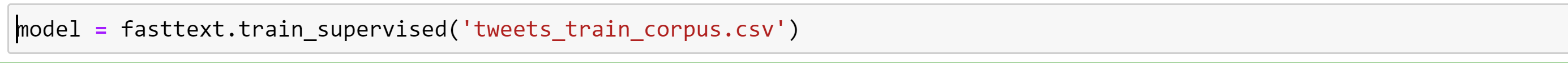
**FastText:**

FastText is one of the efficient libraries used for text classification and word embedding. This is based on skipgram model, where each word is represented as a bag of character n-grams. A vector representation (embedding) is associated to each n-gram for a word. Thus, we can represent a word by the sum of the vector representations of its n-grams. ​ By leveraging n-grams from individual words based on their characters, there is a higher chance for rare words to get a good representation since their character based n-grams should occur across other words of the corpus.

**Text Classification:**​

FastText library provides a pre-built model for text classification for both Supervised and unsupervised. Here we train a supervised model for text classification for our tweets. While preparing the data for the model we make sure each label is prefixed with \_\_label\_\_ through which fastText recognizes what is a label and what is a word. Each line contains the \_\_label\_\_ followed by the tweet. The model is then trained to predict the labels for the tweets given. We split the data into 80%-20% for training and testing data. We have to create two separate file for training and testing and train our model with the training data. Below is the command used to train the model.



After testing the model we achieved a score of 0.89.

**Data Preprocessing:**

To improve the score we have performed Preprocessing on the data by removing all the punctuations, stop words, Retweets, Hashtags, @names which then reduced our score to 0.85. After few hit and trails we observed that Hashtags and @names perform import role in classification of the tweets. Therefore, we had removed Hashtags, @names in the preprocessing step which increased our score to 0.95.

**Tuning the Parameters:**

Fastext represents a word by bag of character of N-grams. Therefore, we tuned the parameter value for the model by varying the n-values from 2 to 6 also changing learning rate and number of epochs. By this we achieved a score of 0.936. Comparison of the measures with their techniques is shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Before Preprocessing | After Preprocessing | After Preprocessing  (Including Hashtags and @Names in tweet) | After tuning the parameters  (wordNGrams=2, lr=0.7) |
| Precision | 0.891 | 0.850 | 0.929 | 0.936 |
| Recall | 0.891 | 0.850 | 0.929 | 0.936 |
| F1 | 0.891 | 0.850 | 0.929 | 0.936 |