Problem Statement

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged as either ham (legitimate) or spam.
We have to train a model which may classify any given message as a ham or spam. This whole dataset can be used for the training of the classifier.

Source:

Tiago A. Almeida (talmeida ufscar.br)

Department of Computer Science

Federal University of Sao Carlos (UFSCar)

Sorocaba, Sao Paulo - Brazil

JosÃ© MarÃa GÃ³mez Hidalgo (jmgomezh yahoo.es)

R&D Department Optenet

Las Rozas, Madrid - Spain

Data Set

The dataset can be found at UCI Machine Learning repository

Attribute Information:

The collection is composed by just one text file, where each line has the correct class followed by raw message.

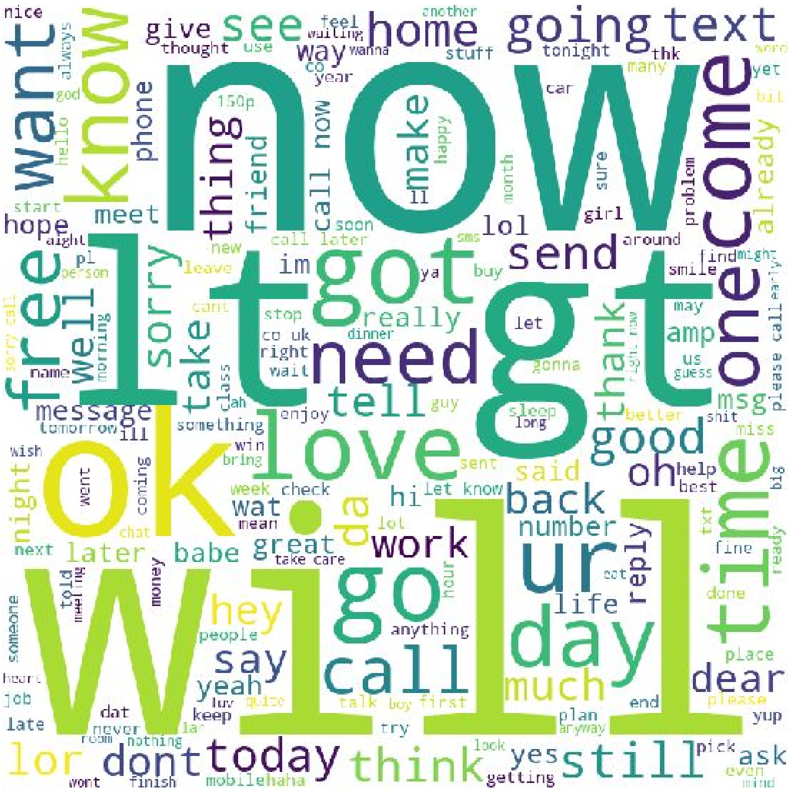
Data

Label	Either ham or spam
Message	The message or the sms which is classified.

HYPOTHESIS

Every message has a semantic and a grammatical meaning. By deciphering which type of semantic and grammatical meaning does a specific sentence carry, we may be able to figure out the difference between a spam and a ham message.

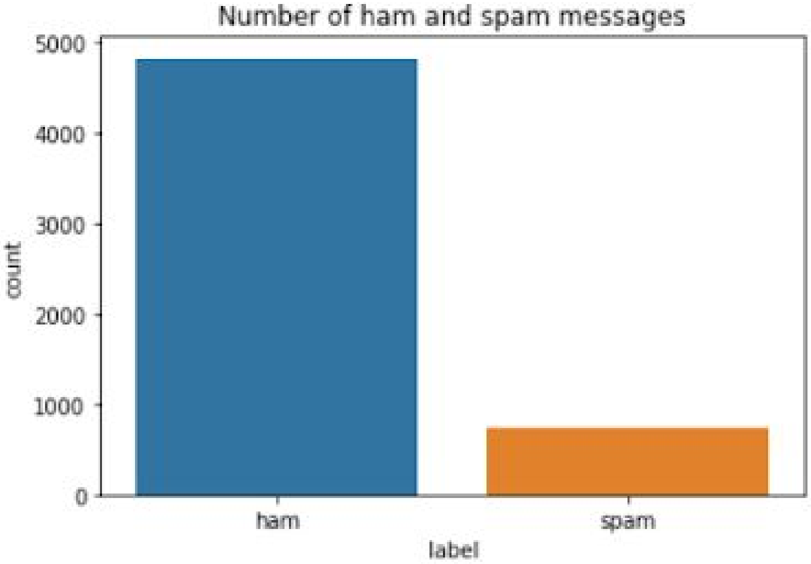
In the same way we can train models which can differentiate between the two categories.
We have been provided by just the Label, and the Message columns.



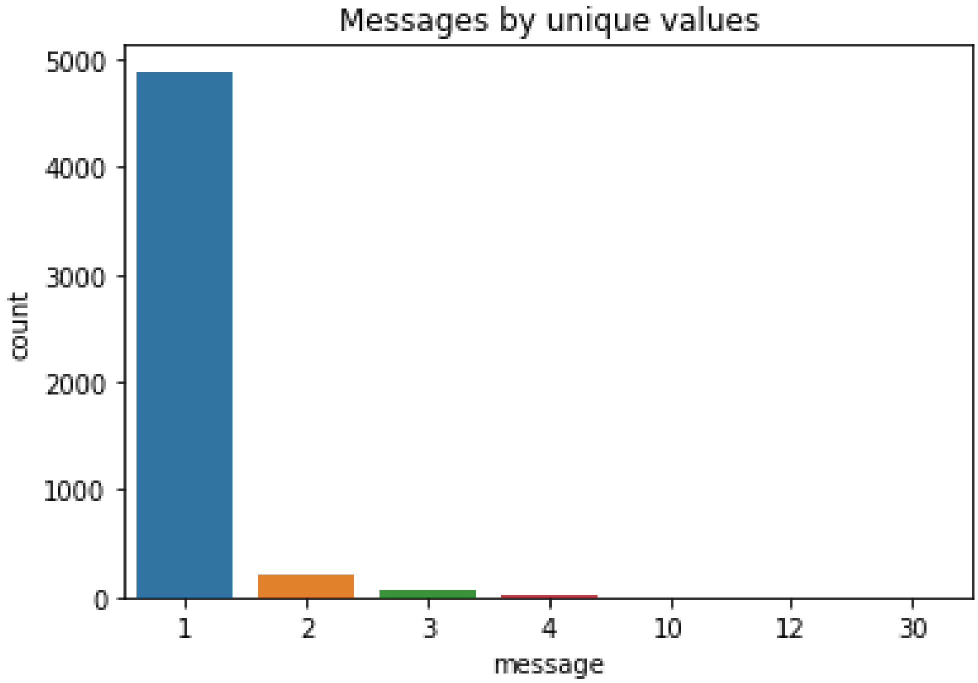
Data Wrangling & Exploratory Data Analysis

The dataset has two columns the label and the message.

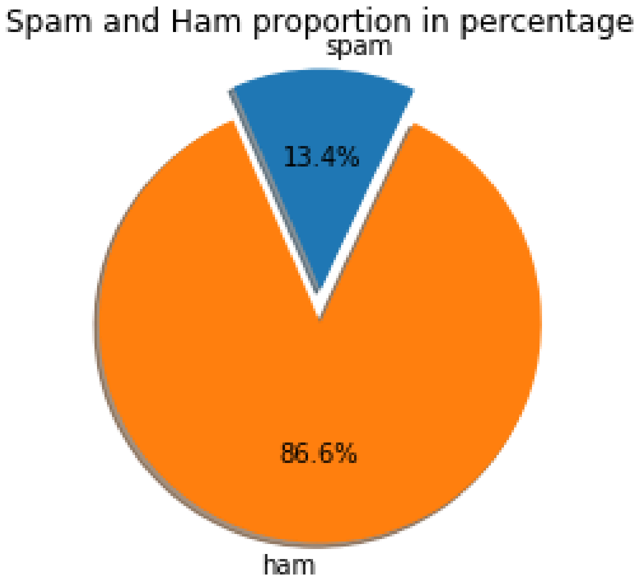
- We check the info() of the label and message columns. We find that there are 5572 non null values in both the columns.
- With the describe method we saw the most returned label is ham with 4825 entries and the most repeated message is Sorry, I’ll call you later.
- 5169 messages are unique



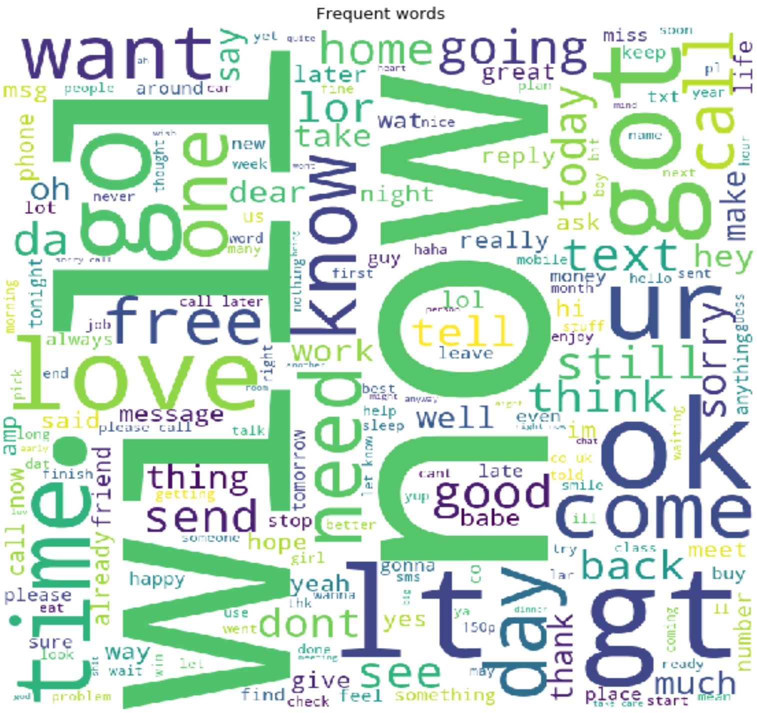
We can see that the majority of the messages are ham.



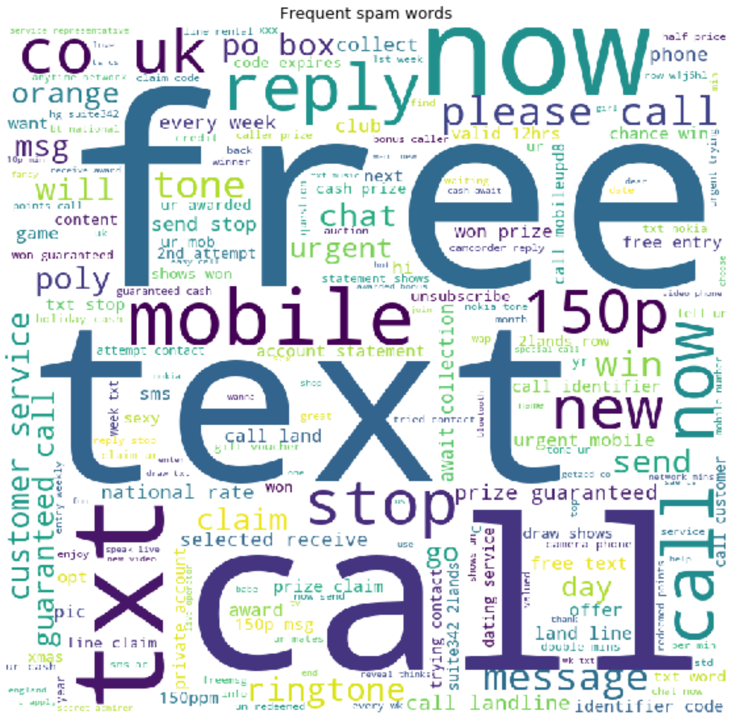
Most of the messages are unique as we can see the blue bar. The other frequencies recorded for messages are 2, 3, 4, 10, 12, and 30 times.



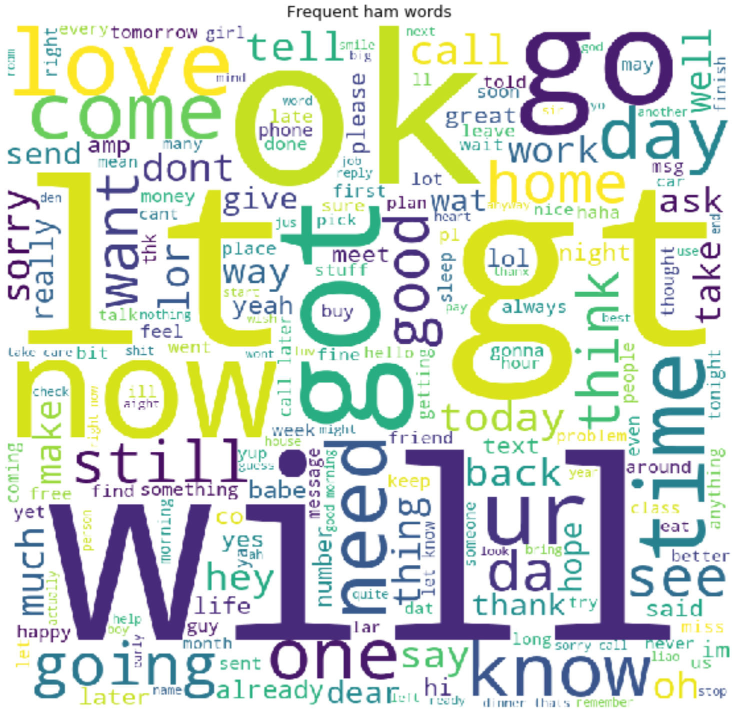
The proportion of ham messages is 86.6%, whereas that of spam messages stands at 13.4%.



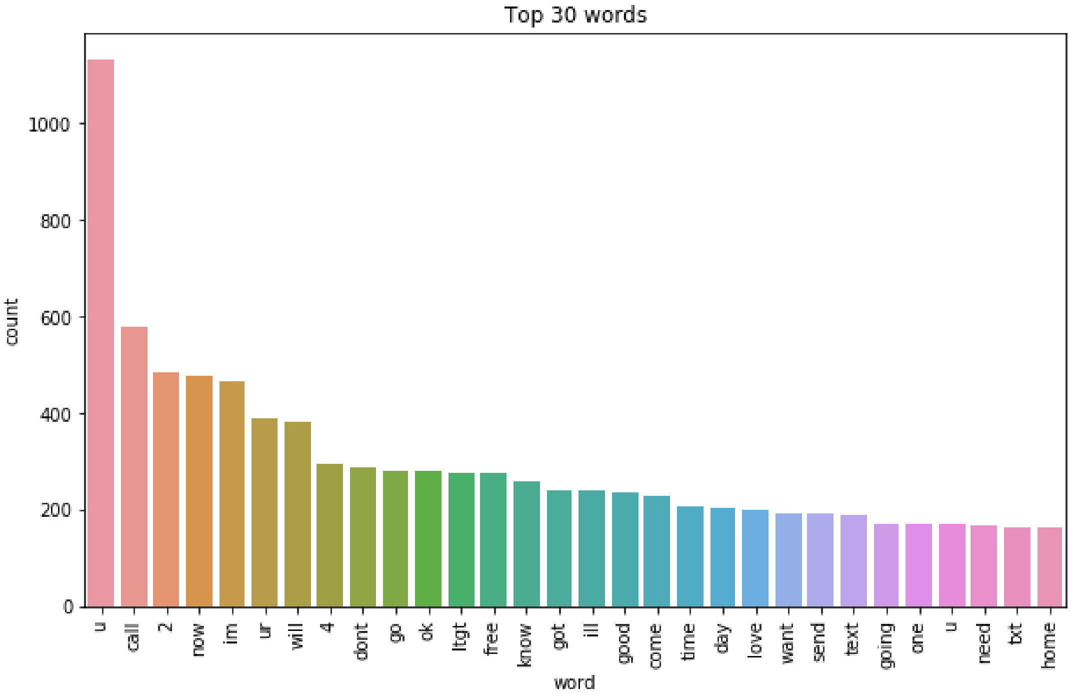
Frequent words used in the messages. Most frequent are will, now, want It, gt. Most frequent words we can see are from the ham messages as these messages are more in number.



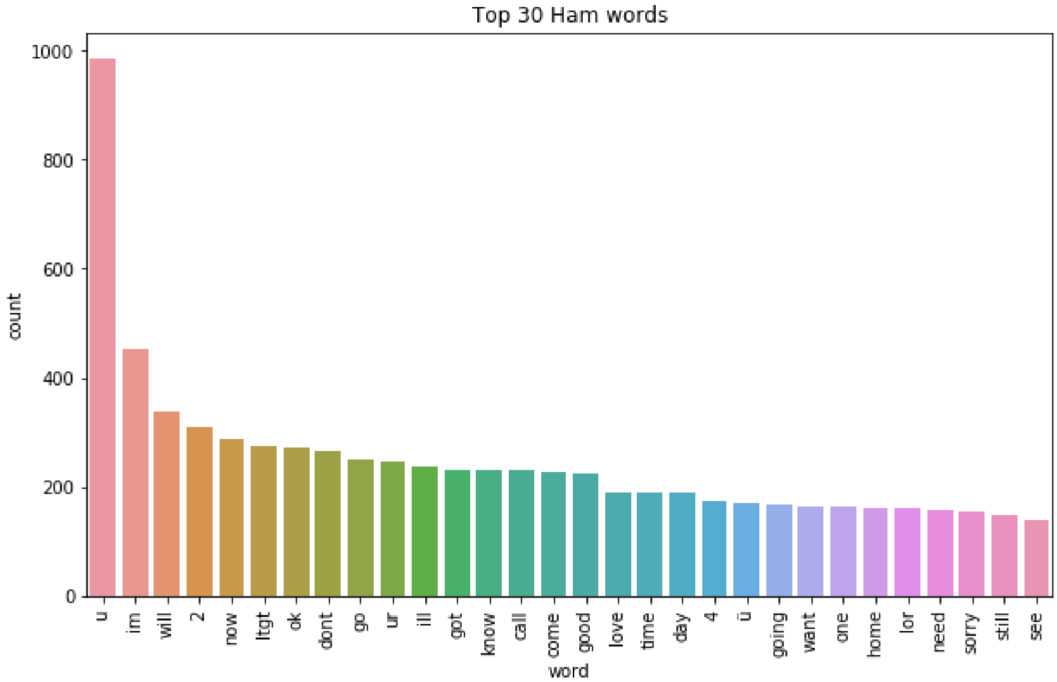
Frequent words used in the spam messages are free, text, call. These words show that the messages are trying to sell something and want the receiver to respond. They may even have malicious URLs.



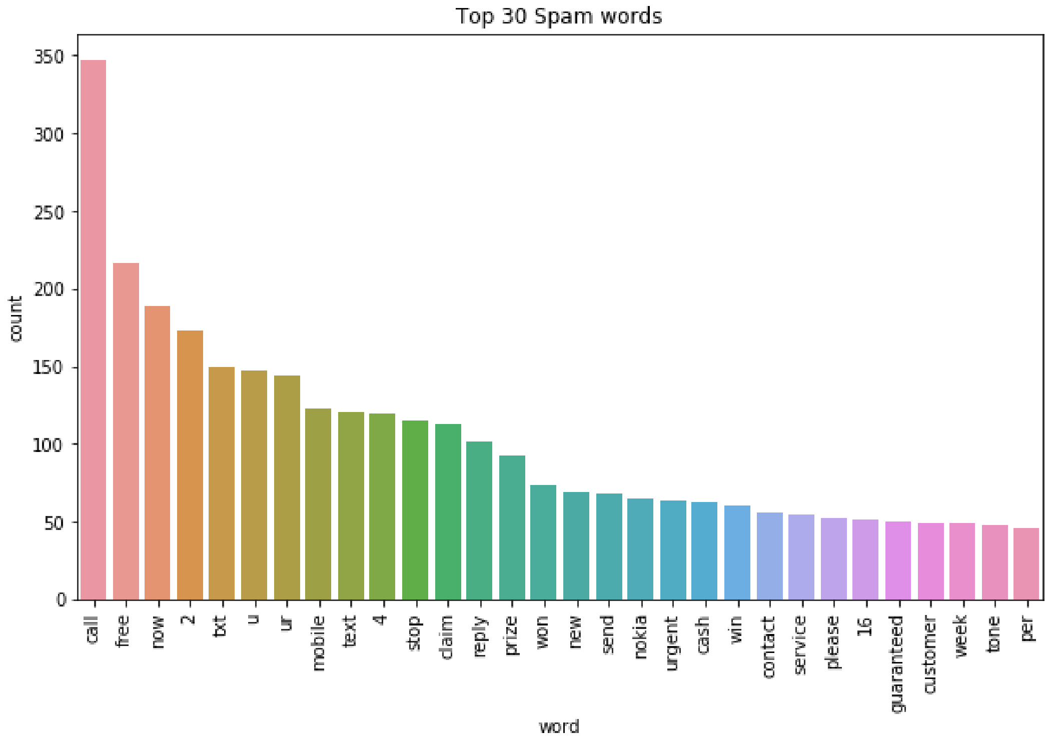
Frequent words used in ham messages are will, lt, ok, gt. They are talking about the real time conversation and talking about the near future.



The top 30 words used in all the messages.



The top 30 words used in the ham messages.



The top 30 words used in the ham messages.

Data Preprocessing

We have to preprocess the textual data first as the models can only work with numbers. All these words are converted to numbers with the **bag of words** model. The **NLTK** library has been used for this.

Basic preprocessing for common NLP tasks includes converting text to lowercase and removing punctuation and stopwords.
Further steps, especially for text classification tasks, are:

- **Tokenization** - models treat every word as bits of information
- **Removal of Stopwords** - removal of words like as, the, for because they don't have any special meaning
- **Vectorization through CountVectorizer** - To convert the text into numbers, so that predictive models can work on them
- **TF-IDF weighting** - many words appear more than others; they need to be properly weighed, so that they don't distort the outcome in their favor
- **Stemming through SnowballStemmer** - words are reduced to their word stems

Modelling

With insights based on exploratory data analysis (EDA), we start to train classification models.

We identified six models to try first:

1. Logistic Regression
   - Predicts the label of a data point by a linear function
   - Accuracy score: `0.9569377990430622`

2. Support Vector Machine Classification
   - **Support-vector machines** (**SVMs**, also **support-vector networks**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.
   - Accuracy score: `0.8666267942583732`
3. Multinomial Naive Bayes
   - Naive Bayes Classifier Algorithm is a family of probabilistic algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of a feature.
   - Accuracy score: `0.9551435406698564`
4. KNeighborsClassifier
   - K-Nearest Neighbors is a method that simply looks at the observation that are nearest to the one it's trying to predict, and classifies the point of interest based on the majority of those around it.
   - Accuracy score: `0.9569377990430622`
5. Decision Tree Classifier

- Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility.
- Accuracy score: `0.9671052631578947`

6. Random Forest Classifier
    - A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging.
    - Accuracy score: `0.9778708133971292`

So ultimately, we chose the Random Forest Classifier.

Cross Validation

The results are : `0.97847534, 0.97668161, 0.97666068, 0.97127469, 0.97396768`

The average is : `0.9754120005474556`

Hyperparameter Tuning

We have used the GridSearchCV here to deduce the best parameters:
Result: `{'n_estimators': 31, 'random_state': 51}`

Summary

- Data Wrangling: We checked for all data types and also filled all the null values as 0.
- Exploratory Data Analysis showed us different kinds of relationships among different features.
- We tried 3 different models for sales prediction.
    1. Logistic Regression
    2. Support Vector Machine Classification
    3. Multinomial Naive Bayes
    4. Decision Tree Classifier
    5. KNeighborsClassifier
    6. Random Forest Classifier

Random Forest Regressor is the best predictor.