Aplicación de técnicas de Machine Learning para Data Science And Engineering Club

Cosmina Ioana Nicu

Resumen – Actualmente en las grandes empresas se generan a diario una gran cantidad de datos que provoca interés en especialistas en el campo de Machine Learning. En este artículo se ha desarrollado la aplicación de técnicas de Aprendizaje Computacional en tres conjuntos de datos para su posterior publicación en el Blog Data UAB. El proceso que se ha realizado para la aplicación de dichas técnicas, ha consistido en la exploración y el análisis de los datos como paso previo a la creación de los modelos. Además se ha llevado a cabo la participación en una competición de Kaggle y unas mejoras al Blog.

Palabras clave– Machine Learning, Analista de datos, Clasificación, Sistemas de recomendación

Abstract– Currently, in big companies, a large amount of data is generated daily, causing interest in specialists in the field of Machine Learning. In this article, the application of Machine Learning techniques have been developed in three data sets for subsequent publication in the Blog Data UAB. The process that has been carried out for the application of these techniques has consisted of the exploration and analysis of the data as a previous step to the creation of the models. Besides, it includes the participation in a Kaggle competition, and some improvements have been made to the Blog.

Keywords– Machine learning, Data Analysis, Classification, Recommender's system.

1 Introducción

Cada vez estamos más acostumbrados a tener presente en muchos aspectos de nuestra vida cotidiana máquinas capaces de aprender por sí solas como Alexa. Por ello, la mayoría de las empresas de hoy en día se interesan en especialistas en Machine Learning y Deep Learning, ramas de la Inteligencia Artificial. Estas están destinadas a la creación de programas que tengan la capacidad de generar comportamientos automáticos.

Aunque haya una gran demanda, no es fácil encontrar especialistas en dichos campos y es por ello que la Universidad Autónoma de Barcelona creó una plataforma web en formato Blog llamado Data Science and Engineering Club [1] dedicado especialmente a los estudiantes de Ingeniería Informática especializados en el campo de Machine Learning e Inteligencia Artificial.

El objetivo principal del Blog es que los estudiantes puedan presentar y proponer soluciones a problemas reales aplicando diferentes técnicas y herramientas de Machine Learning o Deep Learning. Además de ser una herramienta de aprendizaje para los estudiantes, el Blog pretende ser un intermediario entre los miembros del club y las empresas interesadas en especialistas en los campos mencionados, teniendo así la oportunidad de que conozcan a los miembros del club a partir de su trabajo.

Tal y como se ha comentado, el objetivo principal de este trabajo ha sido aplicar diferentes técnicas de Machine Learning y Deep Learning a tres problemas reales por lo que a continuación se detallará cada uno de ellos. Primero se explicará el procedimiento seguido en el análisis de los datos, seguido de las técnicas escogidas a implementar y por último los resultados obtenidos. Finalmente, podemos encontrar la explicación de la competición realizada y las mejoras llevadas a cabo en el Blog DataUAB.

2 Selección de conjuntos de datos

Como primer paso en el proyecto se ha realizado un proceso de selección y análisis de los posibles conjuntos de datos disponibles. Para seleccionar dichos conjuntos se ha tenido en cuenta factores como las técnicas de Machine Learning que se podrán utilizar y que trate temas de interés.

• E-mail de contacto: cosminanicu@gmail.com
• Mención realizada: Computación
• Trabajo tutorizado por: Jordi González Sabaté & Pau Rodríguez (Ciencias de la computación).
• Curso 2019/20

Tras este proceso, se han escogido los tres siguientes conjuntos de datos:

1. **Women's E-Commerce Clothing Reviews** [2]: Con este primer conjunto de datos se ha creado un modelo de predicción capaz de predecir el sentimiento que transmite una reseña. El sentimiento se dividirá en positivo, neutral o negativo, por lo que se tratará de una clasificación multiclase. Este conjunto de datos contiene más de 23.000 reseñas y 10 características sobre ellas extraídas de un comercio anónimo electrónico de ropa femenina.

2. **GoodBooks-10k** [3]: Conteniendo más de diez mil libros diferentes e información sobre más de 50.000 usuarios extraídos de la página web de GoodReads [4], se ha utilizado dicho *dataset* para crear dos de los sistemas de recomendación más conocidos para recomendar el próximo libro a leer, a un usuario: Content-Based y Collaboratve-Filtering.

3. **Intel Image Classification** [5]: Con más de 25.000 imágenes de escenas naturales de al rededor del mundo, se ha buscado identificar si una imagen contiene un paisaje de montaña, mar, o edificios utilizando técnicas de Deep Learning. Con ello, se ha aprovechado también, para introducir el uso de TensorFlow [6].

Se ha desarrollado diferentes modelos de predicción de sentimientos para escoger el mejor, para el primer conjunto de datos. Por lo que respecta al segundo conjunto de datos, se ha desarrollado dos de los modelos más populares de sistemas de recomendación y servirá también como introducción a problemas del mismo estilo. En cuanto al último, además de haber desarrollado un modelo de clasificación, servirá como una introducción a cómo usar TensorFlow desde cero.

3 Publicaciones en Blog

Para cada conjunto de datos mencionados en el apartado anterior, se ha añadido una nueva publicación en el Blog DataUAB con el propósito de añadir más contenido a la página y para difundir también el uso del Blog a más estudiantes que quieran aprender sobre técnicas de Machine Learning.

Por ello, las publicaciones se han escrito mediante la plataforma de Jupyter Notebook [7] en la que el formato del artículo consiste en bloques de código en el lenguaje de programación Python 3.6, y bloques de descripción teórica, manteniendo una estructura similar en todos.

Esto ayudará al lector a seguir de una manera didáctica y fácil de entender qué se está haciendo en todo momento.

A continuación, se detallará para cada uno el análisis realizado, las técnicas aplicadas y los resultados obtenidos.

3.1 Women's E-Commerce Clothing Reviews

En esta primera publicación del Blog, se ha utilizado el conjunto de datos de **Women's E-Commerce Clothing Reviews** [2] con el objetivo de crear un modelo de predicción capaz de predecir el sentimiento que transmite una reseña de un cliente en un comercio electrónico de ropa femenina. Al ser un comercio real, todas las referencias que se hagan hacia el comercio se han reemplazado por la palabra "reatiler"para anonimizar el comercio.

Antes de crear el modelo de predicción, se han analizado todos los datos y por último se han evaluado los resultados obtenidos.

3.1.1 Análisis de datos

El primer paso de este post ha sido explorar los datos y limpiarlos en caso de que fuese necesario, para poder entender nuestro problema y aprovecharlos al máximo. En este conjunto nos encontramos con más de 23000 reseñas y 10 características diferentes sobre ellas como la edad de la persona que ha escrito dicha reseña, la puntuación dada, la categoría del producto, etc.

Primero, se ha comprobado la existencia de valores nulos y se ha decidido qué hacer respecto a ellos ya que es importante decidir si se eliminan, o se rellenan con valores nuevos. En este caso, se han encontrado valores nulos en algunas reseñas y en otros atributos por lo que se han eliminado, dejando así el conjunto de datos limpio.

Una de las tareas mas importantes que se debe llevar a cabo en problemas de Machine Learning es el análisis de los datos, para comprender qué atributos serán los que nos aporten más información a la hora de resolver nuestro problema. Por este motivo, se ha realizado diferente análisis de la distribución de los atributos numéricos tales como la edad del usuario o la relación del sentimiento de la reseña según la edad (Figura 1).

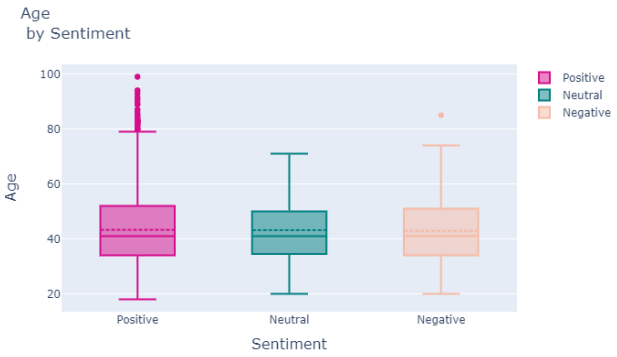

Fig. 1: Relación entre la edad del usuario y el sentimiento.

Por otro lado también se ha analizado la distribución de los atributos categóricos tales como el departamento, la clase del producto o las tallas. Con esto hemos conocido mejor nuestro conjunto de datos, sabiendo que hay 6 categorías diferente y que las más comunes son "Tops"y "Vestidos".
El siguiente paso realizado ha sido analizar y limpiar el texto de la reseña en sí ya que los signos de puntuación y caracteres especiales no aportan información al modelo de Machine Learning. Para limpiar el texto se ha convertido todo a minúsculas, se ha borrado los signos de puntuación y números, y se han utilizado las listas *Stop Words* [8]. Dichas listas contienen las palabras más comunes de una lengua que no aportan información (p.ej: "a", "que", "y") y que por lo tanto, no son necesarias para el modelo.

Esto se ha llevado a cabo con técnicas de NLTK (Natural Language Toolkit) [9], un conjunto de librerías para el procesamiento simbólico y estadístico del procesamiento del lenguaje natural.

Una vez se ha limpiado el texto y preparado para poder pasarlo al modelo, se ha analizado el sentimiento de cada una de las reseñas utilizando el analizador de VADER (Valence Aware Dictionary and Sentiment Reasoner), una herramienta de análisis de sentimientos de NLTK, basada en reglas léxicas que divide el texto en Positivo, Neutral o Negativo.

Un ejemplo de las palabras positivas más usadas las podemos observar en la Figura 2.

Fig. 2: Nube de las palabras positivas más comunes.

3.1.2 Modelo de Machine Learning

La siguiente sección del post está destinada a enseñar diferentes maneras de diseñar un modelo de clasificación y cómo evaluar los resultados.

Sin embargo, la sección comienza explicando cómo preparar los datos ya que se ha encontrado el problema de datos no balanceados. Esto puede ser un problema cuando se trata de crear un modelo de clasificación puesto que el clasificador podría obtener un 90% de precisión prediciendo solamente positivo, causando así *Overfitting*.

Por ello, en esta sección de la publicación, se explica cómo solucionar el problema de los datos no balanceados: utilizando técnicas de *Remuestreo*. En este caso, se han llevado a cabo las siguientes dos técnicas.

- *Under-sampling*: Hacer un muestreo de la clase mayoritaria para mantener solo una parte de estas reseñas.
- *Over-sampling*: Replicar algunos reseñas de la clase minoritaria.

Una vez resuelto el problema y teniendo los datos totalmente balanceados, se ha dividido el conjunto de datos en un 70% para *Train* y el 30% en *Test* y se han creado 5 modelos diferentes: **Regresor logístico**, **Naive Bayes**, **SVM**, **Random Forest** y **Red Neuronal** (Para más información sobre los modelos consultar la documentación [10]).

Para todos los modelos mencionados se han utilizado los parámetros por defecto puesto que no se ha observado mejoras con otras combinaciones.

Primero de todo se han entrenado cada uno de los modelos con su conjunto de Train y se han usado para predecir el sentimiento del conjunto de datos de Test, evaluando cada modelo para posteriormente compararlos y elegir los dos mejores.

3.1.3 Resultados

Una de las conclusiones más importantes que se ha obtenido a partir del análisis realizado, es la correlación entre la edad y la votación que se le da a una reseña tal y como hemos podido observar en la Figura 1.

A partir de este gráfico podemos observar diferentes aspectos tales como que la mayoría de las reseñas las dejan personas de entre 30 y 50 años aproximadamente o que, por otro lado, las personas que dejaron más comentarios también parecen ser los usuarios que dejan puntuaciones más altas.

En cuanto a la evaluación de los modelos de clasificación, se ha comparado el *Accuracy* de cada uno, para escoger así los dos mejores modelos (Tabla 1).

Modelo	Accuracy
Naive Bayes	69.54 %
Regresor Logístico	87.02 %
SVM	88.30 %
Red Neuronal	90.32 %
Random Forest	90.44 %

Tabla 1: Accuracy conseguido de cada modelo.

A parte de la exactitud del modelo, hay diferentes formas en que se puede medir el rendimiento de un clasificador. Primero se calcula la matriz de confusión, que muestra qué tan bien se comporta el modelo en cada una de las clases que entrenamos para predecir.

Por otro lado, se explica el concepto y cómo calcular la curva ROC, una de las métricas de evaluación más importantes para verificar el rendimiento de cualquier modelo de clasificación.

En la figura 3 podemos observar que se ha obtenido un 97% de probabilidad de que el modelo distinga bien entre las tres clases posibles: sentimiento positivo, neutral o negativo.

Por último también se han evaluado otras métricas como la *Precision, Recall* y *F1-Score* creando un informe de la clasificación.

Después de observar los resultados de todas las métricas de evaluación, se puede concluir que los dos mejores modelos: Random Forest y Red Neuronal, tienen una precisión y exactitud del 90% de predecir el sentimiento de una reseña correctamente.

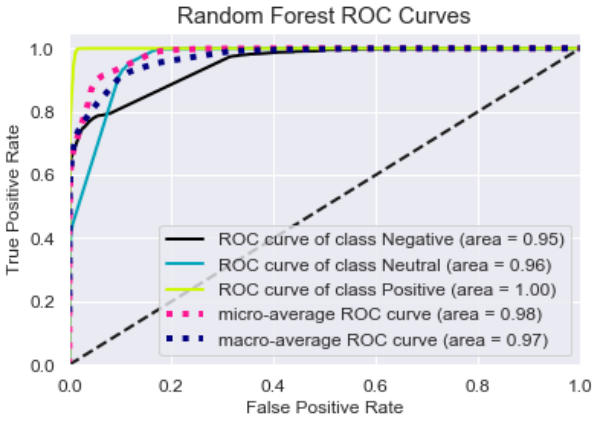

Fig. 3: Curva ROC de Random Forest.

3.2 GoodReads

En esta segunda publicación del Blog, se ha utilizado el conjunto de datos de **GoodReads-10k** [3] para crear dos de los sistemas de recomendación más populares. El objetivo principal ha sido crear dichos modelos para recomendar un libro a un usuario, ya sea por similitud entre libros o por usuarios similares a él.

Para ello, el primer paso antes de crear los dos sistemas, ha sido analizar los datos y por último evaluar los resultados obtenidos.

3.2.1 Análisis de datos

El primer paso de este post, al igual que en el anterior, ha sido explorar y limpiar los datos para poder entenderlos y aprovecharlos al máximo. En este conjunto de datos nos encontramos con más de 10.000 libros y con 23 atributos que caracterizan cada uno de ellos.

A continuación, se ha comprobado la existencia de valores nulos y en este caso, tras encontrarlos, se ha decidido eliminar aquellos que no aportaban valor al sistema de recomendación, tales como el ISBN/ISBN13 o los valores nulos que se han encontrado en el atributo de "original_title". Este último, se ha decidido eliminarlo y no rellenarlo con un valor aleatorio puesto que si no se encuentra el título original se puede buscar en el atributo "title".

Seguidamente, se han eliminado algunos de los atributos que no aportaban valor como las imágenes de la portada.

Como se ha comentado anteriormente, una de las tareas más importantes es el análisis de los datos, comprendiendo así qué atributos nos aportan más información a la hora de llevar a cabo los sistemas de recomendación.

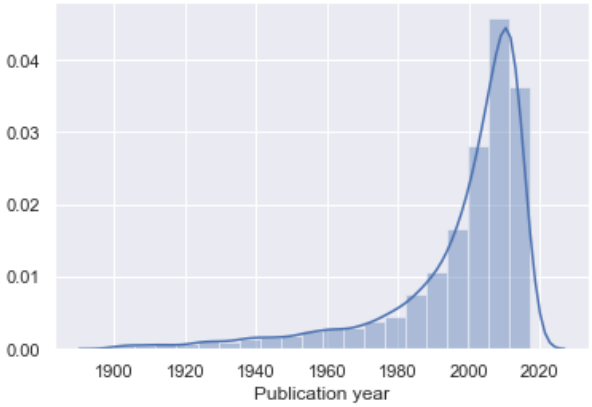

Fig. 4: Distribución de los años de publicación.

Por este motivo, se ha realizado diferentes análisis de la distribución de los atributos numéricos tales como el año de publicación del libro o las votaciones que se han hecho y para ello la mejor manera es realizando un histograma en el cuál se ha añadido también la curva de la distribución normal (Figura 4). Por otro lado, se ha analizado también la distribución de las votaciones de los usuarios a los libros (Figura 5) y se ha visualizado los tags más populares.

Seguidamente, en la publicación se encuentra explicado cómo encontrar la correlación entre diferentes atributos. En este conjunto de datos nos encontramos con atributos como la media de calificación de un libro, el año de publicación, cuántas personas han calificado el libro en total, a partir de

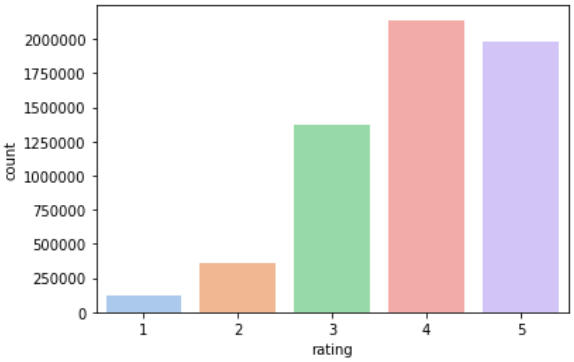

Fig. 5: Histograma de los *Ratings*.

los cuales se ha creado la matriz de correlación para averiguar qué atributos serán de mayor ayuda para el recomendador utilizando el Coeficiente de Pearson. Esto lo podemos observar en un mapa de color como el de la Figura 6.

3.2.2 Sistema de recomendación

Para conseguir el objetivo de crear dos sistemas de recomendación, se han reproducido los dos más populares: **Content-Based Filtering** y **Collaborative-Filtering** [11] y la siguiente sección del post está destinada a explicar la creación de cada uno.

El sistema de recomendación Content-Based sugiere elementos similares basados en un elemento particular. Este sistema utiliza características de los elementos como en este caso sería el género, la descripción de un libro, etc. para hacer las recomendaciones. Se ha utilizado el atributo de "Tags"que el usuario proporciona a un libro para crear el modelo, por lo que, la idea es que el sistema entienda lo que le gusta a un usuario y que recomiende libros similares a sus gustos según los tags de los libros.

En esta sección se explica detalladamente el funcionamiento de este sistema. Primero debe encontrar la similitud entre todos los libros utilizando todos los tags de cada libro y computando el TF-IDF (Term Frequency - Inverse Document Frequency), sub-área del Procesamiento de Lenguaje Natural. Esto crea un vector que se utiliza para evaluar cómo de importante es una palabra de un documento en dicho documento.

A continuación, gracias a este vector se ha calculado la matriz en la cual cada columna representa una palabra de la lista de Tags, y cada fila representa un libro, matriz con la cual se ha calculado la similitud coseno. La similitud entre dos libros es un número entre 0 y 1.

El segundo sistema que se explica en esta sección es el Collaborative Filtering, que se puede clasificar en dos tipos: User-Based o Item-Based. Se ha llevado a cabo el primero de ellos puesto que el segundo es muy similar al Content-Based. La idea principal de este sistema es que recomiende un libro usando la similitud entre usuarios. Por lo tanto, los datos que se han necesitado son las calificaciones de cada usuario por libro.

Este sistema de recomendación se ha realizado utilizando la librería Surprise [12] de Scikit que construye y analiza sistemas de recomendación. Proporciona varios algoritmos de predicción listos para usar, como la factorización basada en matriz y medidas de similitud. Se ha utilizado el algo-

ritmo SVD (Singular Value Descomposition) que es equivalente a la factorización de matriz probabilística que nos permite descubrir las características latentes subyacentes a las interacciones entre usuarios y elementos.

La manera de entrenar dicho modelo es similar a otros modelos de Machine Learning, intentará predecir la calificación de una determinada combinación de usuario-libro y comparará esa predicción con la real. La diferencia entre la calificación real y la predicha se ha medido utilizando medidas de error clásicas como RMSE (Raíz del error cuadrático medio) o MAE (Error medio absoluto).

3.2.3 Resultados

En la Figura 4 podemos observar como la mayoría de los libros que se encuentran en el conjunto de datos son libros publicados a partir del año 2000, por lo que los libros son relativamente recientes.

Por otro lado, al analizar la distribución de las votaciones que los usuarios les ha dado a los libros (Figura 5) podemos observar como la mayoría de los votos son entre 3 y 5 y con ello podemos crear dos hipótesis diferentes: que la gente tiende a votar solamente los libros que les ha gustado o, que las personas están condicionado hacia críticas positivas.

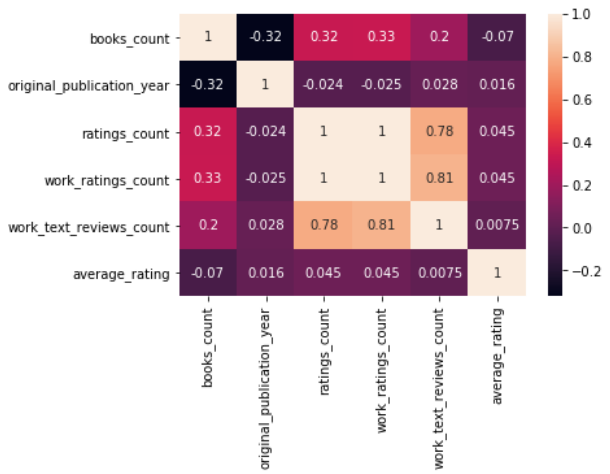

Fig. 6: Matriz de correlación entre diferentes características de un libro.

Gracias a la matriz de correlación que se muestra en la Figura 6 observamos que solamente hay correlaciones bajas entres los atributos y el promedio de calificación de un libro. Esto nos indica que no hay relaciones sólidas entre la calificación que recibe un libro y las otras variables como el número de calificaciones que ha recibido en total.

Por lo que respecta al primer sistema de recomendación, Content-Based, los resultados obtenidos al introducir un ejemplo (Libros similares a ("The Hunger Games") han sido los mostrados en la Tabla 2).

Título	Similitud
En llamas	96 %
Sinsajo	94 %
Divergente	74 %

Tabla 2: Libros similares a "The Hunger Games"obtenidos utilizando el sistema de recomendación Content-Based.

Finalmente, en cuanto a Collaborative-Filering, podemos evaluar el sistema con las métricas RMSE y MAE que se han comentado en el apartado anterior. Dicho sistema ha conseguido un RMSE del 81% y un MAE del 63%.

Igual que para el anterior recomendador, se ha probado con un ejemplo y los resultados obtenidos se muestran en la Tabla 3.

Título	Puntuación estimada
Eclipse	3.99
Amanecer	3.94
Luna Nueva	3.93
Cazadores de Sombras	3.80

Tabla 3: Libros recomendados a un usuario utilizando el sistema de recomendación Collaborative-Filtering

3.3 Intel Image Classification

En la última publicación del Blog, se ha utilizado el conjunto de datos de **Intel Image Classification** [5] con el objetivo de crear un modelo de clasificación con redes neuronales utilizando TensorFlow 2.0 y Keras [13], aprovechando así para crear una guía de uso en el Blog.

Para ello lo primero que se ha realizado ha sido un breve análisis de las imágenes, creando paso a paso un modelo básico de redes neuronales. A continuación, se han creado diferentes mejoras como añadir redes convolucionales, añadir técnicas de *Data Augmentation* o realizar *Fine-Tuning*.

3.3.1 Análisis de datos

En este apartado, al tratarse de un conjunto de datos de imágenes, el análisis no sigue la misma línea que los otros dos anteriores.

El primer paso ha sido explorar los datos, observando así que el conjunto contiene más de 14000 imágenes en el conjunto de entrenamiento y 3000 imágenes para el conjunto de test. Se trata de 6 clases diferentes: calles, montañas, glaciares, mar, edificios y bosque. Podemos observar un par de imágenes en la Figura 7

A continuación, se introduce el concepto Image Data Generator, una herramienta de preprocesamiento que proporciona Keras que genera lotes de datos de imágenes para cargar el conjunto de datos. Para poder utilizar dicho generador, se debe tener los datos organizados de una manera específica.

3.3.2 Modelo Deep Learning

Para conseguir el objetivo de crear un buen clasificador no basta con crear una red neuronal simple. Es por ello que en este post se explica paso a paso como realizar un clasificador que nos proporcione un buen resultado.

El primer paso ha sido realizar una red neuronal simple, explicando así cómo se entrena y se evalúa con TensorFlow. Para ello, primero de todo se introduce el concepto de *Early Stopping*, un método que nos permite especificar una cantidad de iteraciones de entrenamiento que permite detenerlo una vez que el rendimiento del modelo deja de mejorar en el conjunto de datos de validación. Esto nos permite a la vez reducir la posibilidad de *Overfitting*.

Fig. 7: Ejemplo de imágenes del conjunto de datos de Intel Image Classification.

Debido a que las redes neuronales simples no se escalan bien en el tratamiento de imágenes e ignoran la información aportada por la posición de los píxeles y la correlación con los vecinos, se introduce en el post el concepto de las redes convolucionales (CNN, Convolutional Neural Network) creando un nuevo modelo.

Dichas redes, se usan principalmente para el procesamiento y clasificación de imágenes porque reducen las limitaciones que nos podemos encontrar con un modelo de redes neuronales simple.

El siguiente paso ha sido introducir el concepto de Data Augmentation, una técnica que se puede utilizar para expandir artificialmente el tamaño del conjunto de entrenamiento creando así versiones modificadas de las imágenes en el conjunto de datos para reducir *Overfitting*.

Algunas de las técnicas utilizadas han sido las siguientes:

- *Shear_range*: Aplica recortes al azar en la imágenes.
- *Zoom_range*: Aplica zoom aleatorio dentro de las imágenes.
- *Horizontal_flip*: Aplica giros horizontalmente a la mitad de las imágenes.

Un ejemplo del uso de esta técnica se puede observar en la Figura 8:

Fig. 8: Ejemplo de las tres técnicas de Data Augmentation.

Por último, en esta sección se introduce también el concepto de a transferencia de aprendizaje (Transfer Learning) para entender y aplicar posteriormente Fine-Tuning al modelo. Transfer Learning ocurre cuando usamos el conocimiento que se ha obtenido al resolver un problema y lo aplicamos a un problema nuevo, pero similar. Fine-Tuning es una manera de utilizar esta transferencia del aprendizaje que, como su propio nombre indica, se trata de coger un modelo pre-entrenado para un problema determinado y luego ajustar el modelo para que realice una segunda tarea similar.

En este caso, se ha llevado a cabo un Fine-Tuning utilizando la red **MobileNetV2** [14] y se ha descongelado algunas de las capas que en el modelo pre-entrenado se encuentran más cerca de la parte superior. Se ha aplicado esta técnica y no otro, porque las capas convolucionales inferiores detectan características de bajo nivel como bordes y curvas, mientras que el nivel superior, que es mas especializado, detecta características que son aplicables a nuestro problema.

3.3.3 Resultados

Para cada uno de los modelos creados, podemos observar los resultados obtenidos en la evaluación en la Tabla 4.

Modelo	Loss	Accuracy (%)
Red Neuronal Simple	1.37	50.99
CNN	0.52	81.99
CNN aplicando Data Augmentation	0.49	83.17
Fine-Tuning	0.26	91.73

Tabla 4: Tabla comparativa de los resultados de los diferentes modelos Deep Learning.

La primera red neuronal constaba de 3 capas ocultas, con una función de activación 'Relu'. Se entrenó en 12 épocas y tardó aproximadamente 5 minutos. Seguidamente, la CNN constaba de 2 capas convolucionales, 2 MaxPool2D y otras dos capas ocultas, que por su parte tardó 7 épocas en conseguir el resultado observado en la tabla aproximadamente en 6 minutos. Observando los resultados obtenidos, podemos concluir que al realizar un modelo de redes convolucionales mejora bastante, consiguiendo así un 82% de Accuracy.

Al aplicar las técnicas de Data Augmentation, la arquitectura de la red no ha cambiado, pero sí que el tiempo de entrenamiento aumentó a 40 minutos en 16 épocas. Observando los resultados, el Accuracy no mejora tanto como se esperaba pero, ha reducido Overfitting.

Por último, como era de esperar, realizando Fine-Tuning es como más Accuracy se consigue, y cuando se reduce más el sobre-ajuste. Este modelo tardó 16 épocas y aproximadamente 30 minutos en entrenar.

4 Competición en Kaggle

Una de las últimas partes del proyecto ha sido participar en una competición real de Machine Learning de Kaggle [15] en equipo. El equipo ha estado compuesto por otro estudiante que ha realizado un proyecto similar y yo. Para ello, la competición seleccionada ha sido **Shelter Animal Outcomes Competition** [16].

El objetivo principal a parte de colaborar y participar como equipo ha sido el desafío de resolver un problema utilizando métodos combinados de los modelos creados por los dos miembros del equipo.

Se ha seguido la misma estructura que para los demás conjuntos de datos donde primero se ha explorado los datos, se han analizado y se han explicado los modelos de Machine

Learning que se han llevado a cabo y, por último, se han evaluado los resultados.

Cada año en Estados Unidos, aproximadamente más de 7,6 millones de animales de compañía terminan en refugios. Muchos de ellos son adoptados tras unos meses, pero muchos de ellos no son tan afortunados y son sacrificados. Es por ello que el objetivo de esta competición consiste en predecir qué va a pasar con cada animal: si es adoptado, si muere, si lo llevan a otro lugar, etc.

4.1 Análisis de datos

El primer paso antes de crear el modelo ha sido explorar y analizar los datos tal y las relaciones entre los atributos, tal y como se ha hecho en los otros tres conjuntos de datos.

El dataset contiene más de 26 mil datos de diferentes animales, con 10 características diferentes como el tipo de animal, la raza o la edad.

El primer atributo que se ha encontrado ha sido el tipo de animal en el que se ha podido comprobar que solamente se tratan dos animales: perros y gatos. La distribución de dicho atributo en el conjunto de datos se puede observar en la Figura 9. El siguiente paso ha sido analizar los diferentes tipos de salida que puede haber (Outcome) y estos han sido: Muerte, adopción, eutanasia, de vuelta con el dueño, o cambio de refugio.

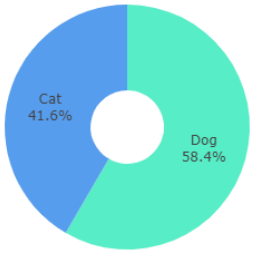

Fig. 9: Gráfico de proporciones de los tipos de animales.

Mientras se ha explorado el conjunto de datos, se ha observado que el atributo del género del animal, incluye a la vez un indicador del estado de reproducción (Castrado, No castrado), por lo que se ha necesitado procesar dicho atributo dividiendo la información.

Por otro lado, también se ha encontrado un problema en el atributo de la edad ya que éste define la edad en diferentes unidades tales como días, semanas, meses o años. Lo que se ha hecho para solucionar este problema ha sido modificar todos los valores pasándolos a una sola unidad: años.

Tras observar los diferentes atributos, se ha podido concluir que atributos como Breed o Color, no nos aportan demasiada información a la hora de entrenar nuestro modelo puesto que tienen un total de 1380 y 366 valores únicos cada uno.

A continuación, se ha comprobado la existencia de valores nulos. Afortunadamente entre los valores nulos que se han encontrado hay dos atributos que no se van a usar para el entrenamiento por lo que podemos prescindir de ellos.

En cambio, se han encontrado muy pocos valores nulos en los atributos de Sex y Age, y en este caso sí que se han eliminado puesto que se utilizan para entrenar el modelo.

Por otra parte, se ha analizado la distribución del tipo de salida del refugio agrupándolos por el género del sexo,

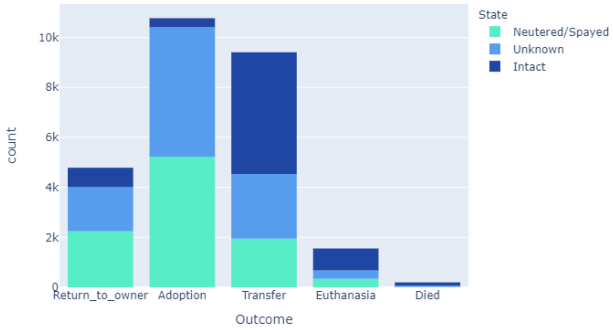

Fig. 10: Relación entre la categoría del estado de reproducción y el motivo de la salida del refugio.

observando así que el tipo de salida es proporcional para los dos géneros.

A continuación, también se ha analizado la relación entre el atributo del estado de reproducción y el motivo de la salida del animal (Figura 10).

En último lugar, un análisis interesante que se ha llevado a cabo ha sido agrupar los datos por la edad. Dado que el atributo de la edad del animal se encontraba en diferentes unidades, tal y como se ha comentado, se convirtió todos en años. Una vez hecho esto, se definieron tres diferentes categorías: Puppy, Adult y Senior. Una vez se tuvo los datos agrupados, se representó la relación entre la categoría de edad y el tipo de salida del refugio como se puede observar en la Figura 11.

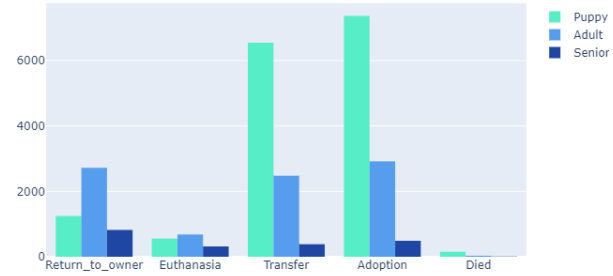

Fig. 11: Relación entre la categoría de edad del animal y el motivo de la salida del refugio.

Podemos extraer dos conclusiones interesantes a partir de la Figura 10, como que los cachorros son los más comunes a la hora de adoptar o mover de refugio. También se puede observar que los animales que suelen volver con los dueños son los adultos.

4.2 Modelo de Machine Learning

Tras explorar y analizar los datos, se ha decidido crear y entrenar un modelo combinado de diferentes modelos para predecir el motivo de salida del refugio del animal con el conjunto de datos de Test.

El conjunto de Test no tiene el atributo a predecir (Outcome) ya que es el que se debe predecir y entregar a la plataforma de la competición y el que será evaluado obteniendo así el equipo una puntuación.

Al tratarse de una competición que se ha hecho en equipo, a la hora de crear los modelos de predicción se han dividido en dos partes: por mi parte he creado dos modelos

Machine Learning, y mi compañero de equipo ha creado diferentes redes neuronales. Posteriormente, se ha hecho un modelo combinado de las dos partes.

El primer paso para la creación de los dos modelos ha sido dividir el conjunto de train y test. Una vez preparados los datos, se ha creado un modelo**Random Forest**, y otro **XGBoost**, entrenándolos y evaluándolos por separado.

Se ha utilizado el método GridSearchCV para encontrar la combinación de parámetros que mejor funciona para cada modelo. Los mejores parámetros y que por lo tanto se han utilizado han sido los siguientes:

- Random Forest: Función para medir la calidad de una división (Criterion): Gini. Número de árboles a 240 y la profundidad máxima a 8.
- XGBoost: Con un total de 5 clases, y máxima profundidad a 5. Learning Rate a un 0.5.

4.3 Ensemble de los modelos

Crear un *ensemble* de los modelos significa que cada modelo produce una predicción diferentes y se combinan para obtener una única predicción. Una de las ventajas de crear estos métodos combinados es que al combinar diferentes modelos, como cada modelo funciona diferente, los errores que puedan cometer tienden a compensarse.

El método de ensemble que se ha escogido ha sido un método de votación por mayoría utilizando VottingClassifier de la librería Sklearn [17]. Hay dos tipos: Hard voting y Soft voting. Se ha escogido Soft voting en el que la clase de salida es la predicción basada en el promedio de la probabilidad dada a esa clase.

4.4 Resultados

A la hora de evaluar los dos modelos Machine Learning, se ha tenido en cuenta el Accuracy y los resultados obtenidos se pueden observar en la Tabla 5.

Modelo	Accuracy (%)
Random Forest	60.09
XGBoost	61.03

Tabla 5: Resultados obtenidos de los modelos de Random Forest y XGBoost.

A continuación, al evaluar el modelo combinado de las dos partes del equipo (los modelos de Deep Learning del compañero de equipo y los modelos Machine Learning comentados), se ha conseguido un 64,04% de Accuracy final.

Tras crear el modelo combinado, se han predicho los datos del conjunto de test para poder subirlo a la competición (en la plataforma de Kaggle), donde se evalúa con "Multiclass Loss", obteniendo un 0,88 de puntuación. Esta puntuación es aproximadamente un 30% más baja que el mejor modelo observado en la tabla de líderes.

Por último, se ha hecho un estudio del equipo con mejores resultados para entender cómo podríamos mejorar nuestra puntuación.

Se ha observado que este equipo ha hecho un análisis de los datos en más profundidad, limpiándolos y creando muchas más variables a partir de las ya existentes, usándolas para entrenar el modelo. Además, este equipo ha creado solamente un modelo: Random Forest.

5 Blog DataUAB

El Blog DataUAB [1] fue creado un par de años atrás, con el objetivo mencionado anteriormente. Una de las últimas tareas del proyecto fue desarrollar nuevas funcionalidades y mejoras estéticas al Blog.

El Blog consiste básicamente en la página principal, una sección de Tags, una sección de contacto y por último otra sección con la información de todos los administradores, tutores y contribuidores.

En la página principal es donde se encuentran publicados los artículos, tal y como podemos ver en la Figura 11 (ya incluidos los tres nuevos artículos).

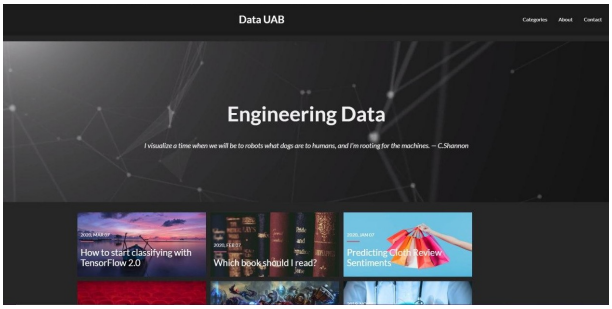

Fig. 12: Página principal Blog DataUAB.

Las mejoras desarrolladas han sido las siguientes:

- **Sección *About*:** El primer cambio realizado ha sido añadir tanto el encabezado como el pie de página a esta sección para darle mayor rigidez y una sensación de continuidad a la página. Además, se ha modificado la estructura con las diferentes fotografías y los enlaces de interés de los miembros del club. Esto lo podemos apreciar en la Figura 13.
- **Sección de contacto:** Esta sección consiste en un formulario donde los usuarios pueden enviar dudas o recomendaciones. La mejora estética ha sido un fondo de pantalla que aporta más seriedad al Blog, añadiendo el encabezado y el pie de página para seguir la misma estructura que en el resto del Blog.

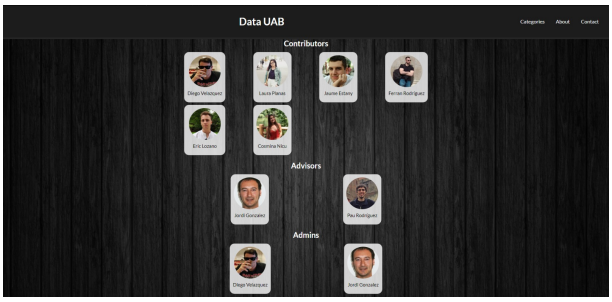

Fig. 13: Mejora estética en la sección *About*.

6 CONCLUSIONES

El objetivo principal de este trabajo se ha cumplido satisfactoriamente extendiendo el Blog DataUAB con tres nuevas publicaciones didácticas y con mejoras estéticas en algunas secciones para una mayor rigidez y una sensación de continuidad.

A lo largo de este artículo se han estudiado y explicado las diferentes técnicas Machine Learning y Deep Learning aplicadas a los diferentes conjuntos de datos juntamente con su análisis y su evaluación de los resultados.

Analizando y creando un modelo de predicción de sentimientos para el dataset de Women's E-commerce Clothing Review se ha podido ver cómo la mayoría de los usuarios que suelen dejar su reseña en la página, suelen ser los que les ha gustado mucho el producto y dejan puntuaciones más altas.

A la hora de crear los sistemas de recomendación se ha podido observar la importancia de tener una gran cantidad de datos a la hora de recomendar un libro. Por otro lado, se ha observado cómo no existe una correlación sólida entre la calificación que recibe un libro y el número de calificaciones totales.

En un problema de Deep Learning, como el de Intel Image Classification, se concluye que aplicando técnicas como Data Augmentation ayuda a reducir el sobre-ajuste a la hora de entrenar un modelo.

Por último, se ha participado en una competición online, aprendiendo nuevas técnicas y comprendiendo cómo los ganadores han conseguido los resultados, además de tener la oportunidad de trabajar en equipo.

AGRADECIMIENTOS

Quiero mostrar mi agradecimiento a mi tutor Jordi González y al co-tutor Pau Rodríguez, por su ayuda y por aconsejarme en todo momento, mostrando interés hacia el proyecto y resolviendo las dudas que hayan surgido durante el camino.

Por otro lado agradecer a mi compañero, Eric Lozano Férriz, el cuál ha sido de gran apoyo durante todo el trabajo, en especial en la competición en la cual hemos formado equipo.

Por último, agradecer a todos los coordinadores y profesores de la universidad, en especial a Jordi Pons por mantenernos siempre al día y ayudar a que todo vaya bien debido al COVID-19 que ha hecho que todo este proceso fuese diferente a otros años y nuevo para todos nosotros.

REFERENCIAS

[1] ≪Data UAB≫. [Online]. Disponible en: https://datauab.github.io/. [Accedido: 25-jun-2020]

[2] ≪Women's E-Commerce Clothing Reviews≫. [Online]. Disponible en: https://kaggle.com/nicapotato/womens-ecommerce-clothing-reviews. [Accedido: 25-jun-2020]

[3] ≪goodbooks-10k≫. [Online]. Disponible en: https://kaggle.com/zygmunt/goodbooks-10k. [Accedido: 25-jun-2020]

[4] ≪Goodreads≫, Goodreads. [Online]. Disponible en: https://www.goodreads.com/. [Accedido: 25-jun-2020]

[5] ≪Intel Image Classification≫. [Online]. Disponible en: https://kaggle.com/puneet6060/intel-image-classification. [Accedido: 25-jun-2020]

[6] ≪TensorFlow≫, TensorFlow. [Online]. Disponible en: https://www.tensorflow.org/?hl=es. [Accedido: 25-jun-2020]

[7] ≪Project Jupyter≫. [Online]. Disponible en: https://www.jupyter.org. [Accedido: 25-jun-2020]

[8] ≪English stop words≫, CountWordsFree. [Online]. Disponible en: https://countwordsfree.com/stopwords. [Accedido: 28-jun-2020]

[9] ≪Natural Language Toolkit — NLTK 3.5 documentation≫. [Online]. Disponible en: https://www.nltk.org/. [Accedido: 25-jun-2020]

[10] M. Przybyla, ≪Machine Learning Algorithms. Here's the End-to-End.≫, Medium, 30-may-2020. [Online]. Disponible en: https://towardsdatascience.com/machine-learning-algorithms-heres-the-end-to-end-a5f2f479d1ef. [Accedido: 27-jun-2020]

[11] S. Luo, ≪Intro to Recommender System: Collaborative Filtering≫, Medium, 06-feb-2019. [Online]. Disponible en: https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26. [Accedido: 27-jun-2020]

[12] N. Hug, ≪Home≫, Surprise. [Online]. Disponible en: http://surpriselib.com/. [Accedido: 25-jun-2020]

[13] ≪Keras: the Python deep learning API≫. [Online]. Disponible en: https://keras.io/. [Accedido: 25-jun-2020]

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, y L.-C. Chen, ≪MobileNetV2: Inverted Residuals and Linear Bottlenecks≫, arXiv:1801.04381 [cs], mar. 2019 [Online]. Disponible en: http://arxiv.org/abs/1801.04381. [Accedido: 27-jun-2020]

[15] ≪Kaggle: Your Machine Learning and Data Science Community≫. [Online]. Disponible en: https://www.kaggle.com/. [Accedido: 25-jun-2020]

[16] ≪Shelter Animal Outcomes≫. [Online]. Disponible en: https://kaggle.com/c/shelter-animal-outcomes. [Accedido: 25-jun-2020]

[17] ≪ML — Voting Classifier using Sklearn≫, GeeksforGeeks, 23-nov-2019. [Online]. Disponible en: https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/. [Accedido: 27-jun-2020]