MACHINE LEARNING

BITS F464

ASSIGNMENT 1

Done By:

Rohan Maheshwari	2017B4A70965H
Keshav Kabra	2018AAPS0527H
Godhala Meganaa	2017B3A70973H

Assignment 1B: **Naive Bayes Classifier**

Model Description

Data was extracted and pre-processed by converting to lower case and removing stopwords. A vocabulary was generated which consisted of the unique words in the corpus. An inverted index was made which mapped words in the vocabulary to a list of data item numbers it appeared in, this dictionary will help reduce computations later.

Data was shuffled and 7-fold cross validation was applied to get train and test sets. First we find number of positive and negative examples in the train set and divide them by total number of train items which gives us P(spam) and P(not spam). Then we compute the P(word|spam) for all the words in the vocabulary by counting the number of positive items containing the given word divided by number of positive items in train set. Similarly P(word|not spam) was computed. Laplace Smoothing was used to avoid division by zero at this stage.

After this we went over the test set and used the Bayes Theorem and Feature Independence rule to compute predicted labels and finally accuracy was computed by dividing number of correctly classified examples divided by total test items. The model returns the overall average accuracy.

The **formulation** of the algorithm can be done as below:

By Bayes Theorem:

$$P(C_i|\, x_1, x_2 \ldots, x_n) = \frac{P(x_1, x_2 \ldots, x_n|C_i).P(C_i)}{P(x_1, x_2 \ldots, x_n)} \; for\; 1 < i < k$$

Using the Feature Independence Assumption("naive" part of Naive Bayes), we get:

$$P(C_i|\, x_1, x_2 \ldots, x_n) = \left(\prod_{j=1}^{j=n} P(x_j|\, C_i)\right) \cdot \frac{P(C_i)}{P(x_1, x_2 \ldots, x_n)} \; for\; 1 < i < k$$

Since P(x1,x2,…,xn) is constant for all classes, we get :

$$P(C_i|\, x_1, x_2 \ldots, x_n)\; \alpha \left(\prod_{j=1}^{j=n} P(x_j|\, C_i)\right) \cdot P(C_i) \; for\; 1 < i < k$$

Accuracy

Iterations **Folds**	**Iteration 1**	**Iteration 2**	**Iteration 3**
1	83.0986	78.8732	84.507
2	80.9859	80.9859	83.0986
3	78.8732	87.3239	78.169
4	82.3944	79.5775	76.7606
5	84.507	81.6901	83.8028
6	80.9859	80.9859	82.3944
7	80.2817	78.8732	81.6901
Overall Avg Accuracy	**81.5895**	**81.1871**	**81.4889**

Limitations of Naive Bayes classifier

1. The biggest limitation of Naïve Bayes is the Independent Feature Assumption. For Naive Bayes we make an assumption that all the attributes are mutually independent. In real life, it's very rare that we get a set of attributes that are completely independent of one another.

2. If a category of a categorical variable in the testing set was not observed in the training set, then the model will assign a 0 probability to that category and will be unable to make a prediction which will result in zero division error. This is known as 'zero-frequency problem'. This problem can be solved by smoothing techniques.

Laplace Smoothing:

One of the techniques to avoid the ‘zero-frequency problem that we have used. The tweaked formula for probability computation is as follows:

$$P(w'|positive) = \frac{\text{number of reviews with w' and y = positive } + \ \alpha}{\text{N } + \ \alpha * K}$$

alpha represents the smoothing parameter,
K represents the number of dimensions (features) in the data, and
N represents the number of reviews with y=positive

in our implementation we have taken alpha = 1 and since we are working in one dimension, K = 1.