PROYECTO FIN DE MASTER

MASTER EN DATA SCIENCE KSCHOOL

CALIDAD DEL USUARIO

DIEGO PÉREZ CAAMAÑO

ÍNDICE

1. Introducción

A mediados de 2017 Google Analytics lanzó una nueva métrica denominada calidad de sesión[1]. Esta nueva métrica utiliza técnicas de machine learning similares a Smart List y Smart Goals con el fin de poder evaluar cada sesión y conocer así la probabilidad que ha tenido esta de hacer una conversión.

Disponer de esta información es de gran utilidad ya que los usuarios con sesiones de mayor calidad y que no han terminado el proceso de compra, son mucho más susceptibles de llegar a convertir si les ayudamos a dar ese último paso. Por ejemplo, los usuarios que han estudiado los detalles del producto o añadido artículos a sus carros han dado claras señales de que están bastante interesados por estos productos. Un seguimiento persuasivo a través de una campaña de remarketing bien diseñada puede proporcionar el último empujón que necesitan para completar el proceso.

1.1. Objetivo del proyecto

Claramente esta nueva funcionalidad tiene un gran potencial para cualquier empresa que tenga procesos online (tengan el objetivo de vender o no). Pero, aunque los requisitos que impone Google analytics no son muy exigentes, ciertos sitios web no los cumplirán o las implementaciones necesarias se demorarán en el tiempo.

Por otro lado, esta nueva métrica pese a tener un gran potencial, está claramente limitada. La gran mayoría de las empresas dispone de varios canales a través de los cuales llegan a sus clientes y está métrica se calcula para cada uno de estos canales por separado sin tener en cuenta si el usuario ha accedido a otros canales y como ha interactuado en estos.

Analytics no nos lo pone de primeras fácil ya que no existe ninguna dimensión por defecto que nos permita diferenciar sesiones, ni tampoco usuarios. Para poder disponer de esta información debemos enviarla a analytics mediante una herramienta de terceros. Y aun disponiendo de esto, trabajar a nivel sesión es complicado (no imposible). Por ejemplo, si intentas hacer una consulta donde se muestre el número de sesiones que han llegado a una página concreta de tu web, la métrica sesiones no te sirve, para hacer esta consulta deberías usar número de páginas vistas únicas.

Por tanto, puesto que considero que lo realmente importante para un negocio es centrarse en el cliente, trabajaremos a nivel usuario e intentaremos aprender como podemos replicar esta nueva métrica de Google para así tener total libertad sobre ella.

1.2. Descripción de la calidad de sesión

Pese a que el objetivo es el usuario, es importante investigar sobre la nueva métrica para saber por dónde tenemos que empezar.

Una sesión es un concepto algo amplio, de modo que antes de meternos en materia, hay que hacer una pequeña definición de lo que Google analytics entiende por esta.

- Sesión: Conjunto de interacciones que tienen lugar en su sitio web en un periodo determinado, las cuales son realizadas por un usuario. Si pasados 30 minutos de inactividad (tiempo sin ninguna interacción), el usuario vuelve a interactuar con la web, estaremos en una nueva sesión.

Realmente Google lanzó dos nuevas métricas que funcionan de forma diferente. El **promedio de calidad de sesión**, calculado a partir de todas las sesiones para el periodo especificado. Y la **calidad de sesión**, calculada individualmente para cada sesión.

En la documentación oficial, se hace referencia a dos métodos de aprendizaje automático, en los que se basa el algoritmo usado por la empresa de la gran G. Los siguientes conceptos provienen de la ayuda de analytics que hace referencia a estos métodos:

- Smart Lists: Analytics aplica el aprendizaje automático a los datos de conversión para determinar qué usuarios tienen más probabilidades de realizar una conversión en sesiones posteriores. El aprendizaje automático usa varias señales, como la ubicación geográfica, el dispositivo, el navegador, la URL de referencia, la duración de la sesión y el número de páginas por visita para identificar a los usuarios de la audiencia.
- Smart Goals: Utiliza el aprendizaje automático para examinar docenas de señales sobre las sesiones del sitio web a fin de determinar las que tienen más probabilidades de realizar una conversión. A cada sesión se le asigna una

puntuación y las "mejores" se convierten en Objetivos Inteligentes. Estos son algunos ejemplos de las señales incluidas en el modelo de Objetivos Inteligentes: duración de la sesión, páginas por sesión, ubicación, dispositivo o navegador.

Estas definiciones ya nos dan pistas sobre qué datos deberemos extraer.

1.3. Origen de los datos

En este proyecto, toda la información que vamos a usar proviene de Google analytics, pero en principio cualquier herramienta de analítica web dispondrá de la misma información o muy similar. Por tanto, simplemente se deberán aplicar las consultas que realicemos a la herramienta de la que se disponga.

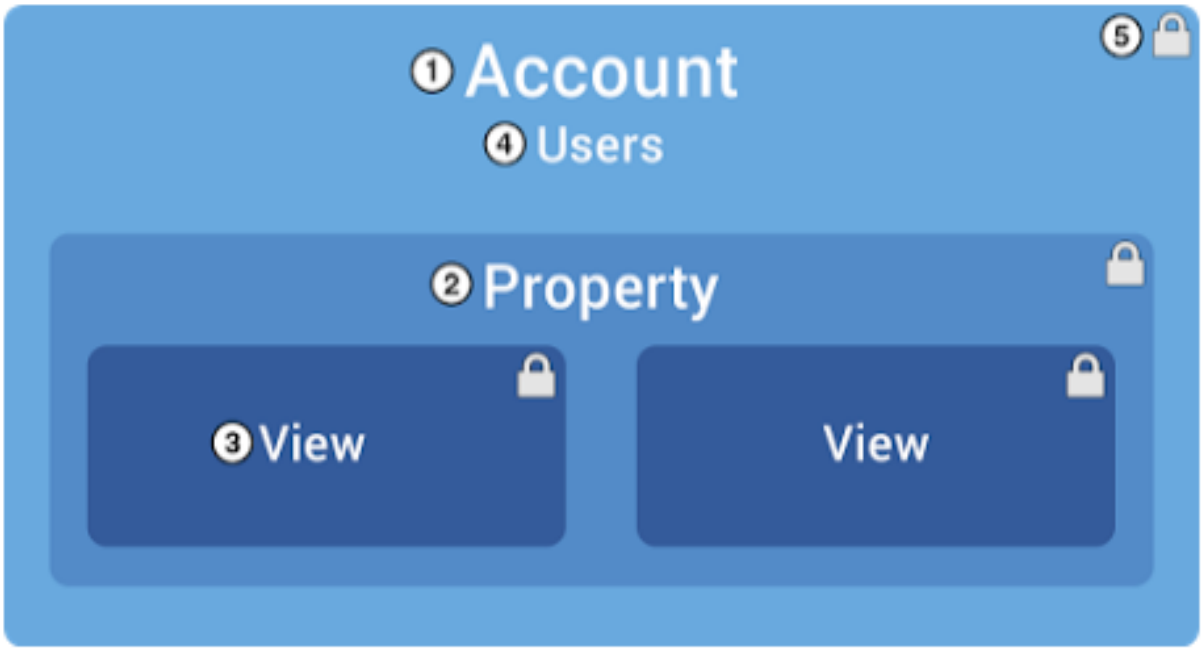
No obstante, si se dispone de algún otro tipo de información complementaria (edad, sexo, estado civil, etc.…) del usuario que aporte valor, es muy recomendable usarla.

2. Descripción de los datos de entrada

El objetivo de este apartado es detallar cuales y como debemos hacer las consultas en la herramienta de analítica web que usemos, detallar el formato de las variables que descarguemos y un diccionario de variables donde se explicará que es y por qué es importante esa variable.

2.1. Consultas a realizar

Lo primero que hay que tener claro es que cuenta, propiedad y vista vamos a utilizar en la herramienta de analítica.



1. Fuente: ayuda de Google analytics (jerarquía de cuentas, usuarios, propiedades y vistas)

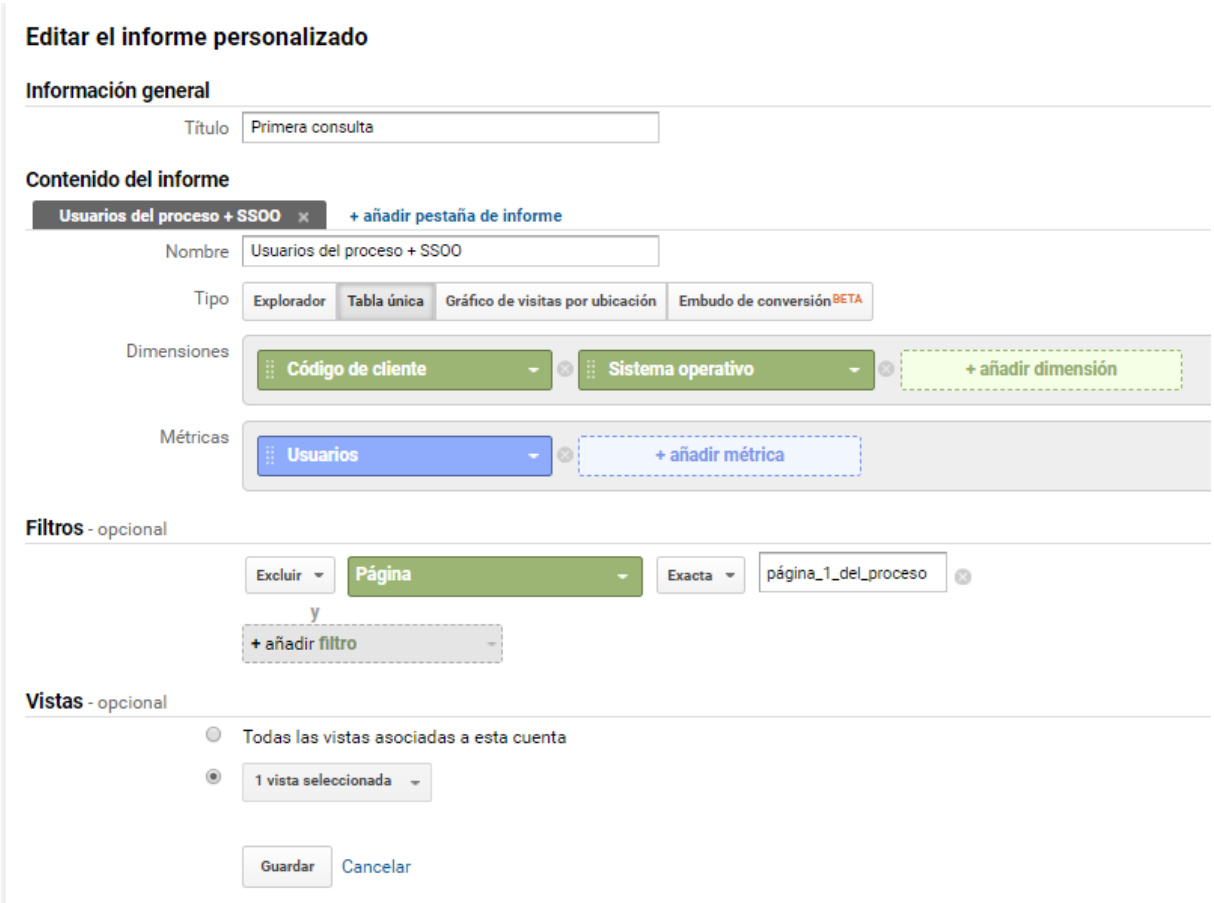
La **cuenta** es el punto de acceso a analytics y corresponde al nivel más alto de la organización.

La **propiedad** es un sitio web, una aplicación o un dispositivo (por ejemplo, un dispositivo en un punto de venta). Podemos tener varias propiedades en una misma cuenta.

Por último, una **vista** es el punto de acceso a los informes; se trata de una vista, que puede estar filtrada o no, de los datos de una propiedad. Podemos tener varias vistas en una misma cuenta.

Para este proyecto, se ha realizado todas las consultas directamente en Google analytics, usando sus informes personalizados. No obstante también se podrían hacer las consultas desde la API para Python (Google Analytics API Client Library for Python).

La primera consulta que haremos deberá estar dirigida a determinar que usuarios nos interesan. En este caso, nos centraremos en un proceso de compra concreto de un sitio web, por lo que deberemos filtrar por los usuarios que han accedido a esas páginas. Para ello, usaremos como dimensión el identificador de cliente del que dispongamos y la métrica usuarios. Añadiendo un filtro que contenga el nombre de la primera página de nuestro proceso conseguiremos quedarnos con los usuarios deseados. Para ahorrarnos una posterior consulta, añadimos la dimensión del SSOO que usa el usuario para acceder a nuestro proceso.



2. Consulta google analytics - Usuarios del proceso

Para finalizar la consulta, definiremos el periodo que nos interese y exportaremos los datos sin muestrear en el formato que deseemos (para este proyecto descargaremos los datos en formato .csv).

Con la siguiente consulta, extraeremos las páginas del proceso que ha visitado el usuario, el tiempo medio, número de visitas y número de visitas únicas de cada una de ellas. Por tanto, usaremos las siguientes dimensiones, métricas y filtros:

- Dimensiones: Identificador de cliente y página
- Métricas: Promedio de tiempo en la página, número de visitas a páginas y número de páginas vistas únicas.
- Filtros: Nombre de cada una de las páginas de nuestro proceso de compra. Para indicar a analytics que queremos varias páginas debemos usar el símbolo de la barra vertical “|”, es decir, page1|page2|page3… Este símbolo vendría a decir que queremos la page1 o la page2 o, etc.

Ahora que ya tenemos definidos los usuarios que nos interesan, las páginas que han visitado de nuestro proceso y el tiempo medio que han estado en ellas, vamos a extraer información general de usuarios. Hay que hacer la siguiente consulta:

- Dimensiones: Identificador de cliente
- Métricas: Sesiones, número de visitas a páginas, número de visitas a páginas únicas y duración media de la sesión.

Como última consulta, intentaremos averiguar los usuarios que han visto algún tipo de error en el proceso. Estos datos solo estarán disponibles si la web tiene etiquetada este tipo de información, por tanto, esta consulta es opcional.

- Dimensiones: Identificador de cliente, página y mensaje de error.
- Métricas: Número de páginas vistas.
- Filtros: Nombre de cada una de las páginas de nuestro proceso de compra y un segundo filtro que excluya las páginas con mensaje de error vacío.

2.2. Formato de los datos

Todos los datos han sido descargados en formato .csv y usando la opción de exportar datos sin muestrear. Por otro lado, la herramienta de analítica de Google generalmente nos proporciona unos datos bastante limpios, por lo que no habrá que hacer un gran trabajo de limpieza.

2.3. Diccionario de datos

A continuación, detallo el significado de cada una de las variables que contendrá el data frame. Cabe destacar, que todos los nombres de variables estarán escritos en minúscula, sin tildes y con barra baja como espacio.

Primero se expondrán las variables realizadas en las consultas y después las variables calculadas o que han tenido que ser tratadas para conseguir el dato deseado.

Variables consultadas

- **sesiones**
    - Tipo: Integer
    - Número de veces que un usuario ha accedido a nuestro sitio web.

- **vistas**
    - Tipo: Integer
    - Total de páginas vistas por el usuario en el periodo de tiempo seleccionado al realizar la consulta.

- **vistas_unicas**
    - Tipo: Integer
    - Páginas vistas únicas en cada sesión. Es decir, dos vistas de la misma página en la misma sesión contarán como una.

- **duracion_media_sesion**
    - Tipo: Integer

- Tiempo medio, expresado en segundos, que un usuario interactúa con nuestra web durante una sesión.

Variables calculadas o tratadas

- **identificador**
  - Tipo: String
  - Representa a cada usuario que ha entrado en nuestro proceso.

- **ssoo**
  - Tipo: Integer
  - Si el usuario ha accedido desde un dispositivo con Mac OS el valor será 0, con Windows el valor será 1 y si ha accedido con ambos el valor será 2.

- **profundidad**
  - Tipo: Integer
  - Puede adquirir los valores 1,2,3 o 4. Este valor dependerá de lo lejos que el usuario haya llegado en el proceso, es decir, 1 ha llegado solo a la primera página, 2 ha llegado a la segunda, 3 ha llegado a la tercera página y 4 ha terminado el proceso.

- **vistas_sesion**
  - Tipo: Float
  - Representa el número de páginas vistas que realiza el usuario de media en cada sesión y es el resultado de dividir las vistas entre las sesiones.

- **vistas_unicas_sesion**
  - Tipo: Float
  - Representa el número de páginas vistas únicas que realiza el usuario de media en cada sesión y es el resultado de dividir las vistas_unicas entre las sesiones.

- **avg_tiempo_pag1**
  - Tipo: Float
  - Tiempo medio que el usuario está en la primera página del proceso en cada sesión.

- **avg_tiempo_pag2**
  - Tipo: Float
  - Tiempo medio que el usuario está en la segunda página del proceso en cada sesión.

- **avg_tiempo_pag3**
  - Tipo: Float
  - Tiempo medio que el usuario está en la tercera página del proceso en cada sesión.

- **avg_tiempo_pag4**
  - Tipo: Float
  - Tiempo medio que el usuario está en la última página del proceso en cada sesión.

- **avg_tiempo_proceso**
  - Tipo: Float
  - Tiempo medio total que el usuario está en el proceso en cada sesión. Resultado de la suma de los tiempos medios en cada página del proceso.

- **avg_tiempo_no_proceso**
  - Tipo: Float
  - Tiempo medio total que el usuario está fuera del proceso en cada sesión. Resultado de la resta de la duración media de la sesión y el tiempo medio que el usuario está en el proceso.

- **vistas_pag1**

    - Tipo: Integer
    - Volumen de visualizaciones de la primera página del proceso.

- **vistas_pag2**
    - Tipo: Integer
    - Volumen de visualizaciones de la segunda página del proceso.

- **vistas_pag3**
    - Tipo: Integer
    - Volumen de visualizaciones de la tercera página del proceso.

- **vistas_pag4**
    - Tipo: Integer
    - Volumen de visualizaciones de la última página del proceso.

- **vistas_unicas_pag1**
    - Tipo: Integer
    - Volumen de visualizaciones únicas por sesión de la primera página del proceso.

- **vistas_unicas_pag2**
    - Tipo: Integer
    - Volumen de visualizaciones únicas por sesión de la segunda página del proceso.

- **vistas_unicas_pag3**
    - Tipo: Integer
    - Volumen de visualizaciones únicas por sesión de la tercera página del proceso.

- **vistas_unicas_pag4**
    - Tipo: Integer

    - Volumen de visualizaciones únicas por sesión de la última página del proceso.

- **user_error**
    - Tipo: Integer
    - Número de errores que un usuario ha visto en el proceso, debidos a un fallo del usuario.

- **process_error**
    - Tipo: Integer
    - Número de errores que un usuario ha visto en el proceso, debidos a un fallo en el proceso.

3. Metodología

Todo el trabajo realizado con las consultas y con el data frame final podemos resumirlo en 5 apartados, que se exponen a continuación.

3.1. Librerías

Para este proyecto se han usado las siguientes librerías

- Tratamiento de datos
  - Pandas
  - Numpy
- Visualización
  - Plotly
  - Matplotlib
  - Seaborn
- Normalización, reducción de dimensionalidad y machine learning
  - Sklearn

3.2. Procesado de datos

Google analytics nos proporciona datos bastante limpios y en un formato muy adecuado para un fácil tratamiento. Por ello, solo ha sido necesario usar pandas y numpy como librerías para el tratamiento de datos.

Básicamente se han ido uniendo variables a un data frame que contenía el identificador de los usuarios que habían entrado al proceso. Exceptuando algún tratamiento ligeramente más complejo para ciertas variables. Estas pequeñas manipulaciones se detallan en el notebook “TFM_QualityUser”.

3.3. Normalización

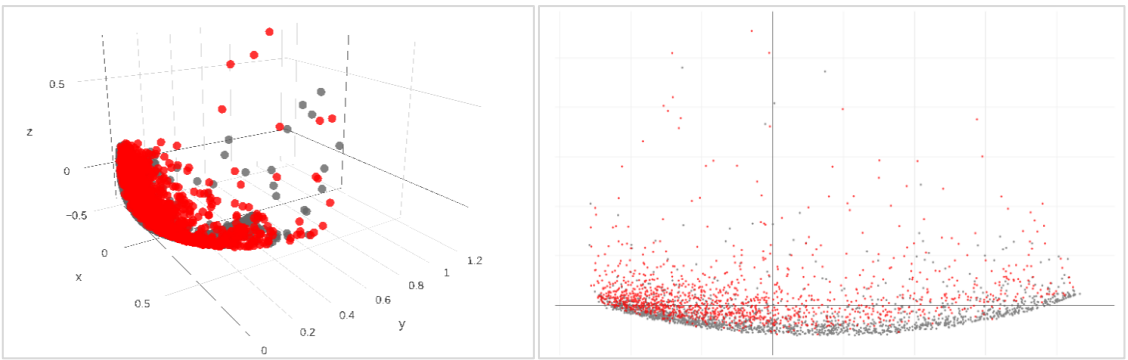
Las características de las variables difieren mucho unas de otras, por ejemplo, las variables de tiempo se expresan en segundos y pueden llegar a darse miles de segundos fácilmente. Mientras que las variables que expresan los errores que ha visto un usuario como mucho llegan a 30, por lo tanto, para evitar que las variables con extremos muy altos adquieran demasiado peso, es necesaria una normalización obligatoriamente.

Para ello se ha usado el paquete preprocessing de sklearn. En concreto se ha usado la clase normalizer**.**

3.4. Reducción de dimensionalidad y visualización

Con el objetivo de poder visualizar la forma que tienen los datos empleados, se ha recurrido a el algoritmo PCA (Principal component analysis) de reducción de dimensionalidad de la librería sklearn. Este tipo de algoritmos nos permiten reducir el número de nuestras variables a la cantidad que queramos. Muy resumidamente lo que hace esta técnica es una combinación lineal de las variables iniciales, consiguiendo sintetizar la mayor parte de la información contenida en los datos originales.

Esta técnica nos permite hacer visualizaciones de nuestros datos como las siguientes.



3. Gráficos realizados con plotly

Este tipo de herramientas nos permiten ver fácilmente como hay diferencias entre los usuarios que han comprado y los que no han comprado.

3.5. Machine learning

Finalmente, para conseguir tan preciada probabilidad de compra necesitamos hacer uso de un algoritmo de clasificación que nos permita predecir si un usuario compró o no compró. Una vez consigamos esto podremos extraer la probabilidad de que la predicción del algoritmo haya sido compró o no compró.

Durante la construcción de este proyecto se han probado varios algoritmos de aprendizaje supervisado y finalmente se decidió emplear un ramdon forest de la librería sklearn.

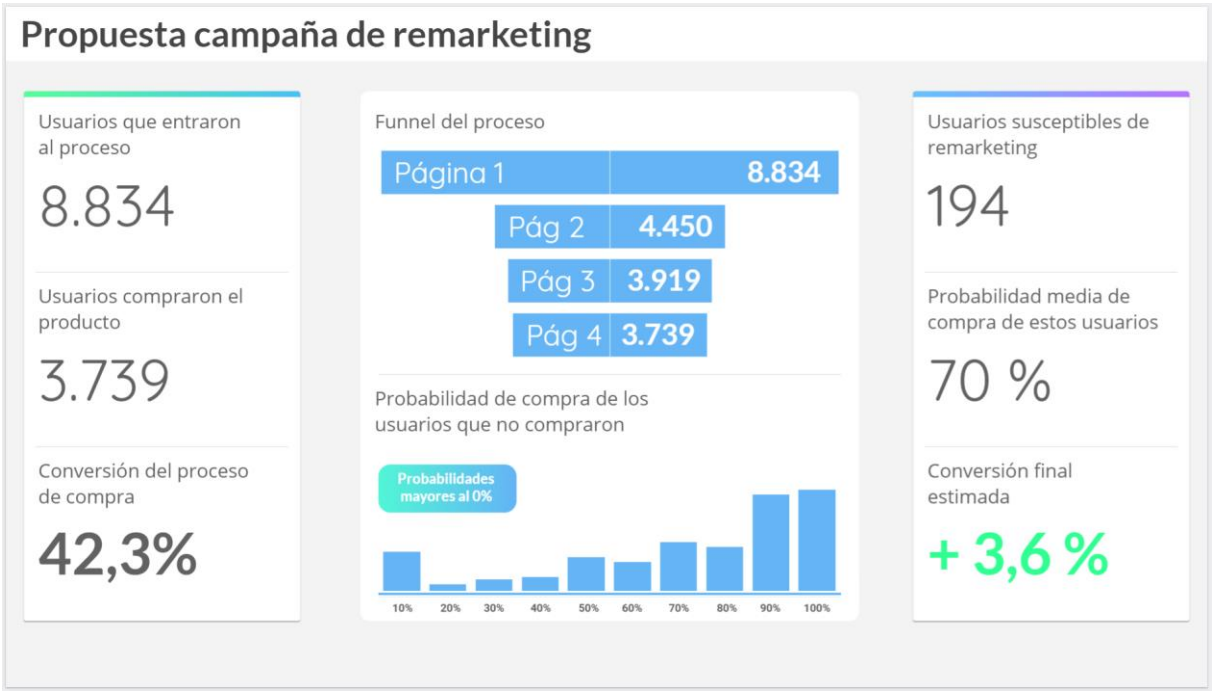
Se podría decir que ramdon forest es una integración de las técnicas *decisión tres, bagging* y ramdon subspace.

Resumidamente, se trata de una combinación en la que cada árbol depende de los valores de un vector aleatorio probado de forma independiente y con la misma distribución para todos ellos.

4. Cuadro de mando

El cuadro de mando elaborado en la herramienta Data Studio, ha sido realizado con el propósito de servir de ejemplo de presentación de los resultados obtenidos y explicación de los posibles beneficios de usar esta información en favor de cualquier empresa.

Enlace al cuadro de mando: https://datastudio.google.com/org//reporting/0B-Jj9ODXb6sKZWJtTEE2ZGNKZzA/page/QdRJ



5. Conclusiones

La dificultad de hacerse con la probabilidad de compra, no radica en todo el proceso del tratamiento de datos, ni en el algoritmo de machine learning usado. De hecho, la idea era bastante simple y la ejecución puedo afirmar a posteriori que no ha sido demasiado complicada. La dificultad de conseguir la probabilidad reside en las cualidades del proceso en el que nos encontremos y el fin para el que se quiera conseguir esta información. La clave, como casi siempre en el mundo de los datos, está en la limpieza de los mismos y en cual es el objetivo que se quiere conseguir.

Como se demuestra en el notebook final, se ha conseguido hacer una replica de la métrica que ofrece Google enfocándonos en el usuario. Una persona con una pequeña base de Python, un mínimo de conocimientos en manejo de herramientas como adobe analytics, webtrekk o Google analytics y espero que, leyendo este proyecto, puede ser capaz de conseguir esta útil información, gracias a todas las herramientas open source de las que disponemos hoy en día.

6. Referencias

- Calidad de la sesión, ayuda de analytics - https://support.google.com/analytics/answer/7303153?hl=es
- Concepto de sesión, ayuda analytics - https://support.google.com/analytics/answer/2731565?hl=es
- Smart List, ayuda de analytics - https://support.google.com/analytics/answer/4628577?hl=es
- Smart Goals, ayuda de analytics - https://support.google.com/analytics/answer/6153083?hl=es
- Informes personalizados, Google analytics - https://www.google.es/intl/es/analytics/features/custom-reports.html
- API Google analytics Python - https://developers.google.com/api-client-library/python/apis/analytics/v3
- Sklearn - http://scikit-learn.org/stable/
- Plotly - https://plot.ly/python/
- Matplotlib - https://matplotlib.org/
- Seaborn - https://seaborn.pydata.org/
- Pandas - http://pandas.pydata.org/
- Numpy - http://www.numpy.org/
- Data Studio - https://support.google.com/datastudio/?hl=es#topic=6267740
- Tableau - https://www.tableau.com/es-es/support/help