Università di Catania

Spam competition

Giovanni Spadaro

Università di Catania

Prof. Salvatore Ingrassia

May 12, 2023

Presentation Overview

The data

The provided data consists of a set of messages labeled as **ham** or **spam**.

	X	class	email
1	2480	spam	Sppok up ur mob with a Halloween ...
2	2909	spam	URGENT! Your Mobile number has been ...
3	4573	spam	\URGENT! This is the 2nd attempt to contact ...
4	3188	spam	This is the 2nd time we have tried 2 contact ...
5	1653	spam	For ur chance to win a å£250 cash ...
6	1970	spam	You have won a guaranteed å£200 award ...

Table: Head of the training data

The problem

The problem is to being able to detected if a message is **spam** or not using the logistic regression.

Since the logistic regression classifier cannot process text data, this task comes down to extract numerical feature from the text data.

The overall approach

1. **Text preprocessing**: removed punctuations and stopwords and converted the text to lowercase
2. **Message lengths**: Explored the message lenghts (string length, words count and numbers count)
3. **Word frequency**: Term frequency – inverse document frequency (TF-IDF)

Message lengths

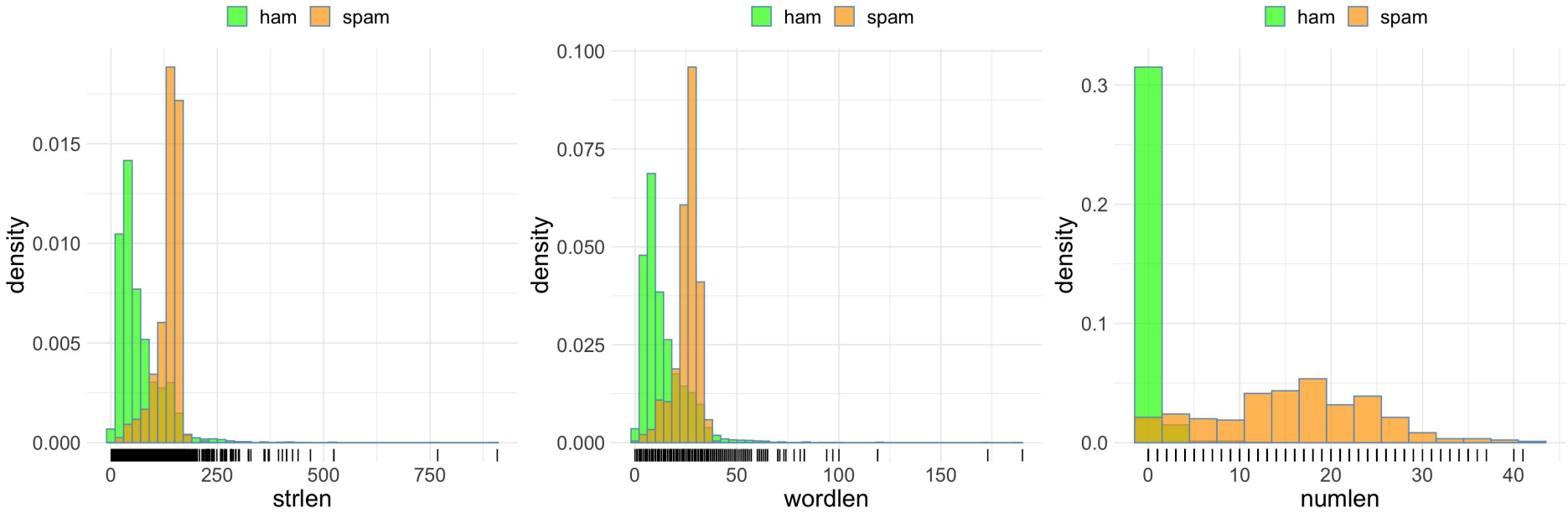


Figure: Left: Histogram of *string length*. Center: Histogram of *words count*. Right: Histogram of *numbers count*.

After having a look at the lengths distribution the string length and the number counts were choosen as features.

TF-IDF

This approach consist in finding a statistic for every term present in the training text. It start by defining 2 coefficients.

$$tf_{t,d} = \begin{cases} log(1 + f_{t,d}) & \text{if } f_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where t is the term, d is a bag of words, and $f_{t,d}$ is a frequency of the term in a bag.

$$idf_{t,D} = log\left(\frac{|D|}{|d \in D : t \in d|}\right) = \frac{N}{df_t} \tag{2}$$

where N is the cardinality of a corpus D (the total number of classes) and the denominator df_t is a number of bags where the term t appears.

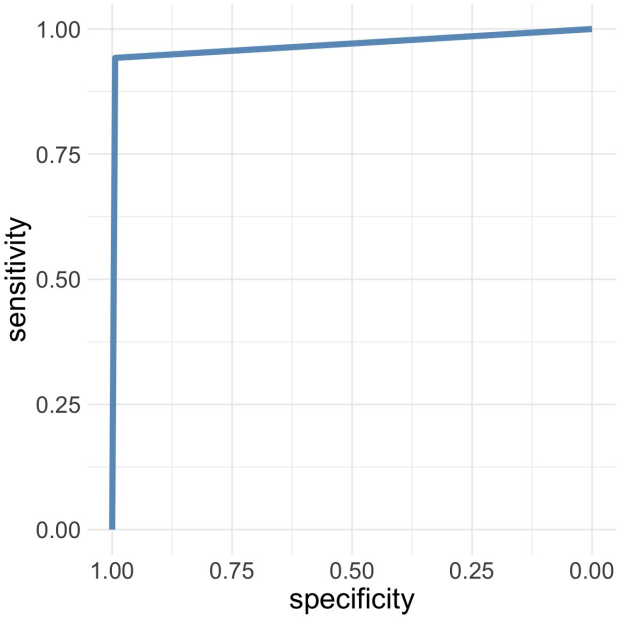
TF-IDF

Then the statistic is computed as it follows:

$$tfidf(t, d, D) = tf_{t,d} * tidf_{t,D} \tag{3}$$

Not all terms present in the messages were used, but only the top 40 ones, considering the count in the bag of words matrix.

The results



The model performance were validated using a train-test split of the training set (80%/20%).

On the left there's the ROC curve and on the bottom there are the main validation metrics.

Accuracy	Specificity	Sensitivity	AUC
0.988	0.994	0.951	0.968

References


jMotif (2016)
Term Frequency - Inverse Document Frequency statistics
https://jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html

Thanks for your attention

The source code for this project is available on GitHub