**Machine Learning - Comp 551 – Mini project 3**

Rony Azrak – 260606812

*YELP*

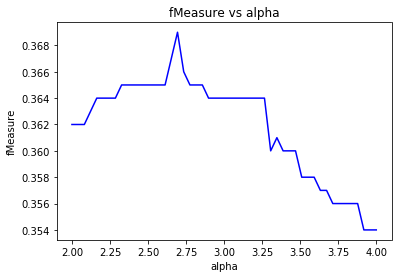
We will report the F-Measure for the different models that we use. There are 5 classes to classify in for this dataset.

**Random Classifier**: 0.205

**Majority-class classifier**: 0.351

**Naïve Bayes classifier – Binary**: For this classifier we tuned the alpha parameter. Based on the validation set, we found the best alpha to be 2.694 and that would give our test set an F-Measure of 0.363

First we tried alpha values between 1e-5 and 50 with a step size of 1, then we used a smaller step size around where the max F-Measure value was. So the final range was 2 to 4, evaluating 50 alpha values in that range.



**Naïve Bayes classifier – Frequency**: No tuning was used for this classifier. The F-Measure was 0.286, which is a pretty big decreases compared to using a binary bag of words which had gave us an F-Measure of 0.364.

**Decision Tree classifier – Binary**: We tuned the parameters for criterion and splitter, iterating the former between ‘gini’ and ‘entropy’, and the latter between ‘best’ and ‘random’. We used the cross product of the 2 parameters, so we had 4 total iterations for parameter tuning. The validation set helped us find the best parameters to be criteria ‘gini’ and splitter ‘random’, which gave our test set an F-Measure of 0.322.

**Decision Tree classifier – Frequency**: We tuned the same parameters with the frequency bag of words. The validation set helped us find the best parameters to be criteria ‘gini’ and splitter ‘random’, which gave our test set an F-Measure of 0.289.

Just like with the Naïve Bayes classifier, the frequency bag of words gave us a worse F-Measure than with using a binary bag of words, 0.289 for frequency vs 0.322 for binary.

**Linear SVM – Binary**: We tuned the parameters for penalty and loss, iterating the former between ‘l1’ and ‘l2’, and the latter between ‘hinge’ and ‘squared\_hinge’. We used the cross product of the 2 parameters, but penalty ‘l1’ could not get combined with a loss ‘hinge’ or ‘squared\_hinge’ so we had 3 total iterations for parameter tuning. The validation set helped us find the best parameters to be penalty ‘l1’, which gave our test set an F-Measure of 0.367.

**Linear SVM – Frequency**: We tuned the same parameters with the frequency bag of words. The validation set helped us find the best parameters to be penalty ‘l2’ and loss ‘squared\_hinge’, which gave our test set an F-Measure of 0.391.

Unlike the other classifiers, the frequency bag of words gave us a better F-Measure than with using a binary bag of words, 0.391 for frequency vs 0.367 for binary.

**Comments**: It looks like Linear SVM gives us the best results, based on the F-Measure. We saw that in most cases, knowing the frequency of words does not help us much more than simply knowing if a word is present or not. For the Linear SVM classifier knowing the frequency does help, so it could be that the other models don’t make use of that advantage as well as Linear SVM. It is possible that hyper-parameter tuning was not optimally made and if it was, could generate better results with a frequency bag of words.

Rony Azrak – 260606812

*IMDB*

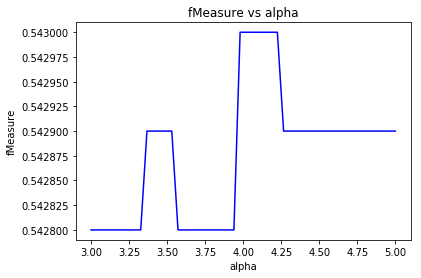
We will report the F-Measure for the different models that we use. There are only 2 classes to classify in for this dataset, so we should generate a better score overall.

**Random Classifier**: 0.501

**Majority-class classifier**: 0.500

**Naïve Bayes classifier – Binary**: For this classifier we tuned the alpha parameter. Based on the validation set, we found the best alpha to be 3.980 and that would give our test set an F-Measure of 0.551.

First we tried alpha values between 1e-5 and 50 with a step size of 1, then we used a smaller step size around where the max F-Measure value was. So the final range was 3 to 5, evaluating 50 alpha values in that range.



**Naïve Bayes classifier – Frequency**: No tuning was used for this classifier. The F-Measure was 0.516, which is a pretty big decreases compared to using a binary bag of words which had gave us an F-Measure of 0.551.

**Decision Tree classifier – Binary**: We tuned the parameters for criterion and splitter, iterating the former between ‘gini’ and ‘entropy’, and the latter between ‘best’ and ‘random’. We used the cross product of the 2 parameters, so we had 4 total iterations for parameter tuning. The validation set helped us find the best parameters to be criteria ‘gini’ and splitter ‘random’, which gave our test set an F-Measure of 0.540.

**Decision Tree classifier – Frequency**: We tuned the same parameters with the frequency bag of words. The validation set helped us find the best parameters to be criteria ‘entropy’ and splitter ‘best’, which gave our test set an F-Measure of 0.540.

The frequency bag of words gave us a the examt same F-Measure than with using a binary bag of words.

**Linear SVM – Binary**: We tuned the parameters for penalty and loss, iterating the former between ‘l1’ and ‘l2’, and the latter between ‘hinge’ and ‘squared\_hinge’. We used the cross product of the 2 parameters, but penalty ‘l1’ could not get combined with a loss ‘hinge’ or ‘squared\_hinge’ so we had 3 total iterations for parameter tuning. The validation set helped us find the best parameters to be penalty ‘l2’ and loss ‘hinge’, which gave our test set an F-Measure of 0.556.

**Linear SVM – Frequency**: We tuned the same parameters with the frequency bag of words. The validation set helped us find the best parameters to be penalty ‘l1’, which gave our test set an F-Measure of 0.617.

Unlike the other classifiers, the frequency bag of words gave us a better F-Measure than with using a binary bag of words, 0.617 for frequency vs 0.556 for binary.

**Comments**: Based on the F-Measure, it looks like Linear SVM again gives us the best results. Just like with Yelp, we saw that in most cases, knowing the frequency of words does not help us much more than simply knowing if a word is present or not. For the Linear SVM classifier knowing the frequency does help, so it could be that the other models don’t make use of that advantage as well as Linear SVM. It is possible that hyper-parameter tuning was not optimally made and that if it was, could generate better results with a frequency bag of words.

So Linear SVM seemed to be our best classifier for both datasets. When we have extra information like the frequency of words, Linear SVM also worked best at improving the F-Measure with that extra piece of information.