



Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en
Inteligencia Artificial

**KAI-U: Sistemas de
Recomendación y
Modelos Predictivos
para fomentar el
consumo audiovisual
español del anime**

Trabajo Fin de Máster

Presentado por:

Nicolás Felipe Trujillo Montero

Jesús Carlos Avecilla de la Herrán

Dirigido por:

Iñaki Fernández Pérez

Ciudad: Málaga y Cádiz, España

Fecha: 10 de Julio de 2024

Índice de Contenidos

1. Introducción	4
1.1. Introducción Contextual	4
1.2. Introducción Mercadotécnica	7
1.3. Soluciones Planteadas	9
2. Contexto y Estado del arte	10
2.1. Contexto	10
2.1.1. Comienzos del anime	10
2.1.2. La etiqueta 'otaku' como marcador cultural	13
2.1.3. La industria del anime en la actualidad	15
2.1.4. Relevancia de nuestra solución en este contexto	18
2.2. Estado del Arte	22
2.2.1. Implicación en comunidades de IA	23
2.2.2. Premisa (I)	25
A.- Cuestiones acerca de los Sistemas Recomendadores	27
B.- <i>Webinar</i> : “Creando Sistemas de Recomendación desde cero”	30
C.- Sistemas Recomendadores en Plataformas de Streaming .	36
2.2.3. Premisa (II)	37
A.- Manuel Cobo	37
B.- Traductor de Manga	38
2.2.4. Validación de las premisas	39
2.2.5. Solución Planteada	40
1.- Proceso ETL	41
2.- Solución 1: Sistema Recomendador	44

3.- Solución 2: Sistema Predictivo	46
3. Objetivos y Metodología	49
3.1. Objetivos a nivel proyecto	49
3.2. Objetivos a nivel contextual	51
3.3. Metodología	53
3.4. Tecnologías Investigadas	57
4. Proceso ETL	59
4.1. Data Scraping (Extracción)	60
4.2. Data Curation (Transformación y Carga)	65
4.2.1. Dataframe del Sistema Recomendador	65
4.2.2. DataFrame del Sistema Predictivo	67
5. Sistema Recomendador de Animes basado en el contenido	68
5.1. Identificación de Requisitos	68
5.1.1. Enfoque actual de los sistemas recomendadores	68
5.2. Descripción de la herramienta software desarrollada	71
5.2.1. Obtención y Preprocesamiento de los Datos	71
5.2.2. Técnicas utilizadas	72
Named-Entity Recognition	72
Text Similarity	74
Clustering	75
Apuntes sobre las técnicas no usadas	75
Topic Modeling	76
5.2.3. Desarrollo de las técnicas	77
Caso de Uso - NER	77
Caso de uso - Text Similarity	79
Caso de uso - Clusterización	80
Caso de uso - Topic Modeling	83
5.3. Aplicación para el end-user	85
5.4. Análisis del desarrollo	87

5.5. Evaluación	91
6. Sistema Predictivo del Impacto del Anime en Plataformas de Streaming	94
6.1. Identificación de Requisitos	94
6.1.1. Enfoque actual	94
6.2. Descripción de la herramienta desarrollada	98
6.2.1. Obtención y Preprocesamiento de los Datos	98
6.2.2. <i>Exploratory Data Analysis</i> (EDA): Análisis Exploratorio de los Datos	100
6.2.3. <i>Feature Engineering</i> : Creación de Atributos	107
6.2.4. Construcción y ejecución de modelos y técnicas	110
Preparación de Datos	110
Clasificación Desbalanceada (Imbalanced Classification)	111
Explicación de la Bondad del Ajuste	112
Introducción a la comparación y evaluación de Modelos y Técnicas	115
Modelo Lógico	115
Modelo Geométrico	120
Modelo Probabilístico	124
Modelo Mixto	125
Ensembles y Tipos	125
6.3. Conclusión	128
7. Conclusiones y trabajos futuros	130
7.1. Conclusiones de los Sistemas	130
7.2. Análisis de la encuesta de evaluación	131
7.3. Feedback de la encuesta	133
7.4. Análisis DAFO	135
7.5. Aspectos sin implementar	136
7.6. Lineas a futuro	137

Índice de Ilustraciones

2.1.	FY2023 Bandai Namco	12
2.2.	Los fansub añaden karaoke o el nombre de los contribuidores	18
2.3.	Planteamiento del Proyecto	22
2.4.	Spain AI («Spain-AI», 2023)	23
2.5.	Repercusión mediática de las comunidades (Cádiz, 2023; Peláez, 2023)	24
2.6.	Creando Sistemas de Recomendación desde cero (González-Fierro, 2023)	26
2.7.	Planteamiento de la técnica (Nvidia, 2024)	26
2.8.	Tipos de Sistemas Recomendadores (González-Fierro, 2023)	30
2.9.	Diagrama de funcionamiento del algoritmo SAR (Microsoft, 2023) . .	32
2.10.	Diagrama del Proceso ETL («Cloud Skew web», 2023)	41
2.11.	Diagrama de la investigación del Sistema Recomendador («Cloud Skew web», 2023)	44
2.12.	Diagrama de la investigación del Sistema Predictivo («Cloud Skew web», 2023)	46
3.1.	Captura del servidor de Discord	54
3.2.	Método Científico	55
4.1.	Captura de MyAnimeList (Gyssler, 2023)	61
4.2.	Columnas obtenidas de MyAnimeList (Gyssler, 2023)	63
5.1.	Caso de Uso: Engagement	85
5.2.	Caso de Uso: Reach	86
5.3.	Instanciando el modelo de Gliner	87
5.4.	Anotaciones en Gliner sobre series de fútbol	88

5.5. Anotaciones en Gliner referentes al tono	89
5.6. Anotación en Gliner referente al género basado en demografía	90
5.7. (a) Named Entity Recognition	91
5.8. (b) Clusterización	92
5.9. (c) Text Similarity	92
5.10. (d) Topic Modeling	93
6.1. Interpretación de los datos nulos	100
6.2. Distribución de notas (score) según el tipo de source	101
6.3. Matriz de Correlación de Probabilidades	103
6.4. Distribución de los cambios en las notas entre la nota de un anime y su adaptación según los usuarios	104
6.5. Imputación de los valores nulos de las columnas por la media de sus columnas correspondientes	107
6.6. Agrupamiento de los valores por contenedores conforme a la puntuación y el <i>ranking</i>	108
6.7. Ejemplo de codificación <i>One-Hot Encoding</i> (Rençberoğlu, 2024) . . .	109
6.8. Ejemplo de uso de SMOTE con el <i>Dataset</i> de entrenamiento	111
6.9. Ejemplo de resultado de la ejecución de la función <code>evaluate_model</code> .	113
6.10. Ejemplo de funcionamiento de un árbol de decisión (Awan, 2024) . .	116
6.11. Métricas que utiliza un árbol convencional vs. árbol ID3 (Mehdi Hamidi, 2024)	117
6.12. Ejecución del clasificador (<i>DecisionTreeClassifier</i>) con 30 ejecuciones con distintas <i>seeds</i> y con 30 iteraciones máximas	118
6.13. Ejemplo de Modelo Geométrico Lineal (Saradalakshmi8074, 2024) . .	120
6.14. Ejemplo de Modelo Geométrico basado en distancia (Saradalakshmi8074, 2024)	121
6.15. Ejecución del clasificador (<i>Logistic Regression</i>) con 30 ejecuciones con distintas seeds y con 30 iteraciones máximas	122
6.16. Ejecución del clasificador (<i>SVC</i>) con 30 ejecuciones con distintas seeds y con 30 iteraciones máximas	122

6.17. Ejecución del clasificador (<i>LinearSVC</i>) con 30 ejecuciones con distintas seeds y con 30 iteraciones máximas	123
6.18. Ejecución del clasificador (<i>RandomForestClassifier</i>) con 30 ejecuciones con distintas <i>seeds</i> y con 30 iteraciones máximas	127
7.1. Feedback aportado por las personas encuestadas	133

Resumen

Esta investigación se centra en el **consumo de anime en plataformas de streaming españolas**, un formato audiovisual japonés que ha ganado popularidad internacionalmente. El aumento en la oferta de *anime* requiere una selección precisa para los consumidores, lo que presenta un desafío para las distribuidoras nacionales.

Se parte de **dos premisas**:

- Las **recomendaciones de series se centran en** criterios clásicos **basados en demografía**, en lugar de las necesidades del usuario.
- Las **distribuidoras no pueden realizar grandes inversiones para localizar productos audiovisuales sin que estas puedan parametrizar previamente el impacto de los *animés*** que salen cada temporada.

Las **soluciones propuestas** para las dos premisas serían:

- Un **Sistema Recomendador basado en sinopsis** utilizando técnicas *Procesamiento del Lenguaje Natural*.
- Una solución de **Machine Learning** para predecir qué series serán relevantes en el mercado español por temporada.

El **objetivo** es que, mediante la investigación realizada, las **distribuidoras locales obtengan ventajas competitivas** y capten la **atención mediática** en un mercado dominado por grandes plataformas de *streaming*, viendo donde implementar una solución tecnológica que aborde estas necesidades.

Palabras Clave: *Sistema Recomendador, Estudios Orientales, Machine Learning, Procesamiento del Lenguaje Natural*

Abstract

This research examines the consumption of anime on streaming platforms, focusing on a Japanese audiovisual format that has garnered international popularity. The rise in anime consumption poses a challenge for local distributors in Spain due to the need for prior selection.

Two premises are identified:

- Recommendations for users are based on traditional demographics rather than being user-oriented.
- Distributors make significant investments without anticipating the impact of seasonal anime.

In order to address these issues, the following solutions are proposed:

- A Recommendation System (*Sistema Recomendador*) based on narrative similarities, utilizing Natural Language Processing techniques (*Procesamiento del Lenguaje Natural*).
- A Machine Learning (*Machine Learning*) solution to predict which series will be relevant enough in the Spanish market.

The objective is to provide local distributors with a competitive advantage and attract media attention in a market dominated by major streaming platforms through the implementation of these technological solutions.

Keywords: Recommender Systems (*Sistema Recomendador*), Marke-

ting Solution (*Sistema Predictivo*), *Machine Learning*, *Natural Language Processing* (*Procesamiento del Lenguaje Natural*)

Capítulo 1

Introducción

1.1. Introducción Contextual

En términos generales, se define como *anime* a cualquier forma de animación japonesa. En su mayoría, vienen precedidas de su correspondiente *manga* en forma de cómic serializado en revistas de forma semanal, para después ser publicada en tomos recopilatorios.

Teniendo en cuenta este impacto previo, los estudios de animación deben garantizarse un éxito debido a los altos costes de producción de lanzar capítulos semanales por parte de las cadenas de televisión japonesas. Al mismo tiempo, las ventas de productos asociados en forma de *merchandising* han ido siendo la principal fuente de ingresos, por lo que estas industrias trabajan en paralelo.

Para la comunidad denominada *otaku*, existe un problema ontológico donde se contempla una comparación del contenido dividida por demografías en cuanto a género y/o edad, que han ido cambiando en el tiempo conforme han aparecido nuevos matices.

De forma clásica, existían las revistas de *manga* orientadas para chicas llamadas *shoujo*, para público maduro existían los *seinen* o aquellas destinadas a un target juvenil *shounen*, que suelen ser la mayoría de series *mainstream* consumidas por

un público mayoritario.

Sin embargo, **no hay una tipología precisa a la hora de categorizarlas**, ya que una serie para jóvenes puede ser de varios subtipos distintos:

- ***Dragon Ball*, en España conocida como Bola de Dragón** (Toriyama, 1986) es un *anime* para jóvenes de género **acción**, conocidos como *nekketsu*.
- ***Captain Tsubasa*, en España conocida como Oliver y Benji** (Takahashi, 1983) es un *anime* para jóvenes de género de **deportes**, llamados *spokon*

Además, **estas series de éxito comercial son recomendadas por defecto, aunque no existe un hilo temático entre ellas**. Los criterios actuales para realizar recomendaciones se basan en la relevancia de la serie en el tiempo, **sin considerar otras distinciones de similitud temática en lo narrativo**.

Con el paso de los años, la oferta comenzó a ser tan amplia que los aficionados occidentales crearon bases de datos como *MyAnimeList* (Gyssler, 2023) para seguir los lanzamientos por temporada. De hecho, **los usuarios de dicha plataforma recomiendan series por comparación de forma manual**, es decir, de forma **subjetiva**.

Sin embargo, **no ha sido lo suficientemente flexible para adaptarse a los nuevos matices en la reciente década:**

- En la esfera del *shounen* se ha popularizado el género de los *isekais* donde las tramas se centran en **personajes que se teletransportan a mundos diferentes al nuestro, al estilo de los MMORPG**, de forma similar a una simulación.
- Asociado al *shoujo* aparece el género *Slice of life* tratando sobre escenas costumbristas de la vida cotidiana, normalmente en la época estudiantil. De aquí, surge el *moe* como género propio consumido mayormente por hombres, aunque partiendo con unos premisas que resultarían más pertinentes a mujeres por su estética.
- Del mismo modo, están apareciendo **series contemplativas sobre el paso del tiempo y una sensibilidad por la naturaleza conocidas como “Mono no Aware”** donde todavía no existe un consenso sobre si considerarlo como una nueva tendencia.

Con estos nuevos matices, así como la inclusión de *OVA*s o *remakes*, al igual que *retellings* de las mismas historias, **han vuelto estas bases de datos inabarcables a la hora de clasificar de forma eficaz**, sin ser útiles a la hora de descubrir nuevo contenido relevante para aficionados en base a intereses previos.

El consumo se ha vuelto transversal y no se asocia por sexo o género, por lo que, **conforme se ha ensanchado la variabilidad de la demografía en los consumidores**, más complejas se vuelve cualquier tipo de clasificación ontológica ante esta inmensa oferta.

1.2. Introducción Mercadotécnica

Desde el apartado mercadotécnico, la creciente demanda de *anime* por parte de los occidentales ha cambiado los patrones de consumo:

1. El primer contacto de los españoles con el *anime* vino de la televisión con al surgimiento de las televisiones autonómicas que doblaban las series en cada lengua regional, a la vez que se traducía *manga*, sobre todo en Cataluña.
2. Con el auge de internet empezaron a surgir comunidades de aficionados denominadas *fansubs* cuyo objetivo era subtítular series que no captaban la atención de los medios occidentales, compartiendo las mismas como archivos de forma pirata.
3. Con la llegada de las plataformas de *streaming*, el acceso legal para consumir *anime* se hacía cada vez más accesible. *Netflix* (Reed Hastings, 2023) empezó a financiar la producción de sus propias series, estimulando a su vez el surgimiento de plataformas de *streaming* específicas de *anime* como *Crunchyroll*.

Las distribuidoras nacionales que doblaban y localizaban *anime* para el mercado nacional no podían seguir el ritmo de traducción por parte de los *fansubs*.

Además, no pueden competir por las licencias de *animés* en estas plataformas que vienen negociadas antes de publicarse, por lo que no cuentan con un indicador de éxito para las series que no sean populares, a la hora de tomar decisiones sobre qué contenido es digno de ser localizado.

Es más, para que una serie sea considerada de éxito, depende de un impacto, tanto en el mercado japonés en forma de venta en físico, como a nivel internacional en forma de *merchandising* que no resulta parametizable.

La popularidad de una reciente serie llamada *Chainsaw Man* (Seko, 2022) ha sido considerada un fracaso por sus propios creadores a pesar de su popularidad (Villa, 2023), ya que en su lanzamiento en físico no han vendido las copias estimadas, dudando sobre su posterior localización en el mercado internacional.

Por muy popular que se haga una franquicia no hay forma de evitar el consumo pirata por accesibilidad al producto sin que esta sea cuantificable, ni monetizable.

De este problema surgen los *simulcast*, eventos donde se publican las series al mismo tiempo a nivel mundial, y creando una experiencia colectiva que vivir en plataformas concretas. Tanto *Selecta Visión* como *Jonu Media* como principales distribuidoras españolas han tratado de crear estas dinámicas en sus portales *Animebox* y *Jonu Play* (GROUP, 2023; S.L.U., 2023).

No todas las series que licencian son capaces de crear estos “eventos”, ya que esta economía de la atención no tiene herramientas que la dinamicen hacia sus plataformas, por lo que se ofrecen estas soluciones para esta compleja situación.

1.3. Soluciones Planteadas

El *naming* de la solución como “**KAI-U**” tendría dos justificaciones:

1. La primera, es la **homofonía con el naponismo kaiju** como un monstruo propio de la cultura audiovisual japonesa, similar a Godzilla.

Además, remite al concepto del “**Leviatán**” proveniente del filosofo Thomas Hobbes, al ser un sistema que centraliza procesos y regula comportamientos.

2. La segunda, es que **KAI** es un acrónimo de “**Knowledge Artificial Intelligence**” mientras que la “**U**” remite al pronombre “**YOU**”, dando poder a los usuarios.

Con la **herramienta de recomendación planteada como un Sistema Recomendador basado en sinopsis** se cubriría tanto el **problema de reach, captando a gente** que se vea interesada en usar las plataformas de *streaming*, como **de engagement**, para hacer que una vez dentro se **mantengan activos** consumiendo otras series similares.

Del mismo modo, **gracias a la herramienta de Análisis de Impacto, planteada como un Sistema Predictivo**, se explorará diferentes métricas de éxito que ayuden a **averiguar qué series merecen la pena ser licenciadas** por las distribuidoras nacionales, midiendo su impacto y/o relevancia con respecto a los usuarios.

Capítulo 2

Contexto y Estado del arte

2.1. Contexto

Como parte de una contextualización sobre la industria del *anime*, nos basaremos en la cronología del libro de **Marc Steinberg** (Steinberg, 2012) donde se explican las **diferencias mercadotécnicas con respecto a la animación occidental**:

1. Comienzos del *anime* como manifestación cultural japonesa
2. La identidad “*otaku*” como target comercial
3. Particularidades de la industria en la actualidad

Habiendo analizando la relevancia de la propuesta, se concluirá con las aportaciones que daría la solución a dicho contexto.

2.1.1. Comienzos del anime

Según el académico Paul Wells, la **animación** proveniente del verbo latino *animare* significa “dar vida” por lo que se define como la habilidad de crear artificialmente una ilusión de movimiento en formas inanimadas fotograma a fotograma, en contraposición a la fotografía como formato audiovisual (Wells, 1998)

A nivel temático, el *anime* presentaba historias complejas que ahondaban en la psicología de los personajes, en lugar del formato cuento que caracterizaba las producciones de Disney. Además, su naturaleza transmedia en diferentes formatos, así como en la venta de *merchandising* asociado a la propiedad intelectual, explica los mayores ingresos que generaban como producto de consumo (Nobuyuki, 1968)

En Japón, el fenómeno nació en formato cinematográfico de larga duración lanzado en cines y en formato televisivo. Comenzó tras la Segunda Guerra Mundial con **Toei Studios como principal clúster del entretenimiento**, conformando una **división centrada en la animación llamada Toei Animation** modelada como “la Disney de Oriente”. Trataban usar técnicas de animación modernas para películas de alto presupuesto, basándose en su propia idiosincrasia cultural (Miyao, 2002).

Es en este contexto donde el creador del *manga Astroboy* (Tezuka, 1963) fundó su propio estudio de animación llamado Mushi Production Studio para adaptar su obra, ajeno a los intentos de Disney y de la Toei.

El estreno de esta serie es la primera muestra en televisión del *anime*, lanzado en Japón desde el 1963 hasta el 1966 de forma semanal con episodios de 30 minutos tomando *frames* del *manga* original, en contraposición a las producciones (bi)anuales cinematográficas. **Lo más relevante de este lanzamiento fue la creación de *merchandising* con la venta de productos como extensión de la franquicia** (Fusanosuke, 1992)

Su creador llamado Osamu Tezuka acabó vendiendo el estudio de animación a la cadena de televisión pública nacional japonesa, por una cantidad menor de lo que costaba producir cada capítulo. A cambio, el creador del *manga* se llevaría los *royalties* por la venta de *merchandising*, que generaba beneficios mayores y constantes, mientras continuaba con la serialización del *manga*.

No hay fuentes veraces sobre esta negociación, aunque **se estima que se vendió cada capítulo por 550 yenes, cuando el valor de los mismos se estimaron en 2,5 millones de yenes, conociéndose esta jugada como “la maldición Tezuka” que justifican los presupuestos millonarios actuales.**

A continuación, se desglosa el año fiscal 2022 del conglomerado Bandai Namco:

	[Sales of IPs (Groupwide)]		(billion yen)		
	FY2021.3	FY2022.3		FY2023.3	
	Full Year Results	1st Half Results	Full Year Results	1st Half Forecasts	Full Year Forecasts
Aikatsu!	2.0	1.3	1.9	0.5	1.0
Anpanman	8.7	4.2	9.3	4.5	10.0
Ultraman	8.6	6.5	16.8	7.0	18.5
KAMEN RIDER	28.9	13.3	29.5	13.0	30.0
Mobile Suit Gundam	95.0	50.2	101.7	53.0	119.0
Super Sentai	5.2	2.6	5.0	2.5	5.0
DRAGON BALL	127.4	61.2	127.6	66.0	133.5
NARUTO	21.4	11.7	23.0	9.5	19.0
Pretty Cure	6.6	3.1	5.8	3.0	6.0
ONE PIECE	38.0	21.6	44.1	29.0	58.0

*Figures are calculated based on sales before elimination of inter-segment transactions.

	[Sales of IPs (Toys and Hobby Business, Japan)]		(billion yen)		
	FY2021.3	FY2022.3		FY2023.3	
	Full Year Results	1st Half Results	Full Year Results	1st Half Forecasts	Full Year Forecasts
Aikatsu!	1.0	0.7	1.0	0.3	0.5
Anpanman	8.3	3.9	8.7	4.0	9.0
Ultraman	4.9	3.8	8.0	4.5	9.5
KAMEN RIDER	24.3	9.9	22.8	10.0	23.0
Mobile Suit Gundam	41.0	21.8	44.2	25.5	52.0
Super Sentai	4.5	2.2	4.4	2.5	5.0
DRAGON BALL	15.4	10.1	19.7	11.0	21.0
Pretty Cure	6.6	3.0	5.7	3.0	6.0
Pokémon	6.0	2.6	5.9	2.5	6.0
ONE PIECE	7.1	5.3	10.0	7.5	14.0

*Figures are calculated based on sales before elimination of inter-segment transactions.

Figura 2.1: FY2023 Bandai Namco

Por lo tanto, no puede entenderse esta industria disociándola de la necesidad de vender productos asociados, siendo incluso más importante que la propia animación en las nuevas creaciones modernas.

Según el pensador Maurizio Lazzarato, no importa tanto crear un producto comercial manufacturado como tal, es más importante crear el mundo en el que ese producto existe.

Como negocio lucrativo a la hora de maximizar beneficios se tiene en mente un scope internacional con presupuestos cada vez mayores. El *target* ya no es solo los propios japoneses, sino los occidentales atraídos por este mundo (Lazzarato, 2004)

De aquí, nace la identidad *otaku* asociada a los aficionados.

2.1.2. La etiqueta 'otaku' como marcador cultural

La fenomenología del *otaku* ha dado lugar a varios estudios. Por ejemplo, como definición preliminar se podría tomar un *otaku* es una persona mayormente masculina, cuya vida gira en torno a la colección, producción y creación de contenido asociado al mundo del *manga* y del *anime*, extensible en cierta medida a los videojuegos.

Aunque en un principio fuese una comunidad cerrada de marginados sociales, su impulso ha generado una industria de billones de dólares, rastreando su origen dividiéndose en tres generaciones a lo largo del tiempo. (Azuma, 2009)

Partiendo del éxito de *Astroboy* (Tezuka, 1963), se quería crear en sus origenes narrativas adultas inspiradas por la ciencia ficción. Esta generación de creadores estaban atravesados por la conmoción política al calor de las manifestaciones estudiantiles inspiradas por el Mayo del 68 francés.

Las manifestaciones estudiantiles fueron de tal violencia que finalmente forzaron el cierre de los campus universitarios en todo Japón, usando cascos de seguridad como símbolo de las luchas.

En toda esta turbulenta época, los artistas que fueron más activos durante este periodo se vieron incapaces de trasladar sus ideas a la industria cinematográfica convencional, por lo que trataron de transmitirlas en un formato aparentemente más infantil mediante la animación.

De aquí viene el impacto de *Mobile Suit Gundam*, como una historia épica de ciencia ficción que trataba sobre las consecuencias de la guerra, emitida en horario infantil de forma semanal durante los años 70 y lanzada en forma de trilogía en los cines para competir contra el éxito de *Star Wars*.

Esta serie llevó más allá la dinámica en la industria del *anime* que inició Osamu Tezuka, ya que los robots que aparecían en la misma acabaron siendo unos populares juguetes montados en forma de maquetas.

De ahí, que se conociera como la “maldición Tezuka” a la mercantilización de una serie cuyo mensaje moral era el pacifismo, acabando por priorizar la violencia en forma de acción para crear otros modelos de robots que vender, invisibilizando el mensaje inicial.

Actualmente, la mitad de esta franquicia estimada en cientos de millones de yenes obtiene la mitad de sus ganancias con la venta de este producto, por lo que aquí tenemos un germen del *otaku* como entusiasmo ante estas manifestaciones culturales.

Con todo, la percepción de estos “frikis” era de exclusión de las normas convencionales hasta que empezaron a convocar la convención Otakon, como un Salón Manga primigenio desde 1994, teniendo Akihabara como punto de encuentro.

Estos eventos presenciales fueron consolidando un *branding* de productos que tenían como *target* estos consumidores, al mismo tiempo que servía de reclamo turístico en el hacia fuera. En esta época, se pudieron cohesionar comunidades nicho

mediante primitivos foros de internet, en forma de *imageboard* centrados en imágenes con comentarios asociados.

Sin embargo, la representación de los mismos no era positiva debido a la mentalidad colectivista japonesa que menospreciaba la individualidad de estos aficionados, siendo para ellos no solo un divertimento, sino una reivindicación social.

Debido a conflictos a principios de los 2000 se acabó privando a Akihabara, un distrito de Tokio donde se reunía esta comunidad, al derecho de reunión durante varios años con distintas restricciones policiales hasta 2011.

Esto hizo que su afición se volcase aún más en Internet, replegándose en sus casas, que fue cuando empezó a surgir la figura del *hikikomori*: ↴

Se denominan *hikikomori* a aquellos jóvenes japoneses que se retiran de la sociedad y se aíslan completamente en sus hogares durante un largo período de tiempo, evitando cualquier tipo de interacción social durante meses, años, o incluso décadas.

Aunque se han observado casos de introversión semejantes a la depresión en otros países, especialmente en un contexto (post)pandémico de por medio, esto toma matices específicos en Japón. La presión social en lo académico y en lo profesional asociada a cambios socioeconómicos bruscos, les impide desarrollar una vida adulta con normalidad.

Sin embargo, un *hikikomori* no tiene porque ser *otaku*, ni un *NEET* (*Not in Education, Employment or Training*) aunque se usara este “hombre de paja” para demonizar al colectivo.

2.1.3. La industria del anime en la actualidad

A pesar de la percepción negativa dentro del país, hubo una resignificación internacional de la identidad *otaku*, ya que el mundo del *otaku*

era parte del poder blando del gobierno como activo económico a escala mundial que exportar al extranjero.

Es en este contexto, donde surge el “**Cool Japan**” como estrategia del **gobierno a principios de los años 2000** que usar como parte de los esfuerzos del **gobierno japonés** para promover las producciones culturales propias con la que estimular su economía. Para ello, cooptarían el *anime*, el *manga*, los videojuegos, la moda, la música y la comida.

El objetivo era capitalizar el interés global en la cultura popular japonesa y utilizarlo como una ventaja competitiva en el mercado internacional, dinamizando una industria multimillonaria que era utilizada al mismo tiempo como herramienta de diplomacia cultural.

Después de la Segunda Guerra Mundial, Japón se encontraba en ruinas y enfrentaba una situación económica y social devastada. Por ello, en las décadas de 1960 y 1970, Japón experimentó un boom económico sin precedentes tras el periodo de posguerra, centrándose en la tecnología y en la automovilística.

Hasta que llegó el crash de los años 90, cuando la economía japonesa sufrió una fuerte recesión después de un largo período de especulación y burbujas financieras que condujo a una década de bajo crecimiento económico, alta tasa de desempleo y deflación.

Se implementaron medidas como la flexibilización monetaria, la reestructuración empresarial y las reformas estructurales. Aunque hubo algunos avances, la economía japonesa ha enfrentado desafíos persistentes, como el envejecimiento de la población, la baja tasa de natalidad y la competencia global.

Si atendemos a esta nueva identidad comercial de carácter internacional, quien se considere *otaku* tiene códigos compartidos que no se circunscriben a la idiosincrasia

propiamente japonesa, por lo que se podría considerar que vivimos en una nueva generación *otaku*.

Por lo tanto, la creación del *anime* y el *manga* mueve unos costes de producción millonarios, teniendo en cuenta un scope internacional desde el principio.

Al hilo de esto, Toshio Okada como co-fundador de uno de los estudios más relevantes de la animación japonesa (GAINAX) cuenta una entrada de su video-blog personal (OKADA) lo siguiente sobre las dinámicas en los comités de producción:

- A la hora de financiar proyectos, se establece los nuevos lanzamientos como una propiedad intelectual, con centenares de empresas como inversores que ejercen control creativo sin ser parte de un estudio de animación. Desde discográficas a empresas de juguetes, pasando por las propias cadenas que retransmitirán estas obras, se financian estas series como si de un activo bursátil se tratara.

De este modo, se forman conglomerados donde se entrelazan estas empresas que crean estos productos de manera manufacturada con presupuestos oscilando entre 20 a 30 millones de dólares.

- Estos números que se manejan pueden ser de un 60-70 % proveniente de capital extranjero, con el que reducir su propio riesgo si la serie no obtiene los resultados deseados.

Se vuelve necesario dividir la propiedad intelectual en diferentes activos más allá de la animación en sí misma por parte de conglomerados solventes, habiendo cada vez más oferta para los potenciales consumidores.

- Aunque en ocasiones se pueda ahogar económica a los propios estudios en cuanto a control creativo según el reputado animador, se necesita por este mismo riesgo que las empresas no compitan entre sí, estableciendo oligopolios en forma de conglomerados donde se reparten responsabilidades y se ponen límites a esta financiación.

2.1.4. Relevancia de nuestra solución en este contexto

Esta dependencia a los productos asociados en la venta de *merchandising* viene también causado por el consumo pirata que se hace del *fansub*, que no hace parametrizable las métricas de éxito para los estudios o el público objetivo a nivel cuantitativo.

En la época de principios del nuevo milenio, los *fansubs* contribuyeron a la globalización del *anime* fuera de Japón, mediante el esfuerzo colectivo de los aficionados que subtitulaban y distribuían este contenido por Internet sin ánimo de lucro, mientras que adelantaban y asentaban tendencias.



Figura 2.2: Los fansub añaden karaoke o el nombre de los contribuidores

Debido a esto, series que se consideran un éxito en cuanto a popularidad no cumplen con los objetivos marcados por los creadores a nivel de consumo, como es el caso de una nueva serie que goza de una gran popularidad llamada *Chainsaw Man* (Seko, 2022) la cual no ha vendido copias físicas de DVD/BluRay en Japón, por lo que incluso se duda sobre invertir en su localización internacional (Villa, 2023)

Como respuesta a esta situación, los estudios pactan de antemano el lanzamiento en plataformas de *streaming* de forma exclusiva, recibiendo dinero por ello para la financiación de los propios proyectos.

Crunchyroll es el servicio de *streaming* especializado en anime más consumido del mundo, habiendo sido creado por cuatro estudiantes de ingeniería en California durante el año 2006. Aunque comenzase como portal de contenido pirata que monetizaban ante el descontento de la comunidad y de los propios estudios de animación, tuvo una inversión en 2008 por parte del fondo de inversión Venrock rondando los 4 millones de dólares (Sevakis, 2008)

Fue en 2013 cuando recibió otra financiación de 100 millones por parte de News Corp, mientras que AT&T se uniría 1 año después con 500 millones. Tras esta ronda de inversores, resulta sintomático que Sony como monopolio comprase tanto **Crunchyroll** (Kun Gao, 2023) como la plataforma de streaming estadounidense llamada **Funimation** que hacía de competencia, mientras invierte en la creación de series que serán publicadas en las mismas.

Ante esta situación, **Netflix** no solo compra exclusividades de series de *anime* para seguir en el mercado, sino que también produce sus propias series originales (Reed Hastings, 2023)

Se podría vincular el éxito de la implementación del *anime* en nuestro país con las televisiones autonómicas emitiendo series de animación para un público infantil/juvenil, que fueron perdiendo espectadores conforme se ha dejado de consumir televisión de forma masiva en detrimento a un contenido en Internet que podían escoger los usuarios, junto a las recientes formas de consumo basadas en plataformas de contenido.

Con esta creciente accesibilidad y la costumbre de ver *anime* con subtítulos debido al consumo mediante **fansubs**, las distribuidoras nacionales que en el pasado localizaban estas series al castellano y otras lenguas oficiales como el catalán, no podían competir ante este ritmo de lanzamientos por temporadas ni tenían formas de averiguar qué series merecían la pena en términos comerciales.

Las principales distribuidoras españolas *Selecta Visión y Jonu Media* (GROUP, 2023; S.L.U., 2023) han creado sus propios servicios de *streaming* con contenido doblado, junto a lanzamientos físicos y los *simulcasts*.

Este concepto remite a la emisión de series de forma simultánea en cuanto se hayan lanzado en Japón, intentando crear un evento como impacto para ser vivido colectivamente por los aficionados de manera internacional.

Sin embargo, estas series no parecen generar el mismo interés que las ya pactadas por las grandes distribuidoras internacionales, que tienen ya garantizadas productos de éxito sin necesidad de doblaje.

A pesar de la iniciativa, no se ha conseguido dinamizar suficiente el visionado hacia plataformas de *streaming* nacionales especializadas en *anime*, debido a que puede hacerse de manera “ilegal” y no hay tanta inmediatez asociada al visionado de *anime* que no tenga una *fanbase* previa o expectación suficiente. Además, el tipo de consumo suele hacerse en modo “*binge-watching*” con maratones donde se prefiere ver la serie de una sentada una vez ha sido completada.

Por tanto, existe una altísima oferta en cuanto a contenido necesitando bases de datos como *MyAnimeList* (Gyssler, 2023) para que los aficionados puedan seleccionar qué series son relevantes en cuanto a popularidad, llevando un seguimiento de las series vistas o aquellas pendientes. Sin embargo, no son capaces de recoger nuevos matices:

- La división clásica de animes se establecía por demografías, dependiendo de edad y género, teniendo categorías como *shounen*, *shoujo* o *seinen*.
- El consumo de anime es cada vez más transversal en cuanto a su target, que apela más a cualquier persona que se autodenomine *otaku*, y esta clasificación no es lo suficientemente flexible para recoger subgéneros recientes como *isekai*, *Slice of life* o *moe*.

Con todo lo expuesto, se investigará acerca de una posible solución para cubrir los problemas de *reach* y *engagement* de las distribuidoras españolas ante esta compleja situación, mediante los algoritmos que se presentarán en los siguientes apartados.

2.2. Estado del Arte

Al comenzar con la **introducción**, se han identificado los **problemas que existen en la industria del anime como producto** en las plataformas de *streaming* y se han introducido las propuestas para solucionar dichos problemas. Más adelante, en el **contexto**, se han analizado las causas desde lo **cultural**, y de lo **económico**.

Partiendo de estos antecedentes, se consideran las siguientes premisas:

- Las recomendaciones de series se centran en criterios clásicos basados en **demografía**, en lugar de las necesidades del usuario.
- Las distribuidoras realizan grandes inversiones para localizar productos audiovisuales sin que se pueda parametrizar previamente el **impacto** de los *animés* que salen cada temporada.

A continuación, se abordará el **Estado del Arte para cada propuesta de desarrollo** a modo de solución tecnológica. Para ello, se ha trazado el realizado un diagrama del planteamiento inicial del trabajo que dará pie a la investigación:

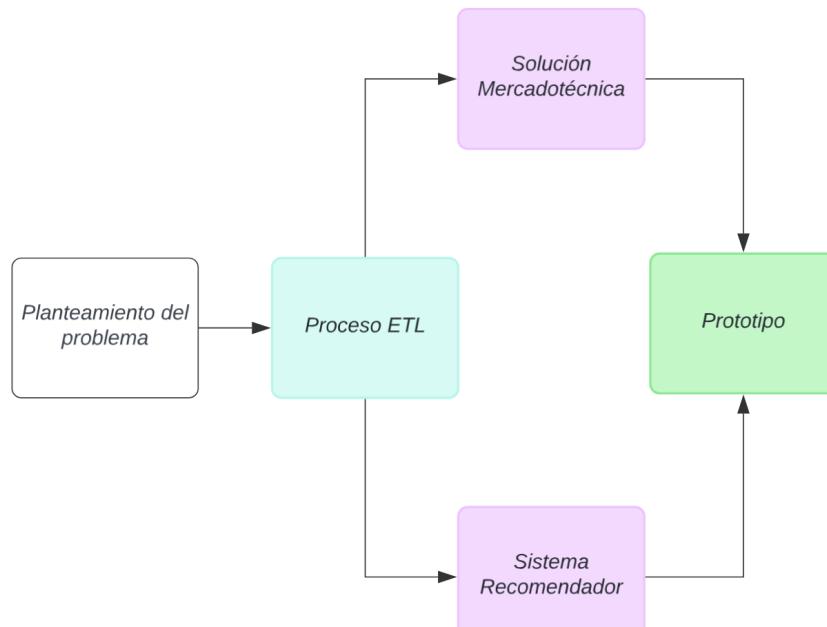


Figura 2.3: Planteamiento del Proyecto

2.2.1. Implicación en comunidades de IA

Gracias a la participación de ambos autores de esta memoria como investigadores, en **comunidades de Inteligencia Artificial** fuera del ámbito académico y/o laboral, ha sido posible la **búsqueda de conocimiento para conocer el Estado del Arte, al mismo tiempo que se ha obtenido feedback en los distintos planteamientos de potenciales algoritmos a lo largo del desarrollo del proyecto.**



Figura 2.4: Spain AI («Spain-AI», 2023)

El apoyo comenzó gracias al descubrimiento de la comunidad española llamada **Spain-AI** (Figura 2.4). cuya intención es la de **construir networking y realizar webinars** acerca de todo lo relacionado con la Inteligencia Artificial.

Una vez incorporados ambos en la comunidad, se asistió a dichas sesiones para aprender acerca de los conceptos que más adelante se explorarán.

Por último, cada uno de los integrantes del equipo de investigación forma parte de la subcomunidad de Spain-AI (Figura 2.4) correspondiente a su ciudad de origen, en este caso Málaga y Cádiz.



Una sesión del grupo de estudio de Málaga AI. Cédida
TECNOLOGÍA
Málaga AI, la comuna de aplicaciones de la inteligencia artificial: "Uno más uno sale más que dos"
Este grupo de estudio sin ánimo de lucro apuesta por la proactividad ante la última gran revolución tecnológica y busca patrocinadores.



(a) Malaga AI en el periódico El Español

(b) Cadiz AI en la cadena de TV Onda Cádiz

Figura 2.5: Repercusión mediática de las comunidades (Cádiz, 2023; Peláez, 2023)

Como se observa en la Figura 2.5, se **crearon grupos de estudios para diversos proyectos en las correspondientes ciudades**. Dichos grupos prestaron ayuda al desarrollo de esta investigación, aportando en la elaboración y planteamiento del problema.

2.2.2. Premisa (I)

Comenzando por la primera premisa “**Las recomendaciones de series se centran en criterios clásicos basados en demografía, en lugar de las necesidades del usuario**”, se va a explicar qué se entiende como *Sistema Recomendador*, qué sistemas existen en el mercado actual y, más específicamente, qué sistemas se usan en las plataformas de streaming

Posteriormente a la explicación de la Premisa II, **se van a validar las premisas planteadas** gracias al estado del arte investigado.

Además, también se han validado dichas premisas mediante el **desarrollo de una encuesta acerca del consumo de anime realizada al público objetivo del proyecto**. Dicha encuesta, tanto en las conclusiones de los desarrollos como en la conclusión final, es explicada en detalle.

Cuando se planteó cómo lidiar con el problema asociado a la validación de la premisa desde cero, se partió **de resolver algunas cuestiones fundamentales** como punto inicial:

- *¿Qué son los sistemas recomendadores?*
- *¿Qué sistemas recomendadores hay en la industria?*
- *¿Qué sistemas recomendadores se usan en los servicios de streaming?*

Para responder a estas, se utilizó como **fuente de investigación una sesión webinar** organizada por Spain AI («Spain-AI», 2023)) a modo de seminario online (Figura 2.6) donde Miguel González-Fierro (Principal Data Science Microsoft) enseña de forma interactiva **qué son los sistemas recomendadores, cuáles son los tipos y algunos ejemplos**.

En esta charla, se da a conocer los principales tipos de *sistemas recomendadores* junto con varios ejemplos de empresas que los emplean, aunque se dieron por sentado algunas cuestiones fundamentales como:



Figura 2.6: Creando Sistemas de Recomendación desde cero (González-Fierro, 2023)

1. ¿Qué es un *Sistema Recomendador*?
2. ¿Qué aplicaciones tiene?
3. ¿Qué beneficios conlleva usarlos?

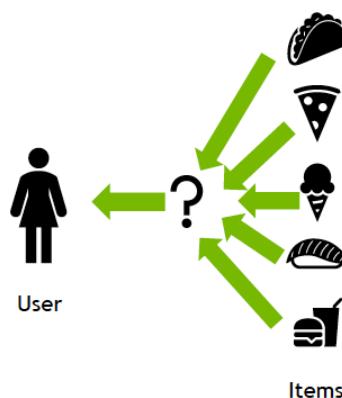


Figura 2.7: Planteamiento de la técnica (Nvidia, 2024)

A.- Cuestiones acerca de los Sistemas Recomendadores

Antes de comenzar con la explicación del *webinar*, necesitamos responder a los aspectos anteriores. Para responder a la **primera pregunta**, se parten de las siguientes definiciones:

- **Nvidia:** “Un sistema de recomendación (o *Sistema Recomendador*) es una clase de aprendizaje automático que utiliza datos para ayudar a predecir, acotar y encontrar lo que la gente busca entre un número exponencialmente creciente de opciones.” (Nvidia, 2024)

En la misma fuente, se describe que el **objetivo de estos sistemas de recomendación es comprender las preferencias, decisiones previas y características de las personas conectando con sus productos** mediante los datos recopilados sobre sus interacciones.

- **Khang Pham:** “Los sistemas de recomendación, o *sistemas recomendadores*, son motores de software diseñados para sugerir artículos a los usuarios en función de lo que les haya gustado o no anteriormente, la participación e interacción con el producto, etc. Los sistemas de recomendación mantienen a los usuarios interesados en lo que el sitio sigue recomendando”

Los motores de recomendación **proporcionan una experiencia de usuario personalizada, ayudando a cada consumidor a identificar y descubrir sus películas, programas de televisión, productos digitales, libros, artículos, servicios y demás favoritos.**” (Pham, 2022)

En resumen, se define *Sistema Recomendador* como una técnica que, mediante técnicas basadas en Inteligencia Artificial, **identifica aquello que los consumidores prefieren** basándose en el **comportamiento sobre los contenidos** que se consumen y/o **comportamiento similar a otros usuarios** con gustos similares.

Por último, se responderá a la cuestión sobre la aplicación de las ventajas que ofrecen estos sistemas:

- **Nvidia:** “Estos datos incluyen **interacciones en forma de reviews, clicks o contenido anterior que se haya consumido**. Gracias a su capacidad para predecir los intereses y deseos de los consumidores de forma altamente personalizada, los **sistemas recomendadores** son los favoritos de los proveedores de contenidos y productos, pudiendo **dirigir a los consumidores hacia cualquier producto o servicio que sea relevante** para los grupos mayoristas.” (Nvidia, 2024)
- **Khang Pham** (Pham, 2022) se enumeran los beneficios del uso de sistemas de recomendación según el autor:
 1. **Aumento de las ventas:** Para las empresas, el hecho de tener **sistemas recomendadores** implementados para vender sus productos les conviene, por lo tanto, necesitan invertir capital para generar más ingresos. Gracias a este tipo de técnicas, **aumenta el compromiso del consumidor en su sitio y capta tiempos de sesión más largos**, resolviendo al mismo tiempo problemas de *reach* y de *engagement* propios de la economía de la atención.
 2. **Menor carga del sistema:** Dichos sistemas **ayudan en la eficiencia de las ventas**, ya que recomiendan los productos adecuados a cada usuario mejorando las ventas, al mismo tiempo que mantienen una menor carga en el sistema y disminuyen los costes a largo plazo.
 3. **Mayor compromiso y satisfacción:** Al **asignar productos relevantes** para los usuarios, **los consumidores seguirán interactuando con el portal**, aumentando así su nivel de satisfacción.

- **Dynabyte** (dynabyte, 2024) se enumeran distintas aplicaciones:

1. **Comercio Electrónico**: El portal web de Amazon se nutre de los históricos de compra/navegación, incluyendo las valoraciones de los usuarios.
2. **Servicios de Entretenimiento**: Plataformas como Netflix o Spotify hacen que el usuario siga consumiendo recursos en las plataformas.
3. **Redes Sociales**: Facebook y Twitter usan estas técnicas para personalizar las *feeds* de los usuarios priorizando el primer contenido que verán, amplificando su contenido deseado.
4. **Medios digitales**: Estos sistemas ayudan a sugerir artículos y noticias que se ajustan a los intereses del lector.
5. **Turismo**: Estos servicios ofrecen ofertas de hoteles, vuelos y paquetes de viaje basados en las preferencias del usuario y sus reservas anteriores.

Además, se recalca una serie de ventajas a la hora de tener un sistema de este tipo, entre las que se destaca la experiencia personalizada del usuario con la cual se ayuda al consumidor a descubrir nuevos productos.

B.- Webinar: “Creando Sistemas de Recomendación desde cero”

Al hilo de lo comentado anteriormente en la explicación sobre los *Sistema Recomendador*, estos sistemas se dividen en dos tipos:

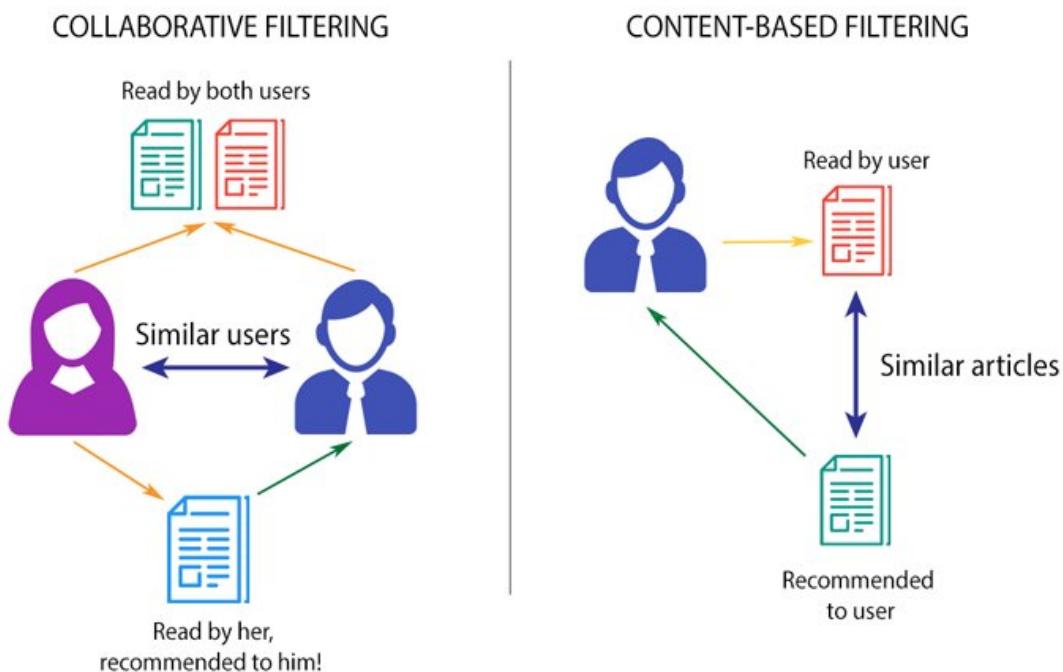


Figura 2.8: Tipos de Sistemas Recomendadores (González-Fierro, 2023)

- **Collaborative-based/behavior-based:** Los *sistemas recomendadores* de tipo colaborativo basan su funcionamiento en el comportamiento de los usuarios. En el primer caso de uso en la figura 2.8, se muestran como dos consumidores dentro del sistema se reconocen como usuarios similares, recomendando contenido similar.
- **Content-based:** Los *sistemas recomendadores* de tipo contenido basan su funcionamiento en el propio producto artículo. En el segundo caso de uso en la figura 2.8, se muestran como dos artículos son catalogados como similares, y cuando uno de los artículos es leído/consumido por cualquier usuario se cataloga el artículo como similar recomendándose a dicho usuario.

Tras la explicación de los tipos, se define la existencia de tres componentes básicos en dichos sistemas: El **usuario**, el **objeto** y la **interacción entre ambos** conforman los **filtros de información que usan aprendizaje automático**.

Estas **interacciones entre los componentes** se dan dependiendo de la forma que la aplicación recopila la información del usuario:

- **Explícita:** Cuando un usuario valora directamente un producto mediante una cantidad entre 1 a 5 estrellas de manera que **el usuario ofrece feedback de manera directa** a la propia aplicación.
- **Implícita:** Cuando un usuario visualiza un vídeo de **TikTok** (ByteDance, s.f.), y **la propia aplicación captura el tiempo de retención** para saber si le ha interesado el vídeo.

Además de la explicación de los tipos y los componentes, se hace hincapié en que **el 35 % de las ventas de** la plataforma de comercio más grande a nivel mundial como **Amazon**, se da gracias a **ofrecer productos mediante un Sistema Recomendador**.

Por último, se profundizan en los conceptos de ambos tipos de sistemas con algunos ejemplos:

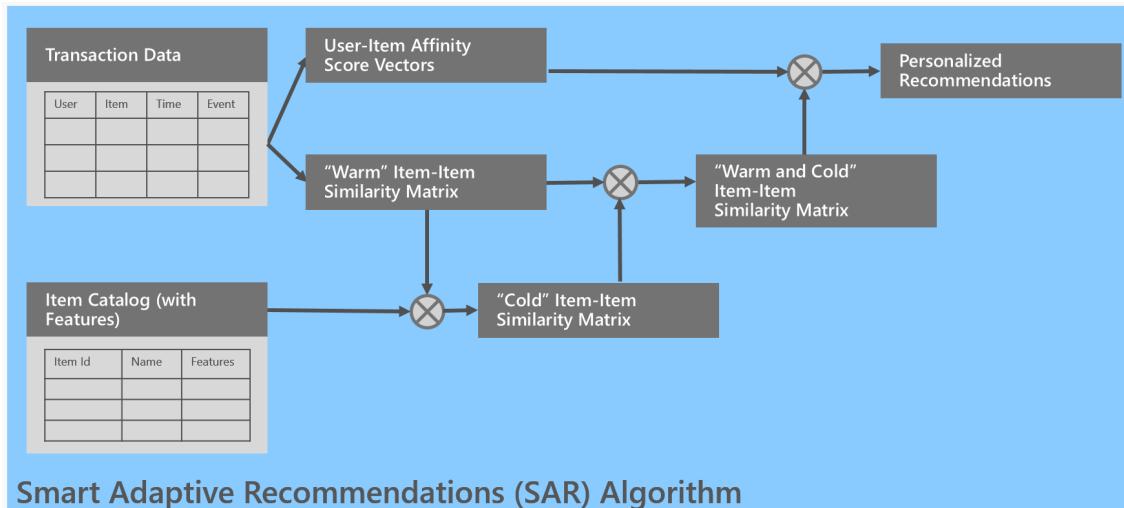


Figura 2.9: Diagrama de funcionamiento del algoritmo SAR (Microsoft, 2023)

Un **Algoritmo SAR (Smart Adaptive Recommendation)** es un **ejemplo de tipo colaborativo desarrollado por Microsoft** (Microsoft, 2023), como solución a un **Sistema Recomendador** modulado por el comportamiento de los usuarios.

Como aparece en la figura 2.9, se parten de **dos conjuntos de Datos** llamados ***Transaction Data* y *Item Catalog***.

En el **primero**, se almacena el identificador de usuario junto con el identificador del artículo correspondiente, el momento y tipo de la interacción que el usuario ha realizado con el artículo.

Por ejemplo, a la **tupla de cuatro elementos** $\langle User_1, Item_1, 2015/06/20T10 : 00 : 00, Click \rangle$ se le llama transacción. En el **segundo**, se almacena la **información de los artículos**.

Partiendo de esos datos, se obtienen las matrices de similitud entre:

$$\begin{bmatrix} & Item_1 & Item_2 & \dots & Item_j \\ Item_1 & Coccus(User_1, Item_1) & Coccus(User_1, Item_2) & \dots & Coccus(User_1, Item_j) \\ Item_2 & Coccus(User_2, Item_1) & Coccus(User_2, Item_2) & \dots & Coccus(User_2, Item_j) \\ \dots & \dots & \dots & \dots & \dots \\ Item_i & Coccus(User_i, Item_1) & Coccus(User_i, Item_2) & \dots & Coccus(User_i, Item_j) \end{bmatrix} \quad (2.1)$$

1. **Objeto-Objeto:** Dicha matriz simétrica muestra, para cada par de elementos, una medida de similitud basada en el **concepto de Co-Occurrence** que es el **número de veces que aparecen juntos dos objetos en una transacción**. Además, para el algoritmo se utilizan dos medidas más que sirven para favorecer aquellas relaciones entre dos objetos en las cuales los objetos han aparecido pocas veces:

Lift measure

$$Lift(Item_i, Item_j) = Coccus(User_i, Item_j) / (occ(Item_i) * occ(Item_j))$$

Jaccard measure

$$Jaccard(Item_i, Item_j) = Coccus(User_i, Item_j) / (occ(Item_i) + occ(Item_j) - Coccus(User_i, Item_j))$$

La diferencia entre ambas medidas es que *Lift* favorece mucho más a aquellas con poco número de ocurrencias de I y J.

$$\begin{bmatrix} & Item_1 & Item_2 & \dots & Item_j \\ User_1 & Aff(User_1, Item_1) & Aff(User_1, Item_2) & \dots & Aff(User_1, Item_j) \\ User_2 & Aff(User_2, Item_1) & Aff(User_2, Item_2) & \dots & Aff(User_2, Item_j) \\ \dots & \dots & \dots & \dots & \dots \\ User_i & Aff(User_i, Item_1) & Aff(User_i, Item_2) & \dots & Aff(User_i, Item_j) \end{bmatrix} \quad (2.2)$$

- 2. Usuario-Objeto:** Dicha matriz **contiene por cada par** $\langle User, Objeto \rangle$ una medida denominada *Affinity Score*. Esta función devuelve una **puntuación correspondiente al usuario con el objeto que depende del número de transacciones en las que aparecen juntos**, lo nuevas que sean esas transacciones e incluso el tipo de evento de la transacción.

$$\begin{bmatrix} & User_x \\ Item_1 & rec(User_x, Item_1) \\ Item_2 & rec(User_x, Item_2) \\ \dots & \dots \\ Item_i & rec(User_x, Item_i) \end{bmatrix} \quad (2.3)$$

Haciendo una serie de operaciones matemáticas podríamos llegar a una Matriz Columna Final con las recomendaciones, para cada usuario, de todas los objetos recomendados.

Un **Algoritmo GBDT** (*Gradient Boosting Decision Tree*) es un **ejemplo de tipo basado en contenido**. En este caso, no se trata de un algoritmo utilizado únicamente en *sistemas recomendadores*, sino que es usado en *Machine Learning*. Esta técnica se basa en utilizar un conjunto de árboles de decisiones, a través de *ensembles* para poder realizar la recomendación apropiada (Maklin, 2023).

Se menciona que los *Frameworks* como XGBoost (*Extreme Gradient Boosting*) y LightGBM (*Light-Gradient Boosting Machine*) son los más populares.

Aunque el objetivo del proyecto no se centra en el desarrollo a nivel matemático de dichos árboles, se remarcará alguna diferencia, con respecto a los originales:

1. **XGBoost:** Cada árbol subsiguiente aprende a partir de los árboles anteriores y no tiene asignado el mismo peso, a diferencia de cómo funciona los *Random Forest*.
2. **Ambos:** Utilizan la técnica de *Gradient Boosting* la cual construye modelos iterativamente que van mejorando a los anteriores.

C.- Sistemas Recomendadores en Plataformas de Streaming

Habiendo abordado los conceptos acerca de qué son los *sistemas recomendadores* y qué aplicaciones/beneficios tienen, se ha investigado acerca de los sistemas implementados en las plataformas de *streaming* de series, como pueden ser *Netflix* (Reed Hastings, 2023) o *Crunchyroll* (Kun Gao, 2023).

En el caso de *Netflix* se utiliza varios sistemas al mismo tiempo (Carlos A. Gomez-Uribe, 2023; Netflix, 2023a, 2023b) como los siguientes algoritmos de recomendación:

- **Personalised Video Ranking (PVR):** Se centra en ofrecer una lista de vídeos personalizada para cada usuario. Utiliza información específica del usuario, como su **historial de visualización, preferencias y comportamiento** en el sitio, para clasificar los vídeos de acuerdo con la probabilidad de que el usuario los encuentre relevantes. Este algoritmo tiene en cuenta las preferencias individuales de cada usuario y adapta las recomendaciones en función de su comportamiento previo.
- **Top-N Video Ranker:** Se enfoca en ofrecer una lista de los vídeos más populares o mejor clasificados para todos los usuarios. En lugar de personalizar las recomendaciones para cada usuario, este algoritmo clasifica los vídeos en función de su popularidad general, como la cantidad de visualizaciones, calificaciones o interacciones en el sitio web. Luego, presenta a cada usuario una lista de aquellos mejor clasificados.
- **Trending Now Ranker:** Se centra en recomendar vídeos que están experimentando un aumento significativo en popularidad en un período de tiempo reciente. Identifica los que están en tendencia en el momento y los presenta a los usuarios para aprovechar el interés actual en esos contenidos. La clasificación de los vídeos se realiza en función de factores como la tasa de crecimiento de las visualizaciones o las interacciones en un corto período.
- **Continue Watching Ranker:** Se recomiendan vídeos a los usuarios que han comenzado a ver un vídeo previamente, pero no lo han

terminado. Utiliza información sobre el progreso de visualización de cada usuario para identificar videos que han sido pausados o abandonados. Luego, recomienda esos videos para que el usuario pueda continuar viéndolos desde el punto en el que los dejó.

- **Video-Video Similarity Ranker** —a.k.a. Because you watched (BYW): Este algoritmo **busca videos similares a los que un usuario ha visto previamente.** Utiliza técnicas de análisis de contenido o colaborativas para encontrar videos que tengan características o temas similares a los de los videos previamente vistos por el usuario. Al recomendar videos similares, se espera aumentar la probabilidad de que el usuario encuentre contenido relevante a sus intereses.

2.2.3. Premisa (II)

Una vez considerada la primera premisa se va a continuar con la segunda premisa “**Las distribuidoras realizan grandes inversiones para localizar productos audiovisuales sin que puedan parametrizar previamente el impacto de los animes que salen cada temporada**”

Cuando se pensó acerca de cómo lidiar con el problema asociado a la hipótesis desde cero, se partió de **resolver dos cuestiones fundamentales** a la hora de elaborar alguna propuesta:

1. **¿Cómo saben las distribuidoras qué anime licenciar antes de su lanzamiento?**
2. **¿Existe alguna métrica que pueda parametrizar el impacto del contenido audiovisual?**

A.- Manuel Cobo

Manuel Cobo es un **investigador de la UGR** que hizo un trabajo llamado “*Analysing the conceptual evolution of qualitative marketing research through science*

mapping analysis" (Martín, 2024a) donde **exploraba si el contenido audiovisual con distintas métricas era garantía de éxito una vez lanzado**. El trabajo no está accesible al público, por lo que no se pudo acceder a esa información.

Por otro lado, existe otro trabajo de investigación llamado "*The Simpsons did it: Exploring the film trope space and its large structure*" (Martín, 2024b) donde **analiza correlaciones en forma de tropos**.

Los tropos son **repeticiones temáticas en la narrativa que crean patrones que la audiencia reconoce**. La solución recoge un *dataset* con 10.766 películas con 25.776 tropos asociados a géneros, valoración en relevancia y popularidad.

Ninguno de estos trabajos resultan relevantes con respecto a los objetivos marcados en esta investigación, ya que **no están adaptadas a las necesidades de la comunidad que consume animación japonesa**, donde los tropos narrativos no pueden tomarse de la sinopsis, y donde el mercado es distinto en comparación a la industria audiovisual occidental.

B.- Traductor de Manga

Se contactó con un **traductor de japonés especializado en mangas** quien **no era capaz de vislumbrar que decisiones mercadotécnicas tomaban las editoriales detrás de licenciar sus publicaciones**, las cuales no son de conocimiento público.

Esta información era relevante para la investigación, ya que **existen motivos similares para que las distribuidoras españolas especializadas se decidan sobre localizar contenido audiovisual**.

2.2.4. Validación de las premisas

Gracias a la investigación del Estado del Arte entorno a ambas premisas, se puede validar que realmente existen estos problemas a día de hoy.

Para la **premisa (I)**, se verifican los siguientes puntos:

- **Dynabyte (dynabyte, 2024)** especifica que una de las aplicaciones de los *sistemas recomendadores* es dirigido a las plataformas de streaming.

(Ref: 2.2.2)

- **Khang Pham (Pham, 2022)** dice que los *sistemas recomendadores* mantienen al usuario una experiencia personalizada lo que conlleva a que el usuario siga interesado en consumir, a modo de *engagement*.

(Ref: 2.2.2)

- **Miguel González-Fierro (González-Fierro, 2023)** confirma nuestra introducción y contexto entorno a la importancia a nivel mercadotécnico en la industria de tener un buen sistema recomendador con el ejemplo del porcentaje de éxito de Amazon.

(Ref: 2.2.2)

- **Netflix (Reed Hastings, 2023)** con la serie Neon Genesis Evangelion (Anno, 1995) confirma que ni la clasificación clásica en demografía ni las etiquetas arbitrarias “De suspense” / “Emocionante” / “Cyberpunk” son suficientes para recomendar contenido similar que sea relevante.

Para la **premisa (II)**, no se ha podido verificar ya que no se ha conseguido información acerca del análisis de mercado que realizan dichas plataformas, así que se dará por sentado que el comportamiento en este aspecto es similar a las empresas de streaming competidoras.

2.2.5. Solución Planteada

Partiendo del diagrama 2.3 introducido al comienzo del epígrafe de Estado del Arte, se va a explorar, por separado, cada una de las partes de la solución planteada como diferentes ámbitos a investigar para poder evaluar el posible alcance para el proyecto.

1. Proceso ETL
2. Sistema Recomendador
3. Sistema Predictivo

En pos de no reiterar en secciones posteriores, se menciona que *GitHub* será utilizado como repositorio en línea para el control de versiones para cada uno de los avances en el desarrollo.

1.- Proceso ETL

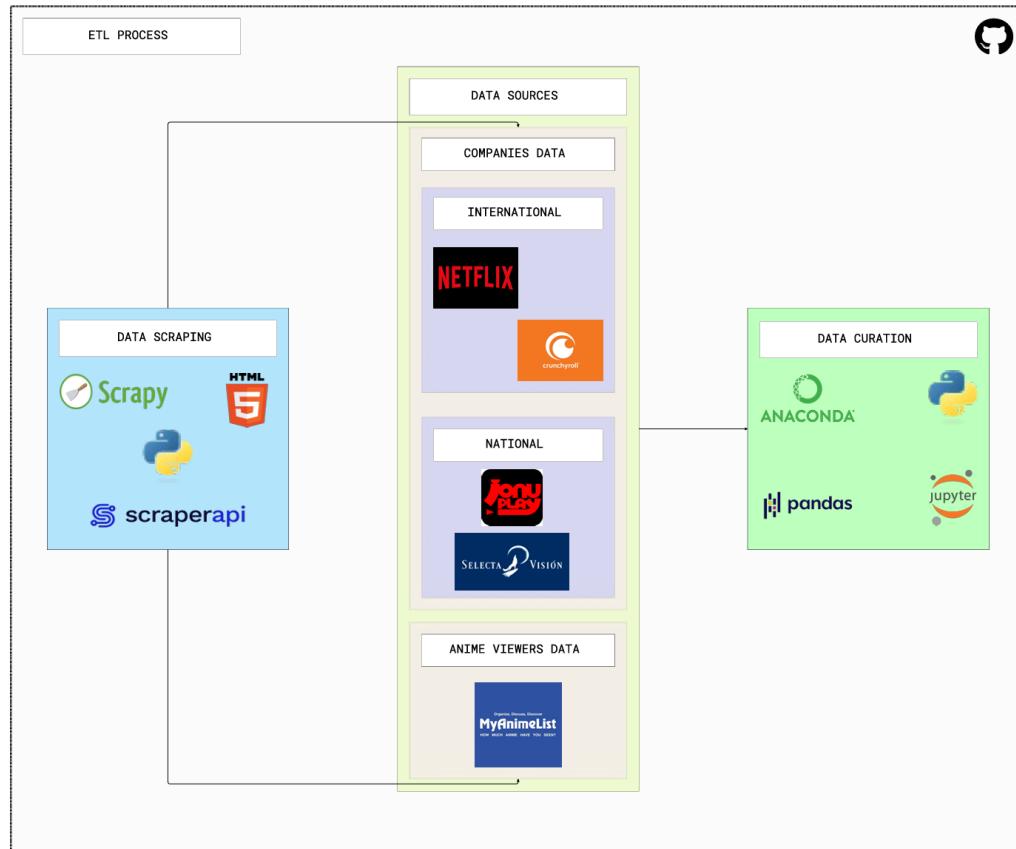


Figura 2.10: Diagrama del Proceso ETL («Cloud Skew web», 2023)

La primera etapa del desarrollo se centra en la **obtención de los datos** para el posterior uso de ellos. Se ha investigado acerca del **proceso ETL (Extract, Transform and Load)** que consiste en **extraer datos de distintas fuentes** en internet mediante *Webscraping* y así, **convertirlos en un formato adecuado para un proceso computacional** concreto.

Este proceso, como se observa en la Figura 2.10, consiste en los siguientes puntos:

1. *Data Scraping*
2. *Data Sources*
3. *Data Curation*

1.- *Data Scraping*

En esta sección se ha investigado la librería *Scrapy* de *Python* para realizar el *Webscraping* de los datos.

2.- *Data Sources*

La idea principal es realizar distintas llamadas a los siguientes *end-points* :

- ***Anime Viewers Data***: Mediante *MyAnimeList* (Gyssler, 2023), una página web gestionada por usuarios que contiene una gran base de datos de *anime* utilizándose como apoyo para crear un *dataset* con el que desarrollar la solución.
- ***Companies Data***: Se trata de una serie de plataformas de *streaming* de nacionales como *Jonu Media* (GROUP, 2023) y *Selecta Visión* (S.L.U., 2023), o internacionales como *Netflix* (Reed Hastings, 2023) y *Crunchyroll* (Kun Gao, 2023) las cuales se utilizan como referencia para saber qué contenido audiovisual será de impacto.

Se encontraron varias dificultades a la hora de empezar de este proceso:

- **Cantidad y Manejo de Peticiones a las páginas**: Actualmente, las páginas web tienen controladas el ancho de banda de los servidores en sus dominios. Esto impide realizar una gran cantidad de peticiones desde una misma dirección IP, por lo que se optó por utilizar un SaaS (*Software as a Service*) denominado *scraperapi* que permitió realizar rotaciones de IPs, llamando a la misma página con direcciones usando IPs distintas similar a proxies.
- **Incapacidad de obtención de datos de plataformas**: Debido al estricto control que hay sobre las plataformas de streaming, no se ha conseguido poder realizar *webscraping* a dichas plataformas, por lo que se ha optado por diferentes vías para obtener los datos como puede ser *Kaggle*, *HuggingFace* o *Wikipedia*.

3.- *Data Curation:*

Tras realizar el *webscraping* de las plataformas mencionadas anteriormente, se construye un *Dataframe* auxiliar con todos los datos recopilados de las distintas fuentes para finalmente construir dos *Dataframes* distintos según las necesidades de cada desarrollo. Para ello, se utiliza *Python* como lenguaje de programación principal. Al mismo tiempo, se usa *Anaconda* como distribución, *Jupyter Notebook* como IDE, y *pandas* como librería principal.

2.- Solución 1: Sistema Recomendador

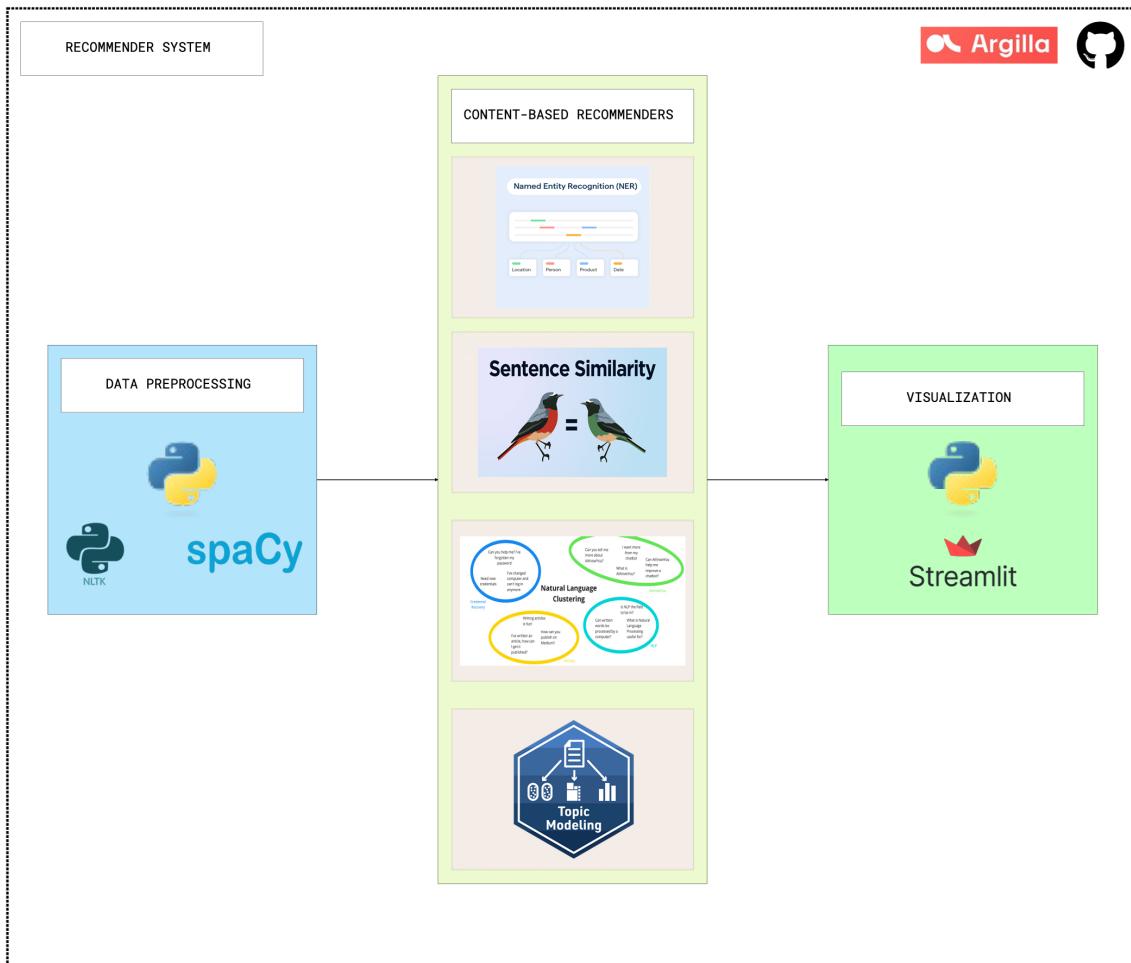


Figura 2.11: Diagrama de la investigación del Sistema Recomendador («Cloud Skew web», 2023)

La segunda etapa del proyecto se centra en la investigación acerca de desarrollar un **Sistema Recomendador** basado en contenido que se centra en recomendar *animés* usando las sinopsis y técnicas de **Procesamiento del Lenguaje Natural**.

1. *Data Preprocessing*
2. *Content-Based Recommenders*
3. *Visualization*

1.- Data Preprocessing

En esta sección se ha investigado las librerías *spaCy* y *NLTK* de *Python*, con el fin de realizar técnicas de pre-procesamiento propias de la lingüística computacional, como pueden ser:

- **Eliminar Stop-words**
- **Tokenización**
- **Lematización**

2.- Content-Based Recommenders

En esta sección se ha investigado posibles técnicas de *Procesamiento del Lenguaje Natural* como pueden ser:

- **Named-Entity Recognition (NER)**: Técnica en la cual se extraerían categorías generales de cada sinopsis y se recomendarían en función a la cantidad de categorías que se compartan entre *animés*.
- **Text Similarity**: Técnica por la cual se compararían *animés* en función de la similitud de las sinopsis.
- **Clusterización**: Técnica muy similar al *Text Similarity*, pero primero realizando agrupaciones de *animés* por géneros/temas, y después realizando la similitud de textos por cada grupo.
- **Topic Modeling**: Técnica consistente en obtener el tono/tema abstracto de cada *anime* para emparejarse entre sí.

3.- Visualization

Para terminar, se ha investigado cuál podría ser el *output* de este sistema. Como punto de partida se ha optado por *Streamlit* para poder mostrar el funcionamiento de cada técnica, pero se ha planteado otra posibilidad como desarrollar una API que reciba solicitudes de recomendación de series y las responda según el algoritmo seleccionado.

3.- Solución 2: Sistema Predictivo

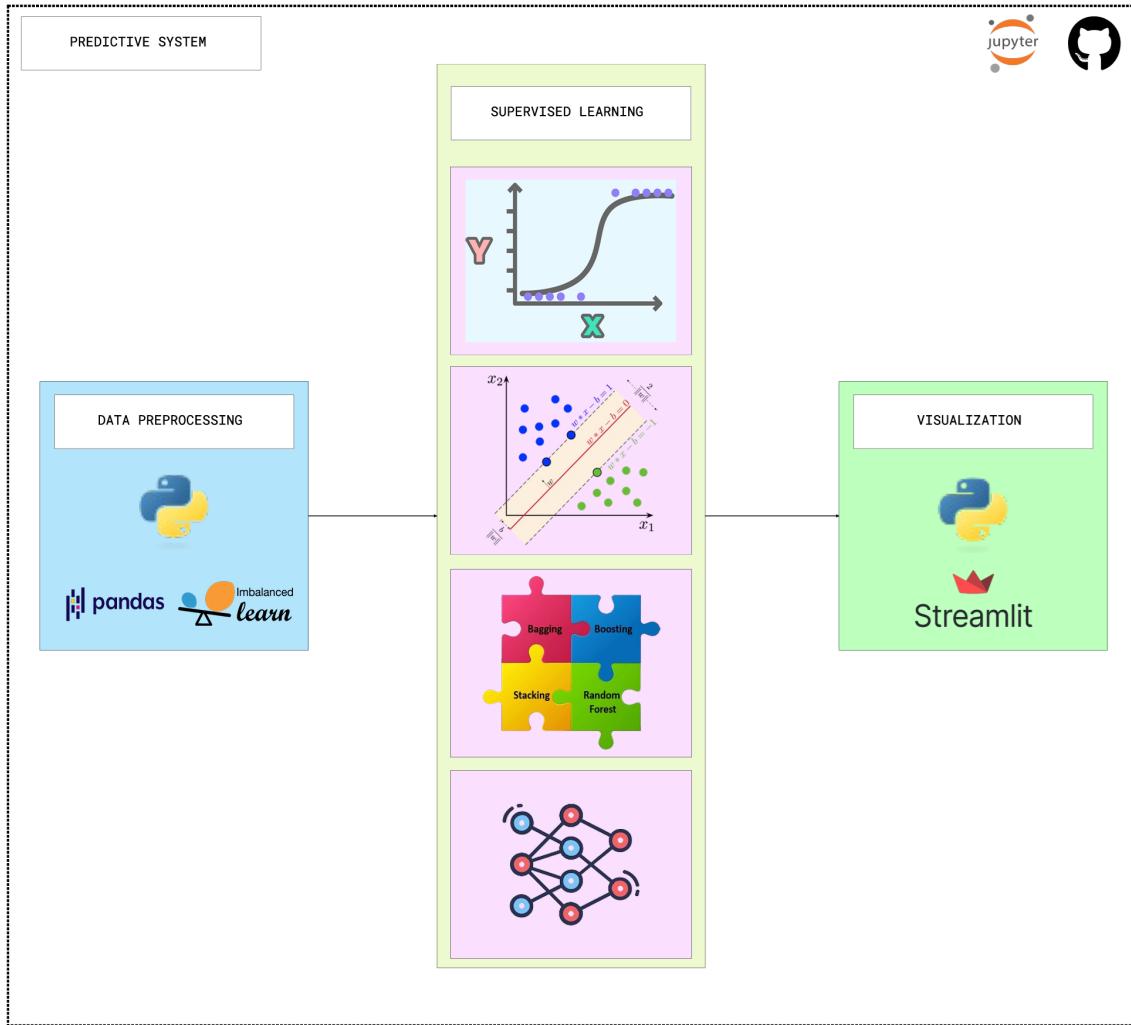


Figura 2.12: Diagrama de la investigación del Sistema Predictivo («Cloud Skew web», 2023)

La tercera etapa del proyecto se centra en la investigación acerca de desarrollar un **Sistema Predictivo** cuyo objetivo es poder **construir un modelo de Inteligencia Artificial** basado en **Machine Learning** que sea capaz de poder **predecir qué anime será de éxito** en pos de dar una ventaja competitiva temporal a la hora de licenciar a las distribuidoras nacionales frente a las internacionales.

1. *Data Preprocessing*

2. *Supervised Learning*

3. *Visualization*

1.- Data Preprocessing

En esta sección se ha investigado las librerías *pandas*, junto con las librerías de visualización *Seaborn* y *Matplotlib* de *Python*, con el fin de realizar la sección correspondiente al **Análisis Exploratorio de los Datos**, conocida como *Exploratory Data Analysis (EDA)* y poder estudiar el comportamiento de los datos.

- Estudiar la coherencia y relevancia de los valores nulos
- Estudiar la distribución de los datos
- Estudiar la correlación de cada una de las variables con respecto a la variable objetivo

Más adelante, se han investigado las mismas librerías con el fin de realizar la sección correspondiente a la **Creación de Atributos**, conocida como *Feature Engineering* y poder obtener y preparar los datos para su posterior consumo por los modelos correspondientes.

- **Binning:** agrupar valores por intervalos/contenedores
- **Transformaciones Matemáticas:** como la aplicación del logaritmo o el escalado
- **One-Hot Encoding y Binning:** Obtener datos cuantitativos gracias al análisis de los cualitativos

Por último, se ha investigado la librería *Imbalanced-learn* para gestionar el problema que hay con respecto al gran número de *animés* que no hay licenciados, frente a los que sí.

2.- Supervised Learning

En esta sección se ha investigado posibles técnicas de ***Machine Learning*** utilizando ***Scikit-Learn*** como principal librería. Dicho métodos planteados son:

- **Clasificadores Básicos:** Modelos generales como son el **Árbol de Decisión Común** (*Decission Tree Classifier*), el **clasificador Naive Bayes** (*Gaussian NB*) y la **Regresión Logística** (*Logistic Regression*) como punto de partida.
- **Máquina de Soporte Vectorial: Modelo No Lineal (SVC)** de las SVM debido a la no linealidad de los datos y su ventaja en los problemas de clasificación binaria.
- **Ensemble:** Ejecución de **varios modelos simultáneamente y agregando sus resultados** de diferentes formas (Voting - *Random Forest*, Stacking, Bagging, Boosting), teniendo la **ventaja de poder reducir la varianza de los resultados** al estar usando varios modelos al mismo tiempo.
- **Redes Neuronales:** Planteamiento de una **red artificial densa** que sea capaz de encontrar mejor los patrones entre los datos y clasifique mejor que los modelos anteriormente mencionados. Aquí, utilizaríamos *Frameworks* para desarrollo de redes neuronales.

3.- Visualization

Para terminar, se ha investigado cuál podría ser la ***output*** de este sistema. Como punto de partida se ha optado por ***Streamlit*** para poder **seleccionar el modelo que usar y ver los resultados/predicciones**, pero se ha planteado otra posibilidad como **desarrollar una API que reciba la serie en cuestión a predecir y se responda con la predicción asociada**.

Capítulo 3

Objetivos y Metodología

3.1. Objetivos a nivel proyecto

Después de la sección sobre el Estado del Arte donde se ha redactado una síntesis de la introducción y el contexto, además de la explicación de las principales premisas en las que se ha fundamentado el trabajo y las principales fuentes de información investigadas, se procede a definir los siguientes objetivos:

1. **A nivel de memoria:** Contribuir como investigadores en un ámbito del que no existe apenas bibliografía, sobre un fenómeno oriental en el mundo académico occidental.
2. **A nivel de desarrollo:** Implementar parte de la solución expuesta partiendo de la creación de un *dataset* obtenido mediante técnicas de *webscrapping*, el desarrollo parcial del **Sistema Recomendador** aplicando la técnica de *Named-Entity Recognition* y el desarrollo parcial del **Sistema Predictivo** utilizando todas las técnicas vistas en la librería *scikit-learn*, sin llegar a plantear una visualización final asociada a ambos desarrollos.
3. **A futuro:** En lo contextual, aportar con un impacto económico y cultural que mediante una solución tecnológica resuelva problemas propias de una comunidad nicho que sustentan a una industria multimillonaria. En lo técnico, mejorar ambos sistemas, desarrollar las técnicas restantes junto con la

visualización final, y seguir investigando acerca de nuevas posibilidades que implementar en consonancia con el modelo de negocio.

La memoria está cohesionada en torno a los antecedentes socioculturales sobre la necesidad identificada, analizando los avances tecnológicos de este campo. Finalmente, se validan los resultados obtenidos con diferentes técnicas usadas en los sistemas.

El núcleo del trabajo se centra en desarrollar dos sistemas.

Por un lado, tendríamos un **Sistema Recomendador** basado en la sinopsis de diferentes *animés*

Por otro lado, un **Sistema Predictivo** para verificar si es relevante localizar cierto contenido al mercado español.

Es necesario realizar esta división de los objetivos, ya que se trata de un trabajo de investigación y hay ciertos algoritmos que no serían abarcables para el *scope* de un equipo conformado por dos personas.

En el mundo de la Inteligencia Artificial, resulta complicado aportar innovaciones al Estado del Arte, a lo que se añade el hecho de que no se ha tratado este problema con anterioridad.

Por tanto, la mayor parte de los esfuerzos han ido destinado a la investigación en torno a encontrar un caso de uso concreto, donde aplicar el conocimiento producido en el mundo empresarial.

3.2. Objetivos a nivel contextual

Actualmente, las empresas licenciadoras de *anime* en ámbito nacional como son *Selecta Visión* y *Jonu Media* presentan los siguientes problemas:

1. **Proceso de *Reach* - Captación/Alcance:** Para llegar a un público que quiera consumir *anime* en plataformas determinadas.
2. **Proceso de *Engagement* - Mantenimiento/Retención:** El propio público consumidor de la plataforma no permanece en la misma, ni por su capacidad de atención ni de interés.

No se ha encontrado una fuente con información sobre la toma de decisiones que afronta la industria audiovisual española, por lo que el **objetivo final es realizar nuestro prototipo de aplicación para poder ofrecer una ventaja competitiva a las distribuidoras españolas**

Se resuelve así el problema, tanto de alcance como de retención, a potenciales consumidores de su contenido.

Se construye un ***Sistema Recomendador*** que solucionará dos de los principales problemas que tienen las plataformas españolas de streaming:

■ **Problema de *Reach*:**

Nuestro sistema priorizaría recomendar por similitud de la sinopsis aquel contenido que exista en la plataforma y, después, el resto del *dataset*.

■ **Problema de *Engagement*:**

Al recomendar contenido que se encuentre dentro de la plataforma, se mantendrá el tiempo que el usuario se mantenga más tiempo en ella.

Dependiendo del modelo de negocio, este procedimiento puede darse dentro de la plataforma o fuera, a modo de página web con otros clientes.

Por otro lado, gracias al ***Sistema Predictivo*** se realizarían predicciones de los posibles ***animés*** que van a ser de éxito, ofreciendo para las distribuidoras nacionales una gran ventaja competitiva para poder obtener una licencia frente a las distribuidoras internacionales. Además, se evitan inversiones de riesgos para las licenciatarias españolas solucionando el problema de *Reach/Engagement*, ya que se licenciarían en base a las necesidades de los usuarios.

3.3. Metodología

Tras haber aclarado los objetivos, se procede a definir la metodología de trabajo. A la hora de buscar un procedimiento que responda a los requerimientos del proyecto, se ha optado por un enfoque similar a la *metodología agile*.

Este sistema nos ha ayudado a organizar el trabajo de una manera rápida y concisa con reuniones predefinidas periódicamente, tanto a nivel de los investigadores que han redactado la memoria, como con el director del proyecto de investigación.

Las tareas se han ido dividiendo para marcar objetivos semanales, con un tratamiento similar a los *sprints* de la metodología *Scrum* usado en el ámbito corporativo.

La división de las tareas se han dividido en dos roles diferenciados que trabajaban conjuntamente:

- **La investigación cualitativa**, identificando las necesidades mercadotécnicas junto a su contexto sociocultural, así como el estudio de las técnicas propias de la lingüística computacional a usar en los *sistemas recomendadores*, corre a cargo de Jesús Carlos Avecilla de la Herrán.

- **La maquetación del planteamiento de la memoria**, junto al proceso de *webscraping* y el desarrollo técnico basado del *Sistema Predictivo*, corre a cargo de Nicolás Felipe Trujillo Montero.

Al ser un proyecto más enfocado al ámbito de investigación, la división y reparto de tareas en el entorno técnico se ha llevado paralelamente a la redacción de las investigaciones en la parte de la memoria.

Este trabajo se ha organizado y planificado mediante una aplicación similar a Microsoft Teams o Slack para la comunicación denominada *Discord*, con el fin de realizar la planificación y el reparto de tareas, recopilando, a su vez, información mediante canales de texto y audio.

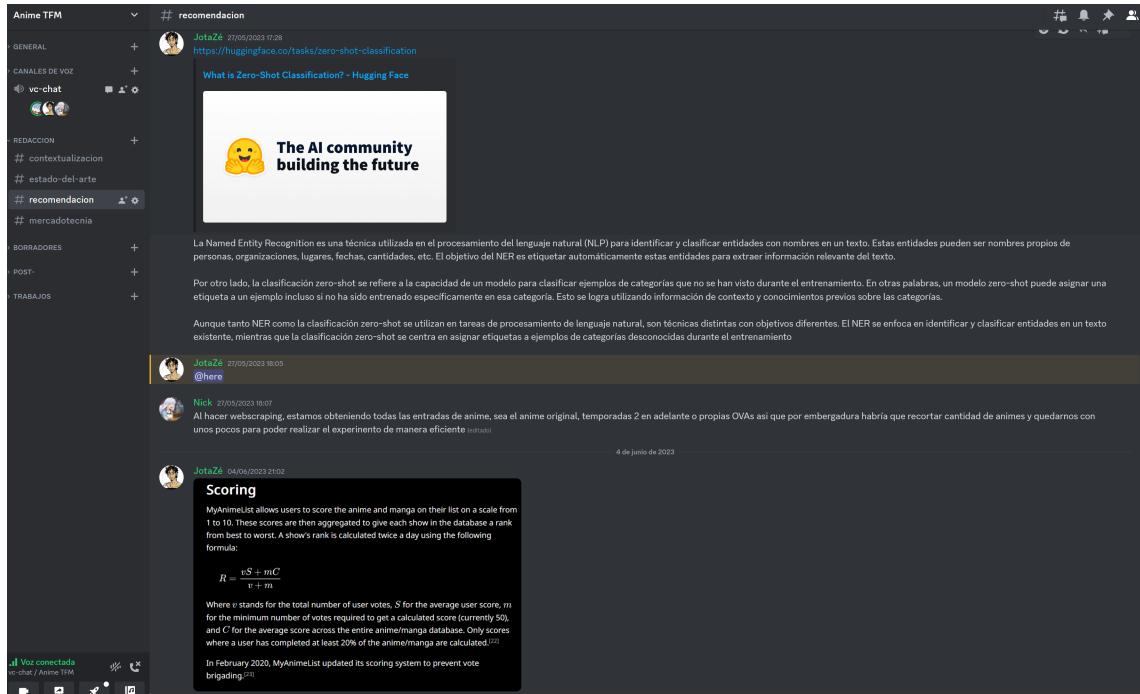


Figura 3.1: Captura del servidor de Discord

Todo este proceso se adscribe a un método científico de tipo deductivo, donde se parten de axiomas para llegar a una conclusión. Por tanto, se ha estructurado el trabajo de investigación siguiendo los pasos vistos en la asignatura de “Investigación en Inteligencia Artificial”.

Etapas del proceso y diseño de la investigación empírico-analítica:

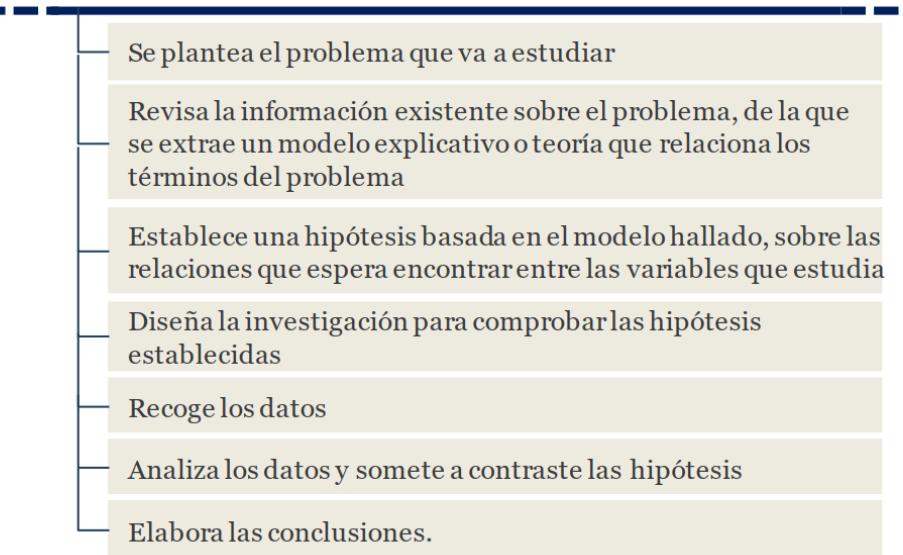


Figura 3.2: Método Científico

■ Problema a estudiar

Deficiencias en el modelo de negocio de las plataformas audiovisuales de streaming españolas, especializadas en anime.

■ Información existente

Aproximación cualitativa sobre el estado de la cuestión, al mismo tiempo que se investiga sobre el Estado del Arte.

■ Hipótesis inicial

Propuesta de solución mercadotécnica al problema expuesto.

■ Diseño de investigación

Desarrollo de *Sistema Recomendador* y de un *Sistema Predictivo*, aplicando diferentes técnicas.

- **Datos a recoger**

Creación de un *dataset* mediante *webscraping*.

- **Contrastando hipótesis**

Encuesta de validación a los potenciales usuarios.

- **Conclusiones**

Exposición de lo observado.

3.4. Tecnologías Investigadas

Tras haber explicado el planteamiento del proyecto a nivel técnico en el Estado del Arte, se ha realizado un capítulo adicional destinado a incidir en todas las tecnologías que se han investigado y/o utilizado a la hora de implementar la solución planteada. Para ello, se van a listar los siguientes tecnologías:

- **Control de Versiones:** Para realizar un seguimiento de las tareas, gestionar el proyecto y tener un repositorio en la nube con el contenido desarrollado se ha utilizado *GitHub*.
- **Lenguaje de Programación:** Debido a que se necesita un lenguaje enfocado en realizar *webscraping*, manejar eficientemente grandes volúmenes de datos, y destinado a realizar labores de *Machine Learning* y *Procesamiento del Lenguaje Natural*, se decidió utilizar *python*
- **Gestión de Entorno:** Para gestionar los *environments* y la instalación de paquetes asociados, junto con el uso de *Jupyter Notebook*, se ha utilizado la distribución de *python* llamada *Anaconda* ya que vienen incluidos bastantes *packages* para la ciencia y el manejo de datos.

Una vez definidas las principales tecnologías, se procede a definir por cada sección de la solución las librerías utilizadas:

1. Proceso ETL: Para realizar el proceso de *webscraping*, se planteó inicialmente usar una librería denominada *BeautifulSoup*, pero, más adelante, debido a ciertas dificultades encontradas se decidió usar *scrapy*, junto con un SaaS (*Software as a Service*) denominada *scraperapi*. Dicha librería nos permitía obtener la información de la página en forma de JSON. Finalmente, se utilizó la librería de *pandas* para finalizar.

2. Sistema Recomendador: Para realizar este proceso se requiere también del uso de *pandas*. Inicialmente, se propuso abordarla creación de las técnicas utilizando librerías propias del mundo del *Procesamiento del Lenguaje Natural*.

El pre-procesamiento de los textos suele realizarse mediante *NLTK* y *spaCy*, se ha descartado su uso ya que la transformación de los strings puede afectar al *output* deseado en cada aproximación, por lo que se procederá a hiperparametrizar como fase posterior a esta investigación. Además, para plantear las técnicas se usarían el *Framework* de *argilla*, junto con las librerías *pyod* y *gliner*.

3. Sistema Predictivo: Para realizar este proceso se requiere también del uso de *pandas*. Para realizar los procesos de Análisis Exploratorio de los Datos y Creación de Atributos se han utilizado las librerías *Matplotlib* y *Seaborn* y, para la creación de los modelos, se han utilizado *Scikit-Learn* para los modelos junto con la librería *Imbalanced-learn* para gestionar el desbalance de etiquetas. Por último, se investigó acerca de usar *TensorFlow* o *PyTorch* como librerías iniciales para la técnica de Redes Neuronales.

4. Visualization: Inicialmente, se planteó usar *shiny* y *R* para elaborar la visualización, aunque se optó por utilizar *streamlit* como herramienta principal al requerir menos coste de despliegue para posteriores versiones.

Capítulo 4

Proceso ETL

La memoria comienza dividiéndose en las secciones de “Introducción” y “Contexto” presentando los antecedentes socioculturales en torno a la animación japonesa. Le sigue “Estado del Arte” y “Objetivos” en las que se analizan los actuales avances tecnológicos de los cuales se parte para resolver la motivación de los investigadores, explicando como se va a trabajar en “Metodologías”

A continuación, se procederá al desarrollo técnico de las soluciones, que resolvrán la problemática planteada.

Haciendo referencia a la Figura 2.10 que vimos en la sección de “Estado del Arte”, **se entrará en detalle sobre el proceso ETL como paso previo** a la creación de los sistemas basados en IA sobre los que pivota la investigación.

4.1. Data Scraping (Extracción)

Para ello, se plantea utilizar **técnicas de extracción de datos a páginas web** para obtener la siguiente información:

- **Cualitativa:** El *Sistema Recomendador* necesita la **sinopsis** de los **animés** para establecer relaciones entre los mismos. Además, se podría complementar con información acerca de los **géneros y temas** asociados a la serie, o incluso la propia **demografía** como parte de una ontología clásica.
- **Cuantitativa:** Para poder realizar un análisis de impacto del **anime** mediante el *Sistema Predictivo*, necesitamos de **datos cuantitativos que ayuden al modelo a encontrar patrones en la clasificación de impacto**. Para ello, se han extraído datos como la **puntuación de los usuarios**, el *ranking* con respecto a otro contenido, o incluso la **puntuación del material original** como pueden ser el *manga* o novelas literarias.

Habiendo identificado los requisitos necesarios, utilizamos la página web ***MyAnimeList*** como **fuente principal** de donde extraer los datos necesarios, ya que su función como **base de conocimiento cumplía con los estándares de datos**.

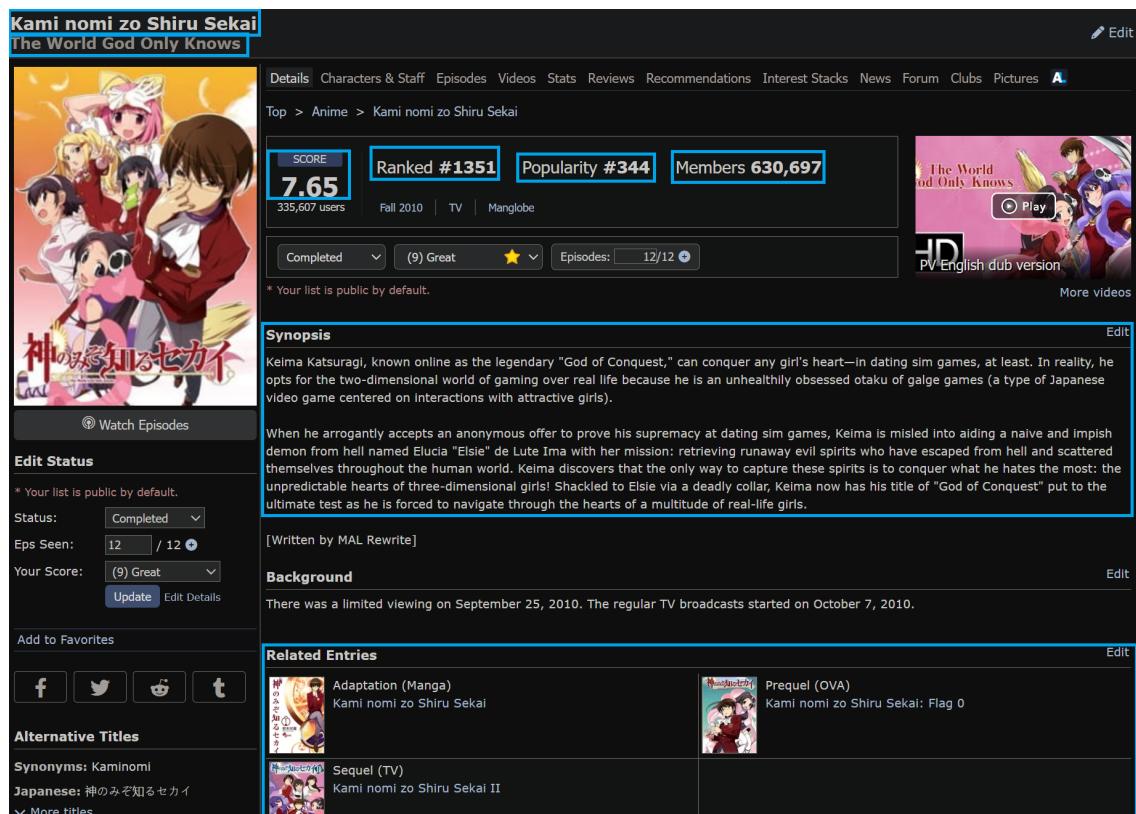


Figura 4.1: Captura de MyAnimeList (Gyssler, 2023)

Para realizar el ***webscraping*** de esta página se decidió usar ***Python*** y la librería ***Scrapy*** para poder obtener, entre otros, los datos que se ven en la Figura 4.1.

Gracias a esta librería, se obtiene el **código HTML5** de una página, a través de bots conocidos como ***spiders*** o ***data scrapers***.

De esta forma, **de manera automatizada**, todos aquellos **datos** que se hayan indicado en el código mediante el uso de los **contenedores** y **otras etiquetas** pertenecientes al lenguaje de marcas, serán **obtenidos**.

Mientras que se desarrollaba el código, se encontró un **problema con las peticiones**. A día de hoy, las páginas web tienen un sistema de **control de acceso**

a los servidores gestionados internamente y con un fichero denominado *robots.txt*. Este sistema imposibilitaba realizar grandes cantidades de peticiones por segundo, por lo que se optó usar un SaaS (Software as a Service) como solución al problema denominado *scraperapi*.

Dicho SaaS hacía posible realizar grandes cantidades de peticiones por segundo, ya que, aunque se hacían todas las peticiones desde la misma IP, esta petición antes de pasar por *MyAnimeList* pasaba por *scraperapi* cambiándose la IP por otra diferente similar a un proxy, con una técnica llamada *Rotating IPs* y llegaba al destino como llamadas desde distintas IPs.

Una vez realizado dicho proceso, obtenemos un archivo “.json” que se convertirá en un *dataframe* de *pandas* con la siguiente información:

- 'JP_Title': Japanese anime name
- 'EN_Title': English anime name (If it exists)
- 'Synopsis': Synopsis about the anime
- 'Type': Type of series: TV=Anime, ONA, ...
- 'Status': If an anime is airing, was finished or will be aired (Currently Airing, Finish airing or Not yet aired)
- 'Aired': The time interval of the Status variable
- 'Premiered': Planned aired season
- 'Source': If the series have some prequel or similar (Usually, an anime is aired because by a previous manga)
- 'Genres': Genres of the anime
- 'Themes': Themes of the anime
- 'Demographic': Demographic of the anime (To whom it is addressed)
- 'Score' (Calculated weighted score by users) (If it doesn't have any, it is obtained from the manga - 0.5)
- 'Ranked': ranking of the animes based on the top animes by score
- 'Popularity': ranking of the animes based on the number of viewers
- 'Members': Quantity of users that have the anime in their anime list
- 'Sequel': Indicates the Sequel from the anime
- 'Prequel': Indicates the Prequel from the anime
- 'Alternative_setting': Indicates the Alternative_setting from the anime
- 'Alternative_version': Indicates the Alternative_version from the anime
- 'Side_story': Indicates the Side_story from the anime
- 'Summary': Indicates the Summary from the anime
- 'Full_story': Indicates the Full_story from the anime
- 'Parent_story': Indicates the Parent_story from the anime
- 'Spin-off': Indicates the Spin-off from the anime
- 'Adaptation': Indicates the Adaptation from the anime
- 'Character': Indicates the Character from the anime
- 'Other': Indicates the Other from the anime
- 'Type Source': Indicates the adaptation from the anime (Source)
- 'JP_Source_Title': Indicates the JP_Source_Title from the anime
- 'Score_source': Indicates the Score_source from the anime
- 'Ranked_source': Indicates the Ranked_source from the anime
- 'Popularity_source': Indicates the Popularity_source from the anime
- 'Members_source': Indicates the Members_source from the anime

Figura 4.2: Columnas obtenidas de MyAnimeList (Gyssler, 2023)

Habiendo completado el proceso de extracción, para el *Sistema Recomendador* ya tenemos las variables necesarias cualitativas para poder desarrollarlo. Sin embargo, para el *Sistema Predictivo* necesitaríamos obtener la variable *target* la cual representaría si el *anime* está licenciado en alguna plataforma de *streaming*.

El enfoque que se optó al comienzo de la investigación era realizar la misma técnica presentada en *MyAnimeList* (Gyssler, 2023), pero para las plataformas de *Netflix*, *Crunchyroll*, *Selecta Visión* y *Jonu Media* (GROUP, 2023; Kun Gao, 2023; Reed Hastings, 2023; S.L.U., 2023), aunque no se pudo realizar debido a las

restricciones de control comentadas con anterioridad.

Al final, se optó por usar dos *dataframes* externos obtenidos de las plataformas de *Kaggle* y *HuggingFace* que incluían los *animes* licenciados en las plataformas internacionales, no teniendo éxito encontrando los nacionales.

4.2. Data Curation (Transformación y Carga)

Tras el ***Data Scraping*** se va a realizar la etapa de ***Data Curation*** en la cual se van a **limpiar y filtrar los datos**, según convenga para ambos desarrollos.

Para comenzar, se filtrará el ***dataframe*** proveniente del ***webscraping*** de ***MyAnimeList*** (Gyssler, 2023) de tal forma que **se eliminará aquellos registros de tipo “Music” o “Unknown”**, ya que no aportan información relevante para ninguno de los dos desarrollos.

A continuación, se crean dos ***dataframes*** los cuales comenzarán siendo copias del ***dataframes*** previo y que, según la necesidad del desarrollo, serán sometidos una serie de cambios.

4.2.1. Dataframe del Sistema Recomendador

Para realizar las transformaciones necesarias para el ***Sistema Recomendador*** tenemos que hacernos las siguientes preguntas:

- **¿El sistema recomendará próximos lanzamientos?** No, no es relevante recomendar algo que no ha salido si no se encuentra disponible.

Por lo tanto, aquellos animes cuya variable “*Aired*” valga “Not Available” son eliminadas.

- **¿Cada registro es único?** No, ya que puede haber varias filas por *anime* dependiendo del número de fuentes primarias que tenga.

Como no tiene sentido que en un ***Sistema Recomendador*** aparezca el mismo *item* más de una vez, se ha convertido a una única entrada.

- **¿El sistema recomendaría series enteras o series por temporadas?**
Se recomendarían series enteras sin que sean dependientes de una temporada concreta, al no haberse visto sin haber consumido las anteriores.

Por lo tanto, se filtra por aquellos *animes* que **no tengan precuela, ni Full Story, ni Parent Story**, como aproximación para solucionar este problema.

En este punto existen anomalías, ya que hay filas correspondientes al mismo *anime* que no se pueden filtrar por lo comentado anteriormente.

- **¿Qué registros se van a recomendar?** Se plantea recomendar únicamente aquellos *animes* que sean de **tipo “TV”, “Movie” o “ONA” (Original Net Anime)**, ya que no es relevante recomendar “OVA” (Original Video Animation) o “Special”, ya que suelen ser series procedentes de otras, existiendo anomalías.

4.2.2. DataFrame del Sistema Predictivo

Para realizar las transformaciones necesarias para el *Sistema Predictivo* tenemos que hacernos las siguientes preguntas:

- **¿Cada registro es único?** Sí, ya que puede haber varias filas por *anime* dependiendo del número de fuentes primarias, las cuales sean relevantes para identificar el comportamiento de los usuarios.
- **¿El sistema clasificará todas las próximas series que se estrenen?** No, solo se clasificarán aquellas que no son de tipo “Original”, es decir, aquellas que provienen de una fuente primaria antes de estrenarse.

Por lo tanto, se filtra por aquellos *animés* que no son de tipo “Original”.

- **¿De dónde obtenemos la variable *target*?** Se ha creado en el *dataframes* una nueva columna llamada “Licensed” que contendrá un 1 o un 0, dependiendo si el *anime* ha sido licenciado o no en alguna de las plataformas mencionadas. Para los próximas series que se vayan a lanzar, se considerarán esta variables sin inicializar, con el fin de predecirla.

Capítulo 5

Sistema Recomendador de Animes basado en el contenido

5.1. Identificación de Requisitos

Los *sistemas recomendadores* están diseñados para sugerir y recomendar elementos de forma personalizada para los usuarios, ayudándoles a encontrar propuestas de interés en medio de una gran cantidad de opciones disponibles.

5.1.1. Enfoque actual de los sistemas recomendadores

Como se ha visto en la investigación cualitativa, hay un **problema en el sector de la industria del anime ante una excesiva oferta**, que no orienta a los usuarios hacia plataformas de streaming gestionadas por distribuidoras españolas, que ven afectado su modelo de negocio de importación audiovisual ante su incapacidad de atraer a potenciales consumidores.

Del mismo modo, los *sistemas recomendadores* de las plataformas de *streaming* aplican una **metodología basada en etiquetas arbitrarias para cada ítem audiovisual, las cuales al aplicarse se dan en relaciones de proximidad por consumo, y no por relevancia sobre el contenido mismo**:

Series de culto que cambiaron el medio como **Cowboy Bebop** (Kuga, 1998) será seguida de otras aclamadas por la crítica como **Ghost in The Shell** (Kamiyama, 2002), pero no lo hará con series de temática futurista similar como **Cyberpunk Edgerunners** (Jaki, 1983) **si no existiese ese etiqueta que da nombre a un género propio**, aplicada manualmente de manera arbitraria.

Se aplicarían etiquetas en los géneros demográficos ya desfasados. Es decir, no dan series con tramas similares relevantes para su potencial *target*:

Si tomamos dos series de enfoque juvenil como **Black Clover** (Tabata, 2002), **shounen** sobre magos, y por otro lado, **Little Witch Academia** (Yoshinari, 2017) como **shoujo** sobre brujas, **tenemos uno para “niños” y otra para “niñas”**.

Esta última es consumida mayormente por hombres, estableciendo **distinciones binarias además de incluir la edad como variable**.

Para la **hipótesis central** de este desarrollo se tomará la **narrativa de las mismas para crear conexiones más certeras** en consonancia con los *Sistema Recomendador Content-Based*.

No sería adecuado un filtrado que se base en contenido que se haya consumido previamente, ya que no resolvería el problema de *reach/engagement* que enfrentan los clientes.

Para ello, se usarán técnicas propias del **Procesamiento del Lenguaje Natural** como puede ser *Named-Entity Recognition (NER)*, *Text Similarity*, *Clustering* y *Topic Modeling*.

Las consideraciones que se toman como hipótesis iniciales constan de las siguientes características...

- La técnica **NER** requerirá **gran coste de entrenamiento**, debido a que las

series se etiquetarían con una ontología creada *ad-hoc*.

- La técnica ***Text Similarity*** no sería **a adecuado para series de tipo humor absurdo**, debido a que carecen de cohesión textual.
- La técnica ***Clustering*** no sería totalmente **a adecuada**, debido a que **solo recomendaría series del mismo género**.
- La técnica ***Topic Modeling*** se considera como la más adecuada, debido a que **puede reconocer el tono de la serie**.

Se ha realizado una **encuesta evaluando si los enfoques planteados resuelven los problemas expuestos con distintos casos de uso**, validándose en el apartado de evaluación.

Cada técnica se explicará con detalle en el resto de este apartado.

5.2. Descripción de la herramienta software desarrollada

En esta sección abordaremos todos los puntos que engloben la secuencia de ejecución del algoritmo, profundizando en cada uno de los puntos.

5.2.1. Obtención y Preprocesamiento de los Datos

Para plantear el uso de los algoritmos de *Procesamiento del Lenguaje Natural* de esta sección, se ha partido de la idea de construir un *dataframe* asociando cada *anime* a su correspondiente sinopsis.

Para ello, se ha construido dicho *dataframe* con las aportaciones de los usuarios de *MyAnimeList* (Gyssler, 2023).

En este página aparecen los datos que buscamos de cada *anime*, obteniendo por cada fila el nombre del anime, su sinopsis, sus géneros y, en ocasiones, su demografía.

Con esto, se ha tomado las sinopsis como dato principal a la hora de aplicar las técnicas de *Procesamiento del Lenguaje Natural* con cada uno de sus enfoques para crear los *sistemas recomendadores* como parte del desarrollo.

5.2.2. Técnicas utilizadas

Entre las técnicas a utilizar de Procesamiento del Lenguaje Natural, se ha planteado los siguientes algoritmos potenciales:

Named-Entity Recognition

En una primera aproximación podríamos usar técnicas de Named-Entity Recognition (**NER**), para establecer vectorizaciones entre strings con la que aplicar un meta-etiquetado con una ontología establecida por temática.

Este sería el enfoque a desarrollar, ya que es la que ofrece resultados a corto plazo a pesar de necesitar entrenamiento mediante hiperparámetros.

La librería de **SpaCy** solo contiene etiquetas pre-entrenadas, por lo que tendríamos que crear nuestras propias etiquetas ad-hoc, pudiendo automatizarse de manera semi-supervisada con modelos de anotación como **Argilla**.

Se ha usado la librería **Gliner** la cual permite asignar etiquetas automáticamente mediante *zero-shot learning*.

El enfoque basado en **NER** no debe confundirse con las técnicas de **Word-embedding**.

Estas técnicas se utilizan para representar palabras en forma de vectores numéricos, capturando información semántica y sintáctica, mientras que el **NER** es una tarea específica en **Procesamiento del Lenguaje Natural** que se centra en identificar y clasificar entidades nombradas en un texto.

Los **Word-embeddings** pueden ser una herramienta útil para el (**NER**), ya que los modelos de (**NER**) pueden aprovechar la información de los vectores de palabras para mejorar su rendimiento en la identificación de entidades.

Un caso de uso sería la sinopsis de One Piece (Oda, 1999) como (*shounen nekketsu*), serie de corte juvenil, dándose las siguientes categorías que la definirían:

- Reconocimiento de la Etiqueta/Entidad “**Piratas**” como tema principal: “Pirate King”, “Grand Age of Pirates”, “standard definition of a pirate”, “toothless pirate ransacking villages for fun”
- Reconocimiento de la Etiqueta/Entidad “**Protagonista masculino joven**”: “Monkey D. Luffy, a 17-year-old boy”
- Reconocimiento de la Etiqueta/Entidad “**Acción**”: “battling strong enemies”
- Reconocimiento de la Etiqueta/Entidad “**Aventuras**”: “exciting adventure” o “experiencing crazy adventures”
- Reconocimiento de la Etiqueta/Entidad “**Misterio**” (Anomalía, ya que no es un tema principal del *anime*): “unveiling dark mysteries”

Otras series de piratas incluyen la sinopsis de **Black Lagoon** (*Katabuchi, 2006*) “a band of ruthless **pirate** mercenaries” o la de **Capitán Harlock** (*Matsumoto, 1978*) “and his ragtag group of **pirates**” ambas con un enfoque contemporáneo y futurista respectivamente.

Se abre aquí un debate sobre si considerar más relevante recomendar otras series fuera del *mainstream* dirigidas a un público joven masculino que la mayoría de personas aficionadas ya conoce como *Kishimoto, 2002* o *Toriyama, 1986*. Es necesario priorizar un contenido fuera de lo comercial siendo prioridad ante los requerimientos de los clientes, que no pueden competir con la acaparación mediática de las mismas que ya aparecen en otras plataformas.

Text Similarity

Entre las técnicas de *Text Similarity* en la que podríamos tomar las tramas en su totalidad, tendríamos *Sentence Transformers*:

A diferencia de los **Word-embeddings**, que se centran en la representación de palabras individuales, los **Sentence Transformers** se enfocan en capturar la información semántica y contextual de oraciones o textos completos.

Si tomamos series de samuráis, el caso de **Rurouni Kenshin** (Watsuki, 1996), **Samurai Champloo** (Watanabe, 2004) o **Gintama** (Sorachi, 2003) presentan enfoques diferentes en su tono, siendo la última de humor absurdo.

Con un algoritmo de cohesión de este estilo podríamos averiguar la divergencia entre estas para mejorar resultados, ya que las series de humor absurdo presentarán menor coherencia semántica.

- Por ejemplo, en la sinopsis de la serie de humor absurdo **Cromartie High School** (Nonaka, 2003) tenemos la frase “*he’s surrounded by mohawked punks, obnoxious robots, and... gorillas? And was that Freddie Mercury riding a horse down the corridor?*” la cual demuestra incoherencia textual que puede dar resultados anómalos.

Clustering

Por otro lado, el uso de clusterización en textos vectorizados es útil para animes musicales con diferentes géneros, desde el heavy metal de **Detroit Metal City** (Nagahama, 2002) o el indie rock de **Beck** (Kobayashi, 2002) se vuelve necesario clusterizar por grupos, ya que en este último no aparece en su sinopsis la palabra “music”.

Apuntes sobre las técnicas no usadas

Cabe recordar que un **Sentiment Analysis** no aplicaría a nuestra metodología, ya que las sinopsis reflejan de manera objetiva la trama sin juicio subjetivo de valor asociado.

Sin embargo, en el caso de los animes que vayan de música no todas contaran con la misma emocionalidad:

- Por ejemplo, en la serie **K-On!** (Yamada, 2009) tendríamos “*revolves around the members of the Light Music Club as they experience their daily high school life. From rehearsing for concerts to just messing around, they are ready to make their last year together an exciting one!*” mientras que en **Your Lie In April** (N. Arakawa, 2014) aparece como “*When his mother suddenly passes away, the subsequent trauma makes him unable to hear the sound of a piano, and he never takes the stage thereafter.*”

Demostrándose así que un *Sentiment Analysis* podría servir en algunos casos, aunque no aportaría para llegar a nuestro objetivo final.

Topic Modeling

Por último, para poder solucionar el problema presentado con *Sentiment Analysis*, se ha planteado usar la técnica *Topic Modeling*.

Dicha técnica se basa en obtener cuál es el tema principal de un texto, solucionando así la disonancia entre **K-On!** (Yamada, 2009) y **Your Lie In April** (N. Arakawa, 2014).

Sin embargo, es una técnica efectiva en textos largos, pero en las sinopsis recopiladas no nos serviría.

5.2.3. Desarrollo de las técnicas

Caso de Uso - NER

NER significa “*Named Entity Recognition*” o “Reconocimiento de Entidades con Nombre”. Es una técnica de procesamiento del lenguaje natural (*Procesamiento del Lenguaje Natural*) que se utiliza para identificar y clasificar entidades con nombre en un texto.

El proceso de NER generalmente implica el uso de técnicas de aprendizaje automático y modelos de *Procesamiento del Lenguaje Natural* entrenados previamente. Estos modelos buscan patrones lingüísticos y contextuales en el texto para inferir y etiquetar las entidades con nombre correctamente. Para entrenar nuestras propias etiquetas, necesitaremos definirlas previamente para salirnos de los modelos pre-entrenados como la que ofrece (*spaCy*).

En el caso que nos ocupa, tomaremos las series *mainstream* de acción para adolescentes, conocidas como *shonen nekкetsu*, para crear subconjuntos de los mismos. Estas suelen recomendarse priorizando su popularidad, sin considerar dividirlas por temas a nivel narrativo.

Aunque ya hemos contemplado el caso de uso de **One Piece** (Oda, 1999), usaremos la serie **Full Metal Alchemist** (H. Arakawa, 2003) y **Black Clover** (Tabata, 2002) que usan poderes de “magia” en una trama centrada entre dos hermanos que podría enlazarse con una serie similar con los mismos elementos, como **Black Clover**.

Para empezar, importamos la librería de (*spaCy*), creando un modelo de lenguaje en blanco para definir nuestras propias etiquetas para entidades definidas como “MAGIC” y “BROTHERS”.

Después, agregamos el componente de reconocimiento de entidades al pipeline

con etiquetas al componente de reconocimiento de entidades. Los datos de entrenamiento anotados serán los siguientes para **Full Metal Alchemist**:

```
1 TRAIN_DATA = [
2     (''embarks on a journey with his younger brother'', {'entities':
3         ''': [(:, ''BROTHER'')]}) ,
4     (''he and his brother Alphonse try to resurrect their dead
5      mother'', {'entities': [(:, ''BROTHER'')]})
6     (''Edward loses his brother as well as two of his limbs'', {'entities':
7         ''': [(:, ''BROTHER'')]})
8     (''The fabled mythical object is rumored to be capable of
9      amplifying an alchemist's abilities '', {'entities': [(:, ''MAGIC'')]})
```

Acto seguido, entrenamos el modelo probándolo con la sinopsis completa. Cabe recordar que ya existen herramientas de etiqueta semi-automático en el mercado que evitan hardcodear el entrenamiento de estas etiquetas, como *argilla*, aunque el tiempo invertido en el mismo no compete al trabajo de investigación.

Caso de uso - Text Similarity

La “*text similarity*” o similitud de texto se refiere a la medida en que dos o más piezas de texto son similares entre sí.

Existen varias formas de calcularse, como el cálculo de la distancia o similitud del coseno entre vectores de representación de palabras/frases como Word2Vec con su modelo pre-entrenado de Google, usando la biblioteca gensim.

Definimos las dos “frases” a comparar, en este caso sinopsis completa:

```
1 sentence1 = '''In the final years of the Bakumatsu era lived a
   legendary assassin known as Hitokiri Battousai. Feared as a
   merciless killer, he was unmatched throughout the country, but
   mysteriously disappeared at the peak of the Japanese Revolution.
   (...) Rurouni Kenshin: Meiji Kenkaku Romantan tells the story
   of Kenshin as he strives to save those in need of saving.
   However, as enemies from both past and present begin to emerge,
   will the reformed killer be able to uphold his new ideals?'''

2
3 sentence2 = '''(...) Enter Gintoki Sakata, an eccentric silver-
   haired man who always carries around a wooden sword and
   maintains his stature as a samurai despite the ban. As the
   founder of Yorozuya, a small business for odd jobs, Gintoki
   often embarks on endeavors to help other people-though usually
   in rather strange and unforeseen ways. (...)'''
```

Por último, tokenizamos las frases y eliminamos palabras que no estén en el vocabulario del modelo, calculando el vector promedio para cada frase al mismo tiempo que la similitud del coseno entre los vectores.

En este caso, hemos tomado series sobre samuráis con diferente tono por lo que concluimos que se queda insuficiente con respecto a la cohesión/coherencia para identificar animes con tramas de humor absurdo.

Caso de uso - Clusterización

La clusterización de textos es un proceso en el que se agrupan documentos similares en clústeres o grupos. Con ello, exploraremos si las sinopsis que contienen un tono dramático pueden agruparse para nuestro caso de uso.

Se diferencia del *Sentiment Analysis* ya que no clasifica los textos en una categorización multiclasificación emocionalidad gradual, en forma de positivo/neutro/negativo.

Este etiquetado se necesita entrenar previamente sin que se vean las relaciones entre los elementos, por lo que el *Sentiment Analysis* no resulta relevante a la hora de establecer semejanzas entre el contenido.

Mediante la similitud entre dos vectores con la similitud de coseno, se puede observar como de alejados se encontrarían los ejemplos en un espacio, similar a los *Word Embeddings*.

Para comparar (con)textos se usa la técnica del TF-IDF, o *Term Frequency-Inverse Document Frequency*, en la que se considera el contenido relevante en una clasificación documental, creando subconjuntos dentro de un corpus.

Por otro lado, se necesitan los *K-Means* como parte de un algoritmo de aprendizaje no supervisado para acotar los clústeres por un numero predefinido por el usuario.

A nivel de código, tomariamos entonces las sinopsis como vectores TF-IDF, calculando la similitud de coseno entre ellos y aplicando los *K-Means* para agruparlos en clústeres, y mostrar finalmente los resultados.

Primero, importamos las librerías para hacer procesos de *Machine Learning* como vectorización con *numPy* y clusterización con *Scikit-Learn*.

A continuación, creamos un conjunto con varias sinopsis a modo de corpus, encajando aquellas series de deporte que traten sobre futbol (Oliver & Benji, Inazuma Eleven, Blue Lock) mezcladas con tenis (Prince of Tennis) y volleyball (Haikkyu)

```
1 synopsis = [
2     ''Captain Tsubasa is the passionate story of an elementary
3         school student whose thoughts and dreams revolve almost entirely
4             around the love of soccer.(...) Representing Japan in the FIFA
5                 World Cup is Tsubasa's ultimate dream, but it will take a lot
6                     more than talent to reach it.'',
7
8     ''While other schools in Japan compete for the title of being
9         the best soccer team in the country, Raimon Middle School's
10            soccer club, Inazuma Eleven, struggles to rise from the verge of
11                being disbanded. (...)'', 
12
13    ''Yoichi Isagi was mere moments away from scoring a goal that
14        would have sent his high school soccer team to the nationals,
15            but a split-second decision to pass the ball to his teammate
16                cost him that reality. (...) When the young striker returns home
17                    , an invitation from the Japan Football Union awaits him. ...
18                    The project's ultimate goal is to turn one of the selected
19                        players into the star striker for the Japanese national team
20                            (...)'', 
21
22    ''At the request of his father, tennis prodigy Ryouma Echizen
23        has returned from America and is ready to take the Japanese
24            tennis scene by storm. Aiming to become the best tennis player
25                in the country, he enrolls in Seishun Academy-home to one of the
26                    best middle school tennis teams in Japan (...)'', 
27
28    ''(...) Even though his attempt to make his debut as a
29        volleyball regular during a middle school tournament went up in
30            flames,(...) when Hinata enrolls in Karasuno High School, the
31                Little Giant's alma mater, he believes that he is one step
32                    closer to his goal of becoming a professional volleyball player.
33                        (...)'', 
34
35]
```

12]
13

Con esta información, tomamos la función *TFIDFVECTORIZER* que representan los textos como vectores numéricos utilizando la técnica de TF-IDF. Luego, se calcula la similitud de coseno entre los vectores de las sinopsis utilizando el producto punto de la matriz TF-IDF. Después, se utiliza el algoritmo *K-Means* para realizar la clusterización en base a TF-IDF guardando en los clústeres en la variable “etiquetas” teniendo por último los resultados de la clusterización con la sinopsis agrupadas en cada clúster.

Caso de uso - Topic Modeling

El Topic Modeling es una técnica de procesamiento de lenguaje natural que se utiliza para descubrir los temas presentes en un conjunto de documentos sin etiquetar, que podría servir como alternativa al modelo NER.

La técnica más usada es el modelo de LDA (Latent Dirichlet Allocation) partiendo de la idea de que cada documento está compuesto por una mezcla de varios temas, y cada tema está representado por una distribución de palabras asumiendo que hay una cantidad fija de temas en el corpus.

Importamos el gensim, librería para procesamiento de texto, y creamos la muestra con las siguientes series *Your Lie In April*, *K-On!*, *Anohana*, *Clannad*, *Violet Evergarden*

```
1             documents = [
2                 ''Kousei Arima is a child prodigy known as the ''Human
3                     Metronome'' for playing the piano with precision and perfection.
4                     Guided by a strict mother and rigorous training, Kousei
5                     dominates every competition he enters (...). When his mother
6                     suddenly passes away, the subsequent trauma makes him unable to
7                     hear the sound of a piano, and he never takes the stage
8                     thereafter. (...) While struggling to get over his mother's
9                     death, he continues to cling to music. His monochrome life turns
10                    upside down the day he encounters the eccentric violinist Kaori
11                    Miyazono (...) These two young musicians grow closer together
12                    as Kaori tries to fill Kousei's world with color.''
13
14                 ''(...) K-On!! revolves around the members of the Light Music
15                     Club as they experience their daily high school life. From
16                     rehearsing for concerts to just messing around, they are ready
17                     to make their last year together an exciting one!''
18
19                 ''Jinta Yadomi is peacefully living as a recluse, spending his
20                     days away from school and playing video games at home instead.
21                     (...) Jinta and his group of childhood friends grew apart after
```

her untimely death, but they are drawn together once more as they try to lay Menma's spirit to rest. Re-living their pain and guilt, will they be able to find the strength to help not only Menma move on-but themselves as well?'

7
8 ''Tomoya Okazaki is a delinquent (...) Nagisa claims they are now friends (...) Tomoya learns Nagisa has been held back a year due to a severe illness and that her dream is to revive the school's drama club (...) He decides to help her achieve this goal along with the help of four other girls. (...)''

9
10 ''The Great War finally came to an end after four long years of conflict (...) the continent of Telesis slowly began to flourish once again (...) Violet Evergarden, a young girl raised for the sole purpose of decimating enemy lines. (...) Violet begins work as an Auto Memory Doll, a trade that will take her on an adventure, one that will reshape the lives of her clients and hopefully lead to self-discovery.''

11]

12

Excepto en el caso de **K-On!**, estas son series dramáticas con temática y narrativa diferentes, queremos averiguar si se toman por similares con esta técnica.

Se preprocessará el texto si fuese necesario, creando diccionarios y una matriz de términos-frecuencia para usarlos como entrada para el modelo de LDA, obteniendo los temas y palabras clave.

5.3. Aplicación para el end-user

En este apartado, aclaremos como es la interacción entre el usuario y el algoritmo.

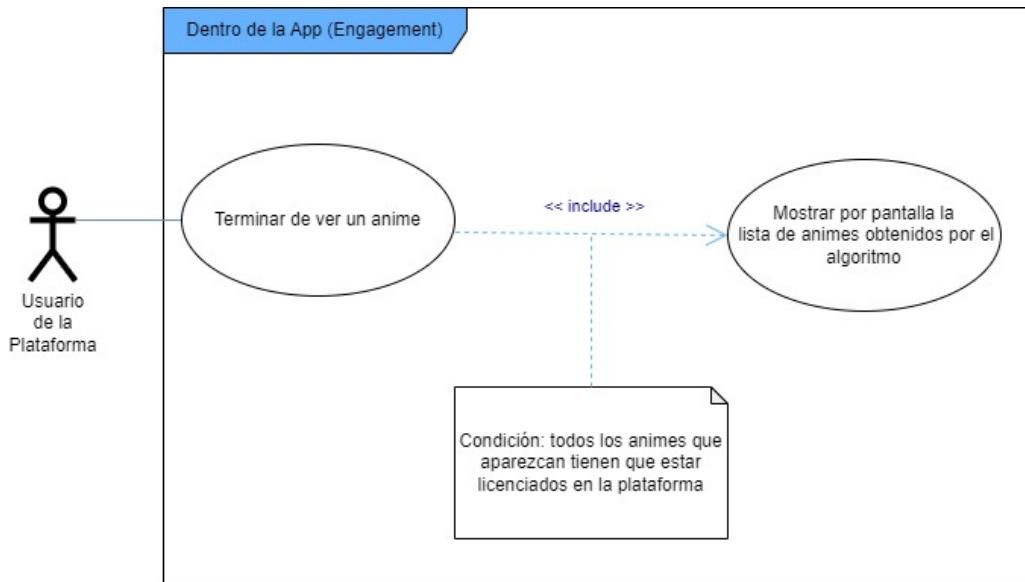


Figura 5.1: Caso de Uso: Engagement

1. Caso de Uso dentro de la Aplicación (*Engagement* → Retención de audiencia):

Un usuario, tras acabar de ver un *anime* licenciado en la plataforma, gracias al *Sistema Recomendador*, se le va a recomendar otro *anime* licenciado en la plataforma, basándose en la sinopsis del mismo.

Es importante remarcar que dependiendo del algoritmo utilizado se ordenará de una forma o de otra. Por ejemplo, si utilizamos *clustering* o *Sentence Transformer* con *text-similarity*, cuanto más cerca del *cluster* o menos distancia haya entre sus vectores, respectivamente, más parecido serán los animes entre sí y más prioridad tendrán a la hora de recomendar.

Sin embargo, para la técnica de *NER* no existe forma implícita de jerarquizar esta información, por lo que según lo planteado, se realizaría un conteo de

las entidades encontradas en ambos animes, siendo dos animes más parecidos cuanto mayor número de entidades tienen en común.

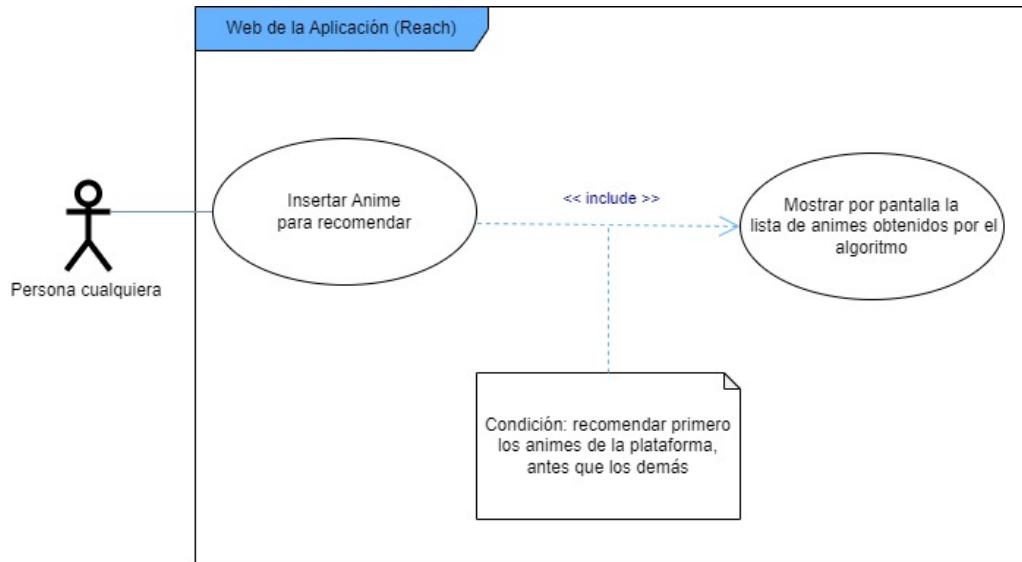


Figura 5.2: Caso de Uso: Reach

2. **Caso de Uso fuera de la Aplicación (*Reach* → Captación de audiencia):** Un usuario en internet accedería a una base de datos de *anime* general, introduciría un *anime* para que le recomiende otro, gracias al *Sistema Recomendador*, se podría recomendar otro contenido que no tiene porqué estar en la aplicación de *streaming*.

Funcionaría de la misma forma que el anterior en cuanto al tema de priorizar, pero añadiendo que se priorizaría primero por los *animés* que estén en la plataforma, y en segundo plano, los que no.

Esto sería nuestro modelo de negocio de realizarse una app viable comercialmente, colaborando con las plataformas interesadas.

Gracias a esto, motivaríamos a las personas que sean usuarios actuales a seguir en la plataforma, mientras que captaríamos la atención de aquellas que no la sean para que consuman contenido en la misma.

5.4. Análisis del desarrollo

En la siguiente muestra, se crea un script que anota una de las sinopsis con unas etiquetas pre-definidas usando *gliner*:

```
bcSynopsis = base_df[base_df["JP_Title"] == "Black Clover"]["Synopsis"].item()
labels = ["Brothers", "Magic"]
model_multitask_large = GLiNER.from_pretrained("knowledgator/gliner-multitask-large-v0.5")
model_type = "Multi Task Large Model"

_ = synopsis_NER(bcSynopsis, labels, model_type, model_multitask_large)

✓ 5.3s

Asking to truncate to max_length but no maximum length is provided and the model has no predefined maximum length. Default to no truncation.
Asta => Brothers
Yuno => Brothers
Wizard King => Magic
Black Clover => Magic
```

Figura 5.3: Instanciando el modelo de Gliner

Se ha implementado un sistema donde se vectorizan las relaciones de las etiquetas con otros *items* del *dataframe* que se ha obtenido.

En este caso de uso, hemos tomado una serie centrada en el fútbol, con un precisión exacta con **Captain Tsubasa**, conocida en España como **Oliver y Benji**.

Le sigue la serie de *volleyball Haikkyu* y otra serie moderna de fútbol llamada *Blue Lock*. A estas se le acompaña una serie sobre tenis, pero muestra anomalías con series que no están relacionadas con el deporte.

Use Case: Inazuma Eleven

```
main_anime_title = "Inazuma Eleven"
main_anime_synopsis = base_df[base_df["JP_Title"] == main_anime_title][["Synopsis"]].item()

model_type = "Multi Task Large Model"
model = model_multitask_large

labels = ["Soccer", "Sport", "Friendship"]

animes_wt_synopsis = {}

for row in range(0, base_df.shape[0]):
    title = base_df.iloc[row]["JP_Title"]
    synopsis = base_df.iloc[row][["Synopsis"]]

    animes_wt_synopsis[title] = synopsis

animes_more_related = recommender_system_NER(main_anime_title, main_anime_synopsis, animes_wt_synopsis, labels, model_type, model)

scores_sort = list(animes_more_related.keys())
scores_sort.sort(reverse=True)

for score in scores_sort:
    print(score, animes_more_related[score])
```

100.0 ['Captain Tsubasa']
66.66666666666666 ['Haikyuu!!', 'Blue Lock']
33.33333333333333 ['Tennis no Ouji-sama', 'Clannad', 'Ano Hi Mita Hana no Namae wo Bokutachi wa Mada Shiranai.', 'Black Clover']
0.0 ['Fullmetal Alchemist', 'Gintama', 'K-On!', 'Violet Evergarden']

Figura 5.4: Anotaciones en Gliner sobre series de fútbol

En un nuevo caso de uso, hemos tomado una serie dramática, la cual no realiza buenas relaciones a la hora de hacer un proceso similar al *Sentiment Analysis* recogiendo el tono.

Use Case: Violet Evergarden

```
main_anime_title = "Violet Evergarden"
main_anime_synopsis = base_df[base_df["JP_Title"] == main_anime_title][["Synopsis"]].item()

model_type = "Multi Task Large Model"
model = model_multitask_large

labels = ["Drama"]

animes_wt_synopsis = {}

for row in range(0, base_df.shape[0]):
    title = base_df.iloc[row]["JP_Title"]
    synopsis = base_df.iloc[row]["Synopsis"]

    animes_wt_synopsis[title] = synopsis

animes_more_related = recommender_system_NER(main_anime_title, main_anime_synopsis, animes_wt_synopsis, labels, model_type, model)

scores_sort = list(animes_more_related.keys())
scores_sort.sort(reverse=True)

for score in scores_sort:
    print(score, animes_more_related[score])
```

[38]

```
... The Great War => Drama
Captain Tsubasa => Drama
drama club => Drama
Light Music Club => Drama
Blue Lock => Drama
100.0 ['Captain Tsubasa', 'Clannad', 'K-On!', 'Blue Lock']
100.0 ['Tennis no Ouji-sama', 'Fullmetal Alchemist', 'Gintama', 'Inazuma Eleven', 'Ano Hi Mita Hana no Namae wo Mada Shiranai.', 'Haikyuu!!', 'Black Clover']
```

Figura 5.5: Anotaciones en Gliner referentes al tono

En el último caso de uso, se ha asignado la macro-etiqueta “*Shounen*” como serie de corte juvenil para adolescentes masculinos, a modo de género basado en la demografía clásica:

```

Use Case: K-On!

main_anime_title = "K-On!"
main_anime_synopsis = base_df[base_df["JP_Title"] == main_anime_title][["Synopsis"].item()]

model_type = "Multi Task Large Model"
model = model_multitask_large

labels = ["Shounen"]

animes_wt_synopsis = {}

for row in range(0, base_df.shape[0]):
    title = base_df.loc[row]["JP_Title"]
    synopsis = base_df.iloc[row]["Synopsis"]

    animes_wt_synopsis[title] = synopsis

animes_more_related = recommender_system_NER(main_anime_title, main_anime_synopsis, animes_wt_synopsis, labels, model_type, model)

scores_sort = list(animes_more_related.keys())
scores_sort.sort(reverse=True)

for score in scores_sort:
    print(score, animes_more_related[score])
]

+ Yuu Hirasawa => Shounen
Ryouma Echizen => Shounen
Gintoki Sakata => Shounen
Shinpachi Shimura => Shounen
Captain Tsubasa => Shounen
Tomoya Okazaki => Shounen
Yoshida Shouhan => Shounen
Inazuma Eleven => Shounen
Mamoru Endou => Shounen
Shuya Gouenji => Shounen
Jirata Yadedai => Shounen
Shouyou Hinata => Shounen
Little Giant => Shounen
Tobio Kageyama => Shounen
Violet Evergarden => Shounen
Auto Memory Doll => Shounen
Asta => Shounen
Yuno => Shounen
Lebuty => Shounen
Yozuchi Isagi => Shounen
106.0 ['Tennis no Ouji-sama', 'Gintama', 'Captain Tsubasa', 'Clannad', 'Inazuma Eleven', 'Ano Hi Mita Hana no Namae wo Bokutachi wa Mada Shiranai.', 'Haikyuu!!', 'Violet Evergarden', 'Black Clover', 'Blue Lock']
9.0 ['Fullmetal Alchemist']

```

Figura 5.6: Anotación en Gliner referente al género basado en demografía

La resolución que se podría extraer es que no recoge apreciaciones semánticas sobre el contenido.

Se necesitaría crear una ontología de etiquetado, además de *fine-tuning* de cada modelo pre-entrenado con su correspondiente *benchmarking*, requiriendo costes de entrenamiento para que el desarrollo pueda salir a producción.

5.5. Evaluación

Ya que no van a desarrollarse todas las soluciones debido al *scope* del proyecto se ha realizado una encuesta a nuestro focus group, cuyo *target* eran personas aficionadas al anime, para validar los resultados esperados con cada aproximación.

Cada técnica venía acompañada del mismo caso de uso expuesto anteriormente, donde las personas encuestadas daban su apreciación sobre si el enfoque era adecuado para sus necesidades.

Primer Caso de Uso - Named Entity Recognition (NER)

57 respuestas

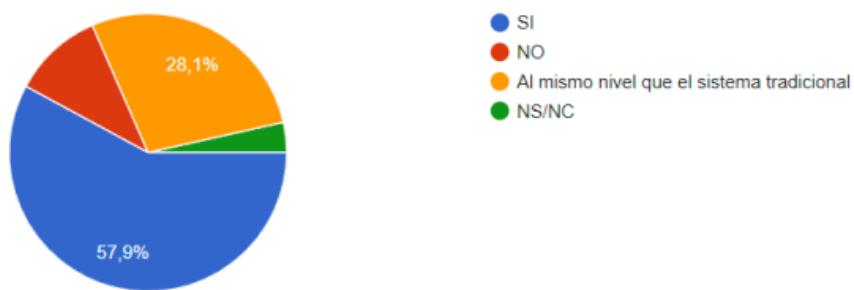


Figura 5.7: (a) Named Entity Recognition

Esta técnica es la que dio peores resultados, además de requerir gran coste de entrenamiento como se especificó en las hipótesis iniciales, debido a la jerarquía de etiquetas y a la adaptabilidad del modelo al caso de uso (**One Piece** (Oda, 1999)), tanto para usuarios como para clientes, por lo que se acepta la hipótesis.

Se podría argumentar en este punto que existe un problema ontológico, ya que el *target* de las series *mainstream* preferirían series conocidas que sean fáciles de consumir, antes que series desconocidas para la mayoría aunque compartieran temática.

Segundo Caso de Uso - Clusterización

57 respuestas

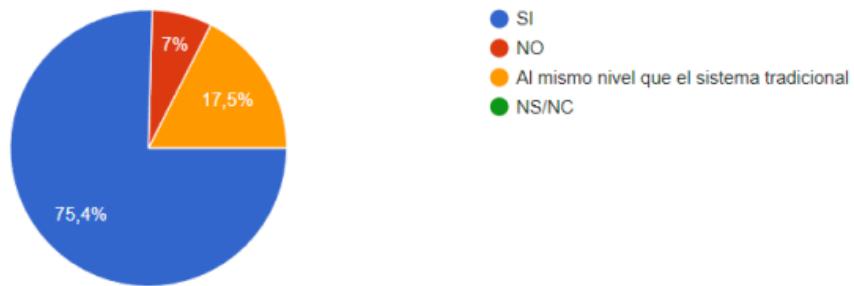


Figura 5.8: (b) Clusterización

El caso de uso (**Oliver & Benji** (Takahashi, 1983)) podría estar sesgado ya que es más sencillo recomendar una categoría de un deporte concreto, sin ofrecer matices. Para ser más certeros, se podría priorizar las series por su antigüedad ya que no tendrán el mismo estilo de animación, ni ritmo, por lo que se acepta la hipótesis.

Tercer caso de uso - Text Similarity

57 respuestas

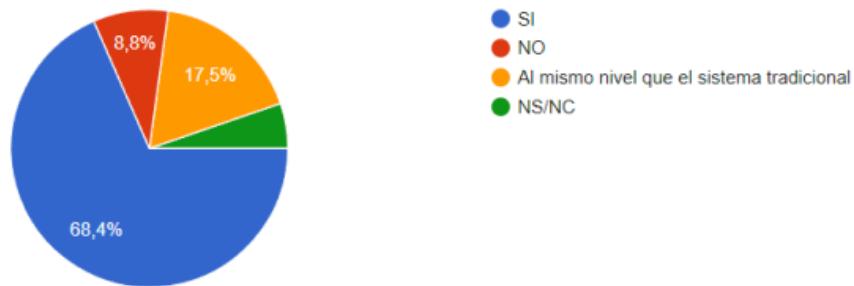


Figura 5.9: (c) Text Similarity

En este caso de uso (**Gintama** (Sorachi, 2003)) se daba una distinción temática y de tono al mismo tiempo, añadiendo matices en lo narrativo que pudo afectar a la valoración, por lo que se acepta la hipótesis a pesar de la falta de consenso.

Cuarto caso de uso - Topic Modeling

57 respuestas

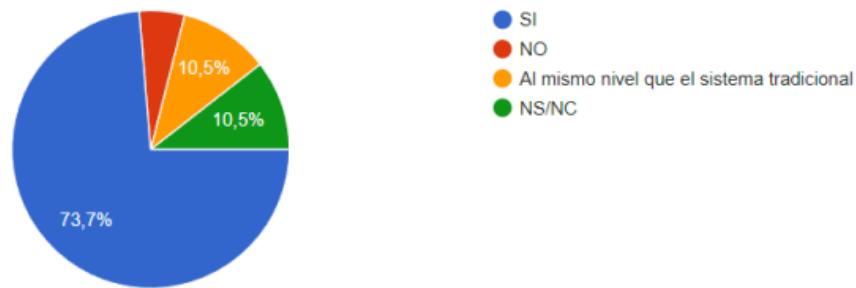


Figura 5.10: (d) Topic Modeling

Al priorizar el tono en el caso de uso (**K-ON!** (Yamada, 2009)), las personas encuestadas identificaron este caso de uso como la mejor opción posible, por lo que se acepta la hipótesis.

Como conclusión, existe un problema ontológico antes que técnico, sobre como jerarquizar la información.

Debido a que la mayoría tuvo resultados mayormente positivos, nos validan las premisas de haber detectado una necesidad en el sector, que esta solución cubre.

Capítulo 6

Sistema Predictivo del Impacto del Anime en Plataformas de Streaming

6.1. Identificación de Requisitos

Los *Sistema Predictivos* están diseñados para ofrecer explicaciones sobre cómo funcionan cierto tipo de eventos. Estos se utilizan para encontrar patrones y realizar predicciones de eventos que todavía no han ocurrido.

6.1.1. Enfoque actual

A día de hoy, el aumento de la cantidad de *animes* es cada vez más alarmante para las distribuidoras nacionales, ya que se necesita hacer un análisis exhaustivo por cada serie para determinar cuál debería licenciarse.

Esta cuestión es planteada por todas las empresas de *streaming* audiovisual, aunque en el mundo del *anime* cobra una mayor importancia.

Esto es debido a que en cada temporada, coincidente con las estaciones del año, se estrenan lotes de gran cantidad de *animes* que, a diferencia de otras plataformas de *streaming* audiovisual, dichos estrenos no suelen estar definido *a priori* en lotes.

Para que se contemple correctamente las cantidades en dichos lotes que se estrenan, pueden salir aproximadamente entre 30 o 60 *animés* por temporada de manera fragmentaria. Con lo cual, sumando cuatro temporadas habría más de 120-240 *animés* por año.

Por este motivo, no se dispone del tiempo ni de los recursos suficientes como para realizar un análisis en profundidad del impacto posible de las series lanzadas.

Otro motivo relevante asociado a este problema es el mero hecho de que, si en una temporada va a lanzarse un *anime* que acapara la atención de todo el público, muchos otros se invisibilizan por motivos de marketing, ya que no han contado con el suficiente respaldo mediático detrás.

Por último, a raíz del motivo anterior, las plataformas a nivel nacional como *Jonu Media y Selecta Visión* (GROUP, 2023; S.L.U., 2023) quedan eclipsadas por aquellas a nivel internacional como *Crunchyroll y Netflix* (Kun Gao, 2023; Reed Hastings, 2023), ya que las plataformas a nivel internacional se benefician de que tienen mayor impacto y más medios que las nacionales, por lo que es usual que los *animés* de impacto se encuentren en las plataformas internacionales.

Resumiendo, los objetivos a lograr serían parte de realizar un sistema que consiga:

- Gestionar el impacto de todos los *animés* que salen en una temporada.
- Controlar el impacto de los *animés* eclipsados por otros con mayor impacto.
- Beneficiar a las empresas nacionales frente a las internacionales, gracias al análisis del impacto realizado *a priori*.

Para dar solución a los objetivos propuestos se plantea construir una *Sistema Predictivo* que implemente *Machine Learning* para solventar el problema.

Las premisas iniciales parten de usar cuatro modelos de ***Machine Learning*** investigados, y posteriormente, combinarlos para poder realizar mejores clasificaciones(IAT, 2024)

Los modelos investigados son los siguientes:

1. **Modelo Lógico**, modelos basados en técnicas de captura de información en base a la aplicación de ciertas reglas de actuación, como son los Árboles de Decisión.
2. **Modelo Geométrico**, modelos basados en técnicas de clasificación según el espacio o la distancia, como son la Regresión Logística o la Máquina de Soporte Vectorial.
3. **Modelo Probabilístico**, modelos basados en el uso de distribuciones de probabilidad en los datos usando estadística bayesiana, como son las Redes Bayesianas de Naive Bayes.
4. **Modelo Mixto**, modelos que implementan la lógica de todos los modelos, como son las Redes Neuronales.

Por último, se ha implementado la técnica de ***ensemble*** para crear una familia de algoritmos que serán capaces de decidir si un ***anime*** será de impacto y con qué probabilidad, mediante Clasificación Binaria.

Como parte de la encuesta de validación se ha preguntado sobre los hábitos de consumo de series en plataformas de *streaming*. Con todo, este análisis de impacto se vislumbrará si es relevante para los usuarios integrar la solución en determinadas plataformas, cubriendo las necesidades mercadotécnicas de la misma.

Finalmente, se plantean una serie de hipótesis que se validarán o se rechazarán en las conclusiones:

- La **Regresión Logística** como técnica que mejor predice el comportamiento de las etiquetas, en cómputo global, debido a la linealidad observada en los *animés* y sus puntuaciones asociadas.
- Los **Árboles de Decisión** como técnica que peor predice el comportamiento de las etiquetas, en cómputo global, debido a que se basan en Modelos Lógicos y no se encuentran reglas definidas a priori en los datos.
- El uso de **Ensembles** como mejor técnica para el caso de uso, debido a que es la técnica más potente computacionalmente y, es posible usar de varias formas distintas (diferente forma de agregación) para predecir el resultado final como *voting, stacking, bagging y/o boosting*. (Gozhulovskyi, 2023)

6.2. Descripción de la herramienta desarrollada

En esta sección abordaremos todos los puntos que engloben la secuencia de ejecución del algoritmo, profundizando en cada uno de los puntos.

6.2.1. Obtención y Preprocesamiento de los Datos

Para plantear el uso de los algoritmos de *Machine Learning* de esta sección, se ha partido de la idea de construir un *dataframe* asociando cada *anime* a una serie de métricas, tanto cualitativas como cuantitativas, que sirvan al modelo para predecir la variable target que sería si un *anime* será de impacto para ser licenciado en una plataforma de *streaming*.

Para ello, se ha planteado construir dicho *dataframe* a partir de una serie de fuentes de información:

- **Base de Datos de Anime *MyAnimeList*** (Gyssler, 2023): Dicha fuente, al igual que en el algoritmo anterior, nos muestra por cada *anime* sus datos asociados con métricas como pueden ser, entre otras:

- La nota puntuada por el público.
- La posición en el *ranking* general según la puntuación del público.
- El número de personas que tienen en su registro de *animes* dicha serie.
- El género, el tema o la demografía asociada al *anime*.

En la figura 4.2 presentada en el apartado de Proceso ETL se encuentra una versión más extensa de todas las variables usadas por este *dataframes*.

Además, como planteamos usar esta herramienta para predecir *animés* próximos a estrenar y, por tanto, no tienen aportaciones de los usuarios en la página relacionadas con la puntuación o el *ranking*, se ha agregado la puntuación de las adaptaciones del *anime* como pueden ser mangas o novelas ligeras del mismo realizados en el pasado, y que cuentan con métricas que serán similares. Por ejemplo, puntuación asociada al *manga* relacionado con la serie en cuestión.

- **Variable Target:** Dicha variable se construye en base a compr si el *anime* está licenciado en alguna plataforma de las siguientes *Crunchyroll, Netflix, Selecta Visión y Jonu Media* (GROUP, 2023; Kun Gao, 2023; Reed Hastings, 2023; S.L.U., 2023)

En conclusión, partiremos de un *dataframe* que contiene, por cada fila, el *anime* junto con una serie de métricas asociadas al mismo y a sus adaptaciones. Además de contar con la variable “Licensed” que indica si está el *anime* licenciado en alguna plataforma de las mencionadas.

Para este *dataframe* hay que tener en cuenta dos aspectos:

1. Generalmente, un *anime* que tiene varias temporadas solo tomará la primera, ya que cuándo una plataforma licencia un *anime* no solo licencia la primera temporada, sino todas las posteriores.
2. Un *anime* que tiene varias adaptaciones aparecerá en varias filas según el número de adaptaciones que tenga.

6.2.2. *Exploratory Data Analysis (EDA): Análisis Exploratorio de los Datos*

Tras la obtención de los datos, el primer paso ha sido el análisis de los datos para poder investigar el comportamiento de los datos como pueden ser la linealidad y distribución de los datos y la detección e interpretación de los datos nulos, entre otros.

Para ello, se han investigado los siguientes apartados:

1. **Calidad del dato**, comprobación de los datos obtenidos a partir del proceso ETL mencionado con anterioridad.

```
In the obtained DF, it is observed that it exist different forms of writing Not Assigned (NA)

print("NA interpreted by float64 variable: {}".format(base_df["JP_Title"] == "Kuro no Danshou")[0]) # NA interpreted by float64 variable -> nan
print("NA interpreted by list of strings: {}".format(base_df[base_df["JP_Title"] == "Cowboy Bebop"]["Score"].values)) # NA interpreted by list of strings -> []
print("'' interpreted by list of strings: {}".format(base_df[base_df["EN_Title"] == "Naruto"][[["EN_Title"].values]]) # '' interpreted by list of strings -> []
print("N/A interpreted by float64 variable: {}".format(base_df[base_df["JP_Title"] == "Mahou Shoujo Lyrical Nanoha"][[["Score_source"].values][0]])) # N/A interpreted by list of strings -> nan

NA Interpreted by float64 variable: nan
NA interpreted by list of strings: []
'' interpreted by list of strings: []
N/A interpreted by float64 variable: nan
```

Figura 6.1: Interpretación de los datos nulos

2. **Datos Nulos**, exploración acerca de la interpretación de los datos nulos en el **dataframe** como se observa en la Figura 6.1 donde se examina el comportamiento de cada variable siendo todas nulas.
3. **Comprensión y Tratamiento de valores nulos**, existe una necesidad de entender que significa que un dato sea nulo.

Para ello, obtenemos que hay cuatro columnas que contienen datos nulos que son: “Score”, “Ranked”, “Score_source” y “Ranked_source”. En este caso, el par de variables que son coherentes con que puedan ser nulos es “Score” y “Ranked”, ya que para un **anime** que no se ha estrenado, no es posible calificarlo en la página, y, por tanto, ponerle un *ranking*.

Para hacer este tratamiento de los datos, se realiza lo siguiente:

- a) Eliminación de aquellos *animes* que tienen el valor nulo en “Score_source” o en “Ranked_source”.
- b) Eliminación de aquellos animes que tengan valores nulos en “Score” o en “Ranked” y que hayan sido lanzados.

¿Qué se realiza con los valores restantes nulos? Se seguirá con el análisis de los datos y en un paso posterior, se estudiará el caso.

4. **Distribución de las puntuaciones:** una cuestión surgida en la investigación ha sido el hecho de saber cómo se distribuyen las puntuaciones de los usuarios con respecto a la adaptación asociadas a los animes, es decir, saber si la media de las puntuaciones normales de los *animes* y las adaptaciones es similar.

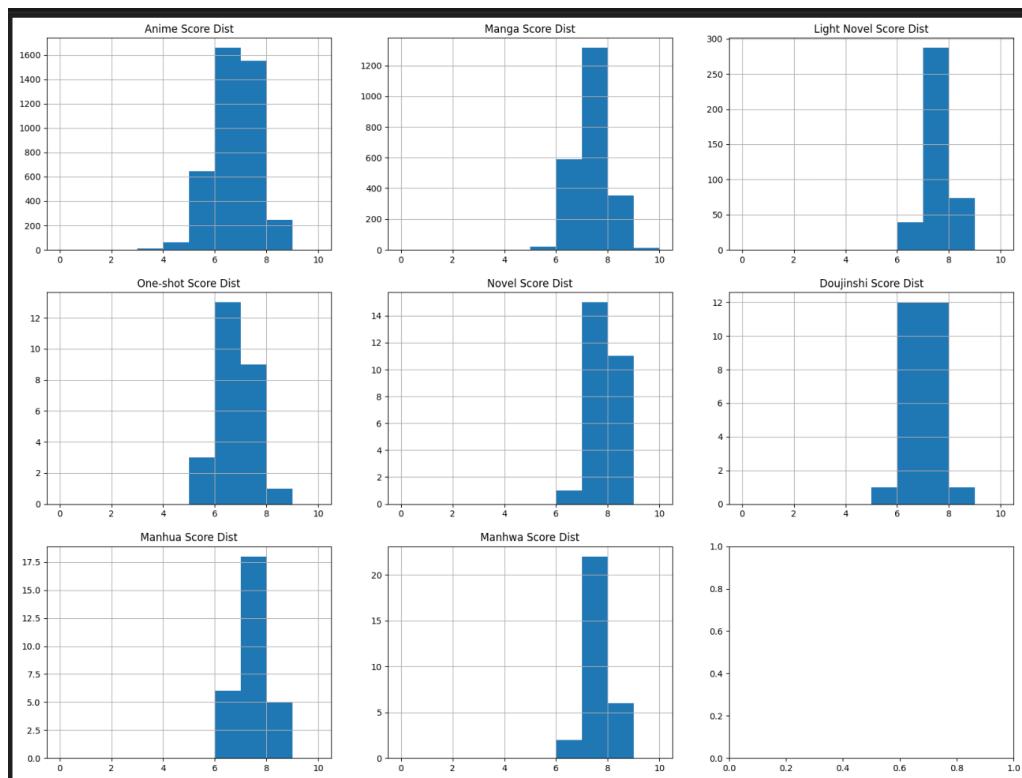


Figura 6.2: Distribución de notas (score) según el tipo de source

¿Qué información nos proporciona?

Se observa en la Figura 6.2 como tanto la puntuación de los *animés*, como todas las adaptaciones asociadas, tienen una media y varianza similar y, por tanto, siguen una distribución normal o Gaussiana.

Además, esto nos ayuda a saber que transformación poder utilizar para los datos nulos, pudiéndose usar media, mediana o moda ya que no se presenta ni una significante curtosis, ni un sesgo en la distribución de los datos.

5. **Linealidad en las variables:** Se ha estudiado la existencia de linealidad entre todas las variables para reconocer linealidad entre los datos.

Para esto, se ha utilizado conceptos estadísticos acerca de:

- a) Probabilidad.
- b) Coeficientes de Pearson y Spearman.
- c) Matriz de correlación estadística.

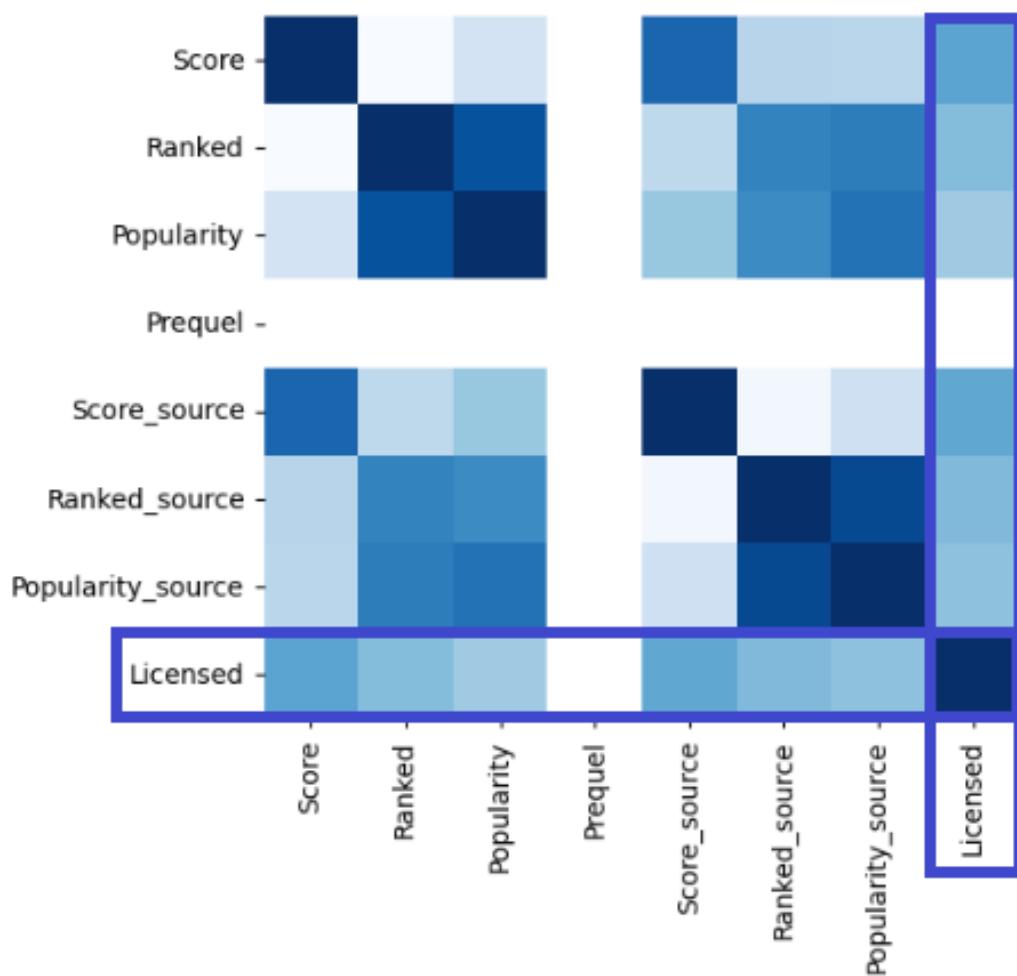


Figura 6.3: Matriz de Correlación de Probabilidades

¿Qué información nos proporciona?

Para interpretar la matriz de correlación de la Figura 6.3 se tiene que prestar atención a la variable “Licensed” resaltada la cual va a ser la que predigamos. Además, gracias al uso de colores utilizado en la matriz, podemos ver claramente cuales variables son más o menos linealmente dependientes a “Licensed”, lo cual ‘se utilizará de apoyo en el momento de usar las técnicas y los modelos vistos.

6. Sustitución de valores nulos:

Por último, para poder usar la variable Score a la hora de predecir para realizar la clasificación, se ha predicho de la siguiente manera.

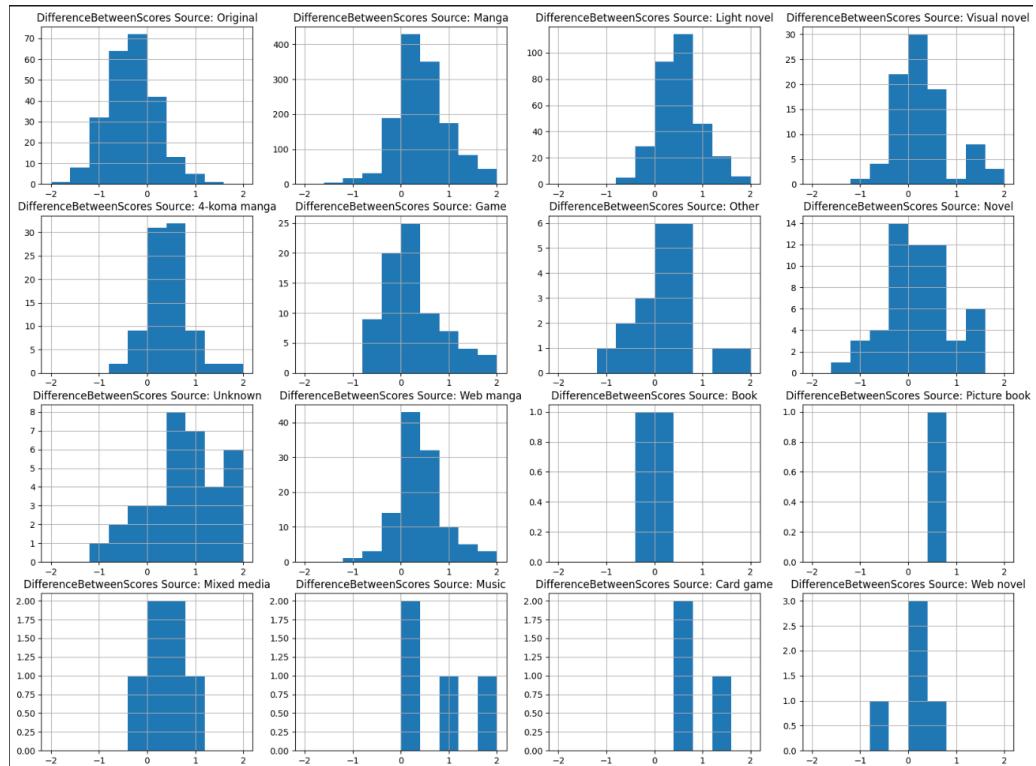


Figura 6.4: Distribución de los cambios en las notas entre la nota de un anime y su adaptación según los usuarios

En la Figura 6.4 se han obtenido las distribuciones de las diferencias de notas entre los *animes* y sus adaptaciones. Todo este estudio ha sido con el fin de reconocer si hay algún indicio de que, si un *anime* tiene un tipo de adaptación en específico, tiene más posibilidad de ser de éxito, pero analizando la imagen se concluye, tal y como se vio en la Figura 6.2, que también siguen distribuciones normales con medias y varianzas similares, por lo tanto, no debería afectar significativamente.

7. Comentarios adicionales:

Aunque se han descritos y desarrollados seis puntos en el EDA, se han investigado otros puntos que no se han llevado a la práctica para el estudio de los datos como son:

- **Gráfico de Dispersión:** Realizar un gráfico de los datos en un plano cartesiano puede revelar visualmente si la relación es lineal, no lineal o si no hay una relación aparente.
- **Gráfico de Residuos:** Despues de ajustar un modelo lineal, es posible mostrar los residuos/errores en el plano. Si los residuos están dispersos aleatoriamente alrededor del eje horizontal, es probable que la relación sea lineal, si se encuentran patrones en los residuos sugieren no linealidad.
- **Pruebas de Linealidad (Prueba RESET o de Falta de Ajuste):**
(1) Realizar una prueba general para la especificación del modelo, incluyendo la linealidad o (2) realizar una comparación de un modelo con una versión más compleja para ver si un modelo más simple (lineal) es adecuado.
- **Regresión Polinómica:** Comparar el ajuste de varios modelos polinómicos como son los lineales, cuadráticos, cúbicos, entre ellos.
- **Regresión Lineal por Tramos:** Identificar puntos donde la relación cambia y ajustar modelos lineales separados a cada segmento.
- **Regresión por Splines:** Ajustar un modelo de regresión por splines y examinar la curva suave que captura la relación.
- **Regresión Loess (Regresión Local):** Ajustar un modelo loess para

capturar relaciones complejas.

- **Transformación de Variables:** Transformar una o ambas variables y reevaluar su relación, como pueden ser las técnicas de PCA.
- **Función de Autocorrelación (ACF) y Función de Autocorrelación Parcial (PACF):** Aunque esta técnica no aplicaría ya que no se trata de una serie de tiempo, se trataría de examinar los gráficos ACF y PACF para detectar dependencias lineales o no lineales en series de tiempo.
- **Tau de Kendall y Rho de Spearman:** Comprobar estas estadísticas para poder capturar relaciones monótonas (que pueden ser no lineales).
- **Información Mutua:** Calcular la información mutua entre las variables para detectar cualquier tipo de dependencia.

6.2.3. *Feature Engineering: Creación de Atributos*

Tras haber realizado el análisis de los datos, el segundo paso es obtener más información a partir de los propios datos utilizando el proceso de *Feature Engineering*.

Para dicho proceso se ha investigado una lista de posibles pasos a seguir en este proceso (Rençberoğlu, 2024) los cuales son:

```
[402]: display(final_df[null_list].head(5))

...
EN_Title      2307
Premiered     2086
Score         313
Ranked        740
Score_source  1524
Ranked_source 595
dtype: int64
['EN_Title', 'Premiered', 'Score', 'Ranked', 'Score_source', 'Ranked_source']

...
   EN_Title Premiered Score Ranked Score_source Ranked_source
0      NaN     Fall 2002  7.55  1566.0       7.06      7698.0
1  Beet the Vandell Buster     Fall 2004  6.93  4284.0       7.18      6032.0
2  Honey and Clover     Spring 2005  8.00  597.0        8.28      324.0
3          NaN     Spring 2004  8.87   25.0        9.15       5.0
4      NaN     Fall 2002  7.99  607.0       8.07      603.0

D ▾
final_df["EN_Title"] = final_df["EN_Title"].apply(lambda x: "-" if pd.isna(x) else x)
final_df["Premiered"] = final_df["Premiered"].apply(lambda x: "-" if pd.isna(x) else x)

# Compute mean scores for each type
type_mean_scores = final_df.groupby("Type")["Score"].mean()
type_mean_ranked = final_df.groupby("Type")["Ranked"].mean()
type_mean_scores_src = final_df.groupby("Type Source")["Score_source"].mean()
type_mean_ranked_src = final_df.groupby("Type Source")["Ranked_source"].mean()

final_df["Score"] = final_df.apply(lambda x: type_mean_scores[x["Type"]] if pd.isna(x["Score"]) else x["Score"], axis=1)
final_df["Ranked"] = final_df.apply(lambda x: type_mean_ranked[x["Type"]] if pd.isna(x["Ranked"]) else x["Ranked"], axis=1)
final_df["Score_source"] = final_df.apply(lambda x: type_mean_scores_src[x["Type Source"]] if pd.isna(x["Score_source"]) else x["Score_source"], axis=1)
final_df["Ranked_source"] = final_df.apply(lambda x: type_mean_ranked_src[x["Type Source"]] if pd.isna(x["Ranked_source"]) else x["Ranked_source"], axis=1)

display(final_df)
```

Figura 6.5: Imputación de los valores nulos de las columnas por la media de sus columnas correspondientes

1. **Imputación:** eliminación o imputación de los datos nulos usando un umbral.

Para ello, se ha realizado una imputación de los valores nulos por la media (debido a lo que asumimos de los punto 4 y 6 del proceso EDA) como se observa en la Figura 6.5.

2. **Manejo y Eliminación de Valores Atípicos:** detección los valores atípicos

y verificar si vale la pena eliminarlos o limitarlos usando los Percentiles. En este caso, no se ha planteado utilizar esta técnica ya que no existen gran cantidad de *outliers* en los datos.

```

final_df["Score_Bin"] = [1 if ( score <= 6.5 ) else # 1 = Low
                        2 if ( ( 6.5 < score ) and ( score <= 7.5 ) ) else # 2 = Medium-Low
                        3 if ( ( 7.5 < score ) and ( score <= 8.5 ) ) else # 3 = Medium-High
                        4 if (8.5 < score) else # 4 = High
                        0 # 0 = NA
                        for score in final_df["Score"]]

print(final_df["Ranked"].min(), final_df["Ranked"].max())
# Numbers are between 2 and 20028
final_df["Ranked_Bin"] = [1 if ( rank <= 5000 ) else # 1 = High
                        2 if ( ( 5000 < rank ) and ( rank <= 10000 ) ) else # 2 = Medium-High
                        3 if ( ( 10000 < rank ) and ( rank <= 15000 ) ) else # 3 = Medium-Low
                        4 if (15000 < rank) else # 4 = Low
                        0 # 0 = NA
                        for rank in final_df["Ranked"]]

0 20028.0

print(final_df["Popularity"].min(), final_df["Popularity"].max())
# Numbers are between 1 and 22809
final_df["Popularity_Bin"] = [1 if ( pop <= 5000 ) else # 1 = High
                             2 if ( ( 5000 < pop ) and ( pop <= 10000 ) ) else # 2 = Medium-High
                             3 if ( ( 10000 < pop ) and ( pop <= 15000 ) ) else # 3 = Medium-Low
                             4 if (15000 < pop) else # 4 = Low
                             0 # 0 = NA
                             for pop in final_df["Popularity"]]

22809

print(final_df["Members"].min(), final_df["Members"].max())
# Numbers are between 1 and 22809
final_df["Members_Bin"] = [1 if ( pop <= 250000 ) else # 1 = Low
                           2 if ( ( 250000 < pop ) and ( pop <= 1500000 ) ) else # 2 = Medium-Low
                           3 if ( ( 1500000 < pop ) and ( pop <= 3000000 ) ) else # 3 = Medium-High
                           4 if (3000000 < pop) else # 4 = High
                           0 # 0 = NA
                           for pop in final_df["Members"]]

```

Figura 6.6: Agrupamiento de los valores por contenedores conforme a la puntuación y el *ranking*

3. **Binning:** agrupación de los valores en categorías ayuda al modelo a no tener que procesar todos los datos de manera continua, sino a procesarlos de forma discreta, mejorando el tiempo de ejecución y el descubrimiento de patrones de la técnica usada, tal y como se plantea en la Figura 6.6.
4. **Transformación Logarítmica:** transformación que ayuda a manejar datos sesgados cambiando la distribución a una más aproximada a la normal, disminuyendo el efecto de los valores atípicos.



User	City
1	Roma
2	Madrid
1	Madrid
3	Istanbul
2	Istanbul
1	Istanbul
1	Roma

User	Istanbul	Madrid
1	0	0
2	0	1
1	0	1
3	1	0
2	1	0
1	1	0
1	0	0

Figura 6.7: Ejemplo de codificación *One-Hot Encoding* (Rençberoglu, 2024)

5. **One-Hot Encoding:** Cambio de los datos categóricos, que son difíciles de entender para los algoritmos, a un formato numérico que permite agrupar los datos categóricos sin perder información como se observa en el ejemplo de la Figura 6.7.
6. **Operaciones de Agrupación:** Agrupación de varias filas en una, ya que no son instancias únicas. En este caso, no existen filas agrupadas ya que por cada fila está agrupada por la serie en cuestión, por lo que no realizaremos este paso.
7. **División de Características:** Obtención de nuevas columnas a partir de dividir los valores de una columna en subcadenas, creando así una cantidad de columnas equivalente al número de subcadenas creado. En este caso, la columna que se utilizaría sería “*Synopsis*” para aplicar esta etapa, pero se generaría tanta cantidad de columnas que lo hemos descartado como posible paso.
8. **Escalado:** Transformación de las variables cuantitativas mediante una normalización de los datos para que todas las variables tengan el mismo peso, ya que puede ocurrir que no todas las variables tengan el mismo rango de valores. En este caso se ha utilizado la normalización estándar (*Standardization*) ya que reduce la problemática de los *outliers* los cuales no se han tratado previamente.
9. **Extracción de Fecha:** Obtención de nuevas columnas en base a separar fechas en columnas relacionadas con los años, meses y días de la propia fecha presentando el mismo concepto que en punto 7. En este caso no se tiene en cuenta ya que no trabajamos con columnas de tipo fecha.

6.2.4. Construcción y ejecución de modelos y técnicas

Una vez realizado el proceso de *Exploratory Data Analysis* y *Feature Engineering*, se va a proceder a explicar todos los pasos investigados y/o implementados.

Preparación de Datos

Antes de comenzar, se ha separado el *dataframe* resultante de realizar ambos procesos anteriores en dos conjuntos de datos. Un conjunto de datos llamado “*non_predict_df*” que contiene aquellos *animes* que ya se han estrenado y tenemos la información de si están licenciados en las plataformas en la columna “*Licensed*” representado con los valores 1 o 0, y otro conjunto de datos llamado “*predict_df*” que contiene aquellos *animes* que no se han estrenado los cuales tienen como valor -1 en la columna “*Licensed*” para diferenciarlos.

Debido a cuestiones de tiempo, ya que dichos *animes* no se han estrenado todavía, no se ha podido realizar una posterior validación de las técnicas implementadas, por lo que trabajaremos con el *dataframe* llamado “*non_predict_df*” para implementar las técnicas en los modelos.

Por último, dicho *dataframe* se ha mezclado y dividido en un conjunto de entrenamiento (*training*), un conjunto de validación (*validation*) y un conjunto de prueba (*test*) usando una proporción de 80 %, 10 % y 10 % respectivamente.

Clasificación Desbalanceada (Imbalanced Classification)

Además, una vez que se ha dividido los conjuntos de datos, se percibe como el conjunto de *training* utilizado para realizar el entrenamiento de las técnicas no está balanceado.

¿Qué significa que no está balanceado? La gran mayoría de los registros del *dataframe* están etiquetados como 0. Este problema viene causado ya que la mayoría de los *animes* existentes no están licenciados en las plataformas de *streaming*.

```
Training set class distribution:  
Licensed  
0    2772  
1    774  
Name: count, dtype: int64  
Resampled training set class distribution:  
Licensed  
1    2772  
0    2772
```

Figura 6.8: Ejemplo de uso de SMOTE con el *Dataset* de entrenamiento

La solución para este problema es usar la clase SMOTE (*Synthetic Minority Over-sampling Technique*) del *package Imbalanced Learn* para realizar un muestreo (*sampling*) sobre el valor minoritario 1 y crear nuevos registros con respecto a ese valor como se observa en la Figura 6.8.

Esta técnica se llama *over-sampling* y se basa en crear muestras sintéticas gracias al uso de la técnica de los *K-Neighbors*.

Explicación de la Bondad del Ajuste

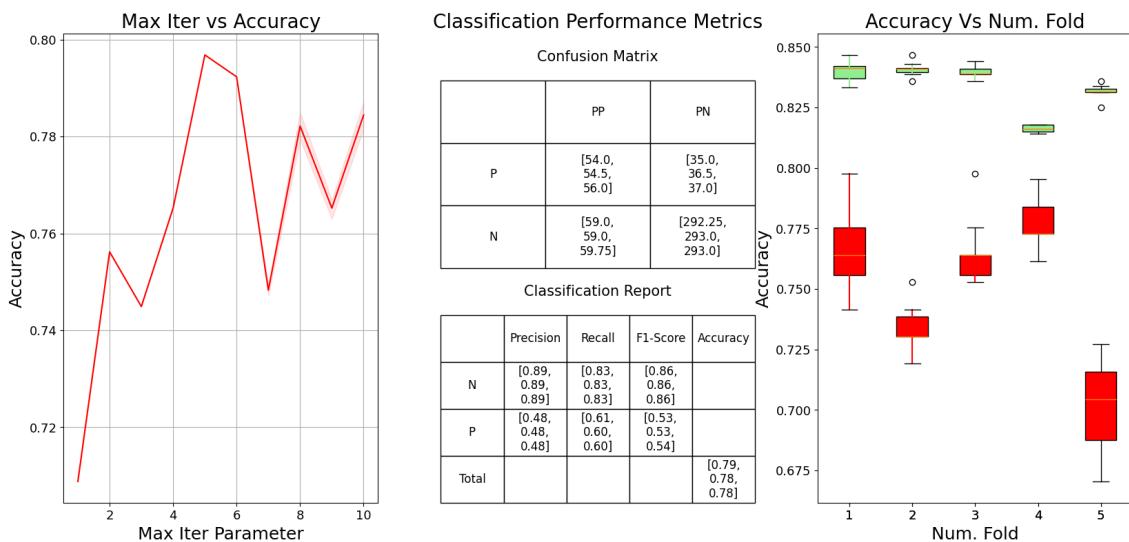
Antes de mostrar las técnicas de los modelos vistos al comienzo de la sección, se va a explicar como se han comparado los resultados entre los modelos ejecutados posibles.

Para ello, se ha procedido a desarrollar una función llamada *evaluate_model*:

```
1 def evaluate_model(technique_type, X, y, seed_data=None, folds=5,  
2     max_iters=20, number_of_execs=20, jobs=None):
```

Dicha función acepta los siguientes parámetros:

- ***technique_type***: Cadena de texto que especifica que tipo de técnica que se va a usar, por ejemplo, “*Decision Tree (Vanilla)*” para usar el clasificador *DecisionTree* usando los hipérparámetros por defecto.
- ***X***: *dataframe* “*non_predict_df*” sin la variable “*Licensed*”.
- ***y***: *dataframe* “*non_predict_df*”, pero solo con la variable “*Licensed*”.
- ***seed_data***: Argumento agregado para realizar la mezcla de los datos de manera aleatoria, pero siempre igual.
- ***folds***: Argumento agregado para indicar el número de pliegues que queremos implementar en el uso de la técnica de *K-Folds* siendo este dato el valor de K.
- ***max_iters***: Argumento agregado para indicar el número de iteraciones máximas del entrenamiento.
- ***number_of_execs***: Argumento que indica el número de ejecuciones donde se crean distintos modelos con *seed* distintas.
- ***jobs***: Argumento que indica el número de procesadores que se pueden utilizar para ejecutar los modelos.



Decision Tree (Vanilla) with 10 different seeds

Figura 6.9: Ejemplo de resultado de la ejecución de la función evaluate_model

¿Qué se observa en cada gráfico de la Figura 6.9?

- En el lado izquierdo, el gráfico denominado “*Max Iter vs Accuracy*” muestra un gráfico que compara el número de iteraciones máximo con la Exactitud del conjunto de modelos.

Se está representando una línea que indica la mediana (Rango intercuantil) de todos las máximas iteraciones de todos los num_of_execs modelos, por ejemplo, el valor (2, 0,748) representa la mediana del conjunto de accuracies obtenido de ejecutar 10 modelos Decision Tree con maxIter = 2 (Median[acc1, acc2, ..., ac10]).

Además, también se muestra una zona más transparente donde se observa el mismo planteamiento que el gráfico de línea pero para el rango intercuantílico entre el 25 % y el 75 %.

- En el medio, el gráfico denominado “*Classification Performance Metrics*” se observan dos tablas, la primera se corresponde con la matriz de confusión y la

segunda con el reporte de la clasificación.

En la matriz de confusión se observa como, en vez de existir un valor correspondiente a cada celda, hay 3. Estos valores corresponden a ejecutar num_of_execs modelos con el número máximo de iteraciones, por ejemplo, el valor de la celda verdadero positivo (*True Positive*) del 25 % se obtiene capturando el valor del percentil 25 % de los 10 valores TP de los 10 modelos ejecutados, y de la misma forma para los demás valores.

El reporte es calculado manualmente en base a los resultados de la tabla de Confusion Matrix, es decir, para calcular el valor 25 % de Precisión, se obtiene el valor del 25 % de los Verdaderos Positivos y del 25 % de los Falsos Positivos, y se calcula mediante la fórmula pertinente.

- En el lado derecho, vemos un gráfico que nos muestra si existe *overfitting* o *underfitting* en los datos. Cada punto indica, con un gráfico de caja y bigotes, la distribución de las *accuracies* por cada *fold* de los conjuntos de *training* y *validation* de todos los modelos, es decir, la caja roja en el *fold* 1 indica la distribución de los *accuracies* de los 10 modelos para el conjunto de *validating* en la técnica de validación cruzada. En el caso de las cajas verdes, es el mismo planteamiento pero para el conjunto de datos de *training*.

Introducción a la comparación y evaluación de Modelos y Técnicas

Como comienzo de esta sección, recordamos los modelos introducidos en el desarrollo que han servido como base para la investigación y ejecución de las técnicas pertinentes (IAT, 2024).

Los modelos investigados son los siguientes:

1. **Modelo Lógico:** modelos basados en técnicas de captura de información en base a la aplicación de ciertas reglas de actuación.
2. **Modelo Geométrico:** modelos basados en técnicas de clasificación según el espacio o la distancia.
3. **Modelo Probabilístico:** modelos basados en el uso de distribuciones de probabilidad en los datos usando estadística bayesiana.
4. **Modelo Mixto:** modelos que implementan la lógica de todos los modelos.

Una vez presentados, por cada modelo, se indicará la investigación realizada entorno al modelo y una ejecución relacionada con alguna técnica que implementa dicho modelo (Saradakshmi8074, 2024).

Modelo Lógico

Los modelos lógicos son aquellos que utilizan para entrenar reglas obtenidas en base a los datos similar al uso de las sentencias condicionales en programación denominadas “if-else” o “switch-case”.

Al igual que ocurre, cuando se analiza la traza de un código que tiene una sentencia condicional, que se despliegan n-ramas por cada condición, con estos modelos ocurre exactamente lo mismo.

Además, estos modelos, al seguir la lógica de desplegarse en ramas, se les reconoce como *Feature Trees*. Si estos árboles tienen como finalidad la clasificación de etiquetas se les conoce como Árboles de Decisión.

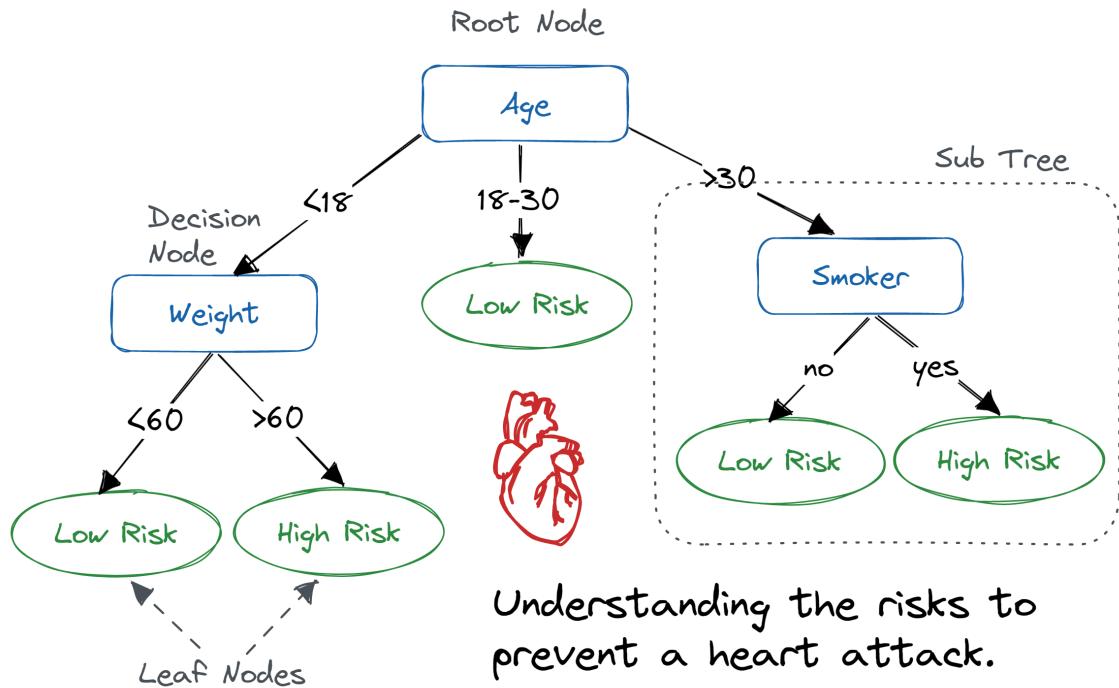


Figura 6.10: Ejemplo de funcionamiento de un árbol de decisión (Awan, 2024)

El objetivo, como se observa en la Figura 6.10 es comenzar analizando una de las columnas (*Root Node*) y partiendo del árbol generado de reglas, ir reduciendo el árbol en subárboles llamados *hyperrectangles* o *Instance Space Segments* para ir bajando en profundidad el árbol hasta llegar a las hojas con el resultado de la predicción.

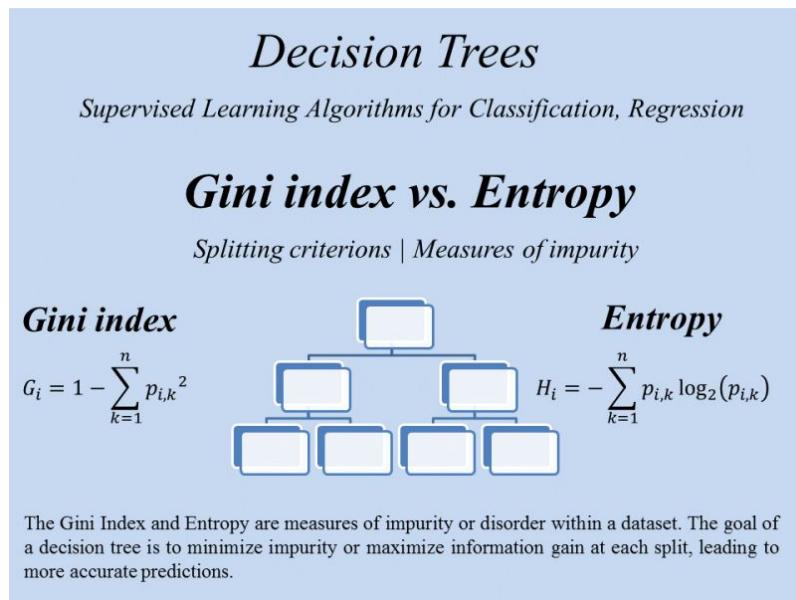
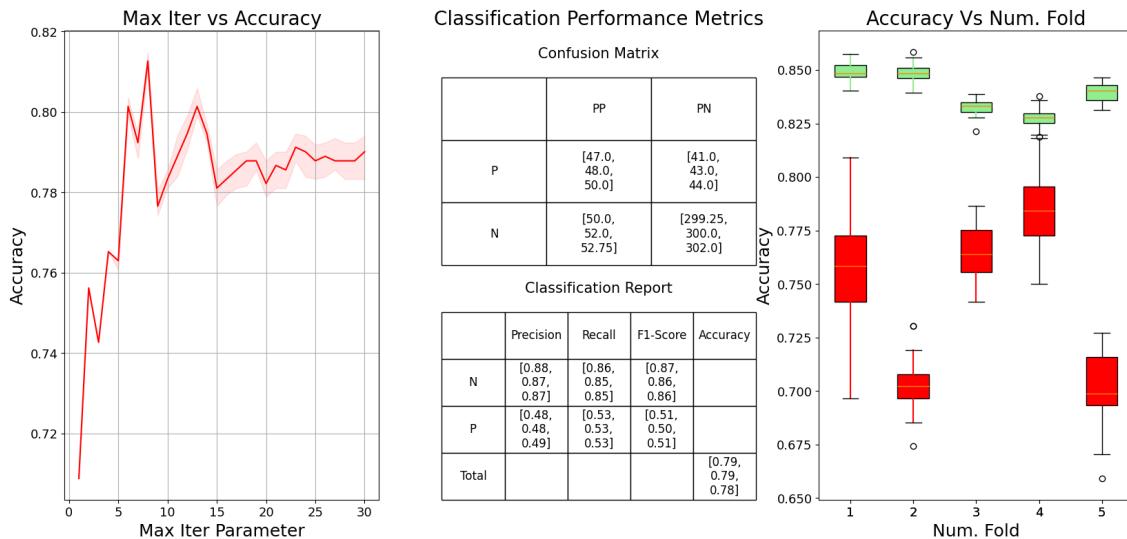


Figura 6.11: Métricas que utiliza un árbol convencional vs. árbol ID3 (Mehdi Hamedi, 2024)

Dicha explicación sirve para explicar la funcionalidad de los árboles más básicos como la técnica *DecisionTreeClassifier* de **Scikit-Learn**, pero existen modelos más complejos como el modelo ID3 (*Iterative Dichotomiser 3*) que parten de conceptos de la entropía de los datos (cómo están distribuidos los datos) y la ganancia (Index Gini) de dichas entropías con referencia a la entropía de la columna a predecir, observándose dicho cálculo en la Figura 6.11.

Una vez explicado el modelo, se ha procedido a ejecutar el clasificador *DecisionTreeClassifier* de **Scikit-Learn** con todos los hiperparámetros por defecto, salvo el número de iteraciones máxima (*max_depth parameter*), la seed utilizada para el modelo (*random_seed*) y el número de procesadores que puede utilizar a la vez (*n_jobs*), por lo que se ha contemplado el uso de este árbol como “Vanilla” o sin hiperparametrizar.



Decision Tree (Vanilla) with 30 different seeds

Figura 6.12: Ejecución del clasificador (*DecisionTreeClassifier*) con 30 ejecuciones con distintas *seeds* y con 30 iteraciones máximas

Una vez explicado como interpretar el gráfico en el punto anterior se pueden sacar varias conclusiones de esta ejecución observando la figura 6.12

- **Max Iter vs Accuracy:** La mediana de las exactitudes se estabiliza entorno a la Iteración 30, por lo que no convendría “sobreentrenar” con más iteraciones el clasificador ya que no va a clasificar mejor pese a utilizar más iteraciones.

Además, es posible que si utilizamos más iteraciones en el clasificador funcione peor, por lo que es relevante utilizar esta curva para el estudio del clasificador.

Por último, se podría limitar el modelo a la iteración 10 que es el pico donde más exactitud se observa y donde comienza a variar el rango intercuantil de exactitudes.

- **Classification Performance Metrics:** La mayoría de los valores predichos con respecto a los reales entre los tres cuartiles no tienen una gran varianza entre ellos, lo que resulta en que todos los modelos clasifican de manera similar.

Ocurre de manera similar en el reporte de la clasificación donde los valores de Precisión (*Precision*), Exhaustividad (*Recall*) y F1-Score no varian entre ellos.

- **Accuracy vs Num Fold:** Se presenta un sobreajuste en los datos tras utilizar la técnica de los *Kfolds* ya que, para cada *fold*, todas las exactitudes del conjunto de validación (rojo) usado están por debajo del conjunto de entrenamiento (verde), es decir, independientemente de la ejecución del clasificador usada, siempre se produce mejor predicción en el conjunto de entrenamiento (verde) que en el conjunto de validación (rojo).

Además, se observa como, para los *folds* del conjunto de entrenamiento, es muy poco disperso, pero para los *folds* del conjunto de validación, son bastante dispersos llegando a variar hasta un 10% en la Exactitud.

- **Conclusión final:** Aunque se presenta una muy buena exactitud general (78%), no se puede olvidar el objetivo principal que es que clasificar con certeza los *animales* que van a ser licenciados, es decir, que tengan etiqueta 1 (Positiva), por lo que priorizamos los resultados del Reporte de la Clasificación correspondiente a la etiqueta Positiva, que en este caso, no llegan al 50% por lo que descartamos usar este modelo.

Modelo Geométrico

Los modelos geométricos son aquellos que utilizan para entrenar la representación de los puntos/datos en lo que se conoce como el espacio de instancia (*instance space*).

El funcionamiento de estos modelos se basa en la clasificación de los datos basándose en la geometría de como se distribuyen en el plano (2 dimensiones) o en el hiperplano (+ de 2 dimensiones) utilizando visión por computador.

Estos modelos se dividen en dos grupos según si la clasificación se construye en base al espacio de instancia conocidos como “Modelos Lineales” o si se construye en base a las distancias entre cada instancia “Modelos Basados en Distancias”.

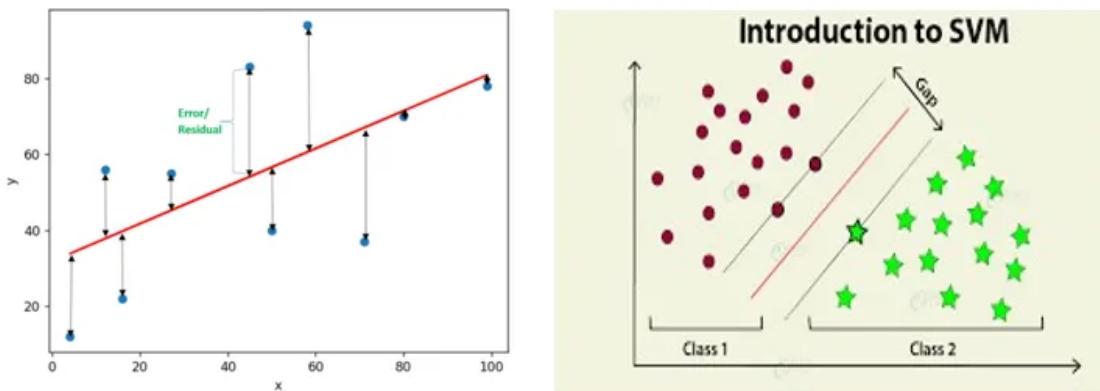


Figura 6.13: Ejemplo de Modelo Geométrico Lineal (Saradalakshmi8074, 2024)

Por un lado, como se aprecia en la Figura 6.13, los modelos lineales son aquellos que dividen el espacio en una o más secciones en base a utilizar (hiper)planos para poder clasificar según la posición de la instancia en el (hiper)plano.

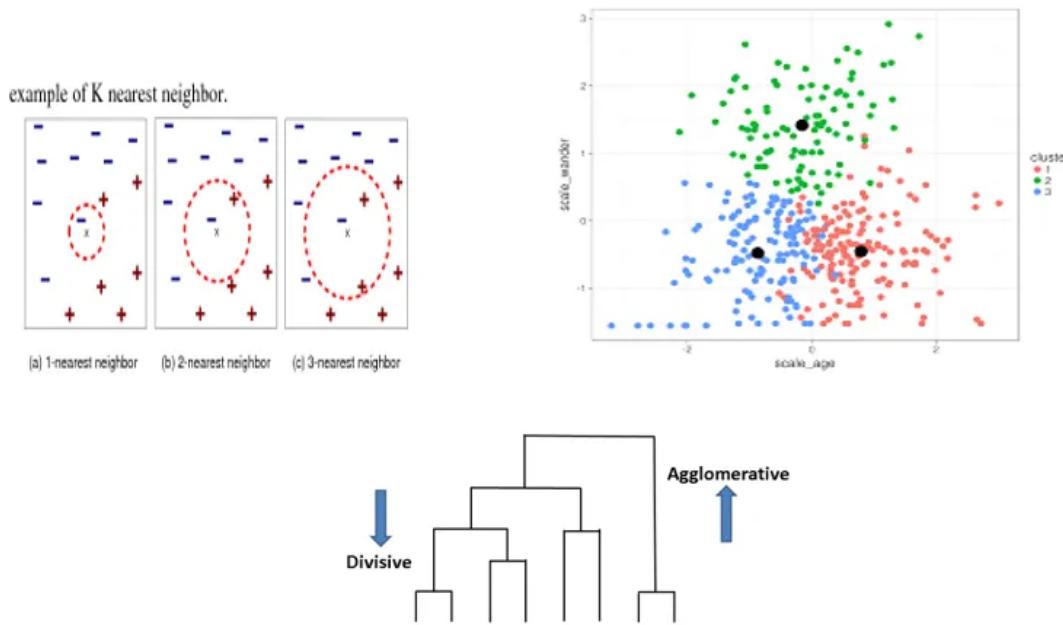
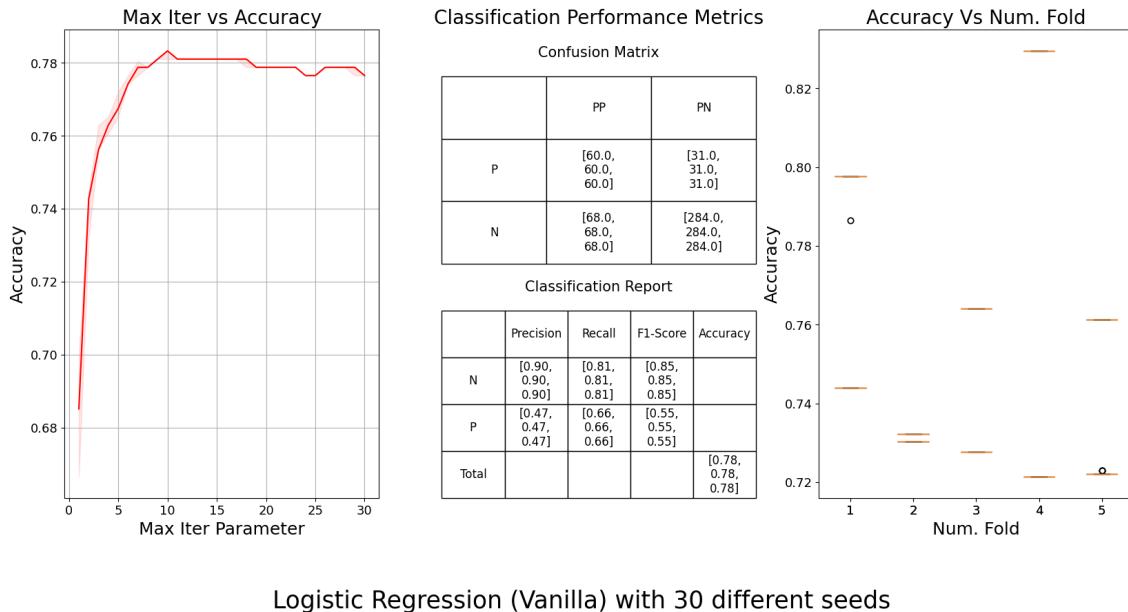


Figura 6.14: Ejemplo de Modelo Geométrico basado en distancia (Saradatalakshmi8074, 2024)

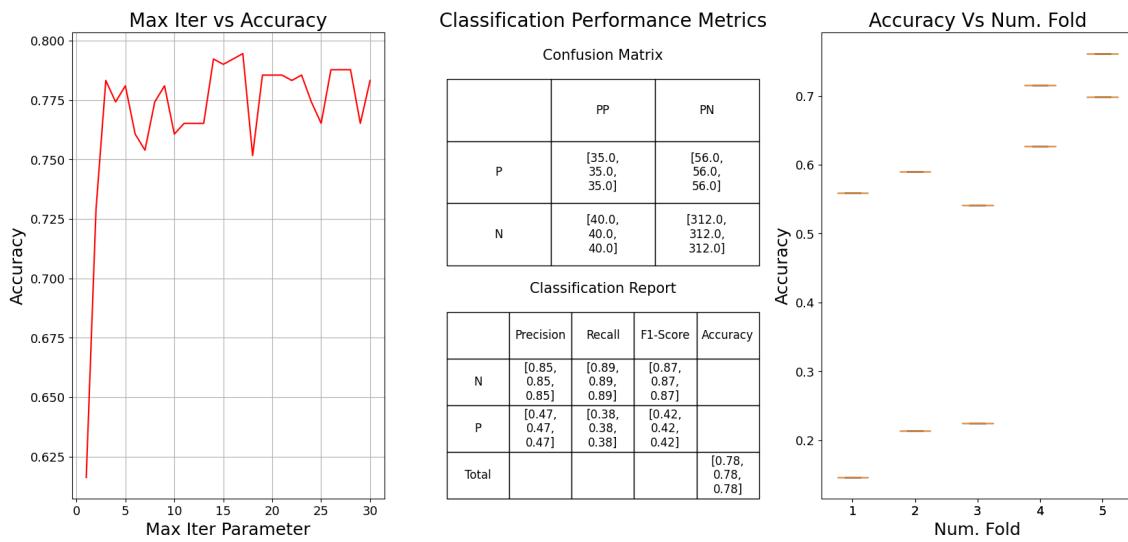
Por otro lado, como se observa en la Figura 6.14, los modelos basados en distancia son aquellos que clasifican usando la distancia entre ellos para determinar cual es su clase.

Una vez explicado el modelo con sus tipos, se ha procedido a ejecutar dos clasificadores geométricos lineales “Vanilla” al igual que se planteó con el anterior modelo. Se han ejecutado los clasificadores que se ven en la figura 6.13 los cuales son *LogisticRegression* y *SVC* junto con su versión lineal *LinearSVC* de **Scikit-Learn**.



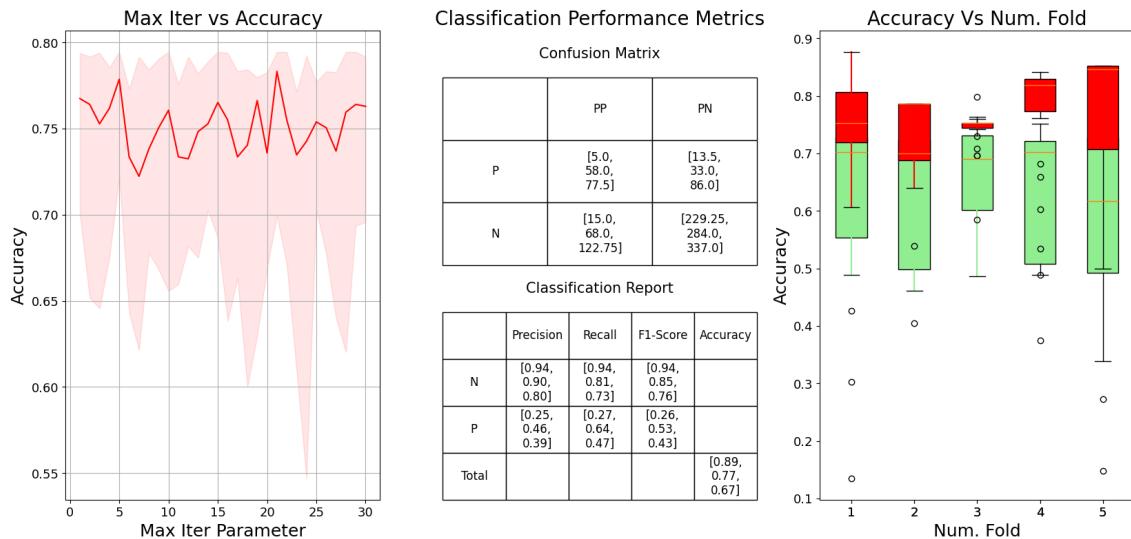
Logistic Regression (Vanilla) with 30 different seeds

Figura 6.15: Ejecución del clasificador (*Logistic Regression*) con 30 ejecuciones con distintas seeds y con 30 iteraciones máximas



Support Vector Machine (Vanilla) with 30 different seeds

Figura 6.16: Ejecución del clasificador (*SVC*) con 30 ejecuciones con distintas seeds y con 30 iteraciones máximas



Linear Support Vector Machine (Vanilla) with 30 different seeds

Figura 6.17: Ejecución del clasificador (*LinearSVC*) con 30 ejecuciones con distintas seeds y con 30 iteraciones máximas

Al igual que hicimos con el *DecisionTreeClassifier*, se va a comparar los tres clasificadores del modelo geométrico lineal escogidos:

- **Max Iter vs Accuracy:** la mediana de las exactitudes, dependiendo del clasificador, se puede estabilizar como en el ejemplo del clasificador *LogisticRegression* en la iteración 10 y, por otro lado, ocurre como se comentó en la explicación del *DecisionTreeClassifier* que un sobreentrenamiento de los datos puede llegar a empeorar los datos como se observa en la gráfica del *SVC* y del *LinearSVC* donde ejecutar más iteraciones no significa una mejora potencial de la Exactitud.

Se observa como la técnica de *LinearSVC* es muy volátil/estocástico. Es decir, dependiendo de la seed pasada a la técnica, varía las exactitudes obtenidas.

- **Classification Performance Metrics:** en este caso no existe una dispersión de valores (salvo el *LinearSVC*) ya que no se ha pasado el argumento correspondiente para aplicar probabilidades a los modelos, ocurriendo lo mismo en el reporte de la clasificación.

- **Accuracy vs Num Fold:** Se presenta (en valores únicos) un sobreajuste en

los datos tras utilizar la técnica de los *Kfolds* ya que, para cada *fold*, todas las exactitudes del conjunto de validación usado están por debajo del conjunto de entrenamiento, es decir, independientemente de la ejecución del clasificador usada, siempre se produce mejor predicción en el conjunto de entrenamiento (verde) que en el conjunto de validación (rojo).

Se observa como para los *folds* de los conjuntos de entrenamiento y validación de los modelos que no son *LinearSVC*) no es disperso debido a lo comentado del argumento de probabilidad, aunque para los *folds* del *LinearSVC*) son bastante dispersos llegando a varias hasta un 30 % en la Exactitud.

- **Conclusión final:** Al igual que el apartado anterior, aunque se presenta una muy buena exactitud general (78 %), se busca clasificar con certeza los *animales* que van a ser licenciados, es decir, que tengan etiqueta 1 (Positiva), por lo que priorizamos los resultados del Reporte de la Clasificación correspondiente a la etiqueta Positiva, que en este caso, no llegan al 50 % por lo que descartamos usar estos modelos.

Modelo Probabilístico

Los modelos probabilísticos son aquellos que utilizan la teoría de la probabilidad y el factor de la aleatoriedad para poder clasificar en función de las probabilidades condicionadas.

El funcionamiento de estos modelos se basa en suponer la probabilidad $P(Y|X)$ siendo de la variable *target* Y condicionada al evento de que ocurra X. Esta probabilidad se puede resolver para todas las combinaciones de Y condicionada por cualquier columna X aplicando el teorema de Bayes.

En nuestro caso, se ha ejecutado el clasificador *GaussianNB* para comprobar el rendimiento, pero al no contar con las probabilidades, ni poder entrenarlo de manera correcta, se ha descartado del desarrollo.

Modelo Mixto

Los modelos mixtos son aquellos que utilizan una combinación de todas las ideas basadas en modelos lógicos, geométricos y probabilísticos.

Los clasificadores más relevantes en torno a los modelos mixtos son las Redes Neuronales, aunque por los costes de desarrollar varias técnicas en *Deep Learning* a la vez, no se ha considerado como parte del desarrollo.

Ensembles y Tipos

Una vez que se han detallado todos los modelos se ha decidido usar varios al mismo tiempo como clasificador general. Para ello, se necesita entender qué es un *ensemble*, sus ventajas y qué tipos existen.

Un *ensemble* es un conjunto de clasificadores que actúan de forma simultánea para ofrecer predicciones de forma conjunta como si se tratara de un clasificador único, reduciendo la varianza de todos los clasificadores por separado y, por consiguiente, reduciendo el error en las predicciones.

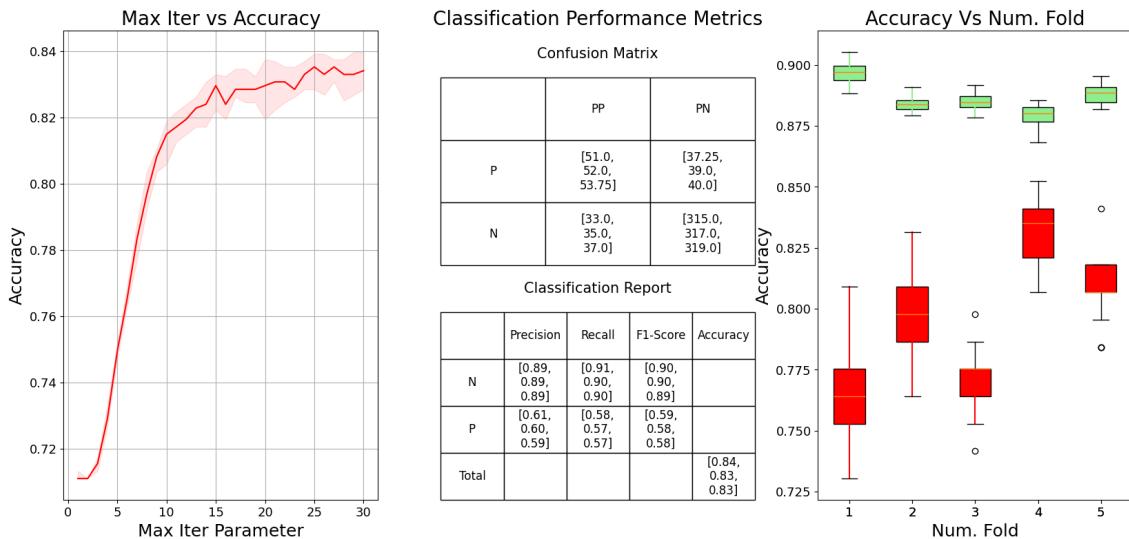
Estos conjuntos de clasificadores, aunque actúen de manera conjunta, tienen que contemplar una forma de agregación de los resultados para llegar a una conclusión final a la hora de clasificar. Dichas agregaciones son las que subdividen a los *ensembles* en sus tipos, los cuales son:

- **Voting:** Se combinan las predicciones de múltiples modelos de *Machine Learning* para tomar una decisión final. Cada modelo en el *ensemble* emite su propia predicción y la clase o resultado final se determina por mayoría, es decir, se selecciona la clase que obtiene más votos. Esta técnica es comúnmente utilizada en problemas de clasificación por lo que se utilizará para el desarrollo.
- **Bagging:** Se generan múltiples muestras de entrenamiento mediante el muestreo con reemplazo de los datos de entrenamiento originales. Se entrena un modelo base en cada una de las muestras y las predicciones de los modelos se

combinan mediante votación por mayoría (en problemas de clasificación) o promedio (en problemas de regresión) para obtener una predicción final. *Bagging* ayuda a reducir la varianza y mejorar la precisión general del modelo.

- **Boosting:** Se construyen modelos de forma secuencial, donde cada modelo intenta corregir los errores del modelo anterior. En cada iteración, se da más peso a los ejemplos de entrenamiento que fueron clasificados incorrectamente por los modelos anteriores, de modo que el nuevo modelo se enfoca más en esos ejemplos difíciles. Al combinar las predicciones de todos los modelos, se obtiene una predicción final. El *boosting* suele lograr modelos con un rendimiento aún mejor que los modelos individuales y puede ser utilizado tanto para problemas de clasificación como de regresión.
- **Stacking:** Se combinan las predicciones de múltiples modelos base, pero en lugar de utilizar una votación simple o promedio, se entrena un modelo de nivel superior, llamado “meta-modelo”, para hacer la predicción final. Los modelos base proporcionan sus predicciones individuales, que se utilizan como características de entrada para el meta-modelo. El meta-modelo aprende a combinar estas características para generar la predicción final. *Stacking* puede ser más poderoso que otros métodos de *ensemble*, ya que permite capturar relaciones más complejas entre las características y las etiquetas en los datos de entrenamiento.

Una vez que se ha enunciado lo que son los *ensembles*, sus ventajas y sus tipos, se ha procedido a ejecutar el “meta-clasificador” denominado *RandomForestClassifier* de **Scikit-Learn** el cual es de tipo *Voting* y combina la ejecución de varios árboles de decisión.



Random Forest (Vanilla) with 30 different seeds

Figura 6.18: Ejecución del clasificador (*RandomForestClassifier*) con 30 ejecuciones con distintas *seeds* y con 30 iteraciones máximas

Por último, se encuentra con el resultado de un *ensembles* realizado sobre los datos como se aprecia en la figura 6.18. A continuación, se realiza una explicación de los gráficos obtenidos:

- **Max Iter vs Accuracy:** la mediana de las exactitudes se estabiliza entorno a la Iteración 30 observando que nos da como resultado la mejor exactitud de todos los modelos ejecutados.
- **Classification Performance Metrics:** la mayoría de los valores predichos con respecto a los reales entre los tres cuartiles no tienen una gran varianza entre ellos, lo que resulta en que todos los modelos clasifican de manera similar. Ocurre de manera similar en el reporte de la clasificación donde los valores de Precisión (*Precision*), Exhaustividad (*Recall*) y F1-Score no varian entre ellos.
- **Accuracy vs Num Fold:** Se presenta un sobreajuste en los datos tras utilizar la técnica de los *Kfolds* ya que, para cada *fold*, todas las exactitudes del conjunto de validación usado están por debajo del conjunto de entrenamiento, es decir, independientemente de la ejecución del clasificador usada, siempre se

produce mejor predicción en el conjunto de entrenamiento (verde) que en el conjunto de validación (rojo).

Además, se observa como para los *folds* del conjunto de entrenamiento, es muy poco disperso, pero para los *folds* del conjunto de validación, son bastante dispersos llegando a variar hasta un 5 % en la Exactitud.

- **Conclusión final:** Se observa como es el clasificador que mejor exactitud ha tenido de todos y, además, si observamos los valores de precisión, exhaustividad y F1-Score sobre la etiqueta positiva 1, cumple con la condición principal planteada ya que clasifica entorno a un 60 % de las etiquetas positivas correctamente, por lo que va a ser el modelo final a utilizar. Además, se observa que las exactitudes en el conjunto de validación son altas y poco dispersas.

6.3. Conclusión

Una vez realizadas todas las ejecuciones de todos los modelos investigados se van a evaluar las hipótesis propuestas al comienzo del desarrollo.

- Una de las técnicas que mejor predice el comportamiento de las etiquetas (en cómputo global) a predecir es la Regresión Logística, ya que existe una linearidad marcada en las puntuaciones de los datos y se trata de una clasificación binaria.

Esta hipótesis queda rechazada debido a que, aunque se ha obtenido una exactitud por encima de un 75 %, la clasificación de la etiqueta positiva no llega al 50 %, por lo que no se puede contemplar que la Regresión Logística haya sido certera.

- La técnica que peor predice el comportamiento de las etiquetas (en cómputo global) son los Árboles de Decisión, ya que se basan en Modelos Lógicos.

Esta hipótesis queda rechazada ya que sí que se ha encontrado reglas lógicas a través del clasificador que ha llevado a realizar una clasificación mejores que otras clasificaciones que se han visto en el estudio y la comparación realizada.

- La mejor técnica va a ser la técnica de *ensemble* debido a que utiliza varias formas de predecir el resultado final (voting, stacking, bagging, boosting) y es más potente computacionalmente. (Gozhulovskyi, 2023)

Esta hipótesis queda aceptada ya que, efectivamente, se partía de la cita mencionada como principal fuente de información y se ha validado en la comparación obteniendo una exactitud de 83 % como mediana y las métricas relacionadas a la etiqueta positiva cercanas al 60 %.

Capítulo 7

Conclusiones y trabajos futuros

Habiendo propuesto los dos sistemas con sus correspondientes prototipos, junto a las encuestas de evaluación, se ha validado que son aproximaciones adecuadas para el público objetivo.

Para finalizar, se analizará las conclusiones obtenidas de los dos sistemas investigados, la encuesta tomando en cuenta las aportaciones a modo de *feedback* junto a las conclusiones de los desarrollos, además de un análisis DAFO de la solución. y viendo la potencial implementación de lo desarrollado en la memoria a futuro.

7.1. Conclusiones de los Sistemas

La creación de un prototipo sobre un *Sistema Recomendador* ha permitido identificar las técnicas más relevantes. Como ya se vio en las premisas, el Topic Modeling sería la aproximación adecuada ya que permite distinguir el tono desde un punto de vista narrativo. Las técnicas de NER, Text Similarity o Clustering no recogen información semántica, creando anomalías en la muestra realizada con la primera de estas técnicas.

Esta conclusión se ve reforzada por los resultados validados de la encuesta, ya el Topic Modeling obtuvo los mejores resultados.

La conclusión del ***Sistema Predictivo*** coincide con la hipótesis inicial planteada en su desarrollo, la mejor aproximación ha sido la utilización de un clasificador de tipo ***ensemble*** como es *RandomForestClassifier* teniendo una exactitud por encima del 80 % y unos valores para la precisión y la exhaustividad de la etiqueta positiva cercana al 60 % lo que favorece a la predicción de ***animés*** de impacto.

Se ha priorizado un desarrollo más extenso del segundo de los sistemas, para cumplir con los plazos estimados.

Esto se debe a que hiperparametrizar clasificadores de ***Machine Learning*** requiere menos inversión de tiempo para obtener resultados, en comparación al *fine-tuning* mediante anotaciones para mejorar los entrenamientos en las técnicas de ***Procesamiento del Lenguaje Natural***.

Cabe añadir que hacer cambios en el *dataset*, aunque se contara con sinopsis en español, no cambiaría los resultados del entrenamiento.

Las técnicas mencionadas no se verían afectadas por cambios en el contenido que procesen, por lo que las alternativas son seguir hiperparametrizando o la búsqueda de otras técnicas.

7.2. Análisis de la encuesta de evaluación

La motivación para crear la encuesta era validar nuestras premisas dirigiéndonos a los potenciales usuarios, ya que no se podía acceder de forma pública a las motivaciones en la toma de decisiones de las propias plataformas de *streaming*.

La encuesta se ha realizado mediante un formulario de Google, compartiéndose por los contactos de los investigadores, **con un total de 57 personas encuestadas respondiendo al formulario**.

El cuestionario estaba dividido en las dos soluciones planteadas, además de *feedback* adicional a modo de comentarios:

- **En la primera sección, se explicó cada caso de uso con la técnica usada correspondiente**, comparando ejemplos de los resultados esperados en el desarrollo con respecto a los sistemas de recomendación tradicionales.
- **En la segunda sección, se consultó sobre los patrones de consumo que realizan los usuarios de diversas plataformas de streaming**, para identificar las necesidades que intenta cubrir con la investigación.

La muestra se diseminó en comunidades nicho a modo de *focus group*, por lo que los resultados pueden estar sesgados. Sin embargo, no todas las personas encuestadas toman los mismos patrones, ya que hay varianza en la frecuencia de consumo.

7.3. Feedback de la encuesta

Aunque el objetivo principal de la encuesta sea parametrizar el éxito de los planteamientos, hemos tomado comentarios de los *end-user* sobre como se podría mejorar nuestra propuesta.

¿Algún comentario que deseas añadir sobre los sistemas recomendadores?

11 respuestas

No tiene mucho sentido recomendar animes principalmente por la sinopsis ya que los animes pueden tener tonos muy distintos

Pienso que el uso de la NER con la sinopsis puede ser un poco desventajoso, ya que muchas sinopsis pueden no plasmar correctamente de qué se trata cada anime.

Me parece complicado valorar la pertinencia de unas recomendaciones sobre otras porque dependerá del público de la obra y lo que busque en cada caso. A nivel teórico todos los modelos me parecen prácticos si se utilizan en los casos adecuados (el NER para un espectador de OP funciona si el usuario ya es espectador habitual de anime pero para un "novato" probablemente le sea mas útil la recomendación de Naruto o DB). El que, a nivel general, me parece un acierto mas evidente es el de la clusterización en el deporte porque tengo la impresión que la gente de futbol es muy de futbol y lo priorizará sobre el resto de elementos. En resumen, todos los modelos y todas las recomendaciones me parecen válidas y útiles pero a todas les acompaña un "depende"

(a) Feedback de los Sistemas Recomendadores

¿Algún aspecto relevante para tener en cuenta a la hora de estudiar el impacto de un anime nuevo?

16 respuestas

Cuenta gente busca/escribe el nombre del anime en redes sociales y en buscadores (Google Analytics)

Memes, fanarts y otros contenidos que haga la comunidad

Podría ser interesante incluir el factor de la puntuación dada por el usuario sobre una serie.

El diseño de las "waifus" o la calidad en la animación.

Creo que las distribuidoras españolas muchas veces funcionan por paquetes o por la facilidad de contratar con una empresa con la que ya tienen una relación comercial habitual pero quizás alguna métrica que pueda ser útil sería picos de búsquedas en Google o impresiones de la ficha en MAL (creo que hay mucha gente que usa la página pero solo a nivel informativo, sin participar de votaciones)

(b) Feedback de la Solución Mercadotécnica

Figura 7.1: Feedback aportado por las personas encuestadas

Sobre el *feedback* de los Sistemas Recomendadores, la técnica basada en NER quedaría descartada precisamente por lo que comenta el usuario en cuestión, además de obtener los peores resultados en el caso de uso correspondiente.

Con respecto al tercer comentario, las sinopsis sirven como linea maestra para redirigir al usuario hacia contenido personalizado.

Sobre el *feedback* de la solución mercadotécnica, se incide en el impacto medido por resultados de búsqueda. No se ha tomado como un factor a considerar ya que no hay correlación entre la popularidad de un personaje con respecto al consumo de la serie en las plataformas correspondientes, que se comparten en redes en forma de *meme*. Este impacto ya se ha parametrizado en el sistema predictivo.

Otro punto relevante a analizar es el comentario sobre las distribuidoras que licencian series pactadas previamente con las entidades con las que ya han tenido relaciones comerciales, como estudios de animación, cadenas de televisión y conglomerados. Este punto es una de las principales motivaciones de la investigación.

El problema que presenta este comentario es que existen *animés* como KonoSuba (Akatsuki, 2006) en los que las dos primeras temporadas están distribuidas por Crunchyroll (Kun Gao, 2023), mientras que, un spin-off desarrollado en 2023 de la misma, está en Jonu Play (GROUP, 2023), siendo desarrollada ambas series por la misma entidad.

El comentario sobre el uso pasivo de MyAnimeList es relevante, ya que marca tendencias sin que los usuarios formen parte activa de la misma, siendo parte del inconsciente colectivo canalizado en Internet.

7.4. Análisis DAFO

Por último, se ha realizado un análisis DAFO desde una perspectiva comercial:

- **Debilidades:** Soluciones de recomendación previamente existentes.

Como investigadores noveles, no se podría competir con las infraestructuras de grandes compañías en cuanto a despliegue tecnológico. Nuestra ventaja competitiva es haber encontrado este nicho de mercado, identificando una necesidad y resolviéndola para los inversores.

- **Amenazas:** Branding negativo hacia la IA.

Debido a los avances en IA generativa, existe cierto rechazo hacia el aprendizaje automático. Al mismo tiempo, se dan oportunidades de mercado que son interesantes por los inversores.

Desconocemos cuál es la opinión de los distribuidores en cuanto a solucionar problemáticas concretas y/o cómo puede esto afectar en su imagen hacia fuera.

- **Fortalezas:** Casos de uso concretos para el mundo del anime.

Con toda nuestra investigación, y nuestro conocimiento previo del medio, hemos usado los últimos avances en *Machine Learning Y Procesamiento del Lenguaje Natural* para resolver una compleja situación que afecta a la industria española.

- **Oportunidades:** Creación de nuevas plataformas

Tanto Jonu Media (GROUP, 2023) como SelectaVisión (S.L.U., 2023) están aún creando sus plataformas, por lo que contar con una herramienta que ayude a impulsar su consolidación en el tiempo con respecto a las plataformas internacionales, es una oportunidad de crecimiento para ambas distribuidoras audiovisuales españolas.

7.5. Aspectos sin implementar

Tras el desarrollo de la investigación, se ha priorizado lo metodológico antes que el despliegue de los resultados en una interfaz gráfica para el prototipado, descartando la visualización del proyecto.

Como se había conceptualizado, el desarrollo de una app serviría como propuesta independiente como parte del modelo de negocio.

La principal feature de la misma sería el sistema recomendador, para que los usuarios puedan obtener sugerencias sobre que consumir. Por tanto, el modelo de negocio se daría con la financiación previa de las plataformas, para que sus productos sean recomendados a los potenciales consumidores: Los contenidos se filtrarían dependiendo del interés que puedan tener los potenciales inversores en promocionar cierto contenido.

Además, se podría hacer un seguimiento de las mismas, similar a lo que hace MyAnimeList, aunque como herramienta española. Del mismo modo, se pueden habilitar campañas de promoción concretas para dinamizar lanzamientos. Con estas *features*, cubrimos tanto los problemas de *reach* como de *engagement*, anteriormente mencionados en el desarrollo de la memoria.

Sin embargo, se contempla la idea de ser una API que integrar en portales web, (chat)bots o implementarse dentro de las distintas plataformas.

7.6. Lineas a futuro

A futuro, se necesitará explorar otras técnicas para un *Sistema Recomendador* que contribuyan a adaptar la propuesta a las necesidades de los potenciales inversores.

Se realizarán contactos para colaborar estrechamente con las compañías interesadas como clientes, ya que no se ha podido encontrar información transparente sobre su toma de decisiones a la interna referente a las licencias que deciden localizar sin que existan otros trabajos de investigación al respecto.

Con este trabajo, se pretende cubrir una necesidad para una comunidad cultural al mismo tiempo que impulsamos la economía de la industria audiovisual española, adaptándonos a sus necesidades para crear una investigación de impacto socio-económico.

Como dijo el autor de Naruto (Kishimoto, 2002), Masashi Kishimoto, .^{El} anime no es solo una forma de entretenimiento, es una ventana a universos llenos de posibilidades que inspiran a generaciones"por lo que esperamos que los avances en tecnología habrán camino a nuevos mundos posibles.

Glosario

Agile

Metodología de trabajo la cual se sustenta en la base de trocear dicho en pequeñas partes. De esta forma, cada parte se tiene que terminar y entregarse en pocas semanas para seguir con la realización de las demás partes. 53

Anaconda

Distribución de los lenguajes de programación **Python** y **R** para computación científica, ciencia de datos y aprendizaje automático. 43, 57

Anime

Toda aquella animación de series de dibujos animados de origen japonés realizado de forma tradicional o a través de ordenador. 1, 4, 5, 7, 10, 11, 13, 14, 16–20, 22, 25, 42, 44, 45, 47, 50–52, 60, 63–68, 71, 73, 85, 86, 94–102, 104, 110, 111, 119, 124, 131, 134, 139, 141, 142, 145

Argilla

Plataforma de curación de datos de código abierto para LLM. Con Argilla, se puede crear modelos lingüísticos solventes mediante una curación de datos más rápida que utiliza comentarios tanto humanos como automáticos (Argilla, 2023). 58, 72, 78

Clustering

Se trata de una técnica de **Machine Learning** de aprendizaje no supervisado por el cual se crean una serie de agrupaciones en base a algunas características. 85

Crunchyroll

La plataforma de streaming bajo demanda centrada en el mundo del *anime* más conocida del mundo. 7, 19, 36, 42, 63, 95, 99

Dataframe

Se trata de una estructura (objeto) que consta de una matriz en la que las filas y las columnas tienen asociadas un nombre. Aparte de esto, dependiendo del lenguaje hay algunos con funciones asociadas dentro de ellos. 43, 63–65, 67, 71, 87, 98–100, 110–112

Demografía

Campo de estudio/Ciencia que estudia ciertos atributos de la población humana de categoría cualitativa y cuantitativa con el fin de poder dividir a la población en grupos, en este caso, por edades. 4

Discord

Aplicación SaaS enfocada en el ámbito de la mensajería y el chat de voz. 54

Ensemble

Conjunto de modelos de **Machine Learning** utilizados simultáneamente para realizar predicciones. 35, 96, 97, 125–127, 129, 131

Fansub

Comunidades de aficionados que traducían mediante subtítulado las series de temporada que no eran licenciadas de forma legal. Este movimiento nació cuando comenzaba a usarse Internet de uso comercial en forma de archivos pirata, aunque no se consideraba ilegal al realizar sus funciones sin ánimo de lucro. 7, 18, 19, 144

GitHub

Plataforma web que proporciona alojamiento para el control de versiones de desarrollo de software mediante Git. Es una plataforma integral para que los desarrolladores colaboren, compartan código y gestionen proyectos. 40, 57

Gliner

Modelo de **NER** capaz de identificar cualquier tipo de entidad mediante un codificador transformador bidireccional tipo BERT, usando *zero-shot classification* como aproximación. Constituye una alternativa práctica a los modelos **NER** tradicionales que se limitan a entidades predefinidas, a los modelos de gran tamaño como LLMs que a pesar de su flexibilidad, resultan costosos por sus enormes costes computacionales. 58, 72, 87

Hikikomori

Jóvenes japoneses que se retiran de la sociedad y se aíslan completamente en sus hogares durante un largo período de tiempo, evitando cualquier tipo de interacción social durante meses, años, o incluso décadas. 15

HuggingFace

Empresa y plataforma en línea destacada en el campo del **Procesamiento del Lenguaje Natural** y el **Machine Learning**. 64

Imbalanced-learn

Package de **Python** usado para tratar con clases desbalanceadas en problemas de clasificación en **Machine Learning** 47, 58

Isekai

Subgénero de *anime* que se centra en un personaje o un grupo de personajes que viajan a un mundo de fantasía de un universo paralelo centrado en la ficción. 6, 20

Jonu Media

Una de las principales licenciadoras españolas a nivel nacional de *anime*, perteneciendo anteriormente al grupo Planeta. 8, 20, 42, 51, 63, 95, 99

Jupyter Notebook

Plataforma de programación interactiva utilizada en navegadores web para programar *Python* con la ventaja de poder trabajar de forma similar a un cuaderno. 43, 57

Kaggle

Plataforma en línea que sirve de centro para que los entusiastas de la ciencia de datos y el *Machine Learning* colaboren, compitan y aprendan. 64

Machine Learning

Machine Learning o Aprendizaje Computacional es un ámbito implementado en el campo de la Inteligencia Artificial por el cual, gracias a realizar ciertos algoritmos, permite a nuestros ordenadores procesar una gran cantidad de datos, encontrar ciertos patrones en ellos y conseguir realizar una predicción con datos nuevos. 1–3, 35, 46, 48, 57, 80, 95, 96, 98, 125, 131, 135, 139–141, 143, 145

Mainstream

Anglicismo referido a la tendencia de la mayoría de la gente en hacer una acción porque está de moda. 4

Manga

Cómic/Tebeo de origen japonés. 4, 7, 11, 13, 16, 17, 38, 60, 99

Matplotlib

Package de *Python* usado para crear visualizaciones estáticas y/o animadas. 47, 58

Merchandising

Productos asociados a una marca que sirven como una muestra de la vinculación afectiva como objeto físico, y fuente de ingresos con la que se articula la industria del *anime* como negocio. 4, 8, 11, 18

MMORPG

Massively Multiplayer Online Role Playing Game, videojuegos donde los jugadores interactúan en un metaverso, creando su propia cultura así como su economía. 6

Moe

Aquellos *animés* protagonizados por chicas adorables con las que el espectador se encariña. El moe puede también asociarse a personajes o actitudes concretas que aparezcan en otros formatos. 6, 20

MyAnimeList

Página web/Base de Datos más activa a nivel mundial de *anime* con la mayor comunidad y cantidad de datos a nivel internacional de occidente. 5, 20, 42, 61–63, 65, 71, 98

Named-Entity Recognition (NER)

Técnica de *Procesamiento del Lenguaje Natural* que trata de establecer vectorizaciones entre strings se suele aplicar un meta-etiquetado con una ontología establecida por temática. 69, 72, 85, 140

Natural Language Tool Kit

Biblioteca de Python que proporciona un conjunto de herramientas y recursos para el procesamiento del lenguaje natural 45, 58

Nekketsu

Series caracterizadas por la acción, donde mediante el compañerismo, se superan adversidades. A esta categoría pertenecen la mayoría de series populares del mundo, con ejemplos conocidos en España como Los Caballeros del Zodiaco (Kurumada, 1986) o Digimon (Hongo, 1999). 5, 73

Netflix

Plataforma de streaming con contenido audiovisual que comenzó a producir su propio contenido, extendiéndose hasta los videojuegos. Su popularidad cimentó el actual modelo de negocio basado en el capitalismo de plataforma para otros medios como Amazon Play o HBO Max. 7, 19, 36, 42, 63, 95, 99

NumPy

Package de librerías de *Python* para el uso de vectores y matrices uni/multidimensionales, junto al uso de funciones matemáticas para operar con ellas. 80, 143

Otaku

Persona muy aficionada/apasionada al mundo del *anime* 4, 10, 13–17, 20

OVA

Siglas para Original Video Animation. Con la llegada de los lanzamientos domésticos en video, surgieron animaciones que se desviaban de la historia principal o rellenaban huecos argumentales expandiendo el original, incluso llevando la trama en otro dirección. No hay que confundirlas con películas, aunque tengan una duración similar. 6

Package

Es un término acuñado en los proyectos de **R**, **pythony** en otros lenguajes de programación referido a un conjunto de códigos/scripts 57, 111, 140–145

Pandas

Package de librerías de **Python** especializada en la manipulación y el análisis de datos, utilizado junto con el paquete **NumPy**. 43, 47, 58, 63

Procesamiento del Lenguaje Natural (NLP)

Área de la Inteligencia Artificial centrada en como los ordenadores entienden, computan y expresan el lenguaje humano. 1–3, 44, 45, 57, 58, 69, 71, 72, 77, 131, 135, 140, 142, 144, 145

Pyod

Package de librerías de **Python** usado para la detección de anomalías en datos, gracias a la detección de patrones y grupos. 58

Python

Lenguaje de programación interpretado (no requiere de compilación a la hora de ejecutarse). Es el lenguaje donde la mayoría de los algoritmos y aplicaciones de Inteligencia Artificial se agrupan. 42, 43, 45, 47, 57, 61, 139–145

R

Lenguaje de programación compilado (requiere de compilación a la hora de ejecutarse). Es un lenguaje enfocado al análisis de datos. 58, 139, 143, 144

Remake

Realización de una nueva versión de una obra previa, ya sea una película, una serie de televisión, una obra de teatro u otro medio. En un remake, la historia principal se mantiene en gran medida, pero se actualizan los elementos técnicos, los efectos visuales, el elenco de actores y, en algunos casos, se realizan cambios menores en el argumento. El objetivo principal de un remake es presentar la historia a una nueva audiencia o revitalizarla para la audiencia actual. 6

Retelling

Volver a contar una historia existente, pero con cambios significativos en la trama, los personajes o el contexto. En un retelling, se puede modificar la época, el escenario, los personajes principales o incluso la perspectiva narrativa, con el fin de ofrecer una nueva interpretación o explorar diferentes aspectos de la historia original. Un retelling tiende a ser más creativo y arriesgado en términos de reinventar la historia, mientras que un remake se mantiene más cercano al material original. 6

Scikit-Learn

Package de librerías de **Python** usado para la mayoría de algoritmos básicos de **Machine Learning**. 48, 58, 80, 117, 121, 126

Scraperapi

Software as a Service maneja la rotación de proxy, navegadores y CAPTCHAs para que se pueda realizar *webscraping* a cualquier página con una sola llamada a la API. 42, 58, 62

Scrapy

Framework de scraping y crawling (*webscraping*) de código abierto, escrito en *Python* para la obtención de datos de forma automática de las páginas web. 42, 58, 61

Seaborn

Package de *Python* usado para mostrar gráficos estadísticos de una forma más ilustrativa y atractiva. 47, 58

Seinen

Aunque etimológicamente signifique “juventud” este género se caracteriza por su temática adulta mostrando violencia de forma explícita. 4, 20

Selecta Visión

Principal empresa distribuidora audiovisual que licencia animes en España, siendo productora de contenidos televisivos juveniles que contribuyeron a su popularidad. 8, 20, 42, 51, 63, 95, 99

Sentence Transformer

Técnicas de *Procesamiento del Lenguaje Natural* que se enfocan en capturar la información semántica y contextual de oraciones o textos completos. 74, 85

Shiny

Package de *R* gratuito y de código abierto para desarrollar aplicaciones web. Uno de los usos de Shiny ha sido la creación rápida de prototipos. 58

Shoujo

Proviene de la palabra japonesa que significa “chica” con tramas románticas que no siempre son narradas desde una óptica femenina. 4, 6, 20, 69

Shounen

Proviene de la palabra japonesa que significa “chico” centrados en la aventura. No todos tienen porque mostrar acción ya que hay algunos con tramas sobre el deporte o en relaciones amorosas. 4, 6, 20, 69, 73

Simulcast

Emisión simultánea a nivel mundial de un estreno de anime de temporada. Con esto, los aficionados no deben esperar a que el anime llegue doblado cuando se licencien en sus países o subtitulado por la comunidad de fansub de manera ilegal. 8, 20

Sistema Predictivo

Algoritmo propuesto que decide si un *anime* que va a salir al mercado va a ser de éxito, en este caso basado en el uso de técnicas de *Machine Learning* para la predicción de la clasificación binaria. 3, 9, 46, 49, 50, 52, 53, 55, 60, 63, 67, 94, 95, 131

Sistema Recomendador

Algoritmo propuesto que recomienda una lista de *animés* en base a uno, en este caso basado en el uso de técnicas de *Procesamiento del Lenguaje Natural* para el procesamiento de la sinopsis (Content-Based recommender system). 1, 2, 9, 25–28, 30–32, 35, 36, 39, 44, 49, 50, 52, 53, 55, 60, 63, 65, 68, 69, 71, 85, 86, 130, 137

Slice of life

Género que surgió en la época de los 2000 donde se dejaba de lado las historias de fantasía para hablar sobre la vida cotidiana de sus personajes, sin obviar elementos inverosímiles propios de la ficción. 6, 20

SpaCy

Package de *Python* para procesamiento y análisis de textos. 45, 58, 72, 77

Spokon

Anime de deportes, tenemos ejemplos desde boxeo, fútbol o baloncesto y hasta juegos de mesa. 5

Streamlit

Framework para aplicaciones *Python* de código abierto enfocado en el ámbito de *Machine Learning* y de la ciencia de datos. 45, 48, 58

Webscraping

Proceso de extracción de contenidos y datos de sitios web mediante software. 41–43, 53, 56–58, 61, 65, 144

Word-Embeddings

Estas técnicas de *Procesamiento del Lenguaje Natural* se utilizan para representar palabras en forma de vectores numéricos, capturando información semántica y sintáctica. 72, 74

Bibliografía

- Akatsuki, N. (2006). *KonoSuba!* (Anime). Studio Deen.
- Anno, H. (1995). *Neon Genesis Evangelion* (Anime). Gainax.
- Arakawa, H. (2003). *Full Metal Alchemist* (Anime). BONES.
- Arakawa, N. (2014). *Your Lie in April* (Anime). A-1 Pictures.
- Argilla. (2023). *Argilla*. <https://areajugones.sport/anime/esta-el-anime-de-chainsaw-man-siendo-un-fracaso-en-japon/> (Recuperado el día 20 de Septiembre de 2023)
- Awan, A. A. (2024). *Decision Tree Image*. <https://www.datacamp.com/tutorial/decision-tree-classification-python> (Recuperado el 09 de junio de 2024)
- Azuma, H. (2009). *Otakus: Japan's database animals*.
- ByteDance. (s.f.). *TikTok*. <https://www.tiktok.com/>
- Cádiz, O. (2023). *Desarrollan una base de datos de letras de agrupaciones mediante inteligencia artificial*. <https://ondacadiz.es/noticias/2023-05/desarrollan-una-base-de-datos-de-letras-de-agrupaciones-mediante-inteligencia> (Recuperado el día 20 de Septiembre de 2023)
- Carlos A. Gomez-Uribe, N. H. (2023). *The Netflix Recommender System: Algorithms, Business Value, and Innovation*. <https://dl.acm.org/doi/10.1145/2843948> (Recuperado el día 20 de Septiembre de 2023)
- Cloud Skew web*. (2023). <https://app.cloudskew.com/editor/f0e0e1cf-6a48-4083-9189-eddd91821cab>

dynabyte. (2024). *Applications and Benefits of Recommender Systems*. <https://www.linkedin.com/pulse/applications-benefits-recommender-systems-dynabyte-2vsaf/> (Recuperado el día 15 de Mayo de 2024)

Fusanosuke, N. (1992). *Tezuka Osamu wa doko ni iru.*

González-Fierro, M. (2023). *Webinar (AI Tech Talk): Creando Sistemas de Recomendación desde Cero*. <https://www.youtube.com/watch?v=qHgT6yHG1u4> (Recuperado el 15 de junio de 2023)

Gozhulovskyi, A. (2023). *Choosing a Model for Binary Classification Problem*. <https://medium.com/@andrii.gozhulovskyi/choosing-a-model-for-binary-classification-problem-f211f7a4e263> (Recuperado el día 20 de Septiembre de 2023)

GROUP, X. (2023). *Jonu Play*. <https://jonuplay.com/> (Recuperado el día 20 de Septiembre de 2023)

Gyssler, G. (2023). *MyAnimeList*. <https://myanimelist.net/> (Recuperado el día 20 de Septiembre de 2023)

Hongo, A. (1999). *Digimon (Anime)*. Toei Animation.

IAT. (2024). *MACHINE LEARNING. TIPOS, MODELOS, TÉCNICAS Y USOS*. <https://iat.es/tecnologias/inteligencia-artificial/machine-learning/> (Recuperado el 09 de junio de 2024)

Jaki, R. (1983). *Cyberpunk Edgerunners (Anime)*. Studio Trigger, Netflix.

Kamiyama, K. (2002). *Ghost in the Shell: Stand Alone Complex (Anime)*. Production I.G.

Katabuchi, S. (2006). *Black Lagoon (Anime)*. Madhouse.

Kishimoto, M. (2002). *Naruto (Anime)*.

Kobayashi, O. (2002). *Beck (Anime)*. Madhouse.

Kuga, C. (1998). *Cowboy Bebop (Anime)*. Sunrise.

Kun Gao, J. L. (2023). *Crunchyroll*. <https://www.crunchyroll.com/es> (Recuperado el día 20 de Septiembre de 2023)

- Kurumada, M. (1986). *Saint Seiya (Anime)*. Toei Animation.
- Lazzarato, M. (2004). *Les Revolutions du Capitalisme*.
- Maklin, C. (2023). *Gradient Boosting Decision Tree Algorithm Explained*. <https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4> (Recuperado el día 20 de Septiembre de 2023)
- Martín, M. J. C. (2024a). *Analysing the conceptual evolution of qualitative marketing research through science mapping analysis*. https://www.academia.edu/11388673/Analysing_the_conceptual_evolution_of_qualitative_marketing_research_through_science_mapping_analysis (Recuperado el 09 de junio de 2023)
- Martín, M. J. C. (2024b). *The Simpsons did it: Exploring the film trope space and its large scale structure*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8011780/> (Recuperado el 09 de junio de 2023)
- Matsumoto, L. (1978). *Capitán Harlock (Anime)*. Toei Animation.
- Mehdi Hamed, M. (2024). *Decision Tree splitting criteria: Gini Index vs Entropy (Image)*. https://www.linkedin.com/posts/mehdihamedi_decisiontrees-machinelearning-classification-activity-7108847598337171456-XcUi/ (Recuperado el 09 de junio de 2024)
- Microsoft. (2023). *Smart Adaptive Recommendations (SAR) Algorithm*. <https://github.com/microsoft/Product-Recommendations/blob/master/doc/sar.md> (Recuperado el 15 de junio de 2023)
- Miyao, D. (2002). *Before Anime*.
- Nagahama, H. (2002). *Detroit Metal City (Anime)*. Studio 4ºC.
- Netflix. (2023a). *¿Cómo funciona el sistema de recomendaciones de Netflix?* <https://help.netflix.com/es-es/node/100639> (Recuperado el día 20 de Septiembre de 2023)

Netflix. (2023b). *Recommendations*. <https://research.netflix.com/research-area/recommendations> (Recuperado el día 20 de Septiembre de 2023)

Nobuyuki, T. (1968). *Nihon animēshon no chikara*. University of Montfort.

Nonaka, E. (2003). *Cromartie High School (Anime)*. Production I.G.

Nvidia. (2024). *Recommendation System*. <https://www.nvidia.com/en-us/glossary/recommendation-system/> (Recuperado el día 15 de Mayo de 2024)

Oda, E. (1999). *One Piece (Anime)*. Toei Animation.

Peláez, D. (2023). Málaga AI, la comuna de aplicaciones de la inteligencia artificial: "Uno más uno sale más que dos". *El Español de Málaga*. (Recuperado el día 20 de Septiembre de 2023).

Pham, K. (2022). *What are Recommendation Systems?* <https://www.nvidia.com/en-us/glossary/recommendation-system/> (Recuperado el día 15 de Mayo de 2024)

Reed Hastings, M. R. (2023). *Netflix*. <https://www.netflix.com> (Recuperado el día 20 de Septiembre de 2023)

Rençberoglu, E. (2024). *Fundamental Techniques of Feature Engineering for Machine Learning*. <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114> (Recuperado el día 15 de Junio de 2024)

Saradalakshmi8074. (2024). *Logical vs Geometrical vs Probabilistic Models in Machine Learning*. <https://medium.com/@saradalakshmitunuguntla/logical-vs-geometrical-vs-probabilistic-models-in-machine-learning-ae3a33e6be1e> (Recuperado el 09 de junio de 2024)

Seko, H. (2022). *Chainsaw Man (Anime)*. MAPPA.

Sevakis, J. (2008). *Video Site with Unauthorized Anime Gets US Dolars 4M Capital*. <https://www.animenewsnetwork.com/news/2008-03-11/video-site-with-unauthorized-anime-gets-us%244m-venture>

S.L.U., S. V. (2023). *Anime-Box*. <https://anime-box.com> (Recuperado el 15 de junio de 2023)

- Sorachi, H. (2003). *Gintama* (Anime). Sunrise.
- Spain-AI. (2023). <https://www.spain-ai.com/> (Recuperado el 15 de junio de 2023)
- Steinberg, M. (2012). *Anime's Media Mix Franchising Toys and Characters in Japan* (Libro). University of Minnesota Press.
- Tabata, Y. (2002). *Black Clover* (Anime). Pierrot.
- Takahashi, Y. (1983). *Captain Tsubasa* (Anime). Tsuchida Production.
- Tezuka, O. (1963). *Astro Boy* (Anime). Mushi Production.
- Toriyama, A. (1986). *Dragon Ball* (Anime). Toei Animation.
- Villa, C. M. (2023). *¿Está el anime de Chainsaw Man siendo un fracaso en Japón?* <https://areajugones.sport.es/anime/esta-el-anime-de-chainsaw-man-siendo-un-fracaso-en-japon/> (Recuperado el día 20 de Septiembre de 2023)
- Watanabe, S. (2004). *Samurai Champloo* (Anime). Manglobe.
- Watsuki, N. (1996). *Rurouni Kenshin* (Anime). Studio Gallop y Studio DEEN.
- Wells, P. (1998). *Understanding Animation*. University of Montfort.
- Yamada, N. (2009). *K-On!* (Anime). Kyoto Animation.
- Yoshinari, Y. (2017). *Little Witch Academia* (Anime). Trigger.