# INFORMATION RETRIEVAL

**ASSIGNMENT 3**

Submitted by –

Vaibhav Varshney(MT17065)

## Dataset

- The dataset consist of five folders namely, “comp.graphics”, “rec.sport.hockey”, “sci.med”, “sci.space”, “talk.politics.misc”.
- The index of the folder names is considered as their labelled class.
- Each folder consist of 1000 files each.

## Assumption

- It is assumed that the distribution followed for the conditional probability is binomial distribution. Hence the decision function for classification followed will be:

$$c_{\text{map}} = \underset{c \in \mathbb{C}}{\arg\max} \left[ \log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c) \right]$$

- It is assumed that all the classes are equally probable.
- The test data doesn’t contains numerical terms.

## Methodology

### (1) Preprocessing

Preprocessing is done in the following sequence:

- Punctuation removal
- Tokenization
- Stop words removal
- Stemming(using Porter Stemmer Algorithm)

### (2) Feature Selection

For optimization of the performance of the model, the following heuristic is implemented. I have calculated the tf-idf values of all the words of the complete corpus. Further, on the basis certain percentage, I have selected the words with high tf-idf value from each document. Taking the union of all the words, a feature vector has been selected.

### (3) Smoothing

During the training phase, the conditional probabilities are calculated of the tokens present in the train data. If some unknown token arrives then it is handled using the **“Laplace Smoothing”.**

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)}$$

## (4) Splitting of the Data

For the splitting of the dataset, "train_test_split" function has been used from the sklearn's library. It provides the split into train size and test size as provided input by the user.

# Analysis

**Without Feature Selection**- Following accuracies have been observed at different train-test split of the dataset.

| Split ratio | Test Accuracy |
|---|---|
| **90:10** | 0.94 |
| **80:20** | 0.952 |
| **70:30** | 0.9534 |
| **50:50** | 0.9376 |

**With Feature Selection**- Following accuracies have been observed at different train-test split of the dataset.

| Split ratio | Feature selection %age | Test Accuracy |
|---|---|---|
| **90:10** | 10% | 0.66 |
| | 30% | 0.848 |
| | 50% | 0.91 |
| | 90% | 0.94 |
| **80:20** | **10%** | **0.617** |
| | 30% | 0.854 |
| | 50% | 0.924 |
| | 90% | 0.956 |

| | | |
|---|---|---|
| **70:30** | **10%** | **0.53** |
| | 30% | 0.9146 |
| | 50% | 0.922 |
| | 90% | 0.961 |
| **50:50** | **10%** | **0.734** |
| | 30% | 0.908 |
| | 50% | 0.894 |
| | 90% | 0.943 |

## Observations

The following facts have been observed by seeing the results of the experiment:

- In different split ratios of train and test data, it has been observed that on increasing the training data size, an increase in the test data's accuracy.
- Further, on performing grid search the percentage of feature selection (i.e., selecting 'x%' of tokens based on tf-idf values), a trend is observed. On increasing the feature selection, the accuracies are increasing for each of the split.

## Inference

- By seeing the accuracies of the test data at different split ratios, it can be stated that the model is best fitting at 70:30 ratio of train-test split data.
- It can be stated that the model is overfitting for 80:20 and 90:10 split ratios. (Refer to graphs)
- In terms of **calculation time,** with performing feature selection, the calculation is taking way more time as compared to without feature selection.

# Graphical Results

## (1) Confusion Matrix

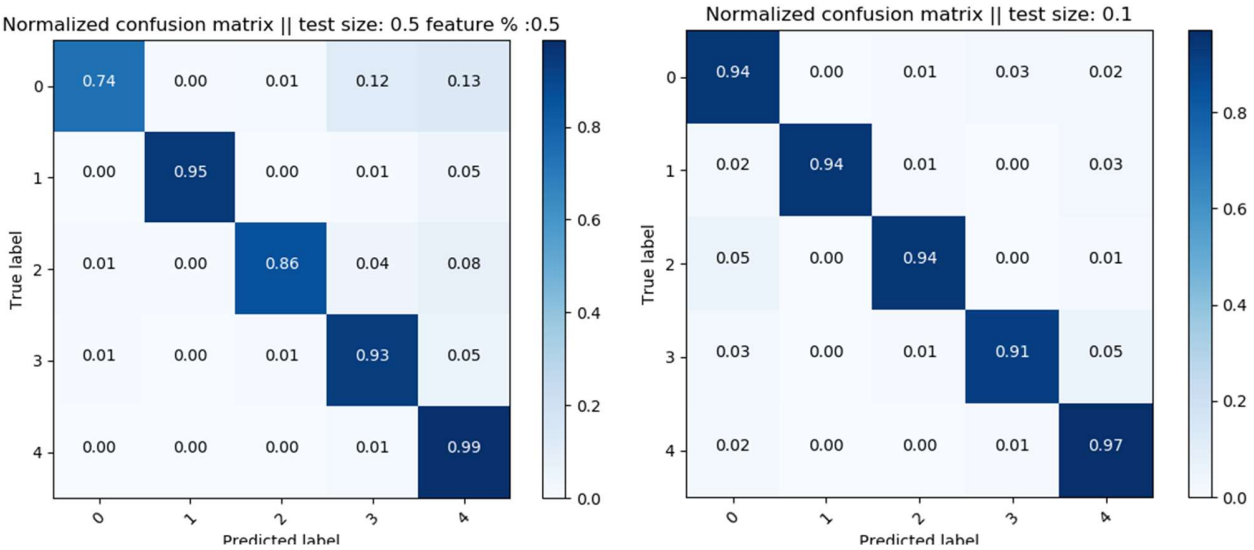


**Fig: with feature selection(left) and without feature selection(right)**

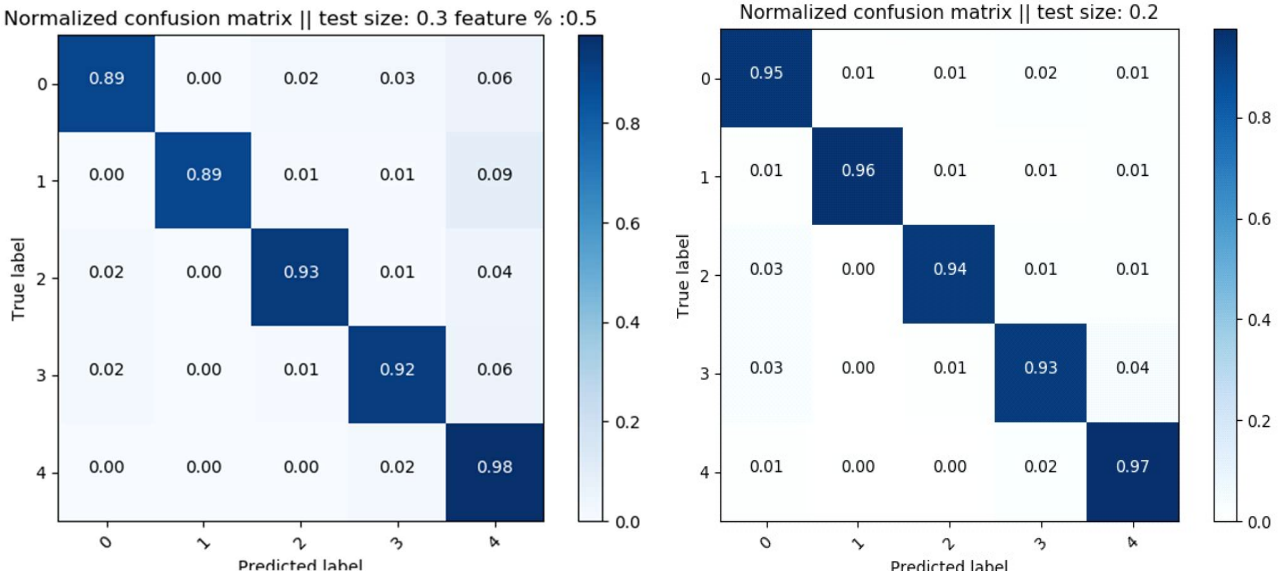


**Fig: with feature selection(left) and without feature selection(right)**

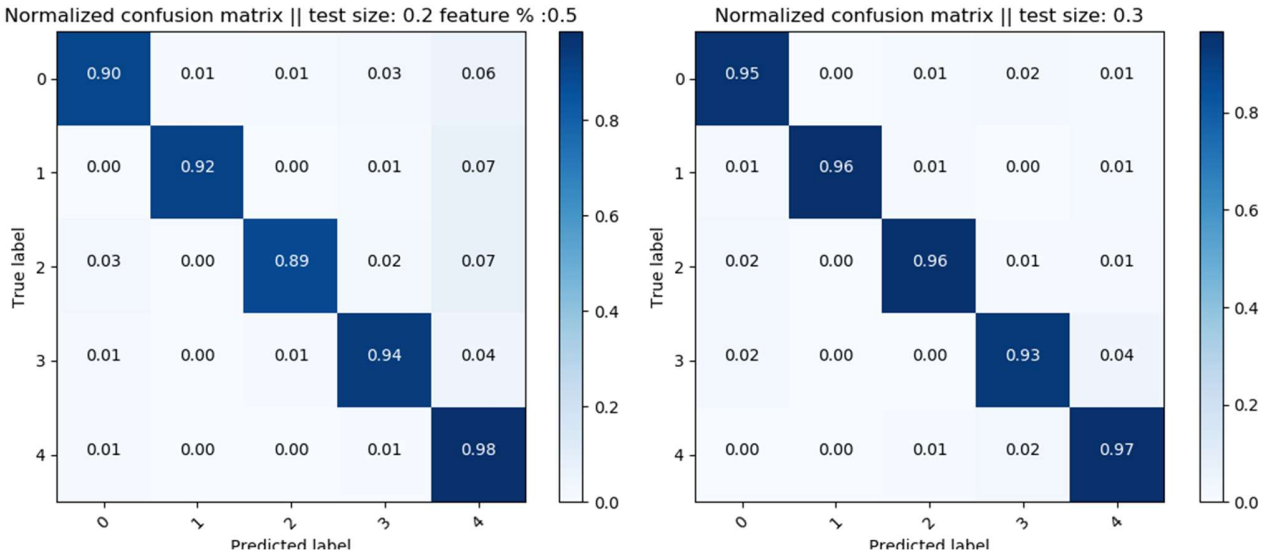


**Fig: with feature selection(left) and without feature selection(right)**

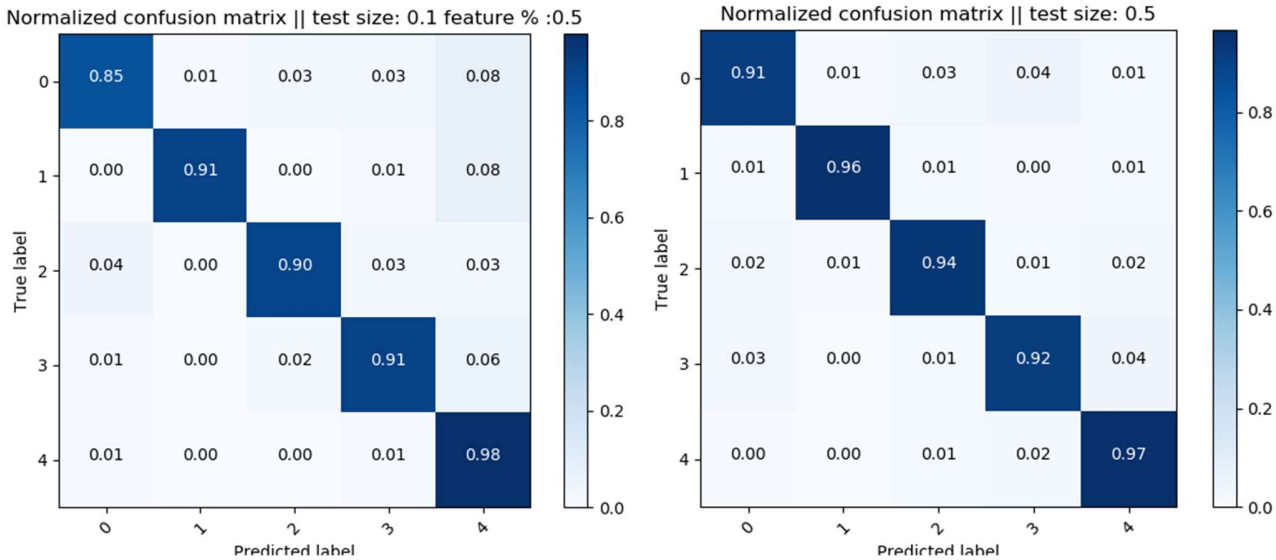


**Fig: with feature selection(left) and without feature selection(right)**

## (2) Line Graphs for accuracies

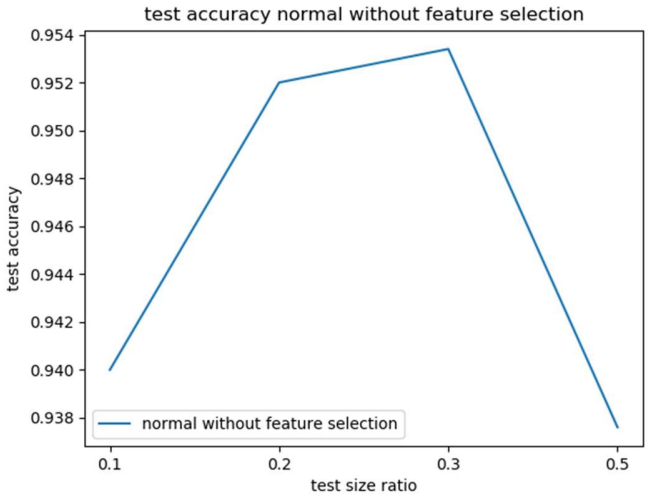


**Fig: Without feature selection**

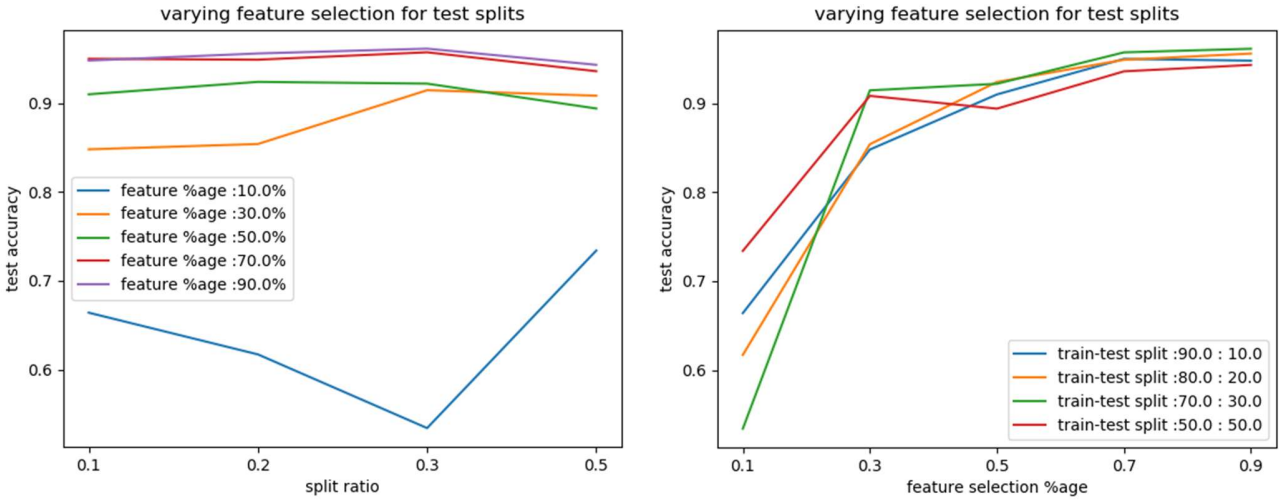


**Fig: With feature selection**

---