Jawaharlal Nehru Technological University Hyderabad

Kukatpally, Hyderabad - 500 085, Telangana, India

Subject 2:
Machine Learning and Deep Learning

Data Science Lifecycle and Preprocess Steps

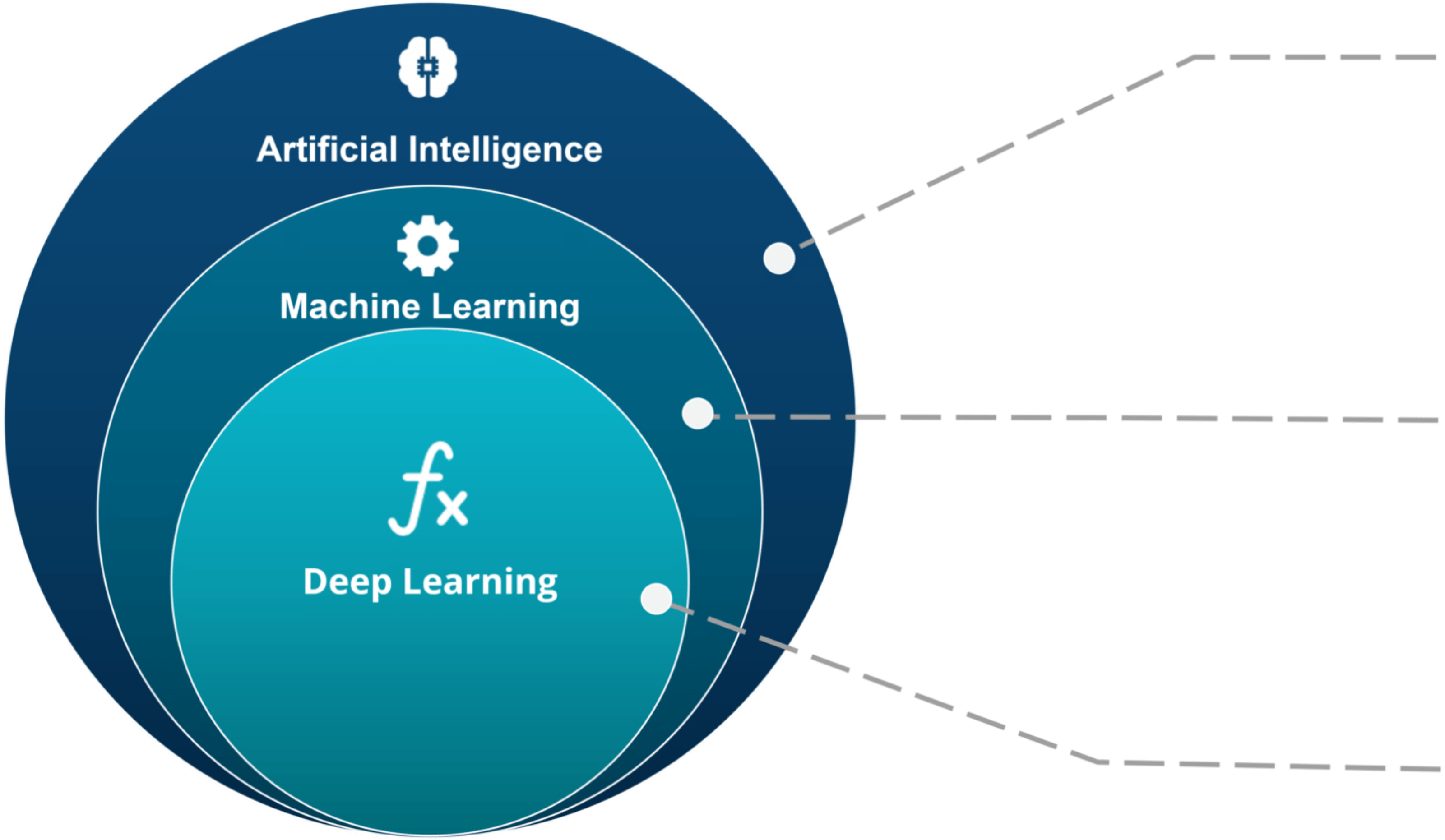
Session 1, 14 Nov 2022

Dr N V Ganapathi Raju

Professor, HOD IT,

G.R.I.E.T. , Hyderabad

AI / ML / DL

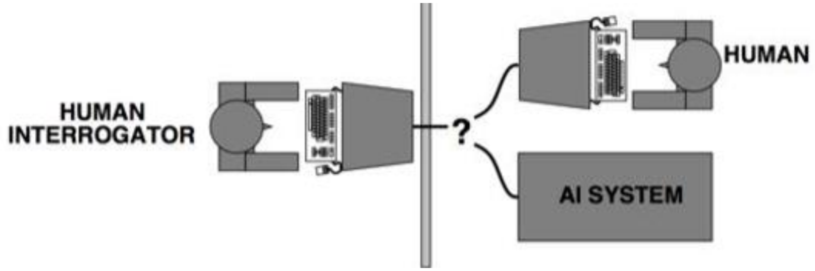


ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Turing Test approach



- A computer passes the test of intelligence, if it can fool a human interrogator.
- The computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or not.
- The computer would need to possess the following capabilities:
  - ✓ **natural language processing** to enable it to communicate successfully in English,
  - ✓ **knowledge representation** to store what it knows or hears;
  - ✓ **automated reasoning** to use the stored information to answer questions and to draw new conclusions;
  - ✓ **machine learning** to adapt to new circumstances and to detect and extrapolate patterns
  - ✓ **computer vision** to perceive objects, and
  - ✓ **robotics** to manipulate objects and move about.

Result of Turing Test

- If the interrogator can not reliably distinguish the human from the computer
- Then the computer does posses artificial intelligence

Vocabulary

- **Target:** Predicted category or value of the data (discrete / continuous)

  Column to be predicted

  Response, Output, Dependent Variable, Labels

- **Features:** Properties of the data used for prediction

  Non-Target columns

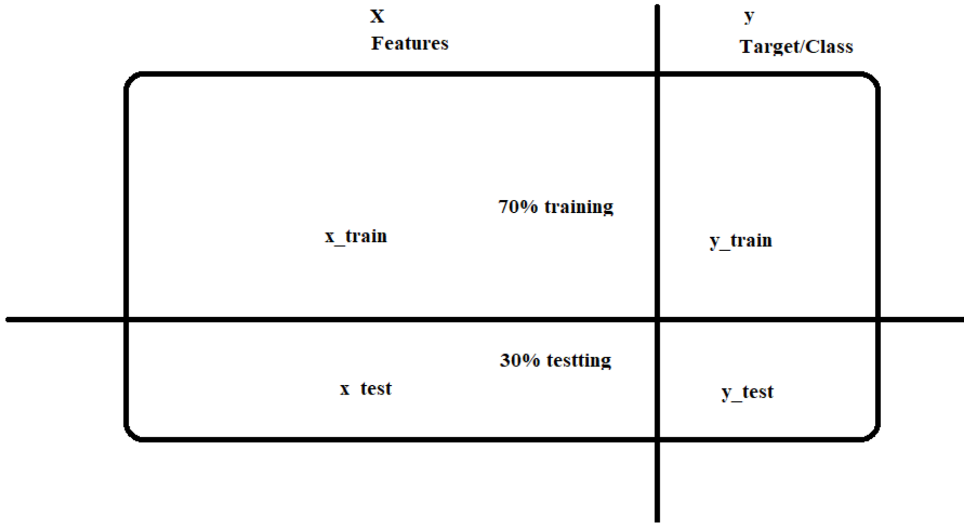
  Predictors, Input, Independent var, attributes

- **Example:** a single data point within the data (one row)

  Observations, Record, Instance, row, data points

- **Label:** The target value for a single data point

  answer, Category, Y axis



X
Features

y
Target/Class

70% training

x_train

y_train

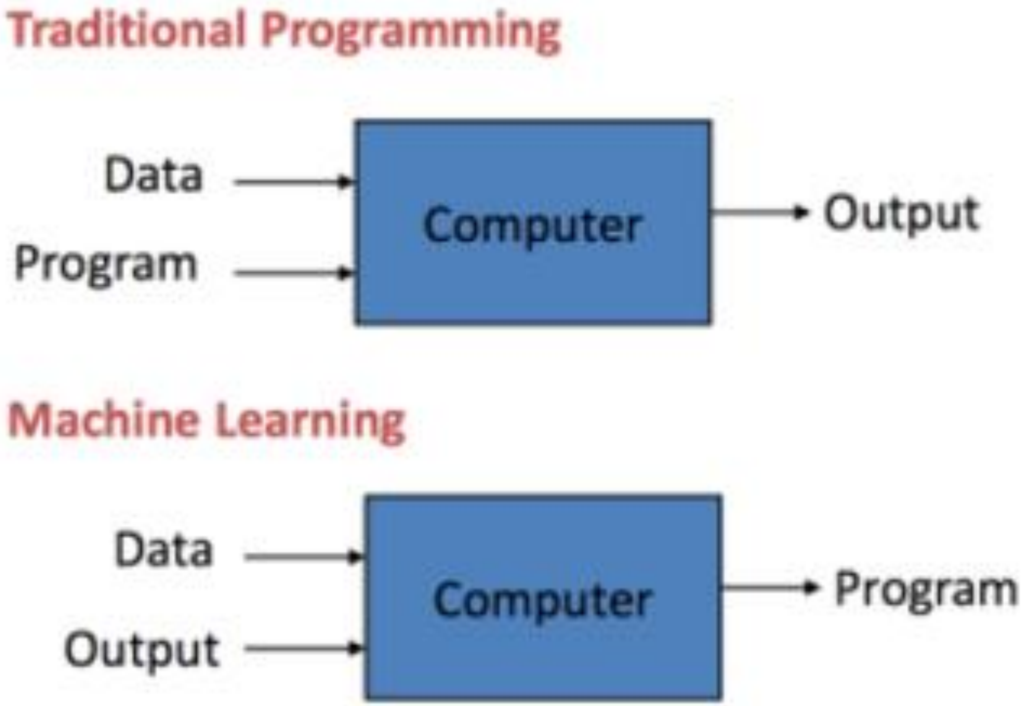
30% testting

x test

y_test

Learning

- **Traditional Programming**: Data and program is run on the computer to produce the output.
- **Machine Learning**: Data and output is run on the computer to create a program. This program can be used in traditional programming.



Traditional Programming

Data

Program

Computer

Output

Machine Learning

Data

Output

Computer

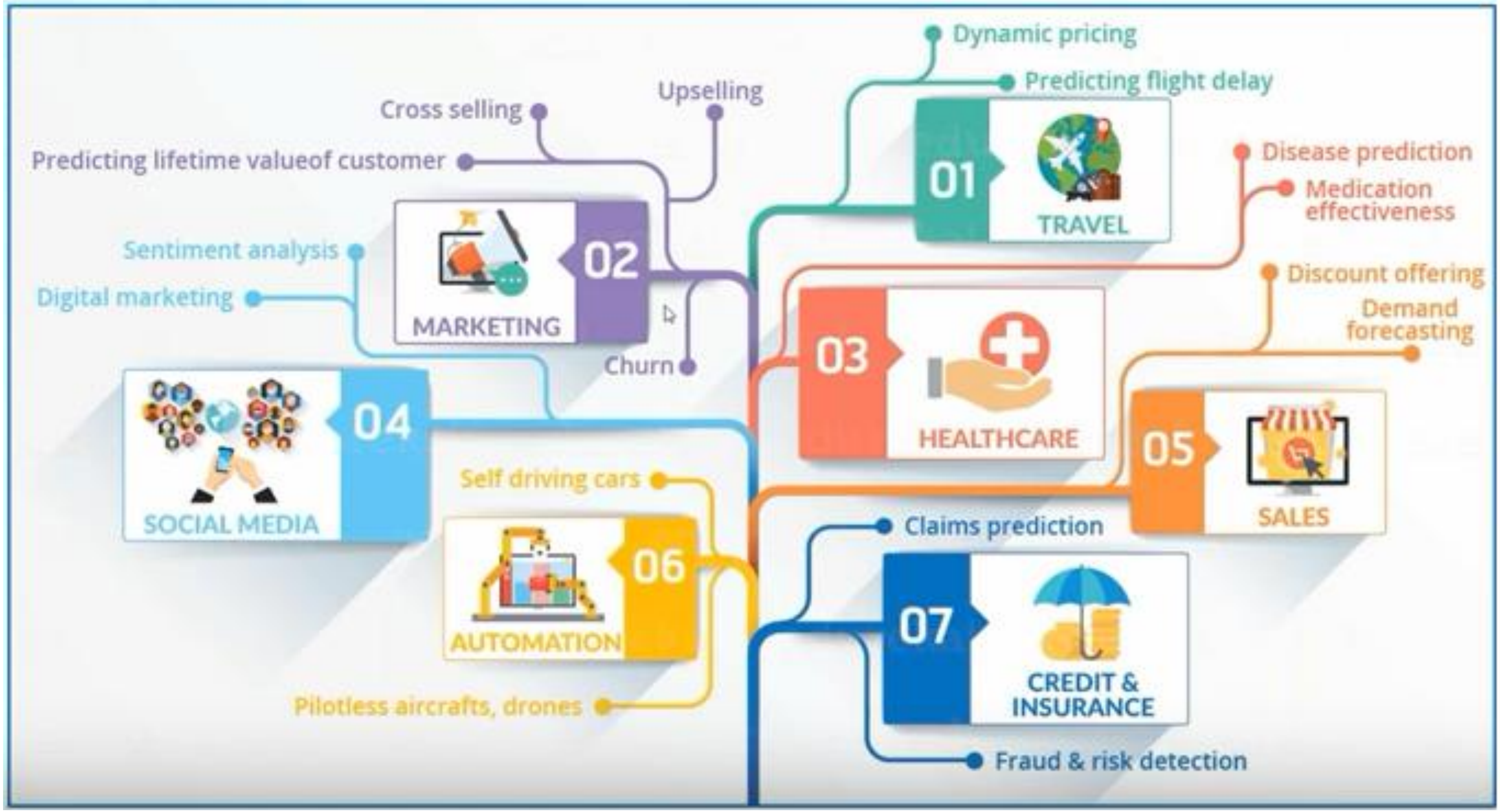
Program

- Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

https://www.ibm.com/in-en/cloud/learn/machine-learning

- Machine learning (ML) is the process of using mathematical models of data to help a computer learn without direct instruction.
- Machine learning uses algorithms to identify patterns within data, and those patterns are then used to create a data model that can make predictions.
- With increased data and experience, the results of machine learning are more accurate—much like how humans improve with more practice.

https://azure.microsoft.com/en-in/resources/cloud-computing-dictionary/what-is-machine-learning-platform/



01 TRAVEL

- Dynamic pricing
- Predicting flight delay

02 MARKETING

- Cross selling
- Upselling
- Predicting lifetime valueof customer
- Churn

03 HEALTHCARE

- Disease prediction
- Medication effectiveness

04 SOCIAL MEDIA

- Sentiment analysis
- Digital marketing

05 SALES

- Discount offering
- Demand forecasting

06 AUTOMATION

- Self driving cars
- Pilotless aircrafts, drones

07 CREDIT & INSURANCE

- Claims prediction
- Fraud & risk detection

Machine Learning

- Allows computers to learn and
- infer from data

Types of Machine Learning

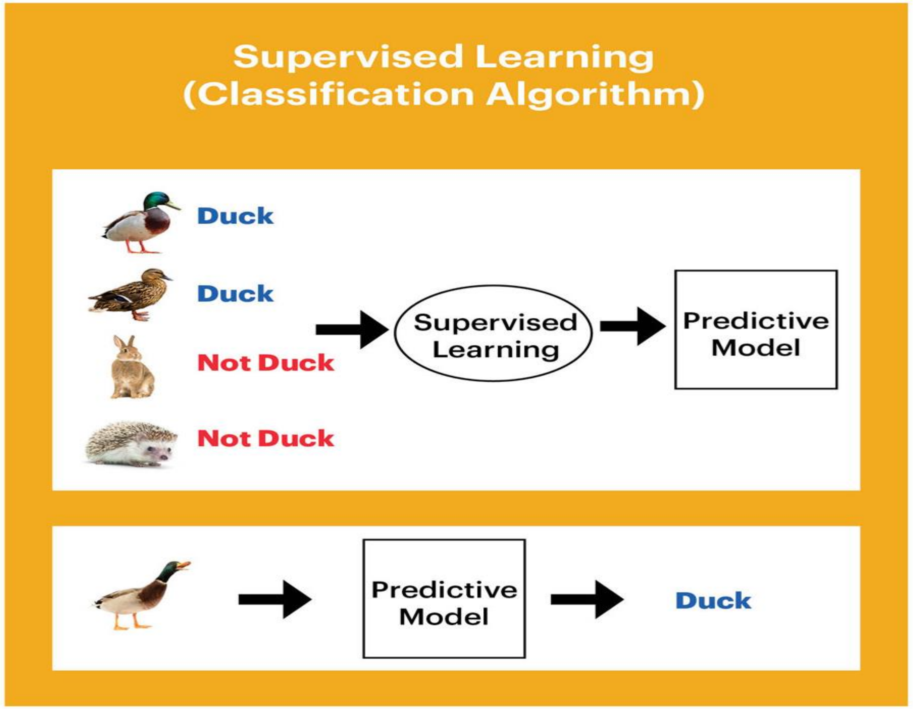
- Supervised
- Unsupervised
- Reinforcement Learning

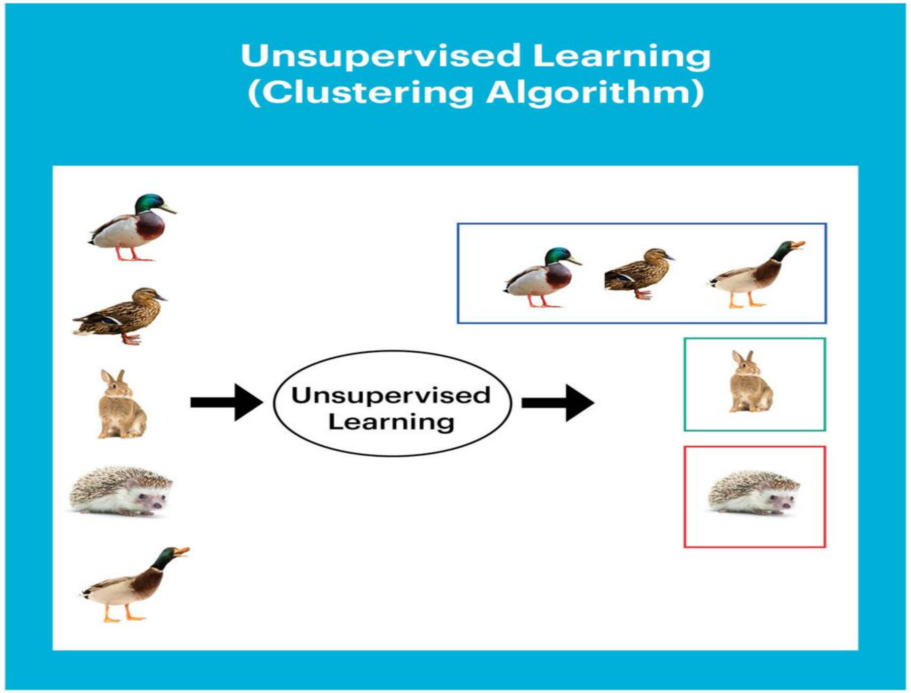
Supervised Learning

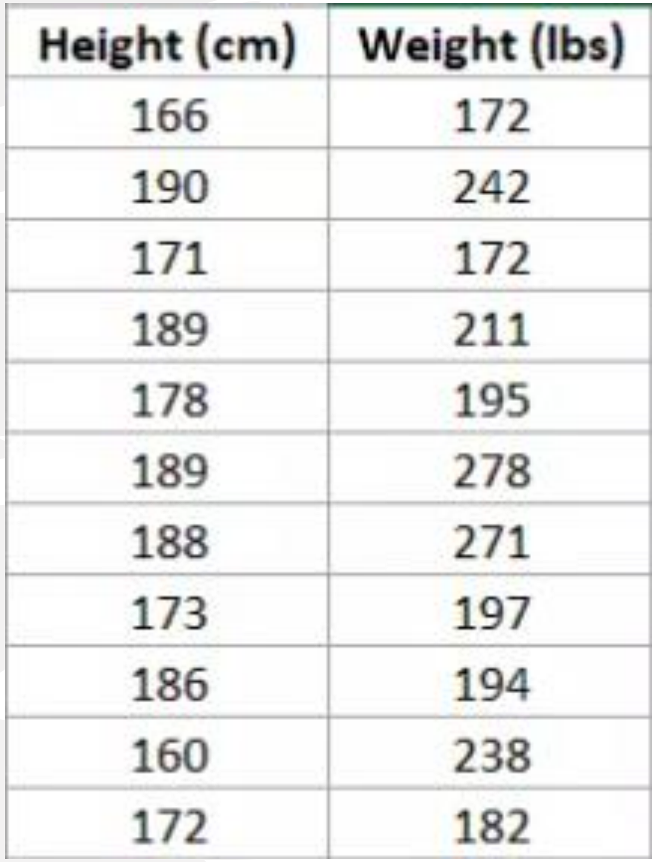
- Data points have a known outcome

Unsupervised Learning

- Data points have unknown outcome







Height (cm)	Weight (lbs)
166	172
190	242
171	172
189	211
178	195
189	278
188	271
173	197
186	194
160	238
172	182

Height (cms)	Female (kg)	Male (kg)
152 (5′ 0″)	40.8/49.9 kg	43.1/53 kg
155	43.1/52.6	45.8/55.8
(157 cm)	(44.9/54.9 kg)	(48.1/58.9 kg)
(160 cm)	(47.2/57.6 kg)	(50.8/61.6 kg)
(163 cm)	(49/59.9 kg)	(53/64.8 kg)
(165 cm)	(51.2/62.6 kg)	(55.3/68 kg)
(168 cm)	(53/64.8 kg)	(58/70.7 kg)
(170 cm)	(55.3/67.6 kg)	(60.3/73.9 kg)
(173 cm)	(57.1/69.8 kg)	(63/76.6 kg)
(175 cm)	(59.4/72.6 kg)	(65.3/79.8 kg)
(178 cm)	(61.2/74.8 kg)	(67.6/83 kg)
(180 cm)	(63.5/77.5 kg)	(70.3/85.7 kg)
183	65.3/79.8	72.6/88.9

Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1

Height	Weight	T Shirt Size
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L

Types of Supervised Learning

Regression

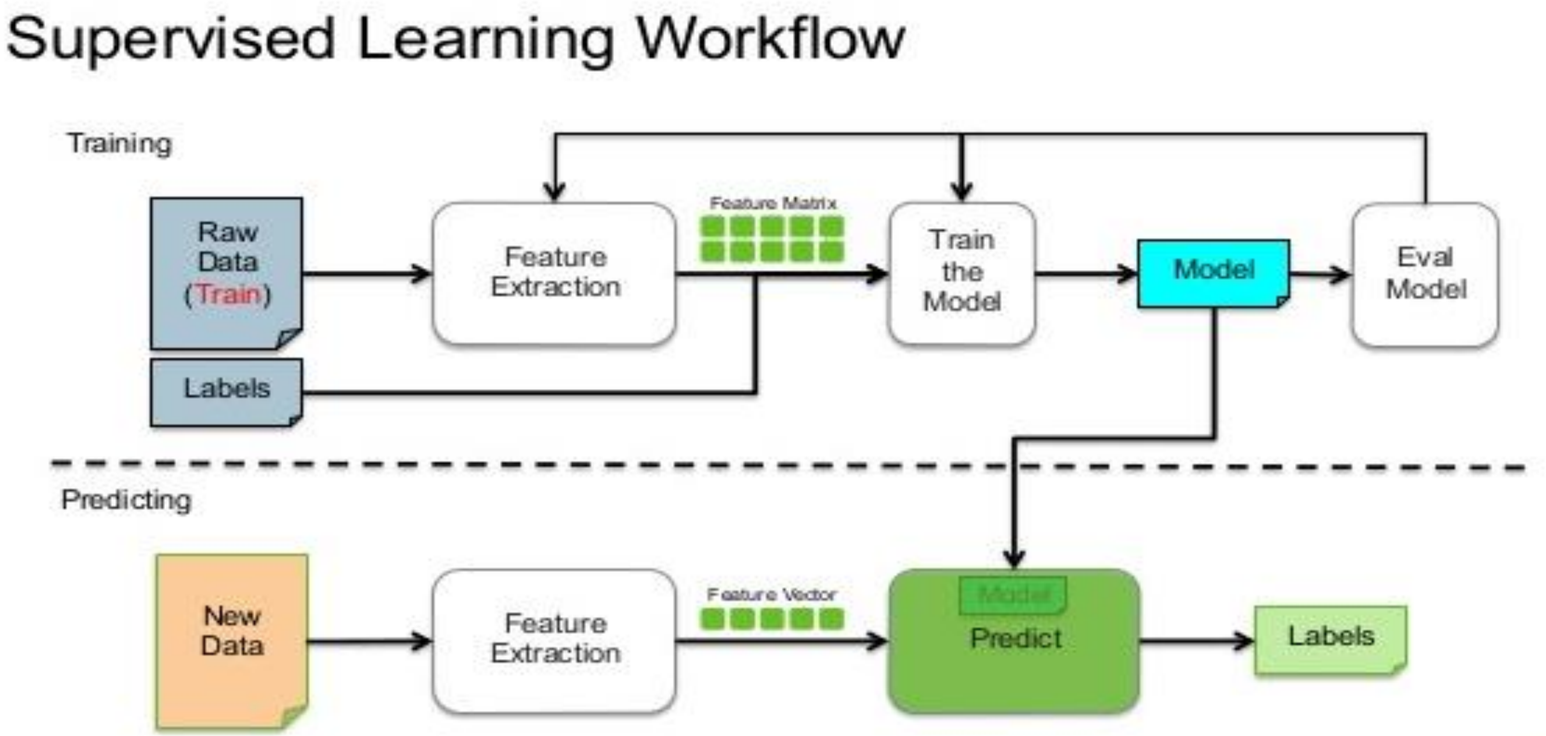
Outcome is continuous (numerical)

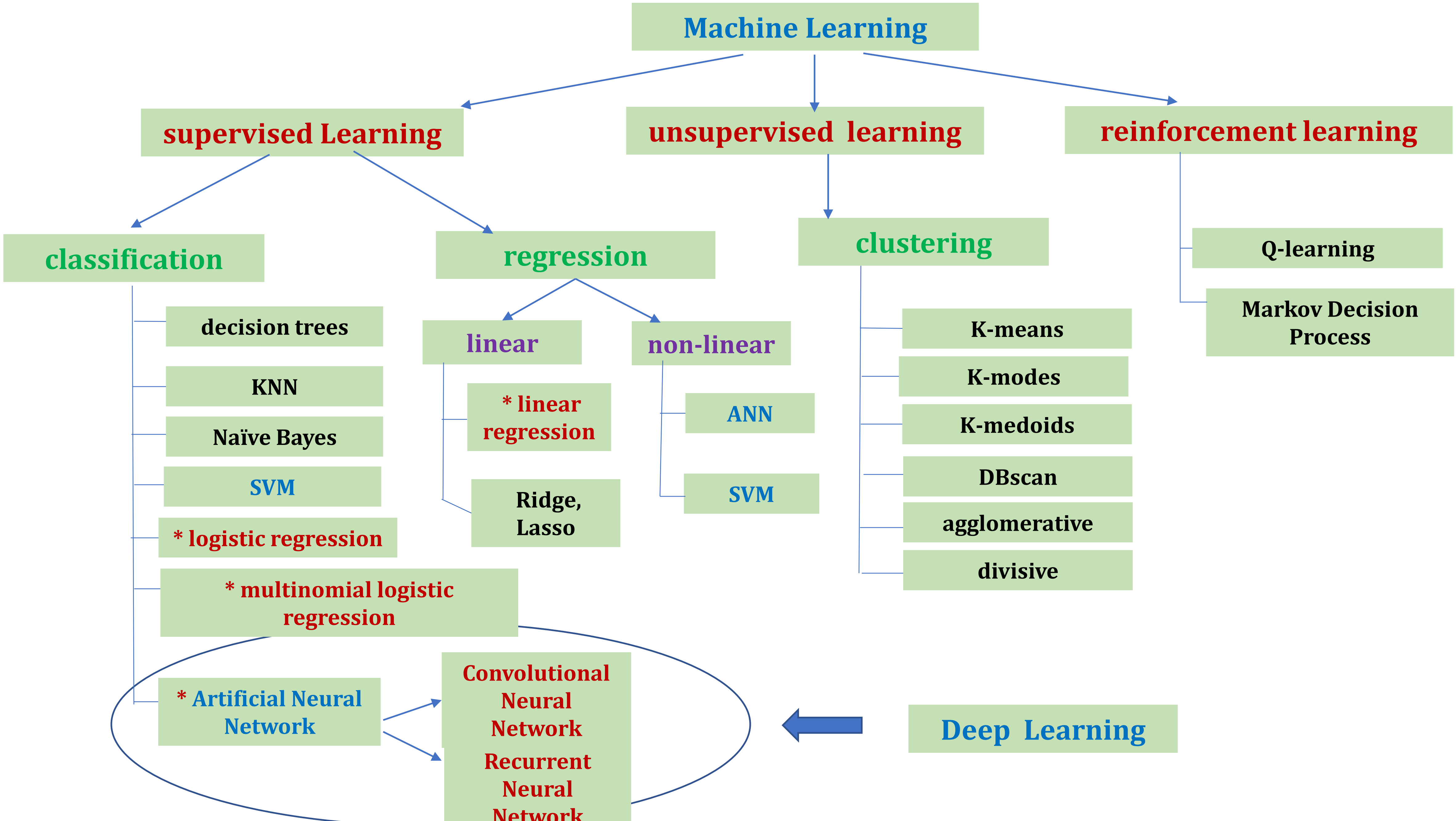
Ex:- home prices, happiness index

Classification

Outcome is a Category

Ex:- Object classes in Images





Machine Learning

supervised Learning

unsupervised learning

reinforcement learning

classification

decision trees

KNN

Naïve Bayes

SVM

* logistic regression

* multinomial logistic regression

* Artificial Neural Network

Convolutional Neural Network

Recurrent Neural Network

regression

linear

* linear regression

Ridge, Lasso

non-linear

ANN

SVM

clustering

K-means

K-modes

K-medoids

DBscan

agglomerative

divisive

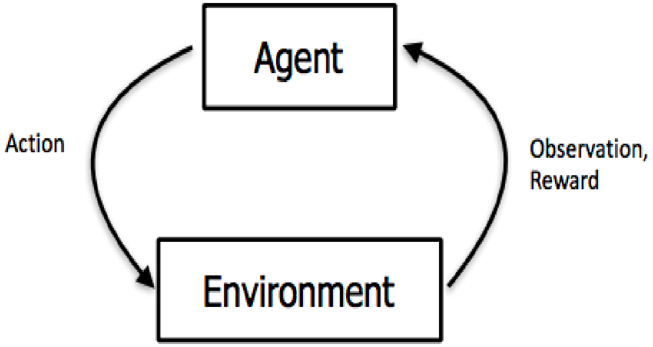
Q-learning

Markov Decision Process

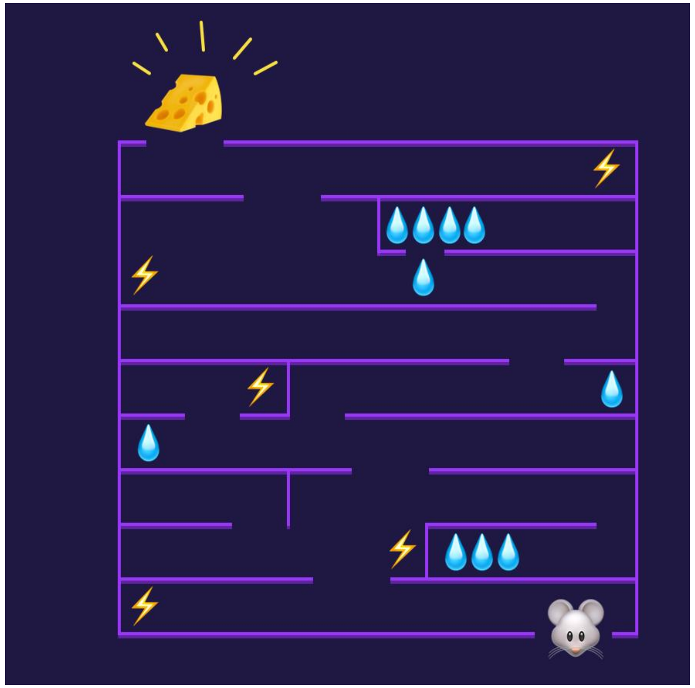
Deep Learning

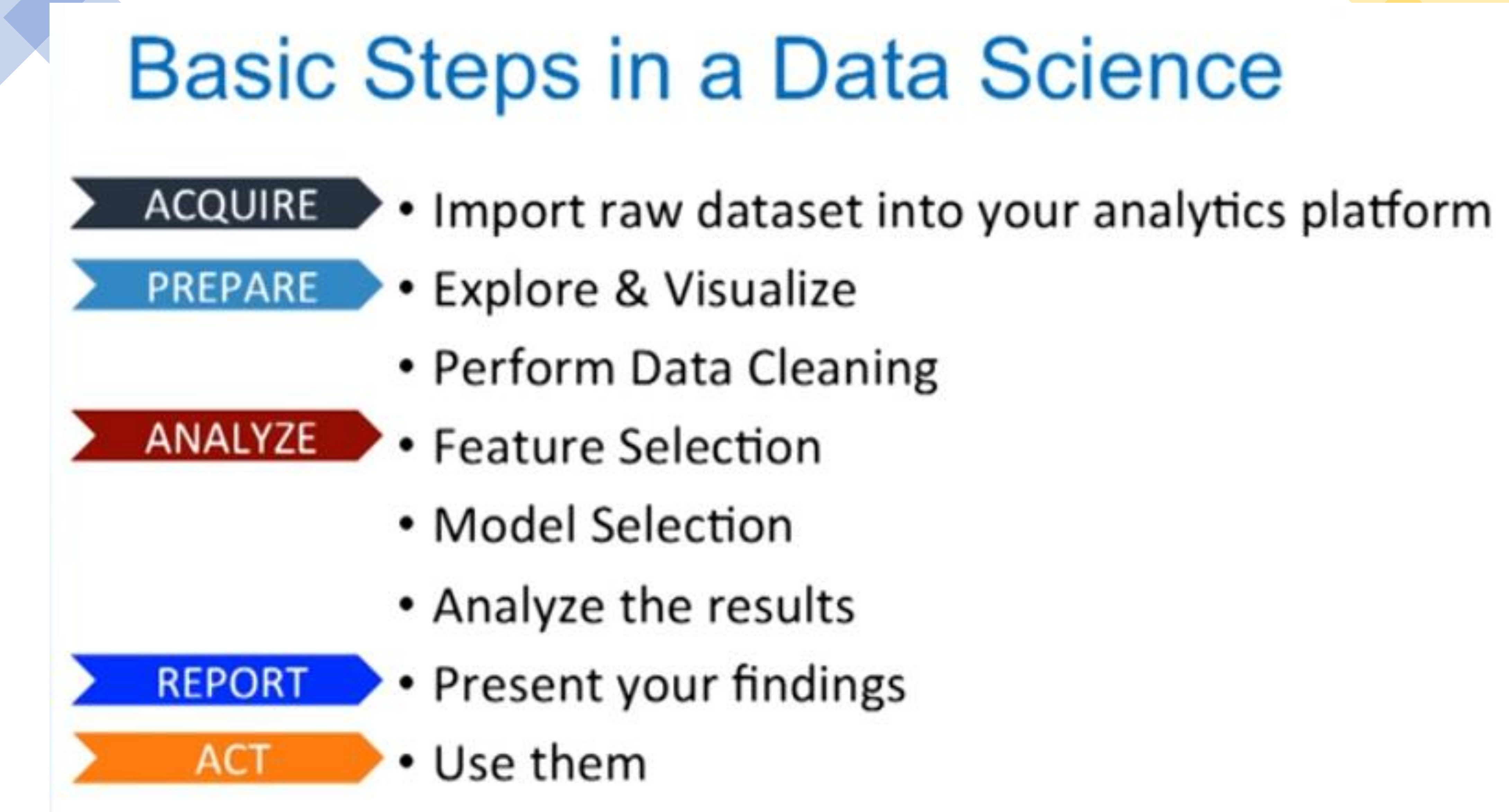
Reinforcement Learning

- In supervised learning, training data comes with an answer key from some godlike “supervisor
- In **reinforcement learning (RL)** there’s no answer key, but your reinforcement learning **agent** still must decide how to act to perform its task.
- In the absence of existing training data, the agent learns from experience.
- It collects the training examples (“this action was good, that action was bad”) through **trial-and-error** as it attempts its task, with the goal of maximizing long-term **reward**.



The agent **observes** the environment, takes an **action** to interact with the environment, and receives positive or negative **reward.**





Basic Steps in a Data Science

ACQUIRE
- Import raw dataset into your analytics platform

PREPARE
- Explore & Visualize
- Perform Data Cleaning

ANALYZE
- Feature Selection
- Model Selection
- Analyze the results

REPORT
- Present your findings

ACT
- Use them

Steps in Data Science

- Problem Identification
- Data Collection/Generation
- Data Preprocessing
- Data Exploration (EDA)
- Feature Selection
- Model Building
- Model Evaluation
- Analyze Results

Data Wrangling / Munging

- Data Imputation
- Data Integration
- Data Encoding / Decoding
- Data Transformation / Normalization
- Dimensionality Reduction
- Feature Engineering

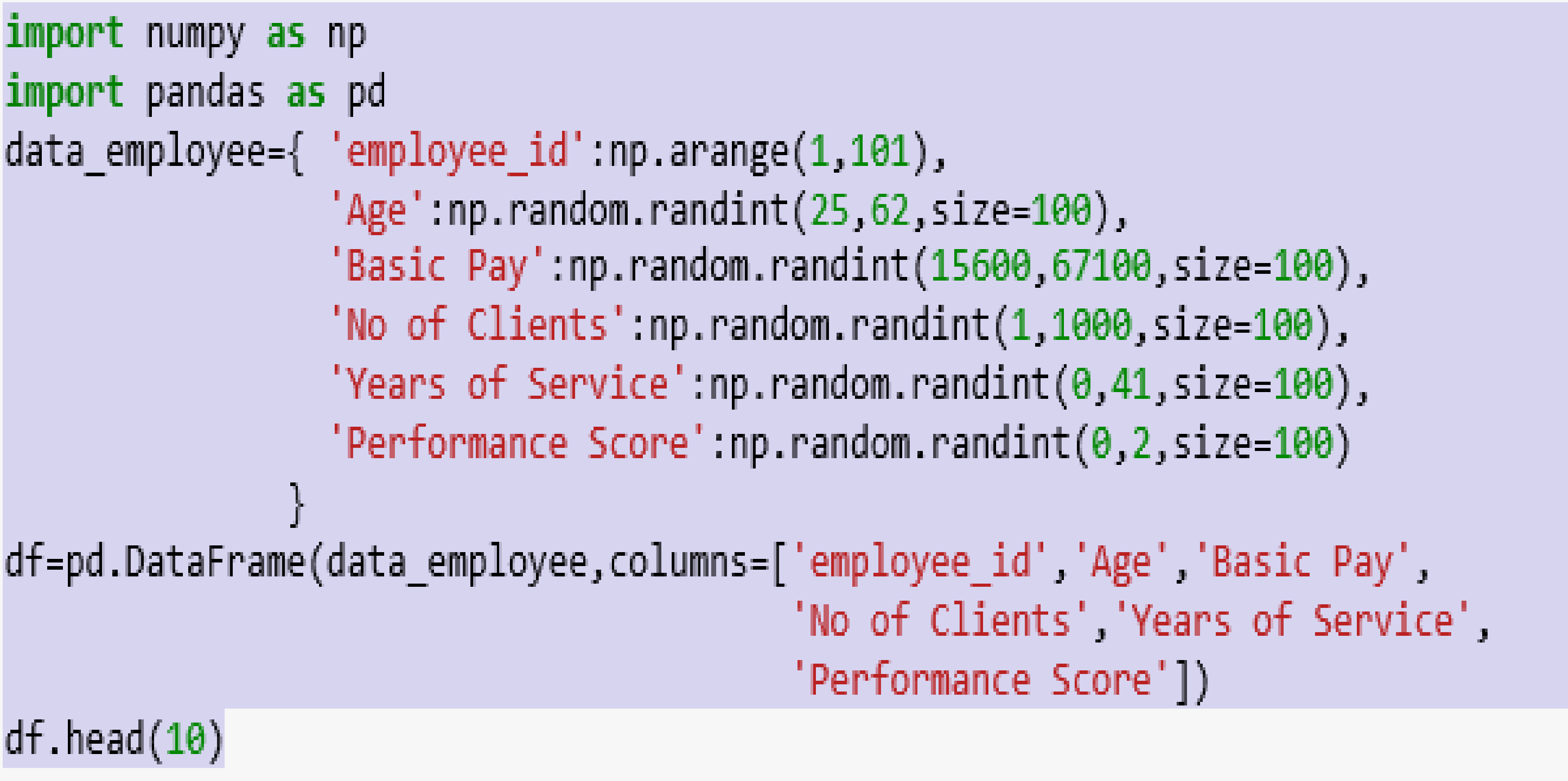
Various Sources of data for Research

- https://www.kaggle.com/competitions
- https://data.gov.in/ https://www.data.gov/ (Government Datasets)
- https://www.kdnuggets.com/
- https://archive.ics.uci.edu/ml/index.php (UCI Machine Learning datasets)
- https://www.ncbi.nlm.nih.gov/ (biomedical research)
- https://registry.opendata.aws/usage-examples/ (Amazon Datasets)
- https://datasetsearch.research.google.com/ (Google’s Datasets Search Engine)
- https://msropendata.com/ (Microsoft Research Open Data)
- https://github.com/awesomedata/awesome-public-datasets (Awesome Public Datasets Collection)
- Generate our own data depending on Problem

Data Generation

Range of this attributes are as follows:

- Employee_id : 1-100
- Age : 25-62
- Basic pay : 15,600-67000
- No.of clients :1-1000
- Years of Services :0-40
- Performance Score:0/1



```
import numpy as np
import pandas as pd
data_employee={ 'employee_id':np.arange(1,101),
                'Age':np.random.randint(25,62,size=100),
                'Basic Pay':np.random.randint(15600,67100,size=100),
                'No of Clients':np.random.randint(1,1000,size=100),
                'Years of Service':np.random.randint(0,41,size=100),
                'Performance Score':np.random.randint(0,2,size=100)
              }
df=pd.DataFrame(data_employee,columns=['employee_id','Age','Basic Pay',
                                       'No of Clients','Years of Service',
                                       'Performance Score'])
df.head(10)
```

Data Imputation

- Many real-world datasets may contain missing values for various reasons. They are often encoded as NaNs, blanks or any other placeholders.

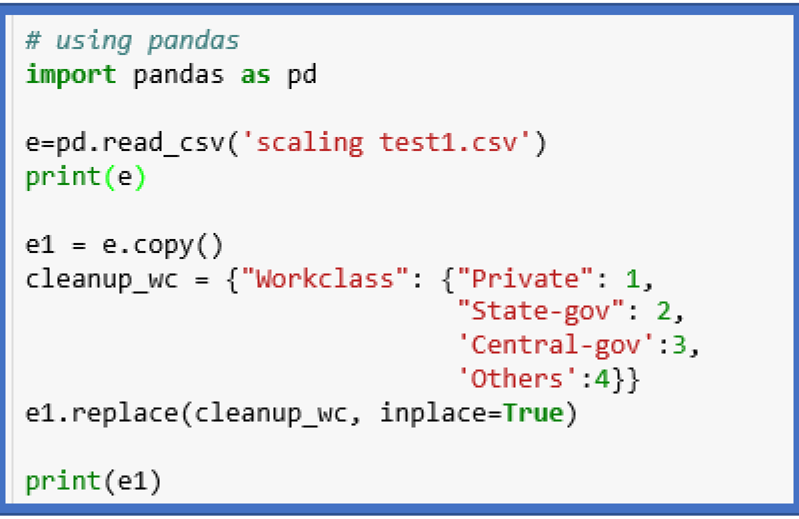
- Rule 1: Discrete/Continuous values will be imputed with mean/median/standard deviation

- Rule 2: Categorical values will be imputed with mode.

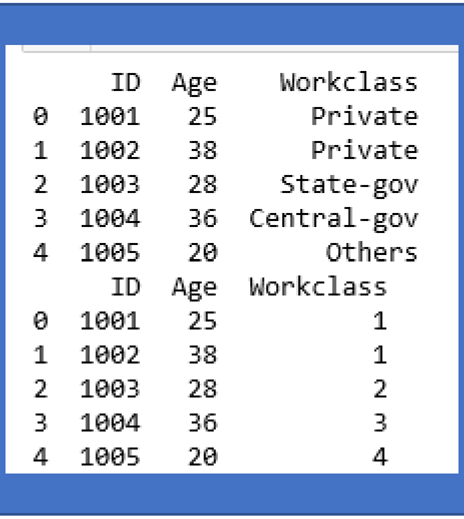
```
e1['Age'].fillna(e['Age'].mean(),inplace=True)
e1['Basic Pay'].fillna(e['Basic Pay'].mean(),inplace=True)
e1['No of Clients'].fillna(e['No of Clients'].mean(),inplace=True)
e1['Years of Service'].fillna(e['Years of Service'].mean(),inplace=True)
print(e1.head(5))
```

Data Encoding

- Most of Machine Learning libraries represent data in numerical values rather than categorical values.
- import “LabelEncoder” class from “sklearn.preprocessing” library and create an object labelencoder_X of the LabelEncoder class. After that use the fit_transform method on the categorical features.
- One hot encoding transforms categorical features to a format that works better with classification and regression algorithms.

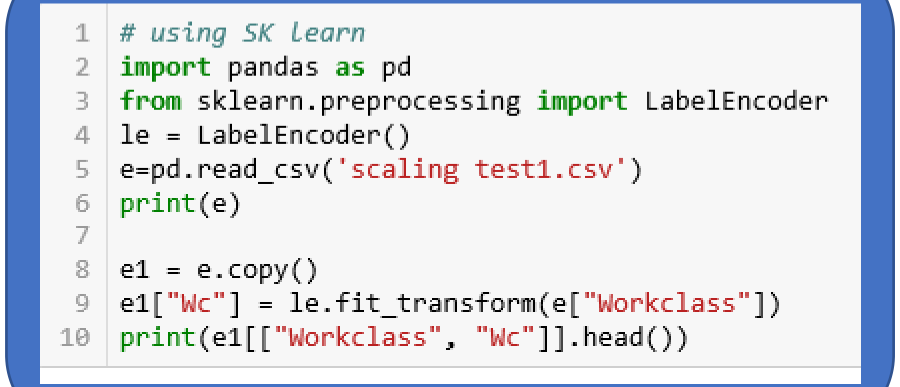


```python
# using pandas
import pandas as pd

e=pd.read_csv('scaling test1.csv')
print(e)

e1 = e.copy()
cleanup_wc = {"Workclass": {"Private": 1,
                            "State-gov": 2,
                            'Central-gov':3,
                            'Others':4}}
e1.replace(cleanup_wc, inplace=True)

print(e1)
```

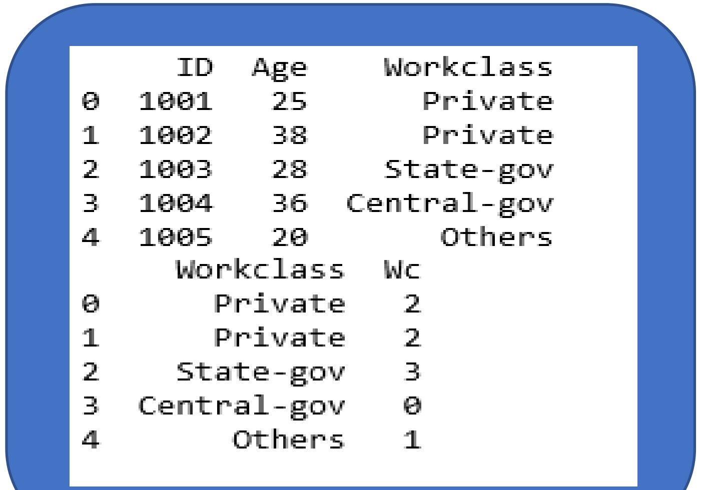


```
     ID  Age    Workclass
0  1001   25      Private
1  1002   38      Private
2  1003   28    State-gov
3  1004   36  Central-gov
4  1005   20       Others
     ID  Age  Workclass
0  1001   25          1
1  1002   38          1
2  1003   28          2
3  1004   36          3
4  1005   20          4
```

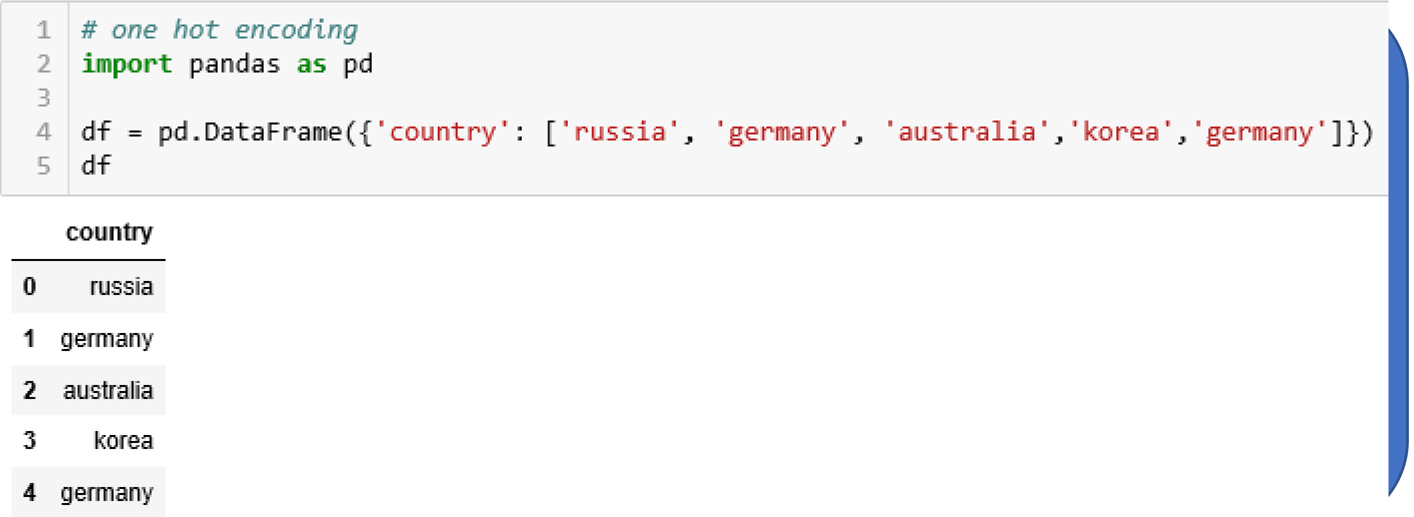
Data Encoding



```
# using SK Learn
import pandas as pd
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
e=pd.read_csv('scaling test1.csv')
print(e)

e1 = e.copy()
e1["Wc"] = le.fit_transform(e["Workclass"])
print(e1[["Workclass", "Wc"]].head())
```

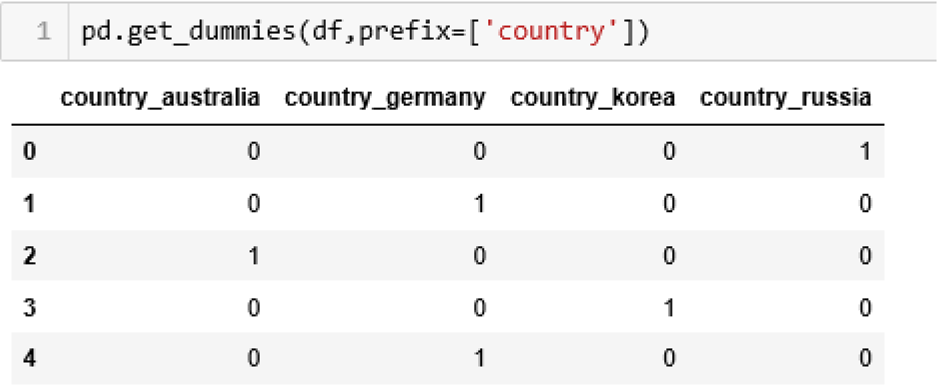


```
     ID  Age     Workclass
0  1001   25       Private
1  1002   38       Private
2  1003   28     State-gov
3  1004   36   Central-gov
4  1005   20        Others
     Workclass  Wc
0      Private   2
1      Private   2
2    State-gov   3
3  Central-gov   0
4       Others   1
```



```
# one hot encoding
import pandas as pd

df = pd.DataFrame({'country': ['russia', 'germany', 'australia','korea','germany']})
df
```

	country
0	russia
1	germany
2	australia
3	korea
4	germany

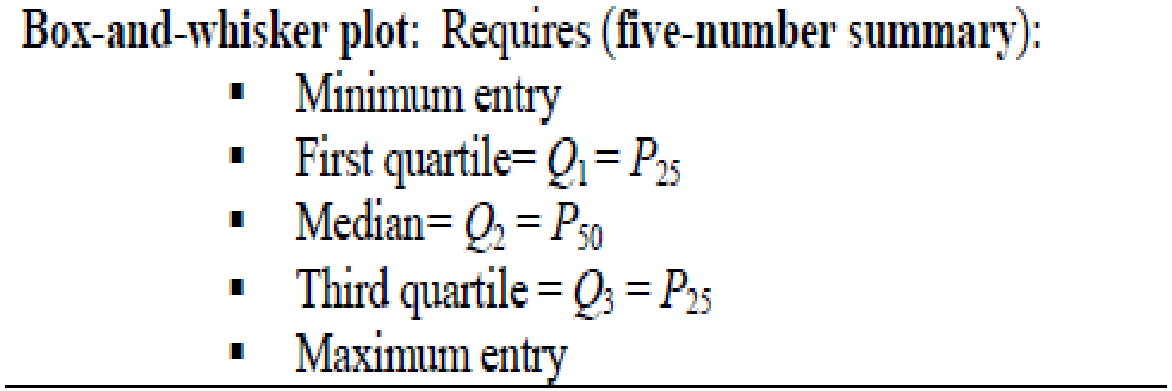


```
pd.get_dummies(df,prefix=['country'])
```

	country_australia	country_germany	country_korea	country_russia
0	0	0	0	1
1	0	1	0	0
2	1	0	0	0
3	0	0	1	0
4	0	1	0	0

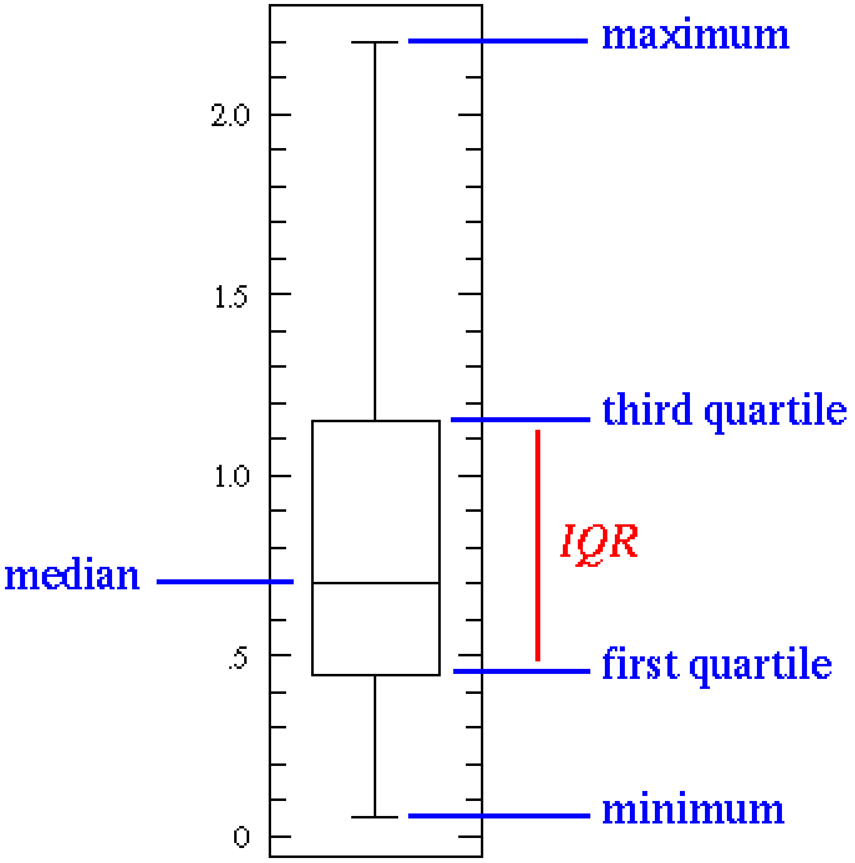
Outlier Detection

- **Outliers** are not just greatest and least values, but values that are very different from the pattern established by the rest of the data. Outliers affect the mean.



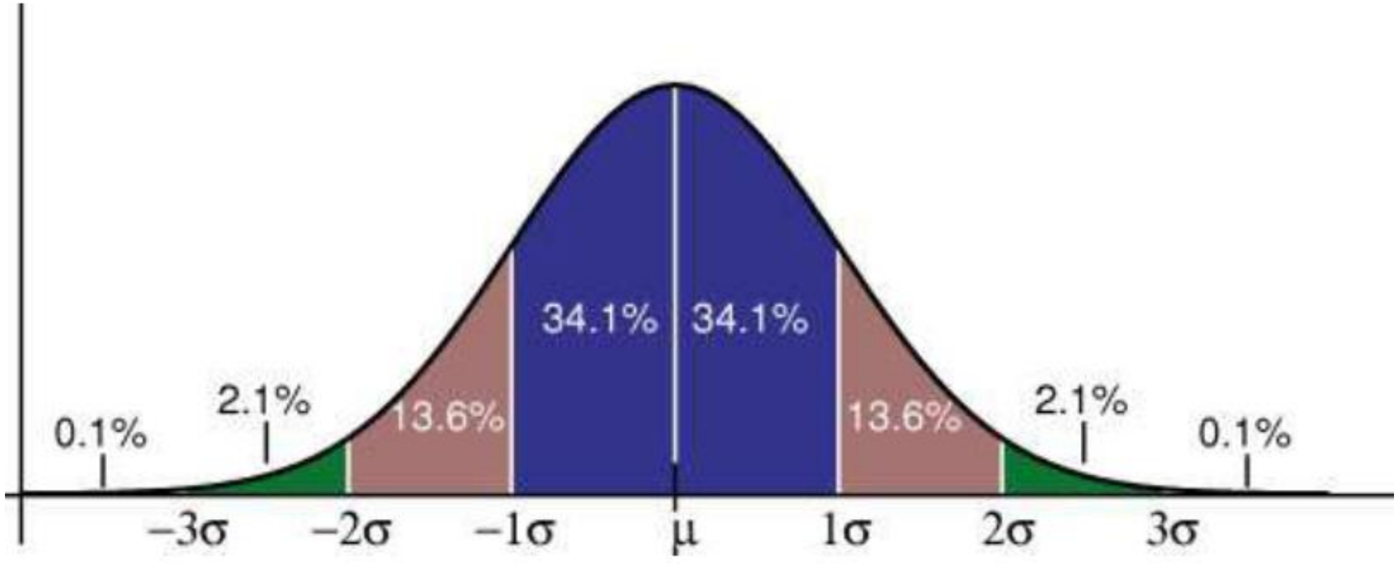
Box-and-whisker plot: Requires (**five-number summary**):

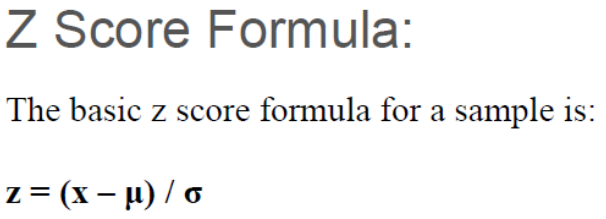
- Minimum entry
- First quartile= $Q_1 = P_{25}$
- Median= $Q_2 = P_{50}$
- Third quartile = $Q_3 = P_{25}$
- Maximum entry



Data Normalization

- A **normal distribution**, sometimes called the bell curve, is a distribution that occurs naturally in many situations.
- The empirical rule tells you what percentage of your data falls within a certain number of standard deviations from the mean:
- 68% of the data falls within one standard deviation of the mean.
- 95% of the data falls within two standard deviations of the mean.
- 99.7% of the data falls within three standard deviations of the mean.

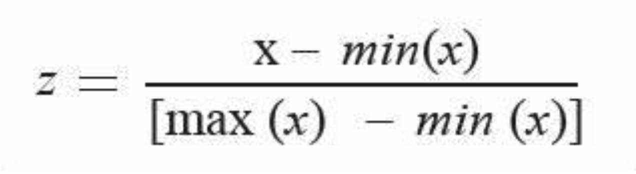




Z Score Formula:

The basic z score formula for a sample is:

z = (x – μ) / σ

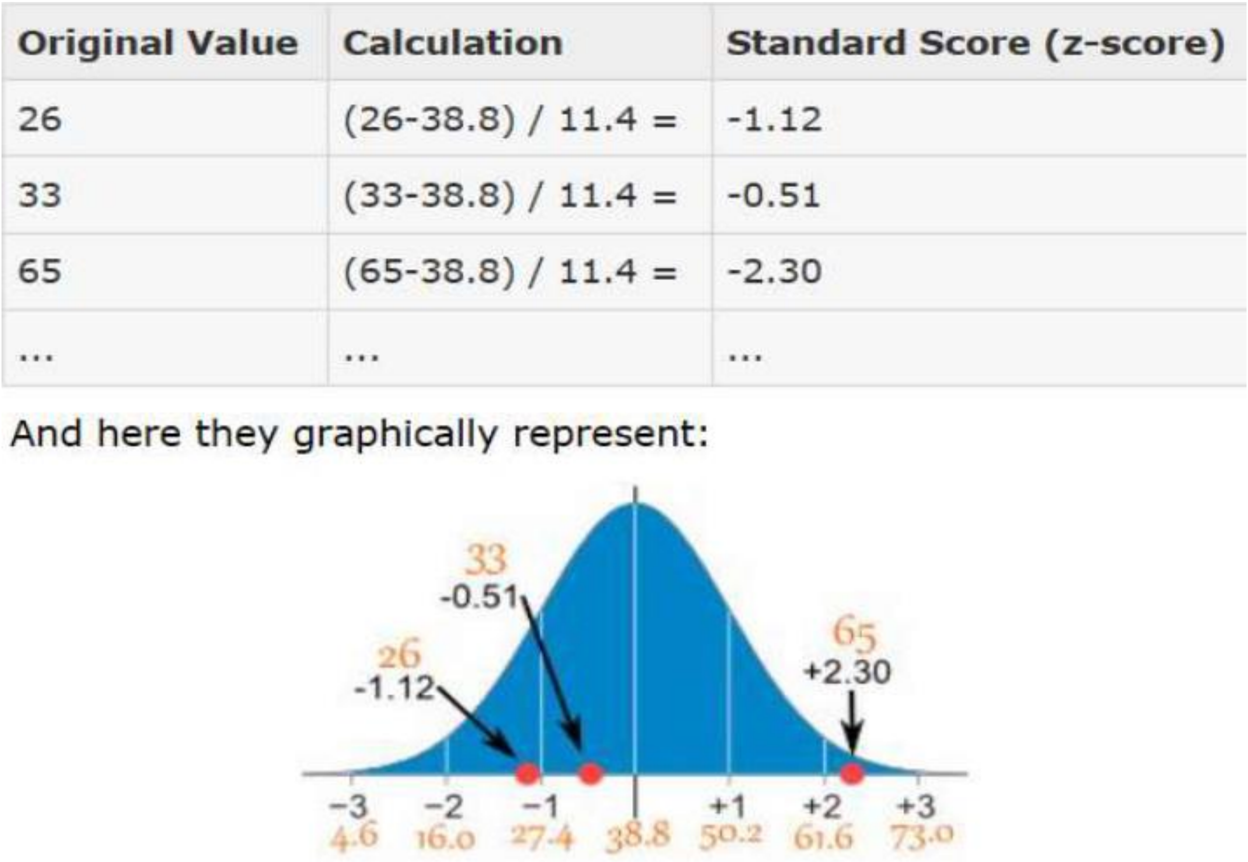


$$z = \frac{x - min(x)}{[\max(x) - min(x)]}$$

A survey of daily travel time had these results (in minutes):

26	33	65	28	34	55	25	44	50	36	26	37	43	62	35	38	45	32	28	34

The Mean is 38.8 minutes, and the Standard Deviation is 11.4 minutes. Convert the values to z - scores and prepare the Normal Distribution Graph.

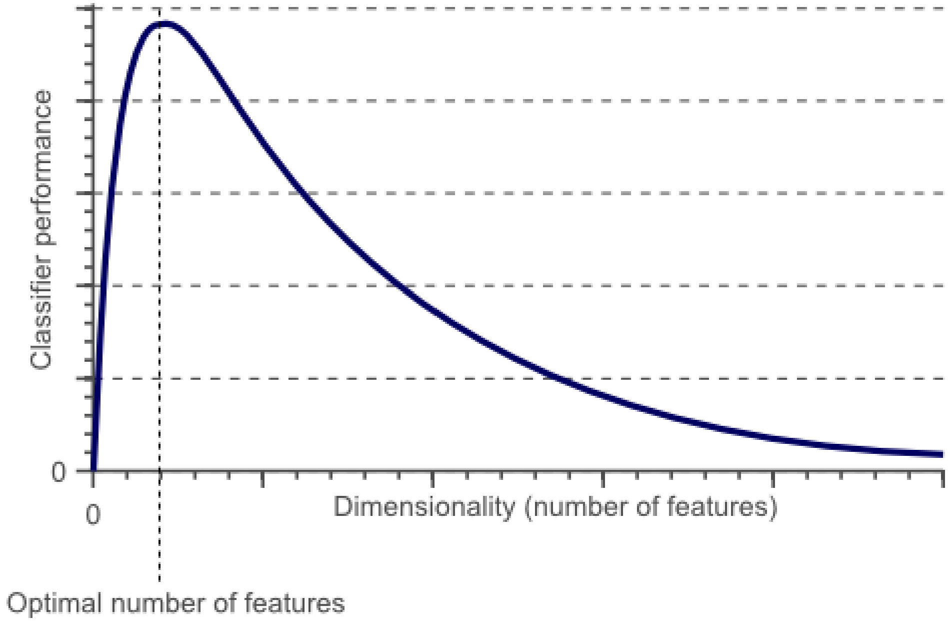


Original Value	Calculation	Standard Score (z-score)
26	(26-38.8) / 11.4 =	-1.12
33	(33-38.8) / 11.4 =	-0.51
65	(65-38.8) / 11.4 =	-2.30
...	...	...

And here they graphically represent:

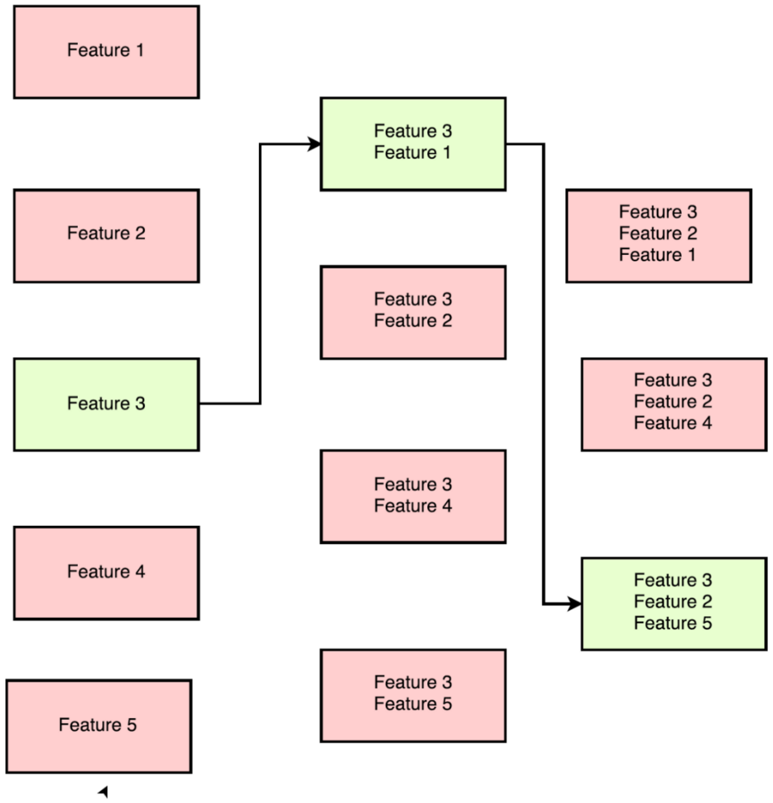
Dimensionality Reduction

- As the number of features increases, the model becomes more complex.
- The more the number of features, the more the chances of overfitting.
- **PCA/LDA/SVD** are examples
- A machine learning model that is trained on many features, gets increasingly dependent on the data it was trained on and in turn overfitted, resulting in poor performance on real data, beating the purpose.



Feature Selection and Engineering

- Feature selection is the process of identifying and selecting relevant features for your sample.
- Feature engineering is manually generating new features from existing features, by applying some transformation or performing some operation on them.



Feature 1

Feature 2

Feature 3

Feature 4

Feature 5

Feature 3
Feature 1

Feature 3
Feature 2

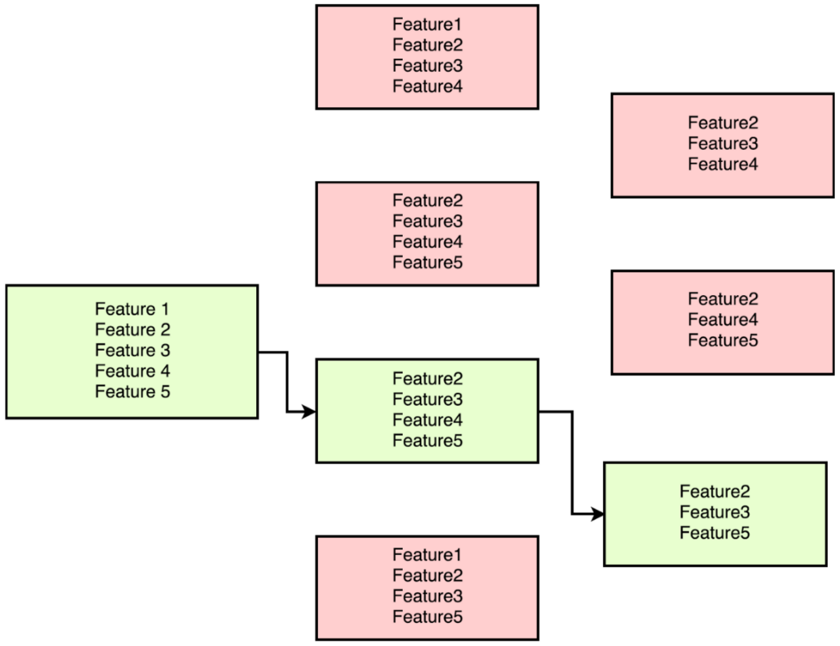
Feature 3
Feature 4

Feature 3
Feature 5

Feature 3
Feature 2
Feature 1

Feature 3
Feature 2
Feature 4

Feature 3
Feature 2
Feature 5



Feature 1
Feature 2
Feature 3
Feature 4
Feature 5

Feature1
Feature2
Feature3
Feature4

Feature2
Feature3
Feature4
Feature5

Feature2
Feature3
Feature4
Feature5

Feature1
Feature2
Feature3
Feature5

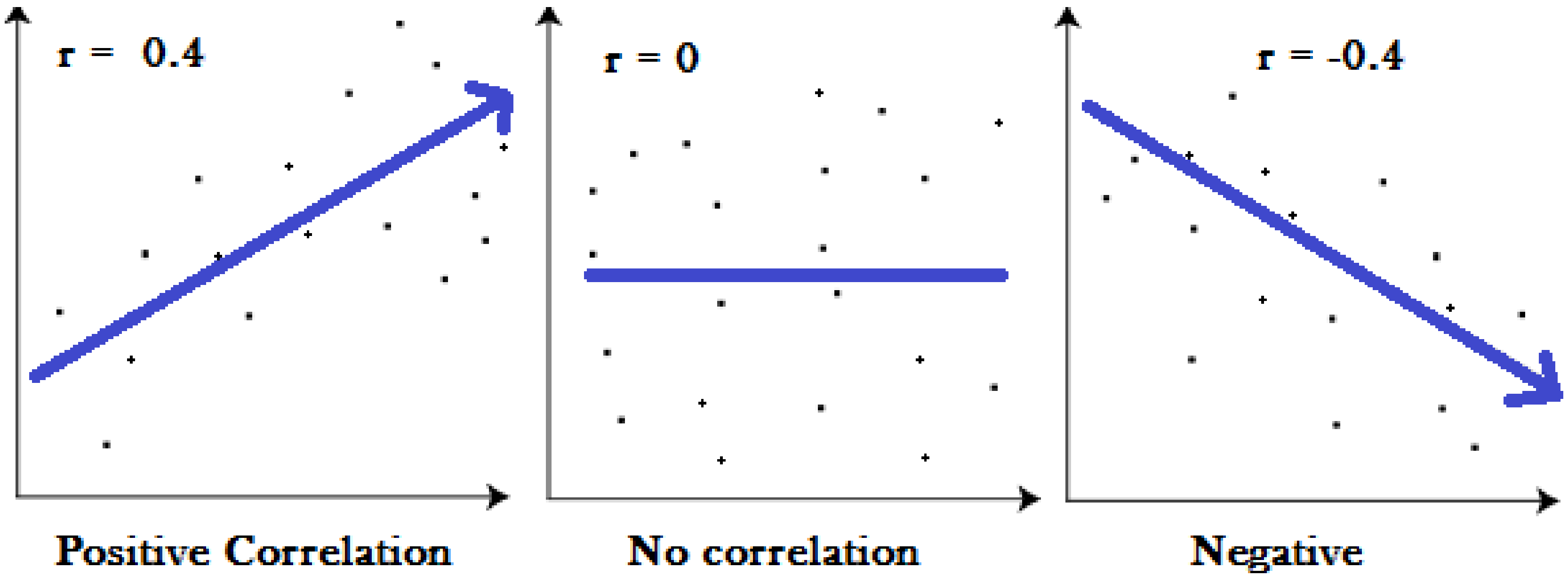
Feature2
Feature3
Feature4

Feature2
Feature4
Feature5

Feature2
Feature3
Feature5

Covariance and correlation

- **Covariance** provides insight into how two variables are related to one another.
- **Correlation** tells that at what degree two or more variables are related.


r = 0.4
Positive Correlation
r = 0
No correlation
r = -0.4
Negative