**Machine Learning ISE-3**

**Sentiment Analysis Using Machine Learning**

**Class:** TE COMP-B **Roll No:** 42

**Under the Guidance of :**

**Dr. Shiwani Gupta**

**Problem Statement**

In recent years, the rise of social media platforms and online shopping has led to an exponential increase in user-generated product reviews and opinions. Businesses and consumers alike heavily rely on these reviews to make informed decisions about purchasing products or services. The main challenge is to develop a robust and accurate sentiment analysis model that can automatically analyze and classify sentiments expressed in social media product reviews. The model should be capable of classifying sentiments as positive, negative, or neutral by understanding the true meaning of the given sentence for the given situation. By leveraging popular Python libraries and machine learning techniques, we aim to extract valuable insights from tweets and understand the overall sentiment surrounding specific topics.

**Introduction:**  
Sentiment analysis is a natural language processing (NLP) technique that focuses on understanding and interpreting emotions, attitudes, and opinions expressed in textual data. The main objective of sentiment analysis is to determine the sentiment polarity associated with a piece of text, whether it is positive, negative, or neutral. Applications of sentiment analysis are diverse, ranging from brand reputation management, market research, and social media monitoring to customer feedback analysis and product recommendation systems.

**Steps to develop Sentiment analysis:**

**Data Collection:**

The primary objective of this step is to gather a substantial dataset of text samples that are associated with their corresponding sentiment labels. Data can be collected from the platform for which we have to develop our sentiment analysis model like social media platforms, online reviews, customer feedback, or any other text-based sources that contain sentiment information.

The following are the points which should be taken into consideration while collecting data:

* **Dataset size:** The dataset's size is crucial for building a robust model. A larger dataset allows the model to learn from a diverse range of sentiments and patterns, leading to better generalization and performance on unseen data.
* **Diversity:** It is essential to ensure that the dataset is diverse and representative of real-world sentiments. This means the data should cover various topics, domains, and contexts to accurately reflect the sentiments that the model will encounter in real-world applications.
* **Data Quality:** The dataset should be real world and should be reliable.
* **Data Balance:** Class imbalances, where one sentiment class dominates the dataset, can lead to biased model predictions and poor performance on minority sentiment classes. So it is necessary to avoid it.

By taking into consideration all the above given points we selected ‘**Sentiment140 dataset’** from Kaggle. The Sentiment140 dataset consists of 1.6 million tweets, which have been labeled for sentiment analysis. Each tweet is annotated with a polarity label (0 for negative, and 4 for positive). The dataset includes information such as tweet IDs, dates, user names, and the actual text of the tweets. This data was collected using the Twitter API and was automatically annotated based on the presence of positive or negative emotions. It was created as part of a project at Stanford University in 2009 and is publicly available for research and learning purposes.

**Exploratory Data Analysis**

**1. Data Visualization:**

Data visualization is a technique used to represent information and insights from the dataset visually. It involves creating charts, graphs, or plots to better understand patterns, trends, and relationships in the data.

In the context of sentiment analysis, data visualization can be helpful in several ways:

* **Sentiment Distribution:** Visualizing the distribution of sentiment labels (positive, negative, neutral) helps understand the balance of different sentiments in the dataset. In our case, the data was perfectly balanced with 50% samples of positive and 50% samples of negative sentiments. A pie chart was drawn to visualize it.
* **Sentiments Over Time:** Sentiment distribution over time can show how sentiments change over time, revealing trends or seasonality. It can reveal trends in sentiment over time. This line plot was drawn to show how sentiment (negative and positive) varies over time.
* **Text Length Distribution:** It involves understanding the length of text samples, which can be important to determine the range within which most sentences fall. This gives a rough idea of how lengthy the sentences are that will be fed to the model at a time. A histogram was created to illustrate the distribution of text lengths. According to this, over 300,000 samples had lengths of 50 characters. The same distribution was performed for just positive and negative sentiments, yielding similar results as above. Finally, the mean text length was calculated to be approximately 74 characters.
* Top of Form
* **Density Plot of Text Lengths:** It involves the probability distribution of text lengths using a kernel density estimation(KDE) plot. If the curve is high at a certain value, it means that value is more common in your data. If the curve is low, it means that the value is less common. It provides a smoother representation of the distribution and helps visualize overlapping regions.
* **Sentiment Counts by User:** It involves identifying users who are particularly positive or negative, which may be valuable information in certain applications. A pie chart is used to represent the percentage of positive or negative comments of a user with respect to the total positive or negative comments. According to it ‘what\_bugs\_u’ is the user name with maximum(246) positive sentiments and ‘lost\_dog’ is the user name with maximum(549) negative sentiments
* **Word Clouds:** Word clouds provide a visual summary of the most common words, which can help in understanding the key topics or themes present in the data. I total, 3 word clouds for total comments, positive comments, and negative comments were included respectively.

**2. Data Preprocessing:**

Data preprocessing involves cleaning and preparing the raw text data for analysis. Several important steps are undertaken to make the data suitable for training the model:

* Converting the text to lowercase which helps to ensure that words with different cases are treated as the same, reducing the dimensionality of the data and avoiding duplication of words.
* Removing URLs because URLs contain a lot of random characters and symbols that do not contribute to the sentiment of the text. Removing them helps reduce noise in the data, allowing the model to focus on the meaningful content.
* Removing unnecessary characters, such as punctuation and special symbols because they are widely used in almost all the sentences and do not contribute any special meaning for predicting sentiment
* Removing numbers or we can say phone numbers because they too like punctuation do not contribute any special meaning in a sentence and can be widely present in almost all sentences.
* Removing stop words, which are common words like "and," "the," and "is," that do not carry significant sentiment. Removing stop words helps reduce noise and improves the efficiency of the model.
* Removing repeating characters as over-exaggeration does not convey any different meaning from the original word and it also helps to reduce the size or dimensionality of the data
* Tokenization is performed to split the text into individual words or tokens, enabling the model to process and analyze individual words effectively.
* Performing lemmatization helps reduce words to their base form (e.g., "running" to "run" or "better" to "good"). This standardizes the language and reduces variations in word forms, helping the model focus on the core sentiment-bearing words.

**Model Training**

During training, the model learns to map the input numerical features to the corresponding sentiment labels.

**1. Text-to-Numerical Conversion:**

Machine learning models require numerical inputs, so the preprocessed text data is transformed into numerical features. This step is crucial for training the model effectively. Two common techniques for text-to-vector conversion are used:

* Bag-of-Words (BoW) representation converts text into a numerical vector representing the frequency of words in each text sample. It creates a sparse representation of the text, focusing on word occurrences.
* Word embeddings, such as Word2Vec convert words into dense vectors, capturing semantic relationships between words. These representations capture the meaning and context of words.

**2. Model Selection:**

Choosing the appropriate model is a crucial decision for sentiment analysis. Several models are considered, including:

* **Naive Bayes:**
* Naive Bayes is used to predict the sentiment of a given text based on the words present in the text.
* The "naive" assumption in Naive Bayes is that the presence of one word in the text is independent of the presence of other words, given the sentiment class.
* To use Naive Bayes for sentiment analysis, the model first calculates the probabilities of each word occurring in positive and negative sentiment samples.
* When given a new text for sentiment analysis, Naive Bayes calculates the probabilities of the text belonging to each sentiment class (positive and negative) based on the frequencies of words in those classes
* It multiplies the probabilities of individual words in the text to get the probability of the entire text being in a specific sentiment class.
* **Logistic Regression:**
* It's commonly used for binary classification problems (where the outcome can be one of two classes, like "Yes" or "No")
* It first finds the weighted sum of the features using the equation of line(used in linear regression) and then
* It then uses a special function called the sigmoid function. This function takes the weighted sum and squashes it to a range between 0 and 1.It basically finds the probability of a given sentiment belonging to the positive (1) class.
* **Support Vector Machines (SVM):**
* SVM works by finding the best possible line or hyperplane that can separate data points belonging to different sentiment classes.
* Once the hyperplane is determined, new data points (new text samples) can be classified into positive or negative sentiments based on which side of the hyperplane they fall.
* **k- nearest Neighbour(k-NN):**
* k-Nearest Neighbors is a simple, instance-based learning algorithm used for both classification and regression tasks.
* Here, initially, we choose a value for 'k', which is the number of nearest neighbors the algorithm will consider when making a prediction. It then plots the instances as data points on the graph.
* For classification of the new instance, it first plots it and then identifies the 'k' data points that are closest to the new point and then takes a majority vote among the 'k' neighbors to determine the class of the new point. For regression, it takes the average of the 'k' neighbors' target values.

**Training**

* So for training, we first converted our dataset into a bag of words.
* For determining the number of columns (frequent) words in the bag of words representation an analysis was conducted in which a list holding values for a number of features was calculated using the ‘**Text Length Distribution’** step in EDA, if a sentence length is 74 character then lengths of the entire text of 16,00,000 sentences will be of 118,400,000 and if we assume that length of each word is approx. 5 characters then minimum(considering preprocessing step) there will be 2,00,00,000 words in our dataset.
* So for a list of the ‘number of features’, values were included ranging from 100 to 2,00,00,000. Both logistic regression and the Naïve Bayes model were trained on the bag of words data, with the number of columns set to each value in the list. The model's performance was evaluated using precision as the metric. The value that resulted in the highest precision was chosen for final model training.
* **Precision metric** was selected because it focuses on eliminating the false positives(people dislike something but it is predicted that they lied it) which can be crucial to eliminate in the case of various applications of sentiment analysis like product recommendation, etc where the product is recommended based on user liking.
* Then the ensemble modeling was done on the best-trained models of logistic regression and Naive Bayes with the help of hard and soft voting.The best precision was observed for hard-voting which was ‘0.7808581386963663’

**Model Optimization:**

Model optimization involves improving the model's performance through multiple techniques like fine-tuning hyperparameters, training new models, ensemble modelling, etc.

Initially, the size of the data in our case was reduced to 20,000 samples to reduce the time taken for training and testing . The data was divided considering that it remained balanced

Model Optimization in our case is divided into 3 categories

* Training New models on the bag-of-words approach

Two new models: k-NN and SVM were trained .

* + The k-NN model was trained on various values of k ranging from 5 to 50.The precision which is 62.685% was best observed for k=10
  + For SVM, the precision was observed to be 73%
* Using Word2vec approach:

Using word2vec approach 3 models were trained

Logistic Regression : 59% precision

KNN :57% precision

SVM :60% precision

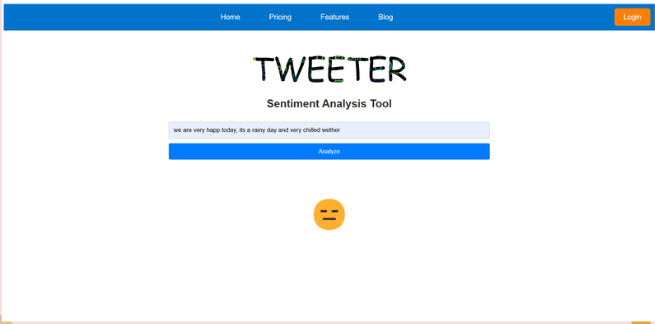
The third method involves making ensemble combinations of word2vec and BOW models

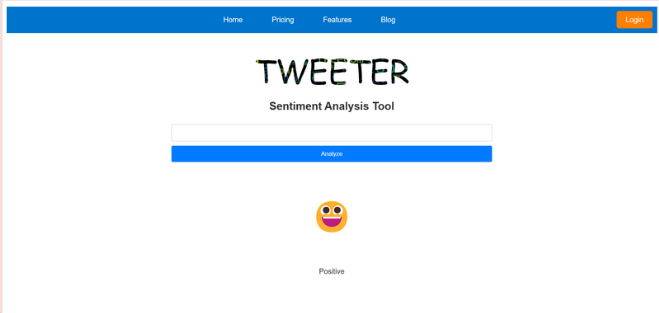


The best model ie is the ensemble model obtained from the combination of Logistic regression and Naïve Bayes giving precision 78%.

**Model Hosting**

* The model was deployed locally with help of flask
* A UI was created for the model providing for better user Interaction
* The UI is as follows





**Conclusion:**

Sentiment analysis is a powerful and valuable tool for understanding public emotions, opinions, and attitudes expressed in text data, especially in social media platforms like Twitter. Building an effective sentiment analysis model involves several key steps, starting with data collection from diverse and reliable sources. Preprocessing the data ensures that it is clean and suitable for analysis, while data visualization provides insights into sentiment distribution and trends. Transforming the text data into numerical features is essential for machine learning models, and techniques like Bag-of-Words and Word Embeddings are commonly used. Selecting the right model for sentiment analysis plays a critical role and Model training and evaluation on a validation set help assess the model's performance and accuracy.

So, by leveraging these steps an effective sentiment analysis model that provides valuable insights to businesses, researchers, and individuals to make informed decisions can be achieved.

