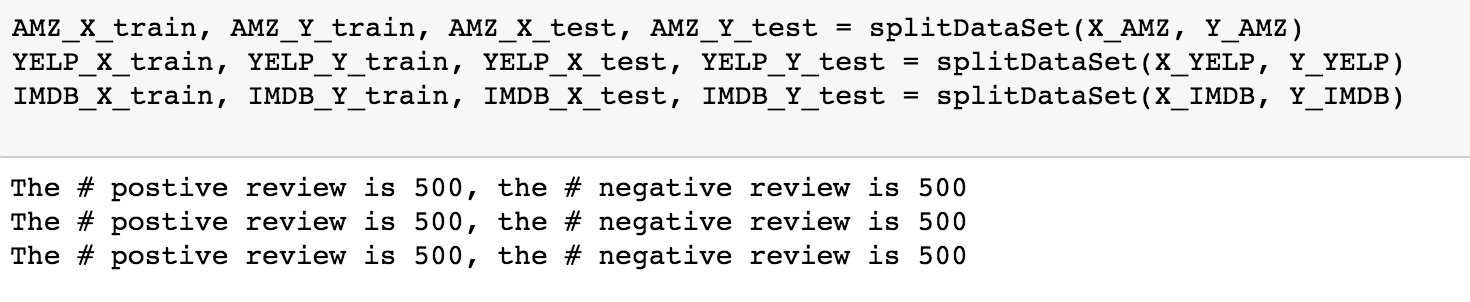
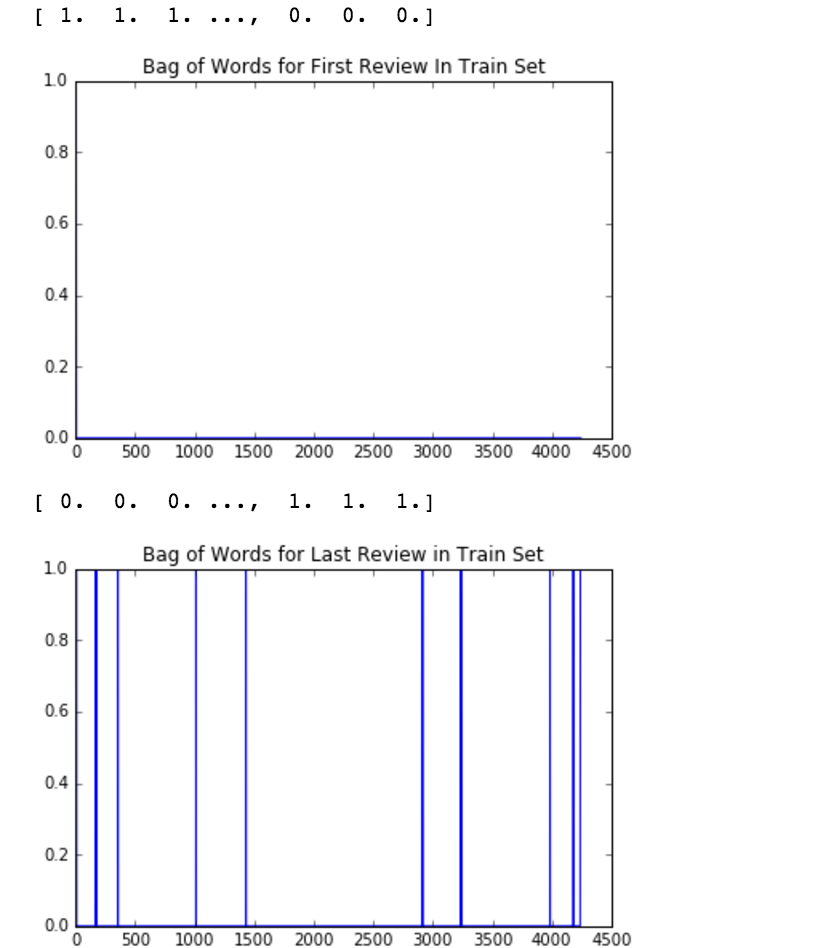
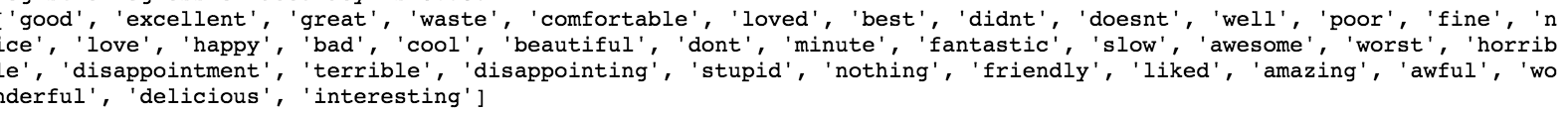
* All those labels are labeled as shown in the graph below, so it is balanced. In parseInFile(name), basically it reads in the file and spilt the reviews and labels by ‘\t’ and ‘\n’. Although preprocessing is not in this step, this step removed punctuation and converted word to lower case for future respective. 
* For preprocessing strategy, I used all. In previous steps, punctuation and lower case is already done. While building the word\_dict, it stripped stop words by checking if this word is in the set(stopwords.words('english')).

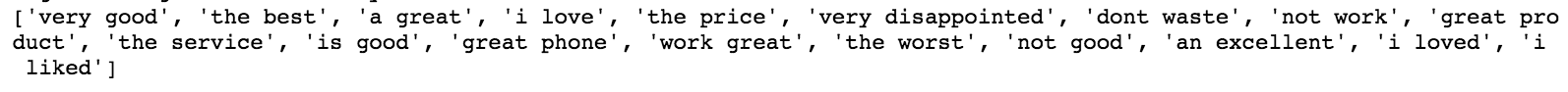
(Note: if you do want to re-run the jupyter notebook, make sure you install nltk.download() which I commented out in the first cell)

The reason why we want to do lowercase is we do not want to label same word at different labels(such as “good “ vs ”GOOD). Punctuation and stop words are useless in the bag of words model.

* Since it is impractical to show the full dimension of the feature vectors, two feature vector was plotted.



* l2 normalization was used in post processing strategy due to lots of 0 in the feature vectors. Compared to l1, L2-norm squares the the model will see a much larger error than the L1-norm, so the model is much more sensitive to this feature vector. Moreover, L1-norm has the property of producing many coefficients with zero values or very small values with few large coefficients, where L2-norm can avoid this problem.
* words that important for Sentiment prediction: 
* words that important for 2-gram:



* K-means perform around 50% in the bag of words, 2-gram and PCA. It is basically same as random guess. There are too many features and the vectors are so sparse in the high dimension, the curse of dimensionality is validated in this case. The logistic regression performs around 72~76%. Because it considers all the feature vectors and sensitive to all changes. However, with higher reduction of PCA, the logistic performs worse due to reduction in number of features for training logistic regression.