NLP: SPAM vs HAM

Paula Luvini, Facundo Marconi, Florencia Ludueña

Octubre 2020

1 Motivación

La detección de "spam" basados en el contenido del mail es una práctica muy popular conocida como spam vs. ham. El objetivo del siguiente trabajo es describir brevemente los clasificadores explorados en una competencia de Kaggle con dicho propósito, enfatizando en el de mayor score.

2 Procedimiento

En primer lugar realizamos un análisis exploratorio de la base de entrenamiento de Enron Corpus, que contaba con un total de 4000 emails. Observamos que la muestra no estaba balanceada dado que solo el 23% de los mails de la muestra de entrenamiento eran spam. También miramos las palabras más comunes y vimos que había algunas que se repetían con frecuencia en ambos grupos, algunos nombres propios como *Enron* o *Vince*, palabras como *Subject* y *ect*, y otras que califican como stopwords y que decidimos luego eliminar.
Comenzamos entonces a preprocesar el texto limpiando los caracteres de puntuación y la palabra *Subject* (que aparecía en el inicio de todos los emails mostrando el asunto). Además, normalizamos el texto considerándolo sólo en minúsculas y lemmatizando las palabras.
Respecto a otros features que consideramos agregar a la clasificación final consideramos la longitud de los emails y la cantidad de signos de puntuación en ambos grupos pero con un boxplot confirmamos que esta era similar en ambos casos por lo que no aportarían mucho a la clasificación. Realizamos también un análisis de *Parts of Speech*, en el que vimos que había diferencias en cantidades de sustantivos propios y de espacios entre el grupo de spam y ham. Agregamos esto al análisis pero al no dar mejores resultados de accuracy en la cross validación los descartamos. Por otro lado, en la "raíz" de la palabra también exploramos con *Stemming* pero la performance era relativamente menor en relación a cuando se utilizó *Lemming*.
Una vez realizado el pre-procesamiento con NLP generamos el vector TF IDF (Term Frequency Inverse Document Frequency), el cual expresa que tan relevante es una palabra para un email teniendo en cuenta todos los email. Este paso es clave para transformar los textos de los emails en una matriz de relevancia de términos numérica, la cual utilizamos para clasificar.
Finalmente para la clasificación utilizamos algoritmos de aprendizaje supervisado: SVM, KNN, RF, Regresión Logística y BOOSTING. Para la elección final del clasificador realizamos grid search y cross-validation con 5 folds. Seleccionamos el SVM con kernel radial y c=1 por tener mejor Accuracy.

3 Conclusiones

Es notable que con un pre-procesamiento de texto muy simple conseguíamos clasificar los emails con un accuracy siempre mayor al 80%. Respecto al pre-procesamiento creemos que tomamos un enfoque bastante completo e integral, acorde a lo visto en clase hasta el momento, probando distintos features y comparando entre los paquetes disponibles en *Python*. En el análisis descartamos varias features porque no había una diferencia significativa entre ambos grupos por lo que al proceso de clasificación no aportarían demasiado. Las que elegimos finalmente fueron más conservadoras en algún aspecto pero dado el contexto y las herramientas con las que contábamos creemos que resultaron la mejor opción.