Utilizando «Data Science» para encontrar nuestro restaurante ideal

[iStock]/Thinkstock

Cada vez que tenemos intención de ir a un nuevo restaurante nos surge la duda de si será el indicado y cubrirá nuestras expectativas como comensales. Como parte de mi trabajo de Fin de Máster del Máster en «Data Science» y «Big Data» de Afi Escuela, buscaré dar respuesta a dicha problemática mediante el uso de distintas técnicas de «Data Science».

Matías Nicolás Caputti @yosoymatias | *Graduado Máster Data Science y Big Data de Afi Escuela de Finanzas*

La empresa estadounidense Yelp, fundada por Jeremy Stoppelman y Russel Simmons, es una plataforma que actúa como red social y permite a los usuarios subir y compartir fotos y opiniones de los restaurantes y establecimientos que visitan. Actualmente Yelp aloja millones de fotos de restaurantes cargadas por sus usuarios de todo el mundo.

Con el fin de poder procesar dichas fotos de manera automática, Yelp presentó en la plataforma Kaggle un desafío titulado «*Clasificación de fotos de restaurantes de Yelp*»[1]. Dicho desafío consiste en construir un modelo que asocie automáticamente restaurantes con múltiples etiquetas, usando un conjunto de datos de fotos subidas por los usuarios a su plataforma.

Análisis de los datos

Para el desarrollo del trabajo se utilizó el set de datos de dicha competencia, dividido en particiones de *train* y *test*. La partición de *train*, con 230.000 imágenes pertenecientes a 2.000 restaurantes, fue utilizada para entrenar los distintos modelos que iré utilizando. La partición de *test*, con 240.000 imágenes pertenecientes a 10.000 restaurantes, fue utilizada para evaluar el desempeño de los modelos utilizados.

Las imágenes del set de datos tienen resolución media de 375*500 píxeles. Para unificar tamaños, fueron ajustadas a 96*96 píxeles. Además, para lograr una correcta representación digital de las características de cada imagen, se ajustaron sus niveles de iluminación, contraste y desenfoque.

El objetivo del trabajo, al igual que el del desafío, fue etiquetar a cada restaurante con una o más de las siguientes etiquetas:

1. *Good for lunch*: Bueno para almorzar
2. *Good for dinner*: Bueno para cenar

3. *Takes reservations*: Toma reservas
4. *Outdoor seating*: Mesas exteriores
5. *Restaurant is expensive*: Restaurante caro
6. *Has alcohol*: Tiene alcohol
7. *Has table service*: Tiene servicio de mesa
8. *Ambiance is classy*: Ambiente elegante
9. *Good for kids*: Bueno para niños

En la siguiente figura se pueden observar algunas de las imágenes del set de datos con sus respectivas etiquetas.

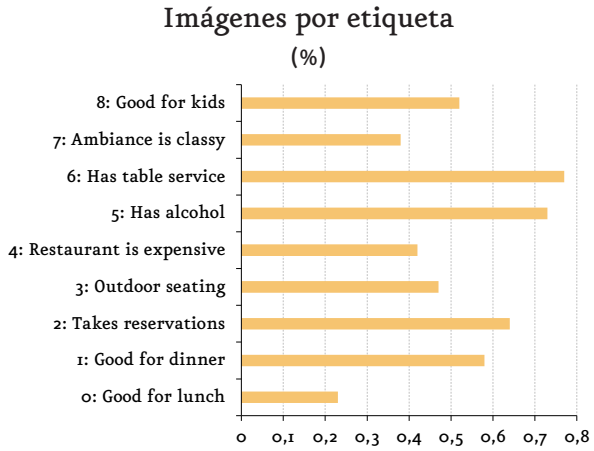
IMÁGENES DEL SET DE DATOS CON SUS ETIQUETAS CORRESPONDIENTES

Fuente: *elaboración propia.*

Dentro de las 9 posibles etiquetas para cada restaurante, y tal como se puede observar en la siguiente figura, se detecta que no todas están asignadas uniformemente en la misma cantidad de imágenes. Las etiquetas que más se repiten, con ocurrencia en más del 75% de las imágenes, son «*Has table service*» y «*Has alcohol*». La que menos se repite, con ocurrencia del 24% es «*Good for lunch*». Sin embargo, mantendré las clases desequilibradas ya que las clases mayoritarias son las que más importan a las personas a la hora de ir a un restaurante.

Modelización

Una vez entendido el problema y los datos que tenemos para resolverlo, lo primero que hice fue comparar distintas técnicas y modelos que me permitieran asignar etiquetas a cada imagen.



Fuente: *elaboración propia.*

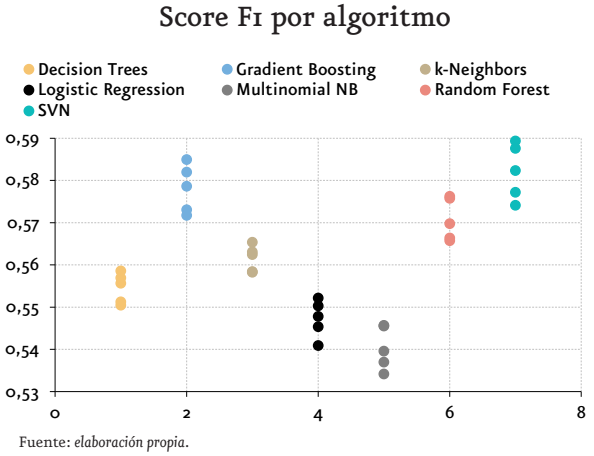
Todos los modelos que probé fueron evaluados mediante la métrica *F1* o *F1 score*[2], la misma que fue utilizada en el desafío publicado en Kaggle para seleccionar al equipo ganador.

$$F1 = \frac{2pr}{p + r}$$

La métrica *F1* brinda un equilibrio en el rendimiento tanto para la «*precision*» (*p*) como para el «*recall*» (*r*), e intenta optimizarlos conjuntamente.

Como a priori es difícil saber qué algoritmo será el que mejor se acople al problema, utilicé una técnica llamada *spot checking*[3] mediante la cual evalué distintos algoritmos sin ajustar sus parámetros, para dar rápidamente con el mejor de ellos.

Los algoritmos que probé mediante esta técnica fueron: *Decision Trees*, *Gradient Boosting*, *k-Neighbors*, *Logistic Regression*, *Multinomial Naive Bayes*, *Random Forest* y *Support Vector Machines* (*SVM*). En la siguiente figura se observa para cada algoritmo un *boxplot* con los 5 valores de *F1* que fueron obtenidos luego de aplicar validación cruzada.



Fuente: *elaboración propia.*

Los mejores resultados son obtenidos por los algoritmos *SVM* con F1 entre 0.57 y 0.59 y, apenas por deba-

jo, *Gradient Boosting* con *F1* entre 0.57 y 0.585. Sin embargo, ningún algoritmo logró superar la barrera de valores *F1* superiores a 0.60, por lo que abordaré el problema desde la perspectiva del aprendizaje profundo o *deep learning* en busca de mejores resultados.

Un modelo de *deep learning* es diseñado para analizar continuamente los datos con una estructura lógica estratificada de algoritmos similar a la que utiliza un ser humano para sacar conclusiones, llamada *red neuronal*. El diseño de una red neuronal artificial está inspirado en la red neuronal biológica del cerebro humano, lo que hace que en problemas de alta complejidad la inteligencia de la máquina sea mucho más capaz que la de los modelos de aprendizaje automático estándar.

Para mi problema de clasificación de restaurantes, principalmente por ser imágenes mi objeto de estudio, utilicé redes neuronales con capas convolucionales que permiten extraer mayor cantidad de características (*features*) de cada imagen, para luego clasificar dichas características en las capas densas superiores de cada red.

Las arquitecturas de red que utilicé fueron las siguientes:

• Red convolucional base: basada en *LeNet-5*, introducida por Yann LeCun[4], que presenta dos grupos de capas convolucionales, seguidas de capas de *pooling*, una capa densa y, finalmente, un clasificador.

• Redes pre-entrenadas y transferencia de aprendizaje: arquitecturas de red más complejas con el fin de mejorar el rendimiento y precisión de la red. Hice pruebas sobre arquitecturas *VGGNet*[5] e *InceptionV3*[6]. Utilicé dichas redes pre-entrenadas sobre los sets de datos *ImageNet*[7] y *Places365*[8]. Por último, realicé un entrenamiento selectivo de sólo algunos bloques de sus capas superiores, para agilizar los tiempos de entrenamiento.

• Redes *Ensemble*: hice pruebas de agrupación sobre las arquitecturas anteriores en busca de una red más robusta. Utilicé dos formas para ensamblar las redes:

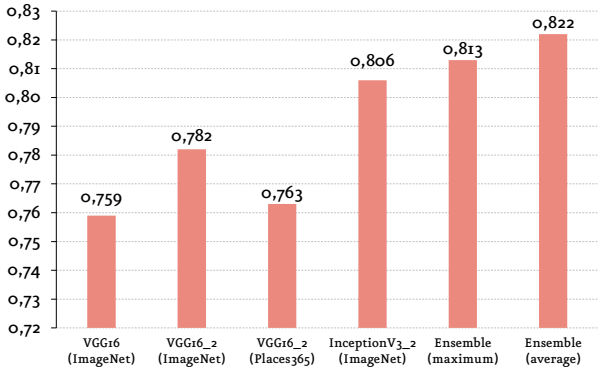
- *Maximum*: evaluando predicciones de cada modelo y tomando las predicciones máximas en cada caso.

- *Average*: evaluando las predicciones de cada modelo y tomando el promedio de estas en cada caso.

Cada arquitectura fue entrenada y evaluada de forma remota en una instancia de Amazon Web Services Elastic Compute Cloud (*AWS EC2*)[9], haciendo uso de una GPU Tesla K80.

En la siguiente figura se pueden ver los rendimientos de cada arquitectura de red evaluada, al igual que con los algoritmos anteriores antes, con el score *F1*.

Score F1 para cada arquitectura de red



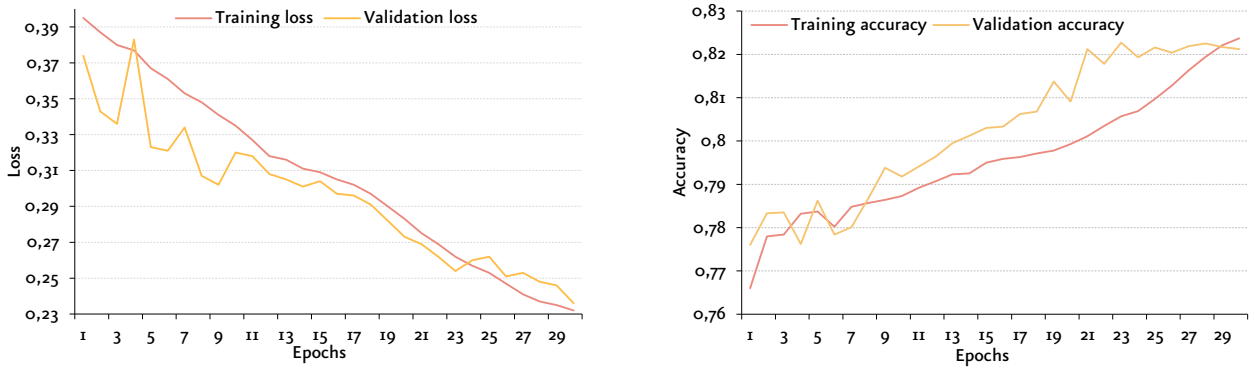
Fuente: *elaboración propia*.

El modelo *Ensemble (average)* fue el modelo que mejor respuesta consiguió frente al problema de etiquetado múltiple, ya que es el que mejor valor *F1* obtuvo y el que posiblemente mejor se comporte cuando se lo ponga a prueba en producción para etiquetar nuevas imágenes. Todas las arquitecturas de red superaron ampliamente a los algoritmos probados anteriormente en el *spot checking*.

En la siguiente figura se observan las curvas de evolución de la pérdida (*loss*) y valor de *F1* a lo largo del entrenamiento del modelo *Ensemble (average)*. El mismo finalizó su entrenamiento en la época 30, y obtuvo su mayor rendimiento en la época 23.

Curvas de evolución de *loss* y *accuracy* del modelo *Ensemble (average)*

(%)



Fuente: *elaboración propia*.

Resultados

Los resultados obtenidos sirvieron para cumplir con el principal objetivo del trabajo. Es decir, se logró crear un modelo que asigne múltiples etiquetas a imágenes de restaurantes, lo que conllevó experimentar gran variedad de algoritmos y redes neuronales, y permitió arribar a una solución con resultados más que adecuados para el problema, logrando sólo un 3% menos de rendimiento que la solución ganadora, lo que significa que este trabajo se hubiera posicionado en un hipotético 350 lugar de la competencia de Kaggle en la cual se presentaron más 350 equipos.

Se presentó como modelo final el *Ensemble (average)* de redes neuronales. Dicho ensemble logró el mejor valor de F1: 0.822 con las imágenes de *train* y 0.803 con las imágenes de *test*, superando ampliamente a los algoritmos del *spot checking* y a las demás arquitecturas de red. Sin embargo, no hay modelos perfectos y, al ser puesto en producción, es probable que en ciertos casos falle.

El desarrollo completo del trabajo y los distintos bloques de código utilizados en el mismo pueden ser accedidos en la memoria del TFM[10], donde también hay otras pruebas realizadas sobre el set de datos, tales como reducción de dimensionalidad y pruebas de etiquetado mediante clustering y aprendizaje no supervisado ::

Nota: este artículo es un extracto del trabajo de fin de curso del Máster en *Data Science* y *Big Data* 2017-2018, Afi Escuela de Finanzas.

[1] «*Yelp Restaurant Photo Classification*», publicado en Kaggle. Consultar aquí.

[2] «*Métrica de evaluación F1 score*». Consultar aquí.

[3] «*How to Develop a Reusable Framework to Spot-Check Algorithms*». Consultar aquí.

[4] «*LeNet-5, convolutional networks*». Consultar aquí.

[5] «*Very Deep Convolutional Networks for Large-scale Image Recognition*» por Karen Simonyan y Andrew Zisserman. Consultar aquí.

[6] «*Rethinking the Inception Architecture for Computer Vision*» por Christian Szegedy, Vincent Vanhoucke y Sergey Ioffe. Consultar aquí.

[7] «*ImageNet dataset*». Consultar aquí.

[8] «*Places dataset*». Consultar aquí.

[9] «*Amazon Elastic Compute Cloud (Amazon EC2)*». Consultar aquí.

[10] Memoria TFM «*Análisis clasificatorio de imágenes de restaurantes*» por Matías Caputti, Junio 2018. Consultar aquí.