Decision trees (DT) and random forest (RF) for emotion analysis

Tyler Liddell, Msci Computer Science, City University of London

Description and Motivation

Emotion analysis is an important and challenging task in the field of NLP. While similar to task of sentiment analysis, emotion analysis goes further and aims to understand subtle human nuances and classify text as a specific emotion. We will take a machine learning approach to the problem using a dataset consisting of Twitter messages that must be labeled one of six emotions: sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5). We aim to solve this problem using the Decision Tree (DT) and Random Forest (RF) algorithms.

Preprocessing and Initial Analysis

- The dataset is composed of 20,000 Twitter messages (rows) and two columns, one containing the text and the other containing the number corresponding to the correct emotion.
- The dataset is made publicly available by DAIR.ai. It was presented in the CARER (Contextualized Affect Representations for Emotion Recognition) paper which gives graph-based algorithm used to create context aware representations of text for the use of emotion recognition in text [1].
- Much of the preprocessing already carried out such as lowercasing and punctuation removal.
- Dataset is moderately imbalanced, with the labels “sadness” and “fear” making up nearly %70 of the data, and surprise only being in %4 of the entries. (figure 1)
- Removed infrequent words and stop words
- Created a separate bag of words for each emotion to analyze the word counts. (figure 2)

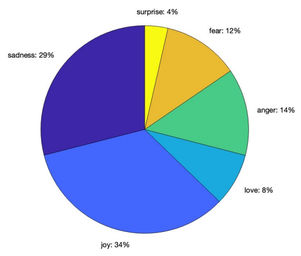

Figure 1. Pie Chart

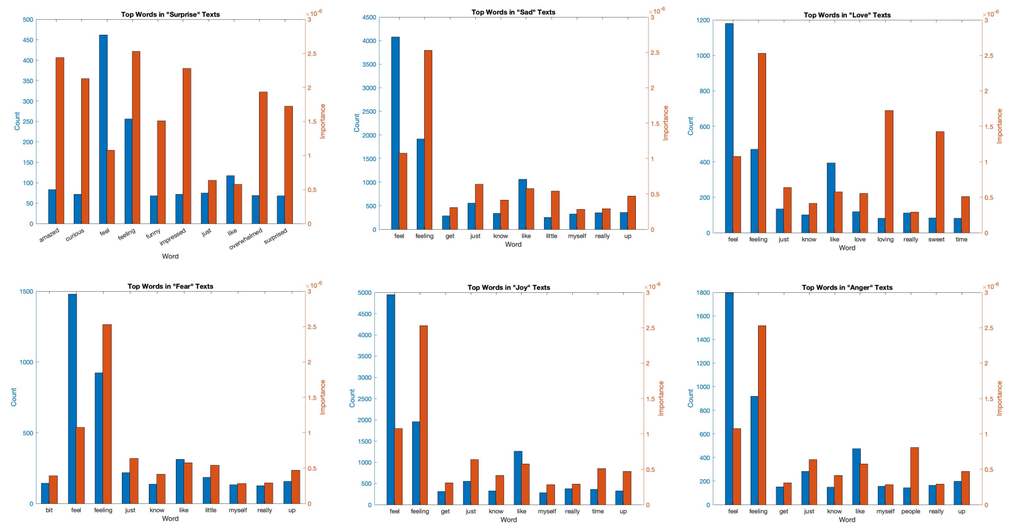

Figure 2. Bar chart of frequency (blue) vs. importance (orange)
* count values adjusted based on size of label’s bag

- Used TF-IDF to turn the text into matrices. TF (term frequency) is the number of times a word appears in a document. IDF (inverse document frequency) is used to determine how important a word is in a document [2].
- Used a DT to find predictor importance. After previous preprocessing was carried out, the dataset contained 5108 features. This number was reduced to 765 using the DT.
- Using this decision tree, the importance of each word is calculated by “summing changes in the risk due to splits on every predictor and dividing the sum by the number of branch nodes.” [3] (figure 2)

Decision Trees

- A decision tree, employed in both classification and regression tasks, is a visual and interpretable algorithm. Its tree-like structure is formed by evaluating features to create nodes, while branches and leaves depict potential outcomes, each associated with a distinct probability. [4]

Strengths
- Easy to implement. Easy to interpret.
- Doesn’t require very much data preparation and can handle missing values.

Weaknesses
- Prone to overfitting.
- Unstable to noise and changes in the data.
- Biased to unbalanced classes.

Random Forests

- A random forest randomly generates a set of possible trees and makes a classification decision based on which class is most popular. [5]

Strengths
- Less prone to overfitting due to multiple trees.
- Can make more accurate predictions with large or complex datasets as opposed to a DT.

Weaknesses
- Unlike a DT the RF can be difficult to understand and difficult to control.
- Can be computationally intensive, especially if many trees are required.

Hypothesis

- Random Forest will be able to realize more complex relationships between the words and outperform decision trees in accuracy by a moderate margin (paper about random forest performing better on sentiment analysis and large datasets).
- Decision Tree may not perform as well in terms of accuracy but will have a significantly faster training and optimization time.
- By utilizing word embeddings to enhance semantic representation, the models will be able to achieve competitive accuracy when compared to the fine-tuned BERT models found on the dataset’s leaderboards as well as the DT and RF implemented on a similar dataset containing 8 emotions [6].

Methodology

- Initial random split of the data 70 - 30 into training(16000 rows) and testing(4000 rows) sets.
- Use TF-IDF to create matrices of the text.
- Employ feature selection using an out-of-the-box decision tree to retain only predictive features.
- Develop custom grid search loops for each model to explore and optimize different parameters.
- Use 10-fold cross validation during the training of the models to test performance and avoid overfitting.
- Compare the optimized DT and RF model’s performance on the test data.

Parameter Selection and Experimental Results

Decision Tree

- Used CART algorithm to fit the model and tested different split-criterion against each other. Gini’s diversity index(GDI) and twoing performed the best as seen in figure 6. Used GDI in final model.
- Increased the observations per node (minimum leaf size) by 2 to find the optimal value of 6. (figure 6)

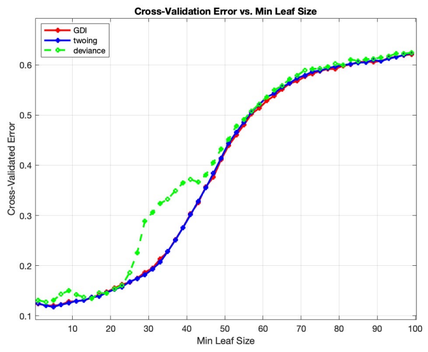

Figure 6.(Left) Grid search for min leaf size.

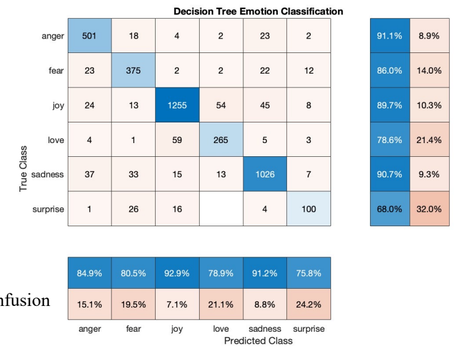

Figure 7.(Right) Confusion Matrix for DT

Random Forest

- Used a grid search to find the optimal number of trees (43) by increasing number by 2 and comparing cross-validation error. (figure 8)
- Tested the different split criterion and used a grid search to find the optimal observations per node. The split criterion are all equal, but RF does best with default min leaf size of 1. (figure 9)

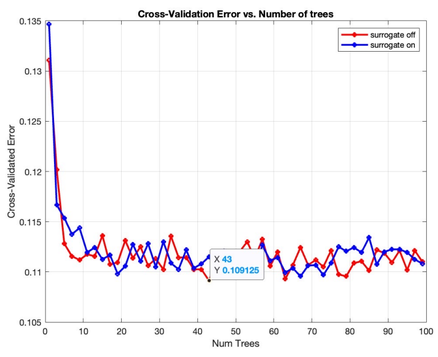

Figure 8.(Left) Grid search for optimal number of trees

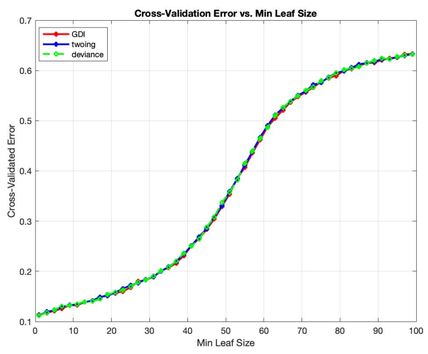

Figure 9.(Left) Grid search for min leaf size.

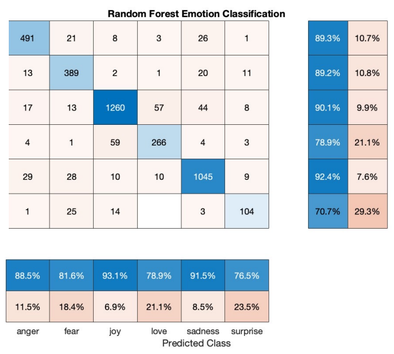

Figure 10.(Right) Confusion Matrix for RF

Model	DT	RF
Training Acc.	.897	.962
Testing Acc.	.881	.887
Precision	.840	.843
Recall	.840	.845
F1 Score	.841	.845
Training Time	4.8s	14.8s
Testing Time	.01s	.80s

Figure 11. Model Performance

Analysis and Evaluation of Results

- Using preprocessing and predictor selection to reduce the number of features (words) resulted in models that performed much faster and with identical or slightly better accuracy. Training the models on the original 15212 features took 230 seconds for the DT and 389 seconds for RF. In contrast training on the reduced 765 features took approximately 5 seconds for DT and 15 seconds for the RF.
- A custom grid search was useful for understanding the effect increasing the number of observations per node has on each of the models as well as different methods of split-criterion. As seen in figures 6 and 9 an increase in min leaf size has a similar effect on both the decision tree and the random forest. Controlling the amount of “leafiness” the models have results in growing deeper trees [7]. This was important to moderate because growing trees that are too deep or too leafy can result in overfitting the models. Similarly, the same was done find the optimal number of trees for the random forest model. In figure 8 we see that the model performs best with a relatively small number of trees.
- RF has a much higher training accuracy; this is typical for these models as they tend to overfit and give generous estimates of their prediction abilities [8]. To avoid overfitting and get a better estimate of the model's performance, I opted for Cross-Validation evaluation during the training stage (figures 8 and 9).
- By evaluating the confusion matrices (figures 7 and 10), we can see just how similarly the models perform. For instance, both models perform the worst when predicting surprise (the minority class) and misclassify the majority of these as fear, which is arguably the most similar emotion. This trend is evident with some of the other classes as well, such as love and joy.

Hypothesis Predictions

- Contrary to the hypothesis statement, the models perform very similarly and using random forest provides a miniscule advantage in terms accuracy. (figure 11) This is due to the depth of the decision tree sufficiently capturing the relationship between features.
- As anticipated in the hypothesis the DT was much simpler to implement and had a much faster training and testing time, taking approximately 10 seconds less to train and only .01 seconds to test.
- The initial goal from hypothesis statement was to achieve competitive performance using word embeddings. This method proved to be unsuccessful and provided poor performance (less than %50). However, using TF-IDF to create matrices was more than sufficient for the task and the accuracy of the models was only approximately %6 less than that of the much more complex BERT models. The DT and RF models were on par with the models shown in [NC people] on the similar 8 emotion dataset.

Lessons Learned

- Sometimes a model fits the problem well and there won’t be much performance gain in terms of overall accuracy when optimizing hyperparameters. In this scenario it is good to focus on other metrics that can be improved such as speed and complexity.
- Bigger does not equal better. I initially hypothesized that the RF model would be able to capture more complex relationships between the texts than the DT. This turned out to be false, and DT was on par with RF in every metric.

Future Work

- Reduce the imbalance of the dataset by bringing in more data for minority classes from DAIR.ai’s other, larger twitter emotion dataset.
- Experiment with some additional NLP techniques such as part-of-speech tagging or syntactic parsing to gain more insight into the linguistic structure of different emotions.
- Use additional weighting in the TF-IDF for words that are more important in the decision tree or provide more information about a specific emotion.

References

[1] E. Saravia et al., "CARER: Contextualized Affect Representations for Emotion Recognition," in Proc. 2018 Conf. Empirical Methods Natural Language Processing, Brussels, Belgium, Oct-Nov. 2018, pp. 3687-3697. [Online]. Available: https://aclanthology.org/D18-1404 DOI: 10.18653/v1/D18-1404.
[2] C. S. Yang and G. Salton, "On the specification of term values in automatic indexing," *J. Documentation*, vol. 29, no. 4, pp. 351–372, 1973. doi: 10.1108/eb026562.
[3] MathWorks. (n.d.). "predictor importance." [Online]. Available: https://uk.mathworks.com/help/stats/compactclassificationensemble.predictorimportance.html
[4] Breiman, L. (1984). Classification and Regression Trees (1st ed.). Routledge. https://doi.org/10.1201/9781315139470
[5] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324
[6] J. Ranganathan, N. Hedge, A. S. Irudayaraj, and A. A. Tzacheva, "Automatic Detection of Emotions in Twitter Data: A Scalable Decision Tree Classification Method," in *Proceedings of the Workshop on Opinion Mining, Summarization and Diversification* (RevOpiD '18), Baltimore, Maryland, 2020, pp. 2-11. doi: 10.1145/3301020.3303751.
[7] MathWorks. (n.d.). "Improving Classification Trees and Regression Trees." [Online]. Available: https://uk.mathworks.com/help/stats/improving-classification-trees-and-regression-trees.html#bsw6baj
[8] MathWorks. "Methods to Evaluate Ensemble Quality." [Online]. Available: https://uk.mathworks.com/help/stats/methods-to-evaluate-ensemble-quality.html