Universidad Técnica de Cotopaxi

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA: INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

"ALGORITMO PARA LA CLASIFICACIÓN DE ASPECTOS DE LENGUAJE NATURAL BASADOS EN WEB SEMÁNTICA"

Proyecto de Titulación presentado previo a la obtención del Título de Ingenieros en Informática y Sistemas Computacionales.

Autores:

Álvarez Lasso Francisco Bolívar

Mayo Pazuña Lenyn Santiago

Tutor:

Ing. MSc. Bravo Mullo Silvia Jeaneth

LATACUNGA - ECUADOR

AGOSTO 2019

Universidad Técnica de Cotopaxi

Ingeniería Informatica Y Sistemas Computacionales

DECLARACIÓN DE AUTORÍA

Nosotros, **ÁLVAREZ LASSO FRANCISCO BOLÍVAR Y MAYO PAZUÑA LENYN SANTIAGO,** declaramos ser autores del presente proyecto de investigación: **"ALGORITMO PARA LA CLASIFICACIÓN DE ASPECTOS DE LENGUAJE NATURAL BASADOS EN WEB SEMÁNTICA"** siendo Ing. MSc. Silvia Bravo Mullo tutor (a) del presente trabajo; y exigimos expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

.....	.....
Francisco Bolívar Álvarez Lasso	Lenyn Santiago Mayo Pazuña
Número de C.I. 050378900-0	Número de C.I. 050321763-0

Ingeniería
Informatica Y Sistemas
Computacionales

AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutora del Trabajo de Investigación sobre el título: **"ALGORITMO PARA LA CLASIFICACIÓN DE ASPECTOS DE LENGUAJE NATURAL BASADOS EN WEB SEMÁNTICA"** de los señores estudiantes, **Álvarez Lasso Francisco Bolívar**, con cédula de ciudadanía **No.- 050378900-0** y **Mayo Pazuña Lenyn Santiago,** con cédula de ciudadanía **No.- 050321763-0**, de la carrera Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Julio, 2019

Tutora de Titulación
Nombre: Ing. MSc. Silvia Jeaneth Bravo Mullo
CC: 050243712-2

Ingeniería
Informatica Y Sistemas
Computacionales

APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la **FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**; por cuanto, el o los postulantes: **Álvarez Lasso Francisco Bolívar**, con cédula de ciudadanía **No.- 050378900-0** y **Mayo Pazuña Lenyn Santiago,** con cédula de ciudadanía **No.- 050321763-0**, con el título de Proyecto de titulación **ALGORITMO PARA LA CLASIFICACIÓN DE ASPECTOS DE LENGUAJE NATURAL BASADOS EN WEB SEMÁNTICA**, han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.
Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, 22 Julio, 2019

Para constancia firman:

Lector 1 (Presidenta)
Nombre: Kantuña Flores Karla Susana
CC: 050230511-3

Lector 2
Nombre: Mayra Susana Albán Taipe
CC: 050231198-8

Lector 3
Nombre: Cadena Moreano José Augusto
CC: 050155279-8

AGRADECIMIENTO

Quiero expresar mi sincero agradecimiento a la Universidad Técnica de Cotopaxi, institución que ha sabido labrarse un espacio de gloria en nuestra provincia, así como a nivel nacional e internacional.

De igual manera a todas aquellas personas que estuvieron presentes siempre en este largo camino universitario.

FRANCISCO BOLÍVAR ALVAREZ LASSO

AGRADECIMIENTO

Agradezco a Dios y a la Virgencita, por darme la vida, por haberme dado la fe y la fuerza necesaria para concluir una etapa importante en mi vida profesional.

A mis padres, a mis hermanos quienes han sido el pilar fundamental para el logro de esta meta y me apoyaron en todo momento, ellos son las personas quienes a lo largo de mi vida me aconsejaron y me motivaron al no desmayar en ninguna adversidad, creyeron en mí en todo momento y no dudaron de mis habilidades.

MAYO PAZUÑA LENYN SANTIAGO

DEDICATORIA

La presente tesis está dedicada a todos quienes me apoyaron durante el trascurso de todo este proceso, sacrificando una confianza infinita y pudiéndome permitir concluir con esta etapa de mi vida.

Agradezco principalmente a Dios por la fuerza que me ha dado para seguir adelante y a mi Manuelito por todos los pedidos cumplidos.

A mis padres Ernesto y Lourdes, quienes me ayudaron incondicionalmente, a mis hermanos David y Luis por darme ánimos necesarios y mis amigos de universidad por tantas risas y enojos.

FRANCISCO BOLÍVAR ALVAREZ LASSO

DEDICATORIA

El presente proyecto de investigación se los dedico a Dios, a mis padres y hermanos. A Dios por las bendiciones recibidas a lo largo de mi vida quien supo guiarme por el buen camino para seguir adelante, no desmayar con los problemas que se me presentaban. A mis padres y hermanos quienes con su amor y confianza han sido mi guía en todo momento, sobre todo a mi madre quien en vida me dio ejemplo de valentía, fortaleza y con su apoyo brindado me permitió el haber llegado hasta este momento tan importante en mi formación profesional dándome la oportunidad de cumplir este sueño tan anhelado.

MAYO PAZUÑA LENYN SANTIAG

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

Título: "ALGORITMO PARA LA CLASIFICACIÓN DE ASPECTOS DE LENGUAJE NATURAL BASADOS EN WEB SEMÁNTICA"

Autores: Álvarez Lasso Francisco Bolívar

Mayo Pazuña Lenyn Santiago

RESUMEN

El presente proyecto de investigación trata sobre el diseño de un algoritmo para clasificar los aspectos de lenguaje natural basados en web semántica. Para ello, se realizó una revisión de la literatura de algoritmos de búsqueda, esta revisión dio como resultado la necesidad de proponer nuevas alternativas de búsqueda para mejorar los resultados de los mismos. Se observó, además, que en la actualidad, existen pocas propuestas que resuelvan este problema empleando herramientas de inteligencia artificial de forma eficiente. Por lo tanto, este trabajo propone emplear los algoritmos Ramdon Forrest y K-Nearest Neighbours (k-NN) en búsquedas web empleando datos basados en lenguaje natural. Para el desarrollo del algoritmo propuesto se empleó Python como lenguaje de programación para la creación y Prototipado del algoritmo de clasificación propuesto. Con este fin, se empleó la herramienta Spyder de la suite Anaconda y la librería Pandas, Sklearn en donde se encuentran los algoritmos de clasificación Random Forest Classifier y KNeighbors Classifier para Random Forest y Knn respectivamente. Random Forest consta de bosques aleatorios formados por un conjunto de árboles de clasificación que se eligen de forma aleatoria construida con N datos de la muestra con reemplazamiento. K-NN se basa simplemente en "recordar" todos los ejemplos que se vieron en la etapa de entrenamiento. Por lo cual, cuando un nuevo dato se presenta al sistema de aprendizaje, este se clasifica según el comportamiento del dato más cercano, la principal dificultad de este método consiste en determinar el valor k, ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo a la mayoría. El proceso experimental empleó cuatro Dataset extraídos de la web, las mismas son GBvideos, que contiene los comentarios sobre música de YouTube, vg1, que corresponde a las ventas de videos juegos, zomato que muestra los comentarios sobre restaurantes y AppStore que contiene los comentarios de las aplicaciones móviles. La cantidad de instancias analizadas corresponde a 57956 instancias. El análisis dio como resultado una tasa de predicción de la clasificación en Random Forest 0.7 o 70% y k-NN 0.6 o 60%. Para evaluar el algoritmo propuesto se empleó Auc Roc que obtuvo 0.7 de exactitud. Con este análisis se concluye que el uso de un algoritmo basado en Random Forest es el más confiable y preciso para la clasificación del lenguaje natural. Además, este algoritmo podría ser considerado como apoyo para estudiantes a fin de que se establezca en proyectos futuros.

Palabras Claves: clasificación de lenguaje natural, metodología, algoritmo, base de datos.

COTOPAXI TECHNICAL UNIVERSITY

ENGINEERING AND APPLIED SCIENCES FACULTY

TOPIC: "ASPECTS OF NATURAL LANGUAGE BASED ON WEB SEMANTICS WITH ALGORITHM CLASSIFACATION"

Authors: Álvarez Lasso Francisco Bolívar

Mayo Pazuña Lenyn Santiago

ABSTRACT

The present researching refers a design of an algorithm to classify aspects of natural language based on semantic web. For doing this, a literature review of search algorithms was carried out, this revision resulted in the need to propose new search alternatives to improve the results of the same. It was also observed that currently, there are few proposals that solve this problem using artificial intelligence tools efficiently. Therefore, this work proposes using Ramdon Forrest and K-Nearest Neighbors (k-NN) algorithms in web searches using data based on natural language. For the development of the proposed algorithm, Python was used as the programming language for the creation and prototyping of the proposed classification algorithm. To this end, the Spyder tool of the Anaconda suite and the Pandas, Sklearn library were used, where the Random Forest Classifier and KNeighbors Classifier, algorithms classified for Random Forest and Knn respectively are used. Random Forest consists of random forests formed by a set of randomly chosen classification trees constructed with N data from the sample with replacement k-NN is based simply on "remembering" all the examples that were seen in the training stage. Therefore, when a new data is presented to the learning system, it is classified according to the behavior of the closest data, the main difficulty of this method is to determine the value k, because if it takes a large value the risk is to do the classification according to the majority. The experimental process used four dataset extracted from the web, the same are GBvideos, which contains the comments on YouTube music, vg1, which corresponds to the sales of video games, zomato that shows the comments on restaurants and AppStore that contains the comments of the mobile applications. The number of instances analyzed corresponds to 57956 instances. The analysis resulted in a prediction rate of the classification in Random Forest 0.7 or 70% and k-NN 0.6 or 60%. To evaluate the proposed algorithm, Auc Roc was used, which obtained 0.7 of accuracy. With this analysis it is concluded that the use of an algorithm based on Random Forest is the most reliable and accurate for the classification of natural language. In addition, this algorithm could be considered as support for students in order to be established in future projects.

KEYWORDS: Natural Language Classification, Methodology, Algorithm, Data Base.

Universidad Técnica de Cotopaxi

CENTRO DE IDIOMAS

AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que: La traducción del resumen del proyecto de investigación al Idioma Inglés presentado por los señores Egresados de la Carrera de: **INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES** de la **FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS, ÁLVAREZ LASSO FRANCISCO BOLIVAR Y MAYO PAZUÑA LENYN SANTIAGO**, cuyo título versa **"ALGORITMO PARA LA CLASIFICACIÓN DE ASPECTOS DE LENGUAJE NATURAL BASADOS EN WEB SEMÁNTICA"**, lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo a los peticionarios hacer uso del presente certificado de la manera ética que estimaren conveniente.

Latacunga, Julio del 2019

Atentamente,

Msc. ALISON MENA BARTHELOTTY
DOCENTE CENTRO DE IDIOMAS
C.C. 050180125-2

CENTRO DE IDIOMAS

1. INFORMACIÓN GENERAL

Título del proyecto

Algoritmo para la clasificación de aspectos de lenguaje natural basados en web semántica

Fecha de inicio

Abril 2019

Fecha de finalización

Agosto 2019

Lugar de ejecución

Universidad Técnica De Cotopaxi

Facultad que auspicia

Facultad Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia

Ingeniería en Informática y Sistemas Computacionales

Proyecto de investigación vinculado:

Algoritmos dirigidos a usuarios en el contexto web semántica – Proyecto formativo

Equipo de trabajo

Tutor

Ing. Silvia Jeaneth Bravo Mullo

Estudiantes

Álvarez Lasso Francisco Bolívar

Mayo Pazuña Lenyn Santiago

Área de conocimiento:

Ingeniería en Informática y Sistemas computacionales

Línea de Investigación

Tecnologías de la información y comunicación (TIC's) y diseño gráfico

Sub Línea de investigación

Sistemas informáticos y métodos de Inteligencia artificial

2. RESUMEN DEL PROYECTO

El presente proyecto de investigación trata sobre el diseño de un algoritmo para clasificar los aspectos de lenguaje natural basados en web semántica. Para ello, se realizó una revisión de la literatura de algoritmos de búsqueda, esta revisión dio como resultado la necesidad de proponer nuevas alternativas de búsqueda para mejorar los resultados de los mismos. Se observó, además, que en la actualidad, existen pocas propuestas que resuelvan este problema empleando herramientas de inteligencia artificial de forma eficiente. Por lo tanto, este trabajo propone emplear los algoritmos Ramdon Forrest y K-Nearest Neighbours (k-NN) en búsquedas web empleando datos basados en lenguaje natural. Para el desarrollo del algoritmo propuesto se empleó Python como lenguaje de programación para la creación y Prototipado del algoritmo de clasificación propuesto. Con este fin, se empleó la herramienta Spyder de la suite Anaconda y la librería Pandas, Sklearn en donde se encuentran los algoritmos de clasificación Random Forest Classifier y KNeighbors Classifier para Random Forest y Knn respectivamente. Random Forest consta de bosques aleatorios formados por un conjunto de árboles de clasificación que se eligen de forma aleatoria construida con N datos de la muestra con reemplazamiento. K-NN se basa simplemente en "recordar" todos los ejemplos que se vieron en la etapa de entrenamiento. Por lo cual, cuando un nuevo dato se presenta al sistema de aprendizaje, este se clasifica según el comportamiento del dato más cercano, la principal dificultad de este método consiste en determinar el valor k, ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo a la mayoría. El proceso experimental empleó cuatro Dataset extraídos de la web, las mismas son GBvideos, que contiene los comentarios sobre música de YouTube, vg1, que corresponde a las ventas de videos juegos, zomato que muestra los comentarios sobre restaurantes y AppStore que contiene los comentarios de las aplicaciones móviles. La cantidad de instancias analizadas corresponde a 57956 instancias. El análisis dio como resultado una tasa de predicción de la clasificación en Random Forest 0.7 o 70% y k-NN 0.6 o 60%. Para evaluar el algoritmo propuesto se empleó Auc Roc que obtuvo 0.7 de exactitud. Con este análisis se concluye que el uso de un algoritmo basado en Random Forest es el más confiable y preciso para la clasificación del lenguaje natural. Además, este algoritmo podría ser considerado como apoyo para estudiantes a fin de que se establezca en proyectos futuros.

Palabras Claves: clasificación de lenguaje natural, metodología, algoritmo, base de datos.

3. JUSTIFICACIÓN DEL PROYECTO

El presente proyecto de investigación propone una alternativa en el proceso de búsqueda mediante la web, logrando una repuesta de clasificación rápida para la persona que interactúa con la máquina. Esta propuesta tiene el fin de eliminar desperdicio de tiempo y fatiga en búsqueda de información innecesaria. Para ello, se desarrolló un algoritmo de clasificación de aspectos de lenguaje natural basados en web semántica que nos permitirá un resultado de búsqueda inmediata y descartando información basura.

Los procesos de búsqueda involucran recorrer un arreglo completo de información con el fin de encontrar algo, lo más común es buscar el menor o mayor elemento (cuando se puede establecer un orden), o buscar el índice de un elemento determinado. Para buscar el menor o mayor elemento de un arreglo de información, se puede usar la estrategia de comparar con cada uno de los elementos, e ir actualizando la respuesta (D. Sánchez, 2014). Por lo tanto, es necesario contar con el algoritmo adecuado para búsquedas de información, lo cual es el objetivo de la presente propuesta. Este trabajo constituye un punto de partida para la mejora e implementación de algoritmos de búsqueda en cualquier plataforma. Esta propuesta nace de la necesidad de indagar un método confiable para la clasificación de aspectos de lenguaje natural dentro de la web semántica (Gutiérrez & Hurtado, 2015).

Los métodos de búsqueda aún no logran alcanzar óptimos niveles de búsqueda (Nicolás Fidalgo Belmonte, 2012), mientras que el algoritmo propuesto logra alcanzar niveles óptimos de búsqueda. Los algoritmos de búsqueda propuestos en la literatura no aplican algoritmos de inteligencia artificial como Random Forest y k-NN (Noi & Kappas, 2018). Con el fin de demostrar la eficiencia de estas técnicas de inteligencia artificial se han propuesto los novedosos algoritmos que se presentan en esta investigación.

En la actualidad la cantidad de usuarios que emplean búsquedas en Internet es de aproximadamente 4.388 millones diariamente, por lo cual resulta sumamente importante mejorar los algoritmos de búsqueda (de Computadores & Alfaro Olave, 2015). Por este mismo motivo investigadores de todo el mundo (Benavides, 2013) (Flores, 2017) (Ruiz-Lobaina & Romero-Suárez, 2017), han centrado sus esfuerzos en optimizar los algoritmos de búsqueda.

Es importante mencionar que para el desarrollo de este proyecto de investigación se cuenta con una asesoría adecuada en el ámbito de investigación relacionado al desarrollo del algoritmo propuesto, el cual aportó con sus conocimientos para que esta propuesta se desarrolle con éxito.

4. BENEFICIARIOS DEL PROYECTO

4.1. Beneficiarios directos

Usuarios e Investigadores que se encuentren relacionados al tema de búsquedas en la web, los investigadores que enfocan su trabajo al análisis semántica.

4.2. Beneficiarios indirectos

Los usuarios que realicen búsquedas en sistemas web, investigadores de proyectos relacionados.

5. EL PROBLEMA DE INVESTIGACIÓN

El instrumento que los seres humanos utilizamos para comunicar el conocimiento es el lenguaje natural. En la actualidad la búsqueda de información en Internet se ha vuelto muy compleja para encontrar las consultas o tareas que se ajusten al lenguaje que habitualmente empleamos los seres humanos. Actualmente, gran parte del conocimiento del ser humano se encuentra en forma digital en distintos tipos de colecciones de datos que se encuentran en la red. Los volúmenes de información sean inmensos, según la International Data Corporation (Gantz & Reinsel, 2011), el mundo genero 1,8 zettabaytes de Información digital en 2011 y en 2020 el mundo va generar 50 veces esa cantidad (J, 2020).

En la actualidad se requiere de propuestas algorítmicas más eficiente que permita la clasificación de lenguaje natural, ya que los algoritmos que se proponen en la literatura no satisfacer las necesidades de búsqueda que requiere el ser humano (Delgado & Amador, 2014). Por lo tanto, se requieren de propuestas que permitan alcanzar niveles óptimos en la identificación de patrones de lectura que deberían interpretar las máquinas con eficiencia (Estudio, 2018).

Los problemas que se identifican a partir del no contar con algoritmos adecuados de búsqueda van desde el desperdicio de tiempo (Nicolás Fidalgo Belmonte, 2012), hasta el obtener resultados que distan en gran medida de la búsqueda solicitada (Derechos, 2012). Lo cual ocasiona que miles de personas no tengan resultados de sus búsquedas realizadas provocando afectaciones en la distribución del conocimiento (Machasilla, 2015), a su vez que pueden generar perjuicios económicos equivalentes a miles de dólares, por el hecho de contar con un mecanismo de búsqueda adecuado (Soria, Pandolfi, Villagra, & Villagra, 2016). Se observa que diariamente 4.388 Millones de personas se conectan a Internet diariamente, lo cual

involucra que el proyecto propuesto afectará directamente a este grupo de personas, que hasta el momento no satisfacen sus necesidades de búsqueda (Tablet, 2017).

Por lo tanto se plantea el siguiente problema de investigación:

¿Qué algoritmo permitiría clasificar de forma óptima el lenguaje natural en la web semántica?

6. OBJETIVOS

6.1. General

Diseñar un algoritmo para la clasificación de aspectos de lenguaje natural basados en web semántica para la mejora en la búsqueda de información empleando técnicas de inteligencia artificial.

6.2. Específicos

- Analizar los mecanismos de clasificación de lenguaje natural que son empleados en los sistemas de búsqueda web mediante la revisión de la literatura.
- Evaluar los algoritmos que son empleados en la inteligencia artificial para seleccionar el más óptimo y aplicarlo en el presente proyecto de investigación.
- Desarrollar un algoritmo basado en inteligencia artificial para la clasificación mediante la aplicación de herramientas orientada a la web semántica.

7. ACTIVIDADES Y SISTEMAS DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS.

Tabla 1: Objetivos y actividades

OBJETIVOS	ACTIVIDAD	RESULTADOS DE LA ACTIVIDAD	MEDIOS DE VERIFICACIÓN
Analizar los mecanismos de clasificación de lenguaje natural que son empleados en los sistemas de búsqueda web mediante la revisión de la literatura.	**Tarea 1:** Investigar datos bibliográficos. **Tarea 2:** Determinar los Artículos que se serán Analizados **Tarea 3:** Analizar los Mecanismos de Clasificación de Lenguaje Natural.	**Resultado 1:** Revisión de la literatura.	Marco teórico del proyecto de investigación.
Evaluar los algoritmos que son empleados en la inteligencia artificial para seleccionar el más óptimo y aplicarlo en el presente proyecto de investigación.	**Tarea 1:** Determinar los Algoritmos empleados en la IA. **Tarea 2:** Evaluar los Algoritmos que se necesiten para La Clasificación de Lenguaje Natural.	**Resultado 1:** Algoritmo de Clasificación	Experimentación de algoritmos en la Web Semántica.
Desarrollar un algoritmo basado en inteligencia artificial para la clasificación mediante la aplicación de herramientas orientada a la Web Semántica.	**Tarea 1:** Seleccionar las herramientas. **Tarea 2:**Elaborar prototipos **Tarea 3:**Evaluar prototipos	**Resultado 1:** Herramientas seleccionadas. **Resultado 2:** Prototipos. **Resultado 3:** Informes de evaluaciones.	Algoritmos de clasificación.

8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA

8.1. Bases teóricas

8.1.1. Definición de lenguaje

Un lenguaje se puede definir de diferentes formas: desde el punto de vista funcional lingüístico se define como una función que expresa pensamientos y comunicaciones entre la gente. Esta función puede realizarse mediante signos escritos (escritura) o mediante Señales y vocales (voz). Desde un punto de vista formal se define como un conjunto de frases, que generalmente es infinito y se forma con combinaciones de elementos tomados de un conjunto (usualmente infinito) llamado alfabeto, respetando un conjunto de reglas de formación (sintácticas o gramaticales) y de sentido (semánticas). Además de las características fundamentales del lenguaje debe considerarse que sea funcional, es decir, el lenguaje debe permitirnos expresar nuestras ideas. El lenguaje será bueno en la medida en que sea fácil de leer, fácil de entender y fácil de modificar. El mismo ocurre en los lenguajes formales (Cortez Vásquez, Vega Huerta, & Pariona Quispe, 2009).

Definimos el lenguaje como un medio de comunicación formado por un sistema de signos arbitrarios codificados que nos permite representar la realidad en ausencia de ésta. Cada signo estará formado por un significante y un significado. Este sistema debe estar socialmente implantado y sólo a través de la interacción social se aprende. El lenguaje es, por tanto, una función mental que permite al hombre comunicarse con sus semejantes y consigo mismo (M. del P. Sánchez, 2010).

El lenguaje es un medio que nos permite comunicarnos a través de signos y códigos, escritos y hablados para poder expresar ideas, sentimientos hacia las demás personas desarrollando así un método de comunicación eficaz que forma parte de la vida cotidiana del ser humano.

8.1.2. Tipos de lenguajes

En la vida cotidiana los seres humanos producimos cientos de mensajes como respuesta a las necesidades de expresión. Esa producción se da básicamente a través de la palabra hablada, o del lenguaje verbal; sin embargo, junto con ella coexisten otros sistemas de comunicación con múltiples formas de expresión que enriquecen y diversifican las posibilidades de comunicación (Alvarez, 2016).

Comúnmente las diferentes formas que usamos para reforzar el contenido del mensaje van acompañadas de otros tipos de “lenguajes”, a los que corresponden subcódigos específicos y

por tanto múltiples que complementan y apoyan la intención del emisor; así el proceso de comunicación se enriquece con la utilización simultánea de otros códigos que funcionan a manera de lenguajes: la moda del vestido y sus códigos de diseño y color usados por algunos afroamericanos en Estados Unidos, acompañados con movimientos corporales pronunciados y ciertas entonaciones marcadas en el uso de expresiones verbales, los caracteriza e identifica como pertenecientes a ese grupo social.

A nivel mundial las personas utilizan diferentes tipos de lenguaje ya sean escritos o hablados teniendo así la misma finalidad de comunicar al emisor lo que se desea interpretar mediante lenguaje verbal y no verbal en el cual se utilizan movimientos corporales por gestos comunicando así al emisor lo mismo que se comunica cuando se utiliza el lenguaje verbal.

8.1.3. Lenguaje natural

Se llama lenguaje natural a las lenguas como son el castellano, alemán, etc., mediante lo cual las personas de una determinada comunidad lingüística se comunican. El lenguaje natural es el lenguaje que utilizamos los seres humanos puede ser hablado escrito o gestual y en si el lenguaje natural puede llegar a ser complejo y espontaneo. También se llama lenguaje ordinario. Sus características fundamentales son la riqueza expresiva y la flexibilidad (Villa, 2014).

Este lenguaje también se presta a las ambigüedades, las anfibologías, etc., por lo que es poco operativo para ser usado con rigor. Por esto Russell advirtió que la gran confusión que reina la filosofía, en el inadecuado planteamiento de los problemas y en su correlativa resolución se debe a la enorme riqueza y ambigüedad del lenguaje natural, que lo hace poco propicio para su uso instrumental también en la ciencia (Villa, 2014).

El lenguaje natural se conoce en ser reflexivo, esto es que se usa en virtud de una predisposición o sea en un acto de la voluntad. Se lo conoce por que nadie puede usarle sin que antes no le haya estudiado (Valverde, 2016).

8.1.4. Lengua y habla

La lengua no es función del sujeto hablante, sino el producto que el individuo registra pasivamente. Nunca supone premeditación y la reflexión no interviene en ella más que para la actividad de clasificar (Cortez Vásquez et al., 2009).

El habla es el acto individual de voluntad y de inteligencia, ya que supone composición premeditada haciendo uso de la lengua.

Cuando hablamos de la lengua y el habla, conviene distinguir:

A, Las combinaciones por lo que el sujeto hablante utiliza el código de la lengua con el objetivo de expresar sus ideas. B. El mecanismo psicofísico que le permite exteriorizar esas combinaciones.

Al separar la lengua del habla se separa a la vez:

a. Lo que es social de lo que es individual

b. Lo que es esencial de lo que es accesorio

La lengua es el idioma que utiliza cada persona para comunicar sentimientos y emociones a la persona que le rodea logrando así correcta recepción de lo que se quiere transmitir al emisor. Al combinarse el habla con la lengua obtenemos un método de comunicación eficaz que es el lenguaje ayudándonos a interpretar diferentes tipos de lenguaje logrando así una correcta interpretación del habla.

8.1.5. Lenguaje de programación

Un lenguaje de programación es un lenguaje formal definido como un conjunto de elementos (componentes léxicos) organizados a través de constructores (reglas gramaticales) que permiten escribir un programa y que éste sea entendido por el computador y pueda ser trasladado a computadores similares para su funcionamiento en otros sistemas. Un programa es una secuencia de instrucciones ordenadas correctamente que permiten realizar una tarea o trabajo específico. Un lenguaje de programación se basa en dos elementos muy importantes:

• Sintaxis: que se encarga del orden correcto de los componentes léxicos.

• Semántica: se encarga de que cada "oración" del lenguaje de programación utilizado tenga un significado correcto (Cortez Vásquez et al., 2009).

Entendemos como lenguaje de programación a dos elementos que se conjugan que son la sintaxis y la semántica permitiendo así formar un leguaje formal con códigos que se pueden utilizar en diferentes sistemas de computadores y que siguen instrucciones permitiendo ejecutar tareas.

Un lenguaje de programación consiste en un conjunto de órdenes o comando que describen el proceso deseado. Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que se combinan para formar los programas de cómputo. Los lenguajes de programación no son aplicaciones, sino herramientas que permiten construir y adecuar aplicaciones (Introducción, 2009).

8.1.5.1. Tipos de lenguaje de programación

El lenguaje máquina

Hasta el momento se ha visto la forma en que es posible representar de forma adecuada datos en un ordenador; dichos datos se almacenarán en la memoria y se trabajará con ellos en la Unidad Aritmética. Sabemos que la Unidad de Control también recupera instrucciones de la memoria y realiza acciones más sencillas según indique cada instrucción (Wall, 2016.).

Lenguaje ensamblador

Aunque aún es bastante críptico resulta más sencillo programar en un lenguaje de este tipo que en código máquina. Los ensambladores fueron desarrollados de forma muy temprana y recibieron este nombre porque las instrucciones básicas del lenguaje ensamblador eran en realidad pequeños programas escritos directamente en código máquina; así cuando un programador debía escribir un nuevo programa con ese lenguaje en realidad estaba "ensamblando" código máquina reutilizable (Wall, 2016.).

Al ser un lenguaje de muy bajo nivel, está muy relacionado con el hardware y la arquitectura del microprocesador. Por eso, el estudio del lenguaje ensamblador y del microprocesador va de la mano.

Lo ligado que el ensamblador está al microprocesador es, a la vez, su punto fuerte y su debilidad. Esta característica hace que programar en ensamblador genere un código muy óptimo y eficiente, teniendo el programador la decisión de todo. Por el contrario, aprender ensamblador para un microprocesador no asegura que se pueda programar para otra familia de procesadores, ya que el lenguaje y sus características cambian (Irvine, 2008).

Lenguajes de alto nivel

Aunque los lenguajes ensambladores supusieron una mejora respecto a la programación directamente en código máquina seguían siendo engorrosos, excesivamente alejados de la forma de pensar humana y específicos de cada tipo de ordenador por lo que era muy difícil, por no decir imposible, transportar un algoritmo de un ordenador a otro (Operacional, 2016).

Estos lenguajes son los más utilizados por los programadores. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina y ensamblador. Un programa escrito en lenguaje de alto nivel es independiente de la

máquina. Los programas escritos en lenguaje de alto nivel pueden ser ejecutados con poca o ninguna modificación en diferentes tipos de computadoras.

Son lenguajes de programación en los que las instrucciones enviadas para que el ordenador ejecute ciertas órdenes son similares al lenguaje humano. Dado que el ordenador no es capaz de reconocer estas órdenes, es necesario el uso de un intérprete que traduzca el lenguaje de alto nivel a un lenguaje de bajo nivel que el sistema pueda entender (Soraya et al., 2018).

8.1.6. Procesamiento de lenguaje natural

El procesamiento del lenguaje natural es una disciplina de la Inteligencia Artificial, la cual se ocupa de la formulación e investigación de mecanismos computacionales para la comunicación entre personas y máquinas, a través del uso de los Lenguajes Naturales, es decir en escritos, hablados o signados (Cortez Vásquez et al., 2009).

El procesamiento del Lenguaje Natural es una disciplina que relaciona directamente la informática con la lingüística. La misma persigue como objetivo, poder conseguir que el lenguaje coloquial (el lenguaje de uso cotidiano de todos nosotros) pueda ser utilizado como una entrada en un sistema informático (Zamoszczyk, De Luca, Ruiz Martínez, & Iturbide, 2018).

El procesamiento de lenguaje natural es en sí una forma de transmitir información entre humanos y computadores por medio de los diferentes tipos de lenguaje natural permitiendo así que la computadora interprete órdenes complejas y sencillas.

Es importante poder destacar entre tres tipos de objetivos que persigue el procesamiento de lenguaje natural:

1. Interfaces en lenguaje natural: Lograr la comunicación con distintos dispositivos a través del lenguaje natural.
2. Procesamiento de textos: Se refiere a lograr extraer datos significativos de textos escritos en lenguaje natural, a efectos de realizar el procesamiento de los mismos. Esto intenta abordar un inconveniente del mundo actual, en donde la mayor cantidad de información se encuentra almacenada en forma de texto. Esto implica que la información de los mismos no puede ser procesada de forma directa. Ejemplos de esto puede ser bases de datos relacionales o registros de transacciones bancarias.
3. Traducción automática: Es el objetivo original del PLN, que consta del análisis y tratamiento de lenguaje natural por medio de la utilización de herramientas tanto lingüísticas como informáticas.

8.1.6.1. Técnicas de procesamiento de lenguaje natural

Para lograr el procesamiento de lenguaje natural, existen un conjunto de técnicas mediante las cuales se extrae del texto información determinada. A continuación, se describirán algunas de las técnicas más comunes utilizadas por los diferentes sistemas NLP para procesar texto escrito en lenguaje natural (Ramos & Velez, 2016).

Minería de datos

La minería de datos proporciona herramientas poderosas para descubrir patrones ocultos y relaciones en datos estructurados. Este proceso asume que los datos ya se encuentran almacenados en un formato estructurado. Por esta razón su pre-procesamiento consiste en la limpieza y normalización de los datos y la generación de numerosos enlaces entre las tablas de las bases de datos (J, 2020).

La minería de datos nos ayuda almacenar datos estructurados en un formato ya estructurado para una mejor organización de las tablas de las base de datos permitiendo así encontrar información de un manera eficaz y en su proceso los datos se normalizan al momento de presentar la información generada.

Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior. Además de la etapa de análisis en bruto, que involucra aspectos de bases de datos y de gestión de datos, de procesamiento de datos, del modelo y de las consideraciones de inferencia, de métricas de Intereses, de consideraciones de la Teoría de la complejidad computacional, de post-procesamiento de las estructuras descubiertas, de la visualización y de la actualización en línea (Discovery, 2016).

Recuperación y extracción de información

La recuperación de información RI, es el proceso de encontrar en un repositorio grande de datos, como son usualmente los documentos de naturaleza no estructurada que son conocidas como texto o semiestructurada de páginas Web, que satisfaga una necesidad de información (Manning, Raghavan, & Schütze, 2011).

Los datos no estructurados no tienen un esquema claro, no están listos para procesar y son lo opuesto a los datos con un esquema estructurado como los que se encuentran en base de datos. Los datos semiestructurados están en documentos con marcas explicitas como el código HTML.

La información encontrada debe ser pertinente y relevante, la relevancia es la medida de como una pregunta se ajusta a un documento y la pertinencia es la medida de como un documento se ajusta a una necesidad informativa (Saini, Singh, & Kumar, 2015).

Las de recuperación de información involucran la transformación del texto en representaciones adecuadas de acuerdo a modelos específicos que cumplan con los propósitos de la clasificación de los lenguajes que se utilicen para la investigación (Wiedau & Harwardt, 2011).

En la dimensión de bases matemáticas, el texto puede ser representado como: conjunto de palabras o frases en donde las coincidencias se logran realizando operaciones de algebra booleana; modelos algebraicos que introducen parámetros e índices para recuperar información con metadatos, calificar y clasificar todos ellos en respuesta a una consulta, lo que lleva a modelos en espacio vectoriales, matriciales o agrupamientos irregulares; modelos probabilísticos que enfocan la solución de los problemas de búsqueda desde el punto de vista probabilístico, aplicando algoritmos (Manning, Raghavan, & Schütze, 2011).

8.1.7. Algoritmo

Un algoritmo se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

Para realizar un programa es conveniente el diseño o definición previa del algoritmo. El diseño de algoritmo requiere creatividad y conocimientos profundos de la técnica de programación. Los algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje diferente de programación. El algoritmo es la infraestructura de cualquier solución, escrita en cualquier lenguaje de programación (D. Sánchez, 2014).

"Un Algoritmo es una secuencia de operaciones detalladas y no ambiguas, que al ejecutarse paso a paso, conducen a la solución de un problema" (D. Sánchez, 2014). En otras palabras es un conjunto de reglas para resolver una cierta clase de problema.

8.1.8. Algoritmos de clasificación

Los clasificadores binarios indican si un objeto es o no miembro de una clase. A veces se combinan para obtener una clasificación multiclases. Dependiendo del algoritmo, la salida será una sola clase o un número de clases con pesos que describen la probabilidad de que el objeto sea miembro de una clase determinada, como es el caso del algoritmo Mahout Bayes.

A veces los clasificadores jerárquicos están organizados en estructuras tipo árboles. En estos casos un documento que pertenece a la clase A, que tiene como hijos B y C, será evaluado con los clasificadores entrenados para reconocer si está en la clase B o C. Si coincide con B, será evaluado para los hijos de esa clase y así sucesivamente hasta llegar al último nivel del árbol (J, 2020).

8.1.9. Tipos de algoritmos

Cualitativos: Son aquellos en los que se describen los pasos utilizando palabras.

Cuantitativos: Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.

8.1.10. Características de los algoritmos

- Preciso. Definirse de manera rigurosa, sin dar lugar a ambigüedades.
- Definido. Si se sigue un algoritmo dos veces, se obtendrá el mismo resultado.
- Finito. Debe terminar en algún momento.
- Puede tener cero o más elementos de entrada.

Debe producir un resultado. Los datos de salida serán los resultados de efectuar las instrucciones.

8.1.11. Tipos de lenguajes algorítmicos

• Gráficos: Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).

• No Gráficos: Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

Un algoritmo puede ser expresado de las siguientes formas.

a) Lenguaje Natural: el uso de términos del lenguaje natural, es una forma de representar un algoritmo.

b) Lenguaje Simbólico: es otra forma de representación de un algoritmo, que además permite una introducción a la programación estructural.

c) Lenguaje Gráfico: es una forma de escribir una secuencia de pasos en forma de diagrama, en la práctica se denomina Diagramas de Flujo.

Una receta de un plato de cocina se puede expresar en español, inglés o francés pero cualquiera sea el lenguaje los pasos para la elaboración del plato se realizarán sin importar el cocinero.

8.1.12. Python

Python es un lenguaje de programación por su claridad y elegancia, básicamente representa la filosofía del lenguaje. Python es un lenguaje multiparadigma en el que coexisten de forma nativa aspectos imperativos, funcionales y orientados a objetos. Estos paradigmas están muy desacoplados lo que permite que la entrada al lenguaje se pueda hacer de forma progresiva (Troyano, 2018).

Una de sus características principales de este leguaje es que es orientado a objetos y es un tipo de paradigma de programación que está basado de clases de programación y ejemplos de clases llamadas objetos.

Python cuenta con una amplia biblioteca estándar con herramientas para resolver muchas tareas de utilidad como, por ejemplo la lectura de ficheros en distintos formatos (JSON, XML, CSV), el desarrollo simple de interfaces graficas la conexión con base de datos.

Es libre y gratuito, funciona sobre todas la plataformas, apareció en 1990 y posee varias implementaciones, entre ellas CPython, Python, IronPython y PyPy (Chazallet, 2016).

La forma de resolver con eficiencia varias tareas hace de este tipo de lenguaje capaz de ayudar a resolver funciones más complejas con mayor facilidad que en otros tipos de lenguaje.

Es importante saber que Python se programa con un editor de texto plano el cual sería el bloc de notas o gEdit en Linux, también tenemos la posibilidad de utilizar varios programas como el Notepad ++ o Geany, es importante saber que estos últimos nos facilitan el trabajo al colorear el código para que se distinga mejor varias cosas y así no confundirlas. (feNiX10ist, 2014).

Es un lenguaje que por la facilidad de su sintaxis es perfecto para cualquier programador principiante que este decidido a aprender por su propia cuenta.

8.1.13. Spyder

Spyder es un MATLAB-like IDE para la computación científica con python. Tiene muchas ventajas de un entorno IDE tradicional, por ejemplo, que todo, desde la edición de código, la ejecución y la depuración es llevado a cabo en un solo entorno, y el trabajo sobre diferentes cálculos se puede organizar como proyectos en el Entorno IDE (Applications, 2016).

Se comprende que Spyder es un entorno IDE que se utiliza con Python y en el cual su principal ventaja es que al empezar a editar el código, ejecutar y depurar la aplicación se lo realiza en un solo entorno y así mismo se puede realizar varios proyectos.

Algunas ventajas de Spyder:

• Potente editor de código, con sintaxis de alta iluminación, introspección de código dinámico e integración con el depurador de python.
• Explorador de variables, símbolo del sistema de IPython.
• Documentación integrada y ayuda.

8.1.14. Pseudocódigo

El pseudocódigo es un lenguaje de especificación de algoritmos (no de programación) basado en un sistema notaciones, con estructuras sintácticas y semánticas, similares a los lenguajes procedurales, aunque menos formales que las de éstos, por lo que no puede ser ejecutado directamente por un computador (Valencia, 2018).
El pseudocódigo utiliza para representar las sucesivas acciones, palabras reservadas - similares a sus homónimas en los lenguajes de programación-, tales como start, end, stop, if-then-else, while-do, repeat-until, (inicio, fin, parar, si-entonces- sino, mientras-hacer, repetir-hasta), etc.

8.1.15. Diagrama de flujo

Los diagramas de flujo son comunes en varios dominios técnicos y se usan para poner en orden los pasos a seguir o las acciones a realizar. Su principal ventaja es que tienen la capacidad de presentar la información con gran claridad, además de que se necesitan relativamente pocos conocimientos previos para entender los procesos y/o el objeto del modelado (Mihaela Juganaru Mathieu, 2013).
Los diagramas de flujo son pasos ordenados que se utilizan para realizar acciones permitiendo así llevar un orden adecuado y presentar la información precisa para que no existan errores en la respuesta que emite al usuario.

8.1.16. Clasificación

Los sistemas de clasificación tienen la función de usar datos estructurados en forma de instancias o ejemplos con distintos tipos de atributos como unidad de información para aprender de ellos y ser capaces de saber clasificar correctamente futuras instancias o ejemplos. En la actualidad existen sistemas de clasificación muy buenos, pero por lo general no están pensados en ser utilizados en entornos BigData (Yánez, 2010).
Ante esta creciente cantidad de dispositivos conectados a redes que generan y procesan gran cantidad de información los problemas de computación ya no se pueden resolver con un solo ordenado. En los últimos años han ido apareciendo tecnologías como la automatización de procesos, internet de la cosas, entre otras. Ahora necesitan ser resueltos por ordenadores más

potentes dándose así un incremento de la popularidad del BigData. Debiendo a esta creciente popularidad han aparecido distintas tecnologías para resolver los problemas BigData (Ortiz, 2018). La clasificación es uno de los temas más estudiados dentro del campo de la minería de datos y el aprendizaje automático, la razón de esto es que hay una gran cantidad de problemas en diferentes áreas como seguridad, medicina o finanzas que necesitan clasificar muchos de los datos que manejan. El objetivo de los clasificadores es el construir un modelo o clasificador a partir de un conjunto de ejemplos ya clasificados que permitan clasificar nuevos ejemplos no vistos anteriormente.

El problema al que nos enfrentamos en este proyecto es el de un problema de clasificación supervisada. Esta clase de problemas se dividen en dos fases principales, a continuación se detallan:

1. El algoritmo de clasificación usará una serie de ejemplos llamados ejemplos de entrenamiento que ya estarán clasificados para aprender de ellos. Usando los datos de entrenamiento creará una serie de reglas o métodos de decisión para clasificar correctamente los ejemplos de entrenamiento.
2. Una vez creado el algoritmo de clasificación se pasara a clasificar ejemplos para ver qué tan bueno es el clasificador que se ha desarrollado.

8.1.17. Estado del arte

En esta sección se describe el estado actual de los diferentes algoritmos de clasificación que componen la parte central de este proyecto de investigación.

Existen diversos algoritmos para realizar la clasificación del lenguaje natural entre ellos se encuentran:

- Algoritmos basados en Random Forest.
- Algoritmos k-NN.

8.1.17.1. Random Forest

Random Forest fue implementado en el lenguaje de programación R por Baccini et al. (2008) para la estimación de biomasa aérea usando sensores remotos de resolución gruesa, y ya ha sido probado en áreas extensas de África para la estimación de la biomasa aérea. Complementariamente se realizó la calibración del algoritmo mediante su aplicación repetida,

analizando los efectos de cada una de las diferentes variables en la explicación del modelo y la posterior calibración de los diferentes parámetros del modelo.

Supuestos:

• Las variables topográficas, climáticas y basadas en sensores remotos son útiles y suficientes para explicar la biomasa aérea en el país.

• Se parte de la premisa de que existe una sensibilidad de la reflectancia óptica a variaciones en la estructura del dosel de la vegetación (Goetz et al., 2009).

• Las diferencias en técnicas de pre-procesamiento y procesamiento utilizadas no están afectando los resultados (Foody et al., 2003), y su geolocalización corresponde al pixel adecuado. • Las parcelas se encuentran bien ubicadas espacialmente.

• Las diferencias en biomasa entre el año de toma de los datos de campo y el año de la imágenes no son considerables(En & Forests, 2010).

Random Forest es una técnica de adherencia desarrollada por Leo Breiman, que mejora la precisión en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual. Esta aleatorización puede introducirse en la partición del espacio (construcción del árbol), así como en la muestra de entrenamiento. Las distintas variantes de Random Forest consisten en cómo se incorpora la aleatoriedad en la construcción (Illanes, 2010).

Este método ha permitido mejorar la clasificación de una serie de datos aleatoriamente ayudando a construir un clasificador individual, pudiendo analizar el método de manera directa sin modificar las variables que lo componen. Al introducir los datos aleatoriamente tanto en la construcción del árbol como en la muestra de entrenamiento nos permitió trabajar con cientos de variables independientes sin excluir ninguna haciendo de este método una opción manejable para el usuario.

El método que usa Random Forest para generar diferentes modelos a partir de los datos se conoce como boostrap aggregatting, o bagging. Este método genera una serie de Dataset de entrenamiento a partir de los datos originales mediante boostrap sampling, para luego ser usados para entrenar un solo modelo con cada uno de los datos de entrenamiento. En el caso de Random Forest, como ya hemos dicho antes, el algoritmo usado para el bagging es un Árbol de Decisión. Este algoritmo utiliza una serie de reglas sencillas (basadas en estrategias Divide y Vencerás) para clasificar a los diferentes individuos. Este algoritmo combina versatilidad con potencia en una sola aproximación al Machine Learning. Como el ensamble solo usa un pequeño subconjunto aleatorio de los datos, Random Forest puede trabajar con Dataset increíblemente grandes, donde la "maldición de la dimensión" haría fracasar a otros algoritmos (Utrera, 2017).

El algoritmo Random Forest es un método de estimación combinado, donde los resultados de la estimación se construyen a partir de los resultados obtenidos mediante el cálculo de n árboles donde los predictores son incluidos al azar.

Es un método complejo con ventajas e inconvenientes respecto a los árboles de clasificación simples para ello mencionamos las ventajas e Inconvenientes que posee el algoritmo Random Forest (Sánchez, 2016).

Ventajas

- Es uno de los algoritmos de aprendizaje más precisos
- Se ejecuta eficientemente en grandes bases de datos
- Permite trabajar con cientos de variables independientes sin excluir ninguna
- Determina la importancia en la clasificación de cada variable
- Recupera eficazmente los valores perdidos de un Dataset (missings)
- Permite evaluar la ganancia en clasificación obtenida a medida que incrementamos el número de árboles generados en el modelo.

Inconvenientes

- A diferencia de los árboles de decisión, la clasificación hecha por Random Forest es difícil de interpretar.
- Favorece las variables categóricas que tienen un mayor número de niveles por encima de aquéllas que tienen un número de categoría más reducido. Comprometiendo la fiabilidad del modelo para este tipo de datos.
- Favorece los grupos más pequeños cuando las variables están correlacionadas.
- Random Forest sobre ajusta en ciertos grupos de datos con tareas de clasificación/regresión ruidosas.

8.1.17.2. K-nearest neighbours (K-NN)

Una forma práctica y de fácil aplicación para predecir o clasificar un nuevo dato, fundamentado en observaciones conocidas o pasadas. Esta metodología se basa, simplemente en “recordar” todos los ejemplos que se vieron en la etapa de entrenamiento. Cuando un nuevo dato se presenta al sistema de aprendizaje, este se clasifica según el comportamiento del dato más cercano (Mora-florez & Barrera-cárdenas, 2008).

El algoritmo de los k vecinos más cercanos es catalogado como clasificador basado en instancias. Para clasificar, compara las instancias no vistas con aquellas etiquetadas del conjunto de entrenamiento utilizando una función de similitud. Generalmente la similitud es medida mediante una función de distancia. A pesar de su simplicidad, el clasificador kNN es uno de los diez algoritmos de clasificación más relevantes Para este proyecto se utilizó el método KNN el cual nos permite clasificar un nuevo dato con facilidad en base a observaciones conocidas detallando con una predicción similar al del Random Forest ya sea el caso(Maillo, Garc, Herrera, & Triguero, 2016).

Por la aplicación que ha tenido, se han desarrollado diferentes alternativas de clasificadores basados en la regla k-NN. Algunos de ellos, aparecen en el estado del arte como clasificadores rápidos k-NN. Estos algoritmos han sido desarrollados para procesar grandes conjuntos de datos, aplicados sobre diversos problemas, como el análisis de valores en línea, control de tráfico aéreo, detección de intrusos, entre otros. Sin embargo, varios de estos problemas, están definidos por conjuntos de datos con alta dimensionalidad, donde la función de comparación puede resultar computacionalmente muy costosa, por lo que se recomienda reducir el número de comparaciones realizadas con los objetos de entrenamiento (Sánchez-Díaz et al., 2013).

Para resolver diversos problemas de clasificación, se requiere procesar conjuntos de datos de entrenamiento muy grandes, los cuales, en ocasiones no es factible almacenarlos en la memoria principal de la computadora (Sánchez-Díaz et al., 2013).

Este método consta de dos fases:

1. Fase de entrenamiento
2. Fase de clasificación

Las mismas que ayudan a mejorar el proceso de clasificación.

Un método de aproximación simple no paramétrica es el basado en la regla del vecino más cercano, que consiste en estimar el valor de un dato desconocido a partir de las características del dato más próximo, según una medida de similitud o distancia. Esta regla tiene propiedades estadísticas bien establecidas y facilidad de aplicación a sistemas reales. El método del vecino más cercano se puede extender utilizando no uno, sino un conjunto de datos más cercanos para predecir el valor de los nuevos datos, en lo que se conoce como los k-vecinos más cercanos (k-NN o k-Nearest Neighbors). Al considerar más de un vecino, se brinda inmunidad ante ruido y se suaviza la curva de estimación.

El método de los k-vecinos más cercanos se adapta fácilmente a la regresión de funciones con valores continuos. El algoritmo asume que todos los datos pertenecen a Rp , y mediante una medida de distancia en ese espacio se determinan los k datos más cercanos al nuevo dato porque para aproximar una función f: Rp → R, a partir de los k valores ya seleccionados (Morales España, Mora Flórez, & Vargas Torres, 2008).

Dificultad

La principal dificultad de este método consiste en determinar el valor de k, ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo a la mayoría (y no al parecido), y si el valor es pequeño puede haber imprecisión en la clasificación a causa de los pocos datos seleccionados como instancias de comparación (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007).

Ventajas

Las ventajas de esta metodología es que no necesita una adaptación para clasificación de más de dos clases y que es sencillo de implementar, por otro lado como inconvenientes encontramos que es muy sensible a datos irrelevantes y la dimensionalidad de los mismos, es sensible al ruido y lento en el caso que tengamos muchos datos de entrenamiento.

Es adecuado indicar también la importancia de la elección de la función distancia para la elección de los vecinos al ejemplo en observación, en la que normalmente se utiliza Euclides. Aun así existen otros métodos mejorados para la elección del vecino en este algoritmo usando la indexación del vector distancia (Ruiz, 2018).

8.1.18. Web semántica

La web se ha convertido en un enorme repositorio de información textual semi-estructurada que abarca casi todas las áreas del conocimiento humano. La literatura universal, el conocimiento científico y la prensa digital son solo algunos de los contenidos web que consumen a diario millones de personas alrededor del mundo.
La web es ahora mucho más social. Solo en 2010 fueron creados aproximadamente 152 millones de blogs en Internet, registradas 175 millones de personas en Twetter1 y 600 millones en Facebook (Hidalgo-Delgado & Rodríguez-Puente, 2013).

La web semántica no pretende sustituir la web actual, sino que es una extensión en la que la información tiene un significado bien definido posibilitando a los humanos y las computadoras trabajar en cooperación, La web semántica puede ser considerada como una evolución natural de la web actual en la que los datos son presentados en un formato único procesable por las computadoras. En este sentido, la World Wide Web Consortium (W3C) de conjunto con investigadores de todo el mundo, han venido trabajando en la última década en la definición de varios estándares, muchos de los cuales han sido utilizados en el desarrollo de múltiples aplicaciones (Hidalgo-Delgado & Rodríguez-Puente, 2013).

La web semántica consiste en un nuevo paradigma web para acceder, buscar, compartir y gestionar información a través de la combinación de tecnologías y de estructuras de gestión del conocimiento. El concepto de web semántica proporciona herramientas para el almacenamiento, intercambio y consulta de esta información mediante el desarrollo y la inclusión de metadatos y ontologías del cuerpo de conocimiento. La estructura de los datos que proporciona permite que sea consultada automáticamente por usuarios humanos o sistemas informáticos, mejorando su interoperabilidad (Mora et al., 2016).

La web semántica ha sido definida de diferentes maneras: como una visión utópica, como una evolución de la web, como una web de datos, o simplemente como un cambio de paradigma en el uso de la web. Pero por encima de todo, la web semántica ha inspirado y comprometido a una extensa comunidad a crear tecnologías y aplicaciones semánticas innovadoras. La web semántica provee una estructura común que permite que los datos sean compartidos y reutilizados, cruzando los límites establecidos por aplicaciones, empresas y comunidades (Perez Castillo, Diaz Hernandez, & Rincon Mosquera, 2015).

Básicamente, el término "web semántica" hace referencia al hecho de poder añadir metadatos semánticos a la World Wide Web. Esto significa que lo que se pretende es añadir información adicional que describe el contenido, el significado y la relación de los datos, y esto se hace por medio de metalenguajes.

A los metalenguajes se les añaden los estándares de representación XML, XML Schema, RDF, RDF Schema y OWL. Con ellos se consigue introducir esa información adicional de la que hablábamos en el párrafo anterior, y a grandes rasgos cada uno de ellos se encarga de la sintaxis de dicha información, la estructura de la misma, el modelo de datos, los recursos y las relaciones que se pueden establecer entre ellos (Agudo, 2015).

8.2. Antecedentes

Hoy en día la Ingeniería en Sistemas permite transformar las necesidades de las personas en proyectos de investigación que ayudan a incluir la inteligencia artificial en la sociedad, integrando actividades en medios apropiados para desarrollar un enfoque interdisciplinario que permite estudiar y comprender la realidad de un propósito de implementar algoritmos mediante herramientas de inteligencia artificial que ayuden a integrar otras disciplinas, para dar solución a un problema encargándose del diseño, programación y obteniendo grandes resultados para recolectar, almacenar, procesar y comunicar datos e información con el objetivo de mantener una gestión eficiente en la organización.

Es de gran importancia saber que un software es la parte lógica donde programas se ponen en funcionamiento en el ordenador que está capacitado para interpretar las instrucciones que reciben a través de los distintos componentes que le autorizan para realizar múltiples tareas.

8.3. Tendencias y principales referentes

Esta sección proporciona el conocimiento básico del algoritmo Fuzzy kNN, el Hybrid Spill Tree y las tecnologías big data utilizada. El algoritmo Fuzzy kNN se propone como una mejora del algoritmo kNN, llegando a mejorarlo en términos de precisión para la mayoría de los problemas de clasificación. Necesita una etapa de pre-computo en el conjunto de entrenamiento, que calcula el grado de pertenencia a la clase. Posteriormente, calcula kNN para cada instancia no vista y decide la clase predicha con el grado de pertenencia. Fuzzy kNN posee dos etapas: cálculo de pertenencia y clasificación. La primera etapa calcula los k vecinos más cercanos para cada instancia del CE, manteniendo un esquema leave-one-out seleccionando las k instancias con una distancia menor (Maillo, Garc, Herrera, & Triguero, 2016).

8.3.1. Principales referentes teóricos

En este punto adjuntamos referentes proyectos similares al proyecto de investigación en desarrollo acerca del algoritmo de clasificación mediante lenguaje natural basado en web semántica, los cuales tomamos como referencia para el proyecto, los mismos que aportaron como medio para la realización del mismo.

La estrategia de regresión basada en el método de los k vecinos más próximos para la estimación de la distancia de falla en sistemas radiales. Germán Morales España, Juan Mora Flórez,

Herman Vargas Torres (2008) Rev. Fac. Ing. Univ. Antioquia N.° 45 pp. 100-108. Septiembre, 2008.

Se presenta una estrategia de regresión para estimación de la distancia de falla en sistemas de potencia radiales, empleando la técnica de los k-Vecinos más próximos (k-NN). Esta propuesta de localización de fallas utiliza las medidas de la componente fundamental de tensión y de corriente disponibles en la subestación, no depende del modelo del sistema de potencia y se adapta a las características particulares de los sistemas radiales. La continuidad y por tanto la calidad del servicio de energía eléctrica se ve afectada por las fallas. Algunas técnicas relevantes aplicables a sistemas de radiales han sido planteadas para la localización de fallas. Utilizando la componente fundamental de la corriente y tensión en pre falla y falla medidas en la subestación, estiman la sección de línea fallada con la comparación de la impedancia obtenida a partir del modelo impuesto por el método y la impedancia equivalente calculadora (Quezada Lucio, 2018).

Se propone el uso de la técnica de aprendizaje supervisado de regresión, conocida como los k vecinos más próximos (k-NN). Esta técnica se aplica a la estimación de la distancia de falla, considerando las características fundamentales de los sistemas radiales, sin depender del modelo del sistema. Inicialmente se presentan los fundamentos básicos de la técnica del vecino más próximo aplicada a la regresión. Luego se discute la aplicación de los k-NN a la localización de fallas (Quezada Lucio, 2018).

Los clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más próximo en Reconocimiento de Patrones. Francisco Moreno Seco (2004). La clasificación no paramétricos más simples es el clasificador basado en la regla del vecino más próximo, que consiste en clasificar el objeto desconocido en la clase de su vecino más próximo según la disimilitud o distancia. Prácticamente todos estos algoritmos se pueden extender fácilmente para encontrar los k vecinos más próximos (Quezada Lucio, 2018).

El objetivo consiste en el estudio de un algoritmo rápido de búsqueda del vecino más próximo, el LAESA, para obtener una disminución en la tasa de error sin coste adicional, utilizando para ello k vecinos; una vez abordado este objetivo, las ideas aplicadas a este algoritmo se han extendido a otros algoritmos con resultados similares e incluso mejores, y se han generalizado en una nueva regla de clasificación: la regla de los k vecinos más próximos. La idea básica de esta regla es utilizar para la clasificación los k candidatos más próximos de entre los seleccionados por el algoritmo de búsqueda. Una buena parte del trabajo de esta tesis se ha dedicado a explorar las posibilidades y el comportamiento de esta regla, que produce tasas de

acierto en la clasificación similares a las de la regla de los k vecinos más próximos pero con el coste computacional de la regla del vecino más próximo (Quezada Lucio, 2018).

9. HIPÓTESIS

El algoritmo basado en técnicas de inteligencia artificial permitirá clasificar aspectos de lenguaje natural en una búsqueda basada en la web semántica.

10. METODOLOGÍAS Y DISEÑO EXPERIMENTAL

10.1. Métodos y materiales

Para el desarrollo del presente proyecto de investigación se aplicó la investigación bibliográfica, mediante etapas de procesos investigativos que se involucran con el estudio de nuestro tema, toda información se extrajeron por medios de revistas, tesis, libros y artículos científicos, puesto que esto proporciona un conocimiento mayor en el ámbito académico o investigativo. La presente investigación se constituye de técnicas y procedimientos que constituyen los medios instrumentales para su iniciación, teniendo como técnica principal la lectura pues a través de ella se adquiere el conocimiento y lograr una práctica hacia el desarrollo del proyecto de investigación. Como se puede ver en los objetivos, nuestro trabajo tiene un componente principalmente prototipo, experimental y otra fundamentalmente práctica. Aunque la práctica es principalmente de apoyo a la otra, creemos que por el esfuerzo y la generalidad del método desarrollado, se debe considerar un objetivo por sí mismo. Para alcanzar el primer objetivo, tratamos de utilizar un enfoque científico, con un método riguroso para validar nuestras conclusiones y un esfuerzo de investigación para determinar la relevancia de nuestros esfuerzos. El segundo objetivo se desarrolló con métodos más cercanos a la ingeniería del software. Utilizando conocimientos previos para realizar un diseño coherente que nos permitiera desarrollar un Algoritmo flexible que se pudiera ir adaptando a las necesidades del proyecto según surgían.

10.2. Prototipado

El modelo de prototipos permite que todo sistema o algunas de sus partes se construyan rápidamente para comprender con facilidad y aclarar ciertos aspectos en los que aseguren que el desarrollador, el usuario y el cliente estén de acuerdo en lo que se necesita así como también la solución que se propone a dicha necesidad y así minimizar el riesgo y la incertidumbre en el desarrollo, este modelo se encarga del desarrollo de diseños para que estos sean analizados y

prescindir de ellos a medida de que se consoliden nuevas especificaciones (Pérez Lozada & Falcón, 2017).

El uso de prototipos se centra en la idea de ayudar a comprender los requisitos que plantea el usuario, sobre todo si este no tiene una idea concluida de lo que realmente desea. Además puede utilizarse cuando el ingeniero en software tiene dudas acerca de la viabilidad de la solución (Pérez Lozada & Falcón, 2017).

10.3. Tipos de modelo

10.3.1. Modelo de prototipos rápidos

Metodología de diseño que desarrolla rápidamente nuevos diseños, los evalúa y prescinde del prototipo cuando el próximo diseño es desarrollado mediante un nuevo prototipo.

10.3.2. Modelo de prototipos reutilizables

También conocido como "Evolutionary Prototyping"; no se pierde el esfuerzo efectuado en la construcción del prototipo pues sus partes o el conjunto pueden ser utilizados para construir el producto real. Mayormente es utilizado en el desarrollo de software, si bien determinados productos de hardware pueden hacer uso del prototipo como la base del diseño de moldes en la fabricación con plásticos o en el diseño de carrocerías de automóviles.

10.4. Tipos de prototipos

10.4.1. Prototipo desechable

Nos sirve para eliminar dudas sobre lo que realmente se quiere en el proyecto además para desarrollar el algoritmo que más nos convenga.

10.5. Experimentación

Por experimentación entendemos analizar el efecto que una o varias variables independientes producen sobre otra variable dependiente. Para ello es necesario controlar y neutralizar la influencia que otros factores puedan ejercer sobre el variable objeto de estudio; con este fin, la experimentación se traspasa a universos aleatorios en los que el control es aleatorio y los resultados obtenidos se estudian a través del análisis de la varianza.

El método de experimentación consiste en reproducir fenómenos a voluntad del investigador (Marqués, 2015).

La principal dificultad de la experimentación consiste en realizar la prueba en las mismas circunstancias que en la realidad, así como en aislar los resultados obtenidos, debido a la variación producida respecto a otras variables no controladas en el experimento (Marqués, 2015).

10.6. Diseño experimental

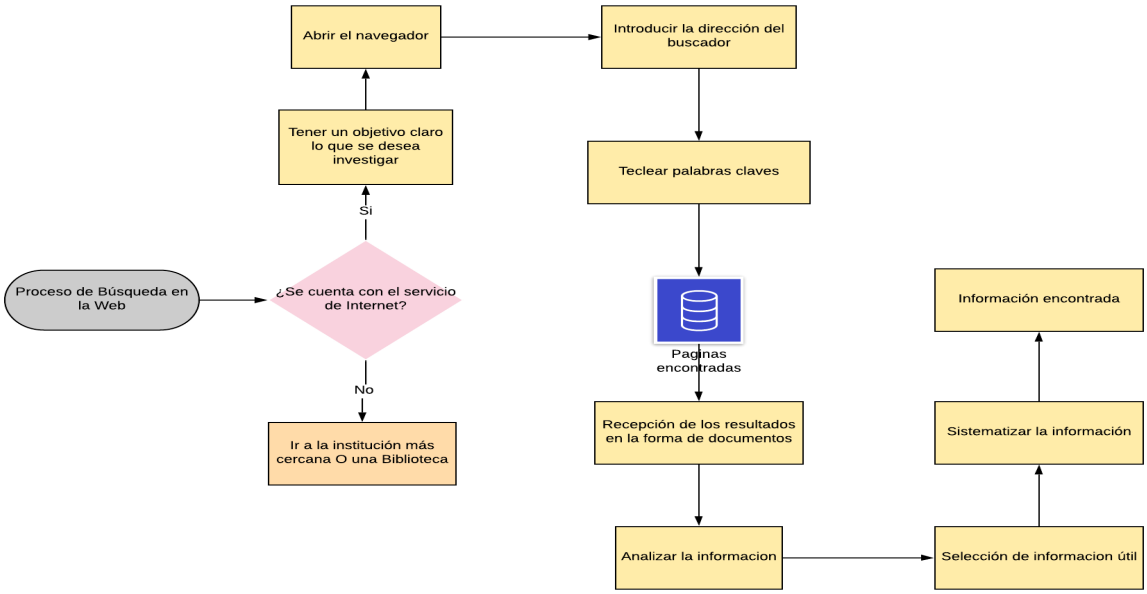
10.6.1. Proceso de búsqueda web

El proceso de búsqueda en la web se lo realizo a través de pasos, el primer paso es preguntar ¿Se cuenta con él servicio de internet? Si la respuesta fue No asumíamos como resultado Ir a la institución más cercana o una biblioteca, caso contrario si la respuesta fue Si continuaba con el siguiente paso que es definir muy claramente cuál es nuestro objetivo concreto en la búsqueda que se desea investigar, el siguiente paso que realizamos es Abrir el navegador e Introducir la dirección del buscador, continuando tenemos Teclear palabras claves, que nos arroja el primer resultado que es Paginas encontradas, como siguiente paso tenemos Recepción de los resultados en la forma de documentos, el cual analiza la información pertinente y seleccionara información útil que nos será de gran ayuda, el próximo paso es Sistematizar la información que nos desplego y por ultimo tenemos como resultado la información encontrada que es la necesitamos.

En cualquier proceso de búsqueda y localización de información por Internet es definir muy claramente cuál es nuestro objetivo concreto de búsqueda en cada momento determinado. Y hacerlo en función de las necesidades de información específicas en cada caso. La localización de información pertinente requerirá establecer una secuencia de distintas fases y tipos de búsqueda y el uso de herramientas diferentes de búsqueda en cada etapa de un proceso mucho más largo de localización y selección de información.

Como se ha señalado previamente la búsqueda de información en la Web ha significado el enfrentamiento a la paradoja de la saturación informativa. Son tantas las bases de datos a las que se puede acceder y es tan amplio el universo informativo, que la ventaja de contar con información variada, muchas veces ha generado en los usuarios la angustia de enfrentarse a datos inabarcables, sobre todo cuando las herramientas de búsqueda no cuentan con interfaces que permitan hacer los datos manejables. Para resolver esos problemas se ha venido desarrollando investigación que ha aportado conocimientos sobre el proceso de búsqueda de información en la Web, así como de nuevos modelos de interacción e interfaces de usuarios que hagan de los buscadores algo más útil y más fácil de usar.

Grafico 1: Proceso de Búsqueda en la Web



10.6.2. Dataset

El concepto de un data set es común a casi todas las disciplinas científicas donde los datos proporcionan la base empírica para actividades de investigación. Sin embargo, ha habido poco análisis de este concepto central. Aunque el término aparece rutinariamente en artículos, artículos e informes, así como en conversaciones informales entre científicos, no existe una definición establecida precisa (Renear, Sacchi, Wickett, & Street, 2010).

Sin embargo, el término Dataset parece no tener problemas en su uso común, lo que sugiere que existe al menos una comprensión general compartida, y de hecho nuestro examen de la literatura, que se resume a continuación, identifica claramente un conjunto de temas recurrentes. Un DataSet, al igual que una base de datos, está compuesto por un conjunto de tablas (colección de clases “DataTable”), cada una de las cuales está compuesta a su vez por un conjunto de tablas (colección de clases “DataColumn”). Dentro de un Dataset pueden establecerse relaciones entre DataTables y hasta restricciones de integridad referencial (Claves Primarias y Foráneas). Internamente, los DataSets representan toda su estructura y datos contenidos en formato XML (MSDN, 2005).

En Python, el acceso a bases de datos se encuentra definido a modo de estándar en las especificaciones de DB-API, que puedes leer en la PEP 249. Esto, significa que independientemente de la base de datos que utilicemos, los métodos y procesos de conexión,

lectura y escritura de datos, desde Python, siempre serán los mismos, más allá del conector (Bahit, 2013).

Un archivo CSV (valores separados por comas) permite que los datos sean guardados en una estructura tabular con una extensión .csv. Los archivos CSV han usados de manera extensiva en aplicaciones de comercio electrónico porque son considerados muy fáciles de procesar (Vaati, 2017).

Para las pruebas de la eficiencia del algoritmo se utilizaron unas diversas bases de datos las cuales explicaremos:

10.6.2.1. Base de Datos GBvideos

El data set que se utilizó en el proyecto de investigación tiene el nombre de GBvideos el cual fue extraída de la página web Kagle (Romero, 2019), el tamaño del data set es de 20.418 KB llegando a ser una base de datos liviana para cualquier computadora, el tipo de archivo del data set es .csv, el tipo de campos que utiliza es el alfanumérico el cual consta de leras y números en distintas celdas y el mismo se lo utilizo conjuntamente con el programa Spyder.

En Excel se presenta una tabla que nos indica que la primera fila presenta 16 campos indicando lo siguiente:

Tabla 2: Atributos de la base de Datos

Title	category_id	views

likes	dislikes	comment_count

Ayudando así a obtener un resultado aleatorio del algoritmo que se realizó. También en la tabla del Dataset sacamos una fracción de 0.001 para reducir la cantidad de Información y esto servirá para realizar una búsqueda aleatoria de videos que se encuentran en YouTube. En la tabla 2 se mencionan a los atributos que se utilizan para las relaciones de búsqueda y clasificación de lo que existen en la base de datos.

10.6.2.2. Base de datos vg1

De la misma manera el data set que se utilizó en el proyecto de investigación para realización de prueba para el algoritmo tiene como nombre de vg1 el cual fue extraída de la página web Kagle (Romero, 2019), el tamaño del data set es de 5.000 KB llegando a ser una base de datos liviana para cualquier computadora, el tipo de archivo del data set es .csv, el tipo de campos que utiliza es el alfanumérico el cual consta de leras y números en distintas celdas y el mismo se lo utilizo conjuntamente con el programa Spyder.

Como el tamaño del data set se muy corta y con una cantidad de información pequeña se optó incluir todos los datos de esta manera. Para la utilización de la mencionada base de datos se desgloso una minoría de datos, lo cual ingresamos por teclado para realizar la predicción. Cabe recalcar que se adjuntara la tabla que se utilizara para poder ingresar los parámetros correspondientes y lograr una rápida interpretación del resultado deseado.

Este conjunto de datos contiene una lista de videojuegos con ventas superiores a 100,000 copias.

Los campos incluyen

- Rango - Ranking de ventas totales
- Nombre - el nombre de los juegos
- Plataforma - Plataforma de lanzamiento de juegos (es decir, PC, PS4, etc.)
- Año - Año de lanzamiento del juego.
- Género - Género del juego
- Editor - Editor del juego
- NA_Sales - Ventas en Norteamérica (en millones)
- EU_Sales - Ventas en Europa (en millones)
- JP_Sales - Ventas en Japón (en millones)
- Other_Sales - Ventas en el resto del mundo (en millones)
- Global_Sales - Ventas mundiales totales.

Para la utilización de la Dataset adjuntaremos una fracción de instancias para realizar un ingreso por teclado hacia el algoritmo.

Tabla 3: Base de datos vg1 (Venta de Juegos)

Name	Platform	Year of Release	GenreEt	Genre	Publisher	NA Sales	EU Sales	JP Sales	Other Sales	Global Sales	Critic Score	Critic Count	User Score	User Count	Developer	Rating
Wii Sports	Wii	2006	2	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76	51	8	322	Nintendo	E
Super Mario Bros.	NES	1985	1	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24						
Mario Kart Wii	Wii	2008	6	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82	73	8.3	709	Nintendo	E
Wii Sports Resort	Wii	2009	2	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80	73	8	192	Nintendo	E
Pokemon Red/Pokemon Blue	GB	1996	4	Role-Playing	Nintendo	11.27	8.89	10.22	1	31.37						
New Super Mario Bros.	DS	2006	1	Platform	Nintendo	11.28	9.14	6.5	2.88	29.8	89	65	8.5	431	Nintendo	E
Wii Play	Wii	2006	3	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	58	41	6.6	129	Nintendo	E
New Super Mario Bros. Wii	Wii	2009	1	Platform	Nintendo	14.44	6.94	4.7	2.24	28.32	87	80	8.4	594	Nintendo	E
Mario Kart DS	DS	2005	6	Racing	Nintendo	9.71	7.47	4.13	1.9	23.21	91	64	8.6	464	Nintendo	E
Pokemon Gold/Pokemon Silver	GB	1999	4	Role-Playing	Nintendo	9	6.18	7.2	0.71	23.1						
Wii Fit	Wii	2007	2	Sports	Nintendo	8.92	8.03	3.6	2.15	22.7	80	63	7.7	146	Nintendo	E
Kinect Adventures!	X360	2010	3	Misc	Microsoft Game Studios	15	4.89	0.24	1.69	21.81	61	45	6.3	106	Good Science Studio	E
Wii Fit Plus	Wii	2009	2	Sports	Nintendo	9.01	8.49	2.53	1.77	21.79	80	33	7.4	52	Nintendo	E
Grand Theft Auto V	PS3	2013	5	Action	Take-Two Interactive	7.02	9.09	0.98	3.96	21.04	97	50	8.2	3994	Rockstar North	M
Grand Theft Auto: San Andreas	PS2	2004	5	Action	Take-Two Interactive	9.43	0.4	0.41	10.57	20.81	95	80	9	1588	Rockstar North	M
Super Mario World	SNES	1990	1	Platform	Nintendo	12.78	3.75	3.54	0.55	20.61						
Brain Age: Train Your Brain in Minutes a Day	DS	2005	3	Misc	Nintendo	4.74	9.2	4.16	2.04	20.15	77	58	7.9	50	Nintendo	E
Pokemon Diamond/Pokemon Pearl	DS	2006	4	Role-Playing	Nintendo	6.38	4.46	6.04	1.36	18.25						

Name	Platform	Year of Release	GenreEt	Genre	Publisher	NA Sales	EU Sales	JP Sales	Other Sales	Global Sales	Critic Score	Critic Count	User Score	User Count	Developer	Rating
Grand Theft Auto V	PS3	2013	5	Action	Take-Two Interactive	7.02	9.09	0.98	3.96	21.04	97	50	8.2	3994	Rockstar North	M
Grand Theft Auto: San Andreas	PS2	2004	5	Action	Take-Two Interactive	9.43	0.4	0.41	10.57	20.81	95	80	9	1588	Rockstar North	M
Grand Theft Auto V	X360	2013	5	Action	Take-Two Interactive	9.66	5.14	0.06	1.41	16.27	97	58	8.1	3711	Rockstar North	M
Grand Theft Auto: Vice City	PS2	2002	5	Action	Take-Two Interactive	8.41	5.49	0.47	1.78	16.15	95	62	8.7	730	Rockstar North	M
Grand Theft Auto III	PS2	2001	5	Action	Take-Two Interactive	6.99	4.51	0.3	1.3	13.1	97	56	8.5	664	DMA Design	M
Grand Theft Auto V	PS4	2014	5	Action	Take-Two Interactive	3.96	6.31	0.38	1.97	12.61	97	66	8.3	2899	Rockstar North	M
Mario Kart Wii	Wii	2008	6	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82	73	8.3	709	Nintendo	E
Mario Kart DS	DS	2005	6	Racing	Nintendo	9.71	7.47	4.13	1.9	23.21	91	64	8.6	464	Nintendo	E
Gran Turismo 3: A-Spec	PS2	2001	6	Racing	Sony Computer Entertainment	6.85	5.09	1.87	1.16	14.98	95	54	8.4	314	Polyphony Digital	E
Mario Kart 7	3DS	2011	6	Racing	Nintendo	5.03	4.02	2.69	0.91	12.66	85	73	8.2	632	Retro Studios, Entertainment Analysis & Development Division	E
Gran Turismo 4	PS2	2004	6	Racing	Sony Computer Entertainment	3.01	0.01	1.1	7.53	11.66	89	74	8.5	272	Polyphony Digital	E
Gran Turismo	PS	1997	6	Racing	Sony Computer Entertainment	4.02	3.87	2.54	0.52	10.95	96	16	8.7	138	Polyphony Digital	E
Gran Turismo 5	PS3	2010	6	Racing	Sony Computer Entertainment	2.96	4.82	0.81	2.11	10.7	84	82	7.5	1112	Polyphony Digital	E
Mario Kart 64	N64	1996	6	Racing	Nintendo	5.55	1.94	2.23	0.15	9.87						

Name	Platform	Year of Release	GenreEt	Genre	Publisher	NA Sales	EU Sales	JP Sales	Other Sales	Global Sales	Critic Score	Critic Count	User Score	User Count	Developer	Rating
Pokemon Red/Pokemon Blue	GB	1996	4	Role-Playing	Nintendo	11.27	8.89	10.22	1	31.37						
Pokemon Gold/Pokemon Silver	GB	1999	4	Role-Playing	Nintendo	9	6.18	7.2	0.71	23.1						
Pokemon Diamond/Pokemon Pearl	DS	2006	4	Role-Playing	Nintendo	6.38	4.46	6.04	1.36	18.25						
Pokemon Ruby/Pokemon Sapphire	GBA	2002	4	Role-Playing	Nintendo	6.06	3.9	5.38	0.5	15.85						
Pokemon Black/Pokemon White	DS	2010	4	Role-Playing	Nintendo	5.51	3.17	5.65	0.8	15.14						
PokÃ©mon Yellow: Special Pikachu Edition	GB	1998	4	Role-Playing	Nintendo	5.89	5.04	3.12	0.59	14.64						
Pokemon X/Pokemon Y	3DS	2013	4	Role-Playing	Nintendo	5.28	4.19	4.35	0.78	14.6						
Pokemon Omega Ruby/Pokemon Alpha Sapphire	3DS	2014	4	Role-Playing	Nintendo	4.35	3.49	3.1	0.74	11.68						

Fuente: Tomado por la página web Kagle 2018.

10.6.2.3. Base de datos zomato

De la misma manera el data set que se utilizó en el proyecto de investigación para realización de prueba para el algoritmo tiene como nombre de zomato el cual fue extraída de la página web Kagle (Romero, 2019), el tamaño del data set es de 547.000 KB llegando a ser una base de datos liviana para cualquier computadora, el tipo de archivo del data set es .csv, el tipo de campos que utiliza es el alfanumérico el cual consta de leras y números en distintas celdas y el mismo se lo utilizo conjuntamente con el programa Spyder.

Sobre el archivo que se utilizó, contiene una descripción la cual es la siguiente: La idea básica de analizar el conjunto de datos de Zomato es tener una idea clara de los factores que afectan la calificación agregada de cada restaurante, el establecimiento de diferentes tipos de restaurantes en diferentes lugares, siendo Bengaluru una de esas ciudades con más de 12,000 restaurantes con restaurantes que sirven platos de por todo el mundo. Cada día que se abren nuevos restaurantes, la industria aún no se ha saturado y la demanda aumenta día a día. A pesar de la creciente demanda, sin embargo, se ha vuelto difícil para los nuevos restaurantes competir con los restaurantes establecidos. La mayoría de ellos sirven la misma comida. Bengaluru es una capital de TI de la India. La mayoría de la gente aquí depende principalmente de la comida del restaurante, ya que no tienen tiempo para cocinar por sí mismos. Con una demanda tan abrumadora de restaurantes, por lo tanto, se ha vuelto importante estudiar la demografía de un lugar. Qué tipo de comida es más popular en una localidad. Hacer toda la localidad le encanta la comida vegetariana. Si es así, entonces es que la localidad está poblada por una secta particular de personas, por ej. Jain, Marwaris, Gujaratis, que son en su mayoría vegetarianos (Romero, 2019).

Para la utilización del algoritmo en esta data set se utilizó los atributos de nombre, dirección, pedidos y las visitas a ese lugar.

Tabla 4: Atributos para la utilización de la clasificación

Name	address	dish_liked	votes

10.6.2.4. Base de datos AppleStore

De la misma manera el data set que se utilizó en el proyecto de investigación para realización de prueba para el algoritmo tiene como nombre de AppleStore el cual fue extraída de la página web Kagle (Romero, 2019), el tamaño del data set en su totalidad es de 819.000 KB llegando a ser una base de datos liviana para cualquier computadora, el tipo de archivo del data set es .csv, el tipo de campos que utiliza es el alfanumérico el cual consta de leras y números en distintas celdas y el mismo se lo utilizo conjuntamente con el programa Spyder.

Sobre el archivo que se utilizó, contiene una descripción la cual es la siguiente: El paisaje móvil en constante cambio es un espacio desafiante para navegar. . El porcentaje de dispositivos móviles sobre escritorio solo está aumentando. Android tiene aproximadamente el 53,2% del mercado de teléfonos inteligentes, mientras que iOS es del 43%. Para que más personas descarguen su aplicación, debe asegurarse de que puedan encontrarla fácilmente. El análisis de aplicaciones móviles es una excelente manera de comprender la estrategia existente para impulsar el crecimiento y la retención de futuros usuarios.

Con millones de aplicaciones en la actualidad, el siguiente conjunto de datos se ha convertido en la clave para obtener las mejores aplicaciones en la tienda de aplicaciones iOS. Este conjunto de datos contiene más de 7000 detalles de aplicaciones móviles de Apple iOS (Romero, 2019).

Al poseer una gran cantidad de datos se optó de reducir a una cuarta parte de información para lograr ingresar cantidades numéricas necesarias para la predicción de resultados. Cabe recalcar que se adjuntara la tabla que se utilizara para poder ingresar los parámetros correspondientes y lograr una rápida interpretación del resultado deseado.

Tabla 5: Base de datos AppleStore (Aplicaciones Móviles)

	id	track_name	Size bytes	currency	price	Rating count tot	Rating count ver	User rating	User rating ver	ver	Cont rating	Prime genre	Sup devices num	ipadSc urls num	Lang num	Vpp lic
1	281656475	PAC-MAN Premium	100788224	USD	3.99	21292	26	4	4.5	6.3.5	4+	Games	38	5	10	1
2	281796108	Evernote - stay organized	158578688	USD	0	161065	26	4	3.5	8.2.2	4+	Productivity	37	5	23	1
3	281940292	WeatherBug - Local Weather, Radar, Maps, Alerts	100524032	USD	0	188583	2822	3.5	4.5	5.0.0	4+	Weather	37	5	3	1
4	282614216	eBay: Best App to Buy, Sell, Save! Online Shopping	128512000	USD	0	262241	649	4	4.5	5.10.0	12+	Shopping	37	5	9	1
5	282935706	Bible	92774400	USD	0	985920	5320	4.5	5	7.5.1	4+	Reference	37	5	45	1
6	283619399	Shanghai Mahjong	10485713	USD	0.99	8253	5516	4	4	1.8	4+	Games	47	5	1	1
7	283646709	PayPal - Send and request money safely	227795968	USD	0	119487	879	4	4.5	6.12.0	4+	Finance	37	0	19	1
8	284035177	Pandora - Music & Radio	130242560	USD	0	1126879	3594	4	4.5	8.4.1	12+	Music	37	4	1	1
9	284666222	PCalc - The Best Calculator	49250304	USD	9.99	1117	4	4.5	5	3.6.6	4+	Utilities	37	5	1	1
10	284736660	Ms. PAC-MAN	70023168	USD	3.99	7885	40	4	4	4.0.4	4+	Games	38	0	10	1
11	284791396	Solitaire by MobilityWare	49618944	USD	4.99	76720	4017	4.5	4.5	4.10.1	4+	Games	38	4	11	1
12	284815117	SCRABBLE Premium	227547136	USD	7.99	105776	166	3.5	2.5	5.19.0	4+	Games	37	0	6	1
13	284815942	Google – Search made just for mobile	179979264	USD	0	479440	203	3.5	4	27.0	17+	Utilities	37	4	33	1
14	284847138	Bank of America - Mobile Banking	160925696	USD	0	119773	2336	3.5	4.5	7.3.8	4+	Finance	37	0	2	1
15	284862767	FreeCell	55153664	USD	4.99	6340	668	4.5	4.5	4.0.3	4+	Games	38	5	2	1
16	284876795	TripAdvisor Hotels Flights Restaurants	207907840	USD	0	56194	87	4	3.5	21.1	4+	Travel	37	1	26	1
17	284882215	Facebook	389879808	USD	0	2974676	212	3.5	3.5	95.0	4+	Social Networking	37	1	29	1
18	284910350	Yelp - Nearby Restaurants, Shopping & Services	167407616	USD	0	223885	3726	4	4.5	11.15.0	12+	Travel	37	5	18	1
20	284993459	Shazam - Discover music, artists, videos & lyrics	147093504	USD	0	402925	136	4	4.5	11.0.3	12+	Music	37	3	16	1

	id	track_name	Size bytes	currency	price	Rating count tot	Rating count ver	User rating	User rating ver	ver	Cont rating	Prime genre	Sup devices num	ipadSc urls num	Lang num	Vpp lic
21	285005463	Crash Bandicoot Nitro Kart 3D	10735026	USD	2.99	31456	4178	4	3.5	1.0.0	4+	Games	47	0	1	1
22	285946052	iQuran	70707916	USD	1.99	2929	966	4.5	4.5	3.3	4+	Reference	43	0	2	1
23	285994151	:) Sudoku +	6169600	USD	2.99	11447	781	5	5	5.2.6	4+	Games	40	5	1	1
24	286058814	Yahoo Sports - Teams, Scores, News & Highlights	130583552	USD	0	137951	131	4	4.5	6.9	4+	Sports	37	2	6	1
25	286070473	Mileage Log | Fahrtenbuch	71203840	USD	5.99	8	0	4.5	0	9.0.5	4+	Business	37	5	3	1
27	286799607	Cleartune - Chromatic Tuner	11423008	USD	3.99	3241	297	4	4	2.1.3	4+	Music	43	2	10	1
28	286906691	Lifesum – Inspiring healthy lifestyle app	188017664	USD	0	5795	12	3.5	4	8.4.1	4+	Health & Fitness	37	5	11	1
29	286911400	Hangman.	4765696	USD	0	42316	248	3	3.5	2.0.6	9+	Games	38	0	1	1
31	288113403	iTranslate - Language Translator & Dictionary	287933440	USD	0	123215	25	3.5	5	10.5.4	4+	Productivity	37	5	23	1
32	288120394	TouchOSC	4263936	USD	4.99	782	7	4	3.5	1.9.8	4+	Music	43	1	1	1
33	288419283	RadarScope	172772352	USD	9.99	3449	23	4	4.5	3.4.1	4+	Weather	37	5	1	1
34	288429040	LinkedIn	273844224	USD	0	71856	62	3.5	4.5	9.1.32	4+	Social Networking	37	2	23	1
35	289084315	Period Tracker Deluxe	40216576	USD	1.99	13350	489	4.5	5	9.6	12+	Health & Fitness	38	0	15	1
36	289446241	Election 2016 Map	2386944	USD	0.99	137	0	3	0	5.0	4+	Entertainment	37	1	1	1
37	289523017	Blackjack by MobilityWare	105431040	USD	0	180087	1101	3.5	4.5	5.5.3	12+	Games	37	5	5	1
39	289894882	White Noise	44129280	USD	0.99	33426	299	4	5	7.2	4+	Health & Fitness	37	5	3	1
40	290638154	iHeartRadio – Free Music & Radio Stations	116443136	USD	0	293228	110	4	3	8.0.0	12+	Music	37	5	2	1
41	290807369	Line Rider iRide™	1646592	USD	1.99	21609	69	3.5	2.5	2.4	9+	Entertainment	40	0	1	1
42	290986013	Deliveries: a package tracker	30016512	USD	4.99	4684	10	4	4	8.0.3	4+	Utilities	37	4	9	1
43	291430598	Hurricane Pro	29518848	USD	2.99	2104	0	4.5	0	5.2	17+	Weather	37	0	1	1
47	292421271	Fieldrunners	66872320	USD	2.99	41633	6	4	3	1.7.177604	9+	Games	37	0	1	1

	id	track_name	Size bytes	currency	price	Rating count tot	Rating count ver	User rating	User rating ver	ver	Cont rating	Prime genre	Sup devices num	ipadSc urls num	Lang num	Vpp lic
48	292628469	Juxtaposer	45229056	USD	2.99	3610	7	4.5	5	3.8.2	4+	Photo & Video	37	5	1	1
49	292738169	Deezer - Listen to your Favorite Music & Playlists	127470592	USD	0	4677	12	3	4	6.19.0	12+	Music	37	5	21	1
50	293118835	iStellar	44241920	USD	3.99	30	0	3.5	0	2.9.0	4+	Navigation	37	0	2	0
51	293523031	Sonos Controller	107983872	USD	0	48905	2691	4.5	4.5	7.2	4+	Music	37	4	12	1
52	293573778	Avertinoo	8044544	USD	4.99	32	0	3	0	3.9.2	4+	Navigation	38	5	7	1
53	293622097	Google Earth	37214208	USD	0	446185	1359	3.5	3.5	7.1.6	4+	Travel	43	5	30	1
54	293760823	iFart - The Original Fart Sounds App	60320768	USD	1.99	21825	10	3	4	4.0.8	9+	Entertainment	37	5	1	1
55	293778748	PAC-MAN	100849664	USD	0	508808	99	3	4.5	6.3.5	4+	Games	38	5	10	1
57	294056623	FOX Sports Mobile	72748032	USD	0	57500	103	3	4	3.5.13	4+	Sports	37	0	1	0
58	294536447	First Words Animals	32164864	USD	1.99	2576	4	4	5	7.0	4+	Games	38	5	1	1
59	294631159	WeatherPro	69079040	USD	1.99	1572	34	4	4.5	4.8.2	4+	Weather	37	0	13	1
60	294934058	HotSchedules	82037760	USD	2.99	3292	2	3	5	4.56.1	4+	Business	37	2	2	1
61	295430577	Star Walk - Find Stars And Planets in Sky Above	150195200	USD	4.99	8932	55	4.5	5	7.2.3	4+	Education	37	0	12	1
62	295646461	The Weather Channel: Forecast, Radar & Alerts	199734272	USD	0	495626	5893	3.5	4.5	8.11	4+	Weather	37	0	33	1
63	295759189	Big Day - Event Countdown	13156352	USD	0.99	812	14	3	4	8.2.0	4+	Lifestyle	38	0	7	1
64	295775656	12 Steps AA Companion - Alcoholics Anonymous	30367744	USD	2.99	1583	2	3.5	4.5	2.5.9.2	12+	Lifestyle	37	5	1	1

Fuente: Tomado por la página Web Kagle 2018.

10.6.3. Programación en spyder

El algoritmo para la clasificación de aspectos de lenguaje natural basados en web semántica, se desarrolló mediante dispositivos que se utilizaran para que la aplicación se ejecute sin ningún problema se debe analizar dos escenarios, la parte física requiere de un computador compuesto por monitor, CPU, teclado, mouse y en la parte interna que trabaje con un sistema operativo Windows 7 en adelante, memoria RAM de 4GB, procesador de 32 o 64 bits, tarjeta gráfica, con un monitor CRT a una resolución de 1024 x 768 pixeles, además se necesita instalar el programa Anaconda 3 de 64 bits, con un peso de 600.967 KB con la versión de Python 2.7 conjuntamente con el programa Spyder con la versión 3.3.3 en el cual realizamos toda la codificación referente al algoritmo y en el mismo se presentó resultados de lo requerido por el usuario.

10.7. Métodos de inteligencia artificial

Clasificación

En el proyecto de investigación se utilizó dos métodos de clasificación el mismo que se encuentra dentro del campo de la minería de datos y el aprendizaje automático, en el que se logró construir un clasificador de nuestros datos y que fue capaz de tratar con diferentes tipos de datos con atributos que se utilizó y en tanto que fue apto para saber clasificar correctamente futuros ejemplos sabiendo que en la actualidad existen sistemas de clasificación muy buenos pero por lo general no están pensados en ser utilizados en entornos BigData.

El problema al que nos enfrentamos al realizar este proyecto de investigación es la clasificación supervisada. Esta clase de problemas se dividen en dos fases principales, a continuación se detallan:

1. El algoritmo de clasificación usara una serie de ejemplos llamados ejemplos de entrenamiento que ya estarán clasificados para aprender de ellos. Usando los datos de entrenamiento creara una serie de reglas o métodos de decisión para clasificar correctamente los ejemplos de entrenamiento.
2. Una vez creado el algoritmo de clasificación se pasara a clasificar ejemplos para ver qué tan bueno es el clasificador que se ha desarrollado.

10.7.1. Random Forest

La metodología seleccionada para el modelo de minería de datos es el Random Forest, como su nombre indica, son bosques aleatorios formados por un conjunto de árboles de clasificación o regresión. Esta metodología es construida mediante un algoritmo que trata de reducir la correlación entre ellos gracias a dos fuentes de aleatoriedad. Una vez construido el algoritmo, este genera una predicción promediando las predicciones individuales de cada árbol, en otras palabras relaciona con una predicción muy exacta hacia lo que se quiere buscar. Esta técnica de clasificación funciona muy bien en comparación con otras técnicas similares como Boosting o los k-NN, para cada uno de los árboles, dada la muestra inicial con N observaciones diferentes, se eligen de forma aleatoria N datos de la muestra con reemplazamiento. De esta forma cada árbol se forma con una muestra ligeramente distinta constituye la primera fuente de aleatoriedad en el algoritmo y es una metodología utilizada en varias técnicas de tratamiento de grandes datos o conocidos como BigData.

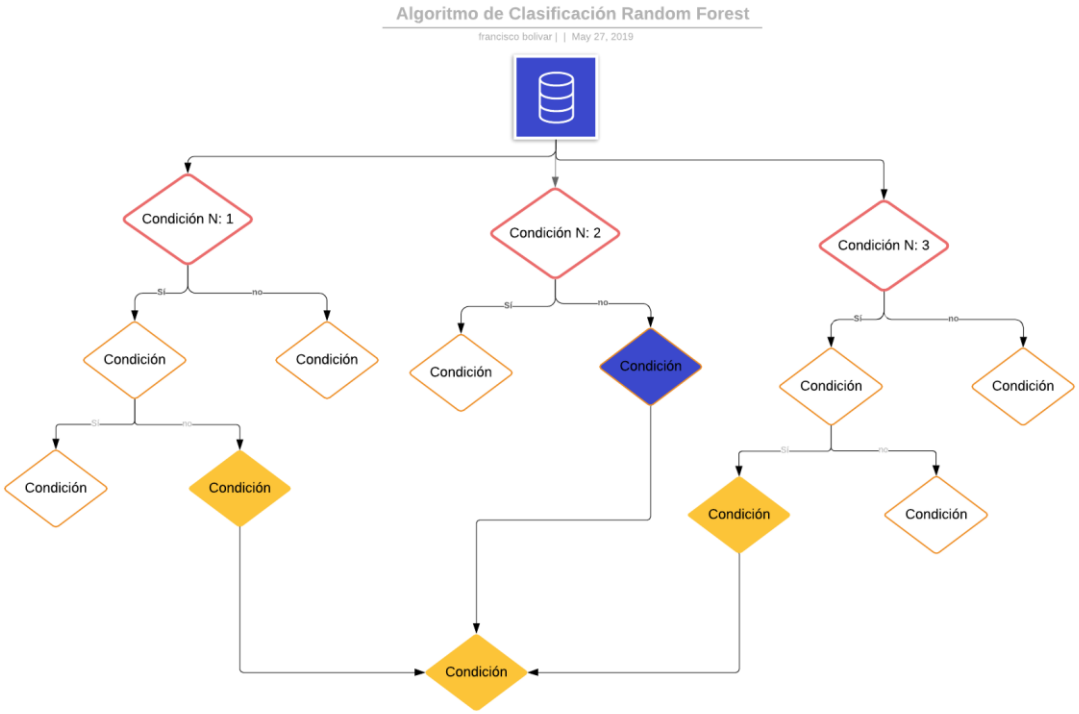
En cada nodo de cada árbol, se eligen de forma aleatoria m< P variables candidatas para la participación. El número de variable m elegido será constante durante todo el proceso de formación del árbol. Esta reducción en el número de variables candidatas constituye la segunda fuente de aleatoriedad del proceso. Teniendo todas estas variables para la aplicación de esta metodología se deja crecer cada árbol sin podar hasta la máxima extensión posible y llegar a la predicción esperada. El número de árboles también tiene efecto en la precisión de la predicción. De forma lógica, cuantos más árboles individuales diferentes se construyan con las distintas muestras de los datos iniciales mejor será el carácter de análisis de Forest y mejor serán las predicciones, puesto que se está promediando con más datos. Sin embargo, como se verá a continuación, existe un cierto valor en el cual se estabiliza el error de predicción, contribuyendo el incremento en el número de árboles de forma muy poco significativa a la reducción del error. El valor **Ntree** es el número óptimo de árboles a construir en este algoritmo, puesto que la construcción de más tiene un salto coste en tiempo que no se traduce en una mejora cuantiosa en la predicción.

Ntree: número de árboles individuales que forman el Forest.

La metodología Random Forest, los intervalos de confianza se construye con las predicciones individuales de cada uno de los árboles que forman el bosque. Estas predicciones se ordenan de forma creciente generando un intervalo y de terminando el nivel de significación deseado, se obtiene un intervalo de confianza para la predicción.

En nuestro algoritmo si se construye un Random Forest con los datos que existen, ntree = 500 una vez generadas las predicciones, estas se ordenan de menor a mayor y para un α = 10, el intervalo de confianza estará formado por las predicciones comprendidas entre 25 y la 475.

Grafico 2: Representación de la clasificación del algoritmo Random Forest



Algoritmo

El desarrollo de un algoritmo es muy estructurado, de tal manera el pseudocódigo es más parecido a un lenguaje de programación y es más fácil de traducir, por lo tanto, es más complicado de escribir. El pseudocódigo es más compacto que un diagrama de flujo. Para el desarrollo del programa va desde el planteamiento de un problema hasta obtener un programa que lo resuelve se divide en varias etapas, cada una con distintos objetivos y métodos de trabajo. Empezar a programar una vez plateado el problema, sin hacer ningún análisis ni recabar más información, suele producir un desperdicio de tiempo y recursos. Por ello el proceso de creación de este algoritmo se dividió en varios pasos:

- **Especificación:** en este paso se determinó los límites y restricciones generales del problema que tendrá que hacer, que no tendrá que hacer, bajo qué condiciones operara, aquí intervienen el docente tutor con la diversas reuniones entre ella y estudiantes.
- **Análisis:** el problema se analiza teniendo presente la especificación que deseamos solucionar. En esta fase se determinan con la mayor precisión posible las tareas

necesarias para la resolución del problema y, si fuera necesario, estas tareas se descomponen en subtareas.

- **Compilación, ejecución y verificación:** el algoritmo se compila, lo cual, las primeras veces, suele dar lugar a errores de compilación errores sintácticos en la escritura del código. Tras solucionar estos errores el programa se ejecuta y se le hace una serie de pruebas que tratan de ser representativas de todos los posibles modos de operaciones se prueba cualquier tipo de entrada posible para el programa y se verifica si la salida es la esperada. Habitualmente esto lleva a descubrir nuevos fallos, esta vez de tipo semántico el programa no hace lo que debe.

De tal manera nuestro algoritmo es una secuencia ordenada de operaciones tal que su ejecución resuelve el determinado problema. En cualquier algoritmo se puede distinguir tres partes: la entrada de datos, procesamiento y salida del resultado. El desarrollo del algoritmo se escribió sin ceñirse a las reglas de un lenguaje. Existen varias formas para describir las operaciones de las cuales tómanos en cuenta las siguientes.

Descripción textual: Consistió en describir los pasos de forma narrativa.

Lista de operaciones: es similar al texto, pero numerando los pasos, utilizando variables, etc.

Pseudocódigo: se utilizan palabras clave para identificar las estructuras del algoritmo, como alternativas, repeticiones, etc.

De las cual utilizamos pseudocódigo para la presentación del algoritmo esta representación está estructurada una parte Mediante la metodología Random Forest.

Tabla 6: Clasificación RFC

Algoritmo Clasificacion_RFC
ENTRADAS Conjunto $L = \{(x_n, y_n) \mid n = 1,2, \dots, N, x_n \varepsilon R^d, y_n \varepsilon \{1,2\}\}$
Número de árboles T
Numero de variables a seleccionar en cada nodo F
for t := **to** T **do**
$L_{bt} := MuestreoBootstrap(l)$
$c_t := Construye\ Arbol\ Aleatorio(L_{bt}, F)$
SALIDAS :
$C(x) = 18\ argmax_y \sum^T I(c_t(x) = y)$

De esta forma los dos únicos parámetros que se deben ajustar para desarrollar estos algoritmos son el número de variables a seleccionar en cada nodo, F, y el número de árboles a construir, T: Las medidas de fuerza y correlación del conjunto de clasificadores justifican este hecho, ya que si se mide la fuerza y la correlación de un conjunto de clasificadores construido con

distintos valores de F, se observa que la fuerza aumenta hasta un determinado punto a partir del cual se estabiliza, mientras que la correlación entre los clasificadores siempre aumenta al aumentar el valor de F. Por tanto, existe un cierto valor de F óptimo en el que para la fuerza máxima, la correlación de los clasificadores es mínima aunque en cualquier caso las oscilaciones en el error al utilizar distintos valores de F no son muy significativas.

El remuestreo genera una muestra aleatoria X∗ 1,. . ., X∗ n mediante el muestreo con reemplazamiento de x. Las variables aleatorias X∗ i son independientes e idénticamente distribuidas de manera uniforme en el conjunto de {x1,. . ., xn}.

Se aprecia como el espacio R^d queda partido en regiones Ri, con i = 1,...., 5, n, donde cada Ri es un rectángulo de lados paralelos a los ejes.

Argmax son los puntos, o elementos, del dominio de alguna función en la que los valores de la función se maximizan.

$\sum^T I(c_t(x) = y)$ Divide la sumatoria en sumatorias más pequeñas que se ajusten a las reglas de sumatorias.

Algoritmo Desarrollado Mediante Random Forest

Algoritmo diseñado, explicando qué hace cada sentencia, y señalando las cosas importantes.

Algoritmo 1: Algoritmo diseñado para la clasificación del Lenguaje Natural

```
#Primero hay que llamar a unas librerías que el programa necesitará
import pandas as pd
import numpy as np
#Cargamos el fichero de datos. Formato en Excel, csv.
Df=pd.read_csv('colocamos el nombre de la base de datos en formato.csv')
#Seleccionamos una muestra pequeña de la base de datos
df=Df.sample(frac=0.001)
df=df.reset_index(drop=True)
#identificara la búsqueda en títulos ya no en números
movie_indices=[i[0] for i in sim_scores]
return df['title'].iloc[movie_indices]
#Hacemos una predicción por categoría  y un tema
X=df[[Colocamos lo que vamos a Probar ]]
y=df[Tren alazar]
```

```
#Método Random Forest aplicado la predicción
rcf=RandomForestClassifier()
rcf.fit(X_train,y_train)
pred2=rcf.predict([[colocamos los campos que se va probar]])
```

10.7.2. K-nearest neighbours (K-NN)

Una forma práctica y de fácil aplicación para predecir o clasificar un nuevo dato, fundamentado en observaciones conocidas o pasadas. Esta metodología se basa, simplemente en "recordar" todos los ejemplos que se vieron en la etapa de entrenamiento. Cuando un nuevo dato se presenta al sistema de aprendizaje, este se clasifica según el comportamiento del dato más cercano (Mora-florez & Barrera-cárdenas, 2008).

Para este proyecto se utilizó el método KNN el cual nos permite clasificar un nuevo dato con facilidad en base a observaciones conocidas detallando con una predicción similar al del Random Forest ya sea el caso.

Este método consta de dos fases:

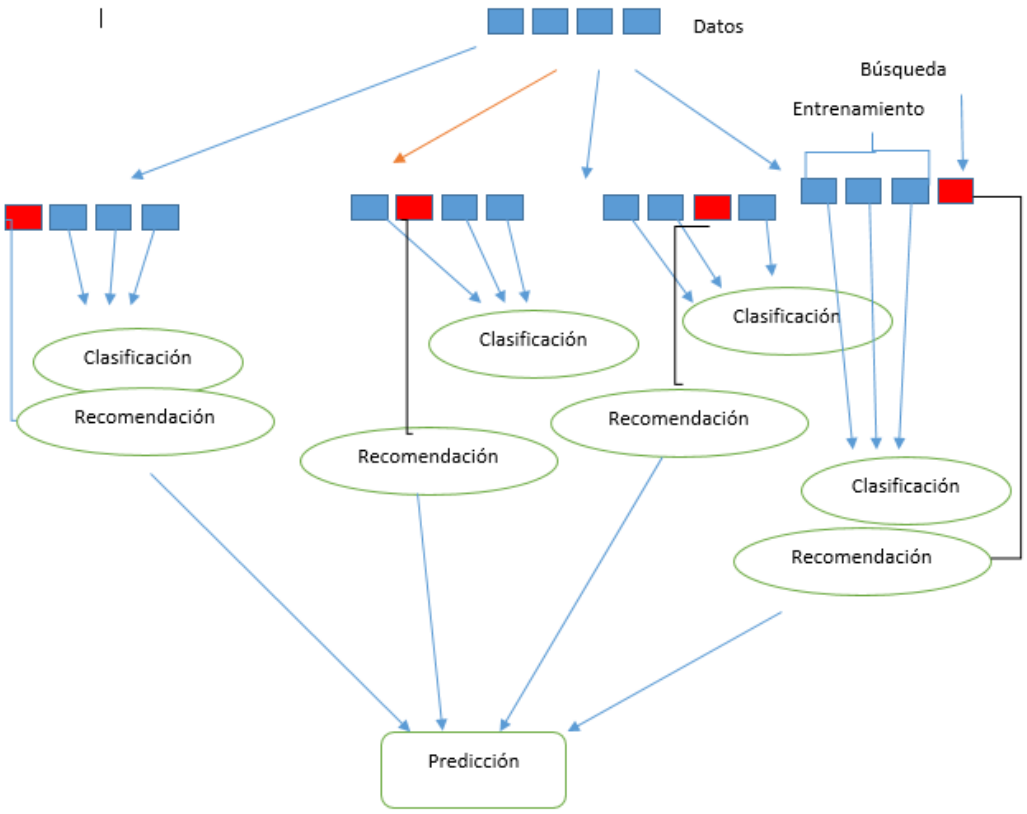
1. Fase de entrenamiento
2. Fase de clasificación

Las mismas que ayudan a mejorar el proceso de clasificación.

El método de los k-vecinos o k-NN es un método retardado y supervisado cuyo argumento principal es la distancia entre instancias. El método básicamente consiste en comparar la nueva instancia a clasificar con los datos k más próximos conocidos, y dependiendo del parecido entre los atributos el nuevo caso se ubicara en la clase que más se acerque al valor de sus propios atributos. La principal dificultad de este método consiste en determinar el valor k, ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo a la mayoría y no al parecido, si el valor es pequeño puede haber imprecisión en la clasificación a causa de los pocos datos seleccionados como instancias de comparación. Para enfrentar este problema se plantear diferentes variaciones del método: en cuanto a la forma de determinar el valor de k.

La experimentación se realizó de la siguiente manera utilizamos los datos extraídos de YouTube la cual explicamos en Dataset el origen de la información, para los cuales se les aplico una serie de pruebas ingresadas por teclado como son k, de tal manera llegar a una predicción y comparación con la metodología Random Forest.

Grafico 3: Esquema K-NN validación, con k=4 y un solo clasificador básico



Descripción del grafico N: 3

En el grafico 3 se representa el esquema K-NN validación con cuatro vecinos, el cual se realizó con un solo clasificador básico y el que posee un conjunto de entrenamiento, el mismo que posee una base de datos, para iniciar con el primer vecino el cual consta de 4 elementos en los cuales el primero es la recomendación que más incide y los 3 restantes son los elementos de clasificación, en el segundo vecino tenemos que el segundo elemento es la recomendación y el primero, tercero y cuarto son los elementos de clasificación, en el tercer vecino tenemos lo siguiente en el tercer elemento tenemos la recomendación y en el primero, segundo y cuarto la clasificación, y por ultimo tenemos el cuarto vecino que viene a ser el vecino de entrenamiento en el cual tenemos que en el cuarto elemento encontramos a la búsqueda en este caso y en el primer, segundo y tercer elemento a la clasificación, una vez encontrado los elementos recomendados y así mismo los elementos recomendados, a continuación seleccionamos los elementos recomendados y si tenemos que en 2 o 3 elementos nos coinciden los resultados por ende el mismo será nuestra respuesta en este caso será la predicción.

Algoritmo knn

El proceso de aprendizaje de este clasificador consiste en almacenar en un vector el conjunto de entrenamiento, junto a la clase asociada a cada muestra de este conjunto (datos ya clasificados). Luego se debe calcular la distancia entre cada muestra de entrenamiento y el vector a clasificar, seleccionando las K muestras más cercanas. Luego se debe hacer el mismo proceso pero con los datos de validación, para así diseñar el clasificador. Luego se calcula el porcentaje de clasificación para poder conocer el poder de generalización.

Grafico 4: Notación para el paradigma K-NN

		X_1	...	X_j	...	X_n	C
$(\mathbf{x}_1, c_1)$	1	x_{11}	...	x_{1j}	...	x_{1n}	c_1
	⋮	⋮		⋮		⋮	⋮
$(\mathbf{x}_i, c_i)$	i	x_{i1}	...	x_{ij}	...	x_{in}	c_i
	⋮	⋮		⋮		⋮	⋮
$(\mathbf{x}_N, c_N)$	N	x_{N1}	...	x_{Nj}	...	x_{Nn}	c_N
$\mathbf{x}$	$N+1$	$x_{N+1,1}$	...	$x_{N+1,j}$	...	$X_{N+1,n}$	?

Fuente: Implementación del algoritmo de los k vecinos más cercanos (k-NN) y estimación del mejor valor local de k para su cálculo por Gonzalo Berástegui Arbeloa 2018

En la representación de la tabla 7 se observa la Clasificación K-NN donde D indica un fichero de n casos, cada uno de los cuales está caracterizado por n variables predictores, x1,. . ., xn y una variable a predecir, la clase c.

Tabla 7: Clasificación k-NN

Algoritmo Clasificacion_kNN
ENTRADAS: D = $\{(x_1, c_1) \ldots (x_n, c_n)\}$
X = $(x_1, \ldots \ldots \ldots, x_n)$ Nuevo caso a clasificar

PARA todo objeto ya clasificado (x_i, c_i)
Calcular $d_i = d(x_i, x)$
Ordenar d_i $(i = 1, \ldots., N)$ en orden ascendente
Quedarnos con los K casos $D\frac{K}{x}$ ya clasificados más cercanos a x
Asignar a x la clase más frecuente en $D\frac{K}{x}$
FIN

Los N casos se denotan por

(x1, c1), . . ., (xN , cN) donde

xi = (xi, 1 . . . xi,n) para todo i = 1, . . . , N

ci ∈ {c 1, . . . , cm} para todo i = 1, . . . , N

c 1 , . . . , cm denotan los m posibles valores de la variable clase C

El nuevo caso que se pretende clasificar se denota por x = (x1,. . ., xn).

Tal y como puede observarse en el mismo, se calculan las distancias de todos los casos ya clasificados al nuevo caso, x, que se pretende clasificar. Una vez seleccionados los K casos ya clasificados, $D\frac{K}{x}$ más cercanos al nuevo caso, x, a este se le asignara la clase más frecuente de entre los K objetos, $D\frac{K}{x}$

11. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Para poder realizar un correcto análisis de los resultados se deben establecer medidas con las cuales se podrán realizar comparativas entre los distintos algoritmos y sus respectivas iteraciones.

Medidas que se lleva a cabo para un correcto análisis.

1. Grado de exactitud: Es el grado de concordancia entre las clases asignadas por el clasificador y sus ubicaciones correctas según los datos recolectados por el usuario y considerados como datos de referencia a tomar.
2. Matriz de Confusión: Es la herramienta más utilizada para la estimación de exactitud de un clasificador, también llamada matriz de error o de contingencia. Esta es una matriz cuadrada de n x n, donde n es el número de clases. Dicha matriz muestra la relación entre las series de medidas correspondientes al área en estudio. En una matriz de confusión las columnas corresponden a los datos de referencia, mientras que las filas corresponden a las asignaciones del clasificador.

Para la realización de los análisis se puede considerar una división de estas según aplicaciones realizadas a los algoritmos planteados con anterioridad y las contextualización de los datos a utilizar por estos:

El desarrollo del algoritmo se formó mediante la unión de dos metodologías Random Forest y k-NN, estos algoritmos permiten clasificar de forma más rápida las recomendaciones de acuerdo a la búsqueda, se desarrolló mediante Python Development Environment | Spyder-

IDE.org, software libre y fácil de instalar, la base de datos se extrajo de un repositorio que especificamos en Dataset, el Objetivo de desarrollar un algoritmo basado en inteligencia artificial para la clasificación mediante la aplicación de herramientas orientada a la Web Semántica, se completó correctamente con presencia de resultados de búsqueda.

K-NN: Para este algoritmo se plantearon dos grandes grupos de análisis:

1. Según K, es decir, según la cantidad de vecinos cercanos, se sugirió utilizar 4 diferentes: 1, 2, 3, 4 vecindades. El principal objetivo es identificar como influye la vecindad en los respectivos análisis deseados.
2. Según porcentaje de datos de entrenamiento: Al ser un algoritmo que necesita de un grupo previo de entrenamiento, se plantea la experimentación según la cantidad de datos utilizados para entrenar al algoritmo versus los que debe clasificar. A través de este experimento se puede analizar como la cantidad de datos de entrenamiento afecta al resultado, considerar que mientras más alta sea el porcentaje de entrenamiento, más pequeño es el conjunto de datos a clasificar.

Random Forest: Dado que el número de casos en el conjunto de entrenamiento es N. Una muestra de esos N casos se toma aleatoriamente pero CON REEMPLAZO. Esta muestra será el conjunto de entrenamiento para construir el árbol i. Si existen M variables de entrada, un número m<M se especifica tal que para cada nodo, m variables se seleccionan aleatoriamente de M. La mejor división de estos m atributos es usado para ramificar el árbol. El valor m se mantiene constante durante la generación de todo el bosque. Cada árbol crece hasta su máxima extensión posible y NO hay proceso de poda.

Contextualización de datos: Se plantearon 4 factores de contextos distintos con el fin de comprender como afecta los datos de entrada al algoritmo en los resultados finales de estos, permitiendo comparar cual metodología podría predominar sobre esta clasificación.

Con el fin de mejorar el rendimiento del algoritmo desarrollado, se ha incorporado un método, la filtración de datos. Este método consiste en la eliminación de un sector de la gama de datos, con el fin de reducir atributos que lleven a redundancia a la hora de aplicar el algoritmo de clasificación. Esta actividad se realiza después de la ponderación, cuando los atributos ya poseen sus respectivos pesos según la categoría a la que pertenecen.

El filtrado es aplicado según alguna razón escogida y que comprenden dentro de esa razón, es decir, se posee una gama de 38.916 datos y se escoge una razón de filtrado de 0.001, entonces

la eliminación comprenderá de 38.877 datos, por lo que el algoritmo utilizaría el 10% de los datos sobrantes.

Cabe señalar que si se trabaja con más datos el algoritmo tendría una tardanza de 3 a 5 minutos por lo que se debe considerar que el 10% comprende una gran cantidad de datos.

Resultados Obtenidos

1. **Resultados con la primera base de datos.**

El presente proyecto de investigación los resultados fueron extraídos mediante el desarrollo del algoritmo de clasificación, para el conjunto de entrenamiento se tomó en cuenta un rango de 1 a 9 para la metodología k-NN y para Random Forest una variable m se seleccionan aleatoriamente.

Tabla 8: Datos de Predicción K-NN y RF de la Dataset GBvideos

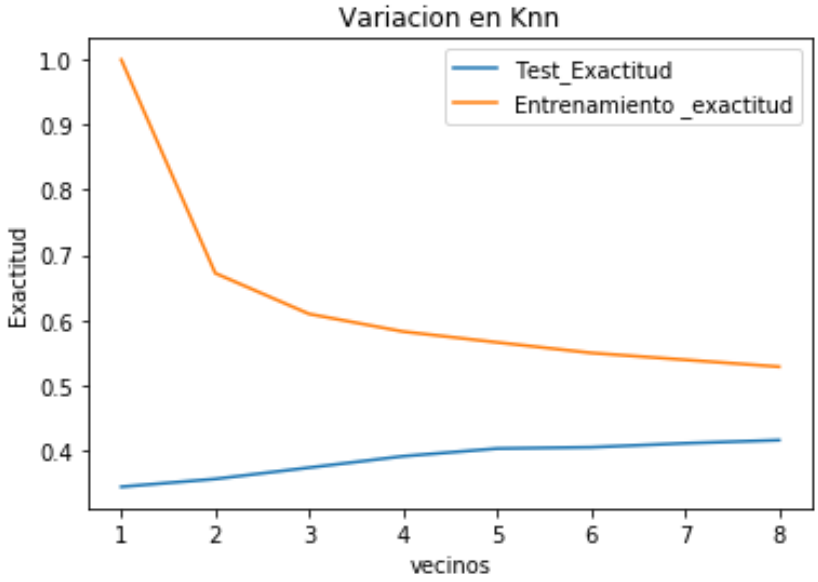
Dataset	**K-NN**		**Random Forest**
38916(Fracción del 0.001 = 39)	1 Vecino	0.3458	0.7107
38916(Fracción del 0.001 = 39)	2 Vecinos	0.3530	0.7000
38916(Fracción del 0.001 = 39)	3 Vecinos	0.3777	0.7070
38916(Fracción del 0.001 = 39)	4 Vecinos	0.3956	0.7040

De un total de 38916 datos que está compuesto el Dataset la predicción que se utiliza en los dos algoritmos es de distintos valores lo cual K-NN es de distintos valores de 1 a 4 vecinos se mantiene un rango de 0.34 a 0.39, en los cuales la precisión menor es de 0.34 con 1 vecino, el siguiente valor de precisión es de 0.35 con 2 vecinos, la precisión 0.37 con 3 vecinos y por ultimo tenemos la predicción con valor de 0.39 tomando en cuenta que es para un pronóstico preciso, Random Forest obtuvimos los siguientes resultados tomando en cuenta así como se realizó en la metodología K-NN con un total de 39 datos y de igual manera con predicciones de 1 a 4 vecinos que sus valores varían entre sí, tenemos que con 1 vecino posee 0.71 la cual es la precisión mayor de todas, con 2 vecinos la precisión tiene el valor de 0.70 a continuación con 3 vecinos la predicción tiene el valor de 0.707 y por ultimo con 4 vecinos el Random Forest tiene el valor de precisión de 0.704.

Una vez evaluada la capacidad predictiva del algoritmo *K-NN*, y los árboles de decisión simples obtenidos mediante el paquete *KNeighborsClassifier*, estimamos el modelo que obtendríamos si ejecutásemos *n* árboles de decisión simultáneamente para *n*= **38916** en nuestro caso mediante el algoritmo *randomForest*.

El algoritmo *randomForest* es un método de estimación combinado, donde el resultado de la estimación se construye a partir de los resultados obtenidos mediante el cálculo de *n* árboles donde los predictores son incluidos al azar.

Grafico 5: Exactitud y vecinos referentes a todos los datos de la Dataset GBvideos



En esta representación creamos un modelo con Python para procesar y clasificar puntos de un conjunto de entrada con el algoritmo k-Nearest Neighbor. Cómo su nombre en inglés lo dice, se evalúan los k vecinos más cercanos para poder clasificar nuevos puntos. Al ser un algoritmo supervisado debemos contar con suficientes muestras etiquetadas para poder entrenar el modelo con buenos resultados. Este algoritmo es bastante simple y como vimos antes necesitamos muchos recursos de memoria y CPU para mantener el Dataset vivo y evaluar nuevos puntos. Esto no lo hace recomendable para conjuntos de datos muy grandes. En la extracción de esta gráfica, sólo utilizamos todos los datos existentes la cual este grafica es la general de toda la Dataset. Finalmente pudimos hacer nuevas predicciones y a raíz de los resultados, comprender mejor la problemática planteada. En la presente grafica nos muestra la exactitud y el entrenamiento que tiene para el desarrollo del proyecto, explicando lo siguiente el test Exactitud nos indica hasta donde se utilizara la predicción de 0.1 hasta 0.5, pero en la de entrenamiento nos da un valor de predicción de un 0.7 creíble que se utilizara en toda la ejecución del algoritmo desarrollado.

En las siguientes graficas se depura el algoritmo desarrollado para la clasificación, de la cual se imprime 4 consultas por cada K vecinos ingresados.

Tabla 9: Ejecución del algoritmo con 1 vecino k-NN y Random Forest n=? Variable de la Dataset GBvideos.

```
--------------------------------------------------
KNN
0.2
[10]
Maroon 5 - Wait
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.4
[1]
CRISTIANO RONALDO E FRED, O GRANDE ENCONTRO
--------------------------------------------------
--------------------------------------------------
La selección es ---- Maroon 5 - Wait ------
Las recomendaciones para esta selección son:
1. Maroon 5 - Wait
2. CRISTIANO RONALDO E FRED, O GRANDE ENCONTRO
--------------------------------------------------
--------------------------------------------------
```

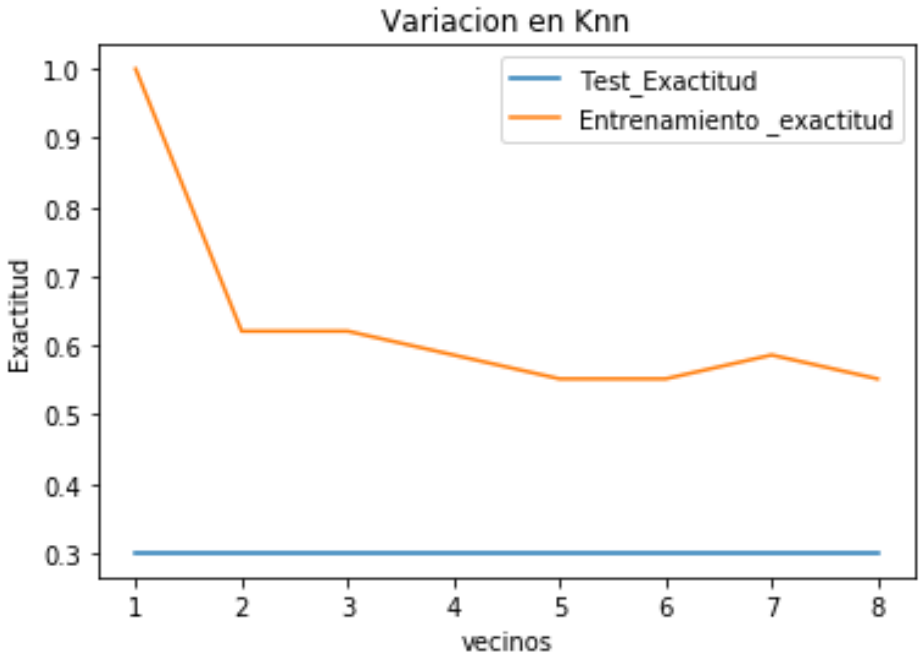
Se obtuvo resultados de k-NN y Random Forest, donde se predice en el primer algoritmo un valor de 0.2 con un puesto de categoría 10 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera el segundo algoritmo predice un valor de 0.4 con un puesto de categoría 1 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que el algoritmo Random Forest supera a k-NN por 0.2 de exactitud.

Tabla 10: Ejecución del algoritmo con 2 vecino k-NN y Random Forest n=? Variable de la Dataset GBvideos.

```
--------------------------------------------------
KNN
0.3
[10]
Maroon 5 - Wait
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.3
[1]
Is Forces of Destiny good?
--------------------------------------------------
--------------------------------------------------
La seleccion es ---- Is Forces of Destiny good? ------
Las recomendaciones para esta selección son:
1. Maroon 5 - Wait
2. Is Forces of Destiny good?
--------------------------------------------------
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en el primer algoritmo un valor de 0.3 con un puesto de categoría 10 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera el segundo algoritmo predice un valor de 0.3 con un puesto de categoría 1 y el Tema de la recomendación que más se asemeja al de la búsqueda.

Grafico 6: Variación en k-NN con 2 Vecinos de la Dataset GBvideos.



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.3, pero en la de entrenamiento nos da un valor de predicción de 0.6 que se va incrementando como indica en el Grafico 6 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 11: Ejecución del algoritmo con 3 vecino k-NN y Random Forest n=? Variable de la Dataset GBvideos.

```
--------------------------------------------------
KNN
0.4
[10]
Judas Priest - Spectre (Official Video)
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.3
[10]
Judas Priest - Spectre (Official Video)
```

```
--------------------------------------------------
--------------------------------------------------
La seleccion es ---- Lonzo Ball Performs Bad and Boujee by Migos | Lip Sync Battle Preview ------
Las recomendaciones para esta selección son:
1. Judas Priest - Spectre (Official Video)
2. Judas Priest - Spectre (Official Video)
--------------------------------------------------
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice el primer algoritmo un valor de 0.4 con un puesto de categoría 10 y el Tema de la recomendación de la que más se asemeja al de la selección, de la misma manera el segundo algoritmo predice un valor de 0.3 con un puesto de categoría 10 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que al tener k-NN 3 vecinos tiene un incremento al Random Forest de 0.1 de fracción, de tal manera al ejecutarse la misma recomendación en ambas metodologías se observa que Random Forest se asemeja al mismo resultado aplicando su método de búsqueda por árboles.

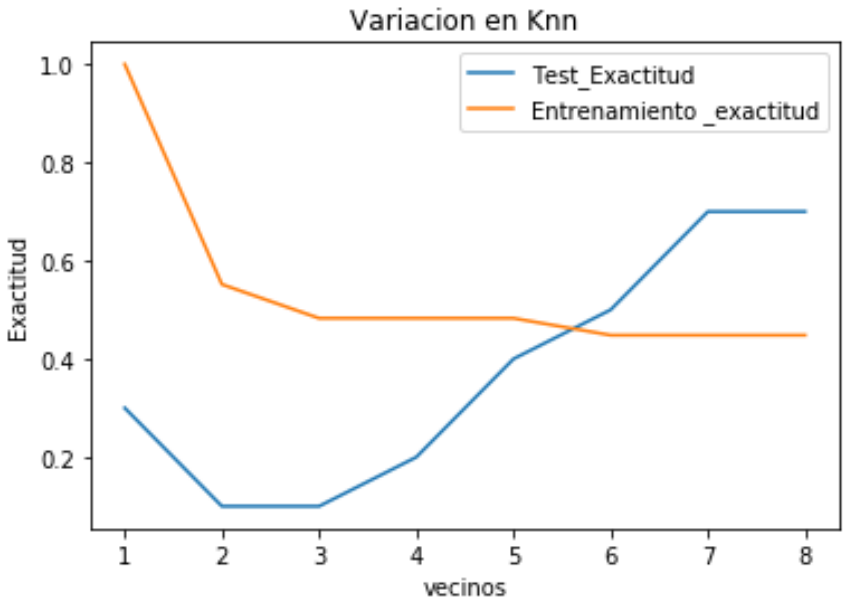
Tabla 12: Ejecución del algoritmo con 4 vecino k-NN y Random Forest n=? Variable de la Dataset GBvideos.

```
--------------------------------------------------
KNN
0.2
[24]
YouTube Red's Cobra Kai Keeps the Karate Kid Legacy Alive
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.5
[24]
YouTube Red's Cobra Kai Keeps the Karate Kid Legacy Alive
--------------------------------------------------
--------------------------------------------------
La selección es ---- Every Kevin Spacey Joke from Difficult People ------
Las recomendaciones para esta selección son:
1. YouTube Red's Cobra Kai Keeps the Karate Kid Legacy Alive
2. YouTube Red's Cobra Kai Keeps the Karate Kid Legacy Alive
--------------------------------------------------
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice el primer algoritmo un valor de 0.2 con un puesto de categoría 24 y el Tema de la recomendación de la que más se asemeja

al de la búsqueda, de la misma manera el segundo algoritmo predice un valor de 0.5 con un puesto de categoría 24 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que Random Forest supera a k-NN por 0.3 de predicción, de tal manera ambas metodologías imprimen el mismo título, cabe recalcar que al tener k-NN 4 vecinos su exactitud en la Grafica es muy corta llego a predecir el mismo tema que Random Forest, con estas pruebas se añade que ambos métodos son compatibles y precisos al momento de una clasificación de n datos.

Grafico 7: Variación en k-NN con 4 Vecinos de la Dataset GBvideos



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.2, pero en la de entrenamiento nos da un valor de predicción de 0.5 que se va incrementando como indica en el Grafico 7 por ende Random Forest recurre a una búsqueda muy similar que se busca.

2. Resultados con la segunda base de datos

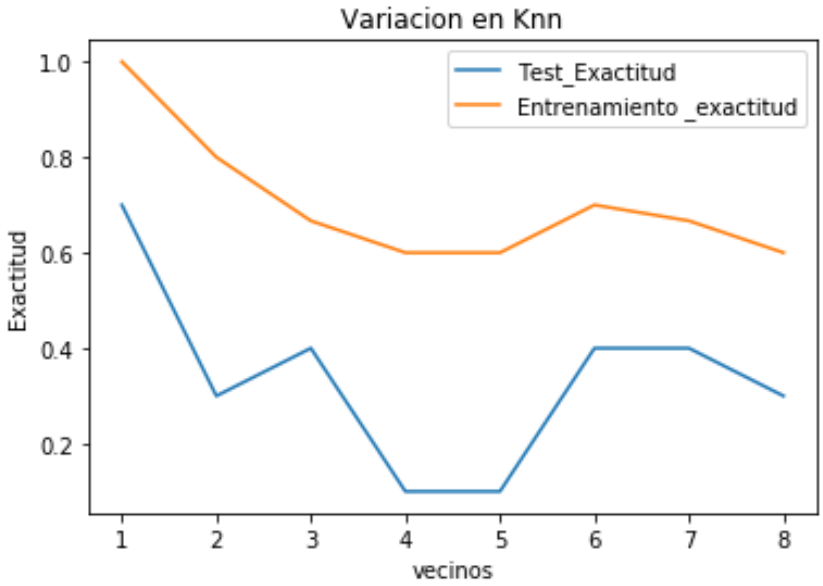
El presente proyecto de investigación los resultados fueron extraídos mediante el desarrollo del algoritmo de clasificación, para el conjunto de entrenamiento se tomó en cuenta un rango de 1 a 9 para la metodología k-NN y para Random Forest una variable m se seleccionan aleatoriamente.

Tabla 13: Datos de Predicción K-NN y RF de la Dataset vg1

Dataset	Predicción K-NN		Random Forest
16600(Fracción del 0.001 = 39)	1 Vecino	0.7	0.6
16600(Fracción del 0.001 = 39)	2 Vecinos	0.5	0.9
16600(Fracción del 0.001 = 39)	3 Vecinos	0.5	0.7
16600(Fracción del 0.001 = 39)	4 Vecinos	0.7	0.7

De un total de 16600 datos que está compuesto el Dataset se tomó una fracción del 0.001 que nos da 39 datos para la prueba y en la que la predicción que se utiliza K-NN es de distintos vecinos de 1 a 4 por lo que varían los mismos, en los cuales la predicción menor es de 0.5 con 1 vecino y 4 vecinos, la siguiente predicción tiene el valor de 0.5 que tiene 3 vecinos y por último la predicción de valor 0.5 con 2 vecinos que es de un pronóstico preciso, Random Forest obtuvimos los siguientes resultados tomando en cuenta así como se realizó en la metodología K-NN con un total de 41 datos y de igual manera con predicciones de 1 a 4 vecinos que sus valores varían entre sí, tenemos que 1 vecino posee un valor de 0.6 la cual es la menor de todas , con 2 vecinos la predicción tiene el valor de 0.9 la cual es el pronóstico preciso a continuación con 3 vecinos tenemos que el Random Forest tiene el valor de 0.7 al igual que con 4 vecinos. Una vez evaluada la capacidad predictiva del algoritmo *K-NN*, y los árboles de decisión simples obtenidos mediante el paquete *KNeighborsClassifier*, estimamos el modelo que obtendríamos si ejecutásemos *n* árboles de decisión simultáneamente para *n*= **41** en nuestro caso mediante el algoritmo *randomForest*. El algoritmo *randomForest* es un método de estimación combinado, donde el resultado de la estimación se construye a partir de los resultados obtenidos mediante el cálculo de *n* árboles donde los predictores son incluidos al azar.

Grafico 8: Variación en k-NN con 1 Vecino de la Dataset vg1



En esta representación creamos un modelo con Python para procesar y clasificar puntos de un conjunto de entrada con el algoritmo k-Nearest Neighbor. Cómo su nombre en inglés lo dice, se evalúan los k vecinos más cercanos para poder clasificar nuevos puntos. Al ser un algoritmo supervisado debemos contar con suficientes muestras etiquetadas para poder entrenar el modelo con buenos resultados. Este algoritmo es bastante simple y como vimos anteriormente necesitamos muchos recursos de memoria y CPU para mantener el Dataset activo y evaluar nuevos puntos. Esto no lo hace recomendable para conjuntos de datos muy grandes. En la extracción de esta gráfica, sólo utilizamos todos los datos existentes la cual esta grafica cambiara por las variables de entrenamiento que se utilicen. Finalmente pudimos hacer nuevas predicciones y a raíz de los resultados, comprender mejor la problemática planteada. En la presente grafica nos muestra la exactitud y el entrenamiento que tiene para el desarrollo del proyecto, explicando lo siguiente el test Exactitud nos indica hasta donde se utilizara la predicción de 0.1 hasta 0.5, pero en la de entrenamiento nos da un valor de predicción de un 0.9 creíble que se utilizara en toda la ejecución del algoritmo desarrollado.

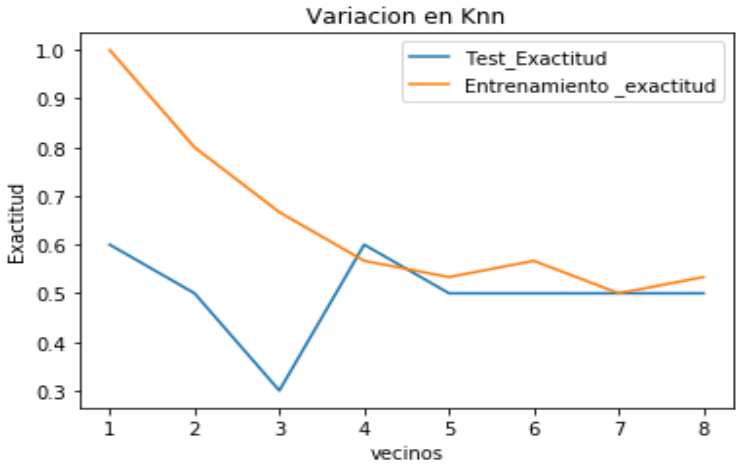
En las siguientes graficas se depura el algoritmo desarrollado para la clasificación, de la cual se imprime 4 consultas por cada K vecinos ingresados.

Tabla 14: Ejecución del algoritmo con 1 vecino k-NN y Random Forest n=? Variable de la Dataset vg1

KNN 0.7 [2] Sports Wii Sports ---------- ---------- RANDOM FOREST 0.6 [2] Sports Wii Sports ---------- ---------- Las recomendaciones para esta selección son: 1. Wii Sports 2. Wii Sports Accuracy: 0.50 (+/- 0.08) [KNN] Accuracy: 0.45 (+/- 0.07) [Random Forest] ----------

Se obtuvo los siguientes resultados de K-NN y Random Forest, donde se predice en la primera metodología un valor de 0.7 con un puesto de categoría de 2 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.6 con un puesto de categoría 2 y el Tema de K-NN es la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología k-NN supera a Random Forest por 0.1 de exactitud.

Grafico 9: Variación en k-NN con 2 Vecinos de la Dataset vg1



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.6, pero en la de entrenamiento nos da un valor de predicción de 0.7 que se va incrementando como indica en el Grafico 9 por ende Random Forest recurre a una búsqueda muy similar que se busca.

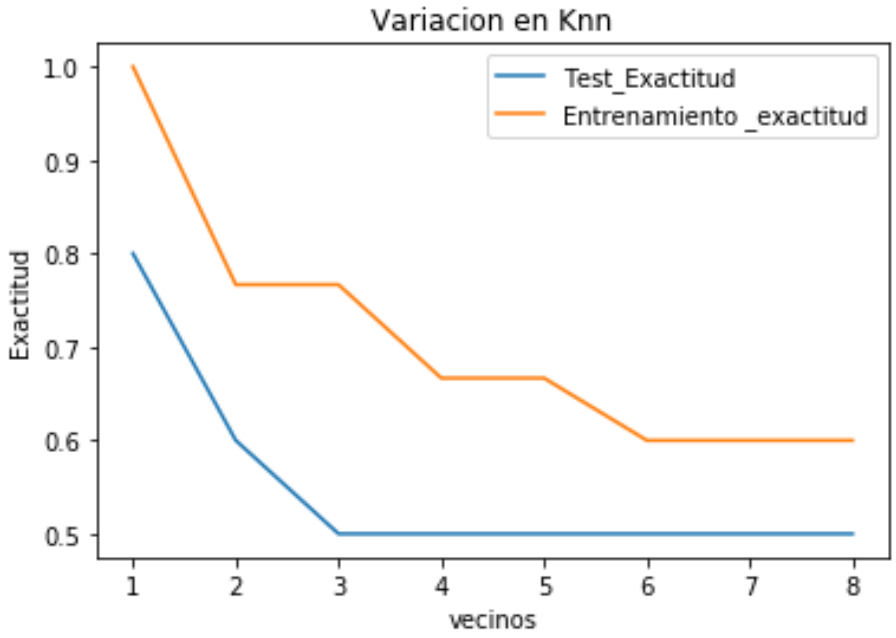
Tabla 15: Ejecución del algoritmo con 2 vecinos k-NN y Random Forest n=? Variable de la Dataset vg1.

```
--------------------------------------------------
KNN
0.5
[2]
Sports
Wii Sports
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.9
[2]
Sports
Wii Sports
--------------------------------------------------
--------------------------------------------------
Las recomendaciones para esta selección son:
1. Wii Sports
2. Wii Sports
Accuracy: 0.34 (+/- 0.11) [KNN]
Accuracy: 0.53 (+/- 0.07) [Random Forest]
--------------------------------------------------
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.5 con un puesto de categoría 2 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera en la segunda metodología predice un valor de 0.9 con un puesto de categoría 2 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología Random Forest supera a k-NN por 0.4 de exactitud.

En resultado en la tabla 15 nos muestra que al poseer una gran predicción se logra un resultado más preciso y confiable para lograr una evaluación más precisa cobre el algoritmo desarrollado.

Grafico 10: Variación en k-NN con 3 Vecinos de la Dataset vg1



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que al tener 1 a 2 vecinos tiene una predicción de 0.5 desde ahí se mantiene el rango con 3 vecinos y los que correspondan y se va incrementando como indica en el Grafico 10 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 16: Ejecución del algoritmo con 3 vecinos k-NN y Random Forest n=? Variable de la Dataset *vg1*.

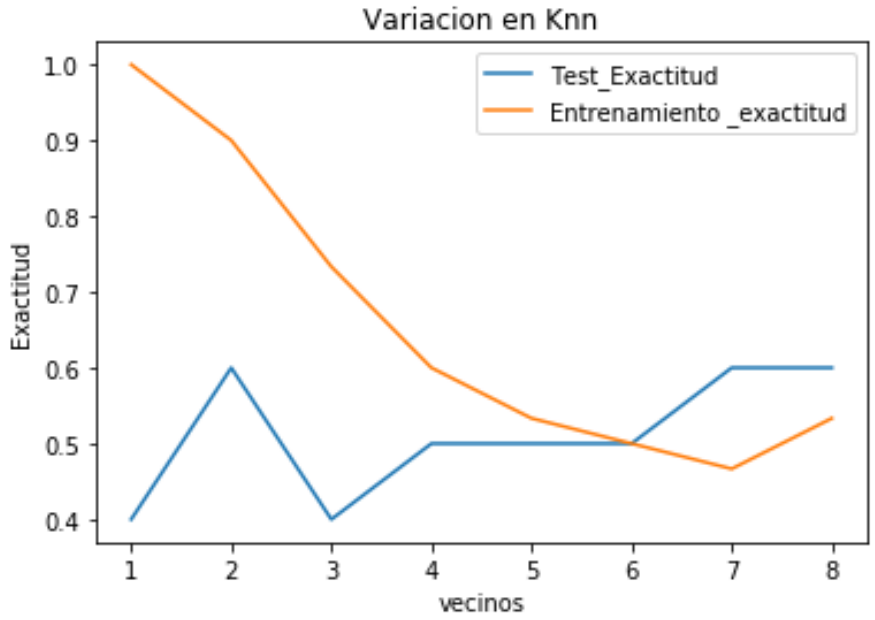
KNN
0.5
[2]
Sports
Wii Sports

RANDOM FOREST
0.7
[2]
Sports
Wii Sports

Las recomendaciones para esta selección son:
1. Wii Sports
2. Wii Sports
Accuracy: 0.39 (+/- 0.18) [KNN]
Accuracy: 0.56 (+/- 0.10) [Random Forest]

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.5 con un puesto de categoría 2 y el Tema de la recomendación de la que más se asemeja al de la selección, de la misma manera la segunda metodología predice un valor de 0.7 con un puesto de categoría 2 y el Tema de la recomendación que más se asemeja al de la búsqueda.

Grafico 11: Variación en k-NN con 4 Vecinos de la Dataset vg1



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.5, pero en la de entrenamiento nos da un valor de predicción de 0.6 que se va incrementando como indica en el Grafico 11 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 17: Ejecución del algoritmo con 4 vecinos k-NN y Random Forest n=? Variable de la Dataset vg1.

KNN
0.5
[2]
Sports
Wii Sports

RANDOM FOREST
0.7
[2]
Sports
Wii Sports

```
--------------------------------------------------
Las recomendaciones para esta selección son:
1. Wii Sports
2. Wii Sports
Accuracy: 0.43 (+/- 0.22) [KNN]
Accuracy: 0.52 (+/- 0.07) [Random Forest]
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.5 con un puesto de categoría 2 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.7 con un puesto de categoría 2 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología Random Forest supera a k-NN por 0.2 de predicción, cabe recalcar que al tener k-NN 4 vecinos su exactitud en la Grafica es muy corta, con estas pruebas se añade que ambos métodos son compatibles y precisos al momento de una clasificación de n datos.

3. Resultados de la tercera base de datos

El presente proyecto de investigación los resultados fueron extraídos mediante el desarrollo del algoritmo de clasificación, para el conjunto de entrenamiento se tomó en cuenta un rango de 1 a 9 para la metodología k-NN y para Random Forest una variable m se seleccionan aleatoriamente.

Tabla 18: Datos de Predicción K-NN y RF de la Dataset zomato

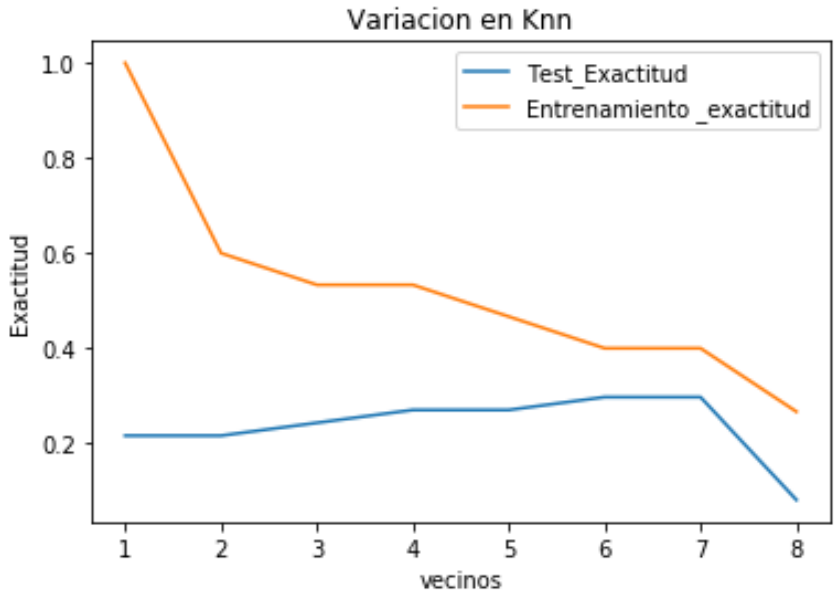
Dataset	Predicción K-NN		Random Forest
12000(Fracción del 0.001 = 39)	1 Vecino	0.21	0.21
12000(Fracción del 0.001 = 39)	2 Vecinos	0.4	0.24
12000(Fracción del 0.001 = 39)	3 Vecinos	0.2	0.5
12000(Fracción del 0.001 = 39)	4 Vecinos	0.18	0.29

De un total de 82.716 datos que está compuesto el Dataset se tomó una fracción del 0.001 de los cuales se obtuvieron un total de 39 datos y en la que la predicción que se utiliza en K-NN es de distintos vecinos de 1 a 4 por lo que varían los mismos, en los cuales la predicción menor es de 0.18 con 3 vecino, la siguiente predicción tiene el valor de 0.21 que tiene 1 vecinos y por último la predicción de valor 0.24 con 2 vecinos que es de un pronóstico pequeño pero a la vez preciso, por motivo que solo se trabaja con un dato de entrenamiento en comparación de los dos Dataset que se utiliza 3 datos para el test.

En Random Forest obtuvimos los siguientes resultados tomando en cuenta así como se realizó en K-NN con un total de 39 datos y de igual manera con predicciones de 1 a 4 vecinos que sus valores varían entre sí, tenemos que 1 vecino posee un valor de 0.21 la cual es la menor de todas, con 2 vecinos la predicción tiene el valor de 0.24 la cual es el pronóstico preciso a continuación con 3 vecinos tenemos que el Random Forest tiene el valor de 0.5.

Una vez evaluada la capacidad predictiva del algoritmo *K-NN*, y los árboles de decisión simples obtenidos mediante el paquete *KNeighborsClassifier*, estimamos el modelo que obtendríamos si ejecutásemos *n* árboles de decisión simultáneamente para *n*= **82.716** en nuestro caso mediante el algoritmo *randomForest*. El algoritmo *randomForest* es un método de estimación combinado, donde el resultado de la estimación se construye a partir de los resultados obtenidos mediante el cálculo de *n* árboles donde los predictores son incluidos al azar.

Grafico 12: Variación en k-NN con 1 Vecinos de la Dataset zomato



En esta representación creamos un modelo con Python para procesar y clasificar puntos de un conjunto de entrada con el algoritmo k-Nearest Neighbor. Cómo su nombre en inglés lo dice, se evalúan los k vecinos más cercanos para poder clasificar nuevos puntos. Al ser un algoritmo supervisado debemos contar con suficientes muestras etiquetadas para poder entrenar el modelo con buenos resultados. Este algoritmo es bastante simple y como vimos anteriormente necesitamos muchos recursos de memoria y cpu para mantener el Dataset activo y evaluar nuevos puntos. Esto no lo hace recomendable para conjuntos de datos muy grandes. En la extracción de esta gráfica, sólo utilizamos todos los datos existentes la cual este grafica es la

general de toda la Dataset. Finalmente pudimos hacer nuevas predicciones y a raíz de los resultados, comprender mejor la problemática planteada. En la presente grafica nos muestra la exactitud y el entrenamiento que tiene para el desarrollo del proyecto, explicando lo siguiente el test Exactitud nos indica hasta donde se utilizara la predicción de 0.1 hasta 0.3, pero en la de entrenamiento nos da un valor de predicción de un 0.6 creíble que se utilizara en toda la ejecución del algoritmo desarrollado.

En las siguientes graficas se depura el algoritmo desarrollado para la clasificación, de la cual se imprime 4 consultas por cada K vecinos ingresados.

Tabla 19: Ejecución del algoritmo con 1 vecino k-NN y Random Forest n=? Variable de la Dataset zomato.

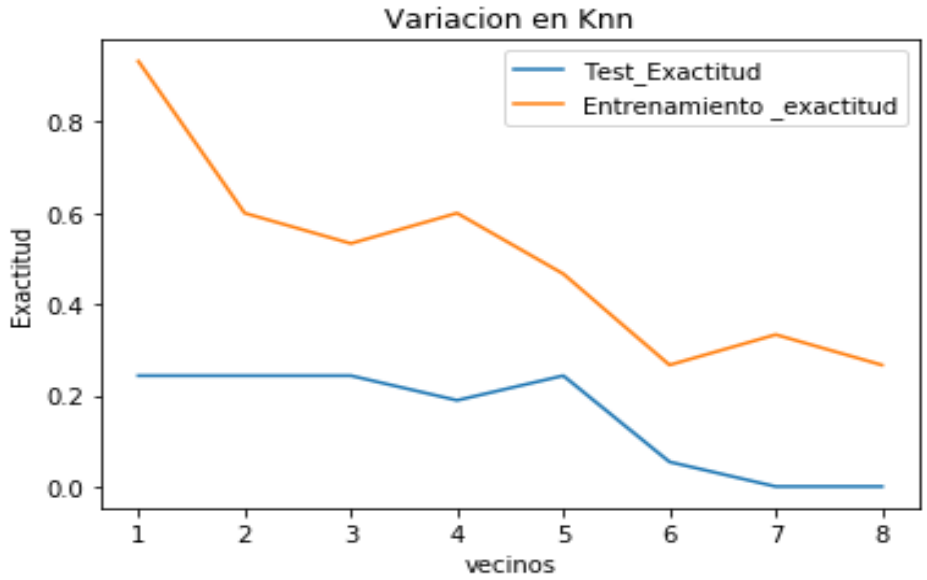
KNN
0.21621621621621623
['3.9/5']
Indiranagar
Indian Chicken Biryani Point

RANDOM FOREST
0.21621621621621623
['3.9/5']
Indiranagar
Indian Chicken Biryani Point

Tu seleccionaste: Cafe, Chicken Grill
Las recomendaciones para esta selección son:
1. Indian Chicken Biryani Point
2. Indian Chicken Biryani Point
Accuracy: 0.352092352 (+/- 0.17) [KNN]
Accuracy: 0.362193362 (+/- 0.16) [Random Forest]

Se obtuvo los siguientes resultados de K-NN y Random Forest, donde se predice en la primera metodología un valor de 0.21 con un puesto de categoría de 3.9 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.21 con un puesto de categoría 3.9 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología k-NN tiene la misma exactitud que Random Forest.

Grafico 13: Variación en k-NN con 2 Vecinos de la Dataset zomato



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.3, pero en la de entrenamiento nos da un valor de predicción de 0.6 que se va incrementando como indica en el Grafico 13 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 20: Ejecución del algoritmo con 2 vecinos k-NN y Random Forest n=? Variable de la Dataset zomato.

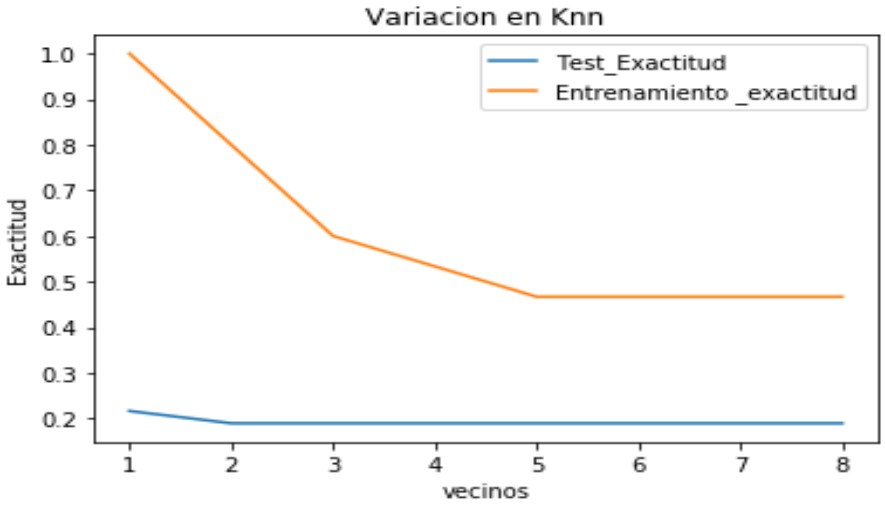
KNN
0.4324324324324326
['3.7/5']
Sarjapur Road
Bharanis

RANDOM FOREST
0.24324324324324326
['3.7/5']
Sarjapur Road
Bharanis

Tu seleccionaste: Cafe, Chicken Grill
Las recomendaciones para esta selección son:
1. Bharanis
2. Bharanis
Accuracy: 0.330687831 (+/- 0.09) [KNN]
Accuracy: 0.259523810 (+/- 0.06) [Random Forest]

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.4 con un puesto de categoría 3.7 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera en la segunda metodología predice un valor de 0.4 con un puesto de categoría 3.7 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que al tener el mismo test de exactitud como indica en el Grafico 14 la metodología Random Forest recurre a una búsqueda muy similar que se busca.

Grafico 14: Variación en k-NN con 3 Vecinos de la Dataset zomato



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.2, pero en la de entrenamiento nos da un valor de predicción de 0.24 que se va incrementando como indica en el Grafico 6 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 21: Ejecución del algoritmo con 3 vecinos k-NN y Random Forest n=? Variable de la Dataset zomato.

KNN
0.1891891891891892
['3.8 /5']
Banaswadi
Chickpet Donne Biriyani Mane

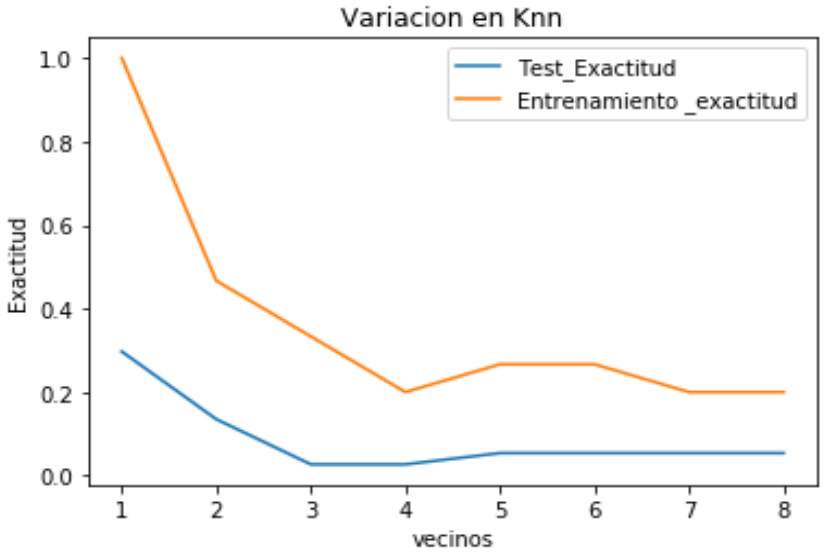
RANDOM FOREST
0.5324324324324326
['4.3 /5']
Indiranagar

BTDT? Been There Done That

Tu seleccionaste: Cafe, Chicken Grill
Las recomendaciones para esta selección son:
1. Chickpet Donne Biriyani Mane
2. BTDT? Been There Done That
Accuracy: 0.427579365 (+/- 0.20) [KNN]
Accuracy: 0.403769841 (+/- 0.20) [Random Forest]

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.18 con un puesto de categoría 3.8 y el Tema de la recomendación de la que más se asemeja al de la selección, de la misma manera la segunda metodología predice un valor de 0.53 con un puesto de categoría 4.3 y el Tema de la recomendación que más se asemeja al de la búsqueda.

Grafico 15: Variación en k-NN con 4 Vecinos de la Dataset zomato



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.18 hasta 0.24, pero en la de entrenamiento nos da un valor de predicción de 0.6 que se va incrementando como indica en el Grafico 15 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 22: Ejecución del algoritmo con 4 vecinos k-NN y Random Forest n=? Variable de la Dataset zomato.

```
--------------------------------------------------
KNN
0.02702702702702703
['2.8/5']
Marathahalli
Shree Saraswathi Sweets Centre
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.2972972972972973
['3.8/5']
Koramangala 4th Block
Kritunga Restaurant
--------------------------------------------------
--------------------------------------------------
Tu seleccionaste: Cafe, Chicken Grill
Las recomendaciones para esta selección son:
1. Shree Saraswathi Sweets Centre
2. Kritunga Restaurant
Accuracy: 0.271102151 (+/- 0.09) [KNN]
Accuracy: 0.421774194 (+/- 0.13) [Random Forest]
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.02 con un puesto de categoría 2.8 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.29 con un puesto de categoría 3.8 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología Random Forest supera a k-NN por 0.27 de predicción, cabe recalcar que al tener k-NN 4 vecinos su exactitud en la Grafica es muy corta, con estas pruebas se añade que ambos métodos son compatibles y precisos al momento de una clasificación de n datos.

4. Resultados de la cuarta base de datos

El presente proyecto de investigación los resultados fueron extraídos mediante el desarrollo del algoritmo de clasificación, para el conjunto de entrenamiento se tomó en cuenta un rango de 1 a 9 para la metodología k-NN y para Random Forest una variable m se seleccionan aleatoriamente.

Tabla 23: Datos de Predicción K-NN y RF de la Dataset AppStore

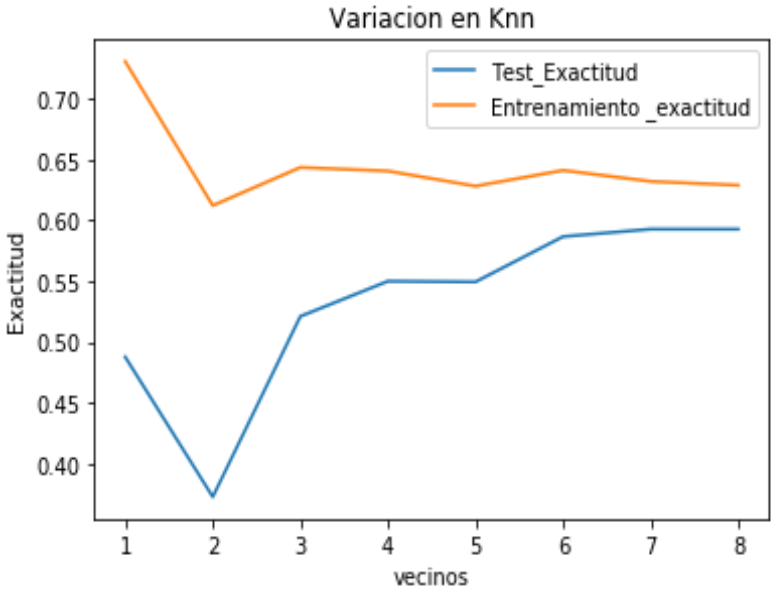
Dataset	**Predicción K-NN**		**Random Forest**
11097(Fracción de 0.001 = 39)	1 Vecino	0.48	0.54
11097(Fracción de 0.001 = 39)	2 Vecinos	0.38	0.54
11097(Fracción de 0.001 = 39)	3 Vecinos	0.6	0.6
11097(Fracción de 0.001 = 39)	4 Vecinos	0.55	0.55

De un total de 11097 datos que está compuesto el Dataset se tomó una fracción del 0.001 datos para la prueba y en la que la predicción que se utiliza en K-NN es de distintos vecinos de 1 a 4 por lo que varían los mismos, en los cuales la predicción menor es de 0.38 con 2 vecino, la siguiente predicción tiene el valor de 0.48 que tiene 1 vecino, la siguiente predicción tiene el valor de 0.6 que tiene 3 vecinos y por último la predicción de valor 0.55 con 4 vecinos que es de un pronóstico preciso.

En Random Forest obtuvimos los siguientes resultados tomando en cuenta así como se realizó en K-NN con un total de 7.198 datos y de igual manera con predicciones de 1 a 4 vecinos que sus valores varían entre sí, tenemos que 3 vecino posee un valor de 0.6 la cual es la mayor de todas, con 1 vecinos la predicción tiene el valor de 0.54 la cual es el pronóstico preciso a continuación con 2 vecinos tenemos que el Random Forest tiene el valor de 0.52 al igual que con 4 vecinos.

Una vez evaluada la capacidad predictiva del algoritmo *K-NN*, y los árboles de decisión simples obtenidos mediante el paquete *KNeighborsClassifier*, estimamos el modelo que obtendríamos si ejecutásemos *n* árboles de decisión simultáneamente para *n*= **7.198** en nuestro caso mediante el algoritmo *randomForest*. El algoritmo *randomForest* es un método de estimación combinado, donde el resultado de la estimación se construye a partir de los resultados obtenidos mediante el cálculo de *n* árboles donde los predictores son incluidos al azar.

Grafico 16: Variación en k-NN con 1 Vecino de la Dataset AppStore



En esta representación creamos un modelo con Python para procesar y clasificar puntos de un conjunto de entrada con el algoritmo k-Nearest Neighbor. Cómo su nombre en inglés lo dice, se evalúan los k vecinos más cercanos para poder clasificar nuevos puntos. Al ser un algoritmo supervisado debemos contar con suficientes muestras etiquetadas para poder entrenar el modelo con buenos resultados. Este algoritmo es bastante simple y como vimos antes necesitamos muchos recursos de memoria y cpu para mantener el Dataset vivo y evaluar nuevos puntos. Esto no lo hace recomendable para conjuntos de datos muy grandes. En la extracción de esta gráfica, sólo utilizamos todos los datos existentes la cual este grafica es la general de toda la Dataset. Finalmente pudimos hacer nuevas predicciones y a raíz de los resultados, comprender mejor la problemática planteada. En la presente grafica nos muestra la exactitud y el entrenamiento que tiene para el desarrollo del proyecto, explicando lo siguiente el test Exactitud nos indica hasta donde se utilizara la predicción de 0.1 hasta 0.54, pero en la de entrenamiento nos da un valor de predicción de un 0.6 creíble que se utilizara en toda la ejecución del algoritmo desarrollado.

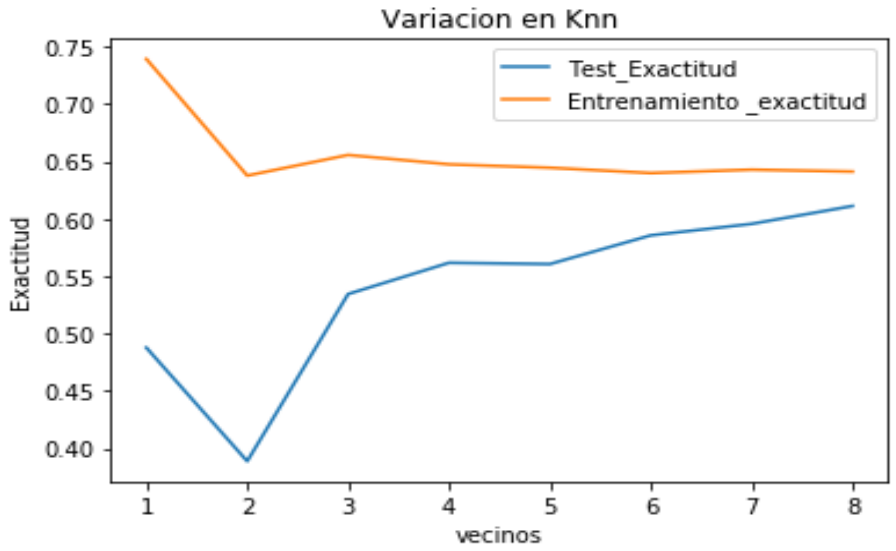
En las siguientes graficas se depura el algoritmo desarrollado para la clasificación, de la cual se imprime 4 consultas por cada K vecinos ingresados.

Tabla 24: Ejecución del algoritmo con 1 vecino k-NN y Random Forest n=? Variable de la Dataset AppStore.

```
--------------------------------------------------
KNN
0.48777777777777775
[4]
Shopping
eBay: Best App to Buy, Sell, Save! Online Shopping
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.5466666666666666
[5]
Weather
WeatherBug - Local Weather, Radar, Maps, Alerts
--------------------------------------------------
--------------------------------------------------
Las recomendaciones para esta selección son:
1. eBay: Best App to Buy, Sell, Save! Online Shopping
2. WeatherBug - Local Weather, Radar, Maps, Alerts
Accuracy: 0.48 (+/- 0.02) [KNN]
Accuracy: 0.54 (+/- 0.00) [Random Forest]
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.48 con un puesto de categoría 4 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.54 con un puesto de categoría 5 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología Random Forest supera a k-NN por 0.06 de exactitud.

Grafico 17: Variación en k-NN con 2 Vecinos de la Dataset AppStore



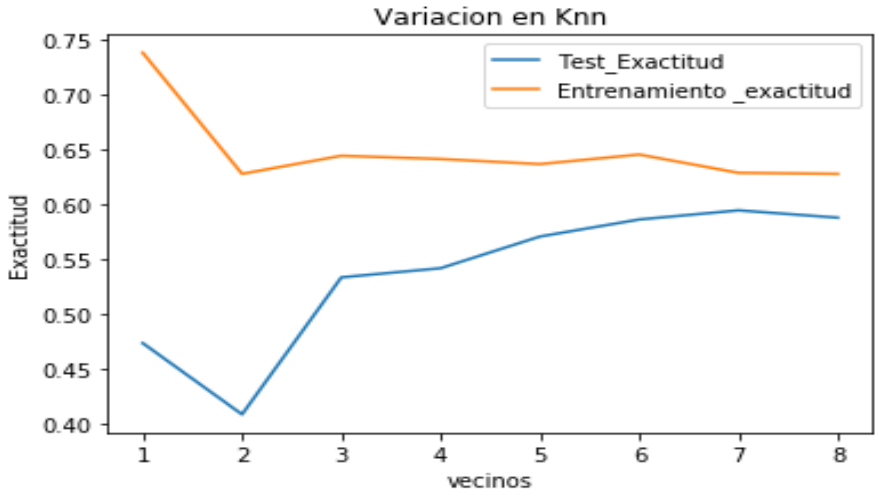
En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.01 hasta 0.48, pero en la de entrenamiento nos da un valor de predicción de 0.6 que se va incrementando como indica en el Grafico 17 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 25: Ejecución del algoritmo con 2 vecinos k-NN y Random Forest n=? Variable de la Dataset AppStore.

```
--------------------------------------------------
KNN
0.3888888888888889
[0]
Finance
Shanghai Mahjong
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.5427777777777778
[0]
Finance
Shanghai Mahjong
--------------------------------------------------
--------------------------------------------------
Las recomendaciones para esta selección son:
1. Shanghai Mahjong
2. Shanghai Mahjong
Accuracy: 0.38 (+/- 0.04) [KNN]
Accuracy: 0.53 (+/- 0.01) [Random Forest]
--------------------------------------------------
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.3 con un puesto de categoría 0 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.54 con un puesto de categoría 0 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que al tener el mismo test de exactitud como indica en el Grafico 16 la metodología Random Forest recurre a una búsqueda muy similar que se busca.

Grafico 18: Variación en k-NN con 3 Vecinos de la Dataset AppStore



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.3, pero en la de entrenamiento nos da un valor de predicción de 0.60 que se va incrementando como indica en el Grafico 18 por ende Random Forest recurre a una búsqueda muy similar que se busca.

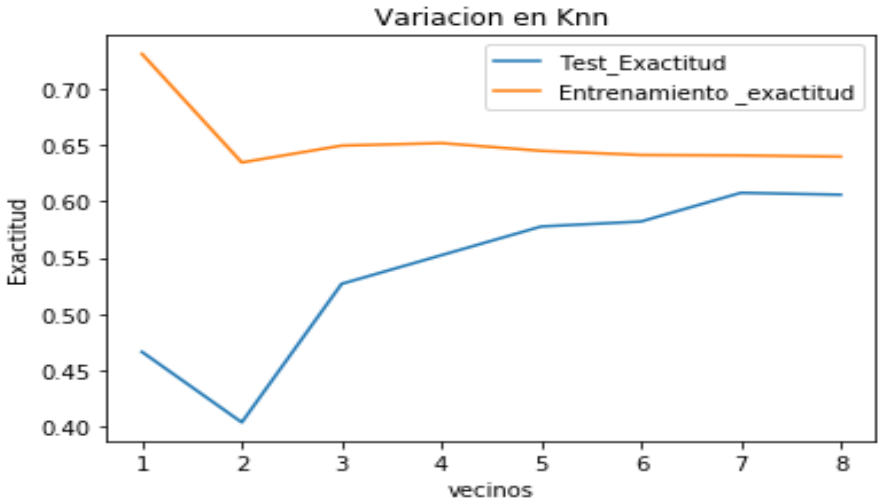
Tabla 26: Ejecución del algoritmo con 3 vecinos k-NN y Random Forest n=? Variable de la Dataset AppStore.

```
--------------------------------------------------
KNN
0.633333333333333
[5]
Weather
WeatherBug - Local Weather, Radar, Maps, Alerts
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.63336666666666
[5]
Weather
WeatherBug - Local Weather, Radar, Maps, Alerts
--------------------------------------------------
--------------------------------------------------
Las recomendaciones para esta selección son:
1. WeatherBug - Local Weather, Radar, Maps, Alerts
2. WeatherBug - Local Weather, Radar, Maps, Alerts
Accuracy: 0.6 (+/- 0.04) [KNN]
Accuracy: 0.6 (+/- 0.01) [Random Forest]
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice en la primera metodología un valor de 0.6 con un puesto de categoría 5 y el Tema de la recomendación de la que más se

asemeja al de la búsqueda, de la misma manera la segunda algoritmo predice un valor de 0.6 con un puesto de categoría 5 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que al tener el mismo test de exactitud como indica en el Grafico 19 la metodología Random Forest recurre a una búsqueda muy similar que se busca.

Grafico 19: Variación en k-NN con 4 Vecinos de la Dataset AppStore



En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente en esta predicción se comenta que el test de exactitud nos indica hasta donde se utilizara de 0.1 hasta 0.53, pero en la de entrenamiento nos da un valor de predicción de 0.60 que se va incrementando como indica en el Grafico 19 por ende Random Forest recurre a una búsqueda muy similar que se busca.

Tabla 27: Ejecución del algoritmo con 4 vecinos k-NN y Random Forest n=? Variable de la Dataset AppStore

```
--------------------------------------------------
KNN
0.5522222222222222
[0]
Finance
Shanghai Mahjong
--------------------------------------------------
--------------------------------------------------
RANDOM FOREST
0.5533333333333333
[5]
Weather
WeatherBug - Local Weather, Radar, Maps, Alerts
--------------------------------------------------
--------------------------------------------------
Las recomendaciones para esta selección son:
1. Shanghai Mahjong
```

```
2. WeatherBug - Local Weather, Radar, Maps, Alerts
Accuracy: 0.54 (+/- 0.04) [KNN]
Accuracy: 0.53 (+/- 0.01) [Random Forest]
--------------------------------------------------
```

Se obtuvo resultados de k-NN y Random Forest, donde se predice la primera metodología un valor de 0.55 con un puesto de categoría 0 y el Tema de la recomendación de la que más se asemeja al de la búsqueda, de la misma manera la segunda metodología predice un valor de 0.55 con un puesto de categoría 5 y el Tema de la recomendación que más se asemeja al de la búsqueda. En esta predicción se comenta que la metodología Random Forest supera a k-NN por 0.001 de predicción, cabe recalcar que al tener k-NN 4 vecinos su exactitud en la Grafica es muy corta llego a predecir el mismo tema que Random Forest, con estas pruebas se añade que ambos métodos son compatibles y precisos al momento de una clasificación de n datos.

EVALUACIÓN DE ALGORITMOS

Al realizarse los análisis y resultados se procede a la evaluación de un algoritmo que son serie de pasos bien definidos que ayudan a llegar a la solución de algún problema que tiene como propósito medir su desempeño, considerando el tiempo de ejecución y los recursos empleados, para obtener una solución satisfactoria. En muchas ocasiones se le da mayor peso al tiempo que tarda un algoritmo en resolver un problema.

Para medir el tiempo de ejecución, el algoritmo se puede transformar a un programa de computadora. Aquí se involucran otros factores, como el lenguaje de programación elegido, sistema operativo empleado, habilidad del programador, etcétera.

Pero también hay otra forma la cual aplicaremos que; se puede medir el número de operaciones que realiza un algoritmo considerando el tamaño de las entradas al mismo (N). Entre más grande es la entrada mayor será su tiempo de ejecución.

Mediante la comprobación debemos asegurarnos que el algoritmo hace lo que debe y funciona correctamente. La comprobación puede realizarse ejecutando el algoritmo que lo hemos hecho y sacarlo con las predicciones más altas de las 4 base de datos y sacar al final una predicción final como se muestra en la tabla 30.

Para proceder con la evaluación de algoritmos la cual permitirá evaluar los datos individuales para cada uno de los algoritmos con su respectiva base de datos que se muestran y verificar el cumplimiento del objetivo establecido, para lo cual se tomó en cuenta los resultados que presentan los mismos, ya que son los que más se acercan al resultado principal y en los cuales tenemos dos algoritmos que son los siguientes el K-NN y el Random Forest, en los cuales en el

primer algoritmo nos muestra un resultado que posee un rango entre 0.0 el cual es de menor precisión y 0.4 el cual es el de mayor precisión, continuando con el algoritmo Random Forest el rango se encuentra entre el de menor precisión que es 0.0 hasta el de mayor precisión con 0.9 en los cuales se realizó respectivamente 15 pruebas tanto a los K-NN como a los Random Forest en los que variaba su resultado al momento de ejecutar su respectiva prueba como se muestra en la tabla 29 cada prueba con su resultado en los K-NN y en el Random Forest.

Tabla 28: Predicciones a los algoritmos K-NN y Random Forest con los Dataset de pruebas.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base de Datos GBVideos															
K-NN	0.2	0.4	0.0	0.1	0.5	0.2	0.2	0.4	0.3	0.2	0.1	0.2	0.1	0.1	0.3
RF	0.2	0.7	0.1	0.4	0.6	0.2	0.2	0.4	0.4	0.3	0.3	0.2	0.4	0.3	0.3
Base de Datos vg1															
K-NN	0.7	0.5	0.5	0.5	0.5	0.4	0.4	0.3	0.5	0.7	0.4	0.6	0.5	0.4	0.5
RF	0.6	0.9	0.7	0.7	0.6	0.7	0.9	0.6	0.5	0.7	0.7	0.7	0.6	0.6	0.5
Base de Datos Zomato															
K-NN	0.21	0.4	0.18	0.2	0.18	0.02	0.08	0.4	0.05	0.24	0.32	0.29	0.10	0.16	0.16
RF	0.21	0.24	0.24	0.5	0.29	0.4	0.16	0.4	0.18	0.21	0.35	0.24	0.10	0.16	0.21
Base de Datos AppleStore															
K-NN	0.48	0.38	0.53	0.6	0.47	0.41	0.5	0.5	0.49	0.41	0.53	0.55	0.48	0.39	0.52
RF	0.54	0.54	0.52	0.6	0.53	0.53	0.54	0.54	0.55	0.55	0.53	0.54	0.53	0.54	0.53

Análisis de Base de datos GBVideos

Tomando en cuenta los resultados de las siguientes pruebas 1, 2, 4, 5 en la base de datos de GBVideos tenemos que en la prueba número 1 en el K-NN y en el RF tenemos una igualdad en sus predicciones con 0.2 eso quiere decir que los 2 se acercan más a la búsqueda que deseamos, en la siguiente prueba realizada tenemos que en K-NN tenemos un resultado de 0.4 y en el RF es de 0.7 el cual es la mayor predicción que se acerca a la búsqueda deseada, en la prueba 4 tenemos que en el K-NN su predicción es de 0.1 y es menor que la predicción que poseemos en el RF que es de 0.4 que se acerca más a la búsqueda realizada, en la prueba 5 tenemos que el K-NN tiene como predicción 0.5 y el RF el resultado de 0.6 el cual es la mayor predicción que se tiene en las pruebas que se tomó en cuenta, por ende este resultado es el que más se acerca a la búsqueda que se realizó.

Análisis de Base de datos vg1

Así mismo tomando en cuenta las pruebas 1, 2, 4, 5 realizadas con la base de datos vg1 tenemos que la prueba número 1 en el K-NN y en el Random Forest tenemos que en la primera el resultado es de 0.7 el cual es la mayor predicción que se acerca a la búsqueda deseada, en la siguiente prueba realizada tenemos que en K-NN tenemos un resultado de 0.5 y en el Random Forest es de 0.9 el cual es la mayor predicción que se acerca a la búsqueda deseada, en la prueba 4 tenemos que en el K-NN su predicción es de 0.5 y es menor que la predicción que poseemos en el RF que es de 0.7 que se acerca más a la búsqueda realizada, en la prueba 5 tenemos que el K-NN tiene como predicción 0.5 y el Random Forest el resultado de 0.6 el cual es la mayor predicción que se tiene en las pruebas que se tomó en cuenta, por ende este resultado es el que más se acerca a la búsqueda que se realizó.

Análisis de Base de datos Zomato

En la pruebas 1, 2, 4, 5 realizadas con la base de datos de Zomato tenemos que en la prueba número 1 en el K-NN y en el RF tenemos una igualdad en sus predicciones con 0.21 eso quiere decir que los 2 se acercan más a la búsqueda que deseamos, en la siguiente prueba realizada tenemos que el k-nn en sus predicción tiene 0.4 y es mayor que el resultado del Random Forest que tiene 0.24 , en la prueba numero 4 tenemos que en el K-NN su predicción es de 0.02 y es menor que la predicción que poseemos en el RF que es de 0.5 que se acerca más a la búsqueda realizada, en la prueba 5 tenemos que el K-NN tiene como predicción 0.18 y el RF el resultado de 0.29 el cual es la mayor predicción que se tiene en las pruebas que se tomó en cuenta, por ende este resultado es el que más se acerca a la búsqueda que se realizó.

Análisis de Base de datos AppleStore

Así mismo tomando en cuenta las pruebas 1, 2, 4, 5 realizadas con la base de datos AppleStore tenemos que la prueba muermo 1 el resultado es de 0.48 y en el Random foreste tenemos que su predicción es de 0.54 el cual es la mayor predicción que se acerca a la búsqueda deseada, en la prueba numero 2 tenemos que en el K-NN su predicción es de 0.38 y es menor que la predicción que poseemos en el RF que es de 0.54 que se acerca más a la búsqueda realizada, en la siguiente prueba realizada tenemos una igualdad en sus predicciones con 0.6 eso quiere decir que los 2 se acercan más a la búsqueda que deseamos, en la prueba 5 tenemos que el K-NN tiene como predicción 0.47 y el Random Forest el resultado de 0.53 el cual es la mayor

predicción que se tiene en las pruebas que se tomó en cuenta, por ende este resultado es el que más se acerca a la búsqueda que se realizó.

Mediante la comprobación debemos asegurarnos que el algoritmo hace lo que debe y funciona correctamente. La comprobación puede realizarse ejecutando el algoritmo con una colección de entradas que cubran un gran abanico de datos ingresados para realizar la consulta. Otra estrategia, más segura pero más compleja, es la verificación formal, en la que se utilizan la lógica para demostrar matemáticamente propiedades de los algoritmos. Esto es lo que hemos hecho con nuestras soluciones aunque también existen métodos para comprobar que los algoritmos finales siguen fielmente la solución. Los algoritmos son relativamente sencillos, pero puede no serlo cuando aumenta la dificultad del programa y no se ha realizado un correcto refinamiento progresivo. De tal manera con la tabla realizada podemos observar que al fusionar las dos metodologías siempre habrá una variación de predicción lo cual Random Forest tiene más probabilidad de encontrar una predicción coherente y más precisa en una gran cantidad de datos expuestos.

Tabla 29: Evaluación del algoritmo

DataSet ***Algoritmos***	***GBVideos***	***Vg1***	***Zomato***	**AppleStore**	***PREDICCION***
k-NN	*0.5*	*0.7*	*0.4*	*0.6*	*0.6*
Random Forest	*0.7*	*0.9*	*0.5*	*0.6*	*0.7*

La evaluación del algoritmo se la realizó por medio de pruebas de la eficiencia del algoritmo en la cual se utilizó cuatro bases de datos; GBVideos, Vg1, Zomato,AppleStore , en los cuales se obtuvo los siguientes resultados con cada uno de sus algoritmos Random Forest y k-NN.

En la primera evaluación realizada con la base de datos GBVideos al algoritmo k-nn se obtuvo el resultado de 0.5 que es menor que el resultado que se presenta en el algoritmo Random Forest, así mismo la evaluación realizada al algoritmo Random Forest se obtuvo el resultado de 0.7 el cual es la mayor predicción que se acerca a la búsqueda deseada.

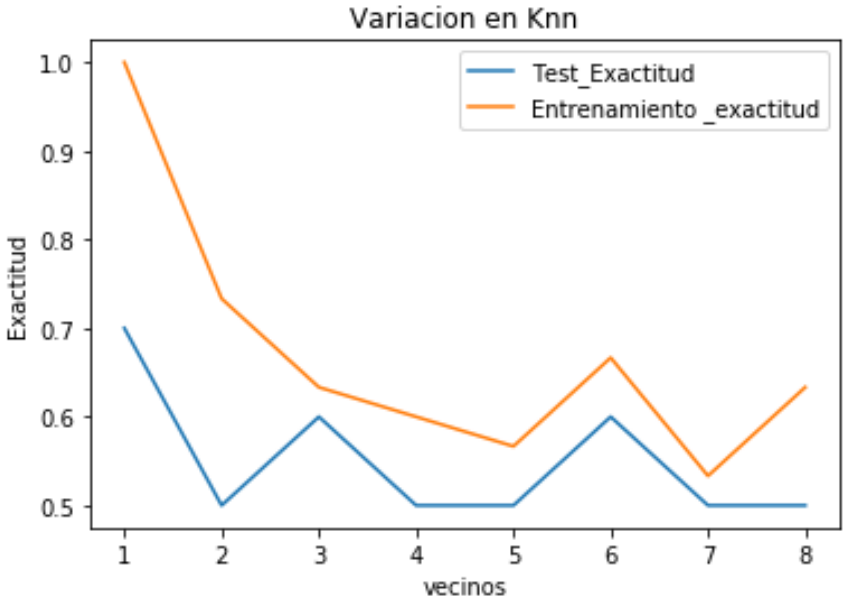
En la segunda evaluación realizada con la base de datos Vg1 al algoritmo k-nn se obtuvo el resultado de 0.7 el cual es menor que el resultado que se presenta en el algoritmo Random Forest, así mismo la evaluación realizada al algoritmo Random Forest se obtuvo el resultado de 0.9 el cual es la mayor predicción que se acerca a la búsqueda deseada.

En la tercera evaluación realizada con la base de datos Zomato al algoritmo k-nn se obtuvo el resultado de 0.4 el cual es menor que el resultado que se presenta en el algoritmo Random Forest, así mismo la evaluación realizada al algoritmo Random Forest se obtuvo el resultado de 0.5 el cual es la mayor predicción que se acerca a la búsqueda deseada.

En la cuarta evaluación realizada con la base de datos AppleStore al algoritmo k-nn y Random Forest tenemos una igualdad en sus predicciones con 0.6 eso quiere decir que los 2 algoritmos se acerca más a la búsqueda deseada.

En la siguiente grafica se presenta la evaluación del algoritmo desarrollado para la clasificación.

Grafico 20: Evaluación del algoritmo



En la realización del algoritmo de clasificación del lenguaje natural llegamos a un análisis que el algoritmo Random Forest es más eficiente y por ende tiene más posibilidad de encontrar una predicción coherente y precisa en la ejecución de pruebas en una mayor cantidad de datos expuestos y el mismo realizado la evaluación es preciso al momento de una clasificación de n datos, los cuales están representadas en las cuatro bases de datos ya mencionadas anteriormente.

En la presente gráfica nos muestra el test de exactitud y el entrenamiento que tiene para el desarrollo del presente proyecto de investigación, explicando lo siguiente el test de Exactitud nos indica hasta donde se utilizara la predicción de 0.1 hasta 0.3, pero en la de entrenamiento nos da un valor de predicción de un 0.7 creíble que se utilizó en toda la ejecución del algoritmo desarrollado.

12. IMPACTOS

12.1. Impacto tecnológico

El mundo atraviesa una importante transformación tecnológica. Los desarrollos de los últimos años en distintas áreas de la ciencia fueron exponenciales y generaron saltos de calidad en materia productiva. Por tal razón el nuevo algoritmo se pretende llegar un seguimiento por usuarios e investigadores.

12.2. Impacto social

Tiene un gran impacto en la sociedad por que al tener un algoritmo de clasificación precisa y fácil de utilizar ayuda a usuarios e investigadores para el uso del algoritmo.

13. PRESUPUESTO PARA LA ELABORACIÓN DEL PROYECTO

Tabla 30: Presupuesto, Gasto Computadora, Gasto Internet, Gasto Impresiones.

PRESUPUESTO PARA EL PROYECTO			
GASTO USO COMPUTADORA			
DESCRIPCIÓN	**PRECIO/HORA**	**HORAS OCUPADAS**	**TOTAL**
Computadora	0,60	200	120,00
TOTAL GASTO USO DE COMPUTADORA			**120,00**
GASTO INTERNET			
DESCRIPCIÓN	**PRECIO/HORA**	**HORAS OCUPADAS**	**TOTAL**
Uso del internet	0,60	80	48,00
TOTAL GASTO INTERNET			**48,00**
GASTO IMPRESIONES			
DESCRIPCIÓN	**VALOR/IMPRESIÓN**	**NUMERO/HOJAS**	**TOTAL**
Impresiones a blanco y negro	0,05	500	25,00
Impresiones a color	0,10	300	30,00

TOTAL GASTO IMPRESIONES	**55,00**
Total de gastos	**223**

13.1. Gastos directos

Tabla 31: Gastos Directos.

DETALLE	**CANTIDAD**	**PRECIO UNITARIO**	**PRECIO TOTAL**
Hojas de papel bond	2	4,00	8,00
Esferos	10	0,40	4,00
Lápices	2	0,50	1,00
Borrador	2	0,50	1,00
Grapadora	1	3,00	3,00
Carpetas	2	0,75	1,50
Cd	2	0,50	1,00
Usb/flash	1	8,50	8,50
Anillado	3	4,00	12,00
TOTAL			40,00

13.2. Gastos indirectos

Tabla 32: Gastos Indirectos.

DETALLE	**PRECIO TOTAL**
Transporte	30,00
Alimentación	40,00
TOTAL	**70,00**

13.3. Resumen de gastos

Tabla 33: Resumen de los Gastos.

RESUMEN GASTOS	
Total Gasto Uso Computadora	60,00
Total Gasto Impresiones	12,00
Total Gasto Internet	36,00
Gastos Indirectos	40,00
Gastos Directos	70,00
TOTAL PRESUPUESTO	**218,00**

14. CONCLUSIONES Y RECOMENDACIONES

14.1. Conclusiones

Después de haber concluido con éxito con el presente proyecto de investigación se puede tener las siguientes conclusiones.

- En conclusión la revisión de la literatura, dio la idea para estructurar el presente proyecto y desarrollar el algoritmo el mismo que ayudo como referencia para obtener conocimiento de los mecanismos empleados en los sistemas de búsqueda web.
- Se logró crear un algoritmo realizando evaluaciones a los algoritmos k-NN y Random Forest y tal algoritmo basado en Random Forest es la más confiable y precisa para la clasificación del lenguaje natural con gran cantidad de datos.
- Se diseñó un algoritmo de búsqueda que permite realizar aproximaciones a sugerencias realizadas por los usuarios. Para ello se aplicaron técnicas de inteligencia artificial que permitieron mayor eficiencia en la búsqueda. Las técnicas aplicadas fueron Random Forest donde se obtuvo el 0.7 o 70% y k-NN en donde se obtuvo el 0.6 o 60%.
- Se está escribiendo un artículo científico para la revista International Journal of Engineering & Technology

14.2. Recomendaciones

Al culminar con la investigación del presente proyecto de investigación se recomienda que:

- Continuar con el desarrollo del algoritmo, que realicen las adaptaciones necesarias y su perfeccionamiento con nuevas funcionalidades.
- Promover el desarrollo de este tipo de proyectos que van en beneficio de la institución, debido a que permite que los futuros profesionales de la Universidad Técnica de Cotopaxi realicen su proyecto de investigación para solucionar problemáticas concretas y reales.
- Podría ser considerado como apoyo para estudiantes a fin de que se establezca en proyectos futuros.

15. BIBLIOGRAFÍA

Agudo, S. (2015). ¿Qué es la web semántica? Retrieved April 22, 2019, from https://rootear.com/web/que-es-la-web-semantica

Alvarez, S. (2006). Tipos de lenguajes, http://www.desarrolloweb.com/articulos/2358.php.

Applications, S. (2016). Introduction to Scientific Computing and Visualization in Python.

Bahit, E. (2013). Capítulo 12. Bases de datos en Python con MySQL (Python para principiantes). Retrieved May 9, 2019, from https://uniwebsidad.com/libros/python/capitulo-12

Benavides, P. (2013). Clasificación, ordenación y búsqueda inteligente, 42.

Cortez Vásquez, A., Vega Huerta, H., & Pariona Quispe, J. (2009). Procesamiento de lenguaje natural. *Revista de Investigación de Sistemas e Informática*, 45–54.

de Computadores, P., & Alfaro Olave, T. (2015). *Algoritmos de Búsqueda y Ordenamiento*. Retrieved from https://www.inf.utfsm.cl/~noell/IWI-131-p1/Tema8b.pdf

Delgado, F. J. P., & Amador, C. E. V. (2014). *Algoritmos Resueltos Con Diagramas De Flujo y Pseudocódigo*.

Derechos, G. (2012). Algoritmos de búsqueda básicos, 1–7.

Discovery, K. (2015). Minería de datos, 1–8.

En, A., & Forests, R. (2010). 1.6 analisis en random forests.

Estudio, U. (2018). ALGORITMOS E INTELIGENCIA ARTIFICIAL EN.

Flores, J. B. A. (2017). Modelos de minería de datos: random forest y adaboost, para identificar los factores asociados al uso de las TIC (internet, telefonía Fija y televisión de paga) en los hogares del Perú. 2014.

Gantz, J., & Reinsel, D. (2011). Extracting Value from Chaos State of the Universe. *IDC IView*, (June), 1–12. https://doi.org/10.1007/s10916-016-0565-7

Gutiérrez, C., & Hurtado, C. (2015). Web Semántica : Realidades y Perspectivas.

Hidalgo-Delgado, Y., & Rodríguez-Puente, R. (2013). La web semántica: una breve revisión.

Revista Cubana de Ciencias Informáticas, *7*(1), 2227–1899. Retrieved from http://rcci.uci.cu

Illanes, G. (2010). Consistencia de random forests y otros clasificadores promediados, 1–22.

Introducción. (2009). Unidad I. 2 LENGUAJES DE PROGRAMACIÓN 1. Plataforma Teórico Conceptual.

Irvine, K. R. (2008). Lenguaje ensamblador, 320.

J, H. M. G. (2020). Hernandez,Gomez 2013, *32*(1), 87–96.

Machasilla, J. F. (2015). Desarrollo de un metabuscador para la búsqueda de apartamentos de alquiler habitual. *Universidad Carlos Iii De Madrid*, *1*, 77.

Maillo, J., Garc, S., Herrera, F., & Triguero, I. (2015). Un enfoque aproximado para acelerar el algoritmo de clasificaci ´ on Fuzzy kNN para Big Data, 1143–1148.

Marqués, E. J. (2015). *Contenido :*

Mihaela Juganaru Mathieu. (2013). *Introducción a la programación / Mihaela Juganaru Mathieu*. https://doi.org/10.1016/j.sleep.2010.07.006

Mora-florez, J., & Barrera-cárdenas, G. M. R. (2008). Evaluación del clasificador basado en los k vecinos más cercanos para la localización de la zona en falla en los sistemas de potencia Evaluating a k -nearest neighbours-based classifier for locating faulty areas in power systems. *Revista Ingeniería E Investigación*, *28*(3), 81–86.

Mora, H. M., Azorín López, ; J, Morenilla, ; A Jimeno, Jl, :, Romero, S., Pujol López, ; F, … Orts Escolano, ; S. (2016). *La Web Semántica Como Herramienta para el Apoyo a la Docencia Sistemas de Gestión del Conocimiento*. Retrieved from https://web.ua.es/es/ice/jornadas-redes-2015/documentos/tema-2/410778.pdf

Morales España, G., Mora Flórez, J., & Vargas Torres, H. (2008). κ-NN based regression strategy used to estimate the fault distance in radial power systems | Estrategia de regresión basada en el método de los k vecinos más cercanos para la estimación de la distancia de falla en sistemas radiales. *Revista Facultad de Ingenieria*, (45), 100–108.

MSDN. (2005). Acceso a datos con ADO.NET. *Visual Basic .NET*.

Nicolás Fidalgo Belmonte. (2012). Algoritmos de ordenación y búsqueda. *Universidad San*

Pablo CEU, 27.

Noi, P. T., & Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors (Switzerland)*, *18*(1). https://doi.org/10.3390/s18010018

Operacional, N. (2016). Conceptos de Lenguajes de Programación, 1–8.

Perez Castillo, J. N., Diaz Hernandez, M. F., & Rincon Mosquera, N. (2015). Semantic Web and its contribution to the strategy of Colombian State open data. *REVISTA CIENTIFICA*, *3*(23), 124–132. https://doi.org/10.14483/udistrital.jour.RC.2015.23.a10

Pérez Lozada, E., & Falcón, N. (2017). Diseño de prototipos experimentales orientados al aprendizaje de la optica. *Revista Eureka Sobre Enseñanza y Divulgación de Las Ciencias*, *6*(3), 452–465. https://doi.org/10.25267/rev_eureka_ensen_divulg_cienc.2009.v6.i3.10

Quezada Lucio, N. (2018). K-Vecino más próximo en una aplicación de clasificación y predicción en el Poder Judicial del Perú. *Pesquimat*, *21*(1), 11. https://doi.org/10.15381/pes.v21i1.15077

Ramos, F. M., & Velez, J. I. (2016). Integración de técnicas de procesamiento de lenguaje natural a través de servicios web.

Renear, A. H., Sacchi, S., Wickett, K. M., & Street, E. D. (2010). Definitions of Dataset in the Scientific and Technical Literature | Simone Sacchi - Academia.edu, 3–6.

Rodríguez Rodríguez, J. E., Rojas Blanco, E. A., & Franco Camacho, R. O. (2007). Clasificación de datos usando el método k-nn. *Vínculos*, *4*(1), 4–18.

Romero, M. (2019). Trending YouTube Video Statistics | Kaggle. Retrieved May 28, 2019, from https://www.kaggle.com/datasnaek/youtube-new#GBvideos.csv

Ruiz-Lobaina, E. M., & Romero-Suárez, P. L. (2017). Búsqueda de patrones para mejorar productos y servicios en las bibliotecas. *Investigacion Bibliotecologica*, *31*(72), 209–225. https://doi.org/10.22201/iibi.0187358xp.2017.72.57830

Ruiz, J. Z. (2018). Comparativa Y Análisis De Algoritmos.

Sánchez-Díaz, G., Escobar-Franco, U. E., Morales-Manilla, L. R., Piza-Dávila, I., Aguirre-

Salado, C., & Franco-Arcega, A. (2013). Incremental k most similar neighbor classifier for mixed data | Un algoritmo de clasificación incremental basado en los k vecinos más similares para datos mezclado. *Revista Facultad de Ingenieria*, (67), 19–30.

Sánchez, D. (2014). *Algoritmos para la Clasificación Multinstancia*.

Sánchez, M. del P. (2010). La comunicación y el lenguaje. *Revista Digital Para Profesionales de La Enseñanza*, 167–184.

Sanchez, S. (2016). Algoritmos de clasificación : K-NN, Árboles de decisión simples y múltiples (random forest). Retrieved May 27, 2019, from https://rstudio-pubs-static.s3.amazonaws.com/237547_0171c04b6d2e4550aea58853c056d29d.html

Soraya, Alvaredo, M., Coco, Lastras, A. D., Santiago, Zunzunegui, A., … Vaquero, M. (2018). Lenguajes De Programacion.

Soria, C. L., Pandolfi, D. R., Villagra, S. M., & Villagra, N. A. (2016). Algoritmos de Búsqueda Dispersa aplicados a problemas de Optimización Discreta. *Informes Científicos - Técnicos UNPA*, *8*(1), 132. https://doi.org/10.22305/ict-unpa.v8i1.154

Tablet, E. (2017). Tich_2017, *2017*, 1–10.

Utrera, R. G. (2017). Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos, (July). https://doi.org/10.13140/RG.2.2.34739.94241

Vaati, E. (2017). Cómo Leer y Escribir Archivos CSV en Python. Retrieved May 9, 2019, from https://code.tutsplus.com/es/tutorials/how-to-read-and-write-csv-files-in-python--cms-29907

Valencia, U. de. (2018). Algoritmos Y Programas, 81–135.

Yánez, M. (2010). UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA. *Tesis*, 88. Retrieved from http://dspace.utpl.edu.ec/bitstream/20.500.11962/20864/1/Iñiguez Jiménez Samuel Olegario.pdf

Zamoszczyk, C., De Luca, S., Ruiz Martínez, S., & Iturbide, L. (2018). Human Query Language. *Ciencia y Tecnología*, *1*(12), 37. https://doi.org/10.18682/cyt.v1i12.642

ANEXOS

16. ANEXOS

16.1. Datos informativos del tutor

UNIVERSIDAD TÉCNICA DE COTOPAXI

DATOS INFORMATIVOS PERSONAL DOCENTE

DATOS PERSONALES

APELLIDOS: Bravo Mullo
NOMBRES: Silvia Jeaneth
ESTADO CIVIL: Casada
CEDULA DE CIUDADANÍA: 0502437122
NÚMERO DE CARGAS FAMILIARES: 0
LUGAR Y FECHA DE NACIMIENTO: Latacunga, 28/11/82
DIRECCIÓN DOMICILIARIA: Ave. José María Velazco Ibarra y Diego Noboa, 3-39
TELÉFONO CONVENCIONAL: 032386446 **TELÉFONO CELULAR:** 0984473586
EMAIL INSTITUCIONAL: silvia.bravom@utc.edu.ec
TIPO DE DISCAPACIDAD:
 # DE CARNET CONADIS:

ESTUDIOS REALIZADOS Y TÍTULOS OBTENIDOS

NIVEL	TITULO OBTENIDO	FECHA DE REGISTRO	CÓDIGO DEL REGISTRO CONESUP O SENESCYT
TERCER	Ingeniera en Informática y Sistemas Computacionales, Universidad Técnica de Cotopaxi	1020-07-781174	2007-09-07
CUARTO	Master en Tecnologías para la Gestión y Práctica Docente	1027-12-86027930	2012-10-11
CUARTO	Doctor en Ingeniería de Sistemas e Informática	En curso	

HISTORIAL PROFESIONAL

UNIDAD ADMINISTRATIVA O ACADÉMICA EN LA QUE LABORA: CIYA
ÁREA DEL CONOCIMIENTO EN LA CUAL SE DESEMPEÑA: INFORMÁTICA
FECHA DE INGRESO A LA UTC: 05/05/2008

FIRMA

16.2. Datos informativos de estudiantes

1- DATOS PERSONALES

NOMBRES Y APELLIDOS: Francisco Bolívar Álvarez Lasso

FECHA DE NACIMIENTO: 21 de Noviembre de 1996

CEDULA DE CIUDADANÍA: 050378900-0

ESTADO CIVIL: Soltero

NUMEROS TELÉFONICOS: 0992848115

2.- ESTUDIOS REALIZADOS

NIVEL PRIMARIO : ESCUELA FISCAL "LUIS NAPOLEÓN DILLON"

NIVEL SECUNDARIO: INSTITUTO "COLEGIO NACIONAL EXPERIMENTAL PROVINCIA DE COTOPAXI"

NIVEL SUPERIOR UNIVERSIDAD TÉCNICA DE COTOPAXI

1- DATOS PERSONALES

NOMBRES Y APELLIDOS: Lenyn Santiago Mayo Pazuña

FECHA DE NACIMIENTO: 25 de Junio de 1992

CEDULA DE CIUDADANÍA: 0503217630

ESTADO CIVIL: Soltero

NUMEROS TELÉFONICOS: 0984725512

2.- ESTUDIOS REALIZADOS

NIVEL PRIMARIO : ESCUELA FISCAL "ISIDRO AYORA"

NIVEL SECUNDARIO: INSTITUTO TECNOLÓGICO SUPERIOR "VICENTE LEÓN"

NIVEL SUPERIOR UNIVERSIDAD TÉCNICA DE COTOPAXI

16.3. Código del algoritmo para la clasificación de datos.

```
import pandas as pd
import numpy as np

from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
from sklearn.neighbors import KNeighborsClassifier

Df=pd.read_csv('Nombre de la base de datos.csv')
#print(len(Df.index))
#print(Df.shape)
df=Df.sample(frac=0.01)
df=df.reset_index(drop=True)
#print(df.shape)
df=df.replace(np.nan,"0")
#print(df['tags'].head())
import string
df['ntags']=df['tags'].str.replace('[{}]'.format(string.punctuation),'')
#print(df['ntags'].head())


tfidf=TfidfVectorizer(stop_words='english')
df['ntags']=df['ntags'].fillna('')
tfidf_matrix=tfidf.fit_transform(df['ntags'])
#print(len(tfidf.vocabulary_))
#print(tfidf_matrix.shape)


cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)

indices = pd.Series(df.index, index=df['title']).drop_duplicates()

def get_recomendations(title, cosine_sim=cosine_sim):
    idx=indices[title]
    sim_scores=list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:5]
    movie_indices=[i[0] for i in sim_scores]
    return df['title'].iloc[movie_indices]

titulo=str(df.iloc[3]['title'])
vistos=str(df.iloc[8]['views'])
Megustas=str(df.iloc[9]['likes'])
disgustas=str(df.iloc[10]['dislikes'])
print("Tu seleccionaste: "+titulo+ " y tu recomendaciones son:")
print(get_recomendations(titulo))

pred=dict(zip(df.category_id.unique(),df.title.unique()))
#sprint(pred)
#Variables para la prediccion
X=df[['views','likes','dislikes']]
y=df['category_id']

X_train,X_test,y_train,y_test=train_test_split(X,y)
#print(X_train.describe())
```

```
neighbors=np.arange(1,9)
train_exactitud=np.empty(len(neighbors))
test_exactitud=np.empty(len(neighbors))
for i,k in enumerate(neighbors):
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,y_train)
    train_exactitud[i]=knn.score(X_train,y_train)
    test_exactitud[i]=knn.score(X_test,y_test)

plt.title('Variacion en Knn')
plt.plot(neighbors,test_exactitud,label='Test_Exactitud ')
plt.plot(neighbors,train_exactitud,label='Entrenamiento _exactitud')
plt.legend()
plt.xlabel('vecinos')
plt.ylabel('Exactitud')
plt.show()

#Metodo Knn
knn=KNeighborsClassifier(n_neighbors=2)
knn.fit(X_train,y_train)
pred1=knn.predict([[vistos,Megustas,disgustas]])


#Metodo Random Forest
rcf=RandomForestClassifier()
rcf.fit(X_train,y_train)
pred2=rcf.predict([[vistos,Megustas,disgustas]])


print("-----------------------------------------------------")
print("KNN")
print(knn.score(X_test,y_test))
print(pred1)
print(pred[pred1[0]])
print("-----------------------------------------------------")
print("-----------------------------------------------------")
print("RANDOM FOREST")
print(rcf.score(X_test,y_test))
print(pred2)
print(pred[pred2[0]])
print("-----------------------------------------------------")
print("-----------------------------------------------------")

print("La selección es ---- "+titulo+" ------ ")
print("Las recomendaciones para esta selección son: ")
print("1. "+pred[pred1[0]]+" ")
print("2. "+pred[pred2[0]]+" ")

print("-----------------------------------------------------")
print("-----------------------------------------------------")
```

16.4. Código del algoritmo

Aplicando todo la base de datos para sacar la predicción exacta con la impresión de la curva.

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split

datos=pd.read_csv('GBvideos.csv')
#datos=youtu.sample(frac=0.001)
#datos=datos.reset_index(drop=True)

datos=datos.replace(np.nan,"0")
#print(df)

#df=pd.DataFrame(datos)
pred=dict(zip(datos.category_id.unique(),datos.title.unique()))
print(pred)

print (datos['category_id'].value_counts())

X=datos[['views','likes','dislikes']]
y=datos['category_id']

X_train,X_test,y_train,y_test=train_test_split(X,y)
#print(X_train.describe())

from sklearn.neighbors import KNeighborsClassifier
neighbors=np.arange(1,9)
train_exactitud=np.empty(len(neighbors))
test_exactitud=np.empty(len(neighbors))
for i,k in enumerate(neighbors):
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,y_train)
    train_exactitud[i]=knn.score(X_train,y_train)
    test_exactitud[i]=knn.score(X_test,y_test)

plt.title('Variacion en Knn')
plt.plot(neighbors,test_exactitud,label='Test_Exactitud ')
plt.plot(neighbors,train_exactitud,label='Entrenamiento _exactitud')
plt.legend()
plt.xlabel('vecinos')
plt.ylabel('Exactitud')
plt.show()
```