ACDS Lecture Series

Lecture - 3

Machine Learning

G. N. Sastry and Team

ADVANCED COMPUTATION AND DATA SCIENCES (ACDS) DIVISION

CSIR-North East Institute of Science and Technology, Jorhat, Assam, India

Learning, Understanding & Knowledge

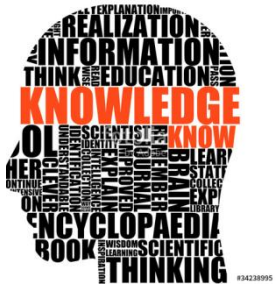
Acquiring new or existing knowledge. Observe a phenomena and recognize a pattern.

- **Unconscious Incompetence**
- **Conscious Incompetence**
- **Conscious Competence**
- **Unconscious Competence**

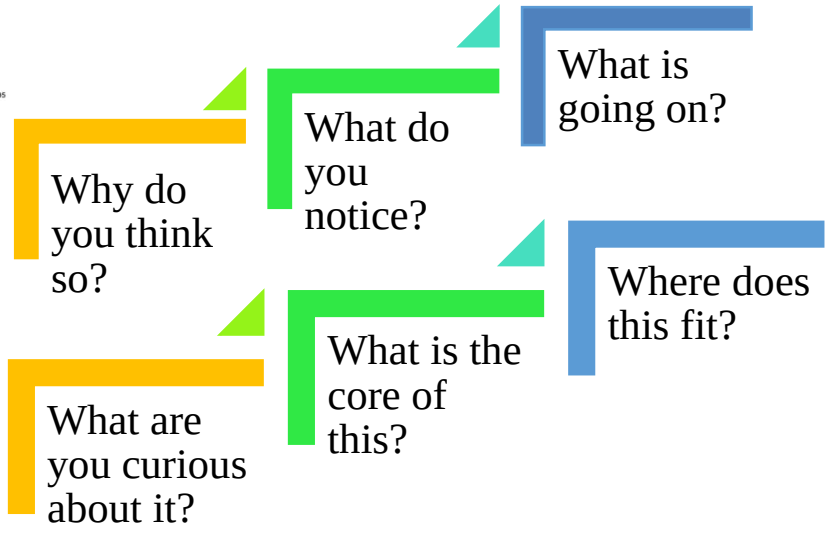
UNDERSTANDING

Ability to perform flexibility with what is taught an applying it beyond the knowledge acquired

Knowledge	Understanding	Wisdom
Perception	Comprehension	Why it exists
Identifying things and situation	Comprehending how things ae caused	Making decisions
Perception of boundaries	Perception of casual relationship	Perception of purpose



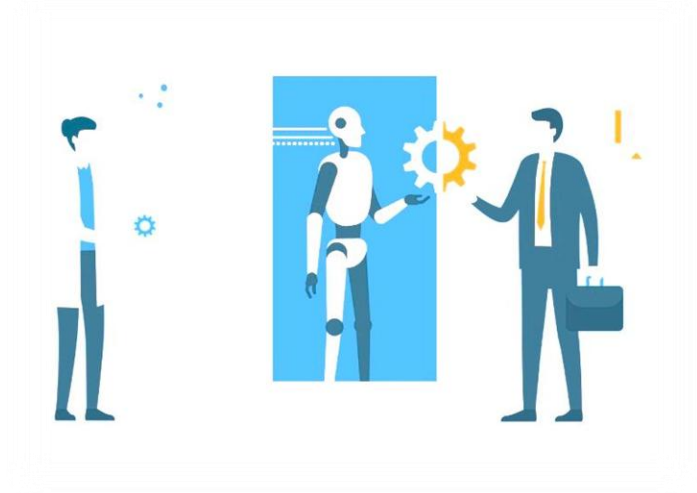
Understanding gained from learning and experiences

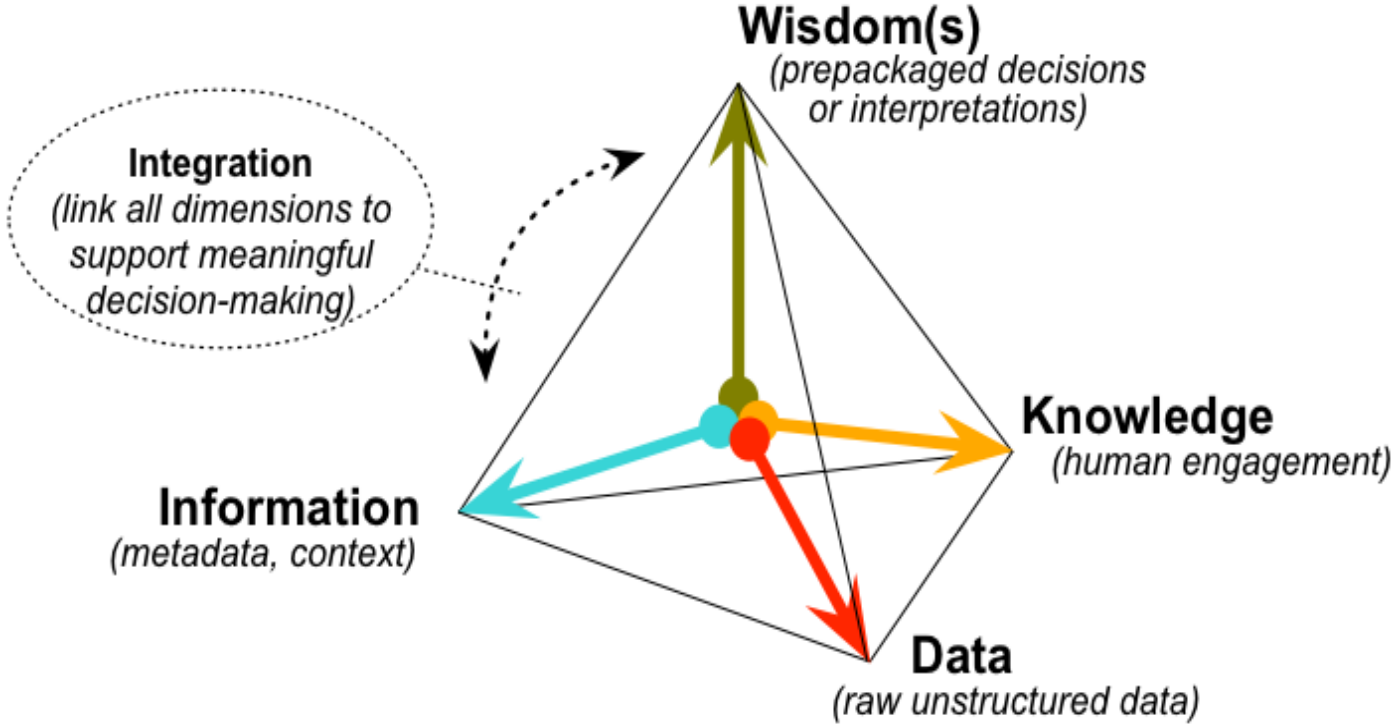


Can a Machine to Mimic the Process of Human Learning?

Whether its Human Learning or Machine Learning — both involve observations and executing about a thing or a process or phenomenon.

MACHINE LEARNING



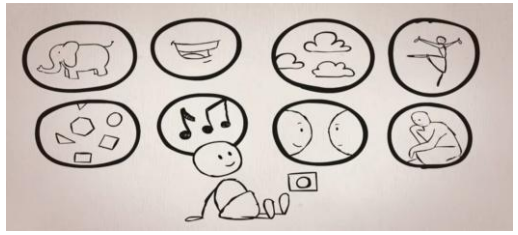


Nevertheless, it is humans who build, operate, and understand the tools that turn data into information, and information into knowledge, and knowledge into wisdom.

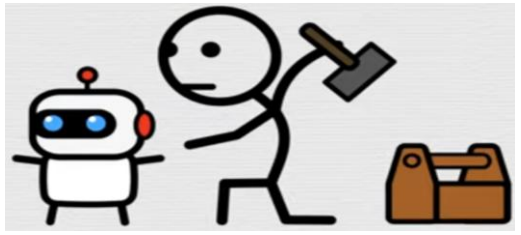
https://www.oreilly.com/content/the-wisdom-hierarchy-from-signals-to-artificial-intelligence-and-beyond/

Depiction of Learning for Human and Machine

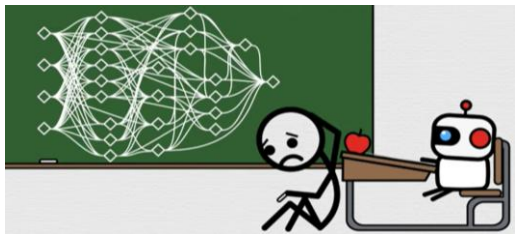
Human Learning

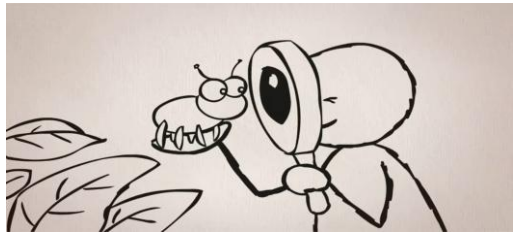


Cognition/Model

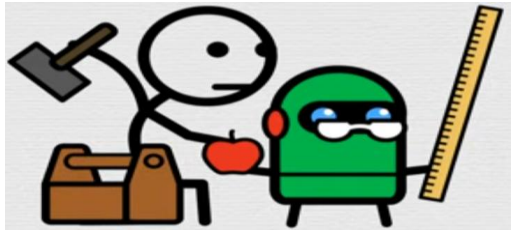


Data Feeding

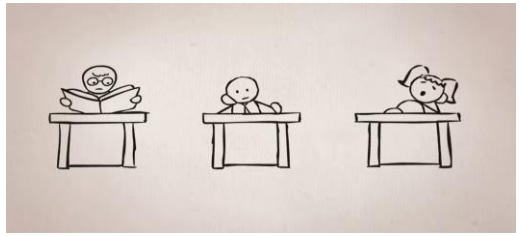




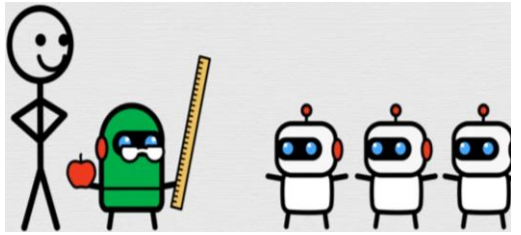
Improvisations



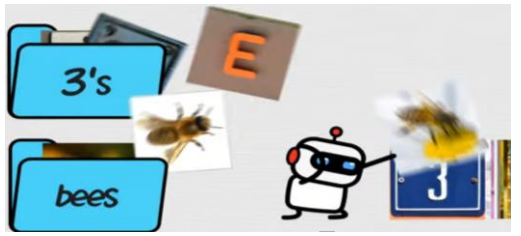
Machine Learning



Training



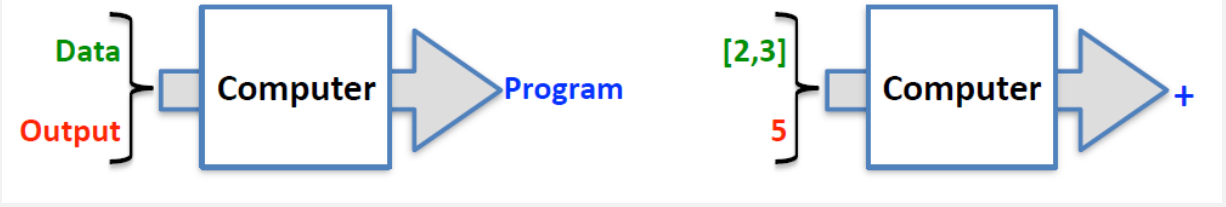
Expertise

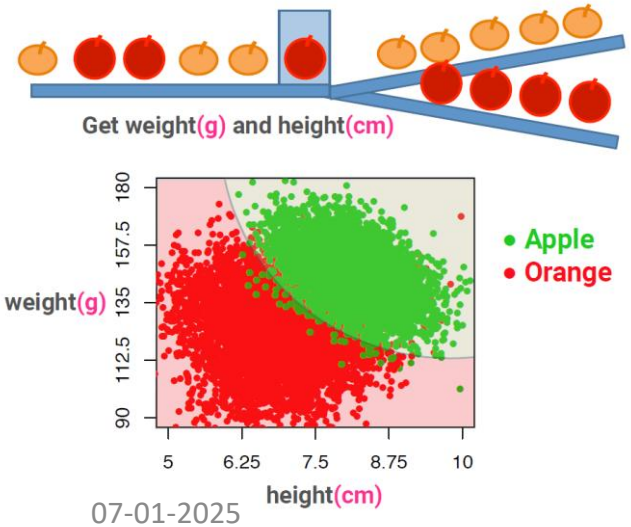


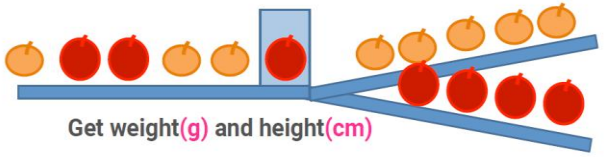
Machine Learning

Traditional Programming

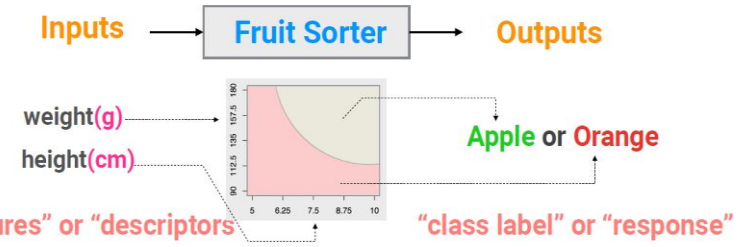
Machine Learning

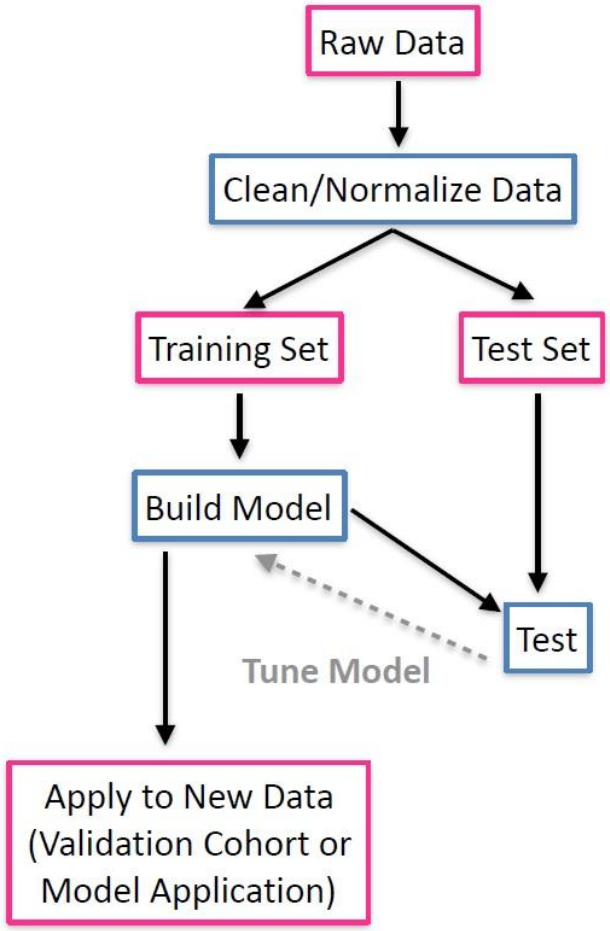






Now we got a computer program for this classification problem.





Unsupervised Learning

Unsupervised Learning

Clustering

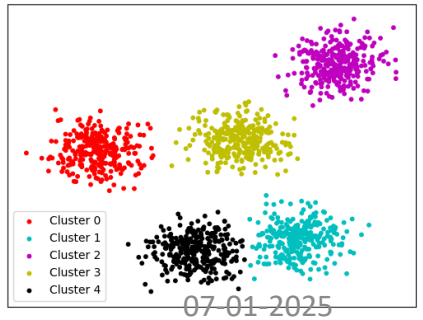
$r_{nk} \in \{0, 1\}$

$$r_{nk} = \begin{cases} 1 & if\ datapoint\ x_n assigned\ to\ cluster\ k \\ 0 & Otherwise \end{cases}$$

$$J = \sum_{n=1}^{K} \sum_{k=1}^{N} r_{nk}\ ||\ x_n - \mu_k||^2$$

$$r_{nk} = \begin{cases} 1 & if\ k = argmin_j\ ||x_n - \mu_k||^2 \\ 0 & Otherwise \end{cases}$$

$$\frac{\partial}{\partial J} = 2 \sum_{n=1}^{N} r_{nk}\ (x_n - \mu_k) = 0 \qquad \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$



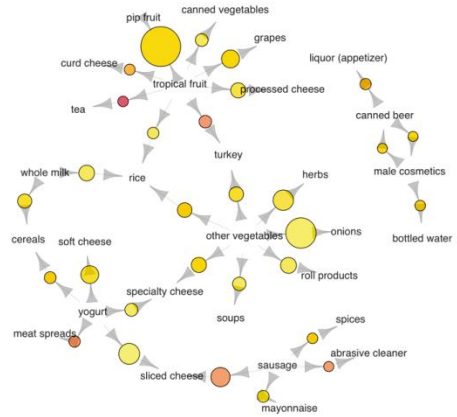
Association Rules

$Rule: x \Rightarrow y$

$$Support\ (S) = \frac{freq(x, y)}{N}$$

$$Confidence\ (C) = \frac{freq(x, y)}{freq(x)}$$

$$Lift\ (L) = \frac{Support)}{Suppurt(x) \times Support(y)}$$



Machine Learning Landscape

Supervised Learning

Classification

[Linear classification]
- Logistic / Softmax regregression
- Linear discriminant analysis
- Naive Bayes classifiers
- Perceptron
- Linear Support Vector Machines (SVM)

[Nonlinear classification]
- k-nearest neighbor classifiers
- Decision trees (Classification trees)
- Polynomial classifiers / Factorization machines
- Tree ensemble classifiers
  - Random Forest classifiers
  - Extra Trees classifiers
  - Gradient Boosted Decision Trees (GBDT)
- Kernel method classifiers
  - Support Vector Machines (SVM)
- Gaussian process classifiers
- Neural network (Deep learning) classifiers
  - Multi-layer perceptrons (MLP)
  - Convolutional networks (CNN)
    - VGG (OxfordNet)
    - Inception (GoogLeNet)
    - ResNet / ResNeXt
    - DenseNet
  - Recurrent networks (RNN)

Regression

[Linear regression]
- Least squares regression
- Principal component regression
- Partial Least Squares (PLS) regression
- Penalized linear regression
  - LASSO regression (L1-penalized)
  - Ridge regression (L2-penalized)
  - ElasticNet regression (L1 & L2-penalized)

[Nonlinear regression]
- k-nearest neighbor regressors
- Decision trees (Regression trees, Model trees)
- Polynomial regressors / Factorization machines
- Tree ensemble regressors
  - Random Forest regressors
  - Extra Trees regressors
  - Gradient Boosted Regression Trees (GBRT)
- Kernel method regressors
  - Support Vector Regression (SVR)
  - Kernel Ridge Regression
- Gaussian process regressors
- Neural network (Deep learning) regressors
  - Multi-layer perceptrons (MLP)
  - Convolutional networks (CNN)
    - VGG (OxfordNet)
    - Inception (GoogLeNet)
    - ResNet / ResNeXt
    - DenseNet
  - Recurrent networks (RNN)

Unsupervised Learning

Clustering
- k-means
- Hierarchical clustering
- Gaussian mixtures
- Spectral methods
- DBSCAN

Decomposition
- Principal component analysis (PCA)
- Independent component analoysis (ICA)
- Canonical correlation analysis (CCA)
- Nonnegative matrix factorization (NMF)
- Latent Dirichlet allocation (LDA)

Manifold learning
- Multidimensional scaling (MDS)
- Self-organizing maps (SOM)
- Isomap
- Locally linear embedding (LLE)
- Spectral embedding (Laplacian eigenmaps)
- t-distributed Stochastic Neighbor Embedding (t-SNE)
- Autoencoders

Density estimation

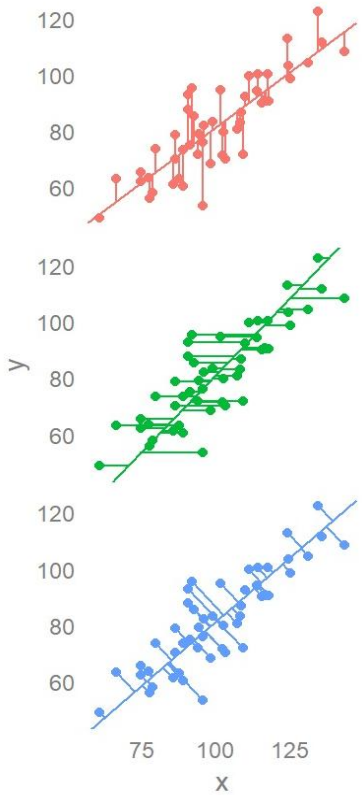
Others

Semi supervised learning
Ranking
Transfer learning
K-shot learning
Domain adaptation
Multitask learning
Reinforcement learning
Imitation learning
Active learning
Model-based optimization
Time series/Sequence models
Probabilistic inference (Bayesian, Generative, Graphical)
Causal inference
Online/Incremental learning
Anomaly/Outlier detection
Ensemble learning
Relational/Network learning
Representation learning
Structured prediction
Meta Learning
:

ACS Catalysis, 2019; 10: 2260-2297. https://doi.org/10.1021/acscatal.9b0418

Supervised Learning

Supervised Learning



Regression

$$y = mx + c$$

$$m_{new} = m_{current} - k\frac{\delta}{\delta m}E(m,c)$$

$$c_{new} = c_{current} - k\frac{\delta}{\delta c}E(m,c)$$

Classification

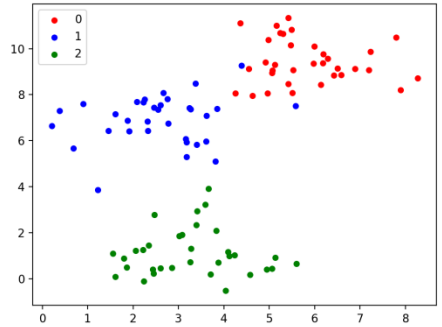
$$class = \begin{cases} C = 1 & if\ P(C = 1|x_1, x_2) > P(C = 0|x_1, x_2) \\ C = o & Otherwise \end{cases}$$

$$P(C|x) = \frac{P(C)p(x|C)}{p(x)}$$

Bayes' Rule

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{\sum_{k=1}^{K} p(x|C_k)P(C_k)}$$

Bayes Classifier



Unsupervised Learning

Unsupervised Learning

Clustering

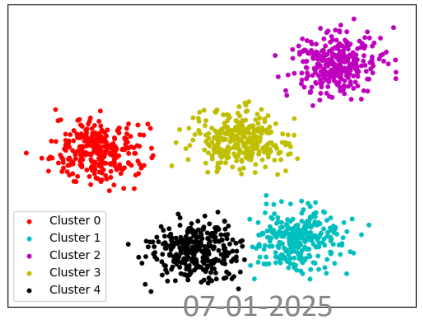
$$r_{nk} \in \{0, 1\}$$

$$r_{nk} = \begin{cases} 1 & if\ datapoint\ x_n assigned\ to\ cluster\ k \\ 0 & Otherwise \end{cases}$$

$$J = \sum_{n=1}^{K} \sum_{k=1}^{N} r_{nk}\ ||\ x_n - \mu_k||^2$$

$$r_{nk} = \begin{cases} 1 & if\ k = argmin_j\ ||x_n - \mu_k||^2 \\ 0 & Otherwise \end{cases}$$

$$\frac{\partial}{\partial J} = 2 \sum_{n=1}^{N} r_{nk}\ (x_n - \mu_k) = 0 \qquad \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$



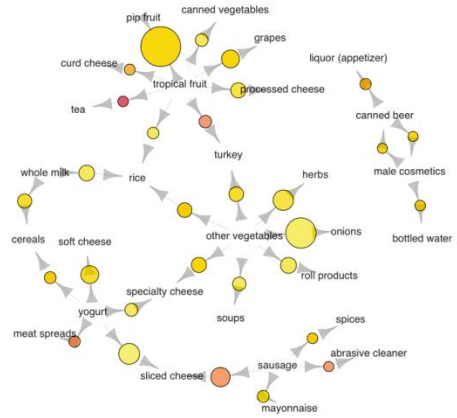
Association Rules

$$Rule: x \Rightarrow y$$

$$Support\ (S) = \frac{freq(x, y)}{N}$$

$$Confidence\ (C) = \frac{freq(x, y)}{freq(x)}$$

$$Lift\ (L) = \frac{Support)}{Suppurt(x) \times Support(y)}$$



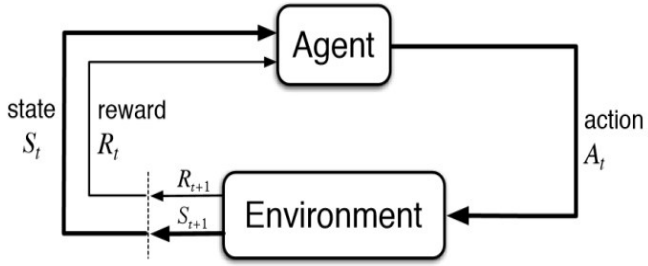
Reinforcement Learning

Reward function, $r(s, a) \rightarrow R$

Markov dynamics, $P(s_t^*|s_{t-1}, a_{t-1}, s_{t-2}, a_{t-2} \dots) = P(s_t|s_{t-i}, a_{t-i}) = P(S)$

Markov Decision Process (MDP): $<S, A, R, T, \gamma>$

S = set of states, A= actions, $R = r(s, a)$, $T = p(s'|s, a)$, $\gamma = (0,1)$

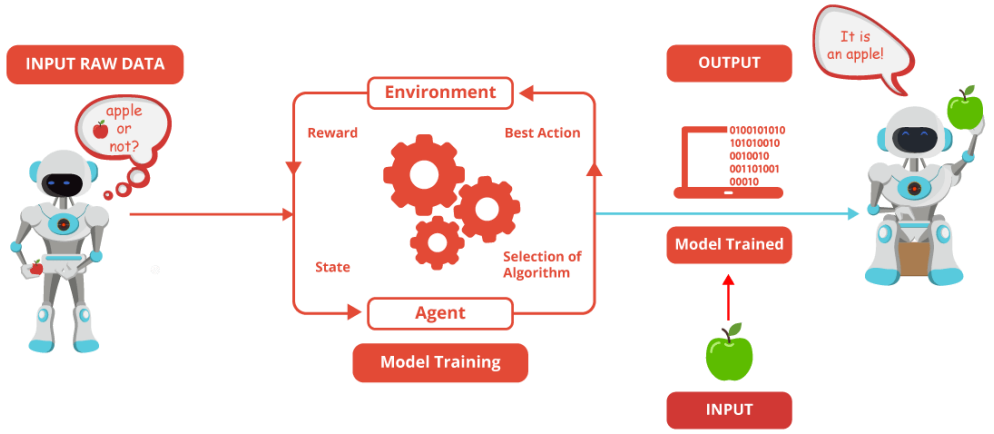


Policy , $\pi: s \rightarrow a$

Model based: directly estimate and use R & T

Value based: use value function $V^{\pi} = E\left[\sum_{i=1}^{T} r_{t+1}\right]$

Policy based: $arg\max_{\pi \in \prod} V^{\pi}$



Human disease = Genetics + Environment + Life Style

Cancer

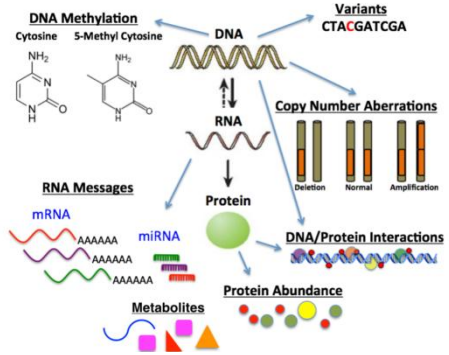
Men:- 1:2 risk of developing
1:4 risk of dying
Women:- 1:3 risk of developing
1:5 risk of dying

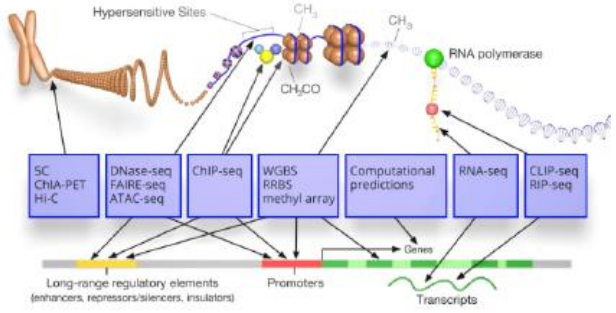
Complex Problems

1. Which patient have high risk?
2. Early biomarkers?
3. Who can be long term or short term survivors?
4. What chemotherauptic might benefit a patient?

Goal of Genomics is to identify genetics/genomic variation related with disease to improve patient care

What we have today?



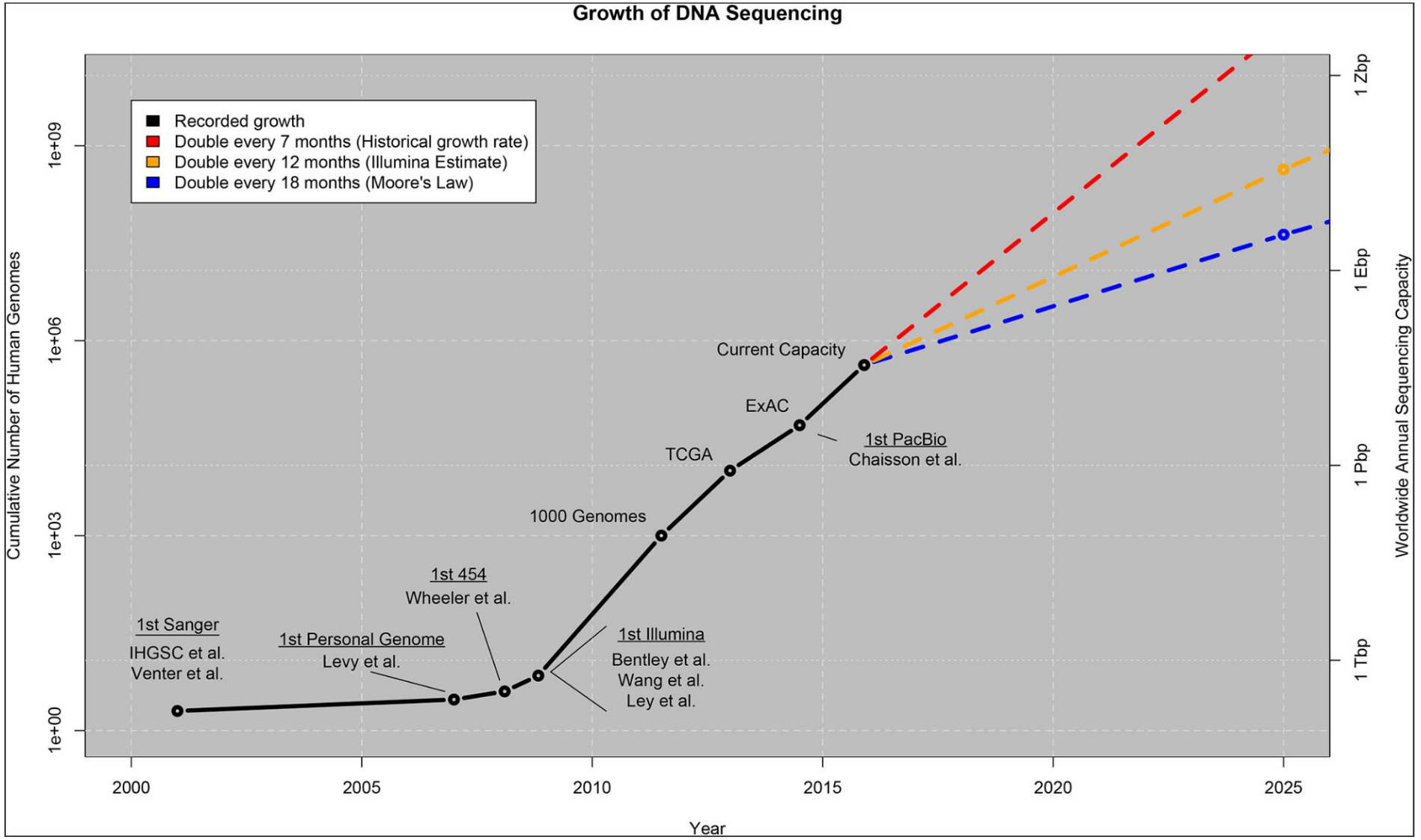


Multidimensional dataset **+** Cell, Tissue, Disease **+** Functional annotations **=** **Big Data**

Complex Problem + Big Data = Machine Learning

Goal of machine learning is not to make a perfect guess but an useful guess that can be used in future

Genomics Data – Big Data



1 zettabyte (ZB) = 1024 EB
1 exabyte (EB) = 1024 PB
1 petabyte (PB) = 1024 TB
1 terabyte (TB) = 1024 GB

Stephens, et al. *PLOS Biology*, 2015

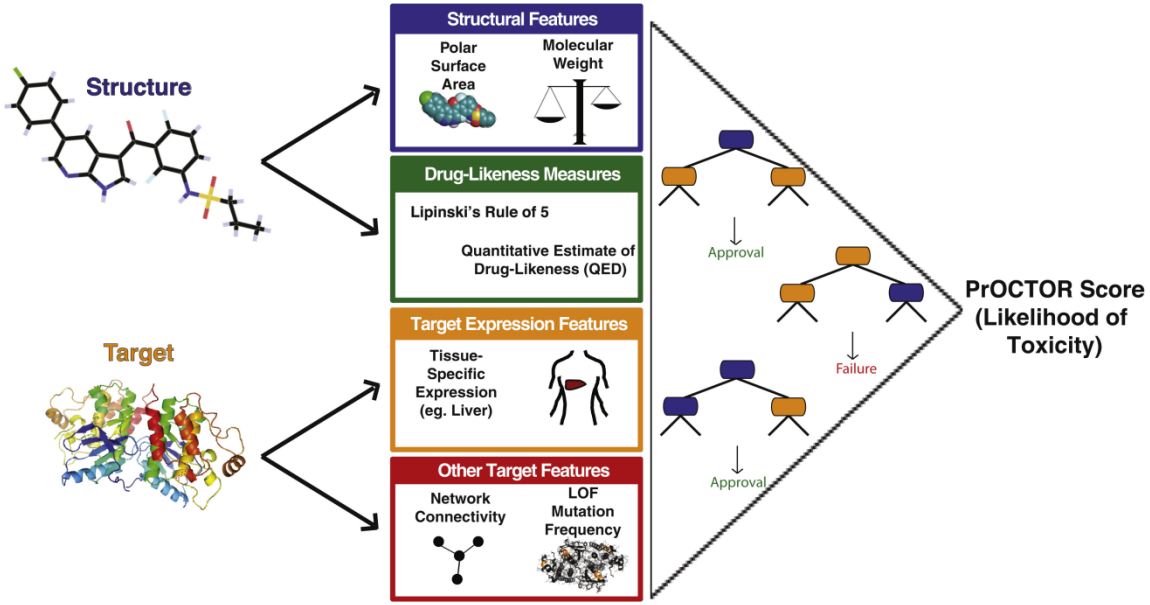
Machine Learning in Drug Discovery

Difficult to identify the unfavourable toxicity properties before conducting clinical trials

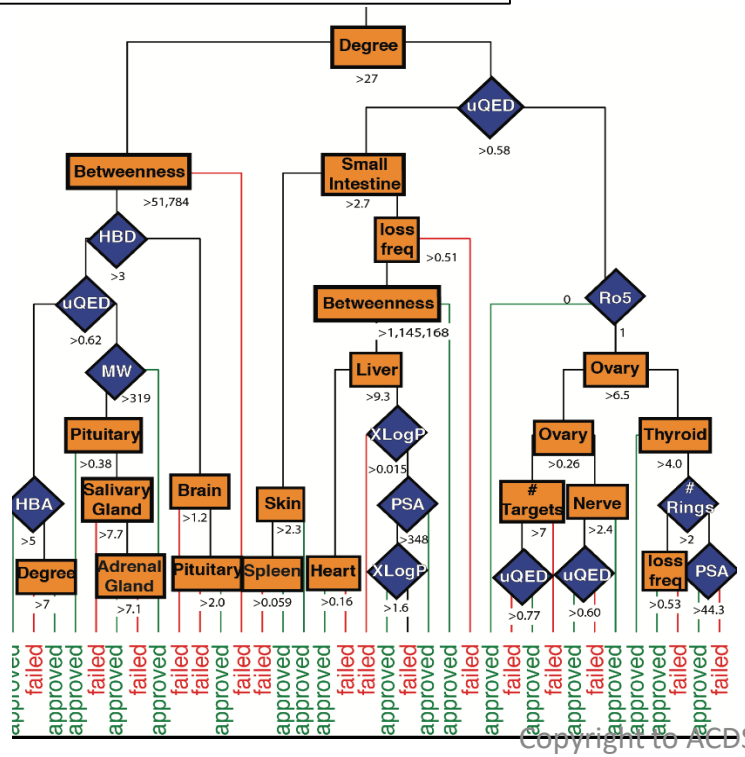
Facts:

1. *FDA approved drugs pass Lipinski Ro5 (80.6%) and Ghose Rule (64.9%)*
2. *FTT drugs pass Lipinski Ro5 (73%) and Ghose Rule (54%)*

Supervised learning with multiple features can predict the likelihood of success and failures in clinical trials



Predicting the odds of clinical trial outcomes using random forest (PrOCTOR)#

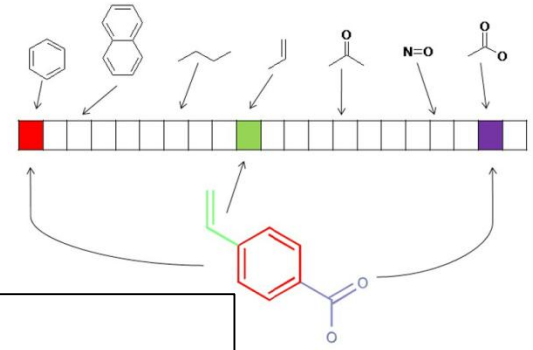


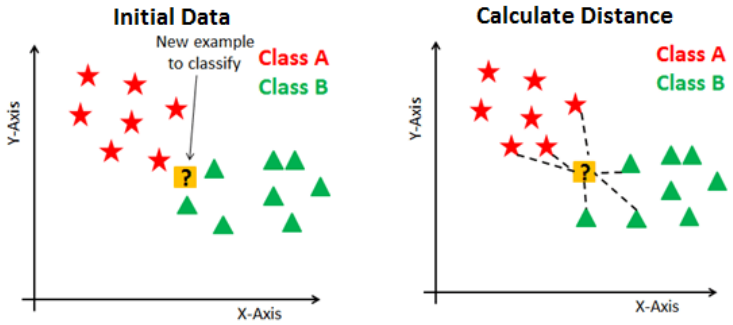
#"A data-driven approach to predicting successes and failures of clinical trials." Cell chemical biology, 2016

Machine Learning in Chemistry

No predefined basis for acceptance or exclusion of a given chemical into a category

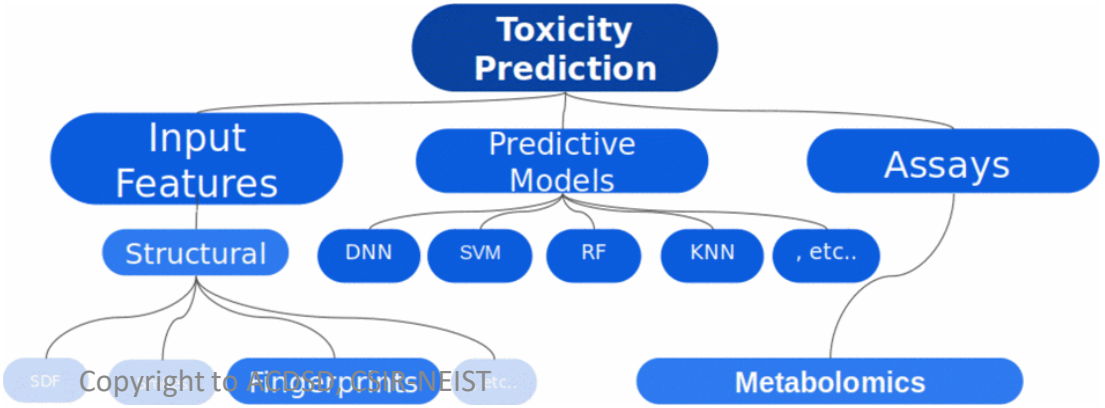
Design a classification model, $y = f(x)$ where
y: response variable & x: fingerprint





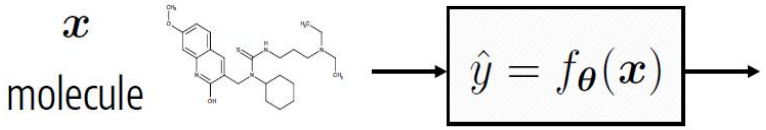
Class A & *class B are acute toxicity based categories*

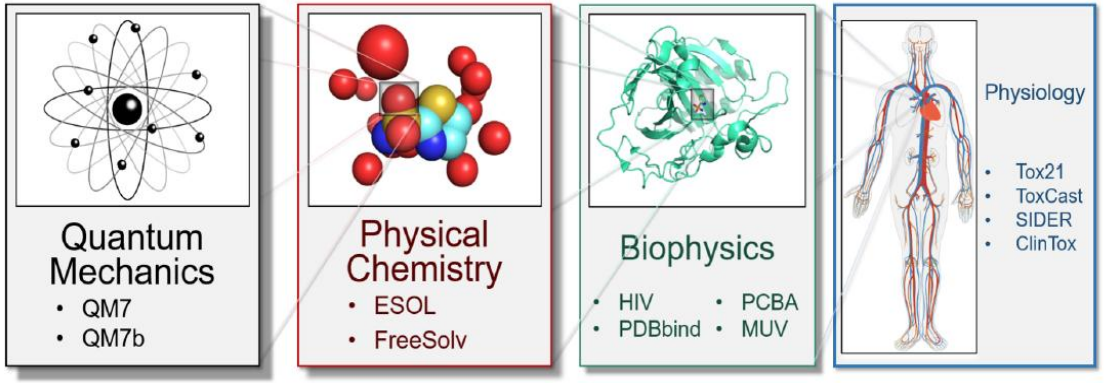
Step. 1: Label the dataset with category for training
Step. 2: For new chemical, calculate distance using techniques like "*Jaccard -Tanimoto*"
Step. 3: Identify closest neighbour using distance matrix
Step. 4: Out of the *k* nearest neighbours, the category having maximum values will be the category of the new chemical
Step. 5: Optimal value of *k* can be found using cross validation



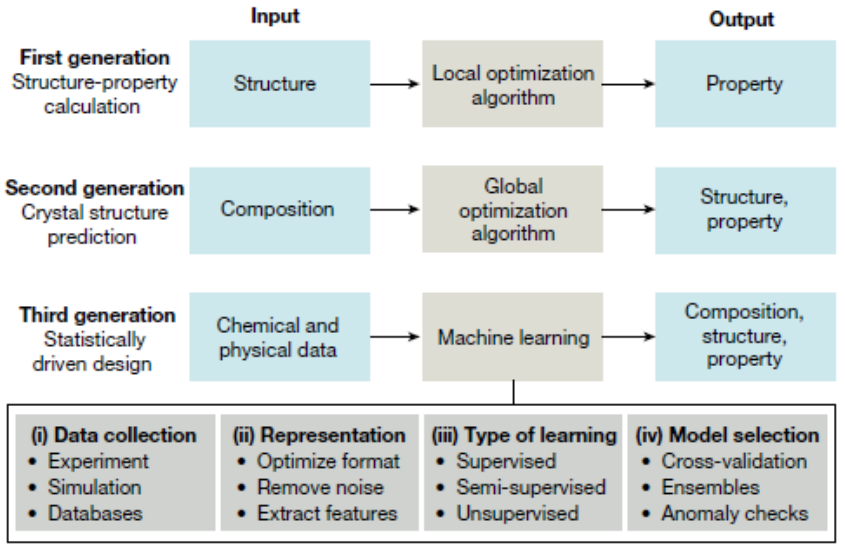
Machine Learning in Molecular & Material Science

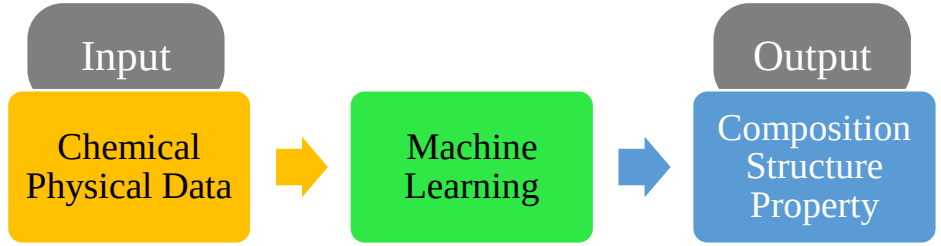
Quantitative structure–activity/property relationship (QSAR/QSPR)



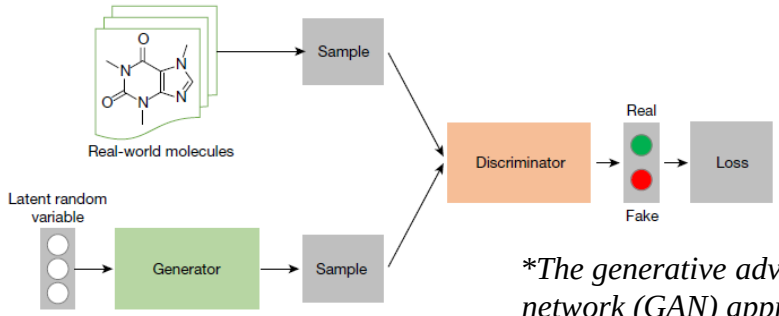


MoleculeNet[#]: A Benchmark for Molecular Machine Learning



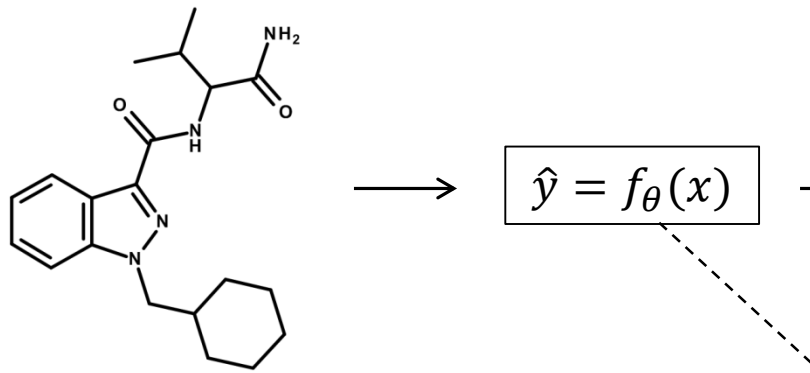


- Assess the likelihood that a product will crystallize
- Targeting discovery of new compounds relating to system descriptors



**The generative adversarial network (GAN) approach to molecular discovery*

[#]*https://arxiv.org/abs/1703.00564*
[#]*https://github.com/deepchem/deepchem*
**Machine learning for molecular and materials science, Nature Reviews, 2018*



$\boldsymbol{X}$

molecule

$\hat{y} = f_\theta(x)$

$\hat{y}$

property value

approximate for the (unknown) underlying mapping for $x \longmapsto y$

"machine teaching"

Given n input-output instances as the training data

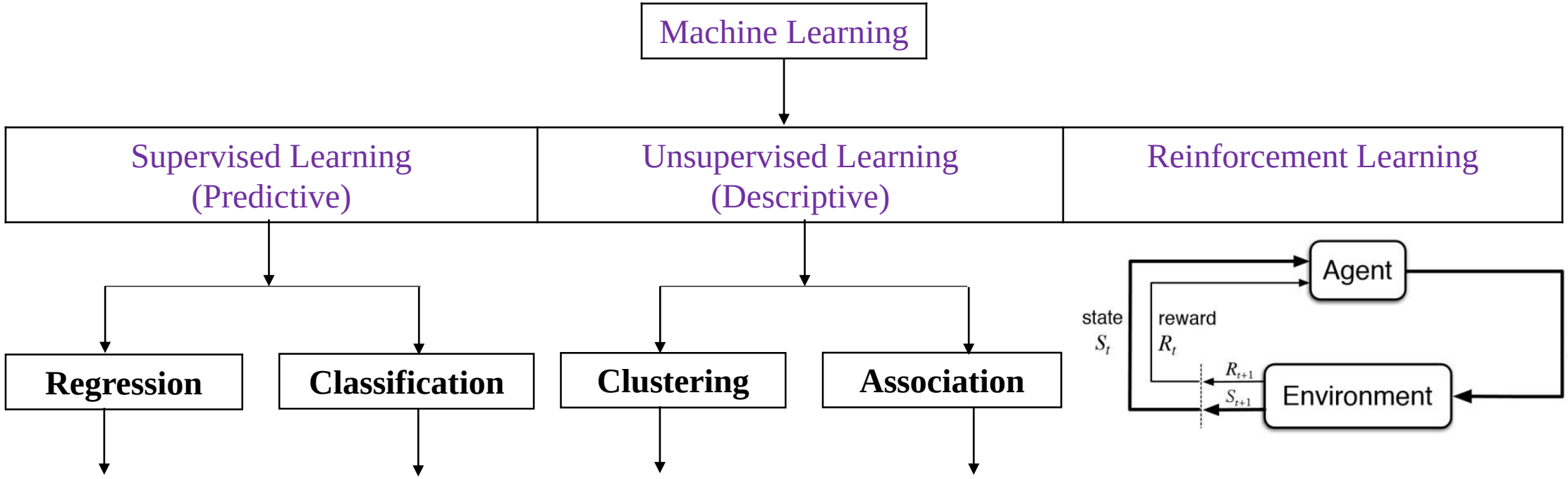
$\{(x_1, y_1), (x_2, y_2), .. , (x_n, y_n)\}$

Fit the model f_θ by tuning Θ as

$min_\theta \sum_{i=1}^{n} error(y_i, \hat{y}_i)$ where $\hat{y}_i = f_\theta\ (x_i)$

Note: the error measure (called "loss function" in ML) depends on problems

Types of Machine Learning

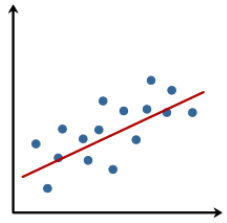


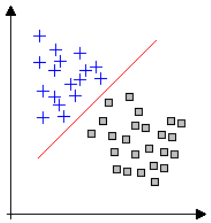
Predict continuous output

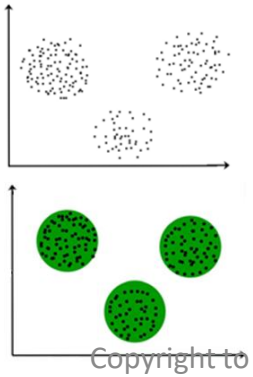
Predicting discrete class

Finding natural grouping

Finding associations among variables



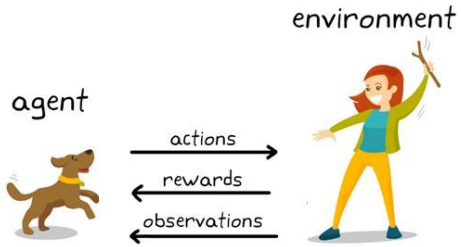




$\{X_1, X_2\}$ → $\{Y_1, Y_2\}$

Antecedent Consequent

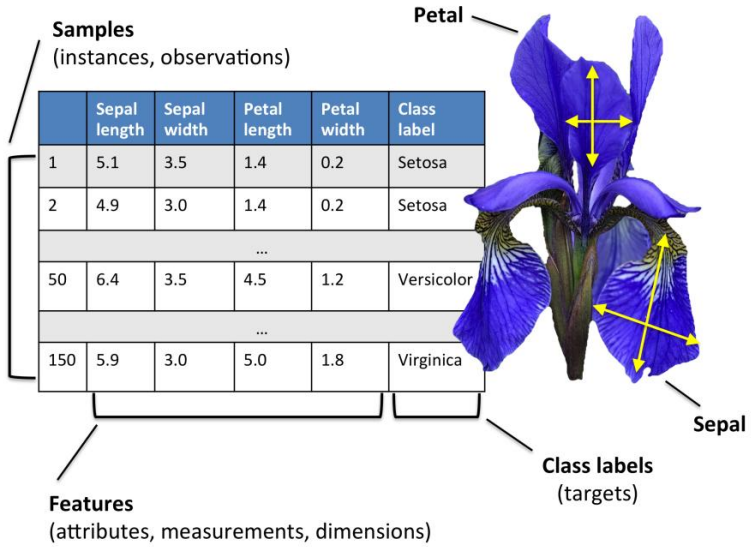
Itemset = $\{X_1, X_2, Y_1, Y_2\}$



Datasets and Learning

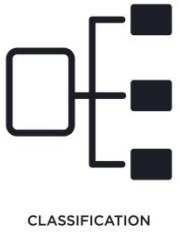

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa

##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor

##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 101          6.3         3.3          6.0         2.5 virginica
## 102          5.8         2.7          5.1         1.9 virginica
## 103          7.1         3.0          5.9         2.1 virginica
## 104          6.3         2.9          5.6         1.8 virginica
## 105          6.5         3.0          5.8         2.2 virginica
```

Dataset 1: Labelled data



Supervised Learning

Make classification of all data based on labels



Three classes "setosa" or "versicolor" or "virginia"

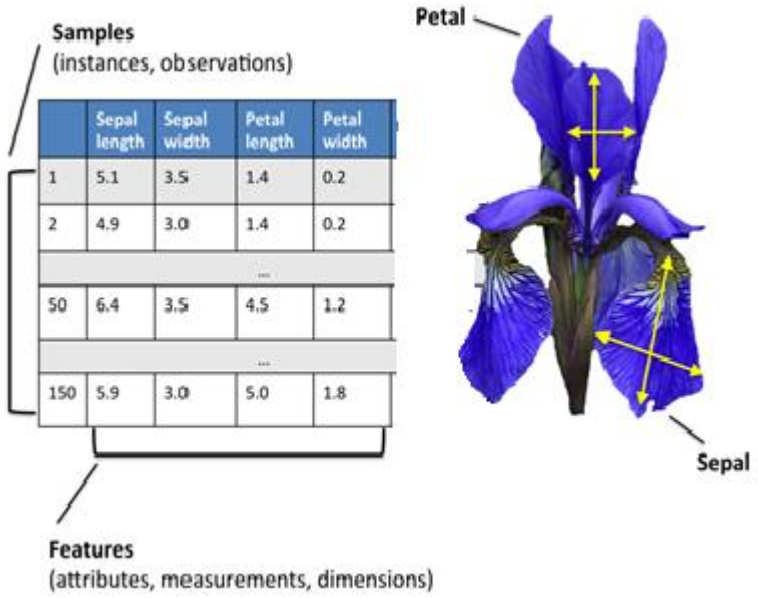
New data with sepal length, sepal width, petal length and petal width but no label??

It belongs to "setosa" or "versicolor" or "virginia"

Datasets and Learning

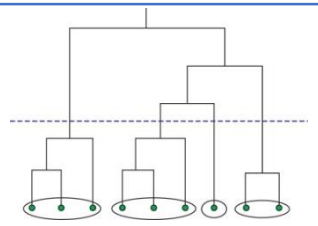
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
## 4          4.6         3.1          1.5         0.2
## 5          5.0         3.6          1.4         0.2

##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 51          7.0         3.2          4.7         1.4
## 52          6.4         3.2          4.5         1.5
## 53          6.9         3.1          4.9         1.5
## 54          5.5         2.3          4.0         1.3
## 55          6.5         2.8          4.6         1.5

##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 101          6.3         3.3          6.0         2.5
## 102          5.8         2.7          5.1         1.9
## 103          7.1         3.0          5.9         2.1
## 104          6.3         2.9          5.6         1.8
## 105          6.5         3.0          5.8         2.2
```

Dataset 2: Unlabelled data



Unsupervised Learning

Cluster the data based on similarity and differences

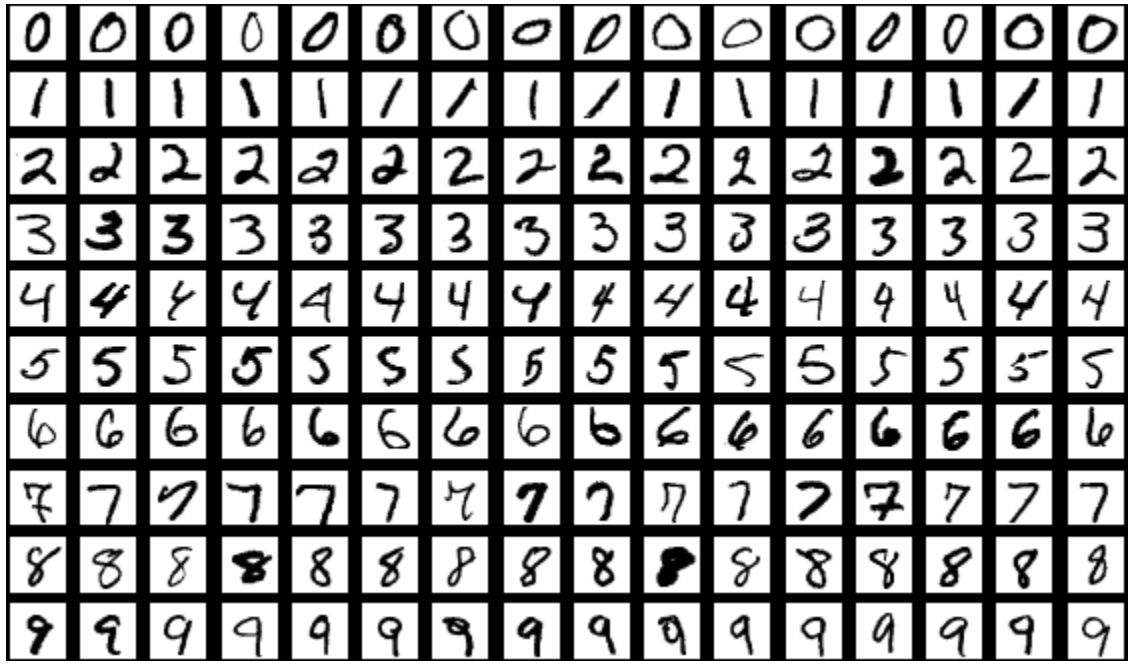


Clusters of similar 'petal length', 'petal width' ..

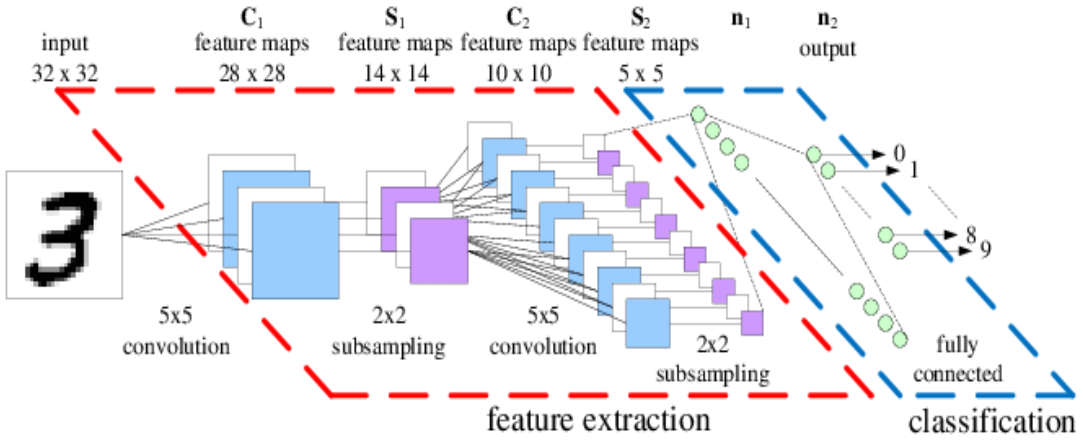
New data with sepal length, sepal width, petal length and petal width but no label??

Identify the cluster to which the new sample belongs

Datasets and Learning



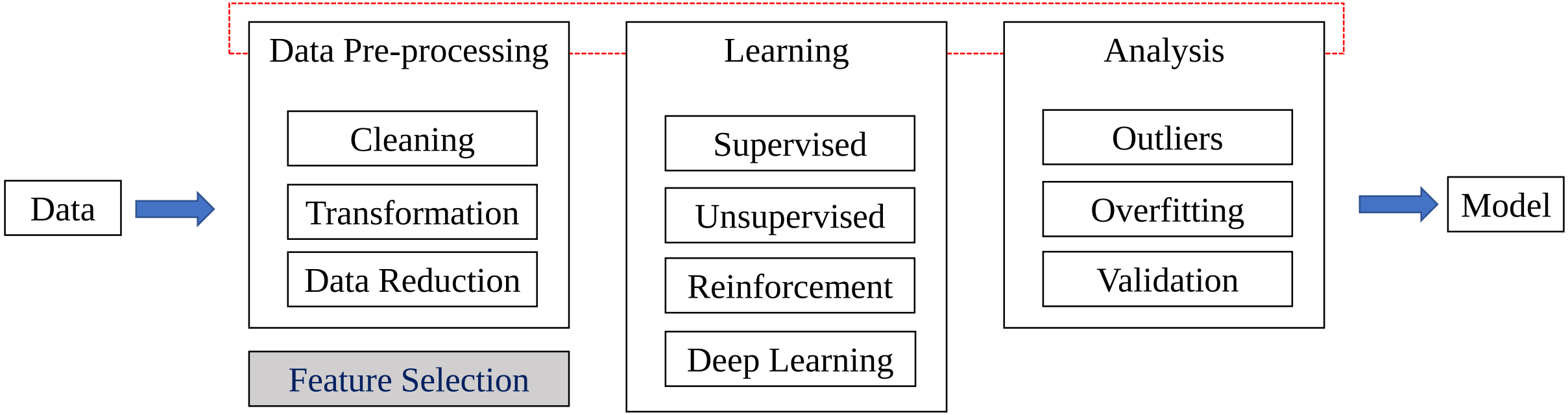
Dataset 3: Unlabelled data



Deep Learning

CLUSTERING

Data Pre-processing



Data Preprocessing		
Data Cleaning	**Data Transformation**	**Data Reduction**
Noisy Data	Discretization	Aggregation
Missing Data	Attribute Selection	Feature Subset Selection
Duplicate Data	Feature Scaling	Dimensionality Reduction

Noisy values

- Due to noise (Data disrupted)
- Handled by collecting more data, principle component analysis, regularization and cross-validation

Missing values

- Due to information not collected or attribute not applicable to all cases.
- Handled by either ignoring/dropping or replacing with some values (mean, median, max_count etc.), predicting missing values, imputation methods

Duplicate data

- Due to collection of data from multiple sources.
- Handled by deleting the duplicate entry.

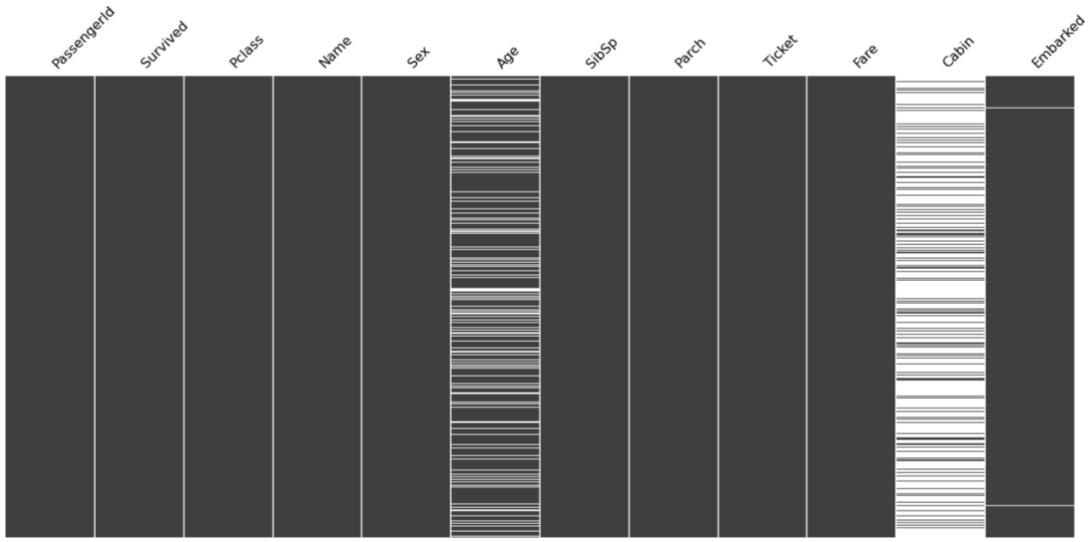
Data Cleaning

Noisy Dataset Example

Value	Colour	Class
0.25	Red	Positive
0.25	Red	Negative
0.98	Green	Negative
1.02	Green	Positive
2.05	?	Negative
=	Green	Positive

Attribute noise **Class noise**

Visualization of missing data in a Dataset



Duplicate data in a Dataset

ID	0	1	2	3	4
34	4.9	3.1	1.5	0.1	Iris-Setosa
37	4.9	3.1	1.5	0.1	Iris-Setosa
142	5.8	2.7	5.1	1.9	Iris-Virginica

Duplicate data

Data Transformation

- **Discretization:** Replace the raw values of numeric attribute by interval levels or conceptual levels.
- **Attribute Selection:** New attributes are constructed from given set of attributes
- **Feature Scaling:** Scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
  - Min-max (Normalization): $v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$
  - Z score (standardization): $v' = \frac{v - mean_A}{stand\_dev_A}$ ***or*** $z_{ij} = \frac{x_{ij} - m_j}{s_j}$

x_{ij} is the value for the i^{th} sample and j^{th} feature
m_j is the average of all x_{ij} for feature j
s_j is the standard deviation of all x_{ij} over all input samples

Discretization

Data discretization converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy.

Techniques for discretization:

Histogram analysis: A plot used to present the underlying frequency distribution of a set of continuous data.
Binning: A data smoothing technique and its helps to group the a huge number of continuous values into a smaller number of bins.
Correlation analysis: Extensively used technique to identifies interesting relationships in data. These relationships help us realize the relevance of attributes with respect to the target class to be predicted.
Cluster analysis: Also commonly known as clustering, is the task of grouping with similar objects in one group, commonly called cluster
Decision tree analysis: Use a decision tree to identify the optimal number of bins. When the model makes a decision, it assigns an observation for each node.
Equal width partitioning: Separating all possible values into 'n' number of bins, each having the same width.

$$width = \frac{(maximum\ value - minimum\ value)}{n}$$ #where n is the number of bins or intervals.

Age	10,11,13,14,17,19,30, 31, 32, 38, 40, 42,70 , 72, 73, 75

Table: Before discretization

Attribute	Age	Age	Age
	10,11,13,14,17,19,	30, 31, 32, 38, 40, 42	70 , 72, 73, 75
After Discretization	Young	Mature	Old

Attribute Selection

The importance of features (attributes) selection:

- Reduce the cost of learning by reducing the number of attributes.
- Provide better learning performance compared to using full attribute set.

There are two approach for attribute selection.

- Filter approach attempt to assess the merits of attributes from the data, ignoring learning algorithm.
- Wrapper approach the attributes subset selection is done using the learning algorithm as a black box.

Most simple filter approach is ranking of attributes according value chi-square for two way tables. Two way table for this case is confusion matrix.

$$x^2 = \sum \frac{(observed\ count - expected\ count)^2}{expected\ count}$$

The expected count in any cell a two-way table is

$$expected\ count = \frac{row\ total\ \times column\ total}{total\ count}$$

		PREDICTED CLASS	
		D^0	BG
ACTUAL CLASS	D^0	TP	FN
	BG	FP	TN

Large value χ^2 better attribute

Feature Scaling – Normalization

Preparing data for processing in machine learning. Normalization brings all data in the data set into a common scale.

Techniques for normalization

- Z-Score Normalization: $z = \frac{x - \mu}{\sigma}$ where, Z = standard score, x = observed value, μ = mean of the sample σ = standard deviation of the sample
- Min-Max Normalization: $\frac{value - min}{max - min}$, where max and min are the maximum and minimum values in the dataset
- Decimal scaling Normalization: $v' = \frac{v}{10^j}$, where j is the lowest integer while Max (|) < 1
- Standard Deviation Normalization: $s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$ where n it the number of data points, x_i is the data point, x' mean of x_i

Normalization

Worked out example on normalizing the data using Min-Max

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
df = pd.DataFrame({
    'Income': [15000, 1800, 120000, 10000],
    'Age': [25, 18, 42, 51],
    'Department': ['HR','Legal','Marketing','Management']
})
```

In [4]:
```
df
```

Out[4]:

	Income	Age	Department
0	15000	25	HR
1	1800	18	Legal
2	120000	42	Marketing
3	10000	51	Management

In [7]:
```
df_scaled = df.copy()
col_names = ['Income', 'Age']
features = df_scaled[col_names]
```

In [8]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

In [9]:
```
df_scaled[col_names] = scaler.fit_transform(features.values)
```

In [10]:
```
df_scaled
```

Out[10]:

	Income	Age	Department
0	0.111675	0.212121	HR
1	0.000000	0.000000	Legal
2	1.000000	0.727273	Marketing
3	0.069374	1.000000	Management

Scaling the data without range

In [11]:
```
scaler = MinMaxScaler(feature_range=(5, 10))

df_scaled[col_names] = scaler.fit_transform(features.values)
df_scaled
```

Out[11]:

	Income	Age	Department
0	5.558376	6.060606	HR
1	5.000000	5.000000	Legal
2	10.000000	8.636364	Marketing
3	5.346870	10.000000	Management

Scaling the data with range (5-10)

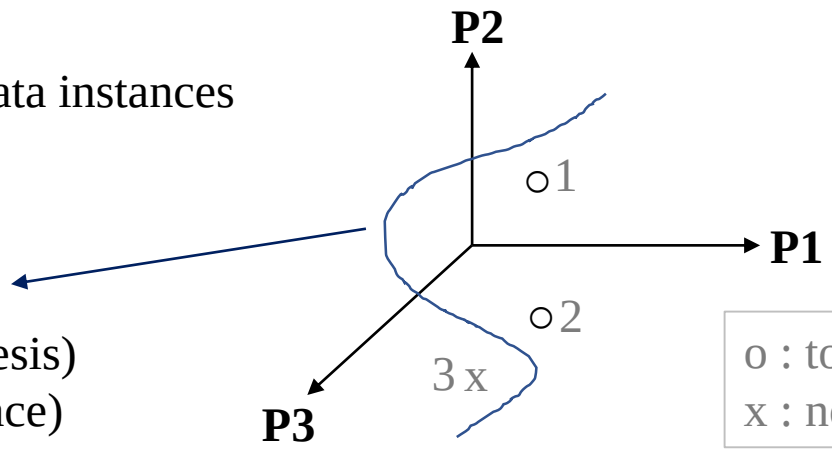
Let us consider the following example – Toxicity prediction (toxic or not?)

(*independent variables*) (*dependent variable*)

Features / Descriptors / Attributes Class

	Property1	Property2	Property3	Toxic or not?
Compound 1 →	a	b	9	yes
Compound 2 →	p	q	20	yes
Compound 3 →	x	y	-50	no

Data items / Data instances



Classifier
(model hypothesis)
(decision surface)

o : toxic
x : non toxic

Accuracy?

But, Is it enough?

- Are data items enough
- Are features enough

Resampling Methods

Let us include more data items and more features to increase accuracy of the model -

	P1	P2	P3	P4	…	…	…	P97	P98	P99	P100	Toxic or not?
Compound 1 →												yes
Compound 2 →												yes
Compound 3 →												no
…	…											
…	…											
…	…											
Compound 998 →												no
Compound 999 →												no
Compound 1000 →												no

Now, Is it enough?

- Are data items enough
- Are features enough

If yes →

What about performance?

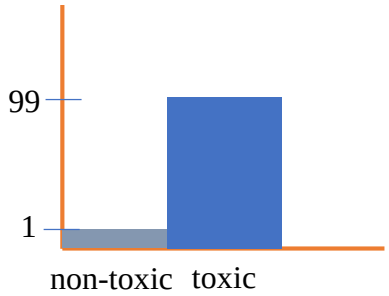
Overfitting - Resampling

Curse of dimensionality - Dimension Reduction

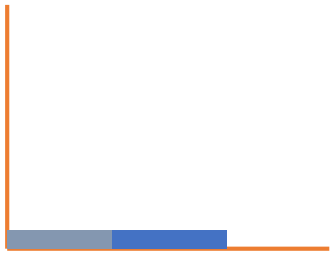
Resampling Methods

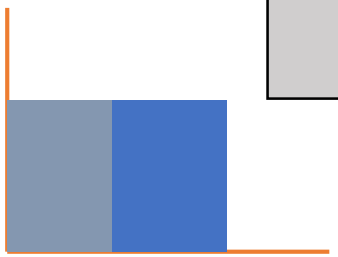
Resampling

- Dataset Perspective (imbalanced dataset)
  - Undersampling
  - Oversampling
  - Random sampling
- Model Performance Perspective (overfitting)
  - Hold out validation
  - K-fold cross validation
  - Bootstrapping




```
check_data(x) {
        model();
        }

model() {
        print("Toxic");
        }
```


Undersampling


Oversampling

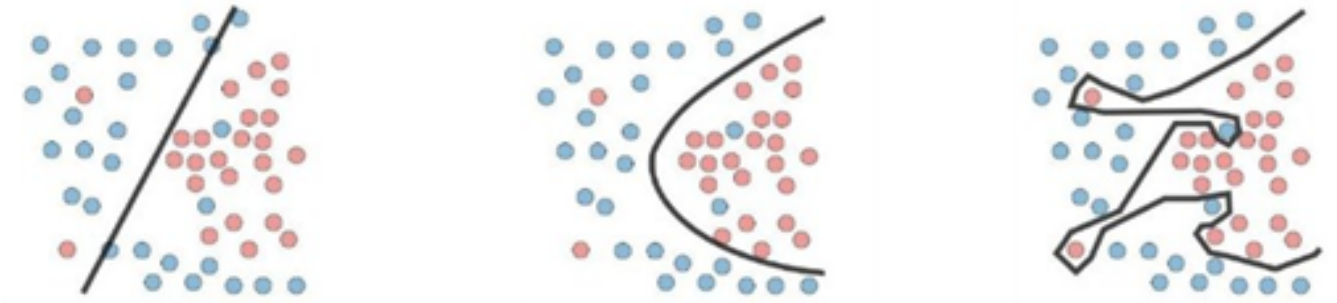
	toxic	non-toxic
toxic	99	1
non-toxic	0	0

99% accuracy on train data (without doing anything)

Resampling Methods

- Generated models have two important aspects to fulfill (learning and generalisation):
  1. They must learn decision surfaces correctly (zero error) from training data.
  2. They must be able to generalize i.e. deal appropriately with new data.
- Usually we want our models to learn well and also to generalize well.

- Zero error is possible - training too much!, but leads to overfitting - trade-off between learning and generalization.
- Overfittings may fit the training data well, even outliers but fail to generalize to new data - loose performance on new data.
- Allowing not classify all training data totally accurately (allowing misclassification) likely to lead better generalization.
- Thus non-perfect learning is better in this case!



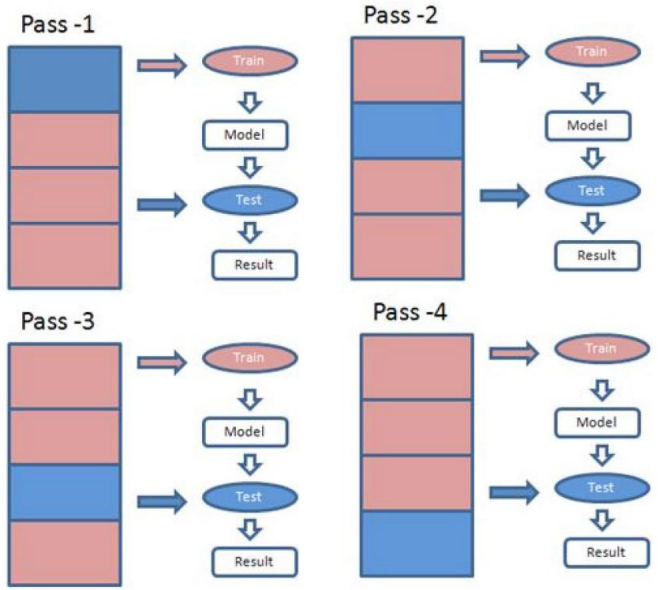
Occam's Razor (1300s): *"plurality should not be assumed without necessity"* - The simplest model which explains the majority of the data is usually the best.

Resampling Methods

- Split-sample or hold-out validation (60-20-20)
  - Reserve some data as a "test set", which must not be used during training.
  - Disadvantage is that it reduces the amount of data available for both training and validation.

- K-fold cross-validation (e.g., leave one out)
  - Data is divided into k subsets and train k times, each time leaving one subset out for computing the error.
  - "Crossing" makes an improvement over split-sampling allowing all data used for training.
  - Disadvantage is that the network must be re-trained many times (k times in k-fold crossing).

- Bootstrapping
  - Works on random sub-samples (random shares) chosen from the full data set.
  - Any data item may be selected any number of times for validation.
  - The sub-samples are repeatedly analysed.



How long should a model be trained?

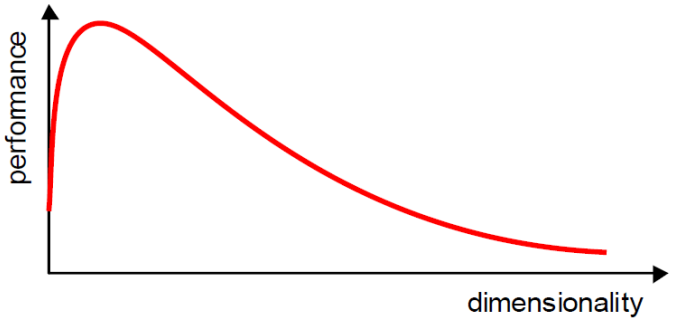
- Stop if the error reaches an acceptable level (e.g., 95%)
- Stop if the error fails to improve (has reached a minimum)
- Stop if the rate of improvement drops below a certain level

➢ The objective is to establish a balance between memorization and generalization.

Dimensionality Reduction

Curse of dimensionality

- Increasing the number of features will not always improve classification accuracy.
- In practice, the inclusion of more features might actually lead to worse performance.



Dimensionality Reduction

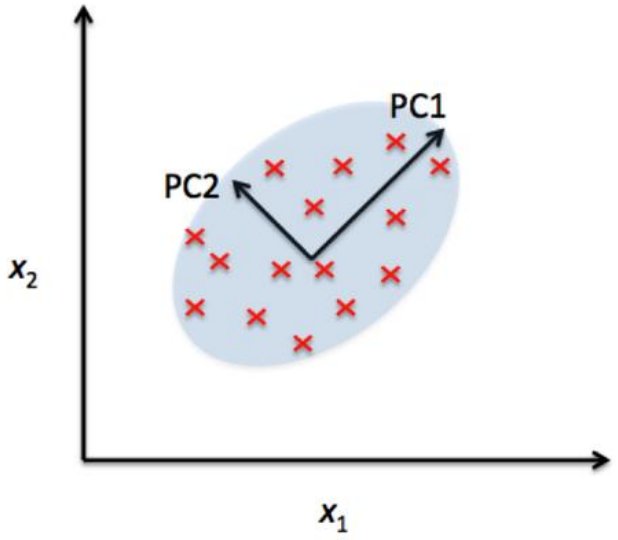
Finds a set of new features from existing features (i.e., through some mapping f())

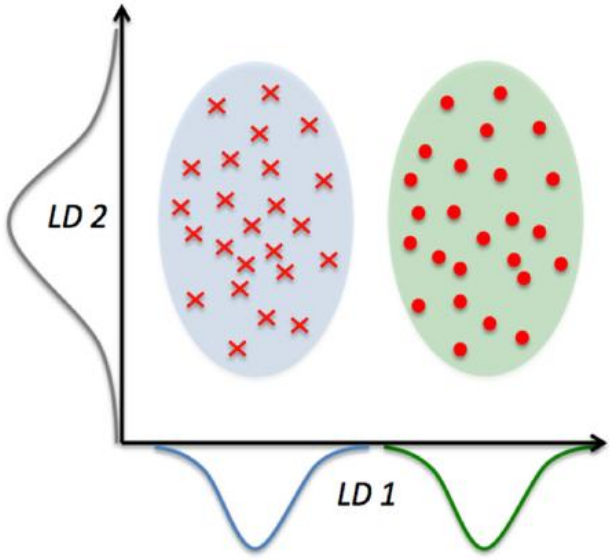
- PCA
- LDA

Dimensionality Reduction

- Principal Components Analysis (PCA): Seeks a projection **component** that **preserves** as much **information** in the data as possible.

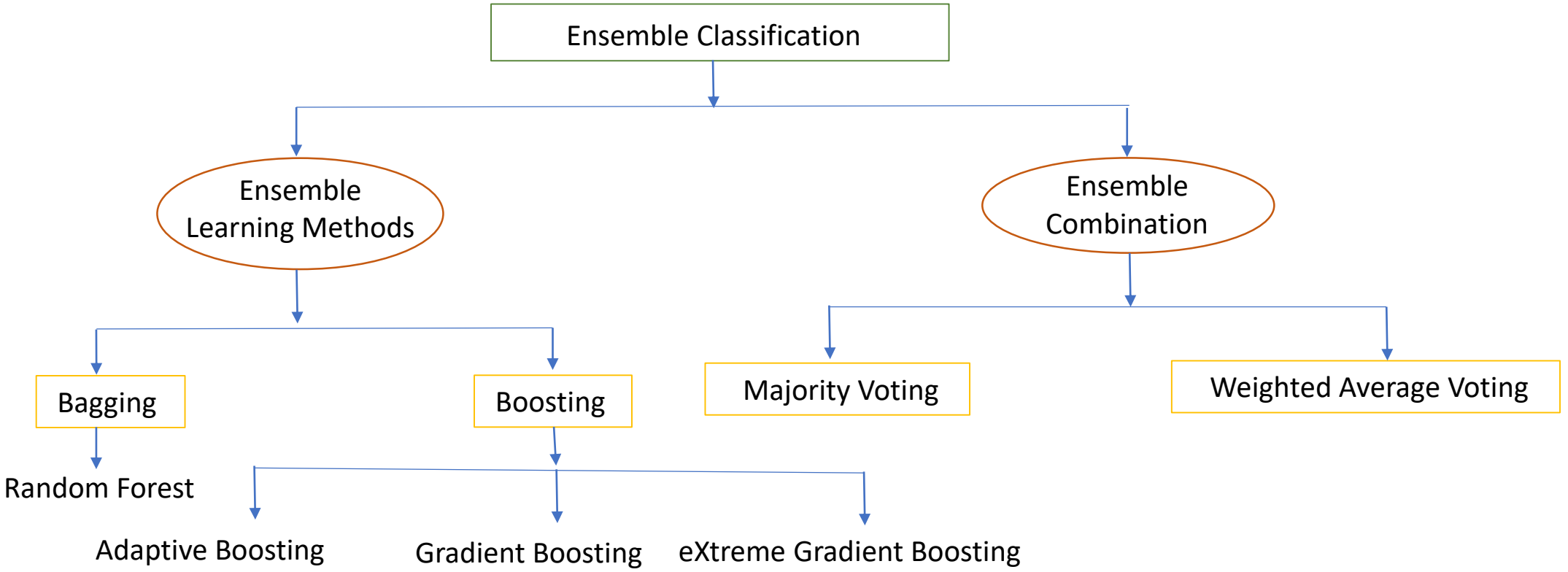
- Linear Discriminant Analysis (LDA): Seeks a projection that **best discriminates** the data.





Ensemble Learning

- Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem.
- Ensemble learning is primarily used to improve the (classification, prediction, function approximation, etc.) performance of a model, or reduce the likelihood of an unfortunate selection of a poor one.



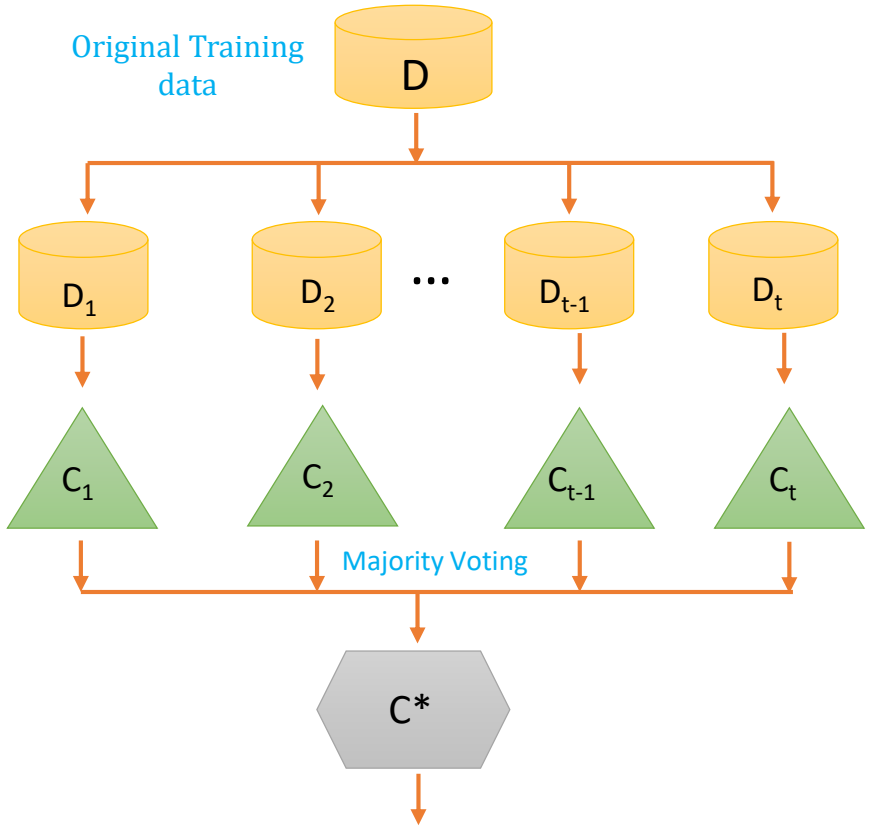
Ensemble Method

Method	Minimize Bias?	Minimize Variance?	Other Comments
Bagging	Complex model class. (Deep DTs)	Bootstrap aggregation (resampling training data)	Does not work for simple models.
Random Forests	Complex model class. (Deep DTs)	Bootstrap aggregation + bootstrapping features	Only for decision trees.
Gradient Boosting (AdaBoost)	Optimize training performance.	Simple model class. (Shallow DTs)	Determines which model to add at run-time.
Ensemble Selection	Optimize validation performance	Optimize validation performance	Pre-specified dictionary of models learned on training set.

- State-of-the-art prediction performance
  - Won Netflix Challenge
  - Won numerous KDD Cups
  - Industry standard

Bagging

- Bagging/bootstrap aggregation is a way to decrease the variance in the prediction by generating additional data for training from dataset using combinations with repetitions to produce multi-sets of the original data.
- In general it reduces variance of an estimate by taking mean of multiple estimates.



Step-1

Create randomly sampled datasets of the original training data (bootstrapping)

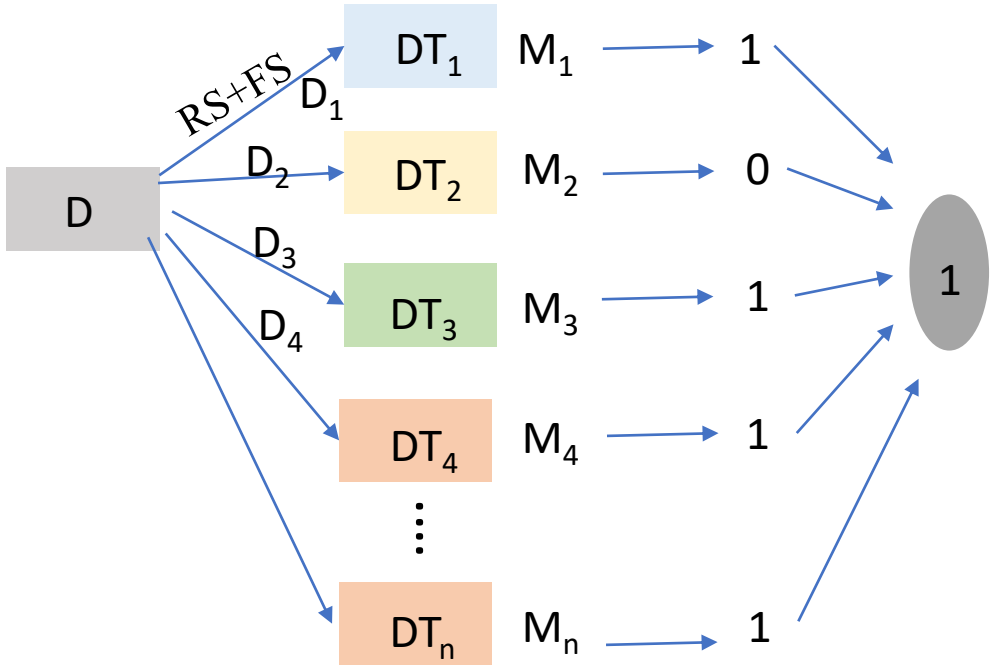
Step-2

Built and fit several classifiers to each of these diverse copies

Step-3

Take the average of all the predictions to make a final overall prediction (Combine classifiers)

Random Forest



- Random forest technique combines various decision trees to produce a more generalized model.
- Random forests are utilized to produce de-correlated decision trees.
- Random forest creates random subsets of the features.
- Smaller trees are built using these subsets, creating tree diversity.
- To overcome overfitting, diverse sets of decision trees are required.

How Random Forest Works?

The dataset consisting of :

- Weather information of last 14 days
- Whether match was played or not on that particular day

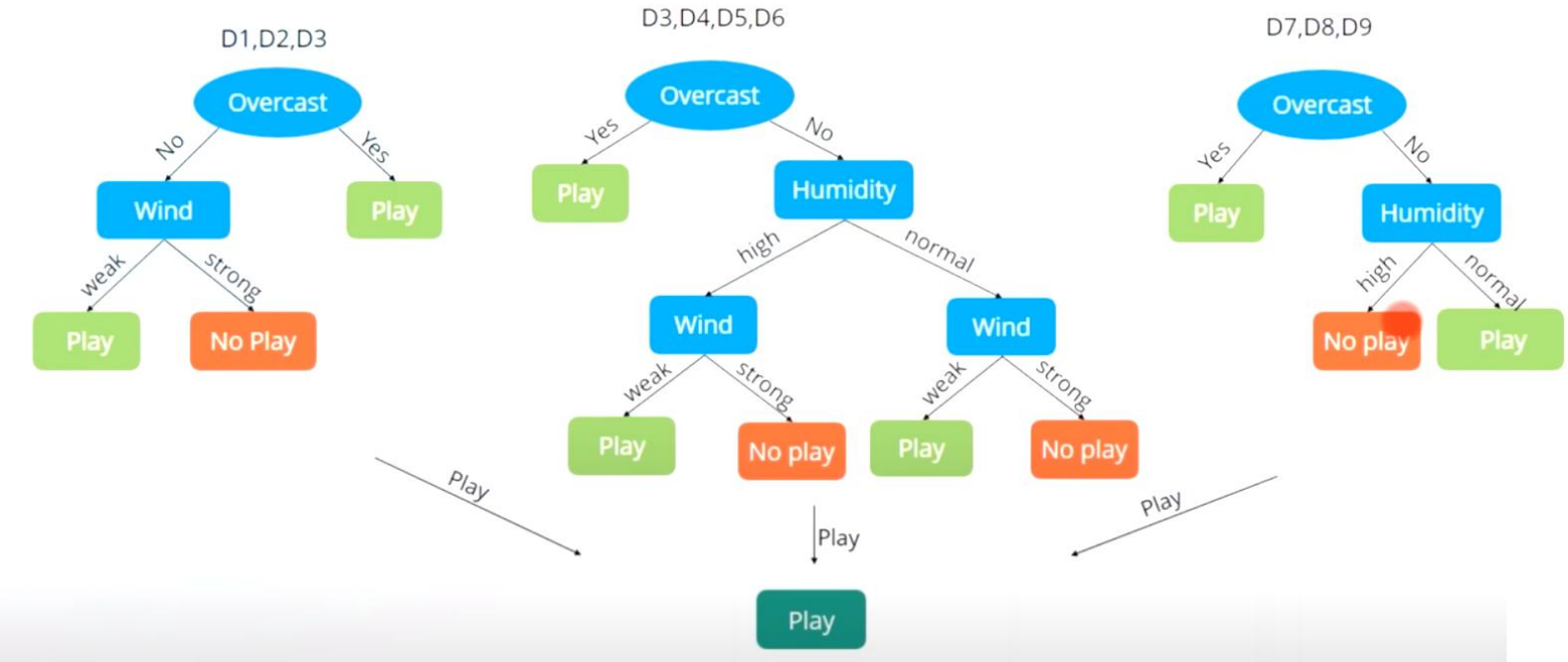
Now using random forest we need to predict whether the game will happen if the weather condition is

Outlook = Rain
Humidity = High
Wind = Weak
Play = ?

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

How Random Forest Works?

- Step 1: The first step in random forest is that it will divide the data into smaller subsets.
- Every subset need not be distinct, some subsets may be overlapped.

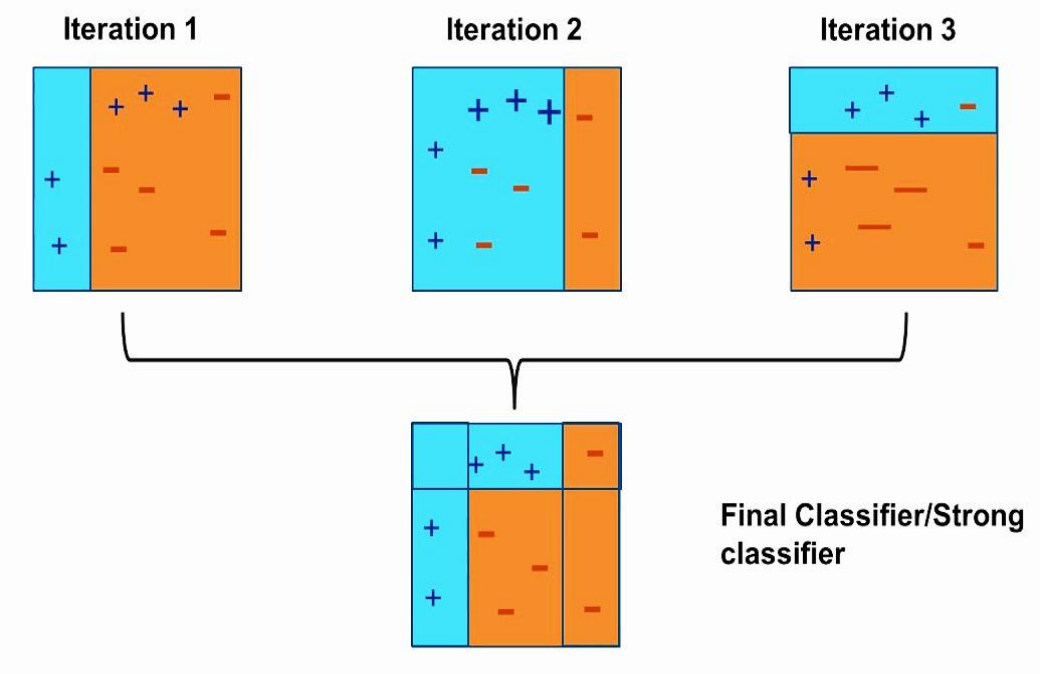


Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Features of Random Forest

Boosting

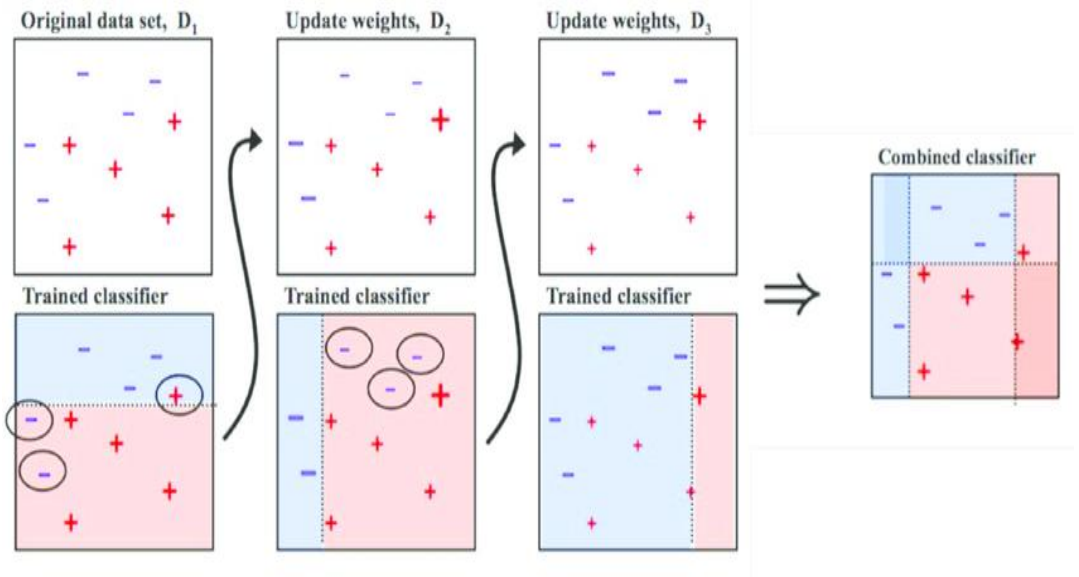
- Boosting is an ensemble learning technique that uses a set of machine learning algorithms in order to convert or combine weak learners to strong learners in order to increase the accuracy of the model.
- So boosting is very effective method in order to increase the efficiency of your model.



Step-1: The base algorithm reads the data and assigns equal weight to each sample observation.

Step-2: False predictions are assigned to the next base learner with a higher weightage on these incorrect predictions

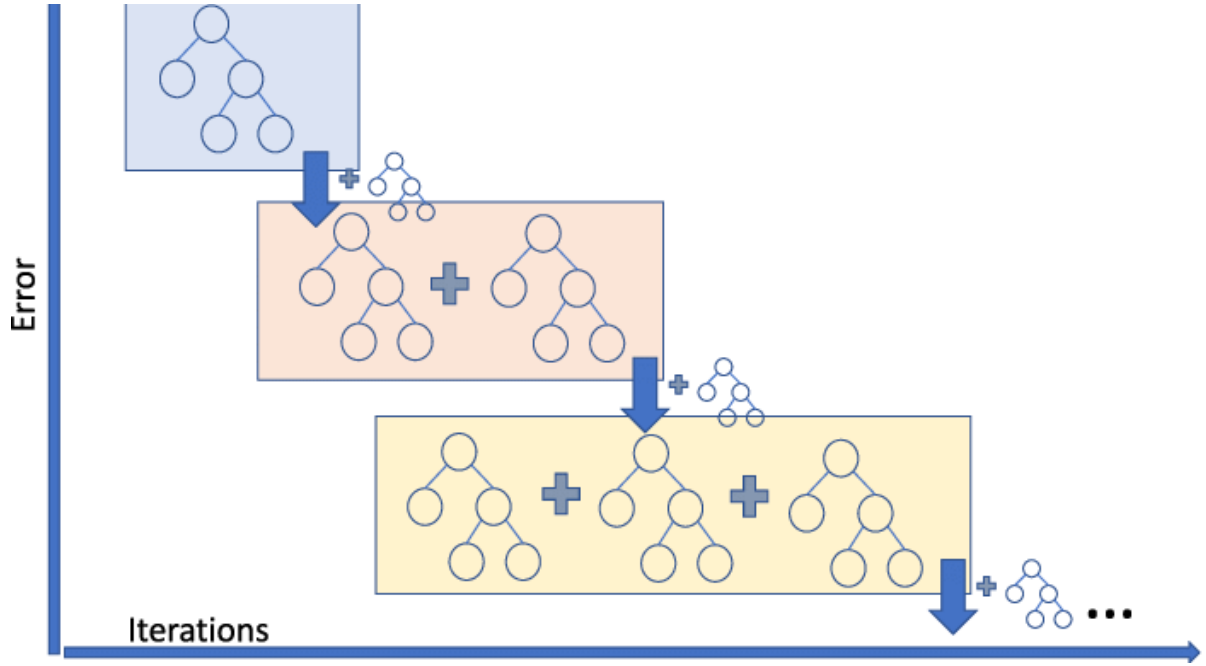
Step-3: Repeat step 2 until the algorithm can correctly classify the output.



1. Each observation is weighted equally for the first decision stump.
2. Mis-classified observations are assigned higher weights.
3. A new decision stump is drawn by considering the observations with higher weights as more significant.
4. Again if any observation are misclassified, they were given higher weight.
5. The process continues until all the observations fall into the right class.

Gradient Boosting

- In Gradient Boosting, base learners are generated sequentially in such a way that the present base learner is always more effective than the previous one.
- The key idea is to set the target outcomes for this next model in order to minimize the error.
- It has three main components:
  - Loss function that needs to be optimized.
  - Weak learner for computing predictions and forming strong learners.
  - An additive Model that will regularize the loss function.



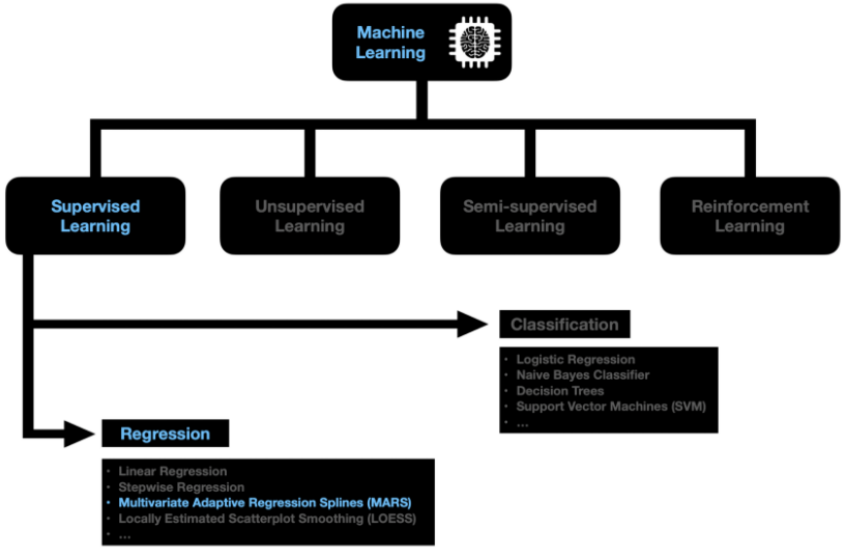
- XGBoost is an advanced version of gradient boosting method that is designed to focus on computational speed and model efficiency.
- It supports parallelization by creating decision trees parallel
- It implements distributed computing methods for evaluating any large or complex method.
- It also uses Out-of-Core computing in order to analyze huge and varied datasets.

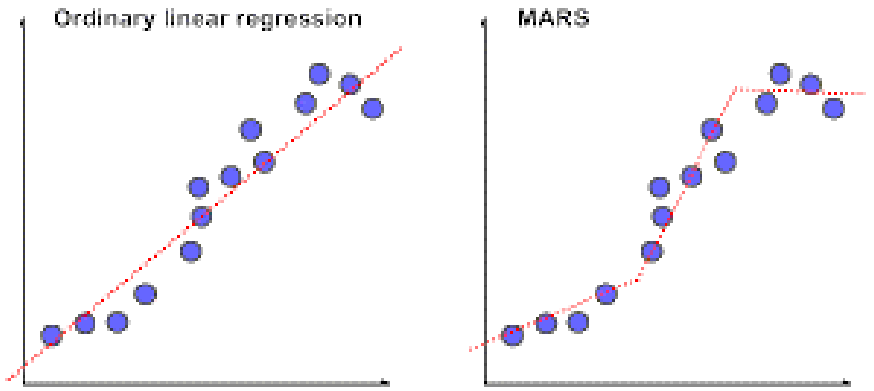
Multivariate Adaptive Regression Splines

- MARS is a type of ensemble of simple linear functions and can achieve good performance on challenging regression problems with many input variables and complex non-linear relationships.

- It is a non-parametric regression technique and can be seen as an extension of linear models that automatically models non- linearities and interactions between variables.

- The algorithm involves finding a set of simple linear functions that in aggregate result in the best predictive performance.

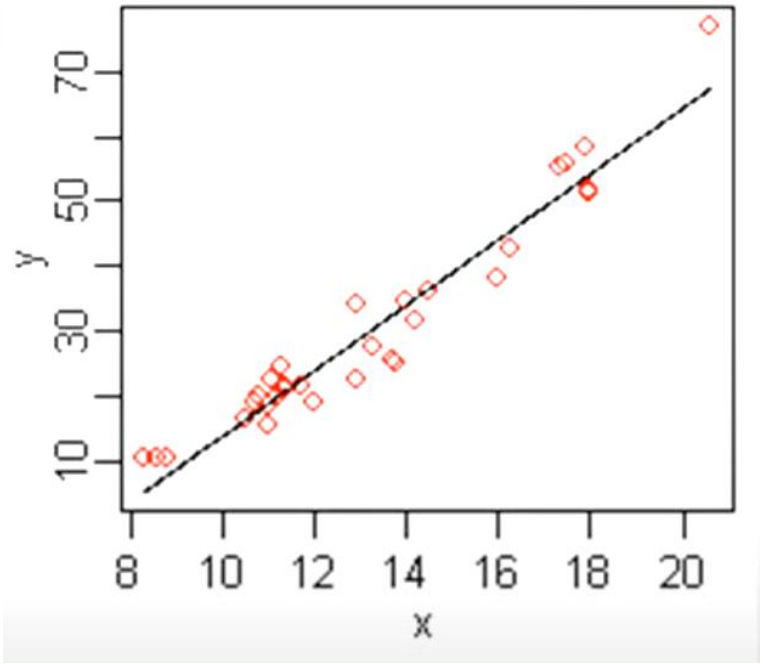


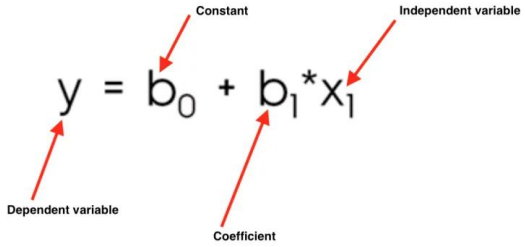


Multivariate Adaptive Regression Splines

Example: A set of data with a matrix of input variables x, and a vector of the observed responses y, with a response for each row in x. For example, the data could be: BASICS x y 10.5 16.4 10.7 18.8 10.8 19.7 ... ... 20.6 77.0

X	Y
10.5	16.4
10.7	18.8
10.8	19.7
...	...
20.6	77

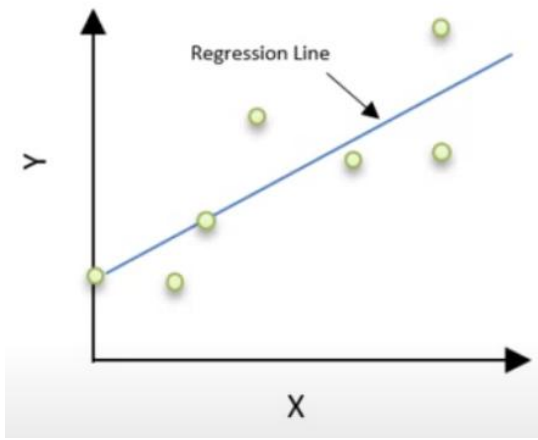


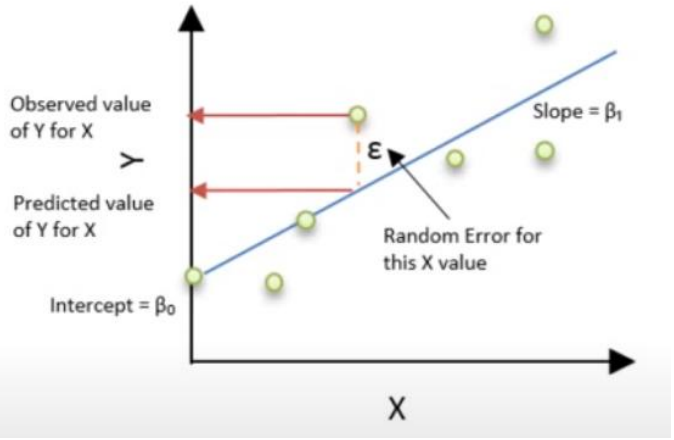


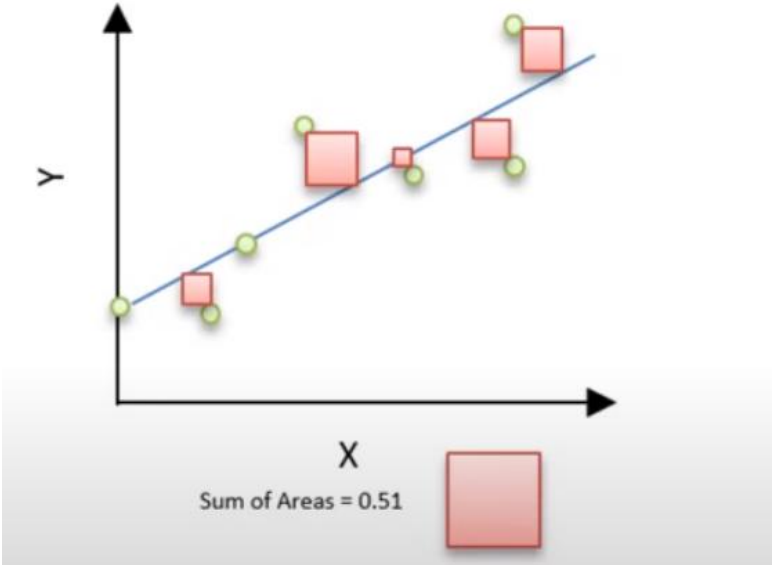
A linear model for the above data is $\tilde{y} = -37 + 5.1x$

The figure shows a plot of this function: a line giving the predicted $\tilde{y}$ versus x, with the original values of y shown as red dots.

Multivariate Adaptive Regression Splines

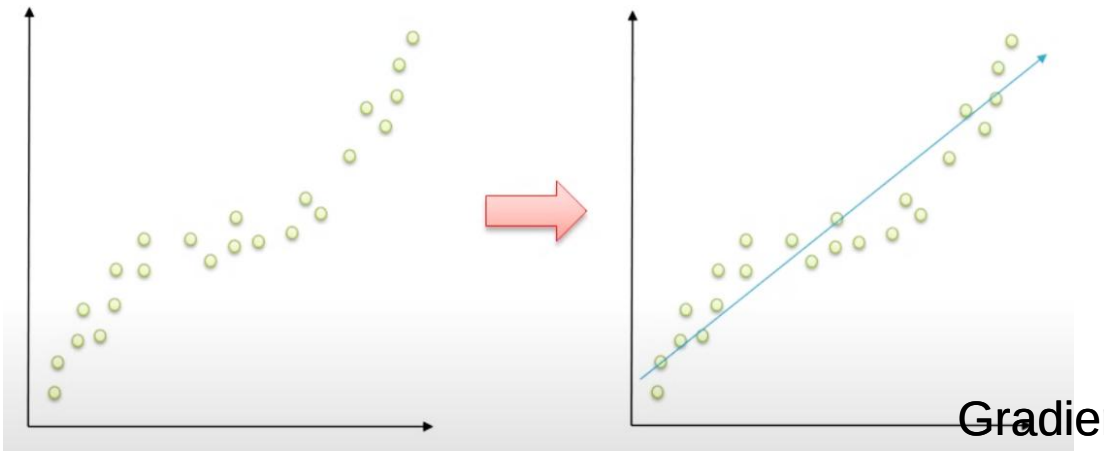




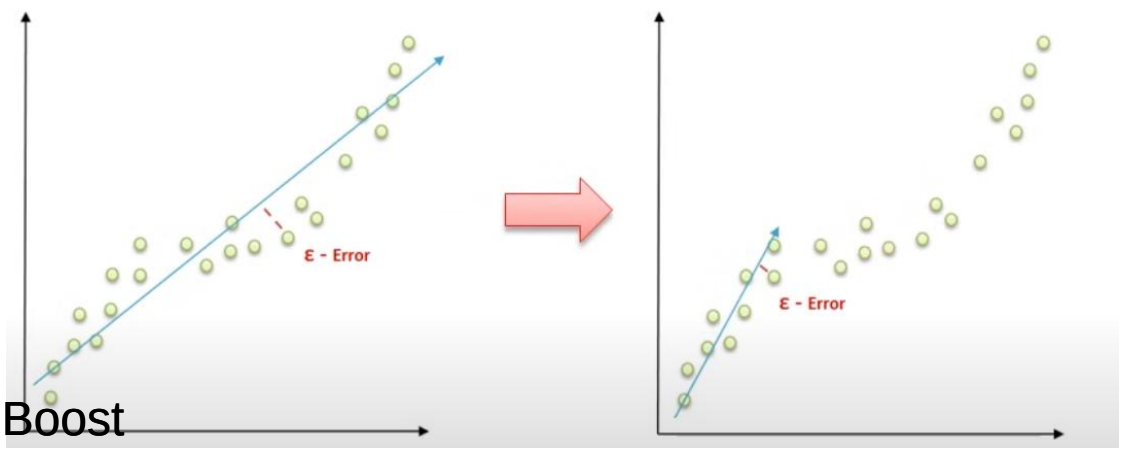


Multivariate Adaptive Regression Splines

1

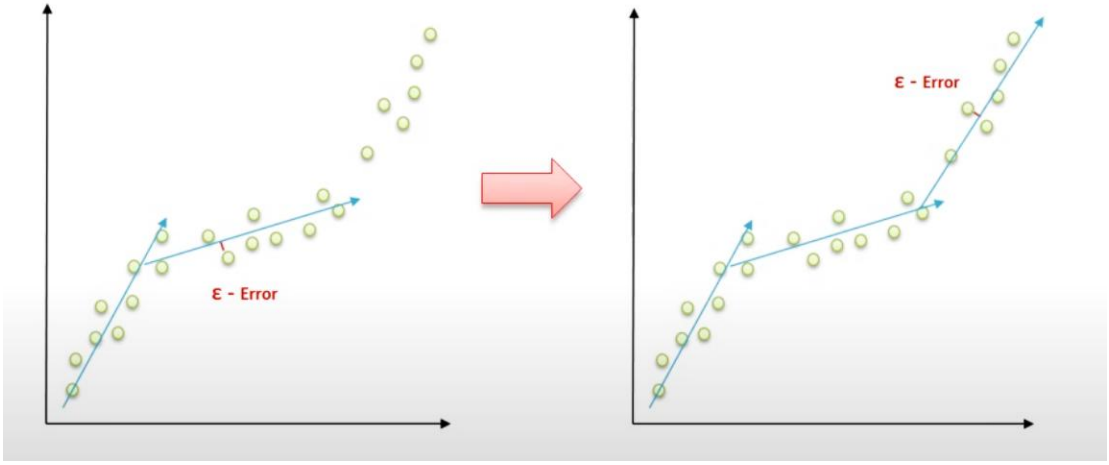


2

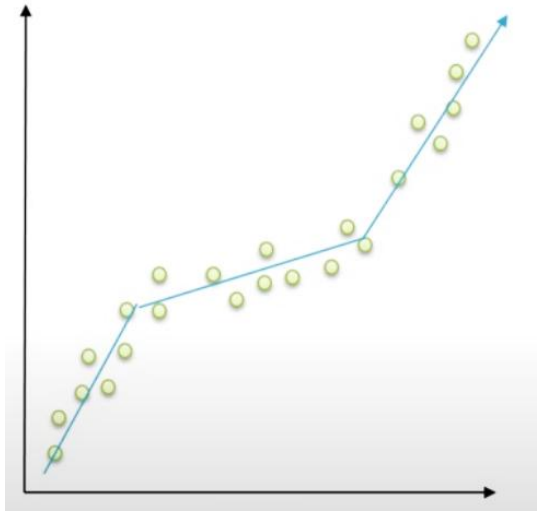


Gradient Boost

3

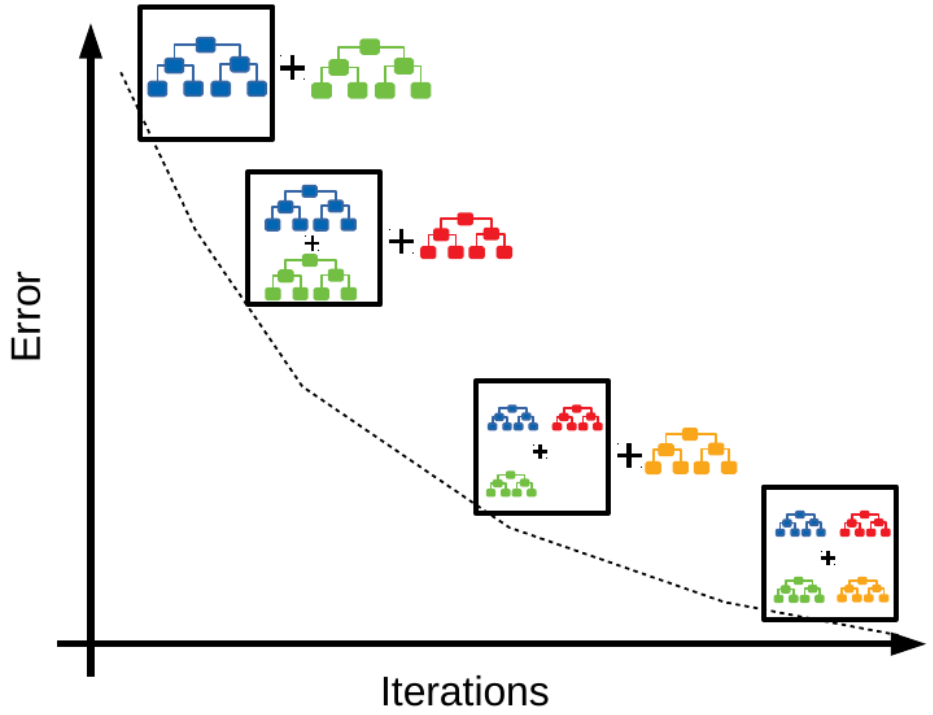


4



Gradient Boosting

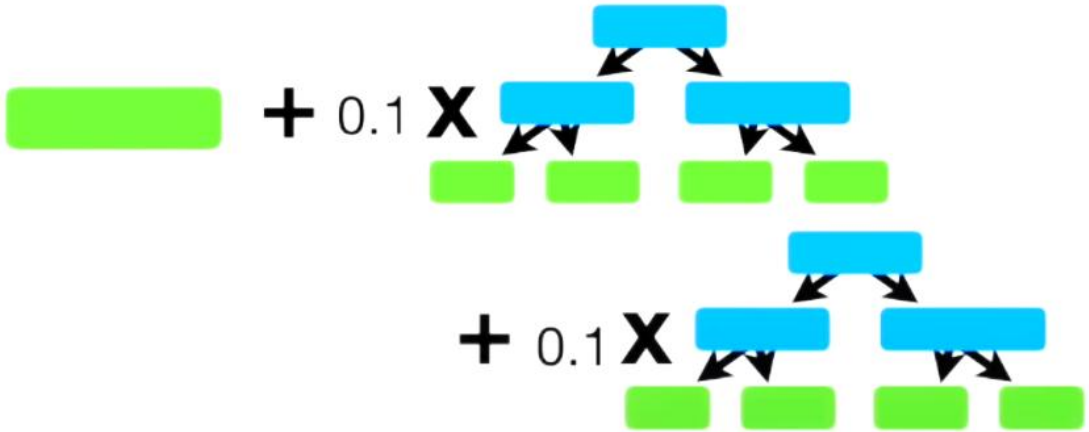
- In gradient boosting machines, or simply, GBMs, the learning procedure consecutively fits new models to provide a more accurate estimate of the response variable.
- A new tree is added, which is trained on labels that result from the errors made by the existing ensemble of trees.
- In gradient boosting, it trains many models sequentially. Each new model gradually minimizes the loss function and needs special attention as it is an error term of the whole system.
- The learning procedure consecutively fit new models to provide a more accurate estimate of the response variable.



Gradient Boosting

- Type of Problem – You have a set of variables vectors x1 , x2 and x3. You need to predict y which is a continuous variable.

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

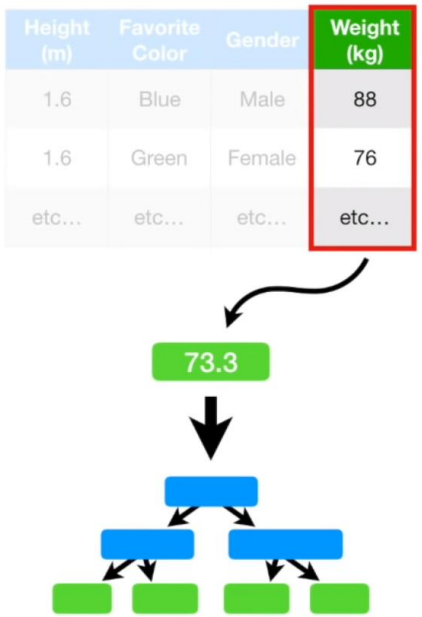


Gradient Boosting

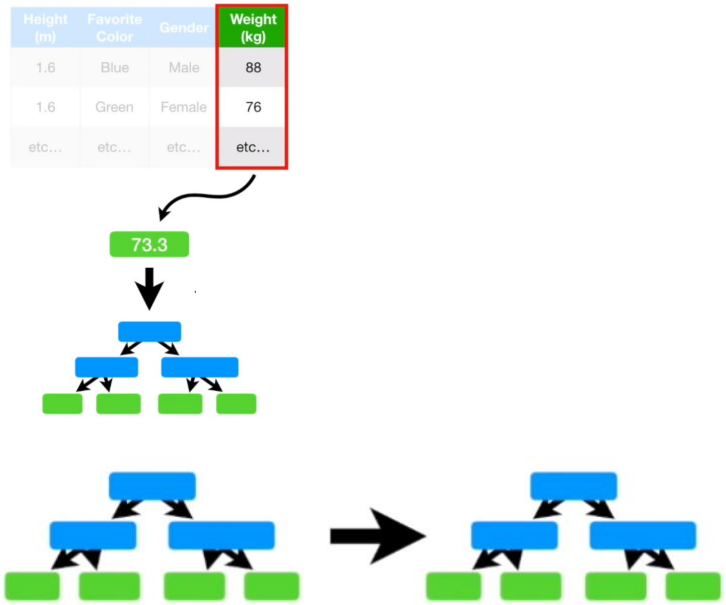
1

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
etc…	etc…	etc…	etc…

2



3



4

Average Weight

71.2

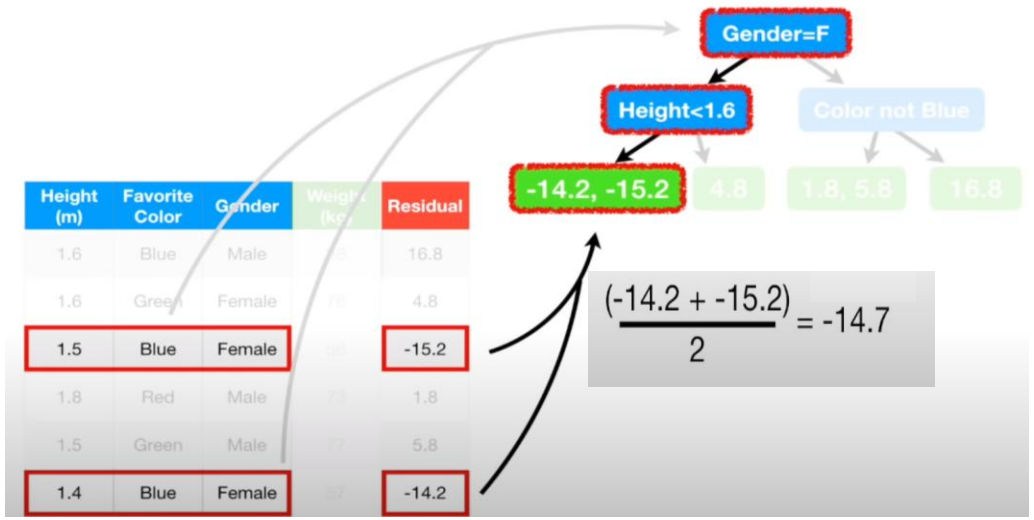
(88 - 71.2) = 16.8

Gradient Boosting

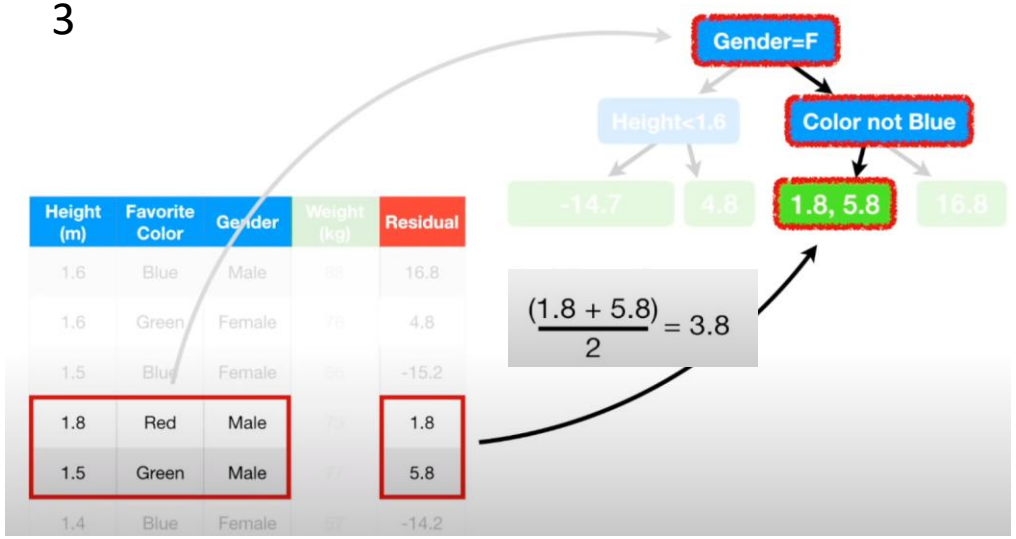
1

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

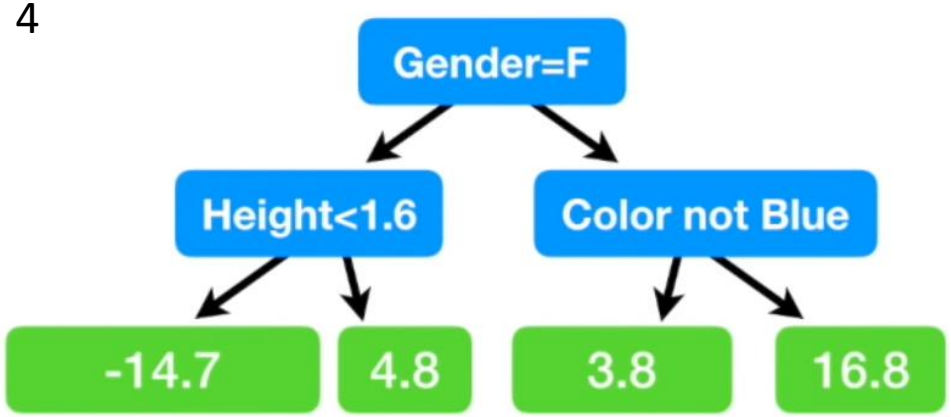
2



3



4



Gradient Boosting

5

Height (m)	Favorite Color	Gender	Weight (kg)	Residual
1.6	Blue	Male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	Male	73	1.8
1.5	Green	Male	77	5.8
1.4	Blue	Female	57	-14.2

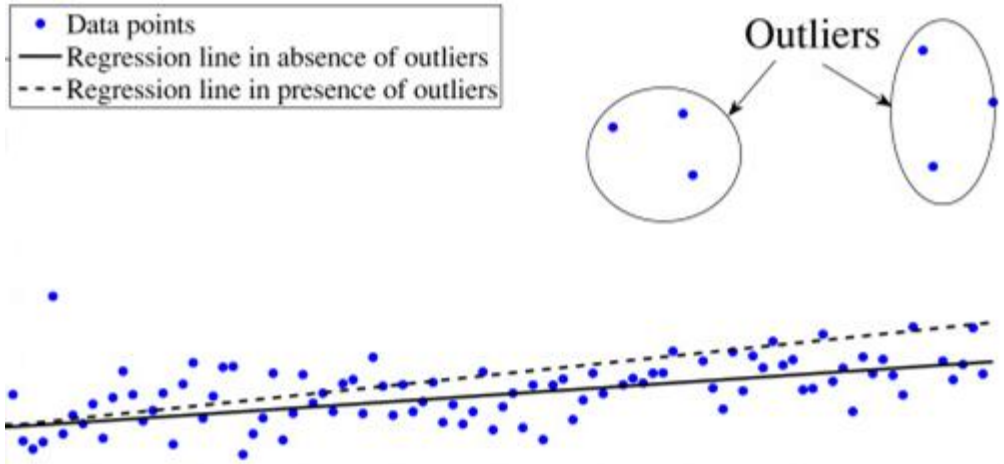
Predicted Weight = 71.2 + 16.8 = 88

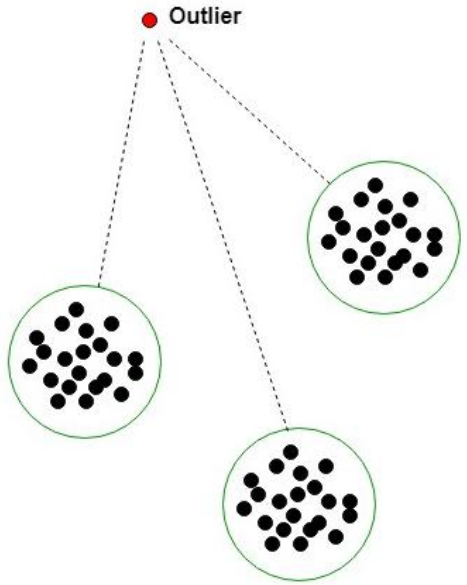
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

Methods for estimating generalization errors

- Single-sample statistics
  - Finding error on each data on generalization.
- Split-sample or hold-out validation (60-20-20)
  - Reserve some data as a "test set", which must not be used during training.
  - The test set must represent the cases that the ANN should generalize to.
  - A re-run with the random test set provides an unbiased estimate of the generalization error.
  - Disadvantage is that it reduces the amount of data available for both training and validation.
- Cross-validation (e.g., leave one out)
  - In k-fold cross-validation the data is divided into k subsets, and the network is trained k times, each time leaving one subset out for computing the error.
  - "Crossing" makes an improvement over split-sampling allowing all data used for training.
  - Disadvantage is that the network must be re-trained many times (k times in k-fold crossing).
- Bootstrapping
  - Works on random sub-samples (random shares) chosen from the full data set.
  - Any data item may be selected any number of times for validation.
  - The sub-samples are repeatedly analyzed.

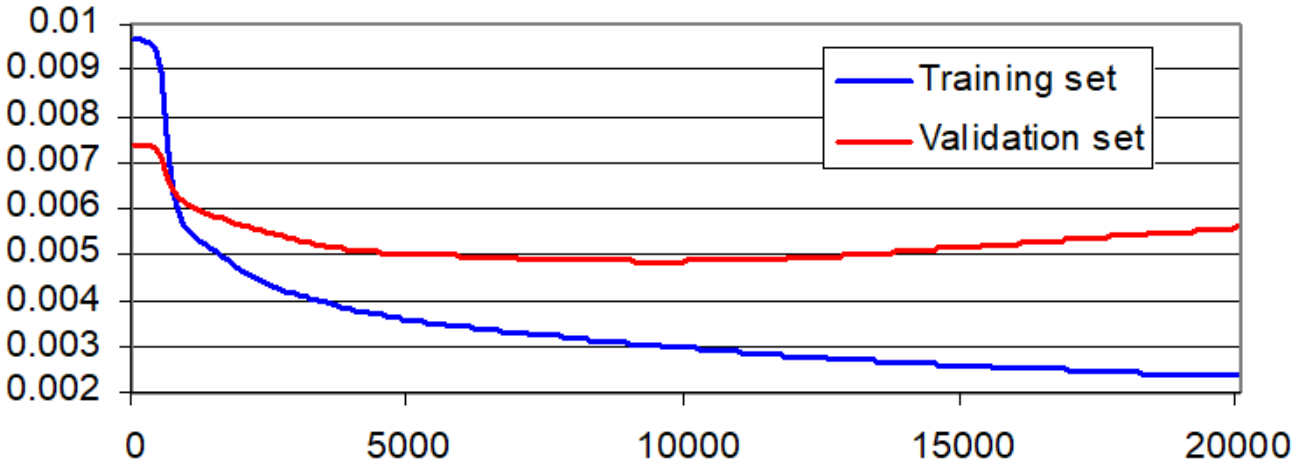
Outlier Detection





Overfitting, Under fitting

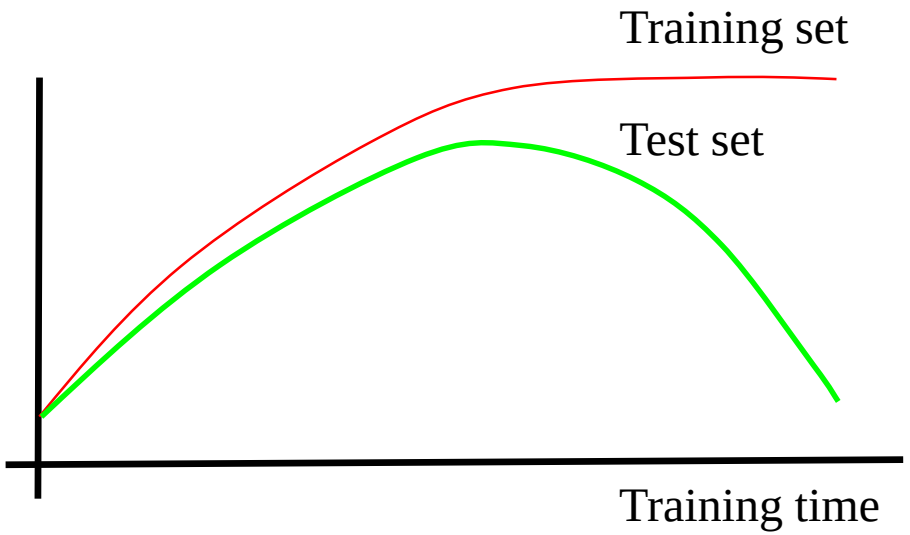
- Due to overtraining (training too much!)
- Overfittings may fit the training data well, even outliers but fail to generalize to new data.
- But loose performance on test sets (new data)



Underfitting (high bias)

Number of weight updates

Overfitting (high variance)



Overfitting, Under fitting

- A tree that classifies the training data perfectly may not lead to best generalization performance.
  - There may be noise in the training data the tree is fitting.
  - The algorithm might be making decisions based on very little data.

A hypothesis *h* is said to overfit the training data

- if there is another hypothesis, h', such that *h* has smaller error than h' on the training data but *h* has larger error on the test data than *h'*.

Avoiding Overfitting

- Two basic approaches
  - Prepruning: Stop growing the tree at some point when there is not enough data.
  - Postpruning: Grow the full tree and then remove nodes not having sufficient evidence.

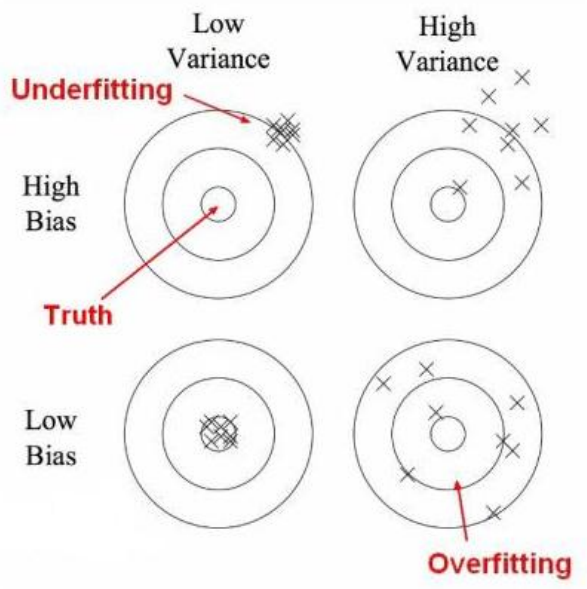
Bias vs Variance Trade Off

Bias

- Bias is how far are the predicted values from the actual values.
- A model having high bias implies that it is too simple and thus underfitting the data.

Variance

- Occurs when the model performs good on trained dataset but not on test dataset.
- A model having high variance causes overfitting the data.



Accuracy

Accuracy: The number of true predictions over total examples.

	ACTUAL CLASS		
PREDICTED CLASS		Class=Yes (Positive)	Class=No (Negative)
	Class=Yes (Positive)	a (TP)	b (FN)
	Class=No (Negative)	c (FP)	d (TN)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Accuracy is **not** a good choice with **unbalanced** classes!
- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Precision and Recall

Precision: Proportion of selected items that are correct.

- Identify only the relevant instances i.e. proportion of relevant instances that actually relevant.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall (Sensitivity): Proportion of correct items that are selected.

- Identify all the relevant instances in a dataset.

$$\text{Recall} = \frac{TP}{TP+FN}$$

Note: Often you have a trade-off between Recall and Precision.

F measure: Combined metric of P/R to find optimal blend P/R tradeoff (Weighted Harmonic Mean).

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

F1 Score: Balanced F measure with α=1/2 and β=1

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

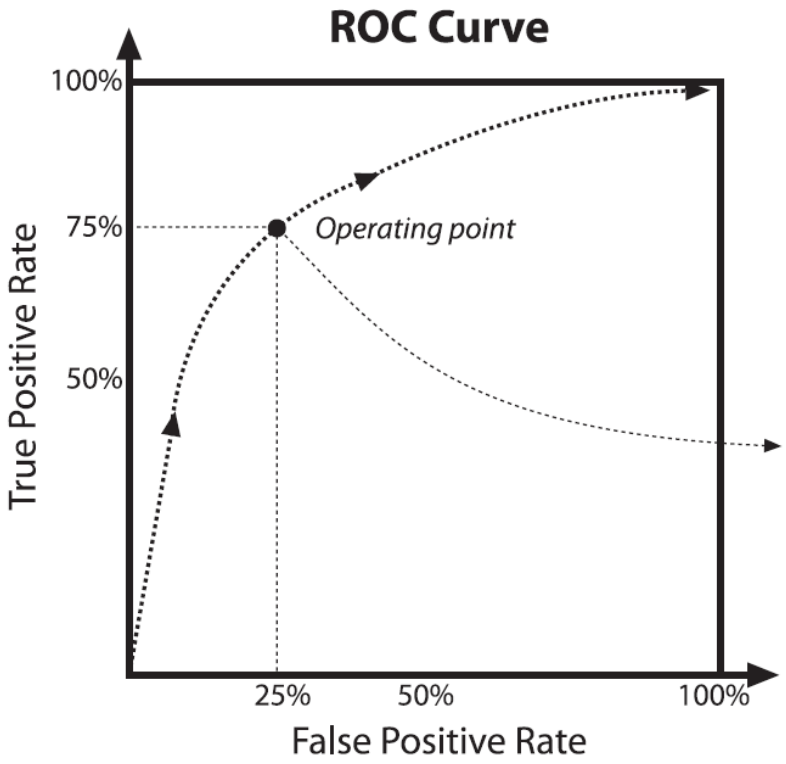
$$= \frac{2TP}{2TP+FN+FP}$$

Note: We use the harmonic mean instead of a simple average because it punishes extreme values.

Receiver Operating Characteristics

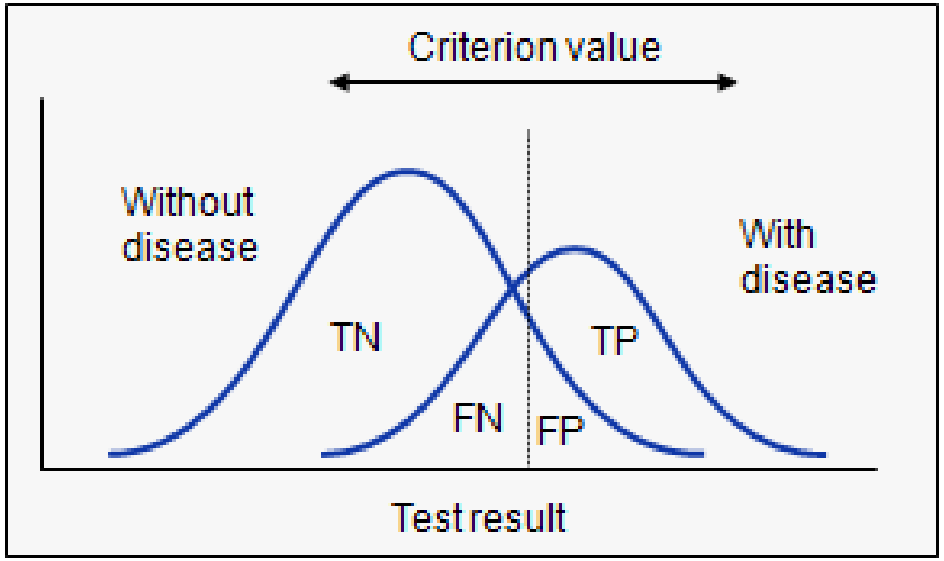
AUROC (Area Under Receiver Operating Characteristic): Combined metric on ROC space.

$$\text{Root}\left(\frac{SNS^2 + SPS^2}{2}\right)$$



Confusion Matrix

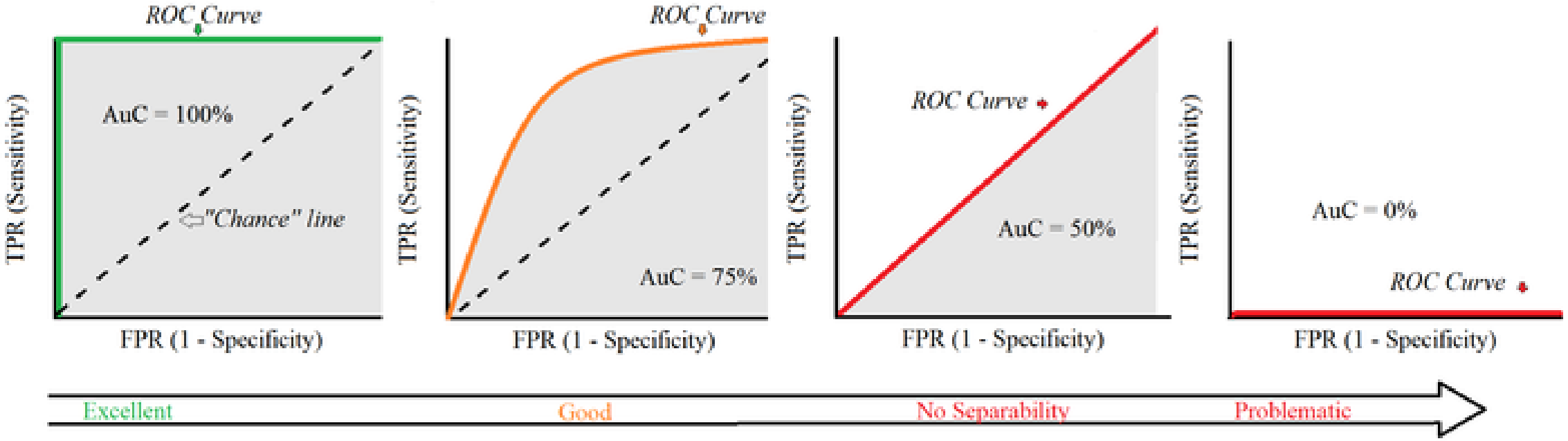
True \ Predicted	T	F
T	75%	25%
F	25%	75%

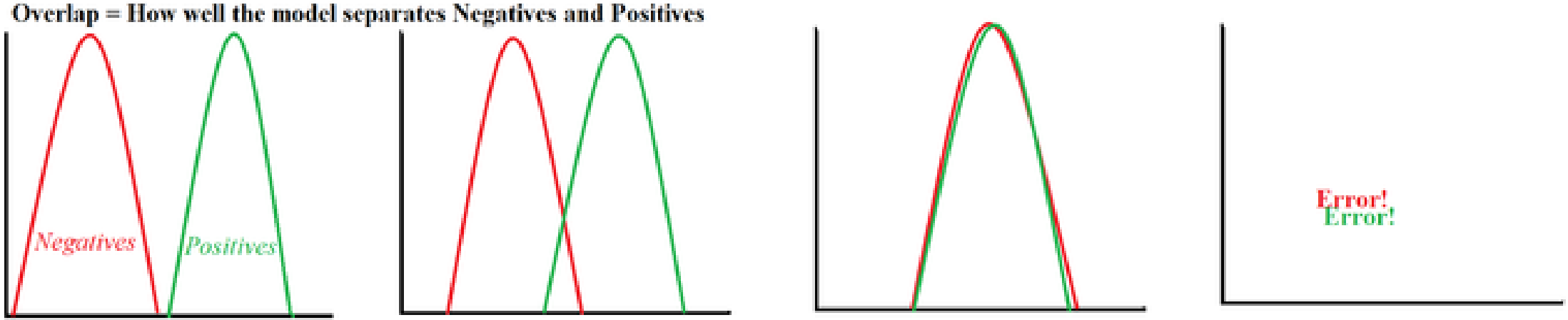


- The ROC graph plots **sensitivity on the y-axis** and **(1-specificity) on the x-axis**.
- We put a point on the graph for each threshold value of the test, plotting the sensitivity and specificity of the test at that value.
- Connecting those points creates a curve - the ROC curve.

http://www.navan.name/roc/

Receiver Operating Characteristics



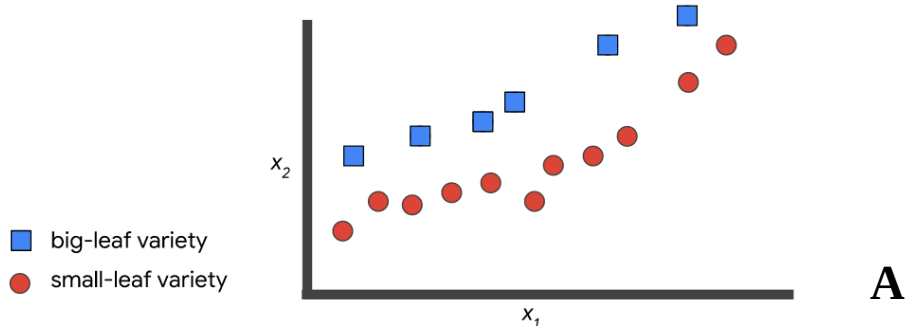


Worked Examples

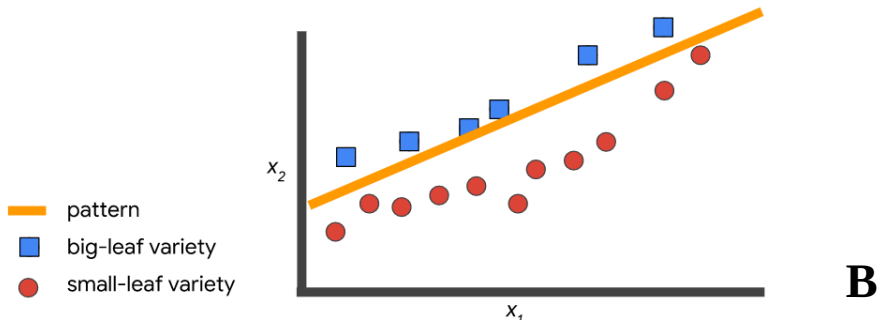
Question 1. Suppose you are an amateur botanist determined to differentiate between two species of the Lilliputian plant genus (a completely made-up plant). The two species look pretty similar. Fortunately, a botanist has put together a data set of Lilliputian plants she found in the wild along with their species name. Leaf width and leaf length are the features, while the species is the label. The goal of the data set is to help other botanists answer the question, "Which species is this plant?"

Leaf Width	Leaf Length	Species
2.7	4.9	small-leaf
3.2	5.5	big-leaf
2.9	5.1	small-leaf
3.4	6.8	big-leaf

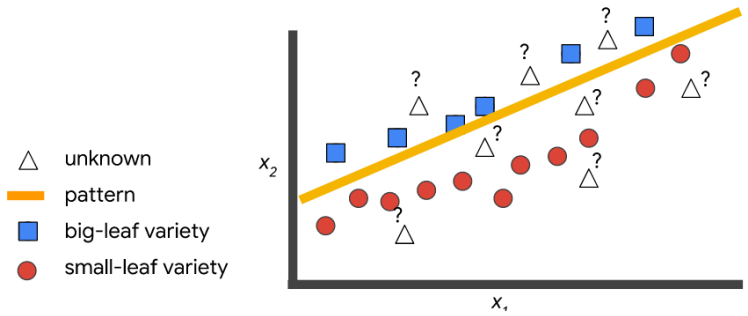
Solution:



A



B



C

Question 2. A test for a rare disease claims that it will report a positive result for 99.5% of people with the disease, and will report a negative result for 99.9% of those without the disease. We know that the disease is present in the population at 1 in 100,000. Knowing this information, what is the likelihood that an individual who tests positive will actually have the disease?

Solution:

This problem provides a simple example of how Bayes' rule can be useful. As before, we begin by establishing our notation,

P(+test | +disease) --- probability of a positive test result given that the patient has the disease. From the data this probability is 0.995.

P(-test | -disease) ---probability of a positive test result given that the patient does not have the disease. From the data this probability is 0.999.

P(+disease) --- probability that the patient has the disease given no other information. From the population average, this value is 0.00001.

P(-disease) --- probability that the patient does not have the disease given no other information. Calculated from the population average, this value is 1-0.00001 or 0.99999.

Worked Examples

What we want to find is

P(+disease | +test) → Probability that a patient has the disease and given a positive test result.
This unknown can be found directly using Bayes' rule

$$P(+disease \mid +test) = [P(+test \mid +disease) \times P(+ disease)] / P(+test)$$

To evaluate this expression, we need to know the probability of a positive test, which can be found by marginalizing over all possible disease states (+disease and disease)

$$\begin{aligned} P(+test) &= P(+test, disease) \\ &= P(+test \mid disease) \times P(disease) \\ &= P(+test \mid +disease) \times P(+disease) + P(+test \mid -disease) \times P(-disease) \\ &= P(+test \mid +disease) \times P(+disease) + (1 - P(-test \mid -disease)) \times P(-disease) \end{aligned}$$

For our given data this works out to be

$$P(+test) = (0.995) \times (0.00001) + (1 - 0.999) \times (0.99999) = 0.00100994$$

Now, all of the elements are known and can be directly calculated disease states

$$\begin{aligned} P(+disease \mid +test) &= [P(+test \mid +disease) \times P(+disease)] / P(+test) \\ &= ((0.995) \times (0.00001))/0.00100994 \\ &= 0.0099 \end{aligned}$$

Thus, if **100** people get a positive result, only one of these people will really have the disease.

Question 3. A researcher hypothesizes that it is possible to detect membrane proteins using the fraction of hydrophobic residues alone. To test this model, the researcher creates a library of 7500 proteins and scores each of these proteins based on their fraction of hydrophobic residues and whether they are membrane proteins. The results of this analysis are shown below. Given this information, we wish to calculate the likelihood that a novel protein that is primarily hydrophobic is also a membrane protein.

	Majority hydrophobic	Majority hydrophilic
Membrane Bound	2911	961
Cytosolic	713	2915

Solution:

To solve this problem, we will first summarize our data as a table of all of the possible combinations of possibilities.

	H	NOT H
M	P(H, M)	P(NOT H, M)
NOT M	P(H, NOT M)	P(NOT H, NOT M)

In this table, H identifies hydrophobic proteins and M membrane proteins.

Next we also include the sums of the probabilities in the margins to calculate the marginal probabilities

	H	NOT H	Sum
M	P(H, M)	P(NOT H, M)	P (M)
NOT M	P(H, NOT M)	P(NOT H, NOT M)	P (NOT M)
Sum	P(H)	P(NOT H)	1

The values in this table can be filled directly from the given data. For example,
P(H,M) = 2911/7500 = 0.388

Note that the sum in the lower right comer must equal one, for it is the sum of all possible outcomes of each variable. Filling in the remaining values, we calculate the following probabilities

	H	NOT H	Sum
M	0.388	0.128	0.516
NOT M	0.095	0.389	0.484
Sum	0.483	0.517	1

By rearranging our definition of conditional probability, we can now answer the question of the likelihood of a novel protein being membrane bound given that it is hydrophobic:

P(M | H)=P(H, M) /P(H)= 0.388 /0.482=0.803

Question 4. Given the following data of transactions at a shop, calculate the support and confidence values of milk → bananas, bananas → milk, milk → chocolate, and chocolate → milk.

Transaction	Items in basket
1	milk, bananas, chocolate
2	milk, chocolate
3	milk, bananas
4	chocolate
5	chocolate
6	milk, chocolate

Solution:

milk → bananas : Support = 2/6, Confidence = 2/4
bananas → milk : Support = 2/6, Confidence = 2/2
milk → chocolate : Support = 3/6, Confidence = 3/4
chocolate → milk : Support = 3/6, Confidence = 3/5

$$Support\ (S) = \frac{freq(x,y)}{N}$$

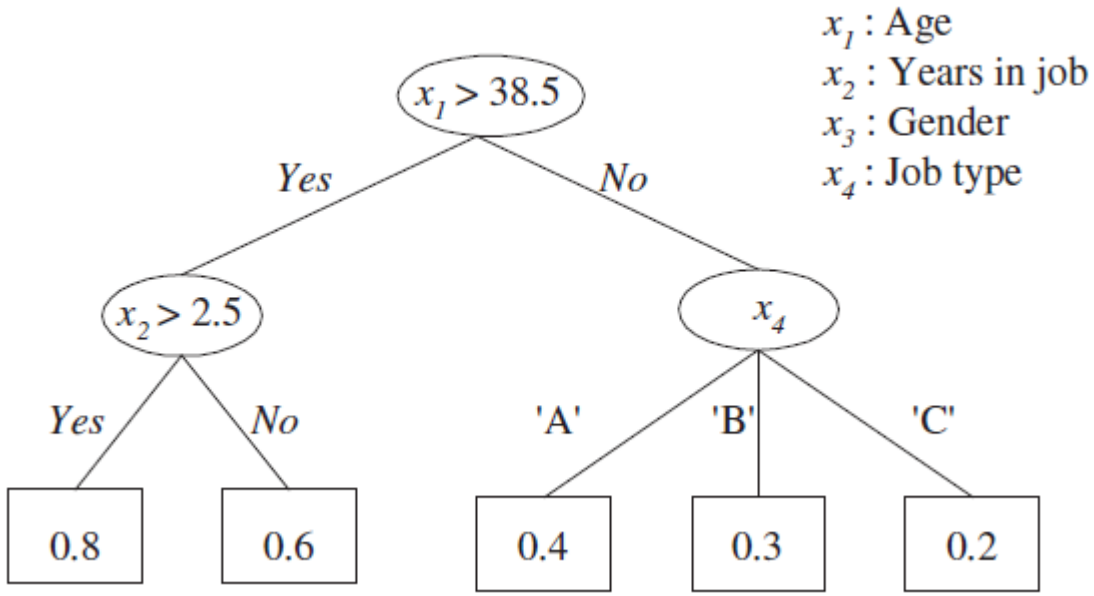
$$Confidence\ (C) = \frac{freq(x,y)}{freq(x)}$$

Though only half of the people who buy milk buy bananas too, anyone who buys bananas also buys milk.

Question 5. Draw a decision tree with the given rules:

R1: IF (age > 38.5) AND (years-in-job > 2.5) THEN $y = 0.8$
R2: IF (age > 38.5) AND (years-in-job ≤ 2.5) THEN $y = 0.6$
R3: IF (age ≤ 38.5) AND (job-type = 'A') THEN $y = 0.4$
R4: IF (age ≤ 38.5) AND (job-type = 'B') THEN $y = 0.3$
R5: IF (age ≤ 38.5) AND (job-type = 'C') THEN $y = 0.2$

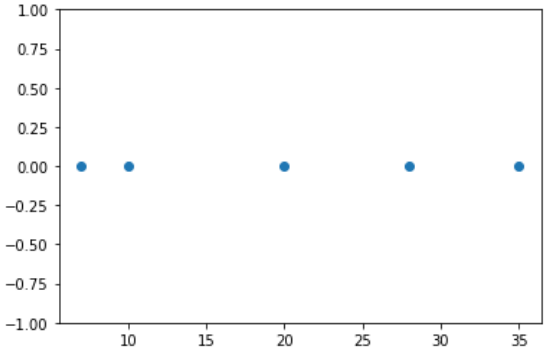
Solution:

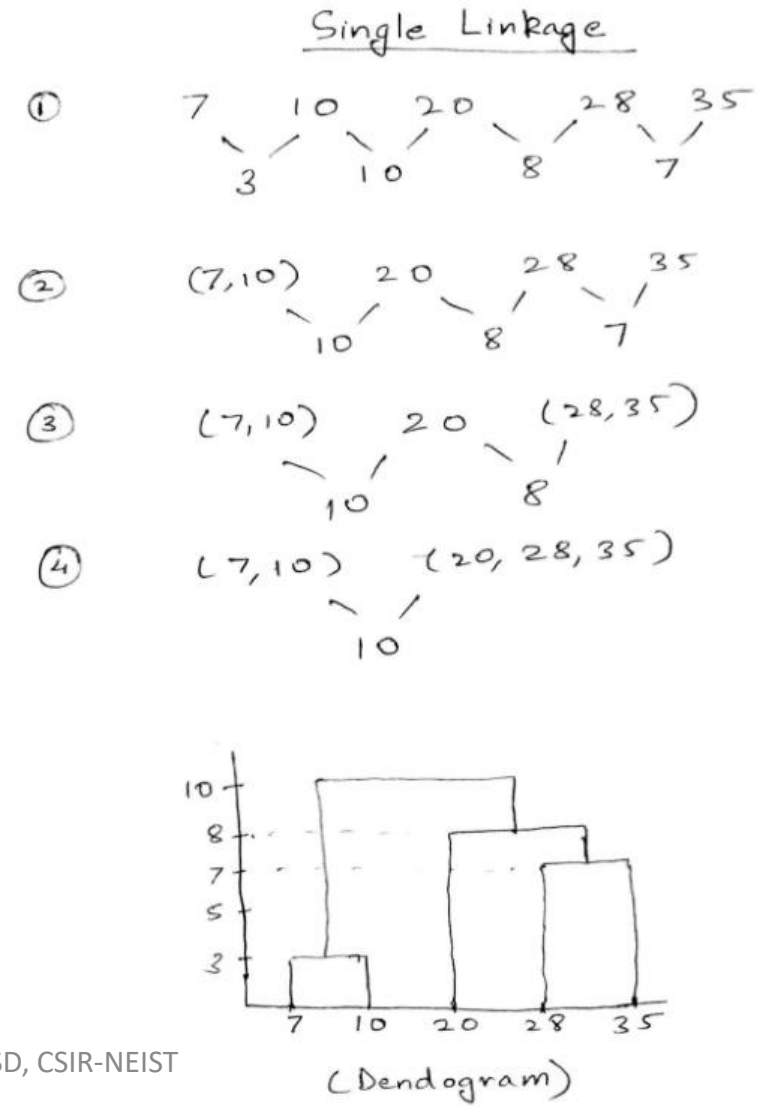


Worked Examples

Question 6. For the one dimensional data set **{7, 10, 20, 28, 35}**, perform hierarchical clustering and plot the dendogram to visualize it using single linkage (*the distance between two clusters is defined as the shortest distance between two points in each cluster*).

Solution:

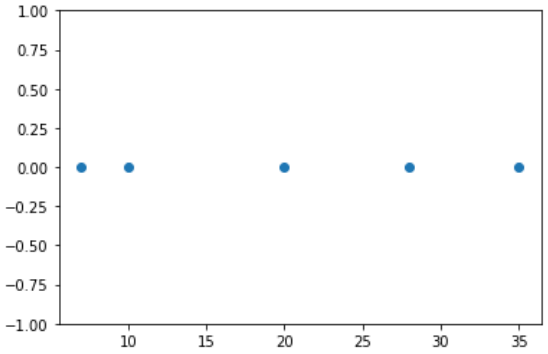


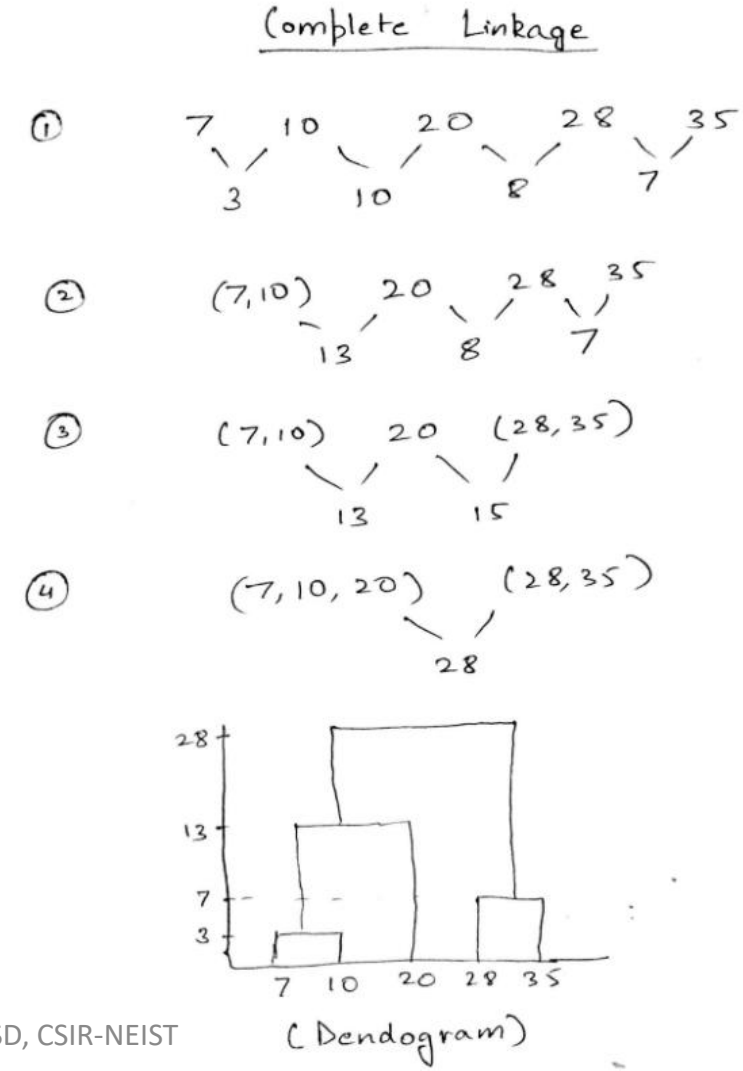


Worked Examples

Question 7. For the one dimensional data set **{7, 10, 20, 28, 35}**, perform hierarchical clustering and plot the dendogram to visualize it using complete linkage (*the distance between two clusters is defined as the largest distance between two points in each cluster*).

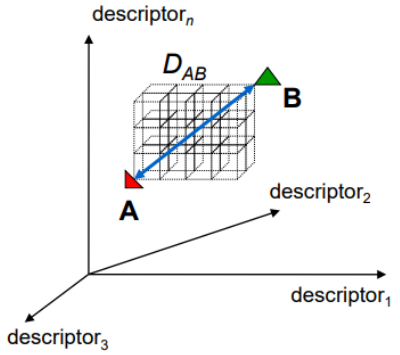
Solution:





Worked Examples

Question 8. Consider a n dimensional space of descriptors as shown in the figure. Find the distance between A = (3, 0, 1) and B = (5, 2, 0)



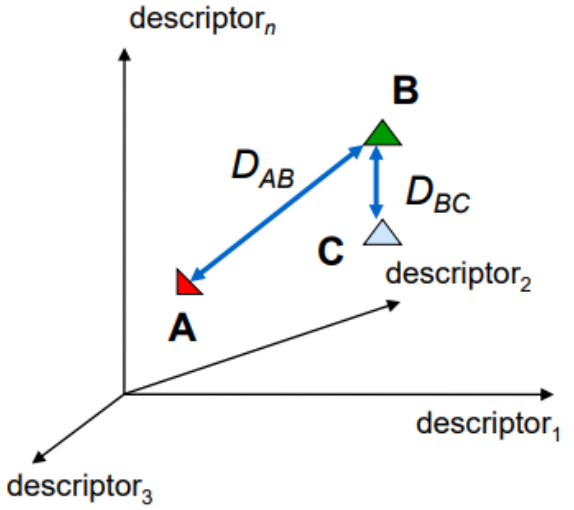
Solution: Given two n-dimensional vectors, A and B –

A = (a_1, a_2,... a_n) - B = (b_1, b_2,...b_n), the Euclidean distance is given by

$$D_{AB} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

A = (3,0,1); **B** = (5,2,0)

$$D_{AB} = \sqrt{(3-5)^2 + (0-2)^2 + (1-0)^2} = 3$$



Question 9. Four molecules in four-dimensional space - descriptor values listed in the table. Predict the most similar molecules based on the descriptors.

Solution: Using Euclidean distance,

$$D_{AB} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

mol	logP(o/w)	b_rotN	a_acc	a_don
1	2.4250	2	3	2
2	3.4700	2	3	3
3	0.7090	0	2	1
4	2.4900	1	2	1

- logP(o/w): index of hydrophobicity
- b_rotN: number of rotatable bonds
- a_acc: number of hydrogen acceptors
- a_don: number of hydrogen donors

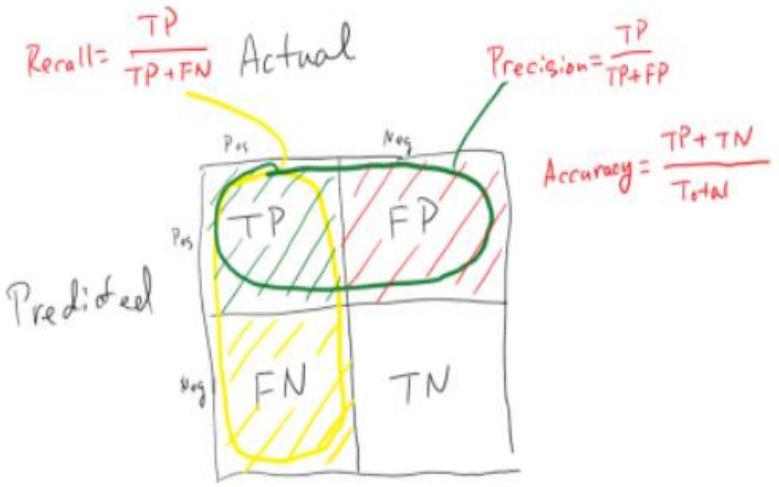
- $D_{12} = [(2.4250 - 3.4700)^2 + (2 - 2)^2 + (3 - 3)^2 + (2 - 3)^2]^{1/2} = 1.45$
- $D_{13} = [(2.4250 - 0.7090)^2 + (2 - 0)^2 + (3 - 2)^2 + (2 - 1)^2]^{1/2} = 2.99$
- $D_{14} = [(2.4250 - 2.4900)^2 + (2 - 1)^2 + (3 - 2)^2 + (2 - 1)^2]^{1/2} = 1.73$
- $D_{34} = [(0.7090 - 2.4900)^2 + (0 - 1)^2 + (2 - 2)^2 + (1 - 1)^2]^{1/2} = 2.04$

Question 9. A model has been developed for prediction of *y* values. From the given data set for actual value and predicted value of variable *y*, build the confusion matrix and find the accuracy, precision and recall of the model.

y	y pred	output for threshold 0.6
0	0.5	0
1	0.9	1
0	0.7	1
1	0.7	1
1	0.3	0
0	0.4	0
1	0.5	0

Solution:

y	y pred	output for threshold 0.6	Recall	Precision	Accuracy
0	0.5	0	1/2	2/3	4/7
1	0.9	1			
0	0.7	1			
1	0.7	1			
1	0.3	0			
0	0.4	0			
1	0.5	0			



Question 10. The confusion matrix for a spam detection model is provided below. Compute the sensitivity and specificity of the model of the model in making the predictions. Also find the precision and recall of the model.

	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal 🕵)
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉✉ 320	FALSE NEGATIVE (FN) ✉🕵 43
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉✉ 20	TRUE NEGATIVE (TN) ✉🕵 538

Solution:

Sensitivity measures how apt the model is to detecting events in the positive class
Specificity measures how exact the assignment to the positive class is.

	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal 🕵)	
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉✉ 320	FALSE NEGATIVE (FN) ✉🕵 43	$Sensitivity = \frac{TP}{TP+FN} = \frac{320}{320+43} = 0.882$
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉✉ 20	TRUE NEGATIVE (TN) ✉🕵 538	$Specificity = \frac{TN}{FP+TN} = \frac{538}{20+538} = 0.964$

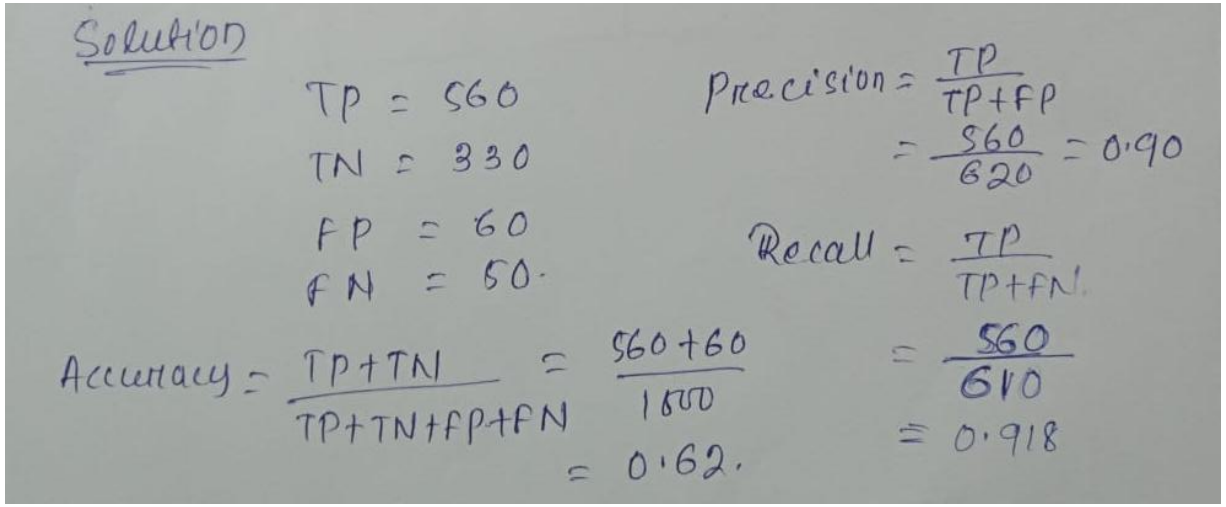
	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal 🕵)	
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉✉ 320	FALSE NEGATIVE (FN) ✉🕵 43	$Recall = \frac{TP}{TP+FN} = \frac{320}{320+43} = 0.882$
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉✉ 20	TRUE NEGATIVE (TN) ✉🕵 538	
	$Precision = \frac{TP}{TP+FP} = \frac{320}{320+20} = 0.941$		

Exercise

Question 1. You want to predict how many people are infected with a contagious virus in times before they show the symptoms, and isolate them from the healthy population. The two values for our target variable would be: Sick and Not Sick. There are 100 observations for which the confusion matrix is given below. Compute the accuracy, precision and recall of the model.

PREDICTED VALUES \ ACTUAL VALUES	POSITIVE	NEGATIVE
POSITIVE	560	60
NEGATIVE	50	330

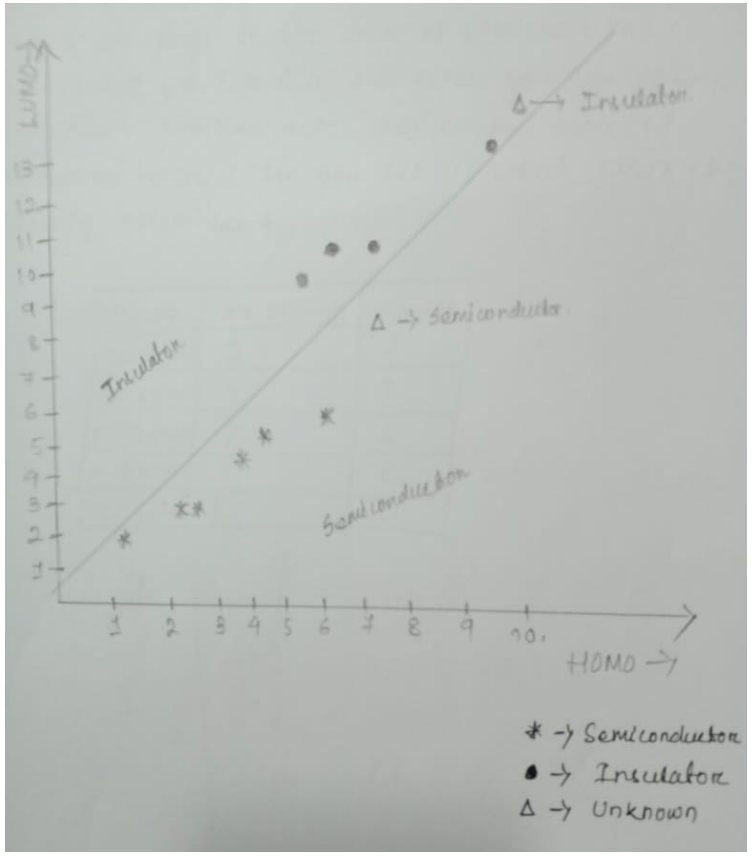
ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN



Question 2: Suppose we have data for HOMO & LUMO for several semiconductors & insulators (just for demonstration purposes) and later you are given a material's HOMO-LUMO data, then on the basis of regression model; can you tell to which class it belongs?

HOMO	LUMO	Property
2.3	2.9	Semiconductor
5.4	9.7	Insulator
6.0	5.6	Semiconductor
1.2	1.8	Semiconductor
3.6	4.2	Semiconductor
5.7	10.0	Insulator
4.5	5.1	Semiconductor
8.7	13.0	Insulator
2.1	2.7	Semiconductor
6.7	11.0	Insulator

Solution:

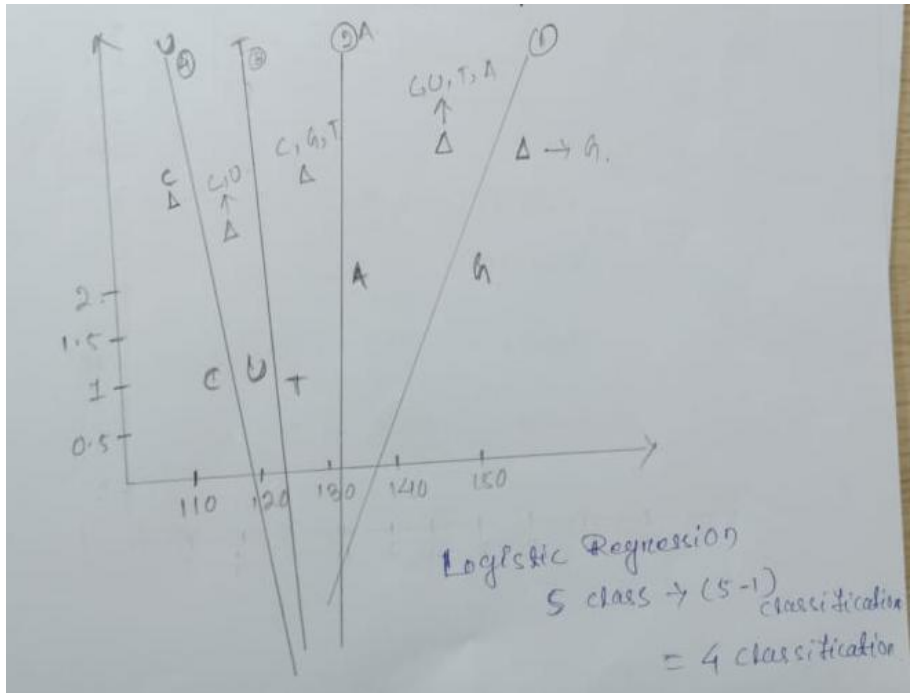


Exercise

Question 3: Suppose we have data for molecular weight & number of rings in the chemical structure for nucleobases (A, T, G, C, U) and later you are given a structure's similar data, then on the basis of regression model; can you tell to which class it closely resembles?

Mol. Wt.	No. of Rings	Species
135.13 g/mol	2	A
126.11 g/mol	1	T
151.13 g/mol	2	G
111.10 g/mol	1	C
112.08 g/mol	1	U

Solution:



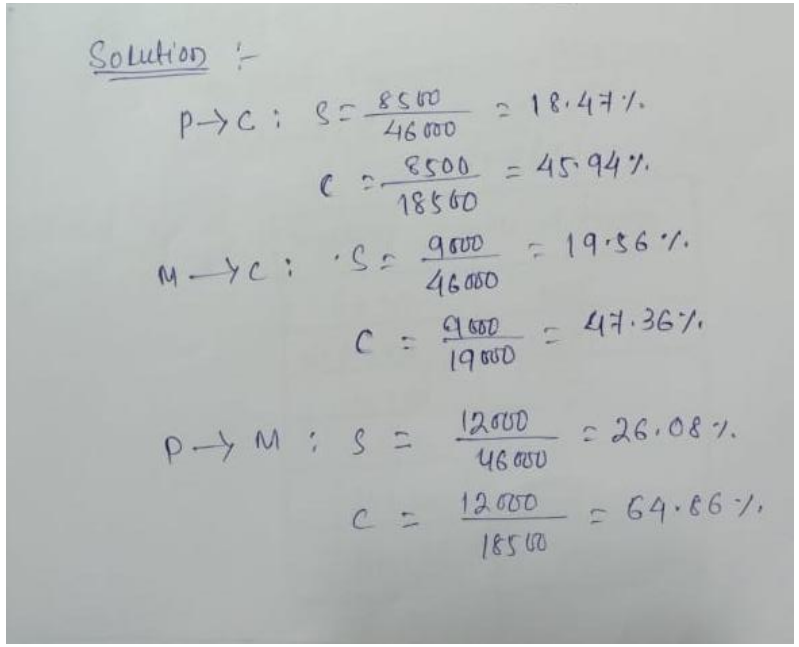
Exercise

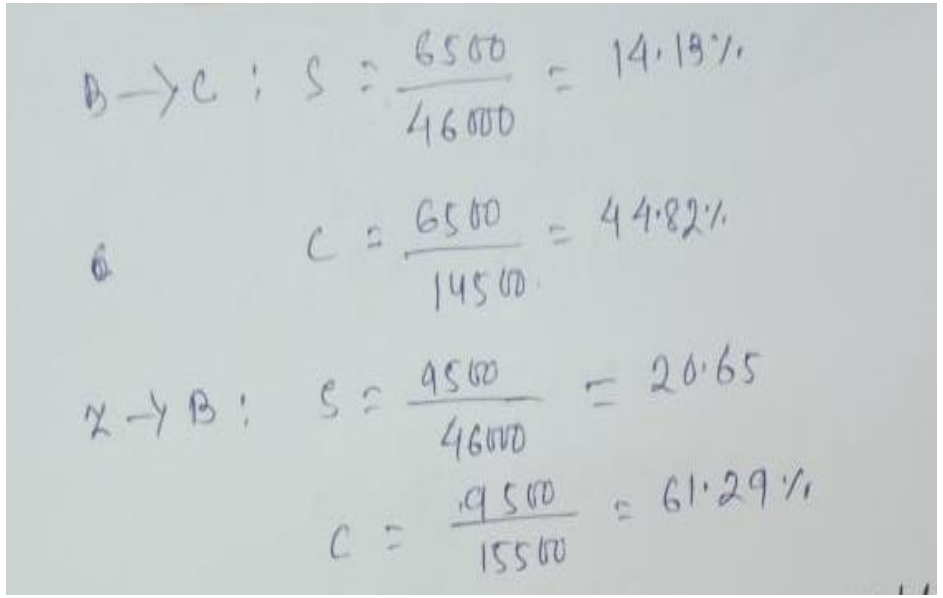
Question 4: The following data for number of students opting for different subject combinations in their final year of graduation; calculate support & confidence values of: P→C; P→M; M→C; B→C; Z→B.

Number ($x10^3$)	Combination
2	PCM
1.5	ZBC
10	PM
7	MC
6.5	PC
8	ZB
6	ZC
5	BC

Where, P=Physics, C=Chemistry, M=Mathematics, B=Botany, Z=Zoology

Given that, Support(S) = $\frac{freq(x,y)}{N}$ & Confidence(C) = $\frac{freq(x,y)}{freq(x)}$





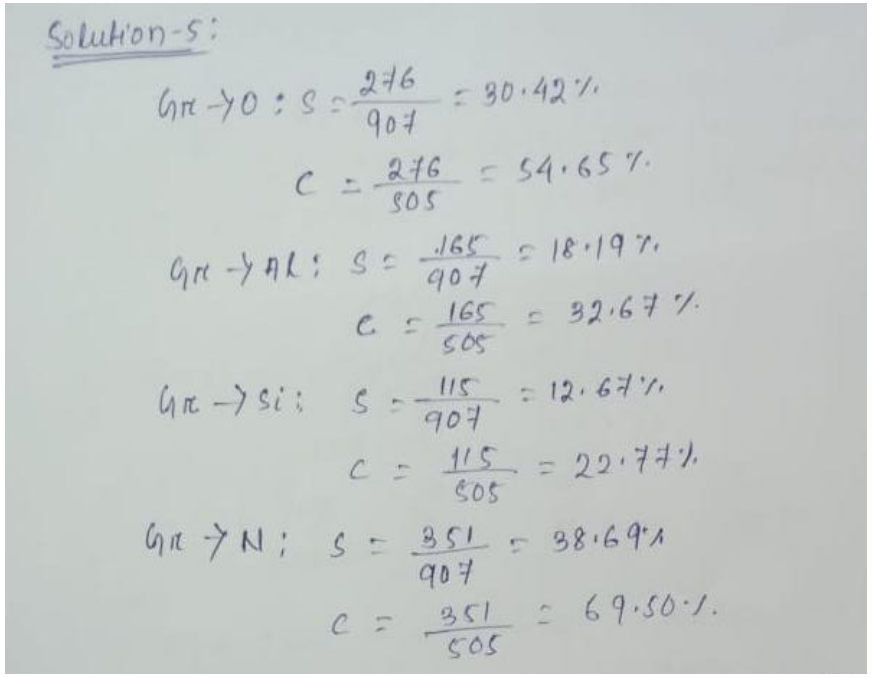
Exercise

Question 5 : The following data for number of papers published on interaction studies with Graphene with different dopants; calculate support & confidence values of: Gr→O; Gr→Al; Gr→B; Gr→ Si; Gr→N.

Where, Gr=Graphene; O=oxygen; B=boron; Al=Aluminum; Si=Silicon; Ge=Germanium; N=Nitrogen

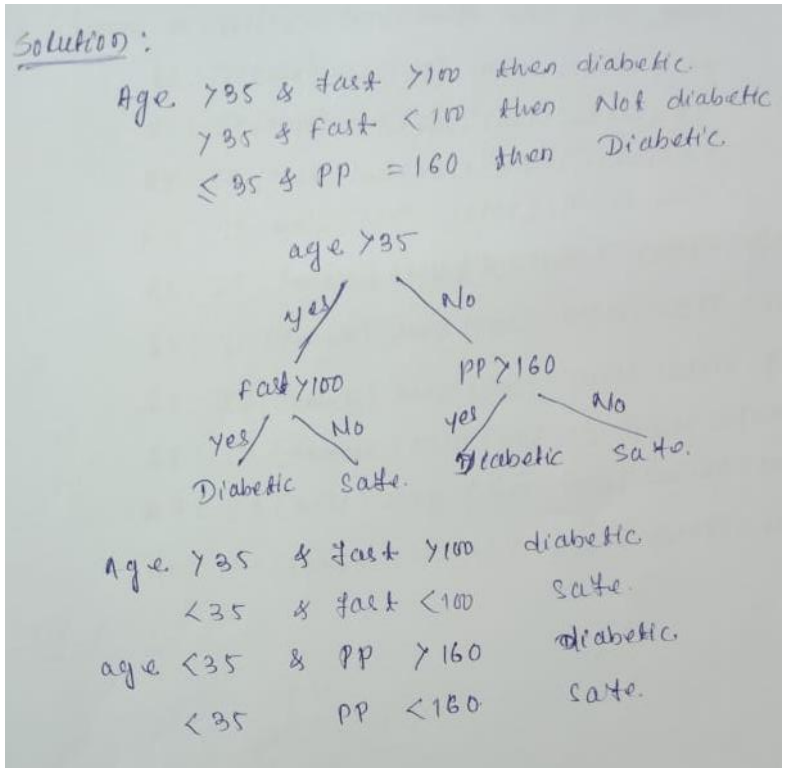
Given that, Support(S) = $\frac{freq(x,y)}{N}$ & Confidence(C) = $\frac{freq(x,y)}{freq(x)}$

System	No. of Papers
Gr+O	125
Gr+B	251
Gr+Al	165
Gr+Si	115
Gr+Ge	151
Gr+N	100



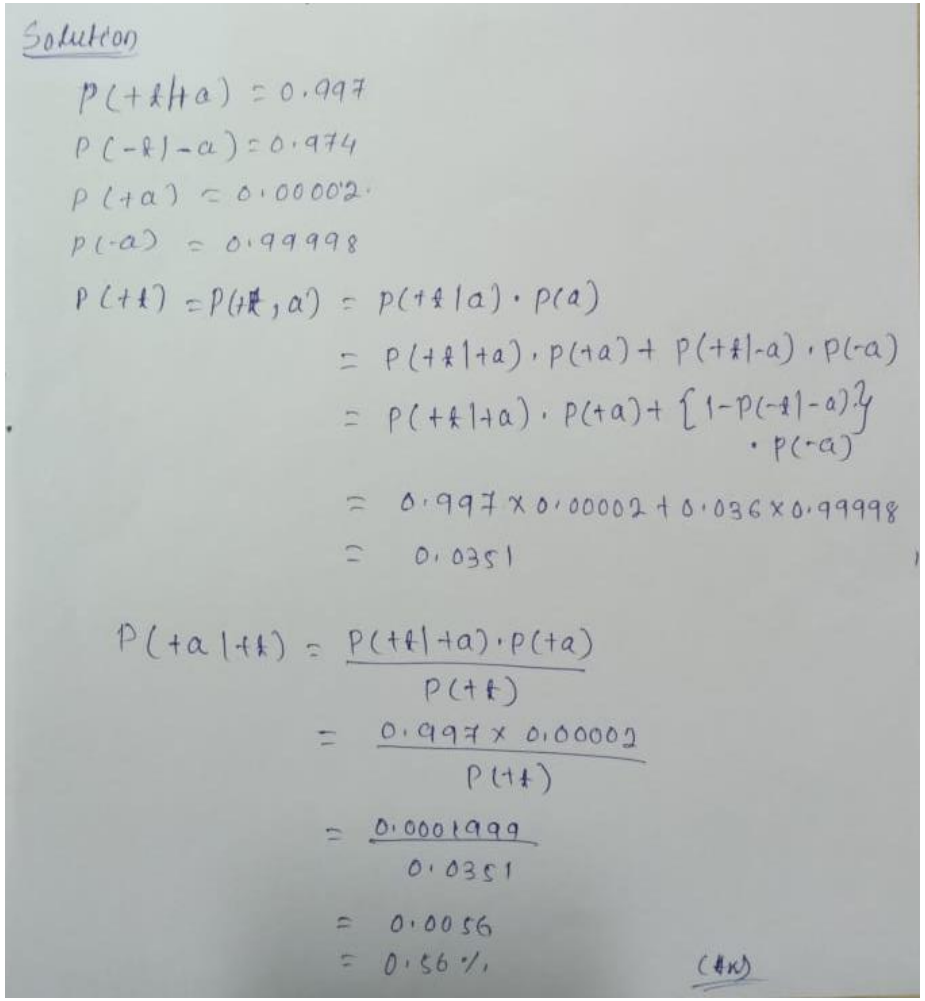
Question 6 : Draw a decision tree with the given data:

R1: IF(age≥25) AND (fasting≥110mg/dL) THEN y=diabetic
R2: IF(age≥25) AND (PP≥200mg/dL) THEN y=diabetic
R3: IF(age≥ 35) AND (fasting≥100mg/dL) THEN y=diabetic
R4: IF(age≥ 35) AND (PP≥160mg/dL) THEN y=diabetic
R5: IF(age≥ 35) AND (RBS≥160~200mg/dL) THEN y=diabetic



Exercise

Question 7: A ML model claims that it can positively report accurate predictions for 99.7%and 97.4% negatively. We know that 2 out of 10000 predictions made by the model are not accurate. What is the likelihood that an input file will achieve convergence using the new software?



Solution

$P(+ve|+a) = 0.997$

$P(-ve|-a) = 0.974$

$P(+a) = 0.00002$

$P(-a) = 0.99998$

$P(+ve) = P(+ve, a) = P(+ve|a) \cdot P(a)$

$= P(+ve|+a) \cdot P(+a) + P(+ve|-a) \cdot P(-a)$

$= P(+ve|+a) \cdot P(+a) + \{1 - P(-ve|-a)\} \cdot P(-a)$

$= 0.997 \times 0.00002 + 0.036 \times 0.99998$

$= 0.0351$

$P(+a|+ve) = \frac{P(+ve|+a) \cdot P(+a)}{P(+ve)}$

$= \frac{0.997 \times 0.00002}{P(+ve)}$

$= \frac{0.00001994}{0.0351}$

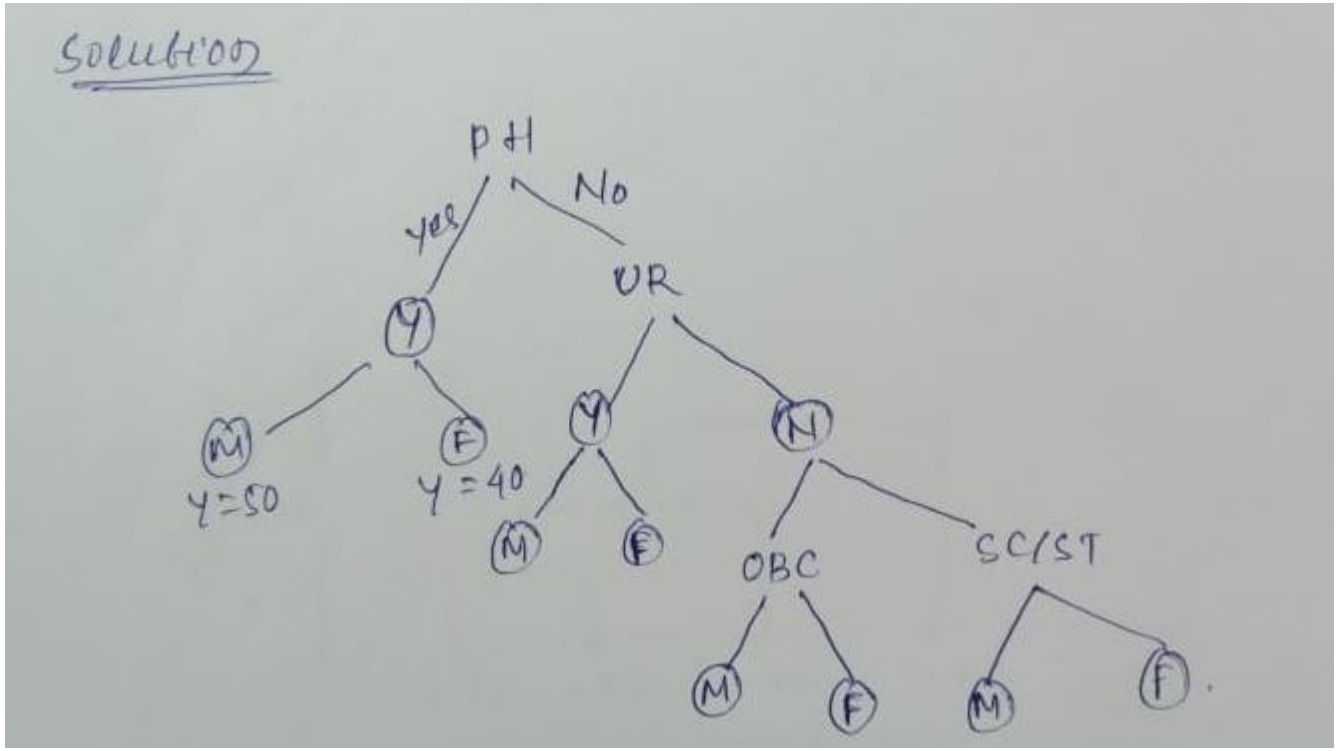
$= 0.0056$

$= 0.56\%$ (Ans)

Question 8: Draw a decision tree with the given data:

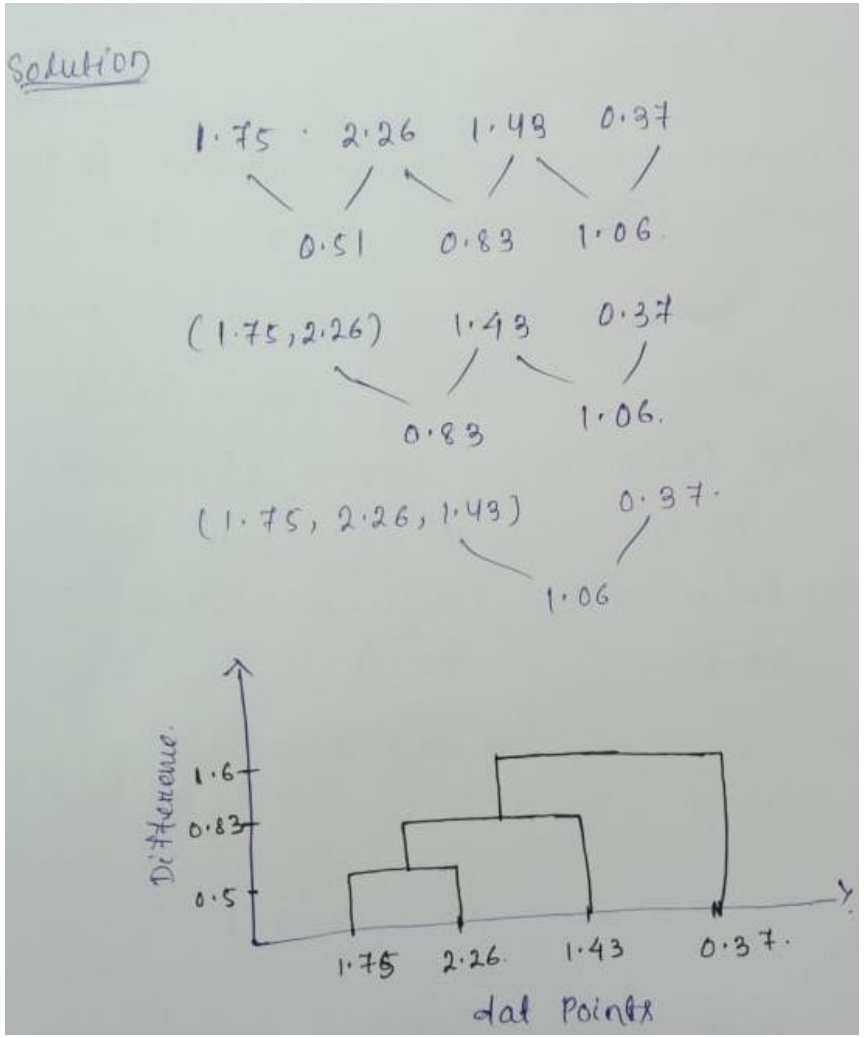
R1: IF(male) AND (UR) THEN cutoff=95
R2: IF(male) AND (OBC) THEN cutoff=85
R3: IF(male) AND (EWS) THEN cutoff=85
R4: IF(male) AND (SC/ST) THEN cutoff=70
R5: IF(female) AND (UR) THEN cutoff=90
R6: IF(female) AND (OBC) THEN cutoff=80
R7: IF(female) AND (EWS) THEN cutoff=80
R8: IF(female) AND (SC/ST) THEN cutoff=65
R9: IF(male) AND (PH) THEN cutoff=50
R10: IF(female) AND (PH) THEN cutoff=40

Exercise



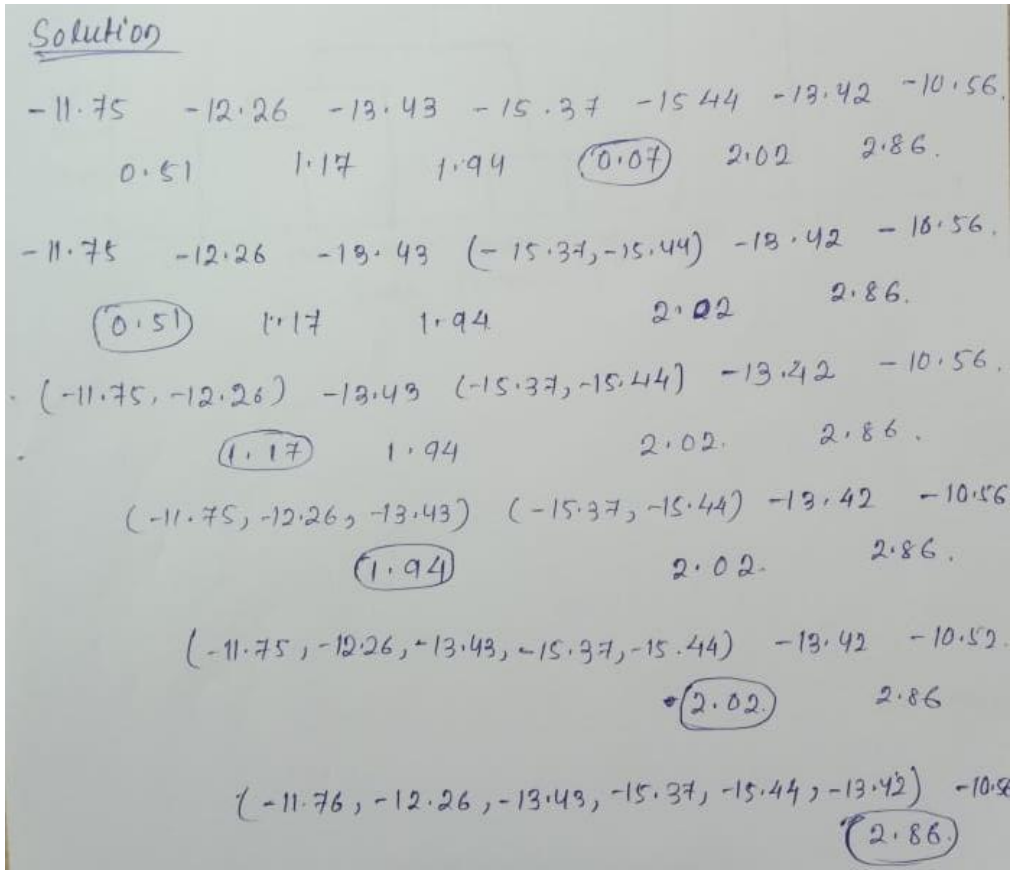
Exercise

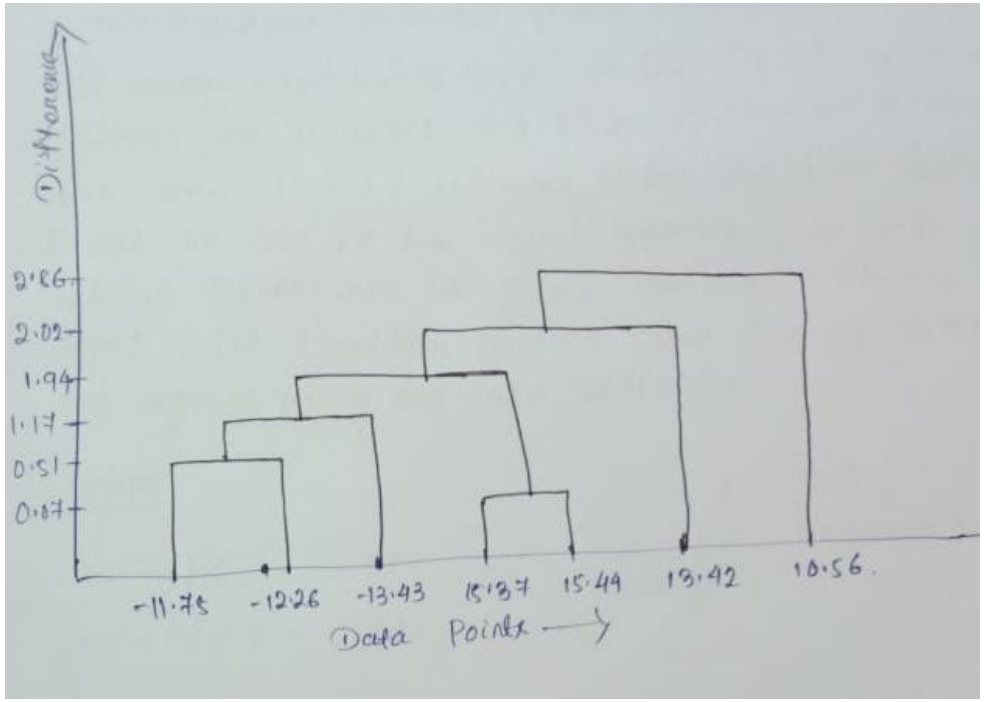
Question 9: For the following one dimensional dataset for the Band Gap of photovoltaic materials **{Cu_3N=1.75, GaP=2.26, GaAs=1.43, PbS=0.37}**, perform hierarchical clustering and plot the dendrogram to visualize it using single linkage.



Exercise

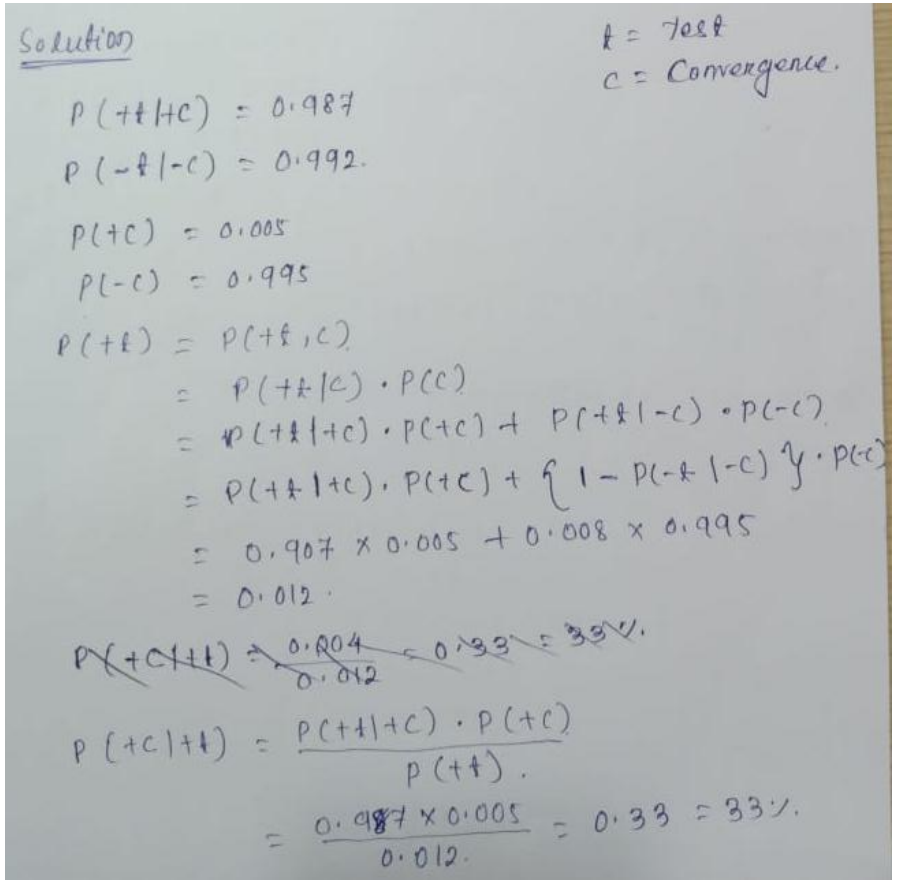
Question 10: For the following one dimensional dataset for the best docked posed energy for a protein ligand system for SARS-COV19 **{-11.75, -12.26, -13.43, -15.37, -15.44, 13.42, 10.56}**, perform hierarchical clustering and plot the dendrogram to visualize it using single linkage.





Exercise

Question 11: A DFT based new software claims that it can report SCF energy values for 98.7% of the input systems positively and negatively for 99.2% systems on comparison with some standard software data. We know that only 5 out of 1000 of the input geometry do not achieve convergence using the standard software. What is the likelihood that an input file will achieve convergence using the new software?



References

- Mitchell, T.M. (1997). *Machine Learning*. McGraw Hill.

➢ Covers the field of machine learning, which is the study of algorithms that allow computer programs to automatically improve through experience.★ ★★ ★

- Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer.

➢ This book reflects these developments while providing a grounding in the basic concepts of pattern recognition and machine learning.★ ★★ ★

- Hastie, T., Tibshirani, R. Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

➢ Describes the important ideas in a variety of fields such as medicine, biology, finance, and marketing in a common conceptual framework. While the approach is statistical, the emphasis is on concepts rather than mathematics.★ ★★

- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.

➢ Provides tools for Statistical Learning that are essential for practitioners in science, industry and other fields. Analyses and methods are presented in R. Covers topics like linear regression, classification, resampling methods, shrinkage approaches, tree-based methods, support vector machines, and clustering. ★ ★★

References

- Zaki, M.J., and Jr., W.B. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithm.* Cambridge University Press.

➢ Fundamental algorithms in data mining, automated methods to analyze patterns and models for all kinds of data, with applications ranging from scientific discovery to business intelligence and analytics are covered. Provides a broad yet in-depth overview of data mining, integrating related concepts from machine learning and statistics including exploratory data analysis, pattern mining, clustering, and classification. ★ ★ ★ ★

- Tang, P.N., Steibach, M., and Kumar, V. (2016). *Introduction to Data Mining.* Pearson.

➢ This book presents fundamental concepts and algorithms for those learning data mining for the first time. Each concept is explored thoroughly and supported with numerous examples. ★ ★ ★

- Shwartz, S.S., and Davis, S.B. (2014). *Understanding Machine Learning: From Theory to Algorithm.* Cambridge University Press.

➢ Introduces machine learning and provides a theoretical account of the fundamentals underlying machine learning and the mathematical derivations that transform these principles into practical algorithms. ★ ★ ★ ★

References

- Machine Learning | Stanford University

https://www.youtube.com/watch?v=PPLop4L2eGk&list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN

- Introduction to Machine Learning | University of Toronto

https://www.youtube.com/watch?v=FvAibtlARQ8&list=PL-Mfq5QSs8iS9XqKuApPE1TSlnZblFHF

- Machine Learning for Intelligent Systems | Cornell

https://www.youtube.com/watch?v=MrLPzBxG95I&list=PLl8OlHZGYOQ7bkVbuRthEsaLr7bONzbXS

- Machine Learning Course | Caltech

https://www.youtube.com/watch?v=mbyG85GZ0PI&list=PLD63A284B7615313A

Thank You