**Assignment for Week 3 - Naive-Bayes**[**¶**](http://localhost:8928/notebooks/Assign_Week3.ipynb#Assignment-for-Week-3---Naive-Bayes)

**Math Exercise:**Scanned Solution was uploaded

**Coding Exercise:**

**Text Classification**

**Approach:**

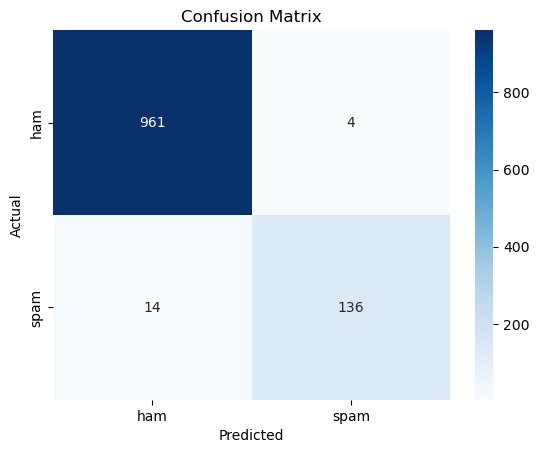
* Given the spam data set, downloaded and used for modeling, The data set has 5,572 data points with two columns, ‘v1’ and ‘v2’.
* These column names are relabelled as labels and messages and preprocessed further for developing a classification model.
* The messages are changed into lowercase to avoid repetitions of words.
* The numerical (numbers) were removed from the dataset as they don’t convey any information in case of text classification.
* All the punctuation marks such as .,!?;:()[]{}<> are removed as they done have any meaning.
* Extra white spaces such as unnecessary spaces, tabs, and newline characters are also removed to ensure uniformity and reduce noise.
* Stop words i.e., common words such as the, is, and, in, to, etc. don’t carry significant meaning and are removed to reduce dimensionality and improve model efficiency.
* Stemmping that reduces words to their root form by chopping off suffixes is done to reduce noise.
* The given data was splitted in to training (80%) and test data (20%).
* A document – term matrix is created Matrix using CountVectorizer in order to split the message in to individaual words.
* The frequency of top 10 words is visualized carefully
* A multinomial naïve bayes classifier model is developed using the test data and trained againest the test data to classify SMS message as spam or not spam (ham).
* Evaluated using some metrics, including accuracy , precision, recall and F1 score and visulized the model using confusion matrix and word cloud.

**Analysis Questions:**

1. What is the accuracy of the model? Report your finding with corresponding tables/graphs.

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 0.98 |
| Precision (macro avg) | 0.98 |
| Recall (macro avg) | 0.95 |
| F1 Score (macro avg) | 0.96 |

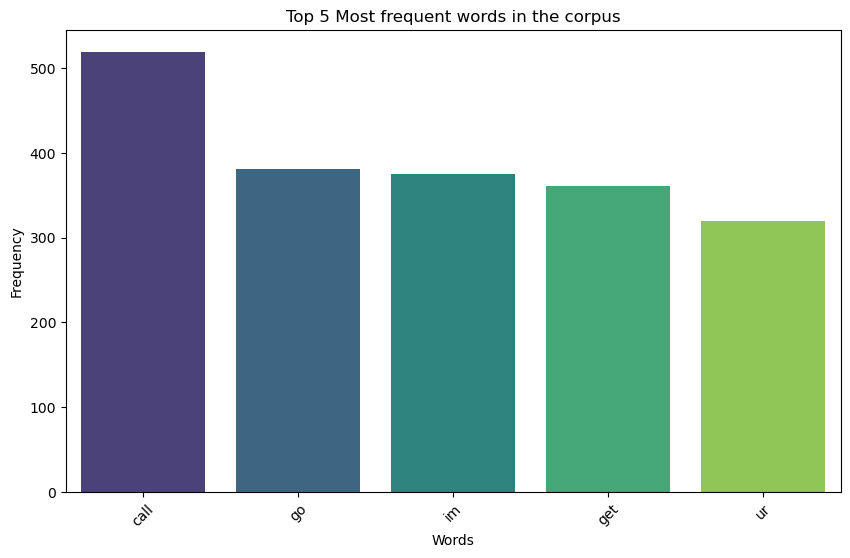
* The accuracy of the Naïve Bayes classifier is 98.3%, as calculated using the test set. This indicates that the model performs very well in distinguishing between spam and ham messages.



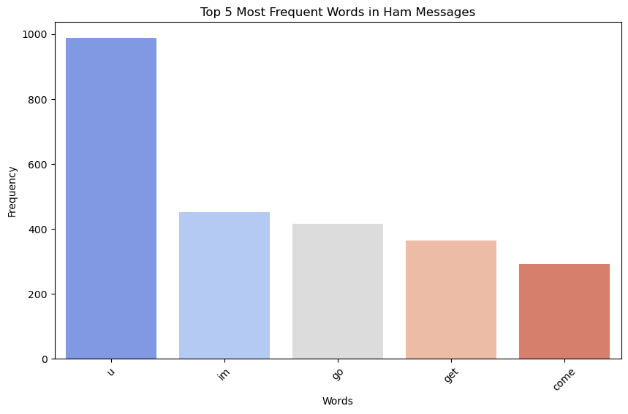
Even most of the datapoint are predicted correctly based on confusion matrix.

2. Print the 5 most frequent words in each class, and their posterior probability generated by the model.

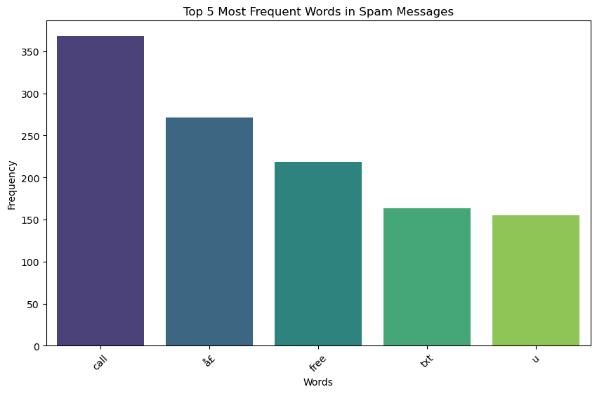
* The messages are convereted in to matrix form of each words and the frequency of the top 5 features are visualized using an histogram



Based on the histogram, the top 5 most frequent words are call, go, im, get and ur in the entire corpus. Whereas, in case of each class the most frequent words are shown in histograms beleow.

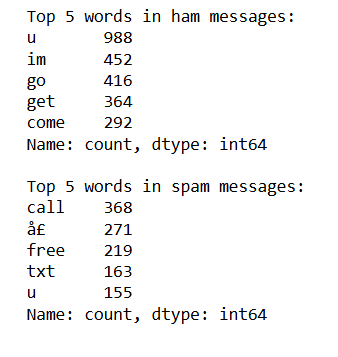


The most frequent words for non spam (Ham) messages are u, im, go, get, and come

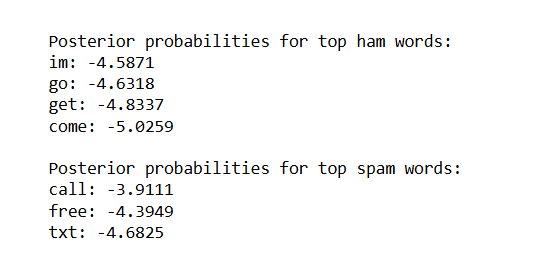


The most frequent words in spam messages are call, ae, free, txt, and u.

The Exact word counts are:

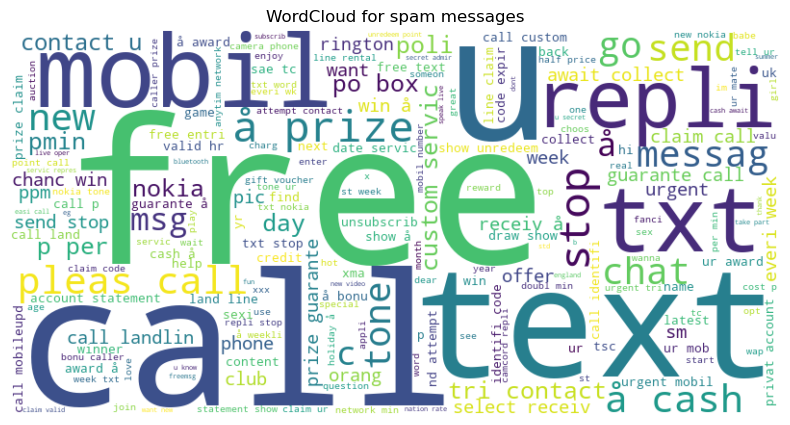


The posterior probabilities of top words are:



Based on the posterior probabilities the words call, free and text and most common and probable to come in spam messages. The most frequent words are also visualized using the word clouds, which also interprests the same point.





3. How would you improve the model performance?

Naive bayes Model performance can be improved using various techniques:

**1. Remove Correlated Features:**

Naive Bayes algorithms perform well on datasets with no correlations in independent features. we can check for correlation of the features and removing the highly correlated features may improve the performance of the algorithm

**2. Feature Engineering:**

Feature engineering techniques such as combine some of the elements, and extract some parts of them out of existing ones. This may help the Naive Bayes algorithm learn the data quickly and results in an accurate model.

**3. Use Some Domain Knowledge:**

Oe should always try to apply some domain knowledge to the dataset and its features and take steps according to it. It may help the algorithm to make decisions faster and achieve higher accuracies.

**4. Probabilistic Features:**

The Naive Bayes algorithm works on the concept of probabilities, so try to improve the features that give more weightage to the algorithms and their probabilities, try to implement those, and run the roses in a loop to know which features are best for the algorithm.

**5. Smoothing or Laplace smoothing Transform:**

In some cases, the category may be present in the test dataset and was not present while training and the model will assign it with zero probability. Here we should handle this issue by using smoothing techniques such as Laplace transform.

**6. Feature Transformation:**

It is always better to have normal distributions in the datasets and try to apply box-cox and yeo-johnson feature transformation techniques to achieve the normal distributions in the dataset.

4. If the data set is bigger, do you think the accuracy increases? Discuss.

**Effect of bigger data set**

1. A larger dataset would likely improve the model's accuracy, as it would provide more examples for the classifier to learn from, especially for rare words or patterns.
2. However, after a certain point the increase in data might not improve the accuracy much but increases the computational cost.

**Reference:** Naive Bayes Algorithms: A Complete Guide for Beginners <https://www.analyticsvidhya.com/blog/2023/01/naive-bayes-algorithms-a-complete-guide-for-beginners/>