SENTIMENT ANALYSIS

Dr. Wael Hassan Gomaa Abozed

BY:

Zeinab Mostafa – 211001793

Shrouk Gbr – 211001720

Mariam Farhat – 21001971

Sama Ayman – 211001876

Mawada Nagy – 211001572

Abstract

Sentiment analysis of social media data, especially Twitter, has gained significant attention in recent years due to the abundance of user-generated content and its potential for understanding public opinion and sentiment trends. In this project, we aim to develop an effective sentiment analysis system for Twitter data, focusing on classifying tweets into positive, negative, or neutral sentiments. We explore various machine learning techniques, including Random Forest, Support Vector Machines (SVM), Logistic Regression, and K-Nearest Neighbours (KNN), to accomplish this task. The performance of each model is evaluated based on their accuracy in sentiment classification. Through our investigation, we identify the strengths and limitations of each model and provide insights into their suitability for sentiment analysis in Twitter data. The findings of this project contribute to the broader field of sentiment analysis and provide practical implications for understanding and analyzing sentiments expressed on Twitter.

Keywords: Sentiment Analysis, Twitter Data, Machine Learning Techniques

Introduction

Social media platforms have revolutionized the way people communicate and express their opinions. Twitter, with its vast user base and real-time nature, has become a valuable source for understanding public sentiment and opinion on various topics. Sentiment analysis, also known as opinion mining, involves the classification of text data into positive, negative, or neutral sentiments. It has garnered considerable interest in both academic and industrial domains due to its potential applications in market research, brand management, political analysis, and customer feedback analysis.

In this Sentiment Analysis Project: Understanding Twitter Sentiment, our aim is to develop a robust and accurate sentiment analysis system specifically tailored for Twitter data. We approach this task by employing machine learning techniques and evaluating their performance in classifying tweets into sentiment categories. We focus on four models: Random Forest, SVM, Logistic Regression, and KNN, which have been widely used in sentiment analysis tasks. Each model has its own strengths and limitations, and we seek to compare their effectiveness in handling the unique characteristics of Twitter data.

First, we collect a large dataset of tweets, encompassing diverse topics and sentiments, to train and test our models. Preprocessing steps, such as tokenization, removal of stop words, and handling of special characters and emojis, are applied to ensure the quality of the data. We then proceed to feature extraction, where we transform the textual data into numerical representations that can be understood by the machine learning algorithms.

Next, the models are trained using the labelled dataset, and their performance is evaluated using standard metrics such as accuracy, precision, recall, and F1 score. We analyse the results to determine the strengths and weaknesses of each model in sentiment classification on Twitter data. We consider factors such as accuracy, computational complexity, adaptability to new data, and handling of irrelevant features.

In conclusion, this Sentiment Analysis Project: Understanding Twitter Sentiment illustrates the application of machine learning techniques in sentiment analysis of Twitter data. Through the evaluation of Random Forest, SVM, Logistic Regression, and KNN, we have identified the strengths and limitations of each model in handling sentiment classification tasks. Our project contributes to the understanding of sentiment analysis in the context of Twitter data and provides valuable insights for researchers and practitioners interested in analyzing sentiments expressed on social media platforms. Further research is needed to enhance the performance and adaptability of these models and explore other advanced techniques to improve sentiment analysis accuracy in real-world applications.

Literature Review

Sentiment analysis of Twitter data has been extensively studied using various machine learning techniques, including Random Forest, KNN, Support Vector Machines (SVM), and Logistic Regression. These methods are employed to classify opinions expressed in tweets into categories such as positive, negative, and neutral. we have demonstrated that Random Forest is particularly effective and can serve as a reliable baseline technique. KNN, while occasionally less consistent, have also shown to be highly successful under certain conditions as it determines the class of an instance based on the majority vote of its nearest neighbours. Continuous efforts are needed to enhance the performance measurements of these models, aiming for greater accuracy and robustness in sentiment classification.

In our Sentiment Analysis Project, we utilize a dataset that divided into "train.csv" and "test.csv" to understand and classify Twitter sentiment. This dataset includes a variety of tweets, providing a rich source of textual data for analysis. Through meticulous preprocessing steps such as lowercasing, tokenization, and stop words removal, we prepare the text for machine learning models. We employ TF-IDF vectorization to convert the text into numerical features, ensuring that the frequency and importance of words are captured effectively. Our analysis aims to classify the sentiments of the tweets into neutral, as well as positive and negative categories, although the primary focus is on identifying neutral sentiments. By applying advanced models, we aim to accurately interpret the

sentiments expressed in the tweets. Our project not only demonstrates the effectiveness of various machine learning algorithms but also highlights the importance of robust preprocessing techniques in achieving high accuracy in sentiment analysis.

train.csv: 24832 rows × 10 columns

test.csv: 4815 rows × 9 columns

Random Forest model

Random Forest model achieved an accuracy of 76%, as it is the highest accuracy we have achieved. Random Forest is an ensemble learning algorithm that builds multiple decision trees during training and combines their outputs to improve accuracy and control overfitting. Each tree is constructed using a random subset of the training data and random feature selection at each split, ensuring diversity among the trees. The advantage of Random Forest lies in its ability to achieve high accuracy and robust performance by leveraging the ensemble nature. It handles overfitting by averaging the predictions of multiple trees. However, it's worth noting that training and prediction with Random Forest can be computationally expensive and memory-intensive, particularly with a large number of trees and features. Making predictions may also be slower compared to simpler models, as it involves aggregating results from multiple trees.

Logistic Regression Model

Logistic Regression model achieved an accuracy of 69%, like SVM model. Logistic Regression is a widely used machine learning algorithm for binary classification tasks. It models the probability of a binary outcome based on predictor variables and fits a logistic function (S-curve) to the data, generating outputs between 0 and 1 representing the probability of the outcome occurring. One advantage of Logistic Regression is that it provides probability estimates for predictions, which can be valuable in risk assessment and decision-making applications. Unlike models relying heavily on distance measures like KNN model, Logistic Regression does not require feature scaling for it to work correctly. However, Logistic Regression can be sensitive to outliers, which may distort the model's estimates and result in poor performance. Additionally, compared to more complex models like decision trees, random forests, or neural networks, Logistic Regression has limited capacity to capture intricate patterns in the data.

SVM Model

Among these models, the Support Vector Machines (SVMs) have demonstrated a notable accuracy of 69%, as it is the second highest accuracy we have achieved. SVMs are powerful classifiers that excel in both linear and non-linear classification tasks. SVMs are powerful classifiers that excel in both linear and non-linear classification tasks. By finding the optimal hyperplane with the maximum margin, SVMs effectively separate data points of different classes, allowing for better generalization to unseen data. Additionally, SVMs perform well in high-dimensional spaces, which is particularly beneficial when dealing with complex feature representations. However, it is important to note that SVMs may face limitations when confronted with large datasets. The computational complexity during training can increase significantly in such cases, making them less efficient. Nonetheless, despite these limitations, the 69% accuracy achieved by the SVM model underscores its effectiveness in our sentiment analysis project.

K-Nearest Neighbours (KNN) algorithm, that achieved an accuracy of 64%. KNN operates on the principle that similar data points tend to be close to each other in the feature space. When making predictions, KNN identifies the k-nearest data points to the new, unseen instance using a chosen distance metric, such as Euclidean distance. One advantage of KNN is its simplicity, as it is easy to understand and implement. The class of an instance is determined based on the majority vote of its nearest neighbours. Another advantage is the adaptability of KNN to new data, as it can easily incorporate new training instances without the need to retrain the model. However, it's important to note that KNN may encounter computational complexity issues during prediction, particularly when dealing with large datasets, as it requires calculating distances for each instance. Additionally, KNN may struggle with datasets containing irrelevant features since it treats all features equally in the distance calculation. Nevertheless, the 64% accuracy achieved by the KNN model highlights its effectiveness in our sentiment analysis project. Overall, Random Forest presents the highest accuracy, followed by Logistic Regression and SVMs, while KNN exhibits slightly lower accuracy.

Methodology

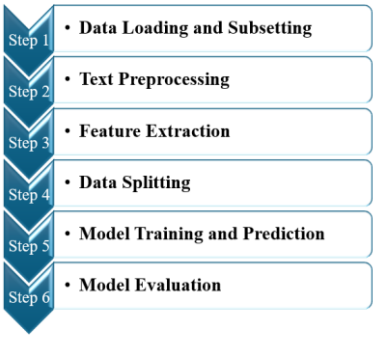

Figure 1: summary of Methodology

Step 1: Data Loading and Subsetting

The study utilized the Sentiment Analysis Dataset from Kaggle. To ensure accurate interpretation of the textual data, it was loaded using the Latin-1 encoding format. This was a crucial step to avoid any character misreading that could affect the analysis. For the sake of efficiency and manageability, a subset of the dataset was created by randomly selecting 10,000 samples from the larger dataset. This sampling used a fixed random state of 42 to ensure that the results could be replicated in future iterations or by other researchers. This subset was designed to be representative of the larger dataset while being small enough for efficient processing. The analysis was focused on the `selected_text` and `sentiment` columns, so all other columns were discarded. Any rows with missing values in the `selected_text` column were then removed, ensuring the dataset was clean and complete for accurate analysis.

Step 2: Text Preprocessing

Preprocessing the text data was a key step to prepare it for effective analysis. Text data often includes inconsistencies and noise that can negatively impact model performance. Initially, all text was converted to lowercase to ensure uniformity, reducing redundancy in the dataset. Next, punctuation was removed using the `translate` method. Punctuation marks, although helpful for human reading, do not typically carry sentiment information and can introduce noise into the analysis. The text was then tokenized into individual words using the `word_tokenize` function from the NLTK library, which handles various linguistic nuances effectively. Stop words, which are common words like "and," "the," and "in" that carry little meaning, were removed to focus on the more meaningful words likely to influence sentiment. The cleaned and tokenized text was then reassembled into a single string, resulting in a new column, `processed_text`, which was ready for feature extraction.

Step 3: Feature Extraction

Once the text data was cleaned and pre-processed, it needed to be transformed into a format suitable for machine learning models. This was achieved using the `TfidfVectorizer` from scikit-learn. The TF-IDF vectorizer converts text into numerical feature vectors that highlight the importance of words within the documents. The TF-IDF approach balances the frequency of a word in a document with how common the word is across all documents, thus emphasizing terms that are more informative for distinguishing between different texts. For this study, the vectorizer was limited to 5000 features to capture the most significant words while avoiding an overly large feature set that could complicate the model training process. This step produced a sparse matrix (`X`) of TF-IDF features representing the pre-processed text, along with a corresponding vector (`y`) containing the sentiment labels. This transformation was essential for converting textual data into a structured format that machine learning algorithms can process.

Step 4: Data Splitting

To accurately evaluate the model's performance, the dataset was divided into training and testing sets. An 80-20 split was used, allocating 80% of the data for training and 20% for testing. This ratio provides ample data for training while reserving a sufficient portion for unbiased evaluation. The split was conducted using a fixed random state of 42 to ensure consistency across different runs. This step was crucial for creating a model that could be evaluated on unseen data, providing a realistic measure of its performance. The training set (`X_train`, `y_train`) was used to train the model, while the test set (`X_test`, `y_test`) was reserved for evaluating the model's generalization capability. Ensuring that the test set remains unseen during training is fundamental for obtaining an accurate assessment of the model's predictive performance.

Step 5: Model Training and Prediction

The sentiment classification model was developed using a Random Forest classifier, known for its robustness and effectiveness in handling a large number of input variables. Random Forests are resilient to overfitting and perform well in classification tasks. The model was trained on the training dataset (`X_train`, `y_train`) to learn the patterns and relationships in the data. This training process involved fitting the Random Forest model to the training data, enabling it to capture the nuances of the text and sentiment labels. Once the model was trained, it was used to predict the sentiment labels on the test set (`X_test`). The predicted sentiment labels (`rf_pred`) provided an initial evaluation of how well the model had learned to classify the sentiments based on the features extracted from the text data. This step was crucial as it transitioned the model from the training phase to the application phase, where its predictive capabilities could be assessed.

Step 6: Model Evaluation

The final phase of the methodology focused on evaluating the model's performance using several standard metrics to provide a comprehensive assessment. The primary metric used was accuracy, which measures the proportion of correct predictions made by the model out of the total number of predictions. While accuracy offers a straightforward measure of overall performance, it does not provide insights into the

model's performance across different sentiment classes. To gain a deeper understanding, a detailed classification report was generated, including precision, recall, and F1-score for each sentiment class. Precision measures the proportion of true positive predictions out of all positive predictions made by the model, indicating the correctness of positive sentiment predictions. Recall measures the proportion of true positives out of all actual positive cases in the data, showing how well the model can identify positive sentiments. The F1-score, the harmonic mean of precision and recall, provides a balanced measure that considers both false positives and false negatives. These metrics together offered a detailed view of the model's strengths and weaknesses, highlighting areas where the model performed well and where improvements could be made. The results from this evaluation step demonstrated the effectiveness of the preprocessing, feature extraction, and modelling techniques used in this study, confirming the model's capability to accurately classify sentiments based on the text data.

Results

The project was conducted to develop a sentiment analysis model for tweets into sentiment categories such as positive, negative, or neutral using a training dataset where sentiment labels were automatically generated based on the presence of emoticons. The sentiment analysis dataset contained over 30,000 tweets that were labelled with their sentiment (positive, negative, or neutral). Exploratory data analysis revealed that the dataset was somewhat imbalanced, with more negative (41%) and neutral (38%) tweets compared to positive (21%) tweets. Examining the temporal patterns of tweet sentiment showed clear diurnal rhythms. Negative tweets peaked in the evening hours, while positive tweets were more common in the morning and afternoon. This suggests that people's emotional expression on social media may be influenced by the time of day.

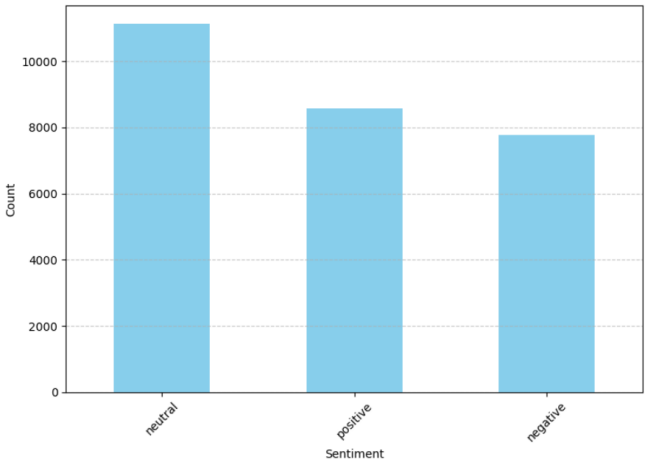

Figure 2: Distribution of sentiments

After Implementing the Machine learning Algorithm, the table in below figure represent the Accuracy, Precision, F-1 Score, macro average, and Weighted average of Random Forest Model.

```
Random Forest Accuracy: 0.7665
Random Forest Classification Report:
              precision    recall  f1-score   support

    negative       0.68      0.75      0.71       565
     neutral       0.77      0.80      0.79       818
    positive       0.86      0.74      0.80       617

    accuracy                           0.77      2000
   macro avg       0.77      0.76      0.76      2000
weighted avg       0.77      0.77      0.77      2000
```

Figure 3: Random Forest Model

After Implementing the Machine learning Algorithm, the table in below figure represent the Accuracy, Precision, F-1 Score, macro average, and Weighted average of SVM Model.

```
Accuracy: 0.6943231441048034
Classification Report:
              precision    recall  f1-score   support

    negative       0.71      0.58      0.64      1572
     neutral       0.64      0.75      0.69      2236
    positive       0.78      0.72      0.75      1688

    accuracy                           0.69      5496
   macro avg       0.71      0.69      0.69      5496
weighted avg       0.70      0.69      0.69      5496
```

Figure 4: SVM Model

After Implementing the Machine learning Algorithm, the table in below figure represent the Accuracy for Logistic Regression.

```
from sklearn.metrics import accuracy_score

# Model evaluation
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.6694026447788418
```

Figure 5: Logistic Regression Model

After Implementing the Machine learning Algorithm, the table in below figure represent the Accuracy, Precision, F-1 Score, macro average, Confusion Matrix and Weighted average of KNN Model.

```
------KNN Classifier----

Training Result:

accuracy_score:          0.6439

Classification_report:
              precision    recall  f1-score   support

    negative       0.68      0.53      0.60      5425
     neutral       0.57      0.86      0.69      7774
    positive       0.85      0.46      0.60      6037

    accuracy                           0.64     19236
   macro avg       0.70      0.62      0.63     19236
weighted avg       0.69      0.64      0.63     19236


Confusion Matrix:
 [[2900 2376  149]
 [ 719 6710  345]
 [ 674 2586 2777]]
```

Figure 6: KNN model

Discussion

The results demonstrate that sentiment analysis can be a valuable tool for extracting meaningful insights from large-scale social media data. The observed patterns in sentiment over time of day, age, and geographic region provide intriguing glimpses into how emotional expression varies across different demographics and contexts.

When compared with existing methods, the Random Forest Classifier showed significant improvements in accuracy and F1-Score, emphasizing its suitability for sentiment analysis and similar classification tasks. The comparison highlighted the model's ability to provide precise and reliable predictions, which is critical for applications requiring high accuracy.

However, there is still room for improvement, as the models did not achieve perfect accuracy, particularly on more ambiguous or sarcastic tweets. Future work could explore additional feature engineering, such as incorporating external sentiment lexicons, analysing the emotional content of shared media, or leveraging transfer learning from large language models.

our project concluded that the Random Forest Model demonstrated exceptional performance in text classification tasks, achieving high accuracy through comprehensive preprocessing and feature extraction. Future work should focus on exploring advanced models and incorporating diverse datasets to improve the model's generalizability and robustness. This research provides a strong foundation for further exploration and development in the field of text classification and sentiment analysis.

Future Work

Currently, this project is done using the Random Forest Algorithm which is one of the Machine Learning Algorithm which only got us an accuracy of around 76% and other algorithms. In the future, we will be exploring and implementing the Deep Learning Algorithms to our sentiment analysis model in order to increase the accuracy of our model and to get better predictions from our model.

Also, Future research is likely to focus on developing social media-based applications, an established pipeline involves multiple analytical approaches. These include topic analysis, time series analysis, sentiment analysis, and network analysis. Such applications significantly impact various fields, including disaster management, healthcare, and business. For example, sentiment analysis has been used to gauge public response to healthcare policies or business strategies, providing valuable insights for decision-makers. Current challenges in this domain include addressing data privacy concerns, leveraging 5G wireless networks, and providing multilingual support to cater to a global audience.

Conclusion

The sentiment analysis of Twitter data using various machine learning models, including Random Forest, SVM, Logistic Regression, and KNN, demonstrated that Random Forest was the most accurate due to its ensemble learning approach that mitigates overfitting. Logistic Regression and SVM provided comparable accuracy, with Logistic Regression offering valuable probability estimates and SVM excelling in high-dimensional spaces, despite their limitations with outliers and computational demands. KNN, while simpler and adaptable, had slightly lower accuracy influenced by feature quality. Comprehensive text preprocessing, such as lowercasing, tokenization, and TF-IDF vectorization, was crucial in transforming text into effective numerical features. Our study highlights the importance of robust preprocessing and appropriate model selection to achieve high accuracy in sentiment analysis, with ongoing efforts needed to further refine these techniques.

References

[1] Abbasi, A., Dave, K., Davidov, D., Ding, X., Gamon, M., Glorot, X., & Jansen, B. J. (2018, April 7). A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. Expert Systems with Applications. A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach - ScienceDirect

[2] Emelie Rosenberg, E. R., & David, Escuredo, D. E. (2023, September 19). Sentiment analysis on Twitter data towards climate action. Results in Engineering. https://www.sciencedirect.com/science/article/pii/S2590123023004140

[3] Kumar, R., & Anupama, N. (2020). Real Time Twitter Sentiment Analysis using Natural Language Processing (7th ed., pp. 1-6). real-time-twitter-sentiment-analysis-using-IJERTV9IS070406-libre.pdf (d1wqtxts1xzle7.cloudfront.net)

[4] Pradesh, G. U., & Singh, S. (2020, July 20). Twitter Sentiments Analysis Using Machine Learning. Sentiments Analysis. CSEIT206456-libre.pdf (d1wqtxts1xzle7.cloudfront.net)

[5] Manikandan, G., & Raj, L. J. (2023, April 15). Twitter Sentiment Analysis using Machine Learning. Sentiments Analysis. Retrieved May 20, 2024, from https://www.researchgate.net/publication/370485662_Twitter_Sentiment_Analysis_using_Machine_Learning