CS 7641 Final Report

Aishani Chakraborty, Asmita Karandikar, Niti Mirkhelkar, Sagar Gupta, Veena Gonugondla

1. Introduction and Background

Our project is about the use of sentiment analysis to classify tweets from X (formerly Twitter) into emotion categories. This is an explored topic with several datasets, classification methods, NLP packages, and literature available. Our purpose in pursuing this project is to build upon the advancements in the field that have occurred in the past decade, with the goal of obtaining more sophisticated classifications beyond degrees of positivity/negativity.

2. Problem Definition

The task is to identify and classify emotions expressed in tweets using both supervised and unsupervised learning techniques. In the supervised approach, we will predict which of six basic emotions—anger, fear, joy, love, sadness, and surprise—match the sentiment of a tweet. Clustering techniques will be used to discover hidden emotional patterns and group tweets with similar sentiments.

3. Methods

Preprocessing Methods

We preprocessed our data by removing noise (links, non alphanumeric symbols), stop words (“a”, “the”, etc.), tokenization (i.e. breaking down each post into smaller, informative pieces), stemming (reducing words to roots by removing prefixes/suffixes) [1]. We primarily used the NLTK (Natural Language Toolkit) library to implement preprocessing since it provides convenient functions for our specific project. We also removed outliers and balanced the sentiment classes to ensure robust training.

Supervised Methods

We implemented an LSTM model along with SVM. RNNs, and more specifically LSTMs, have been shown to be successful for sentiment analysis due to their ability to understand sequential information, and therefore context, in strings. LSTMs also are able to focus more on immediate context, which is crucial to understanding emotions throughout shorter texts [3]. SVMs can specifically be utilized for finding planes/lines or any other specific dimensions by determining hyperplanes. These allow for there to be a clear mathematical line drawn and referenced to later when determining which side of the boundary the tweets are classified into [3].

Unsupervised Methods

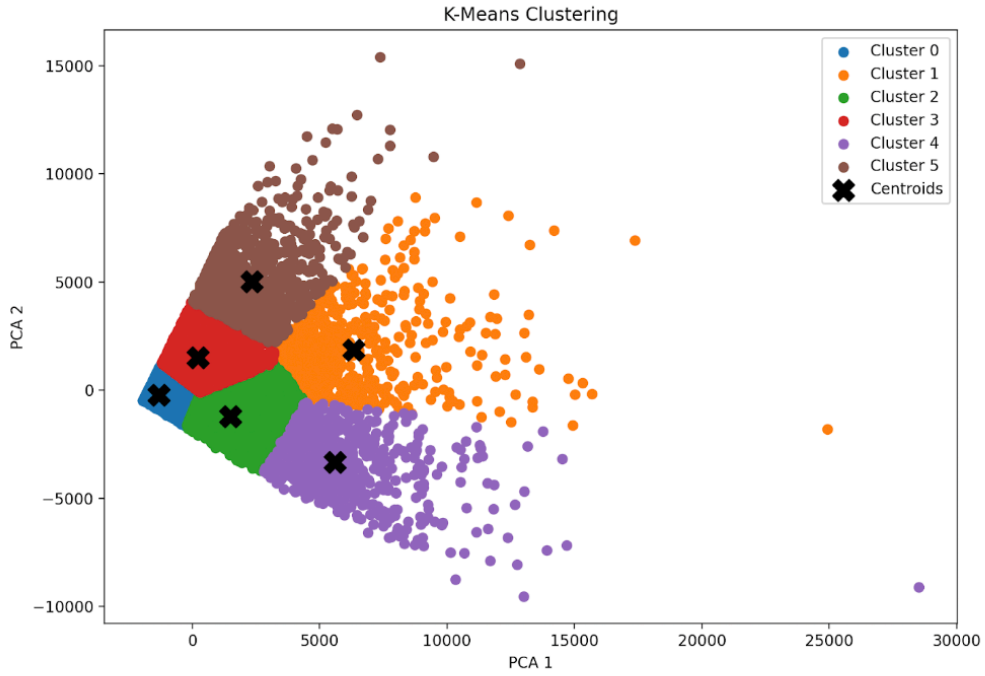
We implemented unsupervised learning by converting the tweets into embeddings using a vectorizer, and apply clustering the K-means algorithm to group similar tweets together. K-means focuses on hard clustering based on features, reducing dimensionality if necessary, and grouping the closest/most similar points together, thus revealing any patterns in the sentiment data [4].

4. Results and Discussion

Unsupervised Method

When working on the unsupervised implementation, we decided that we wanted to implement a K-means algorithm to determine if certain clusters can be extracted from the data. We chose to use K-means because it is a simple but effective algorithm to understand similarities across the datapoints. In order to implement this, we first needed to determine if it was possible to represent the model that we are working on in a dimension that can be visualized, and the way to do this was to additionally use the PCA algorithm. This algorithm allowed us to take the data and shrink it into fewer dimensions that we later ran the K-means algorithm on.

To determine how well the K Means algorithm was doing, we decided to use a Silhouette score. The Silhouette score is a metric to quantify the effectiveness of K means as it shows how well the algorithm clusters the values in an assigned to a cluster and how well these clusters are spaced far from each other. Initially, we focused on determining a score without using the PCA algorithm, but by implementing the PCA algorithm, we saw that the Silhouette score was a lot higher at 0.514 versus a 0.421, meaning that a K-means implementation was a lot more effective once PCA was implemented. Therefore we decided to continue with this method, and we created a visualization of the K-means algorithms that is pasted below.



Next Steps: Unsupervised Method

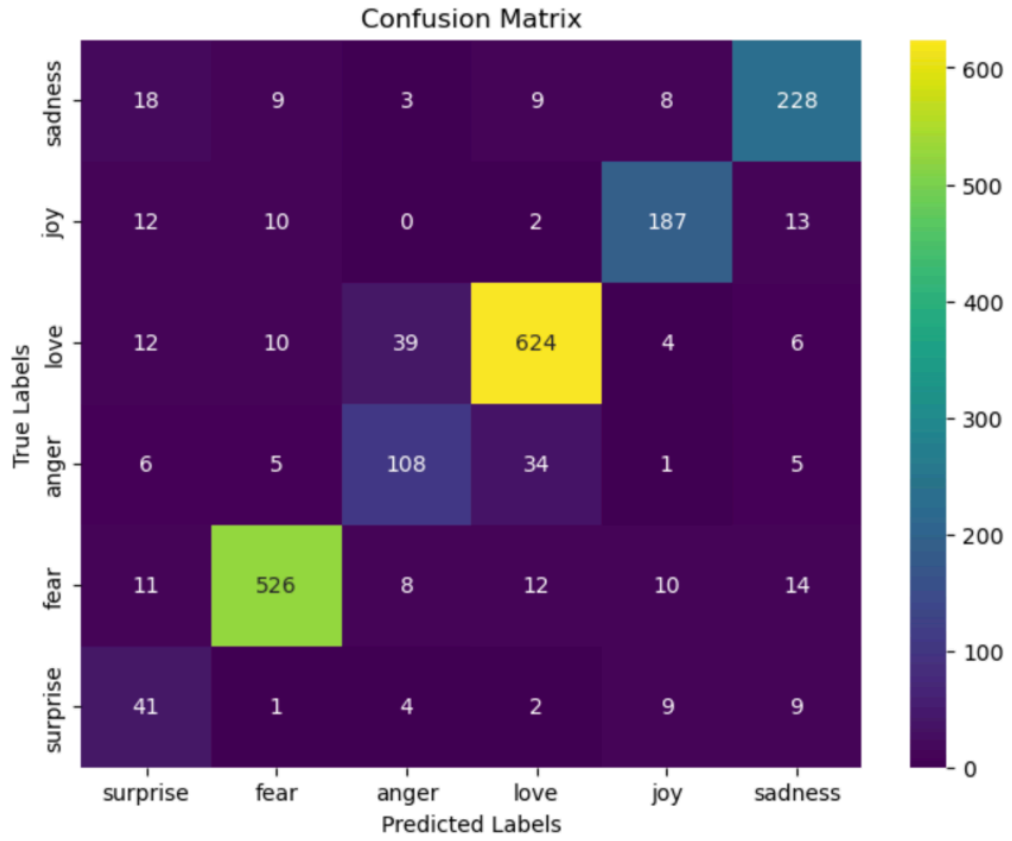
Considering this visualization and the model's silhouette score in the unsupervised method, it is evident that there are improvements that can be made to get the score closer to 1. When looking at the graph, there are no clear groups of points that belong to distinct clusters, causing the clusters to look like they are overlapping. We can make improvements to ensure that the clusters are more distinct as well.

Earlier, we made a bit of improvement by incorporating a PCA before running K-means on the algorithm, but one area that we wanted to make improvements in is the processing of the original data. Currently, we are cleaning the data using a Tokenizer which is able to consider which words are utilized and the frequency of the words, and create a sequence that is used in the K-means process. However, we can try to use other forms of processing such as Word2Vec or GloVe to improve the score by allowing scores that are the most similar to be considered closer in the vectors/sequences created by the preprocessing.

Supervised Method 1

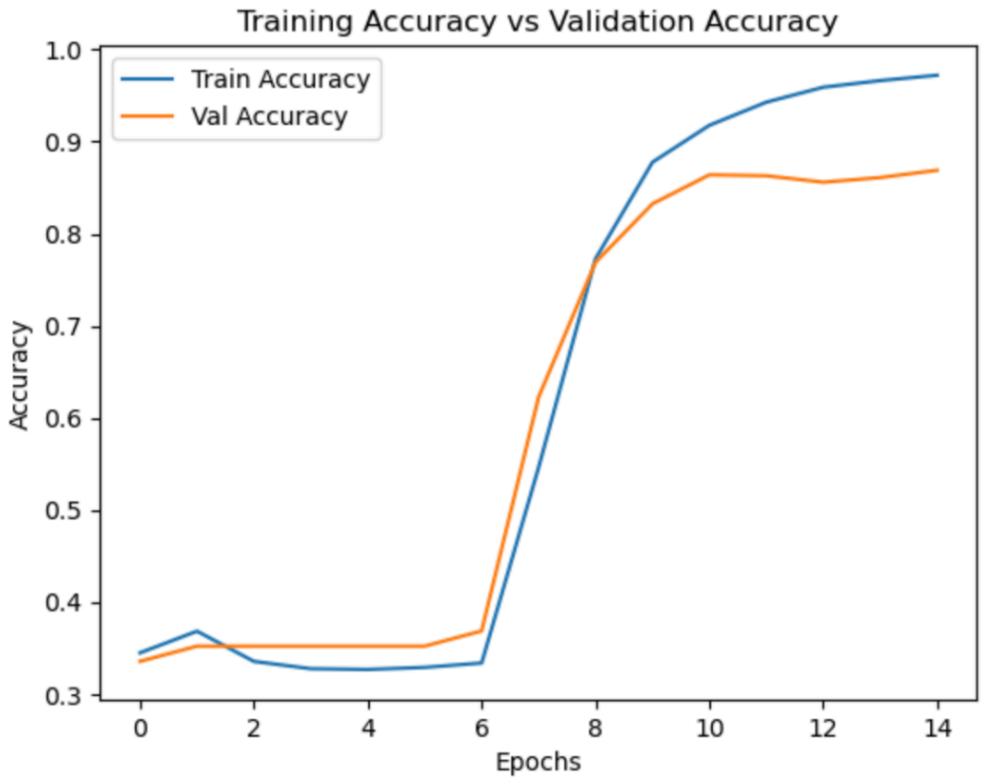
For our supervised methods implementation, we used a neural network model with an embedded layer and LTSM to classify our data into different emotion categories. We chose to work with LSTMs because they have been shown to be successful for sentiment analysis due to their ability to understand sequential information, and therefore context/emotions, in strings. We evaluated the performance of the model

using a confusion matrix, which shows the accuracy of the model's predictions. This confusion matrix shows correct predictions along the diagonal and incorrect classifications on other squares.



Based on the confusion matrix above, the model is the best at recognizing "love", with 624 correct predictions. Additionally, "sadness" is often misclassified as fear, which shows that the model sometimes has trouble distinguishing between the two emotions. Furthermore, "surprise" is the most misclassified, showing that the model struggles with identifying "surprise". Therefore, we can conclude that while our model is good at recognizing certain emotions like "love" because they have certain patterns to them, it struggles with recognizing emotions that have overlapping features that it could not distinguish and limited data.

We also evaluated the performance of our model using a diagram that shows the training and validation accuracy across epochs. The plot below shows that the model's accuracy improves rapidly after about epoch five, but then it starts to plateau around epoch ten. The close alignment between the training accuracy and the validation accuracy shows that the model generalizes well to data it has not seen before without overfitting significantly, which is something our model did well.

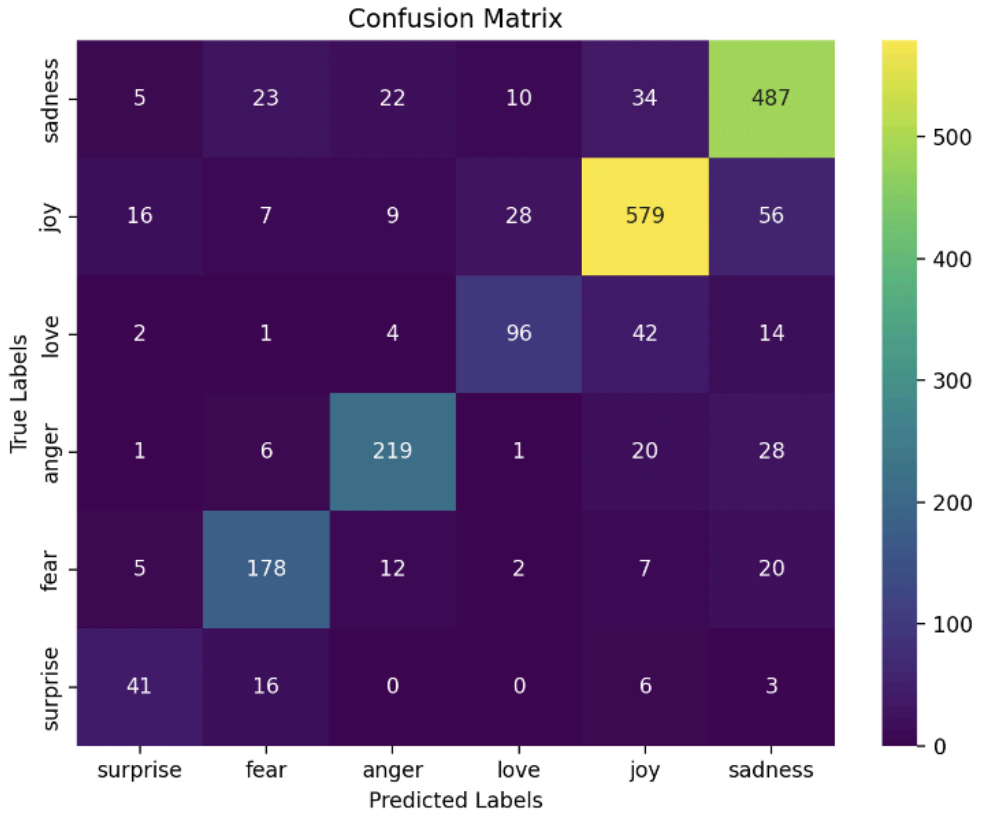


Next Steps: Supervised Method 1

For our next steps, we would improve the model accuracy by experimenting with different data augmentation techniques. For example, in the data, we can try to replace certain words with their synonyms to make classifications better. This would help generate more diverse training samples that could be better for training the model. We will also explore more advanced ways to improve classification like using Transformers because Transformers could make the model's accuracy better by capturing complex relationships and context in the text samples more effectively.

Supervised Method 2

For our second supervised method implementation, we used a support vector machine (SVM) with an RBF kernel to classify our data into different emotion categories. When we first implemented the SVM, we decided that we were going to try to implement it with a linear kernel, but when we checked the accuracy of the model, it was around 70 percent. From there, we tried to determine ways to increase the accuracy of the model. One way that we thought to implement this was to change the way we were preprocessing the data from a tokenizer technique to more of a vectorizer technique. The vectorizer ignores the order of the words and considers the words as just a frequency without a relation between words in a sentence. This is better when considering frequencies for patterns and this improved the accuracy of the model slightly. From there, we realized that it might be hard to find accurate patterns between the different features with a linear kernel, so we decided to use an RBF kernel in order to recognize a non-linear relationship, and from there, we were able to increase the accuracy of the model to 82 percent. We evaluated the performance of the model using a confusion matrix. This matrix shows the number of correct predictions along the diagonal of the matrix and the number of incorrect predictions in all other boxes.



Based on the confusion matrix above, the model was best at recognizing the emotion “joy”, with 579 correct classifications. Therefore, joy is the best-classified emotion, meaning it has patterns that the model can recognize easily compared to the other emotions. However, joy is also frequently misclassified as fear, which shows that the model sometimes struggles with differentiating the two emotions. The emotion that is the most misclassified is “surprise”, Therefore, it can be concluded that while our model is good at recognizing certain emotions like “joy” since they have certain patterns to them, it has some trouble with recognizing emotions that have overlapping features and cannot be distinguished. Below is a table of the precision, recall, f1-score, and support for each emotion category, which provides a detailed evaluation of the performance of the model. 0 represents sadness, 1 represents joy, 2 represents love, 3 is anger, 4 is fear, and 5 represents surprise.

Accuracy: 0.8175
Classification Report:

	precision	recall	f1-score	support
0	0.86	0.88	0.87	581
1	0.79	0.92	0.85	695
2	0.77	0.47	0.58	159
3	0.83	0.76	0.79	275
4	0.81	0.75	0.78	224
5	0.74	0.52	0.61	66
accuracy			0.82	2000
macro avg	0.80	0.71	0.75	2000
weighted avg	0.82	0.82	0.81	2000

Based on this visual, we can see that this model performs at an accuracy of 0.8175. This table also reinforces the fact that “joy” and “sadness” are both emotions that are correctly classified the most since they both have high precision and recall values. It also reinforces the fact that “surprise” is an emotion that the model struggles the most with since it has the lowest F1-score (0.61) and recall (0.52).

Next Steps: Supervised Method 2

For our next steps, we plan on improving the accuracy of the model by experimenting more with hyperparameter tuning for the SVM. For this, we can work on adjusting the C value we used in order to control the trade-off between a smooth decision boundary and correctly classifying training points. We can also modify gamma so that patterns are captured better in the data. We can also experiment with using different kernels like sigmoid and polynomial. This experimentation will allow us to find strengths and weaknesses and further improve our model.

Comparison of Models

In this project, our team decided to use 2 supervised models and 1 unsupervised model on our Kaggle dataset. The first model generated was the supervised model that had a tokenizer for preprocessing along with PCA. The reasons that PCA was chosen for the preprocessing was because we wanted to be able to visualize our features in 2D by plotting these features, and the only way to accomplish this would be by using only the main principle components calculated. When implementing the K-means algorithm on our model, the algorithm performed with a silhouette score of .514. After considering the tweets that were part of each of the sections, labels were determined for each of the sets by sampling 10 tweets and examining the emotions present in each of those tweets. This was beneficial compared to the supervised models because it allowed us to understand what tweets were classified together when the labels are not already pre-assigned to each tweet. We are able to find clusters and also consider possible labels for the clusters given by examining 10 random tweets chosen from each of the clusters and considering what specific classifications they could be. This helped to reveal if there were any other close relationships that were not labelled originally.

When comparing the two supervised models to each other, one of them produced a better accuracy of classifying the tweets based on sentiment correctly. For instance, the model that was trained using the LSTM architecture had a much better accuracy at 86 percent compared to the SVM model's accuracy of 82 percent. Comparing the first supervised method to second, the confusion matrix showed that this surprise label was misclassified the most in both, which shows that both models struggled with identifying "surprise" as an emotion. This suggests that the features representing "surprise" overlap significantly with the features for other emotions or that the dataset does not have sufficient examples to let the models distinguish surprise from other emotions. Although both supervised methods were different in architecture, this similarity shows that there is a consistent feature limitation for this particular emotion. Addressing this could involve extracting features in different ways of optimizing model hyperparameters. Something different between both models is the emotions that were classified the best. For the LSTM model, the emotion that was classified the best was "love," whereas for the SVM model the emotion that was classified the best was "joy". This difference shows how both models handle emotion classification differently. The LSTM most likely classified "love" the best because it is strong at capturing contextual relationships in text, whereas the SVM classified "joy" the best which suggests that the emotion has more clear and distinct features which work well with the SVM's decision boundaries. While both models have strengths, their performance varies based on emotion.

6. GitHub Page Access

Access to our GitHub Repo

7. Presentation Access

Link to Video Presentation

8. GAANT and Contributions Chart

Contributions

Name	Contributions
Aishani Chakraborty	Supervised model #2, Streamlit page
Asmita Karandikar	Final report, final slideshow, final recording, supervised model #2 visualization
Niti Mirkhelkar	Supervised model #2, final report, final slideshow, final recording, supervised model #2 visualization
Sagar Gupta	Supervised model #2
Veena Gonugondla	Final slideshow, final report, unsupervised model visualization revision

9. References

[1] M. Avinash and E. Sivasankar, “A Study of Feature Extraction Techniques for Sentiment Analysis,” Advances in Intelligent Systems and Computing, pp. 475–486, Sep. 2018, doi: https://doi.org/10.1007/978-981-13-1501-5_41.

[2] G. Murthy, S. Rao, B. Andhavarapu, M. Bagadi, and M. Belusonti, “Text based Sentiment Analysis using LSTM.” Available: https://pdfs.semanticscholar.org/0027/d572e43d0c120d59e81c228f2a17b3b05006.pdf

[3] K. Jain, “Sentiment Analysis on Twitter Data,” International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. VI, pp. 3767–3770, Jun. 2021, doi: https://doi.org/10.22214/ijraset.2021.35807.

[4] A. V. Kumar and K. N. Meera, “Sentiment Analysis Using K Means Clustering on Microblogging Data Focused on Only the Important Sentiments,” 2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22), Apr. 2022, doi: https://doi.org/10.1109/icetet-sip-2254415.2022.9791723.