

---

## **#influASYNCer**

---

31 de marzo de 2019

Marco Pérez Muñoz  
Miguel González Sosa  
María del Rocío Cabello Toscano

# Índice general

0.1. Introducción . . . . .	2
0.2. Tecnologías utilizadas . . . . .	2
0.3. Descripción general del modelo generado . . . . .	2
0.4. Descripción técnica del funcionamiento del modelo . . . . .	2
0.4.1. Obtención de datos a través de la API . . . . .	2
0.4.2. Entrenamiento de la Red Neuronal Convolucional . . . . .	3
0.5. Detalles de implementación . . . . .	3
0.6. Planificación a futuro . . . . .	3

## **0.1. INTRODUCCIÓN**

El mundo de las redes sociales cada vez tiene más influencia en nuestro día a día. No es de extrañar que, debido a su uso masivo y continuado, las redes sociales generen una enorme cantidad de datos (fenómeno conocido como *Big Data*).

Esto hace bastante atractiva la idea de cuantificar de alguna forma qué está sucediendo en estas plataformas y en consecuencia, conocer la influencia que tienen ciertos usuarios (*influencers*) sobre otros, y en consecuencia, la influencia de éstos a la sociedad en general.

El objetivo de nuestra aplicación, *#influASYNCer*, no es otro que el de tratar de usar una gran cantidad de datos para generar información útil. La aplicación se compone de dos soluciones principales:

- Conocer quienes son los usuarios y temas con mayor influencia, como puede ser la vida sana, el deporte, la tecnología, etc. Esto se ha solucionado para una red social en concreto: Instagram.

- Hacer posible la clasificación automática de imágenes que contienen un coche Porsche o no.

## **0.2. TECNOLOGÍAS UTILIZADAS**

Se trata de una aplicación web y podría estructurarse de la siguiente forma:

- En el front-end se ha hecho uso de las tecnologías HTML, JavaScript y CSS, además de haber utilizado principalmente para el diseño del mismo la librería de CSS llamada Bootstrap.

- En el back-end se ha utilizado el lenguaje de programación Python y el framework Flask para generar el servidor web. Cabe destacar que se ha utilizado la librería Keras para la generación de un modelo predictivo para clasificar imágenes de coches según si son de la marca Porsche o no.

## **0.3. DESCRIPCIÓN GENERAL DEL MODELO GENERADO**

## **0.4. DESCRIPCIÓN TÉCNICA DEL FUNCIONAMIENTO DEL MODELO**

### **0.4.1. Obtención de datos a través de la API**

Se han realizado las peticiones a la API de manera asíncrona para hacerlo más veloz. Además, se ha tenido en cuenta que muchas de las peticiones se realizan más de una vez para obtener los mismos datos. Esto se ha controlado para realizar cada una de estas peticiones una sola vez.

### **0.4.2. Entrenamiento de la Red Neuronal Convolutiva**

Para la clasificación de imágenes, se ha entrenado una Red Neuronal Convolutiva con Keras, utilizando la técnica data augmentation con el fin de contar con un mayor número de muestras para el entrenamiento.

## **0.5. DETALLES DE IMPLEMENTACIÓN**

En el backend se espera que se configure un cron para ejecutar la actualización de la base de datos completa (una colección de MongoDB con todos los posts y toda la metadata relacionada con cada uno de ellos) cada x intervalo de tiempo con la nueva información que devuelve la api. Se utilizaría el script `instagram_api_retriever.py`.

Evidentemente, por el limitado tiempo que tenemos, el código no es muy limpio y tiene bastantes variables hardcodeadas. La idea era montar la base de datos, el *populate* de la colección con un json descargado anteriormente (por mostrar algo al principio) y el servidor web con flask y el cron configurado.

## **0.6. PLANIFICACIÓN A FUTURO**

Esta aplicación web es bastante flexible a la hora de realizarle cambios para ampliar su utilidad:

- Si se añaden temas o hashtags al fichero 'hashtag.json' que se encuentra en la raíz del proyecto, automáticamente se actualizará la aplicación para estudiar estos nuevos campos y mostrar sus respectivos resultados.
- Siguiendo la misma estructura, se podría adaptar la aplicación sin muchos cambios para que funcionase sobre Twitter.