DataInquest

Big Data Technologies and Data Science training for all!!

Data Science

Topics for Today

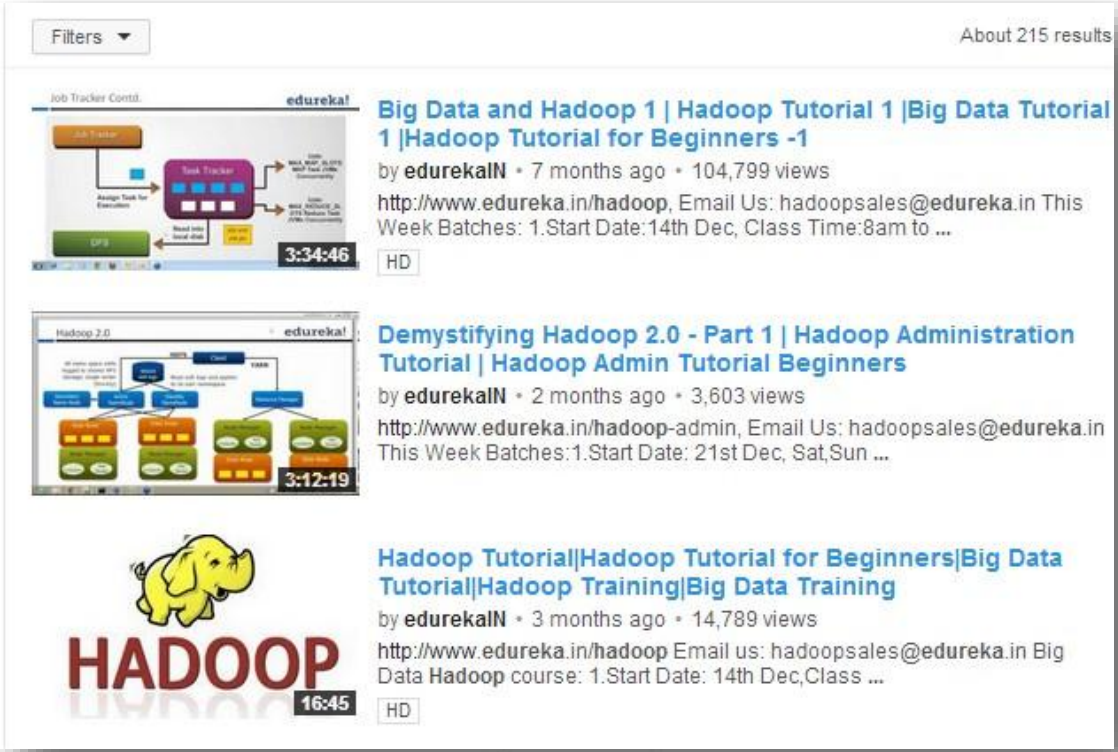
- ✓ Mahout Overview
- ✓ ML Common Use Cases
- ✓ Algorithms in Mahout
- ✓ Understanding Supervised and Unsupervised Learning Techniques
- ✓ What is clustering
- ✓ Similarity Metrics
- ✓ Distance Measure Types: Euclidean, Manhatten, Cosine Measures
- ✓ Creating predictive models
- ✓ Understanding K-Means Clustering
- ✓ Understanding TF-IDF and Cosine Similarity and their application to Vector Space Model
- ✓ Implementing Association rule mining in R
- ✓ Application of Technique on smaller datasets for better understanding using R software

Machine Learning Use Cases – You Tube

YouTube utilizes recommendation systems to bring videos to a user that it believes the user will be interested in.

They are designed to:

- ✓ Increase the numbers of videos the user will watch
- ✓ Increase the length of time he spends on the site, and
- ✓ Maximize the enjoyment of his YouTube experience.



Use Case – You Tube (Contd.)

User Activity:

In order to obtain personalized recommendations, YouTube's recommendation system combines the related videos association rules with the user's personal activity on the site.

This includes several factors:

- ✓ There are the videos that were watched - along with a certain threshold, say by a certain date. After all, you don't want to count videos watched from 2 years ago if the user has watched enough videos, most likely.
- ✓ Also, YouTube factors in with emphasis any videos that were explicitly "liked", added to favourites, given a rating, added to a playlist. The union of these videos is known as the seed set.
- ✓ Then, to compute the candidate recommendations for a seed set, YouTube expands it along the related videos.

Use Case – Wine Recommendation

Use Case – Wine Recommendation (Contd)

What wine will I enjoy? More than 2 million consumers turn to the Internet for the answer to this question every day

Problem

- Mysterious ratings and adjective-based reviews do little to help consumers decide which wine to buy
- They can't even agree amongst themselves

Solution

- Next Glass solves this problem by removing subjectivity and applying science to deliver recommendations based on your previous ratings

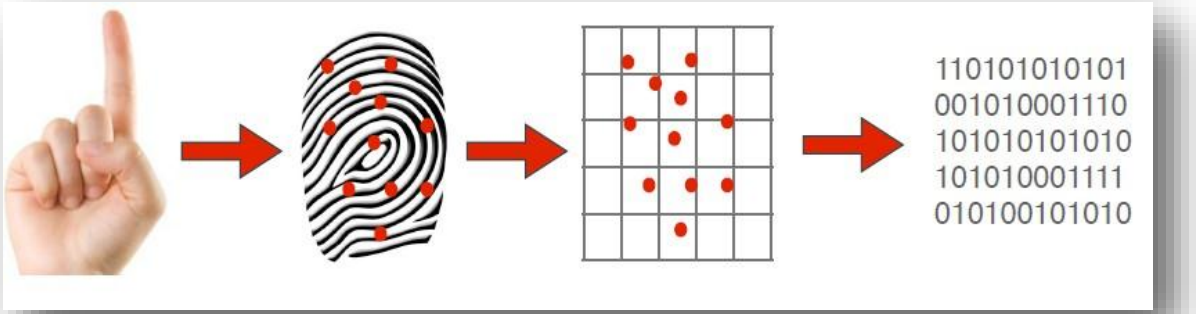
Use Case - Biometrics

Biometrics : The Science of establishing the identity of an individual based on the physical, chemical or behavioral attributes of the person.

Why is it Important?

- ✓ Identify Individual credentials
- ✓ Identify and prevent banking fraud
- ✓ Enforcement of law and security

How Does a Fingerprint Optical Scanner Work?



A fingerprint scanner system has two basic jobs

- ✓ Get an image of your finger
- ✓ Determine whether the pattern of ridges and valleys in this image matches the pattern of ridges and valleys in pre-scanned images

Process:

- ✓ Only specific characteristics, which are unique to every fingerprint, are filtered are saved as an encrypted biometric key or mathematical representation.
- ✓ No image of a fingerprint is ever saved, only a series of numbers (a binary code), which is used for verification. The algorithm cannot be reconverted to an image, so no one can duplicate your fingerprints

Use Case – Aadhaar

India is reportedly creating a biometric database to hold the fingerprints and face images for each of 1.2 Billion citizens as part of its Unique Identification Project.

Use Case – Paycheck Secure System

All Trust Network Paycheck Secure System has enrolled over 6 Million users and over 70 Million Transactions.

What are we going to learn today?

- ✓ What is Learning?
- ✓ Can a Machine learn?
- ✓ How to do it ?

Mahout : Scalable Machine learning Library

Machine Learning is Programming Computers to optimize Performance Criterion using Example Data or Past experience.

- ✓ A branch of artificial intelligence
- ✓ Systems that learn from data
- ✓ Classify data after learning
- ✓ Learn on test data sets
- ✓ Generalisation – the ability to classify unseen data sets

Machine learning

- ✓ Machine Learning is a class of algorithms which is data-driven, i.e. unlike "normal" algorithms it is the data that "tells" what the "good answer" is.

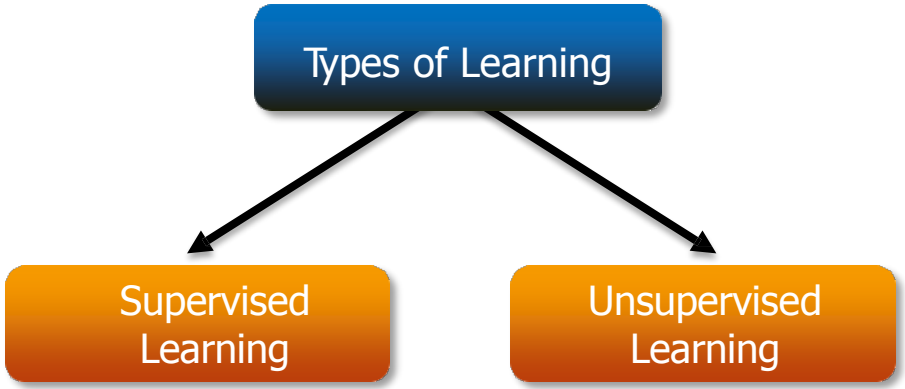
- ✓ **Example:**

  An hypothetical non-machine learning algorithm for face recognition in images would try to define what a face is (round skin-like-colored disk, with dark area where you expect the eyes etc).

  A machine learning algorithm would not have such coded definition, but will "learn-by-examples": you'll show several images of faces and not-faces and a good algorithm will eventually learn and be able to predict whether or not an unseen image is a face.

Learning Techniques

Attain knowledge by study, experience, or by being taught.

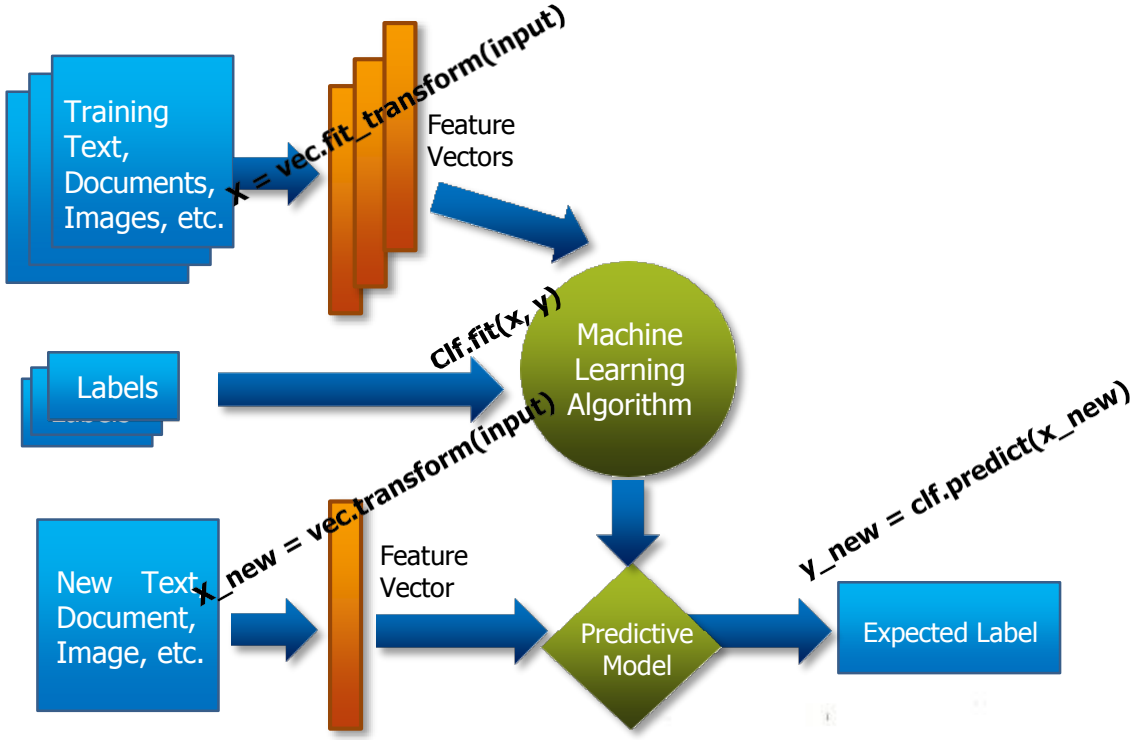


Supervised Learning

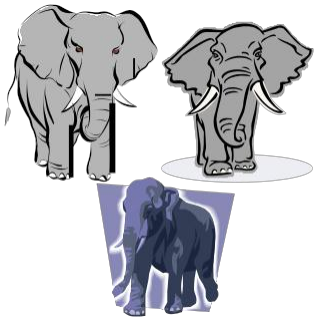
Supervised learning : Training data includes both the input and the desired results.

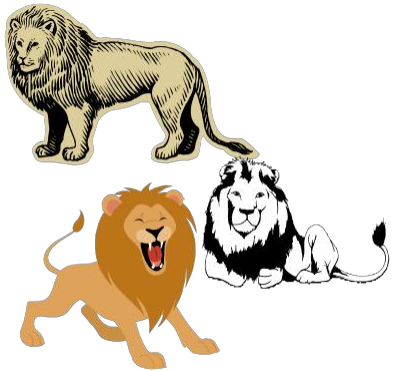
- ✓ For some examples, the correct results (targets) are known and are given in input to the model during the learning process.
- ✓ The construction of a proper training, validation and test set (Bok) is crucial.
- ✓ These methods are usually fast and accurate.
- ✓ Have to be able to generalize: give the correct results when new data are given in input without knowing a priori the target.

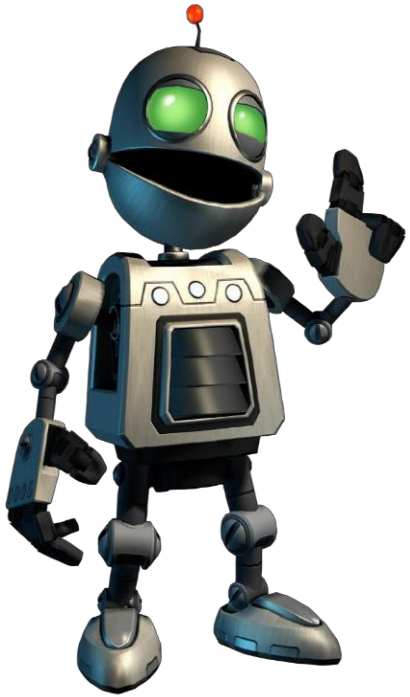
Example – Supervised Learning Model



Supervised Learning





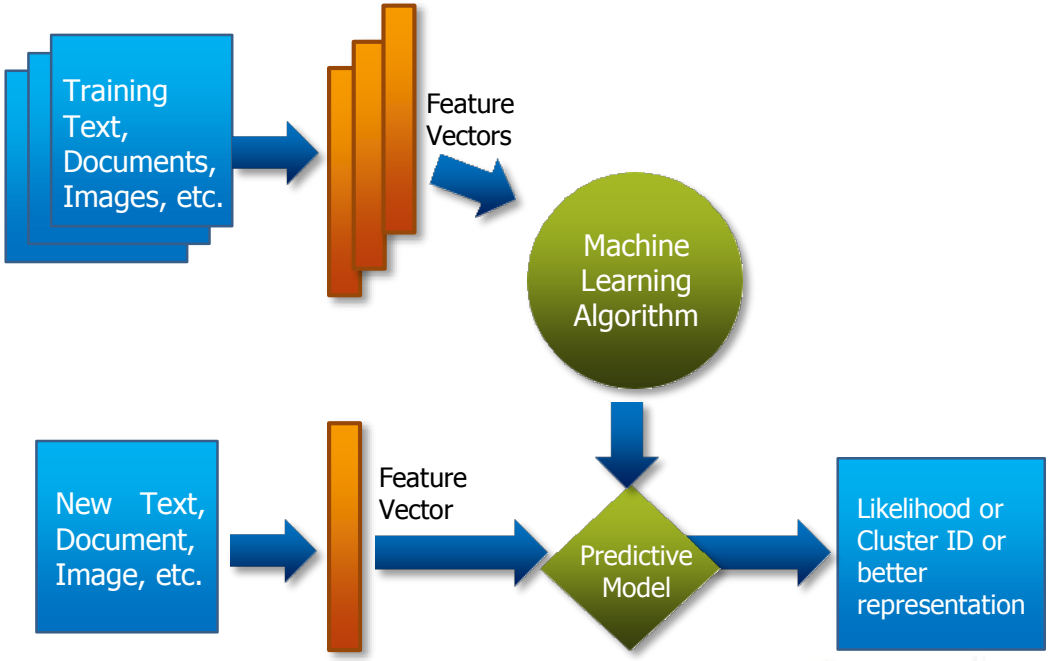


Unsupervised learning

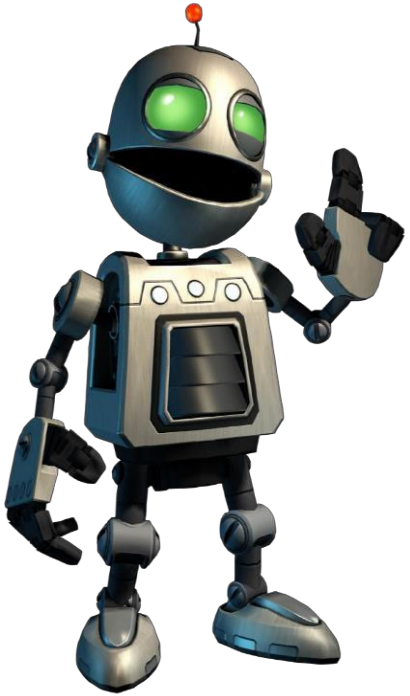
Unsupervised Learning:

- ✓ The model is not provided with the correct results during the training.
- ✓ Can be used to cluster the input data in classes on the basis of their statistical properties only Cluster significance and labeling.
- ✓ The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

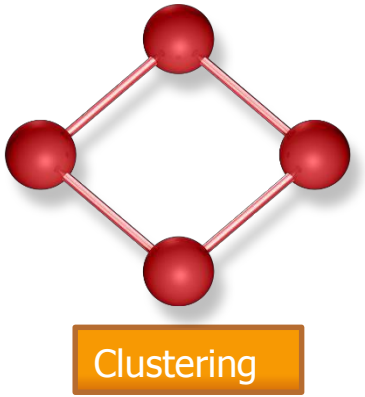
Example – Unsupervised Learning Model



Examples



What is Clustering?



Organizing data into *clusters* such that there is:

- ✓ High intra-cluster similarity
- ✓ Low inter-cluster similarity
- ✓ Informally, finding natural groupings among objects.

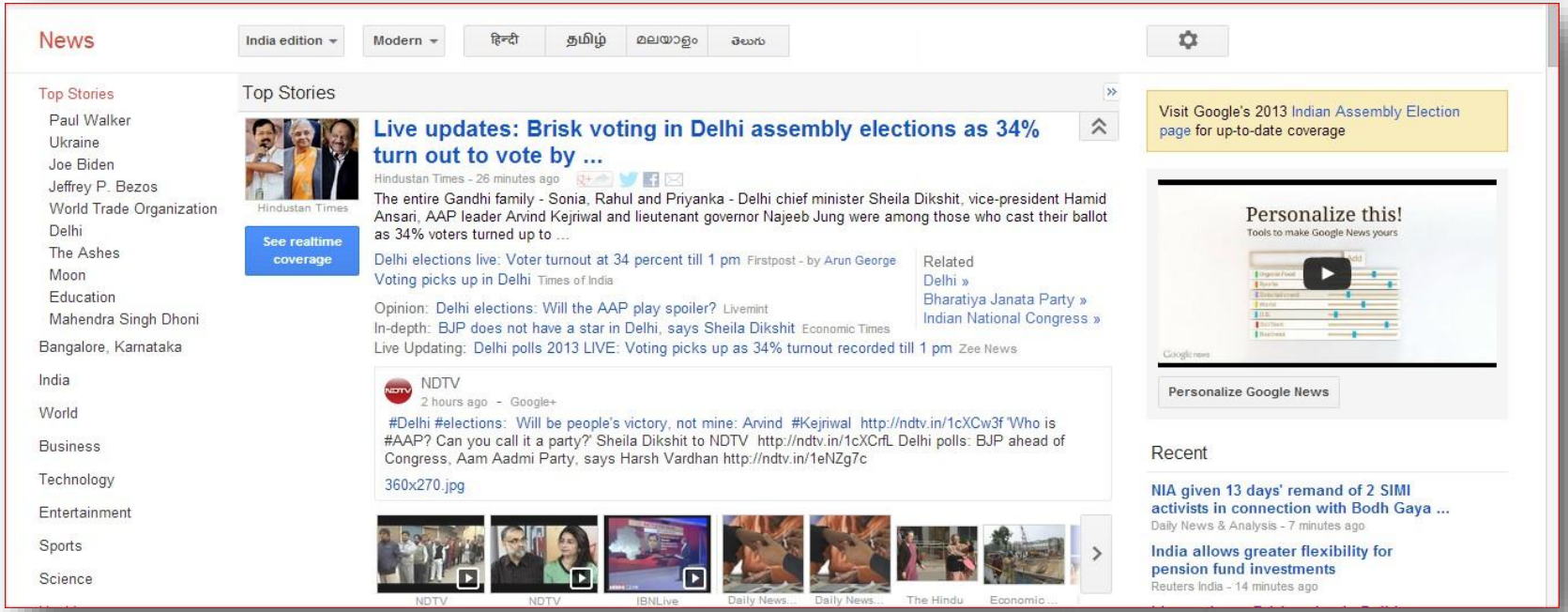
Why do we want to do it??

Why Clustering?

- ✓ Organizing data into clusters shows internal structure of the data
  Ex. Clusty and clustering genes
- ✓ Sometimes the partitioning is the goal
  Ex. Market segmentation
- ✓ Prepare for other AI techniques
  Ex. Summarize news (cluster and then find centroid)
- ✓ Techniques for clustering is useful in knowledge
- ✓ Discovery in data
  Ex. Underlying rules, reoccurring patterns, topics, etc.

Clustering - Example

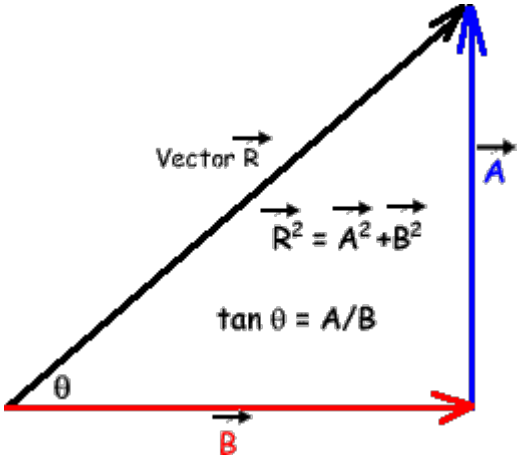
A sample news grouping from Google News:

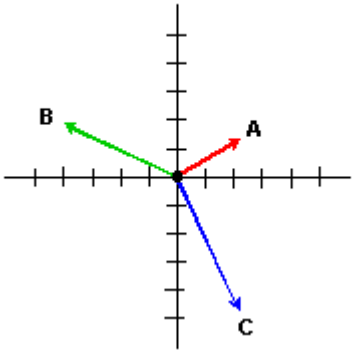


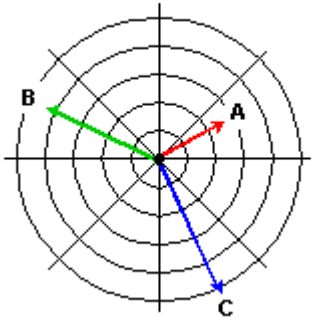
Vector

A **vector** is a quantity or phenomenon that has two independent properties: magnitude and direction.

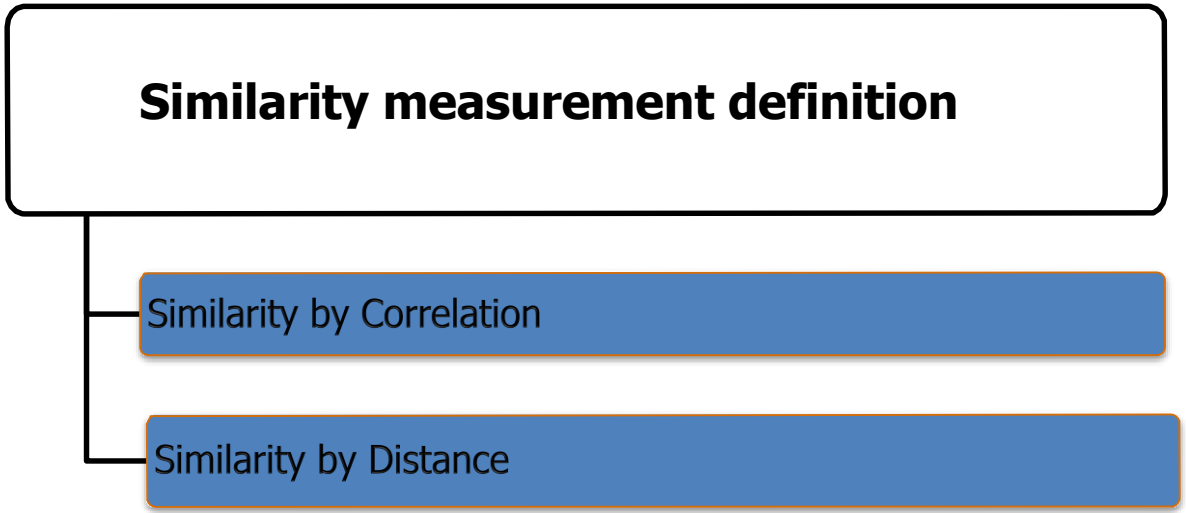
The term also denotes the mathematical or geometrical representation of such a quantity.



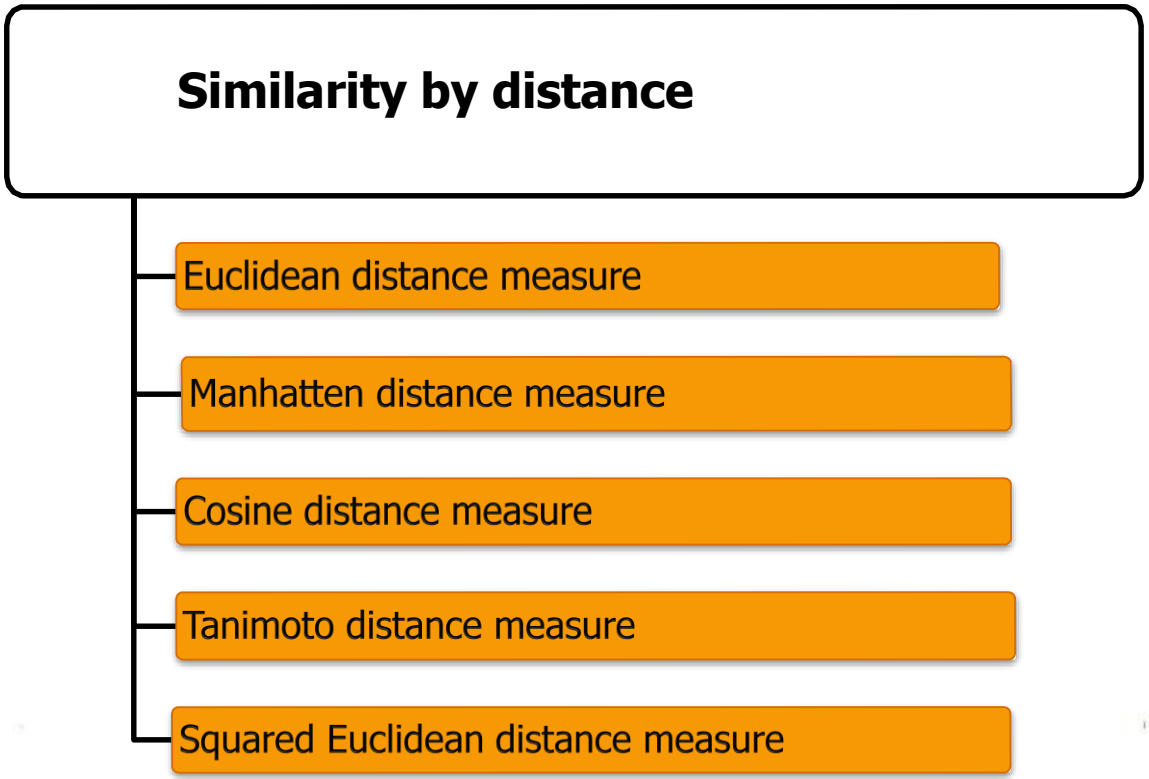




Similarity Measurement



Distance measures



Euclidean distance measure

Mathematically, Euclidean distance between two n-dimensional vectors

(a1, a2, ... , an) and (b1,b2,...,bn) is:

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

Squared Euclidean distance measure

For *n*-dimensional vectors (a1, a2, ... , an) and (b1, b2, ... ,bn) the distance becomes:

$$d = (a1 - b1)2 + (a2 - b2)2 + ... + (an - bn)2$$

Manhatten distance measure

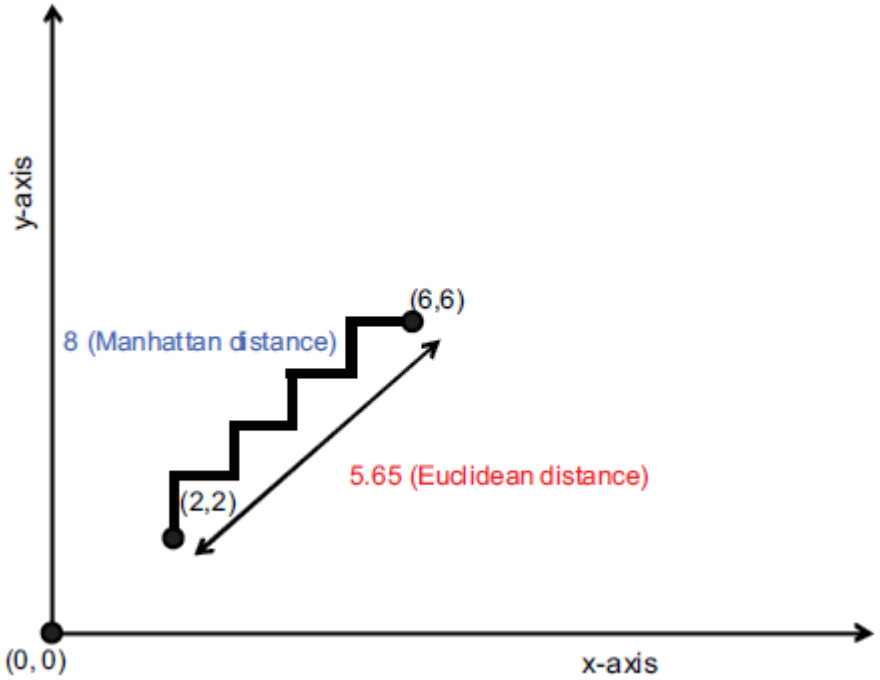
Mathematically, the Manhattan distance between two n-dimensional vectors

(a1, a2, ... , an) and (b1, b2, ... , bn) is

$$d = |a1 - b1| + |a2 - b2| + ... + |an - bn|$$

Difference between Euclidean and Manhattan

From this image we can say that, The Euclidean distance measure gives 5.65 as the distance between (2, 2) and (6, 6) whereas the Manhattan distance is 8.0



Cosine distance measure

The formula for the cosine distance between *n*-dimensional vectors (a1, a2, ... , an) and (b1, b2, ...,bn) is

$$d = 1 - \frac{(a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}{(\sqrt{(a_1^{\,2} + a_2^{\,2} + \ldots + a_n^{\,2})}\sqrt{(b_1^{\,2} + b_2^{\,2} + \ldots + b_n^{\,2})})}$$

Tanimoto distance measure

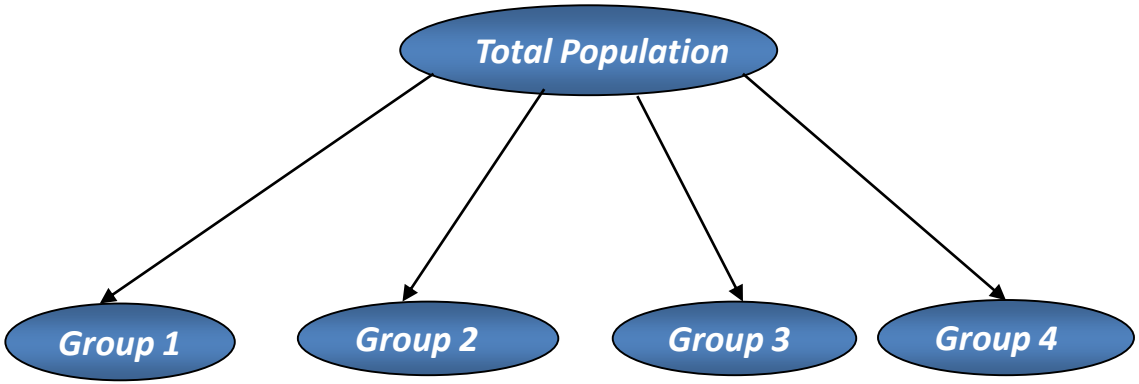
The formula for the Tanimoto distance between two *n*-dimensional vectors (a1, a2, ... , an) and (b1, b2, ... , bn) is

$$d = 1 - \frac{(a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}{\sqrt{(a_1^2 + a_2^2 + \ldots + a_n^2)} + \sqrt{(b_1^2 + b_2^2 + \ldots + b_n^2)} - (a_1 b_1 + a_2 b_2 + \ldots + a_n b_n)}$$

K-Means clustering

K-Means clustering

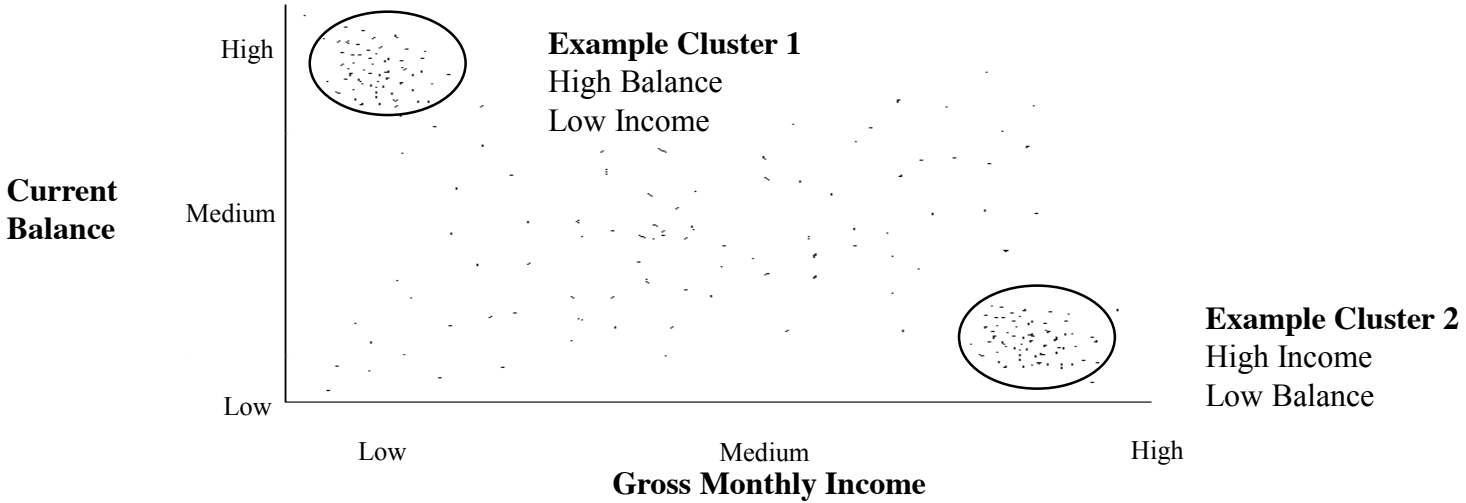
- ✓ The process by which objects are classified into a number of groups so **that they are as much dissimilar as possible from one group to another group, but as much similar as possible within each group.**
- ✓ In other words Cluster analysis means dividing the whole population into groups which are distinct between themselves but internally similar.



- ✓ The objects in group 1 should be as similar as possible.
- ✓ But there should be much difference between an object in group 1 and group 2.
- ✓ The attributes of the objects are allowed to determine which objects should be grouped together.

K-Means clustering

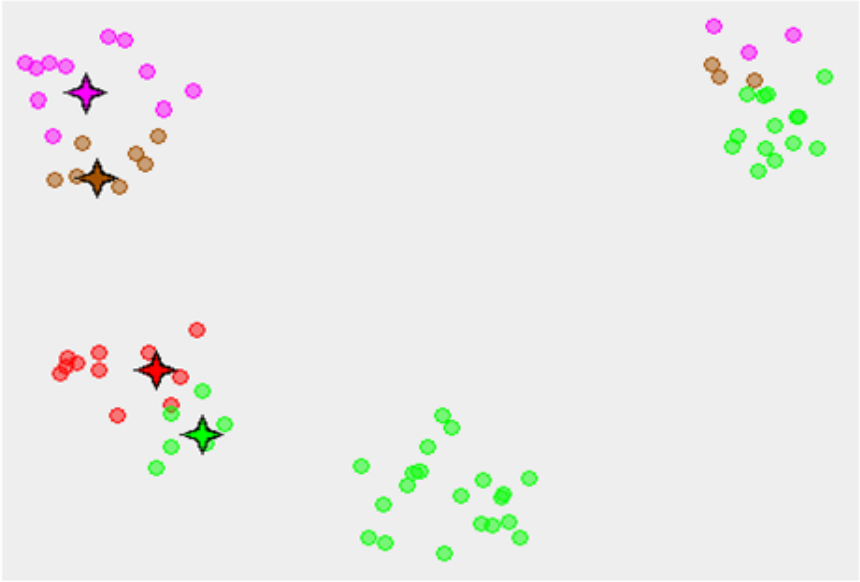
Basic Concepts of Cluster Analysis using Two variables



- ✓ Cluster 1 and Cluster 2 are being differentiated by Income and Current Balance.
- ✓ The objects in Cluster 1 have similar characteristics (High Income and Low balance), on the other hand the objects in Cluster 2 have the same characteristic (High Balance and Low Income).
- ✓ But there are much differences between an object in Cluster 1 and an object in Cluster 2

K-Means clustering steps

1. k initial "means" (in this case k=3) are randomly generated within the data domain.
2. k clusters are created by associating every observation with the nearest mean.
3. The centroid of each of the k clusters becomes the new mean.
4. Steps 2 and 3 are repeated until convergence has been reached.



Step by Step pictorial representation of K-Means clustering

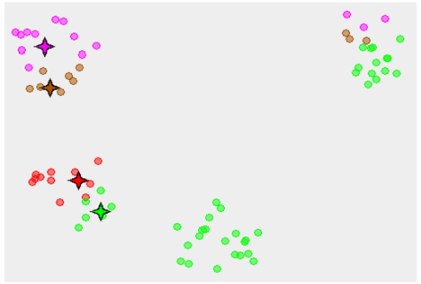


figure-1

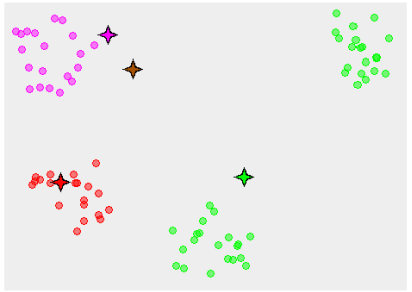


figure-2

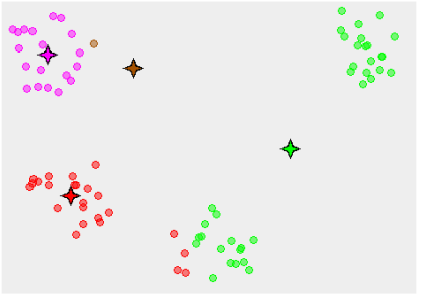


figure-3

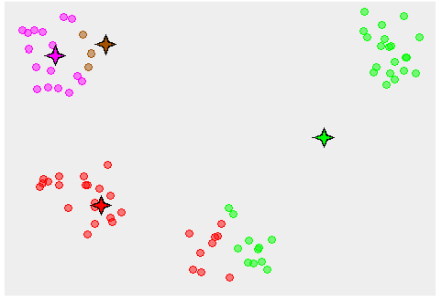


figure-4

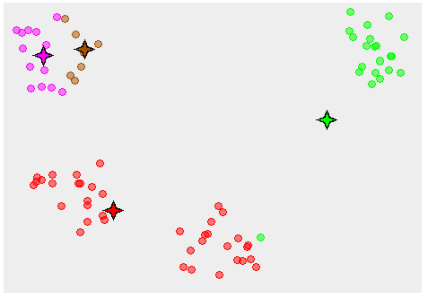


figure-5

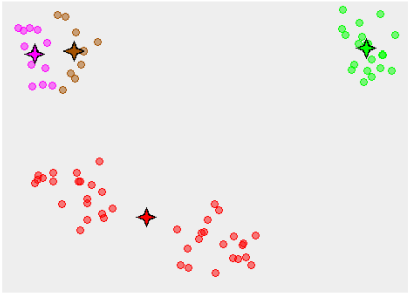


figure-6

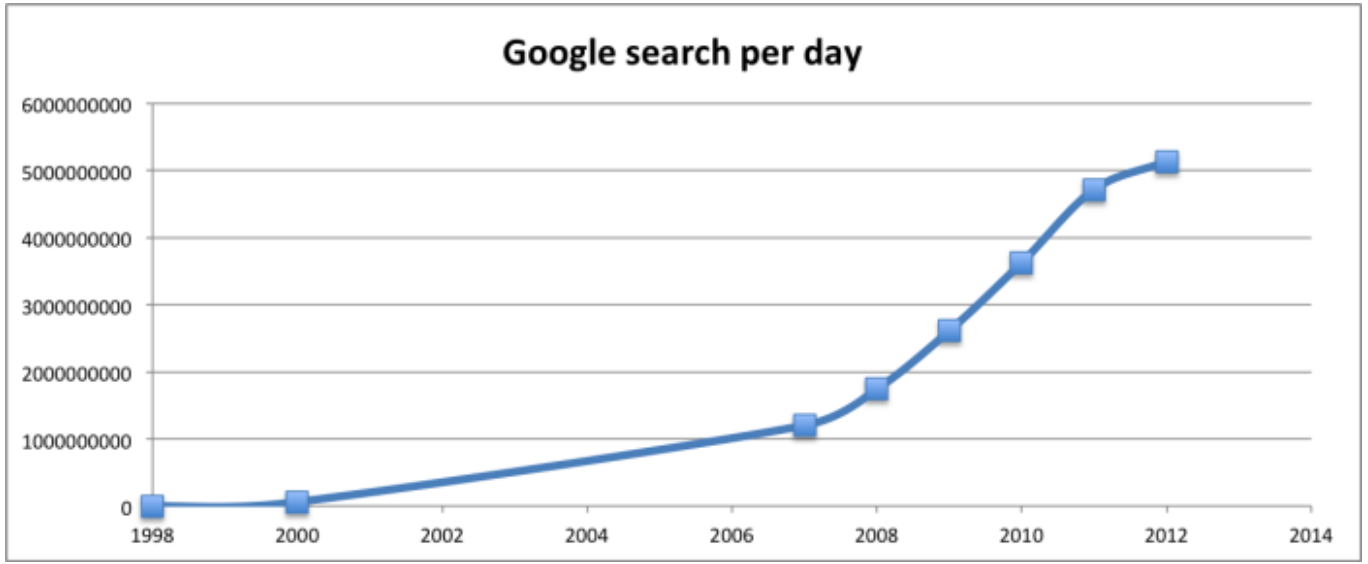
- ✓ The small circles are the data points, the four ray stars are the centroids (means).
- ✓ The initial configuration is on the figure-1.
- ✓ The algorithm converges after five iterations presented on the figures, from figure-2 to figure-6.

Tf-Idf and Cosine similarity

Google Search

- In the year **1998** Google handled **9800** average search queries every day.
- In **2012** this number shot up to **5.13 billion** average searches per day.

The graph given below shows this astronomical growth.

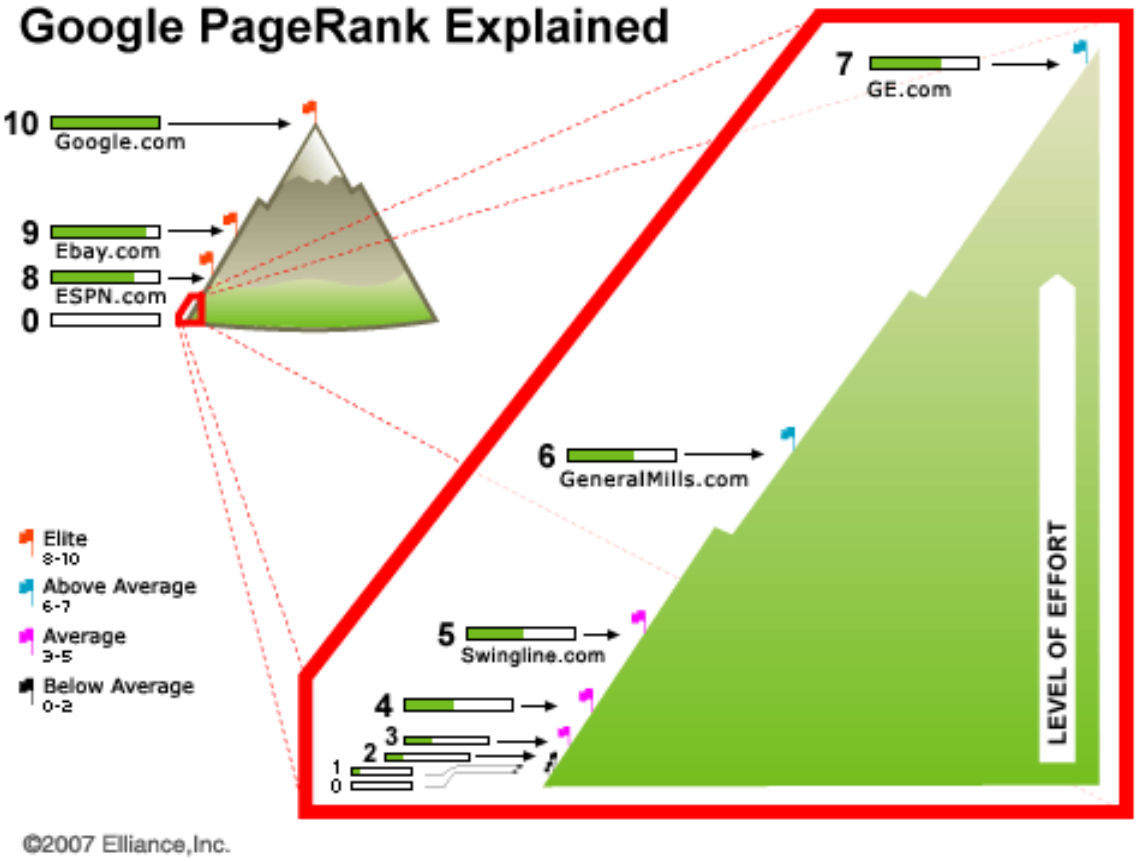


Google's PageRank Algorithm

The major reason for google's success is because of its **PageRank algorithm.**

- **PageRank** determines how trustworthy and reputable a given website is. But there is also another part.

The input query entered by the user should be used to match the relevant documents and score them. Here we will focus on the second part.



Google Search- How it works?

Let's consider 3 documents to show how this works. Take some time to go through them.

- **Document 1:** The game of life is a game of everlasting learning
- **Document 2:** The unexamined life is not worth living
- **Document 3:** Never stop learning

Let us imagine that you are doing a search on these documents with the following query:

"life learning"

The query is a **free text query.**

It means a query in which the terms of the query are typed freeform into the search interface, without any connecting search operators.

Step 1: Term Frequency (TF)

Term Frequency also known as TF measures the number of times a term (word) occurs in a document.

Given below are the terms and their frequency on each of the document.

TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Term Frequency	1	2	2	1	1	1	1	1

TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Term Frequency	1	1	1	1	1	1	1

TF for Document 3

Document3	never	stop	learning
Term Frequency	1	1	1

Step 1 (continued): Normalized Term Frequency (TF)

- In reality each document will be of different size.
- On a large document the frequency of the terms will be much higher than the smaller ones.
- Hence we need to normalize the document based on its size.
- A simple trick is to divide the term frequency by the total number of terms.

e.g:

In Document 1 the term game occurs two times.

The total number of terms in the document is 10.

Hence the normalized term frequency is 2 / 10 = 0.2.

Step 1 (continued): Normalized Term Frequency (TF)

Given below are the normalized term frequency for all the documents.

Normalized TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Normalized TF	0.1	0.2	0.2	0.1	0.1	0.1	0.1	0.1

Normalized TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Normalized TF	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857

Normalized TF for Document 3

Document3	never	stop	learning
Normalized TF	0.333333	0.333333	0.333333

Step 2: Inverse Document Frequency (IDF)

The main purpose of doing a search is to find out relevant documents matching the query.
In the first step all terms are considered equally important.

- Certain terms that **occur too frequently have little power** in determining the relevance.
- Solution: **Weigh down the effects of too frequently occurring terms.**

- The terms that **occur less in the document can be more relevant**.
- Solution: **Weigh up the effects of less frequently occurring terms.**

Logarithms helps us to solve this problem.
Given below is the IDF for terms occurring in all the documents. Since the terms: the, life, is, learning occurs in 2 out of 3 documents they have a lower score compared to the other terms that appear in only one document.

Computing IDF for the term **game:**

IDF(**game**) = 1 + $\log_e$(Total Number Of Documents / Number Of Documents with term **game** in it)

There are 3 documents in all = Document1, Document2, Document3

The term game appears in Document1

IDF(**game**) = 1 + $\log_e$(3 / 1)
= 1 + 1.098726209
= 2.098726209

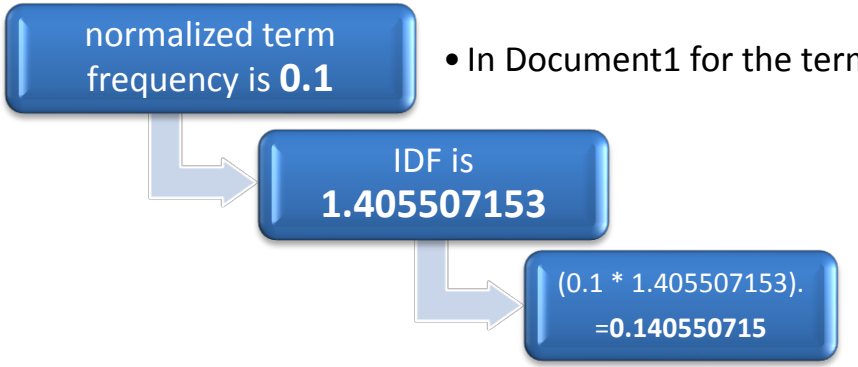
Step 2 (continued): Inverse Document Frequency (IDF)

Terms	IDF
the	1.405507153
game	2.098726209
of	2.098726209
life	1.405507153
is	1.405507153
a	2.098726209
everlasting	2.098726209
learning	1.405507153
unexamined	2.098726209
not	2.098726209
worth	2.098726209
living	2.098726209
never	2.098726209
stop	2.098726209

Step 3: TF * IDF

Remember we are trying to find out relevant documents for the query: **life learning**

- For each term in the query multiply its normalized term frequency with its IDF on each document.



- In Document1 for the term **life**
- Multiplying Frequency and IDF

- Given below is TF * IDF calculations for **life and learning** in all the documents.

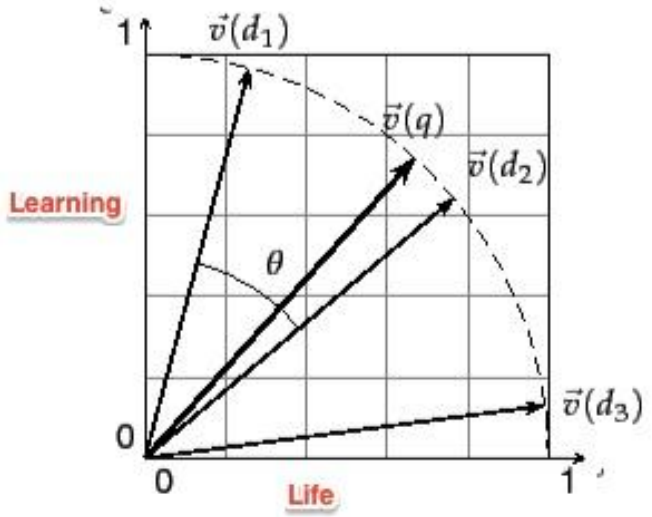
	Document1	Document2	Document3
life	0.140550715	0.200786736	0
learning	0.140550715	0	0.468502384

Step 4 (Continued): Vector Space Model – Cosine Similarity

- From each document we derive a vector.
- The set of documents in a collection then is viewed as a set of vectors in a vector space. Each term will have its own axis.
- Using the formula given below we can find out the similarity between any two documents.

```
Cosine Similarity (d1, d2) =  Dot product(d1, d2) / ||d1|| * ||d2||

Dot product (d1,d2) = d1[0] * d2[0] + d1[1] * d2[1] * … * d1[n] * d2[n]
||d1|| = square root(d1[0]² + d1[1]² + ... + d1[n]²)
||d2|| = square root(d2[0]² + d2[1]² + ... + d2[n]²)
```



- Vectors deals only with numbers. In this example we are dealing with text documents.
- We used **TF and IDF to convert text into numbers so that it can be represented by a vector.**

Step 4 (Continued): Vector Space Model – Cosine Similarity

The query entered by the user can also be represented as a vector.
We will calculate the TF*IDF for the query

	TF	IDF	TF*IDF
life	0.5	1.405507153	0.702753576
learning	0.5	1.405507153	0.702753576

Note:

The cosine value is always between –1 and 1: the cosine of a small angle is near 1, and the cosine of a large angle near 180 degrees is close to –1. This is good, because small angles should map to high similarity, near 1, and large angles should map to near –1.

Step 4 (Continued): Vector Space Model – Cosine Similarity

Let us now calculate the cosine similarity of the query and Document1.

$$\text{Cosine Similarity(Query,Document1)} = \text{Dot product(Query, Document1)} / ||\text{Query}|| * ||\text{Document1}||$$

$$\text{Dot product(Query, Document1)} = ((0.702753576) * (0.140550715) + (0.702753576)*(0.140550715)) = 0.197545035151$$

$$||\text{Query}|| = \text{sqrt}((0.702753576)^2 + (0.702753576)^2) = 0.993843638185$$

$$||\text{Document1}|| = \text{sqrt}((0.140550715)^2 + (0.140550715)^2) = 0.198768727354$$

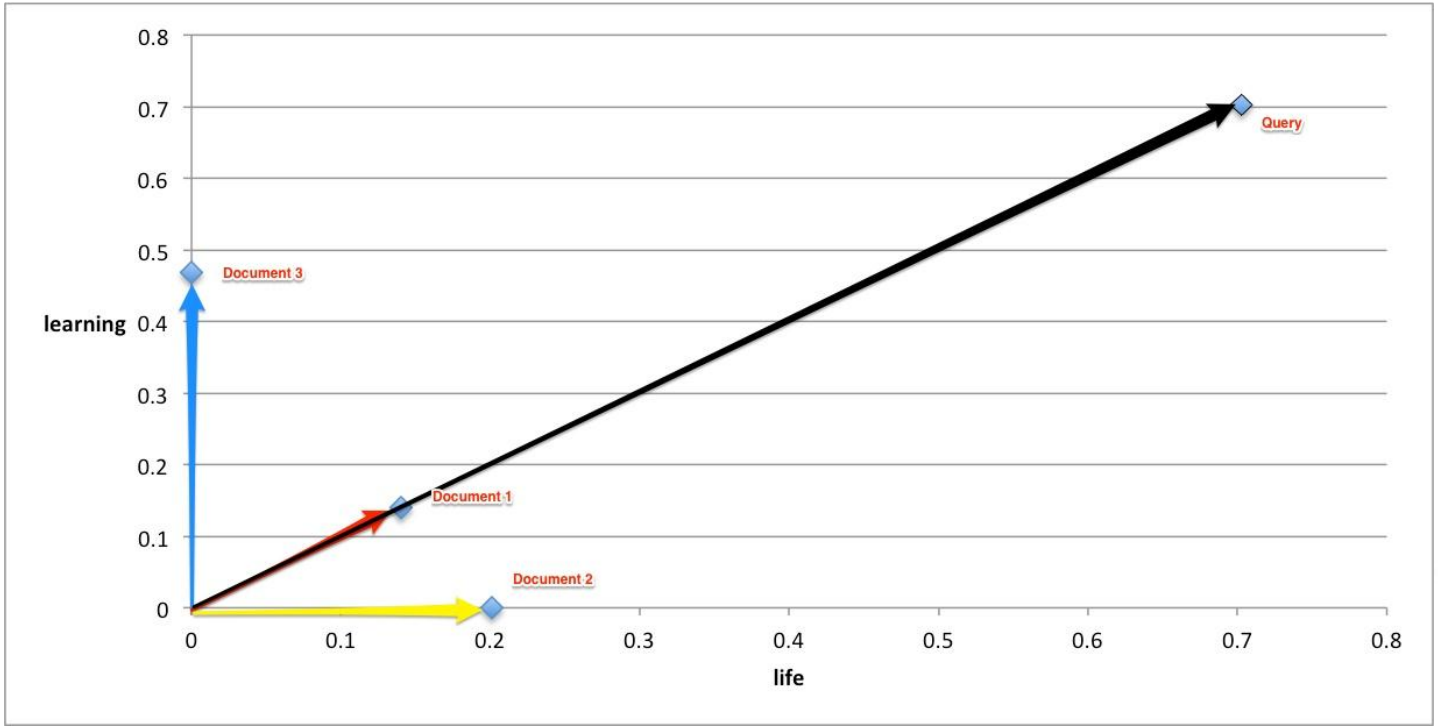
$$\text{Cosine Similarity(Query, Document)} = 0.197545035151 / (0.993843638185) * (0.198768727354) = 0.197545035151 / 0.197545035151 = 1$$

Given below is the similarity scores for all the documents and the query

	Document1	Document2	Document3
Cosine Similarity	1	0.707106781	0.707106781

Step 4 (Continued): Vector Space Model – Cosine Similarity

- ✓ Below is the plot of vector values for the query and documents in 2-dimensional space of life and learning.
- ✓ Document1 has the highest score of 1.
- ✓ This is not surprising as it has both the terms **life** and **learning**.



Association Rule Mining

Association Rule Mining

- In data mining, **association rule learning** is a popular and well researched method for discovering interesting relations between variables in large databases.
- It is intended to identify strong rules discovered in databases using different measures of interests.
- The rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat.
- Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

Sample Input Data

transaction_id	items
1	citrus fruit
1	semi-finished bread
1	margarine
1	ready soups
2	tropical fruit
2	yogurt
2	coffee
3	whole milk
4	pip fruit
4	yogurt
4	cream cheese
4	meat spreads
5	other vegetables
5	whole milk

Association Rule Mining

	lhs	rhs	support	confidence	lift
89	Hard cheese	Whole milk	0.01006609	0.41078838	1.6076815
90	Whole milk	Hard cheese	0.01006609	0.03939515	1.6076815
91	Butter milk	Other vegetables	0.01037112	0.37090909	1.9169159
92	Other vegetables	Butter milk	0.01037112	0.05359958	1.9169159
93	Butter milk	Whole milk	0.01159126	0.41454545	1.6223854
94	Whole milk	Butter milk	0.01159126	0.04536411	1.6223854
95	ham	Whole milk	0.01148958	0.44140625	1.7275091
96	Whole milk	ham	0.01148958	0.04496618	1.7275091
97	Sliced cheese	Whole milk	0.01077783	0.43983402	1.7213560
98	Whole milk	Sliced cheese	0.01077783	0.04218066	1.7213560
99	oil	Whole milk	0.01128622	0.40217391	1.5739675
100	Whole milk	Oil	0.01128622	0.04417031	1.5739675
101	onions	Other vegetables	0.01423488	0.45901639	2.3722681
102	Other vegetables	Onions	0.01423488	0.07356805	2.3722681
103	onions	Whole milk	0.01209964	0.39016393	1.5269647
104	Whole milk	Onions	0.01209964	0.04735376	1.5269647
105	berries	yogurt	0.01057448	0.31804281	2.2798477

Association Rule Mining - Concepts

Constraints on below measures are used to select useful and best rules of all the rules given by R
After analyzing these values for all the rules, best rules for WB have been obtained.

Support

- The support Supp(X)=proportion of transactions in the data set which contain the interest.

Confidence

- The confidence of a rule:
  Conf(x=>y)= Supp(X U Y)/Supp(X)

Lift

- The lift of a rule: Lift(X=>Y)=
  Supp(X U Y)

  (Supp(X) X Supp(Y))

E.g.:- Consider rule: {Jack the Ripper (1988)} => {Strawberry Blonde}
Let Jack the Ripper =X and Strawberry Blonde =Y, Then

Support(X U Y)= No of transactions involving both Jack the Ripper and Strawberry Blonde/ Total no of transactions

Confidence= No of transactions where Strawberry Blonde was also bought when Jack the Ripper was bought/ No of transactions where Jack the Ripper was bought

Lift = Ratio of observed support to the expected support

Association Rule Mining - Concepts

Association rule generation is usually split up into two separate steps:

Step #1:

Minimum support is applied to find all frequent itemsets in a database.

Step #2:

These frequent itemsets and the minimum confidence constraint are used to form rules.

Association Rule Mining-Single Cardinality

S No.	Rules	Support	Confidence	Lift
1	{Strawberry Blonde} => {Canterville Ghost}	6.91%	35.91%	1.838296285
2	{Canterville Ghost} => {Strawberry Blonde}	6.91%	35.38%	1.838296285
3	{Doc Savage: The Man of Bronze} => {Green Slime}	8.28%	38.98%	1.791373861
4	{Green Slime} => {Doc Savage: The Man of Bronze}	8.28%	38.06%	1.791373861
5	{Green Slime} => {She}	8.22%	37.80%	1.769506084
6	{She} => {Green Slime}	8.22%	38.50%	1.769506084
7	{Jack the Ripper (1988)} => {She}	5.94%	35.14%	1.644963145
8	{She} => {Jack the Ripper (1988)}	5.94%	27.81%	1.644963145
9	{Pretty Maids All In A Row} => {Dark of the Sun}	7.37%	34.22%	1.580866863
10	{Dark of the Sun} => {Pretty Maids All In A Row}	7.37%	34.04%	1.580866863
11	{Doc Savage: The Man of Bronze} => {She}	6.97%	32.80%	1.535434995
12	{She} => {Doc Savage: The Man of Bronze}	6.97%	32.62%	1.535434995
13	{Pretty Maids All In A Row} => {Green Slime}	6.85%	31.83%	1.462854278
14	{Green Slime} => {Pretty Maids All In A Row}	6.85%	31.50%	1.462854278
15	{Pretty Maids All In A Row} => {She}	6.62%	30.77%	1.440559441

Sample Interpretation for Rule 1: Those customers buying Strawberry Blonde are usually more prone to also buy Canterville Ghost.

Association Rule Mining-Multiple Cardinalities

S No.	Rules	Support	Confidence	Lift
1	{Green Slime,Jack the Ripper (1988)} => {She}	3.14%	75.34%	3.52739726
2	{Canterville Ghost,Dark of the Sun} => {Strawberry Blonde}	2.57%	66.18%	3.4384273
3	{Jack the Ripper (1988),Strawberry Blonde} => {Canterville Ghost}	2.51%	65.67%	3.362311251
4	{She,Strawberry Blonde} => {Canterville Ghost}	2.51%	63.77%	3.264852954
5	{Canterville Ghost,Pretty Maids All In A Row} => {Strawberry Blonde}	2.57%	62.50%	3.247403561
6	{Dark of the Sun,Doc Savage: The Man of Bronze,She} => {Green Slime}	2.11%	69.81%	3.208389046
7	{Dark of the Sun,Doc Savage: The Man of Bronze,Green Slime} => {She}	2.11%	68.52%	3.207912458
8	{Doc Savage: The Man of Bronze,Pretty Maids All In A Row,She} => {Green Slime}	2.06%	69.23%	3.181708056
9	{Dark of the Sun,Strawberry Blonde} => {Canterville Ghost}	2.57%	60.81%	3.11344239
10	{Pretty Maids All In A Row,Strawberry Blonde} => {Canterville Ghost}	2.57%	60.00%	3.071929825
11	{Doc Savage: The Man of Bronze,Pretty Maids All In A Row} => {Green Slime}	3.26%	66.28%	3.046053836
12	{Doc Savage: The Man of Bronze,Jack the Ripper (1988)} => {She}	2.23%	65.00%	3.043181818
13	{Canterville Ghost,Jack the Ripper (1988)} => {Strawberry Blonde}	2.51%	57.89%	3.008121193
14	{Doc Savage: The Man of Bronze,Green Slime,Pretty Maids All In A Row} => {She}	2.06%	63.16%	2.956937799
15	{Doc Savage: The Man of Bronze,Stranger on the Third Floor} => {Green Slime}	2.57%	64.29%	2.954443195

Sample Interpretation for Rule 1: Those customers who buy ‘Green Slime’ and ‘Jack the Ripper’ are generally more prone to buy ‘She’ also.