UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA MATEMÁTICA APLICADA AL ANÁLISIS DE DATOS

PROYECTO FIN DE GRADO

Predicción de la viralidad de un video en TikTok y asesor de contenido

JUAN CARLOS RONDEAU CADARSO

CURSO 2021-2022

TÍTULO: PREDICCIÓN DE LA VIRALIDAD EN TIKTOK

AUTOR: JUAN CARLOS RONDEAU CADARSO

TITULACIÓN: GRADO EN INGENIERÍA MATEMÁTICA APLICADA AL ANÁLISIS DE DATOS

DIRECTOR/ES DEL PROYECTO: Rafael Muñoz

FECHA: Julio de 2022

RESUMEN

Hoy en día, las redes sociales tienen una gran relevancia en nuestra sociedad. Los llamados millennials pasan en torno a 5 horas al día metidos en sus redes sociales como twitter, Instagram o TikTok. Las empresas invierten una gran cantidad de dinero en promover sus productos o servicios a través de estos nuevos medios de comunicación. En las redes sociales la publicidad es indirecta ya que viene camuflada en publicaciones entretenidas publicadas por los denominados influencers.

En rasgos generales los influencers tienen dos fuentes de ingresos. La primera fuente de ingreso es lo que pagan las redes sociales según el número de visitas y likes que tengas mensualmente y la segunda es lo que las empresas pagan por promocionar sus productos o servicios a través de las publicaciones de los influencers. Tanto las aplicaciones como las empresas pagan según la repercusión que tengas. Es por ello que los influencers dependen de su público. Conseguir hacer un video viral es difícil de por sí pero mantenerse es lo más complicado. Esto genera estrés y ansiedad ya que deben publicar prácticamente a diario para no perder seguidores pero la creatividad y la originalidad del ser humano es limitada. Y a esto hay que sumarle la inseguridad del mismo.

El proyecto consiste en crear una aplicación que cumpla dos funcionalidades principales. La primera funcionalidad y la más importante consiste en predecir la repercusión que va a tener un video en TikTok prediciendo el número de likes según las características del video como la duración, la calidad de imagen, la descripción, las estadísticas de la cuenta del influencer en cuestión y el contenido del video. De esta forma, al saber la cantidad de likes que tendrá un video, el creador de contenido puede probar distintas combinaciones y publicar finalmente el video que mejor resultados obtiene. La segunda funcionalidad se basa en asesorar al usuario con posibles temáticas en sus próximos videos. Esta funcionalidad tiene dos apartados. El primero se encarga de aconsejar al usuario videos de TikTok y Tweets con repercusión que tengan relación con una temática específica concretada por el usuario. El segundo apartado se basa en mostrar al usuario las tendencias en TikTok y twitter para crear una imagen global del contenido más viral en la actualidad.

Palabras clave: Influencer, TikTok, viralidad, predicción, asesoramiento, machine learning, social media.

ABSTRACT

Nowadays, social networks have a great relevance in our society. Millennials spend around 5 hours a day on social networks such as Twitter, Instagram or TikTok. Companies invest a lot of money in promoting their products or services through these new media. In social networks, advertising is indirect since it comes camouflaged in entertaining publications published by the so-called influencers.

In general terms, influencers have two sources of income. The first source of income is what social networks pay according to the number of visits and likes you have monthly and the second is what companies pay for promoting their products or services through influencers' publications. Both apps and companies pay according to the impact you have. That is why influencers depend on their audience. Making a viral video is difficult in itself, but maintaining it is the most complicated part. This generates stress and anxiety as they must publish almost daily to avoid losing followers, but the creativity and originality of human beings is limited. And to this we must add the insecurity of the same.

The project consists of creating an application that fulfills two main functionalities. The first and most important functionality is to predict the impact that a video will have on TikTok by predicting the number of likes according to the characteristics of the video such as duration, image quality, description, the statistics of the account of the influencer in question and the content of the video. Thus, by knowing the number of likes a video will get, the content creator can test different combinations and finally publish the video that gets the best results. The second functionality is based on advising the user with possible themes in their next videos. This functionality has two sections. The first is responsible for advising the user TikTok videos and Tweets with impact that are related to a specific theme specified by the user. The second section is based on showing the user the trends in TikTok and twitter to create an overall picture of the most viral content today.

Keywords: Influencer, TikTok, virality, prediction, advice, machine learning, social media**.**

AGRADECIMIENTOS

A todos aquellos que me han apoyado, ayudado y animado durante esta etapa tan importante. Quiero agradecer a mi tutor Don Rafael Muñoz y a todos los profesores que han fomentado mi desarrollo, ambición y han promovido mis ganas de saltar al mundo profesional.

"Todo el que disfruta cree que lo que importa del árbol es el fruto, cuando en realidad es la semilla. He aquí la diferencia entre los que crean y los que disfrutan."

Friedrich Nietzsche

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Juan Carlos Rondeau
Título del proyecto:	Predicción de la viralidad de un video en TikTok y asesor de contenido
Directores del proyecto:	Rafael Muñoz
El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:	NO
El proyecto ha implementado un producto:	SI
El proyecto ha consistido en el desarrollo de una investigación o innovación:	SI
Objetivo general del proyecto:	Crear una aplicación que prediga la viralidad de un video en TIKTOK y que asesore acerca de posibles videos con relación a una temática especificada por el usuario y a las tendencias.

Índice

Índice de Figuras

Índice de Tablas

RESUMEN DEL PROYECTO

1.1 Contexto y justificación

Tanto en España como en el resto del mundo, las redes sociales son cada vez más relevantes. Un simple tweet de Elon Musk puede hacer que suba o baje el valor de las acciones de una empresa. El poder de las redes sociales ha hecho que la mayoría de empresas vendan sus productos o servicios por medio de ellas. No solo a través de sus cuentas propias sino también a través de otros perfiles que cumplen con la imagen de la empresa y que tienen una gran repercusión en redes sociales. Estos perfiles son los influencers. Cada vez son más los interesados y cada vez se paga mejor pero el problema no es solo llegar a ser influencer sino el mantener diariamente a tus seguidores pendientes de lo que vayas a publicar.

Los influencer se enfrentan a grandes episodios de estrés y ansiedad ya que dependen económicamente del engagement que tengan y que consigan mantener. Deben cumplir con las expectativas que las empresas tienen sobre ellos. Esto significa que tienen que tener un número mínimo de visitas, likes y tags en el perfil. Los humanos no somos una fuente insaciable de creatividad y menos cuando estamos bajo presión. Y aunque seamos muy creativos, la desconfianza es otra característica común del ser humano.

1.2 Planteamiento del problema

Se pretende dar una solución a todos aquellos influencers que sufren ansiedad y estrés por no saber qué publicar ni cuál va a ser la repercusión que tendrán sus publicaciones. Utilizando algoritmos de machine learning, técnicas de scraping, API's y estadística avanzada se podrá predecir la viralidad de las publicaciones en TikTok en términos de likes además de poder asesorar al usuario ofreciendo una serie de ejemplos de videos en TikTok y tweets en relación a una temática especificada por el usuario y a las tendencias.

1.3 Objetivos del proyecto

Este proyecto propone una aplicación dónde los influencer puedan logearse añadiendo su cuenta de TikTok. La aplicación se basa principalmente en 3 páginas/funcionalidades. Una primera página dónde el usuario añada el próximo video que vaya a publicar para que la aplicación prediga el número de likes que va a tener. La segunda página consiste en una buscador de contenidos según la temática que introduzca el usuario ofreciendo videos y tweet con gran repercusión con relación a esta. La última página se basa en ofrecer al usuario un esquema gráfico del contenido más mainstream en la actualidad, enseñando los videos más virales en tiktok y el trending topic de twitter.

1.4 Resultados obtenidos

La aplicación desarrollada ha alcanzado los requisitos establecidos tras la realización de las pruebas pertinentes y su posterior aprobación. Se consigue realizar un modelo machine learning que predice el número de likes que tendrá un video en tiktok al cabo del tiempo

especificado por el usuario. Este algoritmo se alimenta de datos extraídos a partir de diversas técnicas de scraping y API 's de las propias redes sociales (TikTokAPi y Twitpy). En segundo lugar, realizando diversas consultas a las distintas APIs de las redes sociales, la aplicación es capaz de mostrar al usuario una serie de videos y tweets acordes a una temática introducida por el usuario. Por último, es importante que el usuario vea qué se "cuece" hoy en día en redes sociales para ir acorde a las modas por lo que la aplicación muestra un esquema con las tendencias.

1.5 Estructura de la memoria

- **Capítulo 1:** Resumen del proyecto. Se incluye un breve resumen del proyecto incluyendo el contexto y justificación del problema, así como el planteamiento del mismo. También se explica resumidamente los objetivos, resultados y la estructura de la memoria.
- **Capítulo 2:** Antecedentes / Estado del Arte. Capítulo dedicado al planteamiento al análisis del estado del arte, explicación del contexto, justificación y planteamiento del problema de manera más extensa que en el capítulo 1.
- **Capítulo 3:** Objetivos. Se explican los objetivos, tanto generales como específicos, así como los beneficios del problema.
- **Capítulo 4:** Desarrollo del proyecto. Explicación del funcionamiento del sistema. También se menciona la planificación del proyecto, los recursos que se han requerido para su desarrollo, un presupuesto del mismo y la viabilidad y resultados del proyecto.
- **Capítulo 5:** Discusión. En esta sección se hace una reflexión sobre los resultados principales del proyecto y otros aspectos que han influido en el desarrollo del mismo.
- **Capítulo 6:** Conclusiones. Se redactan las conclusiones a las que se ha llegado tras la finalización del proyecto tanto a nivel personal como a nivel de trabajo.
- **Capítulo 7:** Futuras líneas de trabajo. Se exponen posibles trabajos a futuro que se podrían implementar para mejorar y ampliar las funcionalidades de la aplicación.
- **Capítulo 8:** Conclusiones

Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

2.1 Estado del arte

Transcriptor de audio a texto

Para el desarrollo de mi aplicación, he necesitado utilizar un algoritmo llamado Speechrecognition [7]. Funciona descomponiendo el audio de una grabación de voz en sonidos individuales, analizando cada sonido, utilizando algoritmos para encontrar la palabra más probable que encaje en ese idioma y transcribiendo esos sonidos en texto.

El software de reconocimiento de voz utiliza el procesamiento del lenguaje natural (LPN) y las redes neuronales de aprendizaje profundo. "El PNL es una forma de que los ordenadores analicen, comprendan y deriven el significado del lenguaje humano de una forma inteligente y útil", según el blog de Algorithma. Esto significa que el software descompone el discurso en bits que puede interpretar, lo convierte en un formato digital y analiza las piezas de contenido.

A partir de ahí, el software hace determinaciones basadas en la programación y los patrones de habla, haciendo hipótesis sobre lo que el usuario está diciendo realmente. Después de determinar lo que probablemente dijo el usuario, el software transcribe la conversación en texto.

Random Forest Regression

La regresión Random Forest es un algoritmo de aprendizaje supervisado que utiliza el método de aprendizaje conjunto para la regresión. El método de aprendizaje por conjuntos es una técnica que combina predicciones de múltiples algoritmos de aprendizaje automático para realizar una predicción más precisa que la de un solo modelo.

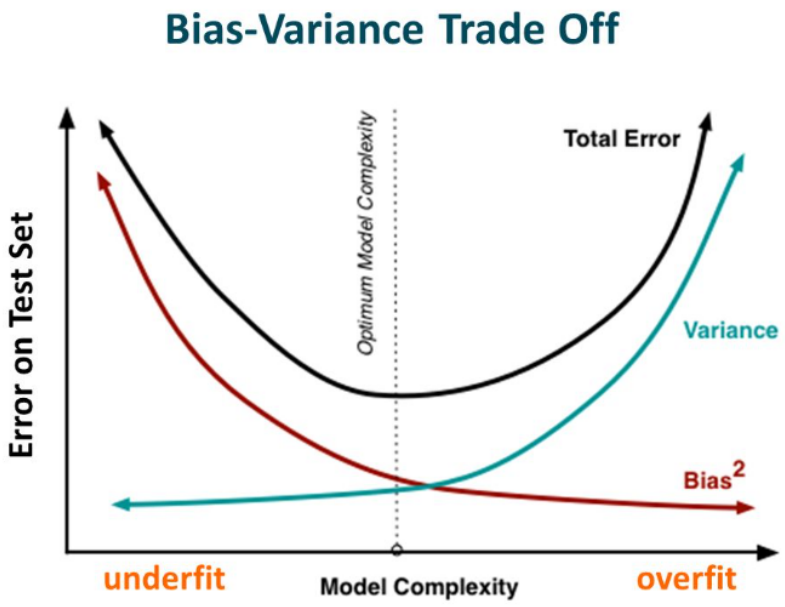
Un modelo Random Forest está formado por un conjunto (ensemble) de árboles de decisión individuales, cada uno entrenado con una muestra aleatoria extraída de los datos de entrenamiento originales mediante bootstrapping). Esto implica que cada árbol se entrena con unos datos ligeramente distintos. En cada árbol individual, las observaciones se van distribuyendo por bifurcaciones (nodos) generando la estructura del árbol hasta alcanzar un nodo terminal. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo.

Los métodos tipo ensamblador están formados de un grupo de modelos predictivos que permiten alcanzar una mejor precisión y estabilidad del modelo.

Un random forest se compone de un número concreto de árboles. Como todos los modelos, un árbol de decisión también sufre de los problemas de sesgo y varianza. Al construir un árbol “pequeño” se obtiene un modelo con baja varianza y alto sesgo. Al incrementar la complejidad del modelo, se reduce el error de predicción debido a un sesgo más bajo en el modelo. En un punto el modelo será muy complejo y se producirá un “overfitting” o sobre-ajuste del modelo el cual empezará a sufrir de varianza alta. Por lo tanto, lo que se busca a la hora de hacer de un

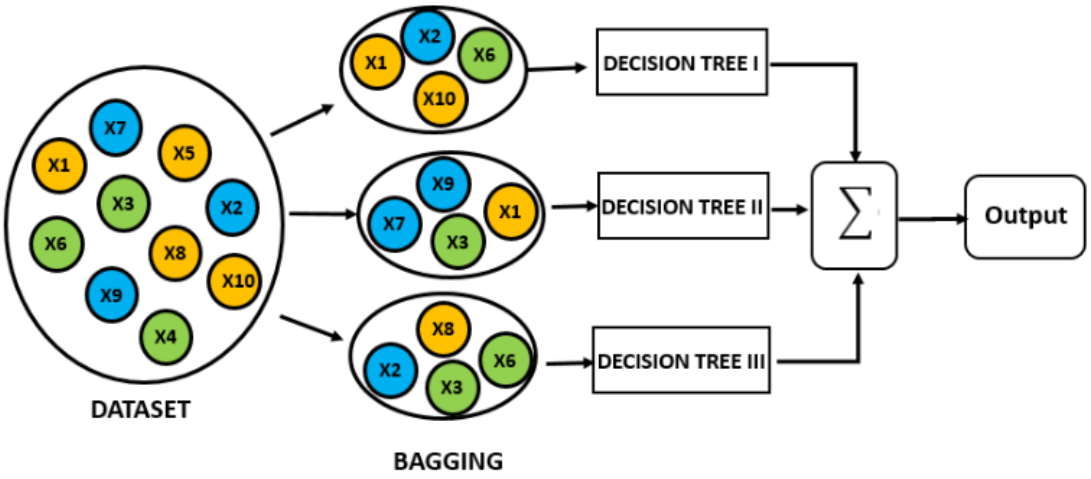
modelo como éste es buscar un punto de equilibrio entre estos 2 errores a esto se le llama "trade-off".

Figura 1: Trade-off en un modelo



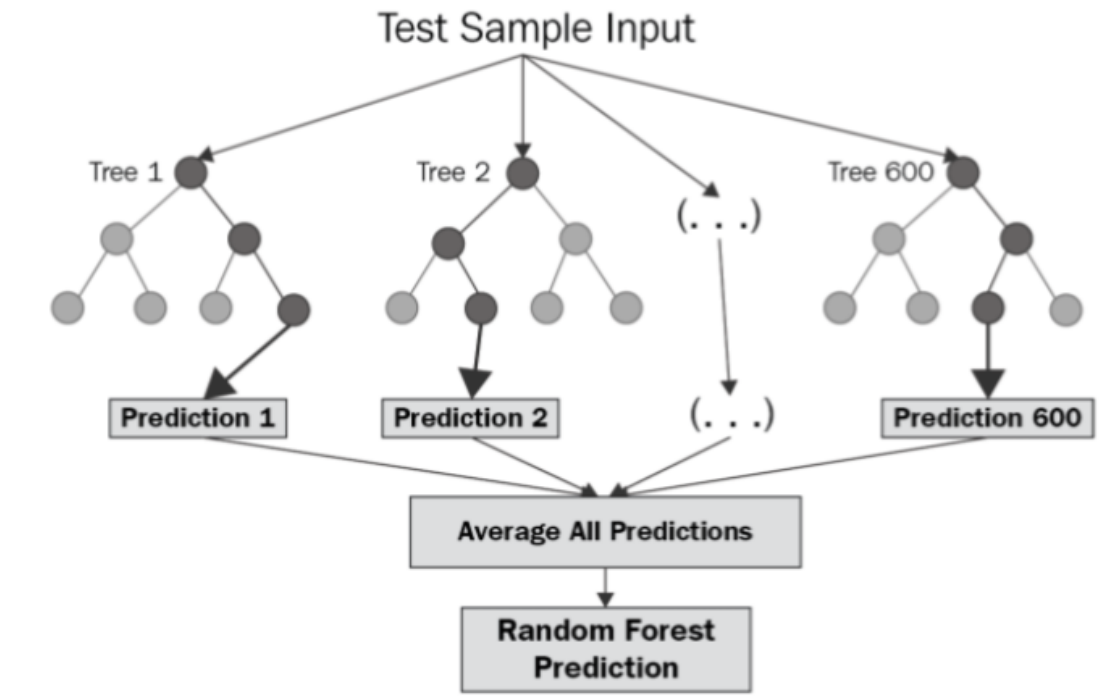
Es importante saber que un Random Forest tiene un tipo de ensamblador llamado bagging. Bagging es una técnica usada para reducir la varianza de las predicciones a través de la combinación de los resultados de varios clasificadores, cada uno de ellos modelados con diferentes subconjuntos tomados de la misma población.

Figura 2: Funcionamiento del proceso de bagging



En resumidas cuentas un random forest crea múltiples subconjuntos de datos, construye múltiples modelos para cada subconjunto y combina los modelos.

Figura 3: Funcionamiento de un random forest



Fuente: (Bakshi, 2020) [3]

API

El término API es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Así pues, podemos hablar de una API como una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones.

Existen APIs para todo, desde para cumplir la función de sistemas de pago en un tienda online o como scrapper en redes sociales en TikTok.

TikTokAPI [4]

Es una API no oficial creada por David Teacher **[4]** que cumple la función de wrapper para TikTok.com en Python. Es decir que facilita al usuario la captura de datos relacionados con publicaciones y usuarios en TikTok. Se compone de funciones que se basan en hacer scraping dentro de la aplicación de TikTok sacando datos a partir de comandos del usuario. Descargando la API, solo tenemos que llamar a funciones como by_hashtag o by_trending para que scrapee todos los datos relacionados con los videos que contienen un hashtag en concreto o los videos que estén en tendencias.

Lo mejor de la Api es la cantidad de datos por video que es capaz de capturar. Gracias a esta API, se puede exprimir cada uno de los vídeos capturando información como la duración del video, descripción, calidad de imagen, autor del video junto a sus estadísticas, autor de la música, número de likes o número de visitas.

Tweepy

Tweepy es una librería que funciona como un "wrapper" para trabajar con la API REST de twitter, facilitando por medio de métodos y objetos en Python la interacción con estos servicios. Esta librería tiene muchas funcionalidades pero la que voy a usar en este proyecto se basa únicamente en capturar el máximo número de datos acerca de tweets. Se pueden extraer tweets que contengan una palabra en específico o que estén en tendencias.

A la hora de scrapear, la API exprime al máximo cada tweet dando datos acerca del tweet en cuestión pero también sobre el autor del mismo. Las variables principales que voy a utilizar en mi proyecto que me proporciona la API son:

- Nombre de usuario
- Texto del tweet
- Enlace de descarga de posibles imágenes o videos adjuntados
- Fecha de publicación
- Estadísticas del tweet como el número de likes, de retweets, de comentario o el número de veces que se ha compartido.
- Estadísticas del usuario como número de seguidores, número de personas a la que sigue o el número de tweets publicados.

Streamlit

Streamlit es una biblioteca de python que permite crear todo tipo de aplicaciones de datos desarrolladas en Python. Estas aplicaciones web son personalizadas para el machine learning y la ciencia de datos.

Uno de los puntos positivos de esta biblioteca es que se puede trabajar con python siendo este el lenguaje de programación que predomina en este proyecto. Esto es beneficioso ya que tanto los modelos como el uso de las api 's se hacen a través de python. Uno de los problemas de usar HTML es que exige aprender a programar este tipo de código desde 0 ya que no es un lenguaje de programación sino que es un lenguaje de marcado de hipertexto o “HyperText Markup Language”. El tipo de programación es totalmente distinto ya que este lenguaje se escribe en su totalidad con elementos que a su vez están constituidos por etiquetas, contenido

y atributos. Sin embargo Streamlit es una biblioteca de python por lo que el uso de los if, else, for, la creación de funciones y demás siguen el mismo funcionamiento que cualquier python al que ya estamos tan acostumbrados.

Usar una biblioteca de python para crear una aplicación hace que se optimice el tiempo ya que este no es invertido en aprender a programar HTML o PHP. La parte positiva y a la vez negativa es que es muy sencillo de implementar pero a su vez tiene muchas limitaciones sobre todo a nivel visual.

2.2 Contexto y justificación

Hoy en día, todos usamos las redes sociales en mayor o menor medida. Pero no es solo que estemos conectados sino que además estamos todo el día recibiendo notificaciones lo que hace que estemos las 24 horas del día pendientes. Esto genera adicción.

Figura 4: Tiempo de uso de las redes sociales diario por persona y por país

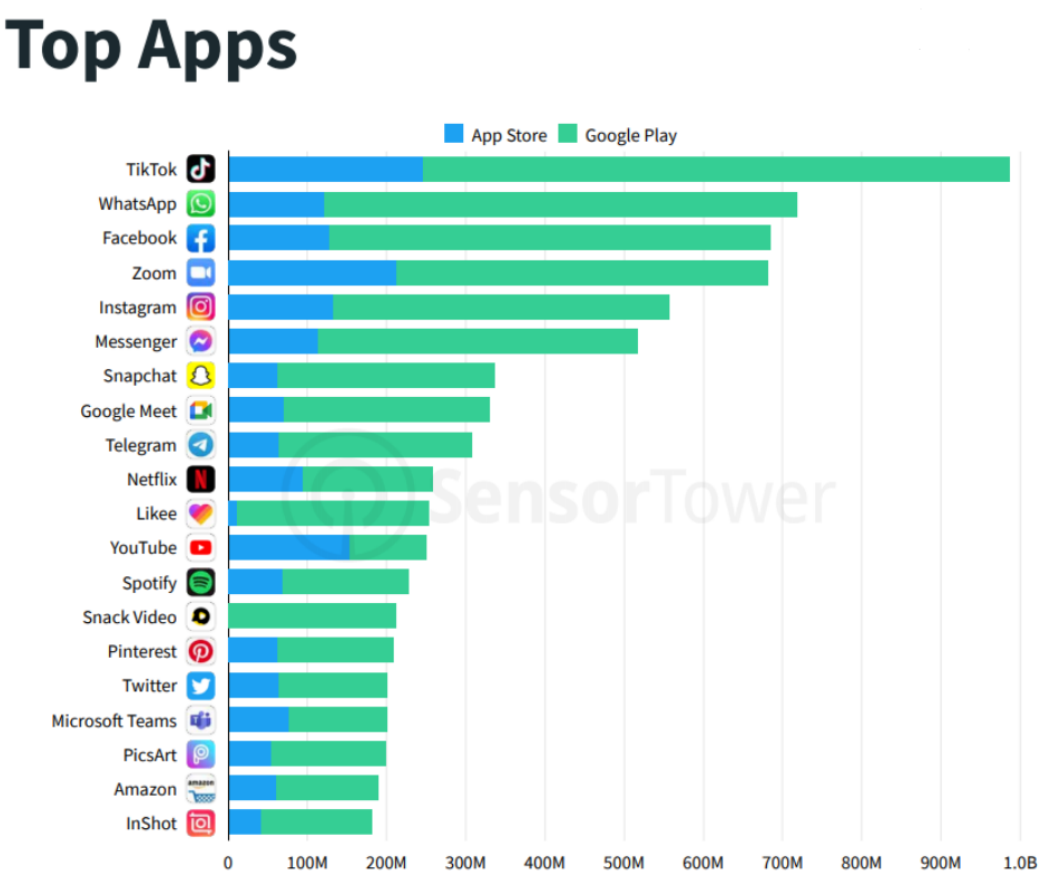
Fuente: (Mónica Mena Roa, 2022) [1]

Las empresas aprovechan esta adicción para promover sus productos o servicios. Pero para alcanzar la máxima difusión, las empresas no solo lo promueven a través de su cuenta personal

sino que contratan personas con mucha audiencia y repercusión que encajen con su imagen de marca. Estas personas se denominan 'influencers' y tienen dos principales fuentes de ingresos, lo que pagan las redes sociales según el número de visualizaciones y likes y lo que pagan las empresas por vender sus productos o servicios.

TikTok es la red social con el crecimiento más exponencial en los últimos años. Con un público joven en su mayoría lo que vuelve a esta aplicación el lugar perfecto para promover productos y servicios por la inocencia y vulnerabilidad de los jóvenes frente a las influencias.

Figura 5: Las aplicación con más descargas en último cuatrimestre de 2021



Fuente (Asselin, 2022) [2]

Los influencers dependen de sus seguidores ya que sus ingresos aumentan o disminuyen de su repercusión. Conseguir seguidores y visitas es muy complicado pero mantener esas cifras lo es más. Mantener la viralidad en tu contenido es algo que no muchos consiguen ya que la originalidad y creatividad no es infinita y la presión por tener que subir publicaciones casi a diario es contraproducente. A esto hay que sumarle la inseguridad del ser humano. Son muchos factores que afectan anímica y psicológicamente al influencer.

Mi proyecto se basa en facilitar al usuario herramientas para combatir estos factores. ¿Cómo? Ofreciendo una interfaz gráfica desde la que poder:

- Predecir el número de likes y visitas que tiene el video que metas como input
- Consultar videos y tweet relacionados con un temática especificada por el usuario
- Consultar videos de TikTok y tweet que estén en tendencias

2.3 Planteamiento del problema

Los influencers de hoy en día, al igual que los artistas, suelen tener un manager para gestionar sus redes sociales y negociar nuevos contratos. Además, los influencer de tiktok también pueden recibir ayuda de la propia aplicación ofreciendo un módulo llamado TikTok Ads Manager [5] que te ayuda a:

- Potenciar tu negocio
- Aumentar tus ventas
- Promocionar tu negocio
- Conseguir más clientes

Sin embargo, un manager o un algoritmo que suple la función de manager no es capaz de predecir el éxito que tendrá un video ni podrá asesorarte con videos y tweets más relevantes relacionados con una temática o con temas que estén de moda (trending). Es decir que no son capaces de evitar la ansiedad y estrés que genera el no saber qué publicar. Ni tranquilizar la inseguridad sobre lo bien o mal qué irá un video.

El proyecto se basa en un aplicación cuyo objetivo es sumplir las funciones que un manager no puede, prediciendo la viralidad y asesorando con los temas mainstream que más favorecen al contenido del usuario.

Contando con:

- Modelos de predicción como Random Forest para predecir el número de likes
- Modelos de procesamiento del lenguaje natural
- Uso de API’s
- Técnicas de scraping para conseguir los temas ‘trending’ de otras aplicaciones según el contenido de los videos del usuario
- Algoritmos de transcripción de video a texto para analizar lo que se dice en cada video

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

El objetivo general del proyecto es crear una aplicación, basada en algoritmos de machine learning que sea capaz de predecir el número de likes que tendrá un video al cabo de x días según el perfil y el video en cuestión. Además de desarrollar una aplicación web dónde los usuarios puedan consultar estas predicciones, los videos y tweets con más repercusión que traten de un tema en específico y los más virales.

3.2 Objetivos específicos

El objetivo general es crear una aplicación compuesta por tres páginas. Un frame para que el usuario pueda consultar la viralidad de un post reflejada en likes. La segunda página consiste en un buscador de videos en TikTok y tweets que tengan la máxima repercusión que traten el tema que el usuario quiera investigar. De esta forma si el usuario quiere publicar un video sobre deportes en la playa, por ejemplo, puede ver todos los videos y tweets que han tenido éxito tratando este tema. La última página está más enfocada en dar soporte a aquellos usuario que no saben qué publicar. Esta página muestra al usuario el tipo de contenido que más repercusión está teniendo en este momento. La gracia de contrastar las tendencias en TikTok con el trending topic de twitter es saber de qué se está hablando fuera de plataforma de TikTok para poder traer estos temas a la misma.

3.2.1 Crear el modelo de predicción

Una aplicación que se basa en modelos de predicción necesita una analítica de datos muy elevada y con una cantidad de datos lo más grande posible. Es por esto que el proceso de ETL es esencial. Este proceso se basa en 3 fases: extracción, transformación y carga de datos.

Extracción:

- Extraer el máximo número de videos en TikTok, capturando el mayor número de características por video para más adelante, cuando seleccione las variables más relevantes, poder tener un amplio abanico de posibilidades.

  Esta extracción se hará directamente sobre la propia aplicación de TikTok mediante querys y técnicas de scraping que vienen incluidas en la API no oficial de la aplicación llamada TikTokApi.

Transformación:

- Leer los datos extraídos en formato JSON desglosando las variables anidadas.
- Solucionar cualquier problema de limpieza de datos relacionado con vacíos, outliers o errores en los formatos de las variables.
- Primer filtrado de variables, desconsiderando variables monótonas (todos los registros son iguales) o únicas (menos los ID).

- Segundo filtrado de variables. Seleccionar las variables más influyentes a priori usando un modelo de regresión llamado GradientBoostingRegressor y viendo cuales son las variables que más han influído con la función feature_importances_.

Carga:

- Cargar los datos finales transformados en un lugar centralizado y exportar a CSV para poder tratar estos datos desde otras aplicaciones

Una vez, el proceso de ETL finaliza es imprescindible saber qué modelo utilizar:

- Estudiar los diferentes algoritmos de predicción y crear una lista con los algoritmos que más se acoplen a los datos y a la variable objetivo.
- Probar los modelos de la lista y realizar una primera optimización cambiando los principales parámetros. Por ejemplo en un Random Forest [3], el número de árboles o la profundidad.
- Decidir cuáles serán las métricas más significativas para escoger el modelo óptimo
- Análisis de los resultados obtenidos
- Realizar los ajustes necesarios en los hyperparámetros, con el fin de escoger el mejor modelo

3.2.2 Crear el asesor de temáticas

Esta página se centra en ofrecer al usuario un abanico de posibilidades acerca de posibles videos que tengan relación con una temática. ¿Cómo? En un principio, la idea era buscar videos en TikTok que se pareciese al tipo de contenido que suele publicar el usuario pero esto hacía que se cerraran muchas puertas ya que lo que interesa al usuario es saber qué tipo de video publicar tratando temas que no estén forzosamente en su zona de confort. La idea de esta página es no cerrar puertas, si el usuario suele publicar videos relacionados con el baloncesto, puede consultar este tipo de contenido para seguir en su línea pero si quiere cambiar de zona también puede.

3.2.3 Crear consultorio de estadísticas

Esta página sigue la misma esencia que la anterior. Es decir, mostrar al usuario una serie de videos de TikTok y Tweets. En este caso, lo que la aplicación va a mostrar son las tendencias. En otras palabras, se podrán ver los videos más virales de TikTok y el trending topic de Twitter. Es esencial estar al corriente de las modas y de los temas del momento. En este caso, la mezcla de TikTok y twitter es aún más interesante ya que en twitter las tendencias reflejan los temas más hablados dentro de un público más amplio tocando temas más relevantes ya que la edad media del usuario es mayor. Es decir que si quieres saber qué está pasando hoy en día en las redes sociales y en el mundo en general, es más interesante leer twitter que ver videos de tiktok los cuales no suelen tener relación entre ellos. En cambio en Twitter se crean hilos de conversación lo que hace que se hable mucho más sobre cada tema. Es por eso que para crear

una imagen global de las tendencias es necesario contrastar las tendencias en TikTok con el trending topic de twitter.

3.3 Beneficios del proyecto

Esta aplicación es la herramienta perfecta para cualquier influencer. Es una fuente de ideas de temáticas para futuros videos y un oráculo para predecir la repercusión que podrá tener un video, según las características del usuario y del contenido del video. Consecuentemente, resolviendo problemas de estrés y ansiedad causadas por las inseguridades del ser humano y la falta de inputs e ideas para seguir contentando al público. No solo sirve para salir del paso sino que puede ayudar a aumentar la repercusión en la red social ya que puede actualizar su contenido y probar distintas temáticas comprobando la repercusión sin la necesidad de publicarlo.

A su vez esta aplicación puede ser utilizada por empresas que quieran contratar a algún influencer. Ya que antes de incluso ponerse en contacto con ellos pueden predecir la repercusión que tendrá el influencer en cuestión si promocionase los servicios o productos de su empresa. Para decirlo con otras palabras, puede servir como primer filtro de selección para una empresa.

Además, la aplicación cuenta con una sencilla interfaz que permitirá al cliente consultar desde cualquier lugar, al ser multiplataforma , de manera sencilla y en tiempo real.

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Planificación del proyecto

Fase 1: Investigación previa

1. Estudio del contexto y el estado del arte de los conceptos y tecnologías del proyecto. Esto incluye la búsqueda de las alternativas y casos de uso actuales
2. Análisis técnico de las tecnologías escogidas, tanto para el componente back-end y el front-end.

Fase 2: Planificación

1. Definir los objetivos principales del proyecto
2. Definir los objetivos específicos del proyecto
3. Determinar el alcance
4. Definir requisitos del proyecto
5. Definir los recursos necesarios

Fase 3: Diseño

1. Diseño de la arquitectura back-end
2. Diseño de la arquitectura front-end
3. Diseño del modelo del algoritmo de transcripción
4. Diseño del modelo de predicción
5. Diseño del proceso de validación

Fase 4: Preparación del entorno

1. Reunir los requisitos necesarios para preparar el entorno de trabajo. Librerías, aplicaciones, plataformas y lenguajes de programación.
2. Creación del entorno virtual
3. Carga de las librerías necesarias para el correcto funcionamiento de los algoritmos y de la aplicación.

Fase 5: Desarrollo e Implementación

1. Proceso ETL:

Creación del proceso de extracción de datos:

Utilización de la API

Automatización del proceso (en tiempo real o en bach con alta frecuencia)

Transformación de datos:

Desarrollar un pipeline adecuado de ingesta y preparado/limpieza de datos. Incluye determinar las librerías necesarias para ello.

2. Hacer un análisis exploratorio de los datos para resumir sus principales características, empleando métodos de visualización de datos (POWER BI)
3. Creación del modelo de predicción:
    - Estudiar las opciones de algoritmos para la tarea de predicción de viralidad.
    - Iniciar la fase de entrenamiento modificando los hiperparámetros con el fin de optimar el modelo
    - Fijar las métricas que se utilizarán para la evaluación
    - Iniciar fase de test.
    - Implementación del proceso en la página web estableciendo que el input del formulario principal del frame sea un archivo MP4.
4. Creación de la funcionalidad para consultar videos de tiktok y tweets con relación a una temática:
    - Conectar las API de TikTok y Twitter al entorno virtual
    - Utilizando la librería de TikTokApi y usando funciones como by_hashtag extraer videos que traten de una temática introducida por el usuario
    - Utilizando la librería de Tweepy extraer tweet que traten de una temática introducida por el usuario
5. Creación de la funcionalidad de consultar las tendencias en TikTok y el trending topic en Twitter
    - Conectar las API de TikTok y Twitter al entorno virtual
    - Extraer videos que estén en tendencias en TikTok usando la función de la librería TikTokApi llamada by_trending
    - Extraer los tweets que estén en trending topic en twitter usando la librería Tweepy

Fase 6: Pruebas y conclusiones

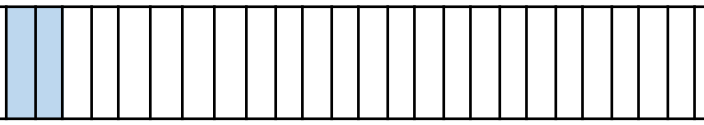
- Periodo de pruebas, refuerzo e identificación de disfunciones. Se deberán aplicar los parches y las medidas necesarias para arreglar posibles fallos de funcionamiento
- Redacción de la memoria y preparación de la presentación
- Presentación del proyecto.

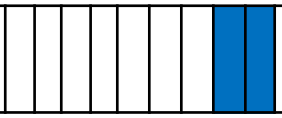
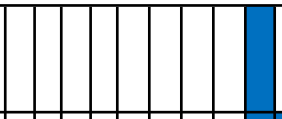
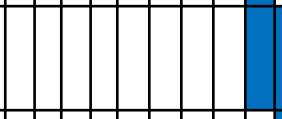
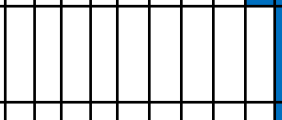
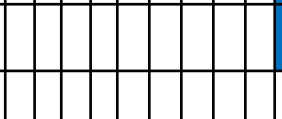
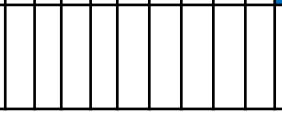
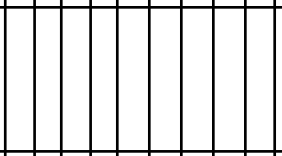
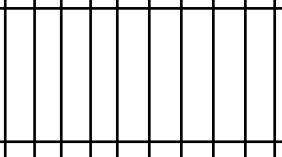
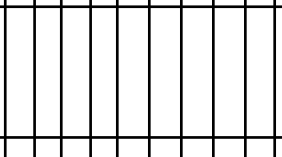
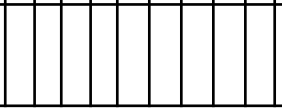
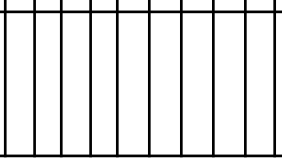
Tabla 1: Gantt de seguimiento

Gantt:

Nombre de la tarea	MES						
	ENERO	FEBRERO	Marzo	ABRIL	MAYO	JUNIO	Nº Horas
INVESTIGACIÓN PREVIA							
Contexto y estado del arte							

17

Análisis técnico de las tecnologías escogidas		
PLANIFICACIÓN		
Definir los objetivos principales del proyecto		12
Definir los objetivos específicos del proyecto		
Determinar el alcance		
Definir requisitos del proyecto		
Definir los recursos necesarios		
DISEÑO		
Diseño de la arquitectura back-end		13
Diseño de la arquitectura front-end		
Diseño del modelo del algoritmo de transcripción		
Diseño del modelo de predicción		
Diseño del proceso de pruebas		
PREPARACIÓN DEL ENTORNO		
Reunir los requisitos necesarios para preparar el entorno de trabajo		7
Creación del entorno virtual		
Carga de las librerías necesarias		
DESARROLLO E IMPLEMENTACIÓN		
ETL		
Creación del proceso de extracción de datos		16
Transformación de datos		

Hacer un análisis exploratorio de los datos		
Creación del modelo de predicción:		
Estudiar las opciones de algoritmos		18
Entrenar modelo modificando los hiperparámetros		
Fijar las métricas para la evaluación		
Testear el modelo		
Implementación del proceso en la página web		
CREACIÓN DE LA PÁGINA PARA CONSULTAR TEMÁTICAS		
Conectar las API de TikTok y Twitter al entorno virutal		14
Crear formulario para que el usuario pueda introducir una temática		
Automatizar las consultas a la API de TikTok para extraer los videos		
Automatizar las consultas a la API de Twitter para extraer los tweets		
Imprimir resultados de los resultados		
CREACIÓN DE LA PÁGINA PARA CONSULTAR LAS TENDENCIAS		
Automatizar las consultas a la API de TikTok para extraer los videos en tendencias		11
Automatizar las consultas a la API de Twitter para extraer los tweets que estén en el trending topic		

Imprimir resultados de los resultados																									
PRUEBAS Y CONCLUSIONES																									
Aplicar los parches y las medidas necesarias para arreglar posibles fallos de funcionamiento																									20
Redacción de la memoria y preparación de la presentación																									
Presentación del proyecto																									

4.2 Descripción de la solución, metodologías y herramientas empleadas

El objetivo general del proyecto es crear una aplicación que se compone de 3 partes principales:

- Un predictor de viralidad
- Un consultor de publicaciones relacionadas a una temática
- Un consultor de publicaciones en tendencias

Cada apartado requiere de una serie de tecnologías que van a intervenir para crear y agilizar el proceso.

Figura 6: Pasos para crear una modelo predictivo

Fuente: (Leonardo Gamboa Uribe, 2020) [6]

Creación del entorno virtual

Un entorno virtual de Python no es más que un ámbito de trabajo aislado del sistema principal o de otros ámbitos. Esto nos permite ejecutar una determinada aplicación, sin afectar al sistema u otros procesos. Igualmente nos permite instalar determinadas librerías o versiones de las librerías, sin que afecte al resto.

Un entorno virtual facilita ejecutar en un mismo sistema diferentes versiones de una misma aplicación.

En resumidas palabras, el fin de usar un entorno virtual es que las librerías y paquetes que vaya a tener la aplicación no afecten al sistema y viceversa.

Para crear un entorno virtual es tan fácil como crear una carpeta en local dónde el proyecto va a desarrollarse. Desde el cmd entramos en la carpeta y con un simple comando creamos el entorno virtual de python.

Lo bueno es que dentro de una misma carpeta podemos tener distintos entornos lo que nos permite ejecutar python en el entorno que creemos o en un python 3 de jupyper notebook.

Extracción de los datos usando TikTokAPI

En un principio, la opción más óptima y rentable para capturar datos acerca de video en tiktok era usando técnicas de scraping directamente en la página web de TikTok. Para hacer el scraping utilicé en primera instancia ParseHub como herramienta. El problema es que un proceso de scraping en ParseHub no es fácil de automatizar y menos aún incluirlo como función dentro de un código python y esto último es un requerimiento esencial. Viendo que esta opción no era posible, opté por hacer el scraping desde python usando librerías como beautifulsoup y request. Fue así como hice la primera extracción de datos sacando los datos más “a la vista” como el número de likes, visitas y comentarios que tenía cada publicación. El scrapping se hacía dentro de la página web de TikTok, en el apartado “Para ti” dónde aparecen los videos que la red social te recomienda según tu perfil. Pero la idea no es capturar los videos que me recomienden a mi personalmente sino que el objetivo es capturar videos que estén en tendencias o los videos que contengan un hashtag en concreto

Instalación y uso de la API

Una vez que el entorno virtual está creado, la instalación de la API es sencilla. Lo primero que hay que saber es que esta api no es oficial de Tik Tok sino que ha sido desarrollada por davidteather [4] y el acceso es público y se encuentra en la plataforma GITHUB. En su repositorio de Github encontramos toda la documentación necesaria para la implementación y utilización de la API además de por su puesto, todos los programas utilizados para crear esta API. Realmente, sin entrar mucho en detalle, esta api se compone de una serie de funciones de scraping basándose en parámetros como el hashtag, trending, nombre de usuario (nickname) y demás.

La implementación es muy sencilla ya que solo es necesario la ejecución de dos comandos en la terminal del entorno virtual. Uno para instalar la api usando un 'pip install' y el otro para instalar el playwright que es una biblioteca Node.js para automatizar Chromium, Firefox y WebKit con una única API.

Una vez tenemos la API instalada, siguiendo la documentación de GITHUB se pueden hacer una serie de consultas básicas para verificar que la API está funcionando correctamente. Pero al contrario de lo que esperaba, la API no respondía como debería. Haciendo consultas como por ejemplo los 30 primeros videos en tendencias, los resultados eran conjuntos vacíos o errores. Centrándome en los errores, la salida del terminal mencionaba un error acerca del playwright diciendo que acababa de ser instalado o actualizado y que hacía falta ejecutar un comando para descargar los nuevos navegadores. Sin embargo, al ejecutar el comando que me recomendaba, el error persistía. Investigando en internet, vi que este error es frecuente y que una posible solución es instalar una versión más antigua, en concreto la versión 1.17.2 . Y efectivamente esto solucionó el problema.

Hablando de versiones, la versión más frecuente de la API de TIkTokAPI funciona correctamente pero la mayoría de la documentación se basa en una versión más antigua. Por temas de comodidad, he decidido usar una versión más antigua ya que realmente al basarse en algoritmos de scraping, los datos extraídos no son correspondientes temporalmente a la versión de la api ya que la api como tal no contiene datos.

Lectura de los datos proporcionados por la api

La API no oficial de tiktok no da los datos en un formato cómodo ya que los como diccionarios. Cada video extraído de tiktok es un diccionario por lo que el conjunto de datos es una lista de diccionarios. El resultado final es un JSON. Un json se compone de variables planas "flat" que son aquellas variables con datos en crudo y de variables anidadas que son variables que contienen más variables. Por lo tanto para capturar el 100% de los datos es necesario a la hora de leer el JSON desglosar las variables anidadas.

Acumulando los videos obtengo un conjunto de datos con esta forma:

Figura 7: Datos en formato JSON


```
{
  "0": {
    "id": "6839416095586159878",
    "desc": "Repost \ud83d\ude07 #catsoftiktok #viral #funny #cute #cutness #foryoupage",
    "createTime": 1592425654,
    "video": {
      "id": "6839416095586159878",
      "height": 1024,
      "width": 576,
      "duration": 13,
      "ratio": "720p",
      "cover": "https://p77-sign-va.tiktokcdn.com/obj/tos-maliva-p-0068/80a7eece06774b96ae8b68df2fb524b2?x-expires=1
      "originCover": "https://p77-sign-va.tiktokcdn.com/obj/tos-maliva-p-0068/da52d1bf0a1d4574830cce5e6fccbbfc_15924
      "dynamicCover": "https://p77-sign-va.tiktokcdn.com/obj/tos-maliva-p-0068/8686754c69594cbd911c685c8f788769_1592
      "playAddr": "https://v16-webapp.tiktok.com/a4465d5c3f4a01809d7cbd4c8b3492bd/627806d5/video/tos/useast2a/tos-us
      "downloadAddr": "https://v16-webapp.tiktok.com/a4465d5c3f4a01809d7cbd4c8b3492bd/627806d5/video/tos/useast2a/to
      "shareCover": [
        "",
        "https://p77-sign-va.tiktokcdn.com/tos-maliva-p-0068/da52d1bf0a1d4574830cce5e6fccbbfc_1592425657~tplv-tiktok
        "https://p77-sign-va.tiktokcdn.com/tos-maliva-p-0068/da52d1bf0a1d4574830cce5e6fccbbfc_1592425657~tplv-tiktok
      ],
      "reflowCover": "https://p77-sign-va.tiktokcdn.com/obj/tos-maliva-p-0068/80a7eece06774b96ae8b68df2fb524b2?x-exp
      "bitrate": 1358289,
      "encodedType": "normal",
      "format": "mp4",
      "videoQuality": "normal",
      "encodeUserTag": "",
      "codecType": "h264",
      "definition": "720p"
    },
    "author": {
      "id": "6832141644758975494",
```


Lo primero marcado en amarillo es la posición del video dentro de la lista de videos en este caso es el primer video de la lista.

Lo segundo marcado es una variable 'flat' ya que solo contiene el dato en crudo en este caso la descripción del video

Lo tercero marcado es una variable anidada ya que dentro del video tenemos distintas características como la duración, el ratio o las dimensiones.

Para poder trabajar correctamente con los datos es necesario transformar las variables para que todas sean 'flat'. Es decir que el objetivo es crear un Dataframe.

Al leer el json con la librería JSON se obtiene este dataframe:

Figura 8: Dataframe de los datos de los videos de TikTok sin desglosar

	id	desc	createTime	video	author	music
0	6839416095586159878	Repost 😇 #catsoftiktok #viral #funny #cute #cu...	1592425654	{'id': '6839416095586159878', 'height': 1024, ...	{'id': '6832141644758975494', 'uniqueId': 'bri...	{'id': '6839416100652894982', 'title': 'origin...
1	6941010483021466885	Dinky the chipmunk can fit more into his	1616079942	{'id': '6941010483021466885',	{'id': '6798572058492339206',	{'id': '6941010281950628614',

Se aprecia que los datos de las variables video, author, music y demás variables que no aparecen en la imagen, son imágenes.

Para poder desglosar el contenido de las variables anidadas es necesario la utilización de bucles obteniendo este dataframe:

Figura 9: Dataframe de los datos de los videos de TikTok

	id	desc	video_id	video_height	video_width	video_duration	video_ratio	video_cover
0	6839416095586159878	Repost 😇 #catsoftiktok #viral #funny #cute #cu...	6839416095586159878	1024	576	13	720p	https://p77-sign-va.tiktokcdn.com/obj/tos-mali...
1	6941010483021466885	Dinky the chipmunk can fit more into his mouth...	6941010483021466885	1024	576	15	720p	https://p77-sign-va.tiktokcdn.com/obj/tos-mali...
2	6916965467907263750	Wee! 🎮 #Wee #fyp #jump #jumping #viral #weeme...	6916965467907263750	960	540	8	720p	https://p77-sign-va.tiktokcdn.com/obj/tos-mali...

Variables principales:

'id': Identificador del video

'desc': Descripción que ha puesto el autor del video

'video_id': Identificador del video

'video_height': Altura de los frames del video

'video_width': Anhura de los frames del video

'video_duration': Duración del video

'video_ratio': Proporción entre el ancho y la altura de un vídeo

'video_cover': Link que dirige a la portada de un video

'video_playAddr': Lnk para ver el video

'video_downloadAddr': Link para descargar el video

'video_bitrate': Es la cantidad de información que se reproduce por segundo, cuanto mayor sea esa información, mejor será la calidad del vídeo

'video_format': Formato del video (por lo general MP4)

'video_definition': Calidad del video. Por ejemplo 720p.

'author_id': Identificador del autor

'author_nickname': Nombre de usuario (alias)

'author_avatarThumb': Foto de perfil

'author_avatarMedium': Foto de perfil. Tamaño: mediano

'author_avatarLarger': Foto de perfil. Tamaño: grande

'author_signature': Estado/descripción del autor

'author_verified': Booleano de si el usuario está verificado

'music_id': Identificador de la música

'music_title': Título de la música

'music_playUrl': URL para reproducir el audio

'music_coverThumb': Portada del audio

'music_coverMedium': Portada del audio. Tamaño: Medio

'music_coverLarge': Portada del audio. Tamaño: Grande

'music_authorName': Nombre del autor del audio

'music_original': Boolean de si el audio es original

'music_duration': Duración del audio

'music_album': Album del audio

'stats_diggCount': Número de likes del video

'stats_shareCount': Número de veces que se ha compartido el video

'stats_commentCount': Número de comentarios que tiene el video

'stats_playCount': Número de veces que se ha reproducido el video

'authorStats_followingCount': Número de personas a las que sigue el usuario

'authorStats_followerCount': Número de seguidores del usuario

'authorStats_heartCount': Número de likes del usuario en total entre todas las publicaciones

'authorStats_videoCount': Número de videos que tiene publicado el usuario

'authorStats_diggCount':

'authorStats_heart':. Número de likes totales del usuario

Transformación de datos

Transcriptor de video a texto

A la hora de analizar un video, datos como la duración, el ratio, la calidad de imagen o el nombre del autor son importantes pero para poder darle más precisión al modelo y poder diferenciar más los videos dentro de un mismo usuario es esencial tener en cuenta la descripción del video y el contenido del video.

Lo importante es saber qué queremos capturar del video y cómo hacerlo. Por lo tanto, lo mejor era capturar el contenido del video es decir la temática y las palabras utilizadas. El problema es que la API no extrae información acerca de lo que ocurre en el video. Lo que más se puede acercar es la descripción que pone el usuario en el video cuando lo publica pero realmente no es la descripción de lo que ocurre en el video sino que es un título o un comentario acerca del video junto a unos hashtags. Con los datos que nos ofrece la API se podría hacer un modelo de predicción bastante acertado (más adelante lo expondré) pero el problema se centra en que realmente el modelo no tendría en cuenta el contenido del video sino que se basaría únicamente en las estadísticas del usuario y en características superficiales como la duración del video, la calidad o el nombre del usuario. Además, si al modelo le metemos dos videos del mismo usuario, la predicción de likes y visita va a ser prácticamente la misma ya que las estadísticas del usuario (likes, visitas, número de videos publicados del usuario totales) coinciden y el resto de variables que diferencian los videos, influyen menos. Es por esto que para tener mayor precisión y para basarnos en conceptos más allá de las estadísticas del usuario, es necesario extraer el contenido del video.

Como he dicho antes, la API no extrae en una variable tipo texto lo que se dice en el video. Sin embargo, nos ofrece un link de descarga de los videos. Esta variable se llama "video_downloadAddr". Entonces, para sacar el contenido de los videos, es necesario automatizar los siguientes pasos:

1. Teniendo el link de descarga, descargar el video en una dirección concreta y con un nombre identificable
2. Transformar el video que está en MP4 a formato WAV que es un formato común de audio
3. Aplicar algoritmo que transcribe lo que se dice en el audio a texto. Este algoritmo se llama SpeechRecognition [7]
4. Splitear el texto para crear un vector de palabras

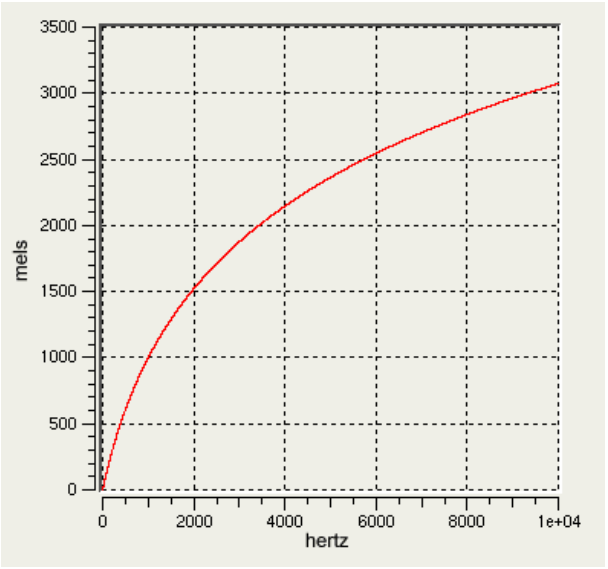
Los primero 2 pasos son fácilmente realizables usando librerías como wget para descargar el video y moviepy.editor [8] para utilizar operaciones básicas como composición de video y procesamiento de video para poder manipular el video y convertirlo en formato audio WAV.

¿Cómo funciona un modelo de transcripción de texto?

Detrás de un transcriptor de texto hay una red neuronal convolucional (CNN).

Para extraer las características distintivas del habla, primero es necesario aplicar un procedimiento de codificación de voz denominado Coeficiente Cepstral de Frecuencia de Mel (MFCC). Los MFCC muestran las características locales de la señal de voz asociadas al tracto vocal. La particularidad básica es que en MFCC las bandas de frecuencia están situadas logarítmicamente.

Figura 10: Escala Mel-Hz



Fuente: Wikipedia (2006)

Se utiliza esta escala porque por encima de 500 Hz., los intervalos de frecuencia espaciados exponencialmente son percibidos como si estuvieran espaciados linealmente, de esta forma, cuatro octavas en la escala de hercios (por encima de 500 Hz.) se comprimen como dos octavas en la escala mel.

Para obtener los vectores de MFCC se suele utilizar el programa Praat.

Pasos que aplica el programa Praat:

1. Aplicar un pre-énfasis a la señal de voz, que aplana espectralmente dicha señal (realza las altas
2. frecuencias), haciendo que su procesamiento sea menos susceptible a truncamientos.
3. Aplicar una ventana de Hamming a la señal obtenida tras el preénfasis, para acotarla.
4. Calcular los valores del banco de filtros distribuidos en frecuencia, según la escala Mel.
5. Calcular el logaritmo de dicho banco, y mediante la DCT obtenemos los coeficientes cepstrales de las frecuencias de mel, que formarán el vector de MFCC.

Figura 11: Transformación de audio a matrices de 48 x 13 (Codificación MFCC)

Fuente: (Machine Learning, 2020)

Una vez hemos convertido las palabras individuales en matrices con los coeficientes cepstrales de frecuencia de Mel, se entrena la red neuronal convolucional.

Este red neuronal pese a que funciona tiene dos graves problemas:

- Es un modelo muy lento. (3 minuto para transcribir un audio de 20 segundos)
- El modelo no es lo suficientemente preciso. Su R^2 es de 0,45.

Es por estos dos motivos por lo que finalmente opté por el uso de una librería de python que a su vez usa una API de google para transcribir.

Utilización de la librería SpeechRecognition de python

Como la red neuronal no era lo bastante precisa, he decidido utilizar una librería de python llamada SpeechRecognition que contiene funciones para tratar audio y transcribirlo.

La instalación y la utilización de la librería es muy simple e intuitiva [11]. Esta librería tiene sus propias funciones internas de tratamiento de audio pero lo realmente útil de la librería es que hace querys a una api de google para transcribir el audio.

He usado esta librería para 3 propósitos:

- Para reducir el ruido de los audio llamando a la función adjust_for_ambient_noise().
- Para sacar la información del audio utilizando record().
- Para transcribir el audio utilizando la función recognize_google que llama a su vez a una API de google.

Con solo estos 3 simples pasos he podido transcribir de la forma más eficiente los videos.

Resultados:

Video original:

https://drive.google.com/file/d/15-mP-yaU2VV1OZH2bsrkEIdghHG-sidX/view?usp=sharing

Audio original:

https://drive.google.com/file/d/1BndYOE8GGcSDkF6wUaTpb2e_QCCY4FUe/view?usp=sharing

Texto transcrito:

```
['tengo', 'dos', 'noticias', 'una', 'buena', 'y', 'una', 'mala', 'no', 'salía',
'contigo', 'y', 'cuál', 'es', 'la', 'mala']
```

Limpieza de datos

Es importante tener siempre en cuenta que el objetivo es predecir el número de likes y visitas que va a tener una publicación en x días. Entonces sabemos que será un modelo de regresión. En estos modelos las variables tienen que ser numéricas o categóricas pero no pueden ser de tipo texto. Además, en un modelo de regresión, no pueden haber datos vacíos, ni indeterminados (infinito).

Estas son las variables que contienen vacíos junto con el número de vacíos:

Tabla 2: Número de vacío por variables

	0
video_playAddr	33
video_downloadAddr	33
video_bitrate	33
video_encodedType	33
video_format	33
video_videoQuality	33
video_encodeUserTag	33
video_codecType	33
video_definition	33

A primera vista vemos que todas estas variables tienen el mismo número de vacíos lo que me hace pensar que posiblemente haya 33 videos dónde todas estas variables están vacías. Es decir, que cuando en una publicación falta, por ejemplo, el dato de “video_downloadAddr”, el resto de datos de las variables de la lista anterior están vacíos.

Comprobación:

Tabla 3: Dataframe con los vacíos

	video_playAddr	video_downloadAddr	video_bitrate	video_encodedType	video_format	video_videoQuality	video_encodeUserTag	video_codecType	video_de
37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
38	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
39	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
43	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
45	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
53	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
54	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
69	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
77	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
86	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
87	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
93	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Efectivamente, hay 33 publicaciones que no tienen estos datos. Estos datos no son recuperables ni predecibles por lo que no es posible rellenarlos. Por lo tanto la única solución es prescindir de estas 33 publicaciones y no es problema ya que no ocupan ni un 1,7% del total de publicaciones.

Quitar duplicados

La extracción de los datos se hace mediante querys a la API de TikTokApi indicando como parámetro un hashtag (por ejemplo #viral) y a su vez la API hace scrapping en la aplicación de TIKTOK para extraer los datos. El número de videos por hashtag que se pueden extraer no es infinito entonces para extraer el máximo de publicaciones he capturado datos a partir de muchos hashtag distintos. Esto me ha servido para ampliar mi base de datos, lo que me ha permitido nutrir más mis modelos que más tarde trataremos. En una publicación se pueden poner los hashtag que quieras. Es por esto que es posible que una misma publicación se extraiga a partir de hashtag distintos.

Ejemplo:

Tabla 4: Ejemplo de usuario incluído en más de un hashtag

	author_nickname	desc	video_height	video_width	video_duration	humor	viral	trending	madre	padre	novia	novio	comida	deporte	sport
55	Thenoezepeda	when mom hates your new girlfriend😂 #mexicanti...	1024	576	71	0	0	0	1	0	0	0	0	0	0
168	Thenoezepeda	when mom hates your new girlfriend😂 #mexicanti...	1024	576	71	0	0	0	0	0	1	0	0	0	0
137	Thenoezepeda	when mom hates your new girlfriend😂 #mexicanti...	1024	576	71	0	0	0	0	0	0	1	0	0	0

Esta publicación ha sido capturada en los hashtags #novio, #novia y #madre.

Que suceda esto no es un problema ya que al estar en hashtag diferentes, se analizará como publicaciones diferentes. El problema es que se repita en un mismo hashtag esto puede suceder ya que para alimentar la base de datos, se repiten las querys a la API con los mismos hashtags ya que las publicaciones en cada hashtag se actualizan pero es posible que se mantenga alguna publicación. Es por eso que pueden ocurrir situaciones como esta:

Tabla 5: Ejemplo publicación que se repite en un mismo hashtag

	author_nickname	desc	humor	viral	trending	madre	padre	novia	novio	comida	deporte	sport
64	CHRXSTOPHER	y'all asked for more, so here's CAMP ROCK 🎸🤣 #...	0	0	1	0	0	0	0	0	0	0
68	CHRXSTOPHER	y'all asked for more, so here's CAMP ROCK 🎸🤣 #...	0	0	1	0	0	0	0	0	0	0

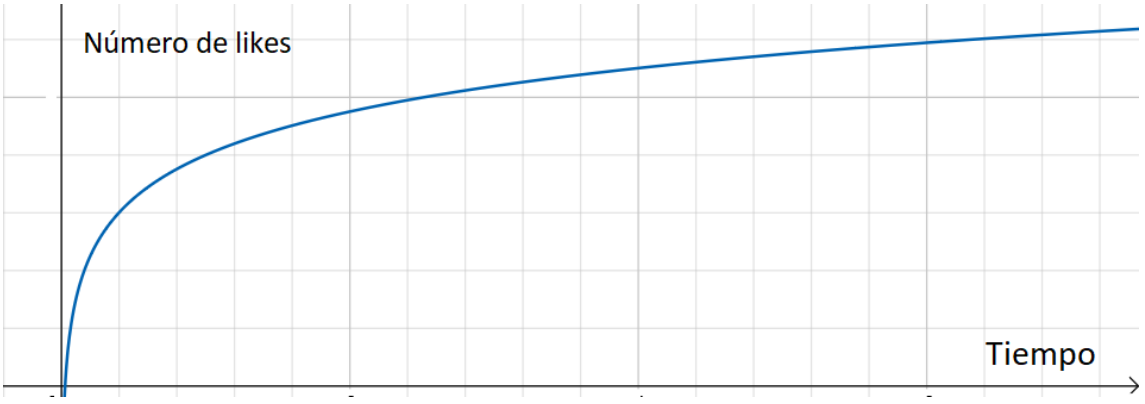
Por lo tanto es necesario eliminar las publicaciones dónde se repitan los valores estas variables:

- author_nickname
- desc
- video_height
- video_width
- video_duration
- Lista de hashtag

Transformación variable Createtime

Una de las variables más relevantes es cuando se creó la publicación. Es esencial ya que el número de likes no es el mismo cuando justo se acaba de publicar a cuando lleva 5 días. Esto se debe a que la evolución de los likes en el tiempo se parece a una función logarítmica. El número de likes aumentar rápidamente al principio y poco se va reduciendo el aumento hasta que se mantiene.

Figura 12: Evolución de los likes en el tiempo



El problema es que la fecha de creación de la publicación viene dada en un formato numérico llamado POSIX timestamp [12] y para saber la fecha es necesario importar una librería que se llama datetime y usar una función que transforma la fecha numérica en formato de fecha datetime. Con esta función conseguimos esto:

```
Fecha en formato POSIX timestamp: 1650209733
Fecha en formato dateTime: 2022-04-17 17:35:33
```

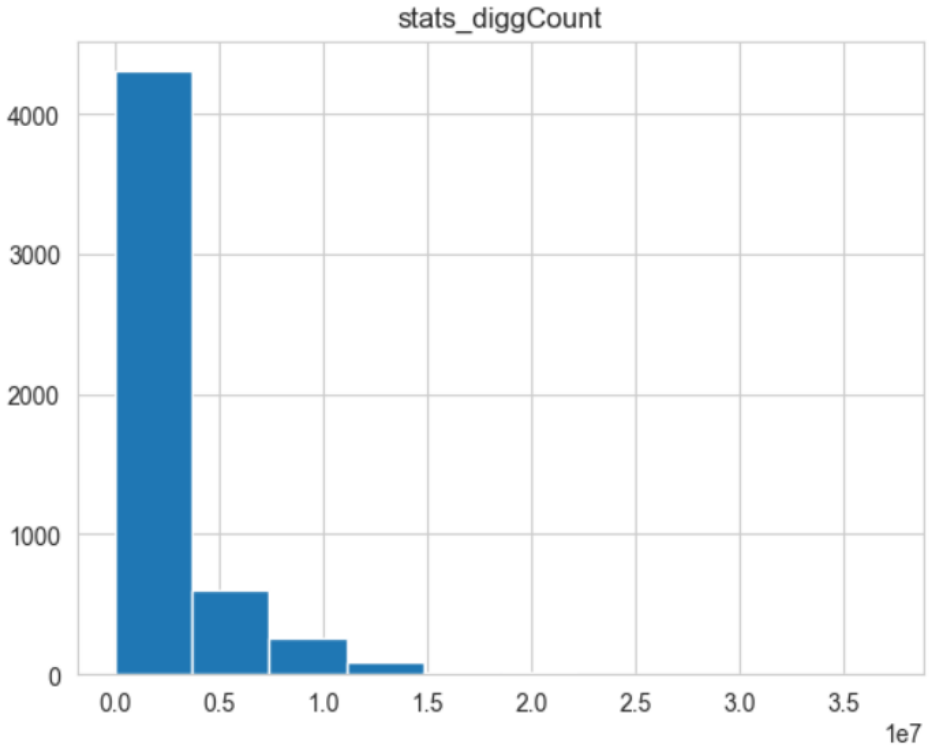
Pero lo importante no es el día en el que se creó sino el tiempo que ha pasado entre que se creó la publicación y la captura de los datos por la API. Entonces una vez que tenemos la fecha de creación solo hay que restar la fecha de creación a la fecha de la captura. Obteniendo:

```
Fecha creación en formato POSIX timestamp: 1650209733
Fecha creación en formato dateTime: 2022-04-17 17:35:33
Fecha hoy en formato dateTime: 2022-05-24 22:15:54.992449
Días que han pasado: 37
```

Tratamiento de los outliers de la variable objetivo (número de likes)

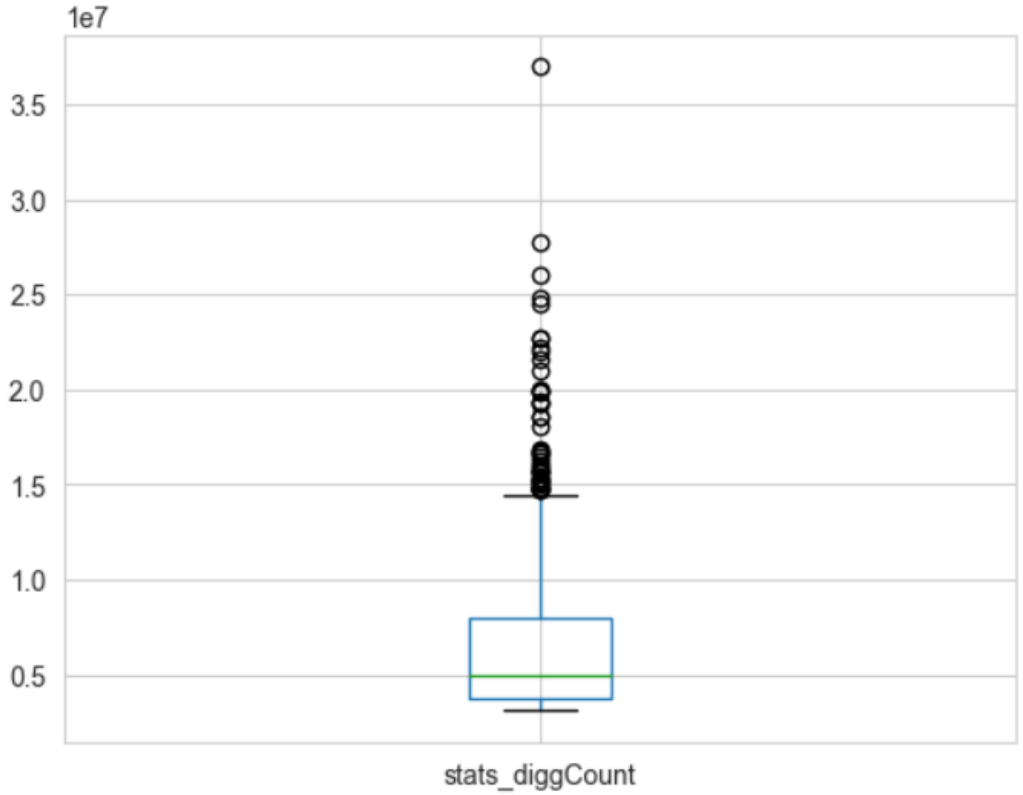
Para optimizar un modelo es importante que la variable que queremos predecir no tenga outliers. Los outliers son “ruido” para el modelo. Ningún modelo va a predecir un outlier lo que provoca que el error aumente ya que la distancia entre lo predicho que el valor atípico va a ser importante. Para ver de manera gráfica la distribución y los posibles outliers de la variable objetivo he optado por un histograma y una diagrama de caja y bigotes:

Figura 13: Histograma del número de likes por publicación



Podemos ver que este histograma sigue una distribución racional básica (inversa). Vemos que la mayoría de los datos están focalizados al principio y repentinamente descienden a gran velocidad. Es decir que la mayoría de las publicaciones tienen entre 0 y 3 millones de likes pero realmente esto no es representativa ya que los límites de la agrupación son demasiado grandes. El problema de este histograma es que se ve afectado por los outliers. El histograma toma como escala del eje x el mínimo y el máximo y luego lo divide en número de barras pero al haber outliers, el máximo está disparado y es por eso por lo que vemos ninguna barra a partir de la mitad del gráfico. Este histograma también nos notifica que los outliers van a ser aquellos datos demasiado elevados y no los pequeños. Es decir que en un diagrama de caja y bigotes, los outliers serán los que superen el Q3. Vamos a comprobarlo:

Figura 14: Diagrama de caja y bigotes de la variable "Número de likes"

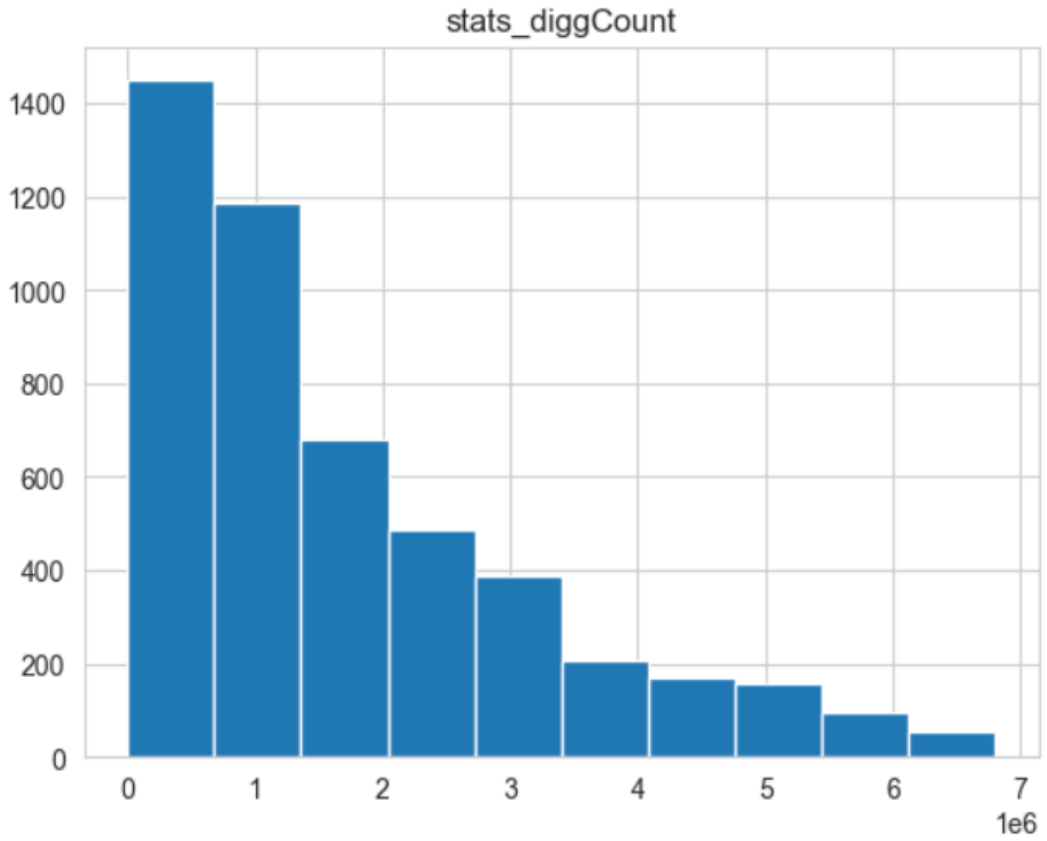


Efectivamente, vemos claramente que hay un gran número de outliers. El diagrama de caja y bigotes considera que a partir de 14 millones de likes la publicación es un outlier. Viendo la forma del diagrama podemos afirmar lo que el histograma señalaba. La mayoría de los datos están cerca del mínimo y poco a poco se van expandiendo más lo que significa que cada vez los datos están más dispersos. Al dispersarse hace que veamos en el histograma menos frecuencia en cada barra.

En resumidas cuentas es esencial tratar estos outliers. Eliminar estos outliers es sencillo. Lo primero que hay que hacer es calcular lo que llamamos el Upper Whisker que es el bigote superior. Para calcularlo hay que restar Q3 y Q1 para saber el valor del rango intercuartílico y luego multiplicarlos por 1,5. Una vez está calculado el Upper Whisker solo hay que filtrar los datos y eliminar aquellas publicaciones que lo superen.

Quitando los outliers obtenemos el siguiente histograma:

Figura 15: Distribución de los likes después de limpieza de outliers



Podemos ver que, efectivamente, la distribución sigue una función inversa pero ahora los datos están un poco más repartidos.

Hay otras variables que contienen outliers como el número de seguidores del usuario pero que no las he corregido porque al ser variable de entrada, es posible que usuario tenga más seguidores de los normal. Lo que no se puede permitir es que la variable que queremos predecir tenga outliers.

Modelo de regresión (Random Forest) - Validaciones (Predicción de likes)

El objetivo es predecir el número de likes que va a tener una publicación después de un número determinado de días. La variable objetivo es una variable numérica por lo que lo más lógico es hacer un modelo de regresión. Antes de desarrollar el modelo de regresión, no quise cerrar puertas y estudié posibles opciones como un modelo de clasificación que predijese si una publicación será viral o no viral. Lo complicado era definir a partir de qué punto una publicación es viral. Otra opción fue segmentar por franjas de likes la variable objetivo y hacer un modelo de clasificación que predijese en qué franja se situará la publicación. El problema es que al haber tanta dispersión de datos en la variable objetivo, las franjas serían demasiado

grandes es decir que agruparía publicaciones con una diferencia de likes de 150 mil. La opción de crear más grupos no es viable ya que serían demasiado.

Es por eso por lo que la opción más viable es hacer un modelo de regresión. Hay que decidir qué modelo es el que más se adecua a mis datos.

Variables principales del modelo

Al ser un modelo de regresión las variables de entrada deben ser numéricas o categóricas que no contengan texto.

Para la selección de variable he empezado por las variables que en teoría pueden afectar el éxito de una publicación como por ejemplo las estadísticas del autor. Es decir, el número de likes totales, número de visitas totales, número de vídeos subidos y el número de personas a las que sigue el autor. Pero estas variables evidentemente no son suficientes. No solo porque son pocas y esto hará que la precisión sea baja y el error alto, sino que como no hemos añadido ninguna característica del video, el número de likes predicho por el modelo siempre será el mismo para un mismo usuario. Es por eso que es importante añadir variables que diferencien los videos como por ejemplo la duración del video, la calidad de la imagen o el formato del video.

Aún así, aunque incluyamos estas características del video, no sabemos realmente lo que pasa en el video. No sabemos si trata de la guerra o de la familia. En un primer lugar utilice el transcriptor de videos a texto para extraer las palabras más importante del contenido del video pero el problema es que al ser de tipo de texto, no era fácil incluirlo en el modelo ya que habría que jugar con los pesos, asociaciones y extracción de características mediante procesamiento del lenguaje natural. Realmente esto último es importante tenerlo en cuenta ya que aunque no vaya a estar incluido en el modelo, se utilizará cuando creemos el asesor. Esta solución no era óptima pero saber el contenido del video o por lo menos las características del video era algo esencial. Se me ocurrió ir al inicio del proceso, es decir, la parte de la utilización de la API para extraer los datos. Estos datos eran extraídos a partir de querys que hacía TikTokApi a la aplicación de TikTok añadiendo como parámetro de búsqueda unos hashtags. Aunque no sepamos el contenido exacto del video, podemos extraer la temática general ya que los hashtag resumen en una palabra el contenido general del video. Por ejemplo, si el un video trata de algo relacionado con el deporte o incluso yendo más allá, sobre el crossfit, el autor suele poner al final de la descripción #deporte #crossfit. Entonces aprovechando que los datos son extraídos a partir de hashtag, he incluído los hashtag de búsqueda como variables en mi modelo. De esta forma tendré una listas de 10 a 15 variables que serán hashtag se rellenarán con 1 ’s y 0 ’s dependiendo de si el hashtag está en la descripción o no.

Obteniendo nuevas variables categóricas con este aspecto:

Tabla 6: Dataframe considerando los hashtag como variables

desc	humor	viral	trending	madre	padre	novia	novio	comida	deporte	sport	familia	family	amigos	friends	work	gym	gimnasio	music
Llevo ya años yendo al gym y aún no veo result...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
#ex #padre #hijo	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
🙋GIRL🙋// 🙋BOY🙋 | Snap : girlvsboy9 ...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Nunca he visto a nadie tan emocinado como esta...	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Nunca he visto a nadie tan emocinado como esta...	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Para empezar a probar los modelos nos quedamos con las siguiente variables:

- video_height
- video_width
- video_duration
- music_duration
- authorStats_followingCount
- authorStats_followerCount
- authorStats_heartCount
- authorStats_videoCount
- authorStats_diggCount
- authorStats_heart
- Lista de hashtags:
    - humor
    - viral
    - trending
    - comida
    - deporte
    - …
    - sport
    - familia
    - friends
    - work
    - music
- Tiempo_Transcurrido
- Verificado

Primeros modelos

Lo primero que he hecho ha sido usar la técnica de prueba y error para ver qué modelo tiene menor error y mayor coeficiente de determinación sin concretar ningún parámetro ni hiper parámetro.

En Python existe una librería que se llama Sklearn que contiene módulos de modelos de regresión, modelos de clasificación, métricas y validaciones. Así que lo primero que hay que hacer es instalar esta librería e importar los mejores modelos de regresión.

De esta forma he importado los siguiente modelos:

Tabla 7: Lista de modelos de regresión

Librería + Módulo	Modelo de regresión
sklearn.ensemble	GradientBoostingRegressor
sklearn.ensemble	RandomForestRegressor
sklearn.ensemble	BaggingRegressor
sklearn.ensemble	ExtraTreesRegressor
sklearn.ensemble	HistGradientBoostingRegressor
sklearn.linear_model	LinearRegression
sklearn.tree	DecisionTreeRegressor
sklearn.neighbors	KNeighborsRegressor
xgboost	XGBRegressor

Todos estos modelos los voy a testar y me quedaré con los 2 que mejores resultados me hayan dado. Una vez haya hecho esta primera selección de 2 modelos, indagaré en los parámetros de los mismos para optimizarlos.

Para validar los modelos voy a utilizar el error cuadrático medio (RMSE), el coeficiente de determinación (R^2) y el error absoluto media. Todas estas métricas están en la librería: sklearn.metrics.

A la hora de separar/dividir (hacer split) los datos en test y entrenamiento usaré la proporción 75% entrenamiento y 25% test. Para hacer está división de datos he utilizado la librería sklearn.model_selection.

Importante recordar que en estos modelos no he incluído ningún parámetro concreto, sino que se autorellena con valores por defecto o por cálculos que hace el propio modelo como la profundidad en un random forest.

Tabla 8: Tabla con los resultados obtenidos después de entrenar los modelos

Modelo utlizado	Resultados obtenidos
GradientBoostingRegressor	R^2= **0.8050611169275138** El error (rmse) de test es: **62814.9** El error absoluto media es de: **44296.3**
RandomForestRegressor	R^2= **0.8192454592105335** El error (rmse) de test es: **63745** El error absoluto media es de: **42544.4**
BaggingRegressor	R^2= **0.7745398502836062** El error (rmse) de test es: **67553.6** El error absoluto media es de: **44942.2**
ExtraTreesRegressor	R^2= **0.7733530631839766** El error (rmse) de test es: **67731.1** El error absoluto media es de: **44383.9**
HistGradientBoostingRegressor	R^2= **0.809477722353872** El error (rmse) de test es: **62099.2** El error absoluto media es de: **42501.8**
LinearRegression	R^2= **0.7554810625162303** El error (rmse) de test es: **70350.9** El error absoluto media es de: **46959.6**
DecisionTreeRegressor	R^2= **0.6037538220970826** El error (rmse) de test es: **89556.3** El error absoluto media es de: **56080.7**
KNeighborsRegressor	R^2= **0.21041298113786777** El error (rmse) de test es: **126419.3** El error absoluto media es de: **89135.4**
XGBRegressor	R^2= **0.7900941357630107** El error (rmse) de test es: **65181.7** El error absoluto media es de: **44988.7**

Quitando el árbol de decisión, k-vecinos y la regresión lineal, en el resto de modelos el coeficiente de determinación supera el 77%. El error cuadrático medio y el error absoluto

medio ronda los 50 mil likes de diferencia. La media de likes de las publicaciones es de 1.614.232 likes así que proporcionalmente el error no es muy significativo.

Viendo el coeficiente de determinación de los modelos, e error cuadrático medio y el error absoluto medio los dos mejores modelos son:

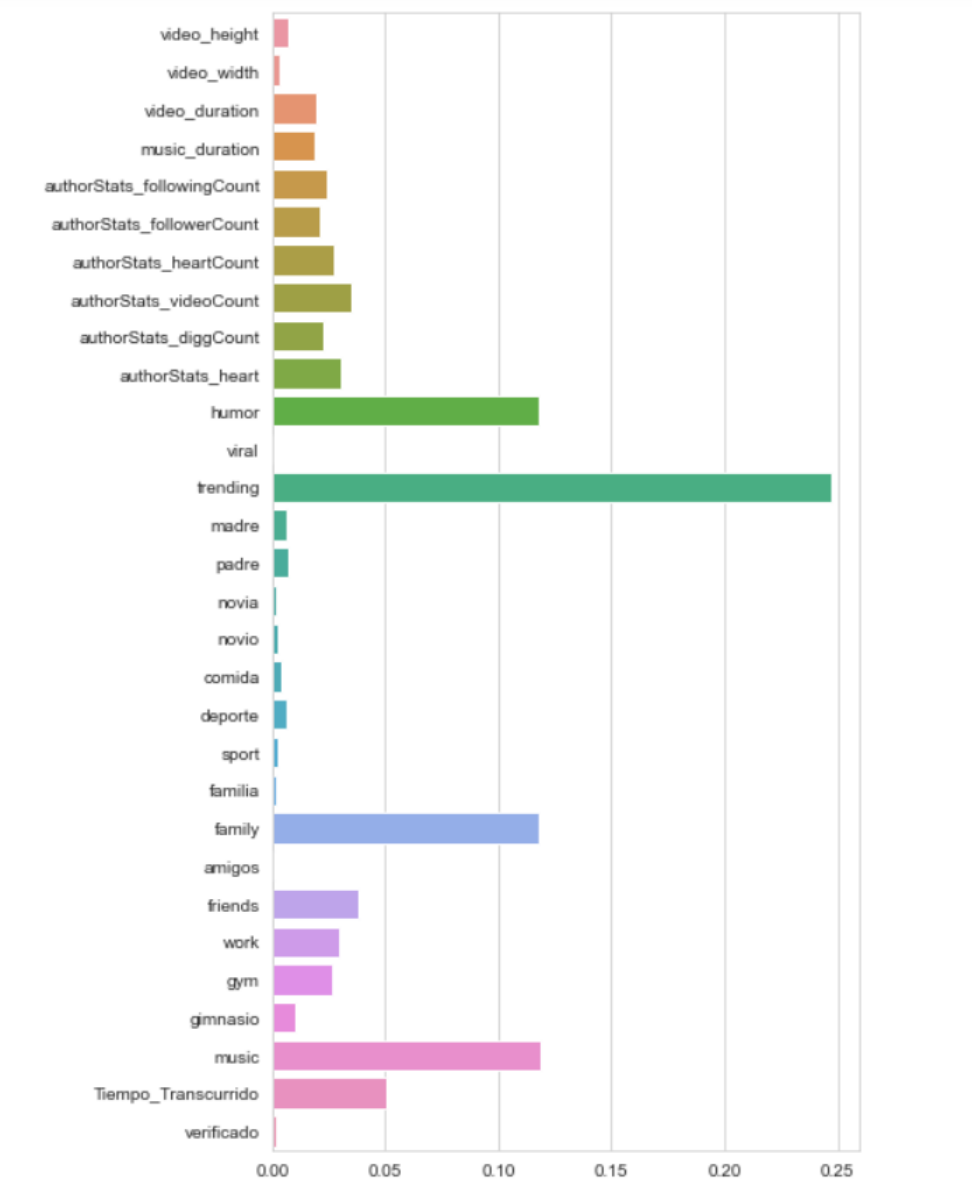
- RandomForestRegressor
- HistGradientBoostingRegressor

He descartado el modelo GradientBoostingRegressor ya que la principal diferencia con el modelo HistGradientBoostingRegressor es que es más lento.

Modelo Random Forest - Predicción del número de likes:

Para saber la importancia de cada variable en los modelos, hay una función que se llama feature_importances_ que asocia pesos a las variables según la relevancia en el modelo.

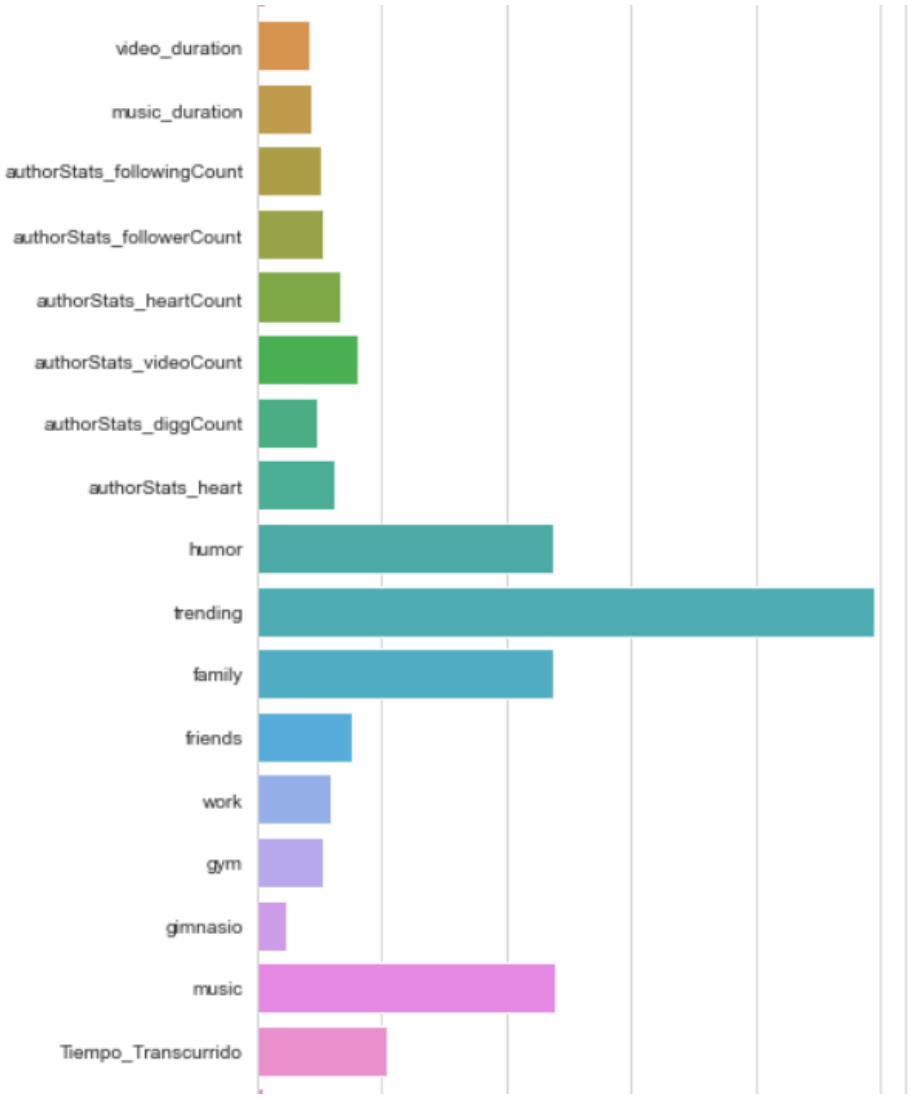
Figura 16: Importancia de las variables del Random Forest



Se puede diferenciar claramente las variables que más han influido en el modelo y aquellas que no han aportado nada. Grosso Modo, la mayoría de las estadísticas del usuario son relevantes. Esto tiene sentido ya que un usuario con muchos seguidores tendrá más likes en su próxima publicación que uno con 100 seguidores. También se observa con los hashtag #trending #familia #music son los hashtag más relevantes a la hora de predecir el éxito de la publicación. Este tipo de observaciones nos ayudarán en un futuro para aconsejar la utilización de

determinados hashtags. Para quitar ruido al modelo es importante quitar las variables que no tengan peso quedándonos finalmente con las siguientes variables:

Figura 17: Variables más relevante para el random forest



Optimización de hiperparámetros

Al tratarse de hiperparámetros, no se puede saber de antemano cuál es el valor más adecuado. La forma de identificarlos es mediante el uso de estrategias de validación, por ejemplo validación cruzada.

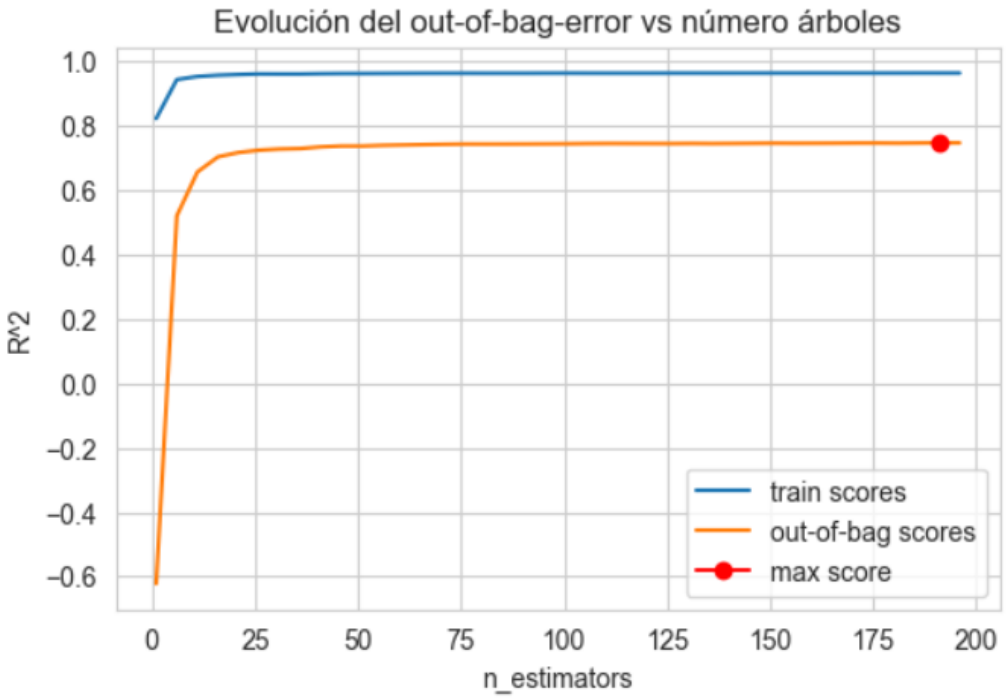
La validación cruzada es computacionalmente costosa pero los modelos Random Forest tienen la ventaja de disponer del Out-of-Bag error, lo que permite obtener una estimación del error de test sin recurrir a la validación cruzada.

Determinación del número de árboles

El número de árboles en un random forest no es un hiperparámetro crítico ya que aumentar el número de árboles solo va a hacer que el coeficiente de determinación aumente. Pero a partir de un cierto número de árboles, la mejora se estabiliza. Seguir aumentando el número de árboles solo se va a ver reflejado en la carga computacional.

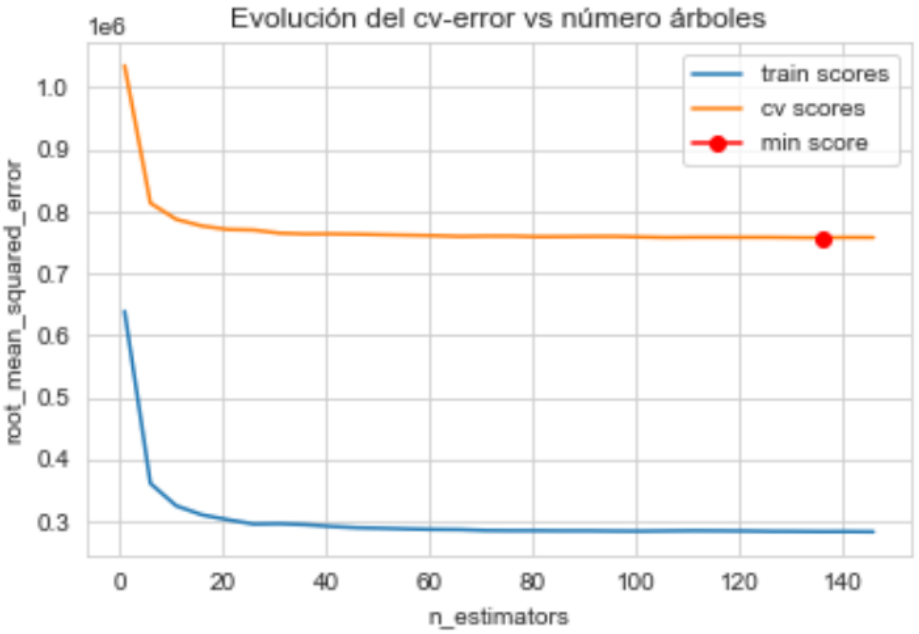
El funcionamiento es simple. La idea es probar el modelo con 1 árbol hasta 150 y ver cuál es el número de árboles óptimo.

Figura 18: Evolución del out-of-bag-error vs número de árboles



Usando out-of-bag-error, llegamos a la conclusión de que el número óptimo de árboles es 191. Aunque este proceso es menos costoso computacionalmente hablando, vamos a realizar el mismo procedimiento pero usando k-cross-validation (validación cruzada) y neg_root_mean_squared_error. Es decir que gracias a un bucle iremos probando distintas cantidades de árboles pero ahora usaremos validaciones cruzadas para evaluar los resultados.

Figura 19: Evolución del cv-error (validación cruzada) vs número de árboles



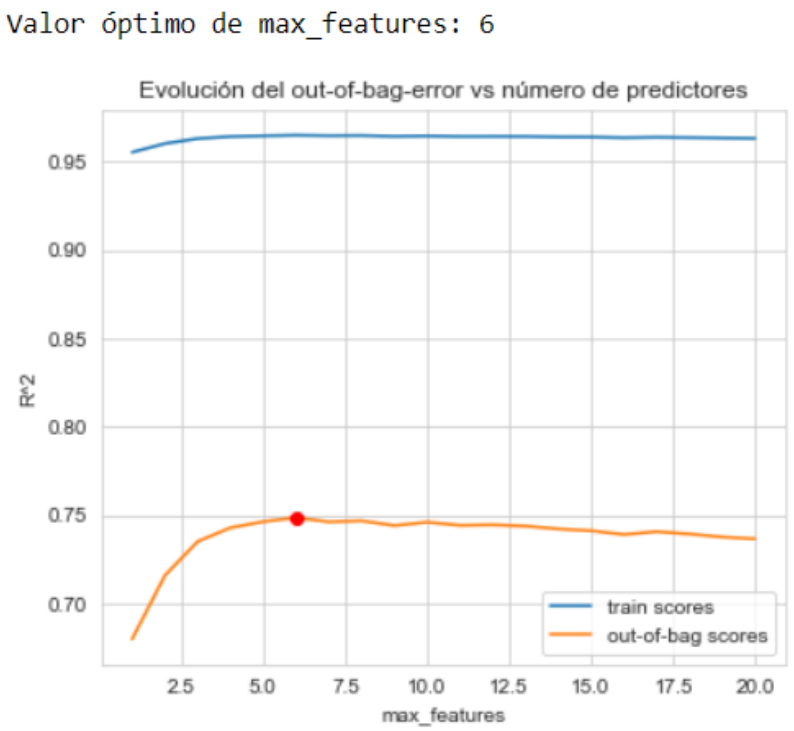
Usando validación cruzada, obtenemos que el número de árboles óptimo es 136. Entonces sabemos que el número óptimo de árboles está entre 136 y 196. Aunque los modelos dicen que el número óptimo está entre 136 y 196, podemos ver gráficamente que a partir de los 40 árboles se estabiliza. Al tener no tener demasiados datos, no es un problema computar con 150 árboles pero en el caso de aumentar la cantidad de datos, la opción más viable es reducir el número de árboles a 40 ya que luego la variación es mínima

Determinación del hiperparámetro 'Max_feature'

El valor del max_features es uno de los hiperparámetros más importantes de random forest, ya que es el que permite controlar cuánto se decorrelacionan los árboles entre sí.

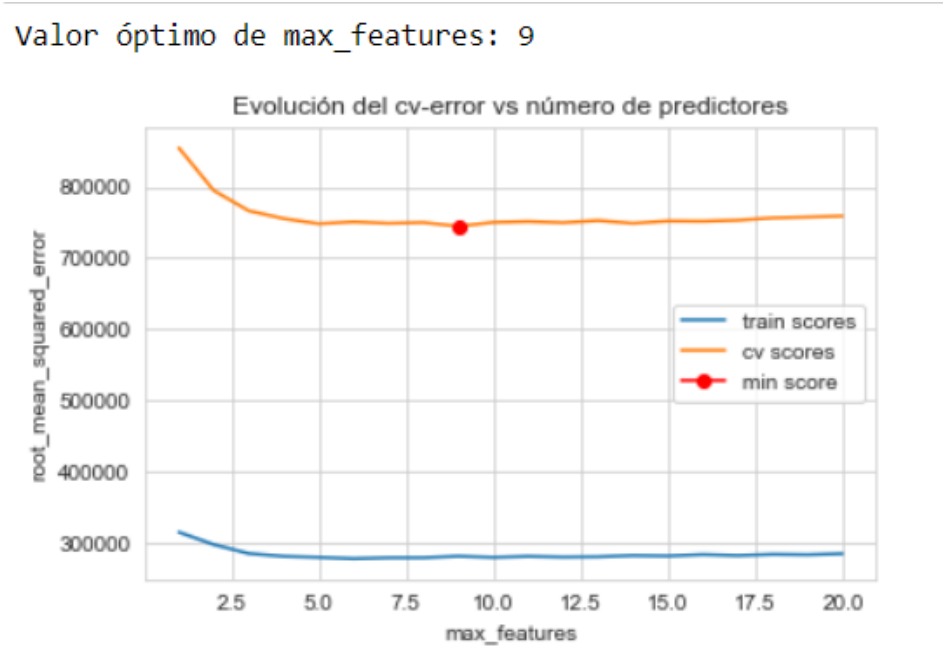
Para determinar el valor óptimo del Max_feature vamos a repetir el mismo procedimiento que con el número de árboles. Primero usando el Out-of-Bag error y luego con validación cruzada.

Figura 20: Gráfico para determinar el valor del Max_feature óptimo usando Out-of-Bag-error



El valor óoptimo de max_fartures es 6 usando Out-of-Bag-Error. Al igual que antes, vamos a repetir esto mismo pero con validación cruzada:

Figura 21: Gráfico para determinar el valor del Max_feature óptimo usando Cross Validation



Acorde a las dos métricas utilizadas, el valor óptimo de max_features está entre 6 y 9.

Hallar los hiperparámetros óptimos no debe hacerse de forma secuencial, ya que cada hiperparámetro interacciona con los demás. Para determinar la combinación de max_feature y número de árboles más óptima es oportuno utilizar un grid search. Básicamente, ejecuta todas las combinaciones posibles entre max_feature, número de árboles y la profundidad. Pero sabemos que:

- El valor óptimo de max_features está entre 6 y 9.
- El número óptimo de árboles está entre 136 y 210

Sabiendo esto, vamos a limitar las combinaciones ya que para cada combinación se va a entrenar y testar un modelo por lo que no acotar las combinaciones puede suponer un gran coste computacional.

Se ejecutan los modelos haciendo todas las combinaciones de los hiperparámetros:

```
Modelo: {'max_depth': None, 'max_features': 6, 'n_estimators': 136} ✓
Modelo: {'max_depth': None, 'max_features': 6, 'n_estimators': 150} ✓
Modelo: {'max_depth': None, 'max_features': 6, 'n_estimators': 180} ✓
Modelo: {'max_depth': None, 'max_features': 6, 'n_estimators': 190} ✓
Modelo: {'max_depth': None, 'max_features': 6, 'n_estimators': 200} ✓
Modelo: {'max_depth': None, 'max_features': 7, 'n_estimators': 136} ✓
Modelo: {'max_depth': None, 'max_features': 7, 'n_estimators': 150} ✓
Modelo: {'max_depth': None, 'max_features': 7, 'n_estimators': 180} ✓
```

Una vez se han ejecutado y testado todas las combinaciones, se crea una lista ordenada de las mejores combinaciones:

Tabla 9: Mejores hyperparámetros

max_depth	max_features	n_estimators
18.0	9.0	200.0
18.0	9.0	190.0
18.0	9.0	180.0
20.0	7.0	180.0

Este modelo se va a entrenar cada día ya que es fundamental estar actualizado ya que los temas de ayer ya no son tan relevantes como los temas de hoy. De esta forma la base de datos irá aumentando y los hiperparámetros podrán modificarse.

Creación del asesor

Es importante comprender que todos los creadores de contenidos tienen un elemento diferenciador dentro del tipo de contenido que publican. En otras palabras todos los influencers tienen un sello personal que les caracteriza. Pero es igual de cierto que la mayoría de los creadores siguen las modas para no estancarse haciendo por ejemplo retos virales, hablando de temas relevantes en la actualidad o bailando las canciones más trending del momento. Es por eso que un asesor no puede aconsejar al influencer sobre cómo hacer su trabajo pero sí que puede ofrecerle un esquema visual de la actualidad en redes sociales. Hay que tener en cuenta que en todas las redes sociales se hace viral el mismo tipo de contenido ya que el público pertenece a nichos de mercado muy diferentes. Como he mencionado en la introducción, la media de edad de los usuarios de TikTok ronda los 16 años cuando en Twitter por ejemplo es mucho más elevada. Esta diferencia de edad hace que el tipo de contenido viral también pueda ser diferente. El humor, los gustos y las preferencias van cambiando a medida que crecemos y maduramos. No obstante, esto no es blanco o negro. Cuando en twitter se hizo trending topic en twitter una polémica sobre un streamer de twitch (plataforma para hacer directos), los videos que se publicaron en TikTok acerca de este tema también se volvieron virales. Es decir que hay temas que son virales sin importar la red social en la que se publique.

Es por esto que a la hora de crear un esquema visual que refleje lo viral, no hay que tener en cuenta únicamente la red social en la que publicamos. En este caso tener en cuenta exclusivamente el contenido que se sube a TikTok hace que se cierren puertas además de impedir innovar. La repercusión de un video tiene que ver con los seguidores del que lo publica, con lo bueno que sea el video y con los origian/innovador que sea en la plataforma dónde se publica. Si solo nos basamos en los videos de TikTok, por muy original que sea el video sigue siendo una copia de otros videos que tratan sobre lo mismo lo que hace que los usuario se cansen y que se estanque. De ahí, la importancia de tener distintos puntos de vistas, distintas referencias acerca de qué es de lo que gente habla. Es por eso, que para poder dar una visión más global voy a ofrecer al usuario, un esquema de lo que es viral en TikTok y también en Twitter.

Obtención de datos con la API oficial de Twitter

Al igual que para extraer datos de TikTok he utilizado una api con una serie de funciones de scrapping, para Twitter voy a seguir el mismo procedimiento. En este caso es una librería de python que se llama Tweepy que hace consultas a la API REST de twitter. Al hacer consultas a una API oficial[13] es necesario la creación de un perfil creando una serie de API keys que funcionan como identificadores para poder utilizar los productos de la API.

Lo primero que se debe hacer es registrarse. Para registrarse hay que dar una serie de datos personales como el mail, el número de teléfono o tu dirección. La diferencia con el resto de API

's es que tienes que asociar una cuenta de Twitter. Esto se debe a que las consultas que vaya a efectuar en un futuro se harán a partir de esta cuenta. Las funciones que voy a utilizar de esta API se basa exclusivamente en scrapear tweets pero también se puede controlar tu cuenta a través de la API como por ejemplo programar tweets para que se publiquen en diferentes momentos del día, tener un control de tus seguidores o ver qué cuentas te siguen pero tú a ellos no. Pero realmente quiero utilizar la API para crear un buscador de tweets siguiendo una serie de parámetros.

Después de registrarte, es necesario crear un proyecto para conseguir todas las credenciales y permisos para utilizar la API. A la hora de crear un proyecto hacen una serie de preguntas previas para consentir el uso como:

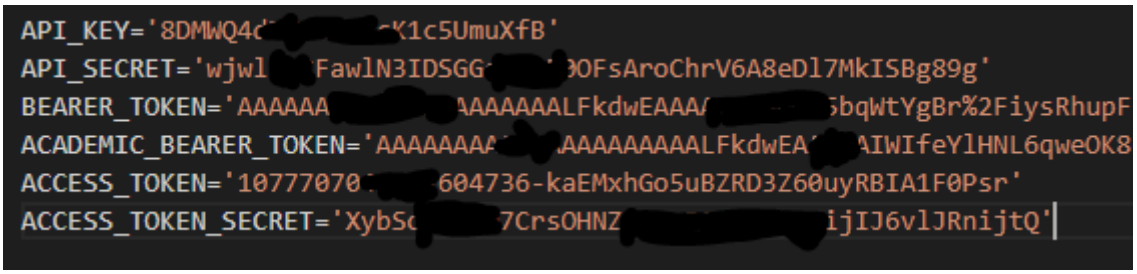
- ¿Cómo vas a utilizar la API?
- ¿Para qué quieres los datos?
- ¿Cómo los vas a analizar?

Una vez has creado el proyecto, hay 3 tipos de accesos:

- Essencial
- Elevated
- Academic

La diferencia entre estos 3 accesos se basa en el número de consultas que se pueden hacer de manera mensual. El acceso Essential nos permite hacer hasta 1 millón de consultas que son muchas más de las necesarias y además es gratuito. Cuando ya tienes asociado un producto al proyecto, te dan una serie de claves para poder acceder a la API:

Figura 22: Definición de credenciales



Ahora solo es necesario importar TWEEPY en el entorno virtual y seguir la documentación de la API para crear las consultas.

Lo primero es crear el acceso desde el entorno virtual creando un cliente con las credenciales de la imagen anterior. Una vez tenemos el cliente creado, empezamos con las consultas. En mi caso solo voy a necesitar dos funciones principalmente:

- search_recent_tweets
- available_trends

La primera función busca los tweets más recientes y con más repercusión que contengan la(s) palabra(s) que se metan como parámetro. Esto lo voy a utilizar cuando el usuario quiere referencias acerca de un hashtag en concreto.

La segunda función extrae los tweets que estén en trending topic, es decir los tweets con más repercusión.

La API ofrece todo tipo de información acerca de cada tweet. Desde el nombre de usuario hasta la fecha de publicación. En nuestro caso los datos que más información considero que aportan son los siguientes:

- Nombre de usuario
- texto/foto/video del tweet en cuestión
- Estadísticas generales del tweet como el número de likes, de retweets o de veces compartido
- Estadísticas generales del usuario como número de seguidores
- Fecha de creación

Usando la función search_recent_tweets y poniendo como palabra de búsqueda “covid-19” se obtiene un dataframe como este:

Tabla 10: Dataframe resultante de la consulta en twitter

username	text	created_at	metrics_tweet_retweet	metrics_user_follower	metrics_us
Jamie93455967	RT @ryck_nancy: @MarkGerretsen Stop lying . S	23/06/2022 16:44	1	478	
RosemaryQuinlin	RT @ABC: Evidence suggests that vaccinated, lo	23/06/2022 16:44	9	1300	
LimitGovtPower	RT @kacrn91: https://t.co/OiU92kAGtG	23/06/2022 16:44	68	480	
midymarshmallow	RT @pskwrk: à¸-à¸µà¸”à¸-à¸ à¸„à¸™à¸«à¸™à¸‰à¸	23/06/2022 16:44	2227	1	
christinermertz	RT @CNN: Mayim Bialik has tested positive for (	23/06/2022 16:44	60	401	
AndreaSantG	RT @ramos_ampudia: @mafe_orquera @Faustc	23/06/2022 16:44	7	307	
JosDom16	RT @lalibrebe: L'Autriche va mettre fin Ã la vac	23/06/2022 16:44	394	256	
EDERSON37561344	RT @claremibueloni: Na Alemanha protesto po	23/06/2022 16:44	9	14198	
MAHeffernan1	RT @RobertKennedyJr: An active-duty senior A	23/06/2022 16:44	139	2369	
wendyvmartinezv	RT @ESESANRAFAEL: #AvisoImportanteðŸ"‰El	23/06/2022 16:44	1	3	
fleur4776	RT @lalibrebe: L'Autriche va mettre fin Ã la vac	23/06/2022 16:44	394	118	
siamonoitv2000	RT @TV2000it: #Covid #Omicron5 | Crescono i c	23/06/2022 16:44	1	6075	
roscuberm811	RT @JCarlos_Valerio: #ÃšltimaHora | Ante el in	23/06/2022 16:44	41	86	

Creación de la aplicación web

Para crear la aplicación web he optado por utilizar streamlit. Es una biblioteca de python que permite crear todo tipo de aplicaciones de datos desarrolladas en Python. Esta ha sido la opción más viable por su sencillez y porque todos los modelos y las API 's se han creado y utilizado en Python. Streamlit contiene una serie de funciones para crear dashboard de forma sencilla y visualmente agradable. La instalación de esta biblioteca es igual de fácil y sencilla que cualquier otra biblioteca de python.

La aplicación se va a componer de 3 páginas principales sin contar con la página de inicio y el formulario para iniciar sesión. Como he dicho antes Streamlit, a diferencia de otras aplicaciones para crear páginas web o aplicaciones, utiliza python como lenguaje de programación esto es útil a la hora de utilizar modelos de predicción en la aplicación. Las tres principales páginas de componen de:

- Predictor de viralidad del video que introduzca el usuario
- Herramienta de consulta de videos en TikTok y tweets según una temática específica introducida por el usuario.
- Herramienta de consulta de videos en TikTok y tweets más virales en la actualidad

Creación de la página para la predicción de viralidad

Esta página se compone principalmente de un formulario dónde el usuario introduce la descripción del video, el propio video en formato MP4 y el plazo deseado de previsión de impacto en días. Una vez el usuario envía el formulario, el resto de datos que deben ser introducidos en el modelo, como las estadísticas de usuario, son capturados por la API. Cuando el usuario se registra en la aplicación, tiene que introducir su cuenta de TikTok de esta forma al tener el nombre de usuario en TikTok, se puede extraer el resto de datos a través de la API.

Una vez el usuario ha introducido el video y ha escrito la descripción se recrea como quedaría en la plataforma de TikTok y se le ofrece al usuario la opción de continuar o cambiar. Si el usuario acepta:

- Se capturan los datos del usuario haciendo una consulta a la API utilizando el nombre de usuario en TikTok el cual tuvo que haber introducido en la página para iniciar sesión. Extrayendo los siguientes datos:
    - Número de seguidores
    - Número de cuentas a las que sigue
    - Número de videos publicados
    - Número de likes totales

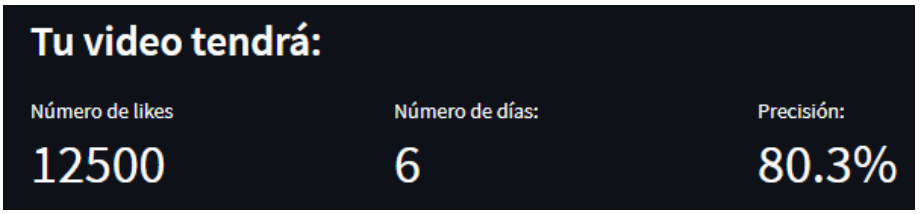
- Se analiza el video y la descripción del video extrayendo:
    - Duración del video y del audio (por lo general es el mismo)
    - Hashtag utilizados (usando procesamiento del lenguaje natural se extrae la palabra que esté a continuación de un #)

Como he mostrado anteriormente, los hashtag se meten como variables independientes. Es decir que cada hashtag es una variable. A la hora de entrenar el modelo, introduje un listado de hashtag y comprobé cuales eran los hashtags más relevantes. Es por eso que al usuario se le tiene que mostrar esta lista de hashtag que en términos generales engloba todas las temáticas

posibles o por lo menos las que más repercusión tienen. Se introducirán como 1 en los hashtag que haya seleccionado/escrito en la descripción.

Estos datos se introducen en el modelo como inputs. Usando la función predict() obtenemos el número de likes que tendrá la publicaciones en el número de días que el usuario haya deseado. Este resultado se le muestra al usuario como una serie de métricas:

Figura 23: Resultado de la predicción de repercusión



Después de imprimir por pantalla los resultados del modelo, se ofrece al usuario la opción de cambiar la descripción y el video para poder saber cual es la mejor opción.

Creación de la página de asesoramiento siguiendo una temática

El objetivo de este frame es mostrar videos de tiktoks y Tweets que tengan relación con el tema que quiera el usuario. Gracias a la api de TikTok se pueden extraer videos que contengan un hashtag en específico creando una dataframe con todos sus datos. Dentro de estos datos, nos encontramos con el link de descargar del video. De esta forma, podemos descargar el video y enseñarselo por pantalla al usuario. En resumidas cuentas, el usuario va a poder ver videos que contengan el hashtag en cuestión junto con sus métricas más relevantes como el número de likes, visitas y comentarios. Para extraer estos videos es necesario utilizar la función by_hashtag de la API de TikTok introduciendo como parámetros la temática que el usuario quiera.

Para complementar esta captura de datos, también se va a usar la API de twitter para extraer los tweets que tengan relación con la misma temática. Usando la función 'search_recent_tweets' e introduciendo la temática, extraemos los tweets con las variables previamente comentadas.

Este frame se va a componer de una serie de filtros que el usuario va a poder modificar como el número de seguidores mínimo, el número de retweets mínimo y demás. Estos filtros son modificables desde la barra lateral de la aplicación. Ver en los resultados del proyecto.

Una vez están fijados los filtros, el usuario debe añadir el tema. Cuando el usuario pulse el enter, se ejecutarán las consultas y se imprimirán por pantalla los resultados. Lo primero que aparecerá serán los videos de tiktok y luego los tweets pero todo en un mismo frame para crear un esquema visual general. Ver en los resultados del proyecto.

La tabla con el conjunto de tweets es desplegable para tener una mejor visualización. Al clicar en el nombre de usuario se muestran una serie de estadísticas como el número de seguidores, el número de personas que sigue este usuario o el número de tweets publicados. Si el usuario clica en el texto del tweet aparecen las estadísticas del tweet en cuestión como el número de retweets, de respuestas o de likes. Ver en los resultados del proyecto.

Creación de la página de tendencias en la actualidad

En esta página se busca crear una imagen de las tendencias en redes sociales particularmente en TikTok (siendo esta la plataforma principal del usuario de la aplicación) y en twitter. Es importante que tanto los videos en tendencias de TikTok como los tweets que estén en trending topic estén en la misma página porque lo importante es contrastar la viralidad en las dos redes sociales.

Para hallar los videos de las tendencias en TikTok se usa la función “by_trending” creándose un dataframe con las mismas variables que genera la función “by_hashtag” previamente mencionada. Una de las variables es el link de descargar del video el cual usaremos para mostrar los videos de manera visual y no como una tabla. En el caso de twitter para extraer los tweets que estén en el trending topic se utiliza la función “available_trends”.

La estructura de esta página es muy similar a la página anterior. En este caso el número mínimo de likes y de retweets no se contemplan como filtros ya que se trata de los contenidos más virales de ambas plataformas. Sin embargo, puede ser relevante el país o el idioma de los contenidos. Es por ello, que los filtros de está página serán el país del que provienen los videos y el idioma.

Una vez el formulario con los filtros se rellena, se imprimen por pantalla los resultados de las tendencias en el país que haya deseado e idioma que el usuario haya seleccionado.

El funcionamiento de la exposición del trending topic de twitter según el país y el idioma es el mismo que en la página anterior. Una tabla con el nombre de usuario y el texto del tweet y si seleccionas un usuario aparecen las estadísticas principales del usuario y si clicas en el texto aparecen las estadísticas del tweet en cuestión.

4.3 Recursos requeridos

Los recursos requeridos para este proyecto son únicamente computacionales. Estos se dividen en componentes hardware y software:

Ordenador ASUS (Windows). Desde aquí se crea un entorno virtual y se instalan una serie de librerías y software para el correcto funcionamiento de la aplicación. Usando el comando pip install se instalan los siguientes paquetes. Parecen mucho pero realmente cuando instalas la API de TikTok, se descargan 15 paquetes más.

Paquetes necesarios junto a la **versión** utilizada:

- altair 4.2.0
- argon2-cffi 21.3.0
- argon2-cffi-bindings 21.2.0
- asttokens 2.0.5
- async-generator 1.10
- attrs 21.4.0
- autopep8 1.6.0
- backcall 0.2.0
- beautifulsoup4 4.11.1
- bleach 5.0.0
- blinker 1.4
- cachetools 5.2.0
- certifi 2021.10.8
- cffi 1.15.0
- charset-normalizer 2.0.12
- click 8.1.3
- colorama 0.4.4
- commonmark 0.9.1
- cryptography 36.0.2
- debugpy 1.6.0
- decorator 5.1.1
- defusedxml 0.7.1
- entrypoints 0.4
- executing 0.8.3
- fastjsonschema 2.15.3
- gitdb 4.0.9
- GitPython 3.1.27
- greenlet 1.1.2
- h11 0.13.0
- helpers 0.2.0
- idna 3.3
- importlib-metadata 4.11.4
- ipykernel 6.13.0
- ipython 8.2.0
- ipython-genutils 0.2.0
- ipywidgets 7.7.0
- jedi 0.18.1
- Jinja2 3.1.1
- jsonschema 4.4.0
- jupyter 1.0.0
- jupyter-client 7.2.2

- jupyter-console 6.4.3
- jupyter-core 4.10.0
- jupyterlab-pygments 0.2.2
- jupyterlab-widgets 1.1.0
- MarkupSafe 2.1.1
- matplotlib-inline 0.1.3
- mistune 0.8.4
- nbclient 0.6.0
- nbconvert 6.5.0
- nbformat 5.3.0
- nest-asyncio 1.5.5
- notebook 6.4.10
- numpy 1.22.3
- oauthlib 3.2.0
- outcome 1.1.0
- packaging 21.3
- pandas 1.4.2
- pandocfilters 1.5.0
- parso 0.8.3
- pickleshare 0.7.5
- Pillow 9.1.1
- pip 22.0.4
- playwright 1.17.2
- plotly 5.8.2
- prometheus-client 0.14.1
- prompt-toolkit 3.0.29
- protobuf 3.20.1
- psutil 5.9.0
- pure-eval 0.2.2
- pyarrow 8.0.0
- pycodestyle 2.8.0
- pycparser 2.21
- pydeck 0.7.1
- pyee 8.1.0
- Pygments 2.11.2
- Pympler 1.0.1
- pyOpenSSL 22.0.0
- pyparsing 3.0.8
- pyrsistent 0.18.1
- PySocks 1.7.1
- python-dateutil 2.8.2
- pytz 2022.1
- pytz-deprecation-shim 0.1.0.post0

- pywin32 303
- pywinpty 2.0.5
- pyzmq 22.3.0
- qtconsole 5.3.0
- QtPy 2.0.1
- requests 2.27.1
- requests-oauthlib 1.3.1
- rich 12.4.4
- scipy 1.8.1
- selenium 4.1.3
- selenium-stealth 1.0.6
- semver 2.13.0
- Send2Trash 1.8.0
- setuptools 58.1.0
- six 1.16.0
- smmap 5.0.0
- sniffio 1.2.0
- sortedcontainers 2.4.0
- soupsieve 2.3.2.post1
- stack-data 0.2.0
- streamlit 1.10.0
- tenacity 8.0.1
- terminado 0.13.3
- TikTokApi 4.1.0
- tinycss2 1.1.1
- toml 0.10.2
- toolz 0.11.2
- tornado 6.1
- traitlets 5.1.1
- trio 0.20.0
- trio-websocket 0.9.2
- tweepy 4.10.0
- typing_extensions 4.2.0
- tzdata 2022.1
- tzlocal 4.2
- urllib3 1.26.9
- validators 0.20.0
- watchdog 2.1.9
- wcwidth 0.2.5
- webencodings 0.5.1
- websockets 10.1
- widgetsnbextension 3.6.0
- wsproto 1.1.0

- zipp 3.8.0

Tanto los modelos, las consultas y la propia aplicación se ejecutan en python pero el conjunto de archivos python es creado desde Visual Studio Code. De esta manera es más fácil gestionar los archivos y la totalidad del entorno virtual ya que cada programa en python no trabaja de manera independiente al resto sino que hay interacciones entre archivos. Esto usar visual studio code facilita la ejecución de los programas en el entorno virtual creado y la ejecución de la aplicación con Streamlit.

4.4 Presupuesto

Tabla 11: Tabla de costes y presupuesto

Tipo de coste	Valor	Comentarios
Horas de trabajo en el proyecto	290 horas	Duración aproximada del proyecto de 4 meses
Recursos materiales (Ordenador)	1500€	Recurso ya adquirido. Precio de mercado actual
Software utilizado		
Anaconda	0€	Gratis
Visual Studio Code	0€	Gratis
Tik TokApi	0€	Prueba gratuita
Twitter API	0€	Prueba gratuita
TikApi	30€	Coste mínimo de la API oficial de TikTok usada para experimentación y documentación
Otros		
Internet	88€	Precio de 22€/mes por 10GB durante 4 meses
TOTAL	1618€	

4.5 Viabilidad

El estado actual del proyecto consiste en un prototipo del producto que se pondría en producción en un futuro. Aún no se puede considerar un producto monetizable, pero en el apartado de Anexos se ha incluido un business model canvas con una propuesta inicial del posible modelo de negocio.

4.6 Resultados del proyecto

Modelo de predicción

Tras realizar todas las combinaciones posibles entre modelos, variables e hiperparámetro, el modelo de predicción que mejores resultados da es el Random Forest.

Las variables del modelos son:

- La duración del video
- La duración del audio
- Número de personas a la que sigue el usuario
- Número de seguidores
- Número de likes en total entre todos los videos del usuario
- Número de videos publicados por el usuario
- Lista de hashtags:
    - humor
    - viral
    - trending
    - comida
    - deporte
    - …
    - sport
    - familia
    - friends
    - work
    - music
- Tiempo transcurrido entre cuando se publicó el video y hoy en día

He entrenado 9 modelos de predicción (ver resultados de cada modelos aquí) para ver cual de los modelos con los hiperparámetros predeterminados era el que mejor se ajustaba. De ahí, concluyo que el mejor modelo es el Random Forest para este caso. Pero ahora es necesario optimizar los hiperparámetros.

Tras utilizar algoritmos de verificación cruzada y Out-of-Bag-Error obtenemos una horquilla de hiperparámetros. Lo que significa que obtenemos el número mínimo y máximo de max_feature, número de árboles y profundidad. Pero para hallar la mejor de las combinaciones, se utiliza un grid search (para ver en detalle: Descripción de la solución). De

esta forma, se obtiene una tabla con las mejores combinaciones de hiperparámetros para nuestro Random Forest:

Tabla 12: Mejores hiperparámetros para el modelo

max_depth	max_features	n_estimators
18.0	9.0	200.0
18.0	9.0	190.0
18.0	9.0	180.0
20.0	7.0	180.0

En conclusión, se ha creado un modelo de random forest optimizado con un R^2 de **0.819%**.

Funcionamiento de la aplicación

Realmente la idea inicial del proyecto se basaba exclusivamente en esta funcionalidad pero, al pensar en el proyecto como un futuro producto que se podría lanzar al mercado, opté por ponerme en la posición del usuario y de esta forma consideré que una aplicación con solo una funcionalidad por muy buena que pueda llegar a ser, no era suficiente. Es por eso que poniéndome en la piel de consumidor llegue a la conclusión de que era necesario que no solo pudiese predecir la repercusión que puede llegar a tener un video sino que añadiría mucho valor si, además, pudiese consultar en la misma aplicación posibles temáticas para futuros videos.

El producto final del proyecto es una aplicación web creada con streamlit. Esta aplicación consta de 3 grandes apartados. El apartado más relevante de la aplicación y del proyecto es el modelo de predicción que predice el número de likes que tendrá una publicación en x días.

Como he mencionado en el desarrollo de la aplicación es esencial que el usuario inicie sesión antes de hacer cualquier consulta ya que la predicción de likes varía enormemente según la persona que publique el video y sin estos datos el modelo no funciona. Al pensar en qué datos pedir al usuario cuando se registre, he pensado en cuales son los datos necesarios que pide el modelo, es decir los inputs. Estos se componen por las estadísticas del usuario y el resto de datos que se capturan a partir del video que quiera publicar más adelante. Por lo tanto a la hora de registrarse solo es necesario que introduzca el nombre de usuario en TikTok, el nombre de usuario que desea tener en TikTokAdvice junto a la contraseña y un correo electrónico para recibir notificaciones. Cuando el usuario rellena el formulario para registrarse, la aplicación hace una consulta a la API de TikTok para extraer las estadísticas del usuario.

Este es el formulario de inicio de sesión:

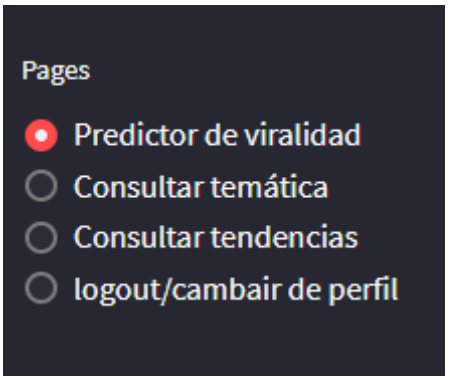
Figura 24: Frame del inicio de sesión en la aplicación

Una vez el usuario se ha registrado, tiene 3 opciones a elegir:

- Predecir la repercusión de su próximo video
- Consultar videos en TikTok y Tweets que sean relevantes en relación a una temática
- Consultar los videos más virales en el mundo o en un país en particular

El usuario puede ir cambiando de página desde la barra lateral de la aplicación de esta manera:

Figura 25: Opciones de página/funcionalidad en la aplicaciones

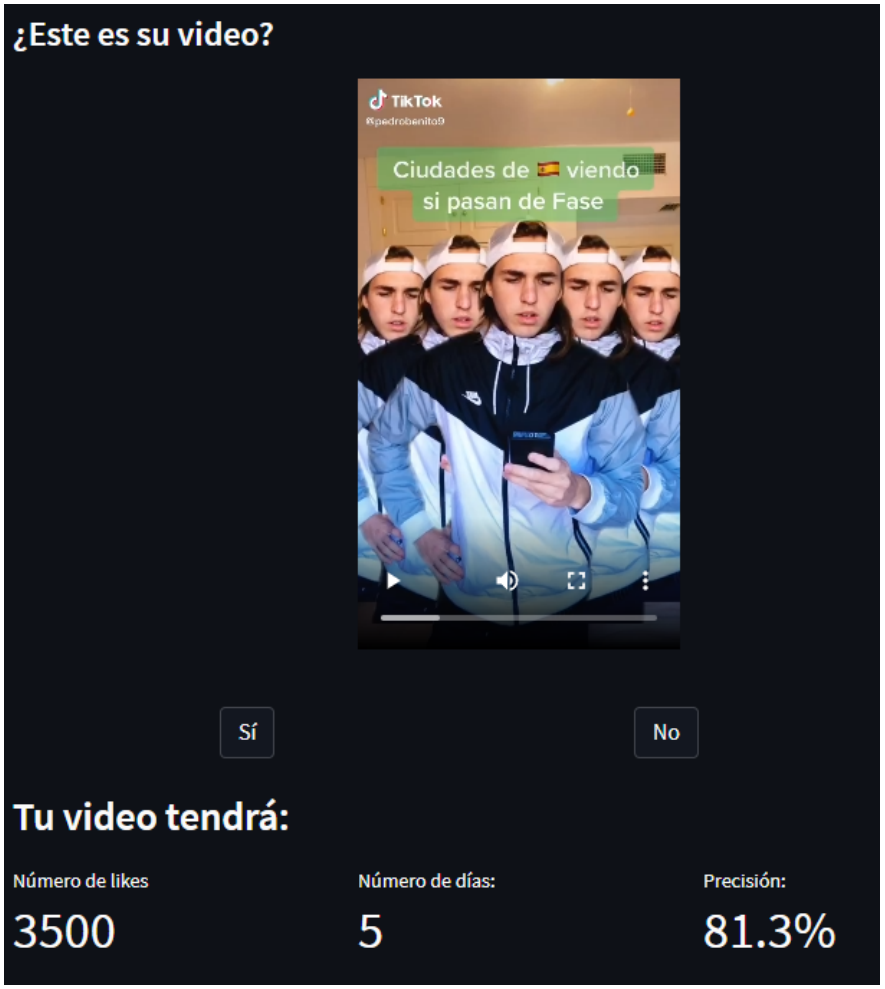


Empezamos por la página dónde el usuario introduce un video, una descripción y el plazo de días para la predicción.

Figura 26: Formulario para introducir los datos al modelo completado

Internamente, la aplicación captura los datos del usuario introducidos cuando se inicia sesión y los datos del video y llama al modelo para que prediga el número de likes. Antes de mostrar los resultados, se le pregunta al usuario si está seguro de que este es el video que le gustaría publicar. Cuando el usuario clica el botón “sí” se imprime por pantalla los resultados:

Figura 27: Resultados de la predicción



En la página de "Consultar temática" nos encontramos con una serie de filtros.

Figura 28: Filtros de la página de "Consultar temática"

Una vez fijados los filtros, el usuario debe introducir el tema como por ejemplo "pareja":

Figura 29: Formulario para introducir la temática

Ahora el usuario solo debe pulsar enter y aparecerán videos de TikTok relacionados con la temática introducida con la opción de deslizar para seguir viendo más videos relacionados. Justo después de los videos de TikTok vemos los tweets que tocan el tema que el usuario quiere consultar.

Figura 30: Resultado de la consulta (Videos TikTok)

Figura 31: Resultados de la consulta (tweets)

La última página (“Consultar tendencias”) sigue el mismo principio que la página anterior con la diferencia de que los filtros son distintos ya que lo que interesa ahora no es tanto el número mínimo de retweets o likes sino que es más relevante el país. Ya que las tendencias son muy diferentes dependiendo del país y a lo mejor al usuario le interesa entrar en un nuevo nicho de mercado. Además, es importante incluir el idioma ya que es común que los videos virales sean en inglés a lo mejor al usuario le interesa más en un idioma específico.

Figura 32: Filtros de la página de “Consultar tendencias”

Una vez se envía este formulario la estructura de la página es la misma que la de la página anterior. Es decir, primero los videos de TikTok y luego los tweets.

Figura 33: Videos en tendencias de TikTok

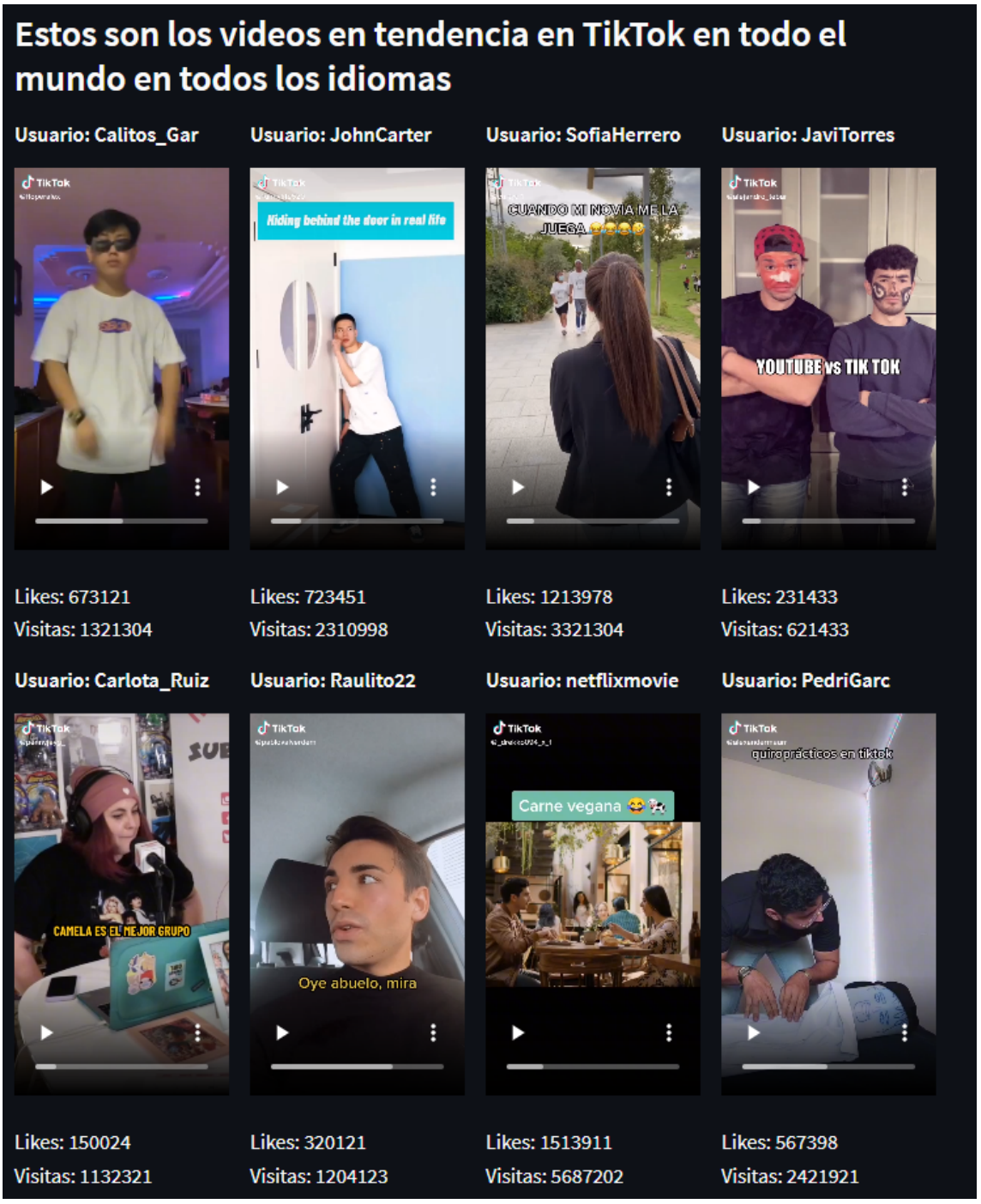
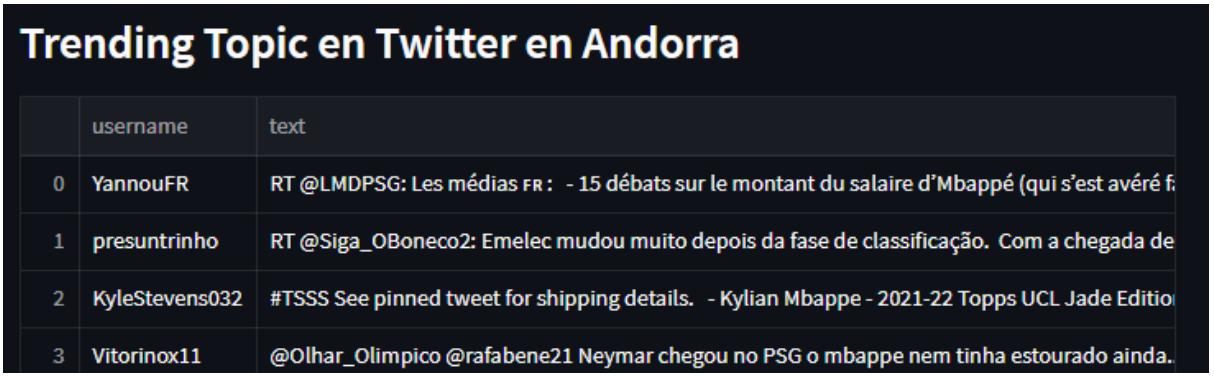


Figura 34: Tweets en trending topic



Trending Topic en Twitter en Andorra

	username	text
0	YannouFR	RT @LMDPSG: Les médias FR : - 15 débats sur le montant du salaire d'Mbappé (qui s'est avéré f
1	presuntrinho	RT @Siga_OBoneco2: Emelec mudou muito depois da fase de classificação. Com a chegada de
2	KyleStevens032	#TSSS See pinned tweet for shipping details. - Kylian Mbappe - 2021-22 Topps UCL Jade Editio
3	Vitorinox11	@Olhar_Olimpico @rafabene21 Neymar chegou no PSG o mbappe nem tinha estourado ainda.

Si clicas en el nombre de usuario en los tweets aparecen sus estadísticas:

Figura 35: Estadísticas de un usuario en específico

metrics_user_followers_count	metrics_user_following_count	metrics_user_tweet_count	metrics_user_listed_count
14	5	389	0

Si clicas en el tweet:

Figura 36: Estadísticas de un tweet en específico

↓ metrics_tweet_retweet_count	metrics_tweet_reply_count	metrics_tweet_like_count	metrics_tweet_quote_count
950	0	0	0

Capítulo 5. DISCUSIÓN

El proyecto nace a partir de una conversación con un influencer muy amigo mío que me comentaba el estrés que siente diariamente al no saber si al día siguiente la gente dejaría de verle o al no saber qué publicar. Y todo empezó con un “Te imaginas saber antes de publicar un video el número de likes que tendrá”. A partir de ahí, me puse a pensar en la solución y con los datos no hay nada imposible.

Al tratarse de un proyecto novedoso/innovador, la idea que tenía al principio de cómo iba a ser el desarrollo ha resultado ser completamente distinta a la realidad. El no poder basarme casi nada en otros estudios, ha hecho que todos los problemas que me he ido encontrando a lo largo del desarrollo lo haya tenido que resolver por mis propios medios sin poder entrar en ningún foro para buscar respuestas.

Eran tantas las opciones e ideas que tenía que uno de los retos más duros fue concretar específicamente de qué manera predecir el número de likes. No sabía si basarme en la portada del video y hacer un modelo de clasificación usando redes neuronales clasificando en viral o no viral o centrarme únicamente en lo que se decía en el video. Había tantos o 's que hasta que empecé con el proyecto definitivo pasaron semanas.

La parte más problemática ha sido la captura de datos a través de la API no oficial de TikTok. Por lo general utilizar APIs suele ser algo bastante sencillo ya que con solo registrarte te dan las credenciales y tienes acceso a una serie de funciones que te ofrece la API y dependiendo de qué acceso contrates tienes más o menos servicios. Al comercializar una API hace que sea obligatorio el buen funcionamiento. Pero en el caso de la API no oficial de TikTok, la mayoría de funciones daban errores. En algunos casos, estos errores eran corregibles pero en otros no. Es por esto que me he visto limitado a la hora de utilizar esta API. En más de una ocasión, he copiado el código de alguna función de la API y la he modificado para poder utilizarla como yo quería. Lo que en un principio parecía un ahorro de tiempo y una reducción de trabajo, se ha convertido en un quebradero de cabeza.

Capítulo 6. CONCLUSIONES

6.1 Conclusiones del trabajo

Cada vez las redes sociales tienen más peso en la sociedad creando tendencias e influencia en el público más joven y manipulable. Considerándose como uno de los canales de marketing digital más destacados en consumo. Debido a la gran suma de dinero que mueven las redes sociales cada vez salen a la luz nuevos influencers en busca de patrocinio. Para los influencers es vital asegurar su impacto. La repercusión en redes sociales es compensada económicamente por lo que los influencer dependen de sus seguidores. Esto genera estrés y provoca errores.

Ante este problema, mi aplicación ofrece una solución fiable e intuitiva aportando un algoritmo de machine learning capaz de predecir el número de likes que tendrá una publicación al cabo de x días. Además cuenta con otra funcionalidad que facilita al usuario un esquema visual de las publicaciones más relevantes actualmente relacionadas con una temática.

Lo que parecía imposible en un comienzo empezó a ser realizable gracias al potencial del big data y la cantidad de tecnologías que aparecen cada día que son cada vez más accesibles y compatibles entre ellas. Crear un entorno virtual combinando distintas tecnologías ha sido un reto técnico que ha requerido un gran esfuerzo en investigación y de reiteración de posibles escenarios hasta encontrar el ecosistema más eficiente y optimizado.

Concluyendo, pese a la gran complejidad de alguna de las fases del proyecto, los objetivos del presente proyecto han sido realizados por completo. Además, el trabajo llevado a cabo ha servido de gran aprendizaje por lo tanto se puede concluir el proyecto con una valoración muy positiva.

6.2 Conclusiones personales

Este proyecto ha sido el reto personal más difícil y largo en toda mi experiencia universitaria. Realmente es la primera vez que me he enfrentado a un proyecto tan duradero. Por lo general, en los grandes trabajos de la carrera he trabajo en equipo y ha sido el profesor el que me ha marcado la dirección del proyecto. Me he aventurado a desarrollar una idea propia e innovadora.

El trabajar con tantas tecnologías distintas y el saber desenvolverme con facilidad en cualquiera de ellas hace que entienda lo útil que han sido estos años de carrera. Pero algo que he aprendido es que nunca se sabe lo suficiente y sobre todo en el campo del big data. Cada día aparecen nuevas aplicaciones más complejas que las anteriores pero con un uso mucho más sencillo. Si dejas de investigar durante unos pocos meses, te quedas obsoleto.

Psicológicamente ha sido duro llevar a cabo este proyecto ya que se vive con mucho estrés pero a la vez con mucha ilusión. En cada problema que surgía me venía abajo pensando que

era una barrera la cual no se podía atravesar pero cada barrera atravesada generaba en mí un sentimiento de felicidad y motivación que nunca antes había sentido.

En resumidas cuentas, emocionalmente ha sido una auténtica montaña rusa.

Capítulo 7. FUTURAS LÍNEAS DE TRABAJO

Predecir la viralidad de un video en TikTok parecía algo imposible pero a medida que me he ido adentrando en los posibles datos que podría capturar me he dado cuenta de las miles de posibilidades que hay.

Pese a que la precisión del modelo de predicción es alta (80%), la variables más relevantes son las estadísticas del usuario. Esto es lógico ya que un usuario con un millón de seguidores siempre va a tener más que uno con 100. Pero lo más interesante del proyecto no es predecir los likes según quién eres únicamente sino que también según el video y descripción que publicas. Si no se tiene en cuenta el video y solo le damos importancia al usuario, cualquier video que publique va a tener los mismos likes para mi modelo. Para solventar este problema he tenido en cuenta los hashtag que aparecen en las descripciones de los videos ya que resumen de manera general el video. Al incluir los hashtag como variable, la precisión de mi modelo se disparó llegando al actual 80%.

Al usar un random forest me he focalizado en las variables numéricas y, por supuesto, no he trabajado con los frames de los videos ni con la portada del video. Aparte de las características técnicas del video como la duración o la calidad, el único dato utilizado del video es el número de palabras pronunciadas sin tener en cuenta cuales son las palabras exactas las cuales he podido extraer pero no he incluido en el modelo.

Puede ser interesante utilizar redes neuronales para clasificar la portada del video en viral o no viral. Al igual que trabajar con los frames del video.

Otra línea de trabajo es la detección de infracciones en las publicaciones que mi usuario quiera publicar. Por ejemplo, además de mostrar el número de likes que obtendrá en x días se podría verificar si aparece en algún momento del video un desnudo o violencia lo cual sería sancionable.

También considero que es importante llevar el almacenamiento de los datos a la nube ya que se trabaja con muchos datos y además para mostrarle al usuario los videos y para transcribirlos a texto, se están descargando cada uno de ellos. Por mi experiencia personal Google Cloud es una de las mejores opciones.

Por último, la experiencia del usuario tiene que ser la mejor posible y usar streamlit es fácil pero tiene muchas limitaciones sobre todo estéticas. Es importante desarrollar una aplicación web utilizando una aplicación como Wordpress o programando HTML y CSS.

Capítulo 8. REFERENCIAS

[1] Mónica Mena Roa (2022). *Infografía: La adicción a las redes sociales en el mundo. [online] Statista Infografías.*
https://es.statista.com/grafico/18988/tiempo-medio-diario-de-conexion-a-una-red-social/#:~:text=A%20nivel%20global%2C%20los%20usuarios,de%20un%20pa%C3%ADs%20a%20otro

[2] Asselin, C. (2022). *TikTok: cifras y estadísticas clave en España, Latam y el mundo 2022. [online] Digimind.com*
https://blog.digimind.com/es/agencias/tiktok-cifras-y-estadisticas-2020

[3] Bakshi, C. (2020). *Random Forest Regression - Level Up Coding. [online] Medium*.
https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

[4] davidteather (2022). *GitHub - davidteather/TikTok-Api: The Unofficial TikTok API Wrapper In Python. [online] GitHub.*
https://github.com/davidteather/TikTok-Api

[5] Business-tiktok.com. (2022). *Get started on TikTok Ads Manager. [online]*
https://www.business-tiktok.com/eu-gofulltiktok-es/?attr_source=google&attr_medium=search-br-ad&attr_adgroup_id=131982930689&attr_term=marketing%20tiktok&no_parent_redirect=1&gclid=Cj0KCQjwg_iTBhDrARIsAD3Ib5hlQL44CbZmurK2Rmb-6deOtDxW6dhzZ7qhLBarpjlZ-FMT3TPqcW8aAq0oEALw_wcB

[6] Leonardo Gamboa Uribe (2020). *Aprendizaje Supervisado, Clasificación y predicción, para todos…con BigML. [online] Medium*
https://novagenio.medium.com/aprendizaje-supervisado-clasificaci%C3%B3n-y-predicci%C3%B3n-para-todos-7ae2edb37e44

[7] CallRail. (2022). *What is Speech Recognition Software? How Does it Work? [online]*
https://www.callrail.com/blog/speech-recognition-software/

[8] Acervolima.com. (2021). **Introducción a MoviePy – Acervo Lima. [online]**
https://es.acervolima.com/introduccion-a-moviepy/#:~:text=MoviePy%20es%20un%20m%C3%B3dulo%20de,o%20para%20crear%20efectos%20avanzados

[9] Machine Learning. (2020). *Neural networks and speech recognition - Machine Learning. [online]*
https://www.gosmar.eu/machinelearning/2020/05/25/neural-networks-and-speech-recognition/

[10] de, C. (2006). *Escala Mel. [online]*
https://es.wikipedia.org/wiki/Escala_Mel

[11] Programmerclick.com. (2019). *Enseñarle a usar Python para el reconocimiento de voz. - programador clic. [online]*
https://programmerclick.com/article/16481215310/

[12] Python.org. (2019). *datetime — Basic date and time types — Python 3.10.4 documentation. [online]*
https://docs.python.org/3/library/datetime.html

[13] Twitter.com. (2022). *Twitter Developer Platform overview*. [online]
https://developer.twitter.com/en/docs/platform-overview?utm_campaign=DES-SMB_RE_GBL_EN_DevPlatform%20All%20Newly%20Approved%20Onboarding%20Campaign_V2%20Version_Day%200_2021&utm_medium=email&utm_source=Eloqua&ref=em-elq-ao-gbl-nurture

[14] WordPress.com. (2019). *WordPress.com: el alojamiento gestionado de WordPress más rápido y seguro*. [online]

https://wordpress.com/es/

Capítulo 9. ANEXOS

Figura 37: Business Model Canvas: propuesta de modelo de negocio