**Sentiment classification for social media**

1. **Preprocessing**

**1.1 Lowercase:**

I lowercased the tweets as opening the file.

**1.2 URL:**

From text classification point of view, a particular URL is not important. However, existent a URL can be an important feature. So, I replaced hyperlinks with URL by using Regular Expression ‘https?://[\S]+’

**Hashtags (#) and Handles (@):**

Hash symbol (#) is used to both naming subjects and phrases that are currently in trending topics and handles always be followed by user name. I changed them to ‘hash\_\1’ and ‘hndl\_\1’ by using Regular Expression ‘#(\w+)’ and ‘@(\w+)’.

**1.3 Repeating Characters:**

People often use repeating characters, like huuuuuungry. I replaced these characters repeating more than twice as two characters by using Regular Expression '(.)\1{1,}' and Replace Expression ‘\1\1’.

**1.4 Punctuations and characters:**

To lemmatize and tokenize contents, I removed all non-alphanumeric characters except spaces, words with only 1 character, and numbers that are fully made of digits by using Replace Expression '[^a-z0-9 ]+' , '\b[a-z]\b', '\b[0-9]+\b' respectively.

**1.5 Tokenization and lemmatization:**

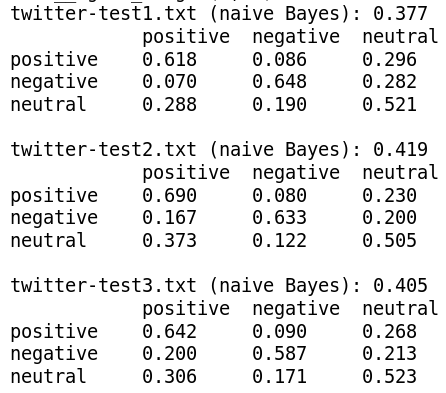
I took advantage of some methods (nltk.word\_tokenize(), nltk.pos\_tag(), nltk.WordNetLemmatizer.lemmatize()) from package ‘nltk’ and defined a ‘get\_wordnet\_pos’ function.

1. **Classifiers:**

To shorten testing time, I created pickle files to save my pretrained models.

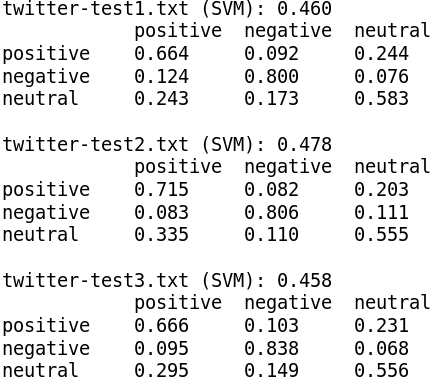
* 1. **Naïve Bayes Classifier:** **sklearn**

In this section, I utilized Pipeline class to combine vectorizer, transformer and classifier. In the classifier, I used MultinomialNB() to train data and grid search to tune parameters. The best parameters in this module are choosing unigram in CountVectorizer(), False in TfidfTransformer(), alpha=0.1 in MultinomialNB(). Finally, the performance of Naïve Bayes Classifier is as following in test sets:



* 1. **SVM Classifier:**

I chose the similar method as the Naïve Bayes classifier. I used SGDClassifier() to train data and grid search to choose parameters. And the best parameters in this classifier are trigram in CountVectorizer(), True in TfidfTransformer() and alpha = 0.001 and penalty = ‘l2’ in SGDClassifier(). Results in test sets are:



* 1. **LSTM Classifier:**

