**Netflix Movies And Shows Clustering**

**Anannya Sagar Das**

**Bhavesh Vitthal Rikame**

**Itisha Jain**

**Kunal Sanjay Borse**

**Data science trainees,**

**AlmaBetter,**

**Bangalore**

**Abstract**

The project discusses clustering which falls under unsupervised machine learning. The clustering is done on a dataset which contains information about content on Netflix which is one of the most leading streaming platforms of our age.

This dataset consists of tv shows and movies available on Netflix as of 2019. The dataset is collected from Flexible which is a third-party Netflix search engine.

In 2018, they released an interesting report which shows that the number of TV shows on Netflix has nearly tripled since 2010. The streaming service’s number of movies has decreased by more than 2,000 titles since 2010, while its number of TV shows has nearly tripled. The aim of the project is to cluster the data into optimal clusters. In order to

To achieve it we performed Natural Language Processing, dimensionality reduction technique-

Principal Component Analysis (PCA). Further we used methods like the Elbow Method, Silhouette

Method and Dendrogram Visualization and found that the optimal number of clusters is six. Eventually, we went ahead to perform clustering using algorithms like K-Modes on categorical variables and K-means on text based columns. Additionally, we performed Agglomerative Hierarchical Clustering and we got the best cluster arrangements. Next, we labeled the clusters and also tried visualizing them using Bokeh, etc. Finally, we developed a reccomendation system using Cosine Similarity and tested it on a random movie title to get recommendations based on that particular title.

**Problem Statement**

The data we have is related to movies and tv shows present in Netflix. We need to use clustering to group the data into an optimal number of clusters based on various text-based features and categorical variables provided in the dataset.

**Introduction :-**

This dataset consists of tv shows and movies available on Netflix as of 2019. The dataset is collected from Flexible which is a third-party Netflix search engine. In 2018, they released an interesting report which shows that the number of TV shows on Netflix has nearly tripled since 2010. The streaming service’s number of movies has decreased by more than 2,000 titles since 2010, while its number of TV shows has nearly tripled. It will be interesting to explore what all other insights can be obtained from the same dataset.

## **Data Description**

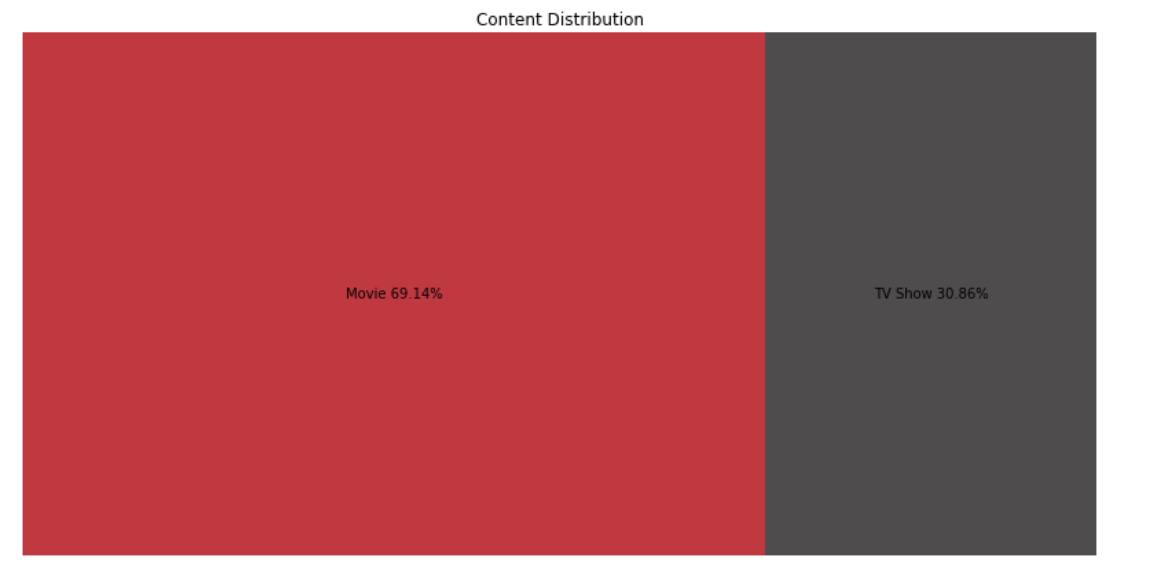
The list of features available to us are given below:-

## **Input variables:**

* show\_id : Unique ID for every Movie / Tv Show
* type : Identifier - A Movie or TV Show
* title : Title of the Movie / Tv Show
* director : Director of the Movie
* cast : Actors involved in the movie / show
* country : Country where the movie / show was produced
* date\_added : Date it was added on Netflix
* release\_year : Actual Release Year of the movie / show
* rating : TV Rating of the movie / show
* duration : Total Duration - in minutes or number of seasons
* listed\_in : Genre
* description: The Summary description

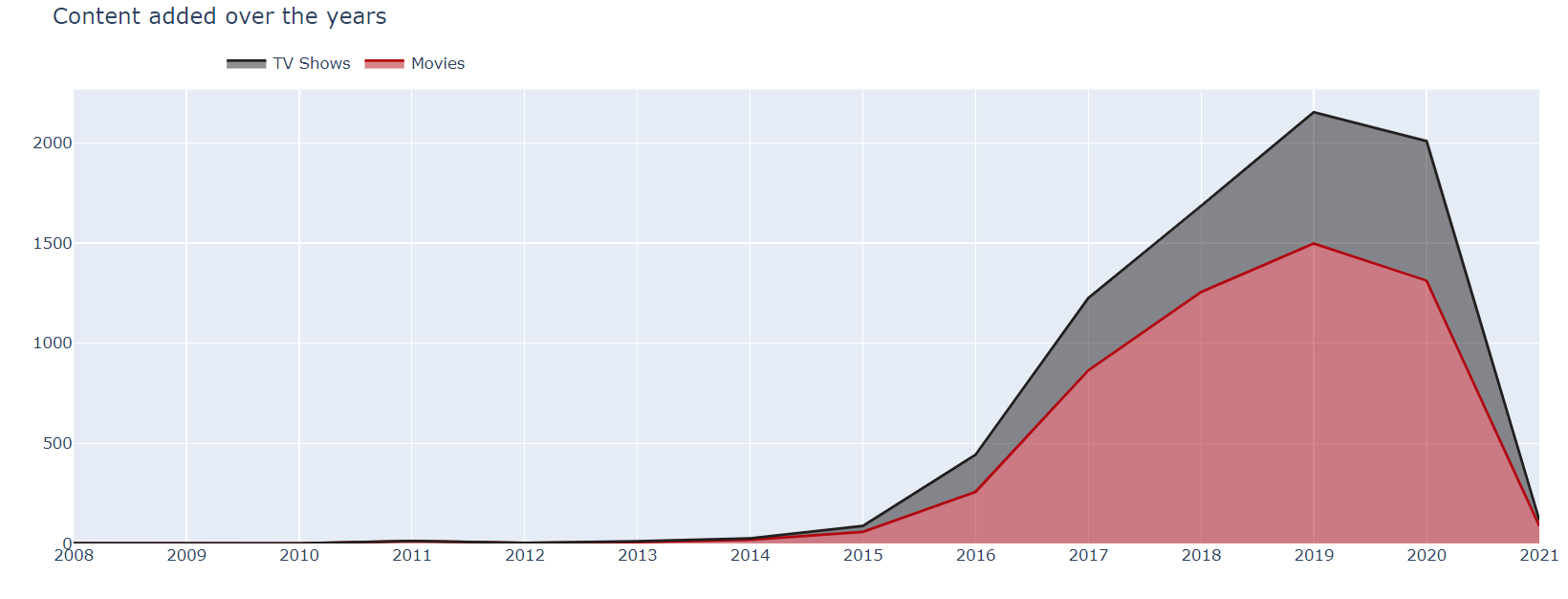
**1.Exploratory Data Analysis**

Content Distribution :-



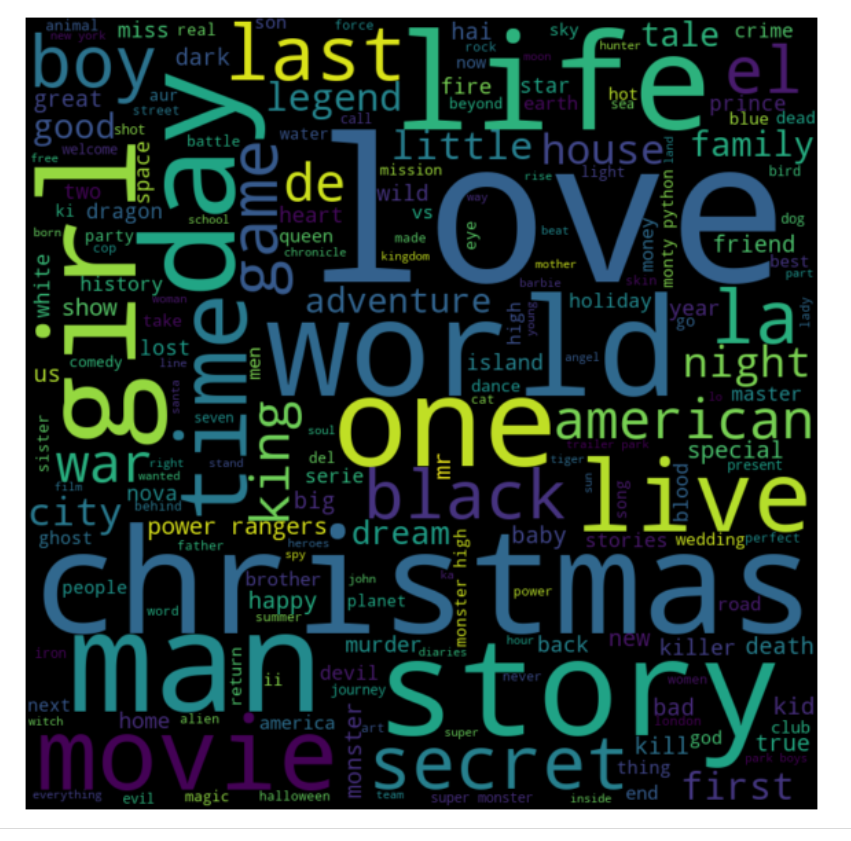
As we can see the whole content distribution is divided into the two parts first one is movies which is 69.14 % of total content and second is TV shows which is 30.86 of the total content

Content Added over the years :-



Nowadays the TV shows have gained a lot more popularity than the movies. By analyzing the above plot we get to know that in recent years netflix is focusing on movies more than TV shows (movies is increased by 80% and TV shows is increased by 73% compare to 2016 data)

Word Cloud:-



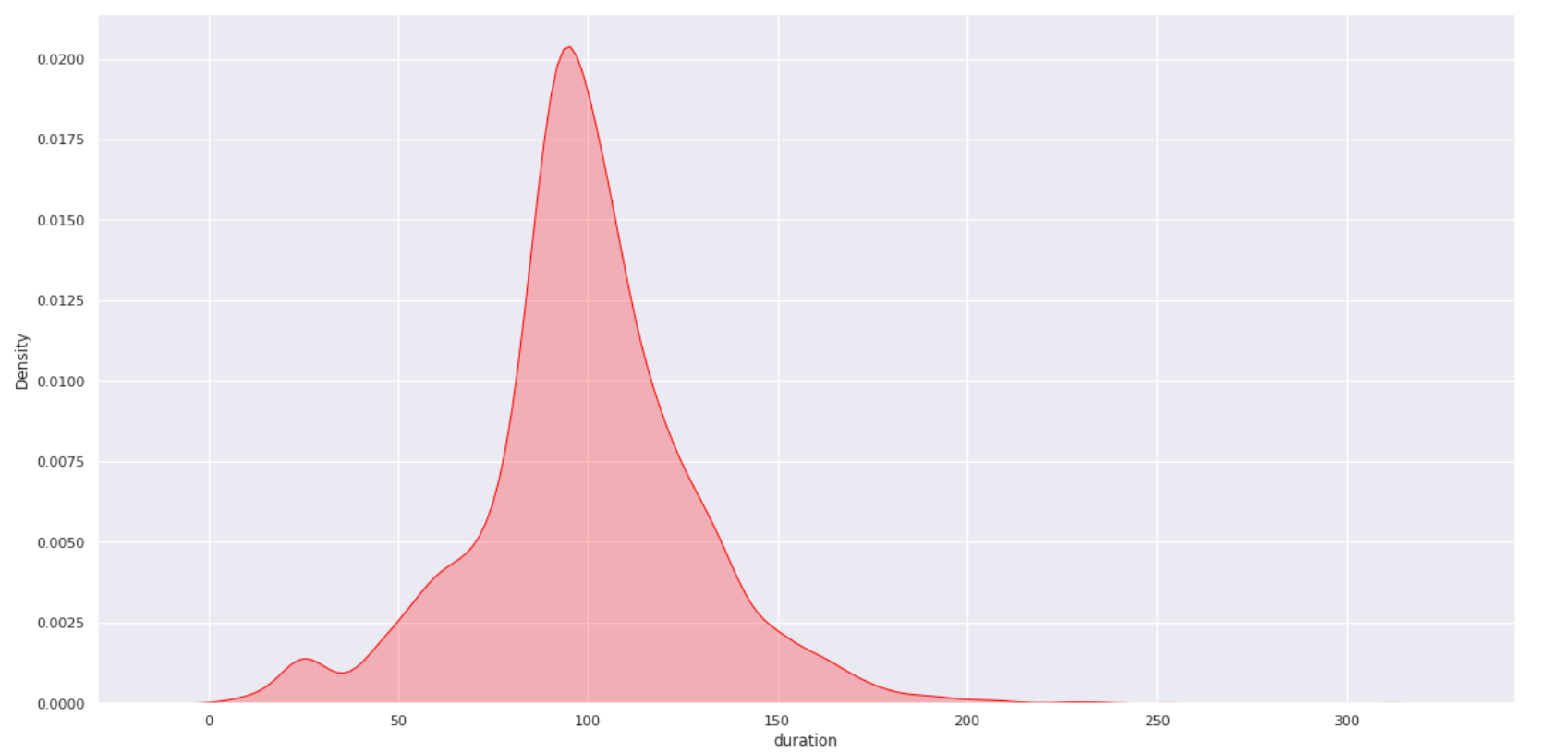
Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud.

Release Over The Years:-



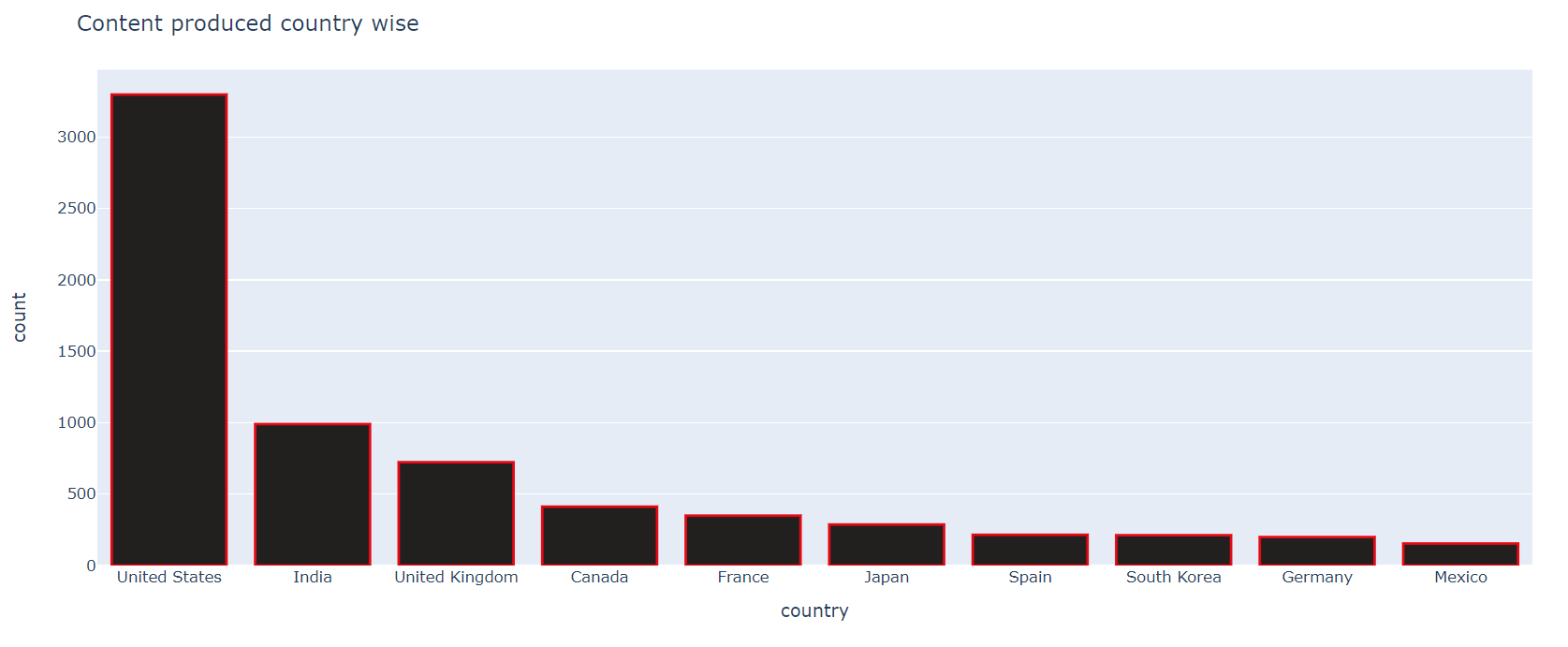
As per the data, Netflix is now a very popular OTT platform. The graph shows that from 2016 to 2020 most of the content is added on the netflix. Most of the content present was released in 2017 and 2018.

**Duration of Movies:-**



As per the given graph,we can see that Most of the movies are 70 to 120 minutes long.

**Content Country wise:-**



As we can see on the graph, most of the content is from the United States. India is the second largest content producer on Netflix followed by the United kingdom and Canada. Also Koria has some content on the netflix such as K-Drama,K-Pop Etc.

**NLP**

**What is natural language processing?**

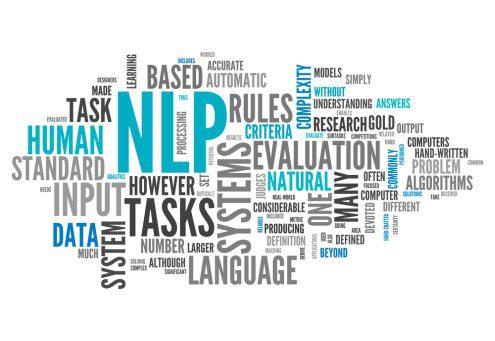
## 

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of [artificial intelligence or AI](https://www.ibm.com/cloud/learn/what-is-artificial-intelligence)—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

## Natural language processing (NLP) is a discipline which is interested in how human languages, and, to some extent, the humans who speak them, interact with technology. NLP is an interdisciplinary topic which has historically been the equal domain of artificial intelligence researchers and linguistics alike; perhaps obviously, those approaching the discipline from the linguistics side must get up to speed on technology, while those entering the discipline from the technology realm need to learn the linguistic concepts.



## **Key Terminologies Used In NLP World:**

## :So here they are, 18 select natural language processing terms, concisely defined, with links to further reading where appropriate.

**1.** [**Natural Language Processing (NLP)**](https://www.kdnuggets.com/tag/natural-language-processing)

Natural language processing (NLP) concerns itself with the interaction between natural human languages and computing devices. NLP is a major aspect of computational linguistics, and also falls within the realms of computer science and artificial intelligence.

**2. Tokenization**

Tokenization is, generally, an early step in the NLP process, a step which splits longer strings of text into smaller pieces, or **tokens**. Larger chunks of text can be tokenized into sentences, sentences can be tokenized into words, etc. Further processing is generally performed after a piece of text has been appropriately tokenized.

**3. Normalization**

Before further processing, text needs to be normalized. Normalization generally refers to a series of related tasks meant to put all text on a level playing field: converting all text to the same case (upper or lower), removing punctuation, expanding contractions, converting numbers to their word equivalents, and so on. Normalization puts all words on equal footing, and allows processing to proceed uniformly.

**4. Stemming**

Stemming is the process of eliminating affixes (suffixes, prefixes, infixes, circumfixes) from a word in order to obtain a word stem.

running → run

**5. Lemmatization**

Lemmatization is related to stemming, differing in that lemmatization is able to capture canonical forms based on a word's [lemma](https://en.wikipedia.org/wiki/Lemma_(morphology)).

For example, stemming the word "better" would fail to return its citation form (another word for lemma); however, lemmatization would result in the following:

better → good

It should be easy to see why the implementation of a stemmer would be the less difficult feat of the two.**6. Corpus**

In linguistics and NLP, corpus (literally Latin for body) refers to a collection of texts. Such collections may be formed of a single language of texts, or can span multiple languages -- there are numerous reasons for which multilingual corpora (the plural of corpus) may be useful. Corpora may also consist of themed texts (historical, Biblical, etc.). Corpora are generally solely used for statistical linguistic analysis and hypothesis testing.

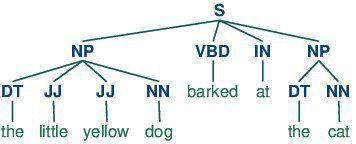
7**. Stop Words**

Stop words are those words which are filtered out before further processing of text, since these words contribute little to overall meaning, given that they are generally the most common words in a language. For instance, "the," "and," and "a," while all required words in a particular passage, don't generally contribute greatly to one's understanding of content. As a simple example, the following [pangram](https://en.wikipedia.org/wiki/Pangram) is just as legible if the stop words are removed:

~~The~~ quick brown fox jumps over ~~the~~ lazy dog.

**8. Parts-of-speech (POS) Tagging**

POS tagging consists of assigning a category tag to the tokenized parts of a sentence. The most popular POS tagging would be identifying words as nouns, verbs, adjectives, etc.



**9. Statistical Language Modeling**

Statistical Language Modeling is the process of building a statistical language model which is meant to provide an estimate of a natural language. For a sequence of input words, the model would assign a probability to the entire sequence, which contributes to the estimated likelihood of various possible sequences. This can be especially useful for NLP applications which generate text.

**10. Bag of Words**

Bag of words is a particular representation model used to simplify the contents of a selection of text. The bag of words model omits grammar and word order, but is interested in the number of occurrences of words within the text. The ultimate representation of the text selection is that of a bag of words (bag referring to the set theory concept of [multisets](https://en.wikipedia.org/wiki/Multiset), which differ from simple sets).

Actual storage mechanisms for the bag of words representation can vary, but the following is a simple example using a dictionary for intuitiveness. Sample text:

"Well, well, well," said John.

"There, there," said James. "There, there."

The resulting bag of words representation as a dictionary:

{

'well': 3,

'said': 2,

'john': 1,

'there': 4,

'james': 1

}

**11. n-grams**

n-grams is another representation model for simplifying text selection contents. As opposed to the orderless representation of bags of words, n-grams modeling is interested in preserving contiguous sequences of N items from the text selection.

An example of the trigram (3-gram) model of the second sentence of the above example ("There, there," said James. "There, there.") appears as a list representation below:

[

"there there said",

"there said james",

"said james there",

"james there there",

]

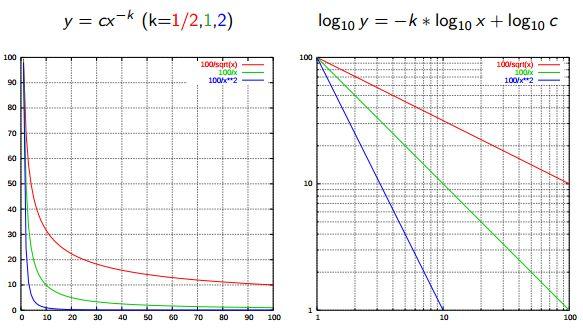
**12. Regular Expressions**

Regular expressions, often abbreviated regex or regexp, are a tried and true method of concisely describing patterns of text. A regular expression is represented as a special text string itself, and is meant for developing search patterns on selections of text. Regular expressions can be thought of as an expanded set of rules beyond the wildcard characters of ? and \*. Though often cited as frustrating to learn, regular expressions are incredibly powerful text searching tools.

**13. Zipf's Law**

Zipf's Law is used to describe the relationship between word frequencies in document collections. If a document collection's words are ordered by frequency, and y is used to describe the number of times that the xth word appears, Zipf's observation is concisely captured as y = cx-1/2 (item frequency is inversely proportional to item rank). More generally, [Wikipedia says](https://en.wikipedia.org/wiki/Zipf's_law):

Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.



**14. Similarity Measures**

There are numerous similarity measures which can be applied to NLP. What are we measuring the similarity of? Generally, strings.

* **Levenshtein** - the number of characters that must be deleted, inserted, or substituted in order to make a pair of strings equal
* **Jaccard** - the measure of overlap between 2 sets; in the case of NLP, generally, documents are sets of words
* **Smith Waterman** - similar to Levenshtein, but with costs assigned to substitution, insertion, and deletion

**16.** [**Semantic Analysis**](https://www.kdnuggets.com/tag/semantic-analysis)

Also known as meaning generation, semantic analysis is interested in determining the meaning of text selections (either character or word sequences). After an input selection of text is read and parsed (analyzed syntactically), the text selection can then be interpreted for meaning. Simply put, syntactic analysis is concerned with what words a text selection was made up of, while semantic analysis wants to know what the collection of words actually means. The topic of semantic analysis is both broad and deep, with a wide variety of tools and techniques at the researcher's disposal.

**17.** [**Sentiment Analysis**](https://www.kdnuggets.com/tag/sentiment-analysis)

Sentiment analysis is the process of evaluating and determining the sentiment captured in a selection of text, with sentiment defined as feeling or emotion. This sentiment can be simply positive (happy), negative (sad or angry), or neutral, or can be some more precise measurement along a scale, with neutral in the middle, and positive and negative increasing in either direction.



**18.** [**Information Retrieval**](https://www.kdnuggets.com/tag/information-retrieval)

Information retrieval is the process of accessing and retrieving the most appropriate information from text based on a particular query, using context-based indexing or metadata. One of the most famous examples of information retrieval would be Google Search.

## **NLP tasks**

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure—these just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful.

Several NLP tasks break down human text and voice data in ways that help the computer make sense of what it's ingesting. Some of these tasks include the following:

* Speech recognition, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.
* Part of speech tagging, also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies ‘make’ as a verb in ‘I can make a paper plane,’ and as a noun in ‘What make of car do you own?’
* Word sense disambiguation is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that determines the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).
* Named entity recognition, or NEM, identifies words or phrases as useful entities. NEM identifies ‘Kentucky’ as a location or ‘Fred’ as a man's name.
* Coreference resolution is the task of identifying if and when two words refer to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., ‘she’ = ‘Mary’), but it can also involve identifying a metaphor or an idiom in the text (e.g., an instance in which 'bear' isn't an animal but a large hairy person).
* Sentiment analysis attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.
* Natural language generation is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

# **Unsupervised-Text-Clustering using Natural Language Processing(NLP)**

What is Supervised Learning and Unsupervised Learning?

The type of Machine Learning models that learn by labeled examples are called ‘Supervised Learning’.

· Classification (Target values are discrete classes)

· Regression (Target values are continuous values)

To find structure in unlabelled data is called ‘Unsupervised Learning’.

· Find groups of similar instances in the data (Clustering)

· Finding unusual patterns (Outlier detection)

So, what is Clustering exactly?

Grouping of similar data together is called Clustering. And this is obtained by calculating the distance between the points.

There are two types of clustering that are predominantly used.

· K-means Clustering

· Hierarchical Clustering

Here, we will look at K-means Clustering.

What is K-means Clustering?

Grouping similar data points together and discovering underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset. The K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster. The ‘means’ in the K-means refers to averaging of the data; that is, finding the centroid.

This project is based on Natural Language processing(NLP)

Now, let us quickly run through the steps of working with the text data.

Step 1: Import the data. (Incident data for server related issues)

Here, when we look at the data , the data consists of comments given by people for a particular incident tickets.we can see that each line has different text and the thing that we are concerned about is the actual problem description on the server.

Here we would need the help of regular expression to get the exact content that we need.

Step 2 : Regular Expression to extract the content from the raw dataset.Step 3: Text Preprocessing — Is nothing but cleaning the data.

1. Removing punctuation.
2. Remove Stopwords
3. Remove Additional Spaces and Digits.
4. And Lemmatize the Text.
5. Returns cleaned list.

Step 4: Vectorization is nothing but creating a vector of words called vocabulary.

TFIDF Vectorizer is used to create a vocabulary. TFIDF is a product of how frequent a word is in a document multiplied by how unique a word is w.r.t the entire corpus. ngram\_range parameter : which will help to create one , two or more word vocabulary depending on the requirement.

Step 5: K Means ClusteringStep 6: To Find the Optimal Number of Clusters.

So, how do we decide on the total number of optimal clusters?

A good cluster is one with the distance between the points within the cluster must be less and the distance between the two centroids of two different clusters must be more.

And we evaluate for the optimal number of clusters using two predominant methods

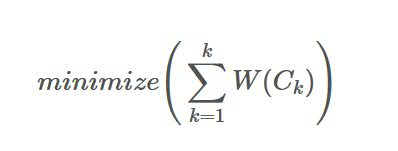
·· Elbow method

· Average Silhouette method

· Gap Statistic

Here, we will look at the Elbow Method.

The “elbow” method to help data scientists select the optimal number of clusters by fitting the model with a range of values for K. If the line chart resembles an arm, then the “elbow” (the point of inflection on the curve) is a good indication that the underlying model fits best at that point. The total WSS(within-cluster sum of square) measures the compactness of the clustering and we want it to be as small as possible. The Elbow method looks at the total WSS as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn’t improve the total WSS.



where Ck is the kth cluster and W(Ck) is the within-cluster variation. The total within-cluster sum of squares (wss) measures the compactness of the clustering and we want it to be as small as possible.

How it calculates :

* For each k, calculate the total within-cluster sum of squares (wss).
* Plot the curve of wss according to the number of clusters k.
* The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

**Clustering Algorithms:**

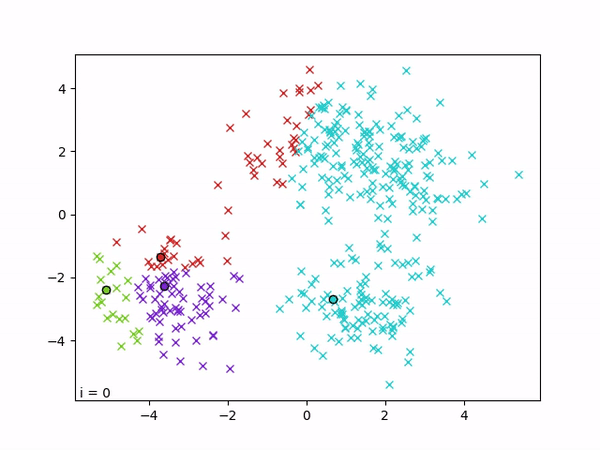
**1.K-means clustering**

K-Means is probably the most well-known clustering algorithm. K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.

The first to propose the discrete k-means algorithm for clustering data in the sense of minimization was Forgy.

How the K-means algorithm works :

The objective of K-means is simple: group similar data points together and discover underlying patterns.



1. Determine the value “K”, the value “K” represents the number of clusters.

A cluster refers to a collection of data points aggregated together because of certain similarities.

2. Randomly select 3 distinct centroids (new data points as cluster initialization)

3. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.

In other words, the K-means algorithm identifies the k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

*The ‘means’ in the K-means refers to averaging of the data; that is, finding the centroid.*

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

* The centroids have stabilized — there is no change in their values because the clustering has been successful.
* The defined number of iterations has been achieved.

**2. K-modes algorithm**

KModes clustering is one of the unsupervised Machine Learning algorithms that is used to **cluster categorical variables.**

You might be wondering, why KModes when we already have KMeans.

KMeans uses mathematical measures (distance) to cluster continuous data. The lesser the distance, the more similar our data points are. Centroids are updated by Means.

But for categorical data points, we cannot calculate the distance. So we go for the KModes algorithm. It uses the dissimilarities between the data points. The lesser the dissimilarities the more similar our data points are. **It uses Modes instead of means.**

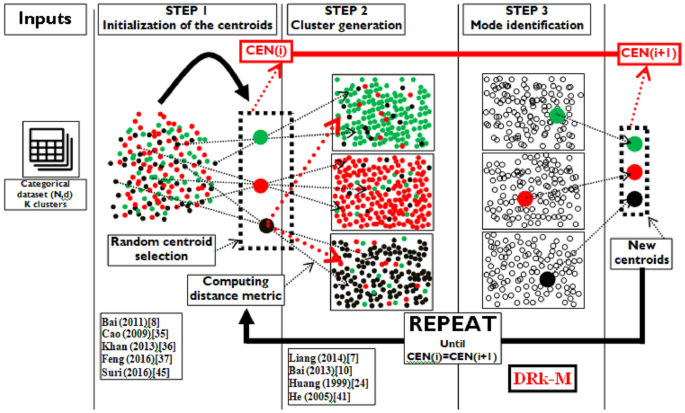
### 

### 

### 

### **How does the KModes algorithm work?**

Like K-means clustering, we need to specify the K at the beginning.

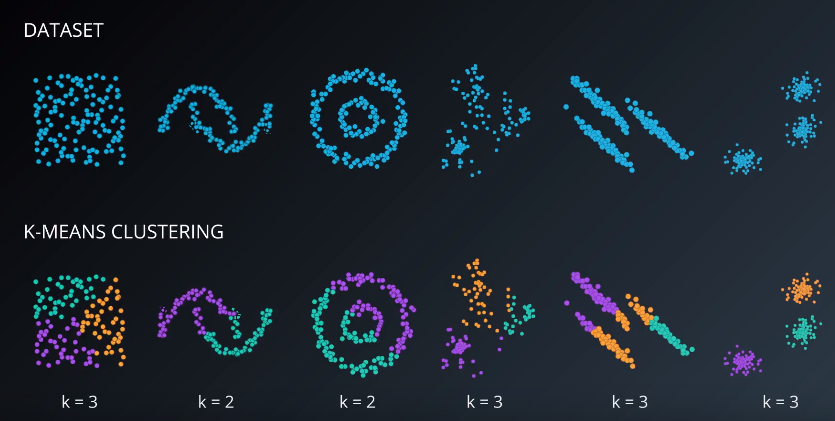


* Pick K observations at random and use them as clusters or defining points of a cluster also known as leaders.
* Calculate the dissimilarities and assign each observation to its closest cluster.
* Define new modes for the clusters.
* Repeat the first and third step until there is no re-assignment required.

**K-Means Limitations**

Although K-Means is a great clustering algorithm, it is most useful when we know beforehand the exact number of clusters and when we are dealing with spherical-shaped distributions.

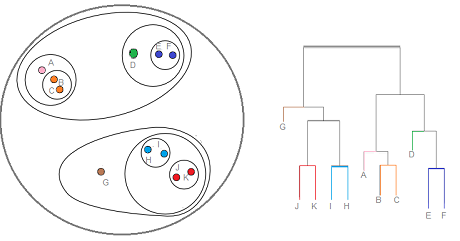
The following picture shows what we would obtain if we use K-means clustering in each dataset even if we knew the exact number of clusters beforehand:



It is quite common to take the K-Means algorithm as a benchmark to evaluate the performance of other clustering methods.

# **3. Hierarchical Clustering**

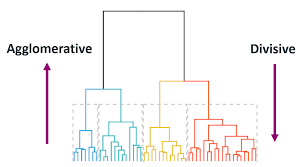
Hierarchical clustering is an alternative to prototype-based clustering algorithms. The main advantage of Hierarchical clustering is that we do not need to specify the number of clusters, it will find it by itself. In addition, it enables the plotting of dendrograms. Dendrograms are visualizations of binary hierarchical clustering.



Observations that fuse at the bottom are similar while those that are at the top are quite different. With dendrograms, conclusions are made based on the location of the vertical axis rather than on the horizontal one.

**Kinds of Hierarchical Clustering**

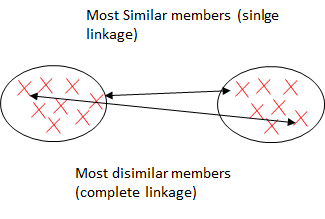
There are two approaches to this type of clustering: Agglomerative and divisive.



* Divisive: this method starts by enlobbing all data points in one single cluster. Then, it will split the cluster iteratively into smaller ones until each one of them contains only one sample.
* Agglomerative: this method starts with each sample being a different cluster and then merging them by the ones that are closer from each other until there is only one cluster.

**Single Linkage & Complete Linkage**

These are the most common algorithms used for agglomerative hierarchical clustering.



**Single Linkage**

As being an agglomerative algorithm, single linkage starts by assuming that each sample point is a cluster. Then, it computes the distances between the most similar members for each pair of clusters and merges the two clusters for which the distance between the most similar members is the smallest.

**Complete Linkage**

Although being similar to its brother (single linkage) its philosophy is exactly the opposite, it compares the most dissimilar data points of a pair of clusters to perform the merge.

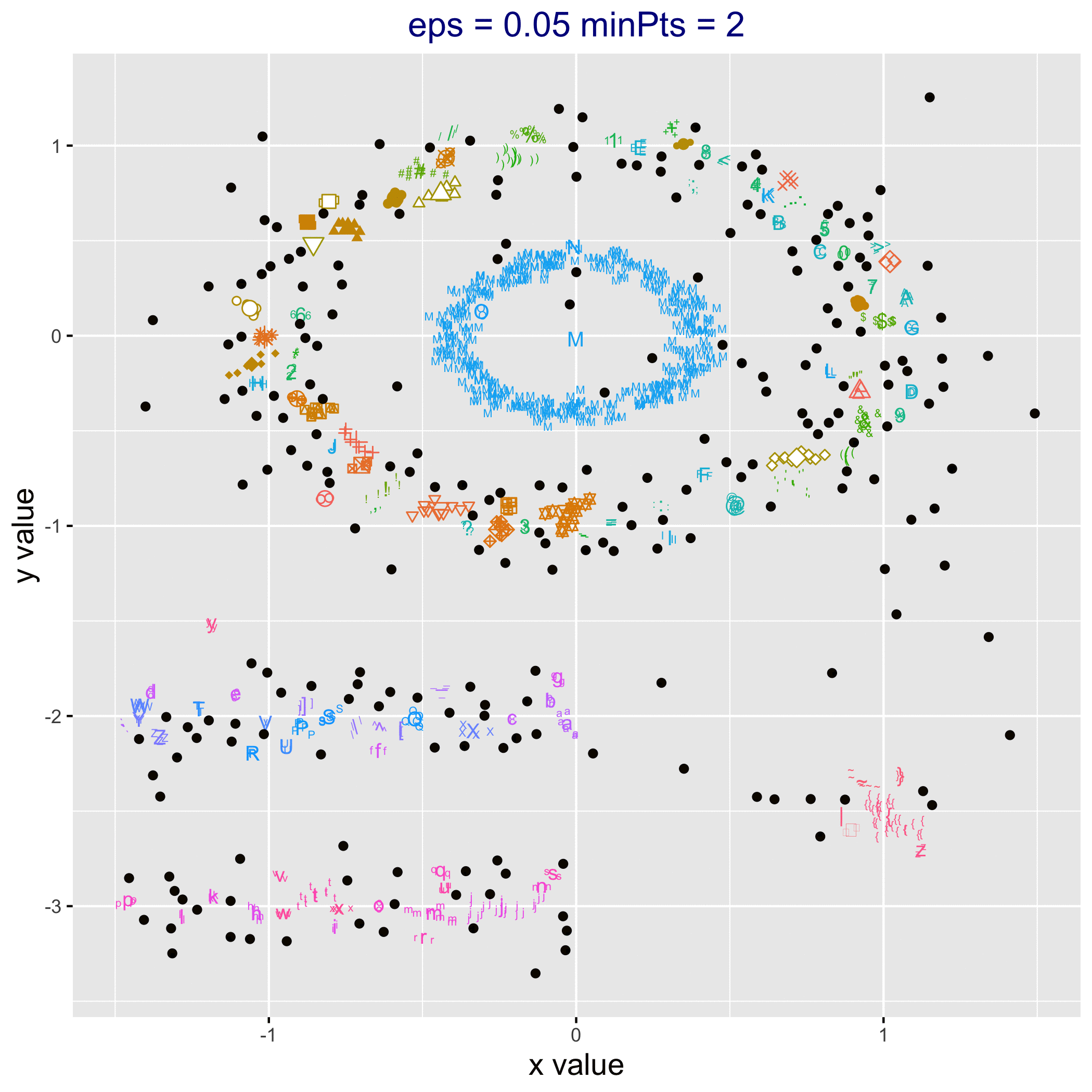
**Advantages of Hierarchical Clustering**

* The resulting hierarchical representations can be very informative.
* Dendrograms provide an interesting and informative way of visualization.
* They are especially powerful when the dataset contains real hierarchical relationships.

**Disadvantages of Hierarchical Clustering**

* They are very sensitive to outliers and, in their presence, the model performance decreases significantly.
* They are very expensive, computationally speaking.

**Density-Based Spatial Clustering of Applications with Noise (DBSCAN)**

****

Density-Based Spatial Clustering of Applications with Noise, or DBSCAN, is another clustering algorithm specially useful to correctly identify noise in data.

It is based on a number of points with a specified radius ε and there is a special label assigned to each datapoint. The process of assigning this label is the following:

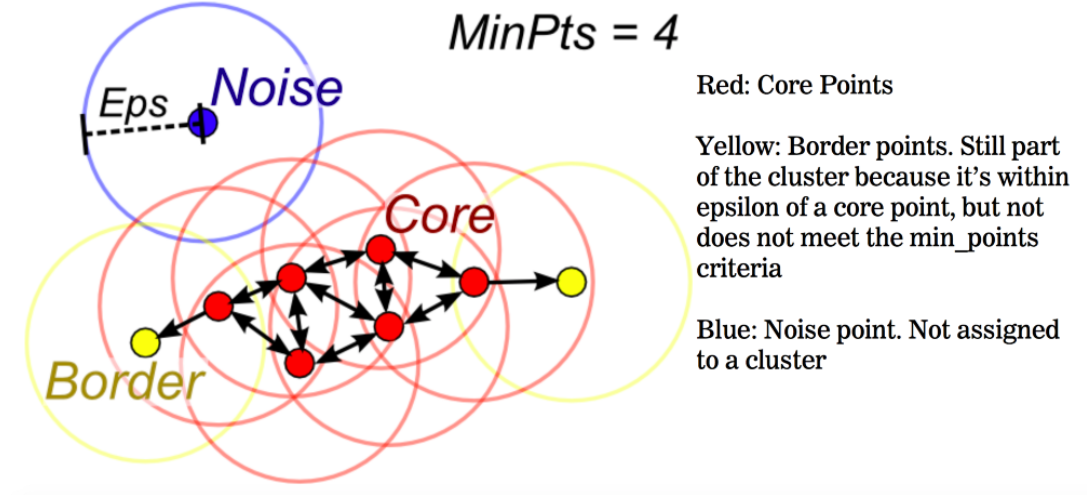
* It has a specified number (MinPts) of neighbor points. A core point will be assigned if there is this MinPts number of points that fall in the ε radius.
* A border point will fall in the ε radius of a core point, but will have fewer neighbors than the MinPts number.
* Every other point will be noise points.

**How does the DBSCAN Algorithm work?**

The algorithm follows the logic:

1. Identify a core point and make a group for each one, or for each connected group of core points (if they satisfy the criteria to be core point).
2. Identify and assign border points to their respective core points.

The following figure summarizes this process very well.



**Principal Component Analysis**

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

STEP 1: STANDARDIZATION

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

STEP 2: COVARIANCE MATRIX COMPUTATION

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

STEP3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data. Before getting to the explanation of these concepts, let’s first understand what do we mean by principal components.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

How PCA Constructs the Principal Components

As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set. For example, let’s assume that the scatter plot of our data set is as shown below, can we guess the first principal component ? Yes, it’s approximately the line that matches the purple marks because it goes through the origin and it’s the line in which the projection of the points (red dots) is the most spread out. Or mathematically speaking, it’s the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).

Principal Component Analysis second principal

The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

This continues until a total of p principal components have been calculated, equal to the original number of variables.

Now that we understand what we mean by principal components, let’s go back to eigenvectors and eigenvalues. What you first need to know about them is that they always come in pairs, so that every eigenvector has an eigenvalue. And their number is equal to the number of dimensions of the data. For example, for a 3-dimensional data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

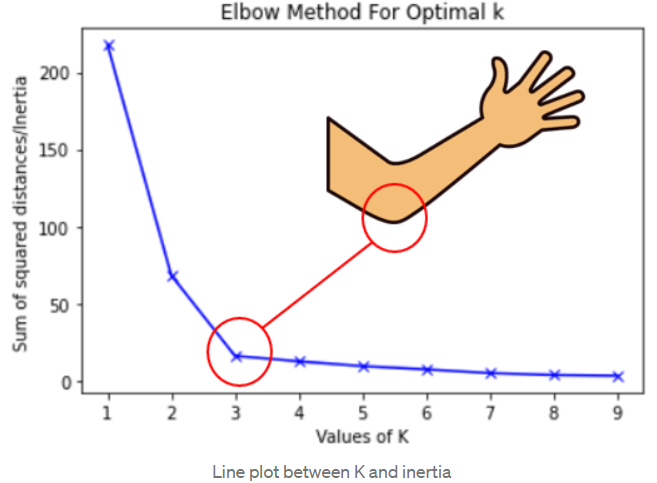
Without further ado, it is eigenvectors and eigenvalues who are behind all the magic explained above, because the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance(most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.

By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

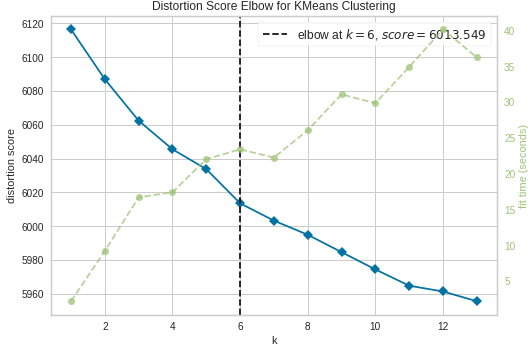
**Model Evaluation**

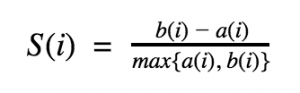
* **Elbow method**

In the Elbow method, we actually vary the number of clusters ( K ) from 1 – 10. For each value of K, we calculate WCSS ( Within-Cluster Sum of Square ). WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1.

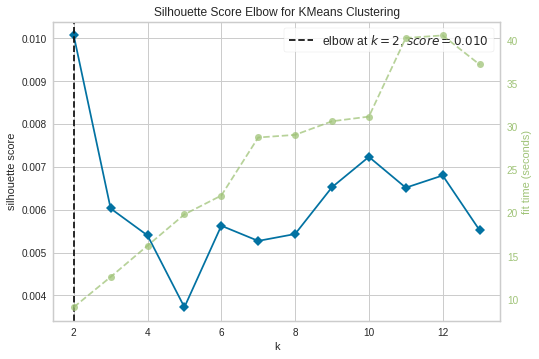


When we analyze the graph we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to this point is the optimal K value or an optimal number of clusters.

 **Silhouette Coefficient**

The Silhouette Coefficient for a point *i* is defined as follows:where *b(i)* is the smallest average distance of point *i* to all points in any other cluster and *a(i)* is the average distance of *i* from all other points in its cluster. For example, if we have only 3 clusters A,B and C and i belongs to cluster C, then *b(i)* is calculated by measuring the average distance of *i* from every point in cluster A, the average distance of i from every point in cluster B and taking the smallest resulting value. The Silhouette Coefficient for the dataset is the average of the Silhouette Coefficient of individual points. The Silhouette Coefficient tells us if individual points are correctly assigned to their clusters. We can use the following thumb rules while using Silhouette Coefficient:

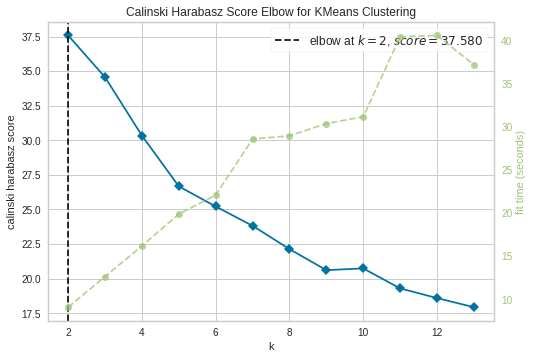
1. *S(i)* close to 0 means that the point is between two clusters
2. If it is closer to -1, then we would be better off assigning it to the other clusters
3. If *S(i)* is close to 1, then the point belongs to the ‘correct’ cluster



* **Calinski-Harabasz Index**

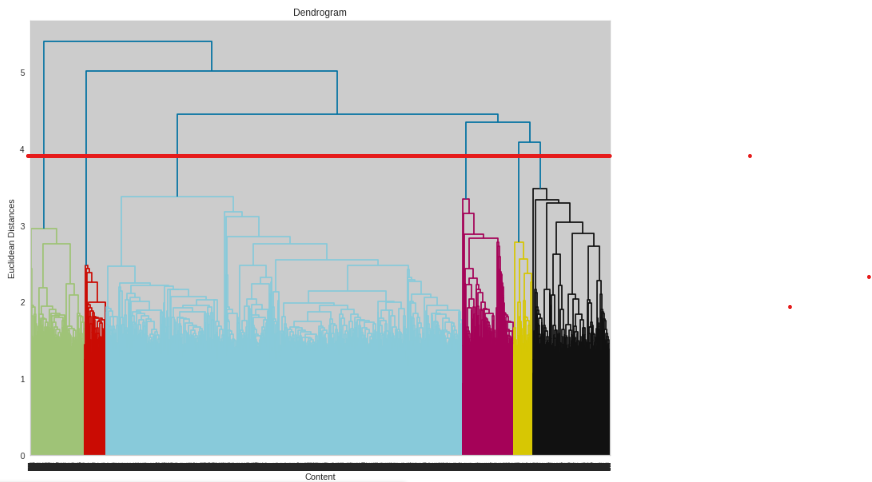
The Calinski-Harabasz Index is based on the idea that clusters that are (1) themselves very compact and (2) well-spaced from each other are good clusters. The index is calculated by dividing the variance of the sums of squares of the distances of individual objects to their cluster center by the sum of squares of the distance between the cluster centers. Higher the Calinski-Harabasz Index value, better the clustering model. The formula for Calinski-Harabasz Index is defined as:

where k is the number of clusters, n is the number of records in data, BCSM (between cluster scatter matrix) calculates separation between clusters and WCSM (within cluster scatter matrix) calculates compactness within clusters.



* **Dendrogram**

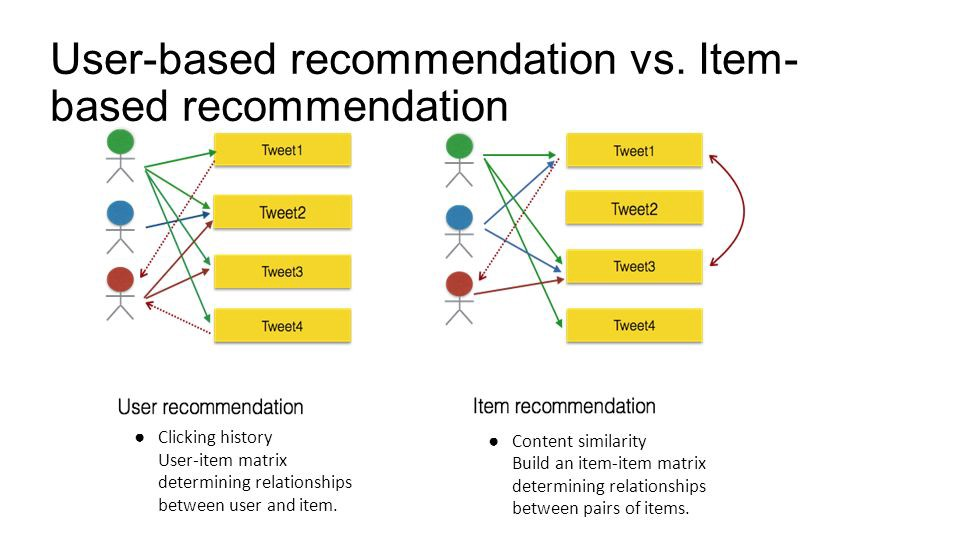
This technique is specific to the agglomerative hierarchical method of clustering. The agglomerative hierarchical method of clustering starts by considering each point as a separate cluster and starts joining points to clusters in a hierarchical fashion based on their distances.To get the optimal number of clusters for hierarchical clustering, we make use of a dendrogram which is a tree-like chart that shows the sequences of merges or splits of clusters.

If two clusters are merged, the dendrogram will join them in a graph and the height of the join will be the distance between those clusters. As a rule of thumb, look for the clusters with the longest ‘branches’, the shorter they are, the more similar they are to the following ‘twigs’ and ‘leaves’. The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold, which in our case is the red line cutting through 6 vertical lines.

**Recommendation Systems**

Recommendation engines are one of the most popular applications of machine learning techniques in the current internet age. These are extensively used in e-commerce websites for recommending similar products and on movie recommender sites. They are responsible for generating various custom tailored news suggestions for us. Which will drive more content engagement from users leading to better user experience and more revenue for the organization. Hence, they are of extreme importance in today’s industry.

Recommendation engines basically filter the data and recommend most relevant results to users. These results are recommended in such a manner that likelihood of interest results in maximum. Now, all the recommendation engines have user data and their history available with them for creating their filtering algorithms to work. Which eventually helps them generate very accurate recommendations for each unique user.



User-based filtering is based on history of users and similarity b/w them from their purchase histories for example. But, Item-based recommendations are based on content based similarity. Like, “how many times a few items are bought together”. Next time, the most frequent of these purchases will be recommended together.

**Content Based Filtering :-**

Content-based filtering. It relies on similarities between features of the items. It recommends items to a customer based on previously rated highest items by the same customer. List of features about these items needs to be generated.

* Each item will have an item profile
* A table structure will list these properties
* Comparing what and how many features match and collect scores
* Recommend highest scored item
* Code will be based on an algorithm, by given some item, the most similar item will be found
* Best scoring match will be provided to the user
* This method relies on item features only, and not the user preferences.

**Collaborative filtering.**

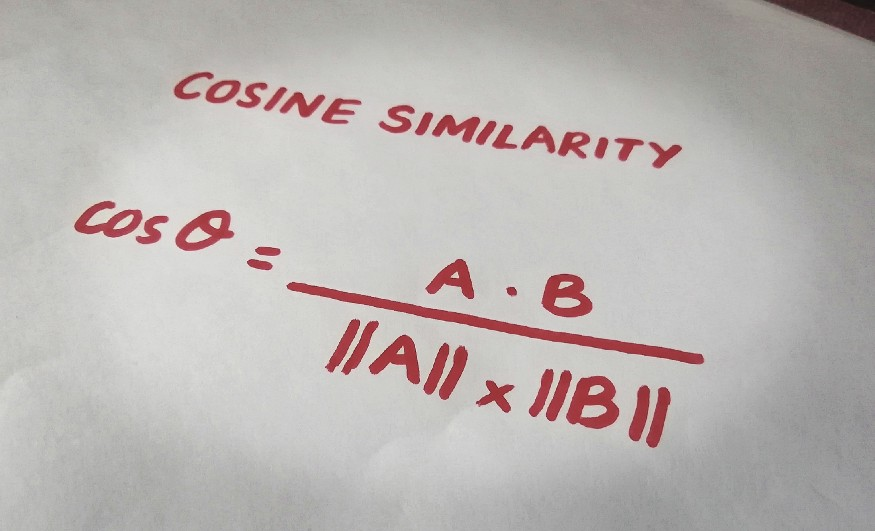
It relies on how other users responded to these same items. It doesn’t rely on features of the item, but the preferences from other users. Similar user survey needs to be done.

* Users will have a table with different rated items of what they choose or liked
* Based on the similarities, predictions can be made of what the user might like, based on what similar users did.
* The list will be filtered and matched to users who used the same items for comparison and recommendations
* Everything will be summed up and highest score will be recommended
* Code will be created based on an algorithm, by given a user x, recommend an item that x might like
* Item with the highest score will be recommended

Problem with this method is that you need to have data to make recommendations. More data you have, the better recommendations will be.

**Using Cosine Similarity :-**

Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–1. 0 means no similarity, whereas 1 means that both the items are 100% similar.



The python Cosine Similarity or cosine kernel, computes similarity as the normalized dot product of input samples X and Y. We will use the sklearn cosine\_similarity to find the cos θ for the two vectors in the count matrix.

def recommend(movie):

index = new[new['title'] == movie].index[0]

distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])

for i in distances[1:7]:

print(new.iloc[i[0]].title)

When we provided a movie name ‘Handsome Devil’ then in recommendation we get similar type of genre movie like Secret time, homecoming king, live in new york….etc in recommendation

**#recommendations for the given movie/tv show**

**recommend('Handsome Devil')**

**Output :-**

* **Bert Kreischer: Secret Time**
* **Hasan Minhaj: Homecoming King**
* **Aditi Mittal: Things They Wouldn't Let Me Say**
* **Dieter Nuhr: Nuhr in Berlin**
* **Wonho Chung: Live in New York**
* **Chingo Bling: They Can't Deport Us All**

**Conclusion**

* The Data set contains 7787 rows and 12 columns. There are missing values in columns director, cast, date\_added and release\_year. In order to not lose important information we have replaced the missing values with “ ” using the .fillna(“ ”) method.
* There are two types of content: TV shows and Movies (30.86% contains TV shows and 69.14% contains Movies) meaning there are more movies than TV shows on Netflix.
* By analyzing the date\_added and release year column with respect to content types it can be observed that over the years Netflix is focusing more on movies than TV shows. (movies is increased by 80% and TV shows is increased by 73% compare to 2016 data)
* The most number of the movies and TV shows release in 2017 and 2020 respectively and United States have the maximum content on Netflix
* International Movies make up the top most genre, and the most of the content is added during the months of October to January.
* The words that occurred most of the time in the ‘title’ column are LOVE, MAN, WORLD, CHRISTMAS.We can infer from the above words that there are more movies/tv shows of the romantic genre and also christmas movies/tv shows.
* The most number of content in the NETFLIX were directed by "Jan Suter", followed by "Raul Campos", "Marcus Roby
* India has the most number of actors whose names come under top 20 movies actors count. India produces the most movies or TV shows across the globe.
* The duration of the movies are about 70 to 120 mins. And most shows on Netflix are of 1 season.
* Looking at methods of finding optimal clusters like the Elbow Method, Dendrogram, Silhouette Method, we could see that the clusters of 6 are optimal.
* Taking the number of clusters as 6 we applied different clustering models for instance Kmeans, K-Modes.
* Further we went ahead to apply Hierarchical Agglomerative clustering on data and we got the best cluster arrangements.
* We also labeled the clusters using 6 as the optimal number of clusters. For the Kmode algorithm the 14 rating variables were clustered properly, each cluster comprising mostly one kind of rated content except cluster 1 and 6 comprising Movies and Tv Shows of various ratings. Under KMeans algorithm the clusters are:

Cluster 1 -Documentaries and Musical Documentaries,

Cluster 2- Dramas and International Movies,

Cluster 3- Children and Family Movies,

Cluster 4-Children and Family Movies,

Cluster 5-International Tv Shows and Tv Dramas,

Cluster 6-Stand Up Comedy and Talk Shows.