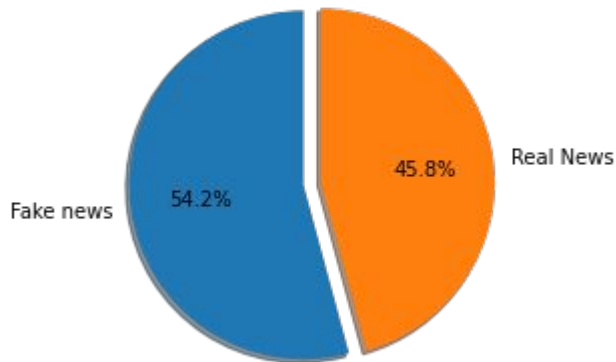


Fake news

grupo 3

DataSet

Los datos para este proyecto vienen separados en dos archivos .csv. El primero contiene 23481 noticias categorizadas como **falsas** y el segundo 21417 noticias categorizadas como reales. Se puede ver que la la distribución de los datos está balanceada ya que tenemos 54.2% falsas y 45.8% verdaderas.



Features

En los dos archivos tenemos las mismas features:

- Título
- Texto
- Categoría
- Fecha

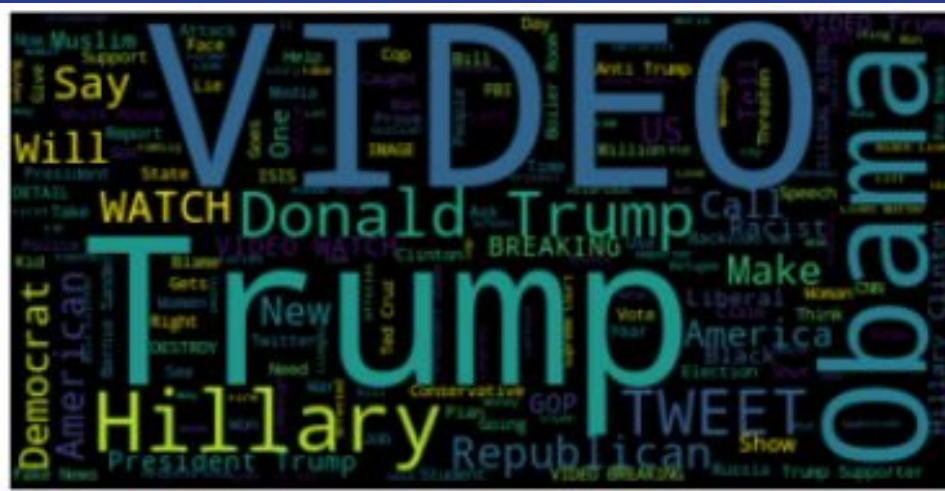
Las cuales cuentan con estas características:

	title	text	subject	date	Fake News	News
count	39105	39105	39105	39105	39105	39105
unique	38729	38646	7	2397	2	39103
top	Factbox: Trump fills top jobs for his administ...		politicsNews	December 6, 2017	False	Pence says NAFTA renegotiation will be a 'win ...
freq	14	445	11217	166	21197	2

Análisis de los datos

Lo primero que hicimos fue crear una nube de palabras para ver cuales son las más relevantes en cada conjunto de datos puesto que esta será la feature más relevante. Podemos ver que en las noticias fake predomina la palabra “video” mientras que en las reales no hay una que destaque sacando a “trump” que aparece en los dos datasets.

FAKE



TRUE



-En la nube de palabras de noticias verdaderas aparecen palabras que indican condicional.

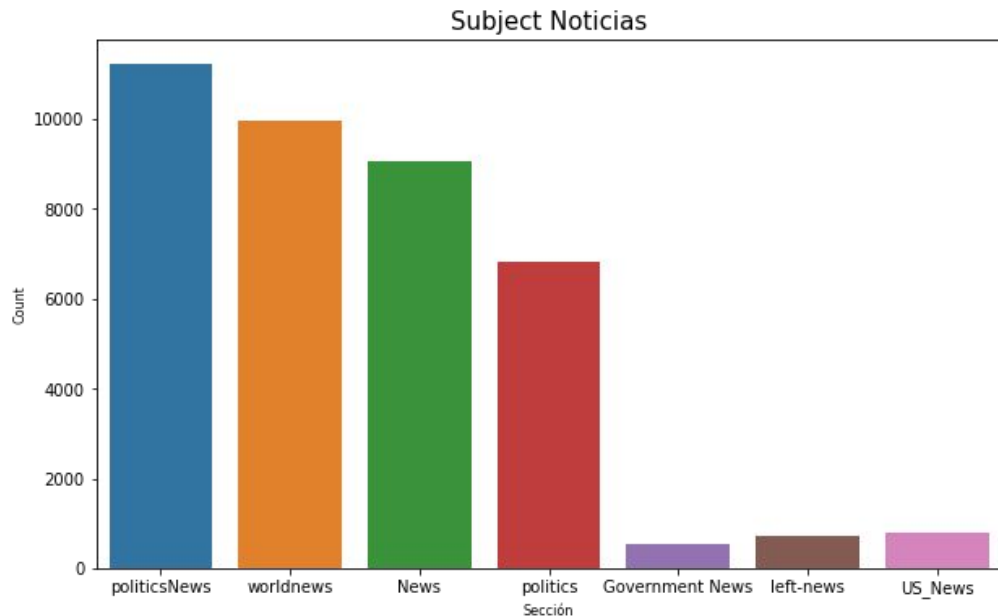
Podría interpretarse como que en noticias falsas no es relevante la rigurosidad de que algo esté confirmado o no.

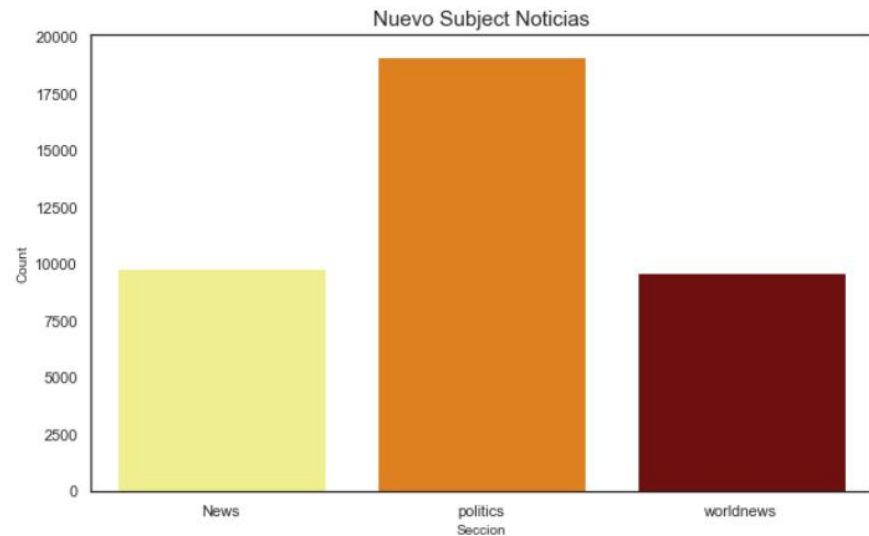
-Aparecen de forma más notoria palabras como WATCH y VIDEO en la nube de fake news

-En las noticias verdaderas hay palabras como OFFICIAL o SOURCE y en las fake, no

¿Que tipo de noticias tenemos?

Al hacer un análisis de las categorías, notamos que había incongruencia en sus nombres. Se puede ver qué “politics” aparece dos veces con distinto nombre y otros tipos de noticias no son significativos para los datos.



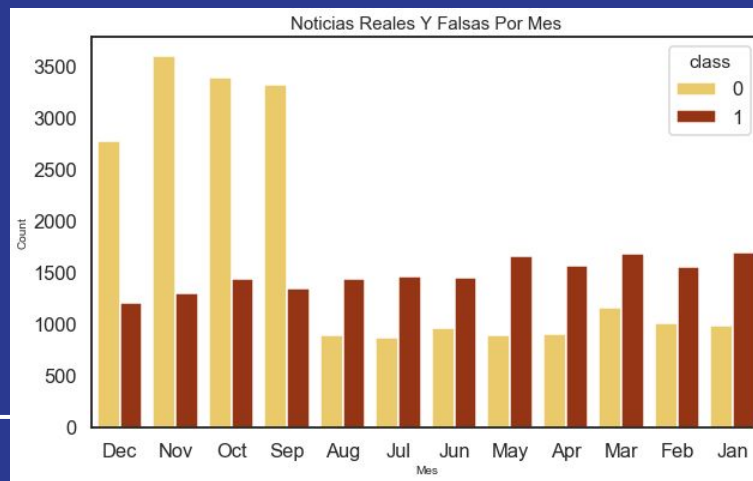
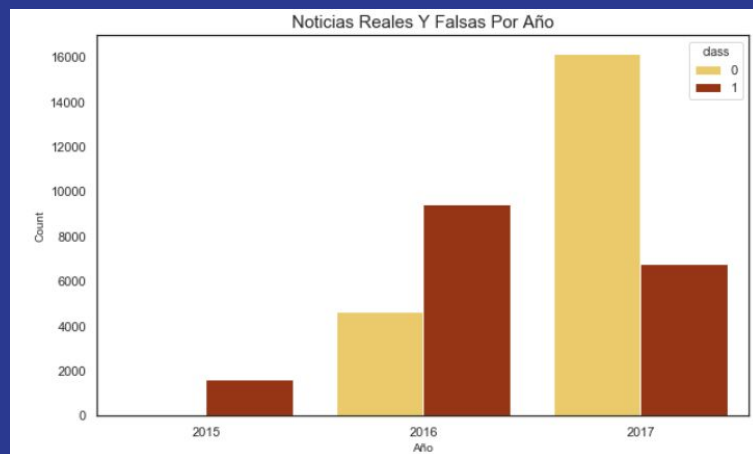


Solución

Lo que hicimos fue unificar las distintas categorías en 3 principales que engloban al resto.

Fechas

Al analizar las fechas vimos que hay una distinción entre las noticias falsas y las reales, por lo tanto es un dato relevante a la hacer un análisis



Fechas

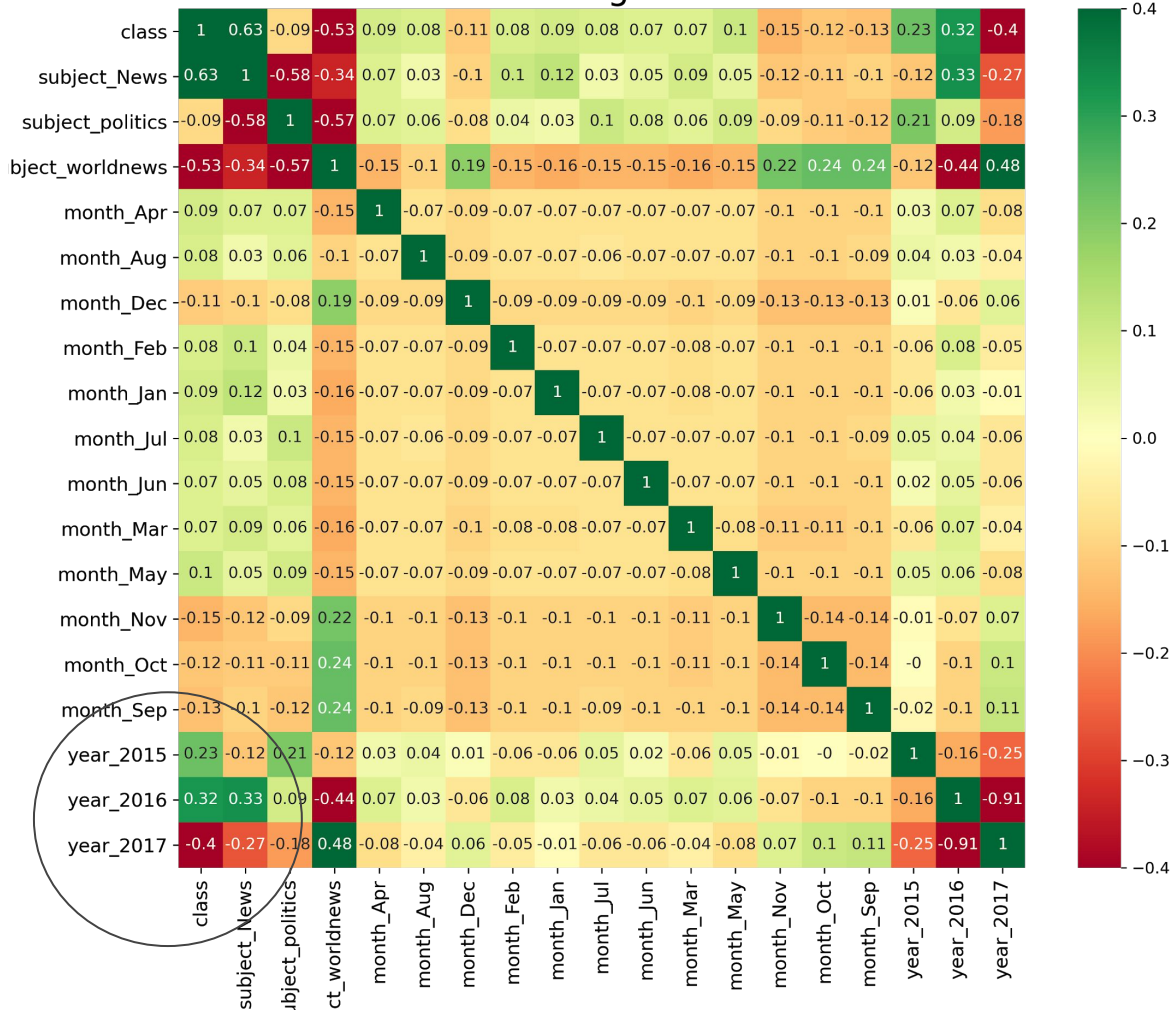
Lo que hicimos para poder tomar esta feature en un formato que sea legible por una red neuronal fue separa el mes y el año de cada noticia en dos columnas y luego generar dummy variables del mes para tenerlo en formato 1 y 0

month_Apr	month_Aug	month_Dec	month_Feb	month_Jan	month_Jul
-----------	-----------	-----------	-----------	-----------	-----------

0	0	1	0	0	0
---	---	---	---	---	---

0	0	1	0	0	0
---	---	---	---	---	---

Correlograma



2017 es el año con más noticias pero no el año con más fake news

2016 es el año con más fake news. Tiene sentido porque fue el año de las elecciones

Mayo, Marzo y Enero son los meses con más fake news. Las elecciones son en Noviembre

Texto

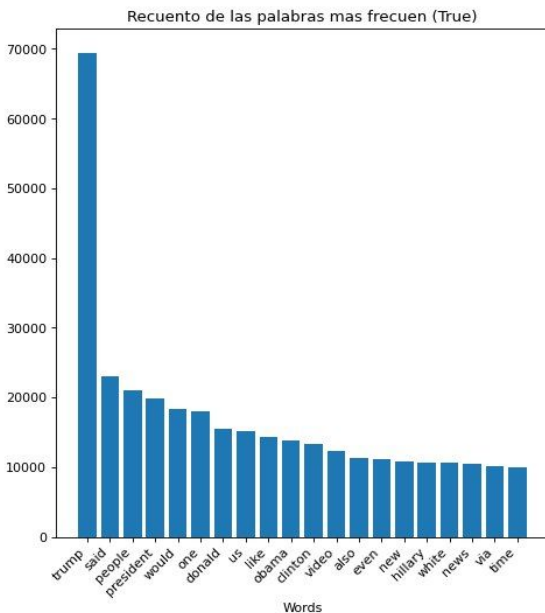
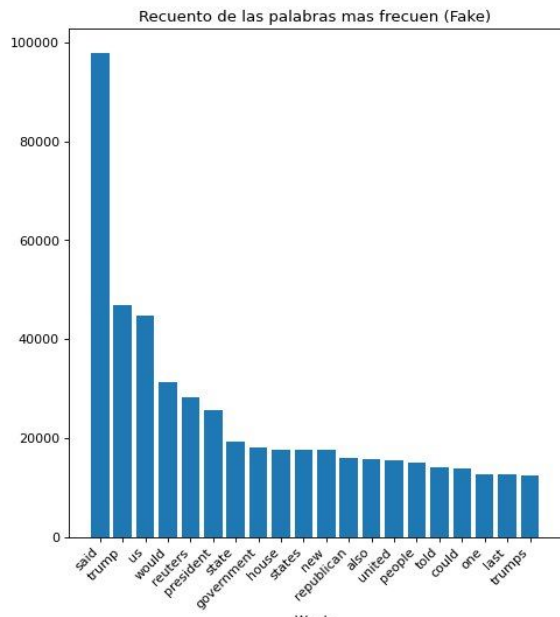
Para poder pasarle el texto a algún algoritmo de IA en un formato con el que pueda trabajar creamos una bolsa de palabras. Esto lo hicimos por distintos métodos, primero creamos una función propia para crearla para tener más control sobre los datos, la función Tokenizer de Keras y una función de nltk que nos dio mejor rendimiento

Esta función nos devuelve una matriz de 38688 filas por 15000 columnas donde cada fila contiene un 1 o un 0 dependiendo de si esa noticia tiene la palabra a la que correspondía esa columna.



Texto

Al utilizar una bag of words, contamos la frecuencia con la que salían las distintas palabras en ambos tipos de noticia:

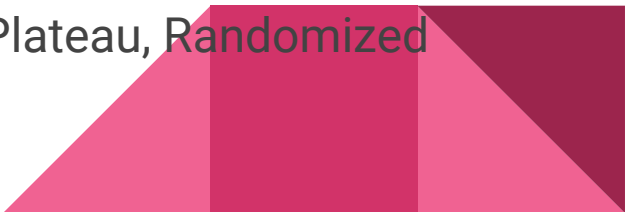


Removimos las stopwords
para sacar basura de la
bolsa.

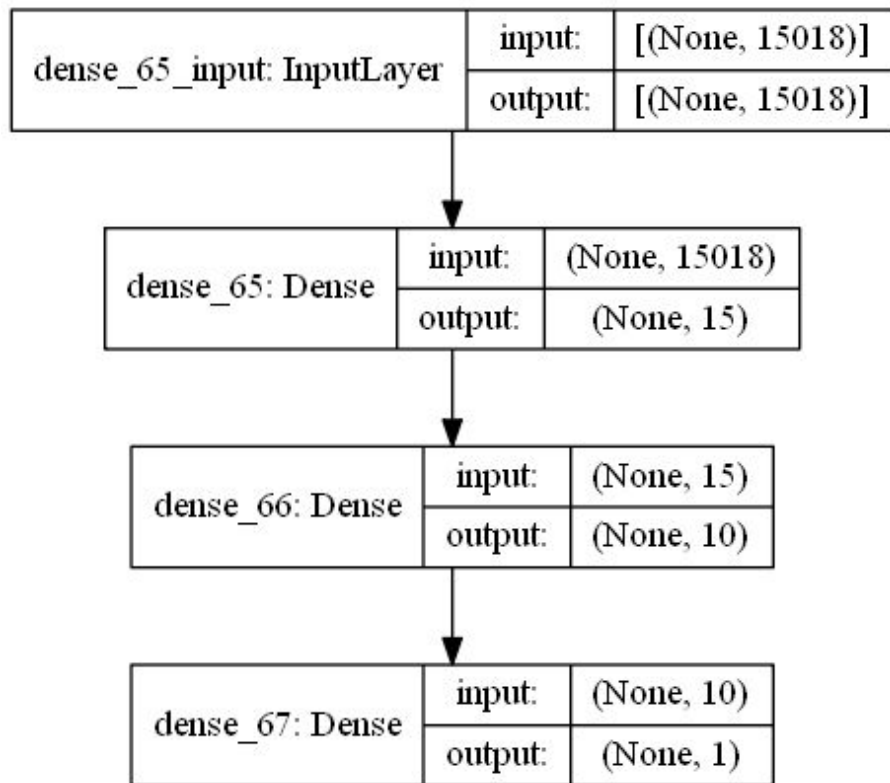


La Red

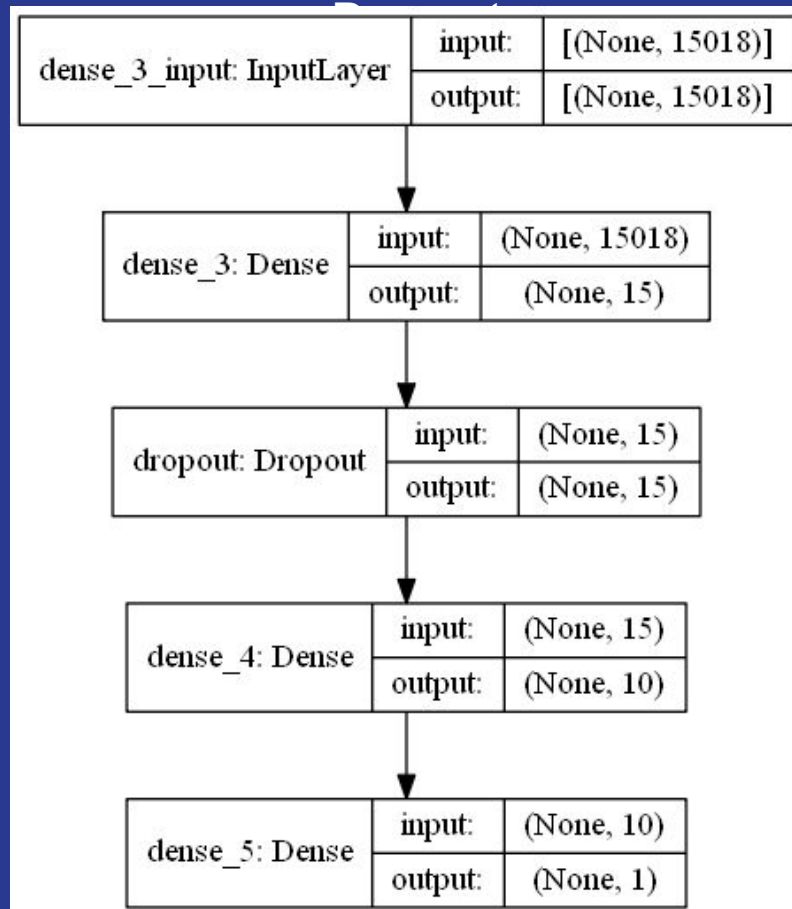
Cosas que fuimos probando:

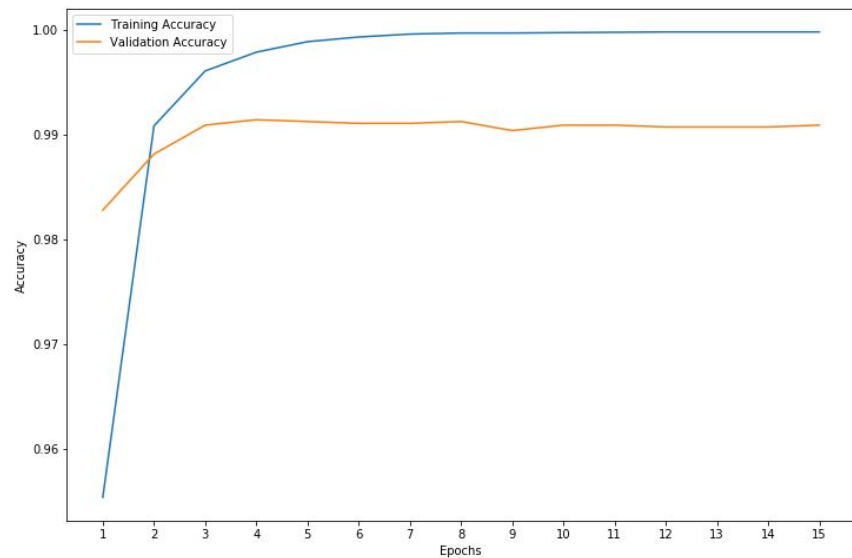
- Modelo de Random Forest con 200 árboles y Gridsearch
 - Pasarle a la Red Neuronal sólo las palabras vectorizadas para que clasifique
 - Pasarle las palabras vectorizadas + el resto de las variables (meses, subject) y comparar si había diferencia
 - Red neuronal con regularización L1, L2, dropout rate (sin regularizar overfitting)
 - La misma red neuronal con 1 y 2 capas. Nos quedamos con la de una capa
 - Sumamos Keras Classifier, Early Stopping, ReduceLROnPlateau, Randomized Search y cross validamos con 3 folds
- 

Red Neuronal sin callbacks ni regularización

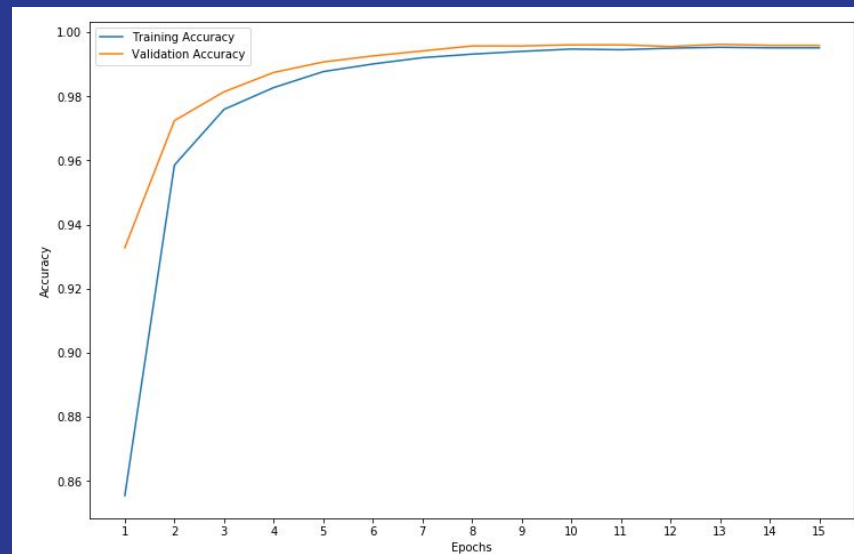


Red Neuronal con regularización y





Accuracy: 99%
Loss: 0.044



Accuracy: 99.6%
Loss: 0.2014

Modelo 3

Por último probamos un modelo usando:

- Keras Classifier
- Early Stopping
- ReduceLROnPlateau
- Cross validation
- Dropout

[85]:

	Model	Train Acc	Test Acc	Train Acc - Test Acc
1	Modelo 1	99.969	99.1212	0.847763
2	Modelo 2	99.5726	99.6123	-0.039655
3	Modelo 3	0.995726	0.996898	-0.00117193

Y obtuvimos este resultado.