Unit I Text Data Analytics

Text Mining: Text Pre-Processing, Shallow Parsing, Stop words, Stemming and Lemmatizing and Word Cloud. Sentiment Analysis: Textblob, nrc, VADER, afinn

Unit II Machine Leaning for Text Data

Recommendation System for Movies- Euclidean distances, Manhattan distance, Cosine distances, Cosine similarity, Unsupervised Machine Learning for Grouping Similar Text- K-means and Hierarchical Clustering, Supervised Machine Learning for Classification Problems - Naïve Bayes and Support Vector Machines

Unit III Image Data Analytics

Image Acquisition and Representation, Image Resizing and Rescaling, Image Rotation, Image Intensity, Image Cropping, Edge Extraction using Sobel Filter, Edge Extraction using Prewitt Filter

Unit IV Machine Learning for Image Data

Recommendation system for Images - Euclidean distances, Manhattan distance, Cosine distances, Cosine similarity, Unsupervised Machine Learning for Grouping Similar Images- K-means and Hierarchical Clustering, Supervised Machine Learning for Classification Problems - Naïve Bayes and Support Vector Machines, Object detection

Unit I Text Data Analytics

Text Mining: Text Pre-Processing, Shallow Parsing, Stop words, Stemming and Lemmatizing and Word Cloud. Sentiment Analysis: Textblob, nrc, VADER, afinn

Text Processing Library

Natural Language Toolkit(NLTK)
GenSim
SpaCy
CoreNLP
TextBlob
AllenNLP
polyglot
scikit-learn

Python libraries for sentiment analysis:

1. Pattern

Topping our list of best Python libraries for sentiment analysis is Pattern, which is a multipurpose Python library that can handle NLP, data mining, network analysis, machine learning, and visualization.

Pattern provides a wide range of features, including finding superlatives and comparatives. It can also carry out fact and opinion detection, which make it stand out as a top choice for sentiment analysis. The function in Pattern returns polarity and the subjectivity of a given text, with a Polarity result ranging from highly positive to highly negative.

Here are some of the main features of Pattern:

Multipurpose library
Finding superlatives and comparatives
Returns polarity and subjectivity of given text
Polarity range from highly positive to highly negative

2. VADER

Another top option for sentiment analysis is VADER (Valence Aware Dictionary and sEntiment Reasoner), which is a rule/lexicon-based, open-source sentiment analyzer pre-built library within NLTK. The tool is specifically designed for sentiments expressed in social media, and it uses a combination of A sentiment lexicon and a list of lexical features that are generally labeled according to their semantic orientation as positive or negative.

VADER calculates the text sentiment and returns the probability of a given input sentence to be positive, negative, or neural. The tool can analyze data from all sorts of social media platforms, such as Twitter and Facebook.

Here are some of the main features of VADER:

Does not require training data
Understand sentiment of text containing emoticons, slangs, conjunctions, etc.
Excellent for social media text
Open-source library

3. BERT

BERT (Bidirectional Encoder Representations from Transformers) is a top machine learning model used for NLP tasks, including sentiment analysis. Developed in 2018 by Google, the library was trained on English WIkipedia and BooksCorpus, and it proved to be one of the most accurate libraries for NLP tasks.

Because BERT was trained on a large text corpus, it has a better ability to understand language and to learn variability in data patterns.

Here are some of the main features of BERT:

Easy to fine tune
Wide range of NLP tasks, including sentiment analysis
Trained on a large corpus of unlabeled text
Deeply bidirectional model

4. TextBlob

TextBlob is another great choice for sentiment analysis. The simple Python library supports complex analysis and operations on textual data. For lexicon-based approaches, TextBlob defines a sentiment by its semantic orientation and the intensity of each word in a sentence, which requires a pre-defined dictionary classifying negative and positive words. The tool assigns individual scores to all the words, and a final sentiment is calculated.

TextBlob returns polarity and subjectivity of a sentence, with a Polarity range of negative to positive. The library's semantic labels help with analysis, including emoticons, exclamation marks, emojis, and more.

Here are some of the main features of TextBlob:

Simple Python library
Supports complex analysis and operations on textual data
Assigns individual sentiment scores
Returns polarity and subjectivity of sentence

5. spaCy

An open-source NLP library, spaCy is another top option for sentiment analysis. The library enables developers to create applications that can process and understand massive volumes of text, and it is used to construct natural language understanding systems and information extraction systems.

With spaCy, you can carry out sentiment analysis to collect insightful information about your products or brand from a wide range of sources, such as emails, social media, and product reviews.

Here are some of the main features of SpaCy:

Fast and easy-to-use
Great for beginner developers
Process massive volumes of text
Sentiment analysis with wide range of sources

6. CoreNLP

Stanford CoreNLP is another Python library containing a variety of human language technology tools that help apply linguistic analysis to text. CoreNLP incorporates Stanford NLP tools, including sentiment analysis. It also supports five languages in total: English, Arabic, German, Chinese, French, and Spanish.

The sentiment tool includes various programs to support it, and the model can be used to analyze text by adding “sentiment” to the list of annotators. It also includes a command line of support and model training support.

Here are some of the main features of CoreNLP:

Incorporates Stanford NLP tools
Supports five languages
Analyzes text by adding “sentiment”
Command line of support and model training support

7. scikit-learn

A standalone Python library on Github, scikit-learn was originally a third-party extension to the SciPy library. While it is especially useful for classical machine learning algorithms like those used for spam detection and image recognition, scikit-learn can also be used for NLP tasks, including sentiment analysis.

The Python library can help you carry out sentiment analysis to analyze opinions or feelings through data by training a model that can output if text is positive or negative. It provides several vectorizers to translate the input documents into vectors of features, and it comes with a number of different classifiers already built-in.

Here are some of the main features of scikit-learn:

Built on SciPy and NumPy
Proven with real-life applications
Diverse range of models and algorithms
Used by big companies like Spotify

8. Polyglot

One more great choice for sentiment analysis is Polyglot, which is an open-source Python library used to perform a wide range of NLP operations. The library is based on Numpy and is incredibly fast while offering a large variety of dedicated commands.

One of the top selling points of Polyglot is that it supports extensive multilingual applications. According to its documentation, it supports sentiment analysis for 136 languages. It is known for its efficiency, speed, and straightforwardness. Polyglot is often chosen for projects that involve languages not supported by spaCy.

Here are some of the main features of Polyglot:

Multilingual with 136 languages supported for sentiment analysis
Built on top of NumPy
Open-source
Efficient, fast, and straightforward

9. PyTorch

Nearing the end of our list is PyTorch, another open-source Python library. Created by Facebook's AI research team, the library enables you to carry out many different applications, including sentiment analysis, where it can detect if a sentence is positive or negative.

PyTorch is extremely fast in execution, and it can be operated on simplified processors or CPUs and GPUs. You can expand on the library with its powerful APIs, and it has a natural language toolkit.

Here are some of the main features of PyTorch:

Cloud platform and ecosystem
Robust framework
Extremely fast
Can be operated on simplified processors, CPUs, or GPUs

10. Flair

Closing out our list of 10 best Python libraries for sentiment analysis is Flair, which is a simple open-source NLP library. Its framework is built directly on PyTorch, and the research team behind Flair has released several pre-trained models for a variety of tasks.

One of the pre-trained models is a sentiment analysis model trained on an IMDB dataset, and it’s simple to load and make predictions. You can also train a classifier with Flair using your dataset. While it is a useful pre-trained model, the data it is trained on might not generalize as well as other domains, such as Twitter.

Here are some of the main features of Flair:

Open-source
Supports a number of languages
Simple to use
Several pre-trained models, including sentiment analysis

11. NRC Library

NRCLexicon is an MIT-approved pypi project by Mark M. Bailey which predicts the sentiments and emotion of a given text. The package contains approximately 27,000 words and is based on the National Research Council Canada (NRC) affect lexicon and the NLTK library's WordNet synonym sets.

Apply methods to classify emotions.

Sr.	Method	Description
1	emotion.words	Return words list.
2	emotion.sentences	Return sentences list.
3	emotion.affect_list	Return affect list.
4	emotion.affect_dict	Return affect dictionary.
5	emotion.raw_emotion_score	Return raw emotional counts.
6	emotion.top_emotions	Return highest emotions.
7	emotion.affect_frequencies	Return affect frequencies

12. Afinn

Afinn is the simplest yet popular lexicons used for sentiment analysis developed by *Finn Årup Nielsen*. It contains 3300+ words with a polarity score associated with each word. In python, there is an in-built function for this lexicon.

After a text is obtained, we start with text normalization.

Text normalization includes:

- converting all letters to lower or upper case
- converting numbers into words or removing numbers
- removing punctuations, accent marks and other diacritics
- removing white spaces
- expanding abbreviations
- removing stop words, sparse terms, and particular words
- text canonicalization