# Introduction to Machine Learning & AI


Talk for indiamart

***Saurabh Singal, ANKAM Private Limited, Singapore***

# Convolution and CNN

*Source: The first 14 equations of Chapter 9 of Deep Learning by Goodfellow, Bengio & Courville*

$$s(t) = \int x(a)w(t-a)da. \qquad s(t) = (x * w)(t).$$

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a). \qquad S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n).$$

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n). \qquad S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n).$$

$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n}, \qquad Z_{i,j,k} = c(\mathbf{K}, \mathbf{V}, s)_{i,j,k} = \sum_{l,m,n} \left[ V_{l,(j-1)\times s+m,(k-1)\times s+n} K_{i,l,m,n} \right].$$

$$Z_{i,j,k} = \sum_{l,m,n} \left[ V_{l,j+m-1,k+n-1} w_{i,j,k,l,m,n} \right]. \qquad Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n,j\%t+1,k\%t+1},$$

$$g(\mathbf{G}, \mathbf{V}, s)_{i,j,k,l} = \frac{\partial}{\partial K_{i,j,k,l}} J(\mathbf{V}, \mathbf{K}) = \sum_{m,n} G_{i,m,n} V_{j,(m-1)\times s+k,(n-1)\times s+l}.$$

$$\mathbf{R} = h(\mathbf{K}, \mathbf{H}, s).$$

$$h(\mathbf{K}, \mathbf{G}, s)_{i,j,k} = \frac{\partial}{\partial V_{i,j,k}} J(\mathbf{V}, \mathbf{K}) = \sum_{\substack{l,m \\ \text{s.t.} \\ (l-1)\times s+m=j}} \sum_{\substack{n,p \\ \text{s.t.} \\ (n-1)\times s+p=k}} \sum_q K_{q,i,m,p} G_{q,l,n}.$$

# A word of caution

- TED talks are all limited to 18 minutes "because the brain is an energy hog. The average adult human brain only weighs about three pounds, but it consumes an inordinate amount of glucose, oxygen, and blood flow. …you cannot inspire people if you put them to sleep. But scientists are beginning to identify how long most people can pay attention before they tune out. The range seems to be in the area of 10 to 18 minutes…"

  ***Carmine Gallo**, The Science Behind TED's 18-Minute Rule*

- Instead of 18 minutes, we have the luxury of 120… we will proceed slowly.

- ***The opening slide was just a jestful attempt to put you at ease. Hardly any equations will trouble you in this session.***

# Confusion matrix:
# 3% of the People use 5-6% of Brain…

They say 3 percent of the people use 5 to 6 percent of their brain
97 percent use 3 percent and the rest goes down the drain
I'll never know which one I am but I'll bet you my last dime
99 percent think we're 3 percent 100 percent of the time.
65 percent of all the world's statistics are made up right there on the spot
82.4 percent of people believe 'em whether they're accurate statistics or not
I don't know what you believe but I do know there's no doubt

I need another double shot of something 90 proof…

-Statistician's Blues by Todd Snider

# Part 1: Introduction and Use Cases

- In the first one hour, we will cover examples where Machine Learning and AI are being applied. At times they seem to work magically.

- We will take a glimpse into how this magic works in the second part of this lecture

# What is Machine Learning?

I have yet to attend a "Introductory Machine Learning" talk which does not ask this question.

- In the beginning, statisticians and computer scientists approached similar problems within their own communities.
- Computer Science community called this discipline ***Artificial Intelligence*** or ***Machine Intelligence*** and statisticians called it ***Statistical Learning***.
- Current understanding :

Machine Learning = Machine Intelligence+ Statistical Learning; programs that learn from data.

- In this lecture, Machine Learning will be used loosely and broadly to include a wide spectrum of statistical techniques.
- Deep Learning is the latest paradigm but the phrase "shallow learning" sounds very silly, so we will use the word "***Traditional Machine Learning***" to denote all "Machine Learning excluding-Deep Learning".

# What is Machine Learning? (contd.)

- Machine learning is a way for computers to learn rules from data without these rules being specifically programmed. *As opposed to* Statistical analysis which is driven by the user, Machine Learning or AI is data-driven.
- The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.
- In machine learning, our goal is either prediction or clustering. Prediction is a process where, from a set of input variables, we estimate the value of an output variable. For example, using a set of characteristics of a house, we can predict its sale price.
- Prediction problems are divided into two main categories:
  - Regression problems, where the variable to predict is numerical
  - Classification problems, where the variable to predict is part of one of some number of pre-defined categories, which can be as simple as "yes" or "no."

# What is Artificial Intelligence?

- Field that deals with the design and application of algorithms for the analysis of, learning from and interpreting data.
- AI encompasses many branches of
  - statistical learning,
  - pattern recognition,
  - clustering, and similarity-based methods,
  - biologically motivated approaches, such as neural networks,
  - evolutionary computing or fuzzy modeling,

These are collectively described as ***Artificial Intelligence or Machine Learning***

# What is Data Science ?

# Background : Applications of Machine Learning

Machine Learning can solve a wide array of real-life problems:

**Finance**

- Financial modeling
- News  Sentiment Analysis
- Trading signals
- Fraud detection

**Biology**

- Patient behaviors and disease
- New Drug Design
- Personalized Medicine
- computer-aided diagnosis

**Retail and Marketing**

- Item recommendation
- Market segmentation
- Targeted Ads
- Buyer Sentiment Analysis

**Vision**

- Face recognition
- Handwriting Recognition
- Image Segmentation
- General Object Classification
- Personal Assistant

**Language**

- Language Translation
- Text Classification
- Sentiment Analysis
- Question Answering
- Summarization

# Examples from a Singapore ML Consultancy

- **Camera surveillance: training machines using images to identify lethal objects**
- **Air traffic control: training machines to identify possible collision scenarios by making them learn through speech recognition data between Pilots & air traffic controllers**
- **Song piracy: training machines using speech recognition to scan internet audio content for piracy**
- **Virtual tour guide for a country's tourism promotion authority to allow potential tourists to plan itinerary using voice & text over internet by training machines to identify key words spoken or written**
- *Computer Aided Learning (CAL):* **language self learning by training machines to recognize speech and hence identify learner's weaknesses and churn out relevant exercises for learner to improve in those areas**
- **emotion detection & identification in text data and hence to explore implementation in real life automated interactive response systems such as commercial chat bots.**

# How can AI help


- It is all very well to talk about the greatness of AI.
- AI can play chess, select music, suggest new friends, solve route planning, help with investing and trading, etc. etc., but how can AI help problems unique and specific to IndiaMART?
- We are going to see two situations where AI can help IndiaMART…
- There are going to be several more areas like this.
- I am hopeful that as we go through the rest of the talk, there will be some sparks ignited and more problems will be suggested by you.

# Some Background: Lead Scoring

- When there are hundreds or thousands of leads, a sales team might need to know which to reach out first. If done manually, it calls for experience.
- This is because there are many variables that can be used.
  - Choose an established company or startup?
  - B2B or B2C?
  - In which geography?
  - What business sector?
- AI can be a natural fit here and predict or prioritize the leads by giving each lead a score.
- Analyze past data to see which leads got converted to sales. This allows some leads to be identified as "opportunities".
- 6Sense, LatticeEngine: which predict the 'fit' of a lead and 'intent to buy'

# Lead Scoring : AI outperforms Humans

- A 2014 paper from MIT Media Lab compared computers and humans in selecting users likely to convert from desktop internet use to mobile internet users.
- The machines won hands down, with a conversion ratio which was 13x
- And 98% of the machine-identified persons who signed up renewed their subscription as opposed to 37% renewal rate for those selected by humans.
- Machine Learning outperforms humans often in high dimensional, high cardinality problems involving numbers and probability.

# indiamart : BuyLead Selection Prediction

**Background**

- Buyers on IndiaMART platform share their buying requirements.
- IndiaMART team works on ALL these buying requirements; verifies and enriches them and makes them available on the platform, in the form of BuyLeads
- Sellers who find the BuyLead relevant to them select the lead to get the contact details of the buyers

**Important Facts**

- IndiaMART incurs heavy cost for the above process of verification and enrichment of the BuyLeads (Verification and Enrichment Process).
- Not all BuyLeads Approved get selected by the sellers. This essentially means that the cost incurred to approve those BuyLeads did not result into ROI.

## Current Process

Use a matrix combining the following variables to estimate the chances of a BuyLead getting selected by the seller:

- ❑ Category+City Combination wise BuyLead Selection (as a % of verified leads) historically
- ❑ Average Order Value of the BuyLeads in the Category
- ❑ Buyer Type (Repeat buyer, Seller as a buyer etc.)

# :BuyLead Selection Prediction-2

**Another Problem for AI to solve**

**Predict, before verification and enrichment,
The probability of the BuyLead getting selected by sellers
thus creating ROI.**

## Objective

- To reduce the verification cost of the low ROI BuyLeads
- To focus more on improving the quality of the high ROI BuyLeads
- To ensure that ALL high ROI BuyLeads are processed at the Verification Process.

## Measure of Success:

- ✓ Reduction in Cost of Verifying a lead
- ✓ Increase in Total Verified BuyLeads and selection of High ROI Leads by sellers

# : BuyLead Relevancy for Seller

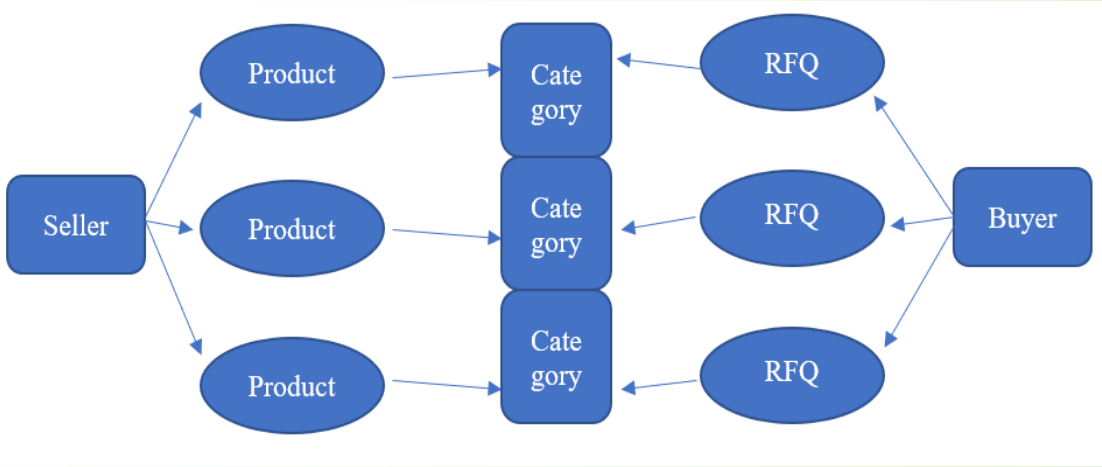
## Background

- Buyers on IndiaMART platform share their buying requirements.
- IndiaMART team verifies and enriches those requirements and makes them available on the platform, in the form of BuyLeads, for sellers to consume.
- Sellers join IndiaMART to get buyers. These BuyLeads are one of the most important sources of buyers for the sellers.
- The sellers should visit the IndiaMART Website to view the Relevant BuyLeads for them and select them to get the contact details of the buyers.
- These buyers can then be contacted by the sellers for the fulfilment of their buying requirement.

## Current Process

Use a matrix combining the following variables to estimate the chances of a BuyLead getting selected by the seller:

- ❑ Preferred City of the seller basis historical behavior
- ❑ Preferred Category of the seller basis historical behaviour
- ❑ Matching of Prime vs Secondary Category Mapping of the Product and the BuyLead
- ❑ Time of the BuyLead Verification




MAPPING BUYER AND SELLER TO CATEGORY

# : BuyLead Relevancy for Seller -2

**Another Problem for AI to solve**

**Create and show BuyLead recommendation (default ranking of available BuyLeads) as per sellers' past behaviour / preference**

## Objective

- To improve seller satisfaction and save seller's time by way of showing most relevant BuyLeads on top.
- To increase Daily Active Suppliers
- To increase BuyLead selection per Supplier
- To increase Mean Reciprocal Rank of the BuyLead selected

To improve the Buyer Fulfilment on IndiaMART platform, it is imperative that all the BuyLeads get consumed by the Sellers. To ensure that Sellers select the maximum BuyLeads, it is further important that the BuyLeads recommended to the sellers are highly relevant to them.
This will improve the sellers' experience of BuyLead selection activity resulting into higher selection and improved satisfaction.

**Measure of Success:**

- ✓ Increase in BuyLead consumption by 50%
- ✓ Increase in Daily Active Suppliers by 50%
- ✓ Increase in BuyLead selection per Supplier by 50%
- ✓ Increase Mean Reciprocal Rank of the BuyLead selected by 50%

# Application1: Predicting Which User Will Click An Ad

- In a Pay-Per-Click (PPC) format, the ad platform is paid by the advertiser for each click on the ad.
- Therefore, determining which ad is most relevant to the end-user is important. If the ad is of interest to the user, the chance of click is higher. This benefits the advertising platform as well as its client (the actual advertiser).
- Several Machine Learning based approaches are possible, including Logistic Regression

# Application1: Click-Through Rate Competition

- Avazu is an advertising platform that delivers ads on websites and mobile applications.
- They organized a competition for Click-Through Rate prediction in 2015 on Kaggle.
- Dataset: Avazu made available a dataset with the log of 11 days of impressions and clicks with information about users.
- Logistic regression is effective.

# Application1 :Predicting Click-through Rates

- It is a very large scale problem.
- A fast method is needed because It is necessary to make predictions many billions of times per day and to quickly update the model as new clicks and non-clicks are observed – this means dataset is also immense and growing fast.
- A sophisticated technique is **Follow the Regularized Proximal Leader (FTRL Proximal).**
- "Ad Click Prediction: a View from the Trenches" by McMahan *et al*. http://www.eecs.tufts.edu/~dsculley/papers/ad-click-prediction.pdf

# Application 2: Real-time Bidding in Sponsored Search Auction

- A keyword auction or Sponsored Search auction refers to those results from a search engine (like Google) which are not the output of the main search algorithm but are advertisements related to the keyword.
- In the beginning all ads were paid by impressions as it is the easiest metric to measure. An **impression** is when an ad is fetched from its source, and is countable. Whether the ad is clicked is not taken into account. Each time an ad is fetched, it is counted as one impression. *(source: Wikipedia)*
- Nowadays, on Google AdWords, a variant of the Generalized Second Price Auction is used. In this auction there are multiple slots and several bidders.
- Bidding is sealed-bid or blind auction.
- The first slot goes to the highest bidder, the second slot to the next highest bidder and so on till all slots are filled; but the but the highest bidder pays the price bid by the second-highest bidder, the second-highest bidder pays the price bid by the third-highest bidder.

# Application 2: Alibaba, Reinforcement Learning & Real-Time Bidding

- *Deep Reinforcement Learning For Sponsored Search Real Time Bidding by* Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, Xiaofei He. This is a recent paper(2018) using Markov Decision Process in a Reinforcement Learning framework.

- Dataset: Numerous ads were selected from Alibaba platform, (at least a 100 million auctions per day). The training dataset was extracted in one month and the test dataset was extracted in the next month. Every sample consisted of the clicks, bids and the ranking board.

# Interlude: Amusing Facts

- Prior to September 1993 the World Wide Web was entirely indexed by hand. There was a list of webservers edited by Tim Berners-Lee and hosted on the CERN webserver.
- The first tool used for searching content (as opposed to users) on the Internet was Archie. The name stands for "archive" without the "v". Nothing to do with the comic character
- The next 2 web search engines were called Veronica and Jughead.

# Application 3: Recommender Systems

- The Traditional definition of Recommender Problem is:
  - ***Predict how much a user will like a certain item, based on past behaviour, context, similarity to other items, relation to other users, ….***
- Collaborative Filtering: method of making automatic predictions (filtering) about the interests of a user
  - by collecting preferences information from several people (collaborating).
  - If Dinesh and Brijesh have similar opinion on cricket, Brijesh might have an opinion on football similar to Dinesh's, as compared to mine!

# Application 3: Netflix Prize

- In October 2006 Netflix announced a competition for a recommender system that could beat its own proprietary Cinematch system by 10%.
- Netflix provided a training **data set** of 100,480,507 ratings that 480,189 users gave to 17,770 movies.
- The first prize was 1 million dollars, awarded in Sep 2009 to team "BellKor's Pragmatic Chaos". The winners achieved a 10.06% improvement over Cinematch's score on the test subset at the start of the contest. The winners leap-frogged over the nearest rivals just 24 minutes before the end of the deadline.

# Application 3: Collaborative Filtering

- In an overly simplistic description, we could search for a group of users
  - Who have a similar ratings pattern to the user in question
  - Use the ratings from this group of users to make predictions
- This is similar to k-Nearest Neighbours at its core, but this is not the algorithm used to solve the problem.

# Application 3: Problems with Collaborative Filtering

- Data Sparsity: Usually the vast majority of ratings are unknown. For example, in the Netflix data 99% of the possible ratings are missing.
- Cold Start Problem (no history for a new user OR a new product)
- Typical CF data exhibit large user and item effects – i.e., systematic tendencies for
  - some users to give higher ratings than others…
  - …and for some items to receive higher ratings than others.

# Application 3: Item-based Collaborative Filtering

- A variation is item to item or item-based collaborative systems or item-to-item collaborative filtering (people who buy ***X*** also buy ***Y***) first used by Amazon.

- *Item-based Collaborative Filtering Recommendation Algorithms* by Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl

# Application 3: Content Based Filtering

- **Content-based filtering** methods are based on a **attributes of the item** and a **profile of the user**'s preferences to recommend items that are similar to those that a user liked in the past, or is examining in the present.
- We need to make **both item-profile and user-profile**. We need to know about the content of the items.
- Example : Ram likes Jalebi;Jalebi and Imarati are similar; recommend imarati
- **LinkedIn** uses content based filtering for "connect with people you may know"
- **Facebook** uses content based filtering when suggesting new friends
  - E.g., Ram bought a cricket bat, so recommend gloves.
- **Mobile recommender systems**: new area, context sensitive.
  - E.g., Taxi routes a taxi driver should take to maximize his revenue.
  - E.g., do not suggest certain recipes in certain countries.

# Association Rules Mining

- Associations Rules Mining is a rule-based Machine Learning technique used to discover interesting relations between variables in large databases.
- Association rules are helpful in identifying things that occur together.
- Illustration: identification of regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets E.g., the rule {blades, shaving cream} ⇒{after shave} indicates that if someone buys blades and shaving cream, he will likely buy after shave.
- Sequential Associations
  - Discovers association rules in time-oriented data
  - Find the sequence or order of the events

# Application 4: Application of Association Rules - Market Basket Analysis

- Market Basket Analysis uses Association Rules Learning for understanding the purchasing behaviour of shoppers; helpful in campaigns for cross-selling and up-selling.
- E.g., shoppers buy shampoo and conditioner together, so do not offer promotions on both at the same time (since increased sales of the discounted item might lead to more sales of the other item at the standard price).
- Helps in designing the layout stores more efficiently.
- It can also be used to cluster the shoppers into groups or micro-segments
  - Eggs+( flour, sugar): baking
  - Eggs + cheese: making omelets
- Amazon's : "customers who bought book A also bought book B"

# Application 4: Interesting Examples of Association Rules

- Milk is the most purchased item so it is always in the back of the store, making you walk by everything thing else, (just as duty free shops are just after passport control and before you reach the luggage belts!)
- Women's shoes are always on the way to men's clothes.
- Chocolates and bananas are at the front of 7-11 stores because they are found to be an impulse buy.
- Beer and Diapers
  - Men between 30- 40 years in age,
  - shopping between 5pm and 7pm on Fridays,
  - who purchased diapers were most likely to also
  - have beer in their carts.
  - This motivated the grocery store to move the beer isle closer to the diaper isle

# Application 4: Apriori Algorithm & Frequent Item-set Mining

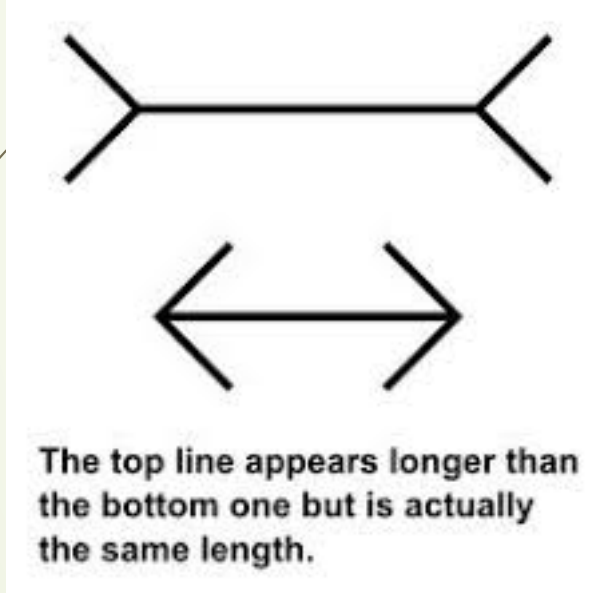
- Example:

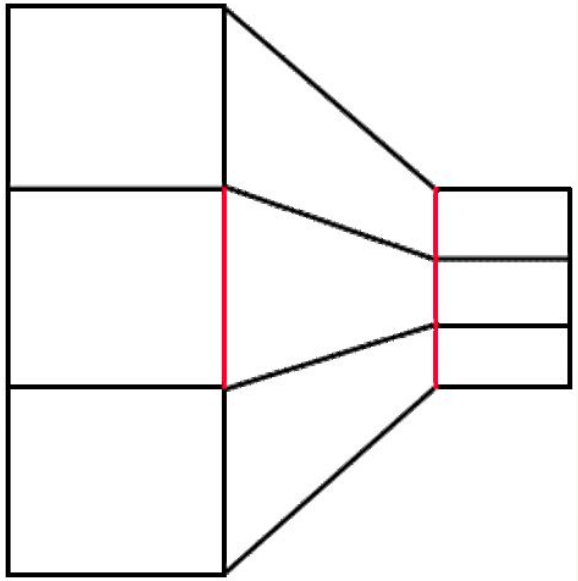
| | | |
|---|---|---|
| Apple | Banana | Cherry |
| Apple | Banana | Cherry |
| Apple | Banana | Cherry |
| Apple | Banana | Grapes |

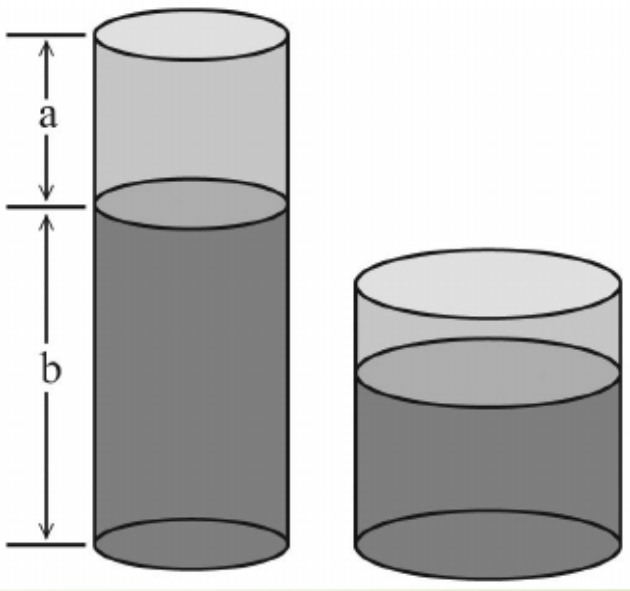
- **Rules**:
  - 100% of those who bought apples bought bananas
  - 75% of those who bought apples (or bananas) also bought cherries
  - 25% of those who bought apples (or bananas) also bought grapes
- **Support**: how frequently the itemset appears in the dataset {A,B} 100%
- **Confidence**: how often the rule has been found to be true {A,B->G} is 25%
- **Lift**: Support(X union Y)/[Support(X).Support(Y)]
  - Lift=1: independent
  - Lift<1 : the items substitute each other and buying X means you are less likely to buy Y.
  - Lift>1: positive effect, presence of X is more likely to cause purchase of Y

# Where Machine Outperforms Man

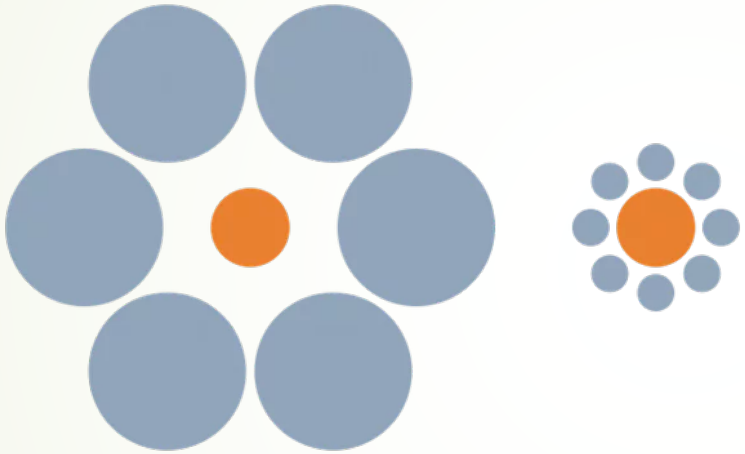
- **Obviously calculations**, but in other visual task as well…



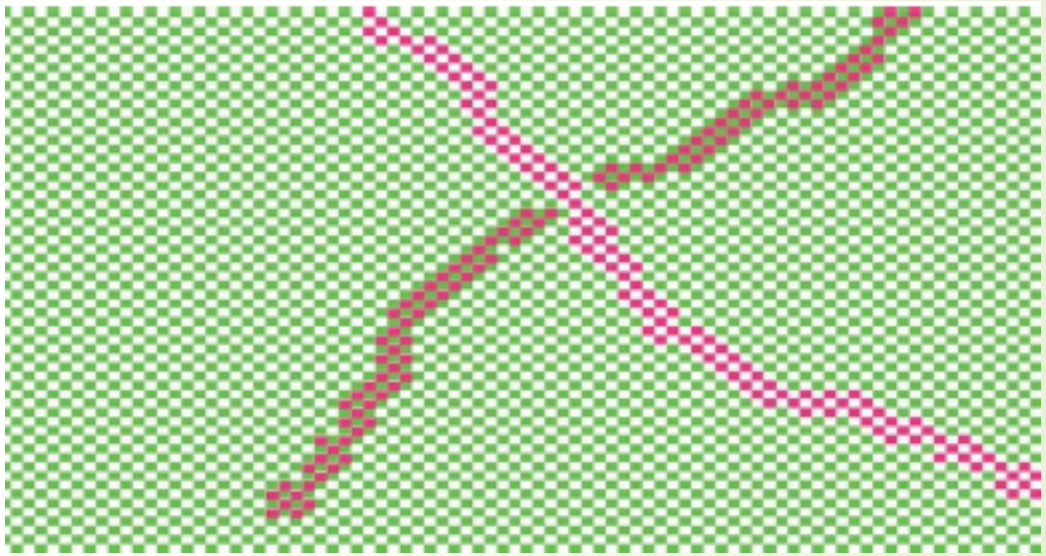





# Illusions with Area and Colour



Ebbinghaus Illusion: The two orange circles have the same radius

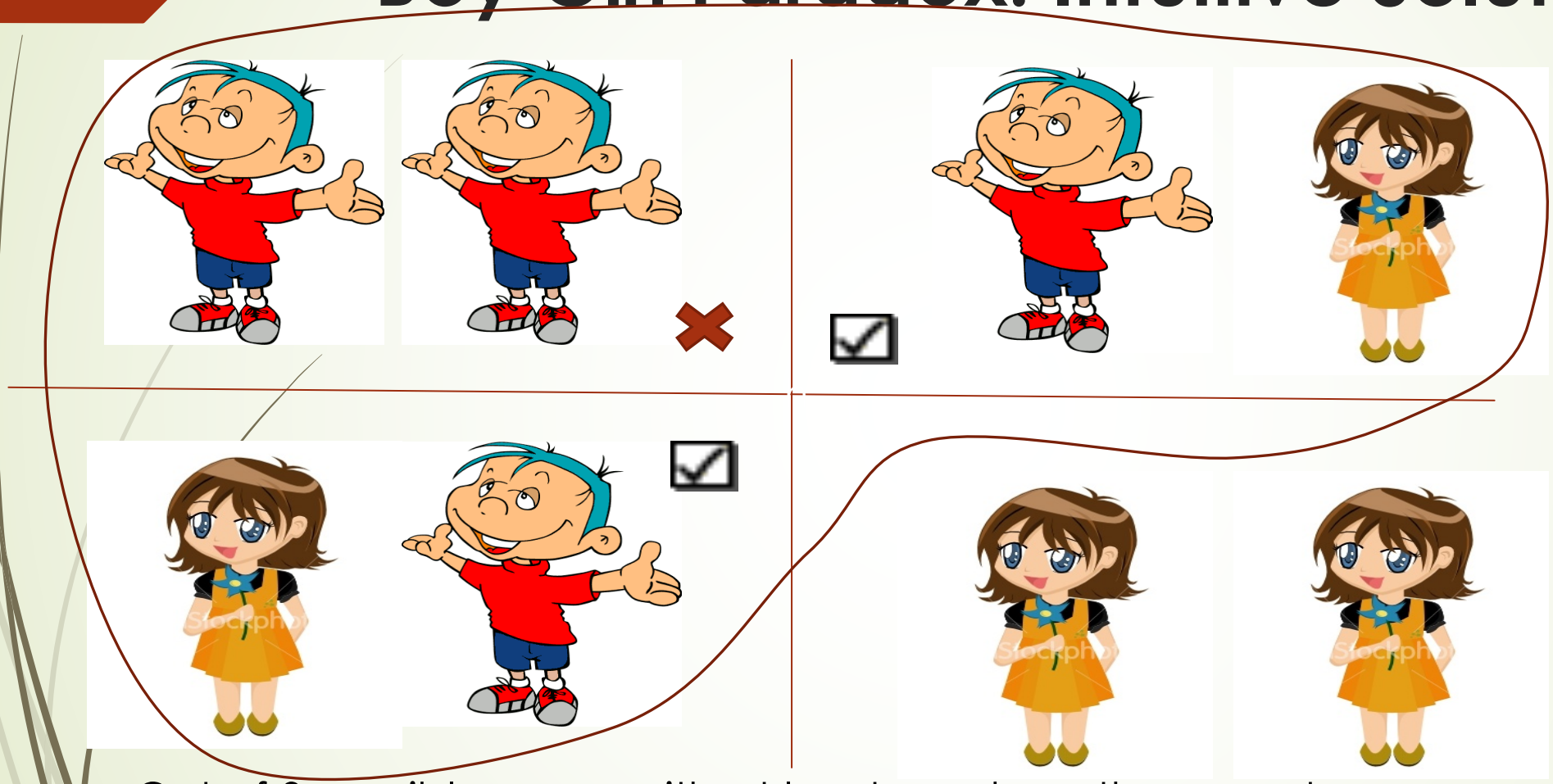


The two lines of pixels which appear pink and red are of the exact same shade

# Probabilities, Conditional Probabilities…
## *The Boy-Girl Paradox*

- Let us start with a fun example which can be solved with or without the use of Bayes Theorem. This will give us a taste for what Bayes' Theorem can do for us!
- A new neighbor moves in next door. You learn that he has **2** children.
- You see that one of the children is a boy.
- What is the probability that the other child is a girl?

# Boy Girl Paradox: Intuitive Solution



Out of 3 possible cases with at least one boy, there are two cases with a girl-

$$Prob(other\ child\ is\ a\ girl) = \frac{2}{3}$$

# Where Man Outperforms Machine

- Recognizing Faces
  - Even a few months old baby does a good job at recognizing faces
- Recognizing language:
  - Grammar,
  - which language?
- Recognizing Speech
- **Recognizing Sarcasm, Recognizing Emotion**
  - **But what about betraying emotions? Getting emotional?**

# Application 5: Automated Negotiation

- "Deal or No Deal? End-to-End Learning for Negotiation Dialogues" by Mike Lewis et al at Facebook AI Research.
- Can an algorithm *negotiate a deal* with a human, and get a better outcome than another human? **Answer : Yes**
- Even though negotiation combines cold logic, analysis and reasoning with acting, bluffing, stubbornness and compromise, some of which are very human characteristics, and also hard to quantify, the computers could be stubborn, learn deception and beat humans.
- The game: a certain number of books, hats and balls are to be divided between two players. Each object is valued differently by each player but the sum of the value was 10 for each.

# Application 5: Deal or No Deal

- For example: 1 book, 2 hats and 3 balls.
- Player 1: Book = 8 points, hat = 1 point each, ball = 0 points each (total 10)
  Player 2: Book = 3 points, hat = 2 points each, ball = 1 point each (total 10)
- The players negotiate amongst themselves trying to divide these goods in such a way that they maximize their wealth. But if after 10 rounds no solution is reached, the game is abandoned and they each get nothing (bad for both).
- Researchers made pairs of human play 5808 rounds and collected the data.
- Based on this, a recurrent neural network (RNN) program was trained. It learn to make proper sentences but lost easily as it was too willing to compromise.

# Application 5: Deal or No Deal(2)

- Then a second program was trained on the data arrived at by collecting the human-RNN interactions.
- This time Reinforcement learning was used.
- Human-human data to build the first AI Negotiator, which in turn was used to train AI Negotiator v2, which negotiated with humans
  - **AI Negotiator v2 negotiated harder and was more stubborn** (used 7.2 turn versus 5.2 for humans; accepted deals less quickly; restated same demands sometimes in different words).
  - **AI Negotiator v2 did not get offended or walk away.**
  - **AI Negotiator v2 learnt deception (**show false interest in a useless item then "compromise" by giving it away).
  - **AI Negotiator v2 learnt to make correct, new sentences that were not in training data.**

# Application 6: Spam and Ham

- Spam:
  - Original meaning: Tinned meat product made from ham.
  - Modern usage: unwanted junk email messages. Roughly 20% messages are spam.
  - Origin of modern usage: apparently from sp(iced h)am. The Internet sense appears to derive from a sketch by the British 'Monty Python' comedy group, set in a cafe in which every item on the menu includes spam..
- Basic idea: when words like *Viagra* , *Viagra* occur in an email there is a good chance the email message is spam.
- As a very simple rule, we might think of classifying a message as "spam" if we see the word Viagra in a message. In fact, this is what most people will actually do.
- The quantity ***spamicity*** of a word W can be defined as the quantity ***M(W)*** which measures "what percentage of occurrences of the word ***W,*** appear in spam messages?" Suppose Viagra occurs 100 times in our corpus of emails; 98 times in spam messages and 2 times in non-spam messages, then spamicity(*Viagra* ) = 0.98

# Application 6: Naïve Bayes & Spam Detection

- Naïve Bayes classifier is based on the Bayes Theorem

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B|A) \times P(A) + P(B|A') \times P(A')}$$

- It is amongst the oldest techniques for spam filtering.
- It is often used in conjunction with a Bag of Words model to calculate whether an email is a spam or not.
- We will classify a message as spam if the probability P(Message is Spam | message contains *Viagra* and other suspicious words) exceeds some threshold.
- Naïve Bayes first computes the probability of each word occurring in a legitimate message and also occurring in a spam message.

# Application 6: Naïve Bayes & Spam Detection-2

- Naïve Bayes first computes the probability of each word occurring in a legitimate message and also occurring in a spam message.

$$P(S|W) = \frac{P(W|S) \times P(S)}{P(W|S) \times P(S) + P(W|H) \times P(H)}$$

where:

$P(S|W)$ is the probability that a message is a spam, knowing that the word *Viagra* is in it;

$P(S)$ is the overall probability that any given message is spam;

$P(W|S)$ is the probability that the word *Viagra* appears in spam messages;

$P(H)$ is the overall probability that any given message is not spam (is "ham");

$P(W|H)$ is the probability that the word *Viagra* appears in ham message.

***Note: Not all documents with the word Viagra are spam, for example this one is non-spam!***

# Application 6: Naïve Bayes & Spam Detection-3

- Now suppose we don't know the percentage of messages that are spam; and assume that *Prob(message is spam)= Prob(message is not spam) = 0.5*
- Now you can see that our *Prob(S | W)* is the same as spamicity of the word W, is the same as the quantity M(W) which we defined earlier
- $P(S|W) = \frac{P(W|S)}{P(W|S)+P(W|H))}$

# Application 6: Naïve Bayes & Spam Detection-4

- Worked Example: Suppose we get 20 emails,
  - 10 of which are spam and
    - 8 of them contain the word *Viagra*.
  - The remaining10 emails are non-spam emails,
    - 1 of them contains the word *Viagra*.
- We want to calculate P(spam | *Viagra*) which is the probability that a message is spam, given the word *Viagra* is in it. .
- P(*Viagra* |spam) is the probability that the word *Viagra* is in a spam message, which is 0.8 (8 out of 10 emails)
- P("spam") is the probability that any message is spam. This is 0.5 by assumption (prior probability).
- P(*Viagra* |ham) is the probability that the word *Viagra* is in a ham message, which is 0.1 (1 out of 10 emails)
- P(ham) is the probability that any message is ham. This is 0.5 (prior).
- $P(Spam|Viagra) = \frac{0.8 \times 0.5}{0.8 \times 0.5 + 0.1 \times 0.5}$, which is 0.89

# Application 6: Naïve Bayes and Spam Detection-4

- However, a message will usually contain many words, not just one. Suppose an email message has *M* words, of which *N* words have high spamicity values Let these words be $W_1, W_2, W_3, \ldots, W_N$.
- We would like to estimate
  - *X = Probability(Message is spam | message contains $W_1$ and $W_2$ and $W_3$ and… $W_N$)* and compare it to
  - *Y = Probability(Message is not spam | message contains $W_{N+1}, W_{N+2}, W_M$)*
- By using Bayes' Theorem, *X* can be written as
- $$\frac{P(Message\ is\ spam) \times P(Message\ contains\ W_1, W_2, \ldots, W_N | Mesasage\ is\ spam)}{P(Message\ contains\ W_1, W_2, \ldots, W_N)}$$

- Now comes the "***naïve***" assumption of independence of words.
- Rather than calculate the values of each attribute value (word)
  *U= $P(W_1, W_2, W_3, .. W_N | message\ is\ spam)$*,
  they are assumed to be conditionally independent as
  *U = $P(W_1 | message\ is\ spam) * P(W_2 | message\ is\ spam) *.. P(W_N | message\ is\ spam)$*

# Application 6: Bag-of-Words Model

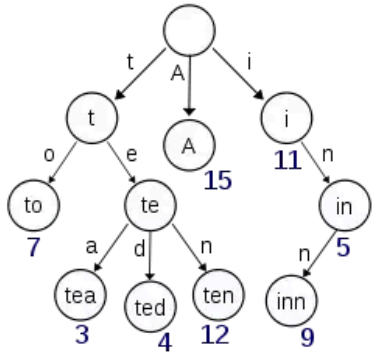
- Bag-of-Words (BoW) is a representation of text data used for document classification and feature extraction
- The task of text modelling is more complicated because Machine Learning cannot work on raw text directly; we need to represent text as numeric vectors.
- We make a vocabulary of known words and then compute the score for each word, that is, each word count is a feature.

# Application 6: Bag of Words-2

- One simple intuition is that two documents are similar if they have similar words.
- We can learn *something* about the of a document by examining these word-scores.
- An n-gram is an n-token sequence of words: a 2-gram or bigram  is a two-word sequence of words like “how are” and a 3-gram  or a trigram  is a three-word sequence like “how are you”.
- It is called a bag-of-words , because any information about the order or structure of words in the document is discarded.
- One limitation of Bag of Words is that word order is not important; thus we cannot infer meaning from context
- For example
  - This is important vs Is this important
  - Good vs “not Good”

# Application 7: Text Prediction and Auto-complete

- There are many software apps which will suggest 4 or 5 choices for the next word, based on a few of the previous characters.
- Traditionally a ***trie*** or prefix tree would be used. *(Figure from Wikipedia)*




A trie for keys "A", "to", "tea", "ted", "ten", "i", "in", and "inn".

(Source: Trie)

# Application 7: Recurrent Neural Networks

- The modern method uses Machine Learning.
- A special type of neural network called Recurrent Neural Network is used.

# Application 8 : Alexa, Siri & Wake Words

- Wake Word Detection: wake word is the word you use to start a conversation with a voice assistant. Example ***ALEXA***
- The echo-dot is always listening for this word.
- The detection of wake words is a branch of AI itself. It is a fairly difficult problem.
  - while the main AI programs of Alexa will run somewhere in the cloud, the wake word detection has to run on the echo dot.
  - Specialised hardware is needed to catch a very small time frame, arrays of microphones might be needed (humans good at this, not computers.)
  - Wake word should be long enough to be distinct, but short and simple so that different speakers do not pronounce it almost the same (siri vs syria)


July 3, 2018

# Time for a Break!

*Manneken Pis, Brussels.*
Designed by Hiëronymus Duquesnoy the Elder and put in place in 1618 or 1619

# Part-2: Technical Discussion

- Welcome back and thanks for returning!
- In this part of the lecture, we will go over a slightly technical introduction to the popular techniques of Machine Learning

# Supervised Learning

- Most of Machine Learning is what is termed as Supervised Learning.
- In supervised learning, for each training example, there are input variables also called predictors and an output variable(s), also called response.
- The goal of the machine learning program is to "learn" the mapping from the input variables to the output variable.
- In other words, given the set of training examples $\{x_i, y_i\}$ the program has to learn the function ***f*** such that $y=f(x)$
- Common examples are regression, neural networks, decision trees

# Unsupervised Learning

- In unsupervised Learning we are not trying to predict something.
- We are trying to discover hidden patterns in the data that can help us identify groups or clusters with in the data
- Clustering is very important in marketing in order to find groups of customers that share common characteristics or exhibit similar purchasing behaviour.
- In medicine, finding patients with  common demographics or biomarkers or DNA is important.
- Examples:
  - Clustering, e.g., k-Means or DBSCAN
  - Dimension Reduction techniques Principal components analysis, Blind Signal Separation (ICA)
  - Association Rules mining
  - Matrix factorization like SVD, SVD++, Nm-negative matrix factorization

# Why is classification important?

- ***Question***: Why is classification useful in ? Why not focus only on prediction using regression ?
- ***Answer***: It is difficult to predict the exact magnitude of a market move. Often, knowing the direction of an asset or knowing whether the next three days will be high volatility or low volatility is enough to trade profitably.

# What is Deep Learning?

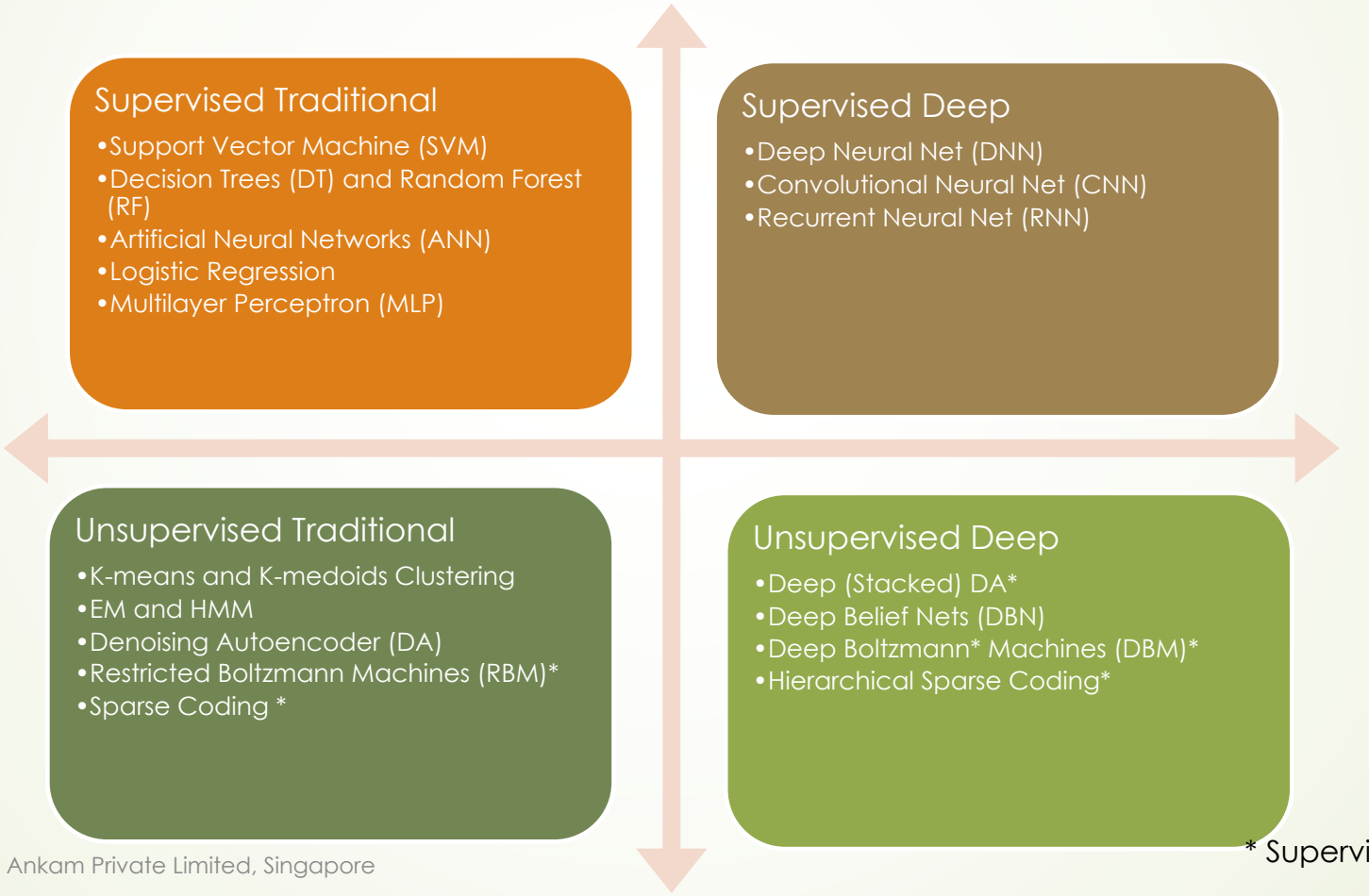
**What is Deep Learning ?**

- It is a paradigm in machine learning in which researchers train computer algorithms to spot meaningful patterns by showing them lots of data, rather than explicitly program the rules
- Neural Nets that mimic the human brain
- Deep architecture to enable predictions or classifications with unseen accuracy
- It enables universal classification: the same model can classify languages, images, videos, audios

**Advantages of Deep Learning**

- Scalable: can scale to billions of parameters to learn complex concepts
- Fast: Model training is fast and a trained model makes online prediction for unseen inputs
- Unified: it brings a unified approach to data-driven knowledge discovery

# Taxonomy of Popular Machine Learning Algorithms






* Supervised version also exists

# Machine Learning Techniques

The most prominent and common algorithms used in machine learning historically and today come in three groups:

**Linear Models:**

- Linear Regression
  - Lasso
  - Ridge Regression
  - Elastic Net
  - Loess
- Logistic Regression
- Support Vector Machines

**Tree-based models**

- Decision Trees
- Random Forests
- XGBoost

**Neural networks**

- Almost everything is now Deep Learning

# *K*-Means Clustering

- It is a heuristic for dividing data into k groups such that members of the same group are closer to each other and as far as possible from the other groups.
- It is a simple scheme, and we only need to choose ***k***, the number of clusters:
  - **Initialize**: First randomly choose ***k*** centroids thus defining the k clusters
  - **Assignment**: Each point's distance from the ***k*** cluster centroids is measured and it is assigned to that cluster to whose centroid is nearest to this point.
  - **Update Centroids**: for each cluster, the centroid is re-calculating by taking the average of all points in the cluster
  - **Repeat Assignment**: each point is re-assigned to the cluster whose centroid is closest to it.
  - **Stop** after reaching a set number of iterations or when the assignments to clusters are stable (centroids stop changing much).

# DBSCAN – Density-Based Spatial Clustering of Applications with Noise

- DBSCAN has two parameters, $\epsilon$ and $\eta$.
- Suppose we are working with *d*-dimension data and there are *N* points.
- Here are the steps in the algorithm:
  1. Around each point, ***p***, make a sphere of radius $\epsilon$ and count the number of points in this sphere. Mark this point as visited
  2. If the no. of points in this sphere > $\eta$ ***then***
     mark p as belonging to a cluster. All the points in the sphere are also marked as belonging to this cluster. For all the points in the sphere that are unvisited, draw a sphere of radius epsilon and repeat this process of counting the no of points, and expanding the cluster ***else*** ignore ***p***

Some points will be left out as belonging to no cluster, and this a very robust scheme and can handle clusters which have strange non-spherical shapes.

# Dimension Reduction

- Sometimes, there are too many many predictors variables (or features).
  - Electronic Health Records : demographic, clinical, and laboratory data
  - If a image is 32 pixels by 32 pixels, it has 1024 pixels, and this is the dimension of 1 KB black and white image.
  - Netflix movie ratings data; each customer rating is a column and each movie is a row. Netflix provided a training **data set** of 100,480,507 ratings that 480,189 users gave to 17,770 movies.
- With high dimensional data, the number of variables $p$ >> number of observations, $N$

# Dimension Reduction: PCA

- Many of these might be redundant and contain no useful information. Some of the predictors might be correlated, or one variable might be a linear combination of some of the other variables.
- Wastage of memory
- Matrix computations become very difficult and time consuming.
- So we want to summarize the information by performing **dimension reduction. Some dimensions might be** useless (e.g., for measuring prob of heart attack, <longitude,latitude,blood pressure>. The first two dimensions are irrelevant and make distances mis-leading.
- One of the most important dimension reduction techniques is called Principal Component Analysis (PCA).

# Dimension Reduction Practical Example

- Usually an object is a three dimensional but we can also represent it by a shadow in two dimensional and in grayscale.

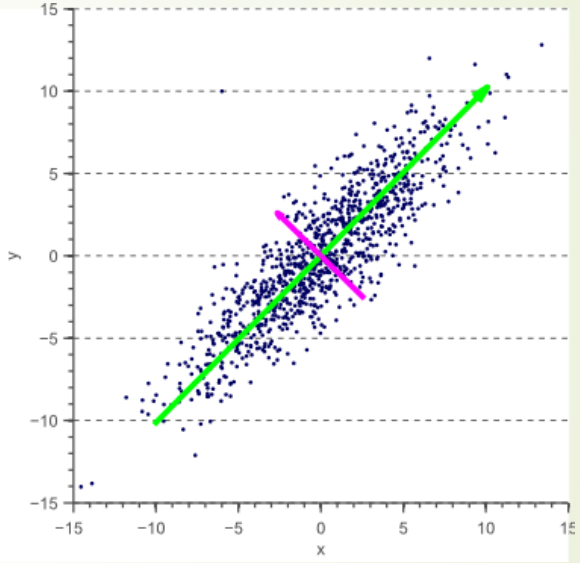
- **A shadow is nothing but a projection from 3-D to 2-D**

- This is an example of **dimension reduction.**
- A realistic, three dimensional model might require100 times more storage as compared to a two dimensional shadow.

- And for some tasks, silhouettes and shadows might be an adequate representation.
- E.g., distinguish a dog from his master.

# Principal Components Analysis

- The figure on the right shows 2 dimensional data,
- For each observation (data point) the first variable is on the x-axis and the second variable is on the y-axis.
- The green line is the first principal component. It explains the largest variation in data.
- The pink line is perpendicular to the green line and is the second principal component.



# PCA for Face Recognition

Basic scheme:

1. Resize each image,
2. Convert to grayscale,
3. Convert to 1-dim vector
   - N*M goes to N*M by 1
4. Normalize
5. Then do a similarity search…

# K-Nearest Neighbours

- *K-Nearest Neighbours* is perhaps the simplest machine learning algorithm (after regressions).
- *K*-NN is a non-parametric technique that can be used to identify historical similar instances. And then we can make predictions by averaging the historical outcomes of the these k-nearest neighbours.
- Almost every one generalizes based on past experience and uses k-NN (perhaps wrongly!) without realizing it.
- K-NN is also called *lazy learning* and it is an example *of instance based learning*.
- One can assign weights to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones. E.g., giving each neighbor a weight proportional to 1/*d*, where *d* is the distance to the neighbour.

# Local Regression : LOWESS, LOESS

- Linear regression under fits, k-NN over fits so perhaps a combination would be better.
- LOWESS stands for **locally weighted scatterplot smoothing.**
- LOESS is a generalization of LOWESS.
- It is a localized linear regression in which we first select *k* nearest neighbours that are similar to our given instance.
- And we then fit a linear regression on the subset of data thus selected.
- We can even use a Lasso regression or Elastic net or Ridge regression on this subset.

# Metrics for Model Evaluation

- Regression models:
  - **Mean Squared Error** is calculated by computing the square of all errors and averaging them over all observations. The lower the MSE, the more accurate were the model's predictions.
  - $R^2$ is the percentage of the observed variance from the mean that is explained (that is, predicted) by the model. $0 <= R^2 <= 1$ and a higher number is better.
- Classification models:
  - ***Accuracy*** is the percentage of observations which were correctly predicted by the model. Accuracy can be misleading when the various classes to predict are unbalanced.
  - ***ROC AUC***, which is a measure of accuracy and stability. ROC stands for "Receiver Operating Characteristic" and AUC stands for "area under the curve". A higher ROC AUC generally means you have a better model.
  - ***Logarithmic loss***, or log loss, is applied when your classification model outputs not strict classifications (e.g., true and false) but class membership probabilities (e.g., a 10 percent chance of being true, a 75 percent chance of being true, etc.). Log loss applies heavier penalties to incorrect predictions that the model made with high confidence.

# Topics in Machine Learning: Cross-Validation Explained

Cross Validation is a **technique for model validation or assessing how our model will perform on new, unseen data instances.**

- The goal of cross validation is to limit overfitting.
  - A model is usually fitted using *known data* (*training dataset*)
  - Then the model is tested against unknown data (*testing dataset*).
  - But we can reserve a part of training data and not use it for training; instead we use this for validation , which means "test during training".

.

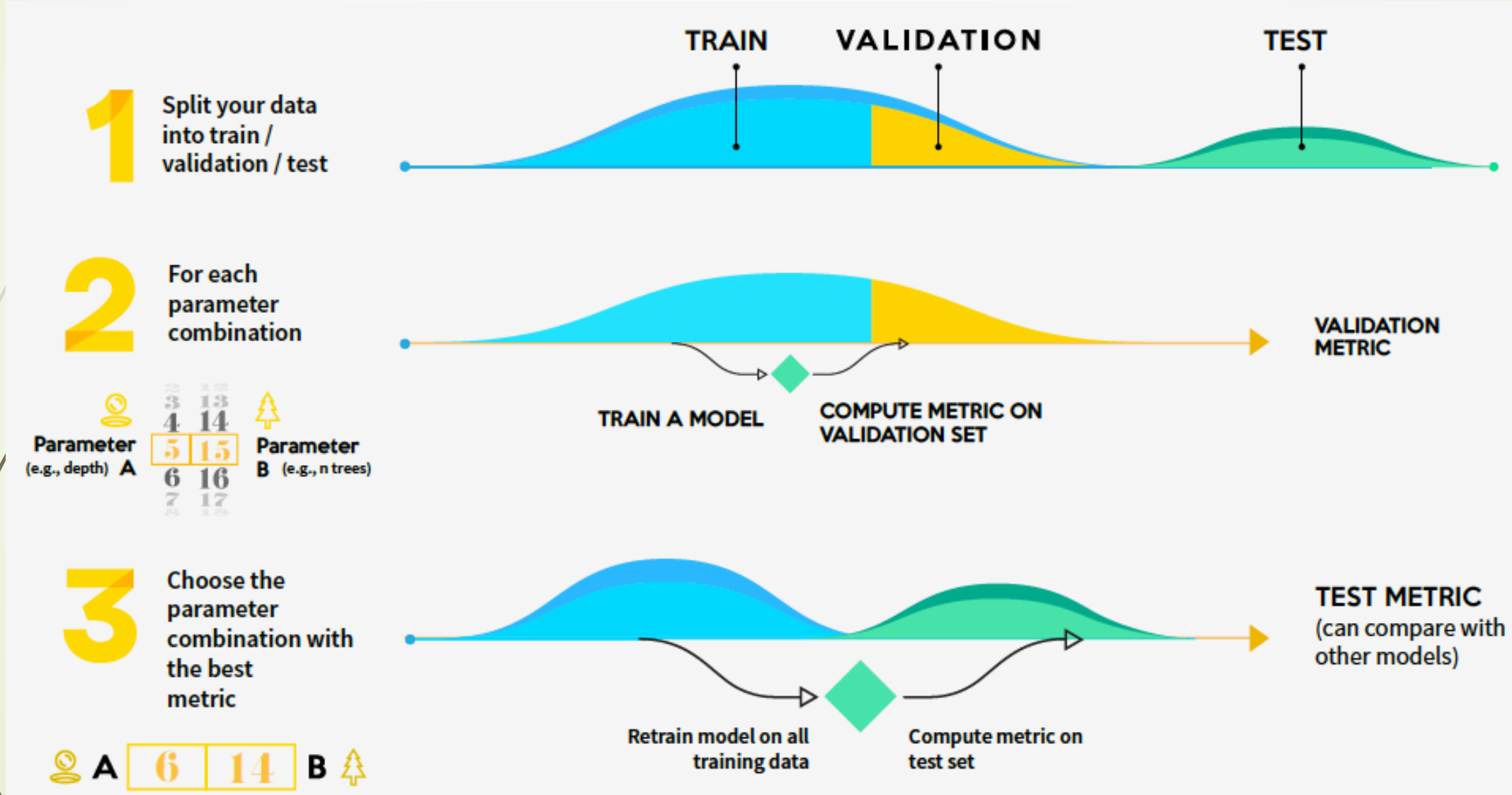
# Leave-one-out Cross Validation & Jackknife

- **Exhaustive cross-validation** methods are cross-validation methods which learn and test on all possible ways to divide the original sample into a training and a validation set.
- The **leave one out cross validation** is similar to the **Jackknife**. The jackknife estimator of a parameter is found by systematically leaving out each observation from a dataset and calculating the estimate and then finding the average of these calculations. Given a sample of size *N*, the jackknife estimate is found by aggregating the estimates of each *N-1* estimate in the sample.
- NOTE: with cross-validation we compute a statistic on the left-out samples, while with jackknifing we compute a statistic from the kept samples only.

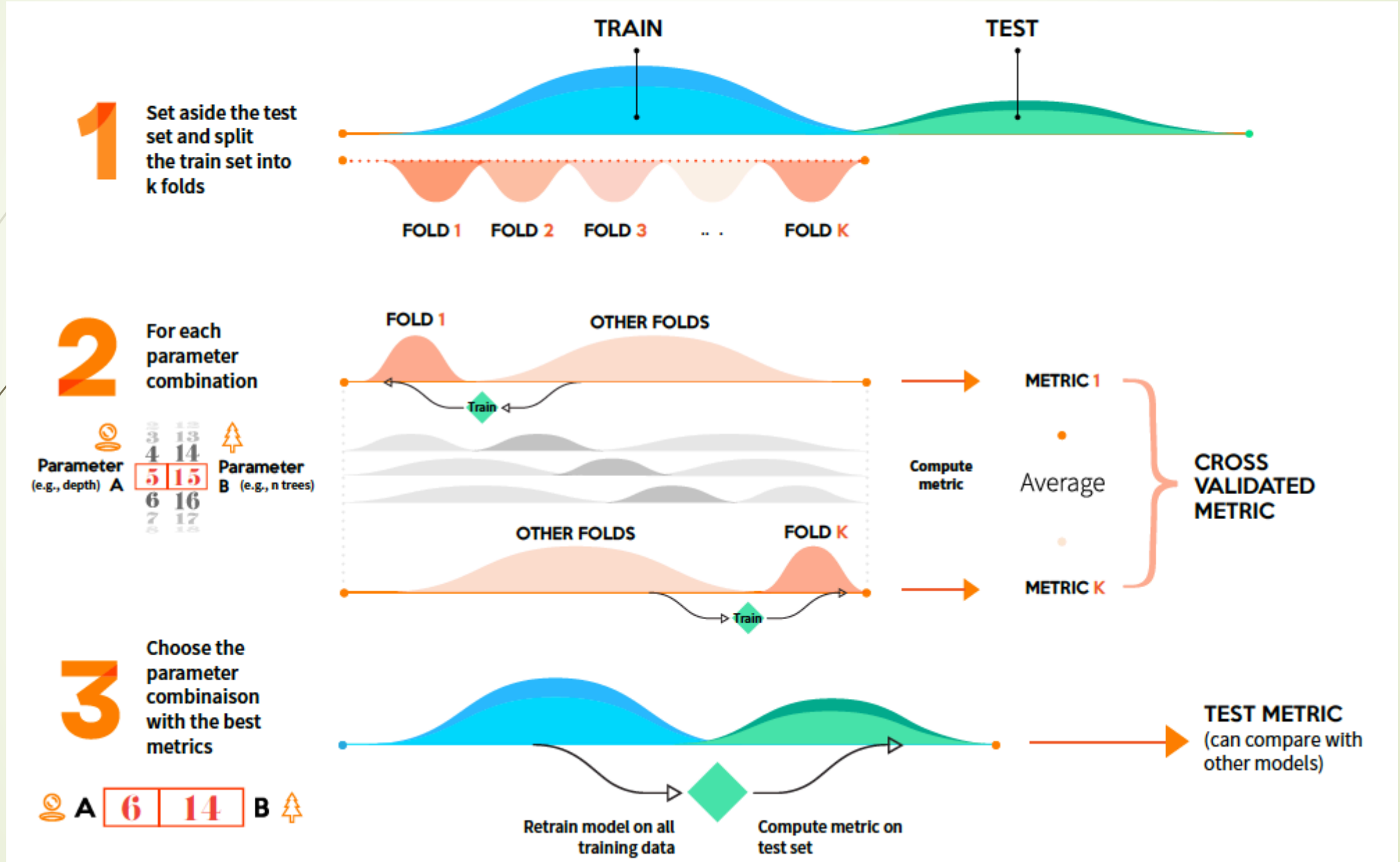
# *k*-fold Cross-Validation

- Non-exhaustive cross validation methods do not compute all ways of splitting the original sample. Those methods are approximations of *leave-p-out cross-validation*.
- For example, one scheme is ***k*-fold Cross-Validation**. In *k*-fold cross-validation, the original sample is randomly partitioned into *k* equal sized subsamples. Of the *k* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining *k* − 1 subsamples are used as training data. The cross-validation process is then repeated *k* times (the *folds*), with each of the *k* subsamples used exactly once as the validation data. The *k* results from the folds can then be averaged to produce a single estimation.
  - All observations are used for both training and validation, and each observation is used for validation exactly once. The most common value f k is 10, resulting in 10-fold.
  - When *k=n* (the number of observations), the *k*-fold cross-validation is exactly the leave-one-out cross-validation.

# Hold Out Strategy

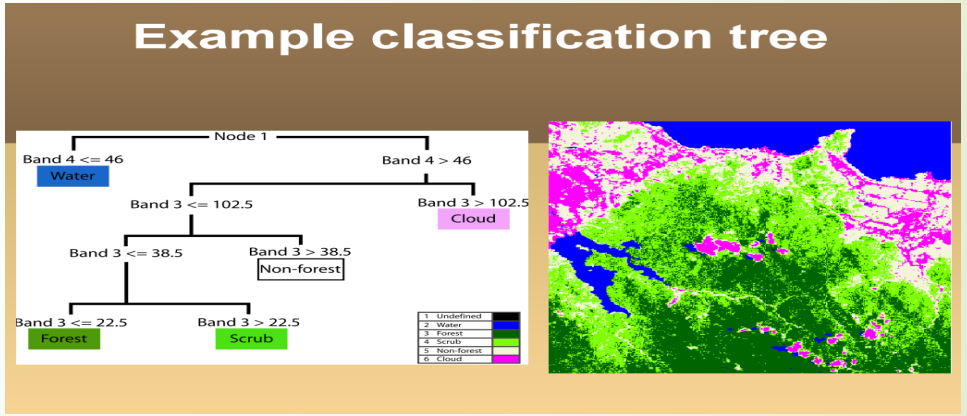


# *k*-fold Strategy



# What is a Decision Tree?

- Decision tree is a predictive model that uses a set of binary rules applied to calculate a target value. It can be used for classification or regression.
  - In Classification, categorical variables are used. E.g., whether a tumor is benign or malignant.
  - In Regression applications, continuous variables are used. E.g., the molecular activity for a compound on a target.
- A recursive tree generating algorithm is used to learn a Decision Tree, and the "best" branching policy is the one that results in the **largest information gain**. To avoid overfitting, pruning is used. Techniques like cross validation can be used for model validation and testing.
- Advantage of decision tree
  - Easy to interpret
  - Robust as regard to outliers in the training data
  - Prediction is fast once rule is developed



# What is a Decision Tree ? Contd.

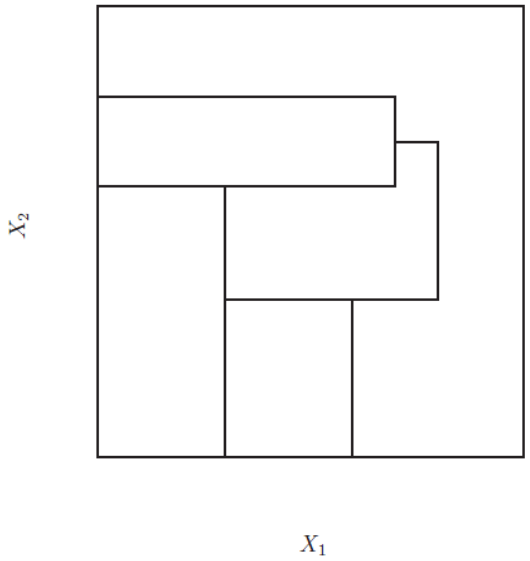
- The goal is minimize the RSS w.

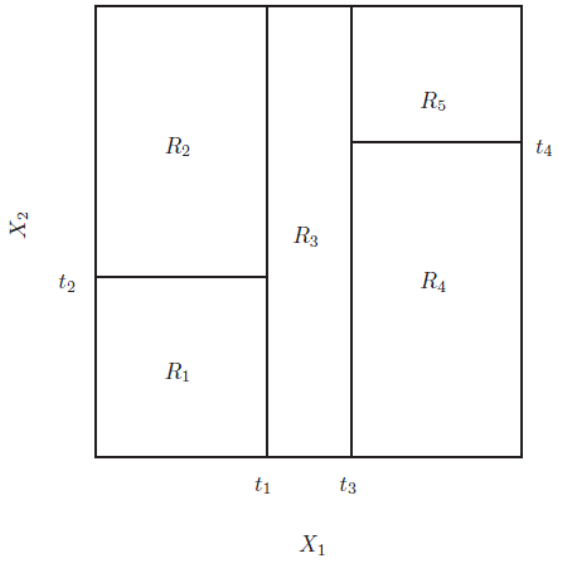
$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

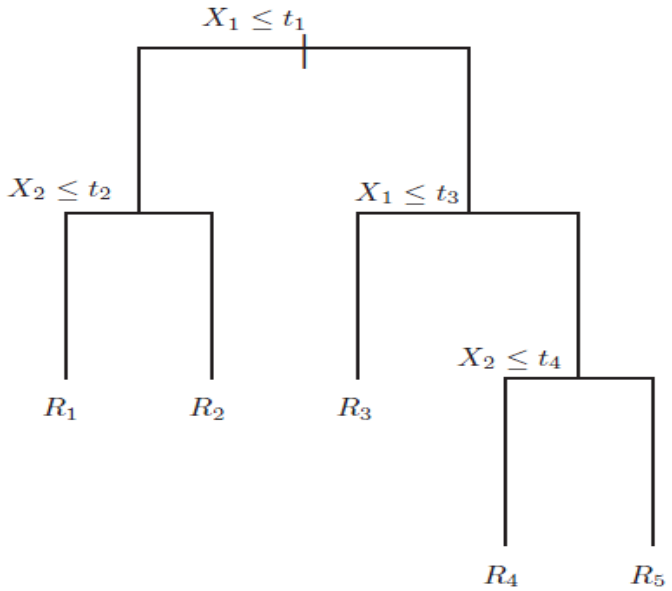
- A *top-down*, *greedy* approach that is known as *recursive binary splitting*.
- The Recursive binary splitting approach is *top-down* because it begins at the top of the tree (at which point all observations belong to a single region) and then successively splits the predictor space; each split is indicated via two new branches further down on the tree.
- It is *greedy* -at each step, the *best* split is made at that particular step, rather than looking ahead. It chooses the **myopically optimal** split: if the learner were only allowed one split, which single split would result in the best classification at the current step?
- The optimal split is the one that gives the maximum **information gain**. Sometimes information gain is used even when the optimality criterion is the sum-of-squares error. An alternative is to use the **Gini index** (a measure of statistical dispersion commonly used measure of inequality, especially income inequality.)

# What is a Decision Tree ? Contd.-2

- The partitioning in the left figure cannot be achieved by Recursive Binary partitioning, where as the middle one can. The associated tree for the middle partitioning is shown in the figure on right



# Algorithm to Create a Decision Tree

1. Use recursive binary splitting to grow a large tree on the training data
2. Stop when each terminal node has fewer than some minimum number of observations.

3. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of a.

4. Use K-fold cross-validation to choose a. That is, divide the training observations into K folds. For each k = 1, . . .,K:

(a) Repeat Steps 1 to 3 on all but the *k-th* fold of the training data.

(b) Evaluate the mean squared prediction error on the data in the left-out *kth* fold, as a function of a.

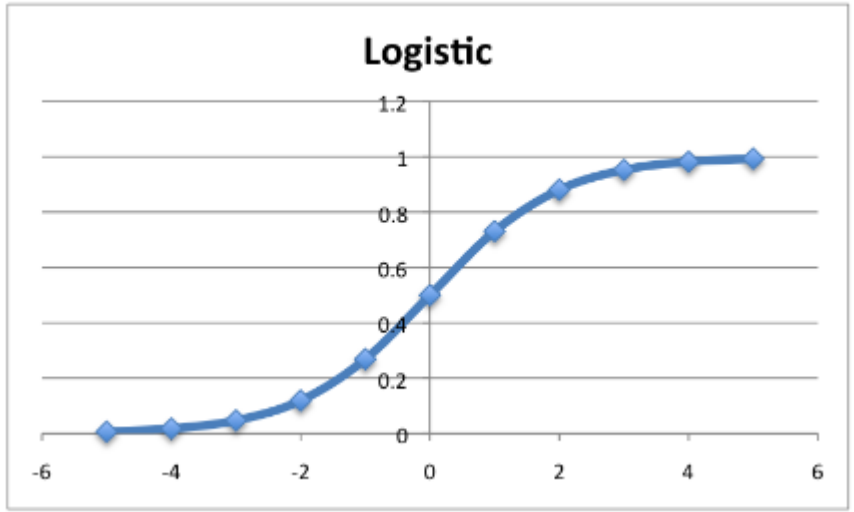
Average the results for each value of a, and pick a to minimize the average error.

5. Return the subtree from Step 3 that corresponds to the chosen value of a.

# What is Logistic Regression?

- Logistic Regression is one of the most basic and important statistical techniques used in Machine Learning.
- It is closely related to neural networks, because the **logistic** function, also called the **Sigmoid** function, is heavily used as the ***activation function*** in Neural Networks. The logistic function is S-shaped and is defined as

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$



# Logistic Regression-contd.

- Logistic function is very useful in modelling probabilities since its range is [0,1]
- Hence it is very useful in Binary Classification as it predicts *p*, the probability of a data point being in class 1. (And therefore also predicts 1, which is *1-p*)
- If we define $t = \beta_0 + \beta_1 X$
- Then the definition of logisitic function becomes

$$\frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

# Softmax Function

- The **Softmax** function is a multinomial generalization of the Logistic function. It is useful when dealing with multi-category classification instead of binary classification. For example, classifying handwritten digits into one of ten classes.
- "squashes" a *K*-dimensional vector **z** of arbitrary real values to a *K*-dimensional vector ***σ(z)*** of real values in the range (0, 1) that add up to 1.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

# Bagging Explained

- Bagging is short form for **Bootstrap Aggregating (bootstrapping** can refer to any test or metric that relies on resampling/random sampling with replacement.)
- This is a general technique used to reduce variance in ensemble learning and to control model overfitting.
- Bagging relies on the fact that averaging a set of observations reduces variance. If we had many training sets, we could build a separate prediction model using each training set, and average the resulting predictions. But we have only one training set. Therefore we create multiple training sets by **sampling uniformly with replacement** from the original training data.
- Decision trees suffer from *high variance*- if we split the training data into two parts at random, and fit a decision tree to both halves, the results that we get could be quite different. By bagging, this can be reduced..Random Forest is an example of Bagging applied to Decision Trees.
- Bagging is most often used in Decision Trees but also in Neural Networks and can be used even in linear regression.

# Boosting Explained

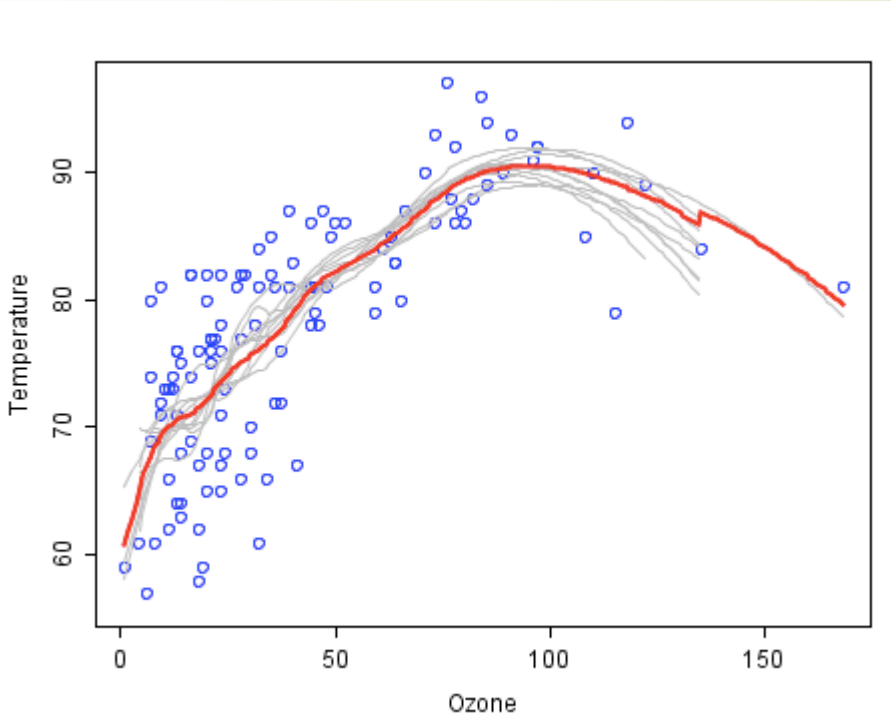
- Boosting is  a paradigm in ensemble  learning to convert weak learners to strong learners.
- Boosting  is the answer (YES!) to the question: Can a set of weak learners create a single strong learner?
  - A weak learner is defined to be a classifier which is only slightly correlated with the true classification; a strong learner is arbitrarily well-correlated with the true classification.
  - Weak learners are combined but they are not all given the same weight- misclassified examples are given higher weights and re-training is performed.
  - Specific  schemes for model averaging using Boosting are ADABoost and BrownBoost
  - Boosting also reduces model variance.

# What is a Random Forest (RF)?

- The **Random Forest** (Breiman, 2001) [1] is an Ensemble approach that uses many Decision Trees (weak learners) to form a refined final classification or regression (strong learner) that is more stable and accurate than all the individual decision trees.

- To train a RF model, a different subset of the training samples is selected ( approx. 2/3 of the original data) with replacement to train each tree. Final classification is made by majority voting and regression is made by tree averaging.

- In contrast to a single Decision Tree, RF is robust to data overfitting and thus no tree pruning is required. It is also robust as regard to outliers in the training data.




The ensemble strong learner (red curve) trained using Random Forest.

# What are Support Vector Machines?

- The Support Vector Machine (SVM) is a technique for classification and regression. Originally the SVM was devised for binary classification, or classifying data into two types. Generalization when there are more than two classes is relatively straightforward.
- For linearly separable data, SVM finds optimal decision boundary using a linear decision surface. When working with non-linearly separable data in the original space, SVM maps the patterns to a higher dimensional feature space in which the transformed data becomes linearly separable. This conversion can be done using kernel function, and the commonly used kernels functions are listed below:

| Name of Kernel Function | Definition |
|---|---|
| Linear | $K(\mathbf{u},\mathbf{v}) = \mathbf{u}^T\mathbf{v}$ |
| Polynomial of degree $d$ | $K(\mathbf{u},\mathbf{v}) = (\mathbf{u}^T\mathbf{v}+1)^d$ |
| Gaussian Radial Basis Function (RBF) | $K(\mathbf{u},\mathbf{v}) = e^{-\frac{1}{2}[(\mathbf{u}-\mathbf{v})^T\Sigma^{-1}(\mathbf{u}-\mathbf{v})]}$ |
| Sigmoid | $K(\mathbf{u},\mathbf{v}) = \tanh[\mathbf{u}^T\mathbf{v}+b]$ |

- Solution to SVM can be formulated as a Quadratic Programming Problem. It can be easily implemented by most of the popular statistical languages (MATLAB, R, etc.) or packages (LibSVM).

# Reinforcement Learning

- Reinforcement Learning is a newer area of Machine Learning.
- Reinforcement Learning is inspired by behavioral psychology.
- Reinforcement Learning models how an agent ought to interact with its environment in order to maximize a cumulative reward.
- The agent learns from the environment by interacting with it and receiving rewards for performing actions; rewards can be negative or positive.
- Very successful in games (AlphaGo which beat the human Go champion in 2016, the Deep Q-network in 2015 which mastered a number of Atari 2600 games to superhuman label with only raw pixels and scores as training inputs).

# Ensembles : Combining Predictions from Different Classifiers

- If predictions from several different models can be combined, the result is usually superior to any of the individual models.
- Creating an *Ensemble* (or group) of classifiers can, and often does, outperform the best individual classifier is akin to averaging and reduces overfitting.
- Even if the classifiers are all similar in performance, an Ensemble classifier can be useful in order to balance out their individual weaknesses and reducing the variance.
- Three different approaches to combining different models are
  - Hard Voting
  - Soft Voting
  - Model Stacking

# Hard Voting or Majority Voting-1

- In Majority Voting, the predicted class label for a particular observation is the class label that represents the majority (mode) of the class labels predicted by each individual classifier. *E.g.*, if the prediction for a given observation is
- Classifier 1 : GOOD
- Classifier 2 : GOOD
- Classifier 3 : BAD
- then the **Ensemble classifier would classify the observation as "good"** based on the majority class label.
- The Majority Vote ensemble shows an error correcting capability.

# Hard Voting or Majority Voting-2

- Suppose there is a sample of 10 observations, and each observation has a label of a 1 or 0 and there are three different classifiers, each with a chance of being correct 70% of the times. If an Ensemble Classifier takes the majority vote among these three classifiers to determine its prediction, it will be correct if at least two classifiers are correct; it will be wrong if only one classifier (or none) is correct.
- A simple calculation shows that
- *Probability (all 3 classifiers are wrong) = $0.3^3 = 0.027$*
- *Probability (exactly 2 classifiers are wrong) = ${}^3C_1 \times 0.3^2 \times 0.7 = 0.1899$*
- *Probability (less than 2 are wrong) = 1 – (0.027 + 0.1899) = 0.7831*
- Hence, taking a majority vote causes the Ensemble accuracy to rise 0.7831.

# Hard Voting or Majority Voting-3

- Hard Voting is ineffective if the individual classifiers are highly correlated.
- Let us take an example of 3 highly correlated classifiers. These classifiers are tasked with predicting the labels of these 10 observations. Assume for the sake of simplicity, that the true label for all of the 10 observations is 1. The three classifiers output the following labels:
- **True Labels**: **11111 11111** (there are 10 one's)
- *Classifier A*: 11111 11100 = 80% correct predictions
- *Classifier B*: 11111 11100 = 80% correct predictions
- *Classifier C*: 10111 11100 = 70% correct predictions
- ***Ensemble***: **11111 11100** = 80% correct predictions
- One can see that the outputs of *A, B, C* tend to be similar; they are wrong at the same time and a majority vote gave the same result as if only A or only B had been deployed.

# Hard Voting or Majority Voting-4

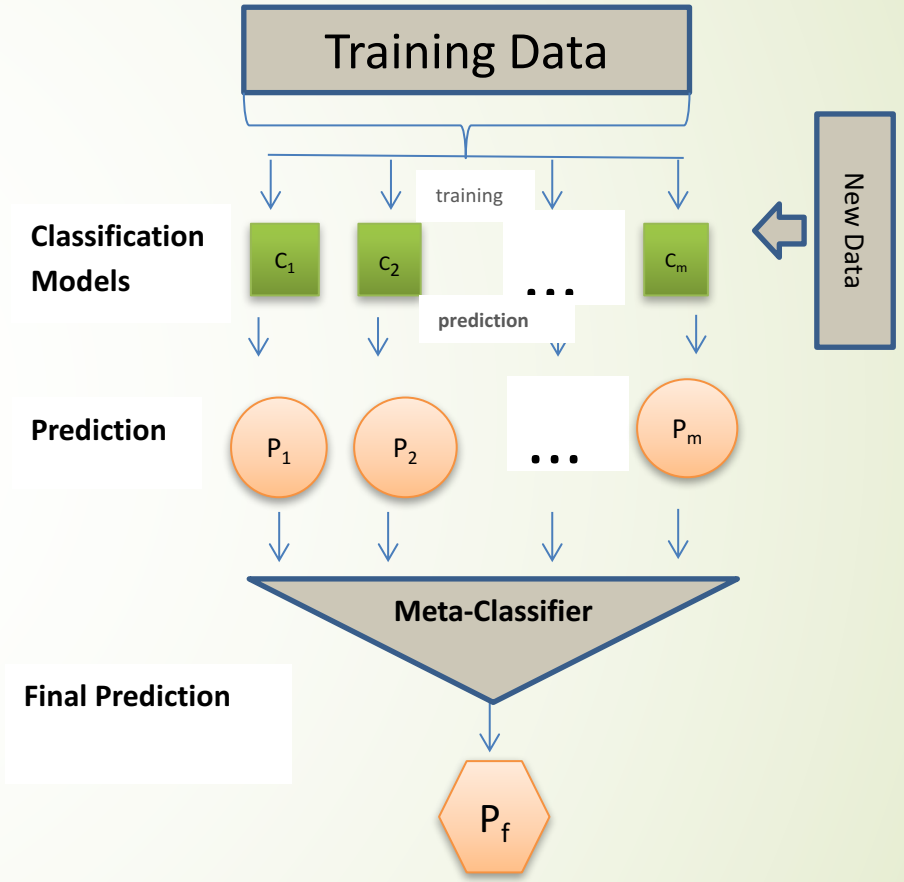
- Now, let us take a look at another set of classifier which are even less accurate than the previous set of classifiers; however, they are also less correlated to each other. This results in their majority vote ensemble to display higher accuracy on the same 10 observations.
- 1111111100 = 80% correct predictions
- 0111011101 = 70% correct predictions
- 1000101111 = 60% correct predictions
- When a majority vote scheme is used to produce an Ensemble of these new three classifiers, the result is:
- 1111111101 = 90% correct predictions, which is better than the best of them individually.

# Soft Voting or Weighted Probability Averaging

- *Soft Voting* returns the class label as *argmax* of the average of predicted probabilities for each class. (*Argmax* of a function is the point(s) at which the function values are maximized.)
- Different classifiers can be given different weights.
- For each class, its predicted probability by each classifier is collected, multiplied by the classifier weight, and averaged.
- The final class label is then derived from the class label with the highest average probability.
- To illustrate this, let's assume we know the probability that each classifier assigns to each class (*good, bad*) for the observation in previous, Majority Voting example.
- Assume that we assign equal weights to all classifiers: $w_1 = 1$, $w_2 = 1$, $w_3 = 1$.

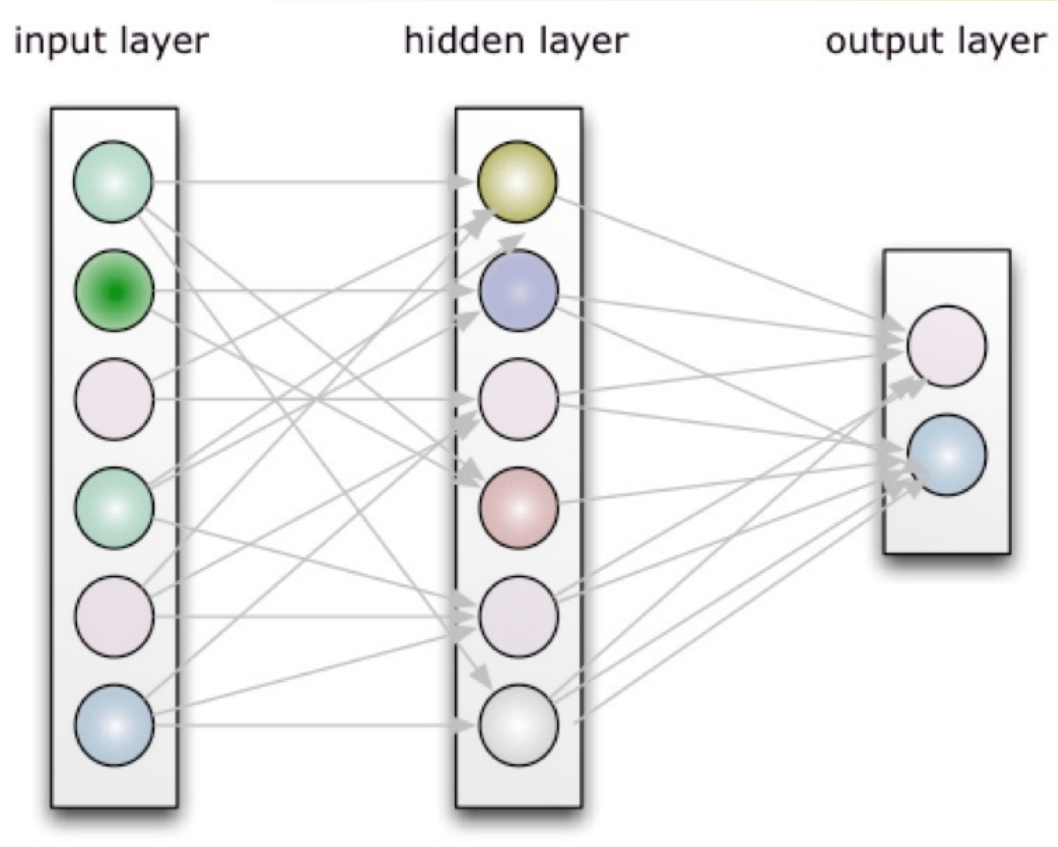
# Model Stacking

- *Model-Stacking* is two stage approach to making predictions.
- First, a number of different classifiers (called *base classifiers*) are trained on the data and their predictions are recorded.
- *Each of these base classifiers can be used for making predictions independently*.
- However, in the model stacking approach, the predictions of these base classifiers are collected and *used as inputs to another algorithm*, which uses these predictions and the original data to make its own prediction.



# What is a Neural Network (NN)?

- Traditional NN uses a feedforward network structure and usually has only one layer. Compared with Deep Neural Network, its structure is simpler and the training is less computationally intensive.
- NN is useful when we have abundance of labeled data but without the knowledge of the underlying mapping function that generates the output. It also shines when data sets are noisy or containing missing variables.
- To train a NN, we first acquire labeled inputs (as high-dimensional vector) and outputs. We then design the structure of the network, such as number of layers and number of neurons in each layer. The formal training process starts with random initialization and feedforward and backpropgation.



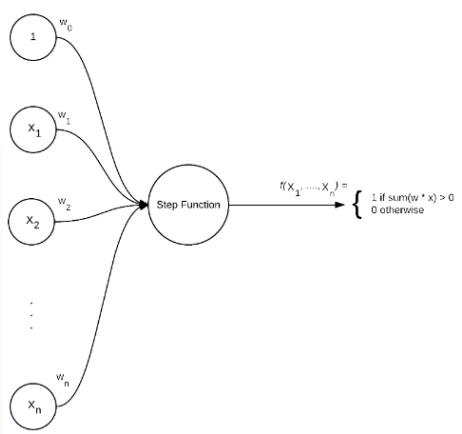
# What is a Neural Network *REALLY* ?

- There has been lot of hype surrounding Neural Networks, including exaggerated comparisons to human brain. This led to very high expectations which were not met, leading to disappointment with Neural Networks and AI for decades.

- Think of a Neural Network simply as a two stage, non-linear statistical model used for classification or regression.

# The Simplest Neural Network: Perceptron

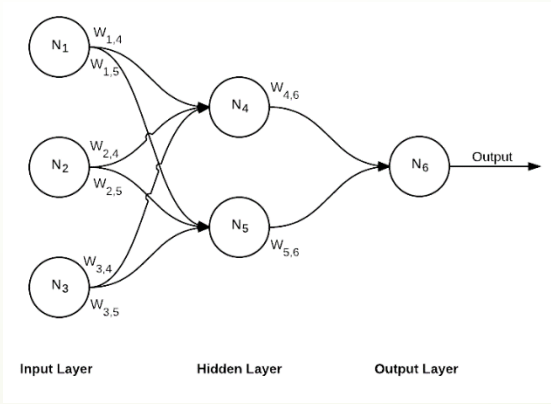
- The simplest neural network is described as a single hidden layer back-propagation network. There are *N* input nodes, one for each entry in the input feature vector, followed by only ***one layer*** in the network with just a ***single node*** in that layer. There exist connections and their corresponding weights, from the input 's to the single output node in the network. This node then takes the weighted sum of inputs and applies a *step function* to determine the output class label. The Perceptron outputs either a *0* or a *1 — 0* for class #1 and *1* for class #2; thus, in its original form, the Perceptron is simply a binary, two-class classifier. Perceptron is a *linear classifier*; it cannot solve non-linear problems such as XOR.



# Multilayer Feed Forward Neural Network

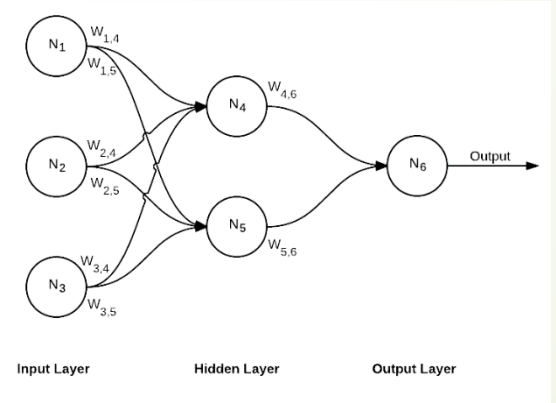
- In order to obtain non-linear separability, we can use ***multi-layer*** feedforward networks with ***non-linear activation functions***.

- A multi-layer feedforward network consists of multiple layers: 1 input layer, *N* hidden layers, and 1 output layer; in our figure we have one input layer, a hidden layer and an output



# Neural Network and Softmax Regression

- In order to obtain non-linear separability, we can use ***multi-layer*** feedforward networks with ***non-linear activation functions***.
- The output of the hidden layer is fed to a Softmax function, as in a multinomial logistic regression function.



# Soft Voting or Weighted Probability Averaging-2

| Classifier | Good | Bad |
|---|---|---|
| Classifier 1 | w1 * 0.55 | w1 * 0.45 |
| Classifier 2 | w2 * 0.60 | w2 * 0.40 |
| Classifier 3 | w3 * 0.25 | w3 * 0.75 |
| weighted average | 0.47 | 0.53 |

The weighted average probabilities for the example

Here, **the predicted class label is *bad*,** since it has the higher average probability.

When hard voting was being used, then both Classifier 1 and Classifier 2 classified this observation as *good* (since both assign higher probability that the observation has class good) and only classifier 3 classified this as *bad*.

The hard-voting ensemble had classified this observation as *good* since that is the majority vote.

Depending on what technique is used to make the ensemble from exactly the same set of classifiers, the final outcome can be very different.

# Gradient Descent and Neural Network Training

- The bottom of the bowl is the minimum loss, or the best set of model parameters or in case of a neural network, the "weights
- The objective is to reach the bottom, taking as few steps as possible…

# The various types of Marketing

- 1. You see a gorgeous girl at a party. You go up to her and say: "I am very rich. "Marry me!" - That's Direct Marketing"

  2. You're at a party with a bunch of friends and see a gorgeous girl. One of your friends goes up to her and pointing at you says: "He's very rich. "Marry him." -That's Advertising"

  3. You see a gorgeous girl at a party. You go up to her and get her telephone number. The next day, you call and say: "Hi, I'm very rich. "Marry me - That's Telemarketing"

  4. You're at a party and see gorgeous girl. You get up and straighten your tie, you walk up to her and pour her a drink, you open the door (of the car) for her, pick up her bag after she drops it, offer her ride and then say: "By the way, I'm rich. Will you "Marry Me?" - That's Public Relations

  5. You're at a party and see gorgeous girl. She walks up to you and says: "You are very rich! "Can you marry ! me?" - That's Brand Recognition

  6. You see a gorgeous girl at a party. You go up to her and say: "I am very rich. Marry me!" She gives you a nice hard slap on your face. - "That's Customer Feedback"

# The various types of Marketing-2

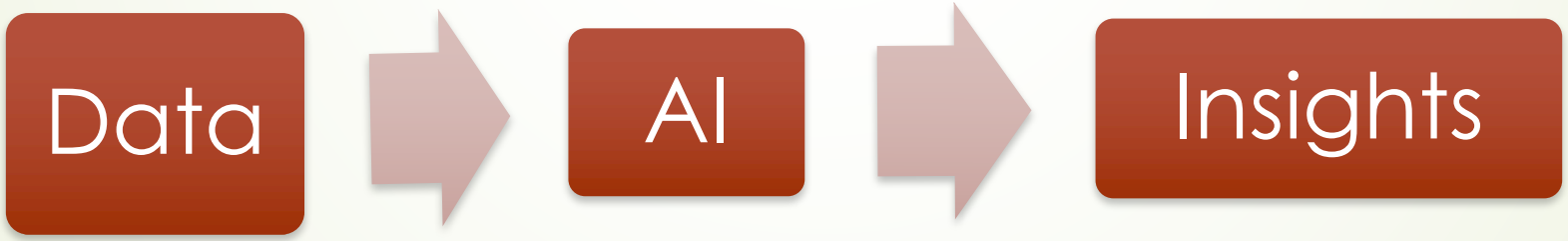
- 7. You see a gorgeous girl at a party. You go up to her and say: "I am very rich. Marry me!" And she introduces you to her husband. - "That's demand and supply gap"

  8. You see a gorgeous girl at a party. You go up to her and before you say anything, another person come and tell her: "I'm rich. Will you marry me?" and she goes with him - "That's competition eating into your market share"

  9. You see a gorgeous girl at a party. You go up to her and before you say: "I'm rich, Marry me!" your wife arrives. - "That's restriction for entering new markets"
- 10. You see a gorgeous girl at a party. You go up to her and find out that she is interested in your profession but does not know its technical details, so you tell her about it. Soon two hours are over, but there is much to talk about yet – so you decide to meet again. After several dates, meeting each other's families you get married and live happily ever after. This is called building a **long term relationship**.
- **We hope each of you will have a long term relationship with AI and Machine Learning.**
- **THANK YOU!**

# Competitive Intelligence: Crayon

- Every company leaves digitial footprints
- These can be tracked, followed and interpreted
- This vast information is difficult to track manually



# Crayon-2

- Details of
  - hiring, firing, downsizing, opening and closing of offices
  - new product launches, product improvements
  - Brand positioning
  - changes in management
  - Customer opinions and reviews
- Natural Language Processing -> Natural Language Understanding-> NL Generation
- Crayon follows over 100 signal such as these and collects information from millions of sources. Then uses AI to gain an understanding from tracking the digital footprints.

# Crayon-3

- Crayon detects activity + what is said online; with some human curation + Machine Learning -> Acts like a digital analyst
- End result is the harvesting of valuable insights from vast amount of data

# Quid

- Quid analyses markets and brand perceptions.
- Web scraping tools can gather information but a layer of AI can help us understand the meaning and content of this information.

# PerfectPrice and Predictive Pricing

- Setting the correct price is an important part of all businesses, because this will optimize revenues.
- We need to use facts and not gut feelings in making decisions.
- There are many readymade solutions for revenue optimization, which do a good job of price prediction but good data is necessary as an input.
- Dynamic pricing is quite mainstream in hotel rooms, plane tickets and taxis.
- Customers are willing to accept dynamic pricing in those products where demand and availability are fluctuating.
- PerfectPrice uses Supervised Machine Learning, Clustering and Reinforcement Learning

# Perfect Price-2

- The product by PerfectPrice generates a demand function that is accurate down to the microsegment level, i.e., specific to very small group of customers.
- In such cases, data about the sale of similar products are gathered over time. For example, when selling used cars, data attributes like type of engine, stereo, mileage clocked colour, owner's neighbourhood, etc.
- A machine learning model will be able to guess the price of the next used car to be sold, even if it has never seen or examined another car with the exact same attributes.
- Regression algorithms and nearest neighbours methods will both work well.

# GoDaddy : Domain Names Appraisal

- Domain name suppliers will have tools for valuing each domain name to be registered.
- Such a tool will use NLP for understanding the meaning of the combination of words in a domain name before coming up with a price

# Programmatic Display of Online Advertisements: AppNexus, Pulsepoint

- AI is well positioned to optimize advertisement investments, especially in the optimization of programmatic display of advertisements.
- AppNexus and Pulsepoint have products that offer insights into the questions:
  - Which advertisers are buying?
  - On which publisher sites?
  - Through which networks and exchanges?
- AppNexus product creates a creates a seamless feedback loop between brand's decision making logic and customer touchpoints.
- AppNexus employs ML to build campaign that become smarter over time, resulting in hyper personalization, enabling marketers to deliver targeted ads to millions of users.

# Pathmatics, MediaMath and Skylads

- Pathmatics offers a product which is like a search engine for ads. Its pipeline for detecting ads, estimating impressions and calculating potential spending uses ML tools that are constantly updated with market data on prices, ad formats, site traffic etc.
- MediaMath has a bidding algorithm for advertisement actions called The Brain and a called Platform Solutions
- Skylads has a product called Skott which works together with MediaMath's algorithm by analysing the performance and fine-tuning it every hour.

# Albert Technologies

- Albert technology offers a ML based virtual assistant called Albert.
- This uses ML to initiates and optimize ad spending across different media channels.
- It measure the results and adapts the ad investments the ROI is changing,
- Albert uses the results of multivariate tests and deep level analysis. Whereas traditional solutions simply analyze the data and wait for a human to make a decision about it, Albert's AI based tools not only analyze the data but also determine the best course of action, execute it and then keeps on re-optimizing its model in real time by learning from new data as it arrives.

# Crobox : AI meets Psychology

- Crobox has a product which combines AI with behavioral psychology for use in e-commerce.
- Companies can gain insights into the WHY behind the BUY.
- This is done by analysis of psychographic data.
- Crobox offering tries to find the best products to high light with persuasive triggers.
- This is like conversion ratio optimization with a layer of psychology overlaid.
- The user can create in depth reports about clients' customers' psychographic profiles throwing light on the sub-conscious motivations of the shoppers and deriving actionable insights into what persuasion tactics work best in triggering online behaviour.

# Marketing Metric

- RFM: Recency, Frequency, Monetary Value
- CLTV: Customer Lifetime Value
- CAC: Customer Acquisition Cost
- Churn Prediction
- Retention by value segment

# Ometria and Retail

- Ometria has a retail-focused product offering.
- AI is used in predicting when a customer is at the risk of never buying again.
- In retail, often a customer is said to be “at risk” if he has not shopped for 12 months successively.
- AI is used to calculate a predictive at-risk score.
- This score is based on all of a customer’s interactions – buying, visiting the website
- Churn prediction is also performed.

# OrderGroove : Predictive Re-ordering

- OrderGroove's product simplifies customer ordering and re-ordering with an app that prompts existing customers to purchase the same products again at the time its prediction algorithms thing the time is right.

# Other Companies with AI offerings

- Equals3's Lucy
- Grammerly – AI for grammar
- Appnexus
- Pulsepoint
- Pathmatics
- AgilOne
- Caliber Mind
- Refuel4
- Spongecell