Machine Learning

Venkatasubramanian R
E-Mail: pvenkat.rv@gmail.com
LinkedIn: https://www.linkedin.com/in/rvsp

Definition & Intro

- Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
- Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Applications of Machine Learning

Finance and Banking

- Credit scoring
- Fraud detection
- Risk analysis
- Client analysis
- Trading exchange forecasting

Retail and E-commerce

- Demand forecasting
- Price optimization
- Recommendations
- Fraud detection
- Customer segmentation

Marketing and Sales

- Market and customer segmentation
- Price optimization
- Churn rate analysis
- Customer lifetime value prediction
- Upsell opportunity analysis
- Sentiment analysis in social networks

Travel and Booking

- Demand forecasting
- Price optimization
- Price forecasting (for dynamically changing prices)

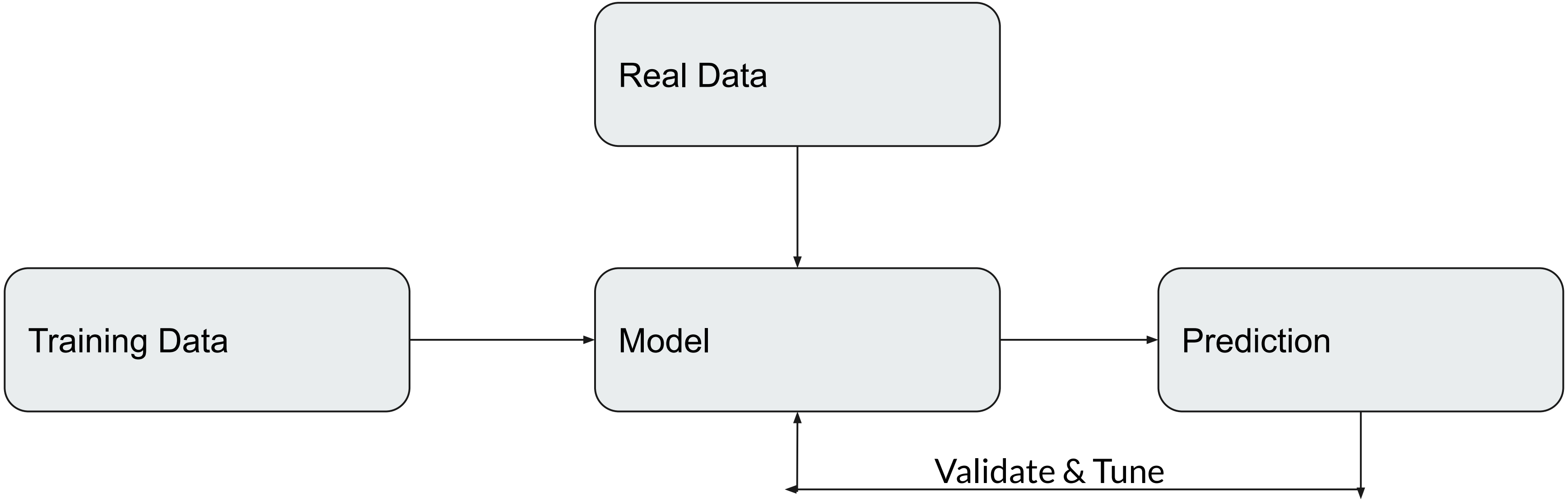
Healthcare and Life Sciences

- Increase in diagnostic accuracy
- Identifying at-risk patients
- Insurance product cost optimization

Other

- Object recognition (photo and video)
- Content recommendations (movies, music, articles and news)
- And more

Working Model



Introduction to Python ML libraries

- **Numpy**
  - NumPy is the fundamental package for scientific computing with Python. It mostly used for solving matrix problems.
- **Pandas**
  - Pandas is the most popular machine learning library written in python, for data manipulation and analysis.
- **Matplotlib**
  - Matplotlib, a great library for Data Visualization.
  - Various types charts (Bar, Funnel, Pie, Line, Maps).
- **SciKit-Learn**
  - A library that provides a range of Supervised and Unsupervised Learning Algorithms. This library mainly focused on model building.

Languages: Python, R, Java, Matlab, Haskell, LISP, JavaScript

Supervised/ unsupervised learning In depth

Supervised learning

- majority of practical machine learning uses supervised learning.
- Supervised learning is where you have input variables (x) and an output variable (Y)
- using an algorithm to learn the mapping function from the input to the output.

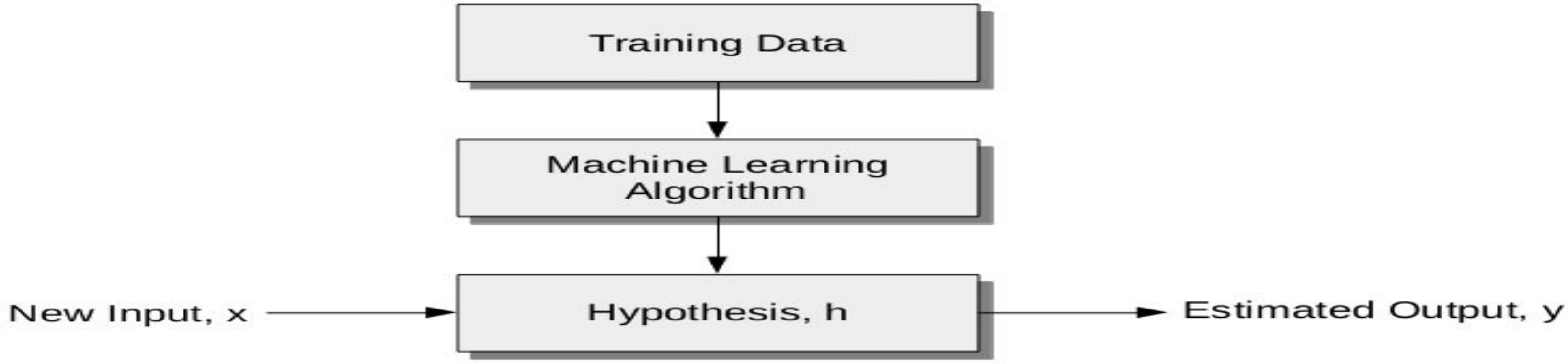
$$Y = f(X)$$

- Supervised learning problems is further grouped into two problems.
    - **Classification:** A classification problem is when the output variable is a category, such as "red" or "blue" or "disease" and "no disease".
    - **Regression:** A regression problem is when the output variable is a real value, such as "dollars" or "weight".

Supervised/ unsupervised learning In depth

Supervised learning (Cont`d)

- Examples
  - Linear regression for regression problems.
  - Random forest for classification and regression problems.
  - Support vector machines for classification problems.



Supervised/ unsupervised learning In depth

Unsupervised Machine Learning

- Unsupervised learning is where you only have input data (X) and no corresponding output variables.
- goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.
- Unsupervised learning problems is grouped into two problems.
  - **Clustering:** A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
  - **Association:** An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.
- examples:
  - k-means for clustering problems.
  - Aprioriate algorithm for association rule learning problems.

Usecase

You’re running a company, and you want to develop learning algorithms to address each of two problems.

Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
Problem 2: You’d like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

Should you treat these as classification or as regression problems?

- Treat both as classification problems.
- Treat problem 1 as a classification problem, problem 2 as a regression problem.
- Treat problem 1 as a regression problem, problem 2 as a classification problem.
- Treat both as regression problems.

Linear Regression

- Regression analysis is one of the most important fields in statistics and machine learning. There are many regression methods available (Single Variate & Multi Variate).
- The dependent features are called the dependent variables, outputs, or responses.
- The independent features are called the independent variables, inputs, or predictors.
- Regression problems usually have one continuous and unbounded dependent variable. The inputs, however, can be continuous, discrete, or even categorical data.
- Regression is used in many different fields: economy, computer science, social sciences, etc.

$Y = mX + b$, where X is the explanatory variable and Y is the dependent variable.

Multiple Linear Regression

Multiple features (variables).

Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Multivariate Exercise:

```
Build a machine learning model to predict salary for the newly appointed
candidates with the help of existing hiring statics (hiring.csv) dataset.

experience of 2 yrs,test score 9 ,interview score  6
experience of 7 yrs,test score 9 ,interview score 10

Answer
53713.86 and 76915.14
```

Logistic Regression

- Logistic regression models the probabilities for classification problems with two possible outcomes(0 or 1).
- It's an extension of the linear regression model for classification problems.

- **Types of Logistic Regression**
  - Binary (Pass/Fail)
  - Multi (Cats, Dogs, Sheep)

Logistic vs Linear

- **Linear Regression** could help us predict the student's test score on a scale of 0 - 100.
- Linear regression predictions are continuous (numbers in a range).
- linear regression model can work well for regression, but fails for classification

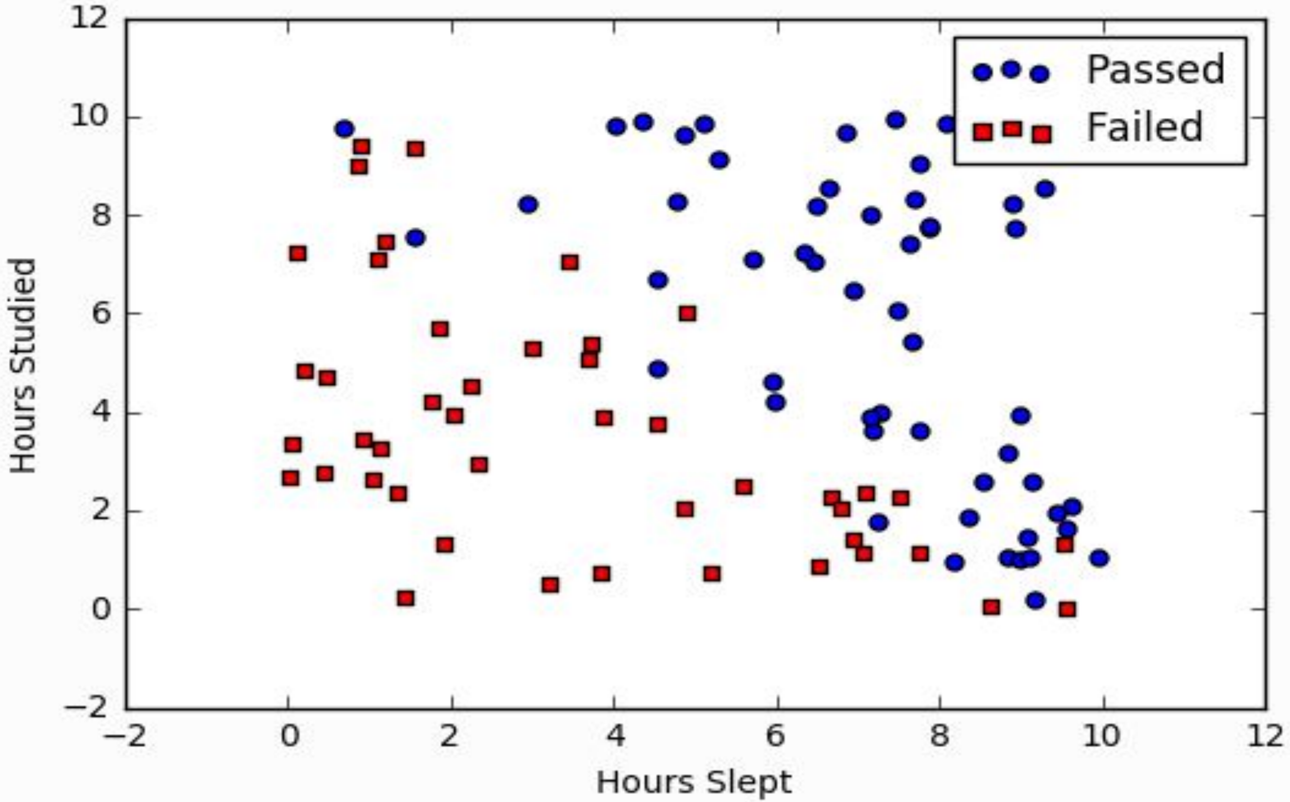
- **Logistic Regression** could help use predict whether the student passed or failed.
- Logistic regression predictions are discrete (only specific values or categories are allowed).

Logistic Regression(Cont`d)

- For example: student exam results and our goal is to predict whether a student will pass or fail based on number of hours slept and hours spent studying. We have two features (hours slept, hours studied) and two classes: passed (1) and failed (0).

Studied	Slept	Passed
4.85	9.63	1
8.62	3.23	0
5.43	8.23	1
9.21	6.34	0

Graphically we could represent our data with a scatter plot.



Logistic Regression(Cont`d)

Sigmoid activation

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

Math

$$S(z) = \frac{1}{1 + e^{-z}}$$

Note

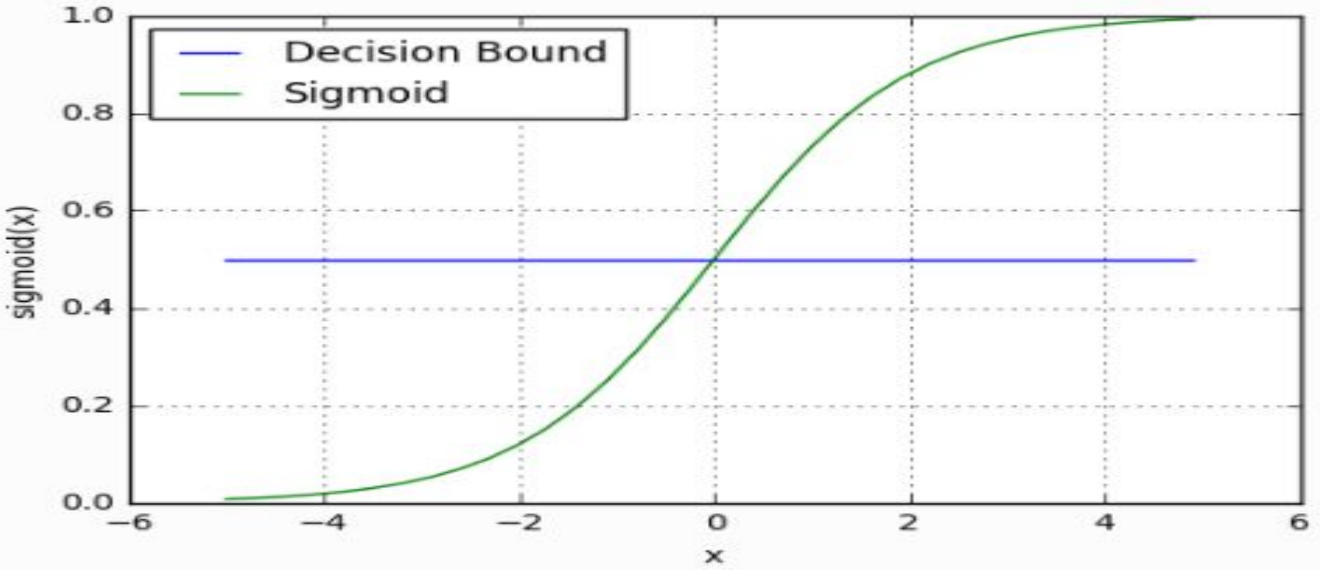
- $s(z)$ = output between 0 and 1 (probability estimate)
- z = input to the function (your algorithm's prediction e.g. mx + b)
- e = base of natural log

Decision boundary

Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

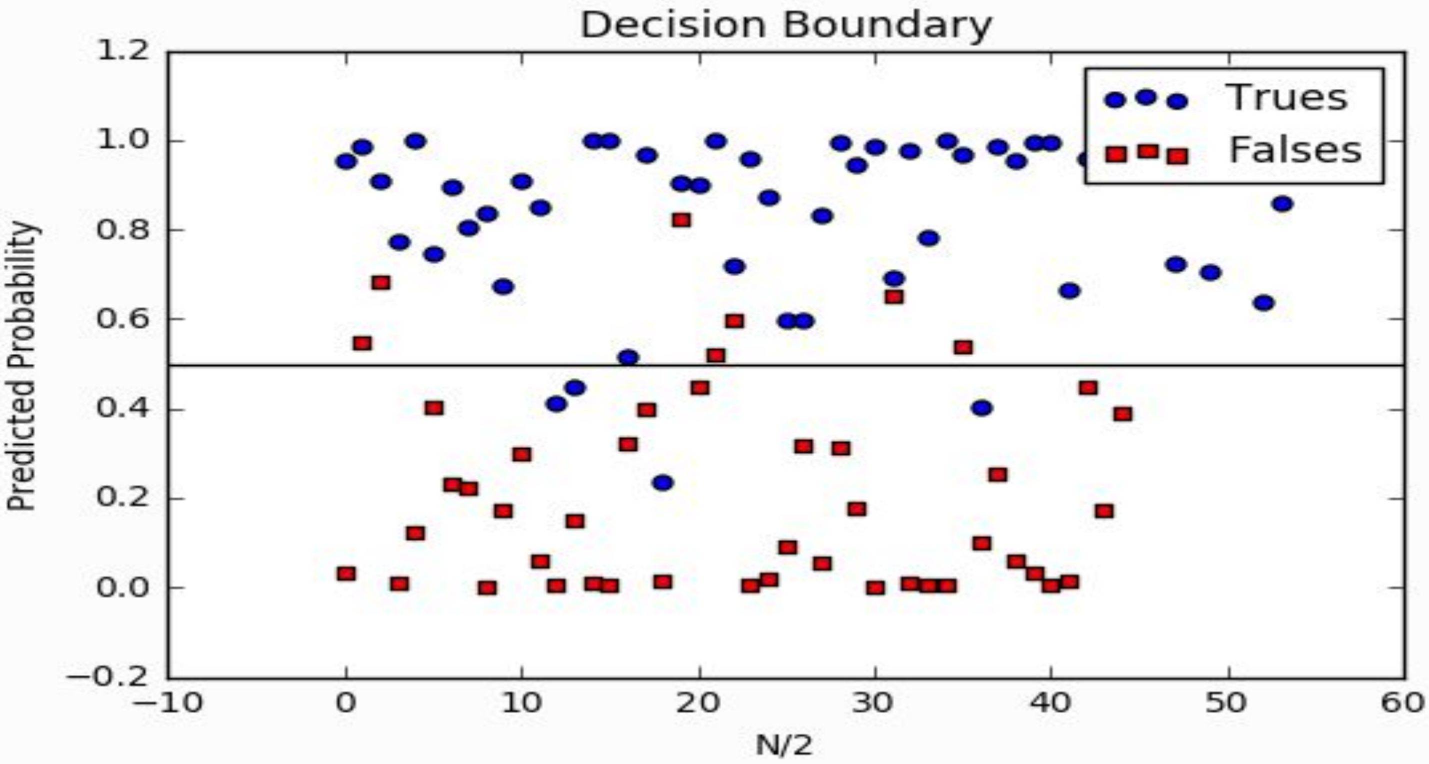
$$p \geq 0.5, class = 1$$
$$p < 0.5, class = 0$$

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction was .2 we would classify the observation as negative. For logistic regression with multiple classes we could select the class with the highest predicted probability.



Decision boundary

Another helpful technique is to plot the decision boundary on top of our predictions to see how our labels compare to the actual labels. This involves plotting our predicted probabilities and coloring them with their true labels.



Classification

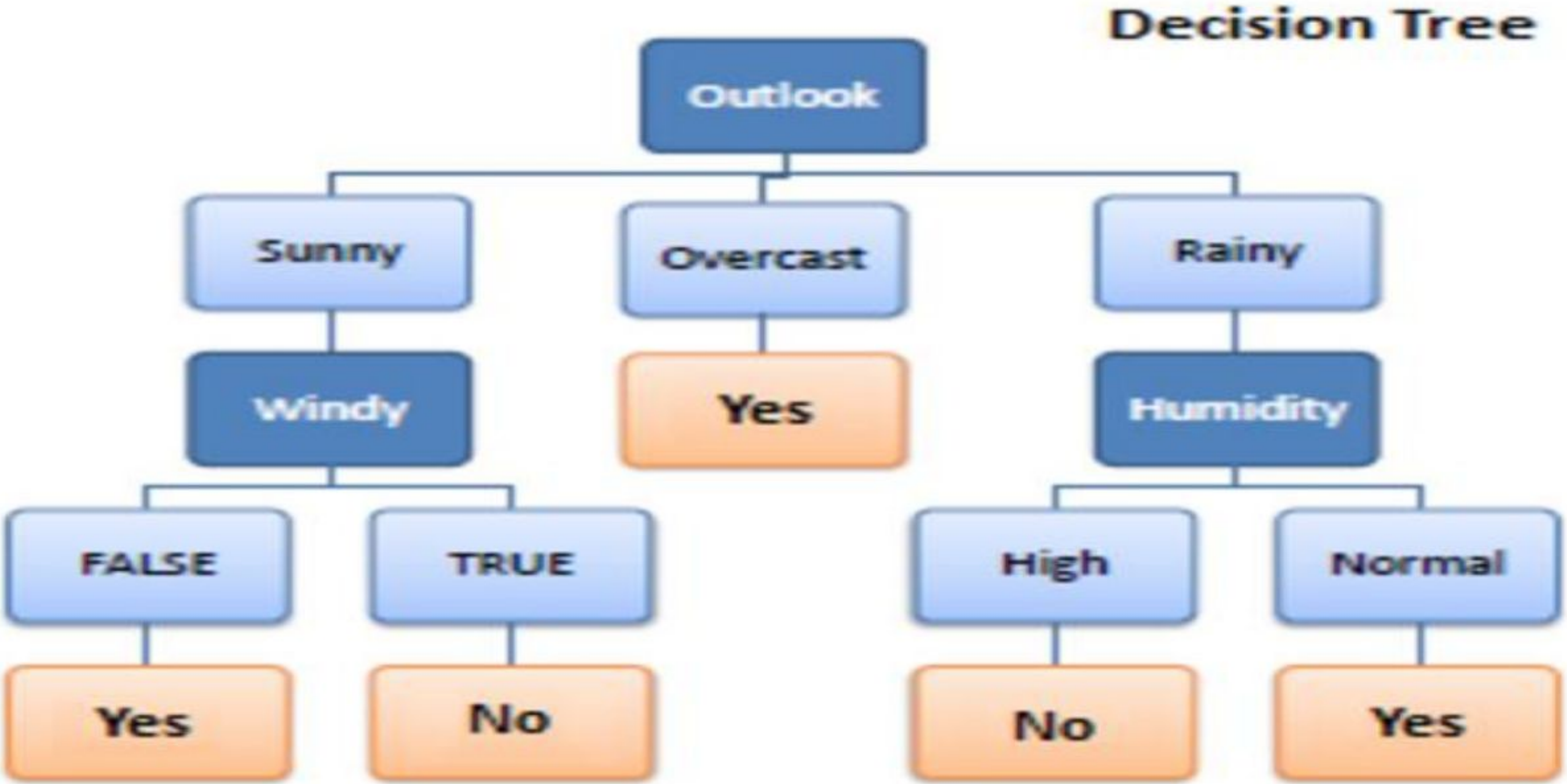
- Logistic Regression : used to predict a binary outcome (1 / 0, Yes / No, True / False)
    - E-Mail: spam / not spam
    - Online Transfer: Fraudulent(yes/no)
- Naive Bayes Classifier : classify a text or a review into either “positive” or “negative”
- Nearest Neighbor
- Support Vector Machines :predicted to belong to a category based on which side of the gap they fall (space data)
- Decision Trees : categorical and numerical data.(Handling late night cravings-> 1 / 0, Yes / No, True / False)
- Random Forest : Decision Trees
- Neural Networks

Decision Tree

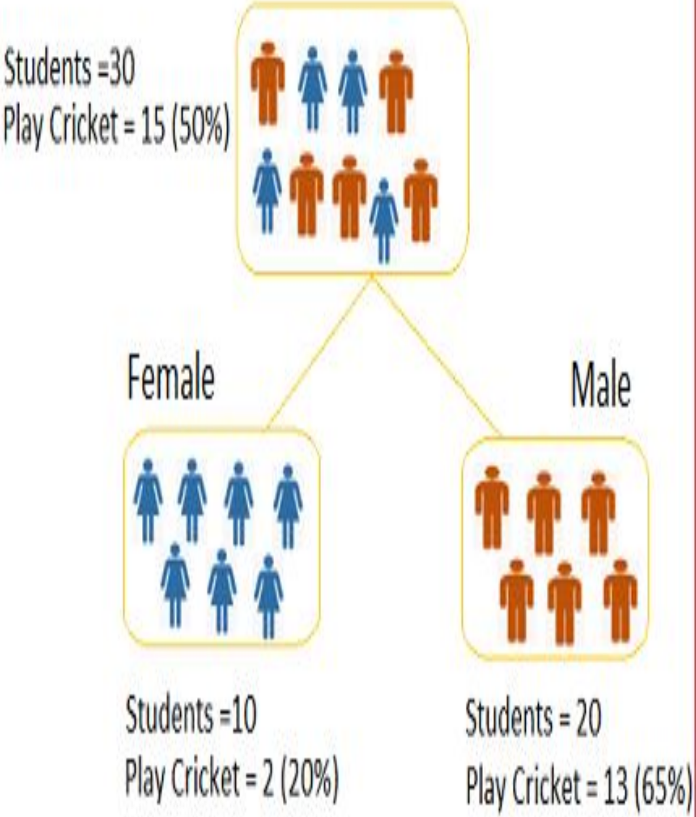
Decision tree identifies the most significant variable and it’s value that gives best homogeneous sets of population

Now the question which arises is, how does it identify the variable and the split?

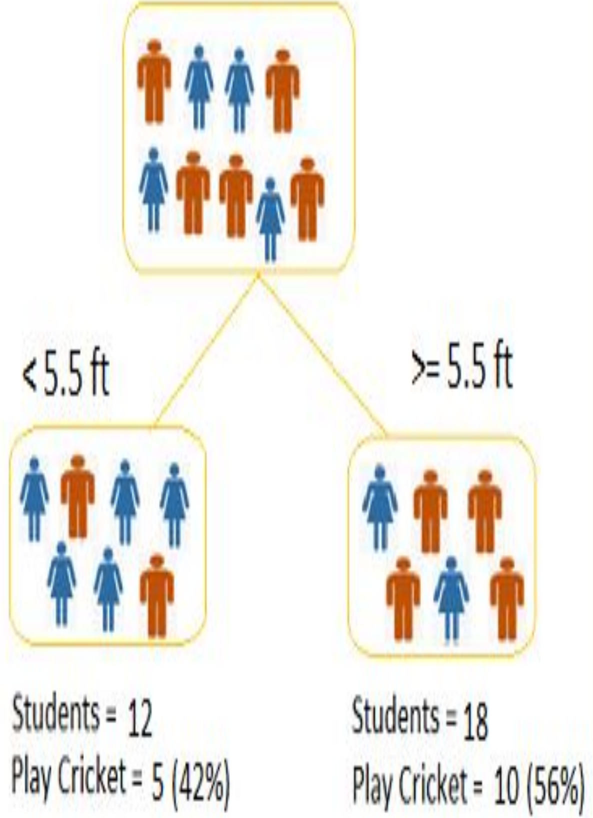
- Categorical Variable Decision Tree
- Continuous Variable Decision Tree


Decision Tree
Outlook
Sunny
Overcast
Rainy
Windy
Yes
Humidity
FALSE
TRUE
High
Normal
Yes
No
No
Yes

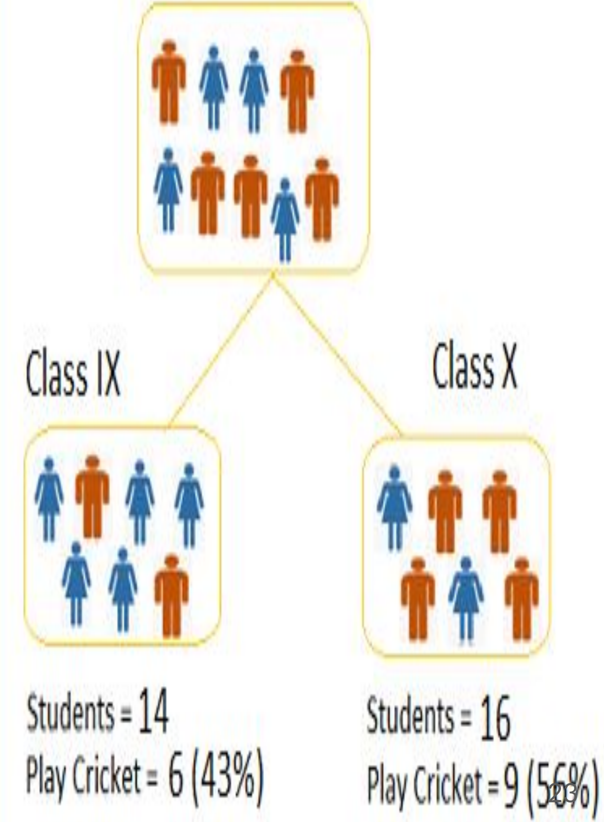
Split on Gender



Split on Height



Split on Class



Decision Tree (cont`d)

Gini Index: Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen categorical target variable “Success” or “Failure”.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Entropy: A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous)

$$Entropy = -p \log_2 p - q \log_2 q$$

Decision tree Advantages

1. Easy to Understand.
2. Useful in Data Exploration - Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables.
3. Less Data Cleaning is Requires - It is not influenced by outliers and missing values to a fair degree.
4. Data type not a constraint - It can handle both numerical and categorical variables.
5. Non-Parametric Method

Decision Tree Disadvantages

1. Over fitting: Overfitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning.
2. Not fit for continuous variables: While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories.

Support Vector Machine

- Support Vector Machines are based on the concept of decision planes that define decision boundaries.
- A decision plane is one that separates between a set of objects having different class memberships.
- SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables

Principal Component Analysis

- Principal Component Analysis (PCA) is a unsupervised learning called **linear dimensionality reduction** technique that can be utilized for extracting information from a high-dimensional space by projecting it into a lower-dimensional sub-space.
- Dimensions are nothing but features that represent the data. For example, A 28 X 28 image has 784 picture elements (pixels) that are the dimensions or features which together represent that image.

PCA (Cont`d)

- Principal components have both direction and magnitude. The direction represents across which *principal axes* the data is mostly spread out or has most variance and the magnitude signifies the amount of variance that Principal Component captures of the data when projected onto that axis.
- Major 2 usecases:
  - Data visualization
  - Speeding ML algorithm

PCA Limitations

Model performance: PCA can lead to a reduction in model performance on datasets with no or low feature correlation or does not meet the assumptions of linearity.

Classification accuracy: Variance based PCA framework does not consider the differentiating characteristics of the classes. Also, the information that distinguishes one class from another might be in the low variance components and may be discarded.

Outliers: PCA is also affected by outliers, and normalization of the data needs to be an essential component of any workflow.

Interpretability: Each principal component is a combination of original features and does not allow for the individual feature importance to be recognized.

Natural Language Processing

- Natural language processing (NLP) is about developing applications and services that are able to understand human languages. Some Practical examples of NLP are speech recognition for eg: google voice search, understanding what the content is about or sentiment analysis etc.

NLP Implementations

- **Search engines** like Google, Yahoo, etc. Google search engine understands that you are a tech guy so it shows you results related to you.
- **Social websites** feed like the Facebook news feed. The news feed algorithm understands your interests using natural language processing and shows you related Ads and posts more likely than other posts.
- **Speech engines** like Apple Siri.
- **Spam filters** like Google spam filters. It’s not just about the usual spam filtering, now spam filters understand what’s inside the email content and see if it’s a spam or not.

Introduction to Machine Learning

What is Big Data?

1. **Volume.** Organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies (such as Hadoop) have eased the burden.
2. **Velocity.** Speed at which data is being generated
3. **Variety.** all formats – structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

BIG DATA! – What happens in 1 minute

500 million tweets **every day**, **100 million** active users

YouTube users upload **500 hours** of video **every minute**

29 Million messages are being sent in WHATSAPP in every 60 seconds

C:\Datasets_BA\Demo's\Aamazon go.mp4

306 items are purchased **every second**
26.6 Million transactions **per day**

100 terabytes of data uploaded **daily**
http://www.dnaindia.com/scitech/report-facebook-saw-one-billion-simultaneous-users-on-aug-24-2119428

Processing **100 petabytes a day (1 petabyte = 1000 terabytes)**

More than **1 million** customer transactions **every hour**
2.5 petabytes of data collected **per hour**

https://www.techinasia.com/alibaba-crushes-records-brings-14.3-billion-singles-day

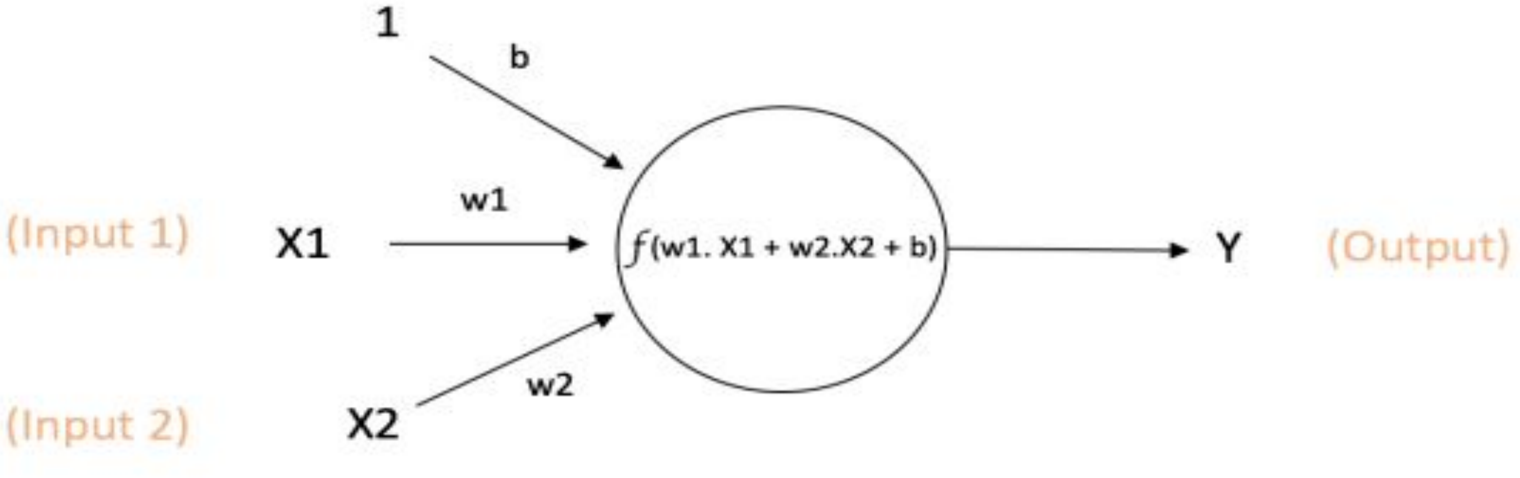
Introduction to Neural Networks

An Artificial Neural Network (ANN) is a computational model that is inspired by the way biological neural networks in the human brain process information.

In Machine Learning research and industry ANN have generated a lot of excitement.

Here i take Single Neuron as example:

- unit of computation in a neural network is the **neuron,** often called a **node** or **unit**.
- receives input from some other nodes, or from an external source and computes an output
- Each input has an associated **weight** (w)



Output of neuron $= Y = f(w1.\ X1 + w2.X2 + b)$

- another input **1** with weight **b** (called the **Bias**)
- **X1, X2,..** are numerical inputs
- **W1, w2,..** are weights
- **Y** is output
- ***f*** is **Activation Function**

Neural Networks(Cont`d)

There are several activation functions you may encounter in practice:

- **Sigmoid:** takes a real-valued input and squashes it to range between 0 and 1

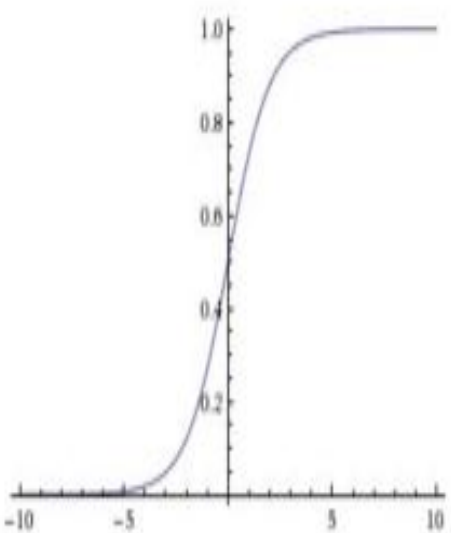
$$\sigma(x) = 1 / (1 + \exp(-x))$$

- **tanh**: takes a real-valued input and squashes it to the range [-1, 1]

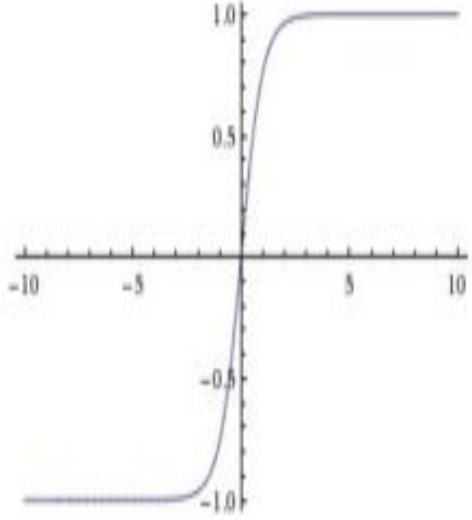
$$\tanh(x) = 2\sigma(2x) - 1$$

- **ReLU**: ReLU stands for Rectified Linear Unit. It takes a real-valued input and thresholds it at zero (replaces negative values with zero)

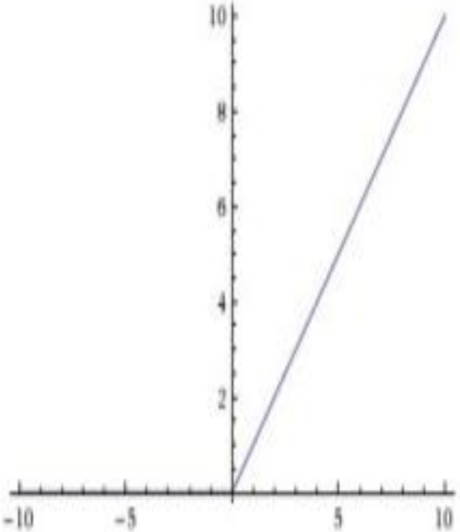
$$f(x) = \max(0, x)$$



Sigmoid



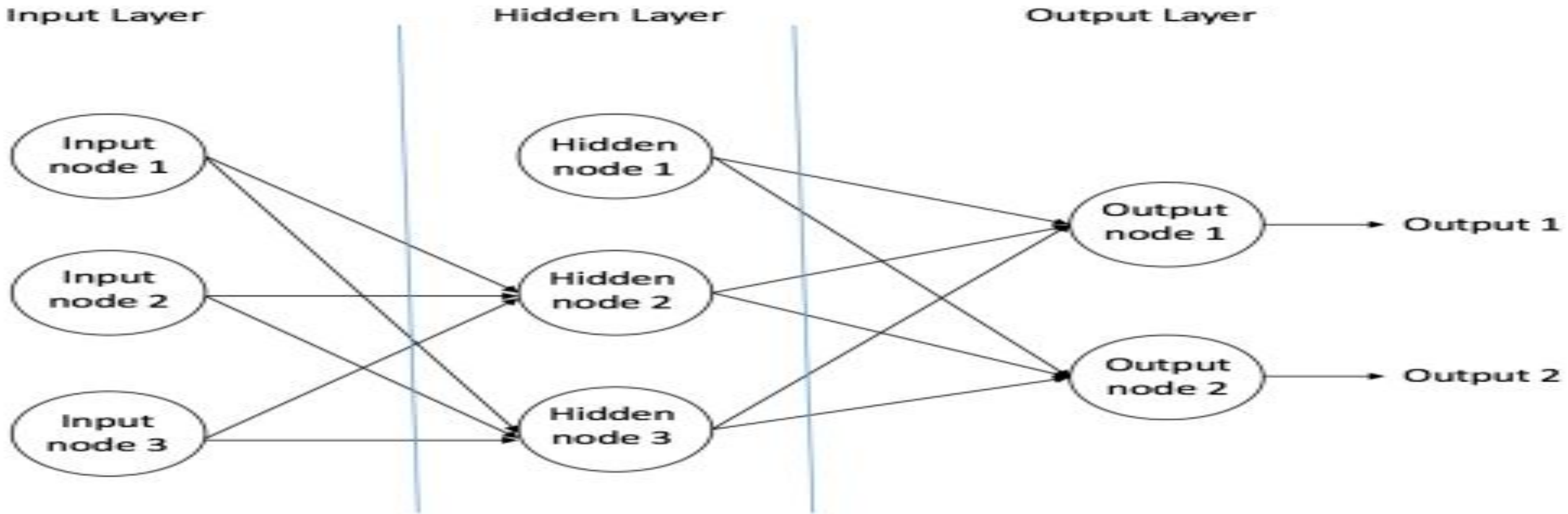
tanh



ReLU

Neural Networks(Cont`d)

- **Forward Neural Network**
  - feedforward neural network was the first and simplest type of artificial neural network
  - It contains multiple neurons (nodes) arranged in **layers**
  - Nodes from adjacent layers have **connections** or **edges** between them
  - All these connections have **weights** associated with them.



Neural Networks(Cont`d)

- Three types of nodes:
  - **Input Nodes –** The Input nodes provide information from the outside world to the network and are together referred to as the "Input Layer". No computation is performed in any of the Input nodes. They simply pass information to hidden layer.
  - **Hidden Nodes –** The Hidden nodes have no direct connection with the outside world. They perform computations and transfer information to the output nodes. A collection of hidden nodes forms a "Hidden Layer". While a feedforward network will only have a single input layer and a single output layer, but it can have zero or multiple Hidden Layers.
  - **Output Nodes –** The Output nodes are collectively referred to as the "Output Layer" and are responsible for computations and transferring information from the network to the outside world.

Note: The information moves in only one direction – forward – from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

Neural Networks(Cont`d)

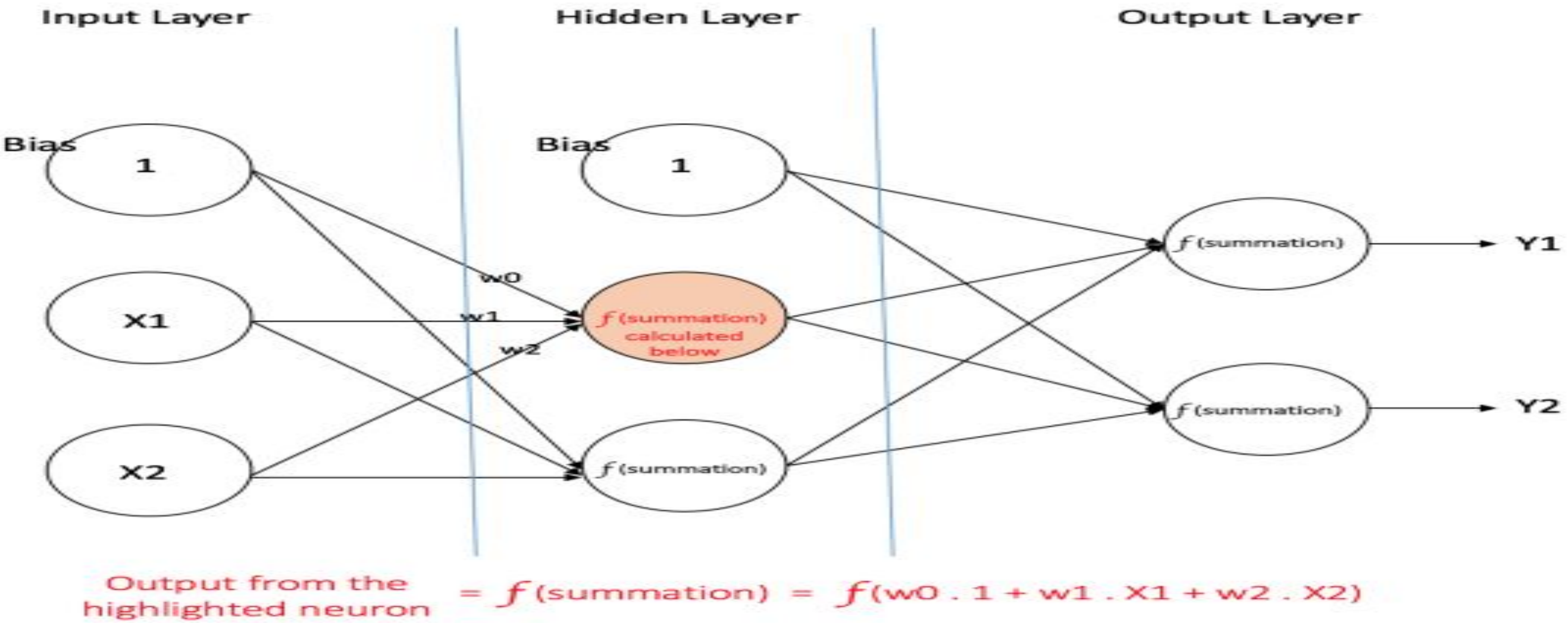
Two examples of feedforward networks

- **Single Layer Perceptron** – This is the simplest feedforward neural network and does not contain any hidden layer. The single layer perceptron can only learn linear functions
- **Multi Layer Perceptron** – A Multi Layer Perceptron has one or more hidden layers.

Neural Networks(Cont`d)

Multi Layer Perceptron:

- A Multi Layer Perceptron (MLP) contains one or more hidden layers and it can also learn non - linear functions.



Multi Layer Perceptron: (Cont`d)

- **Input Layer**:
  - Input layer has three nodes.
  - Bias node has a value of 1.
  - other two nodes take X1 and X2 as external inputs (numerical values).
  - No computation is performed in the Input layer, so the outputs from nodes in the input layer are 1, X1 and X2 respectively, which are fed into the Hidden Layer.
- **Hidden Layer:**
  - Hidden layer also has three nodes with the Bias node having an output of 1.
  - The output of the other two nodes in the Hidden layer depends on the outputs from the Input layer (1, X1, X2) as well as the weights associated with the connections (edges).
  - Previous Figure shows the output calculation for one of the hidden nodes.
  - Similarly, the output from other hidden node can be calculated.
  - Remember that ***f*** refers to the activation function.
  - These outputs are then fed to the nodes in the Output layer.

Multi Layer Perceptron: (Cont`d)

- **Output Layer:**
  - The Output layer has two nodes which take inputs from the Hidden layer and perform similar computations as shown for the highlighted hidden node.
  - The values calculated (Y1 and Y2) as a result of these computations act as outputs of the Multi Layer Perceptron.

Example:

Let's take an example to understand Multi Layer Perceptrons better. Suppose we have the following student-marks dataset:

Hours Studied	Mid Term Marks	Final Term Result
35	67	1 (Pass)
12	75	0 (Fail)
16	89	1 (Pass)
45	56	1 (Pass)
10	90	0 (Fail)

Multi Layer Perceptron: (Cont`d)

Hours Studied	Mid Term Marks	Final Term Result
25	70	?

- Two input columns show the number of hours the student has studied and the mid term marks obtained by the student.
- The Final Result column can have two values 1 or 0 indicating whether the student passed in the final term.
- For example, we can see that if the student studied 35 hours and had obtained 67 marks in the exam.
- Lets predict whether a student studying 25 hours and having 70 marks in the midterm will pass the final term.
- This is a binary classification problem where a multi layer perceptron can learn from the given examples and make an informed prediction. We will see below how a multi layer perceptron learns such relationships.

Multi Layer Perceptron: (Cont`d)

Training our MLP: The Back-Propagation Algorithm

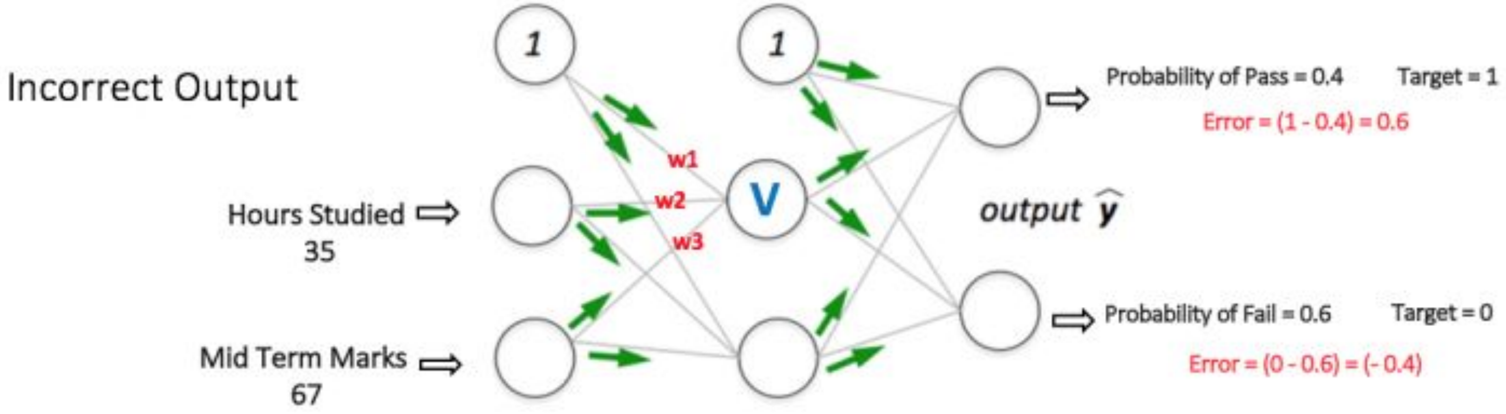
The process by which a Multi Layer Perceptron learns is called the Backpropagation algorithm.

Backward Propagation of Errors, often abbreviated as BackProp is one of the several ways in which an artificial neural network (ANN) can be trained. It is a training scheme, which means, it learns from labeled training data (**learning from mistakes)**

This ANN is said to have learned from several examples (labeled data) and from its mistakes (error propagation).

Multi Layer Perceptron: (Cont`d)

- **Step 1: Forward Propagation**
  - All weights in the network are randomly assigned. Lets consider the hidden layer node marked **V**.
  - Assume the weights of the connections from the inputs to that node are w1, w2 and w3.
  - we know that for inputs 35 and 67, the bias is 1.
  - V = ***f*** (1*w1 + 35*w2 + 67*w3)



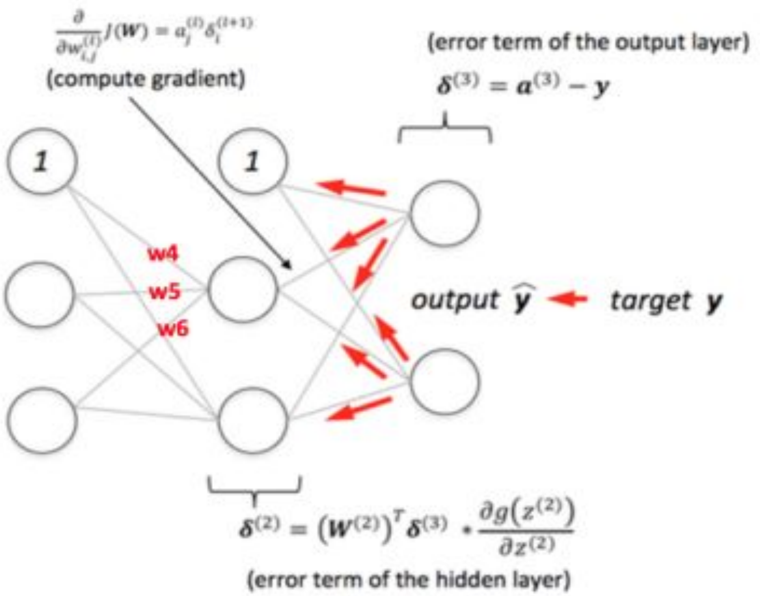
Multi Layer Perceptron: (Cont`d)

- **Step 2: Back Propagation and Weight Updation**
    - We calculate the total error at the output nodes and propagate these errors back through the network using Backpropagation to calculate the *gradients*(adjust weight).
    - Then we use an optimization method such as *Gradient Descent* to 'adjust' **all** weights in the network with an aim of reducing the error at the output layer.

Backpropagation

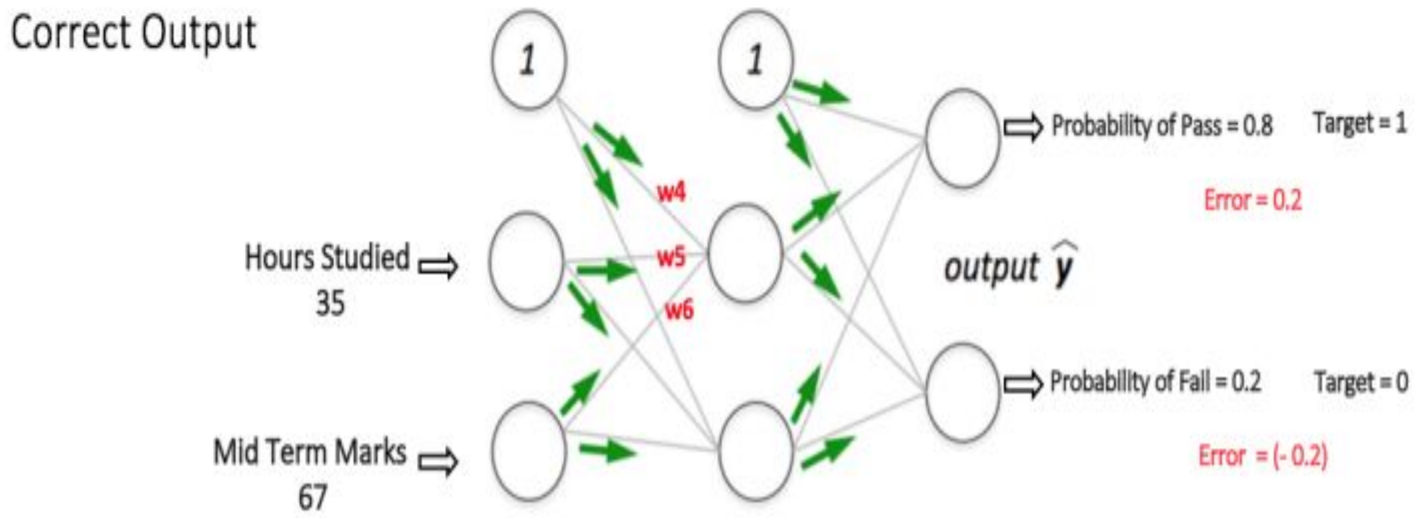
+

Weights Adjusted

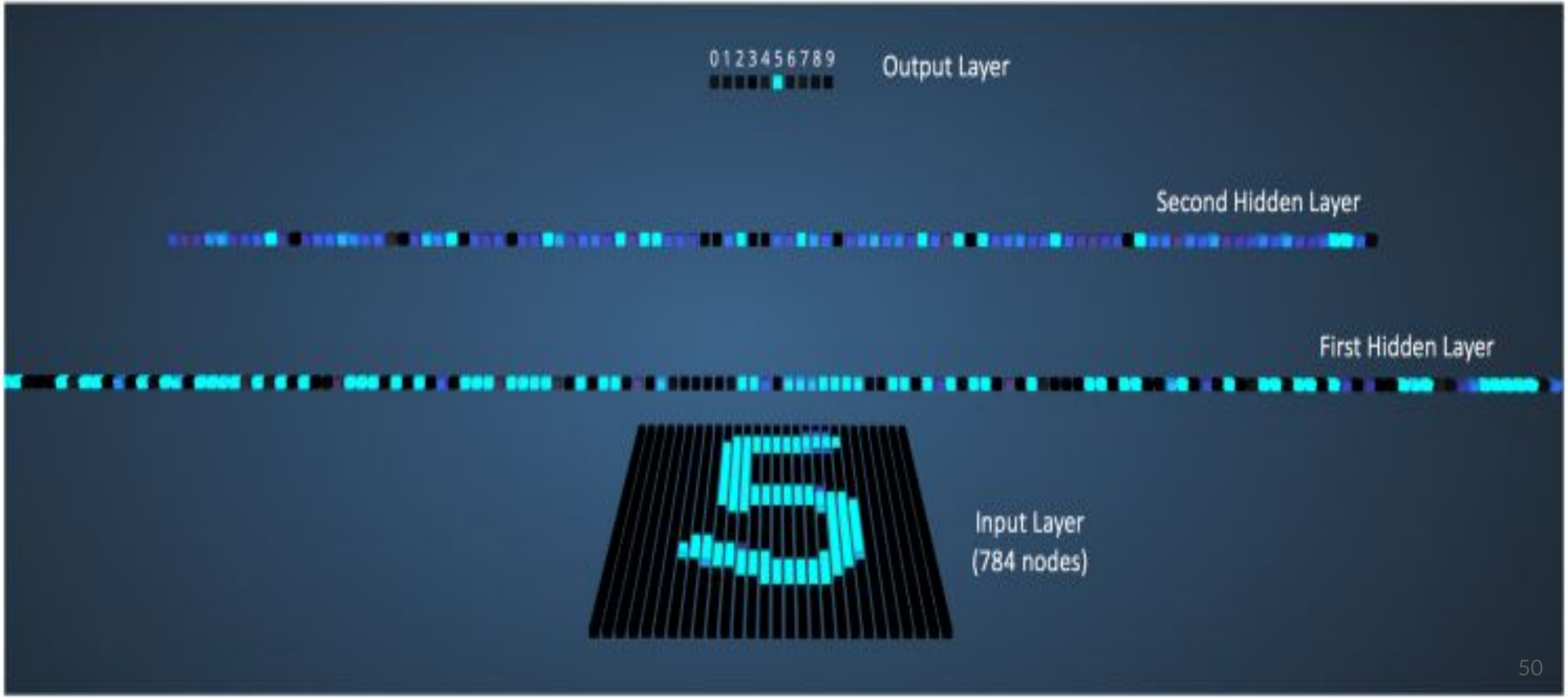


Multi Layer Perceptron: (Cont`d)

- This means that our network has learnt to correctly classify our first training example.
- Similarly, We repeat this process with all other training examples in our dataset. Then, our network is said to have *learnt* those examples.



Multi Layer Perceptron: (Cont`d) for 3D Visualization Multi Layer



Introduction to Deep Learning

- Traditional machine learning models have always been very powerful to handle structured data and have been widely used by businesses for credit scoring, churn prediction(Big Data use case in CRM), consumer targeting, and so on.
- The success of these models highly depends on the performance of the feature engineering phase: the more we work close to the business to extract relevant knowledge from the structured data.
- When it comes to unstructured data (images, text, voice, videos), hand engineered features are time consuming, brittle and not scalable in practice. That is why Neural Networks become more and more popular
- Deep learning is a subset of machine learning, it automatically extracts patterns from raw data(without need of human or initiating).
- http://playground.tensorflow.org
- https://colab.research.google.com
- **Framework: Tensorflow**

What is Deep Learning?

Artificial Intelligence

Any technique that enables computers to mimic human behavior

Machine Learning

Ability to learn without explicitly being programmed

Deep Learning

Extract patterns from data using neural networks

Introduction to Deep Learning(Cont`d)

Applications:

1. Self-driving cars
2. Deep Learning in Healthcare
3. Voice Search & Voice-Activated Assistants
4. Automatically Adding Sounds To Silent Movies
5. Automatic Machine Translation
6. Automatic Text Generation
7. Automatic Handwriting Generation
8. Image Recognition
9. Automatic Image Caption Generation
10. Automatic Colorization
11. Advertising
12. Predicting Earthquakes
13. Brain Cancer Detection
14. Finance (Capital Asset Pricing Model (CAPM) & Cost-Of-Carry relationship)

Puppy or bagel

Labradoodle or fried chicken

Sheepdog or mop

Parrot or guacamole

Introduction to Machine Learning(Cont`d)

- **What is Data Science?**
  - Data science is a multidisciplinary blend of **data inference, algorithm development, and technology** in order to solve analytically complex problems.
  - Data science is ultimately about using this data in creative ways to generate business value.
  - Data Science fall under process area of hadoop(storage, process).
  - Data science is all about uncovering findings from data.
  - Data scientists investigate leads and try to understand pattern or characteristics within the data
  - Data scientists may apply quantitative technique in order to get a level deeper – e.g. inferential models, segmentation analysis, time series forecasting, synthetic control experiments, etc.

Introduction to Machine Learning(Cont`d)

Data Warehouse

Discovery of Data Insight
Quantitative data analysis to help steer strategic business decisions

Development of Data Product
Algorithm solutions in production, operating at scale (e.g. recommendation engines)

Business Value

Links

- https://colab.research.google.com
- http://playground.tensorflow.org
- https://cs.stanford.edu/people/karpathy/ilsvrc/
- http://java-ml.sourceforge.net/
- http://weka.sourceforge.net
- https://cvit.iiit.ac.in/
- http://deeplearning.net/datasets/
- https://iapr.org/
- https://bigml.com/
- http://archive.ics.uci.edu/ml/index.php
- https://douwe.com/projects/machine_learning
- http://googletrends.github.io/data/
- https://deeplearning4j.org/
- https://deepmind.com/
- https://ocw.mit.edu/index.htm
- http://www.openslr.org/12
- https://arxiv.org/corr
- http://ai.stanford.edu/
- https://github.com/ageron
- http://www.robots.ox.ac.uk/~vgg/data/
- https://towardsdatascience.com/mario-vs-wario-image-classification-in-python-ae8d10ac6d63
- https://gogul09.github.io/software/image-classification-python
- http://www.vision.caltech.edu/html-files/archive.html
- http://www.datastuff.tech/
- https://learndigital.withgoogle.com/digitalgarage/course/intro-to-tensorflow-for-deep-learning
- https://open.sap.com/courses
- https://learn.saylor.org/course/index.php
- https://online.stanford.edu/courses
- https://www.data.gov/
- https://open.nasa.gov
- https://data.gov.in/
- https://bhuvan-app3.nrsc.gov.in/data/download/index.php
- http://ml-india.org/datasets/
- https://elearning.iirs.gov.in/