**MALLA REDDY UNIVERSITY CODE HACKATHON**

**TECH FEST - NEUROSPLASH – 2023**

**BATCH 7 DOCUMENTATION REPORT**

**Team Member 1:** Athreya Sai Gopal Acharya Kidambi

**Problem Statement:** Sentiment Analysis of Customer Reviews

Sentiment analysis is a technique for determining the sentiment or emotion expressed in a  
piece of text, such as a customer review or social media post. This information can be useful for businesses to understand customer opinions and feedback about their products and services.

**Objective:** The goal of this project is to build a sentiment analysis model that can accurately classify customer reviews as positive, negative, or neutral. We have a dataset of customer reviews from a range of product categories, such as electronics, clothing, and food. The data includes both the text of the review and the sentiment label. We should explore different algorithms and techniques for sentiment analysis, such as logistic regression, support vector machines, and deep learning models. We should also experiment with different pre-processing steps, such as removing stop words and stemming, as well as feature engineering, such as word embedding’s and n-grams.

**Introduction:** Sentiment analysis is a powerful tool that allows computers to understand the underlying subjective tone of a piece of writing. This is something that humans have difficulty with, and as you might imagine, it isn’t always so easy for computers, either. But with the right tools and Python, you can use sentiment analysis to better understand the sentiment of a piece of writing

**Methodology:**

1. **Loading the Data:** Any sentiment analysis workflow begins with loading data. But what should be done once the data’s been loaded? We need to process it through a natural language processing pipeline before you can do anything interesting with it.

The necessary steps include the following:

1. Tokenizing sentences to break text down into sentences, words, or other units
2. Removing stop words like “if,” “but,” “or,” and so on
3. Normalizing words by condensing all forms of a word into a single form
4. Vectorizing text by turning the text into a numerical representation for consumption by your classifier

All these steps serve to reduce the noise inherent in any human-readable text and improve the accuracy of your classifier’s results. There are lots of great tools to help with this, such as the Natural Language Toolkit, TextBlob, and spaCy.

1. **Choosing Machine Learning Classifiers:**

There are a number of tools available in Python for solving classification problems. Here are some of the more popular ones:

* [TensorFlow](https://www.tensorflow.org/)
* [PyTorch](https://pytorch.org/)
* [scikit-learn](https://scikit-learn.org/)

These are the more widely used machine learning frameworks available in Python. They’re large, powerful frameworks that take a lot of time to truly master and understand.  spaCy provides a fairly straightforward built-in text classifier

Once you have your vectorized data, a basic workflow for classification looks like this:

* Split your data into training and evaluation sets.
* Select a model architecture.
* Use training data to train your model.
* Use test data to evaluate the performance of your model.
* Use your trained model on new data to generate predictions, which in this case will be a number between -1.0 and 1.0.

Machine learning practitioners often split their datasets into three sets:

* **Training:** The training set is used to train your model.
* **Validation:** The validation set is used to help tune the hyper parameters of your model, which can lead to better performance.
* **Test:** The test set is a dataset that incorporates a wide variety of data to accurately judge the performance of the model. Test sets are often used to compare multiple models, including the same models at different stages of training.

1. **Building Your Own NLP Sentiment Analyser:**

The four major stages of building a sentiment analysis pipeline:

* Loading data
* Preprocessing
* Training the classifier
* Classifying data

For building a real-life sentiment analyzer, you’ll work through each of the steps that compose these stages. We used the given dataset to train and test our sentiment analyser.

**Final Accuracy:** 0.8773064017295837 or 87.73064017295837% or 88% (Rounded off)