Prototyping a Reusable Sentiment Analysis Tool for Machine Learning and Visualization

CAREN AMBAT PACOL
Information Technology Department,
Pangasinan State University,
San Vicente, Urdaneta City, Pangasinan,
PHILIPPINES

Abstract: - Evaluating customer satisfaction is very significant in all organizations to get the perspective of users/customers/stakeholders on products and/or services. Part of the data obtained during the evaluation are observations and comments of respondents and these are very rich in insights as they provide information on the strengths as well as the areas needing improvement. As the volume of textual data increases, the difficulty of analyzing them manually also increases. With these concerns, text analytics tools should be used to save time and effort in analyzing and interpreting the data. The textual data being processed in sentiment analysis problems vary in so many ways. For instance, the context of textual data and the language used vary when data are sourced from different locations and areas or fields. Thus, machine learning was utilized in this study to customize text analysis depending on the context and language used in the dataset. This research aimed to produce a prototype that can be used to explore three vectorization techniques and selected machine learning algorithms. The prototype was evaluated in the context of features for the application of machine learning in sentiment analysis. Results of the prototype development and the feedback and suggestions during the evaluation were presented. In future work, the prototype shall be improved, and the evaluators' feedback will be considered.

Key-Words: - Machine Learning, Max Voting Ensemble, Natural Language Processing, Prototyping, Sentiment Analysis, Sentiment Scoring, Vectorization Techniques, Visualization.

Received: August 29, 2023. Revised: February 2, 2024. Accepted: March 2, 2024. Published: April 15, 2024.

1 Introduction

Customer satisfaction is very significant in all organizations. As such, it has been a practice to conduct surveys to get the perspective of users/customers/stakeholders on products and/or services. Aside from the numerical ratings obtained upon aggregating results during evaluations, it is also important to note the observations and comments of respondents. This data is very rich in insights as it provides information on the strengths as well as the areas needing improvement. These can become the basis for developing plans of action for enhancements or intervention measures. To ensure the reliability of data, there should be enough samples drawn. However, as the volume of textual data increases, the difficulty of analyzing them manually also increases. With these concerns, text analytics tools should be used to save time and effort in analyzing and interpreting the data. Machine learning can be taken advantage of to perform text analytics.

In the realm of computer science, machine learning employs Artificial Intelligence (AI) within systems to imbue them with intelligence. Machine learning is centered around developing algorithms capable of potential deployment in real-world AI applications. As enterprises are producing huge amounts of data, it became indispensable to have machine learning techniques in place for discovering business intelligence from data for strategic decision-making, [1]. Machine learning belongs to supervised learning in general and text classification in particular. Thus, it is also called "Supervised Learning". Examples of techniques under supervised learning include Naïve Bayes, Support Vector Machine, Maximum Entropy, K-Nearest Neighborhood, and Neural Networks, [2]. Machine learning enables the capture and analysis of nuances and significance within customer feedback from diverse channels.

One great edge of machine learning is the capability to train algorithms. Natural Language Processing (NLP) along with sentiment packages, sentiment corpora, and other human-labeled sentiment rules are used to continuously improve algorithms thereby making them faster and more

accurate. Machine learning methods have limitations and cannot work at a character level like humans. To improve the performance of the methods some transformations on the original data can be used that make it easier to be processed by the machine learning methods. Each sentence is converted into a vector of features using a Count vector or Term Frequency-Inverse Document Frequency (TF-IDF) representation. Another transformation called Ngrams makes the relation between words more explicit. Others are focused on generating new feature vectors that try to represent the data in a more compact way and the rest are focused on modifying the original data to reduce the feature vector size, [3].

Many available sentiment analysis tools can be used. While these tools can be immensely helpful in various applications, they do come with limitations. Not all of them may be affordable or accessible to everyone. Some advanced sentiment analysis tools may come with high costs or require specialized expertise to use effectively. Since they are typically trained in specific languages or datasets, they may not perform well when analyzing text in different languages or contexts. Thus, the flexibility of these tools could vary depending on the type of dataset fed to the system. A study conducted by [4], found that research scholars primarily utilize tools such as WEKA, R Studio, and Python for implementing sentiment analysis. Programs to run text analysis can be constructed using Python or R. Weka and RapidMiner are also good alternatives for those who prefer a no-code development. Python and R are both very powerful tools that can be used in sentiment analysis, however, they both require programming proficiency. On the other hand, Weka and RapidMiner provide customized workflows and functions, but some users might find them difficult to use. For instance, some users of RapidMiner find difficulties in finding key operators, [4]. From these concerns, a prototype for a reusable sentiment analysis tool with machine learning and visualization was conceptualized. This tool is intended to allow users to devise their training dataset, evaluate and select algorithms to use, create their model, and use the generated model to analyze new dataset. It may become beneficial to students and researchers who are not so inclined to write code but would like to explore the application of machine learning in sentiment analysis in a no-code interface.

Some practical applications of the sentiment analysis tool include understanding customer opinions about their products or services. Users can collect data from platforms to devise a training dataset. They can then evaluate and select algorithms by experimenting with different machine-learning models. Once the model is created, they can use it to analyze new data to gain insights into customer sentiment trends and feedback.

Prior studies have been conducted wherein authors developed tools for visualizing sentiments. In [6], the authors developed a real-time Twitter sentiment analysis and visualization system which they called TwiSent. The system evaluates sentiments as either positive or negative regarding a specific product or service, aiding organizations, political parties, and individuals in gauging the impact of their endeavors and facilitating improved decision-making. In another study by [7], the authors extracted data on public opinion about property and developed a dashboard to visualize sentiment results toward their housing or construction projects. The study [8], also developed a sentiment analysis and information visualization tool to support the evaluation of usability and user experience. They undertook a user study to evaluate the efficiency of data communication in the visualizations, yielding valuable insights for enhancing the dashboard. In [8], a data visualization system was created utilizing a corpus-based approach to help customers comprehend the quality of Internet services provided by various Internet Service Providers (ISPs). In another investigation conducted by [10], an approach was devised whereby entities are dynamically generated through natural language processing of product reviews written in Japanese, and their associated sentiment values are aggregated to generate a visual representation. Another study conducted by [10], employed the Valence Aware Dictionary for Sentiment Reasoning (VADER) to create heatmaps, aiming to visually support the findings and enhance their comprehensibility. In [12], the authors showed visualization tools of sentiment analysis for real-time data of declaration of COVID-19 pandemic. They used Google Data Studio, Python Matplotlib, Carto, and Tableau. Another study by [13], performed real-time sentiment analysis on tweets visualizing the outcomes via distribution charts and tracking sentiment trends across time.

The previous studies conducted developing tools in sentiment analysis were significant because they provided insights into features of tools intended for sentiment analysis. They were, however, subject to limitations. These tools developed in prior studies provided visualizations of sentiment analysis results but did not include enabling their users to train and retrain models, evaluate, and visualize their

performance. Hence, the author aimed to bridge this gap. Specific objectives of this study are: (a) to develop the prototype for machine learning and visualization and (b) to evaluate the prototype in the context of features for machine learning applications in sentiment analysis.

This study builds upon findings of two prior research investigations conducted by [14] and [15]. The first study by [14], formulated a strategy to calculate teacher performance by analyzing textual feedback using a bilingual lexicon approach implemented in R. The second study by [15], conducted experiments in sentiment analysis by applying machine learning algorithms and vectorization techniques, evaluating the performance of various sentiment analysis models to determine the best-performing model. The superior-performing model, identified among the others, was employed alongside visualization techniques to provide comprehensible results of sentiment analysis. This present research serves as a continuation of the previous works, aiming to further explore and expand upon the methodologies and conclusions established in the earlier studies.

2 Methodology

The methodology is divided into the following steps.

2.1 Requirements Planning

The prototype was designed to integrate sentiment classification using machine learning algorithms and sentiment scoring using a lexicon-based approach. The sentiment scoring is based on the approach used in [14], which was integrated into the system using VADER while sentiment classification was implemented using the methodology presented in [15].

2.1.1 Sentiment Classification

The processes conducted in sentiment classification were identified and these are the following:

(1) Create training dataset
(2) Create a sample dataset for testing the model
(3) Use machine learning algorithms to create models and analyze their performance
(4) Select an algorithm and create the model using the selected algorithm
(5) Use the model to analyze new data
(6) Visualize the results

The training dataset is prepared by cleaning the collected textual data, removing sentences that do not have clear meanings, correcting misspelled words, and annotating each sentence with appropriate labels. For instance, sentences are marked as positive, negative, or neutral. A sample dataset for testing the model is also prepared. Unlike the training dataset, the sample dataset does not have annotations. During testing, both the training and test datasets undergo pre-processing.

Pre-processing techniques were used wherein special characters were removed, multiple spaces were substituted with a single space, prefixes were removed, all text was converted to lowercase, and stop words were removed. The Python NLTK package was utilized for pre-processing. Four machine learning algorithms naïve bayes, support vector machines, logistic regression, and random forest were used. An ensemble learning algorithm called random forest was utilized. This ensemble learning algorithm produces decision trees using the data samples. Then it obtains predictions from each of these decision trees. Finally, it selects the best solution through voting. Random forest algorithms are considered superior compared to single decision trees. This is because they avoid overfitting by aggregating the results, [16].

An ensemble of the four algorithms was also included. Single models were combined into one model using a Max Voting ensemble. Generally, max voting is utilized for classification problems. This approach involves using several models to make predictions, with each model's prediction considered a 'vote'. The final prediction is determined by aggregating the votes from most of the models.

Three (3) vectorization techniques were used namely: count/bag-of-words, term frequency–inverse document frequency (TF-IDF), and ngrams. Ngrams and TF-IDF are approaches in text vectorization. Vectorization is the transformation of text into a meaningful vector or array of numbers a machine can understand. In the count vector, the sentence is represented by the words and the number of occurrences of each word in the document generating a bag of word counts for each sentence, [17]. With TF-IDF, more information can be encoded into the vector. Term Frequency-Inverse Document Frequency (TF-IDF) assesses the importance of a word within a corpus of text data. Term frequency (TF) gauges the relevance of terms within documents, while Inverse Document Frequency (IDF) measures the significance of terms across the entire corpus. So, the multiplication of TF and IDF of a word produces the frequency of this word in the document multiplied by the uniqueness of the word, [17]. TF is computed using Eq. 1. If the

term frequency is Tf(wi, D) and document frequency is Df(wi). Then, from Df(wi), inverse document frequency Idf(wi) is calculated using Eq. 2.

Tf(wi, D) = (# of times term w appears in document D) / (Total # of terms in document D) (1)

Idf(wi) = loge (Total # of documents / Df(wi)) (2)

The TF-IDF of feature wi for document D is then calculated as the product: Tf(wi,D) . Idf(wi). The words with high TF-IDF scores in a document frequently occurred in that document and delivered the most important facts about the document, [17].

N-grams consist of words grouped either individually as unigrams or in pairs as bigrams, and so forth. For instance, the sentence "I'm not well" is transformed into the vector ("I'm", "not", "well") for unigrams and ("I'm not", "not well") for bigrams.

A confusion matrix and classification report were used to enable users to assess the classification model. The confusion matrix allows users to view a summary of prediction results identifying the number of correctly and incorrectly classified sentences in each class. To further analyze the performance of the sentiment classifier, the classification report shows the weighted average and macro-average results as well as precision, recall, f1-score, and accuracy for each class. To get the weighted average, individual true positives, false positives, and false negatives are totaled for the various classes and applied to get the statistics. The macro-average calculates the average of the precision and recall values across various classes within the system. Computations for precision, recall, and F1 score in each class are in Eq. 3, Eq. 4, and Eq. 5 while overall accuracy is in Eq. 6. Precision provides the percentage of correct positive predictions, **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**. It is calculated utilizing Eq. 3.

Precision = True Positives / (True Positives + False Positives) (3)

Recall answers what percent of the positive cases were caught, **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**. It is calculated using Eq. 4.

Recall = True Positives / (True Positives + False Negatives) (4)

The F1 score in Eq. 5 is used to seek a balance between precision and recall, **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**.

F1 Score = 2*(Recall * Precision) / (Recall + Precision) (5)

Accuracy measures how many are correct predictions among total predictions and is calculated with Eq. 6.

Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative) (6)

Furthermore, Cohen's Kappa and Matthew's Correlation Coefficient (MCC) were also calculated to further support the evaluation results. All of these are metrics that can be used for evaluating multi-class classifiers on unbalanced datasets. Cohen's kappa is a statistical measure used to assess agreement between annotators. This function computes Cohen's kappa, a score that expresses the level of agreement between two annotators on a classification problem and is defined in Eq. 7.

K= (p0 – pe) / (1- pe) (7)

The p0 is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and pe is the expected agreement when both annotators assign labels randomly. The pe is estimated using a per-annotator empirical prior over the class labels, [19]. The Matthews correlation coefficient is employed in machine learning to evaluate the effectiveness of both binary and multiclass classifications. It considers true positives, false positives, true negatives, and false negatives, making it a balanced metric suitable for scenarios where class sizes differ significantly. MCC is calculated with Eq. 8.

MCC = (TP*TN – FP*FN) / √(TP+FP) (TP+FN) (TN+FP) (TN+FN) (8)

The MCC is in essence a correlation coefficient value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 is an average random prediction, and -1 is an inverse prediction. The statistic is also known as the phi coefficient, [20].

To illustrate the process of training a machine learning model for text classification, a dataset comprising 18,004 textual samples gathered in a prior study was utilized to construct the training dataset. The training data comprises 11,483 positive

sentences, 4,781 negative sentences, and 1,740 neutral sentences. Performance evaluation results of the machine learning models are presented in Section 3.1.

Visualizing the results takes place during the performance evaluation of machine learning models, where results are conveyed using a heatmap for the confusion matrix and tables for the classification report, Cohen's Kappa, and MCC. The visualization of results also occurs when the sentiment classifier is used to analyze new data, employing word clouds, bar graphs, pie charts, and tables.

2.1.2 Sentiment Scoring

The processes involved in sentiment scoring are as follows:
(1) Modify the VADER lexicon
(2) Load the dataset
(3) Calculate sentiment score
(4) Visualize the results

The VADER lexicon is modified by adding more sentiment terms encountered in the textual dataset. Each sentiment term is assigned a score, which can be either a positive or negative value. Analysis of data through sentiment scoring was performed using VADER Sentiment Intensity Analyser to calculate Sentiment Score. Sentiment Intensity Analyser uses the VADER Lexicon. VADER is a long-form for Valence Aware and Sentiment Reasoner, a rule-based sentiment analysis tool. VADER analyzes text to discern emotions and categorize them as positive, neutral, or negative. This analyzer calculates text sentiment and produces four different classes of output scores: positive, negative, neutral, and compound, [21].

Visualizing the results of sentiment scoring utilizes tables and text highlighting, where positive sentences are highlighted in green, negative sentences in pink, and neutral sentences in white.

2.2 User Interface Design

The initial interface design was created using dash plotly in Visual Studio integrating dash core components, dash bootstrap components, and hypertext mark-up language.

2.3 Coding and Development

Coding and development started with the use of Python implemented using dash plotly in Visual Studio. Then the codes were migrated to pythonanywhere.com for user evaluation.

2.4 Evaluation of Prototype

As described in section 2.1, the methodology implemented in the designed prototype was based on the processes conducted in [15]. Hence, an evaluation questionnaire focused on the features that perform these processes was devised. The instructions, general information, and section labels were adapted from the evaluation questionnaire in [22]. The System Usability Scale (SUS), [23], was also used as reference. The five response options (from Strongly agree to Strongly disagree) for respondents in the SUS were adopted.

Researchers who were inclined toward sentiment analysis and machine learning were selected as respondents for the evaluation. To facilitate the evaluation, a walk-through of the prototype was prepared and shared with the respondents. In each section, a space was provided to allow respondents to write their comments and suggestions freely on each of the features. The link to the prototype hosted on pythonanywhere.com was also shared with respondents, and sample training and test datasets were provided. The survey was administered using Google Forms and respondents were requested to fill in the survey form after viewing the walk-through and exploring the prototype of the tool. The evaluation questionnaire used is presented in Table 1 (Appendix).

3 Results and Discussion

This section presents and discusses the results of this study. The results are divided into three sections. First, the performance evaluation results of the machine learning models on the sample dataset used in this study. Second, the screenshots and discussion of the developed prototype. Third, the results of the evaluation of the prototype.

3.1 The Performance Evaluation Results of Machine Learning Models used in the Study

Using textual data collected from previous work, the machine learning algorithms' performance was assessed after applying Count (bag-of-words), TF-IDF, and Ngram vectorization techniques to the dataset, along with the execution of pre-processing steps.

Due to the imbalance in the number of instances in each class, the weighted average results for precision, recall, and F1 score were considered in the training dataset when applying Count (bag-of-words), TF-IDF, and Ngrams.

The outcomes of Count vectorization in sentiment classification demonstrate that among the

base models, Logistic Regression and Support Vector Machines achieved higher weighted average precision, recall, and F1-score. Nevertheless, the ensemble method surpassed both Logistic Regression and Support Vector Machines, achieving precision, recall, and F1 scores of 0.87, 0.88, and 0.87, respectively, compared to Logistic Regression and Support Vector Machines, which attained identical scores of 0.86.

Table 2. Assessment of Machine Learning Algorithms' Performance Using Ngrams Vectorization for Sentiment Classification

Metric	**Machine Learning Algorithms**				
	Logistic Regression (LR)	**Naïve Bayes (NB)**	**Support Vector Machines (SVM)**	**Random Forest (RF)**	**Ensemble (LR+NB+SVM+RF)**
Accuracy	0.87	0.82	0.87	0.84	0.88
Weighted average					
Precision	0.87	0.82	0.86	0.85	0.88
Recall	0.87	0.82	0.87	0.84	0.88
F1	0.87	0.80	0.86	0.84	**0.88**

Results from TF-IDF vectorization indicate that Support Vector Machines achieved the highest precision and recall of 0.87, while both Support Vector Machines and Logistic Regression attained the same F1 score of 0.86. Logistic Regression obtained a precision and recall of 0.86. Naïve Bayes yielded a precision of 0.85, a recall of 0.84, and an F1 score of 0.83. Random Forest achieved precision, recall, and F1 scores of 0.85. Meanwhile, the ensemble achieved precision, recall, and F1 scores of 0.86. This suggests that the ensemble did not surpass Support Vector Machines and Logistic Regression in terms of F1 score. TF-IDF increased the precision and recall of Support Vector Machines by 0.01, but there was no improvement in the F1 score. Furthermore, TF-IDF did not enhance the precision, recall, and F1 scores of the other machine learning algorithms, including the ensemble. Results from Ngram vectorization indicate that Ngrams outperformed Count in terms of F1 score when applied to the ensemble.

The weighted average results for Ngrams in the training dataset are shown in Table 2. Table 2 demonstrates that setting ngram_range to 1, 2 (combining unigrams and bigrams) resulted in the highest precision, recall, and F1 score when applied to the ensemble in the training dataset. Table 3 presents a comparison of the F1 scores achieved by the ensemble using Count and Ngrams. The findings reveal that Ngrams surpassed Count in terms of F1 score when implemented in the ensemble.

Table 3. Overview of Machine Learning Algorithms Achieving the Highest F1 Score in Sentiment Classification

Metric	**Machine Learning Algorithms**	
	Ensemble (LR+NB+SVM+RF) + Count	**Ensemble (LR+NB+SVM+RF) + Ngrams (1, 2)**
F1	0.87	**0.88**

Cohen's Kappa and Matthews Correlation Coefficient (MCC) were employed for additional assessment of the classifiers on an imbalanced dataset. When Ngram vectorization was configured with a ngram_range of 1, 2 (representing unigrams + bigrams) and applied in ensemble, it yielded Cohen's Kappa and MCC scores closest to 1. A Cohen's Kappa of 0.76 suggests substantial agreement between the predicted and actual values. This reinforces the findings that the ensemble approach, when applied with Ngrams, is favored in this case.

Table 4 compares these metrics across Logistic Regression, Support Vector Machines, and the ensemble, as they have shown superior results compared to Naïve Bayes and Random Forest according to the classification report.

Table 4. Comparative Analysis of Cohen's Kappa and Matthews Correlation Coefficient (MCC) for Logistic Regression, Support Vector Machines, and Ensemble Utilizing Count and Ngrams in Sentiment Classification

Machine Learning Algorithm + Text Vectorization Technique	**Metrics Cohen's Kappa**	**MCC**
Logistic Regression +Count	0.7220	0.7244
Logistic Regression +Ngrams(1, 2)	0.7386	0.7420
Logistic Regression +Ngrams(1, 3)	0.7362	0.7402
Logistic Regression +Ngrams(1, 4)	0.7337	0.7379
Support Vector Machines +Count	0.7299	0.7312
Support Vector Machines +Ngrams(1, 2)	0.7331	0.7312
Ensemble (Count)	0.7518	0.7536
Ensemble ngrams(1, 2)	**0.7586**	**0.7605**
SVM(tf-idf)	0.7317	0.7359

3.2 The Prototype for a Sentiment Analysis Tool

The prototype has four (4) sections namely Main page, Analyze Performance of Machine Learning Algorithms, Create Sentiment Classifier, and Classify New Dataset using Sentiment Classifier.

3.2.1 The Main Page

Figure 1 shows the main page which includes a hyperlink to *Guide to Users* to provide insights on the tool and how to explore the prototype. The three functionalities are also presented. When the user click the "Analyze Performance of Machine Learning Algorithms", the machine learning model performance visualizer is shown. The sentiment classifier generator is displayed when "Create Sentiment Classifier" is selected. Finally, the sentiment classifier visualizer is shown when "Classify New Dataset using Sentiment Classifier" is chosen.

Guide to Users

The guide to users is shown in Figure 2. It informs the user of the purpose of the prototype. The three functions of the tool were identified, and additional details were described.

Fig. 1: The Main Page

Fig. 2: Guide to Users

3.2.2 Analyze Performance of Machine Learning Algorithms

Machine Learning Model Performance Visualizer

In Figure 3, a 'Guide Me' tooltip is displayed. This tooltip is provided in each section to offer instructions to users. Users may prepare a list of stop words. Next, they should upload the training dataset by clicking the 'Browse csv file...' button. Subsequently, users can select one of the machines learning algorithms and specify a vectorizer for each algorithm. If the user does not select a vectorizer from the dropdown menu, the tool will assume the Ngram vectorizer. To begin analyzing performance, users must click the 'Check Performance Metrics' button.

Preparation of Stop Words List

When the "Prepare for Stopwords Removal" checkbox in Figure 4 is ticked, the user may tick on "Show Stopwords" to view the default stopwords list.

Adding Stop Words to List

Additional stopwords can be added to the list through the 'Enter stopwords' input field, as demonstrated in Figure 5. To add a new stop word to the list, the user must select the 'Add to list' button.

Loading Training Dataset

Figure 6 shows the count of rows and number of samples for each label in the training dataset. These would inform the user of a balanced or unbalanced training dataset as this affects the selection of performance metrics to use for evaluating the models.

Visualizing Performance of Models with Confusion Matrix

The confusion matrix presented in Figure 7 enables the user to visualize the performance of the model in each class. True labels and predicted labels are shown to help the user identify the number of correctly classified instances for each class.

Visualizing Performance of Models with Classification Report, Cohen's Kappa, and Matthew's Correlation Coefficient

The classification report, along with Cohen's Kappa and Matthew's Correlation Coefficient, are also presented in Figure 8. These metrics are very useful for enabling the user to decide which metric to use for evaluation, depending on whether the training dataset is balanced or unbalanced. For instance, accuracy may be considered if the dataset is balanced, while the F1 score, Cohen's Kappa, and Matthew's Correlation Coefficient are more appropriate when there is a dataset imbalance.

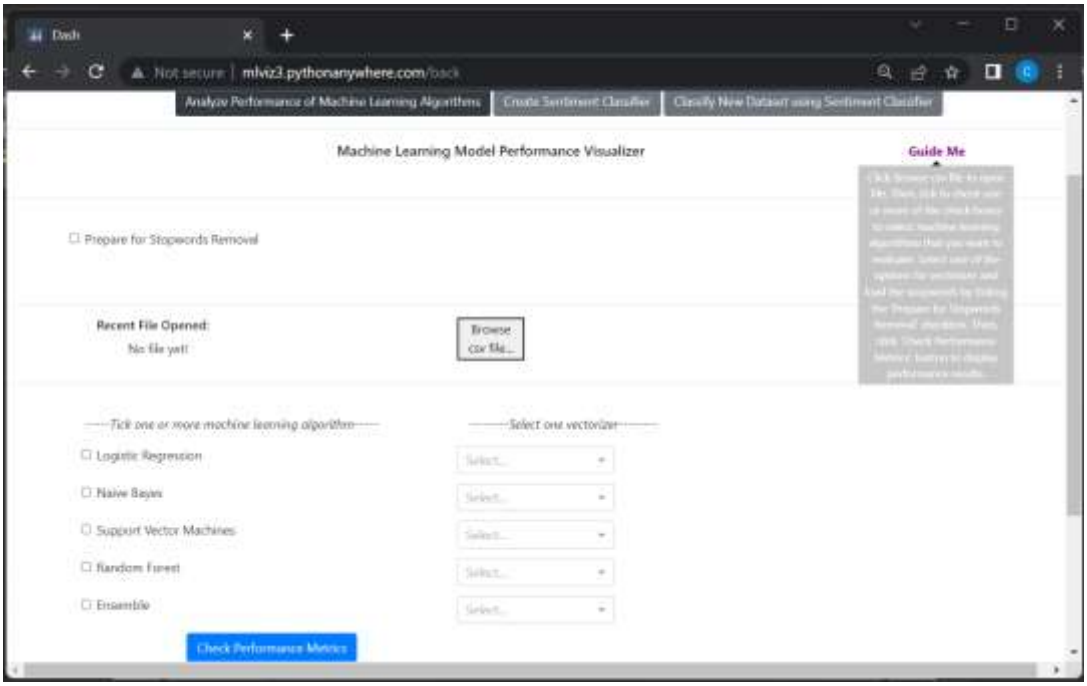

Fig. 3: Machine Learning Model Performance Visualizer

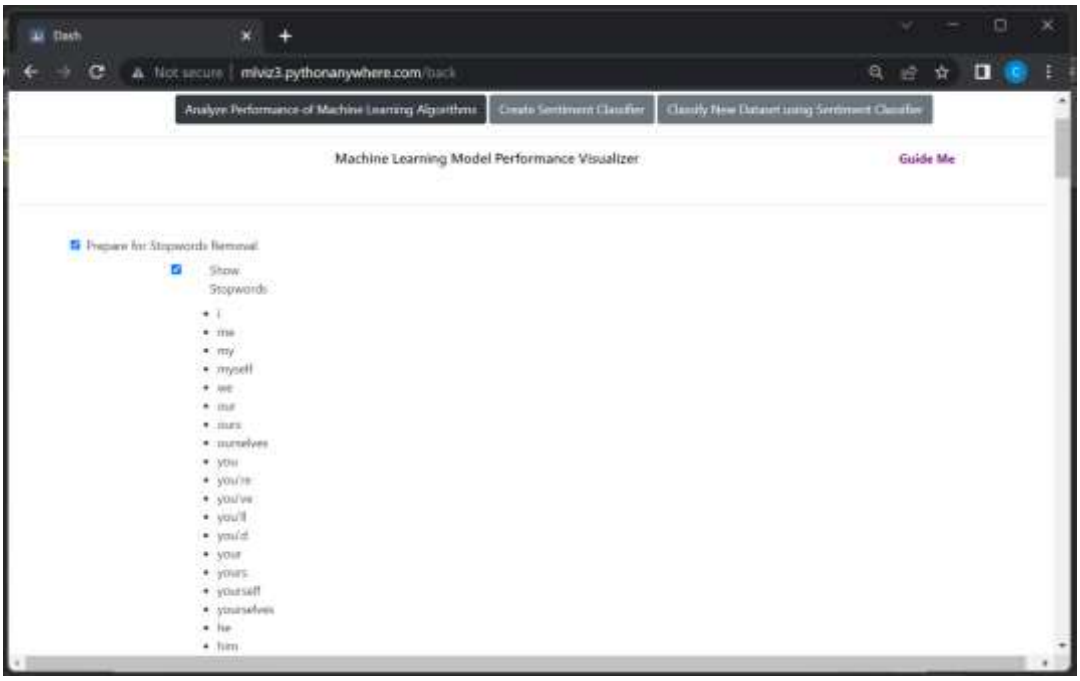

Fig. 4: Preparation of Stop Words List

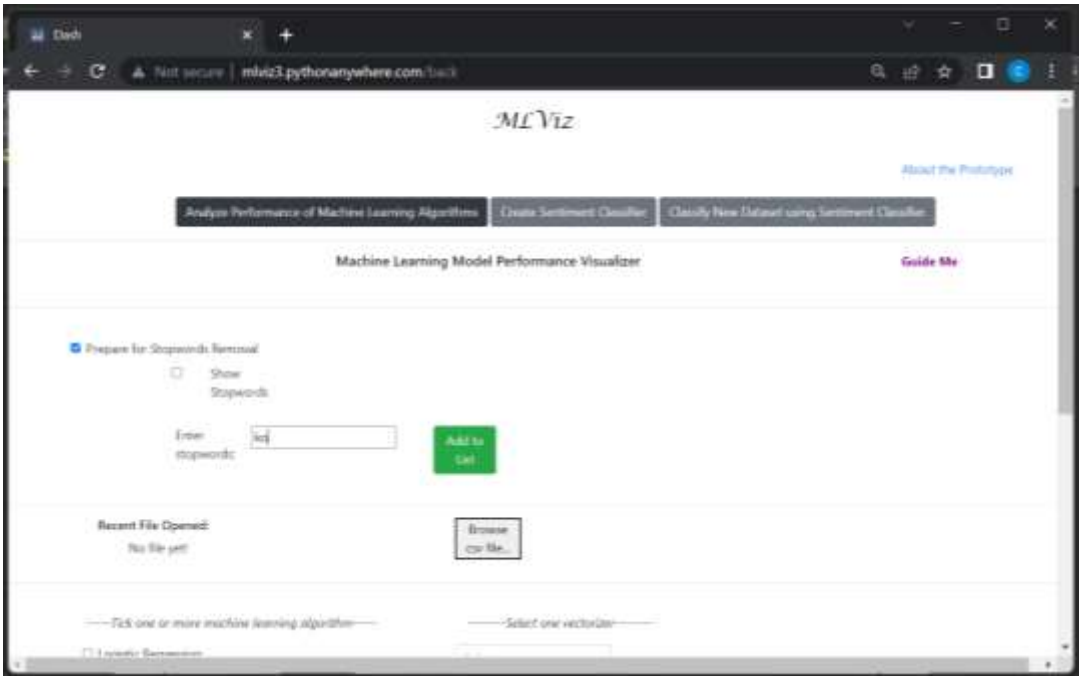

Fig. 5: Adding Stop Words to List

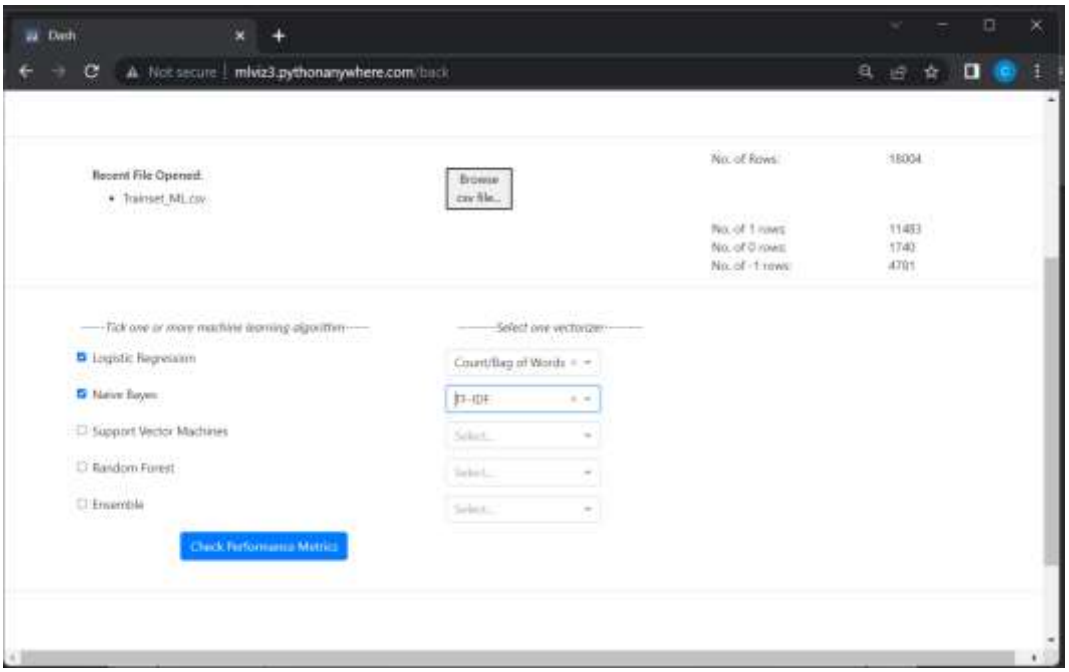

Fig. 6: Loading Training Dataset

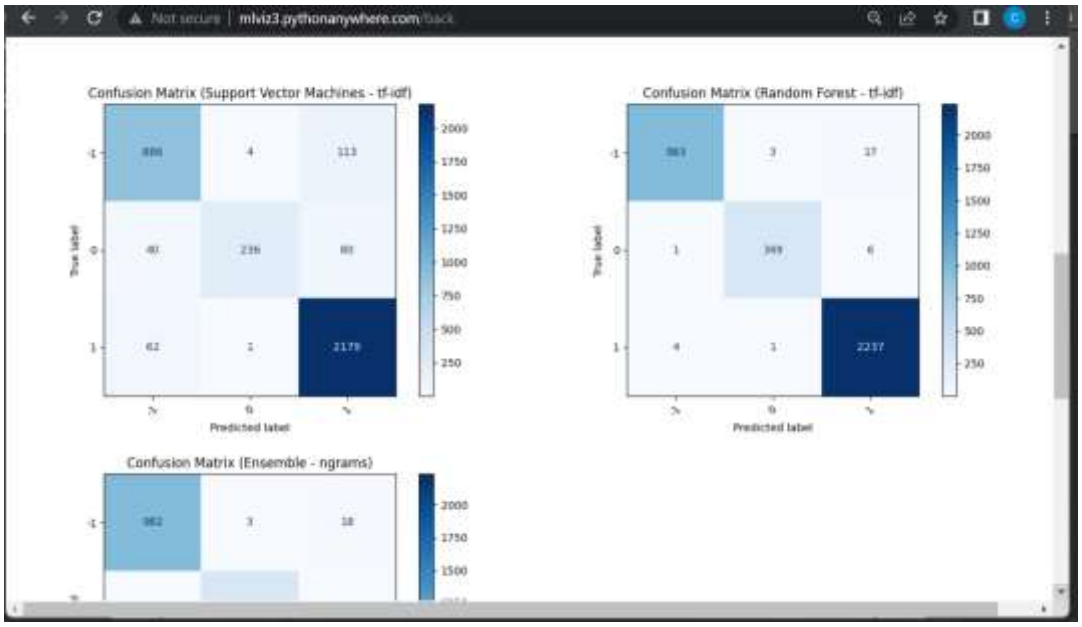

Fig. 7: Visualizing Performance of Models with Confusion Matrix

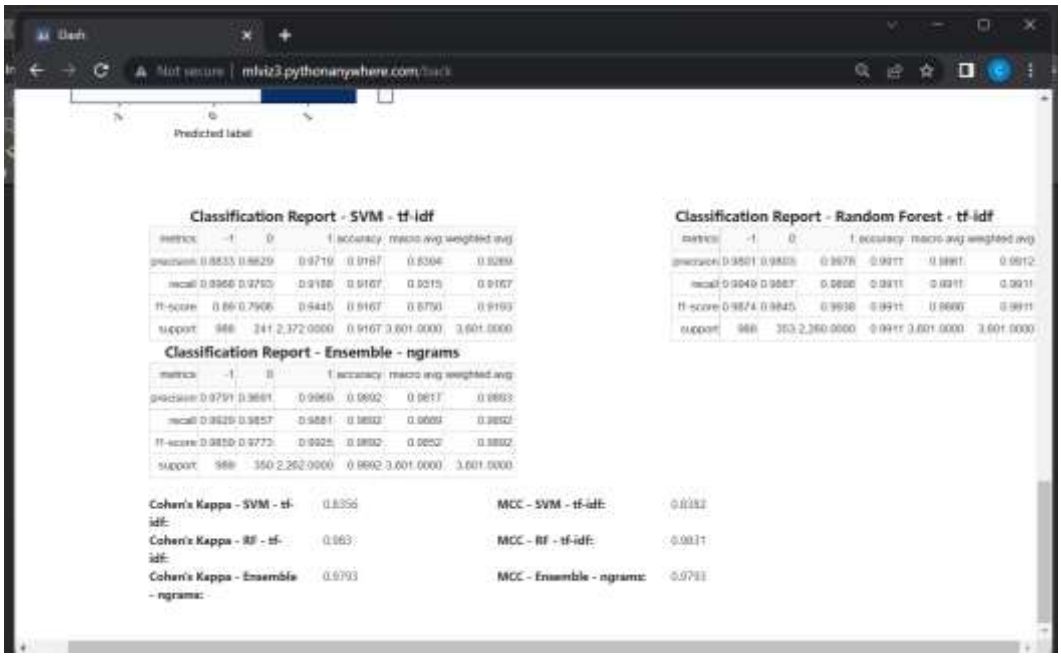

Fig. 8: Visualizing Performance of Models with Classification Report, Cohen's Kappa and Matthew's Correlation Coefficient

3.2.3 Create Sentiment Classifier

Sentiment Classifier Generator – Download Stop Words List

The Sentiment Classifier Generator shown in Figure 9 allows users to create and download the stop words list, vectorizer, and classifier. The stop words list can be downloaded through the "Download Stop Words List" button. Additionally, users can add words to the list before downloading.

Download Vectorizer and Download Classifier

The Sentiment Classifier Generator also allows the creating and downloading the vectorizer and classifier. Once downloaded, these can be retrieved and loaded in the tool whenever needed. Figure 10 shows the downloaded vectorizer and classifier.

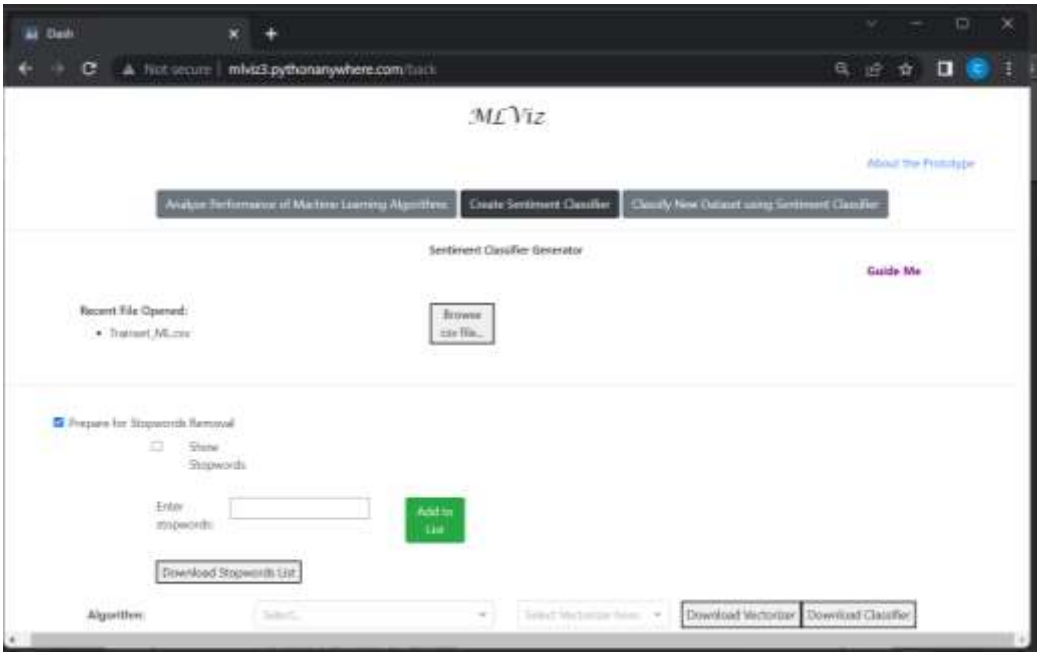

Fig. 9: Sentiment Classifier Generator – Download Stop Words List

Fig. 10: Download Vectorizer and Download Classifier

3.2.4 Classify New Dataset using Sentiment Classifier

Sentiment Classifier Visualizer

In the Sentiment Classifier Visualizer shown in Figure 11, sentiment classifier, stop words list, vectorizer and new dataset to analyze should be loaded.

Word Cloud of Most Frequent Phrases

The top dropdown and the number of words in the phrases dropdown can be specified in Figure 12. When not supplied, default settings are used. The total number of data and predictions for each class are identified. The results of the analysis are displayed in the form of a word cloud of frequent phrases.

Percentages for Each Classification

A pie chart of percentages of textual data predicted as positive, negative, and neutral is shown in Figure 13.

Bar Graph and Table of Most Frequent Ngrams

A bar graph and table of frequent Ngrams in each of the classes are presented in Figure 14. When the 'View Sentiment Classification of Comments' link is selected, it will display what is shown in Figure 15.

Visualize the classification of sentences

Figure 15 shows the complete list of data with highlights. The colors represent the classifications. For instance, green indicates positively classified sentences, pink denotes negatively classified ones, and white indicates sentences classified as neutral.

Visualize Sentiment Scores of Sentences

Figure 16 shows sentiment scores calculated using VADER in Python. An overall average sentiment rating is generated based on the compound sentiment polarity score of each of the sentences.

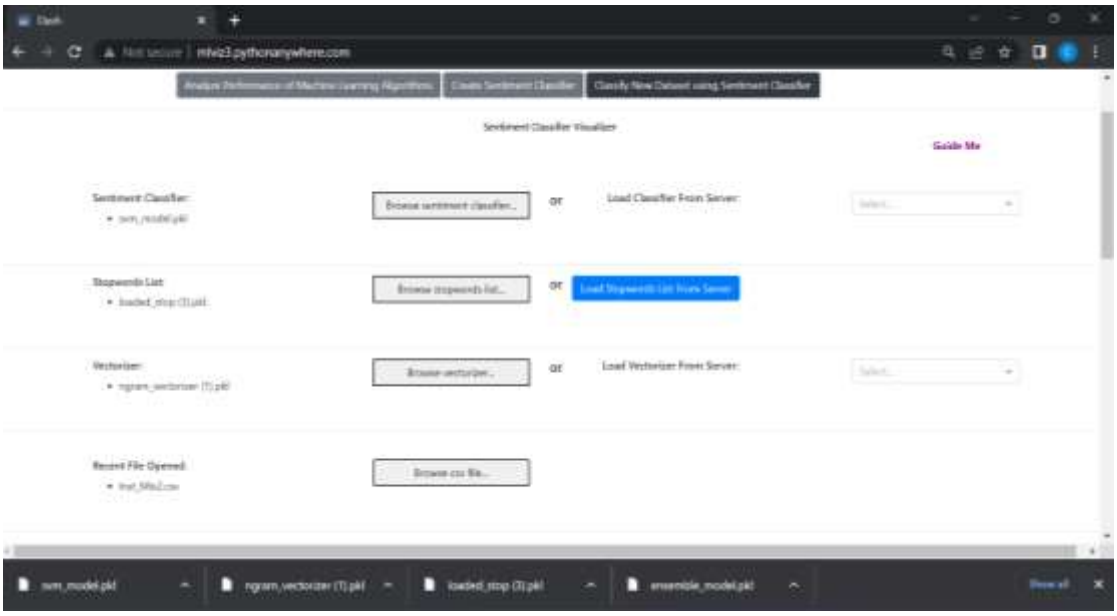

Fig. 11: Sentiment Classifier Visualizer

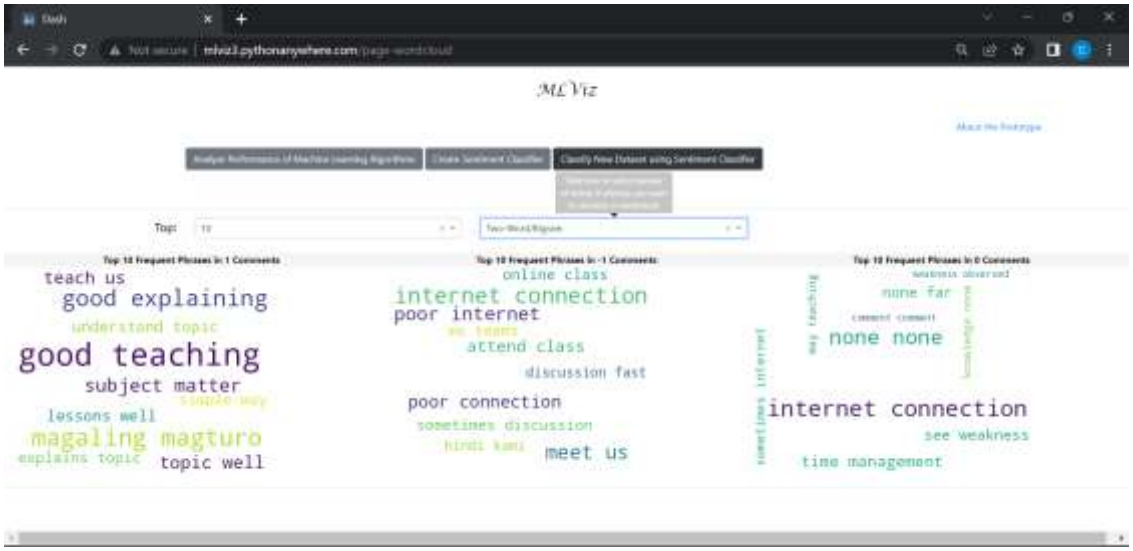

Fig. 12: Word Cloud of Most Frequent Phrases

Fig. 13: Percentages for Each Classification

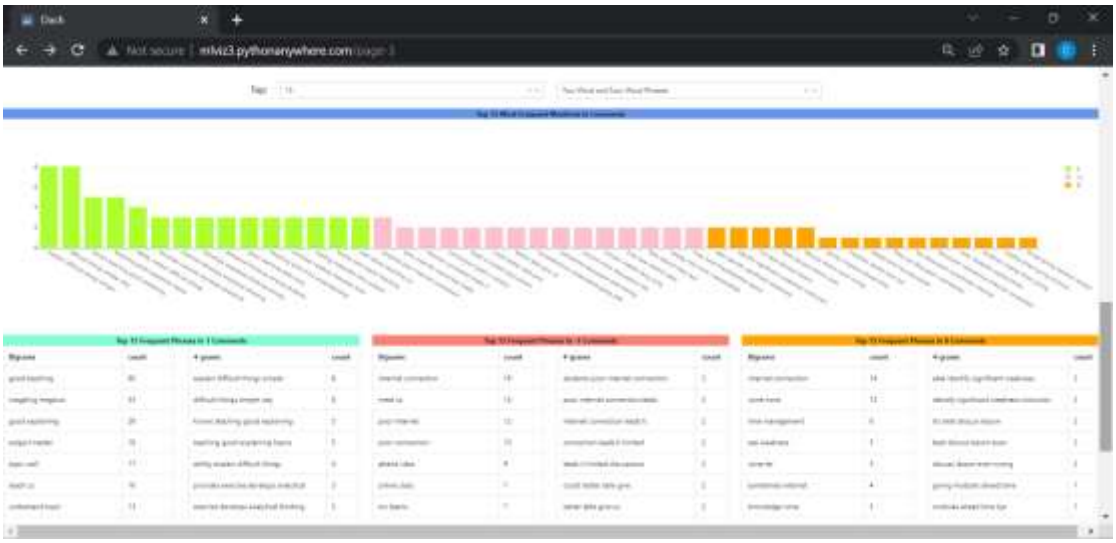

Fig. 14: Bar Graph and Table of Most Frequent Ngrams

Fig. 15: Visualize Classification of Sentences

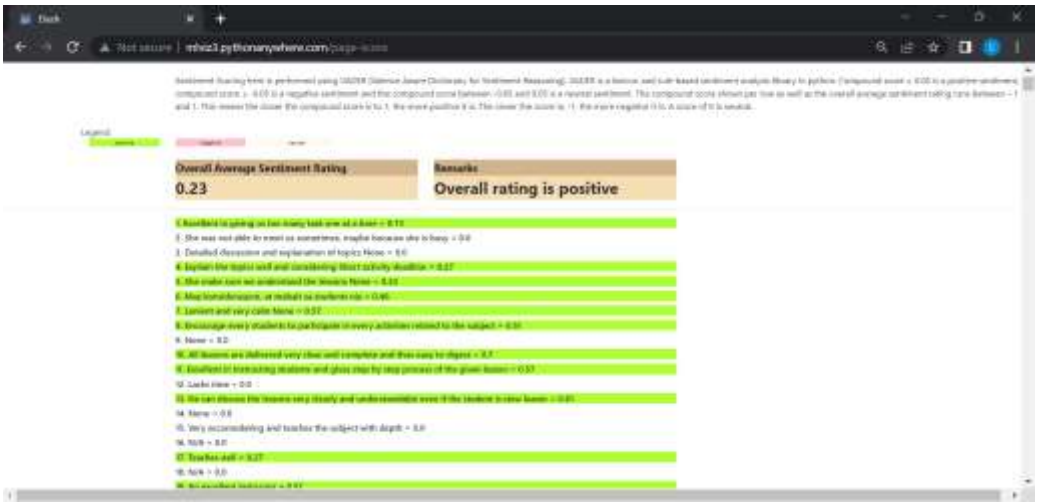

Fig. 16: Visualize Sentiment Scores of Sentences

3.3 Results of Evaluation of the Prototype (based on Features for Machine Learning Application in Sentiment Analysis)

The prototype was evaluated by four faculty researchers who are experienced in conducting sentiment analysis. Table 5 presents the summary of results of the evaluation.

Table 5. Prototype Evaluation Results

Features	Mean Rating	Rounded Off Rating	Descriptive Equivalent
1. The system assists in handling sentiment analysis training and test datasets.	4.75	5	Strongly agree
2. The system provides pre-processing and vectorization techniques.	4.75	5	Strongly agree
3. The system provides options for machine learning algorithms.	4.75	5	Strongly agree
4. The system provides visualization of the model's performance based on the available training dataset.	4.75	5	Strongly agree
5. The system allows users customize/add to stop words list and download it for future use.	4.75	5	Strongly agree
6. The system enables user to generate and download text vectorizer.	4.75	5	Strongly agree
7. The system enables users to generate and download sentiment classifier.	4.75	5	Strongly agree
8. The system enables users utilize the generated sentiment classifier on the new dataset.	4.75	5	Strongly agree
9. The system provides visualization of results after analyzing new data using the generated sentiment classifier.	4.75	5	Strongly agree
10. The system provides quantitative results of analysis through sentiment scoring.	4.75	5	Strongly agree

The prototype yielded a 4.75 mean rating in all features which is equivalent to 5 when rounded off and has a descriptive equivalent of Strongly agree.

This indicates positive feedback on the prototype. The respondents were asked to give comments and suggestions, and these are summarized in Table 6.

Table 6. Comments and Suggestions

Features	**Comments and Suggestions**
1. Handling training and test datasets	The system is quite flexible. The choice of using CSV as format of required data will enable minimization of memory costs.
	Very nice po :)
	The system handled the dataset efficiently.
2. Pre-processing and vectorization techniques	I would like to know if the system will automatically apply other data preprocessing methods such as tokenization, transform cases, etc. that's why it is not included in the options or not.
	The techniques are well utilized and implemented.
3. Options on machine learning algorithms	So far, the choices are great. Probably, just add a button for "mere generation" of the classifier and vectorizer, instead of having an automatic download button.
	With regards to the Support Vector Machine, it is well-known that it is primarily applied for binary classifications problems. In the sample video, it is shown that it is applied for a multiclass classification data set. I would like to know if the system will automatically shift from binary classification SVM if the loaded data has a binary classifier and will automatically shift to multiclass classification SVM if the data has a multiclass classifier.
	The algorithms used are the most efficient algorithms for sentiment analysis in terms of accuracy.
4. Visualization of the models' performance	It is great.
	Very nice po :)
	it is good to include a visualization of the model's performance in the prototype.
5. Customizing/adding to stop words list	If possible, can we have an option not only to add stopwords but also to delete stopwords from the default list.
	Is it possible to remove stop word(s) from the list since there are instances where some words in the pre-identified stop words list are deemed beneficial for a particular study?
	Customizing/adding to stop words list function is considered a good feature in sentiment analysis.
6. Generating and downloading text vectorizer	Probably, just add a button for "mere generation" of the vectorizer, instead of having an automatic download button.
	Very nice po :)
	Generating and downloading a sentiment classifier was a good idea for ease of use of the prototype.
7. Generating and downloading sentiment classifier	Probably, just add a button for "mere generation" of the classifier and vectorizer, instead of having an automatic download button.
	Very nice po :)
	Generating and downloading a sentiment classifier was a good idea for ease of use of the prototype.
8. Utilizing the generated sentiment classifier on a new dataset	Very nice po :)
	Utilizing the generated sentiment classifier on a new dataset was expected in the output of the prototype.
9. Visualization of results after analyzing new data using the generated sentiment classifier:	Very nice po :)
	The visualization of results after analyzing new data using the generated sentiment classifier was presented properly.
10. Sentiment scoring	Very nice po :)
	The sentiment scoring was presented properly. You may include an interpretation of the scores generated for the sentiment analysis.
Other Comments and Recommendations	Overall, the system is very promising because this is useful for students and those who want to explore machine learning sentiment classifiers without doing the hardcode. Very beneficial for research that requires text analytics.

The comments were positive in terms of handling training and test datasets. Meanwhile, a respondent indicated curiosity on whether pre-processing methods were automatically performed as this was not indicated in the prototype. Thus, the default pre-processing methods should be cited in the user guide and contemplate on possible inclusion of options on pre-processing methods. A 'generate' button for vectorizer and classifier can be added instead of having an automatic download button as this was a suggestion. With the options on machine learning algorithms, comments of a respondent indicate that customization may be included to add more flexibility to the algorithms when applied to different classification problems.

In the feature for customizing/adding to the stop words list, a common recommendation was to include options to delete stopwords from the default list. Comments on the visualization of results were positive. Meanwhile, an interpretation of the scores was suggested on the sentiment scoring feature. An overall comment on the prototype is that it is very promising and very beneficial for research that requires text analytics.
Figure 17 shows the word cloud of common phrases found in the positive comments during the evaluation. Meanwhile, Figure 18 has the common phrases found in the suggestions provided by the respondents.

Fig. 17: Common Phrases in Positive Comments

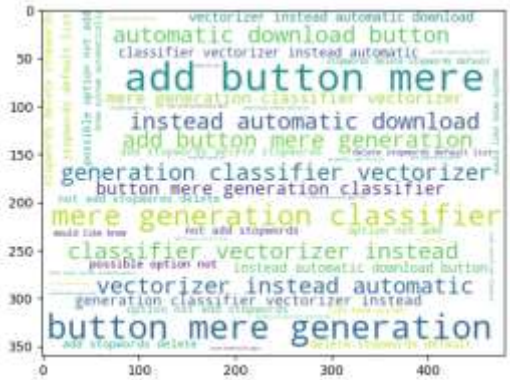

Fig. 18: Common Phrases in the Suggestions

4 Conclusion

Creating a tool that can be used to easily train, generate, evaluate, and compare algorithms applied with machine learning techniques is challenging yet possible through the availability of languages for machine learning and data analytics such as Python. The Python frameworks and libraries offer a reliable environment that reduces software development time. Python-based frameworks such as Dash Plotly provide data visualization that is interactive, and these are ideal for sentiment analysis. Pie charts, bar graphs and word cloud supported with tables and text highlights are comprehensible techniques to provide visualization of sentiment analysis results. Moreover, sentiment scores can assist users in gaining a deeper understanding of respondents' emotions conveyed through their textual feedback.

The evaluation results of the prototype suggest that features should be added to allow users to fine-tune the algorithms and find the optimal combination of configuration values that yield the best performance. Additionally, improving flexibility in customizing stop words, pre-processing, vectorization, and the generation of vectorizer and classifier should also be considered. In future work, the comments and recommendations from evaluators will be considered to improve the system. The author may also explore avenues for optimizing the tool through enhancing the efficiency of machine learning algorithms used and improving the usability and user interface of the tool. Advanced visualization techniques may be integrated to enhance the interpretability and insights provided by the sentiment analysis tool.

More respondents experienced in machine learning and sentiment analysis should be sought and involved in the evaluation to provide additional input for improving the prototype.

References:

[1] Madhuri, D. K. (2019). A machine learning based framework for sentiment classification: Indian railways case study. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, vol. 8, Issue-4, February 2019.

[2] Chaudhari, M. and Govilkar S. (2015). A Survey of Machine Learning Techniques for Sentiment Classification. *International Journal on Computational Science & Applications (IJCSA),* vol.5, No.3, June 2015; DOI:10.5121/ijcsa.2015.5302.

[3] Singh, K.N., Devi, S.D., Devi, H. M., Mahanta, A. K. (2022). A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *International Journal of Information Management Data Insights,* Vol. 2, 2022; https://doi.org/10.1016/j.jjimei.2022.100061.

[4] Rani, S., Gill, N. S., and Gulia, P. (2021). Survey of Tools and Techniques for Sentiment Analysis of Social Networking Data. *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 12, No. 4, 2021. DOI: 10.14569/IJACSA.2021.0120430.

[5] TrustRadius. *RapidMiner*, [Online]. https://www.trustradius.com/products/rapidminer/reviews?qs=pros-and-cons#product-demos (Accessed Date: January 12, 2024).

[6] Mamta and Kumar, E. (2019). A Real-Time Twitter Sentiment Analysis and Visualization System: TwiSent. *International Research Journal of Engineering and Technology (IRJET)*, Vol. 6 Issue: 06, June 2019.

[7] Mahadzir, N. H., Omar, M. F. and Nawi, M. N. M. (2018). A Sentiment Analysis Visualization System for the Property Industry. *International Journal of Technology* (2018) 8: 1609-1617.

[8] Franco, R.Y.S., Lima, R.S.A.D., Paixão, R.M., Santos, C.G.R. and Meiguins, B. S. (2019). UXmood—A Sentiment Analysis and Information Visualization Tool to Support the Evaluation of Usability and User Experience. *Information 2019*, 10, 366; https://doi.org/10.3390/info10120366.

[9] Latiff, M.N.M.A, Saad, A. F. and Yani, A. (2023). Data Visualization Based on Sentiment Analysis to Identify the Quality of Internet Service Providers in Malaysia. *International Journal of Recent Technology and Applied Science*. Vol. 5, 2023; https://doi.org/10.36079/lamintang.ijortas-0502.572.

[10] Khan, M. F. F., Seki, S. and Sakamura, K. (2023). Visualization of Online Product Reviews Written in Japanese Based on Entity Sentiment Analysis for Enhanced Customer Experience. *Int. J. Advance Soft Compu. Appl*, Vol. 15, No. 1, March 2023. DOI: 10.15849/IJASCA.230320.01.

[11] Jain R, Kumar A, Nayyar A, Dewan K, Garg R, Raman S, and Ganguly S. (2023). Explaining sentiment analysis results on social media texts through visualization. *Multimed Tools Appl* 82, 22613–22629 (2023), https://doi.org/10.1007/s11042-023-14432-y.

[12] Lavanya, A., Waqas Ali, Dr. Jaime Lloret, Vidya Sagar, S. D., Chivukula Bharadwaj (2022). A Real-time Visualization of Global Sentiment Analysis on Declaration of Pandemic. *International Journal of Computer Engineering in Research Trends*. Vol. 9, 2022, https://doi.org/10.22362/ijcert/2022/v9/i06/v9i0602.

[13] Karuna, G., Anvesh, P., Singh, C. S., Reddy, K. R., Shah, P. K. and Shankar, S. S. (2023). Feasible Sentiment Analysis of Real Time Twitter Data. *E3S Web of Conferences*, Vol. 430, 2023. *15th International Conference on Materials Processing and Characterization (ICMPC 2023)*, Newcastle, England, September 5-8, 2023, https://doi.org/10.1051/e3sconf/202343001045.

[14] Pacol, C.A. and Palaoag, T.D. (2021). Bilingual Lexicon Approach to English-Filipino Sentiment Analysis of Teaching Performance. *IOP Conference Series: Materials Science and Engineering,* Vol. 1077, *The 5th International Conference on Information Technology and Digital Applications (ICITDA 2020)* 13th-14th November 2020, Yogyakarta, Indonesia, DOI: 10.1088/1757-899X/1077/1/012044.

[15] Pacol, C. (2024). Sentiment Analysis of Students' Feedback on Faculty Online Teaching Performance using Machine Learning Techniques. *WSEAS Transactions on Information Science and Applications*. 2024, https://doi.org/10.37394/23209.2024.21.7

[16] Liu, Y., Wang, Y. and Zhang, J. (2012). New Machine Learning Algorithm: Random Forest. In: Liu, B., Ma, M., Chang, J. (eds) *Information Computing and Applications*. ICICA 2012. Lecture Notes in Computer Science, vol 7473. Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-34062-8_32.

[17] Chakraborty, K., Bhattacharyya, S., Bag, R., and Hassanien, A.A. (2019). Sentiment analysis on a set of movie reviews using deep learning techniques. *Social Network Analytics. Computational Research Methods and Techniques*, pp.127-147, https://doi.org/10.1016/B978-0-12-815458-8.00007-4.

[18] Shung, K. P. (2018). Accuracy, precision, recall or f1?, [Online]. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 (Accessed Date: January 12, 2024).

[19] scikit-learn. *sklearn.metrics.cohen_kappa_score*, [Online]. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html (Accessed Date: July 20, 2021).

[20] scikit-learn. *sklearn.metrics.matthews_corrcoef*, [Online]. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html (Accessed Date: July 20, 2021).

[21] Al-Shabi, M. A. (2020). Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining. *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 20, No.1, January 2020

[22] Yesseyeva, E., Yesseyev, K. and Abdulrazaq, M.M., Lashkari, A. H., Sadeghi, M. (2014). Tri-Pass: A new graphical user authentication scheme. *International Journal of Circuits, Systems and Signal Processing*, Vol. 8, 2014

[23] Lestari, B., Rifiani, P. I., and Gati, A. B. (2021). The Use of the Usability Scale System as an Evaluation of the Kampung Heritage Kajoetangan Guide Ebook Application. *European Journal of Business and Management Research.* Vol. 6, 2021, http://dx.doi.org/10.24018/ejbmr.2021.6.6.1113.

Contribution of Individual Authors to the Creation of a Scientific Article

The sole author of this scientific article independently conducted and prepared the entire work from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This study was funded by the University Research Council of the Pangasinan State University.

Conflict of Interest

The authors have no conflicts of interest to declare.

Table 1. Prototype Evaluation Survey Questionnaire

Prototype Evaluation Survey						
Your cooperation is highly appreciated, and your feedback will help the researcher to evaluate and improve the system. Please assess the proposed prototype system according to the features for machine learning applications in sentiment analysis. After watching the walkthrough and exploring the prototype, express your opinion by rating the system on a scale of 1-5 based on how much you agree with the statement you are reading. A score of 5 means you strongly agree, and 1 means you strongly disagree.						
SECTION I: General Information Name: _____						
SECTION II: Evaluation						
No	Feature	Rating				
		Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
1	The system assists in handling sentiment analysis training and test datasets.					
Comments and suggestions on handling training and test datasets:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
2	The system provides pre-processing and vectorization techniques.					
Comments and suggestions on pre-processing and vectorization techniques:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
3	The system provides options for machine learning algorithms.					
Comments and suggestions on options for machine learning algorithms:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
4	The system provides visualization of the model's performance based on available training datasets.					
Comments and suggestions on visualization of the models' performance:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
5	The system allows users to customize/add to stop words list and download it for future use.					
Comments and suggestions on customizing/adding to stop words list:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
6	The system enables users to generate and download text vectorizers.					
Comments and suggestions on generating and downloading text vectorizer:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
7	The system enables users to generate and download sentiment classifiers.					
Comments and suggestions on generating and downloading sentiment classifier:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
8	The system enables users to utilize the generated sentiment classifier on the new dataset.					
Comments and suggestions on utilizing the generated sentiment classifier on the new dataset:						
No	Feature	Strongly disagree (1)	Somewhat disagree (2)	Neutral (3)	Somewhat agree (4)	Strongly agree (5)
9	The system provides visualization of results after analyzing new data using the generated sentiment classifier.					
Comments and suggestions on visualization of results after analyzing new data using the generated sentiment classifier:						
		Strongly	Somewhat	Neutral	Somewhat	Strongly

No	Feature	disagree (1)	disagree (2)	(3)	agree (4)	agree (5)
10	The system provides quantitative results of analysis through sentiment scoring.					
Comments and suggestions on sentiment scoring:						
Additional comments and suggestions:						