INTRODUCTION TO MACHINE LEARNING

MACHINE LEARNING:

Machine learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the past data experience on their own. The term machine learning was first introduced by Arthur Samuel in 1959.

DEFINITION:

Machine learning enables a machine to automatically learn from data, improve performance from experiences and predict things without being explicity programmed.

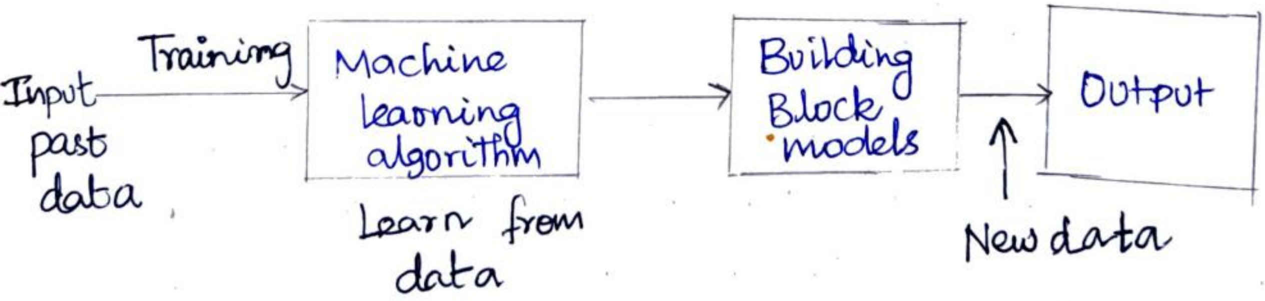


fig: Working of Machine Learning

A machine System learns from historical data, builds the prediction models and whenever it receives new data and predicts the output for it.

Features of Machine learning :

⇒ Machine learning uses data to detect various patterons in a given set.

⇒ It can learn from past data and improve automatically

⇒ It is a data driven technology

⇒ Machine learning is much similar to data mining as it also deals with huge amount of the data.

Classification Of Machine learning :

It is classified into three types they are

* Supervised learning
* Unsupervised learning
* Reinforcement learning

Supervised Learning :

Supervised learning is a type of Machine learning method in which we provide Sample labelled data to the machine learning system in order to train it and on that basis it predict the output

The System creates a model using labelled data to understand the datasets and learn about each data once the training and processing are done then we test the model by providing a sample data to check Wheather it is predicting the exact output or not.

The Supervised data is based on Supervision and it is the same as When the Student learns things In the Supervision of the teacher The example of Supervised learning is Spam filtering

Supervised learning can be grouped further into two categories of algorithms:

* Classification
* Regression

Un Supervised Learning :

Unsupervised learning is a learning method in Which a machine learns without any Supervision..

The training is provided to the machine with the set of data that has not been labeled classified or categorized and the algorithm needs to act on the data without any Supervision. The goal of unsupervised learning is to restructure the input data into few features or a group of objects with Similar patterns

In Supervised learning we dont have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classified into two categories of algorithms:

* clustering

* Association

Reinforcement Learning:

Reinforcement learning is a feedback based learning method, in which learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improve its performance. In Reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points and hence, it improves its performance

Linear Algebra:

Linear algebra is an essential field of mathematics, Which defines the study of vectors, matrices, planes, mapping and lines required for linear transformation

Linear Algebra play a vital role and key fondation in Machine learning and it enables Machine learning algorithms to run a huge number of datasets. The concepts of linear algebra are widely used in developing algorithms in machine learning. It can also perform the following task:

* optimization of data

* Applicable in loss functions, regularisation, Covariance matrices, Singular Value Decomposition (SVD) Matrix operations and Support vector machine classifications

Implementation of Linear Regression in Machine learning.

The linear algebra is also used in neural networks and data science field.

Linear Algebra for Machine learning

Notation:

Notation in linear algebra enables you to read algorithm descriptions in papers, books and Website to understand the algorithm's working. Even if you use for-loops rather than matrix operations, you will be able to piece things together.

Operations:

Working with an advanced level of abstaction in vector and matrices can make concepts clearer and it also help in the description, coding and even thinking capability. In linear algebra, it is required to learn the basic Operation such as addition, multiplication inversion, transposing of matrices, vectors, ect.

Matrix Factorization:

One of the most recommended areas of linear algebra is matrix factorization, Specifically matrix deposition methods Such as SVD and matrix decomposition Techniques.

Examples of Linear Algebra in machine learning:

Data sets and Data Files:

Each Machine learning project works on the dataset and we fit the machine learning model using this dataset.

Each dataset resembles a table-like structure consists of rows and columns. Where each row represents operation and each column represents features and variables. This dataset is handled as a matrix Which is key data structure in Linear Algebra

Further when this dataset is divided into input and output for the Supervised learning model it represents a Matrix (X) and Vector (y), Where the Vector is also an important concept of linear algebra.

Images and photographs:

In machine learning images and photographs are used for Computer vision applications Each image is an example of the matrix from linear algebra because an image is a table structure consists of height and width for each pixel.

One Hot Encoding:

In Machine learning sometimes we need to work with categorial data. These categorical variables are encoded to make them simpler and easier to work with and the popular encoding technique to encode these variables is known as one hot encoding.

In the one hot encoding technique a table is created that shows a variable with one column for each category and one row for each example in dataset. Further each row is encoded as binary vector, which contains either zero or one value. This is a example of Sparse representation. Which is a Subfield of linear algebra

Linear Regression:

Linear Regression is a popular technique of machine learning borrowed from statistics

It describes the Relationship between input and output variables that is used in machine learning to predict numerical values. The most common way to slove linear regression problems using least Square optimization is sloved with the help of Matrix factorization methods are singular value decomposition, which are the concept of linear algebra

Regularization:

This technique is used to minimize the size of co-efficients of a model while it is being fit on data is known as regularization. Common Regularization techniques are L_1 and L_2 regularization. Both of these forms of regularization are in fact a measure of the magnitude or length of the coefficients as a vector and are methods lifted directly from linear algebra called the vector norm.

principal Component Analysis:

Generally each dataset contains thousands of features and fitting the model with such a large dataset is one of the most challenging tasks of Machine learning. Moreover a model built with irrelevant features is less accurate than a model built with relevant features. The are several methods in machine learning that automatically reduce the number of columns of a dataset and these methods are

Dimensionality reduction. the most commonly used dimensionality reduction methods in machine learning is principal Component Analysis or PCA.

Singular Value Decomposition: is also one of the popular dimensionality reducing techniques and is also written as SVD in short form

It is the matrix factorization method for linear algebra and it is widely used in different application such as feature selection, Visulization, noise reduction and many more.

Latent Semantic Analysis:

Natural language processing or NLP is a Subfield of machine learning that works with text and spoken words.

NLP represents a text document as large Matrices with the occurrence of words and rows may contain sequences, paragraphs, pages ect.. with cells in the matrix marked as the count or frequency of the number of times the word Occured. It is a sparse matrix representation of text. Documents processed in this way are much eaiser to compare query and use as basis for a Supervised Machine Learning model. This form of data preparation is called Latent Semantic Analysis or LSA for short and is also known by the name latent Semantic Indexing or LSI

Recommender System:

A Recommender System is a sub field of Machine learning a predictive modelling problem that provides Recommendation of products. For example online Recommendation of book based on the customer previous purchase history, recommendation of movies and TV series, as we see in Amazon and Netfix.

An example of calculating the similarity between sparse customer behaviour vectors using distance measures such as Euclidean distance or dot products

Different matrix factorization methods such as Singular Value decomposition are used in recommender Systems to query, search and compare user data.

Deep learning:

Artificial Neural Networks or ANN are the non linear ML algorithms that work to process the brain and transfer information from one layer to another in a similar way.

Deep learning studies these neural networks, Which implement newer and faster hardware for the training and development of larger networks with a huge dataset. All deep learning methods achieve great results for different challenging tasks such as Machine translation, Speech recognition.

Examples of Machine Learning Applications :

Learning Associations :

In finding an association rule, we are interested in learning a conditional propability of the form $P(X|Y)$ Where Y is the product we would would like to condition on X, Which is the product or the set of The products which we know that the customer has already purchased.

We may want to make a distinction among customer and towards this estimate $P(Y|X, D)$ Where D is the set of customer attributes for example gender, age, matrial status and so on assuming that we have acess to this information

classification :

The classification problem is where there are two classes low risk and high risk customers. The information about the customer makes up the input to the classifier whose task is to assign the input to one of the two classes.

After training with the past data a classification rule learned may be of the form

"IF income > θ_1 AND savings > θ_2 THEN low-risk ELSE high risk"

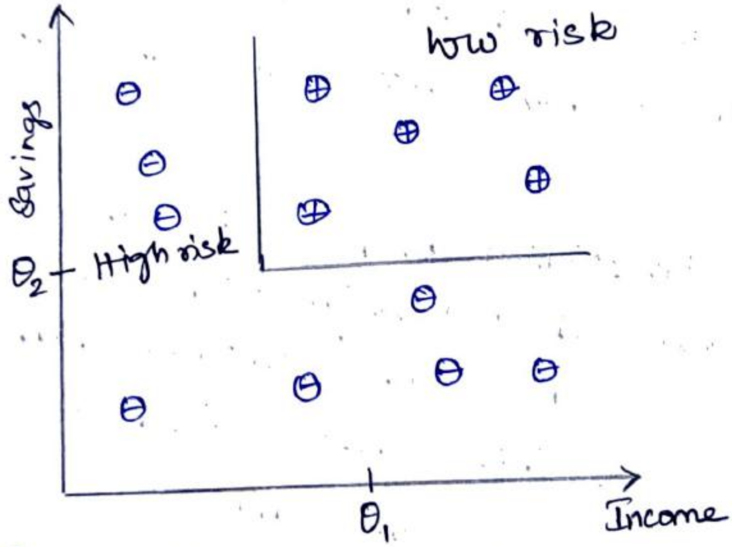
for suitable values of θ_1 and θ_2

Discriminant :

It is a function that separates the examples of different classes.

prediction :

We have a rule that fits the past data if the future is similar to the past, then we can make correct predictions for novel instances. A application with a certain income and savings we can decide wheather it is low risk or high risk.



pattern Recognition :

pattern recognition is the process of recognizing patterns by using a machine learning algorithm. pattern Recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation

eg: speech recognition Speaker indentificati
multimedia document recognition

pattern recognition :

optical character recognition :

It is used to recognize character codes from their images. It is an example for multiple classes as many as there are characters we would like to recognize.

Face Recognition :

In face recognition the input are an image, the classes are people to be recognized and the learning program should learn to associative the face images to identities

Speech Recognition :

In Speech Recognition The input is acoustic and the classes are words the can be uttered. This time the association to be learned is from an acoustic Signal to a word of some language.

Biometric Recognition :

Biometric Recognition is a authentication of people using their physiological and or behavioural characteries that requires an integration of inputs from different modalities. As opposed to the usual identification producers photo, printed signature or password when there are many different input forgeries would be more difficult and The System would be more acurate hopefully without too much inconvenience to The user

Knowledge Extraction :

Learning a rule from data also allows knowledge extraction. The rule is a simple model that explains the data and looking at this model we have an explanation about the process underlying the data.

Compression :

compression is nothing but fitting a rule to the data, requiring less memory to store and less computation to process.

Outlier detection:

Outlier detection is used for finding the instances that do not have characteristics are typical instances share characteristics that can be simply stated and instances share characteristics are typical

Novely detection :

The instance that falls outside as an exception, which may be an anomaly requiring attention such as fraud or may be a novel previously unseen but valid case and hence the other name is novelty detection.

Vapnik Chervonenkis Dimensions:

The dataset containing N points. These N points can be labelled in 2^N ways as positive and negative. Therefore 2N different learning problems can be defined by N data points. If any of these problems we can find a hypothesis $h \in \mathcal{H}$ that sparates the positive examples from the negative, then we say $\mathcal{H}$ shatters N points. That is any learning problem definable by N examples can be learned with no error by a hypothesis drawn from $\mathcal{H}$. The Maximum number of points that can be shatter by $\mathcal{H}$ is called the Vapnik Chervonenkis (VC) dimension of $\mathcal{H}$ is denoted as $VC(\mathcal{H})$ and measures the capacity of $\mathcal{H}$

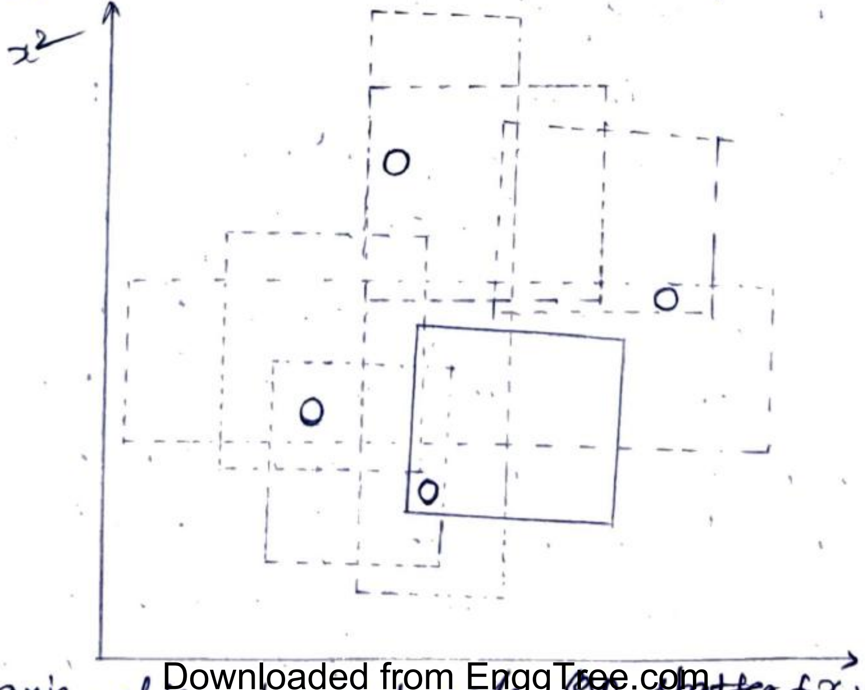


fig: An axis - aligned rectangle can shatter four points only rectangle covering two points are sho

In above diagram we see that an axis aligned rectangle can shatter four points in two dimensions. Then VC(H), When H is the hypothasis class of axis-aligned rectangles in two dimensions Then VC(H), When H is the hypothesis class of axis- aligned rectanges in too dimension is four. In calculating the VC dimensions, it is enough that we find four points that can be Shattered it is not necessary that we able to shatter any four points in two dimensions.

The four points placed on the line cannot be Shattened by rectangles can Separate the positive and negative examples for all possible labellings.

VC dimensions may seem pessimistic. It tell us that using a rectangle as our hypothesis class, we can learn only datasets of four points is not very useful. However this is because the VC dimension is independent of the probability distribution from which instances are drawn. In Real life, The world is smoothly changing instances close by most of the time have the Same label and we need not worry about all possible labellings.

probably Approximately Correct learning

probaly Approximately Correct learning is a framework used for mathematical analysis. A PAC learner tries to learn a Concept by Selecting a hypothesis from a Set of hypotheses that has a low generalization error. In the Context of Machine learning a problem is PAC-learnable if there is an algorithm A when given some idependently drawn Sampes, will produce a hypothesis with a Small error for any Distribution D and any Concept C and that too with a high probability.

In a probably approximately correct (PAC) learning, given a class, C and examples drawn from Some unknown but fixed probability distribution p(x), we want to find the number of examples, N, Such that with probability at least 1- δ, the hypothesis h has error at most ε, for arbitrary $\delta \le 1/2$ and $\varepsilon > 0$

$$P\{C \Delta h \le \varepsilon\} \ge 1 - \delta$$

Where $C \Delta h$ is the region of the difference between C and h

In our case, because S is the tightest possible rectangle, the error region between C and h = S is the sum of four rectangular strips we would like make sure that the probability of a positive example falling in here (and causing an error) is at most ε. For any of these strips if we can goarantee that the probability is upper bounded by ε/4, the the error is at most 4(ε/4) = ε. Note that we count the overlaps in the corners twice and the total actual error in this case is less than 4(ε/4). The probability that a randomly drawn example misses this strip is 1 - ε/4. The probability that all N independent draws miss the strip is $(1-\epsilon/4)^N$ and the probability that all N idependent draws miss the strip $(1-\epsilon/4)^N$ and the probability that all N idependent draws miss any of the four strips is at most $4(1-\epsilon/4)N$, which we would like to be at most δ. we have the inequality

$(1-x) \leq \exp[-x]$

So if we choose N and δ such that we have

$$4 \exp[-\epsilon N/4] \leq \delta$$

We can also write $4(1-\epsilon/4)^N \leq \delta$ Dividing both sides by 4, taking natural log and rearranging terms, we have

$$N \geq (4/\epsilon)\log 4/\delta)$$

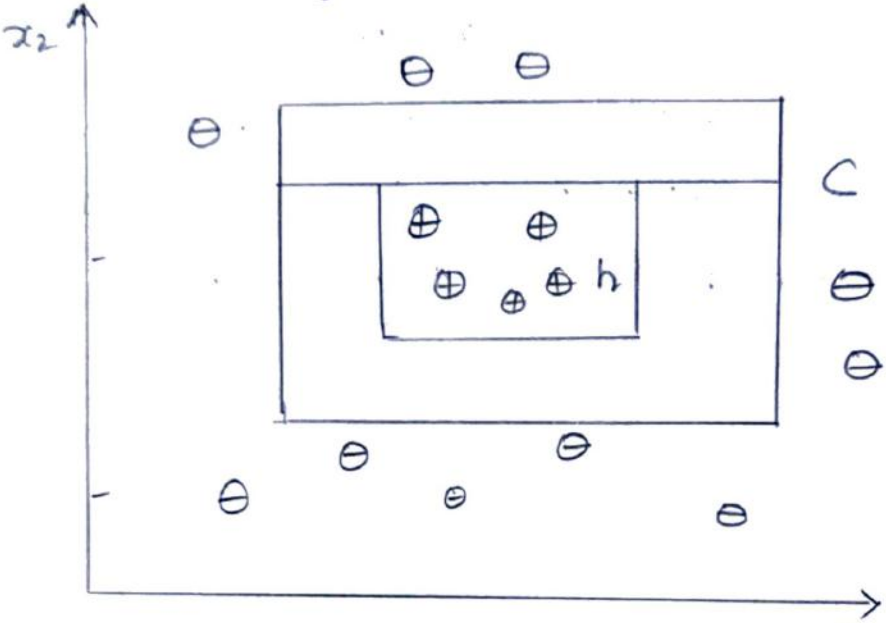


fig: The difference between h and C is the sum of four rectangular strips one of which is shaded.

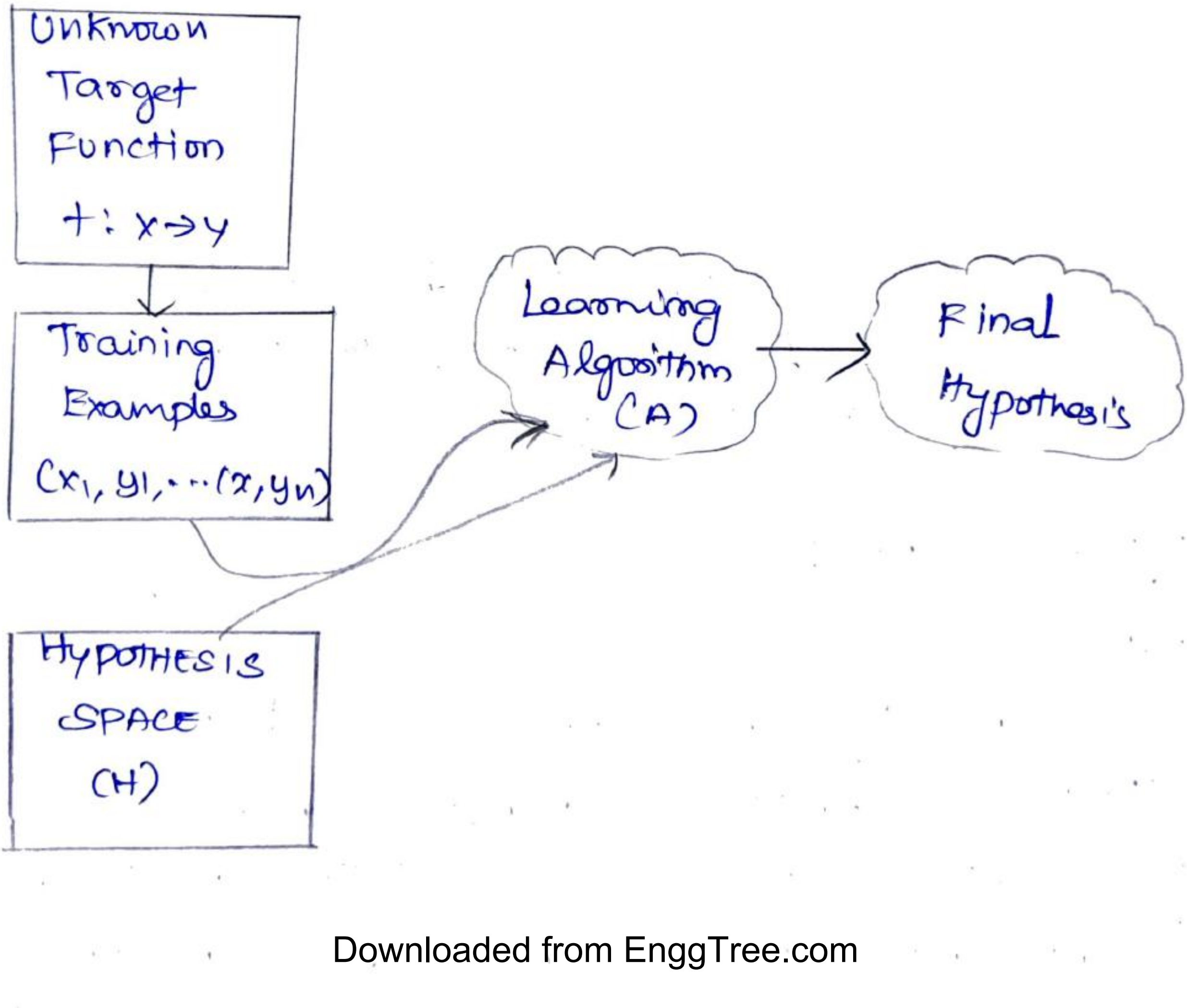
Hypothesis Space:

Hypothesis:

The Hypothesis Space is defined as the Supposition or proposed explanation based on insufficient evidence or assumptions. It is just as guess based on some known facts but has not yet been proven. A good hypothesis is testable, which results is either true or false

Hypothesis in Machine Learning (ML)

The hypothesis is one of the commonly used concepts of Statistics in Machine learning. It is Specifically used in supervised learning Where an ML model learns a function that best maps the input to corresponding outputs with the help of an avilable dataset.



Candidate Eliminate Algorithm :

It is a method for learning concepts from data that is Supervised. Given a hypothesis Space H and a Collection E of instances, the candidate elimination procedures develop the verison Space progressively.

The examples are introducedonby one with each one potentially shrinking the version Space by deleting assumption the Condradict the example. For each new case the candidate elimation method updates the general and particular boundaries.

Verison Space:

Its a cross between a generic and a specific theory. It didnot simply write one hypothesis. It wrote a list of all feasible hypotheses based on the training data.

$$VS_{H,D} \equiv \{ h \in H \mid Consistent(h, D) \}$$

H → hypothesis Space

D → training example.

Where the VSH, D, denoted as Version space and the Subset of hypotheses from H.

For example consider the following dataset,

The classic example of Enjoy Sport

sky	Temp	Humid	Wind	Water	Forest	Enjoy Sport
Sunny	warm	normal	strong	Warm	same	yes
Sunny	warm	high	Strong	Warm	same	yes
rainy	cold	high	strong	Warm	change	no
sunny	warm	high	strong	Cool	change	yes

Specific Hypothesis :

If a hypothesis h covers none of the negative cases and there is no other hypothesis h that covers none of the negative examples then h is strictly more general than h is said to be the most specific hypothesis.

The Specific hypothesis fills in important details about all the variables given in the hypothesis.

$S = \langle '\phi', '\phi', '\phi', '\phi', '\phi', '\phi' \rangle$

General hypothesis:

In general, a hypothesis is an explanation for anything. The general hypothesis explains the relationship between variables in general

$G = \langle '?', '?', '?', '?', '?', '?' \rangle$

Representation:

The most specific hypothesis is represented using ϕ.

The most general hypothesis is represented using '?'.

Why Candidate Elimination Algorithm?

Candidate Elimination Learning Algorithm addresses several of the limitations of FIND-S.

Although the FIND-S algorithm outputs a hypothesis from H, That is consistent with the training examples. This is just one of many hypotheses from H that might fit the training data equally well. They key idea in the CANDIDATE-ELIMINATION Algorithm is to output a description of the set of all hypotheses consitent with the training examples

Algorithm :

1. Initialize both Specific and general hypothesis

$S = \langle 'ϕ', 'ϕ', 'ϕ', 'ϕ', 'ϕ', 'ϕ' \rangle$

$G = \langle '?', '?', '?', '?', '?', '?' \rangle$

Depending on the number of attributes.

2. Take the next example, if the taken example is postive make a specific hypothesis to general

3. If the taken example is negative make the general hypothesis to a more Specific hypothesis.

SOLUTION :

Step 1 :

S_1 = [Sunny , Warm , Normal , strong, Waraug same]

G_1 = [<?, ?, ?, ?, ?, ?>, <?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?>
<?, ?, ?, ?, ?, ?>, < ?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?>]

Step 2 :

S_2 = [Sunny , warm , ? , strong , Warm , Same]

G_2 = [<?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?>
<?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?>]

Step 3 :

S_3 = [Sunny, Warm, ? , strong, Warm, Same]

G_3 = [<sunny, ?, ?, ?, ?, ?> < ?, cold, ?, ?, ?, ?>
< ?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?>
<?, ?, ?, ?, ?, Same>]

Step 4 :

S_4 = [sunny, Warm, ?, strong, ?, ?]

G_4 = [<sunny, ?, ?, ?, ?, ?> <? warm , ?, ?, ?, ?>
<?, ?, ?, ?, ?, ?> <?, ?, ?, ?, ?, ?>
<?, ?, ?, ?, ?, ?>]

SAMPLE ERROR AND TRUE ERROR:

* The Sample error ($err_S(h)$) of h with respect to target function (f) and data sample (s) is the proportion of examples h misclassifies. The Sample test error is the mean error over the test Sample.

$$err_S(h) = \frac{1}{n} \sum_{i=1}^{n} L((f(x_i), h(x_i))$$

* The true error of hypothesis h with respect to target function f and distribution D is the probability that h will misclassify an instance drawn at random according to D.

$$Err(h) = E[L((f(x), h(x)))]$$

An $error_D(h)$ is the true error of hypothesis h with respect to the target function f and data distribution D. It is the probability h will misclassify an instance drawn at random according to D.

An error (h) is the same error of h with respect to the target function f and data Sample set S. It is the proportion of examples in S that h misclassifies

INDUCTIVE BIAS

The candidate Elimination algorithm will converge toward the true concept provided it is given accurate training examples and provided its intial hypothesis Space contains the target Concept.

What if the target Concept is not Containe in the hypothesis space?

Can we avoid this difficulty by using a hypothesis space that includes every possible hypothesis?

How does the size of hypothesis Space influence that ability of algorithm to generalize to unobserved instances.

How does the size of the hypothesis Space influence the number of training examples that must be observed?

In Enjoy Sport example we restricted the hypothesis Space to include only conjuctions of attribute values. Because of this restriction, The hypothesis Space is unable to represent even simple disjunctive target Concepts such as sky = Sunny or Sky = cloudy.

From first two examples: S2: <?, Warm, Normal, strong, cool, change>

This is inconsistent with third examples and there are no hypothesis consistent with these Three examples PROBLEM: We have biased the learner to consider only conjunctive hypotheses. We require a more expressive hypotheses Space.

The obvious solution to the problem of assuring that the target concept is in the hypothesis Space. It is to provide a hypothesis space capable of representing every teachable concept.

Inductive Bias - Fundamental property of Inductive Inference:

A learner that makes no a prior assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instances.

Inductive leap: A learner should be able to generalise training data using prior assumptions in order to classify unseen instances.

The generalization is known as inductive leap and our prior assumptions are the inductive bias of the learner.

Inductive Bias of Candidate-Eliminate algorithm is that the target concept can be represented by a conjunction of attribute Values, the target concept is contained in the hypothesis space and training examples are correct.

Inductive Bias - Form Definition:

Consider a Concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X and let $D_c = \{<x, c(x)>\}$ be an arbitary set of training examples of c

Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c

The Inductive bias of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c the following formula holds.

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \angle X_i, D_c)]$$

BIAS VARIANCE TRADE-OFF:

In the experimental practice we observe an important phenomenon called the bias variance dilemma.

In supervised learning the class value assigned by the learning model build based on the training data may differ from the actual class value The error in learning can be of two types, errors due to 'bias' and error due to 'variance'.

The bias - variance dilemma is the problem of simultaneously minimizing two source of errors that prevent Supervised learning algorithm from generalizing beyond their training set.

The bias is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs.

The variance is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting, modeling the random noise in the training data, rather than the intended outputs.

In order to reduce the model error, the designer can aim at reducing either the bias or the variance, as the noise Coments is irreducible.

As the model increases in complexity, its bias is likely to diminish.

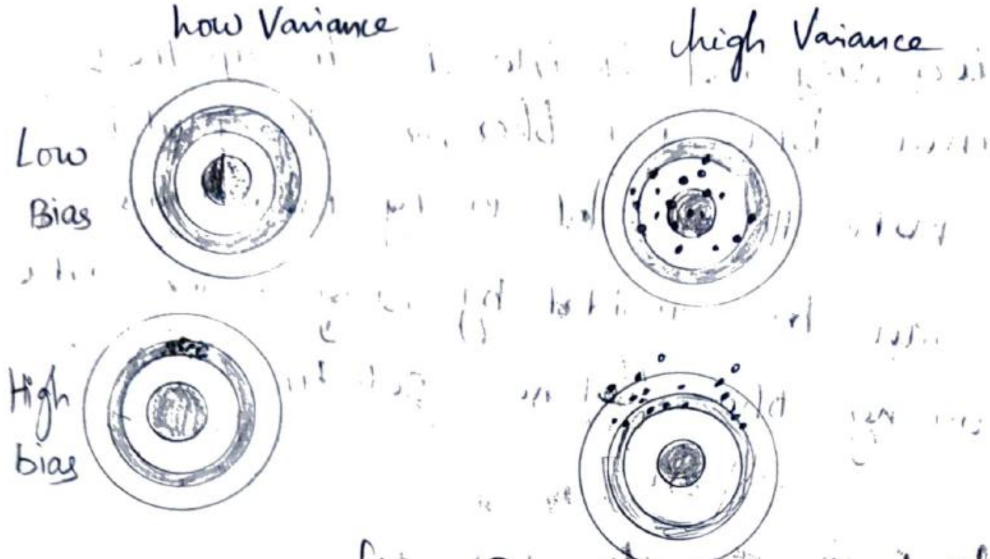
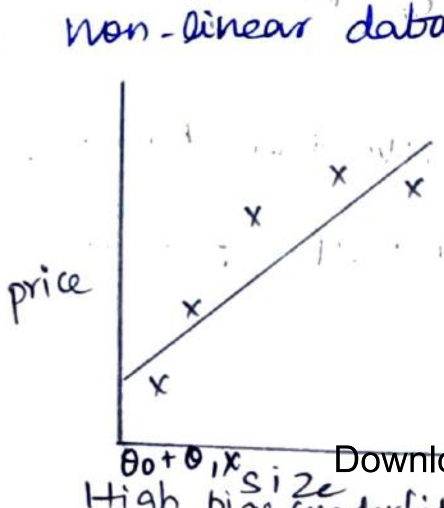


fig : Bias Variance trade off

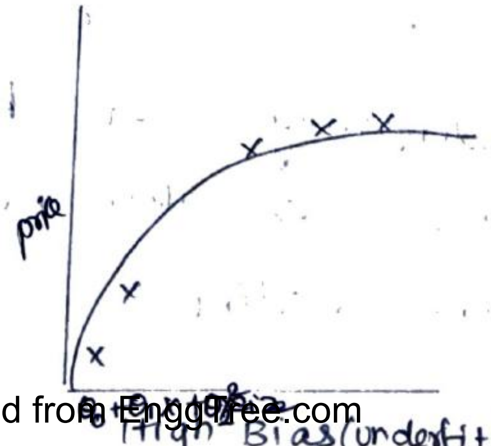
Underfitting (High bias and low variance) :

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data.

It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data.



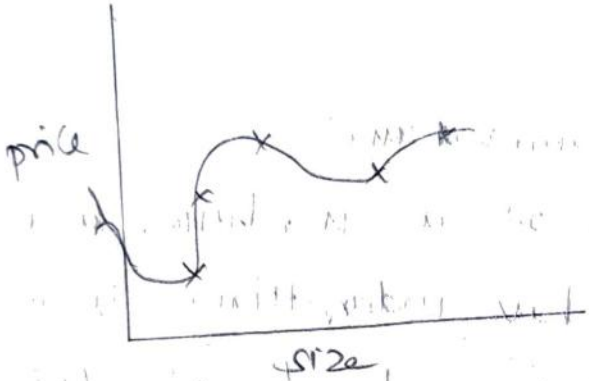
High bias (underfit)



High Bias (underfit)

In such cases the rules of the machine learning model are too easy and flexible to be applied on such minimal data and therefore the model will propably make a list of wrong predictions

Underfitting can be avoided by using more data and also reducing the feature selection.



high Variance (overfit)

In such cases the rules of Machine learning model are too easy and flexible to be applied on such minimal data and therefore the model will propably make a list of wrong predictions

Underfitting can be avoided by using more data and also reducing the features by feature selection.

OVERFITTING (HIGH VARIANCE & LOW BIAS) :

A statistical model is said to be overfitted When we train it with a lot of data.

When a model gets trained with so much of data it starts learning from the noise and inaccurate data entries in our dataset

Then the model does not categorize the data correctly, because of too many details and noise

The cause of overfitting are the non parametric and non linear methods because these types of Machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.

A Solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision tree.

TWO MARKS

1. Define Learning:

Learning is a phenomenon and process which has manifestations of various aspects. Learning process includes gaining of new symbolic knowledge and development of cognitive skills through instruction and practice. It is also discovery of new facts and theories through observation and experiment.

2. Define Machine learning.

A Computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T as measured by P improves with experience E

3. Describe the issues in Machine learning.

* What learning algorithm to be used?
How much training data is sufficient?
* When and how prior knowledge can guide the learning process?
* What is the best strategy for choosing a next training experience?

4. What is an influence of information theory on machine learning?

Information theory is measures of entropy and information content. Minimum description length approaches to learning. Optimal codes and their relationship to optimal training Sequences for encoding a hypothesis

5. What is meant by target fnction of a learning program?

Target fnction is a method for solving a problem than an AI algorithm parses its training data to find. Once an algorithm find its target fnction, that fnction can be used to predict results.

6. Define usefol perspective on Machine learning

One useful perspective on machine learning is that it involves Searching a very large Space of possible hypotheses to determine one that best fits the observed data related to inputs for any prior knowledge held by the learner.

7. What is decision tree?

A decision tree is a tree Where each node represent a feature (attribute) each link (branch) respents a decision (rule) and each leaf represents an outcome (categorial or continues values)

8. Define probably approximate learning.

A concept class C is said to be PAC learnable using a hypothesis class H if there exists a learning algorithm L such that for all concepts in C for all instance distribution D on an instance space X.

9. What are nodes of decision tree?

* Each leaf node has a class label determined by majority vote of training examples reaching that leaf.

* Each internal node is a question on features It branches out according to the answers

* Decision tree learning is a method for approximating discrete-valued target functions.

10. Why tree pruning useful in decision tree induction?

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of overfitting data. Such methods typically use statistical measures to remove the least reliable branches.

11. What is tree pruning?

Tree pruning attempts to identify and remove such branches with the goal of improving classification accuracy on unseen data.

12. Rule POST-PRUNING?

* It is method for finding high acuracy hypotheses.

* Rule post-pruning involves the following steps:

⇒ Infer decision tree from training set

⇒ Convert tree to rules - one rule per branch

⇒ prune each rule by removing preconditions that result in improved estimated accuracy

⇒ Sort the pruned rules by their estimated accuracy and consider them in this sequence when classifying unseen instances.

13. Why convert the decision tree to rules before pruning?

Converting to rules allows distinguishing among the different contexts in which a decision node is used.

Converting to rules removes the distinction between attributes tests that occur near the root of the tree and those that occur near the leaves.

Converting to rules improves readability. Rules are often easier for to understand.

14. What is inductive learning

In inductive learning the learner is given a hypothesis space H from which it must select an output hypothesis and a set of training examples

$D = \{(x_1, f(x_1)) \ldots, x_n, f(x_n))\}$ where $f(x_i)$ is the target value for the instance x_i.

15. MODEL SELECTION :

Learning is not possible without inductive bias and now the question is how to choose the right bias. This is called model Selection Which is choosing between possible $\mathcal{H}$.

16. Define ILL-posed problem :

The training set are contains only a small subset of all possible instances as it generally does, that is if we know What the output should be for only a Small percentage of the cases the Solution is not unique. The data by itself is not Sufficient to find a unique solution.

17. What is OSCam Razor?

The empirical error is simple model would generalize better than a Complex model. This principle is known as OSSCam razor Which States that Simpler explanations are more plausible and any unnecessary complexity Should be Shaved off

18. How Machine learn ?

It has three phases

* Training
* validation
* Application

19. Difference between Supervised and Unsupervised learning.

Supervised learning	Unsupervised learning
* Desired output is given	* Desired output is not given
* It is not possible to learn larger and more complex models than with Supervised learning	* It is possible to learn larger and more Complex models with Unsupervised learning.
* Use training data to infer model	* No training data is used
* Every input pattern that is used to train the network is associated with an output pattern	* The target output is not presented to the network
* Trying to predict a function from labelled data	* Try to detect interesting relations in data
* It requires that the target variable is well defined and that a sufficient number of its values are given	* The target value is unknown or has only been recorded for too small a number of cases
* eg: Optical character Recognition	* eg: Find a face in image
* We can test our model	* We can not test our model
* It is also called classification	* It is also called clustering

20. Elements of Reinforcement learning :

Policy

Reward fnction

Value fnction

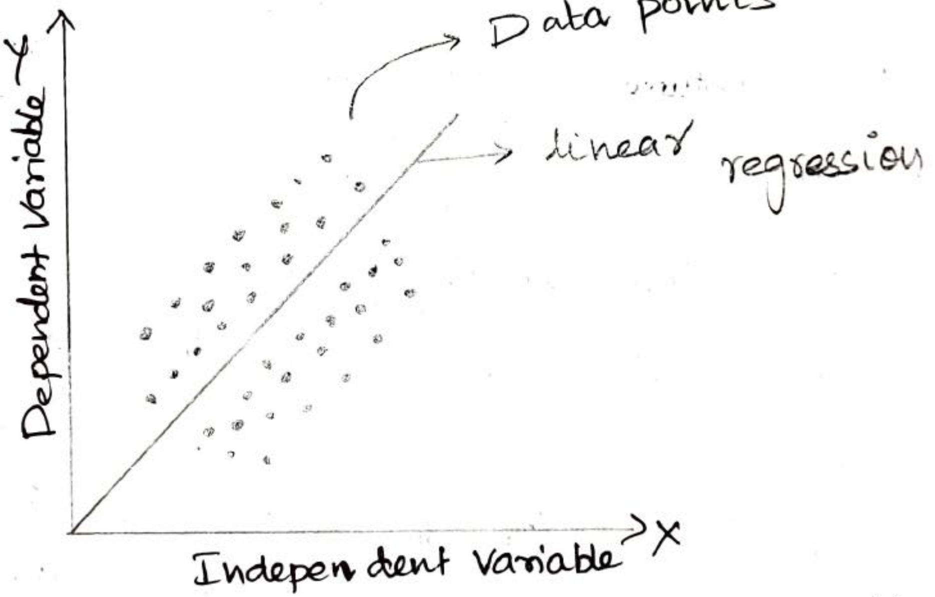
Model of environment

SUPERVISED LEARNING

REGRESSION

Regression find correlations between dependent and independent Variables. If the desired output consists of one or more continous variable then the task is called as regression

Regression algorithm help predict continoous variable such as house prices, market trends, weather patterns oil and gas prices ect.



Regression analysis is a set of statistical model or methods used for the estimation relationship between variables and for modelling the future relationship between them

LINEAR REGRESSION MODELS :

* Linear regression is a statistical method that allows us to summarize and study relationship between two continuous quantitative variables.

* The objectives of a linear regression model is to find a relationship between the input variables and a target variable.

* One variable, denoted y is regared as the response, outcome or dependent variable

* The other, denoted x, is regared as the response, predictor explanatory or independent variable.

Regression models predict a continuous variable such as the sales made on the day or predict temparature of a city. Let's imagine that we fit a line with the training point that we have. If we want to add another data point but to fit it, we need to change existing model.

classification predicts categorial lables (classes), prediction model continuous-valued functions. classification is considered to be supervised learning.

REGRESSION LINE :

It gives the average relationship between the two variables in mathematical form.

For two variables x and y, there are always two lines of regression.

Regression line of x on y gives the best estimate for the value of x for any $X = a + by$

Where

$a \rightarrow x$ - intercept

b = slope of the line

x = Dependent Variable

y = Independent Variable.

Regression line y on x

It gives the best estimate for the value of y for any specific give values of x

$$y = a + bx$$

Where

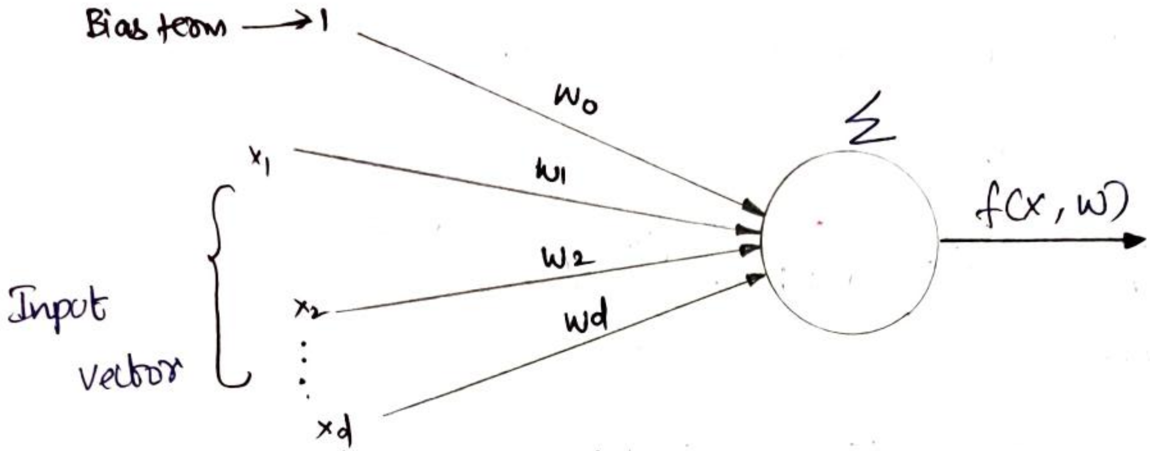
a = y - intercept

b = slope of the line

y = Dependent Variable

x = independen Variable

By using least square method we are able to construct a best fitting to scatter diagram points and then formolate a regression equation in the form of:



$$\hat{y} = a + bx$$

$$\hat{y} = \bar{y} + b(x - \bar{x})$$

Regression analysis is the art and science of fitting straight lines to patterns of data. In linear regression model the variable of interest (dependent variable) is predicted from k other variables (independent Variables using linear equation. If y denotes the the dependent variable and $x_1 \ldots x_k$ are the independent variables then the assumption is that the value of Y at time t in the data sample is determined by the linear equation.

$$Y_1 = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_k x_{kt} + \epsilon_t$$

Where the betas are constants and the epsilons are independent and identically distributed normal random variable with mean zero.

At each split point the "error" between the predicted Value and the actual values is Squared to get a "Sum of Squared Errors (SSE)". The Split point errors across the variables are compared and the variable point yielding the lowest SSE is Chosen as the root node slipt point. This process is recursevily continued.

Advantages:

Training a linear regression model is usually much faster than method such as neural networks.

Linear regression models are Simple and require minimum memory to implement.

LEAST SQUARE:

The method of least Squares is about estimating parameters by minimizing the Squared discrepancies between observed data on the one hand and expected values on the other

The least Square Criterion states that the Sum of Square of error is minimum

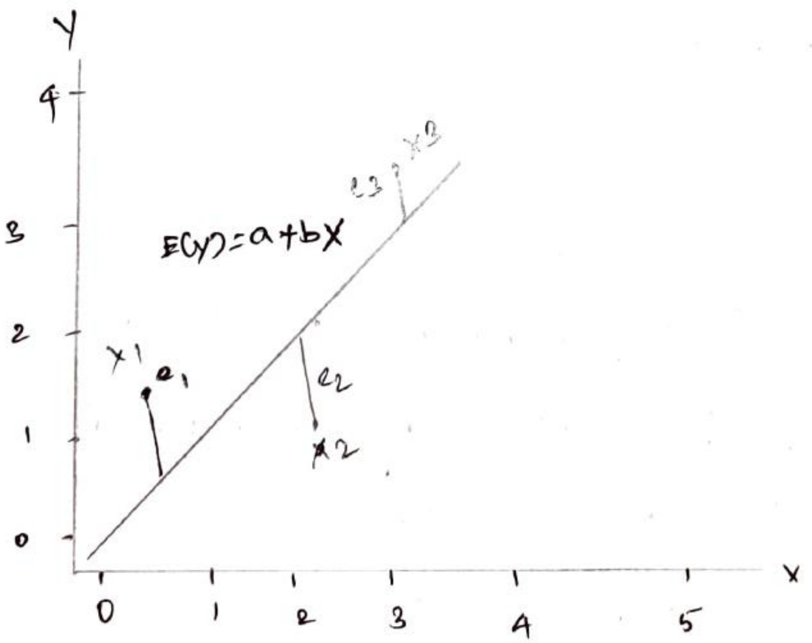
The least square solutions yields $y(x)$ whose sum to 1 but do not ensure outputs to be in the range [0, 1]

How to draw such a line based on the data points observed? Suppose a imaginary line of $y = a + bx$

Imagine a vertical distance between the line and the data point $E = Y - E(Y)$

The error is the deviation of the data point from the imaginary line, regression line. Then what is the best values of a and b?

a & b that minimize sum of such errors.



Deviation does not have good properties for computation Then why do we use square of deviation ?

Let us get a & b that can minimize the sum of Squared deviations. this method is called least Squares.

Least Square method minimizes the sum of Squares of errors. Such as a & b are called least Square estimators. i.e estimators of parameters α & B

The process of getting parameter estimators (eg. a&b) is called estimation. Least square method is the estimation method of ordinary least Squares(OLS).

Disadvantages of least Square

- * Lack robustness to outliers
- * Certain datasets Unsuitable for least Square classification.
- * Decision boundary corresponds to Machine learning Solution

MULTIPLE REGRESSION :

Regression analysis is used to predict the Value of one or more responses from a set of predictors. It can also be used for estimate the linear association between the predictors and responses. predictors can be continuous or categorical or a mixture of both.

If the multiple independent variables affect the response Variable, Then the analysis call for a model different from that used for the Single predictor valuable. In a situvation where more than one independent factor (variable) affects the outcome of a process, a multiple regression model is used This is refermed to as multiple linear regression model or multivariate least Square fitting.

$$Y_1 = \beta_0 + \beta_1 z_{j1} + \beta_2 z_{j2} + \cdots + \beta_r z_{jr} + \varepsilon_j$$

Where ε is the random error

$\beta_i, i = 0, 1 \ldots r$ are un-known ~~co~~ regression

co-efficient

Difference between Simple Regression and Multiple Regression

Simple Regression	Multiple Regression
One dependent Variable Y predicted from one independent Variable X	One dependent Variable Y predicted from a set of independent variables ($x_1, x_2 \ldots x_k$)
One regression coefficient	One regression coefficient for each independent variable

BAYESIAN LINEAR REGRESSION :

* Bayesian linear regression allows a useful mechanism to deal with insufficient data or poor distributed data

* It allows user to put a prior on the coefficients and on the noise so that in the absence of data the priors can take over. A prior is a distribution on a parameter.

* If we could flip the coin an infinite number of times, inferring its bias would be easy by the lais of larage numbers.

* However What if we could only filp the coin a handful of times? Would we guess that a coin is baised if we saw three heads in three flips, an event that happens one out of eight times with unbiased coins? They overfit these data, inferring a coin bias of p=1

* Bayesian methods allows us to estimate model parameters to construct model forecasts and to conduct model comperisons. Bayesian learning algorithms can calculate explicit probabilities for hypotheses.

Bayesian classifiers use a simple idea that the training data are utilized to calculate an observed probability of each class based on feature values.

When the Bayesian classifier is used for unclassified data, it uses the observed probabilities to predict the most likely class for the new features.

Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.

Bayesian methods can accommodate hypotheses that make probabilistic predictions. New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.

Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.

Uses of Bayesian classifiers:

*used in text-based classification for finding spam or junk mail filtering

* Medical diagnosis

* Network security such as detecting illegal instruction.

Basic procedure for implementing Bayesian linear Regression :

Specify priors for the model parameter

Create a model mapping the training inputs to the training outputs.

Have a Markov chain Monte Carlo (MCMC) algorithm draw Samples from the posterior distributions for the parameters.

Gradient Descent :

Gradient descent is a first-order optimization algorithm. To find a local minimum of a fuction using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point.

Gradiant descent is popular for very larage scale optimization problems because it is easy to implement, can handle black box functions and each iteration is cheap

The gradient will give the slope of the curve at that x and its direction will point to an increase in the function. so we can change x in ~~the~~ opposite direction to lower the function value.

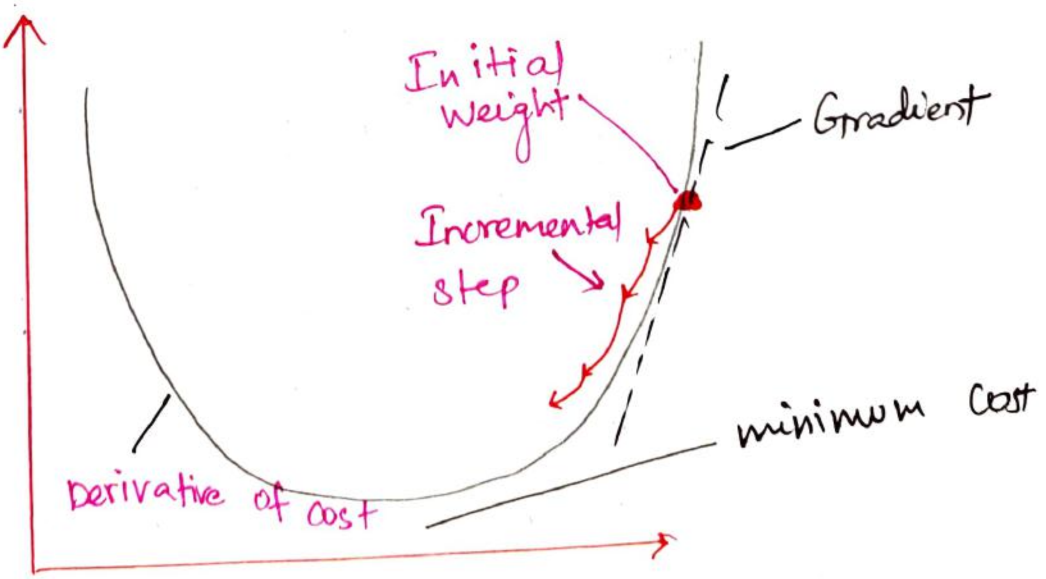
$$X_{k+1} = x_k - \lambda \nabla f(x_k)$$

The $\lambda > 0$ is a small number that forces the algorithm to make small jumps.

Limitation of Gradient Descent

Gradient descent is relatively slow close to the minimum: technically its asymptotic rate of convergence is inferior to many other methods.

For poorly conditioned convex problems gradient descent increasingly 'zigzags' as the gradient points nearly orthogonally to the shortest direction to a minimum point.



If we move towards a negative gradient or away from the gradient of the function at the current point it will give the local minimum of the function.

Whenever we move towards a positive gradient or towards the gradient of the function at the current point, we will get the local maximum of the function.

This entire procedure is known as Gradient ascent which also known as steepest descent. The main objective of using a gradient descent algorithm is to minimize the cost function using iteration.

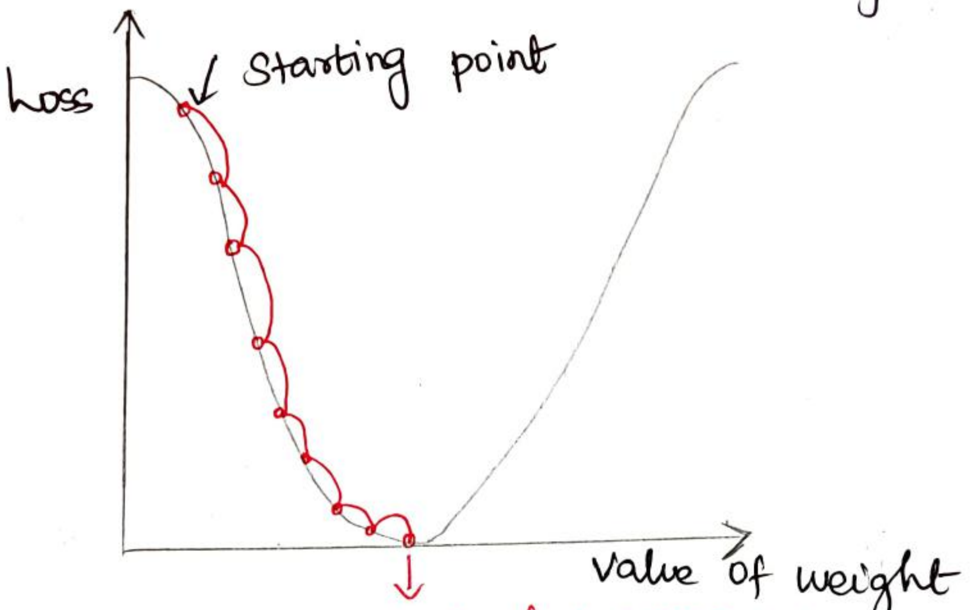
calculate the first order derivative of the function to compute the gradient or slope of the function.

Move away from the direction of the gradient which means slope increased from the current point by alpha times, where alpha is defined as learning Rate. It is a tuning parameter in the optimization process which helps to decide the length of the steps.

Working of Gradient Descent

$$y = mx + c$$

Where m represents the slope of the line and c represents the intercepts on the y-axis



point of convergence
i.e where the cost function is at its minimum

The Slope becomes steeper at the starting point or arbitary point but whenever new parameters are generated then steepness gradually reduces and at the lowest point, it approaches the lowest point Which is called a point of convergence.

Learning Rate: It is defined as the step size taken to reach the minimum or lowest point. This is typically a small value that is evaluated and updated based on the behavior of the cost function. If the learning rate is high it results in larger steps but it also leads to risks of overshooting the minimum.

At the same time a low learning rate shows
the small step sizes which compromises overall efficiency but gives the advanage of more precision.

Types of Gradient Descent

1. Batch Gradient Descent
2. Stochastic gradient Descent
3. Mini Batch Gradient Descent

Batch Gradient Descent :

It is used to find the error for each point in the training set and update the model after evaluvating all training examples. It is known as training epoch.

Stochastic gradient Descent

Stochastic gradient Descent is a type of gradient Descent that runs one training example per iteration. It is more efficient for large datasets.

Mini batch Gradient Descent

Mini batch gradient Descent is the combination of both batch gradient descent and stochastic gradient descent. It divides the training datasets into Small batch size then performs the updates

LINEAR CLASSIFICATION MODELS :

A classification algorithm that makes its classification based on linear predictor function combining a set of weights with the feature vector.

It does classification decision based on the value of a linear combination of the characteristic Imagine that the linear classifier will merge into its weights all the characteristics that define a particular class.

Discriminative functions :

Linear Discriminant Analysis (LDA) is one of the commonly used dimensionality reduction techniques in machine learning to slove more than two - class classification problems. It is known as Normal Discriminant Analysis (NDA) or Discriminant Function Analysis (DFA)

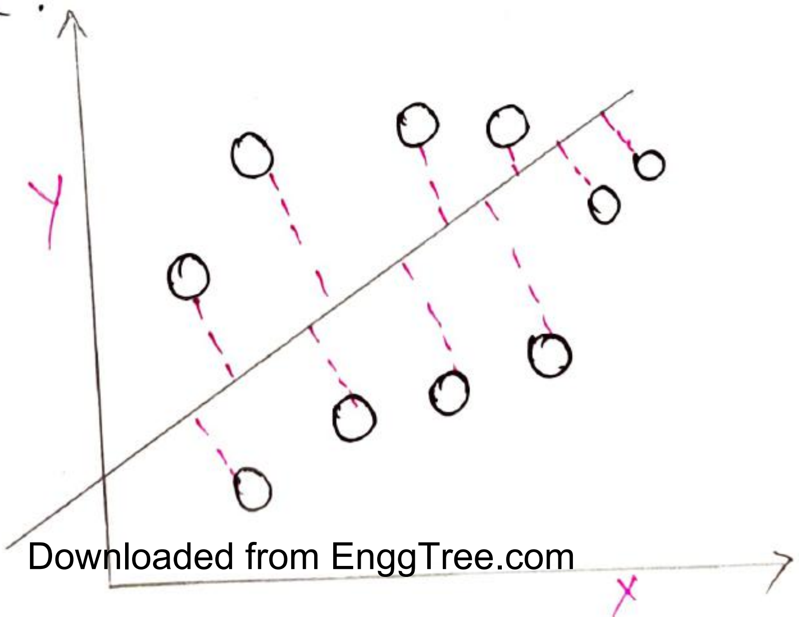
Linear Discriminant analysis is one of the most popular dimensionality reduction techniques used for Supervised classification problems in machine learning. It is also considered a pre-processing step for modeling differences in ML and application of

Linear Discriminant analysis is used as a dimensionality reduction technique in machine learning using Which we can easily transform a 2-D and 3-D graph into a one dimensional plane

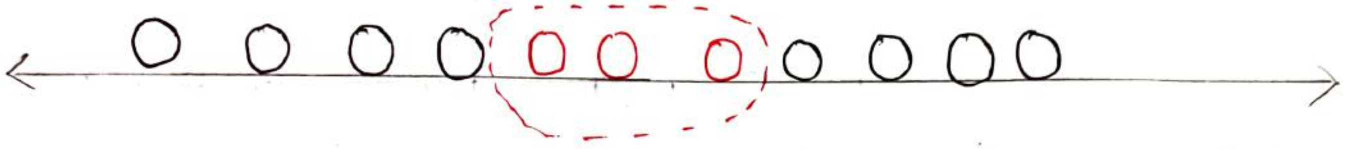
Let's consider an example where we have two classes in a 2-D plane having an X-y axis and we need to classify them efficiently. As we have already seen in the above example that LDA enable us to draw a straight line that can completely separate the two classes of data points.

Here LDA uses an X-Y axis to create a new axis by separating them using a straight line and projecting data onto a new axis.

Hence we can maximize the separation between these classes and reduce the 2-D plane into one dimensional.



pattern classification. For eg, If we have two classes with multiple features and need to separate them efficiently. When we classify them using a single feature, Then it may show overlapping.

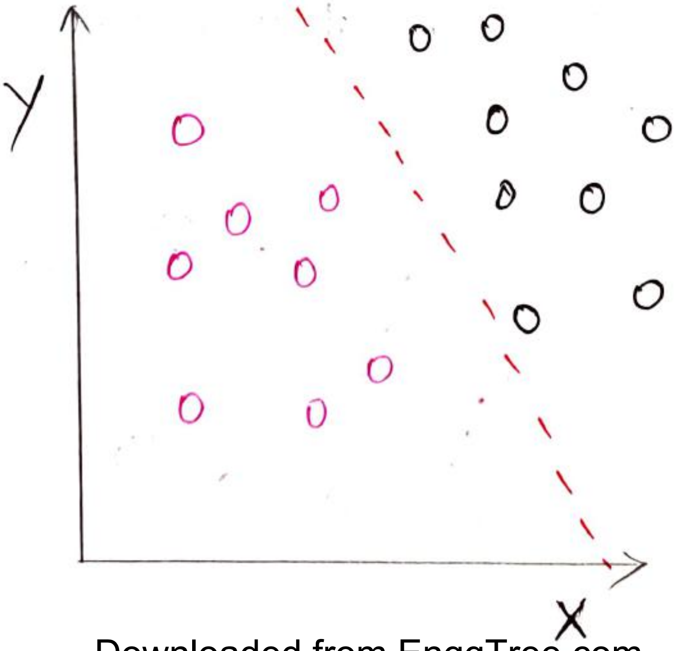


Overlapping.

To overcome the overlapping issue in the classification process, we must increase the number of features regularly.

Eg:

Let's assume we have to classify two different classes having two sets of data points" in a 2-dimensional plane



To create a new axis, Linear Discriminant Analysis uses the following criteria:

⇒ It maximizes the distance between means of two classes.

⇒ It minimizes the variance within the individual class.

In other words we can say that the new axis will increase the separation between the data points of two classes and plot them onto the new axis.

LOGISTIC REGRESSION:

* Logistic regression is one of the most popular Machine learning algorithms, Which comes under the Supervised learning technique.

* It is used for predicting the categorical dependent Variable using a given set of Independent variables.

* Therefore the outcome must be a categorical or discrete value. It can be either yes or No, 0 or 1, tone or flase ect. but Instead of giving the exact value as 0 & 1 it gives the probabilistic Values Which lies between 0 & 1

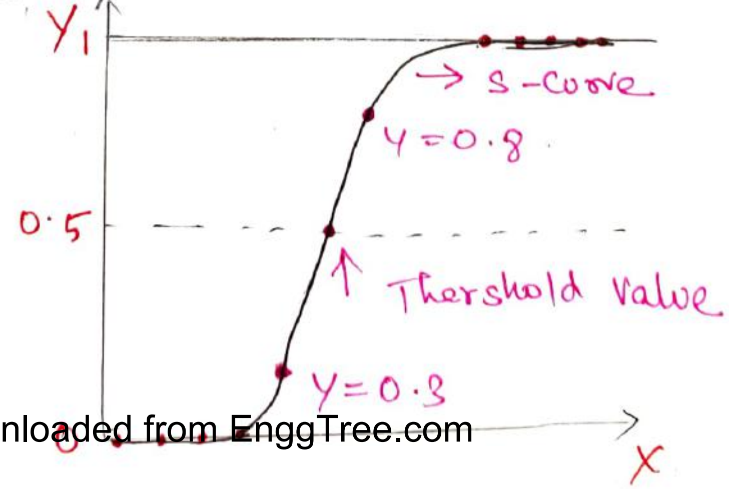
* Logistic regression is used for sloving the classification problems.

* In logistic regression, instead of fitting a regression line, we fit an `S' shaped logistic function, Which predicts two maximum values(0&1)

* In logistic regression instead of fitting a regression line, we fit an "S" shaped logistic function Which predicts ~~two~~ two maximum values (0 and 1)

It is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continous and discrete datasets

It can be used to classify the observation using different types of data and can easily determine the most effective variables used for the classification.



LOGISTIC FUNCTION :

* The sigmoid function or logistic function used to map the predicted values to probabilities

* It maps any real value into another value within a range of 0 and 1 .

* The value of the logistic regression must be between 0 and 1 which cannot go beyond this limit so it forms a curve like the "S" form.

* The dependent variable should be categorical in nature

* The independent variable should not have multi-collinearity.

Logistic Regression Equation:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

In logistic Regression y can be between 0 and 1 only so for this lets divide the above equation by (1-y):

$$\frac{y}{1-y} \; ; \; 0 \text{ for } y=0 \text{ and infinity for } y=1$$

* But we need range between -[infinity] to +[infinity] then take logarithm of the equation it will become

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots b_nx_n.$$

Type of Logistic Regression

Binomial:

In this regression there can be only two possible types of the dependent variables such as 0 or 1 pass or fail ect.

Multinomial:

In this regression there can be 3 or more possible unordered types of the dependent variable such as "cat" or "dog" or "sheep".

Ordinal: In this regression there can be 3 or more possible ordered types of dependent variables such as "low", "medium", or "high".

GENARATIVE MODEL:

* Generative models are a class of statistical models that generate new data instances.

* These models are used in unsupervised machine learning to perform tasks such as probability and likelihood estimation, modelling data points and distinguishing between classes using these probabilities.

* Generative models rely on the Bayes theorem to find the joint probability.

* Generative model describe how data is generated using probabilistic models.

* They predict P(y/x), the probability of y given x, Calculating the P(x,y), the probability of x and y.

NAVIE BAYES CLASSIFIER:

* Navie Bayes algorithm is a Supervised learning algorithm which is based on "Bayes the oream" and used for solving classification problems.

* It is mainly used in text classification that includes high dimensional training dataset.

It helps in building the fast machine learning algorithms or models that can make quick predictions.

* It is a probabilistic classifier which means it predicts on the basis of the probability of an object.

* Some popular examples of Naive Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

* It is called Naive because of it assumes that the occurence of a certain feature is independent of the occurence of other features. Such as if the fruit is identified on the bases of color, shape and taste, then red, spherical and sweet fruit is recognized as an apple.

* Hence each feature individually contributes to identify that it is an apple without depending on each other.

Baye's Theorem:

It is also know as Bayes Rule or Bayes law which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes Theorem is given as

$$P(A/B) = \frac{P(B/A)\,P(A)}{P(B)}$$

Where

P(A/B) is a posterior probability, probability of hypothesis A on the observed event B

P(B|A) is a likelihood probability, probability of the evidence given that the probability of the hypothesis is true

P(A) is a prior probability, probability of hypothesis before evidence.

P(B) is Marginal probability, probability of evidence.

Difference between generative and Discriminative Model

Generative model	Discriminative model
It generates new data	This discriminate between differents kind of data instances
Generative model revolves around the distribution of a dataset to return a probability for a given example.	It makes predictions based on conditional probability and is either used for classification or regression

* Generative model capture the joint probability P(x, y) or just p(x) if there are no labels.	* Discriminative models capture the conditional probability p(y/x)
*A generative model includes the distribution of the data itself, and tells you how likely a given example is	*A discriminative model ignores the question of whether a given instance is likely and just tells you how likely a lable is apply to the instance.
* Generative models are used in unsupervised machine learning to perform task such as probability a likelihood estimation.	* This model particularly used for supervised learning.
* Eg: Gussains, Nawie Bayes	*Eg: logistic Regression SVM

SUPPORT VECTOR MACHINE:

* Support Vector Machines (SVMs) are a set of supervised learning methods which learn from the dataset and used for classification.

* SVM is a classifier derived from statistical learning theory by Vapnik and Chervonenkis.

Simply speaking we can think of an SVM model as representing the examples as points in space mapped so that each of examples of the separate classes are dividied by a gap that is wide as possible.

Example of Bad Decision Boundaries

SVM are primarily two-class classifiers with the distinct characteristic that they aim to find the optimal hyperplane such that the expected generalization error is minimized.

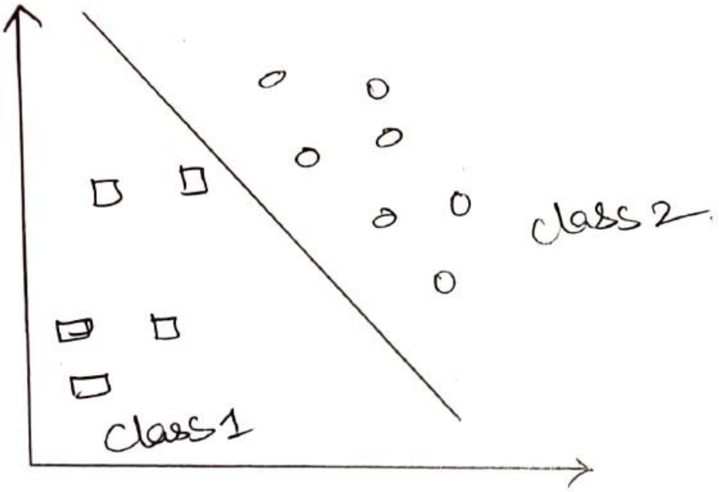
Instead of directly minimizeing the empirical risk calculated from the training data SVMs perform stuctural risk minimization to achieve good generalization.

The empirical risk is the average loss of an estimator for a finite set of data drawn from p.

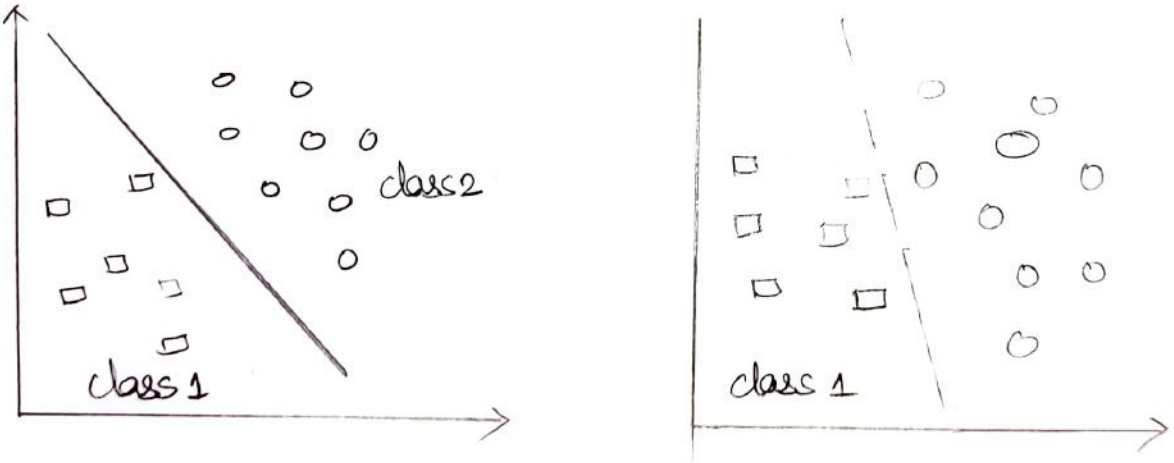
The idea of risk mimization is not only measure the performance of an estimator by its risk, but to actually search for the estimator that minimizes is over distribution p.

* It is a kind of large Margin classifier.

* It is a vector space based machine learning method Where the goal is to find a decision boundary between two classes that is maximally far away any point in training data.



Two class problem



Bad decision boundary of SVM

Given a set of training examples, each marked as belonging to one of two classes an SVM algorithm builds a model that predicts Whether a new example falls into one class or the other

Because we dont know distribution p we instead minimize empirical risk over a training dataset drawn from p. This general learning technique is called empirical risk minimization.

Good Decision Boundary :

The decision bondary should be as far away from the data of both classes as possible.

If the data points lie very close to the bondary, the classifier may be consistent but is more likely to make errors on new instances from the distribution.

Hence we pefere classifiers that maximize the minimal distance of data points to the separator.

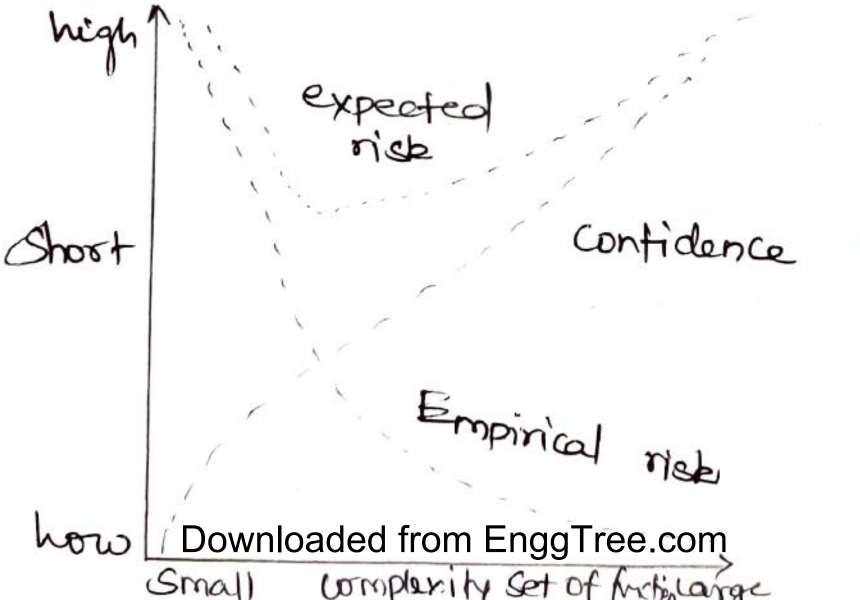
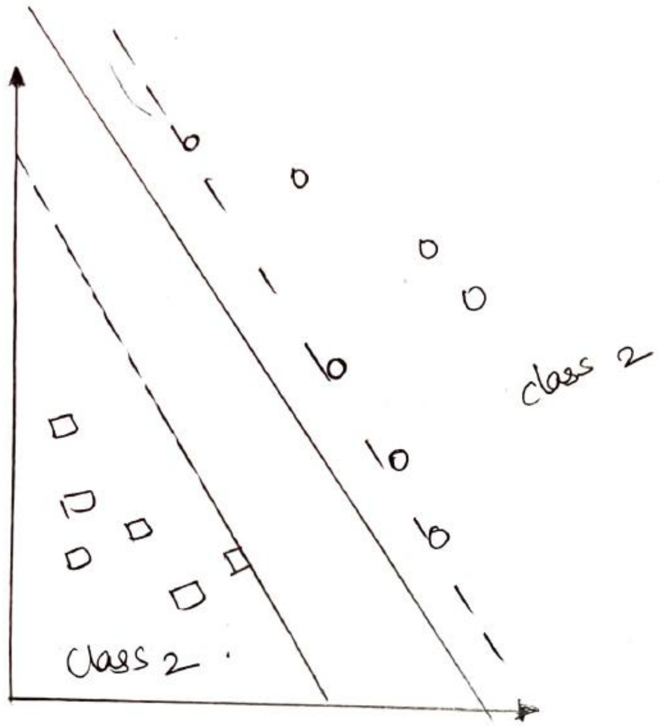


fig: Empirical Risk.



Good decision bomdary.

* The gap between data points and the Classifier bomdary.

* The margin is the minimom distance of any Sample to the decision bomdary.

* Margin of the Separator is distance between Support Vectors.

$$\text{Margin } (m) = \frac{2}{\|w\|}$$

* Maximal margin classifier is a classifier in the family F that maximizes the margin. Maximizing the margin is good according to intoition and PAC Theory.

* Implies that only support vectors matter other training examples are ignorable.

Key properties of SVM :

- * Use a single hyperplane which subdivides the space into two half spaces one which is occupied by class 1 and the other by class 2 .
- * They maximize the margin of the decision bondary using quadratic optimization techniques, which find the optimal hyperplane
- * Ability to handle large feature spaces
- * Overfitting can be controlled by soft margin apporach

SVM Applications

SVM has been used successfully in many real word problems .

- * Text (and hypertext) categorization .
- * Image classification
- * Bioinformatics (protein classification, cancer classification)
- * Hand written character recognition
- * Determination of SPAM email

Limitations of SVM :

* It is sensitive to noise.
* The biggest limitation of SVM lies in the choice of kernel.
* Another limitations is speed and size
* The optimal design for multiclass SVM Classifier is also a research area.

SOFT MARGIN

For the very high dimensional problems common in text classification, sometimes the data are linearly separable.

But in the general case they are not and even if they are, we might prefer a solution that better separates the bulk of the data while ignoring a few weird noise documents.

What if the training set is not linearly separable? slack variables can be added to allow misclassification of difficult or noisy examples resulting margin called soft.

A Soft margin allows a few variables to cross into the margin over the hyperplane.

DECISION TREE

* A decision tree is a simple representation for classifying examples. Decision tree learning is one of the most successful techniques for supervised classification learning

* In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions

* Each leaf node has a class label, determined by majority vote of training examples reaching that leaf.

* Each terminal node is a question on features. It branches out according to the answers.

* Decision tree learning is a method for approximating discrete valued target functions The learned function is represented by decision tree.

* A learned decision tree can also be re-represented as a set of if - then rules

* Decision tree learning is one of the most widely used and practical methods for inductive inference.

* It is robust to noisy data and capable of learning disjunctive expression.

* Decision tree learning method searches a Completely expressive hypothesis

DECISION TREE REPRESENTATION

* Build a decision tree for classifying examples as positive & negative instance of concept.

* Each non leaf node has associated with it an attribute.

* Each leaf node has associated with it a classification (+ or -)

* Each arc node has associated with it one of the possible values of the attribute at the node from which the arc is directed

* Internal node denotes a test on an attribute. Branch represents an outcome of the test. Leaf node represents class labels or class distribution

A decision tree is a flow chart like structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test and tree leaves represent classes or class distributions.

DECISION TREE ALGIORITHM

To generate decision tree from the training tuple of data partition D

Input

Data partition (D)
Attribute List
Attribute Selection method

ALGIORITHM :

⇒ Create a node (N)

⇒ If tuples in D are all same class then

⇒ Return node (N) as a leaf node labeled with the Class C

⇒ If attribute list is empty then return N as a leaf node labeled with the majority class in D

⇒ Apply attribute selection method (D, attribute list) to find the 'best' Splitting criterion.

⇒ Label node N with Splitting attribute

⇒ If splitting attribute is discrete valued and multiway splits allowed.

⇒ Then attribute list -> attribute list -> splitting attribute

⇒ For (each outcome j of splitting criterion)

⇒ Let D_j be the set of data tuples in D statisfying outcome j

⇒ If D_j is empty then attach a leaf labeled with majority class in D to Node N;

⇒ Else attach the node returned by Generate

· Decision tree (D_j, attribute list) to Node N ;

⇒ End of for loop

⇒ Return N

Decision tree generation consists of two phases one is tree construction + pruning.

In tree construction phase, all the training examples are at the root. partition examples recursively based on selected attributes

In tree purning phase, the identification & removal of branches that reflect noise or outliers

Advantages:

* Rules are simple and easy to understand

* Decision tree can handle both nominal & numerical attributes

* Decision tree are capable of handling datasets that may have errors.

* It has capable of handling datasets that may have missing values.

* Decision trees are considered to be a nonparametric method.

* Decision tree are self - explantory.

Disadvantages:

* Most of the algorithms require that the target attribute will have only discrete values

* Some problem are difficult to solve like XOR.

* Decision trees are less appropriate for estimation taks where the goal is to predict the Value of a Continuous attribute

RANDOM FORESTS :

Random forests is a famous system learning set of rules that belongs to the Supervised getting to known method.

It may be used for both classification and regression issues in ML

It is based totally on the concept of ensemble studying that's a process of combining multiple classifiers to solve a complex problem and to enhance the overall performance of the model.

Random forest is a classifier that incorporates some of choice timber on diverse subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Random Forests Algorithm Working

Random forest works in two section first is to create the random woodland by combining N selection trees and Second is to make predictions for each tree created inside the first segment.

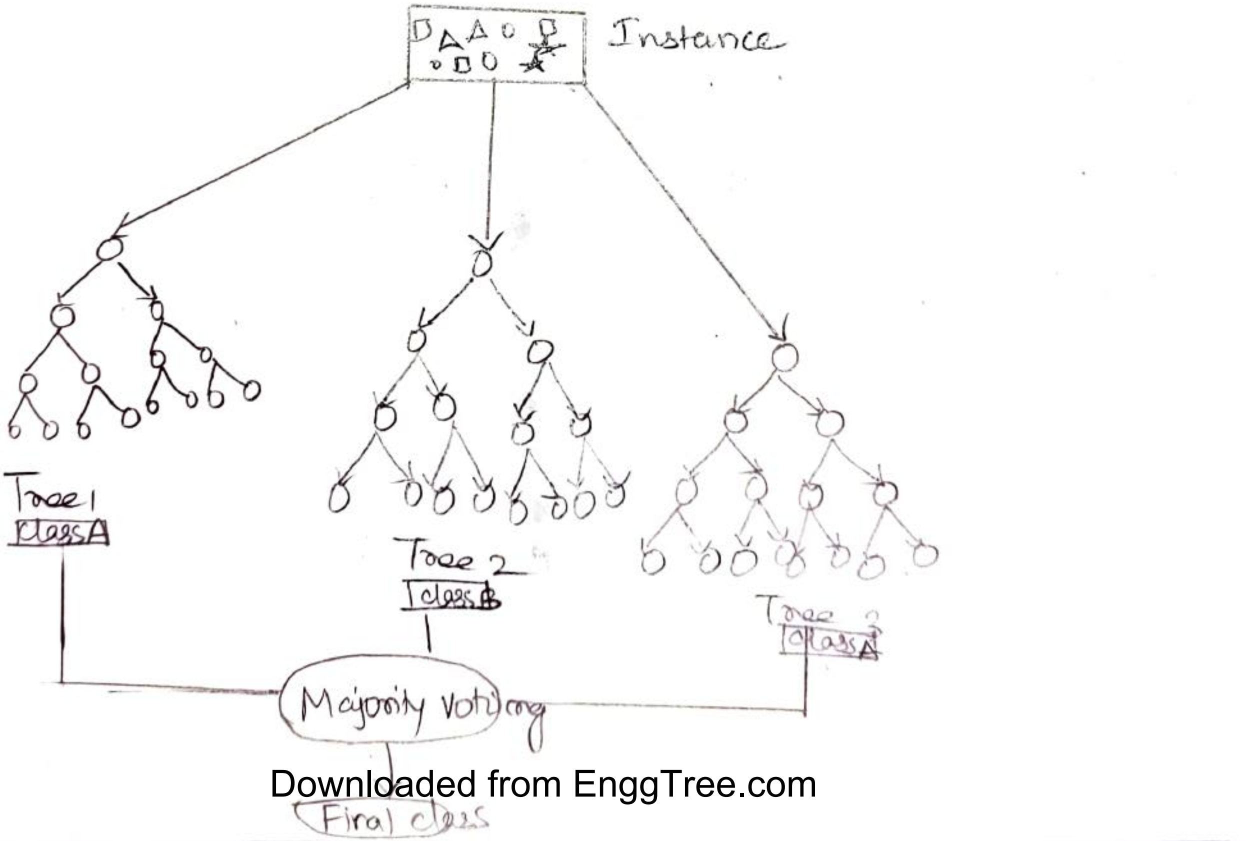
1 ⇒ Select random K statistics points from the Schooling set

2 ⇒ Build the selection trees associated with the selected information points.

3 ⇒ choose wide variety of N for selection trees associated with the seleeted information which we want to build.

4 ⇒ Repeat step 1 and 2

5 ⇒ For new factors locate the predictions of each choice tree and assign the new record factors to the category that wins most people votes.



Application of Random forest :

* Banking ⇒ Banking zone in general uses This algorithm for the identification of loan dangers

* Medicine : With the assistance of this set of roles, disorder traits and risks of the disorder may be recognized.

* Land use : We can perceive the areas of comparable land use with the aid of This algorithm.

* Marketing : Marketing tendencies can be recognized by the usage of this algorithm.

Advantages of Random Forest :

* It is capable of managing large database with high dimensionality.

* It enhances the accuracy of the version and forestalls the overfitting trouble.

Disadvantage :

* Although random forest can be used for both class + regression ~~respon~~ responsibilities it isn't extra appropriate for regression obligations.

UNIT - III

Ensemble Techniques and Unsupervised Learning.

Combining Multiple learners

* When designing a learning machine, we generally make some choices like parameters of machine, training data and representation This implies some of sort of variance in performance

* For example in a classification setting we can use a parametric classifier or in a multilayer perception, we should also decide on the number of hidden units.

* Each learning algorithm dicates a certain model that comes with a set of assumptions.

* This inductive bias leads to errors if the assumptions do not hold for the data

* Different learning algorithm have different acuracies. The no free lunch theorem asserts that no single learning algorithm always acheive the best performance in any domain

* They can be combined to attain high accuracy.

* Data fusion is the process of fusing multiple records representing the same real world object into a single, consistent and clean representation.

* Fusion of data for improving prediction accuracy and reliability is an important problem in machine learning.

* Combining different models is done to improve the performance of deep learning models.

* Building a new model by combination requires less time, data and computational resources

* The most common method to combine models is by averaging multiple models, where taking a weighted average improves the accuracy.

Generating Diverse Learners:

Different Algorithms: We can use different learning algorithms to train different base learners. It make different assumptions about the data and lead to different classifi

Different Hyper-parameters :

We can use the same learning algorithm but use it with different hyper-parameters.

Different Input Representations :

Different representations make different characteristics explicit allowing better identification.

Different Training sets :

Another possibility is to train different base-learners by different subsets of the training set.

Model Combination Schemes

Different methods are used for generating final output for multiple base learners are multiexpert and multistage combination.

Multiexpert Combination :

* It is a methods have base-learners that work in parallel.

* Global approach : given an input all base learners generate an output and those outputs

are used such as voting and stacking.

Local approach: in mixture of experts there is a gating model, Which looks at the input and choose one (or very few) of the learners as responsibl for generating the output.

Multistage Combination:

* It is a methods use a serial approach where the next multistage Combination base-learners are not accurate enough.

* Lets assume that we want to construct a function that maps inputs to outputs from a set of known N train input-output pairs

$$D_{train} = \{(x_i, y_i)\}_{i=1}^{N_{train}}$$

Where $x_i \in X$ is a D dimensional feature input vector, $y_i \in y$ is the output.

Classification: When the output take values in the discrete set class labels $y = \{c_1, c_2 \cdots c_K\}$ Where K is the number of different classes.

Regression consists in predicting continuous ordered outputs $y = R$

voting :

* The Simplest way to combine multiple classifiers is by voting which corresponds to taking a linear combination of the learners.

* Voting is an ensemble machine learning algorithm.

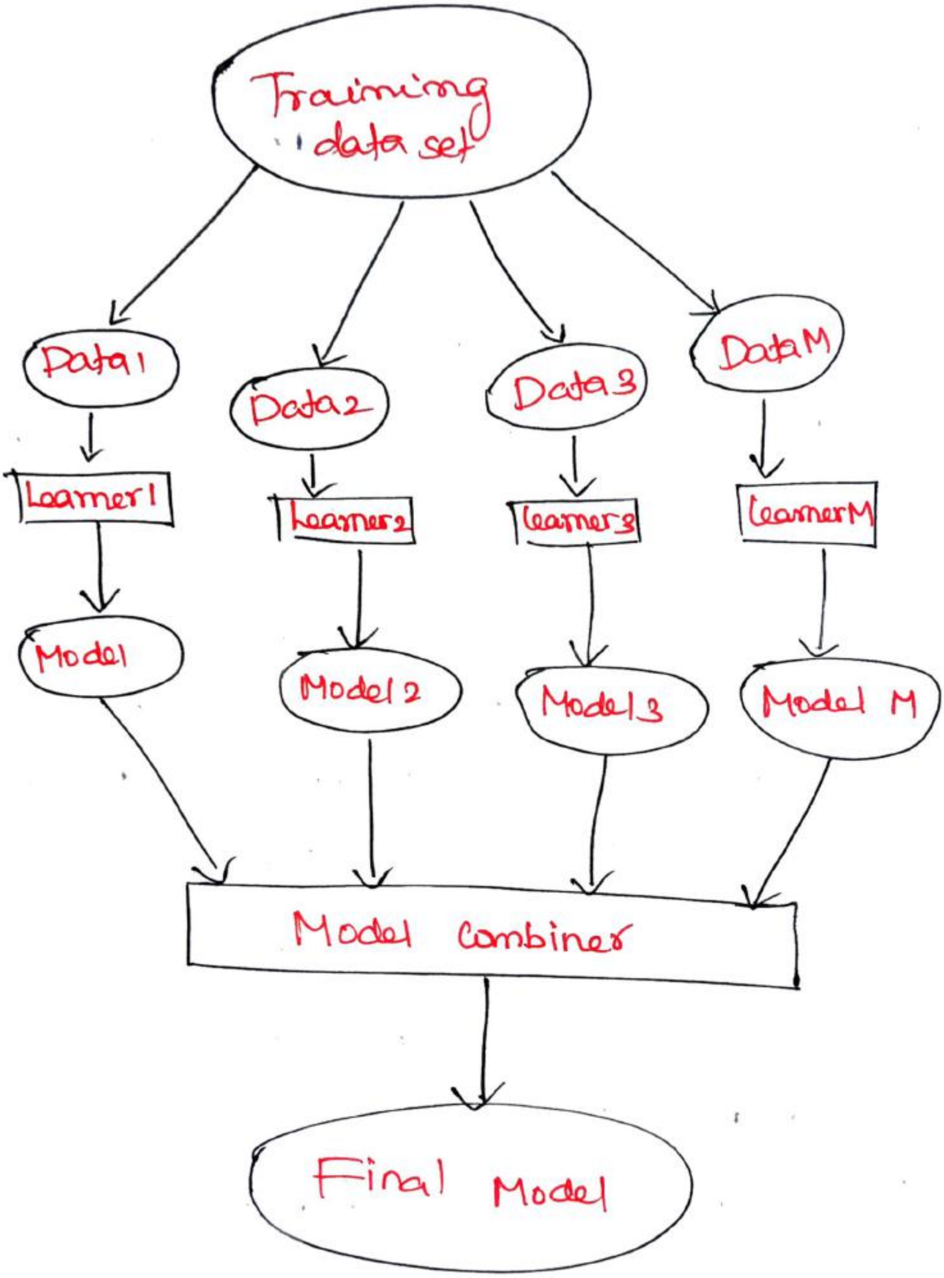
* For regression, a voting ensemble involves making a prediction that is the average of multiple other regression models.

* In classification a hard voting ensembles involves summing the votes for crisp class labels from other models and predicting the class with the most votes..

* A soft voting ensemble involves summing the predicted probabilities for class stables and predicting the class label with the largest sum probability

In this methods, the first step is to create multiple classification/regression models using some training dataset.

Each base model can be created using different splits of the same training dataset and same algorithm or using the same dataset with different algorithms or any other method.



When combining multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out and corrects decisions are reinforced. Human ensembles are demonstrably better

Use a single, arbitrary learning algorithm but manipulate training data to make it learn multiple models.

Error - correcting Output Codes

In error correcting Output Codes main classification task is defined in terms of a number of subtasks that are implemented by all base learners

The idea is that the original task of Separating one class from all other classes may be a difficult problem.

So we want to define a set of simpler Classification problems, each specializing in one aspect of the task and combining these simpler classification we get final classifier.

Base learners are binary classifiers having output -1 +1 and there is a code matrix W of K×L whose

K rows are the binary codes of classes in terms of the L base-learners dj.

Ensemble Learning

* The idea of ensemble learning is to employ multiple learners and combine their predictions.

* If we have a committee of M models with uncorrelated errors, simply by averaging them the average error of a model can be reduced by a factor of M

* Unfortunately, the key assumption that the errors due to the individual models are uncorrelated is unrealistic, in practice, the errors are typically high correlated, so the reduction in overall error is generally small

* Ensemble modelling is the process of running two or more related but different analytical models and then Synthesizing the results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications

* Ensembles of classifiers is a set of Classifiers whose individual decisions combined in some way to classify new examples.

* Ensemble methods combine several decision tree classifiers to produce better predictive performance than a single decision tree classifier.

* The main principle behind the ensemble model is that a group of weak learners come together to from a strong learner thus increasing the accuracy of the model

ENSEMBLE METHODS WORKING

VARIANCE REDUCTION:

If the training sets are completely independent, it will always help to average an ensemble because this will reduce variance without affecting bias (eg, bagging) and reduce sensitivity to individual data points.

BIAS REDUCTION:

For simple methods, average of models has much greater capacity than single model Averaging models can reduce bias substantially by increasing capacity and control variance by Citting one component at a time

BAGGING :

* Bagging is also called Bootstrap aggregating. Bagging and boosting are meta - algorithms that pool decision from multiple classifiers.

* It creates ensembles by repeatedly randomly resampling The training data

* Bagging was the first effective method of ensemble learning and is one of The Simplest method of arching.

* The meta - algorithm which is a Special case of the model averaging, was originally designed for classification and usually applied todecision tree models, but it can be used with any type of model for classification or regression.

* Ensemble classifiers such as bagging, boosting and model averaging are know to have improved accuracy and robustness over a single model

* Although unsuperised model, such as clustering do not directly generate label prediction for each individual they provide useful constraints for the joint prediction of a set of related Objects

* For a given training set of size n Create m samples of size n by drawing n examples from the original data with replacement

* Each bootstrap sample will on average contain 63.2% of the unique training examples, the rest are replicates.

* It combines The m resulting models using simple majority vote.

* In particular on each round, The base learner is trained on what is often called a bootstrap replicate of the original data set.

* Suppose training set consists of n examples.

* Then a bootstrap replicate is a new training set that also consists of n examples and which is formed by repeatedly selecting uniformly at random and with replacement n examples from the original training set.

* This means that the same example may appear multiple times in the bootstrap replicate or it may appear not at all.

* It also decreases error by decreasing the variances in the result due to unstable learner algorithms (like decision trees) whose output can change dramatically when the training data is slightly changed.

Bagging Steps:

* Suppose there are N observations and M features in training dataset

* A sample from training data set is taken randomly with replacement.

* A Subset of M features is Selected randomly and whichever feature gives the best split is used to split the node iteratively.

* The tree is grown to the largest

Above steps are repeated n times and prediction is given based on the aggregation of predictions from n number of trees.

Advantages of Bagging:

* Reduces over fitting of the model.
* Handles higher dimensionality data very well
* Maintains accuracy of missing data

Disadvantages of Bagging

* Since final prediction is based on the mean predictions from subset

BOOSTING

* Boosting is a very different method to generate multiple predictions (function and estimates) combine them linearly.

* Boosting refers to a general and provably effective method of producing a very accurate classifier by combining rough and moderately inaccurate rules of thumb

* Orginally developed by Computational learning theorists to guarantee performance improvements on fitting training data for a weak learner that only needs to generate a hypothesis with a training accuracy greater than 0.5

* Final result is the weighted sum of the results of weak classifier.

* A learner is weak if it produces a classifier that is only slightly better than random guessing, while a learner

is said to be strong if produces a classifier that acheives a low error with high confidence for a given concept.

* Revised to be a practical Algorithm Adaboost for building ensembles that empirically improves generalization performance. Examples are given weight at each iteration a new hypothesis is learned and the examples are reweighted to focus the system

* Boosting is a bias reduction technique It typically improves the performance of a single tree model.

* A reason for this is that we often cannot construct trees which are Sufficienty large due to thining out of observations in the terminal nodes.

* Boosting is then a device to come up with a more complex solution by taking linear combination of trees.

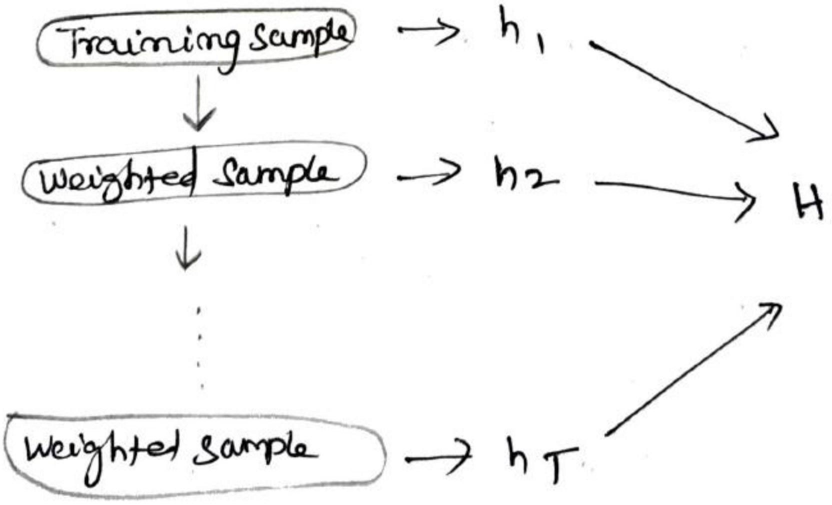
* In presence of high dimensional predictors boosting is also very useful as a regularization technique for additive or interaction modeling

* To begin we define an algorithm for finding the rules of thumb, which we call a weak learner.

* The boosting algorithm repeatedly calls This weak learner, each time feeding it a different distribution over the training data.

* Each call generates a weak classifier and we must combine all of these into a single classifier than hopefully is much more accurate than any one of the rules.

* Training a set of weak hypothesis $h_1 \cdots h_T$. The Combined hypothesis H is a weighted majority vote of the T weak hypotheses. During the training focus on the examples that are misclassified



Ada Boost:

* Ada Boost short for "Adaptive Boosting", is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire who won the prestigious "Godel prize" in 2003 for their work.

* It can be used in Conjonction with many other types of learning algorithms to impore their performance

* It can be used to learn weak Classifiers and final classification based on weighted vote of weak Classifiers.

* It is linear classifier with all its desirable properties. It has good generalization properties

* To use the weak learner to from a highly accurate prediction rule by calling the weak learner repeatedly on different distributions over the training examples

Initially all weights are set equally but each round the weights of incorrectly classified examples are increased so that those observations that the previously classifier poorly eta predicts receive greater weight on the next Iteration.

Advantages of AdaBoost

- * Very Simple to implement
- * Fairy good generalization
- * The prior error need not be known ahead of time.

Disadventages of AdaBoost

- * Suboptimal Solution
- * Can over fit in presence of noise

Boosting Steps:

* Draw a random subset of training samples d_1 without replacement from the training set D to train a weak learner C_1

* Draw second random training subset d_2 without replacement from the training set and add 50 percent of the samples that were previously falsely classified / misclassified to train a weak learner C_2

* Find the training samples d_3 in the training set D on which C_1 and C_2 disagree to train a third weak learner C_3

* Combine all the weak learners via majority voting

Advantage of Boosting:

* Supports different Loss function
* Works well with interactions

Disadvantages of Boosting:

* prone to over fitting
* Requires careful tuning of different hyper-parameters

STACKING:

* Stacking Sometimes called stacked generalization is an ensemble machine learning method that combines multiple heterogeneous base or Component models via a meta model

* The base model is trained on the complete training data and then the meta-model is trained on the predictions of the base models.

* The advantages of stacking is the ability to explore the solution space with different models in the same problem

* The stacking based model can be visualized in levels and has at least two levels of the models.

* The first level typically trains the two or more base learners (can be heterogenous) and the Second level might be single meta learner that utilizes the base models predictions as input and gives the final result as output

A stacked model can have more than more than two such levels but increasing the levels does not always guarantee better performance.

Stacking is concerned with multiple Classifiers generated by different learning algorithms $L_1, \ldots L_N$ on a single dataset S, Which is composed by a feature vector

$S_i = (x_i, t_i)$

The stacking process can be broken into two phases :

Generate a set of base - level classifiers $C_1 \ldots C_N$ Where $C_i = L_i(S)$

Train a meta - level classifier to Combine the output of the base-level classifiers

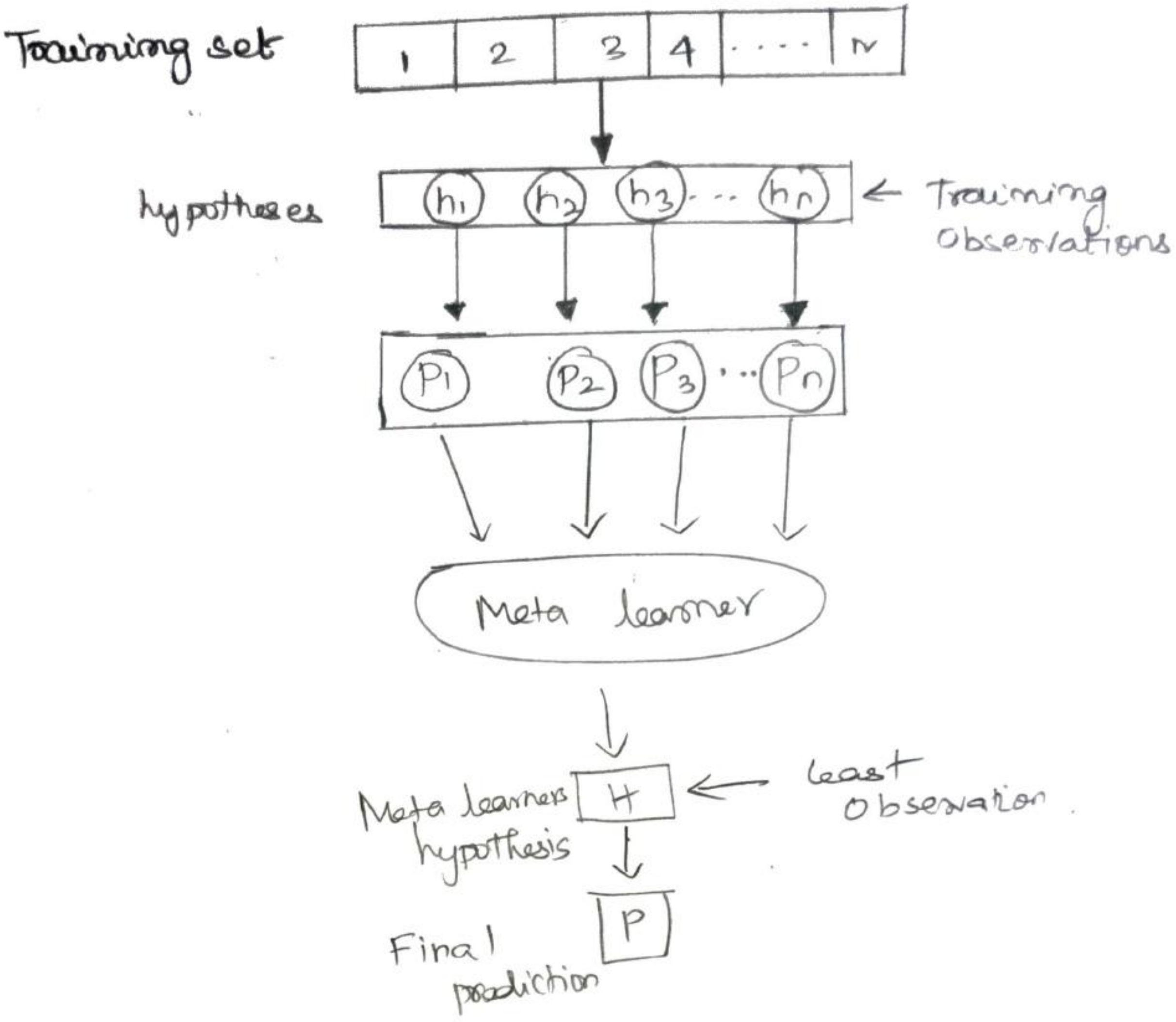


fig: Stacking frame

Based on two basic Observations.

Variance reduction:

If The training sets are Completely independent it will always help to average an ensemble because This will reduction

ADABOOST ALGORITHM:

* Ada Boost algorithm short for adaptive algorithm. It is a Boosting technique used as an ensemble method in machine learning.

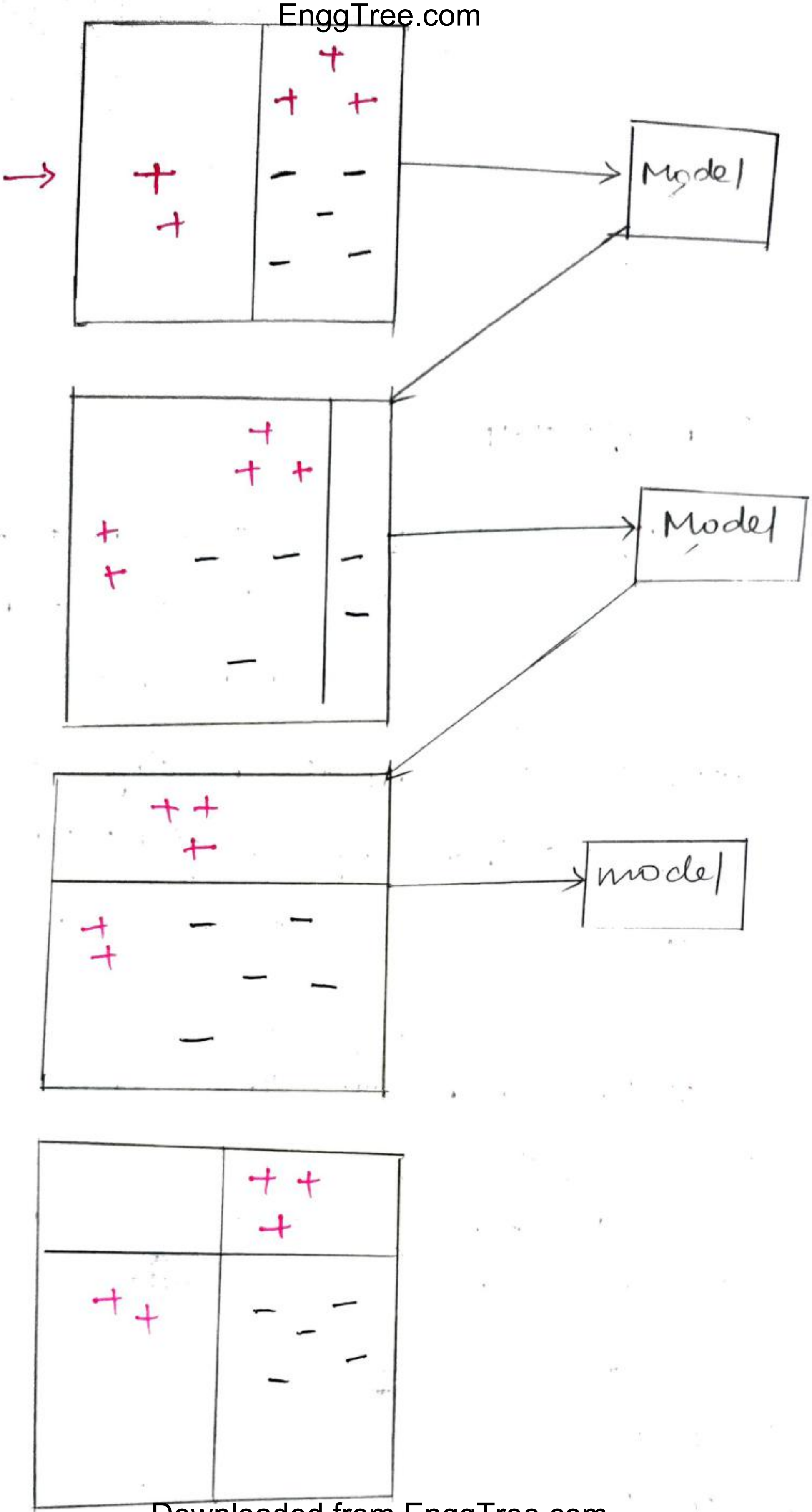
* It is called adpative boosting as the weights are reassigned to incorrectly classified instances

* Boosting is used to reduce bias as well as variance for supervised learning.

* It works on the principle of learners growing sequentially

Except for the first each subsequent learner is grown previously grown learners.

In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the same principle as boosting with a slight difference in detail


Model
Model
model

*The maximul not unsual algorithm used with AdaBoost is selection trees with one stage meaning with decision trees with most effective one split.

*These trees are also referred to as decision stopms

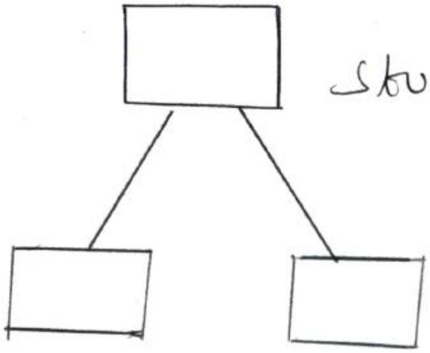
The working of Ada boost version follows the beneath referred to as decision stumps or path :

- * Creation of the base learner
- * Calculation of the total error via the beneath formulation
- * Calculation of performance of the decision stumps
- * Updating the weights in line with the misclassified factors.

Creation of new database :

AdaBoost ensemble :

In the ensemble approach we upload the susceptible fashion sequentially and then teach them the use of weighted schooling records.



We hold to iterate the process till we gain the advent of a pre-set range of vulnerable learners or we can not look at further improvement at the data set

At the end of Algorithm we are left with some vulnerable learners with a stage fee.

Difference between bagging and boosting

Bagging	Boosting
It is a technique that Builds multiple homogeneous models from different Subsamples of the Same training dataset to obtain more accurate predictions	It refers to a group of algorithms that utilize weighted averages to make weak learning algorithms stronger learning algorithms

* It helps in reducing Variance	* It helps in reducing bias and variance
* Every model receives an equal weight	* Models are weighted by their performance

CLUSTERING

*Give an set of objects, place them in groop such that the objects in a groop are similar (or related) to one another and different form (or unrelated to) the objects in other groop

* Closter analysis can be a powerful data mining tool for any organization that needs to identify discrete groups of customers, sale transactions, or other types of behaviours and things.

* For example, insurance providers use cluster analysis to detect fraudulent claims and banks uced it for credit scoring

* Closter analysis uses mathematical tool models to discover group of similar customers based on the smallest variance among customers within

each group.

* Cluster is a group of objects that belong to the same class. In another words the similar object are grouped in one cluster and disimilar grouped in other cluster.

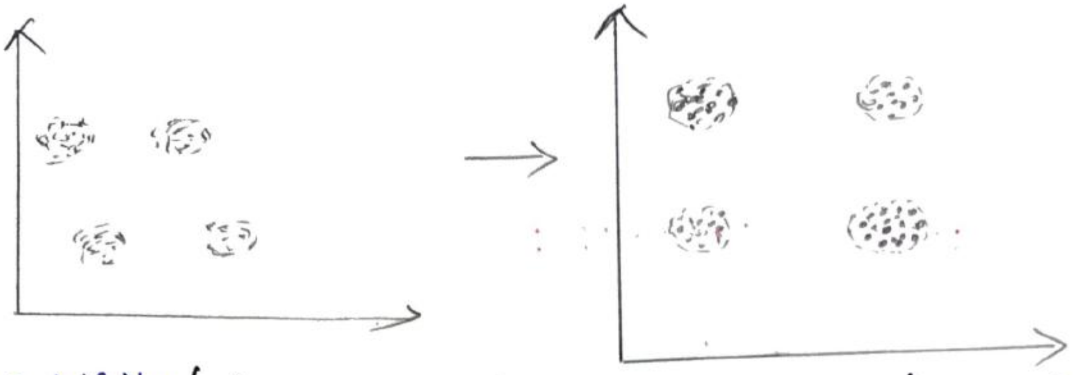
* clustering is a process of partitioning a set of data in a set of meaningful Subclasses.

* Every data in the sub class shares a common trait. It helps a user understand the natural grouping or structure in the data set.

* Various types of clustering methods are partitioning methods, hierarchical clustering, fuzzy clustering, density based clustering and model based clustering.

* Cluster analysis is process of grouping a set of data objects into clusters.

Desirable properties of a clustering algorithm are as follows



* Scalability (in terms of both time and space)

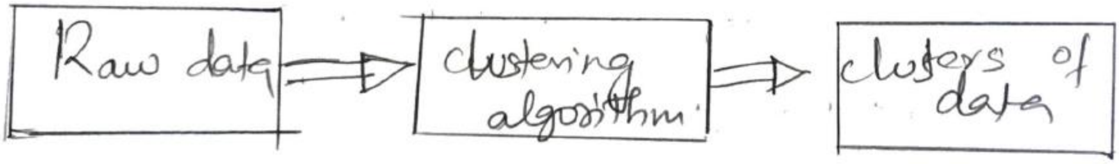
* Ability to deal with different data types

Minimal requirements for domain knowledge to determine input parameters

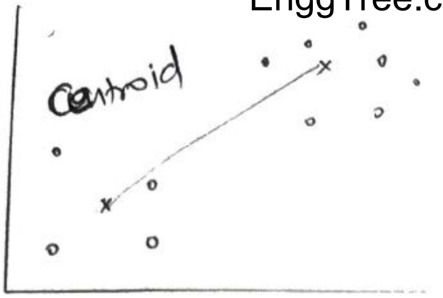
* Interpretability and usability

Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data.

* clustering can be considered the most important unsupervised learning problem.



* Clustering means grouping of data or dividing a large set into smaller data sets of some similarity



CLUSTER CENTROID:

* The centroid of a cluster is a point whose parameter values are the mean of the parameter values of all the points in the cluster.

* Each cluster has a well defined centroid

DISTANCE:

The distance between two point is taken as common metric to see as the similarity among the components of population

The commonly used distance measure is the euclidean metric which defines the distance between two points

$p=(p_1, p_2, \ldots)$ and $q=(q_1, q_2 \ldots)$ is given by

$$d = \sum_{i=1}^{b} (p_i - q_i)^2$$

* The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide What constitutes a good clustering?

* It can be shown that there is no absolute best criterion which should be independent of the final aim of clustering

clustering algorithm can be classified as listed below:

⇒ Exclusive clustering.

⇒ Overlapping Clustering

⇒ Hierarchical Clustering

⇒ probabilistic clustering

A good clustering method will produce high quality clusters intra-class Similarity and low intra class similarity

The Major clustering techniques are

* partitioning methods
* Hierarchical methods
* Density Method

UNSUPERVISED K-MEANS CLUSTERING

* K - Means clustering is heuristic method. Here each cluster is represented by the center of the Cluster.

* K stands for number of clusters, It is typically a user input to the algorithm some Criteria can be used automatically estimate K.

* This method initially takes the number of components of the population equal to the final required number of clusters

* In this step itself the final required number of cluster is chosen such that the points are mutually farthest apart.

* Given k-means algorithm consists of four steps :

⇒ Select intial centroids at random

⇒ Assign each object to the cluster with nearest centroid.

⇒ Compute each centroid as the mean of the objects assigned to it.

⇒ Repeats previous 2 steps until no change

* The $x_1, \ldots x_N$ are data points or vector of observations

Each Observation (vector x_i) will be assigned to one and only one cluster. The C_i denotes cluster number for the ith Observation. K-means minimizes within-cluster point scatter

$$W(C) = \frac{1}{2} \sum_{K=1}^{K} \sum_{C(i)=K} \sum_{C(j)=K} \| x_i - x_j \|^2$$

$$= \sum_{K=1}^{K} N_K \sum_{C(i)K}^{K} \| x_i - m_K \|^2$$

Where

m_K is the mean vector of the K^{Th} cluster

N_K is the number of observations in K^{Th} cluster.

K-Means Algorithm properties

* There are always K cluster

* There is always at least one item in each cluster

* The clusters are non hierarchical and they do not overlap

Every member of cluster is closer to

its cluster than any other cluster

because closeness does not always involve the center of clusters

K-Means Algorithm

1) The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters that have roughly the same number of data points.

2) For each data point

*calculate the distance from the data point to each cluster.

*If the data point is closest to its own cluster, leave it where it is

*If the data point is not closest to its own cluster, move it into the closest clusters

Repeat the above step untill a complete pass through all data points result in no data point moving from one cluster to another

* K Means algorithm is iterative in natune, It converages however only a local minimum is obtained. It works only for numerical data. This method is easy to implement

Advantages of K-Means Algorithm.

- *Efficient in computation
- *Easy to Implement

Weakness

- *Applicable only when mean is defined
- *Need to Specify K the number of clusters in advanced.
- *Trouble with noisy data & outiliers
- *Not suitable to discover clusters with non convex shapes

KNN :

K-nearest Neighbour is one of the Machine learning algorithms based totally on Supervised learning approach.

K-NN algorithm assumes the similarity between the brand new case/facts and available instances and placed the brand new case into the category that is maximum similar to the to be had classes

KNN set of rules shops all of the be had facts and classifies a new statistics point based at the similarity.

This means when new data seems then it may be effortlessly categorized into a properly suite class by using K-NN algorithm

K-NN set of rules can be used for regression as well as for classification however normally its miles used for the classification troubles

KNN is a non parametric algorithm because of this it does no longer makes any assumption on underlying data

It is also refered to as a lazy learner set of rules because it does not longer research

research from the training set immediately as a substitute it shops the dataset and at the time of class it plays an movement at the dataset

The KNN set of roles at the Schooling section simply stones the dataset and when it gets new data then it classifies that statistics into a class that is an awful lot similar to the brand new data.

Example :

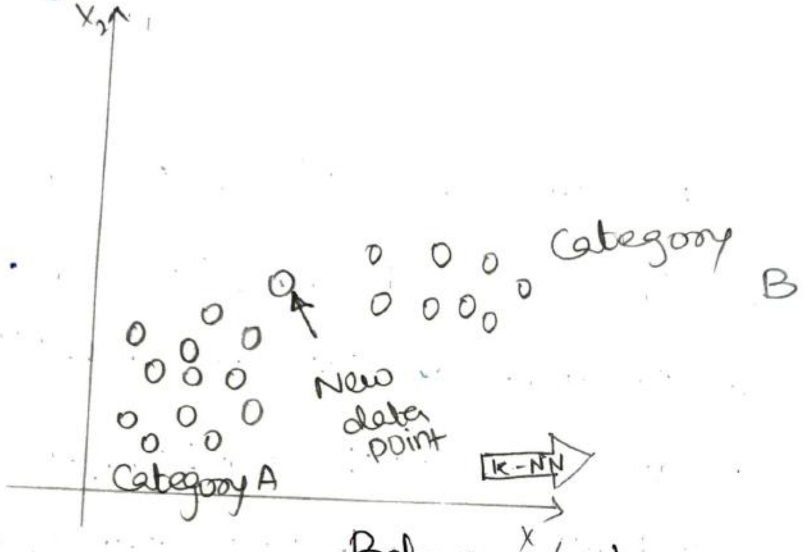
Suppose we have an picture of creature that looks much like cat and dog but we want both it is a cat or dog. So far this identity we are able to use the KNN algoritom, because it works on a similarity degree. Our KNN version will discover the similar features of the new facts set to the cats and dogs snap shots and primarily based on the most similar functions it will place it in both cat or canine class.

Why do we need KNN ?

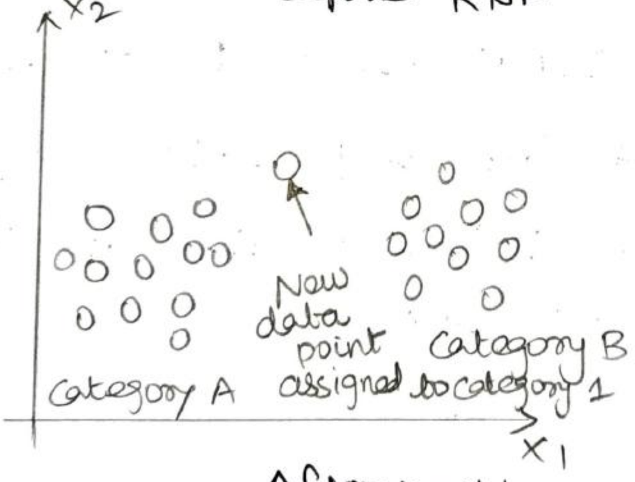
Suppose there are two categories i.e category A and category B and we have a brand new statistics point x and so this fact point will [illegible] of these classes.

To solve this sort of problem we need a K-NN set of roles.

With the help of K-NN we will without difficulty discover the category or class of a selected dataset Consider the underneath diagram.



Before K-NN



After K-NN

KNN working :

The KNN working can be explained on the basis of the below algorithm.

1 ⇒ select the wide variety K of the acquaintances

2 ⇒ Calculate the Euclidean distance of K variety of friends.

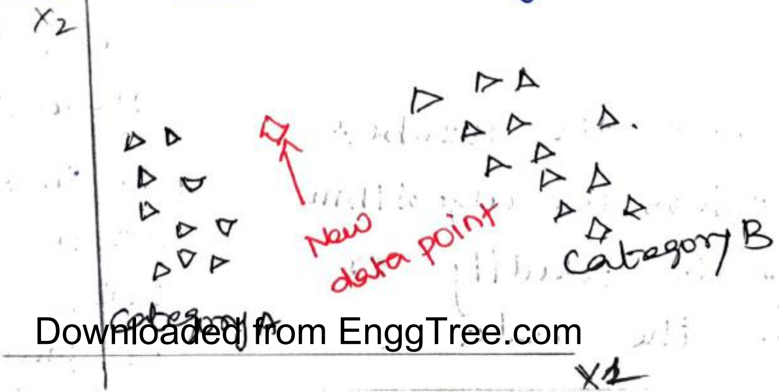
3 ⇒ Take the K nearest neighbors as according to the calalated Euclidean distance

4 ⇒ Among these ok pals, cont number of the data points in each class.

5 ⇒ Assign the brand new record points to that category for which quatity of the neighbor is maximum.

6 ⇒ Our model is ready.

Suppose we have got a brand new information point and we want to place it in the required Category. Consider the under image.



Firstly we are able to pick the number of friends so we are able to select the oK=5.

Next we will claculate the Euclidean distance between the fact points. The Euclidean distance is the gap between points which we have got already studied in geometry. It may be calculated as

$$\text{Euclidean distance} = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}.$$

Difference between K means and KNN

K Means	KNN
* K Means is an unsupervised machine learning algorithm used for clustering.	* KNN is a Supervised machine learning algorithm used for Classification.
* K-means is an eager learner	* KNN is a lazy learner.
* It is used for clustering	* It is used for classification and Sometimes even for regression.
* K-Means is the number of clusters the algorithm is try to identify or learn the	* K in KNN is the number of the nearest neighbour used to classify or predict a test Sample

* K Means require unlabelled data. It gathers and groups data into K number of clusters

* KNN require labelled data and will give new data points accordingly to the K number or the closest data points.

Gaussian Mixture Models:

* Gauissan Mixture models is a soft clustering algorithm where each point probabilistically belong to all clusters. This is different than K means where each point belong to one clusters.

* The gauissan mixture model is a probabilistic model that assumes all the data points are generated from a mix of guassian distributions with unknown parameters.

* Gauissan mixture models consists of two parts: Mean vectors and Covariance matrices.

* A gaussian distribution is defined as a continuous probability distribution that takes a bell shaped curve. Another name of the gaussian distribution is the normal distribution

In one dimensional space the probability density function of a gaussian distribution is given by

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

Where μ is the mean and σ^2 is the variance.

* Gaussian mixture models can be used for a variety of use cases, including identifying customer segments detecting fraudulent activity and clustering images.

* GMM have Variety of real world applications They are

* Used for Signal processing
* Used for customer Churn analysis
* Used for language identification
* Used in video game industry
* Genre classification of Songs

Expectation. Maximization

* In Gaussian mixture models an expectation maximization method is a powerful tool for estimating the parameters of Gaussian mixture model. This expectation is termed as E and maximization is termed M

and maximizing (M) step which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found in the E step.

* In the Expectation step, find the expected values of the latent variables (here you need to use the current parameter values)

* In the Maximization step first plug in the expected values of the latent variables in the log-likelihood of the augmented data. Then maximize this log-likelihood to reevaluvate the parameters.

* EM is a technique used in point estimation Given a set of observable variables x and unknow (latent) variables z we want to estimate parameters θ in a model. It is a widely used maximization likeli-hood estimation procedure for the statistical models when the values of some of the variables in the model are not observed.

* In E step, the algorithm estimates the posterior distribution of the hidden variables Q given the observed datas and the current parameter settings and in the M step algorithm calculates

* Expectation is used to find the gaussian parameters which are used to represent each component of gaussian mixture models. Maximization is termed M and it is involved in determining whether new data points can be added or not.

* This algorithm used in maximum likelihood estimation where the problem involves two sets of random variables of which one x is observable and the other z is hidden.

* The goal of the algorithm is to find the parameter vector ϕ that maximizes the likelihood of the observed values of x $L(\phi|x)$

EM Algorithm :

* It is an interative method used to find maximum likelihood estimates of parameters in ~~probab~~ probabilistic models where the models depends on unobserved also called as latent variables

* EM alternate between performing an expectation (E) step which computes an expectation of the likelihood by including the latent variables as if they were observed

*The ML parameter settings with q fixed

*At the end of each iteration the lower bond on the likelihood is optimized for the given parameter setting (M-step) and the likelihood function is set to that bond E step.

*Generally EM works best When the fraction of missing information is Small and the dimensionality of the data is not too large.

*EM Require many iterations and higher dimensionality can dramatically slow down The E Step.

EM is usefull for several reasons:

- * conceptual simplicity
- * ease of implementation

Sometimes the M-step is a constrained maximization which means that there are constraints on Valid solutions not encoded in the function itself

*Expectation maximization is an effective technique that is often used in data analysis to manage missing data. Indeed expectation maximization overcomes some of the limitations of other techniques, such as mean Substitution or regression Substitution.

* The alternative techniques generate biased estimates and specifically underestimate the standarad errors. Expectation maximization overcome this problem.

UNIT - IV

NUERAL NETWORKS

PRECEPTRON :

* The perceptron is a feed-forward network with one output neuron that learns a Separating hyperplane in a pattern Space

* The "n" linear Fx neurons feed forward to one threshold output Fy neuron. The perceptron Separates linearly separable set of patterns

Single layer preceptron :

* The preceptron is a feed forward network with one output neuron that learns a Separating hyper plane in a pattern space

* The "n" linear Fx neurons feed forward to one threshold output Fy neuron

* The preceptron separates linearly separable set of patterns

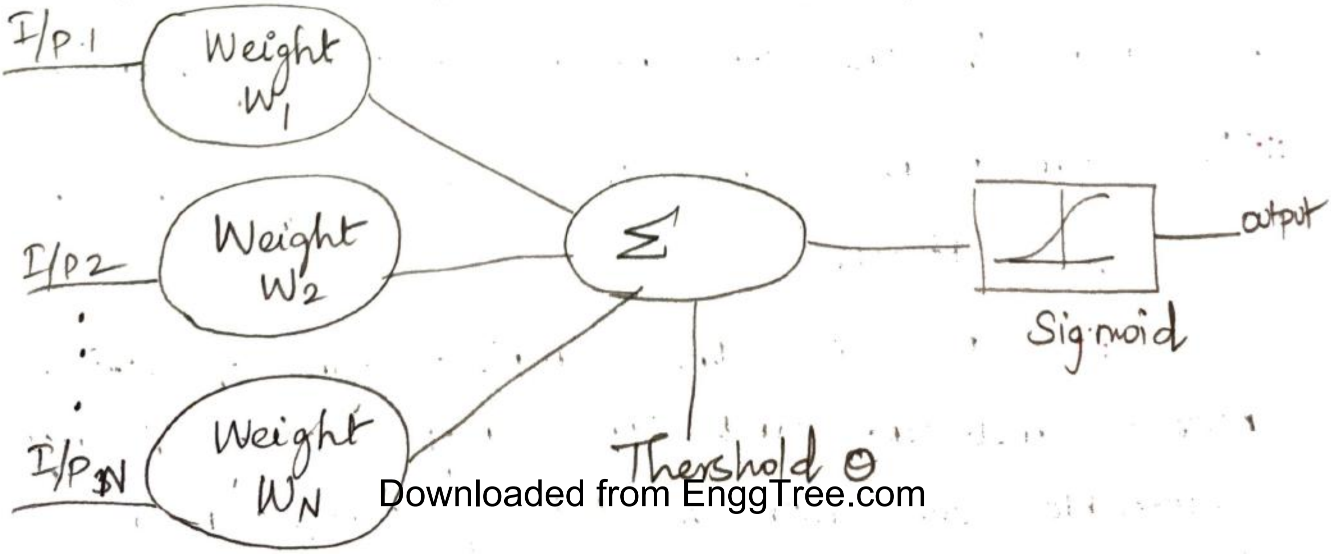
* SLP is the simplest type of artifical neural networks and can only classify linearly Separable cases with a binary target(1,0)

* We can connect any number of McCulloh pitts neurons together in any way we like. An arrangem of one input layer of McCulloch-pitts neurons feeding forward to one ~~out~~put layer of McCulloch-pitts neurons feeding is known as perceptron.

* An Single layer feed-forward network consists of one or more output neurons each of which connected with weighting factor Wij to all of the inputs Xi.

* The perceptron is a kind of single layer artificial network with only one neuron.

* The percepton is a network in which the neuron unit calculates the linear combination of its real valued or boolean inputs and passes it through a threshold activation function.



* The perceptron is Sometimes referred to a Threshold Logic Unit (TLU) since it discriminates the data depending on whether the sum is greater than the Threshold Value.

* In the Simplest case the network has only two inputs and single output. The output of neuron is

$$y = f\left(\sum_{i=1}^{2} w_i x_i + b\right)$$

* In single layer preceptron intial weights Value are assigned radonmly because it does not have previous knowledge.

* It sum all the weighted inputs. If the sum is greather than the Threshold Value then it is activated i.e output = 1.

* If the output does not match the desired output, then the weights need to be changed to reduce the error.

The weight adjusment is done as follows:

$$\Delta w = \eta \times d \times x$$

Where x = input data

d = predicted output and desired output.

η = Learning rate.

* If the output of the perception is correct then we do not take any action.

* If the output is incorrect then the weight vector is $W \rightarrow W + \Delta W$

The process of weight adaptation is called learning.

perceptron learning Algorithm:

1) Select random sample from training set as input

2) If the classification is correct do nothing.

3) If classification is incorrect modify the weight vector W using.

$$W_i = W_i + \eta d(n) x_i(n)$$

4) Repeat the procedure until the entire training set is classified correctly

Multilayer perceptron:

* A Multi layer perceptron (MLP) has the same structure of a single layer perceptron with one or more hidden layers.

* An MLP is a network of simple neurons called perceptrons.

* A typically multilayer network consists of a set of source nodes forming the input layer one or more hidden layers of computational nodes and an output layer of nodes.

* It is not possible to find weights which enables single layer perceptron to deal with non linearly separable problems like XOR

Limitations of learning in perceptron: Linear Separability

* Consider two input patterns (x_1, x_2) being classified into two classes.

* Each point with either symbol of x or o represents a pattern with set of values (x_1, x_2).

* Each pattern is classified into two classess Notice that these classes can be separated with a single line L. They are known as linearly separable patterns.

* Linear separability refers to the fact that classes of patterns with n-dimensional vector $x = (x_1, x_2 \ldots x_n)$ can be separated with a single decision surface.

* If two classes of patterns can be separated by a decision boundary, represented by the linear equation then they are said to be linearly separable. The simple network can correctly classify any patterns

* Decision boundary of linearly separable classes can be determined either by some learning procedures or by solving linear equation systems based on representative pattern of each classes.

* If such decision boundary does not exist then the two classess are said to be linearly inseparable.

Linearly inseparable problems cannot be solved by the simple network, more sophisticated architecture is needed.

Activation Functions:

Activation function also known as transfer functions is used to map input nodes to output nodes in certain fashion.

The activation function is the most important factor in a neural network which decoded whether or not a neuron will be activated or not and transferred to the nex layer

Activation function helps in normalizeing the output between 0 to 1 or -1 to 1. It helps in the process of backpropagation due to their differentiable property.

During backpropagation loss function gets updated and activation functions helps the gradient decesent curves to achaive their local minima.

It basically decides in any neural network that given input or receiving information is relevant or it is irrelevant

These activation function makes the multilayer network to have greater representational power than a single layer network only when non-linearity is introduced

STOCHASTIC GRADIENT DESCENT :

* It means a System or process linked with a random probability. In stochastic gradient descent few samples are selected randomly instead of whole data set for each iteration.

* In gradient Descent there is a term called "batch" which denotes the total number of Samples from a dataset that is used for calculating the gradient for each iteration.

* In typical gradient optimization, the batch is taken to whole dataset.

* Although using the whole dataset is really useful for getting to the minima in a less noisy and less random manner. The problem arises when your dataset is big

If you have a million Samples in your dataset, so if you use a typical Gradient Descent optimization technique, you will have to use all of the one million samples for Completing one iteration, while performing the gradient Descent and it has to be done for every iteration until the minima are reached.

It becomes Computationally very expensive to perform. This problem is solved by Stochastic Gradient Descent. It uses only a single sample.

The sample is randomly Suffled and Selected for performing the Iteration.

SGD is a variant of gradient Descent algorithm used for optimizing machine learning models.

Advantages:

It is faster than other Variants of Gradient descent and it uses only one example to update the parameters.

Since SGD updates it has the ability to escape from local minima and go converage to global minima

Due to noisy updates in SGD it has ability to escape from local minima and converge to a global minimum.

Disadvantages:

The updates in SGD are noisy and have high variance, which can make the optimization process less stable and lead to oscillations around the minimum.

ERROR BACKPROPAGATION

Backpropagation is a training method used for a multi layer neural network.

It is called the generalize delta rule

It is a gradient descent method which minimizes the total squared error of the output computed by the net.

The Backpropagation algorithm looks for the minimum Value of the error function in weight space using a technique called the delta rule or gradient descent.

The weights that minimize the error function that is considered to be a Solution to the learning problem.

Backpropagation is a Systematic method for training multiple layer ANN. It is a generalization of Widrow-Hoff error correction rule. 80%. of ANN Applications uses backpropagation.

Consider a simple neural network

⇒ Nueron has Summing junction and activation junction.

⇒ Any non linear function which differentiable everywhere and increases everywhere with sum can be used as activation function.

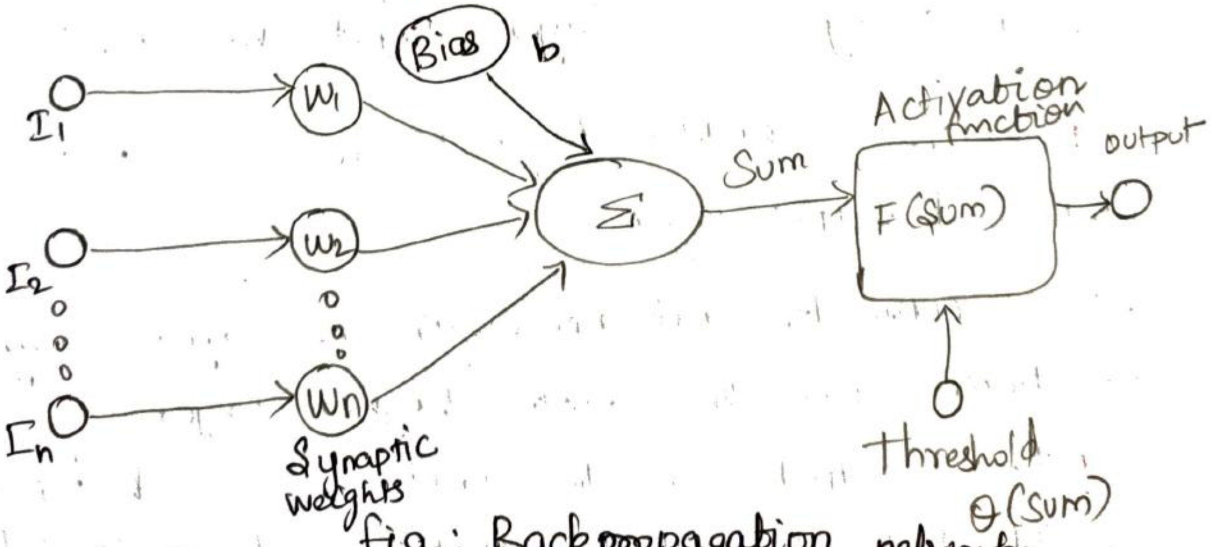


fig: Backpropagation network

Examples: logistic function, arc tangent function, Hyperbolic tangent activation function.

* These activation function makes the multilayer network to have greater representational power than single layer network only when non-linearity is introduced.

Need of hidden layers:

* A network with only two layers (input and output) can only represent the input with whatever representation already exists in the input data

* If the data is discontinuous or non linearly separable, the innate representation is inconsistent and the mapping cannot to be learned using two layers (Input and output)

* Therefore hidden layer(s) are used between input and output layer

Weights connects unit (neuron) in one layer only to those in the next higher layer.

* The output of the unit is scaled by the value of the connecting weight and it is fed forward to provide a portion of the activation for the units in the nex higher layer

Backpropagation can be applied to an artificial neural network with any number of hidden layers.

The training objective is to adjust the weights so that the application of a set of inputs produces the desired outputs.

Training procedure:

The network is usally trained with a large number of input - output pairs

* Generate weights randomly to small random values (both positive and negative) to ensure that the network is not saturated by large values of weights

* choose a training pair from the training set.

* Apply the input vector to network input calculate the network output :

* Calculate the error, the difference between the network output and the desired output

* Adjust the weights of the network in a way that minimizes this error.

* Repeat steps 2-6 for each pair of input - output in the training set until the error for the entire System is acceptably low.

Forward pass and backward pass:

Backpropagation neural network training involves two passes

In the forward pass, the input signals moves forward from the network input to the output.

In backward pass, the calculated error signal propogate backward through the network where they are used to adjust the weights.

In the forward pass, the calculation of output is carried out, layer by layer in the forward direction.

The output of the one layer is the input to the next layer

In reverse pass

The weights of the output neuron layer are adjusted first since the target value of each output neuron is available to guide the adjusment of the associated weights using the delta rule.

Next we adjust the weights of the middle layers. As the middle layer neurons have no target values, it makes the problem complex.

Selection of number of hidden units:

* The numbers of hidden unit depends on the number of input units.

* Never choose h to be more than twice the number of input units.

* You can load p patterns of I elements into $\log_2 p$ hidden units.

* Ensure that we must have at least 1/e times as many training examples.

* Feature extraction requires fewer hidden units than inputs.

* Learning many examples of disjointed inputs requires more hidden units than input.

* The number of hidden units required for a classification task increases with the number of classes in the task.

* Large network require longer training times.

FACTORS INFLUENCING BACK PROPAGATION TRAINING

Bias: Networks with biases can represent relationship between inputs and outputs more easily than networks without biases.

Adding a bias to each neuron is usually desirable to offset the origin of the activation function.

The weight of the bias is trainable similar to weight expect that input is always +1.

Momentom:

The use of momentum enhances the stability of the training process. Momentum is used to keep the training process going in the same general direction analogous to the way that momentum of moving object behaves.

In propagation with momentum, the weight change is a combination of the current gradient and the previous gradient.

Advantages:

It is simple, fast and easy to program.

It is flexible.

A standard approach & works efficiently

No need to have prior knowledge about the network.

Disadvanteges:

It possibly be sensitive to noisy data & irregularity

The performance of [illegible] highly reliant on the input data.

Need excessive time for training

The need for a matrix-based method for back propagation instead of mini batch.

Shallow Networks :

The terms shallow and deep refer to the number of layers in the neural network that have small numbers of layers usally regraded as having a single hidden layer and deep neural networks refer to neural networks that have multiple hidden layer.

Both types of networks perform certain tasks better than The other and selecting the right network depth is important for creating a successful model.

In a shallow neural network, the Values of the feature vector of the data to be classified (the input layer) are passed to hidden layer of nodes (neurons) each of which generates a response according to some activation function g, acting on the weighted sum of those Values.

The response of each unit in the hidden layer is then passed to a final output layer (Which may consist of a single unit) Whose activation produces the classification prediction output.

Deep Network:

Deep learning is a new area of machine learning research which has been introduced with the objective of moving machine learning closer to one of its original goals

Deep learning is about learning multiple levels of representation and abstraction that help to make sense of data such as image sound and text.

It using a neural network with several layer of nodes between input and output.

It is generally better than other methods on image speech and certain other data because the series of layers between input and output do feature identification and processing in series of stages just as brain seems ~~to~~

Deep learning emphaszies the network architecture of todays most successful machine learning approaches

These method based on "deep" multi layer neural networks with many hidden layers.

TENSOR FLOW:

Tensorflow is one of the most popular frameworks used to build deep learning models. this framework is developed by Google brain Team.

Languages like C++, R and python are supported by the framework to create the models as well as libraries. This framework can be accessed from both desktop and mobile.

The translator used by google is the best example of Tensorflow.

In this the model is created by adding the functionalies of text classification natural, language processing, Speech or handwriting recognition, image recognition ect.

The framework has its own visualization toolkit named TensorBoard which helps in powerful data visualizattion Of the network along with its performances.

One more tool added in TensorFlow Tensorflow serving can be used for quick and easy deployment of the newly developed algorithms without introducing any change in the existing API or architechure.

Tensor framework comes along with a detailed docomentation for the users to adapt it quickly and easily, making it the most preferred deep learning framework to model deep learning algorithms.

Some of the Characteristics of Tensorflow is

Multiple GPU supported

One can visualize graphs and queues easily using Tensorboard.

powerful documentation and large support from community

KERAS:

If you are comfortable with python then learning keras will not prove hard to you. This will be the most recommended framework to create deep learning models for ones having a Sound of python.

Keras is built purely on python and can run on the top of tensorflow. Due to this complexity and use of low-level libraries, Tensorflow can be comparitively harder to adapt for the new users as compared kera.

Users those who are beginners in deep learning and find its models difficult to understand in Tensorflow generally prefer keras as it Solves all complex models in no time.

Keras has been developed keeping in mind that complexities in deep learning models and hence it can run quickly to get the results in minimum time.

Convolutional as well as Recurrent Neural networks are Supported in Keras. The framework can easily run on CPU and GPU

It is classified into two Categories

Sequential mode :

The layers in deep learning model are defined in a Sequential manner. Hence the implementation of layer is this model will also be done Sequentially.

Keras Functional API :

Deep learning model That has multiple outputs or has Shared layers (i.e) more complex models can be implemented in keras functional API

Diff b/w Deep & Shallow Network.

Deep Network	Shallow network
Deep networks contain many hidden layers	It Contains one hidden layer

Deep Network	Shallow Network
* Deep Network contains high complex functions over input space	* Shallow networks with one hidden layer cannot place complex functions over the input space.
* Training is easy and no issue of local minima	* Shallow network is more difficult to train with our current algorithms.
* Deep network can fit functions better with less parameters than a shallow network	* Shallow net's needs more parameter to have better fit

VANISHING GRADIENT PROBLEM :

* It is a problem that users face when we are training Neural networks by using gradient based methods like propagation. This problem makes it difficult to learn the parameters of the earlier layers in network.

* The vanishing gradient problem is essentially a situation in which a deep multilayer feed forward network or a Recurrent Neural Network (RNN) does not have the ability to propagate useful

gradient information from the output end of the model back to the layers near the input end of the model.

*It results in model with many layers being rendered unable to learn on a specific dataset. It could even cause models with many layers to prematurely coverage to a substandarad solution.

*Vanishing gradient doesnot necessarily imply the gradient vector is all zero.

*It imples that the gradient are minuscule which would cause the learning to be very slow.

The most important solution to the vanishing gradient problem is a specific type of neural network called long short Term Memory Networks (LSTMS)

Indication of Vanishing gradient problem:

a) The parameters of high layers change to great extent, while the parameters of lower layers barely change.

b) The model weights could become 0 during training.

c) The models learns at a particularly slow pace and the training could stagnate the a very early phase after only a few iterations.

Some methods are proposed to overcome the vanishing gradient problem

a) Residual Neural Networks

b) Multilevel hierarchy

c) Long short term Memory

d) Faster hardware

e) ReLU

f) Batch Normalization.

ReLU:

Rectified linear unit (ReLU) slove the vanishing gradient problem. ReLU is a non linear function or piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

It will commonly used activation function in neural networks especially in Convolutional Neural Networks (CNNs) and multilayer perceptrons

Mathematically it is expressed as

$$f(x) = \max(0, x)$$

Where x : input to neuron.

The derivative of an activation function is required when updating the weights during backpropagation of the error.

The slope of ReLU is 1 for positive values and 0 for negative values. It becomes non differentiable when the input x is zero but it can be safely assumed to be zero and causes no problem in pratice.

ReLU is used in hidden layers instead of Sigmoid both or tanh.

The ReLU function solves the problem of Computational Complexity of the logistic Sigmoid & Tanh functions.

A ReLU activation unit is known to be less likely to create a vanishing gradient problem because its derivative is always 1 for positive values of the argument.

Advantages:

* RoLU is simple to compute and has a predictable gradient for the propagation of the error.

* Easy to implement and very fast

* The calulation speed is very fast. The ReLU function has only a direct relationship.

* It can be used for deepnetwork training.

Disadvantages:

When the input is negative, ReLU is not fully functional which means when it come to wrong number installed, ReLU will die. The problem is also known as the Dead Neurons problem

ReLU function can only be used within hidden layers of Neural networks model.

LReLU

The leaky ReLU is one of the most well known activation function.

It is the same as ReLU for positive numbers

But instead of being 0 for all negative Values it has a Constant slope.

Leaky ReLU is a type of activation function that helps to prevent the function from becoming saturated at zero.

It has a Small slope instead of the standarad ReLU which has an infinite slope.

Leaky ReLUs are one attempt to fix the "dying ReLU" problem.

EReLU

An Elastic ReLU (EReLU) consider a slope randomly drawn from a uniform distribution during the training for The positive inputs to control the amount of non linearity.

The ERELU is defined as:

ERELU(x) = max(Rx;0) in the output range of [0;1) where R is a random number.

At the least time, The ERELU becomes the identity function for positive inputs.

Hyperparameter Tuning:

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning.

While designing the machine learning model, one always has multiple choices for the architectural design for the model. This creates a confusion on which design to choose for the model based on its optimality. And due to this there are always trails for defining a perfect machine learning model.

The perrameters that are used to define these machine learning models are known as hyperparameter tuning.

Hyperparameters are not model parameters, which can be directly trained from data.

Model parameters usually specify the way to transform the input into the required output, whereas hyperparamaters define the actual structure of the model that gives the required data.

Layer size :

Layer size is defined by the number of neurons in a given layer.

Input and output layers are relatively easy to figure out because they correspond directly to how our modelling problems handles input and output.

For the input layer, this will match up to the number of features in the input vector. For the output layer this will either be a single output neuron or a number of neurons matching the number of classess we are trying to predict.

It is the obvious that a neural networks with three layers will give better performance than that of twolayers.

Increasing more than 3 doesnot help that much in neural networks.

In the case of CNN an increasing number of layers makes the model better

Magnitude: Learning rate

The amount that the weights are updated during training is referred to as the step size or learning rate.

Specifically the learning rate is a configurable hyper-parameter used in the training of neural networks that has small positive value, often in the range between 0.0 and 1.0.

For eg, if learning rate is 0.1, then the weights in the network are updated 0.1*(estimated weight error) or 10% of the estimated weight error each time the weights are updated. The learning rate hyperparameter controls the rate or speed at which the model learns

Normalization:

Normalization is a data preparation technique that is frequently used in machine learning.

The process of transforming the columns in a dataset to the same scale is referred to as normalization

Every dataset does not need to be normalized for machine learning.

Normalization makes the features more consistent with each other, which allows the model to predict output more accurately.

The main goal of normalization is to make the data homogeneous over all the records & fields.

It refers to rescaling real-valued numeric attributes into a 0 to 1 range.

Data normalization is used in machine learning to make model training less sensitive to the scale of features.

BATCH NORMALIZATION :

It is a method of adapative reparameterization motivated by the difficulty of training very deep models.

In deep networks, the weights are updated for each layer. So the output will no longer be on the same scale as the input.

When we input the data to a machine or a deep learning algorithm we tend to change the values to a balanced scale because we ensure that or model can generalize appropriately.

Batch normalization is a technique for standarizing the inputs to layers in the neural network.

It is a techique that was designed to address the problem of internal covariate shift, which arises as a consequence of updating multilayers inputs simultaneously in deep neural networks.

It is applied to indivual layers optionally to all of them. In each training iteration, we first normalize the inputs by subtracting their mean and dividing by their standarad deviation where both are estimated based on the statistics of the current mini-batch

We apply a scale co-efficient and an offset to recover the lost degrees of freedom.

It is precisely due to this normalization based on batch statistics that batch normalization derives its name.

We take the output $a^{[i-1]}$ from the preceding layer and multiply by the weights W and add the bias b of the current layer.

When applying batch norm, we correct our data before feeding it to the activation function

To apply batch norm, calculate the mean as well as the variance of current z

$$\mu = \frac{1}{m}\sum_{j=1}^{m} z_j$$

When calculating the variance we add a small constant to the variance to prevent potential division by zero.

$$\sigma^2 = \frac{1}{m}\sum_{j=1}^{m}(z_j-\mu)^2+\varepsilon$$

To normalize the data we substract the mean and divide the expression by the standarad deviation

$$z[i] = \frac{z[i]-\mu}{\sqrt{\sigma^2}}$$

This operation scales the inputs to have a mean of 0 and a standard deviation of 1.

Advantages:

The model is less delicate to hyperparameter tuning

Shrinks internal covariant shift.

Diminishes the reliance of gradients on the scale of the parameters or their underlying values.

Dropout can be evacuated for regularization.

Regularization:

Just have a look at the above figure and we can immediately predict that once we try to cover every minutest feature of the input data that can be irregularities in a extracted features, Which can introduced noise in the output. This is referred to as "overfitting".

This may also happens with the lesser number of features extracted as some of the important details might be missed out. This will leave an effect on the accuracy of the output produced. this is refered as "Underfitting"

This also shows that the Complexity for processing the input elements increases with overfitting. Also neural networks being a Complex interconnection of nodes the issue of overfitting may arise frequently

To eliminate this regularization is used in which we have to make the slightest modification in the design of the neural network and we can get better outcomes

REGULARIZATION IN MACHINE LEARNING

One of the most important factors that affect the machine learning model is overfitting.

The machine learning model may perform poorly if it tries to capture even the noise present in the dataset applied for training the system which ultimately result in overfitting

In this context noise does not mean the ambiguous or false data but those inputs which do not acquire the required features to execute the machine learning model.

Analyze these data inputs may surely make the model flexible, but the risk of overfitting will also increase accordingly.

One of the ways to avoid this is to cross validate the training dataset and decide according the parameters to include that can increase the efficiency and performance of model.

Let this be the simple relation for linear regression:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_p X_p$$

y = learned relation

β = Co-efficient estimators for different Variables and/or predictors(x)

Now we shall introduce a loss function that implements the fitting procedure Which is referred to as "Residual sum of Squares" or RSS

The co-efficient in the function is chosen in such a way that it can minimize the loss function easily.

Hence

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Above equation will help in adjusting the co-efficient function depending on the training dataset.

In case noise is present in the training dataset, Then the adjusted co-efficient wont be generalized When the future datasets will be introduced.

Hence at this point regularization comeinto a picture and makes this adjusted co-efficient shrink towards zero.

One of the methods to implement this is the ridge regression also known as L2 regression. Lets have a quick overview of this.

RIDGE REGRESSION:

Ridge regression also known as L2 regularization is a technique of regularization to avoid the overfitting in training dataset which introduces a small bias in the training model, through which one can get along term predictions for that input.

In this method a penalty term is added to the cost function. this amount of bias altered to the cost function in the model is also known as ridge regression penalty.

Hence the equation for the cost function after introducing the ridge regression.

$$\sum_{i=1}^{m}(y_i-y_i)^2 = \sum_{i=1}\left(y_i-\sum_{j=1}^{n}\beta_1 \times x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}\beta_j^2$$

Here λ is a multiplied by the Square of the weight set for the individual feature of the data.

It regularizes the co-efficient set of the model and hence the ridge regression term deduces the Values of the co-efficient Which ultimately helps in deducing the Complexity of Machine Learning model.

LASSO REGRESSION (L1 REGULARIZATION)

One more technique to reduce the overfitting and thus the complexity of The model is the lasso regression.

Lasso Regression stands for Least Absolute and Selection Operator and is also Sometimes known as L_1 regularization.

The equation for the lasso regression is almost same as that of the ridge regression except for a change that the value of the penalty term is taken as the absolute weights.

The advantage of taking the absolute Values is that its slope can shrink to 0, as compared to the ridge regression where the slope will shrink it near to 0.

The following equation gives the cost function defined in the lasso regression.

$$\sum_{i=1}^{m}(y_i - y_i)^2 = \sum_{i=1}^{n}\left(y_1 - \sum_{j=1}^{n}\beta_j \times x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}|\beta_j|^2$$

Due to acceptance of the absolute values for the cost function some of the features of the input dataset can be ignored completely while evaluvating the machine learning model and hence the feature section and overfitting can be reduced to much extent.

On the other hand, ridge regression does not ignore any feature in the model and includes it all for model evaluvation. The complexity of the model can be reduced using the shrinking of co-efficient in the ridge regression model.

DROPOUT:

Drop out was introduced by "Hinton et al" and this method is now very popular. It consists of setting a zero the output of each hidden neuron in choosen layer with some probability and is proven to be very effective in reducing overfitting.

The following equation gives the output function defined in the lasso regression

$$\sum_{i=1}^{m}(y_i - y_i)^2 = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{n}\beta_j \times x_{ij}\right)^2 + \lambda \sum_{i=0}^{n}|\beta_j|^2$$

Due to acceptance

To acheive dropout regularization some neurons in the artifical neural network are randomly disabled.

That prevents them from being too dependent on one another as they learn the correlation.

Thus the neurons work more independently and the artifical neuron network learns multiple independent correlations in the data based on the different Configurations of the neurons

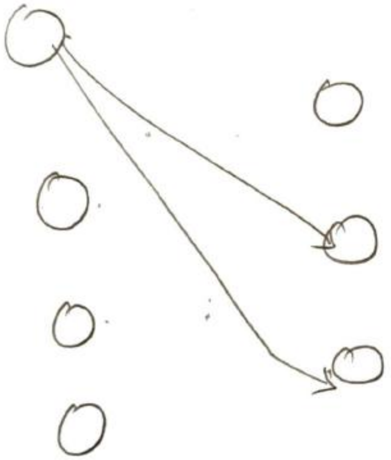
It is used to improve the training of neural network by omitting a hidden unit. It also speeds training.

Dropout is driven by randomly dropping a neuron so that it will not contribute to the forward pass or back propogation

Dropout is an inexpensive but powerful method of regularizing a board family of models.

DROPCONNECT :

DropConnect is known as the generalized version of the output, is the method used for regularizing deep neural network.



Dropconnect has been proposed to add more noise to the network. The primary difference is that instead of randomly dropping the output of the neurons we randomly drop the connection between neurons

In another words the fully connected layer with dropConnect becomes a Sparsley connected layer in which the connections are chosen at random during the training stage

L1 Regularization	L2 Regularization
penelizes the sum of absolote value of Weights	penalizes the Sum of square weights.
It has sparse Solution.	It has non-sparse solution.
It gives multiple Solutions	It has only one solutions.
Constructed in feature selection	No feature selection.
Robust to outliers	Not robust to outliers
It generates Simple and interpretable models.	It gives more acurate predictions When the output Variable is the fnction of whole input Variables
Unable to learn Complex data patterns	Able do learn complex data patterns

UNIT-V

DESIGN AND ANALYSIS OF MACHINE LEARING EXPERIMENTS.

MACHINE LEARNING LIFE CYCLE :

* The machine learning (ML) model management and the delivery of highly performing model is as important as the initial build of the model by choosing right dataset.

* The concepts around model retraining, model versioning, model deployment and model monitoring are the basics for machine learning operations that helps the data science teams deliver highly performing models.

* The use of machine learning has increased substantially in enterprise data analytics scenarios to extract valuable insights from the business data.

* Hence it is very important to have an ecosystem to build the model, build model, compute performance metrics and choose best performing model.

* The model maintenance plays a critical role once the model is deployed into production.

* The maintenance of machine learning model includes keeping the model up to date and relevant in tone with the Sorce data changes as there is a risk model becoming outdated in corse of time.

* Machine learning model lifecycle refers to the process that covers right from Source data identification to model development, model deployment and model maintenance. At high level, the entire activities fall under two broad categories such as ML model development and ML model Operations.

* The Machine learning lifecycle has the following phases.

Business goal Identification.

ML problem framing

Data processing (Data collection, data preprocessing feature engineering)

Model development (Training, tuning, evaluation)

Model deployment (Inference, prediction)

Model monitoring

Business goal:

An organization considering ML should have a clear idea of the problem and the business value to be gained by the solving problems. We most be able to measure business value against specific business objectives and Success Criteria.

ML problem framing:

In this phase, the business problem is framed as a machine learning problem: What is observed and what should be predicted (known as a label or target variable) Determining what to predict and how performance and error metrics most be optimized is a key step in this phase.

DATA PROCESSING:

Training an accurate ML model requires data processing to convert data into a usable format.

Data processing steps include collecting data preparing data and feature engineering That is the process of creating, transforming, extracting and selecting variables from data

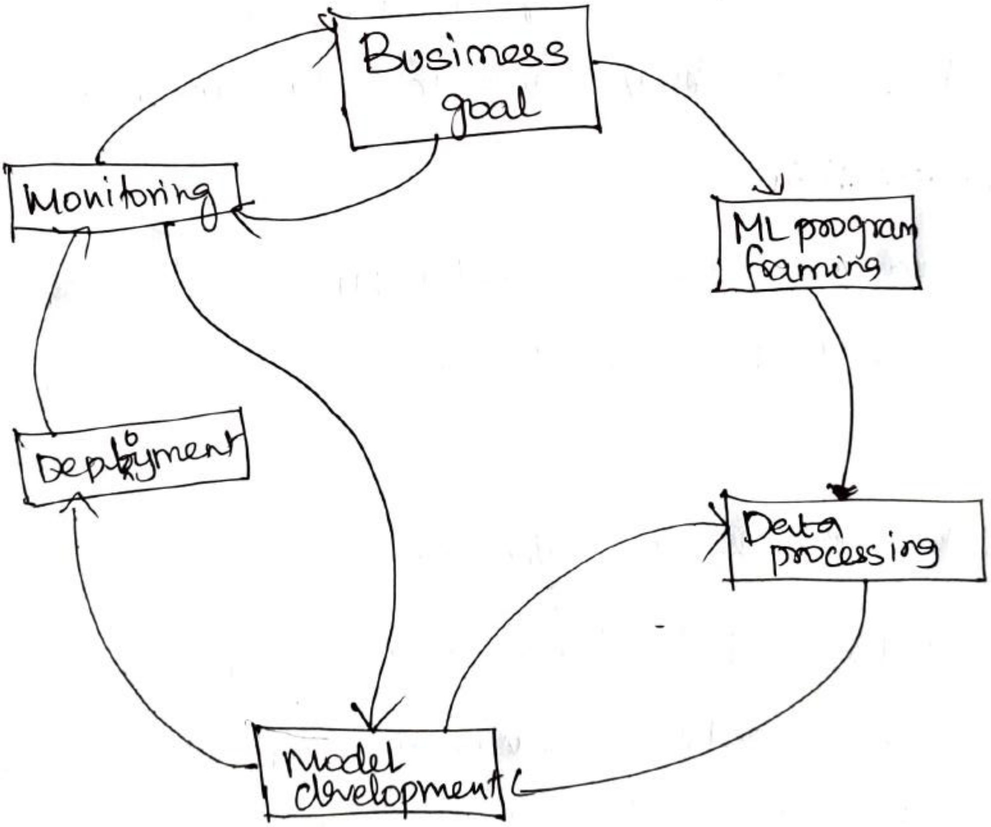
MODEL ~~DEVELOPMENT~~ DEPLOYMENT:

After a model is trained, tuned evaluvated and validated we can deploy the model into the production. we can make predictions and inferences against the model.

MODEL DEVELOPMENT:

* Model development consists of model building, training, tuning and evaluation.

* Model building includes creating a pipeline that automates the build, train and release to staging and production environments.



Machine learning lifecycle process

MONITORING:

Model monitoring System ensures your model is maintaining a desired level of performance through early detection and migitation.

GUIDELINES OF MACHINE LEARNING EXPERIMENTS

AIM OF THE STUDY:

*What are the Objectives (eg. assessing the expected error of an algorithm on a particular problem, ect..)

SELECTION OF THE RESPONSE VARIABLE:

*What should we use as the quality measure (eg. error, precision and recall, complexity, ect)

CHOICE OF FACTORS AND LEVELS:

*What are the factors for the defined aim of the Study (factors are hyperparameters when the algorithm is fix and want to find best hyperparameters, if we are comparing algorithms, the learning algorithm is a factor).

Choice of experimental design:

*Use factorial design unless we are sure that the factors do not interact

Replication number depends on the dataset size; It can be kept Small When The dataset is large.

* Avoid using Small datasets Which leads to response with high variance and the differences will not be significant and results will not be conclusive

Performing the experiment:

* Doings a few trial runs for Some random settings to check that all is as expected, before doing the factorial experiment.

* All the results should be reproducible.

Statistical analysis of the data: Conclusion we get should not be due to chance.

Conclusion and recommendations:

* One frequently conclusion is the need for further experimentation. There is always a risk that our Conclusion be wrong especially if the data is Small and noisy. When our expectations are not met, it is most helpful to investigate why they are not

DATASET PREPARATION :

* Machine learning is about learning some properties of data set and applying them to new data.

* This is why a Common practice in machine learning to evaluate an algorithm is to split the data at hand in two sets, one we call a training set on which we learn data properties and one the call a testing set, on which we test these properties.

* In the training data, data are assign the labels. In test data, data labels are unknown but not given. The training dataset consists of training examples.

* The real aim of Supervised learning is to do well on test data that is not known during learning.

* Choosing the values for the parameters that minimize the loss function on the training data is not necessarily the best policy.

* The training error is the mean error over the training sample.

* The test error is the expected prediction error over an independent test sample.

* problem is the training error is not a good estimator for the test error.

* Training error can be reduced by making the hypothesis more sensitive to training data but this may lead over fitting and poor generalization.

Training set: A set of examples used for learning, where the target value is known.

Test set:

* It is used only to assess the performance of a classifier.

* It is never used during the training process so that the error on the test set provides an unbaised estimate of the generalization error

* Training data is the knowledge about the data source which we use to construct classifier

* In a dataset, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built.

* Data points in the training set are excluded from the test (Validation) set.

* Usally a dataset is divided into a training set, a Validation set (Some people use 'test set' instead) in each iteration or divided into a training set, a Validation set and a test set in each iteration.

* In machine learning we basically try to create a model to predict the test data. So we use the training data to fit the model and testing data to test it.

* The models generated are to predict the results unknown which is named as the test set.

CROSS VALIDATION (CV) AND RESAMPLING

* Validation technique in machine learning are used to get the error rate of the ML model which can be considered as close to the

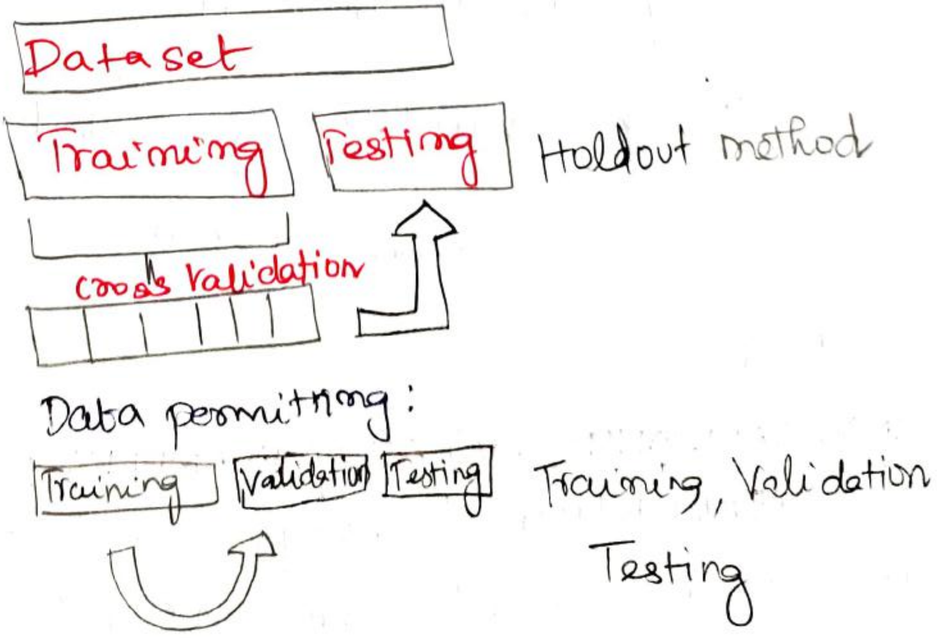
true error rate of population

*If the data Volume is large enough to be representive of the population, you may not need the Validation techniques.

*In Machine learning, model validation is referred to as the process where a trained model is evalovated with a testing dataset.

*The testing data set is a separate portion of the same data set from which the training set is derived.

*The main purpose of using the testing data set is to test the generalization ability of a trained model.



*Cross Validation is a technique for evaluvating ML models by training several ML models on subsets of the data.

*Use cross-validation to detect overfitting i.e falling to generalize a pattern.

*In general ML Involves deriving models from data, with the aim of acheiving some kind of desired behaviour eg, prediction or classification.

*But this generic task is broken down into a number of special cases. When training is done that data was removed can be used to test the performance of the learned model on "new" data

*This is the basic idea for a whole class of model evaluvation methods called cross Validation.

*Types of cross validation methods are holdout, k-fold and leave-one-out.

*The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing data set.

*The fuction approximate fits a fuction using the training set only k-fold cross validation is one way to improve over the holdout method.

* The dataset is divided into k subsets and the holdout method is repeated k times.

* Each time one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set.

* Then the average error across all k trials is computed.

* Leave-one-out cross validation is k-fold cross validation taken to its logical extereme with k equal to N the number of data points in the set.

* That ~~is~~ means that N separate times, the function approximate is trained on the data except for one point and a prediction is made for That point.

k-fold Cross Validation:

* k-fold CV is where a given dataset is split into a k number of sections/folds where each fold is used as testing set at some point

* Lets take the Scenarios of 5-fold cross Validation (k=5). Here, the dataset is Split into 5 folds.

* In the first interaction, the first fold is used to test the model and the rest used to train the model.

* In the Second iteration, 2nd fold is used as the testing set While The rest serve as the training set. This process is repeated until each fold of the 5 folds has been used as the testing data set.

* k-fold cross Validation is performed as per the following steps:

* partition the original training data set into k equal subsets, Each Subset is called a fold. Let the folds be named as f1, f2 ... fk.

For i=1 to i=k

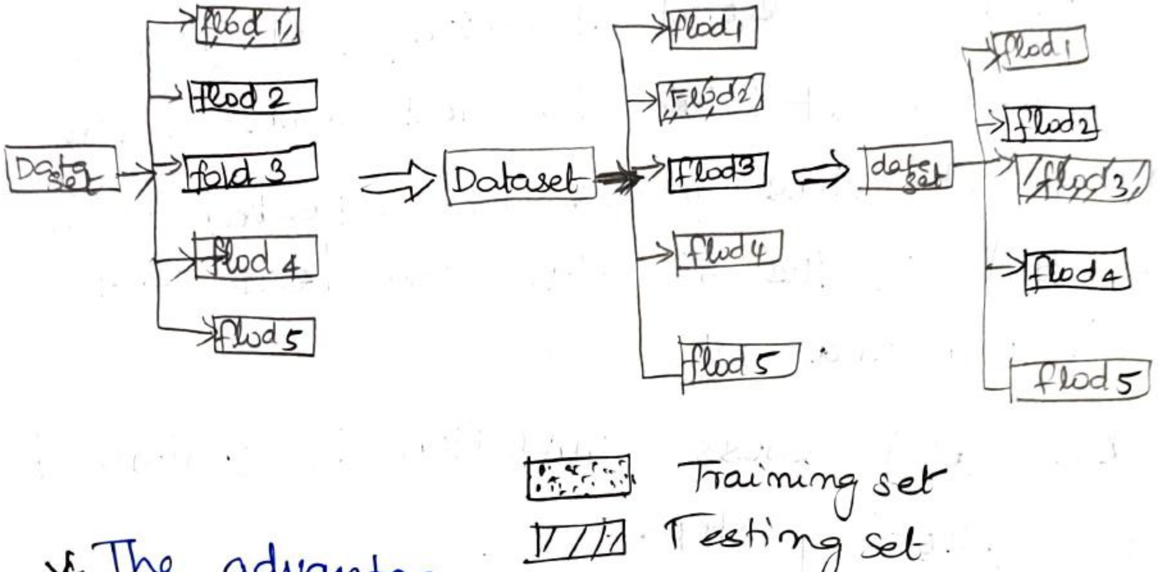
* Keep the fold fi as Validation set and keep all remaining k-1 folds in the Cross Validation training set.

* Estimate the accuracy of your machine learning model by averaging the accuracies

derived in all the kcases of cross Validation.

* In the k- fold cross validation method, all the entries in the original training data set are used for both training as well as Validation.

* Also each entry is used for Validation just once.



* The advantage of this method is that the matters less how the data get divided.

* Every data points gets to be in a test set exactly once and gets to be in a training set k-1 times.

* The variance of the resulting estimating is reduced as k is increased.

* The disadvantage of this method is that the training method has to be rerun scratch k times, Which means it takes k times as much

computation to make evaluation.

* A variant of this method is to randomly divide the data into a test and training set k different training set times

* The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

Bootstrapping :

* It is a method of sample reuse that is much more general than cross validation.

* The idea is to use the observed sample to estimate the population distribution.

* Then samples can be drawn from the estimated population and the sampling distribution of any type of estimator can itself be estimated.

* The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method

* For example it can provide an estimate of the standard error of a coefficient or a confidence interval for that coefficient.

* Suppose that we wish to Invest a fixed Sum of money into two financial assets that yeild returns of X and Y respectively Where X and Y are random quatitities.

* We will invest a fraction, a fraction α of our money in X and will Invest the remaining $1-\alpha$ in Y.

* We wish to choose α to minimize the total risk, or Variance, of our investment.

* In other words, we want to minimize $Var(\alpha X + (1-\alpha) Y)$.

* One can show That the value that minimizes the risk is given by,

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

Where

$\sigma_X^2 = Var(X)$, $\sigma_Y^2 = Var(Y)$ and $\sigma_{XY} = Cov(X,Y)$

* But the values of σ_x, σ_y and σ_{xy} are unknown

* We can compute estimates for these quantities $\hat{\sigma}^2_x, \hat{\sigma}^2_y$ and $\hat{\sigma}_{xy}$ using data set that contains measurements for x and y

* We can then estimate the value of α that minimizes the variance of our investment using

$$\hat{\alpha} = \frac{\hat{\sigma}^2_y - \hat{\sigma}_{xy}}{\hat{\sigma}^2_x + \hat{\sigma}^2_y - 2\hat{\sigma}_{xy}}$$

* To estimate the standard deviation of $\hat{\alpha}$, we repeated the process of simulation 100 paired observations of x and y estimating α 1,000 times.

* We thereby obtained 1,000 estimates for α which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots \hat{\alpha}_{1000}$.

* For these simulations the parameters were set to $\sigma^2_x = 1$, $\sigma^2_y = 1.25$ and $\sigma_{xy} = 0.5$ and also we know that the true value of α is 0.6.

* The means over all 1,000 estimates for α is

$$\hat{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996,$$

Very cose to $\alpha = 0.6$ and the standarad deviation of the estimates is

$$\sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000}(\hat{\alpha}_r - \hat{\alpha})^2}$$

$= 0.083$

* This gives us a very good idea of the acurracy of $\hat{\alpha}$: $SE(\hat{\alpha}) \approx 0.083$.

* So roughly Speaking for a random Sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08 on average.

* There are three forms of bootstrapping Which differ primarily in how the population is estimated.

NONPARAMETRIC BOOTSTRAP :

* In the nonparametric bootstrap a sample of the same size as the data is taken from the data with replacement.

* If we measure 10 Samples, we Create a new sample of size 10 by replicating Some of the Samples that we have already Seen and omitting others

SEMIPARAMETRIC BOOTSTRAP

* The resampling bootstrap can only reproduce the items that were in the original sample.

* The semiparametric bootstrap assumes that the population includes other items that are similar to the observed sample by sampling from a smoothed version of the sample histrogram.

* It turns out that this can be done very simply by first taking a sample with replacement from the Observed sample and then adding noise.

PARAMETRIC BOOTSTRAP:

* parametric bootstrapping assumes that the data comes from a known distribution with unknown parameters.

* We estimate the parameters from the data that you have and then you use the estimated distribution to Stimulate the Samples.

MEASURING CLASSIFIER PERFORMANCE

* A binary classification rule is a method that assigns a class to a object on the basis of its description

*The performances of a binary classifier can be assessed by tabulating its predictions on a test set with known labels in contingency table or Confusion matrix with actual classes in rows and predicted classes in Colomns.

Measures of performance need to statisfy Several criteria:

*They most coherently capture The aspect of performance of interest.

* They most be intutive enough to become widely used, so that the same measures are consistently reproted by researchers, enabling community-wide conclusions to be drawn;

*They most be Computationally tractable, to match the rapid growth in scale of modem data Collection.

*They must be simple to report as a single number for each method - dataset Combination.

performance metrics for binary classification are designed to captured tradeoffs between four fundamental population quantities : True positive, false positives, true negatives and false negatives

*The evaluation measures in classification problems are defined from a matrix with the number of examples correctly and incorrectly classified for each class, named confusion matrix.

*The confusion matrix for a binary classification problem is shown below.

True class	predicted class	
	Negative	postive
positive	False negative	True positive
Negative	True negative	False positive

*A confusion matrix contains about actual and predicted classifications done by a classification System.

*performance of such system is commonly using data in the matrix.

*Confusion matrix is also called as contigency table.

False positive :

Examples predicted as positive, which are from the negative class.

False negative :

Examples predicted as negative, whose true class is positive.

Trese positives :

Examples correctly predicted as pertaining to the positive class.

True negatives :

* Examples are correctly predicted as belongings to the negative class.

* The evaluvation measure most used in practice is the accuracy rate.

$$\text{Accuracy rate} = \frac{\text{True negative} + \text{True positive}}{\text{False negatives} + \text{False positive} + \text{True negative} + \text{True positive}}$$

ACCURACY AND ROC CURVES :

* Binary classification accuracy metric quantify the two types of. Correct predictions and two types of errors.

*Typical metrics are accuracy (ACC), precision, recall, false, positive rate, F1- measure.

* Each metric measures a different aspect of the predicative model.

* Accuracy (ACC) measures the fraction of correct predictions.

* predictions measures the fraction of actual positives among those examples that are predicted as positive.

* Recall measures how many actual positives were predicted as positive.

* F1 -measure is the harmonic mean of precision and recall.

ROC CURVE:

Receiver Operating Characteristics (ROC) graphs have long been used in Signal detection theory to depict the tradeoff between hit rates and false alarm fates over noisy channel.

Recent years have seen an increase in the use of ROC graphs in the machine learning Community.

* An ROC plots true positive rate in the Y axis false positive rate on the x-axis.

* A single contingency table corresponds to a single point in an ROC plot.

* The performance of a Ranker can be assessed by drawing a piecewise linear curve in an ROC plot, known as an ROC curve

* The curve starts in (0,0) finishes in (1,1) and is monitorically non-decreasing in both axes.

* In a ROC curve the true positive rate is plotted in function of the false positive rate (100 Specificity) for different cut-off points of a parameter.

* Each point on the ROC curve represents a sensitivity / specificity pair corresponding to a particular decision threshold.

* The area under the ROC curve is a measure of how well a parameter can distinguish between those two segments.

*An ROC curve is convex if the slopes are monotonically non-increasing when moving along the curve from (0,0) to (1,1).

*A concavity in an ROC curve i.e two or more adjacent segments with increasing slopes, indicates a locally worse than random ranking.

*In this we would get better ranking performance by Joining the segments involved in the Concavity Thus creating a coarser classifier

PRECISION AND RECALL:

*Relevance is a subjective notion. Different users may differ about the relevance or non-relevance of particular documents to given questions.

*In a response to a query an IR System searches its document collection and retrns a ordered list of responses.

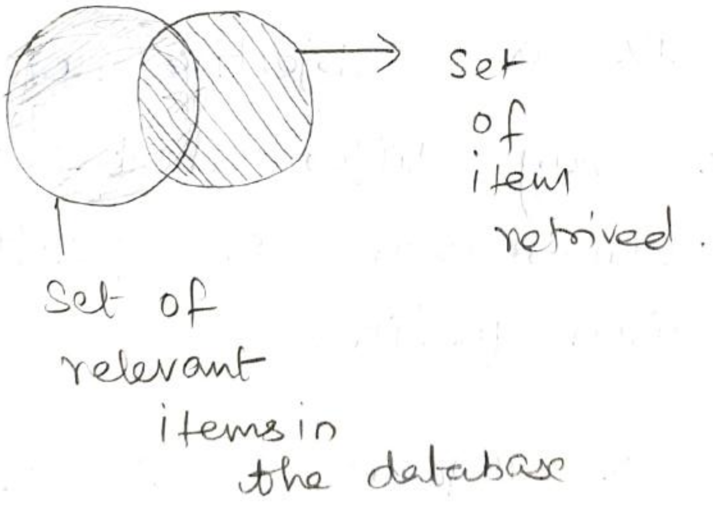
*It is called retrived set of ranked list

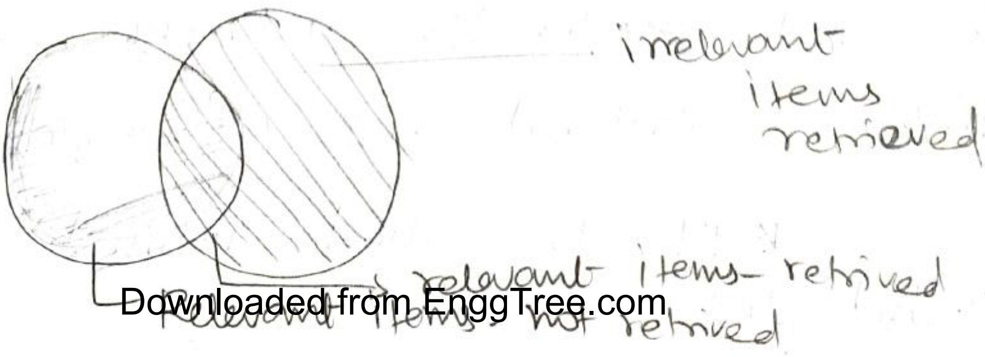
* A better search yields a better ranked list and better ranked lists help the user fill their information need

* This is a set of records in the database which relevant to the search topic

* Records are assumed to be either relevant or irrelevant.

* The actual retrival set may not perfectly match the set of relevant records.

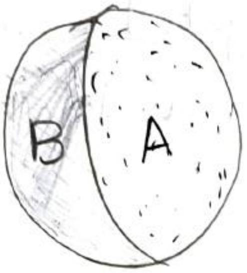




Recall :

* It is the ratio of The number of relevant records retrived to the total number of relevant records in the database.

* It is usually expressed as a percentage

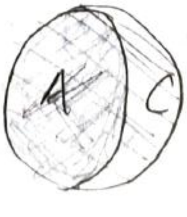


A = number of relevant records retrieved

B = Number of relevant records not retrieved

$$\text{Recall} = \frac{A}{A+B} \times 100\%.$$

* precision is the ratio of the number of relevant records retrieved to the total number of irrelevent and relevant records retrived. It is usually expressed as a percentage



A = number of relevant records retrieved

C = Number of irrelevant records retrieved

$$\text{precision} = \frac{A}{A+C} \times 100\%$$

As recall increase the precision decreases and recall decreases the precision increases.

F- Measure :

* It is a measure of test acuracy and is defined as the weighted harmonic mean of the precision and recall of the test.

* The F-measure or F-Score is one of the most commonly used "single number" measures in Information retrieval, Natural Language processing and Machine learning.

* F-measure comes from Information Retrival (IR) Where Recall is the frequency with which relevant documents are retrieved or recalled by a System, but it is known elsewhere as Sensitivity or True positive Rate (TPR)

* precision is the frequency with which retrieved docoments or predictions are relevant or correct and is properly a form of Acuracy also known as positive predictive Value (PPV) or True positive Acuracy (TPA) F is intended to Combine these into a single measure of Search effectiveness.

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}}$$

MULTICLASS CLASSIFICATION :

*It is a Machine learning classification task that consists of more than two Classes or outputs.

* Each training point belongs to one of N different Classes.

* The goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs

Weighted Average:

Mean average precision (MAP) is also called averag precisim at Seen relevent documents.

* Ɨt determene precision at each point when a new relevent document get retrieved.

* Averag of precision value obtained for the top K document each time a relevant document is retrieved.

$$MAP = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{Q_j} \sum_{i=1}^{Q_j} p(doc_i)$$

Where

Q_j = Number of relevant document for query j

N = number of queries.

p(doc) = precision at ith relevant documents

MULTICLASS CLASSIFICATION TECHNIQUES:

* Each training point belongs to one of N different classes.

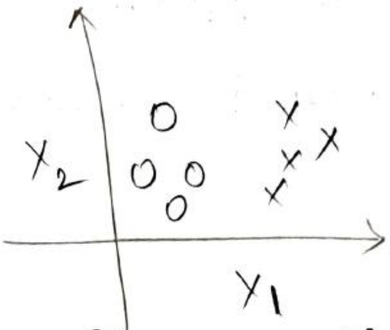
* The goal is to construct a function which given a new data point will correctly predicted the class to which the new point belongs.

* The multi-class classification problem refers to assigning each of the observations into one of K classes.

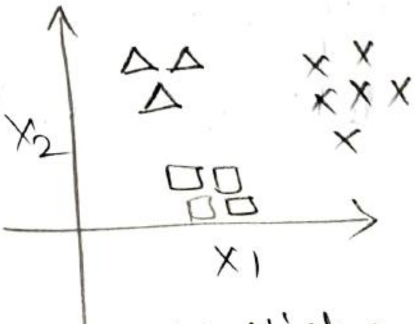
* A common way to combine pair wise comparisims is by voting.

* It constructs a rule for discriminating between every pair of classes and then selecting the class with the most winning two-class decisions.

* Through the voting procedure requires Just pair wise decisions, it only predicts a class label.



Binary classification



Multiclass classification

One Vs All (OVA)

*For this approach we require N = K binary classifiers, where kth classifier is trained with positive examples belonging to class K and negative examples belonging to the other K-1 classes

*When testing an unknown example the classifier producing the maximum output is considered the winner and this class label is assigned to that example.

Error Correcting ouptput Coding:

*Error Correcting code approaches try to Combine binary classifiers in a way that lets you exploit de-correlations and correct errors.

t- Test

+ When a small sample (size < 30) is considered, the tests are inapplicable because the assumptions we made for large sample tests, do not hold good for small samples.

* In case of small samples it is not possible to assume,

i) That the random sampling distribution of a statistics normal

ii) The sample values are sufficiently close to population values to calculate the S.E. of estimate.

* Thus an entirely new approach is required to deal with problems of small samples. But one should note that the methods and theory of small samples are applicable to large samples but it converse is not true.

* When sample size are small, as is often the case in practice, the central limit theorem does not apply. One must then impose stricter assumptions on the population to give statistical validity to the test procedure. One common assumption is that the population from which the sample is taken has a normal probability distribution to begin with.

* Degree of freedom (df): By degree of freedom we mean the number of classes to which the value can be assigned arbitrarily or at will without voicing the restrictions or limitations placed.

* For example, we are asked to choose any 4 numbers whose total is 50. Clearly we are at freedom to choose any 3 numbers say 10, 23, 7 but the fourth number, 10 is fixed since the total is 50 [50 - (10 + 23 + 7) = 10]. Thus we are given a restriction, hence the freedom of selection of number is 4 - 1 = 3.

* The degree of freedom (df) is denoted by ν (nu) or df and it is given by $\nu = n - k$, where n = number of classes and k = number of independent constrains.

t- Test for Single Mean

* When the sample values come from a normal distribution, the exact distribution of "t" was worked out by W.S. Gossett. He called it a t- distribution.

* Unfortunately, there is not one t-distribution There are different t- distributions for each different value of n. If n = 7 there is a certain t- distribution but if n = 43 the t - distribution is a little different. we say that the variable t has a t- distribution with n-1 degrees of freedom

* Suppose a simple random sample of size n is drawn from a population. If the population from which the sample is taken follows a normal distribution, the distribution of the random variable,

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Follows **Student's t-Distribution** with n-1 degrees of freedom.

* The sample mean is $\bar{x}$ and the sample standard deviation is s.
* The **degrees of freedom** are the number of free choices left after a sample statistic such as is calculated. When you use a t-distribution to estimate a population mean, the degrees of freedom are equal to one less than the sample size.

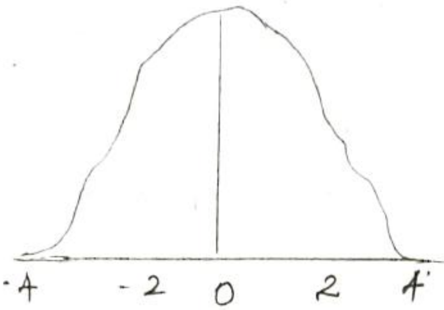
$$d.f. = n-1$$

Assumptions:

1. Population is normal although this assumption can be relaxed if sample size is "large"
2. Random sample was drawn from the population of interest.

- Based on the comparison of calculated 't' value with the theoretical 't' value from the table, we conclude:

Shape of Student's t- distribution



Properties of Students t- Distribution

1. The t- distribution is different for different degrees of freedom.
2. The t- distribution is centered at 0 and symmetric about 0.
3. The total area under the curve is 1. The area to the left of 0 is 1/2 and the area to the right of 0 is 1/2.
4. As the magnitude of t increases the graph approaches but never equals 0.
5. The area in the tails of the t- distribution is larger than the area in the tails of the normal distribution.
6. The shape of the t- distribution is dependent on the sample size n.
7. As sample size n increases, the distribution becomes approximately normal.
8. The standard deviation is greater than 1.

10. The area in the tails of the t- distribution is a little greater than the area in the tails of the standard normal distribution, because we are using s as an estimate of σ, thereby introducing further variability.

9. The mean, median, and mode of the t- distribution are equal to zero.

11. As the sample size n increases the density of the curve of t get closer to the standard normal density curve. This results occurs because as the sample size n increases, the values of s get closer to σ, by the law of large numbers.

T- Critical Values:

*Critical Values for various degrees of freedom for the t- distribution are (compared to the normal)

n	degree of freedom	$t_{0.025}$
6	5	2.571
16	15	2.131
31	30	2.042
101	100	1.984
1001	1000	1.962
Normal	"Infinite"	1.960

T- Test for Correlation:

*The Correlation co-efficient ρ (rho) is a popular statistics for describing the strength of the relationship between two variables.

* The correlation co-effdent is the slope of the regression line between two variables when both variables have been standardized by subtracting their means and dividing by their standarad means or deviations

* The correlation ranges between plus and minus one.

* When ρ is used as a descriptive statistic, no special distributional assumptions need to be made about the variables (Y & X) from which it is calculated

t- test for correlation coefficients

formula

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

with degrees of freedom equal to n-2

* state null and alternative hypothesis

$H_0 = \rho = 0$

$H_a = \rho \neq 0$

Here p is the population correlation co-efficient

- State the Significance level.
- Find the test statistics of correlation co-efficient with the above-defined formula
- To make a decision use the critical value approach or the p-value approach
- Finally state the conclusion.

Mc Nemar's Test:

It is a non-parametric test for paired nominal data.

It is used for finding a change in proportion for paired data.

It Compare the performance of two classifiers on N items form a single test set

This test is used to Compare the performance of two classifiers on the same test set.

* This test works if there are a large number of items on which A and B make the predictions.

* Mc Nemar's test is applied to 2x2 contigency tables with matched pairs of subjects to determine whether the row and column marginal frequencies are equal.

The three main assumptions of test are :

* We must have one nominal variable with two categories and one independent variable with two connected groups.

* The two groups in the dependent variable must be mutually exclusive.

* Sample must be random sample

K - flod CV paired t Test :

We use K - fold cross validation to get K training / validation set pairs

*To train the two classification algorithms on the training sets Ti; Where i=1 ; K and test on the Validation Sets Vi

*The error percentages of the classifiers on the Validation sets are recorded as p_i^1 and p_i^2.

*If the two classification algorithms have the same error rate, then we expect them to have the same mean or equivalently, the difference of their means is 0.

*The difference in error rates on fold i as $p_i = p_1^1 - p_1^2$. This a paired test.

*That is for each i both algorithms seems the same training and Validation sets.

$$m = \frac{\sum_{t=1}^{N} p_i}{K} \qquad S^2 = \frac{\sum_{t=1}^{k} (p_i - m)^2}{K-1}$$

*Under the null hypothesis that $\mu = 0$, we have a statistic that is t-distributed with k-1 degree of freedom.

$$\frac{\sqrt{k}(m-0)}{S} = \frac{\sqrt{k}(m)}{S} \sim t_{k-1}$$

*Thus the k-fold cv paired t-test rejects the hypothesis that two classification algorithms have the same error rate at significance level α if this value is outside the interval $(-t_{\alpha/2,k-1}, t_{\alpha/2,k-1})$

*If we want to test wheather the first algorithm has less error than the second, we need a one sided hypothesis and use a one-tailed test:

$$H_0 : \mu \geq 0 \text{ vs } H_1 : \mu < 0$$

*If the test rejects, our claim that the first one has significantly less error is supported

* Advantage is that each testsetis independent of others.

* But the traimeing set still overlap.

* The overlap may prevent the test from obtaining a good estimat of the amount of Variation that would be observed if each training set were completely independent of previous training sets.

* The variance in the t statistics may be sometimes underestimated, the means one ocassionally poor estimated and this may result in large t Values