Olivier Gruss

Trabajo Práctico 3-Orga de Datos-Reentrega

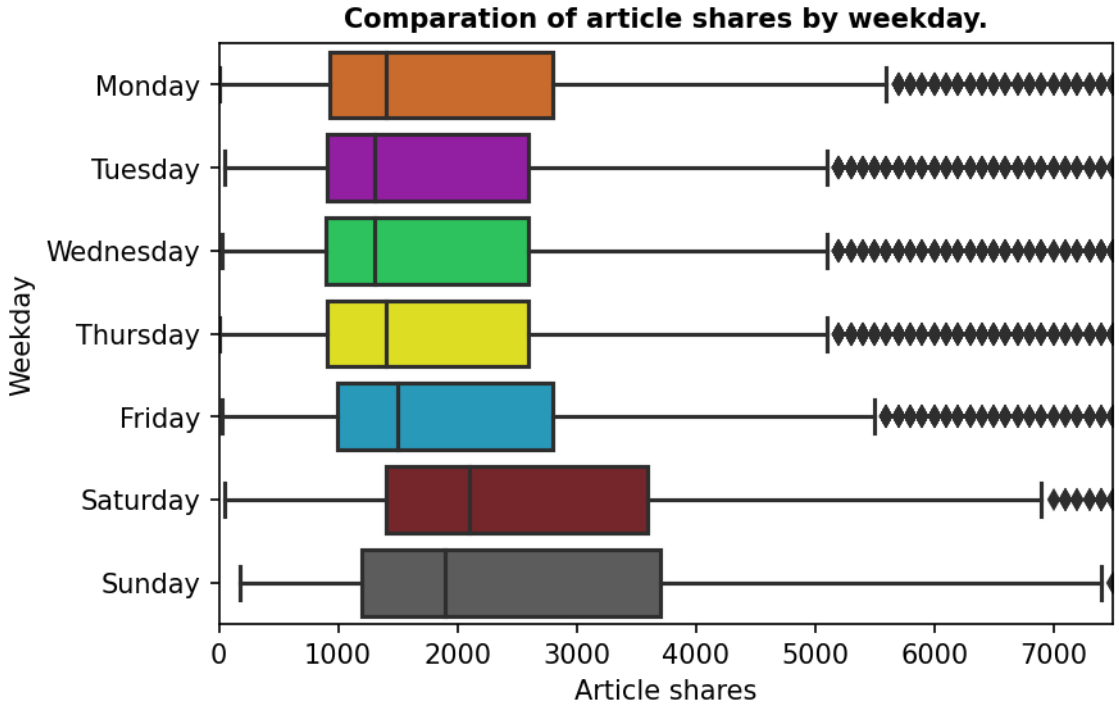
Los archivos utilizados para este TP se encontrarán en la carpeta

Parte I-Análisis exploratorio

Deberán realizar 6 visualizaciones interesantes que ayuden a explicar el target haciendo almenos un plot de cada uno de los siguientes tipos:

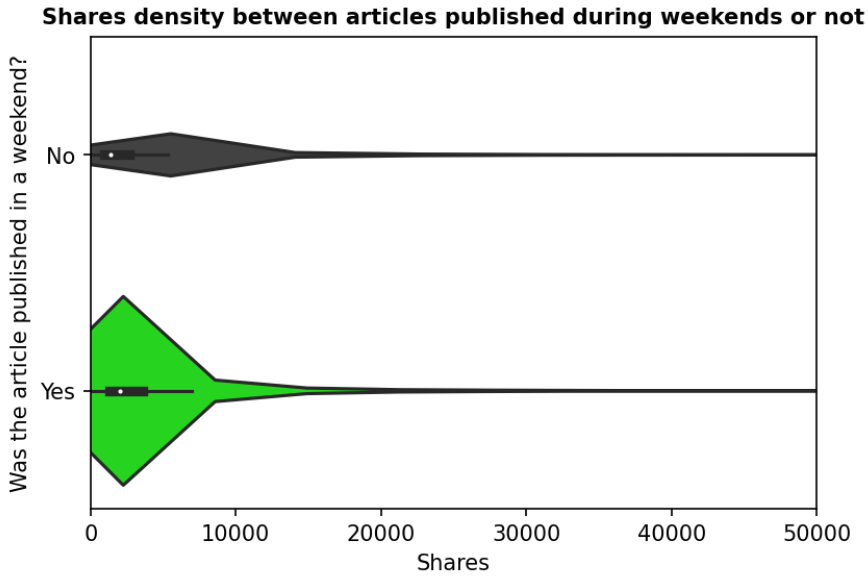
- Bar plot (o stacked bar plot o variaciones)
- Violin plot
- Box plot
- Heatmap

Box plot: https://colab.research.google.com/drive/1vXDs5sX1OU4pGLBHIy_-crtaU5wvKCpt?authuser=1

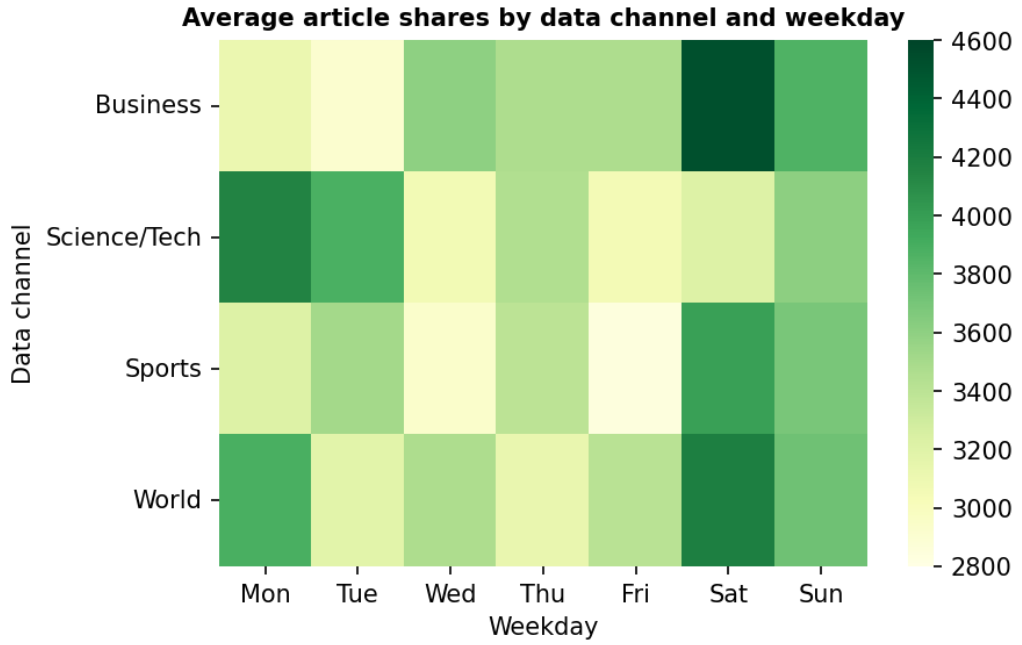


Violin plot:
https://colab.research.google.com/drive/1HZes7EPwyNYpgQUrX7i1oY7748ETmpU3?authuser=1#scrollTo=kT_-bR21O1dE

Olivier Gruss

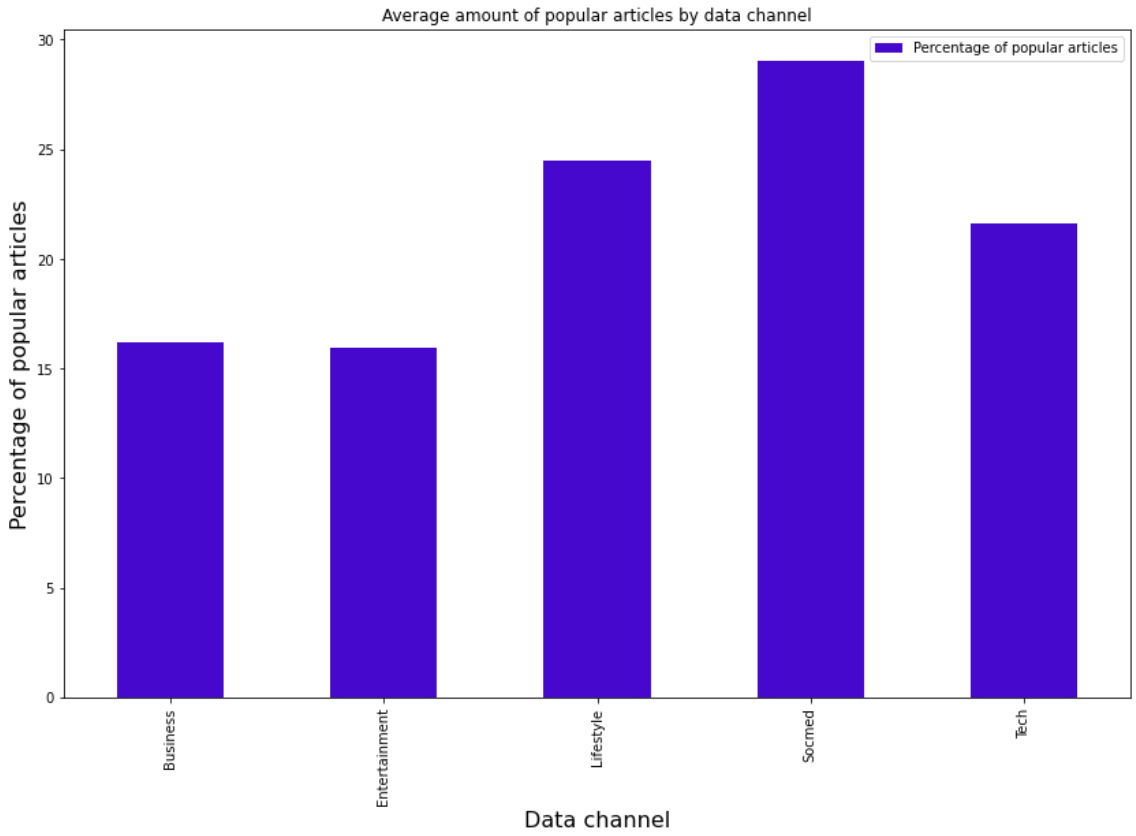


Visualización 3 (Heat map):
https://colab.research.google.com/drive/174RljlhgOO9vapt75jGNNLwCIN2C4AAb?authuser=1



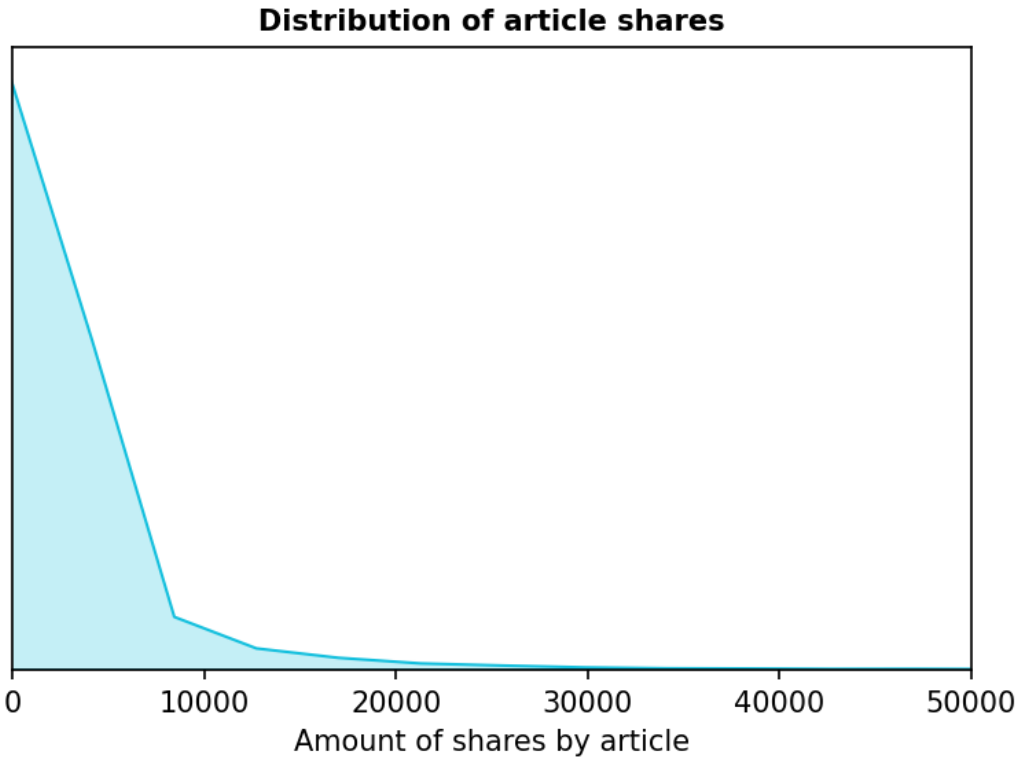
Bar plot: https://colab.research.google.com/drive/1Hy0MmsJotWf5nNIQak0_-RcQm5PsltbS?authuser=1#scrollTo=WZvMoj9NYaee

Olivier Gruss

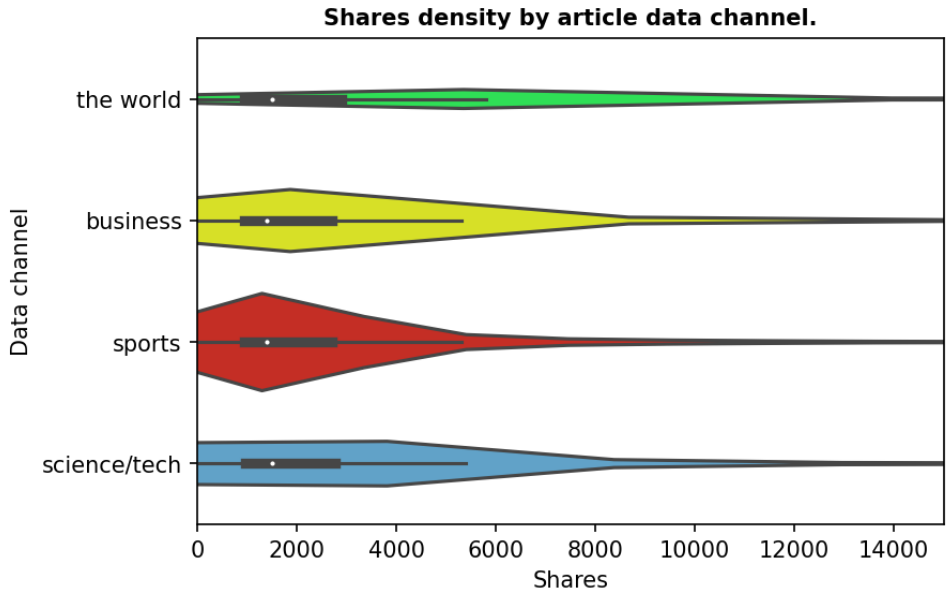


Olivier Gruss

Density plot: https://colab.research.google.com/drive/11kY3gSHgv0VgFN7gxoY5-XV1lkiIxe44?authuser=1



Visualización 6:
https://colab.research.google.com/drive/1CRfQWvp8a1_yxFktrQqjZ4KCM2ckHdSh?authuser=1



Olivier Gruss

Parte II-Machine Learning Baseline

Vamos a construir un modelo muy sencillo para saber qué es lo peor que podemos hacer, en general esta es una tarea muy importante que queremos que repitan en sus proyectos de machine learning. ¿Por qué?

- Navaja de Ockam: “Cuando se ofrecen dos o más explicaciones de un fenómeno, es preferible la explicación completa más simple; es decir, no deben multiplicarse las entidades sin necesidad.” ¿Para qué desarrollar un modelo super complejo si capaz es peor o casi igual que uno muy sencillo?
- Nos sirve para saber si estamos usando bien los modelos más complejos, si su score nos da peor al baseline probablemente se deba a un error de código.
- Nos sirve para rápidamente saber que tan complejo es un problema.
- Los modelos simples son fáciles de entender.

Utilice todas las columnas del dataset (exceptuando columnas que no tenga sentido usar para predecir) con algún encoding donde sea necesario para entrenar una regresión logística, utilizando búsqueda de hiperparametros y garantizando la reproducibilidad de los resultados cuando el notebook corriera varias veces. Conteste las preguntas:

- ¿Cuál es el mejor score de validación obtenido? (¿Cómo conviene obtener el dataset para validar?)
- Al predecir con este modelo para test, ¿Cúal es el score obtenido? (guardar el csv con predicciones para entregarlo después)
- ¿Qué features son los más importantes para predecir con el mejor modelo? Graficar.

Solución:

La mejor puntuación obtenida en el set de validación es de: 0.7058089547690566.

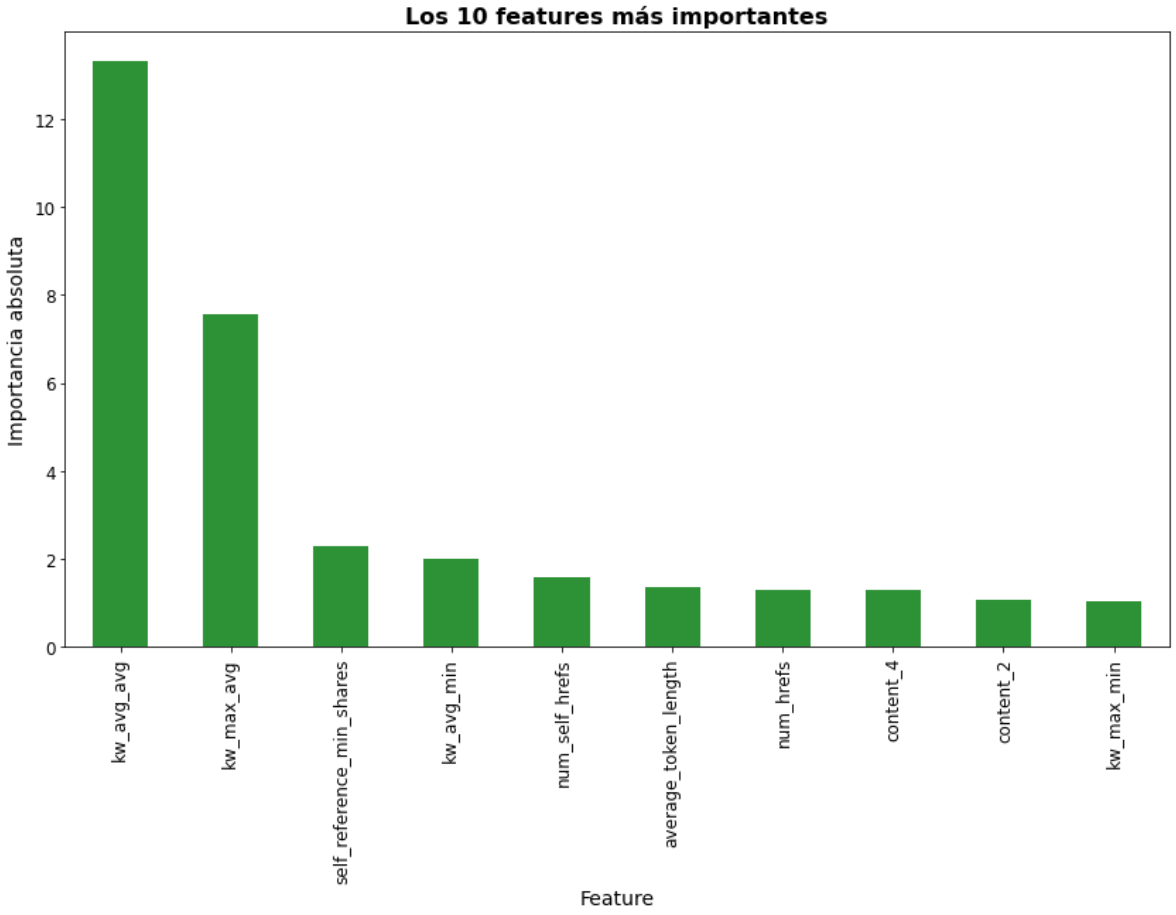
Para separar el dataset de test del set de validación simplemente tome el 10% final del dataset de test ya que, este ya venía ordenado por su parámetro timedelta por ende, no habrá problemas de time-traveling.

Al predecir con el dataset de test el score obtenido es de: 0.701199529877031

El csv con los resultados de la predicción está en el siguiente link:
https://drive.google.com/file/d/14XRII6K1EjTG71GoP7N50E7b5a9seJRk/view?usp=sharing

Gráfico con los features más importantes:

Olivier Gruss



Link al colaboratory: https://colab.research.google.com/drive/1sWaY0A9cp24KAkNp-VTllbtKmVhyyamR?authuser=1#scrollTo=mI9DJpqtHw1k

Parte III-Machine Learning

Entrenar 2 (de tipos distintos, excluyendo regresiones logísticas) modelos (5 puntos cada uno) con búsqueda de hiperparametros (¿cómo conviene elegir los datos de validación respecto de los de train?). Los modelos deben cumplir las siguientes condiciones:

- Deben utilizar AUC-ROC como métrica de validación.
- Deben medirse solo en validación, no contra test!!!
- Deben ser reproducibles (correr el notebook varias veces no afecta al resultado).
- Deben tener un score en validación superior a 0,7.
- Para el feature engineering debe utilizarse imputación de nulos, mean encoding y one hot encoding al menos una vez cada uno.
- Deben utilizar al menos 40 features (contando cómo features columnas con números, pueden venir varios de la misma variable).
- Deben utilizar CountVectorizer o TfIdfVectorizer para algunos features.

Olivier Gruss

- Deberán contestar la siguiente pregunta: Para el mejor modelo de ambos, ¿cuál es el score en test? (guardar el csv con predicciones para entregarlo después).

Solución:

Para la parte III del TP creé un modelo de RandomForestClassifier y otro de XGBClassifier.

El mejor score de validación obtenido con el modelo RandomForestClassifier es de: 0.7238572547995299.

El mejor score de validación obtenido con el modelo XGBClassifier es de: 0.7340630577684908.

El score en test de XGBClassifier (tiene mejor score en valid que RandomForestClassifier) es de: 0.7223654310701518

El csv con los resultados de RandomForestClassifier está en el siguiente link: https://drive.google.com/file/d/1HsVJ4K02lozuhwvRlknCSiJprp3Ws86m/view?usp=sharing

El csv con los resultados de XGBClassifier está en el siguiente link: https://drive.google.com/file/d/1iBKvJWXZNLEhWEozrHgU2yhQeTXR2ssw/view?usp=sharing

Link al colaboratory de XGBClassifier: https://colab.research.google.com/drive/1VtA6ckFQf0HhyuC_XKQLhlOrRSNqN1X-?authuser=1#scrollTo=8HDOsCw4_ni1

Link al colaboratory de RandomForestClassifier: https://colab.research.google.com/drive/113kBOgA7M6SbLNntHe7BnYhgPfqDSrOu?authuser=1#scrollTo=1xQTRXw0gsFl

Ejercicio extra

Se me pidió realizar el siguiente ejercicio extra: "Entrenar una regresión sobre el campo shares. ¿Cúal es su MSE en test y train? ¿Qué tanta accuracy tiene sobre popular un sistema de regresión que luego pasa a binario por medio del percentil 80 de shares?"

Decidí entrenar un KNeighborsRegressor.

El mejor score de MSE obtenido con el set de validación es de: 76278297.69534506.

El accuracy score obtenido sobre popular con el percentil 80 de shares en el set de validación es de: 0.7467077612776688.

El score de MSE obtenido con el set de test es de: 64048500.12995095.

El accuracy score obtenido sobre popular con el percentil 80 de shares en el set de test es de: 0.7513187641296156.

Olivier Gruss

Link al ejercicio: https://colab.research.google.com/drive/1-nYr8fSiIj4YPZFW9py4HQ3DRe0-Kc5d?authuser=1#scrollTo=xEyQ8pJeym4K