Mini Project 1

Alejandra Zambrano

October 2022

1. Dataset Analysis

The data set involved in today's project is called the GoEmotion data set, this data set consists on 58k comments the popular social network Reddit, this platform lets people with similar kinds of interest connect and share their opinions online, which makes an amazing platform to collect data for NLP processing. In this case we are going to focus on emotion and sentiment classification, in particular, of 27 emotion categories identified there is 12 positive, 11 negatives, 4 ambiguous emotion categories, and 1 "neutral" emotion and sentiment categories.

Something which clearly needs to be taken into account from the beginning is how this classes are weighed, meaning how many examples of each classification we have. It's possible to see in the graphs.pdf document that this classes are heavily imbalanced. The count for "neutral" sentiment classified post consists of 32 % of the whole data set, while the second biggest group is "approval" with only 6 % and "grief" being the smallest one with 0.2 %. For the sentiment analysis the classes are not as imbalanced, also due to the fact is only 4 classes, being "positive" sentiment the most predominant and "ambiguous" the lowest.

Working with imbalanced classes can be a big issue when in comes to Machine Learning Models, it normally gives us a false perception of the metrics and performance of the model, due to fact it tends to over fit on the classes that are more predominant, this leads to a poor generalization process. The most affected metric by imbalanced classes is the accuracy, since it only takes into account the percentage of right answers of your model, treating all classes as if they were equal. Evidently for an imbalanced class the accuracy value can be really high but not exactly meaning is a good model. Knowing this, is better to take into account some better and more popular metrics for imbalanced classes, like F-Measure and the confusion matrix to see the performance for each class. In this case we are going to mainly focus on the micro average F1-score to analyze the classes due to the imbalance and the fact that is a multi-class model.

2. Analysis of results

Something to state out is that generally in this project, metrics tend to not be very high for all models. The fitting for this model may not be as good due to data set itself, like duplicated entrances (in this case almost 67 % are duplicated), slang or unusual words, specially when there are that many classes.

2.1. Multinomial Naive Bayes

2.1.1. Emotion classification

Naive Bayes is often used for document classification and NLP due to it's rapid execution and high performance. We can observe that with the default model and the Count2Vectorize we get a f1-score (micro-avg) of 0.39, surprisingly for the tf-idf model the micro-avg went a little lower with a value of 0.37 but with almost a 1.0 in recall score for the neutral class. In general with default values the model does a really good job classifying the neutral class (which was expected) we can see that with a little of hyper parameter tuning the model doesn't get that much higher f1-scores but is able to generalize better. The individual metrics for each class went up. A strong example of this is the class "embarrassment" the precision value went from 0.38 to 0.78. Another thing that I found unusual is the metrics for the gratitude class, even if gratitude is only the sixth most predominant class the metrics are pretty high compared to the to the other classes (more than 0.7). In general the highest f1-score is of 0.39 for this model.When using tf-idf the metrics drastically change, the precision increases but recall has really small values, this could due to the fact that tf-idf gives a lot of weight to ïmportant"words for certain classifications.

2.1.2. Sentiment classification

Something similar to emotion happens with the sentiment classification. It does a really good job classifying "positive" sentiment (the most predominant class) and not that good of a job for the "ambigous" class. We can infer, by the value of the recall, that our model tends to avoid classifying things as ambiguous and most of the examples are classified as neutral. Even after the hyper parameter tuning the metrics don't change a lot from class to class (almost the same), so in the case for the sentiment analysis the default values can be kept.

2.2. Decision Tree Classification

2.2.1. Emotion classification

To my amazement this worked better than I expected, normally I wouldn't use decision tree for NLP, because understanding language or feelings does not follow any specific "rules", which is the principal idea for Decision Trees (based on something we make a decision). In general this model gives a lower f1-score than the other models but is not terrible. The biggest issue is how complex it can get, in case of the default decision tree for the sentiment it was a depth of 1579 and for sentiment 1421 (that's huge!). Something definitely unique about this model is that it doesn't do such a good job with the neutral classification as the other models and instead does better for the class gratitude, this could be due to the fact that gratitude post have an exceptional use of words that makes them easy to classify, for example phrases like "thank you" automatically goes to "gratitude" which makes it really easy for this type of model. After some tuning we can see that the precision is much better but the recall values go really down,this basically means that the model doesn't classify that much those rare classes and focuses more on the predominant ones. In this case, different from MNB CountVectorizer, it does work better with tf-idf.

2.2.2. Sentiment classification

In terms of metrics the sentiment classification is pretty similar to the other models studied in this project. We can see that there is a small improvement when comparing Naive Bayes and Decision Tree, but it's almost imperceptible and minimum when you take into account how much more complex this model is. In general similar to the other models, we get a f1-score of 0.55 (not great but not terrible).I would recommend to only use decision tree for very specific types of problems.

2.3. MultiLayered Perceptron

2.3.1. Emotion classification

The MLP same as the Decision Trees are quite complex and time consuming, but in this case is more worth it for the metrics. We can see right away with the default MLP, the f1-score is of 0.44 which is the highest we have gotten till now. In general there is a better classification for all the classes, for example, “pride” that was never classified with NB and rarely classified with DT, in this model has a precision of 0.5 (but not such a good recall). For most of the classes the recall value may be lower than DT but precision is much higher. For this type of problem, since is not critical, is not necessary that the errors are highly penalized, so good precision should be more relevant. After the tuning there was not a valuable difference so in this case the default model it’s sufficient and actually better.

2.3.2. Sentiment classification

At this point is clear that between models there is not such a significant difference when it comes to sentiment classification (probably because of the class number being small)

2.4. Own Exploration

For this area I decided on two very popular pretrained models. The first one is the glove-wiki-gigaword-50 data set, this data set is based on 2B tweets, 27B tokens, 1.2M vocab, uncased. In general we can see that our model doesn’t show very good metrics, something really important to take into account of this models is the difference in the vector sizes, in case of the google data set, vectors where of size 300, in this case the vector size is only 50 which clearly affects how much information one vector can give and can negatively affect the metrics. Is also very important the type of data that was used for this pretrained model, twitter can be a great place to get data but can also be very damaging since big percentage of tweets are bots, spam or inappropriate content.

3. Reponsabilities

Alejandra Zambrano - Everything