Drug Review Dataset
Examen general de conocimientos

López Velasco Jossé Armando

Marzo 14, 2022

1 Introducción

1.1 Análisis previo

El dataset utilizado para este examen general consta de 2 archivos tsv, el análisis que se muestra a continuación fue realizado sobre el conjunto de entrenamiento (drugsComTrain_raw.tsv)[1]. Este archivo presenta la siguiente lista de atributos:

- drugName: Nombre del medicamento
- condition: Enfermedad o padecimiento que el paciente presenta
- review: Reseña en formato texto para el medicamento prescrito.
- rating: Calificación en una escala del 1-10 que el paciente otorga al medicamento
- date: Fecha en la cual la reseña fue capturada
- usefulCount: No especificado

	drugName	condition	review	rating	date	usefulCount
206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9.0	2012-05-20	27
95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8.0	2010-04-27	192
92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5.0	2009-12-14	17
138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8.0	2015-11-03	10
35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9.0	2016-11-27	37

Figure 1: Muestra del dataset

El tamaño total del archivo es de 161297 registros, de los cuales la distribución de reseñas por medicamento se puede apreciar en la figura 2

El padecimiento más frecuente en la población corresponde a **Birth Control** con 28788 reseñas para distintos medicamentos, seguido de depresión, dolor, ansiedad, acné, etc. Los cuales se muestran en 3+ Existe una correlación rela-

Levonorgestrel	3657
Etonogestrel	3336
Ethinyl estradiol / norethindrone	2850
Nexplanon	2156
Ethinyl estradiol / norgestimate	2117
Ethinyl estradiol / levonorgestrel	1888
Phentermine	1543
Sertraline	1360
Escitalopram	1292
Mirena	1242
Implanon	1102
Gabapentin	1047
Bupropion	1022
Venlafaxine	1016
Miconazole	1000
Medroxyprogesterone	995
Citalopram	995
Lexapro	952
Bupropion / naltrexone	950
Duloxetine	934

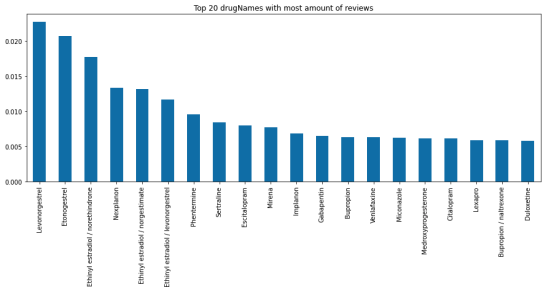

Figure 2: Top 20 medicamentos con mas reseñas

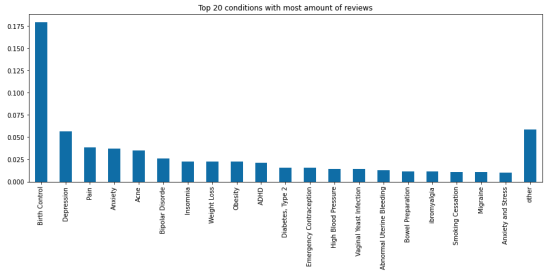

Figure 3: Top 20 condiciones con mas reseñas

	rating	usefulCount
rating	1.000000	0.234185
usefulCount	0.234185	1.000000

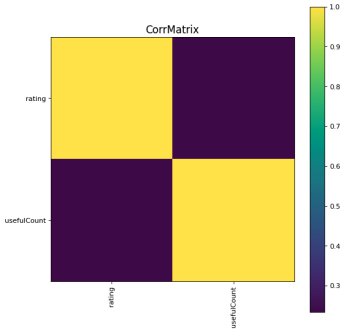

Figure 4: Matriz de correlación

tivamente alta entre las dos variables (rating, usefulCount) como se aprecia en la figura 4

Durante toda la historia que se tiene del medicamento **Levonorgestrel** se aprecia una aceptación buena para padecimientos como **Birth Control y**

[1]https://archive.ics.uci.edu/ml/machine-learning-databases/00462/

Emergency Contraception, sin embargo su aceptación se encuentra en lugares críticos para finales del 2017 para el padecimiento **Abnormal Uterine Bleeding**, esto se puede apreciar de una manera más clara en la figura 5 Al-

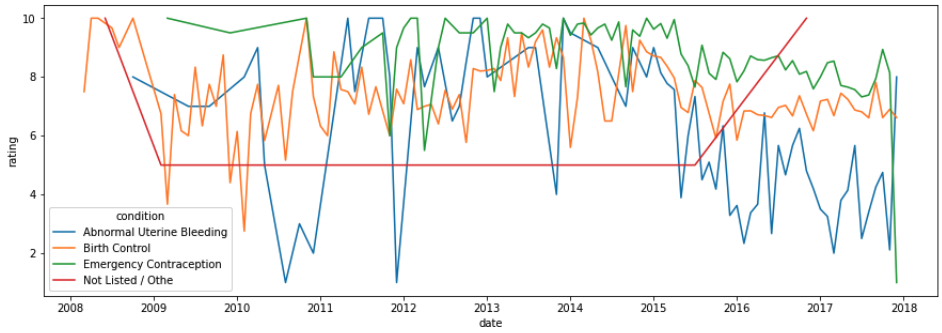

Figure 5: Historia mensual de los medicamentos con mejor calificación

gunos casos críticos que se pueden apreciar es la caída en la aceptación de los medicamentos para ciertas condiciones, como es el caso de **ansiedad y depresión**; ambos han estado presentando una caída en su promedio de rating mensual, para el caso de la **depresión** obtuvo una puntuación máxima de 9 a mediados del 2008 hasta una puntuación de 6 presentada a finales del 2017. Algo similar ocurre con la **ansiedad** cuya tendencia bajista es notoria desde el 2015 pasando de una aceptación de 9.5 hasta un mínimo de 6 para finales del 2017. Lo podemos apreciar de una manera más clara en la figura 6 El rating de

Figure 6: Promedio mensual de rating para depresión y ansiedad

los medicamentos dada una condición ha seguido bajando durante los últimos años 7

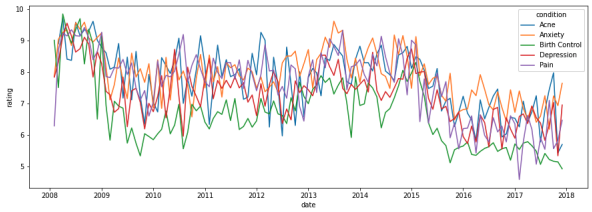

Figure 7: Promedio mensual de rating para top condiciones

2 Análisis del modelo de clasificación para NPS

Dado que tenemos el rating en una escala del 0-10 podemos construir un modelo que identifique promotores y detractores.

2.1 NPS

NPS (Net Promoter Score) es un indicador utilizado para medir la experiencia del usuario, gracias a este escore podemos identificar qué tan probable es que el medicamento sea recomendado por el consumidor. Se calcula con base a estas sencillas reglas:

- Promotores: Puntuación de 9 o 10
- Neutros: Puntuación de 7 u 8
- Detractor: Puntuación de 0-6

2.2 Resultados NPS model

Se utilizó una regresión logística con el fin de obtener un modelo rápido del cual podíamos obtener información útil sobre el comportamiento del texto para una predicción de la clase promotor.

Sobre el archivo de train se realizó un split del 20% para test, los resultados que se muestran a continuación refieren a este subconjunto de datos.

2.2.1 AUC

El modelo obtuvo un auc en el conjunto de prueba del **0.73** y la siguiente matriz de confusión 8

2.2.2 SHAP values

Las palabras que aportan más a una predicción positiva son [year, life, great, work] 9

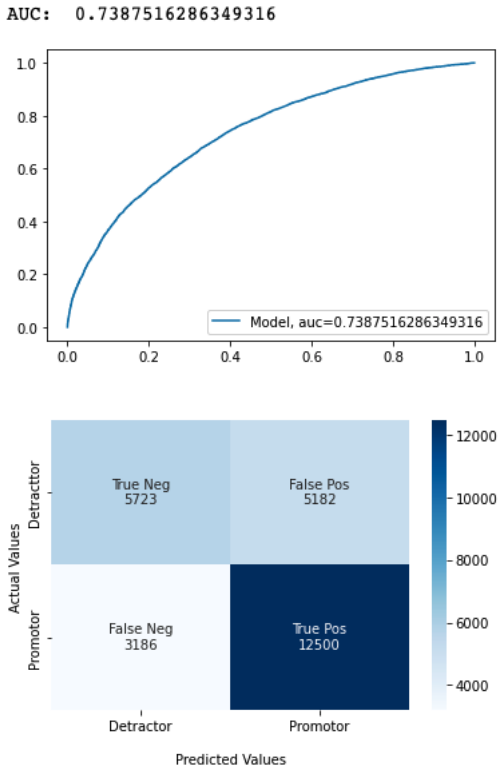

Figure 8: AUC y matriz de confusión para NPS

3 RNN

3.1 LSTM

3.2 Método

Para realizar la tarea central del examen se utilizó una LSTM **?**; es una red neuronal recurrente cuya arquitectura funciona muy bien con secuencias de datos debido a sus conexiones en "reversa". Esta arquitectura está compuesta básicamente de

- Una celda
- Compuerta de entrada
- Compuerta de salida
- Compuerta del olvido

El flujo de datos que se llevó a cabo fue el que se describe a continuación:

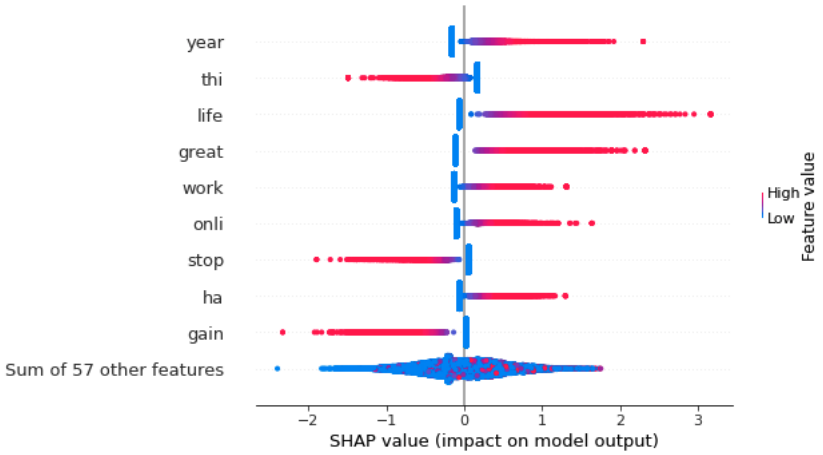

Figure 9: Shap values para modelo de nps

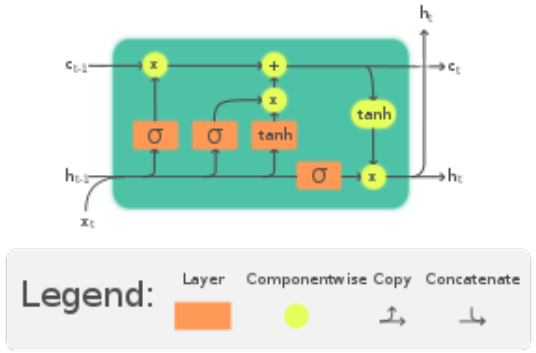

Figure 10: arquitectura de una LSTM

3.2.1 1.- Limpiar el texto

En este punto se realizaron sub-tareas necesarias para mejorar el rendimiento de la red, ya que si no se lleva un proceso adecuado podríamos terminar con un corpus demasiado grande que no aporte información necesaria al modelo. El punto de interés debe radicar en llegar a una representación vectorial que será alimentada a la LSTM. Para esto se realizó:

- Eliminación de "stop words": Este es un proceso necesario ya que palabras como [it, this, that, I, He, etc] no nos aportan mucha información sobre el punto central de la reseña, y además suelen representar la mayor cantidad de frecuencias debido a que son muy necesarios para conectar ideas.
- Eliminación de caracteres especiales

- Vectorizar: Transformar una oración en un vector

3.3 Vectorización del texto en secuencias

Para alimentar una red LSTM es muy importante preservar el orden del texto, de esta manera la red será capaz de trabajar de manera adecuada con el uso del contexto de una palabra. Para transformar el texto a la entrada de la red se realizó lo siguiente:

1. Tokenización del corpus: utilizando tf.keras.preprocessing.text.Tokenizer lo que realizamos es identificar una cantidad máxima de palabras de 2500, las cuales, al recibir el texto de entrenamiento las tokeniza y asigna un índice entero, por el cual será identificado.

2. Crear la secuencia: Una vez hemos transformado cada token a un índice entero, es necesario re-construir la oración manteniendo el orden en el cual estas palabras tokenizadas aparecen en la oración, de esta manera utilizando el tokenizador, y convirtiéndolo a secuencias, para luego hacer un paddding (mantiene el mismo tamaño de vector) obtenemos ejemplos con un máximo de 2500 palabras consideradas en vectores de tamaño 200.

3.3.1 Arquitectura

La LSTM utilizada es la que se muestra en :

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 200, 128)          320000
_________________________________________________________________
lstm_2 (LSTM)                (None, 4)                 2128
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 15
=================================================================
Total params: 322,143
Trainable params: 322,143
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 11: arquitectura utilizada

4 Resultados

Se obtuvo un auc de 0.69 en el conjunto de validación. En **?** podrá encontrar todos los resultados y una comparación con un modelo Naive multiclase con el mismo pre-procesamiento de datos.

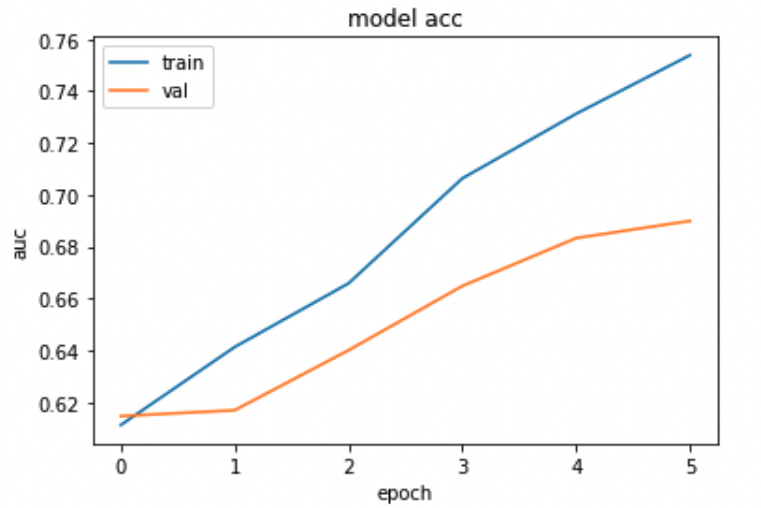

Figure 12: Accuracy Validación 3 clases

	precision	recall	f1-score	support
0	0.71429	0.08547	0.15267	117
1	0.33333	0.12264	0.17931	212
2	0.43504	0.81550	0.56739	271
micro avg	0.42833	0.42833	0.42833	600
macro avg	0.49422	0.34120	0.29979	600
weighted avg	0.45356	0.42833	0.34940	600
samples avg	0.42833	0.42833	0.42833	600

Figure 13: Reporte de precisión y sensibilidad para las tres clases donde 0:Detractor, 1:Neutro, 2:Promotor

5 Siguientes pasos

El modelo resultante es muy lento en su entrenamiento, por lo que se necesita una parte de fine-tunning de parámetros y además exteneder el tamaño del

dataset de entrenamiento y entrenar por más épocas.

Adicionalmente se implementó un modelo pre-entrenado de DistilBert[2] con el fin de implementar una arquitectura más novedosa y ligera(comparada con BERT), se utilizó el pre entrenamiento de **distilbert-base-uncased-finetuned-sst-2-english**; los resultados obtenidos tienen el mismo problema que el modelo propuesto, faltan épocas de entrenamiento. Sin embargo, lo observado en las primeras 10 épocas indican necesario un learning rate mas bajo, y entrenar por más épocas. Esta arquitectura es la siguiente:

```
Model: "model_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_ids (InputLayer)          [(None, 200)]        0
__________________________________________________________________________________________________
attention_mask (InputLayer)     [(None, 200)]        0
__________________________________________________________________________________________________
bert (TFBertMainLayer)          TFBaseModelOutputWit 109482240   input_ids[0][0]
                                                                 attention_mask[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 512)          393728      bert[1][1]
__________________________________________________________________________________________________
dropout_38 (Dropout)            (None, 512)          0           dense_1[0][0]
__________________________________________________________________________________________________
outputs (Dense)                 (None, 3)            1539        dropout_38[0][0]
==================================================================================================
Total params: 109,877,507
Trainable params: 395,267
Non-trainable params: 109,482,240
__________________________________________________________________________________________________
```

Figure 14: Arquitectura DistilBERT

[2]'https://arxiv.org/abs/1910.01108: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

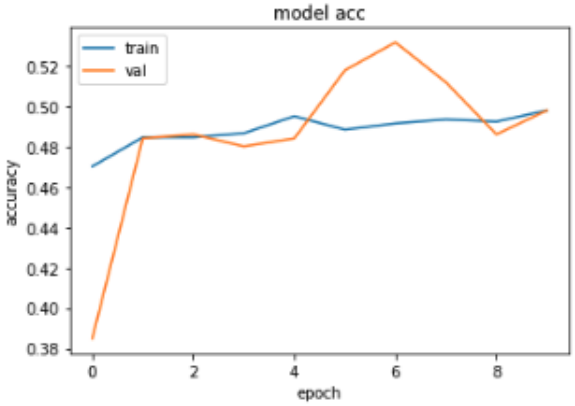

Figure 15: Gráfica de accuracy para distilBERT

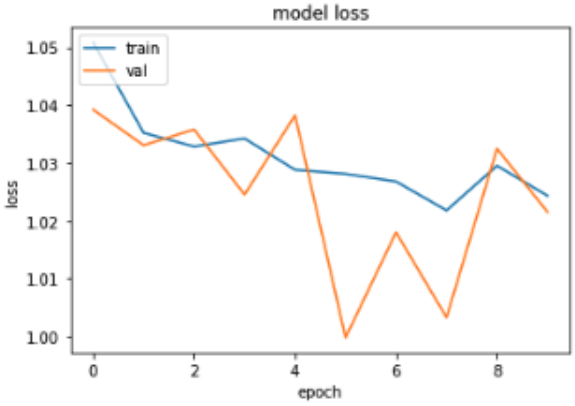

Figure 16: Gráfica de pérdida para distilBERT