector space containing them is called feature space

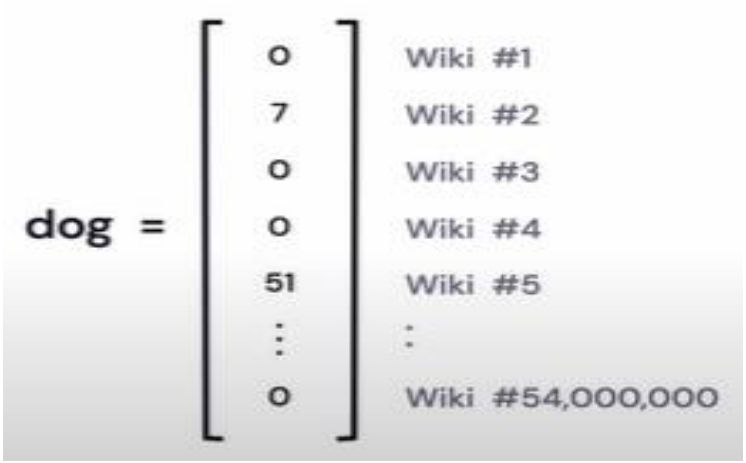
Examples of Data Representations

- **Image**



Documents / Words

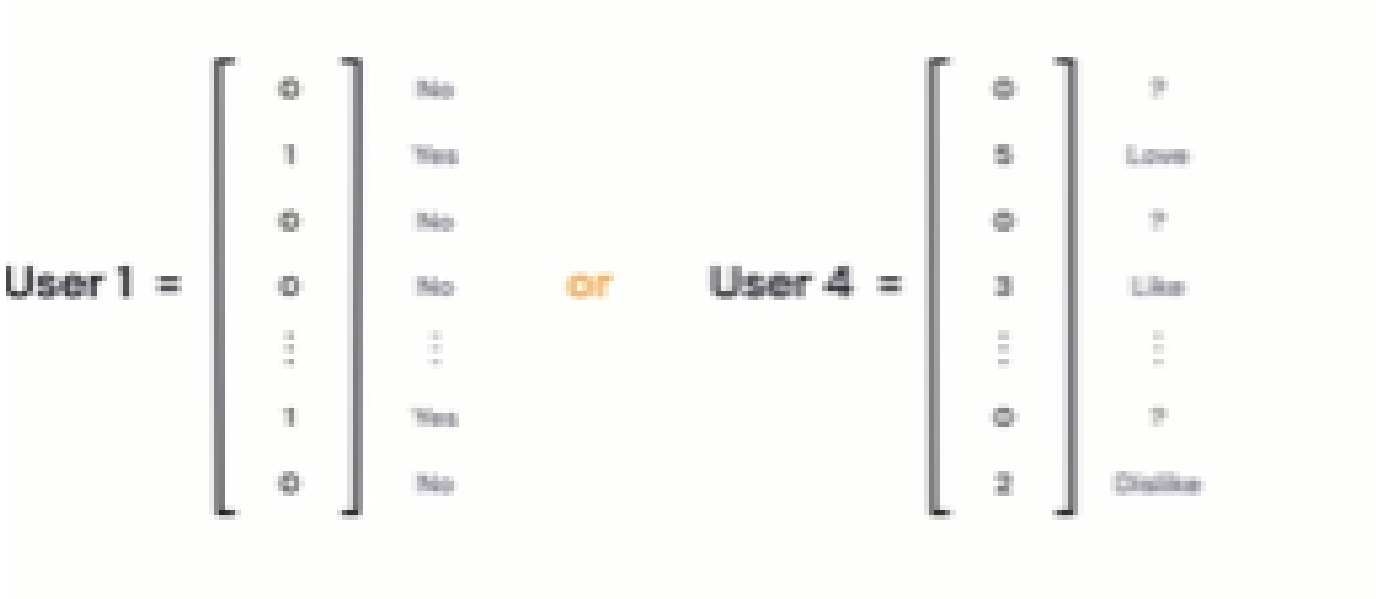
Given a collection of documents assign to every word a vector whose ith entry is the number of times the word appears in the ith document.



These vectors can assemble into large matrix , useful for latent **semantic analysis**

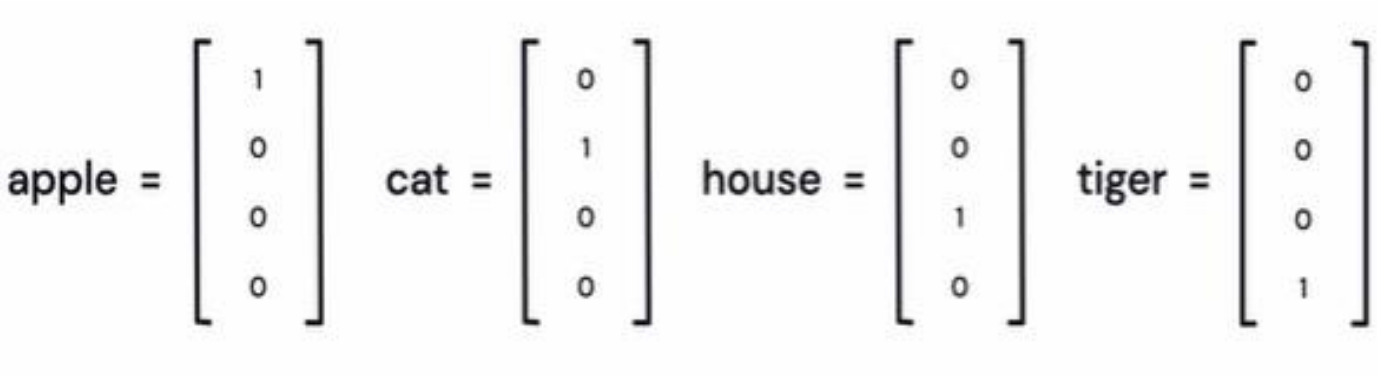
Yes / No or Ratings

Given users and items (e.g movies), vectors can indicate if a user has interacted with the item(1=yes, 0=no) or the users ratings, say a number between 0 and 5.



One Hot Encoding

Assign to each word a vector with one 1 and 0s elsewhere.

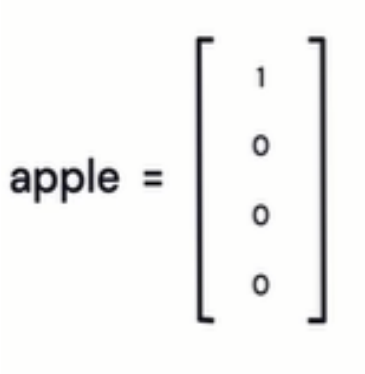


Linear Algebra

Vectors

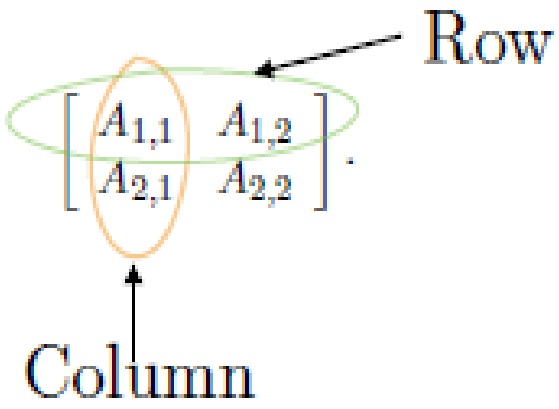
A vector is a 1-D array of numbers:
It can be real, binary, integer etc.
Example notation for type and size:

. $\mathbb{R}^n$



Matrices

A matrix is a 2-D array of numbers:



Example notation for type and shape

$$\boldsymbol{A} \in \mathbb{R}^{m \times n}$$

Tensors

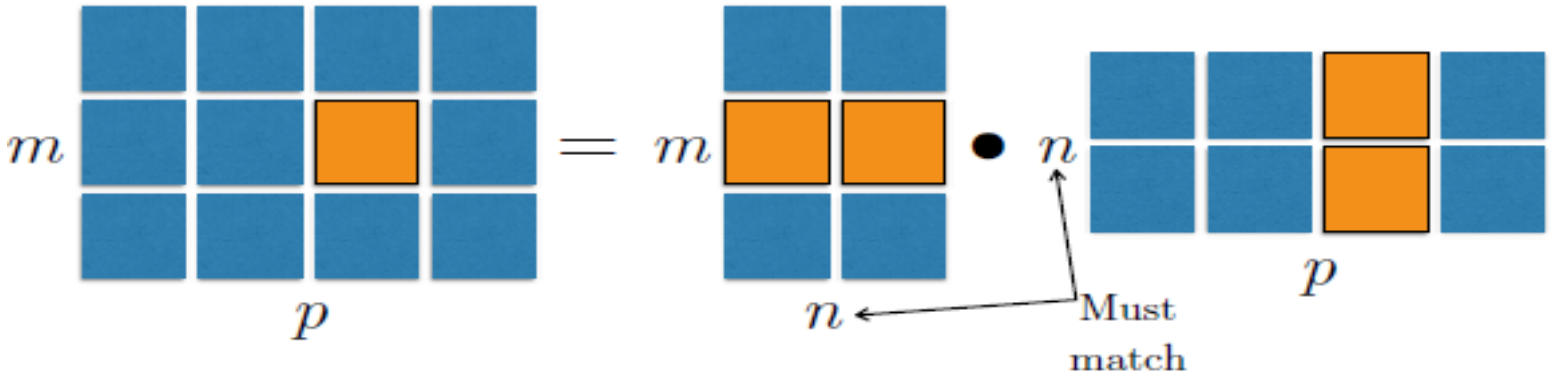
A tensor is an array of numbers, that may have

- zero dimensions, and be a scalar
- one dimension, and be a vector
- two dimensions, and be a matrix
- or more dimensions.

Matrix (Dot) Product

C =AB

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$



Identity Matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{I}_n \boldsymbol{x} = \boldsymbol{x}$$

Matrix operations

- Matrix Addition
- Scalar Multiplication
- Matrix Multiplication
- Transpose

Eigen Vector and Eigen Value

- An **eigenvector** or **characteristic vector** of a linea ansformation is a nonzero vector that changes by a scalar factor wher at linear transformation is applied to it.
- The corresponding **eigenvalue**, often denoted by

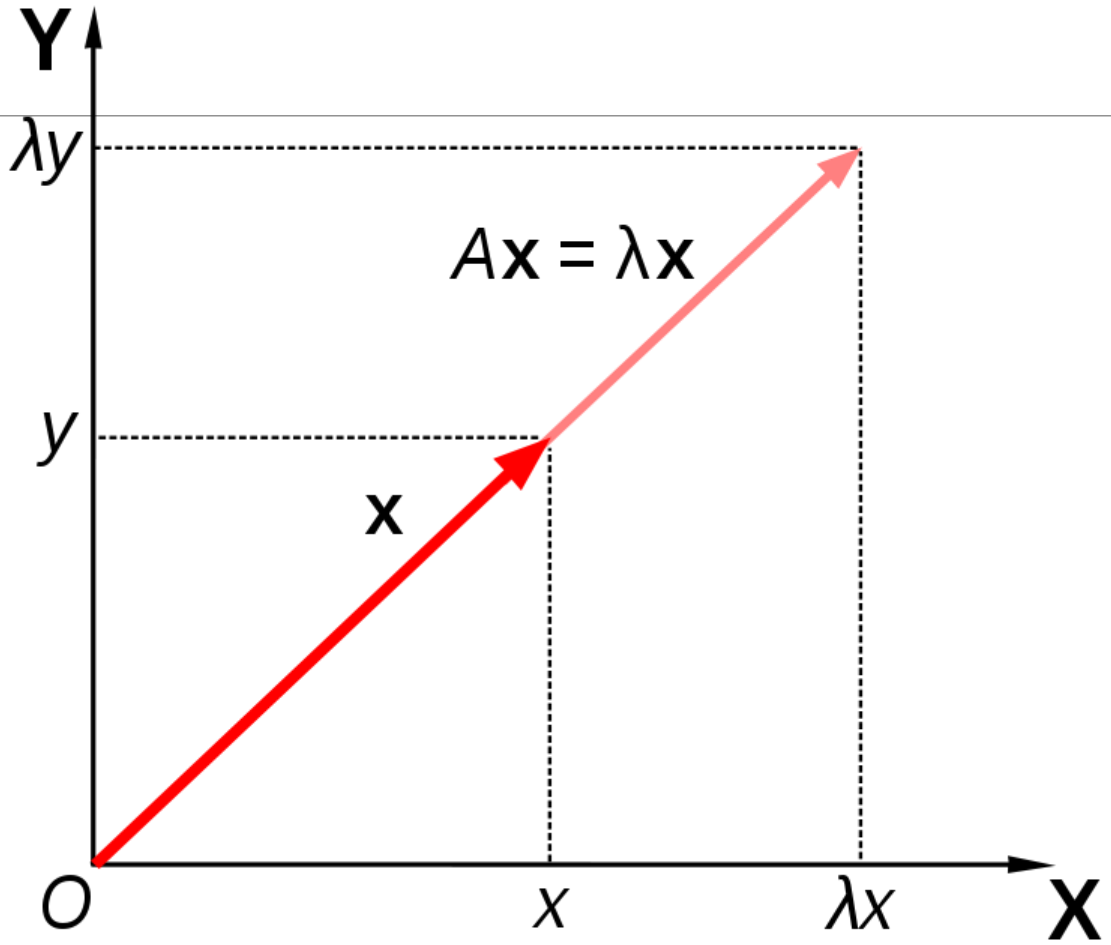
  is the factor by which the eigenvector is scaled.
- Geometrically, an eigenvector, corresponding to a real nonzero eigenvalue, points in a direction in which it is stretched by the transformation and the eigenvalue is the factor by which it is stretched..

$$x = \begin{bmatrix} 1 \\ -3 \\ 4 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} -20 \\ 60 \\ -80 \end{bmatrix}.$$

These vectors are said to be scalar multiples of each other, or parallel or collinear, if there is a scalar λ such that

In this case .

$$\lambda = -1/20.$$


Y
λy
Ax = λx
y
x
O
x
λx
X

Eigen Value and Eigen Vector

Eigenvalues and Eigenvectors are used in one of the most popular dimensionality reduction technique – Principal Component Analysis (PCA).

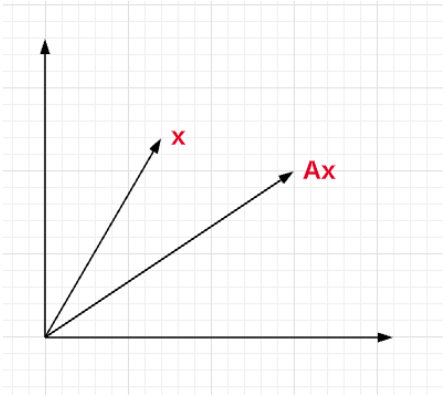
In PCA, these concepts help in reducing the dimensionality of the data resulting in the simpler model which is computationally efficient and provides greater generalization accuracy.

Eigenvectors are the vectors which when multiplied by a results in another vector having same direction but scaled in forward or reverse direction by a magnitude of the scaler multiple which can be termed as Eigenvalue.

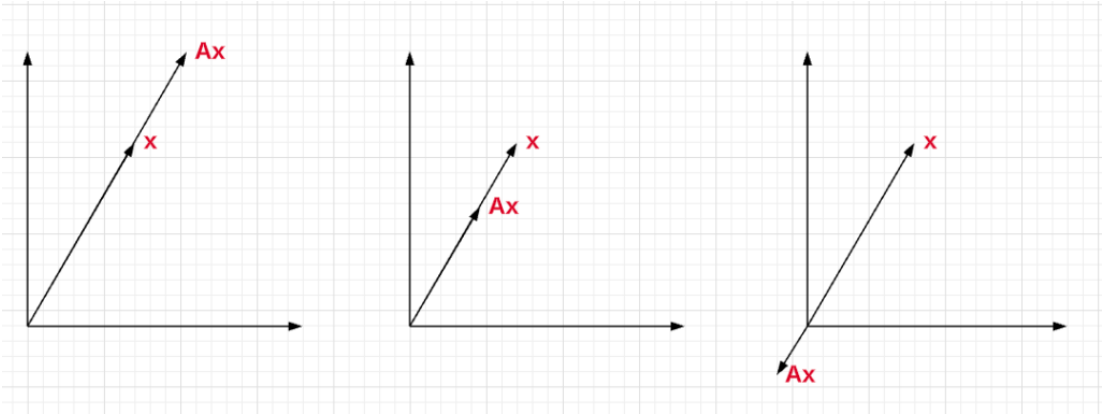
In simpler words, eigenvalue can be seen as the scaling factor for eigenvectors.

Here is the formula for what is called eigen equation.

$$\mathbf{Ax=\lambda x}$$



Non Eigen Vector



Eigen Vector

Thank You