Data Science & Big Data Analytics

Subject Code: 310251

T. E. Computer (2019 Pattern)

Dr. S.R.Khonde

UNIT IV

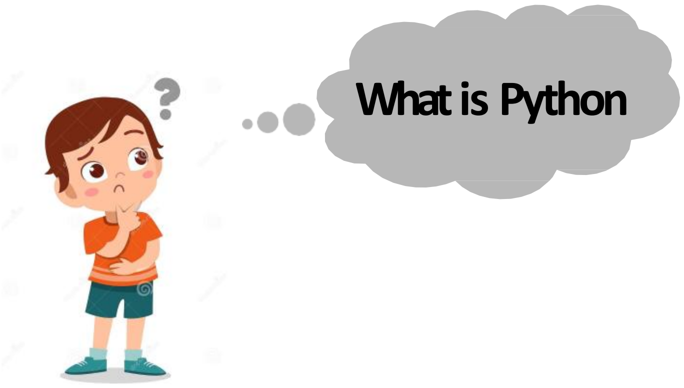
Unit IV	Predictive Big Data Analytics with Python	07 Hours
Introduction, Essential Python Libraries, Basic examples. **Data Preprocessing**: Removing Duplicates, Transformation of Data using function or mapping, replacing values, Handling Missing Data. Analytics Types: Predictive, Descriptive and Prescriptive. **Association Rules**: Apriori Algorithm, FP growth. **Regression**: Linear Regression, Logistic Regression. **Classification**: Naïve Bayes, Decision Trees. **Introduction to Scikit-learn**, Installations, Dataset, mat plotlib, filling missing values, Regression and Classification using Scikit-learn.		
#Exemplar/Case Studies	Use IRIS dataset from Scikit and apply data preprocessing methods	
***Mapping of Course Outcomes for Unit IV**	CO4,CO2	

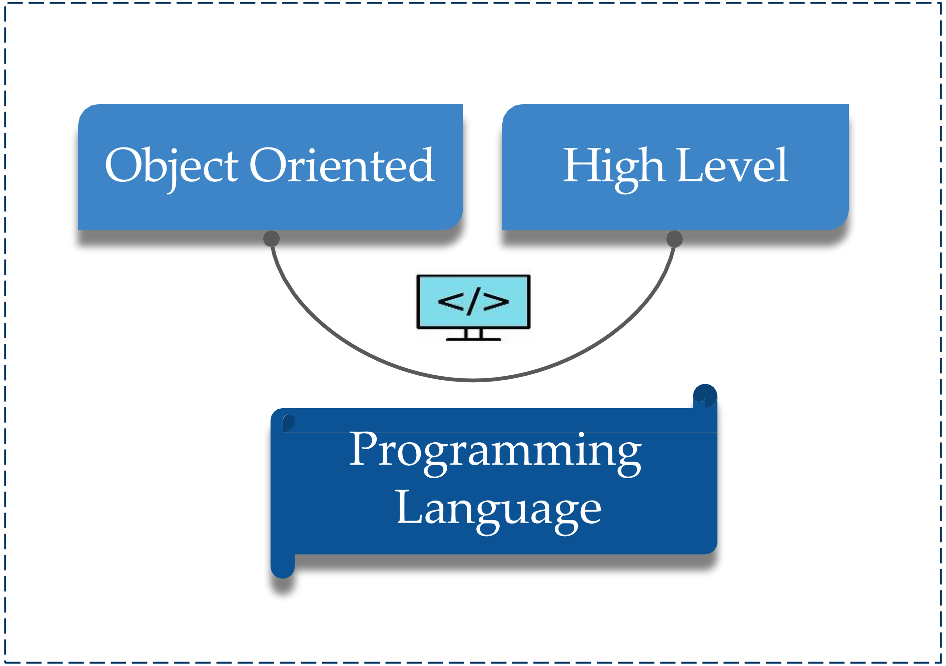
Outline

- Introduction
- Data Preprocessing
- Association Rules
- Regression
- Classification
- Introduction to Scikit-learn

Introduction

Python





Introduction

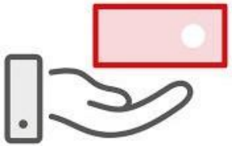
Features of Python

1. High Level → Integrated → Interactive → Object-Oriented → Scripting Language

2. Simple & Easy to learn

Features of Python

3 **Portable**



4 **Free |Open Source**

Features of Python

5 Perform complex tasks using a few lines of code.

6 Run equally on different platforms such as Windows, Linux, Unix,Macintosh, etc

7 Provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting.

Advantages of Python

- Ease of programming
- Minimizes the time to develop and maintain code
- Modular and object-oriented
- Large community of users
- A large standard and user-contributed library

DisAdvantages of Python

- Interpreted and therefore slower than compiled languages
- Decentralized with packages

Essential Python Libraries

- A library is a collection of files (called modules) that contains functions for other programs.
- A Python library is a reusable chunk of code that you may to include in your programs.

Introduction

Essential Python Libraries

01 NumPy

02 Pandas

03 SciPy

04 SciKit-Learn

Essential Python Libraries

01

NumPy

- Numpy (Numerical Python) is a perfect tool for scientific computing and performing basic and advanced array operations.
- The library offers many handy features performing operations on n-arrays and matrices in Python.
- It helps to process arrays that store values of the same data type and makes performing math operations on arrays easier.

Essential Python Libraries

02

Pandas

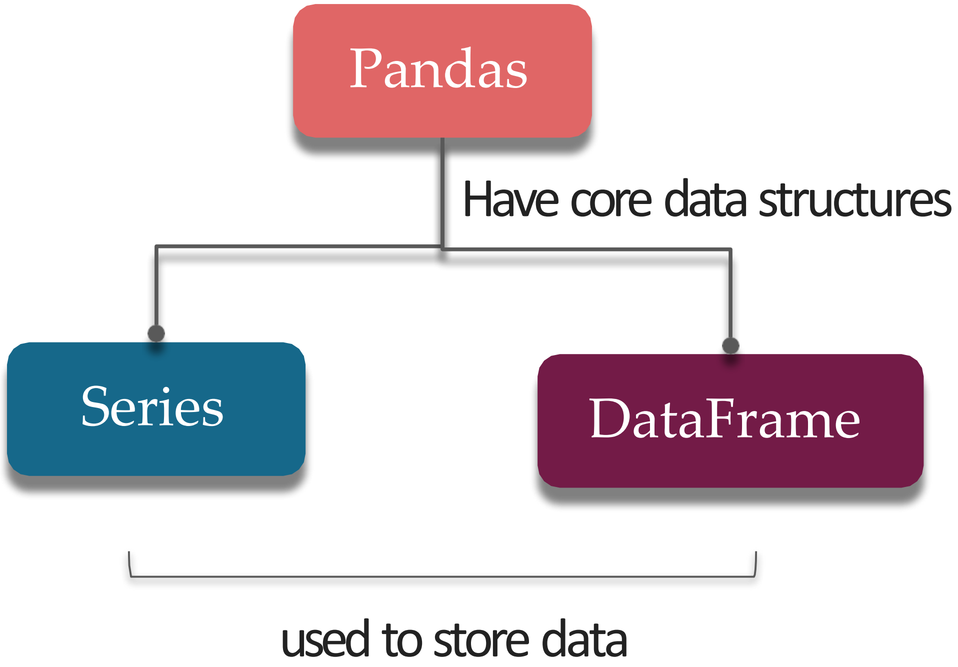
- It is one of the most popular Python libraries in data science.
- It provides support for data structures and data analysis tools.
- The library is optimized to perform data science tasks especially fast and efficiently.
- Pandas is best suited for structured, labelled data, in other words, tabular data,that has headings associated with each column of data.

Introduction

Essential Python Libraries

02

Pandas



Essential Python Libraries

02

Pandas

Series

- The series is a one-dimensional array-like structure
- designed to hold a single array (or 'column') of data and an associated array of data labels called an index.

Introduction

Essential Python Libraries

02

Pandas

DataFrame

- The DataFrame represents tabular data, a bit like a spreadsheet.
- DataFrames are organised into columns.
- each column can store a single data-type, such as floating point numbers, strings, boolean values etc.
- DataFrames can be indexed by either their row or column names.

Essential Python Libraries

03

SciPy

- SciPy contains many different packages and modules to assist in mathematics and scientific computing.
- It's difficult to state a single use case for SciPy considering that it contains so many different useful packages

Essential Python Libraries

03

SciPy

Some of the important packages include:

Matplotlib

- A 2D plotting library that can be used in Python scripts, the Python and IPython shell, web application servers, and more.

Essential Python Libraries

03

SciPy

Some of the important packages include:

IPython

- An interactive console that runs your code like the Python shell, but gives you even more features, like support for data visualizations.

Essential Python Libraries

04

SciKit-Learn

- Scikit-learn is probably the most useful library for machine learning in Python.

Introduction

Essential Python Libraries

04

SciKit-Learn

this library contains a lot of efficient tools

for

machine learning & statistical modeling

Including

- dimensionality reduction
- Classification
- Clustering
- Regression

Essential Python Libraries

04

SciKit-Learn

- Scikit-learn comes loaded with a lot of features

1. Supervised learning algorithms :

   - ❑ Think of any supervised learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn.

Essential Python Libraries

04

SciKit-Learn

- Scikit-learn comes loaded with a lot of features

2. Cross-validation :

   - There are various methods to check the accuracy of supervised models on unseen data.

Essential Python Libraries

04

SciKit-Learn

- Scikit-learn comes loaded with a lot of features

3. Unsupervised learning algorithms :

   - ❑ there is a large spread of algorithms in the offering - starting from clustering, factor analysis, principal component analysis to unsupervised neural networks.

Essential Python Libraries

04

SciKit-Learn

- Scikit-learn comes loaded with a lot of features

4. Various toy datasets:

   - This came in handy while learning scikit-learn.
   - For example : IRIS dataset, Boston House prices dataset.

Essential Python Libraries

04

SciKit-Learn

- Scikit-learn comes loaded with a lot of features

5. Feature extraction :

   - ❑ Useful for extracting features from images and text (e.g. Bag of words).

Data Preprocessing

- Data preprocessing is a data mining technique that involves transforming raw data into an understandable format.

- Aim to reduce the data size, find the relation between data and normalized them.

Why Data Preprocessing

- Data which capture from various sources is not pure.
- It contains some noise.
- It is called dirty data or incomplete data.
- In this data, there is lacking attribute values, interest, or containing only aggregate data. For example : occupation=” “
- Noisy data which contains errors or outliers. For eg. Salary=”-10”.

Why Data Preprocessing

- Inconsistent data which contains discrepancies in codes or names . for example- Age=”51” Birthday =”03/09/1998”.
- Incomplete , Noisy , and inconsistent data are common place properties of large real world databases and data warehouses.
- Incomplete data can occur for a variety of reasons

Steps during pre-processing

1 Data Cleaning

- Data is cleansed through process such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

Steps during pre-processing

2 Data Integration

- Data with different representations are put together and conflicts within the data are resolved

Steps during pre-processing

3 **Data Transformation**

- Data is transformed into the structure required. It is normalized, aggregated and generalized if required.

Steps during pre-processing

4 Data Reduction

- Data is normalized, aggregated and generalized.

Steps during pre-processing

5 **Data Discretization**

- Involves the reduction of number of values of a continuous attribute by dividing the range of attributes intervals.

Removing Duplicates

- Removing Duplicates in the context of data quality is where an organisation looks to identify and then remove instances where there is more than one record of a single person.

Removing Duplicates

- With large scales of data, this will often be done using tools that find and merge duplicate records in an existing database and prevent new ones from entering it based on similarities in specific fields.

Removing Duplicates

- Preparing a dataset before designing a machine learning model is an important task for the data scientist.
- If there are more duplicates then making machine learning model is useless or not so accurate. Therefore, you must know to remove the duplicates from the dataset.

Removing Duplicates

1 Handling missing data values

- Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

Removing Duplicates

1 Handling missing data values

- The various methods for handling the problem of missing values in data tuples are as follows:

Ignoring the tuple

- This is usually done when the class label is missing.

Removing Duplicates

1 Handling missing data values

Manually filling in the missing value

- This approach is time-consuming and may not be a reasonable task for large data sets with many missing values, especially when the value to be filled in is not easily determined.

Removing Duplicates

1 Handling missing data values

Using a global constant to fill in the missing value

- Replace all missing attribute values by the same constant.
- Using a measure of central tendency for the attribute, such as the mean, the median, the mode
- Using the attribute mean for numeric values or attribute mode nominal values, for all samples belonging to the same class as the given tuple.

Removing Duplicates

2 Transformation of data using function or mapping

- Data transformation is the process of converting data from one format or structure into another format or structure.
- Data transformation is critical to activities such as data integration and data management.

Removing Duplicates

2 Transformation of data using function or mapping

Common reasons to transform data:

- ❑ Moving data to a new data store
- ❑ Users want to join unstructured data or streaming data with structured data so user can analyze the data together

Removing Duplicates

2 Transformation of data using function or mapping

Common reasons to transform data:

- Users want to add information to data to enrich it, such as performing lookups.

  Adding geological data, or adding timestamps.
- Users want to perform aggregations, such as comparing sales data from different regions or totalling sales from different regions

Removing Duplicates

2

Transformation of data using function or mapping

Different ways to transform data:

Scripting

- SQL or Python to write the code to extract & transform the data.

Removing Duplicates

2 **Transformation of data using function or mapping**

Different ways to transform data:

On-premise ETL tools

- ETL (Extract, Transform, Load) tools can take much of the pain out of scripting the transformations by automating the process
- These tools are typically hosted on your company's site, and may require extensive expertise & infrastructure cost

Removing Duplicates

2 Transformation of data using function or mapping

Different ways to transform data:

Cloud-based ETL tools

- These ETL tools are hosted in the cloud
- Where u can leverage the expertise and infrastructure of the vendor

Analytics Types

Business analytics is the process of making sense of gathered data

Measuring business performance and producing valuable conclusions

that can help

companies make informed decisions on the future of the business,

through the

use of various statistical methods and techniques.

Analytics Types

- Business Analytics (BA) is the iterative, methodical exploration of an organization's data, with an emphasis on statistical analysis.
- Business analytics is used by companies that are committed to making data-driven decisions.
- Business analytics combines the fields of management, business and computer science.

Analytics Types

- The analytical part requires an understanding of data, statistics and computer science.
- Business analytics utilizes big data, statistical analysis and data visualization to implement organization changes.

Analytics Types

Data-driven decision-making process uses the following steps:

1. Identify the problem or opportunity for value creation

2. Identify primary as well secondary data sources.

3. Pre-process the data for issues such as missing and incorrect data. Generate derived variables and transform the data if necessary. Prepare the data for analytics model building.

Analytics Types

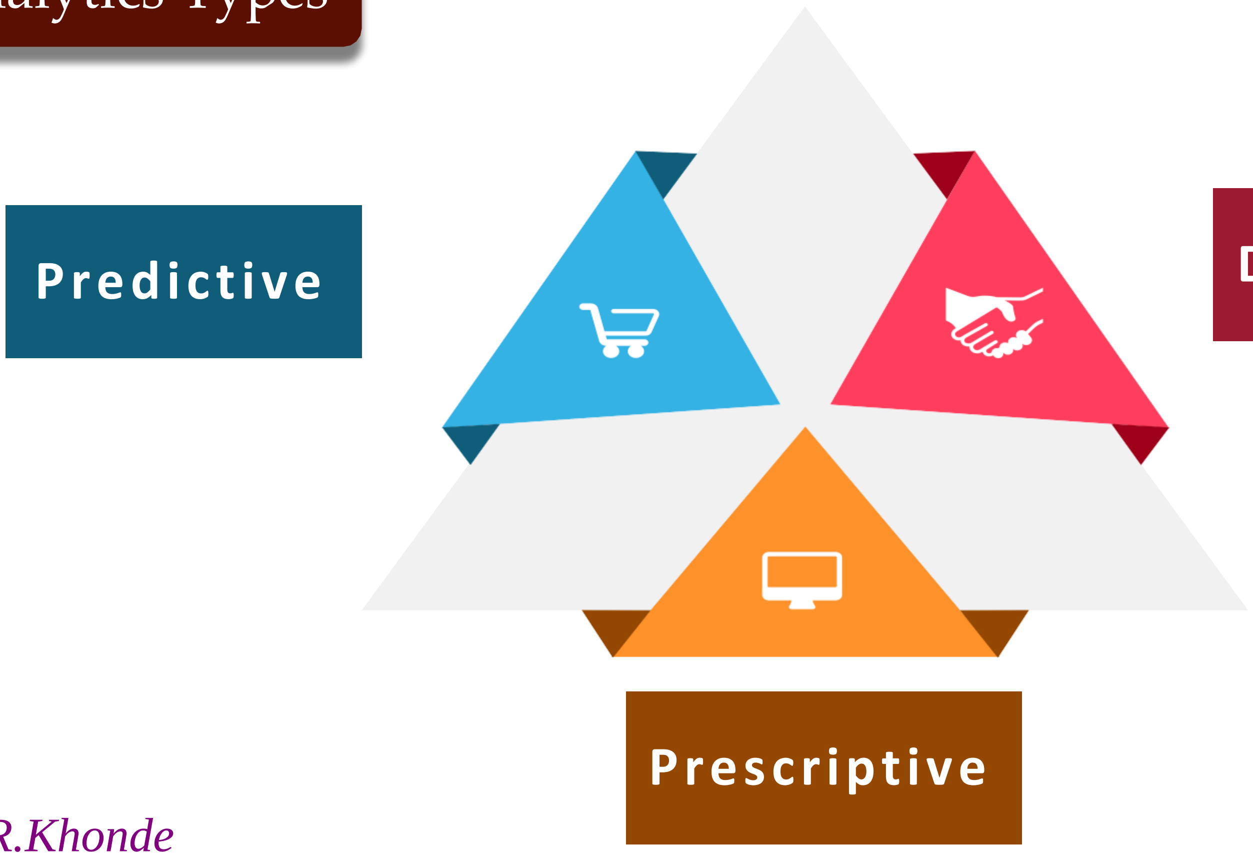
Data-driven decision-making process uses the following steps:

4. Divide the data sets into subsets training and validation data sets.

5. Build analytical models and identify the best model(s) using model performance in validation data.

6. Implement solution / Decision / Develop product.

Analytics Types



Predictive

Predictive analytics tells you what could happen in the future.

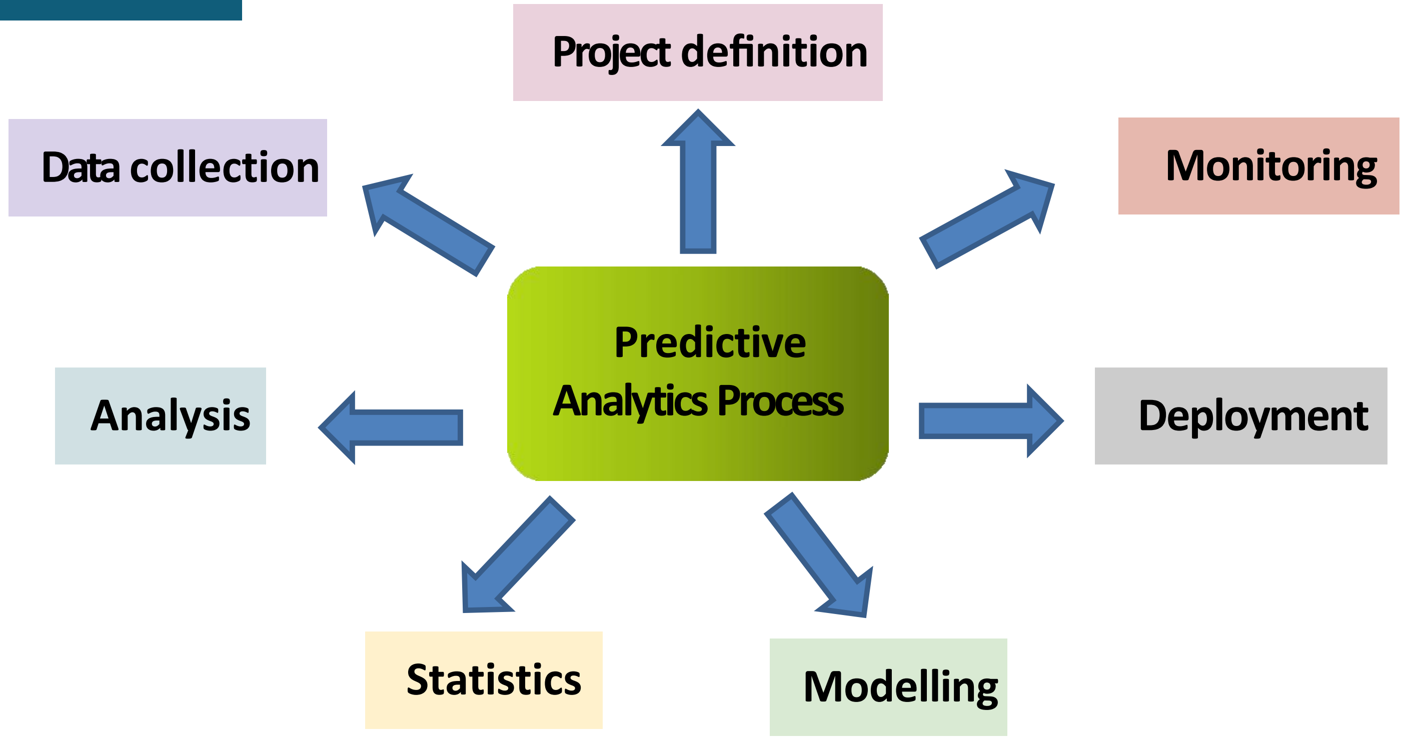
- Predictive analytics helps your organization predict with confidence what will happen next so that you can make smarter decisions and improve business outcomes.
- The purpose of the predictive model is finding the likelihood different samples will perform in a specific way.

Predictive

Predictive analytics tells you what could happen in the future.

- The predictive model typically calculates live transactions multiple times to help evaluate the benefit of a customer transaction.
- Predictive models typically utilize a variety of variable data to make the prediction.
- The variability of the component data will have a relationship with what it is likely to predict.

Predictive


Project definition
Data collection
Monitoring
Predictive Analytics Process
Analysis
Deployment
Statistics
Modelling

Predictive

Project definition

- Identify what shall be the outcome of the project, the deliverables, business objectives and based on that go towards gathering those data sets that are to be used.

Predictive

Data collection

- This is more of the big basket where all data from various sources are binned for usage.
- This gives a picture about the various customer interactions as a single view item

Predictive

Analysis

- the data is inspected, cleansed, transformed and modelled to discover if it really provides useful information and arriving at conclusion ultimately

Data Preprocessing

Predictive

Statistics

- This enables to validate if the findings, assumptions and hypothesis are fine to go ahead with and test them using statistical model.

Predictive

Modelling

- Through this accurate predictive models about the future can be provided.
- From the options available the best option could be chosen as the required solution with multi model evaluation.

Predictive

Deployment

- Through the predictive model deployment an option is created to deploy the analytics results into everyday effective decision.
- This way the results, reports and other metrics can be taken based on modelling.

Predictive

Monitoring

- Models are monitored to control and check for performance conformance to ensure that the desired results are obtained as expected.

Example of Predictive Analytics

Social Media Analysis

Weather

Retail

Health care

Fraud detection

Descriptive

- It is simple method and used in first phase of analytics, involves gathering, organizing tabulating and depicting data then the characteristics of what we are studying

Descriptive

- The descriptive model shows relationships between the product/service with the acquired data.
- This model can be used to organize a customer by their personal preferences.

Descriptive

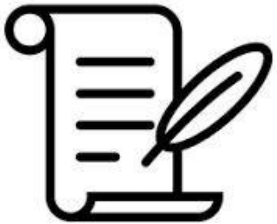
- Descriptive statistics are useful to show things like, total stock in inventory, average dollars spent per customer and year over year change in sales.

- While business intelligence tries to make sense of all the data that's collected each and every day by organizations of all types, communicating the data in a way that people can easily grasp often becomes an issue.

Data Preprocessing

Example of Descriptive Analytics

Reports that provides Historical Insights



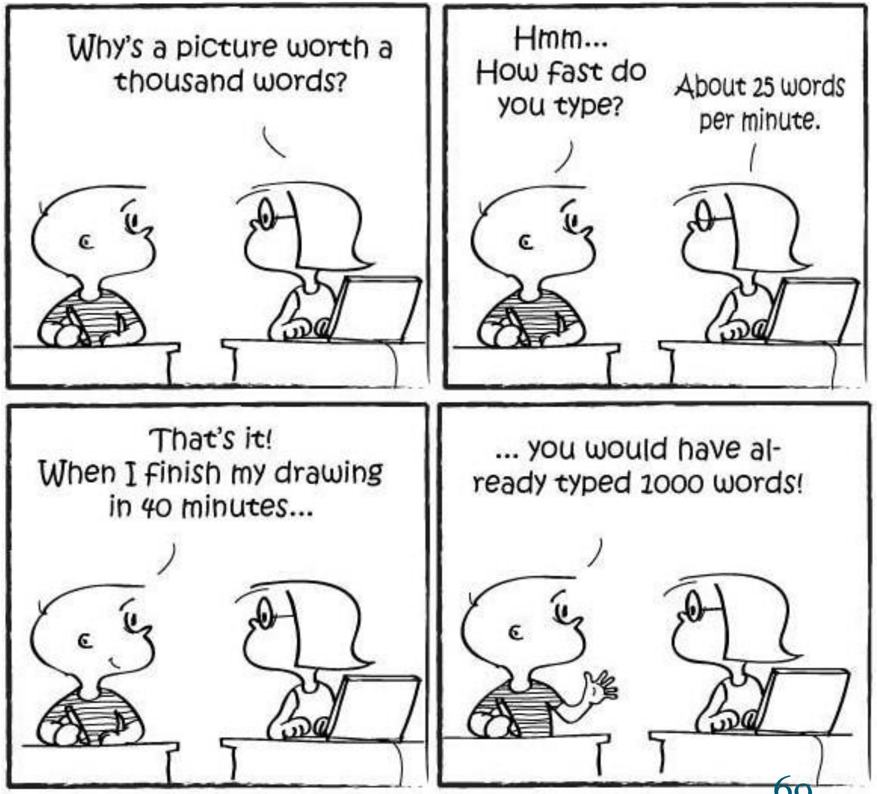
Regarding the company' s

- Production
- Financial
- Operations
- Sales
- Inventory
- Production

Descriptive

- Data visualization evolved because data displayed graphically allows for an easier comprehension of the information, validating the old adage,

- "a picture is worth a thousand words."



Descriptive

- In business, proper data visualization provides a different approach to show potential connections, relationships, etc.
- which are not as obvious in data that's non-visual.
- A business intelligence dashboard is an information management tool that is used to track KPIs, metrics and other key data points relevant to a business, department or specific process.

Prescriptive

- This model suggests a course of action.
- Prescriptive analytics assists users in finding the optimal solution to a problem or in making the right choice/decision among several alternatives.
- The prescriptive model utilizes an understanding of what has happened, why it has happened and a variety of "what-might-happen" analysis to help the user determine the best course of action to take.

Prescriptive

Example of Prescriptive Analytics

Traffic Applications

Product Optimization

Operational Research

Analytics Types

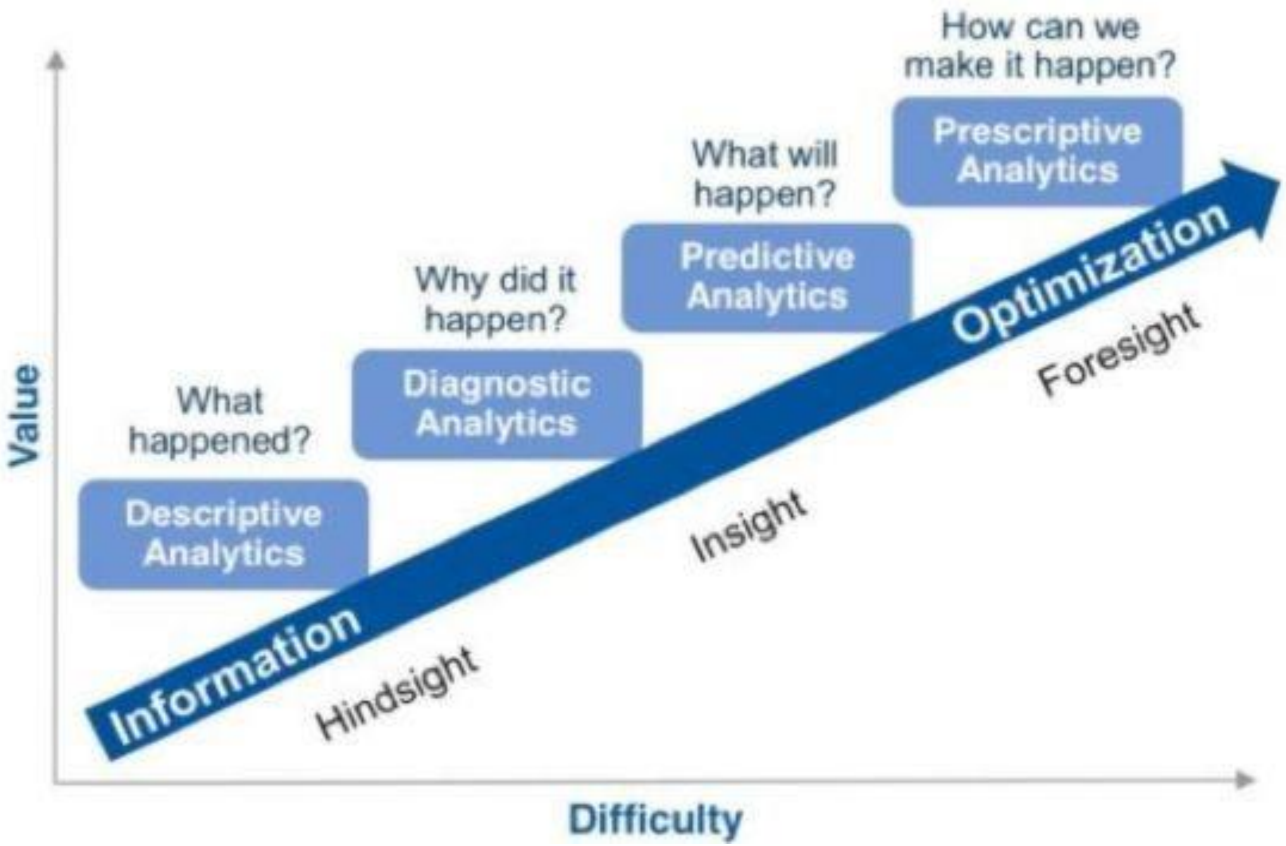


Fig. Relationship between descriptive, predictive & prescriptive analytics

Market Basket Analysis

It is a technique that allow us to discover the relationships between products.

Market Basket Analysis

It can be called

Association Analysis

Frequent itemset mining

Market Basket Analysis Why?

Store Layout

Recommendation Engines

Targeted Marketing

Up Sell & Cross Sell

Catalogue Design

Customer Experience

Use Cases (Applications) of Association Rule Mining

Retail

Telecommunications

Banking

Medical

Manufacturing

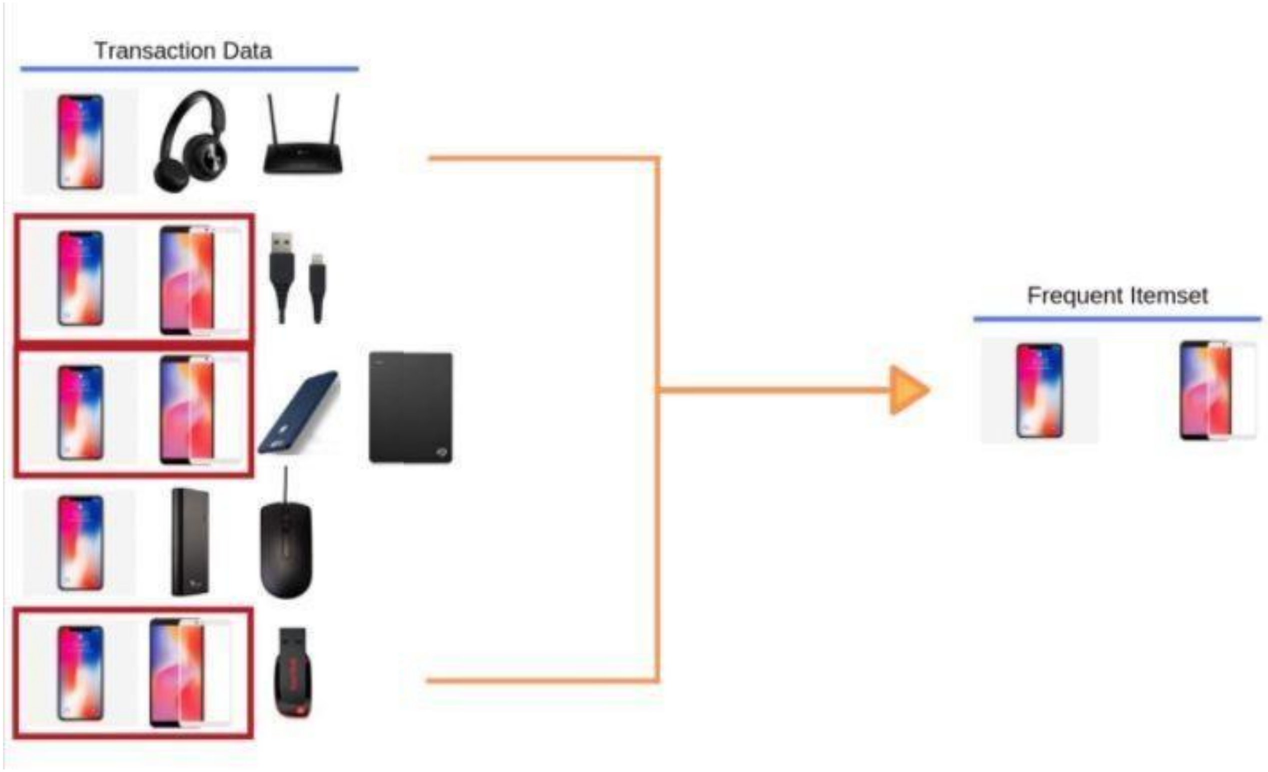
Insurance

Simple Example

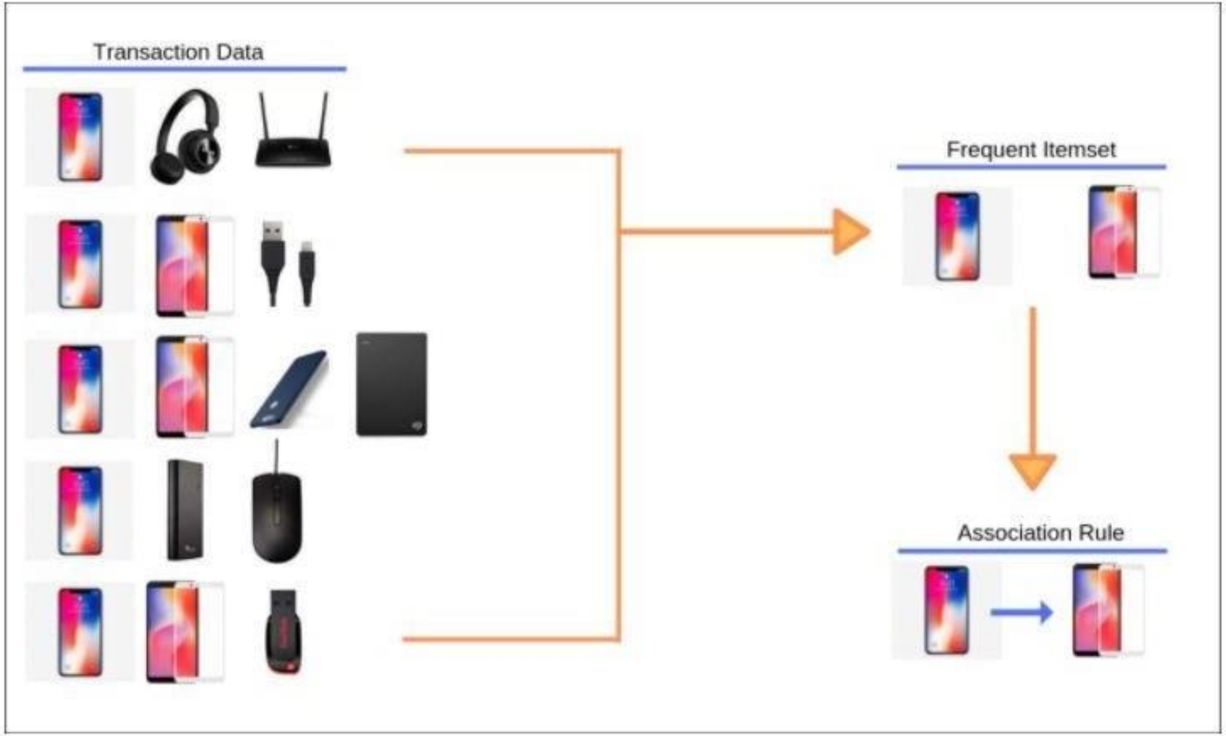
Simple Example -Transaction Data

Simple Example -Transaction Data

Simple Example -Frequent Item Set



Simple Example- Association Rule



Simple Example- Association Rule

Simple Example- Association Rule Support

Association Rule

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

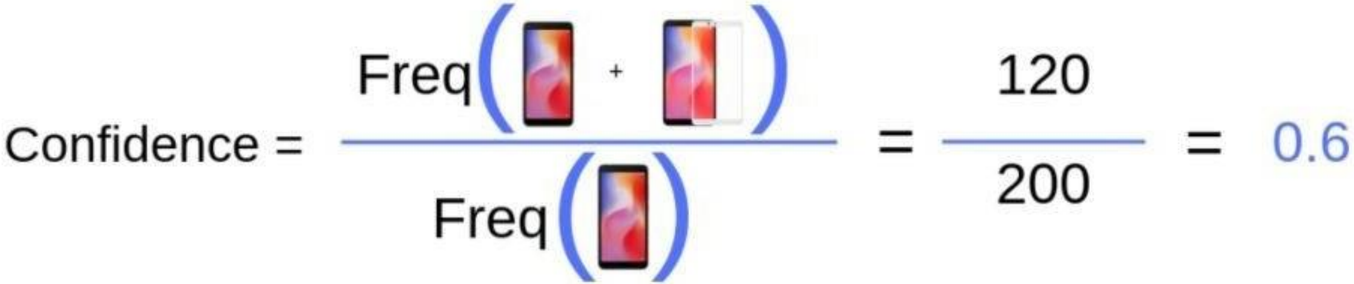
Support = Freq( + ) / Total Transactions

$$= \frac{120}{2000} = 0.06$$

Simple Example- Association Rule Confidence

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120



Simple Example- Association Rule Lift

Total Transactions (N): 2000

Item	Transactions
Mobile	200
Screen Guard	160
Mobile + Screen Guard	120

Simple Example- Association Rule Lift - Interpretation

- **Lift = 1**: implies **no relationship between** mobile phone and screen guard (i.e., mobile phone and screen guard occur together only by chance)
- **Lift > 1**: implies that **there is a positive relationship between** mobile phone and screen guard (i.., mobile phone and screen guard occur together more often than random)
- **Lift < 1**: implies that **there is a negative relationship between** mobile phone and screen guard (i.e., mobile phone and screen guard occur together less often than random)

- Frequent itemsets from the previous section can form **candidate rules such as X implies Y** .

$$X \rightarrow Y$$

$$Rule: \; X \Rightarrow Y$$

Association Rule

Rule: $X \Rightarrow Y$

Appropriateness of Candidate Rule

Support

Confidence

Lift

Association Rule/ Apriori Example

TID	List_Of_Item IDs
T100	I1, I2, I5
T101	I2, I4
T102	I2, I5
T103	I1, I2, I4
T104	I1, I2, I3
T105	I2, I3
T106	I1, I2, I3, I4
T107	I1, I2, I3
T108	I1, I3 , I5

ITEM Set	FREQUENCY
{ I1}	6
{I2}	8
{I 3}	5
{I4}	3
{I5}	3

Example

TID	List_Of_Item IDs
T100	I1, I2, I5
T101	I2, I4
T102	I2, I5
T103	I1, I2, I4
T104	I1, I2, I3
T105	I2, I3
T106	I1, I2, I3, I4
T107	I1, I2, I3
T108	I1, I3 , I5

ITEM Set	FREQUENC Y
{ I1}	6
{I2}	8
{I 3}	5
{I4}	3
{I5}	3

ITEM Set	FREQUENCY
{ I1}	6
{I2}	8
{I 3}	5

Example

ITEM Set	FREQUENCY
{ I1}	6
{I2}	8
{I 3}	5

ITEM Set	FREQUENCY
{ I1, I2}	5
{I1, I3}	4
{I 2, I3}	4

Example

ITEM Set	FREQUENCY
{ I1, I2}	5
{I1, I3}	4
{I 2, I3}	4

ITEM Set	FREQUENCY
{ I1, I2,I3}	3

Example Support

Rule	Frequency of X+ Y	Formula	Putting Values in Formula	Support Value
I1 => I2	5	Freq(X+Y) / No of Transaction	5/9	0.55
I1 => I3	4		4/9	0.44
I2 => I3	4		4/9	0.44

Example- Confidence

Rule	Freq(X)	Freq (X+ Y)	Formula for Confidence	Putting Values in Formula	Confidence (x=>y)
I1 => I2	6	5	$\frac{\text{Freq(X+Y)}}{\text{Freq (X)}}$	5/6	0.83
I1 => I3	6	4		4/6	0.66
I2 => I3	8	4		4/8	0.50

Example **Lift**

Rule	Support of (X+ Y)	Support of X	Support of Y	Formula	Putting Values in Formula	Support Value
I1 => I2	0.55	6/9 = 0.66	8/9 =0.88	Support(X+Y) / Support (X) * Support (Y)	0.55 / (0.66 * 0.88)	0.94
I1 => I3	0.44	6/9 = 0.66	5/9 =0.55		0.44 / (0.66 * 0.55)	1.21
I2 => I3	0.44	8/9 =0.88	5/9 =0.55		0.44 / (0.88 * 0.55)	0.90

Example

Rule	Support	Confidence	Lift
I1 => I2	0.55	0.83	0.94
I1 => I3	0.44	0.66	1.21
I2 => I3	0.44	0.50	0.90

Example

Rule	Support	Confidence	Lift
I1 => I2	0.55	0.83	0.94
I1 => I3	0.44	0.66	1.21
I2 => I3	0.44	0.50	0.90

Example

Rule	Support	Confidence	Lift
I1 => I2	0.55	0.83	0.94
I1 => I3	0.44	0.66	1.21

Applications of Association Rules

The term market basket analysis refers to a specific implementation of association rules

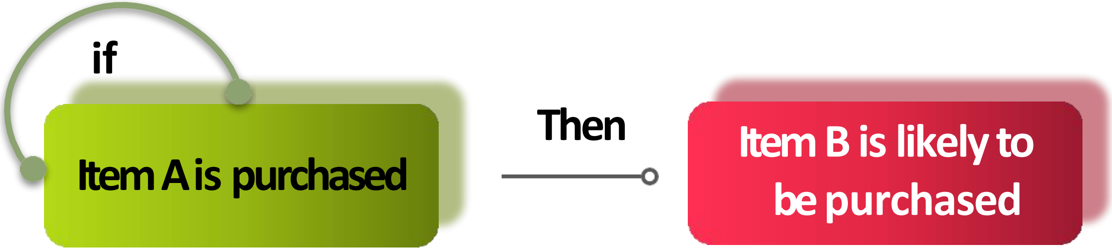
- For better merchandising – products to include/exclude from inventory each month
- Placement of products
- Cross-selling
- Promotional programs—multiple product purchase incentives managed through a loyalty card program

Applications of Association Rules

Association rules also used for

- Recommender systems – Amazon, Netflix
- Clickstream analysis from web usage log files
- Website visitors to page X click on links A,B,C more than on links D,E,F

It creates **If-Then** scenario rules



Rule written as → If {A} Then {B}

Market Basket Analysis

If Part

Called

if

Item A is purchased

Antecedent

It is Condition

Then

Then Part

Item B is likely to be purchased

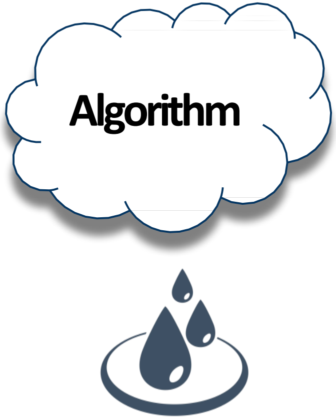
Called

Consequent

It is result

Market Basket Analysis

UNIT - IV



Association Rule

Apriori

Algorithm

Association Rule

Support

measures

Confidence

Lift

Algorithm

Association Rule

Support

- Support is the number of transactions that include items in the (A) & {B} parts of the rule as a percentage of the total number of transactions.
- It is a measure of how frequently the collection of items occur together as a percentage of all transaction.

$$\text{Support} = \frac{A+B}{\text{Total}}$$

Confidence

Algorithm

Association Rule

- Confidence of the rule is the ratio of the number of transactions that include all items in (B) as well as the number of transactions that include all items in (A) to the number of transactions that include all items in (A).

$$\text{Confident} = \frac{A+B}{A}$$

Association Rules

- Association analysis is useful for discovering interesting relationships hidden in large data sets.
- The uncovered relationships can be represented in the form of association rules or sets of frequent items.

Association Rules

- Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or casual structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other forms of data repositories.

- Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a transactional database, relational database or other information repository.

Association Rules

- An example of an association rule would be

"If a customer buys a 1 packet bread, he is 80 % likely to also purchase milk."

ID	Items
1	{Bread, Milk}
2	{Bread, Milk, Cola, Sugar}
3	{Bread, Milk, Tea, Sugar}
...	...

Market basket transaction

{ Bread, Milk } — Example of frequent itemset

{ Bread } $\longrightarrow$ { Milk } — Example of association rule

Association Rules

- Association rule mining can be viewed as a two-step process :

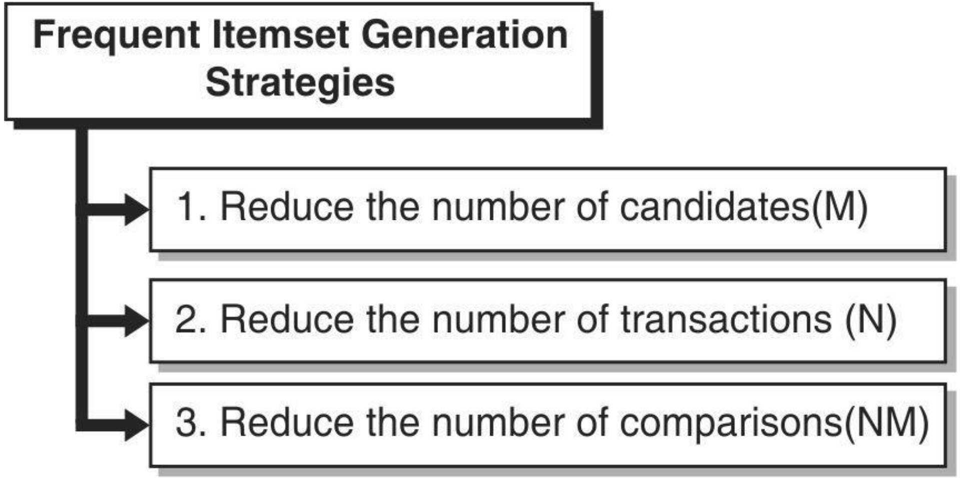
1. Find all frequent itemsets :

By definition, each of these item sets will occur at least as frequently as a predetermined minimum support count, min sup.

2. Generate strong association rules from the frequent item sets :

By definition, these rules must satisfy minimum support and minimum confidence.

Frequent Item Set Generation



Frequent Item Set Generation

- **Reduce the number of candidates (M)**
  - By reducing the number of candidates from the recent itemset generated the complexity automatically reduced.
  - Suppose Complete search : M= 2d here pruning techniques are used to reduce M.

Frequent Item Set Generation

- **Reduce the number of transactions (N)**

  By Reducing the size of N as the size of itemset increases the complexity can be reduced.

Frequent Item Set Generation

- **Reduce the number of comparisons (NM)**
    - By using efficient data structures to store the candidates or transactions, there will be no need to match every candidate against every transaction. So, rework can be eliminated and complexity is reduced.
    - Apriori Algorithm implements strategy by reducing the number of candidates.

Algorithm

Apriori

Apriori Algorithm

Algorithm

Apriori

- In learning association rules, Apriori is a classic machine learning algorithm.
- Apriori is designed to work on databases covering transactions (for example, collections of items bought by customers, or details of a website frequentation).
- The algorithm is aimed to find subsets which are common to at least a minimum number C (the cut off, or confidence threshold) of the itemsets.

Apriori Algorithm

Algorithm

Apriori

- It follows a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data.
- The algorithm continues till no further successful extensions have been found.
- Apriori uses breadth-first search and a hash tree structure to count candidate item sets efficiently.

Apriori Algorithm

Algorithm

Apriori

Key Concepts

1. **Frequent Itemsets**

- The sets of items which have minimum support (denoted by L_i for itemsets of i elements).

2. **Apriori Property**

- Any subset of a frequent itemset must be frequent.
- i.e., if {*AB*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset.
- Iteratively find frequent itemsets with cardinality from 1 to *k* (*k*- itemset).

Apriori Algorithm

Algorithm

Apriori

Key Concepts

4. Join Operation

- To find L_k , a set of candidates k-itemsets is generated by joining L_{k-1} with itself.
- Use the frequent itemsets to generate association rules.

5. Creating frequent sets

- Algorithm uses breadth-first search and a hash tree structure to handle candidate itemsets efficiently then frequency for each candidate itemset is counted.
- Those candidate itemsets that have frequency higher than minimum support threshold are qualified to be frequent itemsets

Apriori Algorithm

Algorithm

Apriori

Let's define

C_k as a candidate itemset of size k.

L_k as a frequent itemset of size k.

Main steps of iteration are :

1. Find frequent itemset L_{k-1} (starting from L_1).
2. Join step : C_k is generated by joining L_{k-1} with itself (cartesian product $L_{k-1} \times L_{k-1}$).
3. Prune step (apriori property) : Any (k – 1) size itemset that is not frequent cannot be a subset of a frequent k size itemset, hence should be removed from C_k.
4. Frequent set L_k has been achieved.

Apriori Algorithm

Algorithm

Apriori

Pseudocode for Apriori Algorithm

Join step

It is generated by joining with itself.

Prune Step

Any (k-1) item set that is not frequent cannot be a subset of a frequent k-item set.

Apriori Algorithm

Algorithm

Apriori

Pseudocode for Apriori Algorithm

Pseudo-code

C_k : Candidate item set of size k

L_k :Frequent item set of size k

L_1 = {frequent items}

For (k = 1; L_k! = Φ; k++) do begin

C_{k+1} = Candidates generated from L_k ;

For each transaction t in database do

Increment the count of all candidates in C_{k+1}

Those are contained in t

L_{k+1} = Candidates in C_{k+1} with min_support

End

Return $U_k\,L_k$;

Apriori Algorithm

Example of Apriori Algorithm

Table P.4.4.3 transaction with 9 items

Algorithm

Apriori

TID	List of Items
T100	I1, I2, I5
T101	I2, I4
T102	I2, I3
T103	I1, I2, I4
T104	I1, I3
T105	I2, I3
T106	I1, I3
T107	I1, I2 ,I3, I5
T108	I1, I2, I3

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Algorithm

Apriori

Solution

- Consider a database, D, consisting of 9 transactions.
- Suppose min. support count required is 2 (i.e. minsup = 2/9 = 22 %)
- Let minimum confidence required be 70%.
- First find out the frequent itemsets using Apriori algorithm. Then, Association rules will be generated using min. support and min. confidence.

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Algorithm

Solution **Step : - 1** **Generating 1-itemset Frequent Pattern**

- In the first iteration of the algorithm, each item is a member of the set of candidates.
- The set of frequent 1-itemsets, L_1, consists of the candidate 1-itemsets satisfying minimum support.

Apriori

Scan D to count of each candidate

Itemset	Sup. Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

C_1

Compare candidate support count with minimum support count

Itemset	Sup. Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Solution Step : - 2

Generating 2-itemset Frequent Pattern

Algorithm

Apriori

Generate C_2 candidates from L_1 (L_1 L_1)

Itemset
{I1, I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I4}
{I3,I5}
{I4,I5}

C_2

Scan D for Count of each candidate

Itemset	Sup. Count
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4}	0
{I3,I5}	1
{I4,I5}	0

Itemset	Sup. Count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

L_2

Compare candidate support count with minimum support count

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Solution **Step : - 3** **Generating 3-itemset Frequent Pattern**

Algorithm

- In order to find C3, compute L2 Join L2.
- C3 = L2 Join L2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}.

Apriori

Now, join step is complete and Prune step will be used to reduce the size of C3. Prune step helps to avoid heavy computation due to large Ck.

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Algorithm

Apriori

Solution **Step : - 3** **Generating 3-itemset Frequent Pattern**

- For example, let's take {I1, I2, I3}. The 2-item subsets of it are {I1, I2}, {I1, I3} & {I2, I3}. Since all 2-item subsets of {I1, I2, I3} are members of L2, we will keep {I1, I2, I3} in C3.
- Let's take another example of {I2, I3, I5} which shows how the pruning is performed. The 2-item subsets are {I2, I3}, {I2, I5} & {I3, I5}. but, {I3, I5} is not a member of L2 and hence it is not frequent and it is violating Apriori Property. Thus {I2, I3, I5} is removed from C3.
- Therefore, C3 = {{I1, I2, I3}, {I1, I2, I5}} after checking for all members of result of Join operation for Pruning.
- Now, the transactions in D are scanned in order to determine L3, consisting of those candidates 3-itemsets in C3 having minimum support.

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Solution **Step : - 3** **Generating 3-itemset Frequent Pattern**

Algorithm

Scan D for count of each candidate L_2

Itemset
{I1, I2, I3}
{I1,I2, I5}
....

C_3

Scan D for count of each candidate

Itemset	Sup. Count
{I1,I2, I3}	2
{I1,I2,I5}	2

C3

Compare candidate support count with minimum support count

Itemset	Sup. Count
{I1,I2, I3}	2
{I1,I2, I5}	2

L3

Apriori

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Solution **Step : - 4** **Generating 4-itemset Frequent Pattern**

Algorithm

Apriori

- The algorithm uses L_3 *Join* L_3 to generate a candidate set of 4-itemsets, C_4. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned since its subset {{I2, I3, I5}} is not frequent.
- Thus, C_4 = φ, and algorithm terminates, having found all of the frequent items. This completes the task of Apriori Algorithm.
- In next step these frequent itemsets will be used to generate strong association rules (where strong association rules satisfy both minimum support & minimum confidence).

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Algorithm

Apriori

Solution **Step : - 5** **Generating Association Rules from Frequent Itemsets**

- For each frequent itemset, generate all nonempty subsets of ***l*.**
- For every nonempty subset ***s*** of ***l*,** output the rule "s □ (l - s)"

  If support count(l)/support count(s) >= minconf

  Where minconf is minimum confidence threshold.

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Algorithm

Apriori

Solution **Step : - 5** **Generating Association Rules from Frequent Itemsets**

- **In the above example the rules generate are,**
  - L = {{I1}, {I2}, {I3}, {I4}, {I5}, {I1, I2}, {I1, I3}, {I1, I5}, {I2, I3}, {I2, I4}, {I2, I5}, {I1, I2, I3}, {I1, I2, I5}}
  - Consider ***l*** = {I1, I2, I5}. It's all nonempty subsets are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, {I5}.
  - ***l*** = {I1, I2, I5}, **s** : {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, {I5}
  - Let minimum confidence threshold is, say 70%.

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Solution

Step : - 5

Generating Association Rules from Frequent Itemsets

Algorithm

Apriori

- The resulting association rules are shown below, each listed with its confidence.

Rule	Confidence	Decision
R1: I1 ^ I2 -> I5	Freq {I1, I2, I5}/Freq {I1, I2} = 2/4 = 50%	R1 is Rejected.
R2: I1 ^ I5 -> I2	Freq {I1, I2, I5}/ Freq {I1, I5} = 2/2 = 100%	R2 is Selected.
R3: I2 ^ I5 -> I1	Freq {I1, I2, I5}/ Freq {I2, I5} = 2/2 = 100%	R3 is Selected.
R4: I1 -> I2 ^ I5	Freq {I1, I2, I5}/ Freq {I1} = 2/6 = 33%	R4 is Rejected.
R5:I2 -> I1 ^ I5	Freq {I1, I2, I5}/ Freq {I2} = 2/7 = 29%	R5 is Rejected.
R6: I5 -> I1 ^ I2	{I1, I2, I5}/ Freq{I5} = 2/2 = 100%	R6 is Selected.

Apriori Algorithm

Example of Apriori Algorithm → Table P.4.4.3 transaction with 9 items

Solution

Algorithm

Apriori

In this way, three strong association rules derived are :

- If (I1 and I2)then I5
- If (I2 and I5) then I1
- If (I5 and I1) then I2

Apriori Algorithm

Algorithm

Apriori

Drawback

The two primary drawbacks of the Apriori Algorithm are:

1. At each step, candidate sets have to be built.
2. To build the candidate sets, the algorithm has to repeatedly scan the database

Frequent Pattern (FP) Growth

Algorithm

FP Growth

- an improvement of apriori algorithm.
- used for finding frequent itemset in a transaction database without candidate generation.
- represents frequent items in frequent pattern trees or FP-tree.

Frequent pattern growth

Algorithm

Example

FP Growth

Transaction ID	Items
T1	{E, K, M, N, O, Y}
T2	{D, E, K, N, O, Y}
T3	{A, E, K, M}
T4	{C, K, M, U, Y}
T5	{C, E, I, K, O, O}

Frequent pattern growth

Example

Algorithm

FP Growth

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
0	4
U	1
Y	3

Frequent pattern growth

Algorithm

FP Growth

Example

- minimum support be 3
- These elements are stored in descending order of their respective frequencies.
- After insertion of the relevant items, the set L looks like this:-

$L = \{K : 5, E : 4, M : 3, O : 3, Y : 3\}$

Frequent pattern growth

Algorithm

FP Growth

Example

Ordered-Item set

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}

Item sorting : Items in a transaction are sorted in descending order of support counts.

Frequent pattern growth

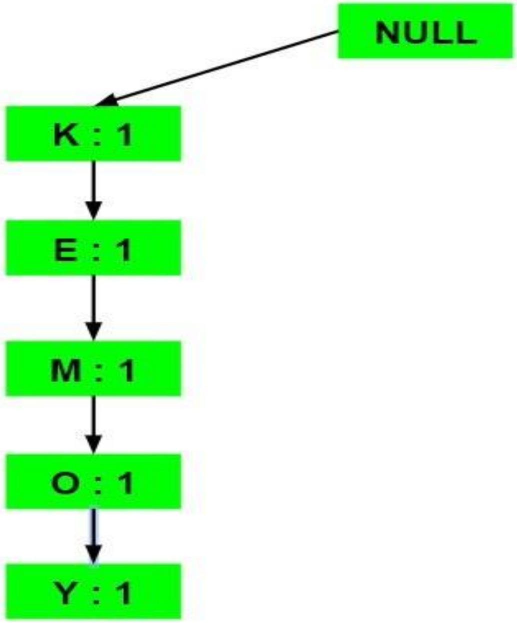
Algorithm

Example

FP Growth

Tree Data Structure: **Inserting the set {K, E, M, O, Y}**

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}



Frequent pattern growth

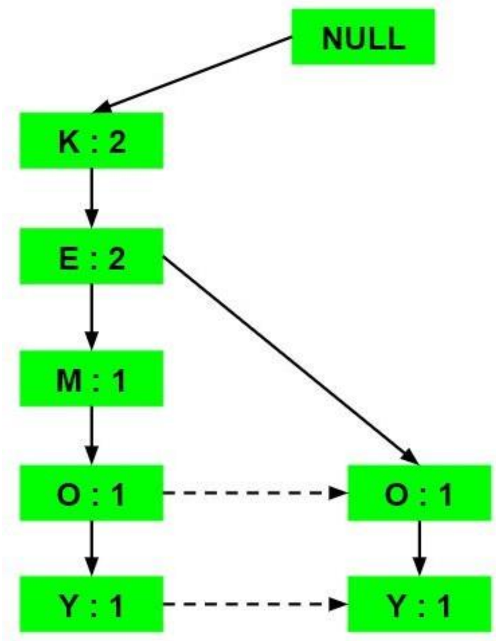
Algorithm

Example

FP Growth

Tree Data Structure: **Inserting the set {K, E,O, Y}**

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}



Frequent pattern growth

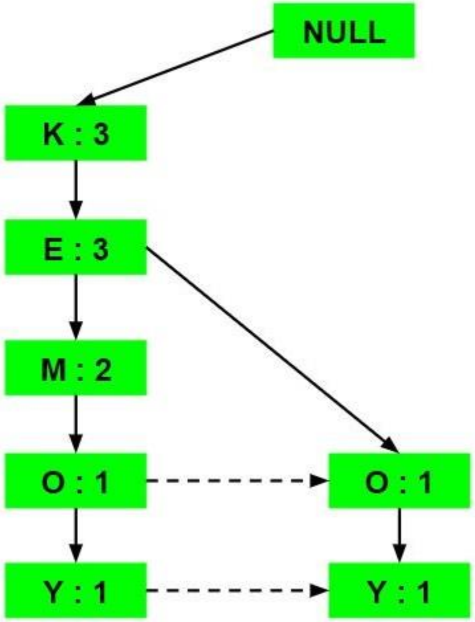
Example

Algorithm

FP Growth

Tree Data Structure: **Inserting the set {K, E,M}**

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}



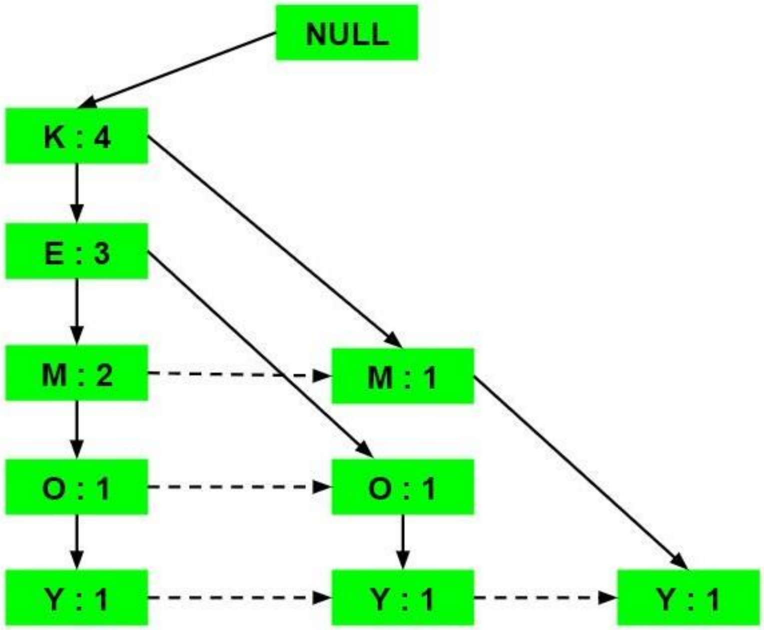
Frequent pattern growth

Algorithm

Example

Tree Data Structure: **Inserting the set {K, M,Y}**

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}



FP Growth

Frequent pattern growth

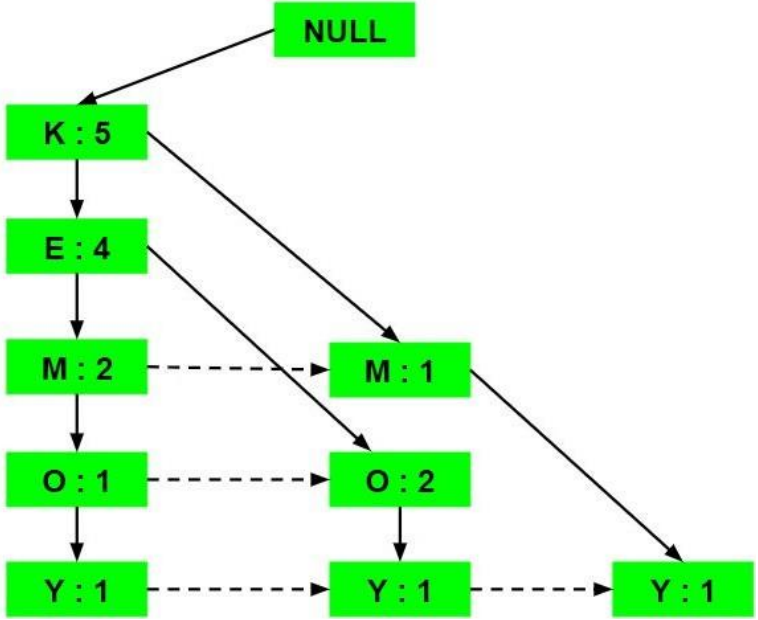
Algorithm

Example

FP Growth

Tree Data Structure: **Inserting the set {K, E, O}**

Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}

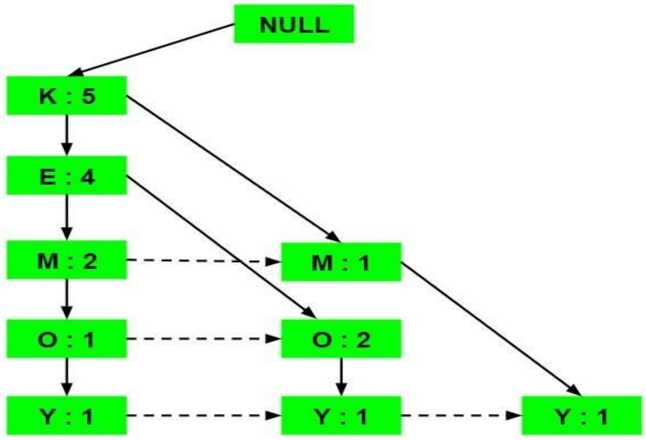


Frequent pattern growth

Algorithm

FP Growth

Example Conditional Pattern Base



Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Frequent pattern growth

Algorithm

FP Growth

Example

Conditional Frequent Pattern Base

It is done by taking the set of elements that is common in all the paths in the Conditional Pattern Base of that item and calculating its support count by summing the support counts of all the paths in the Conditional Pattern Base.

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

Frequent pattern growth

Example

Frequent Pattern rules

Algorithm

FP Growth

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 3>}
K	

Regression

- Regression is a data mining function that predicts a number.
- Profit, sale, mortgage rates, house values, square footage, temperature or distance could all be predicted using regression techniques.
- For example, a regression model could be used to **predict the values of a data warehouse** based on web-marketing, number of data entries, size and other factors

Regression

- A regression task begins with a data set in which the target values are known.
- Regression analysis is a good choice when all of the predictor variables are continuously valued as well.
- For an input x, if the output is continuous, this is called a regression problem.

Regression

- For example, based on historical information of demand for toothpaste in your supermarket, you are asked to predict the demand for the next month.
- Regression is concerned with the prediction of continuous quantities.
- Linear regression is the oldest and most widely used predictive model in field of machine learning.
- The goal is to minimize the sum of the squared errors to fit a straight line to a set of data points.

Regression

Regression Line

Algorithm

Regression

Least squares :

- The least squares regression line is the line that makes the sum of squared residuals as small as possible.
- Linear means "straight line".

Regression Line :

- It is the line which gives the best estimate of one variable from the value of any other given variable.
- The regression line gives the average relationship between the two variables in mathematical form.

Regression

Regression Line Linear Regression

Algorithm

Regression

- For two variables X and Y, there are always two lines of regression.

Regression line of X on Y:

Gives the best estimate for the value of X for any specific given values of Y:

X = a + b Y

where,
a = X - intercept
b = Slope of the line
X = Dependent variable
Y = Independent variable

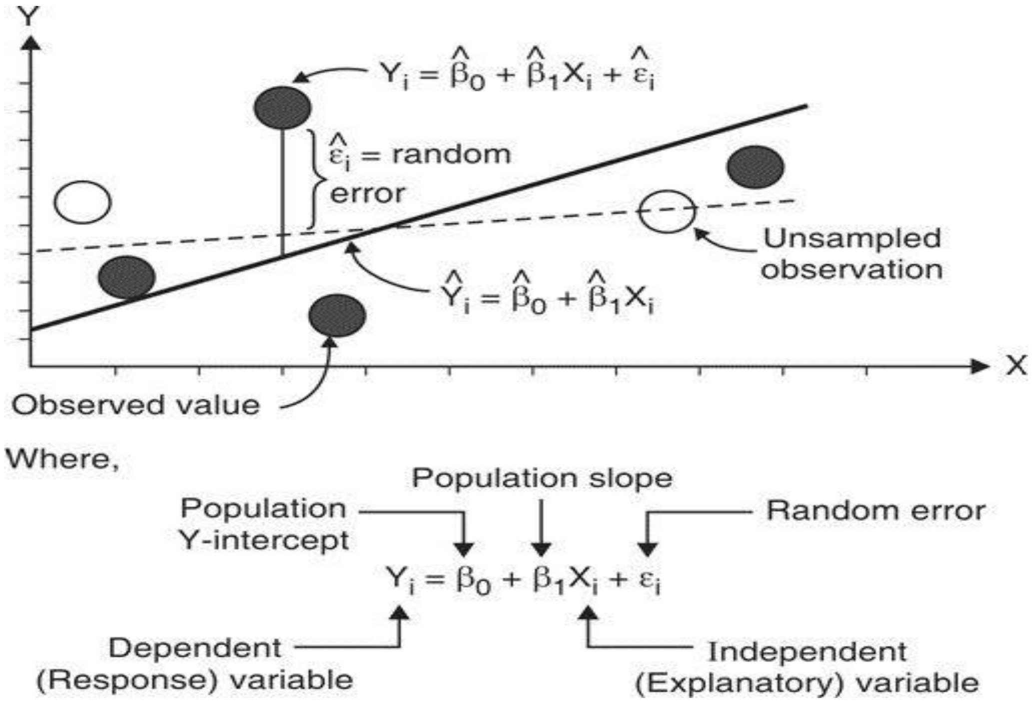
Regression

Regression Line Linear Regression

- For two variables X and Y, there are always two lines of regression.

Algorithm

Regression



Regression

Regression Line

Algorithm

Regression

Linear Regression Example :

- ❑ The simplest form of regression to visualize is linear regression with a single predictor.

- ❑ A linear regression technique can be used if relationship between X and Y can approximated with a straight line.

Regression Line

Algorithm

Regression

Linear Regression Example :

Consider following data

(i) Find values of b0 and b1 w.r.t. linear regression model which best fits given data.

(ii) Interpret and explain equation of regression line.

(iii) If new person rates "Bahubali-Part-I" as 3 then predict the rating of same person for "Bahubali-Part-II"

Regression Line

Algorithm

Regression

Linear Regression Example :

Person	Xi = rating for movie "Bahubali-Part-I" by ith person	Yi = rating for movie "Bahubali-Part-II" by ith person
1^{st}	4	3
2^{nd}	2	4
3^{rd}	3	2
4^{th}	5	5
5^{th}	1	3
6^{th}	3	1

Regression

Regression Line

Algorithm

Regression

Linear Regression Example :

Average of X Values $\bar{X} = 3$

Average of Y Values $\bar{Y} = 3$

x	y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})^2$	$(x-\bar{x})(y-\bar{y})$
4	3	1	0	1	0
2	4	−1	1	1	-1
3	2	0	-1	0	0
5	5	2	2	4	4
1	3	-2	0	4	0
3	1	0	-2	0	0
$\bar{x} = 3$	$\bar{y} = 3$			$\Sigma(x-\bar{x})^2 = 10$	$\Sigma(x-\bar{x})(y-\bar{y}) = 3$

Regression Line

values of β0 and β1 w.r.t. linear regression model

Algorithm

Regression

$$\bar{y} = \beta_0 + \beta_1 \bar{x}$$

$$\beta_1 = \frac{\Sigma (x - \bar{x})(y - \bar{y})}{(x - \bar{x})^2}$$

$$\beta_1 = 3/10 = 0.3$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_0 = 3 - (0.3 * 3)$$

$$\beta_0 = 2.1$$

Equation of Line:

$$\hat{y} = \mathbf{2.1 + 0.3 \ (X)}$$

Regression

Regression Line

Algorithm

Regression

Interpretation 1

For increase in value of x by 0.3 unit there is increases in value of y in one units.

Interpretation 2

Even if x = 0 value of independent variable, it is expected that value of y is 2.1.

Regression Line

Algorithm

Regression

- **If new person rates "Bahubali-Part-I" as 3 then predict the rating of same person for "Bahubali-Part-II"**
  - For x=3 the y value will be
  - Y (Predicted) = 2.1 + 0.3 (3) = 2.1+ 0.9
- If new person rates "**Bahubali-Part-I" as 3** then predict the rating of same person for **"Bahubali-Part-II" is 3.9**

Regression

Logistic Regression

Algorithm

Regression

- ❑ Logistic regression is a form of regression analysis in which the outcome variable is binary.
- ❑ A statistical method used to model binary outcomes using predictor variables.
- ❑ Logistic component : Instead of modeling the outcome, Y, directly, the method models the log odds (Y) using the logistic function.

Regression

Logistic Regression

Algorithm

- Methods used to quantify association between an outcome and predictor variables. It could be used to build predictive models as a function of predictors

- In simple logistic regression, logistic regression with 1 predictor variable.

Regression

Logistic Regression

$$\ln [P/(1-P)] = a_0 + a_1X_1 + a_2X_2 + \text{-------------} + a_kX_k$$

Regression

Logistic Regression

Algorithm

Regression

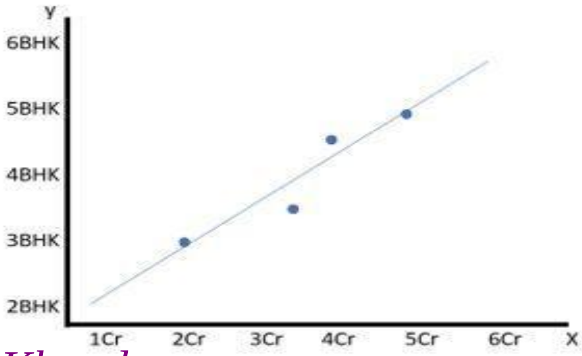
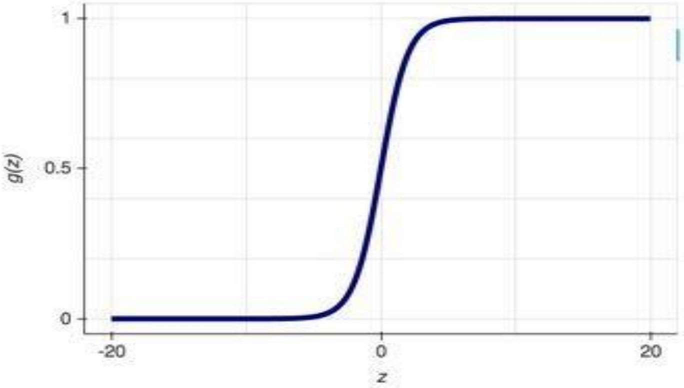
$$odds = \frac{P}{1-P}$$

Y	*1*	*0*
Pr(Y=1)	*P*	*1- P*

**P= Success, 1-P = Failure*

Logistic Regression

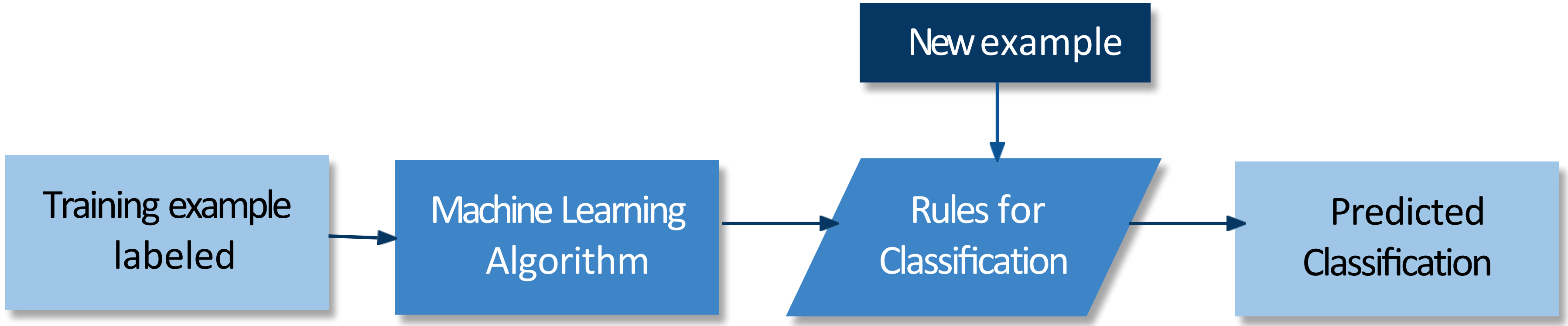
$$\ln\left[P/(1-P)\right] = a_0 + a_1X_1 + a_2X_2 + \text{--------------} + a_kX_k$$

Linear Regression	Logistic Regression
Target is an interval variable	Target is discrete (binary or ordinal) variable
Predicted values are the mean of the target variable at the given values of the input variable	Predicted values are the probability of the particular levels of the given values of the input variable
Solve regression problems	Solve classification problems
Example : What is the Temperature?	Example : Will it rain or not?
Graph is straight line	Graph is S-curve
	

Classification

- It Predicts categorical labels (classes), prediction models continuous-valued functions.
- Classification is considered to be supervised learning.
- Preprocessing of the data in preparation for classification and prediction can involve data cleaning to reduce noise or handle missing values,
- relevance analysis to remove irrelevant or redundant data transformation such as generalizing the data to higher level concepts or normalizing data

Classification



Classification

Naive Bayes

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.

Decision Tree

It is a part of classification algorithm which also provides solutions to the regression problems using the **classification rule**

Joint Probability Example

P(Red and King)

$$= \frac{\text{number of cards that are red and king}}{\text{total number of cards}} = \frac{2}{52}$$

Type	Color		Total
	Red	Black	
King	2	2	4
Non-King	24	24	48
Total	26	26	52

Marginal Probability Example

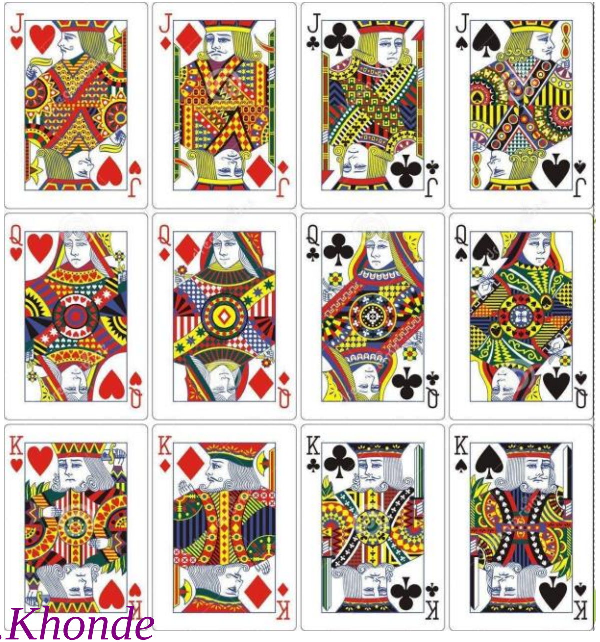
P(King)

$$= P(King\ and\ Red) + P(King\ and\ Black) = \frac{2}{52} + \frac{2}{52} = \frac{4}{52}$$

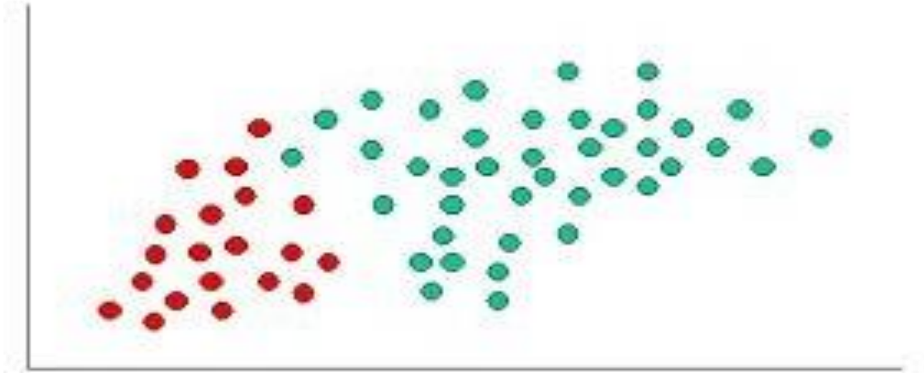
Type	Color		Total
	Red	Black	
King	2	2	4
Non-King	24	24	48
Total	26	26	52

Conditional Probability Example

From the face card the probability of selecting one card of the type Heart and Jack is 1/12. Total number of face cards is 12, which have only one heart of Jack.



Naïve Bayes Classification



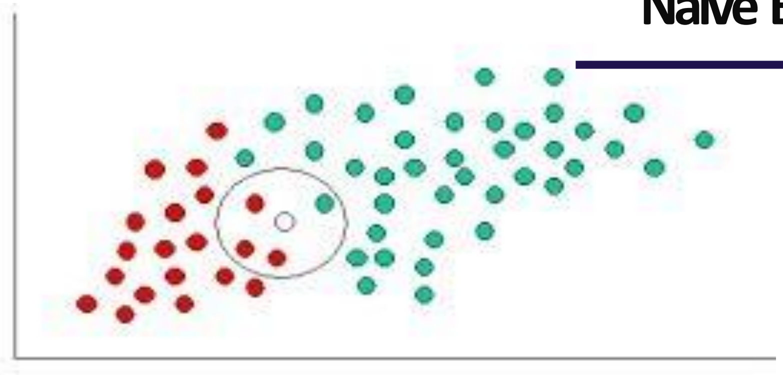
$$\text{Prior probability for GREEN} \propto \frac{\text{Number of GREEN objects}}{\text{Total number of objects}}$$

$$\text{Prior probability for RED} \propto \frac{\text{Number of RED objects}}{\text{Total number of objects}}$$

$$\text{Prior probability for GREEN} \propto \frac{40}{60}$$

$$\text{Prior probability for RED} \propto \frac{20}{60}$$

Naïve Bayes Classification



$$\text{Likelihood of } X \text{ given GREEN} \propto \frac{\text{Number of GREEN in the vicinity of } X}{\text{Total number of GREEN cases}}$$

$$\text{Likelihood of } X \text{ given RED} \propto \frac{\text{Number of RED in the vicinity of } X}{\text{Total number of RED cases}}$$

$$\text{Probability of } X \text{ given GREEN} \propto \frac{1}{40}$$

$$\text{Probability of } X \text{ given RED} \propto \frac{3}{20}$$

Naïve Bayes Classification

$$\text{Posterior probability of } X \text{ being GREEN} \propto$$

$$\text{Prior probability of GREEN} \times \text{Likelihood of } X \text{ given GREEN}$$

$$= \frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$$

$$\text{Posterior probability of } X \text{ being RED} \propto$$

$$\text{Prior probability of RED} \times \text{Likelihood of } X \text{ given RED}$$

$$= \frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$$

Finally, we classify X as RED since its class membership achieves the largest posterior probability.

Naïve Bayes Solved Example

Outlook	Temp	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

$$X = [\underbrace{\text{Outlook}}_{x_1}, \underbrace{\text{Temp}}_{x_2}, \underbrace{\text{Humidity}}_{x_3}, \underbrace{\text{Windy}}_{x_4}$$

$$C_k = [\underbrace{\text{Yes}}_{C_1}, \underbrace{\text{No}}_{C_2}]$$

Naïve Bayes Solved Example

Conditional Probability

$$P(C_k \mid X) = \frac{P(X \mid C_k) * P(C_k)}{P(X)}$$

$$P(C_1 \mid x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \cap x_2 \cap x_3 \cap x_4 \mid C_1) * P(C_1)}{P(x_1 \cap x_2 \cap x_3 \cap x_4)}$$

Naïve Bayes Solved Example

Conditional Probability

$$P(C_k \mid X) = \frac{P(X \mid C_k) * P(C_k)}{P(X)}$$

$$P(C_1 \mid x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \cap x_2 \cap x_3 \cap x_4 \mid C_1) * P(C_1)}{P(x_1 \cap x_2 \cap x_3 \cap x_4)}$$

$$\boxed{P(C_1 \mid x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \mid C_1) * P(x_2 \mid C_1) * P(x_3 \mid C_1) * P(x_4 \mid C_1) * P(C_1)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}}$$

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Naïve Bayes Solved Example

Example

In this example we have 4 inputs (predictors). The final posterior probabilities can be standardized between 0 and 1.

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid \quad Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid \quad No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

Naïve Bayes Solved Example

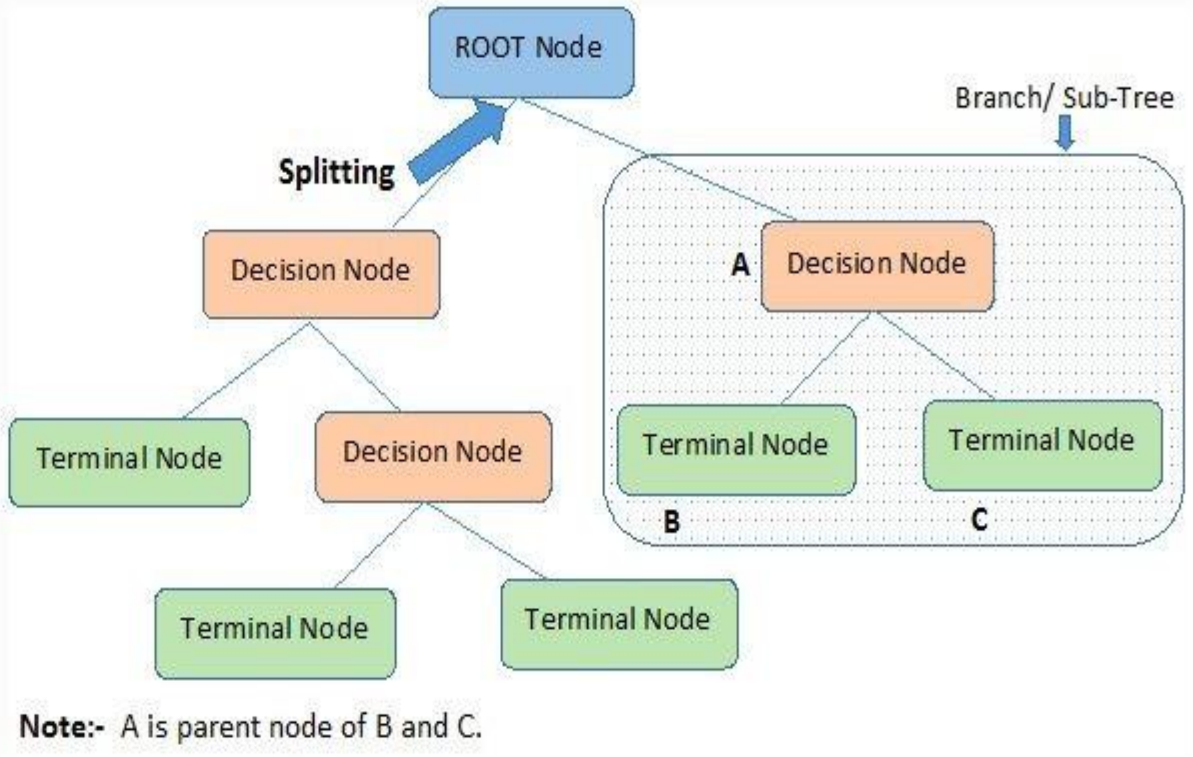
P (N0 | Today) > P (Yes | Today)

So, prediction that golf would be played is ‘No’.

Decision Tree

- To create a training model that can use to predict the class or value of the target **variable by learning simple decision rules** inferred from prior data (training data).
- start from the root of the tree
- compare the values of the root attribute with the record's attribute.
- On the basis of comparison, follow the branch corresponding to that value and jump to the next node.

Decision Trees



Decision Trees

Each node is associated with a feature (one of the elements of a feature vector that represent an object);

Each node test the value of its associated feature;

There is one branch for each value of the feature

Leaves specify the categories (classes)

Can categorize instances into multiple disjoint categories – multi-class

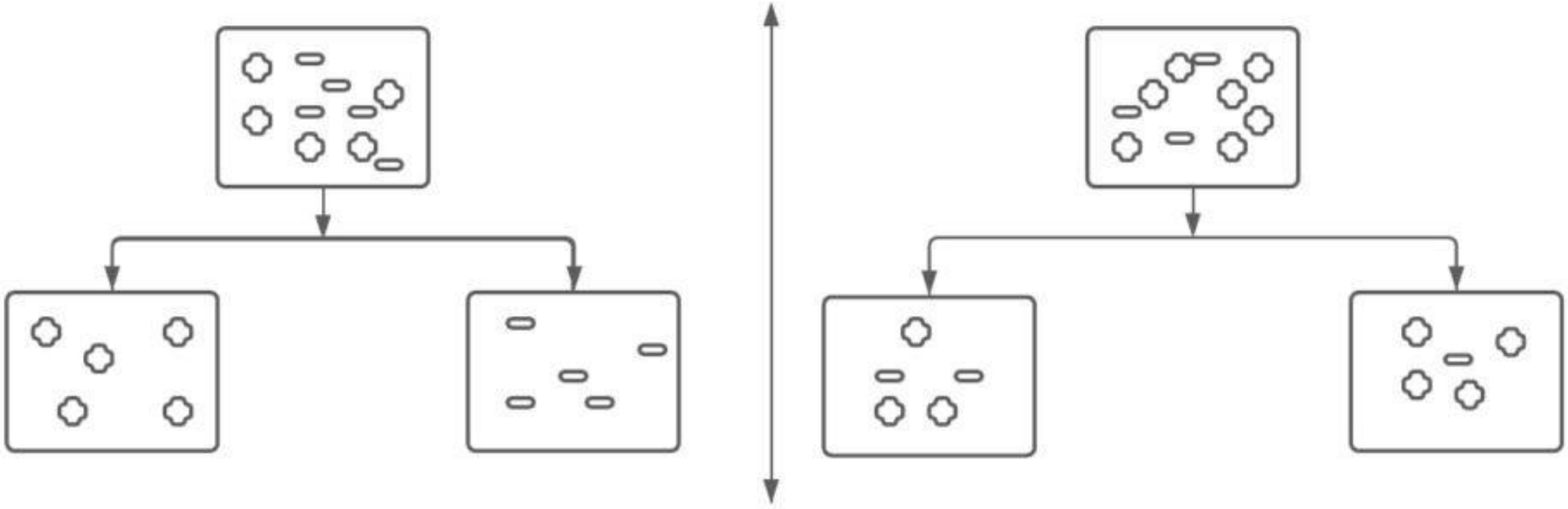
Decision Trees - ID3

- The ID3 algorithm builds decision trees using **a top-down** **greedy search approach** through the space of possible branches with **no backtracking.**
- A greedy algorithm, as the name suggests, **always makes the choice that seems to be the best at that moment.**

Decision Trees - ID3

1. It begins with the **original set S as the root node.**
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and **calculates Entropy(H) and Information gain(IG)** of this attribute.
3. It then selects the attribute which has the **smallest Entropy or Largest Information gain.**
4. The set S is then split by **the selected attribute to produce a subset** of the data.
5. The algorithm continues **to recur on each subset, considering only attributes never selected before.**

Decision Trees - Information Gain



Less Impurities

More Impurities

Information Gain = 1 – Entropy

Decision Trees - Entropy

- The entropy of any random variable or random process **is the average level of uncertainty involved in the possible outcome of the variable or process.**
- To understand it more let's take an example of a **coin flip**
- two probabilities either **it will be a tail, or it will be a head** and if the probability of **tail after flip is p then the probability of a head is 1-p.**
- and the maximum uncertainty is for **p = ½** when there is no reason to expect one outcome over another.
- Here we can say that the entropy here is 1

Decision Trees - Entropy

- Mathematically the formula for entropy is:

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i)$$

Where

X = random variable or process

Xi = possible outcomes

p(Xi) = probability of possible outcomes.

Decision Trees - Information Gain

Gain (S, A) = expected reduction in entropy due to sorting on A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Values (A) is the set of all possible values for attribute A,

S_v is the subset of S which attribute A has value v,

|S| and $|S_v|$ represent the number of samples in set S and set S_v respectively

Gain(S,A) is the expected reduction in entropy caused by knowing the value of attribute A.

Decision Trees

- Play Tennis Example
- Feature values:
  - Outlook = (sunny, overcast, rain)
  - Temperature =(hot, mild, cool)
  - Humidity = (high, normal)
  - Wind =(strong, weak)

Decision Trees

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Decision Trees

UNIT - IV

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

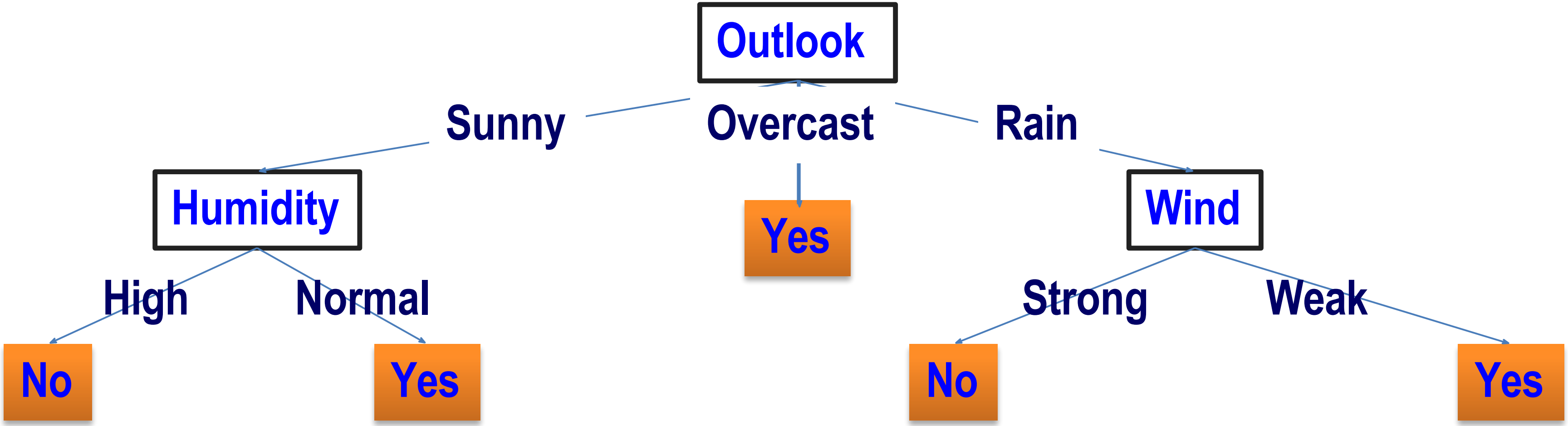
= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

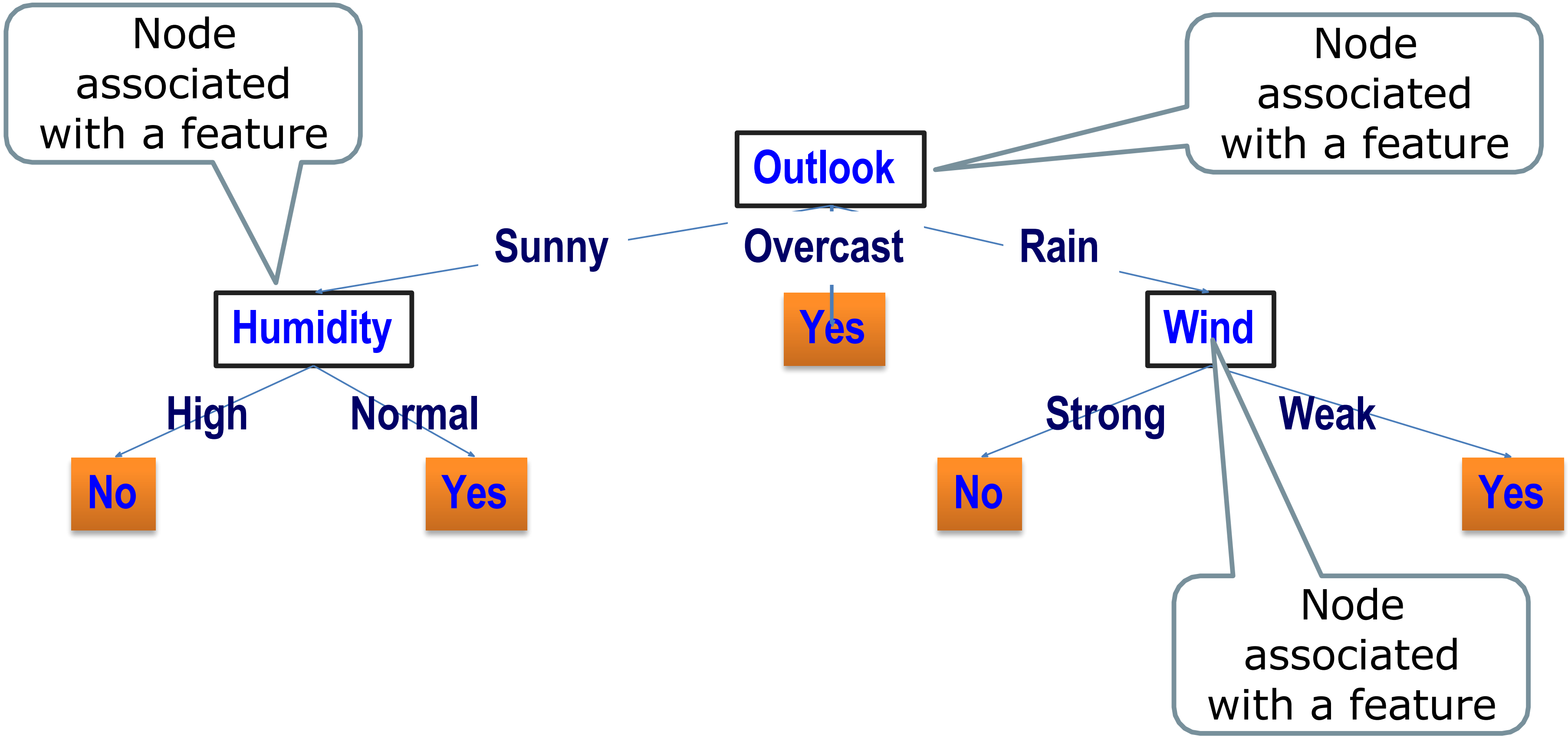
= 0.94

Decision Trees

- Play Tennis Example
- Feature Vector = (Outlook, Temperature, Humidity, Wind)

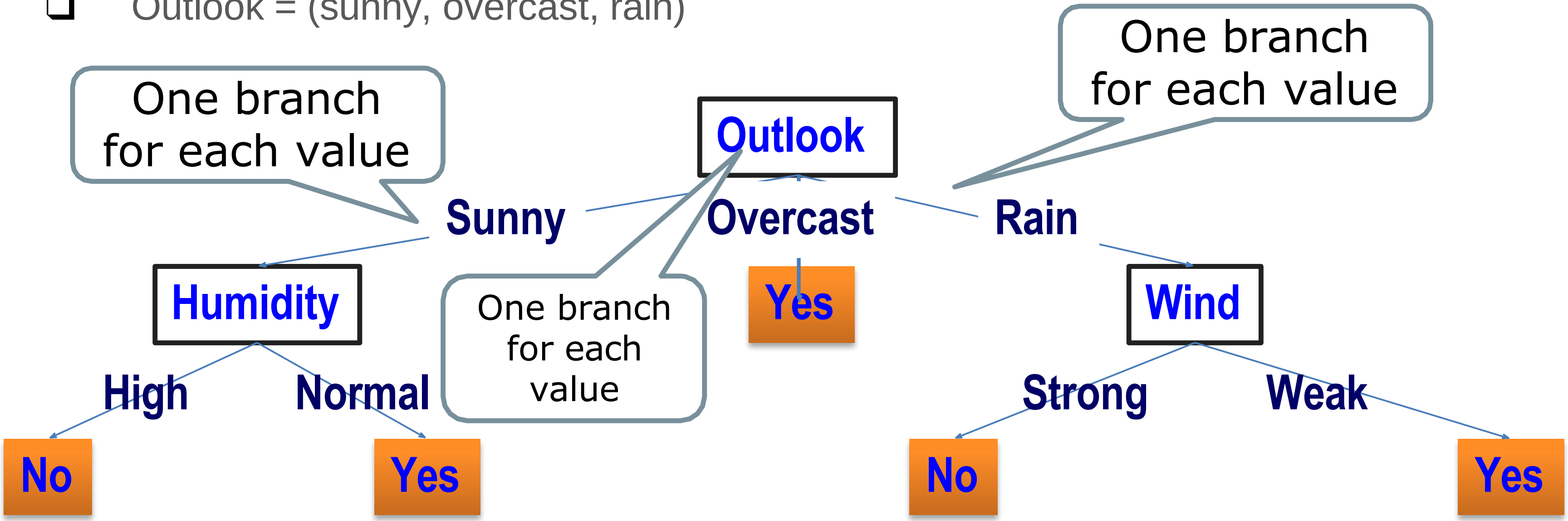


Decision Trees



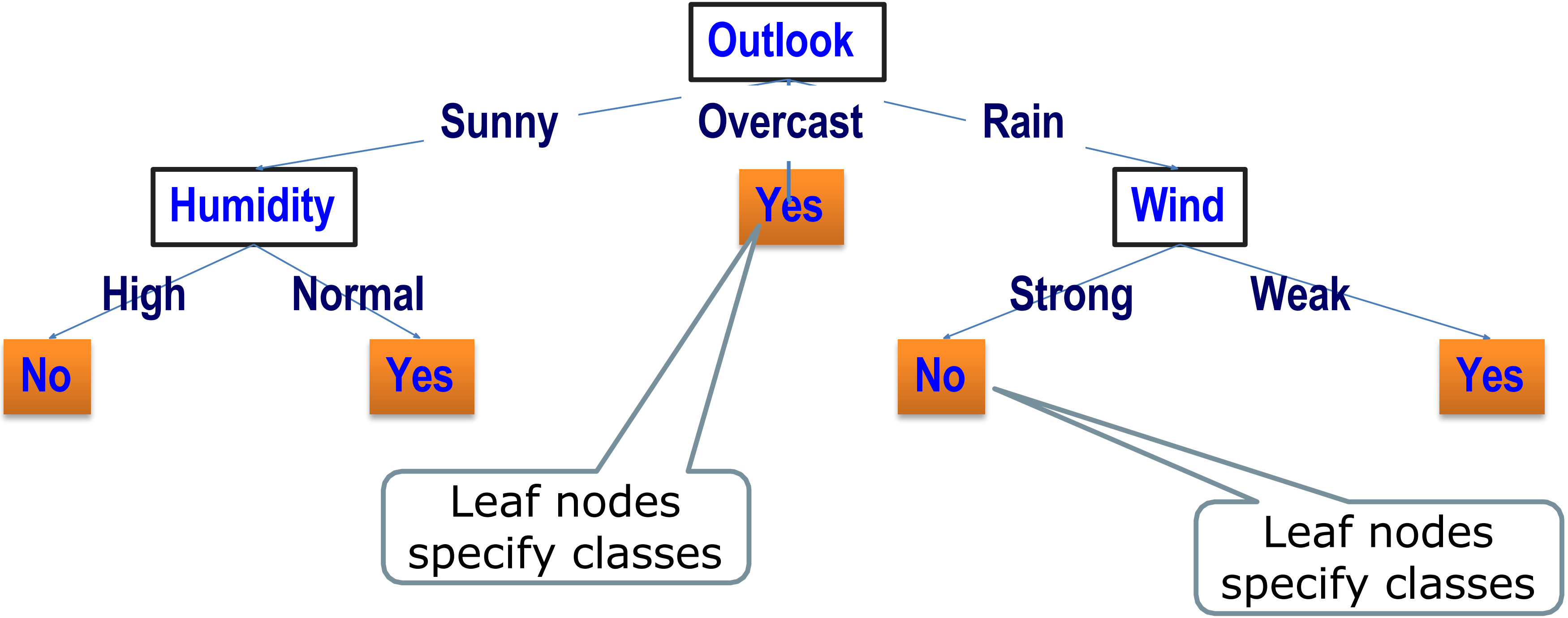
Decision Trees

- Outlook = (sunny, overcast, rain)



Decision Trees

- Class = (Yes, No)



Example

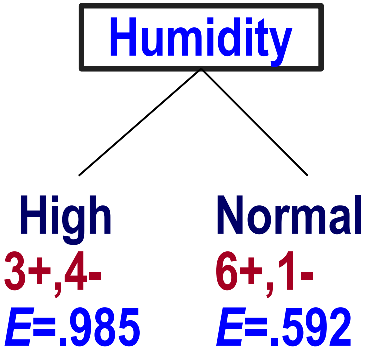
Play Tennis Example

$$\textbf{Entropy(S)} = -\tfrac{9}{14}\log\left(\tfrac{9}{14}\right) - \tfrac{5}{14}\log\left(\tfrac{5}{14}\right) = 0.94$$

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

Example

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

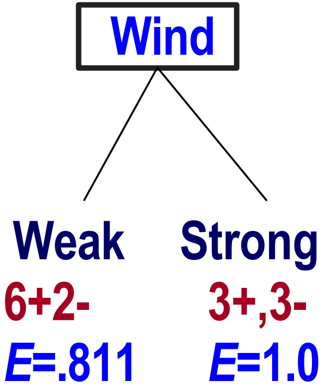


Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Gain(S, Humidity) = .94 - 7/14 * 0.985 - 7/14 *.592 = 0.151

Example

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Gain(S, Wind) = .94 - 8/14 * 0.811 - 6/14 * 1.0 = 0.048

Example

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Outlook

Sunny	Overcast	Rain
2+,3-	4+,0-	3+,2-
0.970	0.0	0.970

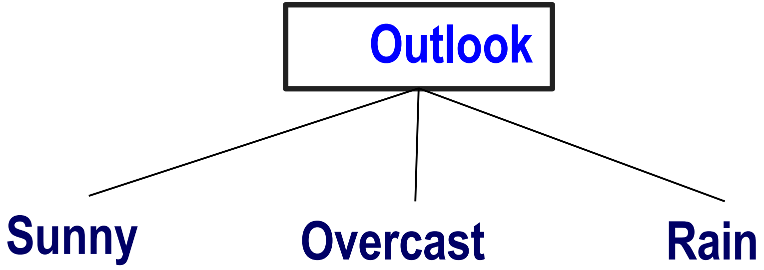
***Gain(S, Outlook)* = 0.246**

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

Example

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

Pick Outlook as the root



Gain(S, Humidity) = 0.151

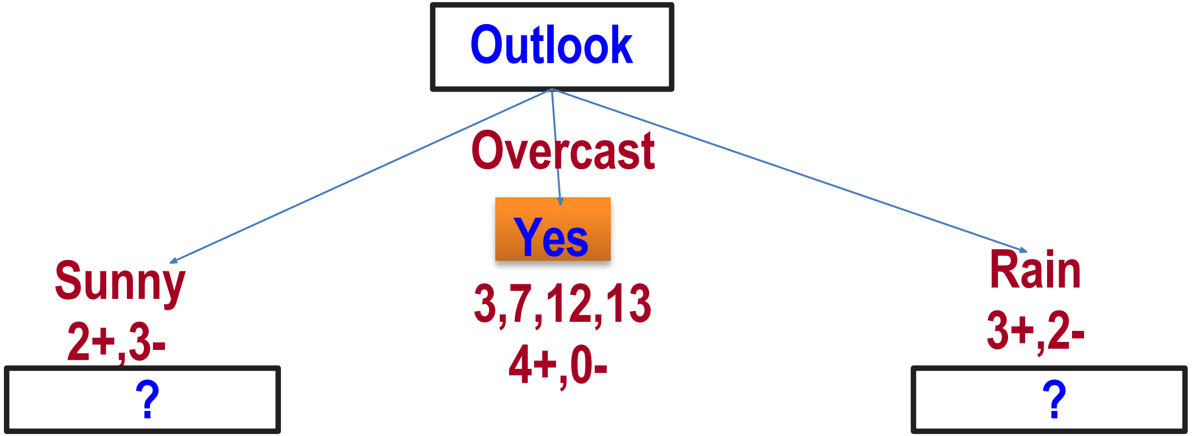
Gain(S, Wind) = 0.048

Gain(S, Temperature) = 0.029

Gain(S, Outlook) = 0.246

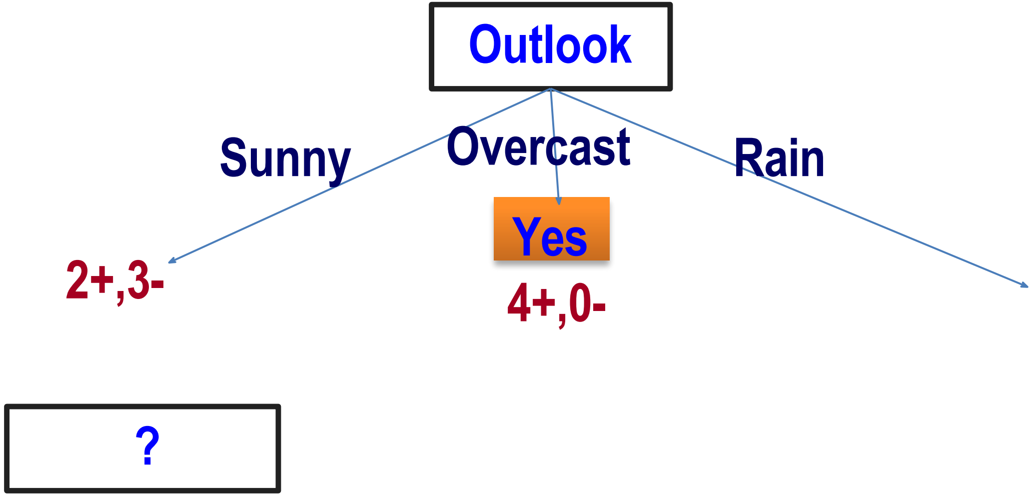
Example

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*



Continue until: Every attribute is included in path, or, all examples in the leaf have same label

Example

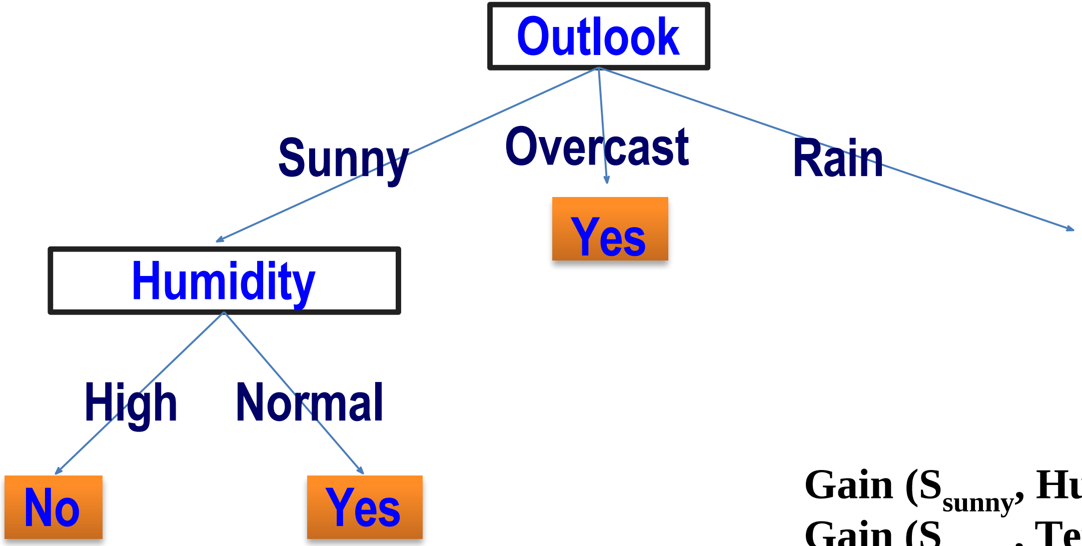


Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

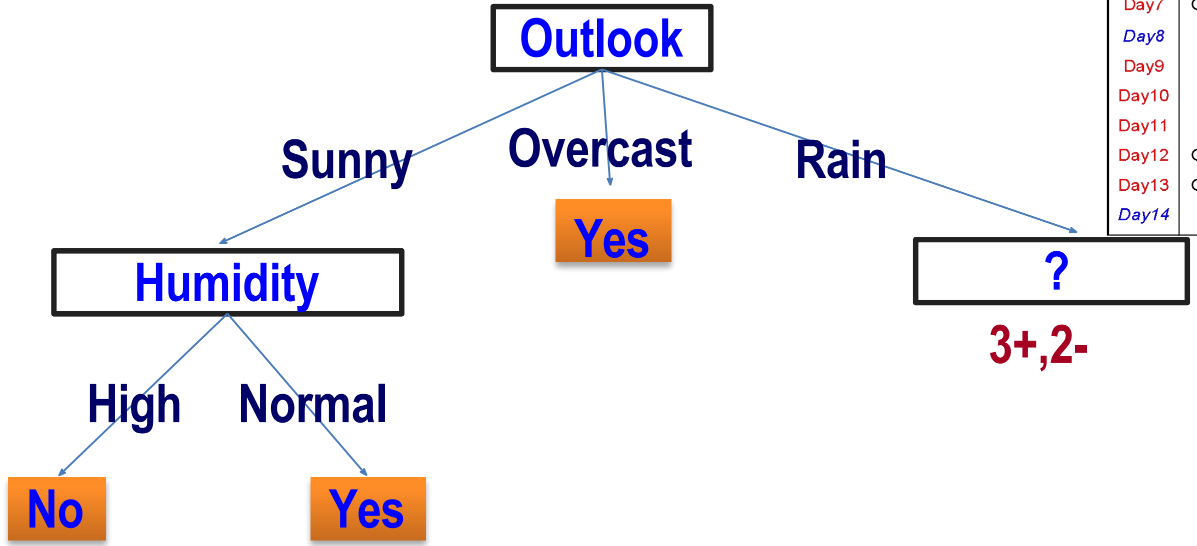
Gain (S_{sunny}, Humidity) = .97-(3/5) * 0-(2/5) * 0 = .97
Gain (S_{sunny}, Temp) = .97- 0-(2/5) *1 = .57
Gain (S_{sunny}, Wind) = .97-(2/5) *1 - (3/5) *.92 = .02

Example



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

Gain (S_{sunny}, Humidity) = .97-(3/5) * 0-(2/5) * 0 = .97
Gain (S_{sunny}, Temp) = .97- 0-(2/5) *1 = .57
Gain (S_{sunny}, Wind) = .97-(2/5) *1 - (3/5) *.92 = .02

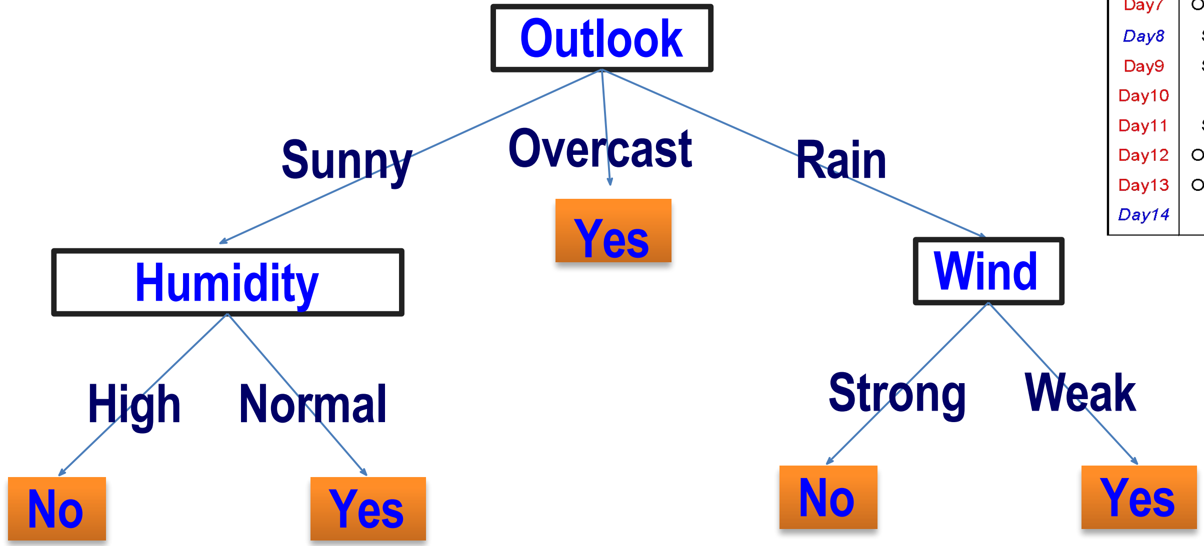


Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Gain (S_{rain}, Humidity) =
Gain (S_{rain}, Temp) =
Gain (S_{rain}, Wind) =

Example

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*



Thank You