UNIVERSIDAD
TORCUATO DI TELLA

Master in Management & Analytics

“Audiences Marketplace + Infonomics:

How well can Machine Learning predict web user demographics?”

Daniela Longás

Director: **Pablo Roccatagliata**

Junio, 2021

RESUMEN

El presente trabajo propone un modelo de machine learning para asignar el campo género a usuarios de dispositivos que formarán parte de diversas audiencias de usuarios a ser impactadas por campañas publicitarias digitales, en las que el atributo demográfico toma un lugar fundamental. El atributo género es un requisito en gran parte de las audiencias ofrecidas. La empresa creadora de audiencias en la que se basa este trabajo recibe información sobre datos demográficos de pocos usuarios, por lo que predecir el género de usuarios de dispositivos de los que no se tengan registros del género será el principal objetivo.

El primer modelo se entrena con características de User Agent, es decir atributos como marca y modelo del dispositivo, navegador, sistema operativo y versión del sistema operativo. Un segundo modelo sumará a los atributos del primero características de dominios visitados por cada dispositivo. Finalmente, un tercer modelo sumará atributos de urls o sitios web específicos, visitados por los usuarios para asignarles género lo más certeramente posible. En todos los casos se intentarán técnicas de modelos de ensamble, como random forest y xgboost, al igual que regresión logística regularizada.

En base al output de este ejercicio, se logrará aumentar el volumen de las audiencias ofrecidas a distintas agencias de publicidad o empresas, y por ende, los ingresos de la compañía generadora de dichas audiencias, en base al enfoque EVI (Economic Value of Information).

Palabras clave: audiencias, género, demográfico, machine learning, random forest, xgboost, sitios web, dominios, user agent, compra programática.

ABSTRACT

The present work proposes a machine learning model to assign gender to device users that will be part of different user audiences to be impacted by digital advertising campaigns, in which the demographic attribute takes a fundamental place. In most of the audiences offered, the gender attribute is required. The audience-creating company on which this work is based receives information about the demographic data of few users, so predicting the gender of device's users for which the data is not available will be the main goal.

A first model will be trained by taking *User Agent* characteristics. That is, attributes such as the brand and model of the device, browser, operative system and version of the operative system. A second model will add to the attributes of the first, characteristics of *domains* visited by each device. Finally, a third model will add attributes of specific *urls* or websites visited by users to assign them gender as accurately as possible. In all cases, assembly model techniques such as random forest and xgboost will be tried. Also, regularized logistic regression.

Having this final model done, it will be possible to increase the volume of audiences offered to different advertising agencies or companies, and therefore, the income of the company that generates such audiences, based on the EVI (Economic Value of Information).

Keywords: audiences, gender, demographic, machine learning, random forest, xgboost, websites, domains, user agent, programmatic.

Índice

1. Introducción

1.2. El mercado de Audiencias

En este proyecto se trabajará con datos de una empresa creadora de audiencias en la industria de marketing digital. Se abordará el campo de la compra de espacios publicitarios en medios de comunicación digital (conocidos como "publishers") por parte de agencias de publicidad o marcas, o dicho de otro modo, "anunciantes" que demandan estos espacios para llegar, mediante campañas publicitarias, al consumidor correcto. En esta transacción se venden y compran cantidades de impresiones mediante el esquema de *compra programática*.

En un mercado con un gran número de publishers, se incorporan los Demand-Side Platform (DSP), que son plataformas que representan a los anunciantes o agencias de publicidad. Los DSP permiten a los anunciantes comprar espacios publicitarios de manera centralizada, siempre tratando de beneficiarlos haciendo la compra de impresiones al menor precio a pagar posible. Es decir que una agencia o anunciante puede entrar a DSP y comprar inventario publicitario de forma agregada desde un mismo lugar. En el otro lado del mercado aparecen los Supply-Side Platform (SSP) que son plataformas que agrupan a los publishers y se conectan con los DSP para ofrecerles subgrupos de medios digitales agregados para que puedan comprar espacios de manera agrupada. Estos SSP representan al publisher e intentan que éste venda su espacio publicitario al mejor precio posible, así como los DPS pretenden que los anunciantes compren impresiones al precio más bajo posible. Surge entonces el *ad exchange*, que nuclea a los DSP y SSP , los cuales agregan inventario de miles de medios y anunciantes. En el ad exchange se realizan subastas para asignar audiencias a anunciantes. El ad exchange logra que esa subasta sea en tiempo real, gracias al sistema Real Time Bidding (RTB). En este mercado la compra de un anuncio se realiza al mismo tiempo que un visitante carga un sitio web.

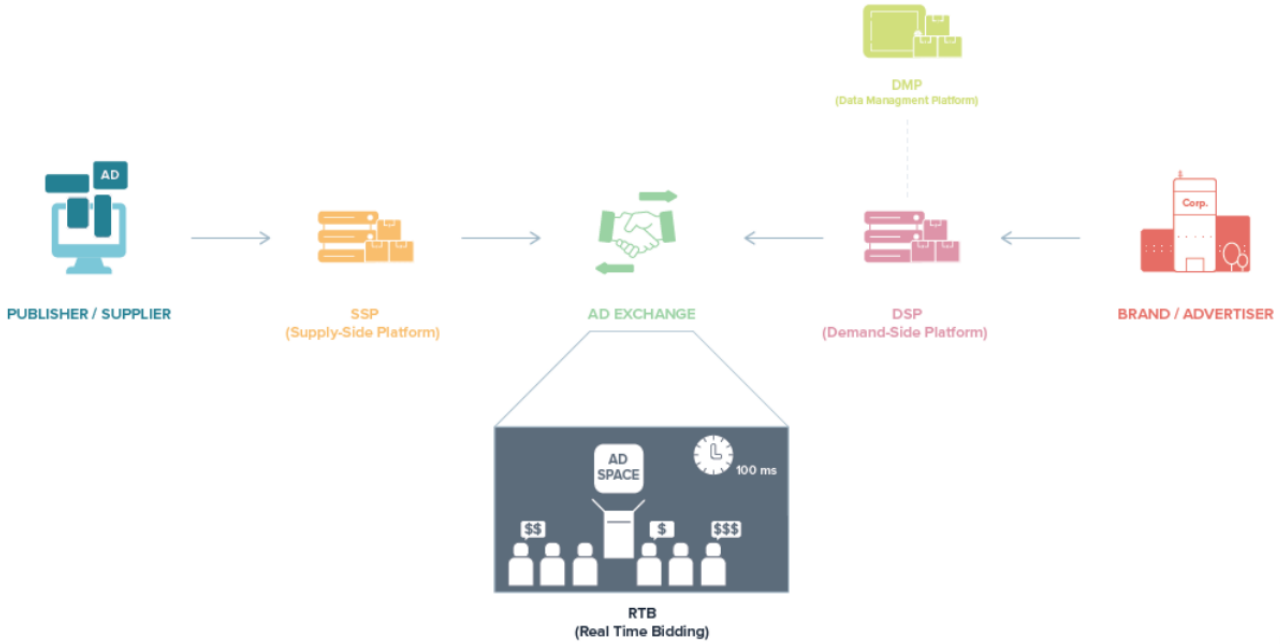


Figura 1: "Programatic Ecosistem" [1]

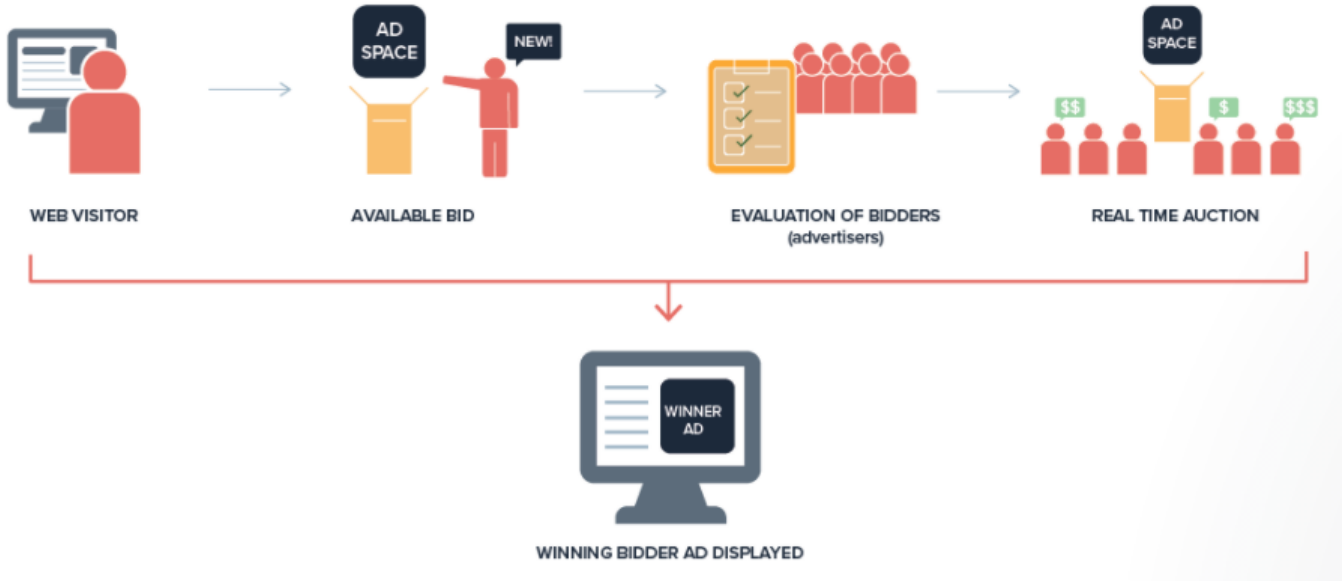


Figura 2: "How an Ad Exchange Work" [2]

El mercado de compra programática viene creciendo a pasos agigantados a nivel mundial. De acuerdo con la consultora OneAudience se espera que este mercado mundial alcance este año ingresos de USD 155 mil millones.

[1] *Fuente: "https://www.match2one.com/blog/what-is-programmatic-advertising/"*

[2] *Fuente: "https://www.match2one.com/blog/what-is-programmatic-advertising/"*

Figura 3: Global Programatic Market Growth, (2017-2021) Billions [3]

La empresa creadora de audiencias en la cual se basa este trabajo participa en este ecosistema. Específicamente busca ofrecer registros con características relevantes para la marca o agencia de publicidad, creando valor en la transacción entre medios y anunciantes, a través del marketplace de audiencias Data Management Platform (DMP). Este DMP o también conocido como "***audience platform***" es una plataforma que permite organizar y crear audiencias que se pueden empujar a distintos canales de activación (Facebook, Instagram, etc) para hacer compra de publicidad segmentada.

En particular, este trabajo estará enfocado en "data exchange". La empresa trabaja con proveedores a los que les compra datos, y esto sirve para alimentar audiencias con datos demográficos y de intereses de afinidad por ejemplo. Las audiencias con datos de proveedores se incorporan en el ecosistema de la compra programática y se disponibilizan en el DMP para que las agencias de publicidad las compren y ejecuten sus campañas de forma dirigida. Es aquí donde entrará en juego el modelo desarrollado en este proyecto que busca predecir características demográficas de los registros. La siguiente imagen muestra una vista general del mercado y el lugar donde se ubica la compañía para la que se realizará el proyecto de predicción de atributos demográficos de usuarios. Concretamente, la compañía aporta valor en el mercado mediante **DMP y agregadores de data.**

[3] *Fuente: "https://www.onaudience.com/resources/top-data-markets/"*

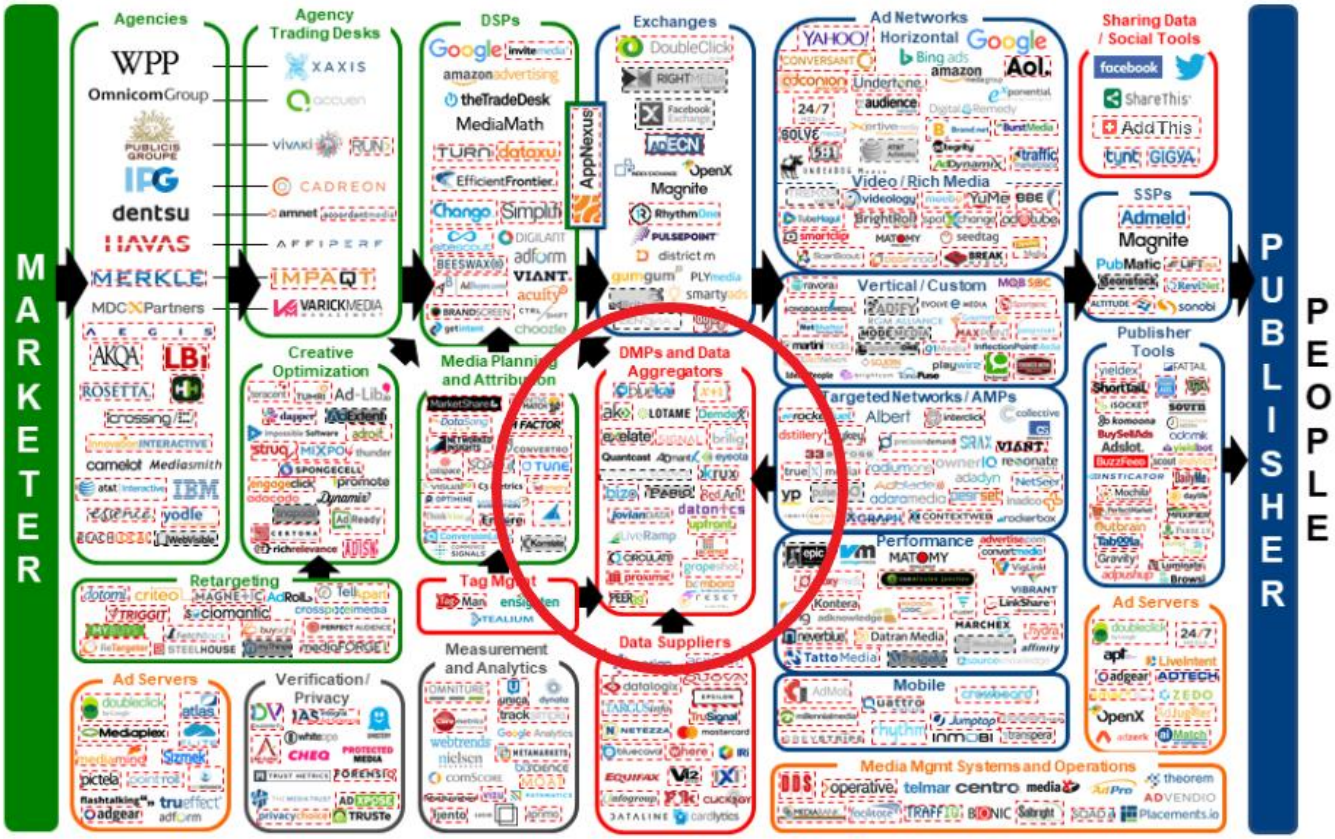


Figura 4: Mercado de audiencias [4]

En cuanto al modelo de negocio en particular, cuando los datos de la empresa se usan en una campaña de marketing, el anunciante paga un fee (CPM o Cost Per Mile, ya que se ofrecen mil impresiones por compra) y la compañía le paga un porcentaje al proveedor de data.

1.3. Problema y Objetivo

Conocer el género y la edad de las personas resulta fundamental en empresas dedicadas a crear audiencias customizadas de usuarios para marcas de productos y/o servicios que quieren dirigir de manera acertada sus anuncios publicitarios.

Si las campañas publicitarias llegan a potenciales compradores del producto o servicio, las agencias de publicidad o compañías generadoras de dichas campañas podrán aumentar sus ingresos, para llegar a más personas y crecer en el mercado. Por lo tanto, empresas dedicadas a construir audiencias que consistan en esos potenciales compradores, si generan gran volumen de usuarios (implicando captación de nuevos clientes) como así también si aciertan con el tipo de usuario (denotado en cantidades de impresiones, clics y conversiones), seguirán siendo elegidas por las mencionadas agencias y compañías.

[4] *Fuente: "https://lumapartners.com/content/lumascapes/display-ad-tech-lumascape/"*

El objetivo de este trabajo es aprovechar parte de la información proporcionada por una empresa dedicada, entre otras cosas, a vender datos customizados a agencias de publicidad, cuyos clientes directos son marcas de productos/servicios, para predecir atributos demográficos de usuarios. Esto se hará con datos de los dispositivos y sus comportamientos web registrados, para el mercado de Argentina, en base a modelos de machine learning.

Estas predicciones permitirán expandir audiencias de usuarios ofrecidas a marcas para que las mismas logren captar la mayor cantidad de clientes posible en diversas plataformas digitales y por ende esto hará aumentar los ingresos de la empresa proveedora de audiencias, ya que al cobrar por impresiones, a mayor cantidad de usuarios ofrecidos, más *Revenue* generado.

Dada esta meta sobre aumento de ingresos, se evaluará el revenue posible de ser generado mediante los segmentos aumentados en volumen de usuarios a través de modelos de Machine Learning. En particular, se analizará el enfoque Business Value of Information (BVI) y el enfoque Economic Value of Information (EVI).

Actualmente la empresa de la que se toman los datos arma sus audiencias a partir de datos de proveedores. Pero esto en muchos casos no resulta suficiente para ofrecer a agencias un volumen considerable de usuarios. Actualmente se aplica un llamado “algoritmo de la moneda” donde se le asigna género de forma aleatoria a los usuarios de los que no se tenga el dato. Siendo así, el género asignado es poco confiable, dando foco único al volumen, restando importancia a la calidad de data entregada. Es entonces que modelar mediante datos de navegación y user agent para expandir audiencias tiene fuertes implicancias y puede significar un mejor posicionamiento de mercado tanto para las marcas de productos, como para las agencias de publicidad y por ende para la empresa creadora de audiencias que seguirá siendo elegida por proveer audiencias confiables y de gran volumen.

Este trabajo entonces busca mostrar cómo, a partir de datos de “user agent” y de navegación web, se pueden generar modelos predictivos competitivos para el mercado de audiencias de usuarios para campañas publicitarias digitales. Concretamente, los modelos serán entrenados para predecir el género de una persona, dados los sitios web y los dominios que visitó en el mes de Enero 2021 y dado el modelo y marca del dispositivo con el cual lo hizo, además del navegador que haya utilizado, la versión del mismo y el sistema operativo. Además, se estudiará el impacto en ingresos y en el negocio que estos nuevos atributos generen a la empresa creadora de audiencias. Es decir, se tratará de mostrar cómo mejora la performance del servicio ofrecido cuando se aumenta la cantidad de usuarios gracias a estos modelos.

El resto del documento se encuentra estructurado de la siguiente manera: en la *sección 2* se presentarán el análisis de ingresos generados gracias al modelado. En la *sección 3* se mostrará qué datos se obtienen y de qué manera, llegando al output final a analizar. En la *sección 4* se describe el marco teórico de técnicas de machine learning que se aplicarán en este proyecto. En la *sección 5* se presentarán los modelos y métodos. En la *sección 6* se describirán estrategias de interpretabilidad de los modelos. En la sección 7 se mostrará una adaptación de los modelos para predecir rangos de edad y finalmente en las últimas 2 secciones se presentan resultados y conclusiones

2. Análisis de Evaluación Económica del proyecto

Previo a modelar, es necesario entender qué caudal de ingresos se podría generar en el negocio de la empresa vendedora de audiencias con este proyecto. Y, de este modo, entender que tan valioso resulta el mismo.

Con los datos disponibles puede observarse que un mayor volumen de usuarios para segmento femenino y masculino se corresponde con mayores ingresos para la empresa proveedora de audiencias, ya que la misma puede ofrecer a las agencias de publicidad y compañías que tiene como clientes, audiencias con más cantidad de usuarios con género asignado. Los datos demográficos resultan fundamentales al momento de vender audiencias a agencias publicitarias o empresas de consumo masivo o servicios, ya que se trata de atributos que muchas veces definen intereses.

El siguiente gráfico muestra ingresos versus tamaño de segmentos de género, con datos del segundo semestre del 2020.

.

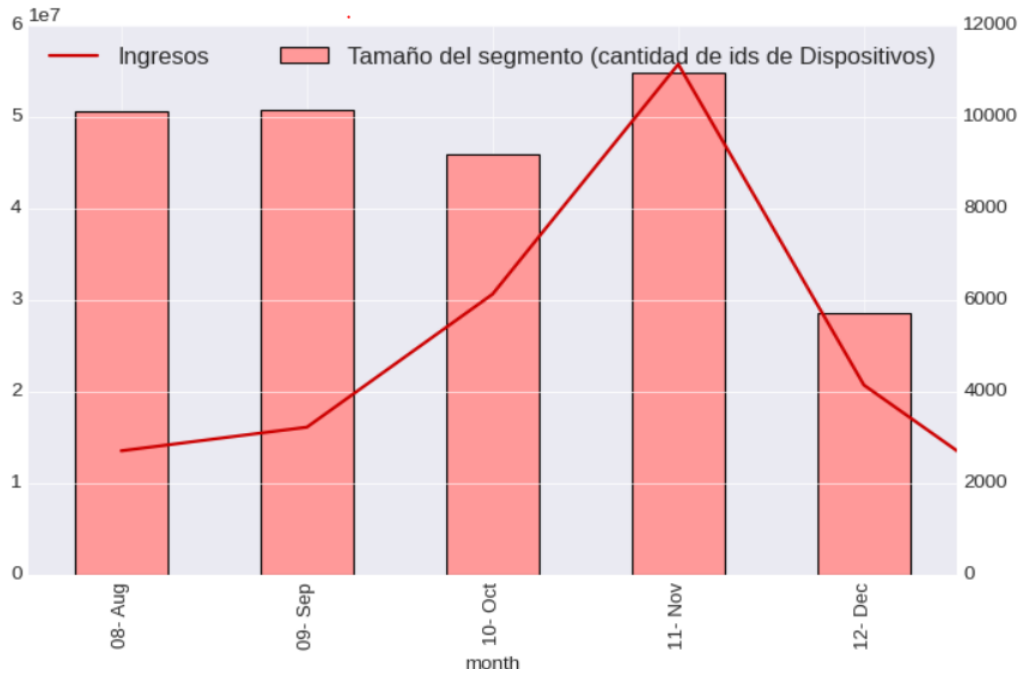


Figura 5: Tamaño del segmento Género Femenino y Género Masculino e Ingresos

Para evaluar económicamente este proyecto, se tomarán los enfoques Economic Value of Information (EVI) y Business Value of Information (BVI) descritos como *Financial Measures* y *Foundational Measures* respectivamente (Douglas B. Lany, 2017) [3].

Figura 6: Gartner Information Asset Valuation Models[5]

2.1 EVI

Este método da como resultado el valor financiero neto de un activo de información aplicando el enfoque de ingresos tradicional para la valoración de activos y luego restando los gastos asociados al ciclo de vida del activo de información. Considera el cambio realizado en los ingresos cuando cierta información se incorpora a uno o más procesos de generación de ingresos. Entonces el costo de adquirir, administrar y aplicar los datos se compensa. Su fórmula se resume de la siguiente manera:

$$EVI = [Revenue\ i - Revenue\ c - (AcqExp + AdmExp + AppExp)] * T/t$$

Para este caso, el nuevo activo de información se compondrá de registros de dispositivos (identificadores de dispositivos) donde el atributo género (sea masculino o femenino) será asignado gracias a un modelo entrenado con técnicas de Machine Learning. Este campo permite ofrecer audiencias con datos demográficos de interés para los clientes.

La alternativa en uso actualmente implica el uso de un "algoritmo de la moneda", que consiste en asignar de manera aleatoria género a usuarios de los que no se tenga el dato determinístico o verdadero de parte de proveedores. Este algoritmo es realizado por la compañía bajo el intento de aumentar el tamaño de audiencias para vender a agencias de publicidad y se utilizará éste como un reemplazo del experimento que el enfoque EVI precisa.

Detallando la fórmula EVI, se considera lo siguiente:

- Revenue ¡ = El ingreso generado usando el nuevo activo de información.

  Tomando datos del año 2020, este revenue suma 2.783.779 dólares aproximadamente.
- Revenue c = El ingreso generado sin usar el nuevo activo de información.
  Se tomará al ingreso generado gracias a usuarios con género asignado mediante proveedores específicos de la compañía creadora de audiencias. Esto será lo que se conoce como Ground Truth

[5] *Fuente: Laney. "Why and How to Measure the Value of Your Information Assets", Gartner Research, November 15,2016*

(GT). En base a datos del 2020, tanto por usuarios femeninos como masculinos, la suma de ingresos por impresiones provenientes de data Ground Truth es 870.507 dólares.

- T = La vida útil promedio esperada de cualquier instancia de información dada o registro. Como T se tomará el tiempo estimado de vida de una "cookie", ya que el estudio se basa en datos de sitios visitados además de datos de user agent. (Una vez muerta la cookie, habrá que volver a predecir el género del usuario). Se estima que una cookie dura en promedio 7 días.
- t = El período de tiempo durante el cual se realizó el ensayo de EVI. Como t, se tomará el tiempo de realización del antes mencionado "algoritmo de la moneda" (3 meses).
- AcqExp, AdmExp, AppExp = Costos del ciclo de vida de la información (costos de adquisición, de administración y de aplicación). Como AdmExp y AppExp se tomará aquel que refiere al espacio en servidores para correr mensualmente los modelos. Se necesitará un servidor para la entrada de identificadores de dispositivos nuevos y sus atributos de navegación, a evaluar por el modelo. Como AcqExp, es decir, costo de adquisición, se tomará el salario abonado al profesional dedicado al proyecto.

La gran diferencia con los modelos a realizar en este proyecto, cabe decir, radica en la calidad de la asignación del género. El proyecto tomará datos de User agent y de navegación web para realizar la mejor asignación posible permitiendo una mayor probabilidad de impacto óptimo de campañas publicitarias hacia los usuarios objetivos.

Sin embargo, el hecho de entender si las audiencias de usuarios con género asignado son impactadas verdaderamente por las campañas publicitarias escapa a este caso de estudio ya que la empresa en la que este trabajo basa su análisis vende audiencias customizadas sin recibir ningún tipo de feedback por parte de sus clientes en cuanto al impacto positivo o negativo de campañas. Es un input que requiere hacer experimentos de A/B testing para poder evaluar causalmente el uplift. Por lo que esto está fuera de nuestro control. Además, cabe mencionar que las agencias de publicidad compran audiencias sin saber qué proporción de usuarios tienen género asignado mediante modelos estadísticos y mediante proveedores (datos reales). De todos modos, ser asertivo en la asignación de género puede derivar en re-elección de parte de los clientes a la empresa de audiencias en un mediano o largo plazo. Y por eso, se hará gran énfasis en la métrica de AUC al momento de modelar, priorizando la mayor separación de clases posible.

En base a lo discutido se obtienen los inputs necesarios para el análisis del EVI. En cuanto al Revenue, se tomarán la cantidad de impresiones cobradas y su precio. Dado que se cobra a las agencias de publicidad mediante CPM (Costo por cada mil impresiones), y el mismo varía entre 0.4 USD y 1.5 USD según la plataforma a la que estén dirigidas las impresiones, se tomará el promedio. Es decir un CPM = 0.95 USD. Como puede observarse, los ingresos son mayores en general cuando se trata de asignación por algoritmo de la moneda, debido a un mayor volumen ofrecido.

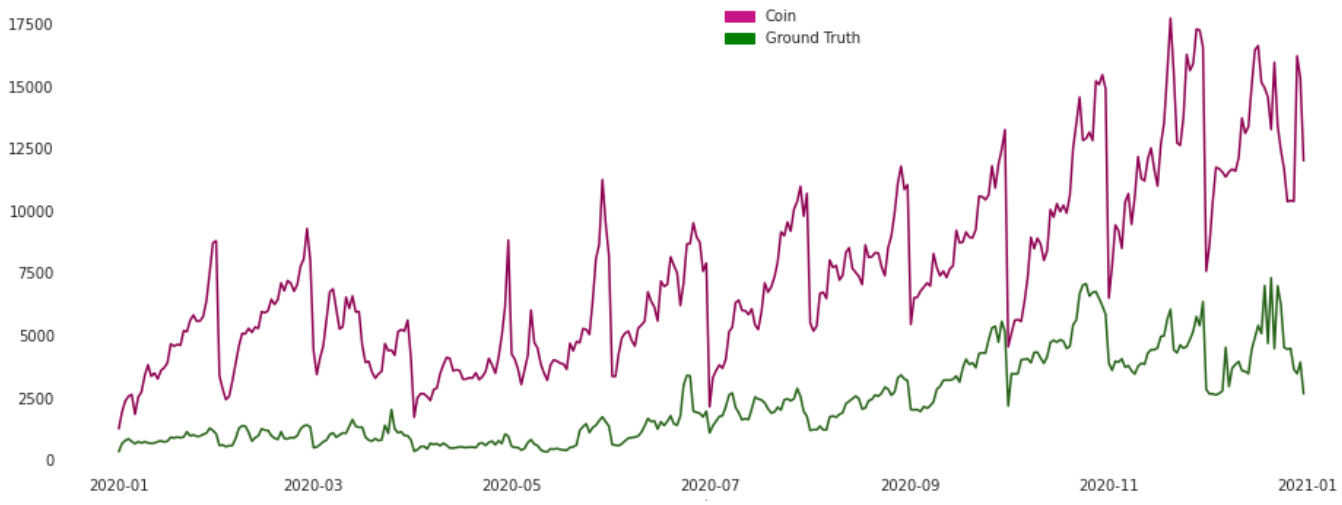


Figura 7: Ingresos generados por segmentos con género asignado por Ground Truth vs Moneda

En el siguiente gráfico se aprecia la proporción de revenue aportado por el modelo y por data GT, siendo el algoritmo el que más revenue genera, como se observó previamente.

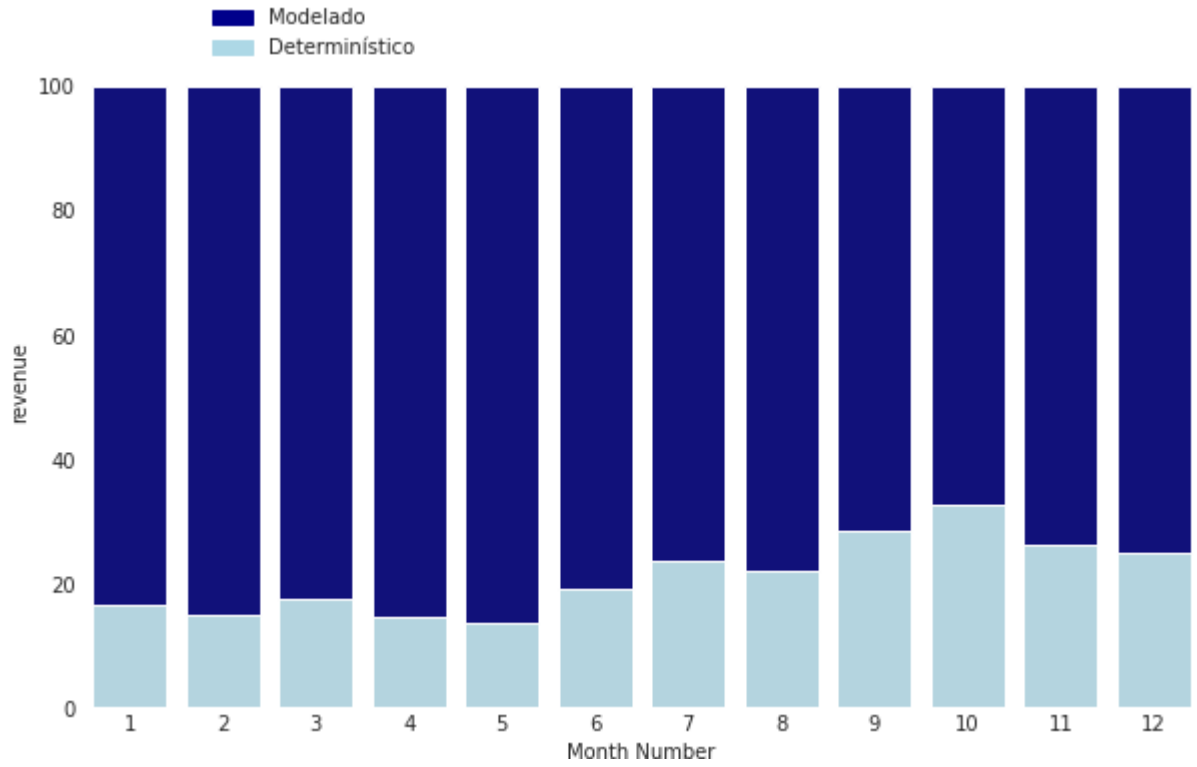


Figura 8: Proporción de Revenue por asignación determinística vs Algoritmo de la Moneda

Al observar la cantidad de usuarios etiquetados tanto por el modelo como por la data GT, se evidencia que más cantidad implica mayor revenue.

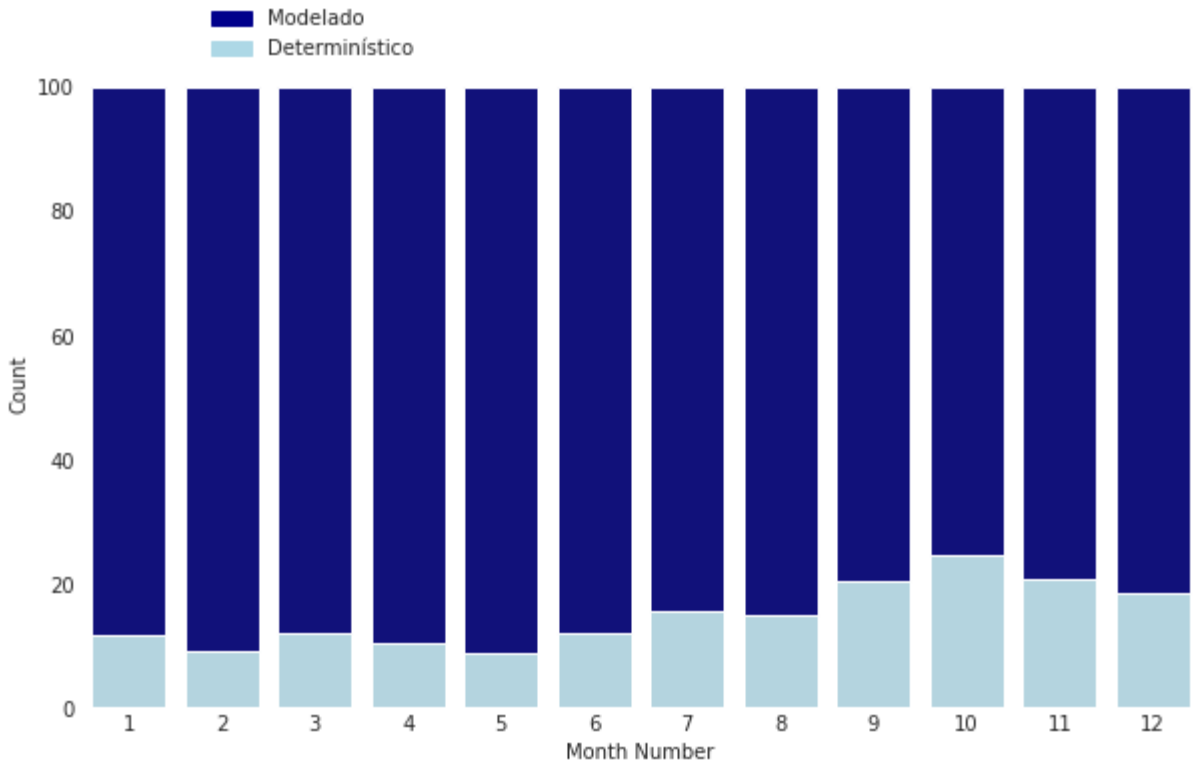


Figura 9: Proporción de registros por asignación determinística vs Algoritmo de la Moneda

Para finalizar con este enfoque, se especifican costos y se estiman tiempos en base a la experiencia.

- AppExp + AdmExp, es decir, el costo del espacio en servidores para correr el modelo por mes. Esto será, en días, 1.6 USD.
- AcqExp, es decir, el salario diario abonado al profesional que se dedicará al modelado. Será de 50 USD.
- T, será 7 días
- t será 90 días

Por lo tanto, el EVI de un mes queda positivo, siendo:

$$EVI \ = \ [231981{,}5\ USD - 72542{,}25\ USD - (48USD + 1500USD)] * 0.233/3 \ = \ 12485{,}16$$

El objetivo es asignar género lo más certeramente posible para poder expandir el volumen de registros completos disponibles para los clientes. Al final de este proyecto, cuando sepamos la performance del modelo elegido, presentamos una visualización de la sensibilidad del EVI respecto al AUC donde se compara el EVI obtenido con y sin nuestro modelo. Para obtener el revenue esperado dado un volumen de registros a ofrecer, estimamos un modelo de regresión lineal que tendrá como variable a predecir el revenue obtenido gracias a segmentos demográficos de género. Los predictores serán por un lado las cantidades diarias históricas en el año 2020 de dispositivos a los que se les asignó género mediante data ground truth y mediante data modelada (por algoritmo de la moneda), una variable binaria que vale 1 cuando esas cantidades fueron generadas por el algoritmo de la moneda y 0 cuando no y finalmente otra variable explicativa que consistirá en las cantidades del primer predictor al cuadrado, ya que el efecto en

EVI podría no ser constante. El objetivo es entender la sensibilidad del EVI respecto al AUC que pueda lograr un algoritmo de machine learning aplicado. En dicho cálculo se tomará como revenue sin modelo aplicado (revenue del grupo de control en la terminología de Laney) a aquel que se obtenga con AUC de 0.5.

La *Ecuación de regresión poblacional a estimar resulta* $Y = \beta 0 + \beta 1 vol + \beta 2 vol2 + \beta 3 Moneda + Error$, siendo vol, $vol2$ y $Moneda$ los predictores, donde "vol" representa los bloques de cantidades de dispositivos con género asignado vendidos por día, vol2 su cuadrado (para permitir efectos no constantes del volumen sobre el precio) y "Moneda" es la variable binaria que indica el tipo de asignación de género (0 si las cantidades son Ground Truth y 1 si corresponde al algoritmo "de la moneda" comentado previamente). La variable a explicar "Y" es el revenue diario generado gracias a dispositivos con género asignado.

	Coef.	Std Err	P>|t|
Intercept	486,6857	38,833	0,000
Vol	0,00070	0,0000122	0,000
Vol2	-3,131e-12	4,11e-13	0,000
Moneda	-1393,0644	67,714	0,000
Statistics			
R-squared	0,9830		
Adj. R-squared	0,9830		

Tabla 1: Resultado de Regresión Lineal

Antes de obtener resultados de la regresión de las variables en niveles se corroboró que las series sean estacionarias en media mediante el test Dickey Fuller. Este paso es necesario para evitar regresiones espúreas al trabajar con series temporales. En particular, el test nos permite rechazar la hipótesis de raíz unitaria en favor de la hipótesis de que cada serie es estacionaria en media. A continuación se observa la sensibilidad que tiene EVI ante aumentos del AUC.

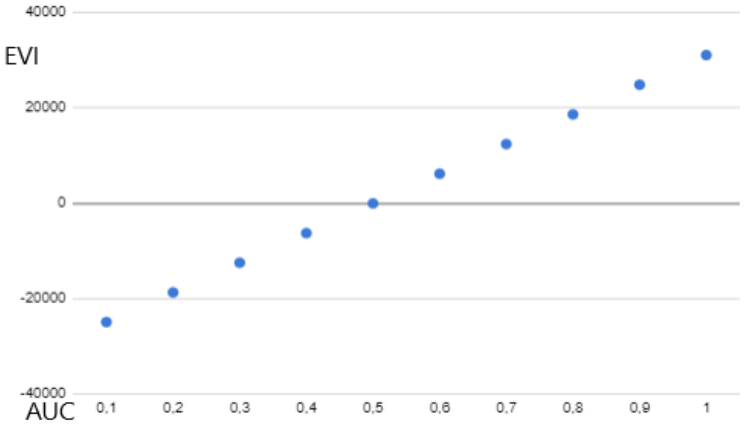


Figura 10: Sensibilidad EVI respecto a AUC [6]

[6] *Se toma como tamaño del dataset al que se utilizará al entrenar modelos y para el cálculo del revenue a incluir en fórmula EVI se usan los coeficientes estimados de la regresión*

2.2 BVI

En lo que respecta al análisis fundamental, se analiza el enfoque "**Business Value of Information**" BVI.

El BVI considera la utilidad de un activo de información para el uso comercial real. Busca poder estimar qué tan buena es la información.

Este método es útil para obtener una visión rápida del beneficio potencial de la información. Su definición es la siguiente:

$$BVI = \sum(Relevance\ p) * Validity * Completeness * Timeliness$$

Donde:

- **Relevance p** = Es qué tan útil la información puede ser para uno o más procesos del negocio.
- **Validity** = Es el porcentaje de registros correctos sin errores en los datos.
- **Completeness** = Es el porcentaje de registros de los que se obtiene el nuevo atributo, sobre el universo total de los que podría obtenerse dicho atributo.
- **Timeliness** = Es qué tan rápido nuevas instancias de datos se encuentran capturadas y accesibles.

Este enfoque se tiene en cuenta para poder rankear entre alternativas. Es decir no es una medida de valor económico sino una medida de valor fundamental, en cuyo cálculo se tiene en cuenta la calidad de los datos. El BVI permite rankear entre distintas alternativas entre las que asignar recursos a iniciativas de calidad de datos. A cada componente insumo del BVI se le asigna un puntaje entre 0 y 1. Cuando uno de esos aspectos toca o se acerca a cero, el producto cae y por lo tanto resulta en un puntaje bajo para activo de información que se esté evaluando. Para este caso se evaluaría BVI para cada audiencia ofrecida (vista como un proceso de negocio puntual), obteniendo de este modo un especie de ranking entre audiencias tomando en cuenta qué tan útil es el género de los usuarios que forman parte de la misma, qué tan rápido, qué tanto y qué tan bien se predijo al género que formará parte de la audiencia evaluada. Algunos ejemplos de audiencias ofrecidas son: 'Tecnología > Indumentaria > Zapatos > Mujer', 'Tecnología > Indumentaria > Zapatos > Hombre', 'Entretenimiento > Películas > Mujer', 'Hogar y Jardín > Manualidades > Mujer', 'Tecnología > Indumentaria > Casual > Hombre', entre otras.

Para mostrar cómo cambia el BVI respecto a valores de AUC, es decir la sensibilidad del BVI ante cambios en la performance de un potencial modelo de machine learning, nos concentramos en la variable *completeness* y dejaremos fijo a valores de relevance, validity y timeless. Como universo total de quienes se podría obtener el atributo género, se toma una cantidad aproximada de habitantes de Argentina con uso de dispositivos celulares, tablets y pc en rangos de edades entre 15 y 59 años: 27.198.716 personas. Este dato es tomado de las proyecciones de población por edad del INDEC correspondientes al año 2020[7] y se asume que todos tienen acceso a los dispositivos mencionados. Además, hay que tener en cuenta que este universo puede ser mucho menor ya que existen otras restricciones que lo definen según cada audiencia evaluada, como lo referido a intereses, zonas de residencia y nivel de ingreso entre otros.

[7] La información está disponible en el siguiente link https://www.indec.gob.ar/indec/web/Nivel4-Tema-2-24-84.

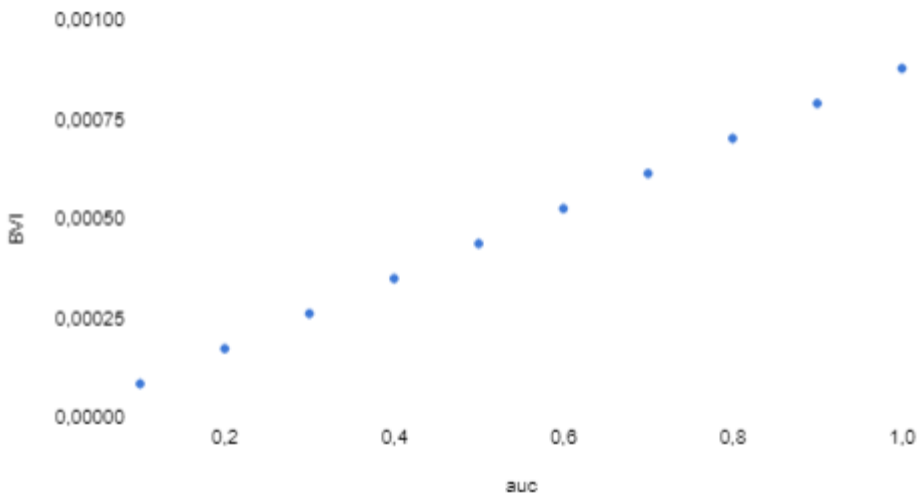


Figura 11: Sensibilidad BVI respecto a AUC [8]

3. Estructura de los datos

El dataset consiste en datos de navegación web y los llamados datos de User Agent, disponibles gracias al traqueo de datos al que tiene acceso la empresa proveedora de audiencias de usuarios.

Por un lado, el tipo de identificador con el que se lograron extraer datos de navegación web es lo que se conoce como *cookies.* Una cookie es un "trozo de información" que pertenece a un dominio particular y persiste en el navegador del usuario. La empresa recibe tráfico web obtenido mediante un código (también llamado *pixel*) propio de la empresa, colocado en ciertos sitios web. Cuando un usuario visita un sitio web, mediante este píxel se genera un "cookie id" en el navegador, que permite reconocer a ese usuario cuando el mismo vuelve a entrar a ese sitio o a otros sitios donde también está el pixel de la empresa. La información se recolecta a través de una API de la empresa creadora de audiencias. El proceso se resume de la siguiente manera: Persona → Dispositivo → Navegador → se identifica al usuario a través de *cookie id*, que se obtiene mediante un código que se coloca en un sitio. La cookie vive en un dominio específico por pocos días. De esta forma se tiene información sobre urls y dominios visitados.

Por otro lado, los datos de user agent vienen directo del navegador utilizado por el usuario del dispositivo, ya que son datos que forman parte del protocolo http. Se obtienen cuando se hace un "request" al sitio web.

Dada la inmensa cantidad de datos recolectados en el mes de enero 2021 para Argentina, no fue posible mostrar todo en un único set de datos, sino que se obtiene la información desde 6 distintos sets a los que se los denominará *Labels, Features, Device Index, Feature Index, Indexed* y *Labels 2*.

[8] *Se toma como tamaño del dataset al que se utilizará al entrenar modelos y se asume relevance 0,9, validity 0,6 y timeless 0,5*

A continuación, se muestra el diagrama de relaciones entre las tablas utilizadas, seguido de una descripción detallada de cada una.

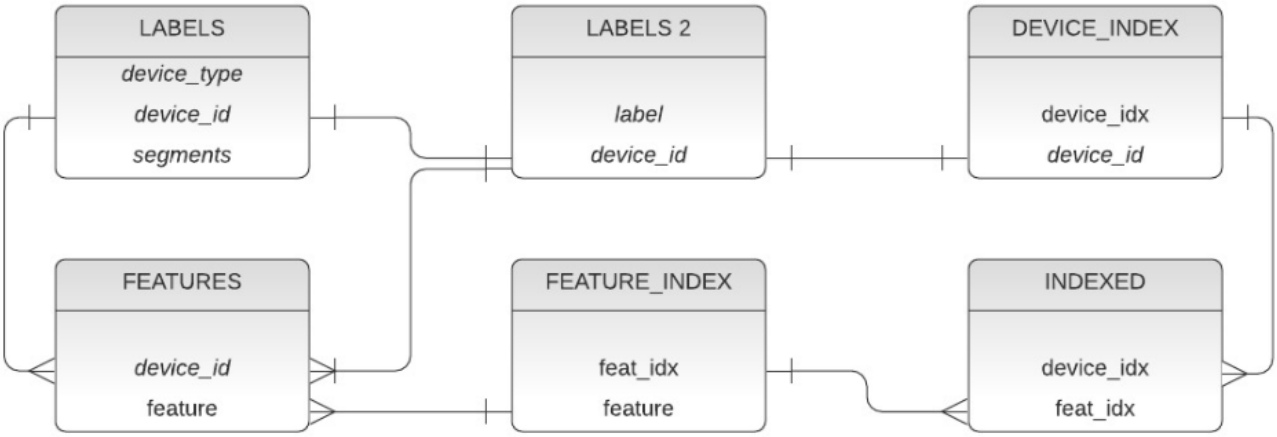


Figura 12: Diagrama de relaciones entre tablas.

Labels	
Formato	csv - separador \t
Columnas	• ***device_type***: tipo de dispositivo (en este caso todos serán “web”) • ***device_id***: el identificador de cada dispositivo • ***Segments***: Listado de segmentos asociados a un dispositivo (separados por ,)
Ejemplo	web 0006c43a-baf0-4562-8b21-6b741683bdd1 3,5 web 0011da8e-f5ca-4950-8713-08302df6b002 6,3 web 0077b5b9-8a57-4847-8841-aaf69fa43d04 6,2 web 00c6279f-56e7-4a61-9f87-ebbdf3b89fcb 7,2 web 0172d0a2-1d0d-4a94-b3db-921cd4ae9035 6,2

Tabla 2: Labels

Features	
Formato	parquet
Columnas	• ***device_id***: el identificador de cada dispositivo • ***feature***: String del valor del feature en formato nombre@valor
Ejemplo	| device_id | feature |00060fcb-fa7f-414...| of@Android

	|00060fcb-fa7f-414...| ov@Android 7 |00060fcb-fa7f-414...| df@LM-X210 |00060fcb-fa7f-414...| bf@Chrome |0000e78c-d5a8-4ea...|url@horoscoponegr... |000edb54-2c5d-414...|url@vix.com/es/im... |0012352c-7851-4a8...|url@damianculotta...

Tabla 3: Features

Device_index	
Descripción	Mapea los dispositivos con sus índices o número de fila.
Formato	parquet
Columnas	• ***device_idx:*** Índice - Entero que representa el número de fila de la matriz de entrenamiento. • ***device_id***: el identificador de cada dispositivo
Ejemplo	|device_idx| device_id| | 0 | 00006070-10bc-44e...| | 1 |0000ea8d-feb4-416...| | 2 |00010524-61b4-472...| | 3 |0001cda4-8dfc-475...| | 4 |00028b8f-e4ff-432...|

Tabla 4: Device Index

Feature_index	
Descripción	Mapea los features con sus índices o número de columna
Formato	parquet
Columnas	• ***feat_idx:*** Índice - Entero que representa el número de columna de la matriz de entrenamiento. • ***feature***: Nombre del feature
Ejemplo	|feat_idx| feature| |0 | bf@Brand_browser| | 1 | bf@Chrome| | 2 | bf@Edge| | 3 | bf@Firefox| | 4 | bf@Google|

Tabla 5: Feature Index

Indexed	
Descripción	Tuplas con interacciones entre dispositivos y features en formato indexado.
Formato	parquet
Columnas	• ***device_idx***: Índice dispositivo - Entero que representa el número de fila de la matriz de entrenamiento. • ***feat_idx:*** Índice de feature - Entero que representa el número de columna de la matriz de entrenamiento.
Ejemplo	|device_idx|feat_idx| | 18886 | 13684| | 67515 | 23635| | 17311 | 25903| | 55304 | 22174| | 29413 | 17882|

Tabla 6: Indexed

Labels 2	
Descripción	Tuplas con interacciones entre devices y features en formato indexado.
Formato	parquet
Columnas	• ***device_id***: el identificador de cada dispositivo • ***label***: Segmentos separado por ",". Se refieren a género y edad.
Ejemplo	| device_id |label| |a9c8b817-c03a-49c...| 2,5| |167304fb-afb3-4ac… | 5,2| |839deab0-c0f9-464… | 3,6| |c5c91ffa-7aa3-412… | 3,6| |ea33c35f-0fcd-453… | 5,2| |7380bcd3-1a02-4f7… | 3,7| |2539805d-553d-4e8...| 6,3|

Tabla 7: Labels 2

Antes de comenzar con el análisis exploratorio de los datos se procesan estos datos de manera tal de obtener un dataset unificando la información necesaria. El mismo se encuentra detallado en el Anexo 1.

3.1. Estructura de los datos en la Matriz de Features

A partir de los conjuntos de datos presentados en la sección anterior, el set de datos del cual se parte para analizar y utilizar en los modelos de machine learning presenta las siguientes características.

Es una matriz compuesta por 0 y 1, donde 1 significa que existe intersección entre el dispositivo (usuario), representado en las filas, y el feature, representado en las columnas. El 0 significa que no existe relación entre el dispositivo y el feature. Luego, la variable target es 1 o 0 según el género. Siendo 1 si es **femenino** y 0 si es **masculino**.

Dataset para entrenar Modelos	
Formato	csv
Columnas	• ***device_idx***: el identificador de cada dispositivo • ***fem***: Variable binaria que indica el género. • ***Feature_Android 7***: 0 o 1 según el dispositivo cuente o no con ese feature..
Ejemplo	|device_idx| feature_Android 7 |...| feature_zonapropr.com.ar |fem| |4562 | 0 |...| 1 | 0| |4562 | 0 |...| 0 | 0| |4562 | 1 |...| 1 | 1| |4562 | 1 |...| 0 | 0|

Tabla 8: Formato de Dataset para Entrenar Algoritmos de Machine Learning

4. Marco Teórico

A continuación, se hará una breve descripción de la teoría que hay por detrás de las técnicas de Machine Learning a implementar en el proyecto, tomando como referencia bibliografía sobre la cual se sustentan y comentando su aplicación en el presente trabajo.

4.1. Naive Bayes

Bayes Ingenuo o Naive Bayes es un algoritmo de clasificación que modela la probabilidad de que una observación i con determinadas características (Xi) pertenezca a una determinada clase k, es decir: $P(Ck \mid Xi)$, donde:
$P(Ck \mid Xi) = P(Ck, Xi) \,/\, P(Xi)$ por **probabilidad condicional** $P(Ck \mid Xi) = (P(Xi \mid Ck) * P(Ck))/P(Xi)$ por **Teorema de Bayes**.
En el teorema de Bayes, el numerador es equivalente a una probabilidad compuesta: $P(C, X1, \ldots., Xn)$, que puede ser descrita de la siguiente manera, aplicando la probabilidad condicional:

$$P(C, X1, \ldots., \quad Xn) = p(C)\, p(X1|C)\; p(X2|C, X1)\, p(X3|C, X1, X2)\, p(X4, \ldots, Xn|C, X1, X2, X3)$$

y así sucesivamente. Aquí aparece el supuesto "ingenuo" de la **independencia condicional** entre variables. Se asume que cada Xi es independiente de cualquier otra Xj para $i \neq j$cuando están condicionadas a C. Siendo: $P(Xi \mid C, Xj) = p(Xi|C)$, con lo cual la probabilidad compuesta puede expresarse como $p(C)\prod p(Xi/C)$. Esto significa que la distribución condicional de C sobre las variables clasificatorias puede

expresarse como: $P(C|\, X1,..,Xn) = 1/Z \ * p(C) \prod p(Xi/C)$, donde Z es un factor que depende únicamente de $X1,\ldots,Xn$.

De este modo, el algoritmo Naive Bayes clasifica observaciones asumiendo que los features o atributos no tienen relación alguna entre sí, dada la variable target, caracterizándose de este modo por ser simple y básico.

4.2. Regresión Logística

Este tipo de modelos son no lineales en los parámetros, por lo que se estiman por máxima verosimilitud, aunque alternativamente se podrían estimar por mínimos cuadrados no lineales Se estiman los parámetros desconocidos buscando maximizar la probabilidad de extraer la muestra observada. En otras palabras, las estimaciones de máxima verosimilitud de los parámetros desconocidos son los valores que dan como resultado un modelo que es más probable que produzca los datos observados.

En una regresión logística la probabilidad de que una observación sea catalogada como *target* igual a 1, es igual a:

$$P(Target\ = 1)\ = exp(\beta 0\ +\ \beta 1\ X1 + \ldots + \beta n\ Xn)\ /\ (1\ +\ exp(\beta 0\ +\ \beta 1\ X1 + \ldots + \beta n\ Xn))$$

Por lo tanto, a medida que los operadores *exp()* tengan valores más altos, el valor de *P(Target=1)*, también lo será. De hecho, en el límite, cuando los operadores *exp()* tienden a infinito, *P(Target=1)* tiende a 1, y cuando los operadores exp() tienden a -infinito, P(Target=1) tiende a 0.

De esta forma, a mayor valor absoluto del coeficiente, más sensibles serán los operadores exp(), siendo el impacto en ellos positivo cuando el signo del beta sea positivo, y negativo cuando el signo del beta sea negativo. En otras palabras, a mayor valor absoluto del coeficiente de un regresor, más *sensible* será la predicción a la variación de la variable asociada (siempre y cuando las variables estén escaladas), y este impacto será positivo o negativo dependiendo del signo del coeficiente del regresor correspondiente.

El output predicho del modelo de regresión logística es una probabilidad, como se indicó anteriormente. Este output define, basándose en un umbral fijado, la clase de la observación a clasificar. Como mecanismo para elegir el umbral de corte, se usa una muestra de entrenamiento y se observa la tasa de error sobre la muestra de test. En este proyecto, la elección se realiza por Cross Validation sobre la muestra de train. Si el modelo tiene una tasa de error sobre test baja, puede decirse que el modelo performa bien.

Para este trabajo, este tipo de modelo será testeado antes que random forest y xgboost para determinar si un usuario posee género femenino o masculino, ya que es un algoritmo sencillo, interpretable y rápido computacionalmente. Luego, se aumenta la complejidad al testear estos otros modelos conocidos como “modelos de ensamble”. Al entrenar Regresión Logística, se indican como opciones de penalidad a las técnicas de regularización lasso o ridge. Mediante el método Grid Search, el modelo eliminará (en el caso

de elección *lasso*) o tratará debidamente a las variables que no son relevantes (en el caso de elegir *ridge*). Luego se evaluará qué tan bien logra clasificar.

4.3. Random Forest

Random Forest es una de las técnicas dentro de modelos de ensambles en Machine Learning. A los modelos que combinan las predicciones de modelos más pequeños (o modelos base) se los conoce como modelos de ensamble. Al igual que la técnica bagging, luego de construida la secuencia de modelos, Forest los ensambla para reducir la varianza del error de predicción de los mismos. En bagging, los árboles pueden estar positivamente correlacionados por ser entrenados con información similar (ya que cada árbol individual usa los mismos predictores). Promediar muchos valores correlacionados no conduce a una reducción importante de la varianza como sí lo haría el tomar promedios de valores no correlacionados. Por lo que la técnica de bagging no conduce a una sustancial reducción de varianza del error de predicción. Random Forest, en cambio, "decorrelaciona" los árboles mediante un pequeño ajuste. Plantea un *re-muestreo doble*. Por un lado, al igual que en bagging, se construyen árboles a partir de muestras de entrenamiento *bootstrap,* que es un método de re-muestreo que permite la obtención de nuevos conjuntos de muestras Z1, Z2, …, ZB desde un único conjunto de datos Z, seleccionando aleatoriamente B observaciones del conjunto de datos original, con reposición, para construir los B conjuntos de datos bootstrap (Z1, Z2, …, ZB) y entrenar al set de árboles. Por otro lado, la diferencia con bagging radica en que cada vez que se considera una división en un árbol, una muestra aleatoria de "m" predictores se elige desde el conjunto completo de "p" predictores para ajustarse al modelo. Es así que los árboles están decorrelacionados haciendo que el promedio de árboles resultantes sea menos variable. A continuación se detalla el algoritmo descrito.

Algorithm *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Algoritmo Random Forest[9]

Para el caso de estudio, se usará clasificación, donde la mayoría de los votos definirá a la clase predicha. Es decir, luego de dividir el dataset en train y test y estandarizar variables, al random forest se le pasará como input el dataset de train con variables dummy (1 y 0 como valor) y algunas variables numéricas. Luego se encontrarán los mejores hiperparámetros mediante técnicas de Random Search y Cross Validation.

4.4. XGBoost

Para entender la implicancia de un modelo XGBoost es necesario partir describiendo modelos de boosting. La idea de un modelo de boosting es ensamblar una *secuencia de modelos* simples para obtener un estimador complejo. Al entrenar modelos secuencialmente, se trata de corregir a su predecesor. En este tipo de algoritmos no se realiza sampling mediante técnicas de bootstrap, sino que cada árbol se ajusta a una versión modificada del conjunto de datos original. Este algoritmo aprende lento. Cada modelo predice la parte no predicha hasta ese momento, es decir, el target o variable a predecir es el residuo de los modelos anteriores, por lo tanto cada árbol adicional que se entrene dependerá de lo que los anteriores predijeron y actualizará los residuos. El parámetro λ (shrinkage parameter) ralentiza el proceso permitiendo más y diferentes árboles para atacar los residuos. En boosting, a diferencia de bagging, la construcción de cada árbol depende fuertemente de los árboles previamente construidos. Es por esto que el algoritmo es un tanto más complejo que los métodos anteriormente descritos.
En su adaptación a problemas de clasificación, existe una generalización que se llama Gradient Boosting Machine (GBM). GBM es un método que funciona agregando secuencialmente modelos a un conjunto, cada uno corrigiendo a su predecesor. Es un caso especial de boosting en el que los errores se minimizan mediante el algoritmo de descenso de gradiente.
XGBoost es una mejor formalización y una implementación inteligente de GBM. Viene a ser como un aumento de gradiente (su nombre, de hecho, hace referencia a un aumento de gradiente extremo). Es una combinación de técnicas de optimización de software y hardware para producir resultados superiores utilizando menos recursos informáticos en el menor tiempo posible.

[9] *Fuente: Hastie T., et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction*

Algorithm *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$
3. Output the boosted model,
   $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

Algoritmo Boosting[10]

Al igual que en random forest, para el caso de estudio la mayoría de votos definirá a la clase predicha (femenino o masculino). Luego de los pasos básicos de split del dataset en train y test y de estandarización de variables, al xgboost se le pasará como input el dataset de train con variables dummy (1 y 0 como valor) y variables numéricas. Luego se encontrarán los mejores hiperparámetros mediante técnicas de Random Search y Cross Validation.

5. Métodos y Procedimientos

Para predecir si un usuario tiene género femenino o masculino se probarán los modelos descritos en la sección anterior. Como modelo "baseline" se corre un algoritmo Naive Bayes, para tener punto de comparación en cuanto a performance. Es elegido como benchmark porque es un algoritmo de clasificación simple, donde se supone independencia condicional entre los features. De ahí el nombre "naive", es decir, "ingenuo". Al hacer el supuesto de *independen*cia *condicional* entre las variables, no suele ser demasiado restrictivo por lo que, asemejándose a la performance de clasificación aleatoria del algoritmo de la moneda, es la opción elegida a superar por los algoritmos aplicados para este proyecto. Por otro lado, es simple de entender e implementar, se entrena fácilmente y es rápido al momento de hacer predicciones.

En primer lugar se tomarán atributos de usuarios que tienen que ver con información obtenida mediante user agent. Esto es lo referido a marca y modelo de dispositivo, tipo de dispositivo (pc, tablet, móvil),

[10] *Fuente: Gareth, J., Witten, D., Hastie, T., Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R.*

navegador utilizado, sistema operativo del dispositivo y versión del sistema operativo. Luego de un análisis exploratorio y un feature engineering, se definirá un modelo final en esta sección. Como paso siguiente, se hará un análisis exploratorio y feature engineering de atributos de dominios visitados por cada dispositivo. Luego, estos se le sumarán al primer modelo final de la sección de user agent. Se probarán los tres algoritmos de aprendizaje estadístico mencionados, buscando los mejores hiperparámetros para obtener un modelo más completo y con mejor capacidad predictiva, bajo un esquema de k-Fold cross validation. Finalmente se estudiarán las urls. Luego de hacer análisis exploratorio y feature engineering de estos atributos, se agregan los mismos al segundo modelo y se correrán nuevamente modelos para obtener una última versión que prediga el género de los usuarios. Antes de puntualizar en el primer paso con atributos de User Agent, se explicará en detalle en qué consiste la estrategia grid search y random serch para optimizar hiperparámetros, y la técnica de k-fold cross validation para obtener la mejor performance posible de cada modelo estadístico. También se describirá qué métricas son elegidas para evaluar cada modelo.

A continuación, se observa la distribución de género que posee la base con la cual se entrenará a los modelos. En la misma, el dato de la variable target viene dado por proveedores, tomándose esto como "ground truth", determinístico o verdadero.

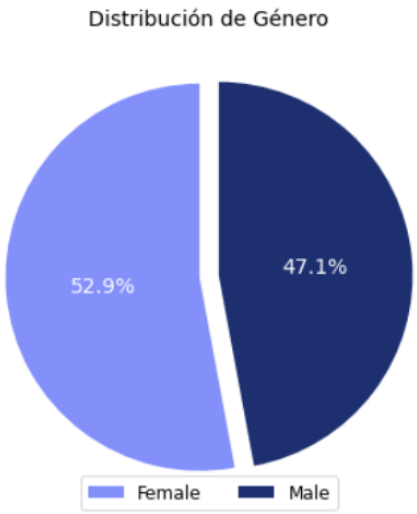


Figura 13: Distribución actual de Género

Por su parte, el dataset a analizar posee las siguientes características.

Las siglas con las que se etiquetan a cada atributo, en columna "**feature_type**", son las siguientes:

- feature type df (**Modelo Dispositivo**)
- feature type ov (**versión del sistema operativo**)
- feature type bf (**Navegador**)
- feature type db (**Marca Dispositivo**)
- feature type of (**Sistema operativo**)
- feature type ip (**Es pc**)
- feature type im (**Es celular**)
- feature type it (**Es tablet**)

- feature type oa (**Antigüedad del dispositivo**)
- feature type dom (**Dominios**)
- feature type url (**Sitios visitados**)

Las demás columnas de las que se parte son **device_idx** (id del dispositivo), **label** (número que indica femenino o masculino), **feature_detail** (el feature per sé, por ejemplo: “Chrome” es un valor de esta columna que corresponde al valor de feature_type “bf”) y **age** (número que indica el rango de edad identificado para el dispositivo). [11]

5.1. Evaluación de Modelos

5.1.1. Estrategias de optimización de hiperparámetros y validación

La performance de un modelo comúnmente depende de varios hiperparámetros. Para que un modelo tenga el mejor rendimiento posible bajo un conjunto de datos determinado, se optimizan los hiperparámetros requeridos en cada caso. Para la búsqueda de estos valores óptimos, existen distintas estrategias que prueban combinaciones de distintos valores posibles, dando como resultado un vector que implique el mejor rendimiento del modelo después del aprendizaje del mismo.

Grid Search

Para el caso de Regresión logística, la optimización de hiper parámetros se realiza mediante la estrategia **Grid Search** o Búsqueda Exhaustiva. Esta técnica define un espacio de búsqueda como una grilla de valores de hiperparámetros y evalúa todas y cada una de las combinaciones posibles en ese dominio. Se selecciona la combinación que produce el mejor rendimiento, evaluado en un conjunto de validación. El mayor inconveniente de esta técnica es el costo computacional.

Random Search

Para el resto de los modelos estadísticos, Random Forest y XGBoost, se buscará a los mejores hiperparámetros mediante la estrategia **Random Search** o Búsqueda Aleatoria. La misma define un espacio de búsqueda con un dominio limitado de valores de hiperparámetros y puntos de muestreo aleatorios en ese dominio. La desventaja de este método es que la selección de parámetros es completamente aleatoria. Sin embargo, posee grandes ventajas al insumir poco tiempo computacional y tener menor riesgo de sobreajuste de los datos en comparación con el método Grid Search.

[11] En el siguiente link se encuentran los notebooks utilizados para el análisis, procesamiento, exploración y ejecución de los modelos: https://github.com/DanyLongas/PrediccionDeGenero

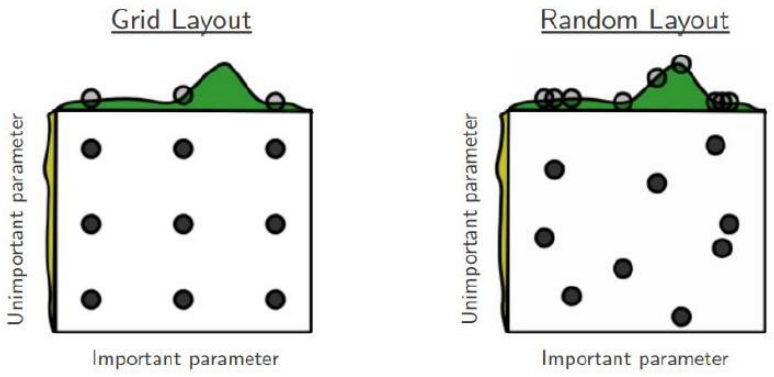


Figura 14: Esquema Grid Search y Random Search

Luego, la técnica utilizada para evaluar si las combinaciones elegidas de los hiperparámetros permiten obtener la mejor performance posible de los modelos es K-Fold Cross Validation.

Tanto Random Search como Grid Search, al ejecutar la búsqueda, no tienen en cuenta los resultados obtenidos hasta el momento. Esto impide el foco en regiones de hiperparámetros de mayor interés, evitando regiones innecesarias. La técnica de Optimización Bayesiana resulta interesante ya que tiene en cuenta estas cuestiones.

Optimización Bayesiana

Este método, también conocido como Optimización basada en modelos secuenciales, consiste en crear un modelo probabilístico (cuya meta es encontrar el mínimo de una función $f(x)$dentro de un conjunto acotado X), en el que el valor de la función objetivo $f(x)$es la métrica de validación del modelo. La idea es utilizar toda la información disponible de evaluaciones previas de $f(x)$y no simplemente depender de aproximaciones Hessianas y de gradiente local. Siendo así, la búsqueda se redirige en cada iteración a las regiones de mayor interés. Las mismas se eligen mediante la incorporaciäon de la creencia previa sobre $f(x)$(prior), actualizando el prior con muestras extraídas de $f(x)$ para obtener un posterior que se aproxime mejor a $f(x)$. Esto permite reducir el número de combinaciones de hiperparámetros con las que se evalúa el modelo, eligiendo sólo a los mejores candidatos.

A continuación, se describe brevemente cómo funciona la optimización bayesiana:

- Se asume que la función $f(x)$ surge de un proceso Gaussiano (GP), el cual se utiliza como prior para la distribución de $f(x)$, definido por la propiedad de que cualquier conjunto finito de Npuntos $\{x_n \in X\}_{n=1}^{N}$induce una distribución Gaussiana en $\mathbb{R}^{N}$.
- Se asume que las observaciones son de la forma $\{x_n, y_n\}_{n=1}^{N}$, donde $y_n \sim \mathcal{N}(f(x_n), v)$, siendo v la varianza del "ruido" introducido en las observaciones de la función.
- Con ambos, prior GP y las observaciones, se construye una distribución "posterior" de funciones que mejor describen la función que se quiere optimizar.
- A medida que el número de observaciones crece, la distribución posterior mejora y el algoritmo se vuelve más específico en cuanto a cuáles son las regiones que más vale la pena explorar de los parámetros, tomando en cuenta lo que sabe de la función target.

- En cada iteración se ajusta un GP a los datos conocidos (puntos previamente explorados) y la distribución posterior, combinada con una función de adquisición (como estrategia de exploración) es utilizada para determinar el próximo punto a explorar. Concretamente, la función de adquisición o de selección donde $a: X \to \mathbb{R}+$, determina qué punto en X debe ser el próximo a evaluarse mediante una optimización proxy $x_{next} = argmax_x\, a(x)$, donde se han propuesto varias funciones diferentes.
- En general, estas funciones de adquisición dependen de observaciones previas así como de los hiperparámetros GP. Dicha dependencia se denota como $a(x; \{x_n, y_n\}, \theta)$.

Existen varias opciones de función de adquisición como *Probability of Improvement, Expected Improvement, GP Upper Confidence Bound*. Bajo el proceso Gaussiano prior, estas funciones dependen del modelo a través de su función de media predictiva $\mu(x; \{x_n, y_n\}, \theta)$ y la función de varianza predictiva $\sigma^2(x; \{x_n, y_n\}, \theta)$. El mejor valor se denota como $x_{best} = argmax_{x_n}\, f(x_n)$. [29]

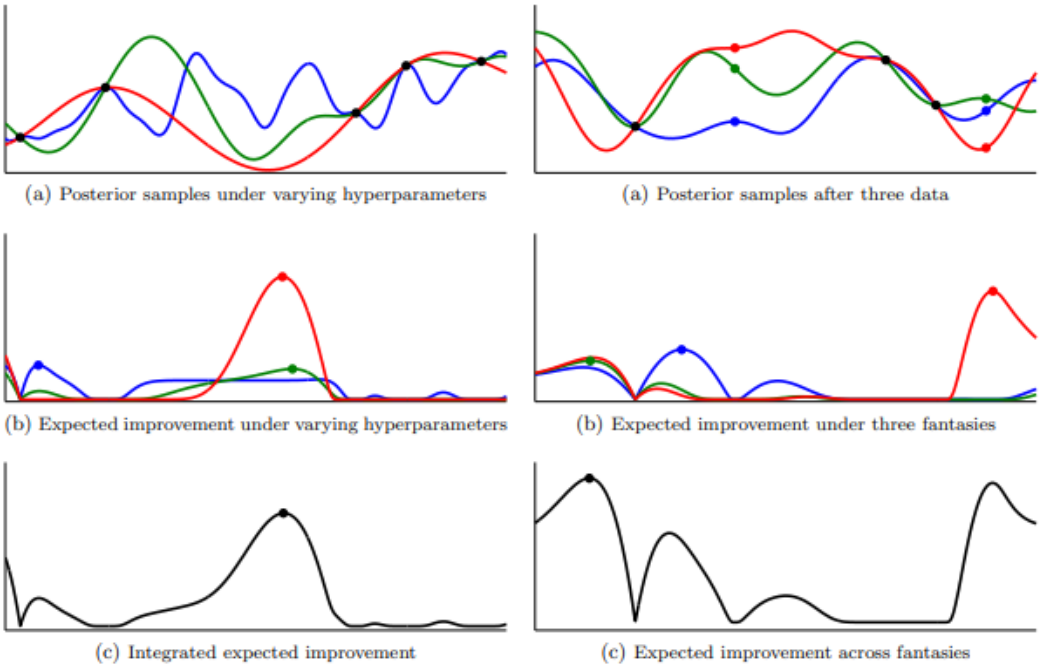


Figura 15: Ejemplo de optimización bayesiana utilizando Expected Improvement. Fuente: [29]

En el presente trabajo no se aplicará dicha técnica. Sin embargo, cabe mencionarla y tenerla en cuenta como opción interesante en una posible extensión.

Cross Validation

Una vez obtenido el dataset final como input de algoritmos de machine learning, se separa a las variables dependientes de la dependiente y luego se lo divide en conjuntos de train y test, con 80% y 20% de los datos para cada conjunto respectivamente en este caso. Train o Entrenamiento es un subconjunto para entrenar el modelo y Test o Prueba es un subconjunto para verificar que el modelo reproduce los resultados deseados. Este último se crea para evaluar el rendimiento de cualquier modelo de machine learning, donde se tiene una muestra de datos que el modelo no toma en cuenta al momento de entrenar.

En base a la validación gracias a estos datos que se dejan aparte, se puede saber qué tan bien performó el modelo. Es durante el entrenamiento del algoritmo en donde se aplica Cross Validation.

La validación cruzada (CV) es la técnica que se utiliza en este proyecto para asegurar que los modelos ajusten de la mejor manera posible. En particular, usaremos K-Fold Cross Validation con k=5, es decir, Validación Cruzada de 5 iteraciones.

Los datos del conjunto de entrenamiento (train) se dividen en K subconjuntos (5 en este caso). Uno de ellos se utiliza como subconjunto de validación y los restantes K-1 como subconjunto de datos de entrenamiento. Como el resultado depende de particiones al azar, para mejorar la estimación de la performance en testeo, el proceso de CV se repite en K iteraciones, por cada uno de los posibles subconjuntos de datos de validación. Luego se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Siendo así, esta técnica en general permite un modelo poco sesgado ya que garantiza que cada observación del conjunto de datos tenga la posibilidad de aparecer en los conjuntos de entrenamiento y de prueba. Es una forma de reducir el riesgo de overfitting o sobreajuste de los datos.

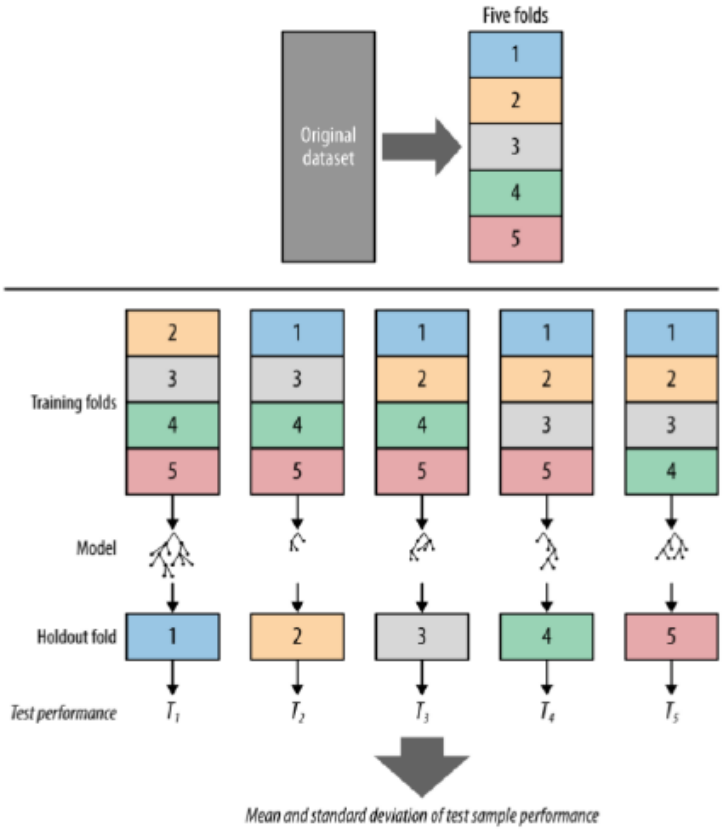


Figura 16: Esquema K-Fold Cross Validation[12]

[12] *Fuente: "https://commons.wikimedia.org/wiki/"*

5.1.2. Métricas a Evaluar

El problema en este caso de estudio involucra una variable binaria, donde 0 refiere al género masculino y 1 al género femenino, por lo que se utilizaron técnicas de machine learning para clasificación.
Cabe mencionar que, si bien mayor volumen de usuarios con género asignado implica mayor revenue, la empresa en la que se basa este trabajo busca vender buena calidad de data para evitar perder clientes por posibles errores de predicción de género de los usuarios entregados en audiencias. Es aquí que se hace mención de la llamada "*Clasificación con métrica sensible al costo*" (Andriy Burkov, 2019) [28]. En este caso, no habría evidencia de diferente valor de revenue por cada 1000 registros según el género y por ende tampoco se cuenta con información acerca de costos por predecir mal. Por lo que, con los datos disponibles, no es posible reproducir una matriz de costo. Sin embargo, suponiendo un costo por clasificar erróneamente dispositivos en cuanto a género y un revenue específico por predecir bien mujeres y por predecir bien hombres, se mencionan a continuación los diferentes escenarios que se podrían dar si el modelo pudiese intervenir en la decisión de clasificar un registro. Esta es precisamente una clasificación con métrica sensible al costo.

		Predicción	
		Masculino: 0	**Femenino: 1**
Real	**Masculino: 0**	- Revenue por venta de usuarios masculinos	Costo por menor cantidad de usuarios Masculino ofrecidos
	Femenino: 1	Costo por menor cantidad de usuarios Femeninos ofrecidos	- Revenue por venta de usuarios femeninos

Tabla 9: Clasificación con métrica sensible al costo

Asumiendo igual matriz de costos para todas las observaciones, la ecuación de costo planteada sería: (- Revenue por venta de registros femeninos)***TP** + (Costo por menos cantidad de mujeres ofrecida)***FN** + (Costo menos cantidad de hombres ofrecida)***FP** + (- Revenue por venta de registros masculinos)***TN**. Esta función sólo se utilizó para plantear el costo de equivocarnos al predecir, pero sin ser necesaria en este caso por falta de datos que diferencien ingresos según masculino/femenino. Si existiesen diferencias en cuanto a ganancias por predecir dichas características, entonces tendría sentido evaluar cuál hubiese sido el rendimiento en función del conjunto de datos de test y elegir aquel modelo de machine learning que arroje menor costo.

Luego de entrenar los algoritmos de machine learning mencionados, las métricas por comparar serán:

- ***Accuracy***: Es la fracción de todas las instancias positivas y negativas que el clasificador identifica correctamente como positivas y negativas, llamando "positiva" a la clase Femenino y "negativa" a la clase Masculino. Su fórmula específica es $(TN + TP) \;/\; (TN + TP + FN + FP)$, donde:
  TN: True Negative o Verdadero Negativo, *TP*: True Positive o Verdadero Positivo, *FN*: False Negative o Falso Negativo y *FP*: False Positive o Falso Positivo. Esta métrica, si bien se utilizará para comparar entre

modelos, será simplemente la métrica base a visualizar. El foco estará puesto en el área bajo la curva ROC, que se describe al final de esta sección.

- ***Recall***: El Recall para la clase 1 (Femenino) indica qué porcentaje el modelo detecta como verdadero para clase 1 por sobre el total real de los clasificados en clase 1. $TP\ /\ (TP + FN)$. Es también llamada True Positive Rate. El Recall para la clase 0 (Masculino) indica qué porcentaje el modelo detecta como verdadero para clase 0 por sobre el total real de los clasificados en dicha clase. $TN\ /\ (TN + FP)$
- ***Precisión***: La precisión para la clase 1 indica qué porcentaje el modelo detecta como verdadero para la clase 1 por sobre el total de predicciones bajo clase 1. En fórmula esto es $TP\ /\ (TP + FP)$.
  Para la clase 0, la precisión indica qué porcentaje se detecta como verdadero para dicha clase por sobre el total de predicciones bajo la misma. Es decir, $TN\ /\ (TN + FN)$.

- ***F1-Score***: Esta métrica representa el balance o equilibrio entre precisión y recall. Su fórmula es $2 * (Precisión * Recall)\ /\ (Precision\ +\ Recall)$

- ***AUC***: Es el área bajo la curva ROC (Receiver Operating Characteristics). Proporciona una medida agregada del desempeño en todos los umbrales de clasificación posibles, tomando como insumo las etiquetas verdaderas y las probabilidades.

  > La curva se obtiene trazando las diferentes combinaciones de **TPR:** $TP\ /\ (TP\ +\ FN)$(tasa de verdaderos positivos, donde se refleja la proporción de aciertos que obtiene el modelo respecto del total de casos realmente positivos, bajo un determinado punto de corte) en el eje Y (vertical); y **FPR:** $FP\ /\ (FP\ +\ TN)$(tasa de falsos positivos, donde se refleja la proporción de errores que obtiene el modelo sobre el total de casos que son realmente negativos bajo cierto punto de corte) en el eje X (horizontal).

  > Cada punto del gráfico de la curva ROC surge del uso de un determinado threshold, punto de corte o umbral discriminante. Iterando sobre umbrales que van desde 0 hasta 1 y trazando resultados de TPR y FPR, queda dibujada la curva ROC.

  > Cuanto más cerca esté la curva de la esquina superior izquierda, mayor será el potencial del modelo para lograr una separación adecuada de clases. En el extremo, la curva ROC de un clasificador aleatorio será la diagonal del gráfico, donde el TPR = FPR y entonces AUC = 0.5. Este último caso correspondería al mencionado “Algoritmo de la moneda”, donde se asigna género de forma aleatoria sin desarrollar inteligencia alguna sobre dicha técnica.

  > Por otro lado, si la curva ROC se inclinara hacia la esquina inferior derecha, esto puede significar un error en la etiqueta de las clases a predecir. En particular, el punto 0-0 es aquel donde los verdaderos positivos son 0 (TP = 0), es decir las etiquetas predichas correctamente son nulas, y los falsos positivos también son 0 (FP = 0), es decir no hubo etiquetas predichas como clase 1 (positivas) de manera incorrecta. Esto es porque en este punto no hay predicciones de clase 1. El threshold o valor de corte aquí es 1 (muy alto), impidiendo predecir a una observación como clase 1.

  > A medida que el threshold disminuye se es menos exigente para clasificar a un registro como positivo

(o de clase 1), y empiezan a aparecer verdaderos positivos y falsos positivos. Por su parte, el punto 1-1 en el otro extremo de la caja, indica que se es poco exigente para clasificar observaciones como clase 1. No hay falsos negativos (FN = 0) ni verdaderos negativos (TN = 0) ya que no se predice ninguna observación como 0, son todas 1 (positivas). El punto de corte aquí es 0 (muy bajo). **>** Por lo tanto, un modelo que predice lo más acertadamente posible es aquel que presenta una curva ROC que rodea a la "caja" en la que está contenida, pegada a los ejes.

De este modo, el área bajo la curva descrita resulta de gran relevancia al momento de comparar métricas de modelos, ya que la misma proporciona un solo valor escalar que resume la performance de un modelo de aprendizaje automático. Un buen modelo es considerado como tal cuanto más cerca de 1 se encuentre el AUC, implicando buen potencial para separar las clases.

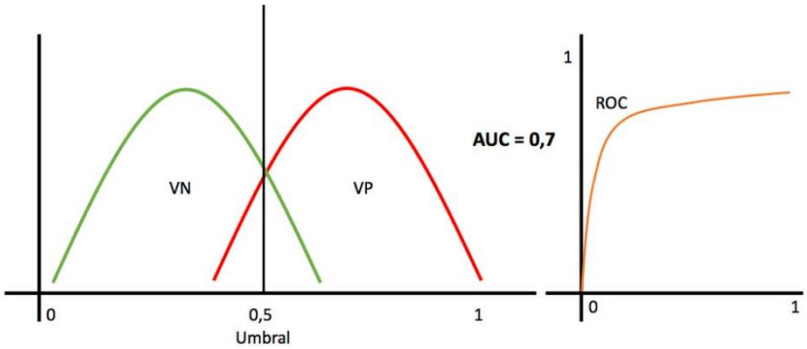


Figura 17: Ejemplo AUC igual a 0,7[13]

El foco estará puesto en el área bajo la curva ROC, ya que mide qué tan bien se clasifican las predicciones y mide la calidad de las predicciones del modelo sin tener en cuenta qué umbral de clasificación se elige. En este caso, es relevante la óptima separación de clases, pues se pretende ofrecer calidad en los datos además de cantidad. Por lo que un valor cercano a 1 en cuanto a AUC será fundamental. Un AUC de 1 significa que hay 100% de probabilidad de que el modelo pueda distinguir entre clase positiva y negativa. Además de dicha métrica, se evaluará y comparará a la métrica de recall ya que, como se mencionó anteriormente, más cantidad de registros predichos implica mayor revenue; precisión ya que no se pierde de vista el nivel de porcentaje de registros predichos correctamente (se observarán curvas precisión - recall obtenidas en cada modelo); el F1-Score como promedio de ambas y el accuracy como métrica base.

5.3. Atributos de User Agent

Se comienza analizando a features desde User Agent. En cuanto a este tipo de características se cuenta con información sobre sistema operativo, versión del mismo, tipo de dispositivo, navegador utilizado, modelo, antigüedad y marca.

A continuación se muestran resultados sobre análisis exploratorio, feature engineering y primeros modelos de predicción de género usando dichos atributos.

[13] *Fuente: https://aprendeia.com/curvas-roc-y-area-bajo-la-curva-auc-machine-learning/*

5.3.1. Análisis exploratorio y Feature Engineering

A los fines de comprender la naturaleza de los datos a utilizar, se detallan diferentes insights encontrados bajo la exploración exhaustiva del dataset a la par de la creación de otras columnas como parte del feature engineering.

El dataset tiene un total de 487.637 filas y 5 columnas. Cada columna posee la siguiente cantidad de valores únicos.

device_idx	88.809
label	2
feature_type	9
feature_detail	79
age	6

Tabla 10: Descripción del dataset

En cuanto al porcentaje de dispositivos únicos en cada uno de los tipos de features, se observa que no todos tienen toda la información de lo que respecta a User Agent.

Sistema Operativo	*99.92%*
Navegador	*99.75%*
Antigüedad	*96.48%*
Versión del Sistema Operativo	*96.05%*
Marca	*33.33%*
Modelo	*23.36%*

Tabla 11: Porcentaje de registros por tipo de atributo

5.3.1.1. Análisis de Modelo de Dispositivo

En lo que respecta al modelo de dispositivos, en la muestra estudiada casi el 6% del total de dispositivos tiene modelo Samsung, seguido por Iphone con el 1% de dispositivos.
En particular, los primeros valores observados, al ordenar los porcentajes de dispositivos (porcentaje sobre el total de dispositivos únicos), de mayor a menor son los siguientes:

Samsung SM-G532M	1.61%
Samsung SM-J710MN	1.56%
Samsung SM-G610M	1.36%

Samsung SM-A105M	1.30%
iPhone	1.00%
Samsung SM-J701M	0.96%
Samsung SM-A205G	0.91%
Samsung SM-A505G	0.83%
Samsung SM-G570M	0.78%
Samsung SM-A515F	0.76%

Tabla 12: Porcentaje de registros por modelo de dispositivo

Al analizar la distribución de la variable, se obtiene que la mayoría de los dispositivos se concentran en pocos modelos. En el eje x del siguiente gráfico se observa la cantidad de dispositivos únicos, y en el eje y la densidad que indica cómo se distribuyen los distintos registros según modelo asociado.

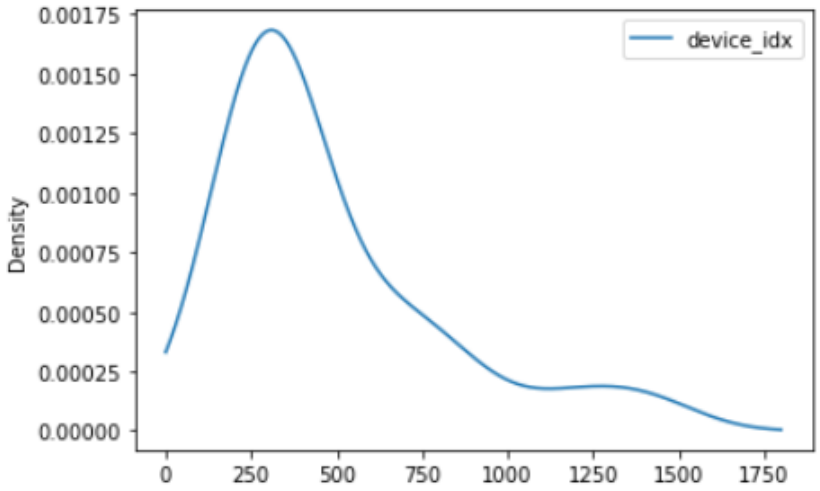


Figura 18: Distribución de dispositivos según Modelos

En cuanto a la relación del feature "*ismobile*" (si es celular) con este feature "modelo", se obtuvo que muchos celulares no tienen modelo asociado. Más del 12% de los dispositivos celulares es modelo Samsung. El 2,3% de los dispositivos celulares son iPhone.

En cuanto a la relación del feature "*istablet*" (si es tablet) con el feature "modelo", casi el 1% de los dispositivos tablet tienen modelo *Samsung SM-J260M.* Luego el 0.3% de los dispositivos tablet tienen modelo Samsung SM-A107M. De todos modos, muy pocos dispositivos tablet tienen modelo asociado, como se aprecia en la siguiente tabla.

Samsung SM-J260M	0.92%
Samsung SM-A107M	0.31%

Samsung SM-A115MN	0.15%
Samsung SM-J710MN	0.15%
Samsung SM-A715F	0.15%
XiaoMi Redmi Note 7	0.15%
Samsung SM-G975F	0.15%

Tabla 13: Porcentaje de registros con tablet por modelo

Respecto a la relación del feature "*ispc*" (si es computadora) con este feature "modelo", el 0.5% de los dispositivos pc tienen modelo Mac. Como en los casos anteriores, muy pocos dispositivos pc tienen modelo asociado.

Mac	0.53%
Samsung SM-G955F	0.00%
moto e5	0.00%
Samsung SM-G570M	0.00%
Samsung SM-A505G	0.00%
Moto E 4	0.00%

Tabla 14: Porcentaje de registros con computadora por modelo

Al observar modelos de dispositivos en relación a la variable target género, se aprecia que existe orientación femenina hacia algunos modelos, como así también orientación masculina por algunos otros. Sin embargo, varios modelos tienen asociados proporciones equilibradas en cuanto a género.

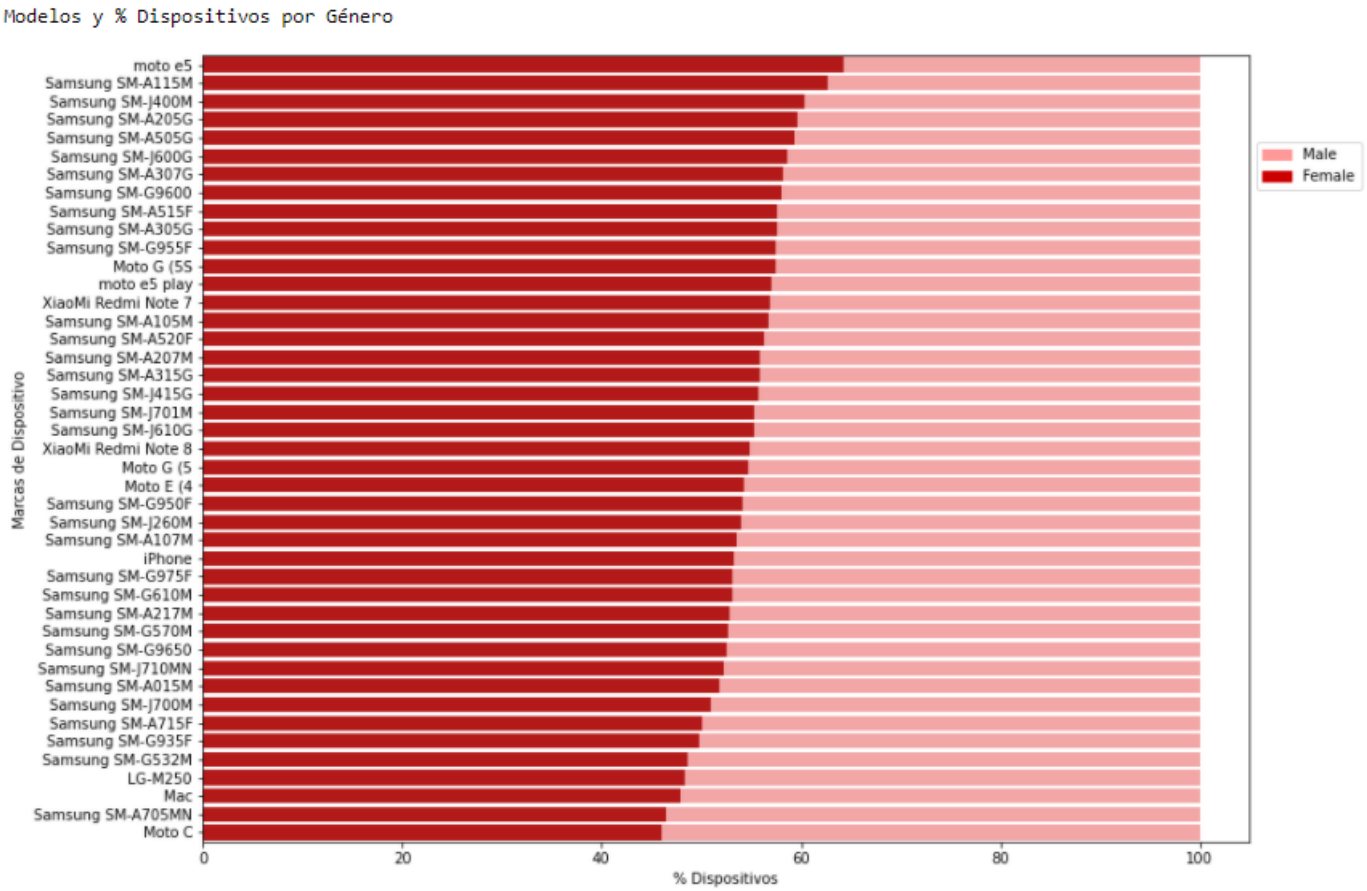


Figura 19: Modelos de Dispositivos y Porcentaje de registros según Género

Entre los modelos de dispositivos disponibles hay algunos que presentan muy poca cantidad de dispositivos asociados, lo cual llevó a la decisión de crear una nueva variable que reemplace a la original donde se agrupa como "Otros" a los valores con menos del 0.33% de dispositivos únicos asociados, ya que a partir de este valor, las cantidades de dispositivos, las cantidades se suman marginalmente.

5.3.1.2. Análisis de Marca de Dispositivo

Así se presentan los porcentajes de dispositivos únicos por marca:

Samsung	24.04%
Apple	3.91%
XiaoMi	2.22%
Huawei	1.39%
LG	1.13%
Motorola	0.64%

Tabla 15: Porcentaje de registros por marca

Es decir, la mayor parte de los dispositivos del dataset tiene marca Samsung. Esto se visualiza en el siguiente gráfico, donde pocas marcas concentran a la mayor parte de la población de la muestra bajo estudio.

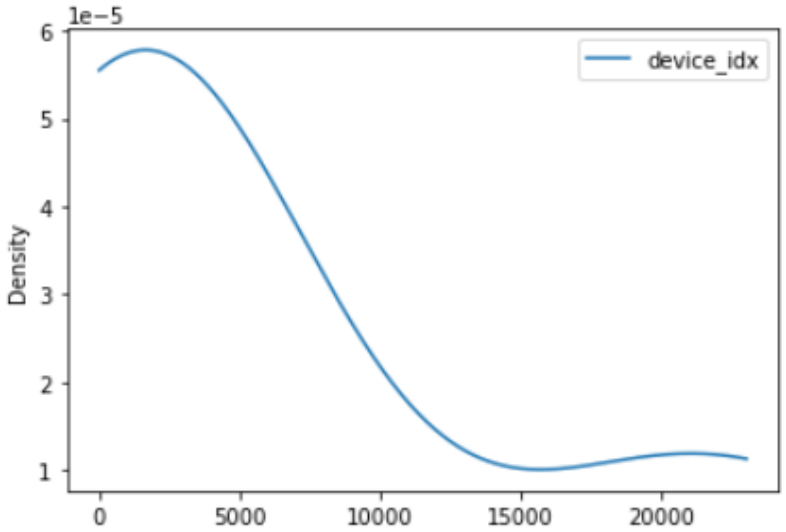


Figura 20: Distribución de Dispositivos por Marca

Observando la relación de marca con respecto a ser de tipo celular, se obtiene que el 55.6% de los dispositivos celulares tienen marca Samsung.

En detalle:

Samsung	55.60%
XiaoMi	5.16%
Huawei	3.24%
LG	2.62%
Apple	2.46%
Motorola	1.48%

Tabla 16: Porcentaje de registros de tipo celular, por marca

Cuando se observa a la variable dicotómica “istablet”, el 19% de los dispositivos tablet tienen marca Apple. Luego el 15.6% de los dispositivos tablet tienen marca Samsung. En este caso, gana Apple en mayoría de dispositivos asociados.

Apple	19.14%
Samsung	15.62%

XiaoMi	0.61%
LG	0.61%
Huawei	0.15%

Tabla 17: Porcentaje de registros de tipo tablet, por marca

Por su parte, casi el 5% de los dispositivos tipo PC son marca Apple, siendo esta marca, al igual que en el caso anterior, mayoritaria entre dispositivos.

Apple	4.91%
Samsung	0.04%
XiaoMi	0.01%
LG	0.00%
Huawei	0.00%

Tabla 18: Porcentaje de registros de tipo computadora, por marca

En un análisis bivariado con la variable target, se observa que no hay grandes discrepancias entre géneros femeninos y masculinos y la marca del dispositivo. Por tal motivo no se hará una variable adicional agrupando valores. Se dejará que los modelos de la siguiente etapa seleccionen (o no) variables relacionadas a este tipo de feature.

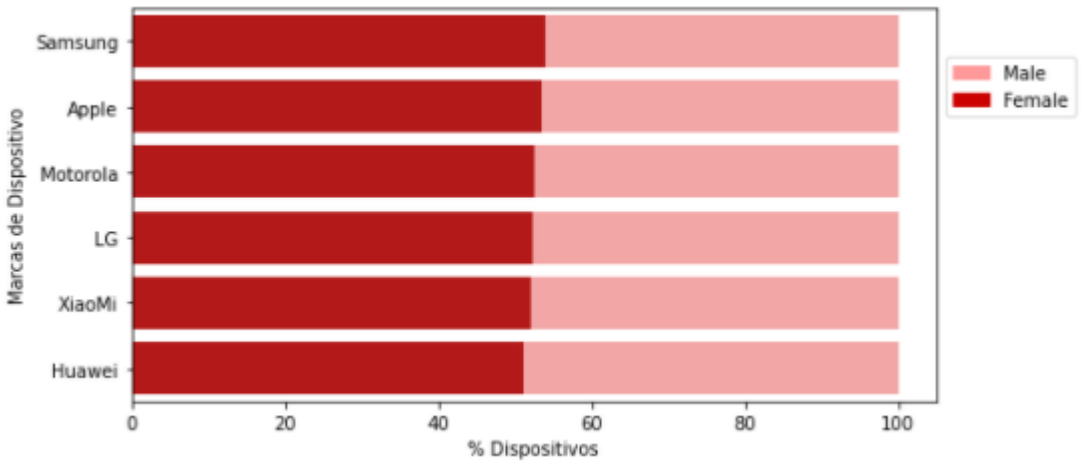


Figura 21: Marcas de Dispositivos y Porcentaje de registros según Género

5.3.1.3. Análisis de Sistema Operativo

Como se observa en la siguiente tabla, alrededor del 95% del total de sistemas operativos pertenece a Windows o Android.

Windows	52.8%
Android	42.48%
Mac OS X	2.77%
iOS	1.14%
Linux	0.73%

Tabla 19: Porcentaje de registros por Sistema Operativo

En cuanto a la relación con el género, el único sistema operativo que tiene cierto sesgo hacia uno de los niveles es Linux.

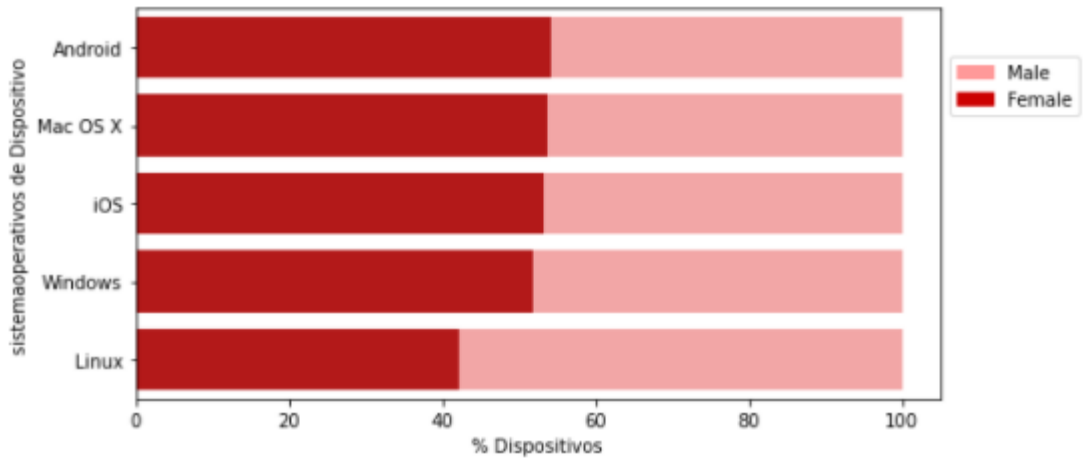


Figura 22: Sistemas Operativos de Dispositivos y Porcentaje de registros según Género

5.3.1.4. Análisis de Versión del Sistema Operativo

Las versiones de sistema operativo de los dispositivos se concentran en Windows 10, Android 10 y Windows 7.

Se observa a continuación el porcentaje de dispositivos por versión de sistema operativo. Pocas versiones concentran gran cantidad de dispositivos.

Windows 10	37.96%
Android 10	16.55%
Windows 7	11.33%
Android 9	8.38%
Andorid 8	8.29%
Android 6	3.66%

Android 7	3.55%
Windows 8	2.54%
Android 5	1.38%
Windows XP	0.85%
Android 4	0.67%
iOS 14	0.55%
iOS 13	0.33%

Tabla 20: Porcentaje de registros por Versión del Sistema Operativo

Al hacer el cruce con el campo género, puede apreciarse que las versiones de sistema operativo presentan cierta variabilidad.

La versión Windows XP tiene un notorio mayor porcentaje de género masculino, por ejemplo. Luego se destaca iOS 13 con un mayor porcentaje femenino que masculino.

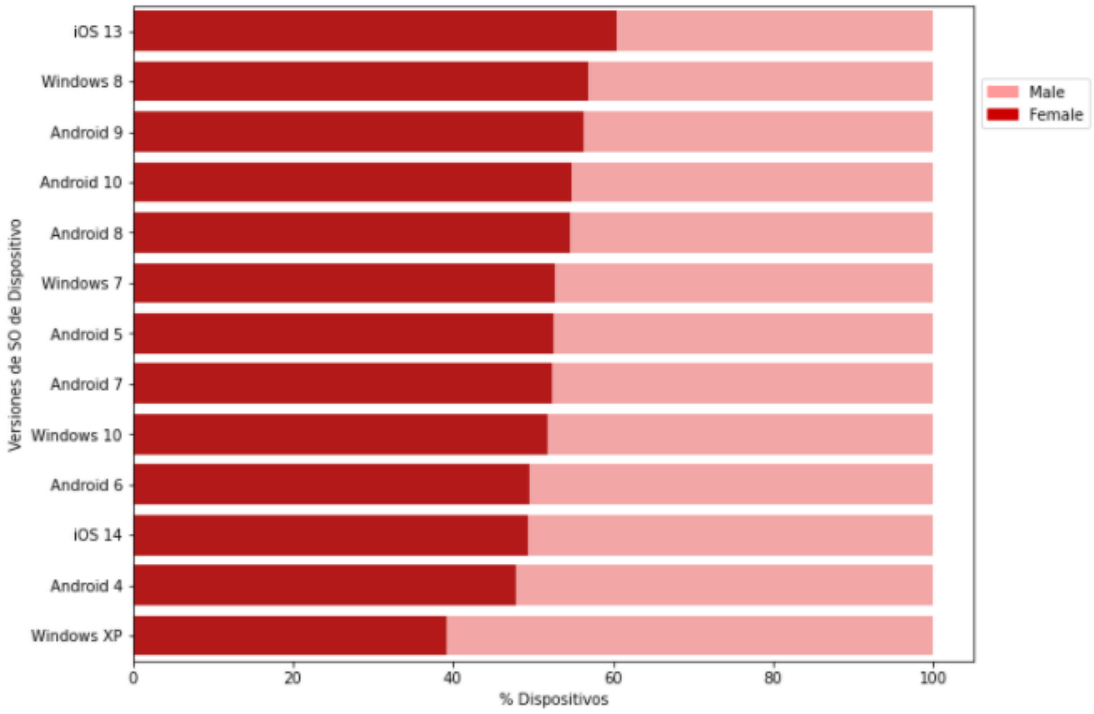


Figura 23: Versiones de Sistemas Operativos de Dispositivos y Porcentaje de registros según Género

Sin embargo, hay versiones que tienen asociados muy pocos dispositivos. Por este motivo, se agrupa en una nueva variable bajo el valor “Otros” a aquellos valores con menos del 2.5% de dispositivos únicos asociados. Al igual que en el caso de modelos de dispositivos, dicho umbral de decisión se tomó en base

a la observación de la suma acumulada de apariciones de cada valor sobre el total de filas del dataset. Luego del top 8, las cantidades se suman marginalmente.

Por otro lado, el feature de antigüedad ofrecido en los datos tiene coherencia con los años de cada versión del sistema operativo, por lo que se utilizará dicha variable como atributo acerca de qué tan nuevos o viejos son los dispositivos.

5.3.1.5. Análisis de Navegador

De este análisis se obtiene que casi el 79% del total de navegadores utilizados por los usuarios proviene de Chrome, luego el 7% proviene de Social App, seguido por el 6,4% de brand browser y finalmente otro 6% repartido entre Firefox, Safari, Edge, IE y Google.

Chrome	78.77%
SocialApp	7.16%
Brand browser	6.41%
Firefox	5.23%
Safari	0.87%
Edge	0.73%
IE	0.32%
Google	0.28%

Tabla 21: Porcentaje de registros Navegador

Al observar la variable género en cruce con el feature del navegador, se aprecia una inclinación masculina en Firefox e IE y luego una inclinación femenina por SocalApp, siendo parejo en los demás casos.

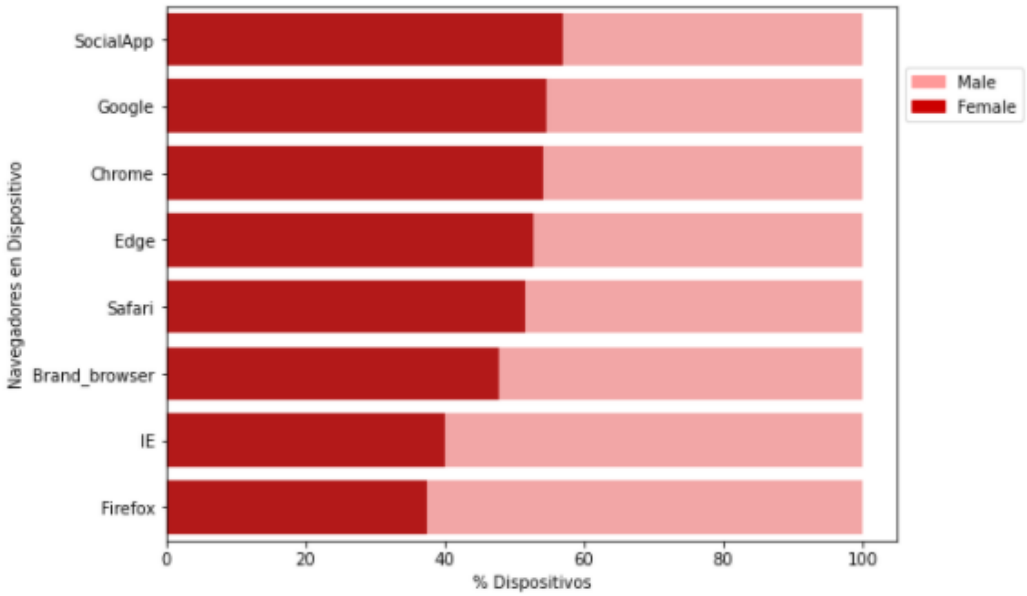


Figura 24: Navegador de Dispositivos y Porcentaje de registros según Género

Sin embargo, siguiendo la misma lógica que versiones de sistema operativo y modelo de dispositivo, existen valores con muy pocos dispositivos asociados que serán taggeados bajo el valor "Otros" en una nueva variable. El umbral de decisión para esta agrupación se tomó en base a la suma acumulada de apariciones en cada valor de navegador sobre el total de filas del dataset. Luego del top 6, las cantidades de dispositivos por navegador se suman marginalmente por lo que esos navegadores serán enmarcadas en "Otros".

5.3.1.6. Análisis de Antigüedad de los dispositivos

El 58% de los dispositivos es nuevo, el 28% es viejo, y el 10.2% es muy viejo. Del 3.5% restante no hay información de antigüedad.

En cuanto a la variable target, se aprecia que en los dispositivos muy viejos, el porcentaje de usuarios es más masculino que femenino. En los restantes nuevo y viejo, la proporción de género es pareja.

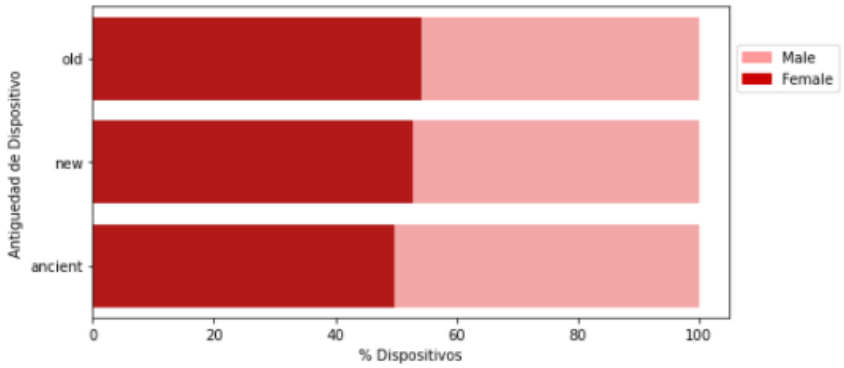


Figura 25: Antigüedad de Dispositivos y Porcentaje de registros según Género

5.3.1.7. Análisis de Is Mobile (Celular)

El 43% del total de dispositivos únicos es Celular, y el género se encuentra parejo en cuanto a posesión de este tipo de dispositivo.

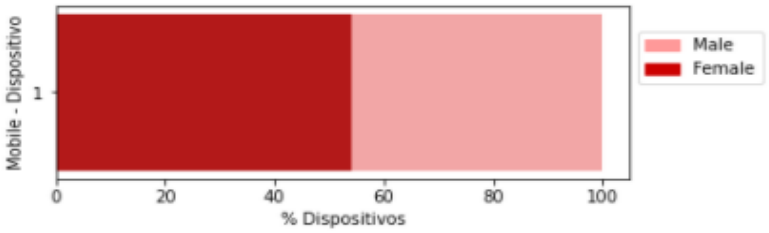


Figura 26: "Es Mobile" y Porcentaje de registros según Género

5.3.1.8. Análisis de Is Tablet

El 0.74% del total de dispositivos únicos es Tablet, y el género se encuentra inclinado a femenino en cuanto a posesión de este tipo de dispositivo.

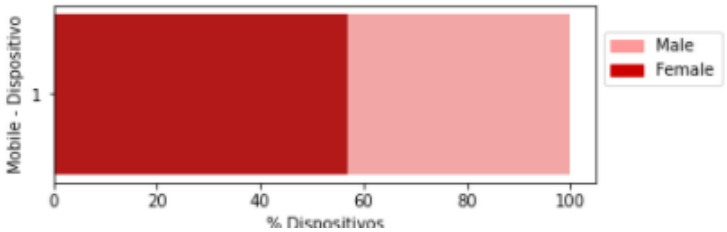


Figura 27: "Es Tablet" y Porcentaje de registros según Género

5.3.1.9. Análisis de Is Pc

El 56.4% del total de dispositivos únicos es PC, y el género se encuentra parejo en cuanto a la posesión de este tipo de dispositivo, al igual que en el caso del tipo mobile.

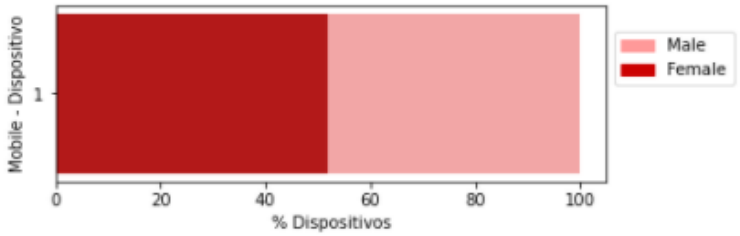


Figura 28: "Es PC" y Porcentaje de registros según Género

5.3.2. Modelos y Resultados con atributos de User Agent

Antes de llevar los datos a modelos de machine learning, se realiza un análisis sobre el factor de inflación de la varianza (VIF: Variance Inflation Factor) en todas las variables que quedaron en la matriz final. El VIF es una medida de redundancia de variables y ayuda a decidir qué variables pueden no servir por perjudicar el poder explicativo de los modelos de clasificación. Cuantifica la intensidad de la multicolinealidad, midiendo la asociación lineal entre cada regresor y el resto. Concretamente, el factor

de inflación de la varianza para beta estimado es: $FIVi = 1/(1 - R^2i)$ donde R²i es el coeficiente de determinación de la ecuación de regresión de mínimos cuadrados que tiene a Xi como una función de las demás variables explicativas. Un R² chico de esa regresión auxiliar implica, por lo tanto, baja multicolinealidad.

5.3.2.1.Baseline

Un aspecto fundamental a considerar antes de entrenar modelos es poder compararlos con un modelo *Baseline*. En este caso, se usó un modelo Naive Bayes. (En el apartado de *Resultados* se mostrará un cuadro comparativo de métricas en todos los modelos probados) La performance del **Naives Bayes** se puede observar en los siguientes gráficos.

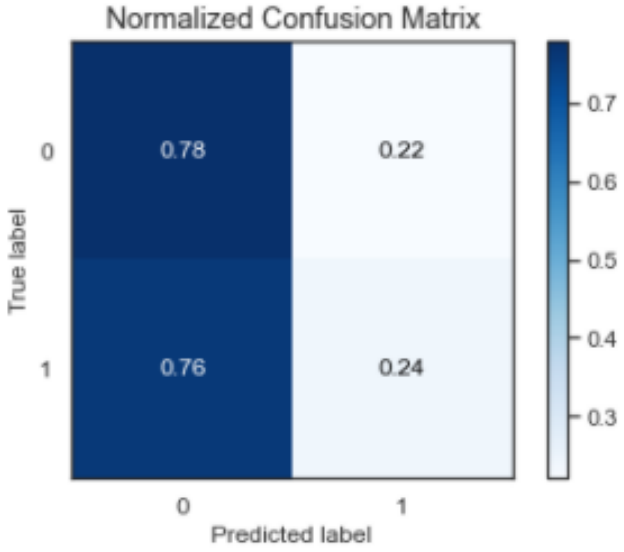


Figura 29: Matriz de Confusión en Modelo Naive Bayes

Por un lado, la matriz de confusión indica que el modelo predijo más género masculino que femenino.

Por su parte, la curva ROC y el área bajo la misma indican que el modelo tuvo una performance poco óptima, dada el pequeño área bajo la curva que se observa. En particular el AUC es de 0.53.

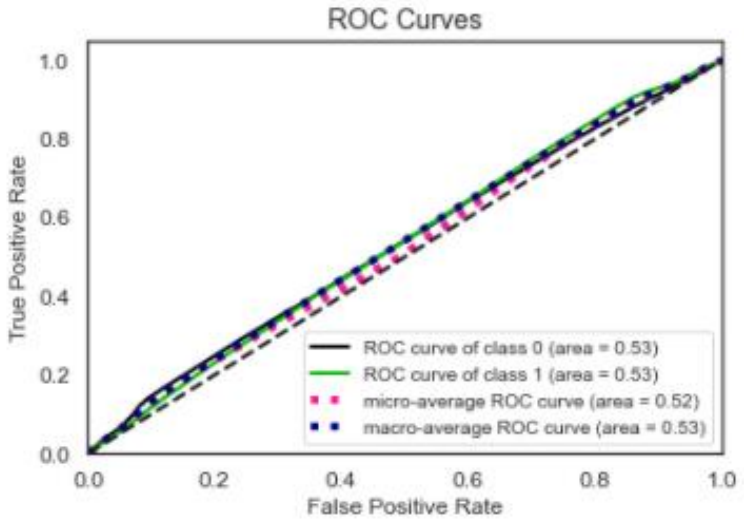


Figura 30: Curva ROC y AUC en Modelo Naive Bayes

5.3.2.2. Regresión Logística

El modelo de **regresión logística**, luego de ajustar hiperparámetros mediante el método Grid Search, arrojó resultados que se observan en la siguiente matriz de confusión. La predicción del modelo se inclinó más hacia el género femenino que masculino.

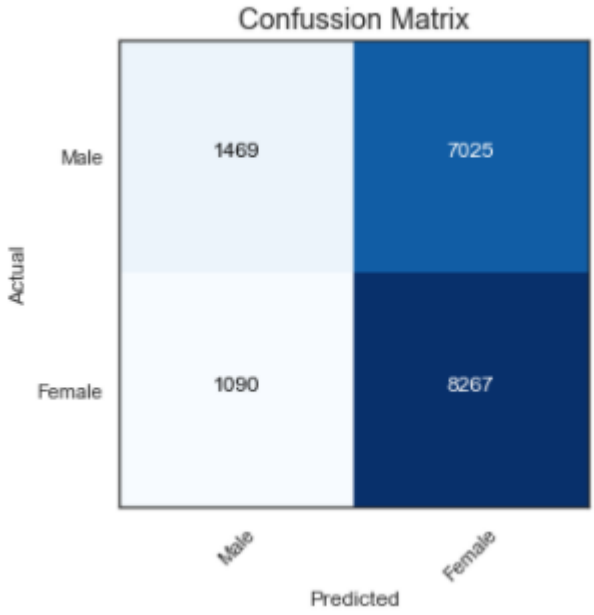


Figura 31: Matriz de Confusión en Modelo Regresión Logística

Antes de observar las curvas Precisión - Recall, cabe destacar la manera de interpretar este tipo de gráficos.

Por lo general, cuando el threshold es bajo, se obtiene un recall mucho más alto que la métrica de precisión. Esto hace sentido recordando la fórmula de cada uno, TP/(TP+FN) para recall y TP/(TP+FP) para

precisión. Cuando el threshold es más bien bajo, se es poco exigente para clasificar observaciones positivas. Habrá muy pocos falsos negativos, haciendo alto al recall. En cambio, habrá más verdaderos positivos y falsos positivos también, pues se predice mucho la clase positiva. Eso hace grande al denominador en precisión y por ende achica a dicha métrica. Cuando el threshold es más bien alto, se observa recall bajo y precisión más alta. Se es más exigente para clasificar a una observación como positiva. Habrá pocos verdaderos positivos y pocos falsos positivos, achicando el denominador de la métrica de precisión, haciéndola más alta en su totalidad. Y también habrá seguramente más falsos negativos (pues se predecirá más 0 que 1, facilitando el aumento de FN), agrandando el denominador del recall, y por ende, haciéndolo bajar.

En el siguiente gráfico se observa la curva de precisión-recall para el género masculino. Ambas curvas se cruzan entre los thresholds 0.50 y 0.55. Antes de 0.5, la métrica de recall cuando se hace referencia al género masculino como positivo es claramente más alta que la precisión. Luego de dicho umbral, la precisión le gana al recall, hasta juntarse ambos cerca de un threshold de 0.7, donde el punto de corte termina siendo tan alto que no se predicen clases positivas, siendo todas clases negativas (bajo recall y baja precisión, pues no hay verdaderos positivos).

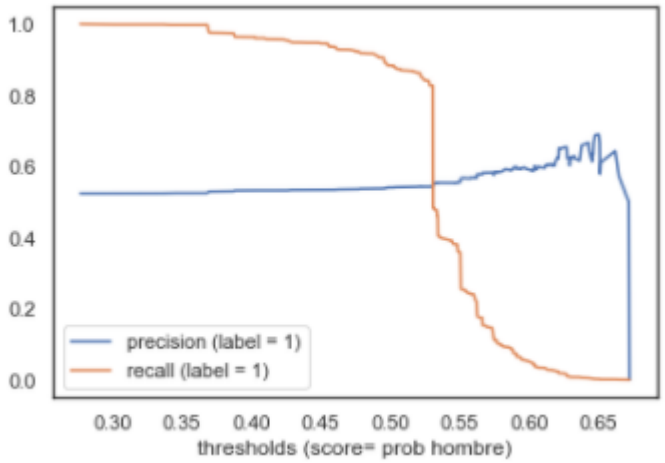


Figura 32: Curva Precisión-Recall en Modelo Regresión Logística para género masculino

Para el caso del género femenino, se obtuvo lo siguiente. Cabe resaltar que para este caso las curvas precisión y recall, tomando como clase positiva al género femenino, no se vuelven a tocar después del primer cruce en el punto de corte 0.45. Antes de 0.45, se predicen más clases positivas que después de dicho punto, lo cual es coherente con el alto recall de un lado y bajo recall del otro, siendo al revés para el caso de la precisión. Aunque esta última métrica no presenta aumentos tan acentuados.

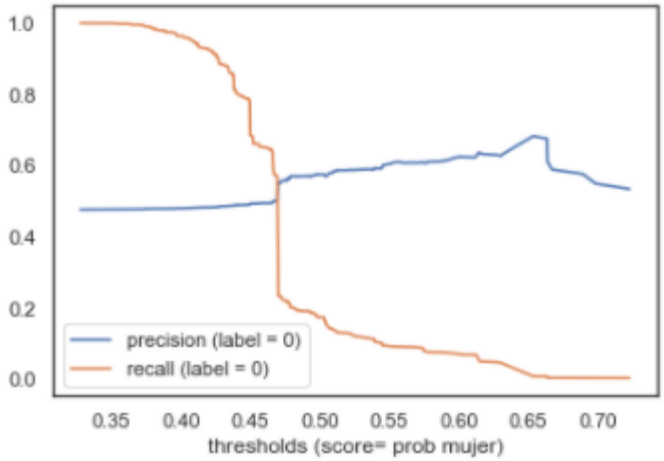


Figura 33: Curva Precisión-Recall en Modelo Regresión Logística para género Femenino

A modo resumen de precisión y recall resulta interesante ver la curva Precision-Recall para ambas clases. Aquí el Average Precision Recall es 0.55. Este valor es una manera de calcular el área bajo la curva PR o PR AUC, o lo que es lo mismo, el resultado de integrar la curva. Cuanto más se acerque su valor a 1, mejor será el modelo. Por ende, la performance de la regresión logística hace notar que existe espacio de mejora, ya que el área promedio bajo la curva precision-recall está algo alejada de ser 1.

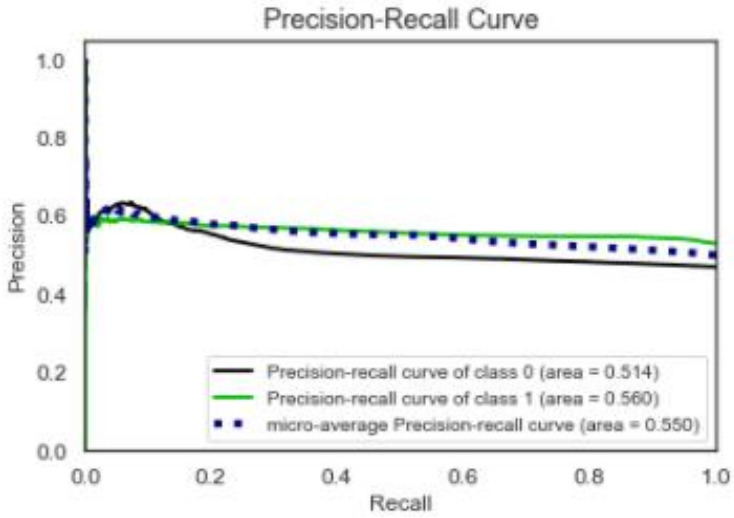


Figura 34: Curva Precisión-Recall en Modelo Regresión Logística

Luego, el área bajo la curva ROC dio algo parecido al modelo baseline, aunque levemente mejor. Sin embargo, el área sigue siendo muy pequeña, poniendo en evidencia una performance de predicción con gran espacio de mejora.

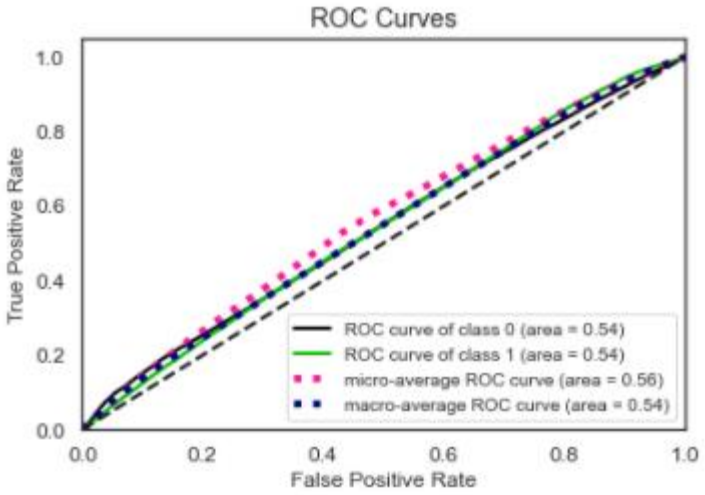


Figura 35: Curva ROC y AUC en Modelo Regresión Logística

Por último, se estudia la "learning curve", que es la curva donde se puede visualizar si existen diferencias sustanciales entre los scores arrojados por el conjunto de train y los arrojados por el conjunto de test bajo la técnica de cross validation, a medida que se toman más observaciones para entrenar a los modelos. Se observa que a medida que aumenta la cantidad de observaciones con las que el modelo entrena, la brecha entre el score obtenido por "train" y el obtenido mediante cross validations se achica. Esto indica que los conjuntos de train y test en cross validation con pocas observaciones no se comportan exactamente igual al momento de clasificar. Pero a medida que aumenta la cantidad de registros utilizados, se puede apreciar que ambos scores se asemejan. Es decir, el modelo entrenado es confiable al momento de interpretar resultados.

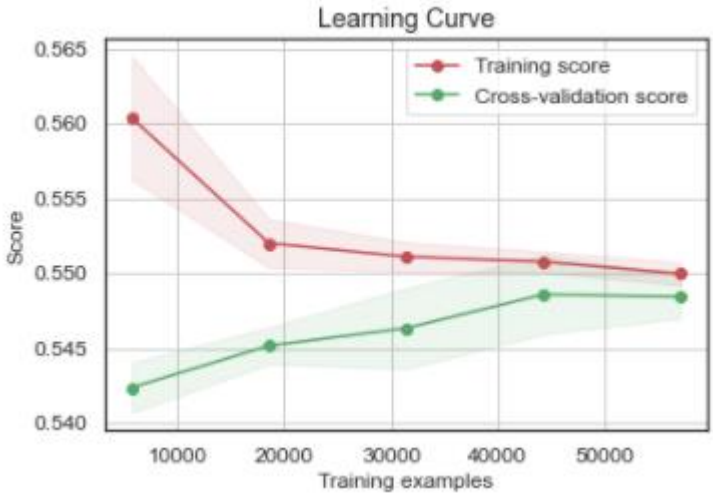


Figura 36: Curva de Aprendizaje en Modelo Regresión Logística

5.3.2.3. Random Forest

Del modelo de **random forest** se obtuvo la siguiente performance.

Luego de ajustar hiperparámetros mediante el método Random Search, arrojó resultados que se observan en la siguiente matriz de confusión. La predicción del modelo se inclinó más hacia el género femenino que masculino como en el modelo anterior.

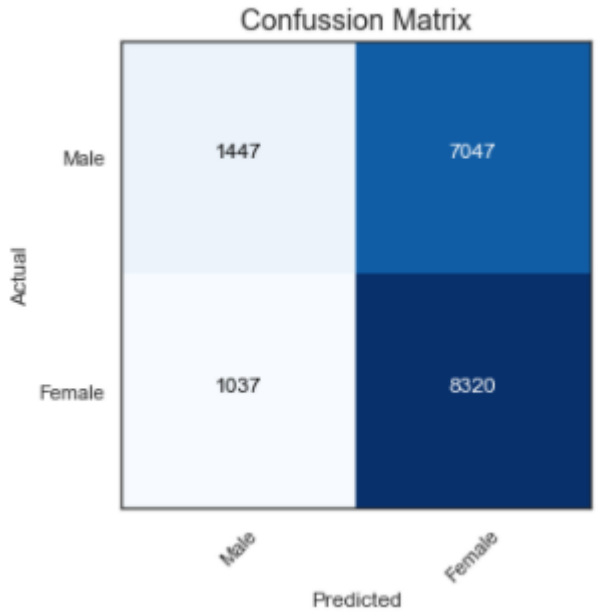


Figura 37: Matriz de Confusión en Modelo Random Forest

La curva de precisión-recall para el género masculino es la siguiente. Se observa que luego del punto de corte 0.5, la métrica de recall empieza a bajar abruptamente, y precisión a subir en menor magnitud.

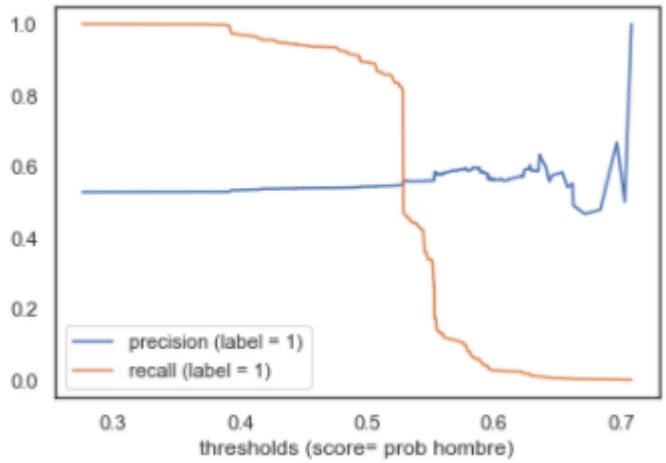


Figura 38: Curva precisión-recal en género masculino, Modelo Random Forest

Para el caso del género femenino, las curvas precisión y recall se comportan parecido al caso masculino, con diferencia en precisión, donde la misma tiene una tendencia alcista después de cruzarse con la curva recall.

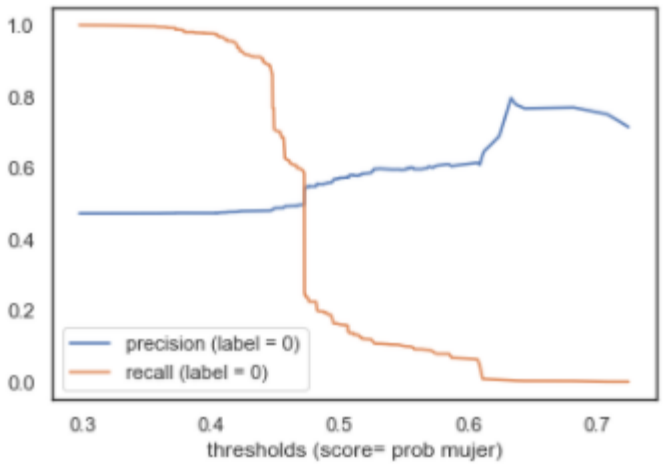


Figura 39: Curva precisión-recall en género femenino, Modelo Random Forest

Por su parte, la curva Precision-Recall para ambas clases. El Average Precision Recall en este modelo random forest es 0.552, lejos del ideal 1 y parecido al resultado en regresión logística.

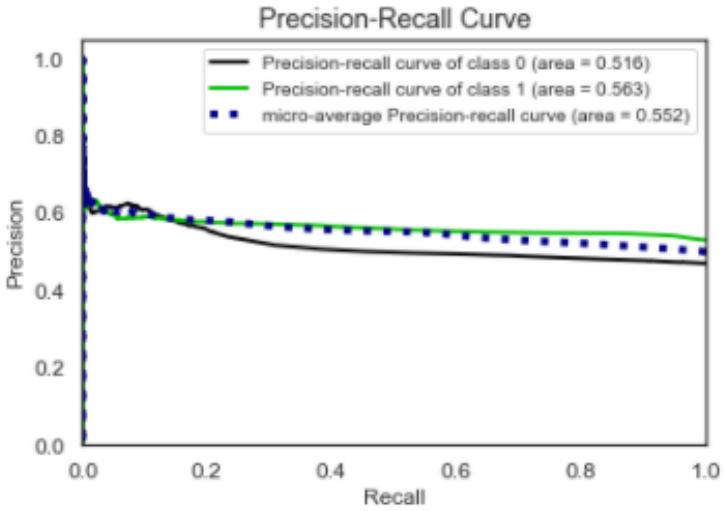


Figura 40: Curva precisión-recall, Modelo Random Forest

El área bajo la curva ROC dio baja, alejada del ideal donde la curva se acerca a los bordes de la caja.

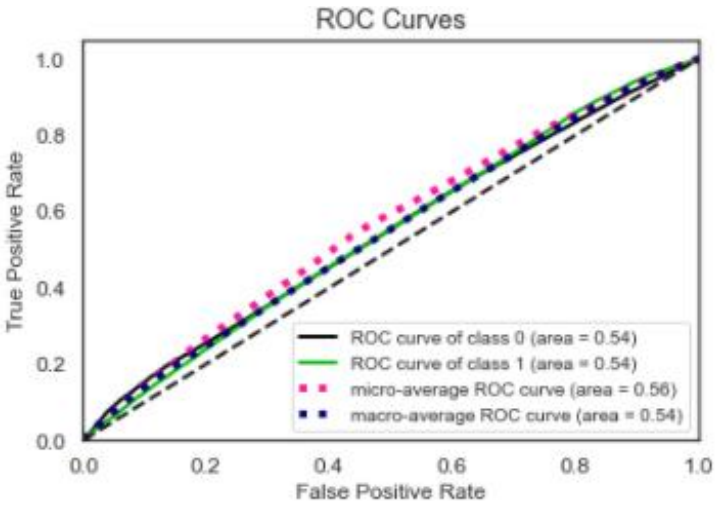


Figura 41: Curva ROC y AUC, Modelo Random Forest

Finalmente, la curva "learning curve" para este caso presenta una pequeña brecha de scores cuando se toma gran cantidad de observaciones para el entrenamiento del modelo, por lo cual se puede inferir que no existe problemas de overfitting en este caso. De hecho, con más cantidad de observaciones, la performance en el conjunto Train es peor, mientras que en el conjunto de validación de Cross Validation la performance mejora. Por lo cual puede decirse que "más es mejor".

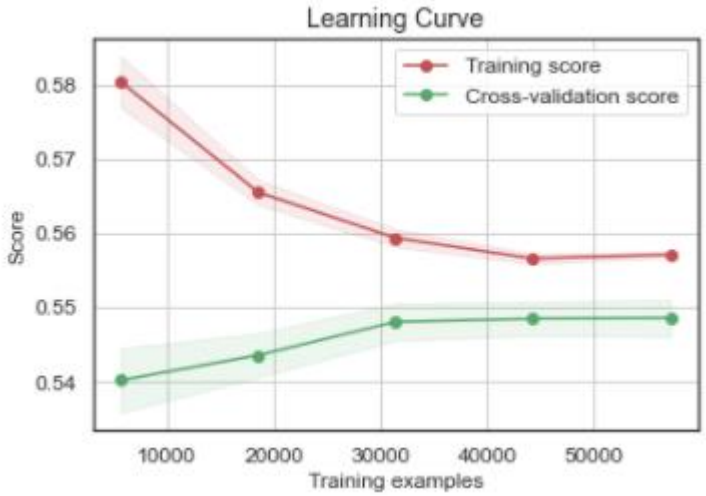


Figura 42: Curva de Aprendizaje, Modelo Random Forest

Se muestra a continuación el ranking de importancia de variables calculado por Random Forest. Los detalles del cálculo que realiza el algoritmo para definir dicha importancia serán explicados en la sección de "Interpretabilidad de Modelos".

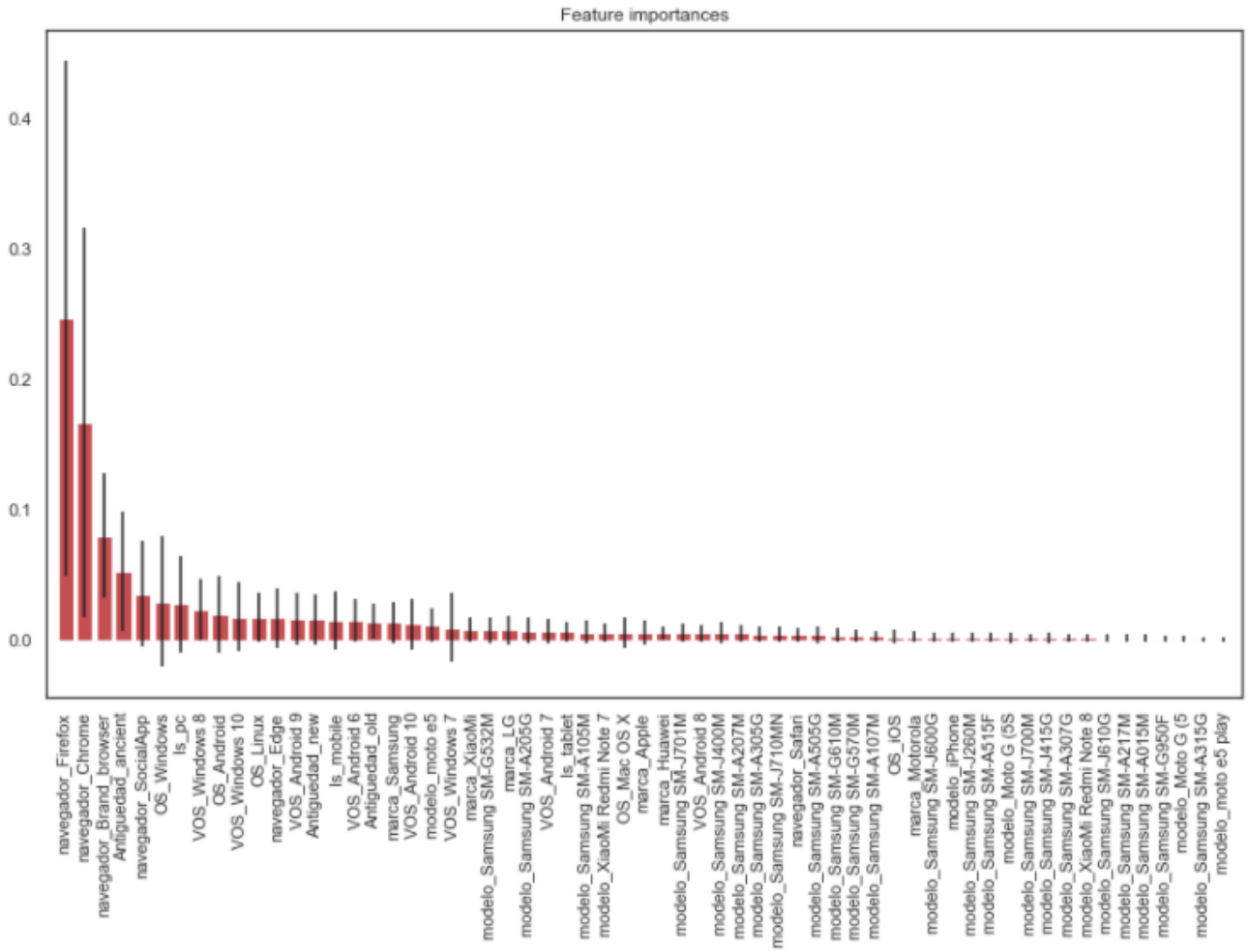


Figura 43: Feature Importance, Modelo Random Forest

Las variables “navegador_Firefox”, “navegador_Chrome” y “navegador_Brand_Browser” fueron las más importantes al momento de clasificar en género para este modelo de árboles aleatorios. Por su parte, varias features de modelos de celular tuvieron baja relevancia al momento de la clasificación de dispositivos. Por ejemplo, algunos modelos de *Samsung* y *Moto* no resultaron grandes predictores.

5.3.2.4.XGBoost

Por último, para lo que respecta a features de User Agent, se corrió un **XGBoost**. Se observa a continuación la performance obtenida.

Luego de ajustar hiperparámetros mediante Random Search, arrojó resultados que se observan en la siguiente matriz de confusión. La predicción del modelo se inclinó más hacia el género femenino.

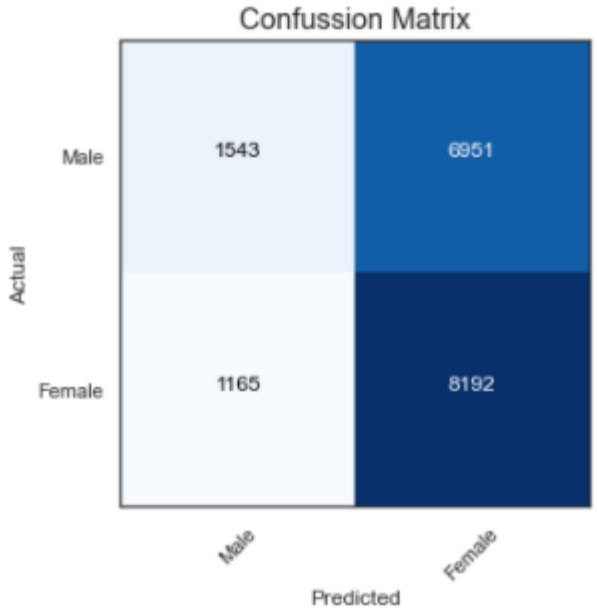


Figura 44: Matriz de Confusión, Modelo XGBoost

La curva de precisión-recall para el género masculino es la siguiente.

Ambas curvas, precisión y recall, se cruzan en un umbral que supera al 0.5, comportándose de manera esperada. A medida que aumenta el threshold, aumenta la precisión y disminuye el recall. Como en los modelos anteriores, el cambio en recall es más acentuado que el cambio en precisión.

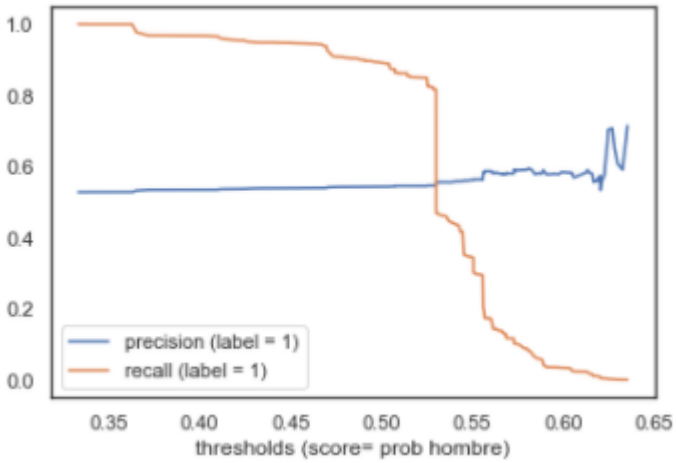


Figura 45: Curva precisión-recall para género masculino, Modelo XGBoost

Para el caso del género femenino, el aumento de la métrica de precisión luego del punto de corte de cruce entre ambas curvas es más acentuado que en caso masculino, dejando en evidencia que el modelo predijo más casos femeninos positivos que masculinos positivos. Es decir, fue más preciso al clasificar mujeres.

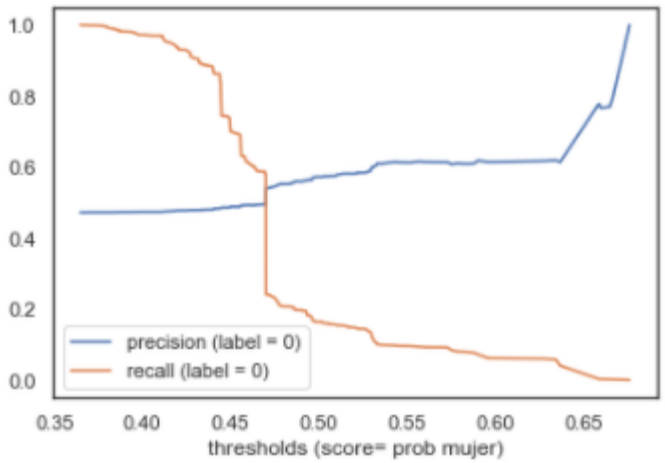


Figura 46: Curva precisión-recall para género femenino, Modelo XGBoost

Por su parte, la curva Precision-Recall para ambas clases presenta un average Precision-Recall de 0.552, alejado del ideal.

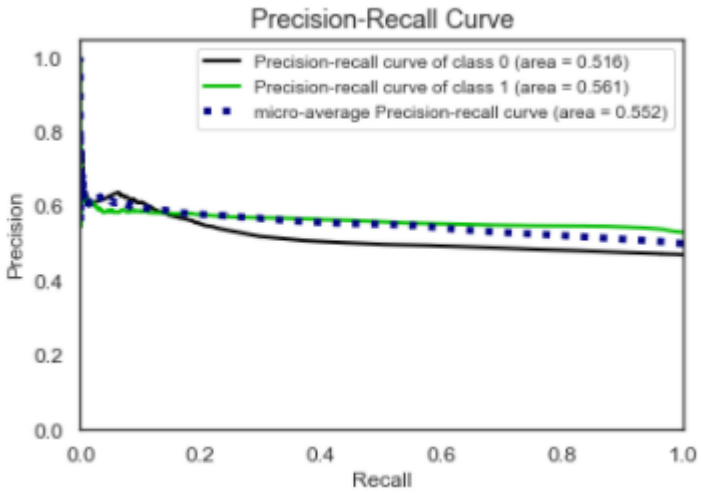


Figura 47: Curva precisión-recall, Modelo XGBoost

El área bajo la curva ROC también muestra bajos valores.

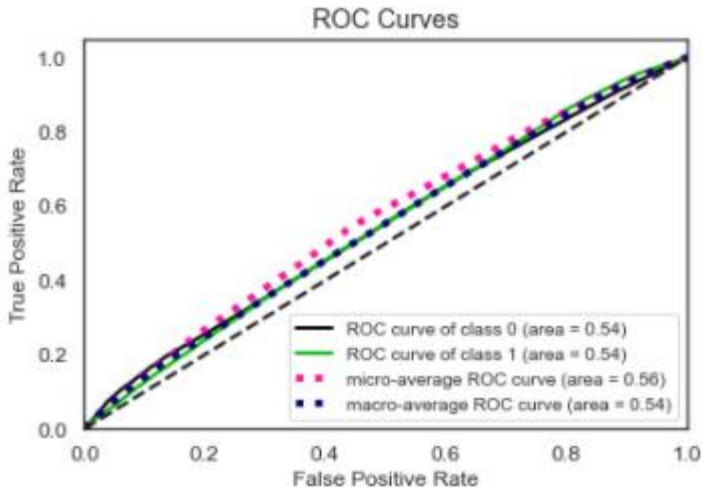


Figura 48: Curva ROC y AUC, Modelo XGBoost

Finalmente, la curva "learning curve" demuestra que no parece existir fenómeno de overfitting en el xgboost entrenado, ni de underfitting, sobre los datos, evidenciándose un ajuste que logra generalizar.

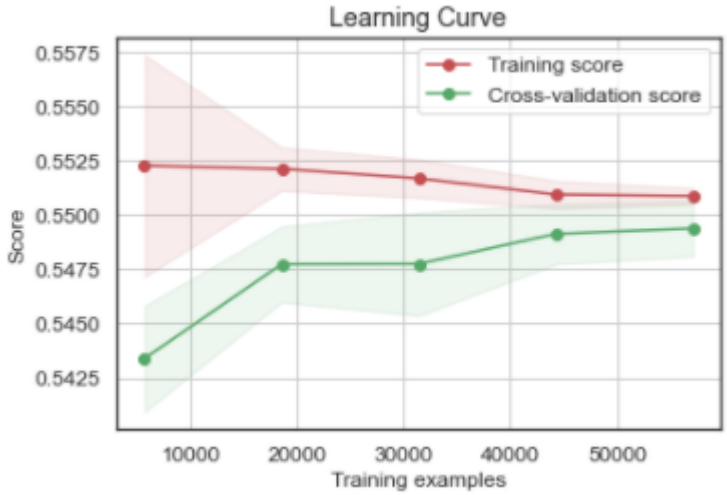


Figura 49: Curva de Aprendizaje, Modelo XGBoost

Los hiperparámetros elegidos para todos los modelos mediante técnicas de grid search y random search se encuentran en la sección "Anexo".

5.4. Atributos de Dominios

El segundo paso para llegar al modelo final que predecirá al género de los usuarios tiene que ver con la inclusión de features de dominio. Se cuenta con la información sobre cuáles dominios navegó cada dispositivo del dataset. A continuación, se muestran resultados sobre análisis exploratorio y feature engineering de estos atributos, para finalizar con modelos de predicción de género donde se suman los atributos de dominio a los atributos de user agent descritos en la sección anterior.

5.4.1. Análisis exploratorio y Feature engineering

Se cuenta con 1587 dominios únicos en todo el dataset, los cuales se repetirán por fila según la cantidad de dispositivos que los hayan visitado. Con 381.200 filas en total para el estudio de dominios, cerca de 90.000 dispositivos únicos, y columnas de device id, label (género masculino o femenino) y el detalle sobre el dominio en sí, se procede a analizar la data.

En primer lugar, luego de verificar la inexistencia de valores nulos y duplicados, se observa la distribución de dispositivos en dominios. Más de 1400 dominios tienen visitas de entre 0 y 600 dispositivos aproximadamente. Es decir, pocos usuarios visitan muchos dominios distintos. Los dominios visitados por más de 10.000 dispositivos son muy pocos. Por lo que pocos dominios concentran a la mayor parte de la población de estudio en términos de visitas. En otras palabras, muchos dispositivos visitan pocos dominios específicos.

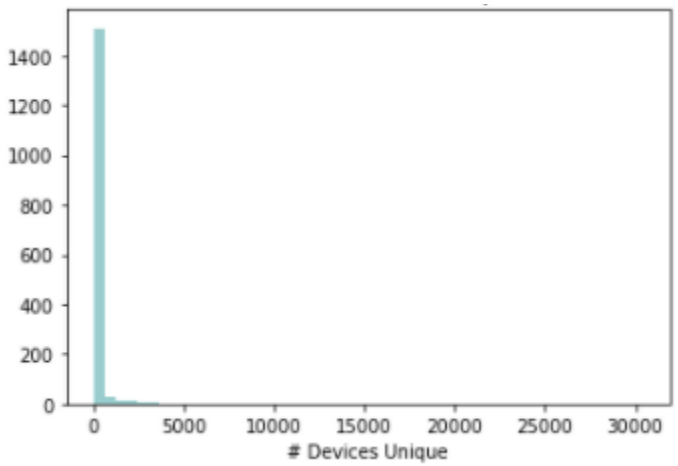


Figura 50: Distribución de dispositivos por dominio

El siguiente gráfico de barras permite visualizar los dominios con más cantidad de visitas, destacándose bumeran.com.ar, buenosaires.gob.ar, zonajobs.com.ar y zonaprop.com.ar. El 30% de los dispositivos visita bumeran.com.ar y casi el 27% de dispositivos visita buenosaires.gob.ar.

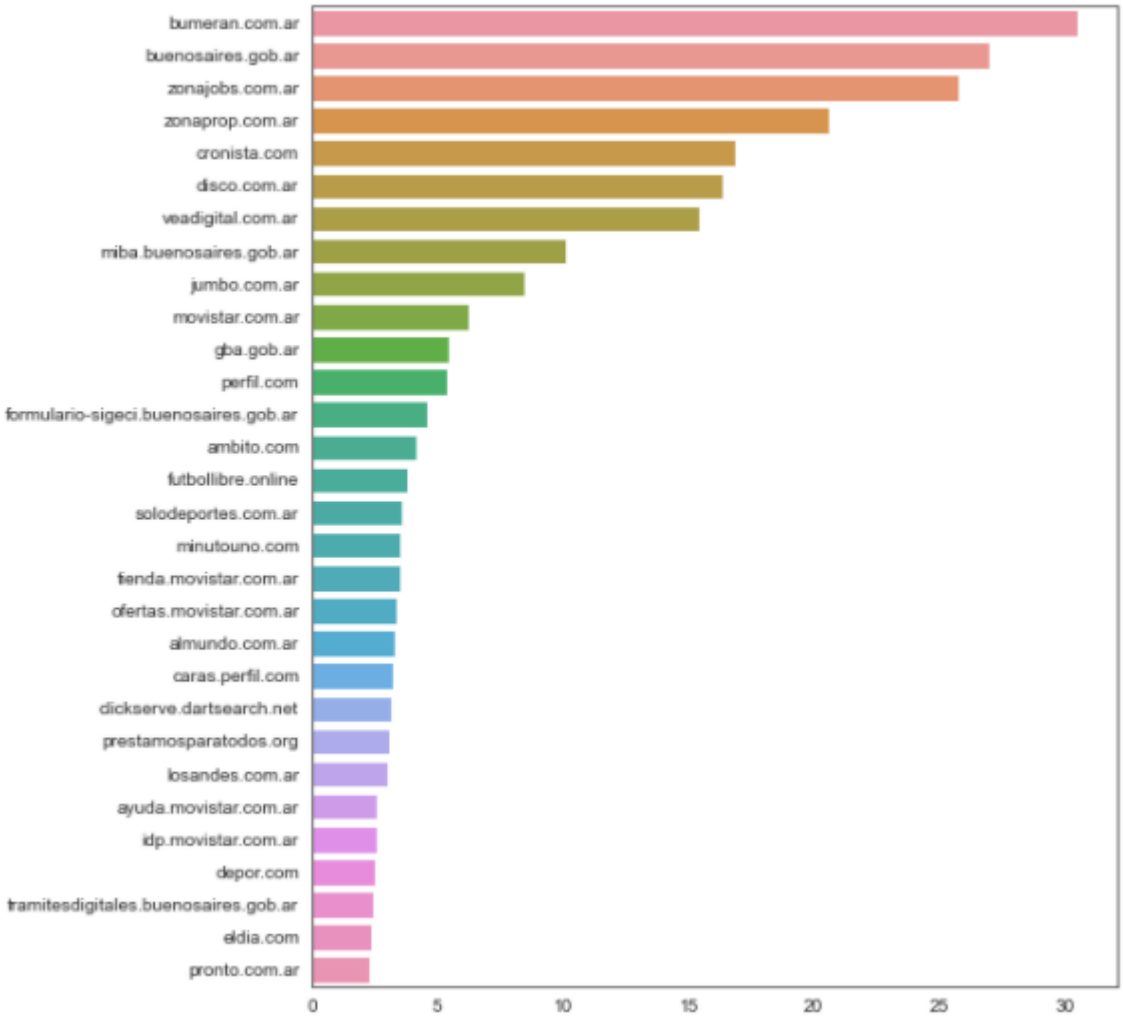


Figura 51: Porcentaje de Visitas (Top 30) por dominio

Dada la gran cantidad de dominios con muy pocas visitas, como parte del análisis y de la ingeniería de atributos se creó una variable nueva que agrupe a aquellos dominios con muy pocos dispositivos en categoría "Otros". Siendo así, se reemplaza a la variable original con los nombres de dominios, por la nueva variable que contiene un valor Otros.

Por otro lado, para simplificar el estudio de esta población y poder visualizar con más claridad intereses de los usuarios de cada dispositivo, se crea una nueva variable "*domain_type*" indicando el tipo de página o rubro al que refiere cada dominio. Para esto se hizo un research exhaustivo de los dominios presentes en el dataset. Se consideraron los siguientes valores para agrupar a los dominios en rubros:

- Busqueda Laboral
- Inmobiliaria
- Noticias
- Deportes
- Entretenimiento
- Alimentos

- Buenos Aires
- Trámites Bancarios
- Farándula
- Viajes
- Automóvil
- Consumo
- Linea Movil
- Educación
- Familia y Hogar
- Salud
- Otros

La variable creada tiene la siguiente distribución.

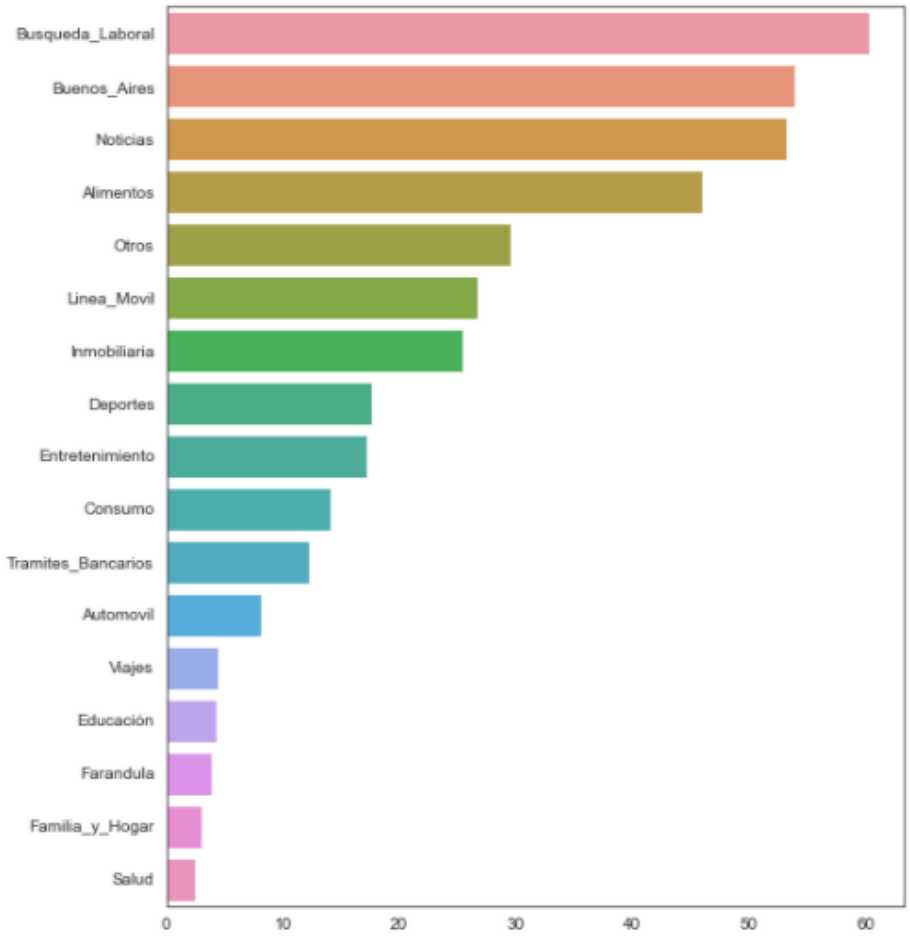


Figura 52: Porcentaje de Visitas por Rubro

Como puede apreciarse en las tablas anteriores, entre la mayor cantidad de visitas por rubro, un 60% de los dispositivos visita páginas sobre empleos, un 54% visita páginas relacionadas a Buenos Aires, un 53% dominios referidos a noticias y casi un 46% visita dominios relacionados con comida y alimentos. Al graficar la distribución de dispositivos en tipos de dominio, se evidencia que pocos tipos de dominios concentran gran parte de la población de estudio.

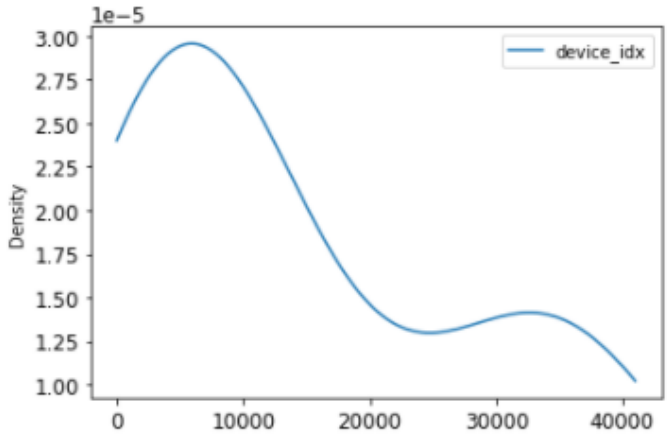


Figura 53: Distribución de dispositivos por Dominio

En cuanto al análisis bivariado, al cruzar la información de rubros de dominios con la variable target de género, se encuentran intereses femeninos y masculinos diferenciados, excepto en los rubros de línea de celular, trámites bancarios y noticias.

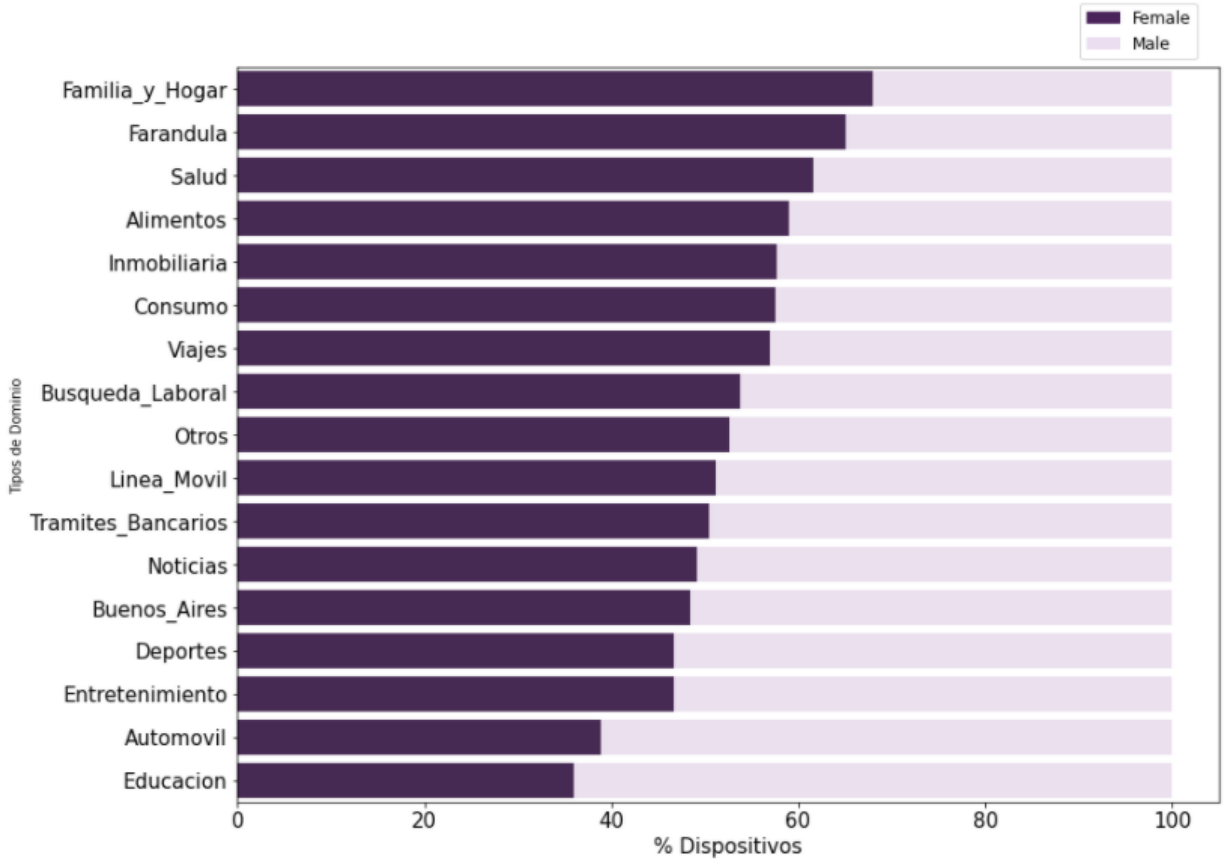


Figura 54: Tipos de Dominio y Porcentaje de Dispositivos por Género

Los rubros de familia, farándula, alimentos y salud muestran una marcada predisposición a ser visitados por el género femenino más que al masculino. En cambio, rubros como automóviles y educación son

visitados más por el género masculino que femenino. A continuación, se detalla la variabilidad en proporción de visitas a cada dominio de los rubros que mayores diferencias presentan en cuanto a la variable target. Los rubros restantes se muestran en detalle en la sección “Anexo” (Anexo 3).

- *Rubro Búsqueda Laboral*

El rubro búsqueda laboral presenta gran variabilidad al cruzarlo con la variable target. Por un lado, los dominios de bumeran y zonajobs tienen asociadas visitas de género femenino más que masculino, como también sitios de trabajo en sitios de Buenos Aires. Por otro lado, los sitios de búsquedas laborales en “emplosit” son más comunes en el género masculino. Este dominio tiene la característica de ser orientado a trabajo en tecnología. Con lo cual, puede inferirse mayor interés específico en empleos en el rubro tecnológico por parte de dicho género.

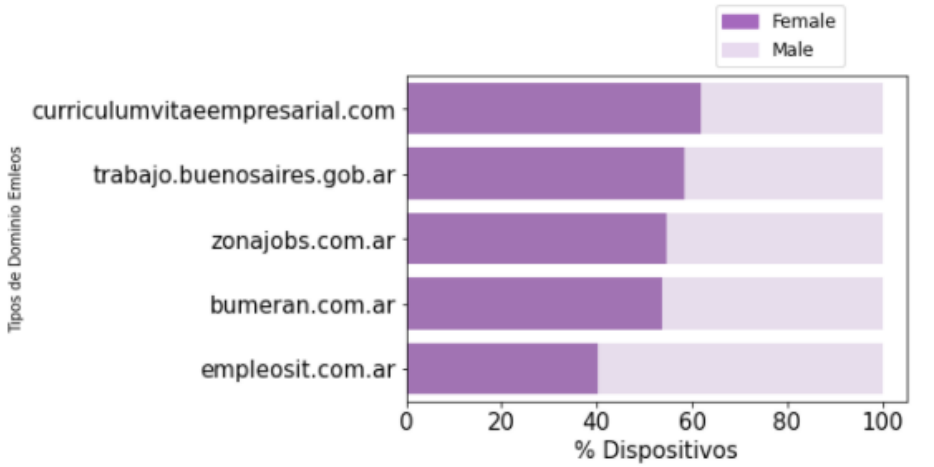


Figura 55: Dominios en Rubro Búsqueda Laboral y Porcentaje de Dispositivos por Género

- *Rubro Noticias*

Noticias es un rubro visitado en forma pareja por ambos géneros. Sin embargo, existe variabilidad al observar en detalle. Tiene sentido que el dominio con mayor proporción de mujeres sea “diariofemenino.con” junto con "caras.perfil.com", que tienen que ver con noticias de moda y farándula respectivamente. Así como también tiene sentido que dominios como "parabrisas.perfil.com" y "noticias.autocosmos" tengan mayoría de visitas masculinas, pues *autocosmos* y *parabrisas* tienen que ver con automóviles además del rubro noticias, y se observó que ese rubro tiene un fuerte componente masculino.

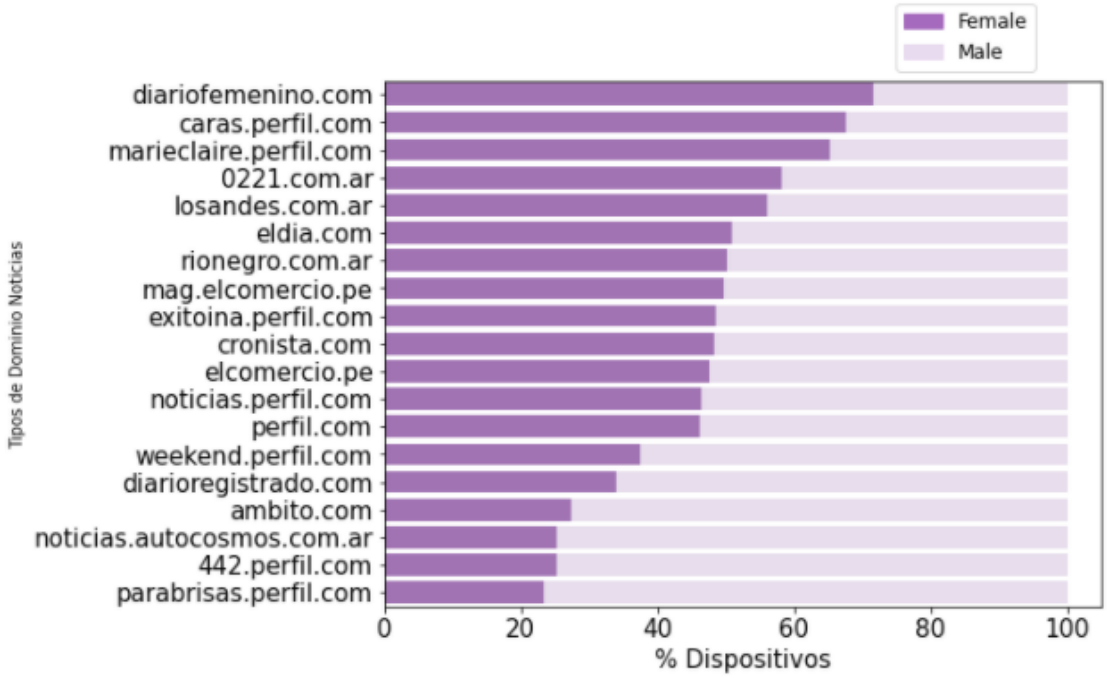


Figura 56: Dominios en Rubro Noticias y Porcentaje de Dispositivos por Género

- *Rubro Deportes*

Deportes es un rubro más visitado por el género masculino que femenino. Únicamente el dominio "solodeportes" posee proporción de visitas pareja en cuanto a género. Los demás dominios son fuertemente masculinos en términos de usuarios que los visitan. Muchos de los cuales hacen referencia al deporte del fútbol, mostrando fuerte interés por el mismo.

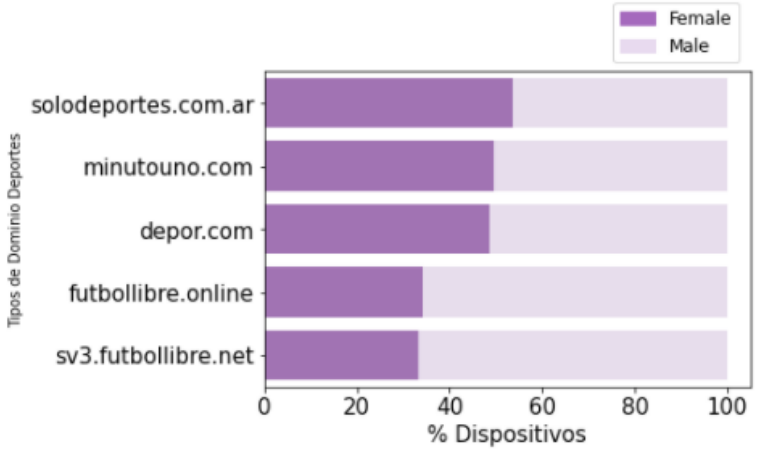


Figura 57: Dominios en Rubro Deportes y Porcentaje de Dispositivos por Género

De esta forma, se procede a armar el dataset con técnicas de One Hot Encoding y asegurando un solo identificador de dispositivo por fila, para luego ser procesado por modelos de machine learning. Se incluye en el dataset final a variables dummy surgidas de los distintos valores categóricos del campo “domain_type” creado. En particular, se excluye a la variable “Línea Móvil” que es tomada como variable de referencia. Se incluye en el Anexo 4 una vista ejemplo del dataset final.

5.4.2. Modelos y Resultados con atributos de User Agent y Dominios

Al igual que en el primer modelo con atributos de User Agent, se evaluarán métricas de accuracy, AUC, recall, precisión y F1-Score que arrojan los modelos al sumar a los features de User Agent, los atributos de dominios. Se evaluará un modelo baseline de Naive Bayes, luego una regresión logística, random forest y finalmente XGBoost.

5.4.2.1. Baseline

Este modelo presenta la siguiente matriz de confusión.

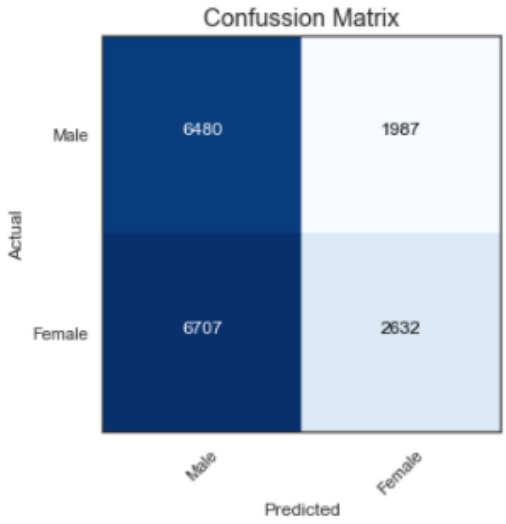


Figura 58: Matriz de Confusión, Modelo Naive Bayes

Al igual que en el primer modelo baseline, naive bayes predice mejor al género masculino que al femenino, arrojando un accuracy de 0.512 y un área bajo la curva ROC de 0.5490. Ambas métricas son mejores que en el caso de utilizar únicamente atributos de user agent.

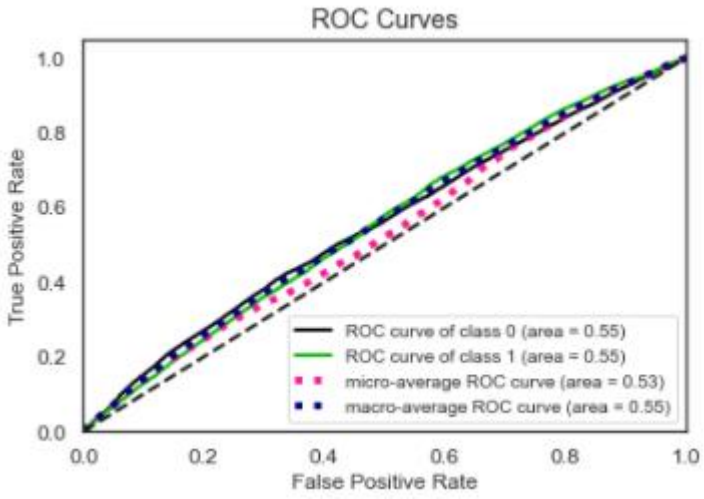


Figura 59: Curva ROC y AUC, Modelo Naive Bayes

5.4.2.2.Regresión Logística

Este modelo logra predecir mejor al género femenino que al masculino, con un 79% de aciertos para la primera clase, y 30% de aciertos para la segunda.

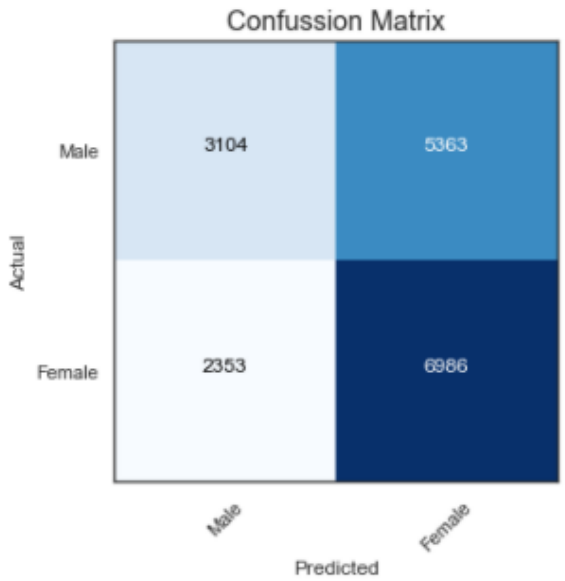


Figura 60: Matriz de Confusión, Modelo Regresión Logística

Al ejecutar las curvas recall-precisión para el género masculino, se observa que pasado el umbral de aproximadamente 0.53, la métrica recall cae acentuadamente, mientras que la precisión aumenta, siguiendo el comportamiento esperado en ambos casos.

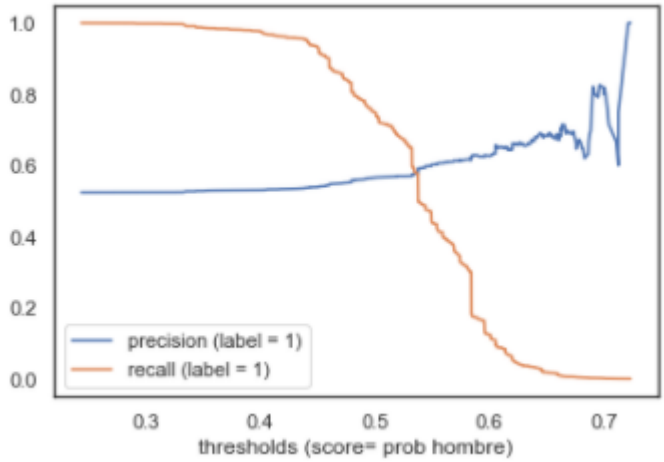


Figura 61: Curva Precisión-Recall en género Masculino , Modelo Regresión Logística

En la curva recall-precisión para el género femenino, ambas curvas se cruzan antes del 0.5 para terminar encontrándose nuevamente en un umbral de 0.75 aproximadamente, donde el punto de corte hace que no se predigan clases positivas, siendo todas clases negativas (bajo recall y baja precisión, ya que no hay verdaderos positivos).

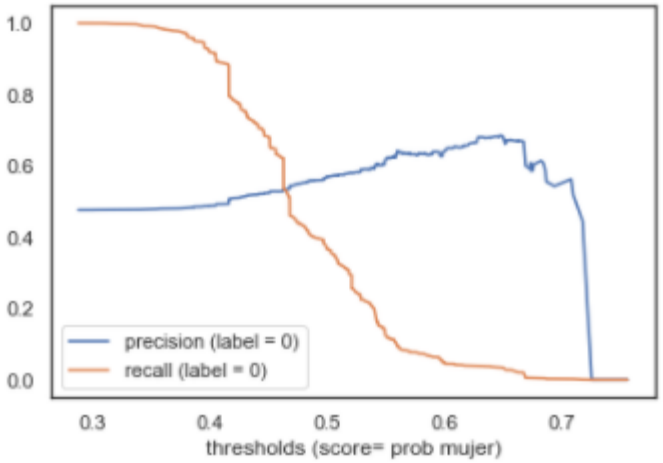


Figura 62: Curva Precisión-Recall en género Femenino, Modelo Regresión Logística

En la curva Precision-Recall para ambas clases, el Average Precision Recall es 0.57 (mayor que teniendo en cuenta sólo User Agent). Este valor, como se mencionó anteriormente, es una manera de calcular el área bajo la curva PR o PR AUC. Cuanto más se acerque su valor a 1, mejor será el modelo.

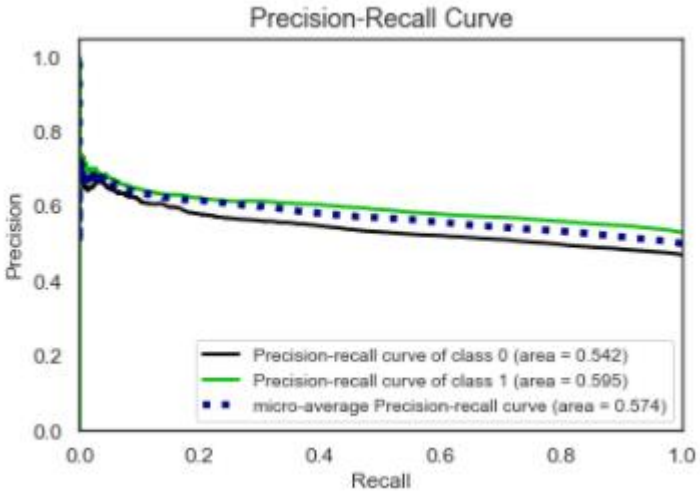


Figura 63: Curva Precisión-Recall, Modelo Regresión Logística

Al observar el AUC, existe una mejoría respecto al primer caso, siendo de 0.58.

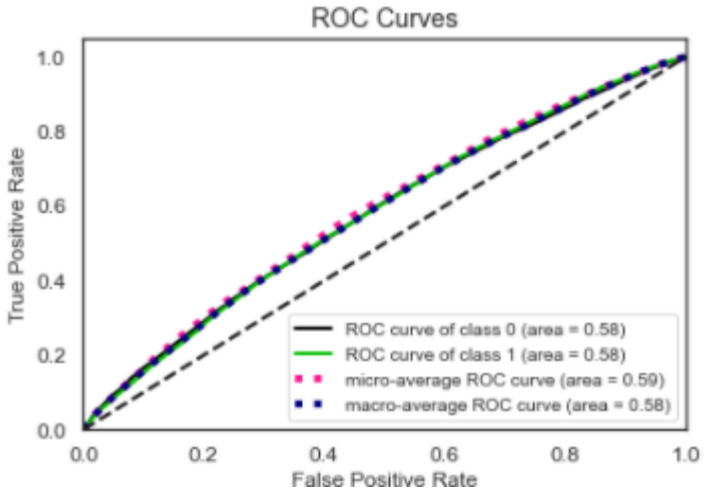


Figura 64: Curva ROC y AUC, Modelo Regresión Logística

Finalmente al observar la curva de aprendizaje, se puede apreciar que a medida que aumentan las observaciones en el conjunto de train, menor es la brecha entre el score que arroja dicho conjunto y el de cross-validation.

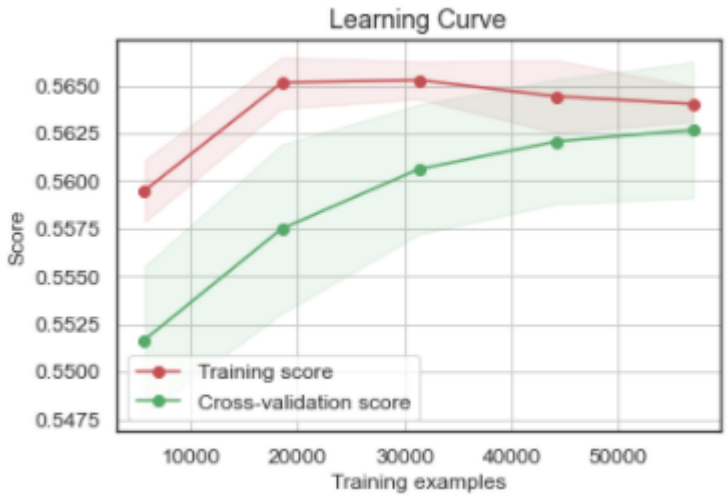


Figura 65: Curva de Aprendizaje, Modelo Regresión Logística

5.4.2.3. Random Forest

Este modelo también predice mejor al género femenino, como se aprecia en la matriz de confusión.

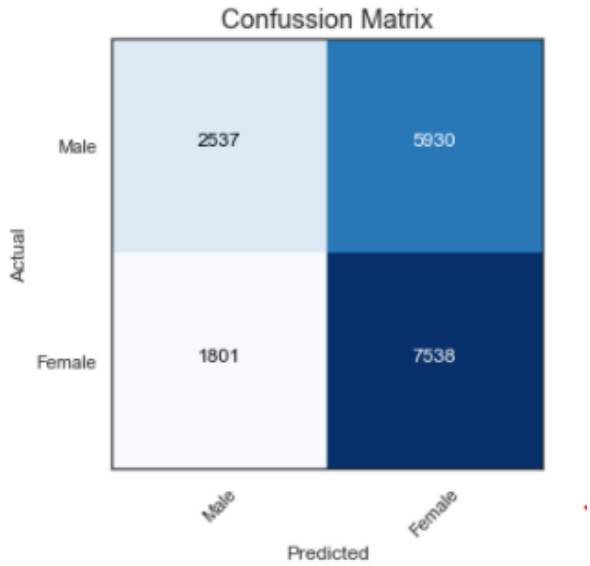


Figura 66: Matriz de Confusión, Modelo Random Forest

A su vez, las curvas recall-precision para el género masculino y femenino poseen las siguientes formas.

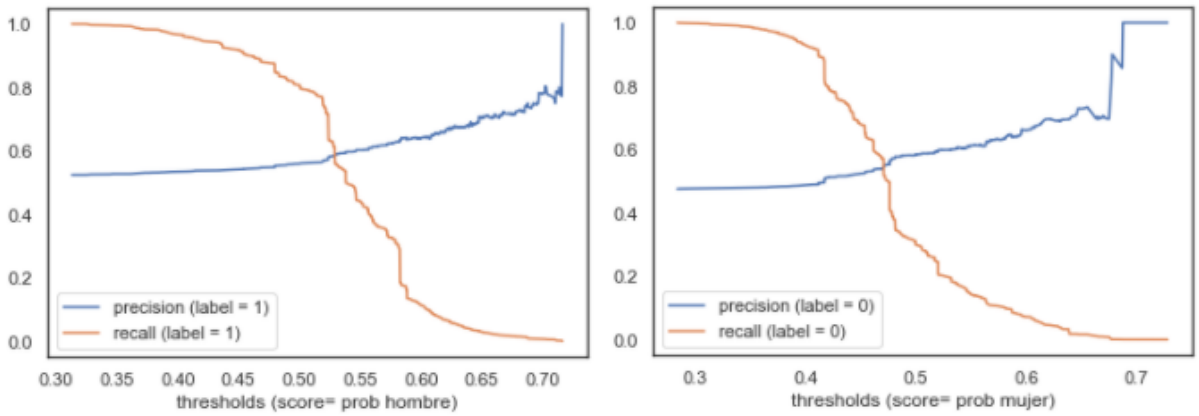


Figura 67: Curvas Precisión-Recall para género masculino y femenino, Modelo Random Forest

Siendo la curva Precision - Recall para ambas clases la siguiente.

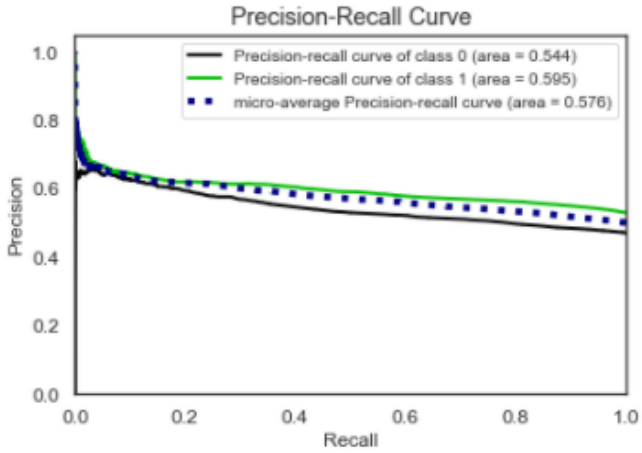


Figura 68: Curva Precisión-Recall, Modelo Random Forest

El AUC para este caso es de más de 0.59.

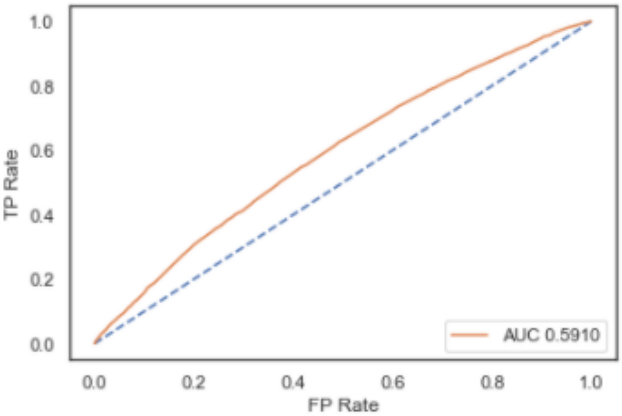


Figura 69: Curva ROC y AUC, Modelo Random Forest

Y la curva Learning Rate también muestra una brecha pequeña entre score en train y cross-validation, a medida que se agranda el conjunto train.

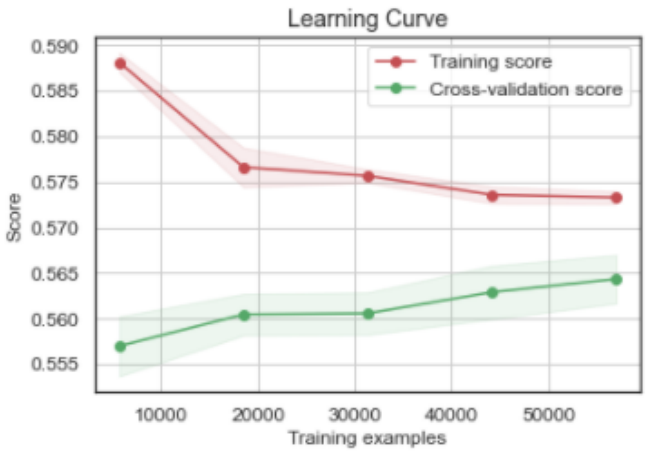


Figura 70: Curva de Aprendizaje, Modelo Random Forest

Al estudiar feature importance en Random Forest, el rubro de dominios “Alimentos” es el predictor que mejor contribuye a la clasificación, seguido por el navegador Firefox, el rubro Buenos Aires, el navegador Chrome, el rubro Inmobiliaria y Automóvil. Es decir que la inclusión de nuevos features al modelo ayudó en gran medida a la performance, pues se ven varios de los nuevos features en los primeros puestos de importancia de atributos, además de mejorar las métricas.

En el siguiente listado se presentan los 20 atributos más importantes.

#	Feature	Importance Value
1	**domain_type_Alimentos**	**0.177974**
2	**navegador_Firefox**	**0.127294**
3	**domain_type_Buenos_Aires**	**0.118103**
4	**navegador_Chrome**	**0.075206**
5	**domain_type_Inmobiliaria**	**0.060765**
6	domain_type_Automovil	0.030785
7	navegador_SocialApp	0.030094
8	domain_type_Noticias	0.027466
9	navegador_Brand_browser	0.024342
10	domain_type_Busqueda_Laboral	0.023579
11	Is_pc	0.022604
12	domain_type_Deportes	0.021353
13	Antiguedad_ancient	0.021066
14	domain_type_Tramites_Bancarios	0.020443
15	OS_Android	0.015225
16	Is_mobile	0.014312
17	OS_Windows	0.014053
18	domain_type_Educacion	0.009931
19	VOS_Windows 10	0.009830
20	Antiguedad_old	0.009265

Tabla 22: Random Forest, Feature Importance Ranking, Top 20

5.4.2.4.XGBoost

Este último modelo, al igual que los dos anteriores, logra predecir mejor al género femenino.

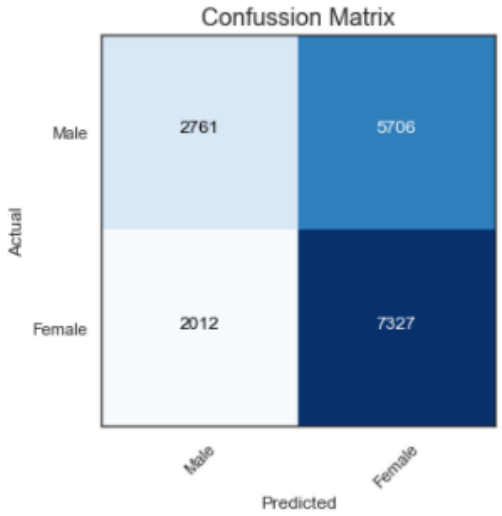


Figura 71: Matriz de Confusión, Modelo XGBoost

Las curvas de recall-precisión para este caso son similares al random forest anterior.

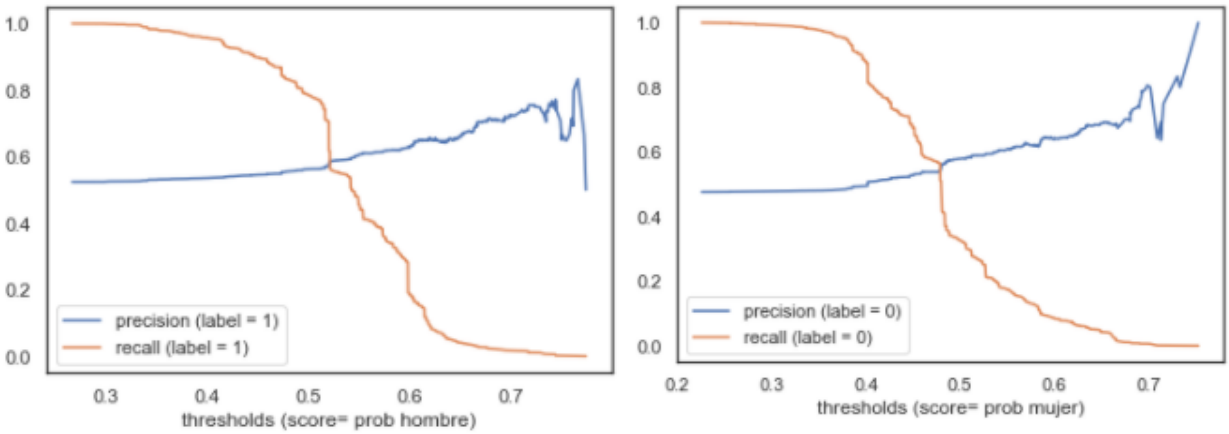


Figura 72: Curvas Precisión-Recall para género masculino y femenino, Modelo XGBoost

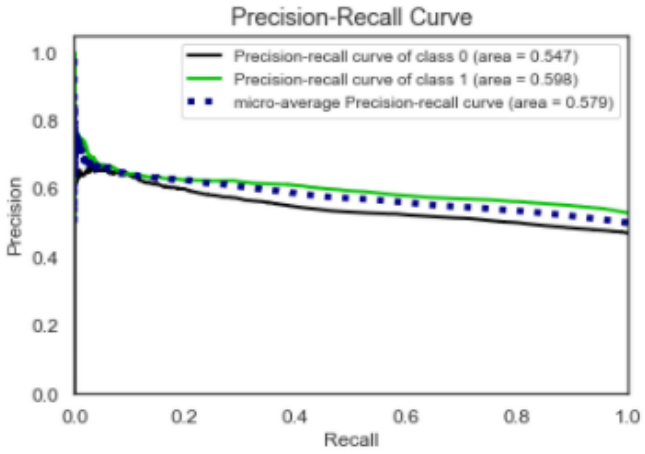


Figura 73: Curva Precisión-Recall, Modelo XGBoost

El AUC dio algo menor al random forest anterior, pero mejor que en el caso de utilizar únicamente atributos de user agent.

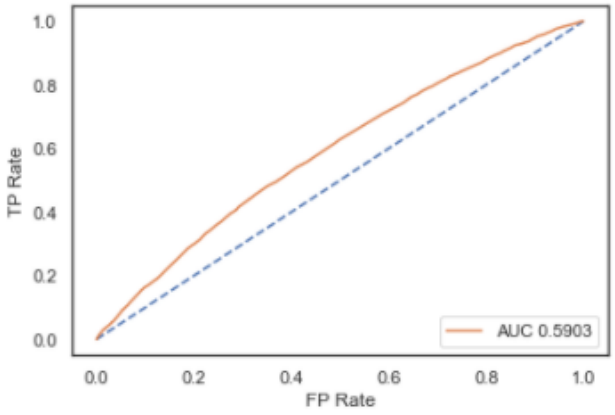


Figura 74: Curva ROC y AUC, Modelo XGBoost

Y la curva de aprendizaje también muestra buenos resultados. Como puede observarse ambas curvas (la del score en training y la del score en el set de validación de cross validation) se acercan cuando se aumenta el número de observaciones. Es decir, se achica la brecha entre ambas mostrando que a mayor cantidad de registros el modelo mejora su performance al evaluarlo con el set de validación, mientras que en el set de training empeora, evidenciando un buen ajuste, ya que el modelo logra “generalizar” de manera óptima.

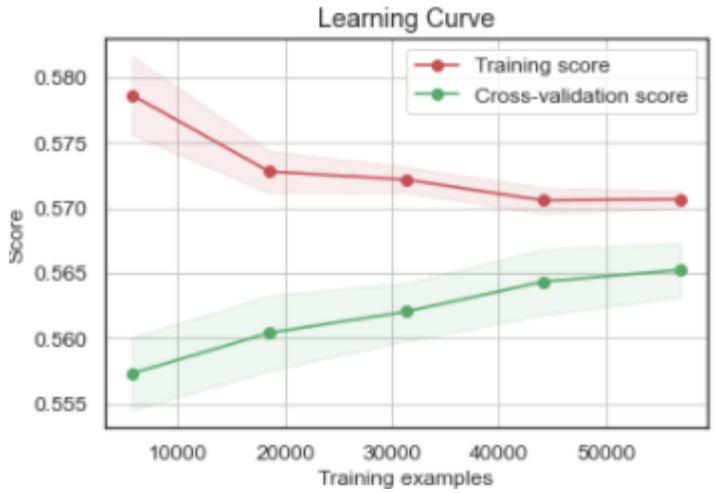


Figura 75: Curva de Aprendizaje, Modelo XGBoost

XGBoost, al igual que random forest permite visualizar la importancia de atributos. Los detalles del cálculo se muestran en la sección de "Interpretabilidad de Modelos". Se muestran a continuación los primeros 20 atributos más importantes para el modelo. A diferencia de random forest, esta técnica de Machine Learning considera al navegador Firefox como el más importante. Sin embargo, el rubro alimentos (puesto 1 de random forest) se encuentra en segundo puesto. El rubro automóvil en este caso aparece como menos importante al compararlo con el ranking de random forest y el sistema operativo Android figura entre los primeros puestos a diferencia del caso del modelo anterior.

importance	feature
0.109823	navegador_Firefox
0.104891	domain_type_Alimentos
0.063444	domain_type_Buenos_Aires
0.057558	OS_Android
0.045226	domain_type_Inmobiliaria
0.044840	OS_Windows
0.038417	Is_mobile
0.031468	domain_type_Busqueda_Laboral
0.028434	navegador_Brand_browser
0.026406	VOS_Windows 10
0.025987	domain_type_Automovil
0.024786	Antiguedad_ancient
0.020042	Is_pc
0.020031	domain_type_Tramites_Bancarios
0.018648	domain_type_Noticias
0.017372	domain_type_Deportes
0.016385	marca_Samsung
0.016362	VOS_Android 8
0.015980	navegador_SocialApp
0.014401	marca_LG

Tabla 23: XGBoost, Feature Importance Ranking, T

Los hiperparámetros elegidos para todos los modelos mediante técnicas de grid search y random search se encuentran en la sección “Anexo”.

5.5. Atributos de Sitios web (Urls)

5.5.1. Análisis exploratorio y Feature engineering

En esta sección se analizan los atributos de urls o sitios web específicos por los que navega cada dispositivo.

Existen 1.576.360 filas con datos de sitios web. Los cuales se reparten en 98.491 dispositivos únicos, que poseen datos de sitios web visitados. De estas 1.576.360 filas, 30.882 webpages son únicas. Las urls se repetirán por cada dispositivo que la visite. Como parte del tratamiento de los datos, se generan nuevos campos separando el dato “url” por dominio, subdominio y sufijo para facilitar el análisis exploratorio y feature engineering. Al obtener el campo “dominio” a partir de las 30.881 urls, se obtienen 678 dominios únicos. Además, se parsea la url completa para generar un campo “path_corpus” que será input de gráficos wordclouds para visualizar palabras clave de cada sitio visitado.

En cuanto a la distribución de dispositivos en sitios web, más de 30.000 sitios web tienen visitas de entre 0 y 420 dispositivos aproximadamente. Es decir, pocos usuarios visitan muchos sitios distintos. Por lo que pocos sitios web o urls concentran a la mayor parte de la población de estudio. En otras palabras, muchos dispositivos visitan pocos sitios web específicos. En particular, 21% de los dispositivos visita zonajobs.com.ar, casi el 20.5% de dispositivos visita bumeran.com.ar y un 15 % visita veadigital.com.ar.

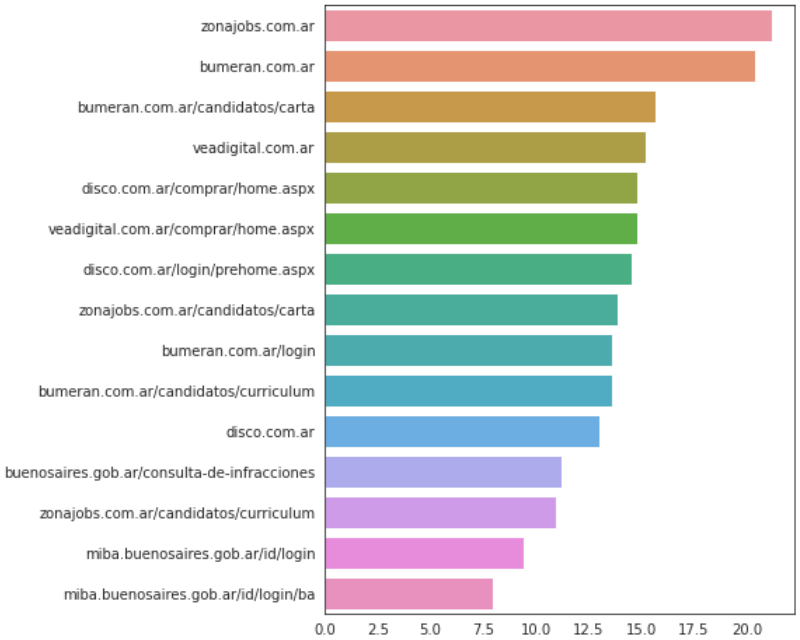


Figura 76: Porcentaje de Dispositivos únicos que visita a cada Url (Top 15)

Respecto a la distribución de dispositivos en dominios desprendidos de cada url, al igual que en el caso de los sitios web, pocos dominios concentran a la mayor parte de la población.

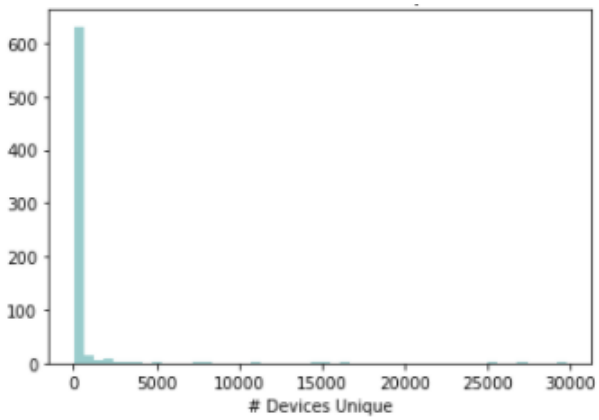


Figura 77: Distribución de Dispositivos, por dominio de Urls

Observando los datos en cuanto a la variable target de género, se observa en detalle la distribución anterior.

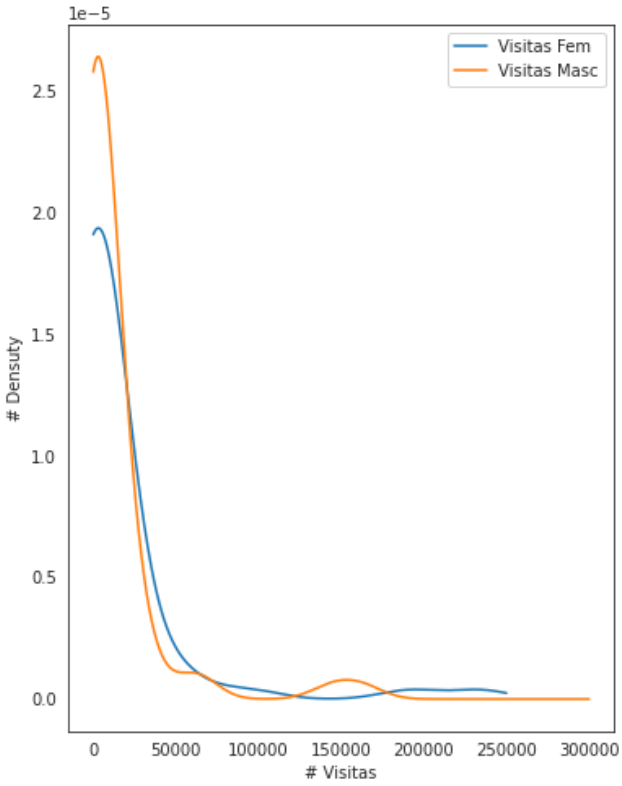


Figura 78: Distribución de Visitas por dominio de Urls, según Género

Por otro lado, se realizaron nubes de palabra con el campo “path_corpus”, que dan una idea de los intereses más comunes en mujeres y hombres que ofrece el dataset.
En el género femenino:

Figura 79: WordCloud en path de Urls, Género Femenino

En el género masculino:

Figura 80: WordCloud en path de Urls, Género Masculino

Como bien se observa, los intereses por búsqueda laboral son los más fuertes en ambos casos. Para conocer mejor los intereses en la población de estudio y luego aplicar técnicas de clasificación, se generará una matriz que cruce a cada uno de los identificadores de dispositivos (filas) con los dominios que visitan (columnas), teniendo como valor la cantidad de veces que el usuario visitó cada dominio según todos los sitios web por el que haya navegado que contengan al dominio en concreto.
Para esto, de los 678 dominios desprendidos de las urls del dataset, se mantienen aquellos con cantidad de visitas relevante, clasificando en “Otros” a los dominios con muy pocas visitas. Más concretamente, observando proporciones de género en los dominios, la nueva variable que agrupa por dominios a los sitios web visitados presenta variabilidad con el género. Los sitios de farándula como "pronto" y "ratingcero", y los sitios de marcas como "swarovski" y "starbucks rewards", así como también sitios de jardinería y alimentos son mayormente de preferencia femenina.
En cambio, sitios sobre deporte como "mundoazulgrana", "rojadirectatv" y "futbollibre" entre otros, más sitios sobre tecnología y autos son visitados más por el género masculino que femenino. Esta inclinación también fue registrada en el estudio de la sección anterior sobre dominios.

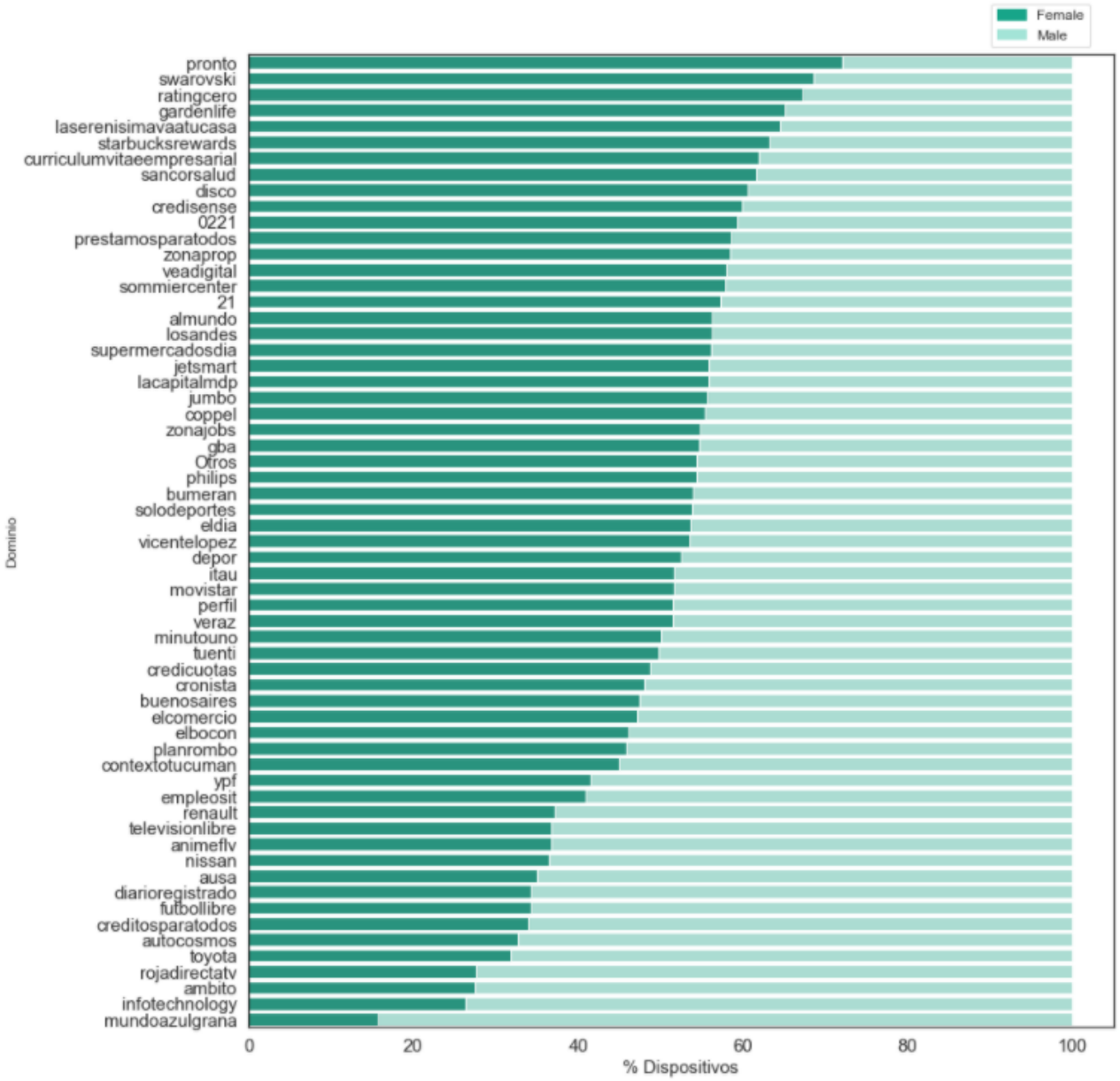


Figura 81: Dominios de Url y Porcentaje de Dispositivos por Género

5.5.2. Modelos y Resultados con atributos de User Agent, Dominios y Urls

En esta sección se presentan los resultados de técnicas de machine learning aplicadas para resolver el problema de clasificación de género con atributos de user agent, dominios y también de urls visitadas por los usuarios.

Mediante GridSearch para el modelo de Regresión logística, se indicó en el hiperparámetro “penalty” las opciones de lasso o ridge como método de regularización para que el modelo elimine (en el caso de lasso) o considere aquellas variables que hacen a la mejor performance del modelo.

Luego se prueba con Random Forest seleccionando hiperparámetros mediante Random Search y

finalmente un XGBoost, también optimizando hiperparámetros con Random Search. En los tres casos, los modelos se ejecutan bajo el esquema de K-Fold cross validation.

5.5.2.1. Baseline

La siguiente matriz de confusión muestra que, a diferencia del modelo baseline en los casos de modelos con features de user agent y user agent más dominios, esta técnica, al agregar los nuevos atributos, predijo mejor al género femenino.

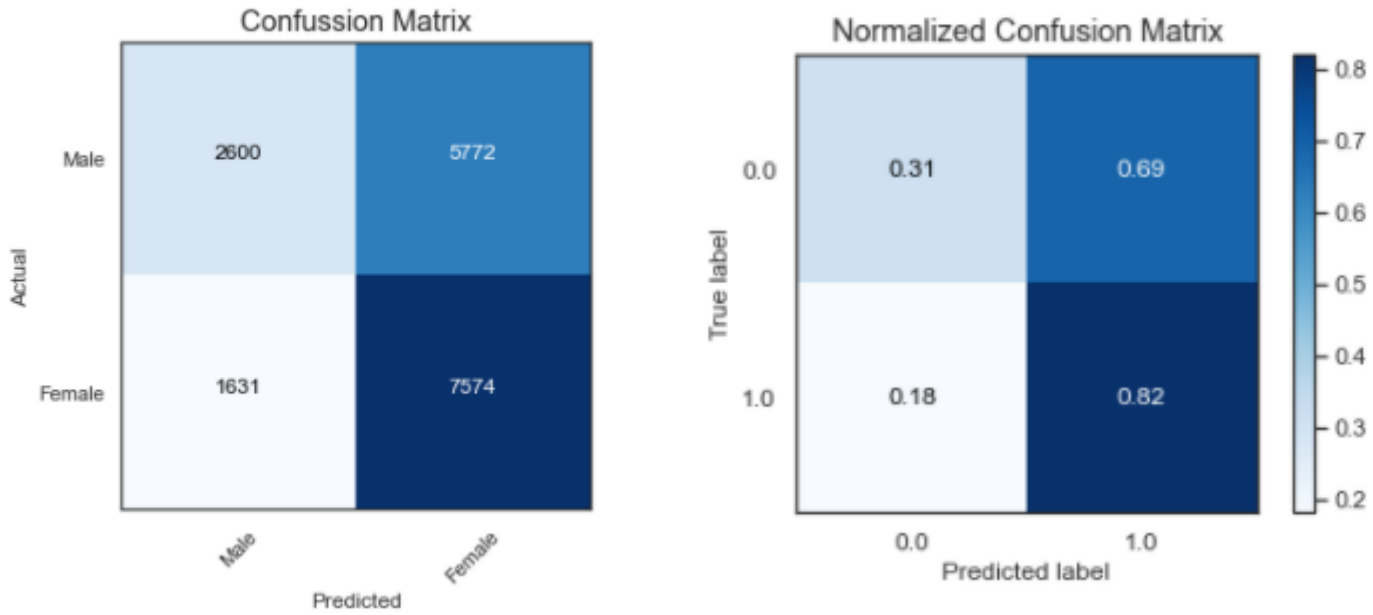


Figura 82: Matriz de Confusión, Modelo Naive Bayes

Siendo el accuracy de 0.579 y el área bajo la curva ROC de 60%.

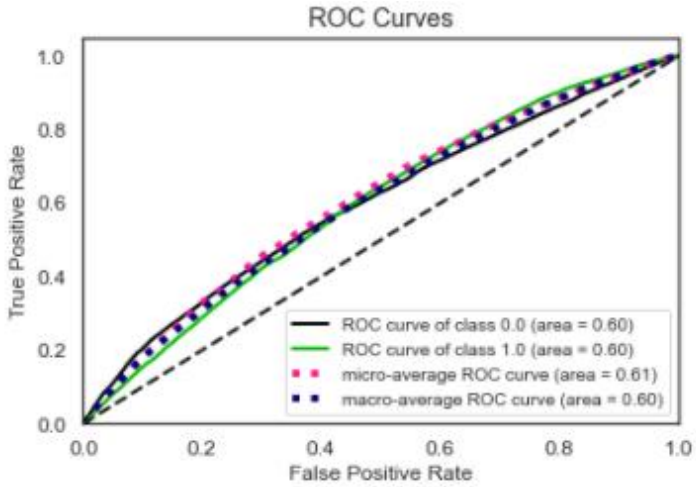


Figura 83: Curva ROC y AUC, Modelo Naive Bayes

5.5.2.2.Regresión Logística

Luego de optimizar hiperparámetros mediante GridSearch, esta técnica predijo correctamente al 83% de las mujeres y al 34% de los hombres.

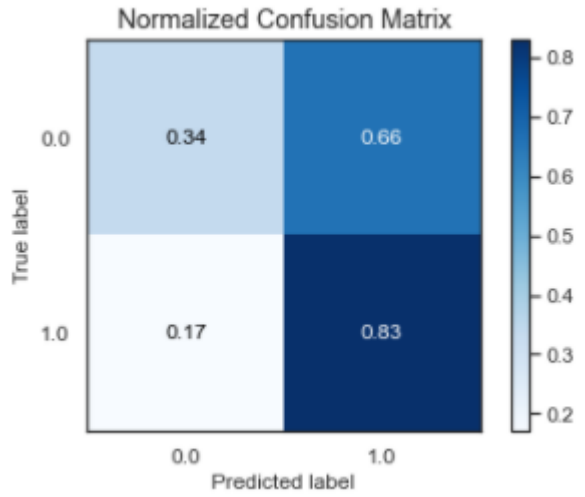


Figura 84: Matriz de Confusión, Modelo Regresión Logística

Observando las curvas Precision-Recall para ambas clases, se observa que para el género masculino, cuando el threshold supera 0.55 aproximadamente, el recall cae abruptamente y la precisión aumenta en menor magnitud.

Para el caso femenino, ambas curvas se cruzan antes del 0.5 y nuevamente en el 1 a partir de donde, como es esperable, no se predicen más clases positivas y por ende tanto recall como precisión son bajos.

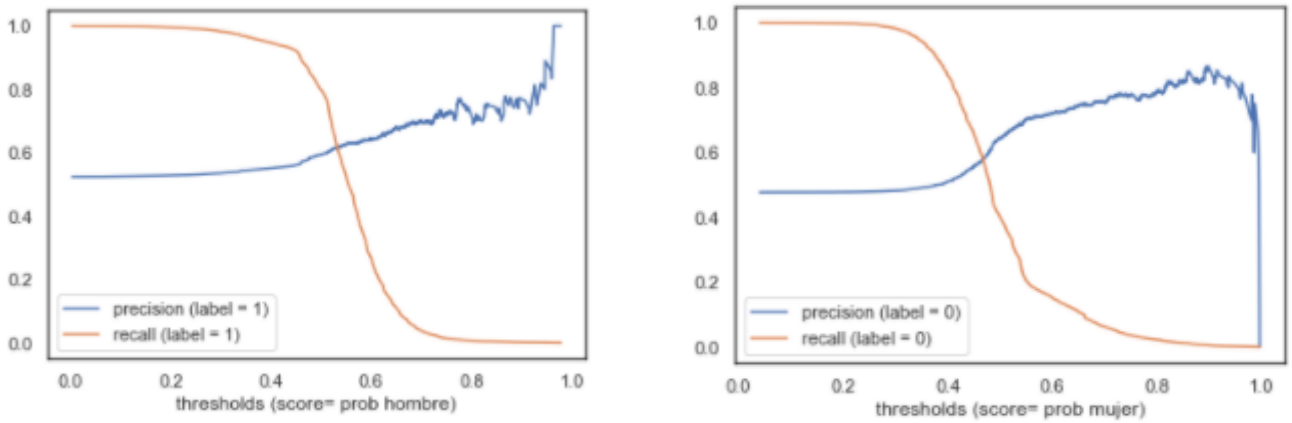


Figura 85: Curvas Precisión-Recall para género masculino y femenino, Modelo Regresión Logística

El average Precision Recall es 0.623, mostrando mejoría respecto a los dos modelos donde se consideraron features de user agent y luego user agent más atributos de dominios.
Y el AUC también presenta mejoría incluyendo este nuevo set de features.

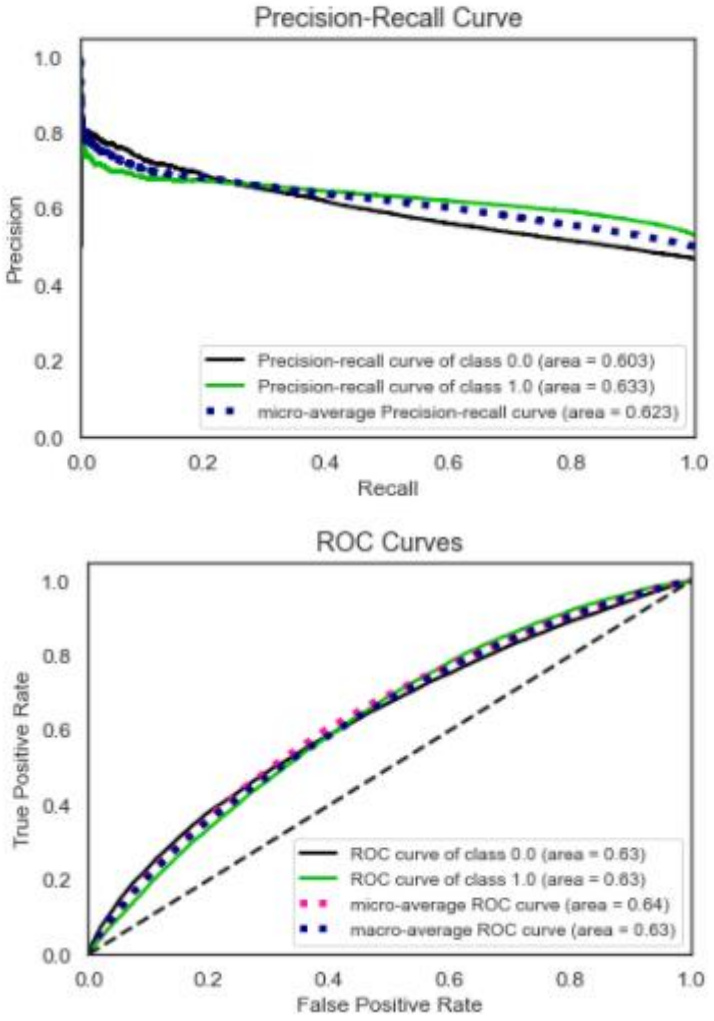


Figura 86: Curvas Precisión-Recall y ROC, Modelo Regresión Logística

5.5.2.3.Random Forest

Esta técnica de modelos de ensamble presenta mejor performance al agregar nuevos atributos de sitios web visitados. Predijo correctamente al 78% de las mujeres y al 44% de los hombres.

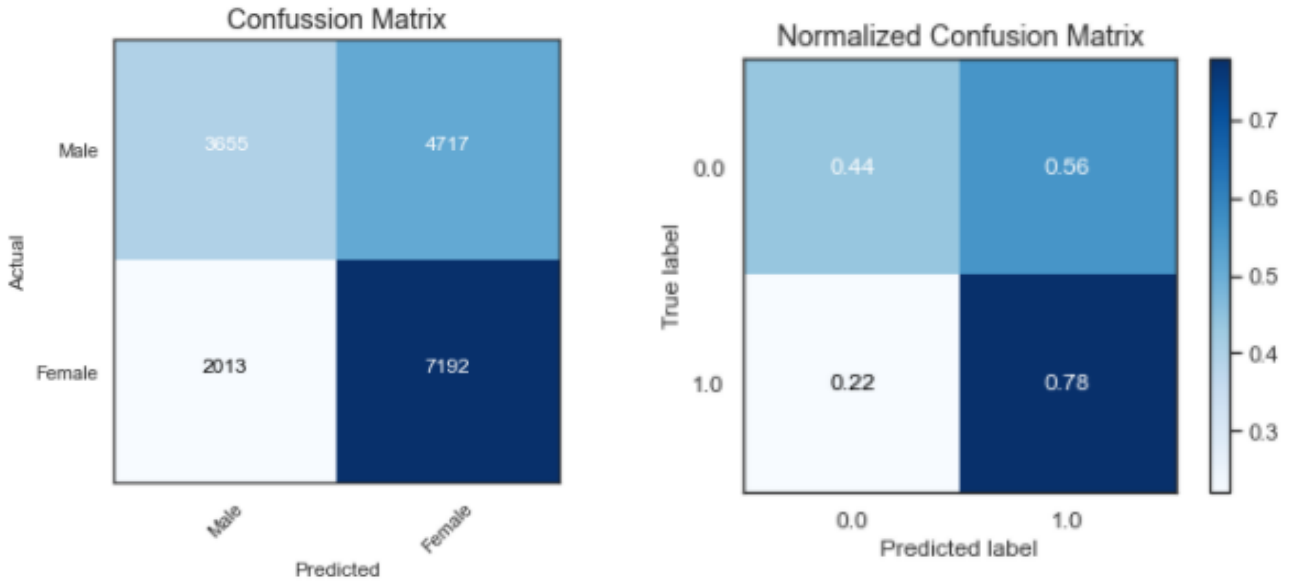


Figura 87: Matriz de Confusión, Modelo Random Forest

En cuanto a las curvas Recall-Precision para el género masculino y femenino, se obtuvo lo siguiente.

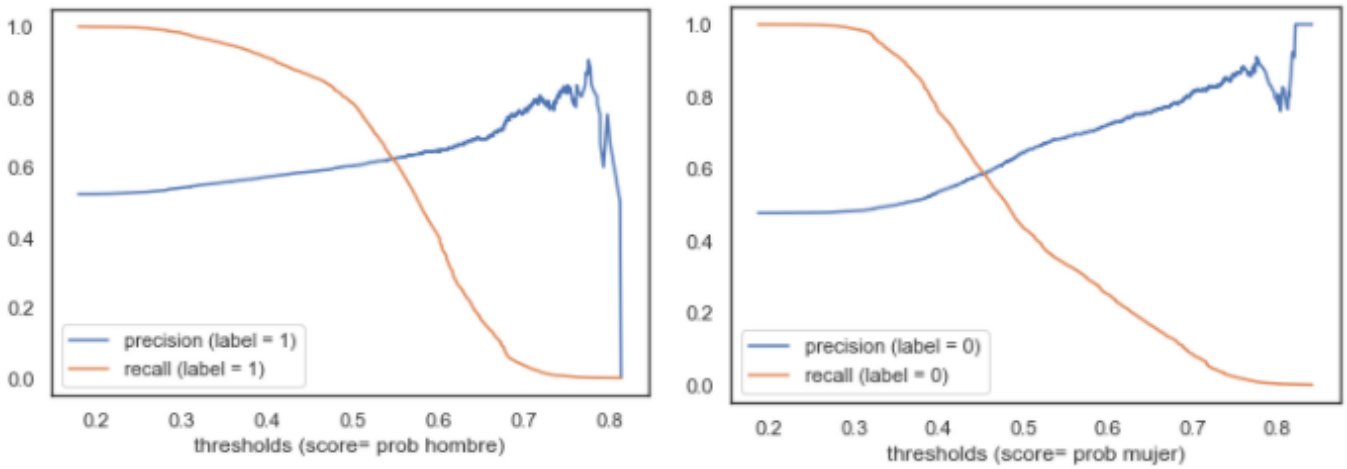


Figura 88: Curvas Precisión-Recall en género masculino y femenino, Modelo Random Forest

La curva Recall-Precision y el AUC presentan una clara mejoría al incluir urls.

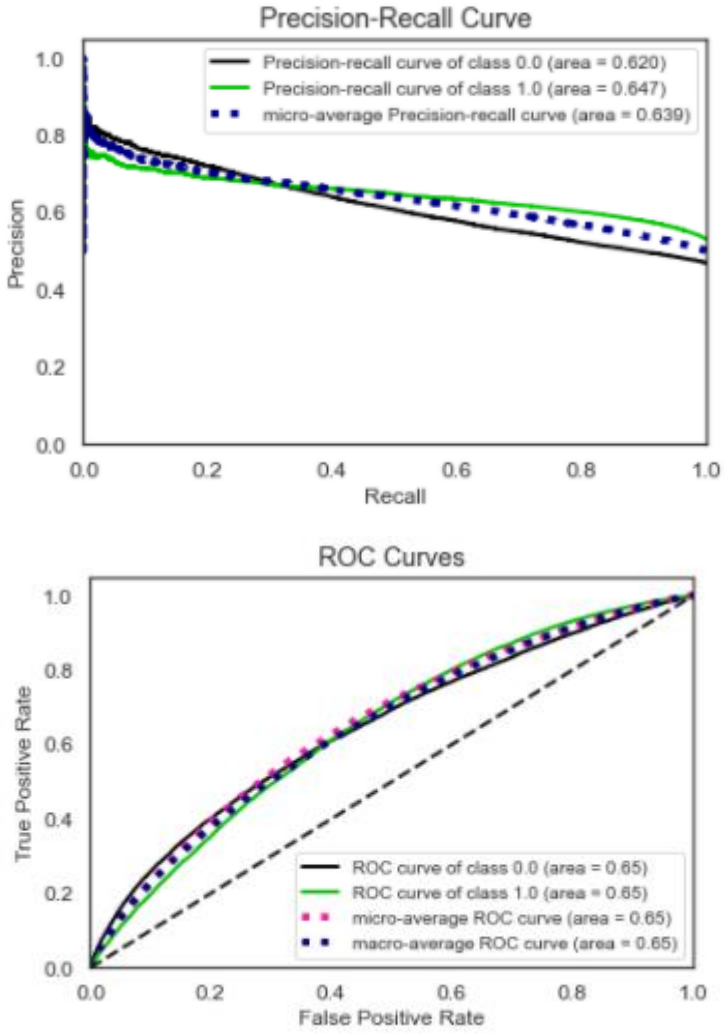


Figura 89: Curvas Precisión-Recall y ROC, Modelo Random Forest

En cuanto a la importancia de atributos calculada, el feature "ambito" es considerado como el más importante al momento de predecir, seguido por "buenos aires", "zonajobs" y "disco". Es decir que estos dominios que se desprendieron de sitios web navegados por los dispositivos bajo estudio, al basarnos en la cantidad de veces que cada uno navegó por sitios con esos dominios en su path, los relacionados con noticias, empleos y supermercados resultaron ser de los que más permiten clasificar usuarios por género. A continuación, se muestra el top 20 del mencionado feature importance.

#	Feature	Importance Value
1	**ambito**	**0.078900**
2	**buenosaires**	**0.068813**
3	**zonajobs**	**0.059919**
4	**bumeran**	**0.059754**

5	**zonaprop**	**0.054331**
6	disco	0.052498
7	futbollibre	0.051634
8	navegador_Firefox	0.038022
9	veadigital	0.035522
10	creditosparatodos	0.034359
11	cronista	0.033616
12	Otros	0.030263
13	infotechnology	0.023244
14	domain_type_Alimentos	0.023082
15	movistar	0.021884
16	jumbo	0.020116
17	ausa	0.019526
18	OS_Android	0.019525
19	pronto	0.019477
20	toyota	0.018718

Tabla 24: Ranking Feature Importance, Random Forest

5.5.2.4.XGBoost

Finalmente, esta técnica también presenta mejor performance con los nuevos atributos. Logró predecir acertadamente al 77% de los usuarios con género femenino y al 44% de los que poseen género masculino.

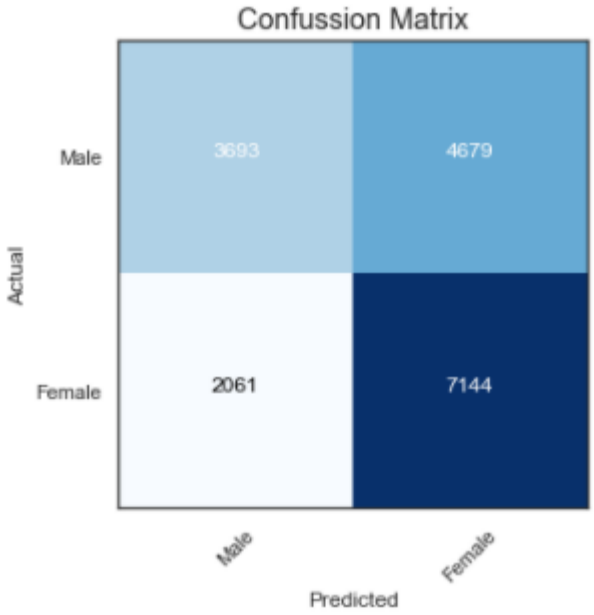


Figura 90: Matriz de Confusión, Modelo XGBoost

Las curvas precision - recall atribuidas a este modelo son las siguientes.

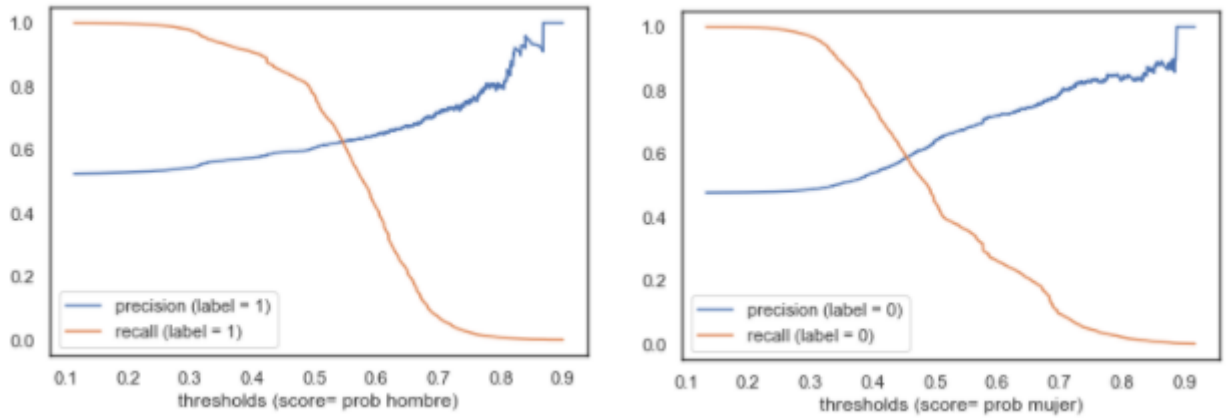


Figura 91: Curvas Precisión-Recall género masculino y femenino, Modelo XGBoost

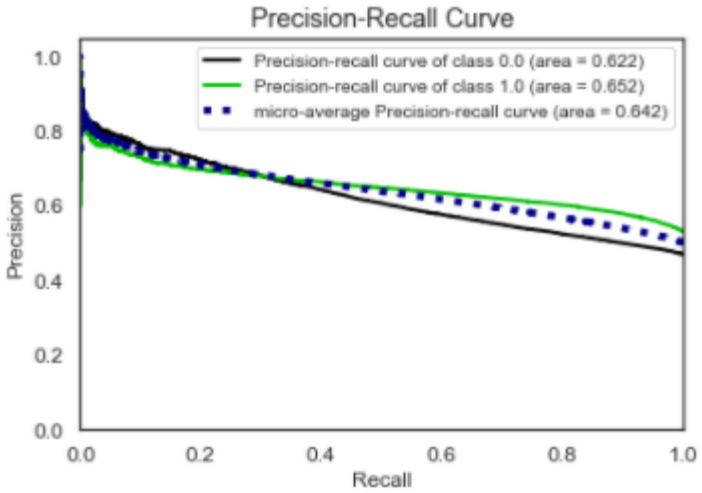


Figura 92: Curva Precisión-Recall, Modelo XGBoost

Como en el caso de random forest, XGBoost también presenta un AUC cercano a 0.65.

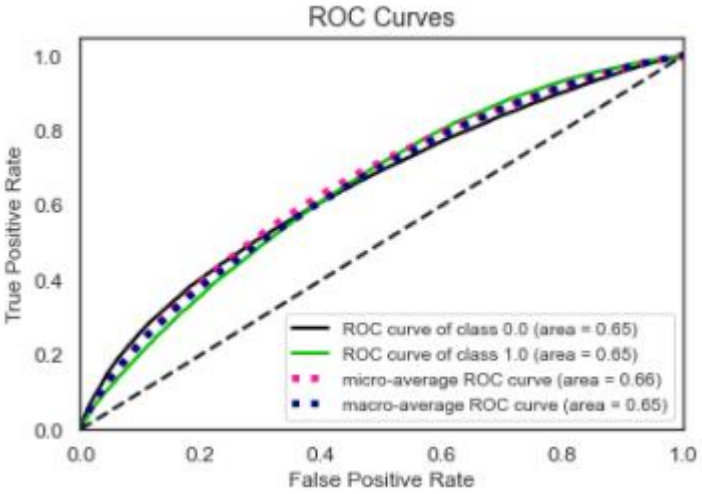


Figura 93: Curva ROC y AUC, Modelo XGBoost

La curva de aprendizaje, por su parte, se comporta de la siguiente manera. Se achica la brecha entre score en training set y testing a medida que se agranda la cantidad de observaciones tomadas para entrenar al modelo.

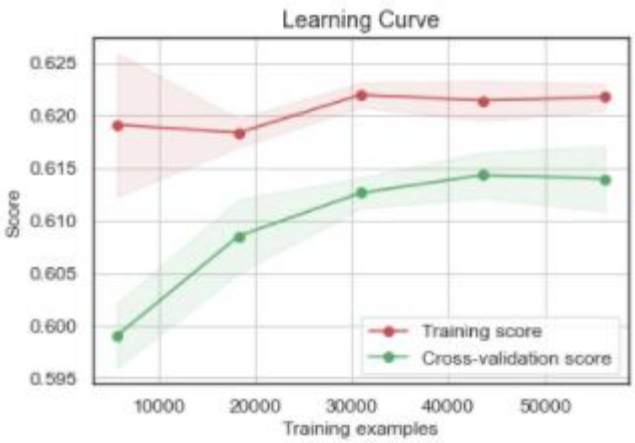


Figura 94: Curva de Aprendizaje, Modelo XGBoost

En cuanto a la importancia de atributos calculada por este modelo, se muestran a continuación los 20 features más importantes. El más importante coincide con el seleccionado por random forest, no así el segundo puesto fue para "navegador_Firefox" como predictor importante, seguido por "creditosparatodos" y el dominio "futbollibre".

importance	feature
0.177673	ambito
0.081775	navegador_Firefox
0.074199	creditosparatodos
0.046456	futbollibre
0.041374	ausa
0.036336	pronto
0.034797	infotechnology
0.032162	toyota
0.027073	autocosmos
0.022249	rojadirectatv
0.021147	mundoazulgrana
0.019850	OS_Android
0.015582	zonaprop
0.015391	disco
0.014565	domain_type_Alimentos
0.014546	animeflv
0.014449	buenosaires
0.014416	starbucksrewards
0.013258	prestamosparatodos
0.012765	veadigital

Tabla 25: XGBoost, Feature Importance Ranking, Top 20

Los hiperparámetros elegidos para todos los modelos mediante técnicas de grid search y random search se encuentran en la sección “Anexo”.

6. Interpretabilidad de Modelos

En la sección anterior se obtuvo un ranking de “feature importance” tanto en el caso del algoritmo Random Forest como el modelo final XGBoost. En esta sección se explicará cómo dichos modelos de aprendizaje automático calcularon la importancia de variables y también se mostrarán resultados de otras técnicas de feature importance. Además se presentan resultados de otras estrategias de interpretabilidad de modelos de machine learning sobre el modelo XGBoost, ya que fue el que obtuvo la mejor performance.

Las estrategias de interpretabilidad que se presentarán en esta sección son:

En lo que respecta a ***Importancia Global,*** siendo aquella que se enfoca en la performance general del modelo*:*

- *Mean Decrease Impurity* (utilizando por Random Forest)
- Feature importance mediante *gain* (utilizado por XGBoost)
- Feature importance mediante *Permutación*
- *Partial Dependence Plot*

En cuanto a ***Importancia Local,*** siendo aquella que se centra en el estudio de instancias específicas*:*

- Aproximación *LIME*
- Aproximación *SHAP,* del cual se obtienen los siguientes gráficos: Individual Force Plot, Summary Plot y Feature Dependence Plot.

Con respecto al feature importance en Random Forest, se utiliza la técnica *mean decrease impurity* para obtener la importancia de atributos en el modelo. Como se detalla en la sección de marco teórico, random forest es un conjunto de árboles de decisión. Cada uno es un conjunto de hojas y nodos internos. En el nodo interno, el atributo seleccionado se utiliza para tomar una decisión sobre cómo dividir el conjunto de datos en 2, con respuestas similares dentro. En cada división de los árboles, se registra el descenso de impureza conseguido en la medida utilizada como criterio de división, siendo la impureza de Gini la métrica para este caso de clasificación. Para cada predictor se calcula este descenso de impureza conseguido en el conjunto de árboles que forman el ensamble. Luego, el feature que más reduce la impureza en promedio, o dicho de otro modo, el que genera conjuntos más homogéneos, es el seleccionado para el nodo interno, por lo tanto, mayor es su contribución en el modelo.

En el caso de XGBoost, el cálculo de feature importance viene dado por la métrica de la de ganancia (“gain”) para medir el descenso de impureza que generan los predictores al momento de realizar la división de datos en 2 conjuntos de los datos. La fórmula de gain es la siguiente:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

El “Gain” implica la contribución relativa de un determinado feature hacia el modelo, tomando la contribución de cada feature para cada árbol del modelo. Un valor más alto de esta métrica cuando se la compara con el de otro atributo, implica que el atributo en cuestión es más importante al momento de generar las predicciones. En específico, esta métrica de ganancia es la mejora en el accuracy aportada por un feature a las ramas en las que se encuentra. La idea es que antes de agregar un nuevo split o división en un feature a la rama en la que está, hay elementos clasificados incorrectamente, y después de agregar el split en este feature, habrá dos nuevas ramas, cada una más precisa.

Además de considerar la importancia de atributos calculada por los algoritmos aplicados, se realizó un estudio de importancia mediante distintas estrategias de interpretabilidad.

En segundo lugar, dentro de los ejercicios de interpretabilidad global, se calculó feature importance mediante ***Permutación***. Dicha técnica consiste en permutar valores de un feature aleatoriamente, dejando la variable target y las demás variables sin permutar, observando la caída en el score del modelo. Este procedimiento “rompe” la relación entre el atributo y la variable target, por lo que dicha caída es indicativa de cuánto el modelo depende de ese atributo sobre el que se permutan valores. Cuanto más alta es la diferencia entre el score del modelo original y el score del modelo cuando se permutan valores de un atributo, mayor es la caída de performance por introducir ruido y por ende más importante será ese atributo. El mismo procedimiento se realiza con todas las columnas.

Weight	Feature
0.0147 ± 0.0022	buenosaires
0.0143 ± 0.0025	disco
0.0134 ± 0.0023	ambito
0.0118 ± 0.0010	bumeran
0.0111 ± 0.0024	futbollibre
0.0100 ± 0.0027	zonaprop
0.0091 ± 0.0007	veadigital
0.0077 ± 0.0029	zonajobs
0.0051 ± 0.0014	creditosparatodos
0.0041 ± 0.0021	toyota
0.0036 ± 0.0018	marca_Samsung
0.0031 ± 0.0020	pronto
0.0028 ± 0.0003	ausa
0.0026 ± 0.0010	infotechnology
0.0018 ± 0.0006	minutouno
0.0017 ± 0.0006	ypf
0.0016 ± 0.0007	autocosmos
0.0016 ± 0.0006	cronista
0.0016 ± 0.0006	rojadirectatv
0.0013 ± 0.0007	Antiguedad_old
… 98 more …	

Figura 95: Feature Importance por Permutación

Como se puede observar en el top 15 de variables más importantes, los dominios buenosaires, disco, ambito y bumeran aparecen como los más relevantes. Tanto este método como los aplicados por XGboost y Random Forest detectan alta importancia en ambito, disco y futbollibre.

Se realizaron además algunos gráficos conocidos como ***Partial Dependence Plot (PDP)***, los cuales, dado un feature, van mostrando cuál es el cambio en la columna a predecir, en función de cambios en ese feature. Es el nivel que toma la predicción en cada caso. Este método asume independencia entre las distintas variables. Es decir, al analizar el efecto de una variable, asume que el resto se mantiene constante. Es por esto que se vuelve necesario ortogonalizar las variables para que no estén correlacionadas entre sí, y entonces poder identificar efectos.
Para el caso de variables con valores no binarios como bumeran, zonajobs y disco, se observan los siguientes plots.

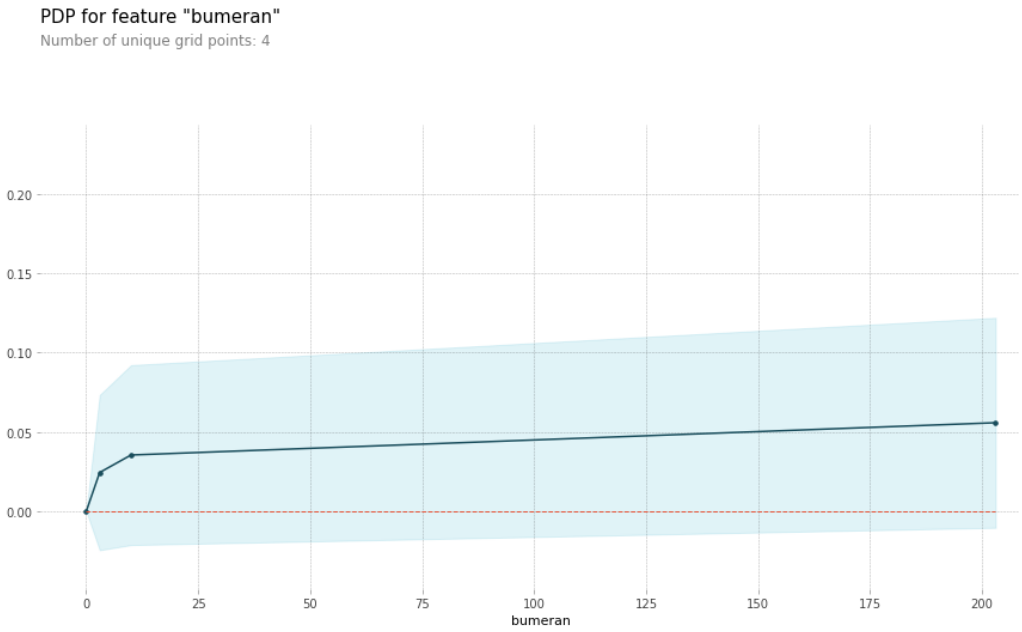


Figura 96: Partial Dependence Plot por atributo "bumeran"

La variable bumeran tiene influencia creciente en la variable target de género hasta el valor 10. Luego, su efecto comienza a ser lineal.

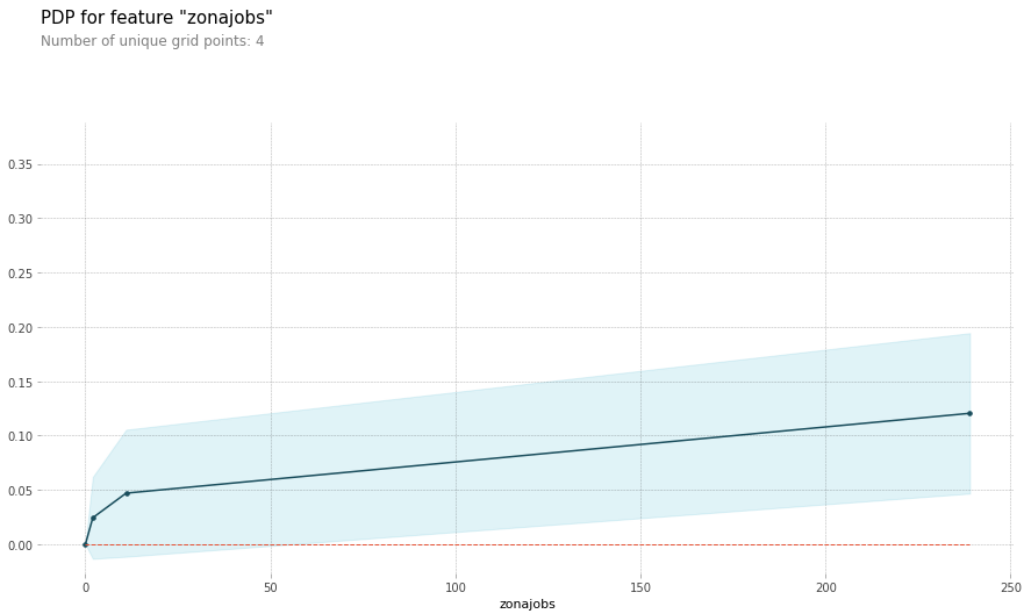


Figura 97: Partial Dependence Plot por atributo "zonajobs"

La variable zonajobs tiene influencia creciente en la variable target hasta el valor 15. Luego, su efecto continúa creciente pero comienza a ser lineal.

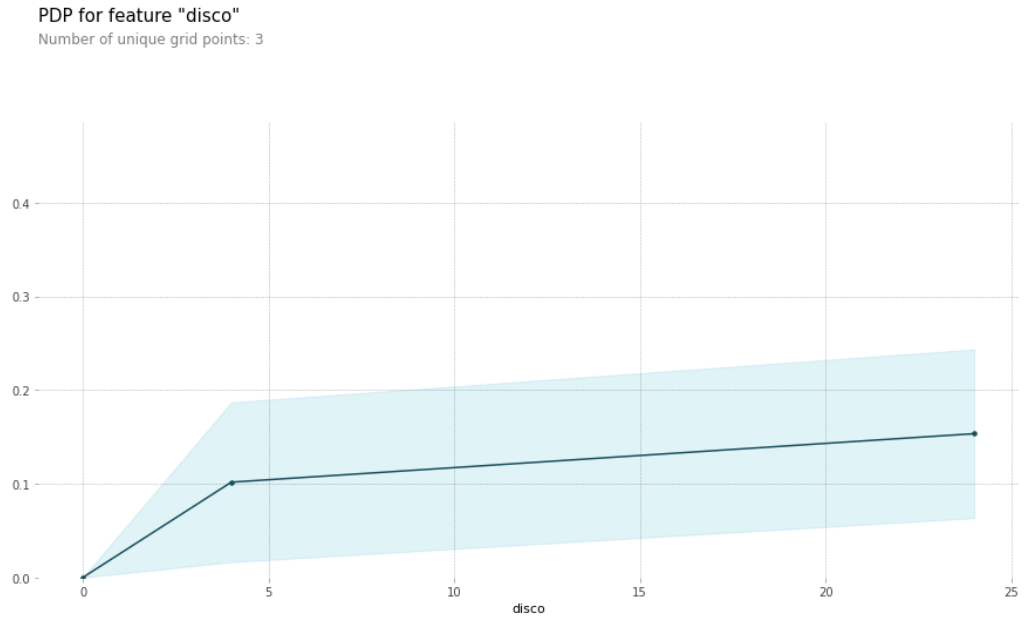


Figura 98: Partial Dependence Plot por atributo "disco"

La variable disco tiene influencia creciente en la variable target hasta el valor 4. Luego, su efecto continúa lineal.

Se debe tener en cuenta que esta técnica PDP tiene algunas desventajas. En primer lugar, sólo se puede representar hasta dos variables al mismo tiempo. En segundo lugar, la mayoría de las visualizaciones PDP no muestran la distribución de las variables. Esto puede ser un problema, ya que puede llevar a que se

sobre-interpreten zonas con poca información. Tercero, como fue mencionado, asume independencia entre features, lo cual hace necesario aplicar técnicas de ortogonalización para poder identificar efectos relevantes. Finalmente, este método puede esconder efectos heterogéneos, ya que solo se muestran efectos marginales promedio, y no uno por uno.

Ahora bien, se analiza la importancia local de variables (Local Surrogate Models). Tomando los resultados del modelo final XGBoost, se observa a continuación el efecto de unas 40 variables según la **aproximación local de LIME** en 25 instancias seleccionadas. Estos registros seleccionados corresponden a los que el módulo "*submodular_pick*" de Lime elige (en base a un problema de optimización) para explicar el modelo en base a ejemplos.

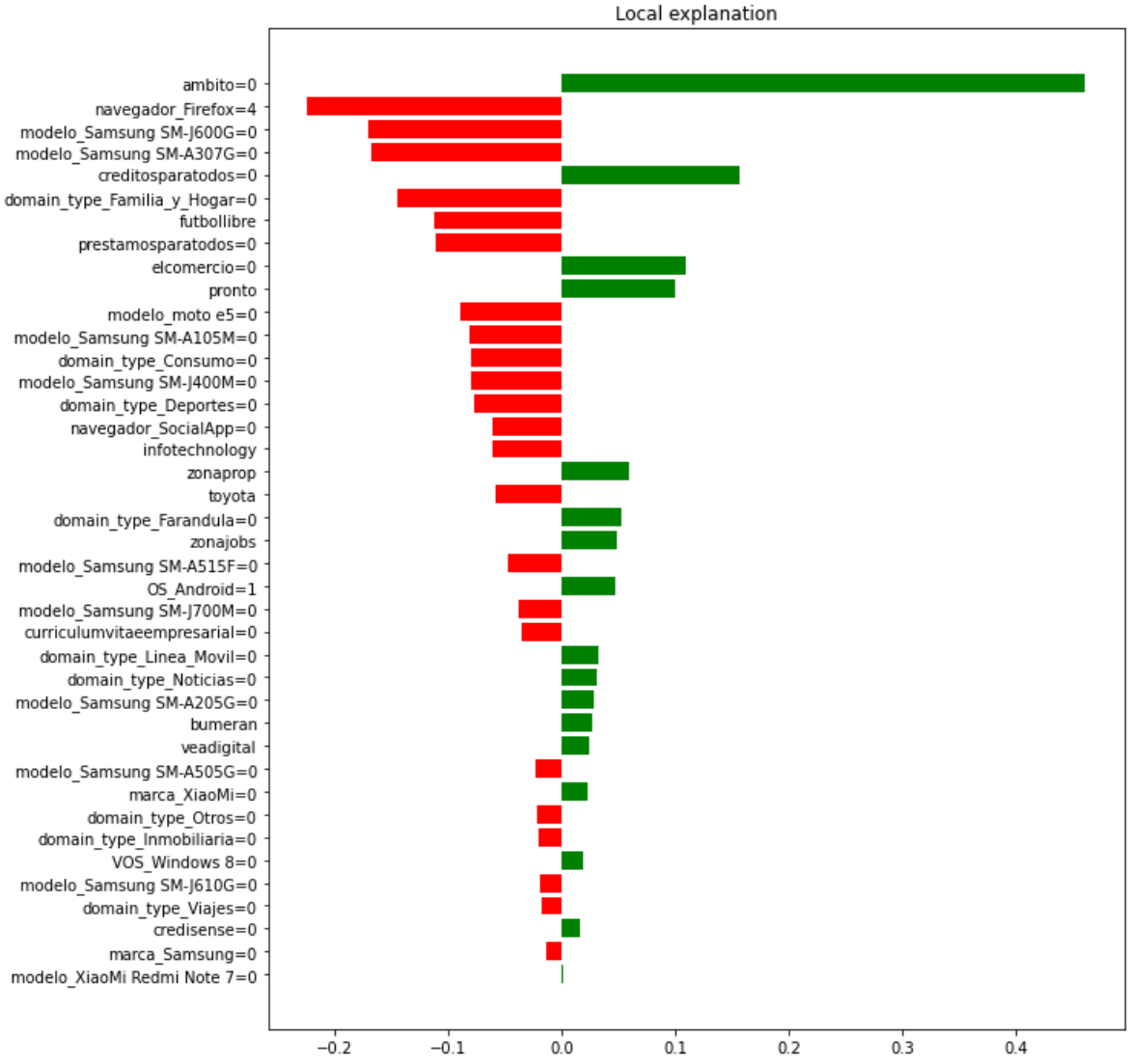


Figura 99: Gráfico de Aproximación Local LIME

Con el método LIME se puede observar si el efecto de cada variable incide de forma positiva o negativa en la variable dependiente target. Por ejemplo, la variable "ambito" influye positivamente en la variable dependiente, como también lo hacen las variables "elcomercio" y "pronto", entre otras. Siendo que

variables como “navegador_Firefox” y “futbillibre” influyen negativamente a la variable target, donde la clase 1 es femenino y la clase 0 es masculino.

Sin embargo, el método LIME presenta algunas desventajas. Genera perturbaciones en el dataset para aproximar localmente el la efecto de cada variable con un modelo interpretable y no incluye en sus cálculos ninguna medida global de comparación, como el promedio, sino que se concentra meramente en la aproximación local. Por otro lado, al aproximar localmente generando observaciones sintéticas alrededor del punto a explicar, la definición correcta de “neighborhood” de un punto es un problema no resuelto al usar LIME con data tabular. Otro problema es que ignora la correlación entre features al samplear y eso puede generar muestras poco realistas. Por lo cual, se presenta otro método de importancia local, conocido como **SHAP**.

Este método busca entender el impacto de cada variable al analizar su contribución en todas las coaliciones posibles de una cantidad finita de features, y no en perturbaciones generadas arbitrariamente, solucionando la desventaja de LIME sobre la elección arbitraria de los entornos a perturbar. Además, SHAP evalúa la importancia de las variables al cuantificar la contribución de éstas para una instancia particular comparada con la predicción promedio del dataset.

En particular, el concepto de **Shapley Value** surge en el campo de teoría de juegos, como un método de distribución de riquezas en un contexto de juegos cooperativos. Para cada juego se asigna un único reparto entre los jugadores del beneficio total generado por la coalición de todos los jugadores. El valor de shapley espera responder a la pregunta *¿Qué tan importante es cada jugador para la cooperación global, y qué recompensa puede él o ella esperar?*. Por lo que, trasladando el concepto al ámbito de machine learning, una predicción puede ser explicada asumiendo que cada valor de un feature en esa instancia es un jugador, y la predicción es el beneficio o “payout” generado por la combinación de esos valores o jugadores. El shapley value muestra cómo distribuir ese “payout” (la predicción) entre los features de manera justa, según cuánto aporten al mismo. Es entonces la **contribución marginal promedio de un valor de atributo a través de todas las coaliciones posibles de predictores**. (Se compara para cada coalición de predictores cuánto dio la predicción versus la predicción promedio y habrá valores distintos de diferencias, con y sin cierto feature. El valor de shapley es una especie de resumen para esas posibles diferencias. La idea es ver cuánto cambia una predicción según yo tenga feature X1 y X2, versus tener X1, X2 y X3 por ejemplo, siempre en referencia a la predicción promedio).

El concepto detrás del cálculo de la importancia de *features* mediante el método SHAP es el siguiente: *features* con altos valores absolutos Shapley son importantes. De esta forma, para calcular la importancia global, simplemente se suman los valores absolutos Shapley por *feature* a lo largo del *dataset:*

$$I_j = \sum_{i=1}^{n} |\phi_j^{(i)}|$$

Si bien el método SHAP es una solución que permite ir de la importancia local a la importancia global, existe una gran diferencia conceptual entre éste y la técnica de permutación: mientras que el método

SHAP se basa en la magnitud de las atribuciones de cada *feature* a las predicciones, el método de importancia mediante la permutación de variables se basa en el decrecimiento de una función de error en la *perfomance* de un modelo.

A continuación, se presentan algunas visualizaciones obtenidas a partir del método SHAP. Dicho método permite no solo entender la importancia de las variables a la hora de predecir sobre nuevas observaciones, sino también entender si su impacto es positivo o negativo sobre la variable target.

El siguiente plot se conoce como "*Individual Force Plot*".

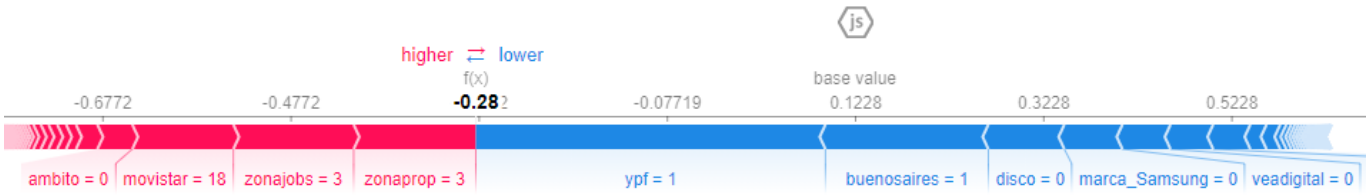


Figura 100: Gráfico "Individual Force Plot" para una determinada observación del dataset train

El ejemplo anterior apunta a una determinada instancia en el conjunto de datos X y elige usar TreeExplainer para calcular las contribuciones de los features de la instancia. El valor base es la predicción promedio de todas las instancias del conjunto de datos X y f(x) es la predicción de la instancia actual.

Se puede observar que los atributos en azul tienen una contribución negativa a la predicción respecto a la predicción promedio, los que figuran en rojo tienen una contribución positiva. Luego, el poder de contribución de cada feature se refleja en el ancho de la característica. En este plot, las variables zonajobs, zonaprop, movistar y ambito tienen un efecto positivo sobre la variable dependiente, mientras que ypf, buenosaires, disco, veadigital y marca_Samsumg, para esta observación particular reflejan efecto negativo sobre la variable respuesta.

Otro caso, para otra instancia del conjunto de datos, se obtiene el siguiente Force Plot. En el cual se observa que disco tiene un claro efecto positivo en la predicción respecto a la predicción promedio, al igual que *ambito, buenosaires, futbollibre* y *cronista*. Variables como *zonaprop*, *zonajobs*, *marca_Samsung* y *bumeran* reflejan un efecto negativo respecto a la predicción promedio en este caso puntual.

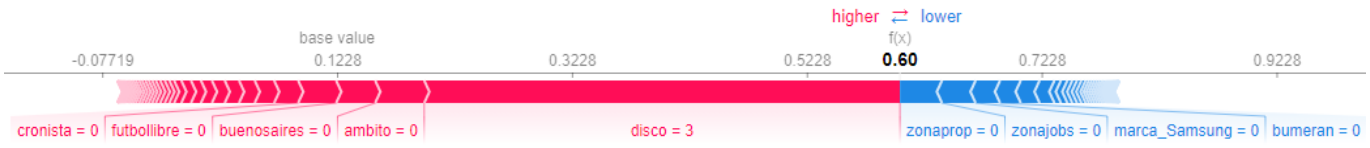


Figura 101: Gráfico "Individual Force Plot" para una determinada observación del dataset train

Luego se obtiene el "*Summary Plot*". Este chart combina la importancia de atributos con los efectos de los atributos ya que muestra relaciones positivas y negativas de los predictores con la variable target. Cada punto en el Summary Plot es un valor Shapley para un feature y una instancia. Por lo tanto, el gráfico permite observar:

-*Feature Importance* ya que las variables se muestran en orden descendiente según importancia.

-*Impacto o efecto* dado por el eje horizontal que muestra si el efecto del valor está asociado a una mayor o menor predicción.

-*Valor original,* dado por el color que evidencia si la variable tiene valores altos (rojo) o bajos (azul) para esa observación.

-*Correlación*, ya que permite observar si una variable está correlacionada de forma positiva o negativa con la variable objetivo y en qué magnitud, dado por valores del eje x y el color, respectivamente.

Para este caso se aprecia, entre otras cosas, que valores altos de SHAP para las variables pronto, disco, zonaprop y zonajobs están relacionados con un alto y positivo valor de target. Mientras que valores bajos de SHAP para dichas variables están relacionados con un bajo valor de la variable dependiente. A su vez, valores bajos de SHAP para variables como ambito, cronista, navegador_Firefox e infotechnology, entre otras, se relacionan con alto valor de la variable respuesta.

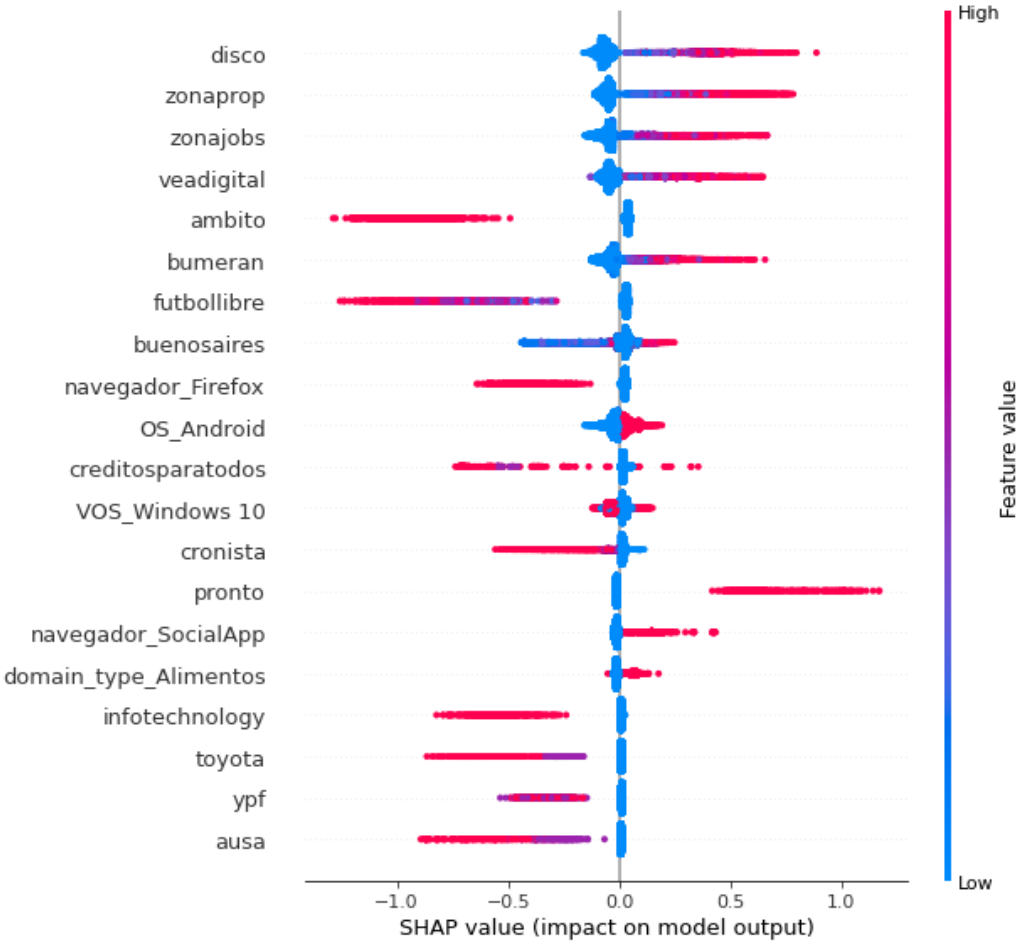


Figura 102: Gráfico "Summary Plot"

Otros gráficos son los llamados "*Feature dependence plot*", realizados para algunas de las variables del modelo. De esta forma, se obtiene mayor detalle sobre las variables más relevantes, con respecto al anterior *Summary Plot.*

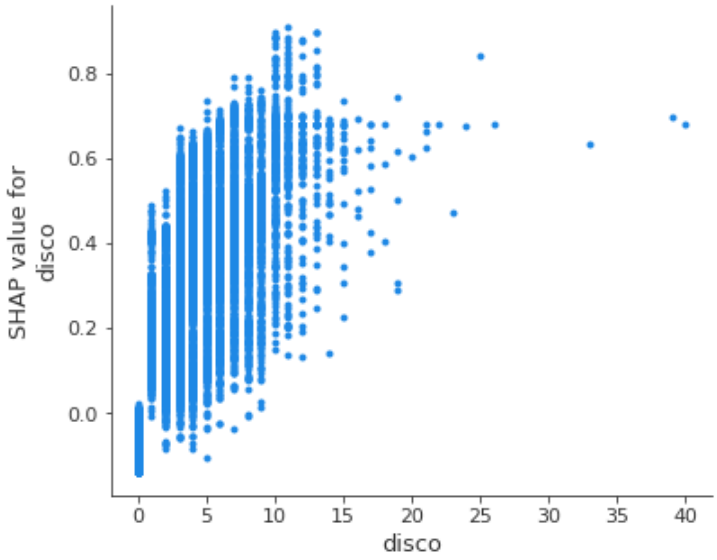


Figura 103: Gráfico "Feature Dependence Plot" en atributo "disco"

Los valores altos de la variable disco están relacionados con valores altos de SHAP.

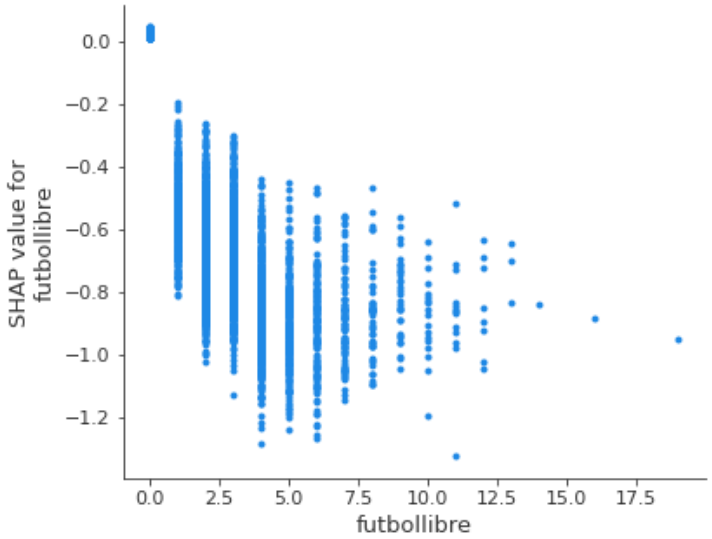


Figura 104: Gráfico "Feature Dependence Plot" en atributo "futbollibre"

A medida que aumenta el valor de la variable futbollibre, disminuyen los valores SHAP hasta el valor 6 de futbolibre.

Por último se muestran algunos resultados del "Collective Force Plot". Cada observación tiene su propia gráfica de Force Plot. Si todas las gráficas individuales de Force Plot se combinan, se rotan 90 grados y se apilan horizontalmente, se obtiene la gráfica de Collective Force Plot para los datos X del conjunto test. El eje Y es el eje X del gráfico "Individual Force Plot". Y el eje X tiene tantas observaciones como filas tenga el conjunto de datos utilizado. Por su parte, la lógica de colores es la misma que en el Individual Force Plot.

Ordenado por similarity:

Figura 105: Gráfico "Collective Force Plot" ordenado por similitud de shapley values

Al ordenar por similitud a las variables en el collective force plot, puede visualizarse a grandes rasgos que muchas de las observaciones poseen valores de shapley que empujan para arriba la predicción (es decir, hacia el 1) al apreciarse más color rojo a lo largo de la gráfica. Sin embargo, tanto en el principio como en el final del chart, se ubican los registros con valores de shapley que influyen negativamente en la predicción, es decir la empujan hacia el 0.

Por ejemplo, el siguiente gráfico está ordenado por la variable "veadigital", donde se observa que para el valor 6 de dicho atributo, en promedio los dispositivos visitan 4,5 veces el dominio *sancorsalud* y cero veces el de *futbollibre*, lo cual contribuye a mayor probabilidad de que el registro sea femenino (valor de predicción 1).

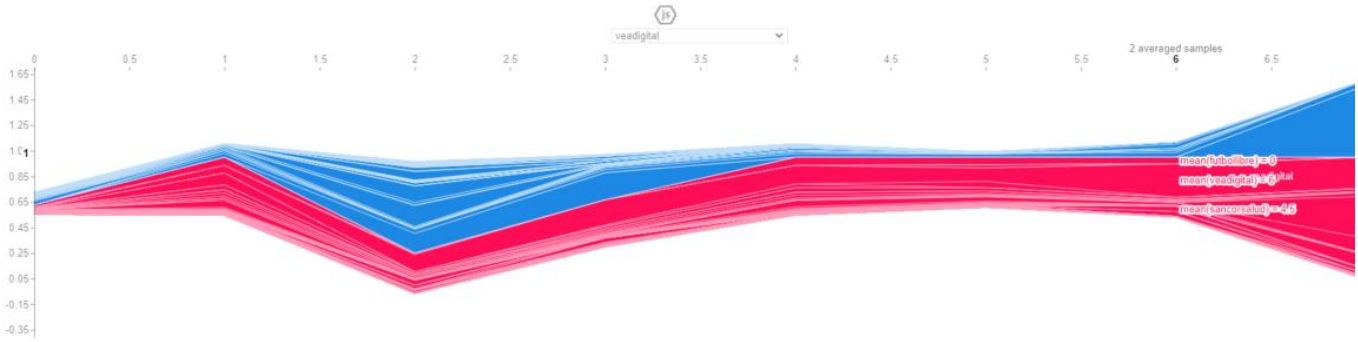


Figura 106: Gráfico "Collective Force Plot" ordenado por "veadigital"

Al ordenar por la variable "zonajobs", no se percibe una fuerte tendencia hacia el 0 o hacia el 1 en términos de predicción. Sino que para ciertos valores de dicho atributo existen ciertos picos "positivos" (picos en rojo) como así también "negativos" (picos en azul) de otras variables. Se puede apreciar por ejemplo que, para el valor 64 del atributo, en promedio los dispositivos entran 2 veces al *almundo* y 0 veces a *futbollibre*, y esto contribuye a una alta probabilidad de género femenino.

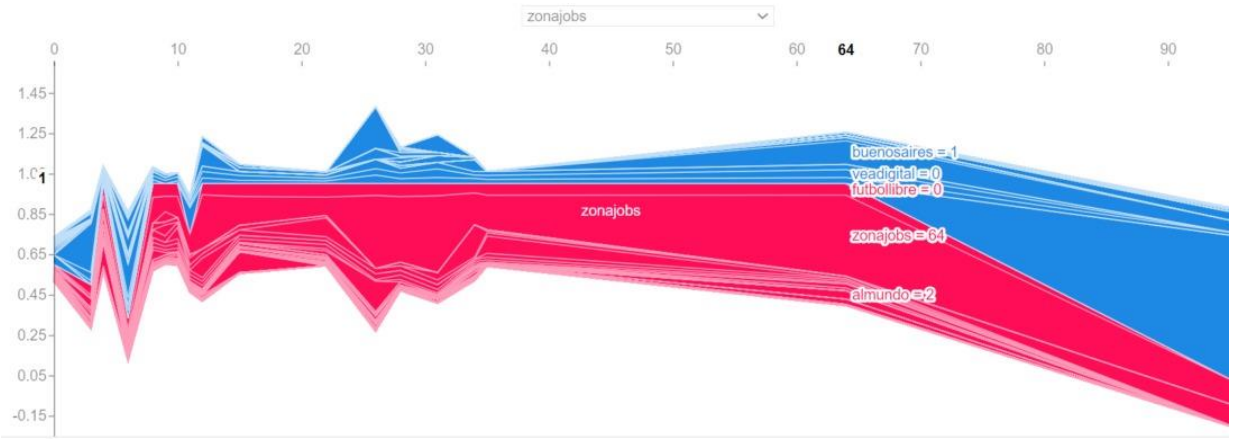


Figura 107: Gráfico "Collective Force Plot" ordenado por "zonajobs"

7. Adaptación a Rangos de edad

En esta sección se plantea una adaptación del modelo XGBoost para predecir rangos de edad, en lugar de género, con los atributos de user agent, dominios y sitios web descritos anteriormente. Como se mencionó anteriormente, los datos demográficos de usuarios de dispositivos son frecuentemente requeridos por cualquier audiencia ofrecida a agencias de publicidad por tener gran relevancia al momento de dirigir una campaña publicitaria online.

Para este caso, las clases a predecir son 6, correspondientes a distintos rangos etarios.

- **[18 - 24]** (label 1)
- **[25 - 34]** (label 2)
- **[35 - 44]** (label 3)
- **[45 - 54]** (label 4)
- **[55 - 64]** (label 5)
- **[+65)** (label 6)

Se cuenta con una muestra con mayorías en rangos de 25-34 y 35-44, como lo muestra el gráfico de barras.

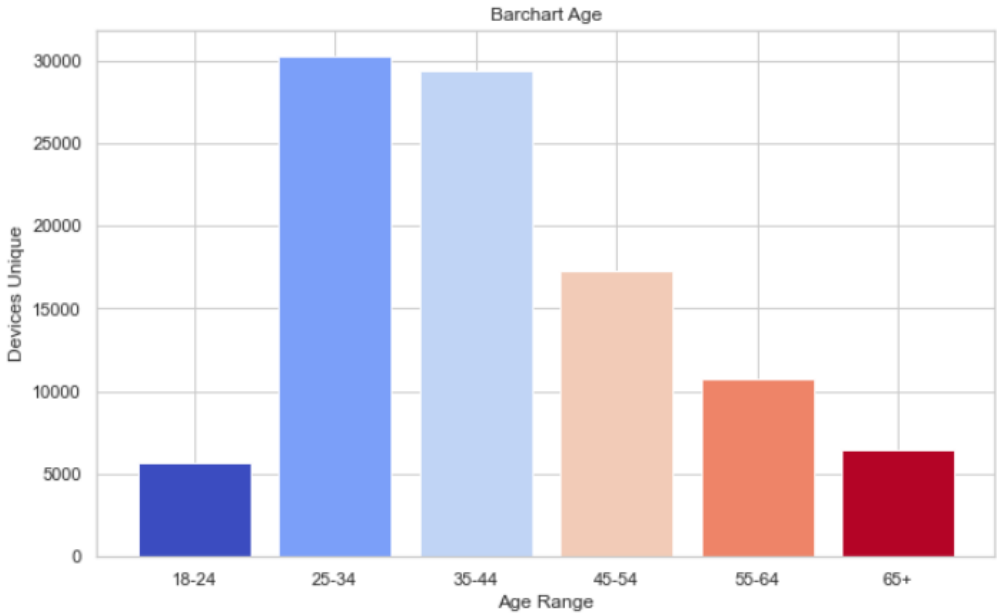


Figura 108: Distribución de dispositivos según Rangos de Edad

Se realiza el mismo proceso de Feature Engineering y se obtiene el mismo esquema de datos finales que se concretó en el caso del input para la predicción de género y se entrena un modelo XGBoost donde la variable target pasa a ser multiclase. Para este caso con más de una clase, el algoritmo considera que cada clase es positiva y el resto negativa en el momento de evaluar su performance. Por ejemplo, la métrica Recall para la clase 35-44 (identificado con el número 3) es:

$$Recall3 \ = \ Aciertos\ en\ dispositivos\ con\ clase\ 3\ /\ Total\ de\ dispositivos\ con\ clase\ 3$$

Todas las métricas pueden además darse en promedio respecto de las clases. Por ejemplo, promedio de acierto en cada clase:

$1/\ (|Ci|)\ *\ \sum i\ Recall\ C$, donde $|Ci|$ es el número de registros en la clase i.

Antes de correr el modelo adaptado, se realizó un análisis exploratorio para observar a grandes rasgos cómo se comporta la muestra bajo estudio respecto a los features que se consideran al momento de la predicción.

En cuanto a modelos de dispositivos utilizados, Samsung SM-G532M es el modelo más popular, donde los rangos 25-34 y 35-44 son los que más lo utilizan. Esto último también se ve en el resto de los modelos de dispositivos, probablemente porque ambos rangos son la mayoría en el dataset utilizado para el caso de estudio.

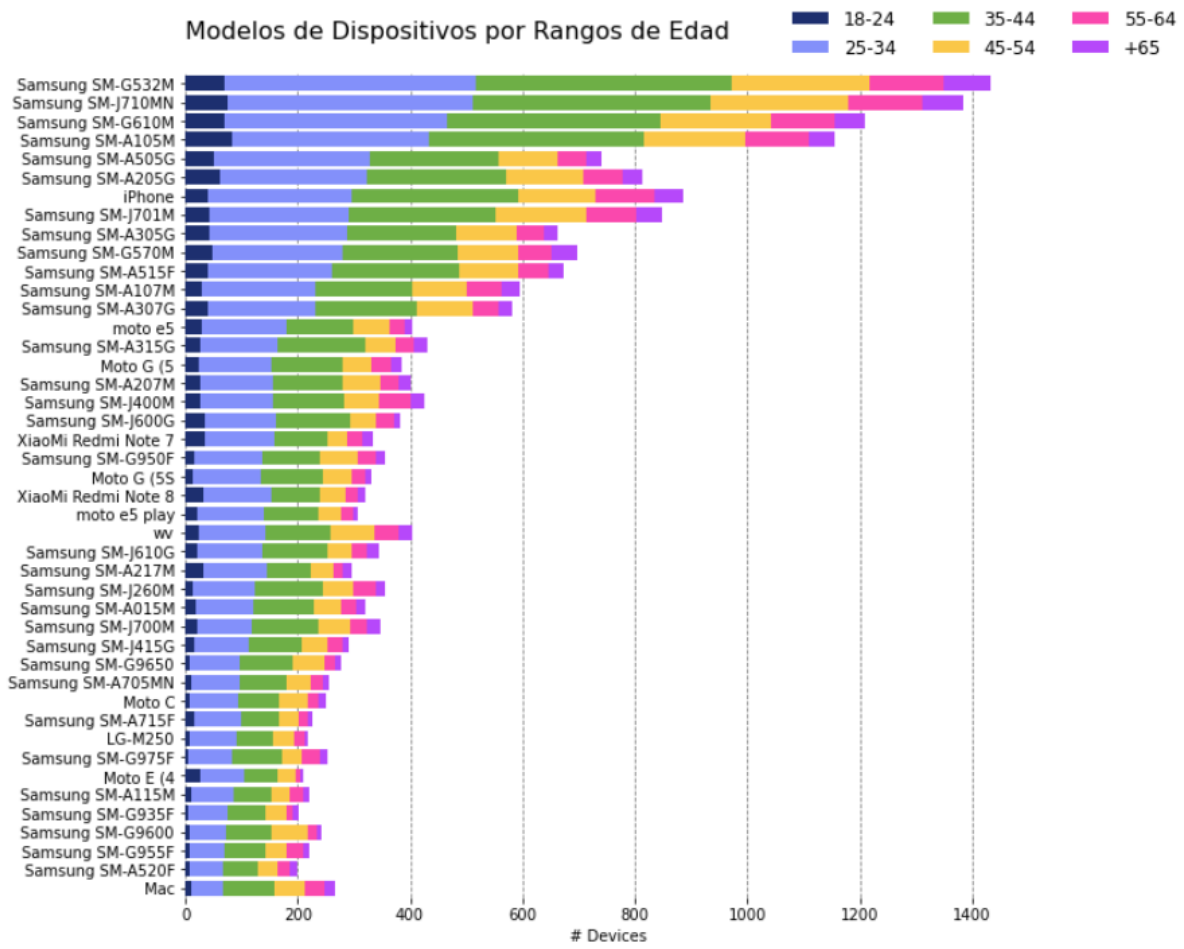


Figura 109: Valor absoluto de Cantidad de dispositivos por Modelo, según Rangos de Edad

Al observar Sistema Operativo por rangos de edad, Windows y Android son los más populares, donde nuevamente las clases 25-34 y 35-44 son las que más los utilizan. Se observa también que los rangos 45-54 y 55-64 utilizan más Windows a comparación con Android. Esto sugiere que los adultos mayores de 45 años utilizan más pc o computadoras que celulares.

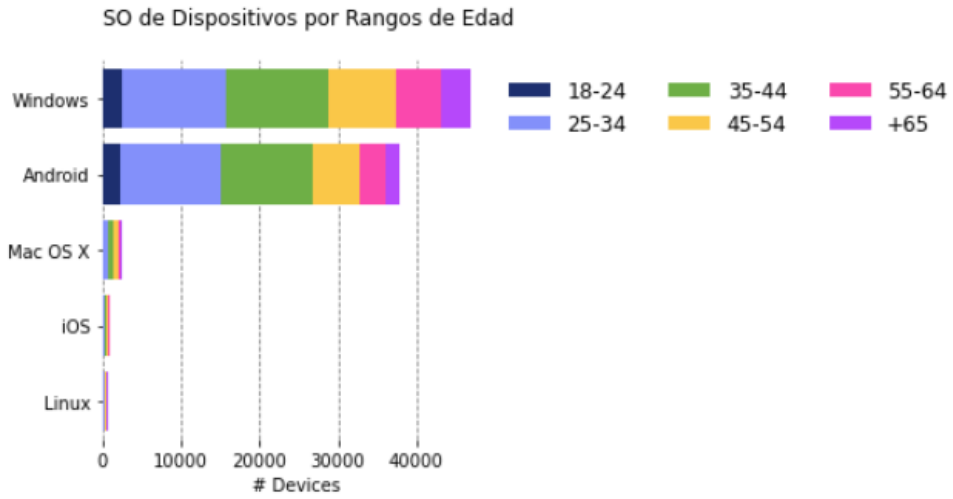


Figura 110: Valor absoluto de Cantidad de dispositivos por Sistema Operativo, según Rangos de Edad

En cuanto a Navegadores, Chrome es el más utilizado por la población bajo estudio, observándose una distribución parecida a los casos anteriores en cuanto a rangos de edad. Para el caso 18-24, sólo utilizan Chrome (aunque unos pocos utilizan SocialApp). Con lo cual Chrome puede ser un predictor importante al momento de predecir.

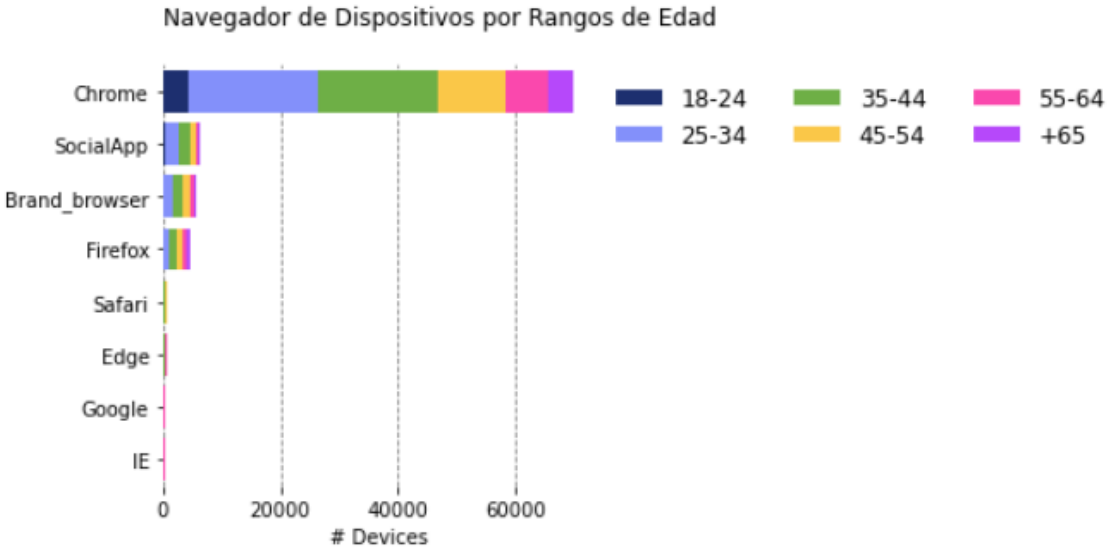


Figura 111: Valor absoluto de Cantidad de dispositivos por Navegador, según Rangos de Edad

Al observar Marcas, el rango más joven (18-24) casi únicamente utiliza Samsung. Lo mismo sucede con el rango más viejo (65 +). En este sentido, Samsung puede resultar un predictor interesante al momento de predecir.

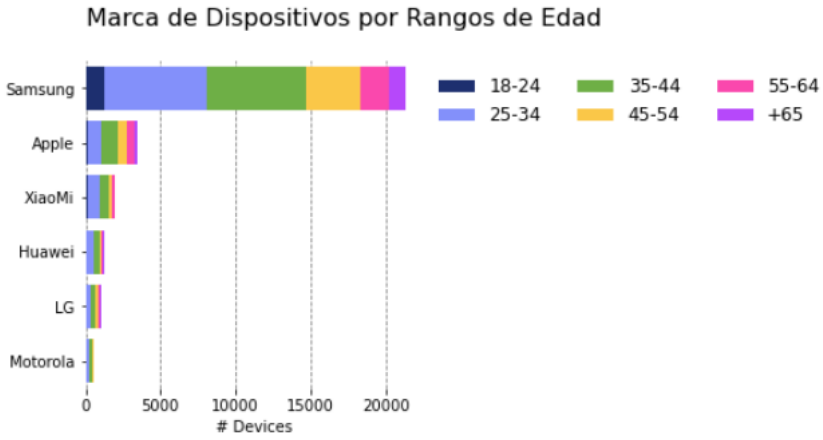


Figura 112: Valor absoluto de Cantidad de dispositivos por Marca, según Rangos de Edad

Por su parte, las variables “Is pc”, haciendo referencia a si el dispositivo es computadora, y “Is Tablet” presentan distintas distribuciones en cuanto a edades. Mientras que “Is pc” posee una distribución por rangos de edad similar a lo visto hasta el momento, “is Tablet” presenta una distribución más equitativa.

Si bien el conjunto de datos está desbalanceado, no parece existir una marcada tendencia por uno u otro rango en dicho feature, a excepción de los jóvenes que casi no utilizan tablet según los datos.

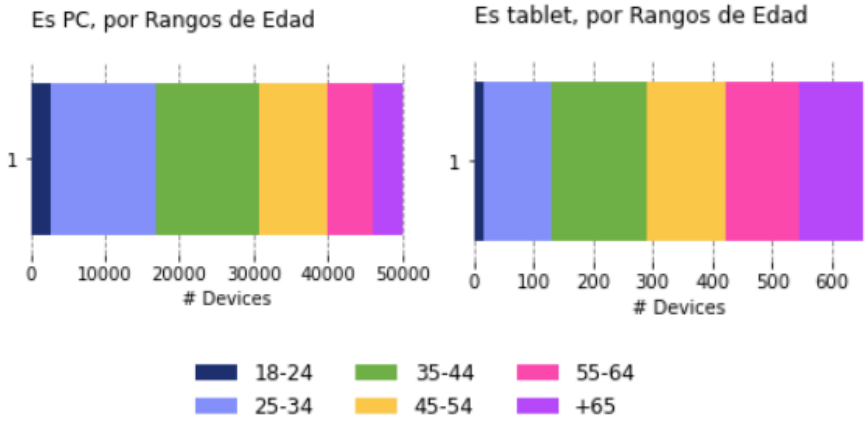


Figura 113: Valor absoluto de Cantidad de dispositivos por Is Pc y Is Tablet, según Rangos de Edad

En referencia a los dominios visitados, focalizando en el top 15 de dominios más visitados, bumeran, buenosaires.gob y disco son los dominios más visitados, con proporciones de visitas por rango de edad similares, siendo 25-34 el que más visitas hace en todos los casos. Cabe destacar, además, que el rango más joven visita más dominios referidos a búsqueda laboral (bumeran y zonajobs) en comparación con los demás dominios, con lo cual se infiere que este tipo de sitios pueden llegar a definir correctamente a la clase joven, que aparentemente buscan trabajo ni bien finalizan el secundario.

Por su parte, la clase +65 visita más dominios relacionados a supermercados (disco) y también dominios del gobierno de la ciudad de buenos aires.

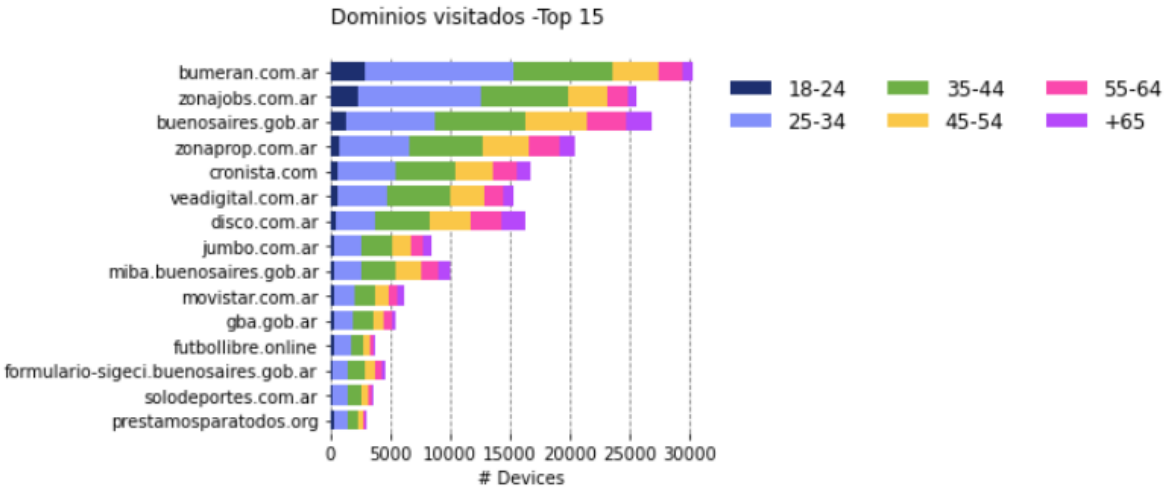


Figura 114: Valor absoluto de Cantidad de dispositivos por Dominio, según Rangos de Edad

Luego de aplicar el feature engineering descrito para el caso de género y de correr al modelo XGBoost, se observan los resultados obtenidos de esta adaptación a Rangos de Edad.

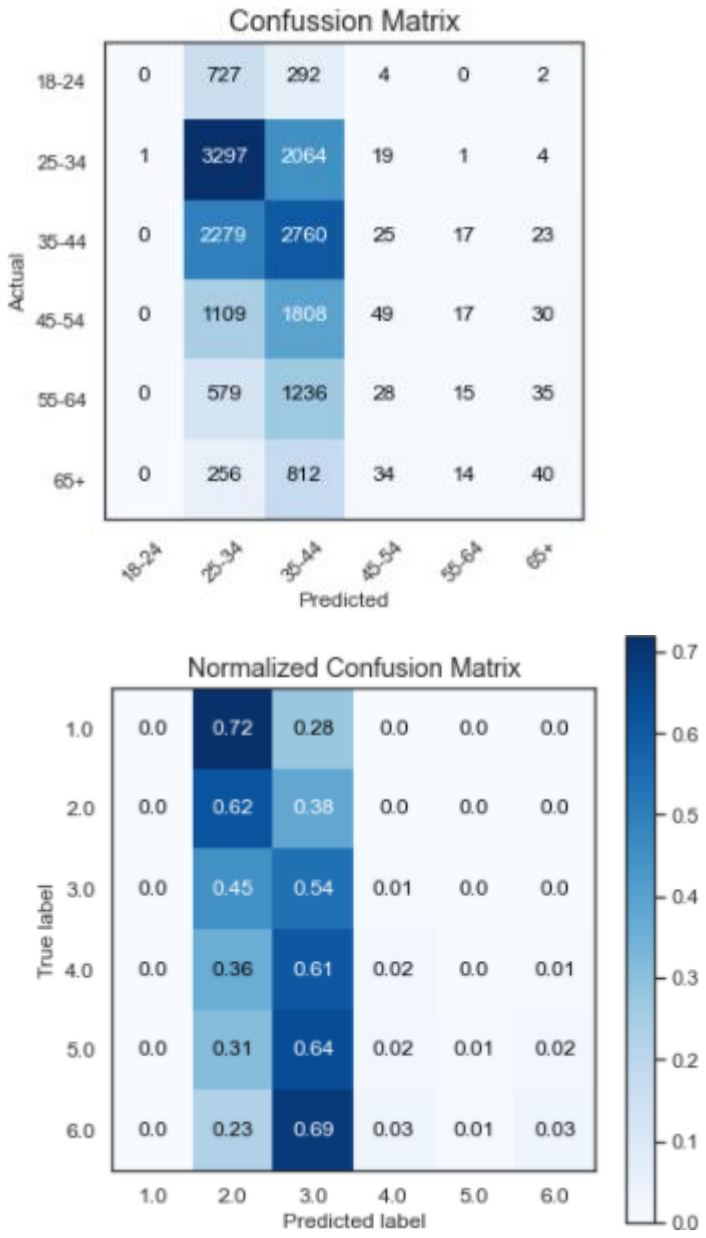


Figura 115: Matriz de Confusión, Modelo XGBoost adaptado a Rangos de Edad

La matriz de confusión muestra a los verdaderos positivos en la diagonal. Se puede apreciar entonces que, con el procesamiento de datos realizado para el caso de la predicción de género, la clase que mejor predice el modelo XGBoost para rangos de edad es la de 25-34 años (clase 2). Los falsos positivos están dados por la suma de valores de la columna i de la matriz de confusión, sin contar a la celda de diagonal de esa columna i. Y los falsos negativos son la suma de la fila i, sin la celda de la diagonal principal.

Por su parte, las áreas bajo curva ROC, curva que representa cómo va cambiando el par (FPR, R), en cada caso son las siguientes.

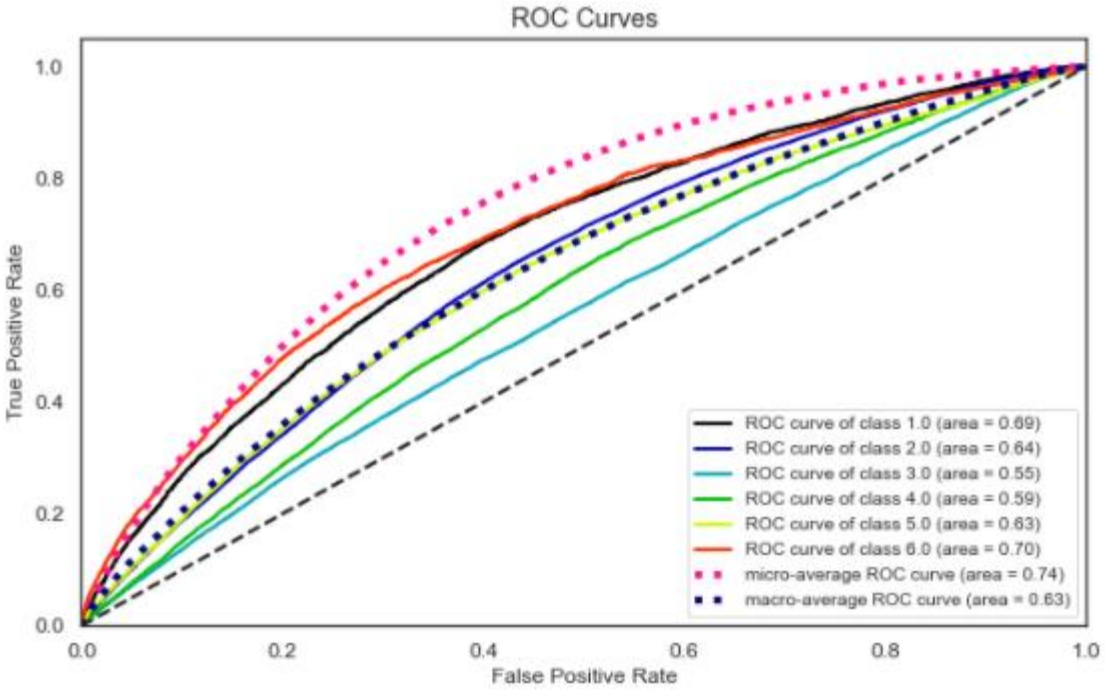


Figura 116: Curvas ROC y AUC, Modelo XGBoost adaptado a Rangos de Edad

La métrica AUC promedio es 0.63. Por su parte, la clase "65 +" es la que presenta el mayor AUC. Sin embargo, los dispositivos en este rango son muy pocos.

Finalmente se presentan a continuación las curvas precision-recall.

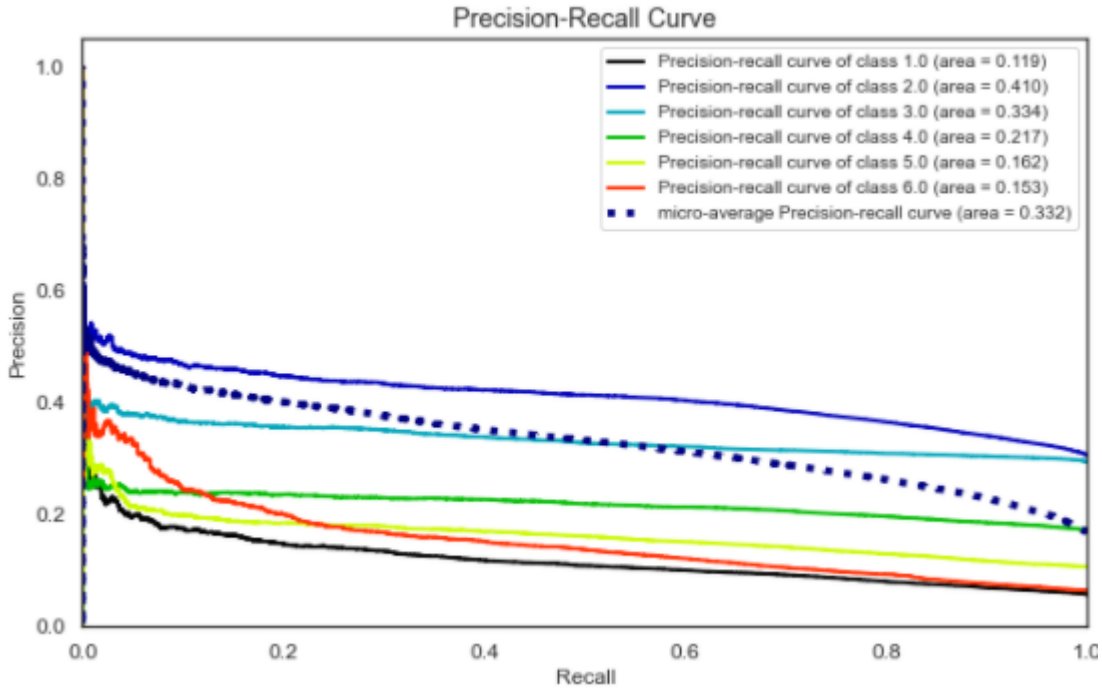


Figura 117: Curvas Precisión-Recall, Modelo XGBoost adaptado a Rangos de Edad

La curva Precision-Recall (PR) representa cómo va cambiando el par (P, R) y lo ideal es que ambos precision y recall sean altos. En este caso se observa que el micro-promedio par P,R es 0.332, alejado del 1 (cuanto más cercano a 1 el área bajo esta curva, mejor). De todos modos, cuando las clases están desequilibradas, las métricas de precisión y recall no son la mejor opción para medir la performance de un modelo. En los casos de la clase 2 y 3, precisión de alto dada la gran cantidad de dispositivos en dichas clases, a comparación con los demás rangos etarios.

Se realiza además un análisis de importancia de atributos, y se observa que bumeran, disco, dominios de búsqueda laboral y trámites bancarios lideran al momento de predecir rangos de edad.

importance	feature
0.046957	bumeran
0.045444	disco
0.041880	domain_type_Busqueda_Laboral
0.031787	domain_type_Tramites_Bancarios
0.031281	zonajobs
0.021937	buenosaires
0.020919	perfil
0.020351	prestamosparatodos
0.016510	veadigital
0.014826	animeflv
0.012463	eldia
0.012423	domain_type_Linea_Movil
0.012400	marca_Samsung
0.011722	navegador_Edge
0.011593	modelo_Samsung SM-A305G
0.011473	movistar
0.011096	jetsmart
0.010998	rojadirectatv
0.010860	domain_type_Alimentos
0.010797	credisense

Tabla 26: Top 20 - Importancia de Atributos - XGBoost

Como punto de mejora para el rendimiento del algoritmo en este caso, dado que las clases están desbalanceadas, se podría configurar el parámetro class_weight de XGBoost, indicando un peso mayor en clases como 18-24, 45-54, 55-64 y más de 65. Esto puede proporcionar cierto desvío hacia las clases minoritarias mientras se entrena el modelo y, por lo tanto, ayuda a mejorar el rendimiento del modelo al clasificar varias clases, al contrabalancear respecto a las clases con gran cantidad de observaciones asociadas. Otras posibles técnicas para equilibrar las clases consisten en eliminar muestras de clases sobre representadas, es decir de las clases 2 y 3 en este caso. O bien agregar más observaciones a las clases subrepresentadas (1, 4, 5 y 6). Otra opción es aplicar el método “SMOTE” antes de entrenar el modelo.

Este es un método de sobre muestreo. Es decir, crea muestras sintéticas o ficticias de las clases minoritarias.

8. Resultados

Al comparar las métricas obtenidas en la primera etapa de modelos, bajo el estudio de features de User Agent, el mejor modelo en cuanto a performance es Random Forest. Se detallan en la siguiente tabla todas las métricas de este primer ejercicio.

Modelos	Accuracy	AUC	Recall	Precisión	F1-Score
Baseline	0.503	0.561	0.241	**0.562**	0.337
Logistic Regression	0.545	0.5404	**0.893**	0.541	0.682
Random Forrest	**0.548**	**0.5437**	0.89	0.543	**0.68**
XGBoost	0.547	0.5421	0.89	0.542	**0.683**

Tabla 27: Comparación de métrica en primera etapa

Agregando los atributos pre procesados de dominio, Random Forest sigue siendo el mejor modelo.

Modelos	Accuracy	AUC	Recall	Precisión	F1-Score
Baseline	0.512	0.5490	0.282	**0.57**	0.428
Logistic Regression	**0.567**	0.5837	0.748	0.566	0.638
Random Forrest	0.566	**0.5910**	**0.807**	0.56	**0.667**
XGBoost	**0.567**	0.5903	0.785	0.562	0.645

Tabla 28: Comparación de métrica en segunda etapa

Finalmente, teniendo en cuenta atributos de user agent, de dominio y de sitios web visitados, el modelo final elegido es el **XGBoost**, el cual presenta los mejores resultados.

Modelos	Accuracy	AUC	Recall	Precisión	F1-Score
Baseline	0.579	0.605	**0.823**	0.567	0.672
Logistic Regression	0.608	0.632	0.798	0.594	**0.682**
Random Forrest	0.616	0.646	0.779	0.604	0.681
XGBoost	**0.617**	**0.65**	0.771	**0.605**	0.679

Tabla 29: Comparación de métrica en última etapa

Por su parte, la curva ROC y el área bajo la curva entre los modelos ganadores presenta una clara mejoría a medida que se agregan features que hacen referencia a características de navegación del usuario. Es decir que los datos sobre navegación web, además de los datos sobre user agent, definen intereses que logran ser captados por los algoritmos de machine learning para separar a los usuarios según su género.

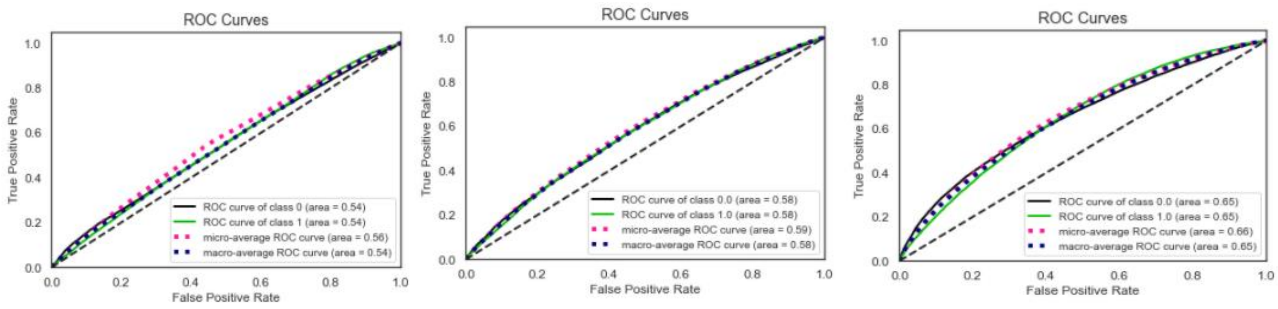


Figura 118: AUC Utilizando atributos de User Agent, luego sumándoles atributos de Dominio y finalmente sumando atributos de Urls

Al observar las curvas y áreas bajo las curvas Precision-Recall, también se aprecian mejoras a medida que se agregan atributos de navegación web.

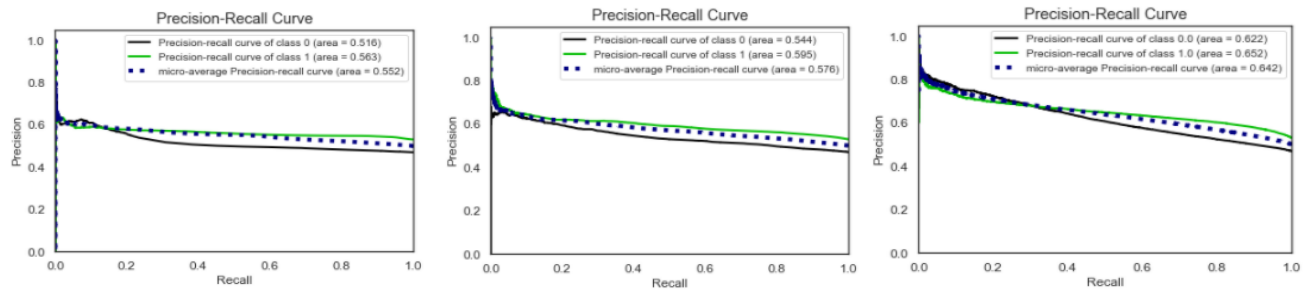


Figura 119: Precision-Recall Curve utilizando atributos de User Agent, luego incorporando atributos de Dominio y finalmente incorporando atributos de Urls

Siendo así, al basar la clasificación de género de dispositivos teniendo en cuenta atributos que tienen que ver con modelos y marcas de los mismos, navegador, sistema operativo y su versión utilizados, dominios visitados y sitios web específicos visitados, se logrará aumentar el volumen de audiencias ofrecidas con el consecuente mayor revenue mejorando la calidad de data entregada a los clientes dado que el modelo elegido no predice género aleatoriamente sino que intenta separar las clases lo mejor posible en base a los datos de los atributos mencionados.

9. Conclusiones

En particular, este trabajo se concentró en el análisis y predicción de género, requisito muy importante al momento de ofrecer audiencias en el mercado de marketing digital. Los modelos de machine learning entrenados permiten aumentar la cantidad de usuarios, con campos demográficos completos, que puede ser ofrecida a los anunciantes.

En base a datos de navegación web y User Agent de Enero 2021 se estructuró el proyecto en 3 etapas. En primer lugar, se analizó la performance de distintos algoritmos de aprendizaje automático considerando únicamente datos de user agent. El modelo que mejores resultados dio fue Random Forest con un AUC de 0.54. En una segunda etapa se incorporan features de dominios. En este caso el modelo que mejor performance mostró también fue un Random Forest que obtiene un AUC de 0.59. Finalmente, en una tercera etapa, se agregan features de Urls. En este caso, el algoritmo ganador fue XGBoost, superando al

random forest de la etapa anterior con un AUC de 0.65. Con lo cual, se logrará ofrecer cierto volumen de usuarios en audiencias que requieren atributos de género. Este campo es valioso para los auspiciantes porque permite llegar al consumidor correcto con mayor probabilidad.

En relación a los features del modelo elegido, se observó que, en cuanto a importancia global en el modelo final, las variables que más impactan al momento de predecir son “buenosaires”, “disco”, “ambito”, “bumeran” y “futbollibre”, entre las más relevantes. Por otro lado, en cuanto a la importancia local, se encontró que features como “ambito”, “creditosparatodos”, “pronto” y “zonaprop” tienen un impacto positivo sobre la predicción. Es decir que influyen positivamente en la predicción del género femenino. Mientras que variables como “navegador_Firefox”, “futbollibre” y “toyota” impactan negativamente. Al mismo tiempo, las variables relacionadas a supermercados y búsqueda laboral tienen shapley values altos influyendo en la predicción positiva, mientras que atributos que tienen que ver con deportes, automóviles y noticias tienen mayormente shapley values bajos, es decir, baja contribución promedio en el modelo, que inciden en la predicción de la clase femenino, mientras que sus pocos valores shapley altos están relacionados con la predicción de género masculino. Siendo así, los datos de navegación juegan un papel muy importante cuando se trata de intereses de usuarios de dispositivos, los cuales parecen definir en gran parte al género.

En cuanto a la evaluación económica, observando la sensibilidad del EVI al AUC, se aprecia que a medida que aumenta el AUC, el EVI se incrementa vía el aumento del revenue. Este impacto fue estimado en base a los coeficientes de la regresión lineal mencionada en la sección anterior. Se observa que aplicando el algoritmo XGBoost, con el AUC obtenido de 0.65, el EVI es mayor al que se obtiene con un AUC de 0.5, siendo este último valor el que puede corresponderse con la performance del algoritmo de la moneda. Con lo cual, si bien se ofrecen menos registros con el nuevo modelo, los mismos tienen un grado considerable de aciertos en las clases de género. Además, se evidencia que si se logra mejorar el modelo con nuevas features y con el pre procesamiento de los datos, el EVI aumentaría dado un área bajo la curva ROC mayor. Por lo que la empresa creadora de audiencias custom percibirá mayores ingresos al contar con registros de usuarios con atributos demográficos correctamente imputados con grandes posibilidades de ser impactados según lo desee el anunciante.

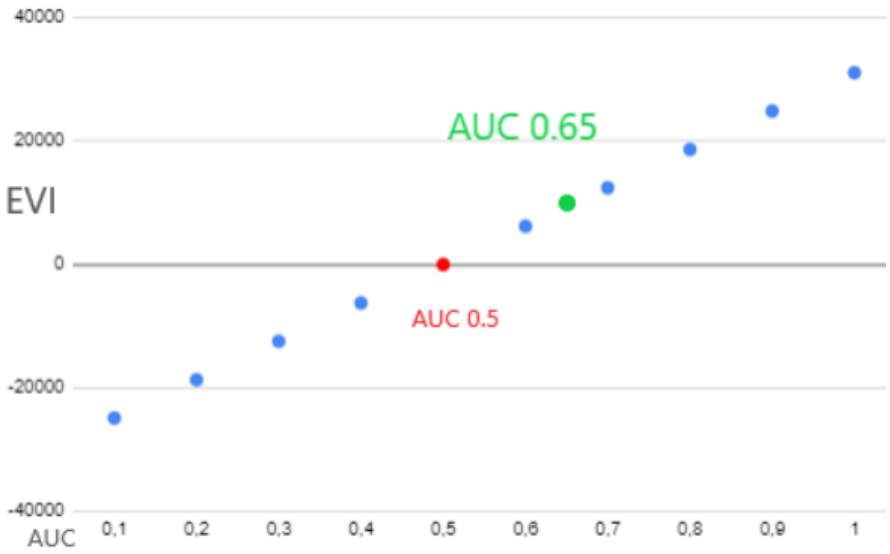


Figura 120: Sensibilidad EVI con AUC, resaltando la performance del Modelo XGBoost vs Algoritmo Moneda

El nuevo activo: *dispositivos con género asignado mediante un modelo de Machine Learning (XGBoost)* generará un valor económico de información mensual positivo a la empresa creadora de audiencias. En particular, el EVI será de aproximadamente **9.276,9** USD en base a 57.725 registros modelados a vender.

Por lo tanto, además de un valor económico positivo para la empresa creadora de audiencias, en un mercado grande como lo es la venta de productos o servicios mediante plataformas digitales, contar con el servicio de una compañía dedicada a la venta de audiencias customizadas según requerimientos del cliente podría generar mejores impactos dado un género correctamente asignado.

Oportunidades de Mejora

Es posible mencionar algunas opciones a considerar para mejorar las métricas del modelo. En primer lugar, los features de user agent pueden ser importantes. Como posibles mejoras al momento de pre procesarlos, una buena forma de mejorar predicciones cuando se trata de dichos atributos es hacer un scrapping de las prestaciones de los dispositivos. Es decir, realizar una investigación exhaustiva sobre las marcas y modelos, teniendo en cuenta el tamaño del dispositivo, la cantidad de pulgadas que posee la pantalla, la capacidad de almacenamiento, la memoria RAM, el color, entre otras prestaciones.

Por otro lado, como posible forma a mejorar el feature engineering en lo que respecta a atributos de navegación web, para lograr modelos con mejor performance, se podría incluir a la cantidad de visitas por dominio de las urls del tercer análisis en los rubros generados a partir de datos de dominios en el segundo análisis, sumando dicha cantidad de visitas del dispositivo en cada uno de los rubros. De este modo, se reduciría la cantidad de variables a incluir como input en los algoritmos de machine learning.

10. Posibles Extensiones

10.1. Boruta: Otra alternativa para seleccionar variables

Boruta es un método que elimina de forma iterativa features que son estadísticamente menos relevantes que un conjunto de variables de ruido artificiales que introduce el mismo algoritmo de Boruta. En cada iteración, las variables rechazadas se eliminan para la siguiente iteración. Generalmente realiza una buena optimización global para la selección de atributos.
Este algoritmo se basa en dos ideas:

1- ***Shadow features***: En Boruta los features no compiten entre sí, sino que compiten con una versión aleatoria de ellos. A partir de X (set de datos train que contiene a los atributos que harán a la predicción de género en este caso) se crea otro marco de datos mezclando aleatoriamente cada feature. Estos features permutados se denominan "Shadow features". Aquí, el dataset shadow se adjunta al dataset original para obtener un nuevo dataset (*X_boruta*) que tiene el doble de columnas que X. Luego, se ajusta un Random Forest en "X_boruta" e "y" (set de datos train que contiene a la variable target (femenino / masculino)), se toma la importancia de cada atributo original y se compara con un umbral. Dicho umbral se define como el atributo de mayor importancia registrado entre los "Shadow features". Cuando la importancia de un feature es superior a ese umbral, entonces es un "acierto" o "hit". **La idea es que un atributo es importante solo si es capaz obtener un valor de *feature importance* mayor que el "*mejor feature aleatorio*".**

2- ***Binomial distribution***: Como en todo algoritmo de Machine Learning, la clave es la iteración. ¿Cuál es la probabilidad de conservar un atributo? El nivel máximo de incertidumbre sobre un atributo se expresa mediante una probabilidad del 50%. Dado que cada experimento independiente puede arrojar un resultado binario (acierto o no acierto), una serie de N ensayos sigue una distribución binomial.
En Boruta, no hay un umbral estricto entre un área de rechazo y aceptación, sino que hay 3 áreas:

1. Área de rechazo (roja en la siguiente imagen): Los features que caigan aquí serán eliminados.
2. Área de indecisión (azul): Boruta es indeciso respecto a eliminar o no a los atributos que caigan en este área.
3. Área de aceptación (verde): Los features que caigan aquí son considerados importantes, por lo que se mantiene.

Las áreas se definen seleccionando las dos colas (extremos) de la distribución. En el ejemplo de la imagen que se muestra a continuación, la variable "age" es una variable predictora y debe mantenerse. La variable "weight" es ruido y debe eliminarse. Y la variable "height" depende de nosotros, ya que Boruta se mostró indeciso en cuanto a mantenerla o no.

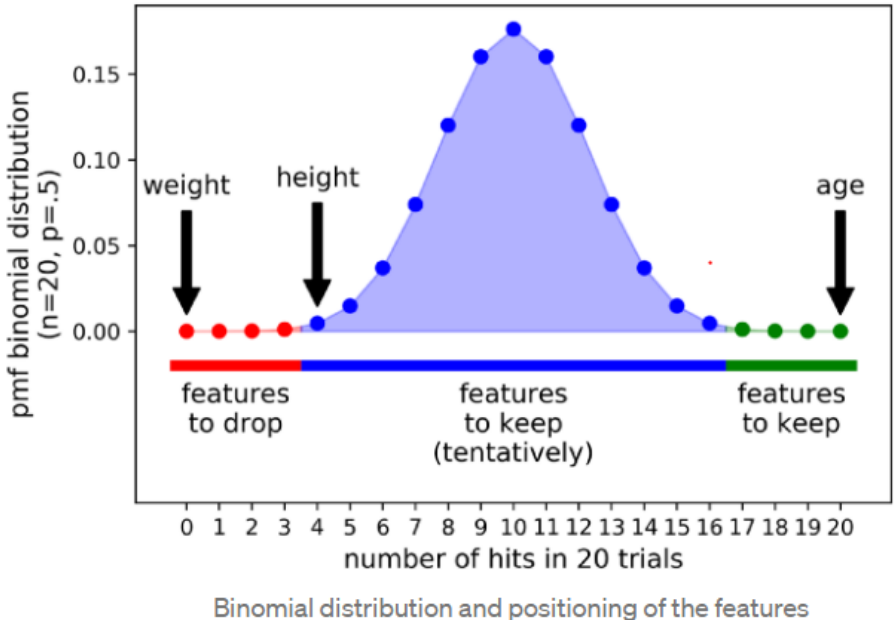


Figura 121: Distribución binomial de features[14]

Por lo tanto, Boruta resulta ser una buena alternativa para realizar una selección de variables sólida y con base estadística en su conjunto de datos. Tomar decisiones importantes sobre los features es fundamental para garantizar el éxito de un modelo predictivo. Siendo así, aplicando Boruta para luego correr el modelo XGBoost, las variables que se mantienen son:

- Is_pc
- Is_mobile
- OS_Android
- OS_Windows
- navegador_Firefox
- navegador_SocialApp
- VOS_Windows 10
- Antiguedad_ancient
- domain_type_Alimentos
- domain_type_Busqueda_Laboral
- ambito
- ausa
- autocosmos
- buenosaires
- bumeran
- creditosparatodos
- cronista

[14] Fuente: Aaron Lee (2020). Boruta Feature Selection (an Example in Python)

- diarioregistrado
- disco
- futbollibre
- infotechnology
- mundoazulgrana
- pronto
- renault
- rojadirectatv
- starbucksrewards
- toyota
- veadigital
- ypf
- zonajobs
- zonaprop

Y las nuevas métricas obtenidas por el XGBoost tomando solo dichas variables son:

Modelos	Accuracy	AUC	Recall	Precisión	F1-Score
XGBoost	**0.617**	**0.65**	0.771	**0.605**	0.679
XGBoost aplicando Boruta	**0.615**	**0.65**	0.77	**0.604**	0.677

Aplicar Boruta es una excelente opción de fácil aplicación. Pero debe tenerse en cuenta que tiene una desventaja. Posee un tiempo de ejecución medido en horas (no en segundos como las herramientas de "sklearn"). Esto hace que sea muy difícil de ajustar dado que cada ajuste de parámetro requerirá un tiempo de CPU adicional extremadamente alto. Por lo cual, debe usarse cuando dicho tiempo de ejecución "valga la pena". Por ejemplo, con sets de datos particularmente complicados, con cientos de variables predictoras potencialmente correlacionadas.

10.2. Target Encoding: Codificación de Variable Respuesta

Como alternativa al armado del dataset final para ser interpretado por algoritmos de Machine Learning, existe el método denominado "Target Encoding". En esta sección se probará al modelo final XGBoost dandole como input una estructura de dataset completamente distinta a la estructura con la que se venía trabajando a lo largo del presente caso.

La codificación de variable respuesta (Target Encoding) se define como el proceso en el cual los atributos categóricos se reemplazan por una combinación de la probabilidad posterior de la variable target dado un valor categórico particular y la probabilidad previa (prior) de la variable target sobre todos los datos de entrenamiento.

Esta funciona de la siguiente manera:

1. Agrupa los datos por cada categoría y cuenta el número de ocurrencias para cada valor de la variable respuesta.
2. Calcula la probabilidad de ocurrencia de uno de los valores de la variable target (valor 1: “Femenino” para este trabajo), dados cada valor categórico.
3. Finalmente, se genera una columna con el valor de probabilidad para ese valor de la variable target, por cada valor categórico. Se obtiene entonces un valor numérico que representa al atributo categórico (por ejemplo: *Sistema Operativo*, para el caso del presente trabajo) que puede ser interpretado por algoritmos de machine learning.

Al realizar este proceso con la librería de python *sklearn*, este toma en cuenta la probabilidad prior, obteniendola con el set de training. Es decir, la probabilidad de que la variable target sea 1 (femenino). Luego, utiliza esa métrica para ayudar a suavizar el valor codificado.

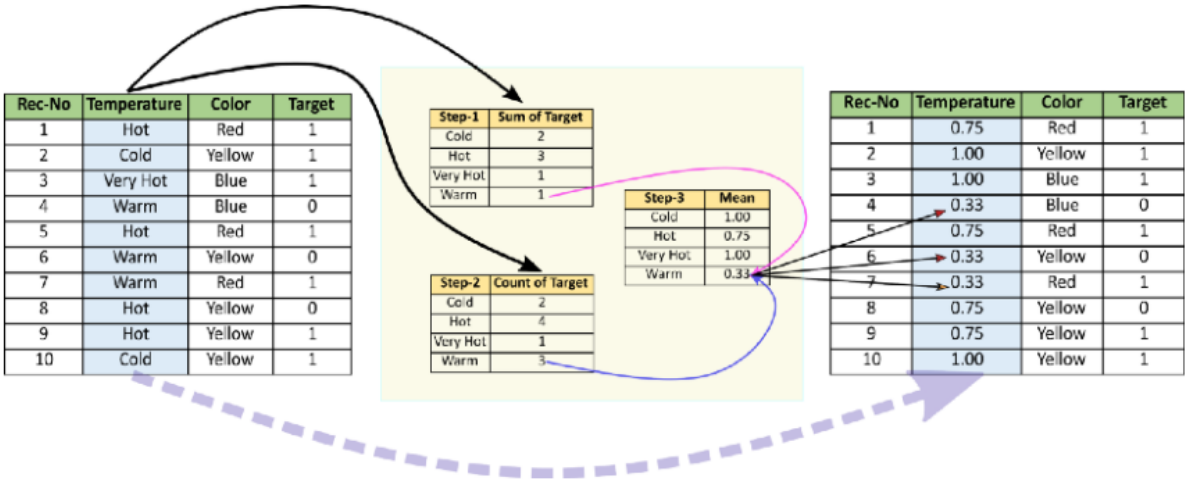


Figura 122: Target Encoding Process[15]

Al aplicar este proceso a los conjuntos de datos disponibles, la nueva estructura del dataset es la siguiente.

device_idx	Target	db_Encoded	url_Encoded	bf_Encoded	ov_Encoded	im_Encoded	of_Encoded	dom_Encoded	df_Encoded	oa_Encoded	ip_Encoded
392.0	0	0.520000	0.777695	0.541049	0.563774	0.541736	0.542094	0.537742	0.548589	0.543225	0.541410
930.0	0	0.528108	0.518895	0.527048	0.517174	0.523171	0.519493	0.548685	0.525897	0.528526	0.519275
1145.0	0	0.528108	0.063212	0.536432	0.532429	0.523171	0.540491	0.606133	0.525897	0.534338	0.541410
2465.0	1	0.528108	0.666667	0.536432	0.532429	0.523171	0.540491	0.599991	0.525897	0.534338	0.541410
2544.0	1	0.528108	0.618193	0.541049	0.517174	0.523171	0.519493	0.637097	0.525897	0.528526	0.519275

Las variables que se observan tienen aplicado el Target Encoding, excepto en el caso de “device_idx” y “Target”, donde 1 es Femenino y 0 es Masculino. Los caracteres delante de “_Encoded” como nombre de variable tienen el siguiente significado:

- **df** (Modelo Dispositivo)

[15] *Fuente: “https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02”*

- **ov** (versión del sistema operativo)
- **bf** (browser)
- **db** (Marca Dispositivo)
- **of** (sistema operativo)
- **ip** (es pc)
- **im** (es celular)
- **ti** (es tablet)
- **oa** (antigüedad del Dispositivo)

Luego de aplicar este proceso al conjunto de datos, se corre el modelo XGBoost optimizando hiperparametros mediante Random Search, aplicando Cross Validation, y se obtienen los siguientes resultados.

Modelos	Accuracy	AUC	Recall	Precisión
XGBoost	**0.581**	**0.59**	**0.731**	**0.582**

Es decir que sin realizar un profundo feature engineering como se hizo anteriormente, únicamente estructurando los datos de esta manera, en los resultados obtenidos mediante validación cruzada no se logra una performance superadora al modelo final obtenido en presente trabajo.
Sin embargo, como puntos de mejora en esta sección podría agregarse otro tipo de ingeniería de atributos, habiendo hecho previamente un análisis de datos riguroso.

11. Anexo

Anexo 1: Datos

Se muestra a continuación una vista de la matriz con data cleansing realizado desde los datos obtenidos para comenzar a ser pre procesado y formar parte del input de modelos de aprendizaje automático.

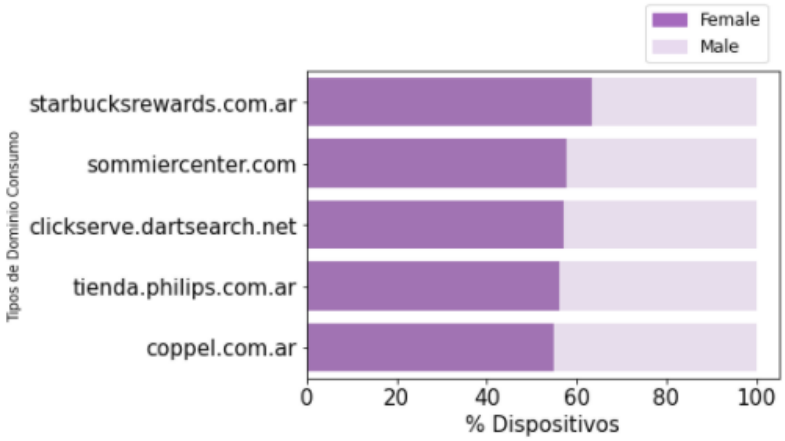
feature_type	feature_detail	device_idx	label
ov	Android 9	65465	2
bf	Chrome	5	3
dom	minutouno.com	6431131	3
url	creditosparatodos.org/gracias.php	3264654	3
db	Xiaomi	168352	2

Anexo 2: Detalles de rubros de intereses a partir de datos de Dominio

Se detallan a continuación gráficos que muestran la variabilidad respecto a la variable target de los dominios incluidos en cada rubro creado.

- *Rubro Consumo*

En lo que es Consumo, se incluyen dominios de páginas que refieren al shop online variado (electrodomésticos, indumentaria, consumo en locales gastronómicos, etc). El sitio "starbucksrewards" tiene mayoría de visitas de género femenino. Esto hace inferir el consumo de bebidas de alta gama por parte del género mencionado. En los demás dominios, es notoria también la mayoría femenina. Solo en dominios como "coppel.com.ar" el porcentaje de visitas se empareja apenas más que en los demás. Esto tiene sentido ya que dicho dominio es un e-commerce de productos variados.

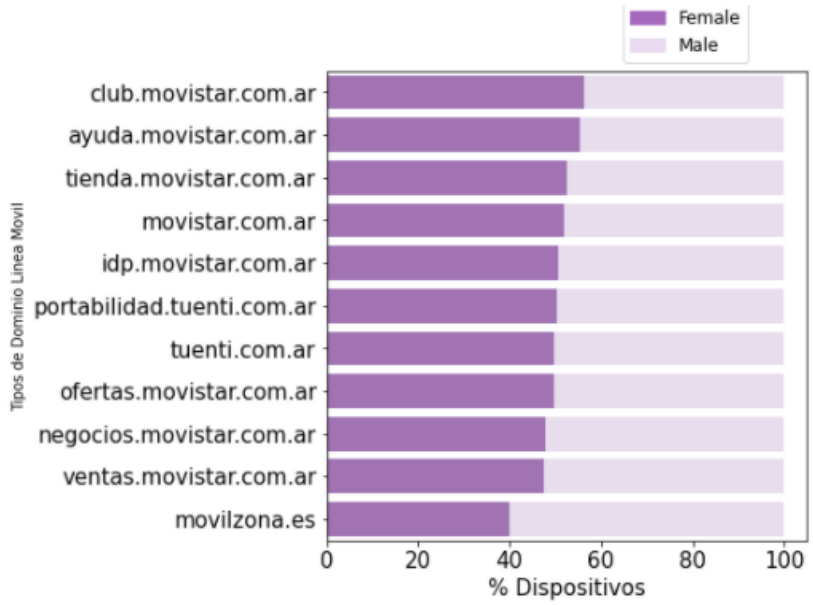


- *Rubro Viajes*

Los dominios de esta categoría pertenecen en su mayoría a la compañía *almundo.* Si bien posee mayoría femenina en términos de visitas, la misma es solo cercana a 60% contra un 40% de visitas masculinas.

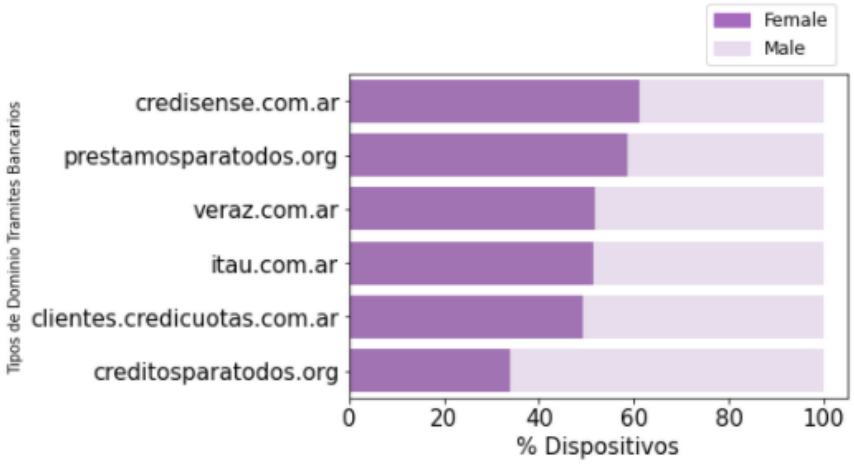
- *Rubro Linea Movil*

Este rubro se caracteriza por una proporción de visitas muy equilibrada en cuanto a género. Casi todos los dominios en esta categoría son igualmente visitados por hombres y mujeres, como puede apreciarse en el gráfico.



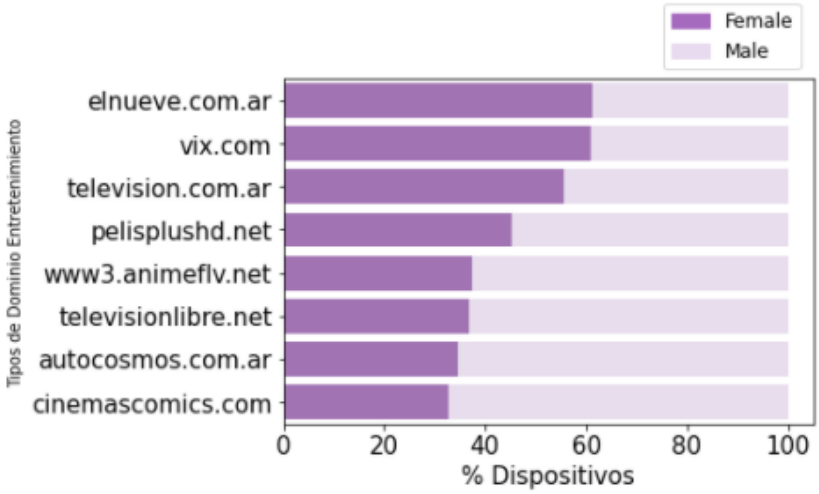
- *Rubro Trámites Bancarios*

El rubro de trámites y bancos tiene una proporción de visitas similares en cuanto a género. Sin embargo, se destacan algunos dominios con mayorías de uno y otro género. Por ejemplo, el dominio “creditosparatodos” es más visitado por hombres que por mujeres. Y el sitio de “credisense” es más frecuentado por el género femenino.



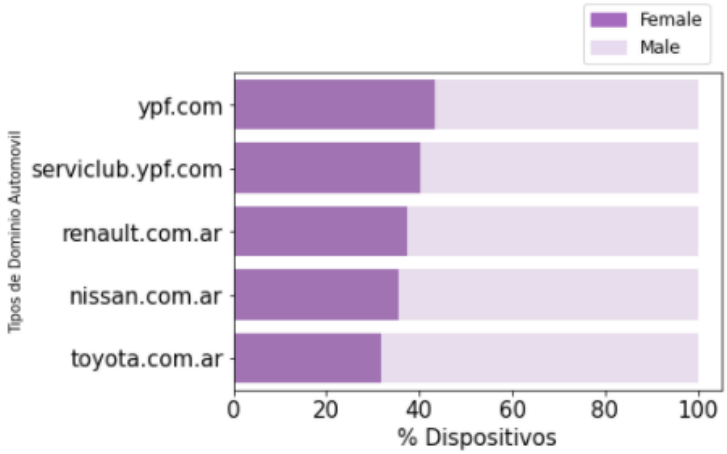
- *Rubro Entretenimiento*

Este rubro presenta cierta variabilidad en cuanto a género al observar en detalle. Por ejemplo, sitios como “elnueve.com.ar”, “vix.com” y “television.com.ar” son marcadamente más visitados por mujeres que por hombres. Y a su vez, dominios como “autocosmos”, “cinemascomics” y “televisionlibre.net” son más frecuentados por el género masculino. Tiene sentido ya que, en particular “autocosmos” tiene que ver con automóviles, rubro que ya se apreció como mayoritariamente masculino.



- *Rubro Automóvil*

Este rubro, como se ha mencionado anteriormente, presenta en general más visitas masculinas que femeninas. Al desplegar el detalle del mismo, se aprecia la fuerte tendencia masculina.

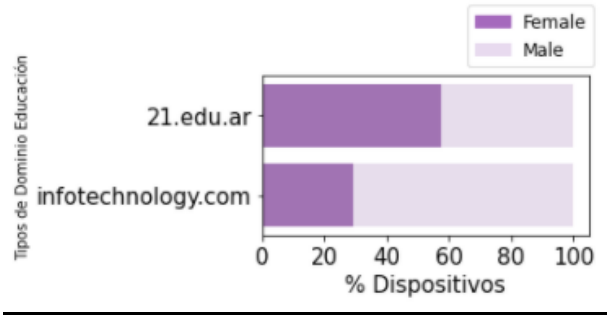


- *Rubro Educación*

El rubro de educación es mayoritariamente visitado por el género masculino. Sin embargo, se observa cierta variabilidad al observar los dominios que lo componen.

Por ejemplo, tiene el dominio “21.edu.ar” es el de la Universidad Siglo 21. Dicha universidad privada, en principio parece ser más buscada por mujeres que hombres.

A su vez, los dominios de “infotechnology” son más frecuentados por hombres. Esto hace sentido habiendo observado también las proporciones en dominios de empleos, donde el género masculino se inclina por sitios de trabajo orientados al área tecnológica.

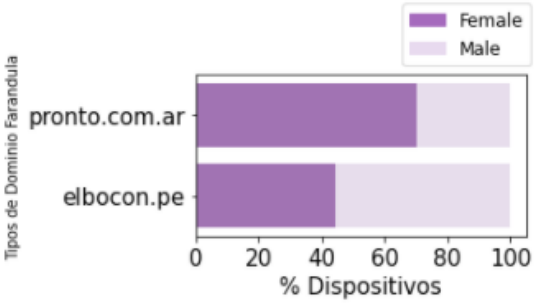


- *Rubro Familia y Hogar*

Este rubro presenta más visitas femeninas que masculinas. La mayor parte de los dominios aquí incluidos tienen que ver con bebés y niños. Esto demuestra una fuerte tendencia femenina por la ocupación del cuidado de los niños, bajo la población de estudio.

- *Rubro Farándula*

Respecto al rubro farándula, el dominio con más porcentaje de visitas de género femenino es "pronto.com.ar" lo cual refleja cierta característica de intereses por parte de dicho género ya que esta revista es casi enteramente sobre farándula y espectáculos en Argentina. Luego, dominios como "elbocon" posee más visitas masculinas que femeninas dentro del total de visitas. Esto también hace sentido ya que dicho dominio es sobre farándula en el fútbol

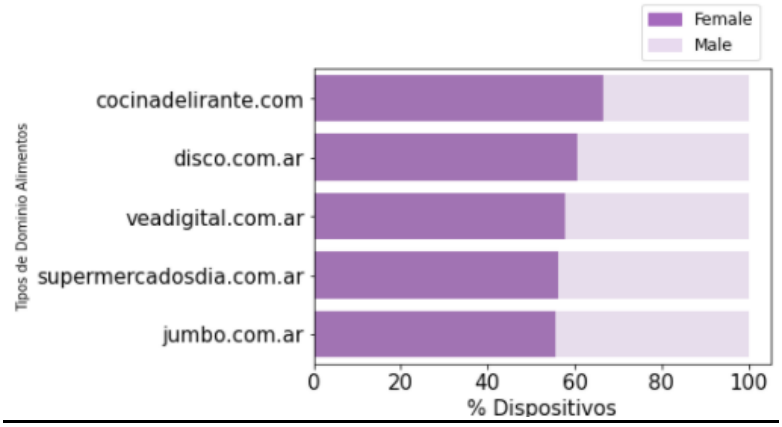


- *Rubro Salud*

Este rubro presenta en general más visitas femeninas que masculinas. El dominio “sancorsalud” es el más visitado con un 60% de usuarios con género femenino.

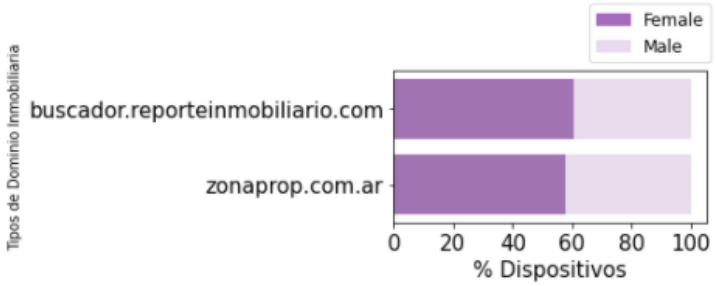
- *Rubro Alimentos*

En Alimentos, los sitios que hacen referencia a recetas de cocina y supermercados tienen porcentajes de visitas altamente femeninos, dejando ver interés y gustos por este rubro más marcado en mujeres que en hombres.



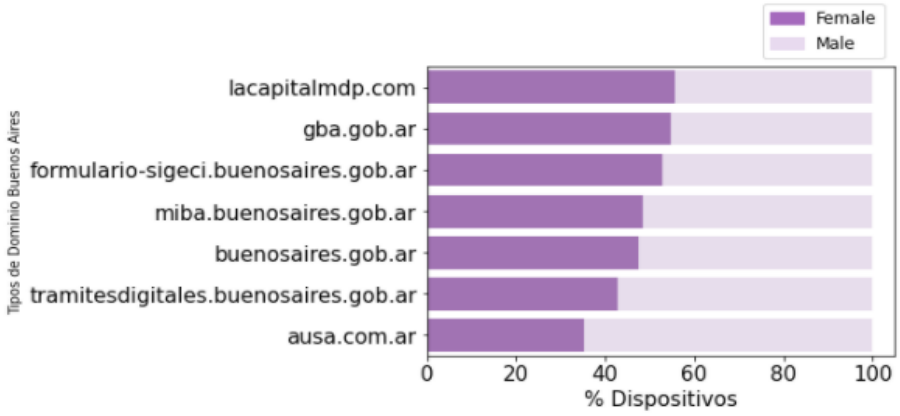
- *Rubro Inmobiliarias*

Tanto el dominio "reporteinmobiliario" como "zonaprop" son frecuentados por mayorías femeninas aunque no tan acentuadamente, alcanzando una proporción 60% mujeres - 40% hombres. Se puede apreciar que los intereses por alquiler o compra de propiedad son atendidos más por mujeres que por hombres, bajo la población de estudio.



- *Rubro Buenos Aires*

Este rubro presenta diferencias en cuanto a visitas según género especialmente en dominios referidos a trámites en la web puntual del gobierno de Buenos Aires y más acentuadamente en "ausa.com.ar", que es un sitio de una sociedad anónima que tiene al gobierno de la ciudad de Buenos Aires como principal accionista. La compañía se dedica a la construcción de autopistas e infraestructura urbana.



Anexo 3: Datasets finales pre procesados

- **Dataset final para atributos de tipo "User Agent"**

En cuanto al dataset final, para el caso de User Agent, se hizo unión de distintos datasets que se corresponden con cada tipo de feature de user agent (df, bd, oa, ip, im, it, bf, ov, of), mediante el id del dispositivo. En los casos en que el dispositivo no posea determinado feature se le asigna 0 si el feature tiene valores binarios (1, 0), y el valor "Otros" en los casos de variables con valores categóricos no dicotómicos. Aquí se aprecian 2 vistas que muestran cómo queda el dataset procesado antes y después de transformar a las variables en dummies respectivamente, para que puedan ser leídas por los algoritmos de machine learning.

device_idx	ls_pc	ls_mobile	ls_tablet	OS	age	fem	navegador	modelo	VOS	Antiguedad	marca
1.0	0	1	0	Android	5.0	1.0	SocialApp	Otros	Android 10	new	Otros
2.0	0	1	0	Android	6.0	0.0	Chrome	Otros	Otros	ancient	Samsung
3.0	0	1	0	Android	4.0	0.0	Chrome	Samsung SM-G610M	Android 7	ancient	Samsung
4.0	0	1	0	Android	5.0	0.0	Chrome	Otros	Android 10	new	Otros
5.0	1	0	0	Windows	5.0	0.0	Chrome	Otros	Windows 7	old	Otros

Luego se elimina al ID (device_idx), para generar un "get dummies" en las variables categóricas. Es decir, se deshace de los strings como valores de las variables, convirtiéndolos en columnas para tener 1 y 0 en toda la base, a nivel de dispositivo y poder así ser procesados por los modelos descritos.

ispc	istablet	ismobile	sistema_op_Linux	sistema_op_Mac OS X	sistema_op_Otros	sistema_op_Windows	sistema_op_iOS	version_sistema_op_Android 6
0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...
0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0

De esta manera, queda el dataset final de variables dependientes y la variable dependiente "fem" para ser procesados por los modelos de machine learning.

- **Dataset final para atributos de tipo "Dominio"**

Se muestra a continuación una vista con los rubros de interés generados a partir de datos de tipo "dom", siendo variables binarias donde 1 representa interés por dicho rubro por parte del dispositivo y 0 sin interés. Cada fila es un identificador de dispositivo.

domain_type_Inmobiliaria	domain_type_Noticias	domain_type_Otros	domain_type_Salud	domain_type_Tramites_Bancarios	domain_type_Viajes	fem
0	1	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0

- **Dataset final para atributos de tipo "Url"**

El dataset procesado para el caso de la los sitios web se hizo en base a los dominios desprendidos de cada uno, contabilizando la cantidad de veces que cada dispositivo los visitó. No es lo mismo que un usuario pase, por ejemplo, por bumeran 1 sola vez que 20 veces, demostrando en este último caso un fuerte interés en empleos.

starbucksrewards	supermercadosdia	swarovski	televisionlibre	toyota	tuenti	veadigital	veraz	vicentelopez	ypf	zonajobs	zonaprop	fem
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	24.0	20.0	1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	17.0	1
0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.0	0

Anexo 4: Hiperparámetros

ETAPA	Algoritmo de Machine Learning	Hiperparámetro	Rango de valores	Valor elegido por técnica de optimización de hiperparámetros
User Agent	Regresión Lineal	**penalty**	['l1','l2']	**l2**
		C	[0.0001, 0.001, 0.01, 0.1, 1, 5]	**0.001**
		solver	['liblinear', 'saga']	**'saga'**
	Random Forest	**n_estimators**	[int(x) for x in np.linspace(start = 50, stop = 200, num = 10)]	**150**
		max_features	['auto', 'sqrt']	**'sqrt'**

		max_depth	list(range(3,12), None)	**None**
		min_samples_split	[2, 5, 10]	**2**
		min_samples_leaf	list(range(1,15))	**9**
		bootstrap	[True, False]	**True**
	XGBoost	**n_estimators**	st.randint(50,200)	**62**
		learning_rate	st.uniform(0.05, 0.4)	**0.16860405745911938**
		max_depth	list(range(3,12), None)	**None**
		colsample_bytree	st.beta(10, 1)	**0.9113010087565172**
		subsample	st.beta(10, 1)	**0.893980755161291**
		gamma	st.uniform(0, 10)	**5.528199769079078**
		reg_alpha	st.uniform(0.05,10)	**0.8659418040024036**
		min_child_weight	st.uniform(1,20)	**17.202267893583617**

ETAPA	Algoritmo de Machine Learning	Hiperparámetro	Rango de valores	Valor elegido por técnica de optimización de hiperparámetros
User Agent + Dominios	Regresión Lineal	**penalty**	['l1','l2']	**'l1'**
		C	[0.0001, 0.001, 0.01, 0.1, 1, 5]	**0.1**
		solver	['liblinear', 'saga']	**'liblinear'**
	Random Forest	**n_estimators**	[int(x) for x in np.linspace(start = 50, stop =	**66**

			200, num = 10)]	
		max_features	['auto', 'sqrt']	**'auto'**
		max_depth	list(range(3,12))	**10**
		min_samples_split	[2, 5, 10]	**10**
		min_samples_leaf	list(range(1,15))	**9**
		bootstrap	[True, False]	**False**
	XGBoost	**n_estimators**	st.randint(50,200)	**177**
		learning_rate	st.uniform(0.05, 0.4)	**0.1405983100791751 7**
		max_depth	list(range(3,12), None)	**9**
		colsample_bytree	st.beta(10, 1)	**0.998326726997658**
		subsample	st.beta(10, 1)	**0.8790530564632983**
		gamma	st.uniform(0, 10)	**5.12093058299281**
		reg_alpha	st.uniform(0.05, 10)	**5.227513505274801**
		min_child_weight	st.uniform(1,20)	**4.487328580099829**

ETAPA	Algoritmo de Machine Learning	Hiperparámetro	Rango de valores	Valor elegido por técnica de optimización de hiperparámetros
User Agent + Dominios + Urls	Regresión Lineal	**penalty**	['l1','l2']	**'l1'**
		C	[0.0001, 0.001, 0.01, 0.1, 1, 5]	**0.01**
		solver	['liblinear',	**'saga'**

			'saga']	
	Random Forest	**n_estimators**	[int(x) for x in np.linspace(start = 50, stop = 200, num = 10)]	**100**
		max_features	['auto', 'sqrt']	**'sqrt'**
		max_depth	list(range(3,12), None)	**None**
		min_samples_split	[2, 5, 10]	**2**
		min_samples_leaf	list(range(1,15))	**9**
		bootstrap	[True, False]	**True**
	XGBoost	**n_estimators**	st.randint(50,200)	**177**
		learning_rate	st.uniform(0.05, 0.4)	**0.14059831007917517**
		max_depth	list(range(3,12, None))	**9**
		colsample_bytree	st.beta(10, 1)	**0.998326726997658**
		subsample	st.beta(10, 1)	**0.8790530564632983**
		gamma	st.uniform(0, 10)	**5.12093058299281**
		reg_alpha	st.uniform(0.05, 10)	**5.227513505274801**
		min_child_weight	st.uniform(1,20)	**4.487328580099829**

12. Bibliografía

[1] Gareth, J., Witten, D., Hastie, T., Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. New York: Springer.

[2] Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning. Zheng & Cassari.

Sebastopol (California): O'Reilly Media.

[3] Douglas B. Laney. (2017). infonomics: How to Monetize, Manage, and Measure Information as an Asset for Competitive Advantage.

[4] Gartner (blog). Why and How to Value Your Information as an Asset.

[5] Brian E. Martin Batista. (2020). Detección de URLs fraudulentas mediante técnicas de aprendizaje automático.

[6] La regresión logística [en línea]. Analytics Lane. <https://www.analyticslane.com/2018/07/23/la-regresion-logistica/>

[7] Naive Bayes Classifiers [en línea]. GeeksforGeeks. <https://www.geeksforgeeks.org/naive-bayes-classifiers/>

[8] Unverstanding Random Forest [en línea]. Towards data science. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

[9] Logistic Regression: Scikit Learn vs Statsmodels [en línea]. Stackexchange. <https://stats.stackexchange.com/questions/203740/logistic-regression-scikit-learn-vs-statsmodels>

[10] Alpaydin, E. (2020). Introduction to machine learning. MIT press.

[11] Scott, S., & Matwin, S. (1999, June). Feature engineering for text classification.

[12] Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. S. (2017, August). Learning Feature Engineering for Classification.

[13] Friedman, J. H. (2002). Stochastic gradient boosting. Computational statistics & data analysis.

[14] Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Eng. Bull.

[15] Suresh M., Usman A. (2020). Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data. Packt Publishing Ltd.

[16] Andriy Burkov (2019) . The Hundred-Page Machine Learning Book: sección 5.6

[17] Trevor Hastie, Robert Tibshirani, Jerome Friedman. (2001). The Elements of Statistical Learning, Data Mining, Inference, and Prediction. New York: Springer.

[18] Jimena R. Perez, Jimena D. Iglesias. (2018).Predicción de género para usuarios de Twitter.

[19] Leo Breiman. Random forests. Machine Learning, 45(1):5–32, Oct 2001

[20] William Koehrsen [en línea]. Random Forest Simple Explanation. <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>

[21] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. Predicting age and gender in online social networks. In Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents, SMUC '11, pages 37–44, New York, NY, USA, 2011. ACM.

[22] Delip Rao and David Yarowsky. Detecting latent user properties in social media. In Proc. of the NIPS MLSN Workshop, 2010.

[23] Explain any models with the SHAP values -- Use the KernelExplainer [en línea]. Towards data science. <https://towardsdatascience.com/explain-any-models-with-the-shap-values-use-the-kernelexplainer-79de9464897a>

[24] Machine Learning Model Explanation using Shapley Values [en línea]. Towards data science. <https://towardsdatascience.com/machine-learning-model-explanation-using-shapley-values-2b9eb0d1aaf1>

[25] Curvas ROC y Área bajo la curva (AUC) [en línea]. Aprendeia <https://aprendeia.com/curvas-roc-y-area-bajo-la-curva-auc-machine-learning/>

[26] Laney. "Why and How to Measure the Value of Your Information Assets", Gartner Research, November 15, 2016

[27] The match2One Blog [en línea] <https://www.match2one.com/blog/what-is-programmatic-advertising/>

[28] Global data, programmatic and display ad market 2017-2021 [en línea]. <https://www.onaudience.com/resources/top-data-markets/>

[29] Jasper Snoek, Hugo Larochelle and Ryan P. Adams (2012). Practical Bayesian Optimization of Machine Learning Algorithms. <https://arxiv.org/pdf/1206.2944.pdf>

[30] Aaron Lee (2020). Boruta Feature Selection (an Example in Python).
<https://towardsdatascience.com/simple-example-using-boruta-feature-selection-in-python-8b96925d5d7a>

[31] Samuele Mazzanti (2020). Boruta Explained Exactly How You Wished Someone Explained to You.
<https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a>

[32] Svideloc (2020). Target Encoding Vs. One-hot Encoding with Simple Examples.
<https://medium.com/analytics-vidhya/target-encoding-vs-one-hot-encoding-with-simple-examples-276a7e7b3e64>

[33] Max Halford (2018). Target Encoding done the right way.
<https://maxhalford.github.io/blog/target-encoding/>