ISSN 1688-2806

Universidad de la República
Facultad de Ingeniería

Análisis del Comportamiento de los Usuarios en una Red Educativa a partir de Consultas DNS

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Alexis Javier Arriola Garcia

en cumplimiento parcial de los requerimientos para la obtención del título de Magister en Informática.

Directores de Tesis

Dr. Ing. Edudardo Grampin ..... Universidad de la República
Dr. Ing. Alberto Castro ..... Universidad de la República

Tribunal

Dr. Ing. Alvaro Martin ..... Universidad de la República
Dr. Ing. Pedro Casas ..... Austrian Institute of Technology
Dr. Ing. Luis Chiruzzo ..... Universidad de la República

Director Académico

Dr. Ing. Eduardo Grampin ..... Universidad de la República

Montevideo
domingo 18 diciembre, 2022

Análisis del Comportamiento de los Usuarios en una Red Educativa a partir de Consultas DNS, Alexis Javier Arriola Garcia.

ISSN 1688-2806

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.1).
Contiene un total de 96 páginas.
Compilada el domingo 18 diciembre, 2022.
`https://www.fing.edu.uy/inco`

Incluso la persona más pequeña puede cambiar el rumbo del futuro.

J.R.R. Tolkien

Agradecimientos

En primer lugar, quería agradecer a mi familia por el apoyo recibido a lo largo de estos años. En especial a mis padres, a quienes debo todo, que siempre creyeron en mí, que fueron los primeros que me alentaron a seguir esta carrera y me enseñaron que con esfuerzo y persistencia se pueden alcanzar las metas que nos proponemos.

También quería agradecer a mis tutores, Eduardo, quien me motivó, luego de culminar la carrera de grado, a dar un esfuerzo más y continuar mi vida académica realizando esta maestría. A Alberto, quien siempre estuvo "con la camiseta puesta" y que desde el día uno, visualizó el potencial e importancia que tendría llevar a cabo una investigación de este estilo, además siempre estuvo presente y siguiendo paso a paso hasta la finalización de esta. Por otro lado, a Germán, quien fue nuestro contacto por parte de Plan Ceibal, el cual siempre tuvo una gran predisposición a la hora de coordinar reuniones y quien nunca dejó de lado su rol de investigador, aportando soluciones cuando se presentaba algún problema, comentando los resultados intermedios que íbamos obteniendo y dando sugerencias de por donde proseguir.

Y por último, pero no menos importantes, a Plan Ceibal, quienes brindaron los datos que fueron la base y el motor principal para la realización de esta tesis. Y a PEDECIBA, que me permitió continuar mis estudios y aportar mi granito de arena a la comunidad.

A mis seres queridos...los que están y los que ya se fueron

Resumen

Este trabajo de tesis se basa en el estudio y análisis del comportamiento de los usuarios dentro de una red con servicio de Internet, teniendo como única fuente de información las consultas Domain Name System (DNS). Hasta el momento de realización de esta tesis, estos datos eran almacenados sin poder aprovechar su valor. Por este motivo, es de gran importancia explorar y analizar estos datos, que pertenecen a una red que está distribuida en todo el país y que cuenta con la peculiaridad de que es utilizada, principalmente, por estudiantes de primaria y secundaria.

Estas consultas DNS fueron recopiladas y aportadas para este trabajo por el principal Proveedor de Servicios Educativos (ESP, siglas en inglés) en Uruguay, Plan Ceibal. Plan Ceibal es un ESP que apoya un programa de computación uno a uno y ha implementado una infraestructura de Tecnología de la Información y las Comunicaciones (TIC) en todas las escuelas primarias y secundarias públicas de Uruguay. Esto ha permitido tener una red desplegada a lo largo y ancho del país con conectividad Wi-Fi, a disposición de todos los estudiantes de los distintos centros educativos, para poder acceder a distintos recursos educativos a través de una conexión estándar a Internet.

En total, se recolectaron aproximadamente más de 32 mil millones de registros DNS durante todo el año lectivo 2019, período de tiempo con el cual se trabajó. Para poder trabajar con esta dimensionalidad de datos, se tuvo que utilizar una plataforma que permita soportar el trabajo con volúmenes de datos a escala *Big Data*.

A lo largo de este estudio se emplearon distintas técnicas de *machine learning* no supervisadas (lineales y no-lineales) para poder obtener información sobre el comportamiento de los usuarios en la red. A partir de esto, fue posible visualizar algunos resultados considerables, como por ejemplo, que el comportamiento de uso de Internet por parte de los estudiantes está muy influenciado por el grupo de edad y hora del día, sin embargo, no depende de la ubicación geográfica de los usuarios.

Quedaron en evidencia las diferencias y semejanzas en cuanto al tipo de contenido consumido por parte de los usuarios. Por ejemplo, el uso de redes sociales con el de dominios de contenido educativo, o la similitud entre el uso de los servicios de *streaming* y el de las redes sociales.

Finalmente, a partir de datos obtenidos mediante una herramienta para analizar los paquetes de red desplegada en algunos centros educativos, y los registros DNS pertenecientes a esos mismos locales, se crea un modelo básico y lineal para la predicción de tráfico/consumo de ciertas aplicaciones (youtube, facebook, google, etc.) en la red.

El valor de los resultados obtenidos se puede ver, por un lado, desde la perspectiva del operador de red, ya que se puede observar qué tipos de contenidos son consumidos por los usuarios y en qué horario. Esto permite tomar decisiones mejor fundamentadas para mejorar la calidad del servicio. Por otro lado, desde la perspectiva educativa, los resultados permiten saber qué tipo de contenido educativo es consumido y por cuáles usuarios, destacando a qué público se llega a través de los programas educativos analizados.

A partir de esta investigación Plan Ceibal trabajó un nuevo enfoque desde la importancia de la recolección y análisis de los datos que provienen de las consultas dns. A su vez, les permitió impulsar su propia plataforma Big Data y nuevas investigaciones.

Tabla de contenidos

Capítulo 1

Introducción

1.1. Introducción y Motivación

Los proveedores de servicios necesitan comprender el comportamiento de los usuarios en la red para mejorar la calidad de servicio y la experiencia de usuario. En este contexto, los sistemas educativos no son la excepción. Hoy en día, el acceso a Internet se ha convertido en un recurso sumamente importante para cualquier centro educativo, ya que muchos de estos basan sus actividades en plataformas digitales de aprendizaje. También es de utilidad para tener un mejor sistema de gestión educativa.

En mayor o menor medida nuestras vidas están vinculadas con la tecnología y la educación no escapa a todo esto. Hoy en día, los niños y adolescentes tienen acceso a *smartphones*, *tablets*, *laptops*, etc. que les permite el acceso a Internet para diferentes usos [1]. Cada vez existe más la necesidad de adoptar todos estos nuevos avances en pro de una mejora educacional para impulsar y motivar más la participación de los estudiantes.

El aumento del uso de la tecnología en la educación genera un gran volumen de datos, que son de mucho interés para la docencia y la investigación, ya que a partir de su análisis es posible desarrollar métodos para la extracción de información interpretable, útil y novedosa, que puede conducir a una mejor comprensión de los estudiantes en sí y los entornos en los que desarrollan su aprendizaje.

Todo este nuevo paradigma educacional basado en el acceso a Internet requiere un despliegue, soporte y mantenimiento de una infraestructura específicamente orientada para soportar las necesidades de los usuarios. Esta responsabilidad generalmente es asumida por los Proveedores de Servicios Educacionales (ESPs en inglés). Estas organizaciones ayudan a los sistemas educativos a implementar reformas integrales hacia la digitalización, brindando a estudiantes y profesores un acceso confiable y de alta calidad a la red. De esta manera, se evitan problemas de accesibilidad a recursos que podrían impactar de manera negativa en la calidad de la enseñanza.

Este caso de uso educacional plantea nuevos desafíos para los servicios de red, tanto desde el punto de vista de los contenidos y aplicaciones, como de los usuarios, mayoritariamente docentes y estudiantes (niños y adolescentes). Por un lado, existen plataformas educativas y contenido educativo a los que se accede en conjunto con otras aplicaciones y sitios web de común acceso (ej. redes sociales, motores de búsqueda y *streaming*). Por otro lado, se pueden identificar prioridades para las necesidades de aprendizaje de diferentes grupos de estudiantes, así como maximizar y priorizar el acceso a recursos de cada centro educativo, e implementar métricas para la evaluación de planes de estudios y/o programas educacionales puntuales. En consecuencia, es muy

importante entender este escenario novedoso, con características bastante singulares que no son frecuentes en otros entornos.

Este trabajo es una introducción a todo este nuevo mundo, necesaria para la mejor comprensión de las relaciones entre las distintas dimensiones, que se refieren al entendimiento del comportamiento de los usuarios en este tipo de redes, buscando una mejor comprensión de este tema. Los resultados son de interés para gran parte de la población, desde padres hasta investigadores, pasando por responsables de políticas educativas, así como también de políticas de accesos a Internet.

Parte de los resultados presentados en este trabajo fueron publicados en la conferencia ICCSA 2020 [2]. Más precisamente, el estudio que refiere solamente a las categorías con métodos lineales.

1.2. Objetivos

A partir de las perspectivas e idea general establecida en la sección anterior, el objetivo general planteado fue el de hacer foco en tratar de comprender el comportamiento de los usuarios en las redes educativas a través del estudio de un gran volumen de datos del Sistema de Nombres de Dominio (DNS) recopilados durante todo el año lectivo 2019 de un importante ESP a nivel nacional, Plan Ceibal. Para poder abordar este objetivo bastante genérico se plantearon una serie de objetivos más específicos.

Primero, se planteó tener una idea general del consumo de contenidos por parte de los usuarios, esto es saber por ejemplo, si se está consumiendo *streaming* por un determinado grupo de usuarios y en qué momento del día sucede esto. En otras palabras, conocer de forma genérica lo que hacen los usuarios cuando se conectan a la red.

Luego, agregando más granularidad al análisis anterior y como objetivo siguiente, se propone la idea de analizar lo que sucede dentro de algunas de esas categorías de contenido que puedan ser de interés. Con este objetivo se da un paso más en el sentido de conocer específicamente qué tipo de dominios son los más consumido, por quiénes son consumidos, en qué momento del día sucede y cómo se relacionan con otro tipo de contenidos.

Por último, se tuvo el objetivo de hacer una incursión en la utilización de modelos predictivos, con la intención de poder generar una noción primaria e inicial de cómo se podrían utilizar las consultas DNS para la predicción de tráfico en la red y qué resultados se obtiene de esto.

1.3. Plan Ceibal

En 2005, durante el World Economic Forum, Nicholas Negroponte, perteneciente al MIT, presenta un proyecto novedoso, conocido como One Laptop per Child (OLPC) [3] que se basa en la entrega de laptops de bajo costo para reducir la brecha digital que existe en países en desarrollo. En 2007 [4], Uruguay llevó a cabo esta idea a nivel nacional con el comienzo del Plan Ceibal, cuya misión inicial era promover la inclusión digital, "Un plan de inclusión e igualdad de oportunidades para apoyar las políticas educativas uruguayas con tecnología", como se expresa en su sitio web [5].

Para poder lograr esto, se tuvo como principales objetivos el poder entregar laptops, de bajo costo, para todos los estudiantes y docentes de la enseñanza y dar acceso a internet (Wi-Fi) a cada centro educativo público. Estos importantes objetivos fueron cumplidos, en todo el país, en los primeros tres años de comenzado el programa. Hoy en día, más de diez años después, Plan Ceibal ha diversificado mucho sus actividades en colaboración con el sistema educativo. Actualmente, se tiene en funcionamiento varias plataformas educativas, como un sistema de tutoría inteligente (ITS) para el

aprendizaje de las matemáticas, además se cuenta con el soporte para recursos educativos, como una biblioteca digital y un sistema de gestión de contenido (CMS). Esto es algo que va en aumento, ya que a medida que avanza el uso de tecnología en el sistema educativo, provoca una mayor demanda de apoyo técnico por parte del Plan Ceibal.

No obstante, como ocurrió en sus comienzos y hasta la actualidad, una de las responsabilidades más relevantes de Plan Ceibal es la de poder brindar conectividad Wi-Fi para todos los estudiantes de los centros educativos a lo largo del país. Para poder lograr esto, cada centro educativo cuenta con una simple, pero eficaz arquitectura de red, la cual consiste en un router con acceso a Internet mediante fibra óptica desde el ISP local, y a través de este router se tiene varios puntos de acceso que posibilitan una conectividad Wi-Fi. La conectividad del Plan Ceibal está desplegada en más de 1500 centros educativos, que comprenden más de 8000 puntos de acceso Wi-Fi, a lo largo de todo el país; esto convierte al Plan Ceibal en uno de los proveedores de Internet inalámbrico más grande a nivel nacional, llegando a un número de dispositivos comparable a los suscriptores de los operadores de redes móviles nacionales.

Hace ya un par de años, Plan Ceibal cuenta con una novedosa solución de DNS basada en la nube, Cisco Umbrella [6], la cual también sirve para fines de seguridad y filtrado de contenido. Otra de las ventajas que tiene este servicio DNS, es que permite acceder de forma simple a todos los registros de las consultas DNS que han sido procesadas. Para esto, se puede configurar el logging del sistema para que los registros se guarden en un bucket de Amazon S3, donde los registros se almacenan durante un período máximo de 30 días. Gracias a esto, es posible automatizar su descarga y almacenarlos de manera permanente, fuera del sistema Umbrella. Por lo tanto, el acceso centralizado a los datos de consultas de DNS de toda la red es bastante sencillo. Una vez recopilados, pueden analizarse más a fondo e integrarse con información de otras fuentes en una plataforma de análisis de datos especializada.

1.4. Trabajos Relacionados

A continuación se describen brevemente algunos trabajos sobre análisis de datos de DNS, si bien existen y hay una gran cantidad de publicaciones científicas al respecto, estas en su mayoría están orientadas hacia la seguridad en las redes [7]. Muchos de los problemas que estos atacan se basan en la detección de actividades maliciosas como *botnet*, *web-spam* y URL maliciosas [8].

Esta tesis no está orientada a la seguridad, sino que busca analizar el comportamiento observado y reconocer patrones relevantes, mediante técnicas no supervisadas de aprendizaje automático. El objetivo es comprender las características típicas y las principales tendencias de las consultas DNS en un contexto educativo en particular. Cabe recalcar que estamos trabajando con datos a nivel nacional y que comprende la mayor parte del sistema educativo del país, por lo que, hasta donde sabemos y dado este escenario, este puede ser unos de los primeros trabajos con estas características.

1.4.1. Descubrimiento de patrones a partir de consultas DNS

En el artículo presentado por Ruan et al. [9], se hace referencia al descubrimiento de patrones en los datos de tráfico de consultas DNS con la finalidad de detectar anomalías e identificar patrones de comportamiento de los usuarios. Debido al gran volumen de datos que se genera con los registros DNS, se adoptan enfoques de minería de datos que pueden descubrir patrones dinámicamente.

En primer lugar, se propone un novedoso problema: la minería de tendencias periódicas. Luego, se utiliza otro método de predicción para el volumen de tráfico y detectar anomalías. Se asume que, si un patrón ocurre con frecuencia en la historia

reciente, puede ocurrir repetidamente con alta probabilidad a menos que ocurra un evento anormal. Dado que los patrones con intervalo de tiempo tienden a ser fortuitos y poco fiables, solo interesa detectar los patrones sin intervalo. Luego, se predice el volumen de tráfico en el momento siguiente, haciendo referencia a los patrones de tráfico de consultas recientes que se han producido periódicamente. En el capítulo 3.2 del citado artículo se presenta un pseudocódigo para el algoritmo de minería de tendencias periódicas (*Periodic trend mining*).

En segundo lugar, se dividen los datos de la serie temporal del tráfico de consultas DNS en *cluster* separados mediante el uso del algoritmo de clustering *Shared-Nearest-Neighbour* (SNN) . Antes de realizar la ejecución del algoritmo se realiza un preprocesamiento de los datos, eligiendo para el estudio una lista de los 676 dominios más populares, y se cuentan las cantidades de visitas que tienen por día y por hora. Como resultado, luego de ejecutado el algoritmo SNN, se obtuvo 7 *clusters*, en donde existen diferencias entre las tendencias del tráfico en cada *cluster*, lo que indica que los comportamientos de los usuarios que visitan los nombres de dominio en diferentes *clusters* tienen características bastante diferentes.

En semejanza con el procedimiento adoptado por Ruan et al., en este trabajo de tesis también se procede a hacer un preprocesamiento de las consultas DNS generando agregaciones, obteniendo como menor granularidad , la cantidad de consultas por hora por día. Además, se utiliza la idea de clustering con la misma finalidad que Ruan et al, poder visualizar diferencias entre los distintos *clusters*, que pueda dar un indicio del comportamiento de los usuarios en la red.

1.4.2. Método SVM mejorado aplicado al análisis del comportamiento de usuarios en línea.

En la publicación [10], se propone una mejora del método supervisado *Support vector machine* (SVM) para analizar el comportamiento en línea del usuario, basado en los registros DNS provenientes de más de 45000 usuarios pertenecientes a un campus universitario.

Primero, presentan un diseño para un método de etiquetado de dominios, en donde a cada dominio al que accede un usuario se le asigna una etiqueta de interés que indica un comportamiento de este usuario. Para esto, se genera un conjunto de palabras claves predefinidas para categorizar los intereses del usuario en varios grupos. Luego, el texto que se extrae de los dominios se utiliza para realizar el etiquetado de categorías. En la Tabla 1.1 se muestra un ejemplo de palabras clave relacionadas con las etiquetas de interés. Como ya se mencionó en el apartado de objetivos, la idea de realizar un análisis por categorías, en donde estas categorías representan ese etiquetado de dominios.

Luego, ahondando en lo que refiere al método SVM, se emplea el algoritmo *Sequential Minimization Optimization* (SMO) como método típico de descomposición, en el cual la selección del conjunto de trabajo es una tarea clave. Sin embargo, la estrategia de selección del conjunto de trabajo del algoritmo SMO convencional, así como las estrategias de selección heurística y aleatoria, aumentan el tiempo de aprendizaje de SVM incluso si no coinciden completamente con la condición Karush–Kuhn–Tucker (KKT). En el capítulo 3 del citado estudio se propone y se implementa una solución para el problema antes mencionado. Esta solución se basa en mejorar la selección del conjunto de trabajo mediante la reescritura de las condiciones de KKT.

Al comparar el algoritmo tradicional de SVM contra el nuevo algoritmo mejorado, usando el mismo conjunto de datos, se aprecia que con el método mejorado de SVM se reduce la tasa de error, se mejora la tasa de precisión y se reduce el tiempo de ejecución. Como resultado se obtiene un modelo que mejora en gran medida la eficiencia computacional y la precisión en el análisis del comportamiento del usuario en la red,

Tabla 1.1: Ejemplo de palabras claves - comportamiento

Comportamiento	**Palabras claves**
Noticias	noticias, revista, diario
Email	mail, hotmail, gmail
Música	mp3, audio, music, spotify
Deportes	basketball, fútbol, rugby, golf
...	...

que puede ser utilizado para grandes volúmenes de datos.

1.4.3. Otros estudios de interés

En lo que refiere al análisis de comportamiento con registros DNS, existen varios trabajos previos a destacar. Plonka et al. [11], describen una metodología de agrupación en *cluster* sensible al contexto, que se aplica a las respuestas de consulta de DNS para generar los agregados deseados, para esto se realiza una preclasificación de los registros DNS en tres clases de consultas, canónico (comportamientos previstos por RFC), sobrecargado (servicios bloqueados) y no deseado (consultas que nunca tendrán éxito).

En [12], se presenta el análisis de un gran conjunto de datos de medidas de desempeño del DNS. Los datos de 600 resolutores de DNS recursivos diferentes, que se distribuyen globalmente, se utilizan para compararlos y descubrir las diferencias y similitudes entre ellos (por ejemplo, la tasa de éxito de la consulta y las causas de los fallos de búsqueda).

El trabajo efectuado por Schomp et al. [13], se centra en poder caracterizar los clientes de una red a través de las consultas DNS, en busca de un objetivo a futuro, que es el de poder implementar un modelo analítico de interacción cliente-DNS. Siguiendo la misma línea de trabajo, de poder comprender y perfilar lo que las personas están haciendo a partir del tráfico de DNS, se tiene la investigación de Li et al. [14], en donde se plantean una variedad de métodos, entre ellos, un clasificador multi escala empleando *Random Forest*, con la finalidad de distinguir las actividades de interés de los usuarios.

El trabajo de Jia et al. [15] hace foco en desarrollar un modelo de comportamiento de acceso para cada usuario dentro de una red perteneciente a un campus, utilizando las consultas DNS. El objetivo final es poder determinar si el acceso del usuario es seguro o no, de esta forma se prevé la propagación de código malicioso, en este caso también se está trabajando con los registros DNS orientado a la seguridad en redes, como se mencionó al principio.

Capítulo 2

Recopilación y análisis de los datos

El objetivo principal de este capítulo es presentar el trabajo de calidad de datos que se realizó previamente, dando como resultado el dominio de datos utilizado para todo el estudio de tesis. En primer lugar, se detalla la forma de obtención y la fuente de los datos que luego serán usados durante todo el proyecto. También se muestran estadísticas generales que se obtuvieron, como por ejemplo la cantidad de consultas DNS por departamento, por día de la semana, por hora, etc.

Además, se presenta la infraestructura y el ambiente de trabajo que se utilizarán a lo largo de la ejecución de este proyecto. Así como también, un preprocesamiento que se realizó a todos los registros DNS que se tenían, descartando los datos corruptos y agregando información complementaria a cada registro.

2.1. Recopilación de datos

Los datos utilizados en este estudio se obtuvieron de la infraestructura de red de Plan Ceibal. Esta red, con 8,587 puntos de acceso Wi-Fi (AP) ubicados en 1,878 edificios educativos (que cubren más del 95 % de los estudiantes de en Uruguay), es una de las redes de comunicación más grandes del país. De esos 1,878 edificios educativos, el 70 % son centros de educación primaria (escuelas públicas) y el 30 % son centros de educación secundaria (liceos), ver figura 2.1.

Además de los datos obtenidos de la red, se tiene información (brindada por Plan Ceibal) que refiere más al aspecto geoespacial de los centros educativos que pertenecen a la red. Esta información geoespacial permite conocer el departamento, ubicación exacta (latitud y longitud), área (rural/urbana), el tipo de centro educativo (colegio, bachillerato u otro) y el quintil socioeconómico.

El quintil socioeconómico [16] es una clasificación en donde se registra la división del total de escuelas públicas en cinco grupos de igual cantidad, de modo que el quintil 1 agrupa al 20 % de las escuelas de contexto más vulnerable y el quintil 5 al 20 % de las de contexto menos vulnerable. Esta clasificación se hace por separado para el conjunto de escuelas urbanas por un lado y para las rurales por otro, considerando distintas dimensiones.

Cada registro en el conjunto de datos estudiados corresponde a una solicitud de consulta DNS. Plan Ceibal, como parte de su infraestructura de red, ha implementado el sistema Cisco Umbrella [6] como su solución de DNS. Es decir, todas las solicitudes de consulta DNS realizadas por los usuarios conectados a la red de Plan Ceibal son registradas y clasificadas por el sistema Cisco Umbrella.

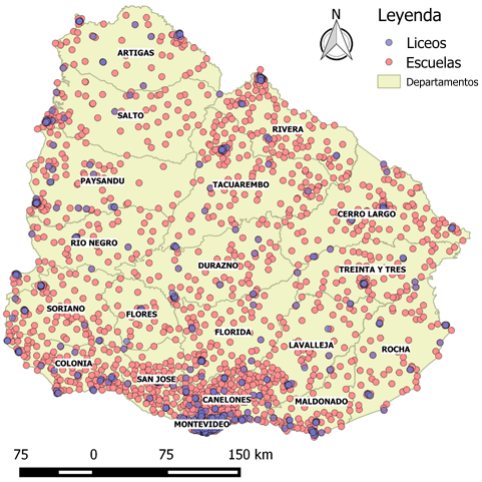


Figura 2.1: Edificios educativos en Uruguay

Hay dos tipos de registros: registros DNS y registros *proxy*. El primero corresponde a las consultas DNS estándar y el segundo a consultas dudosas que se envían a un *proxy* para su posterior inspección. Para este trabajo, solo se consideraron los registros DNS. Para cada consulta que queda registrada [17], se almacenan ciertos datos de interés:

- **Timestamp** - Indica el momento en que se realizó la consulta DNS.
- **InternalIp** - La dirección IP del dispositivo que realiza la consulta, en este caso coincide con la ip del router.
- **ExternalIp** - La dirección IP del router que recibe la consulta.
- **Action** - Campo que indica si la consulta DNS fue permitida o bloqueada.
- **Identities** - Todas las identidades asociadas con la consulta.
- **Most Granular Identity** - La primera identidad correspondiente con la consulta en orden de granularidad.
- **Query Type** - Campo que indica si la consulta DNS fue permitida o rechazada.
- **Response Code** - Campo que indica el código de retorno de DNS para la consulta.
- **Domain** - El dominio que se solicitó.
- **Categories** - Las categorías de seguridad o contenido con las que coincide el destino.

Un ejemplo de un registro de registro DNS es el siguiente:

"2019-03-10 17:48:41", "10.10.1.100", "24.123.132.133", "Permitido", "1202232-Uv1.0-80-umbrella,1202232", "1202232-Uv1.0-80-umbrella", "1 (A)", "NOERROR", "instagram.com", "shared photo, social network"

Dado que cada AP de Wi-Fi tiene una IP única dentro de la red del Plan Ceibal y que se tiene registrado dónde se encuentra cada uno de ellos, utilizamos el campo de registro DNS *ExternalIp* para unir cada consulta DNS con un edificio educativo. De cada edificio hay información sobre el departamento donde se encuentra, el área (rural/urbana) y qué tipo de centro educativo es (escuela, liceo u otros).

En esta tesis, se analizan **32.777.748.989** registros de consultas DNS recopilados durante todo el año 2019. Cabe mencionar que, como Uruguay es un país que se encuentra en el hemisferio sur, el año académico se alinea con el año calendario; de marzo a diciembre.

Es importante resaltar que este estudio fue aprobado por el comité de ética y privacidad de datos de Plan Ceibal. Todos los conjuntos de datos se identifican y manejan de acuerdo con la legislación uruguaya de protección de la privacidad.

2.2. Infraestructura

Para trabajar con el volumen completo de datos, se adopta Hortonworks Data Platform (HDP) [18], utilizando una implementación de nodo único con 40 CPU Intel Xeon y 256 GB de RAM. HDP, cuyo núcleo se basa en Apache Hadoop [19], es un marco de código abierto para almacenamiento distribuido, capaz de procesar grandes conjuntos de datos de múltiples fuentes. Se realizó un trabajo de investigación previo para poder usar la plataforma, y a su vez poder realizar una instalación local. En el apéndice A se detalla la arquitectura de la plataforma, así como algunos de sus principales componentes.

Se implementaron consultas SQL específicas en Apache Spark [20]. Estas se ejecutaron en Apache Hive [21] para poder procesar todo el conjunto de datos y construir las matrices de entrada para los algoritmos. Apache Hive es un almacén de datos que permite un resumen de datos sencillo y consultas *ad hoc* a través de una interfaz similar a SQL, para grandes conjuntos de datos almacenados en un sistema de archivos distribuido, HDFS. Apache Spark es un *framework* de programación para el procesamiento de datos distribuidos. La configuración de parámetros para poder llevar adelante la ejecución de una aplicación Spark es una tarea que depende mucho del tipo de aplicación que se quiere ejecutar, por lo que pueden existir distintas configuraciones. En el apéndice B, se puede encontrar una guía para poder realizar una configuración básica y solventar los problemas que puedan surgir.

El código desarrollado durante esta tesis puede ser encontrado en el siguiente link [22]. Dentro de la carpeta *ZeppelinNotes* se encuentran todos los notebooks empleados para trabajar con los datos en HDP y hacer las agregaciones correspondientes para poder ser utilizados como entrada para los algoritmos de aprendizaje. En la carpeta *SRC* se podrán observar los principales códigos desarrollados para obtener los resultados utilizando los archivos CSV. Los algoritmos auxiliares se encuentran dentro de la carpeta *Algoritmos*. Para cada método de aprendizaje dentro de esta carpeta, se implementaron funciones para facilitar su uso y mostrar los resultados.

2.3. Pre-procesamiento de los datos

Los datos se preprocesaron para obtener registros de mayor calidad, que luego serán utilizados como datos de entrada para los algoritmos de aprendizaje. Como primer paso, eliminamos aquellos registros que estaban corruptos (es decir, falta de datos o datos ilegibles en algunos de los campos). Estos representaron el 0,001 % del total.

A partir de los datos de la marca de tiempo (en formato yyyy-mm-dd hh:mm:ss), se crearon nuevos campos. La fecha se dividió en *year*, *month*, *day*, *dayofweek* (el lunes se denota con 0 y el domingo con 6), y la hora se dividió en *hour*, *minute* y *second*. De esta forma, se facilita un manejo más granular de los datos, dando la posibilidad de seleccionar qué tipo de granularidad se quiere para hacer las agregaciones, por ejemplo, por hora, por semana, por día de la semana, etc. También se agregó la columna *date*

(en formato yyyy-mm-dd), en caso de que sea necesario y brinde una ayuda en etapas posteriores del procesamiento. Por otro lado, como los datos se encontraban en formato horario universal, se realizó un corrimiento de -3 horas para equipararlos a la hora uruguaya. En la figura 2.2, se muestra un ejemplo de los nuevos campos que se crean.

Figura 2.2: Ejemplo del desglose del campo *timestamp*.

Como se especificó, al principio de este capítulo, en la descripción de los datos que se almacenan en cada registro DNS, los campos "Most Granular Identity", "Identities" y "ExternalIp" aportan información sobre el centro donde se realizó cada consulta, la información más importante es que se puede obtener el identificador dado a ese centro, por esto se decidió extraer esa información y usar solo ese dato como reemplazo de las columnas mencionadas, creando una nueva columna *ruee* (Figura 2.3) para identificar cada centro educativo.

Figura 2.3: Ejemplo de nuevo campo identificador que se crea.

Como el identificador del centro educativo ahora está registrado en los datos del DNS, se puede agregar información geoespacial (que está disponible) sobre los centros desde este identificador. Esta información permite agregar una dimensión espacial a cada registro, un ejemplo de estos datos se muestran en la figura 2.4.

Figura 2.4: Ejemplo de datos que se agregan a partir de identificador.

En cuanto al campo *domain*, para una mejor comprensión y lectura de los dominios, se corrigen mediante expresiones regulares y, además, se analizan según Public Suffix List (PLS) [23].

Como se mencionó anteriormente, el sistema Cisco Umbrella asigna a cada consulta DNS una o más categorías [24] que describen el contenido del dominio solicitado. Al existir la posibilidad de que se tenga más de una categoría por consulta, se hace un preprocesamiento de los datos para poder discernir cúal es la categoría que identifica a cada consulta DNS mejor, para esto se decide tomar la categoría más representativa, es decir, la categoría con la cual más veces se categorizó al dominio de la consulta, generando tres nuevas columnas: *category_1*, *category_2* y *category_3* (ver ejemplo en

Figura 2.5), con el fin de distinguir las tres categorías más representativas. En caso de que no exista categoría se asigna por defecto *sin_categoria*. En todo el transcurso de este proyecto, solo se utiliza la categoría más representativa, en otras palabras, cuando nos referimos a categoría, nos estamos refiriendo a la columna *category_1*.

parsed_domian	categories
google	[search_engine]
twitter	[social_networking, blogs]
local	

⇩

parsed_domian	categories	category_1	category_2	categoriy_3
google	[search_engine]	search_engine	software_or_technology	application
twitter	[social_networking, blogs]	social_networking	blogs	news_media
local		sin_categoria	sin_categoria	sin_categoria

Figura 2.5: Ejemplo de procesamiento de categorías.

2.4. Análisis de los datos

Como se muestra en la Figura 2.6, aproximadamente el 42 % y el 50 % de las consultas de DNS corresponden a escuelas y liceos, respectivamente; mientras que menos del 8 % corresponde a otros centros educativos (por ejemplo, escuelas técnicas, centros de formación docente). Para este estudio, solo consideramos los registros de los primeros dos grandes grupos.

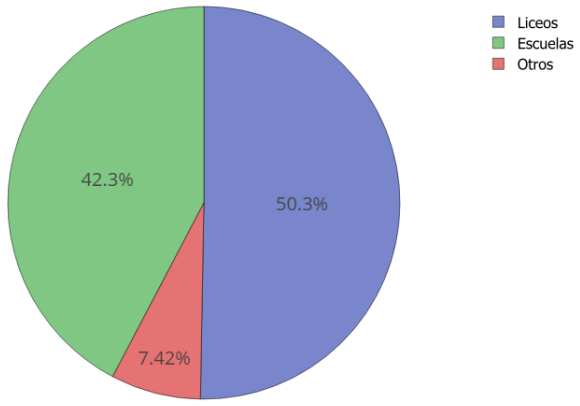


Figura 2.6: Porcentaje del total de consultas DNS solicitadas por categoría de edificio educativo.

En la Figura 2.7, se informa una representación de la cantidad total de consultas DNS agrupadas por departamentos. Es posible ver que solamente Montevideo contiene casi el 26 % de los registros y Canelones casi el 17 %, lo cual era lo que se esperaba, ya que esos dos departamentos representan el 55 % de la población uruguaya total.

La distribución de las consultas en el resto de los departamentos es más equilibrada, los departamentos con menor población poseen menos consultas, pero no se tiene esa diferencia de aproximadamente un 20 % que se tiene con Montevideo.

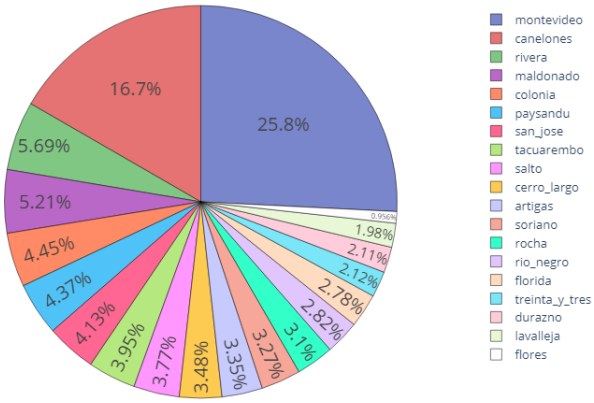


Figura 2.7: Representación de la cantidad total de consultas agrupadas por departamento.

En la Figura 2.8, se muestra la distribución temporal de las consultas durante la semana. En particular, cómo fue el comportamiento, en lo que respecta a cantidad de consultas, de los meses del año para cada día de la semana (los valores están promediados entre la cantidad de días de la semana para cada mes). Se puede observar claramente que los días con la actividad más significativa corresponden precisamente a los días de actividad estudiantil (de lunes a viernes).

Para ver un poco más en detalle el comportamiento temporal de las consultas, se emplea la Figura 2.9, en donde es posible ver el flujo de cantidad de consultas por mes durante las horas del día. Para todos los meses se tiene un comportamiento bastante similar, en donde a partir de las 8:00 hs, se tiene un crecimiento muy significativo de las consultas DNS. También es notorio el decrecimiento de la actividad estudiantil a la hora 12:00, debido a que en esta hora se produce el cambio de turno, generalmente se tiene un turno matutino (de 8:00 hs a 12:00 hs) y uno vespertino (de 13:00 hs a 17:00 hs). En la mayoría de los meses hay un creciente de las consultas en las primeras horas de la tarde y a partir de la hora 15:00. empieza un descenso de la actividad hasta la finalización del día.

Teniendo en cuenta el comportamiento visto, se decidió hacer un “corte” temporal en el set de datos. Dado que unos de los principales objetivos de este proyecto es el de analizar el comportamiento de los usuarios en la red Ceibal, y como se vió a principios de este capítulo, casi un 93 % de las consultas pertenecen a centros educativos de primaria y secundaria, los cuales tienen horarios de funcionamiento fácilmente identificables, se decide hacer un corte temporal, en el cual se toman los datos comprendidos de lunes a viernes, entre las 08:00 y las 17:00 horas, cubriendo el horario de principal funcionamiento de los centros mencionado.

Otro punto a tener en cuenta es que se filtraron los datos para utilizar solamente los días lectivos de los centros educativos. Estos días son diferentes dependiendo del

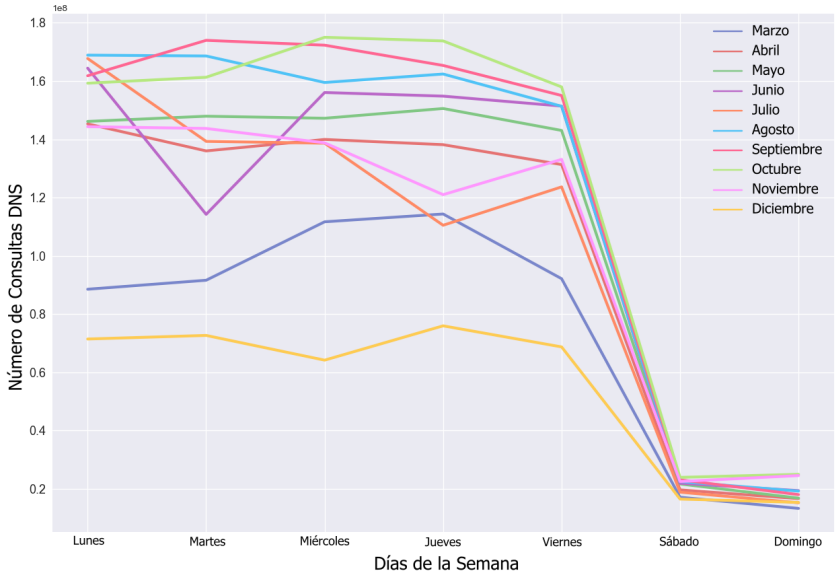


Figura 2.8: Cantidad de consultas totales, en cada mes, por día de la semana.

tipo de centro, ya que los días de vacaciones no son los mismos para los centros de primaria que para los de secundaria. Para los centros de primaria se tuvo una semana (del 1 al 5 de julio) de vacaciones de invierno, mientras que en secundaria hubo dos semanas (del 1 al 12 de julio). Para ambos casos se tuvo una semana de vacaciones de primavera (del 16 al 20 de septiembre) y una semana de vacaciones por semana de turismo (del 15 al 19 de abril). También para ambos tipos de centros se filtraron los feriados nacionales que fueron entre semana, 22 de abril, 1 de mayo, 19 de junio y 18 de julio.

2.5. Conclusiones

En este capítulo se muestra todo el trabajo previo al análisis de los datos, desde su recopilación hasta los procesos de calidad de datos y agregaciones de datos que se realizaron para obtener las matrices finales que se usarán a lo largo de esta tesis. Se introduce la plataforma HDP, que permitió el procesamiento de las 32 mill millones de consultas DNS, y las distintas herramientas (software) que se utilizaron.

Como punto importante se tiene el relacionamiento que se pudo realizar entre los datos de las consultas DNS y los meta datos de cada centro educativo, por medio del identificador del router que aparece en las consultas DNS. Esto permitió tener una información más enriquecida de la consulta, al poder darle una ubicación geoespacial.

Se observo que más de del 90 % del total de consultas DNS pertenecen a escuelas y liceos, grupos seleccionados para este estudio. Se observaron algunos patrones temporales de los datos, por ejemplo el casi nulo uso de la red los fines de semana o en horarios nocturnos. Esto permitió acotar más el universo de los datos que se empleará, centrando los análisis en los horarios y días de mayor actividad de la red.

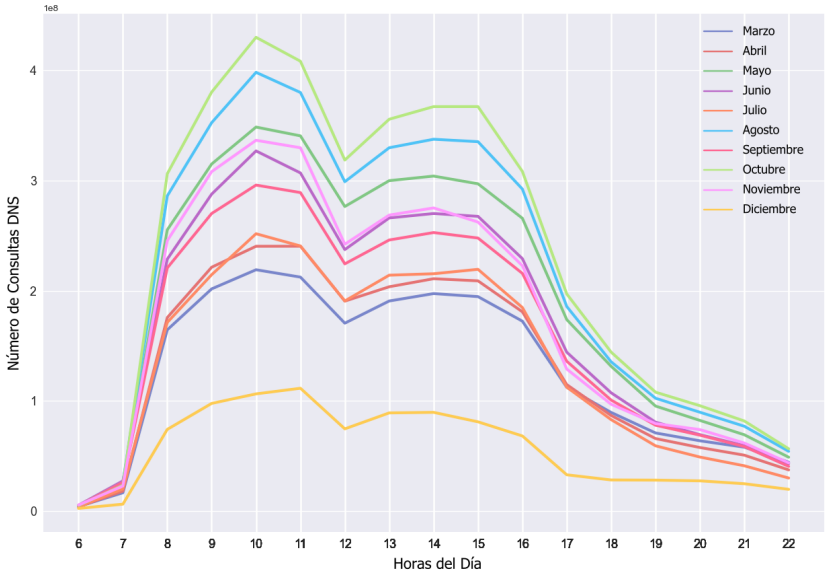


Figura 2.9: Cantidad de consultas totales, en cada mes, por hora del día.

Capítulo 3

Métodos y herramientas

Para el estudio analítico de los datos y poder comprender el comportamiento de los usuarios en Internet, se utilizaron principalmente dos familias de algoritmos distintas: i) algoritmos lineales, en particular, análisis de componentes principales (PCA) y análisis de clusters (CA) utilizando *k-means* y clusters jerárquicos; ii) algoritmos no lineales como *t-Distributed Stochastic Neighbor Embedding* (t-SNE) y *Self-Organizing Map* (SOM).

Ambos grupos de técnicas pertenecen a la familia de métodos no supervisados, lo que significa que no se utiliza información sobre otras variables de respuesta o la pertenencia a grupos para obtener los resultados. Esto hace que estos métodos sean apropiados para el análisis exploratorio de datos, donde el objetivo es la generación de hipótesis más que la verificación de hipótesis [25].

Por último, se presentan los modelos de *Random Forest* y *Linear Regression*, que serán empleados para trabajar posteriormente en el Capítulo 6 como modelos predictores del tráfico en la red. La idea es tener una idea inicial sobre este tema y ver cómo se comporta un modelo inicial y simple, como el modelo lineal, en donde se puede obtener una formulación matemática para determinar la predicción de tráfico, y compararla con otro modelo que utiliza árboles de decisión combinados con la técnica de *bagging*, como Random Forest.

Todos los algoritmos de aprendizaje utilizados se programaron en Python 3.7. Para ejecutar PCA (Principal Component Analisys), Silhouette y CA (Cluster Analisys), usamos la biblioteca *scikit-learn* [26]; mientras, adoptamos la biblioteca *seaborn* [27] para mapas de calor, y la biblioteca *MiniSom* [28] para ejecutar el algoritmo SOM (Self-Organizing Map).

3.1. Principal Component Analysis (PCA)

Cada registro del conjunto de datos que se utilizó para este proyecto posee una gran cantidad de variables que pueden estar relacionadas o no, entre sí. La utilización de PCA no solo permite entender cómo se relacionan estas variables entre sí, sino que también permite realizar una visualización gráfica de los datos en dos dimensiones, aportando un mayor entendimiento de los datos; esto no sería posible si se trabajase con todas las dimensiones del dominio de datos.

La idea central del análisis de componentes principales (Principal Component Analysis en inglés) es reducir la dimensionalidad de un conjunto de datos que contiene una gran cantidad de variables interrelacionadas, al tiempo que conserva la mayor can-

tidad posible de la variabilidad presente en el conjunto de datos. Esto se logra mediante la transformación a un nuevo conjunto de variables, las componentes principales (PC), que no están correlacionados, y que están ordenados para que los primeros retengan la mayor parte de la variación presente en todas las variables originales [29, 30].

La mayor parte de la variación se explica en los primeros PC [31]. Cada objeto se identifica por una puntuación y cada variable por un valor de carga o ponderación. En su representación gráfica (biplot PCA), los vectores que representan variables que forman un ángulo agudo se consideran atributos correlacionados, mientras que los que son perpendiculares se consideran no correlacionados.

Por otro lado, "conservar la mayor variabilidad posible" se transforma en encontrar nuevas variables que sean funciones lineales de las del conjunto de datos original, que maximicen sucesivamente la varianza y que no estén correlacionadas entre sí. Encontrar tales nuevas variables (los PC) se reduce a resolver un problema de valor propio/vector propio (eigenvalues y eigenvectors en inglés).

3.1.1. Vectores Propios

Los vectores propios son un caso particular de multiplicación entre una matriz y un vector [29]. Los vectores propios de una matriz son todos aquellos vectores que, al multiplicarlos por dicha matriz, resultan en el mismo vector o en un múltiplo entero del mismo. Los vectores propios tienen una serie de propiedades matemáticas específicas:

- Los vectores propios solo existen para matrices cuadradas y no para todas. En el caso de que una matriz n x n tenga vectores propios, el número de ellos es n.
- Si se escala un vector propio antes de multiplicarlo por la matriz, se obtiene un múltiplo del mismo vector propio. Esto se debe a que, si se escala un vector multiplicándolo por cierta cantidad, lo único que se consigue es cambiar su longitud, pero la dirección es la misma.
- Todos los vectores propios de una matriz son perpendiculares (ortogonales) entre ellos, independientemente de las dimensiones que tengan.

3.1.2. Valores Propios

Cuando se multiplica una matriz por alguno de sus vectores propios, se obtiene un múltiplo del vector original, es decir, el resultado es ese mismo vector multiplicado por un número. Al valor por el que se multiplica el vector propio resultante se le conoce como valor propio [29]. A todo vector propio le corresponde un valor propio y viceversa.

En el método PCA, cada una de las componentes se corresponde con un vector propio y el orden de componente se establece por orden decreciente de valor propio. Así, la primera componente es el vector propio con el valor propio asociado más alto.

3.1.3. Matriz de covarianza

Una matriz de varianzas-covarianzas es una matriz cuadrada que contiene las varianzas y covarianzas asociadas con diferentes variables. Los elementos de la diagonal de la matriz contienen las varianzas de las variables, mientras que los elementos que se encuentran fuera de la diagonal contienen las covarianzas entre todos los pares posibles de variables.

La covarianza es un valor que indica el grado de variación conjunta de dos variables aleatorias respecto a sus medias. Es el dato básico para determinar si existe una dependencia entre ambas variables.

3.1.4. Cálculo de las componentes principales (PC)

Cada PC (Z_i) se obtiene por combinación lineal de las variables originales [32]. Se pueden entender como nuevas variables obtenidas al combinar de una determinada forma las variables originales. La primera PC de un grupo de variables $(X_1, X_2, \ldots, X_p)$ es la combinación lineal normalizada de dichas variables que tiene mayor varianza:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p \tag{3.1}$$

Donde los términos $(\phi_{11}, \ldots, \phi_{1p})$ reciben el nombre de *loadings* y son los que definen a la componente. ϕ_{11} es el loading de la variable X_1 de la primera PC. Los *loadings* pueden interpretarse como el peso/importancia que tiene cada variable en cada componente y, por lo tanto, ayudan a conocer qué tipo de información recoge cada una de las componentes. Que la combinación lineal sea normalizada implica que:

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1 \tag{3.2}$$

Dado un set de datos X con n observaciones y p variables, el proceso a seguir para calcular la primera PC es:

- Centralización de las variables: se resta a cada valor la media de la variable a la que pertenece. Con esto se consigue que todas las variables tengan media cero.
- Se resuelve un problema de optimización para encontrar el valor de los *loadings* con los que se maximiza la varianza. Una forma de resolver esta optimización es mediante el cálculo de eigenvector/eigenvalue de la matriz de covarianza.

Una vez calculada la primera componente (Z_1) se calcula la segunda (Z_2) repitiendo el mismo proceso, pero añadiendo la condición de que la combinación lineal no pude estar correlacionada con la primera componente. Esto equivale a decir que Z_1 y Z_2 tienen que ser perpendiculares. El proceso se repite de manera iterativa hasta calcular todas las posibles componentes ($min(n-1, p)$) o hasta que se decida detener el proceso. El orden de importancia de las componentes viene dado por la magnitud del valor propio asociado a cada vector propio.

3.2. Cluster analysis

El clustering es una técnica para encontrar y clasificar K grupos de datos (clusters). Así, los elementos que comparten características semejantes estarán juntos en un mismo grupo y separados de los otros grupos con los que no comparten características. Por esta razón es que se decidió usar esta técnica, esperando poder distinguir, en el conjunto de datos, grupos de usuarios que compartan ciertas semejanzas y de esta manera distinguir distintos comportamientos de los usuarios.

3.2.1. K-Means

El método K-means agrupa las observaciones en K clusters distintos, donde el número K lo determina explícitamente antes de ejecutar del algoritmo. K-means encuentra los K mejores clusters, entendiendo como mejor cluster aquel cuya varianza interna sea lo más pequeña posible. Se trata, por lo tanto, de un problema de optimización, en el que se reparten las observaciones en K clusters de forma que la suma de las varianzas internas de todos ellos sea la menor posible [33]. Dos de las medidas más comúnmente empleadas definen la varianza interna de un cluster ($W(C_k)$) como:

- La suma de las distancias euclídeas al cuadrado entre cada observación (x_i) y el centroide (μ) de su cluster. Esto equivale a la suma de cuadrados internos del cluster.
- La suma de las distancias euclídeas al cuadrado entre todos los pares de observaciones que forman el cluster, dividida entre el número de observaciones del cluster.

Minimizar la suma total de varianza interna $\sum_{k=1}^{k} W(C_k)$ de forma exacta (encontrar el mínimo global) es un proceso muy complejo debido a la inmensa cantidad de formas en las que n observaciones se pueden dividir en K grupos. Sin embargo, es posible obtener una solución que, aun no siendo la mejor de entre todas las posibles, es muy buena (óptimo local). El algoritmo empleado para ello es:

1. Asignar aleatoriamente un número entre 1 y K a cada observación. Esto sirve como asignación inicial aleatoria de las observaciones a los clusters.
2. Iterar los siguientes pasos hasta que la asignación de las observaciones a los clusters no cambie o se alcance un número máximo de iteraciones establecido por el usuario.
    - Para cada uno de los clusters calcular su centroide. Entendiendo por centroide la posición definida por la media de cada una de las variables de las observaciones que forman el cluster.
    - Asignar cada observación al cluster cuyo centroide está más próximo.

Esta técnica de clustering requiere que se indique de antemano el número de clusters que se van a crear. Esto puede ser complicado si no se dispone de información adicional sobre los datos con los que se trabaja. Se han desarrollado varias estrategias para ayudar a identificar potenciales valores óptimos de K, aunque todas ellas son orientativas.

Un punto a tener en cuenta es que las agrupaciones resultantes pueden variar dependiendo de la asignación aleatoria inicial de los centroides. Para minimizar este problema se recomienda repetir el proceso de clustering entre 25-50 veces y seleccionar como resultado definitivo el que tenga menor suma total de varianza interna. Aun así, solo se puede garantizar la reproducibilidad de los resultados si se emplean semillas.

Silhouette

En este estudio, se adoptó la puntuación de Silhouette [34] para seleccionar el k óptimo. Es una forma de medir qué tan cerca está cada punto de un grupo de los puntos de sus grupos vecinos. Los valores de Silhouette se encuentran en el rango de [-1, 1]. Un valor cercano a 1 indica que la muestra está muy lejos de su cluster vecino y muy cerca del cluster asignado. De manera similar, un valor cercano a -1 indica que el punto está más cerca de su cluster vecino que del cluster asignado. Por lo tanto, cuanto mayor sea el valor, mejor es la configuración del cluster.

Se define el valor de Silhouette para un punto i perteneciente a un cluster, como *s(i)*, tal que:

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))} \tag{3.3}$$

donde *a(i)* es la distancia media del punto i a todos los otros puntos en el cluster asignado A. Se puede interpretar *a(i)* como qué tan bien se asigna el punto al cluster, cuanto menor sea el valor, mejor será la asignación.

Del mismo modo, se define *b(i)* como la distancia media del punto i a otros puntos de su cluster vecino más cercano B. El punto i no pertenece al cluster B, pero la distancia a este es la más cercana entre todos los demás clusters.

Para que *s(i)* sea cercano a 1, *a(i)* debe ser muy pequeño en comparación con *b(i)*. Esto sucede cuando i está muy cerca de los demás puntos asignados a su mismo cluster. Un gran valor de *b(i)* implica que está muy lejos de su próximo cluster más cercano. En resumen, cuando *s(i)* es cercano a 1, indica que el conjunto de datos i está bien definido como cluster. Para obtener el valor de Silhouette de todo el cluster, alcanza con la media de los valores de Silhouette de todo el conjunto de datos para un cluster.

3.2.2. Análisis de clusters jerárquicos (HCA)

En este estudio se adoptó HCA para realizar un análisis de mapa de calor. La idea central es complementar el análisis con la otra técnica de clustering, HCA utiliza un método distinto para generar las agrupaciones de datos, lo cual permite tener una visión de los resultados más completa. A diferencia de k-means, HCA comienza con cada uno de los n *data points* que pertenecen a n grupos diferentes. En el siguiente paso, los dos *data points* más similares se agrupan obteniendo así n-1 grupos. El proceso se repite hasta que todos los *data points* pertenecen a un grupo. El resultado es un árbol de clasificación jerárquica llamado dendrograma [35].

Para poder llevar a cabo este agrupamiento, primero hay que definir cómo se cuantifica la similitud entre dos clusters. Se tiene que extender el concepto de distancia entre pares de data points para que sea aplicable a pares de grupos, cada uno formado por varios data points. A este proceso se le conoce como *linkage*. Para este estudio se utilizó el linkage *ward* [36].

Ward se basa en el valor óptimo de una función objetivo, pudiendo ser esta última cualquier función definida por el analista. El método *Ward's minimum variance* es un caso particular en el que el objetivo es minimizar la suma total de varianza intra-cluster. En cada paso, se identifican aquellos 2 clusters cuya fusión conlleva menor incremento de la varianza total intra-cluster.

El análisis de mapa de calor es una imagen de color falso con dos dendrogramas para dos objetos diferentes y puede dividir estos dos objetos en varios grupos [37]. Las diferentes características de influencia en estos dos objetos se reordenaron de acuerdo con su similitud según el HCA.

3.3. t-Distributed Stochastic Neighbor Embedding (t-SNE)

Una de las grandes diferencias que tiene el algoritmo de t-SNE con los vistos hasta el momento, es que es un algoritmo no lineal, esto permite complementar los análisis y resultados que se realicen con los algoritmos lineales. También es un algoritmo utilizado para la reducción de dimensiones, como se detallará en los siguientes párrafos, con lo que puede ser un gran método exploratorio y servir como comparación con los resultados de PCA.

t-SNE surge como una extensión del algoritmo *stochastic neighbor embedding* (SNE). Es un algoritmo de reducción de dimensionalidad no lineal [38], para encontrar patrones en los datos, identificando grupos observados, basándose en la similitud de datos con múltiples características.

Pero no es un algoritmo de agrupamiento, es un algoritmo de reducción de dimensionalidad. Esto se debe a que asigna los datos multidimensionales a un espacio de menor dimensión, las entidades de entrada ya no son identificables. Por lo tanto, no puede hacer ninguna inferencia basada solo en la salida de t-SNE. Básicamente, es principalmente una técnica de exploración y visualización de datos.

El algoritmo [39] t-SNE comienza trabajando primero con los datos en su dimensionalidad de origen (la cual se supone que es elevada), transformando las distancias

euclidianas entre los puntos en probabilidades condicionales que representan similitudes. Por lo que la similitud entre dos observaciones o puntos x_j y x_i, puede definirse como la probabilidad condicional $p_{j|i}$, de que el punto x_j fuese seleccionado como vecino del punto x_i, si los puntos pertenecieran a una distribución gaussiana centrada en x_i. Para puntos cercanos, $p_{j|i}$ tiene un valor alto, mientras que para puntos alejados la probabilidad disminuye. La probabilidad condicional se define como:

$$p_{j|i} = \frac{exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\Sigma_{k \neq i} exp(-\left\|x_i - x_k\right\|)^2 / 2\sigma_i^2} \tag{3.4}$$

donde σ_i es la varianza de la distribución gaussiana centrada en x_i.

Siguiendo la idea anterior, se crean dos observaciones iguales a x_j y x_i pero en una dimensionalidad menor, llamémosles y_j y y_i, cuya distancia se define como la probabilidad condicional $q_{j|i}$.

$$q_{j|i} = \frac{exp(-\left\|y_i - y_j\right\|^2}{\Sigma_{k \neq i} exp(-\left\|y_i - y_k\right\|)^2} \tag{3.5}$$

Como resultado se obtiene la probabilidad condicional de similitud entre observaciones en el espacio multidimensional original y en un espacio de menor dimensión. Para que las observaciones y_j y y_i, creadas en el espacio de menor dimensión, sean los más similares posibles a las observaciones x_j y x_i que están en un espacio con mayor dimensionalidad, las probabilidades condicionales $p_{j|i}$ y $q_{j|i}$ deben ser exactamente las mismas.

Para poder obtener las probabilidades condicionales es necesario seleccionar un valor de σ_i. Es muy poco probable que exista un valor de varianza σ_i óptimo para todas las observaciones del conjunto de datos, ya que la densidad de los datos suele variar de una región a otra. Para regiones densas, un valor pequeño de σ_i es mucho más apropiado que para regiones dispersas. t-SNE realiza una búsqueda binaria para encontrar el valor óptimo σ_i con una *perplexity* fijada por el usuario. La *perplexity* puede entenderse como una medida del número de observaciones vecinas que tienen que emplearse en cada estimación local. Suele ser recomendable seleccionar valores entre 5 y 50.

El algoritmo de t-SNE calcula la probabilidad condicional de cada par de observaciones y trata de minimizar la suma de las diferencias entre las probabilidades de la dimensión superior e inferior.

3.4. Self-Organizing Map (SOM)

La principal razón por la que se eligió el algoritmo SOM para esta tesis es que tiene un método de aprendizaje diferente a aquellos descritos anteriormente en este capitulo. En primer lugar, con él se pueden obtener una gran variedad de resultados a partir de una sola ejecución, por ejemplo, permite hacer una reducción de dimensiones, hacer *clustering* y detectar anomalías al mismo tiempo. A su vez, a través de SOM se pueden representar los resultados en diversas formas de visualización, ya sea mapas de calor, gráficas de puntos y hasta combinaciones de varias de ellas.

SOM es un tipo de red neuronal artificial (ANN) que se entrena mediante el aprendizaje no supervisado para producir una representación discretizada (mapa SOM) de baja dimensión (normalmente 2-D) del espacio de entrada [40, 41]. SOM se diferencia de otras redes neuronales artificiales en que aplica el aprendizaje competitivo en oposición al aprendizaje con corrección de errores (como la propagación hacia atrás con

descenso de gradiente) y en el sentido de que utilizan una función de vecindad para preservar las propiedades topológicas del espacio de entrada.

SOM está formado por neuronas ubicadas en una grilla regular, generalmente de 2 o 3 dimensiones. También es posible crear grillas de dimensiones superiores, pero generalmente no se usan porque dificulta su visualización. Las neuronas están conectadas entre sí por una relación/función de vecindad que dicta la estructura del mapa. En el caso bidimensional, las neuronas del mapa se pueden organizar en una red rectangular o hexagonal.

La cantidad de neuronas empleadas se fija antes de comenzar con el procesamiento. El número de neuronas debe ser lo más grande posible, sin embargo, grandes cantidades de neuronas implican un exceso computacional para el procesamiento, lo cual puede llegar a ser muy impráctico. Por lo cual, un número razonable de neuronas a utilizar es $5 * sqrt(n)$ [40], en donde n es la cantidad de datos/observaciones que serán utilizados para el entrenamiento del algoritmo.

Luego hay que inicializar el peso de las neuronas. Típicamente, existen dos formas de hacer eso de manera simple, una es inicializando los valores de manera aleatoria, y la otra es inicializar los vectores de peso con muestras aleatorias del conjunto de datos.

En cada paso de entrenamiento, se elige aleatoriamente un vector de muestra del conjunto de datos de entrada y se calcula una medida de similitud entre este y todos los vectores de peso del mapa (grilla). La unidad de mejor coincidencia (*Best-Matching Unit* - BMU), es la unidad cuyo vector de peso tiene la mayor similitud con la muestra de entrada. La similitud generalmente se define por medio de una medida de distancia, típicamente distancia euclidiana.

Después de encontrar la neurona de la grilla que mejor se asemeja a la muestra de entrada, los vectores de peso, de esta neurona y el de sus vecinos, se actualizan y tratan de asemejarse al vector de la muestra de entrada. Esto hace que la neurona y sus vecinos "tiren" el mapa (grilla) hacia la muestra de entrada. Este proceso de extracción y actualización continúa recorriendo los datos de entrada hasta que termina con un mapa totalmente convergente.

3.5. Linear Regression

Este modelo predictor se eligió para comenzar por ser un modelo simple, que se basa en relaciones lineales de las variables. Permite obtener una representación matemática, una ecuación, que se podrá utilizar para estimar el tráfico consumido en los centros educativos. Al encontrar los coeficientes de la ecuación se puede tener una proyección del peso de las variables utilizadas.

La regresión lineal es un enfoque popular a la hora de trabajar con aprendizaje supervisado. Se utiliza para describir la relación entre una variable dependiente Y, y una o más variables independientes $X_0 ... X_n$. En su versión más simple, se tiene una sola variable dependiente, Y, y una sola variable independiente o predictora, X. Además, se asume que existe una relación lineal entre X e Y. Matemáticamente, podemos escribir esta relación lineal como [42]:

$$Y = \beta_0 + \beta_1 * X + \epsilon \tag{3.6}$$

Siendo β_0 la ordenada en el origen, β_1 la pendiente y ϵ el error aleatorio. Este último representa la diferencia entre el valor ajustado por la recta y el valor real. Recoge el efecto de todas aquellas variables que influyen en Y pero que no se incluyen en el modelo como predictores. Al error aleatorio también se le conoce como residuo. El método de mínimos cuadrados se utiliza para ajustar el modelo de regresión lineal. Ajusta los valores de los coeficientes β_0 y β_1 y minimiza la suma de las desviaciones cuadradas entre el resultado real y el resultado predicho por el modelo.

La regresión lineal simple es un enfoque útil para predecir una respuesta sobre la base de una única variable predictora. Sin embargo, en la práctica, a menudo tenemos más de un predictor.

La regresión lineal múltiple, al igual que la regresión lineal simple, utiliza el método de mínimos cuadrados para ajustar los coeficientes. Una expresión genérica para el modelo de regresión lineal múltiple es la siguiente:

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \beta_3 * X_3 + ... + \beta_n * X_n + \epsilon \tag{3.7}$$

donde X_j representa el j-ésimo predictor y β_j cuantifica la asociación entre esa variable y la respuesta. Interpretamos β_j como el efecto promedio sobre Y de un aumento de una unidad en X_j, manteniendo fijos todos los demás predictores.

3.6. Random Forest (RF)

El modelo de RF se escogió para comparación con el método anterior, ya que también es un modelo simple pero con resultados más exactos. Requiere menos esfuerzo de preprocesado de los datos en comparación a otros métodos de aprendizaje estadístico, por ejemplo, no requieren estandarización. También es muy útil para la exploración de los datos, ya que como resultado intermedio, permite identificar las variables más importantes.

Random Forest (RF), como su nombre lo indica, consiste en una gran cantidad de árboles de decisión individuales que operan en conjunto. Cada árbol individual en RF realiza una predicción de clase y la clase con más votos se convierte en la predicción de nuestro modelo [43].

En ciencia de datos, la razón por la que el modelo de RF muestra un buen desempeño en diferentes aplicaciones es que una gran cantidad de modelos (árboles) relativamente no correlacionados que operan como comité superará a cualquiera de los modelos constituyentes individuales.

La baja correlación entre modelos es la clave. Los modelos no correlacionados pueden producir predicciones de conjunto que son más precisas que cualquiera de las predicciones individuales. La razón de este efecto es que los árboles se protegen entre sí de sus errores individuales (siempre que no se equivoquen constantemente en la misma dirección). Si bien algunos árboles pueden estar equivocados, muchos otros árboles estarán en lo correcto, por lo que, como grupo, los árboles pueden moverse en la dirección correcta. Los requisitos previos para que un bosque aleatorio funcione bien son:

- Es necesario que haya alguna señal real en nuestras funciones para que los modelos creados con esas funciones funcionen mejor que las conjeturas aleatorias.
- Las predicciones (y, por lo tanto, los errores) hechas por los árboles individuales deben tener bajas correlaciones entre sí.

Para asegurarse la baja correlación entre los árboles, RF utiliza la técnica de *bagging*. Los árboles de decisión son muy sensibles a los datos usados en el entrenamiento del modelo. Pequeños cambios en el conjunto de entrenamiento pueden dar como resultado estructuras de árboles significativamente diferentes. RF hace uso de esto al permitir que cada árbol individual emplee para el entrenamiento muestras obtenidas aleatoriamente del conjunto de datos, lo que da como resultado diferentes árboles. Este es el proceso que se conoce como *bagging*.

Otra técnica que se utiliza para obtener baja correlación es la *aleatoriedad de features*. En un árbol de decisión normal, cuando es el momento de dividir un nodo, consideramos todas las características posibles y elegimos la que produce la mayor

separación entre las observaciones en el nodo izquierdo y las del nodo derecho. Por el contrario, cada árbol en RF puede elegir solo de un subconjunto aleatorio de características. Esto obliga a una variación aún mayor entre los árboles en el modelo y, en última instancia, da como resultado una menor correlación entre los árboles y una mayor diversificación.

Capítulo 4

Estudio de Categorías

En este capítulo, se presenta el análisis y los resultados de las consultas DNS, teniendo como eje central las categorías que son asignadas a cada registro. El objetivo no es analizar todas las categorías en conjunto, sino que se procede, inicialmente, a hacer una selección de categorías de interés, las cuales serán utilizadas luego como base para realizar los análisis en profundidad.

En primer lugar, se analiza el uso de las categorías en cada tipo de centro educativo, escuelas y liceos, de todo el país. En segundo lugar, se procede a hacer un análisis del uso de las categorías con respecto al horario de funcionamiento de estos. De este análisis se desprenden las relaciones entre las categorías y las horas del día, así como los períodos de tiempo de mayor uso de la red.

Para poder efectuar estos estudios se utiliza el algoritmo t-sne, para hacer una primera aproximación y exploración de los datos. Luego se procede con los algoritmos lineales de PCA y k-means, a partir de los cuales se obtienen los agrupamientos de datos y cómo estos se relacionan con las *features* o variables del algoritmo, en este caso son las categorías que se deciden analizar. Estos resultados se comparan con SOM (algoritmo no lineal), tanto para validar los resultados anteriores como para obtener información complementaria. Por último, se emplean *heatmaps* y *clusters* jerárquicos, como complemento a los métodos anteriores y para tener una representación más efectiva de los resultados.

4.1. Selección y pre-procesamiento de categorías

Como se menciona en el capítulo 2.3, se realiza un preprocesamiento de las categorías, en donde para cada registro se obtiene la categoría más representativa a la que pertenece el dominio de consulta DNS. A su vez, se lleva a cabo un filtrado temporal, en donde solo se toman en cuenta las consultas de lunes a viernes de 8:00 a 17:00 horas. Como resultado intermedio de este proceso, en la Figura 4.1, se muestra la distribución en porcentaje de las 25 categorías más populares (en términos del número de consultas DNS a las que fueron asignadas), teniendo en cuenta el filtrado temporal antes mencionado. Como puede verse en la Figura 4.1, dentro de este grupo de categorías existe un subgrupo que no refieren al comportamiento de los usuarios en la red, además en el caso de *Computer Security* también cumple la condición de que no aporta información de valor (representan menos de un 0,5 % del total), por lo que se decide eliminar este subgrupo de categorías compuesto por:

- **Infrastructure** - Infraestructura de entrega de contenido y contenido generado

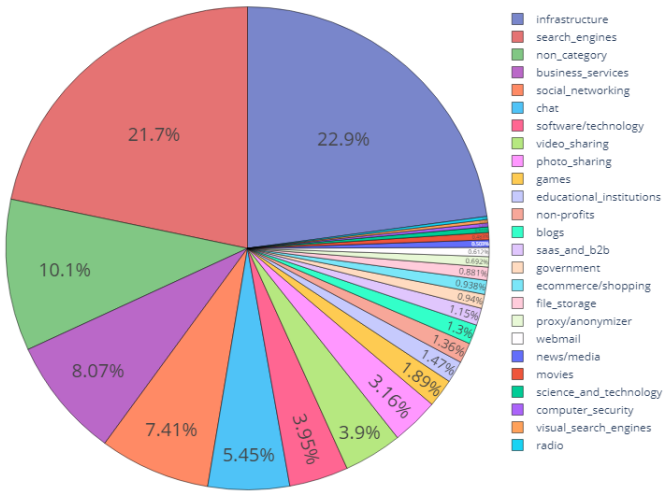


Figura 4.1: 25 categorías más populares. Lunes a viernes, de 8:00 a 17:00 hs.

dinámicamente, sitios web que no pueden clasificarse más específicamente porque son seguros o de otra manera difíciles de clasificar.

- **Non-Category** - Sitios a los que el sistema no pudo asignar una categoría.
- **Business Services** - Sitios para corporaciones y negocios de todos los tamaños, especialmente sitios web de empresas.
- **Software/Technology** - Sitios sobre informática, hardware y tecnología, incluidas noticias, información, código e información del proveedor.
- **SaaS and B2B** - Portales web para servicios comerciales en línea.
- **Proxy/Anonymizer** - Sitios que proporcionan información o servicios de omisión de proxy.
- **Computer Security** - Ofreciendo productos y servicios de seguridad para usuarios corporativos y domésticos.

También existen categorías con un contenido muy parecido entre sí, por lo que se decide crear dos nuevas meta-categorías con el propósito de agrupar contenido similar de varias categorías en una sola. Estas dos nuevas meta-categorías son:

- **Social_Networks** = Social_Networking + Photo_Sharing + Blogs + Chat.
- **Streaming** = Video_Sharing + Movies + Radio.

En donde:

- **Social Networking** - Sitios que promueven la interacción y la creación de redes entre las personas.
- **Chat** - Sitios donde puede chatear en tiempo real con grupos de personas. Incluye sitios de vídeo chat.

- **Photo Sharing** - Sitios para compartir fotos, imágenes, galerías y álbumes.
- **Blogs** - Sitios de revistas personales o grupales, sitios de diarios o publicaciones.
- **Video Sharing** - Sitios para compartir contenido de vídeo.
- **Movies** - Sitios que promocionan películas u ofrecen visualización de películas en línea.
- **Radio** - Sitios que ofrecen escucha de radio en línea o promueven estaciones de radio.

Además, se obtuvo una lista explícita de dominios educativos de interés por parte del equipo de Programas de Educación del Plan Ceibal (Table 4.1). Estos dominios se clasificaron como *educational_institutions*, sin tener en cuenta la clasificación anterior de Cisco Umbrella. Siguiendo el mismo procedimiento, todos los dominios que terminan en "edu.uy" también se categorizaron de la misma manera. Además, para el propósito de este estudio, se decidió omitir las categorías que representan menos del 0.5 % del total de consultas de DNS. En la Figura 4.2, se muestra la distribución final que se utilizará a lo largo de esta tesis, en donde:

- **Search Engines** - Sitios que ofrecen listados de resultados basados en palabras clave.
- **Social Networks** - Nueva meta-catgoría en la que se agrupan las categorías *social_networking*, *photo_sharing*, *blogs* y *chat* de Umbrella.
- **Streaming** - Nueva meta-categoría en la que se agrupan las categorías *video_sharing*, *movies* y *radio* de Umbrella.
- **Games** - Juegos y sitios que ofrecen información sobre juegos.
- **Educational Institutions** - Sitios para escuelas que cubren todos los niveles y tipos de edades.
- **Non-Profits** - Sitios para organizaciones y servicios sin fines de lucro o de caridad.
- **Ecommerce/Shopping** - Sitios que son tiendas en línea de productos y servicios.
- **File Storage** - Sitios que ofrecen espacio para alojar, compartir y realizar copias de seguridad de archivos digitales.
- **Government** - Sitios operados por agencias gubernamentales, incluidos los niveles municipal, estatal, regional y federal.
- **Webmail** - Sitios que ofrecen la posibilidad de enviar o recibir un correo electrónico.
- **News/Media** - Sitios que ofrecen noticias e información, incluidos periódicos, emisoras y otros editores.
- **Science and Technology** - Ciencia y tecnología, como aeroespacial, electrónica, ingeniería, matemáticas y otras materias similares; exploración espacial; meteorología; geografía; medio ambiente; energía (fósil, nuclear, renovable); comunicaciones (teléfonos, telecomunicaciones).

Tabla 4.1: Tabla de dominios

Dominio	URL	Dominio	URL
Crea	ceibal.schoology.com	Code	code.org
Ibirapitá	ibirapita.org.uy	Pilas Bloques	pilasbloques.program.ar
Matific	matific.com	Geogebra	geogebra.org
Tu clase	tuclase.uy	Kahoot	kahoot.it
Uruguay estudia	uruguayestudia.uy	Quizizz	quizizz.com
Open Roberta Lab	lab.open-roberta.org	Wikipedia	wikipedia.org
Microbit	microbit.org	Tig-Tag	tigtag-es.com
Microbit python editor	python.microbit.org	Twig	twig-es.com
Tinkercad	tinkercad.com	Khan Academy	es.khanacademy.org
Desafío profundo	desafioprofundo.org	Edmodo	edmodo.com
Scratch	scratch.mit.edu	Alter Ceibal	alterceibal.uy
Tynker	tynker.com	MIT App Inventor	appinventor.mit.edu
LightBot	lightbot.com	Blockly Games	blockly.games

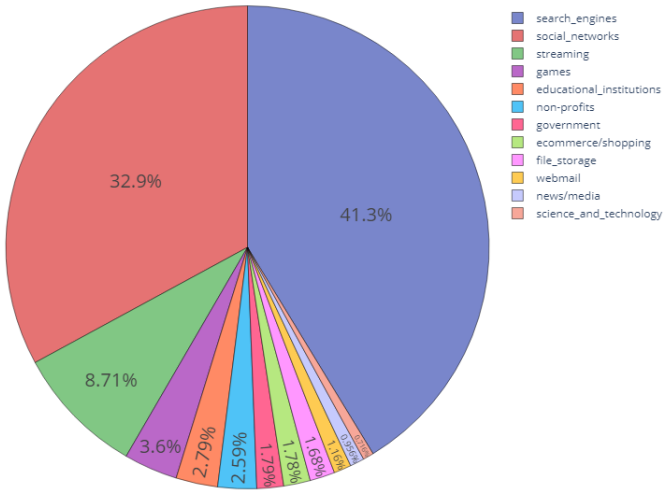


Figura 4.2: Porcentaje del total de consultas DNS solicitadas por categoría.

4.2. Análisis y resultados

En una primera instancia, se trabajó con las categorías que el sistema Cisco Umbrella asigna a cada consulta. En esta primera incursión a los datos se realizaron dos análisis diferentes y con objetivos distintos. Para ambos análisis se utilizaron métodos lineales y no-lineales, aunque al tener objetivos distintos, las matrices de datos que se utilizaron como entrada para los métodos fueron diferentes para cada análisis.

Para el primer estudio, se tiene como objetivo poder distinguir el comportamiento de los diferentes grupos etáreos (niños y adolescentes) que forman parte de los usuarios de la red Ceibal. Para el segundo estudio, se decidió separar el conjunto de datos y examinar las observaciones de la escuela y liceos de forma independiente, con el

principal objetivo de analizar el comportamiento de los usuarios de la red durante las horas del día.

Para lograr el objetivo del primer estudio, se decidió ejecutar una agregación de los datos y construir una matriz que será utilizada como entrada para los distintos métodos. Esta matriz tiene como columnas las 12 categorías descritas en la sección 4.1, y como filas, el tipo de centro educativo ya sea escuela o liceo, por departamento. En total se muestran 38 filas (dos por departamento), en donde cada fila representa la suma por departamento de las consultas efectuadas por las escuelas y liceos durante el año.

Cada departamento tiene diferentes cantidades de escuelas y liceos, por lo que el número de consultas por departamento fue promediado por el número de centros educativos, escuelas o liceos según corresponda, que se encuentran en ese departamento en particular. Además, dado que la importancia de las columnas (*features*) es independiente de su propia varianza, se centra cada una de ellas, restando a cada uno de sus valores observados, la media de la columna. Luego la estandarizamos dividiendo cada valor por la desviación estándar de la columna.

La matriz resultante (38×12) es utilizada como entrada para cada uno de los métodos. Primero se usó t-sne como método exploratorio, para poder tener una representación en dos dimensiones de las observaciones y de esta manera tener un primer acercamiento a los datos. Como se puede visualizar en la Figura 4.3, hay dos o tres agrupaciones de datos definidas, las cuales se corresponden con los tipos de centros educativos presentes en el análisis en donde los liceos podría decirse que forman dos grupos.

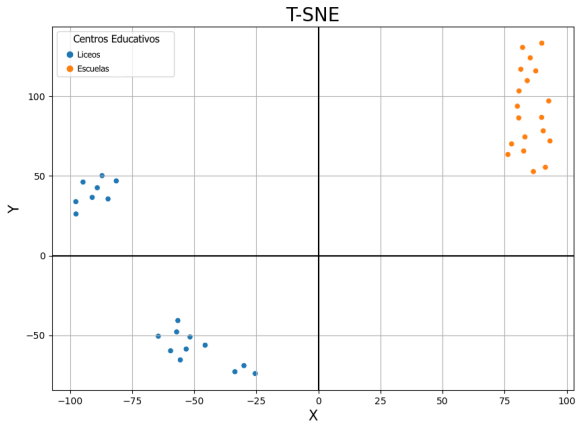


Figura 4.3: Exploración de los datos utilizando t-sne.

No obstante, debido a la naturaleza del algoritmo de t-sne, no es posible hacer ninguna inferencia del comportamiento de los datos teniendo como base solamente la salida del algoritmo. Por esta razón, en un siguiente paso, se usarán herramientas complementarias (PCA y k-means) con el fin de justificar el comportamiento de las observaciones. PCA permitirá disminuir la dimensionalidad y mejorar la visualización de los datos en conjunto con k-means, utilizado para identificar diferentes comportamientos grupales. Para seleccionar el mejor número k de *cluster* para el algoritmo de k-means, se empleó el método de Silhouette. El valor de Silhouette promedio de todos los valores considerados en el conjunto de datos se calculó y representó mediante

boxenplot (Figura 4.4).

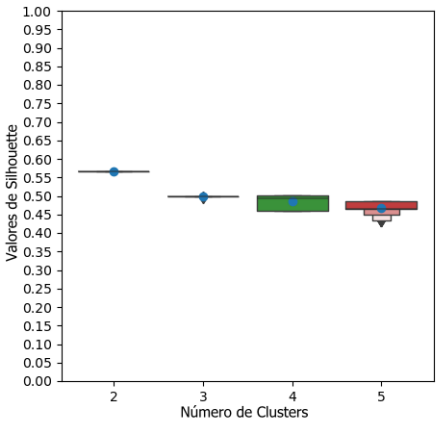


Figura 4.4: Boxenplots de los valores de silhouette.

Los *boxenplots* en la Figura 4.4 muestran una dispersión de los valores de Silhouette casi nula, por lo que se usó el número k de *cluster* correspondiente al valor de Silhouette más alto ($k = 2$).

En la Figura 4.5, se muestra el resultado que se obtuvo al ejecutar PCA y k-means. Para el caso del PCA, se seleccionaron los dos primeros componentes principales, ya que estos acumulan más del 95 % de la varianza (82.07 % y 17.21 % respectivamente). Anteriormente con t-sne no se pudo definir exactamente si existían dos o tres agrupaciones, con la utilización de PCA y k-means sí se identifican dos grandes grupos, en uno se agrupan los puntos pertenecientes a las escuelas, mientras que en el otro se encuentran los puntos de los liceos. Esto demuestra que existe una clara diferencia en el comportamiento de estos dos tipos de centros educativos en cuanto al uso de la red. También es notorio que hay una mayor variación en el PC1 por parte de los liceos, en comparación con las escuelas.

Para reafirmar y validar aún más los comportamientos encontrados hasta el momento, se hace un estudio utilizando el método SOM. Como es un método no-lineal, existe la posibilidad de encontrar nuevas relaciones entre las *features* que no se podrían apreciar con PCA y k-means, al ser métodos lineales.

Para visualizar el resultado de SOM (Figura 4.6), se utiliza un mapa de distancia, usando escala de grises, donde las neuronas de SOM se muestran como celdas y el color representa la distancia (pesos) de las neuronas vecinas. Nuevamente, se confirma la existencia de dos grandes grupos que representan los tipos de centros educativos, pero con el agregado de que pueden verse con claridad las neuronas que activan los centros de cada departamento, distinguiendo así cierta semejanza en cuanto al comportamiento, entre distintos departamentos para un mismo tipo de centro.

Teniendo en cuenta que en el primer estudio varios métodos confirmaron que existe una diferencia notable en el comportamiento de los usuarios entre distintos centros educativos, poder realizar el segundo análisis cobra una mayor importancia para poder distinguir mejor los comportamientos particulares de cada grupo.

Para este segundo estudio, se construyeron nuevas matrices de datos (una para cada tipo de centro educativo), en donde las filas representan las 9 horas del día estudiantil (de 8:00 a 17:00 hs), como se indica en la sección 2.4, y las columnas son las mismas

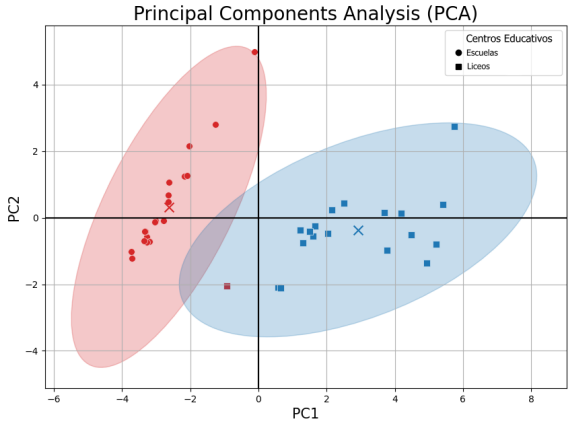


Figura 4.5: Los valores de PCA y k-means, utilizando los datos de escuelas y liceos.

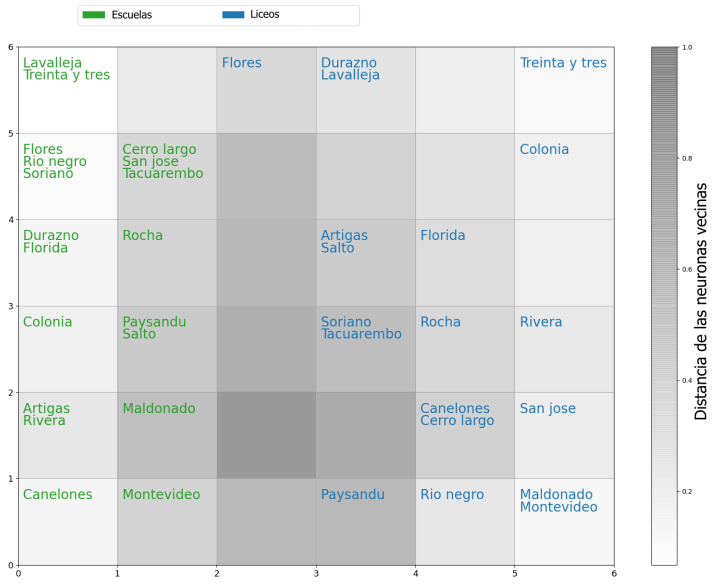
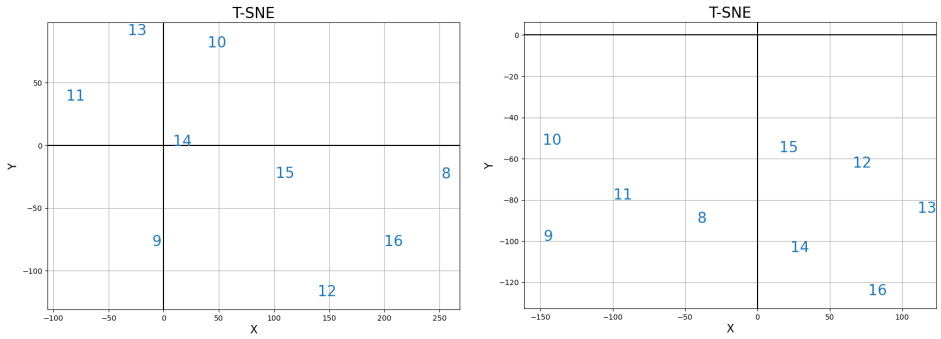


Figura 4.6: Mapa de distanciamiento de las neuronas de SOM.

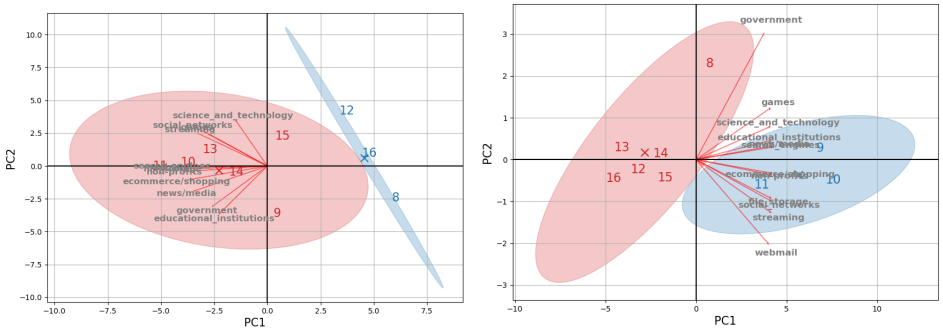
categorías que se utilizaron en el estudio anterior. Para cada fila, que representa una hora del rango con el que se trabaja, se consideró la suma de las consultas de cada categoría para esa hora determinada. De igual manera que en el estudio anterior, los valores de las columnas de las matrices fueron centrados y estandarizados.

Como resultado, se obtiene una matriz de entrada para los algoritmos de (9×12) para los datos de las escuelas y otra para los datos de los liceos. En este análisis se siguió la misma metodología que para el análisis anterior, en donde primero se utiliza



(a) Escuelas (b) Liceos

Figura 4.7: Exploración de los datos utilizando *t-sne*.



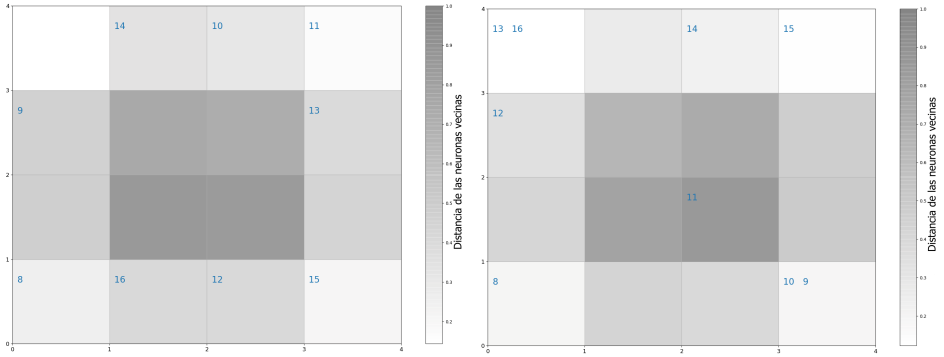
(a) Escuelas (b) Liceos

Figura 4.8: Biplot de PCA para identificar el comportamiento de los estudiantes durante las horas del día.

t-sne como método exploratorio de los datos, y luego, para poder realizar afirmaciones e inferencias sobre los datos, se aplican PCA con k-means como métodos lineales y SOM como método no-lineal.

Los resultados de ejecutar el algoritmo t-sne pueden verse en las Figuras 4.7(a) y 4.7(b). Tanto en las escuelas como en los liceos, la franja horaria parece dividirse en dos grupos. En el caso de las escuelas se encuentra un pequeño grupo con las horas 12, 16 y 8, y las horas restantes parecen encontrarse en el otro grupo. De similar manera, en los liceos, se tiene un pequeño grupo compuesto por las horas 9, 10 y 11, y existe un segundo grupo con las restantes horas, en este caso es más difícil delinear bien esos dos grupos. Para ambos conjuntos de datos, lo que se ve en esta división de la franja horaria en dos grupos es la separación de las horas con mayor y menor actividad de la red.

Para poder verificar y afirmar estos resultados, se procederá primero con la ejecución de los algoritmos PCA y k-means y luego SOM. En esta oportunidad también se eligió el método de Silhouette para obtener el mejor número de *clusters*. En ambos casos, tanto para el conjunto de datos de las escuelas como para el de los liceos, el promedio de los valores de Silhouette correspondiente a k = 2, es el valor más alto entre todas las k opciones.



(a) Escuelas (b) Liceos

Figura 4.9: Mapa de distanciamiento de neuronas del SOM, durante el día.

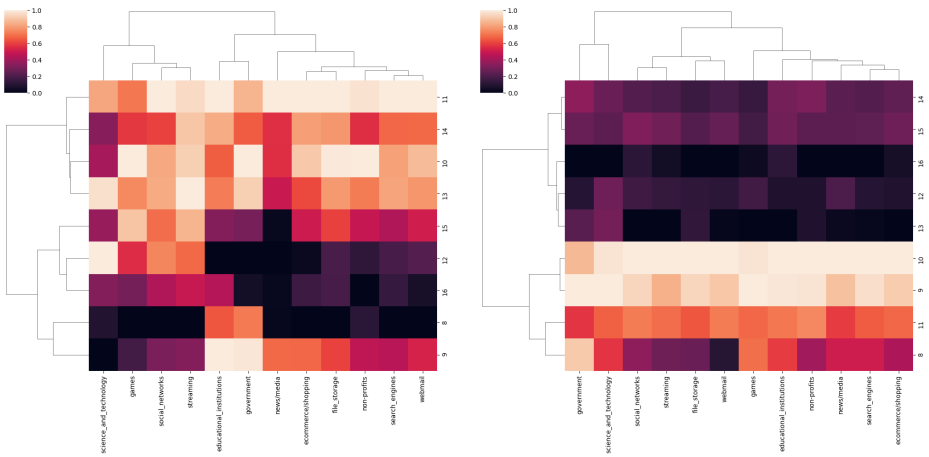
Los resultados obtenidos para el conjunto de datos de las escuelas y liceos se muestran en las Figuras 4.8(a) y 4.8(b), respectivamente. En ambos casos se seleccionaron los dos primeros componentes del PCA, ya que representaban más del 97 % (conjunto de datos de escuelas) y el 98 % (conjunto de datos de liceos) de la varianza. En los resultados que hacen referencia a los datos de las escuelas, se identificaron claramente dos grupos horarios, uno grande, que comprende la mayoría de las horas que abarcan el horario escolar (de 9:00 a 11:00 hs y de 13:00 a 16:00 hs) y otro más pequeño, en donde están las horas 8:00, 12:00 y 16:00. En el caso de los liceos se tiene dos grupos, uno con las horas de la mañana (de 9:00 a 11:00 hs) y el otro grupo, con las restantes horas.

En ambas figuras se muestran los vectores que representan: el rol de cada *feature* (categoría) en los dos componentes principales elegidos, la correlación que existe entre estos y cómo se relacionan con las horas utilizadas en el conjunto de datos. Para ambos casos se puede señalar que los vectores están orientados hacia los *clusters* que representan las horas de mayor actividad en la red, demostrando que durante el horario en donde hay mayor actividad escolar es cuando hay un uso extensivo de Internet.

Además, en el caso de las escuelas (Figura 4.8(a)), al observar los vectores de *educational_institutions* y *social_networks* se puede ver que son casi ortogonales, lo que demuestra su independencia. En el caso de los liceos, la diferencia entre estas dos categorías no es tan notoria, lo que indica que mantienen cierta correlación entre ellas. En particular, para las escuelas, *educational_institutions* tiene uso principalmente durante la mañana y temprano en la tarde, mientras que *social_networks* se utiliza más durante la hora de cambio de turno (12:00 hs) entre las clases de la mañana y la tarde, y luego durante las horas de finalización del horario escolar (de 15:00 a 16:00 hs). Por otro lado, otro punto que tienen en común ambas figuras es que los vectores de *social_networks* y *streaming* están orientados de manera casi igual, lo que indica de que, en ambos grupos etarios, estas categorías se comportan de manera similar.

Al realizar el análisis utilizando SOM, se puede ver que tanto para el conjunto de datos de las escuelas (Figura 4.9(a)) como para el de los liceos (Figura 4.9(b)), existe un grupo de neuronas que se activan y se separan notoriamente del resto, con las horas de menor actividad de la red. Esta agrupación de horas también se visualiza en los *biplots* de los PCA mencionados anteriormente.

En particular, para el caso de las escuelas, puede decirse que existe un grupo formado por las horas 10:00, 11:00, 13:00 y 14:00. Si bien no fue exactamente la misma agrupación que se obtuvo con k-means, se asemejan lo suficiente como para dar la idea



(a) Escuelas (b) Liceos

Figura 4.10: Mapa de calor, durante el día.

de que pueda existir una relación entre estas horas que no puede ser apreciada por el método k-means. Este es uno de los propósitos por los que se ejecutó este método no-lineal.

Como una herramienta analítica complementaria, con la intención de validar y agregar, en lo posible, información a los resultados ya obtenidos con los *biplots* de PCA (Figura 4.8(a) y la Figura 4.8(b)) y con SOM (Figura 4.9(a) y la Figura 4.9(b)), es que se decide ejecutar el análisis de mapas de calor jerárquicos utilizando el enlace *Ward* y la distancia euclidiana. Los resultados obtenidos de este análisis, tanto el conjunto de datos de las escuelas y como el de los liceos, se representan en las Figuras 4.10(a) y 4.10(b), respectivamente.

Visualizando los dendrogramas de la izquierda, para ambos mapas de calor, se destaca que hay dos grupos perfectamente identificables, los cuales se corresponden a las horas en donde hay alta actividad estudiantil en los centros y a las horas en donde la actividad es baja. Si se observa cómo se agrupan las horas para ambos conjuntos de datos, existe cierta semejanza con los resultados obtenidos con SOM.

Si se cambia el punto de vista y se observan los dendrogramas superiores en el caso escolar, el *cluster* formado por las categorías *educational_institutions* y *government*, también es identificable fácilmente en el *biplot* de los PCA del mismo set de datos. También se corrobora un comportamiento de *educational_institutions* que ya se había visto en los PCA, y es que existe un mayor uso durante las horas de la mañana y luego en las primeras horas de la tarde. En líneas generales, para el caso escolar, se puede ver que hay una mayor actividad de la red en las últimas horas del turno matutino y en las primeras del turno vespertino.

Por el contrario, en el caso de los liceos, mirando los dendrogramas de la izquierda, se puede deducir que hay una mayor actividad de la red por la mañana, mientras que por la tarde el uso disminuye. Tampoco se visualiza una gran diferencia del uso entre *educational_institutions* y el grupo formado por *social_networks* y *streaming*, dando a entender que hay tráfico perteneciente a este grupo durante el horario de clase, con la misma intensidad de uso que *educational_institutions*. Este resultado no era del todo claro en los *biplots* de *PCA*, por lo que con este método se mejoraron los resultados obtenidos.

4.3. Conclusiones

Dos principales análisis se presentan en este capítulo, teniendo como base las categorías de las consultas DNS. El primero con el objetivo de poder apreciar comportamientos de los usuarios dependiendo del grupo etáreo al que pertenecen. Como resultado de la ejecución de diferentes técnicas exploratorias de tipo no supervisado, se observó claramente que hay diferencias en el uso de la red dependiendo del centro educativo, esto motivó la separación del dominio de datos por el tipo de centro educativo.

Para el segundo análisis el foco estuvo puesto en poder distinguir los comportamientos particulares de cada grupo durante las horas del día. Nuevamente se aplicaron técnicas exploratorias no supervisadas para poder obtener resultados, en particular para el caso de las escuelas se visualizaron dos grandes grupos, uno que abarca las horas de 9:00 a 11:00 y de 13:00 a 16:00 y otro con las horas 8:00, 12:00 y 16:00. En el caso de los liceos también se tiene dos grupos, uno con las horas de 9:00 a 11:00 y otro con las restantes. Una vez definidos estos grupos, se observan cómo se comportan las categorías con respecto a estos, en donde se aprecia que los vectores de las categorías apuntan hacia los *clusters* que representan las horas de mayor actividad.

En el siguiente capítulo se ahondará más en el tema, en donde se presentará un estudio similar al de este capítulo pero teniendo como base los dominios pertenecientes a ciertas categorías.

Capítulo 5

Estudio de Dominios

Una vez hecho el estudio por categorías y al tener una mejor visión del comportamiento de los usuarios, se procede a hacer foco en ciertas categorías de interés, para poder realizar un estudio de comportamiento de los dominios que pertenecen a las mismas. A estos dominios se les efectúa un preprocesamiento para poder tener una mejor representación de su uso.

En cuanto a los análisis, estos siguen la misma línea que en el capítulo anterior, tratando de distinguir cómo se comportan los usuarios de cada centro educativo durante las horas del día. En este caso, en vez de categorías se tienen en cuenta los dominios.

La metodología utilizada es también muy similar a la empleada en el capítulo anterior. Lo que varía es que en este caso se hacen estudios teniendo en cuenta otras variantes, como, por ejemplo, el comportamiento de los centros educativos con respecto a los quintiles socioeconómicos. También se profundiza en los resultados obtenidos mediante SOM, agregando información recolectada por este algoritmo sobre el comportamiento de cada *feature*.

5.1. Pre-procesamiento de dominios

Dado que el objetivo principal de este trabajo es analizar el perfil de red del usuario de Plan Ceibal, estudiamos los dominios de un subgrupo de tres categorías de interés. En primer lugar, se consideraron las *social_networks* y *streaming*, ya que son dos de las categorías más populares (Figura 4.2). Además, debido a que el Plan Ceibal es un ESP, consideramos la categoría *educational_institutions*. La categoría *Search_engines* no se tuvo en cuenta porque el dominio Google cuenta con el 90 % de las consultas de DNS dentro esta.

Al igual que en el análisis de categorías, en este estudio se reagrupan algunos dominios. En particular, se fusionaron los dominios o sub-dominios que pertenecen a un dominio más general, por el hecho de que varios dominios utilizan estos otros sub-dominios para cargar imágenes, vídeos, etc.

En el caso de *Social_Networks* (Figura 5.1), los dominios que representan menos del 0.5 % del total de consultas DNS dentro de la categoría no fueron considerados en este estudio. Además, se fusionaron los siguientes dominios:

- **facebook** = facebook + fbcdn + fb
- **pinterest** = pinterest + pinimg
- **twitter** = twitter + twimg

- **snapchat** = snapchat + sc-cdn

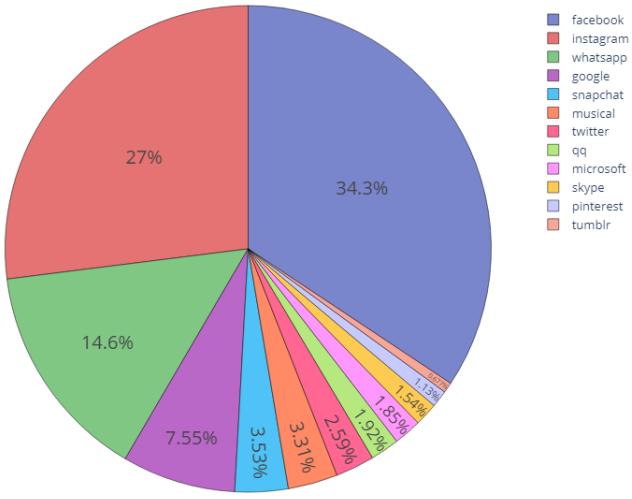


Figura 5.1: Porcentaje del total de consultas DNS en la categoría Social Networks.

Para la categoría *Streaming* (Figura 5.2), existía la peculiaridad de que los primeros cinco dominios ya representaban casi el 99 % de las consultas DNS. Por esta razón, y para aumentar la diversidad de categorías, decidimos agregar los siguientes cinco dominios más populares, diez en total. Además, se fusionaron los siguientes dominios:

- **google** = google + ytimg + like
- **netflix** = netflix + nflxso + nflxvideo + nflximg

Finalmente, con respecto a la categoría *Educational_Institution* (Figura 5.3), los dominios *Android* y *Ubuntu* no fueron considerados, ya que no brindaban información sobre el comportamiento del usuario. Además, los dominios que representaron menos del 0,5 % del total de consultas de DNS dentro de esta categoría no se incluyeron en este estudio.

5.2. Análisis y resultados

Para poder ampliar los resultados del segundo estudio, presentado en la sección 4.2, se tuvo como objetivo centrar un nuevo análisis en ciertas categorías de interés (*educational_institutions*, *social_networks* y *streaming*). Para esto se decidió hacer un análisis de los datos de cada categoría por separado, trabajando con los dominios que las componen. El objetivo es poder visualizar el comportamiento de los usuarios durante las horas del día (las mismas referenciadas anteriormente) con respecto a los dominios que componen a cada categoría.

La matriz de datos utilizada para este caso es muy similar a la ya utilizada en el segundo estudio de la sección 4.2. En la matriz de cada tipo de centro educativo

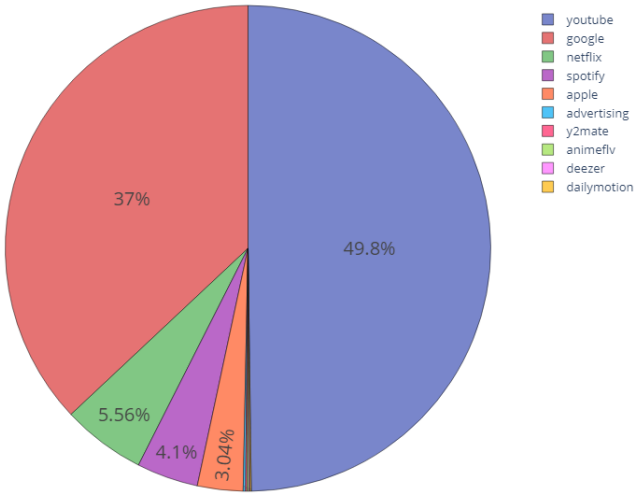


Figura 5.2: Porcentaje del total de consultas DNS en la categoría Streaming.

(escuelas y liceos), las filas representan las 9 horas del horario de clase y las columnas, o *features*, los dominios de cada categoría, presentado en la sección 5.1.

Por último, se realiza un estudio para tratar de encontrar si existe alguna relación entre los centros educativos y los dominios de las categorías de interés. Para esto se emplea una matriz en donde las columnas son los dominios pertenecientes a cada categoría, y las filas son la suma de consultas para ese dominio durante todo el año electivo, para cada centro educativo. En la visualización de estos resultados se procede a mostrar cada centro con el color correspondiente a su quintil. Siguiendo la misma línea, se llevan a cabo mapas de calor con el mismo propósito que en el capítulo anterior, pero agregando los datos por quintil y manteniendo como columnas los dominios.

5.2.1. Dominios dentro de la categoría *social networks*

Como primer análisis a presentar, se procederá a ejecutar los algoritmos de PCA y k-means. Tanto para el set de datos de las escuelas como para el de los liceos, se utiliza k = 2 como número de cluster, obtenido como promedio de valores de Silhouette.

De los resultados obtenidos, tanto para el set de datos de las escuelas (Figura 5.4(a)) como para el de los liceos (Figura 5.4(b)), se puede inferir que, al igual que pasaba con el análisis de categorías, existen dos grandes grupos: uno en el que están las horas de mayor actividad de los centros educativos y en el otro, las horas en donde estos centros tienen menor actividad. En particular, en el caso de las escuelas puede observarse el comienzo del día educativo (8 y 9 horas) pertenece al grupo de menor actividad. Dando a entender, que para el caso de los dominios que componen la categoría *social_networks*, su nivel de uso al comienzo del día se aleja bastante al resto de la jornada.

Si se observan los vectores que representan el rol de cada *feature* o dominio en los

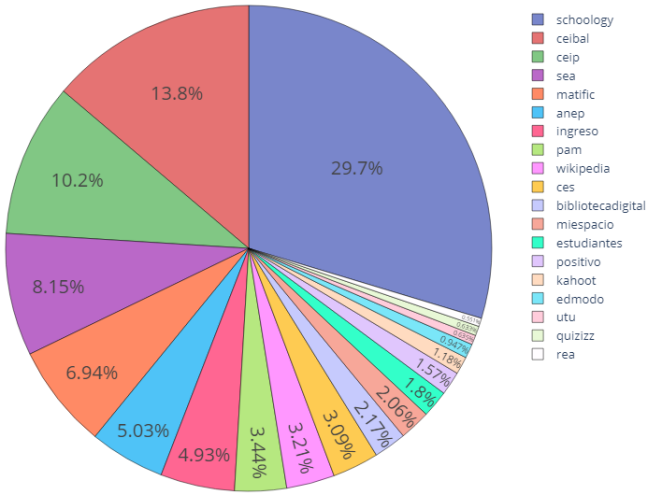
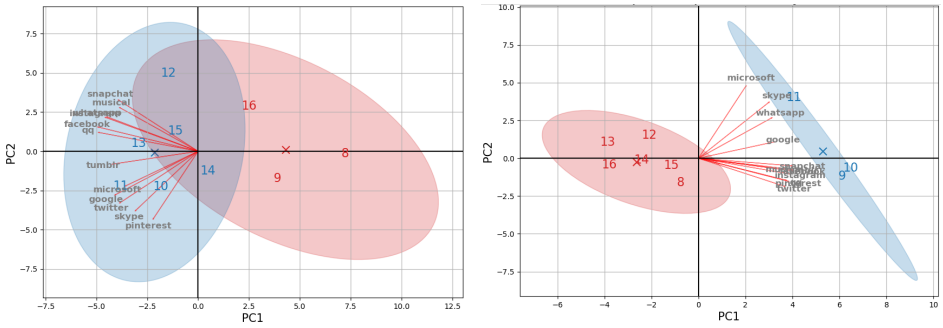


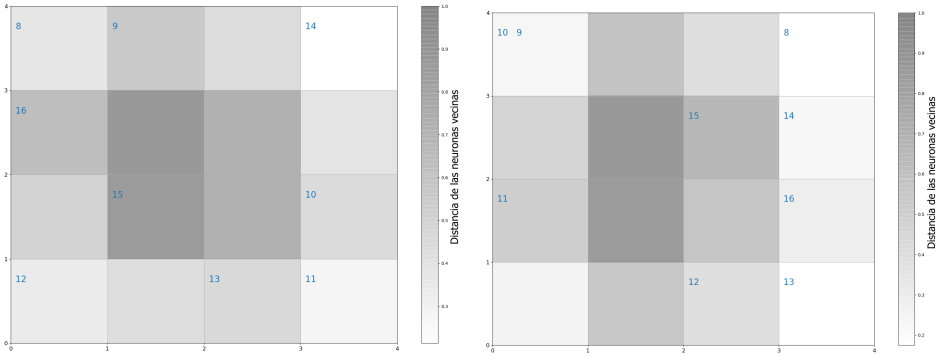
Figura 5.3: Porcentaje del total de consultas DNS en la categoría Educational Institutions.



(a) Escuelas (b) Liceos

Figura 5.4: Biplot de PCA para identificar el comportamiento de los estudiantes durante las horas del día.

dos componentes principales elegidos, se tiene comportamientos distintos dependiendo del conjunto de datos. En el caso de las escuelas se puede decir que existen dos grupos de vectores que sobresalen, el primero compuesto por los dominios *snapchat*, *instagram*, *whatsapp* y *facebook*, orientados hacía las horas de la tarde principalmente, y el segundo grupo compuesto por *twitter* y *skype*, orientado hacía las horas de la mañana mayormente. También puede verse que hay ciertos vectores pertenecientes a estos grupos que son casi ortogonales entre sí, (*snapchat* y *twitter*, por ejemplo), demostrando



(a) Escuelas (b) Liceos

Figura 5.5: Mapa de distanciamiento de neuronas del SOM, durante el día.

su independencia.

En el caso del conjunto de datos de los liceos, se tiene a los vectores *microsoft* y *skype*, orientados hacia las 11 horas de la mañana. El resto, en su mayoría, están casi ortogonal a estos dos vectores, indicando que hay una fuerte independencia entre estos dos grupos de vectores.

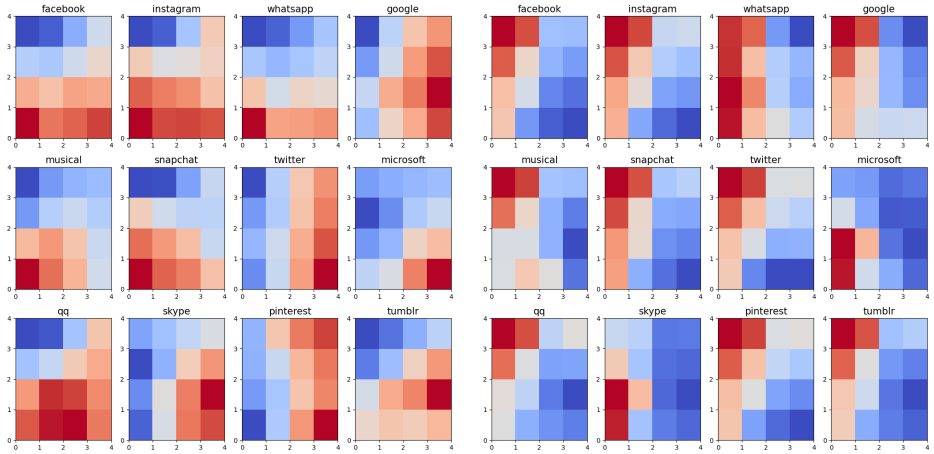
Al observar los resultados de SOM para el conjunto de datos de las escuelas (Figura 5.5(a)) y para el de los liceos (Figura 5.5(b)), puede apreciarse que existen algunas ligeras diferencias con respecto a k-means al observarse las distancias entre neuronas. En el caso de las escuelas se tiene, por un lado, el mismo grupo que en k-means con las horas 8, 9 y 16. Sin embargo, en SOM se puede deducir que estas también están relacionadas con las 12 y 15 horas. En el caso de los liceos, es más marcada la diferencia entre los dos grupos, aunque de igual manera puede verse que la hora 11 tiene un comportamiento ligeramente distinto a las 9 y 10 horas.

Para ver el comportamiento de las features en SOM, se realizan las Figuras 5.6(a) y 5.6(b), para los datos de escuelas y liceos, respectivamente. En estas figuras se representa, a través de un mapa de calor, cómo se activaron las neuronas de SOM teniendo en cuenta el peso de cada feature. En el caso de las escuelas, al observar el mapa de calor correspondiente, puede verse que la zona de mayor activación de *snapchat* se contrapone con la de *twitter*. Este resultado es el mismo que observó con PCA por medio de la ortogonalidad de los vectores *snapchat* y *twitter*. Además, las horas que se encuentran en las zonas de mayor activación de estas dos features coinciden con las horas hacia donde están orientados estos mismos vectores en el PCA.

En el mapa de calor del conjunto de datos de los liceos, se puede apreciar el mismo efecto en donde las features de *microsoft* y *skype*, tienen mayor activación en la neurona que se activa para la hora 11. Y a su vez, la zona de mayor activación de estas dos features se contrapone que las del resto. Este comportamiento ya se había visto en el PCA correspondiente, a través de la ortogonalidad de estos vectores con el resto de features.

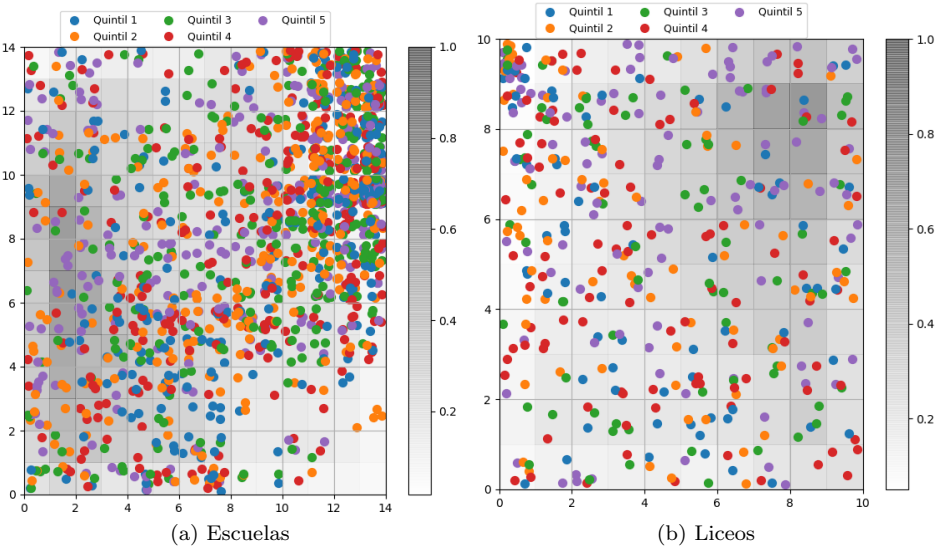
En las Figuras 5.7(a) y 5.7(b), se muestran la distribución de los centros educativos en el mapa de neuronas del SOM, tanto para escuelas como para liceos respectivamente. Cada centro educativo se muestra con un color haciendo referencia al quintil al cual pertenece. En ambos casos puede observarse que hay una zona de neuronas en una escala de grises mayor al resto, lo cual indica que hay una división de los centros educativos en dos grupos.

En cuanto a la relación entre los centros educativos y los quintiles, como puede



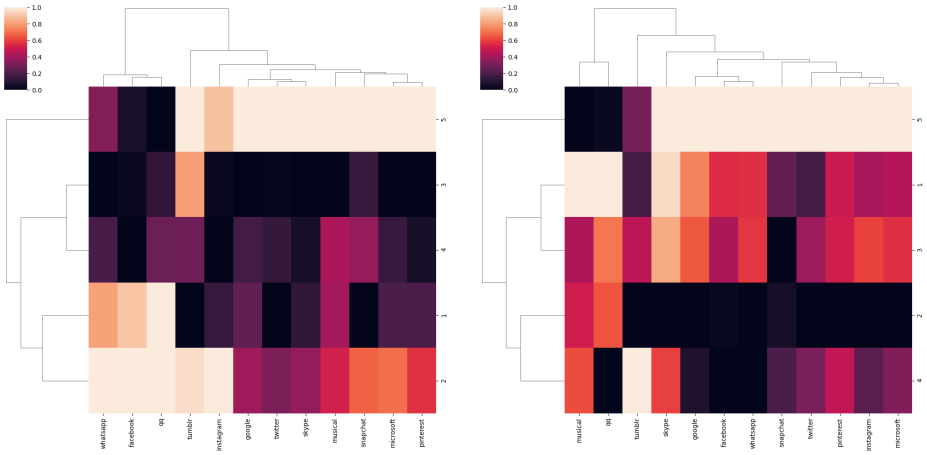
(a) Escuelas (b) Liceos

Figura 5.6: Mapa de calor de la activación de cada neurona para cada *feature*.



(a) Escuelas (b) Liceos

Figura 5.7: Mapa de distanciamiento de neuronas del *SOM* por centro educativo y quintil.



(a) Escuelas (b) Liceos

Figura 5.8: Mapa de calor por quintil.

verse en las figuras, existe una distribución variada y no puede afirmarse plenamente que los grupos que muestra SOM correspondan a un quintil en particular. Para ver un poco mejor el comportamiento de los quintiles con respecto a los dominios, se tiene las Figuras 5.8(a) y 5.8(b), en donde a través de un mapa de calor se muestra el comportamiento de los quintiles con respecto a los dominios.

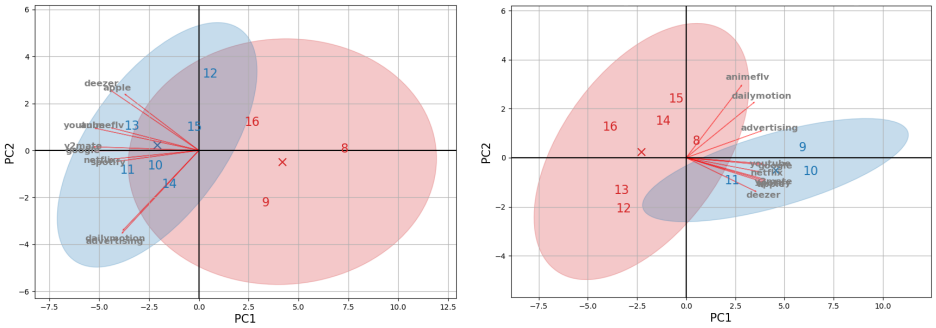
En ambos casos (escuelas y liceos), hay una actividad predominante del quintil 5. En el caso de las escuelas también hay cierta actividad importante de los quintiles 2 y 1, aunque en estos casos la actividad está relacionada con ciertos dominios, entre los que se destaca *instagram*, *whatsapp* y *facebook*. En cuanto a los liceos, existe una distribución más equitativa entre los quintiles, exceptuando a los quintiles 5 y 2, en donde por un lado se tiene al quintil 5 con una alta actividad y por otro el quintil 2 con muy escasa actividad.

5.2.2. Dominios dentro de la categoría *streaming*

Al igual que en el caso anterior, primero se ejecuta PCA y se utiliza k = 2, como número de clusters para k-means, obtenido del promedio de los valores de Silhouette.

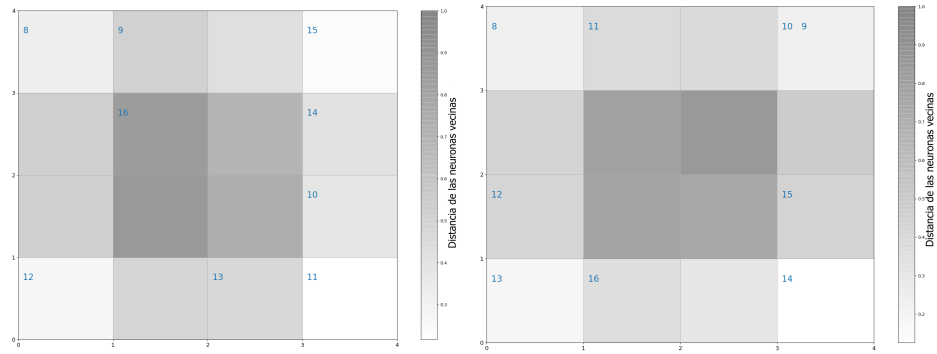
En ambos casos, para el conjunto de datos de las escuelas (Figura 5.9(a)) y el de los liceos (Figura 5.9(b)), los resultados obtenidos son muy similares al del análisis de los dominios de *social networks*, en cuanto a los *clusters* generados y las horas que componen a cada *cluster*. En particular, viendo más precisamente los resultados del set de datos de los liceos, se puede observar más notoriamente cómo las horas varían en el PC1, quedando los horarios de la tarde por un lado y los de la mañana por otro.

En cuanto a los vectores del PCA, en el caso de las escuelas, de similar manera que en el caso del análisis de *social networks*, los vectores se agrupan en tres grupos orientados a horas distintas. Por un lado, se tiene *animeFlv* y *deezer* hacia las últimas horas de la tarde, *netflix* y *youtube* en las horas del mediodía y por último *spotify* y *dailymotion* orientado hacía las horas de la mañana. En el caso del set de datos de los liceos, grosso modo, se pueden definir dos grandes grupos de vectores, los que tienen más peso en las últimas horas de la tarde, como *animeFlv* y *dailymotion*, y los orientados a las horas de la mañana principalmente, como *spotify*, *netflix* y *youtube*.



(a) Escuelas (b) Liceos

Figura 5.9: Biplot de PCA para identificar el comportamiento de los estudiantes durante las horas del día.



(a) Escuelas (b) Liceos

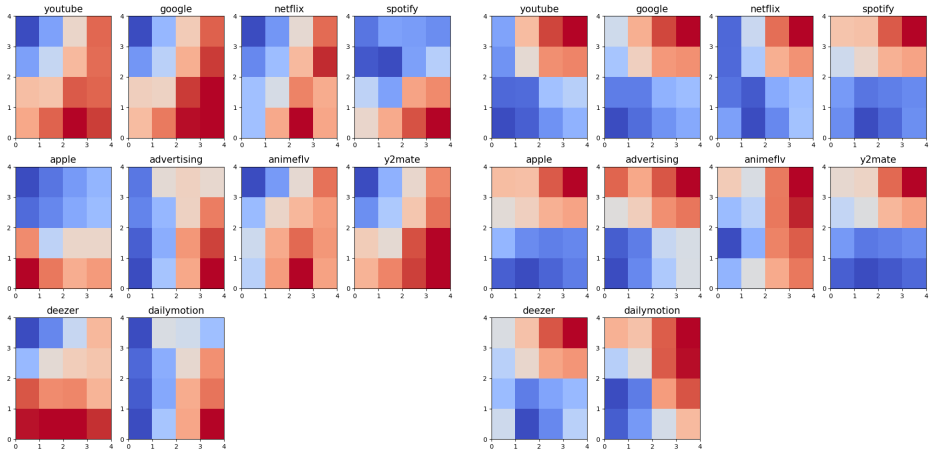
Figura 5.10: Mapa de distanciamiento de neuronas del SOM durante las horas del día.

Con base en estos resultados, se visualiza como el contenido de *streaming* varía dependiendo del horario y del tipo de centro educativo, en el caso de los liceos esto es más notorio. Como dato particular, en el caso de las escuelas, se da la característica de que los vectores de *deezer* y *dailymotion* son casi ortogonales, demostrando su independencia.

Con los resultados de SOM, para el conjunto de datos de las escuelas (Figura 5.10(a)) y de los liceos (Figura 5.10(b)), se presenta algo similar a lo ocurrido con PCA, en donde los resultados obtenidos se asemejan mucho a los resultados de los dominios de *social networks*. Esta similitud indica que los usuarios tienen un comportamiento similar para los dominios de estas dos categorías.

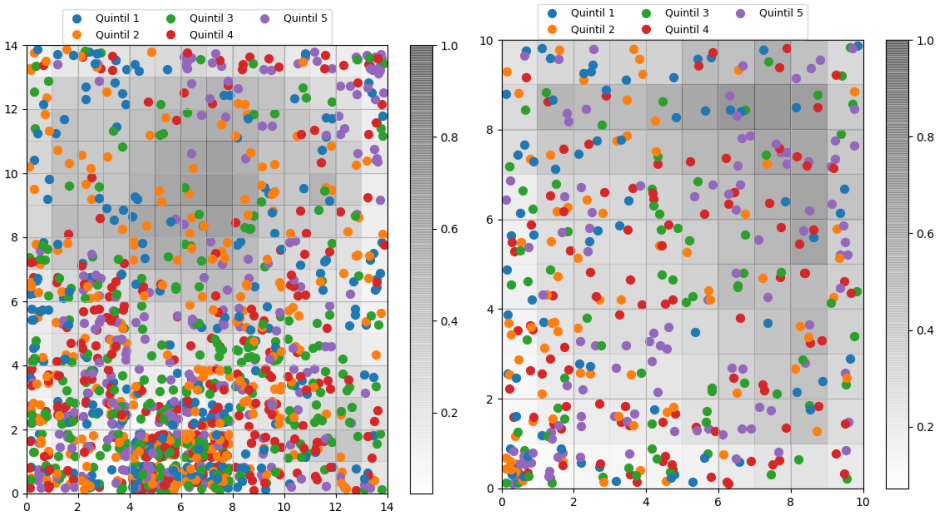
De igual manera que se hizo para *social networks*, se utilizan mapas de calor para ver el comportamiento de las features en SOM, tanto para escuelas (Figura 5.11(a)) como para liceos (Figura 5.11(b)). En estas figuras se puede observar lo mencionado anteriormente con el análisis de los vectores del PCA, en donde la zona de activación de cada *feature* se corresponde con las neuronas que se activaron para las mismas horas hacia donde esas mismas *features* están orientadas en el PCA.

En las Figuras 5.12(a) y 5.12(b), se muestran la distribución de los centros educa-



(a) Escuelas (b) Liceos

Figura 5.11: Mapa de calor de la activación de cada neurona para cada *feature*.

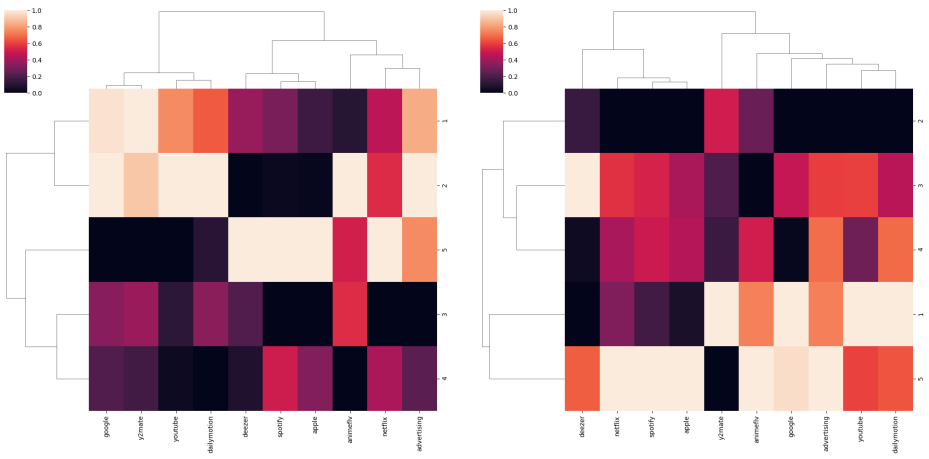


(a) Escuelas (b) Liceos

Figura 5.12: Mapa de distanciamiento de neuronas del *SOM* por centro educativo y quintil.

tivos en el mapa de neuronas del SOM, tanto para escuelas como para liceos respectivamente. Al igual que lo ocurrido con el mismo análisis con los dominios de *social networks*, existe una zona gris que divide el mapa en dos, también puede verse que existe una diferencia notoria sobre la cantidad de centros que componen los grupos.

En este caso tampoco puede apreciarse una relación directa entre el centro educativo y el quintil asociado. En cuanto al comportamiento de los quintiles con respecto a los dominios, se tiene las Figuras 5.13(a) y 5.13(b), para ver la actividad en los distintos quintiles para escuelas y liceos, respectivamente.



(a) Escuelas (b) Liceos

Figura 5.13: Mapa de calor por quintil.

En el caso de las escuelas, se aprecia que hay cierto predomino de la actividad por parte de los quintiles 1, 2 y 5. Este resultado tiene cierta similitud con lo visto anteriormente en la Figura (5.8(a)), denotando la similitud que existe entre estas dos categorías. Esta misma observación puede verse en los liceos, en donde existe cierta oposición, en cuanto a la actividad en los dominios, de los quintiles 2 y 5, lo cual ya se resaltó en el estudio de los dominios de *social networks*.

5.2.3. Dominios dentro de la categoría *educational institutions*

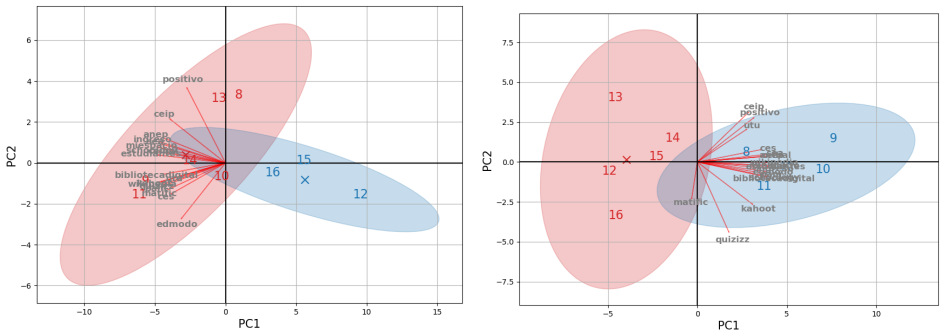
Del promedio de los valores de Silhouette, se utiliza k = 2 como número de clusters para k-means, tanto para el set de datos de las escuelas como el de los liceos. Además, se ejecuta PCA para ambos sets de datos.

Los resultados obtenidos de ejecutar PCA y k-means para el conjunto de datos de las escuelas y para el de los liceos, se pueden visualizar en las Figuras 5.14(a) y 5.14(b), respectivamente. En este análisis, se visualizaron comportamientos distintos a los que se obtuvieron con los análisis anteriores. Si bien en ambos casos se presentaron dos *clusters*, en el caso de las escuelas se resalta la inclusión de la hora 12 en el *cluster* que compone las horas de baja actividad de la red (últimas horas de la tarde). Esta es la hora de cambio de turno, por lo que no hay actividad escolar y hasta ahora no se había visto explícito este resultado en los análisis anteriores.

Cabe recordar que, en este caso, solo se está analizando los dominios pertenecientes a la categoría *educational_institutions*, los cuales se entiende que son utilizados, mayormente, con fines educativos. Esto explicaría por qué en las escuelas la hora 12 es tomada como una hora de baja actividad.

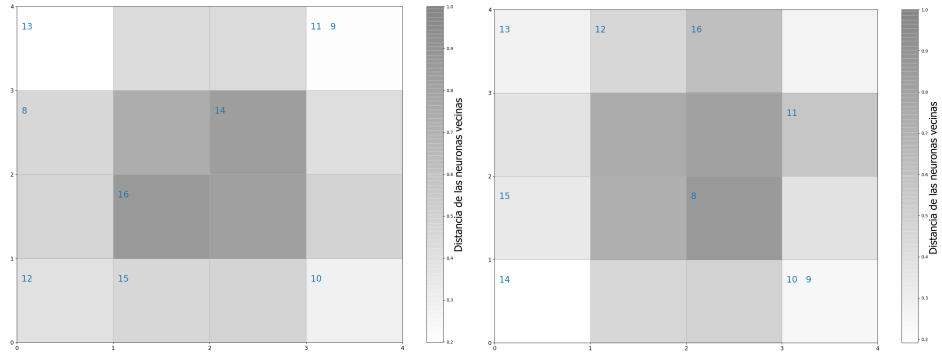
En el caso de los vectores del PCA, al tener una gran cantidad de *features* es difícil ver comportamientos particulares. Sí puede verse un comportamiento grupal de todos los vectores. En el caso de las escuelas, estos están orientados hacia las horas de la mañana y las primeras horas de la tarde, mientras que en el caso de los liceos estos se orientan principalmente hacia las horas de la mañana.

El comportamiento anterior puede ser visible en los resultados de SOM, para el conjunto de datos de las escuelas y para el de los liceos, en las Figuras 5.15(a) y



(a) Escuelas (b) Liceos

Figura 5.14: Biplot de PCA para identificar el comportamiento de los estudiantes durante las horas del día.



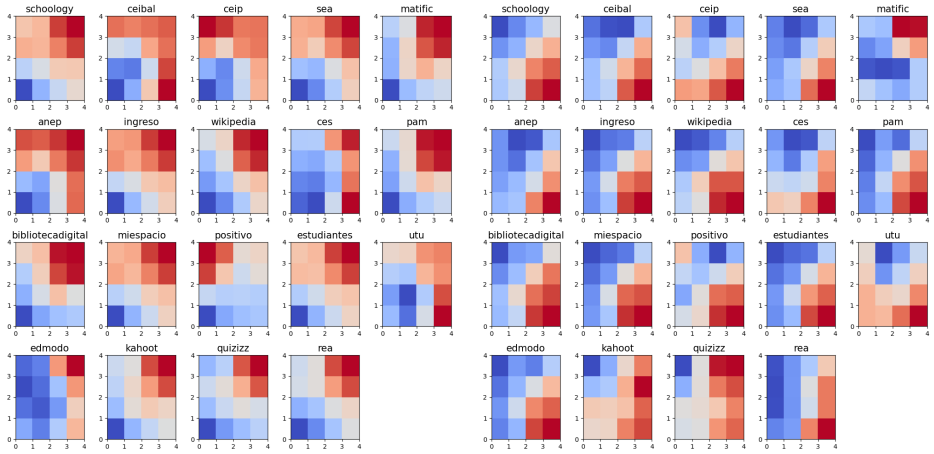
(a) Escuelas (b) Liceos

Figura 5.15: Mapa de distanciamiento de neuronas del SOM durante las horas del día.

5.15(b), respectivamente. En el caso de los liceos, se tiene una zona de neuronas que se activan para las horas 11, 10, 9 y 8. En el caso de las escuelas, si bien se tiene resultados similares a los vistos con k-means, en donde se ve el grupo de horas de menor actividad (12, 15 y 16), el resto de las horas están más dispersas, mostrando que hay un cierto comportamiento en esas horas que no es apreciado con el PCA y k-means.

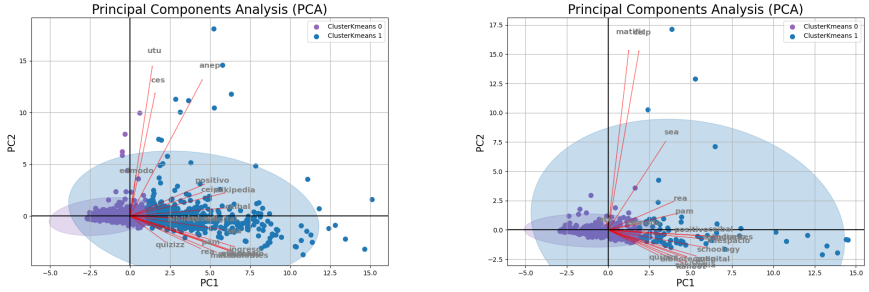
En el mapa de calor, en donde se muestra el peso de cada feature en las neuronas, se puede verificar la observación anterior. Tanto para el caso del set de datos de las escuelas (Figura 5.16(a)), como para el de los liceos 5.16(b)), se aprecia que la gran mayoría de las *features* tienen más actividad en la zona de las neuronas antes mencionada. Esto valida y reafirma los resultados en donde se presenta que las escuelas tienen más actividad en las últimas horas de la mañana y en las primeras de la tarde, mientras que, en los liceos, la mayor actividad se da en las últimas horas de la mañana.

Para observar el comportamiento de los centros educativos durante el año para los dominios de la categoría *educational institutions*, se tienen las Figuras 5.17(a) y 5.17(b), para escuelas y liceos, respectivamente. Para encontrar el número k de clusters, al igual que en los análisis anteriores se utilizó el valor de Silhouette, en ambos casos el mejor valor fue k = 2.



(a) Escuelas (b) Liceos

Figura 5.16: Mapa de calor de la activación de cada neurona para cada *feature*.



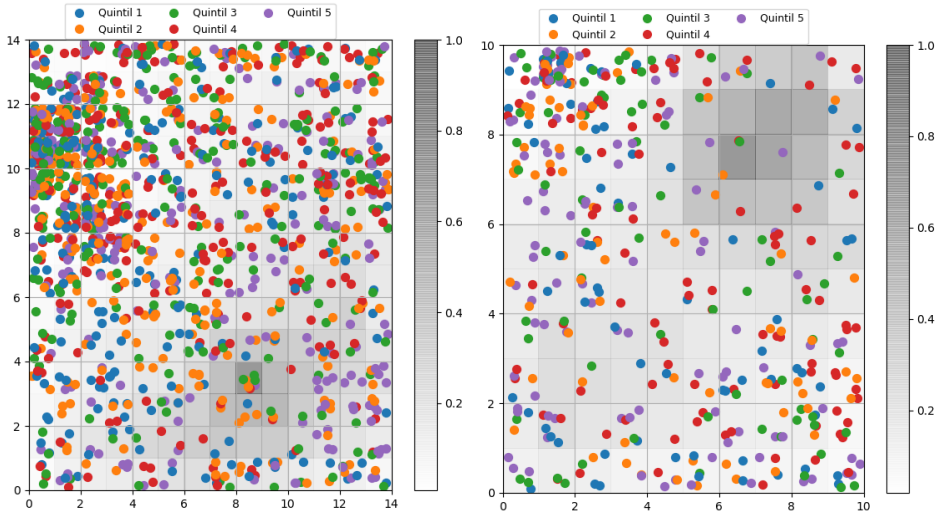
(a) Escuelas (b) Liceos

Figura 5.17: Biplot de PCA para identificar el comportamiento de los centros educativos durante el año.

En ambos casos se ve la formación de dos grupos, los cuales tienen una mayor variación sobre el PC1. Otro punto en común de ambas figuras es la existencia de ciertos centros educativos que tienen una gran variación a lo largo del PC2 y en los cuales puede verse claramente que existen ciertos vectores que se orientan hacia ellos. Estos pequeños grupos de vectores son ortogonales a gran parte del resto, lo cual lleva a deducir que tienen un comportamiento distintivo.

Para ver la distribución de los centros educativos, en el mapa de neuronas de SOM, se tiene la Figura 5.18(a) para las escuelas y la Figura 5.18(b) para los liceos. Nuevamente, se aprecia la división en dos del mapa de neuronas. Esto es notorio en el caso de las escuelas. Sin embargo, en el caso de los liceos, si bien se aprecia una zona con las neuronas en un tono más gris que el resto, denotando su distanciamiento, existe una zona con un gris más claro que podría llegar a interpretarse como un subgrupo.

En este caso, y al igual que en los análisis anteriores, no puede verse con claridad una relación entre los centros educativos y los quintiles. Por el contrario, sí se observan



(a) Escuelas (b) Liceos

Figura 5.18: Mapa de distanciamiento de neuronas de SOM por centro educativo y quintil.

ciertas aglomeraciones de centros educativos en determinados grupos de neuronas, en contraposición con las zonas de un gris más oscuro (neuronas más distantes), en donde la aglomeración de centros es mínima.

5.3. Conclusiones

A partir de análisis de los dominios de cada categoría seleccionada en este capítulo, se puede concluir que las horas de mayor actividad de los dominios seleccionados están fuertemente relacionadas con el horario educativo de cada tipo de centro. Al agregar también la variable quintil socioeconómico de los centros educativos al estudio, se pudo observar la falta de relación entre el uso de la red y el quintil al que pertenece el centro, en otras palabras el uso que le da el usuario a la red no depende tanto del quintil socioeconómico al que pertenece. Otro punto a resaltar es la similitud que existe entre los dominios pertenecientes a las categorías de redes sociales y *streaming*, en cuanto al uso que tienen estos dominios por parte de los usuarios.

Capítulo 6

Utilización de NTOP para predicción de tráfico

Una vez completados los análisis por categoría y dominios, se procede a un último estudio con la finalidad de poder predecir el tráfico en la red a partir de las consultas DNS. A partir de este último estudio se tiene una noción del impacto que tienen ciertas aplicaciones en la red a partir, simplemente, de la cantidad de consultas que se registran. Para esto se utiliza el registro del consumo de bytes que se tiene en ciertos centros educativos gracias al software NTOP.

NTOP es una herramienta que permite la recopilación y clasificación de datos, tanto de subida como de bajada, en las escuelas. Para ello, se basa en la herramienta nDPI [44], que permite una inspección profunda de los paquetes.

La recopilación de los datos se hace mediante una ventana de tiempo de 5 minutos, por lo que cada registro generado se corresponde con el tráfico de ese periodo de tiempo. Para cada registro NTOP que queda almacenado se tienen los siguientes campos:

- **Timestamp** - La hora a la que comenzó la ventana de tiempo de 5 minutos en UTC.
- **Application** - Aplicación identificada por el clasificador de tráfico NTOP (por ejemplo, YouTube, Facebook).
- **Down-link** - La cantidad de bytes descargados de la aplicación específica durante la ventana de tiempo.
- **Up-link** - La cantidad de bytes subidos de la aplicación específica durante la ventana de tiempo correspondiente de 5 minutos.

Un ejemplo de entrada de un registro NTOP es el siguiente:

“2019-03-10 17:48:41”,“Instagram”, “54321”, “2541”

Lamentablemente, no ha sido posible desplegar el uso de esta herramienta en los centros educativos de todo el país, ya que tiene costos altos de infraestructura y puede ser poco práctico. En principio se ha instalado en solo algunos centros: como se aprecia en la Figura 6.1, aproximadamente el 40 % son escuelas y el resto liceos. Más específicamente se cuenta con 64 escuelas y 43 liceos, que tienen en funcionamiento la herramienta NTOP. Además, únicamente se recopilaron datos de los meses correspondientes a septiembre, octubre, noviembre y diciembre de 2019.

Al igual que ocurre con la plataforma de consultas DNS, la herramienta NTOP guarda el nombre de la aplicación a la cual pertenece la información de cada registro

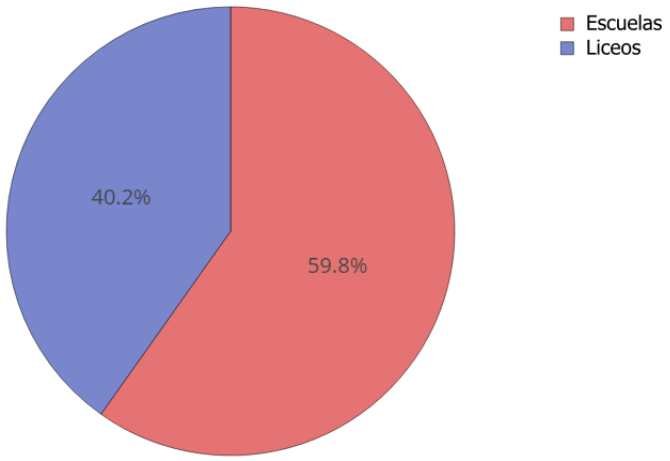


Figura 6.1: Porcentaje del total de registros NTOP por centro educativo.

que almacena. En la Figura 6.2, se muestra el porcentaje de consumo de bytes (*download* + *upload*) utilizado por cada aplicación, de lunes a viernes de 8 a 17 hs. Como puede apreciarse en la imagen anterior, existe un porcentaje de tráfico del cual no se tiene la información correspondiente de la aplicación a la cual pertenece. Esto se da por temas de encriptación de la información, como pueden ser los casos de "SSL", "SSL_No_Cert" y "HTTP".

Otro punto por destacar es que no existe ningún criterio en común entre el sistema NTOP y el sistema de consultas DNS en lo que refiere al *parseo* y a la nomenclatura de los dominios que se extraen en DNS, y la asignación del nombre de aplicación que hace NTOP. Por este motivo se decidió trabajar con una lista reducida de aplicaciones en NTOP, para las cuales resulta relativamente sencillo encontrar su homólogo dentro de los dominios parseados en las consultas DNS. Además, esta selección presenta un interés de estudio para esta tesis. Estas aplicaciones son: *YouTube*, *Google*, *WhatsApp*, *Facebook*, *Instagram*, *Twitter* y *Netflix*.

6.1. Calidad de datos

El despliegue y la implementación del sistema NTOP en los centros educativos se dio de forma paulatina. Es debido a esto que los datos con los que se cuenta de algunos centros educativos son de menor cantidad que de otros. Es decir, de algunos centros se cuenta con datos de setiembre a diciembre, de otros de octubre a diciembre y así sucesivamente.

Por esta razón se decidió hacer un preprocesamiento o filtrado de los centros a analizar, con el fin de tener la mayor cantidad de datos fiables posibles. Para esto se tuvieron en cuenta las consultas DNS y la cantidad de bytes consumidos por los centros educativos.

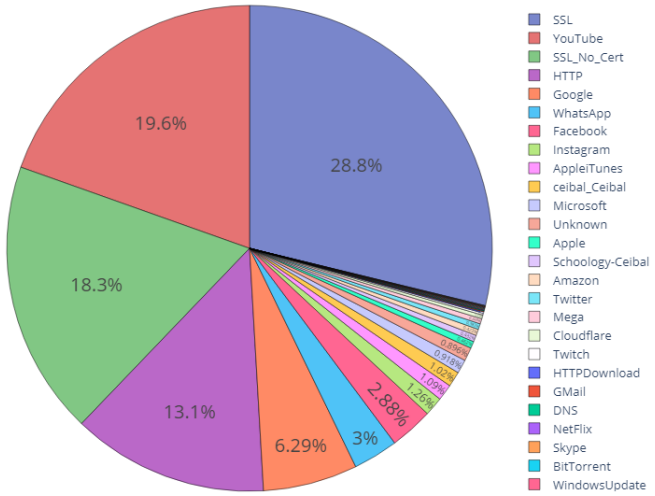


Figura 6.2: Porcentaje del total de tráfico por aplicación. Lunes a viernes, de 8 a 17hs.

Se definen dos filtros principales: primero, por cantidad de consultas DNS, en donde tanto para los liceos como para las escuelas se toma la menor mediana de los 4 meses de los centros educativos en donde no existe NTOP. Este valor se toma como punto de corte para los centros en donde sí hay NTOP. El acumulativo de consultas por mes debe superar este punto para poder ser tomado en cuenta en esta tesis.

En segundo lugar, se realizó el filtro por NTOP. La metodología es muy parecida a la anterior: se toma la menor mediana de los cuatro meses en cuanto a la cantidad de bytes consumidos por los centros educativos que cuentan con NTOP como base. Los centros educativos que pasan por ambos filtros son los que serán estudiados en este capítulo.

Para el caso de las escuelas se toma como punto de corte las 352022 consultas por mes y 481070 consultas por mes para el caso de los liceos. En cuanto a NTOP, para las escuelas se toman como base el consumo de 343059 Mb por mes y 163905 Mb por mes para los liceos. Luego de este filtro, la cantidad de centros que se seleccionan es mucho menor: 3 escuelas y 6 liceos.

6.2. Estadísticas de datos

Luego del filtro de datos que se produce, la cantidad de centros con lo que se trabaja es baja. De todas maneras, se pueden observar ciertos comportamientos estadísticos.

En las Figuras 6.3 y 6.4, se muestran la cantidad de consultas DNS por aplicación, que se obtuvieron tanto para escuelas como para liceos, respectivamente. En el caso de los liceos se puede observar que las curvas descriptas por las aplicaciones se asemejan mucho a las que se muestran en la sección 2.4. Si bien estas últimas describen la cantidad de consultas por mes, el comportamiento durante las horas del día es bastante similar. Esto mismo es más difícil de ver en el caso de las escuelas, debido

principalmente a que solo se tienen datos de 3 de ellas. Se puede apreciar que la aplicación *google* es la predominante en cuanto a cantidad de consultas, en ambos tipos de centros educativos.

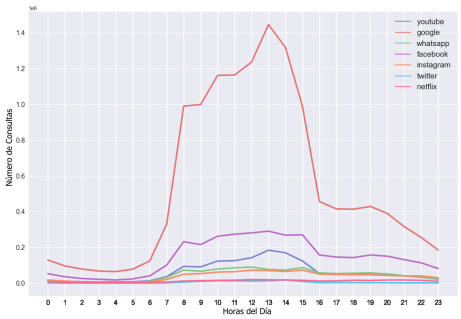


Figura 6.3: Cantidad de consultas por aplicación por cada hora del día en escuelas.

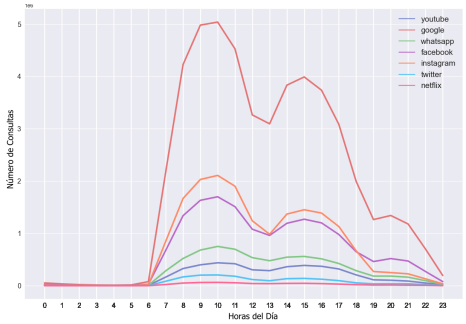


Figura 6.4: Cantidad de consultas por aplicación por cada hora del día en liceos.

El consumo de bytes en los centros estudiados durante las horas del día, se puede ver en las Figuras 6.5 y 6.6, tanto para escuelas como para liceos respectivamente. En este caso, la aplicación predominante en ambos tipos de centros educativos es *youtube*. Este es un resultado esperado, ya que como se vio anteriormente en el capítulo 5 de dominios, esta es la aplicación de *streaming* más utilizada. En el caso de las escuelas, el uso de *youtube* en cuanto a consumo de bytes es mucho mayor que para el resto de las aplicaciones, casi 8 veces mayor.

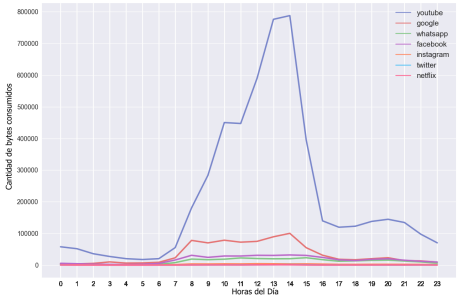


Figura 6.5: Cantidad de bytes consumidos por aplicación por cada hora del día en escuelas.

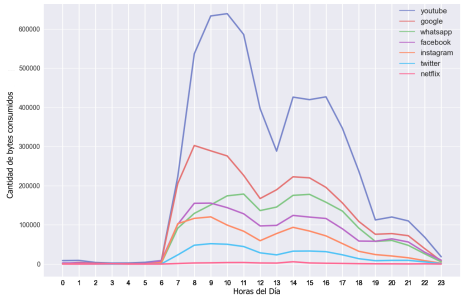


Figura 6.6: Cantidad de bytes consumidos por aplicación por cada hora del día en liceos.

Por último, se tiene en cuenta la cantidad de dispositivos distintos conectados en los centros educativos por hora. Estos se obtienen al considerar que un único dispositivo tiene asignada una única MAC, *Media Access Control*. En las Figuras 6.7 y 6.8, pueden observarse estos valores. En ambos casos, escuelas y liceos, se tienen picos de cantidad de dispositivos distintos conectados sobre las horas del mediodía. Como dato curioso, en el caso de los liceos, se muestra que durante las horas en las que hay más dispositivos distintos conectados (entre las 12 y 13 hs), es también cuando se muestra un descenso del consumo de bytes y de la cantidad de consultas DNS. Esto se puede ver en las imágenes anteriores, denotando que durante esas horas es cuando más dispositivos de conectan, pero que no hay una utilización activa de la red por parte de estos.

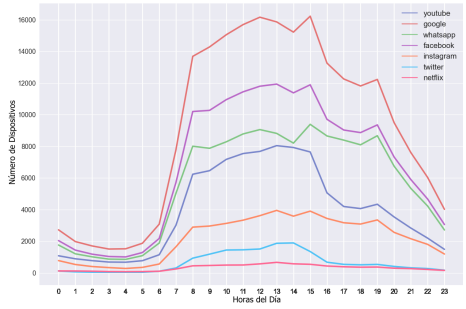


Figura 6.7: Cantidad de dispositivos conectados por aplicación por cada hora del día en escuelas.

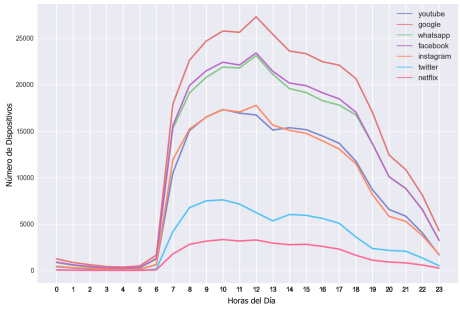


Figura 6.8: Cantidad de dispositivos conectados por aplicación por cada hora del día en liceos.

Tabla 6.1: Ejemplo de matriz de datos NTOP

hour	dayNames	ruee	youtube	google	whatsapp	facebook	instagram	twitter	netflix	throughput
8	Wednesday	1102036	5533	51317	3433	12014	1201	851	514	22666.3
8	Monday	1102036	4974	51396	3491	10774	1212	235	1330	35834.44
8	Friday	1115001	4303	63024	4490	14558	3029	389	393	10136.25
...	...	...	...	...	...	...	...	...	...	...
16	Tuesday	1116118	3912	32658	4268	9621	4792	191	1917	14994.24

6.3. Análisis y resultados

La idea de usar los datos de NTOP en conjunto con los datos de las consultas DNS, es poder crear e investigar un modelo predictor base simple, en donde a partir de consultas DNS se puede estimar la cantidad de tráfico, en Mb, utilizado en la red por ciertas aplicaciones. Para esto se utilizan dos modelos básicos: uno de regresión lineal y el Random Forest, en ambos casos separando el conjunto de datos por centros educativos (escuelas y liceos).

Para esta parte del entrenamiento tanto del modelo lineal como del Random Forest, se trabajó con una matriz basada en la que se muestra en la Tabla 6.1. Las columnas en verde son las empleadas como entradas a los modelos y la columna azul representan el valor a predecir. Esto se aplica tanto para liceos como para escuelas. Se utiliza un *random train set* de 75 % (101 registros de escuelas, 202 de liceos) y el 25 % (34 registros de escuelas, 68 de liceos) restante para *test*.

Para calcular el error y tener una referencia, se relativiza el error cuadrático medio (RMSE) de acuerdo con el promedio de valores de *throughput* con los que se testeó. En otras palabras, se divide el RMSE entre el promedio de valores de *throughput* del testeo.

6.3.1. Regresión Lineal

Escuelas

El modelo predictor lineal utilizado en el conjunto de datos pertenecientes a las escuelas, arrojó un RMSE de alrededor de unos 13Gb, lo que equivale a un 35 % aproximadamente (ver ecuación 6.1) del promedio de megas consumidos por el conjunto de datos de testeo. Si bien es un porcentaje de error alto, cabe resaltar que a causa del filtro de datos que se hizo, se trabajó solamente con datos de tres escuelas. Con esto en cuenta, el error pasa a ser relativamente bajo.

$$RMSE = 12924{,}83MB = 34{,}69\,\% \tag{6.1}$$

En la Figura 6.9 se muestra de forma comparativa el resultado de los coeficientes obtenidos del modelo para cada aplicación. Rápidamente, se puede observar que los coeficientes (y por ende aplicaciones) que más aportan en el aumento del valor de consumo de bytes son *youtube* y *facebook*. Por el contrario, *google* y *whatsapp* son los que más tienen efecto negativo en cuanto el aumento de consumo de bytes, indicando que un aumento en el número de consultas de estas aplicaciones no afecta significativamente el aumento de consumo de bytes.

Por último, en la ecuación 6.2, se muestra la representación matemática que se obtuvo a partir del modelo de regresión lineal.

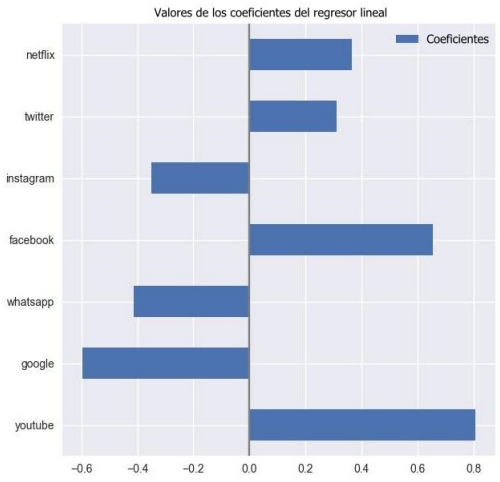


Figura 6.9: Coeficientes del modelo de regresión lineal para escuelas.

$$\begin{aligned} Throughput = {} & 0{,}8083 * youtube - 0{,}5975 * google - 0{,}4131 * whatsapp \\ & + 0{,}6542 * facebook - 0{,}3496 * instagram \\ & + 0{,}3122 * twitter + 0{,}3653 * netflix - 0{,}0119 \end{aligned} \tag{6.2}$$

La Figura 6.10 muestran una comparación del ratio real por hora contra el ratio predicho. Las gráficas no son exactamente iguales debido a que, como ya se mencionó, el modelo tiene un RMSE levemente elevado. Sin embargo, existen similitudes en los valores en las horas matutinas y grandes diferencias en las horas de la tarde.

Esto puede apreciarse mejor en la Figura 6.11, en donde se muestra el consumo de bytes real y el predicho por el modelo específicamente para uno de los centros educativos. Si bien se ven diferencias en líneas generales, el modelo se comportó de una manera correcta prediciendo los valores del consumo de bytes para el centro educativo en concreto.

Liceos

En el caso de los liceos se obtuvo un mejor resultado, en el sentido de que el regresor lineal tuvo un RMSE de alrededor de 6Gb, que equivale a aproximadamente un 15 % (ver ecuación 6.3) del promedio de Mb consumidos por el conjunto de datos de testeo. Este es un error bastante más bajo respecto al que se obtuvo con el conjunto de datos de las escuelas. Cabe recordar que en el caso de liceos se analizaron más del doble de centros educativos que en el conjunto de datos de las escuelas.

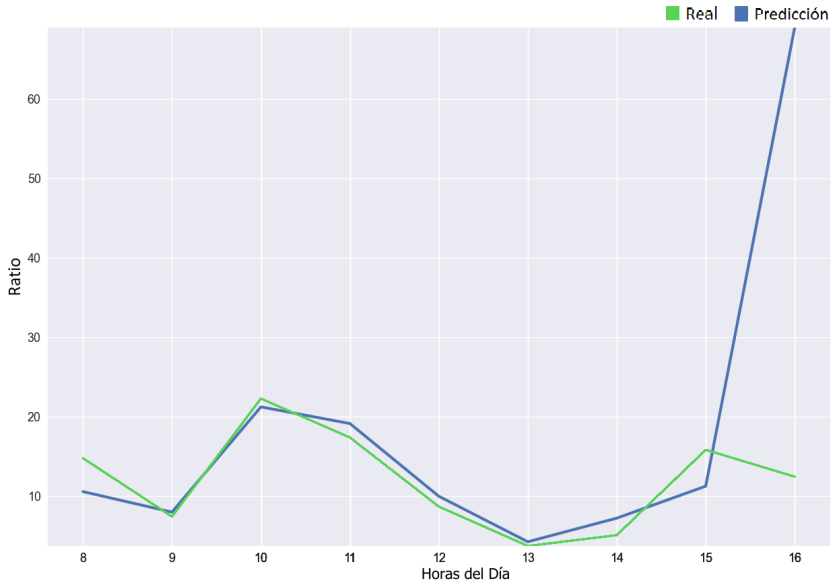


Figura 6.10: Ratio calculado a partir de los datos obtenidos de throughput y de la predicción.

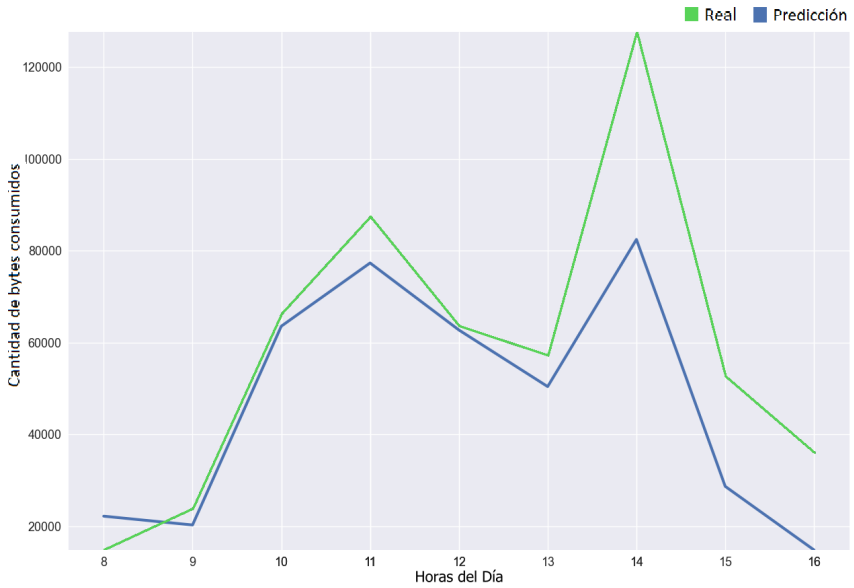


Figura 6.11: Throughput real y predicho para una escuela especifica.

$$RMSE = 5789{,}13MB = 15{,}51\,\% \tag{6.3}$$

En este caso, los coeficientes que resaltan y que están más relacionados con el aumento de consumo de bytes, son los pertenecientes a las aplicaciones de *google* e *instagram*, En tanto que la aplicación que menos incide en el consumo de bytes es *whatsapp*, esto se puede ver en la Figura 6.12.

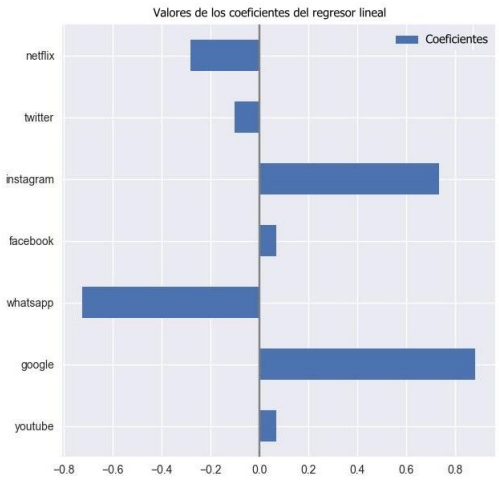


Figura 6.12: Coeficientes del modelo de regresión lineal para liceos.

En la ecuación 6.4, se muestra la representación matemática que se obtuvo a partir del modelo de regresión lineal para liceos.

$$\begin{aligned} Throughput = {} & 0{,}0678 * youtube + 0{,}8840 * google - 0{,}7260 * whatsapp \\ & + 0{,}0687 * facebook + 0{,}7356 * instagram \\ & - 0{,}1011 * twitter - 0{,}2833 * netflix + 0{,}0003 \end{aligned} \tag{6.4}$$

Del mismo modo que en las escuelas, en la Figura 6.13, se muestra una comparación del ratio real con el ratio calculado a partir del volumen de tráfico predicho. Se puede ver que las gráficas son semejantes entre sí. Esta similitud también se ve reflejada en la comparación del consumo de bytes real con el predicho en cada centro. En la Figura 6.14 se muestra un ejemplo, para un liceo en específico, en donde se puede apreciar que la diferencia es mínima.

6.3.2. Random Forest

Escuelas

El otro modelo predictor que se utilizó y con el que se trabajó es Random Forest, el cual tuvo resultados bastante similares a los que se obtuvieron con el modelo de regresión lineal, utilizando una cantidad de 500 árboles de decisión y una profundidad máxima de 3.

Si bien son dos técnicas diferentes, los resultados fueron muy similares (ver ecuación 6.5). Para las escuelas, el RMSE que se obtuvo del Random Forest, fue de aproximadamente 12Gb, que equivalen a un 30 %, aproximadamente, del promedio de consumo de bytes utilizado por los datos de testeo. Random Forest también tiene la particularidad

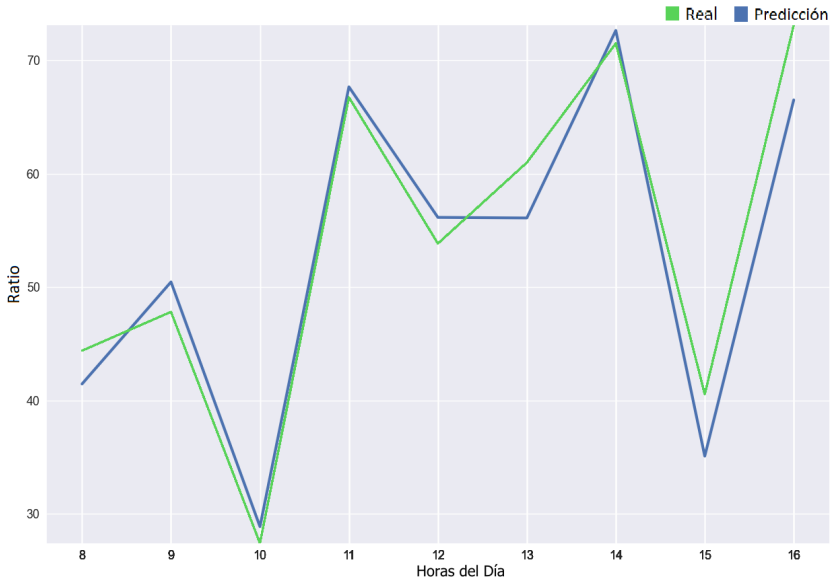


Figura 6.13: Ratio calculado a partir de los datos obtenidos de throughput y de la predicción.

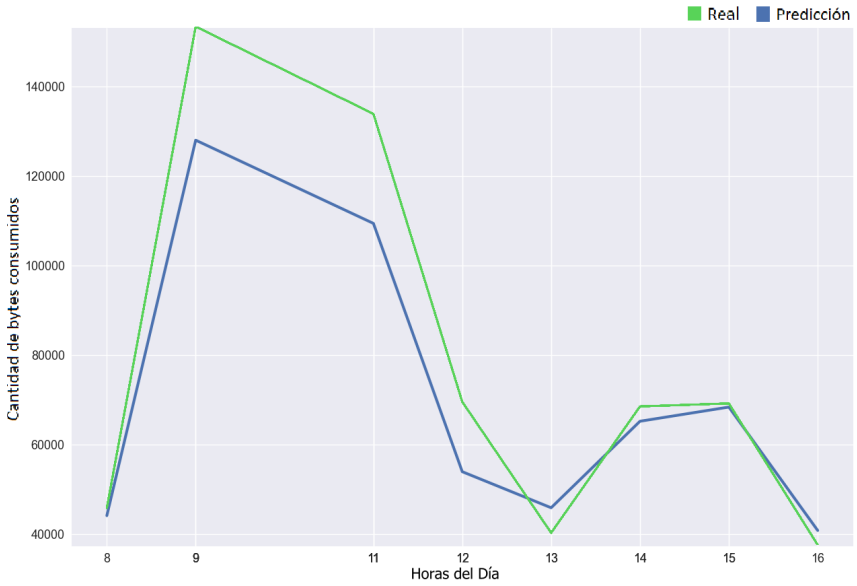


Figura 6.14: Throughput real y predicho para un liceo especifica.

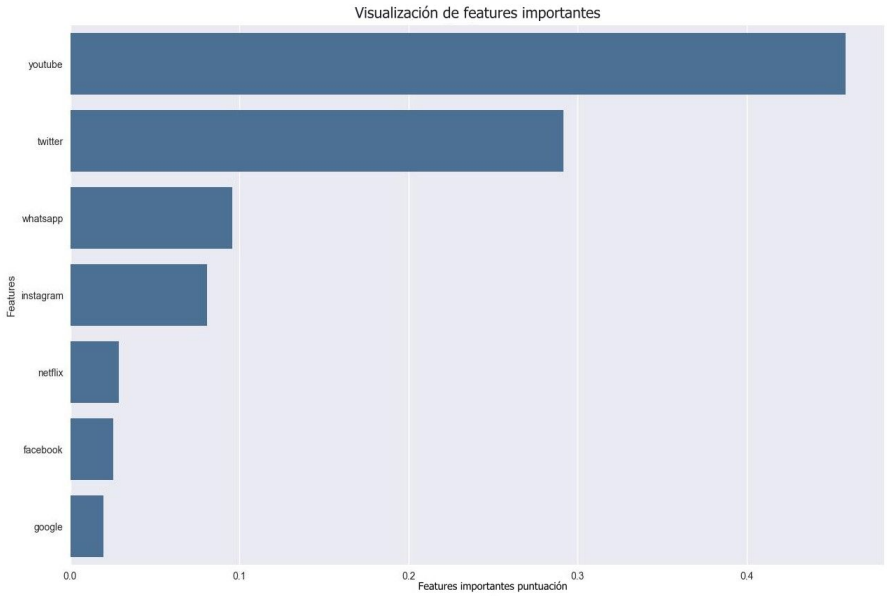


Figura 6.15: Random Forest feature importance en escuelas.

de que puede brindar información adicional, como es el caso de la *feature importance* (ver Figura 6.15). Esta representa la importancia que tiene cada *feature* para el modelo. En este caso se puede ver que *youtube* es la aplicación que más relevancia tienen el modelo para el caso de las escuelas.

$$RMSE = 11678{,}70MB = 30{,}14\,\% \tag{6.5}$$

De igual manera que para el caso con el modelo de regresión lineal, se muestran en la Figura 6.16 la comparación entre el consumo de bytes real y el que se predijo para un centro educativo escolar especifico. En estas se puede apreciar una cierta similitud. La diferencia que sí se encuentra es que los valores de consumo de bytes predichos por el modelo fueron más elevados que los reales en algunas horas en específico.

Liceos

En el caso de los liceos, al igual que ocurrió con el modelo de regresión lineal, los resultados fueron mejores que en el caso de las escuelas.

Utilizando Random Forest, con 400 árboles de decisión y una profundidad máxima de 7, se llegó a un RMSE de aproximadamente de 5Gb, que equivaldrían a un 14.83 % (ver ecuación 6.5) del promedio de bytes consumidos por los liceos del conjunto de datos de testeo.

En cuanto a la importancia de las *features*, se muestran resultados bastante parejos. Aunque la *feature* que más relevancia muestra en el modelo es *youtube* (ver Figura 6.17), las demás también muestran resultados importantes, al contrario de lo que sucede en el caso de las escuelas, en donde hay una sola que predomina sobre las demás.

$$RMSE = 5467{,}97MB = 14{,}83\,\% \tag{6.6}$$

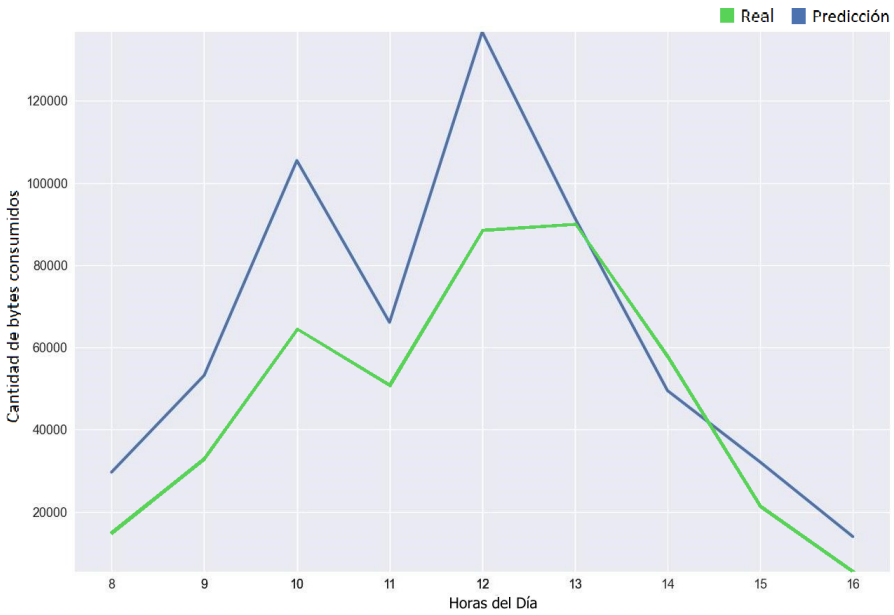


Figura 6.16: Throughput real y predicho por Random Forest para un local escolar.

Para este caso también se compararon el consumo de datos real y el consumo de datos predicho, en la Figura 6.18, para un centro liceal especifico, a modo de ejemplo. En este caso las similitudes son mayores que en el de las escuelas. Esto se puede apreciar mirando directamente a las gráficas.

Por último, se puede apreciar un comportamiento muy similar al hallado en las estadísticas de este capítulo (sección 6.2), en las cuales, para el caso de los liceos, existía un pico de consumo de bytes en las horas de la mañana, decrecía al mediodía, y volvía a subir durante la tarde pero sin alcanzar los valores de la mañana.

6.4. Conclusiones

En este capítulo se presenta el problema de predecir el volumen de tráfico, dependiendo del tipo de centro educativo, a partir de las consultas DNS. Para poder resolver esto, se estudiaron dos modelos bases, uno de regresión lineal simple y otro de Random Forest. No hubo grandes diferencias entre los resultados obtenidos de cada modelo predictor para cada tipo de centro educativo, Obteniendo un RMSE de entre 15 % y 35 % para el caso de liceos y escuelas respectivamente. Si bien el error de predicción es elevado, este disminuye significativamente al considerar mayor cantidad de muestras. Esto se visualiza claramente con el caso de los liceos, que se dispone del doble de centros educativos para el estudio en comparación con las escuelas y se obtiene un mejor RSME como resultado.

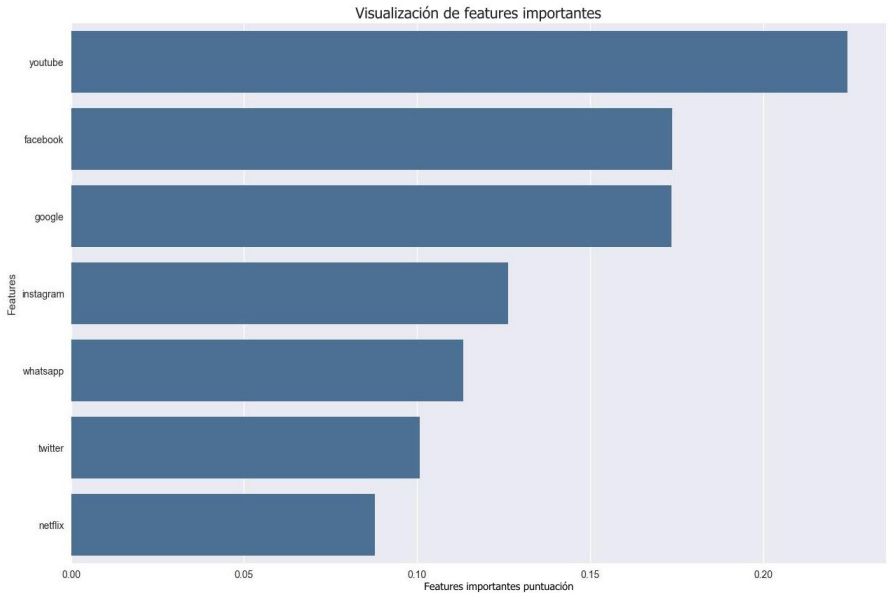


Figura 6.17: Random Forest feature importance en liceos.

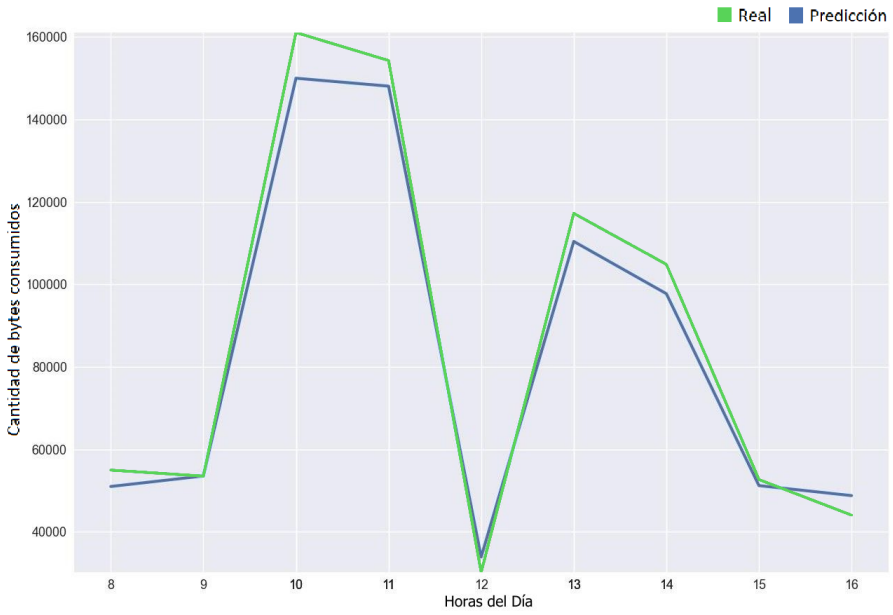


Figura 6.18: Throughput real y predicho por Random Forest para un local liceal.

Capítulo 7

Conclusiones finales y trabajo a futuro

7.1. Conclusiones finales

En esta tesis se realizó un estudio de los usuarios de una red orientada a la educación, compuesta principalmente por estudiantes de entre 6 y 18 años. Este estudio se centró, principalmente, en el análisis de las consultas DNS registradas en la red Plan Ceibal, las cuales fueron recolectadas durante todo el año 2019. También se analizaron datos de consumo de bytes en algunos centros educativos a través de la herramienta NTOP, con el fin de poder llevar a cabo un modelo simple e inicial para la predicción de tráfico a partir de las consultas DNS.

Como punto de partida, se trabajó con las categorías que el sistema Cisco Umbrella asigna a cada consulta. En este dominio de trabajo se efectuaron dos principales estudios con objetivos distintos: el primero, identificar comportamientos diferentes entre estudiantes de primaria y secundaria; el segundo, ya con los resultados a la vista del estudio anterior, tuvo como principal objetivo distinguir el comportamiento de los estudiantes pertenecientes a los diferentes tipos de centros educativos durante las horas de clase. En este último estudio, se observa que los estudiantes de primaria hacen de un mayor uso de la red durante las últimas horas de la mañana y las primeras de la tarde, mientras que los estudiantes de secundaria hacen principal uso de esta durante la mañana.

También se destaca el hecho de que existe cierta oposición en cuanto al uso de las categorías *social networks* y *educational institutions*, para el caso de los estudiantes de primaria. Esta observación no es tan notoria en los estudiantes de secundaria, ya que tienen una mayor actividad de las redes sociales durante el horario de clases.

También se realizaron estos estudios focalizados en los dominios. En ambos casos (categorías y dominios) se procedió con la misma metodología. Primero se hizo un estudio exploratorio de los datos mediante t-SNE. Luego se procedió al análisis de los datos mediante técnicas lineales, como PCA y k-means, y técnicas no-lineales como SOM. Además, se hizo uso de mapas de calor en combinación con *clusters* jerárquicos para la obtención de resultados que no puedan ser vistos por los métodos antes nombrados.

A partir del estudio de dominios, se concluye que las horas de uso más crítico de los dominios trabajados están explícitamente relacionadas con el horario educativo para cada tipo de centro. También es notable la similitud que existe, en términos de

uso, entre los dominios pertenecientes a las categorías de redes sociales y *streaming*. Como punto en común entre las tres categorías de dominios analizadas, se tiene que no presentaron relación alguna entre su uso en los centros educativos y el quintil al que pertenece el mismo.

Finalmente, en conjunto con los datos de NTOP, se logró obtener una predicción del consumo de bytes a partir de la cantidad de consultas DNS para un grupo de aplicaciones seleccionadas. Para poder obtener este resultado, se trabajó con dos modelos bases: Random Forest y regresión lineal. Las predicciones resultantes de cada modelo no presentaron grandes diferencias. Los resultados obtenidos con los centros liceales fueron mejores que con los centros escolares, debido principalmente a que se tenía muchos más datos para entrenar los modelos con el conjunto de datos de los liceos que con el de las escuelas. No obstante, los resultados fueron satisfactorios, teniendo en cuenta que no se buscaba crear un modelo exigente para la predicción de consumo de bytes, sino que el objetivo era obtener un modelo básico y simple.

7.2. Trabajo a futuro

Como es de esperarse cuando se realiza un trabajo de estas características, existen ciertas ramificaciones del objetivo central que pueden ser usadas para futuras investigaciones. A continuación, se detallan algunas de las que se consideraron como más relevantes.

- Mediante el seguimiento de uso y análisis de dominios que son utilizados por los programas educacionales implementados en los centros educativos, se podría estudiar el impacto y la evolución que tienen los mismos programas. Con esto se generaría una herramienta de fácil acceso que genere resultados y datos estadísticos fehacientes en tiempo real, lo que permitiría poder tomar decisiones inmediatas acerca del uso de los programas, en caso de ser necesario.
- A partir del punto anterior, se podría profundizar en la detección de anomalías, pero enfocada en dominios para mantener la seguridad informática en los centros educativos. De esta manera se podrían descubrir plataformas o dominios que son populares entre los usuarios y prever qué impactos podrían tener en la red, en caso de que su uso se masifique para todos los usuarios.
- Otro punto que se puede trabajar es el de considerar la temporalidad de los datos. Es una variable que no se ha tenido en cuenta para esta tesis porque excede los objetivos, pero que puede influir en los resultados. Para esto se puede recurrir al estudio de algoritmos de *clustering* para series temporales.
- También se sugiere investigar la relación entre consultas DNS y el comportamiento real que se muestra en el centro educativo o en el usuario mismo, si es posible. La intención sería saber efectivamente, a partir de las consultas, qué actividad es la que se está realizando con los distintos dominios, como qué tipo de video se está mirando en YouTube, qué clase de información se está buscando en Google, etc.
- En cuanto al trabajo con NTOP, a partir de los resultados obtenidos con el regresor lineal, se puede investigar el uso de modelos más complejos, con redes neuronales, por ejemplo, y poder, de esta manera, mejorar los resultados obtenidos en esta tesis. Esto permitiría tener una herramienta más exacta para predecir el tráfico de la red en cada centro educativo a partir de las consultas DNS. Estos resultados también mejorarían trabajando con una mayor cantidad de datos NTOP.

Apéndice A

Hortonwork Data Plataform (HDP)

En este capítulo se detallarán algunos de los principales componentes de la plataforma, que fueron claves para la realización de este proyecto. No obstante, vale aclarar, que este es un modo de uso que se le dio a la plataforma, por lo que existen muchas herramientas pertenecientes a esta, que no fueron exploradas ni utilizadas.

Hortonworks Data Platform (HDP) [18] es un framework de código abierto para el almacenamiento distribuido y el procesamiento de grandes conjuntos de datos de múltiples fuentes. Que tiene como objetivo facilitar la implementación y administración de clústeres de Hadoop. En comparación con la simple descarga de las diversas bases de código de Apache y tratar de ejecutarlas juntas en un sistema, HDP simplifica enormemente la utilización de Hadoop. HDP permite la implementación ágil de aplicaciones, cargas de trabajo de aprendizaje automático y aprendizaje profundo, almacenamiento de datos en tiempo real y seguridad.

Hortonworks Sandbox [45] es una implementación de un solo nodo de HDP. Se empaqueta como una máquina virtual para que la evaluación y experimentación con HDP sea rápida y fácil. Las funciones de Sandbox están orientadas a explorar cómo HDP puede ayudar a resolver problemas de big data.

En la figura A.1 se puede apreciar, a grandes rasgos, cómo es la arquitectura de HDP. Donde se muestran algunos de los componentes principales que posee la plataforma dependiendo del uso que se le quiera dar.

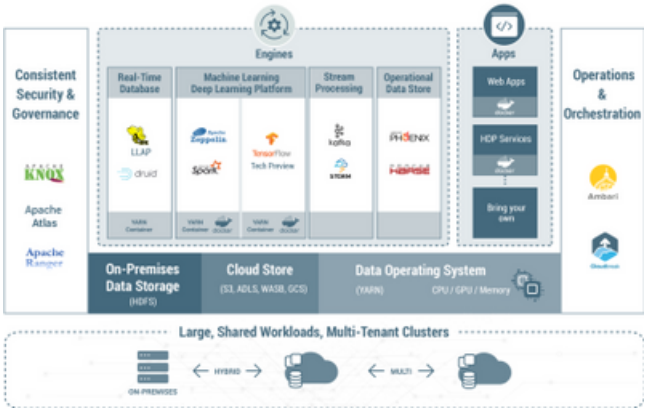


Figura A.1: Arquitectura de HDP v3.1. Fuente: HDP Reference Architecture [46]

A.1. Apache Hadoop

Apache Hadoop es un framework de código abierto para el almacenamiento distribuido y el procesamiento de grandes conjuntos de datos en hardware básico (no potente) [47]. Hadoop permite obtener rápidamente información de grandes cantidades de datos estructurados y no estructurados. Lo que lo transforma en un framework fácil de usar, que puede escalar fácilmente y que brinda tolerancia a fallas y alta disponibilidad para el uso en producción. Los módulos principales y los componentes principales de Hadoop se denominan Hadoop Stack. Juntos, los módulos centrales de Hadoop proporcionan la funcionalidad de trabajo básica para un clúster de Hadoop funcional. A continuación, se ofrece una breve introducción a cada uno de los módulos principales:

- Hadoop Common: contiene bibliotecas y utilidades que necesitan otros módulos de Hadoop.
- Hadoop Distributed File System (HDFS): un sistema de archivos distribuido que almacena datos en máquinas no necesariamente muy potentes (de uso comercial), proporcionando un ancho de banda agregado muy alto en todo el cluster.
- Hadoop YARN: una plataforma de administración de recursos responsable de administrar los recursos informáticos en clusteres y usarlos para programar las aplicaciones de los usuarios.
- Hadoop MapReduce: Un modelo de programación para el procesamiento de datos a gran escala.

A.2. HDFS

HDFS [47] es un sistema de archivos distribuido que está diseñado para almacenar grandes archivos de dato. Se encuentra en la parte superior del sistema de archivos nativo de un sistema operativo. Por ejemplo, HDFS se puede instalar sobre sistemas de archivos ext3, ext4 o XFS para el sistema operativo Ubuntu. HDFS es un sistema de archivos basado en Java que proporciona almacenamiento de datos escalable y confiable, y fue diseñado para abarcar grandes grupos de servidores básicos. Ha demostrado una escalabilidad de producción de hasta 200 PB de almacenamiento, y un solo cluster de 4500 servidores, soportando cerca de mil millones de archivos y bloques. HDFS es un sistema de almacenamiento distribuido, escalable y tolerante a fallas que trabaja en estrecha colaboración con una amplia variedad de aplicaciones concurrentes de acceso a datos, coordinadas por YARN.

Un clúster HDFS se compone de un NameNode, que administra los metadatos del cluster y DataNodes que almacenan los datos. Los archivos y directorios están representados en el NameNode por inodes. Inodes registra atributos como permisos, tiempos de modificación, acceso y espacio en disco.

El contenido del archivo se divide en bloques grandes (típicamente 128 megabytes), y cada bloque del archivo se replica independientemente en múltiples DataNodes (generalmente el factor de réplica es tres, pero puede ser modificable) para brindar redundancia o confiabilidad. Los bloques se almacenan en el sistema de archivos local en los DataNodes.

Namenode supervisa activamente el número de réplicas de un bloque. Cuando se pierde una réplica de un bloque debido a una falla de DataNode o falla de disco, NameNode crea otra réplica del bloque. NameNode mantiene el árbol de espacio de nombres y la asignación de bloques a DataNodes, manteniendo toda la imagen del espacio de nombres en la RAM.

A.3. Apache MapReduce

MapReduce [48], fue introducido por Google como un método para resolver una clase de problemas de petaescala con grandes grupos de máquinas de productos básicos. Es el algoritmo clave que utiliza el motor de procesamiento de datos Hadoop para distribuir el trabajo en un clúster. Un trabajo de MapReduce divide un gran conjunto de datos en fragmentos independientes y los organiza en pares clave y de valor para el procesamiento paralelo. Este procesamiento paralelo mejora la velocidad y la confiabilidad del cluster, devolviendo soluciones más rápidamente y con mayor confiabilidad.

La función Map es un paso inicial de ingestión y transformación en el que los registros de entrada individuales se pueden procesar en paralelo. La función Reduce es el paso de agregación o resumen en el que todos los registros asociados deben ser procesados juntos por una sola entidad.

Una tarea de mapa se puede ejecutar en cualquier nodo informático del cluster y varias tareas de mapa se pueden ejecutar en paralelo en todo el cluster. La tarea de mapa es responsable de transformar los registros de entrada en pares clave/valor. La salida de todos los mapas se particionará y cada partición se ordenará. Habrá una partición para cada tarea de reducción. Las claves ordenadas de cada partición y los valores asociados con las claves son luego procesados por la tarea de reducción. Puede haber varias tareas de reducción que se ejecutan en paralelo en el cluster.

El sistema actual de Apache Hadoop MapReduce está compuesto por JobTracker, que es el maestro, y los esclavos por nodo llamados TaskTrackers. JobTracker es responsable de la gestión de recursos (gestión de los nodos de los trabajadores, es decir, TaskTrackers), el seguimiento del consumo de recursos/disponibilidad y también la gestión del ciclo de vida del trabajo (programación de tareas individuales del trabajo, seguimiento del progreso, proporcionar tolerancia a fallas para tareas, etc.).

A.4. Apache YARN

Hadoop HDFS es la capa de almacenamiento de datos para Hadoop y MapReduce era la capa de procesamiento de datos en Hadoop 1x. Sin embargo, el algoritmo MapReduce, por sí solo, no es suficiente para la gran variedad de casos de uso que se quería resolver.

YARN [47] se introdujo en Hadoop 2.0, como un marco genérico de administración de recursos y aplicaciones distribuidas, mediante el cual se pueden implementar múltiples aplicaciones de procesamiento de datos personalizadas para la tarea en cuestión. MapReduce es ahora solamente una de las aplicaciones de procesamiento de datos distribuidos que se puede usar con YARN.

La idea fundamental de YARN es dividir las dos responsabilidades principales del JobTracker, es decir, la gestión de recursos y la programación/supervisión del trabajo, en demonios separados: un ResourceManager global y un ApplicationMaster (AM) por aplicación. ResourceManager [48] es la máxima autoridad que arbitra recursos entre todas las aplicaciones en el sistema. El ApplicationMaster por aplicación tiene la tarea de negociar recursos desde el ResourceManager y trabajar con los NodeManager para ejecutar y monitorear las tareas componentes. NodeManager es el esclavo por máquina, que es responsable de iniciar los contenedores de las aplicaciones, monitorear el uso de recursos (CPU, memoria, disco, red) e informarlo al ResourceManager.

A.5. Apache Spark

Apache Spark [49, 50] es un framework de programación para procesamiento de datos distribuidos diseñado para ser rápido y de propósito general. Como su propio nombre indica, ha sido desarrollada en el marco del proyecto Apache, lo que garantiza su licencia Open Source.

Apache Spark se basa en Hadoop MapReduce y amplía el modelo de MapReduce para usarlo de manera eficiente para más tipos de cálculos, que incluyen consultas interactivas y procesamiento de flujo. Consta de diferentes APIs (desarrollo en Scala, Java, Python y R) y módulos que permiten que sea utilizado con diferentes propósitos. Desde soporte para análisis interactivo de datos con SQL a la creación de complejos pipelines de machine learning y procesamiento en streaming, todo empleando el mismo motor de procesamiento y las mismas APIs.

Spark no es una versión modificada de Hadoop y, en realidad, no depende de Hadoop porque tiene su propia administración de clusteres. Hadoop es solo una de las formas de implementar Spark. Spark utiliza Hadoop de dos maneras: una es el almacenamiento y la segunda es el procesamiento. Dado que Spark tiene su propio cómputo de administración de clúster, utiliza Hadoop únicamente para fines de almacenamiento.

En la figura A.2 se muestra tres maneras de cómo se puede construir Spark con componentes Hadoop.

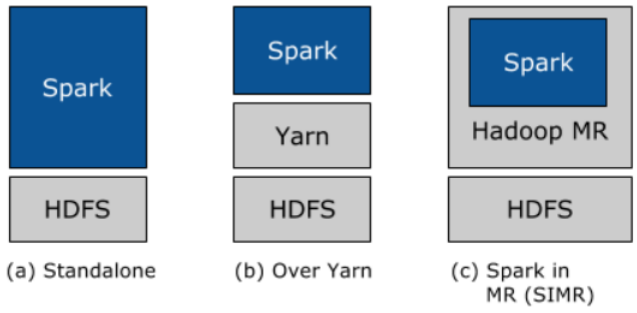


Figura A.2: Tres formas de implementar Spark en un cluster de Hadoop. Fuente: Databricks [51]

A Standalone: se puede asignar estáticamente recursos en todas o un subconjunto de máquinas en un cluster Hadoop y ejecutar Spark junto con Hadoop MapReduce. El usuario puede ejecutar trabajos arbitrarios de Spark en sus datos HDFS. Su simplicidad hace que esta sea la implementación elegida por muchos usuarios de Hadoop 1.x.

B Over Hadoop Yarn: los usuarios de Hadoop que ya han implementado o planean implementar Hadoop Yarn pueden simplemente ejecutar Spark en YARN sin necesidad de preinstalación o acceso administrativo requerido. Esto permite a los usuarios integrar fácilmente Spark en su pila Hadoop y aprovechar toda la potencia de Spark, así como de otros componentes que se ejecutan sobre Spark.

C Spark In MapReduce (SIMR): para los usuarios de Hadoop que aún no ejecutan YARN, otra opción, además de la implementación Standalone, es utilizar SIMR para iniciar trabajos de Spark dentro de MapReduce.

A.5.1. Modelo de ejecución de Spark

La ejecución de la aplicación Spark [50,52] implica conceptos de tiempo de ejecución como controlador (driver), ejecutor (executor), tarea (task), trabajo (job) y etapa (stage). Ver figura A.3. En tiempo de ejecución, una aplicación Spark se asigna a un proceso de controlador único y a un conjunto de procesos de ejecución distribuidos entre los hosts en un clúster.

El proceso del controlador gestiona el flujo de trabajo, programa las tareas y está disponible todo el tiempo que se ejecuta la aplicación. Normalmente, este proceso del controlador es el mismo que el proceso del cliente utilizado para iniciar el trabajo, aunque cuando se ejecuta en YARN, el controlador puede ejecutarse en el cluster.

Los ejecutores son responsables de realizar el trabajo, en forma de tareas, así como de almacenar información en caché. Un ejecutor tiene varios slots para ejecutar tareas.

Invocar una acción dentro de una aplicación Spark desencadena el inicio de un trabajo para cumplirlo. Spark examina el conjunto de datos del que depende esa acción y formula un plan de ejecución. El plan de ejecución ensambla las transformaciones del conjunto de datos en etapas. Una etapa es una colección de tareas que ejecutan el mismo código, cada una en un subconjunto diferente de datos.

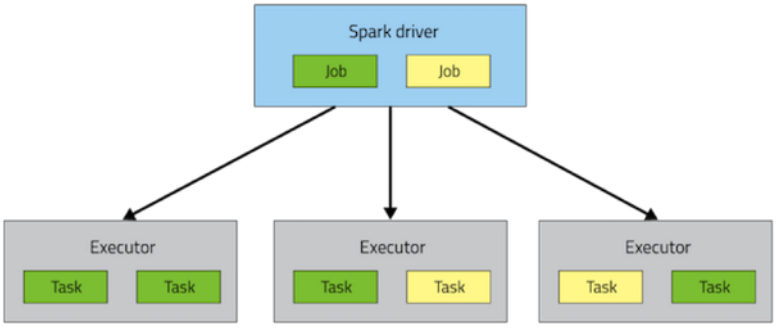


Figura A.3: Modelo de ejecución Spark. Fuente: Clouder Documentation [52]

Apéndice B

Aplicaciones Spark

En un principio se utilizó Zeppelin para la implementación y ejecución de las aplicaciones. Apache Zeppelin [53] es una implementación del concepto de web notebook, centrado en la analítica de datos interactivo mediante lenguajes y tecnologías como Shell, Spark, SparkSQL, Hive, R, etc.

B.1. Configuración de parámetros

Los trabajos (jobs) de Spark utilizan ejecutores (executors), que son aplicaciones que ejecutan tareas (task), que se ejecutan en un nodo del cluster. Los trabajos de Spark se subdividen en tareas que se distribuyen a los ejecutores de acuerdo con el tipo de operaciones y la estructura subyacente de los datos.

Los ejecutores tienen la capacidad de ejecutar múltiples tareas simultáneamente y usar cualquier cantidad de RAM física disponible en un solo nodo. Un ejecutor únicamente puede ejecutarse en un solo nodo (generalmente una sola máquina o VM). La configuración principal está determinada por un conjunto de parámetros:

- *spark.executors.instances*: define el número total de ejecutores disponibles para Spark.
- *spark.executors.cores*: define cuántas CPUs tiene permitido utilizar cada ejecutor. Esto afecta directamente su capacidad multitarea.
- *spark.executors.memory*: define cuánta RAM puede usar cada ejecutor.

En el caso de emplear Zepellin los parámetros se pueden agregar y/o modificar en la definición del intérprete Spark, utilizando los mismos nombres definidos anteriormente. En el caso de usar *spark-submit* es necesario especificar los parámetros en la línea de comando de la ejecución como se muestra en el ejemplo B.1.

```
spark-submit prueba.py \
 --num-executors ? \
 --executor-cores ? \
 --executor-memory ? [...]
```

Ejemplo B.1: Spark-submit definición de parámetros

Estos parámetros están relacionados directamente con la capacidad de hardware disponible para el uso de Spark. En la práctica, se podrían definir valores de los parámetros que no se corresponda con la realidad de recursos que se dispone, aunque obviamente,

esto producirá errores por parte del administrador de recursos (YARN) cuando intente acceder a recursos (por ejemplo memoria) no existentes.

Existen diferentes combinaciones de valores para estos parámetros. Una gran cantidad de ejecutores en general es bueno porque se puede hacer más tareas en paralelo. A su vez, tener más memoria disponible también es algo positivo, al igual que tener más de un núcleo (core) por ejecutor, en particular para el caso del número de núcleos, el número máximo de núcleos que se puede tener es 5. Por el hecho de que HDFS tiene problemas para el procesamiento de muchos hilos concurrentes, por lo que, para lograr un rendimiento de escritura máximo, se recomienda no superar la cantidad de 5 núcleos por ejecutor [54].

El objetivo es encontrar una configuración de parámetros que mejor se adapte a los recursos de hardware disponibles. Y no caer en los extremos (en la jerga se les conoce como *tiny vs fat* ejecutores). Ejecutores *tiny* hace referencia a tener un solo núcleo por ejecutor, lo que significa tener tantos ejecutores por nodo como núcleos disponibles. Los ejecutores *fat*, hace referencia a usar todos los núcleos en un solo nodo para que solo haya un ejecutor por nodo del cluster.

Los pasos [55,56] principales para configurar los parámetros en un cluster de Spark dado el número de nodos del cluster, el número de núcleos por nodo y la cantidad de RAM por nodo:

1 Reservar un núcleo por nodo para YARN/Hadoop.

2 Usar los núcleos restantes como total de núcleos disponibles en el cluster.

3 Establecer *spark.executor.cores* = 5

4 Dividir el total de núcleos disponibles por *spark.executor.cores* para encontrar el número total de ejecutores en el cluster.

5 Reservar un ejecutor para el administrador de la aplicación (reducir el número de ejecutores en uno). Usar el valor resultante para establecer *spark.executor.instances*.

6 Calcular el número de ejecutores por nodo dividiendo el número de ejecutores por el número de nodos en el cluster (redondeando al entero más cercano).

7 Calcular la memoria por ejecutor dividiendo la RAM total del nodo por los ejecutores por nodo.

8 Reducir en un 7 % la memoria del ejecutor para tener en cuenta el *heap overhead* para YARN/Hadoop. Utilizar el número resultante como *spark.executor.memory*.

En el paso 3 se define *spark.executor.cores* = 5, que es el número ideal de núcleos como ya se mencionó anteriormente. Sin embargo, esta definición podría ocasionar que no se utilicen ciertos núcleos en cada nodo, dependiendo de la configuración de hardware que se tenga. En general, las únicas soluciones que nunca dejan ningún núcleo sin usar son tener ejecutores *tiny* o ejecutores *fat*, pero ambas opciones tienen inconvenientes, como ya se mencionó. En la referencia [57] se presenta un algoritmo (ejemplo B.2) en donde el objetivo es encontrar el mejor valor para *spark.executor.cores*:

```
def calc_executor_cores(available_cores):
    executor_cores_max = 5
    if available_cores >= executor_cores_max:
        executor_cores = min(executor_cores_max, available_cores // 2)
    else:
        executor_cores = max(1, available_cores // 2)
    remainder_cores = available_cores % executor_cores
    while remainder_cores > 1 and executor_cores > 2:
        executor_cores -= 1
        remainder_cores = available_cores % executor_cores
```

```
return executor_cores
```

Ejemplo B.2: Definición de *spark.executor.cores*

La función toma el número de núcleos disponibles por nodo (después de eliminar el núcleo que se le asigna a YARN) como entrada y devuelve el valor calculado para *spark.executor.cores*. Sus principales pasos son:

- Inicializar *executor_cores* a un número que pueda ajustarse a los núcleos disponibles.
- Comenzar a reducir *executor_cores* en una unidad hasta que no haya núcleos no utilizados o *executor_cores* = 2.
- Devolver el *executor_cores* resultante.

De esta manera se tiene las herramientas necesarias para poder estimar los tres parámetros necesarios para optimizar el uso de Spark. Por ejemplo, si se tiene como recursos de hardware, un único nodo que cuenta con 25 núcleos y 77 Gb de memoria RAM disponibles para usar por parte de la plataforma. Al aplicar los pasos anteriores para encontrar los parámetros óptimos, tenemos como resultado que:

- *spark.executors.instances* = 6
- *spark.executors.cores* = 4
- *spark.executors.memory* = 11 Gb

No obstante, la configuración de estas variables no garantiza el correcto funcionamiento de Spark. Según las referencias [58, 59] existen problemas al ejecutar aplicaciones *Pyspark*, en el log aparecerá el error *Container killed by YARN for exceeding memory limits.*., por ejemplo.

Cuando se ejecuta *Pyspark*, existen dos procesos: un proceso JVM "on-heap" y un proceso Python "off-heap". El parámetro *spark.executors.memory* define el tamaño del "heap" y ambos procesos están limitados por la memoria del contenedor. Por lo tanto, cuanto más grande sea el "heap", menos memoria tendrá el proceso de Python, y puede alcanzar el límite del contenedor más rápido.

En estos casos se recomienda disminuir el tamaño de *spark.executors.memory*. Otra alternativa es reducir la cantidad de núcleos, para que se ejecuten menos tareas en paralelo dentro del mismo ejecutor. Si bien esto reduce el paralelismo, aumentó la cantidad de memoria disponible para cada tarea lo suficiente como para que sea posible completar el trabajo. De todos modos, en la mayoría de las ocasiones, la configuración de los parámetros depende fuertemente del tipo de aplicación que se está ejecutando y que no existe una única configuración para todas las aplicaciones.

Referencias

[1] Masood Badri, Ali Al Nuaimi, Yang Guang, and Asma Al Rashedi. School performance, social networking effects, and learning of school children: Evidence of reciprocal relationships in abu dhabi. *Telematics and Informatics*, 34(8):1433–1444, 2017.

[2] Alexis Arriola, Marcos Pastorini, Germán Capdehourat, Eduardo Grampín, and Alberto Castro. Large-scale internet user behavior analysis of a nationwide k-12 education network based on dns queries. In *Computational Science and Its Applications – ICCSA 2020*, pages 776–791, Cham, 2020. Springer International Publishing.

[3] One laptop per child. `http://one.laptop.org/`. Última visita: julio 2021.

[4] Creación del proyecto ceibal. `http://www.impo.com.uy/bases/decretos/144-2007/1`. Última visita: julio 2021.

[5] Plan ceibal. `https://www.ceibal.edu.uy/en/institucional`. Última visita: julio 2021.

[6] Umbrella - cisco. `https://umbrella.cisco.com/`. Última visita: junio 2021.

[7] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. A survey on malicious domains detection through dns data analysis. *ACM Comput. Surv.*, 51(4), July 2018.

[8] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi. Detecting internet abuse by analyzing passive dns traffic: A survey of implemented systems. *IEEE Communications Surveys Tutorials*, 20(4):3389–3415, 2018.

[9] Weizhang Ruan, Ying Liu, and Renliang Zhao. Pattern discovery in dns query traffic. *Procedia Computer Science*, 17:80 – 87, 2013. First International Conference on Information Technology and Quantitative Management.

[10] Lin Zuo. Improved svm method applied to the online user behavior analysis. In *2012 IEEE 12th International Conference on Computer and Information Technology*, pages 728–732, 2012.

[11] David Plonka and Paul Barford. Context-aware clustering of dns query traffic. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, page 217–230, New York, NY, USA, 2008. Association for Computing Machinery.

[12] Hongyu Gao, Vinod Yegneswaran, Yan Chen, Phillip Porras, Shalini Ghosh, Jian Jiang, and Haixin Duan. An empirical reexamination of global dns behavior. *SIGCOMM Comput. Commun. Rev.*, 43(4):267–278, August 2013.

[13] K. Schomp, M. Rabinovich, and M. Allman. Towards a model of dns client behavior. In *Passive and Active Measurement (PAM 2016) - Lecture Notes in Computer Science*, volume 9631, pages 1–6. Springer, 2016.

[14] J. Li, X. Ma, L. Guodong, X. Luo, J. Zhang, W. Li, and X. Guan. Can we learn what people are doing from raw dns queries? In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 2240–2248, April 2018.

[15] Z. Jia and Z. Han. Research and analysis of user behavior fingerprint on security situational awareness based on dns log. In *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pages 1–4, Oct 2019.

[16] Observatorio plan ceibal. `https://observatorio.ceibal.edu.uy/`. Última visita: julio 2021.

[17] Cisco umbrella log formats and versioning. `https://docs.umbrella.com/deployment-umbrella/docs/log-formats-and-versioning`. Última visita: junio 2021.

[18] Hortonworks Data Platform. `https://www.cloudera.com/products/hdp.html`. Última visita: julio 2021.

[19] Apache Hadoop. `https://hadoop.apache.org/`. Última visita: abril 2022.

[20] Apache Spark. `https://spark.apache.org/`. Última visita: abril 2022.

[21] Apache Hive. `https://hive.apache.org/`. Última visita: abril 2022.

[22] Repositorio del código implementado en la tesis. `https://gitlab.fing.edu.uy/plan-ceibal/unsupervised-learning`. Última visita: abril 2022.

[23] Public Suffix List. `https://publicsuffix.org/`. Última visita: julio 2021.

[24] Umbrella manage content categories. `https://docs.umbrella.com/deployment-umbrella/docs/content-categories#section-content-categories-definitions`. Última visita: junio 2021.

[25] Angela Gorgoglione, Andrea Gioia, and Vito Iacobellis. A framework for assessing modeling performance and effects of rainfall-catchment-drainage characteristics on nutrient urban runoff in poorly gauged watersheds. *Sustainability*, 11(18), 2019.

[26] Scikit-learn: Machine Learning in Python. `https://scikit-learn.org/`. Última visita: julio 2021.

[27] Seaborn: statistical data visualization. `https://seaborn.pydata.org/`. Última visita: julio 2021.

[28] Giuseppe Vettigli. MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map. `https://github.com/JustGlowing/minisom/`. Última visita: julio 2021.

[29] Sidharth P Mishra, Uttam Sarkar, Subhash Taraphder, Sanjay Datta, Devi P Swain, Reshma Saikhom, and M Laishram. Multivariate statistical data analysis-principal component analysis (pca). *Inernational Journal of Livestock Research*, 7(5):60–78, 2017.

[30] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[31] Mike J Adams. *Chemometrics in analytical spectroscopy*. Royal Society of Chemistry, 2007.

[32] Joaquín Amat Rodrigo. Análisis de componentes principales (principal component analysis, pca) y t-sne. `https://www.cienciadedatos.net/documentos/35_principal_component_analysis`. Última visita: julio 2021.

[33] Joaquín Amat Rodrigo. Clustering y heatmaps: aprendizaje no supervisado. https://www.cienciadedatos.net/documentos/37_clustering_y_heatmaps. Última visita: julio 2021.

[34] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[35] Frank B. Baker and Lawrence J. Hubert. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, 70(349):31–38, 1975.

[36] Joaquín Amat Rodrigo. Clustering con python. https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html. Última visita: julio 2021.

[37] Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.

[38] SAURABH JAJU. Comprehensive Guide on t-SNE algorithm with implementation in R & Python. https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/. Última visita: julio 2021.

[39] Joaquín Amat Rodrigo. T-sne algoritm. https://www.cienciadedatos.net/documentos/35_principal_component_analysis#t-SNE. Última visita: julio 2021.

[40] SOM implementation in SOM Toolbox. http://www.cis.hut.fi/projects/somtoolbox/documentation/somalg.shtml. Última visita: noviembre 2020.

[41] Angela Gorgoglione, Alberto Castro, Andrea Gioia, and Vito Iacobellis. Application of the self-organizing map (som) to characterize nutrient urban runoff. In *Computational Science and Its Applications*, Lecture Notes in Computer Science. Springer, 2020.

[42] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[43] Tony Yiu. Understanding random forest. https://towardsdatascience.com/understanding-random-forest-58381e0602d2. Última visita: julio 2021.

[44] ndpi. https://github.com/ntop/nDPI. Última visita: junio 2021.

[45] Getting Started with HDP Sandbox. https://www.cloudera.com/tutorials/getting-started-with-hdp-sandbox/1.html. Última visita: julio 2021.

[46] Hortonworks Data Platform: Managing Data Operating System. https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.5/data-operating-system/content/apache_yarn.html. Última visita: julio 2021.

[47] Bhushan Lakhe. *Introducing Hadoop*, pages 19–35. Apress, Berkeley, CA, 2014.

[48] Sameer Wadkar and Madhu Siddalingaiah. *Hadoop Concepts*, pages 11–30. Apress, Berkeley, CA, 2014.

[49] Apache Spark - Introduction. https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm. Última visita: julio 2021.

[50] Raul Estrada and Isaac Ruiz. *The Engine: Apache Spark*, pages 97–130. Apress, Berkeley, CA, 2016.

[51] Apache Spark and Hadoop: Working Together. https://databricks.com/blog/2014/01/21/spark-and-hadoop.html. Última visita: julio 2021.

[52] Spark Application Overview - Spark Execution Model. https://docs.cloudera.com/documentation/enterprise/5-8-x/topics/cdh_ig_spark_apps.html#spark_exec_model. Última visita: julio 2021.

[53] Apache Zeppelin. https://zeppelin.apache.org/. Última visita: julio 2021.

[54] Cloudera. Tuning spark applications. https://docs.cloudera.com/documentation/enterprise/5-8-x/topics/admin_spark_tuning.html#spark_tuning__spark_tuning_resource_allocation. Última visita: julio 2021.

[55] Calculating executor memory, number of executors & cores per executor for a spark application. http://www.mycloudplace.com/calculating-executor-memory-number-of-executors-cores-per-executor-for-a-spark-application/. Última visita: julio 2021.

[56] Distribution of executors, cores and memory for a spark application running in yarn. https://spoddutur.github.io/spark-notes/distribution_of_executors_cores_and_memory_for_spark_application.html. Última visita: julio 2021.

[57] Matteo Guzzo. How to optimize your spark jobs. https://matteoguzzo.com/blog/spark-configurator/. Última visita: julio 2021.

[58] Stefano Meschiari. Debugging apache spark pipelines. https://duo.com/labs/tech-notes/debugging-apache-spark-pipelines. Última visita: julio 2021.

[59] Adir Mashiach. Apache spark: 5 performance optimization tips. https://medium.com/@adirmashiach/apache-spark-5-performance-optimization-tips-4d85ae7ac0e3. Última visita: julio 2021.

Índice de tablas

Índice de figuras

Esta es la última página.
Compilado el domingo 18 diciembre, 2022.
https://www.fing.edu.uy/inco