1. ­­­­Introducción 6b59a6
   1. Difusores de contenido audiovisual

Desde la masificación de las redes sociales, en el pasado con plataformas como MySpace, y hoy con plataformas muy masificadas como Facebook o Instagram, se ha buscado la forma de sacar provecho de esta interacción. Una gran cantidad de negocios se han formado en torno a este concepto, principalmente relacionados con la publicidad. De la misma manera, la publicidad se ha ido adaptando a este nuevo entorno, haciéndose cada vez más sutil en algunos casos (*product placement*[[1]](#footnote-1)), y aún más invasiva en otros, como es el caso de portales de contenido viral con *banners* y *popups[[2]](#footnote-2)*. Con el paso del tiempo, los usuarios de internet se han vuelto reacios y menos susceptibles al segundo tipo de publicidad, ocurriendo fenómenos como la llamada **ceguera del banner[[3]](#footnote-3)**, lo que hace necesaria una forma de publicidad más sutil, o más atractiva que un *banner* tradicional. Es en torno a esto que se generan los difusores de contenidos.

Un difusor de contenido, como dice su nombre, cumple la función de **propagar y difundir contenido especializado a través de redes sociales**. La mayoría de los difusores aún enfocan su modelo de negocios en *banners*, usando la difusión en redes sociales de contenidos altamente atractivos para traer finalmente visitas a sus portales web. Kikvi es un difusor de contenidos audiovisuales, pero no enfoca su esfuerzo en *banners*. En Kikvi se explora la segunda posibilidad comentada de publicidad en internet, la publicidad sutil. A continuación se revisará la historia y funcionamiento en detalle de Kikvi.

* 1. Kikvi

Kikvi nace como un proyecto de un grupo de estudiantes en la Feria de Creación de Software de la Universidad Técnica Federico Santa María en el año 2012. En aquel entonces, bajo otro nombre (privado). La idea del producto consistía en una red de usuarios doble, consistente en 2 grupos principales: los **publicadores**, y los **creadores**.

La idea era la siguiente: el grupo de **creadores** consistía en personas generadoras de contenido audiovisual, que habían invertido tiempo y dinero en esto, y necesitaban rentabilizarlo. Su desafío se presentaba al momento de difundir dicho contenido, tomando como vía principal las redes sociales, pero sin contar con el “*peso”*, suficiente para llegar a una masa suficientemente atractiva de público objetivo. Es aquí donde entraría en participación el segundo grupo usuario de la plataforma, los **publicadores**. Este conjunto consistía en una serie de personas adeptas al uso de redes sociales y con alto índice de influencia en sus círculos. La función de los publicadores consistía en compartir los contenidos generados por los creadores en sus respectivas redes sociales, siendo dinero la motivación. El modelo de negocios sería el siguiente: El grupo de **creadores** compraría una cierta cantidad de vistas de “calidad”, ya que se seleccionaba a los usuarios **publicadores** a través de un proceso de filtrado, pagando una cantidad de dinero fijo por vista. Esta cantidad se dividiría entre los **publicadores** que participaran de la campaña (consiguiendo parte de ese dinero por cada vista que consiguieran) y la empresa.

El concepto tuvo un éxito relativo en su fase inicial, consiguiendo financiamiento a través de la incubadora **3IE** de la UTFSM.

Poco tiempo después de este hito, hubo diferencias de opiniones entre los fundadores del proyecto, lo que resultó en la separación de los mismos. Parte de los involucrados siguieron con el proyecto inicial y el resto emprendió nuevos caminos, agregando nuevos integrantes al equipo y formando **Playgue**, plataforma que tenía la misma idea mencionada anteriormente. Luego de poco tiempo de funcionamiento, se hicieron claras las falencias del modelo de negocios, respaldándose además en el bajo éxito del proyecto seguido por el otro grupo de socios originales:

* La idea de “ganar dinero” en internet por poco esfuerzo fue una idea que se explotó mucho en el pasado, lo que hoy genera una fuerte desconfianza por parte del usuario.
* Los clientes (creadores) eran muy reacios sobre el origen de las vistas y su legitimidad. Esto se veía potenciado por servicios de países asiáticos que ofrecían una gran cantidad de vistas a muy bajo costo.
* El grupo de publicadores no tardó en encontrar la forma de optimizar su tiempo y sistema, generando grupos de “ayuda” en Facebook donde, entre ellos mismos, cada uno veía repetidas veces los videos publicados por el resto, generando ganancias para todos (menos para el objetivo real del negocio y los creadores).

En a estas alturas que se tomó la decisión de modificar la manera en la que se estaba abordando el negocio. El concepto de “dinero” en internet produce rechazo, por lo que se adoptó una metodología de puntos. Además, se decide cambiar la dinámica del contenido del sitio, complementando el contenido de marcas y clientes con videos altamente atractivos, pero cuya recompensa de puntos era considerablemente menor a la obtenida por los videos “auspiciados”. Es junto con estos cambios que se hace un fuerte trabajo de diseño de interfaces y la plataforma toma su nombre actual: **Kikvi**.

Los puntos obtenidos a través del portal podían utilizarse para canjear sobre un catálogo de productos, partiendo de cosas simples como entradas dobles al cine, y llegando hasta productos de alto valor como consolas de videojuegos y cámaras para deportes extremos. Este acercamiento provocó gran revuelo y consiguió la participación de muchos usuarios. El desafío consistía en mantener un catálogo de productos constante sin gastar más dinero del que ingresaba en la empresa.

Fue al poco tiempo después que se decidió incluir concursos en la plataforma, lo que en teoría solucionaría 2 aristas con las que se estaba teniendo problemas:

* Lo canjes solían ser por una gran cantidad de puntos, lo que desmotivaba fuertemente a los usuarios.
* Los canjes significaban una gran inversión de dinero (para ser atractivos).

La inclusión de concursos al sistema significó un seguimiento mucho más cercano de los usuarios a la plataforma. Interactuando de forma activa por períodos de tiempo (o al menos esto se creía). Al poco tiempo los concursos habían tomado gran fuerza en la plataforma, desplazando a los canjes inmediatos.

Surge a estas alturas la necesidad de entender de mejor manera el negocio, los usuarios y los procesos de la plataforma. Hasta el momento se estaba avanzando a ciegas: funcionando en base a prueba y error. El hecho de tener información de los procesos y funcionamiento se vuelve una herramienta atractiva y poderosa y se toma la decisión de explotarla.

* 1. El problema

Kikvi funcionó durante largo tiempo a ciegas, sin mucho conocimiento de un mercado muy poco explotado y sin respaldos ni casos de éxito cercanos para seguir. Se hace necesario entender el negocio, entender los procesos y entender a los usuarios para poder mejorar la experiencia y percepción general sobre el producto.

Con el avance del tiempo se hace insostenible mantener una metodología de prueba y error, y se hace necesario tomar los pasos precisos en la dirección correcta. Además, es importante poder definir qué es lo que se considerará un caso de éxito en la plataforma, tanto en relaciona a un usuario como en relación a un video en particular, para de esta forma poder potenciar y emular este tipo de comportamientos.

Kikvi comienza desde los cimientos sin financiamiento, lo que limita las posibilidades de contar con personal especializado para áreas como marketing, o análisis. Esto lleva a que la plataforma funcione de acuerdo a estipulaciones e hipótesis, sin tener claro si el camino emprendido, o la forma de abordar el problema que se pretende resolver con Kikvi, mejorar la difusión de campañas audiovisuales a través de redes sociales, es la correcta.

Hoy se está a puertas de un nuevo lanzamiento, una nueva iteración en la historia del producto, y se espera tomar las decisiones correctas apoyándose en lo que se espera extraer de los datos que se han recolectado hasta la fecha. Este nuevo lanzamiento será complementado con el desarrollo de aplicaciones móviles y una nueva imagen de producto (rediseño de logos, paletas de colores y plataforma web), estos puntos serán abordados más a fondo más adelante en este estudio.

* 1. Objetivos
     1. Objetivo principal

Mejorar la percepción y experiencia usuaria para incrementar el éxito y penetración de campañas de clientes en un difusor de contenidos audiovisuales (Kikvi).

* + 1. Objetivos específicos

Para poder lograr el objetivo principal propuesto, es necesario en primera instancia realizar una serie de pasos relacionados con los datos:

* Hacer distinción de casos de éxito dentro de la plataforma, para así poder analizarlos y replicarlos, aumentando por un lado la satisfacción real del cliente, y por el otro la percepción del usuario.
* Descubrir qué indicadores son de interés para videos, usuarios y la plataforma en general, con el fin de medir su variación al tomar acciones.

* Mejorar la percepción de usuarios enfocando esfuerzos a áreas de interés, al desarrollar un entendimiento de qué es lo que más usan y menos usan de la plataforma.

* Mejorar el porcentaje de rebote de usuarios en la plataforma. Esto quiere decir que se pretende que los usuarios (visitas) no vengan con un objetivo específico a la plataforma y se vayan, si no que se distraigan e interactúen y exploren Kikvi.

1. Estado del arte
   1. Minería de datos

De forma general, la minería de datos consiste en el proceso de analizar datos[[4]](#footnote-4) de diferentes fuentes, desde diferentes perspectivas, con el fin de resumirla en información[[5]](#footnote-5) útil, o sea, información que pueda ser utilizada para aumentar ganancias, disminuir costos, mejorar procesos, etc. Entonces, un software de minería de datos es una herramienta analítica para datos.

Las herramientas de minería de datos permiten a sus usuarios analizar datos recopilados desde muchas dimensiones o ángulos diferentes, resumiéndolo todo en una serie de relaciones identificadas entre las variables estudiadas. Por lo general, la minería de datos se utiliza para encontrar correlaciones o patrones entre docenas de variables, o para encajar en el contexto, campos, de una gran base de datos relacional. A pesar de que la minería de datos es un término relativamente nuevo, la tecnología no lo es. Las compañías han utilizado computadores de alto rendimiento para iterar sobre grandes volúmenes de datos con el fin de generar reportes de interés para análisis durante años. Sobre estos escenarios, la innovación continua sobre herramientas computacionales como procesadores, discos de almacenamiento y software estadísticos, ha logrado incrementar dramáticamente la precisión de los análisis, mientras disminuyen los costos de realizarlos.

De una forma muy simplificada, la minería de datos consiste en la identificación de patrones en sets, generalmente de grandes dimensiones, de datos, con el fin de adquirir algún conocimiento[[6]](#footnote-6).

Los avances en métodos de captura de datos, procesamiento, transmisión de datos y almacenamiento, permiten hoy a las organizaciones integrar sus bases de datos en lo que se conoce como *data warehouses*. *Data warehousing* se define como el proceso de administrar y recuperar datos centralizados. Representa la idea de mantener un repositorio central con todos los datos de una entidad. Esta práctica es necesaria para maximizar el acceso y posibilidades de análisis de los usuarios.

La minería de datos se usa hoy vastamente por todo tipo de organizaciones. Permite determinar relaciones entre tanto variables internas como externas de las compañías. Además, permite determinar factores de retroalimentación como por ejemplo el impacto de una campaña en ventas, satisfacción de los clientes, etc. Con minería de datos, un vendedor podría refinar el mercado objetivo de un producto para enfocar sus esfuerzos de campaña en esa dirección y lograr alta respuesta de clientes.

La minería está compuesta de 5 elementos o etapas principales:

* Extraer, transformar, y cargar datos en el data warehouse.
* Almacenar y administrar los datos en un sistema de bases de datos multidimensional.
* Dar acceso a los datos a analistas del negocio y profesionales de TI.
* Analizar los datos con aplicaciones especializadas.
* Presentar la información en formatos útiles, como gráficos o tablas.

En relación a los niveles de análisis del paso número 4. , hay una serie de algoritmos y/o métodos utilizados comúnmente, como:

* Redes neuronales: Modelos predictivos no lineales que se aprenden a través de entrenamiento.
* Algoritmos genéticos: Técnicas evolutivas que usan procesos como combinaciones genéticas, mutaciones, y selección natural en un diseño basado en los conceptos de evolución natural.
* Árboles de decisión: Estructuras en forma de árboles que representan una línea de pensamiento a través de un set de decisiones. Estas decisiones generan reglas de clasificación para un set de datos.
* Vecino más cercano: Técnica que clasifica a cada registro en base a una combinación de sus k vecinos más cercanos. Llamada también k-*nearest* *neightbours*.
* Reglas de inducción: Extracción de reglas *if-then* útiles, basadas en significancia estadísticas.
* Visualización de datos: Interpretación visual de relaciones complejas en datos multidimensionales. Se usan herramientas gráficas para ilustrar relaciones en los datos.

En la sección 2.1.2.2.1: Clasificación, se ahondará más en estos métodos.

* + 1. Procesos de minería de datos
       1. Proceso de descubrimiento del conocimiento KDD[[7]](#footnote-7)

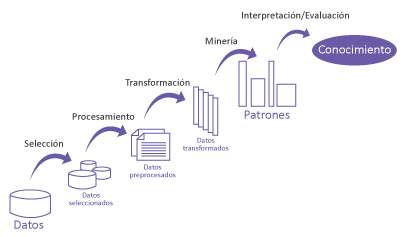
Recibe este nombre el proceso que tiene por entrada la base de datos y sus versiones modificadas, y tiene como salida el subconjunto de patrones que se transformarán en conocimiento, luego de la aplicación de minería de datos. (Referencia 1 - Curso Bussiness Inteligence -- UTFSM)

De acuerdo con *Azevedo y Santos* (Referencia 4 - KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW - Azevedo & Santos), *KDD* es el proceso de usar métodos de minería de datos para extraer lo que es considerado conocimiento de acuerdo a una seria de medidas y umbrales, usando bases de datos en conjunto con cualquier pre-procesamiento necesario, extracto de muestras, o transformación. El proceso cuenta con 5 fases fundamentales: Selección, Pre-proceso, Transformación, Minería de datos, Interpretación/Evaluación.

El *KDD* es precedido por el desarrollo de un entendimiento del área de aplicación, cualquier conocimiento previo relevante y los objetivos del usuario final. Es un proceso iterativo e interactivo, e involucra numerosos pasos con muchas decisiones tomadas en el camino por el usuario.

De forma gráfica, en la Ilustración 1: Proceso del descubrimiento del conocimiento (KDD) se aprecian los 5 pasos del KDD:

Ilustración 1: Proceso del descubrimiento del conocimiento (KDD)



A continuación se revisará más a fondo cada una de las etapas relacionadas con el proceso de descubrimiento del conocimiento.

* + - * 1. Selección

Esta etapa consiste en crear un set de datos, o enfocar los esfuerzos en un sub conjunto de variables de los mismos. Durante esta etapa, es fundamental contar con un conocimiento previo del negocio, que ayudará a definir cuáles variables son relevantes para el estudio y cuales no lo son. Por ejemplo, si se desea descubrir qué clientes son más susceptibles a un esfuerzo de marketing, case con certeza el nombre del cliente no será una variable importante para el estudio, pero sí el segmento económico o el nivel de ingresos del mismo.

* + - * 1. Pre-proceso

Durante esta etapa se busca *limpiar* los datos. Este quiere decir que se tomará una serie de acciones para que los datos no cuenten con inconsistencias u observaciones faltantes/inválidas. Durante esta etapa se realiza una limpieza de los datos:

* Faltantes: En torno a esta situación se pueden tomar una serie de acciones, como ignorar datos con observaciones faltantes, llenarlos manualmente, usar una variable global para llenarlos (como N/A, -inf, etc), poner la media del atributo con respecto a todos los datos, usar la media del atributo considerando sólo los datos de la misma clase, o usar el valor más probable del dato.
* Datos *ruidosos*: Un dato ruidoso es una observación que tiene un error aleatorio en una variable medida.
* Datos inconsistentes: Los datos inconsistentes se generas principalmente por variaciones al momento de ingresarlos, como el uso de diferentes capitalizaciones o faltas de ortografía. Una inconsistencia puede ser, por ejemplo, si en una observación de persona, su ciudad de residencia es “Santiago”, mientras que en otra es “stgo”, mientras que en otra es “Sanitago”. Se entiende que todas las observaciones hacen referencia a la misma ciudad pero, por errores o decisiones humanas, tienen un valor diferente.
  + - * 1. Transformación

En esta etapa se realizan todas las transformaciones necesarias a los datos para que puedan ser interpretados de mejor manera por los algoritmos de minería de datos. Dependiendo de los algoritmos a aplicar, se hace necesario aplicar uno o más tipos de transformación. Algunos tipos de transformación son:

* Normalización: Este método consiste en mapear los valores de las observaciones en un intervalo definido. Por ejemplo, normalizar los datos para que sus valores estén dentro del rango [0,1]. Este método es de particular importancia cuando se planea utilizar técnicas de *clustering* basadas en distancia, ya que, al no aplicarse, se desbalancea la importancia de diferentes variables por culpa de las unidades de medidas usadas. Por ejemplo, de distorsionará la distancia, dándole más importancia a una variable de mayor magnitud, como podría ser el ingreso per cápita de una base de datos de clientes (orden de los cientos de miles y millones) contra la edad.
* Agregación: Este método es utilizado cuando se desea agrupar variables. Por ejemplo, pasar una serie de registros de ingreso mensual a una cantidad más reducida de registros de ingreso anual.
* Generalización: Consiste en, como dice su nombre, generalizar variables. Se trata de reemplazar datos de variables de bajo nivel por un dato de niveles más altos. Por ejemplo, reemplazar datos de ciudades por regiones (Santiago a Región Metropolitana, Temuco a IX región, etc).

* + - * 1. Minería de datos

Esta etapa consiste en la búsqueda de patrones de interés en alguna forma particular de representación, dependiendo del objetivo final de la minería. Fue tratada en el punto 2.1: Minería de datos.

* + - * 1. Interpretación/Evaluación

En esta etapa final, se interpretan y evalúan los patrones encontrados, con el fin de juzgar su utilidad para el objetivo final o negocio, además de su asertividad.

* + - 1. SEMMA (Sample, Explore, Modify, Model and Assess)

*SEMMA* es una serie de pasos secuenciales que guía a la implementación de aplicaciones de minería de datos. Su nombre es acrónimo para *Sample*, *Explore*, *Modify*, *Model* *& Assess*, lo que hace referencia a cada una de las fases del proceso:

1. *Sample* (Muestra): La primera fase del proceso, consiste en la selección de un set de datos para modelar. El desafío recae en que esta muestra debe ser lo suficientemente grande para que sea representativa, y lo suficientemente pequeña como para ser manejada de forma eficiente.
2. *Explore* (Explorar): Esta fase consiste en la visualización de los datos, con el fin de entenderlos al descubrir relaciones anticipadas como no anticipadas entre las variables en ellos, además de la detección de anomalías.
3. *Modify* (Modificar): En esta etapa del proceso se realiza cualquier acción para seleccionar, crear y/o transformar datos con el fin de prepararlos para el modelo.
4. *Model* (Modelar): El objetivo de esta fase es aplicar varias técnicas de modelo sobre las variables preparadas con el fin de crear modelos que puedan posiblemente generar los resultados esperados.
5. *Assess* (Evaluar): Última etapa de *SEMMA*, consiste en la evaluación de los modelos desarrollados, con el objetivo de juzgar si son suficientemente confiables y útiles.

Una crítica que se hace comúnmente a este proceso, es que se enfoca únicamente al modelo en proyectos de minería de datos, dejando los aspectos propios del negocio afuera, a diferencia de otros procesos como *CRIPS-DM[[8]](#footnote-8)*, que cuentan con fases[[9]](#footnote-9) enfocadas e estos aspectos.

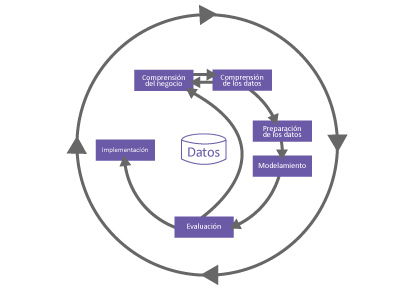
* + - 1. CRISP-DM (Cross-Industry Standard Process for Data Mining)

*CRISP-DM* recibe su nombre del acrónimo en el título (en español, Proceso estándar multi-industria para minería de datos), y consiste en un ciclo compuesto de 6 etapas:

1. **Entendimiento del negocio**: En la primera etapa de *CRISP-DM*, se busca comprender los objetivos y requerimientos del proyecto desde el enfoque del negocio, para luego transformarlo en un problema de minería de datos y un plan preliminar para alcanzar los objetivos.
2. **Entendimiento de los datos**: Comienza con un set de datos inicial, y consiste en actividades con la finalidad de familiarizarse con los datos, para identificar problemas de calidad de los datos, para descubrir una primera mirada sobre los datos o bien descubrir sub conjuntos interesantes para formular una hipótesis para información escondida.
3. **Preparación de los datos**: Esta fase comprende todas las actividades necesarias para generar el set de datos final a partir de los datos en bruto.
4. **Modelo**: Aplicación de varias técnicas de modelo, calibrando sus parámetros a valores óptimos.
5. **Evaluación**: En esta etapa los modelos obtenidos son juzgados y los pasos para construirlos son evaluados con el fin de concluir con seguridad que efectivamente cumple con los objetivos del negocio.
6. **Despliegue**: El término del modelo por lo general no significa el fin del proyecto. El conocimiento obtenido luego debe ser organizado y desplegado de forma que el cliente final pueda utilizarlo.

De forma gráfica, se aprecia en la Ilustración 2 el ciclo de vida de CRISP-DM.

Ilustración 2 Ciclo de vida de CRISP-DM



* + 1. Tareas de minería de datos
       1. Tareas descriptivas

En este tipo de tareas el objetivo es, como dice su nombre, describir los datos existentes. Busca de proporcionar información entre las relaciones existentes entre los datos y sus características. En el contexto, teóricamente se podría llegar a una afirmación como por ejemplo: El que un estudiante tenga actividades extra programáticas en el primer semestre, implica que también tendrá en el segundo.

Para los objetivos de este estudio, este tipo de tareas es de especial interés ya que ayudará a establecer patrones y características que definan a un estudiante de excelencia.

* + - * 1. Visualización

La tarea de visualización consiste en revisar los datos de forma mecánica, para revisar cualquier relación entre variables que se pueda apreciar en primeras instancias. Para facilitar esta tarea hay una gran cantidad de software en buenos estados de desarrollo, de donde destaca *Tableau[[10]](#footnote-10)*.

* + - * 1. Correlaciones y factorizaciones

Esta tarea consiste en desplegar los datos y evaluar si se encuentra alguna correlación entre las variables pertenecientes al estudio. La correlación puede ser lineal, o pueden estar relacionadas de otra manera. Esta tarea solo puede ser realizada, por su naturaleza, sobre variables numéricas.

Asociación

La asociación es una tarea descriptiva, no supervisada[[11]](#footnote-11), que hace referencia a reglas que son capaces de describir los datos en base a ocurrencias en las variables. En otras palabras, describe el comportamiento de una variable en base al de otra (u otras). Por ejemplo, una regla de asociación sería "Si el año de ingreso de un estudiante es igual a 2008, ha tomado actividades extra programáticas y estudia arquitectura, entonces su colegio es subvencionado". Las reglas de asociación solo pueden aplicarse sobre variables nominales (todas las involucradas). La asociación puede presentarse de dos maneras:

1. Reglas de asociación: Son asociaciones recíprocas, o sea, que hay una implicancia doble, describiendo cada una de las variables relacionadas a la asociación a la otra.
2. Dependencias: A diferencia del caso anterior, este tipo de asociaciones son direccionales, o sea que el cumplimiento de una serie de condiciones implican que se cumplirán otras, y no al revés.

Además, la asociación se puede clasificar de acuerdo a los tipos de valores que maneja la regla, dimensiones de los datos involucrados, instantáneas o secuenciales y positivas o negativas. En la Tabla 1: Clasificación de reglas de asociación**,** se ilustran las diferentes clasificaciones de acuerdo a los criterios mencionados.

Tabla 1: Clasificación de reglas de asociación

|  |  |  |
| --- | --- | --- |
| Clasificación | Primer caso | Segundo caso |
| Tipos de valores que maneja la regla: booleana o cuantitativa | Asociación indica ausencia o presencia de elementos. | Asociación describe relaciones entre atributos cuantitativos. |
| Dimensiones de los datos involucrados: unidimensional o multidimensional | Referencia a una única dimensión. | Referencia a dos o más dimensiones. |
| Instantáneas o secuenciales | Indica relaciones inmediatas, relacionadas con el tiempo, contemporáneas | Establece un orden temporal en la relación |
| Positivas o negativas | Indica la ocurrencia o presencia de los elementos relacionados | Indica la ausencia de al menos uno de los elementos relacionados |

Los algoritmos de búsqueda de asociaciones tienen la particularidad de que la mayoría se puede descomponer en dos fases. La primera consta de la búsqueda de set de ítems frecuentes con un soporte[[12]](#footnote-12) mayor o igual al deseado, o sea, que se buscan conjuntos de elementos que cuenten con cierto criterio establecido, sin separarlos aún. Luego, en la segunda fase, se hacen particiones de los set de ítems, calculando la confianza[[13]](#footnote-13) de cada una, y reteniendo las reglas que tengan confianza mayor o igual a la deseada.

Segmentación (Agrupamiento)

Esta tarea consiste en agrupar los datos en diferentes sub conjuntos, o clases, de acuerdo a la relación entre ellos. Se busca que todos los elementos presentes en un grupo definido tengan propiedades parecidas, o sea, que sean similares entre sí y que sean diferentes a los elementos de otros grupos. La segmentación es una técnica de aprendizaje no supervisado, que utiliza el término de *distancia* para describir a los elementos. Dos elementos tendrán poca distancia entre ellos si son parecidos (similares). Análogamente, tendrán harta distancia entre ellos si no son similares. En base a estos términos, la segmentación busca minimizar la distancia de los elementos pertenecientes a un mismo grupo y maximizar la distancia entre los grupos.

La definición de distancia puede variar de acuerdo a la fórmula que se utilice para calcularla:

* Distancia de Minkowksi:

* Distancia de Manhattan (r=1)

* Distancia Euclideana (r=2)

* Distancia de Chebyshev (

Los algoritmos de segmentación pueden ser clasificados por tipo. A continuación se listan algunos de ellos:

* Particionamiento: Estos métodos construyen *k* particiones de un set de datos, representando cada una de estas un grupo. *K* debe ser menor o igual al número de elementos del set de datos mencionado. Ejemplos:
  + K-Means
  + K-Medoids
* Jerárquicos: Como su nombre lo dice, genera una descomposición jerárquica del conjunto de grupos, en otras palabras, crea una serie de sub conjuntos de datos, en los que algunos engloban a otros. Ejemplos:
  + BIRCH
  + CURE
* Basados en densidad: Se va creciendo el grupo (agregando elementos) hasta que la densidad (número de objetos) del mismo alcance algún umbral definido. Ejemplos:
  + DBSCAN
  + OPTICS
* Basados en grilla: Separa el espacio de datos en una grilla (finita), para luego realizar operaciones de agrupamiento sobre ella. Ejemplos:
  + STING
  + CLIQUE
* Basados en modelo: Utilización de modelos hipotéticos para cada uno de los grupos, ajustando los datos a los modelos. Ejemplos:
  + COBWEB, CLASSIT (estadísticos)
  + SOM, mapas auto organizados (redes neuronales)
    - * 1. Detección de anomalias

La detección de valores e instancias anómalas es una tarea necesaria al momento de realizar minería de datos. En todo set de datos se presentarán registros que se escapen de todo patrón o tendencia, y es importante poder reconocerlos para no considerarlos como un patrón común, si no como comportamientos anómalos como fraudes, fallas u *outliers*. Informalmente, un *outlier* es cualquier valor de dato que pareciera estar fuera de lugar con respecto al resto de los datos. De acuerdo a *Douglas M. Hawkins* (Referencia 2 - Identification of Outliers -- Douglas M. Hawkins, 1980):

"The intuitive definition of an outlier would be 'an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism'"*[[14]](#footnote-14)*

* + - 1. Tareas predictivas

Las tareas predictivas, como dice su nombre, son problemas en los que se hace necesario predecir un (o varios) valores para un set de datos. La salida de una tarea predictiva es una categoría (a la que pertenece uno o más datos) o un valor numérico relacionado con el o los datos en cuestión.

Las tareas predictivas de pueden clasificar en:

* Tareas de clasificación.
* Regresión estadística.
* Pronósticos.
  + - * 1. Clasificación

Las tareas predictivas de clasificación buscan definir un modelo que, como dice su nombre, sea capaz de predecir la clase de un objeto que no la tiene definida. A pesar de que sus objetivos son predecir la clase de un elemento en particular, pueden ser utilizadas para predecir un valor que se encuentre perdido o que no se tenga a priori. Estas tareas son supervisadas, por lo que se debe contar con un conjunto de datos de entrenamiento, que ya se encuentran clasificados. El proceso de generación de un modelo predictivo consta de 3 pasos:

1. División de los datos en dos conjuntos: entrenamiento y prueba.
2. Utilización del sub conjunto de entrenamiento para la construcción del modelo.
3. Utilización del sub conjunto de prueba para validar el modelo conseguido en el punto anterior, si el porcentaje de casos exitosos es aceptable, se valida el modelo (útil para clasificar otros casos).

A continuación se revisarán algunos algoritmos de clasificación predictiva.

Árboles de decisión (o clasificación)

Los árboles de decisión son un método de clasificación en el cual se somete un dato a una seria de condiciones, que lo can clasificando de acuerdo a los valores de las variables relacionadas con el mismo. Por ejemplo, se somete primero a un dato a la evaluación de una variable: "si el alumno tiene un promedio mayor a 55, entonces se cuestiona la variable *año de ingreso*, si no, se cuestiona la variable *plan de carrera"*, con el fin de predecir alguna variable en particular. Cabe destacar que no se trata de árboles binarios, si no que se pueden considerar numerosos intervalos o valores para cada variable para generar la clasificación.

Su estructura es similar a un diagrama de flujo, donde cada vértice simboliza una condición a la que se somete el dato a predecir. El último nivel del árbol, los nodos hoja, representan las clases. Su construcción suele llevarse a cabo con estrategias *divide & conquer[[15]](#footnote-15)*, empezando con todos los elementos del grupo de entrenamiento en la raíz, y continuando dividiéndolos en el atributo que se elija para ramificarlo.

Inducción de reglas (reglas de clasificación)

Los métodos de inducción de reglas tienen las mismas propiedades que los métodos de árboles de decisión, describiendo una serie de condiciones *if-then* para llegar a la clasificación deseada. La obtención de dichas condiciones, o reglas, puede ser a partir de un árbol de decisión, a través de algoritmos específicos como *STAR* o *Ripper*, o a partir de reglas de asociación. Además, es posible extraer reglas de clasificación desde una red neuronal, a través del algoritmo *MofN*, propuesto por *Towell* (1990) (Referencia 5 - G. Towell, J. Shavlik, and M. Noordewier, Refinement of approximate domain theories by knowledge-based neural networks", 1990). En particular, permite la extracción de reglas desde una red neuronal multicapa, a través de los siguientes pasos de agrupamiento, extracción de reglas, agrupación de reglas y poda de reglas.

Métodos Bayesianos

Los métodos bayesianos son herramientas estadísticas capaces de predecir las probabilidades de que un elemento en cuestión pertenezca a una clase en particular. Pueden ser del tipo *naive* (ingenuos), que asumen que el valor de cada una de las propiedades es independiente de los valores de las otras (en un mismo elemento), llamada *independencia condicional de clases*. Los métodos Bayesianos pueden ser utilizados tanto para fines descriptivos como predictivos. En el primer caso, se usan para descubrir relaciones de independencia y/o relevancia para poder realizar un estudio más profundo a través de inferencias estadísticas. En el segundo caso, se utilizan como clasificadores. Estos métodos se basan en el teorema de Bayes[[16]](#footnote-16)

Métodos basados en casos y vecindad

Estos métodos se caracterizan por utilizar el conjunto de entrenamiento para clasificar nuevos datos. En esta categoría hay presentes técnicas para segmentación, como *K-Means*, y para clasificación, como *LVQ*. Además, se utilizan métodos de ensamblaje, que combinan varios modelos con el objetivo de conseguir una mejor precisión final en el clasificador

* + - * 1. Regresión estadística

A través de esta tarea, se busca generar una función matemática que sea capaz de estimar el valor de alguna variable de interés a partir del resto de las variables relacionadas con un dato en particular. La regresión estadística puede ser utilizada únicamente para valores numéricos, y la función se puede calcular a través de interpolación, estimación, o logística.

* + 1. Herramientas OLAP

Aunque en estricto rigor, las herramientas OLAP no pertenecen a la minería de datos, si pertenecen a la inteligencia de negocios. OLAP viene del inglés “OnLine Analytical Processing”, que significa “Procesamiento analítico en línea”. OLAP es una funcionalidad de software de inteligencia de negocios que tiene la particularidad de permitir a sus usuarios abordar un set de datos desde distintos puntos de vista con rapidez y facilidad. Las herramientas OLAP son de alto nivel y permiten a usuarios visualizar datos de forma que hagan sentido para su negocio, pudiendo rescatar conocimiento de los mismos.

Hay una gran cantidad de software a disposición que entra dentro de esta categoría, entre los que destaca Tableau, que será utilizado en este estudio debido a que se trata de una herramienta potente que ha sido utilizada con anterioridad por quien realiza el estudio.

* + 1. Herramientas de minería de datos

Hay una amplia gama de herramientas de minería de datos a disposición. Cada herramienta cuenta con implementaciones diferentes de una porción de los algoritmos más utilizados en los procesos de minería. Además, muchas de las herramientas cuentan con interfaces usuarias para facilitar el proceso de minería para quienes no tiene un conocimiento base de líneas de comando o programación. Algunas de las herramientas más utilizadas se listan a continuación.

* *Rapid miner*: Escrita en Java, esta herramienta de minería de datos funciona en torno a interfaces gráficas avanzadas, por lo que el usuario final requiere escribir muy poco código. Cabe destacar que esta herramienta se ofrece como servicio, más que como software local. Además, proporciona funcionalidades de pre-procesamiento y visualización, análisis predictivo, esquemas de aprendizaje y algoritmos de scripts de R. Esta potente herramienta es de código abierto, bajo licencia AGPL[[17]](#footnote-17).
* *Angoss*: Enfocada principalmente para organizaciones involucradas con ventas, marketing y análisis de riesgo, esta herramienta cuenta con una interfaz gráficas avanzada además de un asistente amplio para sus procedimientos. Si bien las interfaces y asistente podrían ser restrictivos para usuarios avanzados, *Angoss* implementa soporte total de línea de comando en R, satisfaciendo así a los usuarios que prefieren personalización por sobre facilidad de uso. Un plus de esta herramienta es que cuenta con una amplia gama de representaciones gráficas de datos. Si bien cuenta con una gama de implementaciones de los algoritmos más conocidos, no cuenta con suficientes herramientas para personalizar los procesos, por lo que no es la opción para quienes prefieran un ambiente fácilmente extensible.
* *KNIME (Konstanz Information Miner)*: Esta herramienta nace como una solución para farmacéuticas a nivel empresarial. Los desarrolladores crearon un producto escalable, modular y de código abierto, teniendo la flexibilidad necesaria para adaptarse rápidamente a las demandas de un campo de estudio en crecimiento como es la minería de datos. Siguiendo su éxito en la industria farmacéutica, otras industrias siguieron la tendencia y utilizan *KNIME* para sus procesos de *CRM[[18]](#footnote-18)* e inteligencia de negocios. Otra ventaja considerable que tiene esta herramienta, es que cuenta con una comunidad activa tanto de desarrolladores como de usuarios. No requiere conocimientos de programación para ser usada, ya que cuenta con interfaces intuitivas y de fácil uso.
* *R:* Más que una herramienta para minería de datos, R es un lenguaje de programación y ambiente para computación estadística y análisis. Es esta la razón que hace a R una potente herramienta de minería de datos. Bajo licencia GPL[[19]](#footnote-19), y de código abierto, R puede ser personalizado abiertamente, sin restricción, lo que se traduce en una cantidad inigualable de algoritmos e implementaciones desarrolladas por usuarios alrededor del mundo, lo que se traduce en una herramienta flexible, escalable y extremadamente personalizable. Por otro lado, a pesar de que hay algunas interfaces para tratar con este lenguaje, se requiere conocimientos de programación (y del lenguaje en sí) para sacar el máximo provecha de esta herramienta.

Para este estudio se decide usar R, en primera instancia por las ventajas mencionadas en el párrafo anterior, y en segunda instancia porque la herramienta ya es conocida por quien realiza el estudio.

1. Desarrollo
   1. Los datos

Antes de ahondar en los procesos y herramientas utilizadas, se describirá la forma y principales características de los datos del estudio. Los datos de la plataforma estudiada se encuentran en una base de datos detrás de un motor MySQL, cuyo modelo de datos se puede apreciar en el Anexo 1: Modelo de datos de Kikvi. Cabe destacar que se incluyeron sólo las tablas y variables más significativas en el diagrama.

Como es común en esta disciplina, no todos los datos son de utilidad para el estudio, por lo que se hace necesario aplicar la primera etapa del Proceso de Descubrimiento de Conocimiento (KDD): La selección. Se decide proceder con el método KDD por sobre SEMA y CRISP-DM por tratarse de procesos muy similares y ser el más metódico y completo. Si bien CRISP-DM integra todas las fases de KDD, no se usa el método debido a que un foco de este estudio es el conocimiento del negocio, variable que se usa como entrada en el caso de CRISP-DM.

* 1. Selección de los datos

Una primera mirada a los datos pone en evidencia que hay una gran cantidad de usuarios que no cuentan con su información actualizada. Así mismo, una parte de los videos tampoco se encuentra completa. Para mitigar esta falta de datos, se utilizan las herramientas (APIs) de Facebook y de YouTube para buscar información faltante de los usuarios y videos respectivamente. Se automatiza el proceso utilizando los scripts presentes en el Anexo 2: Scripts para APIs.

Una vez poblados la mayor cantidad de datos posibles, se procede a seleccionar los datos a usarse en el estudio. La selección inicial apunta a ser amplia y abarcar todos los datos que se cree que pudiesen, aunque sea en muy baja magnitud, influenciar el proceso de minería de datos. En el **Anexo 3: Selección de los datos (R)** se presenta el script en R utilizado para seleccionar los datos directamente desde la base de datos en MySQL.

* 1. Visualizando con Tableau (OLAP)

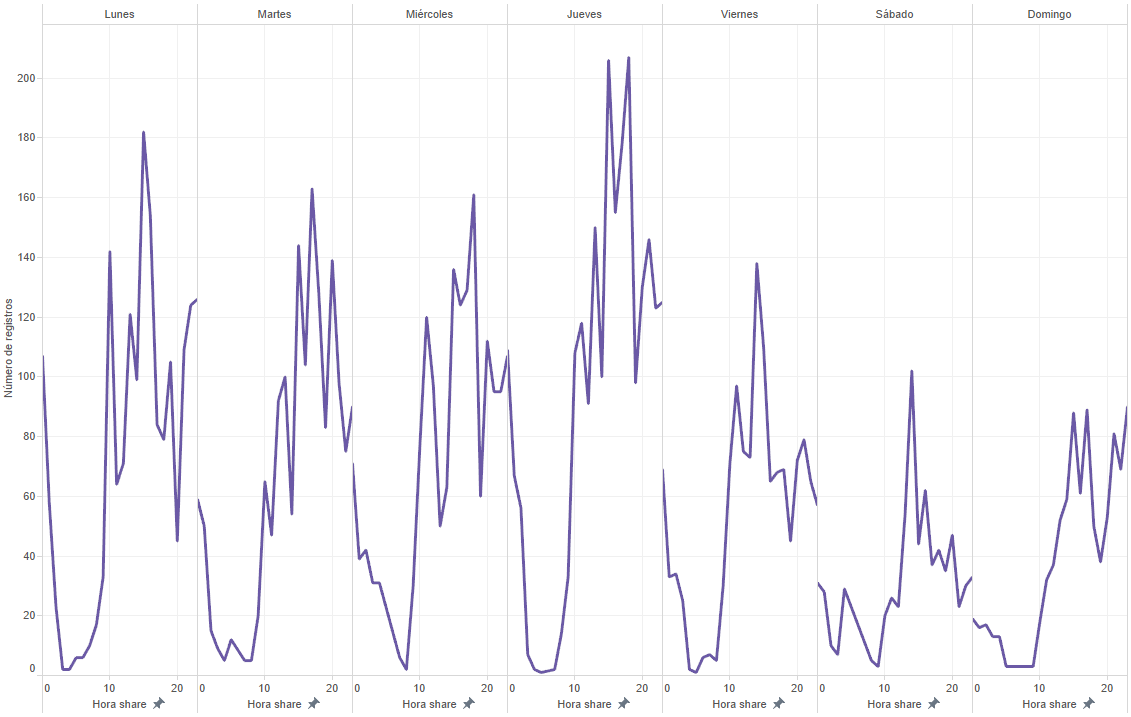
Como primer acercamiento a los datos, se decide utilizar *Tableau*, herramienta OLAP de *business intelligence* que ha sido utilizada con anterioridad por quien realiza el estudio y se sabe de su potencia y facilidad de uso.

El objetivo de esta fase es poder conocer de mejor manera y más a fondo los datos relacionados con el estudio, lo que es fundamental para reconocer *outliers*, identificar patrones y relaciones simples, o simplemente conocer de mejor manera los datos para tomar decisiones más asertivas al momento de procesarlos para ser usados en algoritmos de minería de datos.

Cabe destacar que fue necesario el pre-procesamiento de una cantidad considerable de datos para ser utilizados en esta herramienta, como por ejemplo, el paso de “fecha de nacimiento” de un usuario a “edad”, o el agrupar datos de acuerdo a su día de registro/actualización

A continuación se presentan algunos gráficos generados con esta herramienta, que a primera vista parecen ser importantes para conocer el negocio.

Ilustración 3: Actividad de videos compartidos en días de la semana

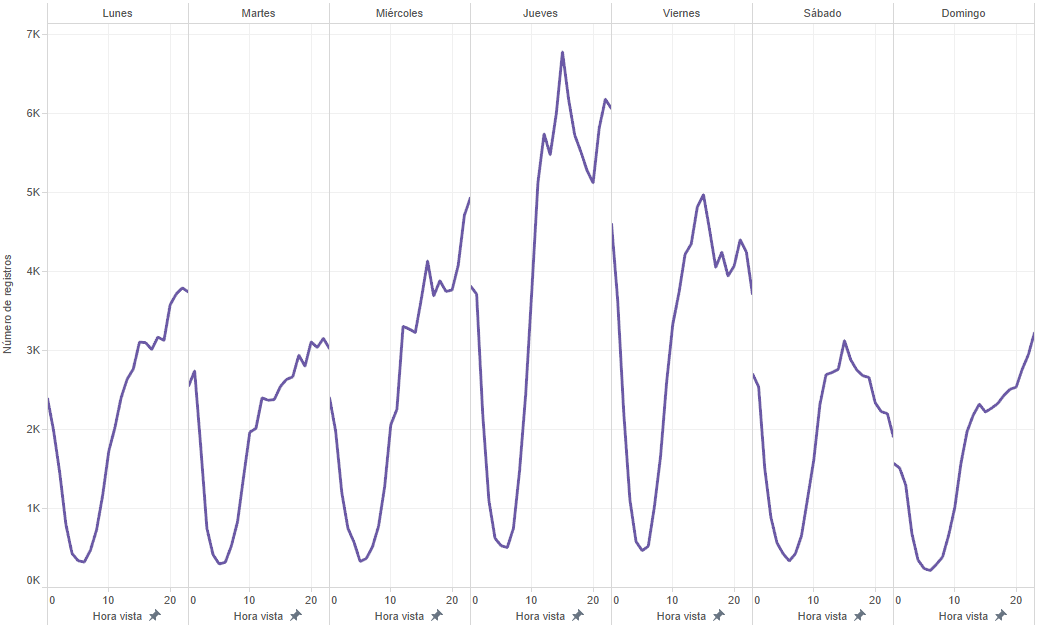


La Ilustración 3: Actividad de videos compartidos en días de la semana, representa en su eje horizontal 7 subdivisiones, que hacen referencia a cada día de la semana, y en cada subdivisión, se consideran las horas de ese día. En el eje vertical se encuentra el promedio los registros de videos compartidos encontrados ese día, en esa hora, por parte de todos los usuarios de la plataforma. Se puede desprender de este gráfico 2 cosas interesantes para el negocio:

* La mayor actividad relacionada con compartir videos (por parte de los usuarios de la plataforma) se registra a eso de las 3 o 4 de la tarde.
* El día de mayor actividad es el día jueves, mientras que el día de menor actividad es el día domingo.

El primer acercamiento fue considerar la siguiente hipótesis: la actividad del día jueves está siendo distorsionada por la presencia de outliers. En base a esto se buscó y detectó 2 fechas en particular que podrían afectar el resultado. Al remover estas fechas del estudio, la tendencia se mantuvo, por lo que se descartó la hipótesis.

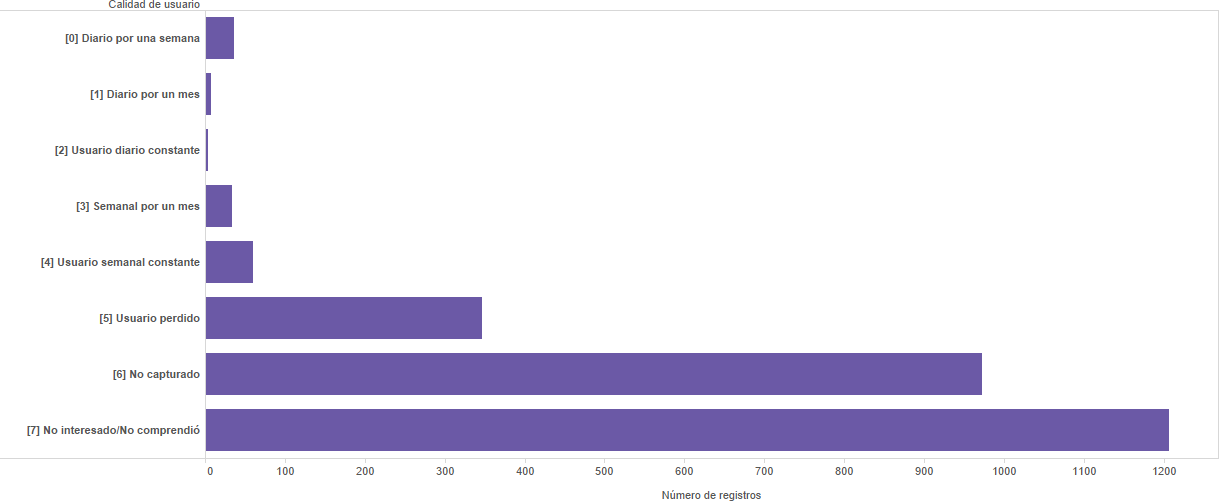
Ilustración 4: Actividad de vistas en días de la semana



En laIlustración 4: Actividad de vistas en días de la semana, se presentan la cantidad de vistas generadas en cada día de la semana, a cada hora. En el eje horizontal se presentan 7 subdivisiones, que hacen referencia a cada día de la semana, y en cada subdivisión, se consideran las horas de ese día. En el eje vertical se encuentra el promedio los registros de vistas para ese día y esa hora en particular.

Este gráfico se comporta de forma similar a la **Ilustración 3: Actividad de videos compartidos en días de la semana**, de lo que se deduce la cantidad de vistas generadas podría estar relacionada con la cantidad de videos compartidos.

Ilustración 5: Cantidad de usuarios por su calidad



En la Ilustración 5: Cantidad de usuarios por su calidad, se presenta la cantidad de usuarios pertenecientes a cada conjunto de “calidad de usuario”. La calidad de un usuario fue juzgada en base a qué tantos videos comparte, y qué tan seguido lo hace. Además, se consideró el período total de actividad en la plataforma (tomando como inicio la fecha de registro, y como término la última vez que compartió un video. A continuación se detalla cada uno de los niveles de calidad considerados:

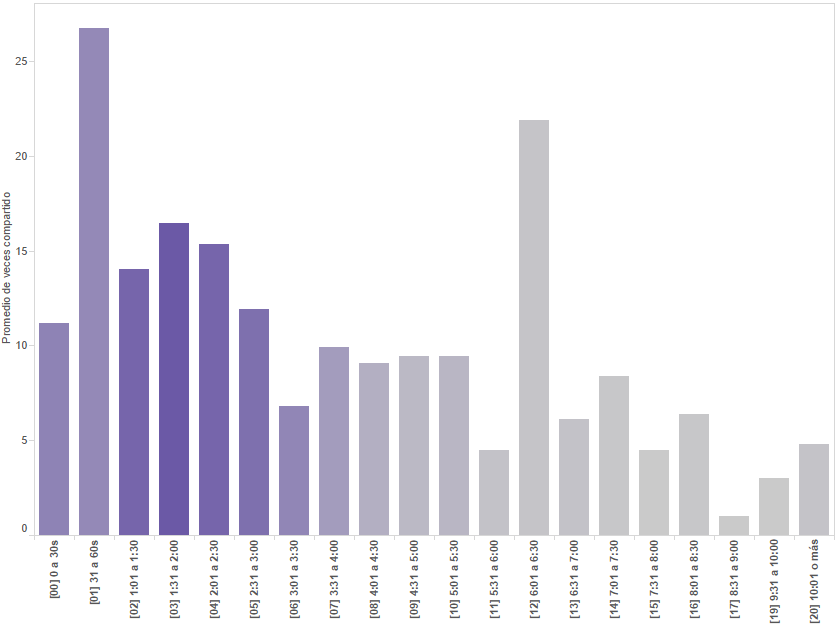
* [0] Diario por una semana: Considera a los usuarios que compartieron al menos un video por día, por un período total de entre 7 y 29 días.

* [1] Diario por un mes: Considera a los usuarios que compartieron al menos 1 video diario, por un período total de entre 1 y 2 meses.
* [2] Usuario diario constante: Considera a los usuarios que compartieron al menos 1 video diario, por un período total de más de 2 meses.

* [3] Semanal por un mes: Considera a los usuarios que compartieron videos al menos 1 vez por semana, por un período total de entre 1 y 2 meses.
* [4] Usuario semanal constante: Considera a los usuarios que compartieron videos al menos 1 vez por semana, por un período total de más de 2 meses.
* [5] Usuario perdido: Considera a los usuarios que compartieron videos diariamente, por menos de una semana, y a los que compartieron videos semanalmente, por menos de un mes.
* [6] No capturado: Considera a los usuarios que compartieron videos únicamente durante 1 día.
* [7] No interesado/No compendió: Considera a los usuarios que se registraron en la plataforma pero que nunca compartieron un video.

De este gráfico se desprenden temas preocupantes, ya que la gran masa de usuarios se registra y luego no interactúa con la plataforma, o interactúa sólo durante un día con la misma. Se considera entonces la accesibilidad de la plataforma. El cómo abordar este tema será discutido en la sección **Conclusiones**.

Ilustración 6: Veces compartidos VS duración

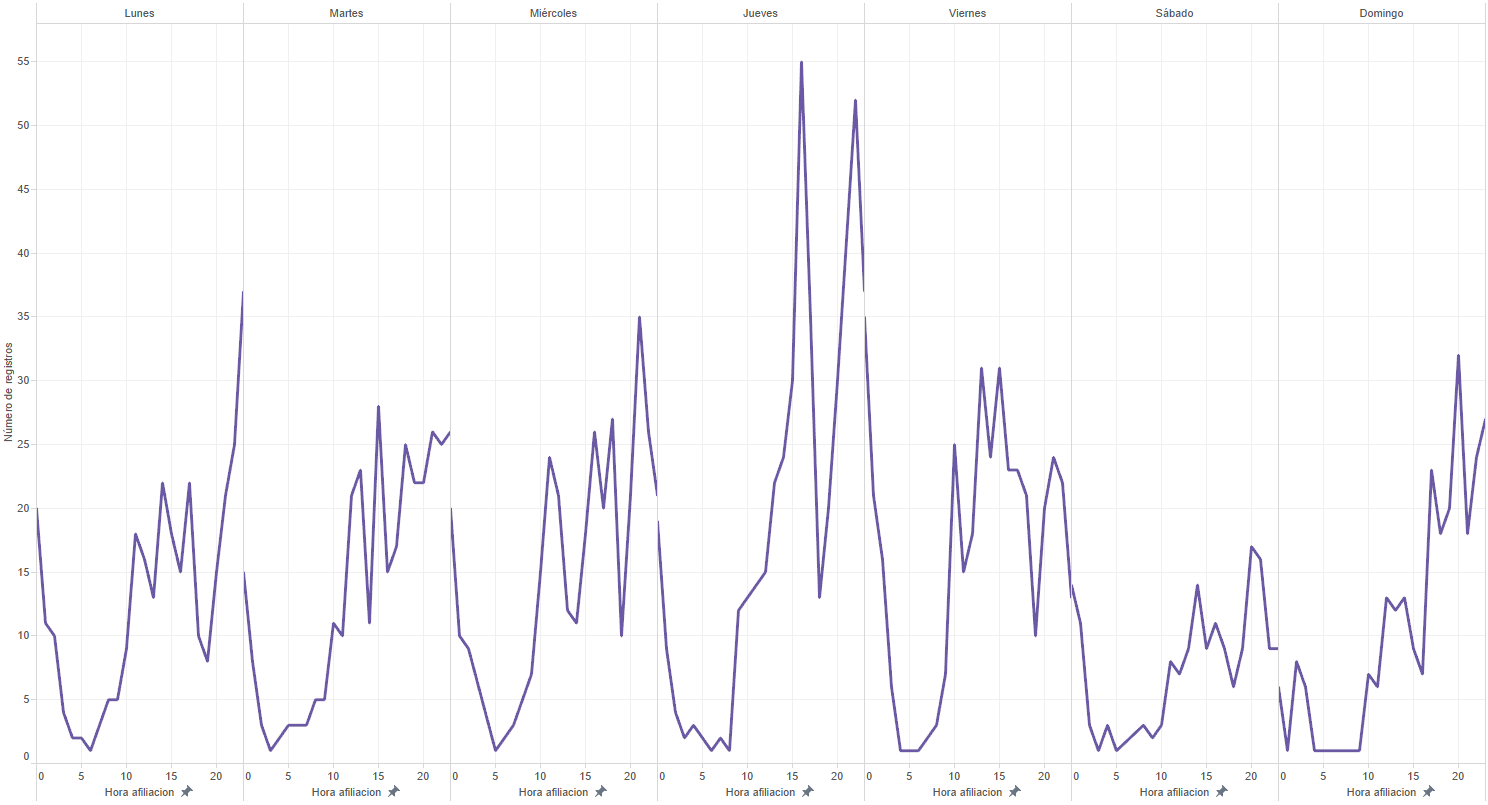


C:\Users\J\AppData\Local\Microsoft\Windows\INetCache\Content.Word\_duracionVSshares_leyenda.png

En la Ilustración 6: Veces compartidos VS duración, se ilustra la distribución de los videos de la plataforma acorde a su duración, comparando esto con la cantidad promedio de veces que es compartido un video de ese rango de duración. El eje horizontal enumera los rangos de duración utilizados, partiendo desde 0, y con intervalos de 30 segundos. El eje vertical representa la cantidad de veces que es compartido en promedio un video de ese rango de duración. Finalmente, el color de las barras representa qué tan significativos son esos datos (cantidad de registros en ese rango de duración).

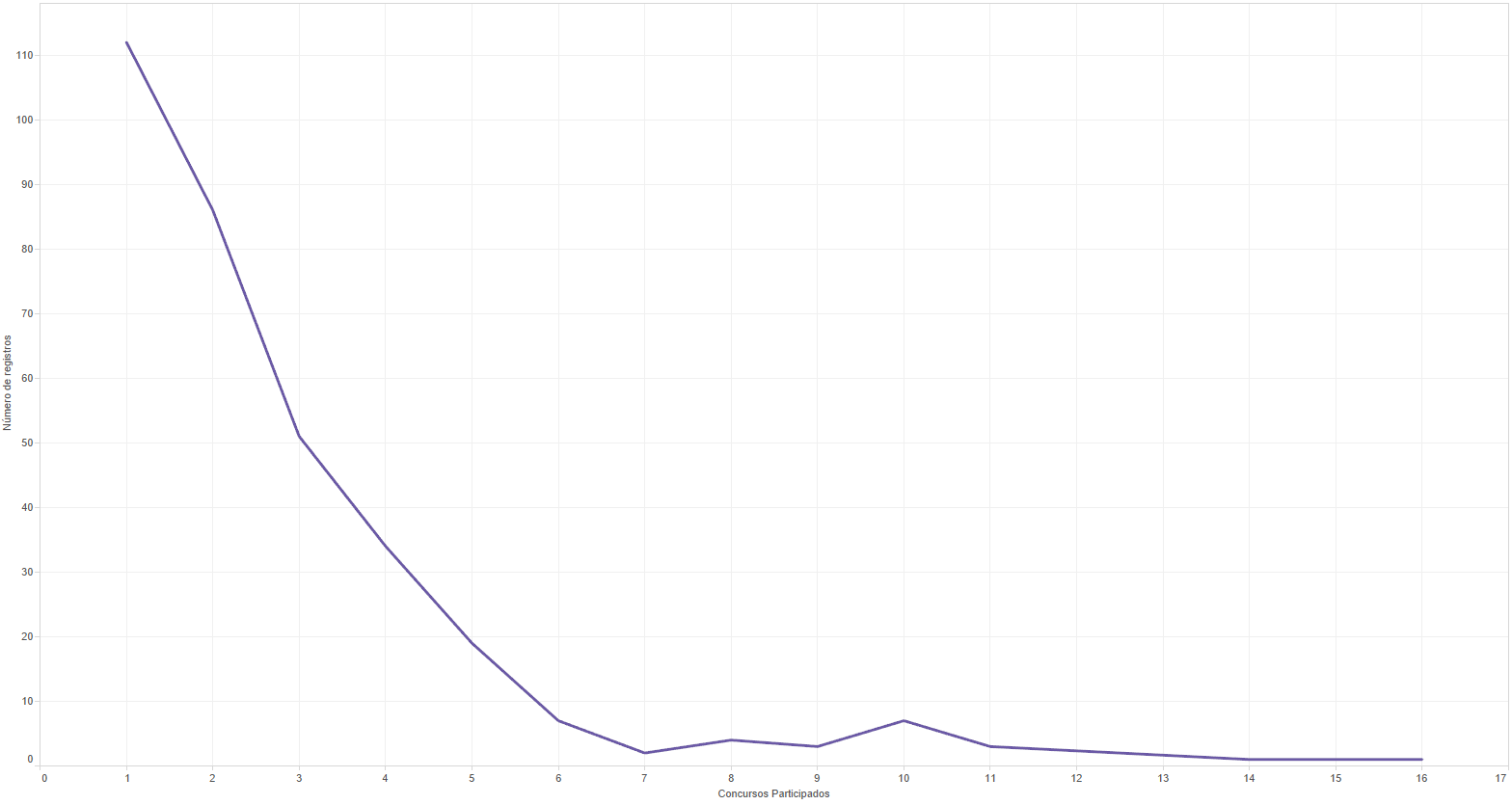
De esta ilustración se puede desprender que los videos de menor duración son compartidos más veces que los videos de duraciones más largas. Se puede apreciar una excepción en el rango de **6:01 a 6:30**, pero del color de la barra se desprende que se trata de muy pocos registros (8 registros), y dentro de ellos, se detectó un *outlier* con 121 veces compartido (en comparación al resto de los registros de ese grupo, que cuentan con un promedio aproximado de 7 veces compartido).

Ilustración 7: Actividad de registro de nuevos usuarios en días de la semana



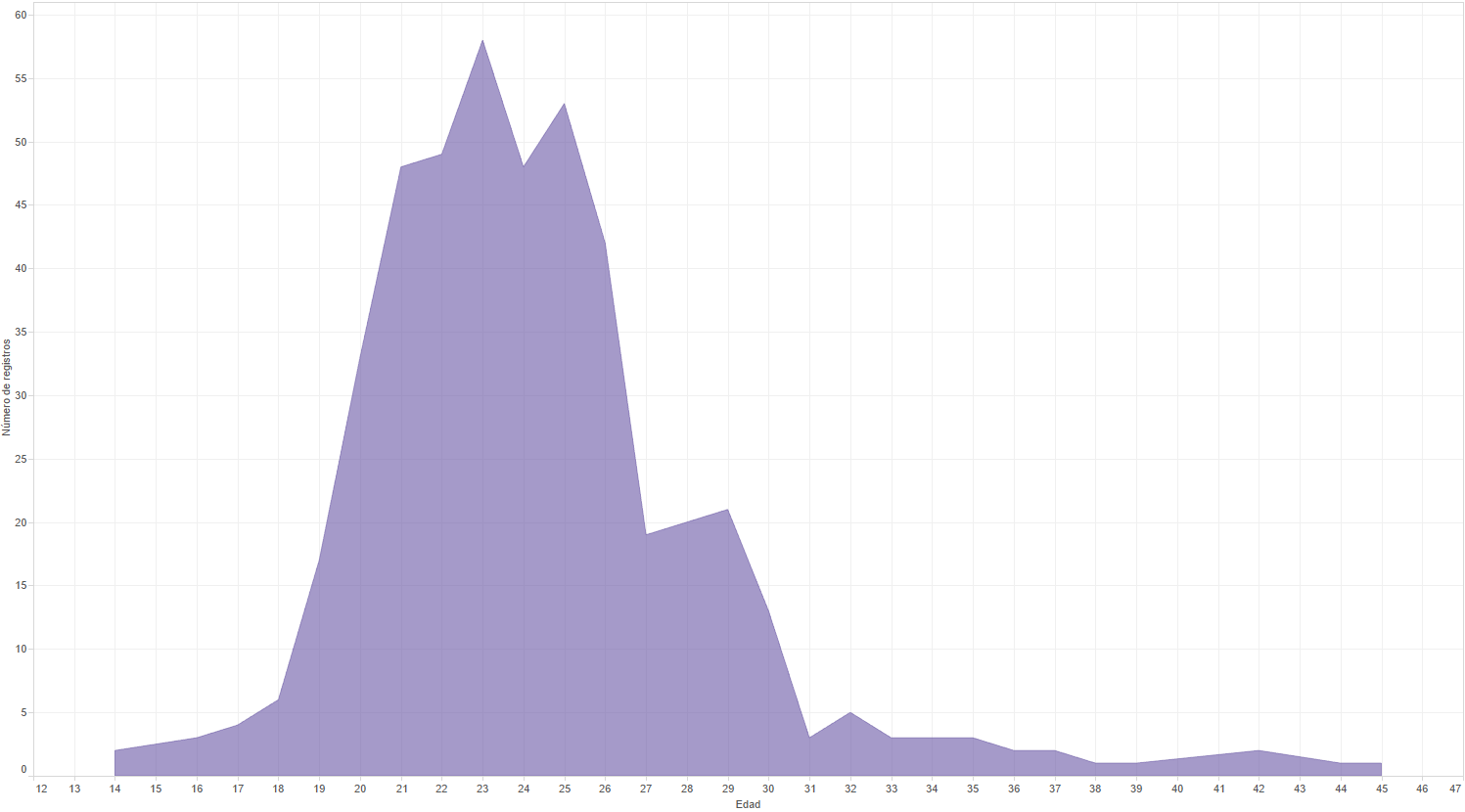
En laIlustración 7: Actividad de registro de nuevos usuarios en días de la semana, se muestra cómo actúa el registro de nuevos usuarios en la plataforma a lo largo de la semana. Al igual que en la **Ilustración 3: Actividad de videos compartidos en días de la semana**, y la **Ilustración 4: Actividad de vistas en días de la semana**, se aprecia una fuerte diferencia de actividad en el día jueves con respecto al resto de los días. Además, se desprende que el sábado es el día con peor actividad en torno a registro de nuevos usuarios en la plataforma.

Ilustración 8: Distribución de usuarios en concursos participados



En la **Ilustración 8: Distribución de usuarios en concursos participados**, se ilustra cómo se distribuyen los usuarios en torno a los concursos que participan. Se puede apreciar una suerte de ley de potencia, en la que la mayoría de los usuarios participan en muy pocos sorteos, y unos pocos participan en muchos. De esta ilustración se quitaron los usuarios que no han participado en ningún sorteo (gran mayoría, aproximadamente un 85% de los registros.)

Ilustración 9: Distribución de usuarios en edad



En la **Ilustración 9: Distribución de usuarios en edad**, se ilustra, como dice el título, la distribución de los usuarios de la plataforma en torno a su edad. En el gráfico, se omitieron los usuarios que no compartieron su edad en Facebook (gran mayoría, aproximadamente el 82% de los usuarios). Se desprende de la ilustración que el 78% de los usuarios que comparten su edad con el sistema, se encuentran entre los 19 y 30 años de edad.

Cabe destacar que si bien la herramienta no fue definitiva para el estudio, se logra tener un primer acercamiento a los usuarios y a la actividad de la plataforma en general, como por ejemplo, el hecho de que una gran porción de los usuarios registrados nunca aportaron al funcionamiento de Kikvi.

* 1. Proceso de descubrimiento del conocimiento

Como se revisó en la sección **2.1.1.1: Proceso de descubrimiento del conocimiento KDD**, este proceso se divide en 5 etapas fundamentales (selección, pre-proceso, transformación, minería de datos, interpretación/evaluación). La primera de estas fases fue descrita en la sección 3.2: Selección de los datos.

* + 1. KDD: Pre-proceso

El pre procesamiento de los datos es una fase fundamental al momento de realizar minería de datos. Durante esta fase se realizan tareas como la discretización de los datos. Diferentes parámetros y decisiones durante esta fase influyen directamente en el resultado final del proceso.

Durante esta etapa se pre-procesaron una serie de variables del estudio, utilizando funciones escritas en R que se pueden apreciar en el Anexo 4: Funciones de pre-procesamiento. En la Tabla 2: Variables pre-procesadas, se listan las variables que fueron pre-procesadas, junto con las funciones usadas y el objetivo final de este procedimiento.

Tabla 2: Variables pre-procesadas

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Tabla | Función utilizada | Resultado |
| dia\_afiliacion | usuarios | getDayName | Se pasa la información del día de afiliación de usuarios de un formato de [0-6] a [“Lunes”-“Domingo”]. |
| edad | usuarios | getEdad | Se usa la fecha de nacimiento de los usuarios para deducir su edad. |
| tickets\_canejados | usuarios | cleanTicketsCanjeados | Se limpian los valores N/A de la variable reemplazándolos por 0. |
| share\_frequency  (nueva variable) | usuarios | setUserShareFrequency | Se calcula la frecuencia ingenua en que comparte un usuario en la plataforma, en base a su actividad total y la cantidad de veces que compartió. |
| release\_difference  (nueva variable) | videos | releaseDifferenceToDays | Se calcula la diferencia tiempo de lanzamiento del video entre YouTube y Kikiv, en días |
| avg\_ppv  (nueva variable) | videos | getAvgPpv | Se calcula la cantidad de puntos promedio que el video entrega por vista. |
| is\_depleted  (nueva variable) | videos | isDepleted | Cada video cuenta con un presupuesto límite, se calcula si se entregó la totalidad de este presupuesto a los usuarios (a través de vistas) |
| release\_day | video | getReleaseDay | Se define el día (lunes a domingo) de lanzamiento del video. |

Además se utilizó la función “*discretize”* del paquete “*arules*” (en R) para discretizar las siguientes variables. Se decide utilizar el método de *clustering* (implementado en la función) para agrupar las variables de acuerdo a su cercanía. En la Tabla 3: Variables discretizadas se listan las variables continuas y su contraparte discretizada.

Tabla 3: Variables discretizadas

|  |  |
| --- | --- |
| Variable continua | Variable discreta |
| tickets\_canjeados | rango\_tickets\_canjeados |
| puntos\_historicos | rango\_puntos\_historicos |
| puntos\_gastados | rango\_puntos\_gastados |
| shares\_totales | rango\_shares\_totales |
| concursos\_participados | rango\_concursos\_participados |
| shares\_frequency | rango\_shares\_frequecy |
| total\_activity | rango\_total\_activity |

Finalmente se renombraron una serie de variables de las bases de datos de estudio. La lista de variables renombradas se puede encontrar en Anexo 5: Variables re nombradas en pre-procesamiento.

* + 1. Transformación

Durante este estudio, con el fin de acomodar los datos para el uso de algoritmos de *clustering* (distancia), se normalizaron los datos. Para lograr este objetivo se utilizaron las funciones encontradas en el Anexo 6: Funciones para normalización.

En la Tabla 4: Variables normalziadas se presenta la lista de variables que fueron normalizadas. Para no reemplazar los datos, se utilizó un nuevo nombre de variable para cada una de ellas. Todas las variables fueron normalizadas utilizando tanto la técnica Min-Max como la técnica Z-Scores.

Tabla 4: Variables normalziadas

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla | Método | Variable | Variable normalizada |
| users | Min-Max | puntos\_historicos | normal\_mm\_puntos\_historicos |
| users | Min-Max | puntos\_gastados | normal\_mm\_puntos\_gastados |
| users | Min-Max | shares\_totales | normal\_mm\_shares\_totales |
| users | Min-Max | tickets\_canjeados | normal\_mm\_tickets\_canjeados |
| users | Min-Max | concursos\_participados | normal\_mm\_concursos\_participados |
| users | Min-Max | total\_activity | normal\_mm\_total\_activity |
| users | Z-Scores | puntos\_historicos | normal\_zs\_puntos\_historicos |
| users | Z-Scores | puntos\_gastados | normal\_zs\_puntos\_gastados |
| users | Z-Scores | shares\_totales | normal\_zs\_shares\_totales |
| users | Z-Scores | tickets\_canjeados | normal\_zs\_tickets\_canjeados |
| users | Z-Scores | concursos\_participados | normal\_zs\_concursos\_participados |
| users | Z-Scores | total\_activity | normal\_zs\_total\_activity |
| videos | Min-Max | duracion | normal\_mm\_duracion |
| videos | Min-Max | release\_difference | normal\_mm\_release\_difference |
| videos | Min-Max | total\_views | normal\_mm\_total\_views |
| videos | Min-Max | points\_given | normal\_mm\_points\_given |
| videos | Min-Max | shares\_first\_day | normal\_mm\_shares\_first\_day |
| videos | Min-Max | shares\_first\_week | normal\_mm\_shares\_first\_week |
| videos | Min-Max | shares\_first\_month | normal\_mm\_shares\_first\_month |
| videos | Min-Max | total\_shares | normal\_mm\_total\_shares |
| videos | Min-Max | total\_users | normal\_mm\_total\_users |
| videos | Min-Max | active\_users | normal\_mm\_active\_users |
| videos | Min-Max | new\_users | normal\_mm\_new\_users |
| videos | Min-Max | active\_raffles | normal\_mm\_active\_raffles |
| videos | Min-Max | avg\_ppv | normal\_mm\_avg\_ppv |
| videos | Z-Scores | duracion | normal\_zs\_duracion |
| videos | Z-Scores | release\_difference | normal\_zs\_release\_difference |
| videos | Z-Scores | total\_views | normal\_zs\_total\_views |
| videos | Z-Scores | points\_given | normal\_zs\_points\_given |
| videos | Z-Scores | shares\_first\_day | normal\_zs\_shares\_first\_day |
| videos | Z-Scores | shares\_first\_week | normal\_zs\_shares\_first\_week |
| videos | Z-Scores | shares\_first\_month | normal\_zs\_shares\_first\_month |
| videos | Z-Scores | total\_shares | normal\_zs\_total\_shares |
| videos | Z-Scores | total\_users | normal\_zs\_total\_users |
| videos | Z-Scores | active\_users | normal\_zs\_active\_users |
| videos | Z-Scores | new\_users | normal\_zs\_new\_users |
| videos | Z-Scores | active\_raffles | normal\_zs\_active\_raffles |
| videos | Z-Scores | avg\_ppv | normal\_zs\_avg\_ppv |

En base a las variables generadas, renombradas y transformadas, finalmente se generaron una serie de vistas minables (*minable views*), las que se describen la Tabla 5: Vistas minables.

Tabla 5: Vistas minables

|  |  |
| --- | --- |
| Vista Minable | Descripción |
| users\_num | Contiene todas las variables numéricas relacionadas con los usuarios del estudio (no normalizadas). |
| users\_num\_mm | Contiene todas las variables numéricas relacionadas con los usuarios del estudio, normalizadas a través del método **Min-Max**. |
| users\_num\_zs | Contiene todas las variables numéricas relacionadas con los usuarios del estudio, normalizadas a través del método **Z-Score**. |
| users\_dis | Contiene todas las variables discretas relacionadas con los usuarios del estudio. |
| videos\_num | Contiene todas las variables numéricas relacionadas con los videos del estudio (no normalizadas). |
| videos\_num\_mm | Contiene todas las variables numéricas relacionadas con los videos del estudio, normalizadas a través del método **Min-Max**. |
| videos\_num\_zs | Contiene todas las variables numéricas relacionadas con los videos del estudio, normalizadas a través del método **Z-Score**. |
| videos\_dis | Contiene todas las variables discretas relacionadas con los videos del estudio. |

Estas vistas minables son las que se utilizarán para las diferentes tareas de minería de datos. Se generan vistas de variables discretas, numéricas y numéricas normalizadas para así satisfacer los requerimientos de entrada de diferentes algoritmos a considerar.

* + 1. Minería de datos

Como se mencionó en el último párrafo de 2.1.4: Herramientas de minería de datos, la herramienta utilizada en por este estudio es R.

* + - 1. Reglas de asociación

Como primer acercamiento, se utilizan algoritmos de reglas de asociación para así tener un acercamiento a cómo se relacionan (y si es que lo hacen) las variables en cuestión. Para este fin se usan las vistas minables **users\_dis** y **videos\_dis**, y los algoritmos “*apriori*” y “*eclat*” implementados en el paquete “*arules*”[[20]](#footnote-20).

* + - * 1. Apriori

Primero, se aplicará el algoritmo *a priori* sobre las vistas minables tanto de usuarios como de videos. En primera instancia, se aplica sobre la vista minable de usuarios (users\_dis, que contiene todas las variables nominales relacionadas con los usuarios de la plataforma), a través de los siguientes comandos:

1 > apriori\_users <- apriori(users\_dis, list(support=0.6))

2 > inspect(apriori\_users)

En la línea 1 del comando, se aprecia que el segundo parámetro entregado a la función es list(support=0.6)), esto quiere decir que sólo se considerarán las reglas de asociación cuyo soporte sea de al menos 60%. Esto con el fin de disminuir los resultados entregados por el algoritmo. En la línea 2 se inspecciona el resultado, revelando 448 reglas de asociación encontradas. Una mirada rápida al resultado (¿Será necesario mostrar el resultado? ¿Las 448 reglas?) revela que la gran mayoría de las reglas de asociación encontradas resuelven implicancias de rengos en los que las variables son (o pueden ser) 0, lo que se refleja acorde en la información de interacción usuaria mostrada en la Ilustración 5: Cantidad de usuarios por su calidad. Se hace necesario entonces refinar el resultado.

Como primera medida, se decide no incluir en el resultado los rangos que incluyen el valor 0 en cada variable, lo que se hace a través de los siguientes comandos:

1 > apriori\_users\_appereance\_list = list(

none = c(

"rango\_tickets\_canjeados=[ 0.00, 6.43)",

"rango\_puntos\_historicos=[ 0, 4651)",

"rango\_puntos\_gastados=[ 0, 4252)",

"rango\_shares\_totales=[ 0.00, 8.69)",

"rango\_concursos\_participados=[ 0.00, 1.21)",

"rango\_shares\_frequecy=[ 0.00, 3.93)",

"rango\_total\_activity=[ 0.0, 33.8)"

),

default = "both"

)

2 > apriori\_users <- apriori(

users\_dis,

parameter =list(support=0.05,confidence=0.4),

appearance = apriori\_users\_appereance\_list

)

3 >inspect(apriori\_users)

En la línea 1 de este comando se crea una lista restrictiva que luego se pasa como parámetro a la función *apriori*, limitando así las ocurrencias de las variables utilizadas. El resultado de este acercamiento da como resultado una única regla de asociación que se puede apreciar en la Tabla 6: Apriori usuarios: Reglas de asociación.

Tabla 6: Apriori usuarios: Reglas de asociación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LHS[[21]](#footnote-21) |  | RHS[[22]](#footnote-22) | Soporte | Confianza |
| dia\_afiliacion=Lunes | => | genero=M | 0.05282877 | 0.4051724 |

Lamentablemente, esta regla no aporta valor al objetivo, básicamente está diciendo que, con una confianza del 40%, el 5% de los datos reflejan que, si un usuario se registra el día lunes, ese usuario es de género masculino.

Se deshecha entonces la primera medida tomada. Como segunda medida, se evalúan las variables utilizadas para la generación de reglas de asociación. Las variables iniciales de **users\_dis** son las siguientes:

> str(users\_dis)

'data.frame': 2669 observaciones de 13 variables:

$ categoria\_dominante : Factor con 7 niveles

$ uni : Factor con 33 niveles

$ genero : Factor con 2 niveles

$ hora\_afiliacion : Factor con 24 niveles

$ dia\_afiliacion : Factor con 7 niveles

$ edad : Factor con 32 niveles

$ rango\_tickets\_canjeados : Factor con 5 niveles

$ rango\_puntos\_historicos : Factor con 5 niveles

$ rango\_puntos\_gastados : Factor con 5 niveles

$ rango\_shares\_totales : Factor con 5 niveles

$ rango\_concursos\_participados: Factor con 5 niveles

$ rango\_shares\_frequecy : Factor con 5 niveles

$ rango\_total\_activity : Factor con 5 niveles

Se decide considerar las variables propias del usuario (las primeras 6), y sólo el rango de *shares\_totales*, además, se decide disminuir el límite de soporte, ya que se sabe que la mayoría de los datos de usuario no generan información de interacción. Este nuevo acercamiento no genera reglas de asociación que otorguen mayor información para el objetivo.

En segundo lugar, se procede a repetir el procedimiento, esta vez utilizando la vista minable de videos (videos\_dis), a través de los siguientes comandos:

1 > apriori\_videos <- apriori(

videos\_dis,

parameter = list(support=0.8, confidence=0.8)

)

2 > inspect(apriori\_videos)

Como se puede apreciar en parameter = list(support=0.8, confidence=0.8), se busca solo reglas que reflejen al menos el 80% de los datos, con una confianza del 80%. En la Tabla 7: Apriori videos: Reglas de asociación, se muestran los resultados de la ejecución del algoritmo a priori sobre la vista minable de videos (sólo variable nominales).

Tabla 7: Apriori videos: Reglas de asociación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | LHS |  | RHS | Soporte | Confianza |
| 1 | - | => | rango\_total\_views=[1, 386) | 0.8415 | 0.8415 |
| Esta regla no tiene condiciones izquierdas, lo que quiere decir que simplemente se trata de una descripción de la muestra. En particular, de esta regla se deduce que un 84.15% de los videos tienen entre 1 y 385 vistas. | | | | |
| 2 | - | => | rango\_points\_given=[0, 13254) | 0.8426 | 0.8426 |
| Esta regla no tiene condiciones izquierdas, lo que quiere decir que simplemente se trata de una descripción de la muestra. En particular, de esta regla se deduce que un 84.26% entregaron entre 0 y 13254 puntos. | | | | |
| 3 | - | => | rango\_release\_difference=[0, 51.8) | 0.8480 | 0.8480 |
| Esta regla no tiene condiciones izquierdas, lo que quiere decir que simplemente se trata de una descripción de la muestra. En particular, de esta regla se deduce que un 84.8% de los videos fueron publicados en la plataforma entre 0 y 51 días después de su publicación en Facebook. | | | | |
| 4 | - | => | is\_depleted=0 | 0.9783 | 0.9783 |
| Esta regla no tiene condiciones izquierdas, lo que quiere decir que simplemente se trata de una descripción de la muestra. En particular, de esta regla se deduce que un 97.83% de los videos no usaron la totalidad de su presupuesto asignado al momento en que se consultaron los datos de este estudio. | | | | |
| 5 | rango\_total\_views=[1, 386) | => | rango\_points\_given=[0, 13254) | 0.8219 | 0.9768 |
| Esta regla no tiene valor, ya que es lógico que si un video tiene sólo entre 1 y 385 vistas, este caerá en el primer intervalo de puntos dados, ya que son variables que se encuentra relacionadas (la única forma que tiene un video de entregar puntos es al generar vistas). | | | | |
| 6 | rango\_points\_given=[0, 13254) | => | rango\_total\_views=[1, 386) | 0.8219 | 0.9755 |
| Esta regla no tiene valor, ya que es lógico que si un video se encuentra en el primer rango de puntos dados, es probable que se encuentre en el primer rango de vistas totales. | | | | |
| 7 | rango\_total\_views=[1, 386) | => | is\_depleted=0 | 0.8404 | 0.9987 |
| Esta regla no tiene valor ya que es de esperarse que si un video se encuentra en el primer rango de vistas, a este no se le haya agotado el presupuesto. | | | | |
| 8 | is\_depleted=0 | => | rango\_total\_views=[1, 386) | 0.8404 | 0.8590 |
| De esta regla se deduce que gran cantidad de los videos que aún tienen presupuesto tienen entre 1 y 385 vistas, lo que coincide con la primera regla de esta tabla, que dice que la gran cantidad de videos tiene entre 1 y 385 vistas. | | | | |
| 9 | rango\_points\_given=[0, 13254) | => | is\_depleted=0 | 0.8404 | 0.9974 |
| Esta regla no tiene valor ya que es lógico que si un video se encuentra en el primer rango de puntos otorgados, a este aún le quede presupuesto. | | | | |
| 10 | is\_depleted=0 | => | rango\_points\_given=[0, 13254) | 0.8404 | 0.8590 |
|  | | | | |
| 11 | rango\_release\_difference=[0, 51.8) | => | is\_depleted=0 | 0.8328 | 0.9821 |
|  | | | | |
| 12 | is\_depleted=0 | => | rango\_release\_difference=[0, 51.8) | 0.8328 | 0.8513 |
|  | | | | |
| 13 | rango\_total\_views=[1, 386),  rango\_points\_given=[0, 13254) | => | is\_depleted=0 | 0.8208 | 0.9987 |
|  | | | | |
| 14 | is\_depleted=0,  rango\_total\_views=[1, 386) | => | rango\_points\_given=[0, 13254) | 0.8208 | 0.9767 |
|  | | | | |
| 15 | is\_depleted=0,  rango\_points\_given=[0, 13254) | => | rango\_total\_views=[1, 386) | 0.8208 | 0.9767 |
|  | | | | |

Las 4 primeras reglas de asociación entregadas por el algoritmo simplemente describen los datos. Por ejemplo, la cuarta regla de la tabla, es el reflejo de que el **97.83%** de los datos (aproximadamente) cumple que **is\_depleted=0**. Estas reglas no aportan valor al negocio, es simplemente una descripción de la distribución de los datos.

En base a estas reglas surge la duda de si hay alguna condición que implique que el presupuesto de un video se utilice en su totalidad (**is\_depleted=1**), por lo que se buscan reglas que cumplan con esta nueva condición de la siguiente manera.

1 > apriori\_videos\_appereance\_list = list(

rhs = c(

"is\_depleted=1"

),

default = "lhs"

)

2 > apriori\_videos\_depleted <- apriori(

videos\_dis,

parameter = list(support=0.01, confidence=0.1),

appearance = apriori\_videos\_appereance\_list

)

3 > inspect(apriori\_videos\_depleted)

Como se aprecia en el segundo comando, en este caso se buscan las reglas con un mínimo de soporte del 1% y confianza del 10%. Se bajaron estos valores en comparación a la ejecución anterior ya que las reglas con alto nivel de soporte no cumplen con esta regla (tomando en cuenta que solo un **2.17%** de los datos cumplecon **is\_depleted=1**). En la Tabla 8: Apriori videos agotados: Reglas de asociación se aprecian las reglas resultantes.

Tabla 8: Apriori videos agotados: Reglas de asociación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | LHS |  | RHS | Soporte | Confianza |
| 1 | rango\_points\_given=[ 93366, 2790607) | => | is\_depleted=1 | 0.0130 | 0.5455 |
| 2 | rango\_release\_difference=[0, 51.8),  rango\_points\_given=[93366, 2790607) | => | is\_depleted=1 | 0.0109 | 0.6667 |
| 3 | rango\_total\_views=[386, 1484),  rango\_total\_users=[3,1633) | => | is\_depleted=1 | 0.0109 | 0.1923 |

* + - * 1. Eclat
    1. Observaciones

1. Conclusiones
2. Bibliografía

[1] **UCLAAnderson** - *school of management,* http://www.anderson.ucla.edu

[2] KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW - *Azevedo & Santos*

[3] “Performance Dashboards: Measuring, Monitoring, and Managing your Business”

[4] **Advanced Tech Computing Group UTPL** - http://advancedtech.wordpress.com/

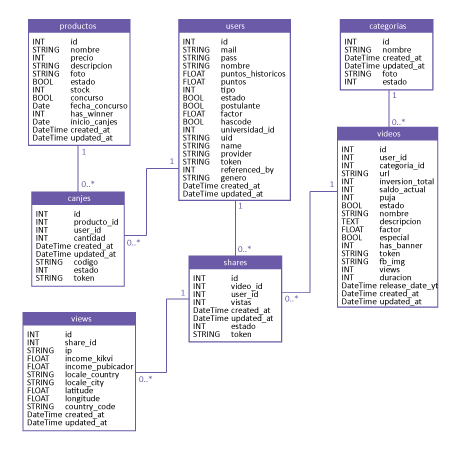
[5] **OLAP TOOLS** - http://www.informationbuilders.com/olap-online-analytical-processing-tools

1. Referencias

* Referencia 1 - Curso Bussiness Inteligence -- UTFSM
* Referencia 2 - Identification of Outliers -- Douglas M. Hawkins, 1980
* Referencia 3 - Indicadores Claves de Desempeño o Key Performance Indicator -- http://www.profitline.com.co/BPO/BusinessProcessOutsourcing/182/indicadores-claves-de-desempeno-o-key-performance-indicator.html
* Referencia 4 - KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW - Azevedo & Santos
* Referencia 5 - G. Towell, J. Shavlik, and M. Noordewier, Refinement of approximate domain theories by knowledge-based neural networks", 1990.
* Referencia 6 - H. MAYER, Huber, Rohde, and Tamme, Rule extraction from artificial neural networks", Universitsat Salzburg, 12th October 2006 2006.

1. Anexos
   1. Anexo 1: Modelo de datos de Kikvi

Se muestran a continuación las tablas relevantes al estudio, se omiten tablas intermedias en relaciones de muchos a muchos, tablas obsoletas, y extensiones de tablas.



* 1. Anexo 2: Scripts para APIs
     1. Facebook (PHP)

<?php

require("fbSDK/autoload.php");

use Facebook\FacebookSession;

use Facebook\FacebookRequest;

use Facebook\GraphUser;

use Facebook\FacebookRequestException;

use Facebook\FacebookRedirectLoginHelper;

$facebook = FacebookSession::setDefaultApplication(<app\_id>, <app\_token>);

$session = FacebookSession::newAppSession();

$sql = "SELECT

users.\*

FROM

users

WHERE

users.id != 0

AND users.id != 8

AND users.tipo = 1

AND users.estado = 1

AND users.uid IS NOT NULL";

$db = mysql\_connect("localhost",<db\_user>,<db\_pass>);

$selected = mysql\_select\_db(<db\_name>);

$rs = mysql\_query($sql);

$getFromFacebook = array();

while($row = mysql\_fetch\_assoc($rs)){

if(!empty($row["uid"])){

$tmp = new StdClass();

$tmp->id = $row["id"];

$tmp->fbid = $row["uid"];

array\_push($getFromFacebook, $tmp);

}

}

mysql\_close();

$total = sizeof($getFromFacebook);

$current = 1;

$updateQuery = "";

foreach($getFromFacebook as $user){

print\_r("Fetching... ".$current."/".$total."\n");

print\_r("/".$user->fbid."\n");

$request = new FacebookRequest($session, 'GET', '/'.$user->fbid);

try{

$response = $request->execute();

$graphObject = $response->getGraphObject();

$gender = $graphObject->getProperty('gender');

$gender = strtoupper(substr($gender,0,1));

$updateQuery .= "UPDATE users SET genero = '$gender' WHERE id = '$user->id';";

}

catch (Exception $e){

print\_r("Error: perfil borrado"."\n");

}

$current++;

}

$file = "sqlQuery.sql";

file\_put\_contents($file, $updateQuery);

?>

* + 1. Youtube (RoR)

def get\_duracion\_videos\_from\_yt

if not params[:videos].nil?

concatenated\_yt\_ids = Video.where(:id => params[:videos]).map(&:url).join(",")

url = <GOOGLE API URL WITH PRIVATE TOKEN>

begin

video\_info = open(url)

video\_info = JSON.parse video\_info.read

video\_info["items"].each do |v|

duration = v["contentDetails"]["duration"]

seconds = 0

duration = duration.sub! "PT", ""

if duration.include? "H" and duration.include? "M" and duration.include? "S"

duration\_hours = duration.split("H")[0].to\_i

duration\_minutes = duration.split("M")[0].split("H")[1].to\_i

duration\_seconds = duration.split("M")[1].gsub! "S",""

duration\_seconds = duration\_seconds.to\_i

seconds = duration\_hours\*3600 + duration\_minutes\*60 + duration\_seconds

elsif duration.include? "M" and duration.include? "S"

duration\_minutes = duration.split("M")[0].to\_i

duration\_seconds = duration.split("M")[1].gsub! "S",""

duration\_seconds = duration\_seconds.to\_i

seconds = duration\_minutes\*60 + duration\_seconds

elsif duration.include? "H" and duration.include? "M"

duration\_hours = duration.split("H")[0].to\_i

duration\_minutes = duration.split("H")[1].gsub! "M",""

duration\_minutes = duration\_minutes.to\_i

seconds = duration\_hours\*3600 + duration\_minutes\*60

elsif duration.include? "H" and duration.include? "S"

duration\_hours = duration.split("H")[0].to\_i

duration\_seconds = duration.split("H")[1].gsub! "S",""

duration\_seconds = duration\_seconds.to\_i

seconds = duration\_hours\*3600 + duration\_seconds

elsif duration.include? "H"

duration\_hours = duration.split("H")[0].to\_i

seconds = duration\_hours\*3600

elsif duration.include? "M"

duration\_minutes = duration.split("M")[0].to\_i

seconds = duration\_minutes\*60

elsif duration.include? "S"

duration\_seconds = duration.split("S")[0].to\_i

seconds = duration\_seconds

end

Video.where(:url => v["id"]).update\_all(:duracion => seconds)

end

rescue StandardError=>e

puts "WOOOPS"

end

end

render :nothing => true

end

* 1. Anexo 3: Selección de los datos (R)

if(!require("RMySQL")){

install.packages("RMySQL",dep=TRUE)

}

library(RMySQL)

warn=1

con <- dbConnect(MySQL(), user=<db\_user>, password=<db\_pass>, dbname=<db\_name>, host="localhost")

# QUERY PARA VIDEOS

videos <- dbGetQuery(con,

"SELECT

videos.id,

url,

saldo\_actual,

categorias.nombre AS category,

puja AS points\_per\_view,

videos.created\_at AS release\_date,

videos.release\_date\_yt AS release\_date\_youtube,

videos.duracion AS duracion,

IFNULL(

TIMESTAMPDIFF(SECOND, videos.release\_date\_yt, videos.created\_at),-1

) AS release\_difference,

views AS total\_views,

(inversion\_total - saldo\_actual) AS points\_given,

#

#

# INFORMACION DE CANTIDAD DE SHARES

#

# \* Primer dia

# \* Primera semana

# \* Primer mes

# \* Totales

# \* Males

# \* Females

#

(SELECT

count(shares.id)

FROM

shares

WHERE

shares.video\_id = videos.id

AND shares.created\_at <= DATE\_ADD(videos.created\_at, INTERVAL 1 DAY)

GROUP BY

videos.id

) AS shares\_first\_day,

(SELECT

count(shares.id)

FROM

shares

WHERE

shares.video\_id = videos.id

AND shares.created\_at <= DATE\_ADD(videos.created\_at, INTERVAL 1 WEEK)

GROUP BY

videos.id

) AS shares\_first\_week,

(SELECT

count(shares.id)

FROM

shares

WHERE

shares.video\_id = videos.id

AND shares.created\_at <= DATE\_ADD(videos.created\_at, INTERVAL 1 MONTH)

GROUP BY

videos.id

) AS shares\_first\_month,

(SELECT

count(1)

FROM

shares

JOIN users ON users.id = shares.user\_id

WHERE

video\_id = videos.id

AND IFNULL(users.genero, (SELECT sexo FROM infos WHERE infos.user\_id = users.id)) = 'M'

) AS male\_shares,

(SELECT

count(1)

FROM

shares

JOIN users ON users.id = shares.user\_id

WHERE

video\_id = videos.id

AND IFNULL(users.genero, (SELECT sexo FROM infos WHERE infos.user\_id = users.id)) = 'F'

) AS female\_shares,

(SELECT

count(shares.id)

FROM

shares

WHERE

shares.video\_id = videos.id

GROUP BY

videos.id

) AS total\_shares,

#

#

# INFORMACION DE CANTIDAD DE USUARIOS

#

#

# \* Usuarios @ release

# \* Usuarios activos la semana anterior al lanzamiento

# \* Usuarios activos hasta 2 semanas anteriores al lanzamiento

# \* Usuarios nuevos la semana anteriore al lanzamiento

# \* Usuarios nuevos hasta 2 semanas anteriores al lanzamiento

#

(SELECT

count(users.id)

FROM

users

WHERE

users.created\_at <= videos.created\_at

) AS total\_users\_at\_release,

(SELECT

count(distinct shares.user\_id)

FROM

shares

WHERE

shares.created\_at >= DATE\_SUB(videos.created\_at, INTERVAL 1 WEEK)

AND shares.created\_at <= videos.created\_at

) AS 1\_week\_active\_users\_at\_release,

(SELECT

count(distinct shares.user\_id)

FROM

shares

WHERE

shares.created\_at >= DATE\_SUB(videos.created\_at, INTERVAL 2 WEEK)

AND shares.created\_at <= videos.created\_at

) AS 2\_week\_active\_users\_at\_release,

(SELECT

count(\*)

FROM

users

WHERE

users.created\_at >= DATE\_SUB(videos.created\_at, INTERVAL 1 WEEK)

AND users.created\_at <= videos.created\_at

) AS 1\_week\_new\_users\_at\_release,

(SELECT

count(\*)

FROM

users

WHERE

users.created\_at >= DATE\_SUB(videos.created\_at, INTERVAL 2 WEEK)

AND users.created\_at <= videos.created\_at

) AS 2\_week\_new\_users\_at\_release,

#

#

# INFORMACION DE CANTIDADES DE CONCURSOS

#

# \* Raffles @ release

# \* Raffles primera semana

# \* Raffles primer mes

#

(SELECT

count(\*)

FROM

productos

WHERE

concurso = 1

AND fecha\_concurso >= videos.created\_at

AND productos.created\_at <= videos.created\_at

) AS active\_raffles\_at\_release,

(SELECT

count(\*)

FROM

productos

WHERE

concurso = 1

AND fecha\_concurso >= videos.created\_at

AND productos.created\_at <= DATE\_ADD(videos.created\_at, INTERVAL 1 WEEK)

) AS 1\_week\_active\_raffles,

(SELECT

count(\*)

FROM

productos

WHERE

concurso = 1

AND fecha\_concurso >= videos.created\_at

AND productos.created\_at <= DATE\_ADD(videos.created\_at, INTERVAL 2 WEEK)

) AS 2\_week\_active\_raffles

FROM

videos

LEFT JOIN categorias ON categorias.id=categoria\_id

WHERE

videos.views > 0

ORDER BY

total\_views DESC, release\_date ASC"

)

write.table(

videos,

file="/home/jleon/Memoria/Data/videos.csv",

append=FALSE,

quote=FALSE,

sep=";",

eol="\n",

na="NA",

row.names=FALSE,

col.names=TRUE,

fileEncoding="UTF-8"

)

# QUERY PARA USUARIOS

users <- dbGetQuery(con,

"SELECT

u.id,

puntos\_historicos,

puntos,

u.created\_at AS fecha\_afiliacion,

IFNULL(universidades.nombre,'NA') as uni,

IFNULL(IFNULL(genero,infos.sexo),'NA') as genero,

IFNULL(infos.f\_nacimiento,'NA') as nacimiento,

HOUR(CONVERT\_TZ(u.created\_at,'+00:00','-03:00')) AS 'Hora afiliacion',

WEEKDAY(CONVERT\_TZ(u.created\_at,'+00:00','-03:00')) AS 'Dia afiliacion',

(puntos\_historicos - puntos) as puntos\_gastados,

(SELECT

count(1)

FROM

shares

WHERE

shares.user\_id = u.id

) AS shares\_totales,

(SELECT

categorias.nombre

FROM

videos

JOIN shares ON shares.video\_id = videos.id

LEFT JOIN categorias ON categorias.id = videos.categoria\_id

WHERE

shares.user\_id = u.id

AND videos.categoria\_id != 0

GROUP BY

categorias.nombre

ORDER BY

count(categoria\_id)

LIMIT

1

) AS categoria\_dominante,

(SELECT

count(1)

FROM

users

WHERE

referenced\_by = u.id

) AS recruitments,

(SELECT

count(distinct canjes.producto\_id)

FROM

canjes

JOIN productos ON productos.id = canjes.producto\_id

WHERE

canjes.user\_id = u.id

AND productos.concurso = 1

) AS concursos\_participados,

(SELECT

count(1)

FROM

canjes

JOIN productos ON productos.id = canjes.producto\_id

WHERE

canjes.user\_id = u.id

AND productos.concurso = 0

) AS premios\_canjeados,

(SELECT

sum(canjes.cantidad)

FROM

canjes

JOIN productos ON productos.id = canjes.producto\_id

WHERE

canjes.user\_id = u.id

AND productos.concurso = 1

) AS tickets\_canjeados,

IFNULL((SELECT

datediff(s1.created\_at, s2.created\_at)

FROM

shares s1

JOIN shares s2 ON s1.user\_id = s2.user\_id

AND s1.id = (SELECT max(id) FROM shares WHERE user\_id = u.id)

AND s2.id = (SELECT min(id) FROM shares WHERE user\_id= u.id)

WHERE

s1.user\_id = u.id

),-1) AS difference\_last\_and\_first\_share,

IFNULL((SELECT

stddev(

datediff(

created\_at,

IFNULL(

(SELECT

MAX( created\_at )

FROM

shares

WHERE

user\_id = s1.user\_id

AND id < s1.id

),0)

)

)

FROM

shares AS s1

WHERE

s1.user\_id = u.id

ORDER BY id ASC

),-1) AS stdv\_share\_difference,

IFNULL((SELECT

datediff(shares.created\_at, canjes.created\_at)

FROM

shares

JOIN canjes ON shares.user\_id = canjes.user\_id

AND shares.id = (SELECT max(id) FROM shares WHERE user\_id = u.id)

AND canjes.id = (SELECT min(id) FROM canjes WHERE user\_id = u.id)

WHERE

shares.user\_id = u.id

),-1) AS difference\_last\_raffle\_first\_share

FROM

users AS u

LEFT JOIN universidades ON u.universidad\_id = universidades.id

LEFT JOIN infos ON infos.user\_id = u.id

WHERE

u.tipo=1

AND u.estado=1

AND u.id != 0

AND u.id != 8"

)

write.table(

users,

file="/home/jleon/Memoria/Data/users.csv",

append=FALSE,

quote=FALSE,

sep=";",

eol="\n",

na="NA",

row.names=FALSE,

col.names=TRUE,

fileEncoding="UTF-8"

)

# QUERY PARA VISTAS

views <- dbGetQuery(con,

"SELECT

views.created\_at AS Fecha,

HOUR(CONVERT\_TZ(views.created\_at,'+00:00','-03:00')) AS 'Hora vista',

WEEKDAY(CONVERT\_TZ(views.created\_at,'+00:00','-03:00')) AS 'Dia vista',

views.locale\_country AS Pais,

views.country\_code AS 'Codigo pais',

views.locale\_city AS Ciudad,

shares.id AS Share,

shares.video\_id AS Video

FROM

views

JOIN shares ON shares.id = views.share\_id"

)

write.table(

views,

file="/home/jleon/Memoria/Data/views.csv",

append=FALSE,

quote=FALSE,

sep=";",

eol="\n",

na="NA",

row.names=FALSE,

col.names=TRUE,

fileEncoding="UTF-8"

)

# QUERY PARA SHARES

shares <- dbGetQuery(con,

"SELECT

shares.created\_at AS Fecha,

HOUR(CONVERT\_TZ(shares.created\_at,'+00:00','-03:00')) AS 'Hora share',

WEEKDAY(CONVERT\_TZ(shares.created\_at,'+00:00','-03:00')) AS 'Dia share',

count(views.id) AS vistas

FROM

shares

JOIN views ON views.share\_id = shares.id

GROUP BY

shares.id"

)

write.table(

shares,

file="/home/jleon/Memoria/Data/shares.csv",

append=FALSE,

quote=FALSE,

sep=";",

eol="\n",

na="NA",

row.names=FALSE,

col.names=TRUE,

fileEncoding="UTF-8"

)

* 1. Anexo 4: Funciones de pre-procesamiento (R)
     1. getDayName

getDayName <- function(x){

if(x==0){

as.factor("Lunes")

}

else{

if(x==1){

as.factor("Martes")

}

else{

if(x==2){

as.factor("Miercoles")

}

else{

if(x==3){

as.factor("Jueves")

}

else{

if(x==4){

as.factor("Viernes")

}

else{

if(x==5){

as.factor("Sabado")

}

else{

as.factor("Domingo")

}

}

}

}

}

}

}

* + 1. getEdad

getEdad <- function(fecha\_nacimiento){

if(!is.na(fecha\_nacimiento)){

nacimiento <- strptime(fecha\_nacimiento, format = "%Y-%m-%d")

hoy <- Sys.Date()

edad <- difftime(hoy,nacimiento)

as.factor(as.integer(as.numeric(edad, units="days")/365))

}

else{

as.factor(NA)

}

}

* + 1. cleanTicketsCanejados

cleanTicketsCanjeados <- function(tickets\_canjeados){

if(is.na(tickets\_canjeados)){

0

}

else{

as.numeric(tickets\_canjeados)

}

}

* + 1. setUserShareFrequency

setUserShareFrequency <- function(difference\_last\_and\_first\_share,shares\_totales){

if(shares\_totales ==0){

0

}

else{

difference\_last\_and\_first\_share / shares\_totales

}

}

* + 1. releaseDifferenceToDays

releaseDifferenceToDays <- function(release\_difference){

if(is.na(release\_difference) || release\_difference == 0 || release\_difference == -1){

return(NA)

}

else {

if(release\_difference < 0){

return(NA)

}

else{

return(round((release\_difference /60/60/24),0))

}

}

}

* + 1. getAvgPpv

getAvgPpv <- function(views, points){

if(points == 0 || views == 0){

return(NA)

}

else {

return(points/views)

}

}

* + 1. isDepleted

isDepleted <- function(saldo, points\_given){

if(saldo < 100 && points\_given > 100){

return(as.factor("1"))

}

else{

return(as.factor("0"))

}

}

* + 1. getReleaseDay

getReleaseDay <- function(date){

return(as.factor(weekdays(as.Date(date))))

}

* 1. Anexo 5: Variables re nombradas en pre-procesamiento

|  |  |  |
| --- | --- | --- |
| Tabla | Nombre original | Nombre nuevo |
| usuarios | difference\_last\_and\_first\_share | total\_activity |
| videos | X1\_week\_active\_users\_at\_release | active\_users |
| videos | X1\_week\_active\_raffles | active\_raffles |
| videos | X1\_week\_new\_users\_at\_release | new\_users |
| videos | total\_users\_at\_release | total\_users |

* 1. Anexo 6: Funciones para normalización
     1. Normalización Min-Max

minmaxNormalization <- function(sample,min,max){

if(is.na(sample)){

return(NA)

}

else{

return((sample-min)/(max-min))

}

}

* + 1. Normalización Z-Scores

zscoreNormalization <- function(sample,sd,mean){

if(is.na(sample)){

return(NA)

}

else{

return((sample-mean)/(sd))

}

}

1. Conocido en español como “publicidad por emplazamiento”, consiste en la inserción de un producto, marca o mensaje dentro de la narrativa de un programa, en este caso en particular, fotografías o videos compartidos en redes sociales. Es un tipo de publicidad sutil indirecta. – [Fuente: Wikipedia.org] [↑](#footnote-ref-1)
2. Un banner es un formato publicitario característico de internet. Consiste en incluir una pieza publicitaria dentro de un sitio web, cuyo objetivo es atraer tráfico al vínculo correspondiente al banner. Un *popup* es una ventana emergente dentro de un sitio web, que por lo general cuenta con un *banner*, formulario o alguna forma de captura de información dentro de él. Su función final suele ser la misma que el *banner* pero de una forma más agresiva. – [Fuente: Wikipedia.org] [↑](#footnote-ref-2)
3. Este fenómeno se hace presente con la masificación de los *banners*. El primer *banner* recordado de la historia de internet, tuvo una conversión del 44%, esto quiere decir que de cada 100 personas que visitaron el sitio del banner, 44 hicieron *click* sobre él. Hoy, la conversión, en el mejor de los casos, llega a 2% (Información de red de *Display de Google*). Estudios de seguimiento de ojos (*eye-tracking*) demuestran a través de mapas de calor, que los usuarios de internet no fijan su atención en los banners, reconociéndolos inconscientemente como basura. [Fuentes: Gorka Garmendia (<http://www.gorkagarmendia.com/la-ceguera-del-banner-el-truco-de-mortadelo>), Red de *Display* de Google] [↑](#footnote-ref-3)
4. Hecho, número, o texto que puede ser procesado por un computador. [↑](#footnote-ref-4)
5. Los patrones, asociaciones o relaciones entre datos pueden generar información. A diferencia de los datos, la información tiene uso, utilizad. [↑](#footnote-ref-5)
6. La información puede ser transformada en conocimiento sobre patrones históricos o modas futuras. [↑](#footnote-ref-6)
7. *Knowledge discovery in databases* [↑](#footnote-ref-7)
8. ***Cr****oss-****I****ndustry* ***S****tandard* ***P****rocess for* ***D****ata* ***M****ining*, o en español, Proceso estándar multi-industria para minería de datos. [↑](#footnote-ref-8)
9. Fase *Business Understanding phase* (Fase de entendimiento del negocio) de *CRISP-DM*. [↑](#footnote-ref-9)
10. *http://www.tableausoftware.com* [↑](#footnote-ref-10)
11. Para más información sobre tareas supervisadas y no supervisadas refiérase al Anexo B: Técnicas supervisadas y no supervisadas [↑](#footnote-ref-11)
12. Medida para cuantificar los casos en los que e antecedente se hace verdadero. Puede ser número de casos o porcentaje. [↑](#footnote-ref-12)
13. Número de casos en que, habiéndose cumplido el antecedente de la regla, se cumple el consecuente. [↑](#footnote-ref-13)
14. Traducción: "La definición intuitiva de un *outlier* sería 'una observación que se desvía tanto de las otras observaciones como para despertar sospechas de que fue generada por un mecanismo diferente'" [↑](#footnote-ref-14)
15. Dividir y conquistar. [↑](#footnote-ref-15)
16. , donde son las probabilidades a priori, la probabilidad de de *B* en la hipótesis *A* y la probabilidad a posteriori. [↑](#footnote-ref-16)
17. Para más información sobre la licencia de tipo AGPL, referirse a http://www.gnu.org/licenses/agpl-3.0.html. [↑](#footnote-ref-17)
18. *CRM*, o *Customer Relationship Management*, es el proceso que gira en torno a pulir y mejorar las relaciones con los clientes, a través de campañas de marketing específicas, servicio personalizado al cliente y gestión del equipo de ventas. [↑](#footnote-ref-18)
19. Más información sobre licencia de R: https://www.r-project.org/COPYING [↑](#footnote-ref-19)
20. Se puede encontrar información sobre este paquete y sus funciones en http://www.rdocumentation.org/packages/arules [↑](#footnote-ref-20)
21. *Left hand side*, o “lado de la mano izquierda”, hace referencia a la porción izquierda de una regla de asociación, es decir, la condición que implica a la condición de la derecha. [↑](#footnote-ref-21)
22. *Right hand side*, o “lado de la mano derecha”, hace referencia a la porción derecha de una regla de asociación, es decir, la condición implicada por la condición de la izquierda. [↑](#footnote-ref-22)