Optimizando @RStatsJobsBot: un modelo de aprendizaje automático para clasificar tweets de ofertas de empleo.

Martin Rodriguez Nuñez[a], Juan Cruz Rodriguez[b]

[a] *CONICET, IMBIV, Universidad Nacional de Córdoba, Argentina*
[b] *FAMAF, Universidad Nacional de Córdoba, Argentina*

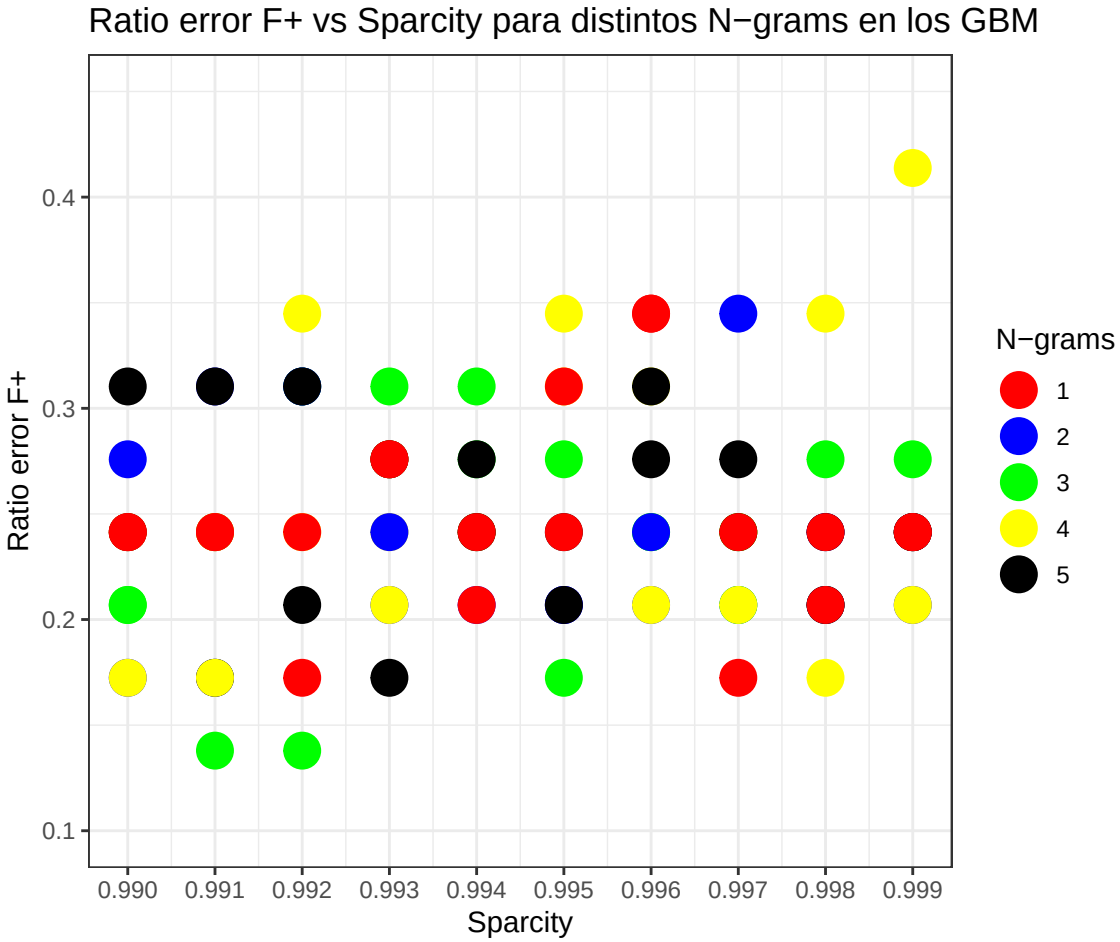
Abstract

Keywords: Mineria de texto, Bot, Twitter, Aprendizaje supervisado

@RStatsJobsBot es el bot de Twitter con más seguidores en el ámbito de la oferta de puestos de trabajo vinculados con R. Su propósito es compartir todos los tweets cuyo objeto sea publicitar vacantes laborales relacionados con el lenguaje de programación R, así facilitando la ardua tarea de buscar un nuevo empleo. Desde su creación, el bot implementó un conjunto de reglas de decisión creadas por su desarrollador para discriminar los tweets que eran verdaderas ofertas de trabajo de las que no lo eran. A pesar de contar con un algoritmo de decisión, el bot cometió muchos errores de clasificación que lo llevaron a publicar contenido erróneo. Desde sus inicios, todos los tweets fueron guardados y curados manualmente por su desarrollador. En la actualidad, existe una base de datos de 6783 tweets de los cuales 249 son verdaderas ofertas de empleo.
Previo al desarrollo de modelo predictivo, es necesario implementar técnicas de limpieza de texto sobre los diferentes tweets con el objetivo de eliminar toda aquello que aporte ruido y de transformar el texto de forma tal de maximizar el aprovechamiento de la información para el modelo predictivo. Para cumplir este objetivo es necesario convertir los datos a un **Corpus** y aplicar distintas técnicas de limpieza y transformación, las más importantes involucradas fueron: remover símbolos y puntuación, eliminar emails y páginas web añadiendo como variable respuesta el número de cada una de ellas en cada tweet, convertir a minúsculas, quitar stop words y realizar stemming, llevando las palabras a su word stem.
Para desarrollar el modelo predictivo es necesario convertir el texto procesado en una *Document Term Matrix* (DTM). A la hora de llevar a cabo el ajuste del modelo surgen dos incógnitas. La primera de ellas se encuentra referida a cuál es el mejor algoritmo de clasificación que ajusta a los datos y la otra se encuentra relacionada a la determinación de los hiper parámetros que fijan

Email addresses: martinrnu@gmail.com (Martin Rodriguez Nuñez), jcrodriguez@unc.edu.ar (Juan Cruz Rodriguez)

la dimensión de la **DTM**. Estos hiper parámetros son el número máximo de **N-grams** a considerar y la **sparsity** que se tomara. Para determinarlos se ajustaron distintos algoritmos de clasificación a todas las combinaciones posibles de estos hiper parámetros, analizando cuál era el que lograba el mejor ajuste. Para el entrenamiento de los distintos algoritmos se procedió a implementar la herramienta de **AutoML** perteneciente al paquete de **h2o**, tomando como métrica a optimizar el **mean_per_class_error**. Para el procedimiento de ajuste se optó por dividir los datos en un 75/10/15 para entrenamiento, testeo y validación. El ajuste de los diferentes algoritmos predictivos se realizó por medio de **10 fold cross validation** y la selección de los mejores modelos se realizó a través del desempeño en la base de datos de **testeo**. El problema de clasificación presentado tiene una variable respuesta con dos posibles categorías: propuesta de trabajo falsa (false) y propuesta de trabajo verdadera (true). El objetivo principal fue minimizar la **ratio de error para los falsos negativos** (propuestas de trabajo verdaderas clasificadas como falsas), además también se buscó minimizar la **ratio de error para los falsos positivos** (propuestas de trabajo falsas clasificadas como verdaderas). Los resultados arrojados por el procedimiento empleado posicionaron el algoritmo de **Gradient Boosting Machine (GBM)** como el mejor algoritmo de clasificación para esta variable respuesta. La combinación de hiper parámetros que minimizan los *false positives* en el set de datos de testeo fue de considerar hasta un máximo de 3 grams con una sparsity de 0.992 y la que minimizó los *false negatives* fue de 2 grams como máximo con una sparsity de 0.997. Para maximizar la capacidad predictiva se planteó unificar ambos modelos en un **ensemble**, con lo cual la clasificación se realiza en dos pasos, primero se clasifica con el modelo que mejor identifica los *true positives* (menor error en *false negatives*) y luego se filtra con el que mejor identifica *true negatives* (menor error en *false positives*).



El objetivo de esta *flash talk* es mostrar el análisis de minería de textos y aprendizaje automático (Silge 2017, Hvitfeldt 2021) que llevó a la definición de un modelo de aprendizaje automático que discrimina qué tweets que son ofertas de trabajo verdaderas y cuáles no.

Referencias

- Silge, J., & Robinson, D. (2017). Text mining with R: A tidy approach. O'Reilly Media, Inc.
- Hvitfeldt, E. & Silge, J. (2021) Supervised Machine Learning for Text Analysis in R. CRC press.