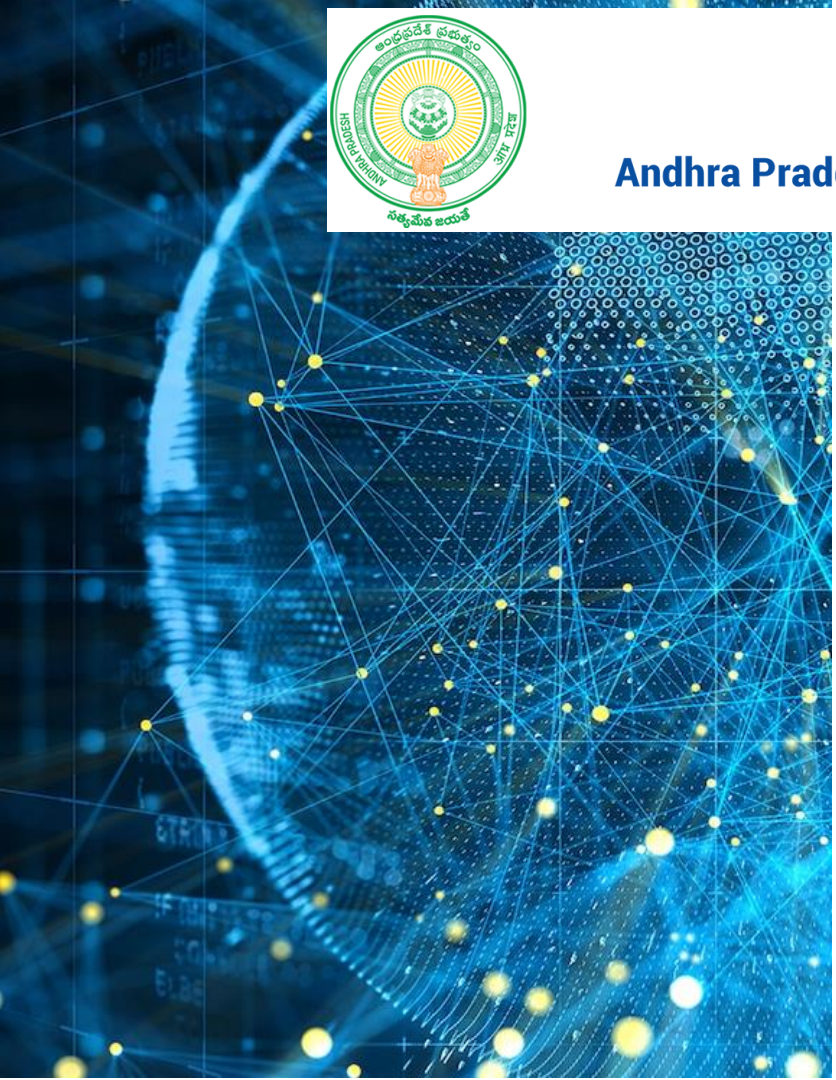


APSSDC

Andhra Pradesh State Skill Development Corporation

Data Science Using Python

ANACONDA

NumPy

pandas

jupyter

matplotlib

scikit learn

WHAT IS DATA SCIENCE

Data science is the domain of study that deals with vast volumes of **data** using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions

WHAT DOES A DATA SCIENTIST DO?

A **data scientist** analyzes business data to extract meaningful insights. In other words, a data scientist solves business problems through a series of steps, including:

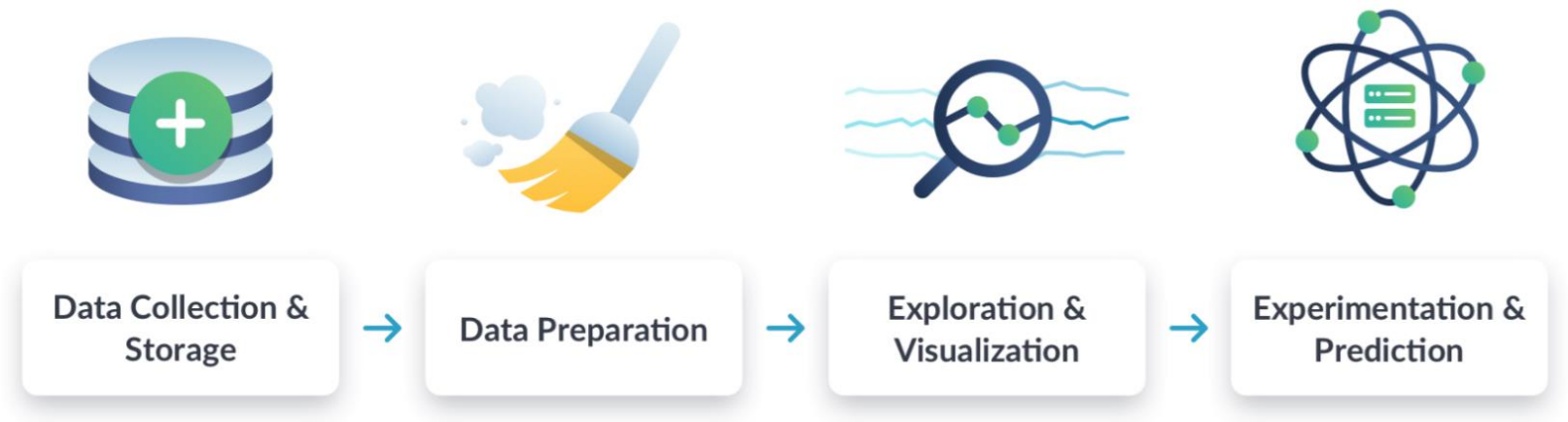
- Ask the right questions to understand the problem
- Gather data from multiple sources—enterprise data, public data, etc
- Process raw data and convert it into a format suitable for analysis
- Feed the data into the analytic system—ML algorithm or a statistical model
- Prepare the results and insights to share with the appropriate stakeholders

DATA SCIENCE APPLICATIONS

Fraud and Risk Detection	Healthcare.	Internet Search	Targeted Advertising
Website Recommendations	Advanced Image Recognition	Speech Recognition	Airline Route Planning

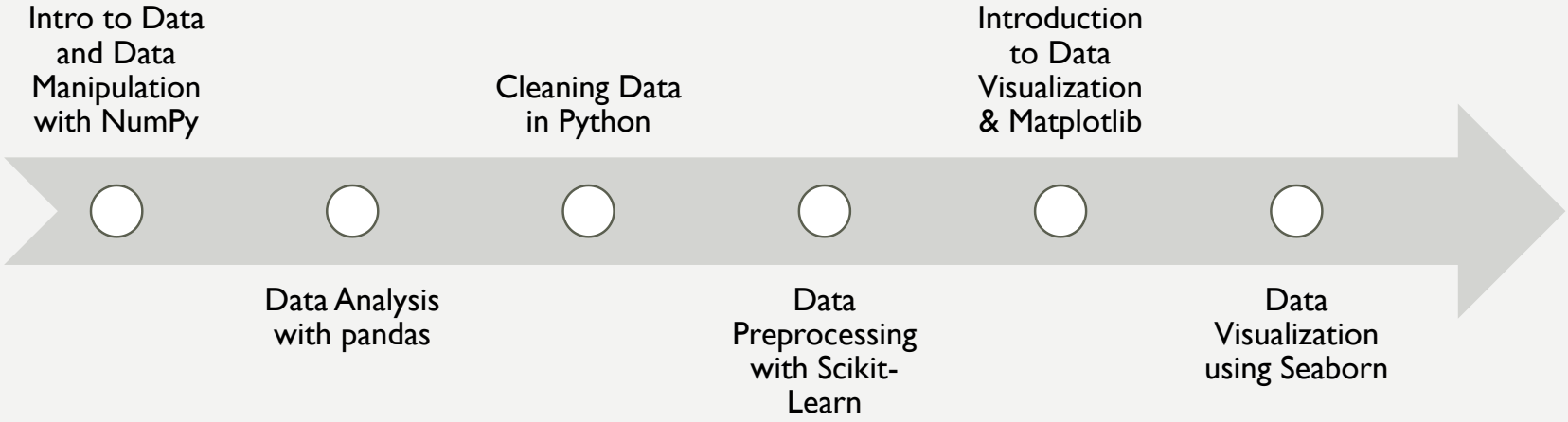
IF DATA IS THE NEW OIL

DATA SCIENCE WORKFLOW



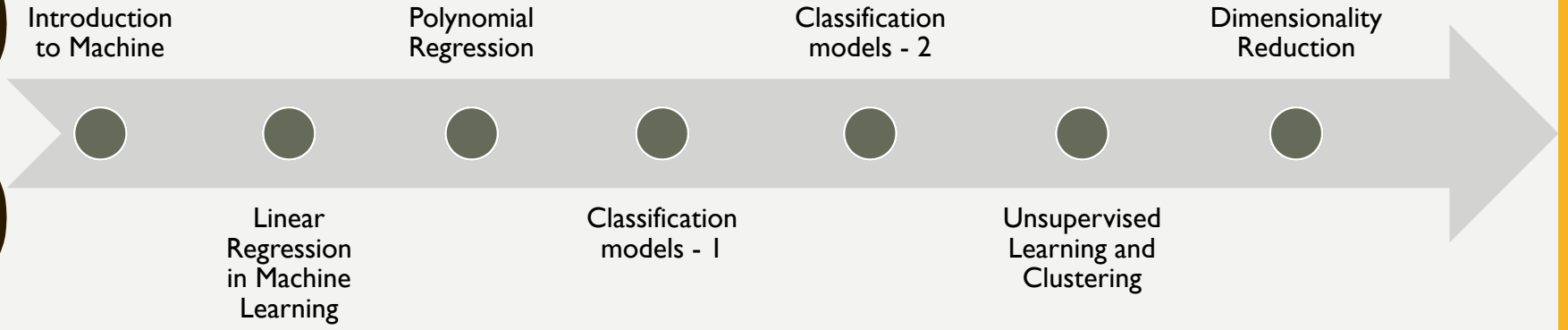
By Anil Kumar APSSDC

AGENDA PART - 1

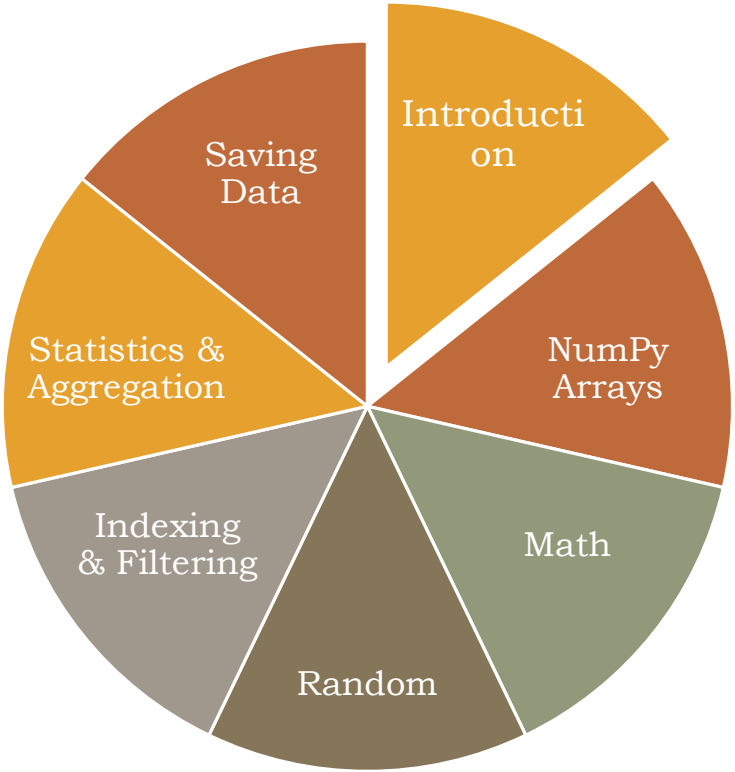


By Anil Kumar APSSDC

AGENDA PART - 2

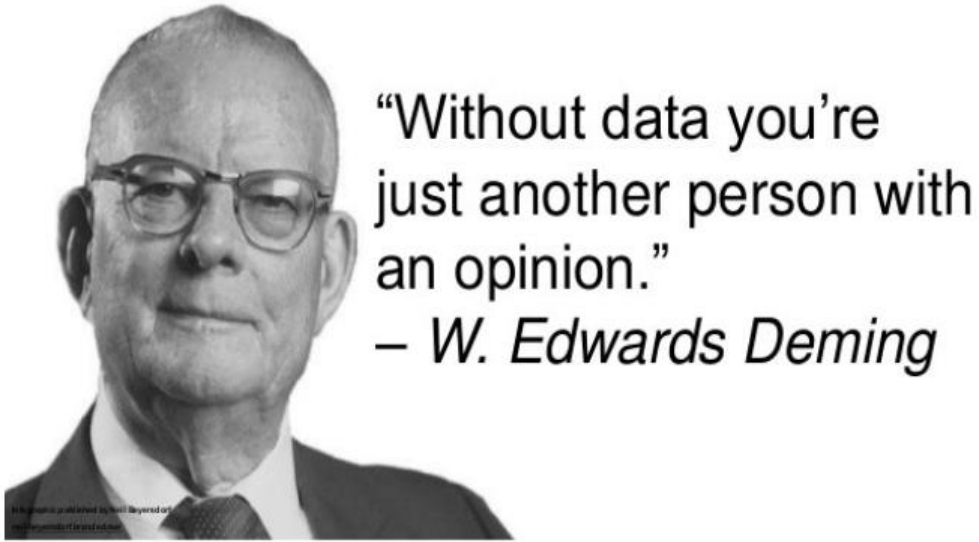


Introduction Data and Data Manipulation with NumPy



What is Data?

Data are facts and statistics collected together for reference or analysis.



Interesting insights

Bombardier showcased its C Series jetliner that carries Pratt & Whitney's Geared Turbo Fan (GTF) engine, which is fitted with 5,000 sensors that generate up to 10 GB of data per second. A single twin-engine aircraft with an average 12-hr. flight-time can produce up to 844 TB of data.

Saudi Aramco laid 650km of new pipelines across a mountain range of red sand dunes. How do they monitor that?

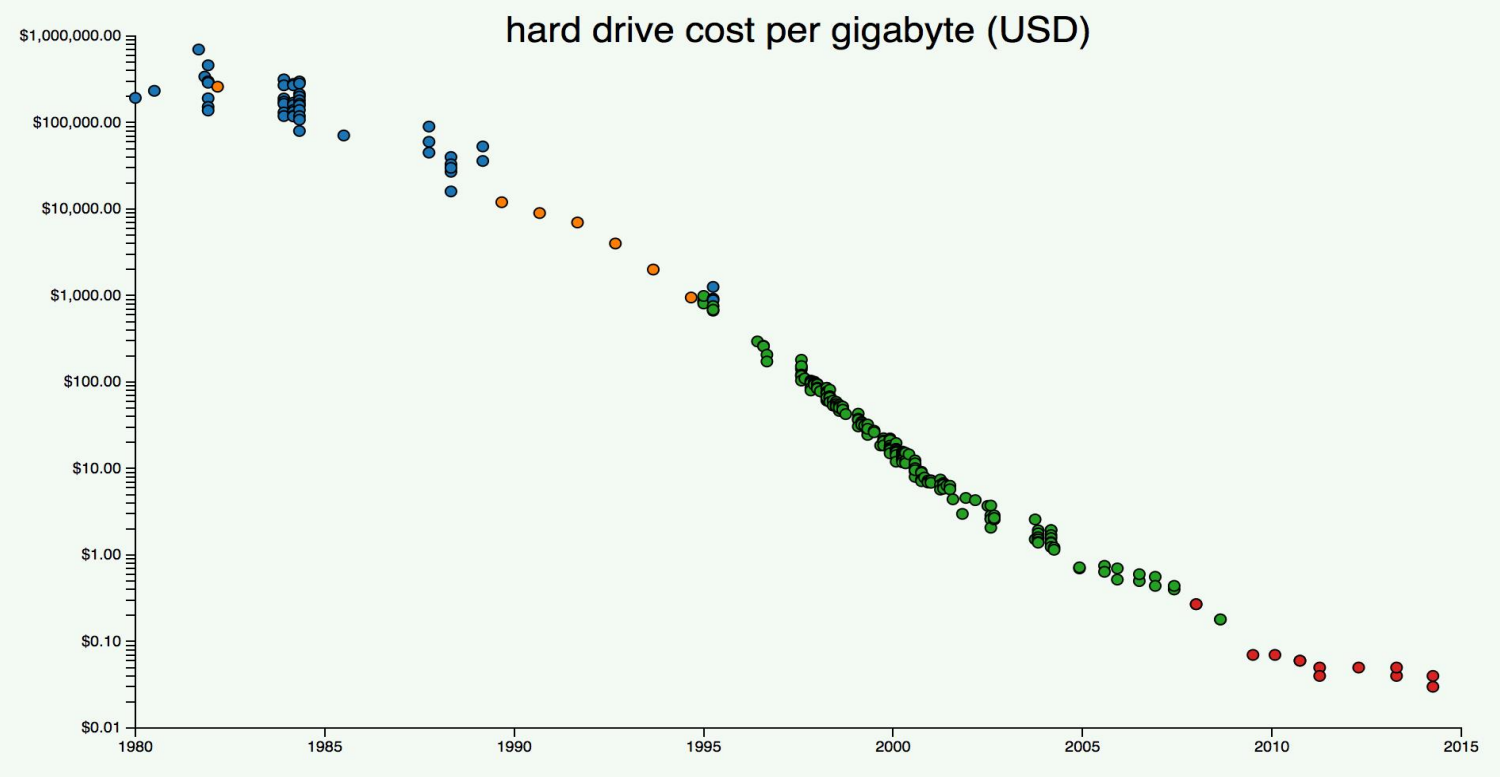
Using 100,000 sensors and data points on wells, pipelines, plants and terminals, it directs every drop of oil and cubic foot of gas that comes out of the kingdom

One study predicts that by 2020, 1.7 MB of data will be created every second for every person on earth.

The average number of AI projects for a business is expected to increase to 35 by 2022 from four this year, according to a Gartner Inc. survey of about 100 organizations of various sizes, many of them with annual revenue of $1 billion to $3 billion. The research and advisory firm also said the number of its clients requesting help in dealing with AI suppliers grew 57% between 2017 and 2018.

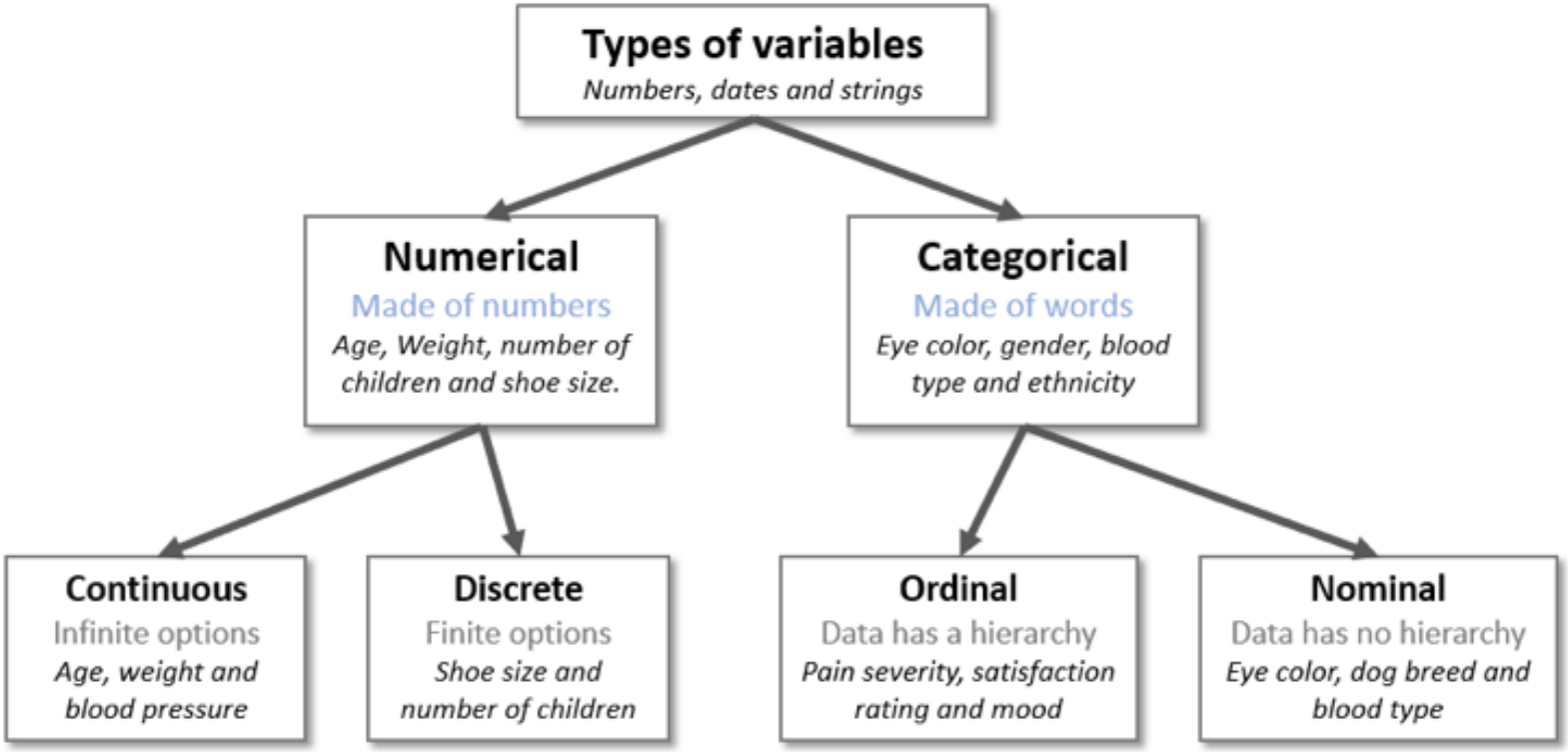
As per the report by NASSCOM and Blueocean, India is reigning big data analytics with a value of $1.2 billion placing it among the top 10 big data analytics markets in the world. They have also anticipated the growth becoming eight-fold by 2025, soaring to $16 billion. With this vision in mind, every sector is now looking forward to Data analytics for its evolution.

Storage capacity, size & cost



Data Generation

DATA TYPES IN STATISTICS



https://en.wikipedia.org/wiki/Statistical_data_type

NUMERICAL DATA

1. DISCRETE DATA

If its values are distinct and separate. In other words: We speak of discrete data if the data can only take on certain values. This type of data can't be measured but it can be counted. It basically represents information that can be categorized into a classification. An example is the number of heads in 100 coin flips.

You can check by asking the following two questions whether you are dealing with discrete data or not: Can you count it and can it be divided up into smaller and smaller parts?

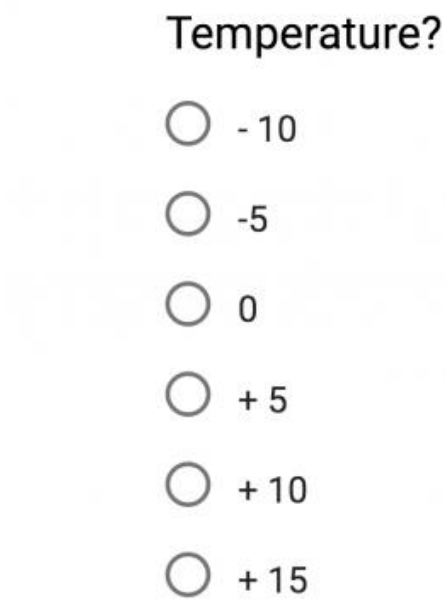
2. CONTINUOUS DATA

Continuous Data represents measurements and therefore their values can't be counted but they can be measured. An example would be the height of a person, which you can describe by using intervals on the real number line.

Contd..

Interval Data

Interval values represent ordered units that have the same difference. Therefore we speak of interval data when we have a variable that contains numeric values that are ordered and where we know the exact differences between the values. An example would be a feature that contains temperature of a given place like you can see



CATEGORICAL DATA

Categorical data represents characteristics. Therefore it can represent things like a person's gender, language etc. Categorical data can also take on numerical values (Example: 1 for female and 0 for male). Note that those numbers don't have mathematical meaning.

NOMINAL DATA

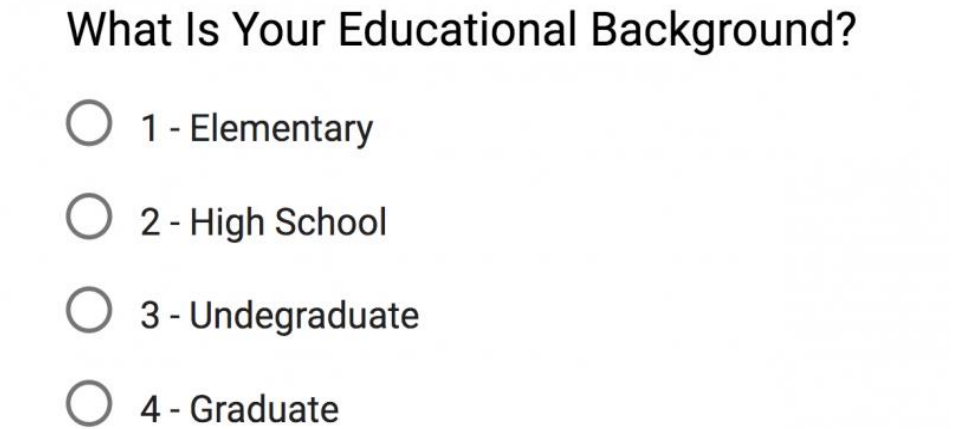
Nominal values represent discrete units and are used to label variables, that have no quantitative value. Just think of them as labels. Note that nominal data that has no order. Therefore if you would change the order of its values, the meaning would not change. You can see two examples of nominal features in the right.

The left feature that describes a persons gender would be called „dichotomous", which is a type of nominal scales that contains only two categories.

Contd..

ORDINAL DATA

Ordinal values represent discrete and ordered units. It is therefore nearly the same as nominal data, except that it's ordering matters. You can see an example below:



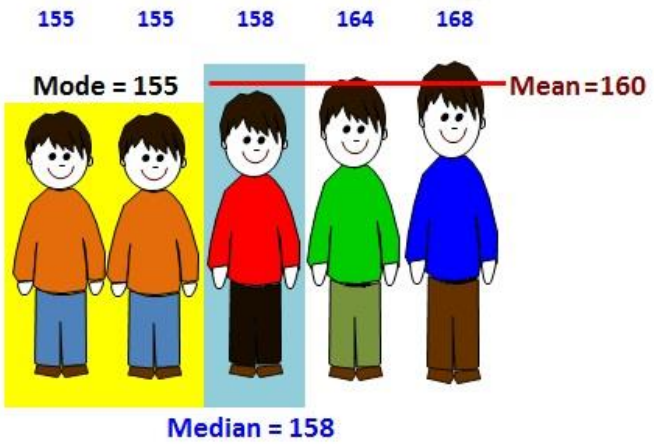
Note that the difference between Elementary and High School is different than the difference between High School and College. This is the main limitation of ordinal data, the differences between the values is not really known.

What is Statistics?

A branch of mathematics that takes and transform the data into some useful information which in turn is used to make some decisions.

Statistics is concerned with

- Processing and analyzing data
- Collecting, presenting and transforming data to assist decision maker



Measures of Dispersion

Range: It is the difference between highest value and the lowest value in the data set.

For a given list of numbers: 10, 20, 40, 10, 70 the range is 70 - 10 = 60.

Variance: The average of the squared differences from the mean.

Steps to calculate variance:

- Calculate mean (mean is nothing but average)
- Find difference of each data from mean
- Square all the differences
- Take the average of the squares.

$$S^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

Standard Deviation: It shows you how much your data is spread out around the mean. Its symbol is **σ** (the Greek letter sigma). It is the square root of the **variance.**

Numerical Python (Numpy)

The library made for scientific and mathematical computations

What is Numpy?

- Numerical Python, popularly known as Numpy has been designed to carry out mathematical computations at a faster and easier rate.
- Further this library enriches the programming language Python by providing powerful data structures like multi dimensional arrays beyond matrices and linear arrays.
- Besides that, Numpy provides a large library of high level mathematical functions to operate on these structures.

How to install Numpy

- In command line

  pip install numpy

- Anaconda distribution

  conda install numpy

Python Objects vs Numpy

python objects

1. high-level number objects: integers, floating point

2. containers: lists (costless insertion and append), dictionaries (fast lookup)

Numpy provides

1. extension package to Python for multi-dimensional arrays

2. closer to hardware (efficiency)

3. designed for scientific computation (convenience)

4. Also known as array oriented computing

Why Numpy when we have “Lists” ?

Python has inbuilt data structure “List” which is also technically an array which allows different data types.

The answer to this question comes in following three aspects

1. **Size** – Numpy data structures take less space
2. **Performance** – They are inherently faster than lists.
3. **Functionality** – Scipy and Numpy have optimized functions.
4. **Vectorization** of the operations

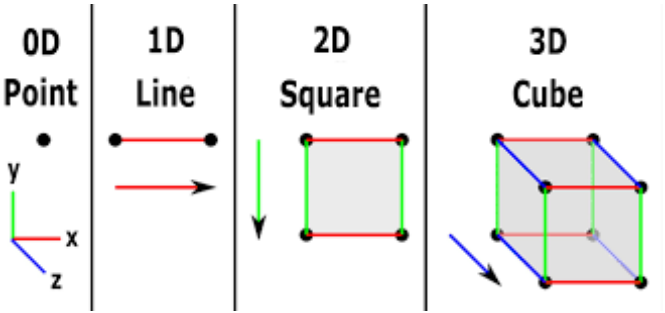
Now let’s create some numpy arrays and play around with them!!

Nd-array object

- Ndarray is multidimensional object which can contain only single data type objects.
- It can be a string type or numeric or integer data type.
- If we mixture of strings and numbers are used, all are converted to strings.

Attributes of ND array object

- Dimension – It tells us the number of dimensions of the nd array object. Number of dimensions can range from 1 to 100s and 1000s



- Shape – It gives the shape of the nd array object. That is the

Attributes of ND array object

- Size - Total number of elements in numpy array
- Dtype – It tells about the type of data being stored in the object.
- Strides – How many steps to be taken to move to next row!!

Some miscellaneous numpy arrays

- Np.zeros()
- Np.arange()
- Np.linspace()
- Np.full()

Resize, reshape , flatten and ravel

- Resize adds zeros if you want to create size larger than current one.

Note – resize() does not work on view of nd array but original one

- Reshape – reshapes array to any size and dimension
- Flatter and ravel – They help in flattening

Array indexing and slicing operations

They are straight forward ways to manipulate data into numpy arrays.
Let’s see how to do it for simple toy numpy arrays

Lets manipulate numpy arrays!

Trigonometric operations

- np.sin(), np.cos(), np.exp(), np.sqrt()

Comparison of numpy objects

- np.equal(ar1, ar2), np.not_equal(ar1, ar2)

Logical operations

- np.logical_or(), np.logical_and(), np.logical_not()

Broadcasting of numpy arrays

- To put it in a more practical context, you often have an array that's somewhat larger and another one that's slightly smaller. Ideally, you want to use the smaller array multiple times to perform an operation (such as a sum, multiplication, etc.) on the larger array.
- 1.First off, to make sure that the broadcasting is successful, the ***dimensions of your arrays need to be compatible*.**
- 2.Two dimensions are also compatible when ***one of them is 1***

Matrix operations

Numpy provides a range of functions to carry out various matrix operations

- Addition
- Matrix dot product
- Matrix element wise multiplication
- Matrix multiplication

Let's create arrays with random numbers

The source of randomness which we inject into machine learning projects is called Pseudo randomness.

1. np.random.rand()
2. np.random.randint()
3. np.random.shuffle()

Concatenate, append and stack numpy arrays

Often, we might want to join different numpy arrays in different ways like column wise, row wise.

Numpy offers a range of functions to do the same namely

- np.append()
- np.concatenate()
- np.vstack()
- np.hstack()

Let’s save the nd array object to a file now!

- After all the processing and manipulation of the data, the most important step that comes is to save data into a file.
- Numpy provides savetxt(<file name>,<nd_array_object>, <delimiter>) function for the same.

Numpy quick start links

- Original documentation

https://numpy.org/devdocs/user/quickstart.html

- Cheat sheet

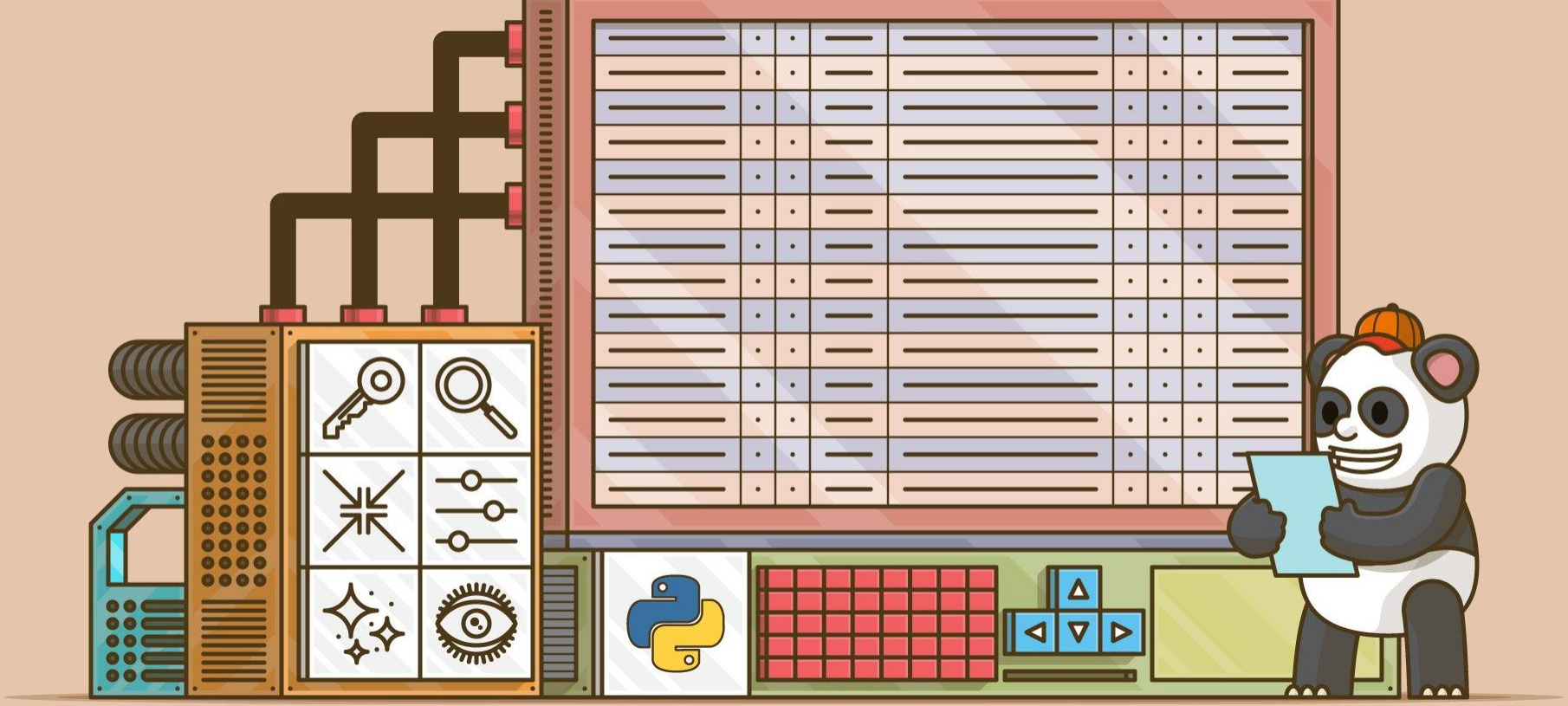
https://www.datacamp.com/community/blog/python-numpy-cheat-sheet

Share your Learning to the Community

Write a blog on LinkedIn, dev.to, toward data science, medium

Ref:
https://medium.com/@karthikayanmailsamy/5-interesting-pytorchs-tensor-functions-55f6d72ace2f

Thanks for taking part in today’s session



Data Analysis using Pandass

Pandas

- Pandas is an open-source python library providing efficient easy-to-use data structures and analysis tools
- Derived from "**PANel Data** – an Econometrics from Multidimensional data".

Data Structures in Pandas

	Dimensions	Description
DataFrames	2	Labeled, heterogeneously typed, size-mutable tabular data
Panels	3	Labeled, size mutable array

All the above data structures are value - mutable

Skill AP APSSDC

pandas

pandas

- The **pandas** Python library provides data structures and methods for manipulating different types of data, such as numerical and temporal data. These operations are easy to use and highly optimized for performance.
- Data formats, such as **CSV** and **JSON**, and databases can be used to create **DataFrames**. DataFrames are the internal representations of data and are very similar to tables but are more powerful since they allow you to efficiently apply operations such as multiplications, aggregations, and even joins. Importing and reading both files and in-memory data is abstracted into a user-friendly interface. When it comes to handling missing data, pandas provide built-in solutions to clean up and augment your data, meaning it fills in missing values with reasonable values.

Advantages of pandas over NumPy

The following are some of the advantages of pandas:

- **High level of abstraction**: pandas have a higher abstraction level than NumPy, which gives it a simpler interface for users to interact with. It abstracts away some of the more complex concepts, such as high-performance matrix multiplications and joining tables, and makes it easier to use and understand.
- **Less intuition**: Many methods, such as joining, selecting, and loading files, are used without much intuition and without taking away much of the powerful nature of pandas.
- **Faster processing**: The internal representation of DataFrames allows faster processing for some operations. Of course, this always depends on the data and its structure.
- **Easy DataFrame design**: DataFrames are designed for operations with and on large datasets.

Disadvantages of pandas

The following are some of the disadvantages of pandas:

- **Less applicable**: Due to its higher abstraction, it's generally less applicable than NumPy. Especially when used outside of its scope, operations can get complex.
- **More disk space**: Due to the internal representation of DataFrames and the way pandas trades disk space for a more performant execution, the memory usage of complex operations can spike.
- **Performance problems**: Especially when doing heavy joins, which is not recommended, memory usage can get critical and might lead to performance problems.
- **Hidden complexity**: Less experienced users often tend to overuse methods and execute them several times instead of reusing what they've already calculated. This hidden complexity makes users think that the operations themselves are simple, which is not the case.

Creating a Dataframe

- A DataFrame represents a rectangular table of data and contains an ordered collection of columns, each of which can be a different value type (numeric, string, boolean, etc.).

```
import pandas as pd
dic = {'Fruits' : ['Apple', 'Banana', 'Orange', 'Grapes'], 'Price/KG':[120,55,60,35}
df = pd.DataFrame(dic)
print(df)
```

- **Output:**

	Fruits	Price/KG
0	Apple	120
1	Banana	55
2	Orange	60
3	Grapes	35

Index Operations

Column indexing: To the column elements by using column names

```
df['Fruits’]
```

Output:

```
0      Apple
1     Banana
2     Orange
3     Grapes
Name: Fruits, dtype: object
```

row indexing: to the get row elements by using **iloc** method row index number

```
df.iloc[0]
```

Output:

```
Fruits       Apple
Price/KG       120
Name: 0, dtype: object
```

Reading and Writing Data from Excel/CSV Formats

- **pd.read_csv()** – to import data into dataframe from csv files
- **df.to_csv()** – to export dataframe into csv file

Syntax:

```
import pandas as pd

df = pd.read_csv('import.csv')

df.to_csv('export.csv')
```

pd.read_excel() – to import data into dataframe from excel files

df.to_excel() – to export dataframe into excel file

Syntax:

```
import pandas as pd

df = pd.read_csv('import.xlsx')

df.to_csv('export.xlsx')
```

Skill AP APSSDC

pandas

Basic Functionalities of a Data Object

- **df.head()** – to get the first n rows from the dataframe.
- **df.tail()** – to get the last n rows from the dataframe.
- **df.columns** - to get columns names of the dataframe
- **df.index** – to get index values of the dataframe.
- **df.info()** – returns the information of the dataframe
- **df.describe()** - returns the descriptive statistics summary
- **df.sum()** – return the sum of each columns
- **df.count()** – return the count the values in the columns
- **df.mean()** – return the mean of the columns
- **df.max()** – returns the maximum value in the dataframe
- **df.idxmax()** – returns the maximum value index from the dataframe
- **df.idxmin()** – returns the minimum value index from the dataframe

Merging of Data Objects

```
import pandas as pd
dic1 = {'Fruits' : ['Apple', 'Banana', 'Orange', 'Grapes'], 'Price/KG':[120,55,60,35}
dic2 = {'Fruits' : ['Apple', 'Banana', 'Orange', 'Grapes'],
'Color':['Red','Yellow','Orange','Green']}
df1 = pd.DataFrame(dic)
df2 = pd.DataFrame(dic2)
print(df1)
Print(df2)
```

Output:

	Fruits	Price/KG
0	Apple	120
1	Banana	55
2	Orange	60
3	Grapes	35

	Fruits	Color
0	Apple	Red
1	Banana	Yellow
2	Orange	Orange
3	Grapes	Green

Skill AP APSSDC

pandas

Merging of Data Objects

To combine the information of two data frames into a single DataFrame, we can use the **pd.merge()** function

- The pd.merge() function recognizes that each DataFrame has an “fruits" column, and automatically joins using this column as a key. The result of the merge is a new DataFrame that combines the information from the two inputs. Notice that the order of entries in each column is not necessarily maintained: in this case, the order of the “fruits" column differs between df1 and df2, and the pd.merge() function correctly accounts for this.

```
pd.merge(df1,df2)
```

output →

	Fruits	Price/KG	Color
0	Apple	120	Red
1	Banana	55	Yellow
2	Orange	60	Orange
3	Grapes	35	Green

Skill AP APSSDC

pandas

Concatenation of Data Objects

Pandas concat() method is used to concatenate pandas objects such as DataFrames and Series. We can pass various parameters to change the behavior of the concatenation operation.

Example:

```
dic1 = {'Fruits' : ['Apple', 'Banana', 'Orange', 'Grapes'], 'Price/KG':[120,55,60,35}
df1 = pd.DataFrame(dic1)
dic1 = {'Fruits':['Pina Apple'],
           'Price/KG':[80]}
df2 = pd.DataFrame(dic)
pd.concat([df1,df2])
```

Output →

	Fruits	Price/KG
0	Apple	120
1	Banana	55
2	Orange	60
3	Grapes	35
0	Pina Apple	80

Types of Joins on Data Objects

one-to-one joins: for example when joining two DataFrame objects on their indexes (which must contain unique values).
many-to-one joins: for example when joining an index (unique) to one or more columns in a different DataFrame.
many-to-many joins: joining columns on columns.
The how argument to merge specifies how to determine which keys are to be included in the resulting table. If a key combination does not appear in either the left or right tables, the values in the joined table will be NA.

Merge method	Join Name	Description
left	Left Outer Join	Use keys from left frame only
right	Right Outer Join	Use keys from right frame only
outer	Full Outer Join	Use union of keys from both frames
inner	Inner Join	Use intersection of keys from both frames

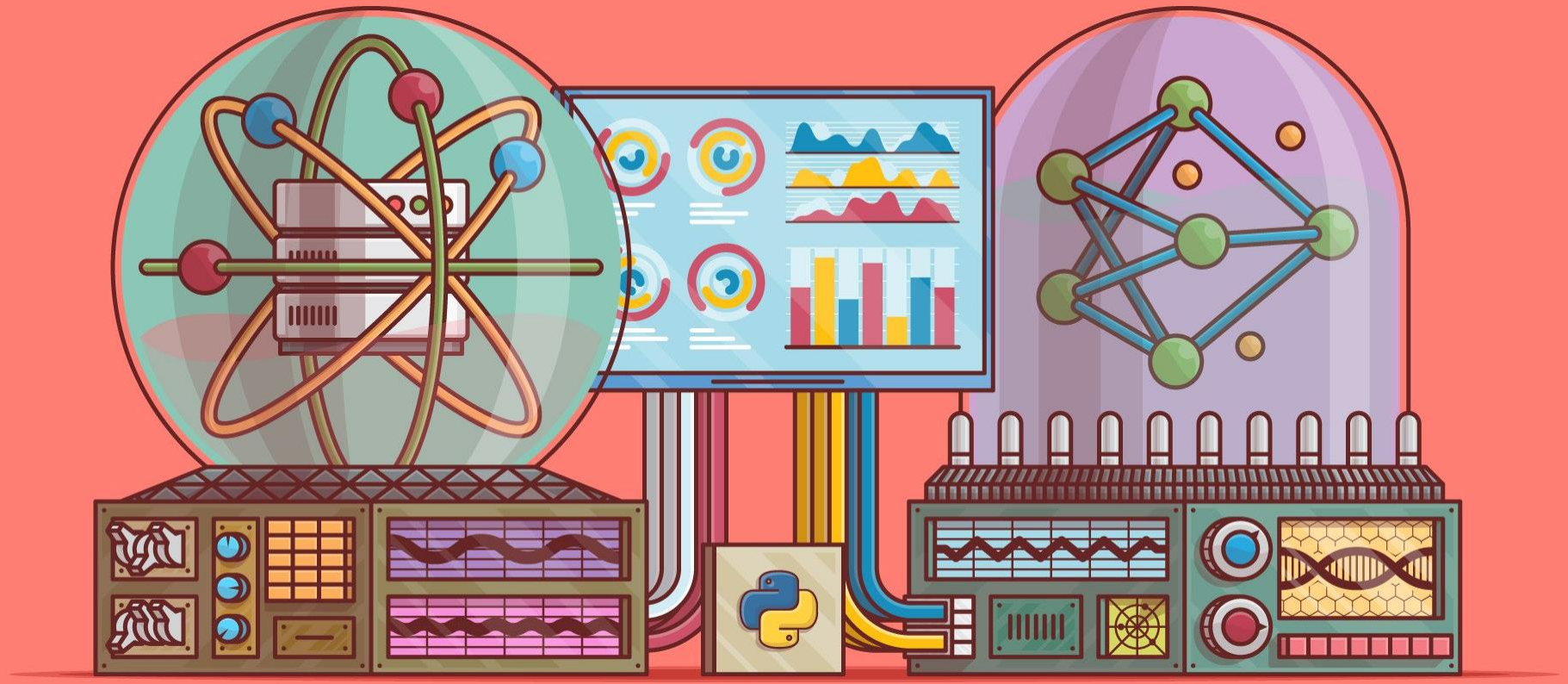
Exploring a Dataset

- Choose a genuine source dataset
- Investigating and exploring the dataset by exploring the columns and rows in the dataset
- Visualize the data explore the data
- Grouping the data and find the relationship between the features and target

Analyzing a dataset

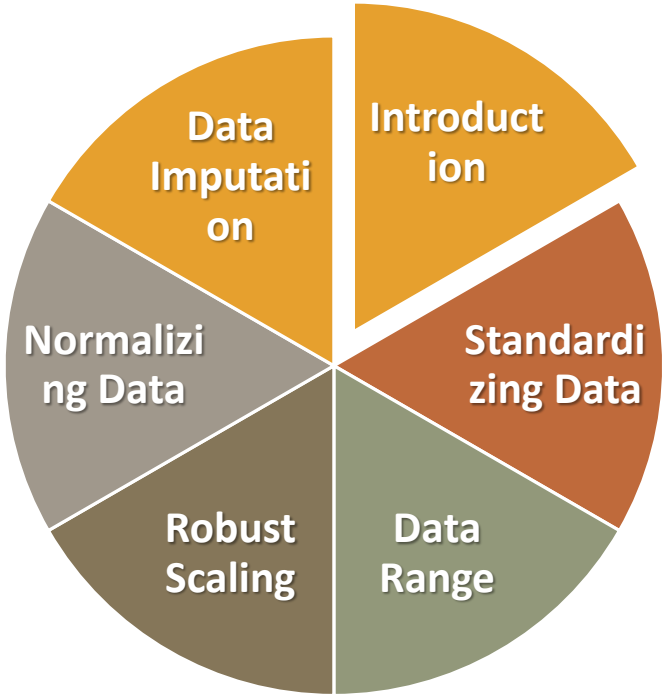
After Loading the data into a dataframe we have to check

1. Duplicates in the Dataframe
2. Missing values in the Dataframe replace them with
   - Remove Rows With Missing Values
   - Mark Missing Values with some value



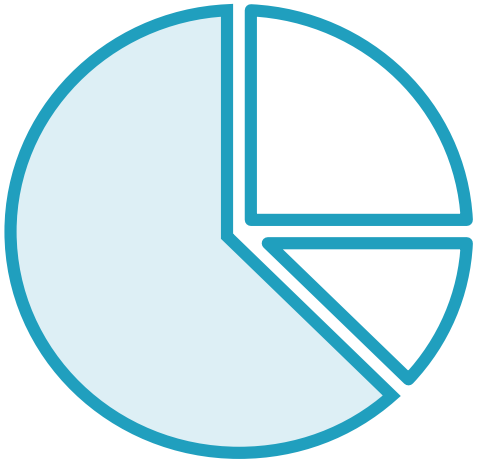
Data Preprocessing Using Scikit-Learn

Data Preprocessing with Scikit-Learn

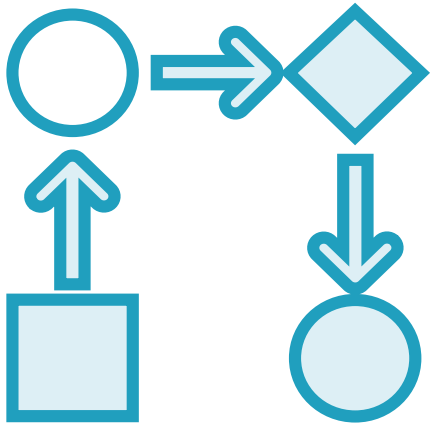


Two Hats of a Data Professional

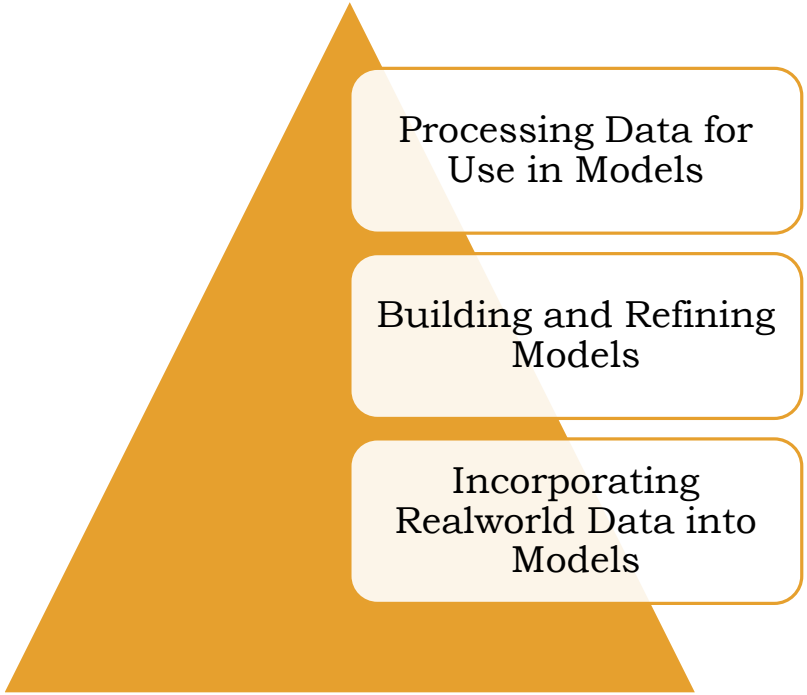
Find The Dots: Identify Important Elements In A Dataset



Connect The Dots: Explain Those Elements Via Relationships With Other Elements



Essential Steps in Connecting the Dots



Standardizing Data

$$\begin{bmatrix} X_{11} & & X_{1k} \\ X_{21} & \dots & X_{2k} \\ \dots & & \dots \\ X_{n1} & & X_{nk} \end{bmatrix}$$

$$\text{avg}(X_1) \quad \dots \quad avg(X_k)$$

$$\text{stdev}(X_1) \quad \dots stdev(X_k)$$

Standardizing Data

$$\begin{bmatrix} \frac{X_{11} - avg(X_1)}{stdev(X_1)} & \cdots & \frac{X_{1k} - avg(X_k)}{stdev(X_k)} \\ \cdots & \cdots & \cdots \\ \frac{X_{n1} - avg(X_n)}{stdev(X_n)} & & \frac{X_{nk} - avg(X_k)}{stdev(X_k)} \end{bmatrix}$$

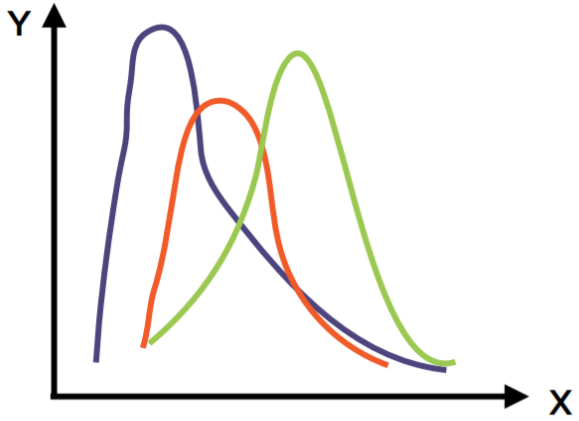
Each column of the standardized data has mean 0 and variance 1

Standardizing Data

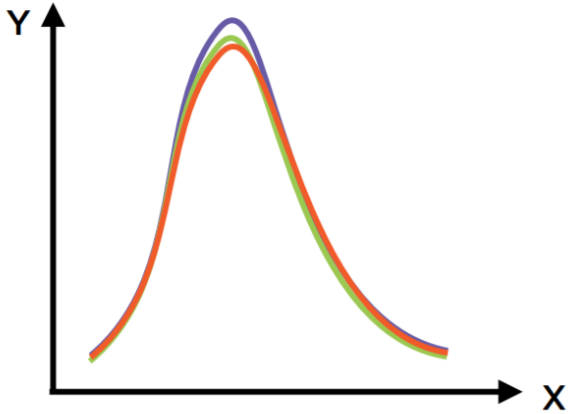
$$Z = \frac{X_i - mean(X)}{stdev(x)}$$

Standardization operates column-by-column and yields features with zero mean and unit variance

Standardizing Data



Before



After

Mean is a measure of central tendency and standard deviation is a measure of dispersion

Robust Standardization

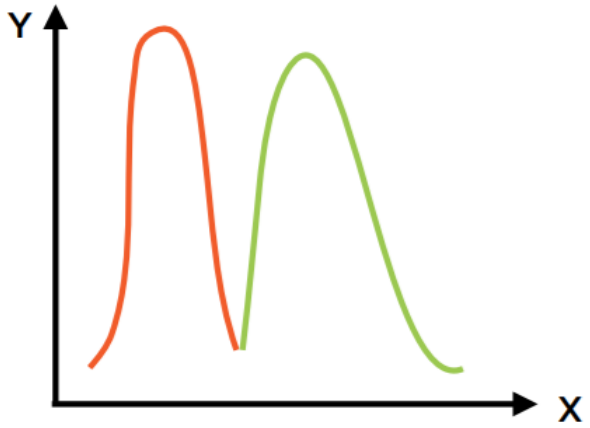
$$Z = \frac{X_i - median(X)}{stdev(x)}$$

Median is also a measure of central tendency and inter-quartile range is also measure of dispersion

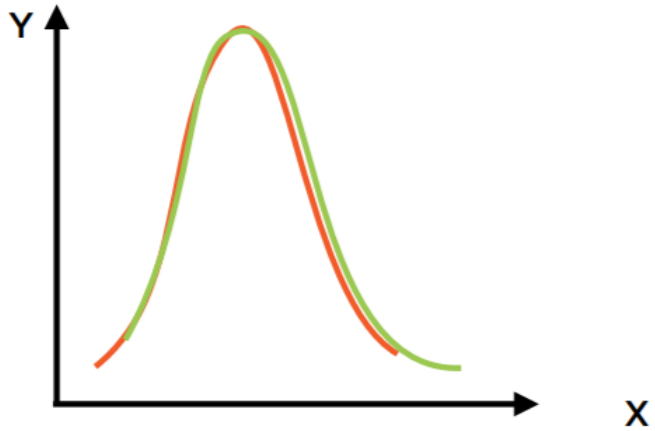
Output does not change much due to outliers

```
from sklearn.preprocessing import RobustScaler
```

Robust Standardization



Before



After

Data Range

we can also scale data by compressing it into a fixed range. One of the biggest use cases for this is compressing data into the range [0, 1].

$$x_P = \frac{x - d_{min}}{d_{max} - d_{min}}$$

from sklearn.preprocessing **import** MinMaxScaler

Normalization

Normalization Process of scaling input vectors individually to unit norm (unit magnitude), often in order to simplify cosine similarity calculations

from sklearn.preprocessing import Normalizer

$$X_{L2} = \left[\frac{x_1}{\ell}, \frac{x_2}{\ell}, ..., \frac{x_m}{\ell}\right], \text{where } \ell = \sqrt{\sum_{i=1}^{m} x_i^2}$$

Data Imputation

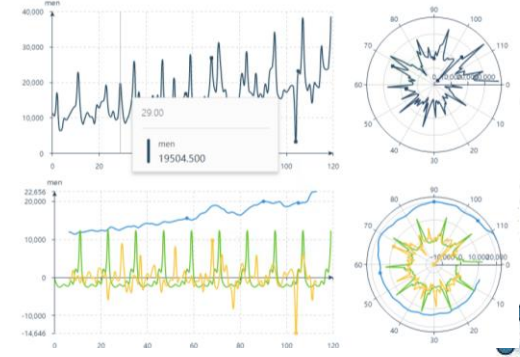
In real life, we often have to deal with data that contains missing values. Sometimes, if the dataset is missing too many values, we just don't use it.

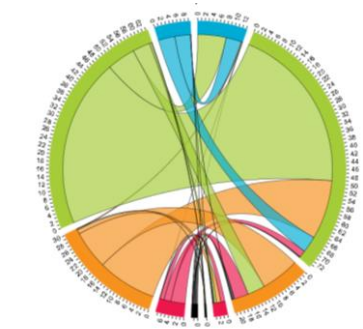
There are many different methods for data imputation. In scikit-learn, the **SimpleImputer** transformer performs four different data imputation methods.

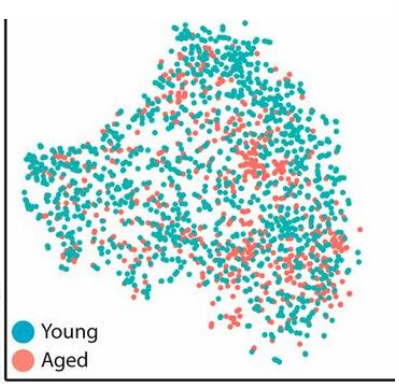
The four methods are:

1. Using the mean value
2. Using the median value
3. Using the most frequent value
4. Filling in missing values with a constant

```
from sklearn.impute import SimpleImputer
```







matplotlib

Python Plotting Library

The Matplotlib Library

Matplotlib is a data visualization tool built upon the Numpy and SciPy framework. It was created by John Hunter in 2002

It is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

- There are several toolkits which are available that extend python matplotlib functionality. Some of them are separate downloads, others can be shipped with the matplotlib source code but have external dependencies.

Contd..

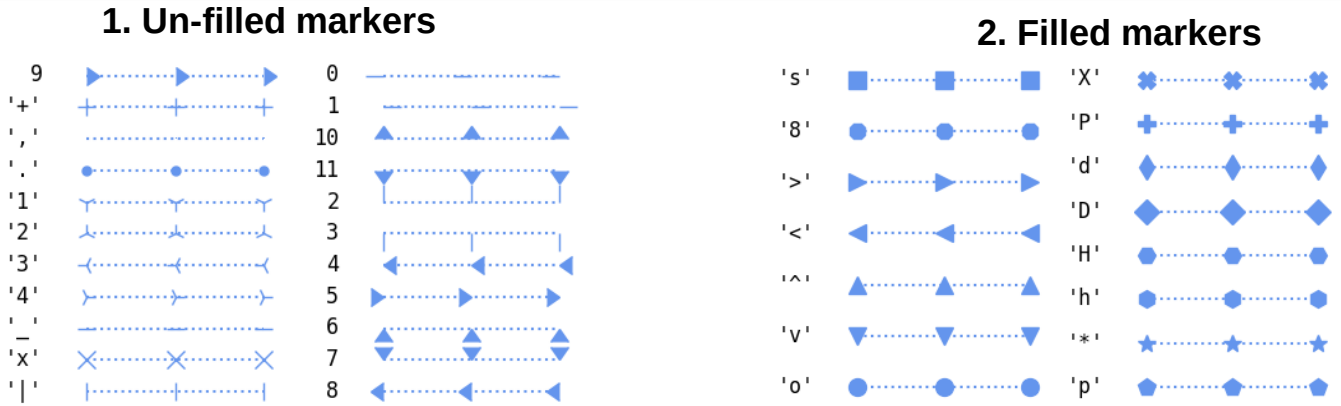
- **Basemap**: It is a map plotting toolkit with various map projections, coastlines and political boundaries.
- **Cartopy**: It is a mapping library featuring object-oriented map projection definitions, and arbitrary point, line, polygon and image transformation capabilities.
- **Excel tools**: Matplotlib provides utilities for exchanging data with Microsoft Excel.
- **Mplot3d**: It is used for 3-D plots.

Grids, Axes, Plots

- **Grid -** The **grid**() function of **axes** object sets visibility of grid inside the figure to on or off. You can also display major / minor (or both) ticks of the grid. Additionally color, linestyle and linewidth properties can be set in the **grid**() function.
- **Axes -** Axes object is the region of the image with the data space. A given figure can contain many Axes, but a given Axes object can only be in one Figure. The Axes contains two (or three in the case of 3D) Axis objects
- **Plots -** The ability to present data in a graphical or pictorial format in an attempt to help people understand its significance is known as data visualization skills. Data visualization skills simply refer to the ability to identify or uncover patterns etc.

Markers and Colors

Markers: there are 2 types of makers



Colors: Commands which take color arguments can use several formats to specify the colors. For the basic built-in colors, you can use a single letter

- b: blue
- c: cyan
- k: black
- g: green
- m: magenta
- w: white
- r: red
- y: yellow

matplotlib

Types of Plots

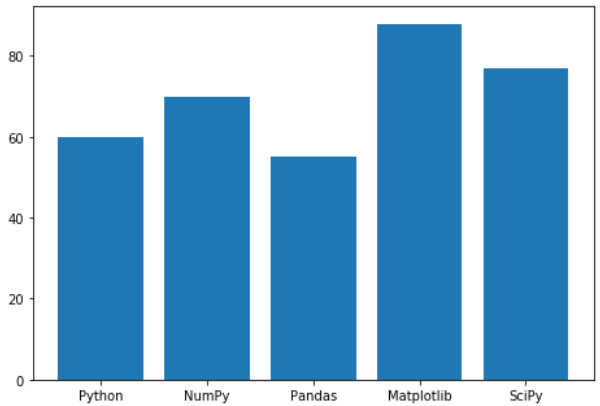
The various types of plots available in matplotlib are

1. Line plot -
2. Scatter Plot
3. Histogram
4. Bar Graph
5. Pie Chart
6. Box Plot
7. Contour plot
8. Polar Plot
9. Log Plot
10. Stream Plot

Bar Graphs in Matplotlib

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
Subjects = ['Python', 'NumPy', 'Pandas', 'Matplotlib', 'SciPy']
Students = [60,70,55,88,77,33]
ax.bar(Subjects, Students)
plt.show()
```

Output →

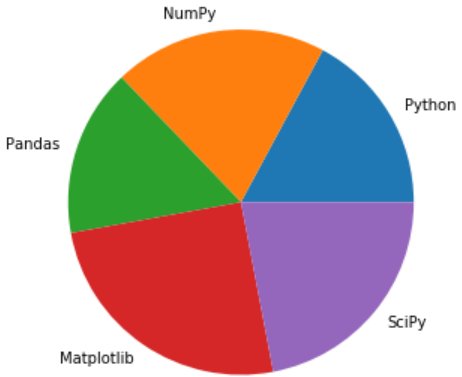


matplotlib

Pie Chart in Matplotlib

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
Subjects = ['Python', 'NumPy', 'Pandas', 'Matplotlib', 'SciPy']
Students = [60,70,55,88,77,33]
ax.bar(Students, labels = Subjects)
plt.show()
```

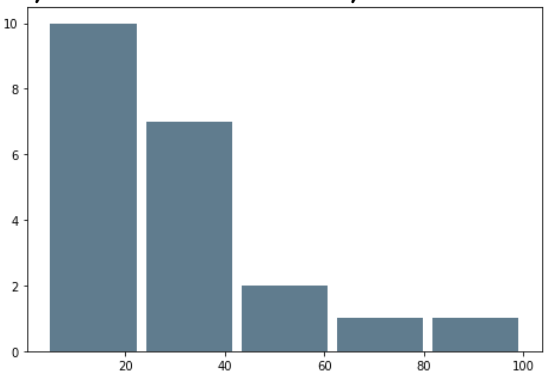
Output →



Pie Chart in Matplotlib

```
import matplotlib.pyplot as plt

fig = plt.figure()

ax = fig.add_axes([0,0,1,1])

x = [21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]

plt.hist(x, bins=5, rwidth=0.9, color='#607c8e')

plt.show()
```

Output →



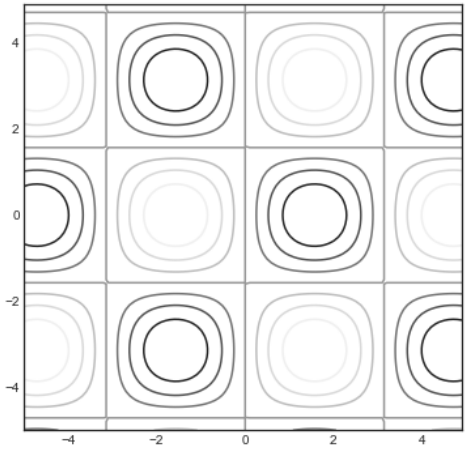
Skill AP APSSDC

matplotlib

Contour Plot

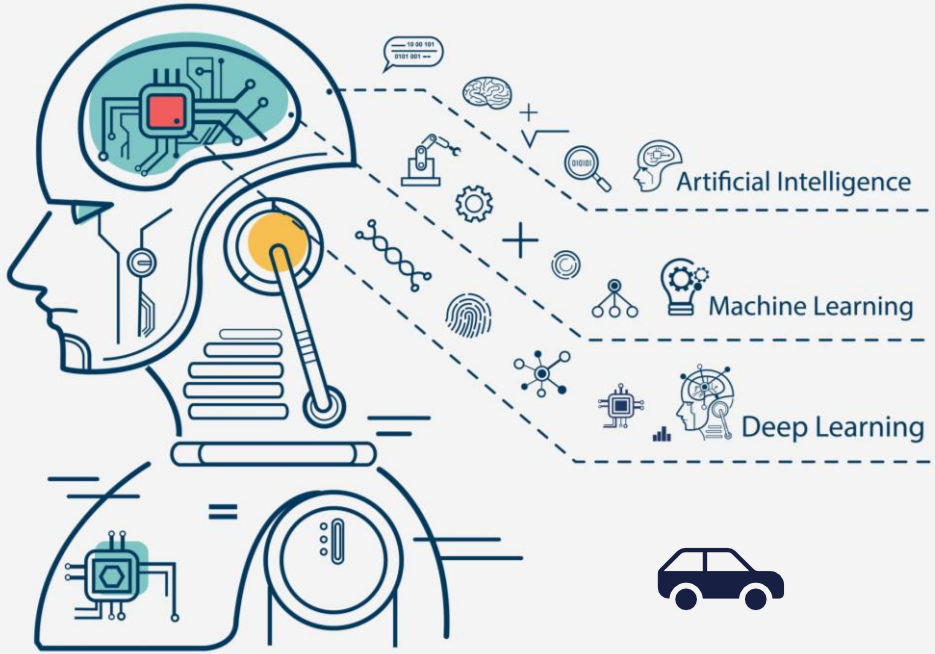
```
x = np.arange(-5.0, 5.0, 0.1)
y = np.arange(-5.0, 5.0, 0.1)
X, Y = np.meshgrid(x, y)
Z = np.sin(X)*np.cos(Y)
fig, ax = plt.subplots(figsize=(6,6))
ax.contour(X,Y,Z)
plt.show()
```

Output:



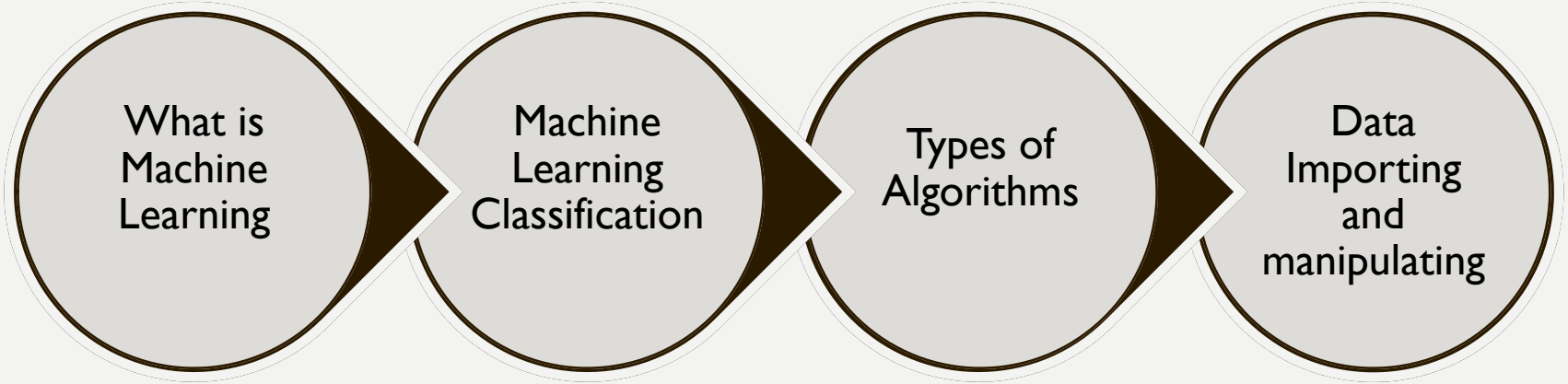
APSSDC

Andhra Pradesh State Skill Development Corporation



MACHINE LEARING USING PYTHON

DAY1 AGENDA


What is Machine Learning
Machine Learning Classification
Types of Algorithms
Data Importing and manipulating

WHAT IS THIS FRUIT?

WHAT IS THIS FRUIT?

APPLE/Fruit

- Color, Shape, Seeing, Smell, eating, Weight
- Green, Heart Symbol, 150grms – 500grms

WHAT IS THIS FRUIT?

- Apple, Half Apple

- Color, Shape, Seeing, Smell, eating, Weight
- Red, Heart Symbol, 150grms – 500grms
- Seeds, Inner Color, Seed Location
- Small, White, Center

QUIZ

Color	Shape	Weight	Size	What is it?
Red	Heart Symbol	100grms	2.5”	
Red	Heart Symbol	18grms	1.375”	
Green	Heart Symbol	150grms	2.7”	
Red	Heart Symbol	223grms	3.25”	

QUIZ

Color	Shape	Weight	Size	What you can do?
Red	Heart Symbol	100grms	2.5”	
Red	Heart Symbol	18grms	1.375”	
Green	Heart Symbol	150grms	2.7”	
Orange	Circle	223grms	3.25”	
green	curved	75grms	3”	
Orange	Oval	150grms	3.5”	
green	circular	80grms	2.5”	
red	oval	550grms	5”	
green	circular	5grams	0.5”	
red	oval	50grms	2”	

QUIZ

Color	Shape	Weight	Size	What is it?
Red	Heart Symbol	10000 grms	2.5”	APPLE

WHAT MACHINE LEARNING ?

"A computer program is said to learn from experience(input data) **E** with respect to some class of tasks(Target) **T** and performance measure **P**, if its performance at tasks in T, as measured by P, improves with experience E."

— Tom Mitchell, Professor at Carnegie Mellon University

- Computer Program → Past Experience(Data) → W. r. to Some task T → with perromance P
- P → T →E

WHAT IS **ARTIFICIAL INTELLIGENCE, MACHINE LEARNING** AND **DEEP LEARNING**

ARTIFICIAL INTELLIGENCE

Artificial Intelligence (**AI)** is the science of making things smart. Can be defined as:

"Human intelligence exhibited by machines"

A broad term for getting computers to perform human tasks. The scope of AI is disputed and constantly changing over time.

AI: COMMON USE CASES

- Object recognition
- Speech recognition / Sound detection
- Natural Language Processing / Sentiment analysis
- Creative (e.g. Style Transfer – Learning to draw an image in the style of an artist)
- Prediction – given some inputs, what is the expected output for unseen examples
- Translation between languages
- Restoration / Transformation – e.g. taking an image and using ML to figure out what should be there, or generating faces based on what it knows face to be.
- Some AI Examples

MACHINE LEARNING

- Machine Learning (**ML**) can be defined generally as:

"An approach to achieve AI through systems that can learn from experience to find patterns in a set of data"

ML involves **teaching a computer to recognize patterns by example, rather than programming it with specific rules.** These patterns can be found within data. In other words, ML is about creating algorithms (or a set of rules) that learn complex functions (or patterns) from data and make predictions on it –a form of "narrow AI"

DEEP LEARNING

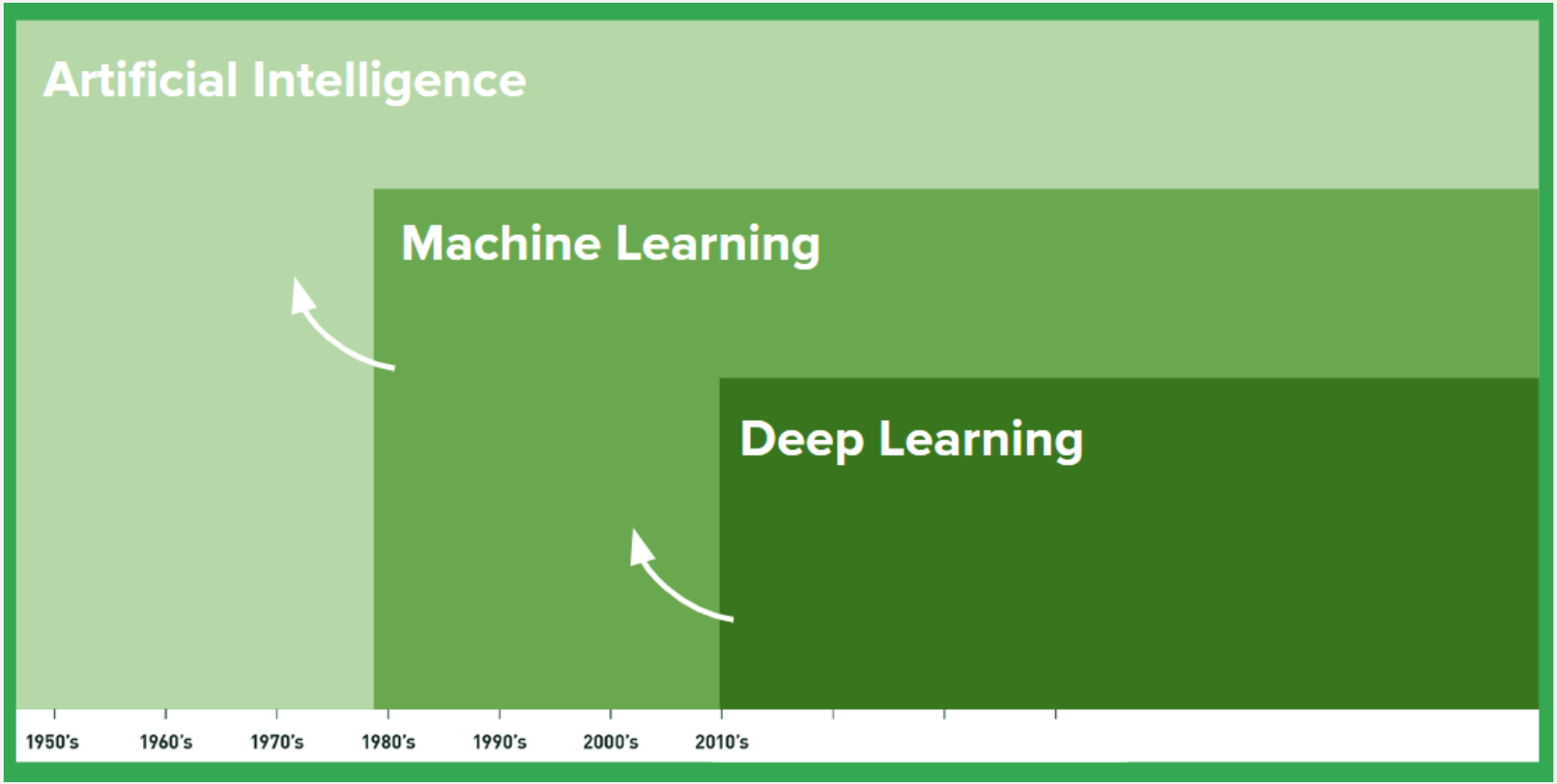
- Deep Learning (**DL** from here on) can be defined generally as:

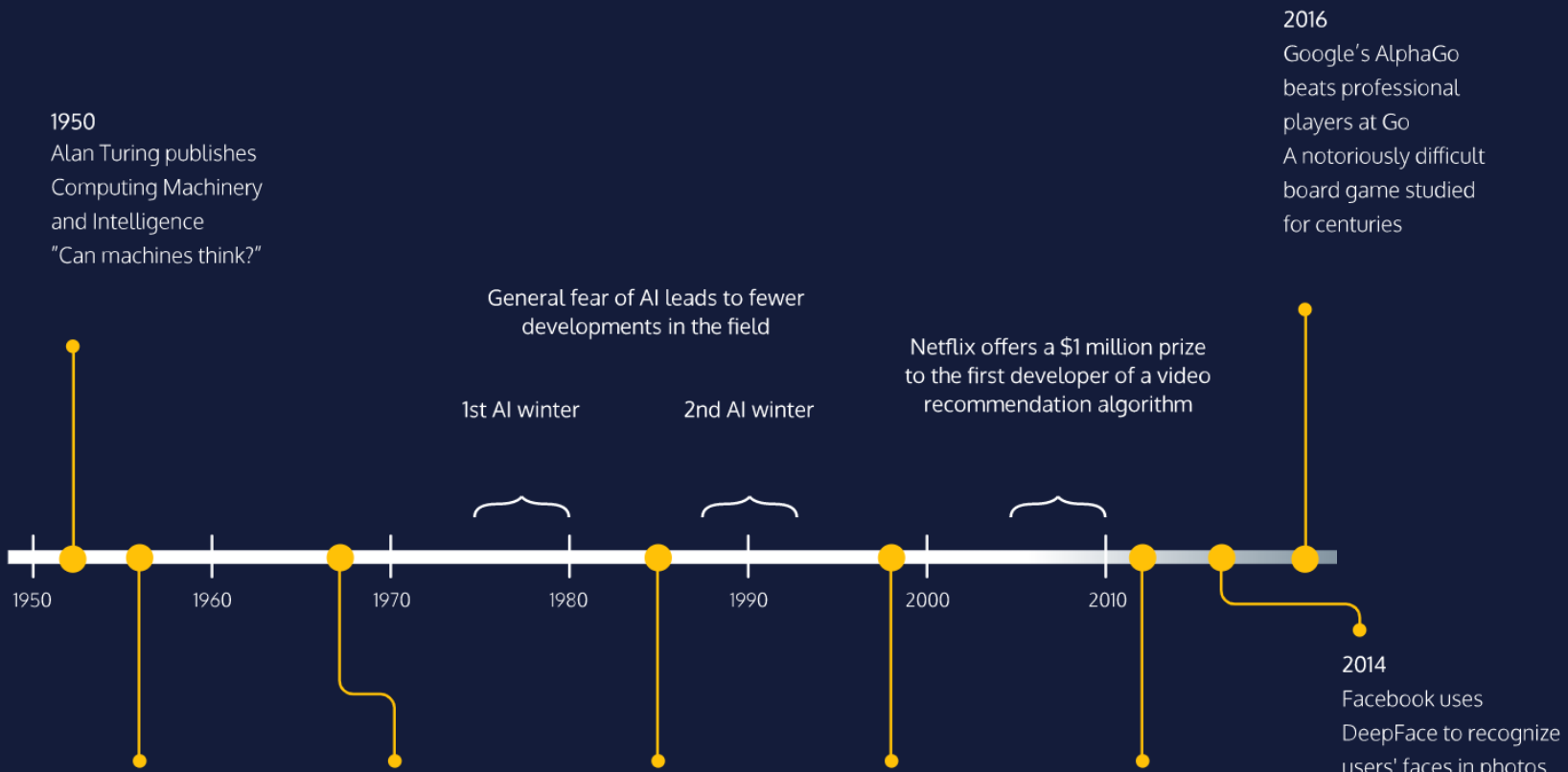
"A technique for implementing Machine Learning"

One such DL technique is a concept known as **deep learning Neural networks (DNNs)** which you may have heard of.

Essentially DL in the context of DNNs is where the code structures you write **are arranged in the layers that loosely mimic the human brain, learning patterns of patterns.**

SUMMARY


Artificial Intelligence
Machine Learning
Deep Learning
1950's
1960's
1970's
1980's
1990's
2000's
2010's


1950
Alan Turing publishes Computing Machinery and Intelligence "Can machines think?"
General fear of AI leads to fewer developments in the field
1st AI winter
2nd AI winter
Netflix offers a $1 million prize to the first developer of a video recommendation algorithm
2016
Google's AlphaGo beats professional players at Go
A notoriously difficult board game studied for centuries
1950
1960
1970
1980
1990
2000
2010
1957
The perceptron algorithm (to imitate a single neuron) is created (LESSON)
1967
Unknown data can be assigned labels with the K-Nearest Neighbors algorithm (LESSON)
1985
First speech recognition algorithm using natural language processing
1998
MNIST database is released and has since become a benchmark for handwriting recognition (PROJECT)
2011
IBM Watson beats human competitors at Jeopardy!
2014
Facebook uses DeepFace to recognize users' faces in photos

FEW OTHER DEFINITIONS

"Machine learning is the hot new thing"

— John L. Hennessy, President of Stanford (2000–2016)

"A breakthrough in machine learning would be worth ten Microsoft"

— Bill Gates, Microsoft Co-Founder

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed"

— Arthur Samuel's

MACHINE LEARNING TYPES

Supervised Learning

- Makes machine Learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/future
- Resolves classification and regression problems

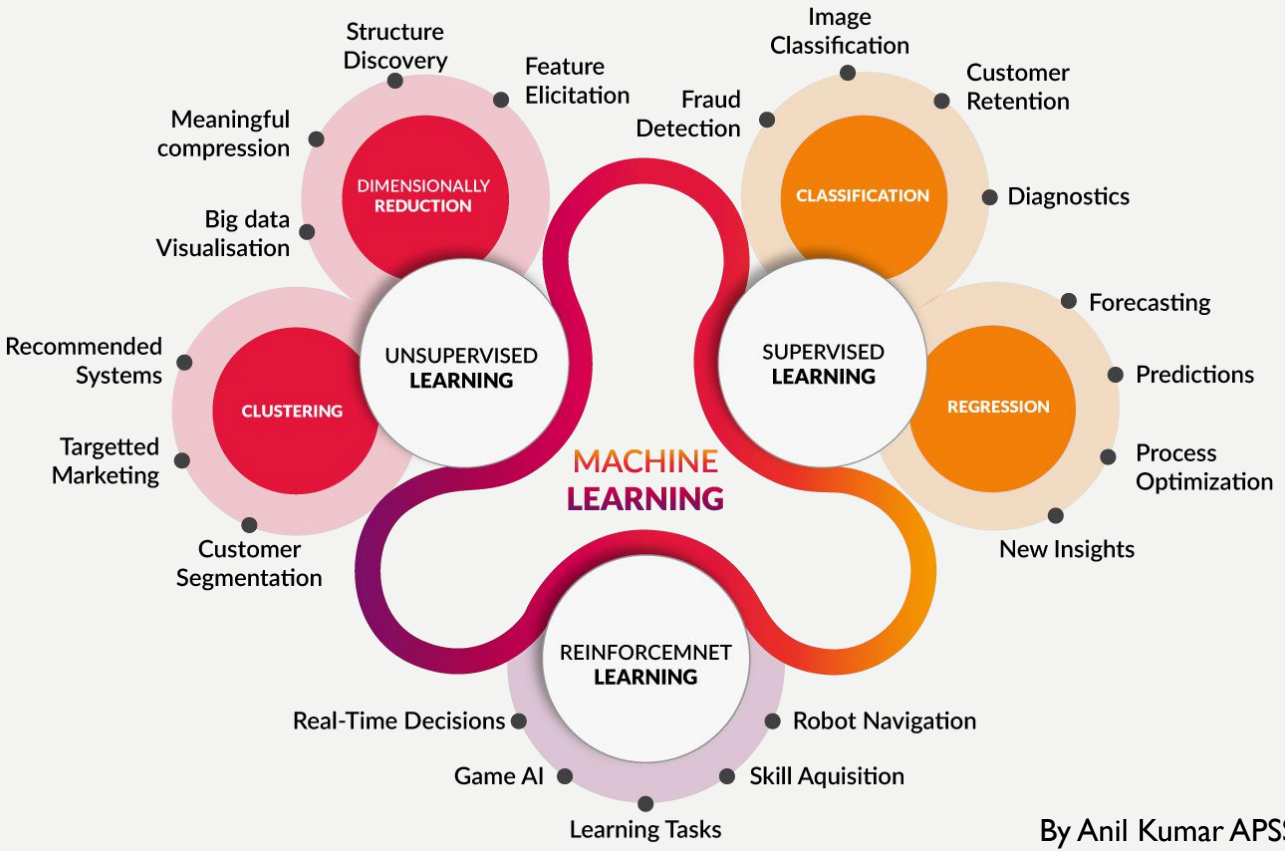
Unsupervised Learning

- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect
- Does not predict/find anything specific

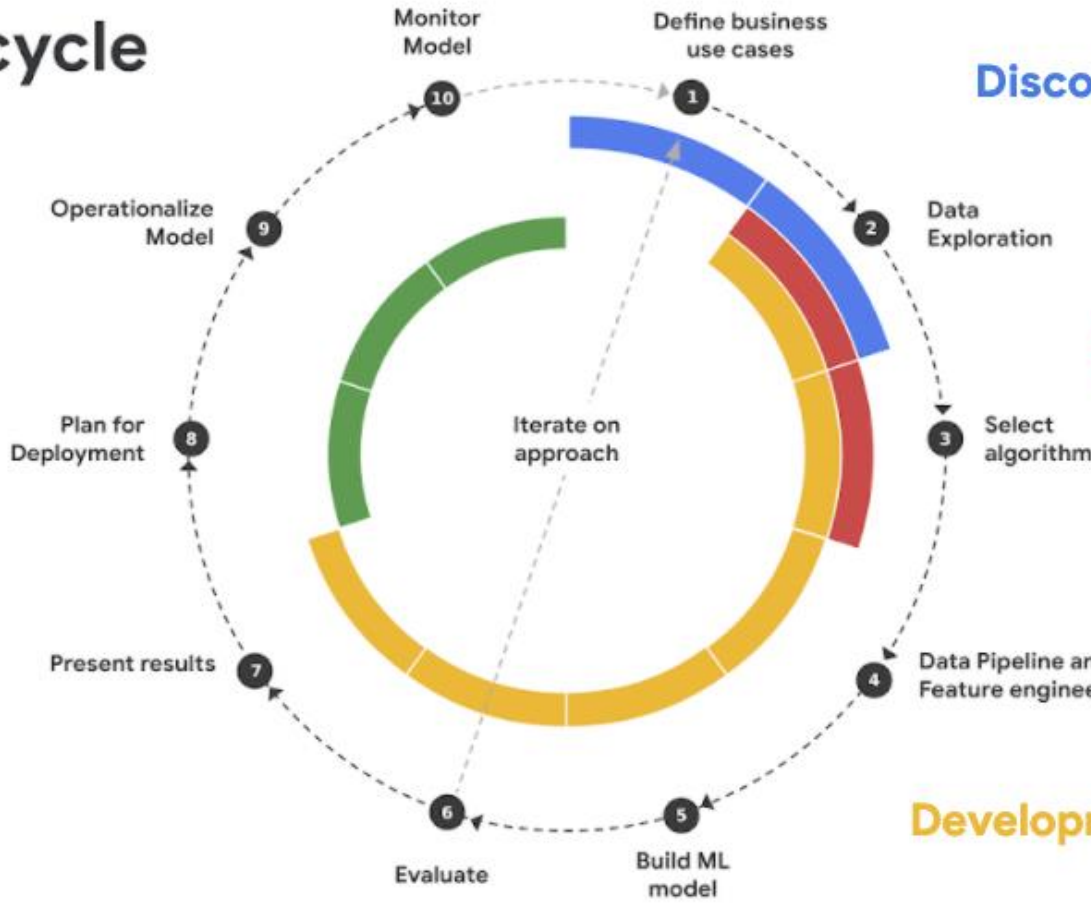
Reinforcement Learning

- An approach to AI
- Reward based learning
- Learning form +ve & +ve reinforcement
- Machine Learns how to act in a certain environment
- To maximize rewards

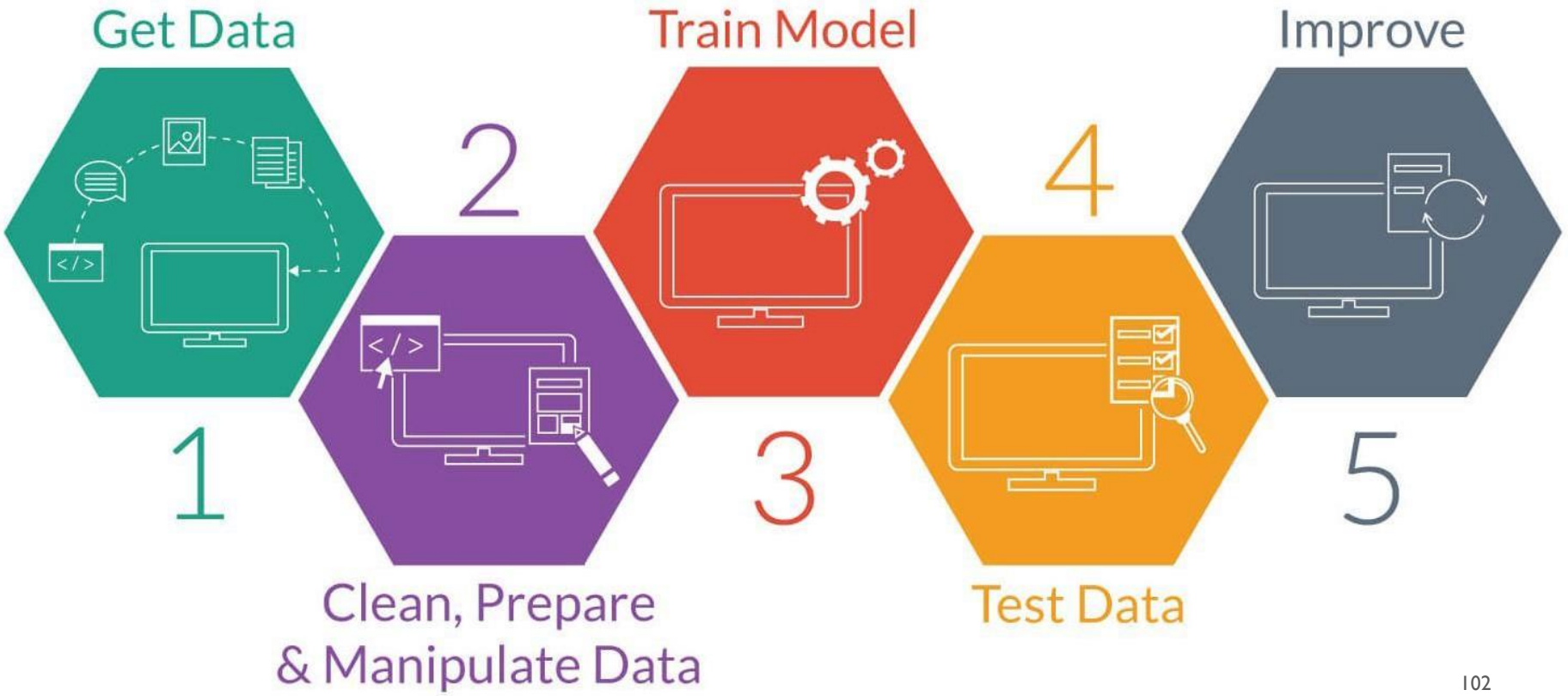
MACHINE LEARNING CATEGORIES



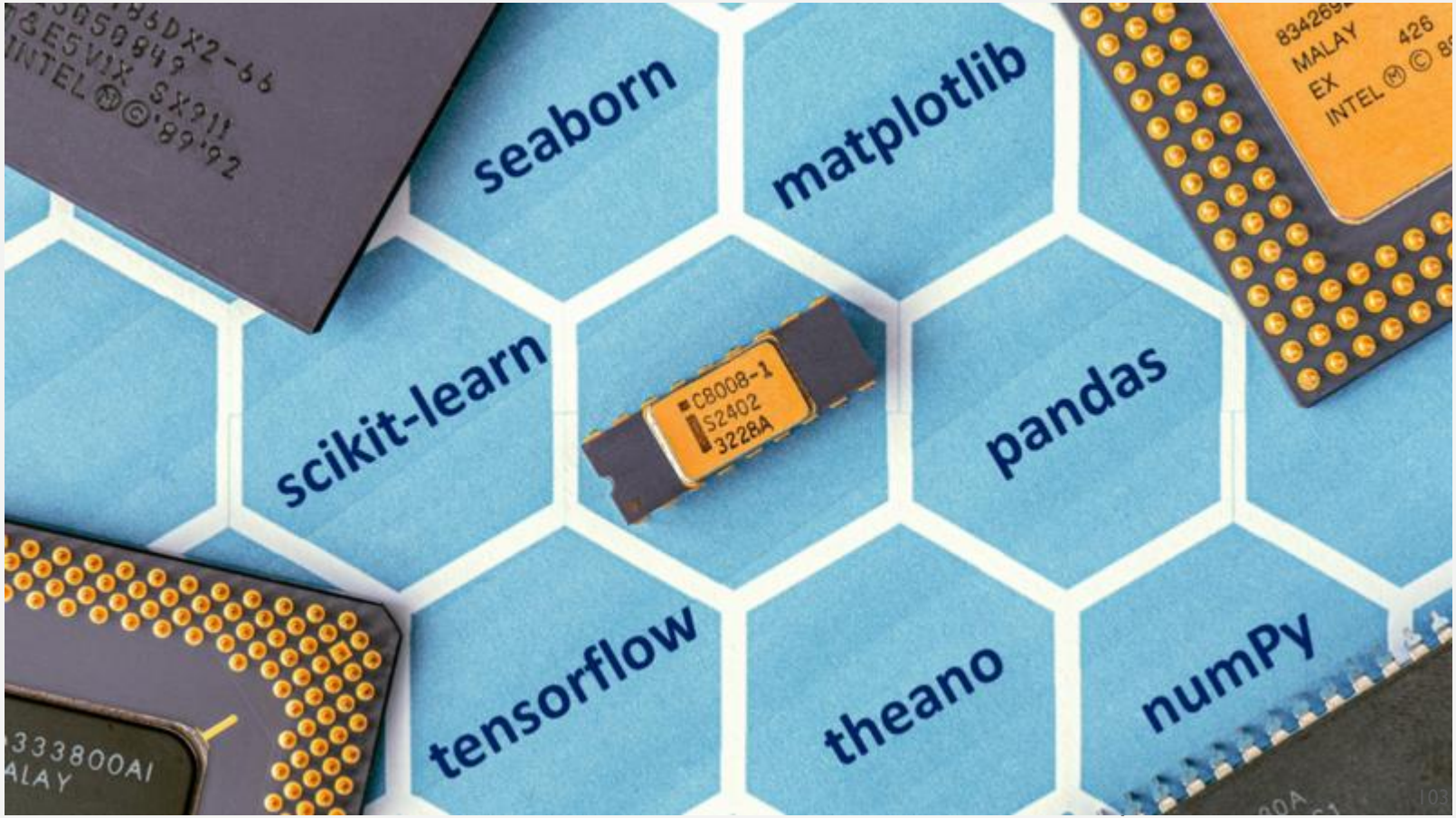
ML Lifecycle



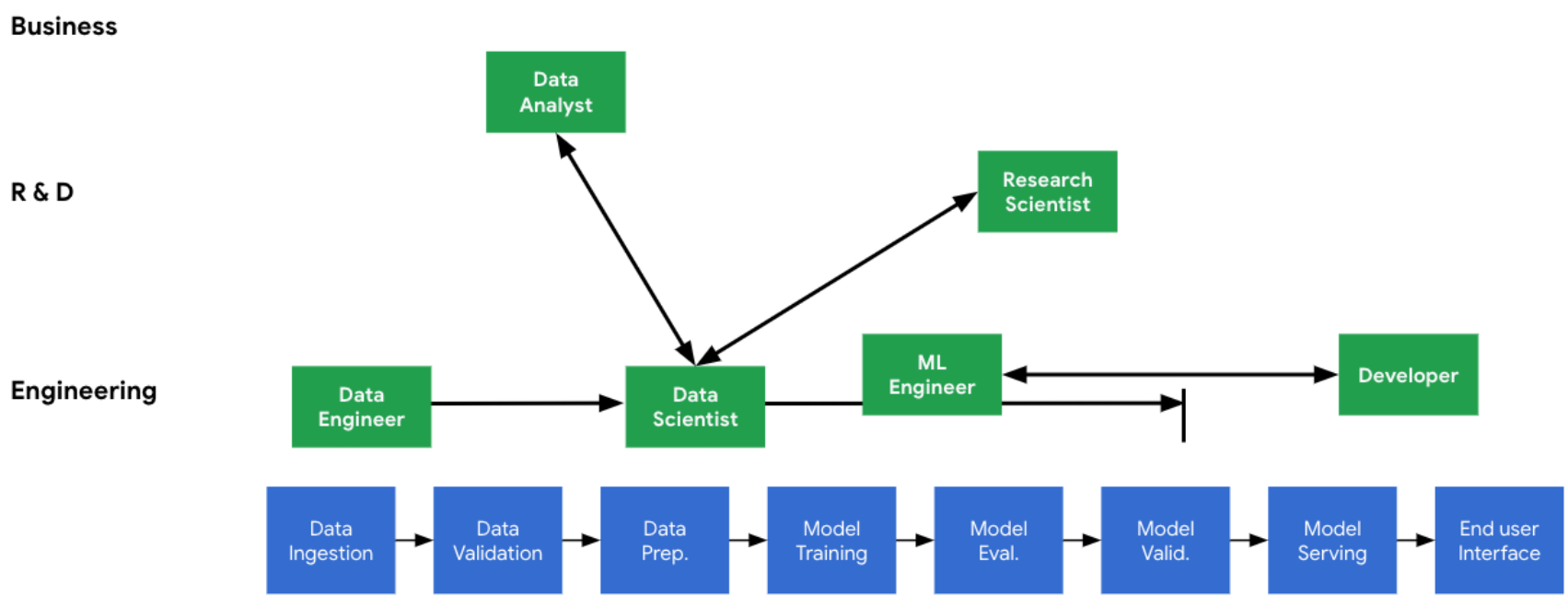
MACHINE LEARNING PROCESS



PACKAGES FOR ML IN PYTHON



THE NEED FOR MACHINE LEARNING DESIGN PATTERNS



HOW TO CHOOSE DATA TO TRAIN THE MODEL

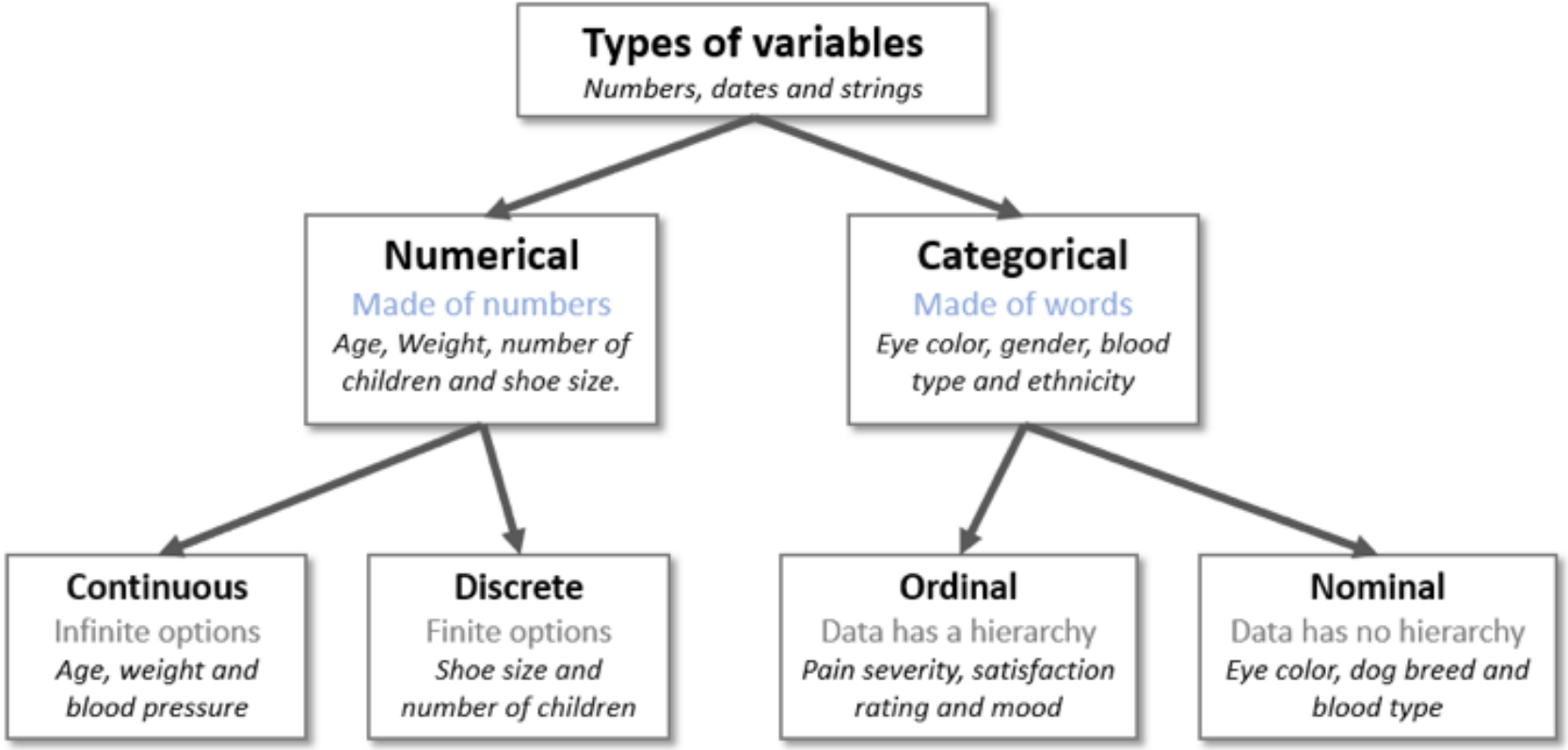
DATA CLASSIFICATION IN REAL WORLD

- Labeled, non labeled → Data
- Categorical, Numerical → stastics

- Structures → If data is having structure → CSV, Excel, DB, HTML Tables, TSV,
- Semi-structured, → XML, JSON, … = {"Key":"Value"}
- Unstructured → Text Files, PPT, Video, Images, Word, Audio,

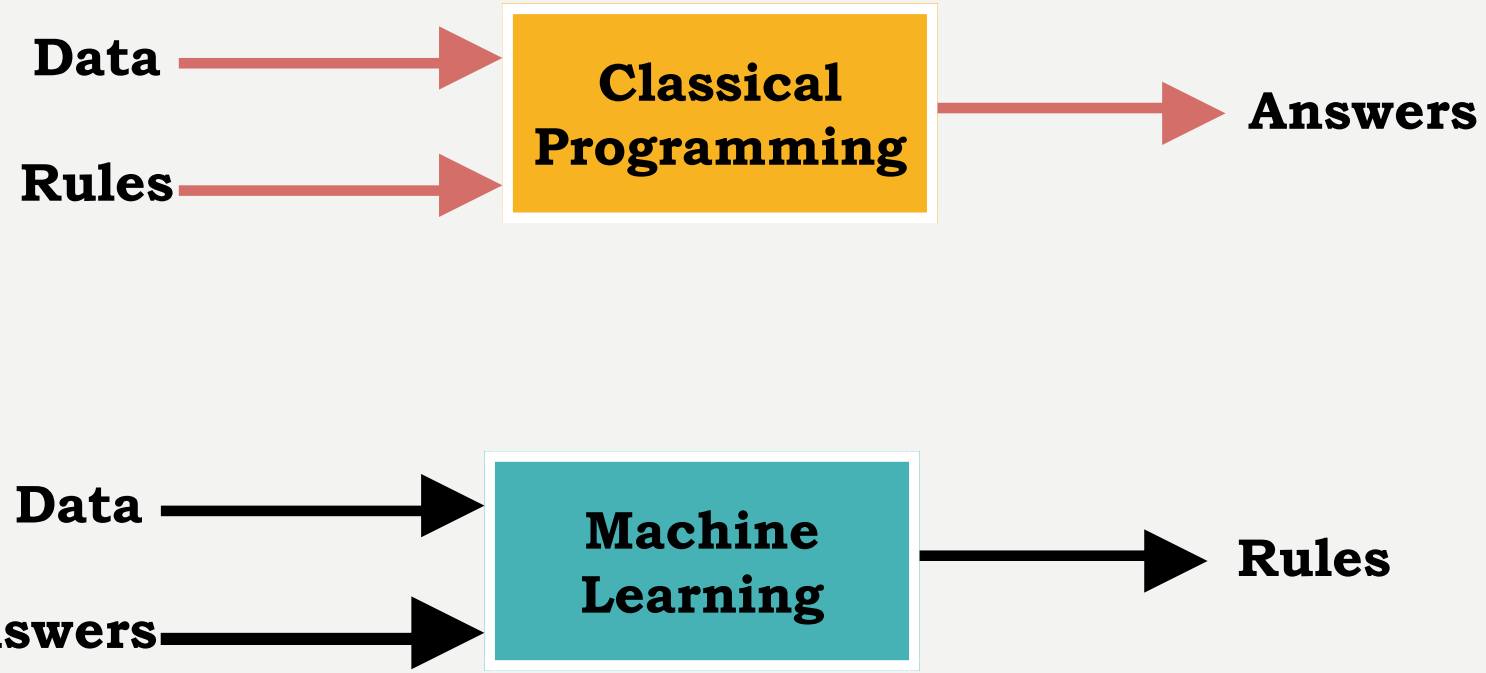
- Def ML
- AI, ML, DL
- Classification of ML
- Life Cycle of ML
- Classification real world data

TYPES OF VARIABLES

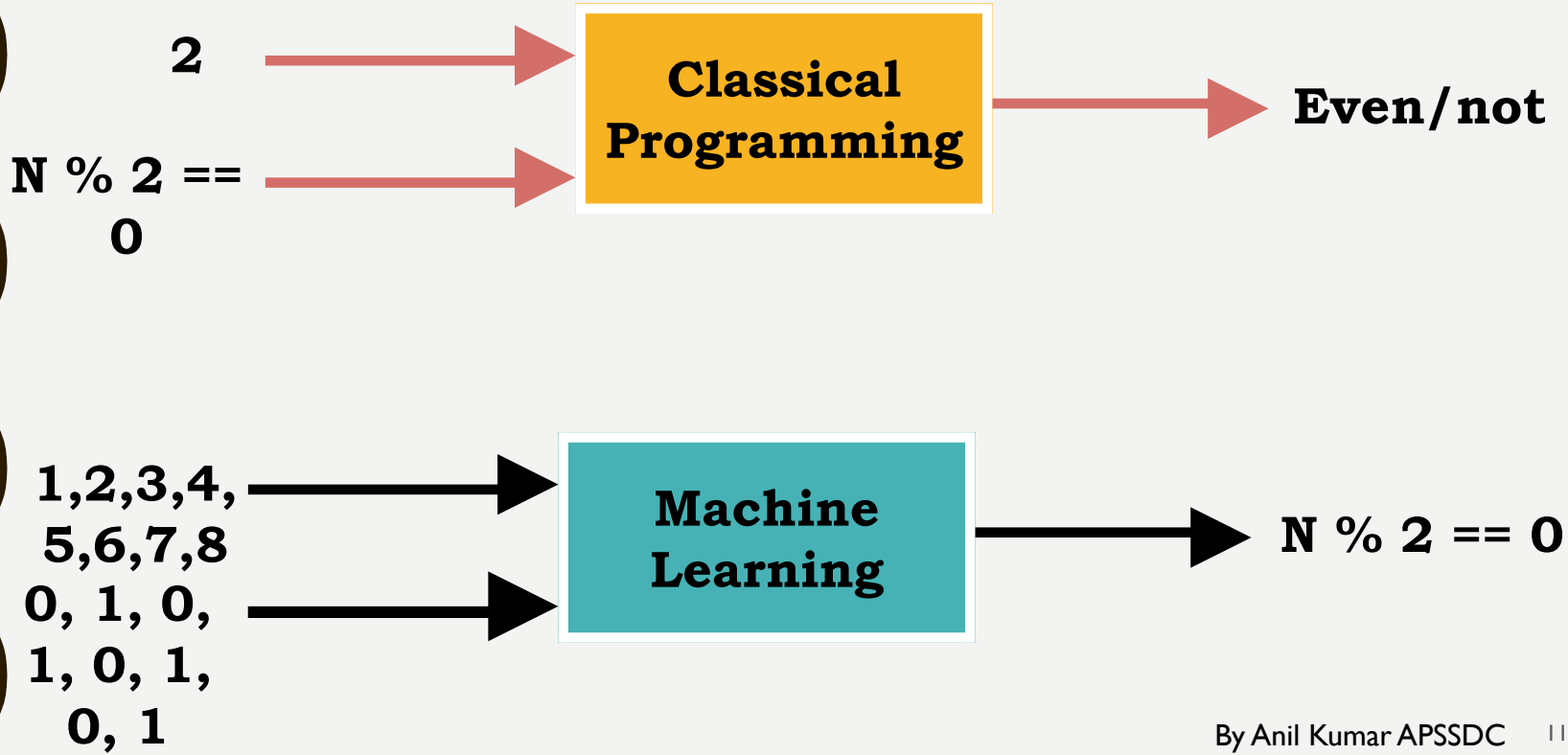


https://en.wikipedia.org/wiki/Statistical_data_type

CLASSICAL PROGRAMMING VS MACHINE LEARNING



CLASSICAL PROGRAMMING VS MACHINE LEARNING



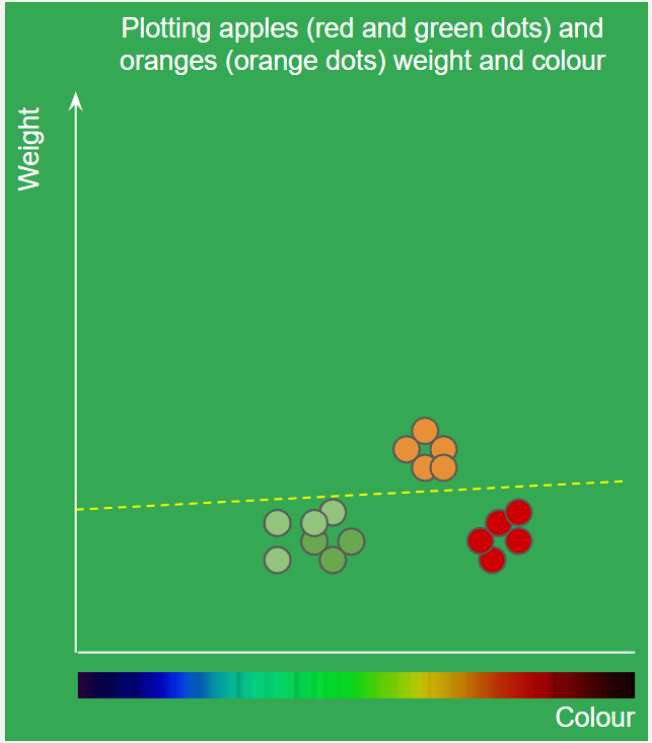
FEATURES / ATTRIBUTES

- **Features (aka attributes) are used to train an ML system.** They are the properties of the things you are trying to learn about.

FEATURES / ATTRIBUTES

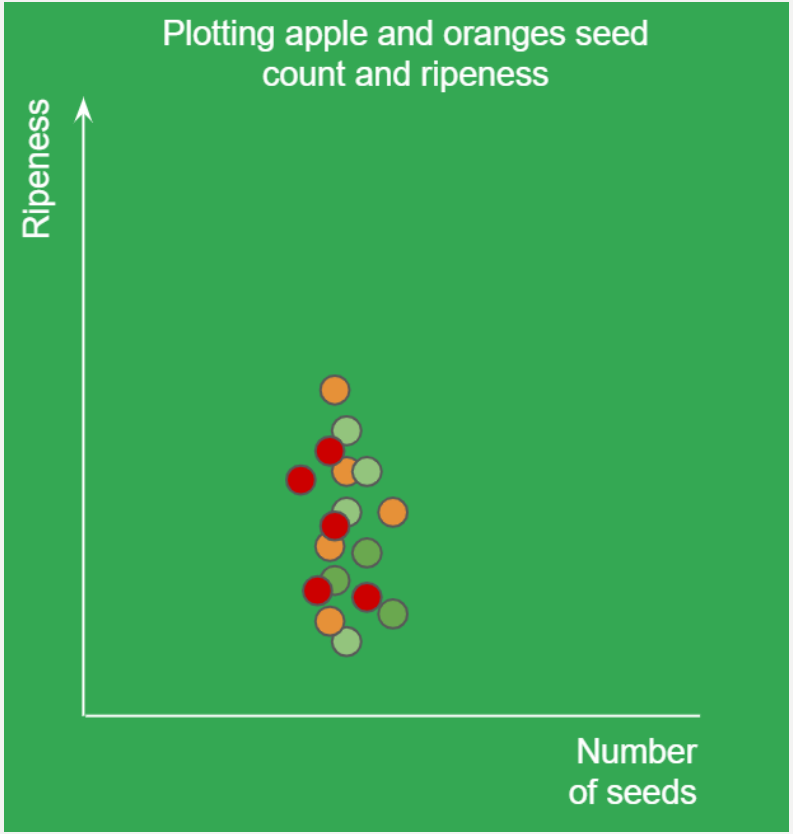
Taking fruit as an example. Features of a fruit might be weight and color. 2 features, would mean there are 2 dimensions. A 2D system may be plotted on a graph if features are represented in a numerical way.

In the plot on the right, the ML system can learn to split the data up with a line to separate apples from oranges. This **can now be used to make future classifications when we plot new points the system has not seen (anything above is orange, below is apple)**



FEATURES / ATTRIBUTES

- **Choosing useful features can have a big impact on the quality of the ML system.** Some features may not be useful enough to separate the data points.
- In this example we take bad features of fruits(ripeness and seed count) that do not allow us to learn any distinguishing factors for the fruit.



WHAT ML CANNOT PREDICT STUFF IT DOESN'T KNOW ABOUT

Lets say you teach an ML system about animals like this:

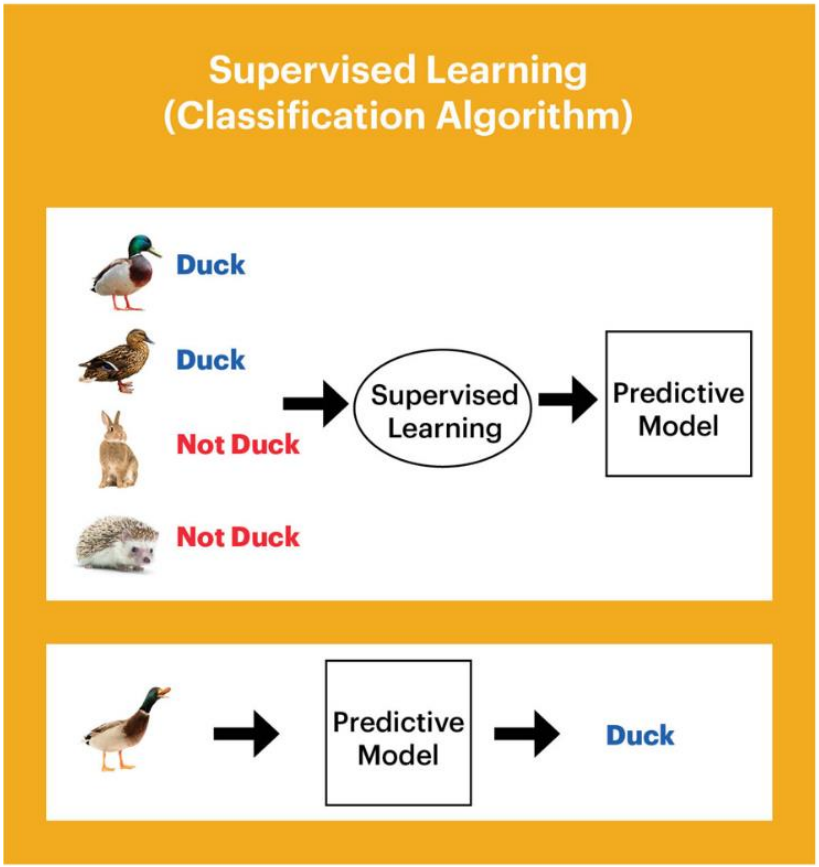
Number of Legs, Color, Weight, Animal:

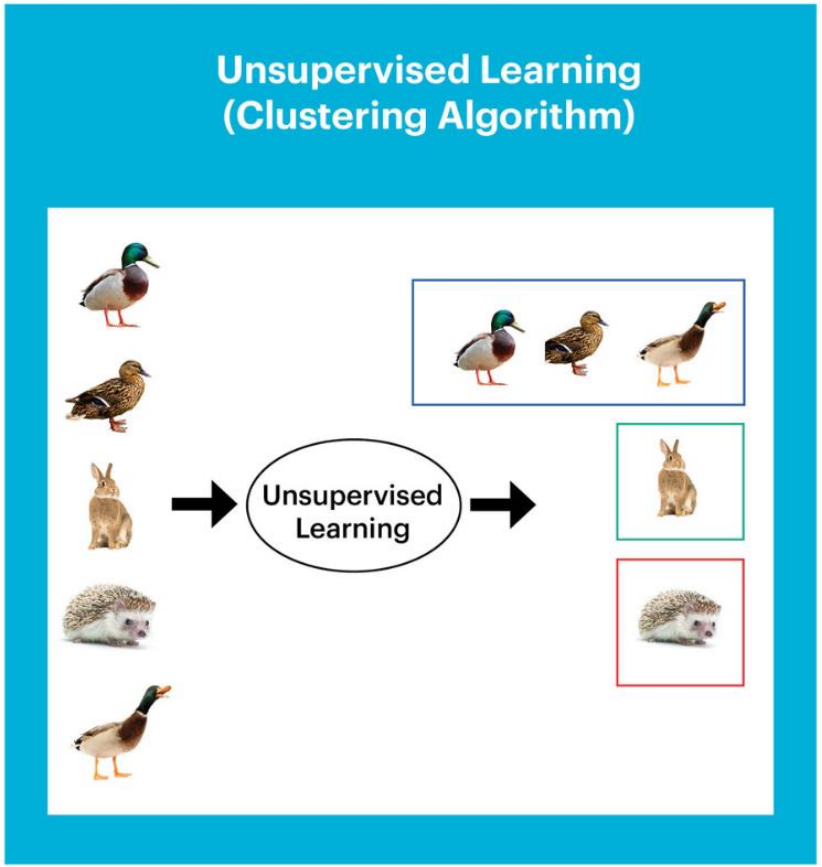
- 4, Black, 10KG, Dog
- 2, Orange, 5KG, Chicken

If you now present it with a Cow: 4 legs, black, 200KG it would predict "Dog". This is because it only knows about dogs and chickens and this was the closest match.

HOW ML SYSTEMS ARE TRAINED (LEARNING STYLE)

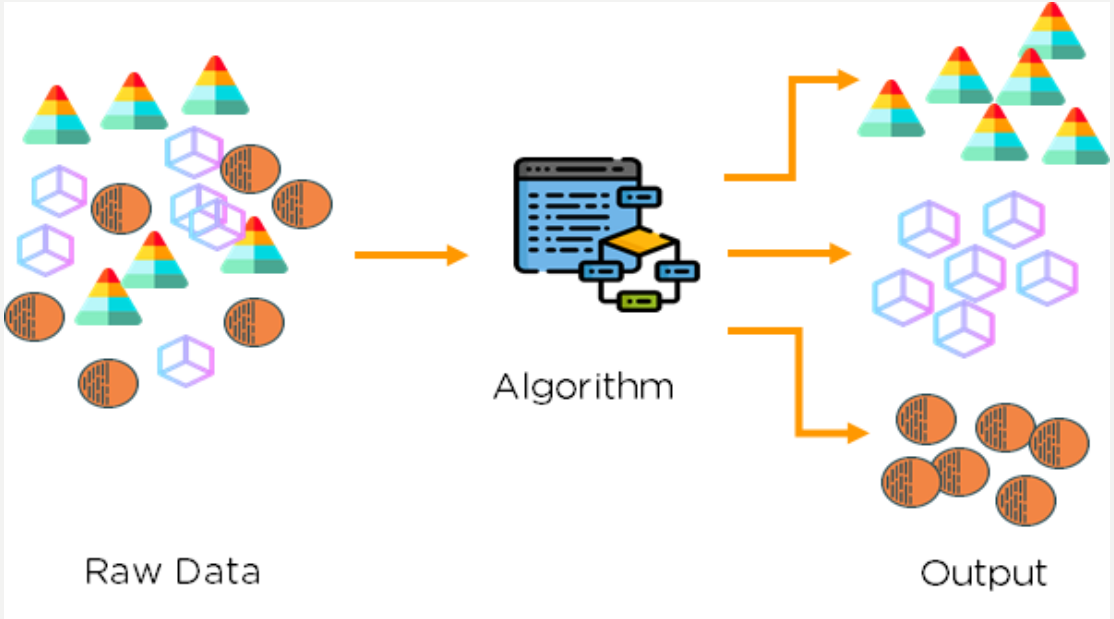
SUPERVISED VS UNSUPERVISED





UNSUPERVISED LEARNING

Unsupervised learning model learns through observation and finds structures in the data. When the model is feed data, it automatically finds patterns and relationships in the data by creating clusters in it. What it cannot do is adding labels to the cluster. Like the picture shown below.



By Anil Kumar APSSDC

MACHINE LEARNING ALGORITHMS SUPERVISED

Regression

- Linear Regression
  - Simple Linear Regression
  - Multi Linear Regression
- Polynomial Regression
  - Polynomial Regression
  - Multi Polynomial Regression

Classification

- Linear Classifiers
  - Logistic Regression
- K - Nearest Neighbors
- Decision Trees
- Random Forest
- Support Vector Machines

CLASSIFICATION VS REGRESSION

UNSUPERVISED

Clustering Types

- Hierarchical clustering
- **K-means clustering**
- DBSCAN
- Spectral clustering

Dimensionality Reduction

- **Principal Component Analysis**
- Independent Component Analysis
- randomized SVD

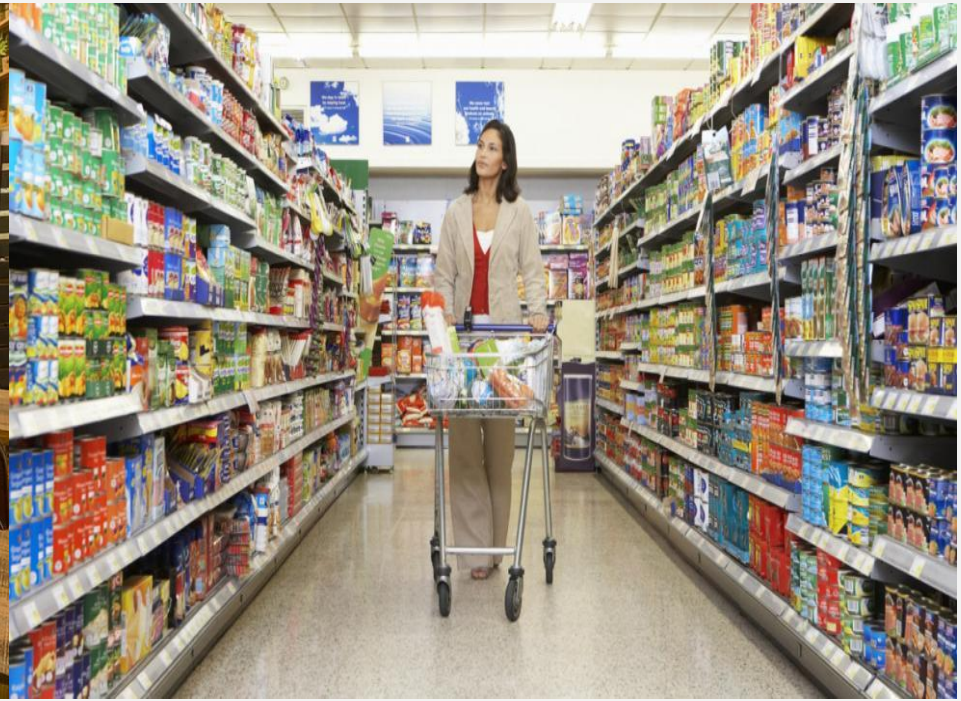
CLASSIFICATION

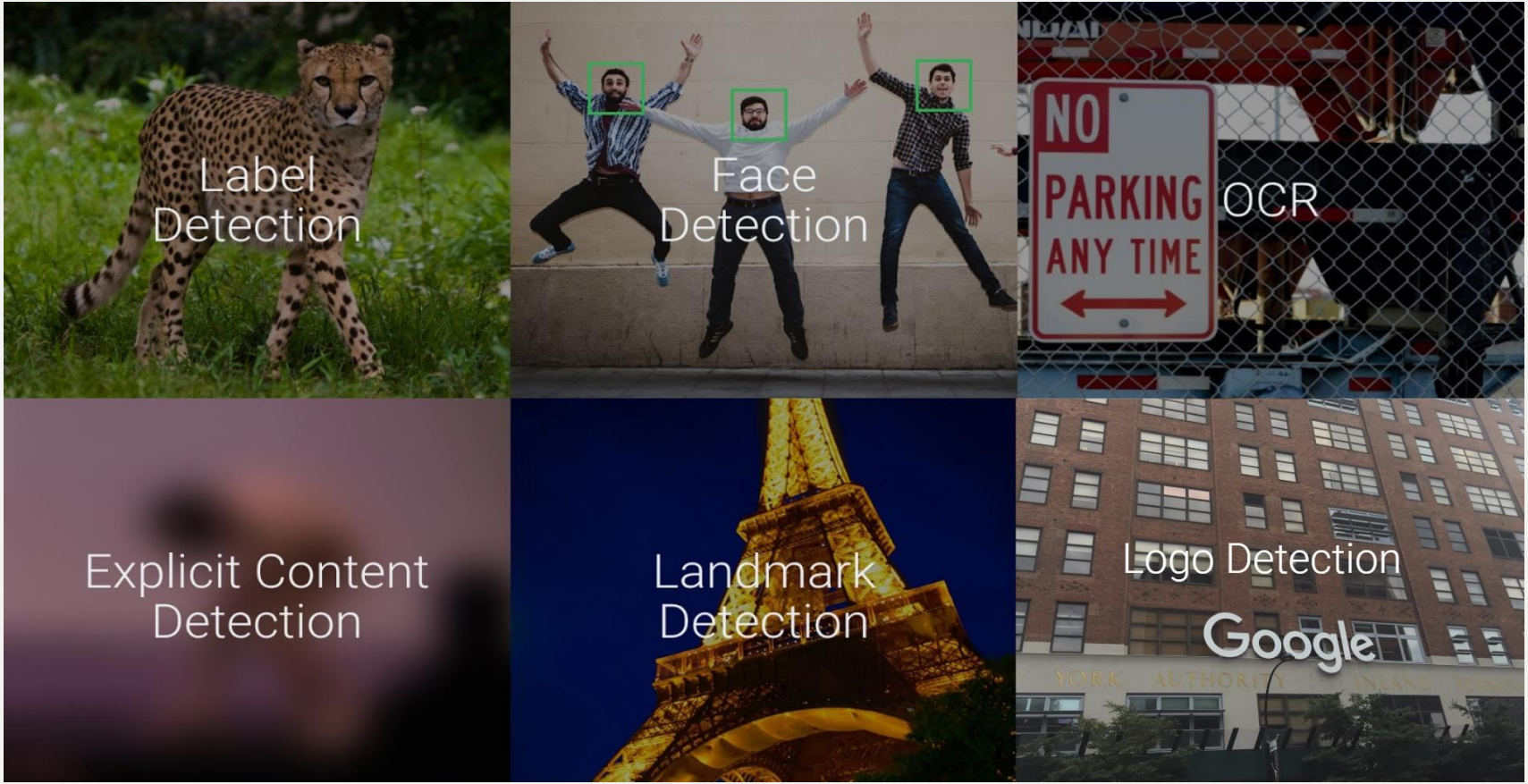
FRAUD DETECTION

HOUSE PRICE PREDICTION

STOCK PREDICTION

CUSTOMER PREDICTION




Label Detection
Face Detection
NO PARKING ANY TIME
OCR
Explicit Content Detection
Landmark Detection
Logo Detection
Google

REFERENCES

- Machine Learning in 45 minutes by Jason Mayes, Senior Creative Engineer at Google
  - **Video:** https://www.youtube.com/watch?v=X4I9QmcSEYo
  - **Slides**: https://goo.gl/fGJ8HJ