SUBJECT: Machine learning and Data Science	
NAME : YASH DATTATRAYA DESAI	CLASS : DIV 3
SEMESTER/YEAR : 7TH (BE)	ROLL NO : C43414

EXPERIMENT NO- 06

TITLE: Text Classification

AIM: Write a program to recognize a document is positive or negative based on polarity words using suitable classification method.

SOFTWARES:

SOFTWARE	VERSION
Jupyter Notebook	V5.1

THEORY:

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.

Text Classification (TC) is the process of assigning tags or categories to text according to its content. It is one of the fundamental tasks in Natural Language Processing (NLP). Text classifiers can be used to organize, structure, and categorize pretty much anything.

There are many approaches to text classification such as:

- Rule-based systems
- Machine learning-based systems
- Hybrid systems

Text classification is mostly used for sentiment analysis, topic labelling, spam detection, and intent detection. Here are some applications that text classification is used for information retrieval.

1. Detecting a document's encoding (ASCII, Unicode UTF-8, etc).
2. Word segmentation.
3. True casing.
4. Identifying the language of a document.
5. The automatic detection of spam pages.
6. The automatic detection of sexually explicit content.
7. Sentiment analysis.

8. Personal email sorting.
9. Topic-specific or vertical search.

Text classification algorithms are at the heart of a variety of software systems that process text data at scale. There are many reasons why everyone is obsessed with using text classification problems.

- Scalability — It takes a lot of time for a human to manually analyze and organize text. Machine learning helps you to do it fast and in an accurate way.
- Real-time analysis — Text classification is used in some companies for critical problems such as sentiment analysis. TC can make accurate precisions and help you to make decisions right away.
- Consistent criteria — This helps by allowing humans to reduce errors with centralized TC problems.

APPROACH:

Step 01: Create a python file and import the following packages.

Step 02: Define a function to extract features.

Step 03: To get the training data, use the following movie reviews from NLTK.

Step 04: Now we will separate the positive and negative reviews.

Step 05: Since we need 2 datasets for this process, divide the data into training and testing datasets.

Step 06: Extract the features.

Step 07: Use the Navie Bayes classifier. Define the object and train it.

Step 08: To find out the most informative words inside the classifier which decides a review is positive or negative, print the following.

Step 09: Create some random movie reviews of your own.

Step 10: Now, run the classifier on those sentences and obtain the predictions.

Step 11: Print the output.

CONCLUSION:

We performed the text classification and recognized the output accordingly.