Emotion-Intensity

ABSTRACT:

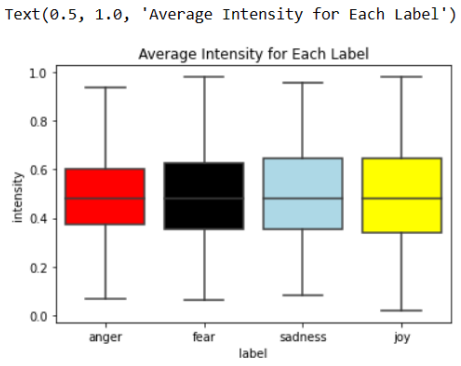
The task given is to predict the intensity of tweets, given the tweet's text and the main emotion. The emotions are divided by four : Anger, Fear, Sadness, Joy. The intensity are graded from 0 to 1, where 0 means the lowest intensity and 1 the highest. There are three data given : the training data, development data, and test data. We can use the training and development data to train the regression models to predict the intensity of the test data.

DATASET:

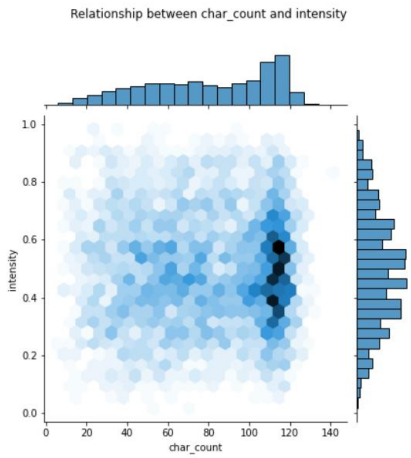
Different dataset are provided for train, dev and test. In each of these dataset, there are four collection , each belong to the emotions under consideration.

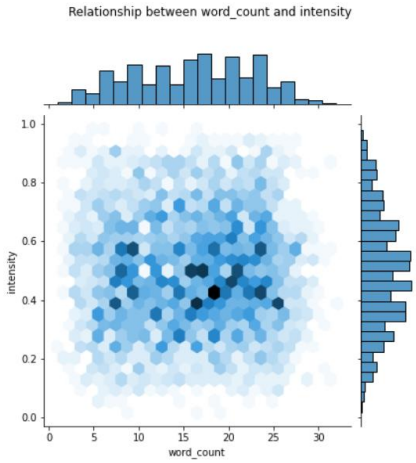
EDA:

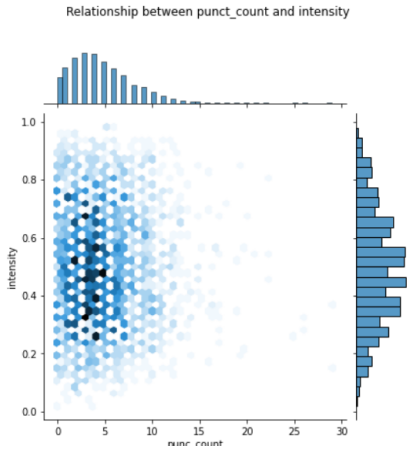
- Average Intensity of each word



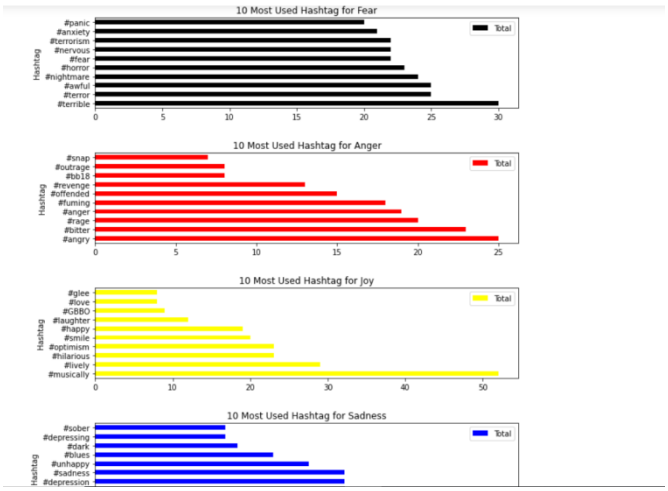
- See the relationship between intensity and character, word, and, punctuation counts



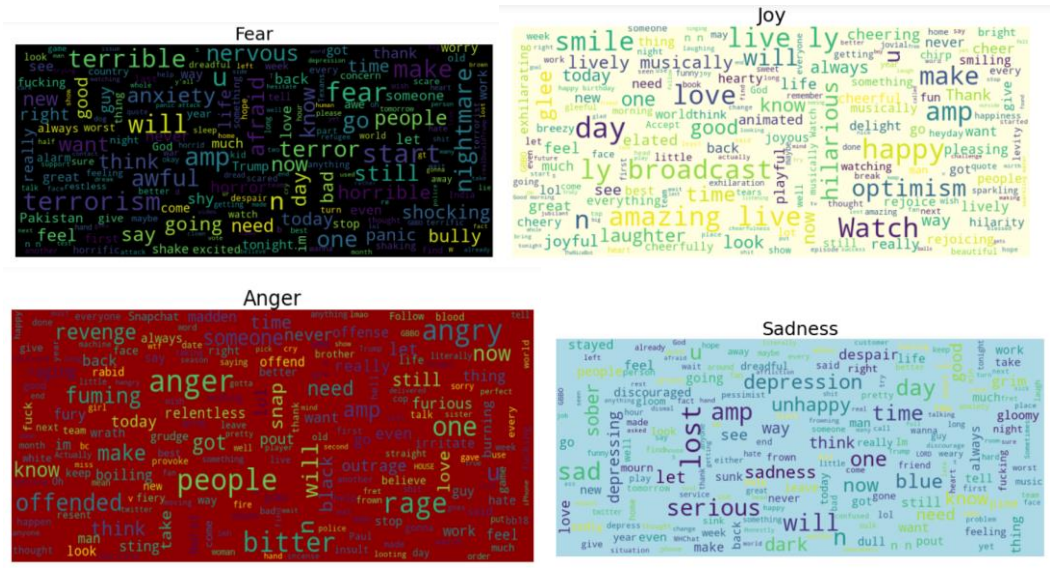




- Show 10 most used hashtag for each emotion



- Word Cloud for each emotion



From the results of the Exploratory Data Analysis, I see that there is no significant difference between the average intensity for the four emotions. It also appears that there is no significant pattern that indicates the relationship between intensity and the number of characters, words, and punctuation. Therefore, I don't use the char_count, word_count, and punct_count features to train the model.

PROPOSED SYSTEM:

Two different approaches are explored for the task. One based on regression models and another based on a deep learning model.

1. **Input Features**

Text from each tweet is cleaned by removing links ,stop words, , numbers and transforming all alphabets to lower case. The words are tokenized and Tf-idf features are extracted from them. TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. For each tweet a feature vector of size 1000 is extracted. Categorical labels are assigned to each of the four emotions. This label is also concatenated to the feature vector. Finally, for each tweet a 1004 long feature vector is obtained.

2. **Regression model**

The algorithms that I try to train the regression model are linear regression, ridge regression, Bayesian, KNN regression, SVR, and decision tree regressors. I also tried using vectorization with Bag of Words and tf idf techniques. I use training data to train models, and data development to test models and see which models produce the best results. It turned out that the SVR model with the tf idf vectorization produced the lowest error, so I chose to make the final model.

The SVR implementation of Support Vector Regression from scikitlearn is employed here. Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

3. **Deep Learning Model**

Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model gives out a prediction. The weights are adjusted to find

patterns in order to make better predictions. Deep learning model with four hidden layers is employed. The first three dense layers have sizes 1004, 512 and 64 with reLu activation. Dropout layers with 0.2 as parameter are provided between them. Adam optimizer is used and Mean Average Error as the loss. Sigmoid activation is given for the output layer.

4. Results

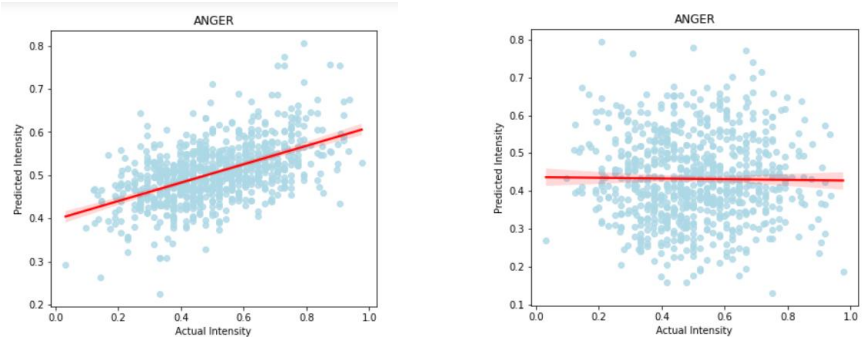


Fig. 1. SVR vs Deep model ANGER

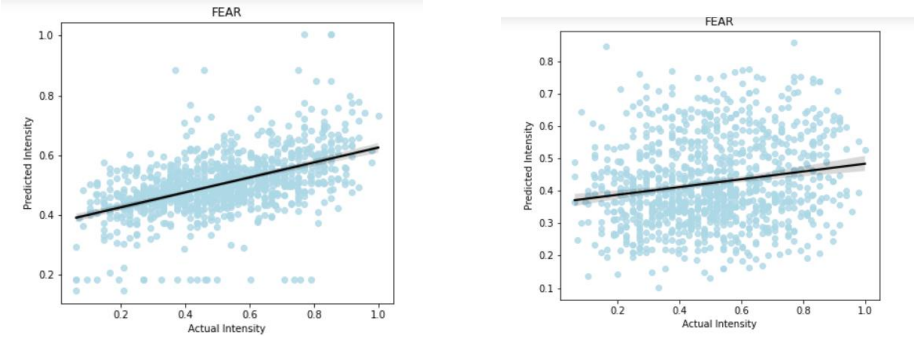


Fig. 2. SVR vs Deep model FEAR

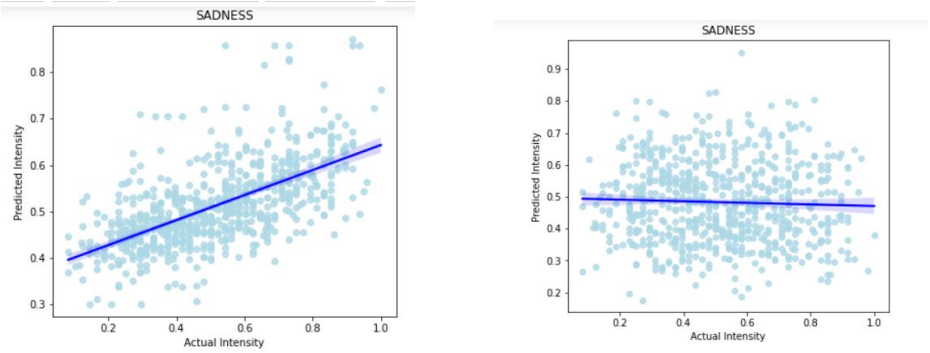


Fig. 3. SVR vs Deep model SAD

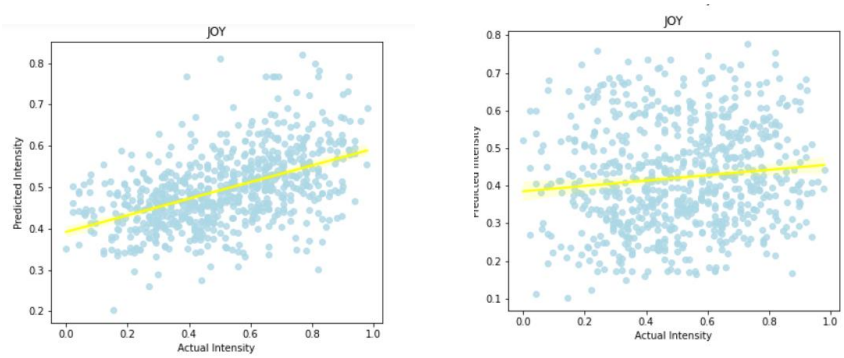


Fig. 4. SVR vs Deep model JOY

Each emotion has its own model. From the plot and assessment metrics for each model, it can be seen that the four models produce better results than if the intensity was chosen randomly. This can be seen in the regression line, where on average if the real intensity increases, the intensity of the predictions also increases. It can also be seen that the anger model produces the lowest errors, while the joy model produces the highest errors.

The Pearson Correlation results also show a correlation of more than 0, which means there is a linear positive relationship between predicted and actual intensity. One thing that is interesting is that the Pearson Correlation on average shows lower results when gold scores are only taken between 0.5-1. This shows that the model works worse when the actual intensities of the predicted tweet is more than average.

But of course the prediction results are still far from perfect. It can be seen that the gradient of the four regression lines of the model has a gradient that is too low, where tweets with very low intensity are usually predicted to have higher intensities, and tweets with very high intensity are usually predicted to have lower intensities. It can also be seen on the plot that there are still many points that are far from the original intensity, such as in the fear model, where some tweets with high intensities are predicted to have low intensities. The Pearson Correlation of the joy model is also quite low when the gold score is between 0.5-1 (0.298). This proves that the Joy model cannot accurately predict tweets with high intensity.

There are several ways that I think can improve the results of the model. The first way is to correct the words in the tweets that are spelled incorrectly, so that the vectorization process will produce results that are more in line with reality. The second way is to increase the number of train tweets, because surely there are still many words contained in the test data that are not in the data train. The third way is to make models with different algorithms for each emotion, because maybe an algorithm works well on one emotion and less well on another emotion.