Preprocessing of SMS Spam Messages: towards clustering and classification purpose

**Data Source and Problem Statement**

**Download** the SMS\_SPAM data set from given link:

<https://drive.google.com/open?id=1whEKTK-BCDCEhHDHumYpw7FuM-tAFVLT>

**Description of dataset:**

This dataset includes the text of SMS messages along with a label indicating whether the message is unwanted. Junk messages are labeled **spam**, while legitimate messages are labeled **ham**.

**Output:**

**Text data into tabular form**

****

Background:

Some examples of spam and ham are shown in the following table:

|  |  |
| --- | --- |
| **Sample SMS ham** | **Sample SMS spam** |
| Better. Made up for Friday and stuffed myself like a pig yesterday. Now I feel bleh. But, at least, its not writhing pain kind of bleh. | Congratulations ur awarded 500 of CD vouchers or 125 gift guaranteed & Free entry 2 100 wkly draw txt MUSIC to 87066. |
| If he started searching, he will get job in few days. He has great potential and talent. | December only! Had your mobile 11mths+? You are entitled to update to the latest colour camera mobile for Free! Call The Mobile Update Co FREE on 08002986906. |
| I got another job! The one at the hospital, doing data analysis or something, starts on Monday! Not sure when my thesis will finish. | Valentines Day Special! Win over £1000 in our quiz and take your partner on the trip of a lifetime! Send GO to 83600 now. 150 p/msg rcvd. |

Looking at the preceding messages, did you notice any distinguishing characteristics of spam? One notable characteristic is that two of the three spam messages use the word "free," yet the word does not appear in any of the ham messages. On the other hand, two of the ham messages cite specific days of the week, as compared to zero in spam messages.

While it's not inconceivable that the word "free" would appear outside of a spam SMS, a legitimate message is likely to provide additional words explaining the context. For instance, a ham message might state "are you free on Sunday?" Whereas, a spam message might use the phrase "free ringtones." The classifier will compute the probability of spam and ham, given the evidence provided by all the words in the message.

**Step 1 – exploring and preparing the data**

The first step towards constructing our classifier involves processing the raw data for analysis. Text data are challenging to prepare, because it is necessary to transform the words and sentences into a form that a computer can understand. We will transform our data into a representation known as **bag-of-words**.

Common structure of data set for any classification methods:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| FEATURES | | | | | | | LABEL |
| Roll No. | 10th CGPA | 12th CGPA | BTech CGPA | Commu. Skill | Tech. Skill | Personality | Placed |
| 01 | 6 | 9 | 8 | 8 | 7 | 8 | YES |
| 02 | 3 | 5 | 8 | 6 | 6 | 10 | NO |
| …. | …. | …. | …. | …. | …. | …. | NO |

Note: SMS dataset do not in this format and so cannot suitable for classification algorithms.

Solution: Text-> Features

(transform text data to tabular data)

Hint: -

**Data preparation – cleaning and standardizing text data**

SMS messages are strings of text composed of words, spaces, numbers, and punctuation. Handling this type of complex data takes a lot of thought and effort. One needs to consider how to remove numbers and punctuation; handle uninteresting words such as *and*, *but*, and *or*; and how to break apart sentences into individual words.

Standardize the words, by removing punctuation and other characters that clutter the result.

For example, we would like the strings

*Hello*!, *HELLO*, and *hello* to be counted as instances of the same word.

Let's continue our cleanup by removing numbers from the SMS messages. Although some numbers may provide useful information but right now we are not considering it important.

Our next task is to remove filler words such as *to*, *and*, *but*, and *or* from our SMS messages. These terms are known as **stop words** and are typically removed prior to text mining. This is due to the fact that although they appear very frequently, they do not provide much useful information for machine learning.

Another common standardization for text data involves reducing words to their root form in a process called **stemming**. The stemming process takes words like *learned*, *learning*, and *learns*, and strips the suffix in order to transform them into the base form, *learn*. This allows machine learning algorithms to treat the related terms as a single concept rather than attempting to learn a pattern for each variant.

After removing numbers, stop words, and punctuation as well as performing stemming, the text messages are left with the blank spaces that previously separated the now-missing pieces.

The following table shows the first three messages in the SMS corpus before and

after the cleaning process.

|  |  |
| --- | --- |
| **SMS messages before cleaning** | **SMS messages after cleaning** |
| **> as.character(sms\_corpus[1:3])**  **[[1]] Hope you are having a good**  **week. Just checking in**  **[[2]] K..give back my thanks.**  **[[3]] Am also doing in cbe only.**  **But have to pay.** | **> as.character(sms\_corpus\_clean[1:3])**  **[[1]] hope good week just check**  **[[2]] kgive back thank**  **[[3]] also cbe pay** |

**Data preparation – splitting text documents into**

**Words (Feature extraction-Tabular form)**

Now that the data are processed to our liking, the final step is to split the messages into individual components through a process called **tokenization**. A token is a single element of a text string; in this case, the tokens are words.

****