**DATE: 17-08-2023**

# Natural Language Processing (NLP)

## K RAMAKRISHNAN COLLEGE OF ENGINEERING

SUBMITTED BY,

NAME: POORANI M B

EMAIL:poorani2243@gmail.com

DEPARTMENT: IT

REG NO: 8115U21IT021

## PROBLEM STATEMENT:

## Assignment Topic:

Natural Language Processing

## Assignment Description:

Build a sentiment analysis classifier using natural language processing techniques. Use a labeled dataset of customer reviews, where each review is labeled as positive or negative. Implement and evaluate the classifier's performance, and provide insights into its potential real-world applications.

## SOLUTION :

steps involved in building a sentiment analysis classifier using natural language processing techniques:

**Collect a labeled dataset of customer reviews:**

This dataset should contain a set of customer reviews, each of which is labeled as positive or negative.

The size of the dataset will depend on the complexity of the task and the desired accuracy of the classifier.

**Preprocess the data**: This involves cleaning the data and removing any noise or irrelevant information.

This may include removing stop words, stemming words, and normalizing the text.

**Choose a machine learning algorithm**: There are many different machine learning algorithms that can be used for sentiment analysis.

Some popular algorithms include Naive Bayes, Support Vector Machines, and Random Forest.

**Train the classifier**: This involves feeding the labeled dataset to the machine learning algorithm.

The algorithm will learn to associate certain words or phrases with positive or negative sentiment.

**Evaluate the classifier's performance**: This can be done by testing the classifier on a held-out set of data that was not used for training.

The performance of the classifier can be measured using metrics such as accuracy, precision, and recall.

**Deploy the classifier**: Once the classifier is trained and evaluated, it can be deployed to production.

This means that it can be used to classify new customer reviews in real time.

Here are some potential real-world applications of sentiment analysis:

**Customer feedback analysis**: Sentiment analysis can be used to analyze customer feedback to identify areas where products or services can be improved.

**Social media monitoring**: Sentiment analysis can be used to monitor social media for mentions of a brand or product.

This can be used to identify positive and negative sentiment, as well as trends in sentiment over time.

**Market research** :Sentiment analysis can be used to gather insights into customer preferences and opinions.

This information can be used to improve products and services, as well as develop new marketing campaigns.

# What is Sentiment Analysis?

Sentiment Analysis, as the name suggests, it means to identify the view or emotion behind a situation. It basically means to analyze and find the emotion or intent behind a piece of text or speech or any mode of communication.

In this article, we will focus on the sentiment analysis of text data.

We, humans, communicate with each other in a variety of languages, and any language is just a mediator or a way in which we try to express ourselves. And, whatever we say has a sentiment associated with it. It might be positive or negative or it might be neutral as well.

Suppose, there is a fast-food chain company and they sell a variety of different food items like burgers, pizza, sandwiches, milkshakes, etc. They have created a website to sell their food and now the customers can order any food item from their website and they can provide reviews as well, like whether they liked the food or hated it.

- User Review 1: I love this cheese sandwich, it's so delicious.
- User Review 2: This chicken burger has a very bad taste.
- User Review 3: I ordered this pizza today.

So, as we can see that out of these above 3 reviews,

The first review is definitely a **positive** one and it signifies that the customer was really happy with the sandwich.

The second review is **negative**, and hence the company needs to look into their burger department.

And, the third one doesn't signify whether that customer is happy or not, and hence we can consider this as a **neutral** statement.

By looking at the above reviews, the company can now conclude, that it needs to focus more on the production and promotion of their sandwiches as well as improve the quality of their burgers if they want to increase their overall sales.

But, now a problem arises, that there will be hundreds and thousands of user reviews for their products and after a point of time it will become nearly impossible to scan through each user review and come to a conclusion.

Neither can they just come up with a conclusion by taking just 100 reviews or so, because maybe the first 100-200 customers were having similar taste and liked the sandwiches, but over time when the no. of reviews increases, there might be a situation where the positive reviews are overtaken by more no. of negative reviews.

Therefore, this is where the Sentiment Analysis Model comes into play, which takes in a huge corpus of data having user reviews and finds a pattern and comes up with a conclusion based on real evidence rather than assumptions made on a small sample of data.

(We will explore the working of a basic Sentiment Analysis model later in this article.)

We can even break these principal sentiments(positive and negative) into smaller sub sentiments such as "Happy", "Love", "Surprise", "Sad", "Fear", "Angry" etc. as per the needs or business requirement.

Real-World Example –

1. There was a time when the social media services like Facebook used to just have two emotions associated with each post, i.e You can like a post or you can leave the post without any reaction and that basically signifies that you didn't like it.
2. But, over time these reactions to post have changed and grew into more granular sentiments which we see as of now, such as "like", "love", "sad", "angry" etc.

And, because of this upgrade, when any company promotes their products on Facebook, they receive more specific reviews which will help them to enhance the customer experience.

And because of that, they now have more granular control on how to handle their consumers, i.e. they can target the customers who are just "sad" in a different way as compared to customers who are "angry", and come up with a business plan accordingly because nowadays, just doing the **bare minimum is not enough**.

This is why we need a process that makes the computers understand the Natural Language as we humans do, and this is what we call Natural Language Processing(NLP). And, as we know Sentiment Analysis is a sub-field of NLP and with the help of machine learning techniques, it tries to identify and extract the insights.

## Step by Step procedure to Implement Sentiment Analysis

First, let's import all the python libraries that we will use throughout the program.

**Basic Python Libraries**

1. Pandas – library for data analysis and data manipulation
2. Matplotlib – library used for data visualization
3. Seaborn – a library based on matplotlib and it provides a high-level interface for data visualization
4. WordCloud – library to visualize text data
5. re – provides functions to pre-process the strings as per the given regular expression

## Import all the python libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import re
```

**This article was published as a part of the Data Science Blogathon**

# What is Sentiment Analysis?

Sentiment Analysis, as the name suggests, it means to identify the view or emotion behind a situation. It basically means to analyze and find the emotion or intent behind a piece of text or speech or any mode of communication.

In this article, we will focus on the sentiment analysis of text data.

We, humans, communicate with each other in a variety of languages, and any language is just a mediator or a way in which we try to express ourselves. And, whatever we say has a sentiment associated with it. It might be positive or negative or it might be neutral as well.

Suppose, there is a fast-food chain company and they sell a variety of different food items like burgers, pizza, sandwiches, milkshakes, etc. They have created a website to sell their food and now the customers can order any food item from their website and they can provide reviews as well, like whether they liked the food or hated it.

- User Review 1: I love this cheese sandwich, it's so delicious.
- User Review 2: This chicken burger has a very bad taste.
- User Review 3: I ordered this pizza today.

So, as we can see that out of these above 3 reviews,



The first review is definitely a **positive** one and it signifies that the customer was really happy with the sandwich.

The second review is **negative**, and hence the company needs to look into their burger department.

And, the third one doesn't signify whether that customer is happy or not, and hence we can consider this as a **neutral** statement.

By looking at the above reviews, the company can now conclude, that it needs to focus more on the production and promotion of their sandwiches as well as improve the quality of their burgers if they want to increase their overall sales.

But, now a problem arises, that there will be hundreds and thousands of user reviews for their products and after a point of time it will become nearly impossible to scan through each user review and come to a conclusion.

Neither can they just come up with a conclusion by taking just 100 reviews or so, because maybe the first 100-200 customers were having similar taste and liked the sandwiches, but over time when the no. of reviews increases, there might be a situation where the positive reviews are overtaken by more no. of negative reviews.

Therefore, this is where the Sentiment Analysis Model comes into play, which takes in a huge corpus of data having user reviews and finds a pattern and comes up with a conclusion based on real evidence rather than assumptions made on a small sample of data.

(We will explore the working of a basic Sentiment Analysis model later in this article.)

We can even break these principal sentiments(positive and negative) into smaller sub sentiments such as “Happy”, “Love”, ”Surprise”, “Sad”, “Fear”, “Angry” etc. as per the needs or business requirement.

Real-World Example –

1. There was a time when the social media services like Facebook used to just have two emotions associated with each post, i.e You can like a post or you can leave the post without any reaction and that basically signifies that you didn’t like it.
2. But, over time these reactions to post have changed and grew into more granular sentiments which we see as of now, such as “like”, “love”, “sad”, “angry” etc.

And, because of this upgrade, when any company promotes their products on Facebook, they receive more specific reviews which will help them to enhance the customer experience.

And because of that, they now have more granular control on how to handle their consumers, i.e. they can target the customers who are just “sad” in a different way as compared to customers who are “angry”, and come up with a business plan accordingly because nowadays, just doing the **bare minimum is not enough**.

Now, as we said we will be creating a Sentiment Analysis Model, but it's easier said than done.

As we humans communicate with each other in a way that we call Natural Language which is easy for us to interpret but it's much more complicated and messy if we really look into it.

Because, there are billions of people and they have their own style of communicating, i.e. a lot of tiny variations are added to the language and a lot of sentiments are attached to it which is easy for us to interpret but it becomes a challenge for the machines.

This is why we need a process that makes the computers understand the Natural Language as we humans do, and this is what we call Natural Language Processing(NLP). And, as we know Sentiment Analysis is a sub-field of NLP and with the help of machine learning techniques, it tries to identify and extract the insights.

Now, let's get our hands dirty by implementing Sentiment Analysis, which will predict the sentiment of a given statement.

## Step by Step procedure to Implement Sentiment Analysis

First, let's import all the python libraries that we will use throughout the program.

**Basic Python Libraries**

1. Pandas – library for data analysis and data manipulation
2. Matplotlib – library used for data visualization
3. Seaborn – a library based on matplotlib and it provides a high-level interface for data visualization

4. WordCloud – library to visualize text data

5. re – provides functions to pre-process the strings as per the given regular expression

```
import pandas as pd
import matplotlib.pyplot
as plt
import seaborn as sns
from wordcloud import
WordCloud
import re
```

# Natural Language Processing

1. nltk – Natural Language Toolkit is a collection of libraries for natural language processing

2. stopwords – a collection of words that don't provide any meaning to a sentence

3. WordNetLemmatizer – used to convert different forms of words into a single item but still keeping the context intact.

```
import nltk
from nltk.corpus import
stopwords
from nltk.stem import
WordNetLemmatizer
```

**Scikit-Learn (Machine Learning Library for Python)**

1. CountVectorizer – transform text to vectors

2. GridSearchCV – for hyperparameter tuning

3. RandomForestClassifier – machine learning algorithm for classification

```
from
sklearn.feature_extraction.text
import CountVectorizer
from sklearn.model_selection
import GridSearchCV
from sklearn.ensemble
import RandomForestClassifier
```

**Evaluation Metrics**

1. Accuracy Score – no. of correctly classified instances/total no. of instances

2. Precision Score – the ratio of correctly predicted instances over total positive instances

3. Recall Score – the ratio of correctly predicted instances over total instances in that class

4. Roc Curve – a plot of true positive rate against false positive rate

5. Classification Report – report of precision, recall and f1 score

6. Confusion Matrix – a table used to describe the classification models

```
from sklearn.metrics import
accuracy_score,precision_score,recall_score,confusion_matr
ix,roc_curve,classification_report
from scikitplot.metrics import plot_confusion_matrix
```