Data Science using Python

Dr. Kamini Solanki, (Associate Professor)
Parul Institute of Computer Application - BCA

Parul®
University

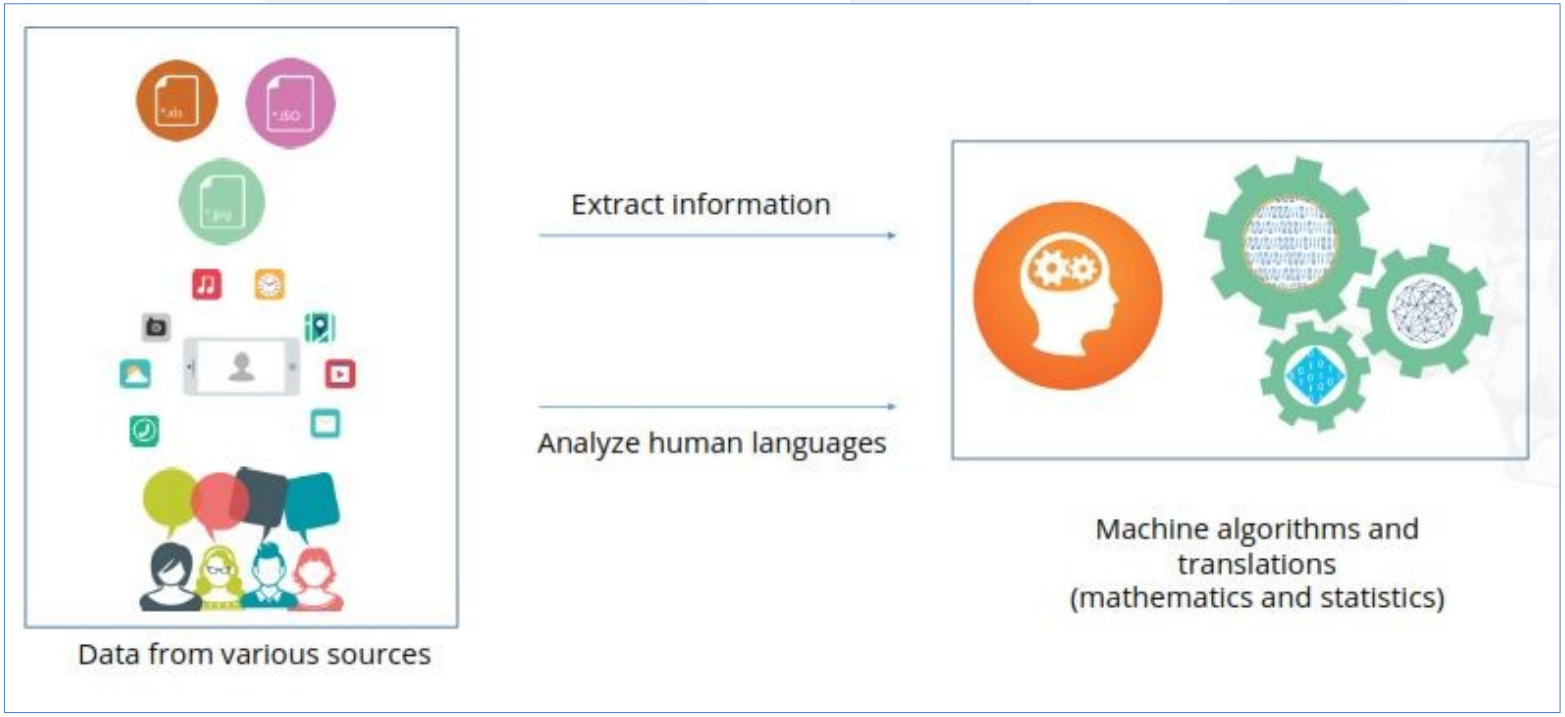
DIGITAL LEARNING CONTENT

CHAPTER - 7

Natural Language Processing (NLP) with SciKit Learn

Introduction to Natural Language Processing

- Natural language processing is an automated way to understand and analyze natural human languages and extract information from such data by applying machine algorithms.



Natural Language Processing

- It is also referred to as, the field of computer science or AI to extract the linguistics information from the underlying data.

Parul® University

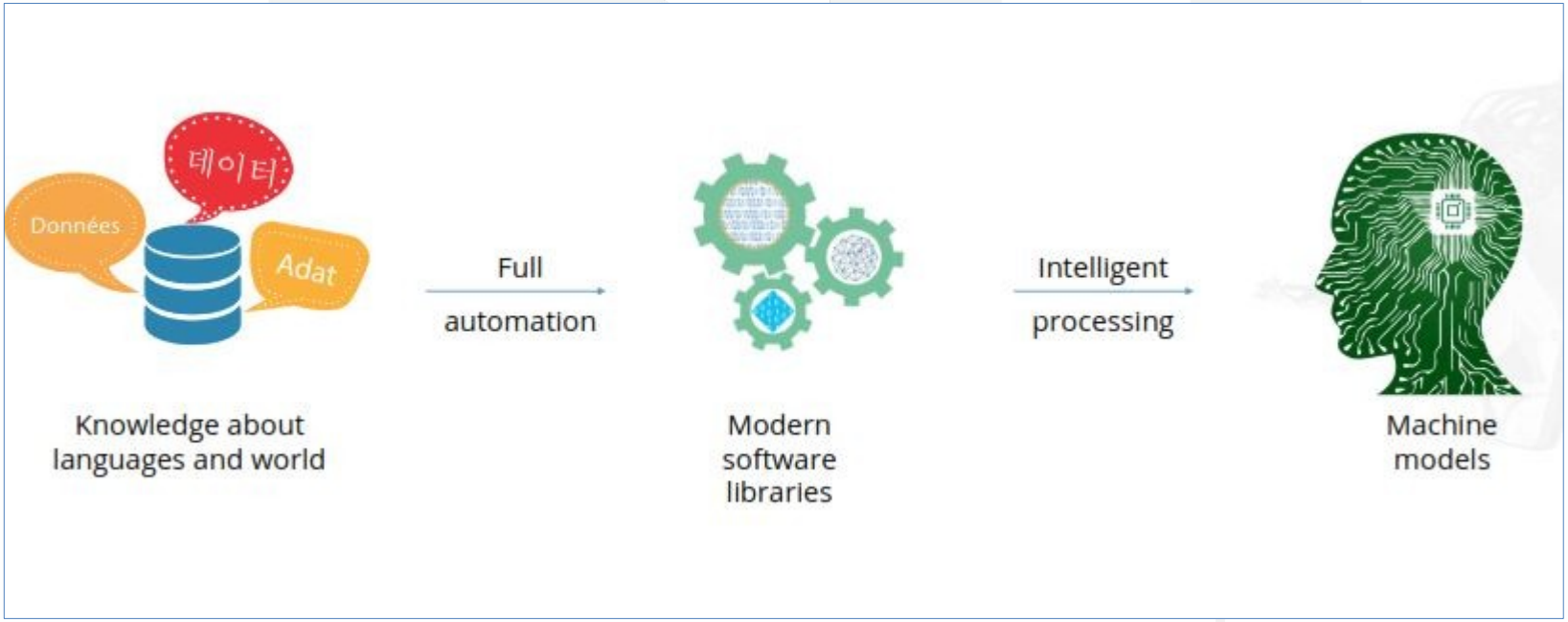
DIGITAL LEARNING CONTENT

Why Natural Language Processing

- The world is now connected globally due to the advancement of technology and devices.
- Analyzing tons of data
- Identifying various languages
- Applying quantitative analysis
- Handling ambiguities

Why Natural Language Processing

- NLP can achieve full automation by using modern software libraries, modules, and packages.



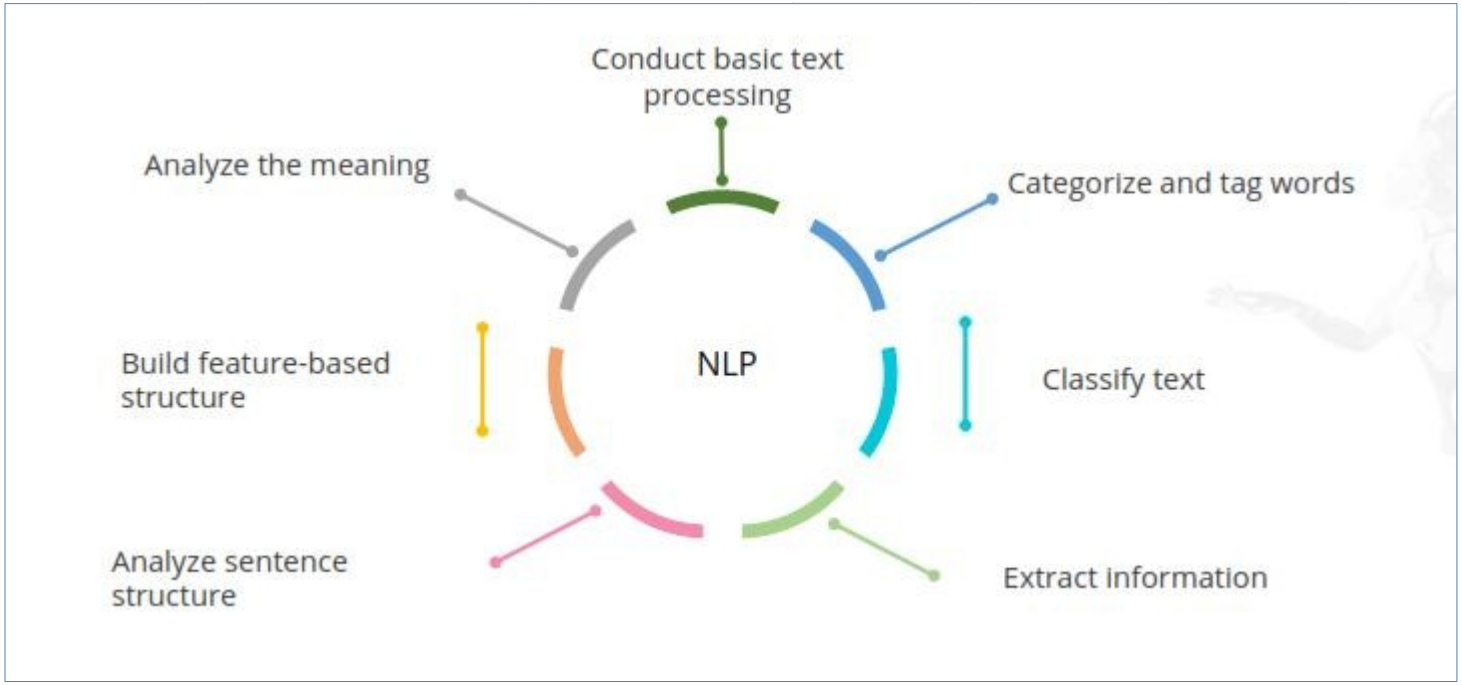
Parul® University

NLP Terminology

- Word boundaries :- Determines where one word ends and the other begins
- Tokenization :- Splits words, phrases, and idioms
- Stemming :- Maps to the valid root word
- Tf-idf :- Represents term frequency and inverse document frequency
- Semantic analytics :- Compares words, phrases, and idioms in a set of documents to extract meaning
- Disambiguation :- Determines meaning and sense of words (context vs. intent)
- Topic models:- Discovers topics in a collection of documents

NLP Approach for Text Data

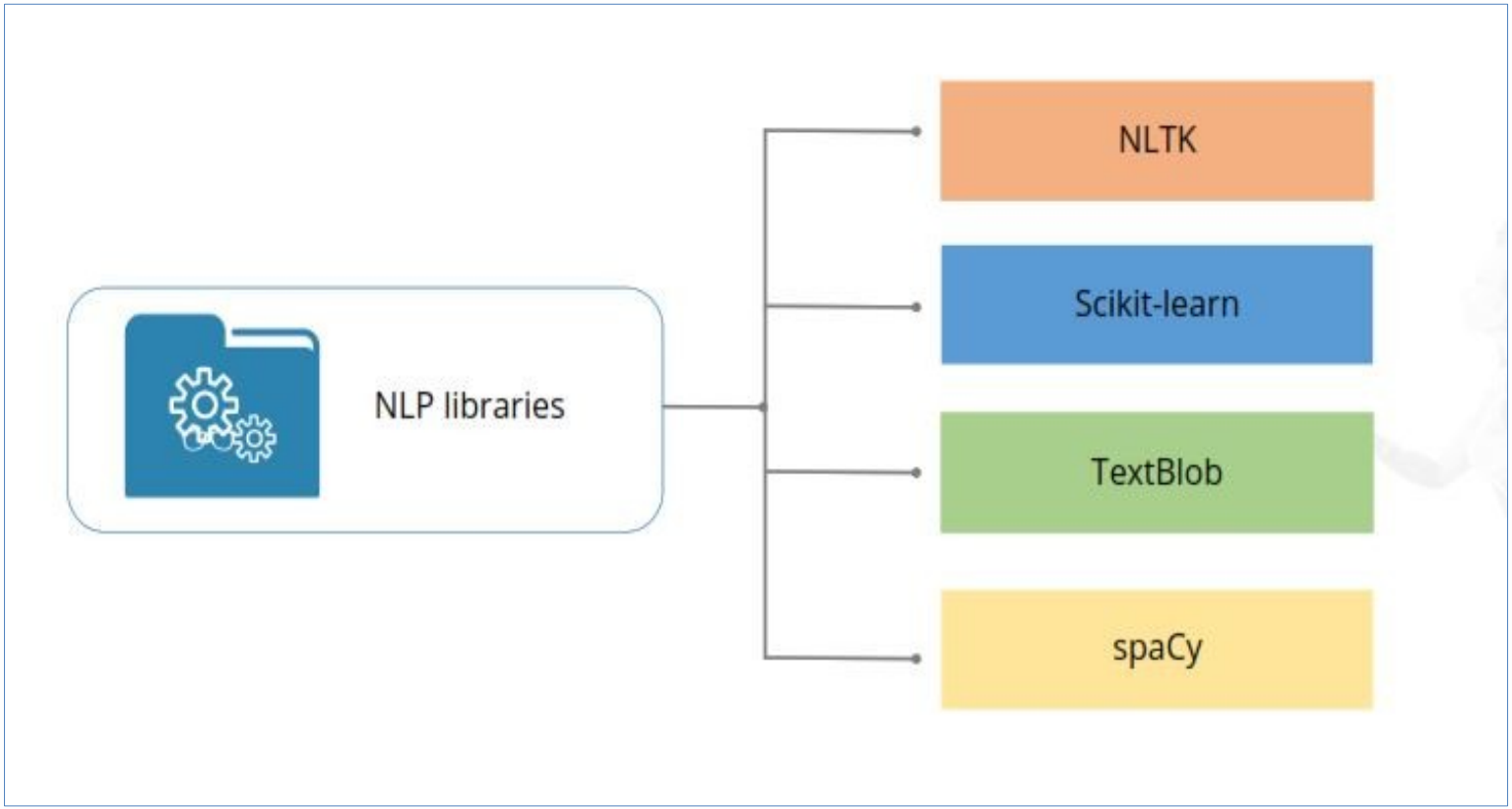
- Let us look at the Natural Language Processing approaches to analyze text data.



Applications of NLP

- Machine Translation :- Machine translation is used to translate one language into another. Google Translate is an example. It uses NLP to translate the input data from one language to another.
- Speech Recognition :- The speech recognition application understands human speech and uses it as input information. It is useful for applications like Siri, Google Now, and Microsoft Cortana.
- Sentiment Analysis :- Sentiment analysis is achieved by processing tons of data received from different interfaces and sources. For example, NLP uses all social media activities to find out the popular topic of discussion or importance.

Major NLP Libraries



Parul® University

DIGITAL LEARNING CONTENT

The Scikit-Learn Approach

- It is a very powerful library with a set of modules to process and analyze natural language data, such as text and images, and extract information using machine learning algorithms.
- Built-in module :- Contains built-in modules to load the dataset's content and categories.
- Feature extraction :- A way to extract information from data which can be text or images.
- Model training :- Analyzes the content based on particular categories and then trains them according to a specific model.

The Scikit-Learn Approach

Pipeline building mechanism

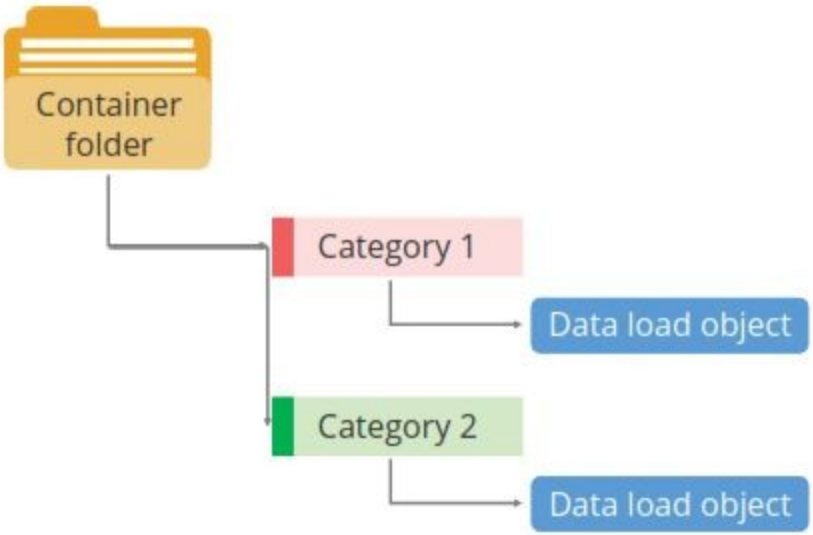
A technique to streamline the NLP process into stages.

Various stages of pipeline learning

1. Vectorization
2. Transformation
3. Model training and application

Modules to Load Content and Category

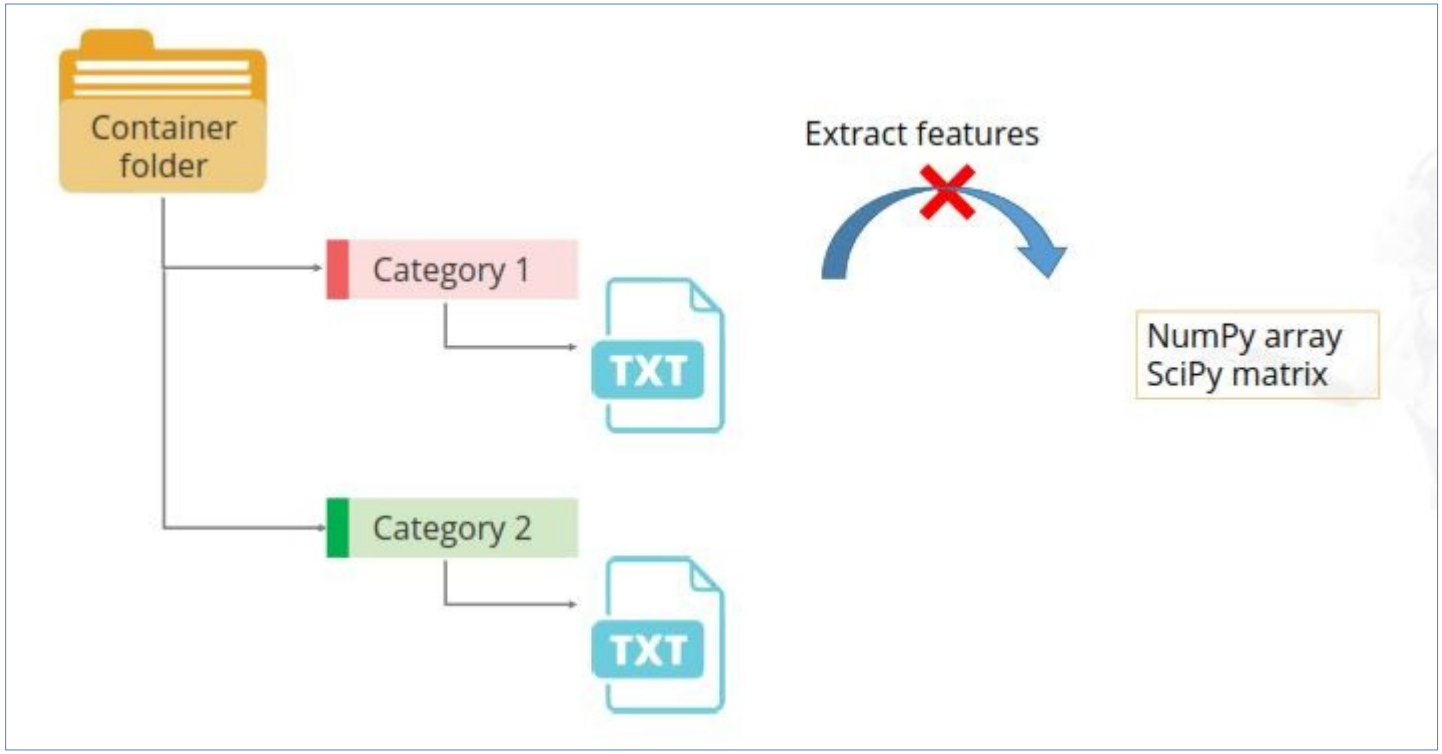
- Scikit-learn has many built-in datasets. There are several methods to load these datasets with the help of a data load object.




```
In [ ]: #Load dataset
        load_data = sklearn.datasets.load_files()
```

Modules to Load Content and Category

- The text files are loaded with categories as subfolder names.



Parul®
University

DIGITAL LEARNING CONTENT

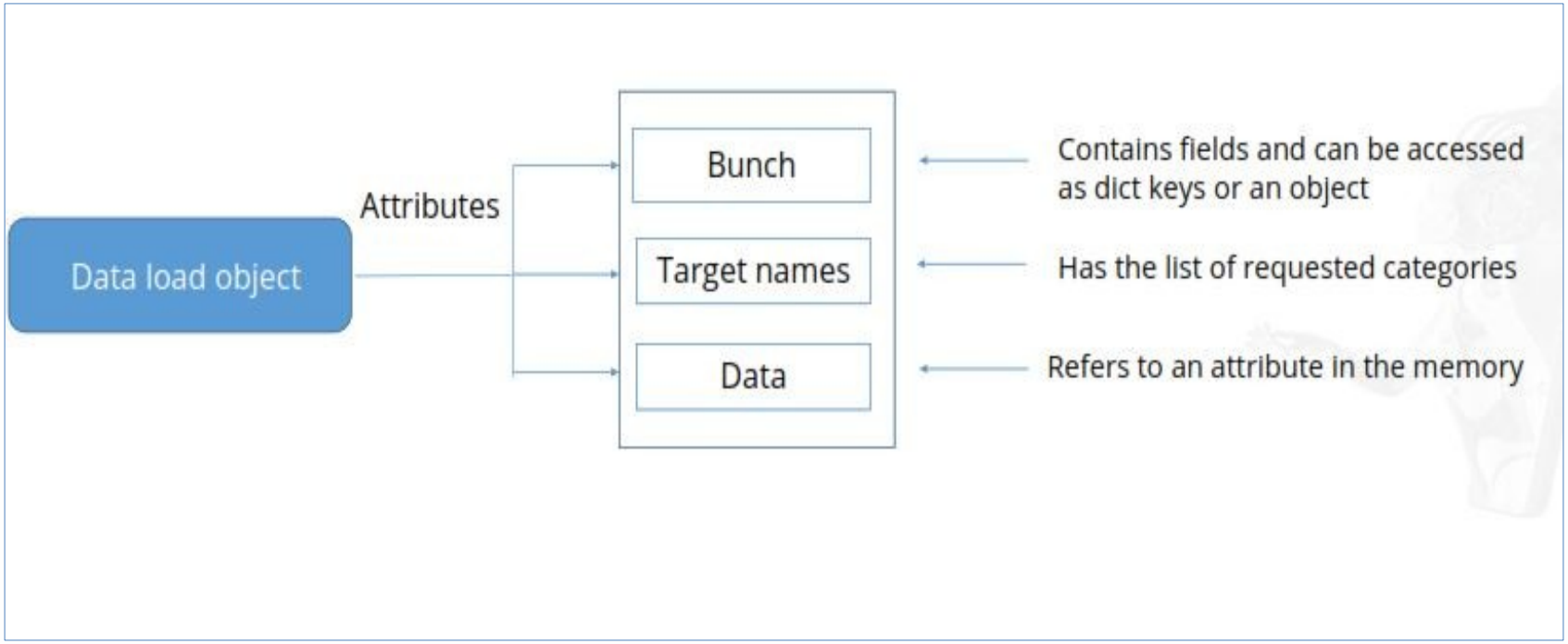
Modules to Load Content and Category

In [ ]:

```
#Build a feature extraction transformer
From sklearn.feature_extraction.text import <appropriate transformer>
```

Modules to Load Content and Category

- The attributes of a data load object are:



Modules to Load Content and Category

- The example shows how a dataset can be loaded using Scikit-learn:

```
In [1]: #Load dataset
        from sklearn.datasets import load_digits
```
← Import the dataset

```
In [2]: #create object of the loaded dataset
        digit_dataset = load_digits()
```
← Load dataset

```
In [3]: # use built in descr function to describe dataset
        digit_dataset.DESCR
```
← Describe the dataset

```
Out[3]: "Optical Recognition of Handwritten Digits Data Set\n==========
        ==\n\nNotes\n-----\nData Set Characteristics:\n    :Number of Instances: 5620\n    :Number of Attribut
        es: 64\n    :Attribute Information: 8x8 image of integer pixels in the range 0..16.\n    :Missing Attr
        ibute Values: None\n    :Creator: E. Alpaydin (alpaydin '@' boun.edu.tr)\n    :Date: July; 1998\n\nThi
        s is a copy of the test set of the UCI ML hand-written digits datasets\nhttp://archive.ics.uci.edu/ml/
        datasets/Optical+Recognition+of+Handwritten+Digits\n\nThe data set contains images of hand-written dig
        its: 10 classes where\neach class refers to a digit.\n\nPreprocessing programs made available by NIST
        were used to extract\nnormalized bitmaps of handwritten digits from a preprinted form. From a\ntotal o
        f 43 people, 30 contributed to the training set and different 13\nto the test set. 32x32 bitmaps are d
        ivided into nonoverlapping blocks of\n4x4 and the number of on pixels are counted in each block. This
        generates\nan input matrix of 8x8 where each element is an integer in the range\n0..16. This reduces d
```

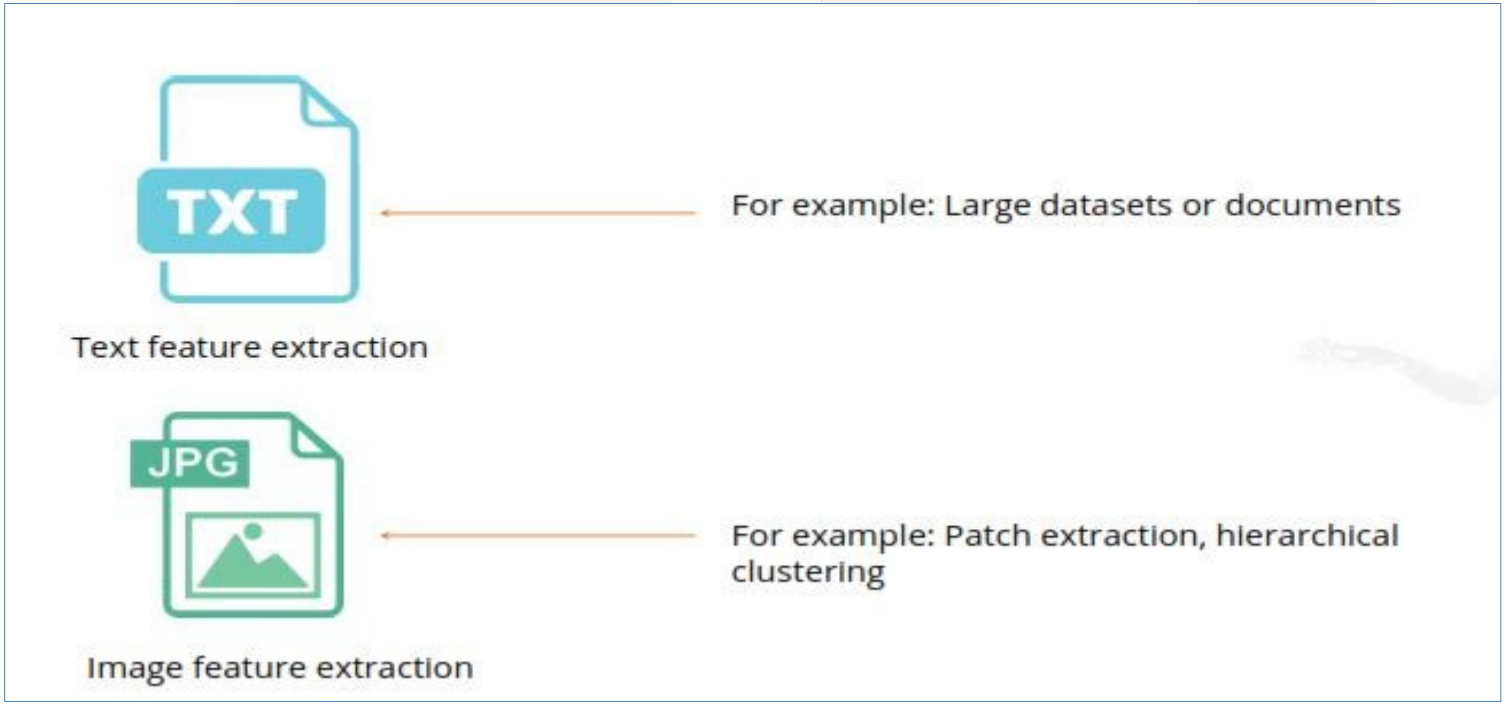
Modules to Load Content and Category

- Let us see how functions like type, .data, and .target help in analyzing a dataset.

```
In [4]: #view type of dataset                          <-- View type of dataset
        type(digit_dataset)

Out[4]: sklearn.datasets.base.Bunch

In [5]: #view data                                     <-- View data
        digit_dataset.data

Out[5]: array([[  0.,   0.,   5., ...,   0.,   0.,   0.],
               [  0.,   0.,   0., ...,  10.,   0.,   0.],
               [  0.,   0.,   0., ...,  16.,   9.,   0.],
               ...,
               [  0.,   0.,   1., ...,   6.,   0.,   0.],
               [  0.,   0.,   2., ...,  12.,   0.,   0.],
               [  0.,   0.,  10., ...,  12.,   1.,   0.]])

In [6]: #view target                                   <-- View target
        digit_dataset.target

Out[6]: array([0, 1, 2, ..., 8, 9, 8])
```

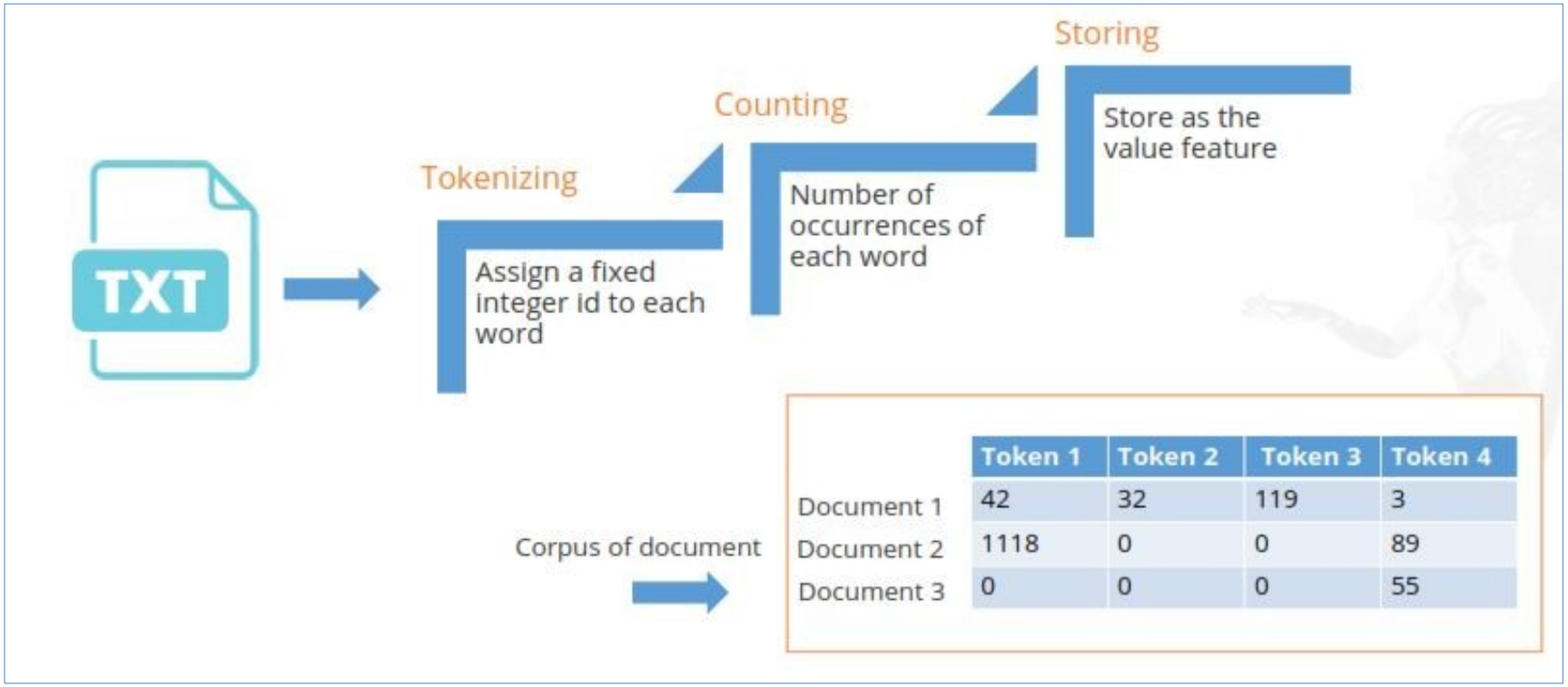
Feature Extraction

- Feature extraction is a technique to convert the content into the numerical vectors to perform machine learning.



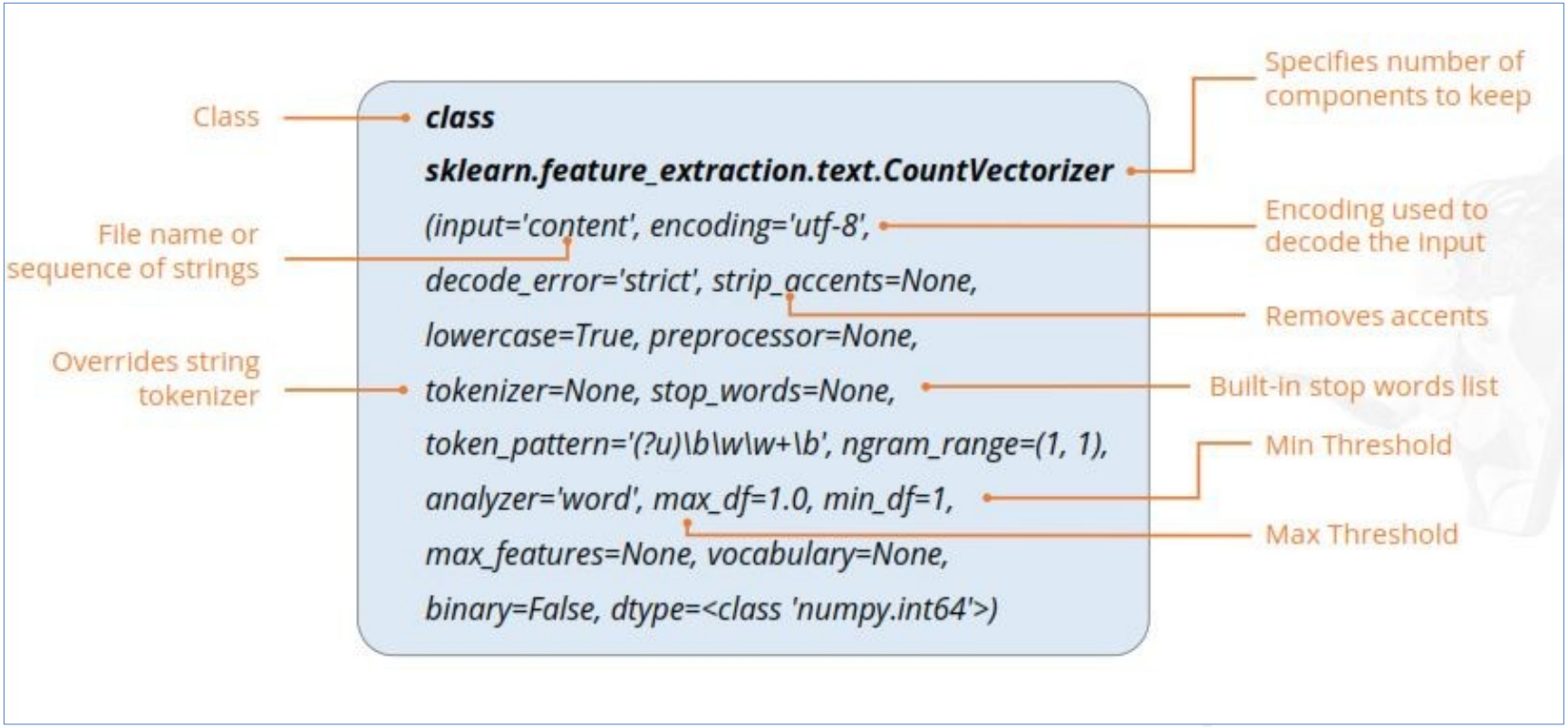
Bag of Word

- Bag of words is used to convert text data into numerical feature vectors with a fixed size.



	Token 1	Token 2	Token 3	Token 4
Document 1	42	32	119	3
Document 2	1118	0	0	89
Document 3	0	0	0	55

CountVectorizer Class Signature



Text Feature Extraction Considerations

- Sparse:- This utility deals with sparse matrix while storing them in memory. Sparse data is commonly noticed when it comes to extracting feature values, especially for large document datasets.
- Vectorizer:- It implements tokenization and occurrence. Words with minimum two letters get tokenized. We can use the analyzer function to vectorize the text data.
- Tf-idf :- It is a term weighing utility for term frequency and inverse document frequency. Term frequency indicates the frequency of a particular term in the document. Inverse document frequency is a factor which diminishes the weight of terms that occur frequently.
- Decoding:- This utility can decode text files if their encoding is specified.

Model Training

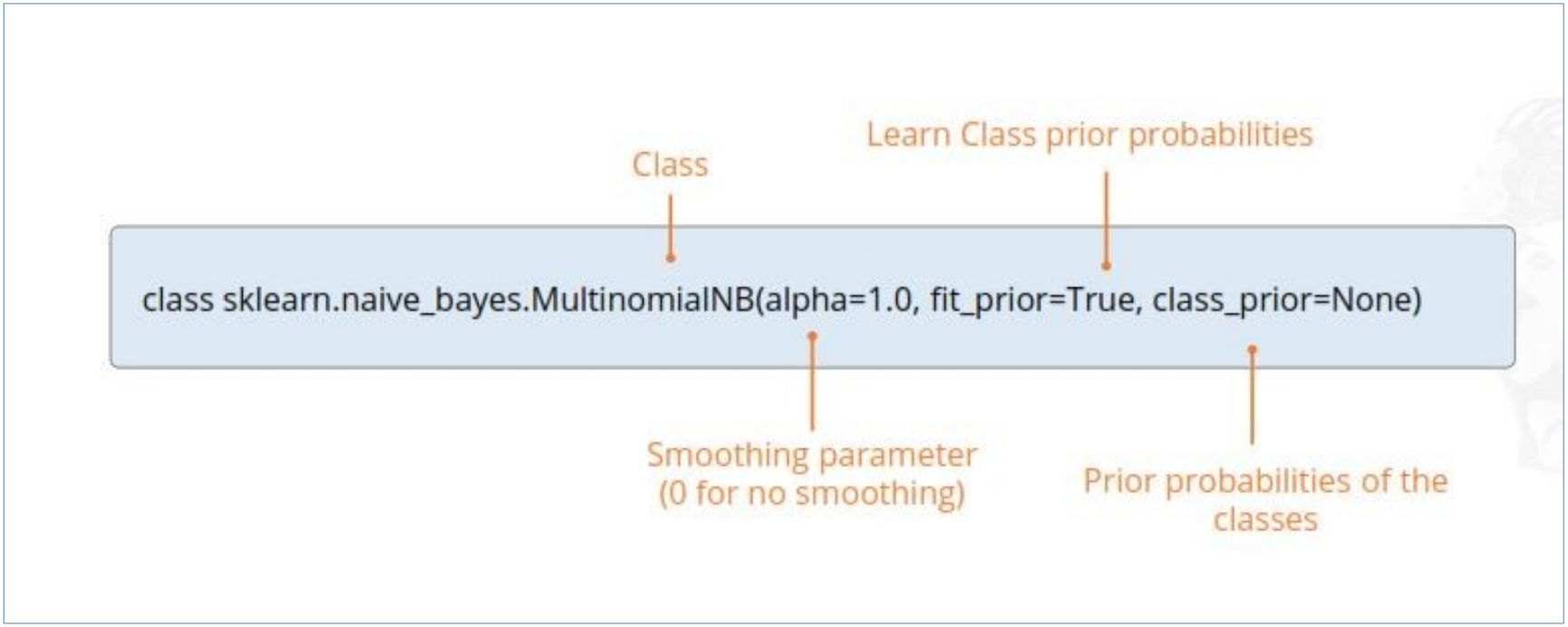
- An important task in model training is to identify the right model for the given dataset. The choice of model completely depends on the type of dataset.
- Supervised :- Models predict the outcome of new observations and datasets, and classify documents based on the features and response of a given dataset. Example: Naïve Bayes, SVM, linear regression, K-NN neighbors
- Unsupervised:- Models identify patterns in the data and extract its structure. They are also used to group documents using clustering algorithms. Example: K-means

Naïve Bayes Classifier

- It is the most basic technique for classification of text.
- **Advantages:**
- It is efficient as it uses limited CPU and memory.
- It is fast as the model training takes less time.
- **Uses:**
- Naïve Bayes is used for sentiment analysis , email spam detection, categorization of documents, and language detection.
- Multinomial Naïve Bayes is used when multiple occurrences of the words matter.

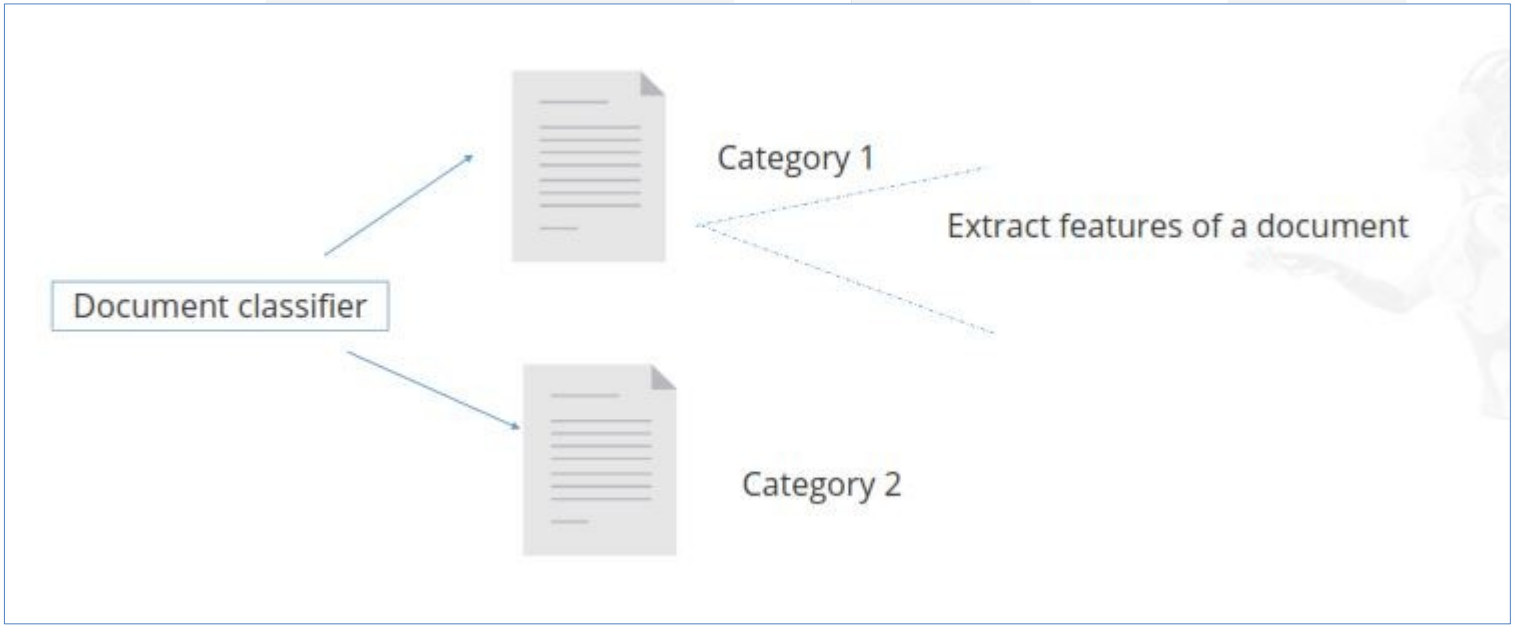
Naïve Bayes Classifier

- Let us take a look at the signature of the multinomial Naïve Bayes classifier:

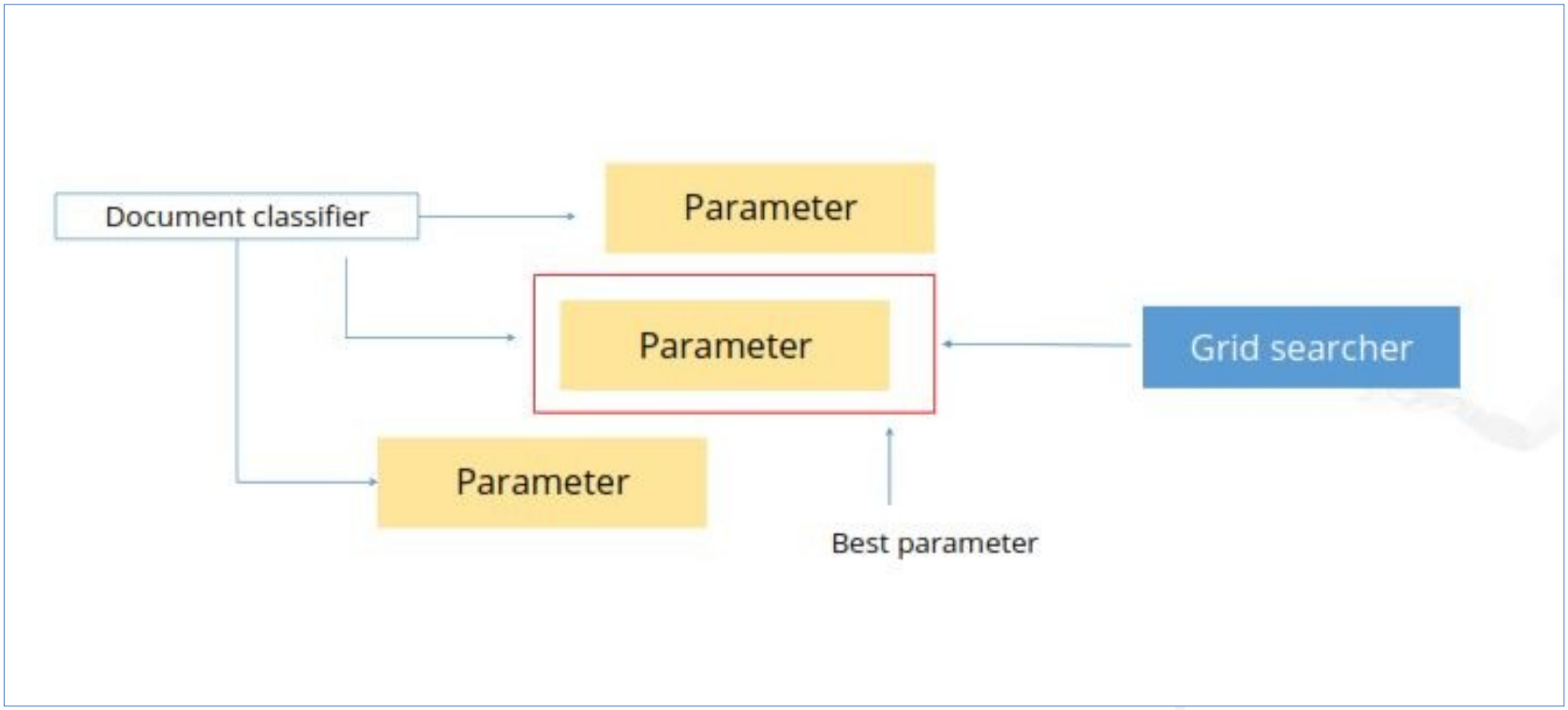


Grid Search and Multiple Parameters

- Document classifiers can have many parameters. A Grid approach helps to search the best parameters for model training and predicting the outcome accurately.

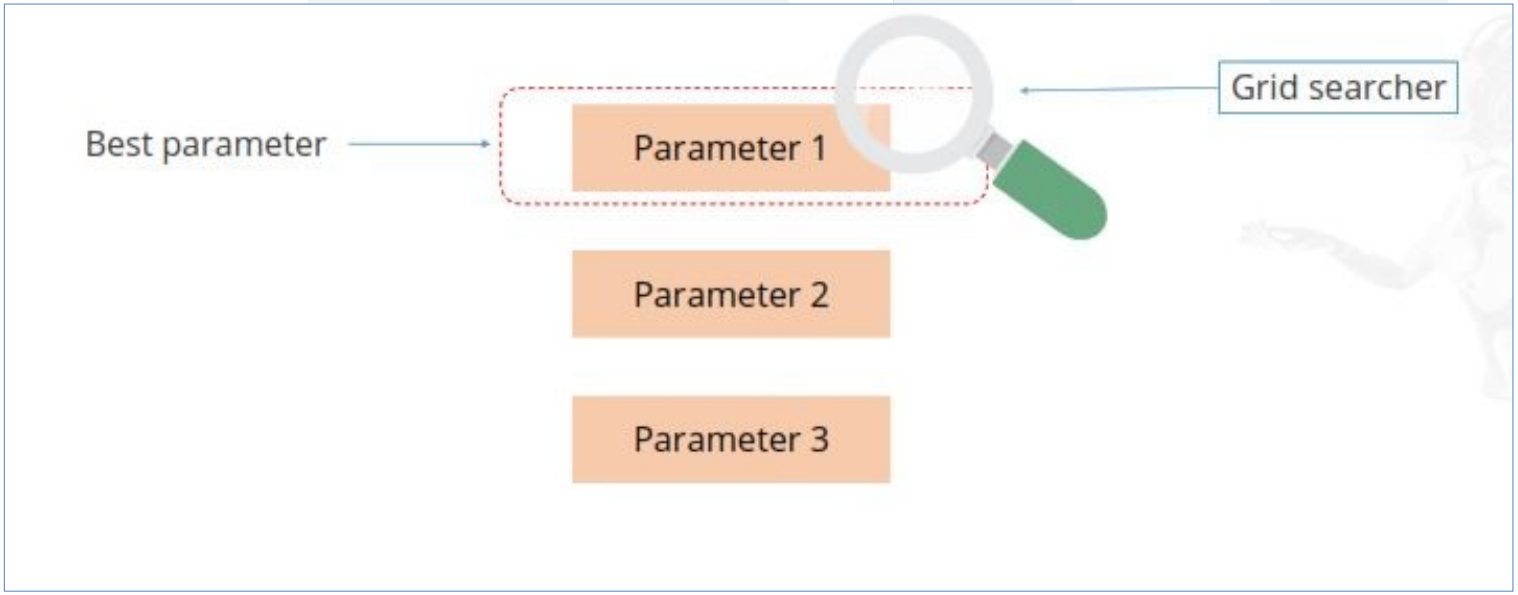


Grid Search and Multiple Parameters

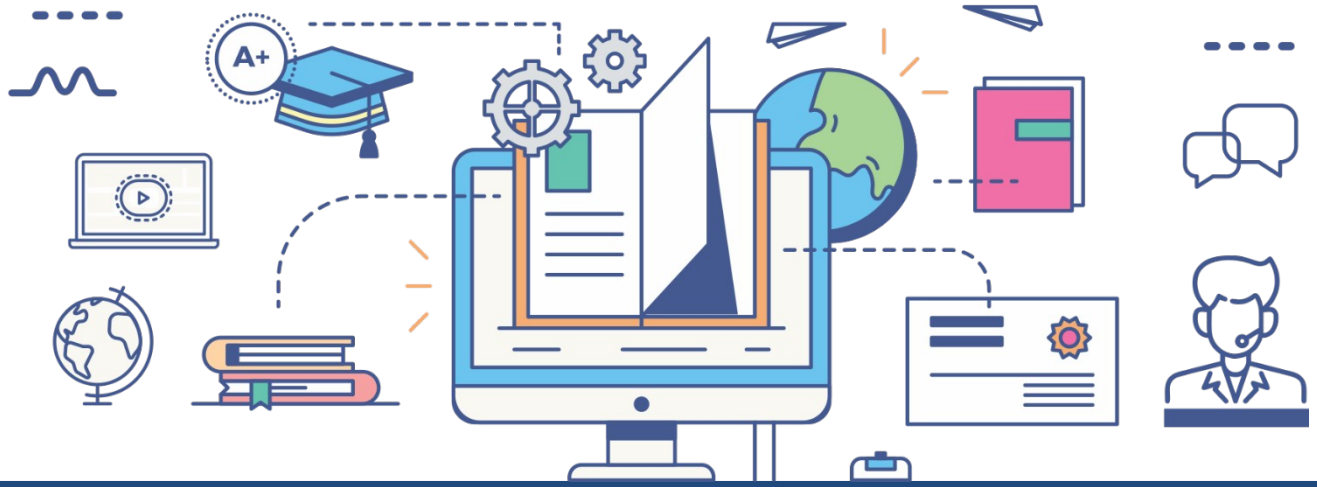


Grid Search and Multiple Parameters

- In grid search mechanism, the whole dataset can be divided into multiple grids and a search can be run on the entire grid or a combination of grids.



DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in