# Introduction

The creation of a spam detection software utilizing the Naive Bayes Classifier with TF-IDF and Bag-of-Words (BoW) features will be covered in this paper. For optimum performance, our attention will be on comprehending the dataset, preprocessing, and choosing the best data structures (Manning et al., 2008).

## Design and Preprocessing of Data

### *Conditions*

- Utilize the Naive Bayes Classifier to create a spam detection algorithm.
- TF-IDF and Bag-of-Words features can be handled as input.
- Create training and test sets from the given dataset.
- Apply text-preprocessing methods.

### *Familiarization with Datasets*

This project will make use of the SMS Spam Collection Dataset (http://archive.ics.uci.edu/ml/machine-learning-databases/00228/). SMS texts classified as "ham" (non-spam) or "spam" are included in the collection. Using a conventional 80/20 or 70/30 ratio, we will divide the data into a training set and a testing set (Alpaydin, 2010).

### *Text Preprocessing*

The first stage in any task involving natural language processing is text preprocessing. For this project, the following preparation methods will be useful:

- Tokenization is the process of separating a text into tokens, or individual words (Jurafsky & Martin, 2019).
- Lowercasing: To maintain uniformity, change all of the text's characters to lowercase.
- Remove terms like "the," "and," "is," etc. that are frequently used but have little meaning (Manning et al., 2008).
- To reduce vocabulary size and increase generalization, words can be stemmed or lemmatized down to their basic form (Jurafsky & Martin, 2019).

### *Data Structures and Code Framework Section*

We'll choose data structures and algorithms based on their effectiveness and efficiency. For instance, storing the term-document matrix in sparse matrices can help conserve memory and computing resources. Functions to handle preprocessing, feature extraction, model training, and evaluation will be included in the initial code structure.

### *Example: Preprocessing SMS Dataset*

Using the SMS dataset, we will illustrate the text preprocessing procedures in this part. The dataset's first few SMS messages are as follows:

We'll use preprocessing methods including stemming, stopword elimination, lowercasing, and tokenization. Here is each example's processed text:

- ham (original): Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
- ham (processed): go jurong point crazy avail bugi great world buffet cine amore wat
- ham (original): Ok lar... Joking wif u oni...
- ham (processed): ok lar joke wif oni
- spam (original): Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
- spam (processed): free entry wkli comp win fa cup final tkt 21st may 2005 text fa 87121 receiv entry question std txt rate apply 08452810075over18

The preprocessed text will be used for feature extraction using both TF-IDF and BoW approaches after the preprocessing techniques have been applied. This will make it easier for the Naive Bayes classifier to comprehend and spot textual patterns that can distinguish between spam and legitimate messages.

## References:


1.Alpaydin, E. (2010). Introduction to Machine Learning. MIT Press.

2.Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing. Prentice Hall.

3.Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.