



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Seguridad Informática

**Aplicación antispam de comentarios para
prevenir delitos informáticos en la
plataforma de YouTube**

Trabajo fin de estudio presentado por:	Rubén Andrés Narváez Albuja
Tipo de trabajo:	Desarrollo de software
Director/a:	Víctor Andres Pimienta García
Fecha:	21/09/2022

Resumen

En la actualidad el spam sigue siendo un desafiante problema para las redes sociales tales como Twitter, Facebook y YouTube; este ha tomado fuerza con la ayuda de automatizaciones que se hacen uso de scripts para generar miles de comentarios engañosos, en muchas ocasiones personifican al creador de contenido para lucrarse económicamente estafando a los usuarios haciéndoles creer que han ganado premios y en otras situaciones leves, convencen a los usuarios para que ingresen en sitios web fraudulentos o de contenido sexual con la finalidad de obtener datos sensibles o difundir malware.

Los intentos de las plataformas por mitigar estas amenazas no son muy efectivos ya que los ciberdelincuentes siempre encuentran diferentes maneras de sobrepasar filtros; este proyecto logra crear un modelo Naïve Bayes para el reconocimiento de spam muy confiable, pero además presenta formas en las que se puede eliminarlo o al menos reportarlo de manera automatizada y masivamente.

Palabras clave: *YouTube, spam, bots, phishing, comportamientos de seguridad.*

Abstract

Nowadays spam is still a challenging problem for social networks such as Twitter, Facebook and YouTube; this has gained strength with the help of automations that use scripts to generate thousands of misleading comments, on many occasions they impersonate the content creator to profit economically by scamming users into believing that they have won prizes and in other minor situations, convince users to enter fraudulent or sexual related websites in order to obtain sensitive data or spread malware.

Platforms' attempts to reduce these threats are not very effective as cybercriminals always find different ways to bypass filters; This project manages to create a very reliable Naïve Bayes model for spam recognition, but also presents ways in which it can be eliminated or at least reported in an automated and massive way.

Keywords: *YouTube, spam, bots, phishing, security behavior.*

Índice de contenidos

1. Introducción	8
1.1. Justificación.....	8
1.2. Problemática	9
1.3. Estructura de la memoria	10
2. Estado del arte	12
2.1. Marco referencial.....	12
2.2. Trabajos relacionados	24
3. Objetivos y Metodología	28
3.1. Objetivos generales y específicos	28
3.2. Metodología.....	28
3.2.1. Recolección y clasificación manual de los datos	29
3.2.2. Entrenamiento de un modelo <i>machine learning</i> y su <i>testeo</i>	30
3.2.3. Diseño de un programa que reporte y elimine comentarios phishing automáticamente.....	30
4. Desarrollo	31
4.1. Obtención de las credenciales para utilizar el API de YouTube	31
4.2. Recolección de los comentarios de YouTube con YouTube Data API v3.....	40
4.3. Entrenamiento de un modelo por Naïve Bayes usando Scikit-Learn.	44
4.4. Diseño de un programa que reporte y elimine comentarios spam.....	58
4.5. Evaluación	63
5. Conclusiones.....	65
5.1. Líneas de trabajo futuro	68
6. Referencias bibliográficas	69
7. Anexo A. Nombres de usuarios marcados como spam	72

Índice de figuras

Figura 1. <i>Comentarios de YouTube eliminados a partir del cuarto trimestre de 2021.</i>	9
Figura 2. <i>Diagrama de flujo para desarrollo de un programa de clasificación de spam.</i>	29
Figura 3. <i>Pantalla de proyectos en la consola de Google Cloud.</i>	32
Figura 4. <i>Creación de un proyecto en la consola de Google Cloud.</i>	32
Figura 5. <i>Búsqueda de YouTube Data API v3 en la consola de Google Cloud.</i>	33
Figura 6. <i>Activación de YouTube Data API v3 en la consola de Google Cloud.</i>	34
Figura 7. <i>Activación de YouTube Data API v3 en la consola de Google Cloud.</i>	34
Figura 8. <i>Creación de credenciales para el YouTube Data API v3.</i>	35
Figura 9. <i>Creación de credenciales en la consola de Google Cloud.</i>	35
Figura 10. <i>Configuración de la pantalla de consentimiento de OAuth.</i>	36
Figura 11. <i>Configuración de permisos para el YouTube Data API.</i>	36
Figura 12. <i>Parámetros requeridos para la configuración de permisos.</i>	37
Figura 13. <i>Configuración del ID de cliente de OAuth para el YouTube Data API v3.</i>	37
Figura 14. <i>Obtención y descarga de credenciales para el YouTube Data API v3.</i>	38
Figura 15. <i>Obtención y descarga de credenciales para el YouTube Data API v3.</i>	39
Figura 16. <i>Pantalla de consentimiento al usar las credenciales de YouTube Data API v3.</i>	39
Figura 17. <i>Pantalla de consentimiento con un usuario de prueba.</i>	40
Figura 18. <i>Scraper de comentarios de YouTube.</i>	41
Figura 19. <i>Código fuente para entrenar un modelo predictivo por Naïve Bayes.</i>	44
Figura 20. <i>Resultados después de entrenar el modelo por Naïve Bayes.</i>	46
Figura 21. <i>Código de la figura 3 adaptado para clasificar los mensajes como spam.</i>	47
Figura 22. <i>Búsqueda de autores que han sido previamente marcados como spam.</i>	48
Figura 23. <i>Ejemplo de comentarios spam en los que su nombre de usuario contendría el spam.</i>	49

Figura 24. <i>Ejemplo de comentarios spam en los que su nombre de usuario sería genérico. ...</i>	50
Figura 25. <i>Entrenamiento Naïve Bayes (por nombres de usuario y por comentarios).....</i>	50
Figura 26. <i>Ejemplo de un antes y después de usar clean-text sobre un comentario cualquiera.</i>	52
Figura 27. <i>Código utilizado para limpiar el dataset previamente clasificado.</i>	53
Figura 28. <i>Dataset antes de realizarse la limpieza a los nombres de usuario.....</i>	54
Figura 29. <i>Dataset después de realizarse la limpieza a los nombres de usuario con clean-text.</i>	54
Figura 30. <i>Dataset antes de realizarse la limpieza de comentarios.</i>	55
Figura 31. <i>Dataset después de realizarse la limpieza de comentarios con clean-text.</i>	55
Figura 32. <i>Entrenamiento de dos modelos por Naïve Bayes (empleando un dataset limpio). </i>	56
Figura 33. <i>Limpieza previa de comentarios para ser analizados por el modelo Naïve Bayes. .</i>	56
Figura 34. <i>Limites en las cuotas por YouTube Data API v3.....</i>	59
Figura 35. <i>Bot para eliminar / reportar comentarios spam mediante el uso de chromedriver.</i>	60
Figura 36. <i>Respuesta resaltada como destacada por medio del ID de comentario.</i>	62
Figura 37. <i>Tipos de peticiones para lidiar con los comentarios spam.</i>	63

Índice de tablas

<i>Tabla 1. Resumen de la clasificación manual de comentarios en los videos seleccionados (15/08/2022).</i>	43
<i>Tabla 2. Comparación entre modelos Naïve Bayes vs YT-Spammer-Purge (15/08/2022).</i>	51
<i>Tabla 3. Comparación entre Naïve Bayes vs YT-Spammer-Purge (15/08/2022).</i>	57
<i>Tabla 4. Naïve Bayes vs YT-Spammer-Purge en cuestión al tiempo de ejecución (15/08/2022).</i>	64

1. Introducción

1.1. Justificación

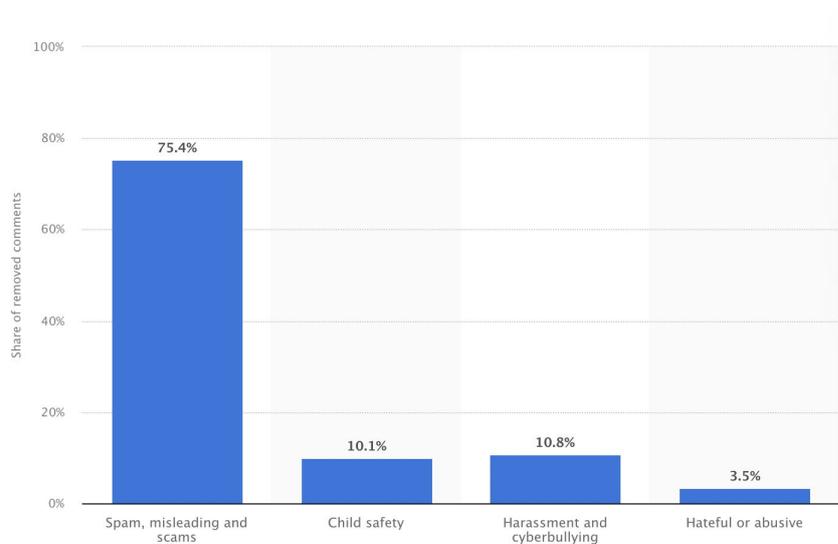
El spam que se encuentra en los comentarios de YouTube arruina la experiencia de los usuarios y afecta a muchos de ellos, siendo víctimas de los ciberdelincuentes mediante estafas de phishing. Por consiguiente, en muchas ocasiones se difama la identidad del creador y de la plataforma como tal.

En la actualidad YouTube, la plataforma de video más grande del mundo y con grandes cantidades de tráfico, fácilmente atrae la atención de muchos estafadores para realizar sus ataques hacia la comunidad que aquí se encuentra.

Por otra parte, se conoce que esta plataforma digital tiene más de 2600 millones de usuarios y millones de canales. Es decir, en otras palabras, cualquier canal que sea popular y cuente con un gran número de usuarios tiende a estar repleto de comentarios spam, sobre todo, de índole fraudulento o engañoso que se dan en canales relacionados con temas de tecnología, inversiones, criptomonedas y otros temas que involucran dinero. Estos comentarios de spam y phishing son intentos furtivos de atrapar a los usuarios en donde se los dirige a otro enlace para estafarlos.

No obstante, en muchas ocasiones es difícil darse cuenta que comentarios son spam por lo que existirán personas que piensen que son comentarios legítimos. De la misma forma, se debe tener en consideración que YouTube no puede mantener bajo control la inmensidad de comentarios que existen y que las funciones actuales de filtración no pueden hacer frente al masivo nivel de spam y sus prácticas evasivas.

Figura 1. Comentarios de YouTube eliminados a partir del cuarto trimestre de 2021.



Fuente: Statista.com (2021).

En la Figura 1 se puede observar que, durante el cuarto periodo del 2021, más del 75 por ciento de los comentarios en la plataforma de Google y YouTube, se eliminaron debido a que eran contenido spam, engañoso o fraudulento. Además, en el mismo trimestre, se eliminó aproximadamente el 10 por ciento de comentarios en los videos por razones de seguridad infantil. Lo cual resulta en más de 1200 millones de comentarios eliminados de YouTube durante el cuarto trimestre de 2021.

1.2. Problemática

En estos videos, hay dos tipos de comentarios de spam y ambos existen principalmente como resultado de los cambios que YouTube ha realizado en sus secciones de comentarios a lo largo del tiempo.

Un tipo de comentario de spam intenta alejar a los usuarios de YouTube del sitio y llevarlos a sitios con contenido para adultos. Esto es especialmente problemático porque muchos de los canales dirigidos atienden a un público más joven o de audiencia menor de edad.

Los estafadores que se hacen pasar por el dueño del canal cambiando su nombre y foto de perfil para que coincidan son el segundo tipo más común de spam. Luego responden a los comentarios afirmando que el usuario ha ganado un premio y dirigiéndolos a un sitio web

externa o sugiriendo que debería tomar contacto por otras plataformas como WhatsApp o Telegram.

Los ciberdelincuentes aprovechan el hecho de que YouTube no requiere nombres de usuario autenticados o verificados, lo que hace que hacerse pasar por otros usuarios y personas sea mucho más fácil. Luego, estos elaboran comentarios bien detallados para hacer uso de *bots* que difundan y automáticamente respondan a comentarios de la comunidad para esperar que alguno de ellos sea víctima de estos ataques.

Existen herramientas de código abierto y creadas por la propia comunidad de YouTube que identifican, clasifican y purgan el spam para que los creadores de contenido puedan proteger a sus seguidores y a sus canales. Estas herramientas están desarrolladas dentro de la comunidad de habla inglesa y requieren de la instalación de librerías para ejecutar código sin ningún tipo de interfaz, siendo estas no muy amigables con los usuarios debido a la usabilidad sin interfaz.

Otras herramientas y softwares más robustos brindan una mejor experiencia de usuario siendo efectivos eliminando el spam, pero no son gratuitos, ya que son más enfocados a monetizarlos con dueños canales de YouTube populares y de audiencia bastante grande.

A través de los años han surgido muchas más investigaciones sobre como identificar y clasificar el spam en las redes sociales de manera altamente eficiente y muy precisa; sin embargo, no se ha hecho mucho para poder mitigar todo el spam que es recabado por estos clasificadores.

1.3. Estructura de la memoria

El proyecto está compuesto por un conjunto de capítulos que detallan los pasos a seguir en todo el proceso. A continuación, un breve resumen sobre cada uno de ellos:

En la **introducción** se describe la justificación que motiva a la realización del trabajo, así como el planteamiento del problema encontrado y la estructura propuesta para solucionar el mismo.

En el **marco referencial** esta subdividido en dos secciones:

En la primera sección se da a conocer el origen del problema mediante investigación y referencias existentes en el contexto en que se desarrolla el proyecto. Se describirán múltiples

fuentes que aportarán datos relevantes a considerar para el desarrollo del trabajo, todo ello sustentará al proyecto.

Para la segunda sección, se probarán diferentes herramientas existentes que sean de código libre para así entender aún mejor las funciones y metodologías presentes en las mismas. De esa manera se estudiarán y clasificarán sus características con el fin de establecer una línea base de cara al desarrollo de funcionalidades de la aplicación, además de que, al identificarlas bien se podrá comprobar que el programa desarrollado en que aspectos es mejor a estas herramientas gratuitas.

En el capítulo de **objetivos y metodología** se describirá resumidamente las metas del trabajo y se enunciará la metodología seleccionada para el desarrollo del mismo. Para que al final, se pueda corroborar si dichos objetivos han sido cumplidos de manera satisfactoria o no.

En lo que se refiere al capítulo de **desarrollo** será todo el conjunto de actividades a realizar para el desarrollo de la aplicación. Se detallarán los pasos a seguir para construir la aplicación, así como la metodología escogida y todas sus funcionalidades. Se llevará a cabo una gran recopilación de mensajes spam ya existentes en la plataforma, dichos datos y sus fuentes serán detallados en este capítulo.

Finalmente, en las **conclusiones** se validarán si los objetivos establecidos al comienzo del proyecto se cumplieron, también se harán recomendaciones y se mencionarán puntos importantes a resaltar sobre el trabajo y futuras mejoras.

2. Estado del arte

2.1. Marco referencial

El spam está relacionado con la baja calidad de información y consiste en contenido no deseado. Por lo general, involucra información en textos, videos e imágenes. Además, la mayor parte del spam se usa para manipular a los usuarios de Internet y de esta forma obtener información personal como phishing y malware. También se usa con fines de publicidad. Generalmente funciona inundando Internet con el mismo mensaje para obligar al usuario a recibirlo, hay muchas investigaciones relacionadas con el spam en el estudio existente, tales como spam en blogs, en la web, en Twitter, en correo electrónico, en YouTube y spam en SMS.

En videos de YouTube en la sección de comentarios, aparecen varios tipos de comentarios spam que, si no se detectan o filtran correctamente, es probable que el canal en particular pierda su popularidad a medida que disminuyen las visualizaciones, los me gusta y los seguidores de ese video debido al bombardeo de los comentarios spam. Además, lo peor de esta situación es que el creador de contenido recibe remuneración de YouTube por obtener más vistas y suscriptores, pero debido a los comentarios spam, la remuneración antes mencionada a veces se ve disminuida. También se han informado varias investigaciones que indican diferentes técnicas de detección de spam para detectarlo con varios métodos.

Machine Learning

En primer lugar, hay que tener en cuenta que esta herramienta es una derivación de Inteligencia artificial cuya función principal es la de manejar gran cantidad de información y lograr un aprendizaje independiente de las maquinas o sistemas. Sin embargo, en la actualidad este sistema de aprendizaje no solo se involucra en temas de robótica también en algunos sectores como educación, finanzas, construcción y medicina (Russell, 2018).

Al igual que lo que sucede en otras aplicaciones que puede funcionar como un filtro de spam, contribuyendo a mejorar la vida de millones de personas.

A breves rasgos el *Machine Learning* consiste en programar a las computadoras para que aprendan de los datos que se les proporciona. Tal como puede ser utilizado para determinar y clasificar de manera sencilla que datos se consideran importantes y que otros son spam. Sin embargo, es clave que la información proporcionada sirva para entrenar ya sea mediante

ejemplos o palabras clave que podrían involucrar por ejemplo información sensible como tarjetas, gratis, etc.

De esta forma si se tratara de spam y con la información pertinente se puede escribir un algoritmo que detecte los patrones que se le ha enseñado; y el software realizaría una clasificación en base a los patrones detectados. Además, hay que tener en cuenta que se debe probar el programa hasta que se considere que los resultados son buenos.

Por otro lado, hay que tener en cuenta que cualquier programa que se desarrolle no es un software, sino que consiste en una lista extensa de reglas que ayudaran a resolver problemas.

Tipos de Sistemas de *Machine Learning*

Existen diferentes tipos de sistemas y se pueden dividir en categorías, dependiendo de si:

- Han sido entrenados con humanos o no.
 - Supervisado.
 - Sin supervisión.
 - Semi-supervisado.
 - Aprendizaje reforzado.
- Si pueden aprender gradualmente.
- Si funcionan simplemente comparando nuevos puntos de datos para encontrar puntos de datos, o pueden detectar nuevos patrones en los datos y luego construir un modelo (Russell, 2018).

Aprendizaje supervisado: Se introducen datos en el algoritmo con la solución deseada y se denominan "etiquetas". Es decir, se utiliza tareas de clasificación mediante patrones que pueden ser detectados y utilizados para etiquetar nueva información.

Los algoritmos supervisados más importantes:

- K-vecinos cercanos.
- Regresión lineal.
- Redes neuronales.
- Máquinas de vectores de soporte.
- Regresión logística.
- Árboles de decisión y bosques aleatorios.

Aprendizaje sin supervisión: A breves rasgos puede adivinar la información sin que este etiquetada mediante la revisión de similitudes o diferencias que permitan la clasificación.

Los algoritmos no supervisados más importantes:

- Clustering: k-means, análisis cluster jerárquico.
- Aprendizaje de reglas de asociación: Eclat, a priori.
- Visualización y reducción de dimensionalidad: kernel PCA, t-distributed, PCA.

Aprendizaje Semi-supervisado: Se conoce como un problema de aprendizaje que involucra una pequeña cantidad de ejemplos etiquetados y una gran cantidad de ejemplos que no han sido etiquetados. Sin embargo, los problemas de aprendizaje de este tipo son desafiantes ya que ni los algoritmos de aprendizaje supervisados ni los no supervisados pueden hacer un uso efectivo de las mezclas de datos etiquetados e indescritibles. Como tal, se requieren algoritmos de aprendizaje semi-supervisados especializados (Russell, 2018).

Aprendizaje reforzado: Dentro de este un agente el "sistema AI" observa el entorno y realiza determinadas acciones, luego recibe t recompensas a cambio. Con este tipo, el agente debe aprender por sí mismo.

Es decir, aquí los datos del sistema no se encuentran etiquetados, pero después de realizar algunas acciones y de cierto tiempo el sistema se retroalimenta a través de actualizaciones. Un ejemplo de este tipo de aprendizaje se puede encontrar en muchas aplicaciones de robótica que aprenden como caminar.

Consideraciones importantes del *Machine Learning*

Es necesario considerar que los sistemas de aprendizaje automático no son como los niños que pueden distinguir manzanas y naranjas en todo tipo de colores y formas. Por lo tanto, requieren muchos datos para funcionar efectivamente, ya sea que esté trabajando con programas y problemas muy simples, o aplicaciones complejas como procesamiento de imágenes y reconocimiento de voz, entre otros.

De otra forma, si se trabaja con datos de entrenamiento que contienen errores y datos atípicos para el sistema resultara muy difícil detectar patrones y funcionar correctamente. Así mismo, si se busca que un programa funcione se debe dedicar tiempo para limpiar los datos de entrenamiento.

Características Ingenieriles para considerar:

- Selección de las funciones más útiles.
- Extracción de características que combinen las existentes para proporcionar unas más útiles.
- Creación de nuevas funcionalidades a partir de datos.

Proceso de pruebas

Para cerciorarse que un modelo funciona bien y que este puede generalizar con nuevos casos se puede probar otros casos con él, poniendo el modelo en el entorno y luego monitoreando cómo funcionará. Esto es un buen método, pero si un modelo es inadecuado, el usuario se quejará (Russell, 2018).

Se puede dividir los datos en dos conjuntos, un conjunto para entrenamiento y el segundo para probar. De esta forma el error de generalización es la tasa por evaluación del modelo en el conjunto de prueba. El valor que se obtenga determinara si el modelo es bueno y si funcionará correctamente. Sin embargo, si la tasa es alta significa que el modelo funcionará mal. Lo recomendable seria que se use el 80% de los datos para entrenamiento y el 20% para propósitos de prueba, por lo que es muy simple probar o evaluar un modelo.

Teorema de Bayes

Es una herramienta utilizada en el campo de la probabilidad conocido por utilizarse en el campo del aprendizaje automático. Puede emplearse en el marco de probabilidad para ajustar un modelo a un conjunto de datos de entrenamiento (Brownlee, 2019).

Es importante mencionar que es un cálculo engañosamente simple y se puede usar para calcular fácilmente la probabilidad condicional de eventos donde la intuición a menudo falla.

Otra definición sencilla el teorema de Bayes es que calcula la probabilidad posterior de un evento. Es decir, emplea la probabilidad previa junto con la probabilidad del evento (Malik, 2020).

Para ejemplificarse los conceptos mencionados con anterioridad se considera estimar cómo se comporta una variable objetivo. Esta variable objetivo puede ser de naturaleza aleatoria.

Por lo tanto, se recopilan los datos para la muestra aleatoria. Estos datos representan un conjunto de muestra con su propia distribución de muestreo. Como cada muestra tiene diferentes puntos de datos, la distribución puede ayudar a cuantificar los errores en las técnicas de muestreo.

Una vez que se recopilan las muestras y se realizan los experimentos se utiliza el teorema de Bayes para obtener nueva información para actualizar la comprensión previa. Es decir, el teorema de Bayes es un marco que permite calcular la probabilidad de que ocurra un evento dado que el otro evento ya ocurrió.

Algoritmo Naïve Bayes, (NBC)

Es indispensable conocer que son las redes bayesianas antes de adentrarse en una definición del algoritmo de Naïve Bayes, estas son importantes no sólo porque ofrecen un análisis cualitativo de los atributos y valores que pueden intervenir en el problema, sino también porque da cuenta de la importancia cuantitativa de esos atributos.

Dentro del aspecto cualitativo se puede representar cómo se relacionan los atributos ya sea en una forma causal, o señalando simplemente la correlación que existe entre variables o atributos. Cuantitativamente da una medida probabilística de la importancia de esas variables en él. Por otra parte, esta es quizá una de las diferencias fundamentales que ofrecen las redes bayesianas con respecto a otros métodos como pueden ser los árboles de decisión y las redes neuronales, que no dan una medida cuantitativa de esa clasificación.

En adición a estas consideraciones, el aprendizaje basado en redes bayesianas es especialmente adecuado en ciertas tareas como puede ser la clasificación de textos, siendo incluso más eficiente que otros.

Ahora bien, el algoritmo Naïve Bayes se conoce también como el Ingenuo Bayes que es uno de los algoritmos más simples y poderosos para la clasificación que se basa en el Teorema de Bayes con una suposición de independencia entre los predictores. Además, este algoritmo es fácil de construir y particularmente útil para conjuntos de datos muy grandes.

Este clasificador probabilístico tiene una fuerte suposición de independencia. Sin embargo, la suposición de la independencia de los atributos se considera una suposición pobre y se viola

a menudo para los conjuntos de datos verdaderos. También proporciona una mejor precisión de clasificación en conjuntos de datos en tiempo real en comparación a cualquier otro clasificador. Al igual que requiere una pequeña cantidad de datos de entrenamiento.

El clasificador Naïve-Bayes aprende de los datos de entrenamiento y luego predice la clase de la instancia de prueba con una probabilidad mayor a la posterior. Así mismo, es útil para datos dimensionales altos ya que la probabilidad de cada atributo se estima independientemente.

Se asume que el efecto de una característica particular en una clase es independiente de otras características. Incluso si estas características son interdependientes, estas características se consideran de forma independiente. Esta suposición simplifica la computación, y por eso se considera ingenua. Esta suposición se denomina independencia condicional de clase.

Puntos fuertes del Algoritmo Naïve Bayes

- Es una manera fácil y rápida para predecir clases también sirve en problemas de clasificación binarios y multiclase.
- En casos en que sea apropiada una presunción de independencia, el algoritmo se comporta mejor que otros modelos de clasificación, inclusive cuando se tienen menos datos de entrenamiento.
- El desacoplamiento de las distribuciones de características condicionales de clase significa que cada distribución puede ser estimada independientemente como si tuviera una sola dimensión. Esto ayuda con problemas derivados de la dimensionalidad y mejora el rendimiento.

Puntos débiles del Algoritmo Naïve Bayes

- Los algoritmos Naïve Bayes son conocidos por ser estimadores pobres. Por lo mismo, no se deben tomar muy en serio las probabilidades que se obtienen.
- La presunción de independencia Naïve con probabilidad no reflejará cómo son los datos en el mundo real.
- Cuando el conjunto de datos de prueba tiene una característica que no ha sido observada en el conjunto de entrenamiento, el modelo le asignará una probabilidad de cero y será inútil realizar predicciones. Además, uno de los principales métodos para

evitar esto es la técnica de suavizado, siendo la estimación de Laplace una de las más populares.

Probabilidad bayesiana

El término bayesiano deriva del matemático y teólogo del siglo XVIII Thomas Bayes, quien proporcionó el primer tratamiento matemático de un problema no trivial del análisis de datos estadísticos utilizando la inferencia bayesiana. Sin embargo, el matemático Pierre-Simon Laplace fue a quien se le atribuye la difusión de la probabilidad bayesiana.

Ahora bien, una probabilidad bayesiana trata de predecir la probabilidad de que ocurran ciertos eventos en el futuro. Sin embargo, es diferente a una probabilidad tradicional debido a que usa una frecuencia para estimar la probabilidad (Shuttleworth, 2009).

Además, este tipo de probabilidad se expresa como un porcentaje. Así en un evento con una probabilidad bayesiana de 0,6 (o 60 %) debe interpretarse como "Con una confianza del 60 %, este evento contiene el resultado verdadero".

También se puede definir como la medida de confianza o creencia que una persona tiene en una proposición. Sin embargo, el uso de la probabilidad bayesiana le permite a un investigador juzgar la cantidad de confianza que tiene en un resultado en particular. La probabilidad de frecuencia, a través de la hipótesis nula tradicional, restringe al investigador a respuestas de sí y no.

Cabe mencionar que los métodos bayesianos se están convirtiendo en una herramienta para evaluar la viabilidad de una hipótesis de investigación.

Por otra parte, para utilizar este tipo de probabilidad un investigador parte de un conjunto de creencias iniciales y trata de ajustarlas, mediante la experimentación y la investigación. De esta forma el conjunto original de creencias se altera luego para acomodar la nueva información. Este proceso sacrifica un poco de objetividad por flexibilidad, lo que ayuda a los investigadores a eludir la necesidad de un diseño de investigación tortuoso.

Para entender de manera sencilla a una probabilidad bayesiana se puede tomar como ejemplo, la acción de lanzar un dado: la teoría de la frecuencia tradicional establece que, si se

lanza los dados seis veces, debe lanzar un seis una vez. Por supuesto, puede haber variaciones, pero estas se promedian con el tiempo. Un proceso similar, aunque más complejo involucra predecir el clima basándose en eventos y ocurrencias anteriores, acertando la mayoría de las veces.

Acotando a los ejemplos anteriores también se pueden citar los filtros de spam en las cuentas de correo electrónico que emplean el teorema de Bayes y hacen un trabajo bastante bueno. Aunque no interceptan todos los correos electrónicos no deseados y pueden asignar erróneamente mensajes legítimos a la carpeta de basura, ciertamente son mejores que tener cientos de mensajes basura esperando en la bandeja de entrada cada vez que se abre la cuenta. Además, cada vez que el programa hace una suposición incorrecta, que es señalada por el destinatario, la nueva información retroalimenta el modelo y facilita una respuesta más precisa la próxima vez.

Modelado: representación de la información

Toda teoría matemática útil en la práctica sobre el procesamiento de la información en un "sistema" dado debe tener una forma de representar y cuantificar numéricamente la información. Por lo tanto, la probabilidad bayesiana utiliza la siguiente representación:

1. Variables: El "sistema" se describe completamente mediante una serie de variables, a las que se les pueden dar valores numéricos. Estas variables pueden ser discretas o continuas también pueden ser escalares, es decir, que consisten en un solo número, o compuesto que consiste en más de un solo número (Bruyninckx, 2002).
2. Información: la información sobre las N variables en el "sistema" es una función de un solo valor sobre el espacio base N -dimensional de las variables.
3. Relaciones: las variables del "sistema" no pueden tomar valores arbitrarios independientes entre sí. Por lo tanto, hay relaciones funcionales que indican la interdependencia de los variables. Además, estas funciones de relación no necesitan ser de un solo valor.
4. Estado: El "sistema" se encuentra, en un instante particular de tiempo, es decir, las funciones de información sobre las variables del sistema tienen valores particulares. El

conjunto de todos los valores posibles que pueden tener las variables de estado se denomina espacio de estado.

Distribución Gaussiana

Esta distribución recibe su nombre debido al matemático Carl Friedrich Gauss, quien se dio cuenta de la forma de la curva mientras estudiaba la aleatoriedad de los errores. Su importancia radica en su utilidad en probabilidad y estadística.

Ahora bien, este tipo de distribución también se conoce como distribución normal y es la más importante en estadística debido a la forma del gráfico. Además, este tipo de probabilidad tiene un tipo de distribución continua que es simétrica con respecto a su media; la mayoría de las observaciones se agrupan alrededor de la media, y cuanto más lejos está una observación de la media, menor es su probabilidad de que ocurra. Al igual que otras distribuciones de probabilidad, la distribución gaussiana describe cómo se distribuyen los resultados de una variable aleatoria (Math.net, 2022).

Algunas de las mediciones repetidas de cantidades siguen una estrecha aproximación a la distribución gaussiana. Así mismo, se pueden citar ejemplos de altura y peso de las personas, azúcar en la sangre, presión arterial, notas en los exámenes, errores en las cantidades medidas, entre otras.

Propiedades de la Distribución Gaussiana

- La distribución gaussiana es una función de probabilidad continua para una variable aleatoria.
- Es simétrica con respecto a la media.
- La media, la mediana y la moda de la distribución gaussiana son iguales y caen en el punto con la frecuencia máxima (Educative Inc., 2022).

Diferencia entre distribución gaussiana y distribución normal

A pesar de que la distribución gaussiana también se la denomina distribución normal existe una clara diferencia entre ambas.

En la distribución gaussiana la mayoría de los datos se centran a una medida con cierta dispersión o varianza. De manera específica, una distribución gaussiana es simétrica y tiene una media y una varianza constantes (Pisa, 2022).

Por lo tanto, permite hacer predicciones sobre un valor desconocido cuando se tiene un conjunto de valores conocidos que siguen una distribución gaussiana. Si la media es cero y la varianza es uno, se conoce como distribución normal estándar.

Es importante mencionar que los términos distribución gaussiana, distribución normal, curva de campana, campana de Gauss se refieren a lo mismo.

Distribución Bernoulli

Es un tipo de distribución de probabilidad discreta en la que cada experimento realizado plantea una pregunta que solo puede responderse con un sí o un no. Es decir, la variable aleatoria puede ser 1 con probabilidad (p) o puede ser 0 con probabilidad ($1 - p$). Por lo tanto, este experimento se conoce como la prueba de Bernoulli (CUEMATH, 2022).

Cabe mencionar que este tipo de distribución se usa para modelar ejemplos de la vida real y también puede emplearse en diferentes de aplicaciones. Aunque considerando que un experimento aleatorio solo puede tener un resultado de 1 o 0.

Es necesario mencionar que este tipo de distribución es muy fácil de entender y también se puede emplear como base para derivar distribuciones más complejas. También se puede emplear para describir eventos que solo pueden tener dos resultados, pudiendo ser estas, éxito o fracaso. Al ser una distribución simple se utilizada en muchos campos como la medicina, ingeniería, entre otras industrias.

Sin embargo, hay algunos términos básicos que se deben comprender:

Evento de Bernoulli: el evento tiene solo dos resultados posibles, estos pueden verse representados como Éxito o Fracaso, Sí o No, o citando un ejemplo sencillo de Cara o Cruz.

Prueba de Bernoulli: Una prueba de Bernoulli es una instanciación de un evento de Bernoulli. Es uno de los experimentos más simples que se pueden realizar en probabilidad y estadística. Es un experimento en el que hay dos resultados posibles (éxito y fracaso).

Se debe considerar que una parte importante de cada prueba de Bernoulli genere una acción que debe ser independiente. Es decir, que las probabilidades deben permanecer iguales a lo largo de las pruebas. Por lo tanto, cada evento debe estar completamente separado y no tener nada que ver con el evento anterior (Statistics How To , 2022).

El proceso de Bernoulli conduce a varias distribuciones de probabilidad:

- La distribución binomial.
- La distribución geométrica.
- La distribución binomial negativa.

También es importante mencionar que la distribución de Bernoulli está estrechamente relacionada con la distribución Binomial. Además, siempre que cada prueba individual de Bernoulli sea independiente, el número de éxitos en una serie de pruebas de Bernoulli tiene una distribución Bernoulli.

Distribución Multinomial

Conocida por ser una versión multivariante de la distribución binomial. Además, de ser la distribución de probabilidad de los resultados de un experimento multinomial.

Este tipo de distribución se utiliza en el caso de un experimento que tiene la posibilidad de dar como resultado más de dos resultados posibles.

También se puede destacar una técnica simple para construir una variable aleatoria multinomial, replicando un experimento.

Aspectos importantes

- La distribución multinomial representa la probabilidad de recibir un determinado conjunto de conteos donde cada ensayo tiene un número discreto de posibles resultados.
- La prueba de bondad de ajuste más directa se basa en la distribución multinomial de patrones de respuesta.
- Sus aplicaciones y casos de uso implican frecuentemente la evaluación de la probabilidad de un conjunto de resultados que suelen ser más de dos o al menos dos.
- Es un tipo de distribución de probabilidad de la cual la distribución binomial es un subtipo.

La distribución multinomial es una distribución discreta multivariada. Los experimentos multinomiales incluyen las siguientes características:

- El experimento consta de n ensayos repetidos.
- Sus resultados tienen significación estadística.
- Hay un número finito de resultados posibles para cada prueba, y la probabilidad de que ocurra cualquier evento es constante durante todo el experimento.
- Los resultados de un experimento no influyen en los resultados de los demás.

Suponiendo que un modelo sea válido, la forma más sencilla de determinar el ajuste de un modelo es utilizar la distribución multinomial de los patrones de respuesta. En consecuencia, existen patrones de respuesta potenciales para n ítems con m categorías de respuesta para cada ítem. Por tanto, todos los patrones de respuesta tienen la misma probabilidad de ocurrir si hay un patrón típico.

Por lo tanto, para estimar las frecuencias de los patrones de respuesta se emplea una distribución multinomial con parámetros n y probabilidad real para todos los patrones de respuesta. De acuerdo con el teorema del límite central multivariante, la distribución normal multivariante puede aproximarse a la distribución para tamaños de muestra grandes (Vaidya, 2022).

Al considerar los datos completos, la distribución de las observaciones tiene una forma multinomial para las observaciones de diferentes distribuciones de Poisson. Las probabilidades en la distribución multinomial se basan en la media de Poisson para cada celda multiplicada por todos los valores medios de Poisson.

2.2. Trabajos relacionados

YouTube está tratando de solucionar el problema del spam (Marques & Andrew, 2022). Muchos grandes creadores de YouTube han expresado su frustración con el aumento de comentarios no deseados en sus canales. El problema ha sido particularmente agudo para los creadores de alto perfil y de gran audiencia, quienes a menudo ven comentaristas maliciosos haciéndose pasar por ellos en un intento de estafar a sus espectadores. Marques relata que recibe emails de sus suscriptores preguntando por la veracidad de los comentarios que han recibido ofreciendo regalos y sorteos; otros emails mencionan sobre cuando recibirán su premio una vez que ya han depositado su dinero para pagar los gastos de envío, algo que nunca fue cierto. Esto resulta que, en ocasiones la comunidad se enfada con los dueños del canal o que muchos usuarios decidan ya no comentar para evitar estos mensajes.

El spam de comentarios de YouTube puede adoptar muchas formas. Los principales creadores a menudo están preocupados por el spam que se hace pasar por ellos, promete a los espectadores enviarles algo bueno para luego dirigir a las personas fuera de YouTube de alguna manera para eventualmente estafarlos. Otros comentarios de spam pueden ser menos maliciosos, pero igualmente molestos o potencialmente dañinos. Marques comenta que YouTube está intentando mitigar el spam; aunque la comunidad ha tratado de crear herramientas para combatir este gran problema por su cuenta propia.

Un enfoque para la detección de spam en los comentarios de YouTube basado en el aprendizaje supervisado (Amir & Zain, 2016) artículo que nos dice es muy importante encontrar una manera de detectar el spam en los videos de YouTube y denunciarlos. En este trabajo, se han evaluado varias técnicas de clasificación de alto rendimiento para tal fin.

Se entrenan varios modelos de aprendizaje automático utilizando un *dataset* público del repositorio UCI Machine Learning Repository. Los algoritmos utilizados fueron: Multilayer Perceptrons, Support Vector Machine, Random Forest, Logistic Regression, k-Nearest Neighbor and Naïve Bayes. El análisis estadístico de los resultados indica que el Perceptrón

multicapa y la Máquina de vectores de soporte muestran buenos resultados de precisión utilizando las librerías de Scikit-Learn.

El objetivo de este trabajo es encontrar qué algoritmos proporcionan la mejor y más alta exactitud y precisión para ayudar a detectar comentarios de spam en YouTube.

Trabajos similares y que emplean el mismo *dataset*, se encuentran los proyectos: **N-GRAMS ASSISTED YOUTUBE SPAM COMMENT DETECTION** (Reddy, Nithin, Meghana, & Sindhu, 2022); **Comparación de diferentes algoritmos de aprendizaje automático para clasificar comentarios de spam en videos de YouTube** (Polanski, 2022) y **Marco de detección de spam en YouTube usando Naïve Bayes y regresión logística** (Samsudin, y otros, 2019) para crear las funciones de clasificación spam mediante el uso de Naïve Bayes y Logistic Regression; en el último artículo mencionado, se probaron los modelos en dos herramientas de minería de datos diferentes, que son Weka y Rapid Miner. Las funciones que se probaron en Weka y RapidMiner muestran una alta precisión, por lo que se utilizan en todo el experimento de esta investigación. Los resultados obtenidos por Naïve Bayes y Logistic Regression en Weka es ligeramente superior al de RapidMiner. Además, el resultado de Naïve Bayes es superior al de Logistic Regression con 87,21 % y 85,29 % respectivamente en Weka. Mientras que en RapidMiner hay una ligera diferencia de precisión entre Naïve Bayes y Logistic Regression 80.41% y 80.88%. Pero, la precisión de Naïve Bayes es mayor que la de Regresión logística.

Por otro lado, en el artículo **Detección de comentarios de spam de Youtube asistida por N-Gram** (Aiyar & Shetty, 2018) se realizaron los mismos experimentos, pero utilizando un propio *dataset* ya que los autores recopilaron 13000 comentarios utilizando el API de YouTube, los comentarios fueron extraídos principalmente de videos musicales y de bastante popularidad en aquel momento.

El artículo también concluye que a lo largo de los años YouTube ha atraído a un número cada vez mayor de usuarios. Esto a su vez, ha llevado al crecimiento de usuarios maliciosos que han comenzado a desarrollar *bots* automatizados capaces de difundir mensajes de spam orquestados a gran escala a través de múltiples canales de manera simultánea. La presencia de estos comentarios daña significativamente la reputación de un canal y también la experiencia de usuario. Se hace hincapié de que YouTube ha intentado abordar este problema con métodos muy limitados que giran en torno al bloqueo de comentarios que contienen

enlaces; pero que dichos métodos han demostrado ser extremadamente ineficaces ya que los *spammers* han encontrado formas de eludir tales heurísticas.

Detección de spam en Twitter utilizando clasificadores tradicionales es un artículo elaborado por M. McCord y M. Chuah. En el mismo se analizan los distintos algoritmos de aprendizaje automático o *machine learning* en otra grande plataforma donde predomina el spam llamada Twitter. Los autores analizan diferentes comportamientos y patrones en los comentarios para poder identificar a los *spammers* de los usuarios común y corrientes. Todos los datos fueron recopilados por ellos usando el API proporcionado por Twitter, tomando como base 1000 usuarios escogidos aleatoriamente y sus 100 publicaciones más recientes para que posteriormente se evaluará y clasificará manualmente cada una de ellas como spam o no spam; tomando como referencia diferentes características del usuario y del contenido que publican. Entre ellos, otros factores como el número de seguidores, frecuencia en sus publicaciones, uso de URLs, hashtags, entre otros (McCord & Chuah., 2011).

Durante su desarrollo, probaron diferentes algoritmos de aprendizaje automático:

- Random Forest
- Support Vector Machine
- Naïve Bayesian
- K-Nearest Neighbor

Se llega a la conclusión de que el modelo entrenado por el algoritmo de Random Forest resulta ser el más preciso con respecto a los otros, además que se podría mejorar mucho más el modelo de clasificación ampliando el *dataset* y utilizando otras plataformas como Facebook para la obtención del *dataset* y así realizar comparativas esperando resultados similares.

Detección de reseñas falsas usando un aprendizaje automático supervisado (Elmogy, Tariq, Mohammed, & Ibrahim, 2021). En este artículo se muestra la importancia de las reseñas son considerados principalmente como un factor crucial para construir y mantener una buena reputación en productos y servicios. Además, las reseñas influyen directamente en el proceso de toma de decisiones para los usuarios. Los autores proponen el enfoque de un aprendizaje automático para identificar reseñas falsas. El documento compara el rendimiento de varios experimentos realizados en un conjunto de datos tomados de Yelp, plataforma de reseñas de restaurantes; se entrenaron varios modelos y se realizaron comparaciones en el desempeño

de los clasificadores escogidos: KNN, Naïve Bayes, SVM, Logistic Regression y Random Forest. Los resultados revelan que el clasificador KNN supera al resto de clasificadores en la detección de reseñas falsas.

YT-Spammer-Purge, proyecto mantenido por ThioJoe, es un programa creado con la finalidad de combatir los comentarios spam en la actualidad, es de código abierto y se encuentra en los repositorios de GitHub (ThioJoe, 2022). El programa utiliza una combinación de filtros para identificar la probabilidad de que un texto sea spam o no a diferencia de entrenar un modelo para que clasifique los comentarios, el mismo aplica un conjunto de filtros sobre cada uno de los comentarios, es bastante confiable ya que específicamente se centra en los comentarios que tratan de redireccionar al usuario a tomar contacto por WhatsApp, Telegram, Instagram, etc. Los cuales han tomado fuerza en los últimos años sobre la comunidad de YouTube. Entre algunos de sus filtros comprenden:

- Escanear comentarios duplicados si los mismos son más de 5
- Comparar la similitud entre comentarios 90% de similitud.
- Uso de algoritmo Levenshtein distance para detectar comentarios similares.
- Detección de usuarios que copien al creador del canal (personificación).
- Detección de links spam / fraudes.

Una vez filtrado los comentarios el programa brinda diferentes opciones para purgar el spam; es altamente efectivo y podría ser usado como referencia para comparar con el proyecto a desarrollar ya que comparten ciertas similitudes y cumplen un mismo fin.

Finalmente, el documento **Evaluación Crítica Sobre La Detección De Contenido De Spam En Redes Sociales** (Chrismanto, Sari, & Suyanto, 2022) tuvo como objetivo llevar a cabo una revisión exhaustiva sobre la detección de contenido de spam en populares redes sociales, incluyendo YouTube. Se recopilan datos de literatura desde 2015 hasta 2021, entre ellas siete bases de datos de repositorios de revistas y 69 artículos. Se llevaron a cabo debates sobre el enfoque, los medios de investigación, el conjunto de datos recopilados, su forma de extracción y clasificación definiendo ciertas características específicas, el lenguaje basado en el contexto o no, el algoritmo, el rendimiento, la dirección de la investigación futura y los desafíos. Se concluye también que hay un aumento en la investigación sobre temas de detección de spam a través de los años y que, en dichos sistemas de detección, predomina el lenguaje inglés.

3. Objetivos y Metodología

3.1. Objetivos generales y específicos

Objetivo General:

Desarrollar una aplicación que automatice las tareas de reportar y eliminar comentarios spam de YouTube para proteger a los usuarios de mensajes engañosos y fraudulentos que buscan lucrarse de las personas.

Objetivos Específicos:

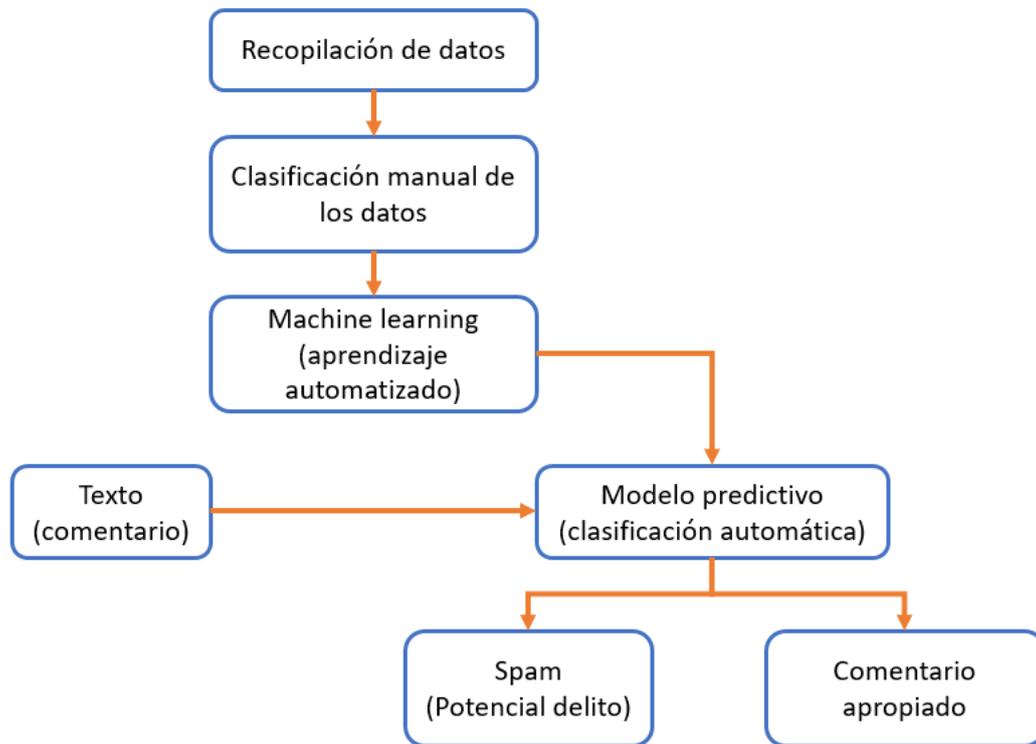
- Recopilar una gran cantidad de comentarios spam ya existentes en canales en YouTube relacionadas con dinero, inversiones y tecnología para así entrenar un modelo que identifique si un texto en español es spam o no.
- Asegurar que la aplicación tenga una alta confiabilidad clasificando comentarios como spam y tenga una tasa baja de falsos positivos.
- Diseñar una interfaz fácil de entender y manipular para el usuario.
- Desarrollar una aplicación antispam que sea de libre acceso para cualquier persona.
- Brindar una herramienta de seguridad ante la suplantación hacia los usuarios de YouTube.
- Evitar al máximo posible que se puedan perpetrar estafas hacia las personas purgando comentarios mal intencionados en YouTube.
- Promover el mantener una comunidad de YouTube libre de spam.

3.2. Metodología

La metodología a desarrollarse en el presente trabajo se la ha dividido en tres partes:

- Recolección y clasificación manual de los datos.
- Entrenamiento de un modelo *machine learning* y su *testeo*.
- Desarrollo de un programa que reporte y elimine comentarios phishing automáticamente.

Figura 2. Diagrama de flujo para desarrollo de un programa de clasificación de spam.



Fuente: Elaboración propia.

3.2.1. Recolección y clasificación manual de los datos

Para obtener una gran cantidad de comentarios spam se los buscara en videos de diferentes canales de YouTube relacionados con temáticas de dinero, inversiones y criptomonedas ya que allí es donde realizan principalmente sus ataques los ciberdelincuentes, principalmente en videos que tengan muchas visitas. En concreto se seleccionarán diez videos en inglés y español para un mejor análisis del phishing.

Al momento de tener toda la información recopilada se procederá a clasificarla manualmente a cada uno de los comentarios, esto bajo el criterio del autor, las características más importantes para considerar un comentario como potencial phishing y/o spam se mencionan a continuación:

- Que el comentario se haya repetido al menos una vez.
- Que el comentario contenga temas relacionados a premios, regalos o sea de índole sexual.
- Que el usuario que realizo el comentario pretenda ser el dueño del video copiando el nombre de usuario del dueño o de alguna manera personifique al dueño del canal.

- Que el comentario intente redirigir al usuario hacia otras plataformas (WhatsApp, Telegram, Instagram, otros sitios web).

3.2.2. Entrenamiento de un modelo *machine learning* y su *testeo*

Con una serie de datos bien clasificados se podrá realizar un entrenamiento a un modelo de *machine learning* por Naïve Bayes, el cual se lo ha escogido por su sencillez y rapidez, además de que es ampliamente empleado en clasificadores de spam.

Con el modelo generado se podrán realizar varias pruebas, comprobar su eficacia falsos positivos y margen de error.

3.2.3. Diseño de un programa que reporte y elimine comentarios phishing automáticamente.

Finalmente se podrá extraer comentarios de YouTube en tiempo real para que de manera automática estos puedan ser clasificados como spam o no, en el caso de ser spam el comentario será eliminado y reportado junto a la cuenta que lo creó. Esto gracias al API de YouTube, el cual permitirá realizar todas esas acciones, que conjuntamente con el modelo entrenado conformarán la aplicación antispam; también se analizarán las restricciones y limitaciones del API de YouTube y se buscarán alternativas al mismo y también se analizará el programa en sí, su usabilidad y efectividad tratando el spam actual en la plataforma de YouTube.

4. Desarrollo

Los diez videos de YouTube seleccionados serán usados como *datasets* para entrenar el modelo por Naïve Bayes, por ello, se escribió un código usando el API de YouTube para obtener todos los comentarios de los videos escogidos, este los recopilará en archivos CSV independientemente del ID del video.

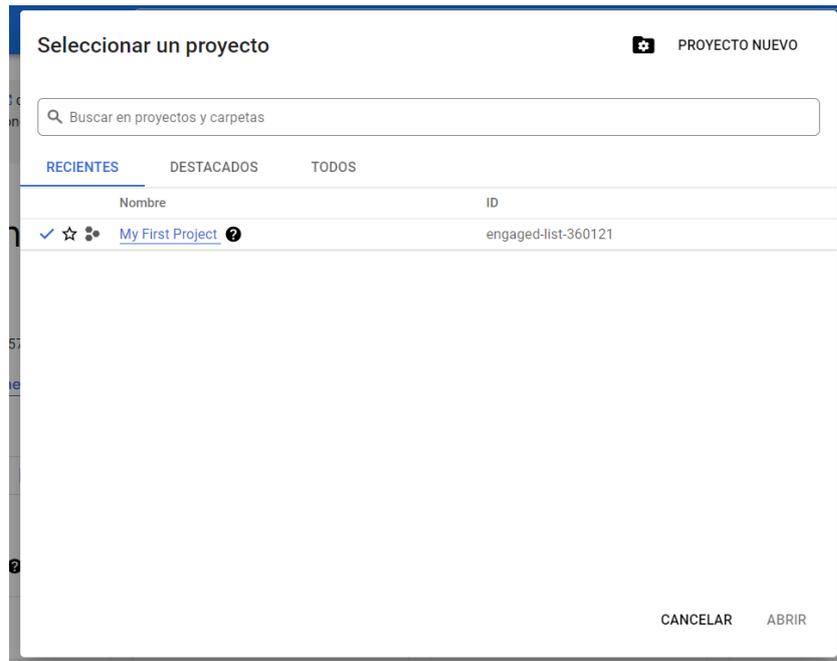
4.1. Obtención de las credenciales para utilizar el API de YouTube

Para poder conseguir todos los comentarios que existan en un video determinado, de manera rápida y eficiente se usará el API de YouTube Data API v3; el mismo también servirá para obtener otros datos relacionados con un video específico como: título del video, descripción, contador de visitas, contador de comentarios, comentarios y *sub-comentarios* (respuestas al comentario) en texto claro así como su ID de comentario y usuario además de toda la información relacionada con el usuario autor y su canal.

A continuación, se detallarán los pasos a seguir para activar este API en una cuenta de Gmail y crear las credenciales; las cuales serán necesarias para descargar todos los comentarios de los videos seleccionados y formar así el *dataset* que servirá para entrenar al modelo Naïve Bayes, también se usaran las credenciales para el programa final, ya que de igual manera se deberán obtener todos los comentarios de un video en particular para que el modelo pueda clasificarlos y posterior a ello tomar acciones con los comentarios spam. Por ese motivo es necesario que también el usuario que desee beneficiarse de este programa deberá también realizar estos pasos para poder hacer uso del API y del clasificador de comentarios en sí.

Para empezar, se ingresa a la consola de Google Cloud: <https://console.cloud.google.com/> y se selecciona la opción de PROYECTO NUEVO.

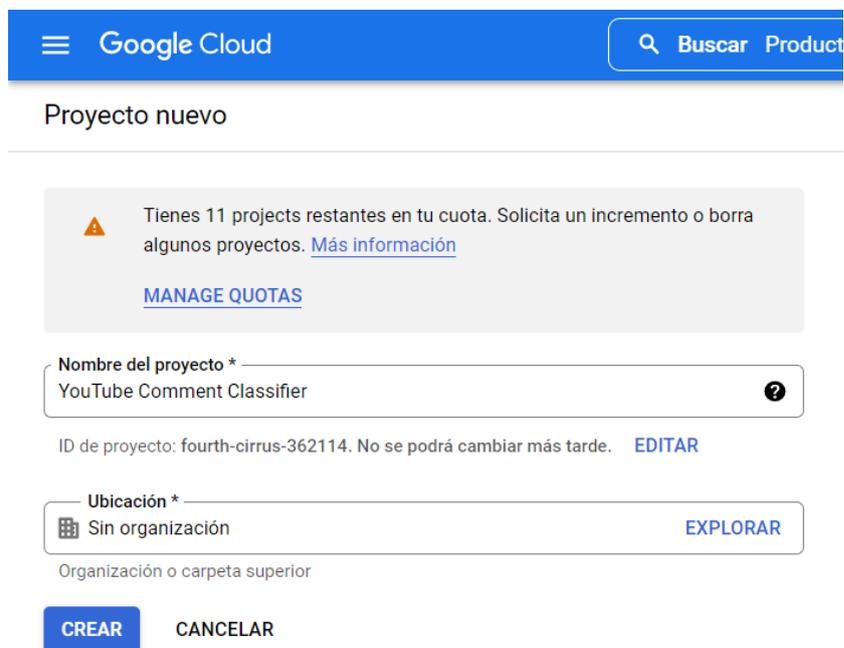
Figura 3. Pantalla de proyectos en la consola de Google Cloud.



Fuente: Elaboración propia.

Se asignará un nombre cualquiera y al gusto del usuario para el proyecto, para este caso ha sido “YouTube Comment Classifier”.

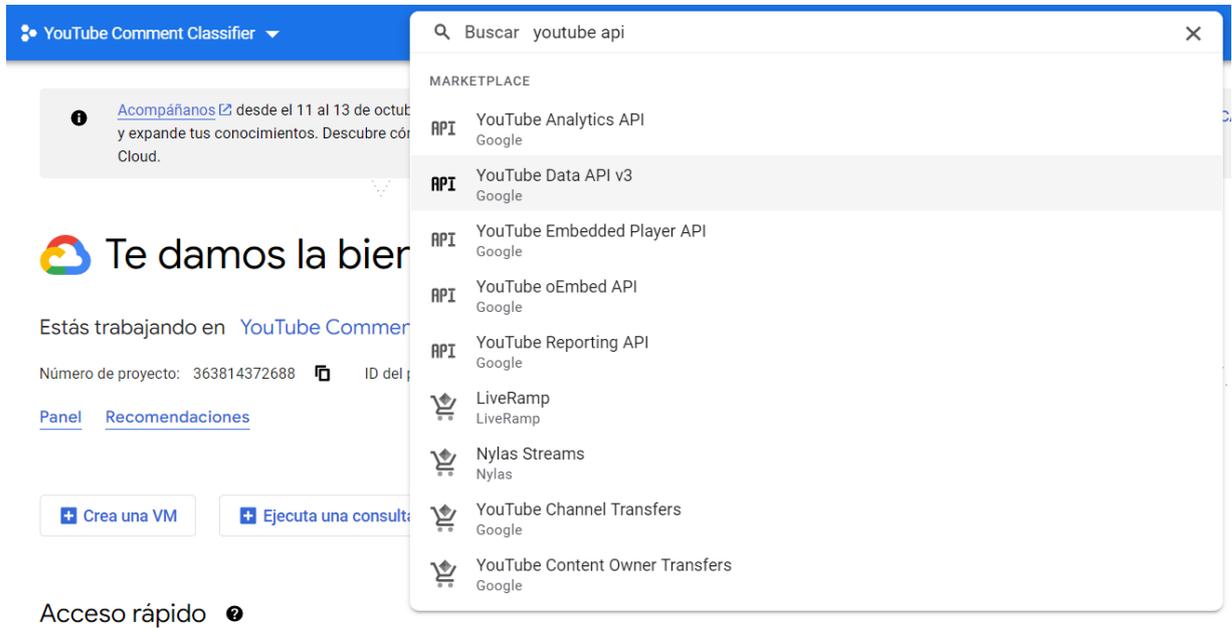
Figura 4. Creación de un proyecto en la consola de Google Cloud.



Fuente: Elaboración propia.

Con el proyecto ya creado, en la barra de búsqueda se escribirán las palabras YouTube api se seleccionará la opción de YouTube Data API v3:

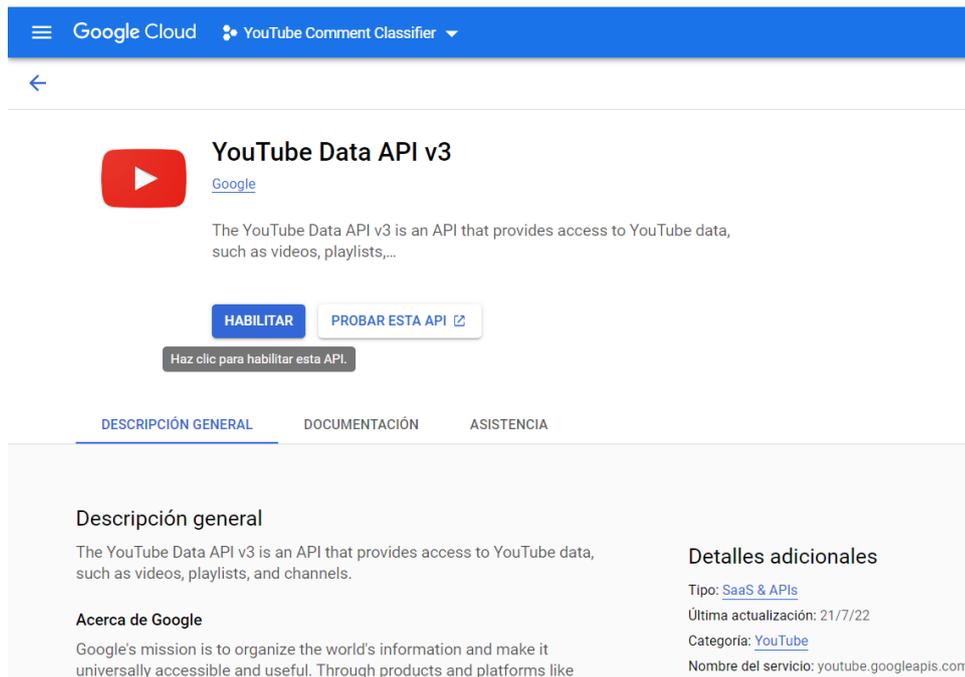
Figura 5. Búsqueda de YouTube Data API v3 en la consola de Google Cloud.



Fuente: Elaboración propia.

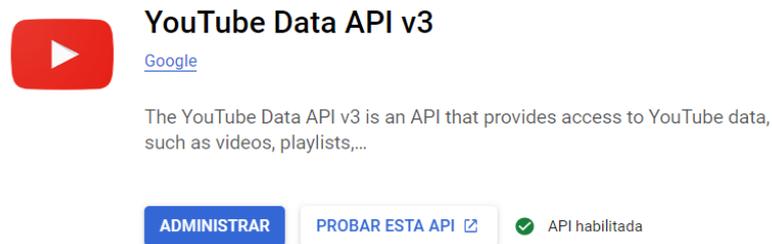
Se habilitará el API, tardará unos segundos para que después aparezca el botón ADMINISTRAR el cual nos llevará a configurar las credenciales del servicio.

Figura 6. Activación de YouTube Data API v3 en la consola de Google Cloud.



Fuente: Elaboración propia.

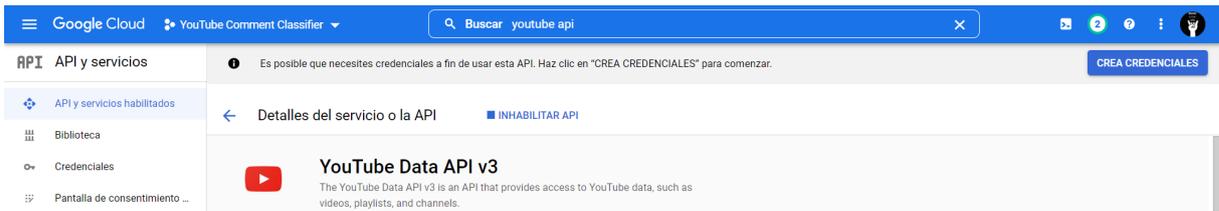
Figura 7. Activación de YouTube Data API v3 en la consola de Google Cloud.



Fuente: Elaboración propia.

En la siguiente pantalla se podrán crear las credenciales y pantalla de consentimiento OAuth, para ello se da clic a CREAR CREDENCIALES. Por defecto estará seleccionada la opción de YouTube Data API v3, así que solo será necesario marcar la opción datos del usuario.

Figura 8. Creación de credenciales para el YouTube Data API v3.



Fuente: Elaboración propia.

Figura 9. Creación de credenciales en la consola de Google Cloud.

1 Tipo de credencial

¿Qué API estás usando?

Las diferentes API usan plataformas Auth distintas, y algunas credenciales pueden llamar solo a determinadas API.

Seleccionar una API *
YouTube Data API v3

¿A qué datos quieres acceder? *

Se requieren credenciales distintas para autorizar el acceso según el tipo de datos que solicites. [Más información](#)

Datos del usuario ?
Son datos que pertenecen a un usuario de Google, como su dirección de correo electrónico o edad. El consentimiento del usuario es obligatorio. Esto creará un cliente de OAuth.

Datos públicos
Son datos de Google disponibles de manera pública, como datos públicos de Maps que muestran información sobre un restaurante. Esto creará una clave de API.

SIGUIENTE

Fuente: Elaboración propia.

El nombre que se le quiera dar a la aplicación quedará a selección del usuario, así como también el correo electrónico de asistencia del usuario, que por lo general será el mismo de la cuenta Gmail.

Figura 10. Configuración de la pantalla de consentimiento de OAuth.

✓ Tipo de credencial

2 Pantalla de consentimiento de OAuth

Información de la aplicación

Esta información aparece en la pantalla de consentimiento y permite que los usuarios finales sepan quién eres y cómo comunicarse contigo

Nombre de la aplicación *
Comment Classifier TFM
El nombre de la aplicación que solicita el consentimiento

Correo electrónico de asistencia del usuario *
demonhack.7@gmail.com
Para que los usuarios se comuniquen contigo si tienen preguntas sobre su consentimiento

Fuente: Elaboración propia.

La información de contacto del desarrollador puede ser el mismo que se ha seleccionado anteriormente, posterior a ello, se guardará y continuará al apartado de permisos donde se agregará el siguiente permiso: `.../auth/youtube.force-ssl` el cual permitirá ver, editar y borrar los comentarios y *sub-comentarios*.

Figura 11. Configuración de permisos para el YouTube Data API.

Información de contacto del desarrollador

Direcciones de correo electrónico *
demonhack.7@gmail.com ✕

Google enviará notificaciones sobre cualquier cambio en tu proyecto a estas direcciones de correo electrónico.

GUARDAR Y CONTINUAR

Fuente: Elaboración propia.

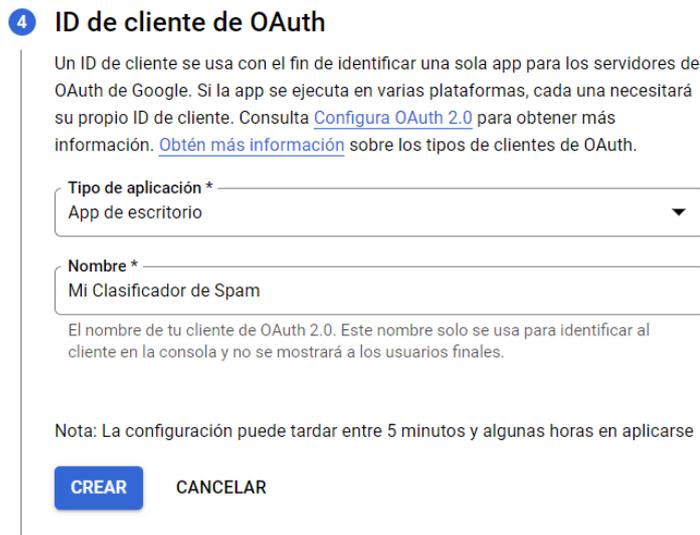
Figura 12. *Parámetros requeridos para la configuración de permisos.*



Fuente: Elaboración propia.

En este apartado se seleccionará el tipo de aplicación de escritorio y un nombre cualquiera.

Figura 13. *Configuración del ID de cliente de OAuth para el YouTube Data API v3.*



Fuente: Elaboración propia.

Al completar todos los pasos anteriores se generará el ID de cliente, este se usará para que la aplicación a desarrollar se conecte.

Figura 14. Obtención y descarga de credenciales para el YouTube Data API v3.

5 Tus credenciales

Descarga tus credenciales

Download this credential information in JSON format. Siempre estará disponible en la [página de credenciales](#).

ID de cliente

↓ DESCARGAR

Your OAuth consent screen has been configured and your app has a publishing status of "Testing". This means it can be used by a limited set of test users, which you can set up on the [OAuth consent screen page](#).

If you want to make your app generally available to all users, you will need to set the publishing status to "In production". You will also need to [submit your app](#) for verification, which can take up to 4-6 weeks (depending on which OAuth scopes your app uses).

LISTO

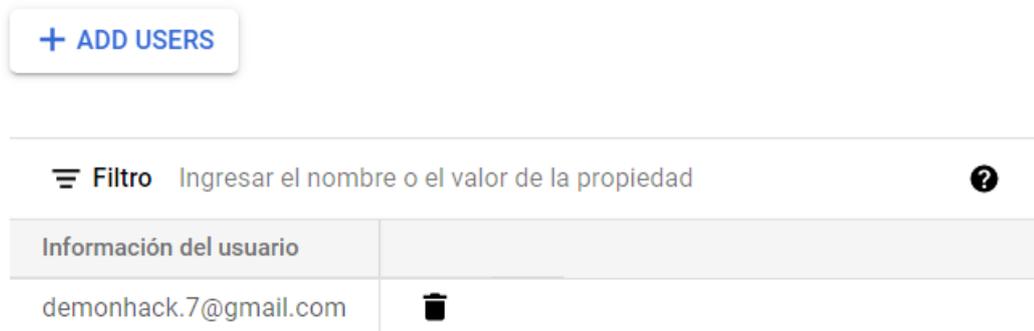
CANCELAR

Fuente: Elaboración propia.

Para culminar con la configuración, se ingresará al apartado de pantalla de consentimiento para adicionar usuarios de prueba que puedan conectarse y trabajar con la YouTube Data API v3 y realizar peticiones con la misma; solo estos usuarios tienen el permiso de hacer uso del API. En este caso se ha seleccionado el mismo usuario de Gmail durante todo el proceso.

Figura 15. Obtención y descarga de credenciales para el YouTube Data API v3.

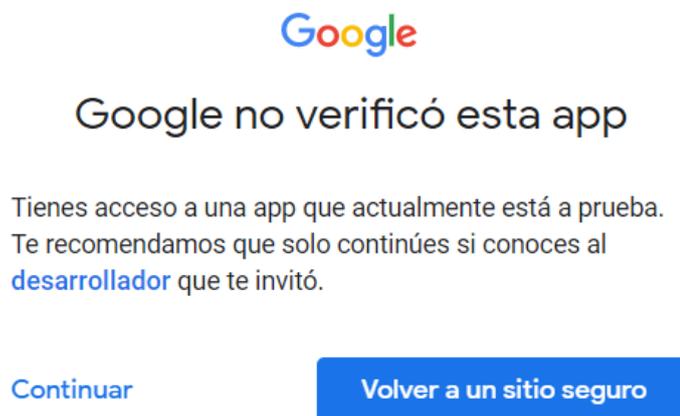
Usuarios de prueba



Fuente: Elaboración propia.

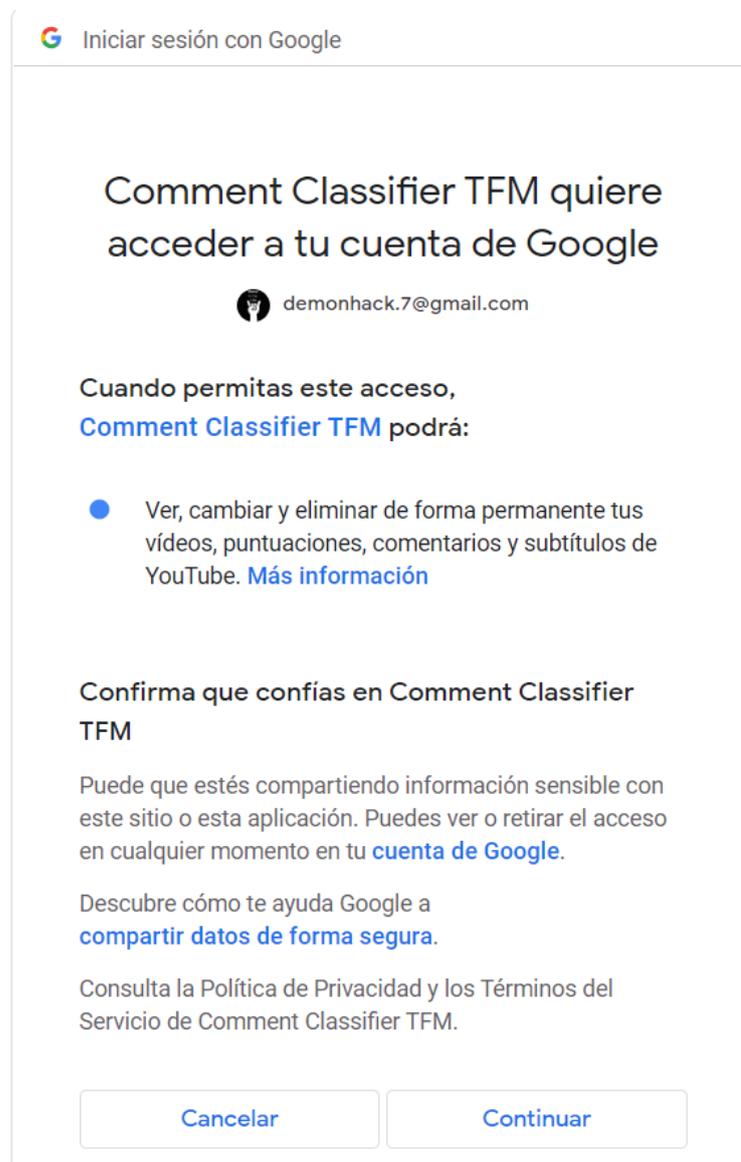
Al momento de querer utilizar las credenciales ya obtenidas para el YouTube Data API v3, aparecerá la siguiente pantalla, en la que el usuario conectará su cuenta de Gmail al proyecto creado, en este caso es el proyecto “Comment Classifier TFM”.

Figura 16. Pantalla de consentimiento al usar las credenciales de YouTube Data API v3.



Fuente: Elaboración propia.

Figura 17. Pantalla de consentimiento con un usuario de prueba.



Fuente: Elaboración propia.

4.2. Recolección de los comentarios de YouTube con YouTube Data API v3

Al disponer de las credenciales del API de YouTube, se las ubicará en el mismo directorio donde se encuentre el script que recopilará todos los comentarios de un video de YouTube especificado. El siguiente script obtendrá los comentarios de 10 videos de YouTube previamente escogidos, los mismos se almacenarán en un archivo CSV.

Figura 18. *Scraper de comentarios de YouTube.*

```
import collections, os, pickle
import pandas as pd
from googleapiclient.errors import HttpError
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.oauth2.credentials import Credentials
from google.auth.transport.requests import Request
from google.auth.transport.requests import Request
videoID =
["OUxIsokxA_Y", "x6YUdao13ZI", "3SLVcUg9SH0", "hw4cUiYH5pA", "nsMzIF6MAEU", "4dwO
03667s0",
    "sEtj34VMClU", "Hwybp38GnZw", "fAmdC64KKdM", "lNonGKWXi_Y"]
credentials = None

#token.pickle ALMACENARÁ LA SESIÓN CON LAS CREDENCIALES DEL API DE YT
if os.path.exists('token.pickle'):
    print('Loading Credentials From File...')
    with open('token.pickle', 'rb') as token:
        credentials = pickle.load(token)

# SI EL TOKEN NO ES VALIDO, RE-LOGEAR
if not credentials or not credentials.valid:
    if credentials and credentials.expired and credentials.refresh_token:
        print('Refreshing Access Token...')
        credentials.refresh(Request())
    else:
        print('Fetching New Tokens...')
        flow =
InstalledAppFlow.from_client_secrets_file('client_secrets.json',
scopes=['https://www.googleapis.com/auth/youtube.force-ssl'])
        flow.run_local_server(port=0, prompt='consent',
authorization_prompt_message='')
        credentials = flow.credentials
        #Save the credentials for the next run
        with open('token.pickle', 'wb') as f:
            print('Saving Credentials for Future Use...')
            pickle.dump(credentials, f)

# creating youtube resource object
youtube = build('youtube', 'v3', credentials = credentials)

def get_comment_thread(pageToken, strVideoID):
    request = youtube.commentThreads().list(part = "snippet,replies",
videoId = strVideoID, maxResults = 100, pageToken = pageToken,
order="orderUnspecified")#top comments.
```

```

response = request.execute()
token = response.get("nextPageToken", "End")
return response, token

def load_comments(match):
    for item in match["items"]:
        comment = item["snippet"]["topLevelComment"]
        author = comment["snippet"]["authorDisplayName"]
        text = comment["snippet"]["textDisplay"]
        plaintext.append("Comment by {}: {}".format(author, text))

        if 'replies' in item.keys():
            for reply in item['replies']['comments']:
                rauthor = reply['snippet']['authorDisplayName']
                rtext = reply["snippet"]["textDisplay"]
                plaintext.append("\tReply by {}: {}".format(rauthor, rtext))
                #print("\n\tReply by {}: {}".format(rauthor, rtext), "\n")

for x in range(0, len(videoID)):
    plaintext = []
    comments = []
    currentPageToken = None
    for i in range(0, 100):
        match = get_comment_thread(currentPageToken, videoID[x])
        currentPageToken = match[1]
        if currentPageToken != 'End':
            load_comments(match[0])
        else:
            break

    print('Total Comments: {}'.format(len(plaintext)))
    spam = []
    #spam = [item for item, count in collections.Counter(plaintext).items()
    if count > 1]
    ham = [item for item, count in collections.Counter(plaintext).items() if
count == 1]

    for item, count in collections.Counter(plaintext).items():
        if count > 1:
            print('[{}] {}'.format(count, item))
            spam.append(item)

    dataframe = pd.DataFrame(ham)
    dataframe.to_csv('{}_okay.csv'.format(videoID[x]), mode = 'w', header =
False)

    dataframe = pd.DataFrame(spam)

```

```
dataframe.to_csv('{}_spam.csv'.format(videoID[x]),
mode='w',header=False)
```

Fuente: Elaboración propia.

Posterior a ello, todos los comentarios fueron clasificados manualmente usando Microsoft Excel, considerando los criterios del apartado 3.2.1, los comentarios spam fueron marcados con un “1” y los demás comentarios fueron marcados con “0”. El total de los comentarios con su respectivo video se puede resumir en la siguiente tabla:

Tabla 1. Resumen de la clasificación manual de comentarios en los videos seleccionados (15/08/2022).

Título del video y link	Comentarios apropiados	Comentarios spam	%
Cómo Minar Criptomonedas Con Cualquier Ordenador https://www.youtube.com/watch?v=3SLVcUg9SH0	4062	756	15. 7
6 claves para el uso inteligente de tarjetas de crédito ¡Que no te controlen! https://www.youtube.com/watch?v=4dwO03667s0	1835	4	0.2
How To Buy Bitcoin SAFELY Step By Step Guide https://www.youtube.com/watch?v=fAmdC64KKdM	4761	1444	23. 2
El Metaverso te Hará Millonario con Estas 5 Criptomonedas – CryptoYoda – https://www.youtube.com/watch?v=hW4cUiYH5pA	1477	149	9.2
I Built Willy Wonka's Chocolate Factory! https://www.youtube.com/watch?v=Hwybp38GnZw	145318	24400	14. 4
Giving Someone \$1000, Then Running Away	19522	673	3.3

https://www.youtube.com/watch?v=INonGKWxi_Y			
17 ideas de ingresos pasivos altamente rentables - 2021 https://www.youtube.com/watch?v=nsMzIF6MAEU	371	8	2.1
Les Doy \$1,000,000 con Solo 1 Minuto para Gastarlo https://www.youtube.com/watch?v=OUxIsokxA_Y	41996	458	1.1
How to Buy Cryptocurrency for Beginners 📖🤖 (#1 Ultimate Guide 2022!) 👑 Step-by-Step (Updated!) 🚀 https://www.youtube.com/watch?v=sEtj34VMCIU	3799	524	12.1
Cómo He Creado 14 Fuentes De Ingresos Con 23 años https://www.youtube.com/watch?v=x6YUdao13ZI	3164	388	10.9

Fuente: Elaboración propia.

En total se encontraron 255,109 comentarios de los cuales 28,804 fueron comentarios spam identificados manualmente. Los archivos CSV clasificados servirán para entrenar el modelo de *machine learning* por Naïve Bayes; para lograrlo se identificaron 2 parámetros en cada archivo, el primer parámetro es el contenido que incluye el comentario en texto plano y también el nombre de usuario, el segundo parámetro es la clasificación de spam con un “1” o “0”.

4.3. Entrenamiento de un modelo por Naïve Bayes usando Scikit-Learn.

Se usó el lenguaje de programación Python y las librerías de Scikit-Learn que facilita el significativamente el entrenamiento de modelos Naïve Bayes (Scikit-Learn, 2022) a partir todos los archivos CSV de comentarios spam recopilados previamente.

Figura 19. Código fuente para entrenar un modelo predictivo por Naïve Bayes.

```
import collections, os, pickle
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.Naïve_bayes import MultinomialNB
```

```
from sklearn.metrics import confusion_matrix, classification_report

data_okay1 = pd.read_csv("3SLVcUg9SH0_okay.csv")
data_okay2 = pd.read_csv("4dw003667s0_okay.csv")
data_okay3 = pd.read_csv("fAmdC64KKdM_okay.csv")
data_okay4 = pd.read_csv("hW4cUiYH5pA_okay.csv")
data_okay5 = pd.read_csv("Hwybp38GnZw_okay.csv")
data_okay6 = pd.read_csv("lNonGKWXi_Y_okay.csv")
data_okay7 = pd.read_csv("nsMzIF6MAEU_okay.csv")
data_okay8 = pd.read_csv("OUxIsokxA_Y_okay.csv")
data_okay9 = pd.read_csv("sEtj34VMClU_okay.csv")
data_okay10 = pd.read_csv("x6YUdao13ZI_okay.csv")

data_spam1 = pd.read_csv("3SLVcUg9SH0_spam.csv")
data_spam2 = pd.read_csv("4dw003667s0_spam.csv")
data_spam3 = pd.read_csv("fAmdC64KKdM_spam.csv")
data_spam4 = pd.read_csv("hW4cUiYH5pA_spam.csv")
data_spam5 = pd.read_csv("Hwybp38GnZw_spam.csv")
data_spam6 = pd.read_csv("lNonGKWXi_Y_spam.csv")
data_spam7 = pd.read_csv("nsMzIF6MAEU_spam.csv")
data_spam8 = pd.read_csv("OUxIsokxA_Y_spam.csv")
data_spam9 = pd.read_csv("sEtj34VMClU_spam.csv")
data_spam10 = pd.read_csv("x6YUdao13ZI_spam.csv")

dataset = pd.concat([data_okay1, data_okay2, data_okay3,
data_spam1, data_spam2, data_spam3, data_spam4, data_spam5, data_spam6,
data_spam7, data_spam8, data_spam9, data_spam10])
dataset.drop(["NUMBER"], axis=1, inplace=True)
dataset.info()

X_train, X_test, y_train, y_test = train_test_split(dataset["CONTENT"],
dataset["CLASS"])

tfidf_vect = TfidfVectorizer(use_idf=True, lowercase=True)
X_train_tfidf = tfidf_vect.fit_transform(X_train)
X_train_tfidf.shape

model = MultinomialNB()
model.fit(X_train_tfidf, y_train)

X_test_tfidf = tfidf_vect.transform(X_test)
predictions = model.predict(X_test_tfidf)

confusion_matrix(y_test, predictions)

print(classification_report(y_test, predictions))

print(model.score(X_test_tfidf, y_test))
```

```
with open("model.pkl", "wb") as model_file:  
    pickle.dump(model, model_file)  
  
with open("tfidf-vect.pkl", "wb") as tfidf_vect_file:  
    pickle.dump(tfidf_vect, tfidf_vect_file)
```

Fuente: Elaboración propia.

Figura 20. Resultados después de entrenar el modelo por Naïve Bayes.

```
PS G:\Mi unidad\TFM\Datasets> & C:/Users/Lenovo/AppData/Local/Microsoft/WindowsApps/python3.10.exe "g:/Mi unidad/TFM/Datasets/TFM_Trainer.py"  
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 259534 entries, 0 to 21034  
memory usage: 5.9+ MB  
      precision    recall  f1-score   support  
  
    0         0.97     1.00     0.99     57655  
    1         0.99     0.80     0.88     7229  
  
 accuracy                0.98     64884  
 macro avg              0.98     0.90     0.93     64884  
weighted avg              0.98     0.98     0.98     64884  
  
0.9763732199001295
```

Fuente: Elaboración propia.

Con el modelo obtenido ya se puede usarlo para identificar si un texto es o no considerado spam, para ello se volvió a utilizar el código de la figura 3, pero con otros videos diferentes a los seleccionados para que finalmente, el mismo modelo clasifique los comentarios en un archivo CSV sin necesidad de hacerlo de manera manual como se realizó en un principio.

Para corroborar la efectividad del modelo entrenado, se utilizó el script de la figura 6 para encontrar todos los comentarios spam en todos los videos con los que se entrenó el modelo.

Figura 21. Código de la figura 3 adaptado para clasificar los mensajes como spam.

```
###... Después de obtener todos comentarios de los videos de YouTube

def loadModel():
    if ("loaded_model" not in LOCAL_CACHE or "tfidf_vectorizer" not in
LOCAL_CACHE):
        with open("model.pkl", "rb") as model_file:
            LOCAL_CACHE["loaded_model"] = pickle.load(model_file)

        with open("tfidf-vect.pkl", "rb") as tfidf_vect_file:
            LOCAL_CACHE["tfidf_vectorizer"] = pickle.load(tfidf_vect_file)

loadModel()
for x in range(0, len(videoID)):
    bannedUsers = []
    plaintext = []
    comments = []
    currentPageToken = None
    try:
        for i in range(0, 7777777):
            match = get_comment_thread(currentPageToken, videoID[x])
            currentPageToken = match[1]
            if currentPageToken != 'End':
                load_comments(match[0])
            else:
                break
    except Exception as e:
        print(e)
        break

print('Total Comments: {}'.format(len(plaintext)))
commentLIST = ['CONTENT']
classLIST = ['CLASS']
idLIST = ['ID']

for item in plaintext:
    item = str(item).split('[[[SEP]]]')
    comment = [item[0].lower()]
    comments_df = pd.DataFrame(data=comment, columns=["CONTENT"])
    comments_tfidf =
LOCAL_CACHE["tfidf_vectorizer"].transform(comments_df["CONTENT"])
prediction =
LOCAL_CACHE["loaded_model"].predict_proba(comments_tfidf)
prediction = prediction[0][0]
commentLIST.append(item[0])
idLIST.append(item[1])
if prediction > 0.7:
```

```

        isSpam = 1
    else:
        isSpam = 0
    user = item[0].split(': ')[0]
    if isSpam == 1:
        bannedUsers.append(user)
    classLIST.append(isSpam)

finalData = {'CONTENT': commentLIST, 'CLASS': classLIST, 'ID': idLIST}
dataframe = pd.DataFrame(finalData, columns = ['CONTENT', 'CLASS',
'ID'])
dataframe.to_csv('PREDICTED.csv', mode = 'w', header = False)

print('TOTAL SPAM FOUND: {}'.format(classLIST.count(1)))

```

Fuente: Elaboración propia.

El script descargará todos los comentarios de un video seleccionado y posteriormente analizará cada comentario individualmente para que finalmente se cree un archivo CSV con los comentarios spam identificados. Como resultado, se obtuvo una cantidad menor de comentarios spam comparado con la serie de datos inicial. Por ende, se optó por añadir al código de clasificación de spam, un apartado que buscara si un usuario había realizado spam, considere que todos sus comentarios serian spam; esto aumento significativamente la cantidad de comentarios spam identificados, pero aun así siendo menor a la serie de datos con la que se había entrenado el modelo.

Figura 22. *Búsqueda de autores que han sido previamente marcados como spam.*

```

bannedUsers = list(set(bannedUsers))
for i in range(1, len(commentLIST)):
    user = commentLIST[i].split(': ')[0]
    isSpam = int(classLIST[i])

    if user in bannedUsers:
        classLIST[i] = 1
        print(user)

```

Fuente: Elaboración propia.

Por todos los resultados obtenidos, se decidió entrenar al modelo de una manera distinta; separando los comentarios y los nombres de usuarios ya que, gran parte del mensaje que

intenta desviar al usuario hacia otras plataformas se encuentra en dentro del nombre de usuario.

Para ello, la serie de datos sería distinta para los usuarios, ya que no todos los comentarios llevaran el contenido spam en su nombre de usuario, por ello se clasificó manualmente todos los nombres de usuario que sean considerados spam, solamente si este tuviera palabras claves como un número de teléfono para contactar en WhatsApp, Telegram o un nombre de usuario que pueda ser buscando en Instagram o Facebook. En el anexo A se puede apreciar una basta lista de nombres de usuario considerados como spam y que su vez, fue usada para el entrenamiento de los modelos predictivos.

Figura 23. Ejemplo de comentarios spam en los que su nombre de usuario contendría el spam.



Fuente: Elaboración propia.

Figura 24. Ejemplo de comentarios spam en los que su nombre de usuario sería genérico.



Fuente: Elaboración propia.

Los nombres de usuarios sin este tipo de patrones no fueron marcados como spam, inclusive y aunque su comentario si haya sido señalado como spam; de igual manera se volvió a clasificar los comentarios para así poder entrenar nuevamente el modelo y por separado. Con la nueva serie de datos consolidada, nuevamente se ejecuta el código de la Figura 4; adaptándolo antes para generar un modelo Naïve Bayes Multinomial por nombres de usuario y aparte, el de comentarios, inmediatamente se obtienen resultados más concisos y más precisos al anterior modelo.

Figura 25. Entrenamiento Naïve Bayes (por nombres de usuario y por comentarios).

```

memory usage: 1.5+ MB
      precision    recall  f1-score   support

   0       0.99      1.00      0.99     14916
   1       0.98      0.82      0.90      1117

 accuracy          0.99     16033
 macro avg         0.98      0.91      0.94     16033
weighted avg         0.99      0.99      0.99     16033

0.9866525291586103
      precision    recall  f1-score   support

   0       1.00      1.00      1.00     11504
   1       1.00      0.20      0.33         60

 accuracy          1.00     11564
 macro avg         1.00      0.60      0.67     11564
weighted avg         1.00      1.00      0.99     11564

0.9958491871324802
    
```

Fuente: Elaboración propia.

Posterior a tener los dos modelos entrenados, se realizaron nuevamente las pruebas con los videos seleccionados anteriormente; además, en esta ocasión se ha utilizado el programa antispam de código abierto para tener una mejor referencia de comparación, YT-Spammer-Purge clasifica los comentarios usando una gran variedad de filtros sin emplear un modelo de predicción (ThioJoe, 2022), caso contrario al desarrollado en el presente trabajo. A continuación, se detallan los resultados:

Tabla 2. Comparación entre modelos Naïve Bayes vs YT-Spammer-Purge (15/08/2022).

Título del video y link	Un Modelo	Dos Modelos	YT-Spammer-Purge
3 Apps Que Pagan Dinero Real En PayPal https://www.youtube.com/watch?v=SjnkUf5aBdE	361	549	580
¿Cómo invertir tus primeros mil dólares en 2022? Clips https://www.youtube.com/watch?v=dvecqwfU6xw	160	220	187
Cómo Invertir Tus Primeros 1000 en 2022 - Paso a Paso (Curso gratuito) https://www.youtube.com/watch?v=3ybL7JLBqIA	184	248	179
¡Como hacer dinero lo más rápido posible con tu negocio! EP. #39 NMP https://www.youtube.com/watch?v=7wSFg0TZXfs	47	65	65
Cómo Invertir en Criptomonedas para Principiantes (2022) https://www.youtube.com/watch?v=pM53BJNGoKw	53	81	91

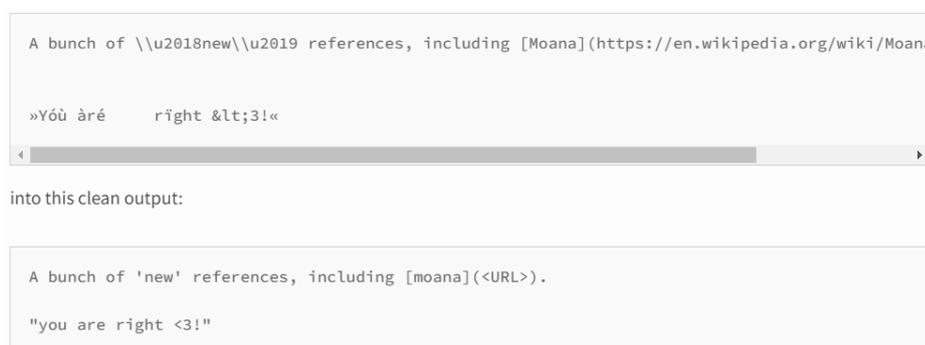
Fuente: Elaboración propia.

En la comparación se utilizó el primer modelo generado en el transcurso del proyecto, el cual analiza el comentario y el nombre de usuario del autor a la vez, también se comparó con los resultados obtenidos por los dos modelos entrenados posteriormente, uno de ellos analiza independientemente los nombres de usuario y el otro únicamente los comentarios, si uno de los dos es identificado como spam, todo el comentario se considerará spam. Por último, los mismos videos fueron usados para escanear y clasificar sus comentarios con la herramienta YT-Spammer-Purge.

Al momento de revisar los resultados se llegó a la conclusión de que existían muchos falsos negativos y algunos verdaderos negativos que escapaban de ser clasificados como spam por el modelo. A pesar de reentrenar el modelo con esta nueva información clasificando los comentarios manualmente como spam, se obtenían los mismos resultados, por lo cual, se dio por hecho que los ciberdelincuentes buscaban diferentes técnicas para la evasión de filtros con tipo de programas clasificadores de spam; motivo por el cual el modelo entrenado no reconocía correctamente a los comentarios spam de los *ham*.

Por aquella razón se buscaron diferentes librerías de Python que permitieran tratar los comentarios, para que se realizará una limpieza del texto antes de clasificar a un comentario como spam. La librería escogida ha sido *clean-text*, que tal y como lo explica en su página de repositorio, normaliza el texto para poder ser más manejable y darle diferentes usos (Filter, 2022).

Figura 26. Ejemplo de un antes y después de usar *clean-text* sobre un comentario cualquiera.



Fuente: (Filter, 2022).

Se procedió a escribir un código que tome el *dataset* ya existente para limpiar tanto los comentarios como nombre de usuarios y así poder entrenar nuevamente el modelo por Naïve

Bayes a partir de una base de datos limpia y más comprensible. Con la limpieza, se decidió eliminar emojis, puntuación, separación de líneas y convertir todo el texto en minúsculas.

Figura 27. Código utilizado para limpiar el dataset previamente clasificado.

```
import pandas as pd
from cleantext import clean

file_name = "Train_Me_Dual.csv"
dataset_comments = pd.read_csv(file_name, encoding='latin1')
classLIST = dataset_comments['CLASS']
commentLIST = []
usernameLIST = dataset_comments['AUTHOR']

for comment in dataset_comments['COMMENT']:
    commentLIST.append(clean(comment, no_emoji=True, lower=True,
no_line_breaks=True, no_punct=True))

dataset = pd.DataFrame({'AUTHOR': usernameLIST, 'COMMENT': commentLIST,
'CLASS': classLIST}, columns = ['AUTHOR', 'COMMENT', 'CLASS'])
dataset.to_csv('Train_Me_Dual_Cleaned.csv', mode = 'w', header = True)
dataset.info()

file_name = "Usernames.csv"
dataset_usernames = pd.read_csv(file_name, encoding='latin1')
classLIST = dataset_usernames['CLASS']
usernameLIST = []

for username in dataset_usernames['AUTHOR']:
    usernameLIST.append(clean(username, no_emoji=True, lower=True,
no_punct=True))

usernames = pd.DataFrame({'AUTHOR': usernameLIST, 'CLASS': classLIST},
columns=['AUTHOR', 'CLASS'])
usernames.to_csv('Usernames_Cleaned.csv', mode = 'w', header = True)
dataset.info()
```

Fuente: Elaboración propia.

Después de ello se obtuvo un *dataset* mucho más comprensible tanto para el ser humano como para clasificarlos con el modelo Naïve Bayes. A continuación, una breve comparación entre el antes y después:

Figura 28. Dataset antes de realizarse la limpieza a los nombres de usuario.

```
19223,Chiquito AM,0
6850,chorizo,0
25935,Cisco 🚀 +12342198244,0
4362,Clan Uchiha,0
19691,Clip Hunter,0
11619,Cosme fulanito,0
24723,Whatsapp Me +14435307288,1
20808,Deise Shimberg,0
6987,DIO BRANDO,0
37541,DM Reginaflores3031 on Telegram,1
5701,Donovan Issac Esparza Ruvalcaba,0
23967, 🇺🇸 +19852479337whatsapp,1
25779, 🇺🇸 496_vincent 🇺🇸 on INSTAGRAM,1
39951, 🇺🇸 BURUHACK EN INSTAGRAM,1
24696, 🇺🇸 🇺🇸 smartbitcoininvestment on telegram,1
223, 🇺🇸 Whatsap +17743580428,1
537, 🇺🇸 Whatsap +447468254752,1
549, 🇺🇸 WHATSAP +447468254752,1
442, 🇺🇸 WHATSAPP +17743580428,1
11990, 🇺🇸 ROBUX gratis en mi perfil 🤩,1
9469, 🇺🇸 whatsapp+15183078298,1
262, 🇺🇸 WHATSAPP +19739356983,1
43176, 🇺🇸 whatsapp +13037108086,1
329, 🇺🇸 #+17139780302 Helpline,1
```

Fuente: Elaboración propia.

Figura 29. Dataset después de realizarse la limpieza a los nombres de usuario con clean-text.

```
42, chiquito am,0
43, chorizo,0
44, cisco+12342198244,0
45, clan uchiha,0
46, clip hunter,0
47, cosme fulanito,0
48, whatsapp me+14435307288,1
49, deise shimberg,0
50, dio brando,0
51, dm reginaflores3031 on telegram,1
52, donovan issac esparza ruvalcaba,0
53, +19852479337whatsapp,1
54, 496vincenton instagram,1
55, buruhack en instagram,1
56, smartbitcoininvestment on telegram,1
57, whatsap10 17743580428,1
58, whatsap10 447468254752,1
59, whatsap10 447468254752,1
60, whatsapp +17743580428,1
61, robux gratidz en mi pergil,1
62, whatsapp+15183078298,1
63, whatsapp+19739356983,1
64, whatsapp13037108086,1
65, +17139780302 helpline,1
```

Fuente: Elaboración propia.

Figura 30. Dataset antes de realizarse la limpieza de comentarios.

```
Gracias por mirar
de inmediato.
nuestras selecciones de monedas ALT, para obtener más información sobre XRP, ADA, ETH, inversiones, BTC y más".
Gracias por mirar
de inmediato.
nuestras selecciones de monedas ALT, para obtener más información sobre XRP, ADA, ETH, inversiones, BTC y más".
Gracias por mirar
de inmediato.
nuestras selecciones de monedas ALT, para obtener más información sobre XRP, ADA, ETH, inversiones, BTC y más".
Gracias por mirar
de inmediato.
nuestras selecciones de monedas ALT, para obtener más información sobre XRP, ADA, ETH, inversiones, BTC y más".
cias por mirar para obtener más información y consejos 👍👍,1
as por mirar para obtener más información y consejos 👍,1
cias por mirar para obtener más información y consejos 👍,1
cias por mirar el mensaje de texto 📄 para obtener más información y consejos 👍👍,1
gracias por mirar 👍 y dejar un comentario 👍 y 📄 DM en Telegram para reclamar el premio,1
gracias por mirar 👍 y dejar un comentario 👍 y 📄 DM en Telegram para reclamar el premio,1
,"Gracias por los comentarios, por más información, 👍. 👍 ideas y estrategias rentables de inversión en criptos
,"Gracias por los comentarios, por más información, 👍. 👍 ideas y estrategias rentables de inversión en criptos
gracias por los comentarios, asegúrate de escribirme en la línea de arriba 📄📄📄",1
gracias por los comentarios, asegúrate de escribirme en la línea de arriba 📄📄📄",1
ias por la retroalimentación. Para obtener más información y asistencia, póngase en contacto conmigo en el número
ias por la retroalimentación. Para obtener más información y asistencia, póngase en contacto conmigo en el número
ias por la retroalimentación. Para obtener más información y asistencia, póngase en contacto conmigo en el número
ias por la retroalimentación. Para obtener más información y asistencia, póngase en contacto conmigo en el número
ias por la retroalimentación. Para obtener más información y asistencia, póngase en contacto conmigo en el número
```

Fuente: Elaboración propia.

Figura 31. Dataset después de realizarse la limpieza de comentarios con clean-text.

```
gracias por mirar espero que esto sea util comuniquese con el numero de arriba para obtener
gracias por mirar espero que esto sea util comuniquese con el numero de arriba para obtener
gracias por mirar espero que esto sea util comuniquese con el numero de arriba para obtener
gracias por mirar comuniquese con el numero de arriba tengo una oferta muy rentable para uste
gracias por mirar comuniquese con el numero de arriba tengo una oferta muy rentable para uste
gracias por mirar comuniquese con el numero de arriba tengo una oferta muy rentable para uste
ias por mirar r envia un mensaje directo de inmediato obtenga acceso exclusivo a nuestras sel
ias por mirar r envia un mensaje directo de inmediato obtenga acceso exclusivo a nuestras sel
ias por mirar r envia un mensaje directo de inmediato obtenga acceso exclusivo a nuestras sel
ias por mirar r envia un mensaje directo de inmediato obtenga acceso exclusivo a nuestras sel
acias por mirar para obtener mas informacion y consejos,1
acias por mirar para obtener mas informacion y consejos,1
acias por mirar para obtener mas informacion y consejos,1
acias por mirar el mensaje de texto para obtener mas informacion y consejos,1
dejar un comentario y dm en telegram para reclamar el premio,1
dejar un comentario y dm en telegram para reclamar el premio,1
acias por los comentarios por mas informacion ideas y estrategias rentables de inversion en c
acias por los comentarios por mas informacion ideas y estrategias rentables de inversion en c
gracias por los comentarios asegurate de escribirme en la linea de arriba,1
gracias por los comentarios asegurate de escribirme en la linea de arriba,1
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
acias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
ias por la retroalimentacion para obtener mas informacion y asistencia pongase en contacto co
```

Fuente: Elaboración propia.

Con los datos ahora limpios se realizará el entrenamiento del modelo para obtener los siguientes resultados:

Figura 32. Entrenamiento de dos modelos por Naïve Bayes (empleando un dataset limpio).

```
precision    recall  f1-score   support

0           0.99     1.00     0.99     15021
1           0.95     0.87     0.91     1012

accuracy          0.99     16033
macro avg         0.97     0.93     0.95     16033
weighted avg      0.99     0.99     0.99     16033

0.9887107840079835
precision    recall  f1-score   support

0           0.99     1.00     1.00     11557
1           1.00     0.58     0.74      171

accuracy          0.99     11728
macro avg         1.00     0.79     0.87     11728
weighted avg      0.99     0.99     0.99     11728

0.9939461118690314
```

Fuente: Elaboración propia.

Por consiguiente, se volvieron a escanear y clasificar los comentarios de los cinco videos escogidos previamente de modo que se pueda verificar si su clasificación de comentarios spam ha mejorado con el modelo entrenado con un *dataset* sin limpieza, así como también calcular la cantidad de falsos negativos y compararlo con el programa YT-Spammer-Purge.

La única modificación al código de la Figura 21 fue, realizar una limpieza a cada uno de los nombres de usuario y comentarios antes de añadirlos a las listas que serán analizadas por el modelo Naïve Bayes:

Figura 33. Limpieza previa de comentarios para ser analizados por el modelo Naïve Bayes.

```
#Ejecución después de obtener los comentarios con el YouTube Data API v3
def load_comments(match, videoid):
    for item in match["items"]:
        comment = item["snippet"]["topLevelComment"]
```

```

author = comment["snippet"]["authorDisplayName"]
text = comment["snippet"]["textDisplay"]
commentid = comment["id"]
usernameLIST.append(clean(author, no_emoji=True, lower=True))
commentLIST.append(clean(text, no_emoji=True, lower=True,
no_line_breaks=True, no_punct=True))
idLIST.append('https://www.youtube.com/watch?v={}&lc={}'.format(vid
eoid, commentid))
if 'replies' in item.keys():
    for reply in item['replies']['comments']:
        rauthor = reply['snippet']['authorDisplayName']
        rtext = reply["snippet"]["textDisplay"]
        rcommentid = reply["id"]
        usernameLIST.append(clean(rauthor, no_emoji=True,
lower=True))
        commentLIST.append(clean(rtext, no_emoji=True, lower=True,
no_line_breaks=True, no_punct=True))
        idLIST.append('https://www.youtube.com/watch?v={}&lc={}'.for
mat(videoid, rcommentid))
    
```

Fuente: Elaboración propia.

Tabla 3. Comparación entre Naïve Bayes vs YT-Spammer-Purge (15/08/2022).

Título del video y link	Dos Modelos (clean-text)	Falsos negativos	YT-Spammer-Purge
3 Apps Que Pagan Dinero Real En PayPal https://www.youtube.com/watch?v=SjnkUf5aBdE	560	4	580
¿Cómo invertir tus primeros mil dólares en 2022? Clips https://www.youtube.com/watch?v=dvecqwfU6xw	191	3	187
Cómo Invertir Tus Primeros 1000 en 2022 - Paso a Paso (Curso gratuito) https://www.youtube.com/watch?v=3ybL7JLBqIA	181	2	179

Cómo hacer dinero lo más rápido posible con tu negocio EP. #39 NMP https://www.youtube.com/watch?v=7wSFg0TZXfs	69	0	65
Cómo Invertir en Criptomonedas para Principiantes (2022) https://www.youtube.com/watch?v=pM53BJNGoKw	86	1	91

Fuente: Elaboración propia.

Se pudo ver una mejora significativa comparado al programa YT-Spammer-Purge debido a que en esta ocasión se contaron los comentarios que no fueron marcados como spam por el modelo de predicción, teniendo una tasa bastante baja de falsos negativos, pudiendo decir que el modelo Naïve Bayes resulta identificar ligeramente mejor los comentarios spam que los filtros utilizados por el programa YT-Spammer-Purge.

Cabe mencionar que los filtros de dicho programa son actualizados constantemente para mejorar la clasificación de spam; de igual manera el modelo puede ser reentrenado la cantidad de veces necesarias para mejorar su precisión aún más, esto quiere decir que a medida que se escaneen y descarguen comentarios y nombres de usuario, si poseen un porcentaje considerable de ser analizado para su clasificación manual, se lo podrá apartar en un archivo adicional para su revisión.

4.4. Diseño de un programa que reporte y elimine comentarios spam.

Como paso final en el desarrollo del proyecto, se necesitará tomar acción con los comentarios y usuarios que han sido identificados previamente como spam, estafas, fraudes, etc. Para ello se hará uso del mismo YouTube Data API v3 y el tipo de peticiones que ofrece para marcar estos comentarios como spam y dado el caso que seamos dueños del canal, eliminarlos debido a que por ningún motivo se puede eliminar comentarios de otros canales si no somos dueños del mismo, únicamente se tiene permitido reportarlos hacia YouTube.

Figura 34. *Limites en las cuotas por YouTube Data API v3.*

Filtro Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Cuota	Dimensiones (p. ej., ubicación)	Límite	Porcentaje de uso actual ↓
<input type="checkbox"/>	Queries per day		10,000	 2.35 %
<input type="checkbox"/>	Queries per minute		1,800,000	 0 %
<input type="checkbox"/>	Queries per minute per user		180,000	– ?

Fuente: Elaboración propia.

También existen ciertas limitaciones con la cuota de peticiones por defecto al usar dicho API, en total 10,000 peticiones por día (Google Developers, 2019), las cuales aproximadamente se pueden desglosar de la siguiente manera:

- 50 unidades por petición al Eliminar / reportar / ocultar comentarios.
- 1 unidad por petición al obtener comentarios en intervalos de 100.
- 100 unidades por obtener los 5 comentarios más recientes.

Aquello se puede traducir en:

- Eliminar / reportar / ocultar 200 comentarios máximos al día.
- Obtener máximo 1,000,000 comentarios al día.
- Obtener 500 comentarios recientes máximo por día

Por ello se deberá considerar la cantidad de comentarios a descargar para poder así tener la suficiente cuota para eliminarlos también.

Teniendo en cuenta los límites de YouTube Data API v3 claros se utilizará las siguientes peticiones para eliminar / bloquear / reportar comentarios que fueron clasificados como spam por el modelo Naïve Bayes, tomando en cuenta que cada acción se realizará por cada uno de los comentarios.

Debido a estas grandes limitaciones se ha optado por desarrollar una alternativa el cual permita tratar los comentarios seleccionados como spam de manera que se pueda reportarlos o de ser posible eliminarlos y banear el usuario indefinidamente de forma automatizada. Para ello se hará uso de Chromedriver, una herramienta de código abierto que sirve para realizar automatizaciones de aplicaciones web en muchos navegadores (Chromium, 2022).

Figura 35. Bot para eliminar / reportar comentarios spam mediante el uso de chromedriver.

```
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import pathlib

def ReportComment(report_menu):
    driver.execute_script("arguments[0].click();", report_menu)

    report_option = '/html/body/ytd-app/ytd-popup-container/tp-yt-paper-dialog/yt-report-form-modal-renderer/tp-yt-paper-dialog-scrollable/div/div/yt-options-renderer/div/tp-yt-paper-radio-group/tp-yt-paper-radio-button[1]'
    report_option = wait.until(EC.presence_of_element_located((By.XPATH, report_option)))
    driver.execute_script("arguments[0].click();", report_option)
    reason = report_option.text

    report_button = '/html/body/ytd-app/ytd-popup-container/tp-yt-paper-dialog/yt-report-form-modal-renderer/div/yt-button-renderer[2]/a/tp-yt-paper-button/yt-formatted-string'
    if IS_OWNER:
        report_button = '/html/body/ytd-app/ytd-popup-container/tp-yt-paper-dialog/yt-report-form-modal-renderer/div/yt-button-renderer[2]/yt-button-shape/button/yt-touch-feedback-shape/div/div[2]'
        report_button = wait.until(EC.presence_of_element_located((By.XPATH, report_button)))
        driver.execute_script("arguments[0].click();", report_button)

    report_done = '/html/body/ytd-app/ytd-popup-container/tp-yt-paper-dialog[2]/yt-confirm-dialog-renderer/div[2]/div[2]/yt-button-renderer[3]'
    report_done = wait.until(EC.presence_of_element_located((By.XPATH, report_done)))
    driver.execute_script("arguments[0].click();", report_done)
    print(reason)

IS_OWNER = False
is_Reply = False
DELETION_MODE = "REPORT"
TARGET_VIDEO = 'zGgd4v0a28M'
TARGET_ID = []#LISTADO DE LOS IDs DE COMENTARIOS SPAM

options = webdriver.ChromeOptions()
user = pathlib.Path().home()
print(user)
```

```

options.add_argument(f"--user-data-
dir={user}\\AppData\\Local\\Google\\Chrome\\User Data")
options.add_argument("--disable-extensions")
options.add_argument('--blink-settings=imagesEnabled=false')
driver = webdriver.Chrome('chromedriver.exe',chrome_options=options)

for i in range(len(TARGET_ID)):
    URL = 'https://www.youtube.com/watch?v={}&lc={}'.format(TARGET_VIDEO,
TARGET_ID[i])
    driver.get(URL)
    wait = WebDriverWait(driver, 100)

    first_comment = '/html/body/ytd-app/div[1]/ytd-page-manager/ytd-watch-
flexy/div[5]/div[1]/div/div[2]/ytd-comments/ytd-item-section-
renderer/div[3]/ytd-comment-thread-renderer[1]/ytd-comment-renderer'
    while True:
        driver.execute_script("window.scrollTo(0, 100)", "")
        try:
            driver.find_element(By.XPATH, first_comment)
            break #SCROLLING EN LA PÁGINA HASTA ENCONTRAR EL PRIMER COMENTARIO
        except:
            pass

    #COMPROBAR SI EL COMENTARIO EXISTE, CASO CONTRARIO CONTINUA CON LA LISTA
    first_comment = driver.find_element(By.XPATH, first_comment)
    comment_badge = '//*[@id="linked-comment-badge"]/ytd-badge-supported-
renderer'
    comment_badge = first_comment.find_elements(By.XPATH, comment_badge)
    if len(comment_badge) == 0:
        print('-> Comentario no encontrado: {}, cargando el siguiente
comentario...'.format(TARGET_ID[i]))
        continue

    if '.' in TARGET_ID[i]:
        is_reply = True
    if is_reply:
        dropdown_menu = '/html/body/ytd-app/div[1]/ytd-page-manager/ytd-watch-
flexy/div[5]/div[1]/div/div[2]/ytd-comments/ytd-item-section-
renderer/div[3]/ytd-comment-thread-renderer[1]/div/ytd-comment-replies-
renderer/div[2]/ytd-comment-renderer/div[3]/div[3]/ytd-menu-renderer/yt-icon-
button/button/yt-icon'
    else:
        dropdown_menu = '/html/body/ytd-app/div[1]/ytd-page-manager/ytd-watch-
flexy/div[5]/div[1]/div/div[2]/ytd-comments/ytd-item-section-
renderer/div[3]/ytd-comment-thread-renderer[1]/ytd-comment-
renderer/div[3]/div[3]/ytd-menu-renderer/yt-icon-button/button'
        print('first base')
        dropdown_menu = wait.until(EC.presence_of_element_located((By.XPATH,
dropdown_menu)))

```

```

driver.execute_script("arguments[0].click();", dropdown_menu)

menu_options = '/html/body/ytd-app/ytd-popup-container/tp-yt-iron-
dropdown/div/ytd-menu-popup-renderer/tp-yt-paper-listbox'
menu_options = wait.until(EC.presence_of_element_located((By.XPATH,
menu_options)))
menu_options = menu_options.find_elements(By.XPATH, '*')
print('second base')
if len(menu_options) > 2:
    IS_OWNER = True
    if DELETION_MODE == 'REPORT':#REPORTAR Y REMOVER COMENTARIO DEL VIDEO
        report_menu = menu_options[-2]
        ReportComment(report_menu)
    else:#OCULTAR COMENTARIO DEL VIDEO Y AL USUARIO DEL CANAL
        hide_comment = menu_options[-1]
        driver.execute_script("arguments[0].click();", hide_comment)

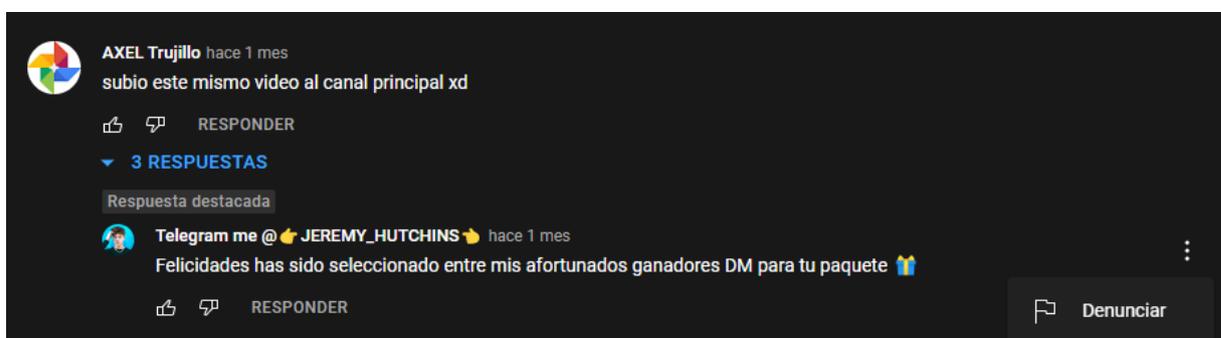
elif len(menu_options) == 1:#REPORTAR SOLO SI NO SE ES DUEÑO DEL CANAL
    report_menu = '/html/body/ytd-app/ytd-popup-container/tp-yt-iron-
dropdown/div/ytd-menu-popup-renderer/tp-yt-paper-listbox/ytd-menu-service-
item-renderer'
    report_menu = wait.until(EC.presence_of_element_located((By.XPATH,
report_menu)))
    ReportComment(report_menu)

```

Fuente: Elaboración propia.

Debido a que YouTube permite resaltar cualquier comentario o *sub-comentario* si se tiene el ID del comentario, es posible poner en primer lugar el comentario para que el *bot* rápidamente lo identifique y pueda enviar clics sobre los elementos web que nos interesen, los cuales serían abrir las opciones de cada comentario, seleccionar eliminarlo o denunciarlo.

Figura 36. Respuesta resaltada como destacada por medio del ID de comentario.



Fuente: Elaboración propia.

Debido a que el comentario spam siempre estará en primer lugar resulta replicable cargar el sitio web con el ID de comentario para efectuar la acción de eliminar / reportar repetidamente hasta terminar con la lista de spam brindada. El script funcionará como usuario dueño del canal eliminando y reportando comentarios, como también una cuenta independiente del canal que solo tendrá permitido reportar los comentarios spam encontrados.

4.5. Evaluación

Para continuar con la comparativa entre YT-Spammer-Purge con el programa desarrollado, se procedió a reportar los comentarios de la misma manera que lo hace el primer programa, utilizando el API de YouTube, se tienen las siguientes opciones para tratar los comentarios spam.

Figura 37. Tipos de peticiones para lidiar con los comentarios spam.

```
# Expresa la opinión de que la persona que realizó uno o más comentarios
deben marcarse como spam.
for comment_id in spamComment:
    result = youtube.comments().markAsSpam(id=comment_id).execute()

# Rechaza un comentario por no ser apto para su visualización.
# Esta acción también oculta todas las respuestas al comentario rechazado.
for comment_id in spamComment:
    result = youtube.comments().setModerationStatus(id=comment_id,
moderationStatus = "rejected", banAuthor = True).execute()

# Elimina el comentario especificado.
for comment_id in spamComment:
    result = youtube.comments().delete(id=comment_id).execute()
```

Fuente: Elaboración propia.

Corroborando que el código funciona correctamente, se realizó una comparación con los tiempos que tardan ambos programas en reportar los comentarios, de una misma lista de videos, pero sobre todo y previamente, el hecho de identificar y clasificarlos como spam o ham.

Tabla 4. Naïve Bayes vs YT-Spammer-Purge en cuestión al tiempo de ejecución (15/08/2022).

Título del video y link	Total de comentarios	Dos Modelos (cleantext)	YT-Spammer-Purge
3 Apps Que Pagan Dinero Real En PayPal https://www.youtube.com/watch?v=SjnkUf5aBdE	992	21 seg.	33 seg.
¿Cómo invertir tus primeros mil dólares en 2022? Clips https://www.youtube.com/watch?v=dvecqwfU6xw	1,490	29 seg.	45 seg.
Cómo Invertir Tus Primeros 1000 en 2022 - Paso a Paso (Curso gratuito) https://www.youtube.com/watch?v=3ybL7JLbqIA	1,403	22 seg.	36 seg.
Cómo hacer dinero lo más rápido posible con tu negocio EP. #39 NMP https://www.youtube.com/watch?v=7wSFgOTZXfs	564	13 seg.	23 seg.
Cómo Invertir en Criptomonedas para Principiantes (2022) https://www.youtube.com/watch?v=pM53BJNGoKw	1,418	28 seg.	46 seg.

Fuente: Elaboración propia.

Por los resultados se puede afirmar que el modelo de Naïve Bayes clasifica más rápido los comentarios spam, siendo esto sumamente importante si esta tarea se la realiza con una gran suma de comentarios donde se notará mucho más la diferencia de tiempo.

Ambos programas toman aproximadamente el mismo tiempo reportar / eliminar comentarios de spam ya que hacen uso del limitado API de YouTube; si se desea realizar estas acciones continuamente sin restricciones la mejor alternativa será empleando el *bot* con Chromedriver.

5. Conclusiones

- El proyecto toma la iniciativa de combatir amenazas tales como spam, prácticas engañosas y estafas dentro de la comunidad de YouTube; las cuales han ido aumentando con los años y que muy probablemente continúan creciendo si las mismas plataformas no ejecutan acciones efectivas; la aplicación desarrollada no solo clasifica los comentarios spam, sino que también permite eliminarlos y reportarlos.
- Se resalta que en algunos de los trabajos relacionados a este proyecto, utilizan como base un *dataset* de comentarios spam público para entrenar un modelo de *machine-learning* de clasificación de spam, esta base de datos recopila 1956 comentarios entre ellos spam y *ham*; sin embargo, la información de dicho *dataset* fue extraída de videos musicales y no de temas relacionados con el dinero o inversiones, además estos comentarios son del 2017, año en el cual este tipo de fraudes y estafas personificando a los creadores de contenido no eran muy comunes. Es por ello que se decidió recopilar un *dataset* con comentarios actuales y que los mismos fueran clasificados manualmente como spam o no spam para así obtener un clasificador más preciso ante las amenazas de la actualidad.
- En la gran mayoría de proyectos investigados solamente se buscan métodos de reconocimiento de spam, mas no de maneras de como mitigarlos o tomar acción frente a ellos una vez han sido identificados. Además de que, en todos los trabajos relacionados encontrados, se desarrollan sistemas de detección de spam de texto en inglés, lo cual presenta una oportunidad para crear sistemas de clasificación de spam basados en otros lenguajes como en este caso es en español; todas esas razones hacen distintivo al presente proyecto.
- Resultó indispensable ejecutar una limpieza previa al *dataset*, acomodándolo de tal forma que sean mejor entendidos por el modelo de aprendizaje automático, estrategia que se aplicó también en la clasificación de spam al momento de descargar los comentarios con el API de YouTube, lo cual aumento significativamente su precisión reduciendo los falsos positivos y verdaderos negativos, siendo este un requisito para poder disponer de un modelo certero y confiable en la identificación de spam.
- A pesar de adicionar el proceso de limpieza previa a los comentarios obtenidos por el API de YouTube, la aplicación desarrollada es bastante rápida comparado con YT-

Spammer-Purge, el cual emplea otros métodos de clasificación de spam pero que al parecer requieren un poco más de tiempo que el modelo entrenado en este trabajo.

- Se comprobó que el *dataset* es un pilar fundamental para disponer de un modelo de predicción preciso y adecuado, es por ello que se necesita obtener una cantidad suficiente de datos, y sobre todo se debe evitar tener un mismo comentario como parte de spam y *ham*; así como también se debe cuidar en el *dataset* como tal, que no existan falsos positivos ni verdaderos negativos debido a que esto afectará de forma directa a los resultados de predicción por parte del modelo.
- YouTube Data API v3 resulta ser bastante eficiente al momento de hacer peticiones para obtener los comentarios de uno o varios videos escogidos, sin embargo, su límite de cuota para peticiones resulta ser extremadamente corta al momento de requerir más de 10,000 peticiones al día, por lo que se podrían buscar alternativas para realizar las peticiones de eliminar / reportar comentarios sin hacer uso del API de YouTube; de no ser posible se podrían generar varias credenciales para aumentar el límite de cuota.
- Una alternativa al API de YouTube propuesta para la eliminación y denuncia de comentarios clasificados como spam de manera ilimitada y automatizada ha sido desarrollar un *bot* empleando chromedriver, aunque puede resultar mucho más lento que hacer peticiones a través de un API, superará eficazmente la eliminación de más 200 comentarios al día, cumpliendo los objetivos iniciales del proyecto y permitiendo a las personas que usen esta herramienta eliminar aproximadamente 1,000,000 de comentarios por día con una sola credencial obtenida de YouTube Data API v3.
- El modelo entrenado por Naïve Bayes demostró ser mucho más rápido y preciso al momento de clasificar los comentarios como spam o *ham* en comparación con utilizar un conjunto de filtros por cada comentario como lo hace TY-Spammer-Purge, por ello es apropiado analizar nuevamente la información utilizada como *dataset* para el entrenamiento de los modelos, ya que se pudieron pasar por alto muchos comentarios que debieron ser catalogados como spam y otros que quizás no.
- La cantidad de spam que exista en un video de YouTube depende de varios factores tales como: el ritmo de publicación de videos por parte del creador de contenido, el tipo de contenido en los videos: temas relacionados con inversiones, criptomonedas,

mercado bursátil, formas de generar dinero en sí; cantidad de suscriptores que disponga el canal de YouTube, ya que mientras más popular sea el canal, este será un mayor objetivo para que los ciberdelincuentes realicen estafas a sus seguidores. Los dueños de estos canales son los que se beneficiarían mayormente de este tipo de herramientas y que son los que protegerán a su audiencia debido a que pueden eliminar directamente este contenido mal intencionado.

- Los estafadores han encontrado muchas maneras para burlar los filtros de YouTube y los programas de clasificación de spam de terceros; empleando caracteres especiales, emojis, espaciado entre textos, subdividir el mensaje en múltiples comentarios y muchas más que complican la identificación del spam. También, estos intentan hacer más convincentes sus ataques hacia los usuarios personificando al creador de contenido, utilizando diálogos convincentes y usando la fotografía del dueño del canal, lo que causa que muchas personas sean convencidas en contactar a los estafadores.
- Los ciberdelincuentes siempre buscaran estar presentes en las plataformas con alto tráfico para intentar realizar sus ataques, por ello es importante que la comunidad actúe para defenderse o por lo mínimo este bien informada para no caer en este tipo de engaños y fraudes. También estas plataformas antes ya mencionadas deberían invertir más recursos en evitar este tipo de delitos como suplantación, estafas, phishing, redirección a contenido sexual, etc. a pesar de que el porcentaje de afectación a sus usuarios sea bajo. YT-Spammer-Purge es un claro ejemplo de que la comunidad trata de tomar acción por cuenta propia ante estas amenazas y a la vez ayudar a que otros lo hagan difundiendo las herramientas gratuitamente.
- Los fraudes informáticos y la suplantación de identidad pueden no ser considerados ciberdelitos en muchos países, sin embargo, esto no descarta que muchos sean víctimas de estas estafas ejecutadas por los que se valen del desconocimiento o imprudencia de las personas. Por ello también, debería ser responsabilidad de las plataformas notificar y hacer saber a su comunidad de este tipo de amenazas continuamente y a medida que vayan surgiendo.

5.1. Líneas de trabajo futuro

- El modelo de predicción es ampliamente mejorable ya que, al momento de clasificar comentarios, primeramente, deberán ser descargados, por lo que estos podrían ser recopilados y enviados de manera anónima y automática por las personas que hagan uso de la herramienta. Esta base de datos estaría destinada a ser revisada, reclasificada para realizar reentrenamiento de mejores modelos de predicción.
- Como trabajos futuros se puede considerar utilizar el mismo *dataset* para entrenar varios modelos *machine-learning* con diferentes algoritmos como se sugiere en otros trabajos relacionados; estos algoritmos pueden ser Random Forest, Support Vector Machine y K-Nearest Neighbor; con la finalidad de comparar cuál de ellos obtiene los mejores resultados en su rendimiento, en la cantidad de falsos positivos y negativos. De forma que se lo establezca como clasificador de spam por defecto en la aplicación.
- Debido a que YouTube no restringe ni controla que se usen determinadas fotos y/o nombres de usuario sin antes verificarlos y validarlos como prueba de identidad. Se puede considerar en un futuro trabajo analizar si la fotografía de los usuarios en los comentarios es la misma o se parece a la del dueño del canal, es decir que ningún usuario se intente hacer pasar por los creadores de contenido, si se llegase a detectar esto, el comentario sería identificado como spam.
- Buscar alternativas al API de YouTube que permitan la obtención de comentarios masivamente, con menos limitantes y restricciones de peticiones al día de forma que se pueda lograr ejecutar un *scraper* que realice escaneos continuos ante el spam y estafas.

6. Referencias bibliográficas

- Aiyar, S., & Shetty, N. P. (2018). N-Gram Assisted Youtube Spam Comment Detection. *Procedia Computer Science*, 132, págs. 174-182.
doi:<https://doi.org/10.1016/j.procs.2018.05.181>
- Alberto, T. (2017). *YouTube Spam Collection*. Obtenido de Center for Machine Learning and Intelligent Systems:
<https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection>
- Amir, A., & Zain, A. M. (2016). An Approach for Spam Detection in YouTube Comments Based on Supervised Learning. *2nd National Conference on Emerging Technologies*.
- Brownlee, J. (4 de October de 2019). *Machine Learning Mastery*. Obtenido de A Gentle Introduction to Bayes Theorem for Machine Learning:
<https://machinelearningmastery.com/bayes-theorem-for-machine-learning/>
- Bruyninckx, H. (November de 2002). Bayesian probability.
- Ceci, L. (04 de Abril de 2022). *Distribution of removed YouTube comments worldwide as of 4th quarter 2021, by removal reason*. Obtenido de Statista:
<https://www.statista.com/statistics/1133165/share-removed-youtube-video-comments-worldwide-by-reason/>
- Chiasmanto, A. R., Sari, A. K., & Suyanto, Y. (30 de Abril de 2022). CRITICAL EVALUATION ON SPAM CONTENT DETECTION IN SOCIAL MEDIA. *Journal of Theoretical and Applied Information Technology*, 100(8), 2642-2667. Obtenido de
<http://www.jatit.org/volumes/onehundred08.php>
- Chromium, W. (2022). *Chromedriver*. Obtenido de Chromium:
<https://chromedriver.chromium.org/>
- CUEMATH. (2022). Obtenido de Bernoulli Distribution:
<https://www.cuemath.com/data/bernoulli-distribution/>
- Derek O'Callaghan, M. H. (2012). Network Analysis of Recurring YouTube Spam Campaigns. *Proceedings of the Sixth International AAI Conference on Weblogs and Social Media*

(págs. 531-534). Dublin: School of Computer Science & Informatics, University College Dublin.

Educative Inc. (2022). Obtenido de What is Gaussian Distribution?: <https://www.educative.io/answers/what-is-gaussian-distribution>

Elijah Bouma-Sims, B. R. (Abril 2021). A First Look at Scams on YouTube. *Presented at the Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb) 2021 hosted at NDSS '21*. Carolina del Norte: <https://arxiv.org/abs/2104.06515v1>.

Elissa M. Redmiles, N. C. (2018). Examining the Demand for Spam: Who Clicks? *CHI 2018* (pág. Paper 212). Montréal: <https://dl.acm.org/doi/pdf/10.1145/3173574.3173786>.

Elmogly, A. M., Tariq, U., Mohammed, A., & Ibrahim, A. (2021). Fake Reviews Detection using Supervised Machine Learning. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 12(1). doi:<http://dx.doi.org/10.14569/IJACSA.2021.0120169>

Filter, J. (2 de 02 de 2022). *clean-text*. Obtenido de pypi: <https://pypi.org/project/clean-text/>

Google Developers. (11 de Junio de 2019). *API Reference*. Obtenido de Google Developers: <https://developers.google.com/youtube/v3/docs>

Google Developers. (11 de Junio de 2019). *YouTube Data API v3*. Obtenido de Google Developers: <https://developers.google.com/youtube/v3/docs/>

Malik, F. (20 de June de 2020). *Towards Data Science*.

Marques, & Andrew. (21 de Abril de 2022). *Youtube is Trying to Fix the Spam Issue*. Obtenido de YouTube: <https://www.youtube.com/watch?v=nfOPc3YhgZM>

Math.net. (2022). Obtenido de Gaussian distribution: <https://www.math.net/gaussian-distribution>

McCord, M., & Chuah., M. (2011). Spam detection on twitter using traditional classifiers. *In Proceedings of the 8th international conference on Autonomic and trusted computing (ATC'11)*, 175–186.

Pisa, M. (22 de April de 2022). Obtenido de Gaussian Distribution: What it is, How to Calculate, and More: <https://blog.quantinsti.com/gaussian-distribution/>

- Polanski, D. (Enero de 2022). Comparison of Different Machine Learning Algorithms for Classifying Spam Comments on YouTube Videos. Bournemouth, Reino Unido. doi:<http://dx.doi.org/10.13140/RG.2.2.24018.07365/1>
- Reddy, Y. S., Nithin, T., Meghana, V., & Sindhu, P. (Abril de 2022). N-GRAMS ASSISTED YOUTUBE SPAM. *YMER*, 21(4), 332-345.
- Russell, R. (2018). Machine Learning. En R. Russell, *Machine Learning Step-by-Step Guide To Implement Machine Learning algorithms with Python*. (pág. 103).
- Samsudin, N. M., Foozy, C. F., Alias, N., Shamala, P., Othman, N. F., & Din, W. I. (2019). Youtube spam detection framework using naïve bayes and logistic regresion. *Indonesian Journal of Electrical Engineering and Computer Science*, 14(3), 1508-1517. doi: 10.11591/ijeecs.v14.i3.pp1508-1517
- Scikit-Learn. (25 de Julio de 2022). *Scikit-Learn*. Obtenido de https://scikit-learn.org/stable/getting_started.html
- Shuttleworth, M. (15 de May de 2009). *Explorable.com*. Obtenido de <https://explorable.com/bayesian-probability>
- Sohom Bhattacharya, S. B. (Septiembre 2021). Machine learning-based Naive Bayes approach for divulgence of Spam Comment in Youtube station. *International Journal of Engineering and Applied Physics* , 278-284.
- Statistics How To . (2022). Obtenido de Bernoulli Distribution: Definition and Examples: <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/probability-distribution/bernoulli-distribution/>
- ThioJoe. (2022). *YT-Spammer-Purge*. Obtenido de Github: <https://github.com/ThioJoe/YT-Spammer-Purge>
- Vaidya, D. (2022). Obtenido de Multinomial Distribution: <https://www.wallstreetmojo.com/multinomial-distribution/>
- Zainuddin, N. (Agosto 2019). Video spam comment features selection using machine. *Indonesian Journal of Electrical Engineering and Computer Science*, 1046-1053.

7. Anexo A. Nombres de usuarios marcados como spam

AUTHOR,CLASS

ROBUX gratis en mi perfil 🐼,1
@REGINAFLORES3031 on Telegram,1
@WhatsApp + 1 6 2 6 7 6 7 4 6 5 1,1
@WhatsApp + 1 2 1 3 3 9 8 6 9 6 6,1
[Jody] ❤️ F**CK ME - CHECK MY PROFILE,1
+16175895526 what'sapp me 🇺🇸,1
+1 2 6 9 3 3 6 6 2 5 4 whatsapp,1
+1 7 6 2 3 4 5 5 1 0 9 WhatsApp Me,1
whatsapp 🌿 +1 2 5 1 5 5 1 2 2 9 8,1
📞 WhatsApp +1 4 0 9 2 6 3 0 5 3 5,1
➔ lordhack24 on Instagram 📷,1
About Crypto 🖱️ BeZosHack On Telegram,1
📞 WhatsApp +1 5 6 1 9 3 3 5 4 1 9,1
📞 WHATSAPPME + 1 7 3 2 5 9 7 9 3 1 3,1
cajal12@gmail.com cajak,1
Whatsapp Me +14435307288,1
DM Reginaflores3031 on Telegram,1
👉 +1 9 8 5 2 4 7 9 3 3 7 whatsapp,1
👉 496_vincent 🇺🇸 on INSTAGRAM,1
👉 BURUHACK EN INSTAGRAM,1
👉 smartbitcoininvestment on telegram,1
👉 Whatsap +1 7 7 4 3 5 8 0 4 2 8,1
👉 Whatsap +4 4 7 4 6 8 2 5 4 7 5 2,1
👉 WHATSAP +4 4 7 4 6 8 2 5 4 7 5 2,1
👉 WHATSAPP +1 7 7 4 3 5 8 0 4 2 8,1
📞 R O B U X gratis en mi perfil 🐼,1
📞 whatsapp +1 5 1 8 3 0 7 8 2 9 8,1
📞 WHATSAPP +1 9 7 3 9 3 5 6 9 8 3,1
📞 Whatsapp +1 3 0 3 7 1 0 8 0 8 6,1
☀️ #+17139780302 Helpline,1
Fernandomartinez @Gmail.com,1
Get vbucks Free on my Channel,1
I recommend @Reginaflores3031 on Telegram,1
I recommended EvaHopkins on telegram,1
Ω Whatsap +1 401-360-6678,1
I'm subbing to everyone who subs to me,1
JACK WhatsApp +1 5 5 1 2 2 5 1 8 8 7,1
Joy 🍑 I WANT SEX 🚫 F U c _ k me,1
Message @REGINAFLORES3031 on Telegram,1
Message 👉 KIMPROCYBERWORLDTECH ON INSTAGRAM 🇺🇸,1
Plus 1 8 1 6 9 1 9 1 2 1 3 whatsapp,1
REGINAFLORES3031 on Telegram,1
REGINAFLORES3031 ON TELEGRAM IS THE BEST,1
RENDLETECH_OFFICIAL ON INSTAGRAM,1
SEARCH BURU_HACK EN INSTAGRAM,1
Sekkuu T[A]P Me!! To Have [S]EX With Me33 ,1
Te-le-gram fwmerit,1

TELEGRAM +①②⑦⑦④①⑧③⑨③④,1
Telegram me @ PepperoniAudits,1
Telegram +1(314)472-5952,1
telegram +⑦⑨⑥⑥⑦③⑧①⑦①②,1
Telegram + ①⑧⑥⑦③⑤⑨①⑤①①,1
Telegram + ①⑧⑥⑦③⑤⑨①⑤①①,1
Telegram +①④⑦⑧④⑥⑤⑨⑦⑦⑦,1
Telegram ⊕①⑨⑧④②②④②⑨③⑦,1
Telegram 🐱 @mrbeast_give,1
Telegram 🐱 +①②⑦③③⑦②⑨⑥⑧⑥,1
TELEGRAM 🐱 +①④⑦①③⑥⑦⑥②⑧④,1
TELEGRAM 🐱 +①⑧⑦②⑥⑦①⑤③①⑧,1
TELEGRAM 🐱 ①④⑦①③⑥⑦⑥②⑧④,1
TELEGRAM 🐱 ②⑤④⑦①⑤⑦⑥⑦⑦②⑤,1
Telegram me @ 🐱 JEREMY_HUTCHINS 🐱,1
Telegram me 🐱 @AustinEvans_1,1
TELEGRAM ME 🐱 poke_rev221 🐱,1
TELEGRAM ME: 🐱 JOEYGAULE,1
Telegram Me: Ac_Hampton1,1
Telegram Me@TheBenAzeltart,1
Telegram@ 🐱 Official_Ferdy_Korpershoek,1
telegram+①⑦⑦③⑥⑨⑥⑥⑥①⑧,1
Telegram+①④⑦①③⑥⑦⑥②⑧④,1
telegram+①⑤⑤⑨③③②⑦⑤⑤①,1
telegram 🐱 +①④③①④⑦⑧②④⑦,1
telegram 🐱 +①⑥③⑦⑦⑤④①⑧④,1
Telegram 🐱 ⊕①⑨②⑨④⑥④⑤⑦⑥④,1
Telegram 🐱 ①④⑦①③⑥⑦⑥②⑧④,1
telegram 📞 +①⑥③⑦⑦⑤④①⑧④,1
TEXT WhatsApp +⑧④⑧①③⑦④③①⑤⑦,1
Text WhatsApp +⑧④⑧①③⑦④③①⑤⑦,1
TEXT+①⑥①⑦②⑥③⑨③②⑦ WHATSAP,1
Thanks to @REGINAFLORES3031 on Telegram,1
VETSPORES ON INSTAGRAM,1
Watsap + ①④②④②⑦⑧①⑥⑦⑥,1
WhatsApp ①②⑦①④⑦⑧③⑦⑧,1
what -Sap. +①⑧⑥⑤③⑦②⑥④⑤②,1
WHAT'SAPP. Me+①⑨①⑨⑦⑤⑥⑤⑧②⑧,1
WHAT'SAPP±①⑨⑦④⑤①⑤③②⑤④,1
whatapp+①⑨⑦⑦⑧①④⑧⑧④⑦,1
WHAT'S APP +①⑧⑤⑦⑤②⑨⑦④①⑧,1
Whats@pp + ①⑧⑦②③⑦⑧④①①⑦,1
Whats@pp+①④⑦①②⑦③⑦②③⑥,1
Whatsap ⊕12013570387,1
Whatsap + 1(818)796-2979,1
Whatsap + ①③②①②⑤②⑧③②⑥,1
Whatsap + ①⑥⑦①⑥⑥③②③③⑧,1
Whatsap + ①⑧①⑧⑦⑨⑥②⑨⑦⑨,1
Whatsap + ①⑧⑥③②⑨⑥⑧①②③,1
"whatsap +①⑧①⑦⑥③①③④⑥⑦",1
Whatsap +1(213)340- 6608,1
whatsap +1(908)271-8705,1

Whatsap +1(908)271-8705,1
Whatsap +16096449943 ,1
WhatsAp ⊕(1)(9)(2)(9)(8)(1)(3)(9)(9)(1)(9),1
Whatsap †1 401-360-6678,1
Whatsap †1786490-9055,1
Whatsap †18022651937,1
WHATSAP †19082484607,1
WhatsAp ⊕ 1 401360-6284,1
Whatsap ⊕(8)(1)(7)(6)(3)(1)(3)(4)(6)(0),1
Whatsap ±1 719-645-6925,1
Whatsap ±1(719)645-6925,1
WHATSAP ±9082484607,1
whatsap +1 8438825274,1
WhatsAp ⊕(1)(6)(1)(9)(6)(0)(7)(3)(4)(7)(7),1
WhatsAP++(1)(2)(1)(3)(9)(8)(0)(3)(9)(6)(7),1
Whatsap+13049551683,1
Whatsap+12316243569,1
whatsap+(1)(2)(0)(2)(6)(8)(1)(1)(0)(0)(5),1
whatsap+(1)(4)(0)(7)(5)(0)(2)(3)(9)(5)(5),1
whatsap+(6)(1)(9)(7)(2)(8)(5)(5)(3)(9),1
WhatSap+(1)(2)(1)(3)(3)(9)(8)(6)(9)(6)(6),1
WHATSAP†19082484607,1
WHATSAP±19082484607,1
Whatsap⊕(1)(6)(5)(1)(3)(2)(7)(0)(6)(2)(8),1
whatsap⊕(1)(8)(0)(2)(6)(7)(1)(5)(3)(1)(8),1
whatsap +1(4)(7)(0)(6)(2)(5)(5)(0)(7)(5),1
Whatsap 🍌 ⊕(1)(3)(8)(5)(2)(0)(4)(3)(6)(8)(6),1
whatsap 🍌 (1)(4)(1)(5)(7)(8)(7)(7)(6)(5)(1),1
whatsap 🍌 (1)(4)(1)(5)(7)(8)(7)(7)(6)(5)(1),1
WHATSAP 🍌 (1)(4)(1)(5)(7)(8)(7)(7)(6)(5)(1),1
WHATSAPP ⊕(1)(2)(4)(0)(3)(9)(4)(9)(3)(1)(5),1
WhatsApp †(1)(8)(1)(6)(9)(1)(9)(1)(2)(1)(3),1
WhatsApp + (1)(4)(3)(5)(2)(6)(1)(2)(4)(3)(2),1
WhatsApp +15614052911,1
WhatsApp +17327569106,1
WHATSAPP +49-15217791678,1
WhatsApp +(1)(3)(2)(1)(3)(5)(5)(2)(1)(0)(2),1
WhatsApp +(1)(6)(5)(1)(5)(3)(8)(5)(5)(9)(7),1
Whatsapp +(1)(6)(6)(2)(4)(2)(6)(6)(0)(9)(6),1
Whatsapp +(1)(8)(4)(7)(8)(9)(3)(7)(9)(8)(5),1
Whatsapp +(3)(5)(1)(9)(1)(4)(1)(8)(8)(8)(7)(0),1
WhatsApp +(1)(7)(2)(4)(9)(3)(1)(1)(3)(8)(6),1
Whatsapp +1(9)(2)(0)(2)(5)(0)(8)(4)(9)(4),1
Whatsapp (1)(7)(1)(3)(3)(6)(4)(2)(2)(2)(8),1
Whatsapp (1)(8)(1)(2)(2)(6)(0)(3)(7)(1)(2),1
WhatsApp †(1)(7)(8)(1)(4)(7)(4)(0)(2)(0)(6),1
WhatsApp †(4)(4)(7)(5)(0)(2)(0)(2)(5)(9)(8)(3),1
WhatsApp ±(1)(7)(6)(0)(4)(2)(8)(9)(7)(9)(0),1
WhatsApp †(1)(5)(6)(1)(4)(8)(5)(0)(4)(3)(6),1
WhatsApp ⊕(1)(6)(1)(4)(3)(8)(2)(0)(6)(9)(9),1
WHatsApp ⊕(1)(6)(7)(8)(5)(9)(0)(0)(7)(7)(1),1
WhatsApp ⊕(1)(2)(1)(3)(7)(7)(2)(5)(3)(1)(7),1
WhatsApp ⊕(1)(8)(6)(2)(2)(4)(5)(1)(1)(6)(4),1

WhatsApp 📞 (1 7 7 4 3 6 1 3 6 2 5),1
WHATSAPP 📞 +15135999492,1
WhatsApp 📞 +(1 6 3 0 4 6 7 5 1 3 7),1
WhatsApp 📞 (1 6 1 4 6 5 6 2 3 9 6),1
WhatsApp me 📞 (1 8 2 5 9 0 1 3 0 5 7),1
WhatsApp Me 📞 (1 3 0 1 4 0 9 2 5 9 6),1
whatsapp me 📞 +(1 9 7 3 4 4 6 9 1 4 7),1
WhatsApp Me: 📞 (1 7 6 2 4 3 3 6 0 0 2),1
WhatsApp me+(1 6 1 2 7 9 9 9 6 1 0),1
whatsapp) +(1 9 7 8 2 3 3 1 7 7 7),1
WhatsApp...+(1 3 2 0 8 5 5 8 2 6 5),1
WHATSAPP+(1 4 0 4 6 6 6 6 7 9 5),1
WhatsApp⊕+(4 4 7 9 0 0 0 1 4 8 4 7),1
WhatsApp⊕(3 5 7 9 6 4 7 9 9 1 2),1
WhatsApp⊕(3 5 7 9 6 4 7 9 9 1 2),1
WhatsApp⊕(4 4 7 8 8 7 3 1 0 6 3 0),1
WhatsApp⊕(4 4 7 7 7 6 7 6 2 2 5 9),1
WhatsApp⊕(4 4 7 4 4 8 1 5 1 3 0 5),1
WhatsApp⊕(4 4 7 4 6 4 4 9 9 8 7 7),1
WhatsApp† (1 5 1 3 7 7 8 0 6 2 8),1
WhatsApp± (3 4 6 0 4 1 3 6 8 6 9),1
Whatsapp± (1 6 0 2 5 5 2 7 4 5 9),1
WhatsApp† (1 5 1 0 5 1 9 6 8 1 3),1
WhatsApp† (1 5 5 9 4 7 4 9 5 5 8),1
WhatsApp† (1 2 4 0 8 0 3 2 4 3 4),1
WhatsApp† (1 3 2 1 3 5 7 3 7 6 9),1
WhatsApp† (1 3 3 4 3 7 7 2 1 6 1),1
WhatsApp† (1 9 5 6 6 2 0 2 6 9 5),1
WhatsApp† 📞 (1 3 2 1 3 5 7 3 7 6 9),1
WhatsApp⊕ (1 8 1 3 6 8 7 8 6 0 0) 📞,1
WhatsApp† (1 4 1 6 9 0 7 4 0 5 9),1
whatsapp† (1 2 4 0 3 9 4 -9 3 1 5),1
WhatsApp† (6 6 1 2 2 0 5 5 1 3),1
WhatsApp† (1 3 5 2 4 6 2 4 5 0 3),1
WhatsApp† (1 9 4 7 2 0 4 2 0 6 3),1
WhatsApp † (1 3 1 2 6 8 2 5 1 8 4),1
WhatsApp 📞 +(1 8 0 1 2 1 0 0 7 2 5),1
WHATSAPP 📞 +15135999492,1
WhatsApp 📞 +15135999492,1
Whatsapp 📞 +(1 2 1 7 5 1 8 2 6 0 0),1
Whatsapp 📞 +(1 7 7 4 3 6 1 3 6 2 5),1
Whatsapp 📞 +(1 9 7 2 6 3 4 6 9 7 0),1
WHATSAPP 📞 +(1 9 7 0 8 1 4 8 8 4 0),1
Whatsapp 📞 +(1 4 4 3 6 7 5 6 4 1 5),1
WhatsApp 📞 +(1 5 4 0 8 7 8 9 2 9 6),1
whatsapp 📞 +(1 8 1 3 8 6 7 4 0 3 7),1
whatsapp 📞 (1 4 1 5 7 8 7 7 6 5 1),1
whatsapp 📞 +(1 2 5 1 5 5 1 2 2 9 8),1
WhatsAppme 📞 (1 3 0 4 8 8 4 5 3 8 7),1
whatsapp+(1 9 7 9 2 7 4 3 1 9 0),1
Whatsapp+(1 6 1 7 2 6 3 9 3 2 0),1

WhtsApp+①⑥①⑦⑤①⑧⑥⑥①⑤,1
 WhtsApp+①⑥①⑦⑧⑨⑧⑤⑦⑦,1
 whtsp 🌿 +①②⑤①⑤⑤①②②⑨⑧,1
 yoesysm@gmail.com sanchez,1
 (Trader_Michael200). telegram,1
 jiiFxmumbledtrader EN TELEGRAM!!!,1
 cansion Achurylucia47@gmail,1
 Codedspyy11 en instagram,1
 cromosfcb.com,1
 DINO HIT7 at Instagram,1
 📩 severianotradingmil En Ig,1
 📩 severianotradingmil En Instagram,1
 Dylancryptorobot iG fb telegram,1
 Facebook yahuza Idris,1
 Facebook— 🌐 Jason Walter Bitcoin,1
 MrBeast010_on_telegram,1
 Opere con Fxmumbledtrader EN TELEGRAM,1
 Tankman WhatsApp,1
 telegram @Dereklunsford1,1
 Telegramchannel:Recovery_Team1,1
 Trader_Michael200 :en telegram,1
 WhatsApp I MX,1
 Whatsapp Ryan,1
 📱 +①⑤④⑧④⑨📱 WhatsApp,1
 📱 +①⑤④⑧④⑨⑦⑦③ WhatsApp,1
 ①⑨⑦④⑥⑧④②⑤①② WhatsApp,1
 WhatsApp 📩 +①⑤④⑦⑧⑧⑨②⑨⑥,1
 WHATSAP 📞 19082484607,1
 WHATSAP 📞 19082484607,1
 WHAT'SAPP. Me+①⑨①⑨⑦⑤⑥⑤⑧②⑧,1
 WHATSAP 📞 19082484607,1
 WHATSAPP 📩 +15135999492,1
 "Telegram me 📩"
 BabishCulinaryUniverse001",1
 Whatsap ±1(719)645-6925,1
 Nick WhatsApp +①⑥①⑨⑦②②⑤①⑦⑤,1
 WhatsAp 📞 1 401360-6284,1
 Telegrama @Adriansaenz01,1
 Telegrama 📩 @Adriansaenz01,1
 Whatsap 📞 12013570387,1
 WhatsApp+①⑨⑤⑥⑥②⑦②⑥⑨⑤,1
 Whatsap 📞 12013570387,1
 Whatsapp ± ①⑥⑦①⑧①④②④③⑦ •,1
 Whatsap+13158320192,1
 WHATSAPP 📩 +15135999492,1
 WhatsAp 📞 1 401360-6284,1
 WHATSAP 📞 19082484607,1
 ①②⑥⑨⑨⑦⑥⑦⑦⑦ WhatsApp ,1
 TELEGRAM +①②⑦⑦④①⑧③⑨③④,1
 Whatsap+13158320192,1
 Whatsap+13158320192,1
 telegram 📞 +①⑥③⑦⑦⑤④①⑧④,1

WHATSAPP 📞 +15135999492,1
telegram + (7) (9) (6) (6) (7) (3) (8) (1) (7) (1) (2),1
WHATSAP ☎️ 19082484607,1
WHATSAPP 📞 + (1) (9) (7) (0) (8) (1) (4) (8) (8) (4) (0),1
WhatsApp ☎️ + (1) (9) (2) (9) (8) (1) (3) (9) (9) (1) (9),1
WHATSAP ☎️ 19082484607,1
📧 WhatsApp + (1) (5) (6) (1) (9) (3) (3) (5) (4) (1) (9),1
Whatsap ☎️ +12013570387,1
WhatsApp ☎️ +1 401360-6284,1
Whatsap ☎️ +18022651937,1
Whatsap ☎️ +18022651937,1
WhatsApp + (1) (9) (5) (6) (6) (2) (0) (2) (6) (9) (5),1
WhatsApp + (1) (5) (5) (9) (4) (7) (4) (9) (5) (5) (8),1
Telegram ID. sumitwiseadvice,1
WhatsApp ☎️ + (1) (9) (2) (9) (8) (1) (3) (9) (9) (1) (9),1
WhatsApp +13158320192,1
telegram + (7) (9) (6) (6) (7) (3) (8) (1) (7) (1) (2),1
WhatsApp +13158320192,1
Adrian Saenz + (1) (2) (5) (2) (8) (5) (8) (0) (3) (3) (2),1
WHATSAP ☎️ 19082484607,1
WhatsApp ☎️ + (1) (9) (2) (9) (8) (1) (3) (9) (9) (1) (9),1
telegram + (7) (9) (6) (6) (7) (3) (8) (1) (7) (1) (2),1
WhatsApp ☎️ +12013570387,1
WHATSAP ☎️ 19082484607,1
WhatsApp +13158320192,1
(1) (2) (5) (2) (8) (5) (8) (0) (3) (3) (2),1
telegram + (7) (9) (6) (6) (7) (3) (8) (1) (7) (1) (2),1
Telegram ID. sumitwiseadvice,1
telegram + (7) (9) (6) (6) (7) (3) (8) (1) (7) (1) (2),1
WhatsApp + (1) (8) (5) (8) (8) (7) (6) (5) (6) (1) (9),1
📧 WhatsApp + (1) (5) (6) (1) (9) (3) (3) (5) (4) (1) (9),1
{WhatsApp } (1) (4) (1) (4) (4) (1) (5) (2) (1) (9) (4),1
Contact 📞 Recovery via telegram 🟢,1
WhatsApp ± (1) (4) (2) (5) (5) (3) (9) (7) (1) (7) (3),1
"SHEWENHACKS AT GMAIL,COM REAL LEGITIMATE SOFTWARE",1
WhatsApp + (1) (3) (5) (2) (8) (7) (1) (7) (7) (1) (9),1
WhatsApp + (1) (5) (3) (1) (2) (4) (8) (2) (8) (4) (1),1
Lascott patricia 📞 on FB,1
WhatsApp + (1) (8) (0) (4) (2) (1) (0) (6) (7) (1) (7),1
{WhatsApp } (1) (4) (1) (4) (4) (1) (5) (2) (1) (9) (4),1
"SHEWENHACKS AT GMAIL,COM",1
MESSAGE billhacker on telegram,1
chat BILLHACKER on telegram,1
telegram 📞 blessedf,1
{WhatsApp } (1) (2) (1) (0) (9) (6) (0) (6) (2) (6) (8),1
WhatsApp + (1) (8) (4) (7) (8) (9) (3) (7) (9) (8) (5),1
Text Sochina tyler on whatsapp + (1) (9) (3) (1) (2) (1) (9) (2) (6) (8) (7),1
Email Pperdavon@gmail,1
(1) (3) (1) (9) (8) (4) (9) (8) (7) (4) (5)..WhatsApp,1
Text 📞 OneWorldhack onTelegram,1
WHATSAP + (1) (2) (0) (1) (3) (6) (1) (4) (3) (3) (8),1
WHATSAP + (1) (8) (0) (8) (2) (1) (8) (9) (7) (7) (0),1

TEXT ME ON TELEGRAM 🖱️ SteveTerreberry,1
"SHEWENHACKS AT GMAIL,COM HELPS",1
jasonmaddison on IG,1
JASONMADDISON2020 on IG,1
Isaac Overall:visit CORETHACK ● COM website,1
Sonia flex 🖱️ CORETHACK●COM,1
Krudcracks_ON INSTAGRAM,1
jasonmaddison2020 on instagram,1
Jasonmaddison2020 via IG,1
CONTACT MELLEY_HACKS ON TELEGRAM,1
John 🖱️ hackerfargo1 on telegram 🖱️,1
JASONMADDISON2020 ON IG,1
Jasonmaddison2020 on Instagram,1
🇺🇸 Dr Olumo The Great On Facebook,1
search JASONMADDISON2020 IG,1
On Instagram Jasonmaddison2020,1
Jasonmaddison2020. on insta,1
Jasonmaddison2020 on IG 🖱️,1
Jasonmaddison2020 via instagram,1
Andrei Whatsapp+①⑧①⑥⑥⑥②④④②⑨,1
Dalk:-Cyber_Agency on telegram is the best,1
MESSAGE 🖱️ MIVACLONE 🖱️ ON TELEGRAM,1
Allan[hackerrambosmart1] On telegram,1
codedspyy11 on Instagram,1
Renee 🖱️ Maxmillhack ON Instagram 🖱️,1
crypto_jennifer_ on Instagram,1
BenJumper;;;;;;;;;;;;;hackerrambosmart1 On telegram,1
spice BEST WITH Cyber_Agency ON TELEGRAM 🇸🇬 ...,1
Hit 🖱️ DARKHUB43 🖱️ On Telegram,1
WhatsApp Mę+①⑦⑧⑥④⑤③⑨④④①,1
Patronize the best Webdack on telegram,1
Referring to CLEVERTECKS On instagram,1
Dm MR_marknic on telegram,1
Ana recommending CLEVERTECKS on ig,1
Referring to CLEVERTECKS on Instagram,1
WhatsApp +(1)(7)(4)(7)(2)(5)(7)(8)(9)(4)(8),1
Trade with "bulktrade " on Telegram ,1
" ● Trade with ""ALEXIS6TRADE" on TELEGRAM ● ",1
whsap plug+⑥③⑨⑤⑤⑤④⑦②⑨⑧⑨,1
WhatsApp me ①④⑦①⑧①⑨⑥⑧⑦⑤,1
DM 🖱️ HACKBYMESSIAH ■ ORG,1
DM 🖱️ HACKBYMESSIAH ■ ORG ,1
Text me on telegram 🖱️ Russell brand54 🖱️,1
"Make use of EagleSpy,Net 🖱️",1
Write him via 🖱️ @Coreytradings on instagram 🖱️,1
Reach out to 🖱️ @Coreytradings on instagram 🖱️,1
Message 🖱️ @Coreytradings on instagram 🖱️,1
Invest with 🖱️ @Coreytradings on instagram 🖱️,1
Trade with 🖱️ Coreytradings at gmail dot com 🖱️,1
Contact 🖱️ @Coreytradings on instagram 🖱️,1
Rubby Cyber_Agency on telegram is the best 💰,1
Krudcracks_HES ON INSTAGRAM,1

W+13527092570,1
Contact HACKERJOE1 on telegram real software,1
Penny.Trustworthy with Cyber_Agency on-telegram,1
Mary I made it with Cyber_Agency on telegram,1
@REGINAFLORES3031 On Telegram,1
REGINAFLORES3031 on Telegram,1
@REGINAFLORES3031 ON TELEGRAM,1
whatsappMe+12178053319,1
Lord Lincoln 🇺🇸 🇺🇸 JINNYHACKS ON TELEGRAM,1
hacker_collins on telegram,1
Text on WhatsApp +①⑨③①②①⑨②⑥⑧⑦,1
WhatsApp plus ①⑥①⑤⑧①⑧③⑦⑥①,1
CHAT Weblordhacks1 on Telegram,1
Recommend {Fredeugene} on Telegram ,1
GMAIL 🇺🇸 BITFUNDFXGROUP,1
CONTACT BITFUNDFX GROUP ON TELEGRAM,1
CONTACT 🇺🇸 BFFXGROUP ON TELEGRAM,1
Text Condathack on instagram,1
xdrhack on Telegram,1
TELEGRAM 🇺🇸 BITFUNDFX GROUP,1
Ⓡ①②⑤③⑤②⑦①④④④WHATSAPP,1
WhatsApp +①⑥①⑥④①⑨⑨①③⑥,1
Thanks Condathack on instagram,1
charlie_chang_trade 🇺🇸 ON TELEGRAM,1
WHATSAPP (+)+①④①④⑥⑥⑥⑦⑨⑤,1
He his RUSEXHACK ON INSTAGRAM,1
WhatsApp Me +①⑦③②③⑤②③⑦⑨⑦,1
WhatsApp Me +①④①⑦⑤①④④⑥⑤,1
Contact VINCRAKTECH ON INSTAGRAM,1
Micheal Rob 🇺🇸 { BLACKAIRHACK ● COM },1
Mia Saavedra: 🇺🇸 CORETHACK ● COM,1
Oluwaseun Otitoju 🇺🇸 BLACKAIRHACK•COM,1
Themba Eric🇿🇿{CORETHACK • COM website,1
Nicholas Donald: 🇺🇸 CORETHACK•COM website,1
Mark Ford 🇺🇸 BLACKAIRHACK ● COM,1
WhatsApp±①⑤⑤①②②⑤①⑧⑧⑦,1
Whatsapp 🇺🇸 +①⑤①⑥④③⑨⑨⑦①③,1
Whatsapp 🇺🇸 +①⑤①⑥④③⑨⑨⑦①③,1
Telegram @marla_roselinefx,1
WhatsApp 🇺🇸 +①⑤①⑥④③⑨⑨⑦①③,1
Emily+①③③⑤⑤①③④⑤⑦,1
I got 6.3BTC from Zhontech On telegram,1
Invest with MinerDon On telegram,1
I Cris recommend Zhontech ON TELEGRAM,1
Message MinerDon On telegram,1
Miner606 On telegram best platform,1
Miner606 got me Credited with 5.4BTC,1
mercerjoan09 On Insta,1
@Patrickbtc12 via Te le gr am,1
RicoTradingFx oN IG,1
Best regards to Nikto_miner 🇺🇸 on telegram 🇺🇸,1
Hackbanzer 🇺🇸 TELEGRAM,1

I recommend MinerDon On telegram,1
INVEST WITH MINERDON ON TELEGRAM,1
whatsapp 📞 +15164399713,1
👉 Jessicageorgefx@gmail.com 👉,1
CONTACT Great_Tec ON INSTAGRAM 📌,1
Reach out to:MBChacks on Instagram,1
Thomas reach out MBChacks on Instagram,1
What'sap-me①⑨⑧①⑦①⑤②⑥⑤③,1
WASTSAP + 109736386317,1
Scott Recommend speerskipper on telegram,1
WhatsApp 📞 +14707488402,1
Chris_uk On telegram,1
EagleSpy 🚩 Net,1
Alex Becker +①②①⑧⑤⑧⑤⑨②②④,1
📞ext④⑦⑨④⑦①①⑥③⑨,1
whatsapp+①-⑤③①⑥①⑦①④②⑨,1
inbox Janiemasonfx on telegram,1
what'sapp+①⑥①⑨⑦⑨③⑤⑨⑤④,1
TRADE_WITH_TOWNSEND 📌 N INSTA,1
WHATSAPP@t13128699076,1
Helpline +①⑤①⑤③⑧⑥①②⑤⑥,1
👉 jpbtc1 👉 on telegram,1
Haliton1 on telegram,1
✈️ Via insta Madrigal_kelvin_,1
Pin by Víctor Márquez,1
I recommend 👉 @Monage111 👉 on Instagram,1
Pined by victor márquez,1
ON IG 📌 TRADE_WITH_TOWNSEND,1
ON IG 👉 trade_with_townsend,1
look @ insta Madrigal_kelvin_,1
Dm: Ddoncarder On telegram,1
Kevin Jenkins 👉 RefCode FODBUU9F,1
follow @fx_with_mark Telegram,1
click @lisanedwardfx 👉 Telegram,1
Telegram handle 👉 marktilburys,1
Jason +①④⑦⑨③⑧⑤②⑦④③,1
Contact;;;;;;;;;;hackerrambosmart1 On telegram,1
📞 cubin_mike 👉 IG,1
DM 👉 CUBIN_MIKE 👉 INSTAGRAM,1
TRADE_WITH_TOWNSEND 📌 I A INSTA,1
👉 cubin_mike 👉 Instagram,1
Madrigal_kelvin_ On insta,1
👉 cubin_mike 👉 Instagram,1
📞 cubin_mike 👉 Instagram,1
MADRIGAL_KELVIN_ 👉 INSTAGRAM,1
A PRO ON 📌 IG MADRIGAL_KELVIN_,1
Marcus Attoh on *Instagram*,1
Marcus attoh on *Instagram*,1
@ Madrigal_kelvin_ on Instagram,1
"Lucia,,," 👉 CRYPT-ON-AID 🟢 COM",1
INBOX NOVADUPMSWORLD ON TELEGRAMmeee,1

VIA I<>G MADRIGAL_KELVIN_,1
 TEXTsixseveneight.foursixfive.eightfouroneone,1
 What'sApp ♡ (1)(9)(7)(8)(7)(0)(6)(3)(0)(7)(6),1
 Telegram @Meredith11,1
 ednaroodney on telegram,1
 text me on telegram @OFFICIAL_TECHNICALGURUJI,1
 Telegram +(1)(8)(1)(5)(6)(9)(7)(9)(8)(4)(2),1
 Evelynfxtrades On telegram,1
 🇵🇸 Hackbanzer 🇵🇸 On Telegram,1
 Zachary Plus(1)(3)(2)(1)(7)(1)(0)(6)(9)(7)(8),1
 i recommend BITFXTRADING on telegam,1
 i recommend BITFXTRADING on telegram,1
 Hackbanzer On Telegram,1
 Hackbanz On Telegram,1
 TELEGRAM 🇵🇸 🇵🇸 Hackbanzer Click it,1
 MADRIGAL_KELVIN_ 🚫 INSTAGRAM,1
 excelutensil On IG,1
 cyba_hackii on iG invest with her,1
 am Carolyn thanking Excelutensil on ig,1
 Follow on insta Logichacker_tech,1
 hack_asap on instagram dm,1
 contact cyba_hacki on iG he is the best,1
 ● On IG ● MADRIGAL_KELVIN_,1
 Elizabeth_william551 on iG invest with her,1
 Condathack on Telegram,1
 TEXT ME ON TELEGRAM @The_Command_Zone,1
 Telegram : Bernie_doran_fx,1
 +(1)(4)(7)(9)(3)(8)(5)(2)(7)(4)(3),1
 Invest via jasonmaddison2020 on IG,1
 Jasonmaddison2020 on insta,1
 Young Zooz;deal with CROWNS_HACK01 on IG,1
 Nicholas;message 🇵🇸 CROWNS_HACK01 ON INSTAGRAM,1
 CROWNS_HACK01 on INSTAGRAM,1
 Harry Giles;dealwith CROWNS_HARCK01 on instagram,1
 "Ennis Brian;chart 🇵🇸 CROWNS_HARCK01 on""instagram""",1
 CROWNS_HACK01 ON INSTAGRAM,1
 CONTACT ORXHACKS ON INSTAGRAM,1
 ORXHACKS 🌐 COM ON WEBSITE,1
 Farley_watson on INSTAGRAM,1
 Message me on telegram@ 🇵🇸 main_Faze_Rug,1
 Madrigal_Kelvin_Via 🇵🇸 I\G,1
 Te le gram 🇵🇸 keilymorrison,1
 Sticctechns on Instagram unlocked mine,1
 Done by Sticctechns on Instagram,1
 Kdrama: cuppyhack on Instagram,1
 Contact Sticctech on Instagram,1
 sticctechns on Instagram,1
 Sticctechns on Instagram fixed mine,1
 Contact on IG. Jasonmaddison2020,1
 Sticctechns on Instagram,1
 Contact Sticctechns on Instagram,1
 netrocyber On Instagram,1
 On insta jasonmaddison2020,1

Whátsapp + (1)(9)(0)(1)(2)(0)(5)(9)(1)(4)(4),1
 Contact 📩 INVISIBLECYBERHACKED ON TELEGRAM,1
 @STICKTECHS ON INSTAGRAM ❤️,1
 Contact speedcodz on instagram,1
 TYRON-----Braffhack on Telegram,1
 INBOX 🔵 AUTHORITY_HACK 🔵 ONTELEGRAM,1
 Robert ⬤ AUTHORITY_HACK VIA TELEGRAM,1
 TELEGRAM ME SPAWNWAVE,1
 WhAtSaPp+(1)(4)(0)(2)(5)(2)(0)(7)(1)(2)(2),1
CAVINTOOLS ON INSTAGRAM,1
 @trade_pay_out is my telegram handle,1
 WASAP+14692966586,1
 DETECTIFYHACK_2020 ON IG,1
 I'm Aisha DOVECYBER ON IG DID MY WORK,1
 pinñéd by•HardcOre Crypt0,1
 whatsapp 📞 (1)(5)(0)(0) 2 (1) 0 0(3)(4)(6),1
 instagram 📩 📩 Madrigal_kelvin_,1
 use 🌟 Rabypaid ON TELEGRAM,1
 Rabypaid ON TELEGRAM 📩,1
 use 📩 Rabypaid ON TELEGRAM 📩,1
 EvelynTHills 📩 (1)(3)(0)(9)(4)(6)(6)(6)(4)(0)(6),1
 Rabypaid ON TELEGRAM,1
 use 📩 Rabypaid ON TELEGRAM,1
 use 🌟 Rabypaid ON TELEGRAM,1
 Dm me+44(738)0197914,1
 Michaelboyer on Telegram,1
 JEFFREY: 📩 GURUCOD3 On Instagram,1
 VISIT JOHN 📩 +1(985) 284-4372 ON WHATSAPP,1
 KENTCRACKZ 📩 on instagram,1
 📩 414s_cybertrustechs on Instagram,1
 Text Jasonmaddison2020 on IG,1
 Alex__Hire Legithacks7 on TELEGRAM,1
 📩 Rabypaid ON TELEGRAM 📩,1
 📩 Rabypaid ON TELEGRAM 📩,1
 *ANTEZHACK*ON Instagram,1
 Telegram 📩 HACKERFUCCEO,1
 Lucinda Plus(1)(6)(8)(9)(2)(4)(4)(0)(2)(6)(9),1
 📩 DAHCYBER ON INSTAGRAM,1
 📩 whatsapp±13023451126 Stormer,1
 dm him + (4)(4)(7)(8)(9)(7)(0)(0)(4)(6)(1)(0),1
 Dm + (4)(4)(7)(8)(9)(7)(0)(0)(4)(6)(1)(0),1
 Contact 📩 Crayboxtech._ on IG,1
 Braffhack on Telegram,1
 lawrence_calvin on Instagram,1
 I victoria recommend @360johntoolz on IG.,1
 📩 Pm #procrack01 en Instagram,1
 Pinned by Solving The Money Problem.,1
 text carmichael007 on Instagram,1
GRATECYBER ON INSTAGRAM,1
 {name} 📩 FUNDBYMITNICK ON INSTAGRAM 📩,1
 Fundsbyintel On INSTAGRAM,1

TEXT MR PROF 📧 +13044599367,1
 pînned by Choíce Crypto,1
 CHECK *Coin__hacks1* ON INSTAGRAM,1
 Coin__hacks1 on Instagram,1
 Technopawns 🌐 com,1
 WhatsApp +14192370950,1
 Am Desmond Santos. using Lamoprofit on Telegram,1
 Lex=Fridman+18782196827,1
 Add Mrs Susan Telegram,1
 WhatsApp +1608 9 9 Four two eight Six,1
 WhatsApp +1447502025983,1
 WhatsApp +12056878759,1
199SUPPORTHELPCENTER on Insta,1
 📠 WhatsAppDM +13238490119,1
 📠 Admin WhatsAppDM +13238490119,1
 📠 Admin WhatsAppDM +13238490119,1
 Text me †19809077435,1
 Text me †19809077435..,1
 ishan Sharma 📧 27603744422,1
 binary_hacks007 on ig binary_hacks007 on ig,1
 DM AND MAKE INSTANT MONEY BOLTEXCHANGEHACKDOTCOM,1
 w+1(743) 887-8134,1
 WHAT'S APP+14054894393,1
 WHAT'S APP ME +16087131890,1
 WhatsApp (+)447828450768,1
 whatsapp +15752160945,1
 vera earn here web emmynitehackers. wordpress. com,1
 JUDICIAL_ CYBER ON INSTAGRAM,1
 You reach her via Telegram investwithsophiamaria,1
 Tiimfunds on IG,1
 Thanks to Tiimfunds on insta,1
 📞 Contact; Tyler.christon on Instagram,1
 Braffhack9 on Telegram,1
 BTAFFHACK9 ON TELEGRAM,1
 Braffhack9 on telegram 📧 📧,1
 telegram me 📧 @officialzhcchannel77,1
 WHATSAPP +12027011059,1
 "Nyla Lasley : 📧 FABULOUS HACKERS,COM 📧 ",1
 I invest with Anonymousehack01 on telegram,1
 Troy 📧 REDDITHACKS ON TELEGRAM 📧,1
 contact @stackbenefit on telegram,1
 +15168562993whatsapp 📧,1
 +15168562993whatsapp,1
 📧 Syman_TeChz On INSTAGRAM,1
 Whatsapp Me 📧 +18054193429,1
 WhatApp +12013782611.,1
 📠 DAHCYBER ON INSTAGRAM,1
 Richard James+13475604534,1
 Pinned_by_엽기팬돌이,1
 MEET 📧 GURUCRACKERS ON IN'STA,1
 hackerclusive on IG,1

Wilson betty- contact hackerclusive on ig,1
 Tom Charles HACKERCLUSIVE ON IG,1
 Jessica Hackerclusive on IG,1
 watsap+①⑥⑦⑤⑨⑤③⑥④①①,1
 +①⑥⑦⑤⑨⑤③⑥④①①watsap,1
 Zoomsfund On Instagram,1
 I recommend 🇺🇸 Zoomsfund On Instagram,1
 I recommend 🇺🇸 Mandaxhack On Instagram,1
 WhatsApchat±④④⑦⑧⑦⑨③①③②⑥①,1
 ALEX BECKER +①③②①③④⑤①⑥⑤⑨,1
 WhatsApp 🇺🇸 +①⑤⑦②⑧④⑨⑤⑤①⑧,1
 Whatsapp 🇺🇸 +①⑥⑥②⑤⑨⑦①⑧①⑨,1
 Akshat S. +①⑥①⑤②⑧⑦⑦⑦⑧⑨,1
 WHATSAPP ♀ ⑥④②①⑦②⑧⑨④⑥⑦①,1
 Whatsapp + ①⑧①②⑤⑤③②③⑦②,1
 WHAT'SAPP + ①⑨①⑦⑤⑨①②①⑦⑦,1
 Whatsapp + ①④①⑨③⑤⑨⑦③⑨⑦ •,1
 Text Me +①⑧③②⑨⑦④①⑦④④,1
 karanaujla_official,1
 Text Me On Telegram @MrBeast61,1
 telegram me 🇺🇸 Beaatphilanthropy,1
 telegram me 🇺🇸 @Beaatphilanthropy,1
 Im subbing To everyone who subs to me,1
 Text mrbeast ±I9167942985,1
 Telegram me @OfficialMrBeastX,1
 telegram 🇺🇸 scottkilmer01 🇺🇸,1
 Hit me up on Telegram 🇺🇸 @Mrbeast66,1