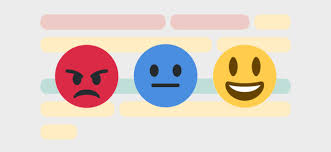
**Sentiment Analysis:**

****

Sentiment analysis is the process of detecting positive or negative sentiment in text. It’s often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers. Since customers express their thoughts and feelings more openly than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment.

## **Types of Sentiment Analysis:**

Sentiment analysis models focus on polarity (*positive, negative, neutral*) but also on feelings and emotions (*angry, happy, sad*, etc), urgency (*urgent, not urgent*) and even intentions (*interested v. not interested*).

Depending on how you want to interpret customer feedback and queries, you can define and tailor your categories to meet your sentiment analysis needs. In the meantime, here are some of the most popular types of sentiment analysis:

#### **Fine-grained Sentiment Analysis:**

If polarity precision is important to your business, you might consider expanding your polarity categories to include:

* Very positive
* Positive
* Neutral
* Negative
* Very negative

This is usually referred to as fine-grained sentiment analysis, and could be used to interpret 5-star ratings in a review, for example:

* Very Positive = 5 stars
* Very Negative = 1 star

#### **Emotion detection**

This type of sentiment analysis aims to detect emotions, like happiness, frustration, anger, sadness, and so on. Many emotion detection systems use lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms.

One of the downsides of using lexicons is that people express emotions in different ways. Some words that typically express anger, like bad or kill (e.g. your product is so bad or your customer support is killing me) might also express happiness (e.g. this is bad ass or you are killing it).

#### **Aspect-based Sentiment Analysis**

Usually, when analyzing sentiments of texts, let’s say product reviews, you’ll want to know which particular aspects or features people are mentioning in a positive, neutral, or negative way. That's where [aspect-based sentiment analysis](https://monkeylearn.com/blog/aspect-based-sentiment-analysis/) can help, for example in this text: "The battery life of this camera is too short", an aspect-based classifier would be able to determine that the sentence expresses a negative opinion about the feature battery life.

#### **Multilingual sentiment analysis**

Multilingual sentiment analysis can be difficult. It involves a lot of preprocessing and resources. Most of these resources are available online (e.g. sentiment lexicons), while others need to be created (e.g. translated corpora or noise detection algorithms), but you’ll need to know how to code to use them.

Alternatively, you could detect language in texts automatically with MonkeyLearn’s language classifier, then train a custom sentiment analysis model to classify texts in the language of your choice.

## **Why Is Sentiment Analysis Important?**

Sentiment analysis is extremely important because it helps businesses quickly understand the overall opinions of their customers. By automatically sorting the sentiment behind reviews, social media conversations, and more, you can make faster and more accurate decisions.

It’s estimated that [90% of the world’s data is unstructured](https://www.forbes.com/sites/bernardmarr/2019/10/16/what-is-unstructured-data-and-why-is-it-so-important-to-businesses-an-easy-explanation-for-anyone/?sh=d0192d615f64), in other words it’s unorganized. Huge volumes of [unstructured business data](https://monkeylearn.com/unstructured-data/) are created every day: emails, support tickets, chats, social media conversations, surveys, articles, documents, etc). But it’s hard to analyze for sentiment in a timely and efficient manner.

**The overall benefits of sentiment analysis include**:

* **Sorting Data at Scale**

Can you imagine manually sorting through thousands of tweets, customer support conversations, or surveys? There’s just too much business data to process manually. Sentiment analysis helps businesses process huge amounts of data in an efficient and cost-effective way.

* **Real-Time Analysis**

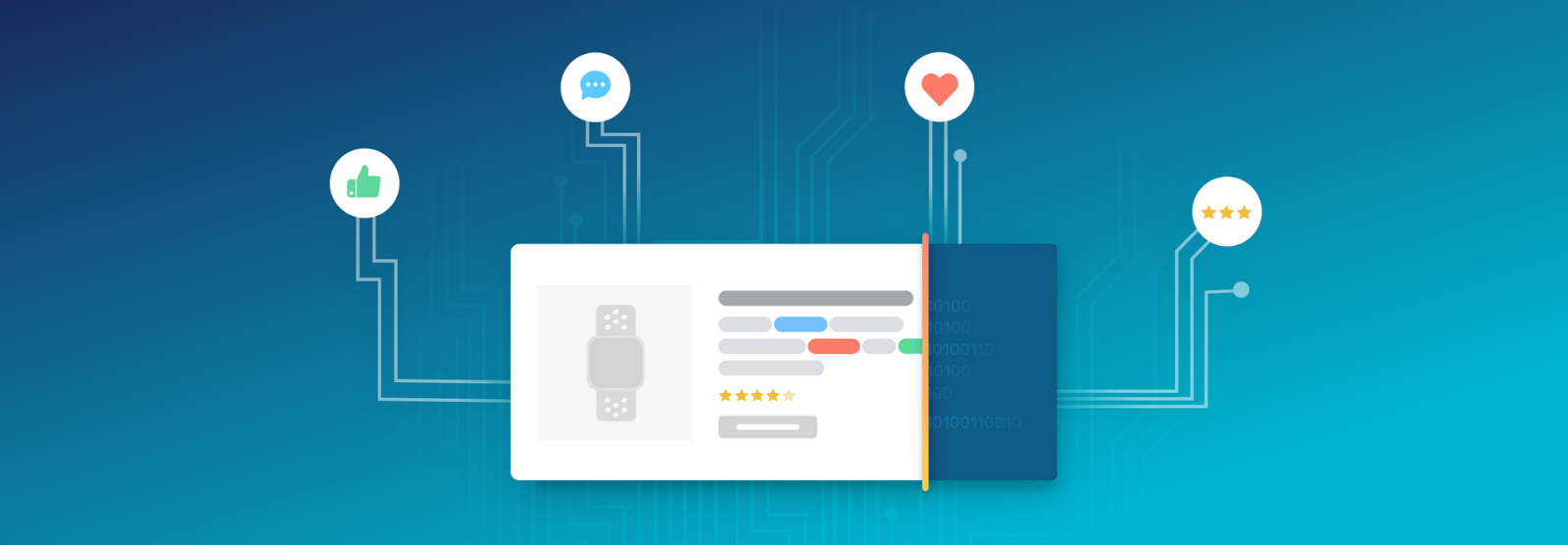
Sentiment analysis can identify critical issues in real-time, for example is a PR crisis on social media escalating? Is an angry customer about to churn? Sentiment analysis models can help you immediately identify these kinds of situations, so you can take action right away.

* **Consistent criteria**

It’s estimated that people only agree around 60-65% of the time when determining the sentiment of a particular text. Tagging text by sentiment is highly subjective, influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data, helping them improve accuracy and gain better insights.

## **How Does Sentiment Analysis Work?**

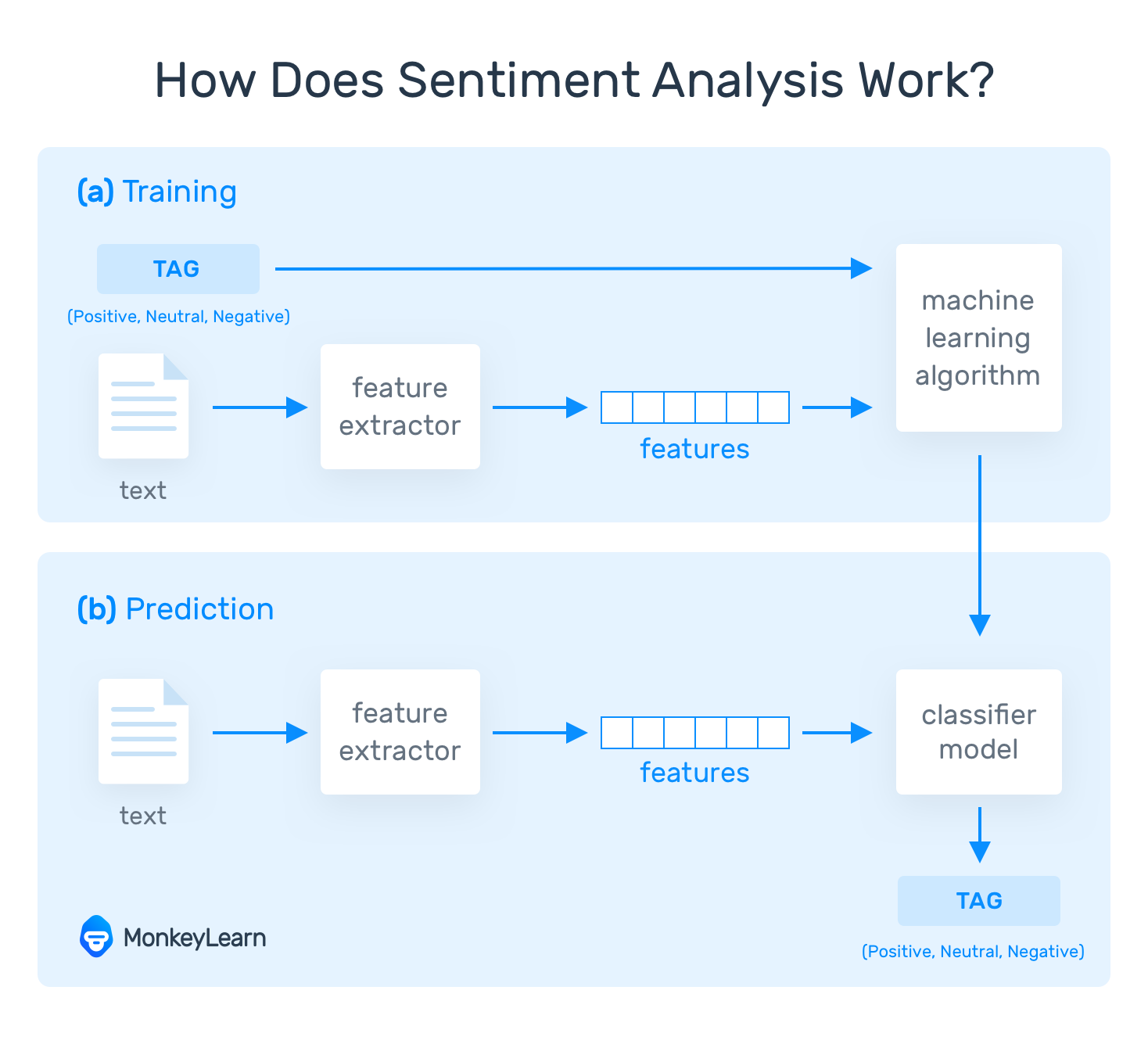
Sentiment analysis, otherwise known as opinion mining, works thanks to [natural language processing (NLP)](https://www.monkeylearn.com/natural-language-processing/) and [machine learning algorithms](https://monkeylearn.com/blog/machine-learning-algorithms/), to automatically determine the emotional tone behind online conversations.



There are different algorithms you can implement in sentiment analysis models, depending on how much data you need to analyze, and how accurate you need your model to be. We’ll go over some of these in more detail, below.

**Sentiment analysis algorithms fall into one of three buckets:**

* **Rule-based:** these systems automatically perform sentiment analysis based on a set of manually crafted rules.
* **Automatic:** systems rely on machine learning techniques to learn from data.
* **Hybrid** systems combine both rule-based and automatic approaches.

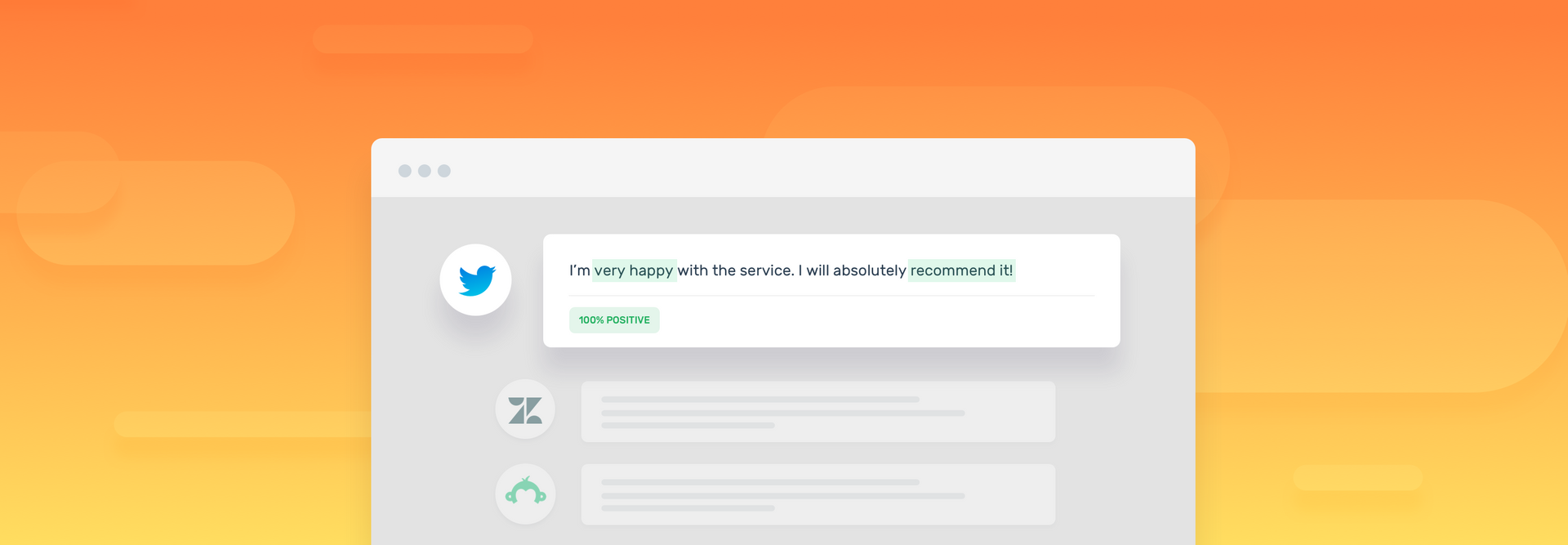


#### **Classification Algorithms:**

The classification step usually involves a statistical model like Naïve Bayes, Logistic Regression, Support Vector Machines, or Neural Networks:

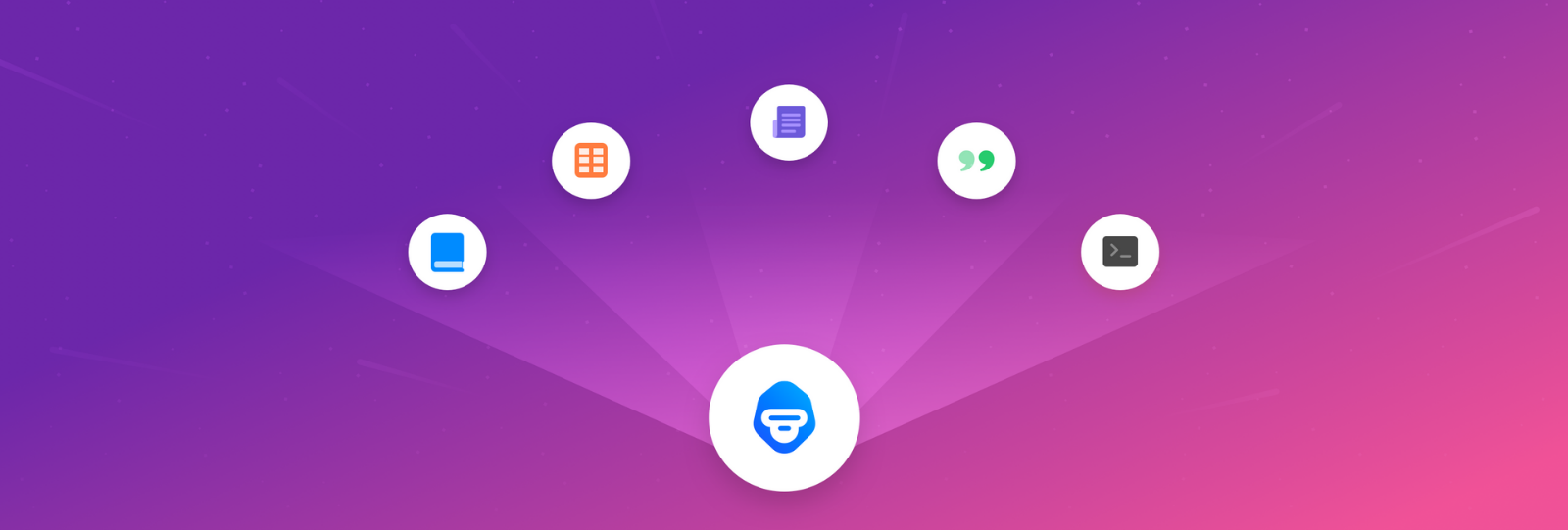
* [Naïve Bayes](https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/): a family of probabilistic algorithms that uses Bayes’s Theorem to predict the category of a text.
* [Linear Regression](https://hackernoon.com/supervised-machine-learning-linear-regression-in-python-541a5d8141ce): a very well-known algorithm in statistics used to predict some value (Y) given a set of features (X).
* [Support Vector Machines](https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/): a non-probabilistic model which uses a representation of text examples as points in a multidimensional space. Examples of different categories (sentiments) are mapped to distinct regions within that space. Then, new texts are assigned a category based on similarities with existing texts and the regions they’re mapped to.
* [Deep Learning](https://machinelearningmastery.com/what-is-deep-learning/): a diverse set of algorithms that attempt to mimic the human brain, by employing artificial neural networks to process data.

## **Sentiment Analysis Use Cases & Applications**



The [applications of sentiment analysis](https://monkeylearn.com/blog/sentiment-analysis-applications/) are endless and can be applied to any industry, from finance and retail to hospitality and technology. Below, we’ve listed some of the most popular ways that sentiment analysis is being used in business:

1. [Social Media Monitoring](https://monkeylearn.com/#sentiment-analysis-in-social-media-monitoring)
2. [Brand Monitoring](https://monkeylearn.com/#sentiment-analysis-in-brand-monitoring)
3. [Voice of customer (VoC)](https://monkeylearn.com/#sentiment-analysis-in-customer-feedback)
4. [Customer Service](https://monkeylearn.com/#sentiment-analysis-in-customer-support)
5. [Market Research](https://monkeylearn.com/#sentiment-analysis-in-market-research)



**All the required packages and libraries which are required for sentiment analysis**

**1.pandas:**

**pandas** (all lowercase) is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a [NumPy](https://www.educative.io/edpresso/how-to-create-an-array-in-numpy) matrix array. This makes pandas a trusted ally in data science and machine learning.

Similar to NumPy, pandas deals primarily with data in 1-D and 2-D arrays; however, pandas handles the two differently.

**Creating a**[**Series**](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.html#pandas.Series)**by passing a list of values, letting pandas create a default integer index:**

**In [3]:** s = pd.Series([1, 3, 5, np.nan, 6, 8])

**In [4]:** s

**Out[4]:**

0 1.0

1 3.0

2 5.0

3 NaN

4 6.0

5 8.0

dtype: float64

**2.pyplot:**

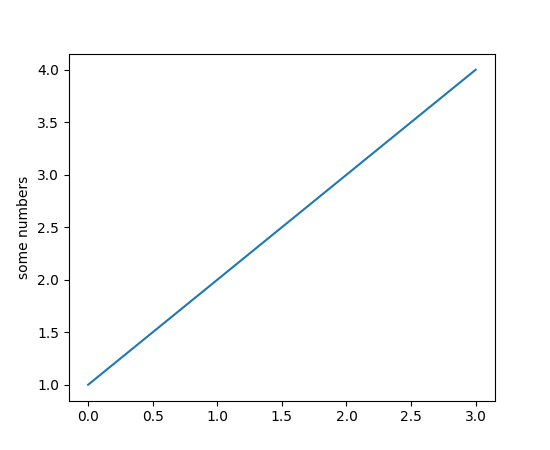
## [matplotlib.pyplot](https://matplotlib.org/2.0.2/api/pyplot_api.html#module-matplotlib.pyplot) is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In [matplotlib.pyplot](https://matplotlib.org/2.0.2/api/pyplot_api.html" \l "module-matplotlib.pyplot" \o "matplotlib.pyplot) various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes .

**import** **matplotlib.pyplot** **as** **plt**

plt.plot([1,2,3,4])

plt.ylabel('some numbers')

plt.show()



## 3.Scikit Learn

Scikit-learn is a machine learning library for Python. It features several regression, classification and clustering algorithms including SVMs, gradient boosting, k-means, random forests and DBSCAN. It is designed to work with Python [Numpy](https://www.journaldev.com/15646/python-numpy-tutorial) and [SciPy](https://www.journaldev.com/18106/python-scipy-tutorial).

Scikit is written in Python (most of it) and some of its core algorithms are written in Cython for even better performance.

Scikit-learn is used to build models and it is not recommended to use it for reading, manipulating and summarizing data as there are better frameworks available for the purpose.

It is open source and released under BSD license.

4.[re](https://docs.python.org/3/library/re.html#module-re) — Regular expression operations

A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

Regular expressions can be concatenated to form new regular expressions; if A and B are both regular expressions, then AB is also a regular expression.

**5. NLTK(**Natural Language Tool Kit**):**

NLTK is a leading platform for building Python programs to work with human language data.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

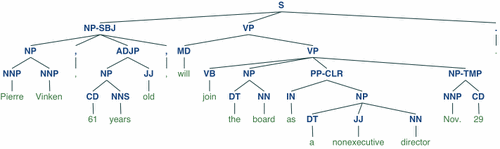
[Natural Language Processing with Python](http://nltk.org/book) provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3.

Display a parse tree:

**>>> from** **nltk.corpus** **import** treebank

**>>>** t = treebank.parsed\_sents('wsj\_0001.mrg')[0]

**>>>** t.draw()



**6.tkinter:**

The [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at ActiveState.)

Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

### **A Simple Hello World Program**

**import** **tkinter** **as** **tk**

**class** **Application**(tk.Frame):

**def** \_\_init\_\_(self, master=**None**):

super().\_\_init\_\_(master)

self.master = master

self.pack()

self.create\_widgets()

**def** create\_widgets(self):

self.hi\_there = tk.Button(self)

self.hi\_there["text"] = "Hello World**\n**(click me)"

self.hi\_there["command"] = self.say\_hi

self.hi\_there.pack(side="top")

self.quit = tk.Button(self, text="QUIT", fg="red",

command=self.master.destroy)

self.quit.pack(side="bottom")

**def** say\_hi(self):

print("hi there, everyone!")

root = tk.Tk()

app = Application(master=root)

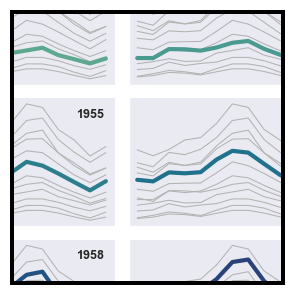
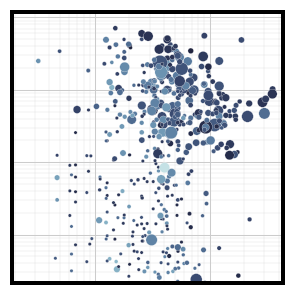
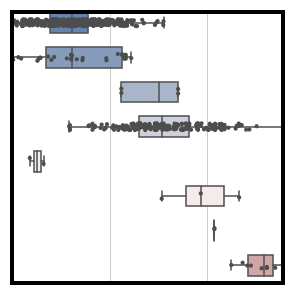
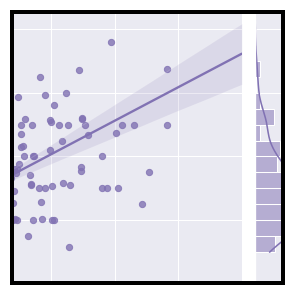
app.mainloop()

OUTPUT:



# **7.seaborn:**

Seaborn is a Python data visualization library based on [matplotlib](https://matplotlib.org/). It provides a high-level interface for drawing attractive and informative statistical graphics.

Pre-processing the text data
8
can u he.lp me with loan? 😊
Unintentional
Characters
Abbreviations Symbols Emojis
can you h...

Pre-processing the text data
9
➢ Removing weird spaces
➢ Tokenization
➢ Spelling correction
➢ Contraction mapping
➢ Stemmi...

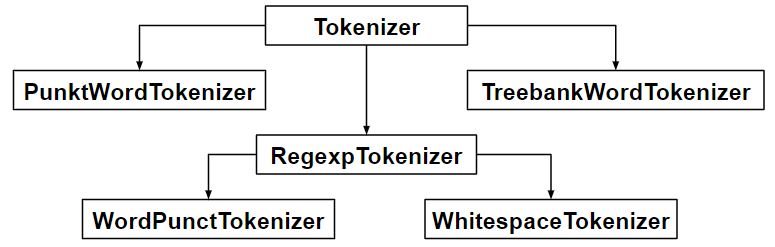
Basic modelling
Tabular Data
(Basic
Features)
Training Set
Validation Set
Logistic
Regression
XGB
Normalization
0.658
0.721
 

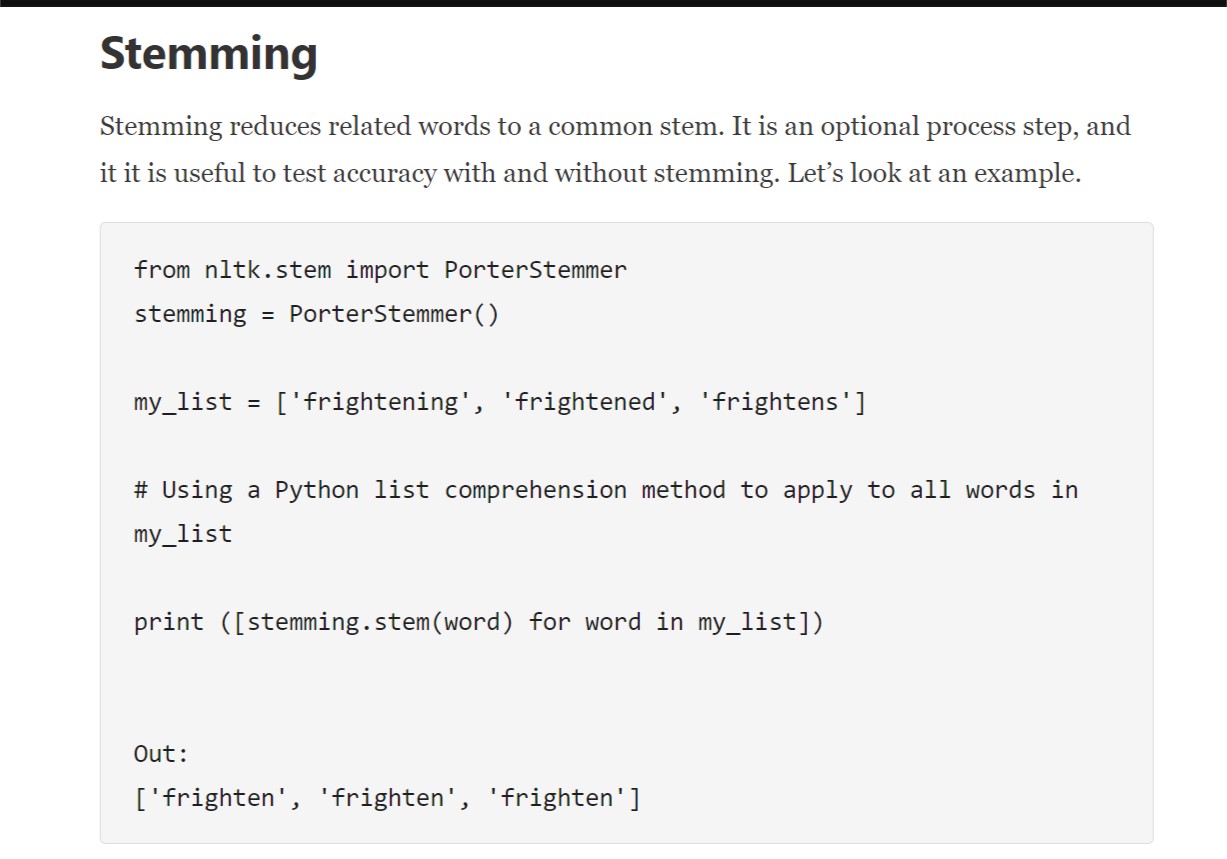
# Pre-processing data: tokenization, stemming, and removal of stop words

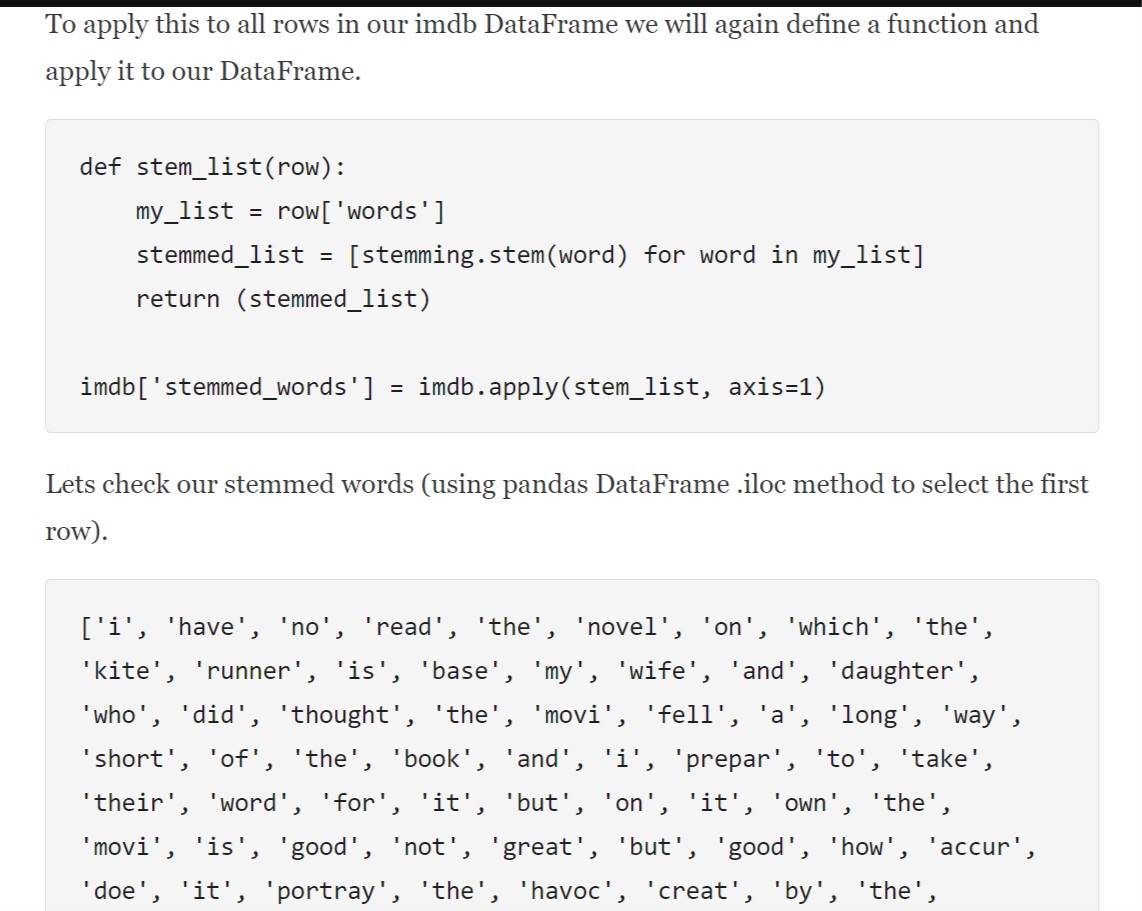
**Tokenization**is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

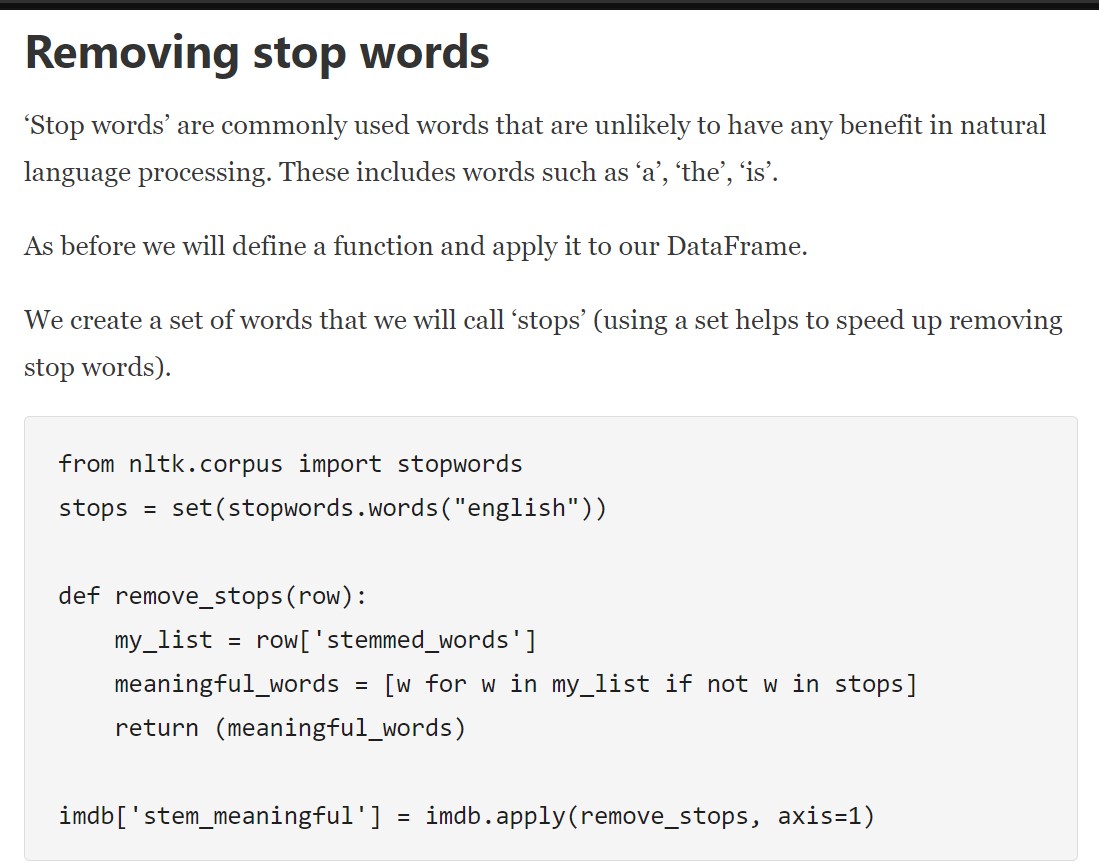
**Key points of the article –**

* Text into sentences tokenization
* Sentences into words tokenization
* Sentences using regular expressions tokenization









Show the stemmed words, without stop words, from the first record.





# **Methods for sentiment analysis:**

# 1.Exploratory Data Analysis(EDA):

# **Intuition**

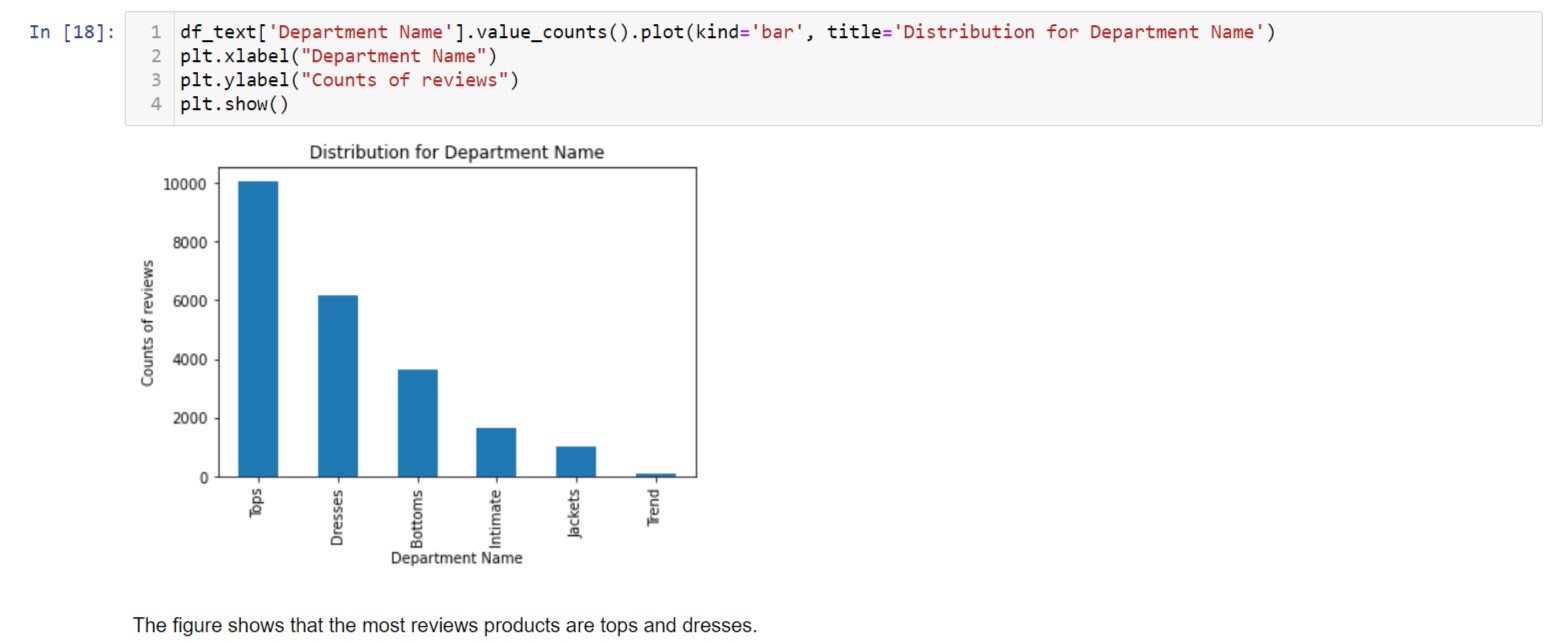
Imagine your wolf pack decides to watch a movie you haven’t heard of.There is absolutely no debate about that,it will lead to a state where you find yourself puzzled with lot of questions which needs to be answered in order to make a decision.Being a good chieftain the first question you would ask, what is the cast and crew of the movie?As a regular practice,you would also watch the trailer of the movie on YouTube.Furthermore,you’d find out ratings and reviews the movie has received from the audience.

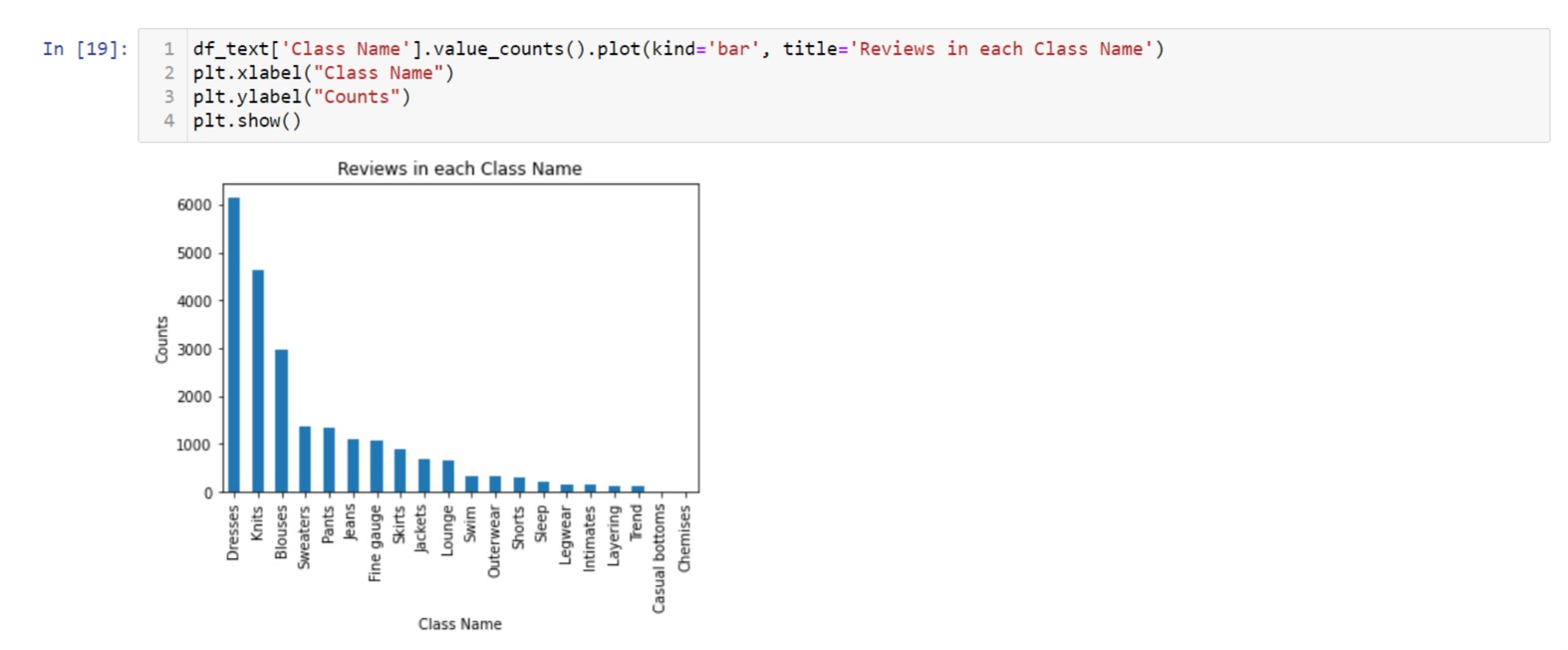


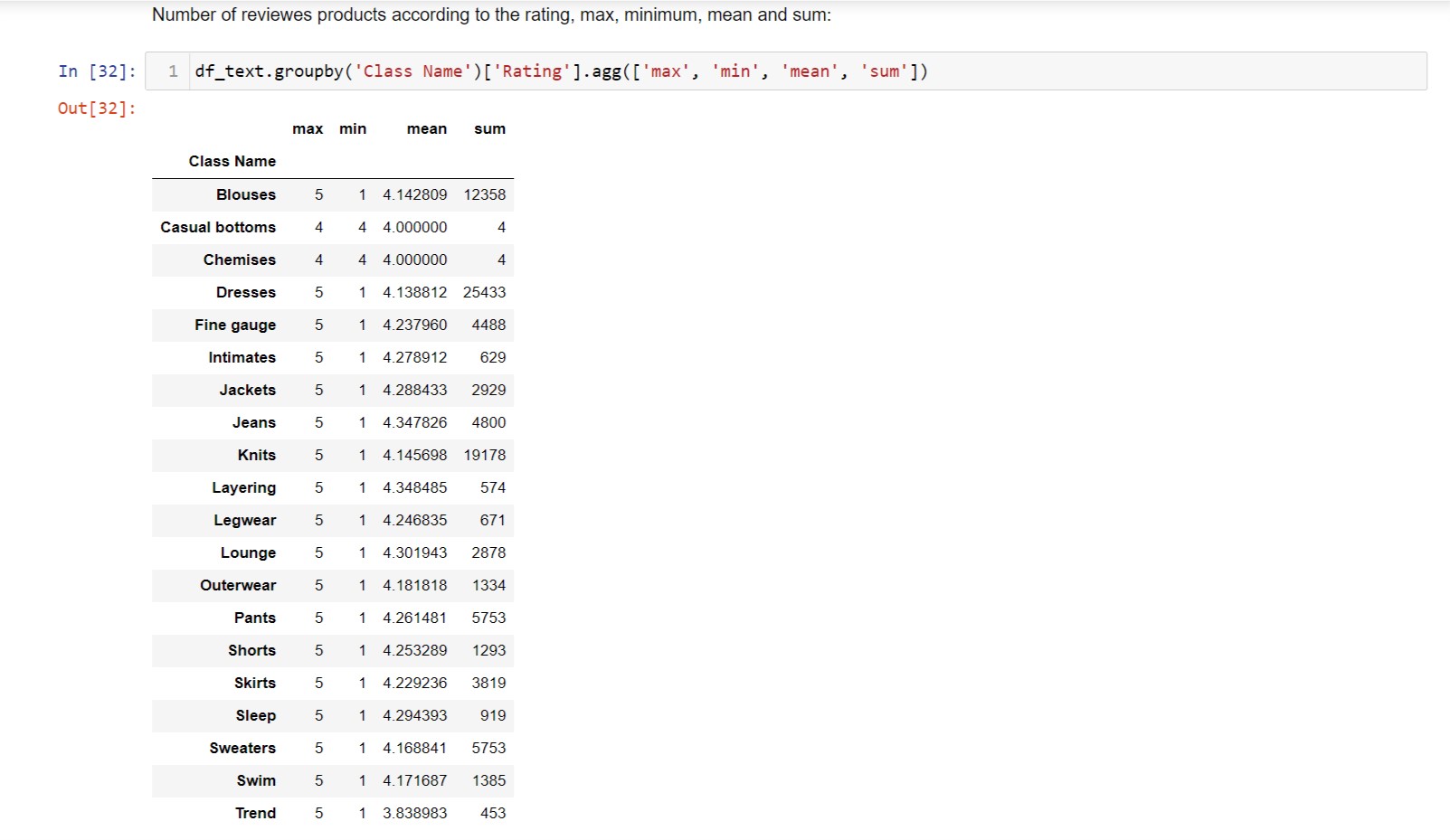
Whatever investigating measures you would take before finally buying popcorn for your clan in theater,is nothing but what data scientists in their lingo call ‘Exploratory Data Analysis’.

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

**Conclusion:**

This show that constumers are sticky to tops if we talking about Department name.But if I go to under the Class Name then dresses are more demanding after dresses kurtis then blouses and so on.





Above figure shows that blouses which has highest mean has maximum rating 5 and minimun is 1 and so on..

What is Naive Bayes algorithm?

It is a [classification technique](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=6stepsnaivebayesarticle) based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

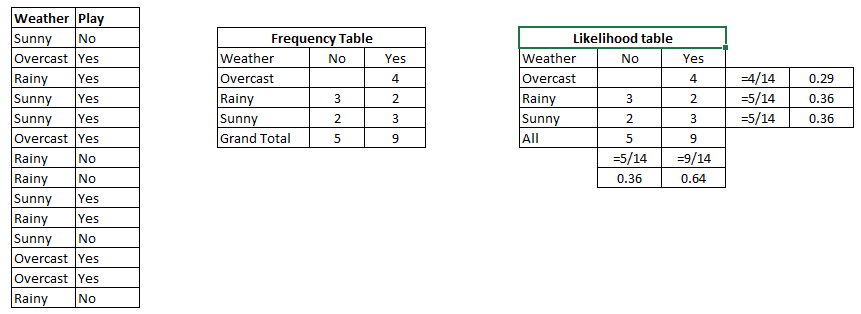


* *P*(*c|x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
* *P*(*c*) is the prior probability of *class*.
* *P*(*x|c*) is the likelihood which is the probability of *predictor* given *class*.
* *P*(*x*) is the prior probability of *predictor*.

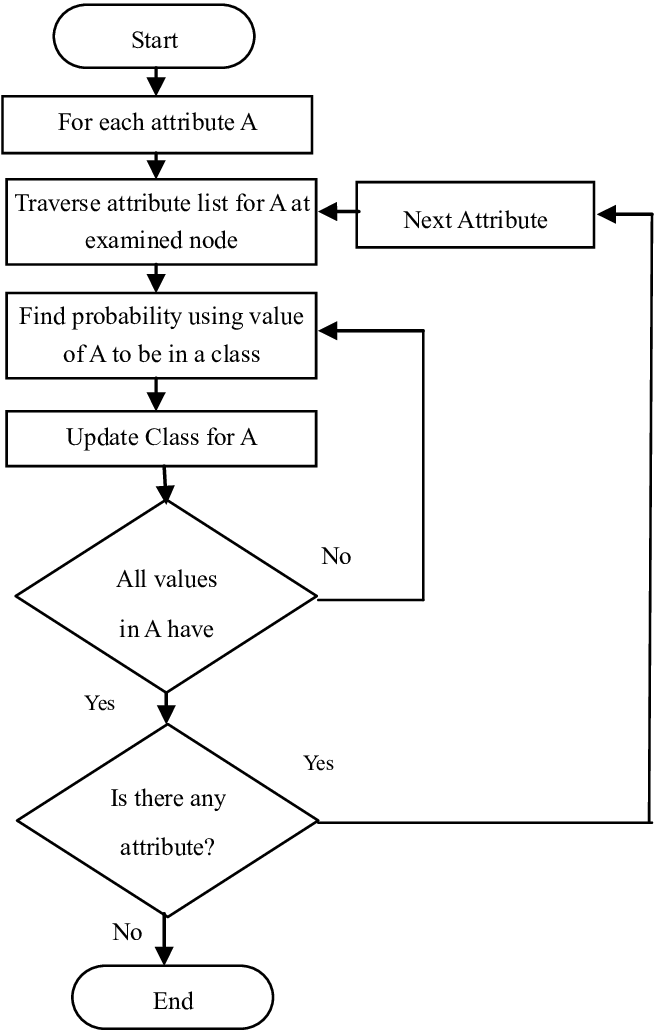
## How Naive Bayes algorithm works?

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

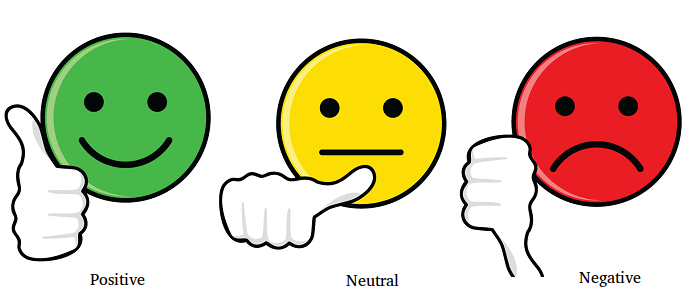
[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Bayes_41.png)

Step 3: Now, use [Naive Bayesian](https://courses.analyticsvidhya.com/courses/naive-bayes?utm_source=blog&utm_medium=naive-bayes-explained) equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.



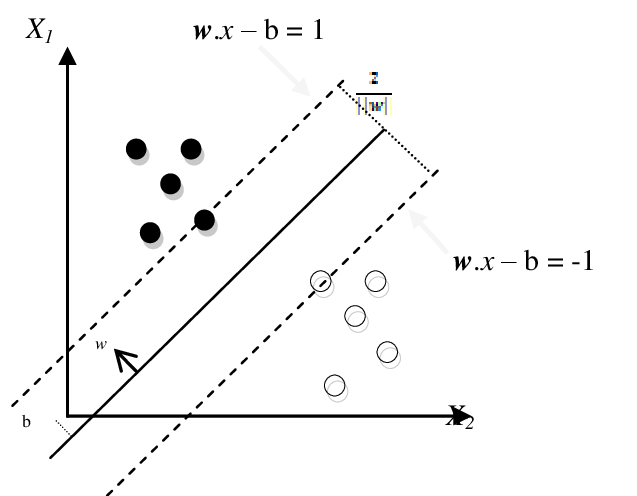
# Sentiment Analysis using SVM

**Sentiment Analysis** is the **NLP** technique that performs on the text to determine whether the author’s intentions towards a particular topic, product, etc. are **positive, negative, or neutral.**



# **What is SVM?**

SVM is a **supervised(feed-me)**machine learning algorithm that can be used for both classification or regression challenges. Classification is predicting a label/group and Regression is predicting a continuous value. SVM performs classification by finding the hyper-plane that differentiate the classes we plotted in n-dimensional space.



SVM draws that hyperplane by transforming our data with the help of mathematical functions called **“Kernels”.**Types of Kernels are**linear, sigmoid, RBF, non-linear, polynomial, etc.,**

AdaBoost Ensemble:

Boosting is a relatively young, yet extremely powerful, machine learning technique. The main idea behind boosting algorithms is to combine multiple weak learners – classification algorithms that perform only slightly better than random guessing – into a powerful composite classifier. Our focus is on the well known AdaBoost algorithm (Freund and Schapire, 1997) based on Multinomial Naive Bayes as base classifiers . AdaBoost and its variants have been applied to diverse domains with great success, owing to their solid theoretical foundation, accurate prediction, and great simplicity (Freund and Schapire, 1997).

For example, Viola and Jones (2001) used AdaBoost to face detection, Hao and Luo (2006) dealt with image segmentation, recognition of handwritten digits, and outdoor scene classification problems. In (Bloehdorn and Hotho, 2004) text classification is explored.

RandomForestClassifier in Sentiment Analysis:

The sklearn.ensemble module contains the RandomForestClassifier class that can be used to train the machine learning model using the random forest algorithm. To do so, we need to call the fit method on the RandomForestClassifier class and pass it our training features and labels, as parameters. Look at the following script:

from sklearn.ensemble import RandomForestClassifier

text\_classifier = RandomForestClassifier(n\_estimators=200, random\_state=0)

text\_classifier.fit(X\_train, y\_train)

#### Making Predictions and Evaluating the Model

Once the model has been trained, the last step is to make predictions on the model. To do so, we need to call the predict method on the object of the RandomForestClassifier class that we used for training. Look at the following script:

predictions = text\_classifier.predict(X\_test)

Finally, to evaluate the performance of the machine learning models, we can use classification metrics such as a [confusion metrix](https://en.wikipedia.org/wiki/Confusion_matrix), [F1 measure](https://en.wikipedia.org/wiki/F1_score), accuracy, etc.

To find the values for these metrics, we can use classification\_report, confusion\_matrix, and accuracy\_score utilities from the sklearn.metrics library. Look a the following script:

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

print(confusion\_matrix(y\_test,predictions))

print(classification\_report(y\_test,predictions))

print(accuracy\_score(y\_test, predictions))

The output of the script above looks like this:

[[1724 101 45]

[ 329 237 48]

[ 142 58 244]]

precision recall f1-score support

negative 0.79 0.92 0.85 1870

neutral 0.60 0.39 0.47 614

positive 0.72 0.55 0.62 444

micro avg 0.75 0.75 0.75 2928

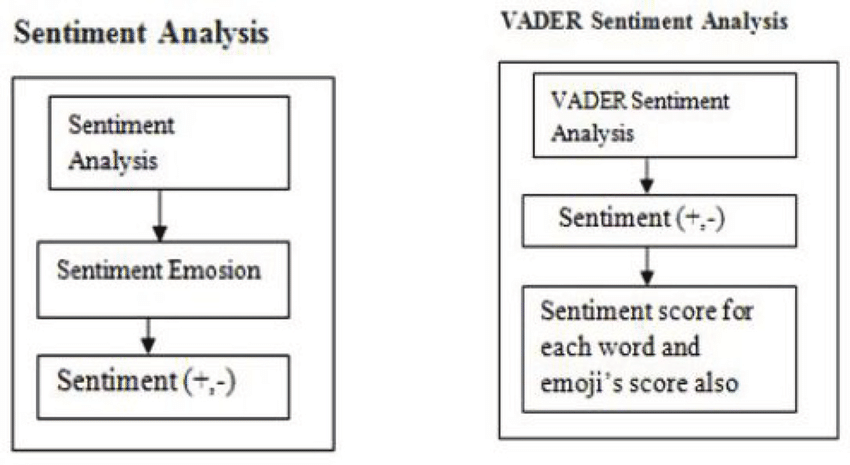
macro avg 0.70 0.62 0.65 2928

weighted avg 0.74 0.75 0.73 2928

0.7530737704918032

From the output, you can see that our algorithm achieved an accuracy of 75.30.

# Sentiment Analysis Made Easy Using VADER





## **Introducing VADER**

VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.

It is used for sentiment analysis of text which has both the polarities i.e. positive/negative. VADER is used to quantify how much of positive or negative emotion the text has and also the intensity of emotion.

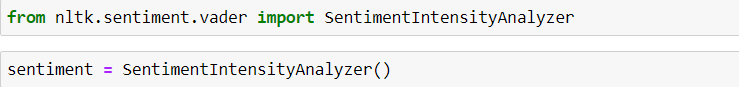
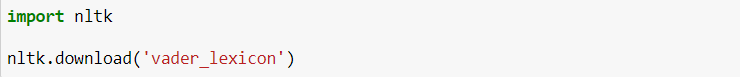
## **Advantages**

Here are the advantages of using VADER which makes a lot of things easier:

* It does not require any training data.
* It can very well understand the sentiment of a text containing emoticons, slangs, conjunctions, capital words, punctuations and much more.
* It works excellent on social media text.
* VADER can work with multiple domains.

Now, let’s start writing our script:

Now, let’s start writing our script:

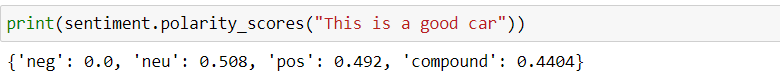


Here, **SentimentIntensityAnalyzer()** is an object and **polarity\_scores** is a method which will  give us scores of the following categories:

* Positive
* Negative
* Neutral
* Compound

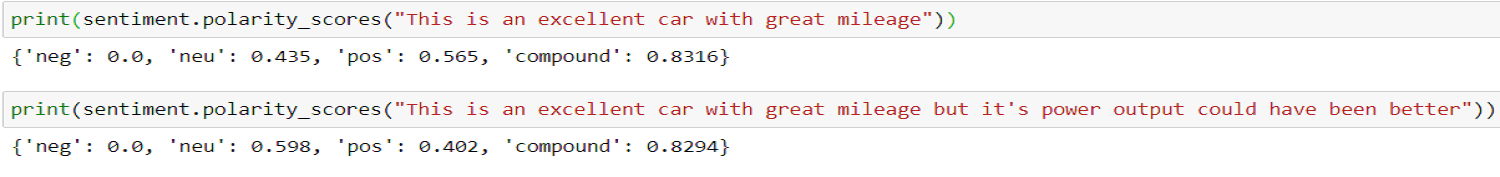
The compound score is the sum of positive, negative & neutral scores which is then normalized between -1(most extreme negative) and +1 (most extreme positive).

The more Compound score closer to +1, the higher the positivity of the text.



Above text is 49.2% Positive, 0% Negative, 50.8% Neutral. While the compound score is 44.04%.

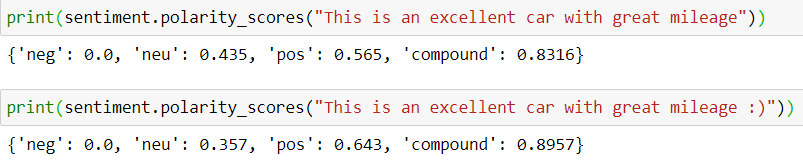
## **Let’s look at some examples**

**Example-1** 

In the above scenario we can see that with the use of conjunction in a sentence, the positive & compound score has decreased.

——————————————————————————————————————————-

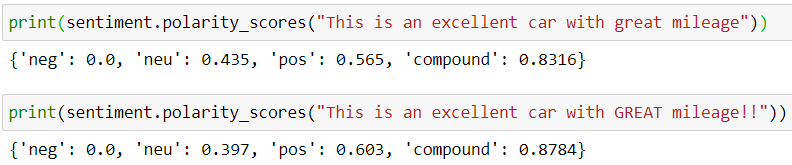
**Example-2**



In this scenario, with the use of smiley (emoji) in a sentence, the positive score and compound score has increased.

——————————————————————————————————————————-

**Example-3**



Here we can see that with the use of capital word & exclamation mark, the positive score & compound score has increased.

The second sentence is more intense, positive and so the compound score of the second sentence is more.

The compound score will increase as the intensity of the text will increase towards positive.

## **Conclusion:**

VADER classifies the sentiments very well. It is easy to use, the ready-made model which can be used across multiple domains, social-media texts, [analysing reviews](https://analyticsindiamag.com/neurips-is-once-again-amid-controversies-due-to-terrible-reviews/) etc. The cherry on the cake when using VADER is it does not require any [training data](https://analyticsindiamag.com/how-do-data-scientists-create-high-quality-training-datasets-for-computer-vision/). Well, we can see that the results obtained are very excellent!!

-----------------------THANK YOU-----------------------

By: Kanishka Kannoujia**(CT20192666916)**