Taller 1 - Aprendizaje Profundo

Elkin Daniel Prada Gómez

21 de marzo de 2023

Resumen

El siguiente trabajo tiene como objetivo principal realizar un análisis de clasificación de un conjunto de datos relacionados con tweets publicados por parte de usuarios de Twitter, con el fin de determinar si dichos tweets provienen de un hombre, mujer o una marca, empleando redes neuronales feed-forward y de esta manera predecir dicha categoría. Para realizar este trabajo se descarga un dataset de la pagina Kaggle, este dataset contiene 26 variables que serán analizadas a continuación. Para realizar el análisis de clasificación, se trabajaran 4 etapas, comprensión del dataset, limpieza de datos, utilización de variables categóricas y finalmente la etapa de ajustes.

1. Comprensión del dataset

Para esta etapa del taller, se descarga el dataset "Twitter User Gender Classificatio"de kaggle y se procede con la revisión de cada una de sus variables. El dataset cuenta con 20050 registros de tweets realizados por diferentes usuarios y 26 variables que describen atributos del usuario dentro de Twitter.

1.1. Descripción de variables

Dentro de las 26 columnas encontradas del dataset de kaggle, se puede apreciar información del perfil del usuario y de la publicación realizada. De esta manera también se identifica el tipo de dato de la variable y si es una variable que brinda una característica para la construcción de la red neuronal. La variable objetivo con la que se trabajará para la predicción es "género", posterior a identificar las variables y su tipo de dato, se procede a identificar variables importantes para la construcción del modelo. Para este caso, es necesario realizar un análisis de distribución y correlación de las variables que poseen información importante para el modelo.

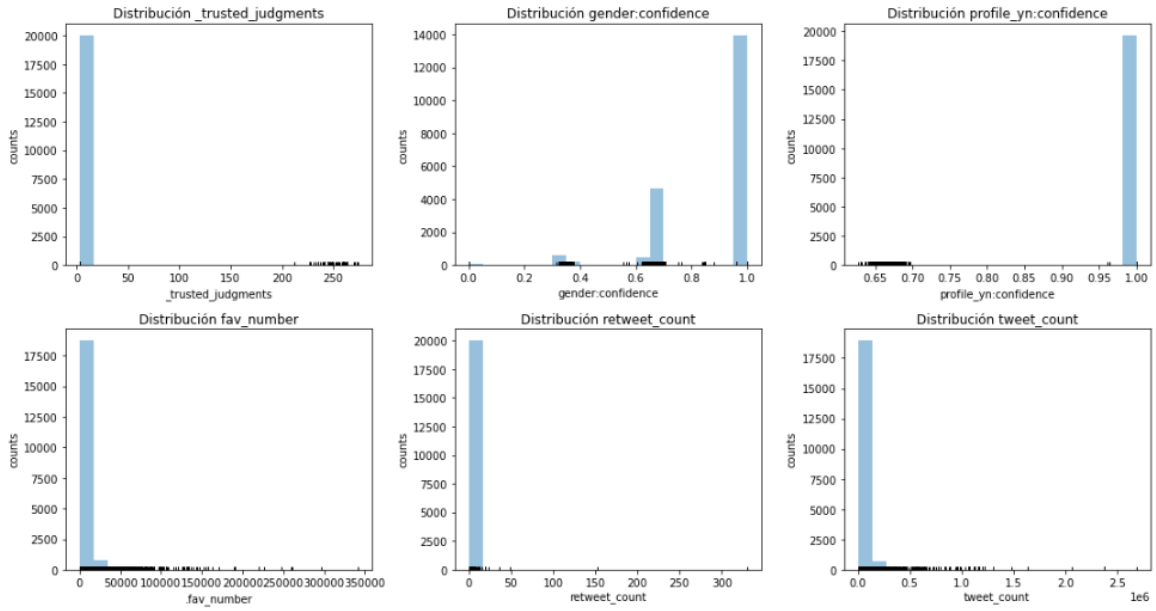


Figura 1: Gráfica de distribución mediante histograma para las variables numéricas.

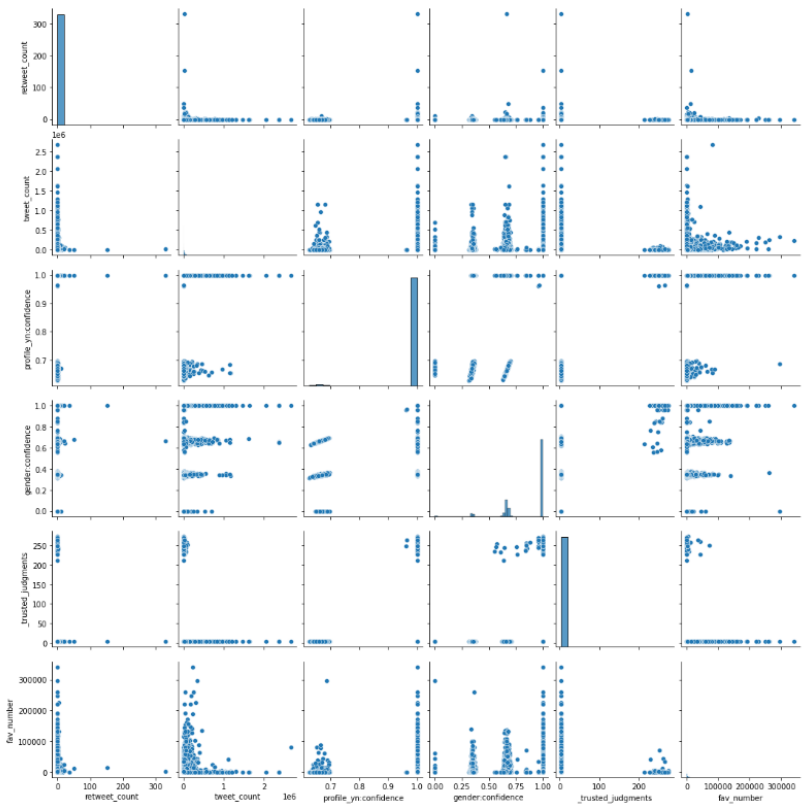
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20050 entries, 0 to 20049
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   _unit_id              20050 non-null  int64
 1   _golden               20050 non-null  bool
 2   _unit_state           20050 non-null  object
 3   _trusted_judgments    20050 non-null  int64
 4   _last_judgment_at     20000 non-null  object
 5   gender                19953 non-null  object
 6   gender:confidence     20024 non-null  float64
 7   profile_yn            20050 non-null  object
 8   profile_yn:confidence 20050 non-null  float64
 9   created               20050 non-null  object
 10  description           16306 non-null  object
 11  fav_number            20050 non-null  int64
 12  gender_gold           50 non-null     object
 13  link_color            20050 non-null  object
 14  name                  20050 non-null  object
 15  profile_yn_gold       50 non-null     object
 16  profileimage          20050 non-null  object
 17  retweet_count         20050 non-null  int64
 18  sidebar_color         20050 non-null  object
 19  text                  20050 non-null  object
 20  tweet_coord           159 non-null    object
 21  tweet_count           20050 non-null  int64
 22  tweet_created         20050 non-null  object
 23  tweet_id              20050 non-null  float64
 24  tweet_location        12566 non-null  object
 25  user_timezone         12252 non-null  object
dtypes: bool(1), float64(3), int64(5), object(17)
memory usage: 3.8+ MB
```

1.2. Distribución y correlación

Para realizar un análisis de distribución se separan las variables en datos cuantitativos y datos cualitativos. A continuación, en la siguiente figura se aprecia un histograma que permite observar la distribución de las variables cuantitativas excluyendo las variables ids '_unit_id' y 'tweet_id' las cuales no son relevantes para el modelo ya que solo representan un código consecutivo.

Variables contenidas en el data set descargado		
ID Variable	Nombre variable	Tipo de Dato
1	_unit_id	Numérico
2	_golden	Booleano
3	_unit_state	String
4	_trusted_judgments	Numérico
5	_last_judgment_at	String
6	gender	String
7	gender:confidence	Numérico
8	profile_yn	String
9	profile_yn:confidence	Numérico
10	created	String
11	description	String
12	fav_number	Numérico
13	gender_gold	String
14	link_color	String
15	name	String
16	profile_yn_gold	String
17	profileimage	String
18	retweet_count	Numérico
19	idebar_color	String
20	text	String
21	tweet_coord	String
22	tweet_count	Numérico
23	tweet_created	String
24	tweet_id	Numérico
25	tweet_location	String
26	user_timezone	String

Cuadro 1: Variables contenidas en el data set descargado.



2. Limpieza de datos

En esta etapa de limpieza de datos, es necesario definir las variables con las que se trabajará el modelo, descartando aquellas que no brindan información relevante. La variable genero es nuestra variable objetivo, por lo que será la variable principal donde se revisarán sus valores y relación con las demás variables. Se eliminan los datos desconocidos o que no entran el la categoría de Hombre y Mujer de esta variable, con el fin de poder obtener los registros que pertenecen tanto a 'Mujer' como 'Hombre' y poder tener los datos de entrenamiento.

A continuación se listan los pasos de limpieza de datos:

- Selección de variable objetivo
- Limpieza de la columna gender eliminando valores vacíos, desconocidos y que no están en las categorías 'male', 'female' y 'brand'.
- Excluir variables ids, las cuales representan un código consecutivo pero no es relevante para el modelo, estás son: _unit_id y tweet_id.
- Excluir variables que no brindan información acerca del perfil del usuario o de la publicación, ya que no son variables controlables, dependen de una evaluación y no brindan una característica significativa del perfil del usuario. Estás variables a excluir son _unit_state, _trusted_judgments, _last_judgment_at, gender:confidence, profile_yn y profile_yn:confidence
- Los valores de la variable _golden son falsos en la mayoría de los registros, está variable también se elimina, al igual que profileimage que contiene la url de las imágenes del perfil de los usuarios, pero estás en su mayoría no corresponden con el genero del usuario.
- Se descartan las variables description, name, text, user_timezone, created, gender_gold y profile_yn_gold, ya que manejan un tipo de dato complejo para el modelo y no brindan la información puntual que determine una predicción de la variable gender. Al descartar estás varibles, tambíen se eliminan las variables tweet_conunt, tweet_created y tweet_location, ya que dependen de las descartadas anteriormente.
- Finalmente se conservan las variables fav_number, link_color, retweet_count y sidebar_color que brindan las caracteristicas que se buscan y que pueden determinar una relación con la variable a predecir. Sin embargo, estás variables se deben analizar para limpiar valores que compliquen el modelo.
- Debido a que deseamos determinar si el tweet proviene de un hombre o de una mujer, se descartan las demás opciones de la variable gender.
- Se eliminan los valores nulos de los datos.
- Para poder trabajar de una mejor manera con la variable 'gender' se convierten los valores 'male' y 'female' en datos númericos o más especificamente en un valor booleano donde male = 1 y female = 0.

3. Utilización variables categóricas

Para realizar una transformación de las variables categóricas seleccionadas, es necesario revisar los valores de las mismas y así poder determinar que tipo de transformación se debe aplicar para generar una mejor facilidad de la implementación del modelo. En este caso como las variables link_color y sidebar_color poseen muchas categorías dentro de sus valores, se determina dejar unos colores fijos establecidos que abarquen un grupo amplio de registros y las categorías sean menores. Quedando de la siguiente manera:

link_color: azul_por_defecto, azul, verde, morado, rojo, rosado, negro y diferente.
sidebar_color: azul_por_defecto, negro, blanco, gris, azul_claro, azul-oscuro, verde, rosado y diferente.

Por otro lado, una técnica muy común para trabajar este tipo de categorías es 'One hot encoding' La cual pretende generar variables booleanas, que determinen si un registro tiene o no tiene dicha categoría, así que para este caso, la variable link_color se convertiría en nuevas variables, una variable por cada categoría, las cuales tendrán el valor de 0 si no posee dicha categoría o 1 si posee la categoría.

4. Construcción del dataset

¿Qué diferencia hay en usar un conjunto de validación?

Para iniciar con la construcción del dataset, se deben definir 3 conjuntos de datos: datos de entrenamiento, datos de validación y datos de prueba. Los datos de entrenamiento son utilizados para ajustar el modelo, en este caso, los datos de entrenamiento serán usados por la Red Neuronal para ajustar los pesos de cada neurona en una etapa de entrenamiento. Los datos de validación se emplean para conocer el funcionamiento del modelo y validar las respuestas, determinar la taza de aprendizaje y evaluar posibles ajustes en los parámetros, tomando comparaciones con los cambios realizados. Finalmente se trabaja con los datos de prueba una vez terminada la etapa de entrenamiento y validación, con el fin de determinar la efectividad del modelo construido y evaluar su rendimiento.

¿Mejora los resultados en la construcción del modelo usar un conjunto de validación?

Si mejora los resultados usar un conjunto de datos de validación, ya que el conjunto de validación evita el sobre-ajuste del modelo. Al utilizar solo el conjunto de entrenamiento, la red neuronal se ajustará en la mayoría de los casos a los datos de entrenamiento, obteniendo resultados satisfactorios sobre-ajustados a los datos de entrenamiento, lo cual se sugiere utilizar el conjunto de validación aparte del conjunto de entrenamiento para la revisión de los resultados y el ajuste de los parametros que se trabajaron en la etapa de entrenamiento y de esta manera evitar que la red neuronal que de muy ajustada con respecto a su entrenamiento.

¿Qué información provee el uso de un conjunto de validación?

El conjunto de validación brinda datos con los cuales se pretende ajustar los parámetros y construcción del modelo, también ayuda a obtener información acerca de efectividad de la etapa de entrenamiento.

5. Elaboración del modelo

Para la elaboración del modelo, se utilizaran las librerías Perceptron y MLPClassifierde de sklearn, las cuales permitirán trabajar con las arquitecturas perceptrón y de capa oculta. Para su elaboración, es necesario dividir los conjustos de datos con los cuales se realizará el trabajo de las etapas de entrenamiento, validación y prueba.

- Perceptrón: scikit-learn contiene una biblioteca para trabajar con redes neuronales usando la arquitectura Perceptrón, esta librería recibe como parametros el factor de aprendizaje que por dejecto es de 1.0. Sin embargo, para este modelo se utilizará un factor de prendizaje de 0.0001. Adicional configuramosmax_iter=1000000 y validation_fraction=0.2.
- Red neuronal con una capa oculta con un numero de neuronas igual al numero de entradas:

  Para este punto se usará la clase MLPClassifierde de scikit-learn. Haciendo uso de la librería anterior, tambíen se puede configurar el factor de aprendizaje desde los parametros.
- Red neuronal con dos capas ocultas, la primera con las mitad de las entradas y la segunda con la misma cantidad de la capa oculta

6. Análisis de resultados

Se ejecutan las sentencias correspondientes haciendo uso de las librerías de scikit-learn, pasando los 3 modelos por entrenamiento y pruebas. Finalmente se aplica la matriz de confusión, la cual se solicita para este trabajo. La matriz de confusión permite evaluar y visualizar el desempeño del algoritmo. En este caso para los modelos de Pereptrón, Red Neuronal de una Capa y Red Neuronal de dos capas se realiza la generación de la matriz de confusión, dando como resultado en cada columna, la representación de predicciones de cada clase y en las filas, la representación de la clase real.

```
In [310]: #perceptron
          from sklearn.linear_model import Perceptron
          rnp = Perceptron(fit_intercept=True, eta0=0.0001, max_iter=1000000, shuffle=False,early_stopping=True,validation_fra
          rnp.fit(X_train, y_train)

Out[310]: Perceptron
          Perceptron(early_stopping=True, eta0=0.0001, max_iter=1000000, shuffle=False,
                     validation_fraction=0.2)

In [311]: rnp.score(X_train,y_train)
Out[311]: 0.4792050412021328
```

```
#Capa oculta
from sklearn.neural_network import MLPClassifier
```

```
rnco = MLPClassifier(hidden_layer_sizes=(1,19),activation='logistic',learning_rate_init=0.0001,max_iter=1000000, shu
rnco.fit(X_train, y_train)
```

```
MLPClassifier
MLPClassifier(activation='logistic', early_stopping=True,
              hidden_layer_sizes=(1, 19), learning_rate_init=0.0001,
              max_iter=1000000, shuffle=False, validation_fraction=0.2)
```

```
rnco.score(X_train,y_train)
```

```
0.47949587978671837
```

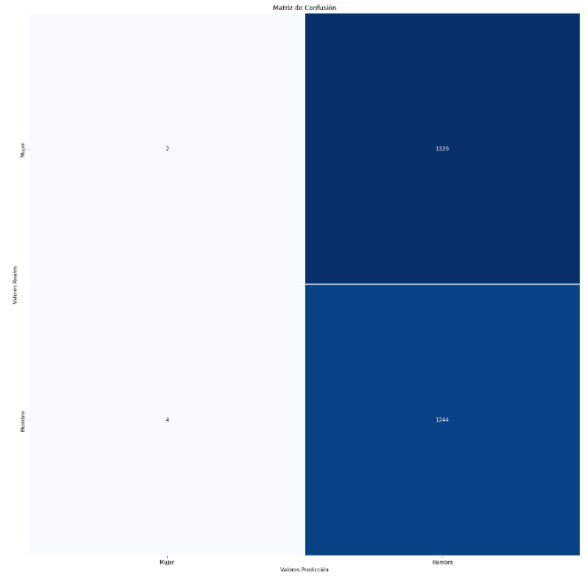
```
# Red Neuronal 2 capas
rn2c = MLPClassifier(hidden_layer_sizes=(2,5),activation='logistic',learning_rate_init=0.0001,max_iter=1000000, shuf
rn2c.fit(X_train, y_train)
```

```
MLPClassifier
MLPClassifier(activation='logistic', early_stopping=True,
              hidden_layer_sizes=(2, 5), learning_rate_init=0.0001,
              max_iter=1000000, shuffle=False, validation_fraction=0.2)
```

```
rn2c.score(X_train,y_train)
```

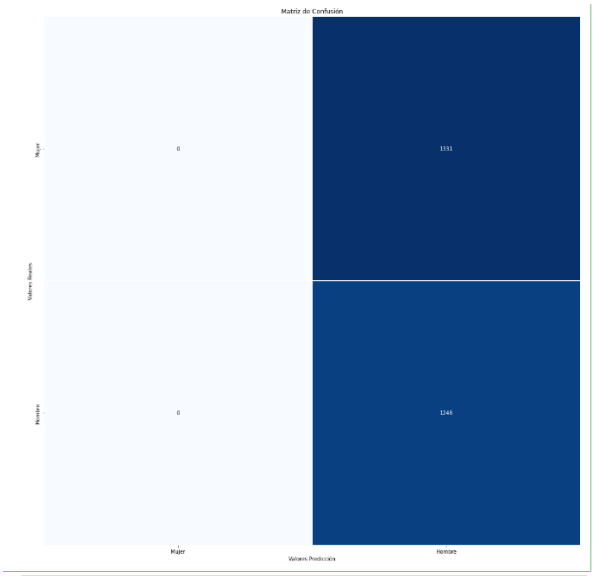
```
0.47949587978671837
```

Sin embargo, los resultados comprobados con la matriz de confusión, no son los esperados como objetivo de la construcción del modelo. Para ello, es necesario ajustar nuevamente las variables seleccionadas, generar un nuevo proceso de limpieza y buscar nuevas características más precisas que generen un algoritmo más eficiente para la predicción de estás dos categorías.



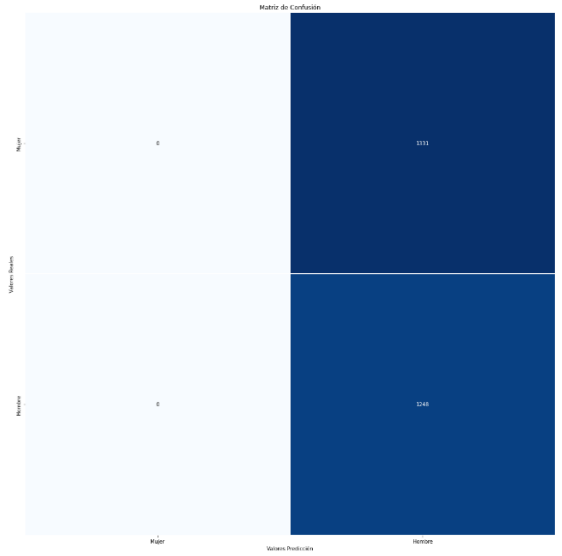

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,rnp.predict(X_test))
```

```
0.483132997857697
```




```
0]: accuracy_score(y_test,rnco.predict(X_test))
```

```
0]: 0.4839084916634354
```




```
[336]: accuracy_score(y_test,rn2c.predict(X_test))
```

```
[336]: 0.4839084916634354
```

7. Ajustes

Al realizar el proceso de construcción del modelo y al ejecutar las etapas de entrenamiento respectivas, se logra evidenciar que estas características seleccionadas pueden no ser suficientes para determinar la predicción, tal vez es necesario evaluar otras variables que brinden más información. El color link_color y el sidebar_color pueden no ser características fuertes que determinen la predicción esperada.