



Machine Learning with Python

Venkatasubramanian R

E-Mail: pvenkat.rv@gmail.com

LinkedIn: <https://www.linkedin.com/in/rvsp>

Introduction to Python



- Python is a general-purpose object-oriented programming language with high-level programming capabilities.
- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
- Python supports modules and packages, which encourages program modularity and code reuse.
- The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed

What is Python?

- ❖ interpreted.
- ❖ high-level.
- ❖ general-purpose programming language.
- ❖ emphasizes code readability.
- ❖ notably using significant whitespace.
- ❖ clear programming on both small and large scales.

Features

- ❖ Easy Syntax.
- ❖ Readability.
- ❖ High-Level Language.
- ❖ Object oriented programming.
- ❖ Open Source.
- ❖ Cross-platform.
- ❖ Extensible.
- ❖ Dynamic.

Advantages & Applications of Python

Advantages:

- ❖ Extensive Support Libraries
- ❖ Integration Feature
- ❖ Improved Programmer's Productivity

Applications:

- ❖ Web and Internet Development
- ❖ Scientific and Numeric
- ❖ Education
- ❖ Desktop GUIs
- ❖ Software Development
- ❖ Business Applications

Introduction to Python (Cont`d)



- Basic Syntaxes
- Loops
- Conditional Statements
- Data types
- Linear Algebra

Introduction to Linear Algebra



Linear Algebra is a branch of mathematics that lets you concisely describe coordinates and interactions of planes in higher dimensions and perform operations on them.

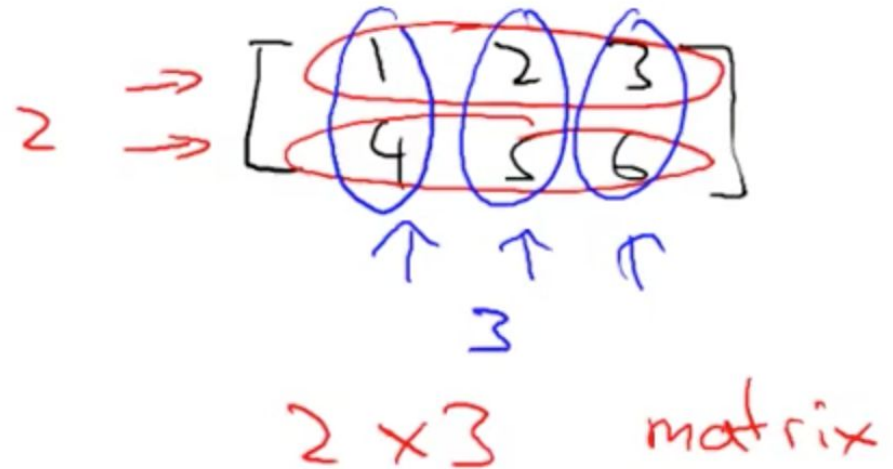
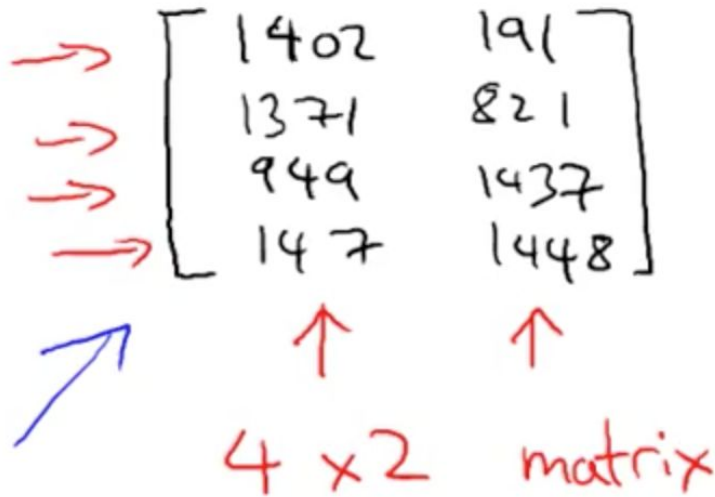
1. Matrices and Vectors
2. Addition and Scalar Multiplication
3. Matrix Vector Multiplication
4. Matrix Matrix Multiplication
5. Properties of Matrix Multiplication

Introduction to Linear Algebra (Cont`d)

1. Matrices and Vectors

- Matrices

- Rectangular array of numbers
- 2D array
- Number of **Rows** x Number of **Columns**



Introduction to Linear Algebra (Cont'd)



Vector

- $n \times 1$ matrix
- $y(i)$: i -th element
- 1-indexed (start from 1-th)
 - Normally this
- 0-indexed (start from 0-th)
 - Used in Machine Learning

Introduction to Linear Algebra (Cont'd)

2. Addition and Scalar Multiplication

a) Addition

- You can only add matrices with the same dimensions (r x c)

The image shows a handwritten example of matrix addition. On the left, a 3x2 matrix is shown with elements 1, 0, 2, 5, 3, 1. The first column elements (1, 2, 3) are circled in red, green, and blue respectively, with red arrows pointing to them from the left. The second column elements (0, 5, 1) are not circled. Below this matrix is the handwritten text "3x2 matrix". To its right is a plus sign, followed by another 3x2 matrix with elements 4, 0.5, 2, 5, 0, 1. The first column elements (4, 2, 0) are circled in red, green, and blue respectively. Below this matrix is the handwritten text "3x2". To the right of the plus sign is an equals sign, followed by a third 3x2 matrix with elements 5, 0.5, 4, 10, 3, 2. The first column elements (5, 4, 3) are colored red, green, and blue respectively. Below this matrix is the handwritten text "3x2" circled in red. The entire equation is written in a casual, handwritten style.

$$\begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 4 & 0.5 \\ 2 & 5 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 0.5 \\ 4 & 10 \\ 3 & 2 \end{bmatrix}$$

3x2 matrix 3x2 3x2

Introduction to Linear Algebra (Cont'd)

b) Scalar (Number) Multiplication

$$\begin{aligned} 3 \times \begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} &= \begin{bmatrix} 3 & 0 \\ 6 & 15 \\ 9 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} \times 3 \\ \begin{bmatrix} 4 & 0 \\ 6 & 3 \end{bmatrix} / 4 &= \frac{1}{4} \begin{bmatrix} 4 & 0 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{3}{2} & \frac{3}{4} \end{bmatrix} \end{aligned}$$

Introduction to Linear Algebra (Cont'd)

3. Matrix Vector Multiplication

- Application to hypothesis by converting given data to matrix \mathbf{x} $\mathbf{prediction} =$

$$\begin{bmatrix} 1 & 3 \\ 4 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 16 \\ 4 \\ 7 \end{bmatrix}$$

$$\begin{aligned} 1 \times 1 + 3 \times 5 &= 16 \\ 4 \times 1 + 0 \times 5 &= 4 \\ 2 \times 1 + 1 \times 5 &= 7 \end{aligned}$$

Introduction to Linear Algebra (Cont'd)

4. Matrix Matrix Multiplication

$$\begin{aligned} & \begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 11 & 10 \\ 9 & 14 \end{bmatrix} \\ & \begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix} = \begin{bmatrix} 11 \\ 9 \end{bmatrix} \\ & \begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \end{bmatrix} \end{aligned}$$


Handwritten annotations: A green box highlights the first column of the second matrix in the first equation, and a green box highlights the first column of the second matrix in the second equation. A blue box highlights the second column of the second matrix in the first equation, and a blue box highlights the second column of the second matrix in the third equation. A green arrow points from the first column of the second matrix in the second equation to the first column of the result matrix. A blue arrow points from the second column of the second matrix in the third equation to the second column of the result matrix.

Introduction to Linear Algebra (Cont`d)

5. Properties of Matrix Multiplication (Cont`d)

- Not commutative

Let A and B be matrices. Then in general,
 $A \times B \neq B \times A$. (not commutative.)

E.g.
$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} = \underline{\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}}$$

$$\begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \end{bmatrix}$$

Introduction to Linear Algebra (Cont'd)

5. Properties of Matrix Multiplication (Cont'd)

- Associative
 - $A \times B \times C = (A \times B) \times C = A \times (B \times C)$

Identity Matrix

Denoted I (or $I_{n \times n}$).

Examples of identity matrices:

$\begin{bmatrix} 1 \end{bmatrix}_{1 \times 1}$ $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2 \times 2}$ $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3}$ $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4}$

1 is identity.

$$1 \times z = z \times 1 = z \quad \text{for any } z$$

Informally:

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

For any matrix A ,

$$A \cdot I = I \cdot A = A$$

$m \times n$ $n \times n$ $m \times m$ $m \times n$ $m \times n$

$I_{n \times n}$

Note:

$$AB \neq BA \text{ in general}$$
$$AI = IA \quad \checkmark$$

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.

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\Amazon
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>

facebook®



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>

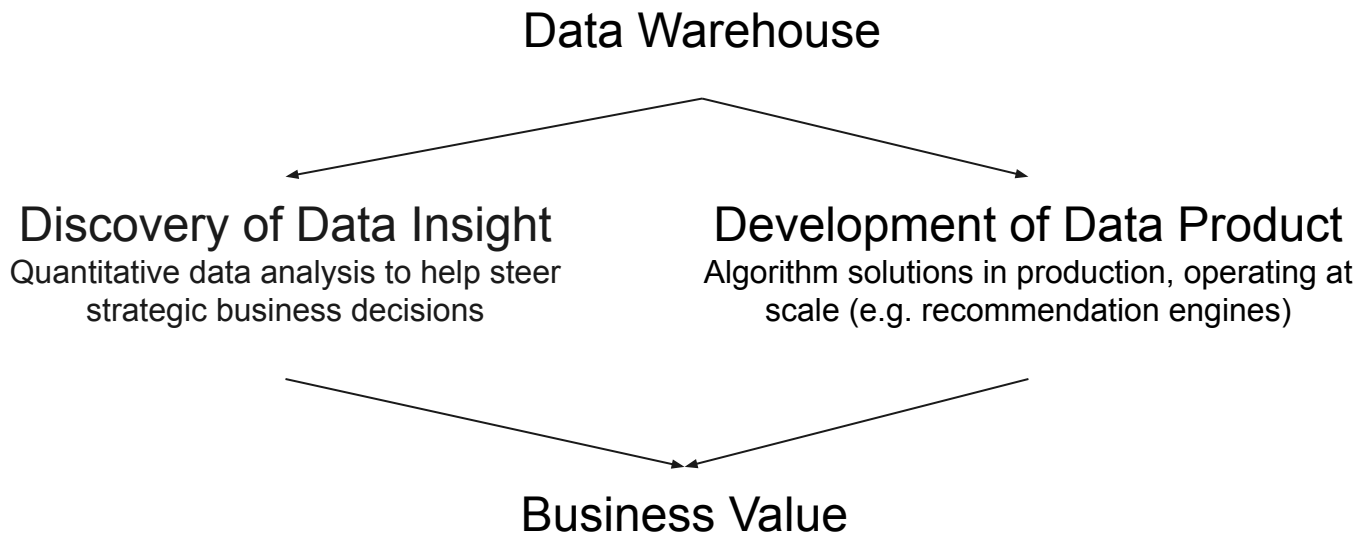
Introduction to Machine Learning(Cont`d)



- **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)



Introduction to Machine Learning(Cont`d)

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.

$$Y = f(X)$$

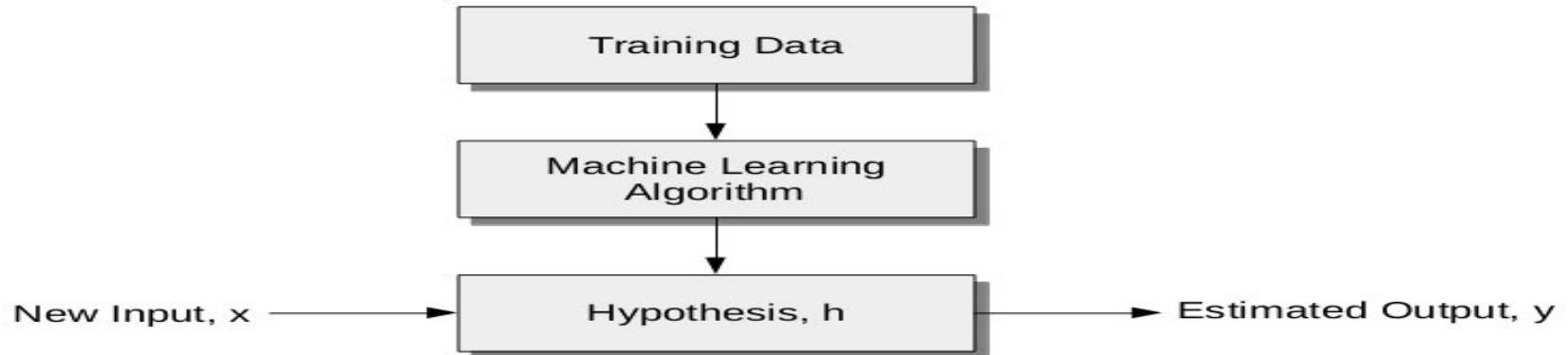
- Supervised learning problems is further grouped into two problems.
 - **Classification:** A classification problem is when the output variable is a category, such as "red" or "blue" or "disease" and "no disease".
 - **Regression:** A regression problem is when the output variable is a real value, such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

Introduction to Machine Learning(Cont`d)

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.



Introduction to Machine Learning(Cont`d)



Supervised/ unsupervised learning In depth (Cont`d)

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.

Introduction to Machine Learning(Cont`d)

Supervised/ unsupervised learning In depth (Cont`d)

Semi-Supervised Machine Learning

- Problems where you have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning problems.
- problems sit in between both supervised and unsupervised learning. Example : album archive.
- Many real world machine learning problems fall into this area.
- It is because it can be expensive or time-consuming to label data as it may require access to domain experts. Whereas unlabeled data is cheap and easy to collect and store.
- You can use unsupervised learning techniques to discover and learn the structure in the input variables.
- You can also use supervised learning techniques to make best guess predictions for the unlabeled data, feed that data back into supervised learning algorithm as training data and use the model to make predictions on new unseen data.

Use Case

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.

Regression



Regression analysis is one of the most important fields in statistics and machine learning. There are many regression methods available. Linear regression is one of them.

Regression searches for relationships among variables.

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.

Multiple Linear Regression

Multiple features (variables).

Size (feet ²)	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

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



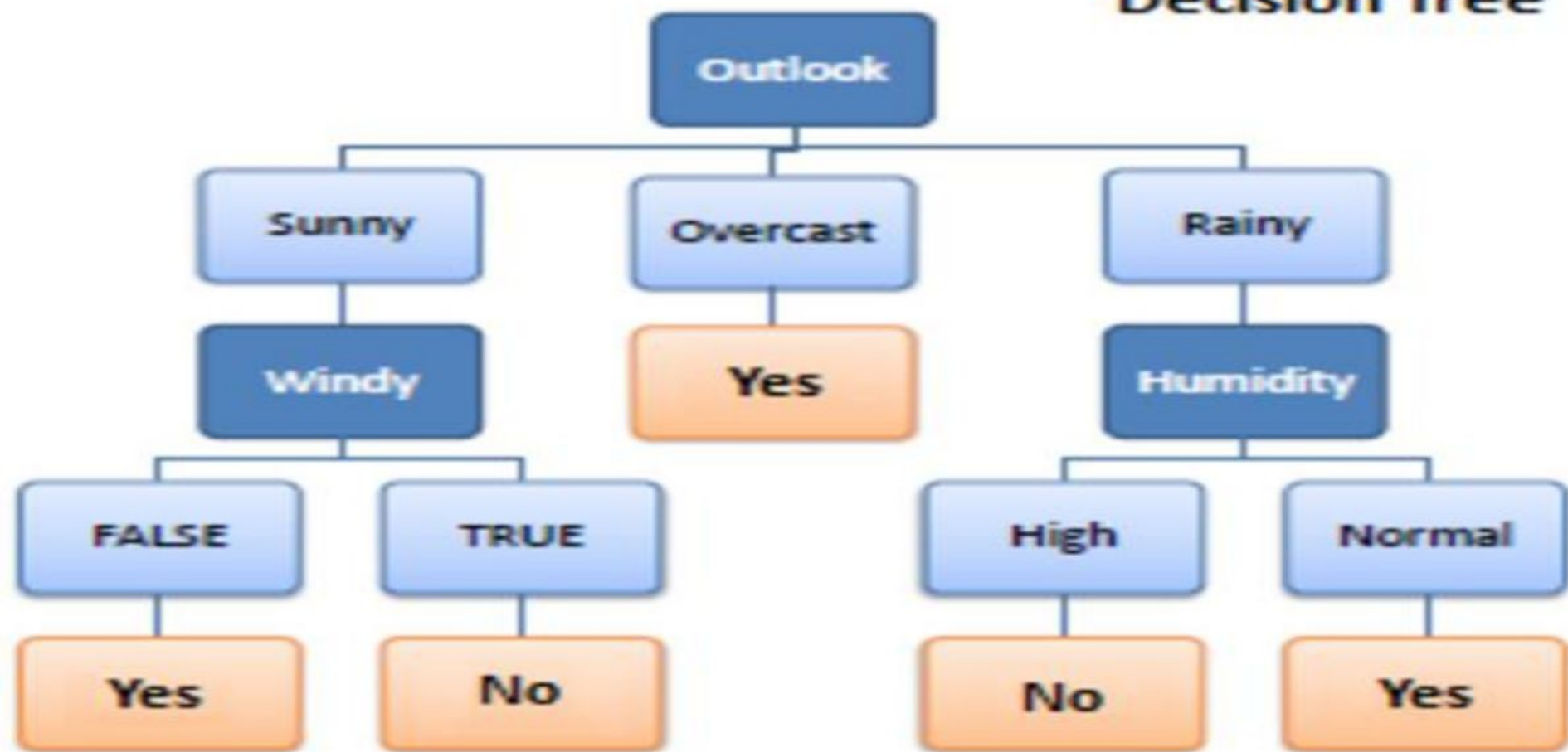
Decision tree

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

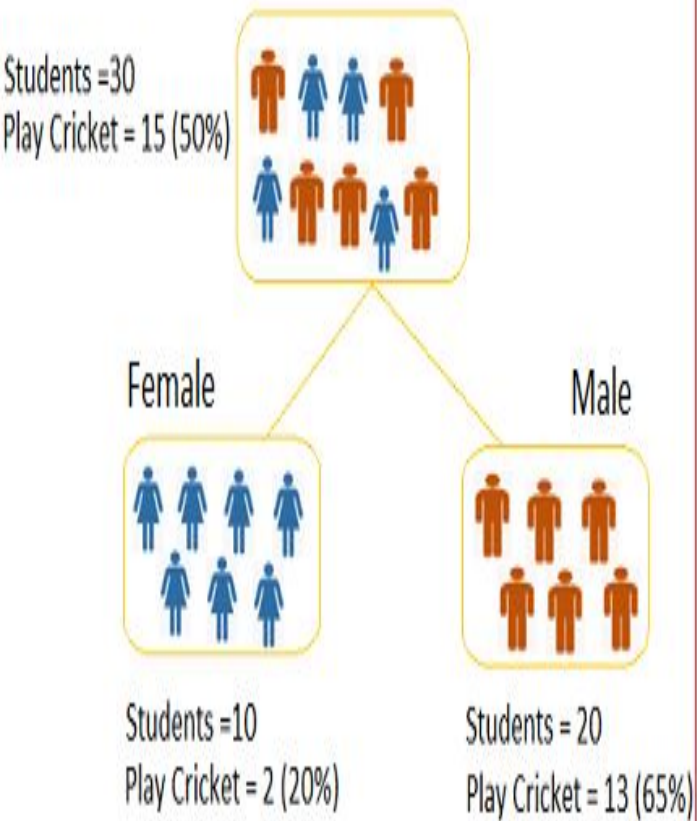
Now the question which arises is, how does it identify the variable and the split?

- Categorical Variable Decision Tree
- Continuous Variable Decision Tree

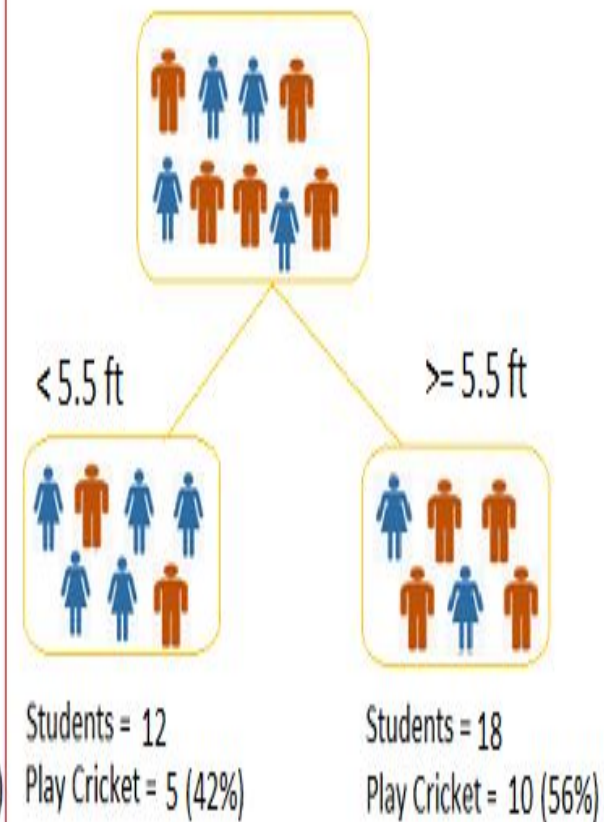
Decision Tree



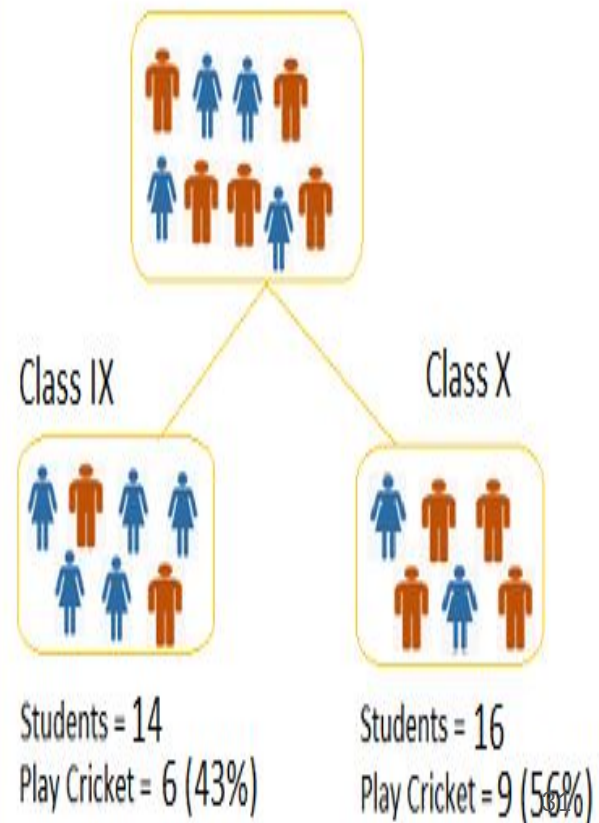
Split on Gender



Split on Height



Split on Class





Decision Tree (cont`d)

Gini Index: categorical target variable "Success" or "Failure".

Entropy $\text{Entropy} = -p \log_2 p - q \log_2 q$

Introduction to Neural Networks

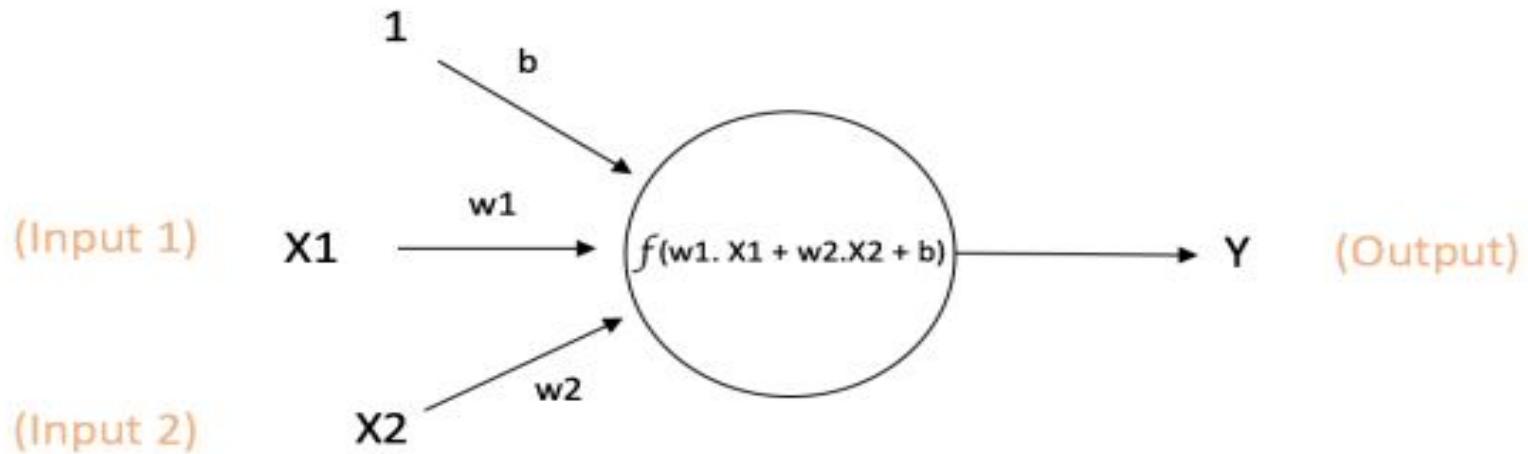


An Artificial Neural Network (ANN) is a computational model that is inspired by the way biological neural networks in the human brain process information.

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)



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

- another input 1 with weight b (called the **Bias**)
- $X1, X2, \dots$ are numerical inputs
- $w1, w2, \dots$ 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

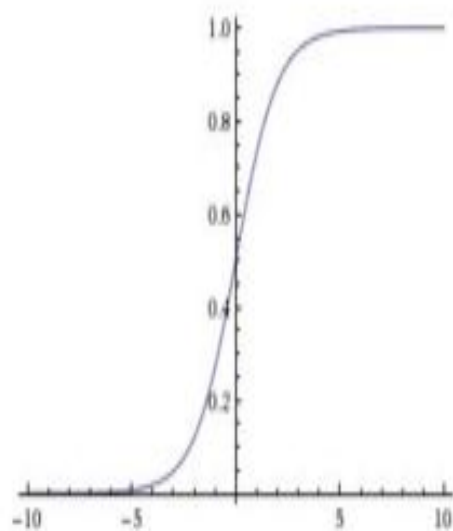
$$\sigma(x) = 1 / (1 + \exp(-x))$$

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

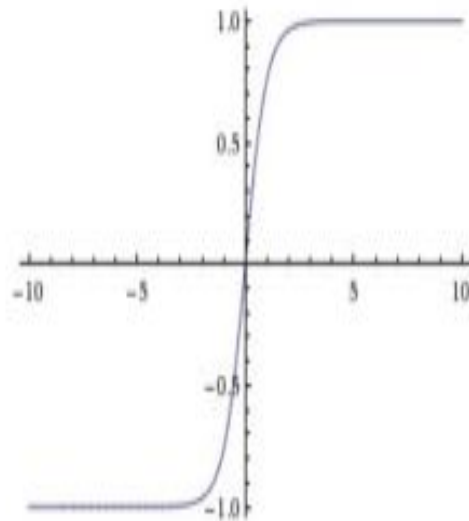
$$\tanh(x) = 2\sigma(2x) - 1$$

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

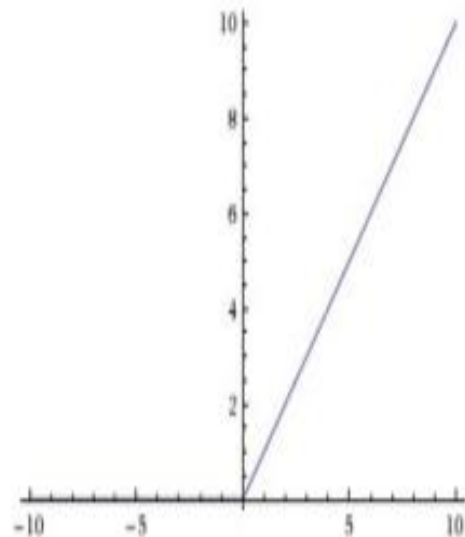
$$f(x) = \max(0, x)$$



Sigmoid



tanh

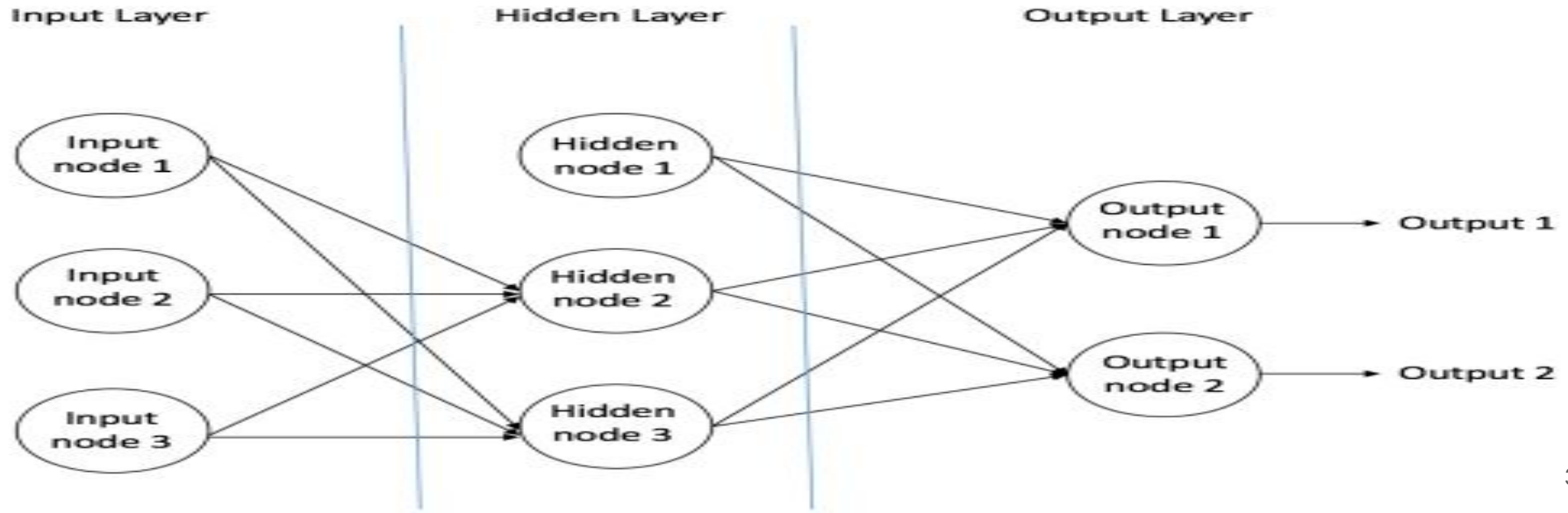


ReLU

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)

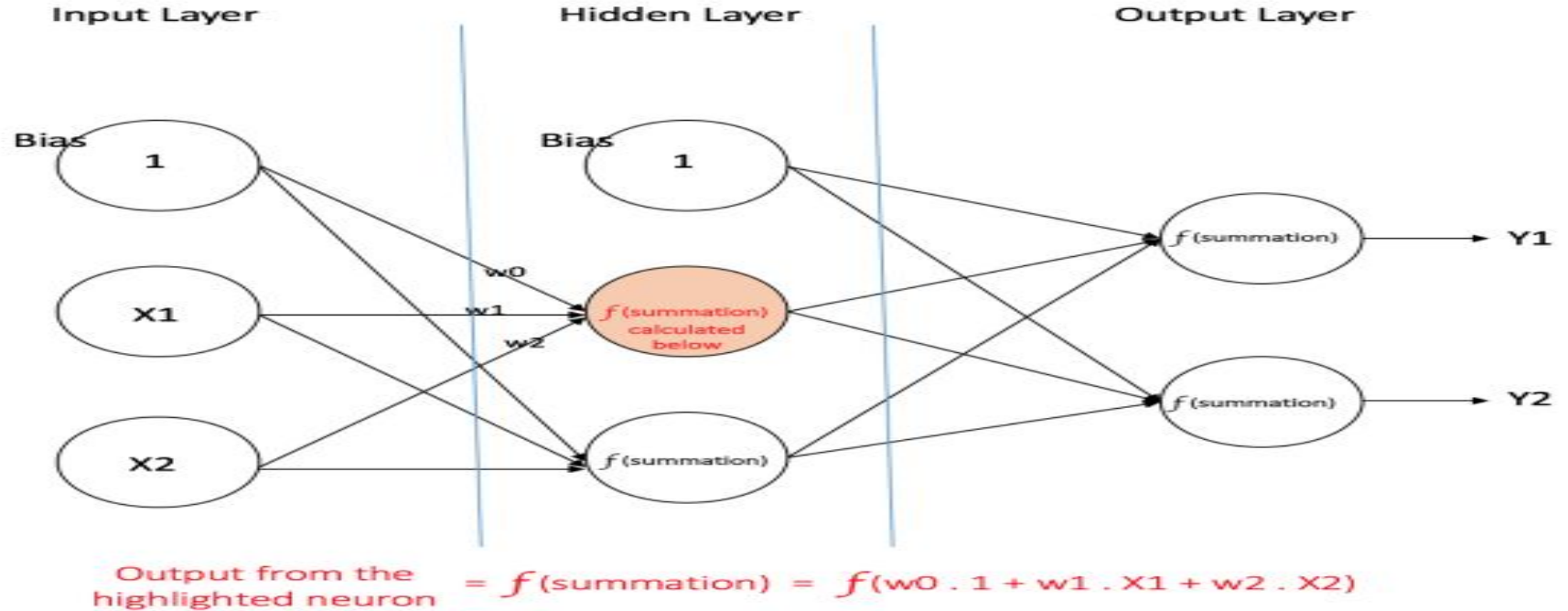
Two examples of feedforward networks

- **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.

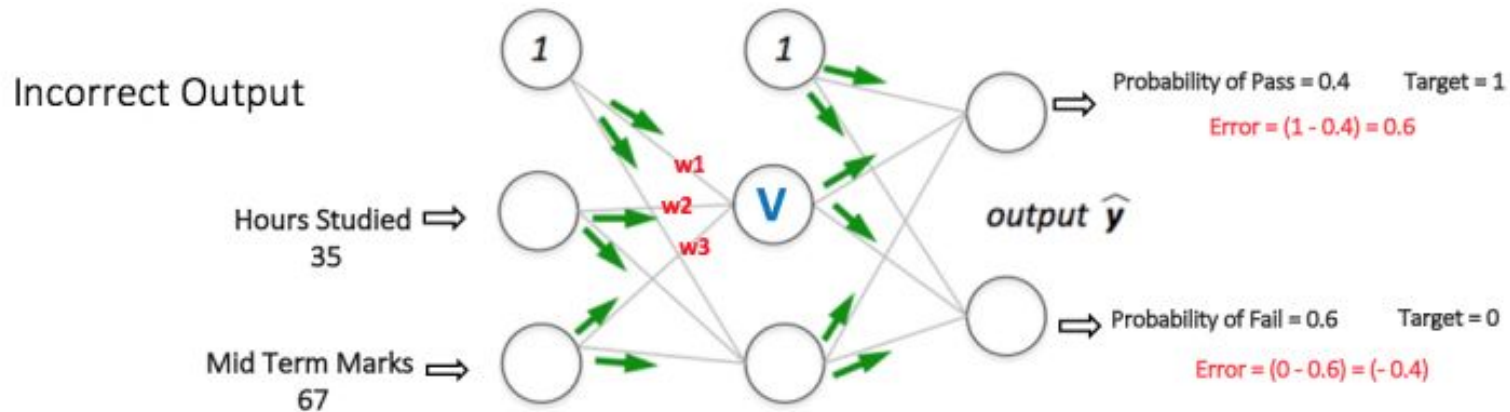
Backward Propagation of Errors, often abbreviated as BackProp is one of the several ways in which an artificial neural network (ANN) can be trained. It is a training scheme, which means, it learns from labeled training data (**learning from mistakes**)

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 w_1 , w_2 and w_3 .
- we know that for inputs 35 and 67, the bias is 1.
- $V = f(1*w_1 + 35*w_2 + 67*w_3)$

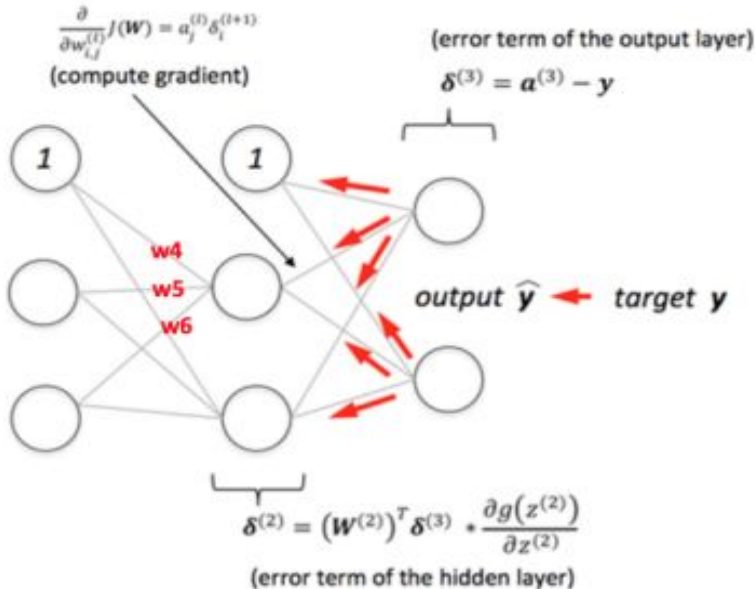


Multi Layer Perceptron: (Cont'd)

- **Step 2: Back Propagation and Weight Update**

- 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.

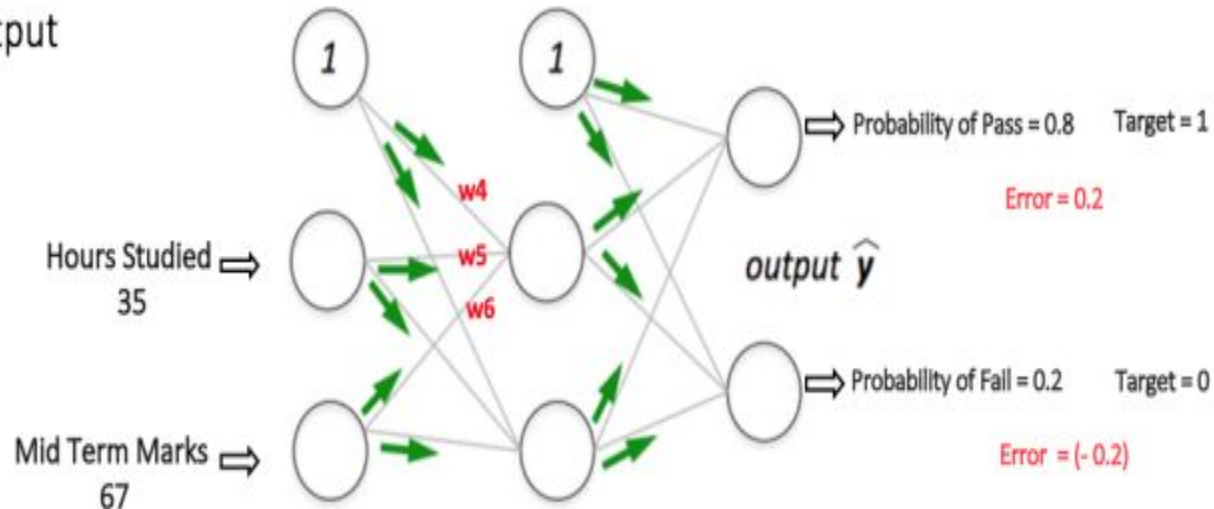
Backpropagation
+
Weights Adjusted



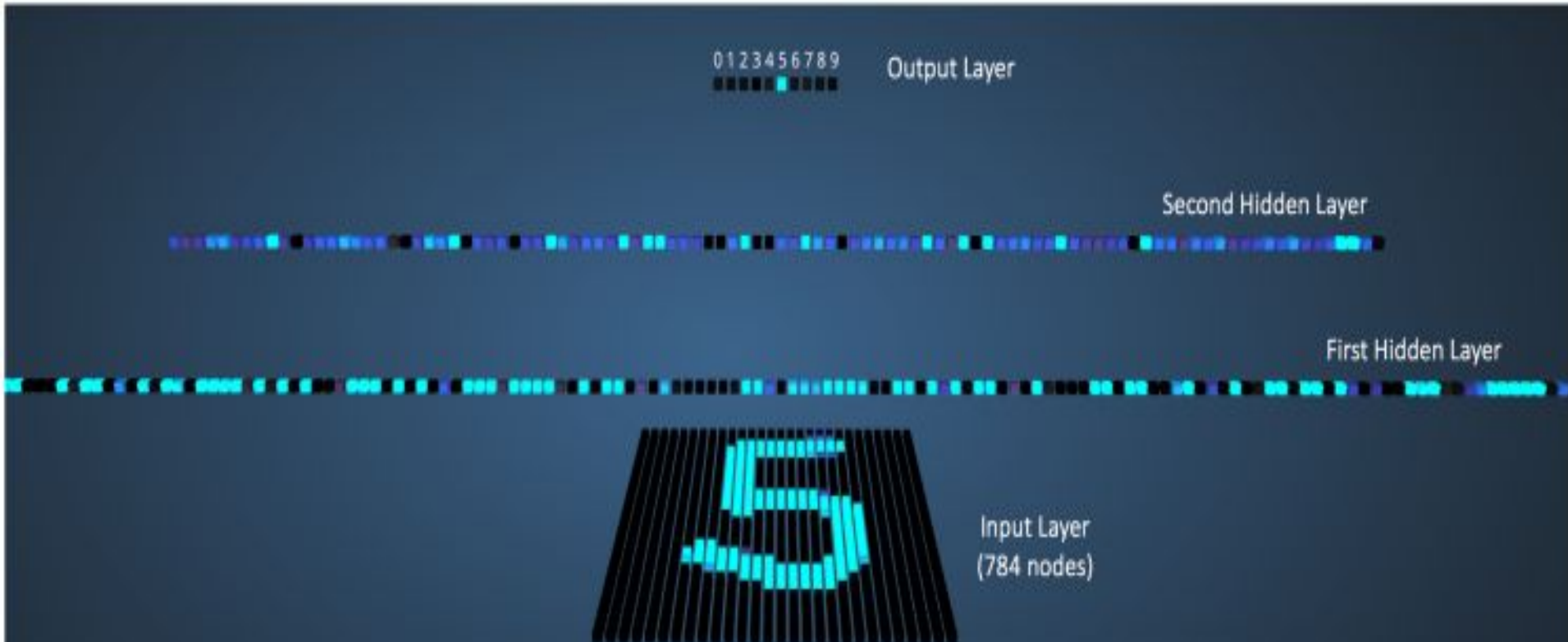
Multi Layer Perceptron: (Cont`d)

- 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.

Correct Output



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



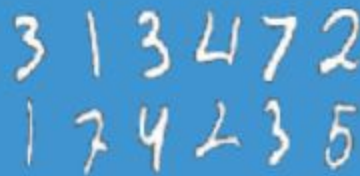
MACHINE LEARNING

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



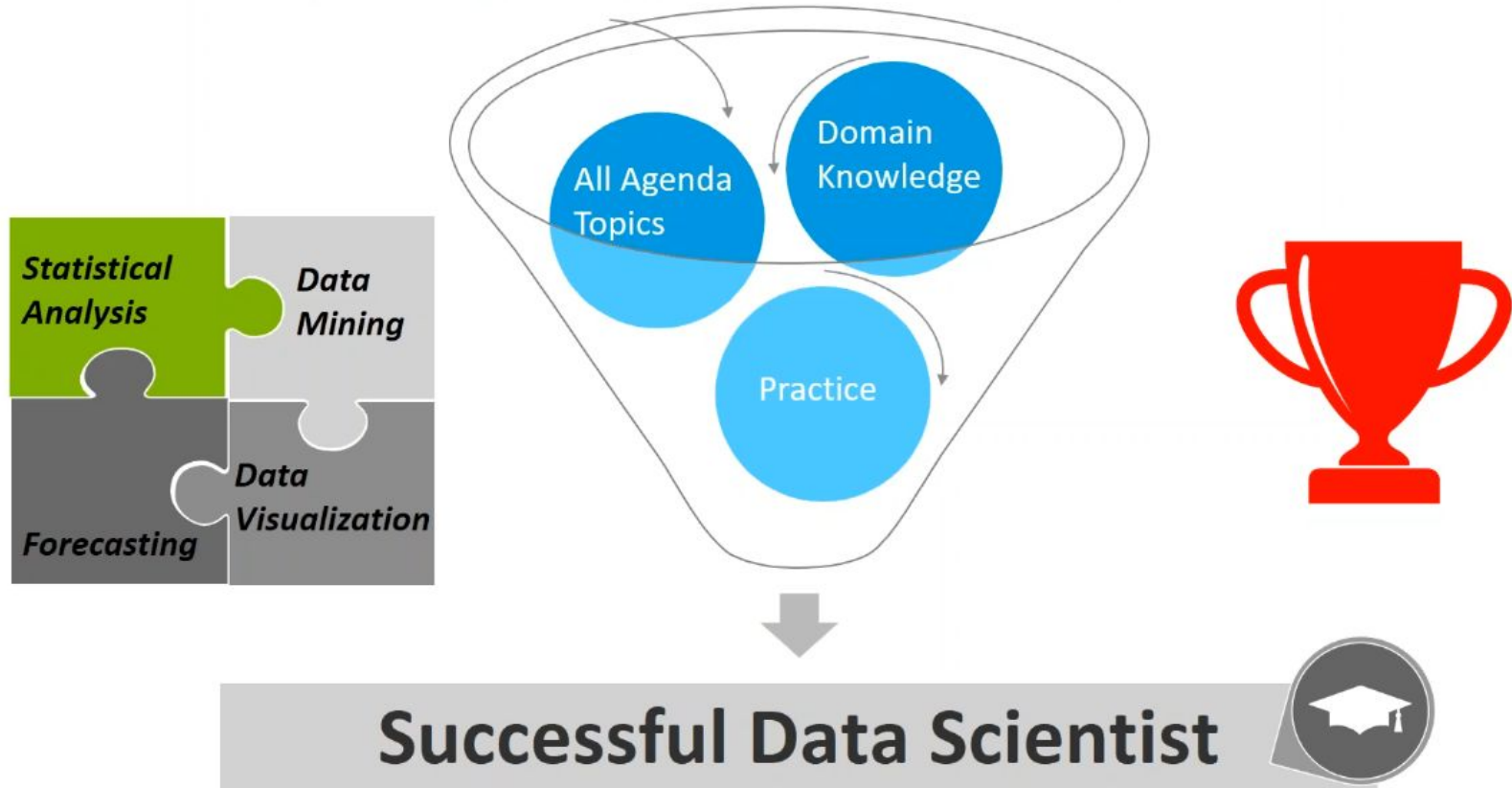
Parrot or guacamole



Links

- <https://colab.research.google.com>
- <http://playground.tensorflow.org>
- <https://cs.stanford.edu/people/karpathy/ilstvrc/>
- <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/>

What does it take to be a DATA SCIENTIST?



OLD CROW



MODERN CROW



Update Yourself - It saves a lot of extra effort