IST707 – Applied Machine Learning

My Research Project Title: EMOTION CLASSIFICATION

CHAITANYA KUNAPAREDDI

ckunapar@syr.edu

Abstract

Emotion Recognition is a common classification task. For instance, given a tweet, you create a model to classify the tweet as being either positive or negative. However, human emotions consist of myriad emotions and cannot be constrained to just these three categories. On the contrary, most of the datasets available for this purpose consist of only two polarities — positive, negative, and at times neutral. These emotions are necessary to identify to solve problems related to human understanding or response to a given circumstance such as feedback of service or a product especially useful for tech companies, marketing agencies, fashion brands, media organizations, and many others. So, we have 6 emotions namely sadness, anger, love, surprise, fear, and joy. This project will help us understand consumer perception.

EMOTION CLASSIFICAITION

In this article, we'll get to know the background of data collection and explore it a bit. The emotion dataset comes from the paper CARER: Contextualized Affect Representations for Emotion Recognition by Saravia et al. The authors constructed a set of hashtags to collect a separate dataset of English tweets from the Twitter API belonging to eight basic emotions, including anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. The data has already been preprocessed based on the approach described in their paper. The dataset is stored as a pandas data frame and ready to be used in an NLP pipeline. The distribution of the data and the list of hashtag examples for each emotion are provided below.

Some people believe that business has no place for emotions, but this couldn't be more wrong. Emotions play a huge role in business and in the success of your business. Emotions have an effect on teamwork, customer satisfaction, manager-employee relationships, and employee retention. Plus, the brain's emotional state affects decision making, planning, and negotiating, and creative thinking. More so than this, emotions have a huge impact on the number of sales you make, the amount of money you make and overall whether your business makes it or breaks it.

It probably comes as no surprise that customers report a higher satisfaction rate when a staff member really seems to understand their problem. But more than customer service, this is also the same for colleagues working together as when people feel that other people understand them, they work better together and create a longer lasting and stronger team.

Hypothesis 1: Predicting the emotion behind a tweet

Method

We would be focusing on the below mentioned process to achieve our agendas.

1. Exploratory data analysis
2. Data pre-processing
3. Modelling
4. Predictions
5. Performance evaluation

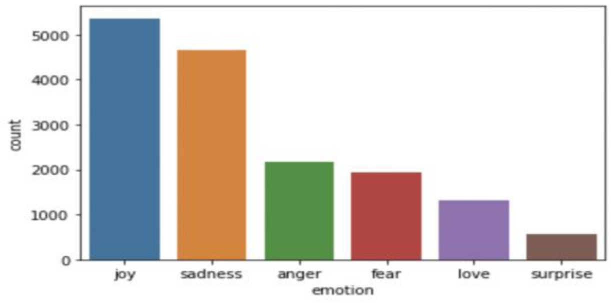
#Exploratory Data Analysis

In this section we are going to have a glance of our data set and understand its structure. I have taken a dataset of 15999 records of data. Once we have the data set, we must look for nulls in the data, if exist we must remove them since this is a textual data, we cannot give any calculated or average value.

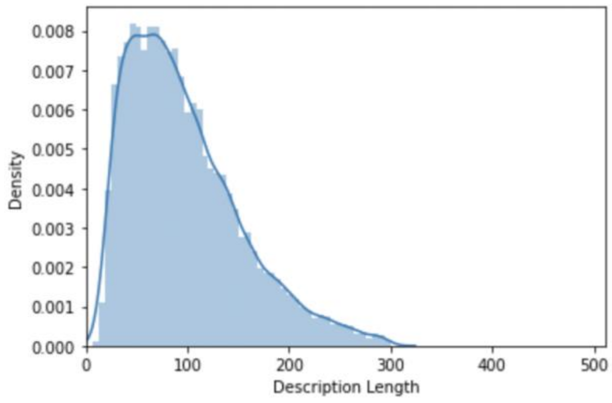
we see we have 2 columns, the first column is the text description from where we need to convert them to vectors and predict the label which is the second column. Our main aim is to predict the emotion behind the tweet. We have 6 unique labels namely joy, sadness, anger, fear, love and surprise.

Column	Row length	Null count	Dtype
desc	15999	No null	Object
Emotion	15999	No null	Object

The above image shows that we have 15999 records which are not null objects of a data frame. Let's now explore the charts that we have made using given data.



As we see in the above count plot, we have jotted the count of records with each label. We have a very uneven dataset. Most of the records are for joy followed by sadness, there are very few records for anger, fear, love, and surprise. This is going to affect our modelling as it would give us biased output. If there is any intersection of vector for different labels it would predict the wrong label.



Next, we have plotted the description length for each record i.e. for each document, this would prove that we have equality in our data set no description is way too longer to have a varied corpus of data.

#Data Preprocessing

In this section we are going to clean up the text data that we must make the best use to create a corpus of data by converting the texts to vectors. This first and foremost step is to convert each document to sentences and then break them into tokens. Now it's easy for us to check for words that add immense value and those that are useless, we now remove all punctuation marks as they don't add value and would take unnecessary space in our sparse vector matrix. Next, we remove all the stopwords, are the commonly used words and are removed from the text as they do not add any value to the analysis. These words carry less or no meaning. NLTK library consists of a list of words that are considered stopwords for the English language. Some of them are : [i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he, most, other, some, such, no, nor, not, only, own, same, so, then, too, very, s, t, can, will, just, don, don't, should, should've, now, d, ll, m, o, re, ve, y, ain, aren't, could, couldn't, didn't, didn't]But it is not necessary to use the provided list as stopwords as they should be chosen wisely based on the project. For example, 'How' can be a stop word for a model but can be important for some other problem where we are working on the queries of the customers. We can create a customized list of stop words for different problems.
It is one of the most common preprocessing steps where the text is converted into the same case preferably lower case. But it is not necessary to do this step every time you are working on an NLP problem as for some problems lower casing can lead to loss of information.

We can perform all the above steps in one step including conversion of texts to vectors by using count vectorizer from sklearn package.

#Modelling

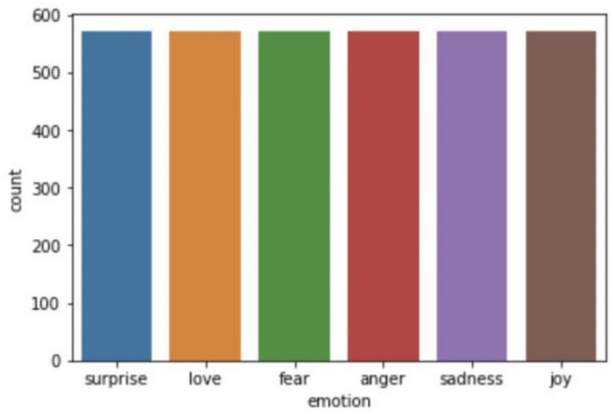
Since we have multi class labelling involved I have chosen 5 different algorithms namely, naïve bayes, decision trees, random forest, SVM and KNN. The reason I have chosen naïve bayes is that It is easy and fast to predict class of test data set. It also performs well in multi class prediction. When assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression and you need less training data. Also, to use decision trees one way of doing it could be to assign a unique integer to each of our classes. Since Random Forest can inherently deal with multiclass datasets, A family of methods that can handle large amount of training data efficiently and that are inherently suited for multi-class problems are based on random forests. The main advantage of KNN over other algorithms is that KNN can be used for multiclass classification. Therefore if the data consists of more than two labels or in simple words if you are required to classify the data in more than two categories then KNN can be a suitable algorithm. In its most simple type, SVM doesn't support multiclass classification natively. It supports binary classification and separating data points into two classes. For multiclass classification, the same principle is utilized after breaking down the multiclassification problem into multiple binary classification problems.

After applying the algorithms with default parameters to our data we obtain the following results.

Algorithm	Accuracy	Precision	Parameters
Naïve bayes	77.1	79.7	-
KNN	44.7	51.09	3 neighbors
Decision Trees	87.2	-	5 min sample leaves
Random Forest	88.2	88.2	100 estimators
SVM	86.9	86.9	Linear kernel

We see that svm, random forest and decision trees have performed great! But can we rely on these results? certainly not as we have improper data and we have also not used the algorithms to their finest best.

#Dealing with imbalanced data
In the earlier section we have worked on data modelling with imbalanced data, in this section we are going to solve that problem by considering a sample of data from every label. This can be achieved by splicing the data for each label and picking equal data in pandas itself. Finally we concatenate all the records to one data set.



As we see we have now taken 572 records from each label and so our 15999 records of data are now reduced to 3432 records. Now we can expect the results to be fair.

#Hyper parameters tuning
This is the last section of our project where we will be focusing on another important aspect, i.e., selecting the best fit parameters. We may not be using the right parameters while modelling, so by trial and error, we are going run a range of values and choose the best fit parameters and their results.
We are going to use the grid search technique to choose the best fit parameters for the model. GridSearchCV is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

Algorithm	Accuracy	Precision	Parameters
Naïve Bayes	71.9	71.9	Alpha = 1
SVM	80.3	80.7	C = 10
Random forest	86.7	86.9	Max-depth = none, min_sample_leaf = 2, n_estimators = 800
Decision Trees	30.9	37.3	Criterion = entropy, Max_depth = 15
KNN	40.7	45.5	N_neighbours = 3

As we see the results, we have done a lot of fine tuning. Let's deep dive into each model, starting with naïve bayes we have taken a range of alpha values like 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, after running an iteration of values we have finally chose the alpha value of 1 to give us the best fit of 71.9 % accuracy. Secondly SVM model, we have taken a wide range of cost function from 0.001,0.1, 0.5, 1,10 and the cost function of 10 has given the best fitting results for SVM with 80.3% accuracy. Since random forest has broader range of values we will be iterating over many parameters. We will be looking into n estimators, max depth, and min sample leaf. We have 200 to 1000 values of n estimators with 5 steps, also min sample leaf with range of 1,2,4 and finally max depth of 10 to 100 with step of 5, the best fitting values give 86.7% accuracy. When it comes to decision trees we will match features of criterion, max depth. We have 2 levels of criterion, which are entropy and gini, and we are considering a random list of depth values 5,7,9,11,13,15. The best parameters give us a poor accuracy of 30.9%. Finally, we have the k neighbors and we have taken a range of neighbors from 1 to 61 with 3 neighbors giving us the best value with 40.7% accuracy

References

Datacamp community posted nltk sample examples - https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

H Avetisyan, O Bruna, J Holub
Overview of existing algorithms for emotion classification. Uncertainties in evaluations of accuracies.
https://iopscience.iop.org/article/10.1088/1742-6596/772/1/012039/pdf