

Machine Learning (CS 431)

Presented by

Dr. Saroj Kr. Biswas
Associate Professor & HoD,
CSE



**Department of Computer Science and Engineering
National Institute of Technology, Silchar**

Machine Learning

Introduction, Decision Trees learning, Probability Primer, Bayes Decision Theory, Maximum-likelihood and Bayesian Parameter Estimation, Non-parametric Techniques, Bayes Networks, Optimization, Primer, Linear Discriminant Functions, Support Vector Machines. (introduction of ML with description of ML frame work)

Unit-2 Unsupervised Learning, Semi Supervised Learning, Reinforcement Learning, Statistical learning methods, PAC learning framework, Occam's Razor. (different ML algorithm)

Text Books

1. Mitchell T. M. , Machine Learning , McGraw Hill
2. Duda R. O., Hart P. E., Strok D. G. , Pattern Classification, Wiley Interscience

Course Outcomes (COs)

- 1.Understand the principles, advantages, limitations and possible applications of machine learning.
- 2. Identify the appropriate machine learning techniques for classification (**for other task also**).
- 3. Apply various pattern recognition, optimization and decision problems in Machine learning. (**Design of ML model to solve different applications in domains**)

Evaluation scheme

Internal Assessment = 20 (test, quiz, attendance, assignment)

Mid Semester = 30

End Semester = 50

What is Learning?

Learning is a process to acquire new or partially new knowledge by improving its performance from experience or environment.

There are different kinds of learning methods.

Learning

Concept learning:

Concept learning is also known as **category learning**, **concept attainment**, and **concept**.

In a concept learning task, a human or machine learner is trained to classify objects by being shown a set of example objects along with their class labels.

Rote Learning (memorization):

Memorizing things without knowing the concept/ logic behind them. A chartered engineer could play the role of a project manager while students play the role of the engineers during a meeting.

Passive Learning (instructions):

Passive learning is a learning paradigm **where learners receive information from the instructor and adopt it.**

Learning from a teacher/expert. (Direct Instruction, Watching Television, Modeled Instruction, University Lectures).

Learning

Active learning:

Active learning is the subset of **machine learning** in which a learning algorithm can query a user interactively to label data with the desired outputs.

Examples: case studies, group projects, think-pair-share, peer teaching, debates

Analogy (experience):

Learning new things from our past experience. Analogy learning can be described as the process of finding knowledge acquired in one domain and "using" it in a different domain by establishing similarities between "concepts" in the two domains and transferring relationships between concepts in one domain to the other.

Analogical learning typically involves:

- (1) identifying a similarity between two entities (concepts), often referred to as a source entity and a target entity, and
- (2) transferring properties or relationships from the source entity to the target entity.

Example: Case-Based Planning

Learning

Two kinds of analogy-based learning are here:

1. Transformational
2. Derivational

Transformational Analogy:

Look for a similar solution and copy it to the new situation making suitable substitutions wherever appropriate. Transformational analogy does not look at how the problem was solved it only looks at the final solution.

Suppose you are asked to prove a theorem in plane geometry. You might look for a previous solution of theorem which is very similar to current one

Derivational Analogy:

The history of the problem solution, the steps involved, is often relevant. We know how to find the area of a triangle and a square. Derivational analogy will help to solve the area/volume of a pyramid from this knowledge.

Learning

Inductive Learning (experience):

On the basis of past experience, formulating a generalized concept. Inductive reasoning makes broad generalizations from specific observations. Basically, there is data, then conclusions are drawn from the data. An example of inductive logic is, "The coin I pulled from the bag is a penny. Second coin from the bag is a penny. A third coin from the bag is a penny. Therefore, all the coins in the bag are pennies."

Deductive Learning:

Deriving new facts from past facts. Deductive reasoning, or deduction, starts out with a general statement, or hypothesis, and examines the possibilities to reach a specific, logical conclusion. For example, the premise "Every A is B" could be followed by another premise, "This C is A." Those statements would lead to the conclusion "This C is B." For example, "All men are mortal. Harold is a man. Therefore, Harold is mortal."

Learning

Abductive reasoning: Abductive reasoning usually starts with an incomplete set of observations and proceeds to the likeliest possible explanation for the group of observations. Abductive reasoning is often used by doctors who make a diagnosis based on test results.

For example, a person walks into their living room and finds torn up papers all over the floor. The person's dog has been alone in the room all day. The person concludes that the dog tore up the papers because it is the most likely scenario

What is ML?

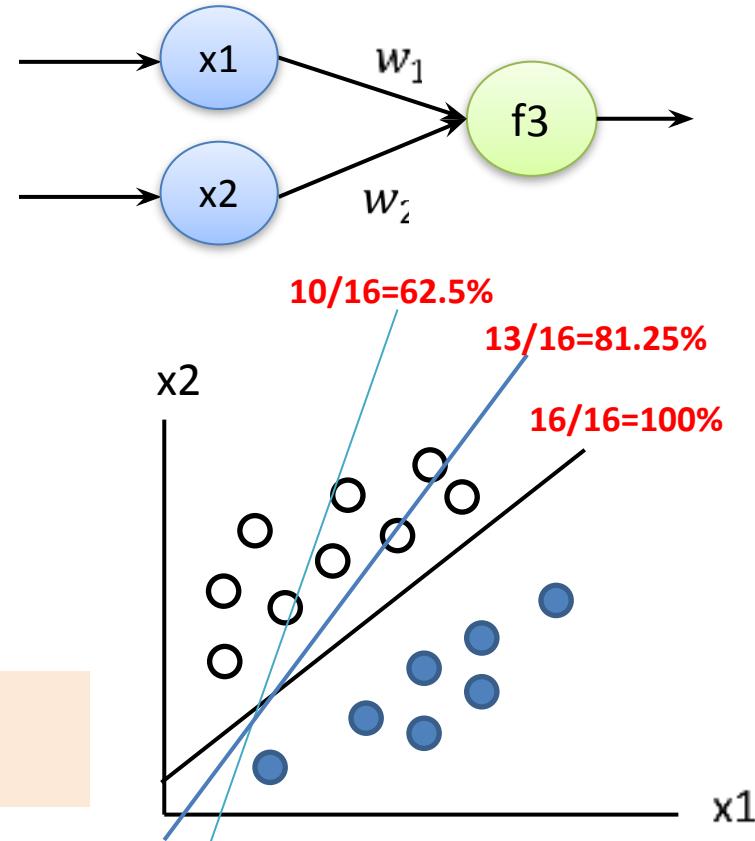
ML is programming computers to optimize a performance criterion using example data or past data. By E. Alpaydin

BP	Heart Beat	Class
120	70	Y
125	65	Y
130	59	N
150	78	N
135	66	N
125	75	N
120	76	Y

$$y = w_1x_1 + w_2x_2 + c$$

$$w_1x_1 + w_2x_2 + c = 0$$

$$x_2 = w_1/w_2 * x_1 + c/w_2$$



What is ML?

- o **Tom Mitchell** provides a more modern **definition**: “A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.” ... To find that logic is called “**machine learning**”

Why Machine Learning ?

- o For some defined tasks, if algorithms are available then machine learning is not required
- o For example, sorting numbers
- o For some tasks, algorithms are not readily available, to solve those tasks machine learning algorithms are required.
- o For example, to tell spam emails from legitimate emails.
- o The problems where no human experts exist. Rainfall forecasting
- o The problems where human experts exist, but they will be unable to explain their expertise. For example speech recognition.
- o The problems where the environment changes frequently. For example share market predictions.
- o The applications that need to be customised for individual users or for a group of users. For example, a program to filter unwanted electronic mail messages.
- o **ML is used to make quicker and reliable decision.**

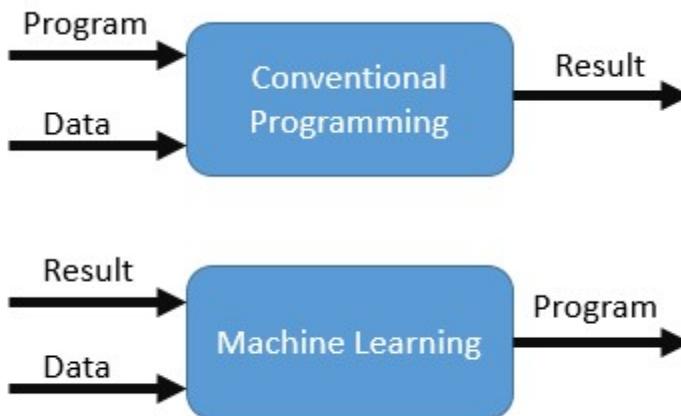
Some Facts about ML

- o ML does a good and useful approximation.
- o ML uses the theory of statistics in building mathematical models, because the core task is making inference from a sample.
- o Application of machine learning methods to large databases is called data mining
- o ML is a part of artificial intelligence.
- o ML helps us to find solutions to many problems in vision, speech recognition and robotics.
- o ML model may be predictive to make predictions in the future.
- o ML model may also be descriptive to gain knowledge or it can be both

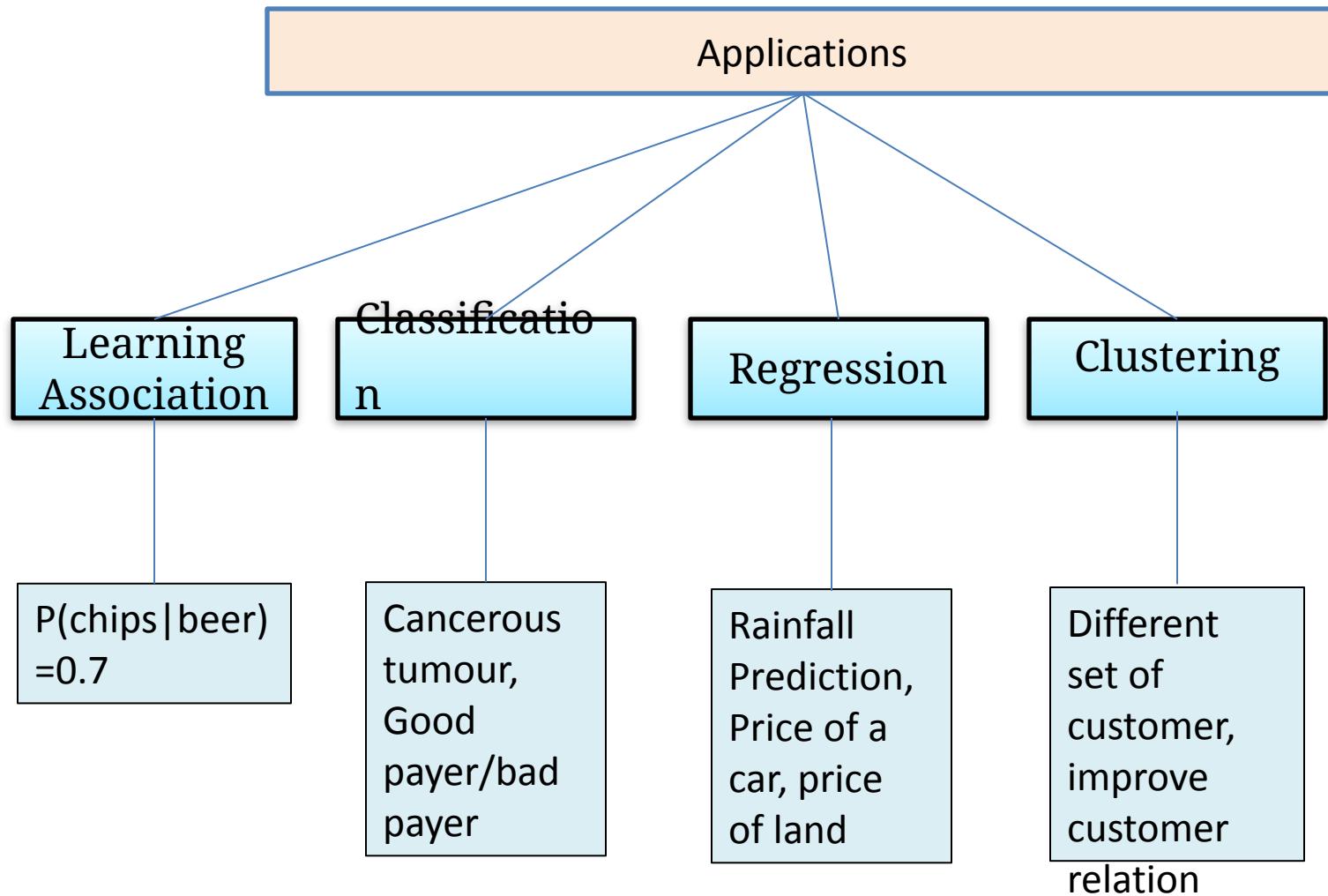
Conventional Programming vs ML Programming

In conventional programming, programs are created manually by providing input data and based on the programming logic, and the computer generates the output.

In machine learning programming, **the input and output data are fed to the algorithm, creating the program**



Kinds of ML Tasks



Kinds of ML Tasks

- o Learning Association
- o Classification
 - o Classification
 - o Prediction
 - o Pattern recognition
 - o Optical Character Recognition (OCR)
 - o Hand Writing Recognition
 - o Face Recognition
 - o Medical Diagnosis
 - o Speech Recognition
 - o Biometrics
 - o Knowledge Extraction
 - o Compression
 - o Outlier Detection
- o Regression

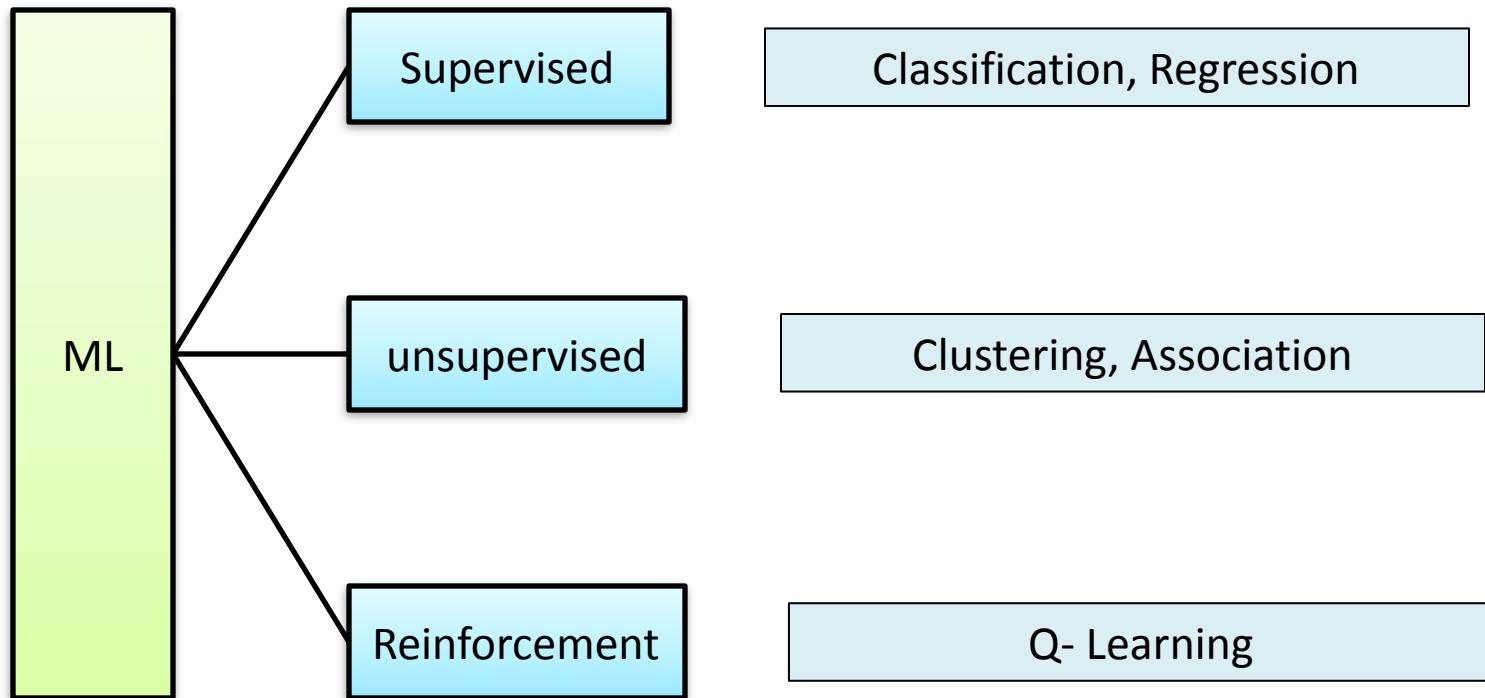
Examples of ML applications

Some most trending real-world applications of Machine Learning:

- **Image Recognition** (face detection)
- **Speech Recognition** (Speech to text)
- **Traffic prediction** (Google Map)
- **Product recommendations** (Amazon, Netflix, etc.)
- **Self-driving cars** (Deep learning is used)
- **Email Spam and Malware Filtering** (Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier)
- **Online Fraud Detection** (Feed Forward Neural network)
- **Stock Market trading** (Recurrent neural network like LSTM)
- **Weather prediction** (rain fall prediction etc. Recurrent neural network like LSTM)
- **Medical Diagnosis** (finding brain tumors; Deep learning)
- **Automatic Language Translation** (GNMT (Google Neural Machine Translation))

Classification of ML

Classification of ML Algorithm



Classification of

ML Supervised

Supervised Learning (SL) is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.

Advantage:

- Huge number of application
- Performance is good

Disadvantages:

- **Slow** (it requires human experts to manually label training examples one by one)
- **Costly** (a model should be trained on the large volumes of hand-labeled data to provide accurate predictions)

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Classification of

ML
Supervised

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- **Artificial Neural Network (ANN)**
- K- Nearest Neighbour (K-NN)
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Classification of

ML Unsupervised

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets.

Advantages:

- solves the problem by learning the data without any labels.
- It is very helpful in finding patterns in data, which are not possible to find using normal methods.
- There is lesser complexity compared to the supervised learning task. Here, no one is required to interpret the associated labels and hence it holds lesser complexities.
- It is reasonably easier to obtain unlabeled data.

Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

Classification of

ML Unsupervised

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets.

Disadvantages:

- has a *limited area of applications* (mostly for clustering purposes)
- provides *less accurate results*
- might require human intervention to understand the patterns and correlate them with the domain knowledge
- cannot get precise information regarding the output

Use:

Anomaly detection, Segmentation,
Dimensionality reduction

Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

Classification of

ML
Unsupervised

- **Clustering**
K-Means
K-Mediod
CURE
BIRCH
- **Association Rule Mining**
Apriori Algorithm
Predictive Apriori Algorithm
Tertius Algorithm
Eclat
FP-Growth

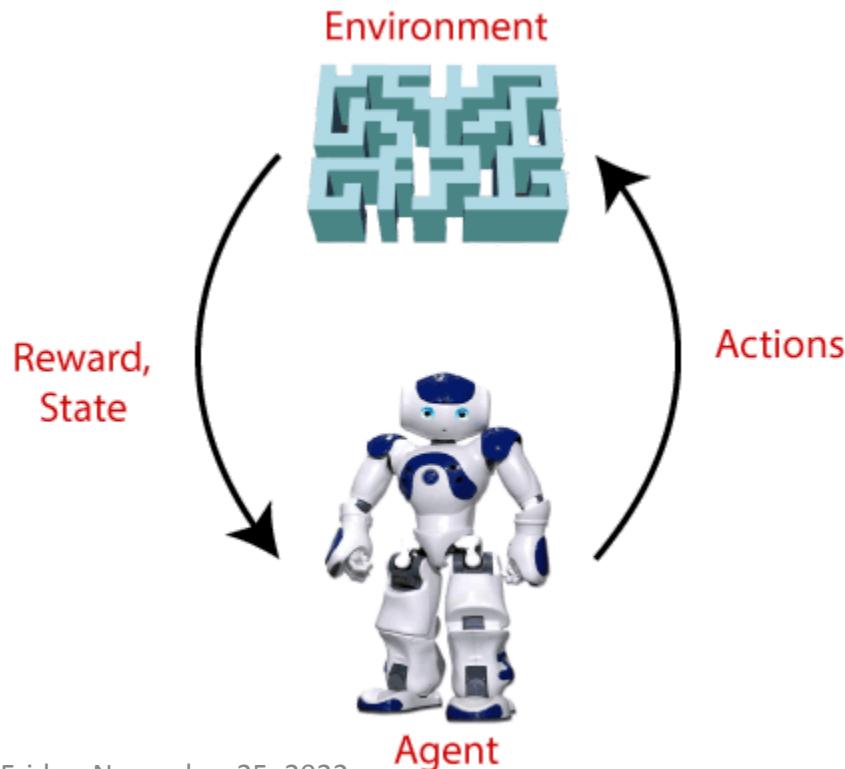
Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

Classification of

ML Reinforcement

Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones.

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. **For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty**



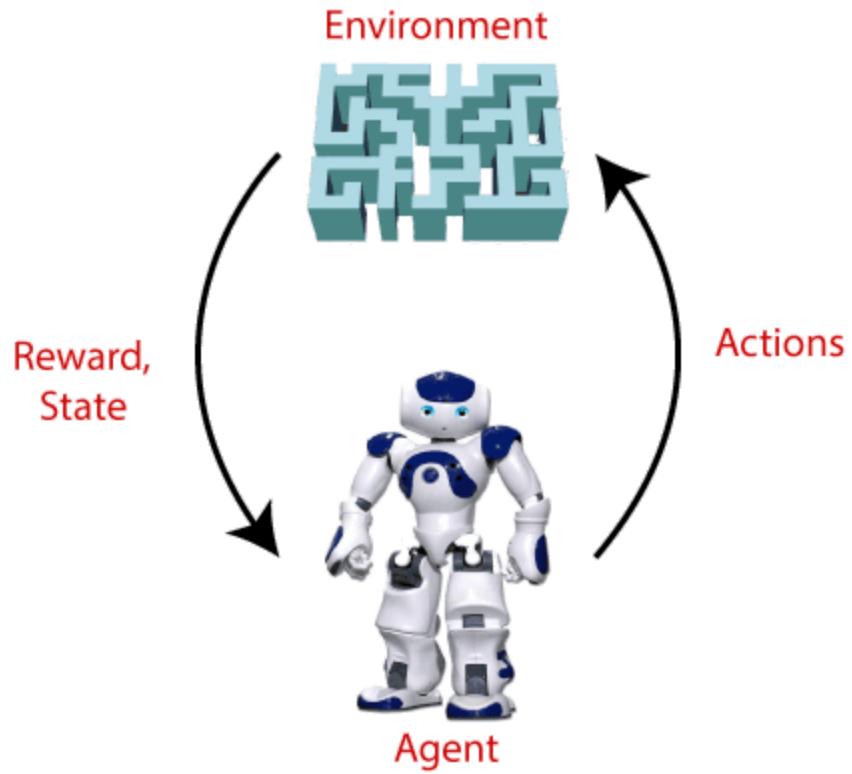
BP	Heart Beat	Weight	Feedback
120	70	50	reward
125	65	60	penalty
130	59	52	penalty
150	78	70	penalty
135	66	85	reward
125	75	82	reward
120	76	90	reward

Classification of

ML Reinforcement

Advantages:

- Reinforcement learning doesn't require large labeled datasets.
- It's **Innovative**.
- **Bias Resistance**
- **Goal-oriented**, Reinforcement learning can be used for sequences of actions.
- Reinforcement learning is **Adaptable**. Reinforcement learning doesn't require retraining because it adapts to new environments automatically on the fly.
- Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
- The model can correct the errors that occurred during the training process



Classification of

ML Reinforcement

Disadvantages:

- Can diminish the results due to too much reinforcement learning
- Not preferable to use for solving simple problems.
- Needs a lot of data and a lot of computation. It is data-hungry
- Assumes the world is Markovian, which it is not
- The curse of dimensionality limits reinforcement learning heavily for real physical systems.

Applications:

Robotics: Robot navigation, walking etc

Control: Adaptive control such as Factory processes etc.

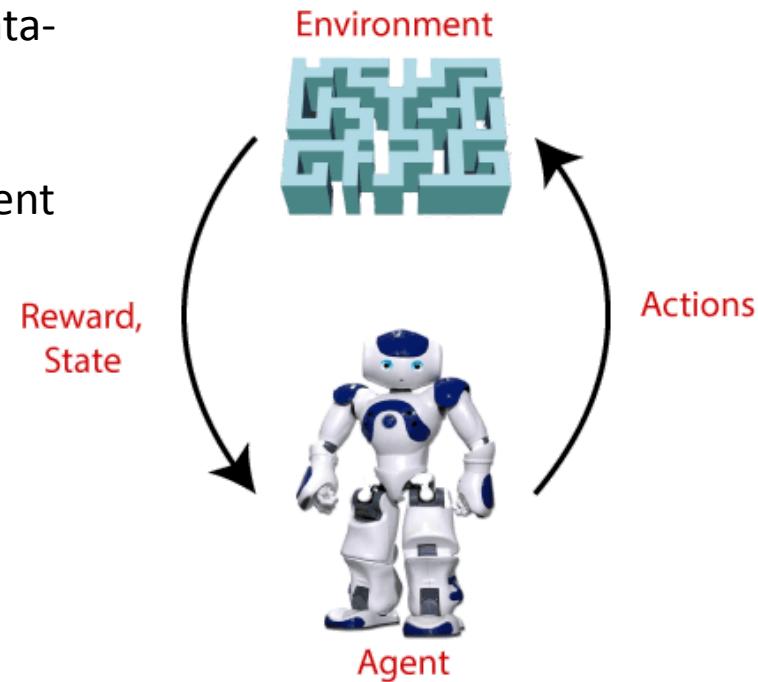
Game Playing: Game playing like chess, etc.

Chemistry: Optimizing the chemical reactions.

Business: business strategy planning

Manufacturing: automobile manufacturing companies

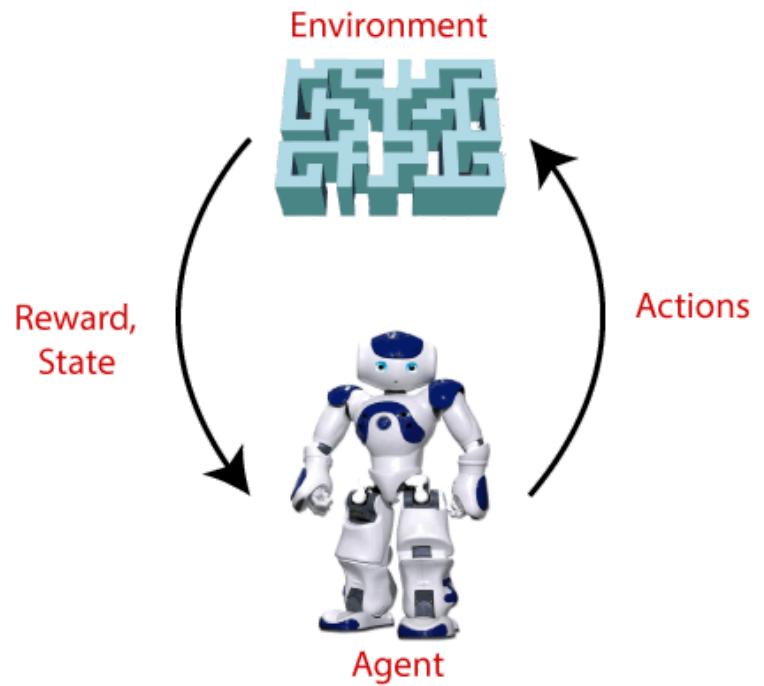
Finance Sector: finance sector for evaluating trading strategies



Classification of

ML Reinforcement

- Markov Decision Process (MDP)
- Q learning: Deep-Q-Neural Network (DQN)
- State Action Reward State Action (SARSA)



Semi-Supervised

Semi-Supervised Learning (SSL) is a learning problem that involves a small number of labeled examples and a large number of unlabeled examples.

Semi-supervised machine learning is a **combination of supervised and unsupervised machine learning methods**.

- Unlike unsupervised learning, SSL works for a variety of problems from classification and regression to clustering and association.
- Unlike supervised learning, the method uses small amounts of labeled data and also large amounts of unlabeled data, which reduces expenses on manual annotation and cuts [data preparation](#) time.

Advantages:

- Easy to understand.
- Reduces the amount of annotated data used.
- A stable algorithm
- High efficiency
- Large applications
- Work as domain expert

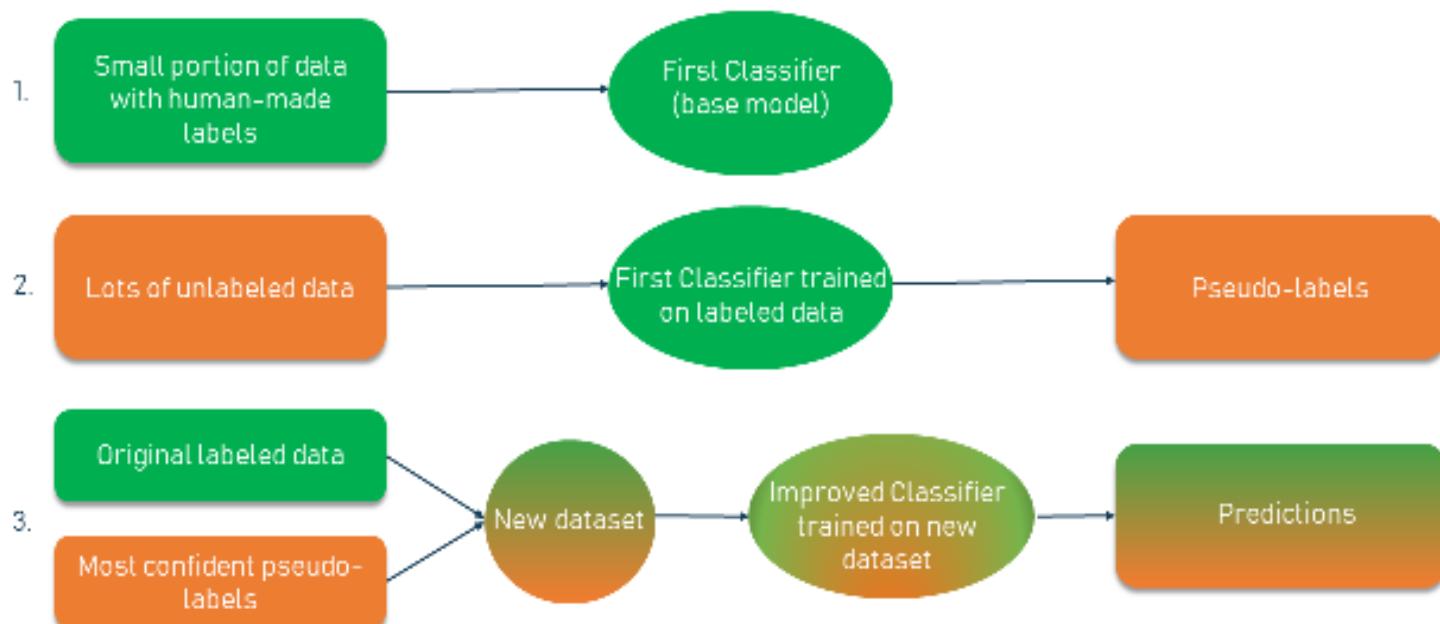
BP	Heart B	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	?
135	66	85	?
125	75	82	?
120	76	90	?

Semi-Supervised

Disadvantages:

- Low accuracy
- Results are not stable
- Not appropriate for complex problems

SEMI-SUPERVISED SELF-TRAINING METHOD

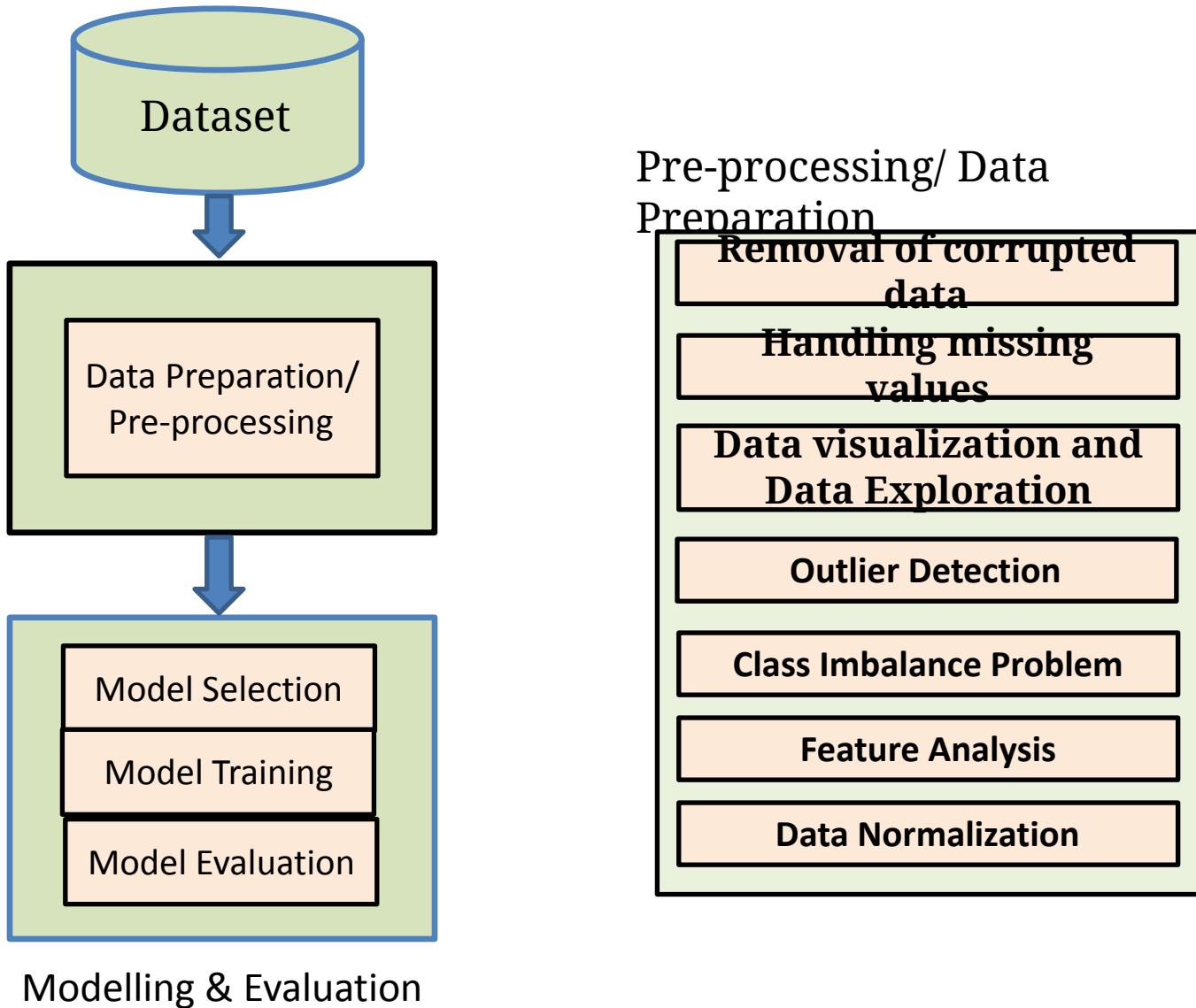


Semi-Supervised

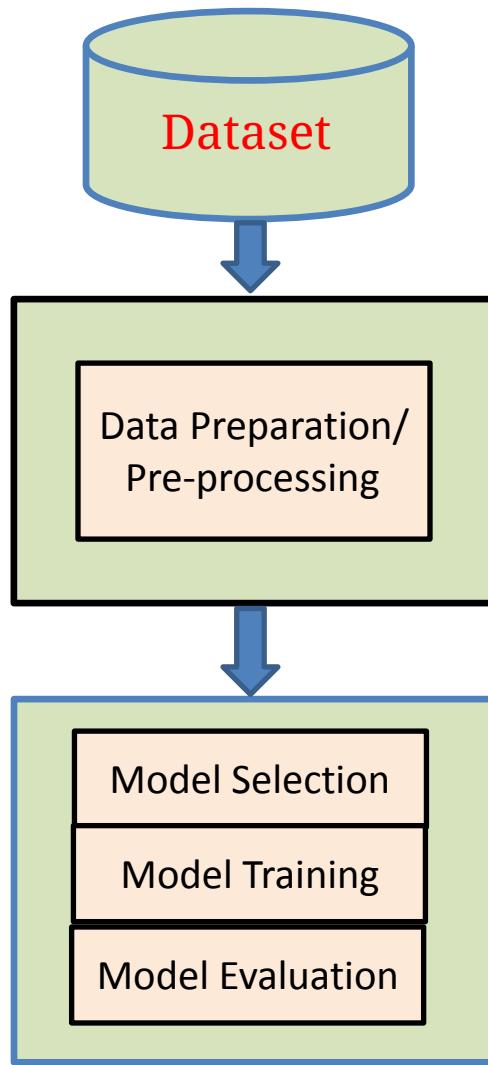
Applications:

- Text document classification
- Speech Analysis
- Protein Sequence Classification
- Internet Content Classification

ML Model

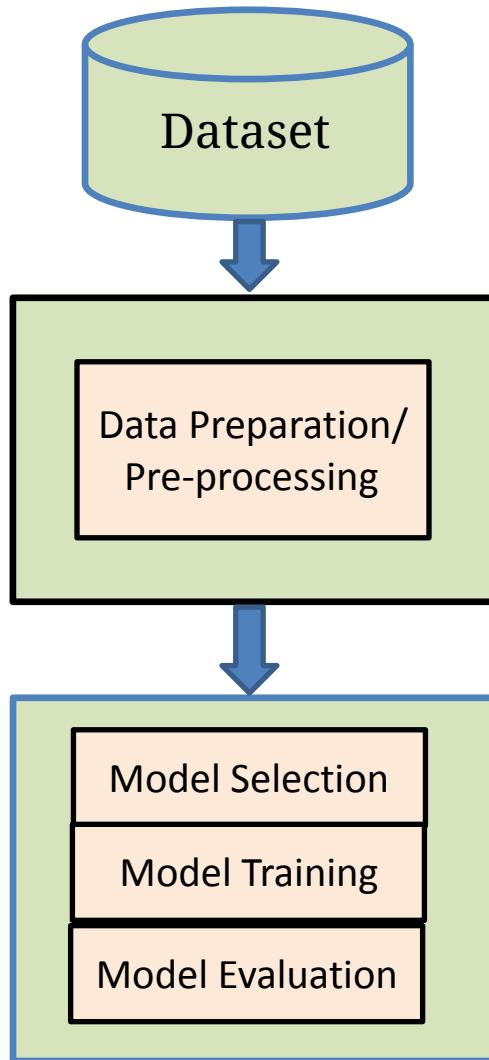


ML Model



Modelling & Evaluation

ML Model



Modelling & Evaluation

Pre-processing/ Data Preparation

Removal of corrupted data

Handling missing values

Data visualization and Data Exploration

Outlier Detection

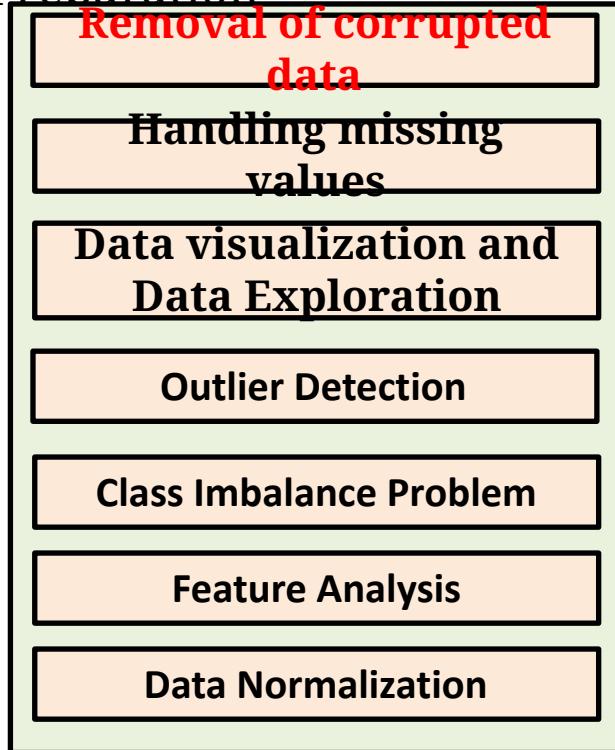
Class Imbalance Problem

Feature Analysis

Data Normalization

ML Model

Pre-processing/ Data Preparation



Improves quality of the training and reliability

Removal of erroneous data: not in format,
not in range, outliers, missing values

A	B	C	D	E

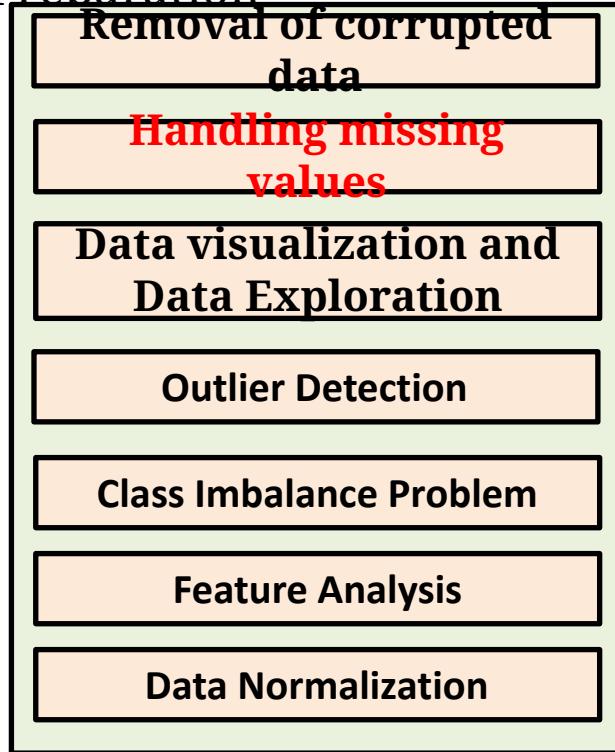
A	B	C	D

Age	weight	Height	pressure	Heart Beat	class
25	54	5.6	110	300	Y
30	64	5.7	130	400	N

Age	weight	Height	pressure	class
25	54	5.6	110	Y
30	64	5.7	130	N

ML Model

Pre-processing/ Data Preparation



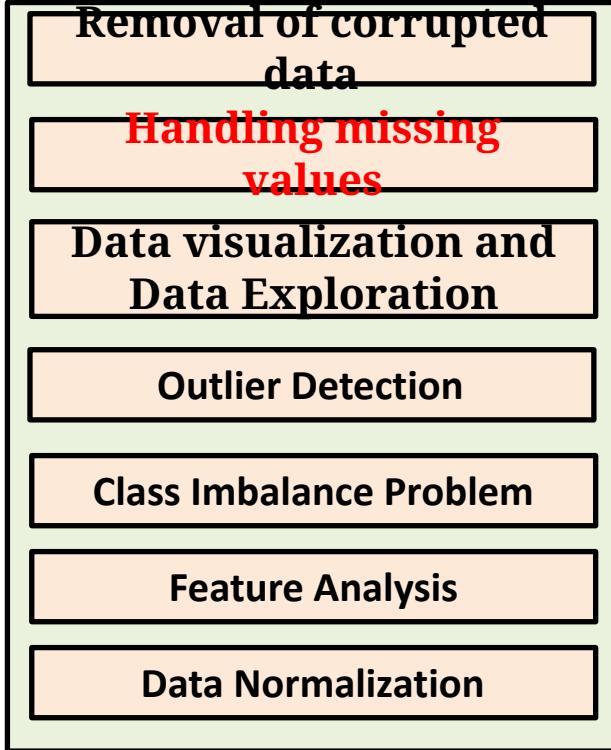
What is a Missing Value?

How is it created ?

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59		N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124	56	81	Y
115	70	73	Y

ML Model

Pre-processing/ Data Preparation



Type of Missing values:

- MCAR: Missing Completely At random
- MAR: Missing At Random
- MNAR: Missing Not At Random

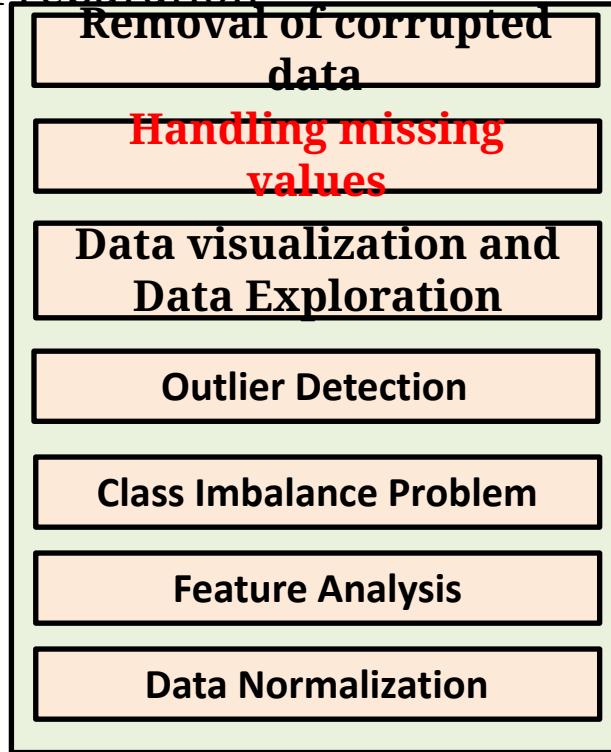
MCAR: Missing Completely At random

- If there is no relationship among the missing data and any other variable of the dataset
- Probability of missing is not related with any other variable.

Roll. No	Due book
120	05
101	
102	03
112	09
105	02

ML Model

Pre-processing/ Data Preparation



Type of Missing values:

- MCAR: Missing Completely At random
- MAR: Missing At Random
- MNAR: Missing Not At Random

MAR: Missing At Random

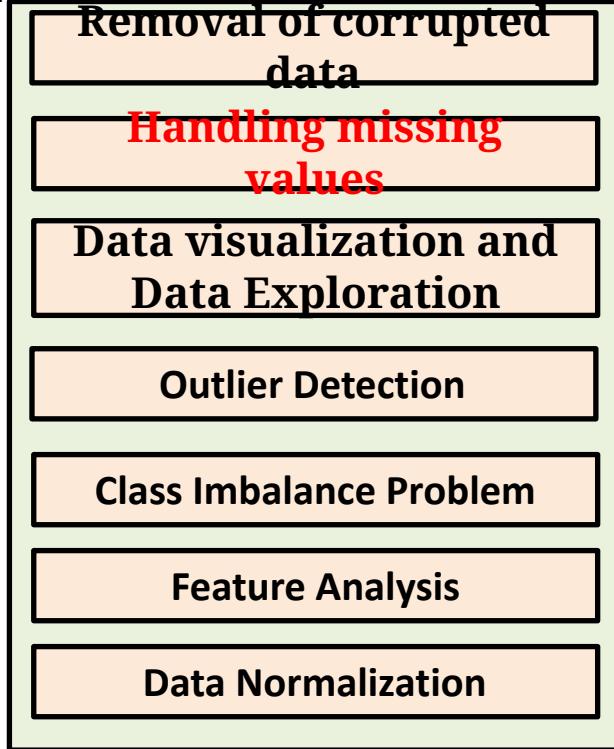
If there is a relationship among the missing data and any other variable of the dataset. Therefore need to analyze the relationship between the missing data and the variable on which it depends upon.

If the probability of being missing is the same only within groups defined by the *observed* data, then the data are missing at random (MAR). MAR is a much broader class than MCAR

Roll. No	Due book	Sex	Roll. No	Due book	Sex
120	05	M	100		M
101		F	103		M
102	03	F	115	03	F
112	09	M	111	09	F

ML Model

Pre-processing/ Data Preparation



Type of Missing values:

- MCAR: Missing Completely At random
- MAR: Missing At Random
- MNAR: Missing Not At Random

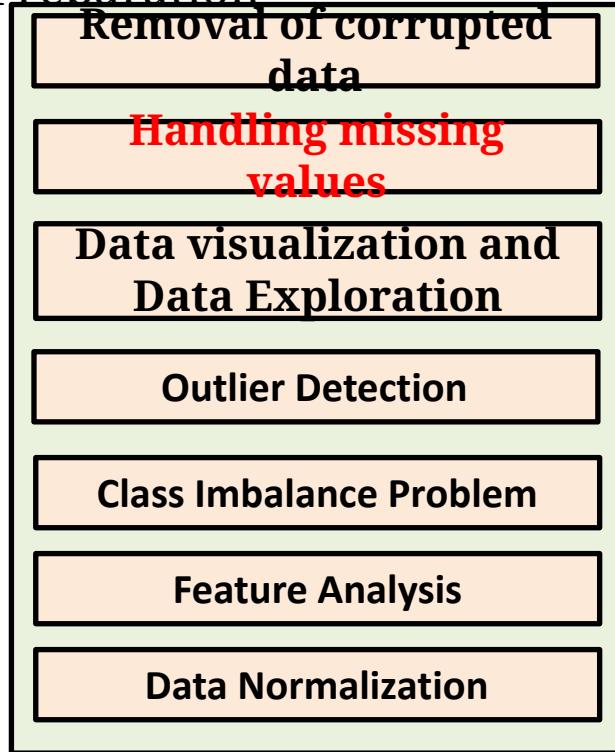
MNAR: Missing Not At Random

There is a relationship between the missing data and the variable itself in which the data is missing.
Required to proper understanding about the variable before making any imputation.

Year	No. Population	Year	No. Population
2005	1000	2010	1800
2006	1100	2011	2000
2007	1300	2012	2300
2008		2013	
2009	1600	2014	2800

ML Model

Pre-processing/ Data Preparation



Improves quality of the training

Handling Missing Values:

1. Deletion
2. Data Imputation

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59		N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124	56	81	Y
115	70	73	Y

Handling Missing Values

BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124	56	81	Y
115	70	73	Y

Deletion:

Deletion methods are used when missing is occurred due to “missing completely at random” and “missing at random”

1. Deleting Rows
2. Deleting Columns
3. Pairwise

Handling Missing Values

Deletion:

BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76	52	Y
124	56	81	Y
115	70	73	Y



BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120	66	85	N
124	75	82	N
154	76	52	Y
124	56	81	Y
115	70	73	Y

1. Deleting Rows
2. Deleting Columns
3. Pairwise

Handling Missing Values

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124		81	Y
115	70	73	Y



BP	Weight	Class
100	50	Y
101	60	Y
102	52	N
120	70	N
120	85	N
124	82	N
154	52	Y
124	81	Y
115	73	Y

Deletion:

- 1. Deleting Rows**
- 2. Deleting Columns**
- 3. Pairwise**

Handling Missing Values

Deletion:

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124		81	Y
115	70	73	Y



BP	Weight	Class
100	50	Y
101	60	Y
102	52	N
120	70	N
120	85	N
124	82	N
124	81	Y
115	73	Y

Pairwise correlation between predictor and target is found to help in deletion

BP-Class-----High

Heart Beat—Class----Low

Weight-Class-----High

Handling Missing Values

BP	Heart Beat	Weigh t	Class
100	70	50	Y
101		60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124		81	Y
115	70	73	Y



BP	Heart Beat	Class
100	70	Y
102	59	N
120	66	N
124	75	N
154	76	Y
115	70	Y

Deletion:

1. Deleting Rows
2. Deleting Columns
3. Pairwise

Pairwise correlation between predictor and target is found to help in deletion

BP-Class-----High

Heart Beat—Class----High

Weight-Class-----Low

Handling Missing Values

Deletion:

Pros: Trained model becomes robust as all the missing values are deleted.

Cons: 1. loss of information and 2. trained model works poorly if deletion is excessive

Handling Missing Values

BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120	67	70	N
120	66	85	N
124	75	82	N
154	76	52	Y
124	56	81	Y
115	70	73	Y

Imputation:

1. Mean
 2. Median
 3. Mode
 4. Linear Interpolation
 5. Linear Regression
 6. K-NN
- Different ML techniques

$$\begin{aligned}\text{Mean} &= (70+65+59+66+75+76+56+70)/8 \\ &= 67.125 \\ &= 67\end{aligned}$$

Handling Missing Values

BP	Heart Beat	Weight	Class
100	Y	50	Y
101	Y	60	Y
102	N	52	N
120	Y	70	N
120	Y	85	N
124	Y	82	N
154	Y	52	Y
124	Y	81	Y
115	N	73	Y

Imputation:

1. Mean
2. Median
3. Mode
4. Linear Interpolation
5. Linear Regression (ML)
6. K-NN (ML)
7. SoftImpute (ML)
8. Mice (ML) (good)
9. MatrixFactorization (ML)
10. miss- Forest (ML)
11. Deductive Imputation (LR)
12. Factor Analysis of Mixed Data (FAMD) (ML)

Categorical Data

K-NN is used value of k = 4

Handling Missing Values

Last Observation Carried Forward (LOCF)

If data is time-series data, one of the most widely used imputation methods is the last observation carried forward. Whenever a value is missing, it is replaced with the last observed value.

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	90	86%
6	6-Jan	155	87%
7	7-Jan	155	89%
8	8-Jan	155	90%
9	9-Jan	180	92%

Handling Missing Values

Next Observation Carried Backward (NOCB)

It is a similar approach like LOCF which works oppositely by taking the first observation after the missing value and carrying it backward ("next observation carried backwards", or NOCB).

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	155	86%
6	6-Jan	155	87%
7	7-Jan	180	89%
8	8-Jan	180	90%
9	9-Jan	180	92%

Handling Missing Values

Linear Interpolation

It is a mathematical method that adjusts a function to data and uses this function to extrapolate the missing data. **The simplest type of interpolation is linear interpolation, where the values before the missing data and after the same is used.**

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	120	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	170	90%
9	9-Jan	180	92%

$$(90+150)/2 = 120$$

$$(160+180)/2 = 170$$

Handling Missing Values

Adding a category to capture NA

This is perhaps the most widely used method of missing data imputation for categorical variables.

This method consists of treating missing data as an additional label or category of the variable.

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	N/A	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	N/A	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	N/A	180	95%



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Missing	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	Missing	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	Missing	180	95%

Handling Missing Values

Frequent category imputation

Replacement of missing values by the most frequent category is the equivalent of mean/median imputation. It consists of replacing all occurrences of missing values within a variable with the variable's most frequent label or category.

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	N/A	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	N/A	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	N/A	180	95%



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Fast+	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	Fast+	180	95%

Handling Missing Values

Missing Value Treatment using most recent data imputation techniques

MICE (Multiple Imputation by Chained Equation)

Handling Missing Values

Multiple Imputation

Multiple Imputation (MI) is a statistical technique for handling missing data.

The key concept of MI is to use the distribution of the observed data to estimate a set of plausible values for the missing data.

Estimates are combined to obtain a set of parameter estimates.

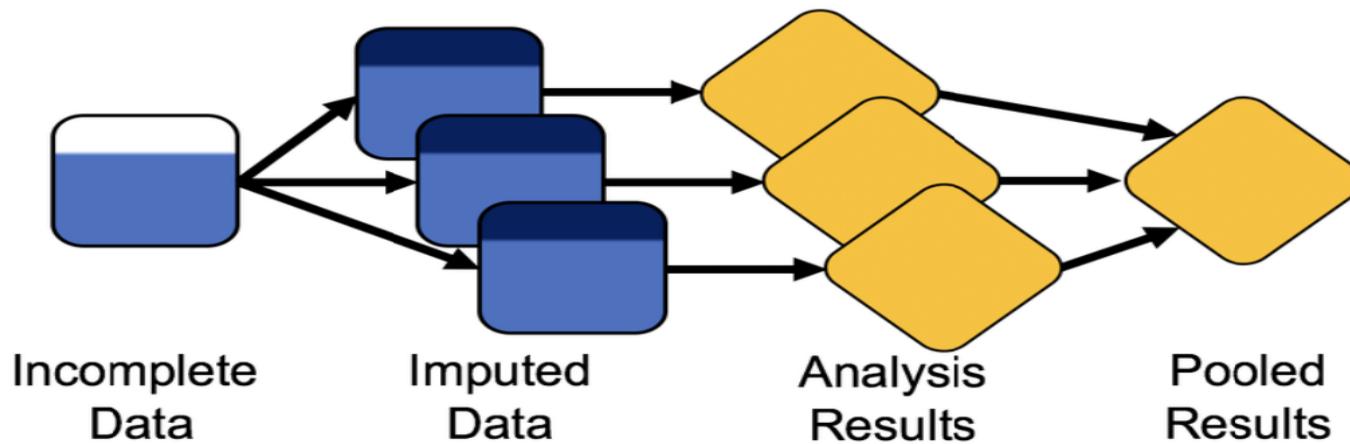
Multiple datasets are created and then analysed individually but identically to obtain a set of parameter estimates.

Multiple Imputation by Chained Equations (MICE) approach is a flexible way of handling more than one missing variable,

The benefit of the multiple imputations is to restore the natural variability of the missing values.

Handling Missing Values

Multiple Imputation



Multiple Imputation

First step would be to remove the "Personal Loan" column as it is the target column, we will not need this column for imputation.



age	experience	salary(K)	Personal loan
25		50	1
27	3		1
29	5	80	0
31	7	90	0
33	9	100	1
	11	130	0



age	experience	salary(K)
25		50
27	3	
29	5	80
31	7	90
33	9	100
	11	130

Multiple Imputation

Second step would be a simple imputation, such as imputing the mean, which is performed for every missing value in the dataset that leads to the formation of zeroth dataset.

age	experience	salary(K)
25		50
27	3	
29	5	80
31	7	90
33	9	100
	11	130



age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
29	11	130

zeroth dataset

Multiple Imputation

Third step would be to remove the "age" imputed values and keep the imputed values in other columns as shown here. Now, we will be
 imputing the columns from left to right.

age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
29	11	130



age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
	11	130

Multiple Imputation

In the fourth step, the remaining features and rows (top 5 rows of experience and salary) become the feature matrix (purple cells), "age" becomes the target variable (yellow cells).

We will run the linear regression model on the fully filled rows with X= experience and salary and Y=age. To estimate the missing age, we will use the missing value row (white cells) as the test data. So, top 5 rows will be training data and the last row that has missing age will be test data. We will use (experience = 11 and salary = 130) to predict corresponding "age" value. When I did this, I found that my model predicted the age as 34.99.

age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
	11	130

Multiple Imputation

In the fifth step, we update the predicted age value in the missing cell in "age" column.

Now, remove "experience" imputed value. The remaining features and rows becomes the feature matrix(purple cells) and "experience" becomes the target variable(yellow cells). We will run the linear regression model on the fully filled rows with X= age and salary and Y=experience. To estimate the missing experience, we will use the missing value row (white cells) as the test data. The predicted value for experience is 0.98.

age	experience	salary(K)
25		50
27	3	90
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

In the sixth step, we update the predicted experience value in the missing cell in "experience" column and remove "salary" imputed value.

The remaining features and rows becomes the feature matrix(purple cells) and "salary" becomes the target variable(yellow cells). We will run the linear regression model on the fully filled rows with $X = \text{age}$ and experience and $Y = \text{salary}$. To estimate the missing salary, we will use the missing value row (white cells) as the test data. The predicted value for Salary is 70.

age	experience	salary(K)
25	0.98	50
27	3	
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

Now we **impute the missing values in the original dataset**

and the predicted values after 1st iteration is shown here.

Let's name this as "**First dataset**".

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

In the seventh step, We will subtract the two datasets (zeroth and first). The resultant dataset is as below:

The diagram illustrates the subtraction of two datasets. On the left, there is a table with 7 rows and 3 columns: age, experience, and salary(K). The middle column, 'experience', is highlighted in cyan. Below this table is the word 'minus'. To the right of 'minus' is another table with 7 rows and 3 columns, also with the 'experience' column highlighted in cyan. A red arrow points from the right side of the second table to the right side of the third table. The third table has 7 rows and 3 columns, with the 'experience' column highlighted in cyan. The values in the 'experience' column of the third table are 6.02, 0, 0, 0, 0, -5.99, and 0 respectively.

age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
29	11	130

minus

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

age	experience	salary(K)
0	6.02	0
0	0	20
0	0	0
0	0	0
0	0	0
-5.99	0	0

If we observe, the absolute difference between 2 datasets are higher in few imputed values. Our aim is to reduce these differences close to 0. To achieve this we have to do many iterations. So, now we repeat the steps 2-6 with the new dataset (first), until we get a stable model. i.e. until the difference between the 2 latest imputed datasets becomes very small, close to 0. Technically, we stop the iterations when a pre-defined threshold is reached or a pre defined maximum number of iterations gets completed.

Multiple Imputation

Now we will use the "first" dataset as our base dataset to do imputations, and discard the "Zeroth" dataset which had the mean imputations. With "first" dataset as base, let's perform all the previous steps and again predict the imputed values for the initial 3 missing values.

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

Here's is the iteration 2 values and the new dataset values are subtracted from the first dataset and got the difference matrix as below:

Iteration 2

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

After all imputations



age	experience	salary(K)
25	0.975	50
27	3	70
29	5	80
31	7	90
33	9	100
34.95	11	130

After
Second - First



age	experience	salary(K)
0	0.005	0
0	0	0
0	0	0
0	0	0
0	0	0
0.004	0	0

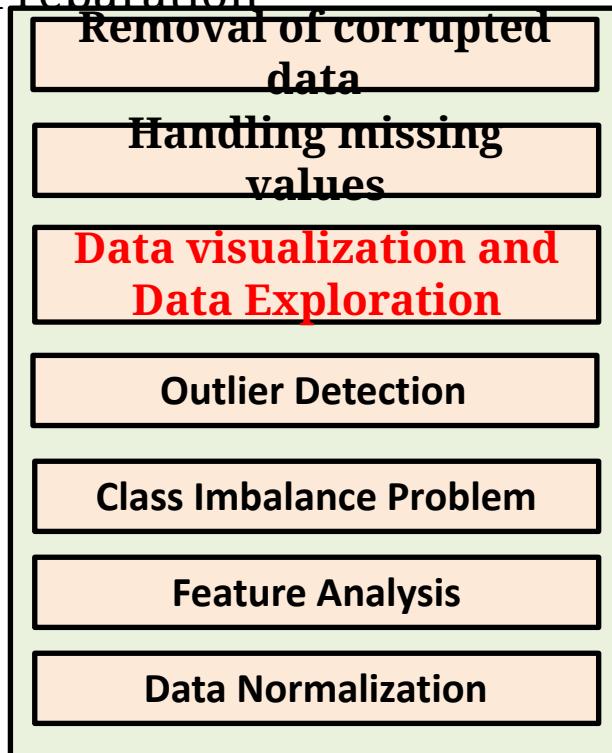
First Dataset

Second Dataset

Difference Matrix

Now, after second iteration, we can see that the difference is very negligible. We can either stop here as we almost got the same numbers, or proceed with next iteration until we get 0 difference.

Pre-processing/ Data Preparation



Helps to understand underlying behaviour of data which helps to take right steps in data preparation and modelling.

Data Visualization and Data Exploration

Data Exploration:

- Mean (central tendency)
- Median (central tendency)
- Variance (Data Spread)

Set of observation=21 89 34 67 96

$$\text{Mean} = (21+89+34+67+96)/5 = 61.4$$

21 34 67 89 96; Median = 67

A1: 44, 46, 48, 45, 47

A2: 34, 46, 59, 39, 52

For both mean and median 46

To measure data dispersion or data spread, variance is measured

variance(A1)= 2; variance(A2)= 79.6

A1 values are concentrated around mean

A2 values are extremely spread out

Set of observation= 21, 20, 23, 24, 25 84 67, 55 96

$$\text{Mean} = (21+20+23+24+25+84+67+55+96)/9 = 46.11$$

20, 21, 23, 24, 25, 55, 67, 84, 96; Median= 25

Set of observation= 21 89 34 67 200

$$\text{Mean} = (21+89+34+67+200)/5 = 82.2$$

21 34 67 89 200; Median = 67

Data Visualization and Data Exploration

Data Visualization:

Box Plot: An effective mechanism to get a one-shot view and understand the nature of data.

It gives a standard visualization of five statistical summary: **minimum**, **first quartile(Q1)**, **median(Q2)**, **third quartile(Q3)** and **maximum**.

Box spans from Q1 to Q3 = **Inter-Quartile Range (IQR)**

Lower Range (LR) extends up to = (Q1 - 1.5 times of IQR)

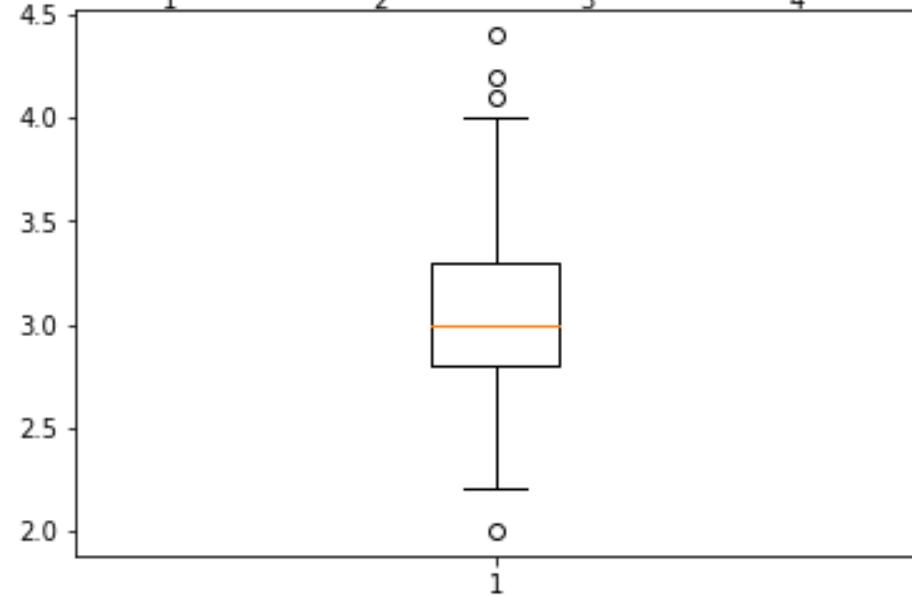
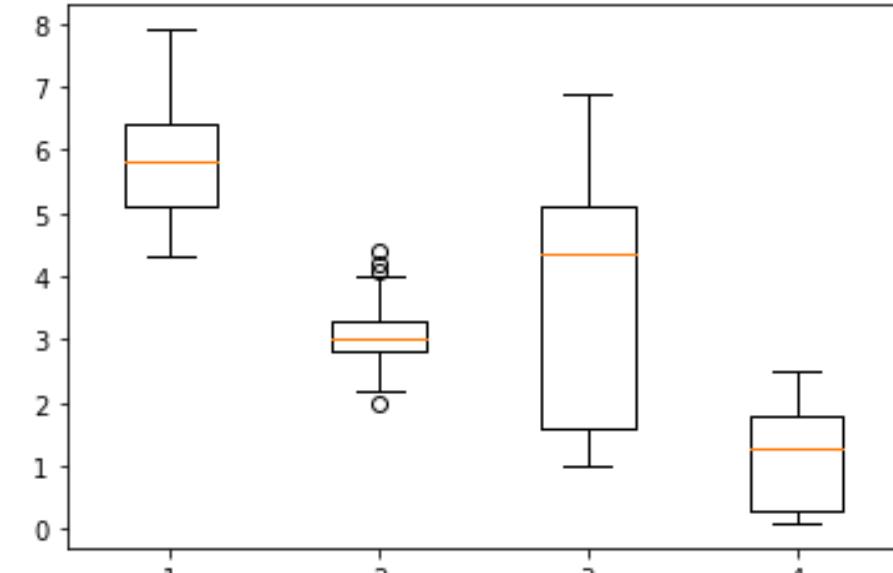
For some x: Q1=73, Q2=76 and Q3=79

$$\text{IQR}=(\text{Q3}-\text{Q1})=(79-73)=6$$

$$\text{LR}=(\text{Q1}-1.5 \times \text{IQR})=(73-1.5 \times 6)=(73-9)=64$$

Say some lower data values of x: 70, 63, 60

Minimum= 70 which is larger than 64



Data Visualization and Data Exploration

Data Visualization:

Upper Range (UR) extends up to = (Q3 + 1.5 times of IQR)

For some x: Q1=73, Q2=76 and Q3=79

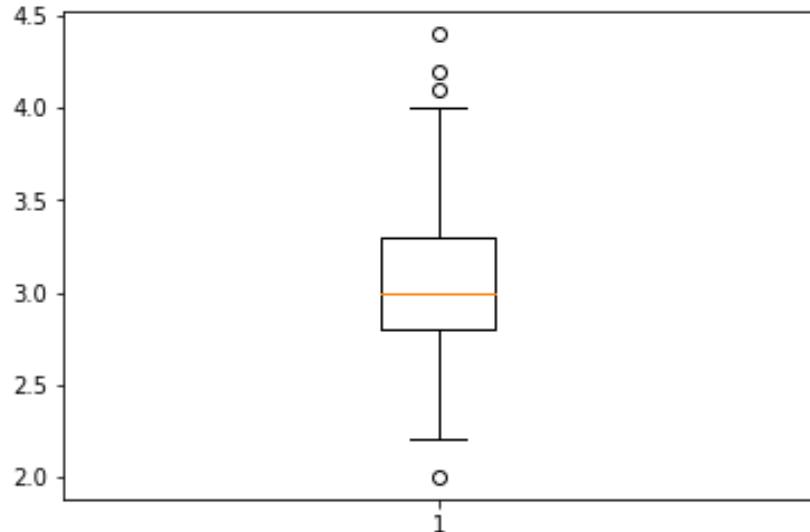
$$\text{IQR}=(\text{Q3}-\text{Q1})=(79-73)= 6$$

$$\text{UR}=(\text{Q3}+1.5*\text{IQR})=(79+1.5*6)=(79+9)=88$$

Say some upper values x: 82, 84, 89

Maximum= 84 which is highest value lower than 88.

x	Frequency	C Frequency
3	4	4
4	204	208 (=4+204)
5	3	211
6	84	295
7	0	295
8	103	398



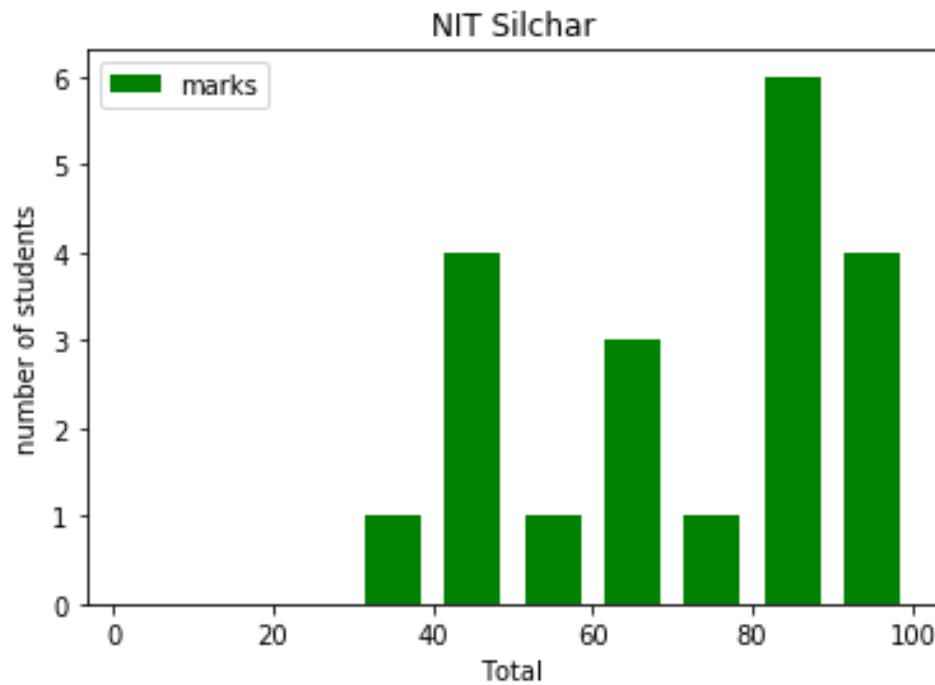
	Fequency/observation	x
Q1	Avg of 99 th and 100th	4
Q2	199	4
Q3	Avg of 298 th and 299th	8
IQR	(Q3-Q1)	4
LR	$\text{Q1}-1.5*\text{IQR}=4-6$	-2
Min		3
UR	$\text{Q3}+1.5*\text{IQR}=8+6$	14
Max		8

It can finds outliers.

Data Visualization and Data Exploration

Data Visualization:

- **Histogram (ranges of data values):** helps in understanding the distribution of a numeric data into series of intervals. Gives us quick understanding of the data.

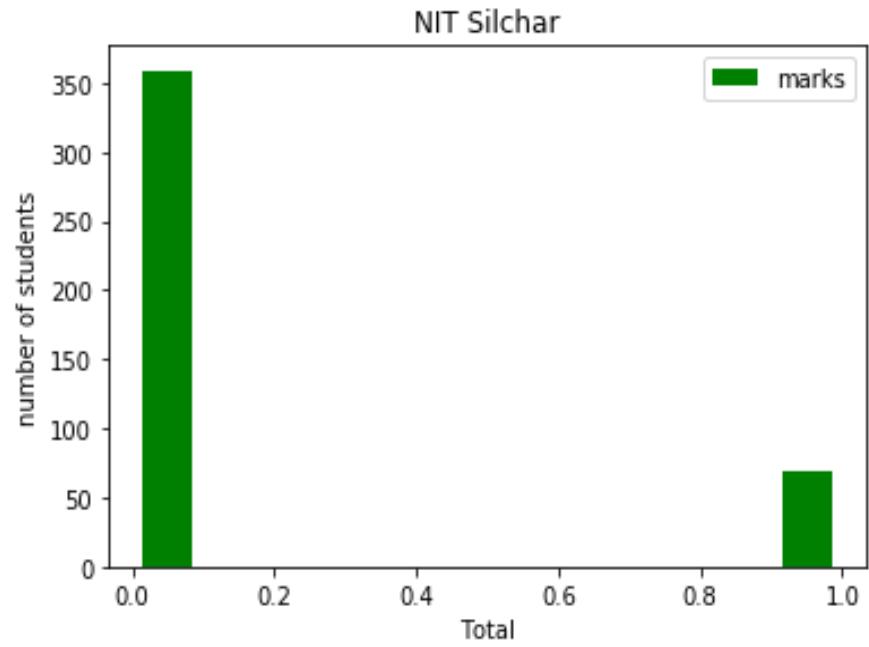
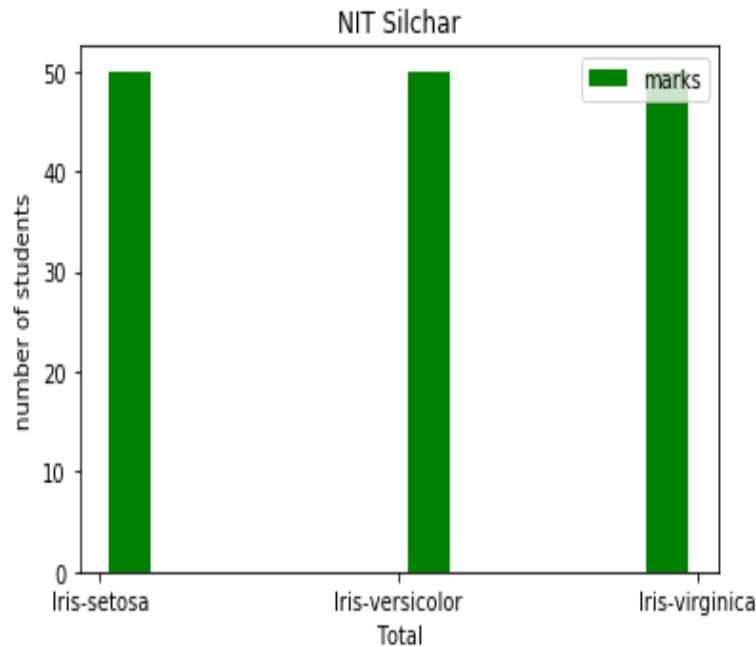


Total
52
45
65
45
68
98
46
88
36
45
86
87
68
86
94
74
93
89
90
86

Data Visualization and Data Exploration

Data Visualization:

- **Histogram:** Gives us quick understanding of the data.

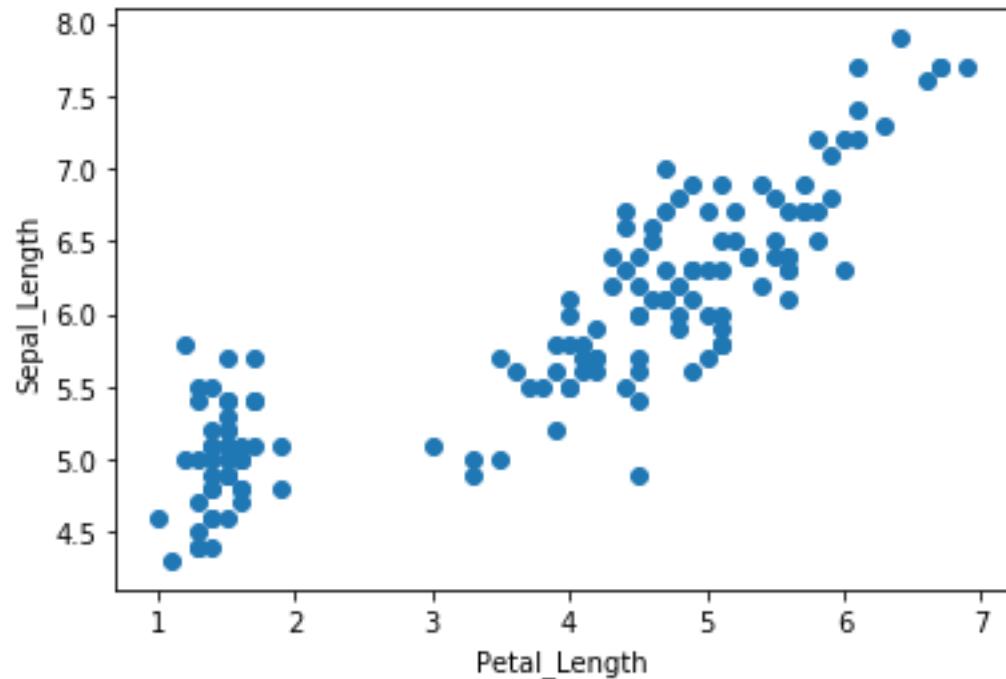


Data Visualization and Data Exploration

Data Visualization:

- Scattered plot: shows relationship between two variables

IRIS DATASET: ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width', 'Species'],



Python Libraries for Machine Learning

1. NumPy (Numerical Python):

- It is array-processing package
- It is used to process large multi-dimensional arrays and matrices
- It is used for handling linear algebra, Fourier transforms, and random numbers.
- Other libraries like TensorFlow uses NumPy at the backend for manipulating tensors.,
- With NumPy, we can define arbitrary data types and easily integrate with most databases.

Some important type of function under NumPy:

i. NumPy **Array Manipulation functions**

```
import numpy
```

```
arr1 = numpy.arange(4)
print('Elements of an array1:\n',arr1)
```

```
res1 = arr1.reshape(2,2)
print('Reshaped array with 2x2 dimensions:\n',res1)
```

Python Libraries for Machine Learning

Some important type of function unders NumPy:

i. NumPy Array Manipulation functions

```
import numpy
```

```
concat = numpy.concatenate((arr1,arr2),axis=1)  
print(concat)
```

ii. NumPy String functions

`numpy.char.add()` function: Concatenates data values of two arrays, merges them and represents a new array as a result.

`numpy.char.capitalize()` function: It capitalizes the first character of the entire word/string.

Python Libraries for Machine Learning

ii. NumPy String functions

numpy.char.lower() function: Converts the case of the string characters to lower string.

numpy.char.upper() function: Converts the case of the string characters to upper string.

numpy.char.replace() function: Replaces a string or a portion of string with another string value.

iii. NumPy Arithmetic functions

numpy.add() function : It adds two arrays and returns the result.

numpy.subtract() function : Subtracts elements of array2 from array1 and returns the result.

numpy.multiply() function : Multiplies the elements of both the arrays and returns the product.

numpy.divide() function : Divides the array1 by array2 and returns the quotient of array values.

numpy.mod() function: Performs modulus operation and returns the remainder array.

numpy.power() function: Returns the exponential value of $\text{array1} ^ \text{array2}$.

Python Libraries for Machine Learning

iv. NumPy Statistical functions

`numpy.median()` : Calculates the median value of the passed array.

`numpy.mean()` : Returns the mean of the data values of the array.

`numpy.average()` : It returns the average of all the data values of the passed array.

`numpy.std()` : Calculates and returns the standard deviation of the data values of the array.

Python Libraries for Machine Learning

2. Pandas: Pandas are turning up to be the most popular Python library that is used for data analysis.

- **The two main types of data structures used by pandas are :** Series (1-dimensional)
- DataFrame (2-dimensional)
- These two put together can handle a vast majority of data: sectors like science, statistics, social, finance, and of course, analytics and other areas of engineering.
- Tabular data with columns of heterogeneous data.
- Ordered and unordered time series data.
- Arbitrary matrix data with the homogeneous or heterogeneous type of data in the rows and columns
- Any other form of statistical or observational data sets.

Some important type of function unders Pandas:

1. `read_csv()`: `read_csv()` function helps read a comma-separated values (csv) file into a Pandas DataFrame. It can also read files separated by delimiters other than comma, like | or tab

2. `head()`: `head(n)` is used to return the first n rows of a dataset. By default, `df.head()` will return the first 5 rows of the DataFrame

Python Libraries for Machine Learning

Some important type of function under Pandas:

3. describe(): **describe() is used to generate descriptive statistics of the data in a Pandas DataFrame or Series.** It summarizes central tendency and dispersion of the dataset. describe() helps in getting a quick overview of the dataset.

4. memory_usage(): memory_usage() returns a Pandas Series having the memory usage of each column (in bytes) in a Pandas DataFrame.

```
data_1.memory_usage(deep=True)
```

5. astype(): astype() is used to cast a Python object to a particular data type.

6. loc[:]: loc[:] helps to access a group of rows and columns in a dataset, a slice of the dataset, as per our requirement.

7. to_datetime(): to_datetime() converts a Python object to datetime format.

8. value_counts(): value_counts() returns a Pandas Series containing the counts of unique values.

9. drop_duplicates(): drop_duplicates() returns a Pandas DataFrame with duplicate rows removed.

```
data_1.drop_duplicates(inplace=True)
```

10. groupby(): groupby() is used to group a Pandas DataFrame by 1 or more columns, and perform some mathematical operation on it.

Python Libraries for Machine Learning

Some important type of function under Pandas:

```
data_1.groupby(by='State').Salary.mean()
```

11. merge(): merge() is used to merge 2 Pandas DataFrame objects or a DataFrame and a Series object on a common column (field)

12. sort_values(): sort_values() is used to sort column in a Pandas DataFrame (or a Pandas Series) by values in ascending or descending order.

```
data_1.sort_values(by='Name', inplace=True)
```

13. fillna(): fillna() helps to replace all NaN values in a DataFrame or Series by imputing these missing values with more appropriate values.

```
data_1['City temp'].fillna(38.5, inplace=True)
```

14. Shape: property will return a tuple of the shape of the data frame.

```
f1.shape
```

15. f1.columns: will give you the column values

16. f1.tail():

17. DataFrame.info(): Pandas **dataframe.info()** function is used to get a concise summary of the dataframe.

Python Libraries for Machine Learning

Some important type of function under Pandas:

18. **dtypes: (f1.dtype)** dtypes shows the data type of each column. (f1.dtype)
19. **Size: (f1.size)** Size, as the name suggests, returns the size of a dataframe which is the number of rows multiplied by the number of columns.
20. **Sample: (f1.sample(n=8))** Sample method allows you to select values randomly from a Series or DataFrame.
21. **isnull:(f1.isnull())** To handle missing values
22. **isna() : (f1.isna().any())** Isna function returns a dataframe filled with boolean values with true indicating missing values.
23. **f1.isnull().sum()** : We can calculate the number of missing values in each column
24. **nunique(): (f1. nunique())** Nunique counts the number of unique entries over columns or rows. It is very useful in categorical features especially in cases where we do not know the number of categories beforehand
25. **index() (f1.index)** searches for a given element from the start of the list and returns the lowest index where the element appears.
26. **nsmallest() (f1. nsmallest(5,'Sepal_Width'))** finds the 5 observations with the smallest value
27. **nlargest() (f1. nlargest(5,'Sepal_Width'))** finds the 5 observations with the Largest value.

Python Libraries for Machine Learning

Some important type of function unders Pandas:

28. Loc and iloc

Loc and iloc are used to select rows and columns.

loc: select by labels

iloc: select by positions

```
f1.loc[:5,['Sepal_Length', 'Sepal_Width']]  
f1.iloc[:5,:6]
```

29. Slicing: Slicing Rows and Columns using labels.

```
f1[0:4]
```

30. **dropna ()** function is used to remove a row or a column from a dataframe which has a NaN or no values in it

31. **query()**: We sometimes need to filter a dataframe based on a condition or apply a mask to get certain values.

```
f1.query('3000<median_value<1000')[:4]
```

32. **insert()** : offers the option to add the new column in any position using **insert** function

```
f1.insert(5, 'new_name', new_col)
```

Python Libraries for Machine Learning

3. Matplotlib:

- Matplotlib is a data visualization library
- It is used for 2D plotting to produce publication-quality image plots and figures in a variety of formats.
- The library helps to generate histograms, plots, error charts, scatter plots, bar charts with just a few lines of code.

4. SciPy (Scientific Python):

This is a python library for **machine learning**, especially for scientific and analytical computing.

- SciPy uses **multi-dimensional array** provided by the NumPy module.
- SciPy depends on NumPy for the array manipulation subroutines.
- The SciPy library offers **modules for linear algebra, image optimization, integration, interpolation, special functions, Fast Fourier transform, signal and image processing, Ordinary Differential Equation (ODE) solving, and other computational tasks in science and analytics.**

5. Scikit-learn:

It has become the most popular Python machine learning library for developing machine learning algorithms.

Python Libraries for Machine Learning

6. Scikit-learn:

- The library can be used for **data-mining and data analysis**.
- The main machine learning functions that the Scikit-learn library can handle are **classification, regression, clustering, dimensionality reduction, model selection, and preprocessing**.

7. Theano:

- Theano is a **python machine learning library** that can act as an optimizing compiler for evaluating and manipulating mathematical expressions and matrix calculations.
- It is built on NumPy.
- **Theano can work on Graphics Processing Unit (GPU) and CPU**.
- Theano has built-in tools for unit-testing and validation, thereby avoiding bugs and problems.

8. TensorFlow:

TensorFlow is a popular computational framework for creating **machine learning models**.

- TensorFlow has a flexible architecture with which it can run on a variety of **computational platforms CPUs, GPUs, and TPUs**.
- TPU stands for Tensor processing unit, a hardware chip built around TensorFlow for machine learning and artificial intelligence.

Python Libraries for Machine Learning

9.Keras: Keras is an open-source library used for **neural networks and machine learning**. Keras can run on top of TensorFlow, Theano etc.

- Keras works with neural-network building blocks like layers, objectives, activation functions, and optimizers.
- Keras also have a bunch of features **to work on images and text images that comes handy when writing Deep Neural Network code**.
- Keras supports convolutional and recurrent neural networks.

10. PyTorch: PyTorch has a range of tools and libraries that support **computer vision, machine learning, and natural language processing**

- PyTorch can smoothly integrate with the python data science stack, including NumPy.
- We will hardly make out a difference between NumPy and PyTorch.
- PyTorch include multi GPU support, simplified preprocessors, and custom data loaders.

11. Neurolab: **It is a simple and powerful Neural Network Library for Python.** It is a library for basic neural networks algorithms with flexible network configurations and learning algorithms for Python.

Python Module and Packages

Modules: Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*. A module is a file containing Python definitions and statements. **The file name is the module name with the suffix .py appended.**

Packages:

Packages are a way of structuring Python's module namespace by using "dotted module names". For example, **the module name A.B designates a submodule named B in a package named A.**

```
import sound.effects.echo
```

This loads the submodule **echo** from package **sound.effects**. It must be referenced with its full name.

An alternative way of importing the submodule is:

```
from sound.effects import echo
```

This also loads the submodule **echo**, and makes it available without its package prefix, so it can be used as follows:

```
echo.echofilter(input, output, delay=0.7, atten=4)
```

```
from sound.effects.echo import echofilter
```

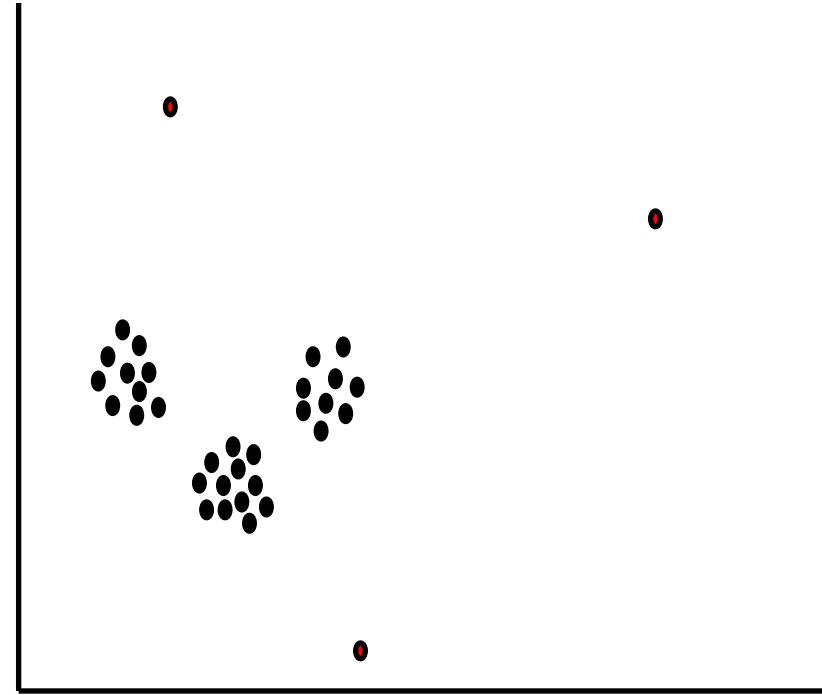
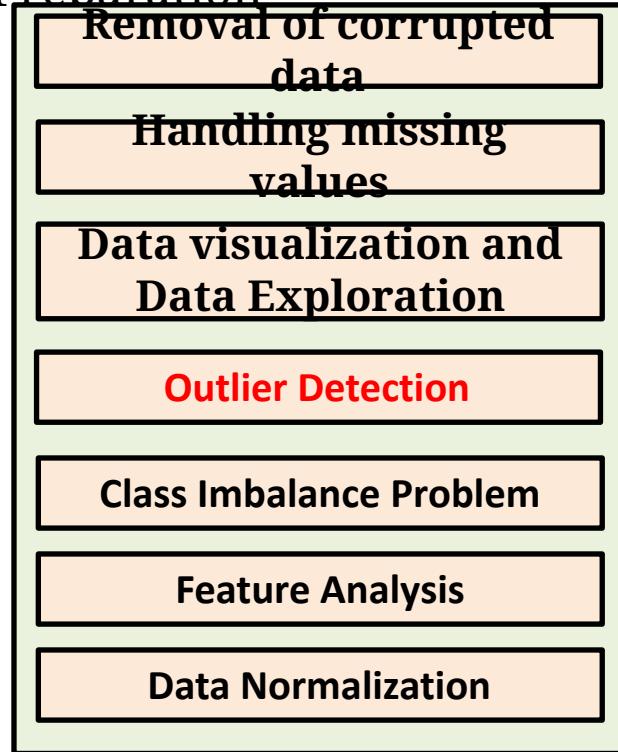
Again, this loads the submodule **echo**, but this makes its function **echofilter()** directly available:

```
echofilter(input, output, delay=0.7, atten=4)
```

DAY-2

Machine Learning Model

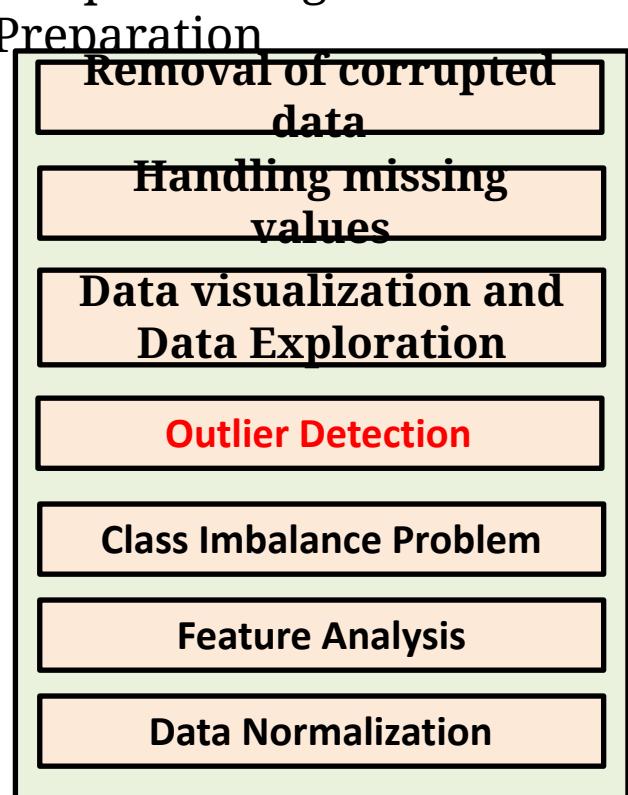
Pre-processing/ Data Preparation



Far from the rest of the observations or the center of mass of observations.

Outlier Detection

Pre-processing/ Data Preparation



Can result in a poor fit and lower predictive modelling performance.

Can fit the data properly and enhance the performance of the model

Histogram or scatter plot can be used for one or two dimensional data.

For high dimensional data, simple statistical methods for identifying outliers can break down.

Outlier Detection

Many techniques are found to detect outlier. Based on learning style, these techniques can be categorized into three classes:

- supervised outlier detection techniques,
- unsupervised outlier detection techniques.
- Semi-supervised outlier detection techniques.

Based on learning measures, these techniques can be categorized into four approaches

- Statistical based
- Distance based
- Density based
- Tree-based

LIMITATIONS OF STATISTICAL BASED APPROACH

- Works well for a single attribute
- In many cases, data distribution may not be known. However statistical based approach needs to know data distribution.
- For multi-dimensional data, it may be difficult to estimate the true distribution

Outlier Detection

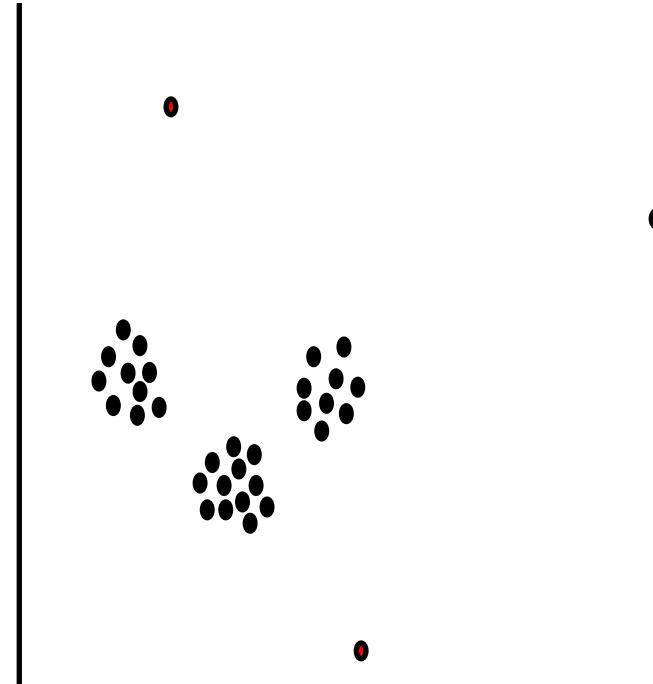
Local Outlier Factor: Each example is assigned a scoring of how isolated or how likely it is to be outliers based on the size of its local neighborhood. Examples with the largest score are more likely to be outliers. (density based unsupervised algorithm)

Isolation Forest: is a tree-based anomaly detection algorithm:

Outliers have attribute-values that are very different from those of normal instances. (Tree based unsupervised algorithm)

One Class SVM: can be used to discover outliers in input data for both regression and classification datasets. This is specially used in imbalanced dataset. (density based unsupervised algorithm)

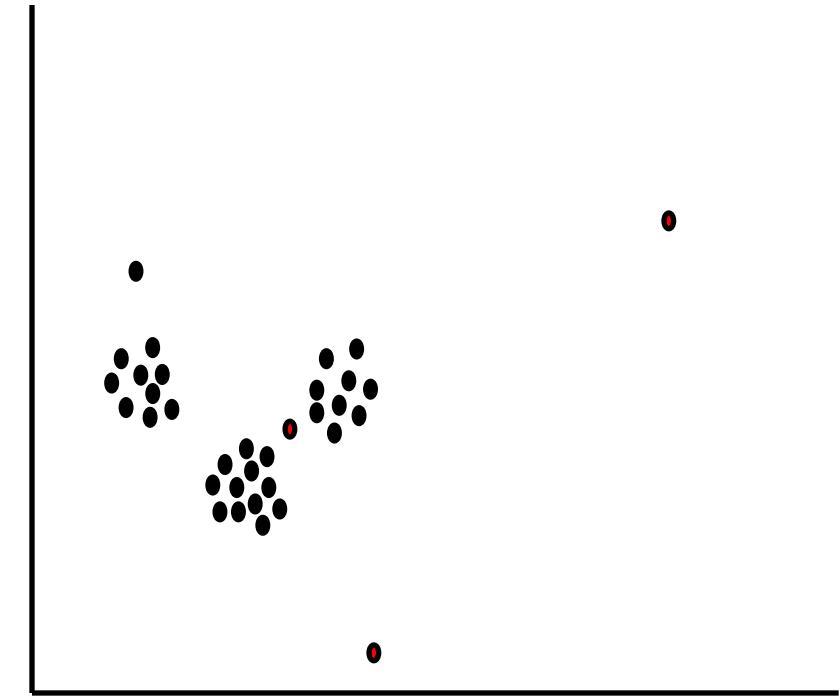
Minimum Covariance Determinant: If the input variables have a Gaussian distribution, then this simple statistical methods can be used to detect outliers. (Statistical based unsupervised algorithm)



Local and Global Outlier

Outliers can be local outlier and global outlier.

Global outlier can be found using distance measure; however local outlier can be found using density based measure



Local Outlier Factors (LOF)

The concept of LOF is based on the statistics of K-Nearest Neighbours (K-NN).
Need to calculate reachability distance.
Need to find LOF score for all the instances.

Kth distance of A [dist-k(A)]= Distance between A and its k-nearest neighbour = DA (k=3)

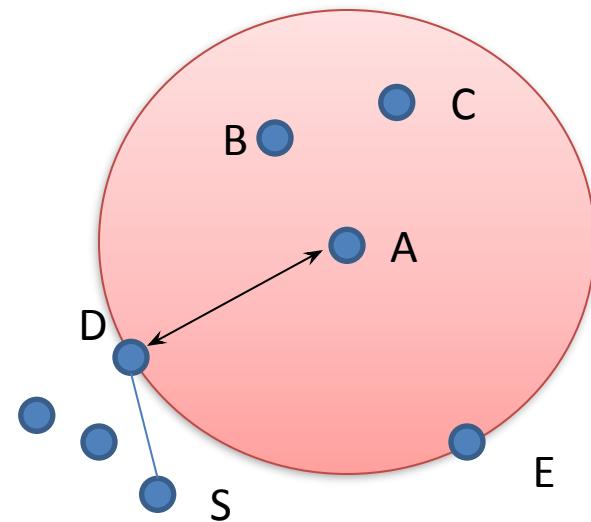
K distance neighbour of A=

$$\begin{aligned} K\text{-dn}(A) &= \{A' | A' \in D, \text{dist}(A, A') \leq \text{dist} - k(A)\} \\ &= \{B, C, D, E\} \end{aligned}$$

Reach-Dist(A,D) = $\max\{\text{Kth-distance}(D), d(A,D)\}$,
= $\max(DS, AD) = AD$

where Kth-distance(D) is the distance of instance D from its Kth nearest neighbor=DS

$d(A,D)$ is the actual distance between A and D=
AD



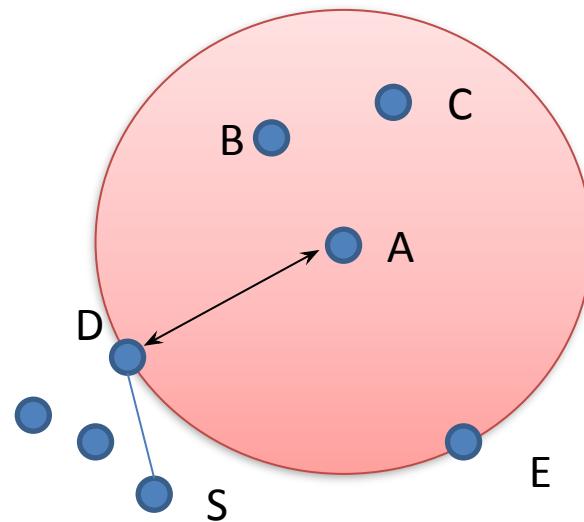
Local Outlier Factors (LOF)

Average RD_A=

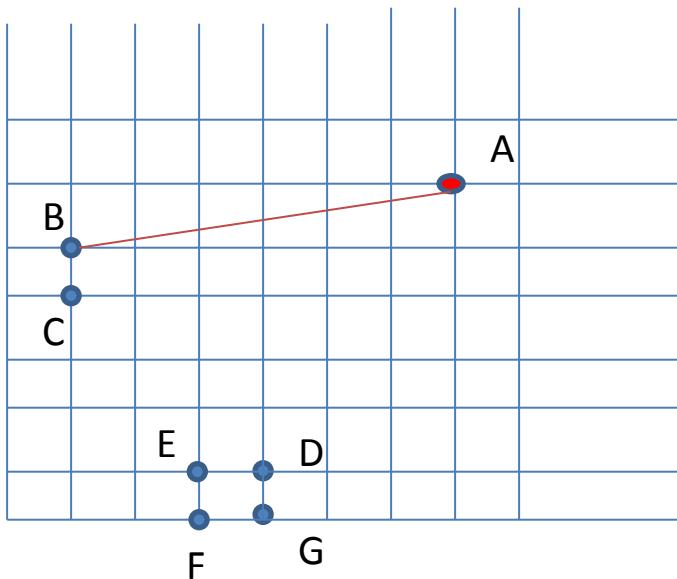
$$\frac{1}{k-dn} \sum_{k-dn} \max[(k^{th} \text{ distance of } A's \text{ neighbor}, \epsilon]$$

=

$$\frac{1}{4} \sum_4 \max[(3^{th} \text{ distance of } A's \text{ neighbor}, distan$$



Local Outlier Factors (LOF)



$$\begin{aligned}
 AB^2 &= 6^2 + 1^2 \\
 &= 36+1 \\
 &= 37 \\
 AB &= 6.08
 \end{aligned}$$

Kth distance of A
[dist-k(A)] = AC = 6.32

K distance neighbour
of A = {B, C, D}

	A1	A2
A	7	6
B	1	5
C	1	4
D	4	1
E	3	1
F	3	0
G	4	0

Distances	
AB	6.08
AC	6.32
AD	5.8
AE	6.4
AF	7.2
BA	6.08
BC	1
BD	5
BE	4.4
BF	5.3
CA	6.32
CB	1
CD	4.2
CE	3.6
CF	4.4
DA	5.8
DB	5
DC	4.2
DE	1
DF	1.4
DG	1

Local Outlier Factors (LOF)

Average $RD_A =$

$$\frac{1}{3} \sum_3 \max[(3^{\text{th}} \text{ distance of } A's \text{ neighbor}, \text{distance}(A, \text{ the neighbor}))]$$

$$= \frac{1}{3} [\max(3^{\text{th}} \text{ dist } B, \text{dist}(A, B)) + \max(3^{\text{th}} \text{ dist } C, \text{dist}(A, C)) + \max(3^{\text{th}} \text{ dist } D, \text{dist}(A, D))]$$

$$= \frac{1}{3} [\max(5, 6.08) + \max(4.2, 6.32) + \max(1.4, 5.8)]$$

$$= \frac{1}{3} [6.08 + 6.32 + 5.8] = 6.06$$

Density is reverse of distance therefore **Local Reachability score LRD**

$$\begin{aligned} LRD_A &= \frac{1}{RD_A} \\ &= 1/6.06 = 0.165 \end{aligned}$$

Distances	
AB	6.08
AC	6.32
AD	5.8
AE	6.4
AF	7.2
BA	6.08
BC	1
BD	5
BE	4.4
BF	5.3
CA	6.32
CB	1
CD	4.2
CE	3.6
CF	4.4
DA	5.8
DB	5
DC	4.2
DE	1
DF	1.4
DG	1

Local Outlier Factors (LOF)

Similarly calculated,

Average $RD_B = 5.17$ and $LRD_B = 0.193$

Average $RD_C = 5.17$ and $LRD_C = 0.193$

Average $RD_D = 5.17$ and $LRD_D = 0.193$

$$LOF_A = \frac{\frac{1}{3}(LRD_B + LRD_C + LRD_D)}{LRD_A} = \frac{\frac{1}{3}(0.193 + 0.193 + 0.193)}{0.165} = \frac{0.193}{0.165} = 1.17$$

Generally, if $LOF > 1$, it is considered as an outlier, but that is not always true.

Distances	
AB	6.08
AC	6.32
AD	5.8
AE	6.4
AF	7.2
BA	6.08
BC	1
BD	5
BE	4.4
BF	5.3
CA	6.32
CB	1
CD	4.2
CE	3.6
CF	4.4
DA	5.8
DB	5
DC	4.2
DE	1
DF	1.4
DG	1

Isolation Forest (IF)

- It's an unsupervised learning algorithm that identifies anomaly by isolating outliers in the data.
- Isolation forests are an effective method for detecting outliers or novelties in data.
- It is a relatively novel method based on binary decision trees.
- Isolation forest's basic principle is that outliers are few and far from the rest of the observations.
- It can work for large datasets with one or multi dimensional feature space.

Isolation Forest (IF)

Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that selected feature. This split depends on how long it takes to separate the points.

Random partitioning produces noticeably shorter paths for anomalies. When a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies.

- An outlier score can be computed for each observation:

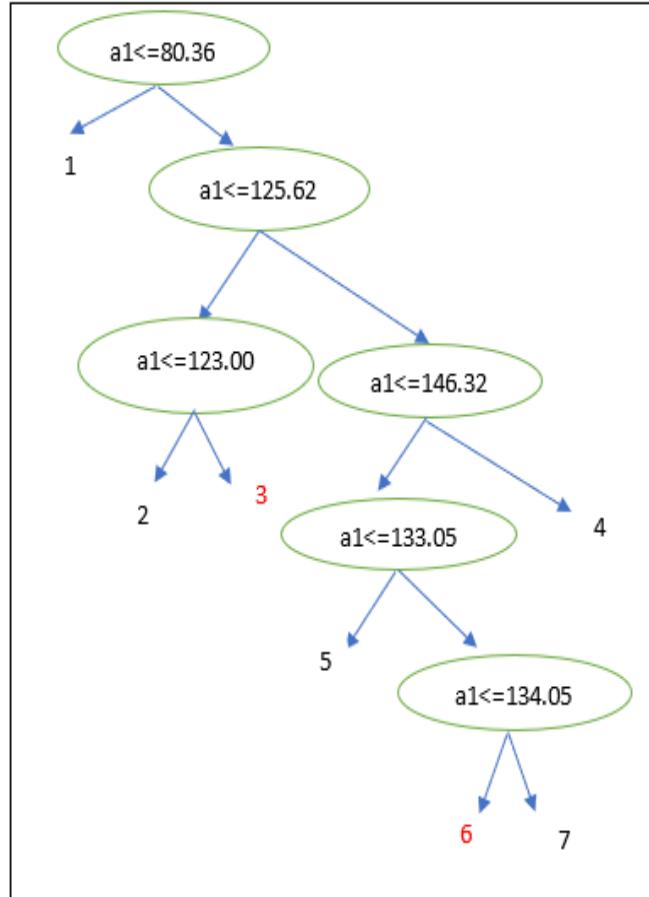
$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

outlier score

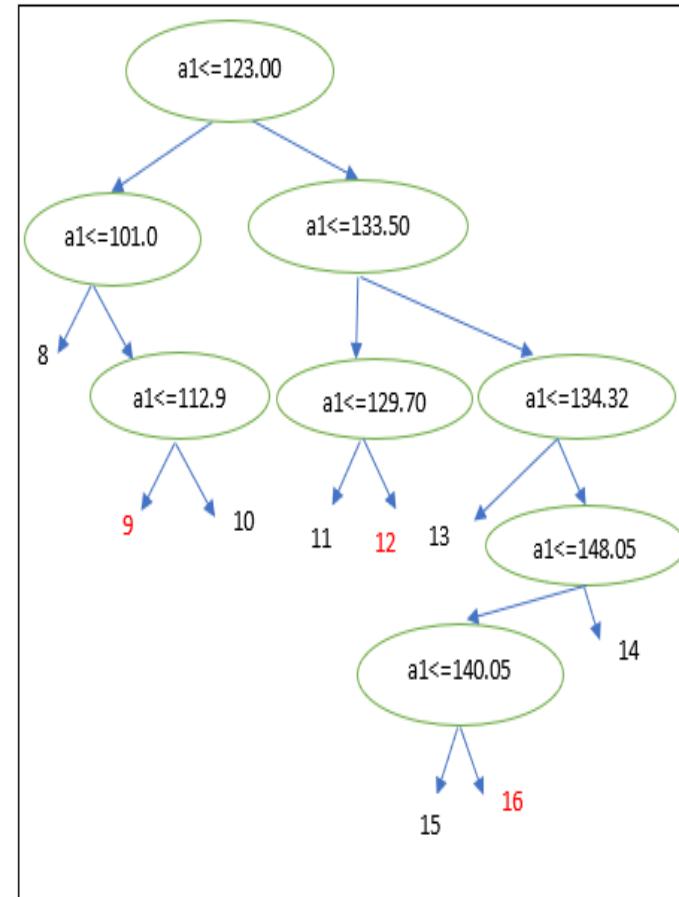
- Where $h(x)$ is the path length of the sample x , and $c(n)$ is the ‘unsuccessful length search’ of a binary tree (the maximum path length of a binary tree from root to external node) n is the number of external nodes. After giving each observation a score ranging from 0 to 1; 1 meaning more outlyingness and 0 meaning more normality. A threshold can be specified (ie. 0.55 or 0.60)

Isolation Forest (IF)

Sl no.	a1	Leaf node
1	15 0	4, 14
2	12 2	2, 10
3	13 5	7, 15
4	13 2	5, 13
5	4	1, 8
6	13 8	7, 15
7	12	2, 10
8	set 12 7	5, 11



Decision
tree 1



Decision
tree 2

ISOLATION FOREST

Node no. 3,6,9,12,16 are unsuccessful path

- The average unsuccessful path length is given by node 6 and 16

which is 5. Since 2 trees are considered therefore $h(x)$ is the

average path of the datapoint for the 2 trees.

$$s_1 = 2^{-\left(\frac{3+4}{2}\right)} = 2^{-(0.7)} = 0.615$$

$$s_2 = 2^{-\left(\frac{3+3}{2}\right)} = 2^{-(0.6)} = 0.65$$

$$s_3 = 2^{-\left(\frac{5+5}{2}\right)} = 2^{-(1)} = 0.5$$

$$s_4 = 2^{-\left(\frac{4+3}{2}\right)} = 2^{-(0.7)} = 0.615$$

$$s_5 = 2^{-\left(\frac{1+2}{2}\right)} = 2^{-(0.3)} = 0.812$$

$$s_6 = 2^{-\left(\frac{5+5}{2}\right)} = 2^{-(1)} = 0.5$$

$$s_7 = 2^{-\left(\frac{3+3}{2}\right)} = 2^{-(0.6)} = 0.65$$

$$s_8 = 2^{-\left(\frac{4+3}{2}\right)} = 2^{-(0.7)} = 0.615$$

Score= $s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$

Sl n o.	weig ht	sco res
1	150	0.615
2	122	0.65
3	135	0.50
4	132	0.615
5	4	0.812
6	138	0.50
7	121	0.65
8	127	0.615

one anomaly is detected

a1	scor es
4	0.812

ISOLATION FOREST

Isolation Forest pros:

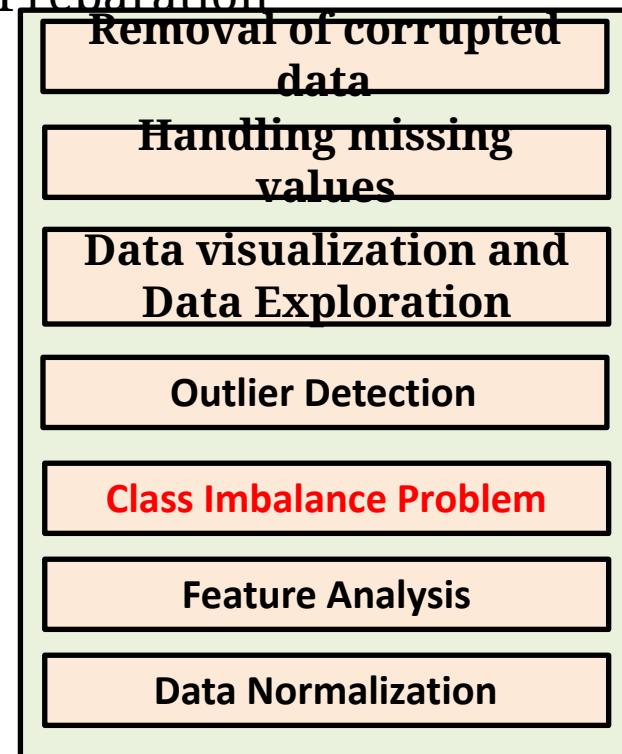
- There is no need of scaling the values in the feature space.
- It is an effective method when value distributions can not be assumed.
- It has few parameters, this makes this method fairly robust and easy to optimize.

Isolation Forest cons:

- Visualizing results is complicated.
- Sometimes, training time can be very long and computationally expensive.

Machine Learning Model

Pre-processing/ Data Preparation

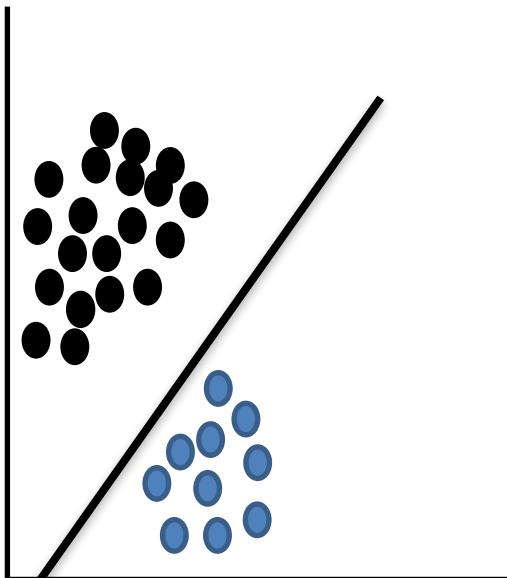


Class imbalance Problem

Class Imbalance: Biases towards majority class(es)

Approaches: Sampling, Algorithmic,
Ensembling

Sampling: Undersampling & oversampling



Undersampling: reduces majority class instances: examples RUS, Tomek link based undersampling, Condensed Nearest Neighbor (CNN) undersampling etc.

Oversampling: increases minority class instances: examples ROS, SMOTE, Border-Line SMOTE, Safe level SMOTE, ADASYN etc.

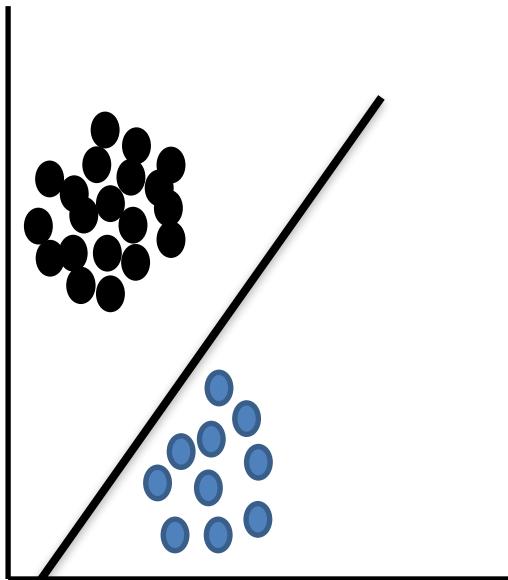
[\[Gaussian SMOTE\]](#) (2017)

[\[kmeans SMOTE\]](#) (2018)

Algorithmic: changes the cost function, higher misclassification cost for minority class cost-sensitive neural network etc.

Ensembling: SMOTEBoost, RUSBoost etc

Class imbalance Problem



RUS: Random Under Sampling technique is random undersampling of the majority class.

This can potentially lead to loss of information about the majority class.

However, in cases where each example of the majority class is near other examples of the same class, this method might yield good results.

$$R = S/L = 0.5$$

$$S = 500$$

$$L = 1500$$

$$L = 1000 \text{ (after undersampling)}$$

Distance Measure

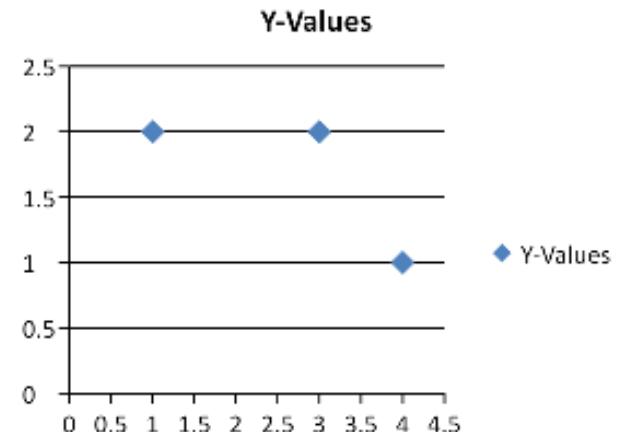
Let $p = (p_1, p_2)$ and $q = (q_1, q_2)$ be two points:

- ✓ City block distance $d(p, q) = |p_1 - q_1| + |p_2 - q_2|$
- ✓ Euclidean distance $d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$
- ✓ Minkowski distance $d(p, q) = (\sum_{i=1}^M |p_i^n - q_i^n|^r)^{\frac{1}{r}}$

For $r=1$, Minkowski distance = City block distance

For $r=2$, Minkowski distance = Euclidean distance

	M=1	M=2
1 st	1	2
2 nd	3	2
3 rd	4	1



$$1^{\text{st}} = (1, 2)$$

$$2^{\text{nd}} = (3, 2)$$

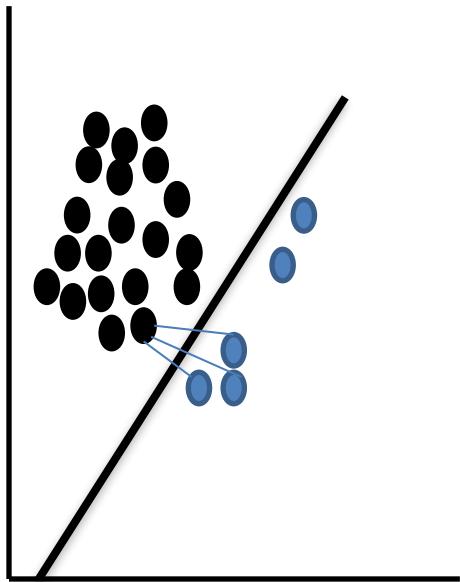
$$3^{\text{rd}} = (4, 1)$$

$$\text{Dis}(1^{\text{st}}, 2^{\text{nd}}) = |1-3| + |2-2| = 2$$

Undersampling

NearMiss-1: Those points from L are retained whose mean distance to the k nearest points in S is lowest.

where k is a tunable hyperparameter

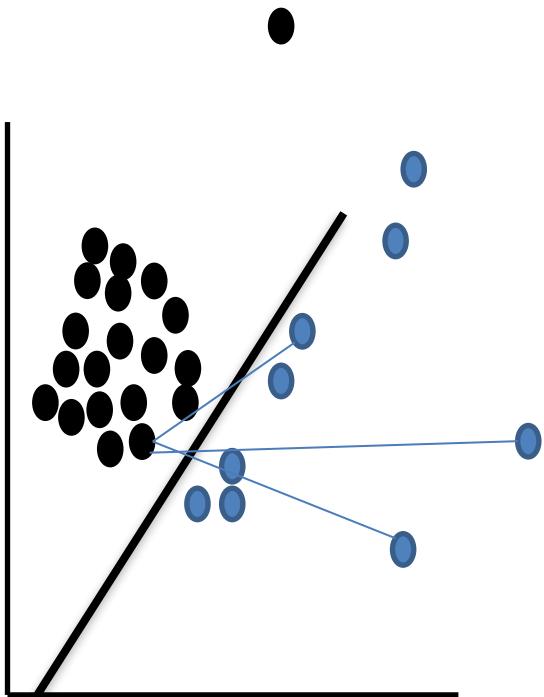


R= S/L=0.5 (As we wish)

$$s_1 = \frac{l_1 + l_2 + l_3}{3} = x_1$$

8, 6, 4, 5, 1, 2, 3, 7, 9, 10, 12, 16, 17, 18, 20, 19, 15, 14, 13, 11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Undersampling



NearMiss-2: keeps those points from L whose mean distance to the k farthest points in S is lowest.

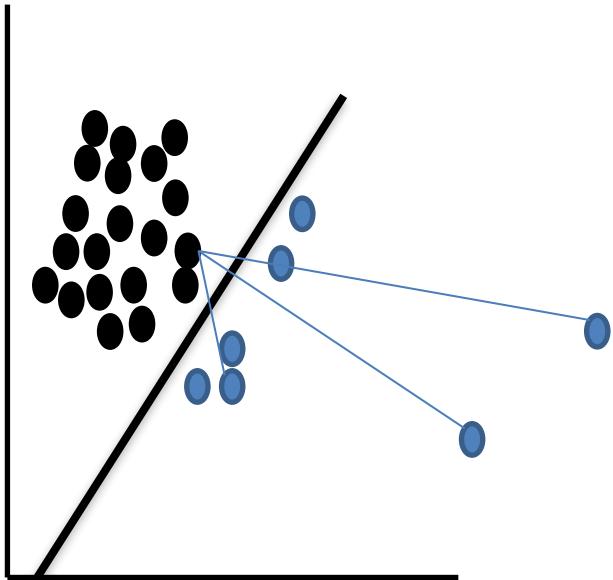
where k is a tunable hyperparameter

$R = S/L = 0.5$ (As we wish)

$$s_1 = \frac{f_1 + f_2 + f_3}{3} = x_1$$

8, 6, 4, 5, 1, 2, 3, 7, 9, 10, 12, 16, 17, 18, 20, 19, 15, 14, 13, 11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Undersampling



NearMiss-2: keeps those points from L whose mean distance to the k farthest points in S is lowest.

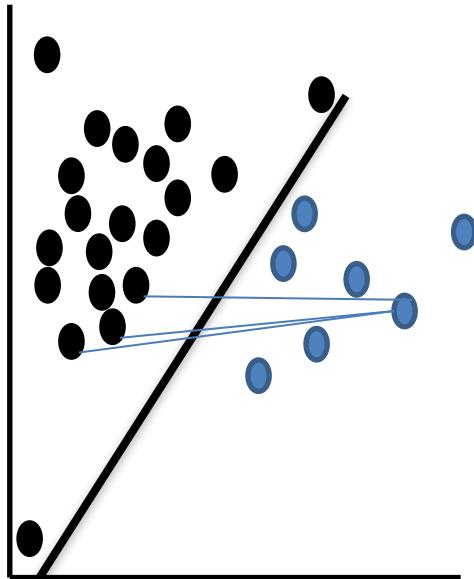
where k is a tunable hyperparameter

R= S/L=0.5 (As we wish)

$$s_1 = \frac{f_1 + f_2 + f_3}{3} = x_1$$

8, 6, 4, 5, 1, 2, 3, 7, 9, 10, 12, 16, 17, 18, 20, 19, 15, 14, 13, 11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

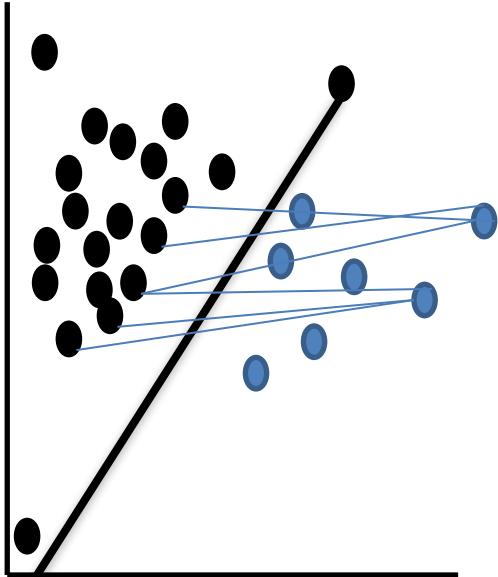
Undersampling



NearMiss-3: selects k nearest neighbors in L for every point in S . In this case, the undersampling ratio is directly controlled by k and is not separately tuned.

where k is a tunable hyperparameter

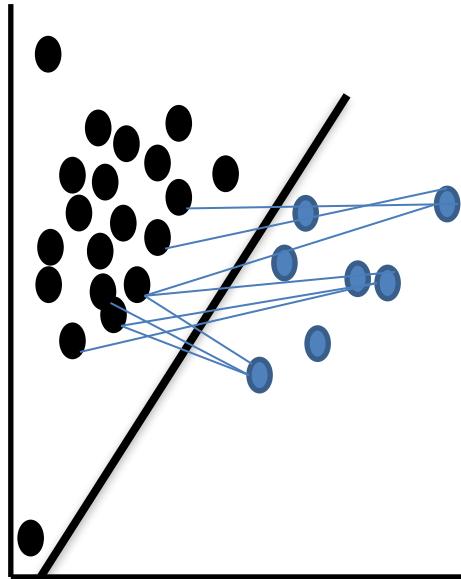
Undersampling



NearMiss-3: selects k nearest neighbors in L for every point in S . In this case, the undersampling ratio is directly controlled by k and is not separately tuned.

where k is a tunable hyperparameter

Undersampling

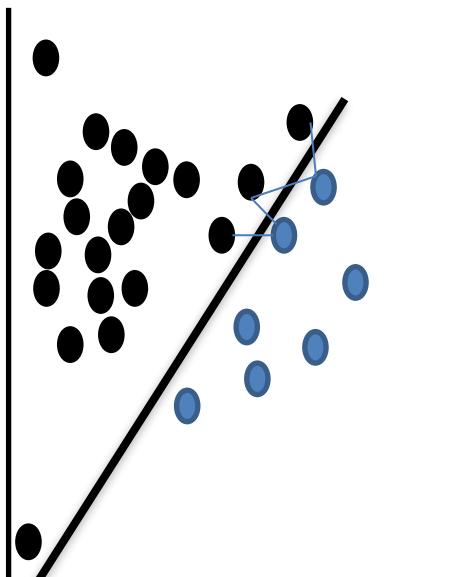


NearMiss-3: selects k nearest neighbors in L for every point in S . In this case, the undersampling ratio is directly controlled by k and is not separately tuned.

where k is a tunable hyperparameter

Undersampling

Condensed Nearest Neighbor (CNN): the goal is to choose a subset U of the training set T such that for every point in T its nearest neighbor in U is of the different class. U can be grown iteratively as follows:



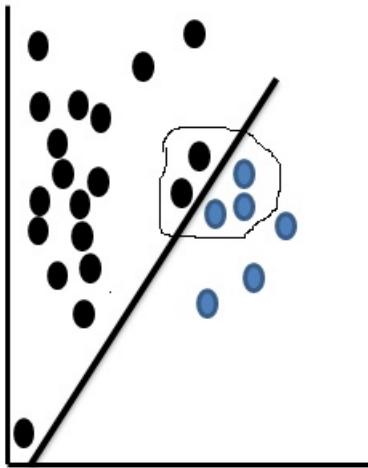
1. Select a random point from T and set $U = \{p\}$.
2. Scan $T - U$ and add to U the first point found whose nearest neighbor in U is of a different class
3. Repeat step 2 until U is maxima

Undersampling via CNN can be slower compared to other methods since it requires many passes over the training data.

Further, because of the randomness involved in the selection of points at each iteration, the subset selected can vary significantly.

A variant of CNN is to only undersample L i.e. retain all points from S but retain only those points in L that belong to U .

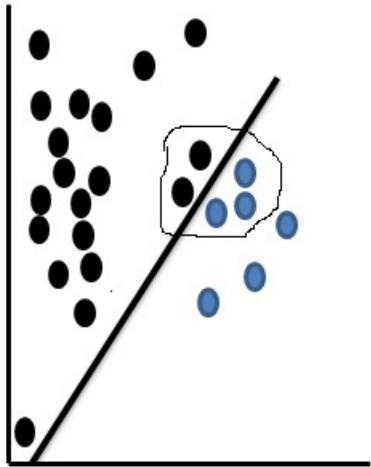
Undersampling



Edited Nearest Neighbor (ENN):
undersampling of the majority class is done by removing points whose class label differs from a majority of its k nearest neighbors.

Say $k=4$

Undersampling

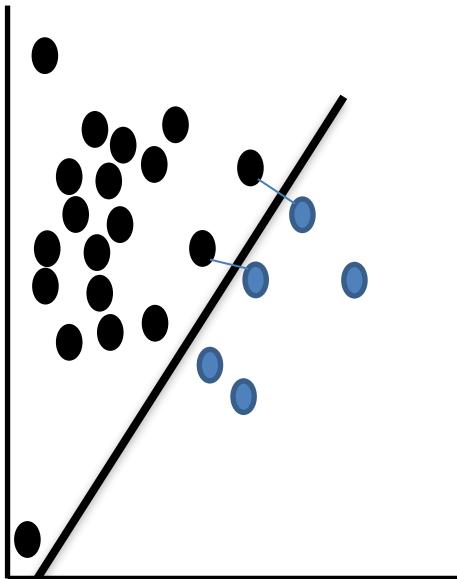


Repeated Edited Nearest Neighbor:

The ENN algorithm is applied successively until ENN can remove no further points.

Say $k=4$

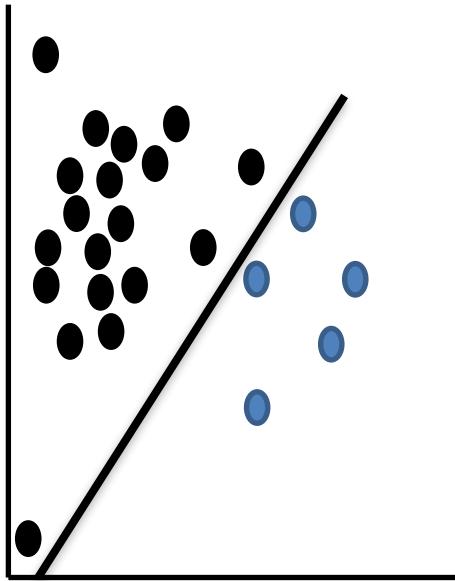
Undersampling



Tomek Link Removal:

- A pair of examples is called a Tomek link if they belong to different classes and are each other's nearest neighbors.
- Undersampling can be done by removing all tomek links from the dataset.
- An alternate method is to only remove the majority class samples that are part of a Tomek link.

Oversampling



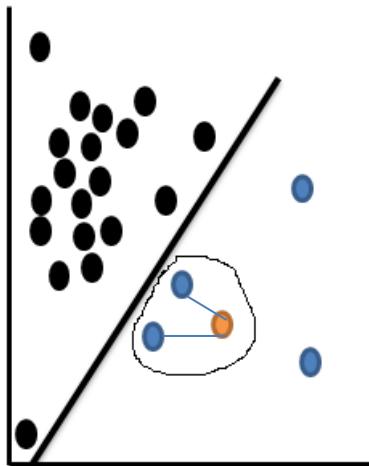
Random oversampling of minority class:

- Points from the minority class may be oversampled randomly.
- This method is prone to overfitting.
- We consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

SMOTE:

A more sophisticated means for oversampling is Synthetic Minority Oversampling Technique (SMOTE)



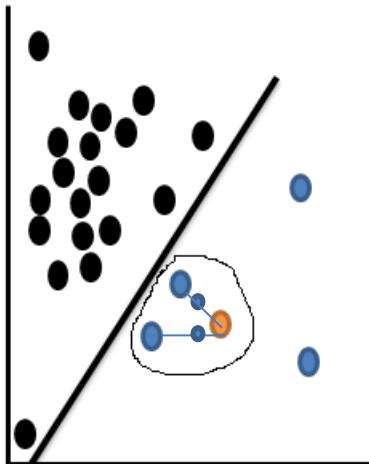
For each point p in S :

1. Compute its k nearest neighbors in S .
 2. Randomly choose $r \leq k$ of the neighbors (with replacement).
 3. Choose a random point along the lines joining p and each of the r selected neighbors.
 4. Add these synthetic points to the dataset with class S .
-
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

SMOTE:

A more sophisticated means for oversampling is Synthetic Minority Oversampling Technique (SMOTE)



For each point p in S :

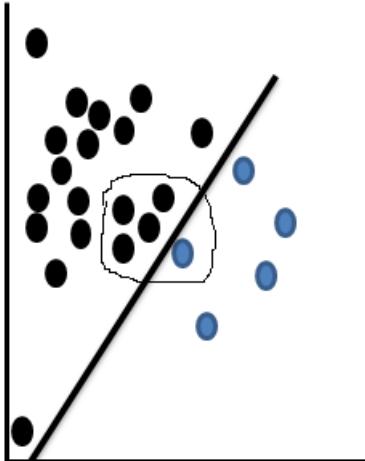
1. Compute its k nearest neighbors in S .
 2. Randomly choose $r \leq k$ of the neighbors (with replacement).
 3. Choose a random point along the lines joining p and each of the r selected neighbors.
 4. Add these synthetic points to the dataset with class S .
-
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

BORDERLINE SMOTE-1:

There are two enhancements of SMOTE, termed borderline SMOTE, which may yield better performance than SMOTE.

For each point p in S :



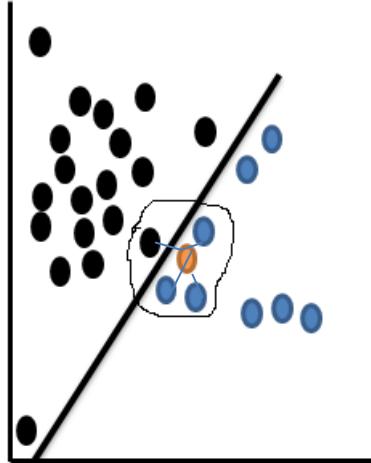
1. Compute its m nearest neighbors in T . Call this set M_p and let $m_0 = |M_p \cap L|$.
 2. If $m_0 = m$, p is a noisy example. Ignore p and continue to the next point.
 3. If $0 \leq m_0 \leq m/2$, p is safe. Ignore p and continue to the next point.
 4. If $m/2 < m_0 < m$, add p to the set DANGER.
 6. For each point d in DANGER, apply the SMOTE algorithm to generate synthetic examples.
-
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

Borderline-SMOTE1:

There are two enhancements of SMOTE, termed borderline SMOTE, which may yield better performance than SMOTE.

For each point p in S :



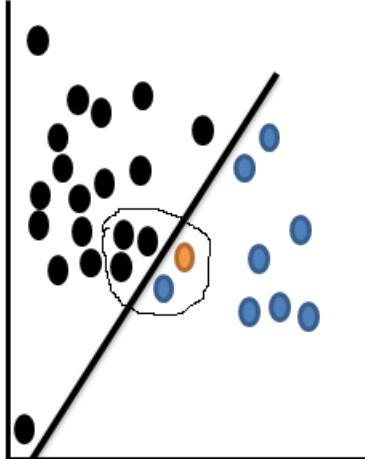
1. Compute its m nearest neighbors in T . Call this set M_p and let $m_0 = |M_p \cap L|$.
 2. If $m_0 = m$, p is a noisy example. Ignore p and continue to the next point.
 3. If $0 \leq m_0 \leq m/2$, p is safe. Ignore p and continue to the next point.
 4. If $m/2 < m_0 < m$, add p to the set DANGER.
 6. For each point d in DANGER, apply the SMOTE algorithm to generate synthetic examples.
- We may consider the result of oversampling of S to achieve $\pi_S > 0.5$.

Oversampling

Borderline-SMOTE1:

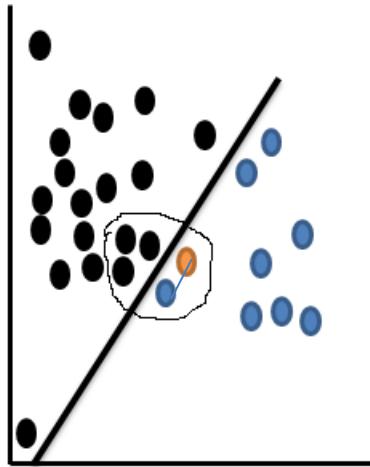
There are two enhancements of SMOTE, termed borderline SMOTE, which may yield better performance than SMOTE.

For each point p in S :



1. Compute its m nearest neighbors in T . Call this set M_p and let $m_0 = |M_p \cap L|$.
 2. If $m_0 = m$, p is a noisy example. Ignore p and continue to the next point.
 3. If $0 \leq m_0 \leq m/2$, p is safe. Ignore p and continue to the next point.
 4. If $m/2 < m_0 < m$, add p to the set DANGER.
 6. For each point d in DANGER, apply the SMOTE algorithm to generate synthetic examples.
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

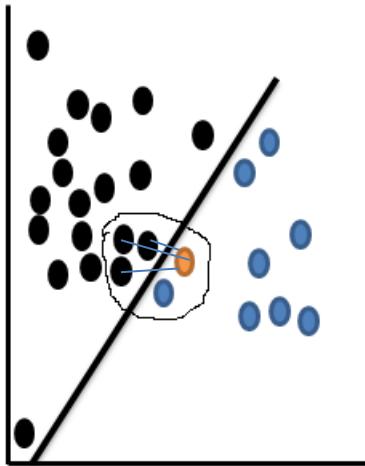


Borderline-SMOTE2:

Borderline-SMOTE2 is similar to Borderline-SMOTE1 except in the last step, new synthetic examples are created along the line joining points in DANGER to either their nearest neighbors in S or their nearest neighbors in L.

- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling



Borderline-SMOTE2:

Borderline-SMOTE2 is similar to Borderline-SMOTE1 except in the last step, new synthetic examples are created along the line joining points in DANGER to either their nearest neighbors in S or their nearest neighbors in L.

- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling + Undersampling

Combination methods

Performing a combination of oversampling and undersampling can often yield better results than either in isolation.

SMOTE + Tomek Link Removal

	$ L $	$ S $
Before resampling	6320	680
After resampling	6050	3160

SMOTE + ENN

	$ L $	$ S $
Before resampling	6320	680
After resampling	4894	3160

Treatment of Class imbalance Problem

Algorithmic level solution

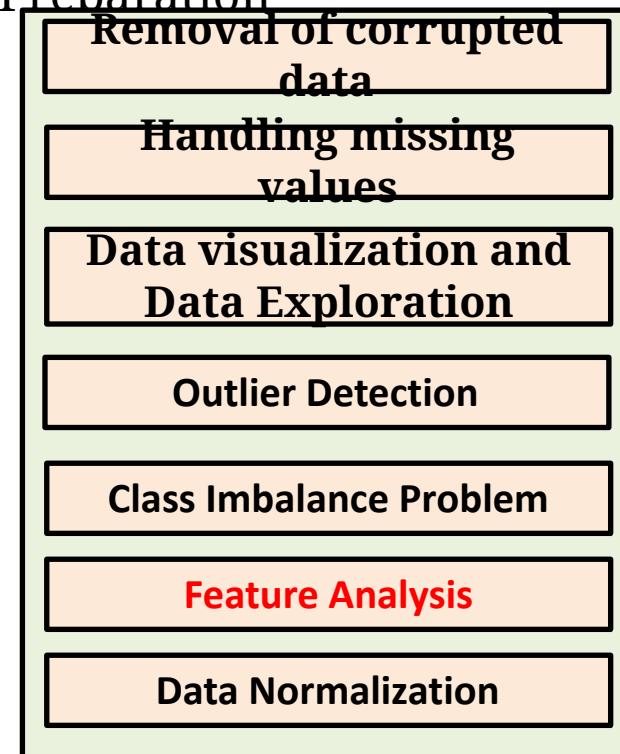
- Cost Sensitive Neural Network

Ensemble methods

- EasyEnsemble
- BalanceCascade
- SMOTEBost
- RUSboost

Machine Learning Model

Pre-processing/ Data Preparation



What is a Feature?

Feature is a quantifiable/measurable characteristic of a phenomenon being observed.

BP	Heart Beat	Weight	Diabetes
120	70	50	Y
125	65	60	Y
130	59	52	N

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor

Types of feature?

Discrete Feature: can only take certain values. Can be numeric or categorical: the results of rolling 2 dice.

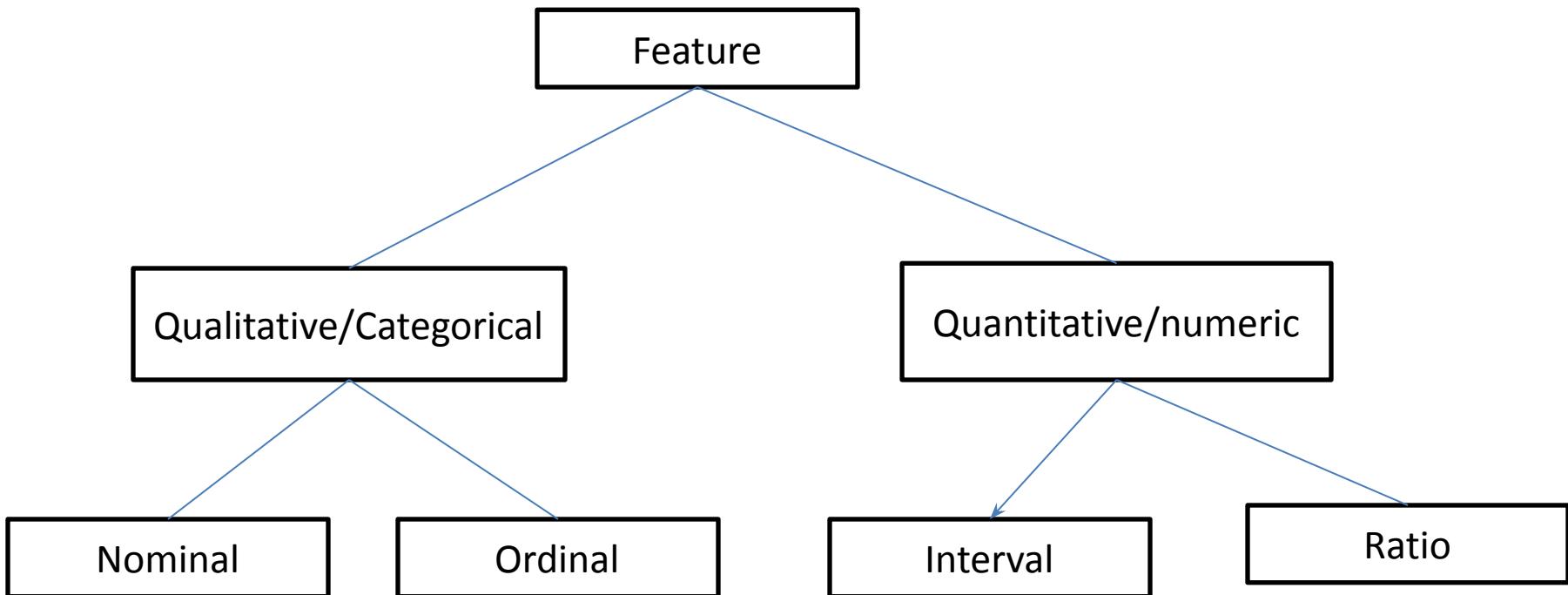
Sepal Length	Sepal Width	Petal Length	Petal Width	Species
big	medium	medium	small	Virginica
big	medium	small	small	Setosa
big	small	medium	small	Versicolor

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
3	2	2	1	Virginica
3	2	1	1	Setosa
3	1	2	1	Versicolor

Continuous Feature: can take any value (within a range)

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor

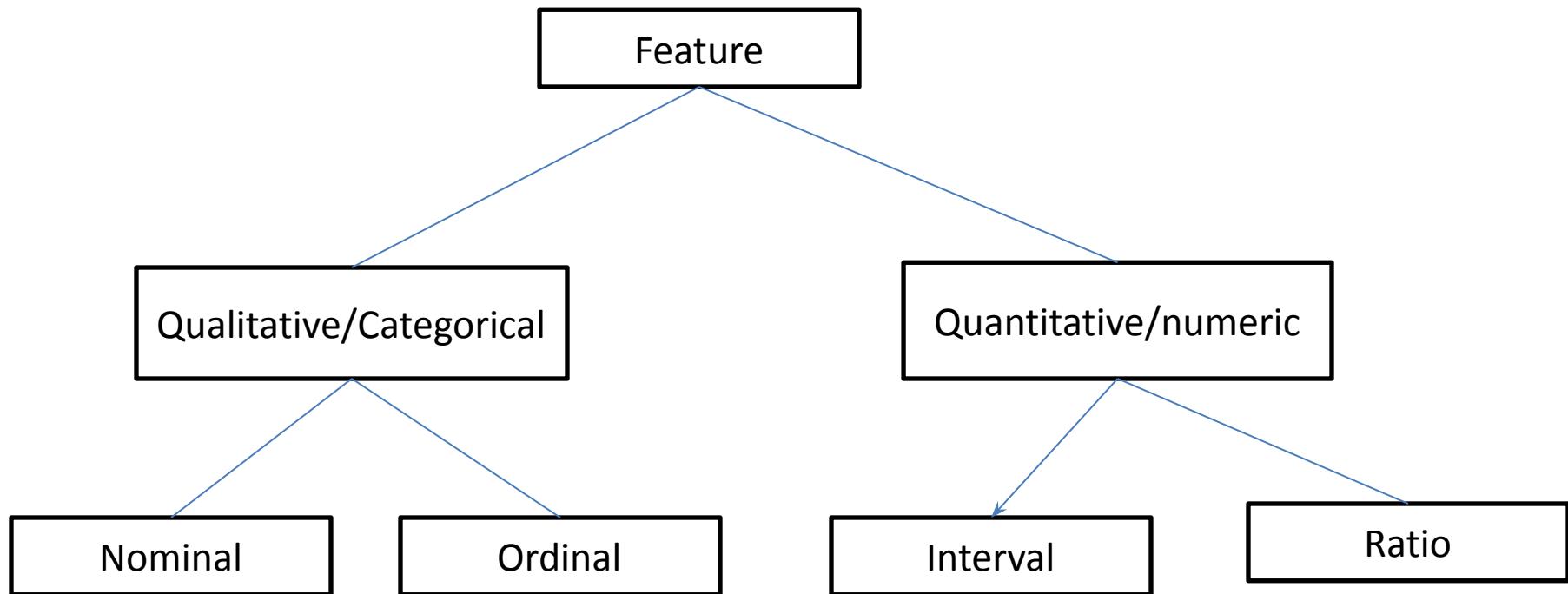
Types of feature?



Nominal Data: the range of values is not ordered in any sense, but simply **named** (hence the nom). Blood groups, gender, etc.

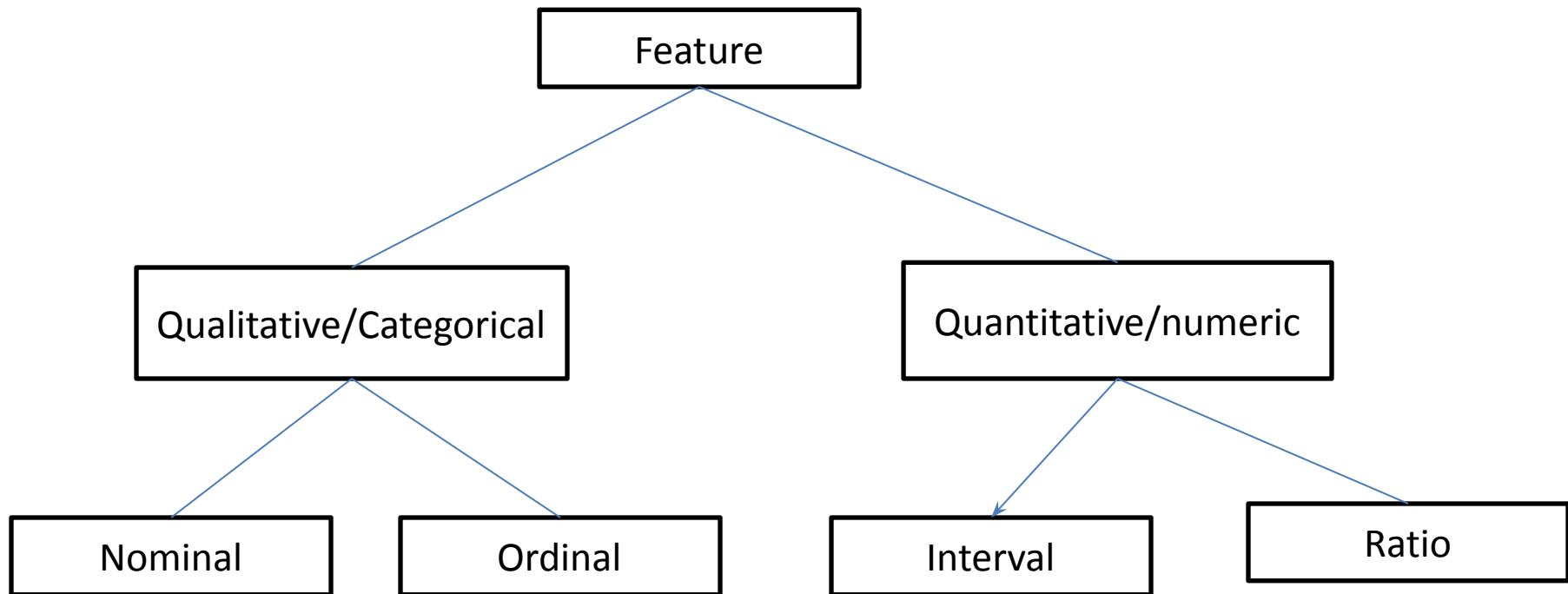
Ordinal Data: the range of values is ordered: **socio economic status** (“low income”, “middle income”, “high income”), education level (“high school”, “BS”, “MS”, “PhD”), income level (“less than 50K”, “50K-100K”, “over 100K”), satisfaction rating (“extremely dislike”, “dislike”, “neutral”, “like”, “extremely like”)

Types of feature?



Interval Data: Interval data is a type of data which is measured along a scale, in which each point is placed at an equal distance (interval) from one another. **The difference between 100 degrees Fahrenheit and 90 degrees Fahrenheit is the same as 60 degrees Fahrenheit and 70 degrees Fahrenheit.** Time of each day, Age, Dates, Voltage.

Types of feature?

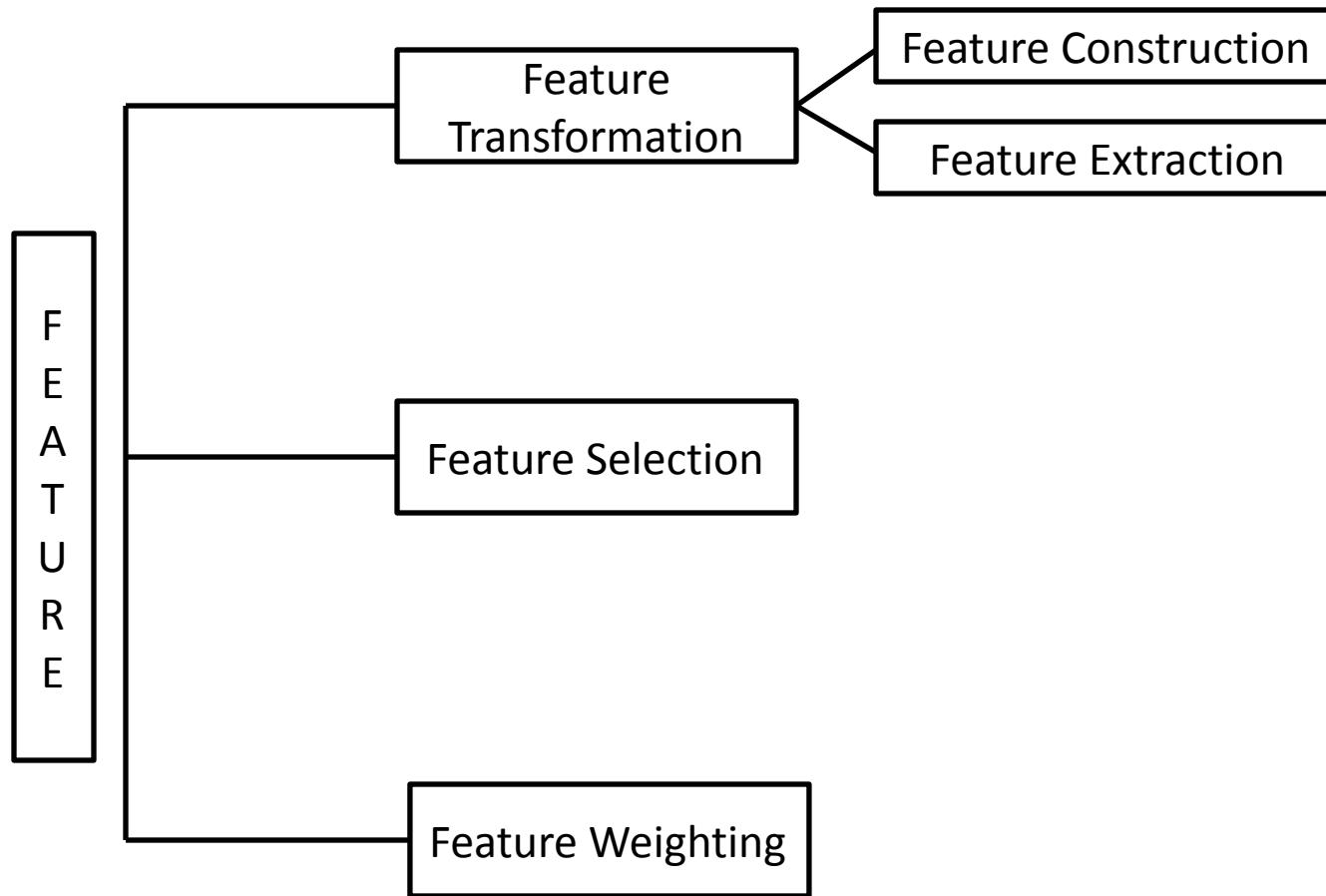


Ratio Data: Unlike interval data, ratio data has a true zero. This basically means that zero is an absolute, below which there are no meaningful values. Speed, age, or weight are all excellent examples since none can have a negative value (you cannot be -10 years old or weigh -160 pounds!)

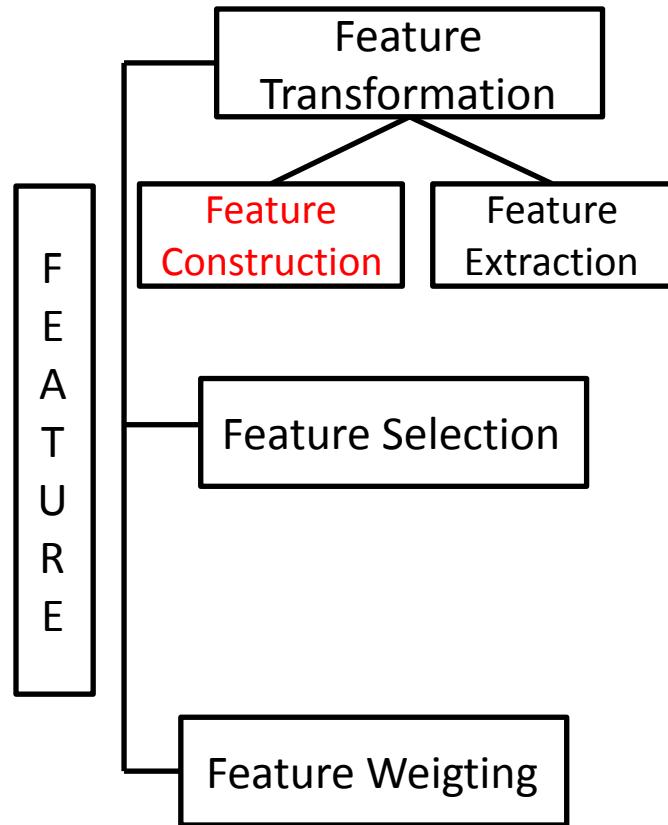
THE FOUR LEVELS OF MEASUREMENT:

	Nominal	Ordinal	Interval	Ratio
Categorizes and labels variables	✓	✓	✓	✓
Ranks categories in order		✓	✓	✓
Has known, equal intervals			✓	✓
Has a true or meaningful zero				✓

Feature Analysis



Feature Construction



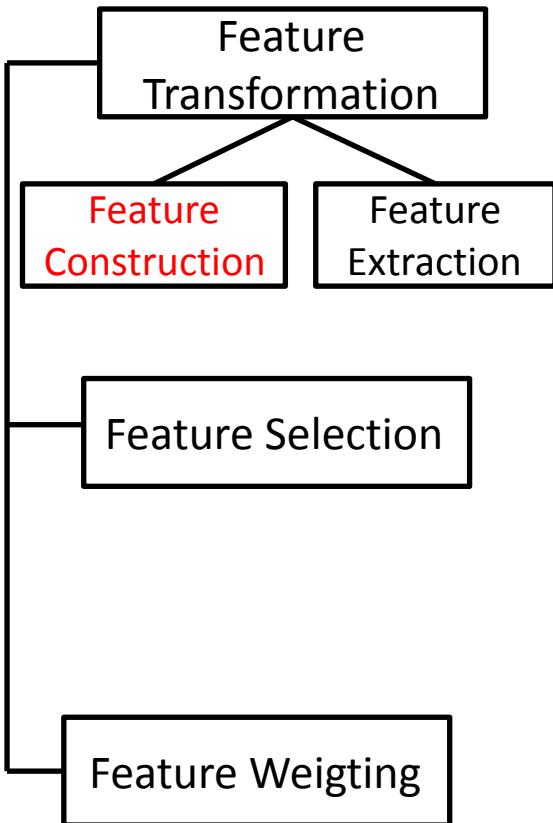
Feature Construction: Generates a new set of more powerful feature from a given set of input feature. Say, there are n features, after feature construction m more feature are added. Now total features = $(n+m)$

A Length	A Breadth	A price in RS
80	59	2360000
54	45	1215000
78	56	2184000

A Length	A Breadth	A Area	A price in RS
80	59	4720	2360000
54	45	2430	1215000
78	56	4368	2184000

Feature Construction

F
E
A
T
U
R
E



Need of Feature Construction:

- When features have categorical value and machine learning needs numeric value inputs.
- When features having continuous value and need to be converted to discrete/ordinal values.
- When **text-specific feature construction** needs to be done.
- Sometimes improves system performance

Feature Construction

Encoding categorical (nominal) variables:

Age	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N

Age	City A	City B	City C	Parents athlete_Y	Win_chance_1
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0

Feature Construction

Encoding categorical (ordinal) variables:

Science	maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D

Science	maths	Grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4

Feature Construction

Numeric features (continuous) to categorical features:

A_Area	A_Price
4720	2300000
2430	1200000
4368	2100000
3969	1900000
6142	3000000

A_Area	A_Grade
4720	Medium
2430	Low
4368	Medium
3969	Low
6142	High

A_Area	A_Grade
4720	2
2430	1
4368	2
3969	1
6142	3

Feature Construction

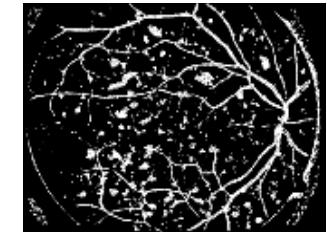
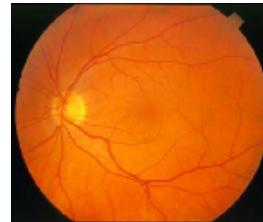
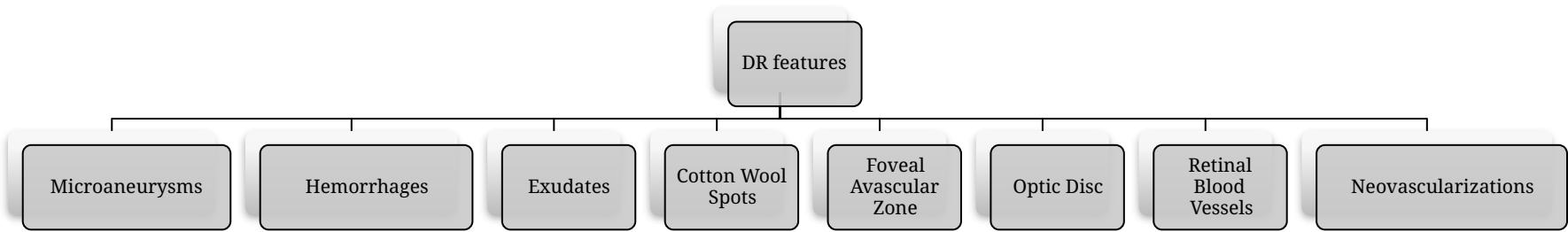
Text Specific feature construction and representation: Vector Space Model (VSM) and Graph Based Model (GBM)

Thrilling	movie	good	attractive	looser	lonely	expectation	bore	class
1	1	1	1	0	0	1	0	1
1	0	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	1
0	0	0	0	1	1	1	1	0
1	1	0	0	1	1	0	1	0
0	1	0	0	1	1	0	1	0

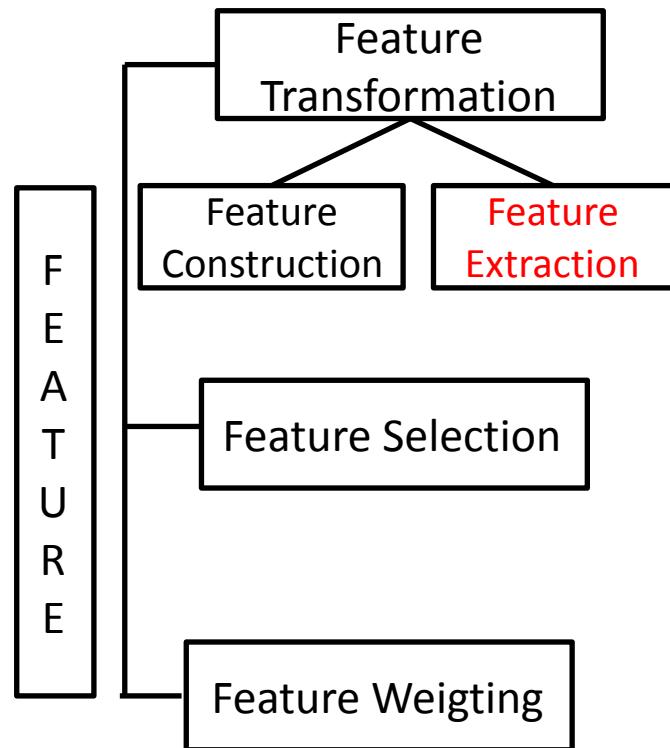
1. The movie is good, thrilling, attractive. It is up to my expectation ----- 1 (good)
3. The movie is good and attractive and is up to expectation ----- 1 (good)
5. The movie was looser, lonely. It was bore. It was thrilling. ----- 0 (bad)

Feature Construction

Image Feature Construction for diabetic retinopathy



Feature Extraction



Feature Extraction: New features are created from a combination of original features: some operators:
Boolean Features: Conjunction, Disjunction, Negation etc.

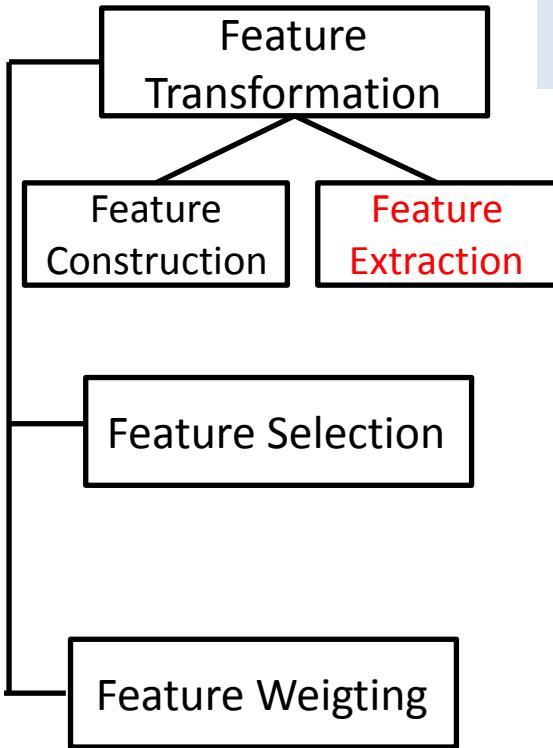
Nominal Features: Cartesian product, M of N etc.

Numerical Features: Min, Max, Addition, Subtraction, Multiplication, Division, Average, Equivalence, Inequality etc.

$$F' = f(F); \quad F_1' = k_1 F_1 + k_2 F_2; \quad m < n$$

Feature Extraction

F
E
A
T
U
R
E



Feat_A	Feat_B	Feat_C	Feat_D
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2

Feat_1	Feat_2
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
37.02	238.42

$$\text{Feat_1} = 0.3 * \text{Feat_A} + 0.9 * \text{Feat_B}$$

$$\text{Feat_2} = \text{Feat_B} + 0.5 * \text{Feat_C} + 0.6 * \text{Feat_D}$$

Feature Extraction

Some of the popular feature extraction algorithms used in ML are given below:

1. Singular Value Decomposition (SVD)
2. Linear Discriminant Analysis (LDA)
3. Principle Component Analysis (PCA)
4. Fisher's Linear Discriminant (FLD)

Principal Component Analysis

PCA is a useful statistical ML technique that has found application in fields such as face recognition and image compression, and **is a common technique for finding patterns in data of high dimension.**

The other main advantage of PCA is that once we have found these patterns in the data, then we can compress the data, ie. by reducing the number of dimensions, without much loss of information.

Method:

Step 1: Get some data

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Data =

Principal Component Analysis

Step 2: Subtract the mean

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9
Mean	
X'=1.81	Y'=1.91

$$2.5 - 1.81 = .69$$

$$2.4 - 1.91 = .49$$

DataAdjust =

(X-X')	(Y-Y')
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.0

Principal Component Analysis

Step 3: Calculate the covariance matrix of x and y (two dimensional data)

$$\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix}$$

If you calculate the covariance between one dimension and itself, you get the variance.

The formula for variance could also be written like this:

$$var(X) = \frac{\sum_1^n (X_i - X') (X_i - X')}{(n - 1)}$$

$$cov(X, Y) = \frac{\sum_1^n (X_i - X') (Y_i - Y')}{(n - 1)}$$

Principal Component Analysis

$$\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix}$$

$$var(X) = \frac{\sum_1^n (X_i - X') (X_i - X')}{(n - 1)}$$

$$cov(X, Y) = \frac{\sum_1^n (X_i - X') (Y_i - Y')}{(n - 1)}$$

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

cov(X, Y)=(.69*.49)+(-1.31*-.21)+(.39*.99)+(.09*.29)+(1.29*1.09)+(.49*.79)+(.19*-.31)+(-.81*-.81)+(-.31*-.31)+(-.71*-1.0)=5.539

(X-X')	(Y-Y')
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.0
5.539/9=0.615444444	
Var(X)=	
5.549/9=0.616555556	
Var(Y)=	
6.4289/9=0.714322222	

Principal Component Analysis

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

$$eigenvalues = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

Step 5: Choosing components and forming a feature vector

It turns out that the eigenvector with the highest eigenvalue is the principle component of the data set.

The eigenvector with the largest eigenvalue is the one that points the middle of the data.

It is the most significant relationship between the data dimensions.

Principal Component Analysis

Step 5: Choosing components and forming a feature vector

$$eigenvalues = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

FeatureVector= $eig_1eig_2eig_3eig_4$

$$= \begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

Principal Component Analysis

Step 6: Deriving the new data set

we simply take the transpose of the vector and multiply it on transpose of *DataAdjust*

$$= \begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

$$\text{FinalData} = \text{RowFeatureVector} \times \text{RowDataAdjust}$$

$$\begin{pmatrix} -.68 & -.74 \end{pmatrix} \times \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & . . \\ .49 & -1.21 & .99 & .29 & 1.09 & .79 \end{pmatrix}$$

$$= (-.68 * .69) + (-.74 * .49)$$

$$=(-0.8318, 1.7862, -0.9978, -0.2758, -1.6838, -0.9178, 0.1002, 1.1502, 0.4402, 1.2228)$$

Feature Selection

Feature Subset Selection: is the most critical pre-processing in ML and selects a subset which keeps meaningful contribution in a ML task.

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3

Feature set= {F1, F2, F3, F4,FN}

Feature Subset= {F1, F2,,FM}

Where M<=N

- High-dimensional data: DNA analysis, GIS, Social Networking etc.
- DNA microarray data can have up to 450000 variables(gene)
- Text data is extremely high dimensional data

Feature Subset Selection: is the most critical pre-processing in ML and selects a subset which keeps meaningful contribution in a ML task.

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3

Advantages of Feature Selection:

1. Improve accuracy
2. Improve efficiency by reducing time complexity
3. It helps in the simplification of the model so that it can be easily interpreted by the researchers.

Feature Selection

Jobs of Feature Subset Selection— 1. Feature relevance 2. Feature redundancy

2. Feature Relevance: i. Irrelevant feature ii. Relevant feature

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

2. Feature Redundancy:

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
14	1.25	25.3

Measures of feature relevance:

- **Information Gain**
- **Mutual information**
- **Fisher score**
- **Analysis of Variance (ANOVA)**
- **Chi-Square**
- **Dispersion ratio**
- **Relief**

Feature Selection

Measures of feature relevance using **Information Gain**:

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	Students
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Students
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Students
T6	Quiet	Baked	Sandals	Students
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Students

Information Gain

- Let C_1, C_2, \dots, C_m be the number of classes where $m > 1$.
- V_0 = Database (set of data)
- $Y(k, 0)$ = number of training patterns of class C_k in the set V_0
- $Z(0)$ = number of patterns in the set V_0

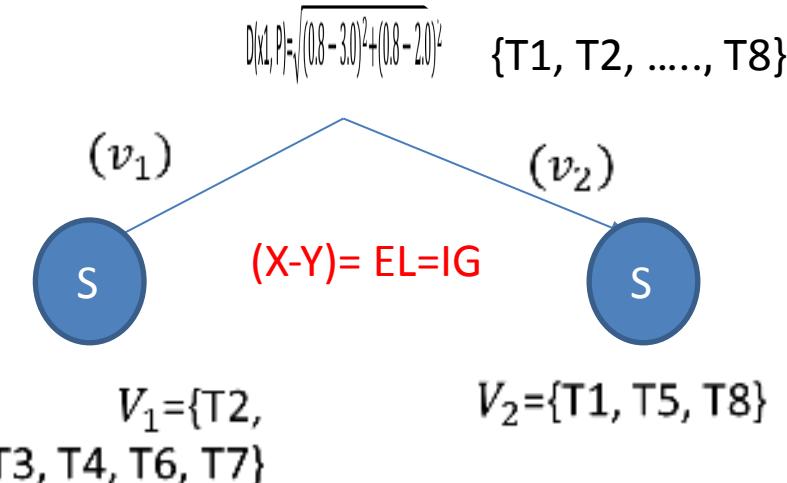
$$Z(0) = \sum_{k=1}^m Y(k, 0)$$

- The probability that a pattern in V_0 belongs to class C_k is

$$\frac{Y(k, 0)}{Z(0)}$$

- The information required to classify a pattern of V_0 into the class C_k , for $1 \leq k \leq m$ is expressed as

$$-\log \frac{Y(k, 0)}{Z(0)}$$



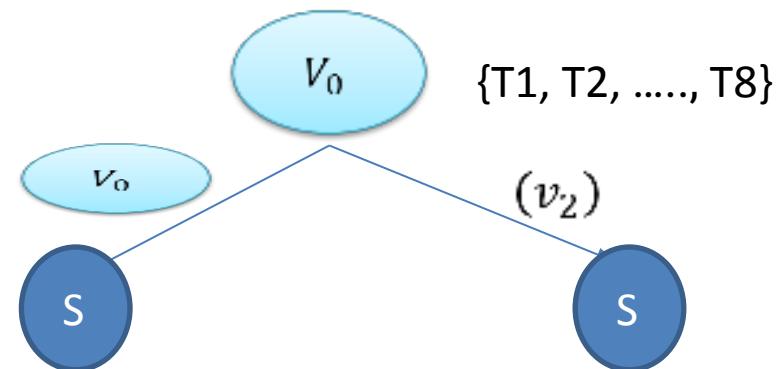
NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Information Gain

- The weighted average information required to classify a pattern of set V_0 into one of the m classes is expressed as

$$I(V_0) = \sum_{k=1}^m \frac{Y(k, 0)}{Z(0)} (-\log \frac{Y(k, 0)}{Z(0)})$$

$I(V_0)$ is called the entropy of the set V_0 .



Density is reverse of distance therefore Local Reachability score LRD

$$\begin{aligned} LRD_A &= \frac{1}{RD_A} \\ &= 1/6.06 = 0.165 \end{aligned}$$

- Similarly for the set V_j ,

$$I(V_j) = \sum_{k=1}^m \frac{Y(k, j)}{Z(j)} (-\log \frac{Y(k, j)}{Z(j)})$$

- The weighted average information required to classify a pattern into one of class k in set V_0 after it has been split by the attribute A into sets V_1 to V_n is given by

$$I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} I(V_j)$$

- $I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} (-\log \frac{Y(k,j)}{Z(j)})$
 $I_A(V_0)$ is called the entropy of the attribute A for the set V_0
- The gain in information caused by attribute A splitting set V_0 into sets V_1 to V_n is

$$g_A(V_0) = I(V_0) - I_A(V_0)$$

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Example

Evaluate the entropy $I(V_0)$ of the Professor-student training set.

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTW EAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roast ed	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

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

Therefore,

$$\begin{aligned}I(V_0) &= - \sum_{k=1}^m \frac{Y(k,0)}{Z(0)} \log \frac{Y(k,0)}{Z(0)} \\&= - \sum_{k=1}^2 \frac{Y(k,0)}{Z(0)} \log \frac{Y(k,0)}{Z(0)} \\&= - \frac{3}{8} \log \frac{3}{8} - \frac{5}{8} \log \frac{5}{8} \\&= 0.9544\end{aligned}$$

Example

Information gain of HABIT

- S and P are the two classes, hence m=2.
- The training set $V_0 = \{T1, T2, T3, \dots, T8\}$.
- $Y(1,0)=3$ (number of patterns in V_0 of class P)
- $Y(2,0)=5$ (number of patterns in V_0 of class S)
- $Z(0)=8$ (number of patterns in V_0)
- $I(V_0)=0.9544$
- $V_1=\{T1, T2, T3, T5, T7\}$ at node y_1 , where HABIT=gabby.
 - $Y(1,1)=2$ (number of patterns in V_1 of class P)
 - $Y(2,1)=3$ (number of patterns in V_1 of class S)
 - $Z(1)=5$ (number of patterns in V_1)
- $V_2=\{T4, T6, T8\}$ at node y_2 , where HABIT=quiet.
 - $Y(1,2)=1$ (number of patterns in V_2 of class P)
 - $Y(2,2)=2$ (number of patterns in V_2 of class S)
 - $Z(2)=3$ (number of patterns in V_2)

$$\begin{aligned}I_{Habit}(V_0) &= - \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \log \frac{Y(k,j)}{Z(j)} \\&= [5/8\{(2/5(-\log2/5)+(3/5(-\log3/5)\}\\&\quad + 3/8\{(1/3(-\log1/3)+(2/3(-\log2/3)\}\\&= 0.9499\end{aligned}$$

$$\begin{aligned}G_{Habit}(V_0) &= \frac{I(V_0) - I_{Habit}(V_0)}{I(V_0)} \\&= (0.9544 - 0.9499) \\&= 0.0100\end{aligned}$$

Example

Information gain of EATS

- S and P are the two classes, hence m=2.
- The training set $V_0 = \{T1, T2, T3, \dots, T8\}$.
- $Y(1,0)=3$ (number of patterns in V_0 of class P)
- $Y(2,0)=5$ (number of patterns in V_0 of class S)
- $Z(0)=8$ (number of patterns in V_0)
- $I(V_0)=0.9544$
- $V_1 = \{T1, T3, T6\}$ at node y_1 , where EATS=baked.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T4, T5, T7, T8\}$ at node y_2 , where EATS=fried.
- $Y(1,2)=2$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=4$ (number of patterns in V_2)
- $V_3 = \{T2\}$ at node y_3 , where EATS=roasted.
- $Y(1,3)=1$ (number of patterns in V_3 of class P)
- $Y(2,3)=0$ (number of patterns in V_3 of class S)
- $Z(3)=1$ (number of patterns in V_3)

$$I_{Eats}(V_0) \\ = - \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \log \frac{Y(k,j)}{Z(j)} \\ = 0.5$$

$$G_{Eats}(V_0) = \frac{I(V_0) - I_{Eats}(V_0)}{I(V_0) - I_{Eats}(V_0)} \\ = (0.9544 - 0.5) \\ = 0.4544$$

Example

information gain of FOOTWEAR

- S and P are the two classes, hence m=2.
- The training set $V_0 = \{T1, T2, T3, \dots, T8\}$.
- $Y(1,0)=3$ (number of patterns in V_0 of class P)
- $Y(2,0)=5$ (number of patterns in V_0 of class S)
- $Z(0)=8$ (number of patterns in V_0)
- $I(V_0)=0.9544$
- $V_1 = \{T1, T5, T8\}$ at node y_1 , where FOOTWEAR=clogs.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T2, T3, T4, T6, T7\}$ at node y_2 , where FOOTWEAR=sandals.
- $Y(1,2)=3$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=5$ (number of patterns in V_2)

Therefore,

$$I_{Footwear}(V_0) = - \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \log \frac{Y(k,j)}{Z(j)}$$
$$= 0.6066$$

$$G_{Footwear}(V_0) = \frac{I(V_0) - I_{Footwear}(V_0)}{I(V_0)}$$
$$= (0.9544 - 0.6066)$$
$$= 0.3478$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabb y	Bake d	Clogs	S
T2	Gabb y	Roas ted	Sanda ls	P
T3	Gabb y	Bake d	Sanda ls	S
T4	Quiet	Fried	Sanda ls	P
T5	Gabb y	Fried	Clogs	S
T6	Quiet	Bake d	Sanda ls	S
T7	Gabb y	Fried	Sanda ls	P
T8	Quiet	Fried	Clogs	S

Example

$$G_{Habit}(V_0) = 0.0100$$

$$G_{Eats}(V_0) = 0.4544$$

$$G_{Footwear}(V_0) = 0.3478$$

Measures of feature redundancy:

1. Similarity-based measure

- i. Pearson Correlation Coefficient
- ii. Spearman's Correlation
- iii. Kendall's Tau
- iv. Jaccard Index/Coefficient
- v. Simple Matching Coefficient (SMC)
- vi. Cosine Similarity

2. Distance-based measure

- i. Euclidean distance
- ii. Manhattan distance

Correlation-based measure: Measures linear dependency between two random variables.

Feature Redundancy

Pearson Correlation Coefficient: measures linear dependency between two random variables

$$\alpha = \frac{cov(F1, F2)}{\sqrt{var(F1) \cdot var(F2)}}$$

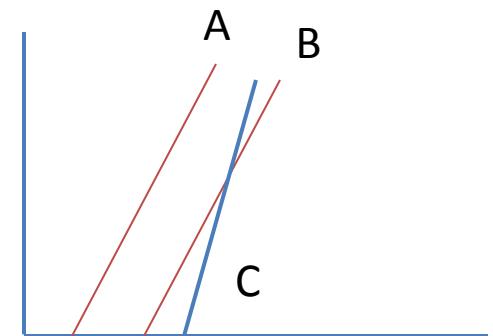
$$cov(F1, F2) = \sum \{(F1_i - F1').(F2_i - F2')\} / (n - 1)$$

$$var(F1) = \sum (F1_i - F1')^2 / (n - 1), \text{ where } F1' = \frac{1}{n} \sum F1_i$$

$$var(F2) = \sum (F2_i - F2')^2 / (n - 1), \text{ where } F2' = \frac{1}{n} \sum F2_i$$

It ranges between +1 and -1

Covariance is a measure of the relationship between two random variables. The metric evaluates how much – to what extent – the variables change together. In other words, it is essentially a measure of the variance between two variables



Feature Redundancy

Spearman's Correlation Coefficient: measures linear dependency between two random variables. It uses rank of each value. Data are represented as

$$\mathbf{x} = \mathbf{x}^r$$

$$\mathbf{y} = \mathbf{y}^r$$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

$$S(F1, F2) = \frac{cov(F1, F2)}{\sqrt{var(F1).var(F2)}}$$

$$cov(F1, F2) = \sum\{(F1_i^r - F1'^r) \cdot (F2_i^r - F2'^r)\} / (n-1)$$

$$var(F1) = \sum(F1_i^r - F1'^r)^2 / (n-1) \quad \text{where } F1'^r = \frac{1}{n} \sum F1_i^r$$

$$var(F2) = \sum(F2_i^r - F2'^r)^2 / (n-1), \quad \text{where } F2'^r = \frac{1}{n} \sum F2_i^r$$

It ranges between +1 and -1

If $S > P$ it means that we have a monotonic relationship, not a linear relationship.

Feature Redundancy

Kendall's Tau: measures linear dependency between two random variables. It uses rank of each value. Kendall's Tau has smaller variability when using larger sample sizes. However, Spearman's measure is more computationally efficient, as Kendall's Tau is $O(n^2)$ and Spearman's correlation is $O(n\log(n))$.

Data are represented as

$$x=x^r$$

$$y=y^r$$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

It ranges between +1 and -1

Feature Redundancy

$$\text{Kendall's Tau} = (C - D) / (C + D)$$

Feature1: 1 2 3 4 5 6 7 8 9 10 11 12

Feature 2: 1 2 4 3 6 5 8 7 10 9 12 11

Step1: Make a table of rankings. The rankings for Feature 1 should be in ascending order

Feature1: 1 2 3 4 **12** 6 7 8 9 10 11 **5**

Feature 2: 1 2 4 3 6 5 8 7 10 9 12 11

Feature1: 1 2 3 4 **5** 6 7 8 9 10 11 **12**

Feature 2: 1 2 4 3 **11** 5 8 7 10 9 12 **6**

Feature1	Feature 2
1	1
2	2
3	4
4	3
5	6
6	5
7	8
8	7
9	10
10	9
11	12
12	11

Feature Redundancy

Step 2: Count the number of concordant pairs,

using the second column. Concordant pairs are how many larger ranks are below a certain rank.

For example, the first rank in the second Feature's column is a "1", so all 11 ranks below it are larger.

Feature 1	Feature 2	Concor- dant	Discord- ant
1	1	11	
2	2	10	
3	4	8	
4	3	8	
5	6	6	
6	5	6	
7	8	4	
8	7	4	
9	10	2	
10	9	2	
11	12	0	
12	11		

Feature Redundancy

Step 3: Count the number of discordant

pairs and insert them into the next column.

The number of discordant pairs is similar to Step 2, only you're looking for smaller ranks, not larger ones.

Feature 1	Feature 2	Concordant	Discordant
1	1	11	0
2	2	10	0
3	4	8	1
4	3	8	0
5	6	6	1
6	5	6	0
7	8	4	1
8	7	4	0
9	10	2	1
10	9	2	0
11	12	0	1
12	11		

Feature Redundancy

Step 4: Sum the values in the two columns:

Step 5: Insert the totals into the formula:

$$\text{Kendall's Tau} = (C - D / C + D) = (61 - 5) / (61 + 5) = 56 / 66 = .85.$$

The Tau coefficient is .85, suggesting a strong relationship between the rankings.

Feature 1	Feature 2	Concordant	Discordant
1	1	11	0
2	2	10	0
3	4	8	1
4	3	8	0
5	6	6	1
6	5	6	0
7	8	4	1
8	7	4	0
9	10	2	1
10	9	2	0
11	12	0	1
12	11		
Total		61	5

Feature Redundancy

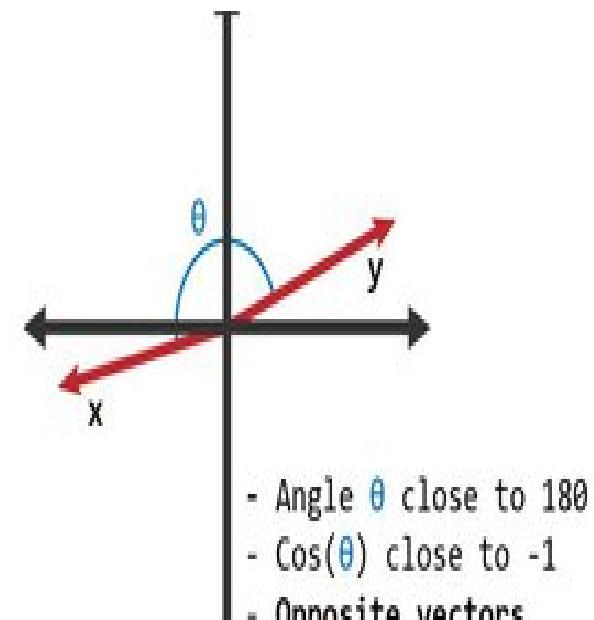
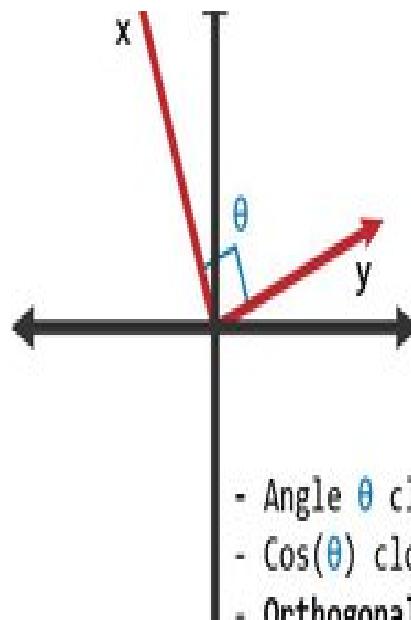
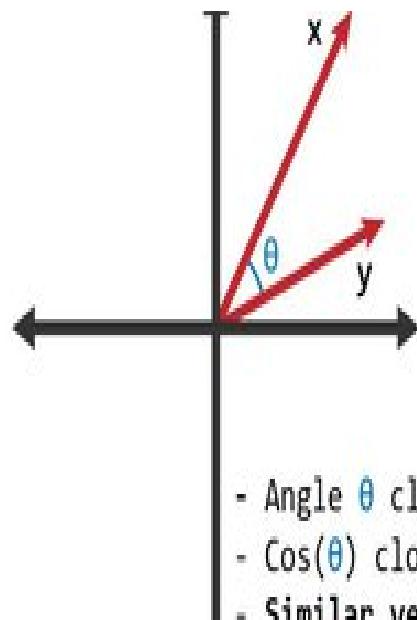
Perfect Correlation

Tau = $(66 - 0) / (66 + 0) = 1$, which is (as we expect) perfect agreement.

Feature 1	Feature 2	Concordant	Discordant
1	1	11	0
2	2	10	0
3	3	9	0
4	4	8	0
5	5	7	0
6	6	6	0
7	7	5	0
8	8	4	0
9	9	3	0
10	10	2	0
11	11	1	0
12	12		
Total		66	0

Feature Redundancy

Cosine Similarity: The cosine similarity calculates the cosine of the angle between two vectors. The cosine similarity can take on values between -1 and +1. If the vectors point in the exact same direction, the cosine similarity is +1. If the vectors point in opposite directions, the cosine similarity is -1.



Feature Redundancy

Cosine Similarity: The cosine similarity calculates the cosine of the angle between two vectors. The cosine similarity can take on values between -1 and +1. If the vectors point in the exact same direction, the cosine similarity is +1. If the vectors point in opposite directions, the cosine similarity is -1.

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=0}^{n-1} x_i y_i}{\sqrt{\sum_{i=0}^{n-1} (x_i)^2} \times \sqrt{\sum_{i=0}^{n-1} (y_i)^2}}$$

$\mathbf{x}=(2, 4, 0, 0, 2, 1, 3, 0, 0)$ and $\mathbf{y}=(2, 1, 0, 0, 3, 2, 1, 0, 1)$

x. y=2*2+4*1+0*0+0*0+2*3+1*2+3*1+0*0+0*1=19

$$\|\mathbf{x}\| = \sqrt{2^2 + 4^2 + 0^2 + 0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} = 5.83$$

$$\|\mathbf{y}\| = \sqrt{2^2 + 1^2 + 0^2 + 0^2 + 3^2 + 2^2 + 1^2 + 0^2 + 1^2} = 4.47$$

$$\text{Cos } (\mathbf{x}, \mathbf{y}) = \frac{19}{5.83 \times 4.47} = 0.729$$

Feature Redundancy

Jaccard Coefficient: measures similarity between two features having binary values.

$$J = \frac{n_{11}}{n_{01} + n_{10} + n_{11}}$$

F1	0	1	1	0	1	0	1	0
F2	1	1	0	0	1	0	0	0

$$J = \frac{2}{1+2+2} = 0.4$$

$$\text{Jaccard distance} = 1 - J = 1 - 0.4 = 0.6$$

Feature Redundancy

Simple Matching Coefficient (SMC): Same as Jaccard coefficient except the fact that it includes a number of cases where both the features have a value 0.

$$SMC = \frac{n_{11} + n_{00}}{n_{01} + n_{10} + n_{11} + n_{00}}$$

F1	0	1	1	0	1	0	1	0
F2	1	1	0	0	1	0	0	0

$$SMC = \frac{2+3}{1+2+2+3} = 0.5$$

Distance-Based Measure

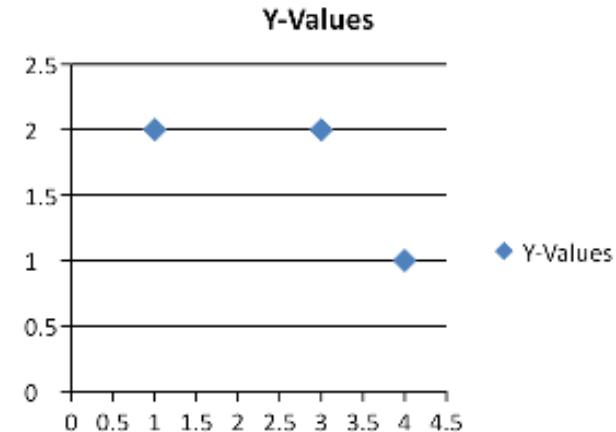
Let $p = (p_1, p_2)$ and $q = (q_1, q_2)$ be two points:

- ✓ City block distance $d(p, q) = |p_1 - q_1| + |p_2 - q_2|$
- ✓ Euclidean distance $d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$
- ✓ Minkowski distance $d(p, q) = (\sum_{i=1}^M |p_i^n - q_i^n|^r)^{\frac{1}{r}}$

For r=1, Minkowski distance =City block distance

For r=2, Minkowski distance = Euclidean distance

	M=1	M=2
1 st	1	2
2 nd	3	2
3 rd	4	1



$$1^{\text{st}} = (1, 2)$$

$$2^{\text{nd}} = (3, 2)$$

$$3^{\text{rd}} = (4, 1)$$

$$\text{Dis}(1^{\text{st}}, 2^{\text{nd}}) = |1-3| + |2-2| = 2$$

DAY-5

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. Embedded approach

Filter Approach:

- Filter methods pick up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance.
- statistical measure used, no learning algorithm;
- These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods.
- (A, B, C, D,E): Filter approach= 5---Wrapper $(5+4+3+2+1)=15$
- Selection of feature is evaluated individually (don't have a dependency on other features) but will lag when a combination of features can lead to increase in the overall performance of the model.

Feature Selection Approaches

- Stopping criteria for selecting the best subset are usually pre-defined by the person training the model such as when the performance of the model decreases or a specific number of features has been achieved.

(A, B, C, D,E): (when the performance of the model decreases)

- A= 0.76
- B= 0.90
- C= 0.65
- D= 0.50
- E= 0.95

{E}=80%; {E, B}= 82%; {E, B, A}=90%; {E, B, A, C}=88%;

(A, B, C, D,E): (a specific number (2) of features has been achieved)

{E, B}= 82%;

- Some of statistical test conducted on features are as follows:
 - i. Pearson's correlation
 - ii. information gain
 - iii. Fisher score
 - iv. Analysis of Variance (ANOVA)
 - v. Chi-Square,
 - VI. Correlation Coefficient
 - VII. Variance Threshold
 - VIII. Mean Absolute Difference (MAD)
 - IX. Dispersion ratio
 - X. Mutual Dependence
 - XI. Relief
 - XII. Missing Value Ratio

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. Embedded approach

Filter Approach: The implementation of filter approach

Set of all features → Selecting the best subset → Learning algorithm → Performance

- Some of statistical test conducted on features are as follows.
 - i. Pearson's correlation
 - ii. information gain
 - iii. Fisher score
 - iv. Analysis of Variance (ANOVA)
 - v. Chi-Square,
 - VI. Correlation Coefficient
 - VII. Variance Threshold
 - VIII. Mean Absolute Difference (MAD)
 - IX. Dispersion ratio
 - X. Mutual Dependence
 - XI. Relief
 - XII. Missing Value Ratio

Feature Selection Approaches: Filter Approach

Fisher score:

- Fisher score is one of the most widely used supervised feature selection methods.
- It selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features.
- The score of the i-th feature S_i will be calculated by Fisher Score,

$$S_i = \frac{\sum n_j (\mu_{ij} - \mu_i)^2}{\sum n_j * p_{ij}^2}$$

where μ_{ij} and p_{ij} are the mean and the variance of the i-th feature in the j-th class, respectively,

- n_j is the number of instances in the j-th class and μ_i is the mean of the i-th feature.
- The features are ranked according to the Fisher Score.

Example

A1	A2	Class
2	0.25	1
5	1.02	0
7	1	0
3	0.75	1
2.5	.6	0
1.98	1	0

Fisher Score of feature A1 is

$$S_{A1} = \frac{\sum n_j(\mu_{ij} - \mu_i)^2}{\sum n_j * p_{ij}^2}$$

$$\mu_{A11} = 2.5$$

$$p_{A11} = 0.25$$

$$\mu_{A10} = 4.12$$

$$p_{A10} = 4.07$$

$$n_1 = 2$$

$$n_0 = 4$$

$$\mu_{A1} = 3.58$$

Fisher Score of feature A2 is

$$S_{A2} = \frac{\sum n_j(\mu_{ij} - \mu_i)^2}{\sum n_j * p_{ij}^2}$$

$$\mu_{A21} = 0.5$$

$$p_{A21} = 0.062$$

$$\mu_{A20} = 0.91$$

$$p_{A20} = 0.031$$

$$n_1 = 2$$

$$n_0 = 4$$

$$\mu_{A2} = 0.77$$

$$S_{A1} = \frac{2(2.5 - 3.58)^2 + 4(4.12 - 3.58)^2}{(2 * 0.25^2) + (4 * 4.07^2)}$$

$$= 0.05$$

$$S_{A2} = \frac{2(0.5 - 0.77)^2 + 4(0.91 - 0.77)^2}{(2 * 0.062^2) + (4 * 0.031^2)}$$

$$= 16.43$$

- The feature A2 has a higher rank than the feature A1
- Hence feature A2 is more important in the prediction process than feature A1

Relief algorithms

There are three Algorithms in the Relief Family:

- **Basic Relief algorithm:** It is limited to classification problems with two classes.
- **ReliefF :** Extension of Relief . Which can deal with multiclass problems.
- **RReliefF:** Then ReliefF was adapted for continuous class (regression) problems resulting in RReliefF algorithm.

However the basic idea in all the three algorithms remains the same.

The core idea is on the basis of how well the attribute can distinguish between instances that are near to each other.

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**

- 1. set all weights $W[A] := 0.0;$
- 2. for $i := 1$ to m do begin
3. randomly select an instance $R_i;$
4. find nearest hit H and nearest miss $M;$
5. for $A := 1$ to a do
6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m;$
7. end;

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0 & ; \text{value}(A, I_1) = \text{value}(A, I_2) \\ 1 & ; \text{otherwise} \end{cases}$$

for nominal attributes

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

for numerical attributes

I_1, I_2, \dots, I_n are examples.

Each example is a vector of attributes $A_i, i = 1, \dots, a$, where a is the number of attributes, and each example has a target value t_j .

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**

- 1. set all weights $W[A] := 0.0;$
- 2. for $i := 1$ to m do begin
3. randomly select an instance $R_i;$
4. find nearest hit H and nearest miss $M;$
5. for $A := 1$ to a do
6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m;$
7. end;

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0 & ; \text{value}(A, I_1) = \text{value}(A, I_2) \\ 1 & ; \text{otherwise} \end{cases}$$

for nominal attributes

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

for numerical attributes

A random instance R_i (line 3) and its two nearest neighbors: one of the same class that R_i belongs to known as nearest hit H and other of the different class known as nearest miss M (line 4).

The whole process is repeated m number of times and m is a user defined parameter.

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Select 2 best attributes.

Let $m = 2$

Step 1 (1): Let all attributes weight be 0 ,

$A=B=C=0$,

Step 2(3) : Row 5 is randomly selected instance. (i.e 6,0,0)

Step 3(4) : Using Manhattan distance

Nearest hit:

Row 4: $|6-8| + |0-3| + |0-1| = 6$

Row 3: $|6-9| + |0-3| + |0-2| = 8$

Row 4 is nearest hit.

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Step 3(4) : Using Manhattan distance

Nearest miss:

Row 2: $|6-5| + |0-1| + |0-0| = 2$

Row 1: $|6-9| + |0-2| + |0-2| = 7$

Row 2 is nearest miss.

Step 4(6) : Update weights of attributes

A,B,C : current weight = 0

$$\mathbf{A} = 0 - ((|6-8|/(9-5))/2) + ((|6-5|/(9-5))/2) = 0 - (0.5/2) + (0.25/2) = \mathbf{-0.1875}$$

$$\begin{aligned}\mathbf{B} &= 0 - ((|0-3|/(3-0))/2) + ((|0-1|/(3-0))/2) \\ &= 0 - (1/2) + (1/6) = \mathbf{-0.33}\end{aligned}$$

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Step 4(6) : Update weights of attributes

A,B,C : current weight = 0

$$C = 0 - ((|0-1|/(2-0))/2) + ((|0-0|/(2-0))/2)$$

$$= 0 - (1/4) + 0 = -0.25$$

Second Iteration:

Step 2: Row 4 is selected randomly.

Step 3:

Row 3 is selected the nearest hit : $|8-9| + |3-3| + |1-2| = 2$

Row 1 is selected the nearest miss : $|8-9| + |3-2| + |1-2| = 3$.

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Step 4(6) : Update weights of attributes

Current weight:

$$A = -0.1875; \quad B = -0.33; \quad C = -0.25.$$

$$A = -0.1875 - ((|8-9|/(9-5))/2) + ((|8-9|/(9-5))/2) = \textcolor{red}{-0.1875}$$

$$B = -0.33 - ((|3-3|/(3-0))/2) + ((|3-2|/(3-0))/2) = -0.33 - 0 + 0.166 = \textcolor{red}{-0.167}$$

$$C = -0.25 - ((|1-2|/(2-0))/2) + ((|1-2|/(2-0))/2) = \textcolor{red}{-0.25}$$

A and B as our 2 best features

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

- The original Relief can deal with nominal and numerical attributes.
- However, it cannot deal with incomplete data and is limited to two-class problems.
- Its extension, which solves these and other problems, is called ReliefF.

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

1. Filter Approach
2. **Wrapper Approach**
3. Hybrid Approach
4. Embedded approach

Wrapper Approach: (A, B, C, D,E): --Wrapper $(5+4+3+2+1)=15$

- Wrappers require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset.
- It follows a greedy search approach by evaluating all the possible combinations of features against some evaluation criterion.
- Learning algorithms are used as black box.
- Computationally very expensive, however performance is superior than filter approach;

1. Filter Approach
2. **Wrapper Approach**
3. Hybrid Approach
4. Embedded approach

Wrapper Approach: learning algorithms are used as black box. Computationally very expensive, however performance is superior than filter approach;

Sensitivity Analysis by ANN: Input set= {a, b, c, d, e}

Subset={a, b, c}= 96%

Subset= {a, b, c, d}=94%

Indicates feature d has negative impact, we can drop it

Subset= {a, b, c, e}= 97%

Indicates feature e has positive impact, we can add it.

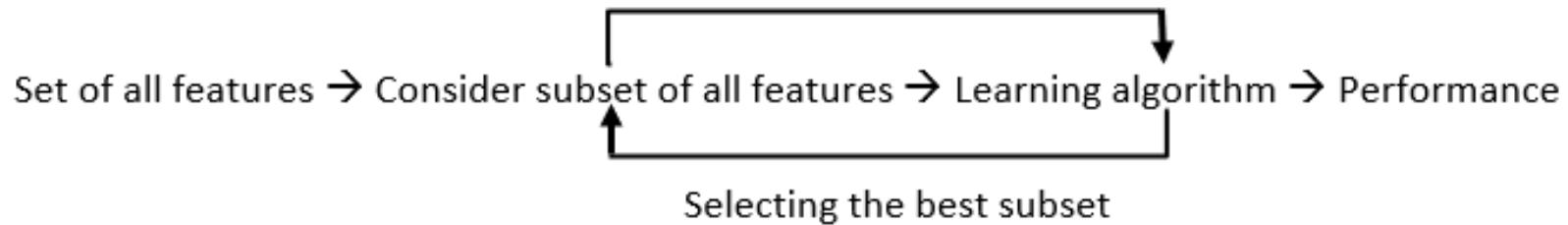
Hence final subset consists of {a, b, c, e}

a	b	c	d	e	class
12	25	42	45	23	1
32	23	23	86	51	1
56	12	14	401	23	0
30	15	63	47	21	0
45	20	54	98	20	1

Feature Selection Approaches

Wrapper Approach:

- The implementation of the wrapper approach is given below:



1. Filter Approach
2. **Wrapper Approach**
3. Hybrid Approach
4. Embedded approach

Some of the algorithm under wrapper approach are as follows:

- **Exhaustive Feature Selection**
- **Forward Feature Selection**
- **Backward Feature Elimination**
- **Recursive Feature Elimination**
- **Bi-directional elimination**

Exhaustive Search

The table gives subset of 3 features out of 5. This procedure is impractical as if we want to choose 12 features out of 24, 2.7 million feature subsets must be evaluated.

Sl. No	F1	F2	F3	F4	F5
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	0	1	1	1	0
5	1	0	0	1	1
6	1	0	1	0	1
7	1	0	1	1	0
8	1	1	0	0	1
9	1	1	0	1	0
10	1	1	1	0	0

Sequential Forward Selection

- First, the best single feature is selected
- Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.
- Next, triplets of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
- This procedure continues until a predefined number of features are selected.

Suppose we are interested to select 3 features out of 5. Feature set= {F1, F2, F3, F4, F5}

Starts with S= {};

1st iteration:

Say ANN is used: {F1} = 60%, {F2} = 52%, {F3} = 52%, {F4} = 53%, {F5} = 70%

S= {F5}

2nd iteration:

{F5, F1}= 85%, {F5, F2}= 82%, {F5, F3}= 80% {F5, F4}= 82%

S= {F5, F1}

3rd iteration:

{F5, F1, F2}= 88%, {F5, F1, F3}= 80%, {F5, F1, F4}= 90%,

Final subset= {F5, F1, F4}

Sequential Forward Selection

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \arg \max_{x \notin Y_k} J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+$; $k = k + 1$
4. Go to 2

- **Drawback: once selected can not be dropped**

Sequential Backward Selection

- First, the criterion function is computed for all n features.
- Then, each feature is deleted one at a time, the criterion function is computed for all subsets with n-1 features, and the worst feature is discarded.
- Next, each feature among the remaining n-1 is deleted one at a time, and the worst feature is discarded to form a subset with n-2 features.
- This procedure continues until a predefined number of features are left.

Suppose we are interested to select 3 features out of 5. Feature set= {F1, F2, F3, F4, F5}

Starts with S= {F1, F2, F3, F4, F5}= 95%

1st iteration: Say ANN is used: {F1, F2, F3, F4} = 96%, {F1, F2, F3, F5} = 94%, {F1, F2, F4, F5} = 95% {F1, F3, F4, F5} = 92%, {F2, F3, F4, F5} = 91%

S= {F1, F2, F3, F4}

2nd iteration: {F1, F2, F3} = 95%, {F1, F2, F4} = 98%, {F1, F3, F4} = 92%, {F2, F3, F4} = 90%

S= {F1, F2, F4}

Sequential Backward Selection

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg \max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-$; $k = k + 1$
4. Go to 2

Drawback: once dropped cannot be taken back

Bi-directional Selection

BDS applies SFS and SBS simultaneously:

SFS is performed from the empty set.

SBS is performed from the full set.

To guarantee that SFS and SBS converge to the same solution:

Features already selected by SFS are not removed by SBS

Features already removed by SBS are not added by SFS

$$S = \{F_1, F_2, F_3, F_4, F_5\}$$

Desired number of features = 3

SFS: $F_1 = 60\%$, $F_2 = 62\%$, $F_3 = 80\%$, $F_4 = 50\%$, $F_5 = 55\%$

$$S_1 = \{F_3\}$$

$$S = \{F_1, F_2, F_4, F_5\} = 86\%$$

SBS: $\{F_1, F_2, F_4\} = 80\%$; $\{F_1, F_2, F_5\} = 82\%$; $\{F_1, F_4, F_5\} = 85\%$; $\{F_2, F_4, F_5\} = 88\%$

$$S = \{F_2, F_4, F_5\}$$

SFS: $F_3 F_2 = 82\%$; $F_3 F_4 = 85\%$; $F_3 F_5 = 80\%$

$$S_2 = \{F_3, F_4\}$$

SBS: $\{F_2, F_5\} = 60\%$; F_5 is deleted

$$S = \{F_2\}$$

SFS: $\{F_2, F_3, F_4\} = 90\%$

Final Subset (S-F) = $\{F_2, F_3, F_5\}$

1. Start SFS with $Y_F = \{\emptyset\}$

2. Start SBS with $Y_B = X$

3. Select the best feature

$$x^+ = \arg \max_{\substack{x \notin Y_{F_k} \\ x \in F_{B_k}}} J(Y_{F_k} + x)$$

$$Y_{F_{k+1}} = Y_{F_k} + x^+$$

4. Remove the worst feature

$$x^- = \arg \max_{\substack{x \in Y_{B_k} \\ x \notin Y_{F_{k+1}}}} J(Y_{B_k} - x)$$

$$Y_{B_{k+1}} = Y_{B_k} - x^-; k = k + 1$$

5. Go to 3

- The problem of sequential forward and sequential backward can be overcome by “Plus-L, minus-R” selection (LRS).
- However its main limitation is the lack of a theory to help choose the optimal values of L and R.

- The drawback of sequential forward and backward selection is called nesting effect. This nesting effect can be overcome by Sequential Floating Selection.
- The drawback of “Plus-L, minus-R” selection (LRS) is also overcome by Sequential Floating Selection.
 1. Sequential floating forward selection
 2. Sequential floating backward selection

Sequential Floating Forward Selection

- Sequential floating forward selection (SFFS) starts from the empty set.
- After each forward step, SFFS performs backward steps as long as the objective function increases.

Step1: Let k=0

Step2: If $k=\text{desired size}$, terminate; otherwise add the most significant feature to the current sub-set of size k . Let $k=k+1$

Step3: Conditionally, remove the least significant feature from the current subset

Step4: If the current subset is the best subset of size $(k-1)$ found so far, let $k=(k-1)$ and go to Step3. Else return the conditionally removed feature and go to Step2.

Consider the feature set = $\{f_1, f_2, f_3, f_4, f_5\}$

Target is to select subset of 2 features

1. $F=\{\}$
 2. The most significant feature is f_3 ; $F= \{f_3\}$
 3. The least significant feature is f_3 ; $F= \{\}$
 4. Removal of f_3 does not improve performance; Hence $F= \{f_3\}$
 5. The most significant feature is f_2 using SFS; $f_3f_2=70\%$, $f_3f_1=50\%$, $f_3f_4=52\%$, $f_3f_5= 60\%$
 $S=\{f_2, f_3\}$
 6. The least significant feature is f_2 ; $F=\{f_3\}$
 7. Removal of f_2 does not improve performance; Hence $F= \{f_3, f_2\}$
 8. $\{f_2, f_3, f_1\}= 72\%$, $\{f_2, f_3, f_4\}= 71\%$, $\{f_2, f_3, f_5\}= 75\%$
 $S= \{f_2, f_3, f_5\}$
- SBS: $\{f_2, f_3\}= 70\%$, $\{f_2, f_5\}= 65\%$, $\{f_3, f_5\}= 72\%$

Hence the optimal subset is $F= \{f_2, f_3, f_5\}$

Sequential Floating Forward Selection

1. $Y = \{\emptyset\}$

2. Select the best feature

$$x^+ = \arg \max_{x \notin Y_k} J(Y_k + x)$$

$$Y_k = Y_k + x^+; k = k + 1$$

3. Select the worst feature*

$$x^- = \arg \max_{x \in Y_k} J(Y_k - x)$$

4. If $J(Y_k - x^-) > J(Y_k)$ then

$$Y_{k+1} = Y_k - x^-; k = k + 1$$

Go to step 3

Else

Go to step 2

*Notice that you'll need to do book-keeping to avoid infinite loops

Sequential Floating Backward Selection

- Sequential floating backward selection (SFBS) starts from the full set.
- After each backward step, SFBS performs forward steps as long as the objective function increases.

Step1: Let $k=n$ (f_1, f_2, f_3, f_4)

Step2: If $k=\text{desired size}$, terminate; otherwise remove the least significant feature from the current subset of size k . Let $k=k-1$. (f_4)

Step3: Conditionally, add the most significant feature from the features not in the current subset. (f_1, f_2, f_3) + (f_4)

Step4: If the current subset is the best subset of size $(k+1)$ found so far, let $k=(k+1)$ and go to Step3. Else remove the conditionally added feature and go to Step 2.
((f_1, f_2, f_3) and go to Step 2)

Sequential Floating Backward Selection

Step1: Let k=n (f1, f2, f3, f4)

Step2: If k=desired size, terminate; otherwise remove the least significant feature from the current subset of size k. Let k=k-1. (f4)

Step3: Conditionally, add the most significant feature from the features not in the current subset. (f1, f2, f3) + (f4)

Step4: If the current subset is the best subset of size (k+1) found so far, let k=(k+1) and go to Step3. Else remove the conditionally added feature and go to Step 2.

((f1, f2, f3) and go to Step 2)

Friday, November 25, 2022

Consider the feature set = {f1, f2, f3, f4, f5}

Target is to select subset of 2 features

1. SBS: F={f1, f2, f3, f4, f5}= 64%
{f1, f2, f3, f4}= 60%; {f1, f2, f3, f5}= 62; {f1, f2, f4, f5}= 65%; {f1, f3, f4, f5}= 68%; {f2, f3, f4, f5}=55%

SB {f1, f3, f4, f5}= 68%; F= {f2}

2. Most significant feature f2 from set F; adding this to subset does not improve accuracy as {f1, f2, f3, f4, f5}= 64%
3. Say least significant is f5
1. The most significant feature is f3; F= {f3}
2. The least significant feature is f3; F= {}
3. Removal of f3 does not improve performance; Hence F= {f3}
4. The most significant feature is f2 using SFS; f3f2=70%, f3f1=50%, f3f4=52%, f3f5= 60%
S={f2, f3}
6. The least significant feature is f2; F={f3}
7. Removal of f2 does not improve performance; Hence F= {f3, f2}
8. {f2, f3, f1}= 72%, {f2, f3, f4}= 71%, {f2, f3, f5}= 75%

Feature Selection Approaches: Hybrid Approach

1. Filter Approach
2. Wrapper Approach
3. **Hybrid Approach**
4. Embedded approach

Hybrid Approach: takes advantages of both filter and wrapper approaches.

Filter Approach---{S1}---Wrapper Approach----(S2) | S1>S2

Wrapper Approach---{S1}--- Filter Approach-----(S2) | S1>S2

Feature Selection Approaches: Embedded approach

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. **Embedded approach**

Embedded Approach:

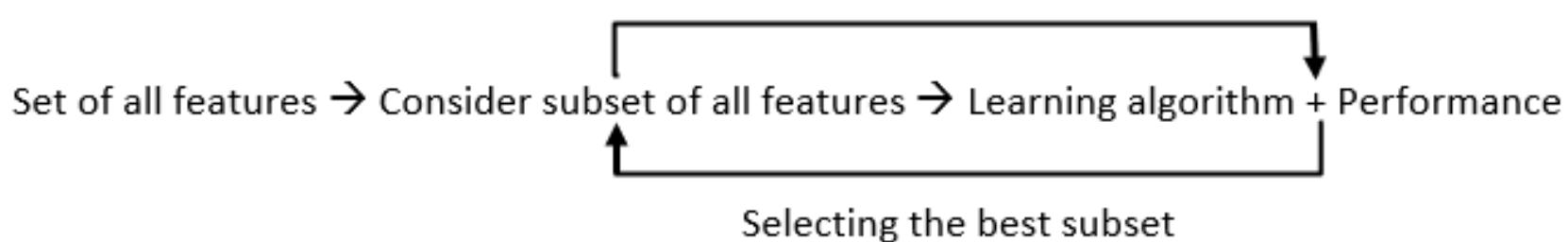
- These methods encompass the benefits of both the wrapper and filter methods by including interactions of features but also maintaining reasonable computational cost.
- Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extracts those features which contribute the most to the training for a particular iteration.
- Similar to wrapper approach but performs feature selection and classification simultaneously.
- These methods are faster like those of filter methods and more accurate than the filter methods and take into consideration a combination of features as well.

Feature Selection Approaches: Embedded approach

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. **Embedded approach**

Embedded Approach: Some of the algorithms under embedded approach are given below:

- **LASSO Regularization (L1)**
- **Random Forest Importance**
- **Extra Tree Classifier**



Feature Selection Methods

How to choose a Feature Selection Method (filter-based feature selection)

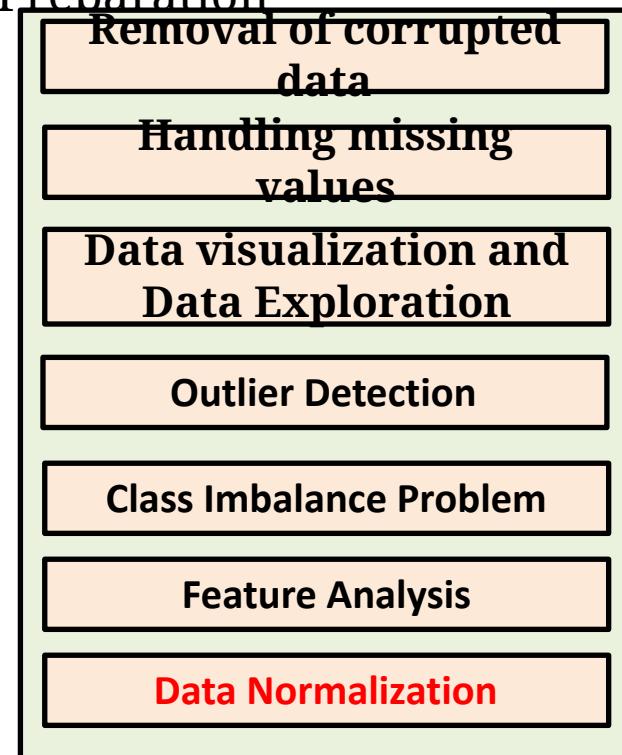
Input Variable	Output Variable	Feature Selection technique
Numerical	Numerical	<ul style="list-style-type: none">• Pearson's correlation coefficient (For linear Correlation).• Spearman's rank coefficient (for non-linear correlation).
Numerical	Categorical	<ul style="list-style-type: none">• ANOVA correlation coefficient (linear).• Kendall's rank coefficient (nonlinear).
Categorical	Numerical	<ul style="list-style-type: none">• Kendall's rank coefficient (linear).• ANOVA correlation coefficient (nonlinear).
Categorical	Categorical	<ul style="list-style-type: none">• Chi-Squared test (contingency tables).• Mutual Information

Feature weighting

- Normalized Max Filter (NMF)
- Normalizing Range Filter (NRF)
- Normalizing Linear Filter (NLF)
- Normalizing Sigmoid Filter (NSF)
- Monotonically Decreasing Function (MDF)
- Sequential Ranking Weighting (SRW)
- Correlation-based Feature Weighting (CFW)
- Feature weighted Naive Bayes (FWNB)
- Gain Ratio-based Feature Weighting method (GRFW)
- Decision Tree-based Feature Weighting method (DTFW)
- Kullback-Leibler Measure-based Feature Weighting method (KLMFW)
- Evolution-based Feature Weighting Wrapper (EFWW)
- Conditional Log Likelihood-based Feature Weighting Wrapper (CLLFWW)
- Mean Squared Error-based Feature Weighting Wrapper (MSEFWW)
- Feature sensitivity using ANN
- Feature relevance using ANN
- Feature activity using ANN
- Feature saliency using ANN

Machine Learning Model

Pre-processing/ Data Preparation



Data Normalization

Changes the values to a common scale.

Normalization gives equal weights/importance to each variable so that no single variable steers model performance in one direction just because they are bigger numbers.

For machine learning, every dataset does not require normalization. It is required only when features have different ranges.

It improves (significantly) the performance of some machine learning algorithms and does not work at all for others.

Age	Salary	Experience
30	200000	H
50	500000	H
60	20000	L
100	70000	L

Decision is biased towards **Salary**

Age	Salary	Experience
1	1	H
3	b	H
2	1	L
a	2	L

Values can go upto a and b (a and b are equal or very closed)

Data Normalization

1. **Min-max normalization** is the simplest of all methods.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

X1	X2
10	20
5	10
6	12
7	15
25	12

$$(10-5)/(25-5)=5/20= 0.4$$

$$(5-5)/(25-5)= 0/20=0$$

$$(6-5)/(25-5)=1/20=0.05$$

$$(7-5)/(25-5)=2/20=0.10$$

$$(25-5)/(25-5)=20/20=1$$

2. **Mean normalization** uses the mean of the observations in the transformation process

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

X1	X2
0.4	1
0	0
0.05	0.2
0.10	0.5
1	0.2

3. **Z-score normalization/standardization** uses Z-score and is widely used in machine learning algorithm.

$$z = \frac{x - \mu}{\sigma}$$

z is the standard score, μ is the population mean and σ is the population standard deviation

Data Normalization

1. **Min-max normalization** is the simplest of all methods.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

X1	X2
10	20
5	10
6	12
7	15
25	12

$$(10-10.6)/(25-5)=-.6/20=-0.03$$

$$(5-10.6)/(25-5)=-5.6/20=-0.28$$

$$(6-10.6)/(25-5)=-4.6/20=-0.23$$

$$(7-10.6)/(25-5)=-3.6/20=-0.18$$

$$(25-10.6)/(25-5)=14.4/20=0.72$$

2. **Mean normalization** uses the mean of the observations in the transformation process

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

X1	X2
-0.03	
-0.28	
-0.23	
-0.18	
0.72	

3. **Z-score normalization/standardization** uses Z-score and is widely used in machine learning algorithm.

$$z = \frac{x - \mu}{\sigma}$$

z is the standard score, μ is the population mean and σ is the population standard deviation

Data Normalization

3. Z-score normalization/standardization:

$$\text{New value} = (x - \mu) / \sigma$$

where:

x : Original value

μ : Mean of data

σ : Standard deviation of data

$$\mu = \frac{\text{sum of the terms}}{\text{number of terms}} = 21.2$$

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} = 29.8$$

$$\text{New value} = (x - \mu) / \sigma$$

$$\text{New value of } 3 = (3 - 21.2) / 29.8$$

$$\text{New value of } 3 = -0.61$$

Data	Z-Score Normalized Value
3	-0.61
5	-0.54
5	-0.54
8	-0.44
9	-0.41
12	-0.31
12	-0.31
13	-0.28
15	-0.21
16	-0.17
17	-0.14
19	-0.07
22	0.03
24	0.09
25	0.13
134	3.79

Data Normalization

3. Z-score normalization/standardization

- The mean of the normalized values is **0** and the standard deviation of the normalized values is **1**.
- The normalized values represent the amount of standard deviations that the original value is from the mean.

For example:

- The first value in the dataset is **0.61** standard deviations below the mean.
- The second value in the dataset is **0.54** standard deviations below the mean.
- The last value in the dataset is **3.79** standard deviations above the mean.

Benefits:

- The benefit of performing this type of normalization is that the clear outlier in the dataset (134) has been transformed in such a way that it's no longer a massive outlier.

Data Normalization

The Robust Scaling:

- Each feature of the dataset is scaled by subtracting the median and then dividing by the interquartile range.
- The interquartile range (IQR) is defined as the difference between the third and the first quartile and represents the central 50% of the data. Mathematically the robust scaler can be expressed as:

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

- where $Q_1(x)$ is the first quartile of the attribute x , $Q_2(x)$ is the median, and $Q_3(x)$ is the third quartile.
- This method comes in handy when working with datasets that contain many **outliers** because it uses statistics that are robust to **outliers**.

Data Normalization

Date: 4, 6, 8, -20, 10, 12, 200, 15, 13, 18, 20, 14

Data: -20, 4, 6, 8, 10, 12, 13, 14, 15, 18, 20, 200

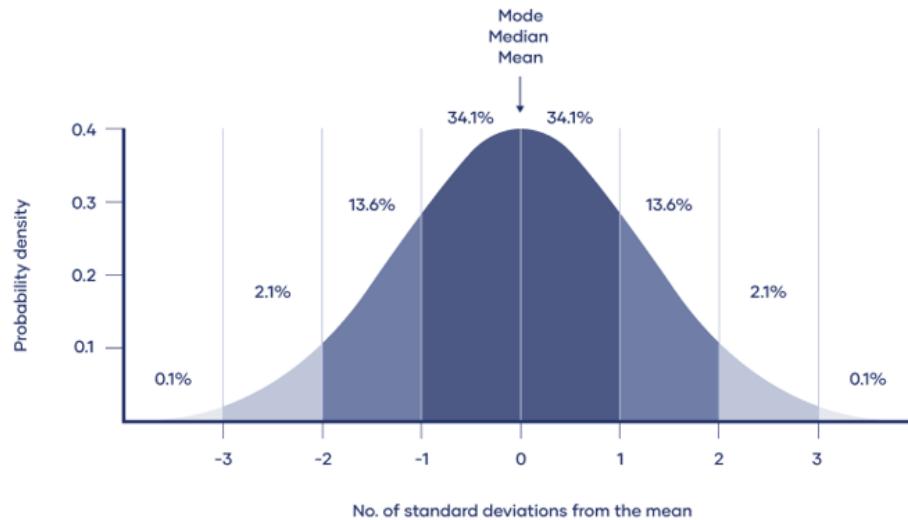
$$\text{Median} = (12+13)/2 = 12.5$$

Data: -20, 4, 6, 8, 10, 12, 13, 14, 15, 18, 20, 200

$$Q_1 = 6; \quad Q_3 = 18; \quad IQR = 18 - 6 = 12$$

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

Standard normal distribution



Data Normalization

The Robust Scaling:

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

$$Q_2(x) = \frac{4+5}{2} = 4.5$$

$$Q_3(x) = 6.5$$

$$Q_1(x) = 2.5$$

$$1 = \frac{1-4.5}{6.5-2.5} = -3.5/4 = -0.875$$

$$30 = \frac{30-4.5}{6.5-2.5} = 25.5/4 = 6.375$$

variable1	variable2
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8

Variable 1	Variable 1
-0.875	-0.875
-0.625	-0.625
-0.375	-0.375
-0.125	-0.125
0.125	0.125
0.375	0.375
0.625	0.625
6.375	0.875

Data Normalization

Min-max normalization

The **min-max scaling** shifts the variable 1 towards 0 due to the presence of an **outlier** as compared with variable 2 where the points are evenly distributed in a range from 0 to 1.

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

	variable1	variable2
0	0.000000	0.000000
1	0.034483	0.142857
2	0.068966	0.285714
3	0.103448	0.428571
4	0.137931	0.571429
5	0.172414	0.714286
6	0.206897	0.857143
7	1.000000	1.000000

Data Normalization

Z-score normalization

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

	var1	var2
0	-0.710	-1.527
1	-0.596	-1.091
2	-0.482	-0.654
3	-0.369	-0.218
4	-0.255	0.218
5	-0.142	0.654
6	-0.028	1.091
7	2.585	1.526

Data Normalization

Min-max normalization, Z-score normalization and The Robust Scaling

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

	variable1	variable2
0	0.000000	0.000000
1	0.034483	0.142857
2	0.068966	0.285714
3	0.103448	0.428571
4	0.137931	0.571429
5	0.172414	0.714286
6	0.206897	0.857143
7	1.000000	1.000000

	var1	var2
0	-0.710	-1.527
1	-0.596	-1.091
2	-0.482	-0.654
3	-0.369	-0.218
4	-0.255	0.218
5	-0.142	0.654
6	-0.028	1.091
7	2.585	1.526

Variable 1	Variable 1
-0.875	-0.875
-0.625	-0.625
-0.375	-0.375
-0.125	-0.125
0.125	0.125
0.375	0.375
0.625	0.625
6.375	0.875

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

New value = $(x - \mu) / \sigma$

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

Data Normalization

Classification of Diabetes data using k-NN

Number of sample 768

Number of independent parameter = 8

Number of dependent parameter= 1

Binary Classification problem

Training = 70%

Training = 30%

Number of sample 703

Number of independent parameter = 8

Number of dependent parameter= 1

Binary Classification problem

Training = 70%

Training = 30%

Normalization	Accuracy
No normalization	73%
Min-Max	75%
Standardization	74%
Robust Scaler	76%

Normalization	Accuracy
No normalization	76%
Min-Max	73%
Standardization	73%
Robust Scaler	75%

Data Normalization

Normalization vs Standardization

Normalization	Standardization
Minimum and maximum value of features are used for scaling.	Mean and standard deviation is used for scaling.
It is used when features are of different scales.	It is used when we want to ensure zero mean and unit standard deviation.
Scales values between [0, 1] or [-1, 1].	It is not bounded to a certain range.
It is really affected by outliers.	It is much less affected by outliers.
It is useful when we don't know about the distribution.	It is useful when the feature distribution is Normal or Gaussian.
It is often called as Scaling Normalization.	It is often called as Z-Score Normalization.

Data Normalization

Gradient Descent Based Machine learning algorithms like [linear regression](#), [logistic regression](#), [neural network](#) etc converge more quickly towards the minima if features are on a similar scale.

Distance based Machine learning algorithms like [KNN](#), [K-means](#), and [SVM](#) most drastically improve the performance minima if features are on a similar scale.

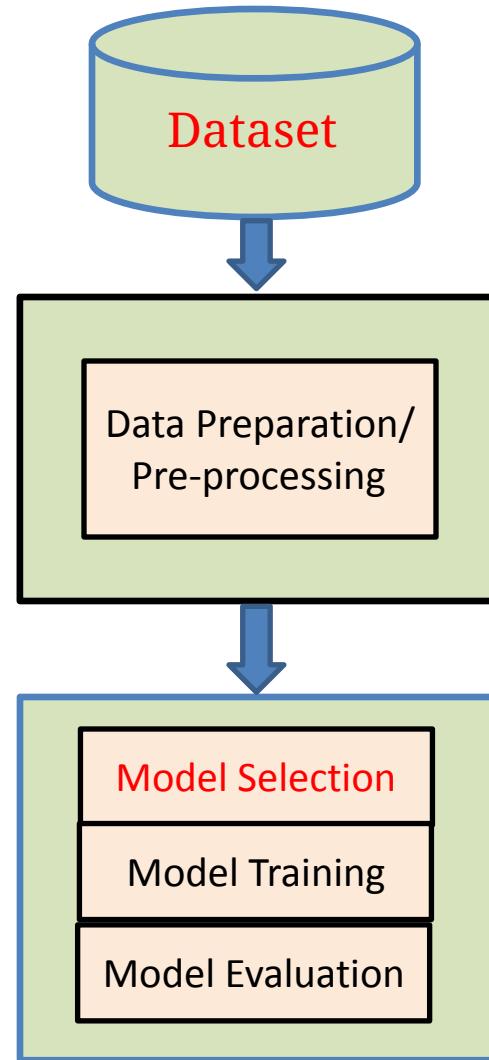
Tree based Machine learning algorithms like DT are insensitive to data normalization.

Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. K-Nearest Neighbors and Neural Networks.

Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true.

Data normalization reduces the variance and applies equal weights to all features; therefore, a lot of important information is lost in the process.

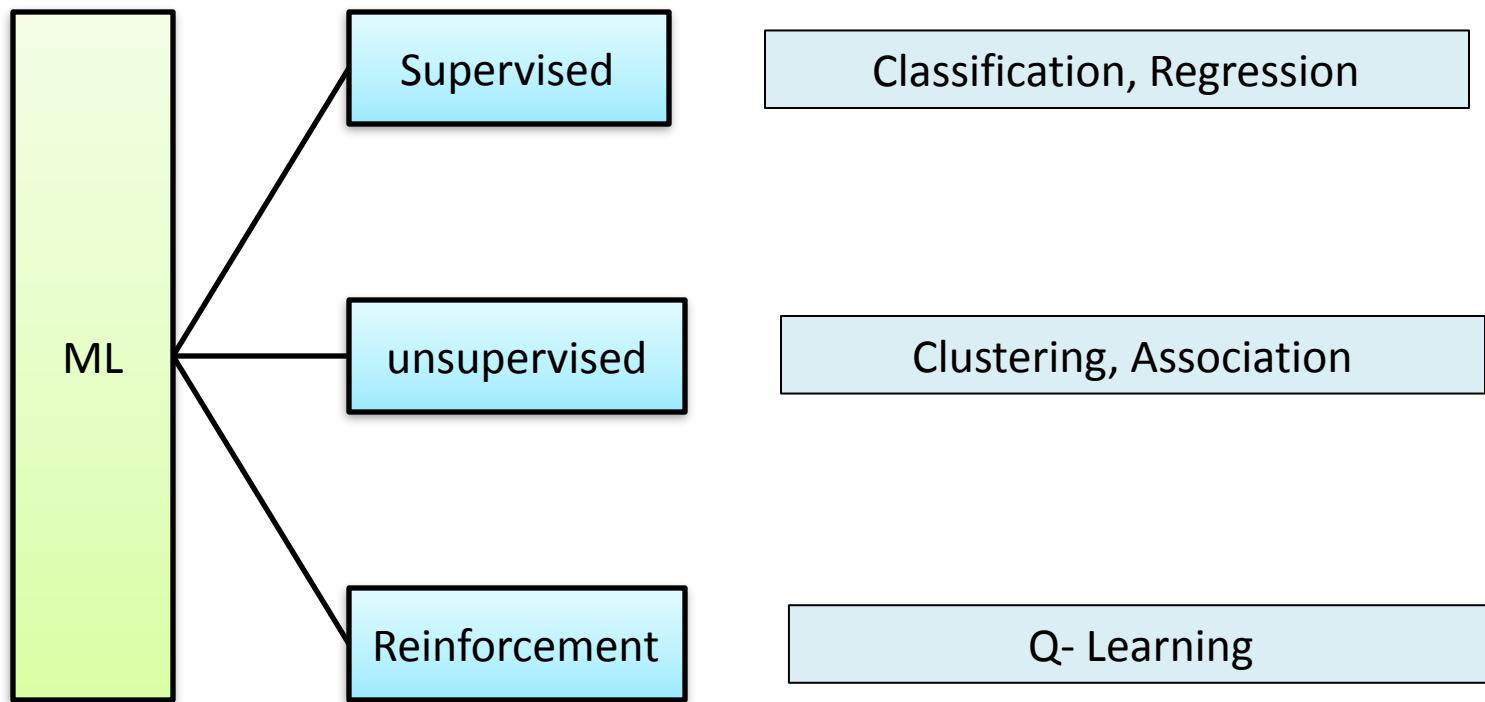
ML Model



Modelling & Evaluation

MODEL SELECTION

Classification of ML Algorithm



MODEL SELECTION

Classification of ML Algorithms

Supervised

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- K- Nearest Neighbour (K-NN)
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

MODEL SELECTION

Classification of ML Algorithms

Unsupervised

- Clustering
 - K-Means
 - K-Mediod
 - CURE
 - BIRCH
- Association
 - Apriori Algorithm
 - Predictive Apriori Algorithm
 - Tertius Algorithm
 - E clat

Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

MODEL SELECTION

Classification of ML Algorithms

Reinforcement

- Markov Decision Process (MDP)
- Q learning: Deep-Q-Neural Network (DQN)
- State Action Reward State Action (SARSA)

BP	Heart Beat	Weight	Feedback
120	70	50	reward
125	65	60	penalty
130	59	52	penalty
150	78	70	penalty
135	66	85	reward
125	75	82	reward
120	76	90	reward

MODEL SELECTION

The most important two factors to select the model for solving a machine learning problem are

- **The kind of problem we want to solve using machine learning**
- **The nature of the underlying data.**

**The kind of problem we want to solve
using machine learning**

Prediction of categorical values or discrete values (classification)

Black Box: k-NN, Naïve Bayes, ANN, SVM, Random Forest

White Box: Decision Tree, Rule extraction from Neural Network

BP	Heart Beat	Weigh t	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N

```
if (credit_history = 'existing paid'  
and credit_amount <=12204 )  
then class="good"  
else class="bad"
```

MODEL SELECTION

The kind of problem we want to solve
using machine learning

Prediction of continuous values (Regression)

Linear Regression, Logistic Regression, Polynomial Regression, ANN, Ridge Regression, LASSO Regression, Elastic Net Regression

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Descriptive: basket analysis for transactional data

Clustering

Spherical Shape: k-means, k-mediod

Non Spherical Shape: Clustering Using Representatives (CURE)

MODEL SELECTION

The most important two factors to select the model for solving a machine learning problem are

- The kind of problem we want to solve using machine learning
- The nature of the underlying data

The nature of the underlying data

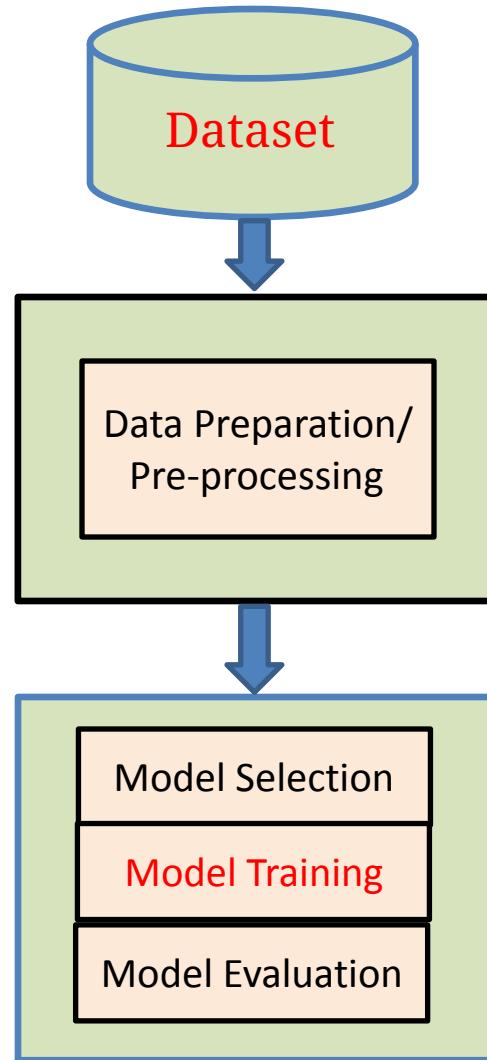
Various statistical measures: mean, median, variance, correlation between variables

visualization tools: Histograms, scatterplot

The training data size is an important factor to be considered, if the training dataset is small: Naïve Bayes

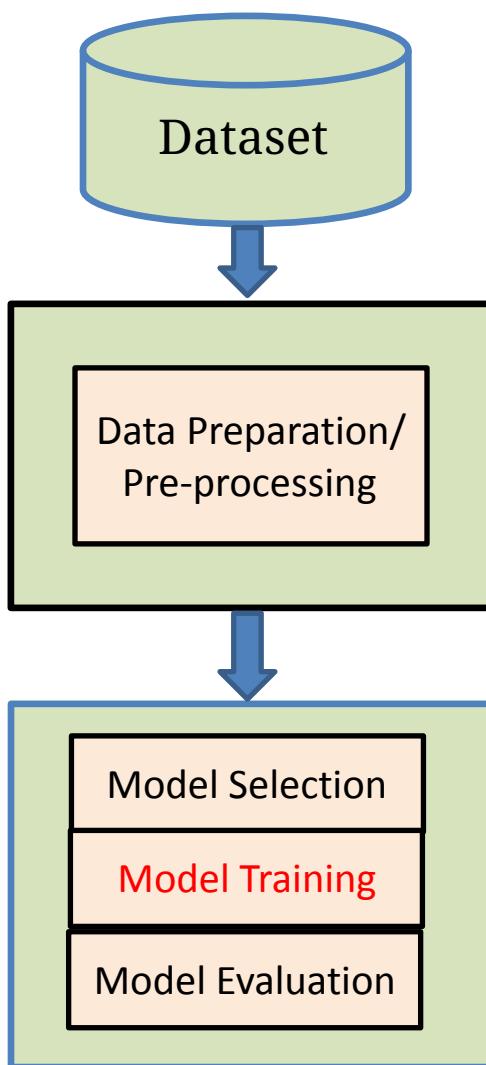
If the training dataset is large: Logistic Regression, SVM, ANN

ML Model



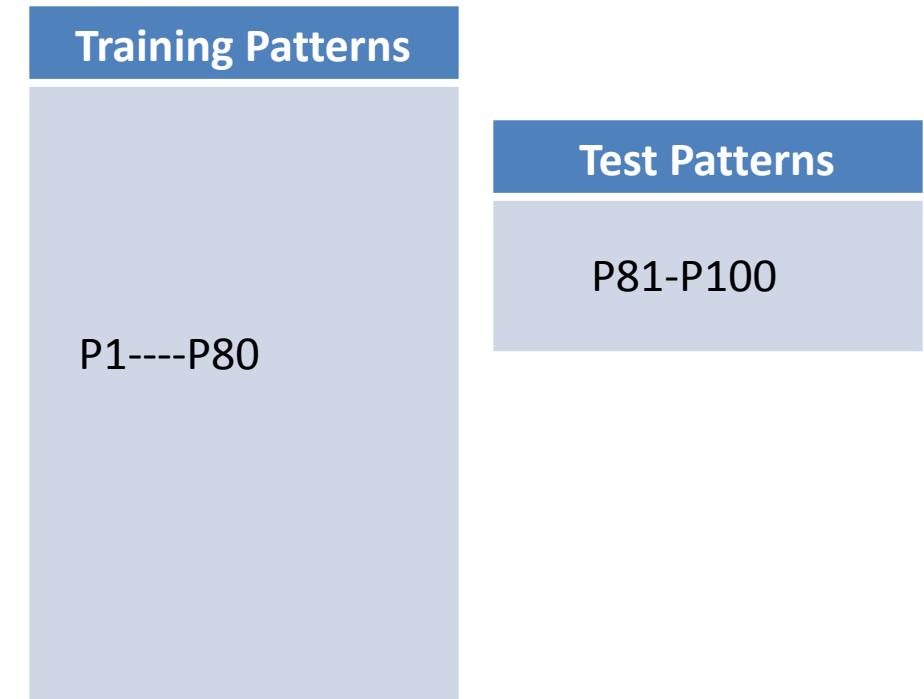
Modelling & Evaluation

Model Training



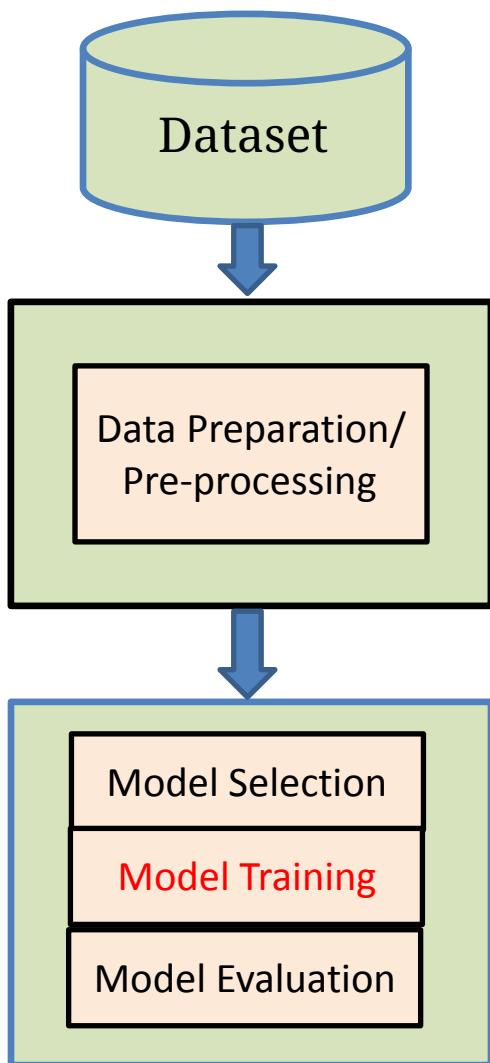
Training for supervised learning: i. Holdout method

- Partition can be: 80-20 or 70-30
 - Sometimes partition into 3 partitions: training, validation, testing
 - **Suffers extremely for imbalance data**
1. Biased Model
 2. Erroneous Model



Modelling & Evaluation

Model Training



i. Holdout method

Advantages:

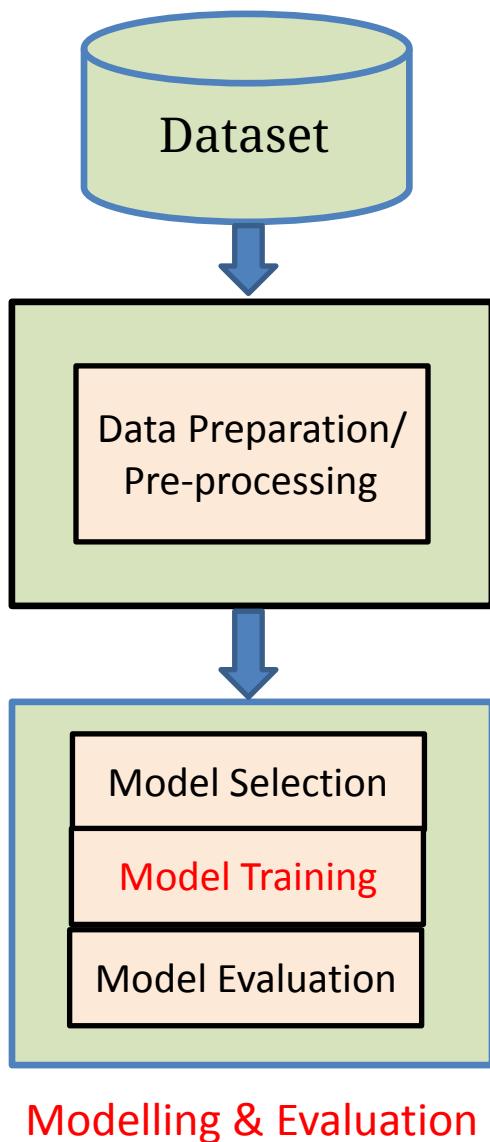
- Its time complexity is less.
- Therefore the hold-out method is good to use when we have a very large dataset.
- An initial model can be built.

Disadvantages:

- The hold-out method score dependent on how the data is split into train and test sets.
- It is less generalized.
- It may be suffering from overfitting problem.
- It may be suffering from imbalanced dataset

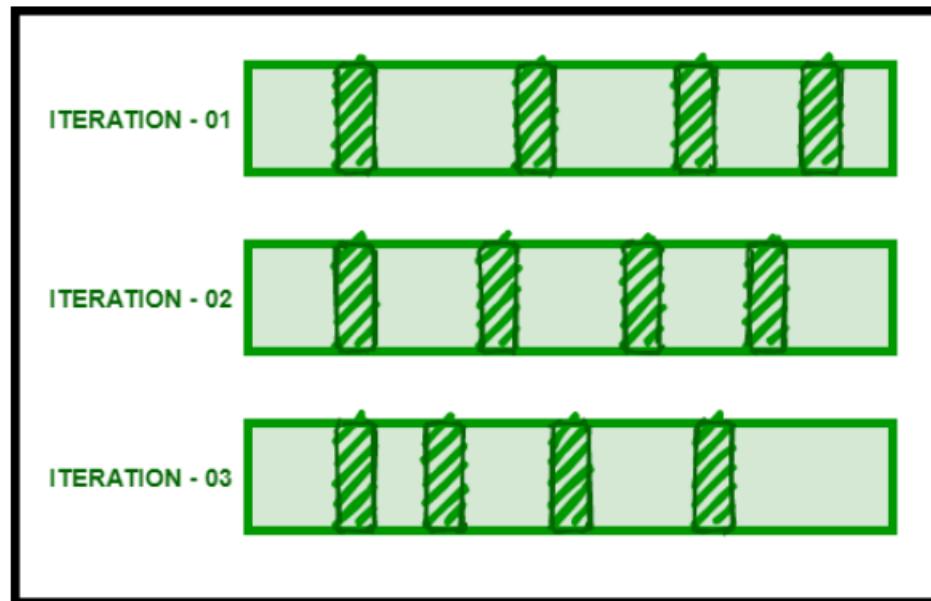
Modelling & Evaluation

Model Training



Training for supervised learning: **ii. Repeated Holdout method**

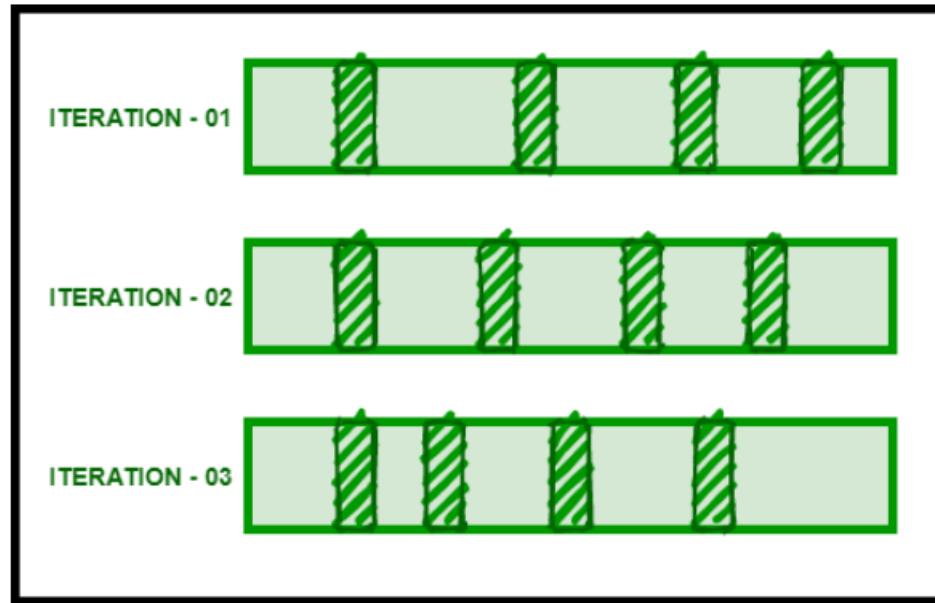
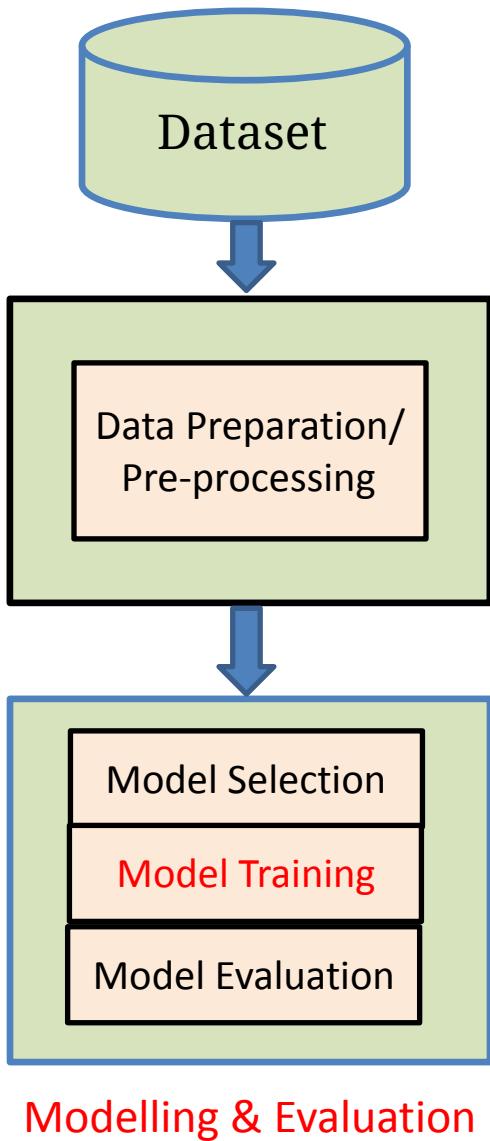
- it is the repeated execution of the holdout method.
- This method can be repeated — ‘K’ times/iterations.
- Random sampling of the dataset is employed.
- Let we repeat the holdout method for 3 iterations.



The shaded portions represent test sets and the unshaded portions training sets.

Model Training

ii. Repeated Holdout method



Accuracy of

iteration 01= S1

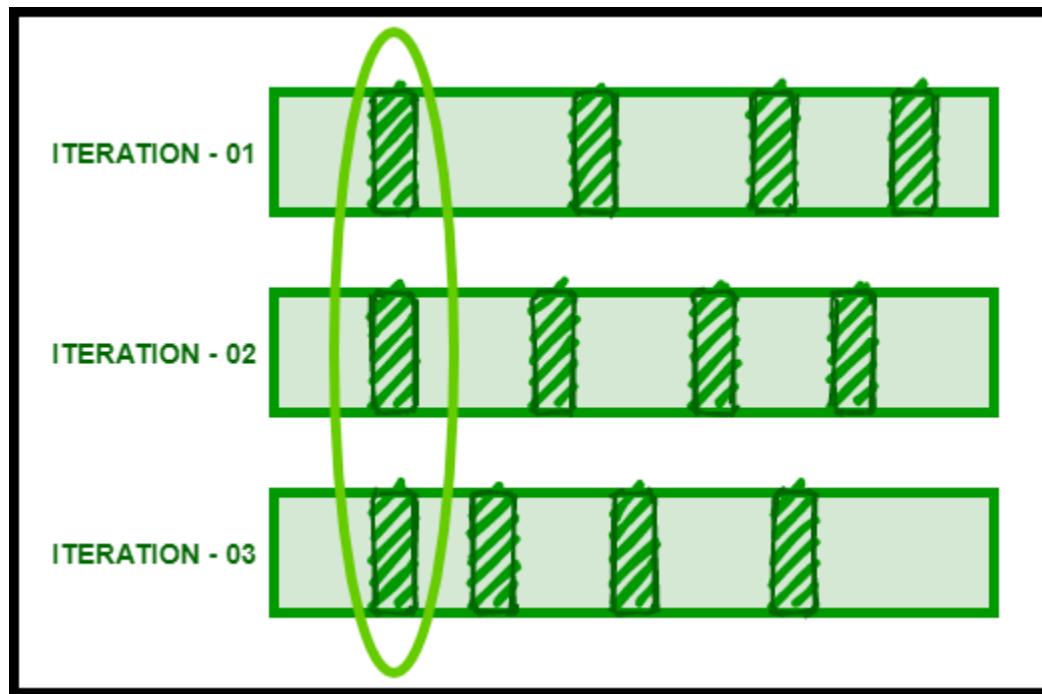
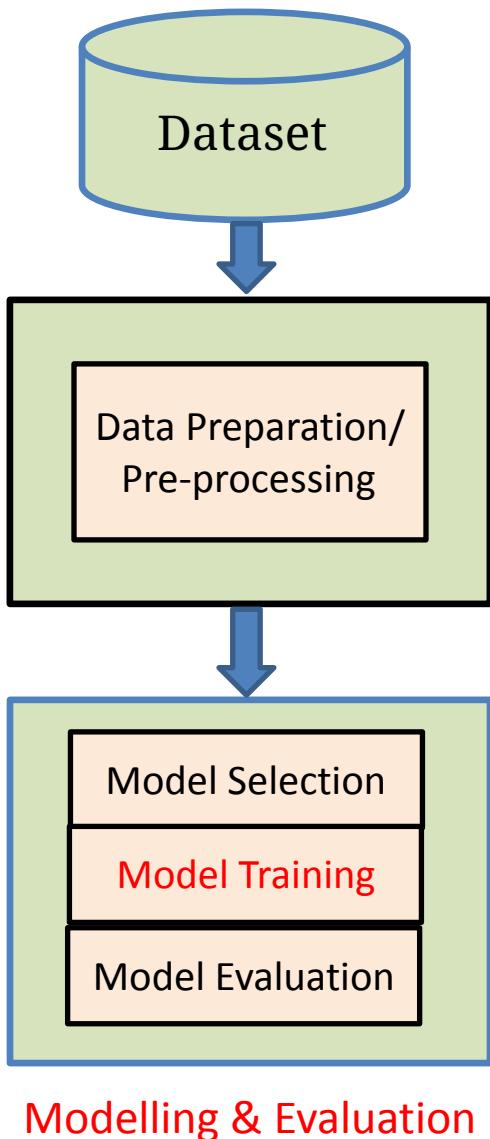
iteration 02= S2

iteration 03= S3

Final Accuracy= $(S1+S2+S3)/3$ (it can be any performance measure)

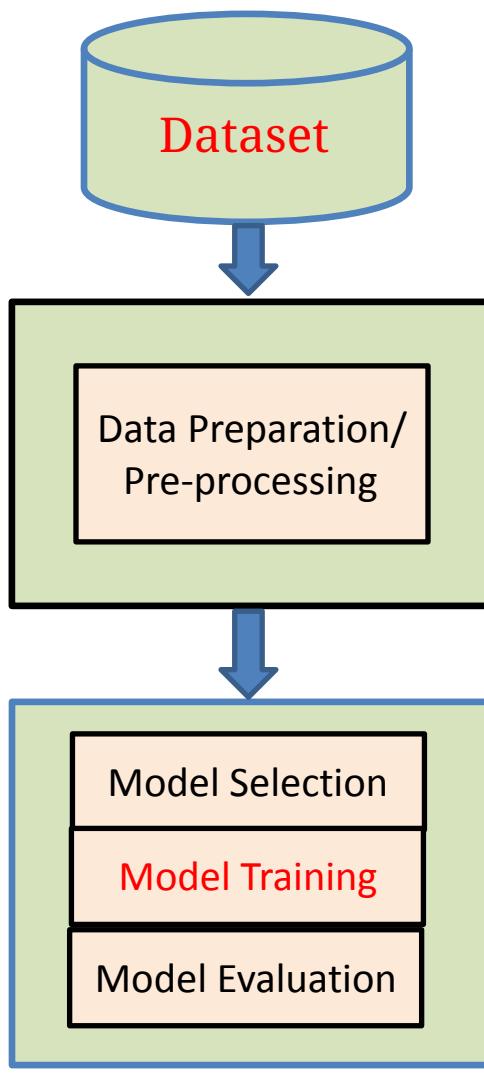
Model Training

ii. Repeated Holdout method: Drawback



Overlapping test set problem

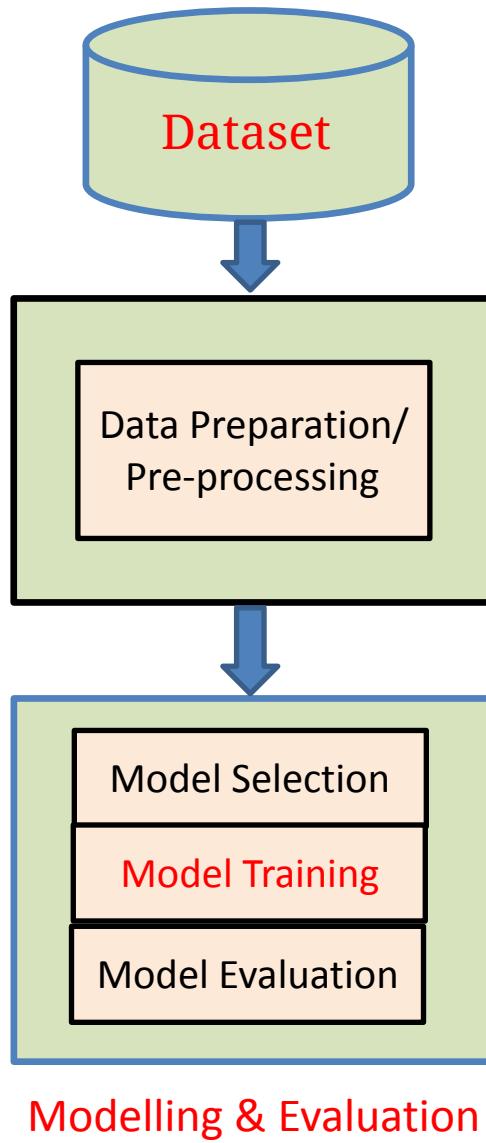
Since we partition the dataset randomly into a training set and test set, **there are some data items/examples that could not be placed in the training set at all**



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P41----P50	
P51----P60	
P61----P70	
P71----P80	
P81----P90	
P91----P100	



iii. K-fold Cross-validation method:

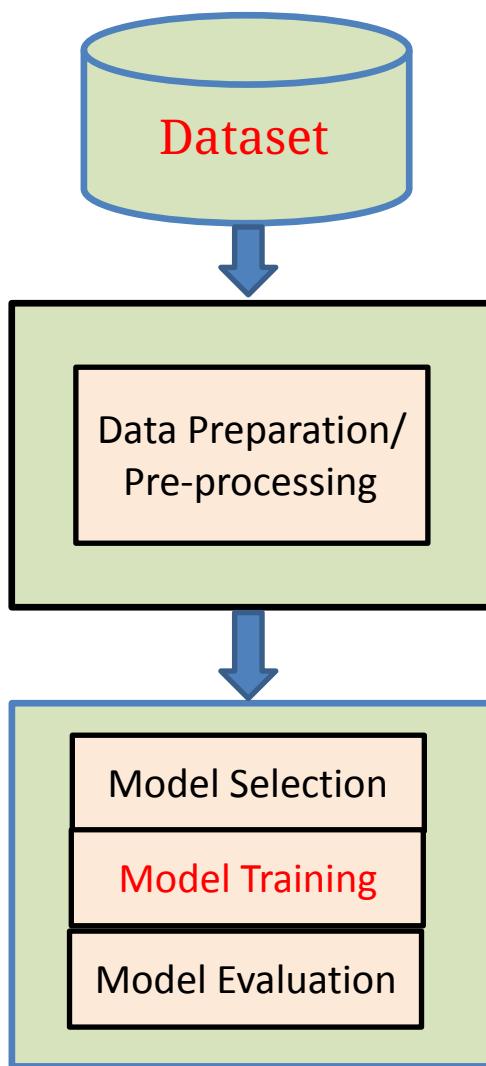
a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns

- P1----P10
- P11----P20
- P21----P30
- P31----P40
- P41----P50
- P51----P60
- P61----P70
- P71----P80
- P81----P90**

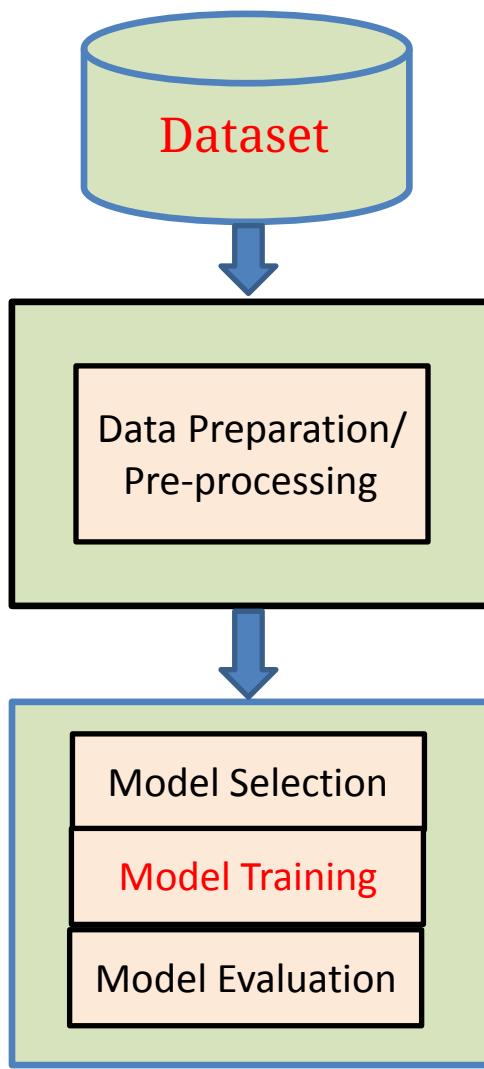
Test Patterns



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
 - b. Leave-One-Out Cross-Validation

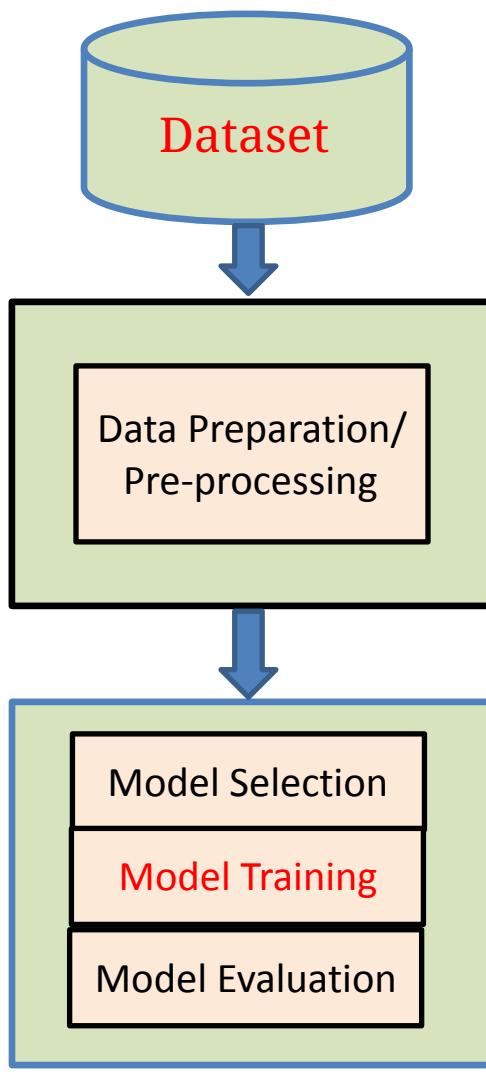
Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P41----P50	
P51----P60	
P61----P70	
P71----P80	
P91----P100	P81----P90



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1---P10	
P11---P20	
P21---P30	
P31---P40	
P41---P50	
P51---P60	
P61---P70	
P81---P90	P71---P80
P91---P100	

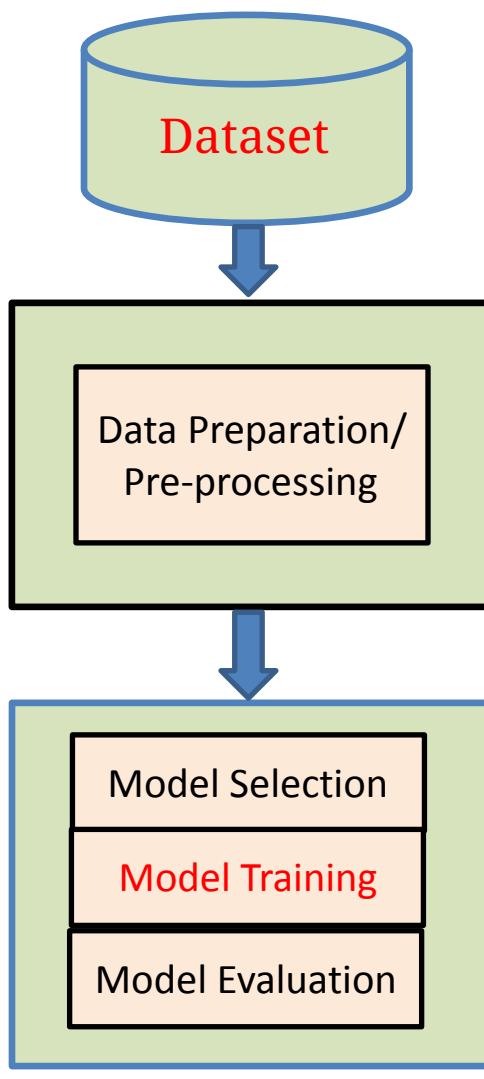


iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P11----P20
P21----P30
P31----P40
P41----P50
P51----P60
P71----P80
P81----P90
P91----P100

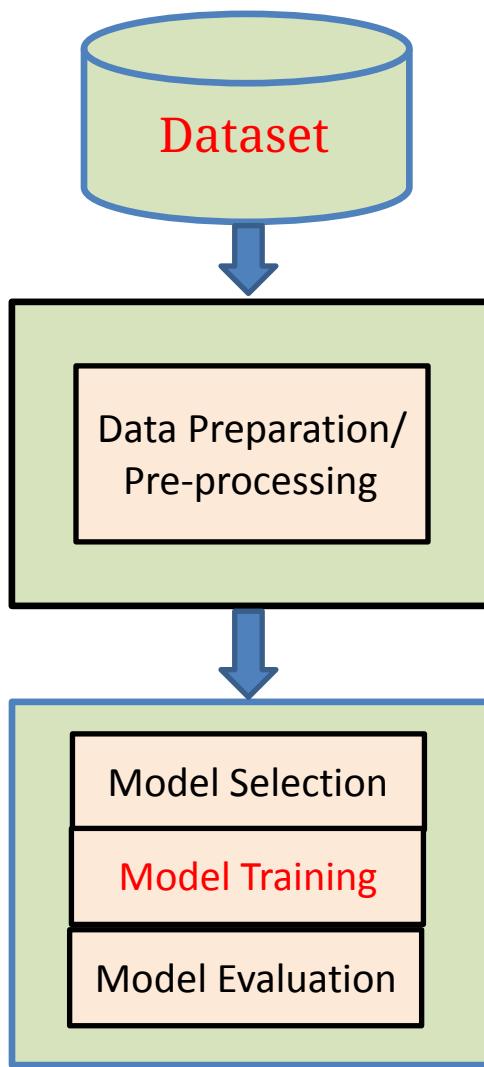
Test Patterns
P61----P70



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P41----P50	
P51----P60	
P61----P70	
P71----P80	
P81----P90	
P91----P100	

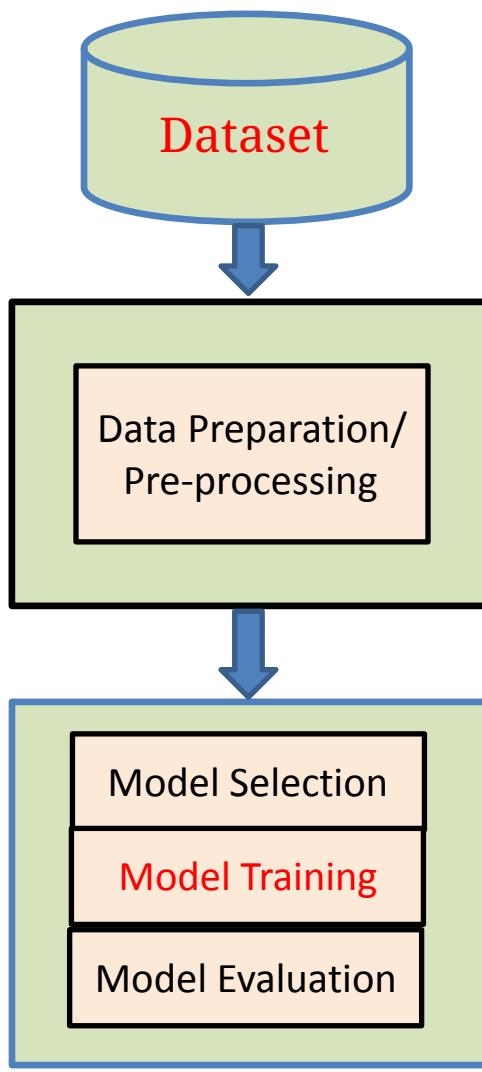


iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P51----P60	
P61----P70	
P71----P80	
P81----P90	
P91----P100	
	P41----P50



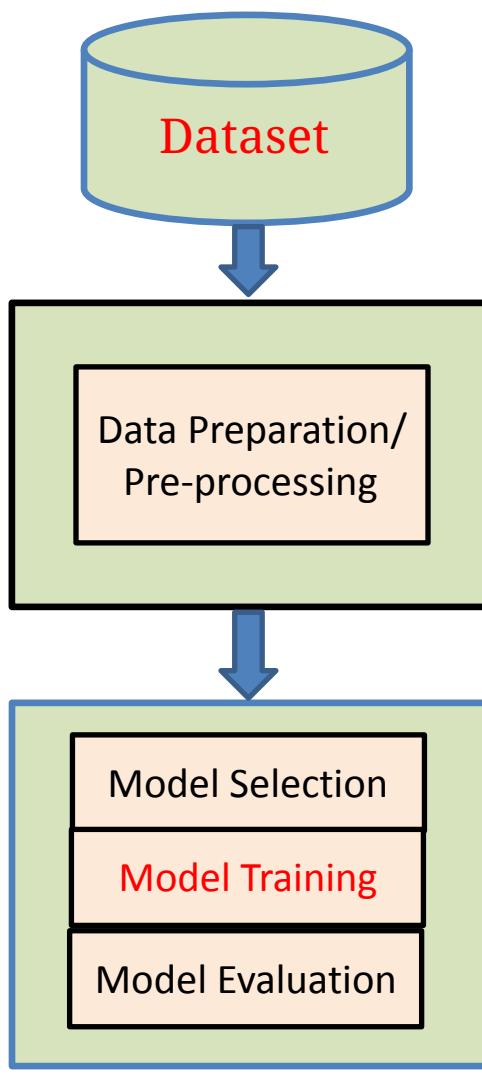
iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P11----P20
P21----P30
P41----P50
P51----P60
P61----P70
P71----P80
P81----P90
P91----P100

Test Patterns
P31----P40



iii. K-fold Cross-validation method:

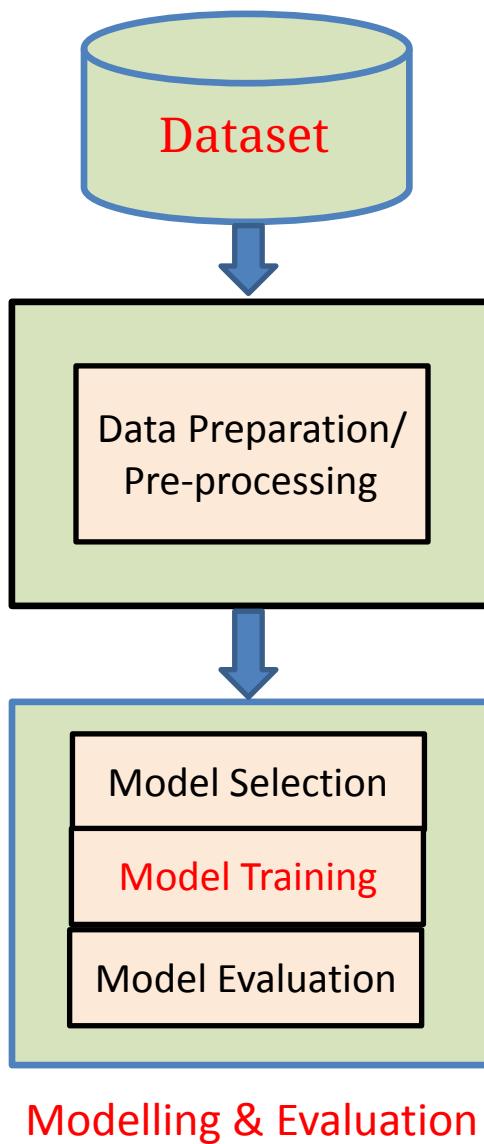
a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1---P10
P11---P20
P31---P40
P41---P50
P51---P60
P61---P70
P71---P80
P81---P90
P91---P100

Test Patterns
P21---P30

Modelling & Evaluation



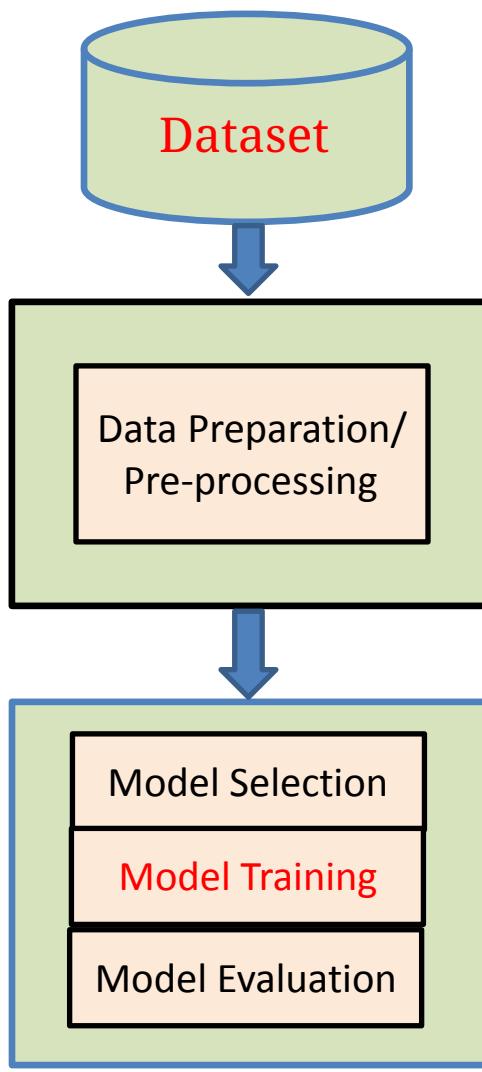
iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P21----P30
P31----P40
P41----P50
P51----P60
P61----P70
P71----P80
P81----P90
P91----P100

Test Patterns
P11----P20

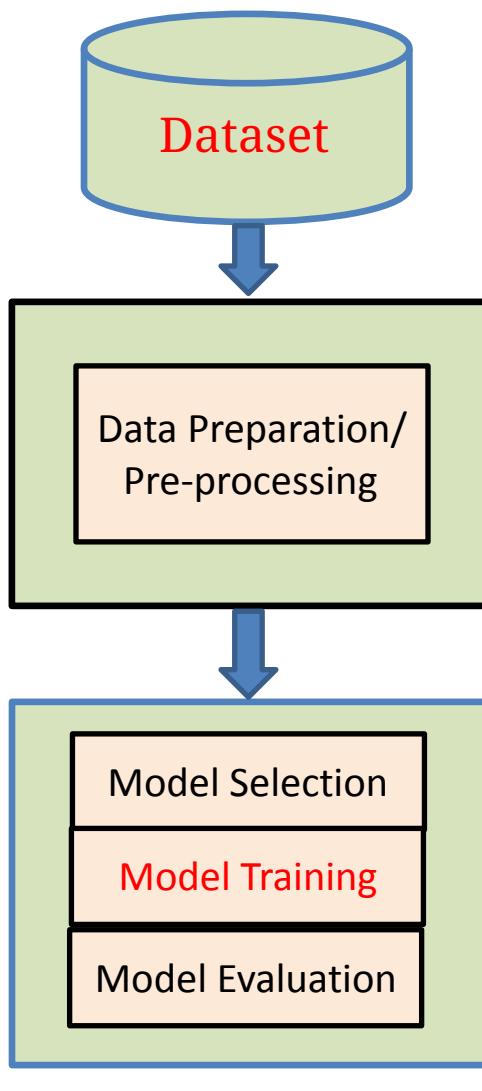


iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
 - b. Leave-One-Out Cross-Validation

- P11----P20
- P21----P30
- P31----P40
- P41----P50
- P51----P60
- P61----P70
- P71----P80
- P81----P90
- P91----P100

Test Patterns



iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

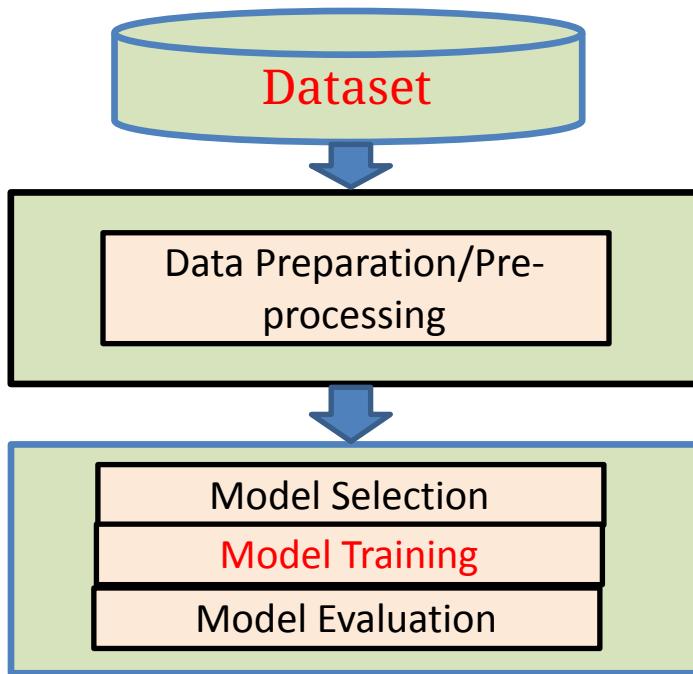
b. Leave-One-Out Cross-Validation

Training Patterns
P1---P10
P11---P20
P21---P30
P31---P40
P41---P50
P51---P60
P61---P70
P71---P80
P81---P90
P91---P100

Test Patterns
P91---P100
P81---P90
P71---P80
P61---P70
P51---P60
P41---P50
P31---P40
P21---P30
P11---P20
P1---P10

Model Training

iii. K-fold Cross-validation method

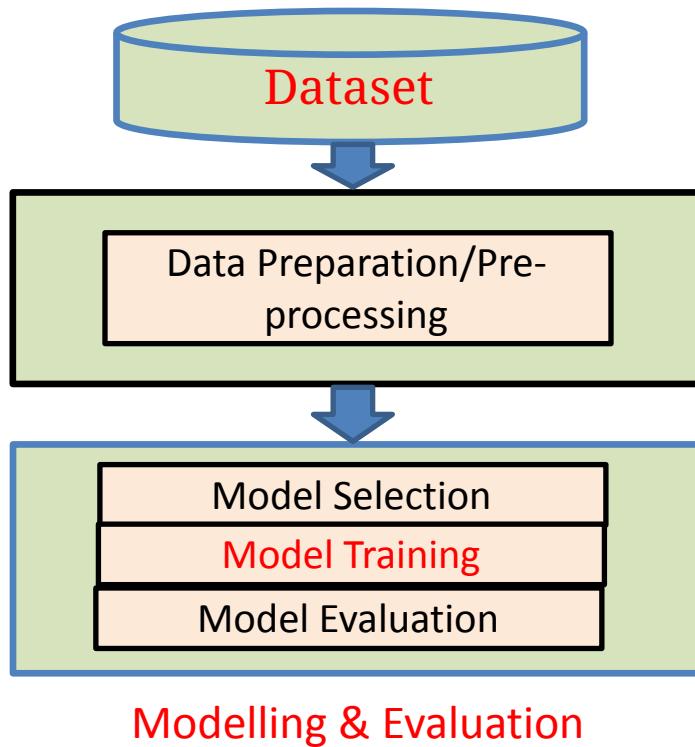


Modelling & Evaluation

Modelling & Evaluation

Split	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

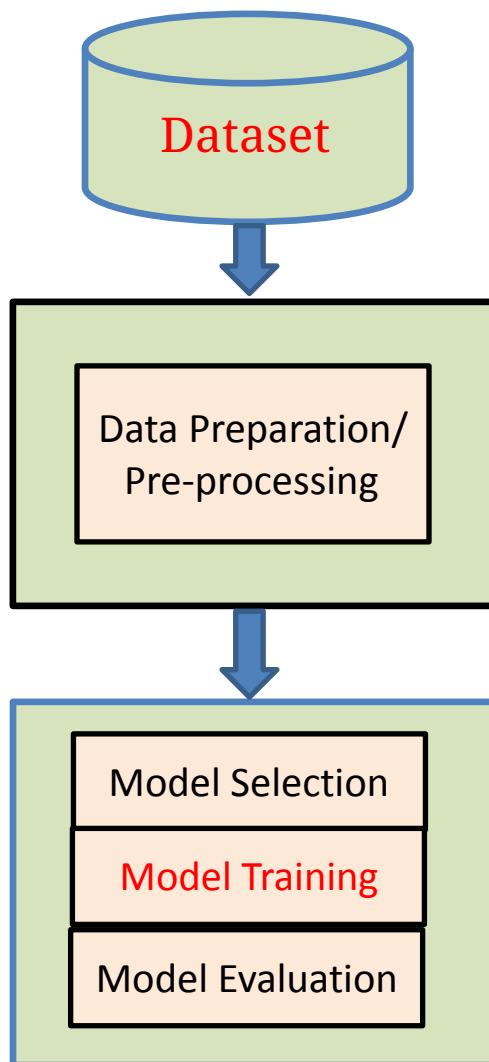
Training data
Test data



iii. K-fold Cross-validation method

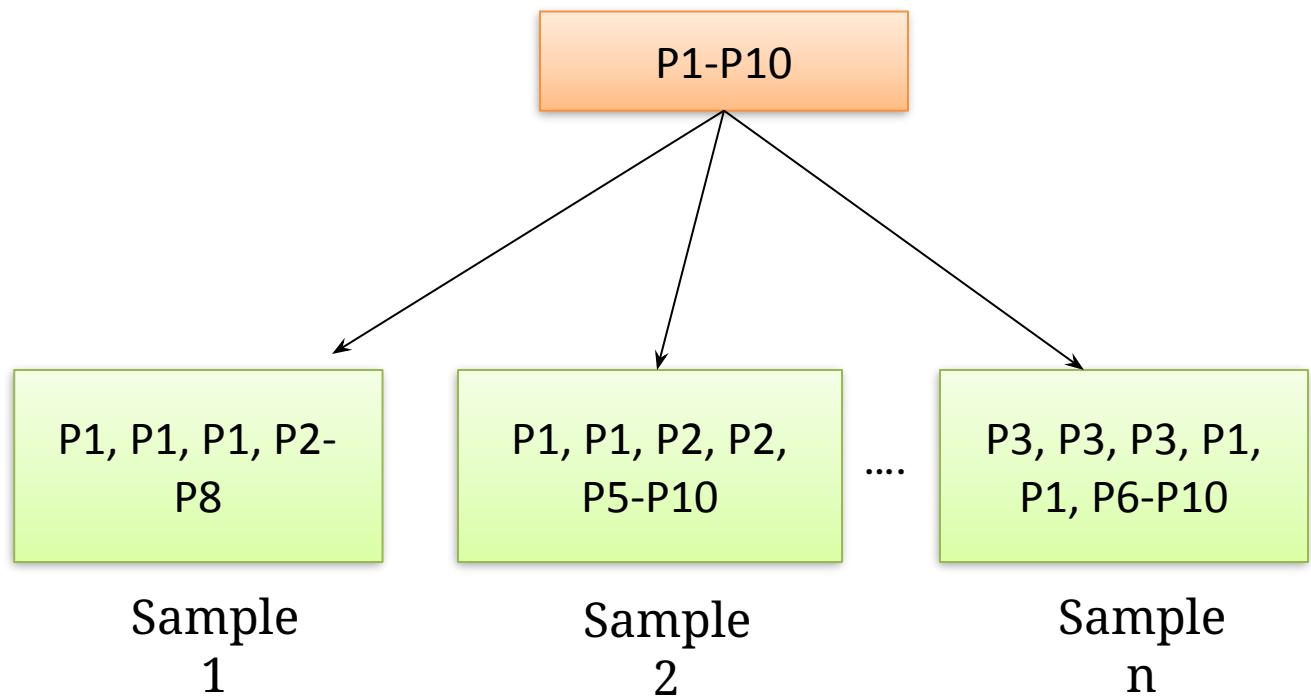
Disadvantages:

1. Time complexity is more.

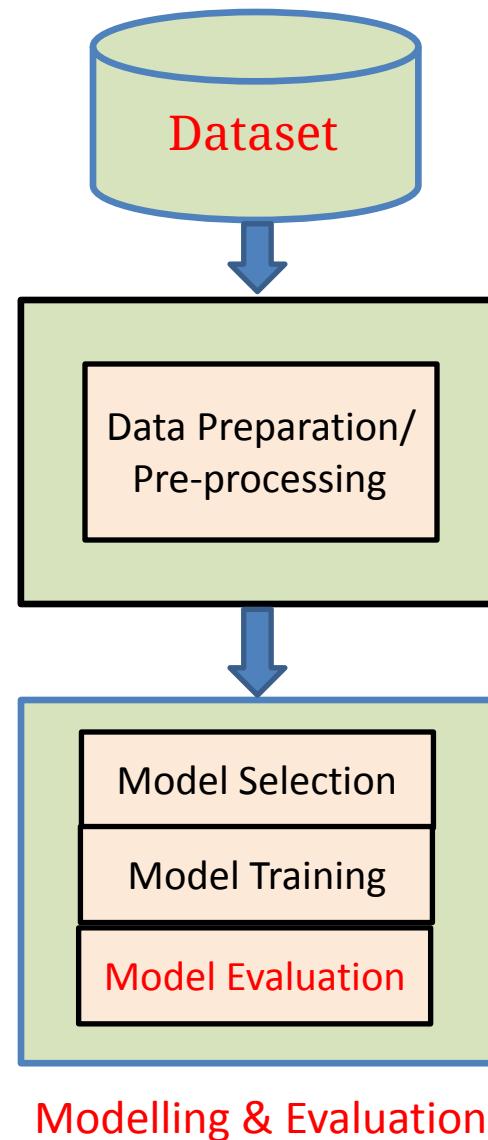


iii. Bootstrap sampling or bootstrapping:

- It uses Simple Random Sampling with Replacement
- It is used for small datasets
- Possible number of training/test data samples is unlimited



Data Science Model



Performance measures for Classification

Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP (3)	FN (1)
	Negative	FP (2)	TN(2)

TP = True Positive; FN= False Negative

FP= False Positive; TN= True Negative

Accuracy	
----------	--

Recall (high)	
Precision (low)	

F-measure	$(2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) = 2 * 0.6 * 0.75 / (0.6 + 0.75) = 0.9 / 1.35 = 67\%$
-----------	---

Model Evaluation

Performance measures for Classification

Matthews Correlation Coefficient (MCC)

Receiver Operating Characteristic (ROC) curves

Statistical Hypothesis Test:

- T test
- Z test
- ANOVA Test
- Chi-Square Test

ROC CURVE

- ROC curve helps in visualizing the performance of a classification model.
- It shows the efficiency of a model in the detection of true positives while avoiding the occurrences of false positives.

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

- In the ROC curve, the FP rate is plotted in the horizontal axis against True Positive Rate (TPR) in the vertical axis at different classification thresholds.

ROC CURVE

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

Y-A	Y-Pre	Y-0	Y-0.2	Y-0.4	Y-0.6	Y-0.8	Y-1
1	0.8	1	1	1	1	1	0
0	0.6	1	1	1	1	0	0
1	0.4	1	1	1	0	0	0
0	0.2	1	1	0	0	0	0

For Logistic Regression problem

- Prediction at threshold 0
- Prediction at threshold 0.2
- Prediction at threshold 0.4
- Prediction at threshold 0.6

At Y-0:

$$TPR = 2/(2+0) = 1; FPR = 2/(2+0) = 1$$

At Y-0.2

$$TPR = 2/(2+0) = 1; FPR = 2/(2+0) = 1$$

At Y-0.4

$$TPR = 2/(2+0) = 1; FPR = 1/(1+1) =$$

0.5

At Y-0.6

$$TPR = 1/(1+1) = 0.5; FPR = 1/(1+1) =$$

0.5

At Y-0.8:

$$TPR = 1/(1+1) = 0.5 \\ FPR = 0/(0+2) = 0$$

At Y-1

$$TPR = 0/(0+2) = 0 \\ FPR = 0/(0+2) = 0$$

ROC CURVE

At Y-0:

$$TPR = 2/(2+0) = 1; \quad FPR = 2/(2+0) = 1$$

At Y-0.2

$$TPR = 2/(2+0) = 1; \quad FPR = 2/(2+0) = 1$$

At Y-0.4

$$TPR = 2/(2+0) = 1; \quad FPR = 1/(1+1) = 0.5$$

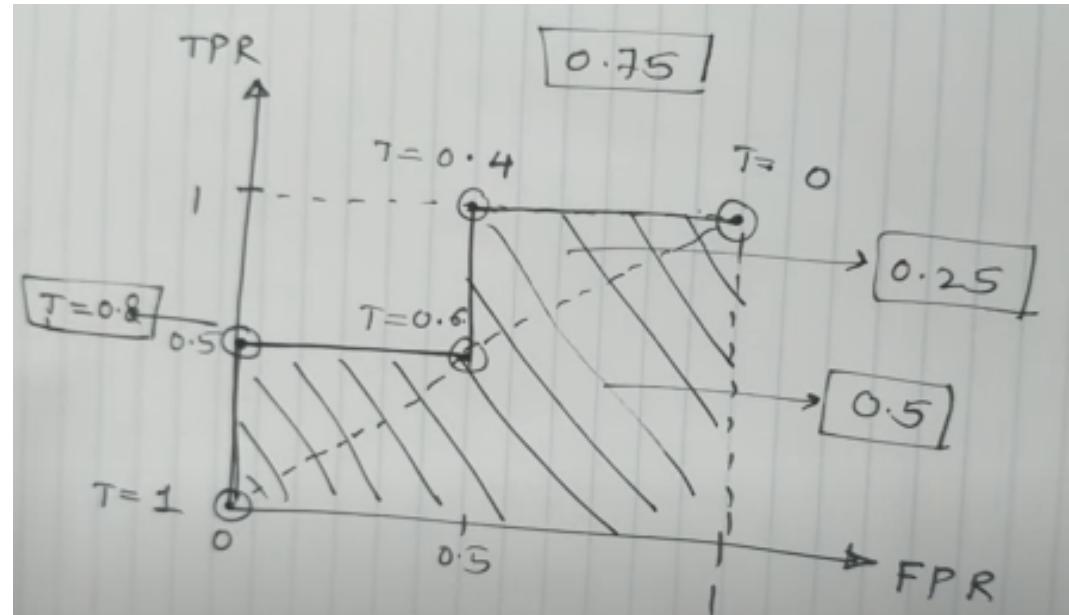
At Y-0.6

$$TPR = 1/(1+1) = 0.5; \quad FPR = 1/(1+1) = 0.5$$

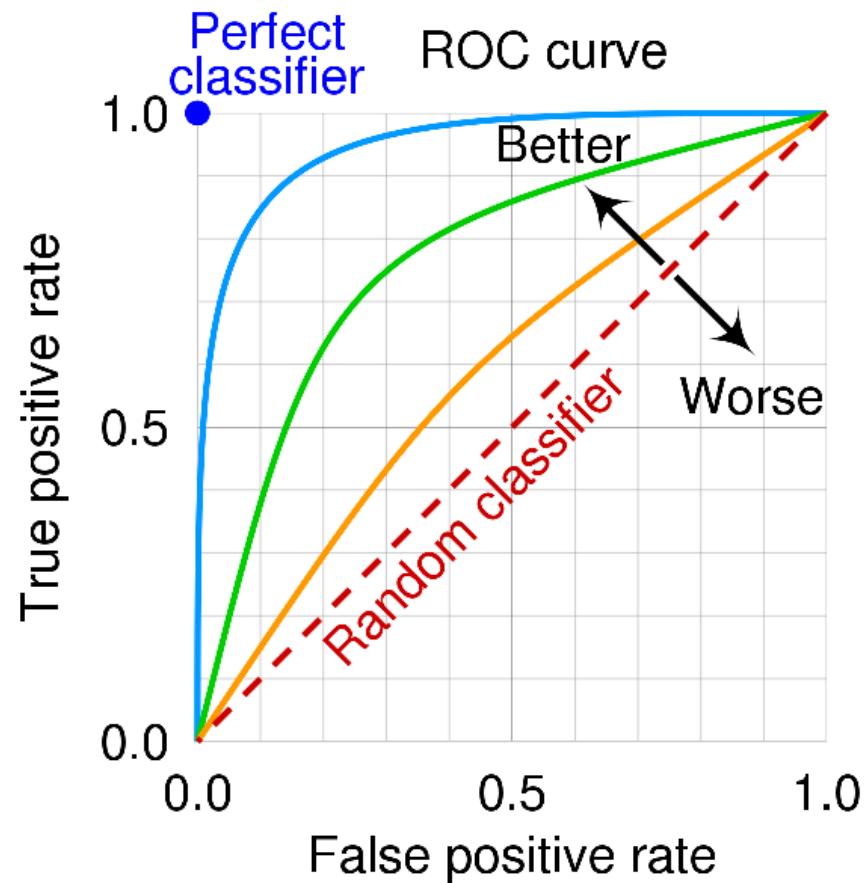
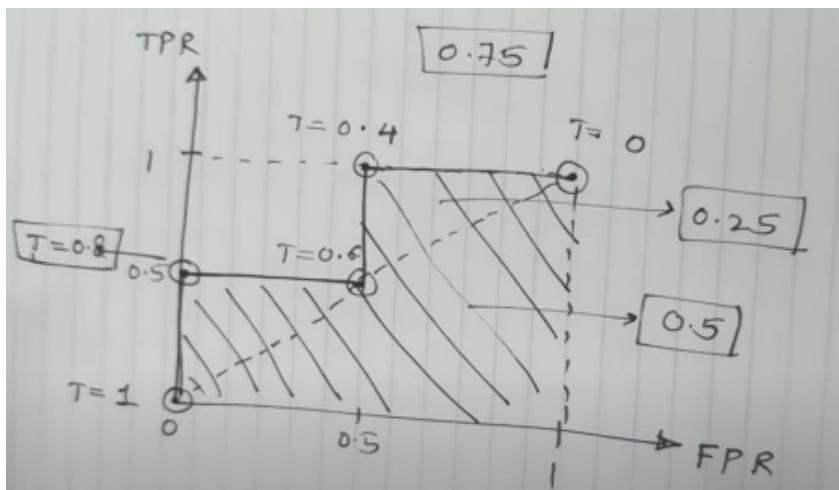
At Y-0.8:

$$TPR = 1/(1+1) = 0.5; \quad FPR = 0/(0+2) = 0$$

At Y-1: $TPR = 0/(0+2) = 0; \quad FPR = 0/(0+2) = 0$



ROC CURVE



Model Evaluation

Performance measures for Regression: R-squared

R-squared is a good measure to evaluate the model fitness.

The R-squared value lies between 0 to 1 (0% to 100%).

Large value represents a better fit.

$$R^2 = \frac{SST - SSE}{SST}$$

Sum of Squared Total (SST) = squared differences of each observation from the overall mean = $\sum_{i=1}^n (y_i - \bar{y})^2$ where \bar{y} is the mean.

Sum of Squared Errors (SSE) of prediction= sum of the squared residuals= $\sum_{i=1}^n (Y_i - y^*)^2$ where y^* is the predicted value of y_i and Y_i is the actual values of y_i

Evaluation of Regression

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

$$\begin{aligned}
 M_{Ext} &= 19.13 + 1.89 \times M_{Int} \\
 &= 19.13 + 1.89 \times 15 \\
 &= 19.13 + 28.35 \\
 &= 47.48
 \end{aligned}$$

$$SST = \sum_{i=1}^n (y_i - y')^2$$

$$SSE = \sum_{i=1}^n (y_i - y^*)^2$$

$$R^2 = \frac{SST - SSE}{SST}$$

$$\begin{aligned}
 &= (1148.4 - 328.51) / 1148.4 \\
 &= 819.89 / 1148.4 \\
 &= 0.71
 \end{aligned}$$

71%

		Square d Diff			
49	-7.8	60.84	47.48	1.52	2.31
63	6.2	38.44	62.6	0.4	0.16
58	1.2	1.44	53.15	4.2	17.64
60	3.2	10.24	62.6	-2.6	6.76
58	1.2	1.44	64.49	-6.49	42.12
61	4.2	17.64	60.71	0.29	0.08
60	3.2	10.24	60.71	-0.71	0.50
63	6.2	38.44	55.04	7.96	63.36
60	3.2	10.24	55.04	4.96	24.60
52	-4.8	23.04	49.37	2.63	6.92
62	5.2	27.04	64.49	-2.49	6.2
30	-26.8	718.24	39.92	-9.92	98.41
59	2.2	4.84	64.49	-5.49	30.14
49	-7.8	60.84	49.37	-0.37	0.14
68	11.2	125.44	62.6	5.4	29.16
56.8	SST = 1148.4			SSE = 328.51	

ii. Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

$$s_1 = \frac{l_1 + l_2 + l_3}{3} = x_1$$

Suppose the model has an RMSE value of Rs 500. Since the typical range of salary is between Rs 70,000 and Rs 300,000, this RMSE value is extremely low.

Suppose the model has an RMSE value of Rs 500. If the typical range of monthly house rent is Rs 1,500 – Rs 4,000, this RMSE value is quite high.

$$\text{Normalized RMSE} = \text{RMSE} / (\text{max value} - \text{min value}) = 4.679/(68-30) = 0.123$$

This produces a value between 0 and 1.

49	47.48	1.52	2.31
63	62.6	0.4	0.16
58	53.15	4.2	17.64
60	62.6	-2.6	6.76
58	64.49	-6.49	42.12
61	60.71	0.29	0.08
60	60.71	-0.71	0.50
63	55.04	7.96	63.36
60	55.04	4.96	24.60
52	49.37	2.63	6.92
62	64.49	-2.49	6.2
30	39.92	-9.92	98.41
59	64.49	-5.49	30.14
49	49.37	-0.37	0.14
68	62.6	5.4	29.16
56.8			SSE = 328.51
			MSE = 328.51/15 = 21.90

Model Evaluation

Performance measures for Regression:

iii. Mean Absolute Error (MAE)

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Normalized MAE = MAE / (max value – min value)= $3.69/(9.92-0.29)= 0.38$

Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the **RMSE should be more useful when large errors are particularly undesirable**. Both ranges from 0 to infinity.

49	47.48	1.52
63	62.6	0.4
58	53.15	4.2
60	62.6	2.6
58	64.49	6.49
61	60.71	0.29
60	60.71	0.71
63	55.04	7.96
60	55.04	4.96
52	49.37	2.63
62	64.49	2.49
30	39.92	9.92
59	64.49	5.49
49	49.37	0.37
68	62.6	5.4
56.8		AE = 55.43 MAE= 55.43/15=3.69

Model Evaluation

Performance measures for

Clustering

It is generally not known how many clusters can be formulated from a particular dataset.

Even if the number of clusters is given, the same number of clusters can be formed with different groups of data instances.

Internal Evaluation

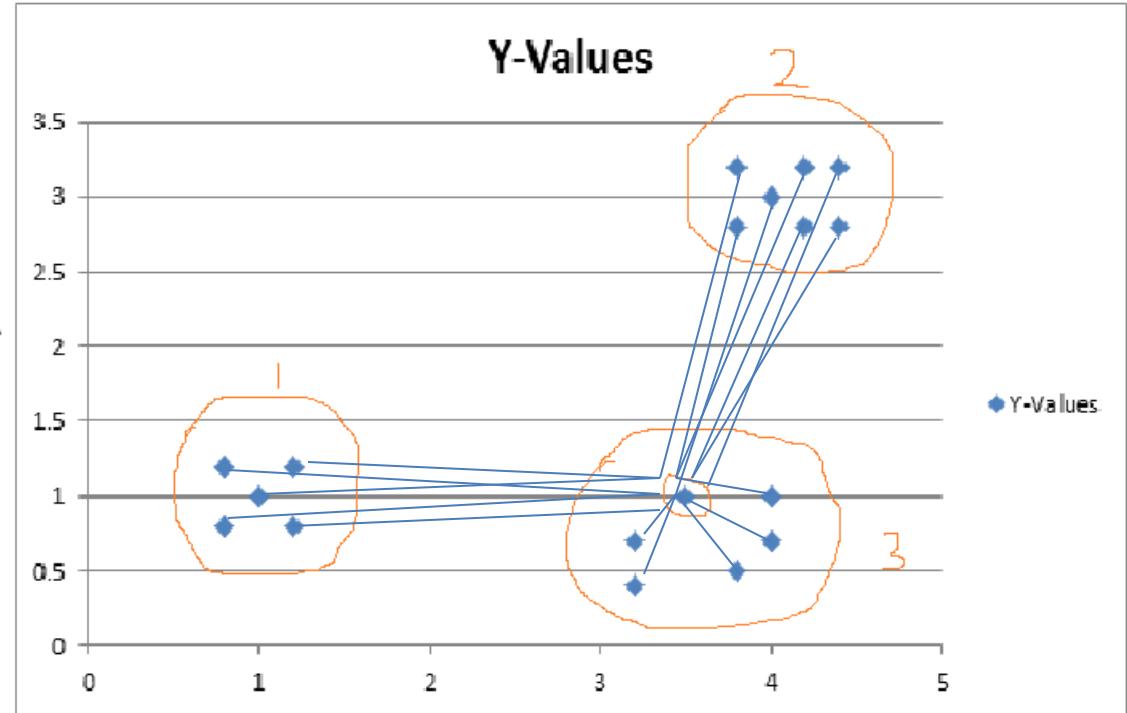
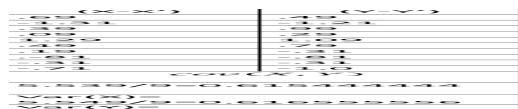
$$\text{Silhouette width} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

It ranges from -1 to +1

a(i) is the average distance between the ith data instance and all other data instances belonging to the same cluster

b(i) is the lowest average distance between the ith data instance and data instances of all other clusters.

a(i) is the average of the distances $a_{i1}, a_{i2} \dots \dots a_{in3}$ of the different data elements from the ith data elements in cluster 3, n3= data elements of cluster 3.



Average distance from ith elements of cluster 3 to cluster 1

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

Similarly b_{32} can be calculated

$$b(i) = \min [b_{32}(\text{average}), b_{31}(\text{average})]$$

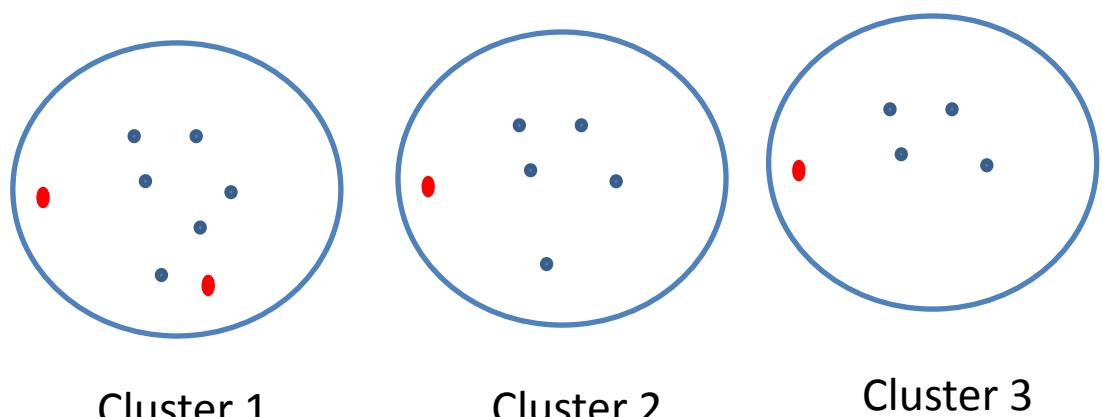
Model Evaluation

External Evaluation:

Purity: This is only applicable for class labels data though class labels are not used for clustering

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

$\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ = set of clusters $\mathbb{C} = \{c_1, c_2, \dots, c_J\}$ = set of classes N is total data instances



Supervised ML Algorithms

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- K- Nearest Neighbour (K-NN)
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

REGRESSION

Regression is a supervised learning which predicts a continuous value

- Predicting the price of a car
- Predicting the amount of rainfall
- Predicting the cost of a land

The most common regression algorithms are

- Simple linear regression
- Multiple linear regression
- Ridge Regression
- LASSO Regression
- Elastic Net Regression
- Polynomial regression
- Multivariate adaptive regression splines
- Logistic regression

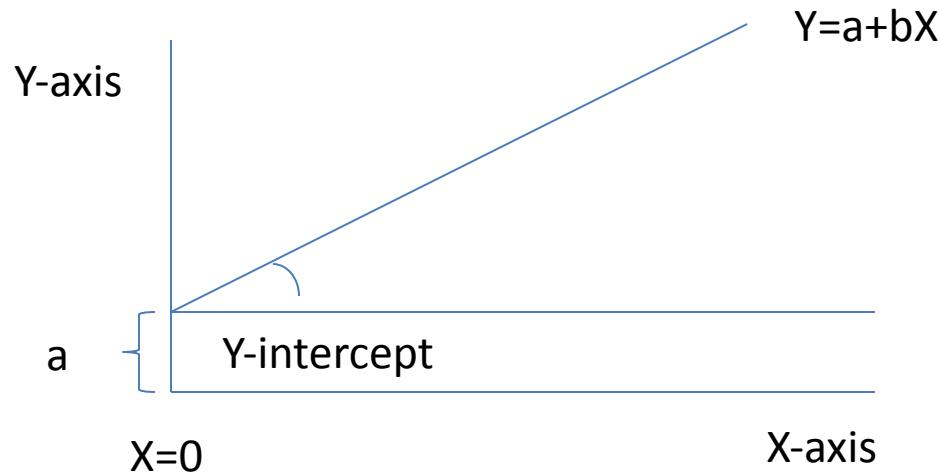
Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Simple Linear Regression

Simple Linear Regression is the simplest regression model which involves only one predictor.

This model assumes a linear relationship between the dependent variable and the predictor variable

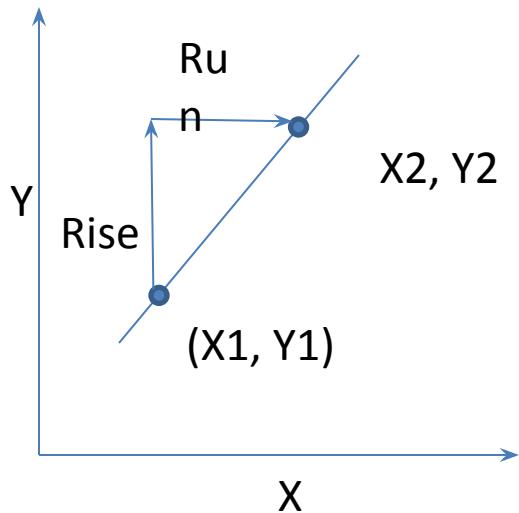
$Y = a + bX$ (price of a property as the dependent variable and the area of the property as the predictor variable)



Slope of the simple Linear Regression model

Slope of a straight line represents how much the line in a graph changes in the vertical direction over a change in the horizontal direction.

$$\text{Slope} = \text{Change in Y}/\text{Change in X}$$



$$\text{Slope} = \text{Rise}/\text{Run} = \frac{Y_2 - Y_1}{X_2 - X_1}$$

Let lower point = (-3, -2); higher point = (2, 2)

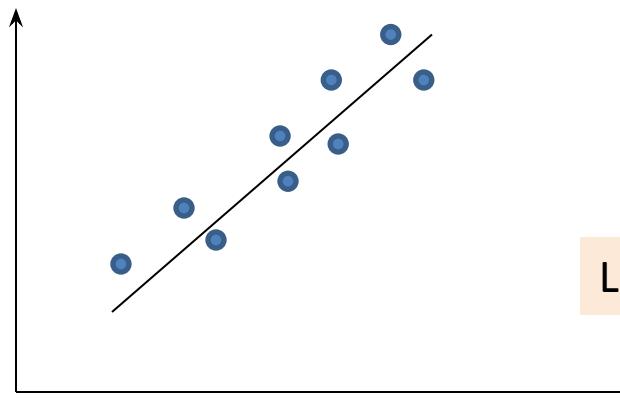
$$\text{Rise} = (2 - (-2)) = 2 + 2 = 4$$

$$\text{Run} = (2 - (-3)) = 2 + 3 = 5$$

$$\text{Slope} = \text{Rise}/\text{Run} = 4/5 = 0.8$$

Slopes in a Linear Regression

- There are two types of slopes: positive slope and negative slope
- Different types of regression lines based on the type of slope are
 - Linear positive slope
 - Curve linear positive slope
 - Linear negative slope
 - Curve linear negative slope



Linear positive slope

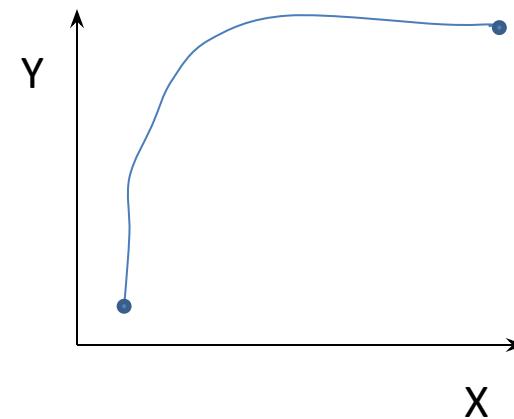
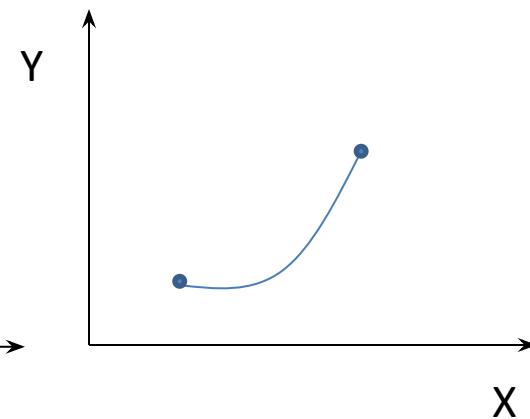
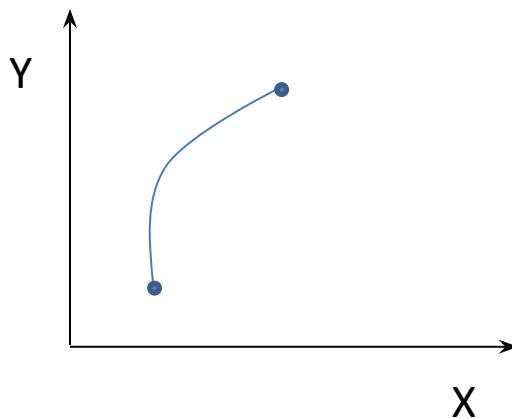
$$\text{Slope} = \text{Rise/Run} = \frac{\Delta(Y)}{\Delta(X)}$$

Scenario 1 for positive slope: $\Delta(Y)$ is positive and $\Delta(X)$ is positive

Scenario 2 for positive slope: $\Delta(Y)$ is negative and $\Delta(X)$ is negative

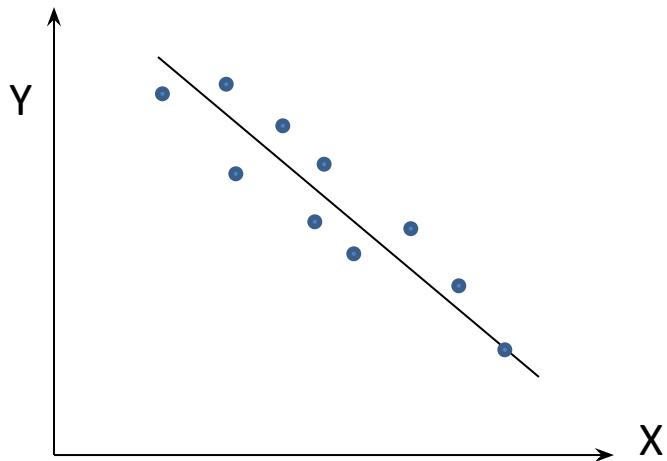
Slopes in a Linear Regression

Curve Linear Positive Slope



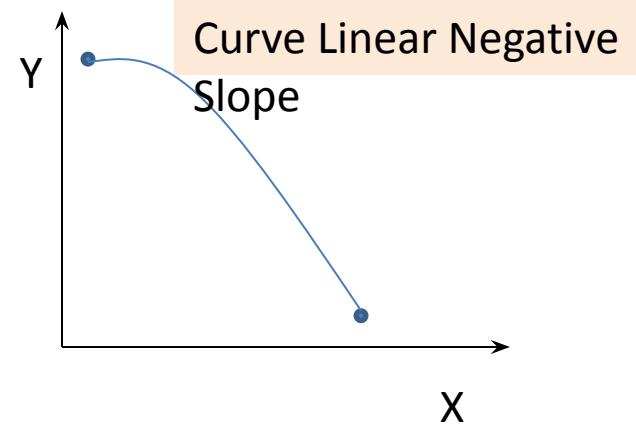
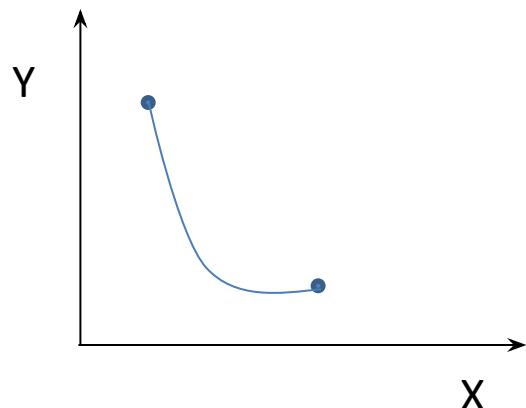
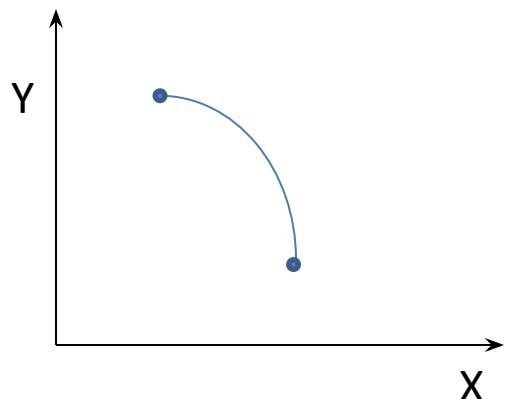
Slopes in a Linear Regression

Linear Negative Slope

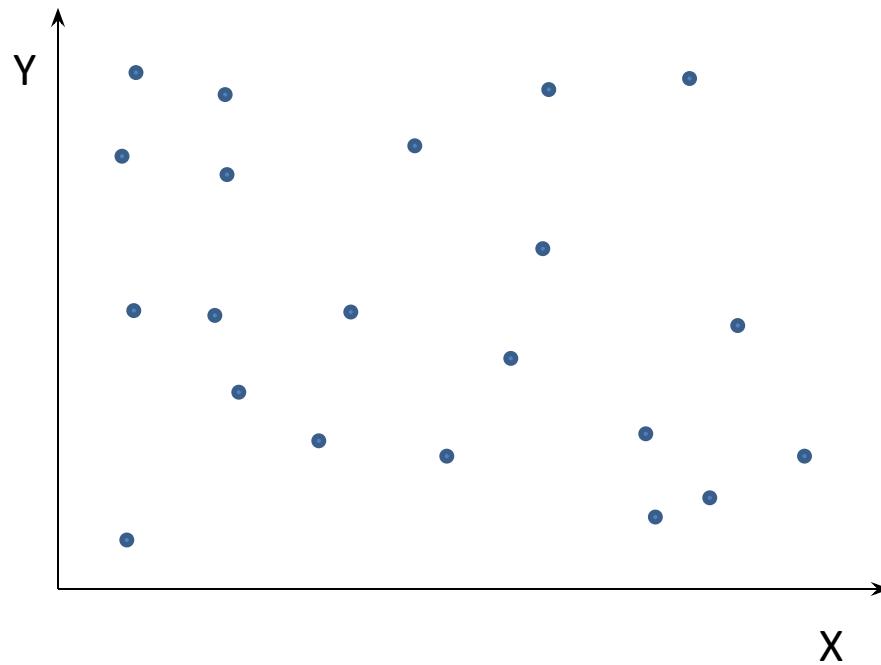


Scenario 1 for negative slope: Delta(Y) is positive and Delta(X) is negative

Scenario 2 for negative slope: Delta(Y) is negative and Delta(X) is positive



No Relationship Graph



Error in Simple Regression

X and Y values are provided to the machine to find the values of a and b by relating the values of X and Y.

However identifying the exact match of values for a and b is not always possible. There will be some error (ε). This error is called marginal or residual error.

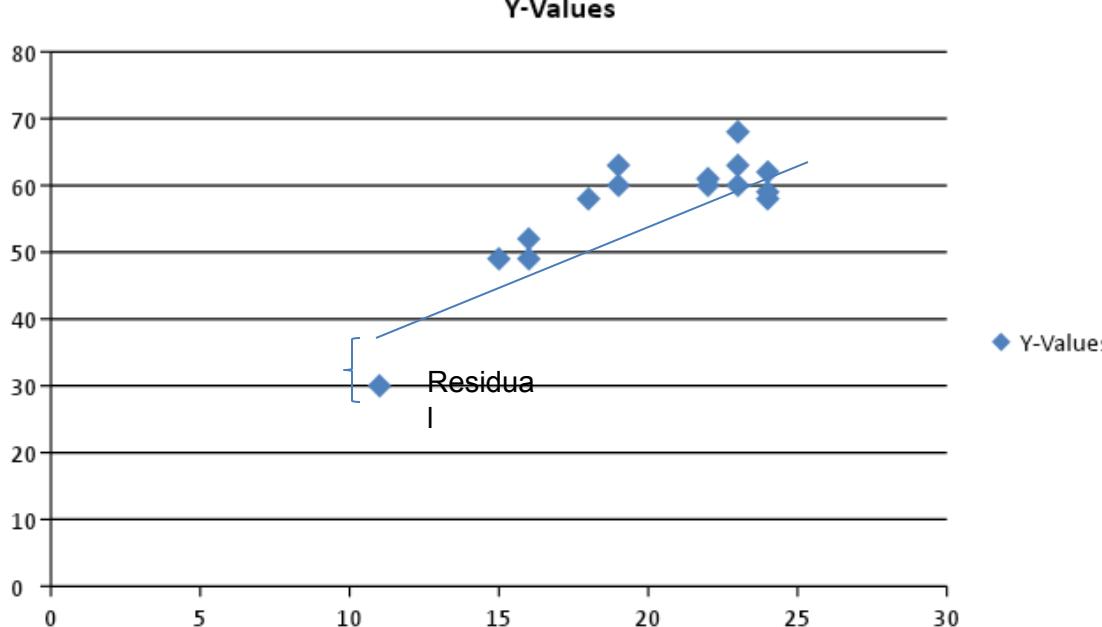
$$Y = (a + bX) + \varepsilon$$

Linear Regression Technique with example

Finding the relationship between internal examination and external examination from the given data samples

Internal Exam	15	23	18	23	24	22	22	19	19	16	2 4	11	24	16	23
External Exam	49	63	58	60	58	61	60	63	60	52	6 2	30	59	49	68

Linear Regression Technique with example



Residual is the distance between the predicted point and actual point.

$$Y = (a + bX) + \epsilon$$

If values of a and b are known, it is easy to predict the value of Y for any given X .
How to calculate the values of a and b for a given set of X and Y values with minimum error.

Linear Regression Technique with example

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

OLS Algorithm

Step1: Calculate the mean of X and Y.

Step2: Calculate the errors for each values of X and Y.

Step3: Get the product for each corresponding values.

Step4: Get the summation of the products.

Step5: Square the difference of X and mean(X).

.

Step6: Get the sum of the squared differences.

X	Y	X-mean(X)	Y-mean(Y)		
15	49	-4.93	-7.8	38.454	24.3049
23	63	3.07	6.2	19.034	9.4249
18	58	-1.93	1.2	-2.316	3.7249
23	60	3.07	3.2	9.824	9.4249
24	58	4.07	1.2	4.884	16.5649
22	61	2.07	4.2	8.694	4.2849
22	60	2.07	3.2	6.624	4.2849
19	63	-0.93	6.2	-5.766	0.8649
19	60	-0.93	3.2	-2.976	0.8649
16	52	-3.93	-4.8	18.864	15.4449
24	62	4.07	5.2	21.164	16.5649
11	30	-8.93	-26.8	239.324	79.7449
24	59	4.07	2.2	8.954	16.5649
16	49	-3.93	-7.8	30.654	15.449
23	68	3.07	11.2	34.384	9.4249
19.9	56.			429.8	226.9335
3	8				

Linear Regression Technique with example

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

OLS Algorithm

Step7: Divide output of step4 by output of step 6 to calculate b.

$$b = 429.28 / 226.93 = 1.89$$

Step8: Calculate 'a' using the value of b.

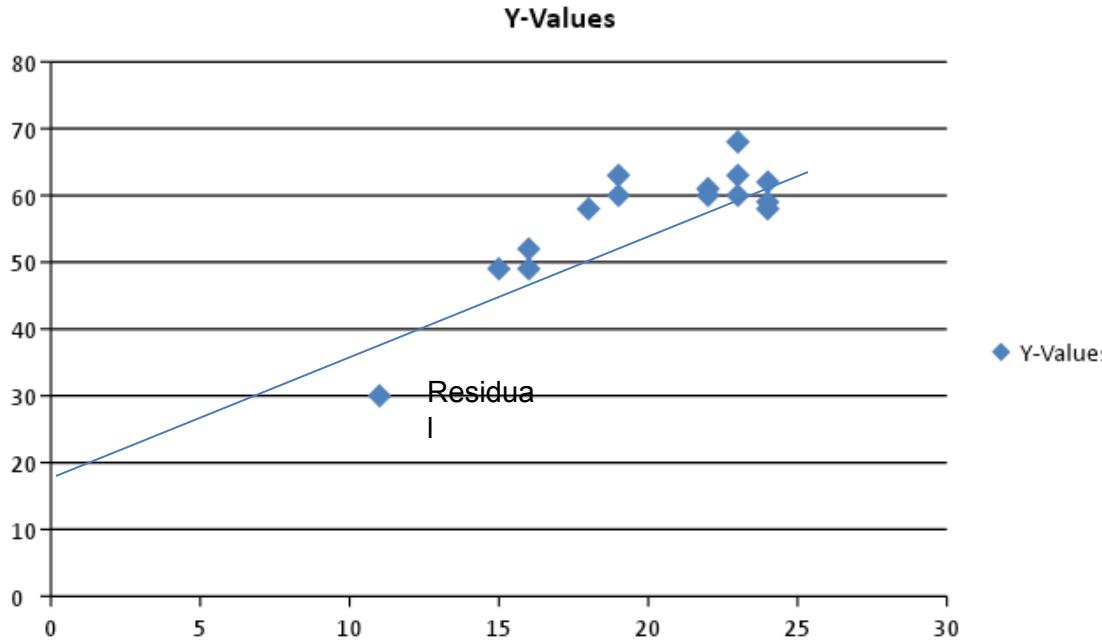
$$a = Y' - bX'$$

$$a = 56.8 - 1.89 * 19.93$$

$$a = 19.13$$

X	Y	X-mean(X)	Y-mean(Y)		
15	49	4.03	-7.8	38.454	24.3049
23	63	3.07	6.2	19.034	9.4249
18	58	-1.93	1.2	-2.316	3.7249
23	60	3.07	3.2	9.824	9.4249
24	58	4.07	1.2	4.884	16.5649
22	61	2.07	4.2	8.694	4.2849
22	60	2.07	3.2	6.624	4.2849
19	63	-0.93	6.2	-5.766	0.8649
19	60	-0.93	3.2	-2.976	0.8649
16	52	-3.93	-4.8	18.864	15.4449
24	62	4.07	5.2	21.164	16.5649
11	30	-8.93	-26.8	239.324	79.7449
24	59	4.07	2.2	8.954	16.5649
16	49	-3.93	-7.8	30.654	15.449
23	68	3.07	11.2	34.384	9.4249
19.9 3	56. 8			429.28	226.9335

Intercept and Slope



Recall (high)	$\frac{TP}{TP + FN} = \frac{3}{3+1} = \frac{3}{4} = 0.75$
Precision (low)	$\frac{TP}{TP + FP} = \frac{3}{3+2} = \frac{3}{5} = 0.6$

- Intercept 19.13 indicates that 19.13 is the portion of the external examination marks not explained by the internal examination marks.
- Slope = 1.89 tells us that the average value of the external examination marks increases by 1.89 for each additional 1 mark in the internal examination.

Evaluation of Regression

Performance measures for Regression: R-squared

R-squared is a good measure to evaluate the model fitness.

The R-squared value lies between 0 to 1 (0% to 100%).

Large value represents a better fit.

$$R^2 = \frac{SST - SSE}{SST}$$

Sum of Squared Total (SST) = squared differences of each observation from the overall mean = $\sum_{i=1}^n (y_i - \bar{y})^2$ where \bar{y} is the mean.

Sum of Squared Errors (SSE) of prediction= sum of the squared residuals= $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ where \hat{y}_i is the predicted value of y_i .

Linear Regression Technique with example

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

$$\begin{aligned}
 M_{Ext} &= 19.13 + 1.89 \times M_{Int} \\
 &= 19.13 + 1.89 \times 15 \\
 &= 19.13 + 28.35 \\
 &= 47.48
 \end{aligned}$$

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$R^2 = \frac{SST - SSE}{SST}$$

$$\begin{aligned}
 &= (1148.4 - 328.51) / 1148.4 \\
 &= 819.89 / 1148.4 \\
 &= 0.71
 \end{aligned}$$

71%

		Square d Diff			
49	-7.8	60.84	47.48	1.52	2.31
63	6.2	38.44	62.6	0.4	0.16
58	1.2	1.44	53.15	4.2	17.64
60	3.2	10.24	62.6	-2.6	6.76
58	1.2	1.44	64.49	-6.49	42.12
61	4.2	17.64	60.71	0.29	0.08
60	3.2	10.24	60.71	-0.71	0.50
63	6.2	38.44	55.04	7.96	63.36
60	3.2	10.24	55.04	4.96	24.60
52	-4.8	23.04	49.37	2.63	6.92
62	5.2	27.04	64.49	-2.49	6.2
30	-26.8	718.24	39.92	-9.92	98.41
59	2.2	4.84	64.49	-5.49	30.14
49	-7.8	60.84	49.37	-0.37	0.14
68	11.2	125.44	62.6	5.4	29.16
56.8	SST= 1148.4			SSE = 328.51	

Multiple Linear Regression

Two or more independent variables are involved in this model.

$$\text{Price} = f(\text{Area}, \text{location}, \text{floor}, \text{ageing}, \text{amenities})$$

To determine price of the property; area, location, floor, number of years since purchase and amenities are considered.

The following equation describes the relation with 2 independent variables

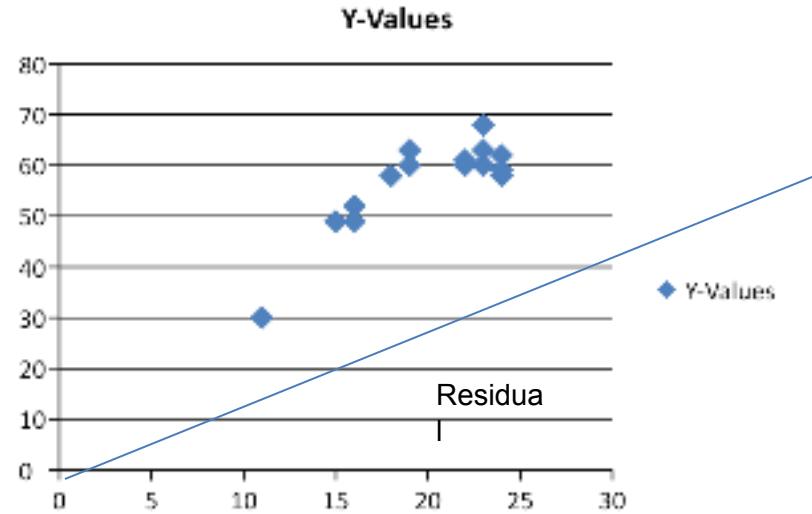
$$Y = a + b_1 X_1 + b_2 X_2$$

Parameters b_1 and b_2 are referred as partial regression coefficient.

Assumption in Linear Regression Analysis

- The dependent variable can be calculated as a linear function of a set of independent variables plus error term.
- Number of observation is greater than the number of parameters
- Regression line can be valid only over a limited range of data. If the line is extended outside the range of extrapolation, it may only lead to wrong predictions.
- Variance is the same for all values of X.
- The error term is normally distributed. This also means that the mean of the error has an expected value of 0.

$$Y = a + b_1X_1 + b_2X_2$$



Primary problems in Multiple Regression

- **Multicollinearity:** A multiple regression equation can make good predictions when there is multicollinearity, However,
 - it is difficult for us to determine how the dependent variable will change if each independent variable is changed one at a time.
 - When multicollinearity is present, it increases the standard errors of the coefficients.
 - By overinflating the standard errors, multicollinearity tries to make some variables statistically insignificant when they actually should be significant.
- **Heteroskedasticity:** Heteroskedasticity refers to the changing variance of the error term . If the variance of the error term is not constant across datasets, there will be erroneous predictions.

Improving accuracy of Linear Regression Model

High bias= low accuracy

High variance= low prediction

Low bias= high accuracy

Low variance= high prediction

Therefore balancing out bias and accuracy is essential in a regression model.

In the regression model, it is assumed that the number of observation is greater than the number of parameters to be estimated.

However if observation is not much larger than parameters, then there can be high variability in the least fit, resulting in overfitting and leading to poor prediction.

Accuracy of linear regression can be improved using the following three approaches:

1. Shrinkage Approach
2. Subset Selection
3. Dimensionality Reduction

Shrinkage Approach

By limiting (shrinking) the estimated coefficients, variance can be reduced at the cost of a negligible increase in bias. This leads substantial improvements in the accuracy of the model.

The two best-known techniques for shrinking the regression coefficients towards zero are

1. Ridge regression
2. LASSO (Least Absolute Shrinkage Selection Operator)

If $k > n$, then the least squares estimates do not ever have a unique solution, whereas ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance. Thus ridge regression works best in situation where the least squares estimates have high variance.

However ridge includes all k predictors in the final model. This may not be a problem for prediction accuracy but it can create a challenge in model interpretation in setting in which the number of variables k is quite large.

$$f(x) = c_0 + c_1 X^1 + c_2 X^2 + c_3 X^3$$

Keeps all the attributes.

Shrinkage Approach

LASSO overcomes this disadvantage by forcing some of the coefficients to zero value.
It's a simple and more interpretable model.

LASSO works better when small number of predictors have substantial coefficients,
and the remaining predictors have coefficients that are very small or equal to zero.

$$f(x) = c_0 + \textcolor{red}{c}_1 X^1 + c_2 X^2 + c_3 X^3$$

Next,

$$f(x) = c_0 + 0X^1 + c_2 X^2 + c_3 X^3$$

$$f(x) = c_0 + c_2 X^2 + c_3 X^3$$

Subset Selection

A subset of predictors that is assumed to be related to the response is selected to fit a model. There are different kinds of methods for subset selection, some of which are given below:

1. Exhaustive search
2. Branch and Bound Search,
3. Selection of Best Individual Features
4. Sequential Forward Selection
5. Sequential Backward Selection
6. Sequential Floating Forward Selection
7. Sequential Floating Backward Selection

Dimensionality Reduction

- Predictors are transformed and the model is set up using the transformed variables after dimensionality reduction.
- The number of variables is reduced using the dimensionality reduction method.
- Principal component analysis is one of the most important dimensionality reduction techniques.

Ridge Regression

$$P1 = (1, 2.3)$$

$$P2 = (3, 5.3)$$

$$Y = (a + bX)$$

$$b = (5.3 - 2.3) / (3 - 1) = 3/2 = 1.5$$

$$Y = (a + bX)$$

$$a = Y - bX$$

$$= 2.3 - 1.5 * 1 = 2.3 - 1.5 = 0.8$$

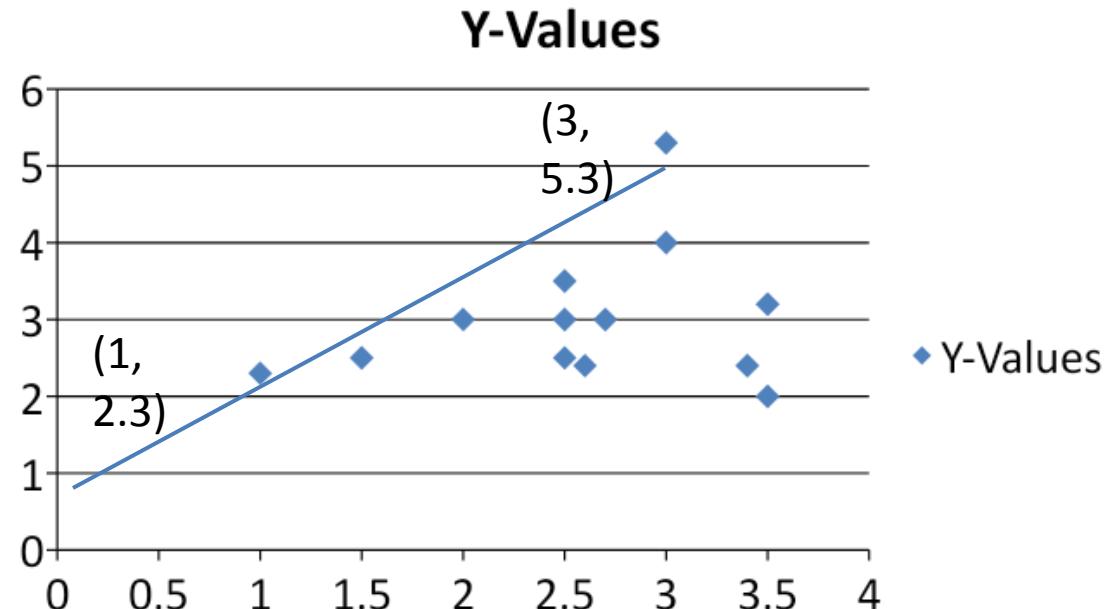
$$y = 0.8 + 1.5x$$

There is no bias (bias=0) as :

$$x=1: y = 0.8 + 1.5 * 1 = 0.8 + 1.5 = 2.3$$

$$x=3: y = 0.8 + 1.5 * 3 = 0.8 + 4.5 = 5.3$$

x	1	3
y	2.3	5.3



$$SSE = \sum_{i=1}^n (Y_i - y^*)^2 = (2.3 - 2.3)^2 + (2.3 - 2.3)^2 = 0 + 0 = 0$$

Means **overfitting** problem is there

How can we understand overfitting and underfitting from the line

Ridge Regression

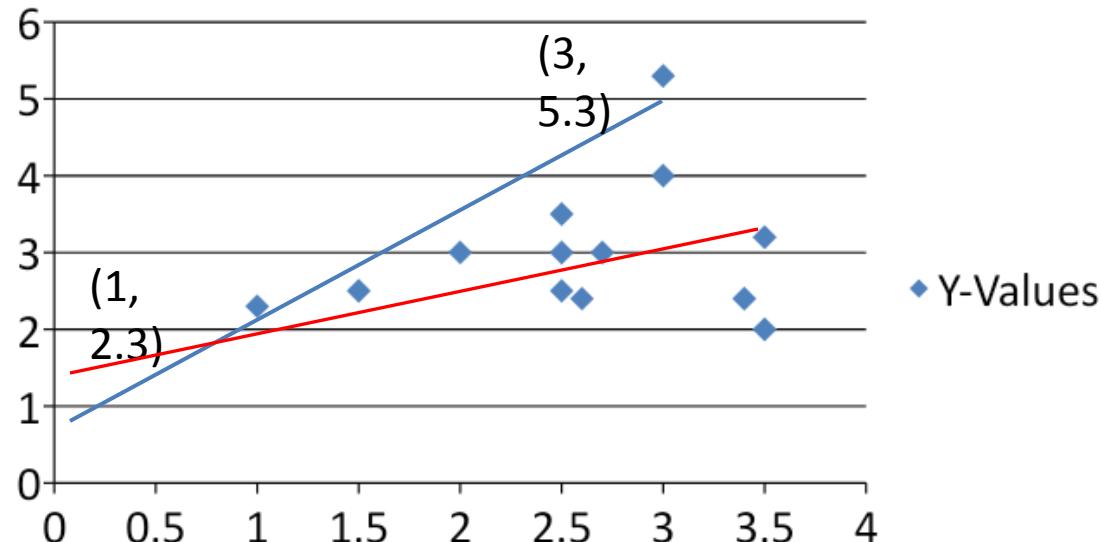
$y = 0.8 + 1.5x$ variance is high

$y = 1.5 + 0.9x$ (imaginary line to reduce variance)

Our normal objective is to reduce

$SSE = \sum_{i=1}^n (Y_i - y^*)^2$ (but would not be working as it increases bias for the imaginary line)

Y-Values



To decrease the variance of the imaginary line, the objective function is redefined.

Therefore, now objective is to reduce $SSE = \sum_{i=1}^n (Y_i - y^*)^2 + \lambda(b)^2 = \text{LOSS}$

Ridge Regression

$y = 0.8 + 1.5x$ variance is high

$y = 1.5 + 0.9x$ (imaginary line to reduce variance)

Therefore, now objective is

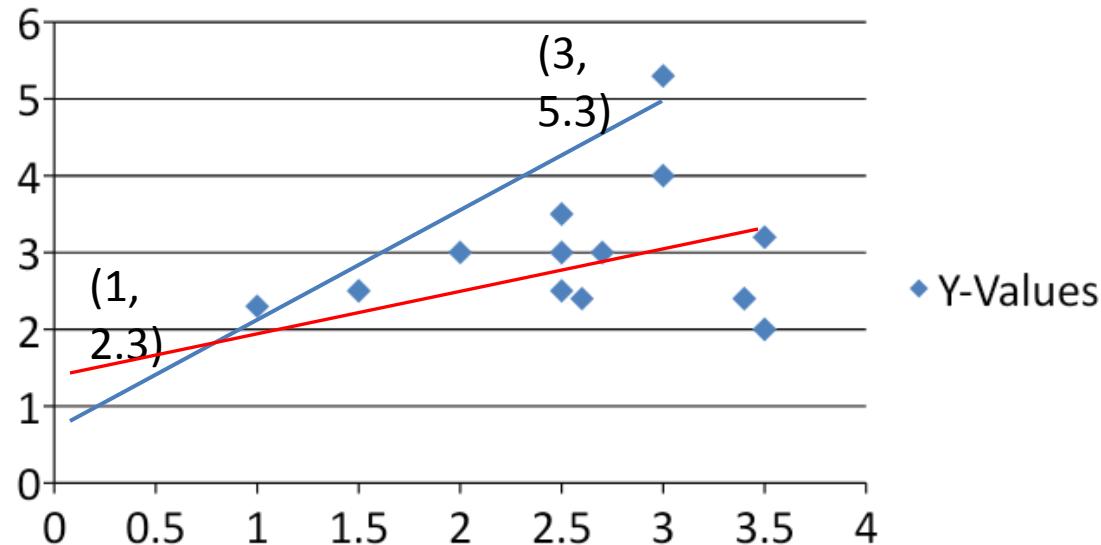
to reduce SSE =

$$\sum_{i=1}^n (Y_i - y^*)^2 + \lambda(\mathbf{b})^2 = \text{LOSS}$$

$$x=1: y = 0.8 + 1.5 * 1 = 0.8 + 1.5 = 2.3$$

$$x=3: y = 0.8 + 1.5 * 3 = 0.8 + 4.5 = 5.3$$

Y-Values



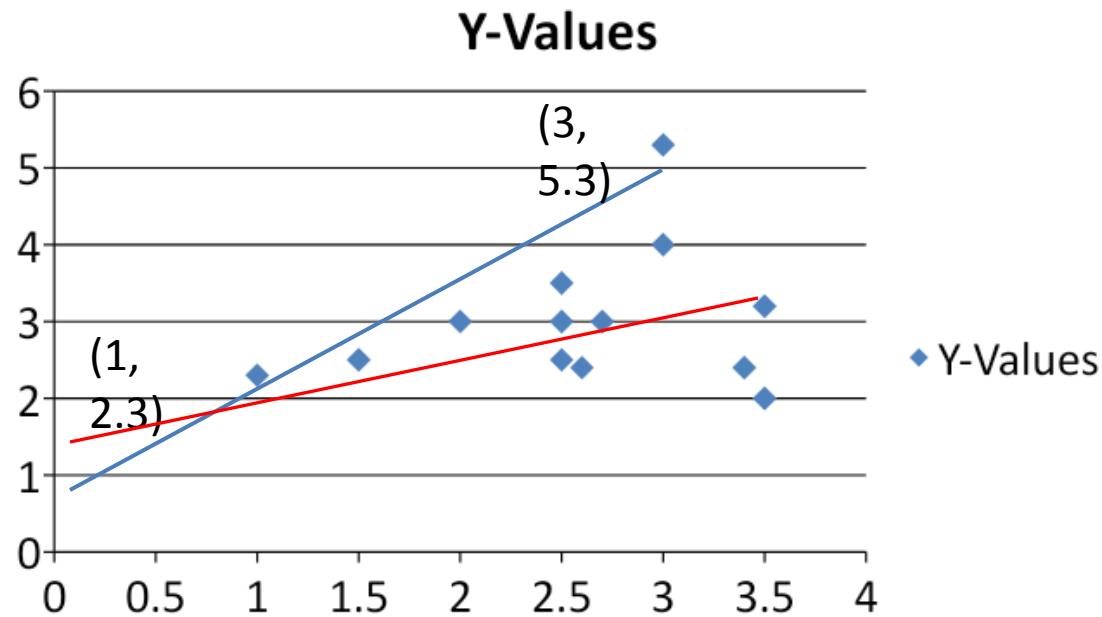
LOSS

LOSS

Ridge Regression

-
1. $\text{loss}(w) = \frac{1}{2} \|w\|_2^2$

$$\begin{aligned} &= \frac{1}{3} \left[\max \left(3^{\text{th}} \text{ dist } B, \text{dist}(AB) \right) + \max \left(3^{\text{th}} \text{ dist } C, \text{dist}(AC) \right) + \right. \\ &\quad \left. \max \left(3^{\text{th}} \text{ dist } D, \text{dist}(AD) \right) \right] \end{aligned}$$



Polynomial Regression Model

It is the extension of the simple linear model by adding extra predictors obtained by raising each of the original predictors to a power.

$$f(x) = c_0 + c_1X^1 + c_2X^2 + c_3X^3$$

c_0, c_1, c_2 and c_3 are the coefficients. It's a degree 3 polynomial.

Internal Exam	15	23	18	23	24	22	22	19	19	16	2 4	11	24	16	23
External Exam	49	63	58	60	58	61	60	63	60	52	6 2	30	59	49	68
	3375														
	225														
	15	23	18	23	24	22	22	19	19	16	24	11	24	16	23
f(x)	49	63	58	60	58	61	60	63	60	52	62	30	59	49	68

Polynomial Regression Model

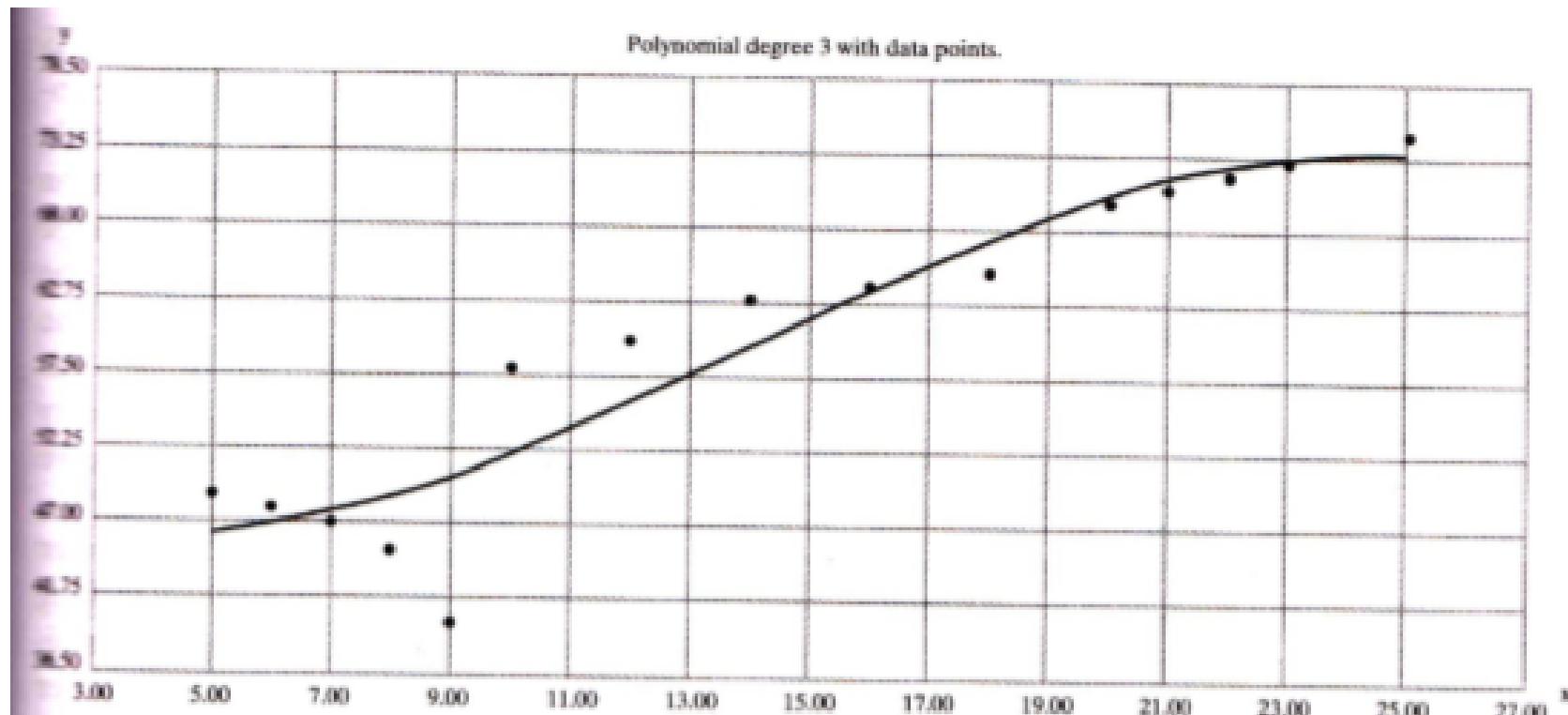


FIG. 8.16
Polynomial regression degree 3

Polynomial Regression Model

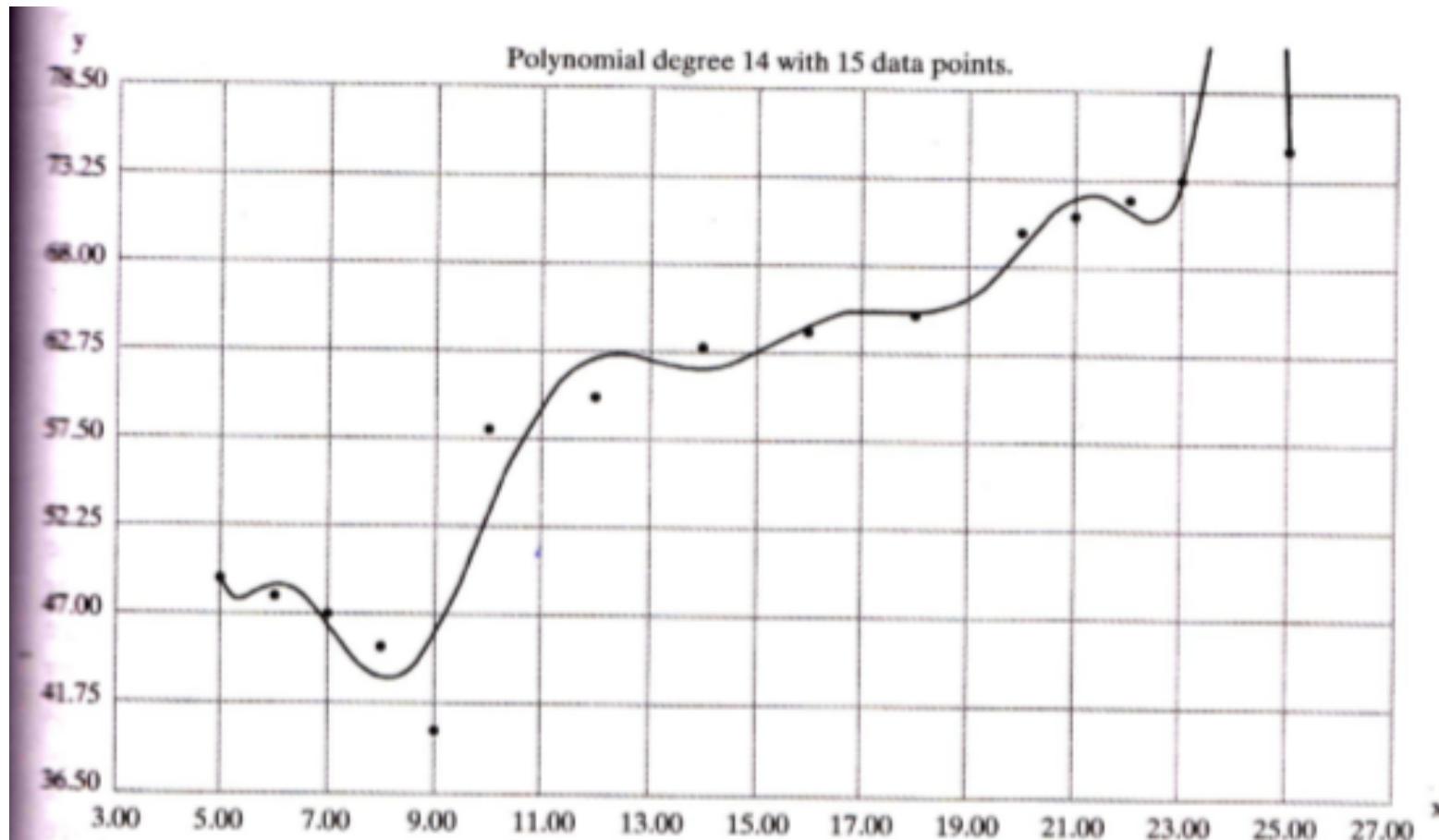


FIG. 8.17
Polynomial regression degree 14

Polynomial Regression Model

There are some relationships that a researcher will hypothesize is curvilinear. Clearly, such types of cases will include a polynomial term.

Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y .

An assumption in usual multiple linear regression analysis is that all the independent variables are independent. In polynomial regression model, this assumption is not satisfied.

Logistic Regression

- Logistic Regression is both classification and regression technique depending on the scenario used.
- It (logic regression) is a type of regression analysis used for predicting the outcome of a categorical dependent variable.
- Dependent variable (Y) is binary (0,1) and independent variables are continuous in nature.
- The goal of logistic regression is to predict the likelihood that Y is equal to 1 given certain values of X.
- So we predict probabilities rather than the scores of the dependent variable.

An example:

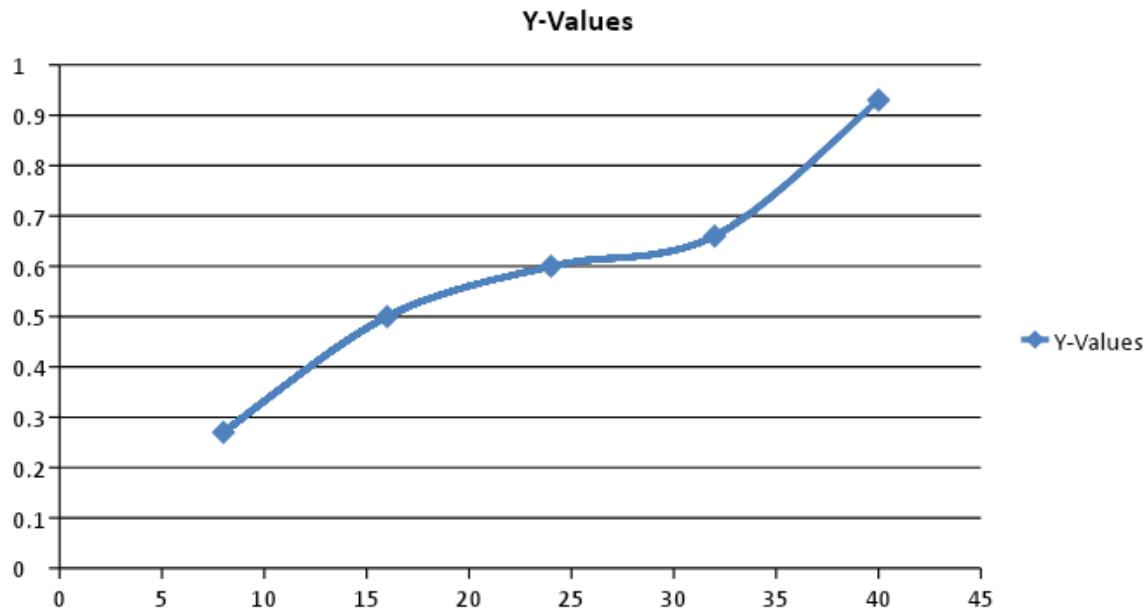
X	Y
0-8	0.27
9-16	0.5
17-24	0.6
25-32	0.66
33-40	0.93

X= experience in years

Y= Probability to be 1

Logistic Regression

X	Y
0-8	0.27
9-16	0.5
17-24	0.6
25-32	0.66
33-40	0.93



- A perfect relationship represents a perfectly curved S rather than a straight line.
- To model this relationship, we need some mathematics that accounts for the bend in the curve.

Logistic Regression

- Probability (P) can be computed from the regression equation.
- If we know the regression equation, we could calculate the expected probability that Y=1 for a given value of X.

$$P = \frac{\exp(a + bX)}{1 + \exp(a + bX)}$$

Given a height of 150 cm

We need to predict whether the person is male or female.

Let a = -100; b = 0.6

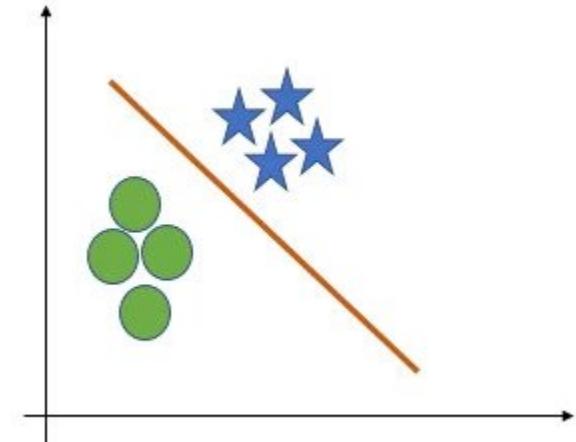
$$y = \frac{e^{(a+b \times X)}}{1+e^{(a+b \times X)}} = 0.000046$$

X	Y
0-8	0.27
9-16	0.5
17-24	0.6
25-32	0.66
33-40	0.93

Logistic Regression

- For a binary classification problem, target is (0 or 1)
- The Logistic function is defined as:

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



$$P = \frac{\exp(a + bX) / \exp(a + bX)}{1/\exp(a + bX) + \exp(a + bX)/\exp(a + bX)}$$

$$P = \frac{1}{1/\exp(a + bX) + 1}$$

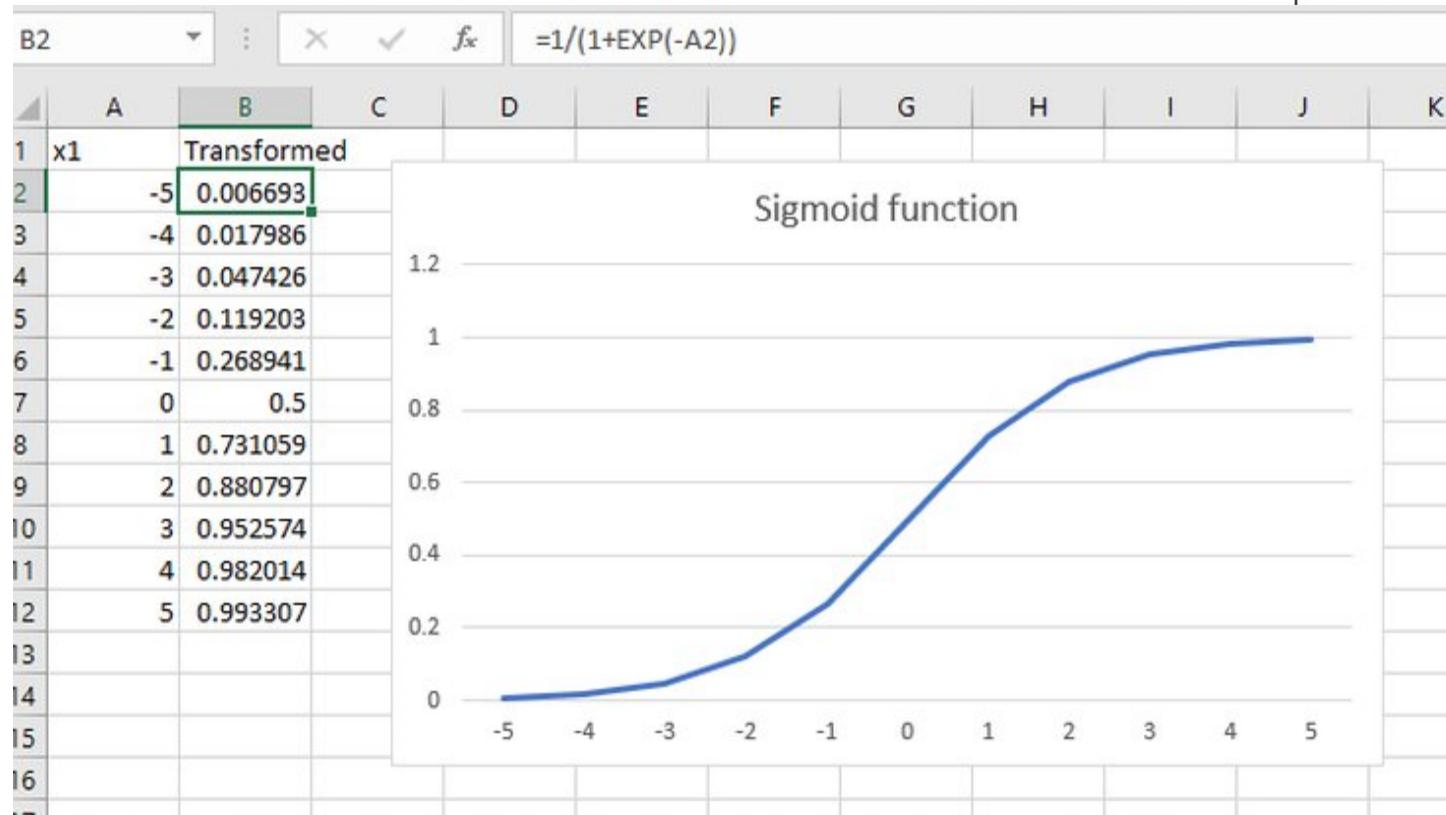
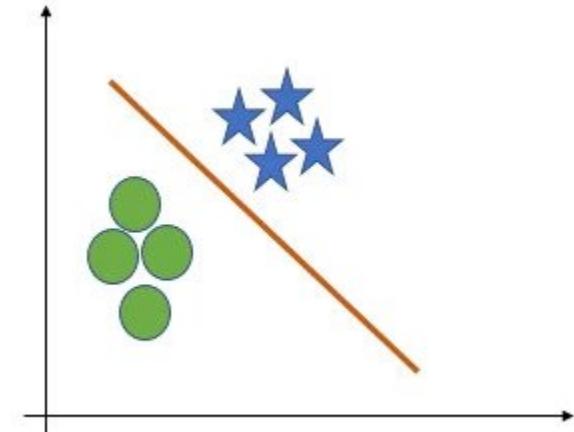
If, $z = (a + bX)$

$$P = \frac{1}{1/\exp(z) + 1} = \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression

- For a binary classification problem, target is (0 or 1)
- The Logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad \text{If, } z = (a + bX)$$



If probability is
> 0.5 then
default class
(class 0),
otherwise other
class (class 1)

Logistic Regression

The Logistic function is defined as:

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

$$Z = b_0 + b_1 * x_1 +$$

- The following is a dataset with 3 variables, where X1 and X2 are independent variable and Y is a dependent variable.

X1	X2	Y
2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
3.06407232	3.005305973	0
7.627531214	2.759262235	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
8.675418651	-0.242068655	1
7.673756466	3.508563011	1

Logistic Regression

$$Z = b_0 + b_1 * x_1 + b_2 * x_2$$

X1	X2	Y
2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
3.06407232	3.005305973	0
7.627531214	2.759262235	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
8.675418651	-0.242068655	1
7.673756466	3.508563011	1

- The job of the learning algorithm will be to discover the best values for the coefficients (b_0 , b_1 and b_2) based on the training data.
- Unlike linear regression, the output is transformed into a probability using the logistic function.

Logistic Regression

Logistic regression involves the following steps:

- Calculation of the Logit function that is $z = (a + bX)$
- Application of the Sigmoid function (Logistic function) to logit that is

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

- Calculation of the error, Cost function (Maximum Log-Likelihood).
- Application of learning algorithm to reduce the error and then repeat (if some error based learning algorithm is used)

Logistic regression by Stochastic Gradient Descent

- It works by using the model to calculate a prediction for each instance in training set and calculate error for each prediction.

Logistic Regression

We can calculate coefficients for logistic regression model as follows:

Given each training instance:

- Calculate a prediction using the current values of the coefficients.
- Calculate new coefficient values based on the error in the prediction.

This process is repeated until the model is accurate enough for fix number of iterations.

Probability of first training instance that belongs to class 0 that is $X_1 = 2.7810836$,

$x_2=2.550537003$, $Y=0$. Let

$b_0 = 0$; $b_1 = 0$; $b_2 = 0$

$$z = (b_0 + b_1 * x_1 + b_2 * x_2) = 0.0 + 0.0 * 2.7810836 + 0.0 * 2.550537003$$

$$\text{prediction} = 1 / (1 + e^{-z})$$

$$\text{prediction} = 1 / (1 + e^{(-(0.0 + 0.0 * 2.7810836 + 0.0 * 2.550537003))})$$

$$\text{Prediction } (f(z)) = 0.5$$

New Coefficients using gradient descent

$$b = b + \alpha * (y - \text{prediction}) * \text{prediction} * (1 - \text{prediction}) * x [\text{(change of weight} = \alpha \text{ex})]$$

Logistic Regression

New Coefficients using gradient descent

$$b = b + \alpha * (y - \text{prediction}) * \text{prediction} * (1 - \text{prediction}) * x$$

b_0 (intercept) will not have x value so it is assumed as 1 every time.

$$b_0 = 0 + 0.3 * (0 - 0.5) * 0.5 * (1 - 0.5) * 1.0 = -0.0375$$

$$b_1 = 0 + 0.3 * (0 - 0.5) * 0.5 * (1 - 0.5) * 2.7810836 = -0.104290635$$

$$b_2 = 0 + 0.3 * (0 - 0.5) * 0.5 * (1 - 0.5) * 2.550537003 = -0.09564513761$$

Now, repeat this process for $X_1 = 1.465489372$, $x_2 = 2.362125076$, $Y=0$.

$$b_0 = -0.0375; b_1 = -0.104290635; b_2 = -0.09564513761$$

$$z = (b_0 + b_1 * x_1 + b_2 * x_2) = -0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076)$$

$$\text{prediction} = 1 / (1 + e^{-(-0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076))})$$

$$\text{prediction} = 0.397$$

Logistic Regression

New Coefficients using gradient descent

Now, repeat this process for $X_1 = 1.465489372$, $x_2 = 2.362125076$, $Y=0$.

$b_0 = -0.0375$; $b_1 = -0.104290635$; $b_2 = -0.09564513761$

$$z = (b_0 + b_1 * x_1 + b_2 * x_2) = -0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076)$$

$$\text{prediction} = 1 / (1 + e^{-(-0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076))})$$

$$\text{prediction} = 0.397$$

Update:

$$b_0 = -0.0375 + 0.3 * (0 - 0.397) * 0.397 * (1 - 0.397) * 1.0 = -0.06605$$

$$b_1 = -0.104290635 + 0.3 * (0 - 0.397) * 0.397 * (1 - 0.397) * 1.465489372 = -0.1461$$

$$b_2 = -0.09564513761 + 0.3 * (0 - 0.397) * 0.397 * (1 - 0.397) * 2.362125076 = -0.1631$$

Logistic Regression

The first epoch coefficients are as follows:

X1	X2	Y	Prediction	Intercept	Coefficient X1	Coefficient X2
2.7810836	2.550537003	0	0.5	-0.0375	-0.104290635	-0.095645138
1.465489372	2.362125076	0	0.3974114	-0.06605	-0.146131968	-0.163086406
3.396561688	4.400293529	0	0.2175459	-0.07716	-0.183864977	-0.211970051
1.38807019	1.850220317	0	0.3263876	-0.09869	-0.213747009	-0.251801137
3.06407232	3.005305973	0	0.1808849	-0.10673	-0.238382985	-0.275964615
7.627531214	2.759262235	1	0.063777	-0.08996	-0.110466041	-0.229690614
5.332441248	2.088626775	1	0.2388946	-0.04844	0.110916445	-0.14297885
6.922596716	1.77106367	1	0.6144753	-0.02104	0.300586661	-0.094453991
8.675418651	-0.242068655	1	0.9314728	-0.01973	0.311970996	-0.094771646
7.673756466	3.508563011	1	0.885111	-0.01623	0.338866771	-0.082474472

$$b_0 = -0.01623; b_1 = 0.3388; b_2 = -0.0824$$

Logistic Regression

The 10th epoch is as follows:

X1	X2	Y	Prediction	Intercept	Coefficient X1	Coefficient X2	Prediction_round
2.781084	2.550537	0	0.316724951	-0.405242149	0.767058635	-1.101824964	0
1.465489	2.362125	0	0.131955957	-0.409776561	0.760413502	-1.112535813	0
3.396562	4.400294	0	0.06166044	-0.410846834	0.756778254	-1.117245328	0
1.38807	1.85022	0	0.193482972	-0.419904583	0.744205462	-1.134004159	0
3.064072	3.005306	0	0.175428153	-0.427517452	0.720879082	-1.156883159	0
7.627531	2.759262	1	0.867480845	-0.422947218	0.755738686	-1.144272685	1
5.332441	2.088627	1	0.77153982	-0.410866281	0.820159574	-1.119040116	1
6.922597	1.771064	1	0.963906322	-0.410489561	0.822767453	-1.118372921	1
8.675419	-0.24207	1	0.999087205	-0.410489311	0.822769619	-1.118372981	1
7.673756	3.508563	1	0.878613167	-0.406605464	0.852573316	-1.104746259	1

Thus, final coefficients are:

$$b_0 = -0.4066054641; b_1 = 0.8525733164; b_2 = -1.104746259$$

Now, prediction is < 0.5 then 0 else 1.

$$\text{accuracy} = (\text{correct predictions} / \text{number predictions made}) * 100$$

$$\text{accuracy} = (10 / 10) * 100$$

$$\text{accuracy} = 100\%$$

We can take new data and get prediction value

List of Popular Regression Algorithms

- [Linear Regression](#)
- [Polynomial Regression](#)
- [Logistic Regression](#)
- [Quantile Regression](#)
- [Ridge Regression](#)
- [Lasso Regression](#)
- [Elastic Net Regression](#)
- [Principal Components Regression \(PCR\)](#)
- [Partial Least Squares \(PLS\) Regression](#)
- [Support Vector Regression](#)
- [Ordinal Regression](#)
- [Poisson Regression](#)
- [Negative Binomial Regression](#)
- [Quasi Poisson Regression](#)
- [Cox Regression](#)
- [Tobit Regression](#)

Supervised ML Algorithms

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- **K- Nearest Neighbour (K-NN)**
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Nearest Neighbour Based Classifiers

One of the simplest classifiers that can be used for classification is the nearest neighbour.

It classifies a sample based on the category of its nearest neighbour.

When large samples are involved, nearest neighbour classifier gives better result than any other classifiers.

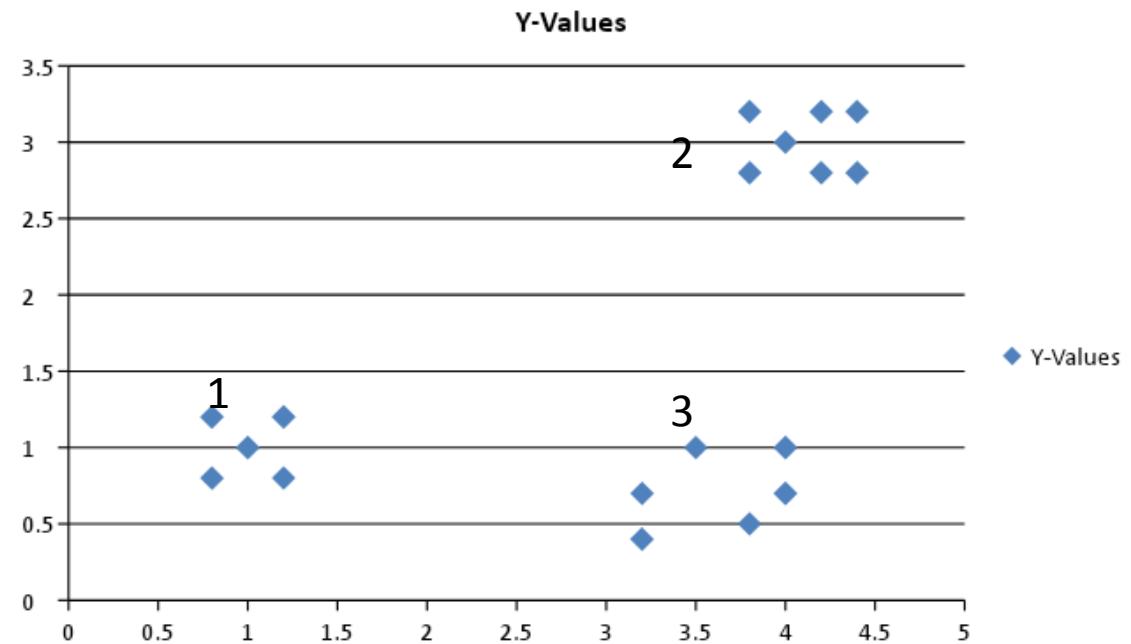
Nearest Neighbour Algorithm

The nearest neighbour algorithm assigns to a test sample the class label of its closest neighbour. Let there be n training patterns, (X_1, θ_1) , (X_2, θ_2) , (X_n, θ_n) where X_i is of dimension d and θ_i is the class label. If P is a test sample then if

$d(P, X_k) = \min\{d(P, X_i)\}$ where $i = 1, 2 \dots n$. Pattern P is assigned to the class θ_k associated with X_k .

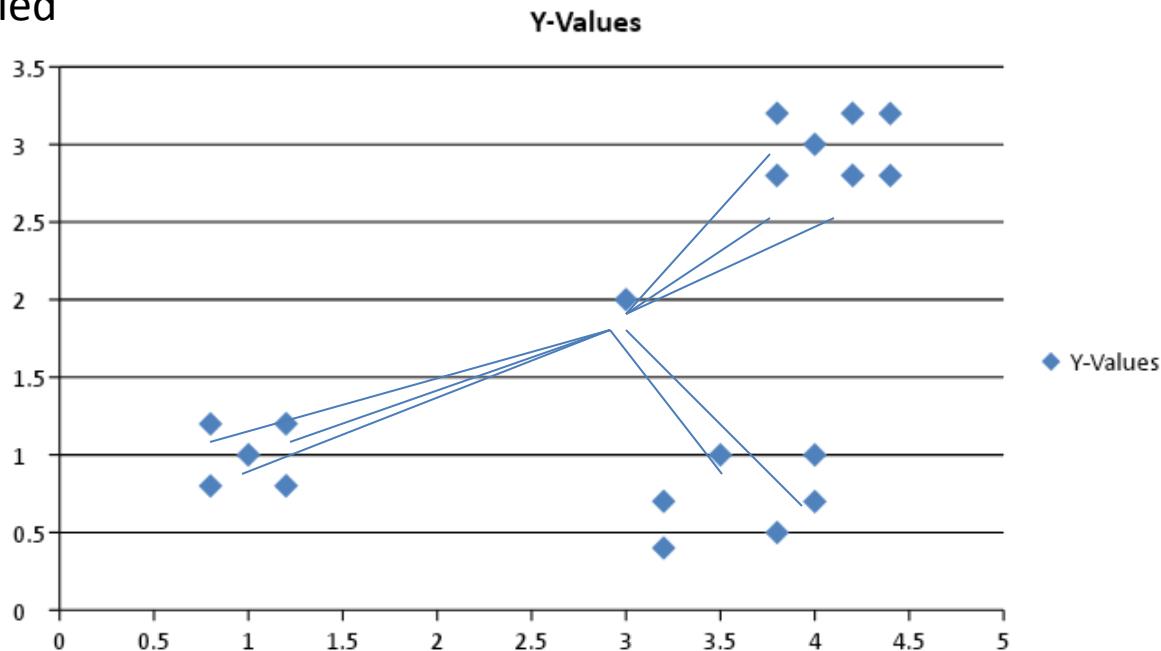
Nearest Neighbour Algorithm

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



Nearest Neighbour Algorithm

Suppose a new point (3.0, 2.0) P is given to be classified

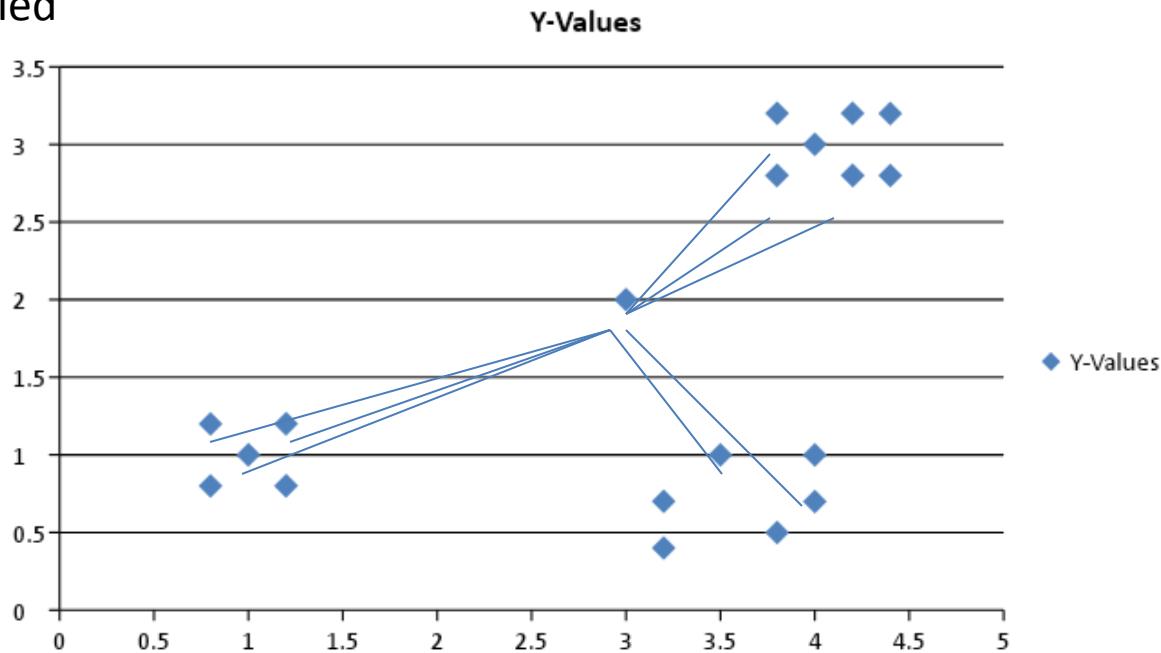


$$D(x_1, P) = \sqrt{(0.8 - 3.0)^2 + (0.8 - 2.0)^2} = 2.51$$

The closest neighbour of P is X16; Hence P belongs to class 3

k-Nearest Neighbour Algorithm

Suppose a new point (3.0, 2.0) P is given to be classified



If k is taken to be 5, the five nearest neighbours of P are X_{16} , X_7 , X_{14} , X_6 and X_{17} . The majority class of these five patterns is class 3. Hence P is classified as class 3.

The value chosen for k is crucial.

k-Nearest Neighbour Algorithm

This method will reduce the error in classification when training patterns are noisy.

For large dataset, k can be larger to reduce the error.

The value of k can be determined by experimentation using the concept of validation set.

- If P is a new pattern (4.2, 1.8), its nearest neighbour is X17 and hence P is classified to class 3.
- If the 5 nearest neighbours are taken, it is classified to class 2. The 5 nearest neighbours are X17, X16, X8, X7 and X11. X17 and X16 belonging to class 3, and X8, X7 and X11 belonging to class 2.

Drawbacks of Nearest Neighbour Algorithm

It's a lazy learning

For big dataset it takes huge time to find nearest neighbours

Modified k-Nearest Neighbour Algorithm (MkNN)

This algorithm is similar to the kNN algorithm.

The only difference is that these k nearest neighbours are weighted according to their distance from the test point.

This is also called the distance-weighted k-nearest neighbour algorithm.

Weight of each neighbour is defined as

$$w_j = \begin{cases} \frac{d_m - d_j}{d_m - d_1} & \\ & \dots \\ & 1 \end{cases}$$

The image shows a software window titled 'CLS Application'. It contains a table with columns labeled 'X' and 'Y'. Below the table, there is a formula: $w_j = \frac{d_m - d_j}{d_m - d_1}$. To the right of the formula, there is a note: 'Step 2 Divide output of step 1 by output of step 3 to calculate w'. Below that, it says 'Step 3 Calculate w using this value'. At the bottom, there is a button labeled 'OK'.

$$M_{Ext} = 19.13 + 1.89 \times M_{Int}$$

Modified k-Nearest Neighbour Algorithm (MkNN)

$$P = (3.0, 2.0)$$

Distances of the five nearest points from P are

$$d(P, X_{16}) = 1.12; \quad d(P, X_7) = 1.13; \quad d(P, X_{14}) = 1.32; \quad d(P, X_6) = 1.41; \quad d(P, X_{17}) = 1.41$$

$$w_{16} = 1$$

$$w_7 = \frac{1.41 - 1.13}{1.41 - 1.12} = 0.97$$

$$w_{14} = \frac{1.41 - 1.32}{1.41 - 1.12} = 0.31$$

$$w_6 = 0$$

$$w_{17} = 0$$

Class 1 sums = 0 (none of the patterns belongs to class 0)

Class 2 sums = $0.97 + 0 = 0.97$ (X7 and X6)

Class 3 sums = $1 + 0.31 + 0 = 1.31$ (X16, X14 and X17)

Finally P is classified to Class 3

r Nearest Neighbours

r-nearest neighbour takes all the neighbours within some distance r of the point of interest.
The algorithm is as follows:

Step1: Given the point P, determine the sub-set of data that lies in the ball of radius r centred at P.

$$B_r(P) = \{X_i \in X \mid \|P - X_i\| \leq r\}$$

Step2: If $B_r(P)$ is empty, then output the majority class of the entire dataset

Step3: If $B_r(P)$ is not empty, output the majority class of the data points in it.

This algorithm can be used to identify outliers.

The choice of the radius r is crucial to the algorithm.

P= (3.0, 2.0) patterns which are in a radius of 1.45 are X6, X7, X8, X9, X14, X16 and X17.
Majority patterns belong to class 2. P is therefore assigned to class 2.

Drawbacks of Nearest Neighbour Algorithm

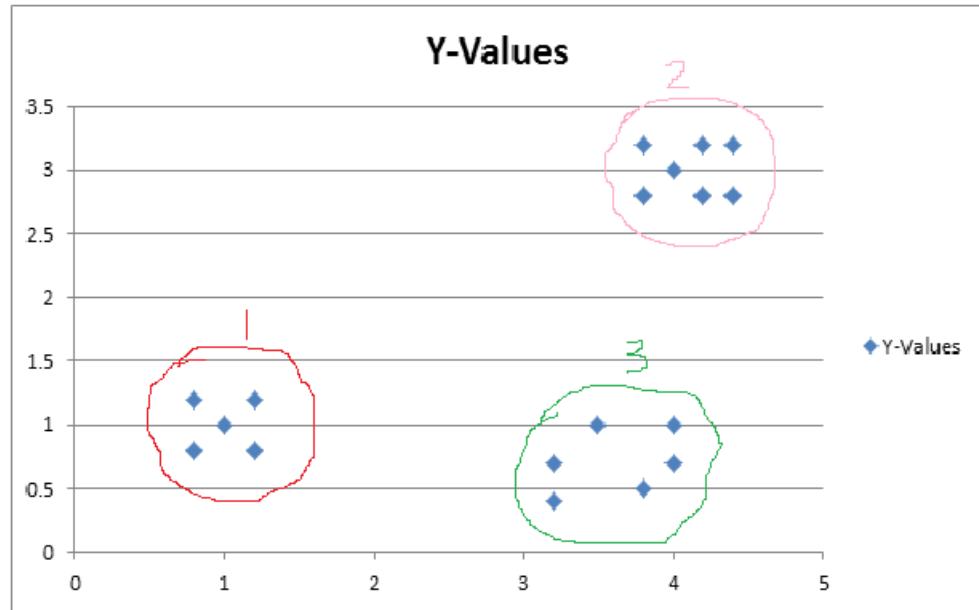
It's a lazy learning

For big dataset it takes huge time to find nearest neighbours

Nearest Neighbour with Clustering

Now a densed region can be represented by a representative pattern

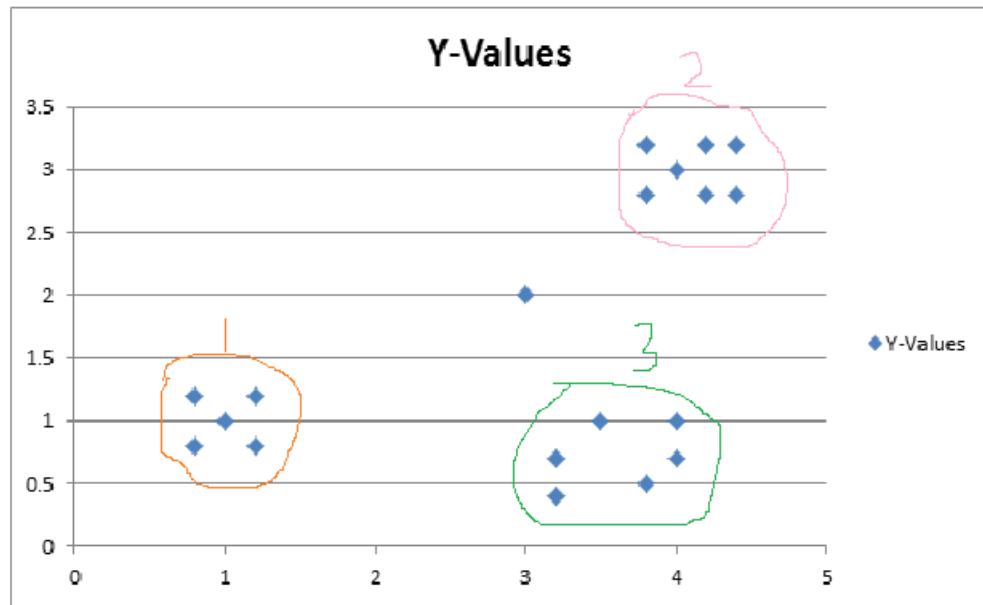
Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



Suppose centroid is the representative patterns
 Centroids are:
 $C1=(1.0, 1.0)$
 $C2=(4.11, 3)$
 $C3=(3.62, 0.72)$

Nearest Neighbour with Clustering

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



$$d(C_1, P) = 3.30$$

$$d(C_2, P) = 1.20$$

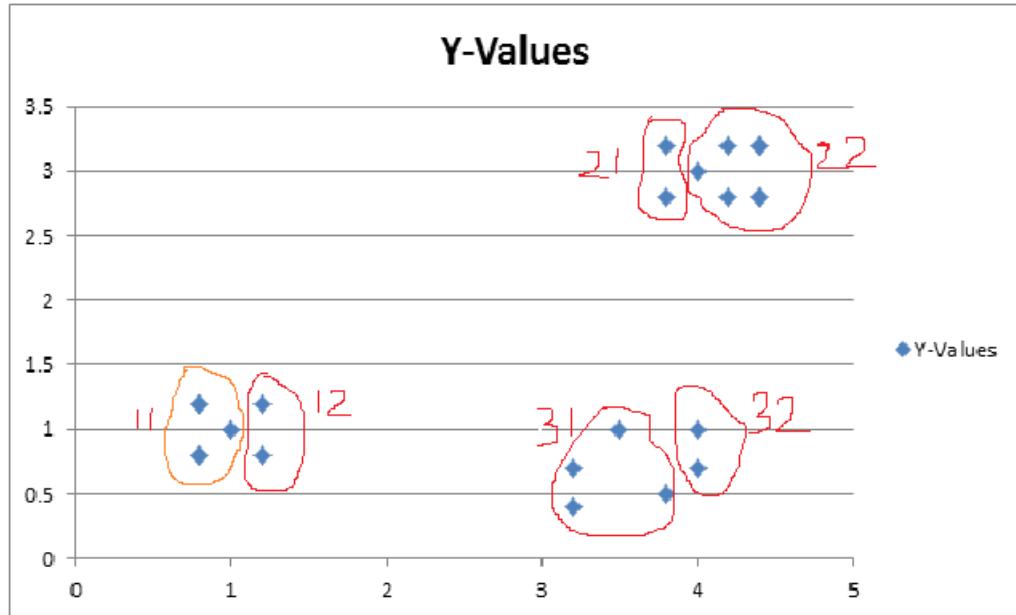
$$d(C_3, P) = 1.23$$

P belongs to class 2

Nearest Neighbour with Clustering

It is also possible to have more clusters for each class

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



$$C11 = (1.0, 0.867)$$

$$C12 = (1.0, 1.2)$$

$$C21 = (3.8, 3.0)$$

$$C22 = (4.24, 3.0)$$

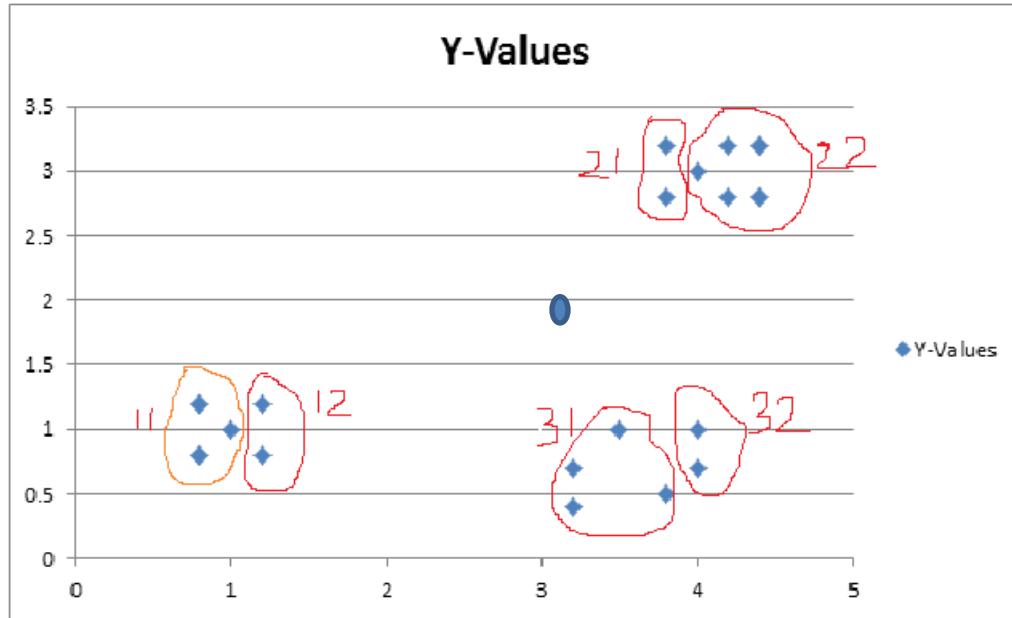
$$C31 = (3.43, 0.65)$$

$$C32 = (4.0, 0.85)$$

Nearest Neighbour with Clustering

It is also possible to have more clusters for each class

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



$$D(C_{11}, P) = 3.33$$

$$D(C_{12}, P) = 3.26$$

$$D(C_{21}, P) = 1.26$$

$$D(C_{22}, P) = 1.20$$

$$D(C_{31}, P) = 1.38$$

$$D(C_{32}, P) = 0.97$$

P is classified as class 3

Naive Bayes Classifier

Simplifying Bayes Classification

- Estimates probabilities of occurrence of different attribute values for the different classes in a training set.
- It uses these probabilities to classify recall patterns.

Name of pattern	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
T1	Gabby	Baked	Clogs	Student
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Student
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Student
T6	Quiet	Baked	Sandals	Student
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Student
R1	Quiet	Baked	Clogs	?
R2	Quiet	Roasted	Sandals	?
R3	Gabby	Roasted	Clogs	?
R4	Quiet	Roasted	Clogs	?

Simplifying Bayes Classification

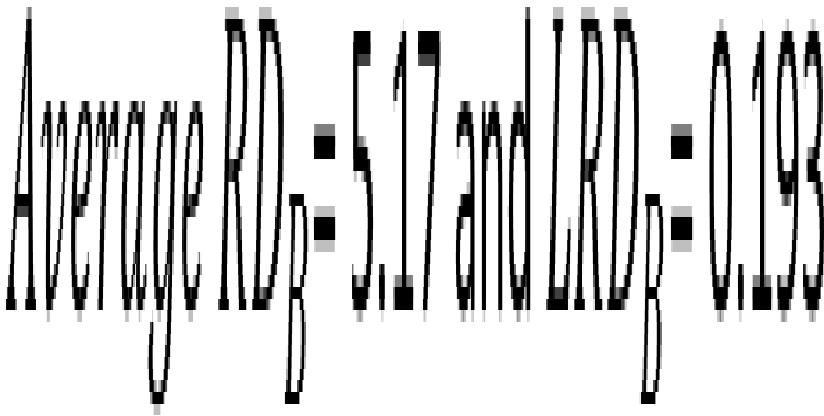
- o Estimates probabilities of occurrence of different attribute values for the different classes in a training set.
- o It uses these probabilities to classify recall patterns.

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals)	2/5	3/7

Simplifying Bayes Classification

- Suppose a training set has classes $C_1, C_2, C_3, \dots, C_m$, where ($m \geq 1$) and array of attributes $\bar{A} = A_1, A_2, A_3, \dots, A_M$ where ($M \geq 1$)
- The following probabilities are calculated.
 - **P (\bar{A})** = probability that a training pattern has attribute array \bar{A} .
 - **Prior probability, $P(C_k)$** : probability that a training pattern belongs to class C_k .
 - **Posterior probability, $P(C_k | \bar{A})$** : probability that a training pattern with attribute array \bar{A} belongs to class C_k . The attribute has discrete values.
 - **Conditional probability, $P (\bar{A} | C_k)$** : probability that a training pattern of class C_k has attribute array \bar{A} , the attributes having discrete values.

Simplifying Bayes Classification



Estimation of $P(C_k | \bar{A})$ from training set (cont..)

- Estimating $P(\bar{A}|\mathcal{C}_k)$ needs an impractically a large training set to consider values for all the attributes $A_1, A_2, A_3, \dots, A_M$. If these attributes $A_1, A_2, A_3, \dots, A_M$ are assumed to be class conditionally independent , then this classifier is a Naïve Bayes Classifier.

$$P(\bar{A}|\mathcal{C}_k) = \prod_{i=1}^M P(A_i|\mathcal{C}_k)$$

- To classify a pattern with attributes $A_1, A_2, A_3, \dots, A_M$, the equation,

is maximized, which is obtained by substituting $\prod_{i=1}^M P(A_i | C_k)$ for $P(\bar{A} | C_k)$ in equation (1).

Estimation of Prior Probabilities

- Let the number of patterns in class C_k is $|C_k|$ for $1 \leq k \leq m$.
 - In case of maximum-likelihood estimation,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}, (P(C_k) \geq 0) \dots \dots \dots (3)$$

- Alternatively, according to the Bayesian estimation

$$P(C_k) = \frac{|C_k|+1}{m + \sum_{j=1}^m |C_j|}, (P(C_k) > 0) \dots \dots \dots (4)$$

Estimation of Conditional Probabilities

- To maximize $P(\mathcal{C}_k) \prod_{i=1}^M P(A_i | \mathcal{C}_k)$, probability of $\prod_{i=1}^M P(A_i | \mathcal{C}_k)$ is required.
- The possible values of A_i be $V_{i_1}, V_{i_2}, V_{i_3}, \dots, V_{i_n}$ for $1 \leq i \leq M$.
- $|\mathcal{C}_k^{ij}|$ be the number of training patterns of class \mathcal{C}_k for which the attribute A_i is V_j where $1 \leq k \leq m, i_1 \leq j \leq i_n$.
- According to Maximum-likelihood estimation,

$$P(A_i = V_j | \mathcal{C}_k) = \frac{|\mathcal{C}_k^{ij}|}{|\mathcal{C}_k|}$$

- According to Bayesian estimation,

$$P(A_i = V_j | \mathcal{C}_k) = \frac{|\mathcal{C}_k^{ij}| + 1}{i_n + |\mathcal{C}_k|}$$

EXAMPLE

Name of pattern	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
T1	Gabby	Baked	Clogs	Student
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Student
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Student
T6	Quiet	Baked	Sandals	Student
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Student
R1	Quiet	Baked	Clogs	?
R2	Quiet	Roasted	Sandals	?
R3	Gabby	Roasted	Clogs	?
R4	Quiet	Roasted	Clogs	?

Name of pattern	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
T1	Gabby	Baked	Clogs	Student
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Student
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Student
T6	Quiet	Baked	Sandals	Student
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Student
R1	Quiet	Baked	Clogs	?
R2	Quiet	Roasted	Sandals	?
R3	Gabby	Roasted	Clogs	?
R4	Quiet	Roasted	Clogs	?

Average RD_A =

$$\frac{1}{3} \sum_3 \max \left[\text{3rd distance of } A's \text{ neighbor}, \text{distance}(A, \text{the neighbor}) \right]$$

$$\begin{aligned}
 &= \frac{1}{3} [\max(5, 6.08) + \max(4.2, 6.32) + \max(1.4, 5.8)] \\
 &= \frac{1}{3} [6.08 + 6.32 + 5.8] = 6.06
 \end{aligned}$$

All the Probabilities

Probability	Estimates	
	Maximum-likelihood	Bayesian
$P(\text{professor})$	3/8	4/10
$P(\text{HABIT} = \text{gabby}) \text{professor}]$	2/3	3/5
$P(\text{HABIT} = \text{quiet}) \text{professor}]$	1/3	2/5
$P(\text{EATS} = \text{baked}) \text{professor}]$	0/3	1/6
$P(\text{EATS} = \text{fried}) \text{professor}]$	2/3	3/6
$P(\text{EATS} = \text{roasted}) \text{professor}]$	1/3	2/6
$P(\text{FOOTWEAR} = \text{clogs}) \text{professor}]$	0/3	1/5
$P(\text{FOOTWEAR} = \text{sandals}) \text{professor}]$	3/3	4/5
$P(\text{student})$	5/8	6/10
$P(\text{HABIT} = \text{gabby}) \text{student}]$	3/5	4/7
$P(\text{HABIT} = \text{quiet}) \text{student}]$	2/5	3/7
$P(\text{EATS} = \text{baked}) \text{student}]$	3/5	4/8
$P(\text{EATS} = \text{fried}) \text{student}]$	2/5	3/8
$P(\text{EATS} = \text{roasted}) \text{student}]$	0/5	1/8
$P(\text{FOOTWEAR} = \text{clogs}) \text{student}]$	3/5	4/7
$P(\text{FOOTWEAR} = \text{sandals}) \text{student}]$	2/5	3/7

Classifying the Professor-Student patterns

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals) student]	2/5	3/7

- Let us classify the R3 pattern of the professor-student recall set.

- The attribute values of the pattern are HABIT = gabby, EATS = roasted and FOOTWEAR = clogs.

$$\begin{aligned} P(C_k | \bar{A}) &= \frac{P(C_k)P(A|C_k)}{P(\bar{A})} = P(C_k)P(\bar{A}|C_k) \\ &= P(C_k) \prod_{i=1}^M P(A_i|C_k) \end{aligned}$$

- Using maximum-likelihood estimates belong to Professor,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}$$

$$P(A_t = V_j | C_k) = \frac{|C_k^{ij}|}{|C_k|}$$

$P(\text{professor}) \times P(\text{HABIT} = \text{gabby}) | \text{professor}] \times P[(\text{EATS} = \text{roasted}) | \text{professor}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{professor}]$

$$= \frac{3}{8} * \frac{2}{3} * \frac{1}{3} * \frac{0}{3} = 0$$

Classifying the Professor-Student patterns

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals) student]	2/5	3/7

- Using **maximum-likelihood estimates** belong to student class

$P(\text{student}) \times P(\text{HABIT} = \text{gabby}) | \text{student}] \times P[(\text{EATS} = \text{roasted}) | \text{student}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{student}]$

$$= \frac{5}{8} * \frac{3}{5} * \frac{0}{5} * \frac{3}{5} = 0$$

- The recall pattern R3 is rejected because the values of both the classes are zero. This is a disadvantage of maximum-likelihood estimates of probabilities, for one zero probability has nullified the influence of the other probabilities, for each class.

Classifying the Professor-Student patterns (cont..)

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals) student]	2/5	3/7

- Using Bayesian estimation belong to Professor
- $$P(\text{professor}) \times P(\text{HABIT} = \text{gabby} | \text{professor}) \times P[(\text{EATS} = \text{roasted}) | \text{professor}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{professor}]$$
- $$= \frac{2}{5} * \frac{3}{5} * \frac{1}{3} * \frac{1}{5} = 0.016$$

Belong to student:

- $$P(\text{student}) \times P(\text{HABIT} = \text{gabby} | \text{student}) \times P[(\text{EATS} = \text{roasted}) | \text{student}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{student}]$$
- $$= \frac{3}{5} * \frac{4}{7} * \frac{1}{8} * \frac{4}{7} = 0.0245$$
- Since, the value of student is 0.0245 which is more than the value of the professor which is 0.016, the recall pattern R3 is classified as student.

Classifying the Professor-Student patterns (cont..)

Recall Patterns	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor
R3	Gabby	Roasted	Clogs	Student
R4	Quiet	Roasted	Clogs	Student

Naïve Bayes with Continuous Attribute

- It performs well in case of [categorical data as compared to numeric data.](#)
- So, how do we perform classification using [Naïve Bayes](#) when the data is continuous in nature.
- There are two ways to handle continuous attributes in naïve Bayes classifiers:
 - i. We can discretize each continuous attribute and then replace the continuous [attribute](#) value with its corresponding discrete interval. However the estimation error depends on the discretization strategy, as well as the number of discrete intervals. If the number of intervals is too large, there are too few training records in each interval to provide a reliable estimate. if the number of intervals is too small, then some intervals may aggregate records from different classes and we may miss the correct decision boundary. Hence, there is no rule of thumb on the discretisation strategy.

Bayes

Problems with discretization strategy

Humidity

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

Large no of intervals

65-69	1
70-75	4
76-80	2
81-85	1
86-90	2
91-95	2
96-100	1

Less no of intervals

65-85	8 (6 yes, 2 no)
86- 100	6 (3 yes, 3 no)

Bayes

ii. We can assume a probability distribution for the continuous variable and estimate the parameters of the distribution using the training data. A Gaussian distribution is usually chosen to represent the class-conditional probability for continuous attributes. The distribution is characterized by two parameters, its mean and variance.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

where μ_a is the *sample mean*: $\mu_a = \frac{1}{|D_a|} \sum_{x \in D_a} x.a$
 σ_a is the *sample standard deviation*, and
 σ_a^2 the *sample variance*: $\sigma_a^2 = \frac{1}{|D_a|-1} \sum_{x \in D_a} (x.a - \mu_a)^2$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

- Maximum-likelihood estimation,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}$$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

- Maximum-likelihood estimation,

$$P(A_i = V_j | C_k) = \frac{|C_k|^{ij}}{|C_k|}$$

$$P(\text{Outlook=sunny} | \text{Yes}) = \frac{2}{9}$$

$$P(\text{Outlook=overcast} | \text{Yes}) = \frac{4}{9}$$

$$P(\text{Outlook=Rainy} | \text{Yes}) = \frac{3}{9}$$

$$P(\text{Outlook=sunny} | \text{No}) = \frac{3}{5}$$

$$P(\text{Outlook=overcast} | \text{No}) = \frac{0}{5}$$

$$P(\text{Outlook=Rainy} | \text{No}) = \frac{2}{5}$$

$$P(\text{Windy=false} | \text{Yes}) = \frac{6}{9}$$

$$P(\text{Windy=true} | \text{Yes}) = \frac{3}{9}$$

$$P(\text{Windy=false} | \text{No}) = \frac{2}{5}$$

$$P(\text{Windy=true} | \text{No}) = \frac{3}{5}$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

- Maximum-likelihood estimation,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}$$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

Continuous attribute:

Temperature: Yes : 83, 70, 68, 64, 69, 75, 75, 72, 81

$$\mu_T = 73; \sigma_T = 6.2$$

Temperature: No : 85, 80, 65, 72, 71

$$\mu_T = 75; \sigma_T = 7.9$$

Humidity: Yes : 86, 96, 80, 65, 70, 80, 70, 90, 75

$$\mu_H = 79; \sigma_H = 10.2$$

Temperature: No : 85, 90, 70, 95, 91

$$\mu_H = 86; \sigma_H = 9.7$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

We are going to classify an instance
 $x = \langle \text{Outlook}=\text{sunny}, \text{Temperature}=66, \text{Humidity}=90, \text{Windy=True} \rangle$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

Probability for Play= Yes

$$P(\text{Outlook}=\text{sunny} | \text{Yes}) = \frac{2}{9}$$

$$P(\text{Windy}=\text{true} | \text{Yes}) = \frac{3}{9}$$

Temperature=66

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

$$x = 66; \mu_T = 73; \sigma_T = 6.2$$

$$f(x=\text{Temp}) = \frac{1}{\sqrt{2\pi} \cdot 6.2} e^{-\frac{(66-73)^2}{2(6.2)^2}} = \frac{1}{\sqrt{38.955}} e^{-\frac{49}{76.88}}$$

$$= 0.16e^{-0.64} = 0.084$$

Humidity: $x = 90; \mu_H = 79; \sigma_H = 10.2$

$$f(x=\text{Hum}) = \frac{1}{\sqrt{2\pi} \cdot 10.2} e^{-\frac{(90-79)^2}{2(10.2)^2}} = \frac{1}{\sqrt{64.056}} e^{-\frac{121}{208.08}}$$

$$= 0.13e^{-0.58} = 0.073$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

We are going to classify an instance
 $x = \langle \text{Outlook}=\text{sunny}, \text{Temperature}=66, \text{Humidity}=90, \text{Windy=True} \rangle$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

Posterior probability of x to belong to Yes class
 $(play=yes)$

$$\begin{aligned} P(x/\text{yes}) * P(\text{yes}) &= P(\text{sunny}/\text{yes}) * \\ P(\text{Temperature}=66/\text{yes}) * P(\text{Humidity}=90/\text{yes}) * \\ P(\text{True}/\text{yes}) * P(\text{yes}) \\ &= (2/9) * 0.084 * 0.073 * (3/9) * (9/14) = 0.00029 \end{aligned}$$

Probability for Play= No

$$\begin{aligned} P(\text{Outlook}=\text{sunny} | \text{No}) &= \frac{3}{5} \\ P(\text{Windy}= \text{true} | \text{No}) &= \frac{3}{5} \end{aligned}$$

Temperature=66

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

$$x = 66; \mu_T = 75; \sigma_T = 7.9$$

$$\begin{aligned} f(x=\text{Temp}) &= \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}} = \\ \frac{1}{\sqrt{49.612}} e^{-\frac{81}{124.84}} \end{aligned}$$

$$= 0.14 e^{-0.65} = 0.073$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

We are going to classify an instance

$x = \langle \text{Outlook}=\text{sunny}, \text{ Temperature}=66, \text{ Humidity}=90, \text{ Windy}=\text{True} \rangle$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

Humidity: $x = 90; \mu_H = 86; \sigma_H = 9.7$

$$\begin{aligned} f(x=\text{Hum}) &= \frac{1}{\sqrt{2 \cdot 3.14 \cdot 9.7}} e^{-\frac{(90-86)^2}{2(9.7)^2}} = \\ &\frac{1}{\sqrt{60.92}} e^{-\frac{16}{188.18}} \\ &= 0.128 e^{-0.085} = 0.12 \end{aligned}$$

Posterior probability of x to belong to No class
($\text{play}=\text{no}$)

$$\begin{aligned} P(x/\text{no}) * P(\text{no}) &= P(\text{sunny}/\text{no}) * P(\text{Temperature}=66/\text{no}) \\ &\quad * P(\text{Humidity}=90/\text{no}) * P(\text{True}/\text{no}) * P(\text{no}) \\ &= (3/5) * 0.073 * 0.12 * (3/5) * (5/14) = 0.00113 \end{aligned}$$

$0.00113 > 0.00113 = (\text{play}=\text{no}) > (\text{play}=\text{yes})$:

Classification — NO

Decision Tree

- Decision Tree produces **interpretable** output in human readable form.
- **Decision rules** are constructed directly from decision tree output, traversing path from the root node to a given leaf node.
- Decision rules have form **IF antecedent THEN consequent**.
- Antecedent consists of attributes values from branches of given path.

Classifying the Recall patterns

If Footwear = Clogs

Then pattern class= Student

If Footwear = Sandals

and Eats = Baked,

Then pattern class= Student

If Footwear = Sandals

and Eats = Fried,

Then pattern class=

Professor

If Footwear = Sandals

and Eats = Roasted,

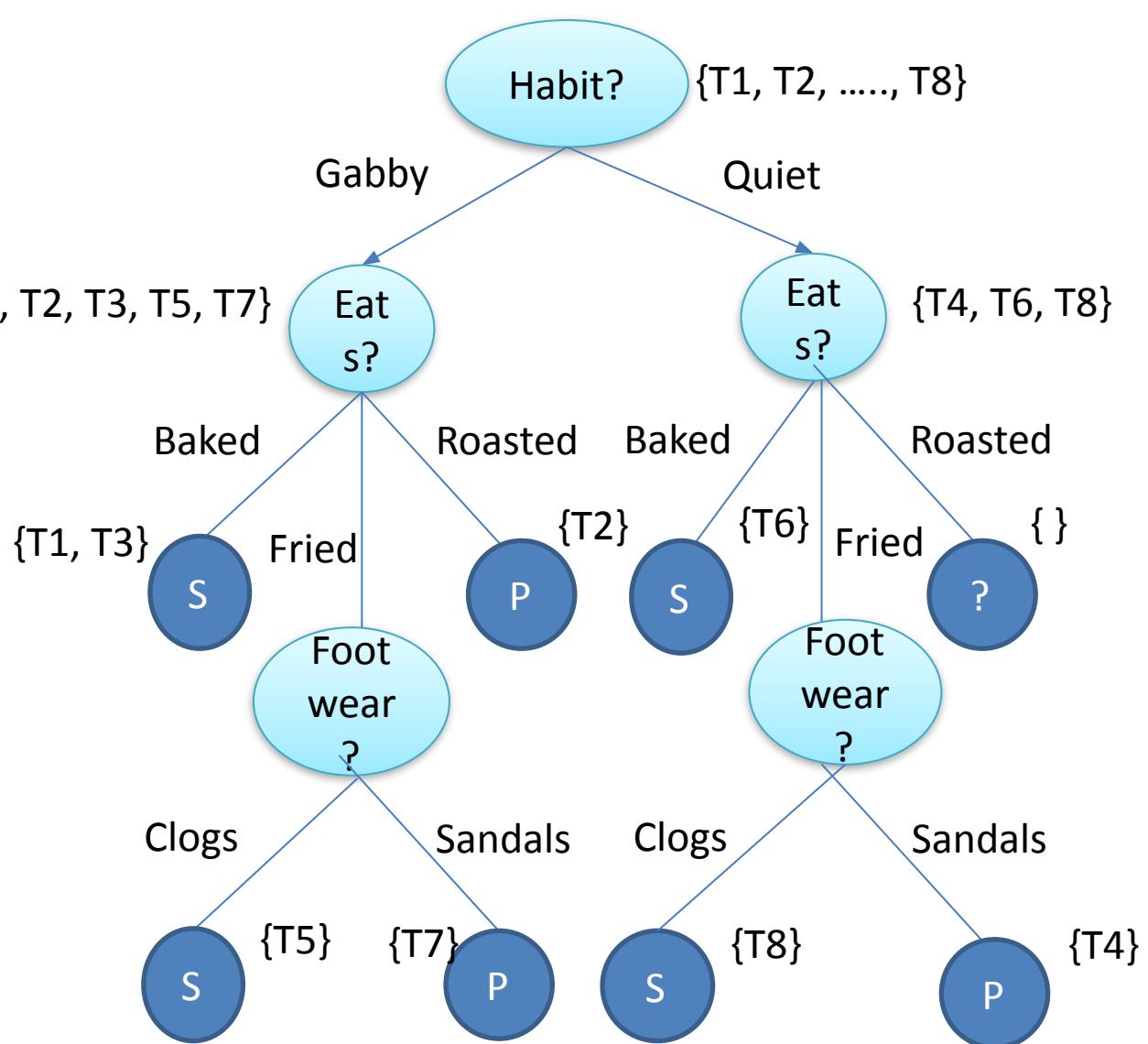
Then pattern class= Professor

Recall	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor
R3	Gabby	Roasted	Clogs	Student
R4	Quiet	Roasted	Clogs	Student

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	Students
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Students
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Students
T6	Quiet	Baked	Sandals	Students
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Students

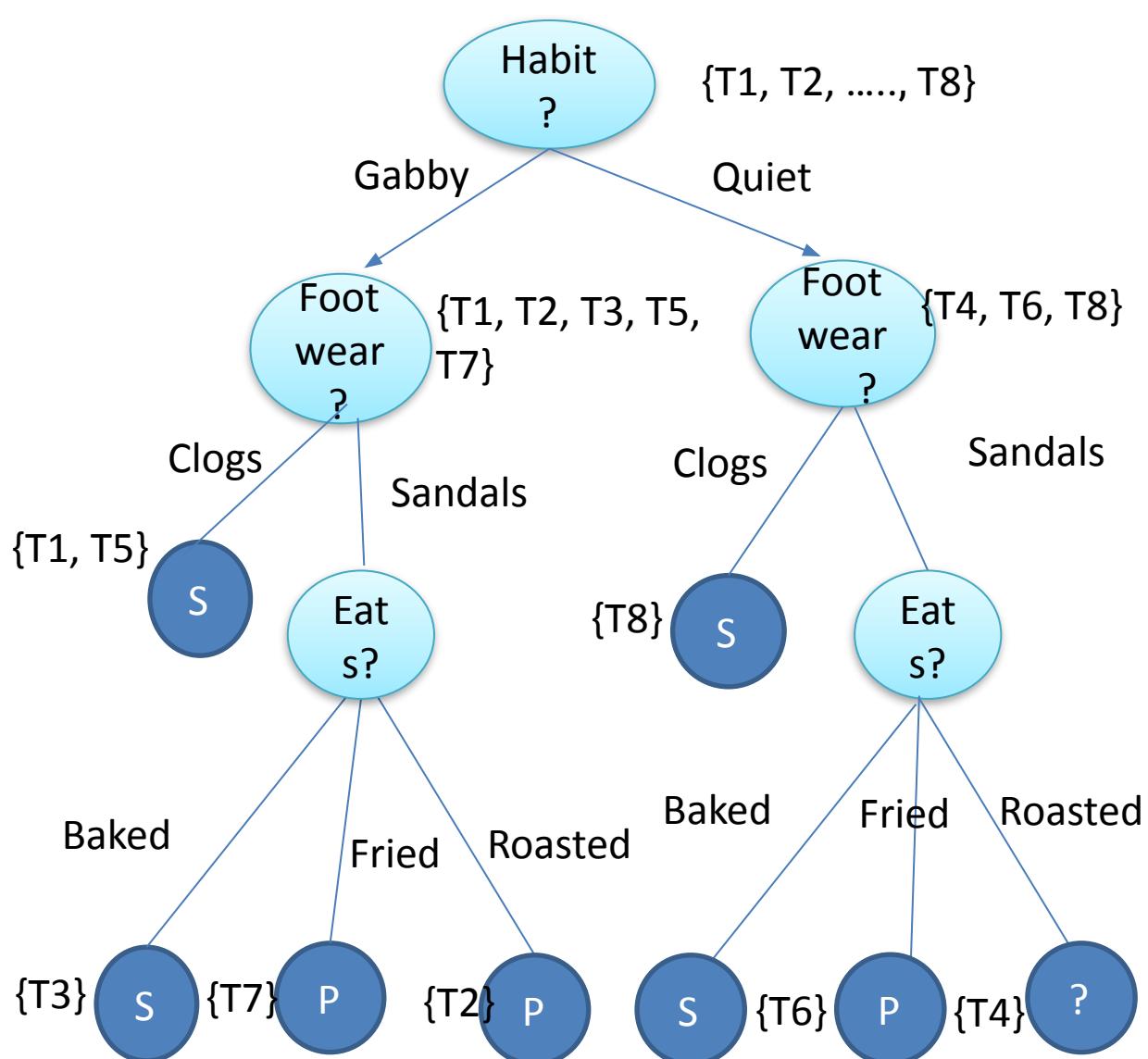
NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022



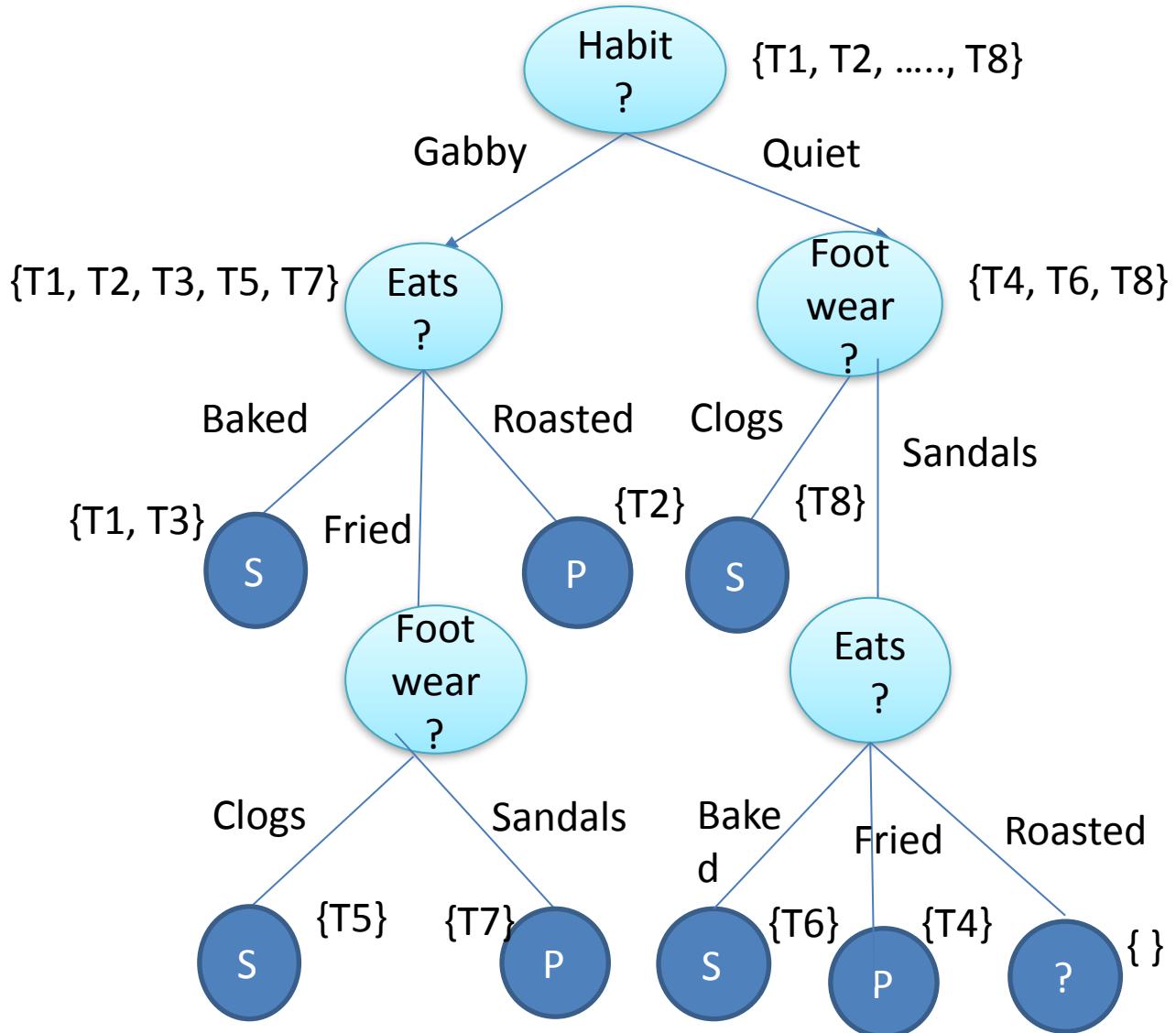
NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022



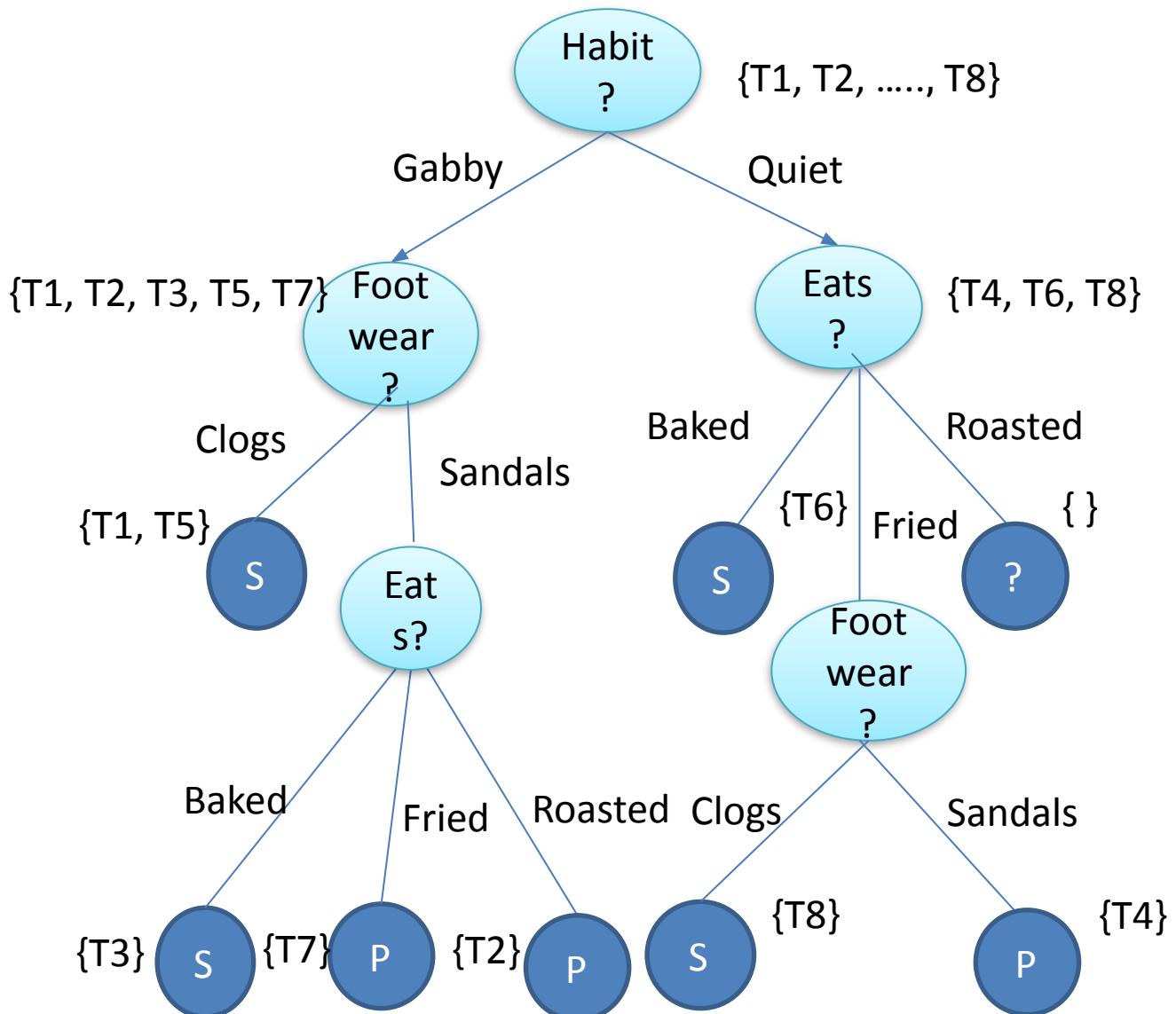
NAME of trainin g patter n	Attributes			Clas s
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022



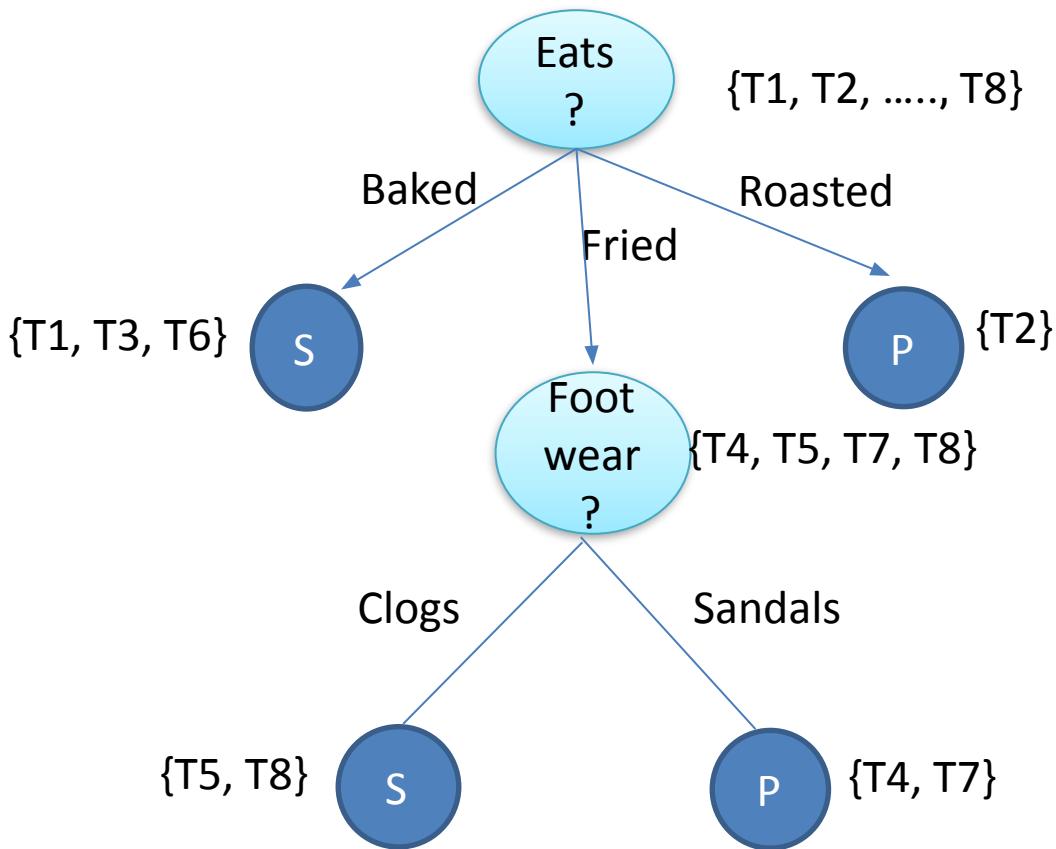
NAME of trainin g patter n	Attributes			Clas s
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Baked	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Baked	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022



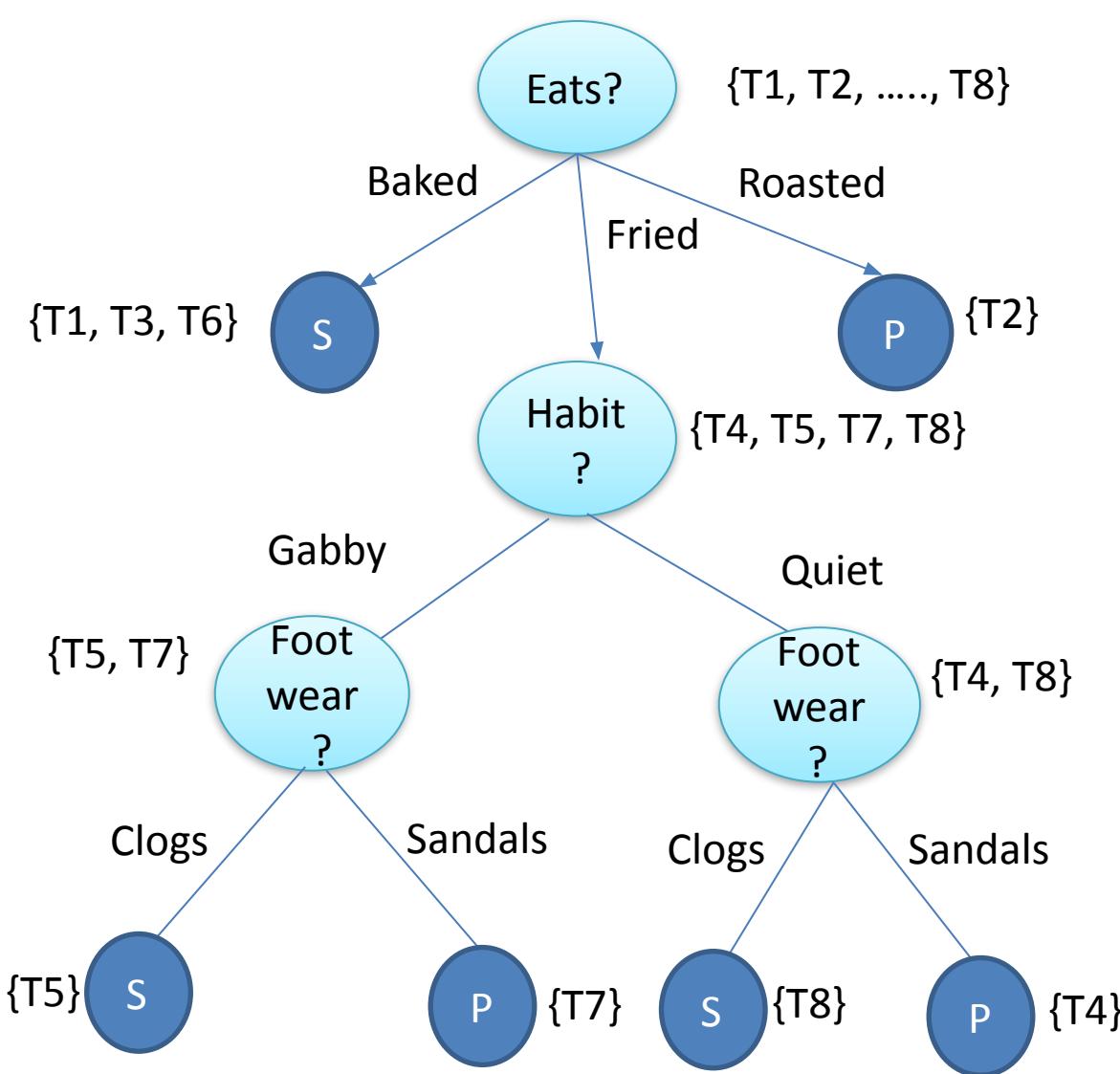
NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022

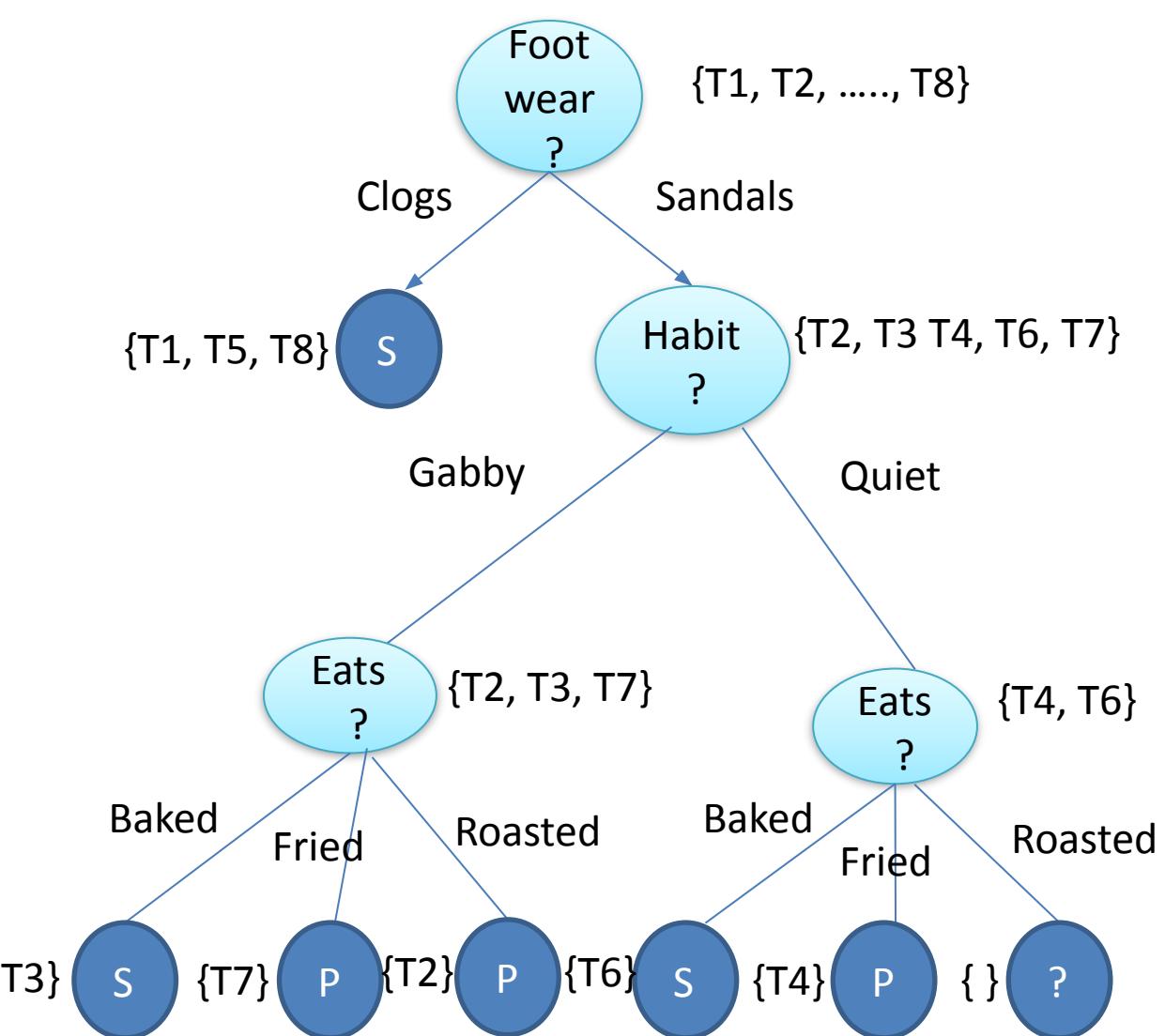


NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

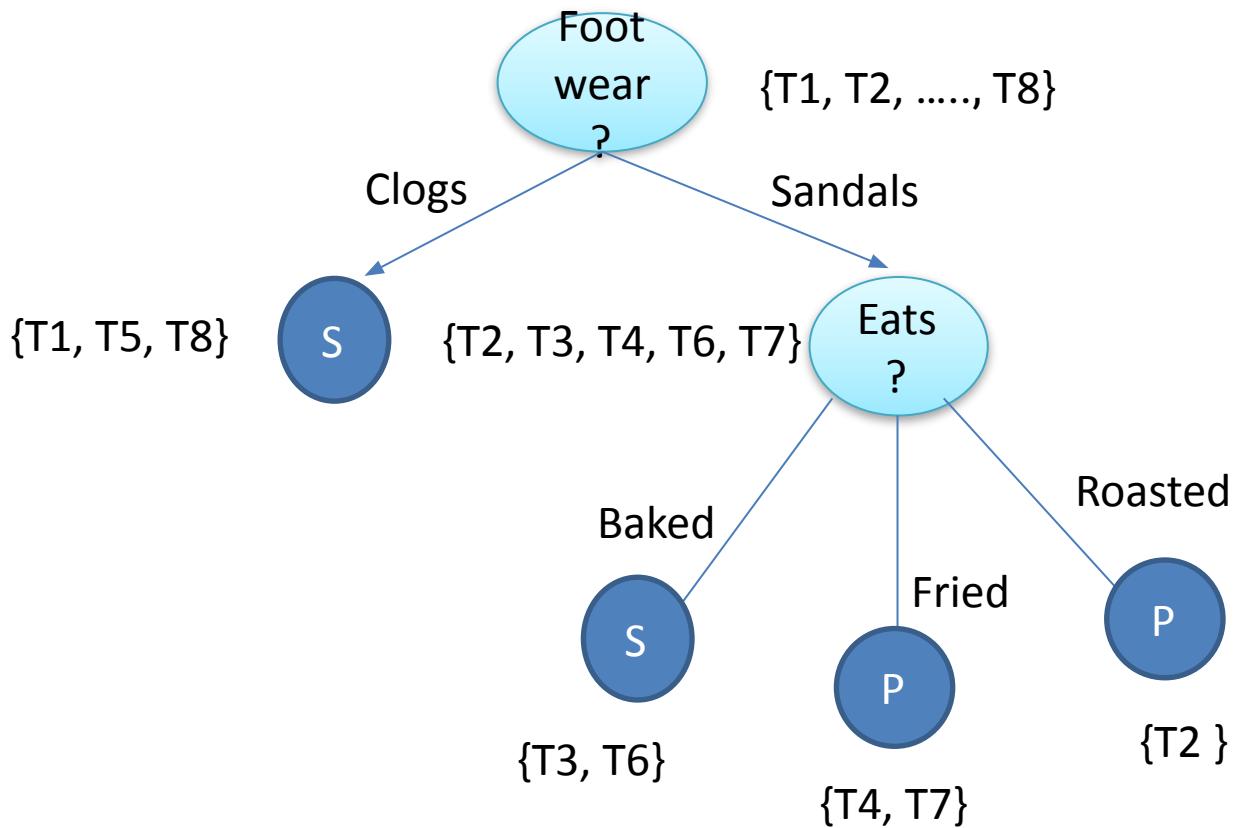
25-11-2022



NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

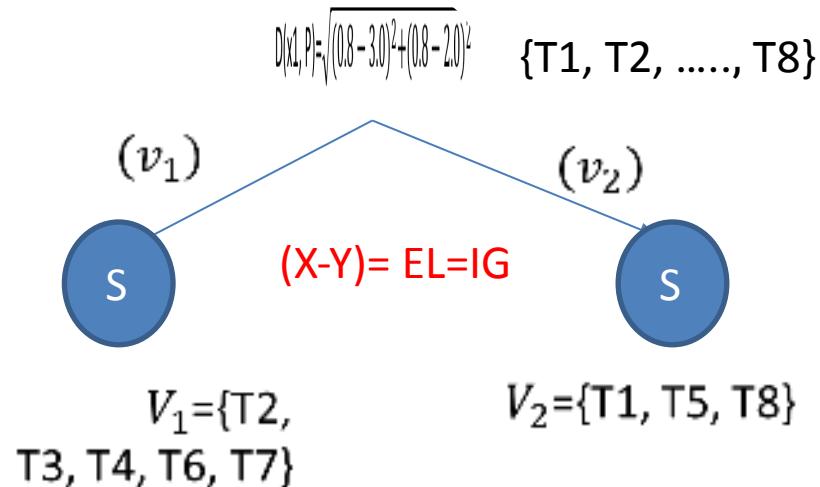


Classifying the Recall patterns

Recall Patterns	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor (some Rejected)
R3	Gabby	Roasted	Clogs	Student/ Professor
R4	Quiet	Roasted	Clogs	Student/ Professor

Ratio of Information Gain

$$\begin{aligned} & \frac{1}{3} \left[\max\left(3^{\text{th}} \text{ dist } B, \text{dist}(AB)\right) + \max\left(3^{\text{th}} \text{ dist } C, \text{dist}(AC)\right) + \right. \\ & \quad \left. \max\left(3^{\text{th}} \text{ dist } D, \text{dist}(AD)\right) \right] \end{aligned}$$



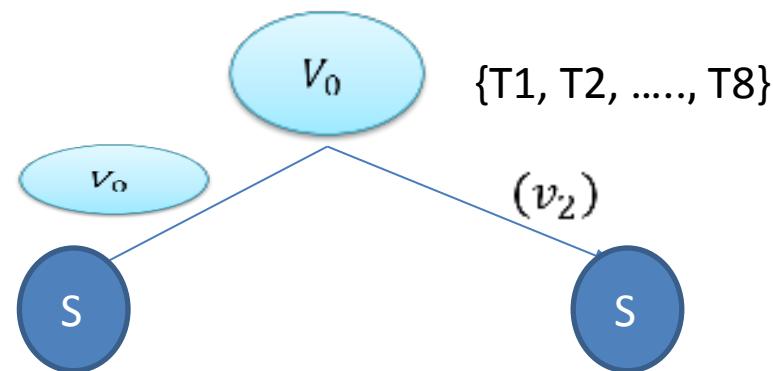
NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Ratio of Information Gain

- The weighted average information required to classify a pattern in set V_0 into one of the m classes is expressed as

$$I(V_0) = \sum_{k=1}^m \frac{Y(k, 0)}{Z(0)} (-\log \frac{Y(k, 0)}{Z(0)})$$

$I(V_0)$ is called the entropy of the set V_0 .



Density is reverse of distance therefore Local Reachability score LRD

$$\begin{aligned} LRD_A &= \frac{1}{RD_A} \\ &= 1/6.06 = 0.165 \end{aligned}$$

- Similarly for the set V_j ,

$$I(V_j) = \sum_{k=1}^m \frac{Y(k, j)}{Z(j)} (-\log \frac{Y(k, j)}{Z(j)})$$

- The weighted average information required to classify a pattern into one of class k in set V_0 after it has been split by the attribute A into sets V_1 to V_n is given by

$$I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} I(V_j)$$

- $I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right)$
 $I_A(V_0)$ is called the entropy of the attribute A for the set V_0
- The gain in information caused by attribute A splitting set V_0 into sets V_1 to V_n is

$$g_A(V_0) = I(V_0) - I_A(V_0)$$

Split information

$$\frac{Z(j)}{Z(0)}$$

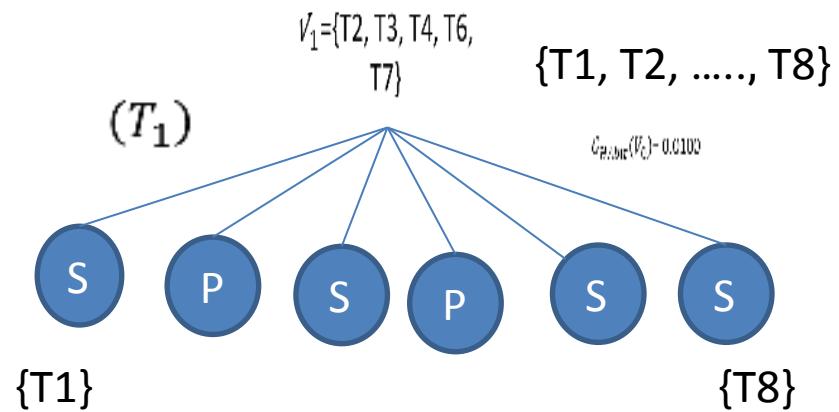
Information needed to extract set V_j from V_0 is

$$-\log \frac{Z(j)}{Z(0)}$$

- The weighted average information needed by attribute A to split set V_0 into sets V_1 to V_n is

$$S_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \left(-\log \frac{Z(j)}{Z(0)} \right)$$

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



- The ratio of information gain of the attribute A for set V_0 is defined as

$$G_A(V_0) = \frac{g_A(V_0)}{S_A(V_0)}$$

- The ratio of information gain is represented as

$$G_A(V_0) = \frac{I(V_0) - I_A(V_0)}{S_A(V_0)}$$

Example

Evaluate the entropy $I(V_0)$ of the Professor-student training set.

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTW EAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roast ed	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Node no. 3,6,9,12,16 are unsuccessful path
The average unsuccessful path length is given by node 6 and 16 which is $\frac{5}{2}$. Since 2 trees are considered therefore $h(x)$ is the average path of the datapoint for the 2 trees.

$$\begin{aligned}
 s_1 &= 2^{-\left(\frac{3+4/2}{5}\right)} = 2^{-(0.7)} = 0.615 \\
 s_2 &= 2^{-\left(\frac{3+3/2}{5}\right)} = 2^{-(0.6)} = 0.65 \\
 s_3 &= 2^{-\left(\frac{5+5/2}{5}\right)} = 2^{-(1)} = 0.5 \\
 s_4 &= 2^{-\left(\frac{4+3/2}{5}\right)} = 2^{-(0.7)} = 0.615 \\
 s_5 &= 2^{-\left(\frac{1+2/2}{5}\right)} = 2^{-(0.3)} = 0.812 \\
 s_6 &= 2^{-\left(\frac{5+5/2}{5}\right)} = 2^{-(1)} = 0.5 \\
 s_7 &= 2^{-\left(\frac{3+3/2}{5}\right)} = 2^{-(0.6)} = 0.65 \\
 s_8 &= 2^{-\left(\frac{4+3/2}{5}\right)} = 2^{-(0.7)} = 0.615
 \end{aligned}$$

$$G_{Footwear}(V_0) = \frac{I(V_0) - I_{Footwear}(V_0)}{I(V_0) - I_{Footwear}(V_0)}$$

$$= (0.9544 - 0.6066)$$

$$= 0.3478$$

The ratio of information gain of HABIT

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

$$= \begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

$$\text{eigenvalues} = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

α_2

FinalData = RowFeatureVector \times RowDataAdjust

The ratio of information gain of EATS

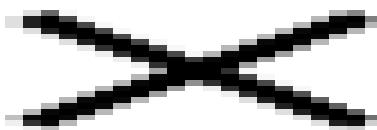
$$s_1 = \frac{l_1 + l_2 + l_3}{3} = x_1$$

$$s_1 = \frac{f_1 + f_2 + f_3}{3} = x_1$$

$$\begin{pmatrix} -.68 & -.74 \end{pmatrix}$$

- $V_1 = \{T_1, T_3, T_6\}$ at node y_1 , where EATS=baked.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T_4, T_5, T_7, T_8\}$ at node y_2 , where EATS=fried.
- $Y(1,2)=2$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=4$ (number of patterns in V_2)
- $V_3 = \{T_2\}$ at node y_3 , where EATS=roasted.
- $Y(1,3)=1$ (number of patterns in V_3 of class P)
- $Y(2,3)=0$ (number of patterns in V_3 of class S)
- $Z(3)=1$ (number of patterns in V_3)

The ratio of information gain of FOOTWEAR



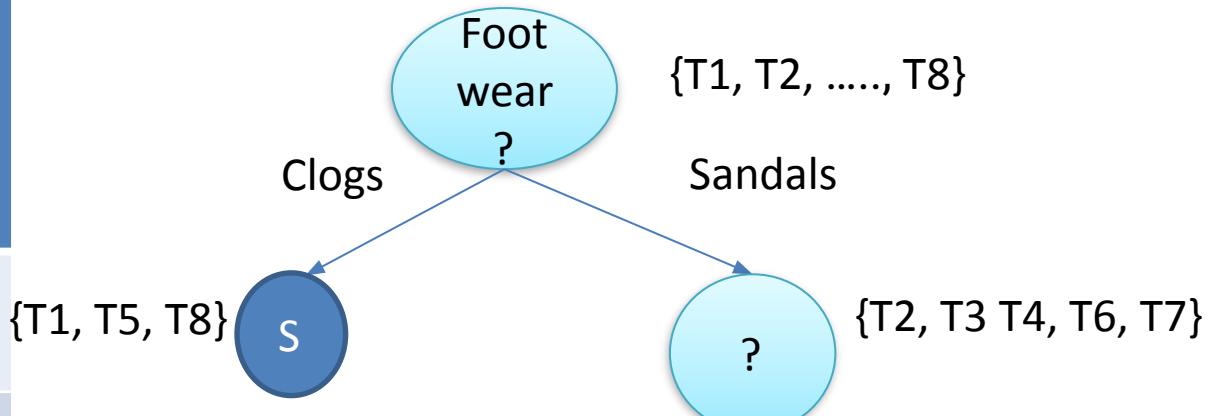
- $V_1 = \{T1, T5, T8\}$ at node y_1 , where FOOTWEAR=clogs.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T2, T3, T4, T6, T7\}$ at node y_2 , where FOOTWEAR=sandals.
- $Y(1,2)=3$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=5$ (number of patterns in V_2)

(.69 - 1.31 39 .09 1.29 49 .19 - .81 - .31 - .71)
(.49 - 1.21 .99 .29 1.09 .79 - .31 - .81 - .31 - 1.0)

$$S_{Footwear}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} (-\log \frac{Z(j)}{Z(0)}) \\ = 0.9544$$

$$G_{Footwear}(V_0) = \frac{I(V_0) - I_{Footwear}(V_0)}{S_{Footwear}(V_0)} \\ = 0.3640$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabb y	Baked	Clogs	S
T2	Gabb y	Roasted	Sandals	P
T3	Gabb y	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabb y	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabb y	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



$$\|x\| = \sqrt{2^2 + 4^2 + 0^2 + 0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} = 5.83$$

Pearson correlation coefficient: measures linear dependency between two random variables

$$\rho = \frac{\text{cov}(F1, F2)}{\sqrt{\text{var}(F1)\text{var}(F2)}}$$

$$\text{cov}(F1, F2) = \sum ((F1_i - F1') \cdot (F2_i - F2')) / (n - 1)$$

$$\text{var}(F1) = \sum (F1_i - F1')^2 / (n - 1), \text{ where } F1' = \frac{1}{n} \sum F1_i$$

$$\text{var}(F2) = \sum (F2_i - F2')^2 / (n - 1), \text{ where } F2' = \frac{1}{n} \sum F2_i$$

- Let C_1, C_2, \dots, C_m be the number of classes
- N denotes the number of training patterns of class C_k in the set V_t
- $Z(t)$ is the number of patterns in the set V_t
- $Z(t) = \sum_{k=1}^m Y(k, t)$
- The probability that a pattern in V_t belongs to class C_k is $\frac{Y(k, t)}{Z(t)}$
- The information required to classify a pattern in V_t into the class C_k for $1 \leq k \leq m$ is expressed as $-\log \frac{Y(k, t)}{Z(t)}$

The ratio of information gain of EATS at the right child of the root of a decision tree



- $V_1 = \{T3, T6\}$ at node y_1 , where EATS=baked.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=2$ (number of patterns in V_1 of class S)
- $Z(1)=2$ (number of patterns in V_1)

Fisher score

- Fisher score is one of the most widely used supervised feature selection methods.
 - It selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features.
 - The score of the i-th feature S_i will be calculated by Fisher Score,
- $$S_i = \frac{\sum n_j (\mu_{ij} - \mu_i)^2}{\sum n_j \sigma_{ij}^2}$$
- where μ_{ij} and σ_{ij} are the mean and the variance of the i-th feature in the j-th class, respectively.
 n_j is the number of instances in the j-th class and μ_i is the mean of the i-th feature.
- The features are ranked according to the Fisher Score.

$$\text{Slope} = \text{Rise/Run} = \frac{\Delta(Y)}{\Delta(X)}$$

$(X-X')$	$(Y-Y')$
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.0
$cov(X, Y)$	

$$5.539/9=0.6154444444$$

$$\text{Var}(X) =$$

$$5.549/9=0.6165555556$$

$$\text{Var}(Y) = \sum_{j=1}^n z(j) (-\log \frac{z(j)}{Z(0)})$$

$$6.4289/9=0.7143222222 \\ =1.5218$$

$$G_{Eats}(V_0) = \frac{I(V_0) - I_{Eats}(V_0)}{S_{Eats}(V_0)} \\ = 0.6377$$

The ratio of information gain of HABIT at the right child of the root of a decision tree

Step 6: Deriving the new data set

we simply take the transpose of the vector and multiply it on transpose
DataAdjust

Spearman's correlation coefficient: measures linear dependency between two random variables. It uses rank of each value. Data are represented as
 $\mathbf{x} = \mathbf{x}'^T$
 $\mathbf{y} = \mathbf{y}'^T$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

$$S(\mathbf{F1}, \mathbf{F2}) = \frac{\text{cov}(\mathbf{F1}, \mathbf{F2})}{\sqrt{\text{var}(\mathbf{F1}) \cdot \text{var}(\mathbf{F2})}}$$

$$\text{cov}(\mathbf{F1}, \mathbf{F2}) = \sum_i ((\mathbf{F1}_i'^T - \mathbf{F1}'^T) \cdot (\mathbf{F2}_i'^T - \mathbf{F2}'^T)) / (n-1)$$

$$\text{var}(\mathbf{F1}) = \sum_i (\mathbf{F1}_i'^T - \mathbf{F1}'^T)^2 / (n-1) \quad \text{where } \mathbf{F1}'^T = \frac{1}{n} \sum_i \mathbf{F1}_i'^T$$

$$\text{var}(\mathbf{F2}) = \sum_i (\mathbf{F2}_i'^T - \mathbf{F2}'^T)^2 / (n-1), \quad \text{where } \mathbf{F2}'^T = \frac{1}{n} \sum_i \mathbf{F2}_i'^T$$

It ranges between +1 and -1

Kendall's Tau: measures linear dependency between two random variables. It uses rank of each value. Kendall's Tau has smaller variability when using larger sample sizes. However, Spearman's measure is more computationally efficient, as Kendall's Tau is $O(n^2)$ and Spearman's correlation is $O(n \log(n))$.

Data are represented as

$$\mathbf{x} = \mathbf{x}'^T$$

$$\mathbf{y} = \mathbf{y}'^T$$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

Therefore,

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z(0)} (-\log \frac{Y(k,0)}{Z(0)})$$

$$= 0.9710$$

$$I_{Habit}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} (-\log \frac{Y(k,j)}{Z(j)})$$

$$= 0.9507$$

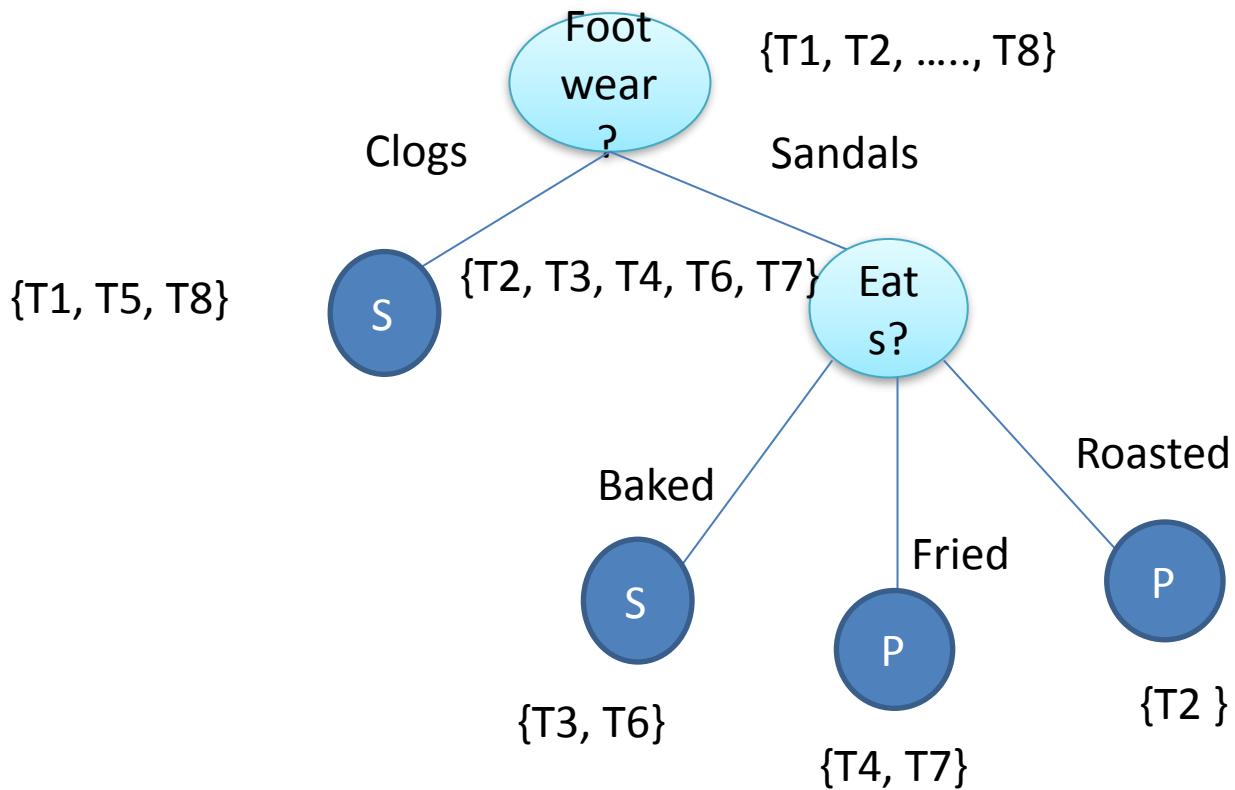
$$S_{Habit}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} (-\log \frac{Z(j)}{Z(0)})$$

$$= 0.9710$$

$$G_{Habit}(V_0) = \frac{I(V_0) - I_{Habit}(V_0)}{S_{Habit}(V_0)}$$

$$= 0.0208$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



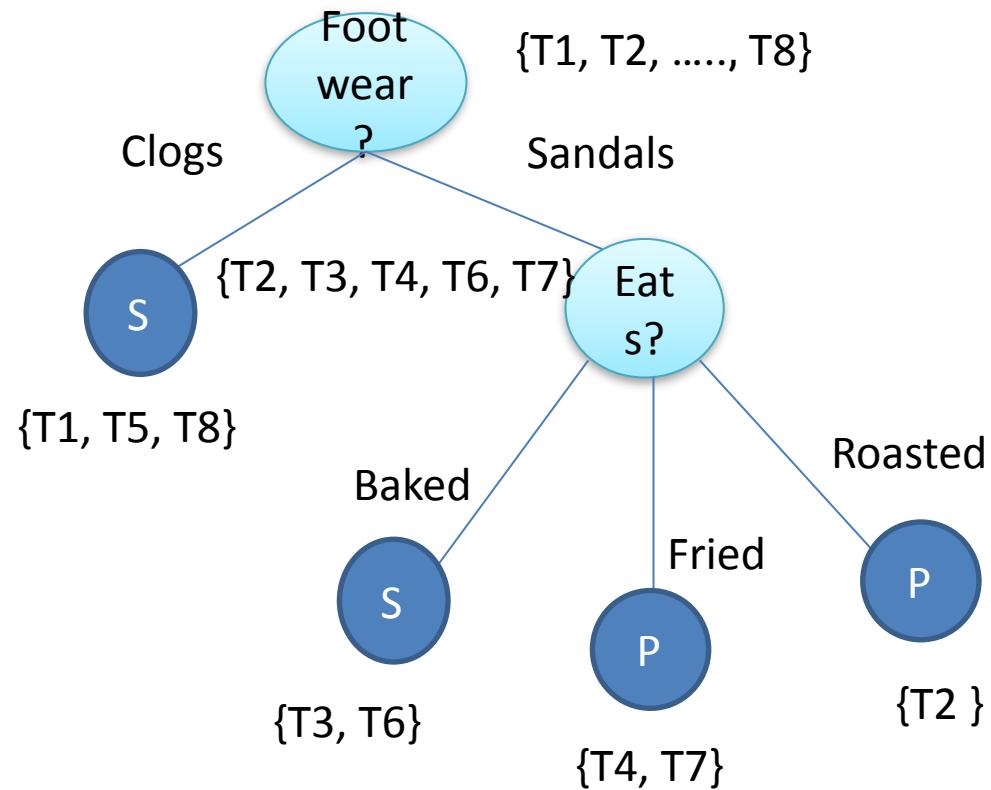
$$G_{Habit}(V_0) = 0.0208$$

If Footwear = Clogs
 Then pattern class= Student

If Footwear = Sandals
 and Eats = Baked,
 Then pattern class= Student

If Footwear = Sandals
 and Eats = Fried,
 Then pattern class= Professor

If Footwear = Sandals
 and Eats = Roasted,
 Then pattern class= Professor



Classifying the Recall patterns

- If Footwear = Clogs
Then pattern class= Student
- If Footwear = Sandals
and Eats = Baked,
Then pattern class= Student
- If Footwear = Sandals
and Eats = Fried,
Then pattern class= Professor
- If Footwear = Sandals
and Eats = Roasted,
Then pattern class= Professor

Recall	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor
R3	Gabby	Roasted	Clogs	Student
R4	Quiet	Roasted	Clogs	Student

Strength of DT

- It produces very simple understandable rules.
- Works well for most of the problem
- It can handle both numerical and categorical features
- It can work well for small and large training datasets
- DT shows which features are more useful for classification

Weakness of DT

DT is often biased towards features having more number of possible values

DT gets over-fitted and under-fitted easily.

DT is prone to errors with many classification and with small number of training examples.

DT is computationally expensive to train

Difficult to understand large DT.

Decision Tree with Missing Attributes

Values

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	Students
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Students
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	----	Students
T6	Quiet	Baked	Sandals	Students
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Students

Decision Tree with Missing Attributes Values

Let C_1, C_2, \dots, C_m be the number of classes where $m > 1$.

V_0 = a set at node y_0

We want to evaluate the ratio of information gain for attribute A at node y_0 but attribute A is having some missing values in V_0

A splits V_0 in to V_1, \dots, V_n as A is having n discrete values

V_{n+1} = is a set of patterns having missing values for A

For $1 \leq k \leq m$ and $0 \leq i \leq n$

$Y(k, i)$ be the number of training patterns of class C_k in the set V_i

For $0 \leq i \leq (n+1)$; $Z(i)$ is the number of patterns in the set V_i

Let ; $Z'(0) = Z(0) - Z(n+1)$

$Z'(0)$ = set of patterns for which the value of A is known

Decision Tree with Missing Attributes Values

The entropy of the set V_0

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z'(0)} \left(-\log \frac{Y(k,0)}{Z'(0)} \right)$$

The entropy of A for set V_0 .

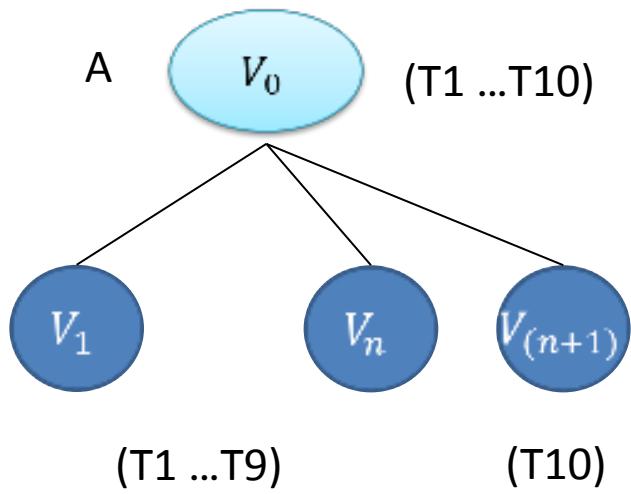
$$\bullet I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z'(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right)$$

Patterns of V_{n+1} are not included in the above calculation because A of this set does not provide any information about their class.

Information gain due to A is calculated

$$g_A(V_0) = \frac{Z'(0)}{Z(0)} [I(V_0) - I_A(V_0)]$$

$\frac{Z'(0)}{Z(0)}$ = the probability of the value of attribute A is known in the set V_0 .



Decision Tree with Missing Attributes Values

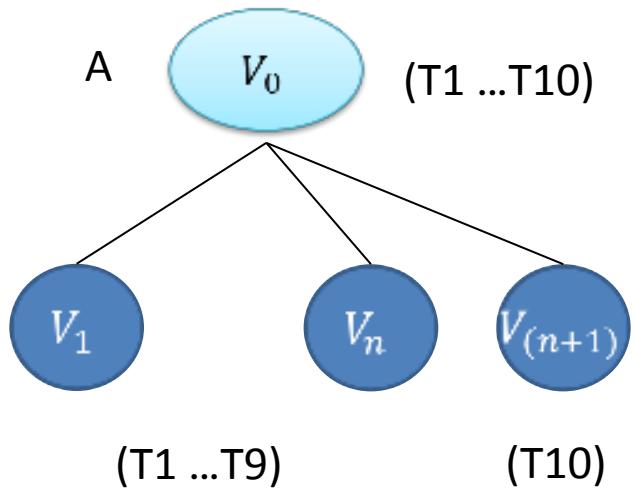
A has effectively split V_0
into $(n+1)$ sets

Therefore the split information of attribute A becomes

$$S_A(V_0) = \sum_{j=1}^{n+1} \frac{z(j)}{z(0)} \left(-\log \frac{z(j)}{z(0)} \right)$$

The ratio of information gain of attribute A for
set V_0 is as follows

$$G_A(V_0) = \frac{g_A(V_0)}{S_A(V_0)}$$



NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$G_{Habit}(V_0) = 0.0047$

$G_{Eats}(V_0) = 0.3233$

Ratio of information gain for Footwear

$V_0 = \{T1, T2, T3, \dots, T8\}$.

$Y(1,0) = 3$ (number of known Footwear-value patterns in patterns in V_0 of class P)

$Y(2,0) = 4$ (number of known Footwear-value patterns in V_0 of class S)

$Z(0) = 8$ (number of patterns in V_0)

$Z'(0) = 7$

$V_1 = \{T1, T8\}$ where Footwear= clogs

$Y(1,1) = 0$ (number of patterns in V_1 of class P)

$Y(2,1) = 2$ (number of patterns in V_1 of class S)

$Z(1) = 2$ (number of patterns in V_1)

$V_2 = \{T2, T3, T4, T6, T7\}$ where Footwear= sandals

$Y(1,2) = 3$ (number of patterns in V_2 of class P)

$Y(2,2) = 2$ (number of patterns in V_2 of class S)

$Z(2) = 5$ (number of patterns in V_2)

$V_3 = \{T5\}$ missing values of Footwear

$Z(3) = 1$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z'(0)} \left(-\log \frac{Y(k,0)}{Z(0)} \right) = 0.9852$$

$$I_{Footwear}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z'(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right) = 0.6936$$

$$g_{Footwear}(V_0) = \frac{Z'(0)}{Z(0)} [I(V_0) - I_A(V_0)] = 0.25515$$

$$S_{Footwear}(V_0) = \sum_{j=1}^{(n+1)} \frac{Z(j)}{Z(0)} \left(-\log \frac{Z(j)}{Z(0)} \right) = 1.3$$

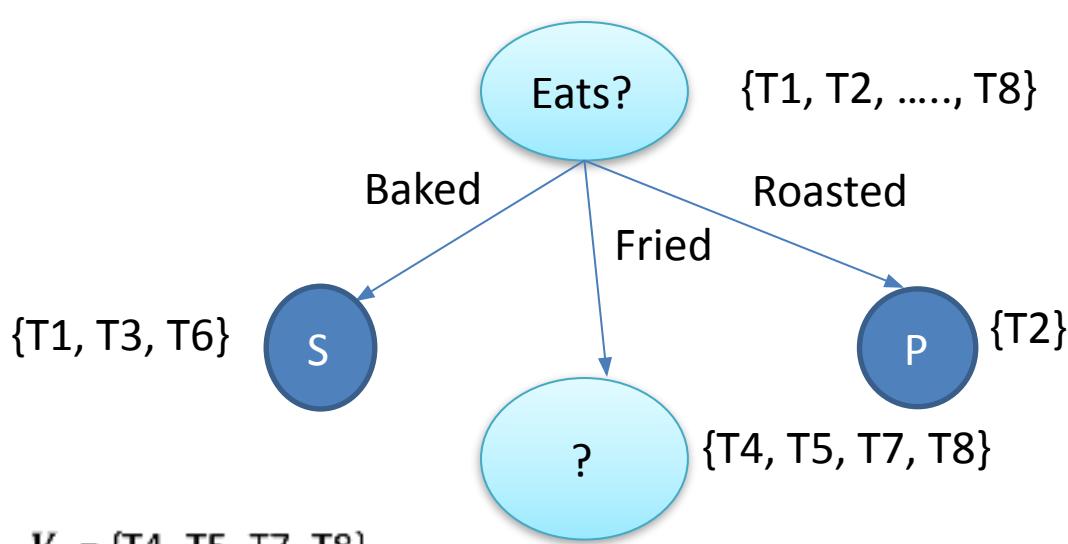
$$G_{Footwear}(V_0) = \frac{g_{Footwear}(V_0)}{S_{Footwear}(V_0)} = 0.1963$$

$$G_{Habit}(V_0) = 0.0047$$

$$G_{Eats}(V_0) = 0.3233$$

NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d		S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022



$$V_0 = \{T4, T5, T7, T8\}.$$

$Y(1,0)=2$ (number of known Footwear-value patterns in patterns in V_0 of class P)

$Y(2,0)=1$ (number of known Footwear-value patterns in V_0 of class S)

$Z(0)=4$ (number of patterns in V_0)

$$Z'(0)=3$$

$V_1 = \{T8\}$ where Footwear= clogs

$Y(1,1)=0$ (number of patterns in V_1 of class P)

$Y(2,1)=1$ (number of patterns in V_1 of class S)

$Z(1)=1$ (number of patterns in V_1)

$V_2 = \{T4, T7\}$ where Footwear= sandals

$Y(1,2)=2$ (number of patterns in V_2 of class P)

$Y(2,2)=0$ (number of patterns in V_2 of class S)

$Z(2)=2$ (number of patterns in V_2)

$V_3 = \{T5\}$ missing values of Footwear

$$Z(3)=1$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z'(0)} \left(-\log \frac{Y(k,0)}{Z(0)} \right) = 0.9183$$

$$I_{Footwear}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z'(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right) = 0$$

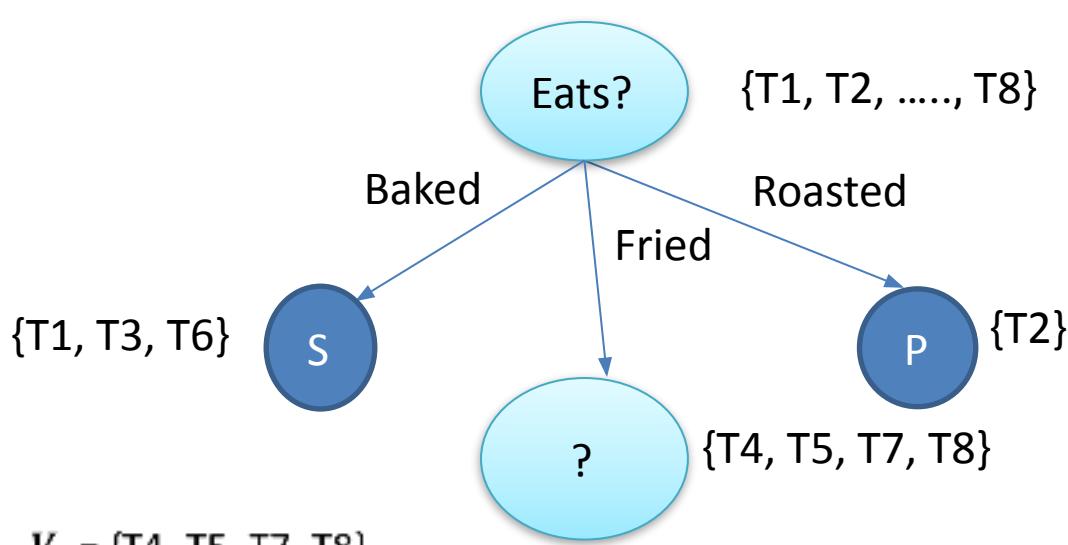
$$g_{Footwear}(V_0) = \frac{Z'(0)}{Z(0)} [I(V_0) - I_A(V_0)] = 0.9183$$

$$S_{Footwear}(V_0) = \sum_{j=1}^{(n+1)} \frac{Z(j)}{Z(0)} \left(-\log \frac{Z(j)}{Z(0)} \right) = 1.4466$$

$$G_{Footwear}(V_0) = \frac{g_{Footwear}(V_0)}{S_{Footwear}(V_0)} = 0.6351$$

NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

25-11-2022



$$V_0 = \{T4, T5, T7, T8\}$$

$Y(1,0)=2$ (number of patterns in V_0 of class P)

$Y(2,0)=2$ (number of patterns in V_0 of class S)

$Z(0)=4$ (number of patterns in V_0)

$$V_1 = \{T5, T7\} \text{ where Habit= gabby}$$

$Y(1,1)=1$ (number of patterns in V_1 of class P)

$Y(2,1)=1$ (number of patterns in V_1 of class S)

$Z(1)=2$ (number of patterns in V_1)

$$V_2 = \{T4, T8\} \text{ where Habit= quiet}$$

$Y(1,2)=1$ (number of patterns in V_2 of class P)

$Y(2,2)=1$ (number of patterns in V_2 of class S)

$Z(2)=2$ (number of patterns in V_2)

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Bake d	Clogs	S
T2	Gabby	Roast ed	Sandal s	P
T3	Gabby	Bake d	Sandal s	S
T4	Quiet	Fried	Sandal s	P
T5	Gabby	Fried		S
T6	Quiet	Bake d	Sandal s	S
T7	Gabby	Fried	Sandal s	P
T8	Quiet	Fried	Clogs	S

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z(0)} \left(-\log \frac{Y(k,0)}{Z(0)} \right) = 1.0$$

$$I_{Habit}(V_0) = \sum_{j=1}^n \frac{z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right) = 1.0$$

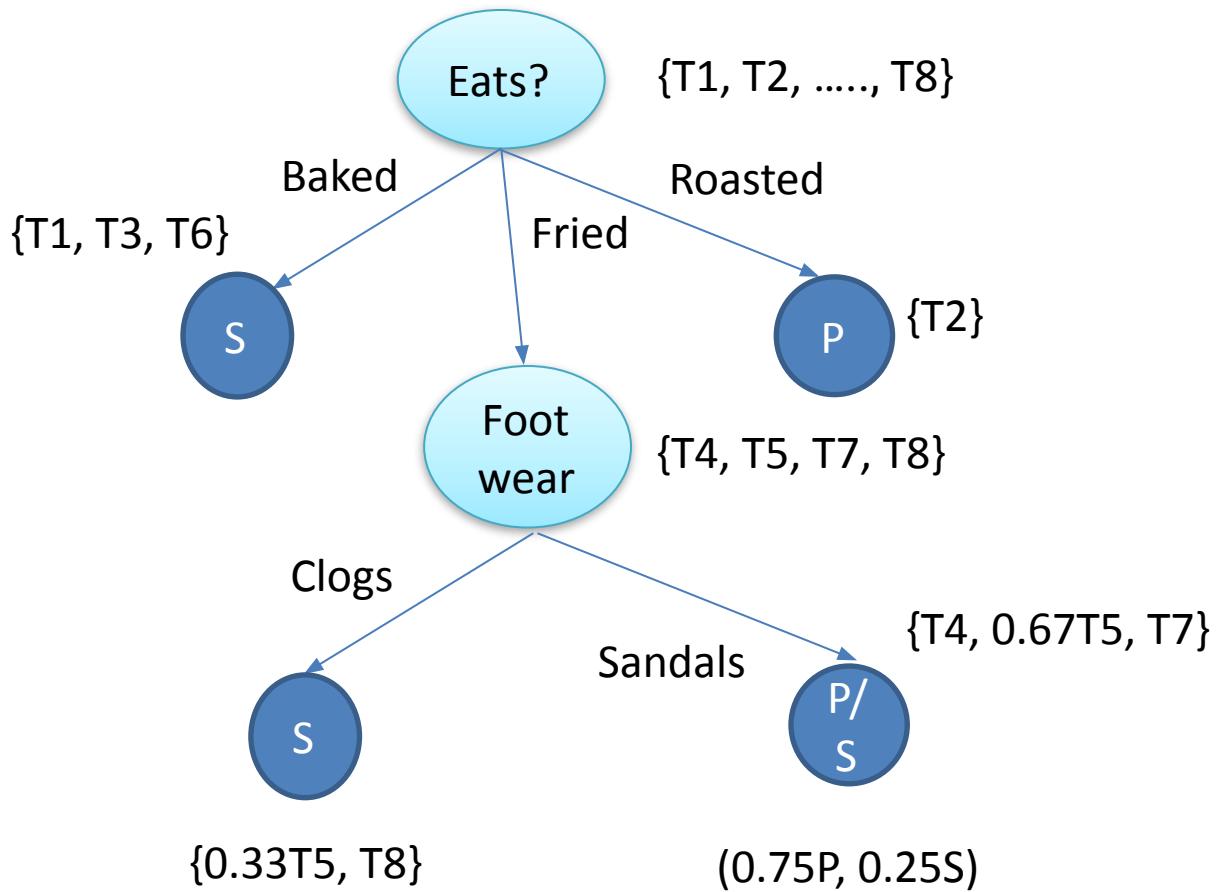
$$S_{Habit}(V_0) = \sum_{j=1}^{(n)} \frac{z(j)}{Z(0)} \left(-\log \frac{z(j)}{Z(0)} \right) = 1.0$$

$$G_{Habit}(V_0) = \frac{g_{Habit}(V_0)}{S_{Habit}(V_0)} = 0$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$$G_{Footwear}(V_0) = \frac{g_{Footwear}(V_0)}{S_{Footwear}(V_0)} = 0.6351$$

$$V_1 = \{T_2, T_3, T_4, T_6, T_7\}$$



Decision Tree with Missing Attributes Values

If some patterns are there with missing values of attribute A. So, in practise, a fraction of each such pattern is added to the set associated as follows

1. For $1 \leq i \leq n$, let w_i be equal to the sum of the weights of the training patterns in set V_i
2. The sum of the weights is calculated as $\text{sum_of_weight} = \sum_{i=1}^n w_i$
3. For every training pattern T in V_0 whose value of attribute A is missing do step 3.1 and 3.2
 - 3.1 Let f_0 be the fraction of pattern T present in V_0
 - 3.2 For $j = 1, 2, \dots, n$ do step 3.2.1.
 - 3.2.1 Add to set V_j the f_j th fraction of pattern T where

$$f_j = \frac{w_j}{\text{sum_of_weight}} f_0$$

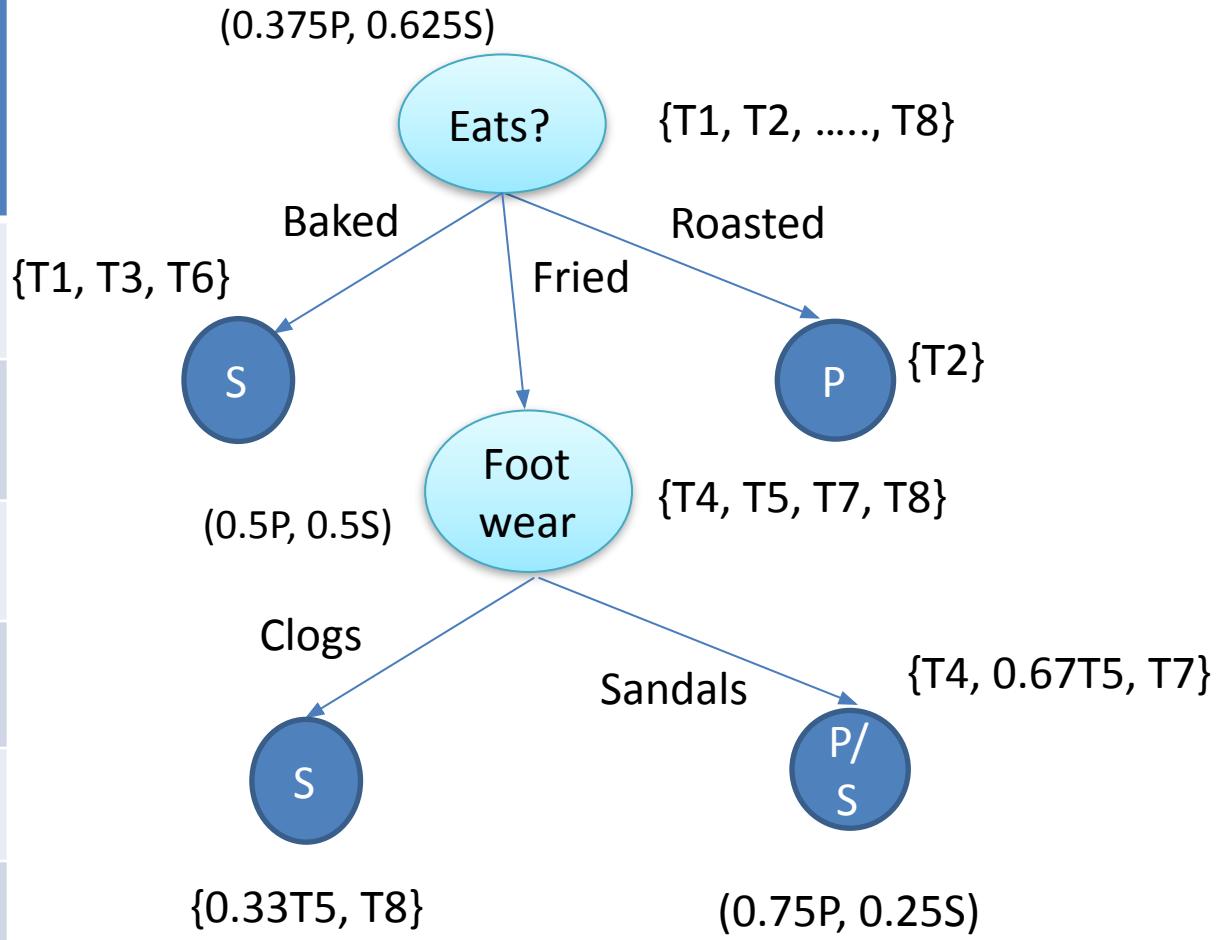
$$\text{T5 as clogs} = \frac{1}{3} * 1 = 0.33$$

$$\text{T5 as sandals} = \frac{2}{3} * 1 = 0.67$$

Out of 2.67, 2 is professor therefore professor = $2/2.67 = 0.75P$

Out of 2.67, 0.67 is student therefore student = $.67/2.67 = 0.25S$

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOT WEAR	
R1	Gabby	Fried	Clogs	S
R2	----	Fried	Sandals	0.75P, 0.25S
R3	-----	Fried	0.5P, 0.5S
R4		Baked		S
R5		Roasted		P
R6	Fried	Clogs	S
R7	Clogs	0.375P 0.625S
R8	Gabby	0.375P 0.625S



Decision Tree Algorithm

1. If $m = 1$ (that is, there is only one class), then create a single node x , label x with the name of the single class, and terminate the procedure. The decision tree consists of single node x , with its class label.
2. Initialize lists OPEN and CLOSED to empty.
3. Initialize subscript i to empty. Create a root node x_i , and associate the training set V_i with it.
4. Put node x_i in OPEN.
5. If OPEN is empty, return from the procedure. The nodes in CLOSED constitute the decision tree built. The subscript and the label of each node in CLOSED, together with the labelled arc from the node's parent, delineate the decision tree (step 10.1 explains how the arcs are labelled).
6. Remove the frontmost node x_i from OPEN. Create a candidate set of attributes for x_i , where the set contains all those attributes that have not been examined at any node on the path from the root to x_i . Select an attribute A from the candidate set for examining at node x_i (the criterion developed in Section 2.5 can be used to select attributes; for now, it may be selected arbitrarily).

Decision Tree Algorithm

7. If subscript i = empty (that is, x_i is the root), then put the following at the back of the list CLOSED: node x_i , together with the attribute A being examined at it.
8. If $i \neq$ empty (that is, x_i is not the root), then put the following at the back of the list CLOSED: node x_i , together with the attribute A being examined at it and the label of the arc from the parent-of- x_i to x_i .
9. Examine attribute A as follows. If A has n possible values v_1, v_2, \dots, v_n , then expand node x_i to generate its n children $x_{i1}, x_{i2}, \dots, x_{in}$.
10. For $j = 1, 2, \dots, n$ do steps 10.1 to 10.8 .
 - 10.1. Label the arc from node x_i to node x_{ij} with attribute value v_j .
 - 10.2. Associate with x_{ij} the set $V_{ij} \subseteq V_i$, such that the value of attribute A for the patterns in V_{ij} is v_j .
 - 10.3. If all the patterns in set V_{ij} belong to one class, then label node x_{ij} with the name of that class. Go to step 10.7.

Decision Tree Algorithm

10.4. If V_{ij} is empty, then label node x_{ij} with ‘?’ To indicate rejection , that is , failure to classify a given pattern. Go to step 10.7.(The question mark symbol, ‘?’ has been used to indicate rejection , which means that the question of pattern’s class still exists . You can replace it by a symbol of your choice .)

10.5. If the patterns in V_{ij} belong to more than one class and all the attributes have been examined on the path from the root to x_i , then label x_{ij} with ‘?’ To indicate rejection . (One could say that ideally we should have more attributes available to put the patterns of V_{ij} into separate classes , but more attributes may not be available. An alternative to rejection is to label node x_{ij} with the probability of the occurrence of different classes in V_{ij} ; more on this is given in Chapter 3.) Go to step 10.7.

10.6. Put the following at the back of the list OPEN: the node x_{ij} (it is a non-leaf node), and the label of the arc from node x_i to x_{ij} . Go to step 10.8.

10.7. Put the following at the back of the list CLOSED: node x_{ij} , a mark to indicate that x_{ij} is a leaf node, the label of x_{ij} , and the label of arc from x_i to x_{ij} .

10.8. Continue.

11. Go to step 5.

Decision Tree Pruning

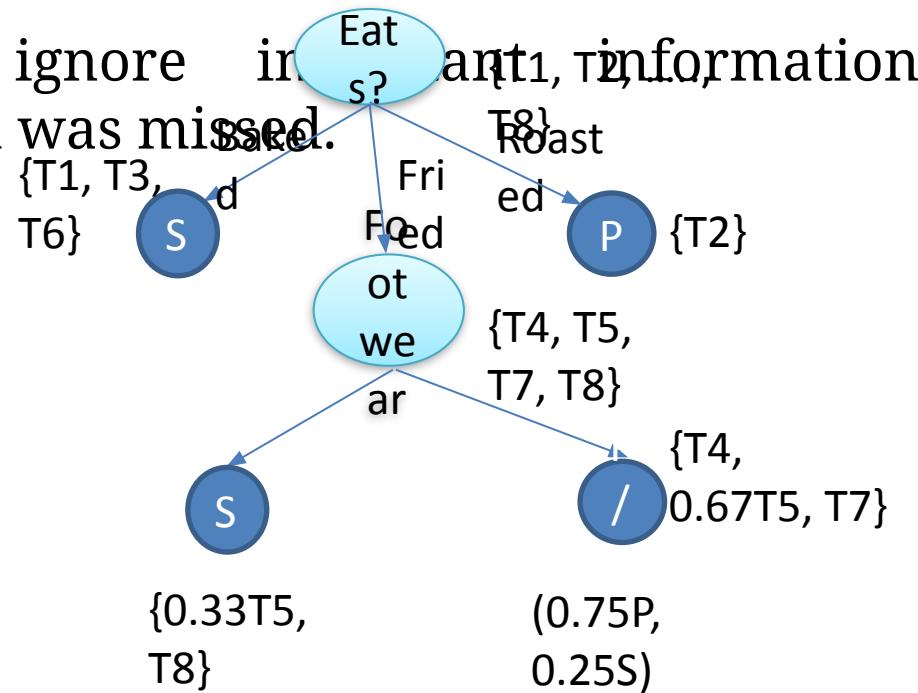
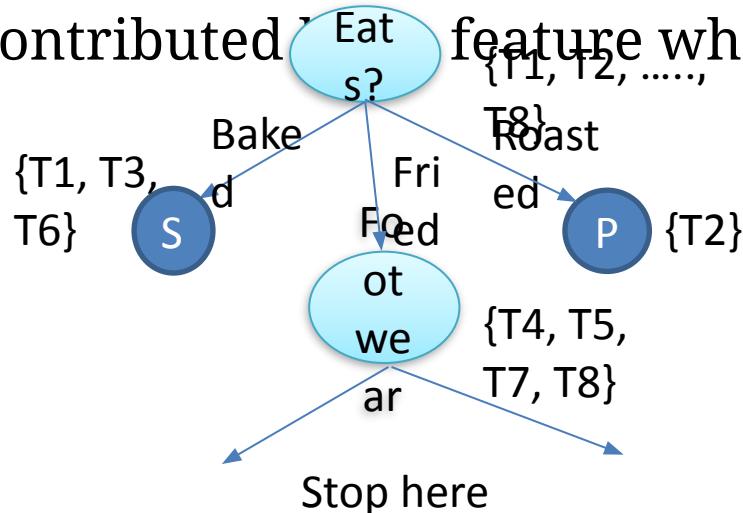
To prevent a DT getting overfitted to the training data, pruning of decision tree is essential.

Pruning a decision tree reduces the size of the tree such that the model is more generalized.

1. Pre-pruning: Stop growing the tree before it reaches perfection.

This avoids over-fitting as well as optimizes computational cost.

It stands a chance to ignore information contributed by a feature which was missed.



Decision Tree Pruning

To prevent a DT getting overfitted to the training data, pruning of decision tree is essential.

Pruning a decision tree reduces the size of the tree such that the model is more generalized.

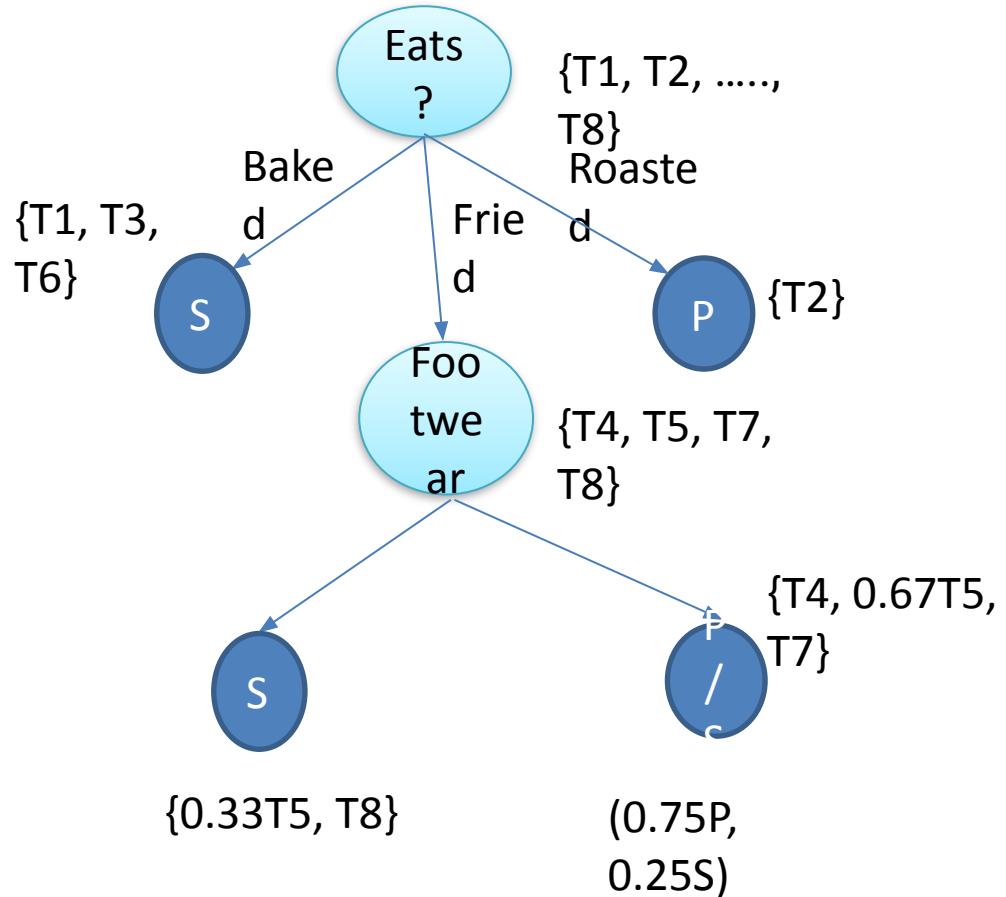
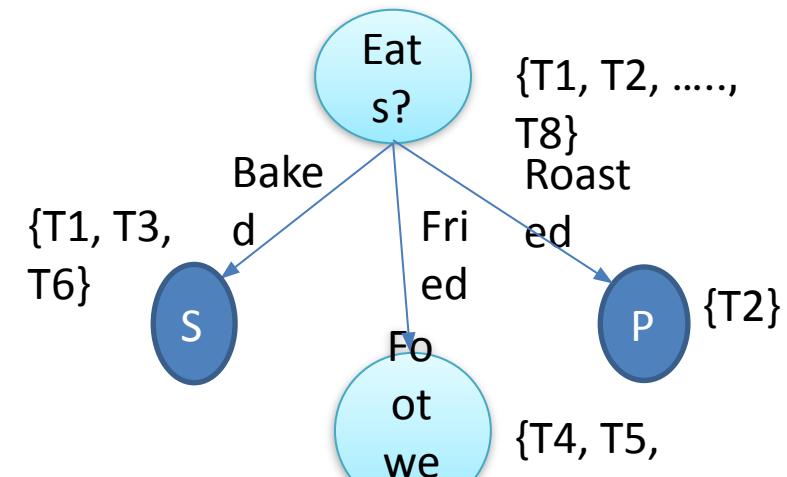
2. Post-Pruning: Allow the tree to grow entirely and then post-prune some of the branches from it.

By using certain pruning criterion, error rate, the size of the tree is reduced.

This is better for classification accuracy

Computational cost is more than pre-pruning

Decision Tree Pruning



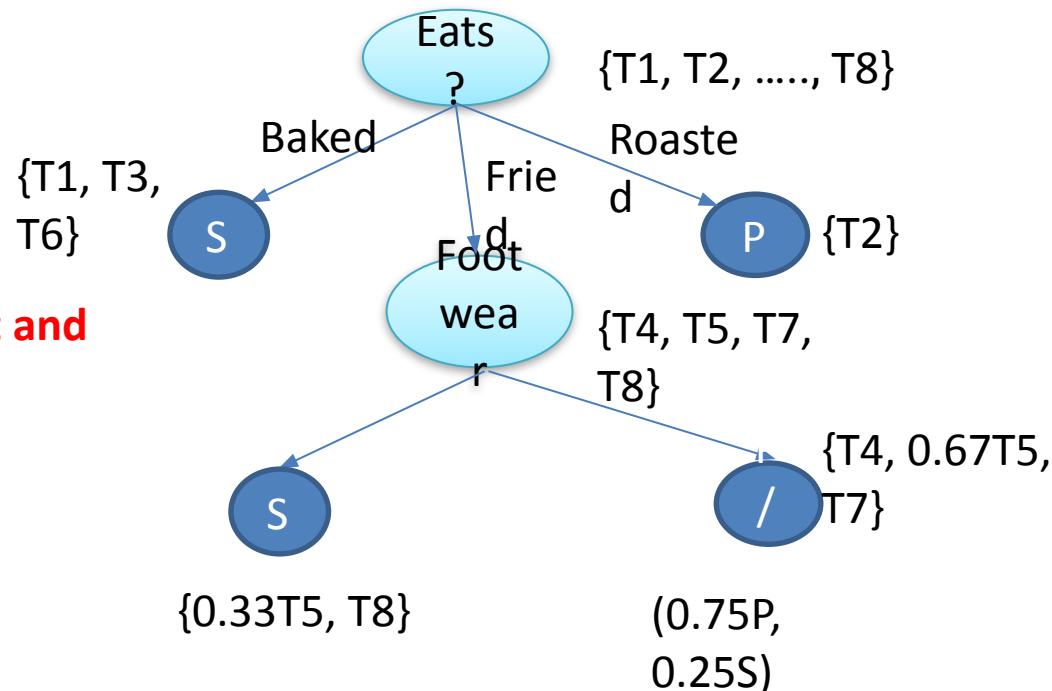
Error rate

Suppose class C is the most frequent in set V' associated with a node. A pattern at that node will be classified into C with error rate.

$$e(V') = \frac{|V'| - \text{no of patterns of class } C \text{ in } V'}{|V'|}$$

If $|V'| = 90$ and the no of patterns of class C in V' is 72

$$\text{Error rate} = (90-72)/90 = 0.2$$



For footwear (assume 3 student and 1 professor class)

$$(4-1)/4 = 0.25$$

Strength of DT

- It produces very simple understandable rules.
- Works well for most of the problem
- It can handle both numerical and categorical features
- It can work well for small and large training datasets
- DT shows which features are more useful for classification

Weakness of DT

DT is often biased towards features having more number of possible values

DT gets over-fitted and under-fitted easily.

DT is prone to errors with many classification and with small number of training examples.

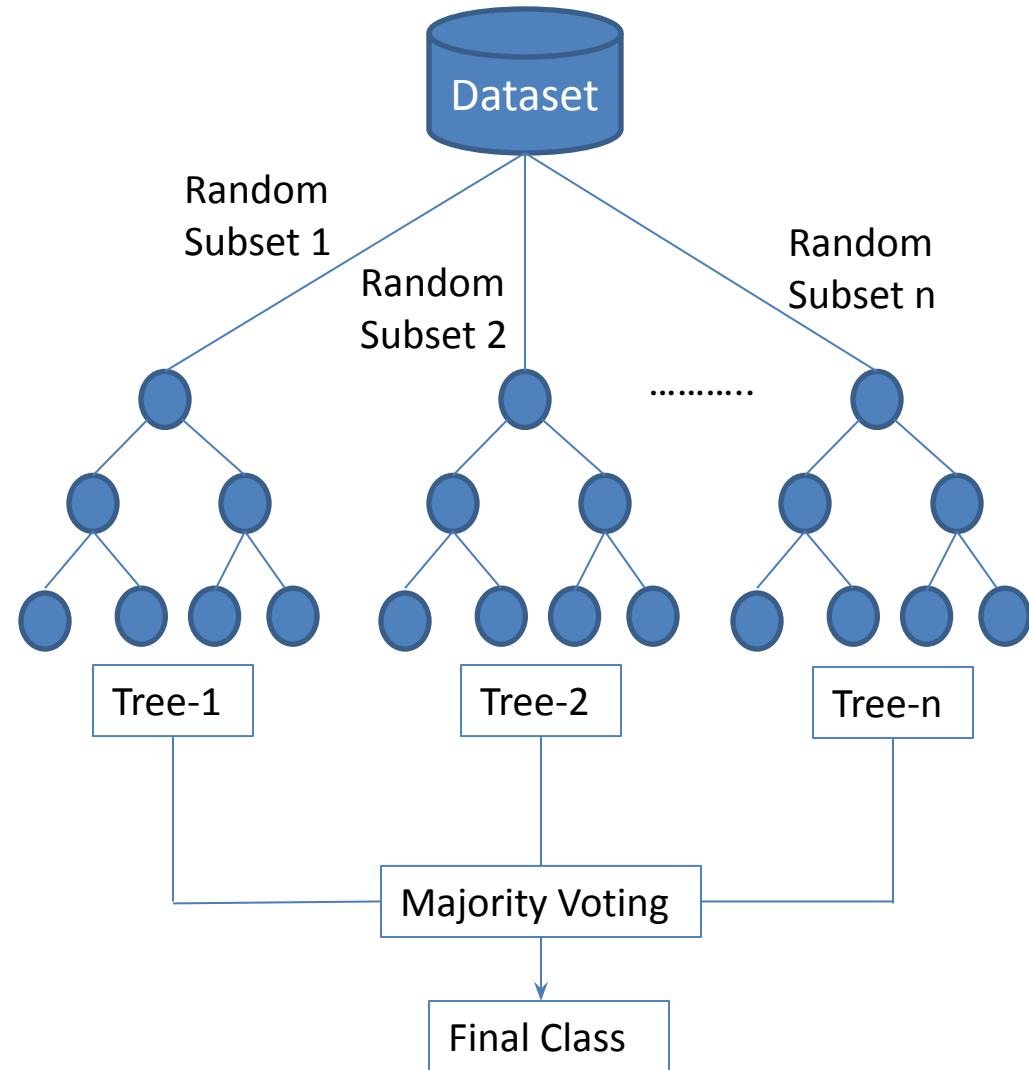
DT is computationally expensive to train

Difficult to understand large DT.

Random Forest (RF)

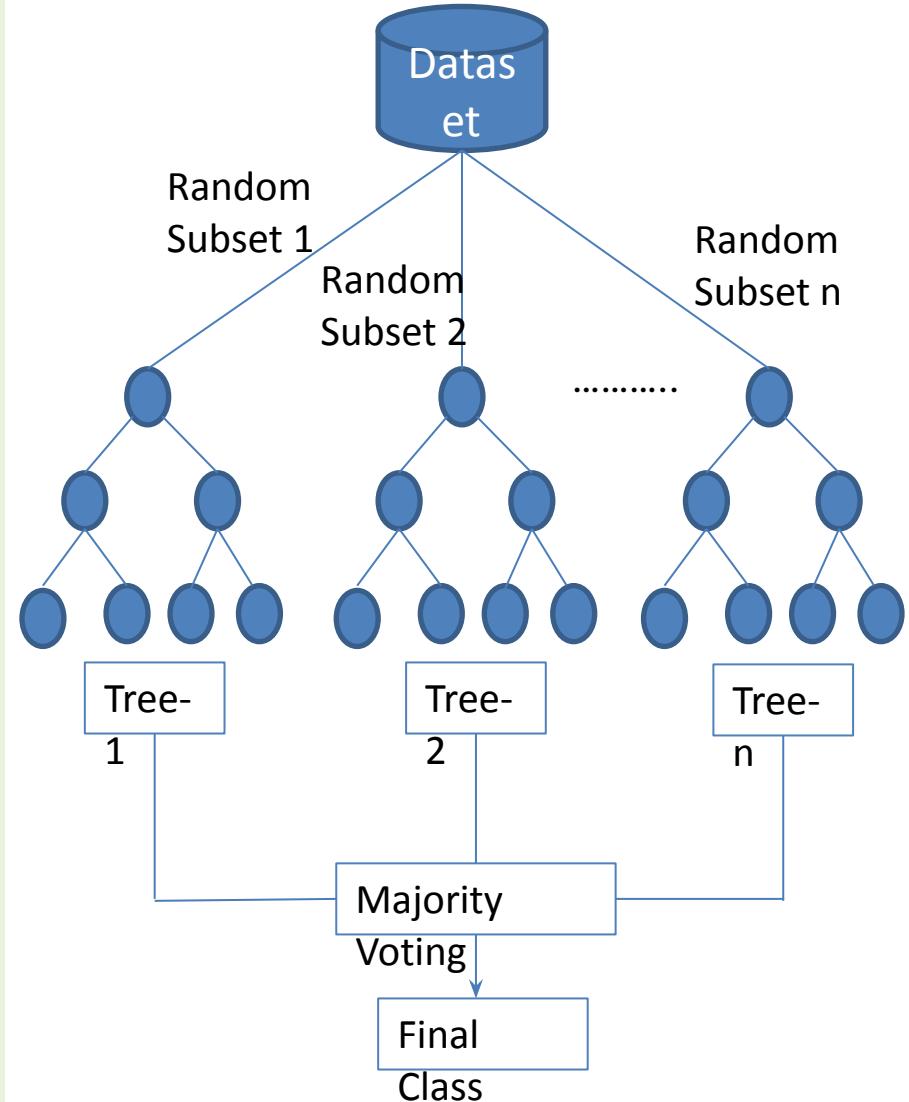
Large number of trees in RF is generated to train the trees enough such that contribution from each feature comes in a number of models.

Majority vote is applied to combine the output of the different trees,



Random Forest (RF) Algorithm

1. If there are N features, select a subset of m ($m < N$) features at random out of the N features. Also data instances should be picked randomly.
2. Use the best split principle on these m features to calculate the number of nodes d.
3. Keep splitting the nodes to child nodes till the tree is grown to the maximum possible extent.
4. Select a different subset of the training data with replacement to train another decision tree following steps 1 to 3. Repeat this to build and train 'n' decision trees.
5. Final class assignment is done on the basis of the majority votes from the 'n' trees.



Strength of RF

It runs efficiently on large datasets.

RF has a robust method for estimating missing data.

It can work well for imbalanced datasets as it has powerful techniques for balancing errors.

RF can be used to solve both classification and regression problem.

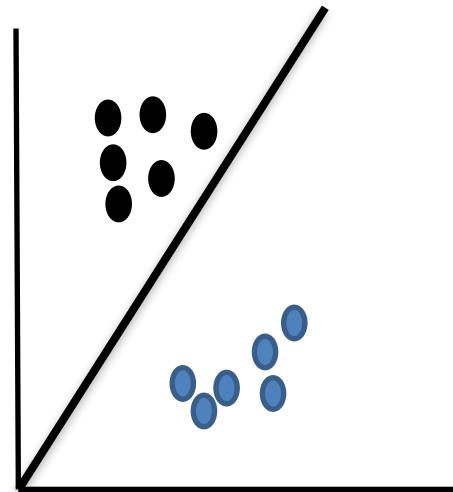
Weaknesses of RF

RF is not as easy to understand as a decision tree model.
It is computationally much more expensive than a simple model like DT.

Support Vector Machine (SVM)

SVM is a linear classification and regression model.

SVM is based on the concept of a surface called a hyperplane.



In a two dimensional space, the data instances belonging to different classes fall in different sides of a straight line.

If the same concept is extended to a multi-dimensional feature space, the straight line transforms to a hyperplane.

There may be many possible hyperplanes, and one of the challenges with the SVM model is to find the optimal hyperplane.

Support Vector Machine (SVM)

Support vectors are the data points, the critical component in a data set which are near the identified set of lines (hyperplanes)

Mathematically, in a two-dimensional feature space, the following hyperplane is created

$$c_0 + c_1 X_1 + c_2 X_2 = 0$$

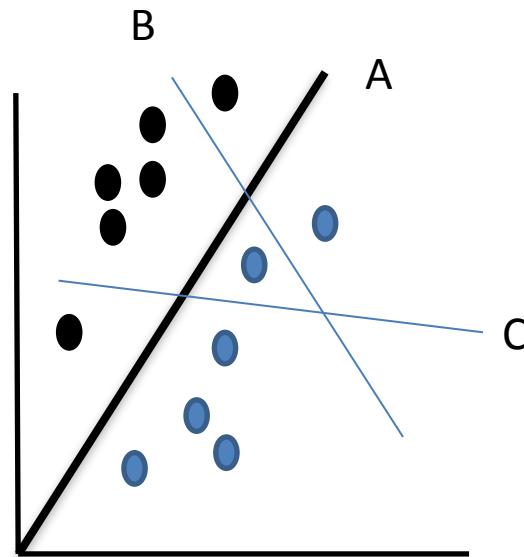
For N-dimensional space

$$c_0 + c_1 X_1 + c_2 X_2 + \dots + c_N X_N = 0$$

- The further from the hyperplane the data points lie, the more confident we can be about correct categorization.
- The distance between hyperplane and data points is known as margin.

Identifying the correct hyperplane in SVM

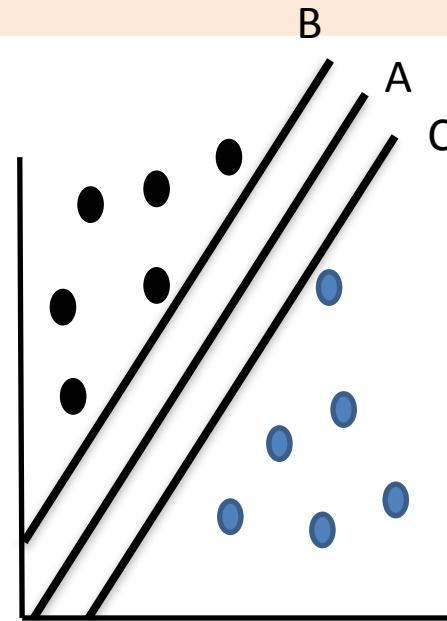
Out of A, B and C, A can perform the task nicely.



Identifying the correct hyperplane in SVM

- To identify the correct hyperplane, we need to identify the hyperplane which solves the problem in the best possible way.
- Here, maximizing the distances between the nearest data points of both the classes and hyperplanes will help us decide the correct hyperplane. This distance is called margin.

- A is the correct hyperplane.
- Hyperplane with higher margin makes it robust.
- If we select a hyperplane having a lower margin, then there is a high probability of misclassification.



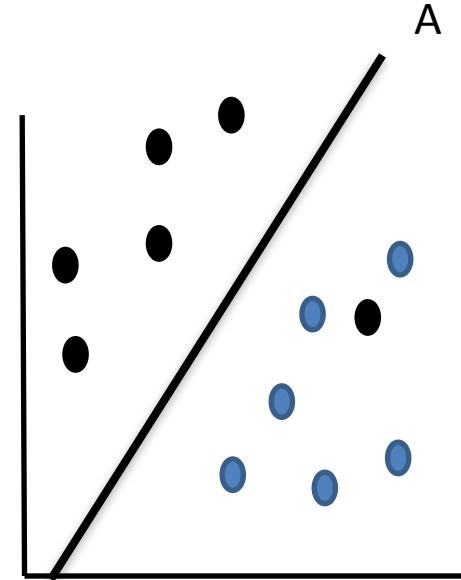
Identifying the correct hyperplane in SVM

SVM has a feature to ignore outliers and find the hyperplane that has the maximum margin.

Hence we can say that SVM is robust to outliers.

Observations:

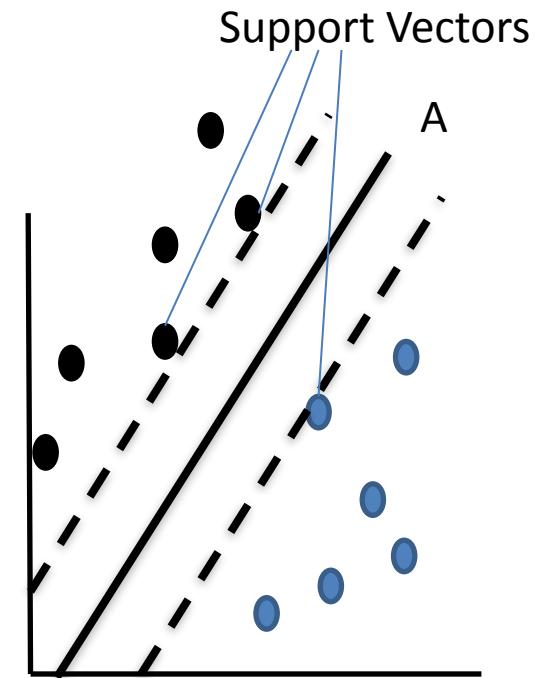
1. The hyperplane should segregate the data instances belonging to the two classes in the best possible way.
2. It should maximize the distances between the nearest data points of both the classes, means maximize the margin.
3. If there is a need to prioritize between higher margin and lesser misclassification, the hyperplane should try to reduce misclassifications.



Maximum Margin Hyperplane

Finding the Maximum Margin Hyperplane (MMH) is nothing but identifying the hyperplane which has the largest separation with the data instances of the two classes.

- It helps us in achieving more generalization and hence less number of issues in the classification of unknown data.
- There should be at least one support vector from each class.
- Modelling a problem using SVM requires identifying the support vectors and MMH corresponding to the problem space.



Identifying the MMH for linearly separable problem

Outer boundary needs to be drawn for the data instances belonging to the different classes which is known as convex hull.

The MMH can be drawn as the perpendicular bisect of the shortest line between the convex hulls.

A hyperplane in the N-dimensional features space can be given by the equation:

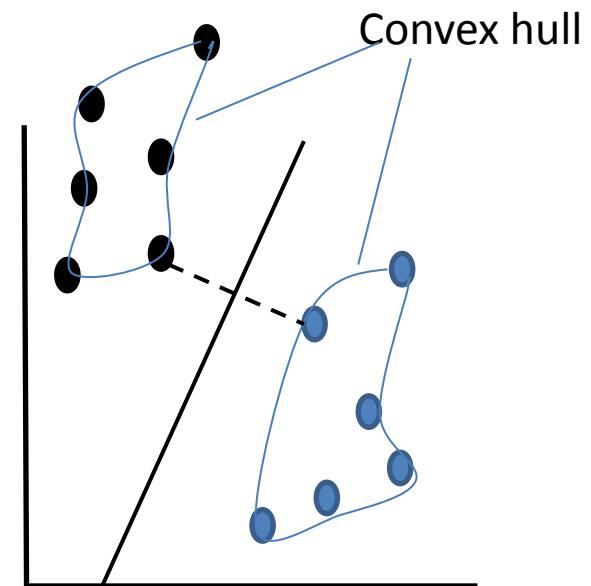
$$\vec{c} \cdot \vec{X} + c_0 = 0$$

Now the objective is to find a set of values for the vector \vec{c} such that two hyperplane can be specified:

$$\vec{c} \cdot \vec{X} + c_0 \geq +1$$

$$\vec{c} \cdot \vec{X} + c_0 \leq -1$$

It ensures that all data instances that belong to one class falls above one hyperplane and all the data instances belonging to the other class falls below another hyperplane.

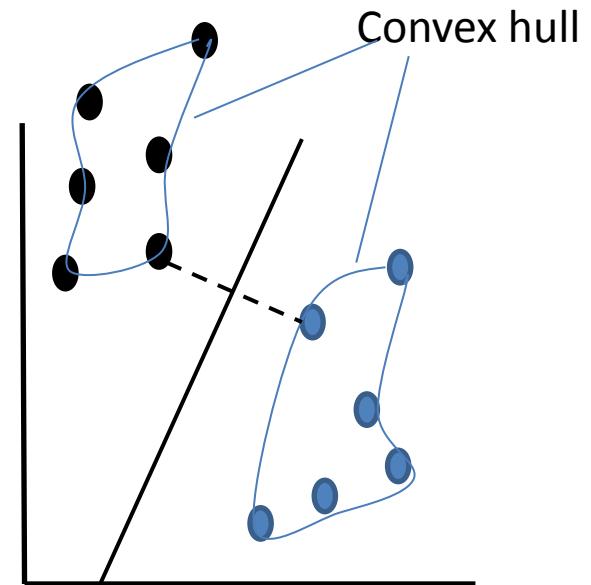


Identifying the MMH for linearly separable problem

The distance between these two planes should be $\frac{2}{\vec{c}}$

Now to maximize this distance, \vec{c} should be minimized.

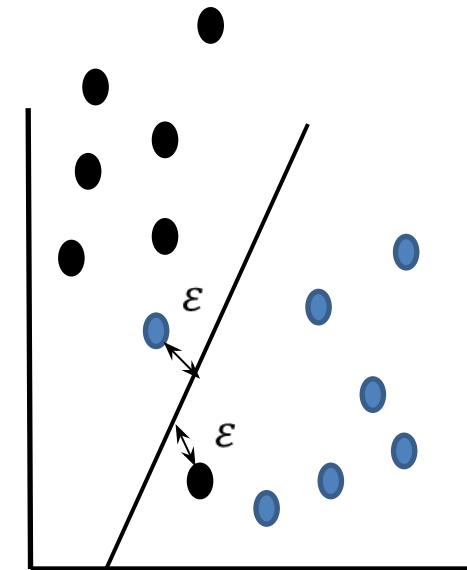
Hence the task of SVM is to solve the optimization problem $\min \left(\frac{1}{2} \vec{c}^2 \right)$



Identifying the MMH for non-linearly separable problem

- A soft margin (ε_i) is calculated for data instances in one class that fall on the wrong side of the hyperplane.
- A cost value C is imposed on all such data instances that fall on the wrong side of the hyperplane.
- The task of SVM is now to minimize the total cost which is given below:

$$\left(\min \left(\frac{1}{2} \vec{c} \right) + C \sum_{i=1}^N \varepsilon_i \right)$$



Kernel Trick

- Kernel trick is there to solve machine learning problems involving non-linearly separable datasets.
- Kernel function can transform lower dimensional input space to a higher dimensional space.
- In this process, SVM converts linearly non-separable data to a linearly separable data.

Some of the kernel functions which transform a lower dimension i to higher dimension j are given below:

Polynomial kernel: It is popular in image processing. d is the degree of the polynomial.

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$$

Gaussian kernel: It is a general-purpose kernel; used when there is no prior knowledge about the data.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

Kernel Trick

Gaussian Radial Basis Function (RBF): It is a general-purpose kernel; used when there is no prior knowledge about the data.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

$$\gamma > 0$$

Sometimes, $\gamma = 1/2\sigma^2$

Laplace RBF kernel: It is general-purpose kernel; used when there is no prior knowledge about the data.

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right)$$

Hyperbolic tangent kernel: We can use it in neural networks.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$$

Kernel Trick

Sigmoid kernel: We can use it as the proxy for neural networks.

$$k(x, y) = \tanh(\alpha x^T y + c)$$

Bessel function of the first kind Kernel: We can use it to remove the cross term in mathematical functions. j is the Bessel function of first kind.

$$k(x, y) = \frac{J_{v+1}(\sigma \|x - y\|)}{\|x - y\|^{-n(v+1)}}$$

ANOVA radial basis kernel: We can use it in regression problems

$$k(x, y) = \sum_{k=1}^n \exp(-\sigma(x^k - y^k)^2)^d$$

Kernel Trick

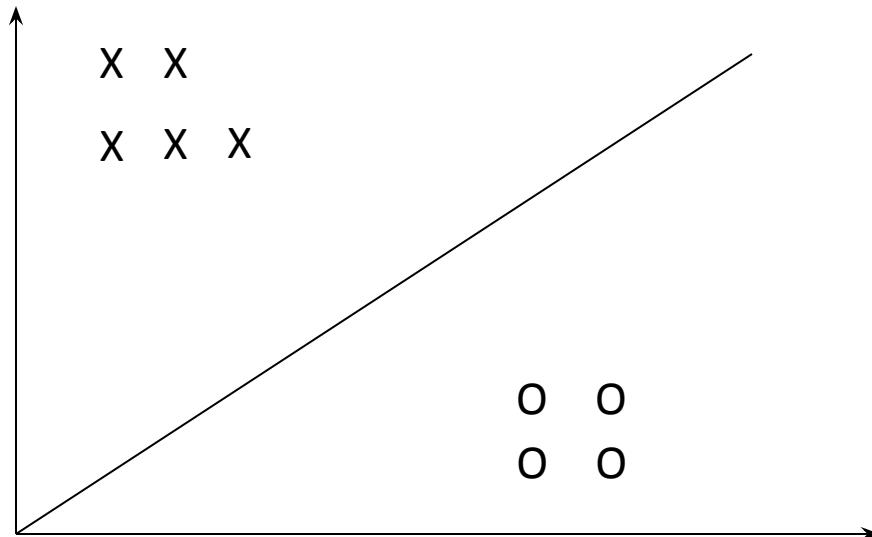
Linear splines kernel in one-dimension: It is useful when dealing with large sparse data vectors. It is often used in text categorization. The splines kernel also performs well in regression problems.

$$k(x, y) = 1 + xy + xy \min(x, y) - \frac{x+y}{2} \min(x, y)^2 + \frac{1}{3} \min(x, y)^3$$

Linear Discriminant Functions

Linear discriminant function can be used to discriminate between patterns belonging to two or more classes.

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O



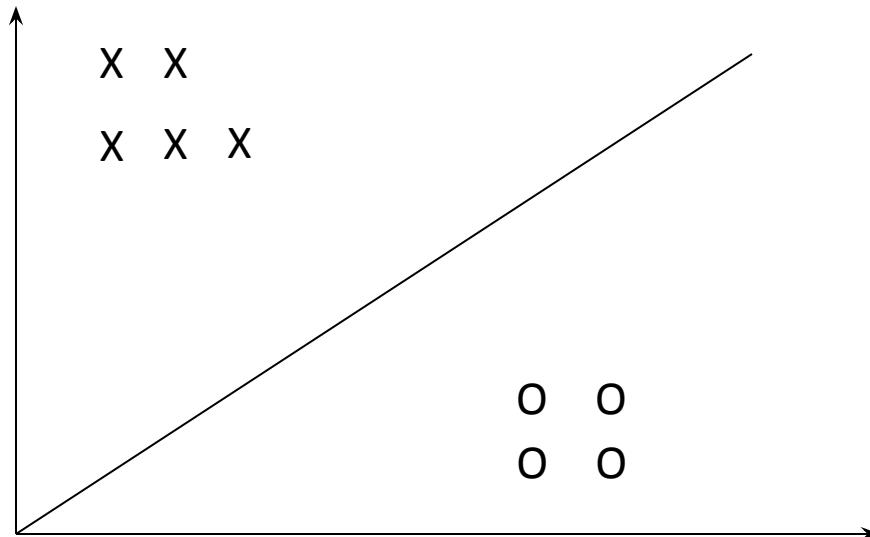
A line $x_1 = x_2$ can classify the patterns very easily.
All the patterns labelled X are to the left of the line and patterns labelled O are on the right side.

Another line $x_1 < x_2$ can solve the problem; if $x_1 - x_2$ is less than zero it belongs to class X. Pattern 1 ($0.5 - 3.0 = -2.5$)

Linear Discriminant Functions

Linear discriminant function can be used to discriminate between patterns belonging to two or more classes.

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O



If $x_1 - x_2 > 0$ it belongs to class O. For pattern 5 ($5.5 - 0.5 = 5.0$)

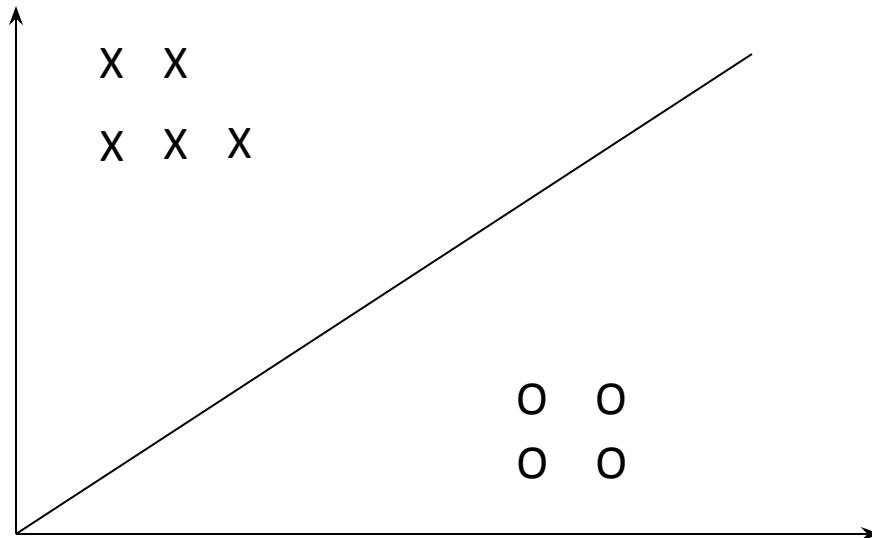
Based on the location of the patterns with respect to the line, we can say O are on the positive side and X are on the negative side.

There are possible infinite ways of realising the decision boundary, a line which can solve the problem.

Linear Discriminant Functions

Linear discriminant function can be used to discriminate between patterns belonging to two or more classes.

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O



$$x_1 - x_2 = 1, \text{ this line also can solve the problem}$$

$$x_1 - x_2 = 0$$

$$3x_1 - 2x_2 = 1$$

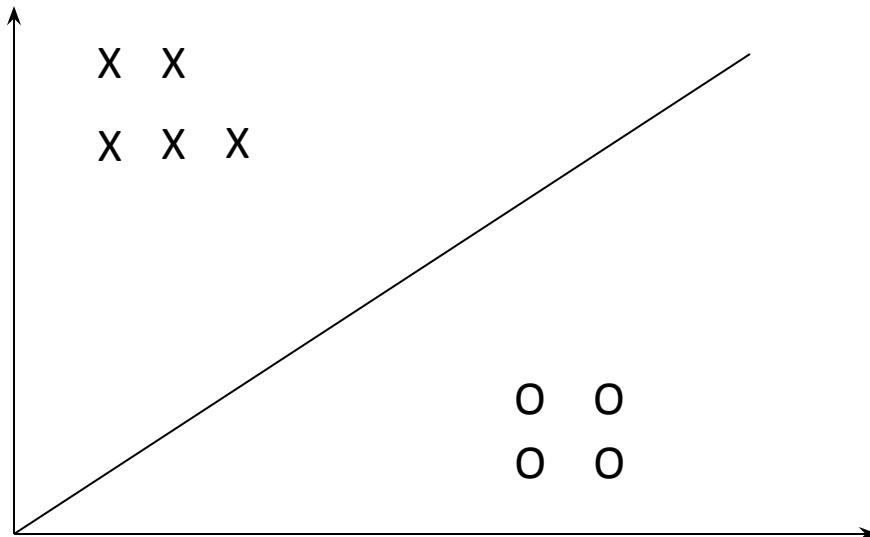
It is convenient to abstract all such lines using the following functional form.

$$f(x) = w_1 x_1 + w_2 x_2 + b = 0$$

Linear Discriminant Functions

Linear discriminant function can be used to discriminate between patterns belonging to two or more classes.

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O



$$f(x) = w_1x_1 + w_2x_2 + b = 0$$

$x_1 - x_2 = 0$ has $w_1=1$; $w_2=-1$ and $b = 0$

$x_1 - x_2 = 1$ has $w_1=1$; $w_2=-1$ and $b = -1$

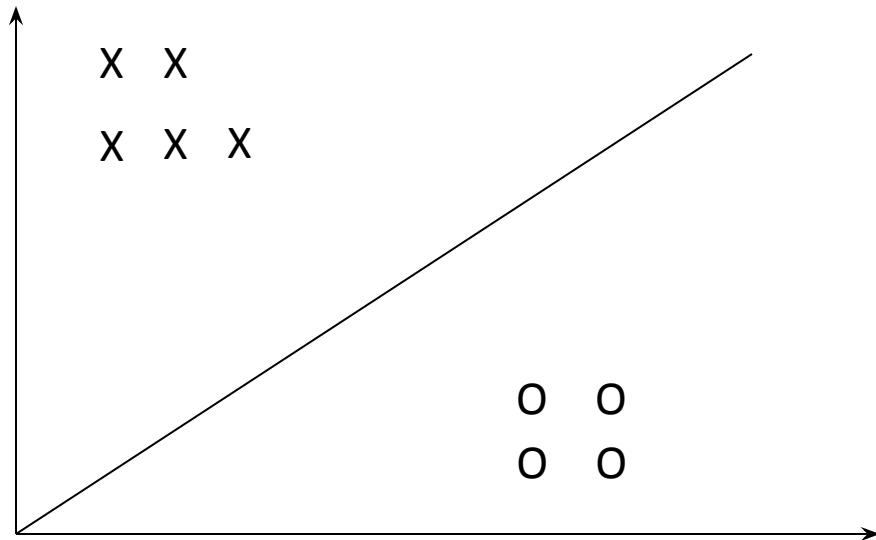
In a d-dimensional space, the decision boundary is a hyperplane and can be represented by

$$f(x) = w^t x + b = 0$$

Where w and x are d-dimensional vectors.

Linear Discriminant Functions

Linear discriminant function can be used to discriminate between patterns belonging to two or more classes.



$$f(x) = w^t x + b = 0$$

When $b=0$, it is homogeneous form of representation otherwise non-homogeneous form.

X and O are linearly separable if we can find a weight vector w and a scalar b such that

$w^t x + b > 0$ for all patterns x belonging to one class

$w^t x + b < 0$ for all patterns belonging to other class

Linear Discriminant Functions

The linear classifier based on vector $w^t = (1, -1)$ and $b = -1$

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O

Pattern no	1	2	class	
1	0.5	3.0	X	-3.5
2	1	3	X	-3.0
3	0.5	2.5	X	-3.0
4	1	2.5	X	-2.5
5	1.5	2.5	X	-2.0
6	4.5	1	O	2.5
7	5	1	O	3.0
8	4.5	0.5	O	3.0
9	5.5	0.5	O	4.0

$w^t x + b < 0$ for the X labelled patterns and $w^t x + b > 0$ for O labelled patterns.

The value of b plays a role in deciding the location of the decision boundary .

Decision boundary is characterized by $w^t x + b = 0$

Linear Discriminant Functions

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O

Pattern no	1	2	class	
1	0.5	3.0	X	-3.5
2	1	3	X	-3.0
3	0.5	2.5	X	-3.0
4	1	2.5	X	-2.5
5	1.5	2.5	X	-2.0
6	4.5	1	O	2.5
7	5	1	O	3.0
8	4.5	0.5	O	3.0
9	5.5	0.5	O	4.0

Following are the observations with respect to the value of b

1. When $b=0$; the origin lies in the decision boundary as $w^t x + b = 0$ for $x=0$ (origin) and $b=0$ for any value of w
2. When $b>0$; the origin lies in the positive side as $w^t x + b > 0$ for $x=0$ and $b>0$ for any value of w
3. When $b<0$; the origin lies in the negative side as $w^t x + b < 0$ for $x=0$ and $b<0$ for any value of w .

Linear Discriminant Functions

Distance between a vector and the decision boundary:

Consider $x_1 - x_2 = 0$ as the decision boundary $w = (1, -1)$

The cosine angle between w and a pattern x is $\frac{w^t x}{\|w\| \|x\|}$

Consider pattern 7: $\frac{1*5+(-1)*1}{\sqrt{1^2+(-1)^2} * \sqrt{5^2+(-1)^2}} = \frac{4}{\sqrt{2}\sqrt{26}} = \text{positive}$

This means w is orthogonal to the decision boundary and points to positive half space.

Distance between a point x and the decision boundary .

Any point x can be written as a sum of two vectors: one vector along the decision boundary and another orthogonal to it.

$$\text{So, } x = x_b + x_o$$

x_b = projection of x along the decision boundary and x_o = orthogonal component

We know w is orthogonal to the decision boundary

$x_o = p \frac{w}{\|w\|}$ p is a real number and is positive if x is from class O and negative if it is from class X.

$$f(x) = w^t x + b = w^t x_b + b + w^t x_o = 0 + w^t x_o = p w^t \frac{w}{\|w\|} = p \|w\|$$

Pattern no	1	2	class
1	0.5	3.0	X
2	1	3	X
3	0.5	2.5	X
4	1	2.5	X
5	1.5	2.5	X
6	4.5	1	O
7	5	1	O
8	4.5	0.5	O
9	5.5	0.5	O

Linear Discriminant Functions

$$f(x) = w^t x + b = w^t x_b + b + w^t x_o = 0 + w^t x_o = p w^t \frac{w}{|w|} = p ||w||$$
$$p = \frac{w^t x + b}{||w||}$$

P is the shortest distance between x and the decision boundary.

If the decision boundary is $x_1 - x_2 - 1 = 0$; $||w||^2 = 2$ and $b=-1$

$$\text{Distance from } x=(0, 0); p = \frac{w^t x + b}{||w||} = \frac{0*1+0*(-1)+(-1)}{\sqrt{2}} = \frac{-1}{\sqrt{2}}$$

$$\text{Distance from } x=(1, 1); p = \frac{w^t x + b}{||w||} = \frac{1*1+1*(-1)+(-1)}{\sqrt{2}} = \frac{-1}{\sqrt{2}}$$

SVM

Here $(1, 1)$ and $(2, 2)$ are from class X and $(2, 0)$ is from class O.

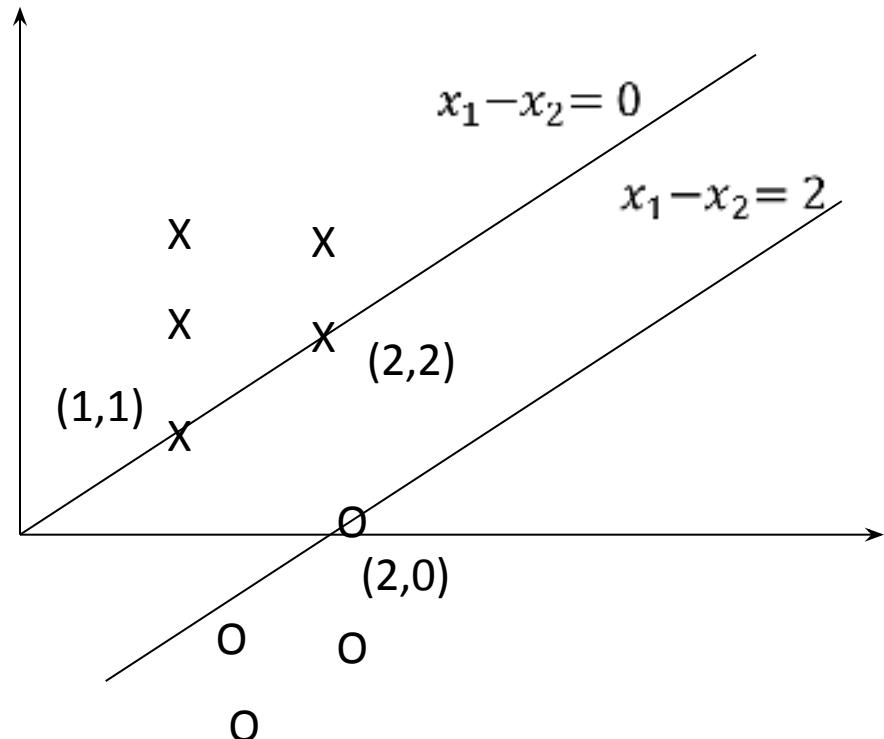
$$x_1 - x_2 = 0 \quad \text{and} \quad x_1 - x_2 = 2$$

Boundaries of the classes X and O respectively.

These lines are called support lines and points are called support vectors.

Any point of class X satisfying the property $x_1 - x_2 < 0$ is correctly classified.

Any patterns from O satisfying the property $x_1 - x_2 > 2$ is also correctly classified



The line $x_1 - x_2 = 1$ is equidistant from the two decision lines and right choice of decision boundary.

Points $x_1 - x_2 < 1$ are classified as members of class X and $x_1 - x_2 > 1$ are of class O.

SVM

The following observations are drawn

1. SVM may be viewed as a binary classifier
2. Support vectors are some of the vectors falling on the support planes in a d-dimensional space.
3. SVMs learn the linear discriminant that maximises the margin
4. When the two classes are linearly separable, it is easy to perceive and compute the maximum margin.

SVM: Linearly Separable case

Several lines can be drawn which can separate two classes.

SVM chooses the line which provides maximum margin as a decision boundary.

Two lines: $x_1 - x_2 = 0$ and $x_1 - x_2 = 2$ characterise the margin.

The decision boundary which is equidistant from these two lines is $x_1 - x_2 = 1$

This decision boundary can be observed as a linear classifier of the form $f(x) = w^t x + b$

(1,0) and (2,1) are on the decision boundary

So,

$$w_1 + b = 0 \quad (\text{point } (1, 0))$$

$$2w_1 + w_2 + b = 0 \quad (\text{point } (2, 1))$$

From these two equations $w_1 + w_2 = 0$; w is of the form $(a, -a)$ and correspondingly $b = -a$
 a is any constant

$w^t x + b = 0$ for all the points on the decision boundary

$w^t x + b = 1$ for all the points on the support line $x_1 - x_2 = 2$

$w^t x + b = -1$ for all the points on the support line $x_1 - x_2 = 0$

This is achieved when $a = 1$

SVM: Linearly Separable case

So, correspondingly $w = (1, -1)$ and $b = -1$

Normal distance between $w^t x + b = 1$ and $w^t x + b = 0$ is $\frac{w^t x + b}{\|w\|}$

$w^t x + b = 0$ is the decision boundary and is $x_1 - x_2 - 1 = 0$

X on $w^t x + b = 1$ when $x = (2, 0)$

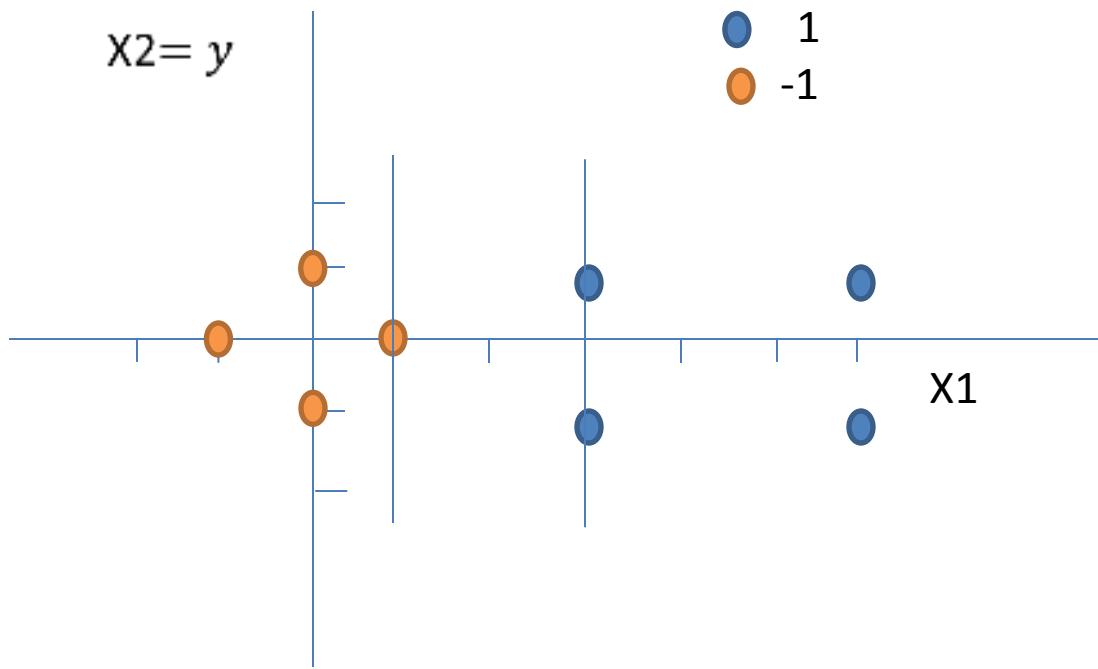
$$\text{Therefore } \frac{w^t x + b}{\|w\|} = \frac{(1*2) + (-1*0) - 1}{\|w\|} = \frac{(2) + 0 - 1}{\|w\|} = \frac{1}{\|w\|}$$

Similarly distance between $w^t x + b = -1$ and $w^t x + b = 0$ is $\frac{1}{\|w\|}$

The distance between two support line is $\frac{2}{\|w\|}$

This called margin. Now to maximize margin, we have to minimize $\|w\|$

Numeric Example



Pattern no	X1	X2	class
1	3	1	1
2	3	-1	1
3	6	1	1
4	6	-1	1
5	1	0	-1
6	0	1	-1
7	0	-1	-1
8	-1	0	-1

Three support vectors:

$$S_1 = (1, 0)$$

$$S_2 = (3, 1)$$

$$S_3 = (3, -1)$$

Numeric Example

Three support vectors:

$$S1 = (1, 0)$$

$$S2 = (3, 1)$$

$$S3 = (3, -1)$$

Vectors are augmented from d dimension to (d+1)th dimension

(d+1)th dimension is bias and is represented as 1 (This is called data transformation)

$$S1' = (1, 0, 1)$$

$$S2' = (3, 1, 1)$$

$$S3' = (3, -1, 1)$$

To calculate three weights following equations are established:

$$\alpha_1 S1'.S1' + \alpha_2 S1'.S2' + \alpha_3 S1'.S3' = -1$$

$$\alpha_1 S2'.S1' + \alpha_2 S2'.S2' + \alpha_3 S2'.S3' = 1$$

$$\alpha_1 S3'.S1' + \alpha_2 S3'.S2' + \alpha_3 S3'.S3' = 1$$

Numeric Example

To calculate three weights following equations are established:

$$\alpha_1 S1'.S1' + \alpha_2 S1'.S2' + \alpha_3 S1'.S3' = -1$$

$$\alpha_1 S2'.S1' + \alpha_2 S2'.S2' + \alpha_3 S2'.S3' = 1$$

$$\alpha_1 S3'.S1' + \alpha_2 S3'.S2' + \alpha_3 S3'.S3' = 1$$

$$\alpha_1(1, 0, 1).(1, 0, 1) + \alpha_2(1, 0, 1).(3, 1, 1) + \alpha_3(1, 0, 1).(3, -1, 1) = -1$$

$$\alpha_1(1+0+1) + \alpha_2(3+0+1) + \alpha_3(3-0+1) = -1$$

$$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$$

$$\alpha_1(3, 1, 1).(1, 0, 1) + \alpha_2(3, 1, 1).(3, 1, 1) + \alpha_3(3, 1, 1).(3, -1, 1) = 1$$

$$\alpha_1(3+0+1) + \alpha_2(9+1+1) + \alpha_3(9-1+1) = 1$$

$$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$$

$$\alpha_1(3, -1, 1).(1, 0, 1) + \alpha_2(3, -1, 1).(3, 1, 1) + \alpha_3(3, -1, 1).(3, -1, 1) = 1$$

$$\alpha_1(3-0+1) + \alpha_2(9-1+1) + \alpha_3(9+1+1) = 1$$

$$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$$

Numeric Example

$$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$$

$$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$$

$$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$$

$$\alpha_1 = -3.5; \quad \alpha_2 = 0.75; \quad \alpha_3 = 0.75$$

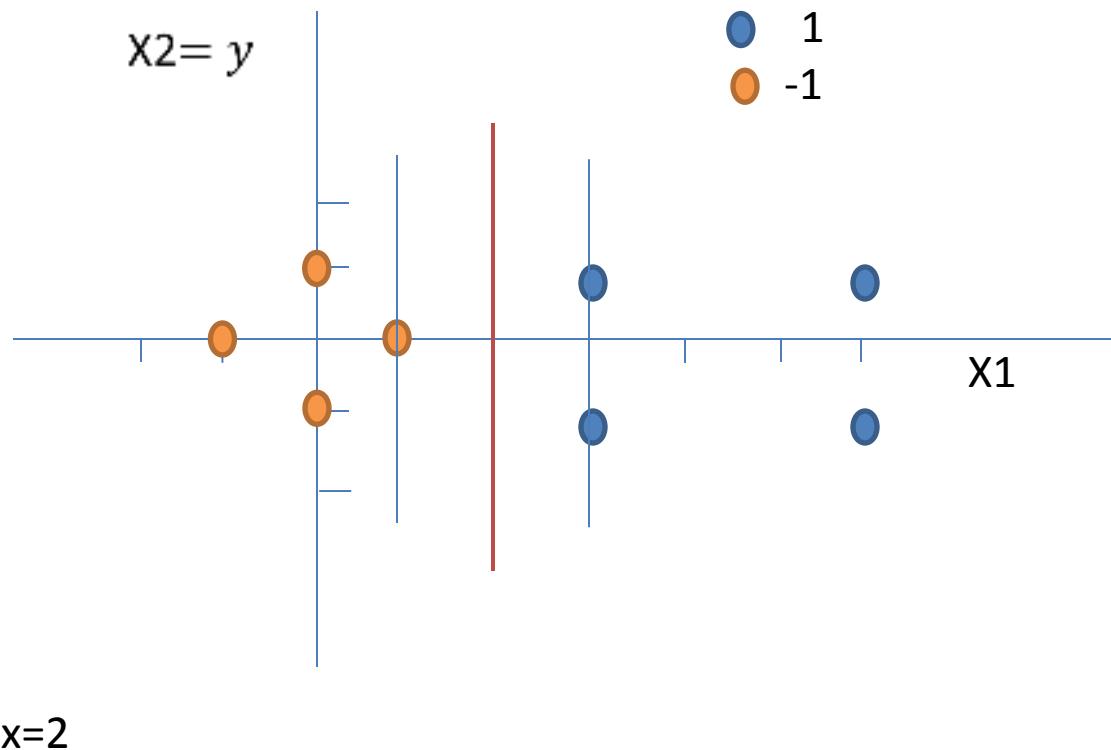
$$\begin{aligned} w' &= \sum_i \alpha_i S'_i \\ &= -3.5 (1 \ 0 \ 1)^t + 0.75 (3 \ 1 \ 1)^t + 0.75 (3 \ -1 \ 1)^t \\ &= (-3.5 \ 0 \ -3.5)^t + (2.25 \ 0.75 \ 0.75)^t + (2.25 \ -0.75 \ 0.75)^t \\ &= (1 \ 0 \ -2)^t \end{aligned}$$

$$\begin{aligned} w &= (1 \ 0)^t \\ b &= -2 \end{aligned}$$

$$(c_0 + c_1 X_1 + c_2 X_2 + \dots + c_N X_N = 0)$$

$$\begin{aligned} w1x1 + w2x2 + c &= 0; \\ 1. \times 1 + 0. \times 2 + (-2) &= 0; \\ x1 &= 2 \end{aligned}$$

Numeric Example

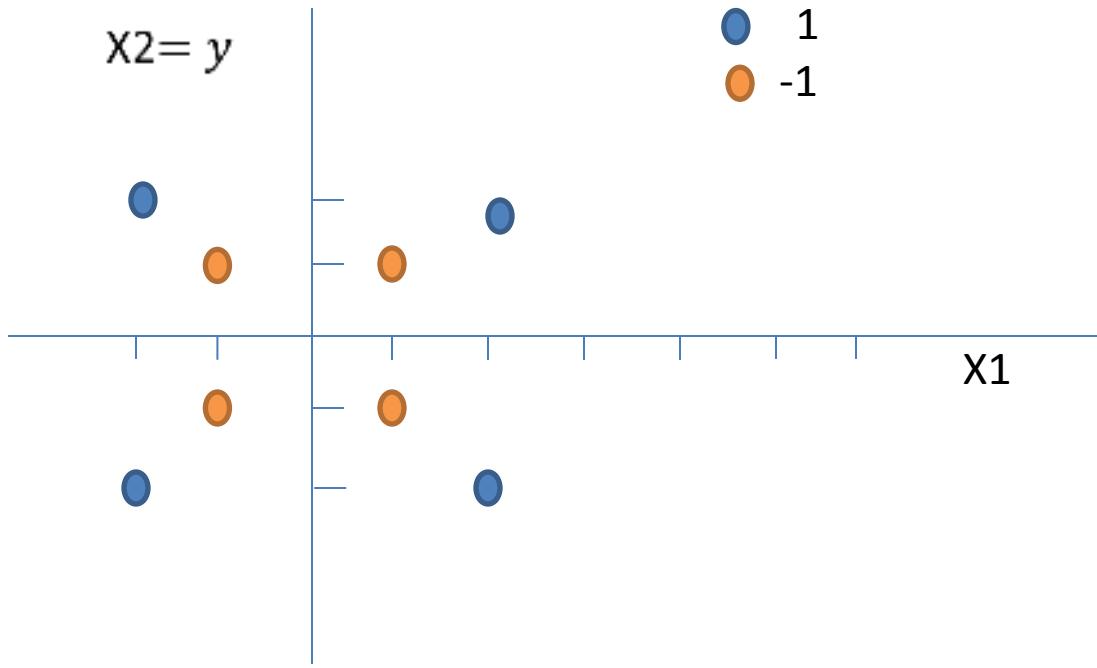


Pattern no	X_1	X_2	class
1	3	1	1
2	3	-1	1
3	6	1	1
4	6	-1	1
5	1	0	-1
6	0	1	-1
7	0	-1	-1
8	-1	0	-1

$x=2$

- The given equation: $x=2$ a vertical straight line parallel to y -axis at a distance of 2 unit on the right side hence it slope (m) will be $m=\tan(\pi/2)=\infty$
- The line: $x=2$ meets the y -axis at infinity hence its y -intercept will also be ∞

Numeric Example for Non Linear Problem



Pattern no	X_1	X_2	class
1	2	2	1
2	2	-2	1
3	-2	-2	1
4	-2	2	1
5	1	1	-1
6	1	-1	-1
7	-1	-1	-1
8	-1	1	-1

- The objective is to find a hyperplane which can solve the problem but no such hyperplane exists in this feature space
- Hence we need to transform from this feature space to another feature space where we can have a hyperplane that can solve the problem

Numeric Example

$$\lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4-x_2 + |x_1 - x_2| \\ 4-x_1 + |x_1 - x_2| \end{pmatrix}, & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{Otherwise} \end{cases}$$

Positive Examples

$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix} \right\} \rightarrow \left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 10 \end{pmatrix} \right\}$$

Negative Examples

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\} \rightarrow \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$$

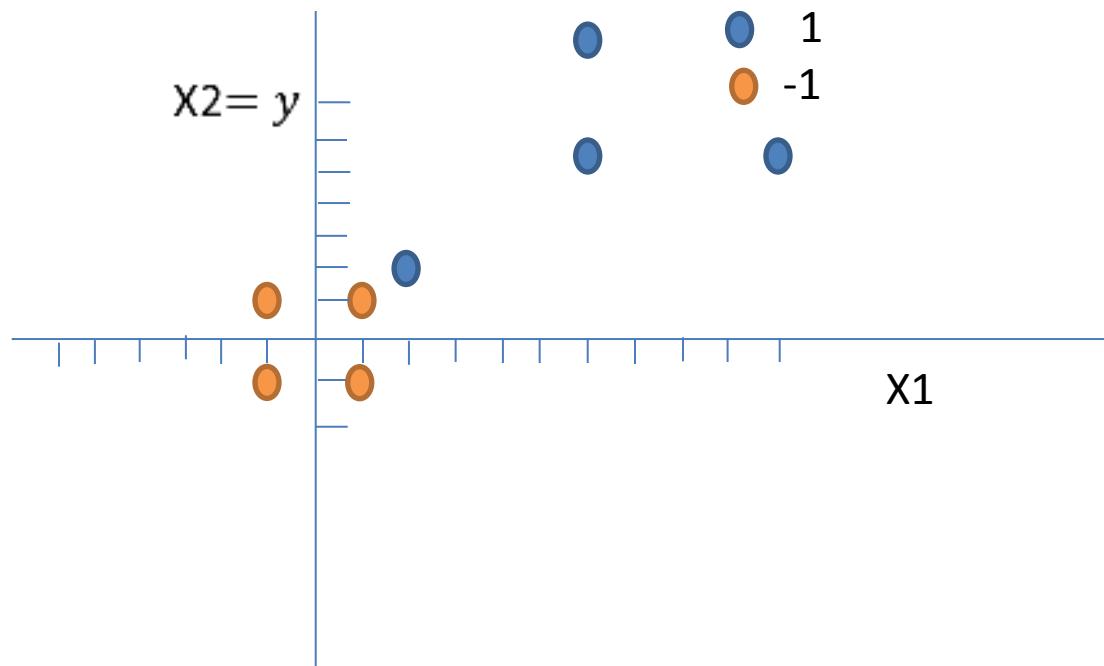
Numeric Example

Positive Examples

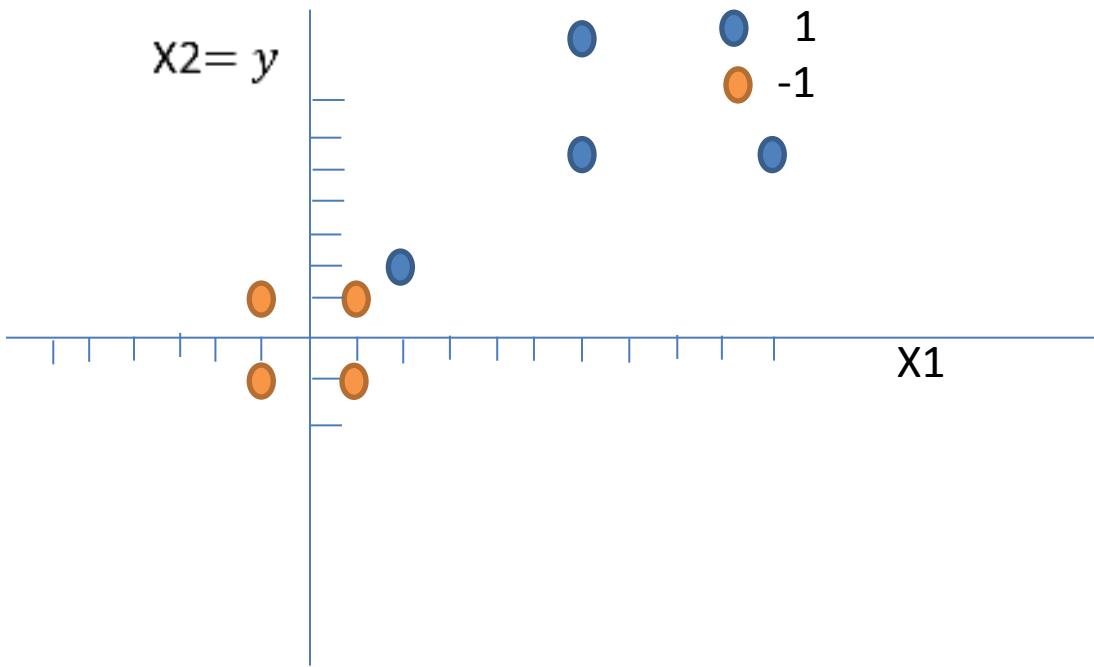
$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix} \right\} \rightarrow \left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 10 \end{pmatrix} \right\}$$

Negative Examples

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\} \rightarrow \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$$



Numeric Example



Support Vectors are $S_1 = (1, 1)$
 $S_2 = (2, 2)$

Vectors are augmented with bias

$$S_1' = (1, 1, 1)$$
$$S_2' = (2, 2, 1)$$

Numeric Example

$$S1' = (1, 1, 1)$$

$$S2' = (2, 2, 1)$$

To calculate three weights following equations are established:

$$\alpha_1 S1'.S1' + \alpha_2 S1'.S2' = -1$$

$$\alpha_1 S2'.S1' + \alpha_2 S2'.S2' = 1$$

$$\alpha_1(1, 1, 1).(1, 1, 1) + \alpha_2(1, 1, 1).(2, 2, 1) = -1$$

$$\alpha_1(1+1+1) + \alpha_2(2+2+1) = -1$$

$$3\alpha_1 + 5\alpha_2 = -1$$

$$\alpha_1 S2'.S1' + \alpha_2 S2'.S2' = 1$$

$$\alpha_1(2, 2, 1).(1, 1, 1) + \alpha_2(2, 2, 1).(2, 2, 1) = 1$$

$$\alpha_1(2+2+1) + \alpha_2(4+4+1) = 1$$

$$5\alpha_1 + 9\alpha_2 = 1$$

$$\alpha_1 = -7 \quad \alpha_2 = 4$$

Numeric Example

$$\begin{aligned}w' &= \sum_i \alpha_i S'_i \\&= -7(1\ 1\ 1)^t + 4(2\ 2\ 1)^t + \\&= (1\ 1\ -3)^t\end{aligned}$$

$$\begin{aligned}w &= (1\ 1)^t \\b &= -3\end{aligned}$$

$$(c_0 + c_1 X_1 + c_2 X_2 + \dots + c_N X_N = 0)$$

$$w_1x_1 + w_2x_2 + c = 0$$

$$1 \cdot x_1 + 1 \cdot x_2 + (-3) = 0$$

$$x_1 + x_2 - 3 = 0$$

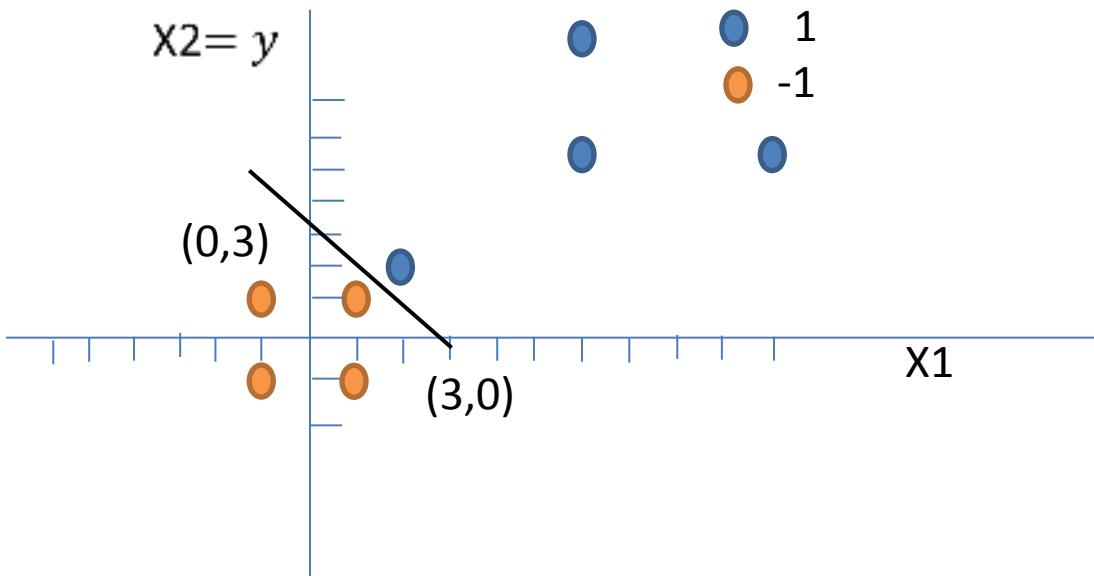
$$x_2 = -x_1 + 3$$

If $x_1 = 0$; $x_2 = 3$ point $(0, 3)$

If $x_2 = 0$; $x_1 = 3$ point $(3, 0)$

Thus, a line of 45 degree angle is created

Numeric Example



$$w_1x_1 + w_2x_2 + c = 0$$

$$1 \cdot x_1 + 1 \cdot x_2 + (-3) = 0$$

$$x_1 + x_2 - 3 = 0$$

$$x_2 = -x_1 + 3$$

If $x_1 = 0$; $x_2 = 3$ point $(0, 3)$

If $x_2 = 0$; $x_1 = 3$ point $(3, 0)$

Thus a line of 45 degree angle is created

Applications of SVM in Real World

Face detection – SVMs classify parts of the image as a face and non-face and create a square boundary around the face.

Text and hypertext categorization – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value.

Classification of images – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.

Bioinformatics – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.

Protein fold and remote homology detection – Apply SVM algorithms for protein remote homology detection.

Handwriting recognition – We use SVMs to recognize handwritten characters used widely.

Generalized predictive control(GPC) – Use SVM based GPC to control chaotic dynamics with useful parameters.

Classification of Machine Learning Algorithms

Unsupervised

- Clustering
 - K-Means
 - K-Medoid
 - CURE
 - BIRCH
- Association
 - Apriori Algorithm
 - Predictive Apriori Algorithm
 - Tertius Algorithm
 - E clat

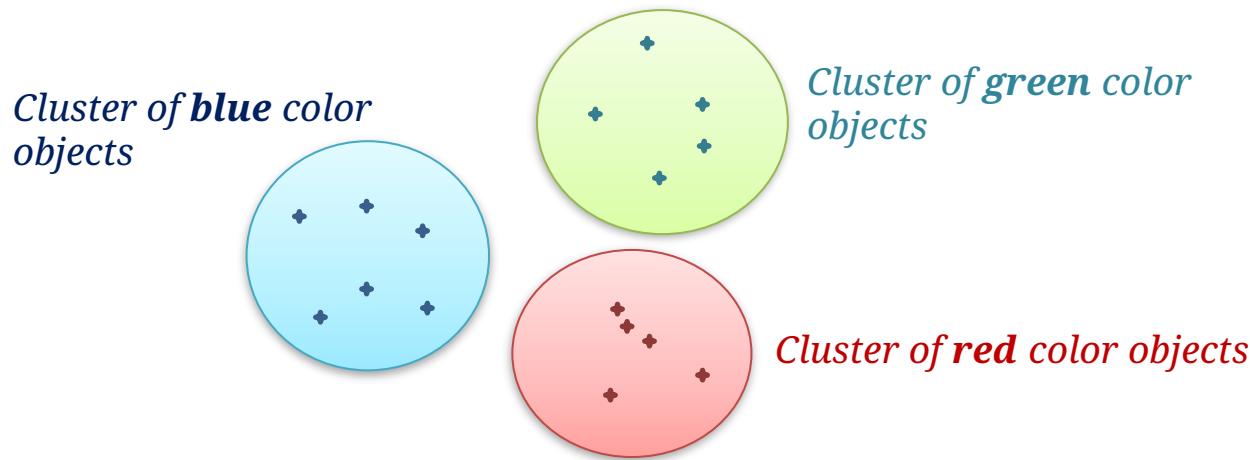
Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

$$\text{eigenvalues} = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

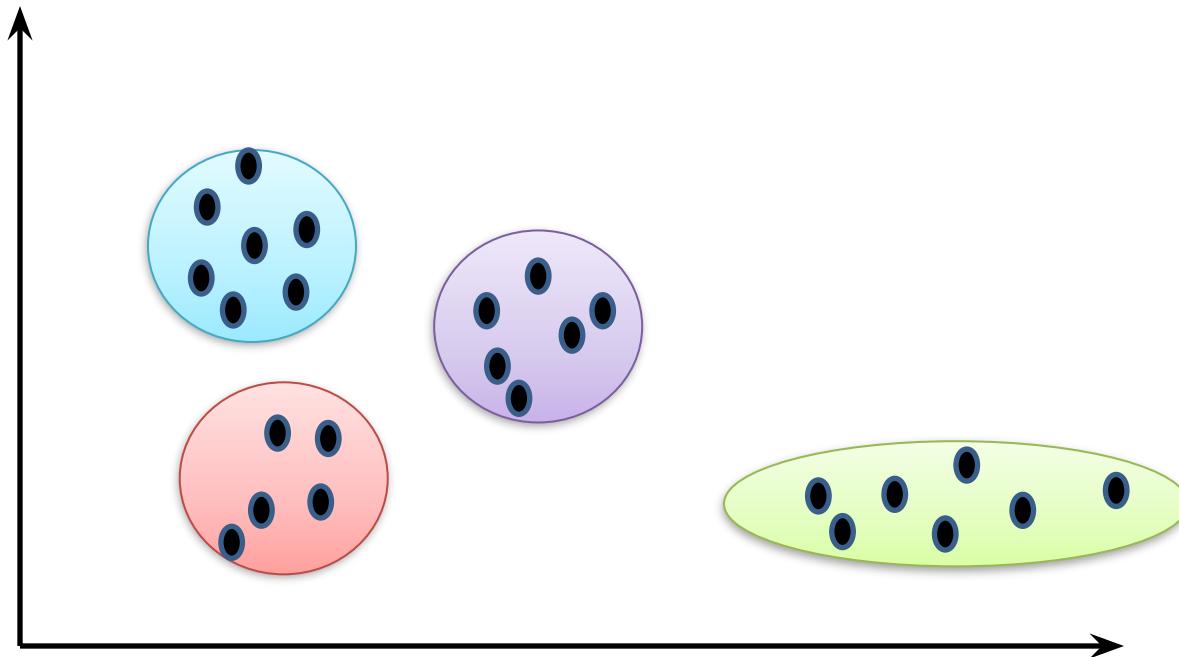
Clustering

- Patterns can be put into groups based on the similarity of their attributes values. Such groups are called clusters.
- The process of forming clusters is called clustering
- Clustering is an unsupervised learning task.



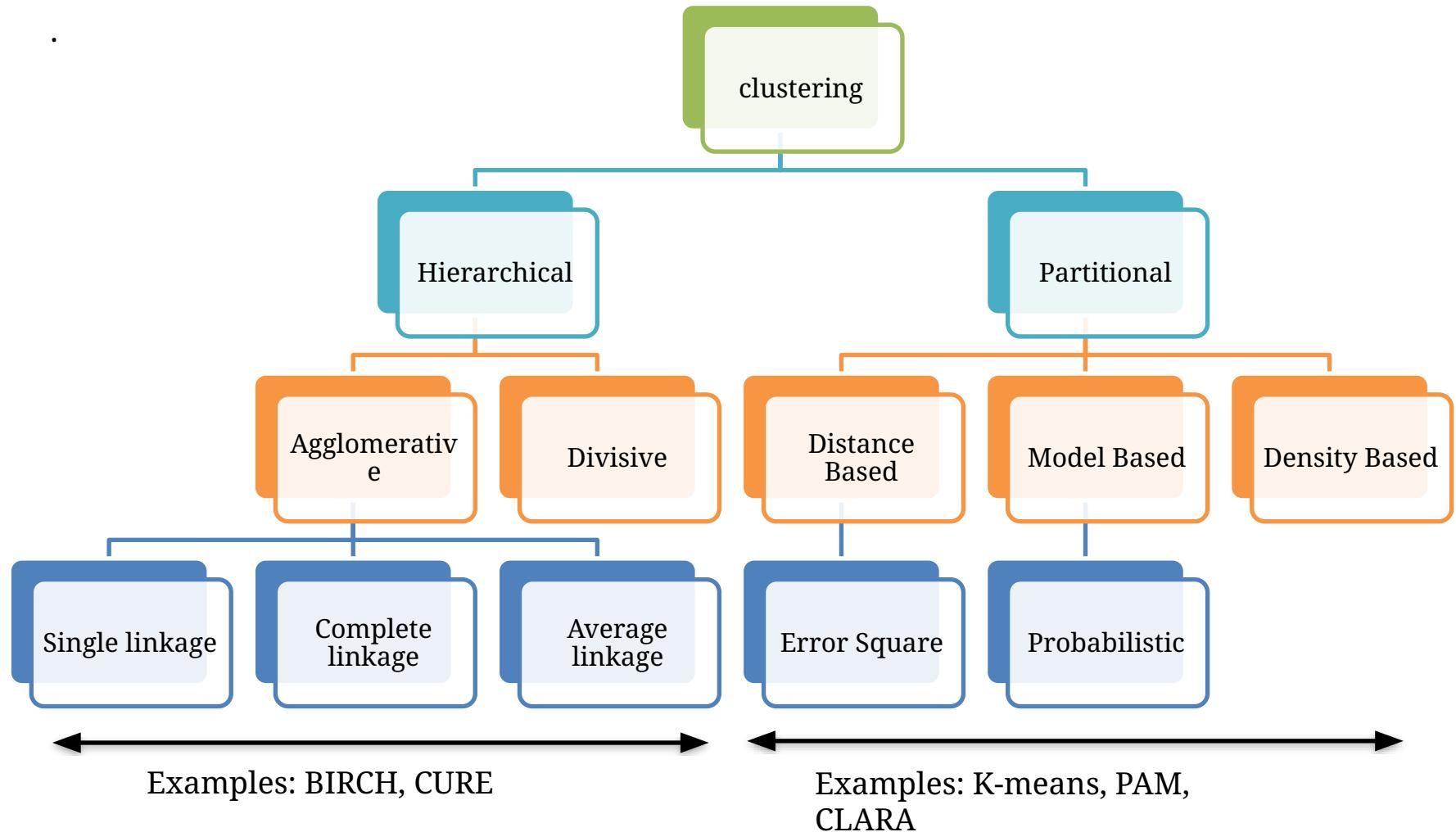
- Principle: Maximizing intra-class similarity & minimizing interclass similarity.
- Applications: Economic Science (especially market research), WWW (Document classification, Cluster Weblog data to discover groups of similar access patterns), Pattern Recognition, Spatial Data Analysis (creating thematic maps in GIS by clustering feature spaces), Image Processing etc.

Clustering



- There are some algorithms which can make clusters of spherical shape and considers attributes are uniformly distributed.
- They can capture linear relationship of attributes. Ex. K-Means
- There are some algorithms which can make clusters of non-spherical shape and attributes are not uniformly distributed. Ex. CURE

Classification of Clustering



Hierarchical clustering

1. Calculate distances between patterns

Dataset S

Patterns	Value of attributes	
	A1	A2
X1	1	1
X2	2	3
X3	3	1
X4	4	4
X5	5	2

Let $p = (p_1, p_2)$ and $q = (q_1, q_2)$ be two points:

- ✓ City block distance $d(p, q) = |p_1 - q_1| + |p_2 - q_2|$
 - ✓ Euclidean distance $d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$
 - ✓ Minkowski distance $d(p, q) = \sum_{i=1}^M (|p_i^n - q_i^n|^r)^{\frac{1}{r}}$
- For r=1, Minkowski distance –City block distance
 For r=2, Minkowski distance – Euclidean distance

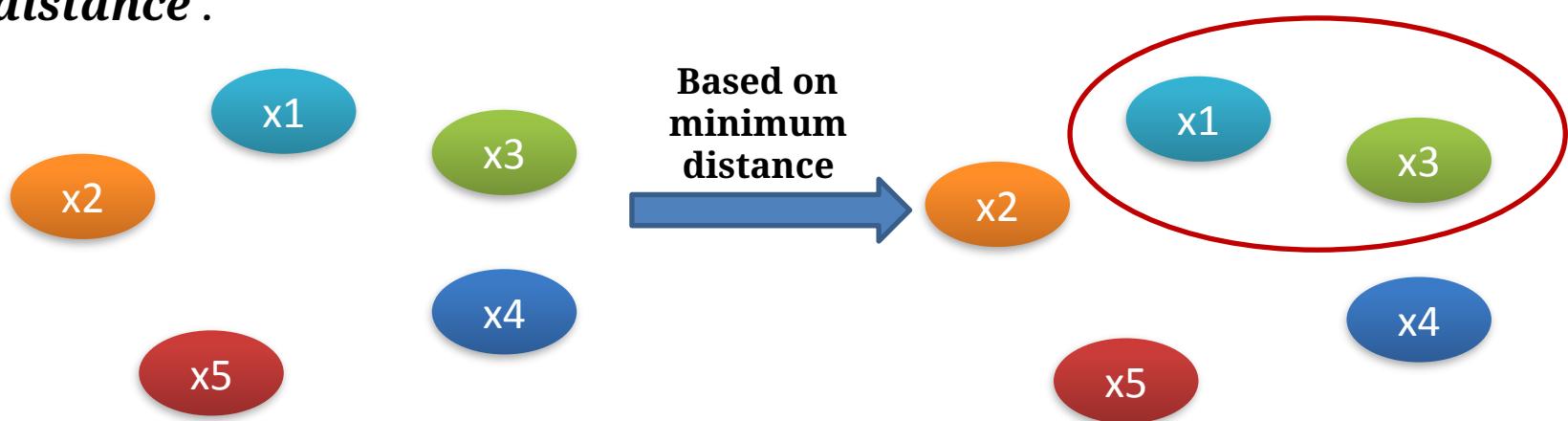
Patterns	X1	X2	X3	X4	X5
X1	0	2.2	2	4.2	4.1
X2	2.2	0	2.2	2.2	3.2
X3	2	2.2	0	3.2	2.2
X4	4.2	2.2	3.2	0	2.2
X5	4.1	3.2	2.2	2.2	0

Clustering (cont..)

Euclidean distance between the patterns

Patterns	X1	X2	X3	X4	X5
X1	0	2.2	2	4.2	4.1
X2	2.2	0	2.2	2.2	3.2
X3	2	2.2	0	3.2	2.2
X4	4.2	2.2	3.2	0	2.2
X5	4.1	3.2	2.2	2.2	0

2. *Each pattern is a cluster. Group clusters with minimum distance .*



Clustering (cont..)

3. Recalculate distances between clusters.

Some of the clustering techniques to recalculate distances: Single linkage, complete linkage, Average linkage.

a) Single

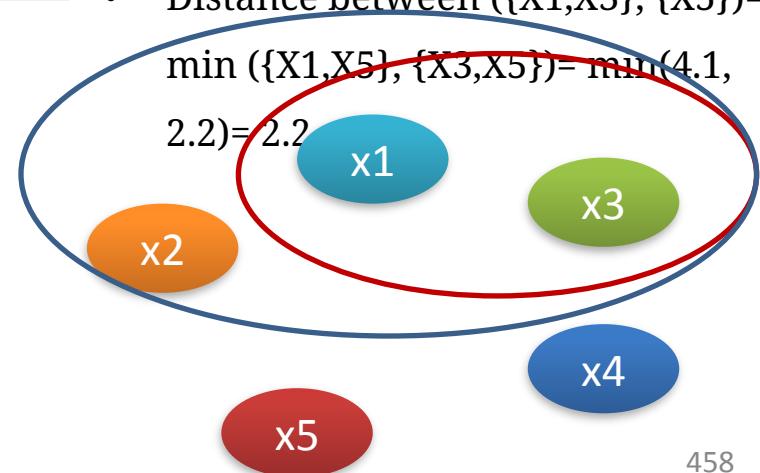
linkage Euclidean distance between the

Patterns	X1	X2	X3	X4	X5
X1	0	2.2	2	4.2	4.1
X2	2.2	0	2.2	2.2	3.2
X3	2	2.2	0	3.2	2.2
X4	4.2	2.2	3.2	0	2.2
X5	4.1	3.2	2.2	2.2	0

New distances using Single Linkage

Patterns	{X1,X3}	{X2}	{X4}	{X5}
{X1,X3}	0	2.2	3.2	2.2
{X2}	2.2	0	2.2	3.2
{X4}	3.2	2.2	0	2.2
{X5}	2.2	3.2	2.2	0

- Distance between (<{X1,X3}, {X2})= min ({X1,X2}, {X3,X2})= min(2.2, 2.2)= 2.2
- Distance between (<{X1,X3}, {X4})= min ({X1,X4}, {X3,X4})= min(4.2, 3.2)= 3.2
- Distance between (<{X1,X3}, {X5})= min ({X1,X5}, {X3,X5})= min(4.1, 2.2)= 2.2



Clustering (cont..)

3. Recalculate distances between clusters.

- Some of the clustering techniques to recalculate distances: Single linkage, complete linkage, Average linkage.

b) Complete linkage:

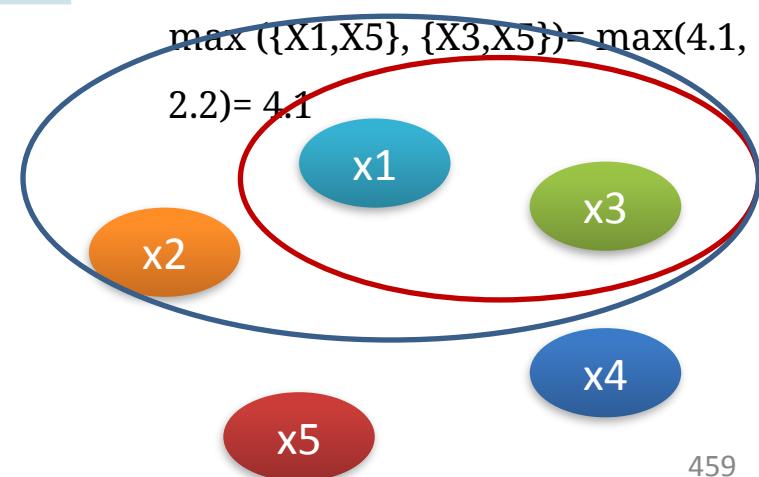
TABLE 2: Euclidean distance between the patterns

Patterns	X1	X2	X3	X4	X5
X1	0	2.2	2	4.2	4.1
X2	2.2	0	2.2	2.2	3.2
X3	2	2.2	0	3.2	2.2
X4	4.2	2.2	3.2	0	2.2
X5	4.1	3.2	2.2	2.2	0

TABLE 4: New distances using complete Linkage

Pattern s	{X1,X3 }	{X2}	{X4}	{X5}
{X1,X3}	0	2.2	4.2	4.1
{X2}	2.2	0	2.2	3.2
{X4}	4.2	2.2	0	2.2
{X5}	4.1	3.2	2.2	0

- Distance between (<{X1,X3}, {X2})= max ({X1,X2}, {X3,X2})= max(2.2, 2.2)= 2.2
- Distance between (<{X1,X3}, {X4})= max ({X1,X4}, {X3,X4})= max(4.2, 3.2)= 4.2
- Distance between (<{X1,X3}, {X5})= max ({X1,X5}, {X3,X5})= max(4.1, 2.2)= 4.1



Clustering (cont..)

3. Recalculate distances between clusters.

- Some of the clustering techniques to recalculate distances: Single linkage, complete linkage, Average linkage.

c) Average linkage:

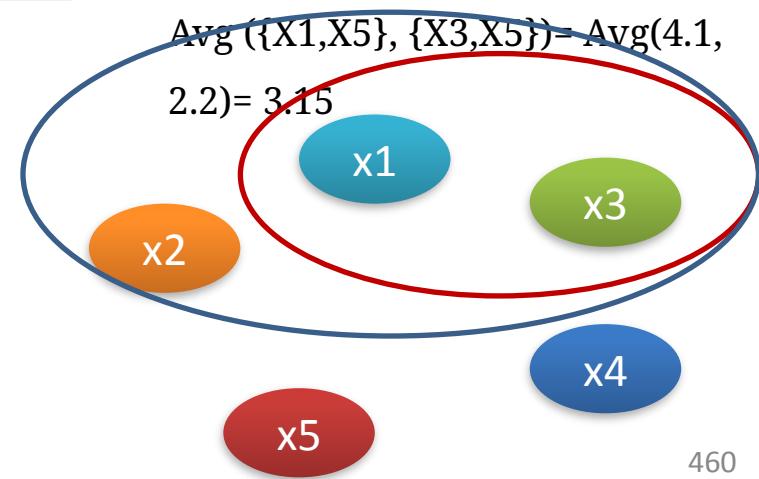
TABLE 2: Euclidean distance between the patterns

Patterns	X1	X2	X3	X4	X5
X1	0	2.2	2	4.2	4.1
X2	2.2	0	2.2	2.2	3.2
X3	2	2.2	0	3.2	2.2
X4	4.2	2.2	3.2	0	2.2
X5	4.1	3.2	2.2	2.2	0

TABLE 5: New distances using average linkage

Patterns	{X1,X3}	{X2}	{X4}	{X5}
{X1,X3}	0	2.2	3.7	3.15
{X2}	2.2	0	2.2	3.2
{X4}	3.7	2.2	0	2.2
{X5}	3.15	2.2	2.2	0

- Distance between (<{X1,X3}, {X2})= Avg ({X1,X2}, {X3,X2})= Avg(2.2, 2.2)= 2.2
- Distance between (<{X1,X3}, {X4})= Avg ({X1,X4}, {X3,X4})= Avg(4.2, 3.2)= 3.7
- Distance between (<{X1,X3}, {X5})= Avg ({X1,X5}, {X3,X5})= Avg(4.1, 2.2)= 3.15



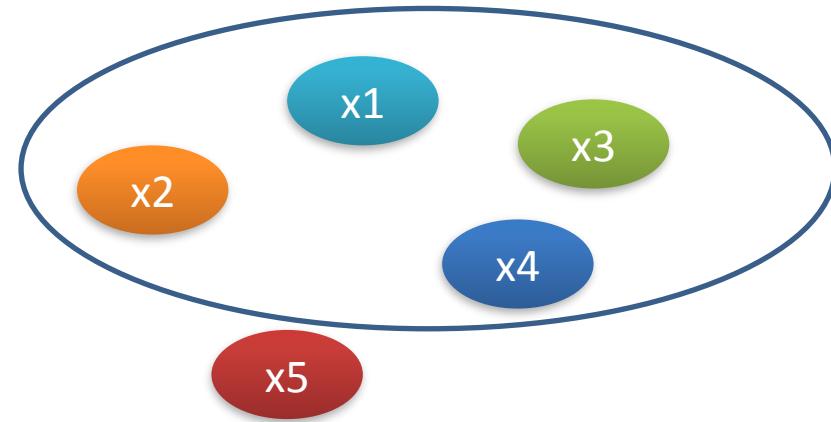
Hierarchical Clustering

4. Continue forming new clusters using the selected clustering technique until new cluster can be formed.

- Continuing with the example.

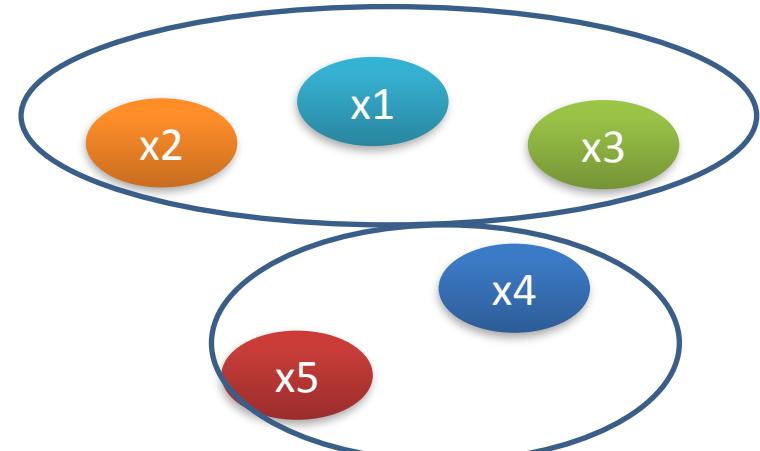
a) Single linkage:

Patterns	{X1,X2,X3}	{X4}	{X5}
{X1,X2,X3}	0	2.2	2.2
{X4}	2.2	0	2.2
{X5}	2.2	2.2	0



b) Complete linkage:

Patterns	{X1,X2,X3}	{X4}	{X5}
{X1,X2,X3}	0	4.2	4.1
{X4}	4.2	0	2.2
{X5}	4.1	2.2	0



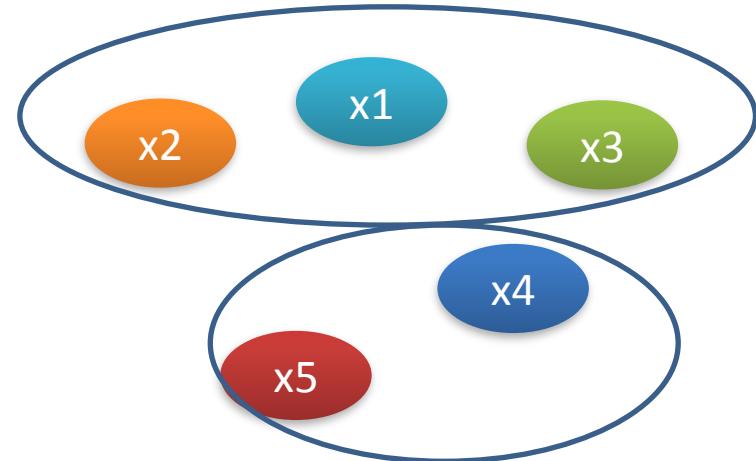
Hierarchical Clustering

4. Continue forming new clusters using the selected clustering technique until new cluster can be formed.

- Continuing with the example.

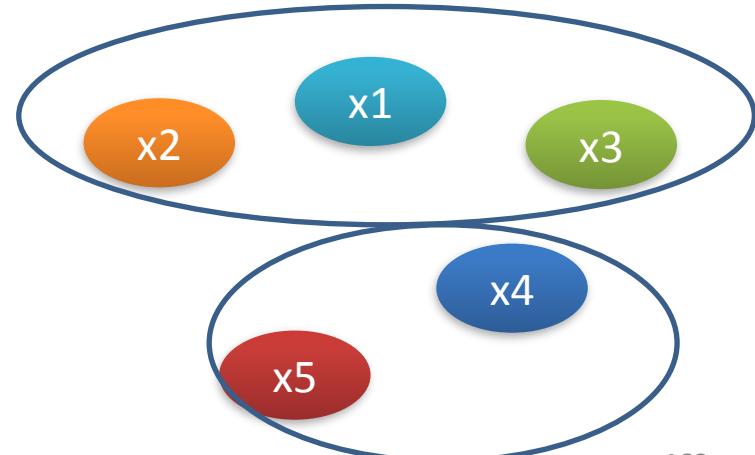
c) Average linkage:

Patterns	{X1,X2,X3}	{X4}	{X5}
{X1,X2,X3}	0	3.2	3.2
{X4}	3.2	0	2.2
{X5}	3.2	2.2	0



d) Centroid linkage:

Patterns	{X1,X2,X3}	{X4}	{X5}
{X1,X2,X3}	0	3.07	3.02
{X4}	3.07	0	2.2
{X5}	3.02	2.2	0



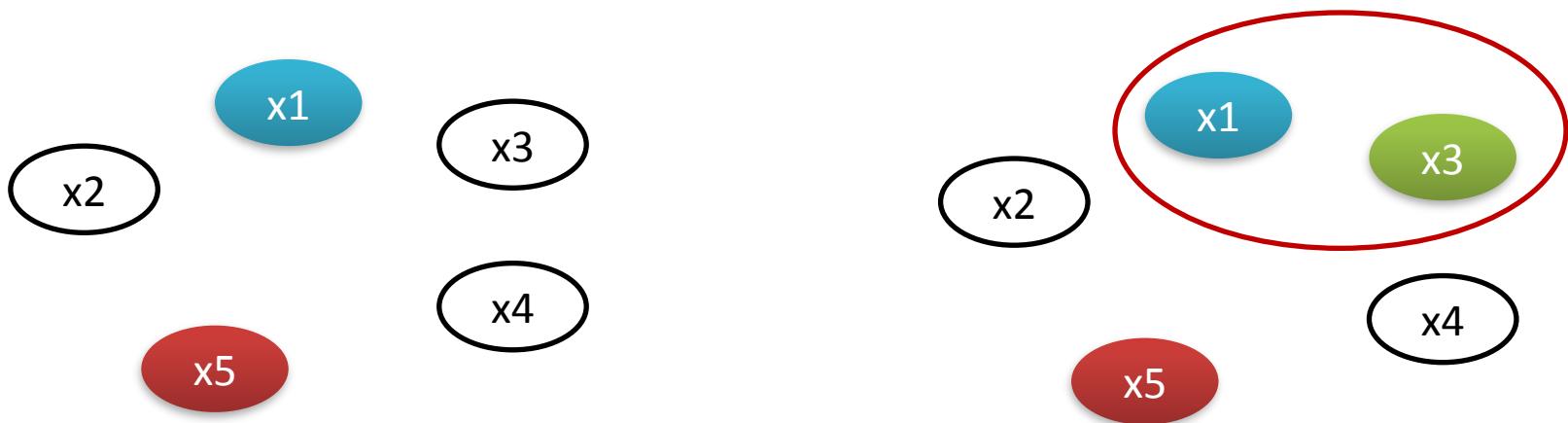
Summary of Hierarchical Clustering

Procedure	The two clusters obtained
Single-Linkage	{ X1, X2, X3, X4} and {X5}
Complete-linkage	{ X1, X2, X3} and {X4, X5}
Average-linkage	
Centroid-linkage	

Partition Clustering

Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k \leq n$.

An Example given below:



Partition Clustering

k-means Clustering

- A partitioning method of clustering.
- Each cluster is associated with a centroid.
- Each point is assigned to the cluster with the closest centroid.
- Number of clusters, k must be specified.

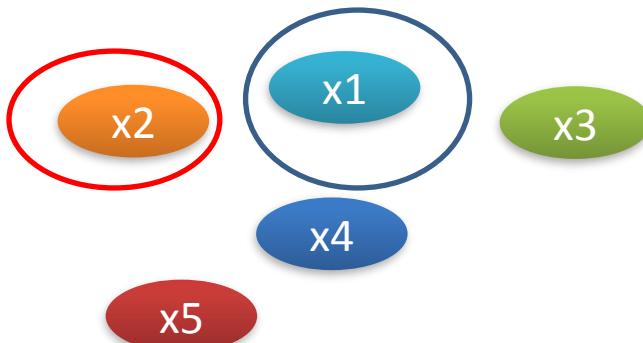
Partition Clustering

k-means example:

Datasets

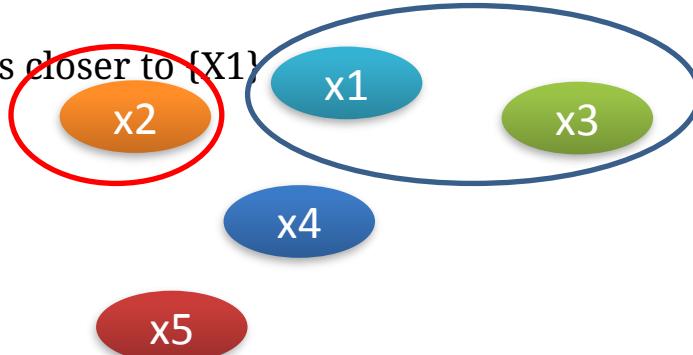
Patterns	Value of attributes	
	A1	A2
X1	1	1
X2	2	3
X3	3	1
X4	4	4
X5	5	2

Step 1: Consider K=2. Suppose {X1} and {X2} are the two clusters.



Step 2:

- Distance of X3 and the centroid of cluster {X1} is 2
- Distance of X3 and the centroid of cluster {X2} is 2.2
- X3 is closer to {X1}



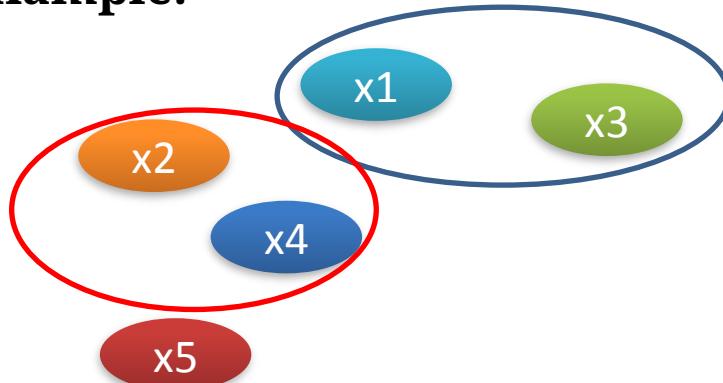
centroid of the two clusters

{X1,X3}	{X2}
(2,1)	(2,3)

- Distance of X4 and the centroid of cluster {X1,X3} is 3.6
- Distance of X4 and the centroid of cluster {X2} is 2.2
- X4 is closer to {X2}

Partition Clustering

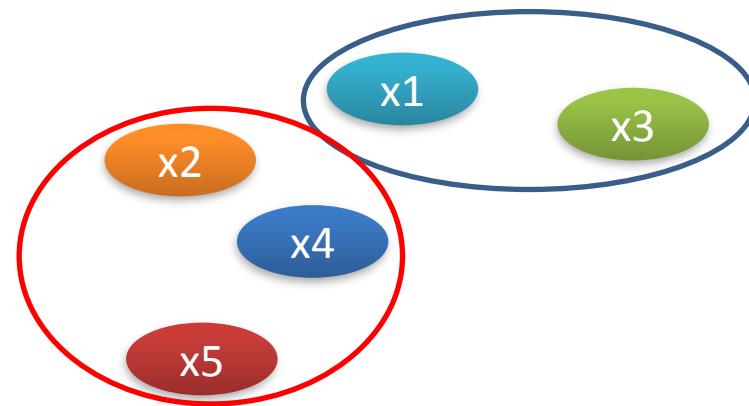
k-means
example:



centroid of the two clusters

{X1,X3}	{X2,X4}
(2,1)	(3,3.5)

- Distance of X5 and the centroid of cluster {X1,X3} is 3.2
- Distance of X5 and the centroid of cluster {X2,X4} is 2.5
- X5 is closer to {X2,X4}



centroid of the final clusters

{X1,X3}	{X2,X4,X5}
(2,1)	(3.67,3)

Step3: Distances of the patterns from the final clusters

Patterns	{X1,X3}	{X2,X4,X5}
X1	1	3.3
X2	2	1.67
X3	1	2.1
X4	3.6	1.1
X5	3.2	1.7

k-means

Algorithm

1. Consider k of the r given patterns to each form a cluster, leaving the remaining patterns as such.
2. For $i=k+1, k+2, \dots, r$ do
 - Put the i^{th} pattern in the cluster whose centroid is nearest to the pattern.
 - Compute the cluster's new centroid .
3. For $i=1, 2, \dots, r$ do steps 3.1 to 3.3
 - 3.1 Let C' be the cluster which has the i^{th} pattern. Calculate the distance between the i^{th} pattern and the centroids of each of the k clusters. If the pattern is closest to the centroid of the cluster C' , then its does not changes its cluster , go to step 3.3.
 - 3.2 Let C'' be the cluster whose centroid is closer to the i^{th} pattern . Move the pattern from C' to C'' . Compute the new centroids of the clusters C' and C'' . Go to step 3.3.
 - 3.3 continue.
4. No patterns changed clusters in the last iteration; hence return from the procedure with the k clusters.

Advantages of k-means Algorithm

1. Robust
2. Easy to understand
3. Comparatively efficient
4. If data sets are distinct, then gives the best results
5. Produce tighter clusters
6. Flexible
7. Easy to interpret
8. Enhances Accuracy
9. Works better with spherical clusters

Disadvantages of k-means Algorithm

1. Needs prior specification for the number of cluster centers
2. If there are two highly overlapping data, then it cannot be distinguished and cannot tell that there are two clusters
3. Cannot handle outliers and noisy data
4. Do not work for the non-linear data set
5. Lacks consistency
6. With the different representations of the data, the results achieved are also different
7. It gives the local optima of the squared error function
8. Sometimes choosing the centroids randomly cannot give fruitful results
9. It can be used only if the meaning is defined
10. If very large data sets are encountered, then the computer may crash.
11. Prediction issues

Summary of Clustering

Procedure	The two clusters obtained
Single-Linkage	{ X1, X2, X3, X4} and {X5}
Complete-linkage	{ X1, X2, X3} and {X4, X5}
Average-linkage	
Centroid-linkage	
k-means	{ X1, X3} and {X2, X4, X5}

Non-numeric Attributes

Clustering

Clustering example with non-numeric attributes:

Patterns	Attributes			Class
	Habit	Eat	Footwear	
T1	GABBY	BAKED	CLOGS	STUDENT
T2	GABBY	ROASTED	SANDALS	PROFESSOR
T3	GABBY	BAKED	SANDALS	STUDENT
T4	QUIET	FRIED	SANDALS	PROFESSOR
T5	GABBY	FRIED	CLOGS	STUDENT
T6	QUIET	BAKED	SANDALS	STUDENT
T7	GABBY	FRIED	SANDALS	PROFESSOR
T8	QUIET	FRIED	CLOGS	STUDENT

1. Calculate the Hamming distance between the patterns
Let 1101 1001 and 1001 1101 be two strings.

$$11011001 \oplus 10011101 = 01000100.$$

Since, this contains two 1s, the Hamming distance, $d(11011001, 10011101) = 2$.

Hamming distance between patterns

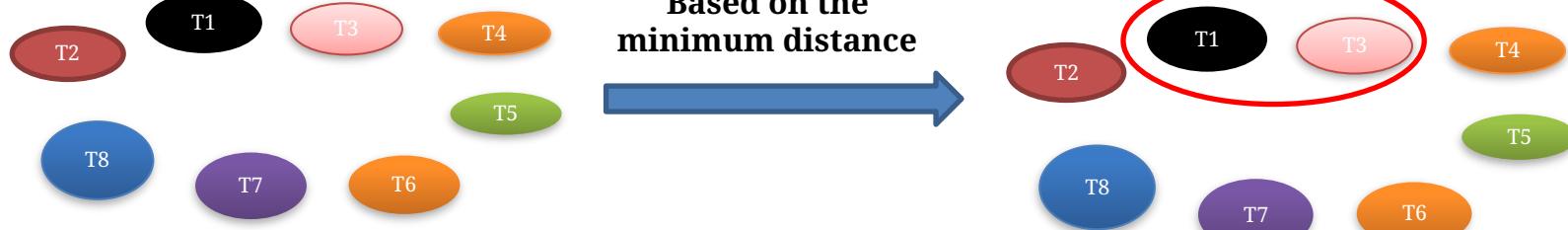
Patterns	T1	T2	T3	T4	T5	T6	T7	T8
T1	0	2	1	3	1	2	2	2
T2	2	0	1	2	2	2	1	3
T3	1	1	0	2	2	1	1	3
T4	3	2	2	0	2	1	1	1
T5	1	2	2	2	0	3	1	1
T6	2	2	1	1	3	0	2	2
T7	2	1	1	1	1	2	0	2
T8	2	3	3	1	1	2	2	0

Clustering

Hamming distance between patterns

Patterns	T1	T2	T3	T4	T5	T6	T7	T8
T1	0	2	1	3	1	2	2	2
T2	2	0	1	2	2	2	1	3
T3	1	1	0	2	2	1	1	3
T4	3	2	2	0	2	1	1	1
T5	1	2	2	2	0	3	1	1
T6	2	2	1	1	3	0	2	2
T7	2	1	1	1	1	2	0	2
T8	2	3	3	1	1	2	2	0

2. Each pattern is a cluster. Group clusters with minimum distance from table 11



Clustering

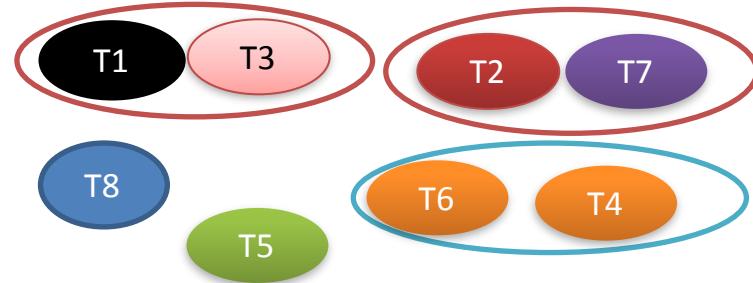
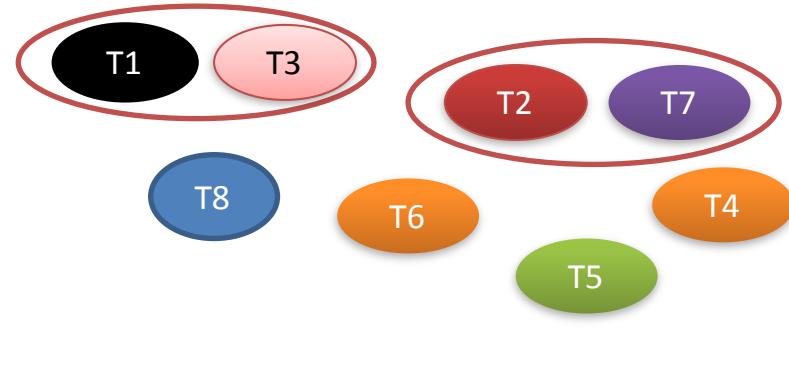
3. Recalculate the distance between the clusters using Average linkage

New distance using Average

Patterns	linkage } {T1,T3}	{T2 } {T2 }	{T4 } {T4 }	{T5 } {T5 }	{T6 } {T6 }	{T7 } {T7 }	{T8 } {T8 }
{T1,T3}	0	1.5	2.5	1.5	1.5	1.5	2.5
{T2}	1.5	0	2	2	2	1	3
{T4}	2.5	2	0	2	1	1	1
{T5}	1.5	2	2	0	3	1	1
{T6}	1.5	2	1	3	0	2	2
{T7}	1.5	1	1	1	2	0	2
{T8}	2.5	3	1	1	2	2	0

4. Continue forming new clusters until new clusters can be formed

Patterns	{T1,T3}	{T2,T7}	{T4}	{T5}	{T6}	{T8}
{T1,T3}	0	1.5	2.5	1.5	1.5	2.5
{T2,T7}	1.5	0	1.5	1.5	2	2.5
{T4}	2.5	1.5	0	2	1	1
{T5}	1.5	1.5	2	0	3	1
{T6}	1.5	2	1	3	0	2
{T8}	2.5	2.5	1	1	2	0



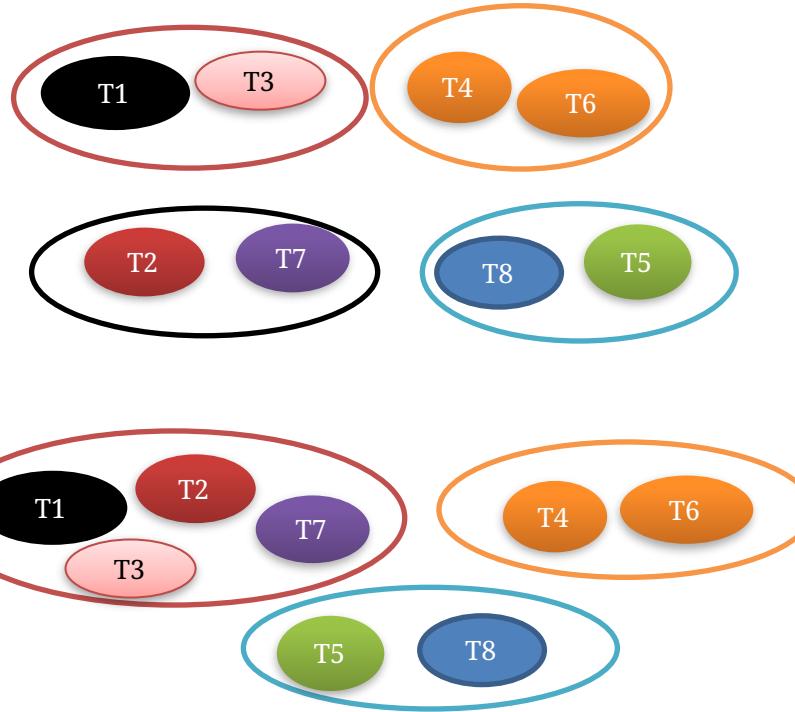
Clustering

Continuing with the example

Patterns	{T1,T3}	{T2,T7}	{T4,T6}	{T5}	{T8}
{T1,T3}	0	1.5	2	1.5	2.5
{T2,T7}	1.5	0	1.75	1.5	2.5
{T4,T6}	2	1.75	0	2.5	1.5
{T5}	1.5	1.5	2.5	0	1
{T8}	2.5	2.5	1.5	1	0

Patterns	{T1,T3}	{T2,T7}	{T4,T6}	{T5,T8}
{T1,T3}	0	1.5	2	2
{T2,T7}	1.5	0	1.75	2
{T4,T6}	2	1.75	0	2
{T5,T8}	2	2	2	0

Patterns	{T1,T2,T3,T7}	{T4,T6}	{T5,T8}
{T1,T2,T3,T7}	0	1.88	2
{T4,T6}	1.88	0	2
{T5,T8}	2	2	0



**THANK
YOU
!!!!**

Machine Learning (CS 431)

Presented by

Dr. Saroj Kr. Biswas
Associate Professor & HoD,
CSE



**Department of Computer Science and Engineering
National Institute of Technology, Silchar**

Machine Learning

Introduction, Decision Trees learning, Probability Primer, Bayes Decision Theory, Maximum-likelihood and Bayesian Parameter Estimation, Non-parametric Techniques, Bayes Networks, Optimization, Primer, Linear Discriminant Functions, Support Vector Machines. (introduction of ML with description of ML frame work)

Unit-2 Unsupervised Learning, Semi Supervised Learning, Reinforcement Learning, Statistical learning methods, PAC learning framework, Occam's Razor. (different ML algorithm)

Text Books

1. Mitchell T. M. , Machine Learning , McGraw Hill
2. Duda R. O., Hart P. E., Strok D. G. , Pattern Classification, Wiley Interscience

Course Outcomes (COs)

- 1.Understand the principles, advantages, limitations and possible applications of machine learning.
- 2. Identify the appropriate machine learning techniques for classification (**for other task also**).
- 3. Apply various pattern recognition, optimization and decision problems in Machine learning. (**Design of ML model to solve different applications in domains**)

Evaluation scheme

Internal Assessment = 20 (test, quiz, attendance, assignment)

Mid Semester = 30

End Semester = 50

What is Learning?

Learning is a process to acquire new or partially new knowledge by improving its performance from experience or environment.

There are different kinds of learning methods.

Learning

Concept learning:

Concept learning is also known as **category learning**, **concept attainment**, and **concept**.

In a concept learning task, a human or machine learner is trained to classify objects by being shown a set of example objects along with their class labels.

Rote Learning (memorization):

Memorizing things without knowing the concept/ logic behind them. A chartered engineer could play the role of a project manager while students play the role of the engineers during a meeting.

Passive Learning (instructions):

Passive learning is a learning paradigm **where learners receive information from the instructor and adopt it.**

Learning from a teacher/expert. (Direct Instruction, Watching Television, Modeled Instruction, University Lectures).

Learning

Active learning:

Active learning is the subset of **machine learning** in which a learning algorithm can query a user interactively to label data with the desired outputs.

Examples: case studies, group projects, think-pair-share, peer teaching, debates

Analogy (experience):

Learning new things from our past experience. Analogy learning can be described as the process of finding knowledge acquired in one domain and "using" it in a different domain by establishing similarities between "concepts" in the two domains and transferring relationships between concepts in one domain to the other.

Analogical learning typically involves:

- (1) identifying a similarity between two entities (concepts), often referred to as a source entity and a target entity, and
- (2) transferring properties or relationships from the source entity to the target entity.

Example: Case-Based Planning

Learning

Two kinds of analogy-based learning are here:

1. Transformational
2. Derivational

Transformational Analogy:

Look for a similar solution and copy it to the new situation making suitable substitutions wherever appropriate. Transformational analogy does not look at how the problem was solved it only looks at the final solution.

Suppose you are asked to prove a theorem in plane geometry. You might look for a previous solution of theorem which is very similar to current one

Derivational Analogy:

The history of the problem solution, the steps involved, is often relevant. We know how to find the area of a triangle and a square. Derivational analogy will help to solve the area/volume of a pyramid from this knowledge.

Learning

Inductive Learning (experience):

On the basis of past experience, formulating a generalized concept. Inductive reasoning makes broad generalizations from specific observations. Basically, there is data, then conclusions are drawn from the data. An example of inductive logic is, "The coin I pulled from the bag is a penny. Second coin from the bag is a penny. A third coin from the bag is a penny. Therefore, all the coins in the bag are pennies."

Deductive Learning:

Deriving new facts from past facts. Deductive reasoning, or deduction, starts out with a general statement, or hypothesis, and examines the possibilities to reach a specific, logical conclusion. For example, the premise "Every A is B" could be followed by another premise, "This C is A." Those statements would lead to the conclusion "This C is B." For example, "All men are mortal. Harold is a man. Therefore, Harold is mortal."

Learning

Abductive reasoning: Abductive reasoning usually starts with an incomplete set of observations and proceeds to the likeliest possible explanation for the group of observations. Abductive reasoning is often used by doctors who make a diagnosis based on test results.

For example, a person walks into their living room and finds torn up papers all over the floor. The person's dog has been alone in the room all day. The person concludes that the dog tore up the papers because it is the most likely scenario

What is ML?

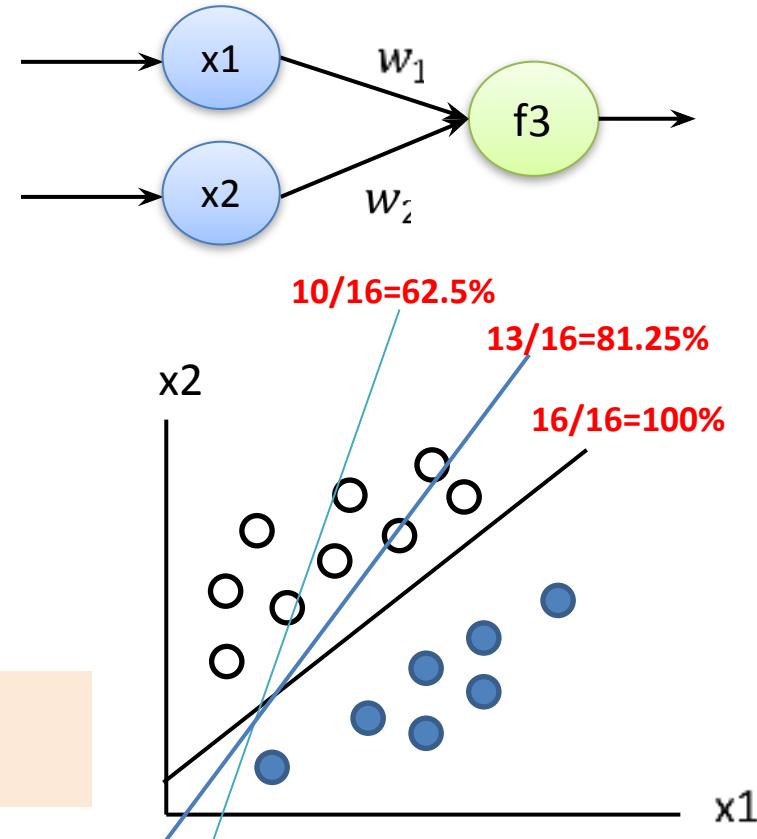
ML is programming computers to optimize a performance criterion using example data or past data. By E. Alpaydin

BP	Heart Beat	Class
120	70	Y
125	65	Y
130	59	N
150	78	N
135	66	N
125	75	N
120	76	Y

$$y = w_1x_1 + w_2x_2 + c$$

$$w_1x_1 + w_2x_2 + c = 0$$

$$x_2 = w_1/w_2 * x_1 + c/w_2$$



What is ML?

- o **Tom Mitchell** provides a more modern **definition**: “A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.” ... To find that logic is called **“machine learning”**

Why Machine Learning ?

- o For some defined tasks, if algorithms are available then machine learning is not required
- o For example, sorting numbers
- o For some tasks, algorithms are not readily available, to solve those tasks machine learning algorithms are required.
- o For example, to tell spam emails from legitimate emails.
- o The problems where no human experts exist. Rainfall forecasting
- o The problems where human experts exist, but they will be unable to explain their expertise. For example speech recognition.
- o The problems where the environment changes frequently. For example share market predictions.
- o The applications that need to be customised for individual users or for a group of users. For example, a program to filter unwanted electronic mail messages.
- o **ML is used to make quicker and reliable decision.**

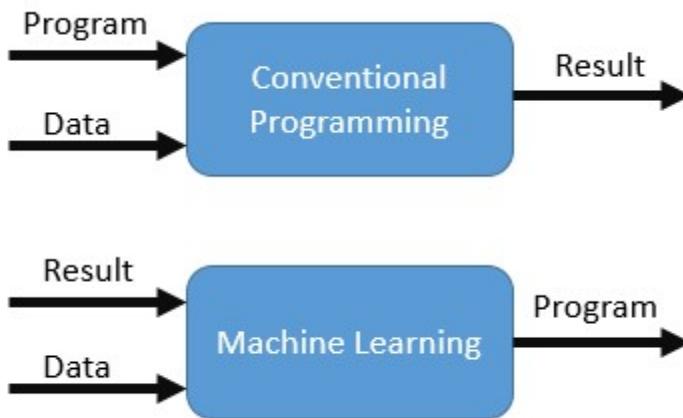
Some Facts about ML

- o ML does a good and useful approximation.
- o ML uses the theory of statistics in building mathematical models, because the core task is making inference from a sample.
- o Application of machine learning methods to large databases is called data mining
- o ML is a part of artificial intelligence.
- o ML helps us to find solutions to many problems in vision, speech recognition and robotics.
- o ML model may be predictive to make predictions in the future.
- o ML model may also be descriptive to gain knowledge or it can be both

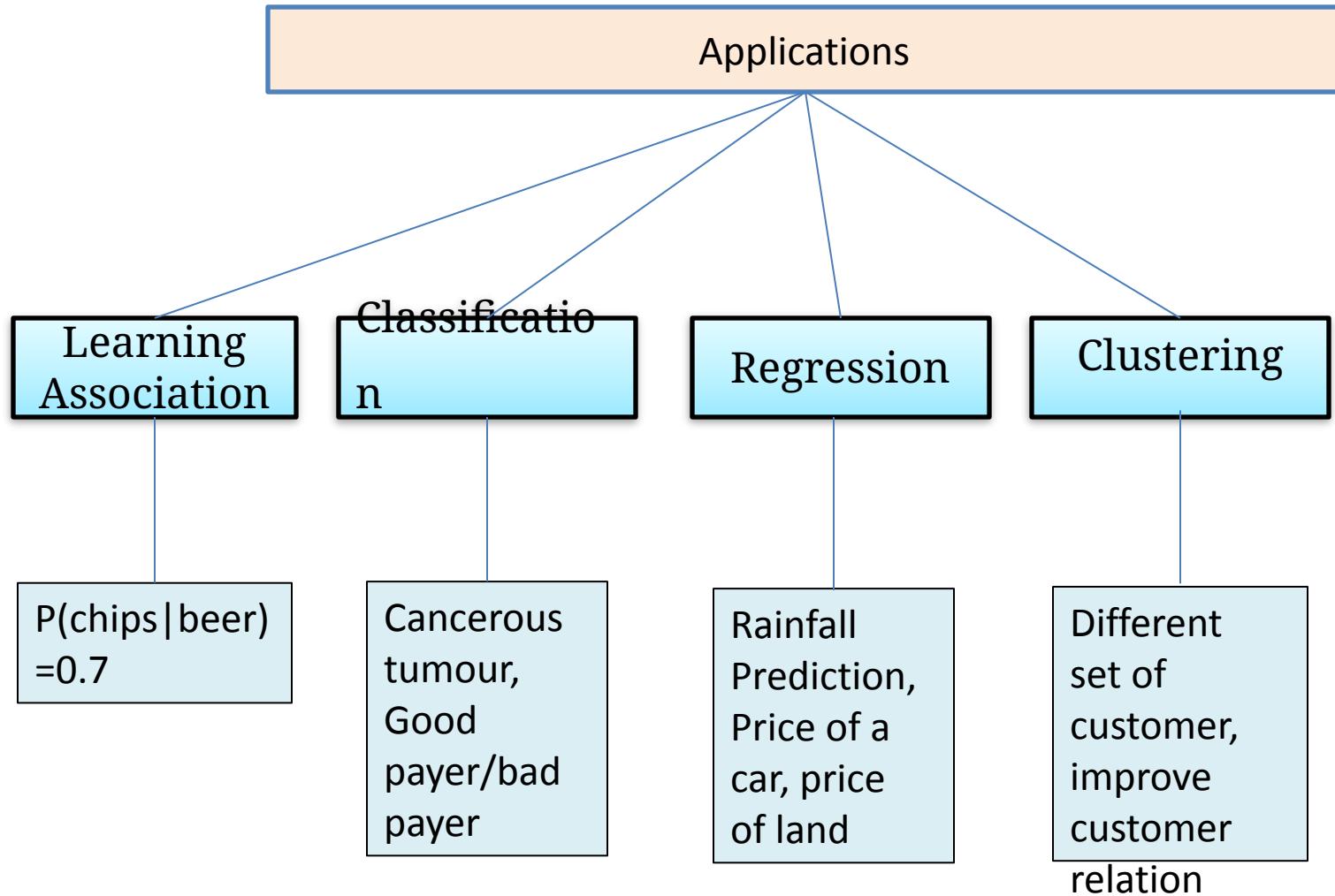
Conventional Programming vs ML Programming

In conventional programming, programs are created manually by providing input data and based on the programming logic, and the computer generates the output.

In machine learning programming, **the input and output data are fed to the algorithm, creating the program**



Kinds of ML Tasks



Kinds of ML Tasks

- o Learning Association
 - o Classification
 - o Classification
 - o Prediction
 - o Pattern recognition
 - o Optical Character Recognition (OCR)
 - o Hand Writing Recognition
 - o Face Recognition
 - o Medical Diagnosis
 - o Speech Recognition
 - o Biometrics
 - o Knowledge Extraction
 - o Compression
 - o Outlier Detection
-
- o Regression

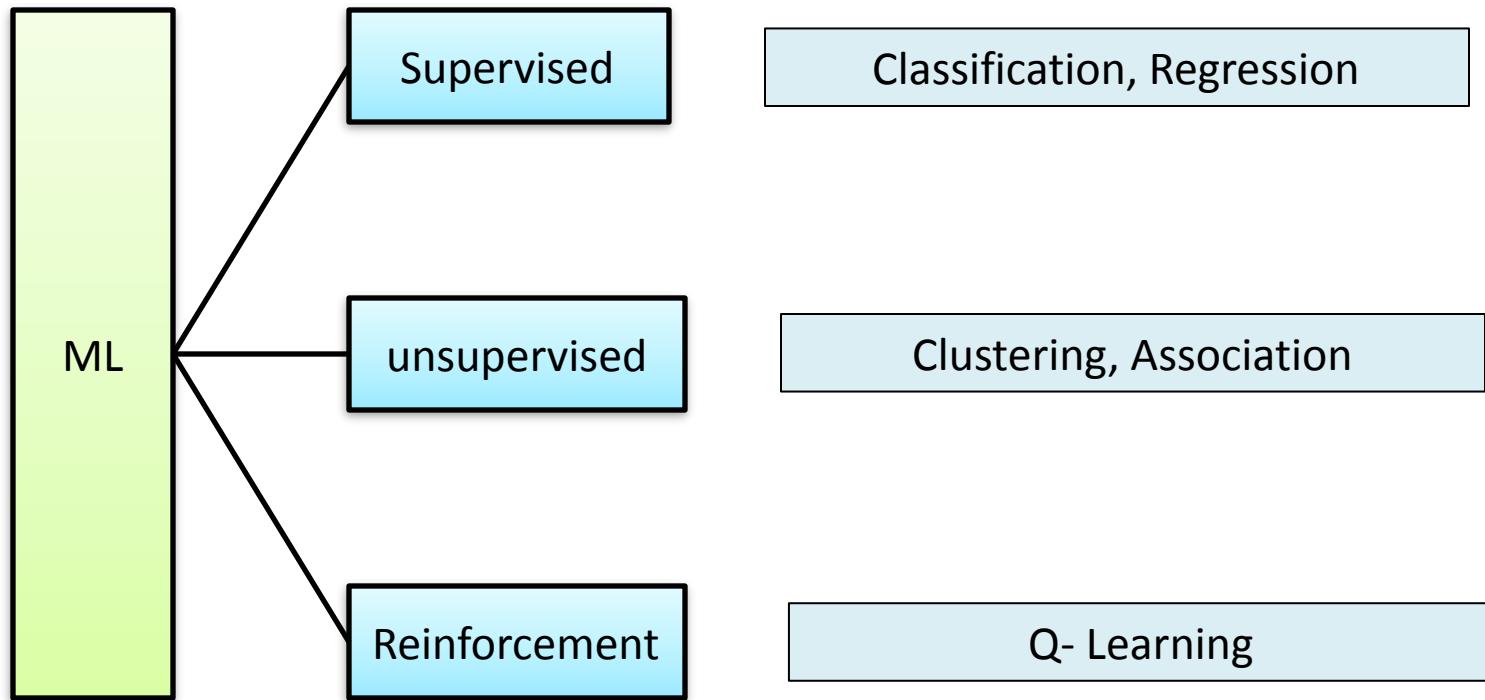
Examples of ML applications

Some most trending real-world applications of Machine Learning:

- **Image Recognition** (face detection)
- **Speech Recognition** (Speech to text)
- **Traffic prediction** (Google Map)
- **Product recommendations** (Amazon, Netflix, etc.)
- **Self-driving cars** (Deep learning is used)
- **Email Spam and Malware Filtering** (Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier)
- **Online Fraud Detection** (Feed Forward Neural network)
- **Stock Market trading** (Recurrent neural network like LSTM)
- **Weather prediction** (rain fall prediction etc. Recurrent neural network like LSTM)
- **Medical Diagnosis** (finding brain tumors; Deep learning)
- **Automatic Language Translation** (GNMT (Google Neural Machine Translation))

Classification of ML

Classification of ML Algorithm



Classification of

ML Supervised

Supervised Learning (SL) is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.

Advantage:

- Huge number of application
- Performance is good

Disadvantages:

- **Slow** (it requires human experts to manually label training examples one by one)
- **Costly** (a model should be trained on the large volumes of hand-labeled data to provide accurate predictions)

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Classification of

ML
Supervised

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- **Artificial Neural Network (ANN)**
- K- Nearest Neighbour (K-NN)
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Classification of

ML Unsupervised

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets.

Advantages:

- solves the problem by learning the data without any labels.
- It is very helpful in finding patterns in data, which are not possible to find using normal methods.
- There is lesser complexity compared to the supervised learning task. Here, no one is required to interpret the associated labels and hence it holds lesser complexities.
- It is reasonably easier to obtain unlabeled data.

Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

Classification of

ML Unsupervised

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets.

Disadvantages:

- has a *limited area of applications* (mostly for clustering purposes)
- provides *less accurate results*
- might require human intervention to understand the patterns and correlate them with the domain knowledge
- cannot get precise information regarding the output

Use:

Anomaly detection, Segmentation,
Dimensionality reduction

Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

Classification of

ML
Unsupervised

- **Clustering**
K-Means
K-Mediod
CURE
BIRCH
- **Association Rule Mining**
Apriori Algorithm
Predictive Apriori Algorithm
Tertius Algorithm
Eclat
FP-Growth

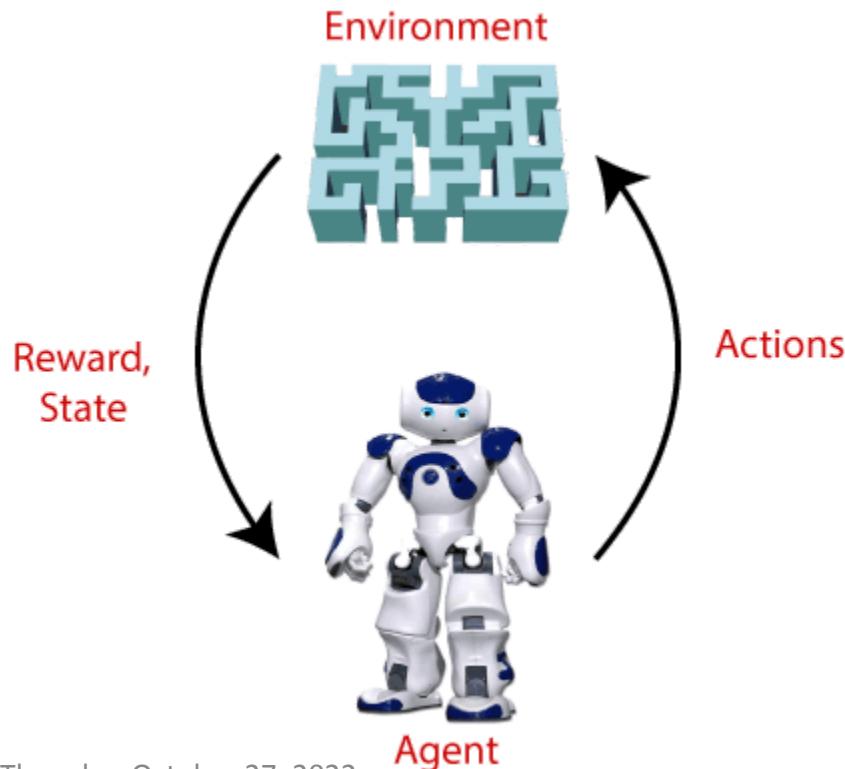
Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

Classification of

ML Reinforcement

Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones.

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. **For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty**



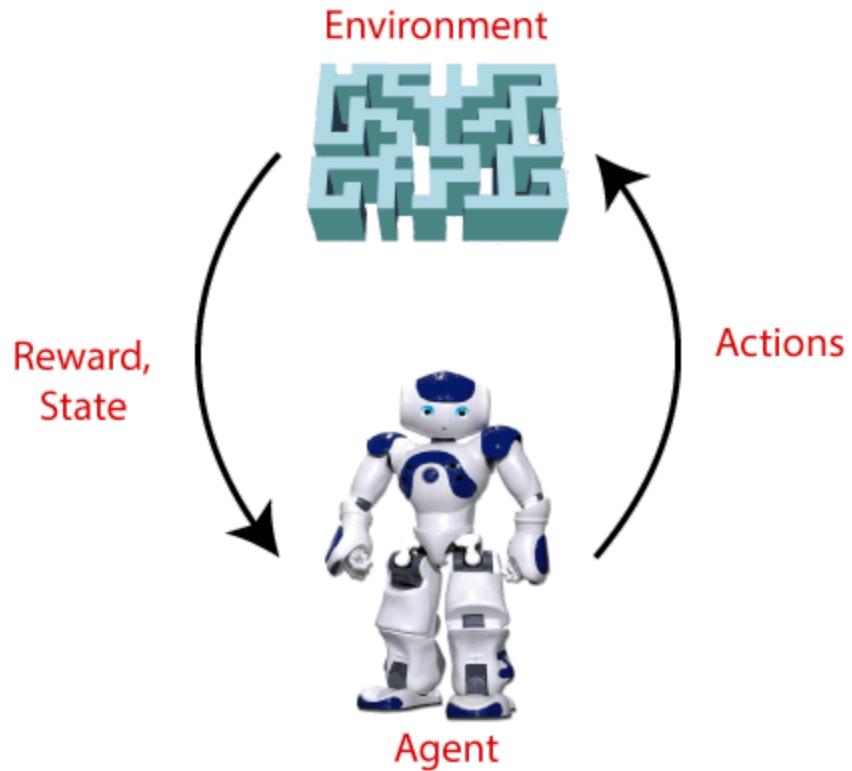
BP	Heart Beat	Weight	Feedback
120	70	50	reward
125	65	60	penalty
130	59	52	penalty
150	78	70	penalty
135	66	85	reward
125	75	82	reward
120	76	90	reward

Classification of

ML Reinforcement

Advantages:

- Reinforcement learning doesn't require large labeled datasets.
- It's **Innovative**.
- **Bias Resistance**
- **Goal-oriented**, Reinforcement learning can be used for sequences of actions.
- Reinforcement learning is **Adaptable**. Reinforcement learning doesn't require retraining because it adapts to new environments automatically on the fly.
- Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
- The model can correct the errors that occurred during the training process



Classification of

ML Reinforcement

Disadvantages:

- Can diminish the results due to too much reinforcement learning
- Not preferable to use for solving simple problems.
- Needs a lot of data and a lot of computation. It is data-hungry
- Assumes the world is Markovian, which it is not
- The curse of dimensionality limits reinforcement learning heavily for real physical systems.

Applications:

Robotics: Robot navigation, walking etc

Control: Adaptive control such as Factory processes etc.

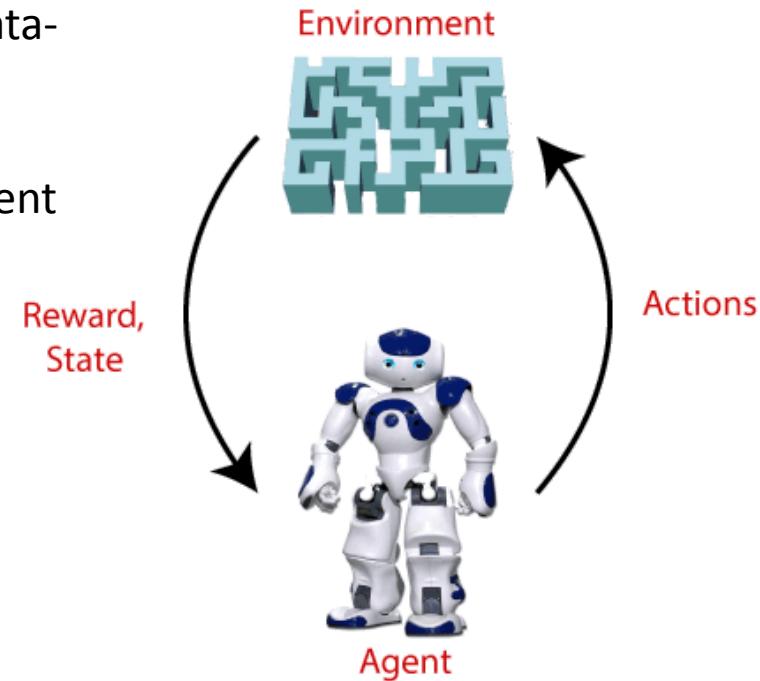
Game Playing: Game playing like chess, etc.

Chemistry: Optimizing the chemical reactions.

Business: business strategy planning

Manufacturing: automobile manufacturing companies

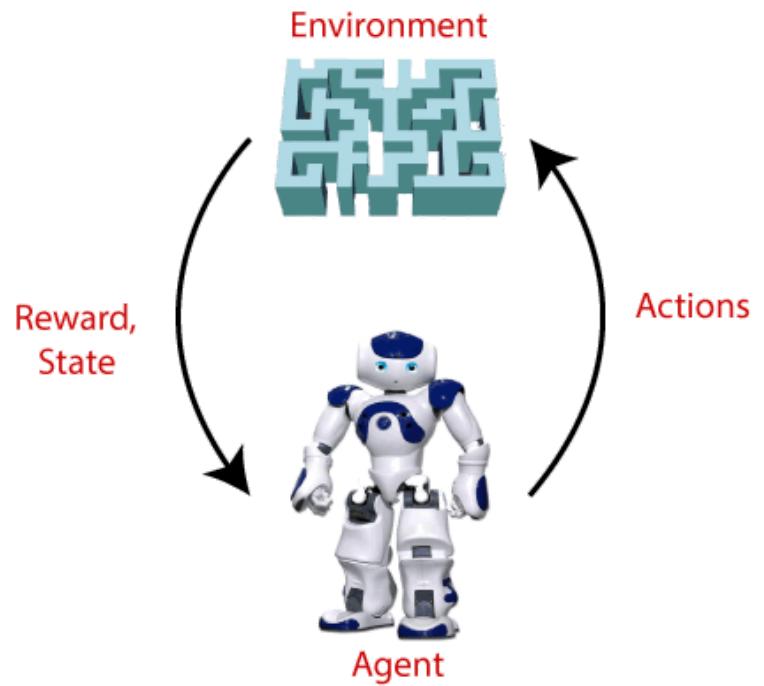
Finance Sector: finance sector for evaluating trading strategies



Classification of

ML Reinforcement

- Markov Decision Process (MDP)
- Q learning: Deep-Q-Neural Network (DQN)
- State Action Reward State Action (SARSA)



Semi-Supervised

Semi-Supervised Learning (SSL) is a learning problem that involves a small number of labeled examples and a large number of unlabeled examples.

Semi-supervised machine learning is a **combination of supervised and unsupervised machine learning methods**.

- Unlike unsupervised learning, SSL works for a variety of problems from classification and regression to clustering and association.
- Unlike supervised learning, the method uses small amounts of labeled data and also large amounts of unlabeled data, which reduces expenses on manual annotation and cuts [data preparation](#) time.

Advantages:

- Easy to understand.
- Reduces the amount of annotated data used.
- A stable algorithm
- High efficiency
- Large applications
- Work as domain expert

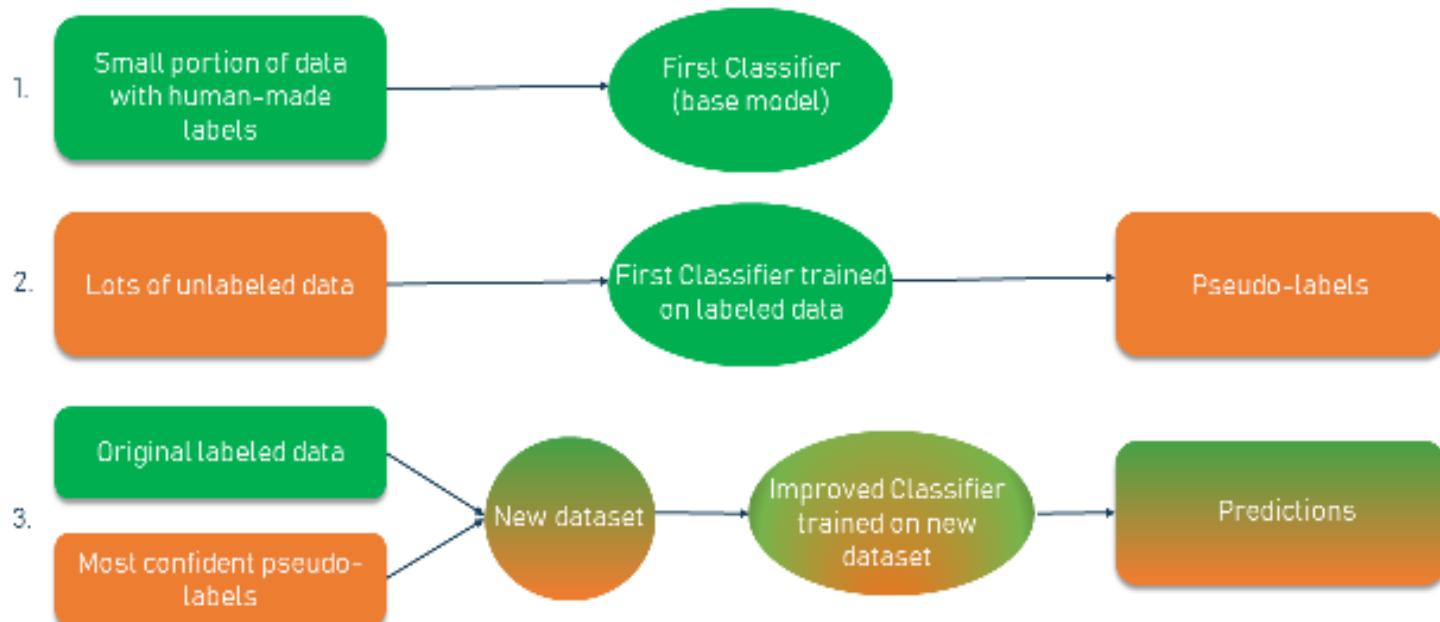
BP	Heart B	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	?
135	66	85	?
125	75	82	?
120	76	90	?

Semi-Supervised

Disadvantages:

- Low accuracy
- Results are not stable
- Not appropriate for complex problems

SEMI-SUPERVISED SELF-TRAINING METHOD

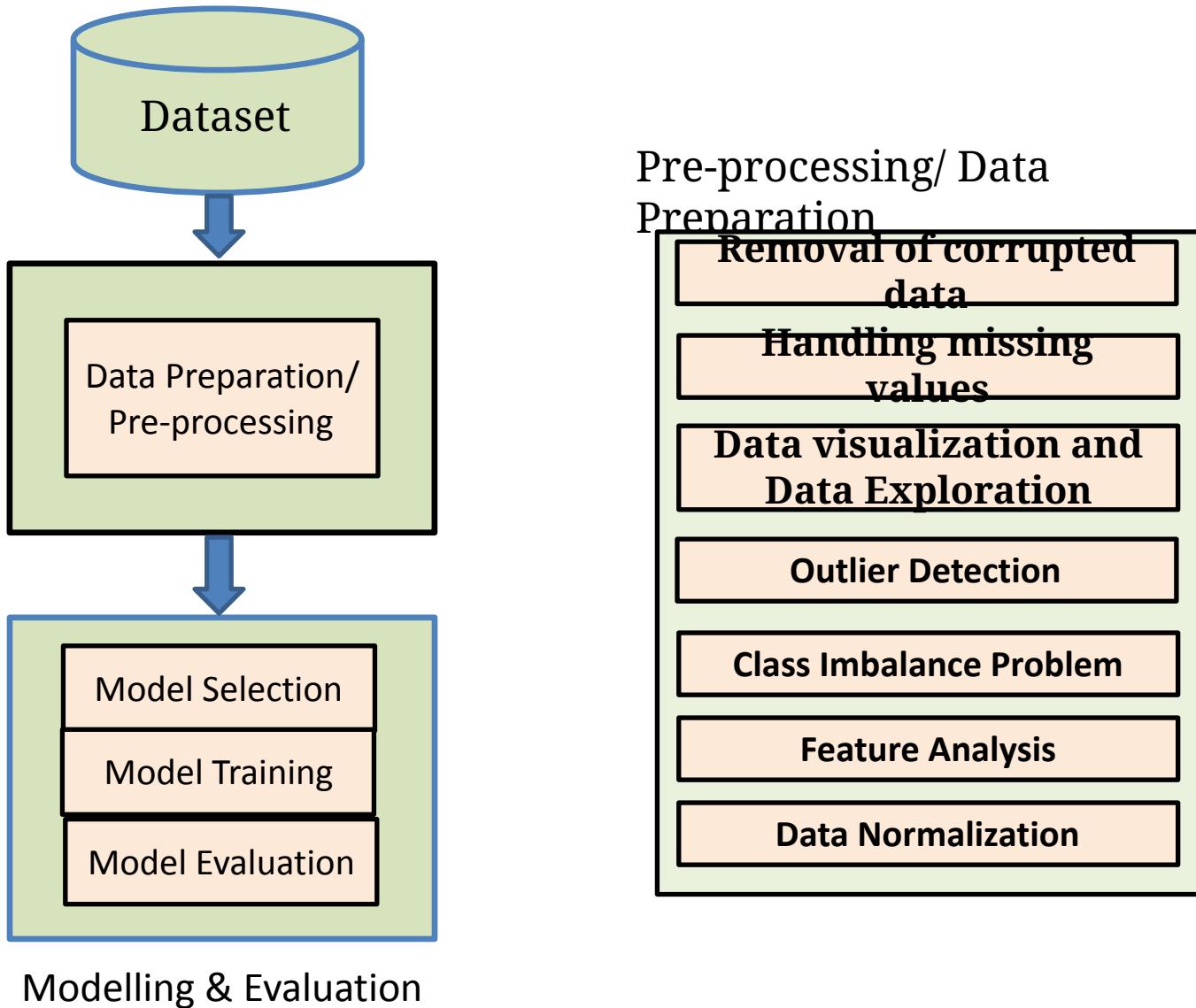


Semi-Supervised

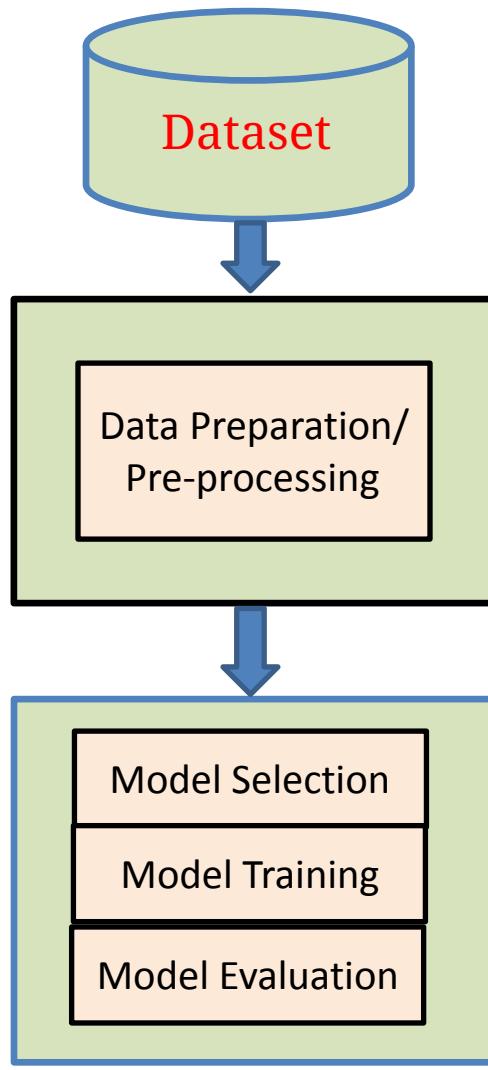
Applications:

- Text document classification
- Speech Analysis
- Protein Sequence Classification
- Internet Content Classification

ML Model

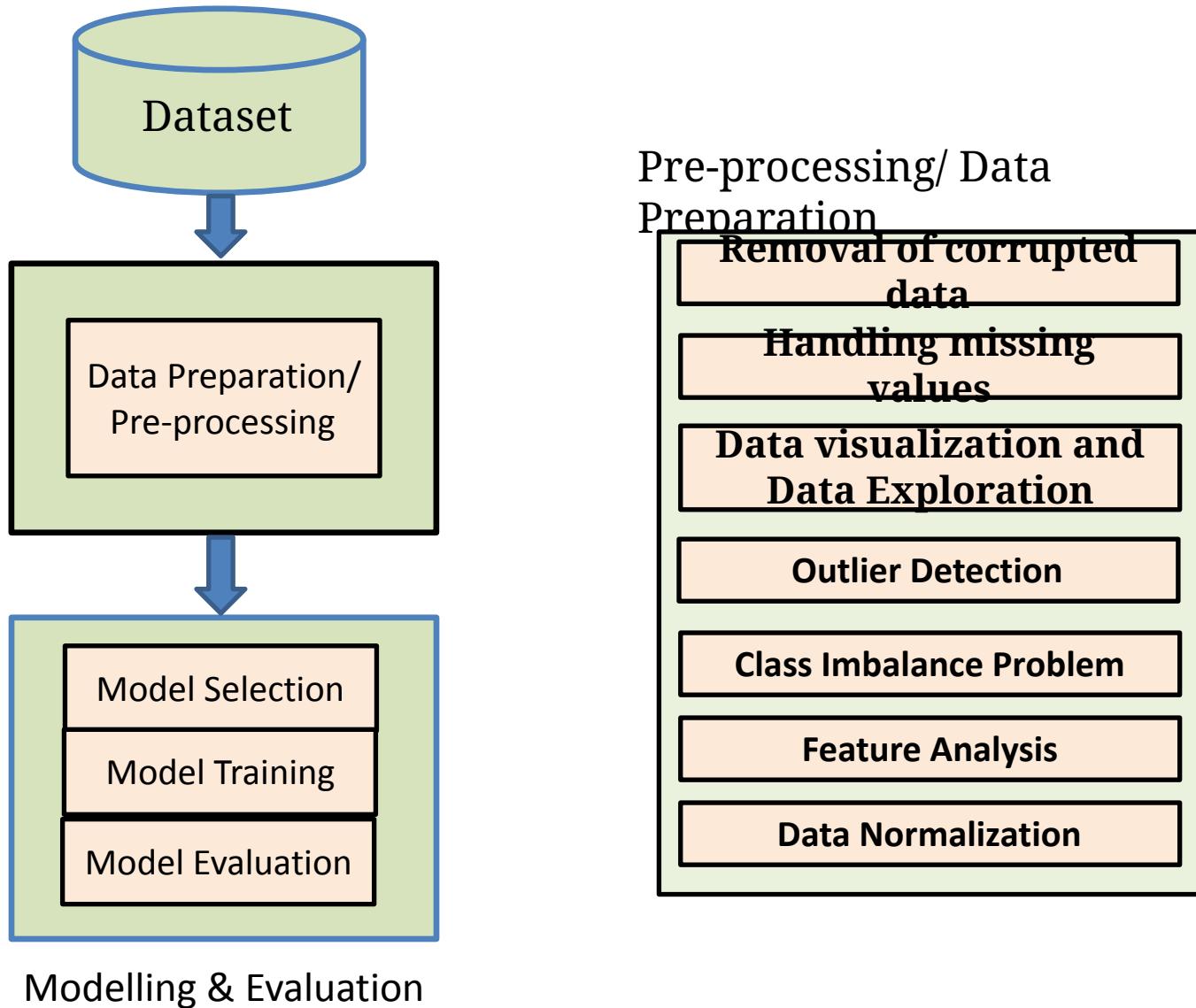


ML Model



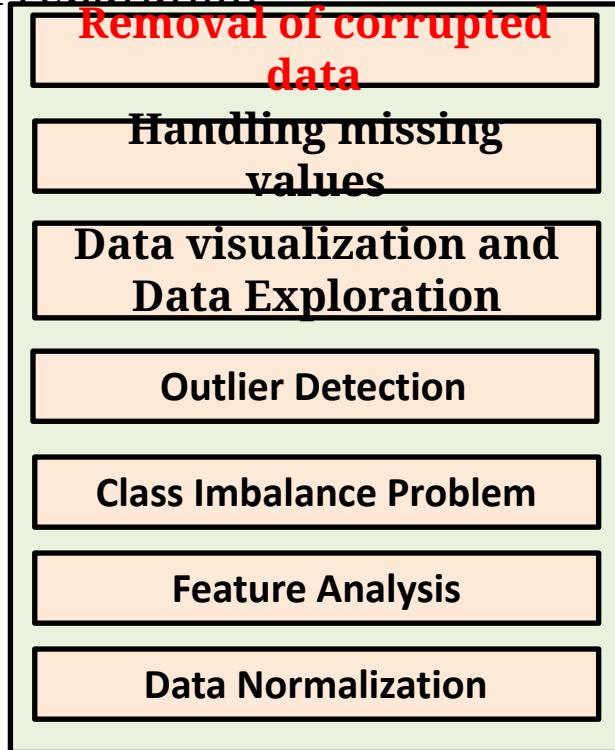
Modelling & Evaluation

ML Model



ML Model

Pre-processing/ Data Preparation



Improves quality of the training and reliability

Removal of erroneous data: not in format,
not in range, outliers, missing values

A	B	C	D	E

→

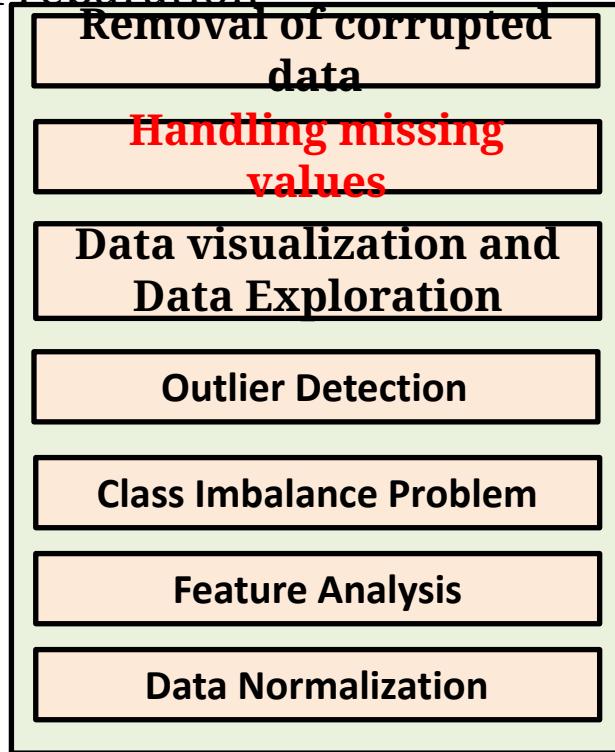
A	B	C	D

Age	weight	Height	pressure	Heart Beat	class
25	54	5.6	110	300	Y
30	64	5.7	130	400	N

Age	weight	Height	pressure	class
25	54	5.6	110	Y
30	64	5.7	130	N

ML Model

Pre-processing/ Data Preparation



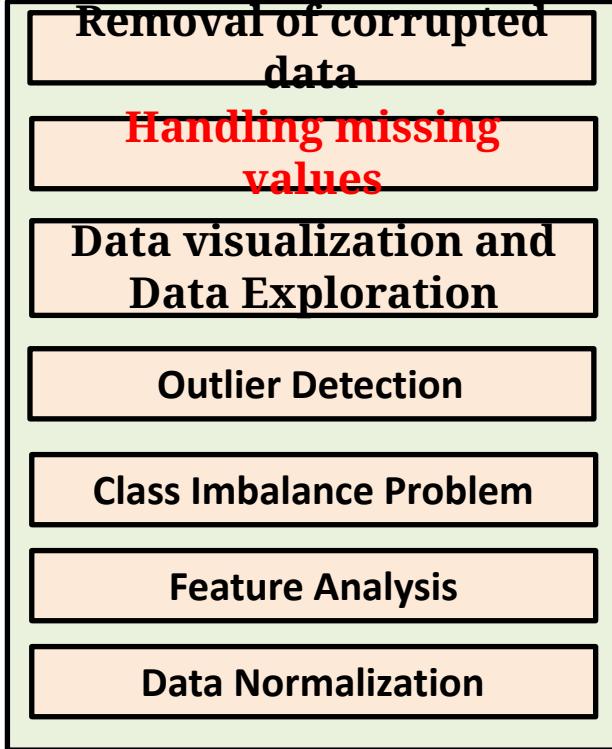
What is a Missing Value?

How is it created ?

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59		N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124	56	81	Y
115	70	73	Y

ML Model

Pre-processing/ Data Preparation



Type of Missing values:

- MCAR: Missing Completely At random
- MAR: Missing At Random
- MNAR: Missing Not At Random

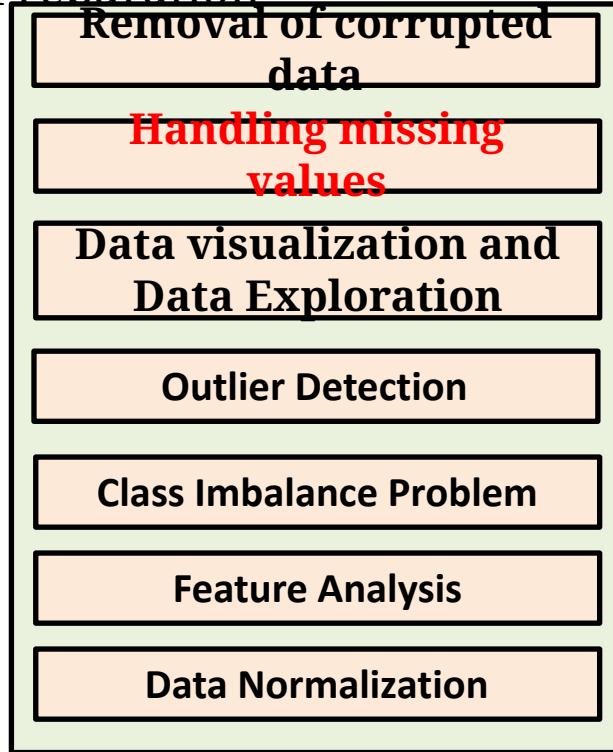
MCAR: Missing Completely At random

- If there is no relationship among the missing data and any other variable of the dataset
- Probability of missing is not related with any other variable.

Roll. No	Due book
120	05
101	
102	03
112	09
105	02

ML Model

Pre-processing/ Data Preparation



Type of Missing values:

- MCAR: Missing Completely At random
- MAR: Missing At Random
- MNAR: Missing Not At Random

MAR: Missing At Random

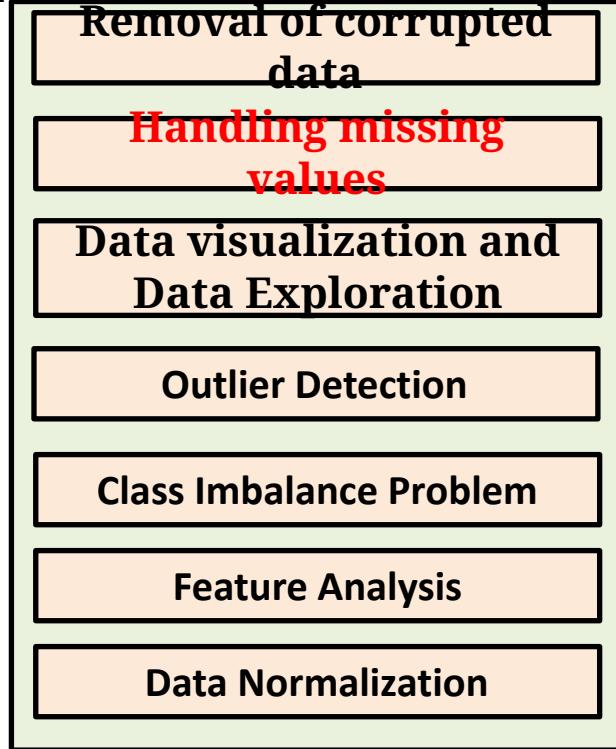
If there is a relationship among the missing data and any other variable of the dataset. Therefore need to analyze the relationship between the missing data and the variable on which it depends upon.

If the probability of being missing is the same only within groups defined by the *observed* data, then the data are missing at random (MAR). MAR is a much broader class than MCAR

Roll. No	Due book	Sex	Roll. No	Due book	Sex
120	05	M	100		M
101		F	103		M
102	03	F	115	03	F
112	09	M	111	09	F

ML Model

Pre-processing/ Data Preparation



Type of Missing values:

- MCAR: Missing Completely At random
- MAR: Missing At Random
- MNAR: Missing Not At Random

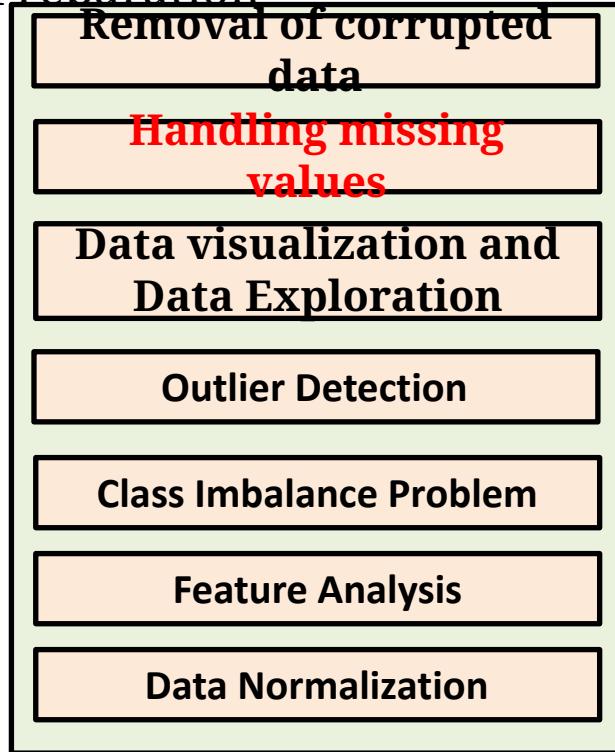
MNAR: Missing Not At Random

There is a relationship between the missing data and the variable itself in which the data is missing.
Required to proper understanding about the variable before making any imputation.

Year	No. Population	Year	No. Population
2005	1000	2010	1800
2006	1100	2011	2000
2007	1300	2012	2300
2008		2013	
2009	1600	2014	2800

ML Model

Pre-processing/ Data Preparation



Improves quality of the training

Handling Missing Values:

1. Deletion
2. Data Imputation

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59		N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124	56	81	Y
115	70	73	Y

Handling Missing Values

BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124	56	81	Y
115	70	73	Y

Deletion:

Deletion methods are used when missing is occurred due to “missing completely at random” and “missing at random”

1. Deleting Rows
2. Deleting Columns
3. Pairwise

Handling Missing Values

Deletion:

BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76	52	Y
124	56	81	Y
115	70	73	Y



BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120	66	85	N
124	75	82	N
154	76	52	Y
124	56	81	Y
115	70	73	Y

1. Deleting Rows
2. Deleting Columns
3. Pairwise

Handling Missing Values

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124		81	Y
115	70	73	Y



BP	Weight	Class
100	50	Y
101	60	Y
102	52	N
120	70	N
120	85	N
124	82	N
154	52	Y
124	81	Y
115	73	Y

Deletion:

- 1. Deleting Rows**
- 2. Deleting Columns**
- 3. Pairwise**

Handling Missing Values

Deletion:

BP	Heart Beat	Weight	Class
100	70	50	Y
101		60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124		81	Y
115	70	73	Y



BP	Weight	Class
100	50	Y
101	60	Y
102	52	N
120	70	N
120	85	N
124	82	N
124	81	Y
115	73	Y

Pairwise correlation between predictor and target is found to help in deletion

BP-Class-----High

Heart Beat—Class----Low

Weight-Class-----High

Handling Missing Values

BP	Heart Beat	Weigh t	Class
100	70	50	Y
101		60	Y
102	59	52	N
120		70	N
120	66	85	N
124	75	82	N
154	76		Y
124		81	Y
115	70	73	Y



BP	Heart Beat	Class
100	70	Y
102	59	N
120	66	N
124	75	N
154	76	Y
115	70	Y

Deletion:

1. Deleting Rows
2. Deleting Columns
3. Pairwise

Pairwise correlation between predictor and target is found to help in deletion

BP-Class-----High

Heart Beat—Class----High

Weight-Class-----Low

Handling Missing Values

Deletion:

Pros: Trained model becomes robust as all the missing values are deleted.

Cons: 1. loss of information and 2. trained model works poorly if deletion is excessive

Handling Missing Values

BP	Heart Beat	Weight	Class
100	70	50	Y
101	65	60	Y
102	59	52	N
120	67	70	N
120	66	85	N
124	75	82	N
154	76	52	Y
124	56	81	Y
115	70	73	Y

Imputation:

1. Mean
 2. Median
 3. Mode
 4. Linear Interpolation
 5. Linear Regression
 6. K-NN
- Different ML techniques

$$\begin{aligned}\text{Mean} &= (70+65+59+66+75+76+56+70)/8 \\ &= 67.125 \\ &= 67\end{aligned}$$

Handling Missing Values

BP	Heart Beat	Weight	Class
100	Y	50	Y
101	Y	60	Y
102	N	52	N
120	Y	70	N
120	Y	85	N
124	Y	82	N
154	Y	52	Y
124	Y	81	Y
115	N	73	Y

Imputation:

1. Mean
2. Median
3. Mode
4. Linear Interpolation
5. Linear Regression (ML)
6. K-NN (ML)
7. SoftImpute (ML)
8. Mice (ML) (good)
9. MatrixFactorization (ML)
10. miss- Forest (ML)
11. Deductive Imputation (LR)
12. Factor Analysis of Mixed Data (FAMD) (ML)

Categorical Data

K-NN is used value of k = 4

Handling Missing Values

Last Observation Carried Forward (LOCF)

If data is time-series data, one of the most widely used imputation methods is the last observation carried forward. Whenever a value is missing, it is replaced with the last observed value.

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	90	86%
6	6-Jan	155	87%
7	7-Jan	155	89%
8	8-Jan	155	90%
9	9-Jan	180	92%

Handling Missing Values

Next Observation Carried Backward (NOCB)

It is a similar approach like LOCF which works oppositely by taking the first observation after the missing value and carrying it backward ("next observation carried backwards", or NOCB).

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	155	86%
6	6-Jan	155	87%
7	7-Jan	180	89%
8	8-Jan	180	90%
9	9-Jan	180	92%

Handling Missing Values

Linear Interpolation

It is a mathematical method that adjusts a function to data and uses this function to extrapolate the missing data. **The simplest type of interpolation is linear interpolation, where the values before the missing data and after the same is used.**

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%



Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	120	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	170	90%
9	9-Jan	180	92%

$$(90+150)/2 = 120$$

$$(160+180)/2 = 170$$

Handling Missing Values

Adding a category to capture NA

This is perhaps the most widely used method of missing data imputation for categorical variables.

This method consists of treating missing data as an additional label or category of the variable.

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	N/A	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	N/A	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	N/A	180	95%



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Missing	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	Missing	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	Missing	180	95%

Handling Missing Values

Frequent category imputation

Replacement of missing values by the most frequent category is the equivalent of mean/median imputation. It consists of replacing all occurrences of missing values within a variable with the variable's most frequent label or category.

Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	N/A	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	N/A	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	N/A	180	95%



Mobile ID	Mobile Package	Download Speed	Data Limit Usage
1	Fast+	157	80%
2	Fast+	99	70%
3	Fast+	167	10%
4	Fast+	90	80%
5	Lite	76	70%
6	Fast+	155	10%
7	Fast+	200	95%
8	Lite	76	77%
9	Fast+	180	95%

Handling Missing Values

Missing Value Treatment using most recent data imputation techniques

MICE (Multiple Imputation by Chained Equation)

Handling Missing Values

Multiple Imputation

Multiple Imputation (MI) is a statistical technique for handling missing data.

The key concept of MI is to use the distribution of the observed data to estimate a set of plausible values for the missing data.

Estimates are combined to obtain a set of parameter estimates.

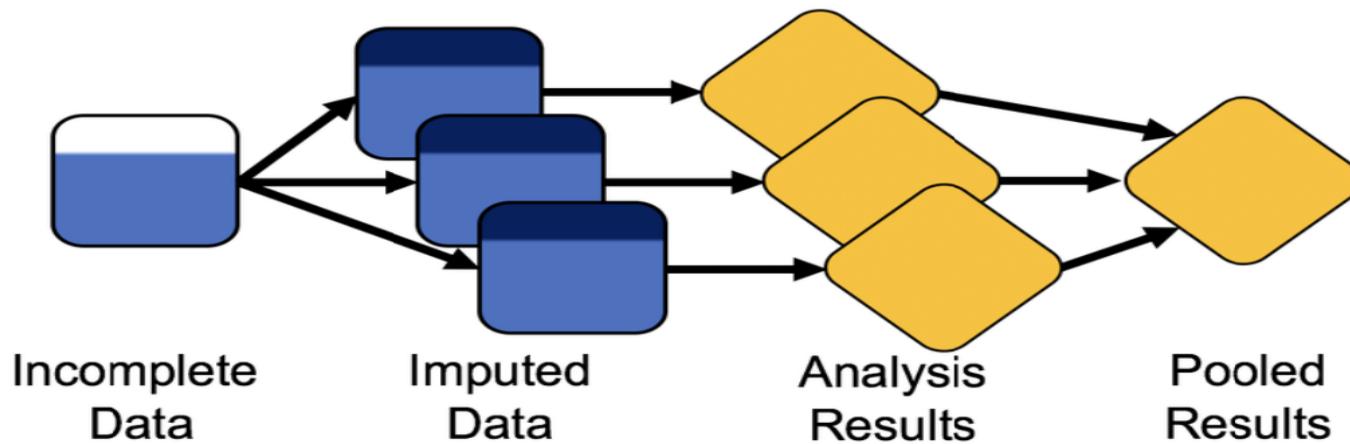
Multiple datasets are created and then analysed individually but identically to obtain a set of parameter estimates.

Multiple Imputation by Chained Equations (MICE) approach is a flexible way of handling more than one missing variable,

The benefit of the multiple imputations is to restore the natural variability of the missing values.

Handling Missing Values

Multiple Imputation



Multiple Imputation

First step would be to remove the "Personal Loan" column as it is the target column, we will not need this column for imputation.



age	experience	salary(K)	Personal loan
25		50	1
27	3		1
29	5	80	0
31	7	90	0
33	9	100	1
	11	130	0



age	experience	salary(K)
25		50
27	3	
29	5	80
31	7	90
33	9	100
	11	130

Multiple Imputation

Second step would be a simple imputation, such as imputing the mean, which is performed for every missing value in the dataset that leads to the formation of zeroth dataset.

age	experience	salary(K)
25		50
27	3	
29	5	80
31	7	90
33	9	100
	11	130



age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
29	11	130

zeroth dataset

Multiple Imputation

Third step would be to remove the "age" imputed values and keep the imputed values in other columns as shown here. Now, we will be
 imputing the columns from left to right.

age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
29	11	130



age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
	11	130

Multiple Imputation

In the fourth step, the remaining features and rows (top 5 rows of experience and salary) become the feature matrix (purple cells), "age" becomes the target variable (yellow cells).

We will run the linear regression model on the fully filled rows with X= experience and salary and Y=age. To estimate the missing age, we will use the missing value row (white cells) as the test data. So, top 5 rows will be training data and the last row that has missing age will be test data. We will use (experience = 11 and salary = 130) to predict corresponding "age" value. When I did this, I found that my model predicted the age as 34.99.

age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100
	11	130

Multiple Imputation

In the fifth step, we update the predicted age value in the missing cell in "age" column.

Now, remove "experience" imputed value. The remaining features and rows becomes the feature matrix(purple cells) and "experience" becomes the target variable(yellow cells). We will run the linear regression model on the fully filled rows with X= age and salary and Y=experience. To estimate the missing experience, we will use the missing value row (white cells) as the test data. The predicted value for experience is 0.98.

age	experience	salary(K)
25		50
27	3	90
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

In the sixth step, we update the predicted experience value in the missing cell in "experience" column and remove "salary" imputed value.

The remaining features and rows becomes the feature matrix(purple cells) and "salary" becomes the target variable(yellow cells). We will run the linear regression model on the fully filled rows with $X = \text{age}$ and experience and $Y = \text{salary}$. To estimate the missing salary, we will use the missing value row (white cells) as the test data. The predicted value for Salary is 70.

age	experience	salary(K)
25	0.98	50
27	3	
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

Now we **impute the missing values in the original dataset**

and the predicted values after 1st iteration is shown here.

Let's name this as "**First dataset**".

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

In the seventh step, We will subtract the two datasets (zeroth and first). The resultant dataset is as below:

The diagram illustrates the subtraction of two datasets. On the left, there is a table with 6 rows and 3 columns: age, experience, and salary(K). The middle column (experience) is highlighted in cyan. Below this table is the word "minus". To the right of "minus" is another table with 6 rows and 3 columns, also with the middle column (experience) highlighted in cyan. A red arrow points from the right side of the second table to the right side of the third table, which shows the result of the subtraction.

age	experience	salary(K)
25	7	50
27	3	90
29	5	80
31	7	90
33	9	100

minus

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100

age	experience	salary(K)
0	6.02	0
0	0	20
0	0	0
0	0	0
0	0	0
-5.99	0	0

If we observe, the absolute difference between 2 datasets are higher in few imputed values. Our aim is to reduce these differences close to 0. To achieve this we have to do many iterations. So, now we repeat the steps 2-6 with the new dataset (first), until we get a stable model. i.e. until the difference between the 2 latest imputed datasets becomes very small, close to 0. Technically, we stop the iterations when a pre-defined threshold is reached or a pre defined maximum number of iterations gets completed.

Multiple Imputation

Now we will use the "first" dataset as our base dataset to do imputations, and discard the "Zeroth" dataset which had the mean imputations. With "first" dataset as base, let's perform all the previous steps and again predict the imputed values for the initial 3 missing values.

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

Multiple Imputation

Here's is the iteration 2 values and the new dataset values are subtracted from the first dataset and got the difference matrix as below:

Iteration 2

age	experience	salary(K)
25	0.98	50
27	3	70
29	5	80
31	7	90
33	9	100
34.99	11	130

After all imputations



age	experience	salary(K)
25	0.975	50
27	3	70
29	5	80
31	7	90
33	9	100
34.95	11	130

After
Second - First



age	experience	salary(K)
0	0.005	0
0	0	0
0	0	0
0	0	0
0	0	0
0.004	0	0

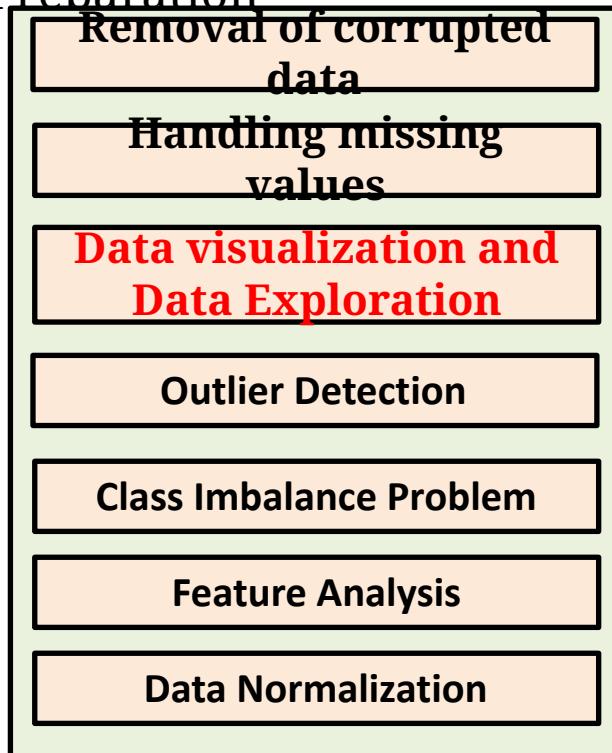
First Dataset

Second Dataset

Difference Matrix

Now, after second iteration, we can see that the difference is very negligible. We can either stop here as we almost got the same numbers, or proceed with next iteration until we get 0 difference.

Pre-processing/ Data Preparation



Helps to understand underlying behaviour of data which helps to take right steps in data preparation and modelling.

Data Visualization and Data Exploration

Data Exploration:

- Mean (central tendency)
- Median (central tendency)
- Variance (Data Spread)

Set of observation=21 89 34 67 96

$$\text{Mean} = (21+89+34+67+96)/5 = 61.4$$

21 34 67 89 96; Median = 67

A1: 44, 46, 48, 45, 47

A2: 34, 46, 59, 39, 52

For both mean and median 46

To measure data dispersion or data spread, variance is measured

variance(A1)= 2; variance(A2)= 79.6

A1 values are concentrated around mean

A2 values are extremely spread out

Set of observation= 21, 20, 23, 24, 25 84 67, 55 96

$$\text{Mean} = (21+20+23+24+25+84+67+55+96)/9 = 46.11$$

20, 21, 23, 24, 25, 55, 67, 84, 96; Median= 25

Set of observation= 21 89 34 67 200

$$\text{Mean} = (21+89+34+67+200)/5 = 82.2$$

21 34 67 89 200; Median = 67

Data Visualization and Data Exploration

Data Visualization:

Box Plot: An effective mechanism to get a one-shot view and understand the nature of data.

It gives a standard visualization of five statistical summary: **minimum**, **first quartile(Q1)**, **median(Q2)**, **third quartile(Q3)** and **maximum**.

Box spans from Q1 to Q3 = **Inter-Quartile Range (IQR)**

Lower Range (LR) extends up to = (Q1 - 1.5 times of IQR)

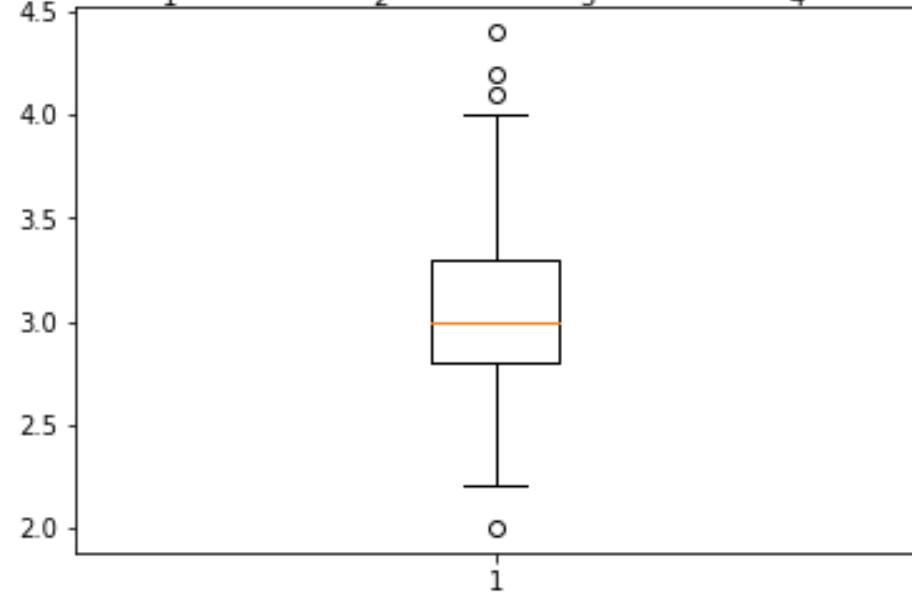
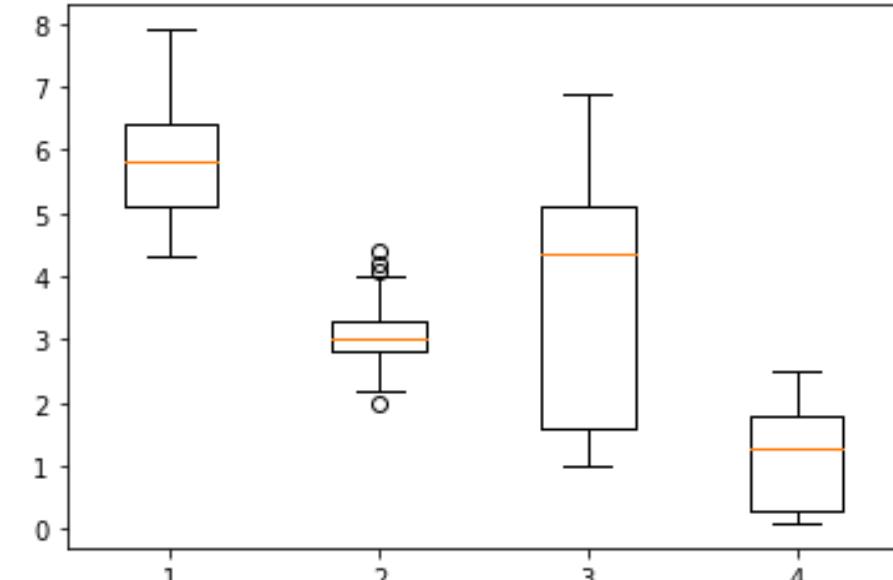
For some x: Q1=73, Q2=76 and Q3=79

$$\text{IQR}=(\text{Q3}-\text{Q1})=(79-73)=6$$

$$\text{LR}=(\text{Q1}-1.5 \times \text{IQR})=(73-1.5 \times 6)=(73-9)=64$$

Say some lower data values of x: 70, 63, 60

Minimum= 70 which is larger than 64



Data Visualization and Data Exploration

Data Visualization:

Upper Range (UR) extends up to = (Q3 + 1.5 times of IQR)

For some x: Q1=73, Q2=76 and Q3=79

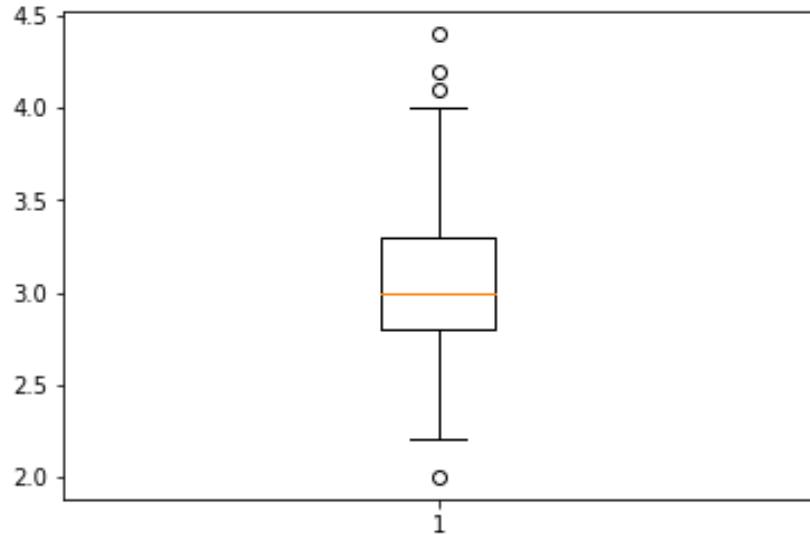
IQR=(Q3-Q1)=(79-73)= 6

UR=(Q3+1.5*IQR)=(79+1.5*6)=(79+9)=
88

Say some upper values x: 82, 84, 89

Maximum= 84 which is highest value lower than 88.

x	Frequency	C Frequency
3	4	4
4	204	208 (=4+204)
5	3	211
6	84	295
7	0	295
8	103	398



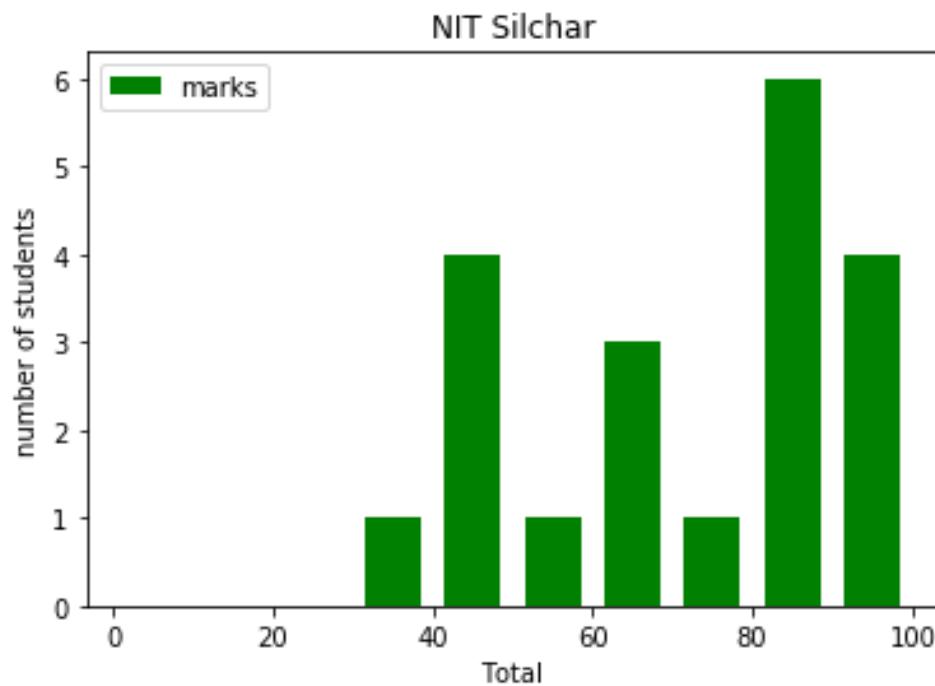
	Fequency/observation	x
Q1	Avg of 99 th and 100th	4
Q2	199	4
Q3	Avg of 298 th and 299th	8
IQR	(Q3-Q1)	4
LR	Q1-1.5*4=4-6	-2
Min		3
UR	Q3+1.5*4=8+6	14
Max		8

It can finds outliers.

Data Visualization and Data Exploration

Data Visualization:

- **Histogram (ranges of data values):** helps in understanding the distribution of a numeric data into series of intervals. Gives us quick understanding of the data.

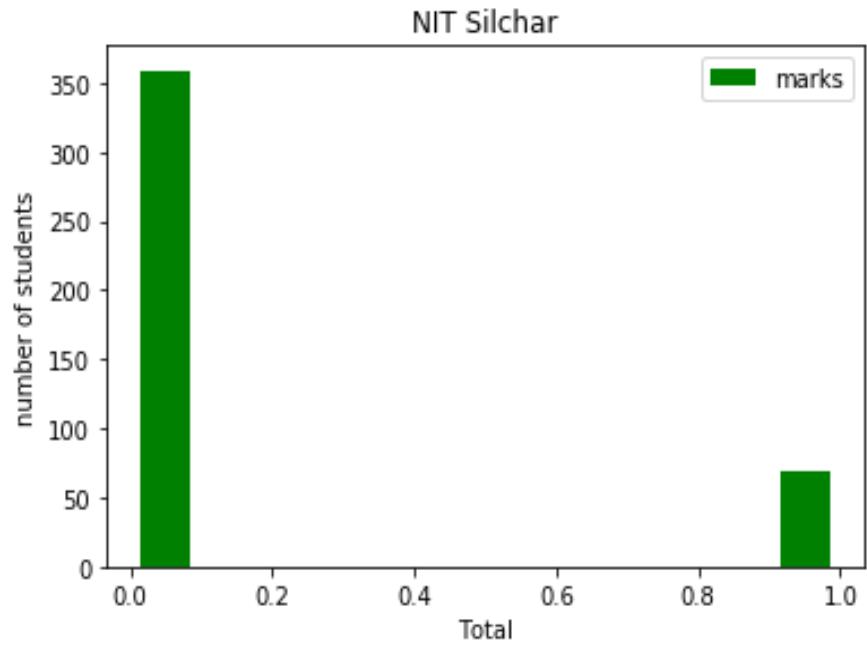
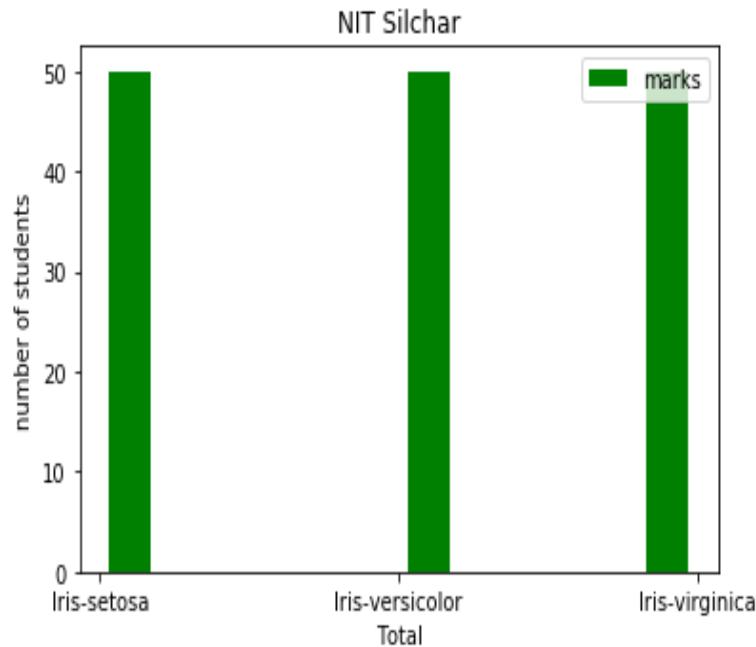


Total
52
45
65
45
68
98
46
88
36
45
86
87
68
86
94
74
93
89
90
86

Data Visualization and Data Exploration

Data Visualization:

- **Histogram:** Gives us quick understanding of the data.

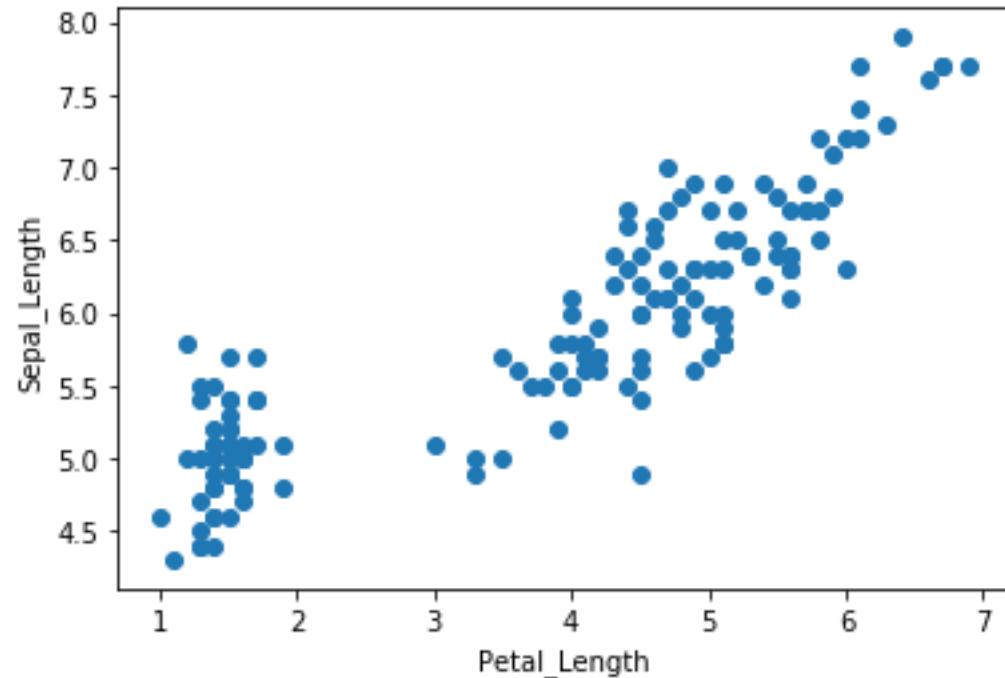


Data Visualization and Data Exploration

Data Visualization:

- Scattered plot: shows relationship between two variables

IRIS DATASET: ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width', 'Species'],



Python Libraries for Machine Learning

1. NumPy (Numerical Python):

- It is array-processing package
- It is used to process large multi-dimensional arrays and matrices
- It is used for handling linear algebra, Fourier transforms, and random numbers.
- Other libraries like TensorFlow uses NumPy at the backend for manipulating tensors.,
- With NumPy, we can define arbitrary data types and easily integrate with most databases.

Some important type of function under NumPy:

i. NumPy **Array Manipulation functions**

```
import numpy
```

```
arr1 = numpy.arange(4)
print('Elements of an array1:\n',arr1)
```

```
res1 = arr1.reshape(2,2)
print('Reshaped array with 2x2 dimensions:\n',res1)
```

Python Libraries for Machine Learning

Some important type of function unders NumPy:

i. NumPy Array Manipulation functions

```
import numpy
```

```
concat = numpy.concatenate((arr1,arr2),axis=1)  
print(concat)
```

ii. NumPy String functions

`numpy.char.add()` function: Concatenates data values of two arrays, merges them and represents a new array as a result.

`numpy.char.capitalize()` function: It capitalizes the first character of the entire word/string.

Python Libraries for Machine Learning

ii. NumPy String functions

numpy.char.lower() function: Converts the case of the string characters to lower string.

numpy.char.upper() function: Converts the case of the string characters to upper string.

numpy.char.replace() function: Replaces a string or a portion of string with another string value.

iii. NumPy Arithmetic functions

numpy.add() function : It adds two arrays and returns the result.

numpy.subtract() function : Subtracts elements of array2 from array1 and returns the result.

numpy.multiply() function : Multiplies the elements of both the arrays and returns the product.

numpy.divide() function : Divides the array1 by array2 and returns the quotient of array values.

numpy.mod() function: Performs modulus operation and returns the remainder array.

numpy.power() function: Returns the exponential value of $\text{array1} ^ \text{array2}$.

Python Libraries for Machine Learning

iv. NumPy Statistical functions

`numpy.median()` : Calculates the median value of the passed array.

`numpy.mean()` : Returns the mean of the data values of the array.

`numpy.average()` : It returns the average of all the data values of the passed array.

`numpy.std()` : Calculates and returns the standard deviation of the data values of the array.

Python Libraries for Machine Learning

2. Pandas: Pandas are turning up to be the most popular Python library that is used for data analysis.

- **The two main types of data structures used by pandas are :** Series (1-dimensional)
- DataFrame (2-dimensional)
- These two put together can handle a vast majority of data: sectors like science, statistics, social, finance, and of course, analytics and other areas of engineering.
- Tabular data with columns of heterogeneous data.
- Ordered and unordered time series data.
- Arbitrary matrix data with the homogeneous or heterogeneous type of data in the rows and columns
- Any other form of statistical or observational data sets.

Some important type of function unders Pandas:

1. `read_csv()`: `read_csv()` function helps read a comma-separated values (csv) file into a Pandas DataFrame. It can also read files separated by delimiters other than comma, like | or tab

2. `head()`: `head(n)` is used to return the first n rows of a dataset. By default, `df.head()` will return the first 5 rows of the DataFrame

Python Libraries for Machine Learning

Some important type of function under Pandas:

3. describe(): **describe() is used to generate descriptive statistics of the data in a Pandas DataFrame or Series.** It summarizes central tendency and dispersion of the dataset. describe() helps in getting a quick overview of the dataset.

4. memory_usage(): memory_usage() returns a Pandas Series having the memory usage of each column (in bytes) in a Pandas DataFrame.

```
data_1.memory_usage(deep=True)
```

5. astype(): astype() is used to cast a Python object to a particular data type.

6. loc[:]: loc[:] helps to access a group of rows and columns in a dataset, a slice of the dataset, as per our requirement.

7. to_datetime(): to_datetime() converts a Python object to datetime format.

8. value_counts(): value_counts() returns a Pandas Series containing the counts of unique values.

9. drop_duplicates(): drop_duplicates() returns a Pandas DataFrame with duplicate rows removed.

```
data_1.drop_duplicates(inplace=True)
```

10. groupby(): groupby() is used to group a Pandas DataFrame by 1 or more columns, and perform some mathematical operation on it.

Python Libraries for Machine Learning

Some important type of function under Pandas:

```
data_1.groupby(by='State').Salary.mean()
```

11. merge(): merge() is used to merge 2 Pandas DataFrame objects or a DataFrame and a Series object on a common column (field)

12. sort_values(): sort_values() is used to sort column in a Pandas DataFrame (or a Pandas Series) by values in ascending or descending order.

```
data_1.sort_values(by='Name', inplace=True)
```

13. fillna(): fillna() helps to replace all NaN values in a DataFrame or Series by imputing these missing values with more appropriate values.

```
data_1['City temp'].fillna(38.5, inplace=True)
```

14. Shape: property will return a tuple of the shape of the data frame.

```
f1.shape
```

15. f1.columns: will give you the column values

16. f1.tail():

17. DataFrame.info(): Pandas **dataframe.info()** function is used to get a concise summary of the dataframe.

Python Libraries for Machine Learning

Some important type of function under Pandas:

18. **dtypes: (f1.dtype)** dtypes shows the data type of each column. (f1.dtype)
19. **Size: (f1.size)** Size, as the name suggests, returns the size of a dataframe which is the number of rows multiplied by the number of columns.
20. **Sample: (f1.sample(n=8))** Sample method allows you to select values randomly from a Series or DataFrame.
21. **isnull:(f1.isnull())** To handle missing values
22. **isna() : (f1.isna().any())** Isna function returns a dataframe filled with boolean values with true indicating missing values.
23. **f1.isnull().sum()** : We can calculate the number of missing values in each column
24. **nunique(): (f1. nunique())** Nunique counts the number of unique entries over columns or rows. It is very useful in categorical features especially in cases where we do not know the number of categories beforehand
25. **index() (f1.index)** searches for a given element from the start of the list and returns the lowest index where the element appears.
26. **nsmallest() (f1. nsmallest(5,'Sepal_Width'))** finds the 5 observations with the smallest value
27. **nlargest() (f1. nlargest(5,'Sepal_Width'))** finds the 5 observations with the Largest value.

Python Libraries for Machine Learning

Some important type of function unders Pandas:

28. Loc and iloc

Loc and iloc are used to select rows and columns.

loc: select by labels

iloc: select by positions

```
f1.loc[:5,['Sepal_Length', 'Sepal_Width']]  
f1.iloc[:5,:6]
```

29. Slicing: Slicing Rows and Columns using labels.

```
f1[0:4]
```

30. **dropna ()** function is used to remove a row or a column from a dataframe which has a NaN or no values in it

31. **query()**: We sometimes need to filter a dataframe based on a condition or apply a mask to get certain values.

```
f1.query('3000<median_value<1000')[:4]
```

32. **insert()** : offers the option to add the new column in any position using **insert** function

```
f1.insert(5, 'new_name', new_col)
```

Python Libraries for Machine Learning

3. Matplotlib:

- Matplotlib is a data visualization library
- It is used for 2D plotting to produce publication-quality image plots and figures in a variety of formats.
- The library helps to generate histograms, plots, error charts, scatter plots, bar charts with just a few lines of code.

4. SciPy (Scientific Python):

This is a python library for **machine learning**, especially for scientific and analytical computing.

- SciPy uses **multi-dimensional array** provided by the NumPy module.
- SciPy depends on NumPy for the array manipulation subroutines.
- The SciPy library offers **modules for linear algebra, image optimization, integration, interpolation, special functions, Fast Fourier transform, signal and image processing, Ordinary Differential Equation (ODE) solving, and other computational tasks in science and analytics.**

5. Scikit-learn:

It has become the most popular Python machine learning library for developing machine learning algorithms.

Python Libraries for Machine Learning

6. Scikit-learn:

- The library can be used for **data-mining and data analysis**.
- The main machine learning functions that the Scikit-learn library can handle are **classification, regression, clustering, dimensionality reduction, model selection, and preprocessing**.

7. Theano:

- Theano is a **python machine learning library** that can act as an optimizing compiler for evaluating and manipulating mathematical expressions and matrix calculations.
- It is built on NumPy.
- **Theano can work on Graphics Processing Unit (GPU) and CPU**.
- Theano has built-in tools for unit-testing and validation, thereby avoiding bugs and problems.

8. TensorFlow:

TensorFlow is a popular computational framework for creating **machine learning models**.

- TensorFlow has a flexible architecture with which it can run on a variety of **computational platforms CPUs, GPUs, and TPUs**.
- TPU stands for Tensor processing unit, a hardware chip built around TensorFlow for machine learning and artificial intelligence.

Python Libraries for Machine Learning

9.Keras: Keras is an open-source library used for **neural networks and machine learning**. Keras can run on top of TensorFlow, Theano etc.

- Keras works with neural-network building blocks like layers, objectives, activation functions, and optimizers.
- Keras also have a bunch of features **to work on images and text images that comes handy when writing Deep Neural Network code**.
- Keras supports convolutional and recurrent neural networks.

10. PyTorch: PyTorch has a range of tools and libraries that support **computer vision, machine learning, and natural language processing**

- PyTorch can smoothly integrate with the python data science stack, including NumPy.
- We will hardly make out a difference between NumPy and PyTorch.
- PyTorch include multi GPU support, simplified preprocessors, and custom data loaders.

11. Neurolab: **It is a simple and powerful Neural Network Library for Python.** It is a library for basic neural networks algorithms with flexible network configurations and learning algorithms for Python.

Python Module and Packages

Modules: Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*. A module is a file containing Python definitions and statements. **The file name is the module name with the suffix .py appended.**

Packages:

Packages are a way of structuring Python's module namespace by using "dotted module names". For example, **the module name A.B designates a submodule named B in a package named A.**

```
import sound.effects.echo
```

This loads the submodule **echo** from package **sound.effects**. It must be referenced with its full name.

An alternative way of importing the submodule is:

```
from sound.effects import echo
```

This also loads the submodule **echo**, and makes it available without its package prefix, so it can be used as follows:

```
echo.echofilter(input, output, delay=0.7, atten=4)
```

```
from sound.effects.echo import echofilter
```

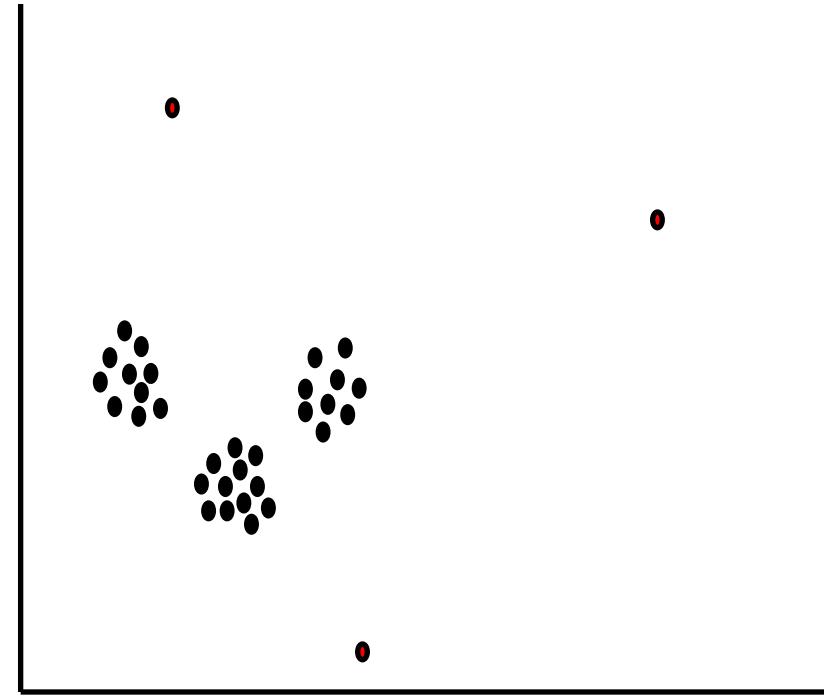
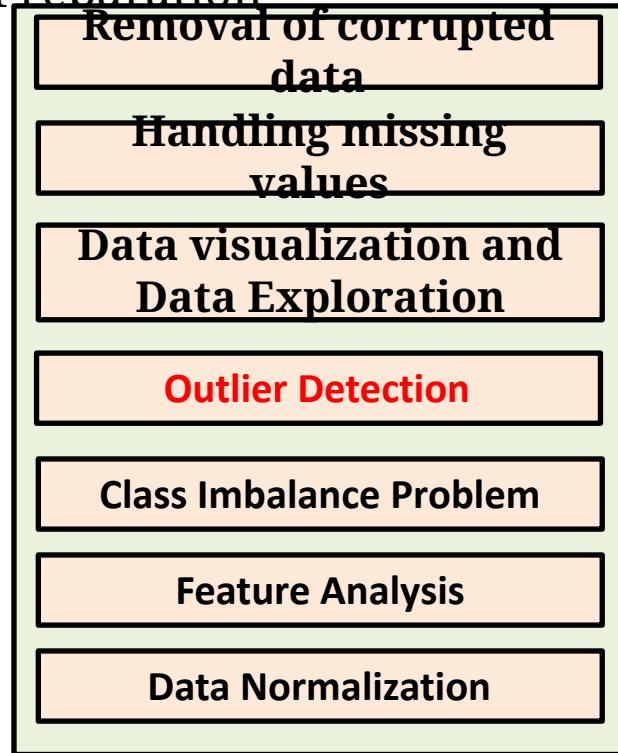
Again, this loads the submodule **echo**, but this makes its function **echofilter()** directly available:

```
echofilter(input, output, delay=0.7, atten=4)
```

DAY-2

Machine Learning Model

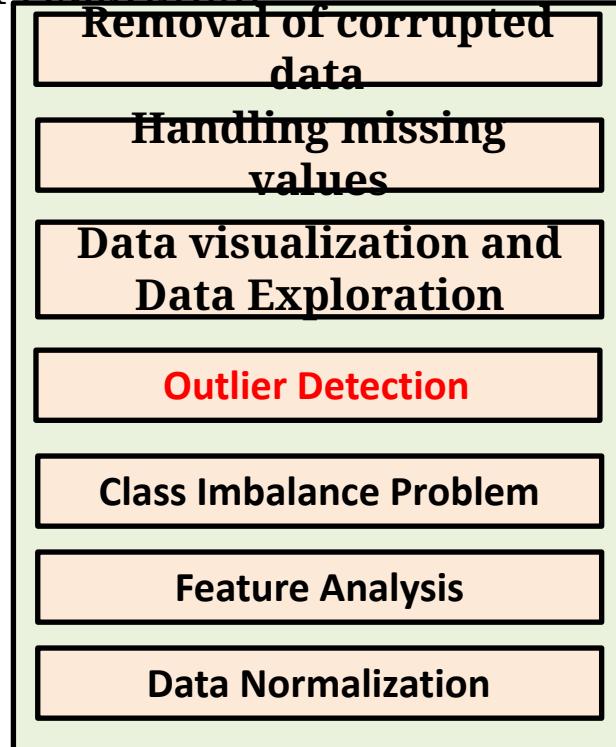
Pre-processing/ Data Preparation



Far from the rest of the observations or the center of mass of observations.

Outlier Detection

Pre-processing/ Data Preparation



Can result in a poor fit and lower predictive modelling performance.

Can fit the data properly and enhance the performance of the model

Histogram or scatter plot can be used for one or two dimensional data.

For high dimensional data, simple statistical methods for identifying outliers can break down.

Outlier Detection

Many techniques are found to detect outlier. Based on learning style, these techniques can be categorized into three classes:

- supervised outlier detection techniques,
- unsupervised outlier detection techniques.
- Semi-supervised outlier detection techniques.

Based on learning measures, these techniques can be categorized into four approaches

- Statistical based
- Distance based
- Density based
- Tree-based

LIMITATIONS OF STATISTICAL BASED APPROACH

- Works well for a single attribute
- In many cases, data distribution may not be known. However statistical based approach needs to know data distribution.
- For multi-dimensional data, it may be difficult to estimate the true distribution

Outlier Detection

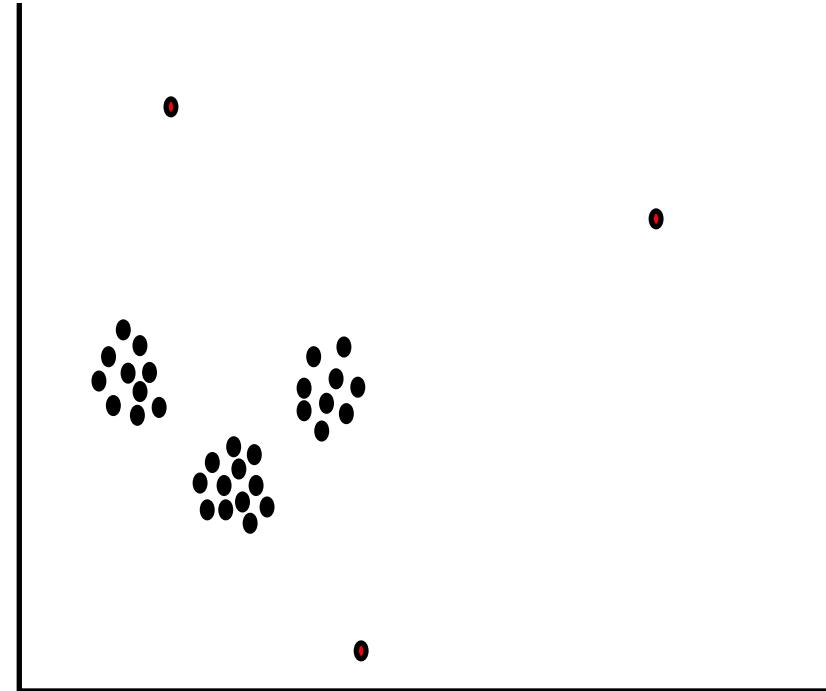
Local Outlier Factor: Each example is assigned a scoring of how isolated or how likely it is to be outliers based on the size of its local neighborhood. Examples with the largest score are more likely to be outliers. (**density based unsupervised algorithm**)

Isolation Forest: is a tree-based anomaly detection algorithm:

Outliers have attribute-values that are very different from those of normal instances. (**Tree based unsupervised algorithm**)

One Class SVM: can be used to discover outliers in input data for both regression and classification datasets. This is specially used in imbalanced dataset. (**density based unsupervised algorithm**)

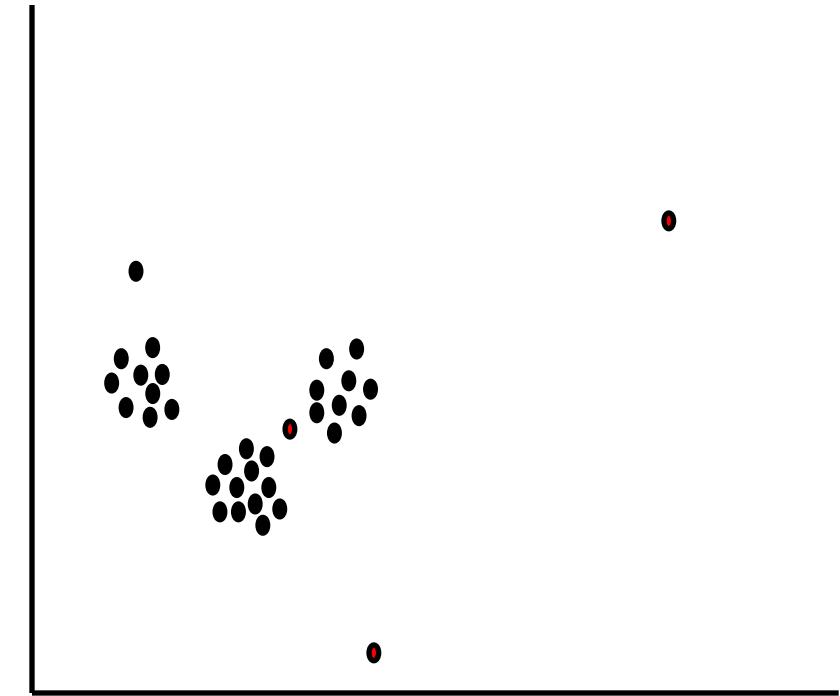
Minimum Covariance Determinant: If the input variables have a Gaussian distribution, then this simple statistical methods can be used to detect outliers. (**Statistical based unsupervised algorithm**)



Local and Global Outlier

Outliers can be local outlier and global outlier.

Global outlier can be found using distance measure; however local outlier can be found using density based measure



Local Outlier Factors (LOF)

The concept of LOF is based on the statistics of K-Nearest Neighbours (K-NN).
Need to calculate reachability distance.
Need to find LOF score for all the instances.

Kth distance of A [dist-k(A)]= Distance between A and its k-nearest neighbour = DA (k=3)

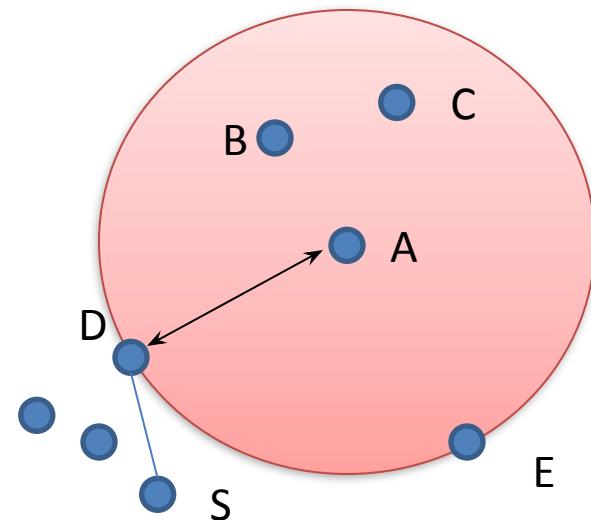
K distance neighbour of A=

$$\begin{aligned} K\text{-dn}(A) &= \{A' | A' \in D, \text{dist}(A, A') \leq \text{dist} - k(A)\} \\ &= \{B, C, D, E\} \end{aligned}$$

Reach-Dist(A,D) = $\max\{\text{Kth-distance}(D), d(A,D)\}$,
= $\max(DS, AD) = AD$

where Kth-distance(D) is the distance of instance D from its Kth nearest neighbor=DS

$d(A,D)$ is the actual distance between A and D=
AD



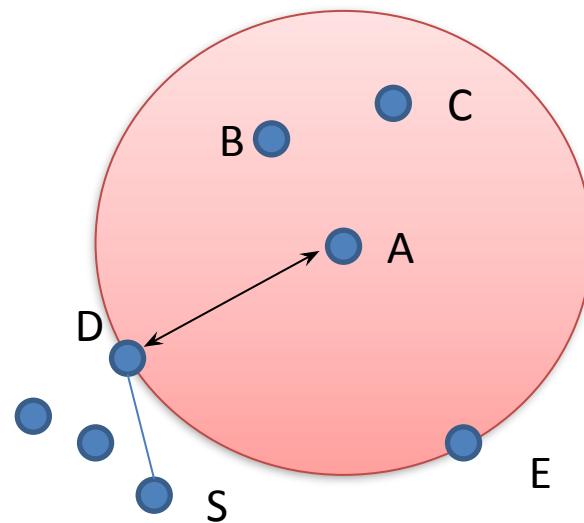
Local Outlier Factors (LOF)

Average RD_A=

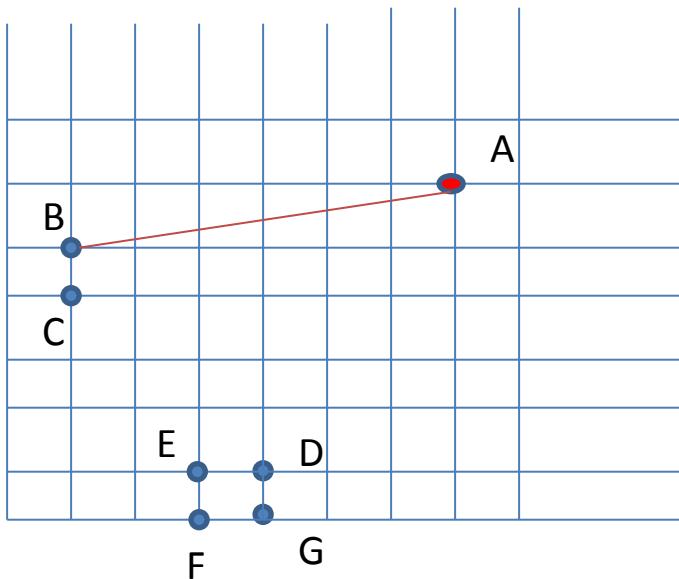
$$\frac{1}{k-dn} \sum_{k-dn} \max[(k^{th} \text{ distance of } A's \text{ neighbor}, \epsilon]$$

=

$$\frac{1}{4} \sum_4 \max[(3^{th} \text{ distance of } A's \text{ neighbor}, distan$$



Local Outlier Factors (LOF)



$$\begin{aligned}
 AB^2 &= 6^2 + 1^2 \\
 &= 36+1 \\
 &= 37 \\
 AB &= 6.08
 \end{aligned}$$

Kth distance of A
 $[dist-k(A)] = AC = 6.32$

K distance neighbour
of A = {B, C, D}

	A1	A2
A	7	6
B	1	5
C	1	4
D	4	1
E	3	1
F	3	0
G	4	0

Distances	
AB	6.08
AC	6.32
AD	5.8
AE	6.4
AF	7.2
BA	6.08
BC	1
BD	5
BE	4.4
BF	5.3
CA	6.32
CB	1
CD	4.2
CE	3.6
CF	4.4
DA	5.8
DB	5
DC	4.2
DE	1
DF	1.4
DG	1

Local Outlier Factors (LOF)

Average $RD_A =$

$$\frac{1}{3} \sum_3 \max[(3^{\text{th}} \text{ distance of } A's \text{ neighbor}, \text{distance}(A, \text{ the neighbor}))]$$

$$= \frac{1}{3} [\max(3^{\text{th}} \text{ dist } B, \text{dist}(A, B)) + \max(3^{\text{th}} \text{ dist } C, \text{dist}(A, C)) + \max(3^{\text{th}} \text{ dist } D, \text{dist}(A, D))]$$

$$= \frac{1}{3} [\max(5, 6.08) + \max(4.2, 6.32) + \max(1.4, 5.8)]$$

$$= \frac{1}{3} [6.08 + 6.32 + 5.8] = 6.06$$

Density is reverse of distance therefore **Local Reachability score LRD**

$$LRD_A = \frac{1}{RD_A}$$

$$= 1/6.06 = 0.165$$

Distances	
AB	6.08
AC	6.32
AD	5.8
AE	6.4
AF	7.2
BA	6.08
BC	1
BD	5
BE	4.4
BF	5.3
CA	6.32
CB	1
CD	4.2
CE	3.6
CF	4.4
DA	5.8
DB	5
DC	4.2
DE	1
DF	1.4
DG	1

Local Outlier Factors (LOF)

Similarly calculated,

Average $RD_B = 5.17$ and $LRD_B = 0.193$

Average $RD_C = 5.17$ and $LRD_C = 0.193$

Average $RD_D = 5.17$ and $LRD_D = 0.193$

$$LOF_A = \frac{\frac{1}{3}(LRD_B + LRD_C + LRD_D)}{LRD_A} = \frac{\frac{1}{3}(0.193 + 0.193 + 0.193)}{0.165} = \frac{0.193}{0.165} = 1.17$$

Generally, if $LOF > 1$, it is considered as an outlier, but that is not always true.

Distances	
AB	6.08
AC	6.32
AD	5.8
AE	6.4
AF	7.2
BA	6.08
BC	1
BD	5
BE	4.4
BF	5.3
CA	6.32
CB	1
CD	4.2
CE	3.6
CF	4.4
DA	5.8
DB	5
DC	4.2
DE	1
DF	1.4
DG	1

Isolation Forest (IF)

- It's an unsupervised learning algorithm that identifies anomaly by isolating outliers in the data.
- Isolation forests are an effective method for detecting outliers or novelties in data.
- It is a relatively novel method based on binary decision trees.
- Isolation forest's basic principle is that outliers are few and far from the rest of the observations.
- It can work for large datasets with one or multi dimensional feature space.

Isolation Forest (IF)

Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that selected feature. This split depends on how long it takes to separate the points.

Random partitioning produces noticeably shorter paths for anomalies. When a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies.

- An outlier score can be computed for each observation:

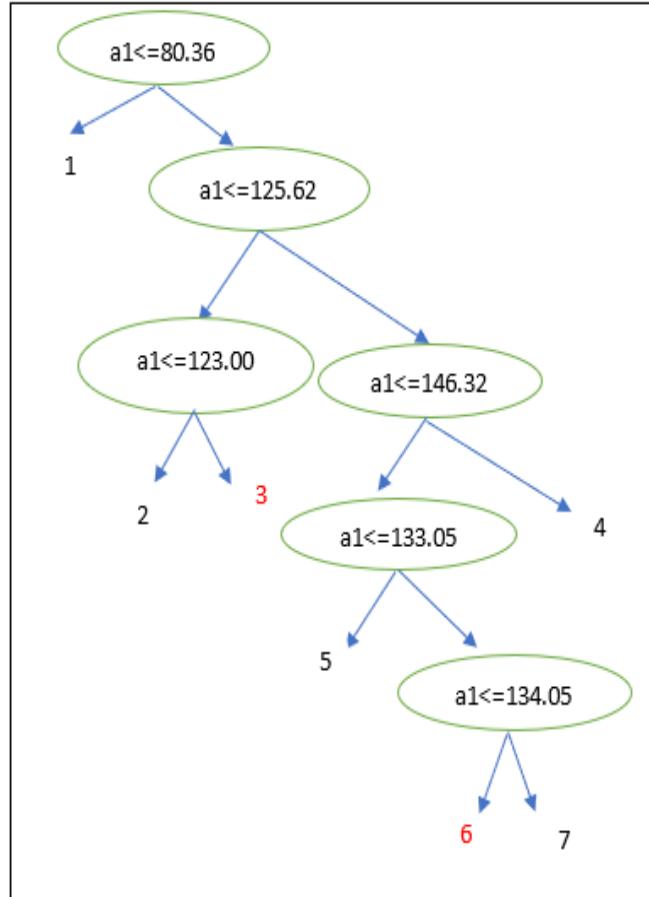
$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

outlier score

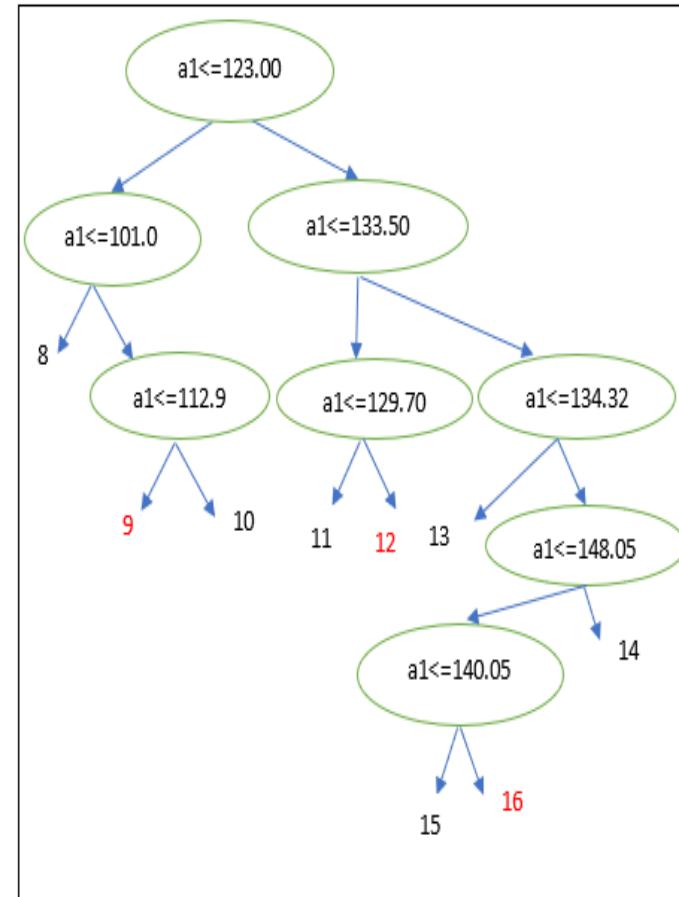
- Where $h(x)$ is the path length of the sample x , and $c(n)$ is the ‘unsuccessful length search’ of a binary tree (the maximum path length of a binary tree from root to external node) n is the number of external nodes. After giving each observation a score ranging from 0 to 1; 1 meaning more outlyingness and 0 meaning more normality. A threshold can be specified (ie. 0.55 or 0.60)

Isolation Forest (IF)

Sl no.	a1	Leaf node
1	15 0	4, 14
2	12 2	2, 10
3	13 5	7, 15
4	13 2	5, 13
5	4	1, 8
6	13 8	7, 15
7	12	2, 10
8	set 12 7	5, 11



Decision
tree 1



Decision
tree 2

ISOLATION FOREST

Node no. 3,6,9,12,16 are unsuccessful path

- The average unsuccessful path length is given by node 6 and 16

which is 5. Since 2 trees are considered therefore $h(x)$ is the

average path of the datapoint for the 2 trees.

$$s_1 = 2^{-\left(\frac{3+4}{2}\right)} = 2^{-(0.7)} = 0.615$$

$$s_2 = 2^{-\left(\frac{3+3}{2}\right)} = 2^{-(0.6)} = 0.65$$

$$s_3 = 2^{-\left(\frac{5+5}{2}\right)} = 2^{-(1)} = 0.5$$

$$s_4 = 2^{-\left(\frac{4+3}{2}\right)} = 2^{-(0.7)} = 0.615$$

$$s_5 = 2^{-\left(\frac{1+2}{2}\right)} = 2^{-(0.3)} = 0.812$$

$$s_6 = 2^{-\left(\frac{5+5}{2}\right)} = 2^{-(1)} = 0.5$$

$$s_7 = 2^{-\left(\frac{3+3}{2}\right)} = 2^{-(0.6)} = 0.65$$

$$s_8 = 2^{-\left(\frac{4+3}{2}\right)} = 2^{-(0.7)} = 0.615$$

Score= $s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$

Sl n o.	weig ht	sco res
1	150	0.615
2	122	0.65
3	135	0.50
4	132	0.615
5	4	0.812
6	138	0.50
7	121	0.65
8	127	0.615

one anomaly is detected

a1	scor es
4	0.812

ISOLATION FOREST

Isolation Forest pros:

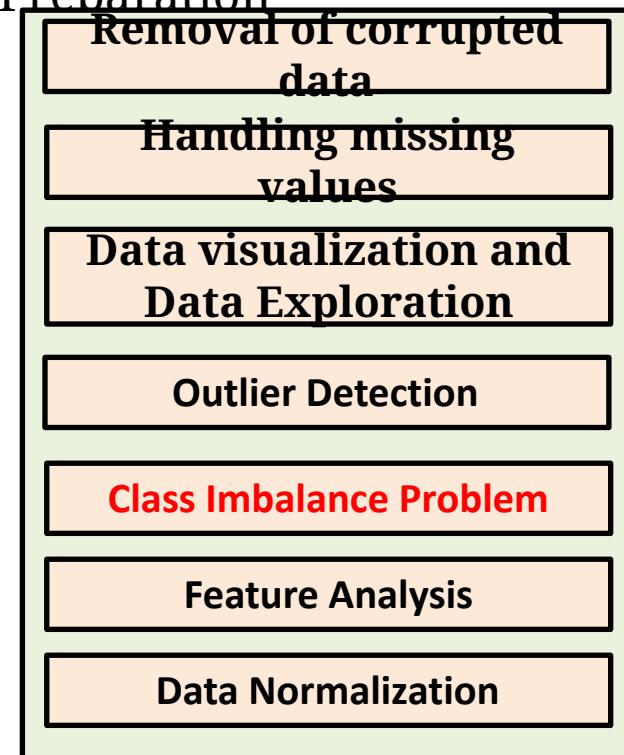
- There is no need of scaling the values in the feature space.
- It is an effective method when value distributions can not be assumed.
- It has few parameters, this makes this method fairly robust and easy to optimize.

Isolation Forest cons:

- Visualizing results is complicated.
- Sometimes, training time can be very long and computationally expensive.

Machine Learning Model

Pre-processing/ Data Preparation

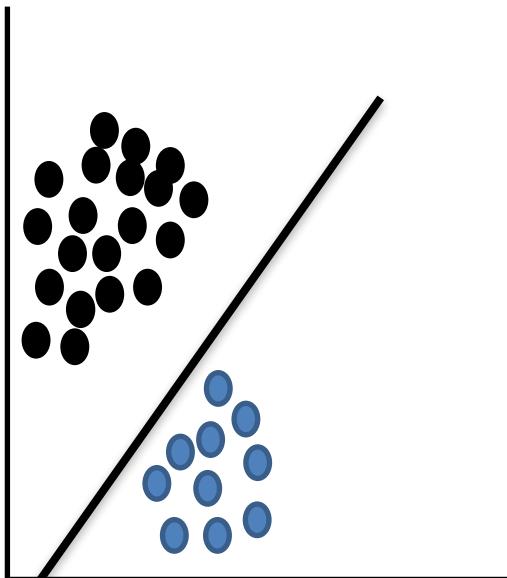


Class imbalance Problem

Class Imbalance: Biases towards majority class(es)

Approaches: Sampling, Algorithmic,
Ensembling

Sampling: Undersampling & oversampling



Undersampling: reduces majority class instances: examples RUS, Tomek link based undersampling, Condensed Nearest Neighbor (CNN) undersampling etc.

Oversampling: increases minority class instances: examples ROS, SMOTE, Border-Line SMOTE, Safe level SMOTE, ADASYN etc.

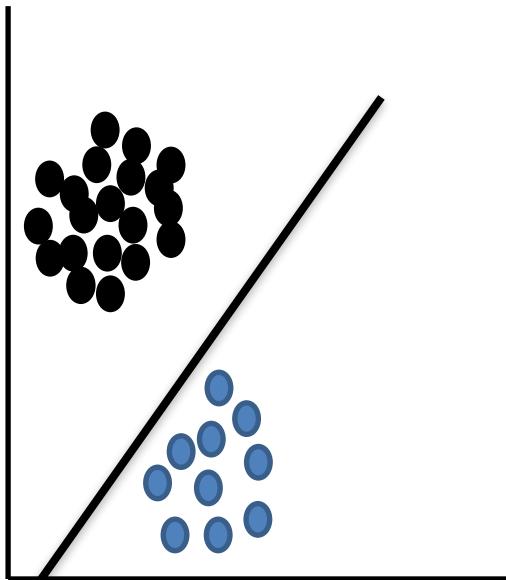
[\[Gaussian SMOTE\]](#) (2017)

[\[kmeans SMOTE\]](#) (2018)

Algorithmic: changes the cost function, higher misclassification cost for minority class cost-sensitive neural network etc.

Ensembling: SMOTEBoost, RUSBoost etc

Class imbalance Problem



RUS: Random Under Sampling technique is random undersampling of the majority class.

This can potentially lead to loss of information about the majority class.

However, in cases where each example of the majority class is near other examples of the same class, this method might yield good results.

$$R = S/L = 0.5$$

$$S = 500$$

$$L = 1500$$

$$L = 1000 \text{ (after undersampling)}$$

Distance Measure

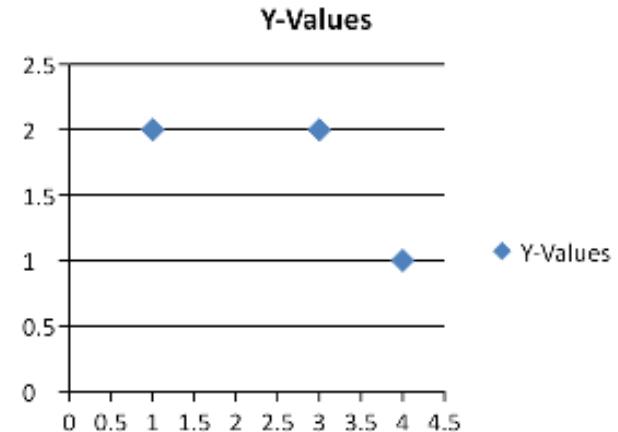
Let $p = (p_1, p_2)$ and $q = (q_1, q_2)$ be two points:

- ✓ City block distance $d(p, q) = |p_1 - q_1| + |p_2 - q_2|$
- ✓ Euclidean distance $d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$
- ✓ Minkowski distance $d(p, q) = (\sum_{i=1}^M |p_i^n - q_i^n|^r)^{\frac{1}{r}}$

For r=1, Minkowski distance = City block distance

For r=2, Minkowski distance = Euclidean distance

	M=1	M=2
1 st	1	2
2 nd	3	2
3 rd	4	1



$$1^{\text{st}} = (1, 2)$$

$$2^{\text{nd}} = (3, 2)$$

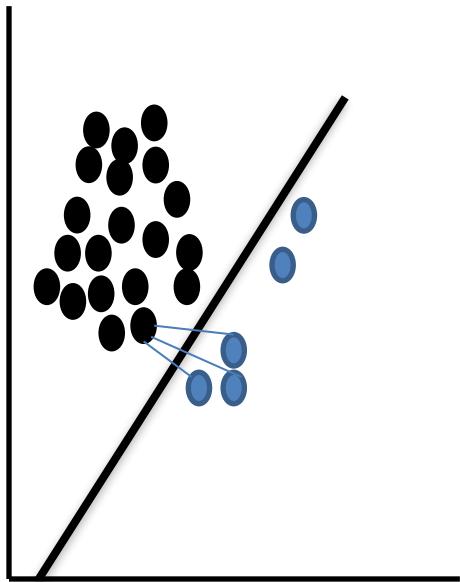
$$3^{\text{rd}} = (4, 1)$$

$$\text{Dis}(1^{\text{st}}, 2^{\text{nd}}) = |1-3| + |2-2| = 2$$

Undersampling

NearMiss-1: Those points from L are retained whose mean distance to the k nearest points in S is lowest.

where k is a tunable hyperparameter

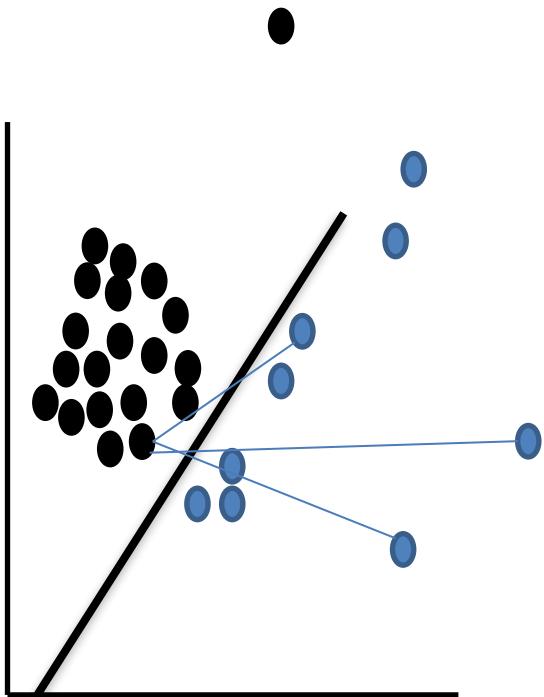


R= S/L=0.5 (As we wish)

$$s_1 = \frac{l_1 + l_2 + l_3}{3} = x_1$$

8, 6, 4, 5, 1, 2, 3, 7, 9, 10, 12, 16, 17, 18, 20, 19, 15, 14, 13, 11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Undersampling



NearMiss-2: keeps those points from L whose mean distance to the k farthest points in S is lowest.

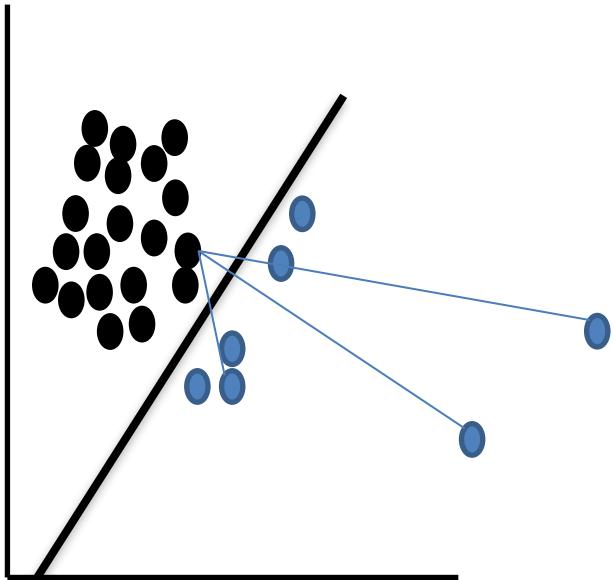
where k is a tunable hyperparameter

$R = S/L = 0.5$ (As we wish)

$$s_1 = \frac{f_1 + f_2 + f_3}{3} = x_1$$

8, 6, 4, 5, 1, 2, 3, 7, 9, 10, 12, 16, 17, 18, 20, 19, 15, 14, 13, 11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Undersampling



NearMiss-2: keeps those points from L whose mean distance to the k farthest points in S is lowest.

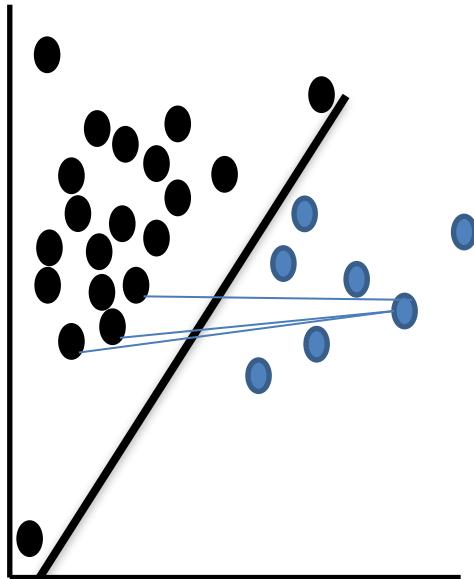
where k is a tunable hyperparameter

R= S/L=0.5 (As we wish)

$$s_1 = \frac{f_1 + f_2 + f_3}{3} = x_1$$

8, 6, 4, 5, 1, 2, 3, 7, 9, 10, 12, 16, 17, 18, 20, 19, 15, 14, 13, 11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

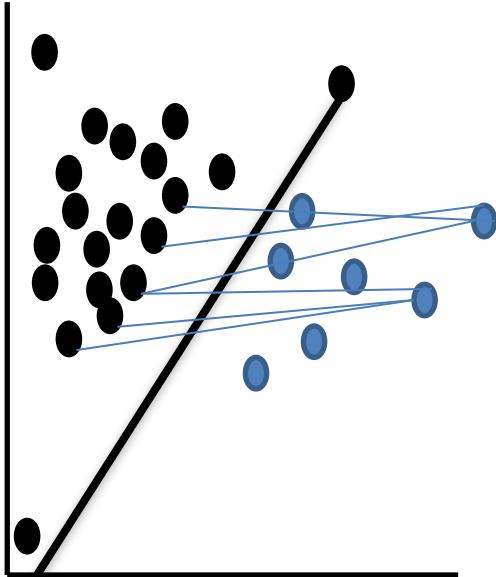
Undersampling



NearMiss-3: selects k nearest neighbors in L for every point in S . In this case, the undersampling ratio is directly controlled by k and is not separately tuned.

where k is a tunable hyperparameter

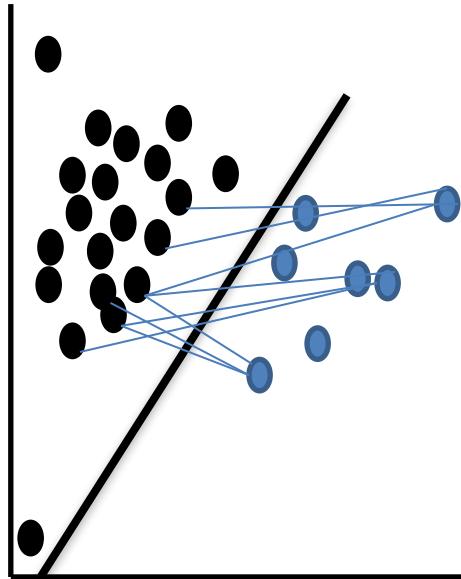
Undersampling



NearMiss-3: selects k nearest neighbors in L for every point in S . In this case, the undersampling ratio is directly controlled by k and is not separately tuned.

where k is a tunable hyperparameter

Undersampling

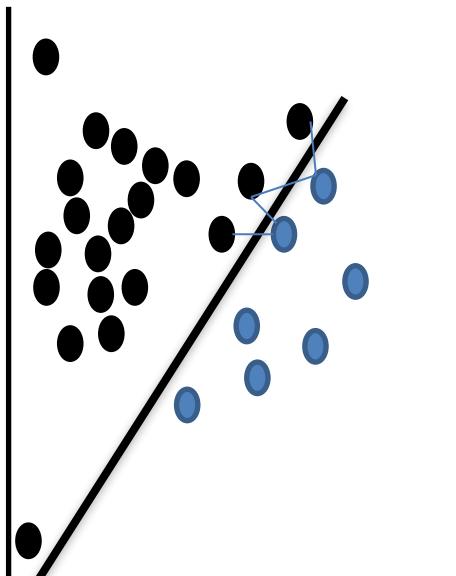


NearMiss-3: selects k nearest neighbors in L for every point in S . In this case, the undersampling ratio is directly controlled by k and is not separately tuned.

where k is a tunable hyperparameter

Undersampling

Condensed Nearest Neighbor (CNN): the goal is to choose a subset U of the training set T such that for every point in T its nearest neighbor in U is of the different class. U can be grown iteratively as follows:



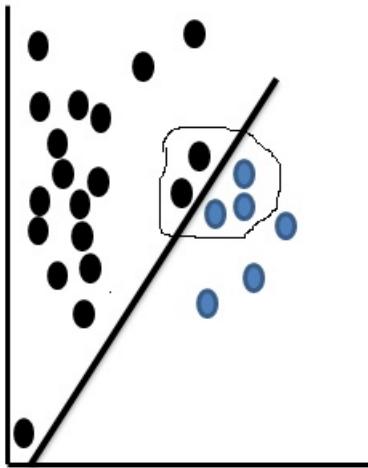
1. Select a random point from T and set $U = \{p\}$.
2. Scan $T - U$ and add to U the first point found whose nearest neighbor in U is of a different class
3. Repeat step 2 until U is maxima

Undersampling via CNN can be slower compared to other methods since it requires many passes over the training data.

Further, because of the randomness involved in the selection of points at each iteration, the subset selected can vary significantly.

A variant of CNN is to only undersample L i.e. retain all points from S but retain only those points in L that belong to U .

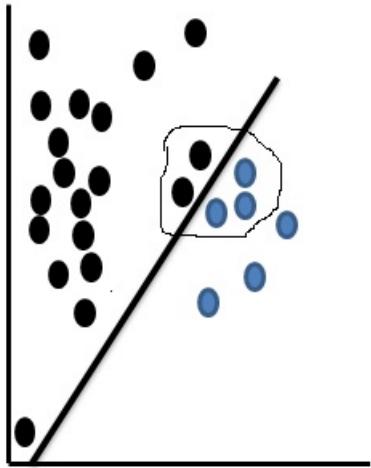
Undersampling



Edited Nearest Neighbor (ENN):
undersampling of the majority class is done by removing points whose class label differs from a majority of its k nearest neighbors.

Say $k=4$

Undersampling

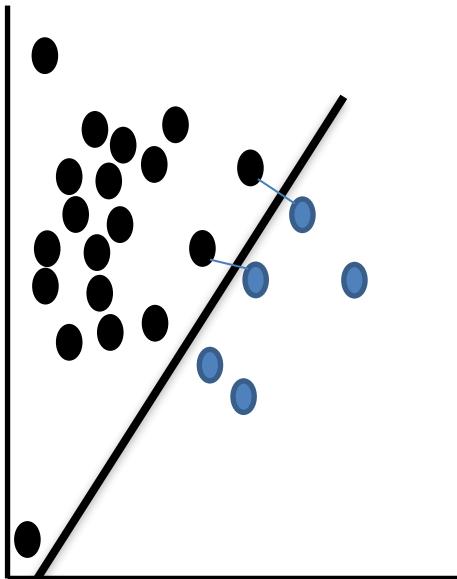


Repeated Edited Nearest Neighbor:

The ENN algorithm is applied successively until ENN can remove no further points.

Say $k=4$

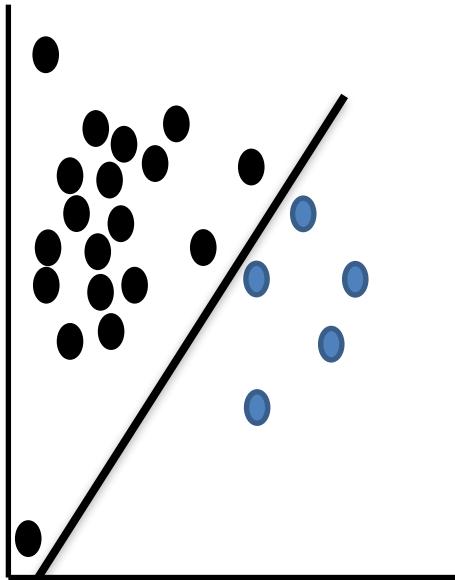
Undersampling



Tomek Link Removal:

- A pair of examples is called a Tomek link if they belong to different classes and are each other's nearest neighbors.
- Undersampling can be done by removing all tomek links from the dataset.
- An alternate method is to only remove the majority class samples that are part of a Tomek link.

Oversampling



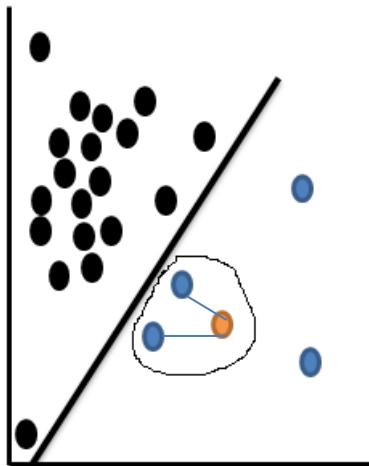
Random oversampling of minority class:

- Points from the minority class may be oversampled randomly.
- This method is prone to overfitting.
- We consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

SMOTE:

A more sophisticated means for oversampling is Synthetic Minority Oversampling Technique (SMOTE)



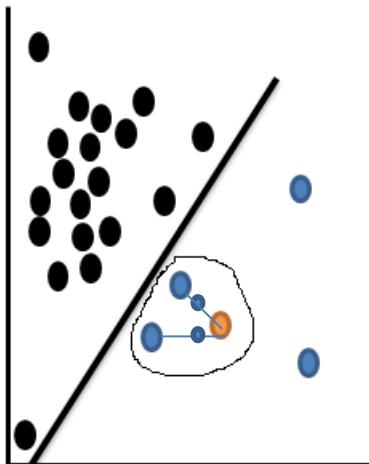
For each point p in S :

1. Compute its k nearest neighbors in S .
 2. Randomly choose $r \leq k$ of the neighbors (with replacement).
 3. Choose a random point along the lines joining p and each of the r selected neighbors.
 4. Add these synthetic points to the dataset with class S .
-
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

SMOTE:

A more sophisticated means for oversampling is Synthetic Minority Oversampling Technique (SMOTE)



For each point p in S :

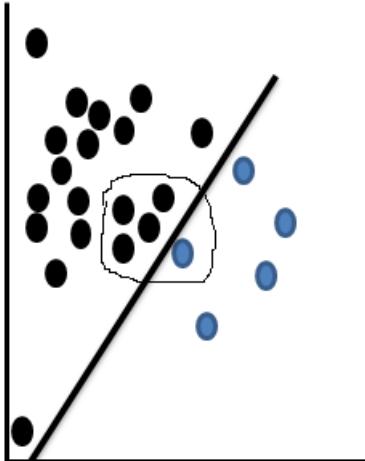
1. Compute its k nearest neighbors in S .
 2. Randomly choose $r \leq k$ of the neighbors (with replacement).
 3. Choose a random point along the lines joining p and each of the r selected neighbors.
 4. Add these synthetic points to the dataset with class S .
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

BORDERLINE SMOTE-1:

There are two enhancements of SMOTE, termed borderline SMOTE, which may yield better performance than SMOTE.

For each point p in S :



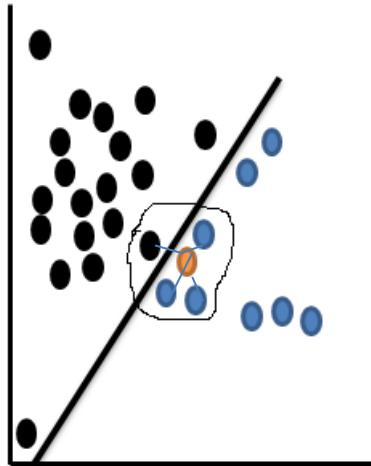
1. Compute its m nearest neighbors in T . Call this set M_p and let $m_0 = |M_p \cap L|$.
 2. If $m_0 = m$, p is a noisy example. Ignore p and continue to the next point.
 3. If $0 \leq m_0 \leq m/2$, p is safe. Ignore p and continue to the next point.
 4. If $m/2 < m_0 < m$, add p to the set DANGER.
 6. For each point d in DANGER, apply the SMOTE algorithm to generate synthetic examples.
-
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

Borderline-SMOTE1:

There are two enhancements of SMOTE, termed borderline SMOTE, which may yield better performance than SMOTE.

For each point p in S :



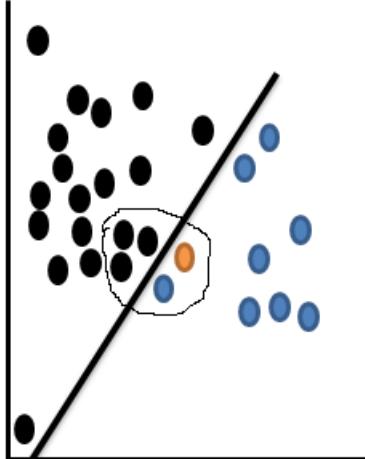
1. Compute its m nearest neighbors in T . Call this set M_p and let $m_0 = |M_p \cap L|$.
 2. If $m_0 = m$, p is a noisy example. Ignore p and continue to the next point.
 3. If $0 \leq m_0 \leq m/2$, p is safe. Ignore p and continue to the next point.
 4. If $m/2 < m_0 < m$, add p to the set DANGER.
 6. For each point d in DANGER, apply the SMOTE algorithm to generate synthetic examples.
- We may consider the result of oversampling of S to achieve $\pi > 0.5$.

Oversampling

Borderline-SMOTE1:

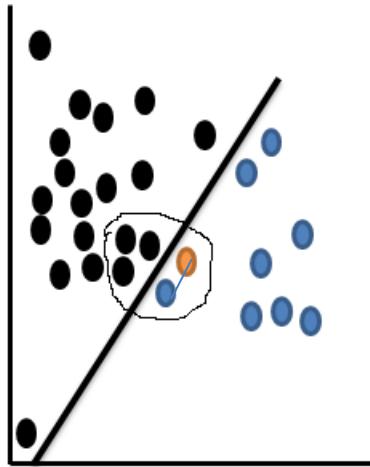
There are two enhancements of SMOTE, termed borderline SMOTE, which may yield better performance than SMOTE.

For each point p in S :



1. Compute its m nearest neighbors in T . Call this set M_p and let $m_0 = |M_p \cap L|$.
 2. If $m_0 = m$, p is a noisy example. Ignore p and continue to the next point.
 3. If $0 \leq m_0 \leq m/2$, p is safe. Ignore p and continue to the next point.
 4. If $m/2 < m_0 < m$, add p to the set DANGER.
 6. For each point d in DANGER, apply the SMOTE algorithm to generate synthetic examples.
- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling

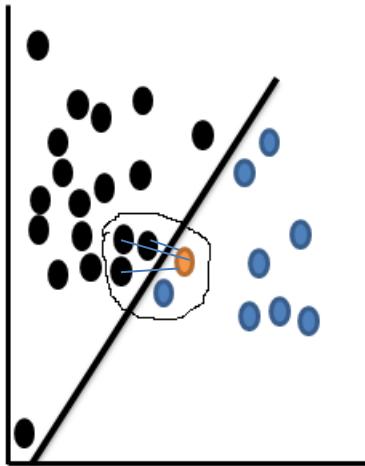


Borderline-SMOTE2:

Borderline-SMOTE2 is similar to Borderline-SMOTE1 except in the last step, new synthetic examples are created along the line joining points in DANGER to either their nearest neighbors in S or their nearest neighbors in L.

- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling



Borderline-SMOTE2:

Borderline-SMOTE2 is similar to Borderline-SMOTE1 except in the last step, new synthetic examples are created along the line joining points in DANGER to either their nearest neighbors in S or their nearest neighbors in L.

- We may consider the result of oversampling of S to achieve $r = 0.5$.

Oversampling + Undersampling

Combination methods

Performing a combination of oversampling and undersampling can often yield better results than either in isolation.

SMOTE + Tomek Link Removal

	$ L $	$ S $
Before resampling	6320	680
After resampling	6050	3160

SMOTE + ENN

	$ L $	$ S $
Before resampling	6320	680
After resampling	4894	3160

Treatment of Class imbalance Problem

Algorithmic level solution

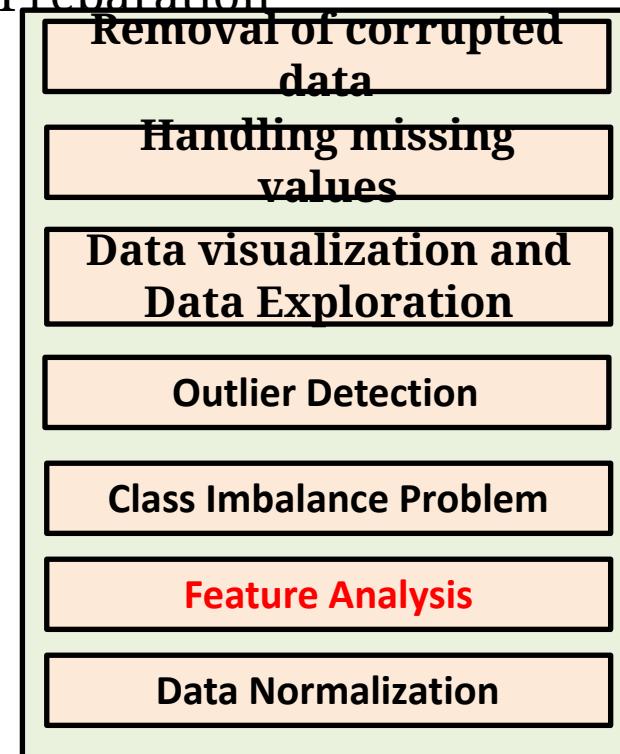
- Cost Sensitive Neural Network

Ensemble methods

- EasyEnsemble
- BalanceCascade
- SMOTEboost
- RUSboost

Machine Learning Model

Pre-processing/ Data Preparation



What is a Feature?

Feature is a quantifiable/measurable characteristic of a phenomenon being observed.

BP	Heart Beat	Weight	Diabetes
120	70	50	Y
125	65	60	Y
130	59	52	N

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor

Types of feature?

Discrete Feature: can only take certain values. Can be numeric or categorical: the results of rolling 2 dice.

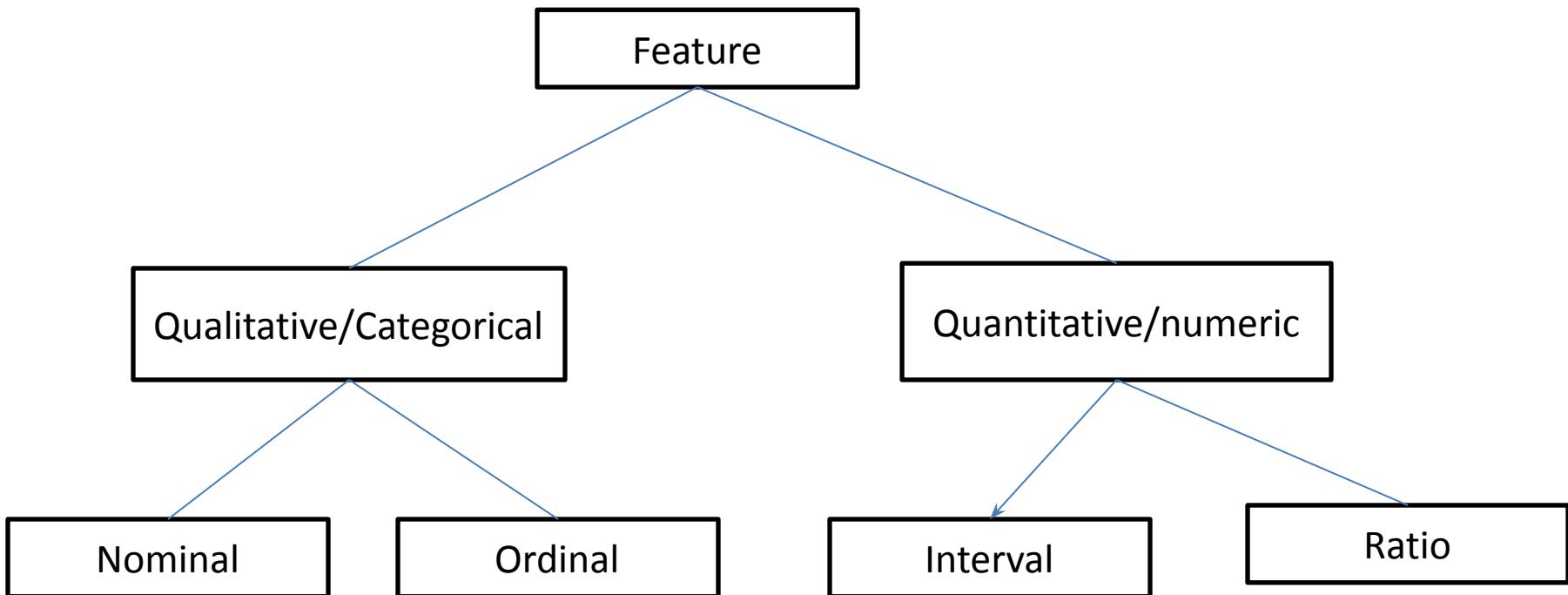
Sepal Length	Sepal Width	Petal Length	Petal Width	Species
big	medium	medium	small	Virginica
big	medium	small	small	Setosa
big	small	medium	small	Versicolor

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
3	2	2	1	Virginica
3	2	1	1	Setosa
3	1	2	1	Versicolor

Continuous Feature: can take any value (within a range)

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor

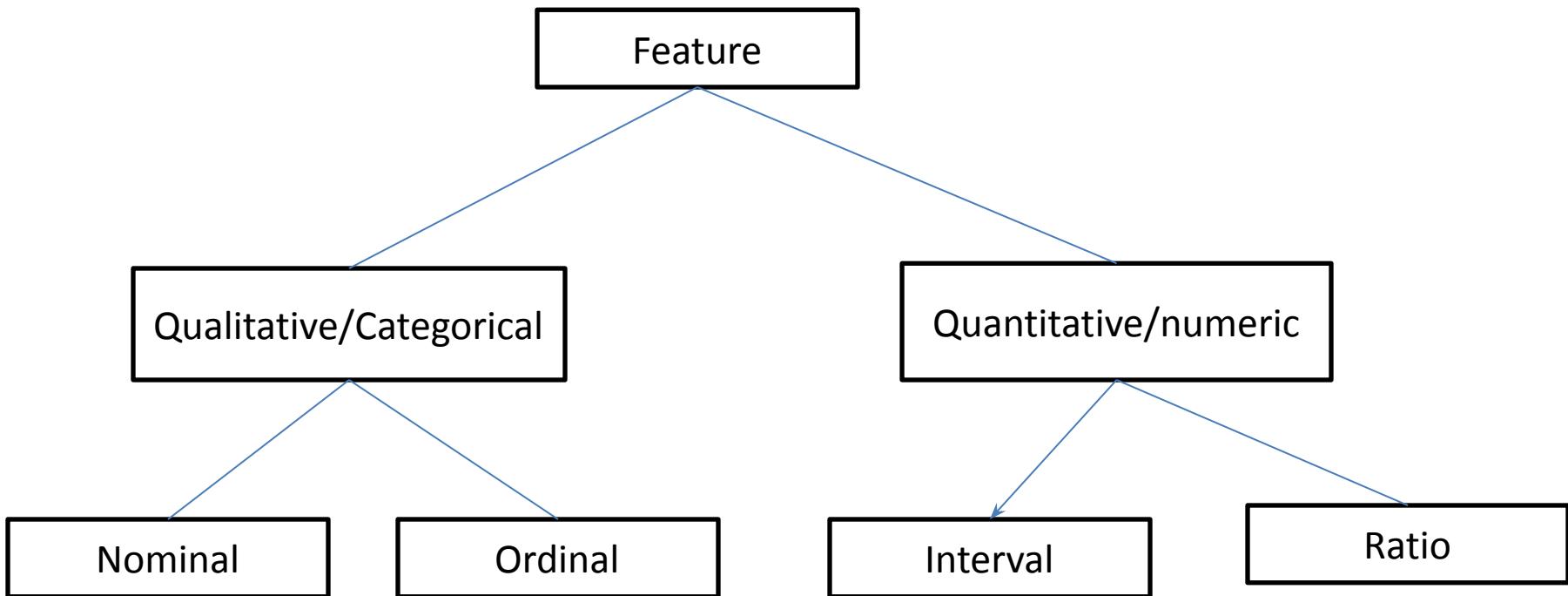
Types of feature?



Nominal Data: the range of values is not ordered in any sense, but simply **named** (hence the nom). Blood groups, gender, etc.

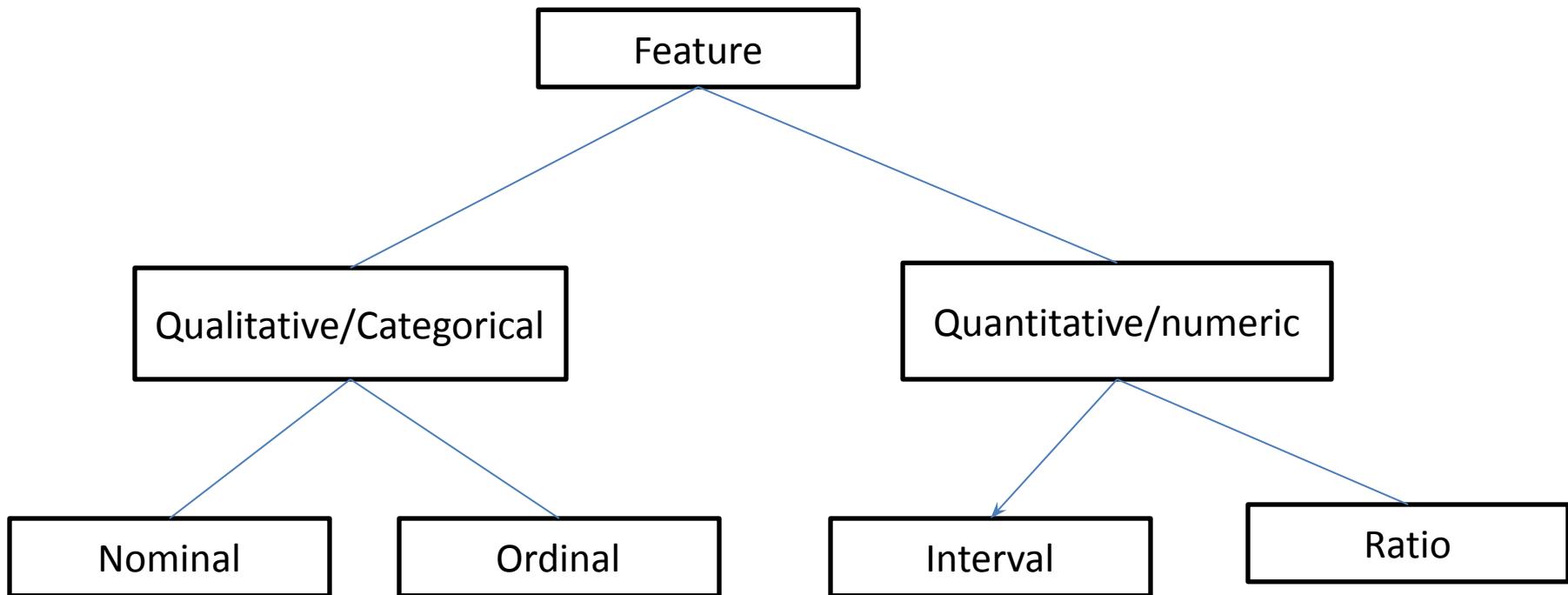
Ordinal Data: the range of values is ordered: **socio economic status** (“low income”, “middle income”, “high income”), education level (“high school”, “BS”, “MS”, “PhD”), income level (“less than 50K”, “50K-100K”, “over 100K”), satisfaction rating (“extremely dislike”, “dislike”, “neutral”, “like”, “extremely like”)

Types of feature?



Interval Data: Interval data is a type of data which is measured along a scale, in which each point is placed at an equal distance (interval) from one another. **The difference between 100 degrees Fahrenheit and 90 degrees Fahrenheit is the same as 60 degrees Fahrenheit and 70 degrees Fahrenheit.** Time of each day, Age, Dates, Voltage.

Types of feature?

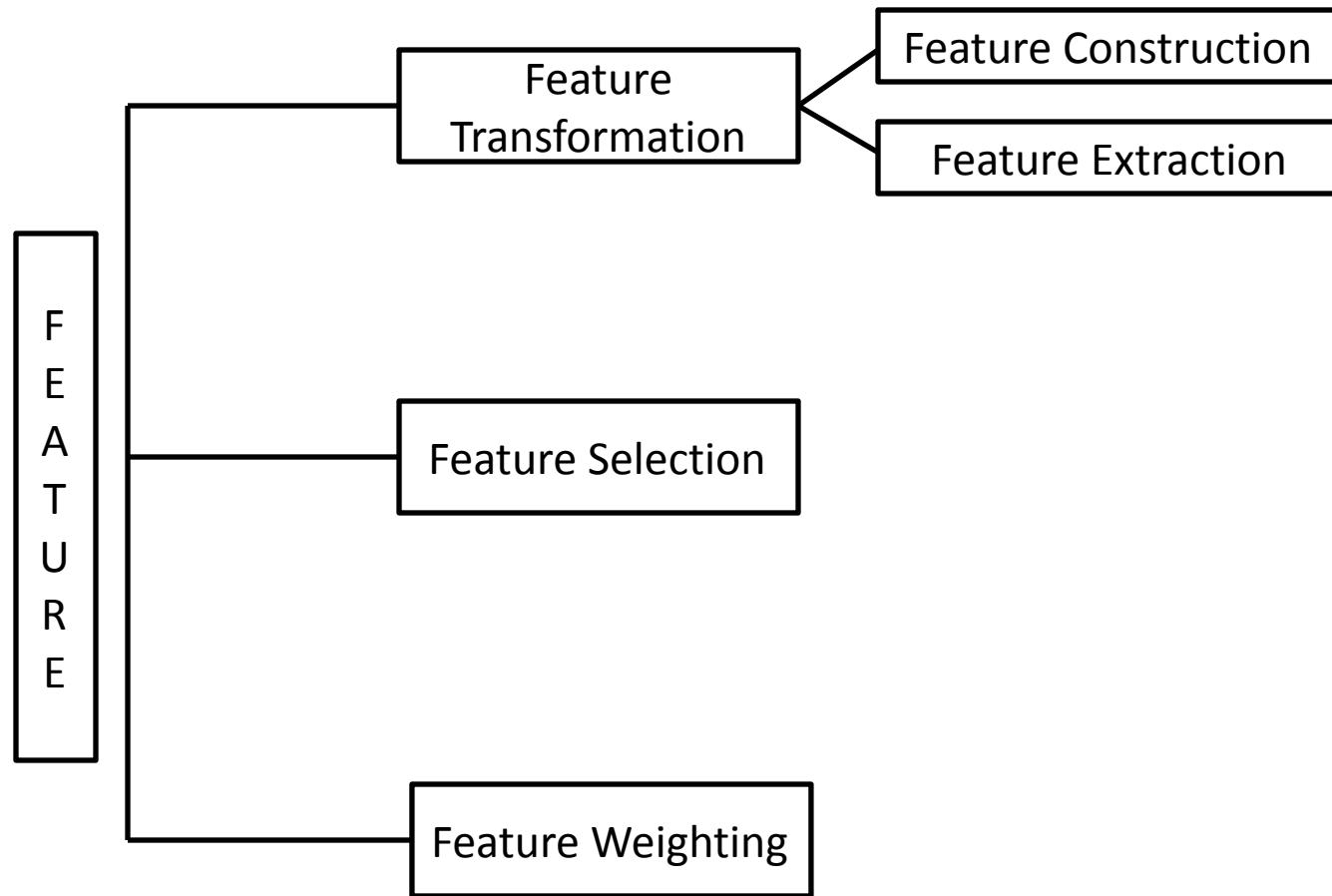


Ratio Data: Unlike interval data, ratio data has a true zero. This basically means that zero is an absolute, below which there are no meaningful values. Speed, age, or weight are all excellent examples since none can have a negative value (you cannot be -10 years old or weigh -160 pounds!)

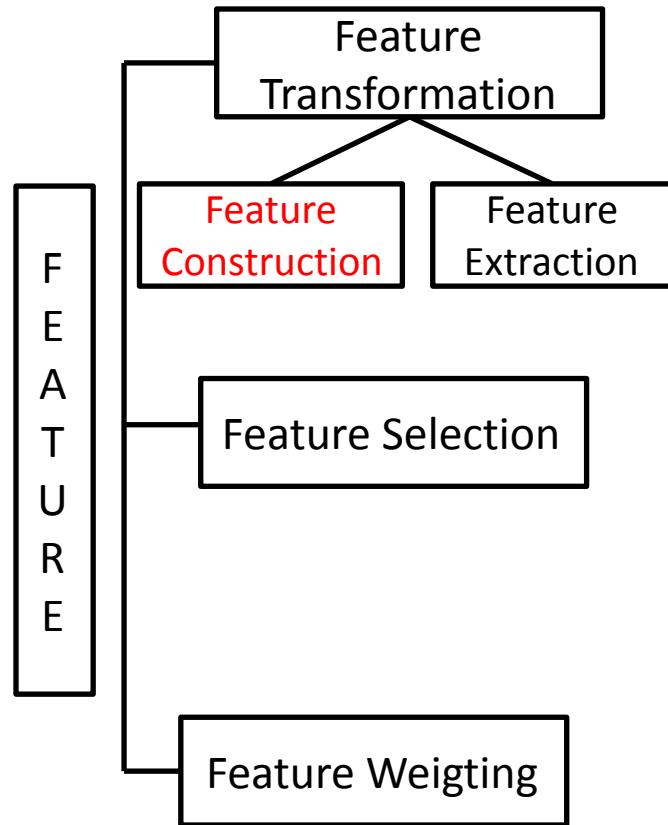
THE FOUR LEVELS OF MEASUREMENT:

	Nominal	Ordinal	Interval	Ratio
Categorizes and labels variables	✓	✓	✓	✓
Ranks categories in order		✓	✓	✓
Has known, equal intervals			✓	✓
Has a true or meaningful zero				✓

Feature Analysis



Feature Construction



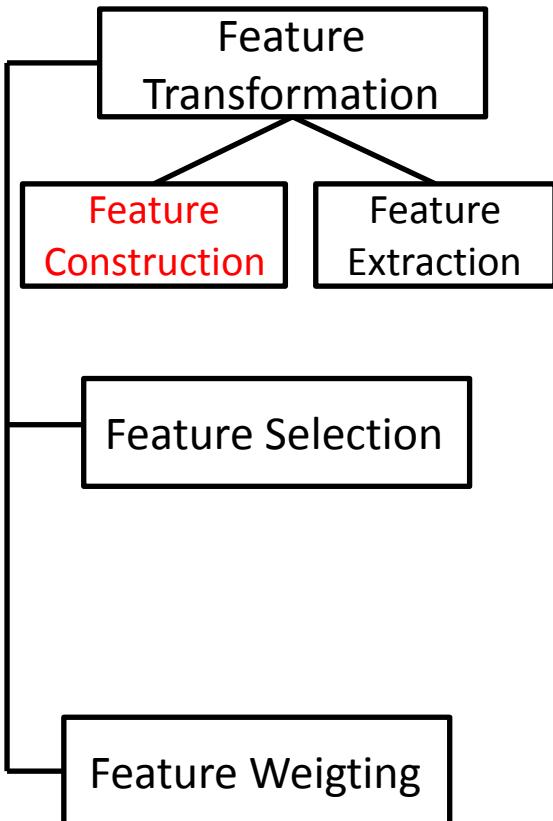
Feature Construction: Generates a new set of more powerful feature from a given set of input feature. Say, there are n features, after feature construction m more feature are added. Now total features = $(n+m)$

A Length	A Breadth	A price in RS
80	59	2360000
54	45	1215000
78	56	2184000

A Length	A Breadth	A Area	A price in RS
80	59	4720	2360000
54	45	2430	1215000
78	56	4368	2184000

Feature Construction

F
E
A
T
U
R
E



Need of Feature Construction:

- When features have categorical value and machine learning needs numeric value inputs.
- When features having continuous value and need to be converted to discrete/ordinal values.
- When text-specific feature construction needs to be done.
- Sometimes improves system performance

Feature Construction

Encoding categorical (nominal) variables:

Age	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N

Age	City A	City B	City C	Parents athlete_Y	Win_chance_1
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0

Feature Construction

Encoding categorical (ordinal) variables:

Science	maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D

Science	maths	Grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4

Feature Construction

Numeric features (continuous) to categorical features:

A_Area	A_Price
4720	2300000
2430	1200000
4368	2100000
3969	1900000
6142	3000000

A_Area	A_Grade
4720	Medium
2430	Low
4368	Medium
3969	Low
6142	High

A_Area	A_Grade
4720	2
2430	1
4368	2
3969	1
6142	3

Feature Construction

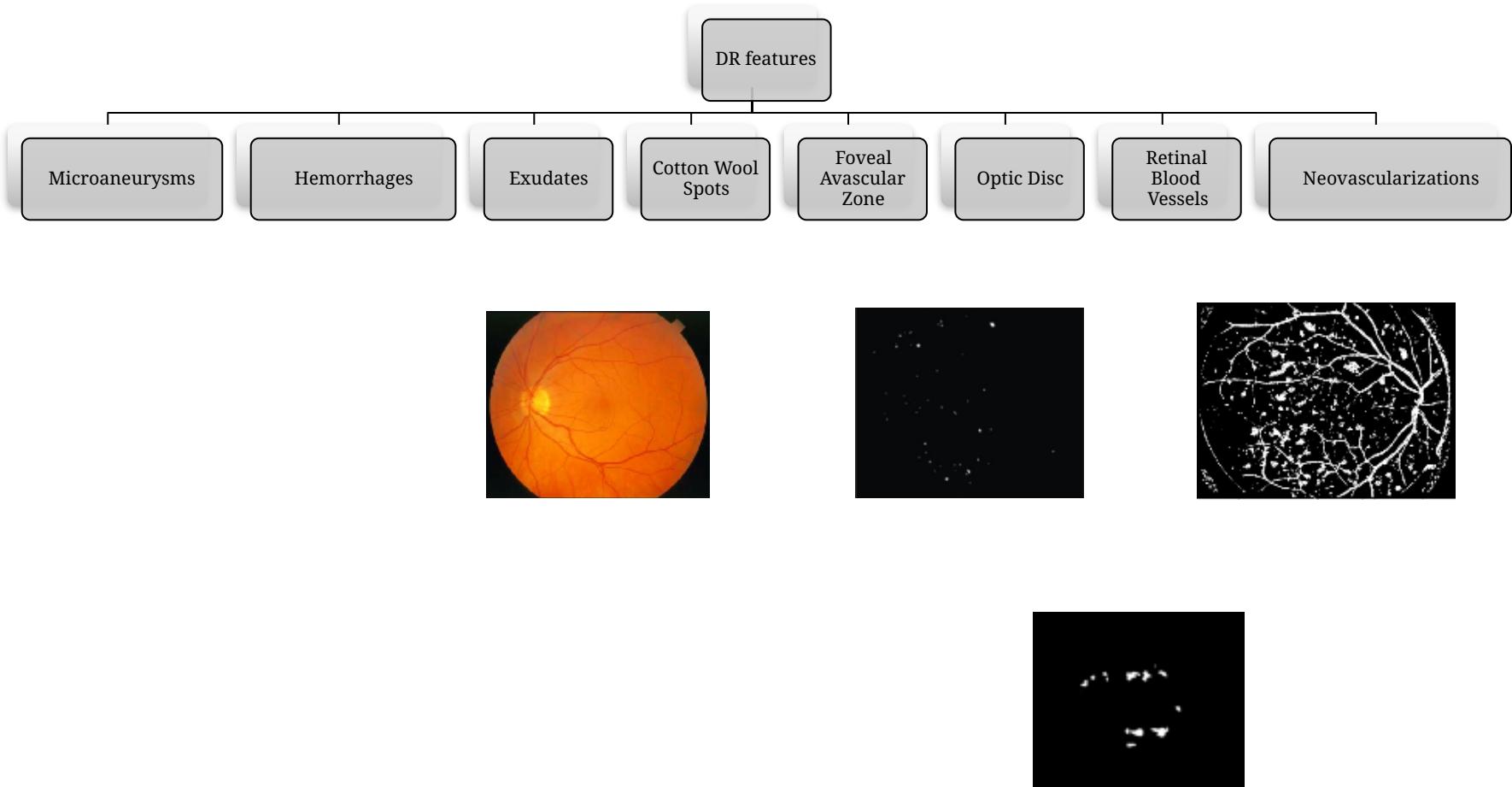
Text Specific feature construction and representation: Vector Space Model (VSM) and Graph Based Model (GBM)

Thrilling	movie	good	attractive	looser	lonely	expectation	bore	class
1	1	1	1	0	0	1	0	1
1	0	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	1
0	0	0	0	1	1	1	1	0
1	1	0	0	1	1	0	1	0
0	1	0	0	1	1	0	1	0

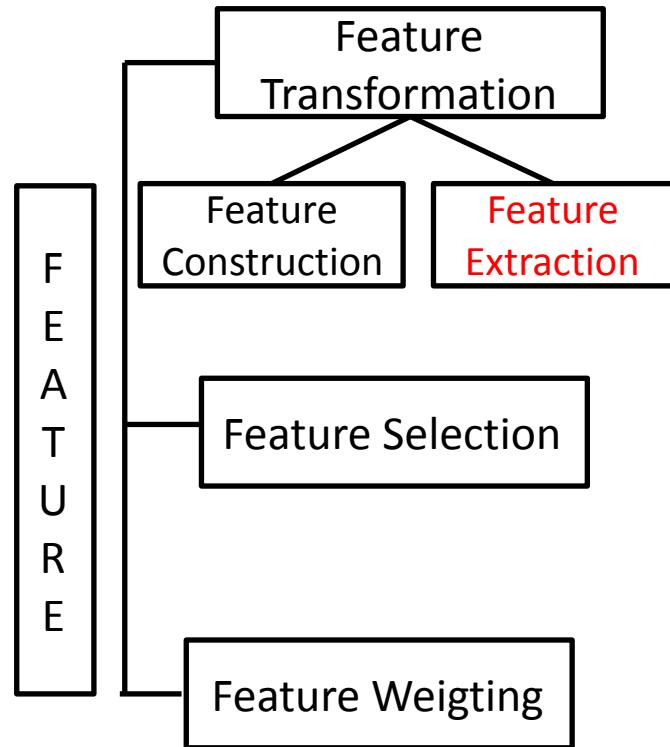
1. The movie is good, thrilling, attractive. It is up to my expectation ----- 1 (good)
3. The movie is good and attractive and is up to expectation ----- 1 (good)
5. The movie was looser, lonely. It was bore. It was thrilling. ----- 0 (bad)

Feature Construction

Image Feature Construction for diabetic retinopathy



Feature Extraction



Feature Extraction: New features are created from a combination of original features: some operators:
Boolean Features: Conjunction, Disjunction, Negation etc.

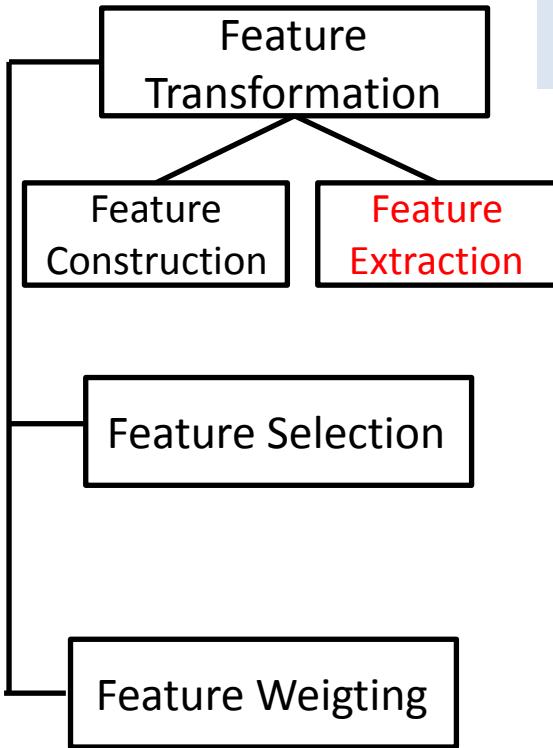
Nominal Features: Cartesian product, M of N etc.

Numerical Features: Min, Max, Addition, Subtraction, Multiplication, Division, Average, Equivalence, Inequality etc.

$$F' = f(F); \quad F_1' = k_1 F_1 + k_2 F_2; \quad m < n$$

Feature Extraction

F
E
A
T
U
R
E



Feat_A	Feat_B	Feat_C	Feat_D
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2

Feat_1	Feat_2
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
37.02	238.42

$$\text{Feat_1} = 0.3 * \text{Feat_A} + 0.9 * \text{Feat_B}$$

$$\text{Feat_2} = \text{Feat_B} + 0.5 * \text{Feat_C} + 0.6 * \text{Feat_D}$$

Feature Extraction

Some of the popular feature extraction algorithms used in ML are given below:

1. Singular Value Decomposition (SVD)
2. Linear Discriminant Analysis (LDA)
3. Principle Component Analysis (PCA)
4. Fisher's Linear Discriminant (FLD)

Principal Component Analysis

PCA is a useful statistical ML technique that has found application in fields such as face recognition and image compression, and **is a common technique for finding patterns in data of high dimension.**

The other main advantage of PCA is that once we have found these patterns in the data, then we can compress the data, ie. by reducing the number of dimensions, without much loss of information.

Method:

Step 1: Get some data

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Principal Component Analysis

Step 2: Subtract the mean

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9
Mean	
X'=1.81	Y'=1.91

$$2.5 - 1.81 = .69$$

$$2.4 - 1.91 = .49$$

DataAdjust =

(X-X')	(Y-Y')
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.0

Principal Component Analysis

Step 3: Calculate the covariance matrix of x and y (two dimensional data)

$$\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix}$$

If you calculate the covariance between one dimension and itself, you get the variance.

The formula for variance could also be written like this:

$$var(X) = \frac{\sum_1^n (X_i - X') (X_i - X')}{(n - 1)}$$

$$cov(X, Y) = \frac{\sum_1^n (X_i - X') (Y_i - Y')}{(n - 1)}$$

Principal Component Analysis

$$\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix}$$

$$var(X) = \frac{\sum_1^n (X_i - X') (X_i - X')}{(n - 1)}$$

$$cov(X, Y) = \frac{\sum_1^n (X_i - X') (Y_i - Y')}{(n - 1)}$$

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

$cov(X, Y) = (.69 * .49) + (-1.31 * -1.21) + (.39 * .99) + (.09 * .29) + (1.29 * 1.09) + (.49 * .79) + (.19 * -.31) + (-.81 * -.81) + (-.31 * -.31) + (-.71 * -1.0) = 5.539$

(X-X')	(Y-Y')
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.0
5.539/9=0.615444444	
Var(X)=	
5.549/9=0.616555556	
Var(Y)=	
6.4289/9=0.714322222	

Principal Component Analysis

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

$$eigenvalues = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

Step 5: Choosing components and forming a feature vector

It turns out that the eigenvector with the highest eigenvalue is the principle component of the data set.

The eigenvector with the largest eigenvalue is the one that points the middle of the data.

It is the most significant relationship between the data dimensions.

Principal Component Analysis

Step 5: Choosing components and forming a feature vector

$$eigenvalues = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

FeatureVector= $eig_1eig_2eig_3eig_4$

$$= \begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

Principal Component Analysis

Step 6: Deriving the new data set

we simply take the transpose of the vector and multiply it on transpose of *DataAdjust*

$$= \begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

$$\text{FinalData} = \text{RowFeatureVector} \times \text{RowDataAdjust}$$

$$\begin{pmatrix} -.68 & -.74 \end{pmatrix} \times \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & . . \\ .49 & -1.21 & .99 & .29 & 1.09 & .79 \end{pmatrix}$$

$$= (-.68 * .69) + (-.74 * .49)$$

$$=(-0.8318, 1.7862, -0.9978, -0.2758, -1.6838, -0.9178, 0.1002, 1.1502, 0.4402, 1.2228)$$

Feature Selection

Feature Subset Selection: is the most critical pre-processing in ML and selects a subset which keeps meaningful contribution in a ML task.

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3

Feature set= {F1, F2, F3, F4,FN}

Feature Subset= {F1, F2,,FM}

Where M<=N

- High-dimensional data: DNA analysis, GIS, Social Networking etc.
- DNA microarray data can have up to 450000 variables(gene)
- Text data is extremely high dimensional data

Feature Subset Selection: is the most critical pre-processing in ML and selects a subset which keeps meaningful contribution in a ML task.

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3

Advantages of Feature Selection:

1. Improve accuracy
2. Improve efficiency by reducing time complexity
3. It helps in the simplification of the model so that it can be easily interpreted by the researchers.

Jobs of Feature Subset Selection— 1. Feature relevance 2. Feature redundancy

2. Feature Relevance: i. Irrelevant feature ii. Relevant feature

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

2. Feature Redundancy:

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
14	1.25	25.3

Measures of feature relevance:

- **Information Gain**
- **Mutual information**
- **Fisher score**
- **Analysis of Variance (ANOVA)**
- **Chi-Square**
- **Dispersion ratio**
- **Relief**

Feature Selection

Measures of feature relevance using **Information Gain**:

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	Students
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Students
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Students
T6	Quiet	Baked	Sandals	Students
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Students

Information Gain

- Let C_1, C_2, \dots, C_m be the number of classes where $m > 1$.
- V_0 = Database (set of data)
- $Y(k, 0)$ = number of training patterns of class C_k in the set V_0
- $Z(0)$ = number of patterns in the set V_0

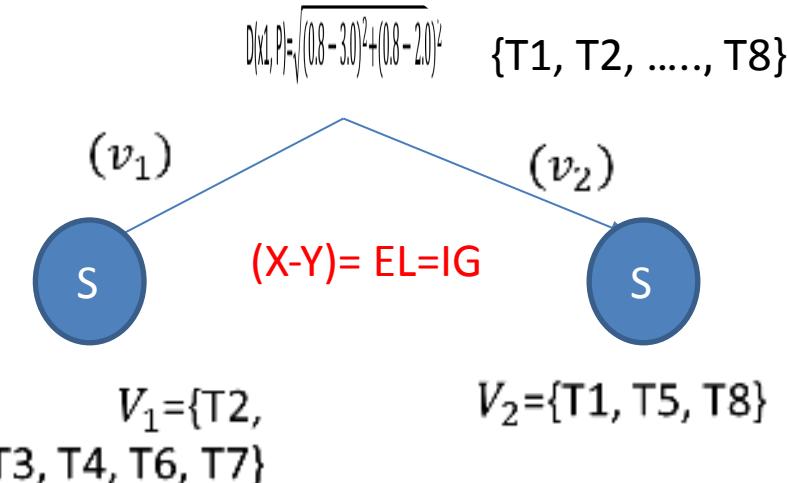
$$Z(0) = \sum_{k=1}^m Y(k, 0)$$

- The probability that a pattern in V_0 belongs to class C_k is

$$\frac{Y(k, 0)}{Z(0)}$$

- The information required to classify a pattern of V_0 into the class C_k , for $1 \leq k \leq m$ is expressed as

$$-\log \frac{Y(k, 0)}{Z(0)}$$



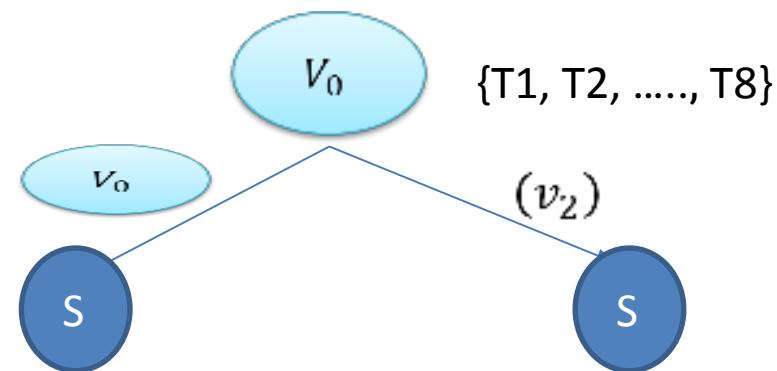
NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Information Gain

- The weighted average information required to classify a pattern of set V_0 into one of the m classes is expressed as

$$I(V_0) = \sum_{k=1}^m \frac{Y(k, 0)}{Z(0)} (-\log \frac{Y(k, 0)}{Z(0)})$$

$I(V_0)$ is called the entropy of the set V_0 .



Density is reverse of distance therefore Local Reachability score LRD

$$\begin{aligned} LRD_A &= \frac{1}{RD_A} \\ &= 1/6.06 = 0.165 \end{aligned}$$

- Similarly for the set V_j ,

$$I(V_j) = \sum_{k=1}^m \frac{Y(k, j)}{Z(j)} (-\log \frac{Y(k, j)}{Z(j)})$$

- The weighted average information required to classify a pattern into one of class k in set V_0 after it has been split by the attribute A into sets V_1 to V_n is given by

$$I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} I(V_j)$$

- $I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right)$
 $I_A(V_0)$ is called the entropy of the attribute A for the set V_0
- The gain in information caused by attribute A splitting set V_0 into sets V_1 to V_n is

$$g_A(V_0) = I(V_0) - I_A(V_0)$$

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Example

Evaluate the entropy $I(V_0)$ of the Professor-student training set.

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTW EAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roast ed	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

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

Therefore,

$$\begin{aligned} I(V_0) &= - \sum_{k=1}^m \frac{Y(k,0)}{Z(0)} \log \frac{Y(k,0)}{Z(0)} \\ &= - \sum_{k=1}^2 \frac{Y(k,0)}{Z(0)} \log \frac{Y(k,0)}{Z(0)} \\ &= - \frac{3}{8} \log \frac{3}{8} - \frac{5}{8} \log \frac{5}{8} \\ &= 0.9544 \end{aligned}$$

Example

Information gain of HABIT

- S and P are the two classes, hence m=2.
- The training set $V_0 = \{T1, T2, T3, \dots, T8\}$.
- $Y(1,0)=3$ (number of patterns in V_0 of class P)
- $Y(2,0)=5$ (number of patterns in V_0 of class S)
- $Z(0)=8$ (number of patterns in V_0)
- $I(V_0)=0.9544$

- $V_1 = \{T1, T2, T3, T5, T7\}$ at node y_1 , where HABIT=gabby.
- $Y(1,1)=2$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=5$ (number of patterns in V_1)

- $V_2 = \{T4, T6, T8\}$ at node y_2 , where HABIT=quiet.
- $Y(1,2)=1$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=3$ (number of patterns in V_2)

$$\begin{aligned}I_{Habit}(V_0) &= - \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \log \frac{Y(k,j)}{Z(j)} \\&= [5/8\{(2/5(-\log2/5)+(3/5(-\log3/5)\}\\&\quad + 3/8\{(1/3(-\log1/3)+(2/3(-\log2/3)\}\\&= 0.9499\end{aligned}$$

$$\begin{aligned}G_{Habit}(V_0) &= \frac{I(V_0) - I_{Habit}(V_0)}{I(V_0)} \\&= (0.9544 - 0.9499) \\&= 0.0100\end{aligned}$$

Example

Information gain of EATS

- S and P are the two classes, hence m=2.
- The training set $V_0 = \{T1, T2, T3, \dots, T8\}$.
- $Y(1,0)=3$ (number of patterns in V_0 of class P)
- $Y(2,0)=5$ (number of patterns in V_0 of class S)
- $Z(0)=8$ (number of patterns in V_0)
- $I(V_0)=0.9544$
- $V_1 = \{T1, T3, T6\}$ at node y_1 , where EATS=baked.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T4, T5, T7, T8\}$ at node y_2 , where EATS=fried.
- $Y(1,2)=2$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=4$ (number of patterns in V_2)
- $V_3 = \{T2\}$ at node y_3 , where EATS=roasted.
- $Y(1,3)=1$ (number of patterns in V_3 of class P)
- $Y(2,3)=0$ (number of patterns in V_3 of class S)
- $Z(3)=1$ (number of patterns in V_3)

$$I_{Eats}(V_0) \\ = - \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \log \frac{Y(k,j)}{Z(j)} \\ = 0.5$$

$$G_{Eats}(V_0) = \frac{I(V_0) - I_{Eats}(V_0)}{I(V_0) - I_{Eats}(V_0)} \\ = (0.9544 - 0.5) \\ = 0.4544$$

Example

information gain of FOOTWEAR

- S and P are the two classes, hence m=2.
- The training set $V_0 = \{T1, T2, T3, \dots, T8\}$.
- $Y(1,0)=3$ (number of patterns in V_0 of class P)
- $Y(2,0)=5$ (number of patterns in V_0 of class S)
- $Z(0)=8$ (number of patterns in V_0)
- $I(V_0)=0.9544$
- $V_1 = \{T1, T5, T8\}$ at node y_1 , where FOOTWEAR=clogs.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T2, T3, T4, T6, T7\}$ at node y_2 , where FOOTWEAR=sandals.
- $Y(1,2)=3$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=5$ (number of patterns in V_2)

Therefore,

$$\begin{aligned} I_{Footwear}(V_0) \\ &= - \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \log \frac{Y(k,j)}{Z(j)} \\ &= 0.6066 \end{aligned}$$

$$\begin{aligned} G_{Footwear}(V_0) &= \frac{I(V_0) - I_{Footwear}(V_0)}{I(V_0)} \\ &= (0.9544 - 0.6066) \\ &= 0.3478 \end{aligned}$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabb y	Bake d	Clogs	S
T2	Gabb y	Roas ted	Sanda ls	P
T3	Gabb y	Bake d	Sanda ls	S
T4	Quiet	Fried	Sanda ls	P
T5	Gabb y	Fried	Clogs	S
T6	Quiet	Bake d	Sanda ls	S
T7	Gabb y	Fried	Sanda ls	P
T8	Quiet	Fried	Clogs	S

Example

$$G_{Habit}(V_0) = 0.0100$$

$$G_{Eats}(V_0) = 0.4544$$

$$G_{Footwear}(V_0) = 0.3478$$

Measures of feature redundancy:

1. Similarity-based measure

- i. Pearson Correlation Coefficient
- ii. Spearman's Correlation
- iii. Kendall's Tau
- iv. Jaccard Index/Coefficient
- v. Simple Matching Coefficient (SMC)
- vi. Cosine Similarity

2. Distance-based measure

- i. Euclidean distance
- ii. Manhattan distance

Correlation-based measure: Measures linear dependency between two random variables.

Feature Redundancy

Pearson Correlation Coefficient: measures linear dependency between two random variables

$$\alpha = \frac{cov(F1, F2)}{\sqrt{var(F1) \cdot var(F2)}}$$

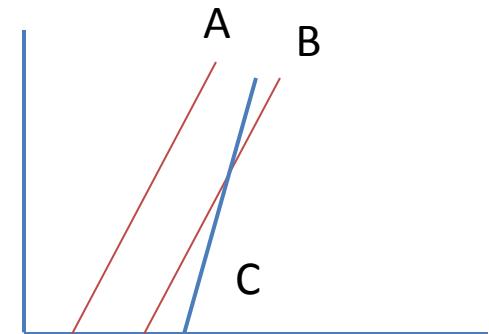
$$cov(F1, F2) = \sum \{(F1_i - F1').(F2_i - F2')\} / (n - 1)$$

$$var(F1) = \sum (F1_i - F1')^2 / (n - 1), \text{ where } F1' = \frac{1}{n} \sum F1_i$$

$$var(F2) = \sum (F2_i - F2')^2 / (n - 1), \text{ where } F2' = \frac{1}{n} \sum F2_i$$

It ranges between +1 and -1

Covariance is a measure of the relationship between two random variables. The metric evaluates how much – to what extent – the variables change together. In other words, it is essentially a measure of the variance between two variables



Spearman's Correlation Coefficient: measures linear dependency between two random variables. It uses rank of each value. Data are represented as

$$x = x^r$$

$$y = y^r$$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

$$S(F1, F2) = \frac{cov(F1, F2)}{\sqrt{var(F1).var(F2)}}$$

$$cov(F1, F2) = \sum\{(F1_i^r - F1'^r) \cdot (F2_i^r - F2'^r)\} / (n-1)$$

$$var(F1) = \sum(F1_i^r - F1'^r)^2 / (n-1) \quad \text{where } F1'^r = \frac{1}{n} \sum F1_i^r$$

$$var(F2) = \sum(F2_i^r - F2'^r)^2 / (n-1), \quad \text{where } F2'^r = \frac{1}{n} \sum F2_i^r$$

It ranges between +1 and -1

If $S > P$ it means that we have a monotonic relationship, not a linear relationship.

Feature Redundancy

Kendall's Tau: measures linear dependency between two random variables. It uses rank of each value. Kendall's Tau has smaller variability when using larger sample sizes. However, Spearman's measure is more computationally efficient, as Kendall's Tau is $O(n^2)$ and Spearman's correlation is $O(n\log(n))$.

Data are represented as

$$x = x^r$$

$$y = y^r$$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

It ranges between +1 and -1

Feature Redundancy

$$\text{Kendall's Tau} = (C - D) / (C + D)$$

Feature1: 1 2 3 4 5 6 7 8 9 10 11 12

Feature 2: 1 2 4 3 6 5 8 7 10 9 12 11

Step1: Make a table of rankings. The rankings for Feature 1 should be in ascending order

Feature1: 1 2 3 4 **12** 6 7 8 9 10 11 **5**

Feature 2: 1 2 4 3 6 5 8 7 10 9 12 11

Feature1: 1 2 3 4 **5** 6 7 8 9 10 11 **12**

Feature 2: 1 2 4 3 **11** 5 8 7 10 9 12 **6**

Feature1	Feature 2
1	1
2	2
3	4
4	3
5	6
6	5
7	8
8	7
9	10
10	9
11	12
12	11

Feature Redundancy

Step 2: Count the number of concordant pairs,

using the second column. Concordant pairs are how many larger ranks are below a certain rank.

For example, the first rank in the second Feature's column is a "1", so all 11 ranks below it are larger.

Feature 1	Feature 2	Concor- dant	Discord- ant
1	1	11	
2	2	10	
3	4	8	
4	3	8	
5	6	6	
6	5	6	
7	8	4	
8	7	4	
9	10	2	
10	9	2	
11	12	0	
12	11		

Feature Redundancy

Step 3: Count the number of discordant

pairs and insert them into the next column.

The number of discordant pairs is similar to Step 2, only you're looking for smaller ranks, not larger ones.

Feature 1	Feature 2	Concordant	Discordant
1	1	11	0
2	2	10	0
3	4	8	1
4	3	8	0
5	6	6	1
6	5	6	0
7	8	4	1
8	7	4	0
9	10	2	1
10	9	2	0
11	12	0	1
12	11		

Feature Redundancy

Step 4: Sum the values in the two columns:

Step 5: Insert the totals into the formula:

$$\text{Kendall's Tau} = (C - D / C + D) = (61 - 5) / (61 + 5) = 56 / 66 = .85.$$

The Tau coefficient is .85, suggesting a strong relationship between the rankings.

Feature 1	Feature 2	Concordant	Discordant
1	1	11	0
2	2	10	0
3	4	8	1
4	3	8	0
5	6	6	1
6	5	6	0
7	8	4	1
8	7	4	0
9	10	2	1
10	9	2	0
11	12	0	1
12	11		
Total		61	5

Feature Redundancy

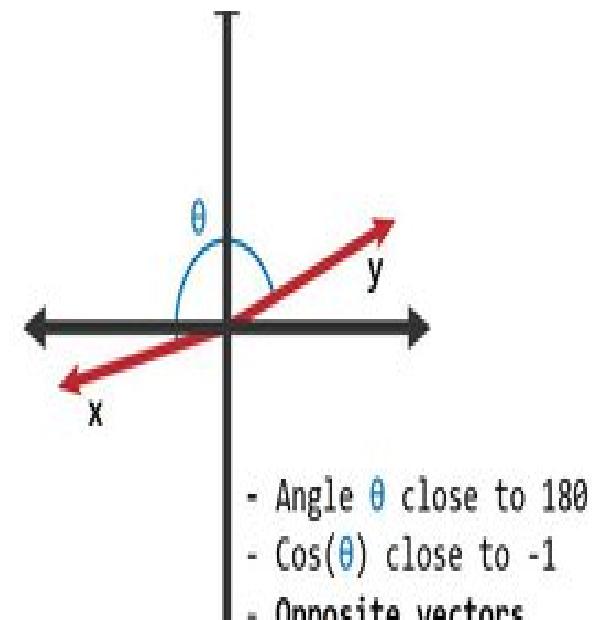
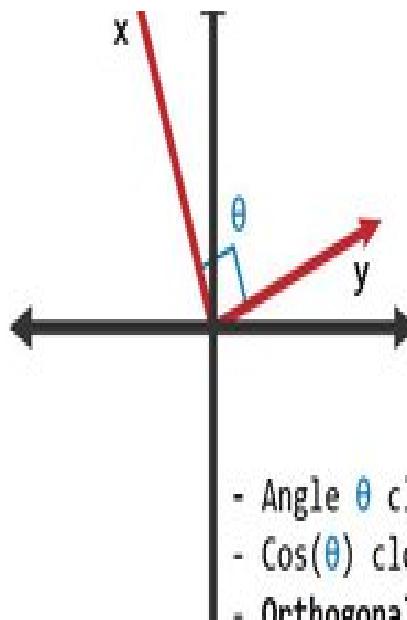
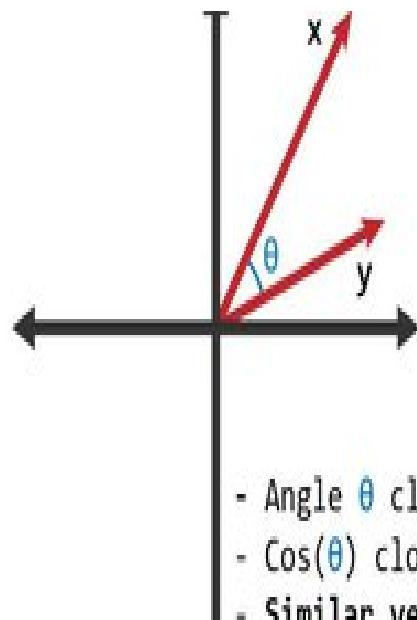
Perfect Correlation

Tau = $(66 - 0) / (66 + 0) = 1$, which is (as we expect) perfect agreement.

Feature 1	Feature 2	Concordant	Discordant
1	1	11	0
2	2	10	0
3	3	9	0
4	4	8	0
5	5	7	0
6	6	6	0
7	7	5	0
8	8	4	0
9	9	3	0
10	10	2	0
11	11	1	0
12	12		
Total		66	0

Feature Redundancy

Cosine Similarity: The cosine similarity calculates the cosine of the angle between two vectors. The cosine similarity can take on values between -1 and +1. If the vectors point in the exact same direction, the cosine similarity is +1. If the vectors point in opposite directions, the cosine similarity is -1.



Feature Redundancy

Cosine Similarity: The cosine similarity calculates the cosine of the angle between two vectors. The cosine similarity can take on values between -1 and +1. If the vectors point in the exact same direction, the cosine similarity is +1. If the vectors point in opposite directions, the cosine similarity is -1.

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=0}^{n-1} x_i y_i}{\sqrt{\sum_{i=0}^{n-1} (x_i)^2} \times \sqrt{\sum_{i=0}^{n-1} (y_i)^2}}$$

$\mathbf{x}=(2, 4, 0, 0, 2, 1, 3, 0, 0)$ and $\mathbf{y}=(2, 1, 0, 0, 3, 2, 1, 0, 1)$

x. y=2*2+4*1+0*0+0*0+2*3+1*2+3*1+0*0+0*1=19

$$\|\mathbf{x}\| = \sqrt{2^2 + 4^2 + 0^2 + 0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} = 5.83$$

$$\|\mathbf{y}\| = \sqrt{2^2 + 1^2 + 0^2 + 0^2 + 3^2 + 2^2 + 1^2 + 0^2 + 1^2} = 4.47$$

$$\text{Cos } (\mathbf{x}, \mathbf{y}) = \frac{19}{5.83 \times 4.47} = 0.729$$

Feature Redundancy

Jaccard Coefficient: measures similarity between two features having binary values.

$$J = \frac{n_{11}}{n_{01} + n_{10} + n_{11}}$$

F1	0	1	1	0	1	0	1	0
F2	1	1	0	0	1	0	0	0

$$J = \frac{2}{1+2+2} = 0.4$$

$$\text{Jaccard distance} = 1 - J = 1 - 0.4 = 0.6$$

Feature Redundancy

Simple Matching Coefficient (SMC): Same as Jaccard coefficient except the fact that it includes a number of cases where both the features have a value 0.

$$SMC = \frac{n_{11} + n_{00}}{n_{01} + n_{10} + n_{11} + n_{00}}$$

F1	0	1	1	0	1	0	1	0
F2	1	1	0	0	1	0	0	0

$$SMC = \frac{2+3}{1+2+2+3} = 0.5$$

Distance-Based Measure

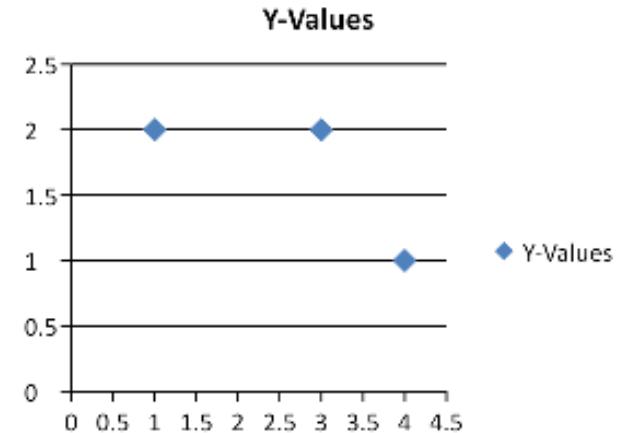
Let $p = (p_1, p_2)$ and $q = (q_1, q_2)$ be two points:

- ✓ City block distance $d(p, q) = |p_1 - q_1| + |p_2 - q_2|$
- ✓ Euclidean distance $d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$
- ✓ Minkowski distance $d(p, q) = (\sum_{i=1}^M |p_i^n - q_i^n|^r)^{\frac{1}{r}}$

For r=1, Minkowski distance = City block distance

For r=2, Minkowski distance = Euclidean distance

	M=1	M=2
1 st	1	2
2 nd	3	2
3 rd	4	1



$$1^{\text{st}} = (1, 2)$$

$$2^{\text{nd}} = (3, 2)$$

$$3^{\text{rd}} = (4, 1)$$

$$\text{Dis}(1^{\text{st}}, 2^{\text{nd}}) = |1-3| + |2-2| = 2$$

DAY-5

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. Embedded approach

Filter Approach:

- Filter methods pick up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance.
- statistical measure used, no learning algorithm;
- These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods.
- (A, B, C, D,E): Filter approach= 5---Wrapper $(5+4+3+2+1)=15$
- Selection of feature is evaluated individually (don't have a dependency on other features) but will lag when a combination of features can lead to increase in the overall performance of the model.

Feature Selection Approaches

- Stopping criteria for selecting the best subset are usually pre-defined by the person training the model such as when the performance of the model decreases or a specific number of features has been achieved.

(A, B, C, D,E): (when the performance of the model decreases)

- A= 0.76
- B= 0.90
- C= 0.65
- D= 0.50
- E= 0.95

{E}=80%; {E, B}= 82%; {E, B, A}=90%; {E, B, A, C}=88%;

(A, B, C, D,E): (a specific number (2) of features has been achieved)

{E, B}= 82%;

- Some of statistical test conducted on features are as follows:
 - i. Pearson's correlation
 - ii. information gain
 - iii. Fisher score
 - iv. Analysis of Variance (ANOVA)
 - v. Chi-Square,
 - VI. Correlation Coefficient
 - VII. Variance Threshold
 - VIII. Mean Absolute Difference (MAD)
 - IX. Dispersion ratio
 - X. Mutual Dependence
 - XI. Relief
 - XII. Missing Value Ratio

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. Embedded approach

Filter Approach: The implementation of filter approach

Set of all features → Selecting the best subset → Learning algorithm → Performance

- Some of statistical test conducted on features are as follows.
 - i. Pearson's correlation
 - ii. information gain
 - iii. Fisher score
 - iv. Analysis of Variance (ANOVA)
 - v. Chi-Square,
 - VI. Correlation Coefficient
 - VII. Variance Threshold
 - VIII. Mean Absolute Difference (MAD)
 - IX. Dispersion ratio
 - X. Mutual Dependence
 - XI. Relief
 - XII. Missing Value Ratio

Feature Selection Approaches: Filter Approach

Fisher score:

- Fisher score is one of the most widely used supervised feature selection methods.
- It selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features.
- The score of the i-th feature S_i will be calculated by Fisher Score,

$$S_i = \frac{\sum n_j (\mu_{ij} - \mu_i)^2}{\sum n_j * p_{ij}^2}$$

where μ_{ij} and p_{ij} are the mean and the variance of the i-th feature in the j-th class, respectively,

- n_j is the number of instances in the j-th class and μ_i is the mean of the i-th feature.
- The features are ranked according to the Fisher Score.

Example

A1	A2	Class
2	0.25	1
5	1.02	0
7	1	0
3	0.75	1
2.5	.6	0
1.98	1	0

Fisher Score of feature A1 is

$$S_{A1} = \frac{\sum n_j(\mu_{ij} - \mu_i)^2}{\sum n_j * p_{ij}^2}$$

$$\mu_{A11} = 2.5$$

$$p_{A11} = 0.25$$

$$\mu_{A10} = 4.12$$

$$p_{A10} = 4.07$$

$$n_1 = 2$$

$$n_0 = 4$$

$$\mu_{A1} = 3.58$$

Fisher Score of feature A2 is

$$S_{A2} = \frac{\sum n_j(\mu_{ij} - \mu_i)^2}{\sum n_j * p_{ij}^2}$$

$$\mu_{A21} = 0.5$$

$$p_{A21} = 0.062$$

$$\mu_{A20} = 0.91$$

$$p_{A20} = 0.031$$

$$n_1 = 2$$

$$n_0 = 4$$

$$\mu_{A2} = 0.77$$

$$S_{A1} = \frac{2(2.5 - 3.58)^2 + 4(4.12 - 3.58)^2}{(2 * 0.25^2) + (4 * 4.07^2)}$$

$$= 0.05$$

$$S_{A2} = \frac{2(0.5 - 0.77)^2 + 4(0.91 - 0.77)^2}{(2 * 0.062^2) + (4 * 0.031^2)}$$

$$= 16.43$$

- The feature A2 has a higher rank than the feature A1
- Hence feature A2 is more important in the prediction process than feature A1

Relief algorithms

There are three Algorithms in the Relief Family:

- **Basic Relief algorithm:** It is limited to classification problems with two classes.
- **ReliefF :** Extension of Relief . Which can deal with multiclass problems.
- **RReliefF:** Then ReliefF was adapted for continuous class (regression) problems resulting in RReliefF algorithm.

However the basic idea in all the three algorithms remains the same.

The core idea is on the basis of how well the attribute can distinguish between instances that are near to each other.

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**

- 1. set all weights $W[A] := 0.0;$
- 2. for $i := 1$ to m do begin
3. randomly select an instance $R_i;$
4. find nearest hit H and nearest miss $M;$
5. for $A := 1$ to a do
6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m;$
7. end;

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0 & ; \text{value}(A, I_1) = \text{value}(A, I_2) \\ 1 & ; \text{otherwise} \end{cases}$$

for nominal attributes

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

for numerical attributes

I_1, I_2, \dots, I_n are examples.

Each example is a vector of attributes $A_i, i = 1, \dots, a$, where a is the number of attributes, and each example has a target value t_j .

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**

- 1. set all weights $W[A] := 0.0;$
- 2. for $i := 1$ to m do begin
3. randomly select an instance $R_i;$
4. find nearest hit H and nearest miss $M;$
5. for $A := 1$ to a do
6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m;$
7. end;

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0 & ; \text{value}(A, I_1) = \text{value}(A, I_2) \\ 1 & ; \text{otherwise} \end{cases}$$

for nominal attributes

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

for numerical attributes

A random instance R_i (line 3) and its two nearest neighbors: one of the same class that R_i belongs to known as nearest hit H and other of the different class known as nearest miss M (line 4).

The whole process is repeated m number of times and m is a user defined parameter.

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Select 2 best attributes.

Let $m = 2$

Step 1 (1): Let all attributes weight be 0 ,

$A=B=C=0$,

Step 2(3) : Row 5 is randomly selected instance. (i.e 6,0,0)

Step 3(4) : Using Manhattan distance

Nearest hit:

Row 4: $|6-8| + |0-3| + |0-1| = 6$

Row 3: $|6-9| + |0-3| + |0-2| = 8$

Row 4 is nearest hit.

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Step 3(4) : Using Manhattan distance

Nearest miss:

Row 2: $|6-5| + |0-1| + |0-0| = 2$

Row 1: $|6-9| + |0-2| + |0-2| = 7$

Row 2 is nearest miss.

Step 4(6) : Update weights of attributes

A,B,C : current weight = 0

$$\mathbf{A} = 0 - ((|6-8|/(9-5))/2) + ((|6-5|/(9-5))/2) = 0 - (0.5/2) + (0.25/2) = \mathbf{-0.1875}$$

$$\begin{aligned}\mathbf{B} &= 0 - ((|0-3|/(3-0))/2) + ((|0-1|/(3-0))/2) \\ &= 0 - (1/2) + (1/6) = \mathbf{-0.33}\end{aligned}$$

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Step 4(6) : Update weights of attributes

A,B,C : current weight = 0

$$C = 0 - ((|0-1|/(2-0))/2) + ((|0-0|/(2-0))/2)$$

$$= 0 - (1/4) + 0 = -0.25$$

Second Iteration:

Step 2: Row 4 is selected randomly.

Step 3:

Row 3 is selected the nearest hit : $|8-9| + |3-3| + |1-2| = 2$

Row 1 is selected the nearest miss : $|8-9| + |3-2| + |1-2| = 3$.

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}$$

Step 4(6) : Update weights of attributes

Current weight:

$$A = -0.1875; \quad B = -0.33; \quad C = -0.25.$$

$$A = -0.1875 - ((|8-9|/(9-5))/2) + ((|8-9|/(9-5))/2) = \textcolor{red}{-0.1875}$$

$$B = -0.33 - ((|3-3|/(3-0))/2) + ((|3-2|/(3-0))/2) = -0.33 - 0 + 0.166 = \textcolor{red}{-0.167}$$

$$C = -0.25 - ((|1-2|/(2-0))/2) + ((|1-2|/(2-0))/2) = \textcolor{red}{-0.25}$$

A and B as our 2 best features

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

Feature Selection Approaches: Filter Approach

Relief algorithms

- **Basic Relief algorithm:**
- 1. set all weights $W[A] := 0.0$;
- 2. for $i := 1$ to m do begin
- 3. randomly select an instance R_i ;
- 4. find nearest hit H and nearest miss M ;
- 5. for $A := 1$ to a do
- 6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
- 7. end;

- The original Relief can deal with nominal and numerical attributes.
- However, it cannot deal with incomplete data and is limited to two-class problems.
- Its extension, which solves these and other problems, is called ReliefF.

	A	B	C	D
1	9	2	2	0
2	5	1	0	0
3	9	3	2	1
4	8	3	1	1
5	6	0	0	1

1. Filter Approach
2. **Wrapper Approach**
3. Hybrid Approach
4. Embedded approach

Wrapper Approach: (A, B, C, D,E): --Wrapper $(5+4+3+2+1)=15$

- Wrappers require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset.
- It follows a greedy search approach by evaluating all the possible combinations of features against some evaluation criterion.
- Learning algorithms are used as black box.
- Computationally very expensive, however performance is superior than filter approach;

1. Filter Approach
2. **Wrapper Approach**
3. Hybrid Approach
4. Embedded approach

Wrapper Approach: learning algorithms are used as black box. Computationally very expensive, however performance is superior than filter approach;

Sensitivity Analysis by ANN: Input set= {a, b, c, d, e}

Subset={a, b, c}= 96%

Subset= {a, b, c, d}=94%

Indicates feature d has negative impact, we can drop it

Subset= {a, b, c, e}= 97%

Indicates feature e has positive impact, we can add it.

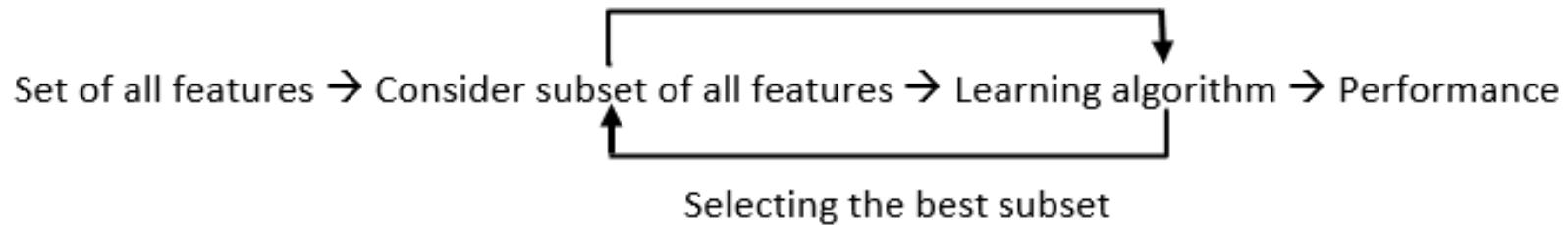
Hence final subset consists of {a, b, c, e}

a	b	c	d	e	class
12	25	42	45	23	1
32	23	23	86	51	1
56	12	14	401	23	0
30	15	63	47	21	0
45	20	54	98	20	1

Feature Selection Approaches

Wrapper Approach:

- The implementation of the wrapper approach is given below:



1. Filter Approach
2. **Wrapper Approach**
3. Hybrid Approach
4. Embedded approach

Some of the algorithm under wrapper approach are as follows:

- **Exhaustive Feature Selection**
- **Forward Feature Selection**
- **Backward Feature Elimination**
- **Recursive Feature Elimination**
- **Bi-directional elimination**

The table gives subset of 3 features out of 5. This procedure is impractical as if we want to choose 12 features out of 24, 2.7 million feature subsets must be evaluated.

Sl. No	F1	F2	F3	F4	F5
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	0	1	1	1	0
5	1	0	0	1	1
6	1	0	1	0	1
7	1	0	1	1	0
8	1	1	0	0	1
9	1	1	0	1	0
10	1	1	1	0	0

Sequential Forward Selection

- First, the best single feature is selected
- Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.
- Next, triplets of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
- This procedure continues until a predefined number of features are selected.

Suppose we are interested to select 3 features out of 5. Feature set= {F1, F2, F3, F4, F5}

Starts with S= {};

1st iteration:

Say ANN is used: {F1} = 60%, {F2} = 52%, {F3} = 52%, {F4} = 53%, {F5} = 70%

S= {F5}

2nd iteration:

{F5, F1}= 85%, {F5, F2}= 82%, {F5, F3}= 80% {F5, F4}= 82%

S= {F5, F1}

3rd iteration:

{F5, F1, F2}= 88%, {F5, F1, F3}= 80%, {F5, F1, F4}= 90%,

Final subset= {F5, F1, F4}

Sequential Forward Selection

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \arg \max_{x \notin Y_k} J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+$; $k = k + 1$
4. Go to 2

- **Drawback: once selected can not be dropped**

Sequential Backward Selection

- First, the criterion function is computed for all n features.
- Then, each feature is deleted one at a time, the criterion function is computed for all subsets with n-1 features, and the worst feature is discarded.
- Next, each feature among the remaining n-1 is deleted one at a time, and the worst feature is discarded to form a subset with n-2 features.
- This procedure continues until a predefined number of features are left.

Suppose we are interested to select 3 features out of 5. Feature set= {F1, F2, F3, F4, F5}

Starts with S= {F1, F2, F3, F4, F5}= **95%**

1st iteration: Say ANN is used: **{F1, F2, F3, F4} = 96%, {F1, F2, F3, F5} = 94%, {F1, F2, F4, F5} = 95%{F1, F3, F4, F5} = 92%, {F2, F3, F4, F5} = 91%**

S= {F1, F2, F3, F4}

2nd iteration: **{F1, F2, F3} = 95%, {F1, F2, F4} = 98%, {F1, F3, F4} = 92%, {F2, F3, F4} = 90%**

S= {F1, F2, F4}

Sequential Backward Selection

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg \max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-$; $k = k + 1$
4. Go to 2

Drawback: once dropped cannot be taken back

Bi-directional Selection

BDS applies SFS and SBS simultaneously:

SFS is performed from the empty set.

SBS is performed from the full set.

To guarantee that SFS and SBS converge to the same solution:

Features already selected by SFS are not removed by SBS

Features already removed by SBS are not added by SFS

$$S = \{F_1, F_2, F_3, F_4, F_5\}$$

Desired number of features = 3

SFS: $F_1 = 60\%$, $F_2 = 62\%$, $F_3 = 80\%$, $F_4 = 50\%$, $F_5 = 55\%$

$$S_1 = \{F_3\}$$

$$S = \{F_1, F_2, F_4, F_5\} = 86\%$$

SBS: $\{F_1, F_2, F_4\} = 80\%$; $\{F_1, F_2, F_5\} = 82\%$; $\{F_1, F_4, F_5\} = 85\%$; $\{F_2, F_4, F_5\} = 88\%$

$$S = \{F_2, F_4, F_5\}$$

SFS: $F_3 F_2 = 82\%$; $F_3 F_4 = 85\%$; $F_3 F_5 = 80\%$

$$S_2 = \{F_3, F_4\}$$

SBS: $\{F_2, F_5\} = 60\%$; F_5 is deleted

$$S = \{F_2\}$$

SFS: $\{F_2, F_3, F_4\} = 90\%$

Final Subset (S-F) = $\{F_2, F_3, F_5\}$

1. Start SFS with $Y_F = \{\emptyset\}$

2. Start SBS with $Y_B = X$

3. Select the best feature

$$x^+ = \arg \max_{\substack{x \notin Y_{F_k} \\ x \in F_{B_k}}} J(Y_{F_k} + x)$$

$$Y_{F_{k+1}} = Y_{F_k} + x^+$$

4. Remove the worst feature

$$x^- = \arg \max_{\substack{x \in Y_{B_k} \\ x \notin Y_{F_{k+1}}}} J(Y_{B_k} - x)$$

$$Y_{B_{k+1}} = Y_{B_k} - x^-; k = k + 1$$

5. Go to 3

- The problem of sequential forward and sequential backward can be overcome by “Plus-L, minus-R” selection (LRS).
- However its main limitation is the lack of a theory to help choose the optimal values of L and R.

- The drawback of sequential forward and backward selection is called nesting effect. This nesting effect can be overcome by Sequential Floating Selection.
- The drawback of “Plus-L, minus-R” selection (LRS) is also overcome by Sequential Floating Selection.
 1. Sequential floating forward selection
 2. Sequential floating backward selection

Sequential Floating Forward Selection

- Sequential floating forward selection (SFFS) starts from the empty set.
- After each forward step, SFFS performs backward steps as long as the objective function increases.

Step1: Let k=0

Step2: If $k=\text{desired size}$, terminate; otherwise add the most significant feature to the current sub-set of size k . Let $k=k+1$

Step3: Conditionally, remove the least significant feature from the current subset

Step4: If the current subset is the best subset of size $(k-1)$ found so far, let $k=(k-1)$ and go to Step3. Else return the conditionally removed feature and go to Step2.

Consider the feature set = $\{f_1, f_2, f_3, f_4, f_5\}$

Target is to select subset of 2 features

1. $F=\{\}$
 2. The most significant feature is f_3 ; $F= \{f_3\}$
 3. The least significant feature is f_3 ; $F= \{\}$
 4. Removal of f_3 does not improve performance; Hence $F= \{f_3\}$
 5. The most significant feature is f_2 using SFS; $f_3f_2=70\%$, $f_3f_1=50\%$, $f_3f_4=52\%$, $f_3f_5= 60\%$
 $S=\{f_2, f_3\}$
 6. The least significant feature is f_2 ; $F=\{f_3\}$
 7. Removal of f_2 does not improve performance; Hence $F= \{f_3, f_2\}$
 8. $\{f_2, f_3, f_1\}= 72\%$, $\{f_2, f_3, f_4\}= 71\%$, $\{f_2, f_3, f_5\}= 75\%$
 $S= \{f_2, f_3, f_5\}$
- SBS: $\{f_2, f_3\}= 70\%$, $\{f_2, f_5\}= 65\%$, $\{f_3, f_5\}= 72\%$

Hence the optimal subset is $F= \{f_2, f_3, f_5\}$

Sequential Floating Forward Selection

1. $Y = \{\emptyset\}$

2. Select the best feature

$$x^+ = \arg \max_{x \notin Y_k} J(Y_k + x)$$

$$Y_k = Y_k + x^+; k = k + 1$$

3. Select the worst feature*

$$x^- = \arg \max_{x \in Y_k} J(Y_k - x)$$

4. If $J(Y_k - x^-) > J(Y_k)$ then

$$Y_{k+1} = Y_k - x^-; k = k + 1$$

Go to step 3

Else

Go to step 2

*Notice that you'll need to do book-keeping to avoid infinite loops

Sequential Floating Backward Selection

- Sequential floating backward selection (SFBS) starts from the full set.
- After each backward step, SFBS performs forward steps as long as the objective function increases.

Step1: Let $k=n$ (f_1, f_2, f_3, f_4)

Step2: If $k=\text{desired size}$, terminate; otherwise remove the least significant feature from the current subset of size k . Let $k=k-1$. (f_4)

Step3: Conditionally, add the most significant feature from the features not in the current subset. (f_1, f_2, f_3) + (f_4)

Step4: If the current subset is the best subset of size $(k+1)$ found so far, let $k=(k+1)$ and go to Step3. Else remove the conditionally added feature and go to Step 2.
((f_1, f_2, f_3) and go to Step 2)

Sequential Floating Backward Selection

Step1: Let k=n (f1, f2, f3, f4)

Step2: If k=desired size, terminate; otherwise remove the least significant feature from the current subset of size k. Let k=k-1. (f4)

Step3: Conditionally, add the most significant feature from the features not in the current subset. (f1, f2, f3) + (f4)

Step4: If the current subset is the best subset of size (k+1) found so far, let k=(k+1) and go to Step3. Else remove the conditionally added feature and go to Step 2.

((f1, f2, f3) and go to Step 2)

Thursday, October 27, 2022

Consider the feature set = {f1, f2, f3, f4, f5}

Target is to select subset of 2 features

1. SBS: F={f1, f2, f3, f4, f5}= 64%
{f1, f2, f3, f4}= 60%; {f1, f2, f3, f5}= 62; {f1, f2, f4, f5}= 65%; {f1, f3, f4, f5}= 68%; {f2, f3, f4, f5}=55%

SB {f1, f3, f4, f5}= 68%; F= {f2}

2. Most significant feature f2 from set F; adding this to subset does not improve accuracy as {f1, f2, f3, f4, f5}= 64%
3. Say least significant is f5
1. The most significant feature is f3; F= {f3}
2. The least significant feature is f3; F= {}
3. Removal of f3 does not improve performance; Hence F= {f3}
4. The most significant feature is f2 using SFS; f3f2=70%, f3f1=50%, f3f4=52%, f3f5= 60%
S={f2, f3}
6. The least significant feature is f2; F={f3}
7. Removal of f2 does not improve performance; Hence F= {f3, f2}
8. {f2, f3, f1}= 72%, {f2, f3, f4}= 71%, {f2, f3, f5}= 75%

Feature Selection Approaches: Hybrid Approach

1. Filter Approach
2. Wrapper Approach
3. **Hybrid Approach**
4. Embedded approach

Hybrid Approach: takes advantages of both filter and wrapper approaches.

Filter Approach---{S1}---Wrapper Approach----(S2) |S1>S2

Wrapper Approach---{S1}--- Filter Approach-----(S2) |S1>S2

Feature Selection Approaches: Embedded approach

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. **Embedded approach**

Embedded Approach:

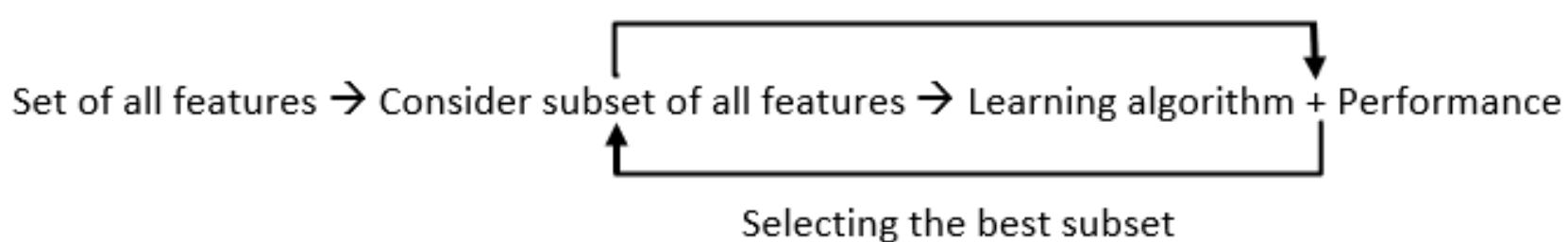
- These methods encompass the benefits of both the wrapper and filter methods by including interactions of features but also maintaining reasonable computational cost.
- Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extracts those features which contribute the most to the training for a particular iteration.
- Similar to wrapper approach but performs feature selection and classification simultaneously.
- These methods are faster like those of filter methods and more accurate than the filter methods and take into consideration a combination of features as well.

Feature Selection Approaches: Embedded approach

1. Filter Approach
2. Wrapper Approach
3. Hybrid Approach
4. **Embedded approach**

Embedded Approach: Some of the algorithms under embedded approach are given below:

- **LASSO Regularization (L1)**
- **Random Forest Importance**
- **Extra Tree Classifier**



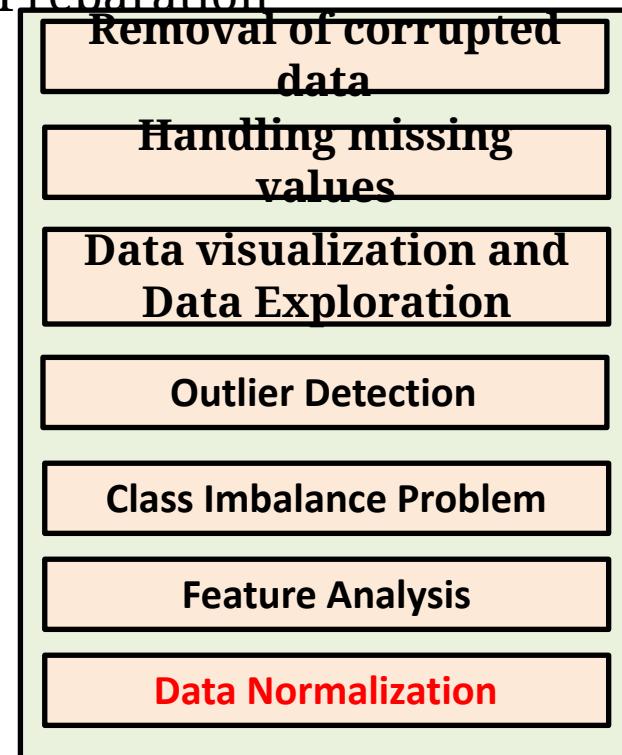
Feature Selection Methods

How to choose a Feature Selection Method (filter-based feature selection)

Input Variable	Output Variable	Feature Selection technique
Numerical	Numerical	<ul style="list-style-type: none">• Pearson's correlation coefficient (For linear Correlation).• Spearman's rank coefficient (for non-linear correlation).
Numerical	Categorical	<ul style="list-style-type: none">• ANOVA correlation coefficient (linear).• Kendall's rank coefficient (nonlinear).
Categorical	Numerical	<ul style="list-style-type: none">• Kendall's rank coefficient (linear).• ANOVA correlation coefficient (nonlinear).
Categorical	Categorical	<ul style="list-style-type: none">• Chi-Squared test (contingency tables).• Mutual Information

Machine Learning Model

Pre-processing/ Data Preparation



Data Normalization

Changes the values to a common scale.

Normalization gives equal weights/importance to each variable so that no single variable steers model performance in one direction just because they are bigger numbers.

For machine learning, every dataset does not require normalization. It is required only when features have different ranges.

It improves (significantly) the performance of some machine learning algorithms and does not work at all for others.

Age	Salary	Experience
30	200000	H
50	500000	H
60	20000	L
100	70000	L

Data Normalization

1. **Min-max normalization** is the simplest of all methods.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

X1	X2
10	20
5	10
6	12
7	15
25	12

$$(10-5)/(25-5)=5/20= 0.4$$

$$(5-5)/(25-5)= 0/20=0$$

$$(6-5)/(25-5)=1/20=0.05$$

$$(7-5)/(25-5)=2/20=0.10$$

$$(25-5)/(25-5)=20/20=1$$

2. **Mean normalization** uses the mean of the observations in the transformation process

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

X1	X2
0.4	1
0	0
0.05	0.2
0.10	0.5
1	0.2

3. **Z-score normalization/standardization** uses Z-score and is widely used in machine learning algorithm.

$$z = \frac{x - \mu}{\sigma}$$

z is the standard score, μ is the population mean and σ is the population standard deviation

Data Normalization

1. **Min-max normalization** is the simplest of all methods.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

X1	X2
10	20
5	10
6	12
7	15
25	12

$$(10-10.6)/(25-5)=-.6/20=-0.03$$

$$(5-10.6)/(25-5)=-5.6/20=-0.28$$

$$(6-10.6)/(25-5)=-4.6/20=-0.23$$

$$(7-10.6)/(25-5)=-3.6/20=-0.18$$

$$(25-10.6)/(25-5)=14.4/20=0.72$$

2. **Mean normalization** uses the mean of the observations in the transformation process

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

X1	X2
-0.03	
-0.28	
-0.23	
-0.18	
0.72	

3. **Z-score normalization/standardization** uses Z-score and is widely used in machine learning algorithm.

$$z = \frac{x - \mu}{\sigma}$$

z is the standard score, μ is the population mean and σ is the population standard deviation

Data Normalization

3. Z-score normalization/standardization:

$$\text{New value} = (x - \mu) / \sigma$$

where:

x : Original value

μ : Mean of data

σ : Standard deviation of data

$$m = \frac{\text{sum of the terms}}{\text{number of terms}} = 21.2$$

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} = 29.8$$

- New value = $(x - \mu) / \sigma$
- New value = $(3 - 21.2) / 29.8$
- New value = -0.61

Data	Z-Score Normalized Value
3	-0.61
5	-0.54
5	-0.54
8	-0.44
9	-0.41
12	-0.31
12	-0.31
13	-0.28
15	-0.21
16	-0.17
17	-0.14
19	-0.07
22	0.03
24	0.09
25	0.13
134	3.79

Data Normalization

3. Z-score normalization/standardization

The mean of the normalized values is **0** and the standard deviation of the normalized values is **1**.

The normalized values represent the number of standard deviations that the original value is from the mean.

For example:

- The first value in the dataset is **0.61** standard deviations below the mean.
- The second value in the dataset is **0.54** standard deviations below the mean.
- The last value in the dataset is **3.79** standard deviations above the mean.

Benefits:

The benefit of performing this type of normalization is that the clear outlier in the dataset (134) has been transformed in such a way that it's no longer a massive outlier.

Data Normalization

Normalization vs Standardization

Normalization	Standardization
Minimum and maximum value of features are used for scaling.	Mean and standard deviation is used for scaling.
It is used when features are of different scales.	It is used when we want to ensure zero mean and unit standard deviation.
Scales values between [0, 1] or [-1, 1].	It is not bounded to a certain range.
It is really affected by outliers.	It is much less affected by outliers.
It is useful when we don't know about the distribution.	It is useful when the feature distribution is Normal or Gaussian.
It is often called as Scaling Normalization.	It is often called as Z-Score Normalization.

Data Normalization

The Robust Scaling:

- we scale each feature of the data set by subtracting the **median** and then dividing by the **interquartile range**.
- The **interquartile range (IQR)** is defined as the difference between the **third and the first quartile** and represents the central 50% of the data. Mathematically the robust scaler can be expressed as:

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

- where **Q1(x)** is the **first quartile** of the attribute **x**, **Q2(x)** is the **median**, and **Q3(x)** is the **third quartile**.
- This method comes in handy when working with data sets that contain many outliers because it uses statistics that are robust to outliers

Data Normalization

The Robust Scaling:

$$x_{rs} = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

$$Q_2(x) = \frac{4 + 5}{2} = 4.5$$

$$Q_3(x) = 6.5$$

$$Q_1(x) = 2.5$$

$$1 = \frac{1 - 4.5}{6.5 - 2.5} = -3.5 / 3.5 = -1$$

$$30 = \frac{30 - 4.5}{6.5 - 2.5} = 25.5 / 3.5 = 7.285714$$

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

	variable1	variable2
0	-1.000000	-1.000000
1	-0.714286	-0.714286
2	-0.428571	-0.428571
3	-0.142857	-0.142857
4	0.142857	0.142857
5	0.428571	0.428571
6	0.714286	0.714286
7	7.285714	1.000000

Data Normalization

Min-max normalization:

The **min-max scaling** shifts the variable 1 towards 0 due to the presence of an **outlier** as compared with variable 2 where the points are evenly distributed in a range from 0 to 1.

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

	variable1	variable2
0	0.000000	0.000000
1	0.034483	0.142857
2	0.068966	0.285714
3	0.103448	0.428571
4	0.137931	0.571429
5	0.172414	0.714286
6	0.206897	0.857143
7	1.000000	1.000000

Data Normalization

The maximum absolute scaling:

The **maximum absolute scaling** rescales each feature **between -1 and 1** by dividing every observation by its maximum absolute value.

$$x_{scaled} = \frac{x}{\max(|x|)}$$

$$x_{11} = \frac{1}{30} = 0.03$$

$$x_{12} = \frac{2}{30} = 0.06$$

$$x_{25} = \frac{5}{8} = 0.625$$

	var1	var2
0	0.033	0.125
1	0.066	0.25
2	0.1	0.375
3	0.133	0.5
4	0.166	0.625
5	0.2	0.75
6	0.233	0.875
7	1.	1.

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

Data Normalization

Z-score normalization:

[-0.71023874, -1.52752523],
[-0.59660054, -1.09108945],
[-0.48296234, -0.65465367],
[-0.36932414, -0.21821789],
[-0.25568595, 0.21821789],
[-0.14204775, 0.65465367],
[-0.02840955, 1.09108945],
[2.585269

	variable1	variable2
0	1	1
1	2	2
2	3	3
3	4	4
4	5	5
5	6	6
6	7	7
7	30	8

Data Normalization

Gradient Descent Based Machine learning algorithms like [linear regression](#), [logistic regression](#), [neural network](#) etc *converge more quickly towards the minima if features are on a similar scale.*

Distance based Machine learning algorithms like [KNN](#), [K-means](#), and [SVM](#) most drastically improve the performance *minima if features are on a similar scale.*

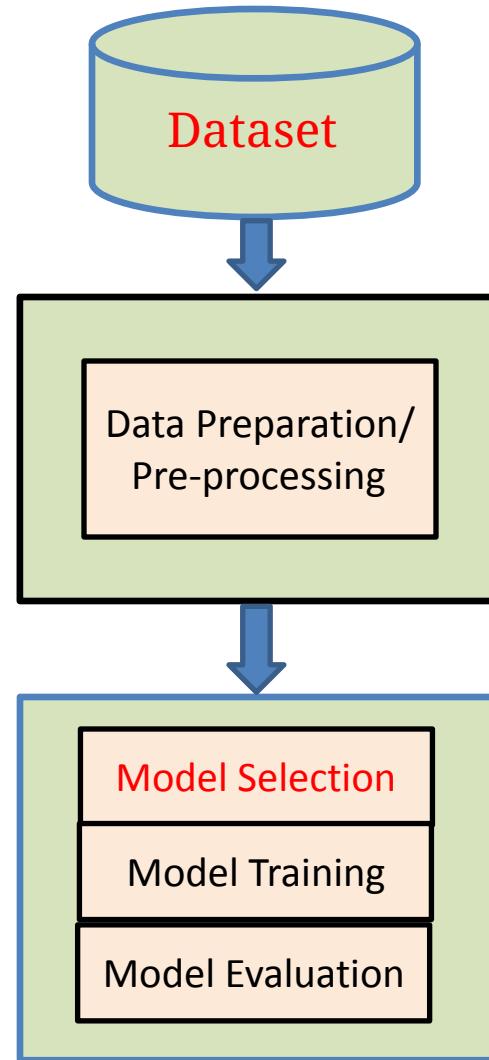
Tree based Machine learning algorithms like DT are insensitive to data normalization.

Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. K-Nearest Neighbors and Neural Networks.

Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true.

Data normalization reduces the variance and applies equal weights to all features; therefore, a lot of important information is lost in the process.

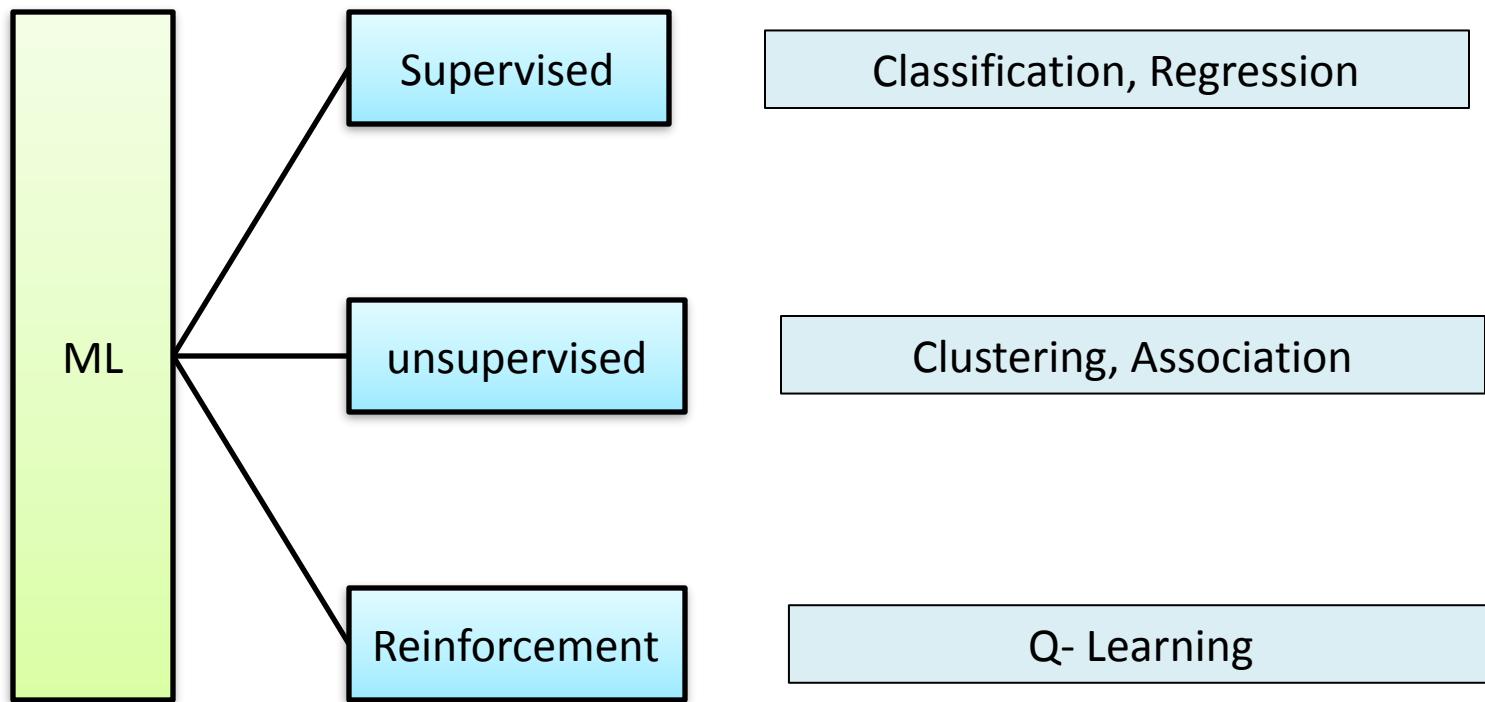
ML Model



Modelling & Evaluation

MODEL SELECTION

Classification of ML Algorithm



MODEL SELECTION

Classification of ML Algorithms

Supervised

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- K- Nearest Neighbour (K-NN)
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

MODEL SELECTION

Classification of ML Algorithms

Unsupervised

- Clustering
 - K-Means
 - K-Mediod
 - CURE
 - BIRCH
- Association
 - Apriori Algorithm
 - Predictive Apriori Algorithm
 - Tertius Algorithm
 - E clat

Patterns	Value of attributes		
	A1	A2	A3
X1	1	1	3
X2	2	3	6
X3	3	1	2
X4	4	4	2
X5	5	2	1

MODEL SELECTION

Classification of ML Algorithms

Reinforcement

- Markov Decision Process (MDP)
- Q learning: Deep-Q-Neural Network (DQN)
- State Action Reward State Action (SARSA)

BP	Heart Beat	Weight	Feedback
120	70	50	reward
125	65	60	penalty
130	59	52	penalty
150	78	70	penalty
135	66	85	reward
125	75	82	reward
120	76	90	reward

MODEL SELECTION

The most important two factors to select the model for solving a machine learning problem are

- **The kind of problem we want to solve using machine learning**
- **The nature of the underlying data.**

**The kind of problem we want to solve
using machine learning**

Prediction of categorical values or discrete values (classification)

Black Box: k-NN, Naïve Bayes, ANN, SVM, Random Forest

White Box: Decision Tree, Rule extraction from Neural Network

BP	Heart Beat	Weigh t	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N

```
if (credit_history = 'existing paid'  
and credit_amount <=12204 )  
then class="good"  
else class="bad"
```

MODEL SELECTION

The kind of problem we want to solve
using machine learning

Prediction of continuous values (Regression)

Linear Regression, Logistic Regression, Polynomial Regression, ANN, Ridge Regression, LASSO Regression, Elastic Net Regression

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Descriptive: basket analysis for transactional data

Clustering

Spherical Shape: k-means, k-mediod

Non Spherical Shape: Clustering Using Representatives (CURE)

MODEL SELECTION

The most important two factors to select the model for solving a machine learning problem are

- The kind of problem we want to solve using machine learning
- The nature of the underlying data

The nature of the underlying data

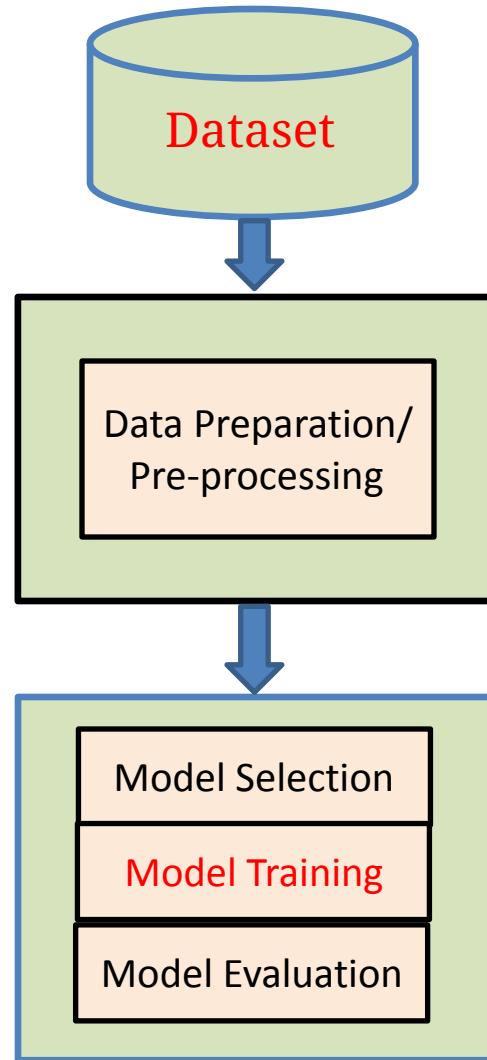
Various statistical measures: mean, median, variance, correlation between variables

visualization tools: Histograms, scatterplot

The training data size is an important factor to be considered, if the training dataset is small: Naïve Bayes

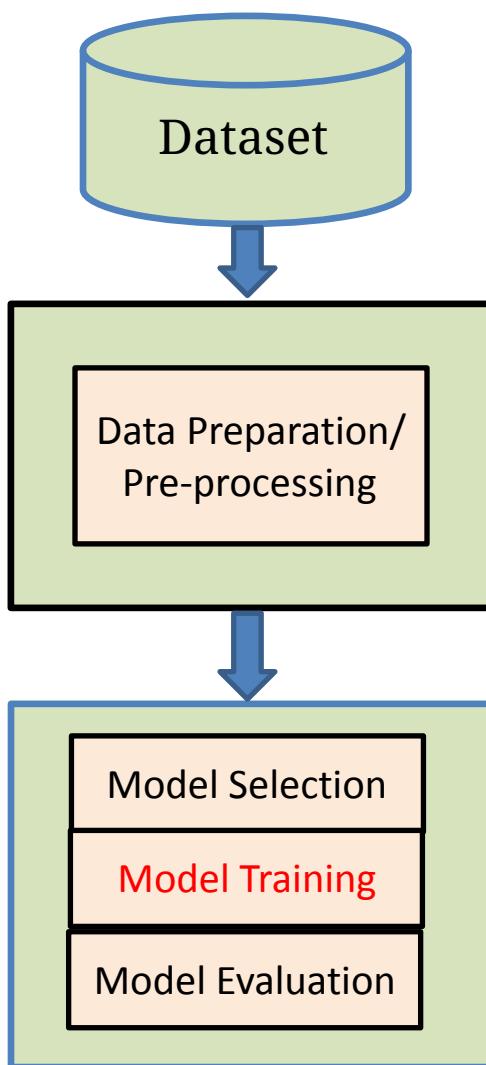
If the training dataset is large: Logistic Regression, SVM, ANN

ML Model



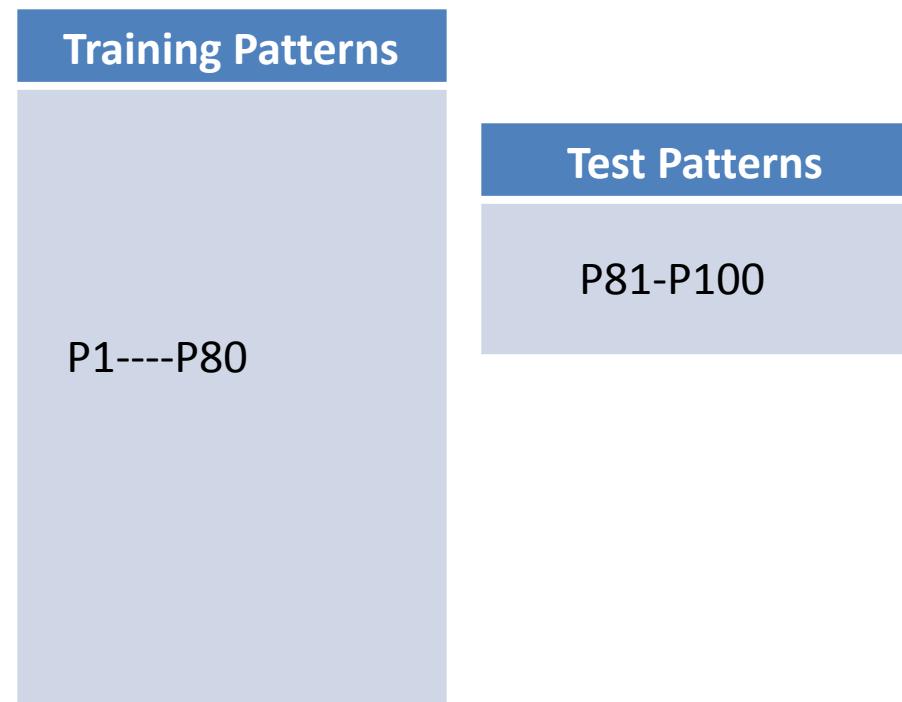
Modelling & Evaluation

Model Training



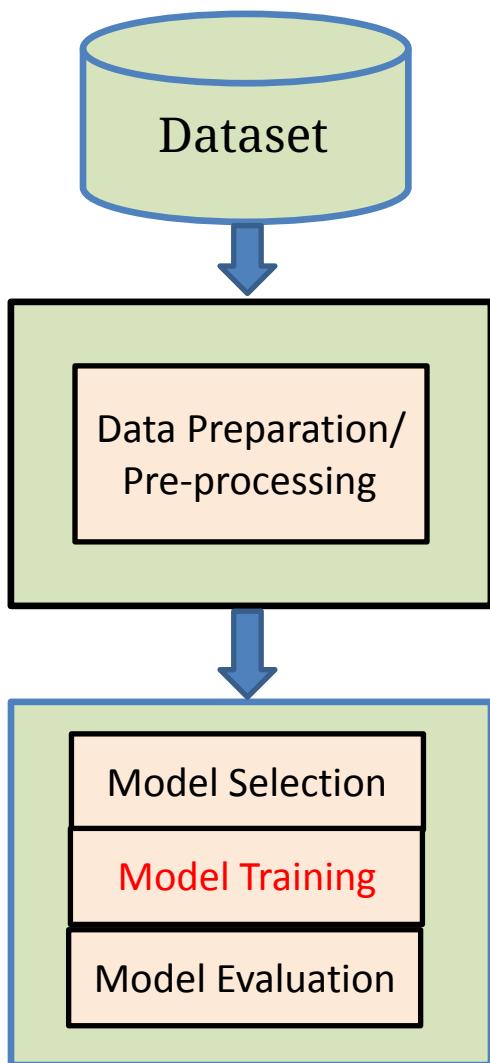
Training for supervised learning: i. Holdout method

- Partition can be: 80-20 or 70-30
 - Sometimes partition into 3 partitions: training, validation, testing
 - **Suffers extremely for imbalance data**
1. Biased Model
 2. Erroneous Model



Modelling & Evaluation

Model Training



i. Holdout method

Advantages:

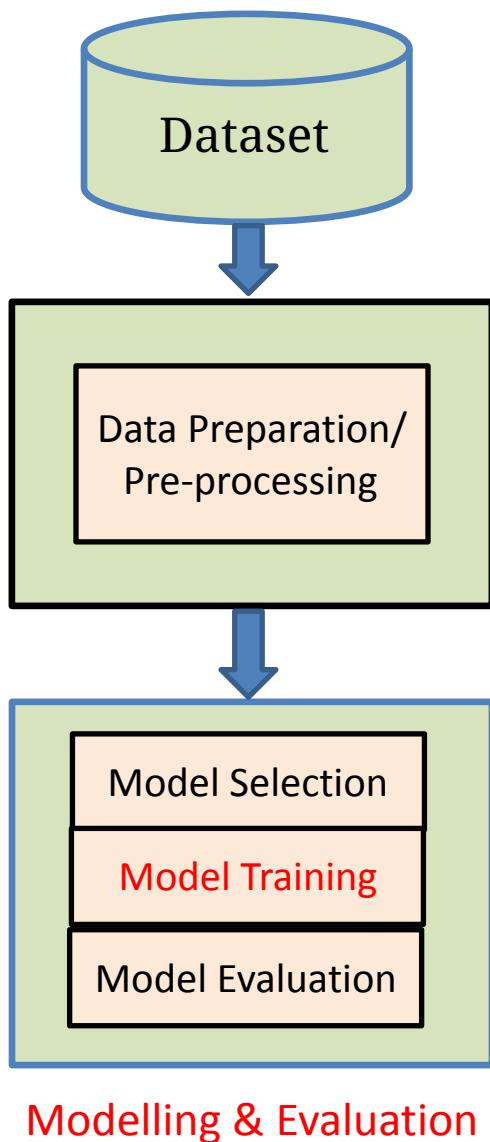
- Its time complexity is less.
- Therefore the hold-out method is good to use when we have a very large dataset.
- An initial model can be built.

Disadvantages:

- The hold-out method score dependent on how the data is split into train and test sets.
- It is less generalized.
- It may be suffering from overfitting problem.
- It may be suffering from imbalanced dataset

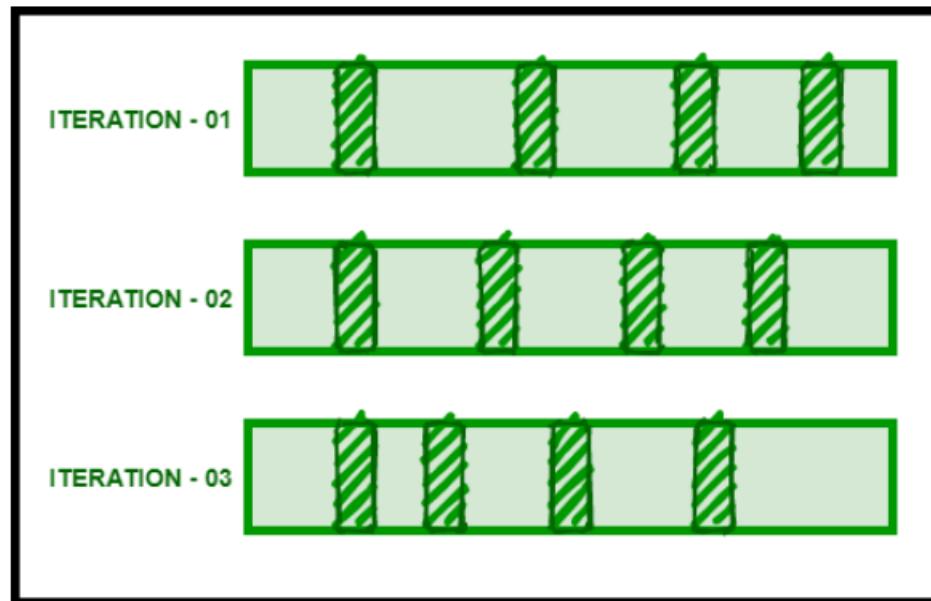
Modelling & Evaluation

Model Training



Training for supervised learning: **ii. Repeated Holdout method**

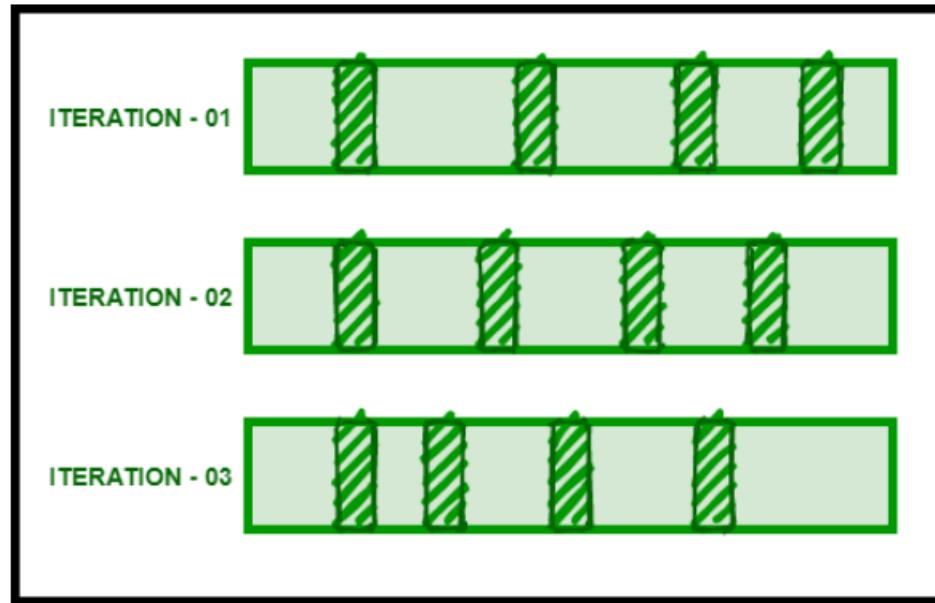
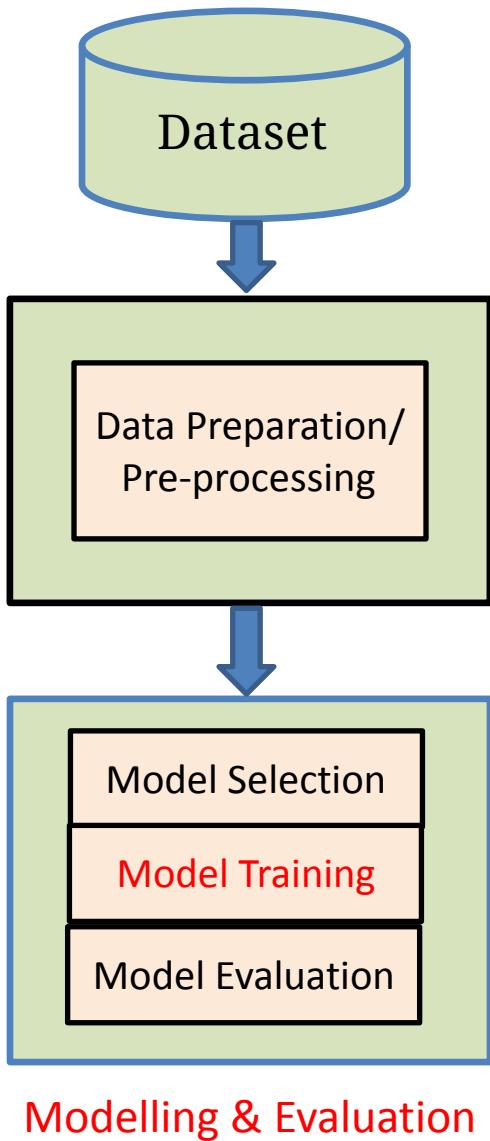
- it is the repeated execution of the holdout method.
- This method can be repeated — ‘K’ times/iterations.
- Random sampling of the dataset is employed.
- Let we repeat the holdout method for 3 iterations.



The shaded portions represent test sets and the unshaded portions training sets.

Model Training

ii. Repeated Holdout method



Accuracy of

iteration 01= S1

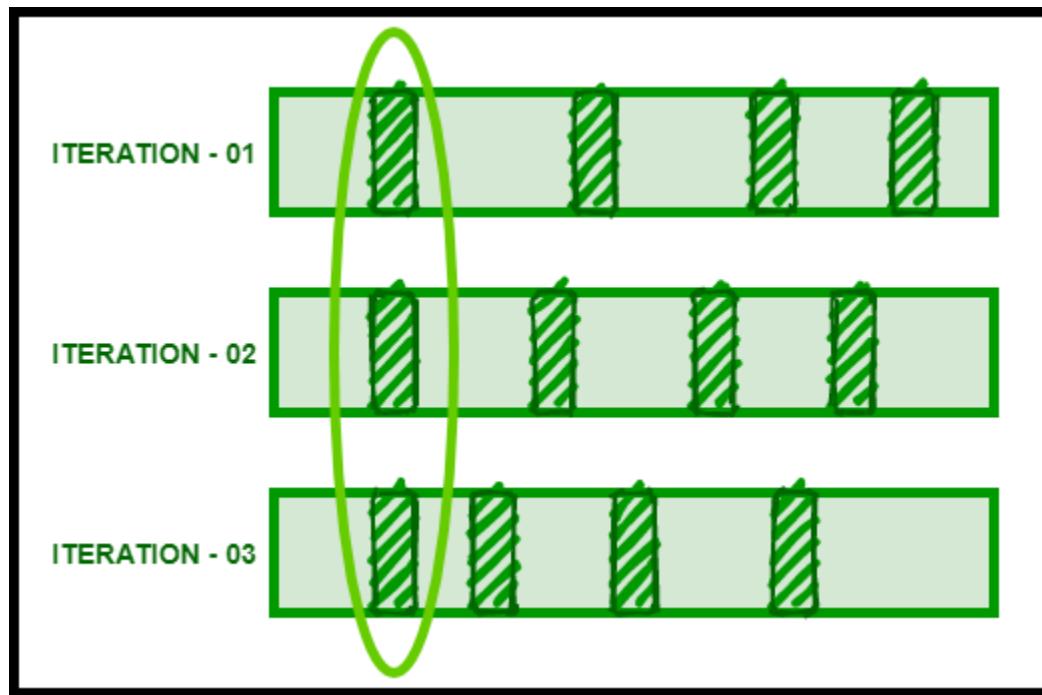
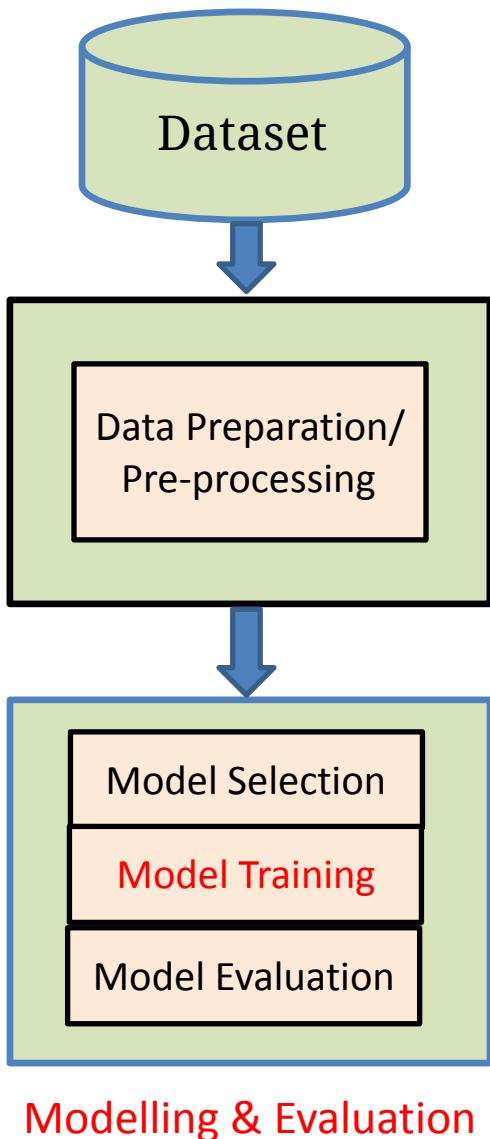
iteration 02= S2

iteration 03= S3

Final Accuracy= $(S1+S2+S3)/3$ (it can be any performance measure)

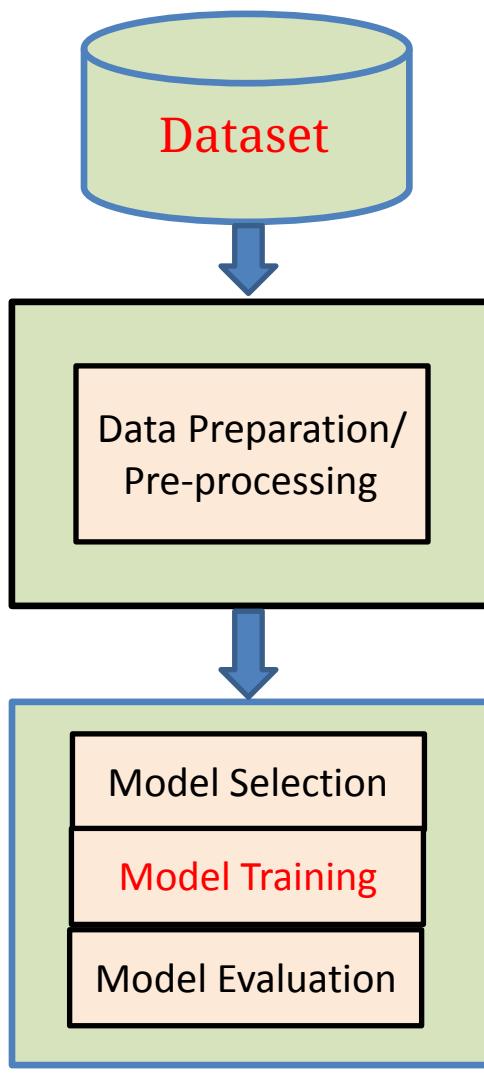
Model Training

ii. Repeated Holdout method: Drawback



Overlapping test set problem

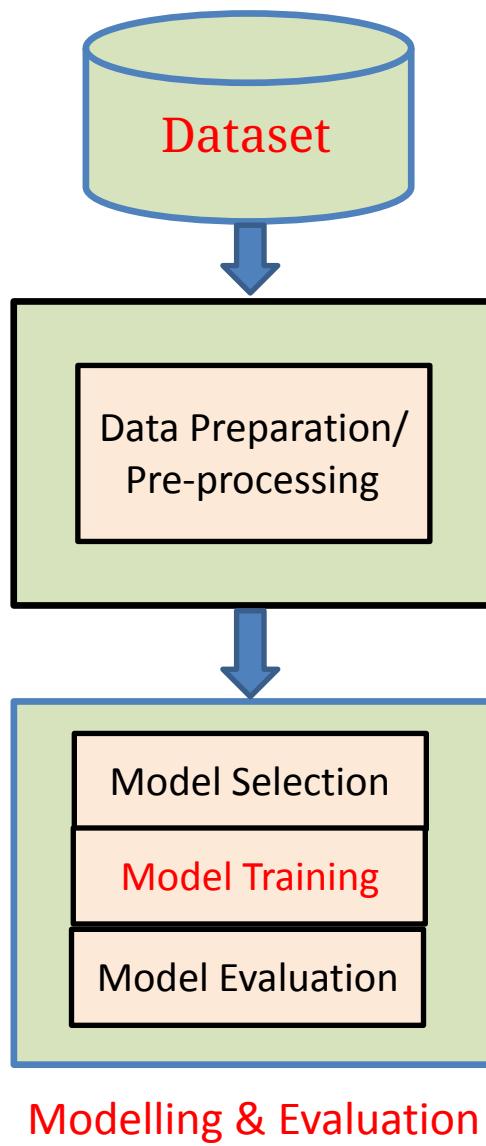
Since we partition the dataset randomly into a training set and test set, **there are some data items/examples that could not be placed in the training set at all**



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P41----P50	
P51----P60	
P61----P70	
P71----P80	
P81----P90	
P91----P100	



iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

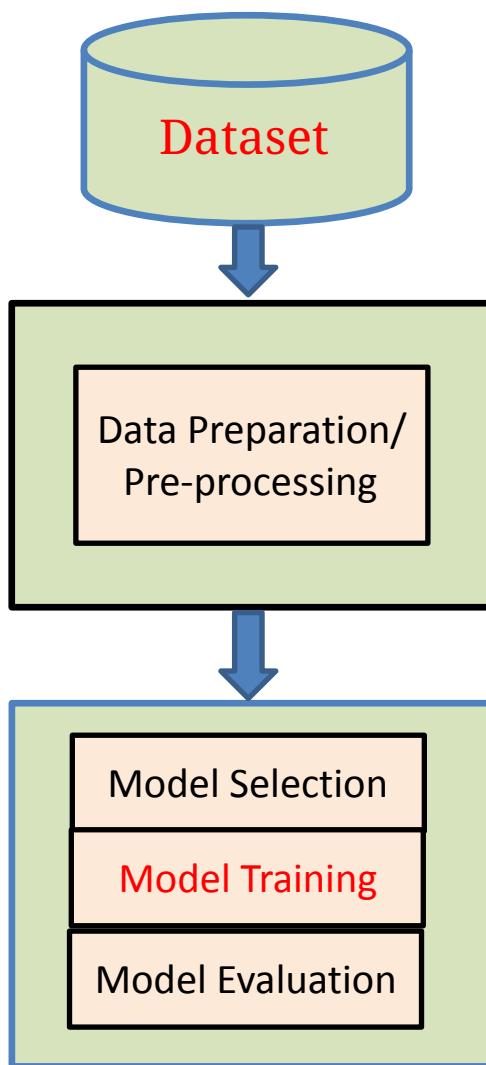
b. Leave-One-Out Cross-Validation

Training Patterns

- P1----P10
- P11----P20
- P21----P30
- P31----P40
- P41----P50
- P51----P60
- P61----P70
- P71----P80
- P81----P90**

Test Patterns

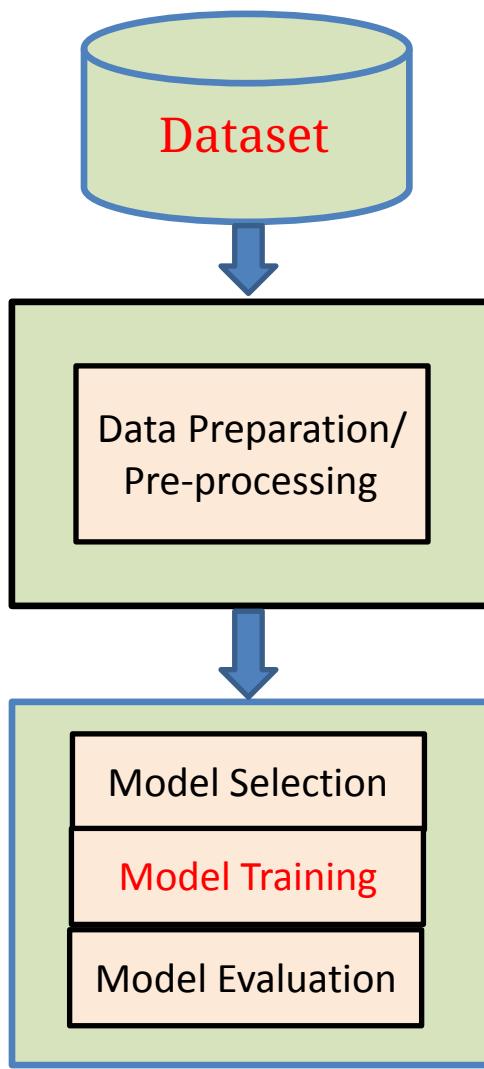
P91----P100



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
 - b. Leave-One-Out Cross-Validation

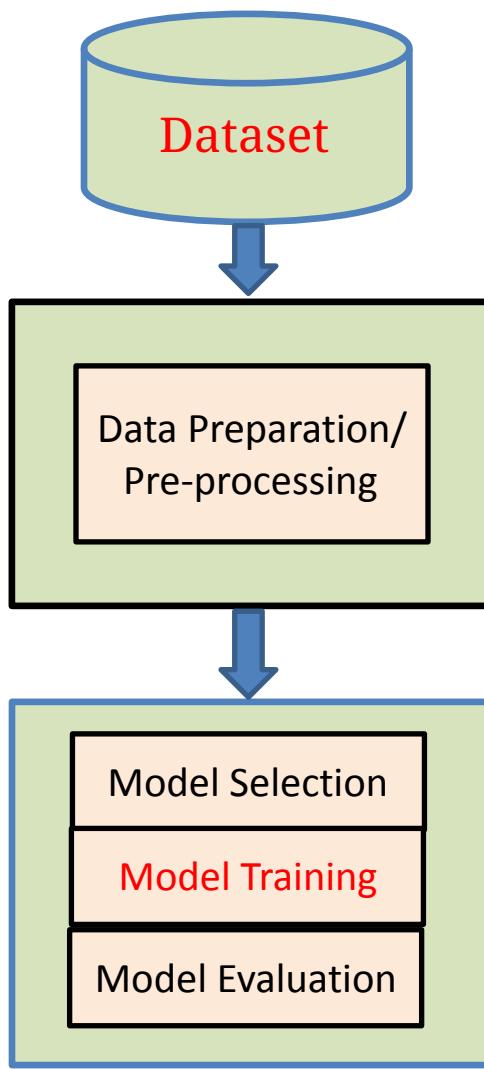
Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P41----P50	
P51----P60	
P61----P70	
P71----P80	
P91----P100	P81----P90



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1---P10	
P11---P20	
P21---P30	
P31---P40	
P41---P50	
P51---P60	
P61---P70	
P81---P90	P71---P80
P91---P100	

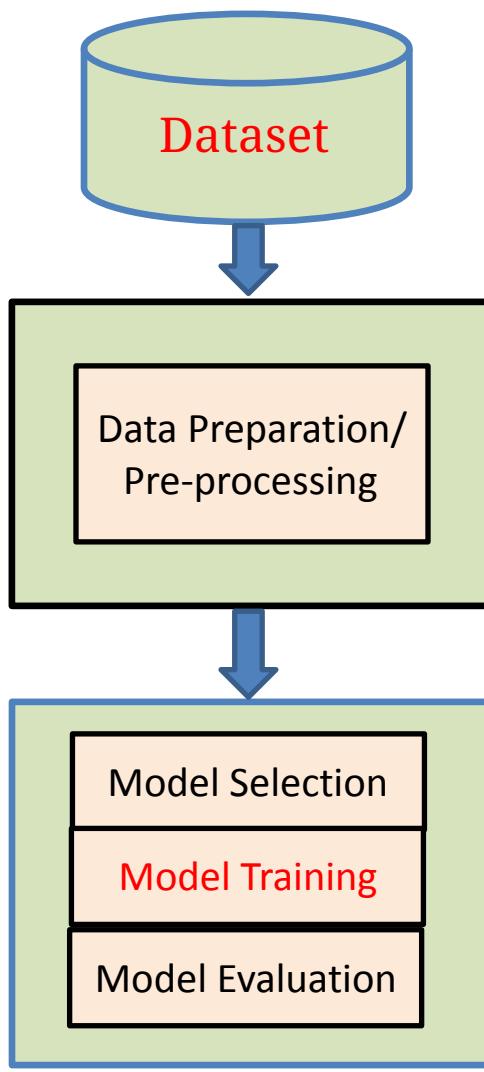


iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P11----P20
P21----P30
P31----P40
P41----P50
P51----P60
P71----P80
P81----P90
P91----P100

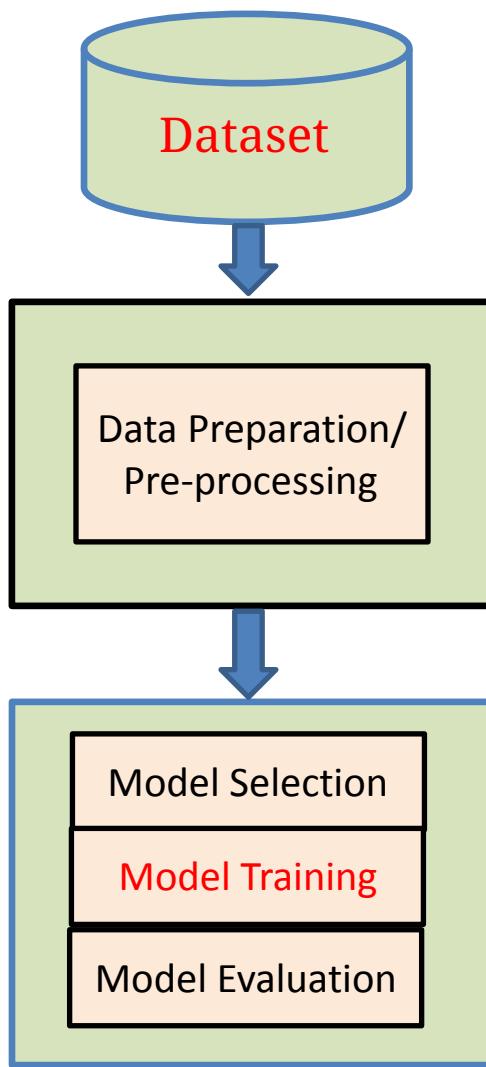
Test Patterns
P61----P70



iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
- b. Leave-One-Out Cross-Validation

Training Patterns	Test Patterns
P1----P10	
P11----P20	
P21----P30	
P31----P40	
P41----P50	
P51----P60	
P61----P70	
P71----P80	
P81----P90	
P91----P100	



iii. K-fold Cross-validation method:

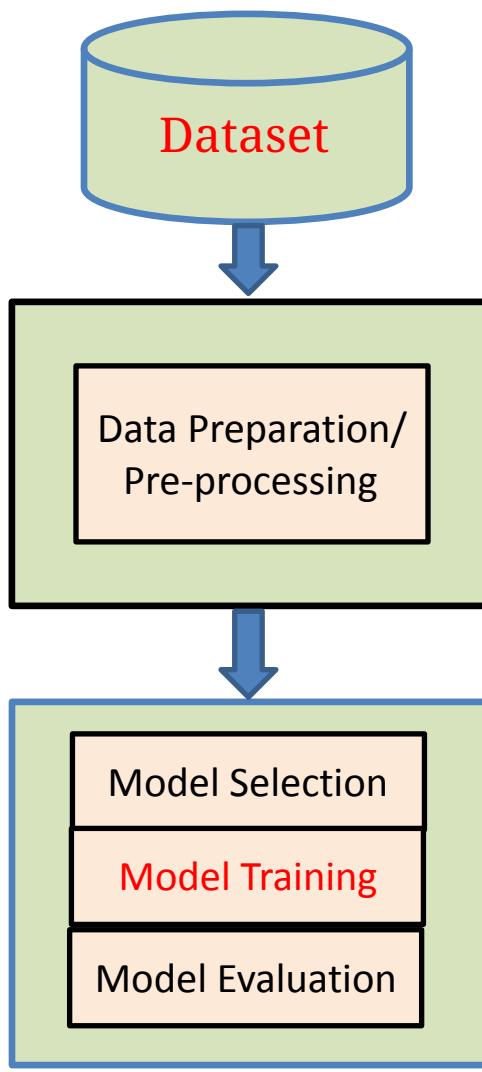
a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P11----P20
P21----P30
P31----P40
P51----P60
P61----P70
P71----P80
P81----P90
P91----P100

Test Patterns
P41----P50

Modelling & Evaluation



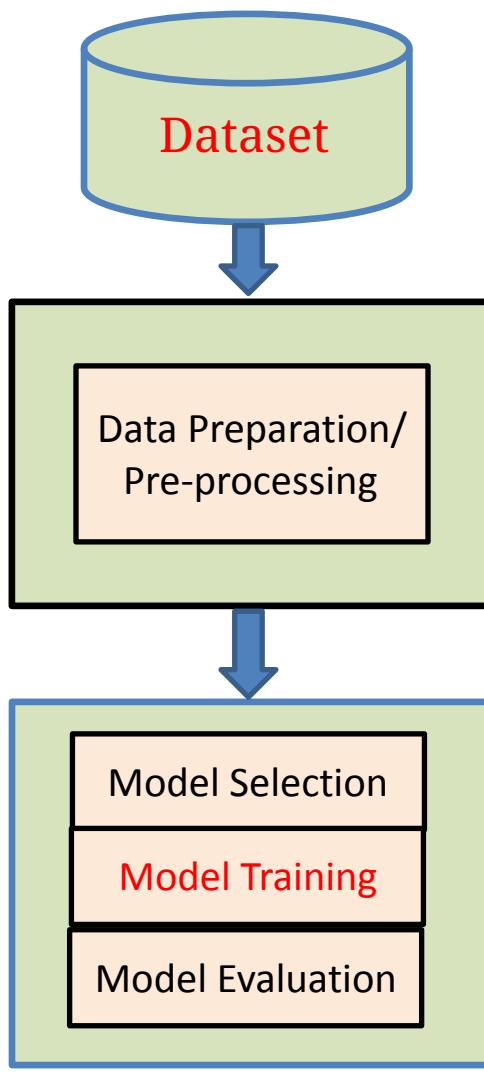
iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P11----P20
P21----P30
P41----P50
P51----P60
P61----P70
P71----P80
P81----P90
P91----P100

Test Patterns
P31----P40



iii. K-fold Cross-validation method:

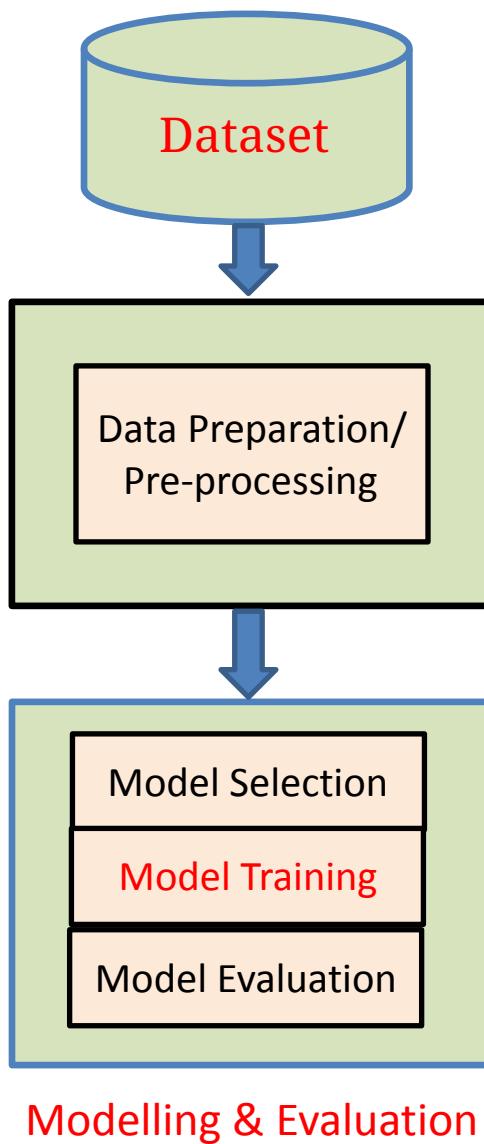
a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1---P10
P11---P20
P31---P40
P41---P50
P51---P60
P61---P70
P71---P80
P81---P90
P91---P100

Test Patterns
P21---P30

Modelling & Evaluation



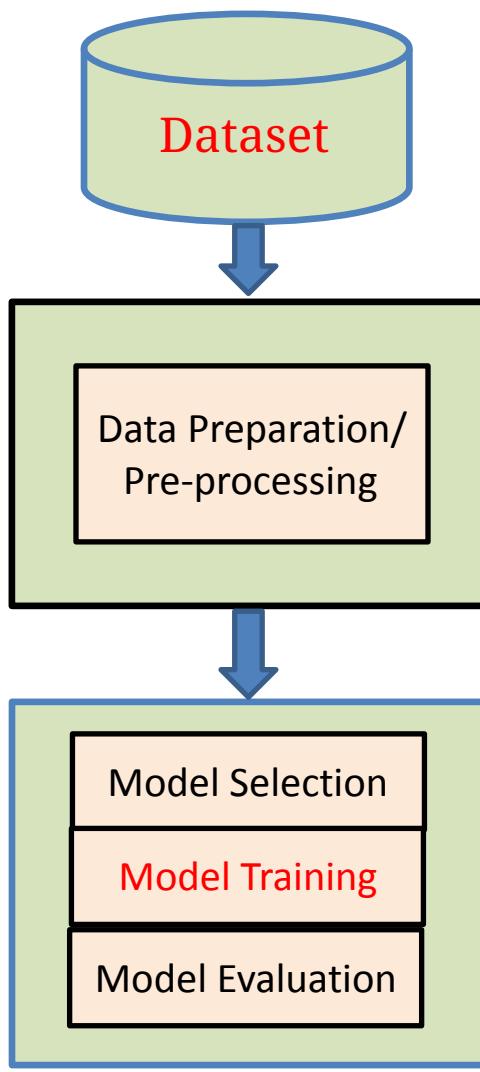
iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

b. Leave-One-Out Cross-Validation

Training Patterns
P1----P10
P21----P30
P31----P40
P41----P50
P51----P60
P61----P70
P71----P80
P81----P90
P91----P100

Test Patterns
P11----P20

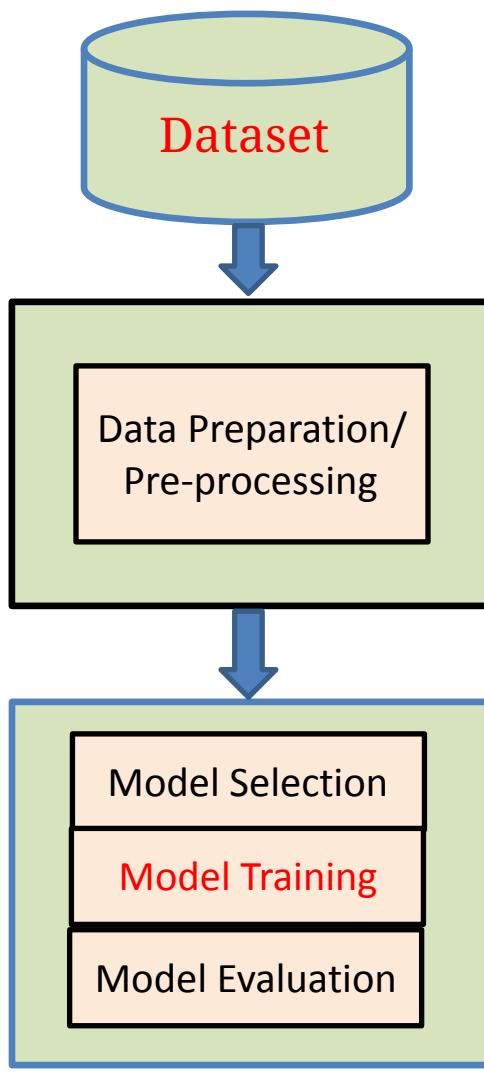


iii. K-fold Cross-validation method:

- a. 10-fold Cross-Validation
 - b. Leave-One-Out Cross-Validation

- P11----P20
- P21----P30
- P31----P40
- P41----P50
- P51----P60
- P61----P70
- P71----P80
- P81----P90
- P91----P100

Test Patterns



iii. K-fold Cross-validation method:

a. 10-fold Cross-Validation

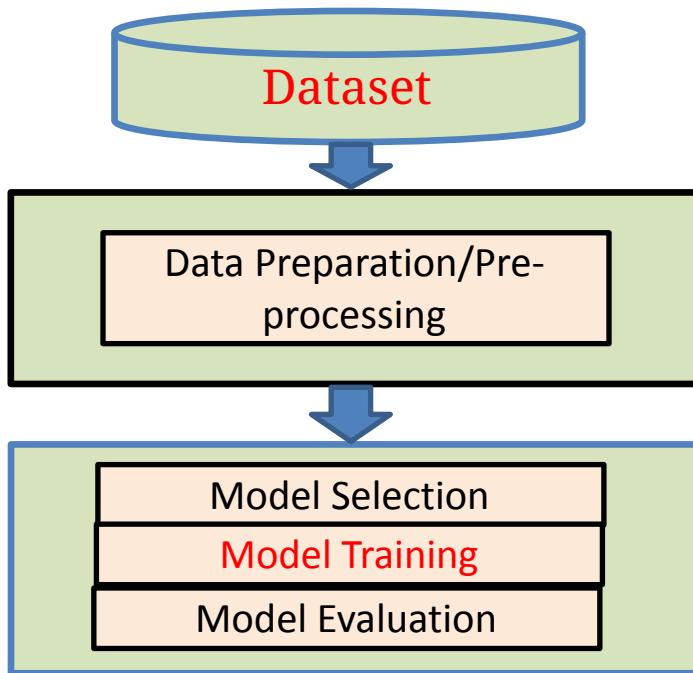
b. Leave-One-Out Cross-Validation

Training Patterns
P1---P10
P11---P20
P21---P30
P31---P40
P41---P50
P51---P60
P61---P70
P71---P80
P81---P90
P91---P100

Test Patterns
P91---P100
P81---P90
P71---P80
P61---P70
P51---P60
P41---P50
P31---P40
P21---P30
P11---P20
P1---P10

Model Training

iii. K-fold Cross-validation method



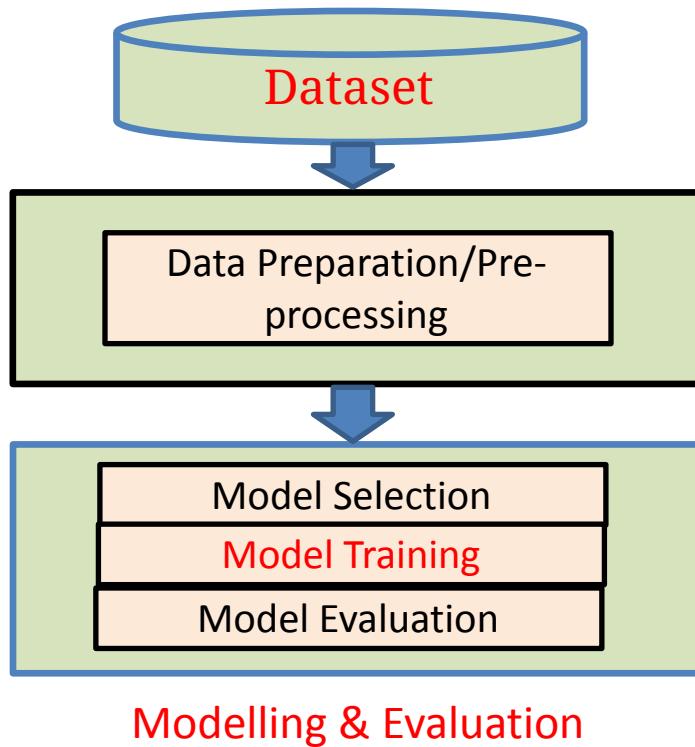
Advantages:

- 1. Reduces Overfitting:** the model attains the generalization capabilities which is a good sign of a robust algorithm.
- 2. Hyperparameter Tuning:** Cross Validation helps in finding the optimal value of hyperparameters.
- 3.** Cross-validation gives us an idea about how the model will perform on an unknown dataset.

Modelling & Evaluation

Split	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

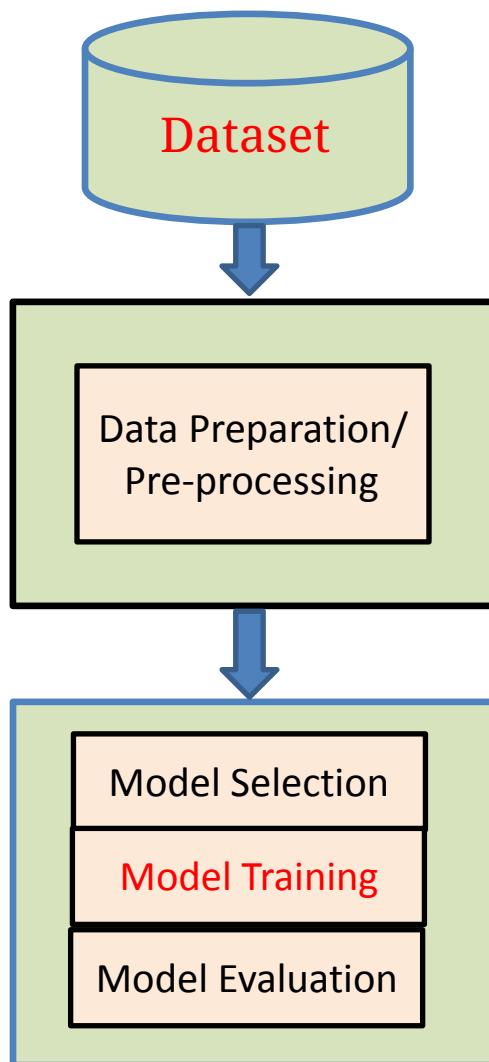
Training data
Test data



iii. K-fold Cross-validation method

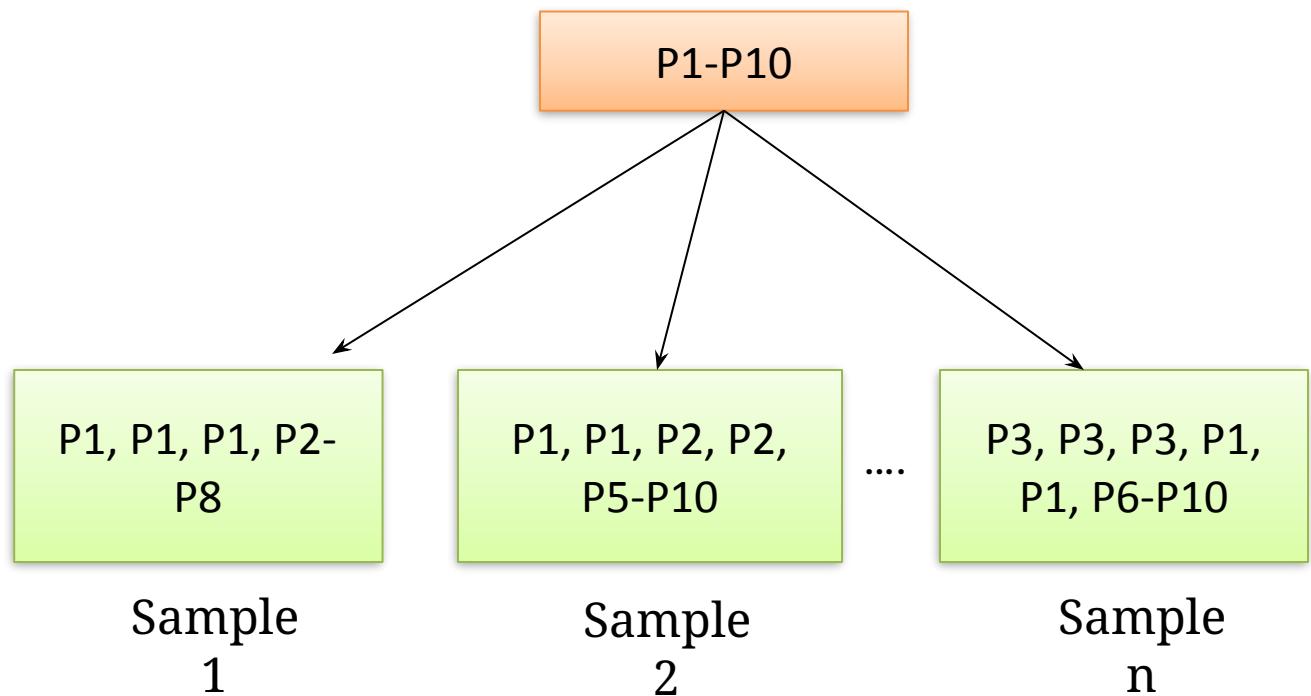
Disadvantages:

1. Time complexity is more.

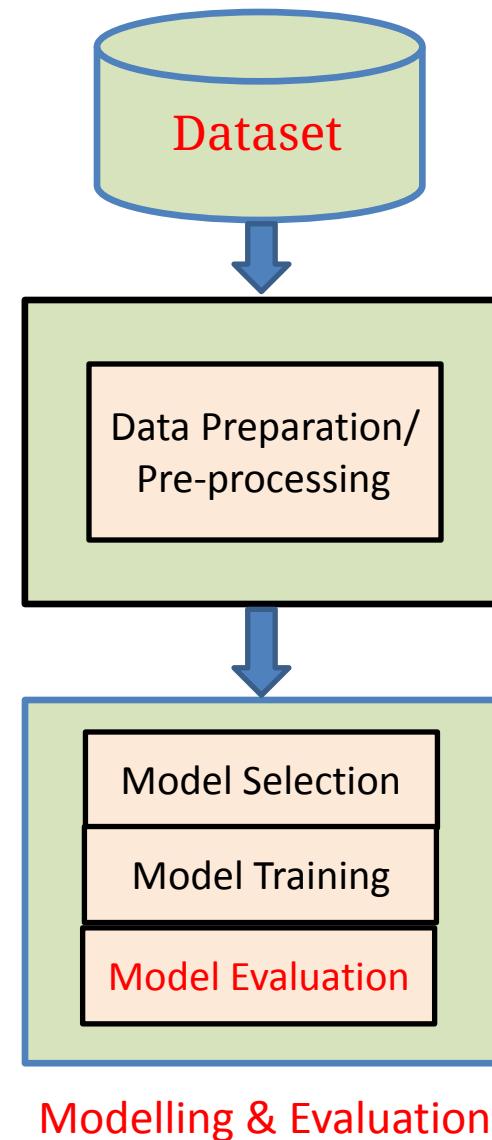


iii. Bootstrap sampling or bootstrapping:

- It uses Simple Random Sampling with Replacement
- It is used for small datasets
- Possible number of training/test data samples is unlimited



Data Science Model



Performance measures for Classification

Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP (3)	FN (1)
	Negative	FP (2)	TN(2)

TP = True Positive; FN= False Negative

FP= False Positive; TN= True Negative

Accuracy	
----------	--

Recall (high)	
Precision (low)	

F-measure	$(2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) = 2 * 0.6 * 0.75 / (0.6 + 0.75) = 0.9 / 1.35 = 67\%$
-----------	---

Model Evaluation

Performance measures for Classification

Matthews Correlation Coefficient (MCC)

Receiver Operating Characteristic (ROC) curves

Statistical Hypothesis Test:

- T test
- Z test
- ANOVA Test
- Chi-Square Test

Model Evaluation

Performance measures for Regression: R-squared

R-squared is a good measure to evaluate the model fitness.

The R-squared value lies between 0 to 1 (0% to 100%).

Large value represents a better fit.

$$R^2 = \frac{SST - SSE}{SST}$$

Sum of Squared Total (SST) = squared differences of each observation from the overall mean = $\sum_{i=1}^n (y_i - \bar{y})^2$ where \bar{y} is the mean.

Sum of Squared Errors (SSE) of prediction= sum of the squared residuals= $\sum_{i=1}^n (Y_i - y^*)^2$ where y^* is the predicted value of y_i and Y_i is the actual values of y_i

Evaluation of Regression

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

$$\begin{aligned}
 M_{Ext} &= 19.13 + 1.89 \times M_{Int} \\
 &= 19.13 + 1.89 \times 15 \\
 &= 19.13 + 28.35 \\
 &= 47.48
 \end{aligned}$$

$$SST = \sum_{i=1}^n (y_i - y')^2$$

$$SSE = \sum_{i=1}^n (y_i - y^*)^2$$

$$R^2 = \frac{SST - SSE}{SST}$$

$$\begin{aligned}
 &= (1148.4 - 328.51) / 1148.4 \\
 &= 819.89 / 1148.4 \\
 &= 0.71
 \end{aligned}$$

71%

		Square d Diff			
49	-7.8	60.84	47.48	1.52	2.31
63	6.2	38.44	62.6	0.4	0.16
58	1.2	1.44	53.15	4.2	17.64
60	3.2	10.24	62.6	-2.6	6.76
58	1.2	1.44	64.49	-6.49	42.12
61	4.2	17.64	60.71	0.29	0.08
60	3.2	10.24	60.71	-0.71	0.50
63	6.2	38.44	55.04	7.96	63.36
60	3.2	10.24	55.04	4.96	24.60
52	-4.8	23.04	49.37	2.63	6.92
62	5.2	27.04	64.49	-2.49	6.2
30	-26.8	718.24	39.92	-9.92	98.41
59	2.2	4.84	64.49	-5.49	30.14
49	-7.8	60.84	49.37	-0.37	0.14
68	11.2	125.44	62.6	5.4	29.16
56.8	SST = 1148.4			SSE = 328.51	

ii. Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

$$s_1 = \frac{l_1 + l_2 + l_3}{3} = x_1$$

Suppose the model has an RMSE value of Rs 500. Since the typical range of salary is between Rs 70,000 and Rs 300,000, this RMSE value is extremely low.

Suppose the model has an RMSE value of Rs 500. If the typical range of monthly house rent is Rs 1,500 – Rs 4,000, this RMSE value is quite high.

$$\text{Normalized RMSE} = \text{RMSE} / (\text{max value} - \text{min value}) = 4.679/(68-30) = 0.123$$

This produces a value between 0 and 1.

49	47.48	1.52	2.31
63	62.6	0.4	0.16
58	53.15	4.2	17.64
60	62.6	-2.6	6.76
58	64.49	-6.49	42.12
61	60.71	0.29	0.08
60	60.71	-0.71	0.50
63	55.04	7.96	63.36
60	55.04	4.96	24.60
52	49.37	2.63	6.92
62	64.49	-2.49	6.2
30	39.92	-9.92	98.41
59	64.49	-5.49	30.14
49	49.37	-0.37	0.14
68	62.6	5.4	29.16
56.8			SSE = 328.51
			MSE = 328.51/15 = 21.90

Model Evaluation

Performance measures for Regression:

iii. Mean Absolute Error (MAE)

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Normalized MAE = MAE / (max value – min value)= $3.69/(9.92-0.29)= 0.38$

Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the **RMSE should be more useful when large errors are particularly undesirable**. Both ranges from 0 to infinity.

49	47.48	1.52
63	62.6	0.4
58	53.15	4.2
60	62.6	2.6
58	64.49	6.49
61	60.71	0.29
60	60.71	0.71
63	55.04	7.96
60	55.04	4.96
52	49.37	2.63
62	64.49	2.49
30	39.92	9.92
59	64.49	5.49
49	49.37	0.37
68	62.6	5.4
56.8		AE = 55.43 MAE= 55.43/15=3.69

Model Evaluation

Performance measures for

Clustering

It is generally not known how many clusters can be formulated from a particular dataset.

Even if the number of clusters is given, the same number of clusters can be formed with different groups of data instances.

Internal Evaluation

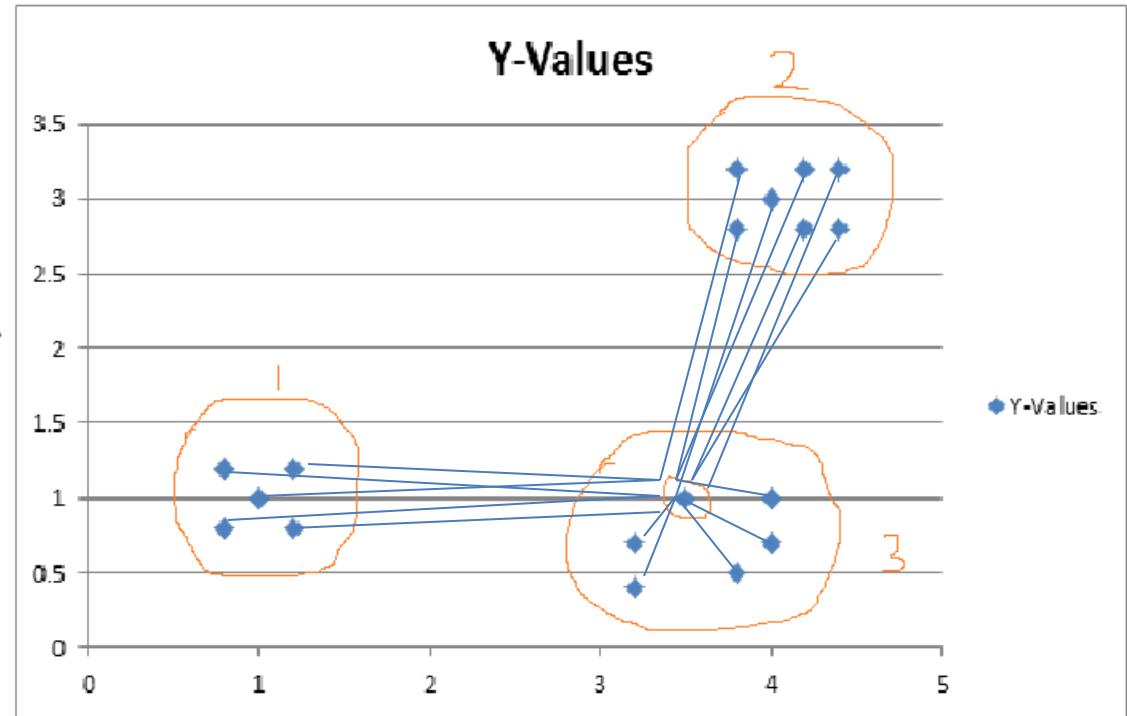
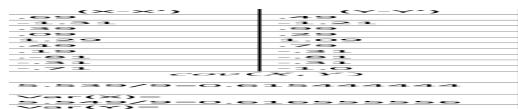
$$\text{Silhouette width} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

It ranges from -1 to +1

a(i) is the average distance between the ith data instance and all other data instances belonging to the same cluster

b(i) is the lowest average distance between the ith data instance and data instances of all other clusters.

a(i) is the average of the distances $a_{i1}, a_{i2} \dots \dots a_{in3}$ of the different data elements from the ith data elements in cluster 3, n3= data elements of cluster 3.



Average distance from ith elements of cluster 3 to cluster 1

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

Similarly b_{32} can be calculated

$$b(i) = \min [b_{32}(\text{average}), b_{31}(\text{average})]$$

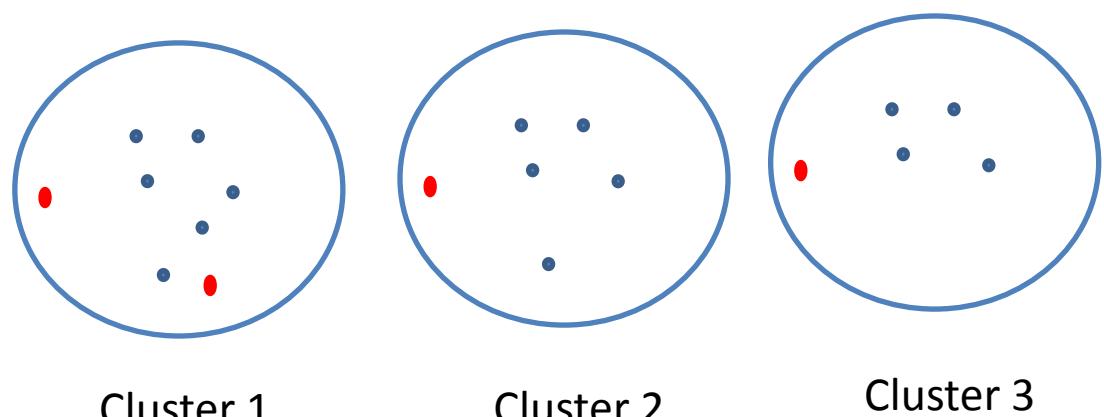
Model Evaluation

External Evaluation:

Purity: This is only applicable for class labels data though class labels are not used for clustering

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

$\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ = set of clusters $\mathbb{C} = \{c_1, c_2, \dots, c_J\}$ = set of classes N is total data instances



Supervised ML Algorithms

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- K- Nearest Neighbour (K-NN)
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

REGRESSION

Regression is a supervised learning which predicts a continuous value

- Predicting the price of a car
- Predicting the amount of rainfall
- Predicting the cost of a land

The most common regression algorithms are

- Simple linear regression
- Multiple linear regression
- Ridge Regression
- LASSO Regression
- Elastic Net Regression
- Polynomial regression
- Multivariate adaptive regression splines
- Logistic regression

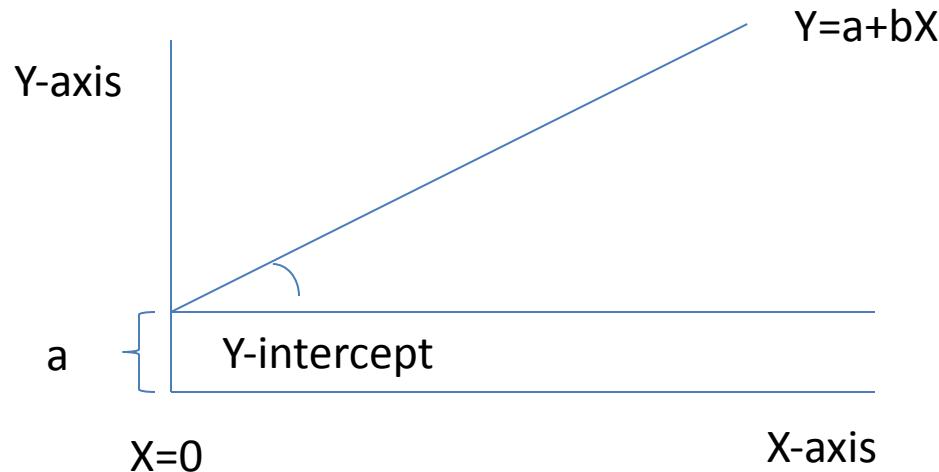
Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Simple Linear Regression

Simple Linear Regression is the simplest regression model which involves only one predictor.

This model assumes a linear relationship between the dependent variable and the predictor variable

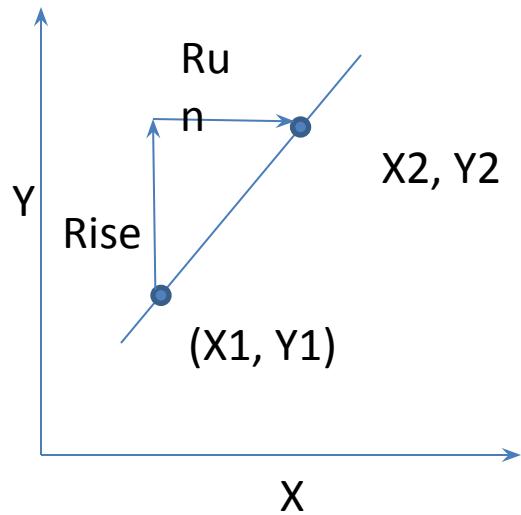
$Y = a + bX$ (price of a property as the dependent variable and the area of the property as the predictor variable)



Slope of the simple Linear Regression model

Slope of a straight line represents how much the line in a graph changes in the vertical direction over a change in the horizontal direction.

$$\text{Slope} = \text{Change in Y}/\text{Change in X}$$



$$\text{Slope} = \text{Rise}/\text{Run} = \frac{Y_2 - Y_1}{X_2 - X_1}$$

Let lower point = (-3, -2); higher point = (2, 2)

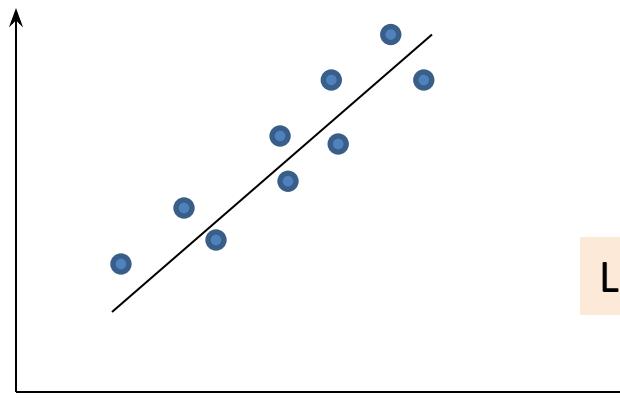
$$\text{Rise} = (2 - (-2)) = 2 + 2 = 4$$

$$\text{Run} = (2 - (-3)) = 2 + 3 = 5$$

$$\text{Slope} = \text{Rise}/\text{Run} = 4/5 = 0.8$$

Slopes in a Linear Regression

- There are two types of slopes: positive slope and negative slope
- Different types of regression lines based on the type of slope are
 - Linear positive slope
 - Curve linear positive slope
 - Linear negative slope
 - Curve linear negative slope



Linear positive slope

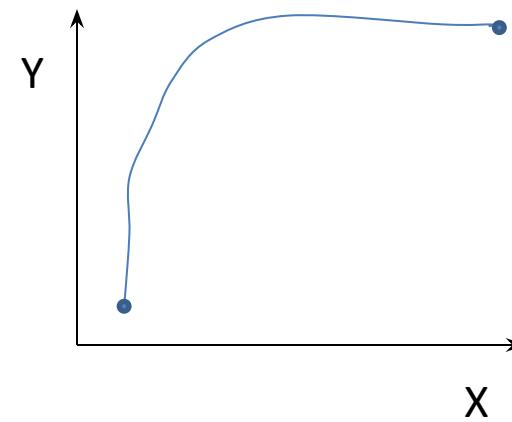
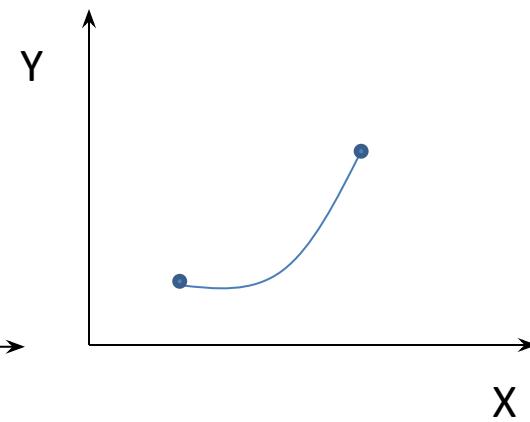
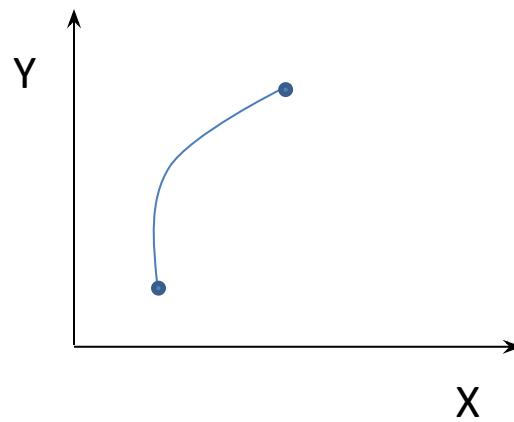
$$\text{Slope} = \text{Rise/Run} = \frac{\Delta(Y)}{\Delta(X)}$$

Scenario 1 for positive slope: $\Delta(Y)$ is positive and $\Delta(X)$ is positive

Scenario 2 for positive slope: $\Delta(Y)$ is negative and $\Delta(X)$ is negative

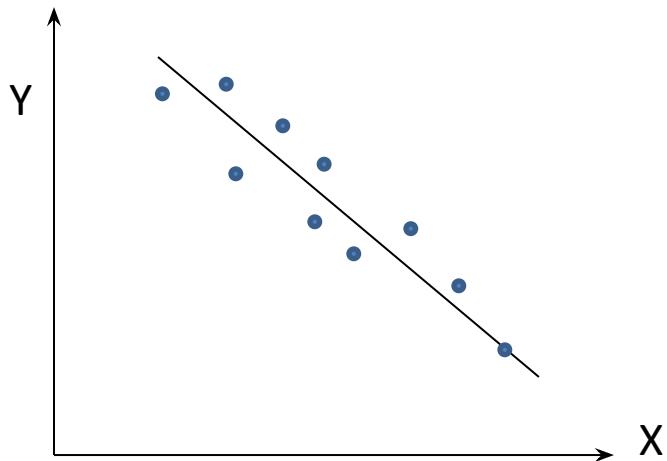
Slopes in a Linear Regression

Curve Linear Positive Slope



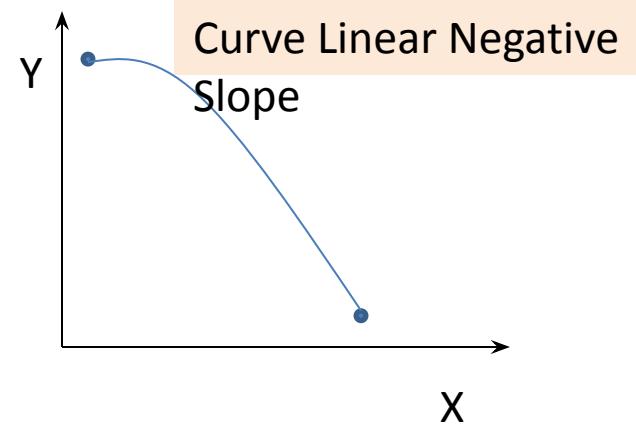
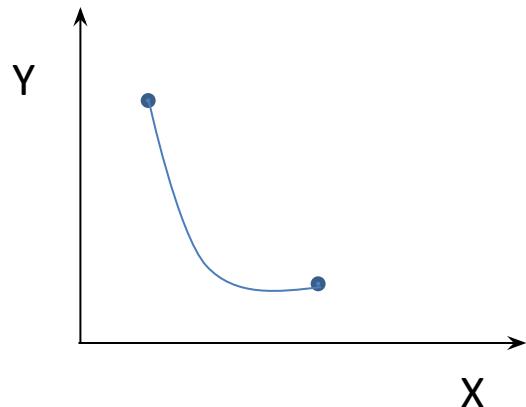
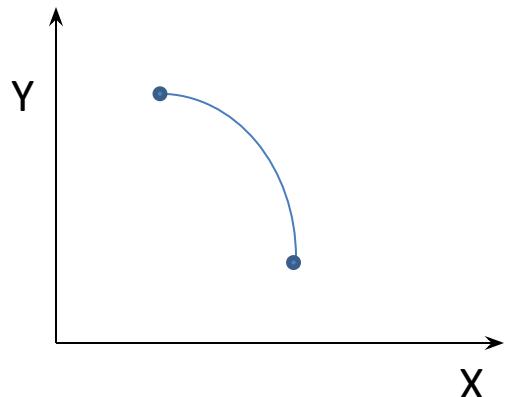
Slopes in a Linear Regression

Linear Negative Slope

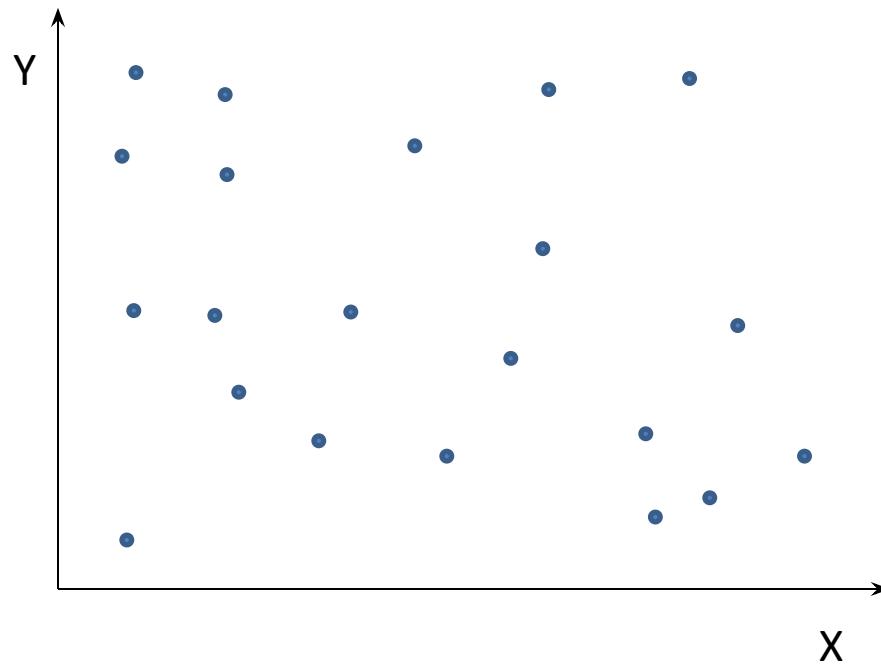


Scenario 1 for negative slope: Delta(Y) is positive and Delta(X) is negative

Scenario 2 for negative slope: Delta(Y) is negative and Delta(X) is positive



No Relationship Graph



Error in Simple Regression

X and Y values are provided to the machine to find the values of a and b by relating the values of X and Y.

However identifying the exact match of values for a and b is not always possible. There will be some error (ε). This error is called marginal or residual error.

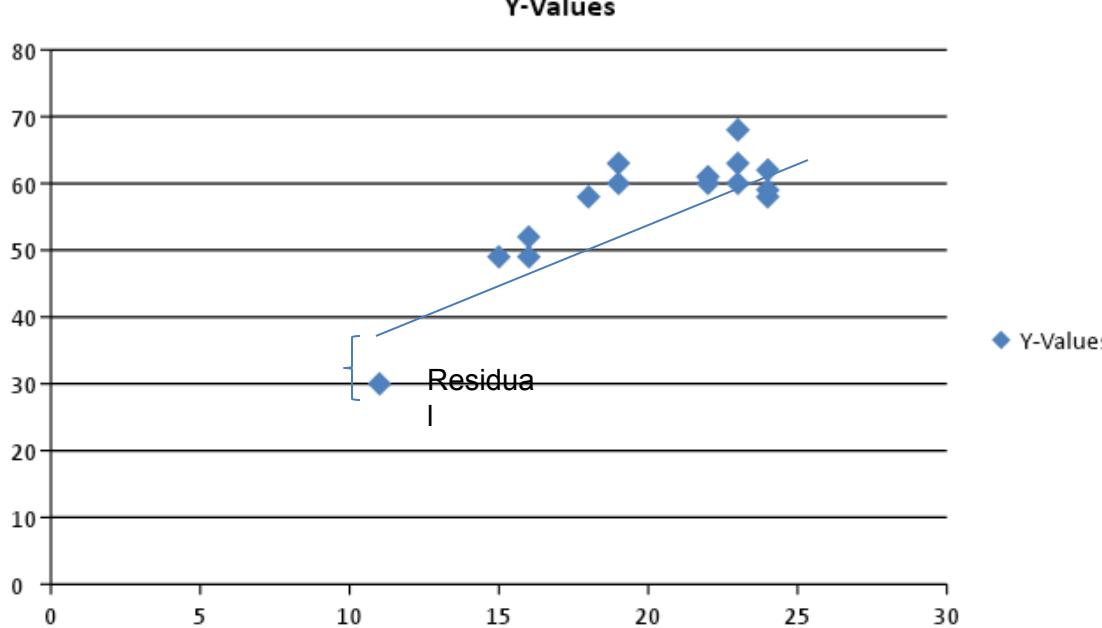
$$Y = (a + bX) + \varepsilon$$

Linear Regression Technique with example

Finding the relationship between internal examination and external examination from the given data samples

Internal Exam	15	23	18	23	24	22	22	19	19	16	2 4	11	24	16	23
External Exam	49	63	58	60	58	61	60	63	60	52	6 2	30	59	49	68

Linear Regression Technique with example



Residual is the distance between the predicted point and actual point.

$$Y = (a + bX) + \epsilon$$

If values of a and b are known, it is easy to predict the value of Y for any given X .
How to calculate the values of a and b for a given set of X and Y values with minimum error.

Linear Regression Technique with example

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

OLS Algorithm

Step1: Calculate the mean of X and Y.

Step2: Calculate the errors for each values of X and Y.

Step3: Get the product for each corresponding values.

Step4: Get the summation of the products.

Step5: Square the difference of X and mean(X).

.

Step6: Get the sum of the squared differences.

X	Y	X-mean(X)	Y-mean(Y)		
15	49	-4.93	-7.8	38.454	24.3049
23	63	3.07	6.2	19.034	9.4249
18	58	-1.93	1.2	-2.316	3.7249
23	60	3.07	3.2	9.824	9.4249
24	58	4.07	1.2	4.884	16.5649
22	61	2.07	4.2	8.694	4.2849
22	60	2.07	3.2	6.624	4.2849
19	63	-0.93	6.2	-5.766	0.8649
19	60	-0.93	3.2	-2.976	0.8649
16	52	-3.93	-4.8	18.864	15.4449
24	62	4.07	5.2	21.164	16.5649
11	30	-8.93	-26.8	239.324	79.7449
24	59	4.07	2.2	8.954	16.5649
16	49	-3.93	-7.8	30.654	15.449
23	68	3.07	11.2	34.384	9.4249
19.9	56.			429.8	226.9335
3	8				

Linear Regression Technique with example

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

OLS Algorithm

Step7: Divide output of step4 by output of step 6 to calculate b.

$$b = 429.28/226.93 = 1.89$$

Step8: Calculate 'a' using the value of b.

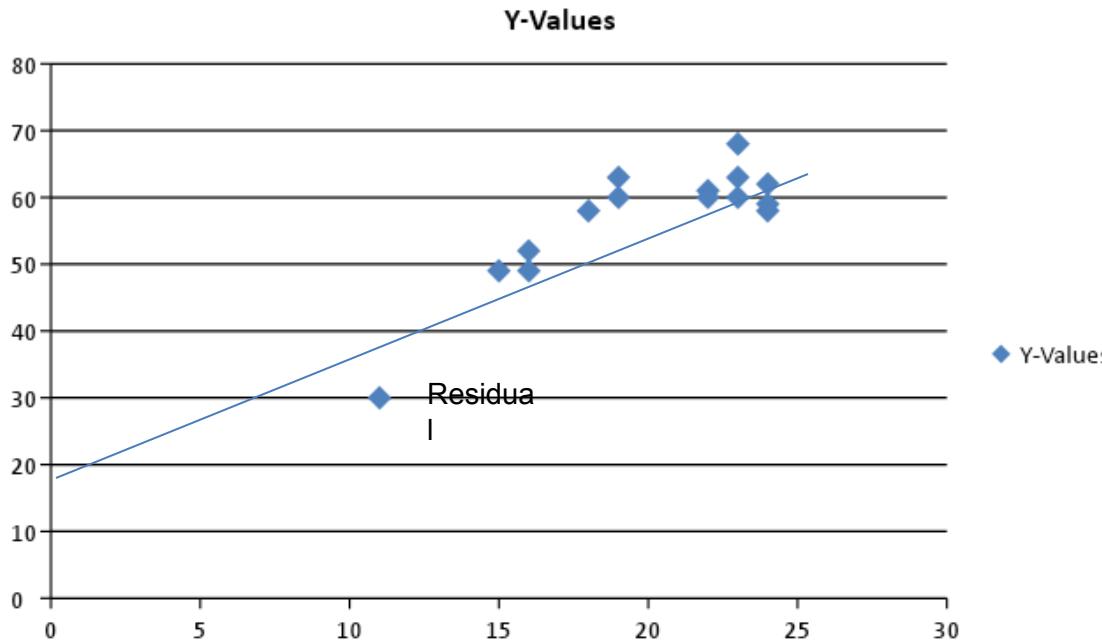
$$a = Y' - bX'$$

$$a = 56.8 - 1.89 * 19.93$$

$$a = 19.13$$

X	Y	X-mean(X)	Y-mean(Y)		
15	49	4.03	-7.8	38.454	24.3049
23	63	3.07	6.2	19.034	9.4249
18	58	-1.93	1.2	-2.316	3.7249
23	60	3.07	3.2	9.824	9.4249
24	58	4.07	1.2	4.884	16.5649
22	61	2.07	4.2	8.694	4.2849
22	60	2.07	3.2	6.624	4.2849
19	63	-0.93	6.2	-5.766	0.8649
19	60	-0.93	3.2	-2.976	0.8649
16	52	-3.93	-4.8	18.864	15.4449
24	62	4.07	5.2	21.164	16.5649
11	30	-8.93	-26.8	239.324	79.7449
24	59	4.07	2.2	8.954	16.5649
16	49	-3.93	-7.8	30.654	15.449
23	68	3.07	11.2	34.384	9.4249
19.9 3	56. 8			429.28	226.9335

Intercept and Slope



Recall (high)	$\frac{TP}{TP + FN} = \frac{3}{3+1} = \frac{3}{4} = 0.75$
Precision (low)	$\frac{TP}{TP + FP} = \frac{3}{3+2} = \frac{3}{5} = 0.6$

- Intercept 19.13 indicates that 19.13 is the portion of the external examination marks not explained by the internal examination marks.
- Slope = 1.89 tells us that the average value of the external examination marks increases by 1.89 for each additional 1 mark in the internal examination.

Evaluation of Regression

Performance measures for Regression: R-squared

R-squared is a good measure to evaluate the model fitness.

The R-squared value lies between 0 to 1 (0% to 100%).

Large value represents a better fit.

$$R^2 = \frac{SST - SSE}{SST}$$

Sum of Squared Total (SST) = squared differences of each observation from the overall mean = $\sum_{i=1}^n (y_i - \bar{y})^2$ where \bar{y} is the mean.

Sum of Squared Errors (SSE) of prediction= sum of the squared residuals= $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ where \hat{y}_i is the predicted value of y_i .

Linear Regression Technique with example

Different kind of algorithms are there. One of them is Ordinary Least Square (OLS)

$$\begin{aligned}
 M_{Ext} &= 19.13 + 1.89 \times M_{Int} \\
 &= 19.13 + 1.89 \times 15 \\
 &= 19.13 + 28.35 \\
 &= 47.48
 \end{aligned}$$

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$R^2 = \frac{SST - SSE}{SST}$$

$$\begin{aligned}
 &= (1148.4 - 328.51) / 1148.4 \\
 &= 819.89 / 1148.4 \\
 &= 0.71
 \end{aligned}$$

71%

		Square d Diff			
49	-7.8	60.84	47.48	1.52	2.31
63	6.2	38.44	62.6	0.4	0.16
58	1.2	1.44	53.15	4.2	17.64
60	3.2	10.24	62.6	-2.6	6.76
58	1.2	1.44	64.49	-6.49	42.12
61	4.2	17.64	60.71	0.29	0.08
60	3.2	10.24	60.71	-0.71	0.50
63	6.2	38.44	55.04	7.96	63.36
60	3.2	10.24	55.04	4.96	24.60
52	-4.8	23.04	49.37	2.63	6.92
62	5.2	27.04	64.49	-2.49	6.2
30	-26.8	718.24	39.92	-9.92	98.41
59	2.2	4.84	64.49	-5.49	30.14
49	-7.8	60.84	49.37	-0.37	0.14
68	11.2	125.44	62.6	5.4	29.16
56.8	SST= 1148.4			SSE = 328.51	

Multiple Linear Regression

Two or more independent variables are involved in this model.

$$\text{Price} = f(\text{Area}, \text{location}, \text{floor}, \text{ageing}, \text{amenities})$$

To determine price of the property; area, location, floor, number of years since purchase and amenities are considered.

The following equation describes the relation with 2 independent variables

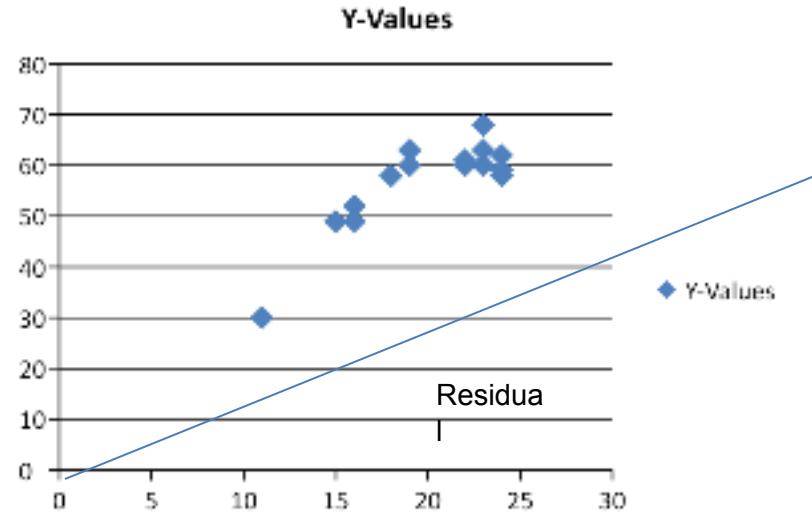
$$Y = a + b_1 X_1 + b_2 X_2$$

Parameters b_1 and b_2 are referred as partial regression coefficient.

Assumption in Linear Regression Analysis

- The dependent variable can be calculated as a linear function of a set of independent variables plus error term.
- Number of observation is greater than the number of parameters
- Regression line can be valid only over a limited range of data. If the line is extended outside the range of extrapolation, it may only lead to wrong predictions.
- Variance is the same for all values of X.
- The error term is normally distributed. This also means that the mean of the error has an expected value of 0.

$$Y = a + b_1X_1 + b_2X_2$$



Primary problems in Multiple Regression

- **Multicollinearity:** A multiple regression equation can make good predictions when there is multicollinearity, However,
 - it is difficult for us to determine how the dependent variable will change if each independent variable is changed one at a time.
 - When multicollinearity is present, it increases the standard errors of the coefficients.
 - By overinflating the standard errors, multicollinearity tries to make some variables statistically insignificant when they actually should be significant.
- **Heteroskedasticity:** Heteroskedasticity refers to the changing variance of the error term . If the variance of the error term is not constant across datasets, there will be erroneous predictions.

Improving accuracy of Linear Regression Model

High bias= low accuracy

High variance= low prediction

Low bias= high accuracy

Low variance= high prediction

Therefore balancing out bias and accuracy is essential in a regression model.

In the regression model, it is assumed that the number of observation is greater than the number of parameters to be estimated.

However if observation is not much larger than parameters, then there can be high variability in the least fit, resulting in overfitting and leading to poor prediction.

Accuracy of linear regression can be improved using the following three approaches:

1. Shrinkage Approach
2. Subset Selection
3. Dimensionality Reduction

Shrinkage Approach

By limiting (shrinking) the estimated coefficients, variance can be reduced at the cost of a negligible increase in bias. This leads substantial improvements in the accuracy of the model.

The two best-known techniques for shrinking the regression coefficients towards zero are

1. Ridge regression
2. LASSO (Least Absolute Shrinkage Selection Operator)

If $k > n$, then the least squares estimates do not ever have a unique solution, whereas ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance. Thus ridge regression works best in situation where the least squares estimates have high variance.

However ridge includes all k predictors in the final model. This may not be a problem for prediction accuracy but it can create a challenge in model interpretation in setting in which the number of variables k is quite large.

$$f(x) = c_0 + c_1 X^1 + c_2 X^2 + c_3 X^3$$

Keeps all the attributes.

Shrinkage Approach

LASSO overcomes this disadvantage by forcing some of the coefficients to zero value.
It's a simple and more interpretable model.

LASSO works better when small number of predictors have substantial coefficients,
and the remaining predictors have coefficients that are very small or equal to zero.

$$f(x) = c_0 + \textcolor{red}{c}_1 X^1 + c_2 X^2 + c_3 X^3$$

Next,

$$f(x) = c_0 + 0X^1 + c_2 X^2 + c_3 X^3$$

$$f(x) = c_0 + c_2 X^2 + c_3 X^3$$

Subset Selection

A subset of predictors that is assumed to be related to the response is selected to fit a model. There are different kinds of methods for subset selection, some of which are given below:

1. Exhaustive search
2. Branch and Bound Search,
3. Selection of Best Individual Features
4. Sequential Forward Selection
5. Sequential Backward Selection
6. Sequential Floating Forward Selection
7. Sequential Floating Backward Selection

Dimensionality Reduction

- Predictors are transformed and the model is set up using the transformed variables after dimensionality reduction.
- The number of variables is reduced using the dimensionality reduction method.
- Principal component analysis is one of the most important dimensionality reduction techniques.

Ridge Regression

$$P1 = (1, 2.3)$$

$$P2 = (3, 5.3)$$

$$Y = (a + bX)$$

$$b = (5.3 - 2.3) / (3 - 1) = 3/2 = 1.5$$

$$Y = (a + bX)$$

$$a = Y - bX$$

$$= 2.3 - 1.5 * 1 = 2.3 - 1.5 = 0.8$$

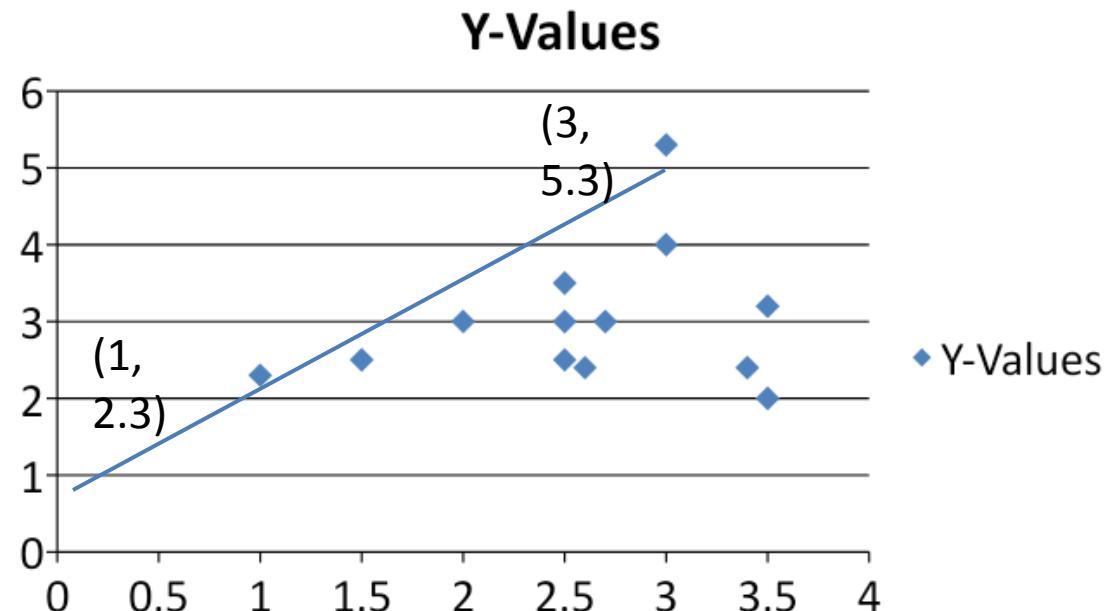
$$y = 0.8 + 1.5x$$

There is no bias (bias=0) as :

$$x=1: y = 0.8 + 1.5 * 1 = 0.8 + 1.5 = 2.3$$

$$x=3: y = 0.8 + 1.5 * 3 = 0.8 + 4.5 = 5.3$$

x	1	3
y	2.3	5.3



$$SSE = \sum_{i=1}^n (Y_i - y^*)^2 = (2.3 - 2.3)^2 + (2.3 - 2.3)^2 = 0 + 0 = 0$$

Means **overfitting** problem is there

How can we understand overfitting and underfitting from the line

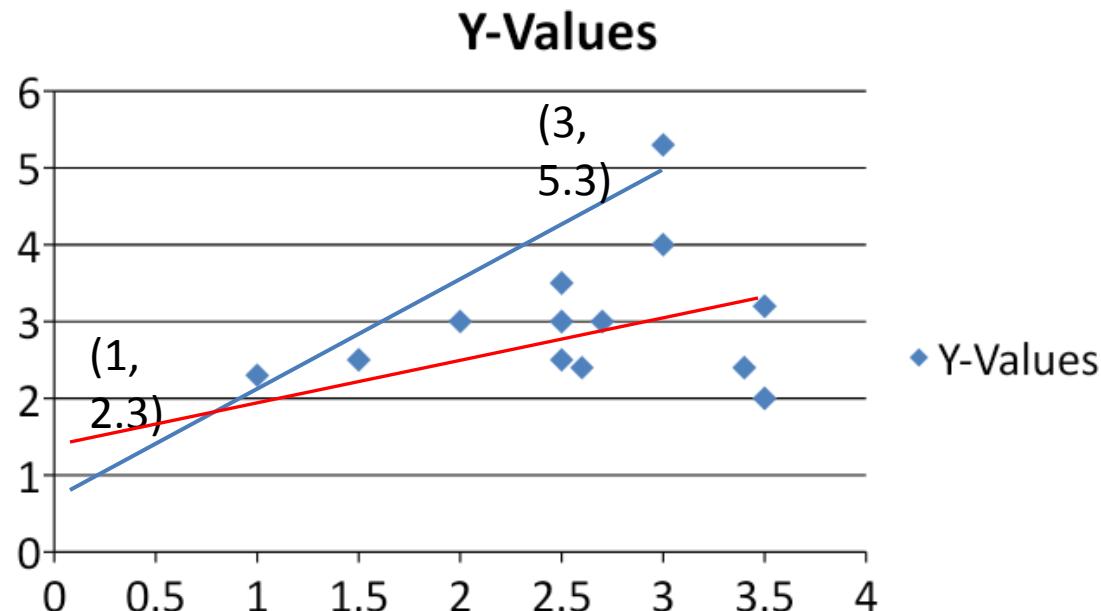
Ridge Regression

$y = 0.8 + 1.5x$ variance is high

$y = 1.5 + 0.9x$ (imaginary line to reduce variance)

Our normal objective is to reduce

$SSE = \sum_{i=1}^n (Y_i - y^*)^2$ (but would not be working as it increases bias for the imaginary line)



To decrease the variance of the imaginary line, the objective function is redefined.

Therefore, now objective is to reduce $SSE = \sum_{i=1}^n (Y_i - y^*)^2 + \lambda(b)^2 = \text{LOSS}$

Ridge Regression

$y = 0.8 + 1.5x$ variance is high

$y = 1.5 + 0.9x$ (imaginary line to reduce variance)

Therefore, now objective is

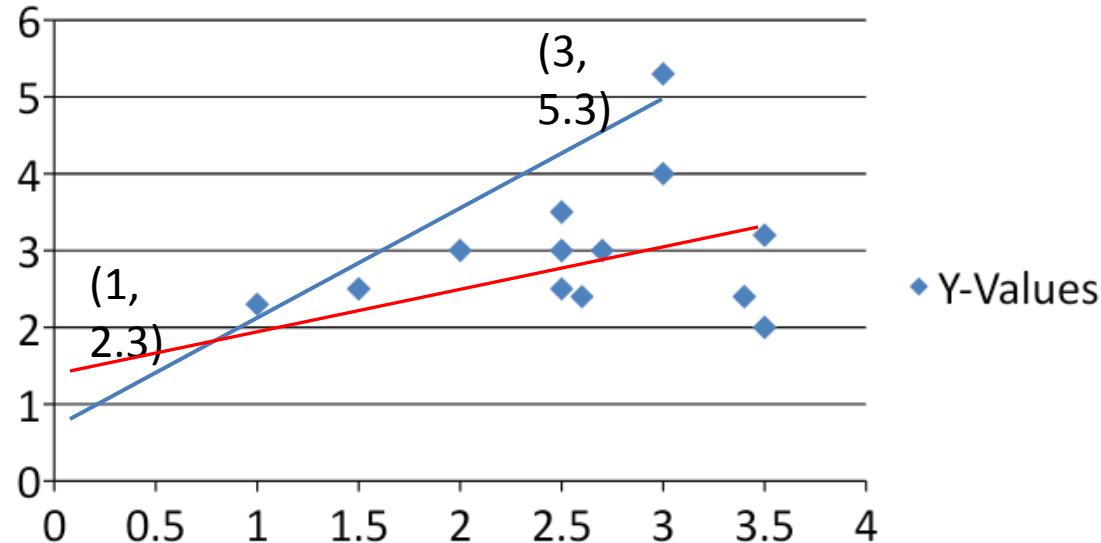
to reduce SSE =

$$\sum_{i=1}^n (Y_i - y^*)^2 + \lambda(\mathbf{b})^2 = \text{LOSS}$$

$$x=1: y = 0.8 + 1.5 * 1 = 0.8 + 1.5 = 2.3$$

$$x=3: y = 0.8 + 1.5 * 3 = 0.8 + 4.5 = 5.3$$

Y-Values



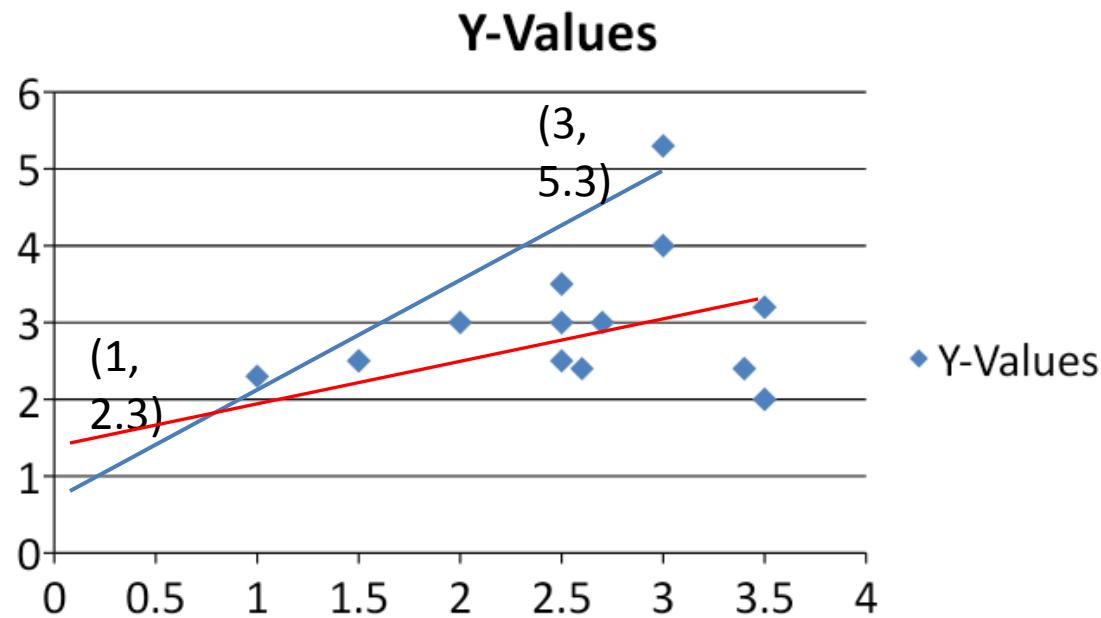
LOSS

LOSS

Ridge Regression

$$-\frac{1}{6}[600 + 620 + 60] = 606$$

$$= \frac{1}{3} [\max(3^{\text{th}} \text{ dist } B, \text{dist}(AB)) + \max(3^{\text{th}} \text{ dist } C, \text{dist}(AC)) + \max(3^{\text{th}} \text{ dist } D, \text{dist}(AD))]$$



Polynomial Regression Model

It is the extension of the simple linear model by adding extra predictors obtained by raising each of the original predictors to a power.

$$f(x) = c_0 + c_1X^1 + c_2X^2 + c_3X^3$$

c_0, c_1, c_2 and c_3 are the coefficients. It's a degree 3 polynomial.

Internal Exam	15	23	18	23	24	22	22	19	19	16	2 4	11	24	16	23
External Exam	49	63	58	60	58	61	60	63	60	52	6 2	30	59	49	68
	3375														
	225														
	15	23	18	23	24	22	22	19	19	16	24	11	24	16	23
f(x)	49	63	58	60	58	61	60	63	60	52	62	30	59	49	68

Polynomial Regression Model

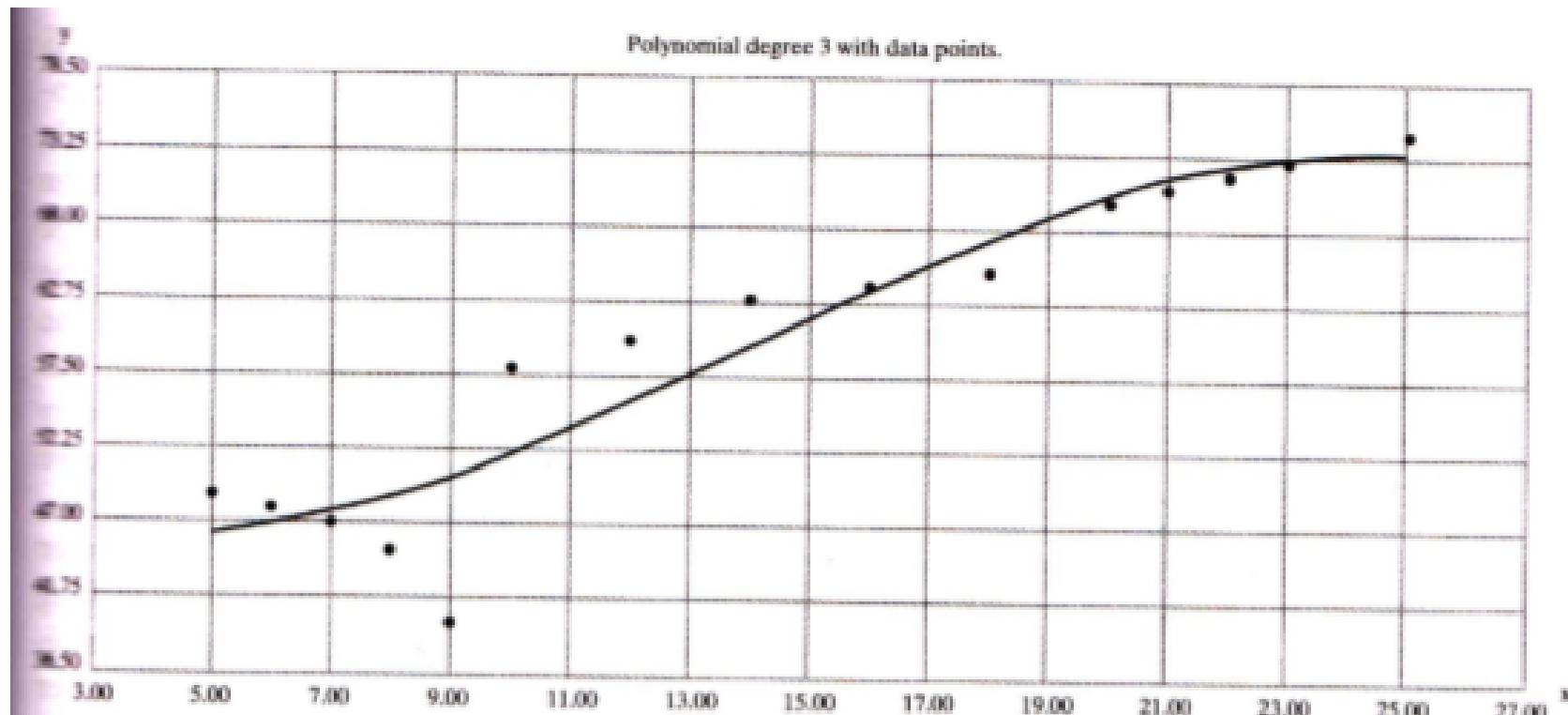


FIG. 8.16
Polynomial regression degree 3

Polynomial Regression Model

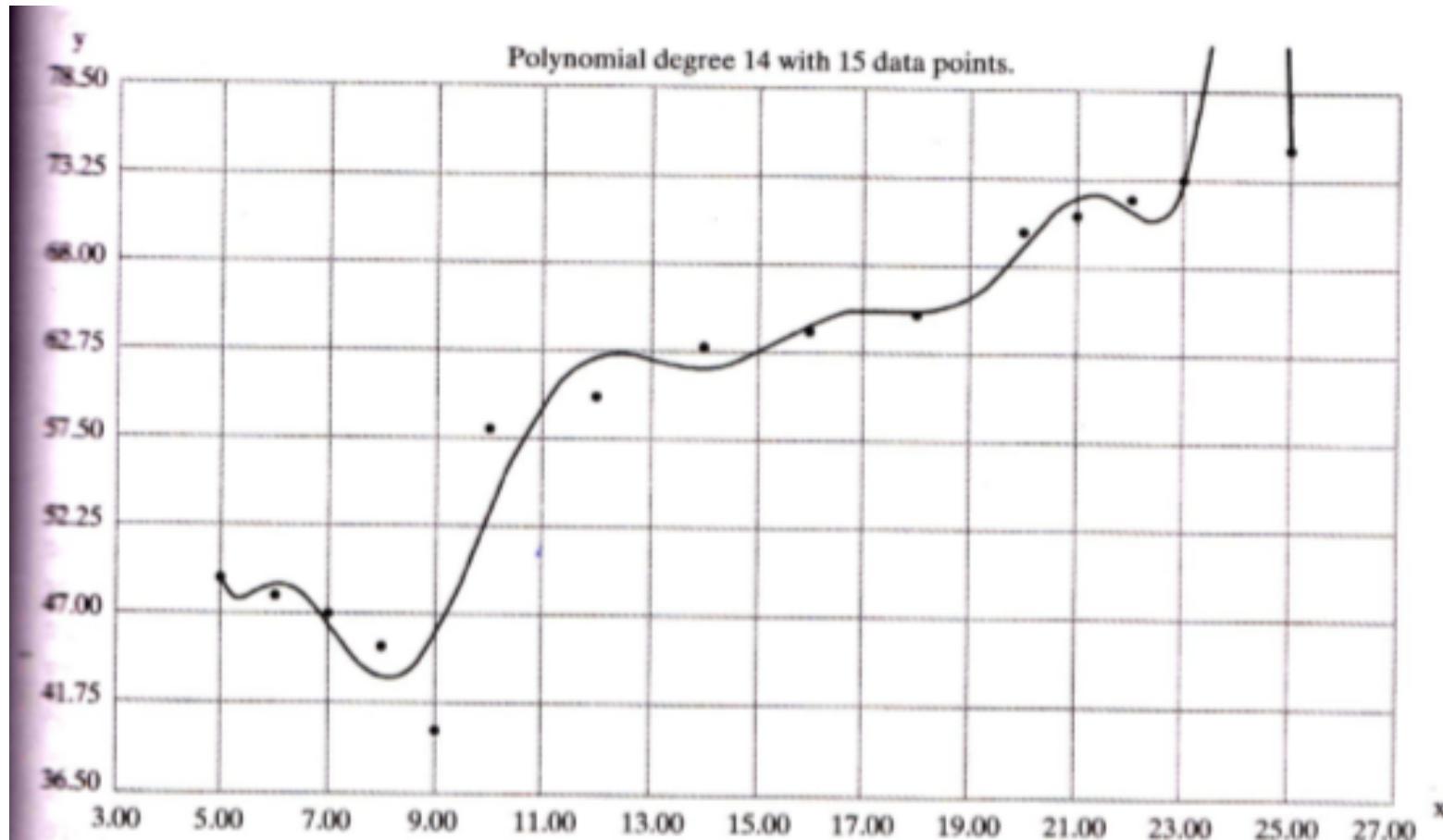


FIG. 8.17
Polynomial regression degree 14

Polynomial Regression Model

There are some relationships that a researcher will hypothesize is curvilinear. Clearly, such types of cases will include a polynomial term.

Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y .

An assumption in usual multiple linear regression analysis is that all the independent variables are independent. In polynomial regression model, this assumption is not satisfied.

Logistic Regression

- Logistic Regression is both classification and regression technique depending on the scenario used.
- It (logic regression) is a type of regression analysis used for predicting the outcome of a categorical dependent variable.
- Dependent variable (Y) is binary (0,1) and independent variables are continuous in nature.
- The goal of logistic regression is to predict the likelihood that Y is equal to 1 given certain values of X.
- So we predict probabilities rather than the scores of the dependent variable.

An example:

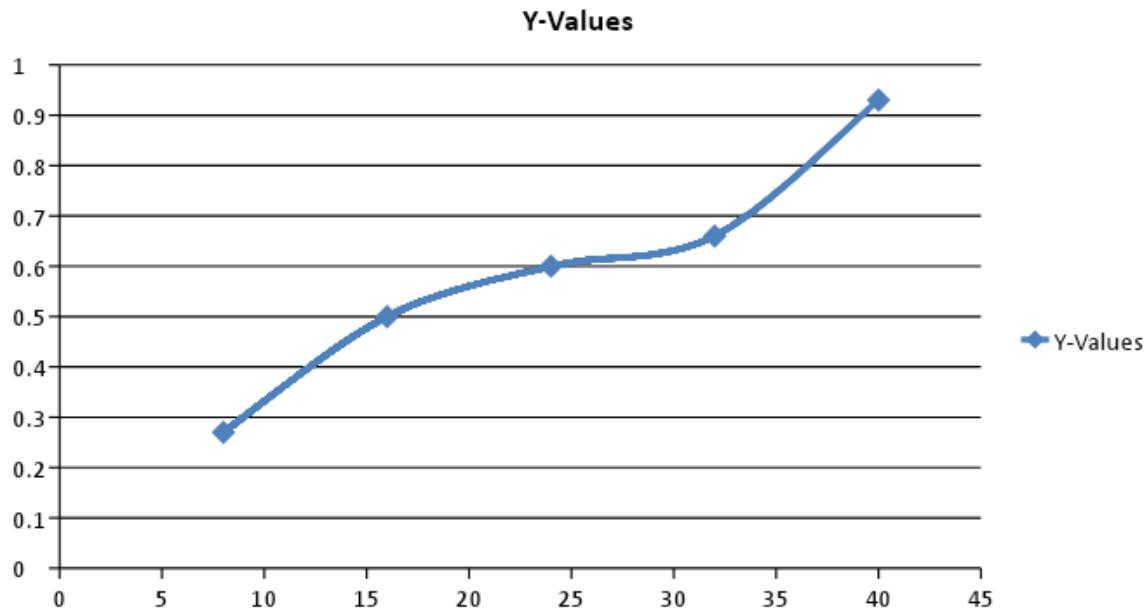
X	Y
0-8	0.27
9-16	0.5
17-24	0.6
25-32	0.66
33-40	0.93

X= experience in years

Y= Probability to be 1

Logistic Regression

X	Y
0-8	0.27
9-16	0.5
17-24	0.6
25-32	0.66
33-40	0.93



- A perfect relationship represents a perfectly curved S rather than a straight line.
- To model this relationship, we need some mathematics that accounts for the bend in the curve.

Logistic Regression

- Probability (P) can be computed from the regression equation.
- If we know the regression equation, we could calculate the expected probability that Y=1 for a given value of X.

$$P = \frac{\exp(a + bX)}{1 + \exp(a + bX)}$$

Given a height of 150 cm

We need to predict whether the person is male or female.

Let a = -100; b = 0.6

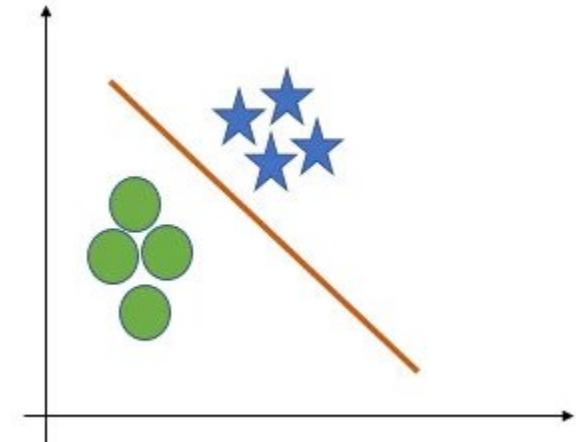
$$y = \frac{e^{(a+b \times X)}}{1+e^{(a+b \times X)}} = 0.000046$$

X	Y
0-8	0.27
9-16	0.5
17-24	0.6
25-32	0.66
33-40	0.93

Logistic Regression

- For a binary classification problem, target is (0 or 1)
- The Logistic function is defined as:

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



$$P = \frac{\exp(a + bX) / \exp(a + bX)}{1/\exp(a + bX) + \exp(a + bX)/\exp(a + bX)}$$

$$P = \frac{1}{1/\exp(a + bX) + 1}$$

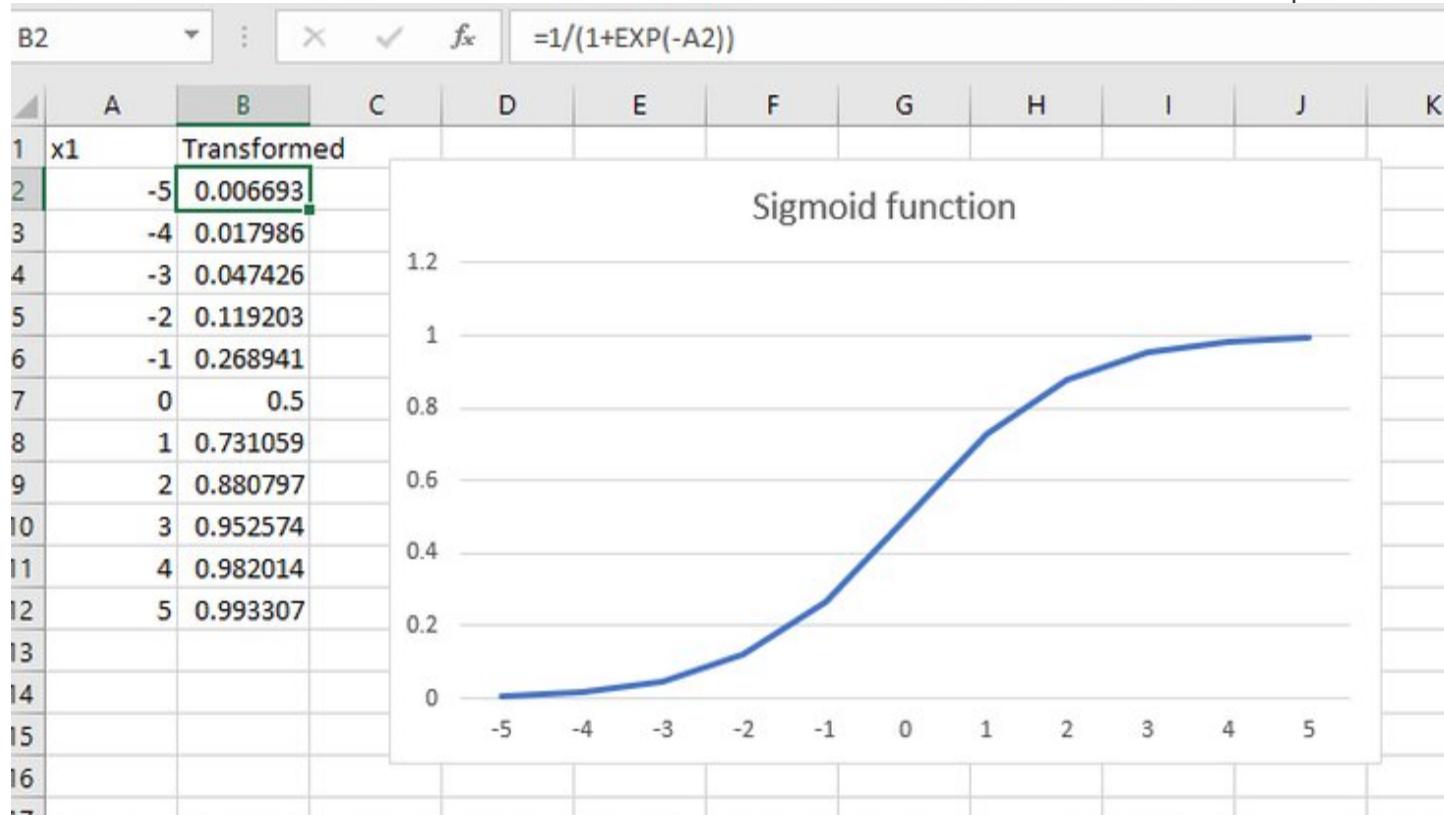
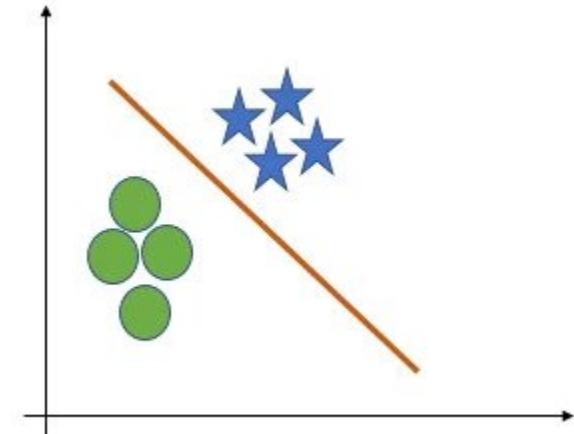
If, $z = (a + bX)$

$$P = \frac{1}{1/\exp(z) + 1} = \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression

- For a binary classification problem, target is (0 or 1)
- The Logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad \text{If, } z = (a + bX)$$



If probability is
> 0.5 then
default class
(class 0),
otherwise other
class (class 1)

Logistic Regression

The Logistic function is defined as:

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

$$Z = b_0 + b_1 * x_1 +$$

- The following is a dataset with 3 variables, where X1 and X2 are independent variable and Y is a dependent variable.

X1	X2	Y
2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
3.06407232	3.005305973	0
7.627531214	2.759262235	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
8.675418651	-0.242068655	1
7.673756466	3.508563011	1

Logistic Regression

$$Z = b_0 + b_1 * x_1 + b_2 * x_2$$

X1	X2	Y
2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
3.06407232	3.005305973	0
7.627531214	2.759262235	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
8.675418651	-0.242068655	1
7.673756466	3.508563011	1

- The job of the learning algorithm will be to discover the best values for the coefficients (b_0 , b_1 and b_2) based on the training data.
- Unlike linear regression, the output is transformed into a probability using the logistic function.

Logistic Regression

Logistic regression involves the following steps:

- Calculation of the Logit function that is $z = (a + bX)$
- Application of the Sigmoid function (Logistic function) to logit that is

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

- Calculation of the error, Cost function (Maximum Log-Likelihood).
- Application of learning algorithm to reduce the error and then repeat (if some error based learning algorithm is used)

Logistic regression by Stochastic Gradient Descent

- It works by using the model to calculate a prediction for each instance in training set and calculate error for each prediction.

Logistic Regression

We can calculate coefficients for logistic regression model as follows:

Given each training instance:

- Calculate a prediction using the current values of the coefficients.
- Calculate new coefficient values based on the error in the prediction.

This process is repeated until the model is accurate enough for fix number of iterations.

Probability of first training instance that belongs to class 0 that is $X_1 = 2.7810836$,

$x_2=2.550537003$, $Y=0$. Let

$$b_0 = 0; \quad b_1 = 0; \quad b_2 = 0$$

$$z = (b_0 + b_1 * x_1 + b_2 * x_2) = 0.0 + 0.0 * 2.7810836 + 0.0 * 2.550537003$$

$$\text{prediction} = 1 / (1 + e^{-z})$$

$$\text{prediction} = 1 / (1 + e^{(-(0.0 + 0.0 * 2.7810836 + 0.0 * 2.550537003))})$$

$$\text{Prediction } (f(z)) = 0.5$$

New Coefficients using gradient descent

$$b = b + a * (y - \text{prediction}) * \text{prediction} * (1 - \text{prediction}) * x [\text{(change of weight} = aex)]$$

Logistic Regression

New Coefficients using gradient descent

$$b = b + \alpha * (y - \text{prediction}) * \text{prediction} * (1 - \text{prediction}) * x$$

b_0 (intercept) will not have x value so it is assumed as 1 every time.

$$b_0 = 0 + 0.3 * (0 - 0.5) * 0.5 * (1 - 0.5) * 1.0 = -0.0375$$

$$b_1 = 0 + 0.3 * (0 - 0.5) * 0.5 * (1 - 0.5) * 2.7810836 = -0.104290635$$

$$b_2 = 0 + 0.3 * (0 - 0.5) * 0.5 * (1 - 0.5) * 2.550537003 = -0.09564513761$$

Now, repeat this process for $X_1 = 1.465489372$, $x_2 = 2.362125076$, $Y=0$.

$$b_0 = -0.0375; b_1 = -0.104290635; b_2 = -0.09564513761$$

$$z = (b_0 + b_1 * x_1 + b_2 * x_2) = -0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076)$$

$$\text{prediction} = 1 / (1 + e^{-(-0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076))})$$

$$\text{prediction} = 0.397$$

Logistic Regression

New Coefficients using gradient descent

Now, repeat this process for $X_1 = 1.465489372$, $x_2 = 2.362125076$, $Y=0$.

$b_0 = -0.0375$; $b_1 = -0.104290635$; $b_2 = -0.09564513761$

$$z = (b_0 + b_1 * x_1 + b_2 * x_2) = -0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076)$$

$$\text{prediction} = 1 / (1 + e^{-(-0.0375 + (-0.104290635 * 1.465489372) + (-0.09564513761 * 2.362125076))})$$

$$\text{prediction} = 0.397$$

Update:

$$b_0 = -0.0375 + 0.3 * (0 - 0.397) * 0.397 * (1 - 0.397) * 1.0 = -0.06605$$

$$b_1 = -0.104290635 + 0.3 * (0 - 0.397) * 0.397 * (1 - 0.397) * 1.465489372 = -0.1461$$

$$b_2 = -0.09564513761 + 0.3 * (0 - 0.397) * 0.397 * (1 - 0.397) * 2.362125076 = -0.1631$$

Logistic Regression

The first epoch coefficients are as follows:

X1	X2	Y	Prediction	Intercept	Coefficient X1	Coefficient X2
2.7810836	2.550537003	0	0.5	-0.0375	-0.104290635	-0.095645138
1.465489372	2.362125076	0	0.3974114	-0.06605	-0.146131968	-0.163086406
3.396561688	4.400293529	0	0.2175459	-0.07716	-0.183864977	-0.211970051
1.38807019	1.850220317	0	0.3263876	-0.09869	-0.213747009	-0.251801137
3.06407232	3.005305973	0	0.1808849	-0.10673	-0.238382985	-0.275964615
7.627531214	2.759262235	1	0.063777	-0.08996	-0.110466041	-0.229690614
5.332441248	2.088626775	1	0.2388946	-0.04844	0.110916445	-0.14297885
6.922596716	1.77106367	1	0.6144753	-0.02104	0.300586661	-0.094453991
8.675418651	-0.242068655	1	0.9314728	-0.01973	0.311970996	-0.094771646
7.673756466	3.508563011	1	0.885111	-0.01623	0.338866771	-0.082474472

$$b_0 = -0.01623; b_1 = 0.3388; b_2 = -0.0824$$

Logistic Regression

The 10th epoch is as follows:

X1	X2	Y	Prediction	Intercept	Coefficient X1	Coefficient X2	Prediction_round
2.781084	2.550537	0	0.316724951	-0.405242149	0.767058635	-1.101824964	0
1.465489	2.362125	0	0.131955957	-0.409776561	0.760413502	-1.112535813	0
3.396562	4.400294	0	0.06166044	-0.410846834	0.756778254	-1.117245328	0
1.38807	1.85022	0	0.193482972	-0.419904583	0.744205462	-1.134004159	0
3.064072	3.005306	0	0.175428153	-0.427517452	0.720879082	-1.156883159	0
7.627531	2.759262	1	0.867480845	-0.422947218	0.755738686	-1.144272685	1
5.332441	2.088627	1	0.77153982	-0.410866281	0.820159574	-1.119040116	1
6.922597	1.771064	1	0.963906322	-0.410489561	0.822767453	-1.118372921	1
8.675419	-0.24207	1	0.999087205	-0.410489311	0.822769619	-1.118372981	1
7.673756	3.508563	1	0.878613167	-0.406605464	0.852573316	-1.104746259	1

Thus, final coefficients are:

$$b_0 = -0.4066054641; b_1 = 0.8525733164; b_2 = -1.104746259$$

Now, prediction is < 0.5 then 0 else 1.

$$\text{accuracy} = (\text{correct predictions} / \text{number predictions made}) * 100$$

$$\text{accuracy} = (10 / 10) * 100$$

$$\text{accuracy} = 100\%$$

We can take new data and get prediction value

List of Popular Regression Algorithms

- [Linear Regression](#)
- [Polynomial Regression](#)
- [Logistic Regression](#)
- [Quantile Regression](#)
- [Ridge Regression](#)
- [Lasso Regression](#)
- [Elastic Net Regression](#)
- [Principal Components Regression \(PCR\)](#)
- [Partial Least Squares \(PLS\) Regression](#)
- [Support Vector Regression](#)
- [Ordinal Regression](#)
- [Poisson Regression](#)
- [Negative Binomial Regression](#)
- [Quasi Poisson Regression](#)
- [Cox Regression](#)
- [Tobit Regression](#)

Machine Learning (CS 431)

Presented by

Dr. Saroj Kr. Biswas
Associate Professor & HoD,
CSE



**Department of Computer Science and Engineering
National Institute of Technology, Silchar**

Supervised ML Algorithms

- Naïve Bayes
- Decision Tree (DT) [ID3, C4.5, C 5.0, CART]
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- **K- Nearest Neighbour (K-NN)**
- **Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

BP	Heart Beat	Weight	Class
120	70	50	Y
125	65	60	Y
130	59	52	N
150	78	70	N
135	66	85	N
125	75	82	N
120	76	90	Y

Roll. No	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3

Nearest Neighbour Based Classifiers

One of the simplest classifiers that can be used for classification is the nearest neighbour.

It classifies a sample based on the category of its nearest neighbour.

When large samples are involved, nearest neighbour classifier gives better result than any other classifiers.

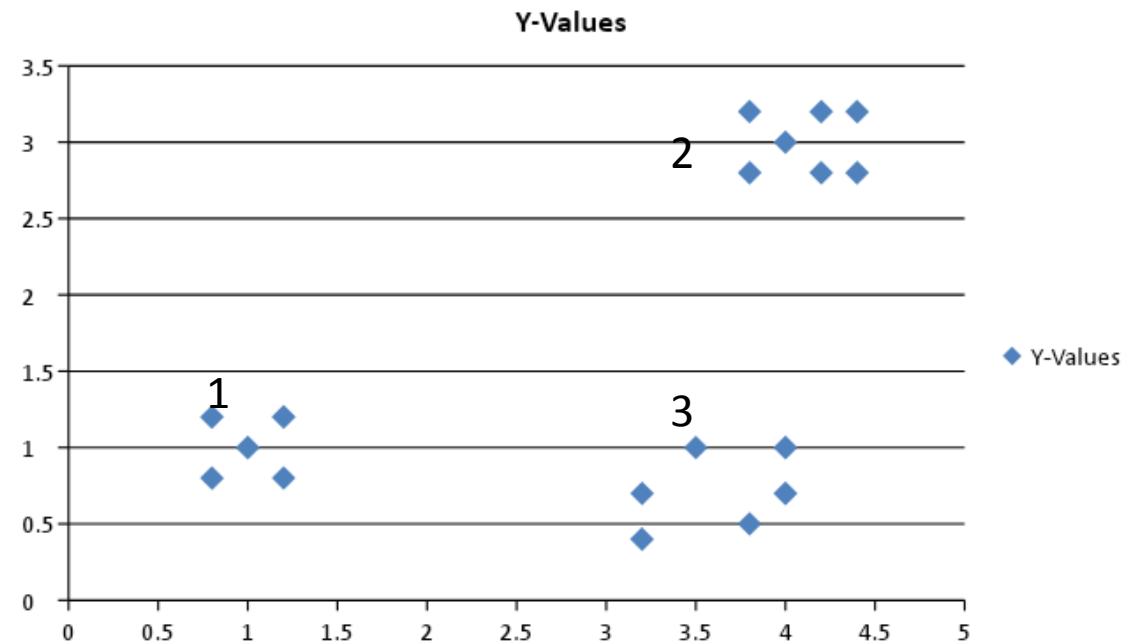
Nearest Neighbour Algorithm

The nearest neighbour algorithm assigns to a test sample the class label of its closest neighbour. Let there be n training patterns, (X_1, θ_1) , (X_2, θ_2) , (X_n, θ_n) where X_i is of dimension d and θ_i is the class label. If P is a test sample then if

$d(P, X_k) = \min\{d(P, X_i)\}$ where $i = 1, 2 \dots n$. Pattern P is assigned to the class θ_k associated with X_k .

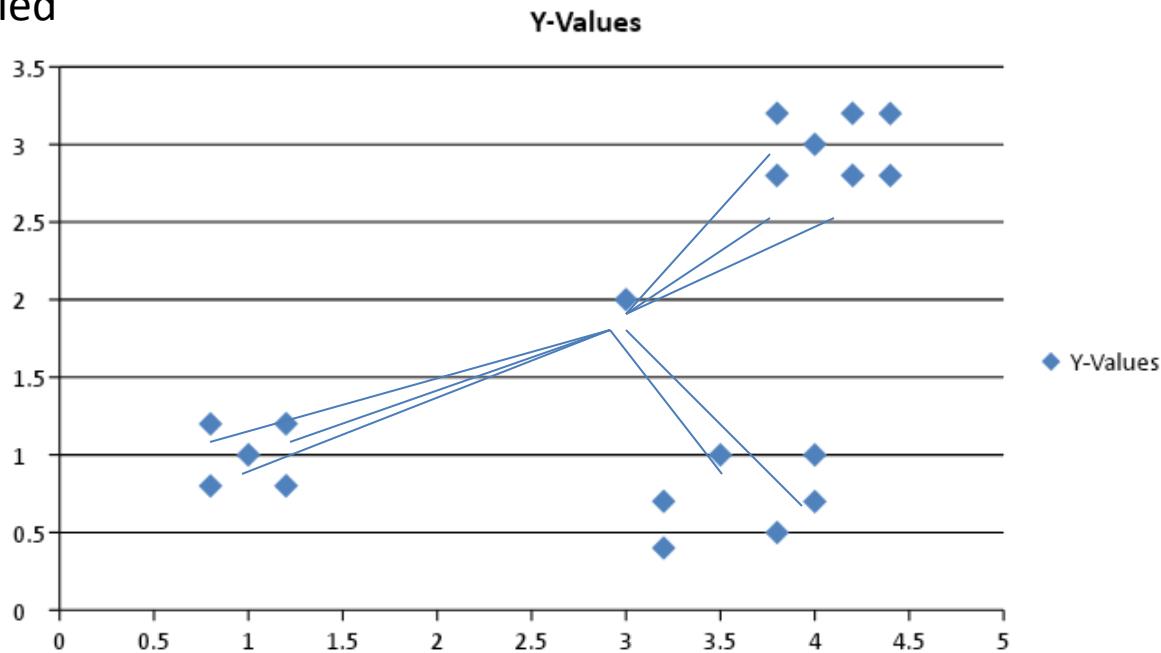
Nearest Neighbour Algorithm

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



Nearest Neighbour Algorithm

Suppose a new point (3.0, 2.0) P is given to be classified

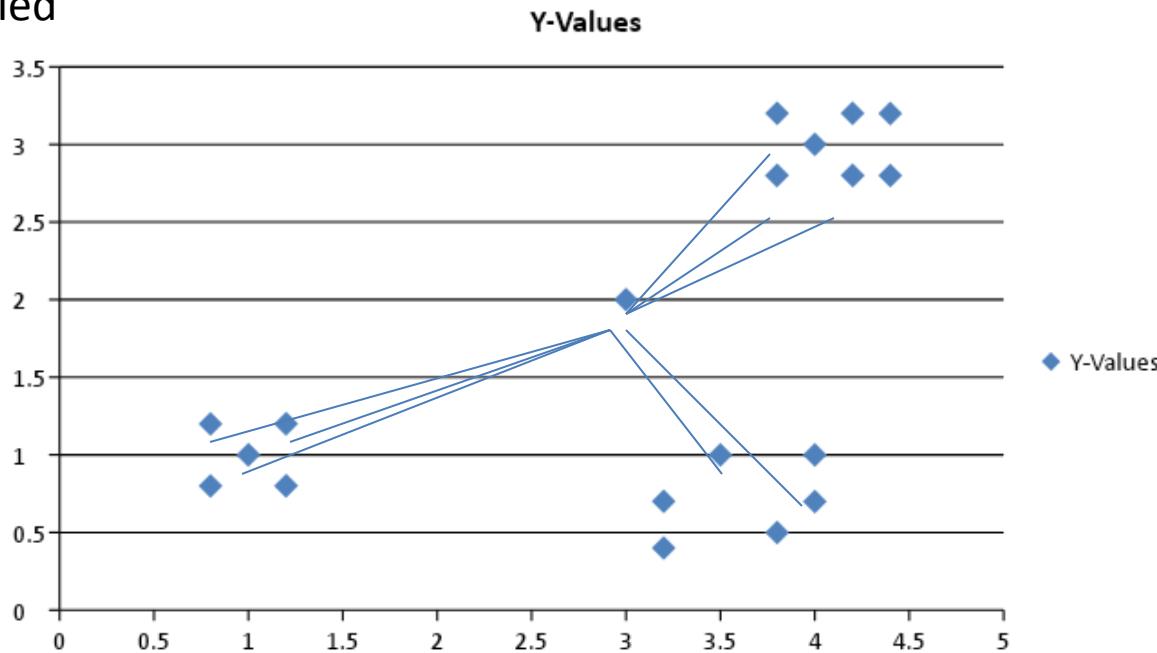


$$D(x_1, P) = \sqrt{(0.8 - 3.0)^2 + (0.8 - 2.0)^2} = 2.51$$

The closest neighbour of P is X16; Hence P belongs to class 3

k-Nearest Neighbour Algorithm

Suppose a new point (3.0, 2.0) P is given to be classified



If k is taken to be 5, the five nearest neighbours of P are X_{16} , X_7 , X_{14} , X_6 and X_{17} . The majority class of these five patterns is class 3. Hence P is classified as class 3.

The value chosen for k is crucial.

k-Nearest Neighbour Algorithm

This method will reduce the error in classification when training patterns are noisy.

For large dataset, k can be larger to reduce the error.

The value of k can be determined by experimentation using the concept of validation set.

- If P is a new pattern (4.2, 1.8), its nearest neighbour is X17 and hence P is classified to class 3.
- If the 5 nearest neighbours are taken, it is classified to class 2. The 5 nearest neighbours are X17, X16, X8, X7 and X11. X17 and X16 belonging to class 3, and X8, X7 and X11 belonging to class 2.

Drawbacks of Nearest Neighbour Algorithm

It's a lazy learning

For big dataset it takes huge time to find nearest neighbours

Modified k-Nearest Neighbour Algorithm (MkNN)

This algorithm is similar to the kNN algorithm.

The only difference is that these k nearest neighbours are weighted according to their distance from the test point.

This is also called the distance-weighted k-nearest neighbour algorithm.

Weight of each neighbour is defined as

$$w_j = \begin{cases} \frac{d_m - d_j}{d_m - d_1} & \\ & \end{cases}$$

The image shows a software window titled 'CLS Application'. It contains a table with columns for 'X' and 'Y'. Below the table is a formula: $w_j = \frac{d_m - d_j}{d_m - d_1}$. To the right of the formula, there is a note: 'Step 2 Divide output of step 1 by output of step 3 to calculate w'. Below that is another note: 'Step 3 Calculate w using the value of d'. At the bottom of the window, there are buttons for 'OK', 'Cancel', and 'Exit'.

$$M_{Ext} = 19.13 + 1.89 \times M_{Int}$$

Modified k-Nearest Neighbour Algorithm (MkNN)

$$P = (3.0, 2.0)$$

Distances of the five nearest points from P are

$$d(P, X16) = 1.12; \quad d(P, X7) = 1.13; \quad d(P, X14) = 1.32; \quad d(P, X6) = 1.41; \quad d(P, X17) = 1.41$$

$$w_{16} = 1$$

$$w_7 = \frac{1.41 - 1.13}{1.41 - 1.12} = 0.97$$

$$w_{14} = \frac{1.41 - 1.32}{1.41 - 1.12} = 0.31$$

$$w_6 = 0$$

$$w_{17} = 0$$

Class 1 sums = 0 (none of the patterns belongs to class 0)

Class 2 sums = $0.97 + 0 = 0.97$ (X7 and X6)

Class 3 sums = $1 + 0.31 + 0 = 1.31$ (X16, X14 and X17)

Finally P is classified to Class 3

r Nearest Neighbours

r-nearest neighbour takes all the neighbours within some distance r of the point of interest.
The algorithm is as follows:

Step1: Given the point P, determine the sub-set of data that lies in the ball of radius r centred at P.

$$B_r(P) = \{X_i \in X \mid \|P - X_i\| \leq r\}$$

Step2: If $B_r(P)$ is empty, then output the majority class of the entire dataset

Step3: If $B_r(P)$ is not empty, output the majority class of the data points in it.

This algorithm can be used to identify outliers.

The choice of the radius r is crucial to the algorithm.

P= (3.0, 2.0) patterns which are in a radius of 1.45 are X6, X7, X8, X9, X14, X16 and X17.
Majority patterns belong to class 2. P is therefore assigned to class 2.

Drawbacks of Nearest Neighbour Algorithm

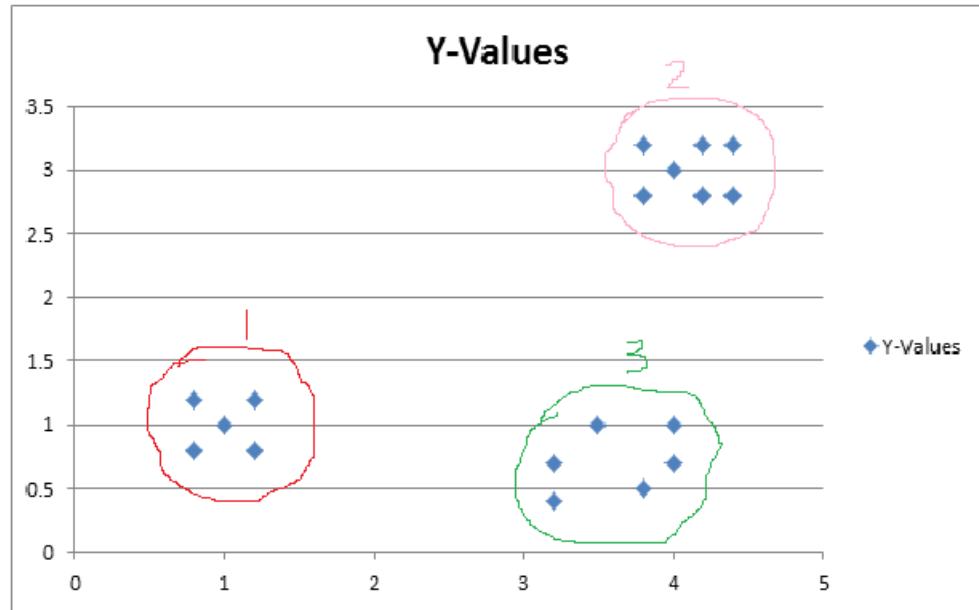
It's a lazy learning

For big dataset it takes huge time to find nearest neighbours

Nearest Neighbour with Clustering

Now a densed region can be represented by a representative pattern

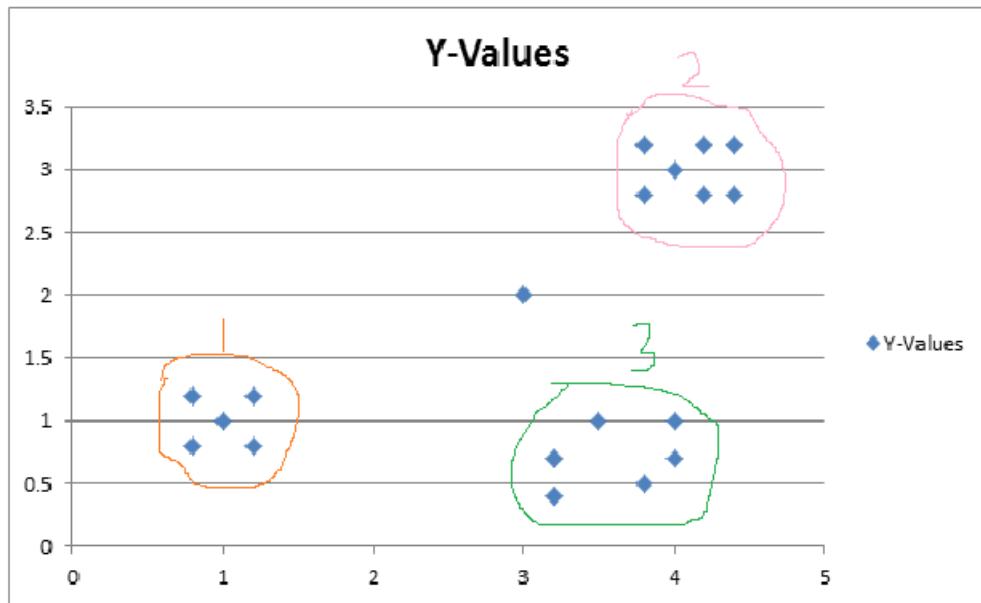
Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



Suppose centroid is the representative patterns
 Centroids are:
 $C1=(1.0, 1.0)$
 $C2=(4.11, 3)$
 $C3=(3.62, 0.72)$

Nearest Neighbour with Clustering

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



$$d(C_1, P) = 3.30$$

$$d(C_2, P) = 1.20$$

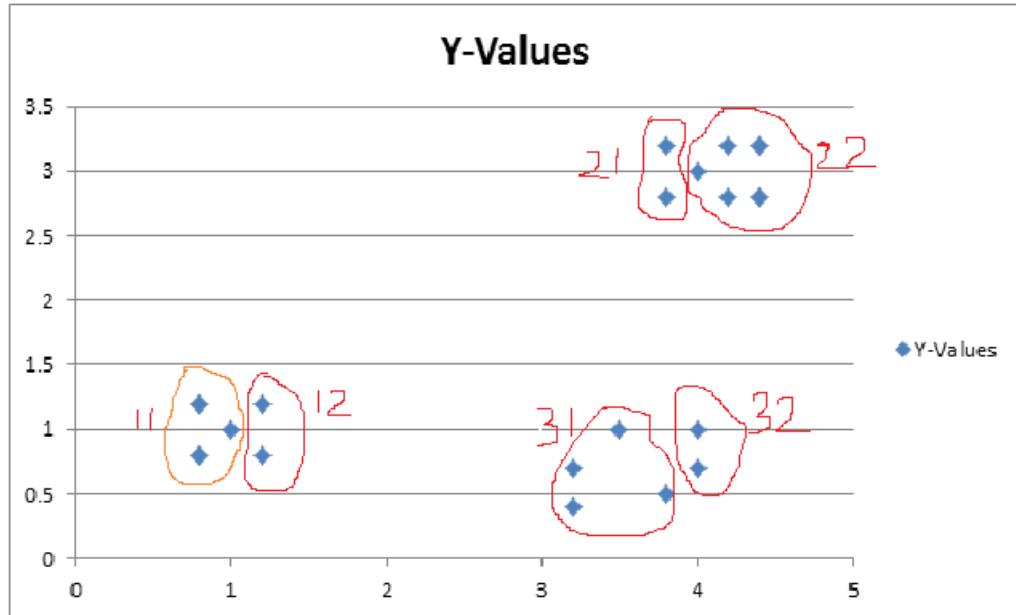
$$d(C_3, P) = 1.23$$

P belongs to class 2

Nearest Neighbour with Clustering

It is also possible to have more clusters for each class

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



$$C11 = (1.0, 0.867)$$

$$C12 = (1.0, 1.2)$$

$$C21 = (3.8, 3.0)$$

$$C22 = (4.24, 3.0)$$

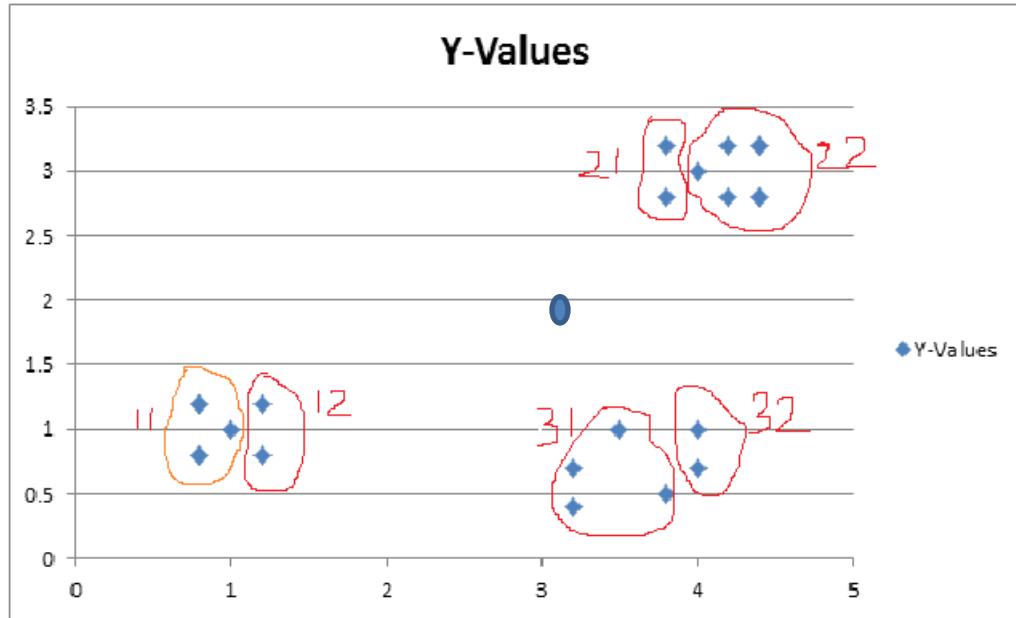
$$C31 = (3.43, 0.65)$$

$$C32 = (4.0, 0.85)$$

Nearest Neighbour with Clustering

It is also possible to have more clusters for each class

Patterns	A1	A2	Class
X1	0.8	0.8	1
X2	1.0	1.0	1
X3	1.2	0.8	1
X4	0.8	1.2	1
X5	1.2	1.2	1
X6	4.0	3.0	2
X7	3.8	2.8	2
X8	4.2	2.8	2
X9	3.8	3.2	2
X10	4.2	3.2	2
X11	4.4	2.8	2
X12	4.4	3.2	2
X13	3.2	0.4	3
X14	3.2	0.7	3
X15	3.8	0.5	3
X16	3.5	1.0	3
X17	4.0	1.0	3
X18	4.0	0.7	3



$$D(C_{11}, P) = 3.33$$

$$D(C_{12}, P) = 3.26$$

$$D(C_{21}, P) = 1.26$$

$$D(C_{22}, P) = 1.20$$

$$D(C_{31}, P) = 1.38$$

$$D(C_{32}, P) = 0.97$$

P is classified as class 3

Naive Bayes Classifier

Simplifying Bayes Classification

- Estimates probabilities of occurrence of different attribute values for the different classes in a training set.
- It uses these probabilities to classify recall patterns.

Name of pattern	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
T1	Gabby	Baked	Clogs	Student
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Student
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Student
T6	Quiet	Baked	Sandals	Student
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Student
R1	Quiet	Baked	Clogs	?
R2	Quiet	Roasted	Sandals	?
R3	Gabby	Roasted	Clogs	?
R4	Quiet	Roasted	Clogs	?

Simplifying Bayes Classification

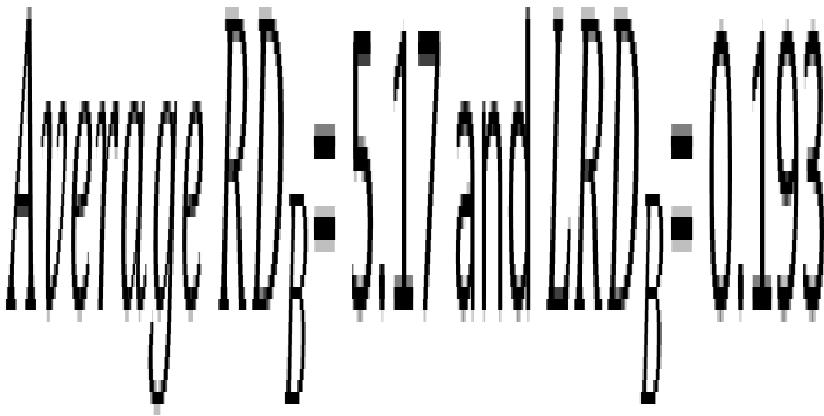
- o Estimates probabilities of occurrence of different attribute values for the different classes in a training set.
- o It uses these probabilities to classify recall patterns.

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals)	2/5	3/7

Simplifying Bayes Classification

- Suppose a training set has classes $C_1, C_2, C_3, \dots, C_m$, where ($m \geq 1$) and array of attributes $\bar{A} = A_1, A_2, A_3, \dots, A_M$ where ($M \geq 1$)
- The following probabilities are calculated.
 - **P (\bar{A})** = probability that a training pattern has attribute array \bar{A} .
 - **Prior probability, $P(C_k)$** : probability that a training pattern belongs to class C_k .
 - **Posterior probability, $P(C_k | \bar{A})$** : probability that a training pattern with attribute array \bar{A} belongs to class C_k . The attribute has discrete values.
 - **Conditional probability, $P (\bar{A} | C_k)$** : probability that a training pattern of class C_k has attribute array \bar{A} , the attributes having discrete values.

Simplifying Bayes Classification



Estimation of $P(C_k | \bar{A})$ from training set (cont..)

- Estimating $P(\bar{A}|\mathcal{C}_k)$ needs an impractically a large training set to consider values for all the attributes $A_1, A_2, A_3, \dots, A_M$. If these attributes $A_1, A_2, A_3, \dots, A_M$ are assumed to be class conditionally independent , then this classifier is a Naïve Bayes Classifier.

$$P(\bar{A}|\mathcal{C}_k) = \prod_{i=1}^M P(A_i|\mathcal{C}_k)$$

- To classify a pattern with attributes $A_1, A_2, A_3, \dots, A_M$, the equation,

is maximized, which is obtained by substituting $\prod_{i=1}^M P(A_i | C_k)$ for $P(\bar{A} | C_k)$ in equation (1).

Estimation of Prior Probabilities

- Let the number of patterns in class C_k is $|C_k|$ for $1 \leq k \leq m$.
 - In case of maximum-likelihood estimation,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}, (P(C_k) \geq 0) \dots \dots \dots (3)$$

- Alternatively, according to the Bayesian estimation

$$P(C_k) = \frac{|C_k|+1}{m + \sum_{j=1}^m |C_j|}, (P(C_k) > 0) \dots \dots \dots (4)$$

Estimation of Conditional Probabilities

- To maximize $P(\mathcal{C}_k) \prod_{i=1}^M P(A_i | \mathcal{C}_k)$, probability of $\prod_{i=1}^M P(A_i | \mathcal{C}_k)$ is required.
- The possible values of A_i be $V_{i_1}, V_{i_2}, V_{i_3}, \dots, V_{i_n}$ for $1 \leq i \leq M$.
- $|\mathcal{C}_k^{ij}|$ be the number of training patterns of class \mathcal{C}_k for which the attribute A_i is V_j where $1 \leq k \leq m, i_1 \leq j \leq i_n$.
- According to Maximum-likelihood estimation,

$$P(A_i = V_j | \mathcal{C}_k) = \frac{|\mathcal{C}_k^{ij}|}{|\mathcal{C}_k|}$$

- According to Bayesian estimation,

$$P(A_i = V_j | \mathcal{C}_k) = \frac{|\mathcal{C}_k^{ij}| + 1}{i_n + |\mathcal{C}_k|}$$

EXAMPLE

Name of pattern	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
T1	Gabby	Baked	Clogs	Student
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Student
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Student
T6	Quiet	Baked	Sandals	Student
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Student
R1	Quiet	Baked	Clogs	?
R2	Quiet	Roasted	Sandals	?
R3	Gabby	Roasted	Clogs	?
R4	Quiet	Roasted	Clogs	?

Name of pattern	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
T1	Gabby	Baked	Clogs	Student
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Student
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Student
T6	Quiet	Baked	Sandals	Student
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Student
R1	Quiet	Baked	Clogs	?
R2	Quiet	Roasted	Sandals	?
R3	Gabby	Roasted	Clogs	?
R4	Quiet	Roasted	Clogs	?

Average RD_A =

$$\frac{1}{3} \sum_3 \max \left[\text{3rd distance of } A's \text{ neighbor}, \text{distance}(A, \text{the neighbor}) \right]$$

$$\begin{aligned}
 &= \frac{1}{3} [\max(5, 6.08) + \max(4.2, 6.32) + \max(1.4, 5.8)] \\
 &= \frac{1}{3} [6.08 + 6.32 + 5.8] = 6.06
 \end{aligned}$$

All the Probabilities

Probability	Estimates	
	Maximum-likelihood	Bayesian
$P(\text{professor})$	3/8	4/10
$P(\text{HABIT} = \text{gabby}) \text{professor}]$	2/3	3/5
$P(\text{HABIT} = \text{quiet}) \text{professor}]$	1/3	2/5
$P(\text{EATS} = \text{baked}) \text{professor}]$	0/3	1/6
$P(\text{EATS} = \text{fried}) \text{professor}]$	2/3	3/6
$P(\text{EATS} = \text{roasted}) \text{professor}]$	1/3	2/6
$P(\text{FOOTWEAR} = \text{clogs}) \text{professor}]$	0/3	1/5
$P(\text{FOOTWEAR} = \text{sandals}) \text{professor}]$	3/3	4/5
$P(\text{student})$	5/8	6/10
$P(\text{HABIT} = \text{gabby}) \text{student}]$	3/5	4/7
$P(\text{HABIT} = \text{quiet}) \text{student}]$	2/5	3/7
$P(\text{EATS} = \text{baked}) \text{student}]$	3/5	4/8
$P(\text{EATS} = \text{fried}) \text{student}]$	2/5	3/8
$P(\text{EATS} = \text{roasted}) \text{student}]$	0/5	1/8
$P(\text{FOOTWEAR} = \text{clogs}) \text{student}]$	3/5	4/7
$P(\text{FOOTWEAR} = \text{sandals}) \text{student}]$	2/5	3/7

Classifying the Professor-Student patterns

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals) student]	2/5	3/7

- Let us classify the R3 pattern of the professor-student recall set.

- The attribute values of the pattern are HABIT = gabby, EATS = roasted and FOOTWEAR = clogs.

$$\begin{aligned} P(C_k | \bar{A}) &= \frac{P(C_k)P(A|C_k)}{P(\bar{A})} = P(C_k)P(\bar{A}|C_k) \\ &= P(C_k) \prod_{i=1}^M P(A_i|C_k) \end{aligned}$$

- Using maximum-likelihood estimates belong to Professor,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}$$

$$P(A_t = V_j | C_k) = \frac{|C_k^{ij}|}{|C_k|}$$

$P(\text{professor}) \times P(\text{HABIT} = \text{gabby}) | \text{professor}] \times P[(\text{EATS} = \text{roasted}) | \text{professor}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{professor}]$

$$= \frac{3}{8} * \frac{2}{3} * \frac{1}{3} * \frac{0}{3} = 0$$

Classifying the Professor-Student patterns

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals) student]	2/5	3/7

- Using **maximum-likelihood estimates** belong to student class

$P(\text{student}) \times P(\text{HABIT} = \text{gabby}) | \text{student}] \times P[(\text{EATS} = \text{roasted}) | \text{student}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{student}]$

$$= \frac{5}{8} * \frac{3}{5} * \frac{0}{5} * \frac{3}{5} = 0$$

- The recall pattern R3 is rejected because the values of both the classes are zero. This is a disadvantage of maximum-likelihood estimates of probabilities, for one zero probability has nullified the influence of the other probabilities, for each class.

Classifying the Professor-Student patterns (cont..)

Probability	Estimates	
	Maximum-likelihood	Bayesian
P(professor)	3/8	4/10
P(HABIT = gabby) professor]	2/3	3/5
P(HABIT = quiet) professor]	1/3	2/5
P(EATS = baked) professor]	0/3	1/6
P(EATS = fried) professor]	2/3	3/6
P(EATS = roasted) professor]	1/3	2/6
P(FOOTWEAR = clogs) professor]	0/3	1/5
P(FOOTWEAR = sandals) professor]	3/3	4/5
P(student)	5/8	6/10
P(HABIT = gabby) student]	3/5	4/7
P(HABIT = quiet) student]	2/5	3/7
P(EATS = baked) student]	3/5	4/8
P(EATS = fried) student]	2/5	3/8
P(EATS = roasted) student]	0/5	1/8
P(FOOTWEAR = clogs) student]	3/5	4/7
P(FOOTWEAR = sandals) student]	2/5	3/7

- Using Bayesian estimation belong to Professor
- $$P(\text{professor}) \times P(\text{HABIT} = \text{gabby} | \text{professor}) \times P[(\text{EATS} = \text{roasted}) | \text{professor}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{professor}]$$
- $$= \frac{2}{5} * \frac{3}{5} * \frac{1}{3} * \frac{1}{5} = 0.016$$

Belong to student:

- $$P(\text{student}) \times P(\text{HABIT} = \text{gabby} | \text{student}) \times P[(\text{EATS} = \text{roasted}) | \text{student}] \times P[(\text{FOOTWEAR} = \text{clogs}) | \text{student}]$$
- $$= \frac{3}{5} * \frac{4}{7} * \frac{1}{8} * \frac{4}{7} = 0.0245$$
- Since, the value of student is 0.0245 which is more than the value of the professor which is 0.016, the recall pattern R3 is classified as student.

Classifying the Professor-Student patterns (cont..)

Recall Patterns	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor
R3	Gabby	Roasted	Clogs	Student
R4	Quiet	Roasted	Clogs	Student

Naïve Bayes with Continuous Attribute

- It performs well in case of [categorical data as compared to numeric data.](#)
- So, how do we perform classification using [Naïve Bayes](#) when the data is continuous in nature.
- There are two ways to handle continuous attributes in naïve Bayes classifiers:
 - i. We can discretize each continuous attribute and then replace the continuous [attribute](#) value with its corresponding discrete interval. However the estimation error depends on the discretization strategy, as well as the number of discrete intervals. If the number of intervals is too large, there are too few training records in each interval to provide a reliable estimate. if the number of intervals is too small, then some intervals may aggregate records from different classes and we may miss the correct decision boundary. Hence, there is no rule of thumb on the discretisation strategy.

Bayes

Problems with discretization strategy

Humidity

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

Large no of intervals

65-69	1
70-75	4
76-80	2
81-85	1
86-90	2
91-95	2
96-100	1

Less no of intervals

65-85	8 (6 yes, 2 no)
86- 100	6 (3 yes, 3 no)

Bayes

ii. We can assume a probability distribution for the continuous variable and estimate the parameters of the distribution using the training data. A Gaussian distribution is usually chosen to represent the class-conditional probability for continuous attributes. The distribution is characterized by two parameters, its mean and variance.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

where μ_a is the *sample mean*: $\mu_a = \frac{1}{|D_a|} \sum_{x \in D_a} x.a$
 σ_a is the *sample standard deviation*, and
 σ_a^2 the *sample variance*: $\sigma_a^2 = \frac{1}{|D_a|-1} \sum_{x \in D_a} (x.a - \mu_a)^2$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

- Maximum-likelihood estimation,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}$$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

- Maximum-likelihood estimation,

$$P(A_i = V_j | C_k) = \frac{|C_k|^{ij}}{|C_k|}$$

$$P(\text{Outlook=sunny} | \text{Yes}) = \frac{2}{9}$$

$$P(\text{Outlook=overcast} | \text{Yes}) = \frac{4}{9}$$

$$P(\text{Outlook=Rainy} | \text{Yes}) = \frac{3}{9}$$

$$P(\text{Outlook=sunny} | \text{No}) = \frac{3}{5}$$

$$P(\text{Outlook=overcast} | \text{No}) = \frac{0}{5}$$

$$P(\text{Outlook=Rainy} | \text{No}) = \frac{2}{5}$$

$$P(\text{Windy=false} | \text{Yes}) = \frac{6}{9}$$

$$P(\text{Windy=true} | \text{Yes}) = \frac{3}{9}$$

$$P(\text{Windy=false} | \text{No}) = \frac{2}{5}$$

$$P(\text{Windy=true} | \text{No}) = \frac{3}{5}$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

- Maximum-likelihood estimation,

$$P(C_k) = \frac{|C_k|}{\sum_{j=1}^m |C_j|}$$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

Continuous attribute:

Temperature: Yes : 83, 70, 68, 64, 69, 75, 75, 72, 81

$$\mu_T = 73; \quad \sigma_T = 6.2$$

Temperature: No : 85, 80, 65, 72, 71

$$\mu_T = 75; \quad \sigma_T = 7.9$$

Humidity: Yes : 86, 96, 80, 65, 70, 80, 70, 90, 75

$$\mu_H = 79; \quad \sigma_H = 10.2$$

Temperature: No : 85, 90, 70, 95, 91

$$\mu_H = 86; \quad \sigma_H = 9.7$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

We are going to classify an instance
 $x = \langle \text{Outlook}=\text{sunny}, \text{Temperature}=66, \text{Humidity}=90, \text{Windy=True} \rangle$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

Probability for Play= Yes

$$P(\text{Outlook}=\text{sunny} | \text{Yes}) = \frac{2}{9}$$

$$P(\text{Windy}=\text{true} | \text{Yes}) = \frac{3}{9}$$

Temperature=66

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

$$x = 66; \mu_T = 73; \sigma_T = 6.2$$

$$f(x=\text{Temp}) = \frac{1}{\sqrt{2\pi} \cdot 6.2} e^{-\frac{(66-73)^2}{2(6.2)^2}} = \frac{1}{\sqrt{38.955}} e^{-\frac{49}{76.88}}$$

$$= 0.16e^{-0.64} = 0.084$$

Humidity: $x = 90; \mu_H = 79; \sigma_H = 10.2$

$$f(x=\text{Hum}) = \frac{1}{\sqrt{2\pi} \cdot 10.2} e^{-\frac{(90-79)^2}{2(10.2)^2}} = \frac{1}{\sqrt{64.056}} e^{-\frac{121}{208.08}}$$

$$= 0.13e^{-0.58} = 0.073$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

We are going to classify an instance
 $x = \langle \text{Outlook}=\text{sunny}, \text{Temperature}=66, \text{Humidity}=90, \text{Windy=True} \rangle$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

Posterior probability of x to belong to Yes class
 $(play=yes)$

$$\begin{aligned} P(x/\text{yes}) * P(\text{yes}) &= P(\text{sunny}/\text{yes}) * \\ P(\text{Temperature}=66/\text{yes}) * P(\text{Humidity}=90/\text{yes}) * \\ P(\text{True}/\text{yes}) * P(\text{yes}) \\ &= (2/9) * 0.084 * 0.073 * (3/9) * (9/14) = 0.00029 \end{aligned}$$

Probability for Play= No

$$\begin{aligned} P(\text{Outlook}=\text{sunny} | \text{No}) &= \frac{3}{5} \\ P(\text{Windy}= \text{true} | \text{No}) &= \frac{3}{5} \end{aligned}$$

Temperature=66

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

$$x = 66; \mu_T = 75; \sigma_T = 7.9$$

$$\begin{aligned} f(x=\text{Temp}) &= \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}} = \\ \frac{1}{\sqrt{49.612}} e^{-\frac{81}{124.84}} \end{aligned}$$

$$= 0.14 e^{-0.65} = 0.073$$

Bayes

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

We are going to classify an instance

$x = \langle \text{Outlook}=\text{sunny}, \text{ Temperature}=66, \text{ Humidity}=90, \text{ Windy}=\text{True} \rangle$

$$P(\text{Yes}) = \frac{9}{9+5} = \frac{9}{14}$$

$$P(\text{No}) = \frac{5}{9+5} = \frac{5}{14}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

Humidity: $x = 90; \mu_H = 86; \sigma_H = 9.7$

$$\begin{aligned} f(x=\text{Hum}) &= \frac{1}{\sqrt{2 \cdot 3.14 \cdot 9.7}} e^{-\frac{(90-86)^2}{2(9.7)^2}} = \\ &\frac{1}{\sqrt{60.92}} e^{-\frac{16}{188.18}} \\ &= 0.128 e^{-0.085} = 0.12 \end{aligned}$$

Posterior probability of x to belong to No class
($\text{play}=\text{no}$)

$$\begin{aligned} P(x/\text{no}) * P(\text{no}) &= P(\text{sunny}/\text{no}) * P(\text{Temperature}=66/\text{no}) \\ &\quad * P(\text{Humidity}=90/\text{no}) * P(\text{True}/\text{no}) * P(\text{no}) \\ &= (3/5) * 0.073 * 0.12 * (3/5) * (5/14) = 0.00113 \end{aligned}$$

$0.00113 > 0.00113 = (\text{play}=\text{no}) > (\text{play}=\text{yes})$:

Classification — NO

Decision Tree

- Decision Tree produces **interpretable** output in human readable form.
- **Decision rules** are constructed directly from decision tree output, traversing path from the root node to a given leaf node.
- Decision rules have form **IF antecedent THEN consequent**.
- Antecedent consists of attributes values from branches of given path.

Classifying the Recall patterns

If Footwear = Clogs

Then pattern class= Student

If Footwear = Sandals

and Eats = Baked,

Then pattern class= Student

If Footwear = Sandals

and Eats = Fried,

Then pattern class=

Professor

If Footwear = Sandals

and Eats = Roasted,

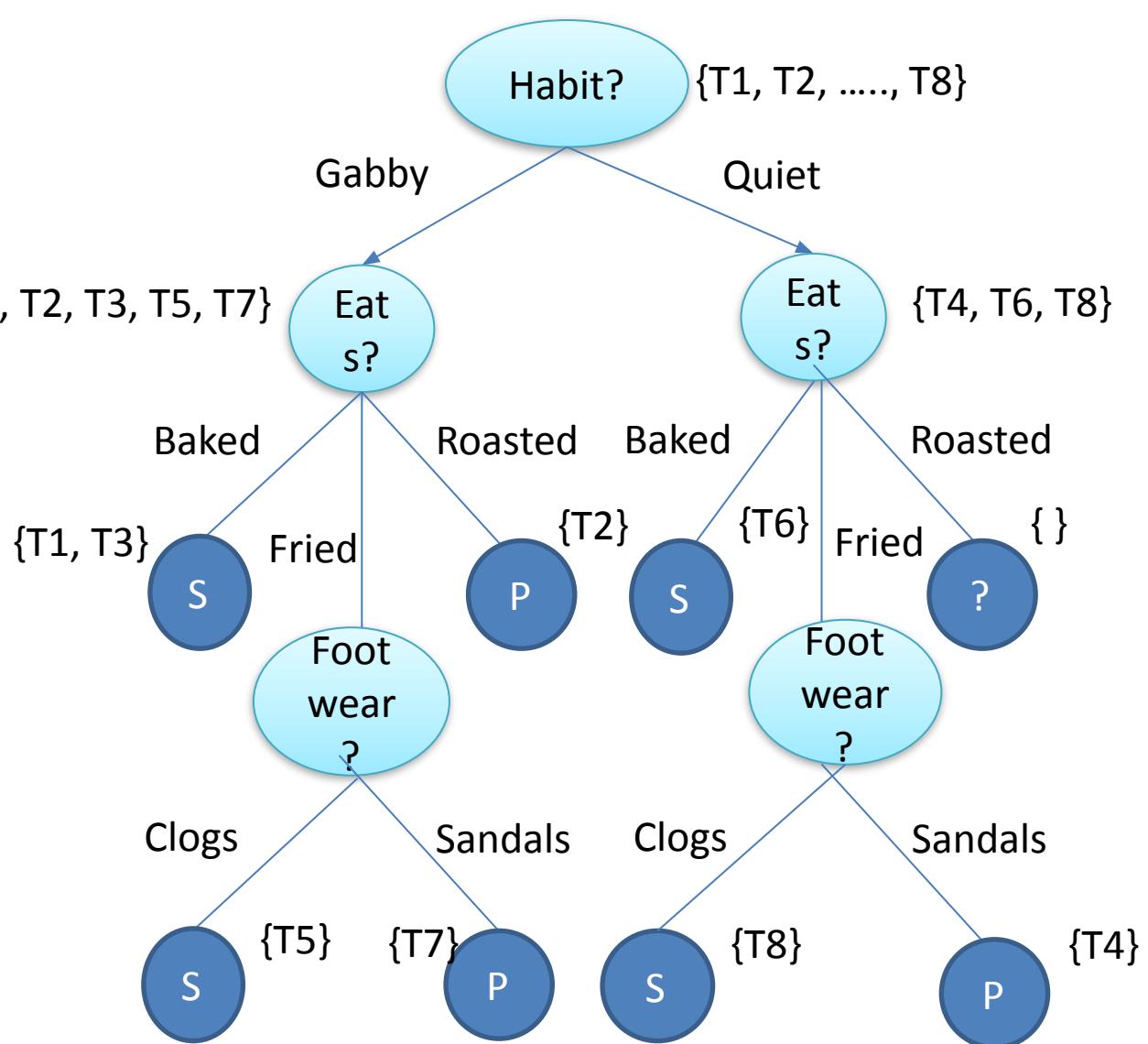
Then pattern class= Professor

Recall	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor
R3	Gabby	Roasted	Clogs	Student
R4	Quiet	Roasted	Clogs	Student

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	Students
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Students
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	Clogs	Students
T6	Quiet	Baked	Sandals	Students
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Students

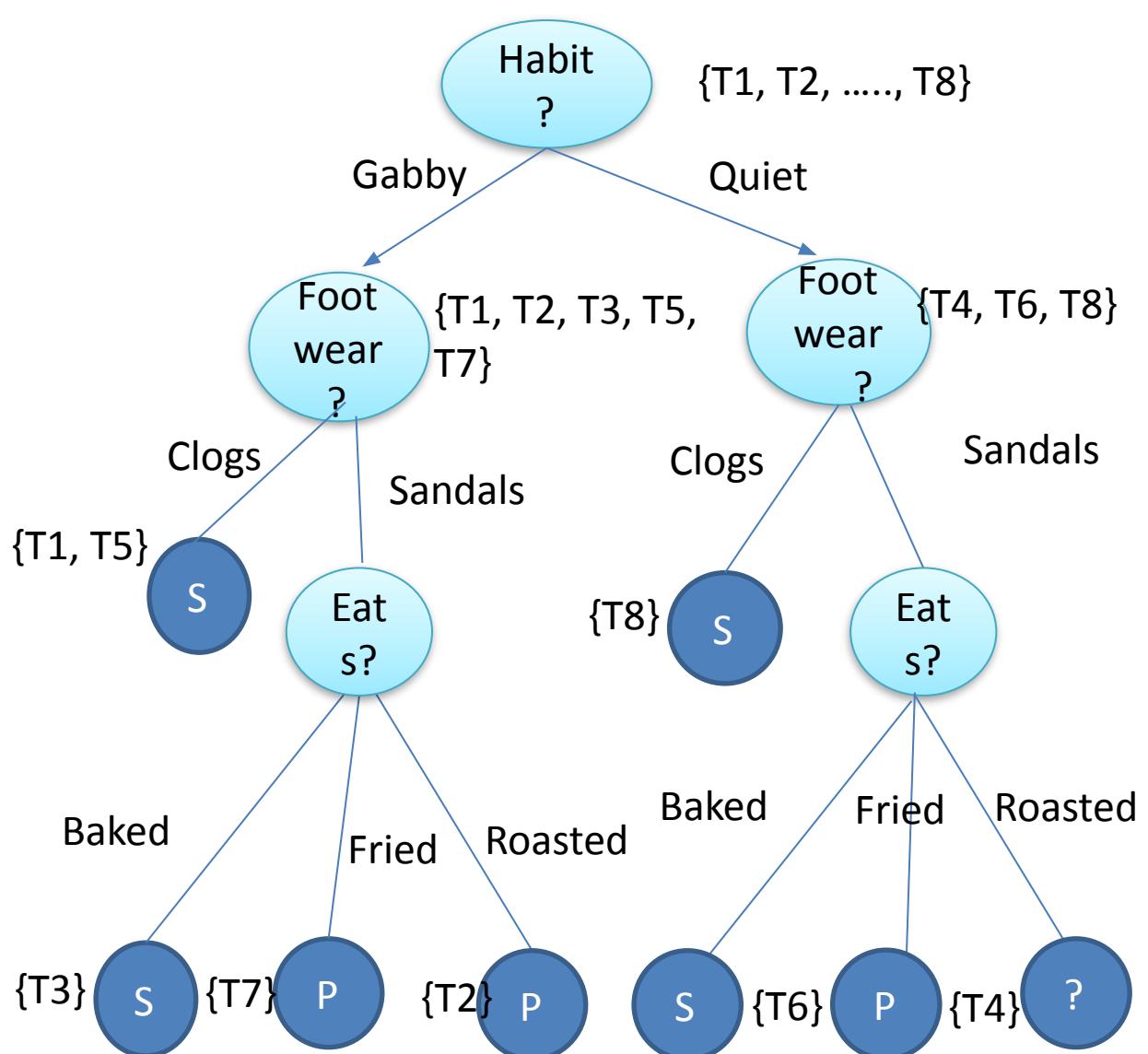
NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

12-11-2022



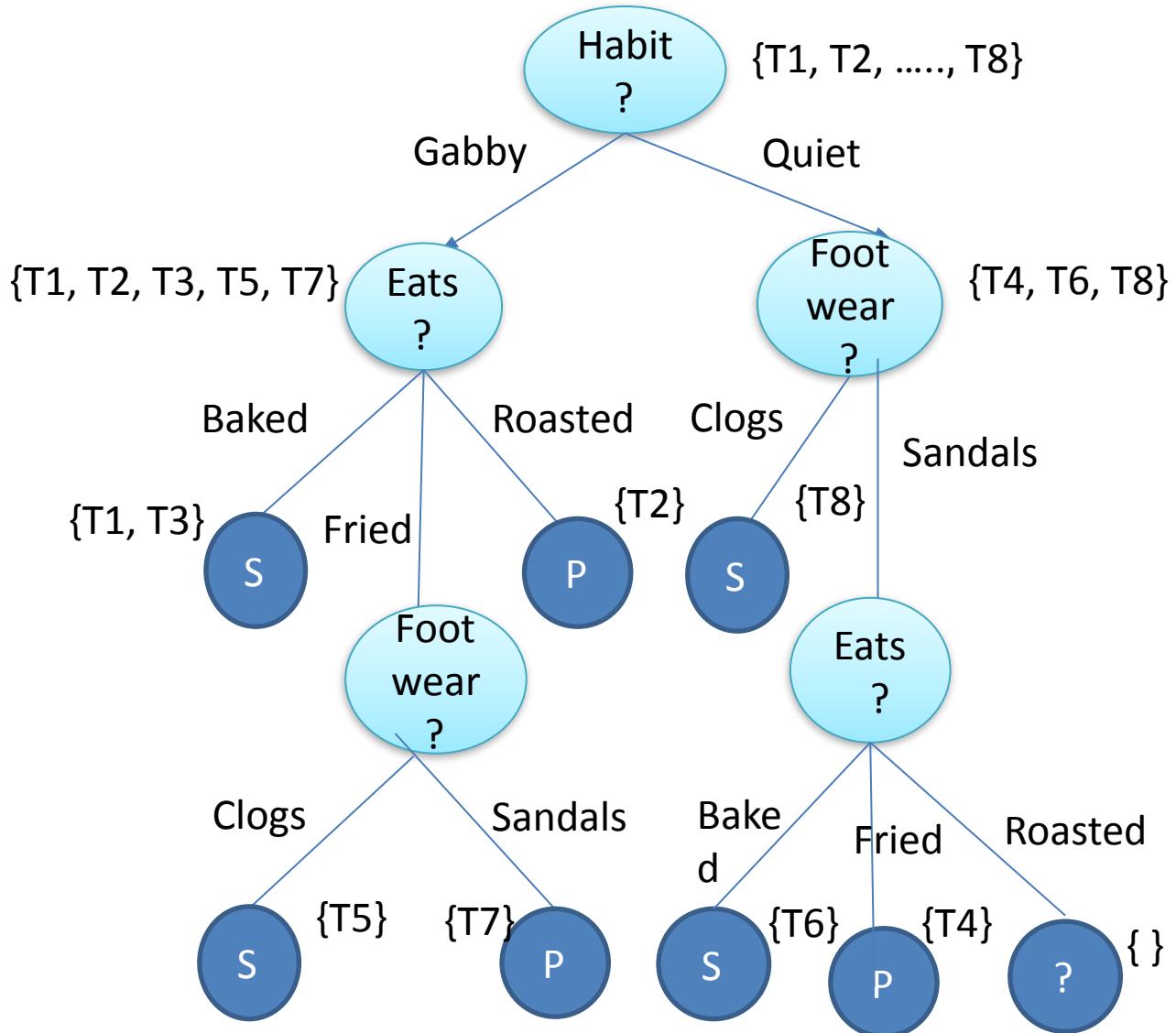
NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

12-11-2022



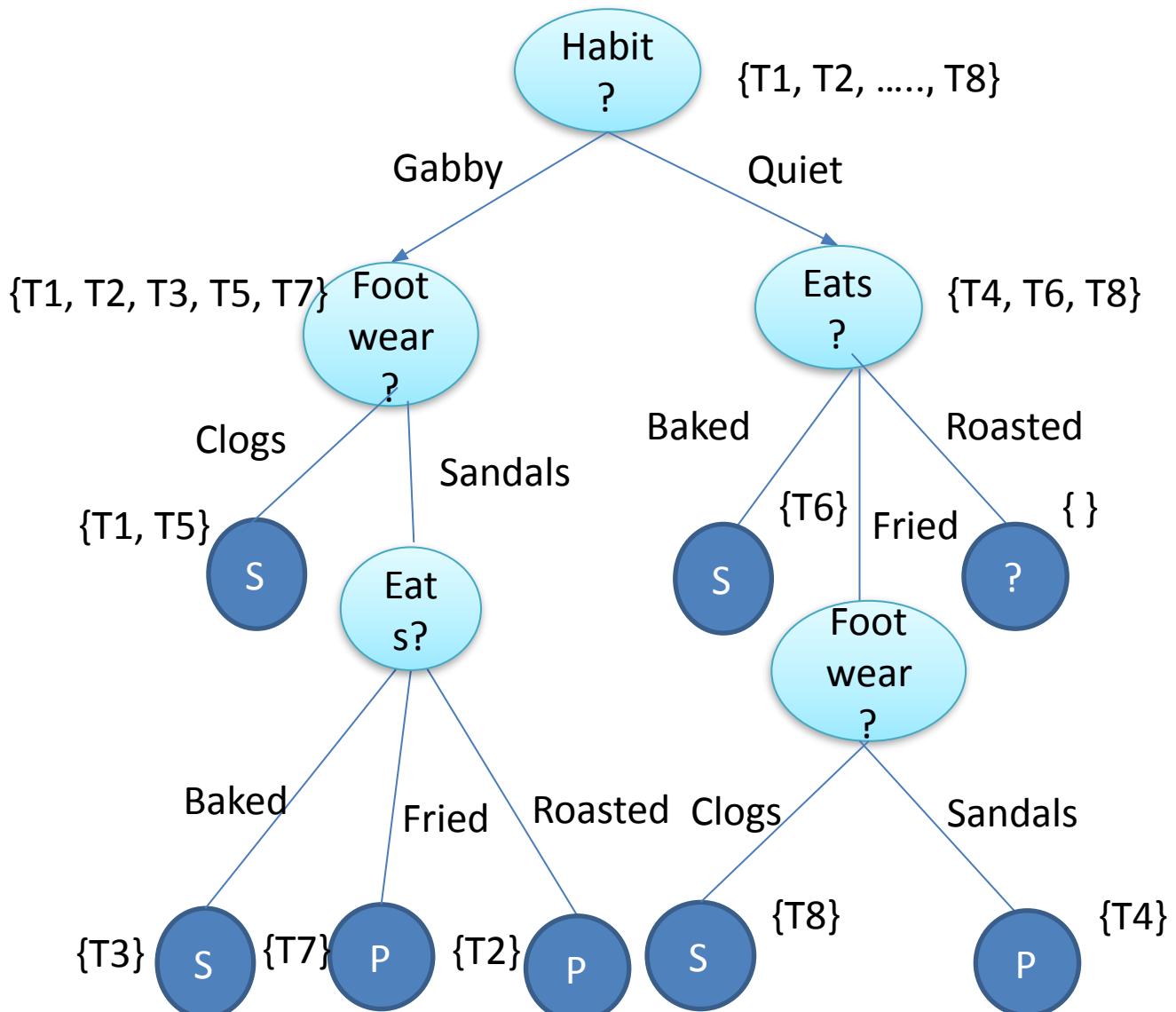
NAME of trainin g patter n	Attributes			Clas s
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

12-11-2022



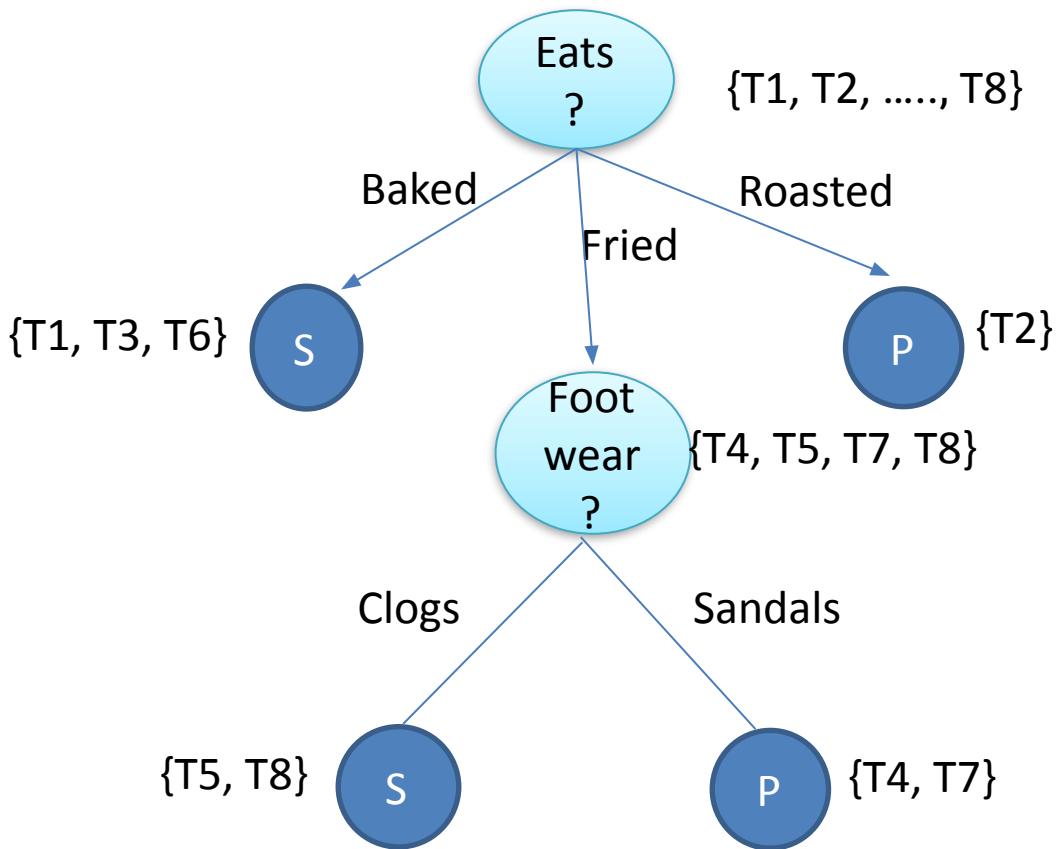
NAME of trainin g patter n	Attributes			Clas s
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

12-11-2022

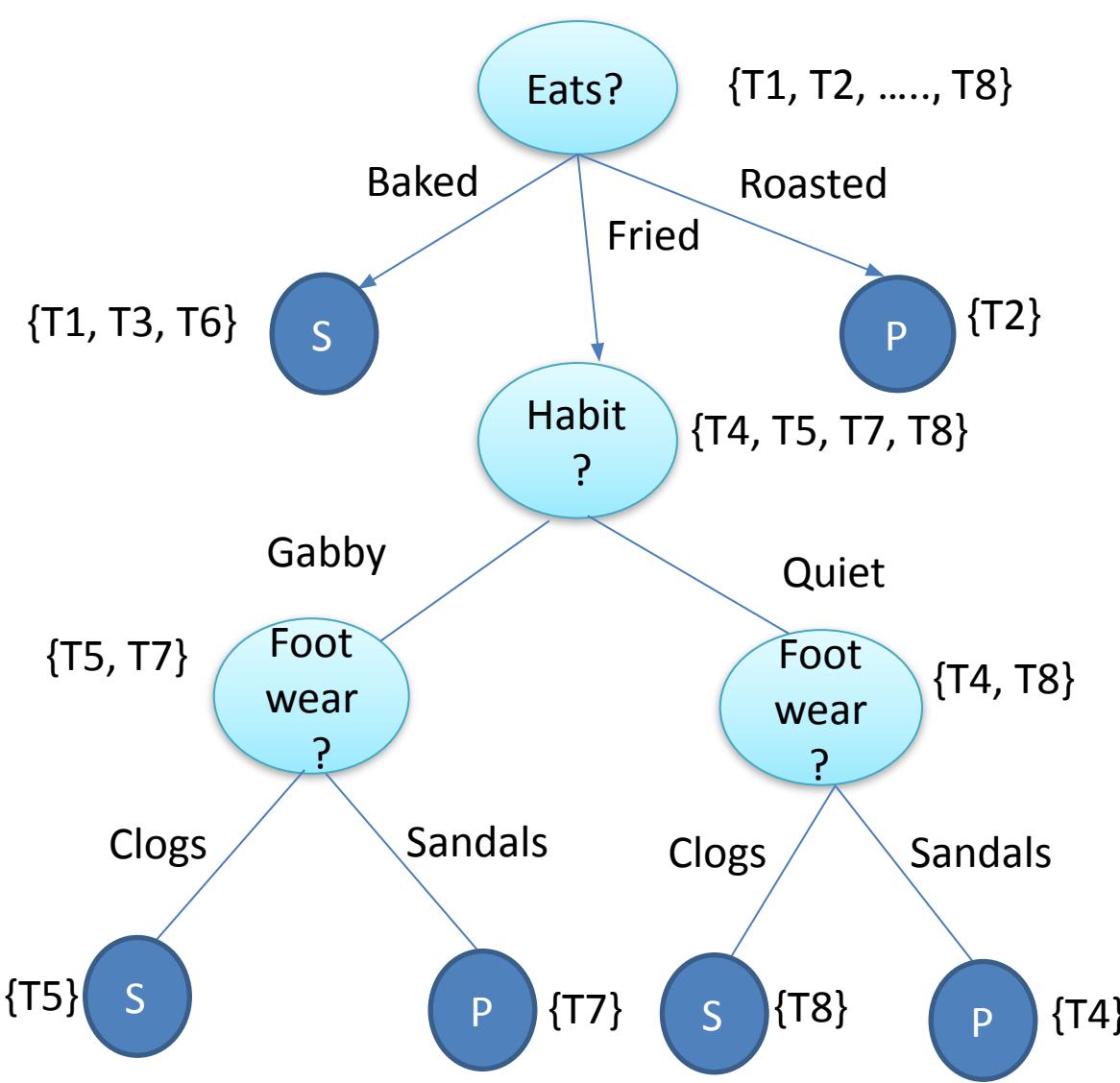


NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

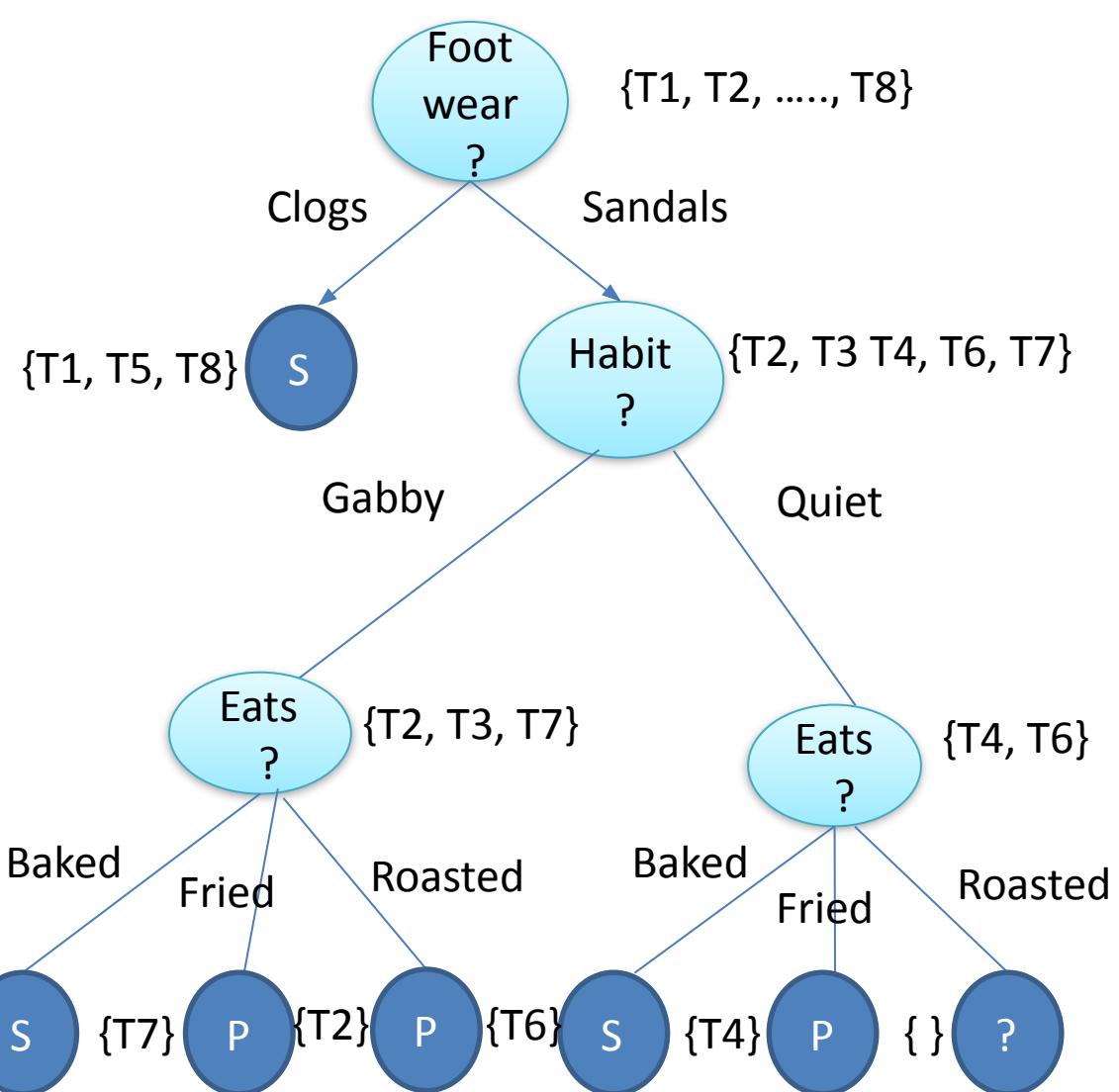
12-11-2022



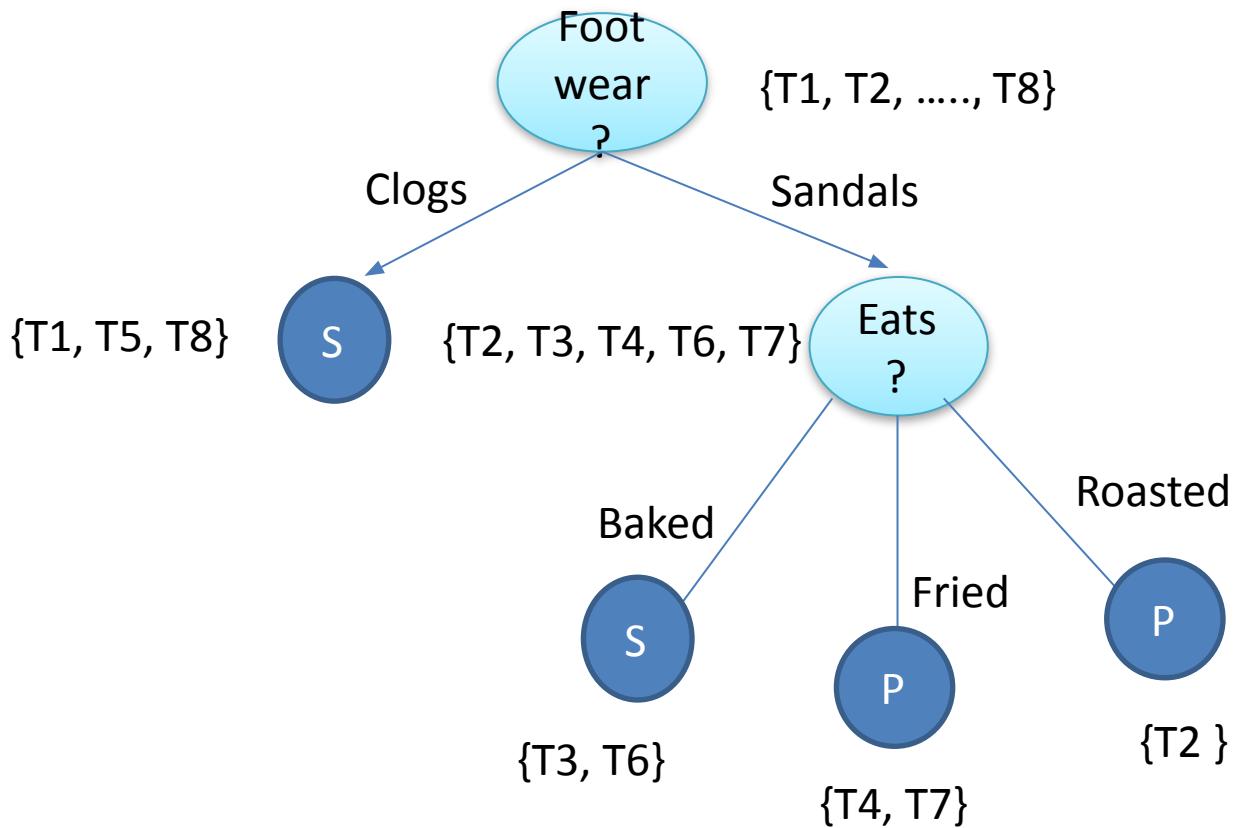
NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S



NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

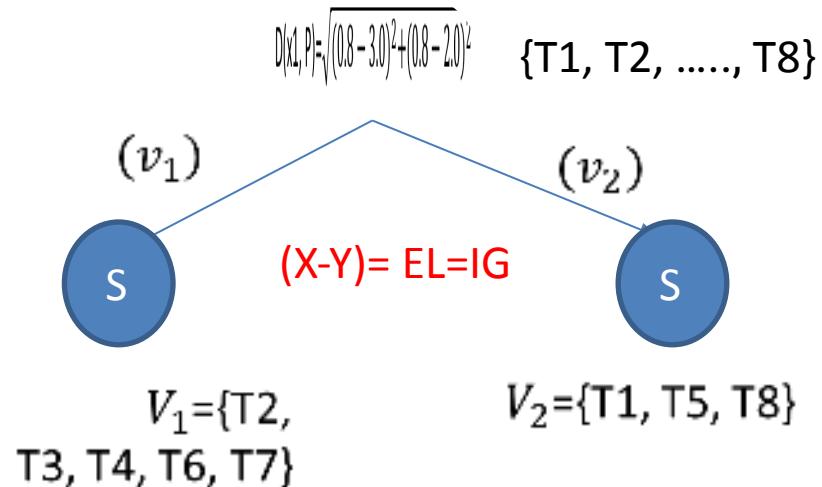


Classifying the Recall patterns

Recall Patterns	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor (some Rejected)
R3	Gabby	Roasted	Clogs	Student/ Professor
R4	Quiet	Roasted	Clogs	Student/ Professor

Ratio of Information Gain

$$\begin{aligned} & \frac{1}{3} \left[\max\left(3^{\text{th}} \text{ dist } B, \text{dist}(AB)\right) + \max\left(3^{\text{th}} \text{ dist } C, \text{dist}(AC)\right) + \right. \\ & \quad \left. \max\left(3^{\text{th}} \text{ dist } D, \text{dist}(AD)\right) \right] \end{aligned}$$



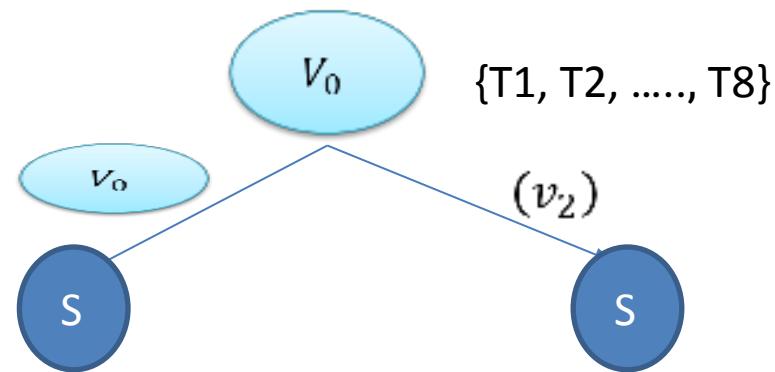
NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Ratio of Information Gain

- The weighted average information required to classify a pattern in set V_0 into one of the m classes is expressed as

$$I(V_0) = \sum_{k=1}^m \frac{Y(k, 0)}{Z(0)} (-\log \frac{Y(k, 0)}{Z(0)})$$

$I(V_0)$ is called the entropy of the set V_0 .



Density is reverse of distance therefore Local Reachability score LRD

$$\begin{aligned} LRD_A &= \frac{1}{RD_A} \\ &= 1/6.06 = 0.165 \end{aligned}$$

$V_2 = \{T1, T5, T8\}$

- Similarly for the set V_j ,

$$I(V_j) = \sum_{k=1}^m \frac{Y(k, j)}{Z(j)} (-\log \frac{Y(k, j)}{Z(j)})$$

- The weighted average information required to classify a pattern into one of class k in set V_0 after it has been split by the attribute A into sets V_1 to V_n is given by

$$I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} I(V_j)$$

- $I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right)$
 $I_A(V_0)$ is called the entropy of the attribute A for the set V_0
- The gain in information caused by attribute A splitting set V_0 into sets V_1 to V_n is

$$g_A(V_0) = I(V_0) - I_A(V_0)$$

Split information

$$\frac{Z(j)}{Z(0)}$$

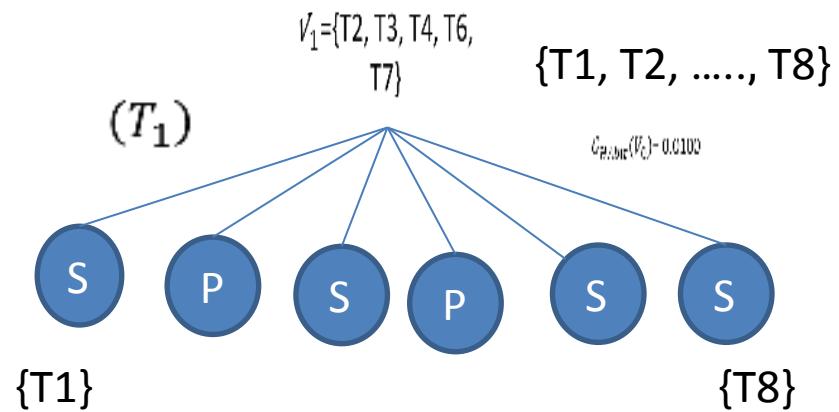
Information needed to extract set V_j from V_0 is

$$-\log \frac{Z(j)}{Z(0)}$$

- The weighted average information needed by attribute A to split set V_0 into sets V_1 to V_n is

$$S_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \left(-\log \frac{Z(j)}{Z(0)} \right)$$

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



- The ratio of information gain of the attribute A for set V_0 is defined as

$$G_A(V_0) = \frac{g_A(V_0)}{S_A(V_0)}$$

- The ratio of information gain is represented as

$$G_A(V_0) = \frac{I(V_0) - I_A(V_0)}{S_A(V_0)}$$

Example

Evaluate the entropy $I(V_0)$ of the Professor-student training set.

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTW EAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roast ed	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

Node no. 3,6,9,12,16 are unsuccessful path
The average unsuccessful path length is given by node 6 and 16 which is $\frac{5}{2}$. Since 2 trees are considered therefore $h(x)$ is the average path of the datapoint for the 2 trees.

$$\begin{aligned}
 s_1 &= 2^{-\left(\frac{3+4/2}{5}\right)} = 2^{-(0.7)} = 0.615 \\
 s_2 &= 2^{-\left(\frac{3+3/2}{5}\right)} = 2^{-(0.6)} = 0.65 \\
 s_3 &= 2^{-\left(\frac{5+5/2}{5}\right)} = 2^{-(1)} = 0.5 \\
 s_4 &= 2^{-\left(\frac{4+3/2}{5}\right)} = 2^{-(0.7)} = 0.615 \\
 s_5 &= 2^{-\left(\frac{1+2/2}{5}\right)} = 2^{-(0.3)} = 0.812 \\
 s_6 &= 2^{-\left(\frac{5+5/2}{5}\right)} = 2^{-(1)} = 0.5 \\
 s_7 &= 2^{-\left(\frac{3+3/2}{5}\right)} = 2^{-(0.6)} = 0.65 \\
 s_8 &= 2^{-\left(\frac{4+3/2}{5}\right)} = 2^{-(0.7)} = 0.615
 \end{aligned}$$

$$G_{Footwear}(V_0) = \frac{I(V_0) - I_{Footwear}(V_0)}{I(V_0)}$$

$$= (0.9544 - 0.6066)$$

$$= 0.3478$$

The ratio of information gain of HABIT

$$\begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.714322222 \end{pmatrix}$$

$$= \begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

$$\text{eigenvalues} = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

$$\alpha_2$$

FinalData = RowFeatureVector \times RowDataAdjust

The ratio of information gain of EATS

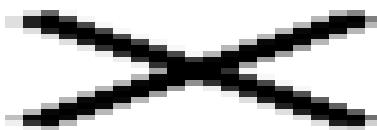
$$s_1 = \frac{l_1 + l_2 + l_3}{3} = x_1$$

$$s_1 = \frac{f_1 + f_2 + f_3}{3} = x_1$$

$$\begin{pmatrix} -.68 & -.74 \end{pmatrix}$$

- $V_1 = \{T_1, T_3, T_6\}$ at node y_1 , where EATS=baked.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T_4, T_5, T_7, T_8\}$ at node y_2 , where EATS=fried.
- $Y(1,2)=2$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=4$ (number of patterns in V_2)
- $V_3 = \{T_2\}$ at node y_3 , where EATS=roasted.
- $Y(1,3)=1$ (number of patterns in V_3 of class P)
- $Y(2,3)=0$ (number of patterns in V_3 of class S)
- $Z(3)=1$ (number of patterns in V_3)

The ratio of information gain of FOOTWEAR



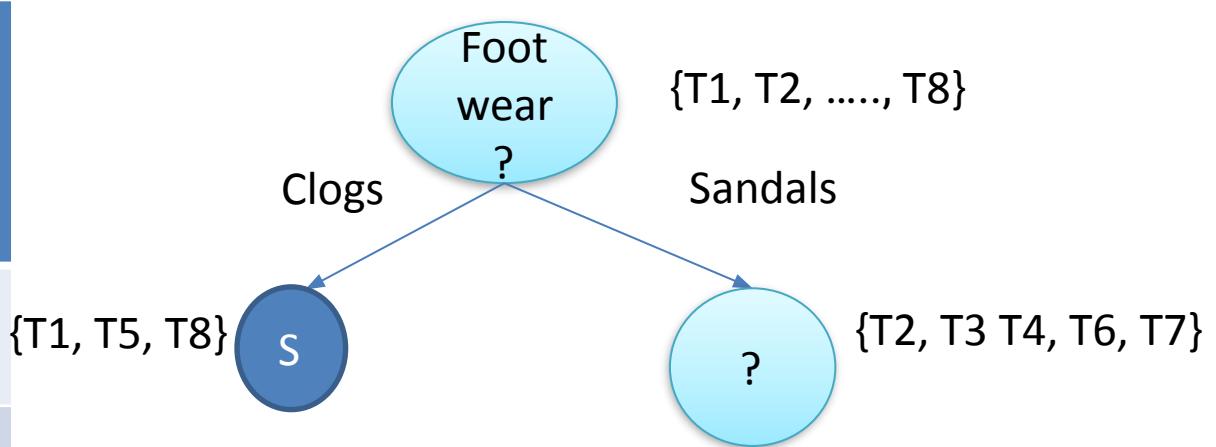
- $V_1 = \{T1, T5, T8\}$ at node y_1 , where FOOTWEAR=clogs.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=3$ (number of patterns in V_1 of class S)
- $Z(1)=3$ (number of patterns in V_1)
- $V_2 = \{T2, T3, T4, T6, T7\}$ at node y_2 , where FOOTWEAR=sandals.
- $Y(1,2)=3$ (number of patterns in V_2 of class P)
- $Y(2,2)=2$ (number of patterns in V_2 of class S)
- $Z(2)=5$ (number of patterns in V_2)

(.69 - 1.31 39 .09 1.29 49 .19 - .81 - .31 - .71)
(.49 - 1.21 .99 .29 1.09 .79 - .31 - .81 - .31 - 1.0)

$$S_{Footwear}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} (-\log \frac{Z(j)}{Z(0)}) \\ = 0.9544$$

$$G_{Footwear}(V_0) = \frac{I(V_0) - I_{Footwear}(V_0)}{S_{Footwear}(V_0)} \\ = 0.3640$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabb y	Bake d	Clogs	S
T2	Gabb y	Roas ted	Sanda ls	P
T3	Gabb y	Bake d	Sanda ls	S
T4	Quiet	Fried	Sanda ls	P
T5	Gabb y	Fried	Clogs	S
T6	Quiet	Bake d	Sanda ls	S
T7	Gabb y	Fried	Sanda ls	P
T8	Quiet	Fried	Clogs	S



$$\|x\| = \sqrt{2^2 + 4^2 + 0^2 + 0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} = 5.83$$

Pearson correlation coefficient: measures linear dependency between two random variables

$$\rho = \frac{\text{cov}(F1, F2)}{\sqrt{\text{var}(F1)\text{var}(F2)}}$$

$$\text{cov}(F1, F2) = \sum ((F1_i - F1') \cdot (F2_i - F2')) / (n - 1)$$

$$\text{var}(F1) = \sum (F1_i - F1')^2 / (n - 1), \text{ where } F1' = \frac{1}{n} \sum F1_i$$

$$\text{var}(F2) = \sum (F2_i - F2')^2 / (n - 1), \text{ where } F2' = \frac{1}{n} \sum F2_i$$

- Let C_1, C_2, \dots, C_m be the number of classes
- N denotes the number of training patterns of class C_k in the set V_t
- $Z(t)$ is the number of patterns in the set V_t
- $Z(t) = \sum_{k=1}^m Y(k, t)$
- The probability that a pattern in V_t belongs to class C_k is $\frac{Y(k, t)}{Z(t)}$
- The information required to classify a pattern in V_t into the class C_k for $1 \leq k \leq m$ is expressed as $-\log \frac{Y(k, t)}{Z(t)}$

The ratio of information gain of EATS at the right child of the root of a decision tree



- $V_1 = \{T3, T6\}$ at node y_1 , where EATS=baked.
- $Y(1,1)=0$ (number of patterns in V_1 of class P)
- $Y(2,1)=2$ (number of patterns in V_1 of class S)
- $Z(1)=2$ (number of patterns in V_1)

Fisher score

- Fisher score is one of the most widely used supervised feature selection methods.
 - It selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features.
 - The score of the i-th feature S_i will be calculated by Fisher Score,
- $$S_i = \frac{\sum n_j (\mu_{ij} - \mu_i)^2}{\sum n_j p_{ij}^2}$$
- where μ_{ij} and p_{ij} are the mean and the variance of the i-th feature in the j-th class, respectively.
 n_j is the number of instances in the j-th class and μ_i is the mean of the i-th feature.
- The features are ranked according to the Fisher Score.

$$\text{Slope} = \text{Rise/Run} = \frac{\Delta(Y)}{\Delta(X)}$$

$(X-X')$	$(Y-Y')$
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.0
$cov(X, Y)$	

$$5.539/9=0.6154444444$$

$$\text{Var}(X) =$$

$$5.549/9=0.6165555556$$

$$\text{Var}(Y) = \sum_{j=1}^n z(j) (-\log \frac{z(j)}{Z(0)})$$

$$6.4289/9=0.7143222222 \\ =1.5218$$

$$G_{Eats}(V_0) = \frac{I(V_0) - I_{Eats}(V_0)}{S_{Eats}(V_0)} \\ = 0.6377$$

The ratio of information gain of HABIT at the right child of the root of a decision tree

Step 6: Deriving the new data set

we simply take the transpose of the vector and multiply it on transpose
DataAdjust

Spearman's correlation coefficient: measures linear dependency between two random variables. It uses rank of each value. Data are represented as
 $\mathbf{x} = \mathbf{x}'^T$
 $\mathbf{y} = \mathbf{y}'^T$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

$$S(\mathbf{F1}, \mathbf{F2}) = \frac{\text{cov}(\mathbf{F1}, \mathbf{F2})}{\sqrt{\text{var}(\mathbf{F1}) \cdot \text{var}(\mathbf{F2})}}$$

$$\text{cov}(\mathbf{F1}, \mathbf{F2}) = \sum_i ((\mathbf{F1}_i'^T - \mathbf{F1}'^T) \cdot (\mathbf{F2}_i'^T - \mathbf{F2}'^T)) / (n-1)$$

$$\text{var}(\mathbf{F1}) = \sum_i (\mathbf{F1}_i'^T - \mathbf{F1}'^T)^2 / (n-1) \quad \text{where } \mathbf{F1}'^T = \frac{1}{n} \sum_i \mathbf{F1}_i'^T$$

$$\text{var}(\mathbf{F2}) = \sum_i (\mathbf{F2}_i'^T - \mathbf{F2}'^T)^2 / (n-1), \quad \text{where } \mathbf{F2}'^T = \frac{1}{n} \sum_i \mathbf{F2}_i'^T$$

It ranges between +1 and -1

Kendall's Tau: measures linear dependency between two random variables. It uses rank of each value. Kendall's Tau has smaller variability when using larger sample sizes. However, Spearman's measure is more computationally efficient, as Kendall's Tau is $O(n^2)$ and Spearman's correlation is $O(n \log(n))$.

Data are represented as

$$\mathbf{x} = \mathbf{x}'^T$$

$$\mathbf{y} = \mathbf{y}'^T$$

If the raw data are [0, -5, 4, 7], the ranked values will be [2, 1, 3, 4].

Therefore,

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z(0)} (-\log \frac{Y(k,0)}{Z(0)})$$

$$= 0.9710$$

$$I_{Habit}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} (-\log \frac{Y(k,j)}{Z(j)})$$

$$= 0.9507$$

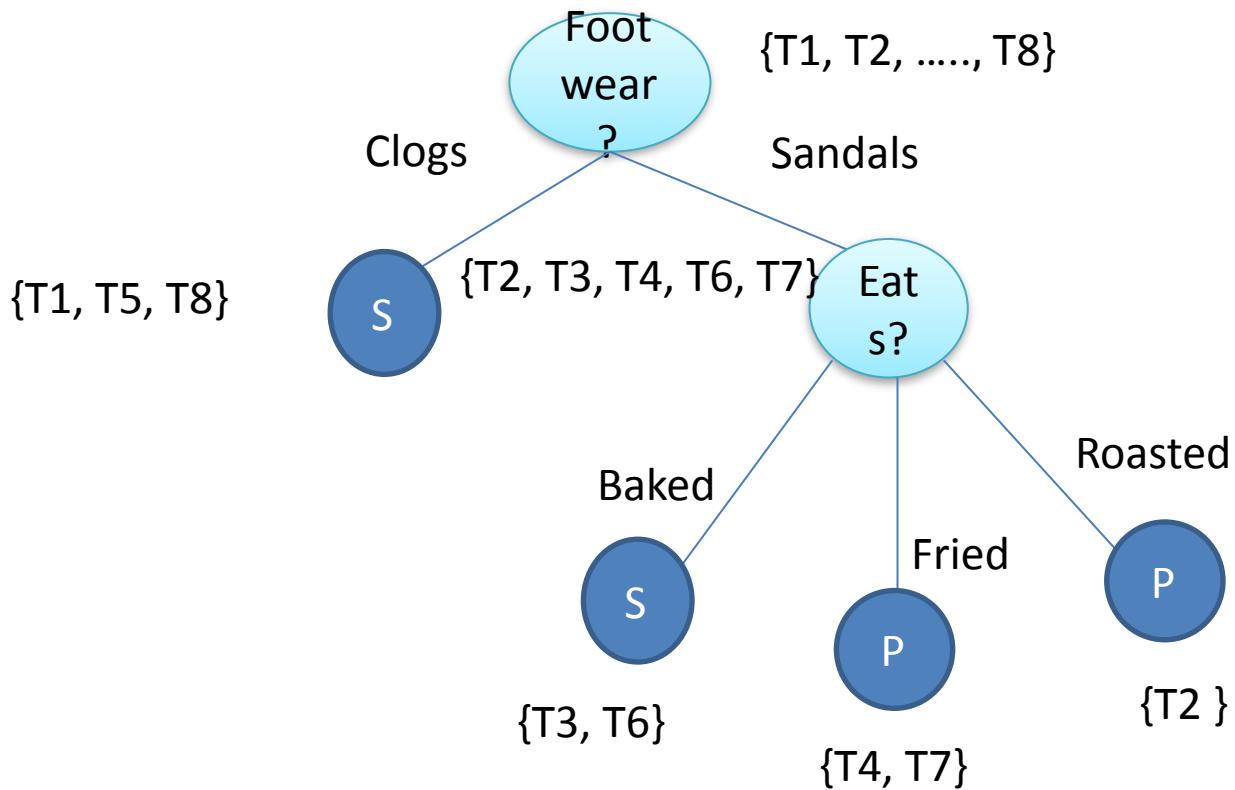
$$S_{Habit}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z(0)} (-\log \frac{Z(j)}{Z(0)})$$

$$= 0.9710$$

$$G_{Habit}(V_0) = \frac{I(V_0) - I_{Habit}(V_0)}{S_{Habit}(V_0)}$$

$$= 0.0208$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried	Clogs	S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S



— — — — —

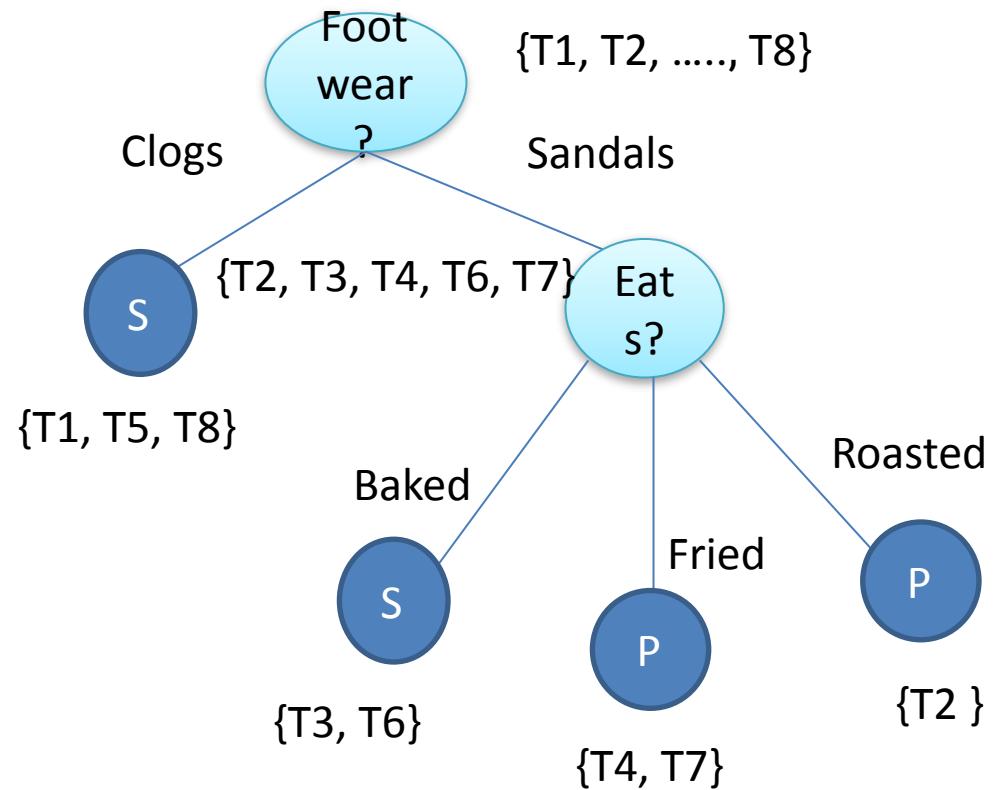
$$G_{Habit}(V_0) = 0.0208$$

If Footwear = Clogs
 Then pattern class= Student

If Footwear = Sandals
 and Eats = Baked,
 Then pattern class= Student

If Footwear = Sandals
 and Eats = Fried,
 Then pattern class= Professor

If Footwear = Sandals
 and Eats = Roasted,
 Then pattern class= Professor



Classifying the Recall patterns

- If Footwear = Clogs
Then pattern class= Student
- If Footwear = Sandals
and Eats = Baked,
Then pattern class= Student
- If Footwear = Sandals
and Eats = Fried,
Then pattern class= Professor
- If Footwear = Sandals
and Eats = Roasted,
Then pattern class= Professor

Recall	Attributes			Classification
	HABIT	EATS	FOOTWEAR	
R1	Quiet	Baked	Clogs	Student
R2	Quiet	Roasted	Sandals	Professor
R3	Gabby	Roasted	Clogs	Student
R4	Quiet	Roasted	Clogs	Student

Strength of DT

- It produces very simple understandable rules.
- Works well for most of the problem
- It can handle both numerical and categorical features
- It can work well for small and large training datasets
- DT shows which features are more useful for classification

Weakness of DT

DT is often biased towards features having more number of possible values

DT gets over-fitted and under-fitted easily.

DT is prone to errors with many classification and with small number of training examples.

DT is computationally expensive to train

Difficult to understand large DT.

Decision Tree with Missing Attributes

Values

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOTWEAR	
T1	Gabby	Baked	Clogs	Students
T2	Gabby	Roasted	Sandals	Professor
T3	Gabby	Baked	Sandals	Students
T4	Quiet	Fried	Sandals	Professor
T5	Gabby	Fried	----	Students
T6	Quiet	Baked	Sandals	Students
T7	Gabby	Fried	Sandals	Professor
T8	Quiet	Fried	Clogs	Students

Decision Tree with Missing Attributes Values

Let C_1, C_2, \dots, C_m be the number of classes where $m > 1$.

V_0 = a set at node y_0

We want to evaluate the ratio of information gain for attribute A at node y_0 but attribute A is having some missing values in V_0

A splits V_0 in to V_1, \dots, V_n as A is having n discrete values

V_{n+1} = is a set of patterns having missing values for A

For $1 \leq k \leq m$ and $0 \leq i \leq n$

$Y(k, i)$ be the number of training patterns of class C_k in the set V_i

For $0 \leq i \leq (n+1)$; $Z(i)$ is the number of patterns in the set V_i

Let ; $Z'(0) = Z(0) - Z(n+1)$

$Z'(0)$ = set of patterns for which the value of A is known

Decision Tree with Missing Attributes Values

The entropy of the set V_0

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z'(0)} \left(-\log \frac{Y(k,0)}{Z'(0)} \right)$$

The entropy of A for set V_0 .

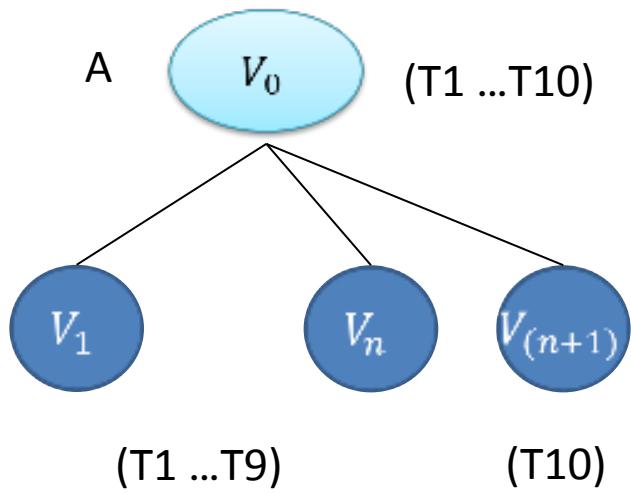
$$\bullet I_A(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z'(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right)$$

Patterns of V_{n+1} are not included in the above calculation because A of this set does not provide any information about their class.

Information gain due to A is calculated

$$g_A(V_0) = \frac{Z'(0)}{Z(0)} [I(V_0) - I_A(V_0)]$$

$\frac{Z'(0)}{Z(0)}$ = the probability of the value of attribute A is known in the set V_0 .



Decision Tree with Missing Attributes Values

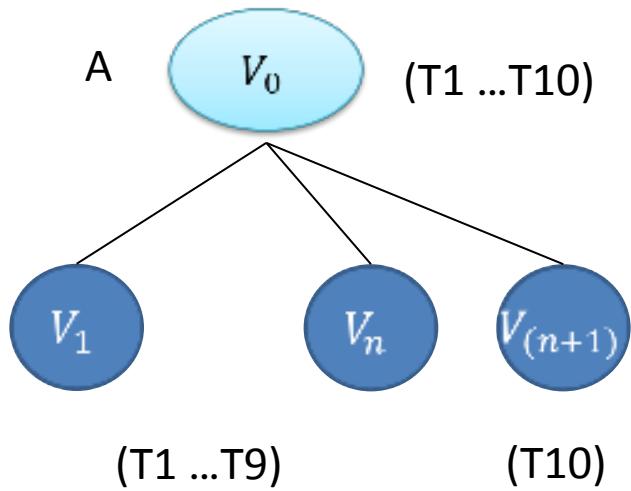
A has effectively split V_0
into $(n+1)$ sets

Therefore the split information of attribute A becomes

$$S_A(V_0) = \sum_{j=1}^{n+1} \frac{z(j)}{z(0)} \left(-\log \frac{z(j)}{z(0)} \right)$$

The ratio of information gain of attribute A for
set V_0 is as follows

$$G_A(V_0) = \frac{g_A(V_0)}{S_A(V_0)}$$



NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$G_{Habit}(V_0) = 0.0047$

$G_{Eats}(V_0) = 0.3233$

Ratio of information gain for Footwear

$V_0 = \{T1, T2, T3, \dots, T8\}$.

$Y(1,0) = 3$ (number of known Footwear-value patterns in patterns in V_0 of class P)

$Y(2,0) = 4$ (number of known Footwear-value patterns in V_0 of class S)

$Z(0) = 8$ (number of patterns in V_0)

$Z'(0) = 7$

$V_1 = \{T1, T8\}$ where Footwear= clogs

$Y(1,1) = 0$ (number of patterns in V_1 of class P)

$Y(2,1) = 2$ (number of patterns in V_1 of class S)

$Z(1) = 2$ (number of patterns in V_1)

$V_2 = \{T2, T3, T4, T6, T7\}$ where Footwear= sandals

$Y(1,2) = 3$ (number of patterns in V_2 of class P)

$Y(2,2) = 2$ (number of patterns in V_2 of class S)

$Z(2) = 5$ (number of patterns in V_2)

$V_3 = \{T5\}$ missing values of Footwear

$Z(3) = 1$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z'(0)} \left(-\log \frac{Y(k,0)}{Z(0)} \right) = 0.9852$$

$$I_{Footwear}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z'(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right) = 0.6936$$

$$g_{Footwear}(V_0) = \frac{Z'(0)}{Z(0)} [I(V_0) - I_A(V_0)] = 0.25515$$

$$S_{Footwear}(V_0) = \sum_{j=1}^{(n+1)} \frac{Z(j)}{Z(0)} \left(-\log \frac{Z(j)}{Z(0)} \right) = 1.3$$

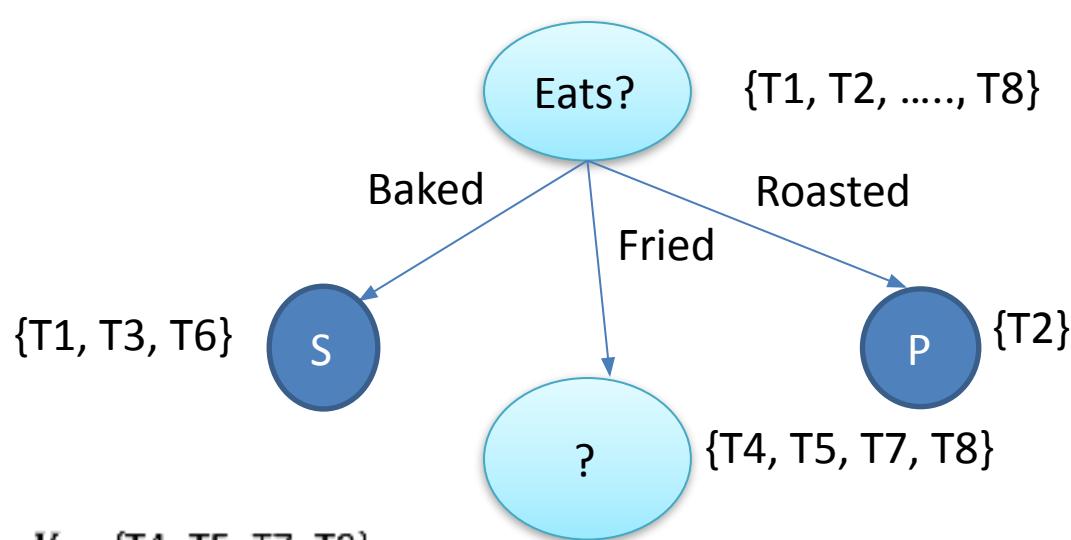
$$G_{Footwear}(V_0) = \frac{g_{Footwear}(V_0)}{S_{Footwear}(V_0)} = 0.1963$$

$$G_{Habit}(V_0) = 0.0047$$

$$G_{Eats}(V_0) = 0.3233$$

NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d		S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

12-11-2022



$$V_0 = \{T4, T5, T7, T8\}$$

$Y(1,0)=2$ (number of known Footwear-value patterns in patterns in V_0 of class P)

$Y(2,0)=1$ (number of known Footwear-value patterns in V_0 of class S)

$Z(0)=4$ (number of patterns in V_0)

$$Z'(0)=3$$

$V_1 = \{T8\}$ where Footwear= clogs

$Y(1,1)=0$ (number of patterns in V_1 of class P)

$Y(2,1)=1$ (number of patterns in V_1 of class S)

$Z(1)=1$ (number of patterns in V_1)

$V_2 = \{T4, T7\}$ where Footwear= sandals

$Y(1,2)=2$ (number of patterns in V_2 of class P)

$Y(2,2)=0$ (number of patterns in V_2 of class S)

$Z(2)=2$ (number of patterns in V_2)

$V_3 = \{T5\}$ missing values of Footwear

$$Z(3)=1$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z'(0)} \left(-\log \frac{Y(k,0)}{Z(0)} \right) = 0.9183$$

$$I_{Footwear}(V_0) = \sum_{j=1}^n \frac{Z(j)}{Z'(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right) = 0$$

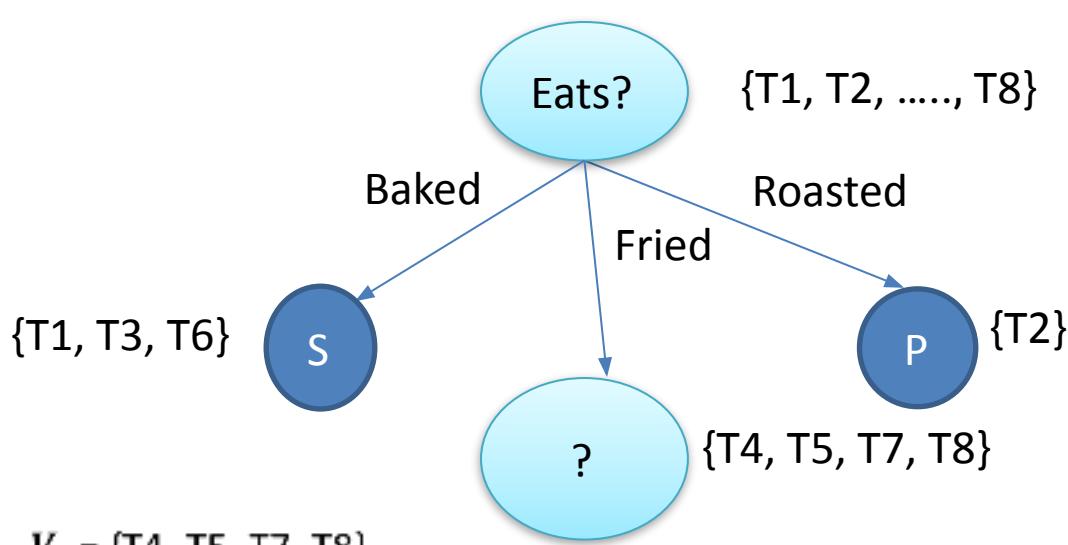
$$g_{Footwear}(V_0) = \frac{Z'(0)}{Z(0)} [I(V_0) - I_A(V_0)] = 0.9183$$

$$S_{Footwear}(V_0) = \sum_{j=1}^{(n+1)} \frac{Z(j)}{Z(0)} \left(-\log \frac{Z(j)}{Z(0)} \right) = 1.4466$$

$$G_{Footwear}(V_0) = \frac{g_{Footwear}(V_0)}{S_{Footwear}(V_0)} = 0.6351$$

NAME of trainin g patter n	Attributes			Class
	HABI TS	EAT	FOOT WEA R	
T1	Gabb y	Bak ed	Clogs	S
T2	Gabb y	Roa sted	Sand als	P
T3	Gabb y	Bak ed	Sand als	S
T4	Quiet	Frie d	Sand als	P
T5	Gabb y	Frie d	Clogs	S
T6	Quiet	Bak ed	Sand als	S
T7	Gabb y	Frie d	Sand als	P
T8	Quiet	Frie d	Clogs	S

12-11-2022



$$V_0 = \{T4, T5, T7, T8\}$$

$Y(1,0)=2$ (number of patterns in V_0 of class P)

$Y(2,0)=2$ (number of patterns in V_0 of class S)

$Z(0)=4$ (number of patterns in V_0)

$$V_1 = \{T5, T7\} \text{ where Habit= gabby}$$

$Y(1,1)=1$ (number of patterns in V_1 of class P)

$Y(2,1)=1$ (number of patterns in V_1 of class S)

$Z(1)=2$ (number of patterns in V_1)

$$V_2 = \{T4, T8\} \text{ where Habit= quiet}$$

$Y(1,2)=1$ (number of patterns in V_2 of class P)

$Y(2,2)=1$ (number of patterns in V_2 of class S)

$Z(2)=2$ (number of patterns in V_2)

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Bake d	Clogs	S
T2	Gabby	Roast ed	Sandal s	P
T3	Gabby	Bake d	Sandal s	S
T4	Quiet	Fried	Sandal s	P
T5	Gabby	Fried		S
T6	Quiet	Bake d	Sandal s	S
T7	Gabby	Fried	Sandal s	P
T8	Quiet	Fried	Clogs	S

$$I(V_0) = \sum_{k=1}^m \frac{Y(k,0)}{Z(0)} \left(-\log \frac{Y(k,0)}{Z(0)} \right) = 1.0$$

$$I_{Habit}(V_0) = \sum_{j=1}^n \frac{z(j)}{Z(0)} \sum_{k=1}^m \frac{Y(k,j)}{Z(j)} \left(-\log \frac{Y(k,j)}{Z(j)} \right) = 1.0$$

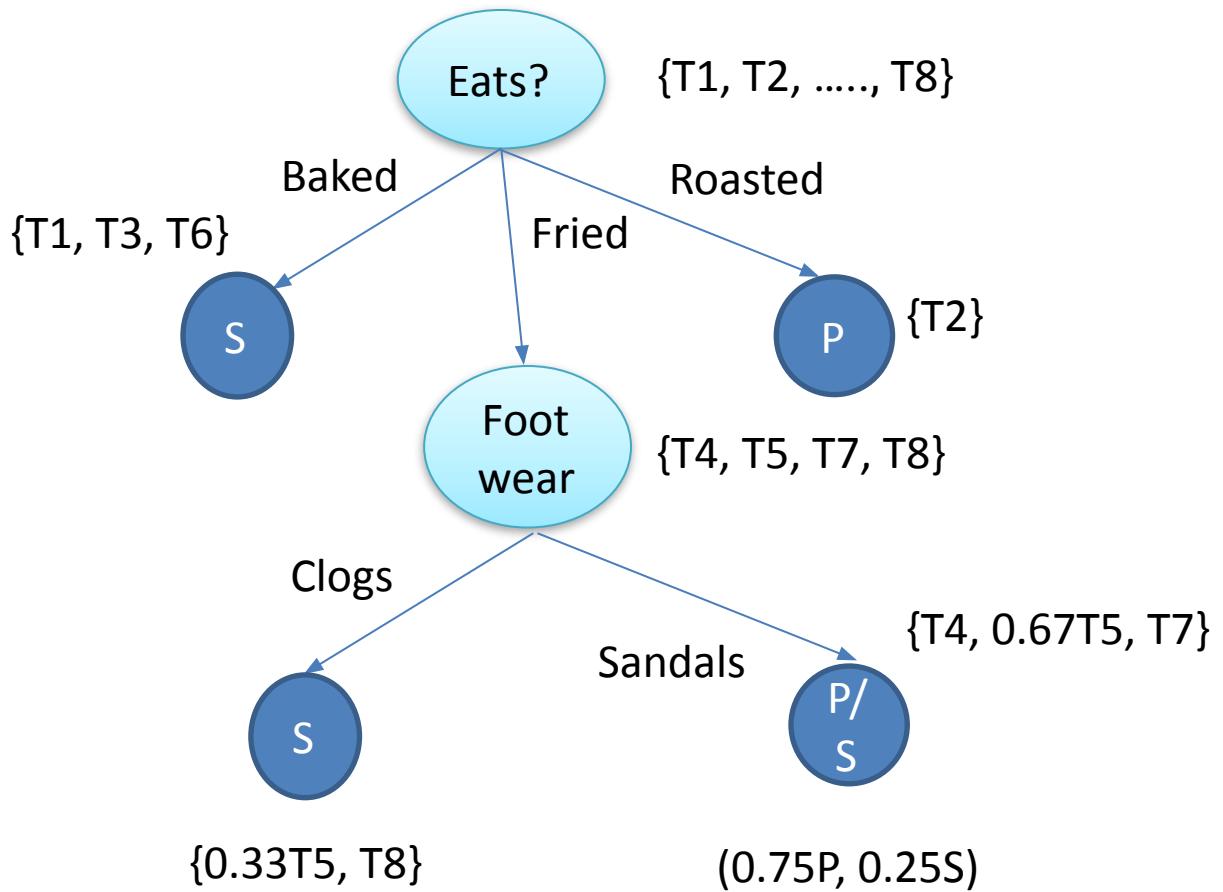
$$S_{Habit}(V_0) = \sum_{j=1}^{(n)} \frac{z(j)}{Z(0)} \left(-\log \frac{z(j)}{Z(0)} \right) = 1.0$$

$$G_{Habit}(V_0) = \frac{g_{Habit}(V_0)}{S_{Habit}(V_0)} = 0$$

NAME of training pattern	Attributes			Class
	HABIT S	EAT	FOOT WEAR	
T1	Gabby	Baked	Clogs	S
T2	Gabby	Roasted	Sandals	P
T3	Gabby	Baked	Sandals	S
T4	Quiet	Fried	Sandals	P
T5	Gabby	Fried		S
T6	Quiet	Baked	Sandals	S
T7	Gabby	Fried	Sandals	P
T8	Quiet	Fried	Clogs	S

$$G_{Footwear}(V_0) = \frac{g_{Footwear}(V_0)}{S_{Footwear}(V_0)} = 0.6351$$

$$V_1 = \{T_2, T_3, T_4, T_6, T_7\}$$



Decision Tree with Missing Attributes Values

If some patterns are there with missing values of attribute A. So, in practise, a fraction of each such pattern is added to the set associated as follows

1. For $1 \leq i \leq n$, let w_i be equal to the sum of the weights of the training patterns in set V_i
2. The sum of the weights is calculated as $\text{sum_of_weight} = \sum_{i=1}^n w_i$
3. For every training pattern T in V_0 whose value of attribute A is missing do step 3.1 and 3.2
 - 3.1 Let f_0 be the fraction of pattern T present in V_0
 - 3.2 For $j = 1, 2, \dots, n$ do step 3.2.1.
 - 3.2.1 Add to set V_j the f_j th fraction of pattern T where

$$f_j = \frac{w_j}{\text{sum_of_weight}} f_0$$

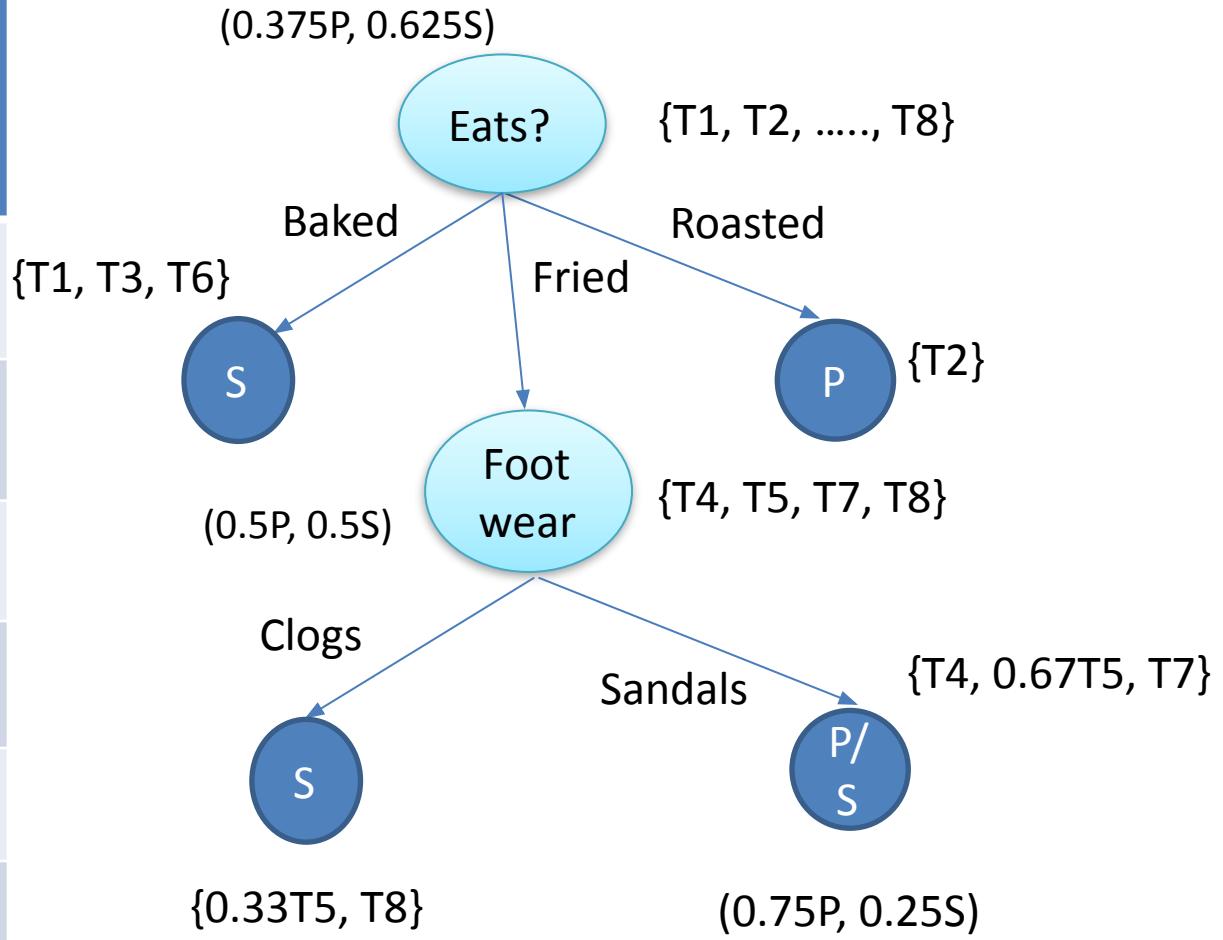
$$\text{T5 as clogs} = \frac{1}{3} * 1 = 0.33$$

$$\text{T5 as sandals} = \frac{2}{3} * 1 = 0.67$$

Out of 2.67, 2 is professor therefore professor = $2/2.67 = 0.75P$

Out of 2.67, 0.67 is student therefore student = $.67/2.67 = 0.25S$

NAME of training pattern	Attributes			Class
	HABITS	EAT	FOOT WEAR	
R1	Gabby	Fried	Clogs	S
R2	----	Fried	Sandals	0.75P, 0.25S
R3	-----	Fried	0.5P, 0.5S
R4		Baked		S
R5		Roasted		P
R6	Fried	Clogs	S
R7	Clogs	0.375P 0.625S
R8	Gabby	0.375P 0.625S



Decision Tree Algorithm

1. If $m = 1$ (that is, there is only one class), then create a single node x , label x with the name of the single class, and terminate the procedure. The decision tree consists of single node x , with its class label.
2. Initialize lists OPEN and CLOSED to empty.
3. Initialize subscript i to empty. Create a root node x_i , and associate the training set V_i with it.
4. Put node x_i in OPEN.
5. If OPEN is empty, return from the procedure. The nodes in CLOSED constitute the decision tree built. The subscript and the label of each node in CLOSED, together with the labelled arc from the node's parent, delineate the decision tree (step 10.1 explains how the arcs are labelled).
6. Remove the frontmost node x_i from OPEN. Create a candidate set of attributes for x_i , where the set contains all those attributes that have not been examined at any node on the path from the root to x_i . Select an attribute A from the candidate set for examining at node x_i (the criterion developed in Section 2.5 can be used to select attributes; for now, it may be selected arbitrarily).

Decision Tree Algorithm

7. If subscript i = empty (that is, x_i is the root), then put the following at the back of the list CLOSED: node x_i , together with the attribute A being examined at it.
8. If $i \neq$ empty (that is, x_i is not the root), then put the following at the back of the list CLOSED: node x_i , together with the attribute A being examined at it and the label of the arc from the parent-of- x_i to x_i .
9. Examine attribute A as follows. If A has n possible values v_1, v_2, \dots, v_n , then expand node x_i to generate its n children $x_{i1}, x_{i2}, \dots, x_{in}$.
10. For $j = 1, 2, \dots, n$ do steps 10.1 to 10.8 .
 - 10.1. Label the arc from node x_i to node x_{ij} with attribute value v_j .
 - 10.2. Associate with x_{ij} the set $V_{ij} \subseteq V_i$, such that the value of attribute A for the patterns in V_{ij} is v_j .
 - 10.3. If all the patterns in set V_{ij} belong to one class, then label node x_{ij} with the name of that class. Go to step 10.7.

Decision Tree Algorithm

10.4. If V_{ij} is empty, then label node x_{ij} with ‘?’ To indicate rejection , that is , failure to classify a given pattern. Go to step 10.7.(The question mark symbol, ‘?’ has been used to indicate rejection , which means that the question of pattern’s class still exists . You can replace it by a symbol of your choice .)

10.5. If the patterns in V_{ij} belong to more than one class and all the attributes have been examined on the path from the root to x_i , then label x_{ij} with ‘?’ To indicate rejection . (One could say that ideally we should have more attributes available to put the patterns of V_{ij} into separate classes , but more attributes may not be available. An alternative to rejection is to label node x_{ij} with the probability of the occurrence of different classes in V_{ij} ; more on this is given in Chapter 3.) Go to step 10.7.

10.6. Put the following at the back of the list OPEN: the node x_{ij} (it is a non-leaf node), and the label of the arc from node x_i to x_{ij} . Go to step 10.8.

10.7. Put the following at the back of the list CLOSED: node x_{ij} , a mark to indicate that x_{ij} is a leaf node, the label of x_{ij} , and the label of arc from x_i to x_{ij} .

10.8. Continue.

11. Go to step 5.

Decision Tree Pruning

To prevent a DT getting overfitted to the training data, pruning of decision tree is essential.

Pruning a decision tree reduces the size of the tree such that the model is more generalized.

1. Pre-pruning: Stop growing the tree before it reaches perfection.

This avoids over-fitting as well as optimizes computational cost.

It stands a chance to ignore important information contributed by a feature which was missed.

2. Post-Pruning: Allow the tree to grow entirely and then post-prune some of the branches from it.

By using certain pruning criterion, error rate, the size of the tree is reduced.

This is better for classification accuracy

Computational cost is more than pre-pruning

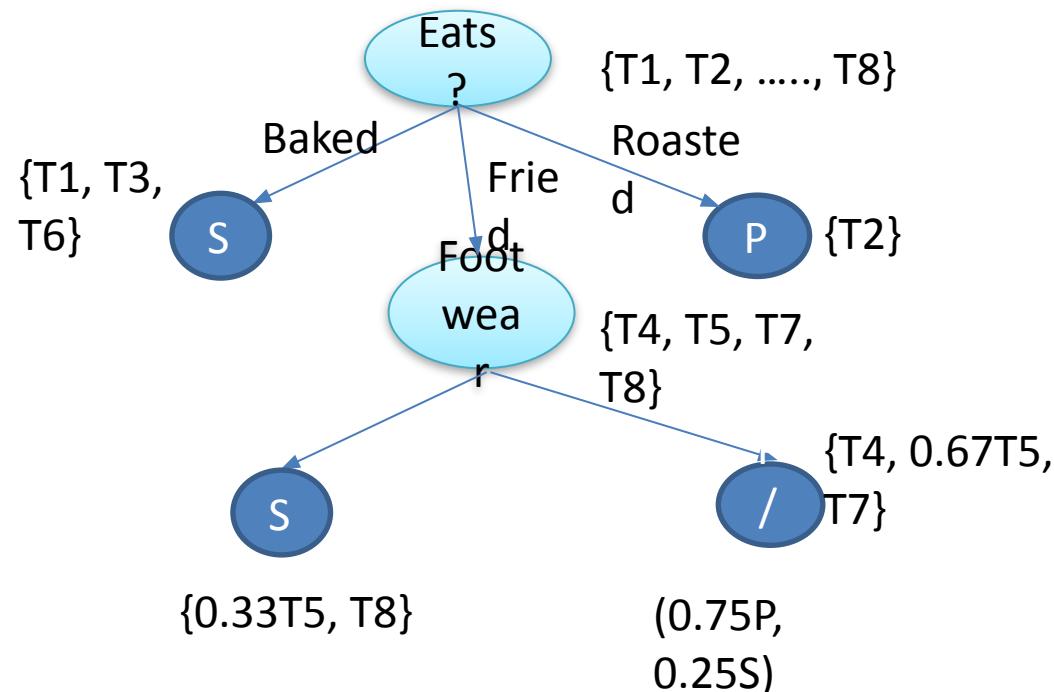
Error rate

Suppose class C is the most frequent in set V' associated with a node. A pattern at that node will be classified into C with error rate.

$$e(V') = \frac{|V'| - \text{no of patterns of class } C \text{ in } V'}{|V'|}$$

If $|V'| = 90$ and the no of patterns of class C in V' is 72

$$\text{Error rate} = (90-72)/90 = 0.2$$



Strength of DT

- It produces very simple understandable rules.
- Works well for most of the problem
- It can handle both numerical and categorical features
- It can work well for small and large training datasets
- DT shows which features are more useful for classification

Weakness of DT

DT is often biased towards features having more number of possible values

DT gets over-fitted and under-fitted easily.

DT is prone to errors with many classification and with small number of training examples.

DT is computationally expensive to train

Difficult to understand large DT.

**THANK
YOU
!!!!**