Classifying Text Using Machine Learning

Swetha Kolalapudi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Feature extraction using the bag of words model

Use K-Means clustering to identify a set of topics

Using the K-Nearest Neighbors model for classifying text into those topics

Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Identifying Themes

Article 1

Article 2

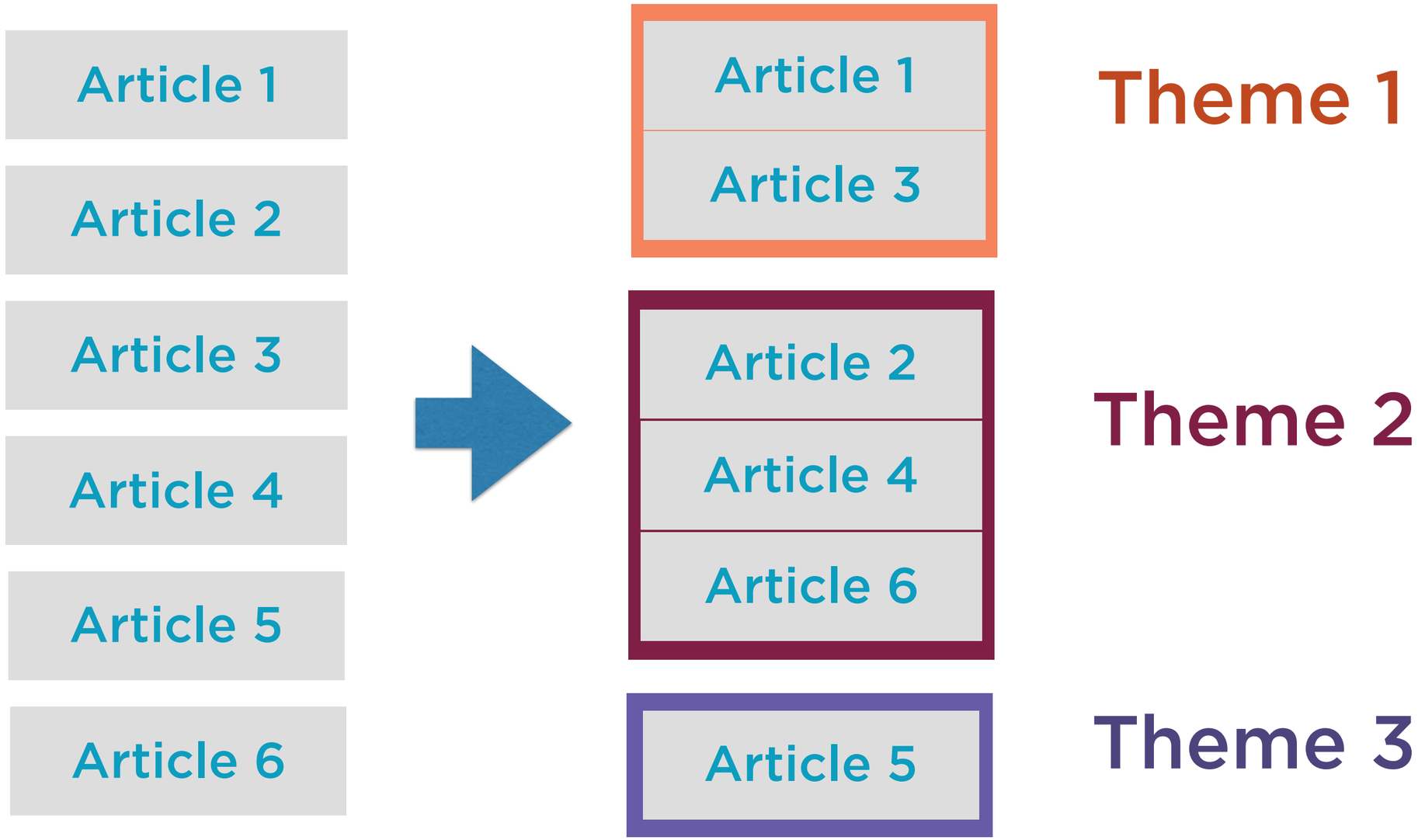
Article 3

Article 4

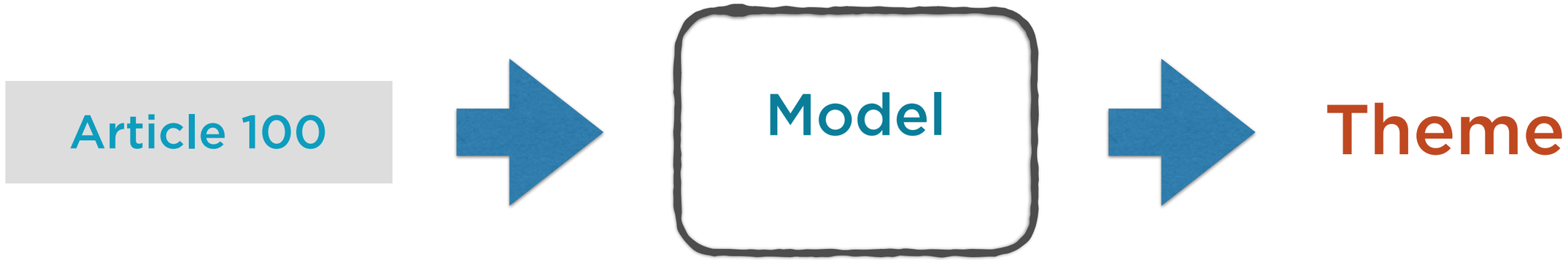
Article 5

Article 6

Identifying Themes



Assigning a Theme



Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Demo

Collect articles from a blog

Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Pick your Problem

We are given a large group of articles

Divide the articles into **groups** based on some **common attributes**

Clustering

Clustering

Group items together based on some measure of similarity

Clustering

The objective is to divide all users into groups i.e. clusters

Clustering

Items in a group must be **“similar”** to one another

Maximize intracluster similarity

Items in different groups must be **“dissimilar”** to one another

Minimize intercluster similarity

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Represent Data

Use meaningful numeric attributes to represent text

Term Frequency

TF-IDF

Features

Create a list representing the universe of all words that can appear in any text

(W_1, W_2, ... W_N)

(hello, this, is, the, universe, of, all, words, in, any, text, a, an, test, goodbye)

Any text can then be represented using the frequencies of these words

Features

Hello, this is a test

(hello, this, is, the, universe, of, all, words, in, any, text, a, an, test, goodbye)

(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0)

Term Frequency Representation

Features

Hello, this is a test

(hello, this, is, the, universe, of, all, words, in, any, text, a, an, test, goodbye)

(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0)

Information on the order of words is lost

Bag of Words Model

Term Frequency

Some words characterize a document more than others

The house was in New York

Term Frequency

The **house** was in **New York**

Words which occur more rarely, clearly differentiate a document from other documents

Term Frequency

The **house** was in **New York**

Words which are very common don’t do much to differentiate a document

Term Frequency - Inverse Document Frequency

Weight the term frequencies to take the rarity of a word into account

(hello, this, is, the, universe, of, all, words, in, any, text, a, an, test, goodbye)

$$\text{Weight} = \frac{1}{\#\text{ documents the word appears in}}$$

Term Frequency - Inverse Document Frequency

$$\text{Weight} = \frac{1}{\#\text{ documents the word appears in}}$$

TF-IDF

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

K-Means Clustering

Documents are represented using TF-IDF

Each document is a tuple of N Numbers

N is the total number of distinct words in all documents

K-Means Clustering

A tuple of N Numbers

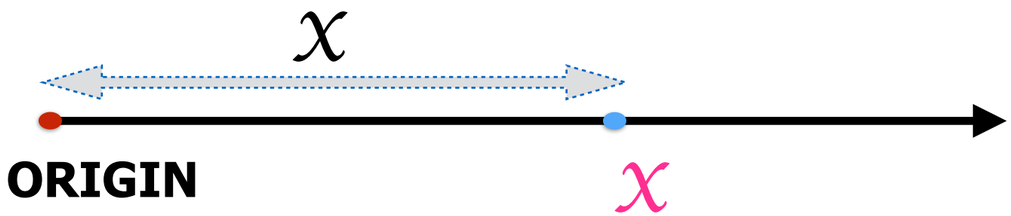
A point in an N-Dimensional Hypercube

N-Dimensional Hypercube

N-Dimensional Hypercube

A line is a 1-dimensional shape

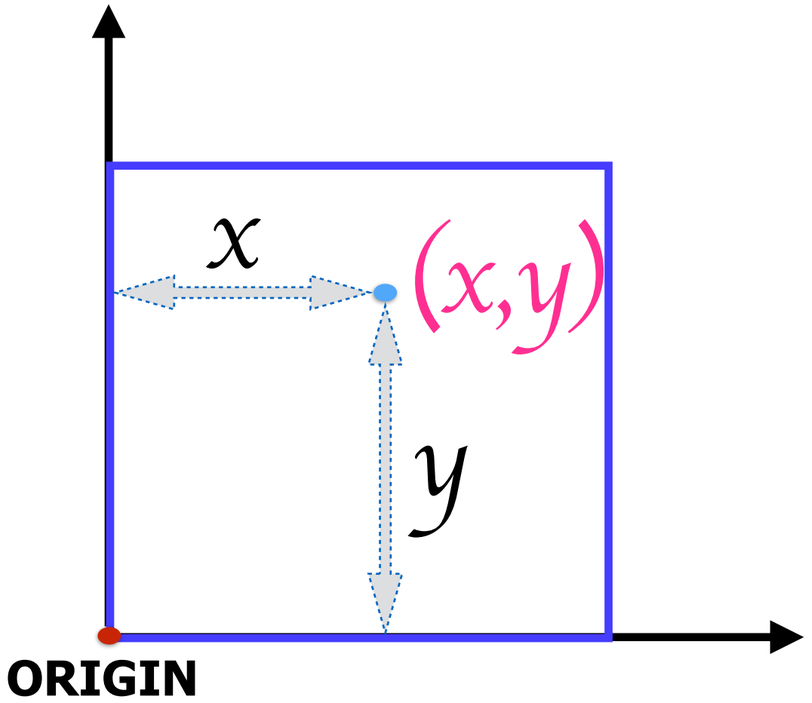
Any point on a line can be represented using 1 number



N-Dimensional Hypercube

A square is a 2-Dimensional shape

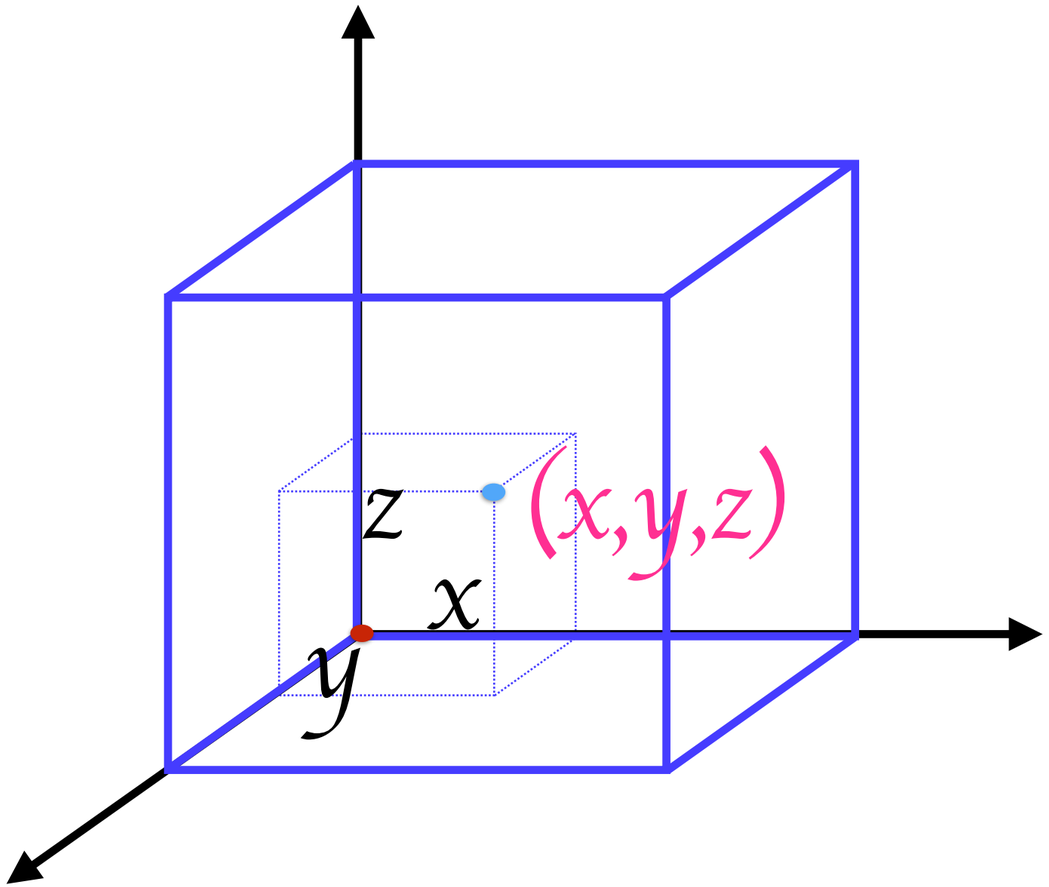
Any point in a square can be represented using 2 numbers



N-Dimensional Hypercube

A cube is a 3-dimensional shape

Any point in a cube can be represented with 3 numbers



N-Dimensional Hypercube

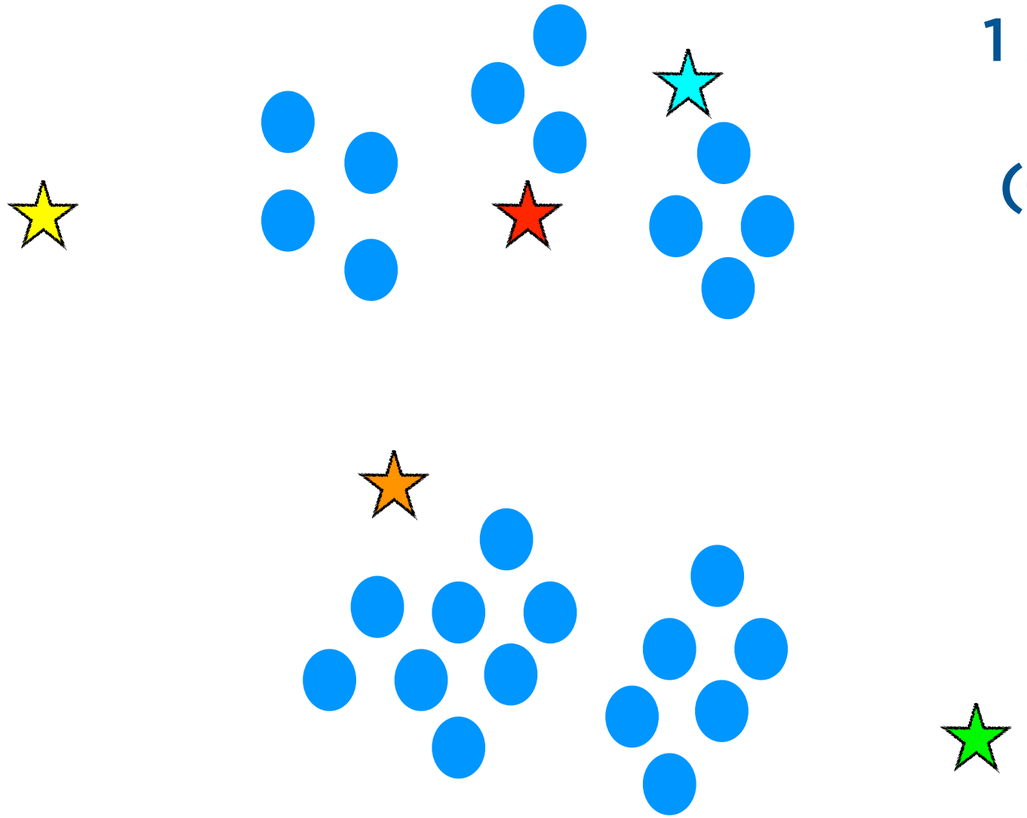
A set of N numbers represents a point in an N-Dimensional Hypercube

K-Means Clustering

A tuple of N Numbers

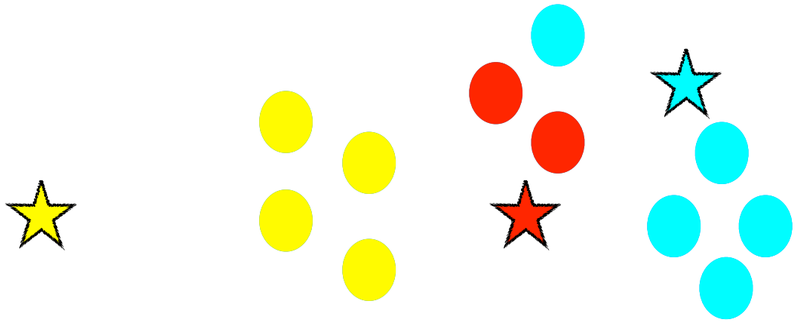
A point in an N-Dimensional Hypercube

K-Means Clustering

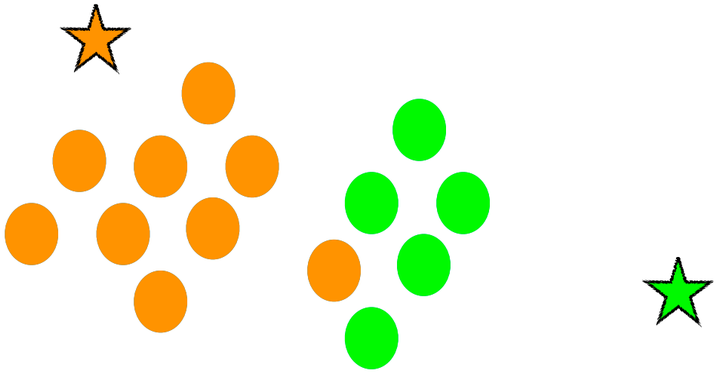


1 . Initialize a set of points as the “K” Means (Centroids of the clusters you want to find)

K-Means Clustering

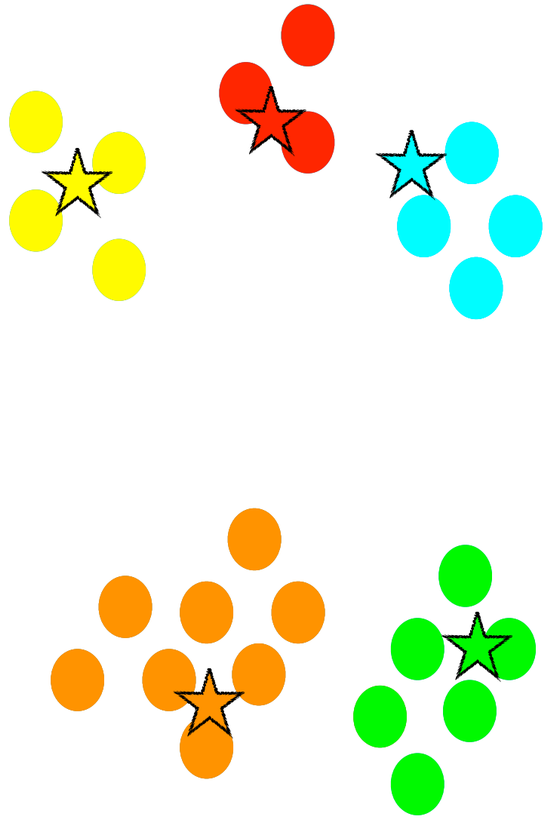


2. Assign each point to the cluster belonging to the nearest mean



3. Find the new means/ centroids of the clusters

Convergence



2. Assign each point to the cluster belonging to the nearest mean

3. Find the new means/ centroids of the clusters

Rinse and repeat steps 2,3 until the means don't change anymore

Demo

Cluster articles into groups representing different themes

Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Classifying Articles

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Start with a corpus of articles

Identify underlying themes

Assign themes to new articles

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Typical ML Workflow

Pick your Problem

Identify which type of problem we need to solve

Represent Data

Represent data using numeric attributes

Apply an Algorithm

Use a standard algorithm to find a model

Pick your Problem

We are given an article

Classify the article into one of the identified themes

Classification

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

“Train a model” using the training data

Test

“Test the model” using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Problem Statement

This classifier is like a black box

Machine Learning Objective

Build this black box

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Use the TF-IDF representation

Train

"Train a model" using the training data

Test

"Test the model" using test data

Typical Classification Setup

There are several standard algorithms to choose from

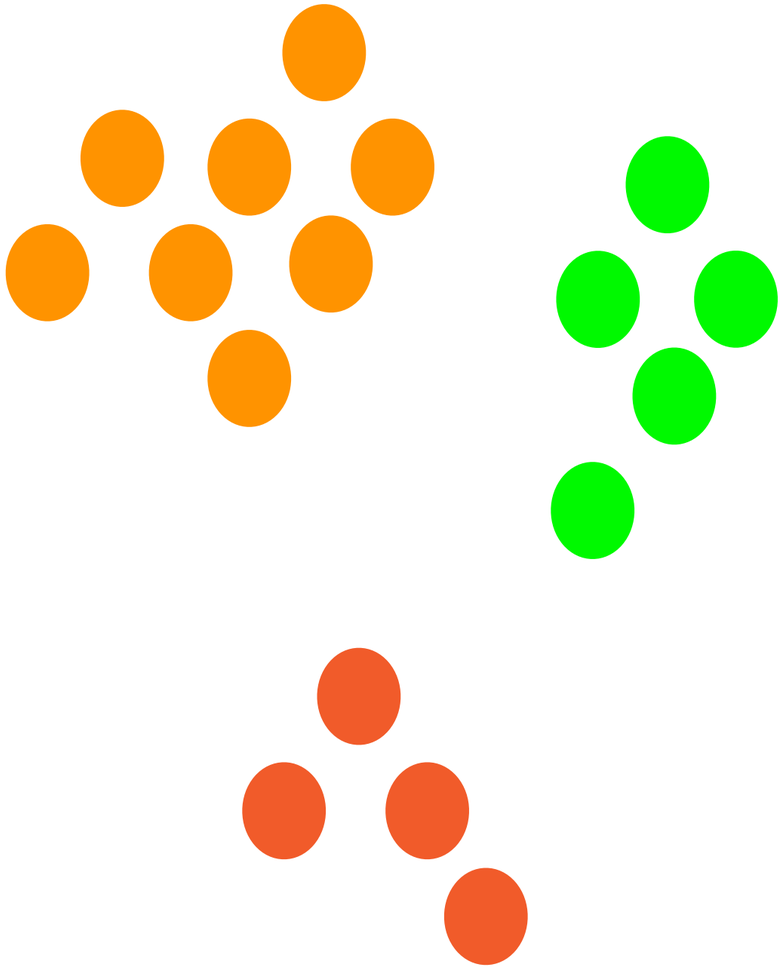
Training

"Train a model" using the training data

Test

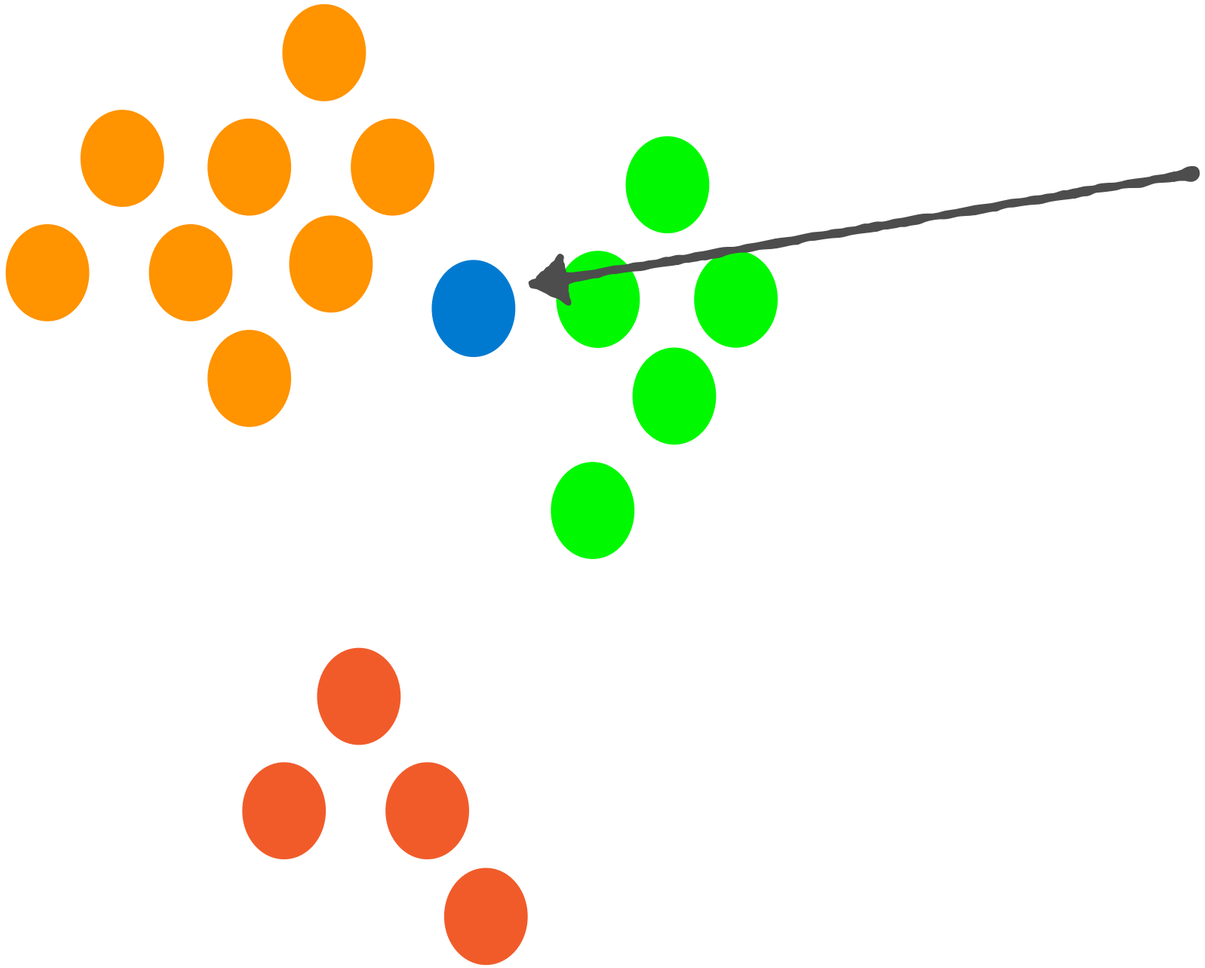
"Test the model" using test data

K-Nearest Neighbors



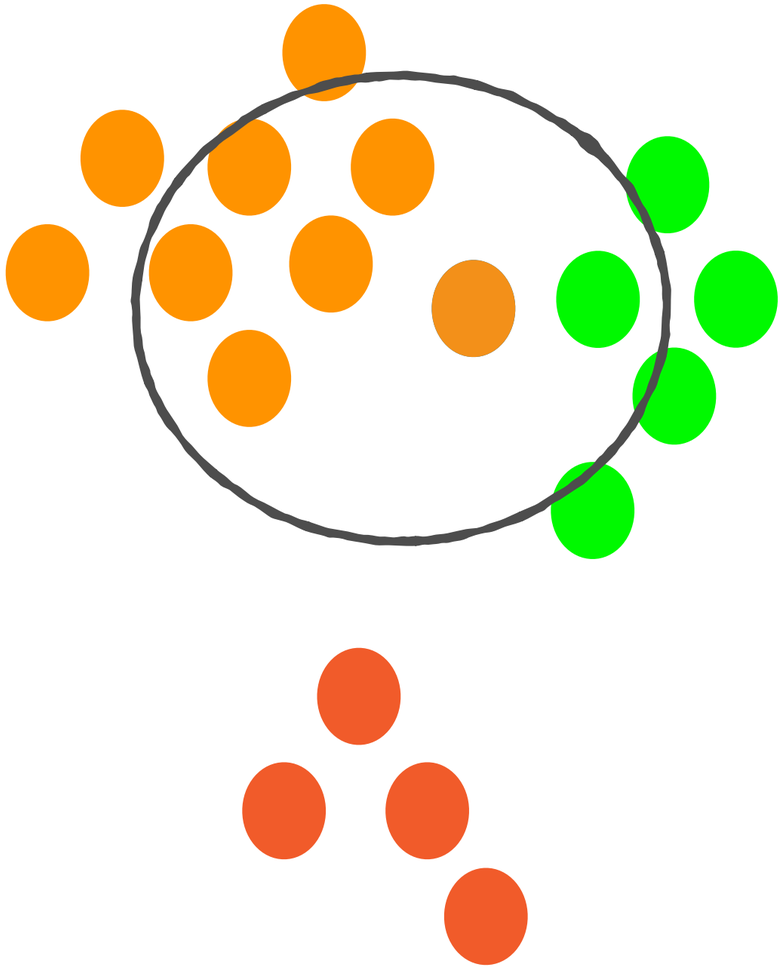
From the clustering step we have articles grouped in different themes

K-Nearest Neighbors



A new article to be classified

K-Nearest Neighbors



Find the K “nearest” neighbors

Take a majority vote

Demo

Classify a new article into one of the identified themes

Summary

Feature extraction using the bag of words model

Use K-Means clustering to identify a set of topics

Using the K-Nearest Neighbors model for classifying text into those topics