Data Analysis and Machine Learning 4 - Coursework 2

April 5, 2023

1 Abstract

This report investigates the use of machine learning techniques in sentiment analysis on text data collected from multiple sources. The objective of the project is to develop an accurate sentiment analysis model that can classify text data into different sentiments. The text data is pre-processed to remove noise. Supervised learning algorithms such as Random Forest, Naive Bayes, and Support Vector Machines (SVM) are used to train the sentiment analysis model on labeled data, and the performance of the model is evaluated using metrics such as accuracy F1-score, and confusion matrix. A chosen model will be used to classify sentiment for movies reviews.

2 Introduction

The widespread use of social media and the internet has made text data a valuable source of information for businesses, organizations, and individuals. Sentiment analysis is a subfield of natural language processing that allows for the extraction of insights from text data by analyzing sentiment. This technique can be used to understand customer opinions, preferences, and attitudes towards products, services, or events[1]. This report presents the results of a project aimed at conducting sentiment analysis on text data from multiple websites. The objectives of the project were to perform sentiment analysis on specific topics, and predict the sentiment of customer reviews. The report describes the methodology used to pre-process, analyze the data, and presents the findings and their implications for decision-making.

3 Data Analysis

The data used in this project was provided on the university platform and consisted of text, source, and sentiment. The text represented comments, while the source was the website/category from which the comments were extracted. The sentiment labels were either positive or negative for comments from movies, stock, tweets, and Yelp. In contrast, comments about climate were labeled as being pro, neutral, anti, or factual news.

Fig. 1 shows that the sentiment of the text varies across different sources. Yelp text has a similar ratio of positive and negative sentiment, while the movie category has an almost equal number of positive and negative sentiment text. Stock data has more positive text than negative, and tweets have more neutral texts than positive and negative texts. The climate data category has mostly "pro" sentiment text, with the least being "anti" sentiment.

3.1 Common Words:

To gain an understanding of the contents and composition of the "Sentiment Soup" dataset, it is crucial to examine the frequently occurring words within it. The process involved using the stop words list from NLTK to eliminate unnecessary words from every review[2]. Subsequently, a word cloud was generated to highlight the frequently used words. Despite this, some words imply that additional preprocessing is required. For example, in Fig. 2 the presence of "https" suggests the need to remove links from the text.

By looking at the unprocessed text, it is apparent that one frequently occurring word does not contribute any additional significance to the text's meaning. This motivates us to preprocess the data prior to modeling and training.

3.2 Preprocessing

Preprocessing plays a significant role in preparing raw data for analysis and modeling. This involves cleaning and converting data into a more usable format since raw data can often contain various kinds of errors, inconsistencies, and irrelevant information, which can adversely affect the accuracy and performance of models. By applying preprocessing methods before modeling and training, we can ensure that our models are dependable, precise, and efficient.

3.2.1 Remove Punctuation and Stop Words:

Eliminating punctuations and stop words is crucial in natural language processing as they add little to the sentence's significance and can impede accurate text analysis. Stop words, including "and" and "the," are frequently removed to prioritize meaningful words, while punctuations like commas and periods can hinder text parsing, leading to language model inaccuracies. Removing these elements enhances the efficiency and accuracy of natural language processing models[3], [4]. Some words within NLTK's list possess significant relevance to the sentiment analysis task, such as "should" and "shouldn't". Hence, these words will be excluded from the stop words list.

The removed words are : again, no', nor, not, can, will, do ,don, don't, should, should've, ain, aren, aren't, couldn, couldn't, didn, didn't, doesn, doesn't, hadn, hadn't, hasn, hasn't, haven, haven't, isn, isn't, ma, mightn, mightn't, mustn, mustn't, needn, needn't, shan, shan't, shouldn, shouldn't, wasn, wasn't, weren, weren't, won, won't, wouldn, wouldn't

3.2.2 Remove Repeated Characters:

Social media users often employ unconventional grammar, abbreviations, and misspellings (e.g., "ohhhh," "wowww," "likeee") to convey emotions. To facilitate proper interpretation by computers in NLP applications, these repetitive and non-standard characters were removed in some representations of the reviews[5], [6].

3.2.3 Remove URLs, Account Names, and Hashtags:

Eliminating URL links, account names, and hashtags from text is crucial for sentiment analysis, as they add no value to the text's overall meaning and can potentially create distortion and partiality. By eliminating them, the analysis can be more precise and dependable, focusing solely on the relevant text[7].

3.2.4 Remove HTML Tags and Non-ASCII Characters

To ensure accurate analysis, HTML tags and non-ASCII characters should be eliminated from text. These tags and characters lack semantic meaning and can potentially disrupt the analysis process[8].

3.2.5 Case Normalization

Case normalization is important in natural language processing because different cases of the same word can have different meanings. Normalizing text to lowercase or uppercase can help in achieving consistency in the representation of text data, reduce the number of unique words, and improve the accuracy of text classification tasks[9].

3.3 Common Words - Revisited:

Now that the text has undergone preprocessing, we can revisit the word clouds and observe that the revised version provides a more accurate depiction of the text and the project's objectives. Fig. 3 show word clouds of different categories after pre-processing the text. It is worth noting that repeated characters were eliminated during preprocessing. For instance, the words "food" and "good" will appear as "fod" and "god" in the word cloud.

3.4 Text Length and Source Type:

Our curiosity also led us to investigate whether the text content differs among various sources. To this end, the correlation between text length and the source was explored. As evidenced by the box plot in Fig. 4, we can confirm that there is indeed variation in the length of text across different sources.

3.5 Principle Component Analysis:

The visualization of the correlation between common was accomplished by utilizing Principle Component Analysis in combination with word2vec, which will be explained in more detail later. PCA was employed to condense the dimensions of each vector into a 2D representation. Fig. 5 displays that the word2vec depiction of the words retains crucial information about them. This is evident in the proximity of words like "good", "bad", and "great" to each other, as well as "movie" and "film".

4 Machine Learning

In recent years, there has been a growing interest in the use of machine learning for processing languages and determining the emotional tone of a piece of text. In this section, we will discuss the implementation of machine learning for sentiment analysis and explore the different approaches and techniques that can be used to achieve accurate and reliable results.

4.1 Classification Tasks

4.1.1 Task 1: Predict Sentiment for Each Source:

The goal of this task is to train classifiers that can forecast the sentiment of text from a particular source. Each classifier will undergo individual training on the training set from that specific source to predict the sentiment of the text provided. The performance of the classifier will then be assessed using the test set.

4.1.2 Task 2: Classify Sentiment for All Sources:

The objective of this assignment is to train classifiers that can forecast the sentiment of text originating from all sources. Each classifier will undergo independent training on all the data from the various sources to predict the sentiment of the text provided. Afterwards, the performance of the classifiers will be assessed using the test set.

4.1.3 Task 3: Classify Source for All Sentiments:

The objective of this task is to train classifiers that can predict the source of text across various sentiments. Each classifier will undergo individual training on all available data to predict the source of the provided text. The performance of the classifier will then be evaluated using the test set.

4.2 Feature Extraction and Data Representation

In machine learning, extracting significant features from available data is crucial. Different techniques for representing and extracting features enable algorithms to concentrate on relevant data while disregarding less significant information. This leads to improved performance and accuracy of the algorithm.

4.2.1 Bag-of-Words:

Bag-of-words (BoW) involves transforming a document into a vector representation based on word frequency. The resulting vectorizer is then used as a feature extraction method to extract features from the training and testing sets. In each task, a vectorizer will be trained on the training set to be used later to vectorize both train and test sets.

4.2.2 Term Frequency-Inverse Document Frequency:

Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic used to determine the importance of a term in a document of documents.The overall score for a term is calculated by multiplying the term frequency with the inverse document frequency. In each task, a vectorizer will be trained on the training set to be used later to vectorize both train and test sets.

4.2.3 Word2Vec:

Word2vec generates word embeddings, which are numerical representations of words in a vector space. A pre-trained Word2vec model trained on a part of the Google News dataset will be used to extract features as 300-dimensional vectors from a document[10]. Word2vec may not be effective when the words employed in the text are absent from the vocabulary, as this can lead to the vector containing infinite values [11].

4.2.4 FastText:

FastText is a Facebook AI Research library that creates word embeddings and text classification. It enables unsupervised or supervised learning algorithms to obtain vector representations for words. A pre-trained FastText model in English language will be utilized for this project[12].

4.2.5 FastText Trained on Sentimental Soup:

FastText has an advantage over Word2Vec in that it can generate embeddings for out-of-vocabulary (OOV) words. This has led us to explore the potential of training a FastText model on the Sentimental Soup dataset to potentially enhance its performance.

4.2.6 Bidirectional Encoder Representations from Transformers:

Bidirectional Encoder Representations from Transformers (BERT) is a language model that uses Transformers to generate contextualized word embeddings. Trained on large amounts of text data, it has achieved state-of-the-art results in various natural language processing tasks including sentiment analysis.A pre-trained BERT model will be used in this project[13]. The encoder may generate vectors that contain negative values. However, certain classifiers only accept positive values. Hence, it is advisable to use the absolute value of the vectors returned by the encoder for such classifiers.

4.3 Classifiers

Selecting an appropriate classifier for machine learning classification tasks is crucial, given that classifiers possess varying strengths based on the nature of the data being analyzed. The project's time constraints necessitated focusing on hyperparameter tuning for using BoW to classify sentiment for all sources, with the intention of using the outcomes to carry out subsequent tasks.An 80% portion of the dataset will serve as the training set, while 20% will serve as the testing set. To tune the hyperparameters, three-fold grid search cross-validation will be used, and the hyperparameters of the top-performing model will be used for future classification tasks.

4.3.1 Maximum Entropy:

Maximum entropy classifier is a probabilistic machine learning technique that assigns probabilities to each possible outcome, given the observed data. It is suitable for NLP tasks, particularly sentiment analysis, because it can handle large feature spaces, model complex dependencies between features, and make accurate predictions even when the data is noisy or incomplete[14]. The results of hyperparameter tuning in Table 1 suggests that the optimum value for the inverse of regularization strength, C, is 0.1.

C	0.001	0.01	0.1	1	10
Accuracy	65.8%	72.9%	76.0%	75.8%	73.9%

Table 1: Average Validation Accuracy for Maximum Entropy Classifier's Hyperparameter Tuning

4.3.2 Random Forest:

A random forest classifier is a machine learning algorithm that creates a forest of decision trees using a random subset of features for each tree. It combines the predictions of multiple decision trees to produce a more accurate and stable prediction than a single decision tree. Based on the hyperparameter tuning results in Table 2, it can be inferred that the ideal number of trees is 100 and the maximum depth should be 20.

4.3.3 Naive Bayes:

Naive Bayes classifiers are a family of probabilistic machine learning models based on Bayes' theorem. They are simple yet effective algorithms for text classification tasks. **Multinomial Naive Bayes** classifier is suitable for text classification tasks, where each feature represents the frequency of a word or a term in a document. It assumes that the frequencies of different words are independent of each other, given the class label. **Bernoulli Naive Bayes** classifier is also suitable for text classification tasks, but it represents the presence or absence of words in a document using binary features. It assumes that the presence or absence of different words is independent of each other, given the class label. Table 3 shows that the optimum value of the smoothing parameter,alpha , for Multinomial NB classifier is 1 and for Bernoulli NB classifier is 0.1.

4.3.4 Support Vector Machine Classifier:

Support vector machine (SVM) classifier works by finding the optimal hyperplane that separates different classes of data in a high-dimensional feature space. However, SVMs require solving a quadratic optimization problem, which is computationally expensive. Stochastic Gradient Descent (SGD) is less expensive and can effectively handle non-linear relationships between the input features and target variable, so it will be used to train SVMs [15].Table 4 shows that the optimum value of the regulization term,alpha , for the SVM classifier is 0.001.

4.3.5 Multi-Layer Perceptron:

A multilayer perceptron (MLP) is a type of artificial neural network that consists of multiple layers of interconnected nodes. MLPs are used in sentiment analysis because they can effectively learn complex relationships between input data and output labels, allowing them to accurately classify text based on its sentiment. A 3 hidden layer MLP with 10 nodes in each hidden layer will be used.

4.4 Evaluation:

Assessing the models is a crucial step as it allows us to contrast various feature extraction techniques, classification methods, and so forth to determine which models perform better with the given data. Additionally, evaluating the models can inform us about the techniques that are better suited for the specific task at hand, providing insight into which approaches may be more effective in achieving the desired outcomes.

4.4.1 Accuracy:

Accuracy is a metric used to evaluate the performance of a classification model. It measures the proportion of correctly classified instances out of the total instances. It is calculated by dividing the number of correctly classified instances by the total number of instances.

4.4.2 F1 Score:

F1 score is a metric used to evaluate the performance of a classification model. It is the harmonic mean of precision and recall. F1 macro score takes the average F1 score across all classes, which is more relevant to sentiment analysis tasks where the dataset may have imbalanced classes and where it is important to equally weight the evaluation of each sentiment class.

4.4.3 Confusion Matrix:

A confusion matrix is a table that displays the true positives, true negatives, false positives, and false negatives for a classification model. It is important for evaluating a model's performance and identifying areas for improvement. It is relevant in sentimental analysis because it allows for a detailed analysis of the model's performance across different sentiment classes, helping to identify any biases or limitations in the model's predictions.

4.5 Setup:

The data for each task underwent filtering and was divided into training and testing sets, with the training set containing 80% of the filtered data. For each task, the same training and testing sets were used for different classification and feature extraction methods. Preprocessing was performed to represent text for each method, but removing duplicated characters was only done for BOW and TF-IDF representations. Following this, the models were trained, and accuracy and F1 score were computed, and the confusion matrix was displayed.

4.6 Results and Analysis:

4.6.1 Task 1 and Task 2:

It is evident that using FastText with Sentimental Soup to extract features is generally more accurate with a higher F1 score compared to the other methods. Word2Vec was only trained for Task 1 - Climate because of time constraints. The confusion matrices show that random forests are not ideal for this task as they tend to classify text to the most common class, while other classifiers perform well. Some classifiers, such as Multinomial NB, experience an accuracy drop when using BERT or pretrained FastText which can be due to using the absolute value function to represent the input vector for these methods. The highest classification results were obtained for the Yelp and movie data in Task 1 as it can be seen in Table 5-10.

4.6.2 Task 3:

All classifiers and feature extraction methods performed well in this section, with the exception that FastText trained on Sentimental Soup using maximum entropy classifier performed better than the other methods as it can be seen in Table 11.

4.7 Performance on External Data:

Classifying a new dataset with a trained model can help assess its generalization capability to reviews outside the dataset. The IMDB movie review dataset has labeled reviews as positive or negative. Preprocessing and classifying reviews using a maximum entropy classifier trained with FastText can determine model generalization. Evaluation methods are used to assess classification effectiveness. The model obtained a 90.2% accuracy on IMDB data with an F1 macro score of 0.902, indicating its ability to perform on external data and generalize beyond the trained dataset.

5 Conclusion:

In conclusion, this project successfully performed sentiment analysis on a diverse range of data categories including Yelp reviews, tweets, stock data, movie reviews, and climate data. Multiple methods were used to represent text and several classifiers were employed for classification. Additionally, the project demonstrated that the accuracy and F1 score of classifiers varied depending on the method used to represent text. Overall, this project highlights the importance of selecting appropriate methods for representing text and classifiers when conducting sentiment analysis on different data categories.

References

[1] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *Third IEEE International Conference on Data Mining*, 2003, pp. 427–434. DOI: 10.1109/ICDM.2003.1250949.

[2] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[3] S. N. Shivhare and S. Khethawat, "Emotion detection from text," *arXiv preprint arXiv:1205.4944*, 2012.

[4] R. M. Rakholia and J. R. Saini, "Lexical classes based stop words categorization for gujarati language," in *2016 2nd international conference on advances in computing, communication, & automation (ICACCA)(Fall)*, IEEE, 2016, pp. 1–5.

[5] M. Kanakaraj and R. M. R. Guddeti, "Nlp based sentiment analysis on twitter data using ensemble classifiers," in *2015 3Rd international conference on signal processing, communication and networking (ICSCN)*, IEEE, 2015, pp. 1–5.

[6] D. D. Gosai, H. J. Gohil, and H. S. Jayswal, "A review on a emotion detection and recognization from text using natural language processing," *International journal of applied engineering Research*, vol. 13, no. 9, pp. 6745–6750, 2018.

[7] F. Sun, A. Belatreche, S. Coleman, T. M. McGinnity, and Y. Li, "Pre-processing online financial text for sentiment classification: A natural language processing approach," in *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, IEEE, 2014, pp. 122–129.

[8] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, "Natural language processing (nlp) in management research: A literature review," *Journal of Management Analytics*, vol. 7, no. 2, pp. 139–172, 2020.

[9] A. Amaar, W. Aljedaani, F. Rustam, S. Ullah, V. Rupapara, and S. Ludi, "Detection of fake job postings by utilizing machine learning and natural language processing approaches," *Neural Processing Letters*, pp. 1–29, 2022.

[10] *Google code archive - long-term storage for google code project hosting.* [Online]. Available: https://code.google.com/archive/p/word2vec/.

[11] *Gensim: Topic modelling for humans*, Dec. 2022. [Online]. Available: https://radimrehurek.com/gensim/models/keyedvectors.html.

[12] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[13] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[14] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 workshop on machine learning for information filtering*, Stockholom, Sweden, vol. 1, 1999, pp. 61–67.

[15] R. G. Wijnhoven and P. de With, "Fast training of object detection using stochastic gradient descent," in *2010 20th International conference on pattern recognition*, IEEE, 2010, pp. 424–427.

6 Appendix:

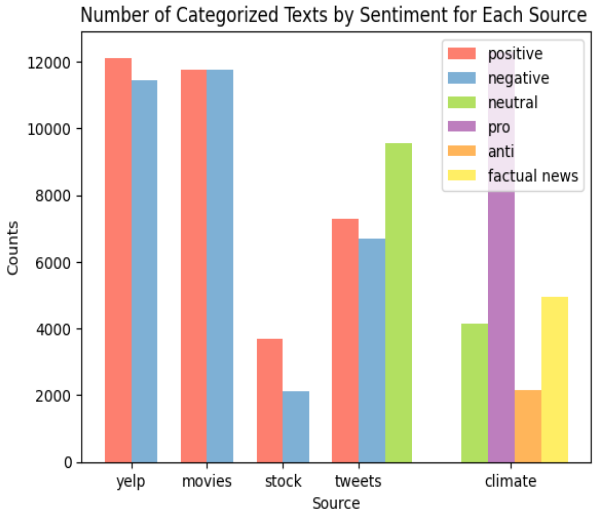


Figure 1: The figure shows the variation of sentiment for different sources

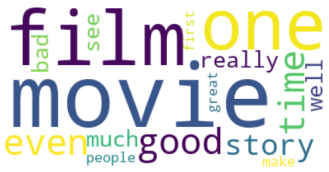
maximum depth/number of trees	10	50	100
10	43.8%	47.2%	47.7%
15	51.3%	48.0%	49.2%
20	49.9%	50.8%	51.9%

Table 2: Average Validation Accuracy for Random Forest Classifier's Hyperparameter Tuning

alpha	0.001	0.1	1	10	100
Multinomial NB Accuracy	67.0%	68.8%	69.0%	64.8%	58.8%
Bernoulli NB Accuracy	60.6%	61.5%	58.9%	47.4%	35.1%

Table 3: Average Validation Accuracy for Naive Bayes Classifiers' Hyperparameter Tuning

(a) climate word cloud

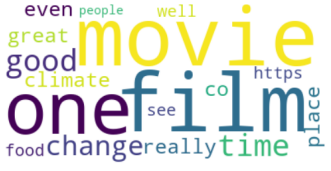


(b) movies word cloud

(c) stock word cloud

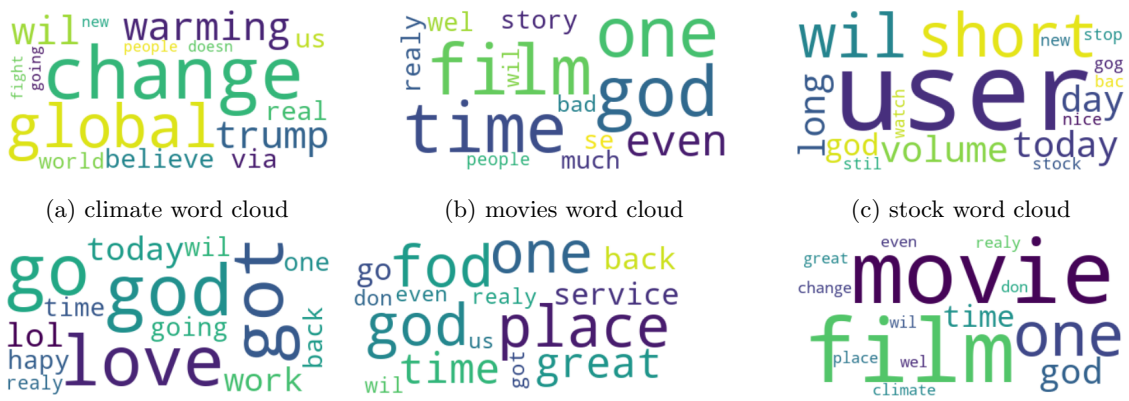
(d) tweets word cloud

(e) yelp word cloud



(f) all sources word cloud

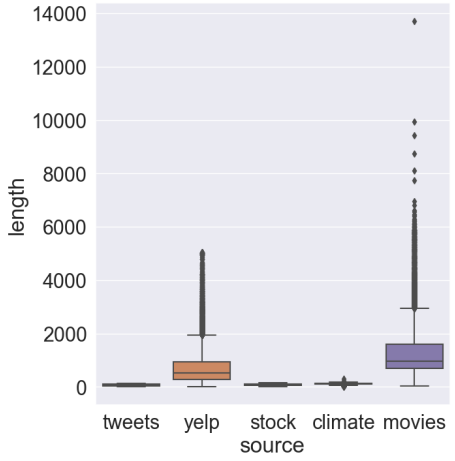
Figure 2: Word Clouds of Unprocessed Text Across Different Categories



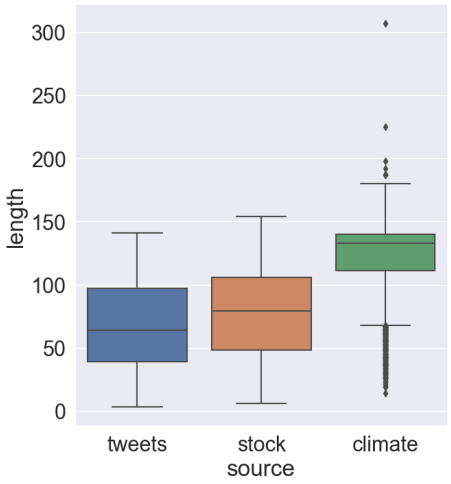
(a) climate word cloud (b) movies word cloud (c) stock word cloud

(d) tweets word cloud (e) yelp word cloud (f) all sources word cloud

Figure 3: Word Clouds of Preprocessed Text Across Different Categories



(a) sources : tweets, yelp, stock, climate, movies



(b) sources : tweets, stock, climate

Figure 4: Box Plots Showing Variations in Text Length Across Different Sources

alpha	0.001	0.1	1	10	100
Accuracy	73.5%	64.6%	59.7%	49.3%	42.85%

Table 4: Average Validation Accuracy for SVM Classifier's Hyperparameter Tuning Using SGD

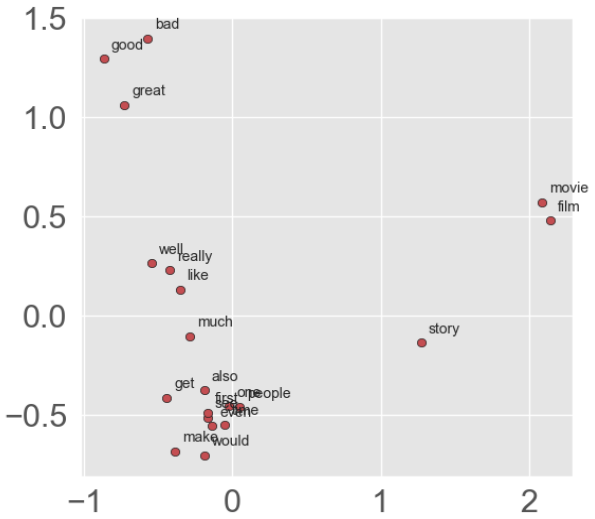


Figure 5: The figure shows a 2D representation of word embedding of common words in the dataset

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	67.5%	0.563	68.6%	0.578	66.7%	0.231	60.9%	0.372	68.7%	0.590	63.3%	0.471
R.F.	54.2%	0.195	54.6%	0.204	65.6%	0.312	66.2%	0.482	69.3%	0.617	63.9%	0.432
MNB	67.2%	0.549	61.1%	0.349	66.7%	0.231	53.5%	0.174	53.6%	0.176	53.5%	0.174
BNB	65.9%	0.590	65.9%	0.592	56.3%	0.327	53.5%	0.469	64.9%	0.596	52.9%	0.480
SVM	67.1%	0.537	69.0%	0.578	63.2%	0.450	57.2%	0.291	68.6%	0.580	60.1%	0.376
MLP	58.7%	0.508	60.7%	0.526	65.5%	0.418	66.3%	0.570	68.4%	0.609	64.4%	0.556

Table 5: Task 1: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for Climate Reviews

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	88.1%	0.881	88.4%	0.884	DO THIS%	DO THIS	79.0%	0.790	88.6%	0.886	80.9%	0.809
R.F.	85.2%	0.852	83.5%	0.835	- %	-	81.9%	0.819	88.6%	0.886	77.9%	0.779
MNB	85.5%	0.855	85.9%	0.859	- %	-	75.4%	0.754	49.8%	0.337	73.8%	0.737
BNB	84.7%	0.847	84.6%	0.846	-%	-	75.2%	0.752	88.7%	0.887	75.5%	0.755
SVM	87.6%	0.876	88.4%	0.884	-%	-	77.2%	0.772	88.6%	0.886	80.7%	0.806
MLP	86.8%	0.868	85.9%	0.858	-%	-	85.5%	0.855	88.7%	0.887	81.1%	0.811

Table 6: Task 1: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for Movies Reviews

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	88.1%	0.881	92.0%	0.92	-%	-	84.7%	0.846	91.4%	0.914	86.9%	0.869
R.F.	84.6%	0.846	82.5%	0.823	-%	-	86.2%	0.862	91.4%	0.914	83.9%	0.838
MNB	85.5%	0.855	88.6%	0.886	-%	-	75.6%	0.746	51.8%	0.341	76.8%	0.763
BNB	84.7%	0.847	78.8%	0.784	-%	-	77.1%	0.771	91.3%	0.913	80.3%	0.803
SVM	87.1%	0.871	92.3%	0.922	-%	-	84.4%	0.844	91.4%	0.914	87.5%	0.874
MLP	86.8%	0.868	90.4%	0.904	-%	-	90.0%	0.90	91.4%	0.914	87.5%	0.874

Table 7: Task 1: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for Yelp Reviews

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	77.9%	0.738	78.8%	0.751	-%	-	64.5%	0.403	79.4%	0.767	72.4%	0.636
R.F.	67.6%	0.497	68.0%	0.509	-%	-	72.4%	0.64	78.4%	0.767	72.0%	0.632
MNB	77.1%	0.742	73.6%	0.65	-%	-	64.1%	0.391	64.1%	0.391	64.1%	0.391
BNB	77.5%	0.751	77.4%	0.75	-%	-	66.1%	0.637	78.9%	0.77	69.3%	0.668
SVM	79.4%	0.77	78.8%	0.77	-%	-	64.3%	0.398	78.7%	0.768	75.3%	0.698
MLP	76.4%	0.743	77.3%	0.751	-%	-	70.1%	0.677	78.5%	0.767	73.5%	0.715

Table 8: Task 1: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for Stock Reviews

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	68.9%	0.687	68.9%	0.689	-%	-	65.2%	0.646	71.2%	0.713	69.4%	0.695
R.F.	49.9%	0.404	48.2%	0.371	-%	-	64.5%	0.639	70.3%	0.706	61.3%	0.601
MNB	65.8%	0.661	62.1%	0.607	-%	-	40.0%	0.196	64.6%	0.619	40.3%	0.204
BNB	63.1%	0.634	62.2%	0.624	-%	-	63.0%	0.632	71.0%	0.713	64.4%	0.647
SVM	68.9%	0.685	69.7%	0.696	-%	-	63.4%	0.622	69.8%	0.7	69.2%	0.694
MLP	60.7%	0.610	60.2%	0.605	-%	-	69.4%	0.698	70.8%	0.711	67.3%	0.676

Table 9: Task 1: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for Tweets Reviews

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	77.2%	0.674	77.1%	0.669	-%	-	69.9%	0.51	77.5%	0.687	72.4%	0.586
R.F.	51.1%	0.235	51.8%	0.25	-%	-	71.7%	0.577	77.5%	0.700	68.2%	0.527
MNB	70.1%	0.527	68.8%	0.43	-%	-	40.4%	0.143	64.8%	0.363	44.2%	0.195
BNB	61.8%	0.596	61.8%	0.591	-%	-	59.0%	0.535	74.6%	0.667	58.9%	0.539
SVM	73.8%	0.568	74.2%	0.588	-%	-	64.5%	0.394	75.8%	0.612	67.9%	0.427
MLP	72.6%	0.631	70.6%	0.612	-%	-	75.3%	0.649	77.5%	0.697	74.3%	0.656

Table 10: Task 2: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for All Reviews

Clf	BOW Accuracy	BOW F1 Score	TFIDF Accuracy	TFIDF F1 Score	W2V Accuracy	W2V F1 Score	Pre-Trained FT Accuracy	Pre-Train FT F1 Score	FT on Sentimental Soup Accuracy	FT on Sentimetnal Soup F1 Score	BERT Accuracy	BERT F1 Score
M.E.	97.0%	0.956	97.5%	0.968	-%	-	93.3%	0.915	98.0%	0.974	97.1%	0.960
R.F.	90.7%	0.749	90.5%	0.748	-%	-	96.1%	0.939	98.0%	0.974	94.3%	0.906
MNB	91.2%	0.906	88.1%	0.854	-%	-	90.0%	0.742	92.0%	0.845	87.9%	0.728
BNB	94.6%	0.946	94.7%	0.947	-%	-	92.9%	0.898	97.5%	0.968	93.8%	0.917
SVM	96.1%	0.935	96.9%	0.952	-%	-	91.0%	0.879	97.9%	0.971	96.5%	0.952
MLP	96.6%	0.955	95.7%	0.936	-%	-	96.7%	0.957	98.0%	0.974	96.8%	0.958

Table 11: Task 3: Accuracy and F1 Score for Different Classifiers using Different Feature Extractors for Source Classification