BITS F464

MACHINE LEARNING

ASSIGNMENT 1B

Naive Bayes Classifier

Brief Description of the Model and Implementation:

Naïve Bayes classifier applies to learning tasks where the class output is dependent on a set of attribute values together. A set of training examples is provided, and a new instance is presented, described by a tuple of attributes <a_1,$a_{2,}$…,a_n>. The task of the classifier is to predict the target value, or class, of the new instance.

The Bayesian approach to classification is purely probability based. The class that is predicted, for a test instance, by the classifier is the one that is most probable, given the tuple of attribute values. If v_m is the predicted class,

$$v_m = \text{argmax } P\ (v_j \mid <a_1,a_2\ldots,a_n>),\ \text{ for every class } v_j$$

Using Bayes Theorem, this probability for each class is

$$P(v_j|\langle\, a_1, a_2, \ldots, a_n\rangle) \ = \ \frac{P(<a_1,a_{2,\ldots,}a_n>|v_j)\times P(v_j)}{P(<a_1,a_{2,\ldots,}a_n>)}$$

After calculating all class probabilities from the above, the classifier will assign the highest probability among all class probabilities to the test instance.

It is not feasible to calculate $P(< a_1, a_{2,\ldots,} a_n >|v_j)$ for every test example, so the ***naïve*** Bayes classifier assumes that all attribute values <a_1,$a_{2,}$…,a_n> are independent of each other.

This assumption simplifies the required probability for each class to the following-

$$P(v_j|\langle\, a_1, a_2, \ldots, a_n\rangle) \ = \ P(v_j) \times \prod_i P(a_i|v_j)$$

In the case of text classification, this loosely translates to “given class *j*, what is the probability of this word (attribute value) occurring?”

This probability is much easier to calculate. Once we have all such probabilities for a word, the largest one out of them is the predicted class output.

The training dataset provided has 1000 examples belonging to either of 2 classes, 0 or 1. The examples are in the form of 1000 sentences; the last character in each sentence denotes the class that a particular example belongs to.

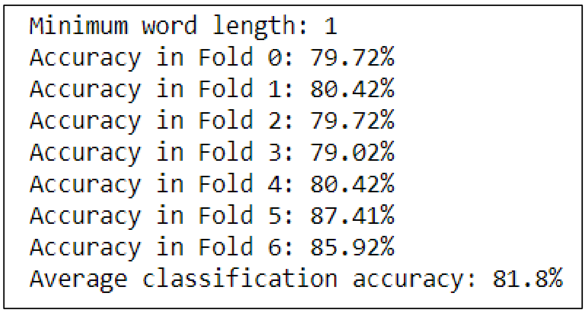
In pre-processing the dataset, the regular expression library ***re*** has been used to eliminate words that contain non-word symbols. If the remaining words have any symbols or numbers, defined in *puncs* and *numbers*, those words are also removed. Additionally, all words are transformed to lower case. This removes the distinction between occurrences “These” and “these” that the classifier may treat as distinct but whose difference does not contribute significantly to the successful prediction. This action will not have a serious impact on our results.

k groups are created for k-fold cross validation. We are required to use 7-fold cross validation to train and test the data. The entire dataset is shuffled and divided it into 7 groups. In each of the 7 iterations, 1/7th of the 1000 examples (= 142 or 143) will be used for testing the data, while the remaining (858 or 857) examples are used to train the classifier.

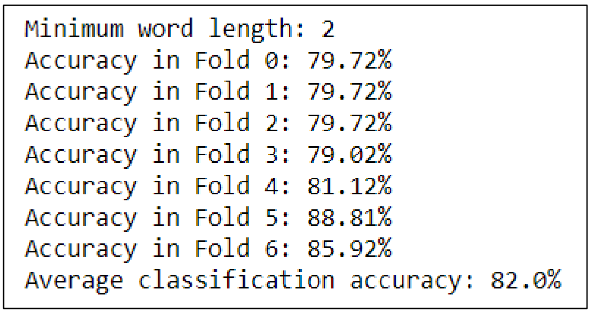
The count of each word occurring in either class is maintained in a dictionary.

There may be words in class 0 that do not appear in the sentences that belong to class 1, which will drive the probability of being in either of the classes to 0. To overcome this issue, one technique is called Laplace smoothing, which assigns a count of 1 to all the words in Class 1 that are not in Class 0, and similarly a count of 1 to all the words in Class 0 that do not occur not in 1. The count of all words is increased by 1 as well. This maintains the inference that if a word has not occurred in the class, then the probability of its occurrence remains very low. Hence, encountering that word will reduce the probability of it belonging to its non-parent (according to training data) class when a test case is encountered.

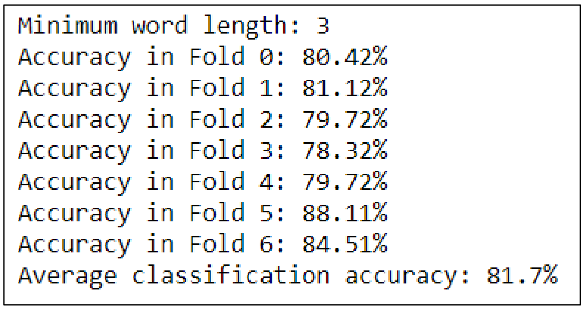
Accuracy of Model Over Each Fold and the Overall Average Accuracy:



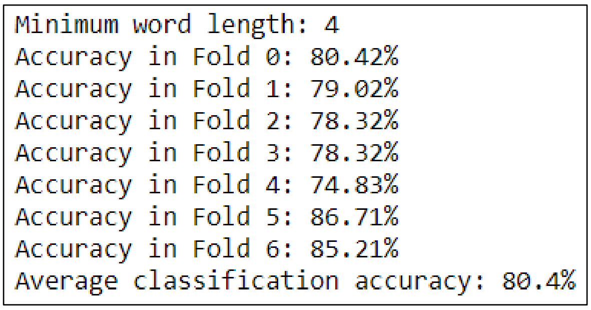
```
Minimum word length: 1
Accuracy in Fold 0: 79.72%
Accuracy in Fold 1: 80.42%
Accuracy in Fold 2: 79.72%
Accuracy in Fold 3: 79.02%
Accuracy in Fold 4: 80.42%
Accuracy in Fold 5: 87.41%
Accuracy in Fold 6: 85.92%
Average classification accuracy: 81.8%
```



```
Minimum word length: 2
Accuracy in Fold 0: 79.72%
Accuracy in Fold 1: 79.72%
Accuracy in Fold 2: 79.72%
Accuracy in Fold 3: 79.02%
Accuracy in Fold 4: 81.12%
Accuracy in Fold 5: 88.81%
Accuracy in Fold 6: 85.92%
Average classification accuracy: 82.0%
```



```
Minimum word length: 3
Accuracy in Fold 0: 80.42%
Accuracy in Fold 1: 81.12%
Accuracy in Fold 2: 79.72%
Accuracy in Fold 3: 78.32%
Accuracy in Fold 4: 79.72%
Accuracy in Fold 5: 88.11%
Accuracy in Fold 6: 84.51%
Average classification accuracy: 81.7%
```



```
Minimum word length: 4
Accuracy in Fold 0: 80.42%
Accuracy in Fold 1: 79.02%
Accuracy in Fold 2: 78.32%
Accuracy in Fold 3: 78.32%
Accuracy in Fold 4: 74.83%
Accuracy in Fold 5: 86.71%
Accuracy in Fold 6: 85.21%
Average classification accuracy: 80.4%
```

Important results:

- The accuracy of the model is roughly the same, (81.7 - 82.0%) until we filter out words with 3 or less than 3 characters.
- The drop in accuracy from 82.0% to 80.4% tells us that there are 3 letter words in the dataset that contribute to the decision of which class an unseen sentence may belong to. Similar observations can be made in the case of individual fold's results as well.

Major Limitations of the Naïve Bayes Model:

1. To simplify the calculation of probabilities of the model, the assumption of class conditional independence is made; it is assumed that the features (here, words in a sentence, length, position of each word in the sentence etc.) of an example are independent of each other. This assumption is very rarely true, and hence the Naïve Bayes algorithm is less accurate, but fast. More complicated algorithms attempt to relax these assumptions.
2. If Laplace Smoothing is not performed, some sentences may wrongly be assigned zero probability of belonging to a class, merely due to absence of enough training examples. Smoothing techniques are ways to overcome this issue, while the ideal solution to this would be training examples that exhaust the universal set of attribute values (here, entire English dictionary).
3. The algorithm is probabilistic. It's outcomes and the probabilities it assigns to various classes may not always be reliable.

TEAM DETAILS –

- **PRAJJWAL VIJAYWARGIYA – 2017B3A70954H**
- **PARTH KRISHNA SHARMA – 2017B3A70907H**
- **SIDDHI MAHESH BURSE – 2017B3A70972H**