Natural Language Processing- Project Report

Topic: Emotion Classification

Report by

Group 27

Farhan Rasheed Chughtai

Bardia Mouhebat

Ananya Kapoor

Abstract

This project focuses on classification of textual data used to express emotion of a person. Our project tries to solve an NLP problem by designing an efficient machine learning model written in python to be able to distinguish these different emotions expressed. We used five machine learning models: Logistic Regression, Naïve Bayes, Random Forest, Feed Forward Neural Network and Bidirectional LSTM recurrent Neural Network to achieve this goal. We identified it as the best performing model based on F-1 score for all the classes.

1.0 Introduction

Emotion detection is an important element of understanding human behavior. Since human emotions affect their decision-making, their interaction with other human beings, and contribute to human intelligence. Recent studies have shown that real-time emotional analysis has been very useful in predicting emotional and behavioral consumer features. Consumer preferences were found to change based on their emotional state, especially in pre-sales and after sales purchases. What was more interesting for us was to use this classification problem to help in understanding customer reviews on a particular product and adjusting this variable into the rating system, instead of manually averaging the rating given by customers. This led us to dig deeper into this topic and we interestingly found that this is not only an NLP problem but also a widely studied topic in psychology, neuroscience, and behavioral science.

Also, we found that this analysis of emotions is highly useful for financial prediction, political decisions, presidential election prediction, personalized recommendation, healthcare (e.g., depression screening), and online teaching (ex, curriculum arrangement) and feedback evaluations.

2.0 Data description

There are two broad categories of human emotions which are positive emotions (happiness/joy Surprise, love) and negative emotions (anger, fear, disgust). The dataset is made up of 20,000 sentences and the distribution of the labels is as shown below:

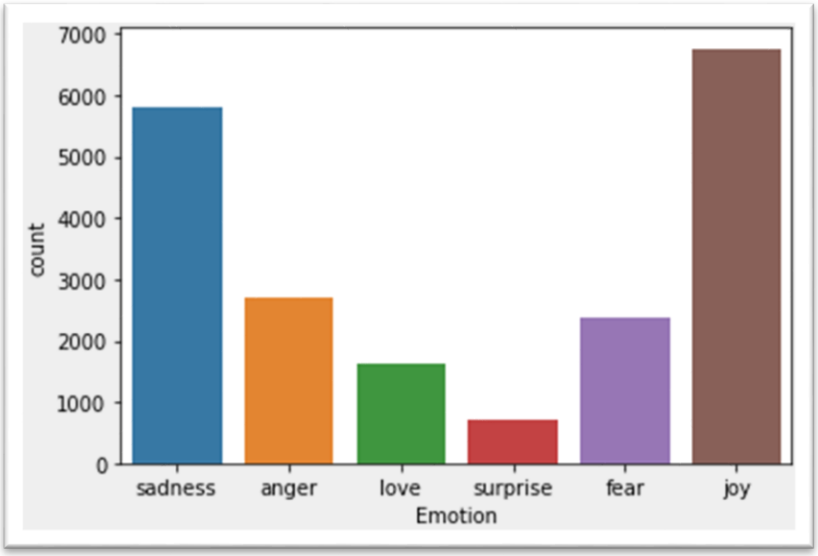

Figure 1: Data distribution of emotions dataset.

3.0 Methodology

The model's we selected for our task of emotion detection are:

1. Logistic Regression
2. Naïve Bayes
3. Random Forest
4. Feed Forward Neural Network
5. Bidirectional LSTM recurrent Neural Network

Out of these we will be using libraries for model logistic regression, naïve bayes and random forest and we will be using additional NLP methods to implement Neural Network models partially designed by us and partially by using libraries.

3.1 Logistic Regression:

Logistic regression is a simple statistical model that calculates probability of an event occurring based on the dataset we have given. So, it is a very basic but useful probabilistic model used for classification task that is why we chose it.

3.2 Naive Bayes:

Naive Bayes again is a probabilistic model like logistic regression but the probability it calculates is based on the bayes theorem with the assumption that there is strong independence between the features.

3.3 Random Forest

Random forest model is also a popular classification model that uses a multitude of decision trees and the class most picked by these random decision trees the class that is predicted by the model.

3.4 Feed Forward Neural Network

Are the most simplified form of neural network Models. They have no cycles or loops, or the information is not shared backwards it simply goes through the input nodes to the hidden nodes and then to the output nodes.

3.5 Bidirectional LSTM Recurrent Neural Network

It is especially designed for NLP tasks. LSTM long short-term memory recurrent neural network is used to recognizing the relationship between words from start to the end, so it helps to look at the whole sentence together instead of looking at words individually or in a window. We also used the bidirectional version of it so it does two pass one from start to the end and then from the end to the start which is very beneficial to figure out relationship between words as reading the sentence from end to the start can make a lot of difference in understanding it's context.

3.6 Training the Models/Data Preprocessing

We started out by doing a train, validation and test set split with the following proportions 80,10 and 10 respectively. This was done at the very start of the project so we don't do anything that will influence our results and just do our preprocessing of the training data.

The first hurdle we came across was to fix the disproportion of the label's distributions in our data set. Mainly love and surprise emotions had fewer samples than the rest of the emotions. To fix that we tried the over sampling technique which gave us some good results. Do be clear over sampling was done only on the training set.

Model	Emotion	F1-Score without Over Sampling	F1-Score with Over Sampling
Random Forest	**Surprise**	**0.75**	**0.77**
Random Forest	**Love**	**0.75**	**0.78**

Table 1: Random Forest results with and without over sampling

So according to the results above **we were getting better performance for the emotions that had less samples,** so we decided to go with oversampling for all the models except the neural network ones.

After that we started our feature engineer process for our model's and for that we tried different things the first thing we tried was to compare **binarized bag of words with normal bag of words and we saw a small increase in performance when we used binarized bag of words on the validation set so we went with that.** The next thing we did was to i**ncorporate emotion lexicon as features** for our model's and the result we got from that was not what we were expecting. We incorporated the following lexicons negative, positive, anger, fear, joy, sadness and surprise.

Model	Accuracy without lexicons	Accuracy with lexicons
Random Forest	**0.87**	**0.80**
Logistic Regression	**0.90**	**0.89**

Table 2: Results with and without lexicons

From the results above you can see that our overall accuracy of the models decreased and in some models by a lot, which was very surprising to us as we expected it to increase but that was not the case. We later found out the reason was that many words are common in these lexicons due to the nature of emotions a word can be used to describe many emotions, it is the context it was used in that tells us the actual emotion. So, due to this, the lexicon features were not helpful because they did not add information about the training set, which is why we saw reduced performance in some of our models.

Moving on we created the bag of words using the training set that means the count vectorizer only had the vocabulary of the training set and we used that same vectorizer on the validation set and the test set so that means any words that were not present in the training set were not included in the bag of words for the validation and test set. One further modification we did was that while creating the bag of words we ignored the stop words, so they were not included in the vocabulary of our bag of words. This is how our first three models were trained.

3.7 Architecture of Neural Network Models:

Moving on to the neural network models, we did not use any over sampling because it was not making any difference to our results. For our first Neural Network Model which was the feed forward neural network model we used the below architecture:

1. Embedding Layer (Glove 100d Embedding Matrix)
2. Output layer (6, activation=” SoftMax”)

It is a simple two-layer architecture with embedding layer populated by the Glove-100d embeddings. We saw that its performance was adequate, but it was not able to beat the logistic regression model so that is why we decided to move on to Bidirectional LSTM Recurrent Neural Network models which are much more powerful and geared towards NLP tasks.

For our Bidirectional LSTM Model, we used the below architecture:

1. Embedding Layer (Glove 100d Embedding Matrix)
2. Bidirectional (LSTM (64, return_sequences = True)
3. Bidirectional (LSTM (64, return_sequences = True)
4. Bidirectional (LSTM (32,)
5. Dense (32, activation = 'relu')
6. Dense (6, activation = 'softmax')

So, this one had 4 hidden layers, three of them being bidirectional LSTM layers and one of them being a relu activation layer with 32 hidden nodes.

4.0 Results and Conclusion

Overall, we experimented with five different models for emotion analysis. We try to tune the hyperparameters and find the best version of each model based on the validation results. The results for the test set for these models can be seen in table 3. It should be mentioned that the inputs for the first 3 models are bag of words and for Neural Networks we use word embeddings.

Model	F1-score	Accuracy	Precision	Recall
Random Forests	**0.85**	**0.87**	**0.83**	**0.87**
Naïve Bayes	**0.77**	**0.81**	**0.75**	**0.79**
Logistic Regression	**0.87**	**0.90**	**0.85**	**0.89**
Feed Forward Neural Network	**0.77**	**0.83**	**0.77**	**0.77**
Bidirectional LSTM Recurrent Neural Network	**0.89**	**0.93**	**0.89**	**0.90**

Table 3: Results on the test set for all the models

The reported F1-score, precision and recall are macro-average scores, which are calculated as arithmetic mean of individual classes scores. As shown in the table, Bidirectional LSTM Recurrent Neural Network has the best performance for the task of emotion analysis. Also, Logistic Regression and Random Forests have good performances for this task with 0.87 and 0.85 F1-scores respectively.

Regarding Feed Forward Neural Network we expected better results, at least as good as logistic regression which is a simple Neural Network model. However, the input for the models is different. We use word embeddings in Neural Networks compared to bag of words for Logistic Regression.

The results for Bidirectional LSTM Recurrent Neural Network can be seen in table 4. The model performs best in classifying sentences with joy and sadness. It also has the worst performance on sentences with surprise which has the least occurrences in our dataset.

Emotion	Precision	Recall	F1-Score	Support
Anger	0.91	0.95	0.93	261
Fear	0.90	0.79	0.84	231
Joy	0.96	0.95	0.96	675
Love	0.85	0.86	0.86	161
Sadness	0.96	0.97	0.97	584
Surprise	0.77	0.86	0.81	88
Macro Average	0.89	0.90	0.89	2000
Weighted Average	0.93	0.93	0.93	2000

Table 4: Results on the test set for Bidirectional LSTM Neural Network

5.0 Future Experiment/Work

We could have tried the BERT Bidirectional Encoder Representations from Transformers language model to see its performance compared with the rest of the models we used in the project.

Due to time constraint and limited resources, we could not experiment with truly deep learning models by that we mean models with at least 15 or 20 layers to see if that would have increased our neural network performance. Another approach we could have tried would see how wide models would perform instead of deep multi-layer models.

We did not test our model on any other data set so for our future work we would test our models on different emotion datasets to see how it would perform on other types of data. One potential dataset could be tweets so we can predict what type of emotion the tweeter is expressing.

6.0 Works Cited:

https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

https://www.kaggle.com/code/aryan7781/emotions-classification-using-lstm

https://stackabuse.com/python-for-nlp-movie-sentiment-analysis-using-deep-learning-in-keras

https://www.codementor.io/@agarrahul01/multiclass-classification-using-random-forest-on-scikit-learn-library-hkk4lwawu

https://www.mygreatlearning.com/blog/bag-of-words/

https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm