Sentiment Analysis on Movie Reviews

Machine Learning and Data Mining

Project

Name: Arshdeep Singh Bhogal
Zid: z5225346

GIT HUB: https://github.com/SinghArshdeep/Sentiment-Analysis-on-Movie-Reviews

Google Drive:
https://drive.google.com/drive/folders/1o9ODoklNFpX-rWmZLORBko9DDCaBIJ5G?usp=sharing

1. Introduction

Since the beginning of time, sentiments/emotions have influenced people’s decision-making, social interaction, learning, creativity, and more. This has led to either great moments in history or horrific apocalypses. Even in our day to day life, people convey their emotions through speech, expressions, body language, many more. Therefore, understanding the emotions/sentiment behind a phrase is very important. Imagine it has been a bad day, and suddenly you need to write a speech for your boss, a tool that keeps your tone in check will save the day. In this project, I try to make such a tool that will be able to predict the sentiments in a person’s comment. Such tools/models can reveal public opinions and help us see the effect of various things.

The challenge is to finely sort new movie reviews accurately in a cost-effective way by looking at the phrases. Humans can identify other persons intent by only listening to their tone of voice. Mapping this to a computational tool will be a life-changing innovation. The dataset and project can be found at the Kaggle website (https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews).

2. Data Analysis

2.1 Input Datasets

The datasets provided contains 156,060 phrases, each labelled into the following categories:

0. Negative
1. Somewhat Negative
2. Neutral
3. Somewhat Positive
4. Positive

After evaluating the data, we find that the amount of data in each category is inconsistent and varies with massive amounts. The negative and positive data constitutes only 10% of the data, while the neutral reviews are more than 50% of the data. This variance will create a strong bias in the model, deciphering more reviews to be neutral.

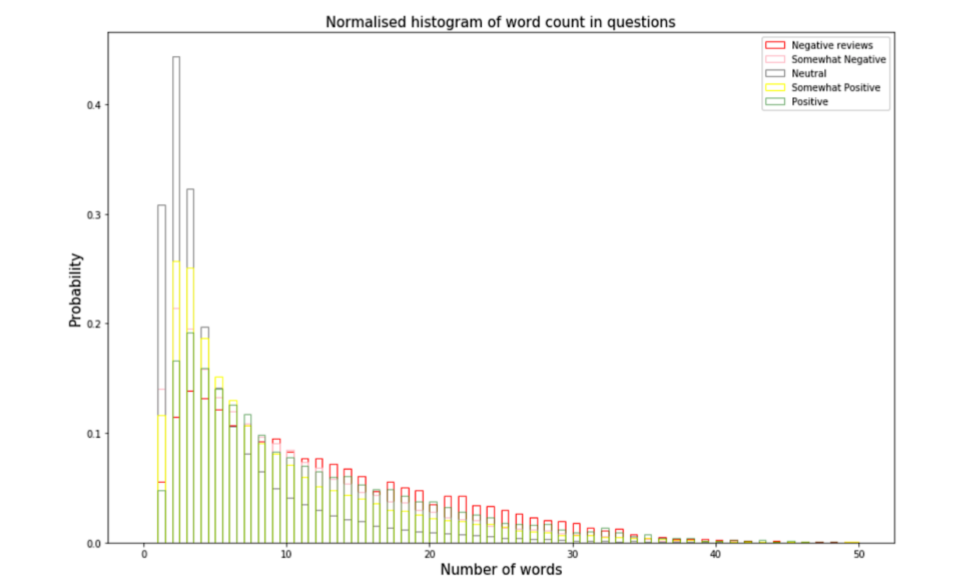


Figure 1. Number of words in a Phrase

Another way to check a bias is to check each phrase's length in all of the categories. As shown in figure 1, the negative reviews tend to be slightly longer than positive or neutral ones. Furthermore, neutral reviews tend to be more abrupt and short. This disparity shows people's bias towards expressing more sorrow than joy. To remove the biasing in our model, we try to use the TF-IDF vectorizer, which penalizes the frequent words interest domain and evaluates its relevance as a whole. The predictions for this model are listed in the results.

2.2 Text Representation

Having a look at the consistent vocabulary in the phrases gives an excellent idea for word feature selection. It would be useful to make a word cloud for each sentiment and check the most common words.

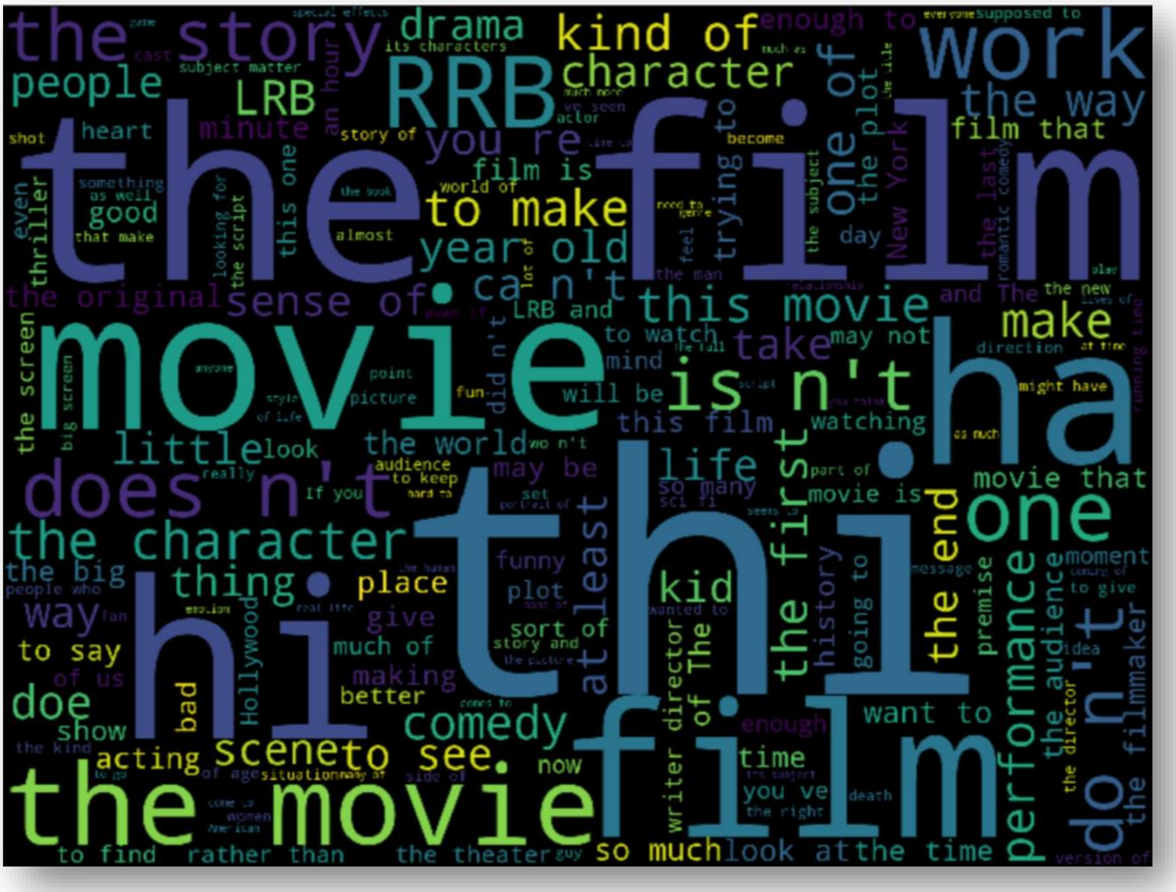


Figure 2. Word Cloud for Neutral Reviews

Examining figure 2 and Appendix A, it is hard to miss that the words in most of the reviews try to depict an emotion; hence, there should be a feature associated with it. Also, removing the punctuations and the filler words will clean the dataset. This process would reduce redundancy and further improve our features.

3. Methodology

3.1 Feature Engineering

This step is one of the most critical factors determining how well our model will perform. First, the text is converted to lowercase to reduce the discrepancies along with removing all the punctuations. Further, we lemmatize the phrases, meaning reducing them to their present form or to their roots (playing, plays changes to play). Next, all the stop words like of, the, which, are removed. This pre-processing would drastically reduce the computational expenses.

We try to add another feature that correlates the number of words in a sentence to the type of review. Since this number would be much higher than one, the model might interpret it has more weight and add a bias to it. To avoid it, the logarithmic value of this number is taken, which keeps the value closer to 1.

After this, we categorize the words into eight emotional categories provided by the NRC Word-Emotion Association Lexicon (aka EmoLex). This Lexicon also provides us with a score between 0 and 1 relating it to the type of emotions (Value closer to 1 means a more definite relation, which will add more weight to it). Adding these scores creates a new score, which can be used as a feature to relate the words to its emotions.

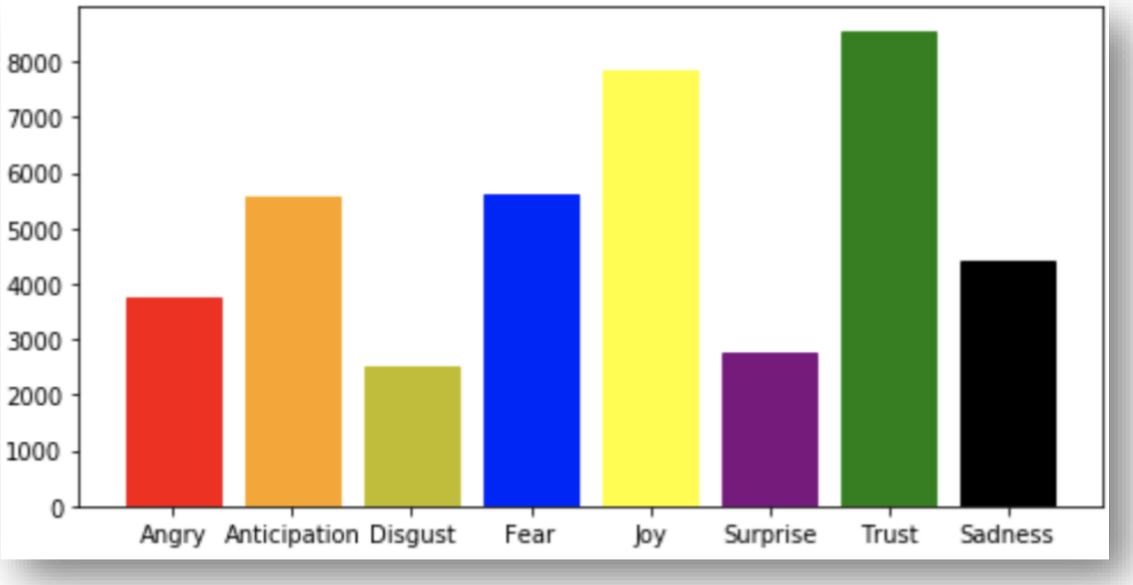


Figure 3. Score for Emotions in the Neutral Reviews

Figure 3 and Appendix B shows the scores of emotions in a type of review.
Dark emotions like sadness and anger have a higher score in negative reviews, while light emotions like joy and trust have a higher score in positive reviews. These scores show us the range of emotions in a review and help associate them with a feature.

For input into a model, we need to combine all these features into a suitable format. A good option is to use a Data Frame Mapper. A mapper can combine features by applying different processing to parts of it. With the help of a Count Vectorizer, we tokenize the lemmatized words and process them for extracting additional features. To avoid outliers, we drop the words which occur less than five times or are too common. We now combine these vectors to other features using a mapper, which generates a vector form. Since most of the rows will contain zero, this is converted into a sparse matrix for faster computations and saving memory.

3.2 Model Training

Firstly, we will fit simple/faster models to evaluate the features and check which type will perform the best on the data. They are trained on the whole dataset but only with the best 109 features.

Table 1. Accuracy for Simple Models

Model	Accuracy
K Nearest Neighbours	53.1
Multinomial Naïve Bayes	52.8
Logistic Regression	53.3
Decision Tree	55.3

As seen in Table 1, all the models give a similar result. Decision Tree Classifier seems promising, with an accuracy of 55%. It breaks down the data into smaller subsets and uses keywords as a deciding node. With these results, we can move to use more sophisticated and complex models like SVM or a Random Forest Classifier.

SVM is well known for text analysis and working in higher dimensions. However, it is costly to train. In contrast, Random Forest works on the same implementation as Decision Tree and should perform well by categorizing essential words in each phrase.

3.2 Hyper Parameter Training

Randomized Search CV takes a list as input and runs cross-validation on a random combination of the parameters to find the best one. It is more time-efficient than grid search and works well on a large dataset. The optimal parameters for the Random Forest Classifier come to be:

{'n_estimators': 150, 'min_samples_split': 10, 'min_samples_leaf': 1,
‘max_features': 'auto', 'max_depth': None, 'criterion': 'entropy', 'bootstrap': False}

After splitting the dataset in a 20% test and 80% train set, the model is trained with the above parameters.

4. Results

To remove the bias in the data, a Multinomial Naïve Bayes model is trained on a TFID processed 80% training data and tested on the remaining. The results are as following:

Table 2. Accuracy with TF-IDF processing

Sentiment	Precision	Recall	F1-Score
Negative	0.05	0.56	0.10
Somewhat Negative	0.28	0.52	0.36
Neutral	0.88	0.61	0.72
Somewhat Positive	0.40	0.52	0.45
Positive	0.04	0.66	0.11
ACCURACY		**58.8%**	

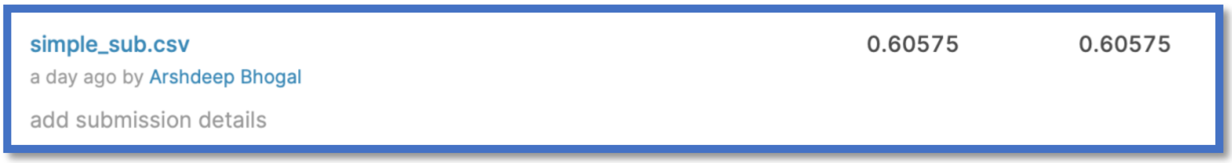
In contrast to what we predicted, the TFID is not able to remove the bias for classes with less amount of training data. This statement can be confirmed by looking at the precision of predicting the negative/positive reviews, which is substantially low compared to the Neutral reviews.

Our final, Random Forest Classifier predicts the sentiments with an accuracy of 64% (Table 3) and high precision for negative/positive reviews. The following table shows the results obtained from the model.

Table 3. Accuracy with the Final Model

Sentiment	Precision	Recall	F1-Score
Negative	0.37	0.46	0.41
Somewhat Negative	0.49	0.54	0.51
Neutral	0.79	0.73	0.76
Somewhat Positive	0.52	0.55	0.45
Positive	0.39	0.52	0.45
ACCURACY		**64.1%**	

When trained on the whole training set and submitted, the model receives an accuracy score of 0.61 on Kaggle.



simple_sub.csv 0.60575 0.60575

a day ago by Arshdeep Bhogal

add submission details

Figure 4. Kaggle Submission

5. Discussion and Future Work

In this project, we use Random Forest Classifier to associate movie reviews with a sentiment. The metric used for comparison is accuracy, which is the ratio of the correct predictions to the number of predictions. The model is reasonably accurate, with an accuracy of 64%. However, as predicted, it can classify neutral reviews with more precision due to the significant bias in the dataset. To remove this bias, we try to use a TF-IDF. However, this did not give a good result, as discussed above. Other ways to remove biases include sampling, SMOTE, and many more, which can be pursued in further studies.

The biggest challenge faced in text sampling and similar projects is to handle sarcasm and negation in sentences. Since this model only uses a word as a feature, it would be hard to correlate the meaning of a few words and the meaning of the sentence. Few examples:

- I do not dislike pop music. (Negation)
- I truly love Justin Bieber. (Sarcasm)
- This is badass.
- Disliking carbonated drinks are not my thing.

This problem can be handled using multiple words as a feature, which turns out to be very expensive, with more than 60,000 features in this project.
One more way to classify movie reviews is to find the helpfulness in the sentence rather than the sentiment [2].

6. Conclusion

The primary purpose of this project is to find more straightforward ways to evaluate sentiments in a phrase. While Random Forest Classifier shows promising results in predicting sentiments in a phrase, it should be considered that with so many challenges and possibilities in the future ahead, more research should be done. More expensive and complex models like SVM, LSTM might be more suitable for text analysis.

References

[1] Pang and L. Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In ACL, pages 115–124.

[2] Liu, Yang; Huang, Xiangji; An, Aijun; Yu, Xiaohui (2008). "Modeling and predicting the helpfulness of online reviews" (PDF). ICDM'08. Eighth IEEE international conference on Data mining. IEEE. pp. 443–452.

[3] Arif, M.N.I., Paplu, S.H. and Berns, P.D.K. (2019). Understanding Emotional Expressions in Text-Based Communication. [Academic Seminar]. Available at: https://www.academia.edu/43764275/Understanding_Emotional_Expressions_in_Text_Based_Communication.

[4] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Chris Manning, Andrew Ng and Chris Potts. Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).

[5] Jonathansoma.com. 2020. NRC Emotional Lexicon. [online] Available at: http://jonathansoma.com/lede/algorithms-2017/classes/more-text-analysis/nrc-emotional-lexicon/

[6] Ganesan, K. (2019). 10+ Examples for Using CountVectorizer. [online] Kavita Ganesan. Available at: https://kavita-ganesan.com/how-to-use-countvectorizer/#.Xy-zny2r3RZ

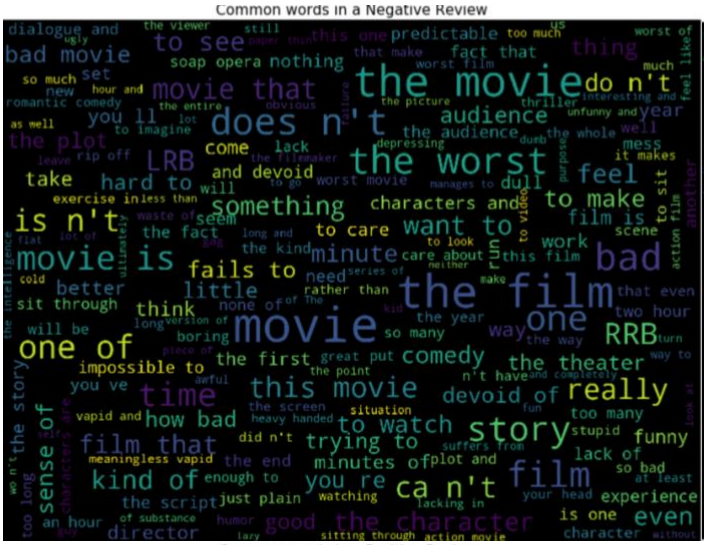
[7] Abusalah, M. (2019). Re-sampling Imbalanced Training Corpus for Sentiment Analysis. [online] Available at: https://medium.com/analytics-vidhya/re-sampling-imbalanced-training-corpus-for-sentiment-analysis-c9dc97f9eae1

[8] Scikit-learn.org. (2019). Working With Text Data — scikit-learn 0.21.2 documentation. [online] Available at: https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

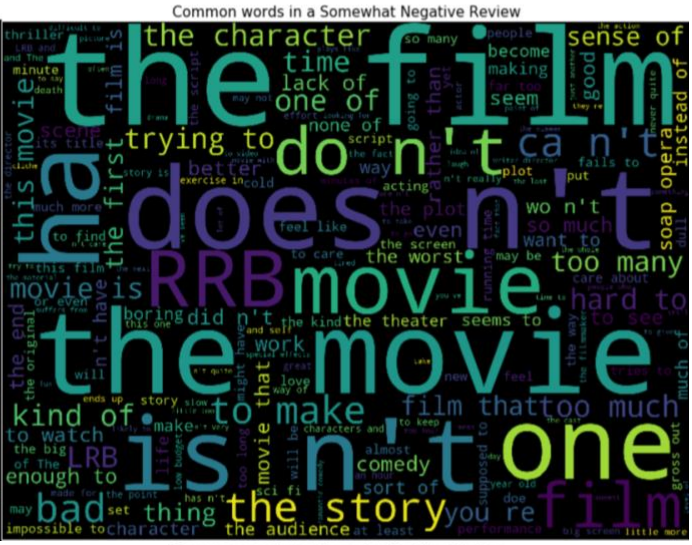
[9] Nabi, J. (2018). Machine Learning — Text Processing. [online] Available at: https://towardsdatascience.com/machine-learning-text-processing-1d5a2d638958

APPENDIX A

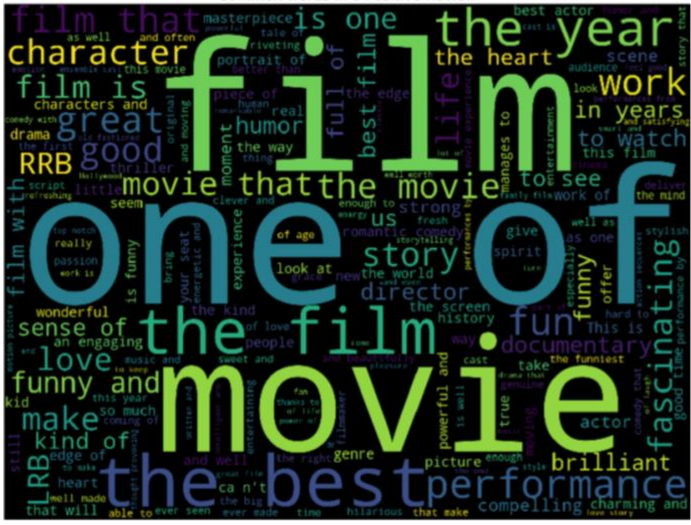
Common words in a Negative Review



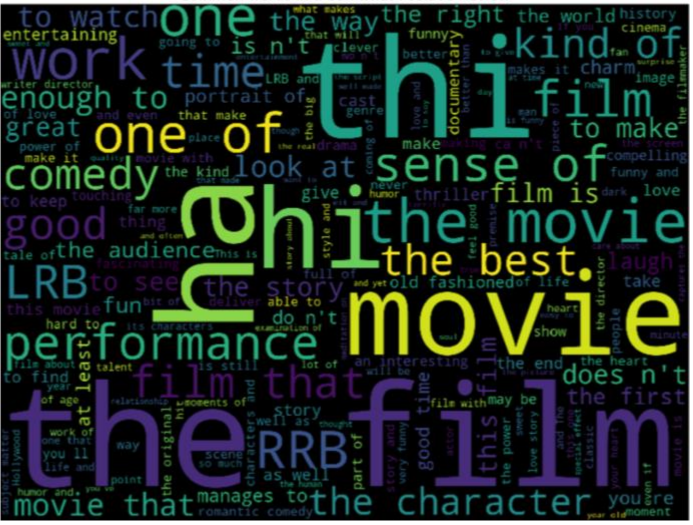
Common words in a Somewhat Negative Review



Common words in a Positive Review



Common words in a Somewhat Positive Review



APPENDIX B

CountVectorizer():
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

TfidfVectorizer():
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

KNeighborsClassifier():
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

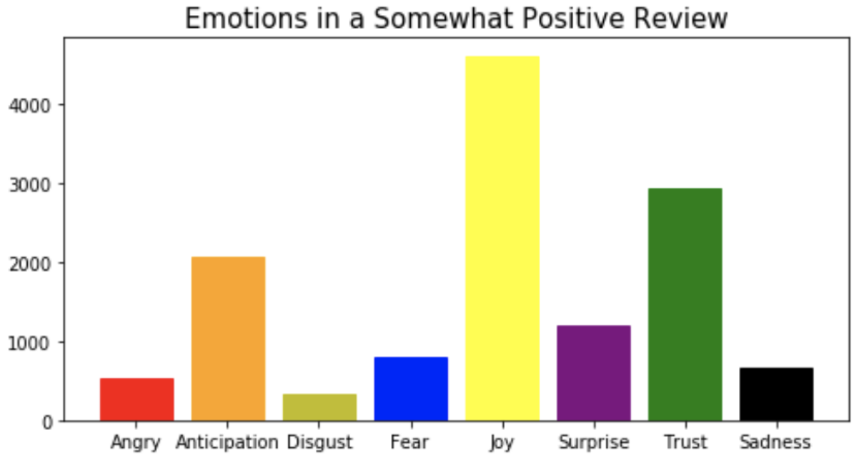
DescisionTreeClassifier():
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

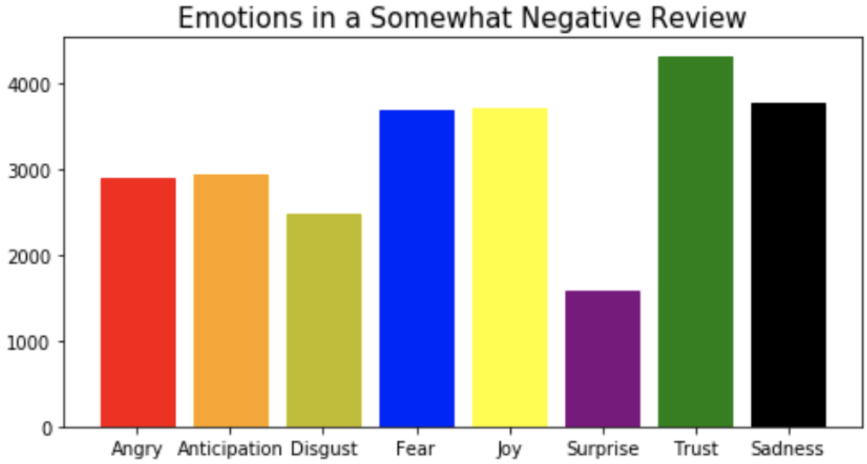
RandomForestClassifier():
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

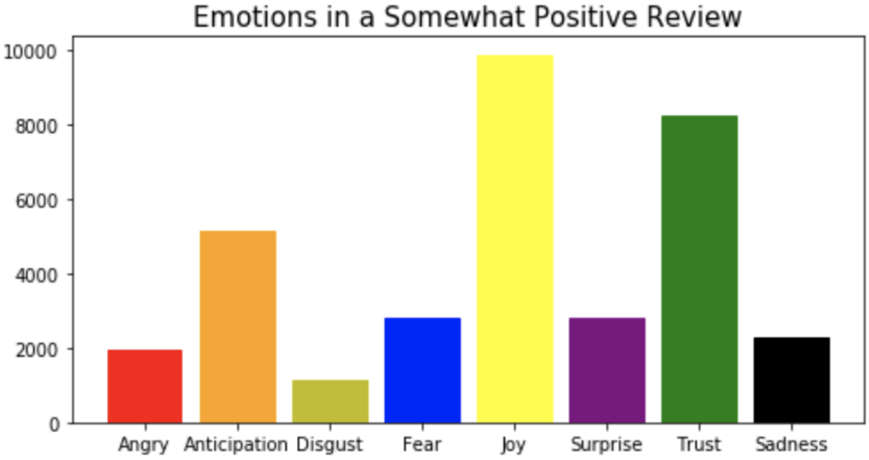
RandomizedSearchCV():
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

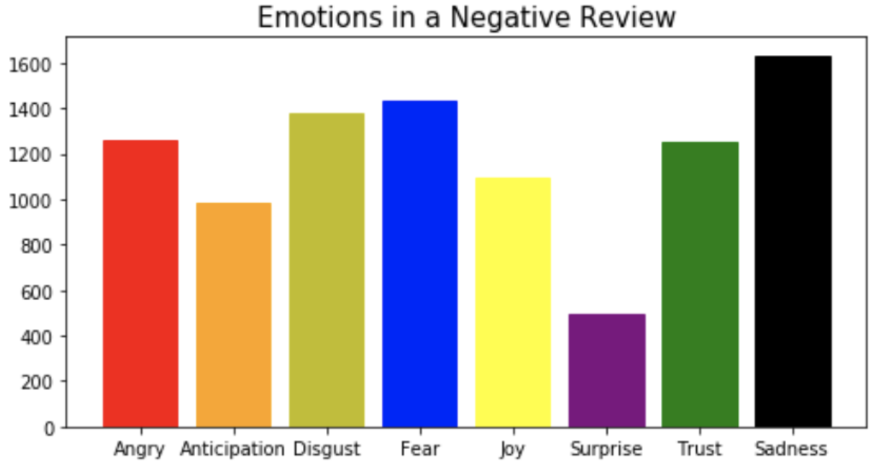
DataFrameMapper():
https://pypi.org/project/sklearn-pandas/1.5.0/

APPENDIX C


Emotions in a Somewhat Positive Review


Emotions in a Somewhat Negative Review


Emotions in a Somewhat Positive Review


Emotions in a Negative Review