The Dataset Name: BBC articles full text and category

Dataset Details:

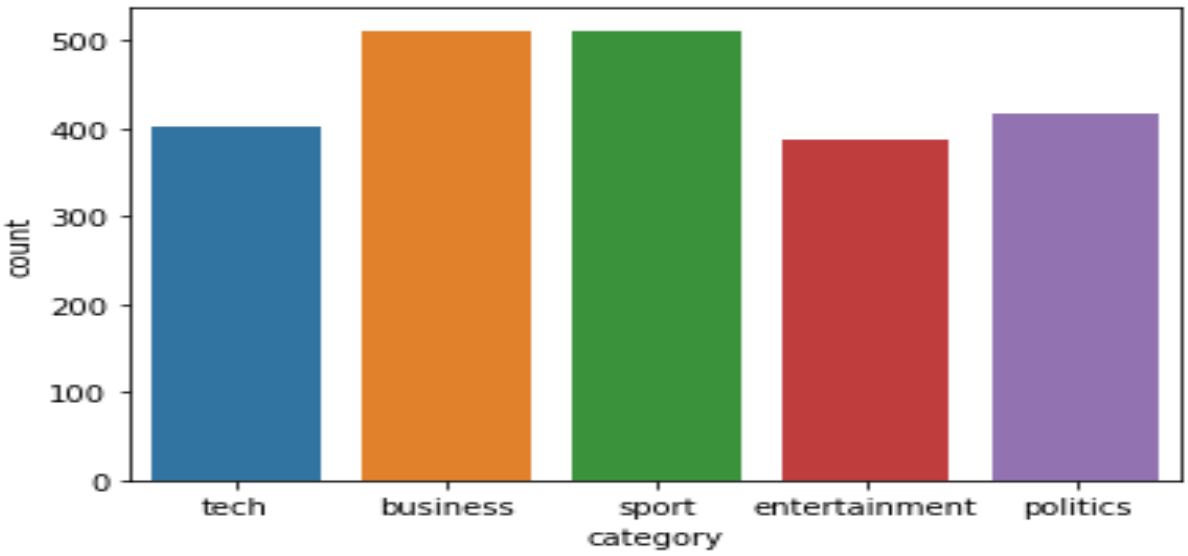
This dataset is contain articles from BBC news website

It has 5 category sport, business, entertainment, politics, tech

It come in csv file that has 2 column the category column and the text or article column each row has an article and type.

It has 510 articles in business, 386 in entertainment, 417 politics, 511 sport, 401 tech.

Overall there is 2126 articles.



Link to the dataset on Kaggle:

https://bit.ly/3Ftwon7

Project and docs Link on Google Drive:

https://bit.ly/3EItvxS

Project idea in Details:

Our task is to use the above dataset to train a machine learning model that well predict the Category of the assign article based on the information that the model obtained during the training phase.

In our task specifically we will predict the article category from given article itself we will used the supervised strategy to deal with it we will train our model on 2126 articles that labeled manually using support vector machine.

But we cannot feed our raw text to machine to deal with

We just go throw the normal steps in machine learning model

First, we bring the dataset that we chose BBC article dataset

Then we just check is there an invalid value or there is null value because it will affect our model then we need to convert

Our category column to numerical we use here label Encoding

From sklearn then we now just need to cleaning and preprocessing the text then we convert it to numerical using TfidfVectorizer then we train our support vector classifier

Then we are ready to predict any article 🙂.

The apps in the market that similar to our project:

First is the made or provide by parascript

You can find it here https://bit.ly/3Jg2khb

The second provide by adlib

You can find it here https://bit.ly/3sAP8h0

There is a lot of paper in this task here is a few:

1-One-Class SVMs for Document Classification

https://bit.ly/3H8Z5WY

2- SVM multi-classifier and Web document classification ieee

https://bit.ly/33NOulS

3- Text document pre-processing with the KNN for classification using the SVM ieee

https://bit.ly/3szGtLN

4-An Optimal SVM-Based Text Classification Algorithm

https://bit.ly/3ev6K5v

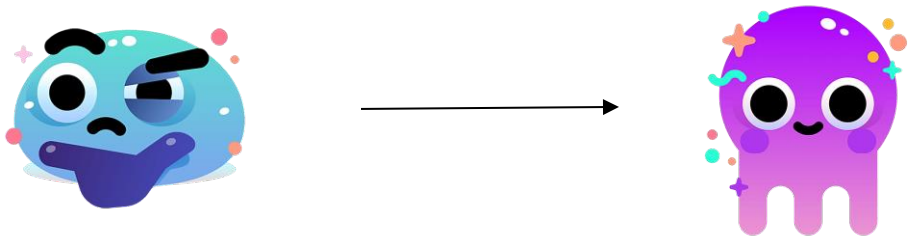
5-Is Naïve Bayes a Good Classifier for Document Classification?

https://bit.ly/3psoFQF

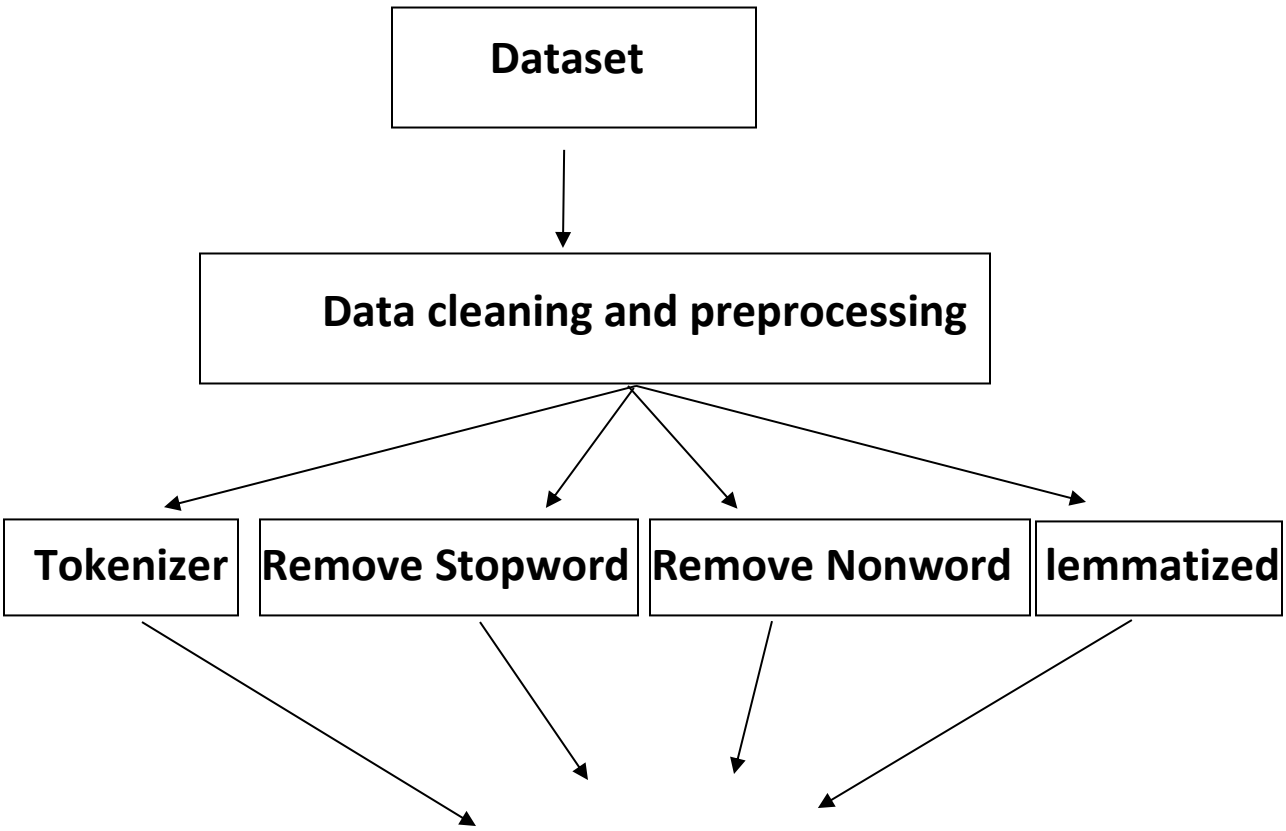
Main functionalities in our project from user respective that he will give use unknown article and we will give him the subject category.

Answer the user question about category that

what is our model do 😉.



Our algorithm or approach:



.

Transform the list to string Then do The Vectorization To convert string to number

↓

Split the dataset to training and testing

↓

Using the Support vector machine To train the model

The Details algorithm and library used:

1- We gather The Dataset

- **This is most important element you'll need for training your Machine Learning model. The dataset needs to contain enough documents or examples for each category so that the algorithm can learn how to differentiate between them. The Number and the quality of data in The Dataset is critical when training a classifier with machine learning.**

2- We Store data in data frame from pandas library

- **Now it's the time to load the data from csv file to python But how can we do that? Dataframe it's right We use the pandas library to Load csv file and store it In Dataframe to use it in python.**

3- We clean data using nltk library

Here We do some Natural Language Processing Using the python nltk library . We need to clean Our data before feed it to the model To ensure that It will achieve the Maximize result. To Tokenize or Convert our string to list to make the edit easy We use nltk.word_tokenize model . There is a lot Of useless word that we use in every sentence but It not effect the main topic like (The , a , an , of , in , ….) We need to remove them so we use

nltk.corpus.stopwords.words

***We need To remove anything other than char we use regx to achieve that we use regular expression it will match The string with some pattern.**

4- we use label encoder to encode the category column

- **Because Machine work only with number and most Machine leaning algorithm we need to convert the String into number so in the category column we Use label encoder.**

4-We transform the cleaned text using TfidfVectorizer

- **We need to convert the article to number so We use the TfidfVectorizer that is stand for Term frequency inverse document frequency It will just Give a number to each word in docs Based on the Number of time that appear in And in the same time it not frequency that match In the whole articles.**

5-Then We split the dataset into training and testing

Before we train our model we need to split our dataset

We do that to test our model after train it and come with

The accuracy score that determine how is our model

Good at predict new article we split our data into 20% for

Testing and the rest for training.

5-Then we train our model using support vector classifier (svc)

- **We use support vector machine because It's one of the simplest algorithm that will give Us a maximize answer on the topic of document classification. The support vector machine work only By draw a line between data but that line will has A maximize margin from the vector nodes.**

6-Then we evaluate the model using the accuracy score

- **We compare the predict result of the x_test With the y_test and come with the ratio Of accuracy**
- **We use confusion matrix to Visualize The Number Of false positive and false negative.**

Analysis:

We have achieved 0.98 accuracy score.

Disadvantage: The TfidfVectorizer has one limitation is that the vocabulary can become very large. This, in turn, will require large vectors for encoding documents and impose large requirements on memory and slow down algorithms

But we can solve this using Hashing with HashingVectorizer