

Trabajo Práctico 2: Machine Learning

[75.06 / 95.58] Organización de Datos
Segundo cuatrimestre de 2018

Grupo Datatouille

Alumno	Padrón	Mail
Bojman, Camila	101055	camiboj@gmail.com
del Mazo, Federico	100029	delmazofederico@gmail.com
Hortas, Cecilia	100687	ceci.hortas@gmail.com
Souto, Rodrigo	97649	rnsoutob@gmail.com

<https://github.com/FdelMazo/7506-Datos/>

<https://kaggle.com/datatouille2018/>

Curso 01

- Argerich, Luis Argerich
- Golmar, Natalia
- Martinelli, Damina Ariel
- Ramos Mejia, Martín Gabriel

Contents

1	Introducción	1
2	Investigación	1
3	Feature engineering	2
3.1	Features básicos	2
3.2	Suma total de eventos	2
3.3	Cantidad de eventos por mes	3
3.4	Eventos sin contar mayo	4
3.5	Eventos en última semana	4
3.6	Distribución mensual de las conversiones	4
3.7	Informacion de los últimos eventos registrados por usuario	4
3.8	Precios de la ultima conversion realizada por el usuario	4
3.9	Porcentaje de la actividad de la ultima semana	5
3.10	Porcentaje de la actividad del ultimo mes	5
3.11	Días entre el último checkout y última actividad	5
3.12	Estados de celulares	5
3.13	Varianza logarítmica de productos vistos	6
3.14	¿El usuario compró más de la media?	6
3.15	¿Cuántas veces vio el último modelo que compró?	6
3.16	¿Cuántas veces vio la última marca que compró?	6
3.17	Comportamiento en sesiones de las últimas semanas	6
4	Algoritmos utilizados	6
5	Organización del Trabajo	6
6	Desarrollo	8
7	Resultados obtenidos	8
8	Conclusiones	8
9	Fuentes	8

1 Introducción

Se propone exponer en el siguiente informe el desarrollo del Trabajo Práctico para predecir la probabilidad de que un usuario de Trocafone realice una conversión en un período determinado de tiempo. Con dicho propósito se utilizan como base dos archivos csv brindados por la empresa. En un primer lugar el archivo `events_up_to_01062018.csv` que contiene la información de los eventos realizados por un conjunto de usuarios hasta el 31 de mayo de 2018. En un segundo lugar el archivo `labels_training_set.csv` que determina si un conjunto de usuarios realizó o no una conversión desde el 01/06/2018 hasta el 15/06/2018.

Se propone como objetivo desarrollar una métrica a fin de evaluar los distintos resultados obtenidos. Se presentan en el informe las distintas decisiones adoptadas para elegir un resultado por sobre otro. Así mismo, se propuso utilizar distintos algoritmos de la librería `sklearn` para elegir el que modelice mejor.

Por otro lado se crearon distintos csv a fin de poder desarrollar de mejor manera el proceso de feature engineering y obtener un pre-procesamiento de los datos que permita encontrar los resultados más altos.

2 Investigación

Como se mencionó en la *Introducción*, se realizó un proceso de exploración de los datos para crear nuevos atributos y extraer nuevos features. De esta manera se crearon distintos csv con información extraída tanto de internet como de los propios datos para luego concatenar al set de datos final. Se procede a dar una breve explicación de la funcionalidad de cada uno de ellos. Muchos de ellos se encuentran detallados en el 'Notebook Anexo' del TP1¹.

- `brands.csv`

Se agregó una columna al dataframe que detalla qué marca está involucrada en el evento del usuario.

- `os.csv`

Se agregó una columna al dataframe que detalla qué sistema operativo está involucrada en el evento del usuario.

- `browsers.csv`

Se agregó una columna al dataframe que detalla qué explorador de internet se accede al sitio.

- `sessions.csv`

Se agregó el concepto de sesión, que se define como la agrupación de una serie de eventos por usuario, los cuales están todos con menos de 30 minutos de inactividad entre el actual y el anterior. Esto fue fijado con un criterio arbitrario a fin de poder discretizar el tiempo y definir este concepto.

- `prices.csv`

¹<https://fdelmazo.github.io/7506-Datos/TP1/TP1.html>

Se agregó una columna al dataframe que indica el precio del producto involucrado en el evento del usuario. Para ello se extrajeron los precios de la página de Trocafone² considerando el sku, el modelo, el color, la capacidad de almacenamiento y la condición.

3 Feature engineering

A partir del nuevo dataframe obtenido con la unión de todos los csv descritos en el inciso anterior se procedió a la búsqueda de features. En esta etapa del desarrollo del Trabajo Práctico se buscó explotar las distintas ideas y después con un proceso de selección que será explicado más adelante elegir los features pertinentes y más útiles al modelo.

3.1 Features básicos

Se detallan los features generales considerados como pertinentes al modelo.

- `is_viewed_product`: el usuario vió un producto
- `is_checkout`: el usuario llegó a checkout con un producto
- `is_conversion`: el usuario compró un producto
- `session_checkout_first`: el usuario en su primera sesión realizó un checkout
- `session_conversion_first`: el usuario en su primera sesión realizó una conversión
- `session_ad_first`: el usuario en su primera sesión llegó con una campaña publicitaria
- `session_ad_checkout_event`: el usuario en su primera sesión llegó con una campaña publicitaria e hizo checkout
- `session_ad_conversion_event`: el usuario en su primera sesión llegó con una campaña publicitaria y compró el producto

3.2 Suma total de eventos

A los features agregados como *features básicas* se le calcula el total por usuario y se obtienen el siguiente listado de features:

- `total_viewed_products`: cantidad de productos que vio el usuario en el período de tiempo determinado.
- `total_checkouts`: cantidad de veces que el usuario hizo checkout en el período de tiempo determinado.
- `total_conversions`: cantidad de compras que realizó el usuario en el período de tiempo determinado.

²<https://www.trocafone.com/>

- `total_events`: cantidad de eventos totales que el usuario hizo en el período de tiempo determinado.
- `total_sessions`: cantidad total de sesiones del usuario
- `total_session_checkout`: cantidad total de sesiones donde el usuario hizo checkout
- `total_session_conversion`: cantidad total de sesiones donde el usuario convirtió.
- `total_events_ad_session`: cantidad total de sesiones donde el usuario ingresó por una campaña publicitaria.
- `total_ad_sessions`: cantidad total de sesiones donde el usuario ingresó por primera vez por una campaña publicitaria.

A partir de estos features se deducen los siguientes:

- `avg_events_per_session`: porcentaje de cantidad total de eventos sobre cantidad de sesiones
- `avg_events_per_ad_session`: porcentaje de cantidad total de eventos donde el usuario ingresó por una campaña publicitaria sobre cantidad total de sesiones donde el usuario ingresó por una campaña publicitaria
- `percentage_session_ad`: porcentaje de cantidad total de sesiones donde el usuario ingresó por primera vez por una campaña publicitaria sobre el total de sesiones
- `percentage_session_conversion`: porcentaje de cantidad total de sesiones donde el usuario ingresó por primera vez y compró sobre la cantidad total de sesiones

3.3 Cantidad de eventos por mes

Se agregan una serie de features relacionados a la cantidad de eventos y sesiones por mes que se consideraron pertinentes al modelo.

- `total_viewed_products_month`: cantidad de productos vistos por mes por usuario
- `total_checkouts_month`: cantidad de productos que llegaron a checkout por mes por usuario
- `total_conversions_month`: cantidad de productos que llegaron a ser comprados por mes por usuario
- `total_events_month`: cantidad de eventos por mes por usuario
- `total_sessions_month`: cantidad total de sesiones por mes
- `total_session_checkouts_month`: cantidad total de sesiones donde el usuario hace checkout por mes

- `total_session_conversions_month_`: cantidad total de sesiones donde el usuario compra un producto por mes
- `total_events_ad_session_month_`: cantidad total de sesiones donde el usuario ingresa a la página por una campaña publicitaria por mes
- `total_ad_sessions_month_`: cantidad total de sesiones donde el usuario ingresa a la página por primera vez por una campaña publicitaria por mes

3.4 Eventos sin contar mayo

3.5 Eventos en última semana

3.6 Distribución mensual de las conversiones

Se agrega en cuántos meses el usuario compró suponiendo que dicha distribución denota si el usuario es un comprador habitual o sólo compró alguna vez aisladamente. El feature se llama `"amount_of_months_that_have_bought"`.

3.7 Informacion de los últimos eventos registrados por usuario

Se busca extraer información de los días que transcurrieron hasta el último evento de un usuario. De esta manera se espera que el modelo aprenda un factor importante para la predicción. Por ejemplo, si un usuario vio un producto hace muchos días es muy probable que no lo compre pero si hizo checkout hace 1 día es probable que en un futuro cercano compre.

- `days_to_last_event`: cantidad de días hasta el último evento
- `days_to_last_checkout`: cantidad de días hasta el último checkout. Si el usuario no hizo checkout se considera un número mayor a la cantidad de días del período de tiempo comprendido.
- `days_to_last_conversion`: cantidad de días hasta la última compra del usuario. Si el usuario nunca compró se considera un número mayor a la cantidad de días del período de tiempo comprendido.
- `days_to_last_viewed_product`: cantidad de días hasta el último día que el usuario vio un producto. Si el usuario nunca vio un producto se considera un número mayor a la cantidad de días del período de tiempo comprendido.

En paralelo con estos features se consideran los días de la semana, del mes, del año y la semana del año donde ocurren estos últimos eventos.

3.8 Precios de la ultima conversion realizada por el usuario

Se consideró que podría considerarse el precio de la última conversión del usuario como un feature pero a la hora de la selección reflejó una importancia muy baja. Por lo tanto consideramos impertinente la descripción de la idea que habíamos pensado desarrollar.

3.9 Porcentaje de la actividad de la ultima semana

Aquí la idea pensada era reflejar la cantidad de eventos del usuario de la última semana sobre el total. Si el usuario ingresó muchas veces a la página en la última semana de mayo es muy probable que compre en la primera semana de junio. De la misma manera, si el usuario compró la última semana de mayo es probable que no compre por las siguientes dos.

Por lo tanto se pensaron los siguientes features:

- `percentage_last_week_activity`: porcentaje de la cantidad de eventos de esa semana sobre el total de eventos
- `percentage_last_week_conversions`: porcentaje de la cantidad de compras de esa semana sobre el total de eventos
- `percentage_last_week_checkouts`: porcentaje de la cantidad de checkouts de esa semana sobre el total de eventos
- `percentage_last_week_viewed_products`: porcentaje de la cantidad de productos vistos de esa semana sobre el total de eventos

3.10 Porcentaje de la actividad del ultimo mes

Una lógica análoga a la sección precedente se sigue en esta parte. Los motivos de este feature son simplemente una ampliación de la idea anterior. Si el usuario ingresó muchas veces a la página en mayo es muy probable que compre en la primera semana de junio. De la misma manera, si el usuario compró en mayo es algo probable que no compre por las siguientes dos.

De más está decir que se pensaron los siguientes features:

- `percentage_last_month_activity`: porcentaje de la cantidad de eventos de ese mes sobre el total de eventos
- `percentage_last_month_conversions`: porcentaje de la cantidad de compras de ese mes sobre el total de eventos
- `percentage_last_month_checkouts`: porcentaje de la cantidad de checkouts de ese mes sobre el total de eventos
- `percentage_last_month_viewed_products`: porcentaje de la cantidad de productos vistos de ese mes sobre el total de eventos

3.11 Días entre el último checkout y última actividad

La intención de este feature es medir la diferencia de días que tiene cada usuario entre la compra de un celular y la ultima vez que visualizo el producto comprado. De esta forma poder predecir en base a los productos vistos si es posible que se haga una compra.

3.12 Estados de celulares

Utilizando la lógica de que hay empresas que compran celulares en mal estado con el único fin de usar sus partes como respuestas se plantea agregar una columna que indique porcentaje de celulares en estado Bom - Sem Touch ID vs Bom sobre todos los celulares vistos.

3.13 Varianza logarítmica de productos vistos

Lo que se plantea es analizar la varianza en los precios de los productos visitados. Es decir, si los usuarios ven telefonos de un rango pequeño de precio o, por el contrario, artículos de precios muy variados. Se utilizó una escala logarítmica para seguir manteniendo las proporciones sin tener una gran diferencia entre la varianza de un usuario y la de otro.

3.14 ¿El usuario compró más de la media?

3.15 ¿Cuántas veces vio el último modelo que compró?

3.16 ¿Cuántas veces vio la última marca que compró?

3.17 Comportamiento en sesiones de las últimas semanas

4 Algoritmos utilizados

5 Organización del Trabajo

Para el desarrollo del Trabajo se utilizaron una serie de notebooks que buscaron organizar de la manera más clara posible los distintos pasos a realizar para desarrollar un submit. Se describen los distintos notebooks que fueron utilizados:

1. Investigación

En un primer lugar se quiso investigar si existió alguna relación relevante en el dataset brindado en el Trabajo Práctico 1 con los datos que se utilizarán en este.

Se recopiló la siguiente información:

- (a) No se repiten usuarios en los datasets.
- (b) En el primer dataset (TP1) hay 27624 usuarios de los cuales 13967 tuvieron actividad en junio. Entre el 1 y el 15 (inclusive) de junio 82 usuarios compraron productos.
- (c) En el segundo dataset hay 19414 usuarios de los cuales 980 compraron en Junio.

Por lo tanto se concluyó que hacer un merge de los datos del TP1 con los del TP2 presentaría un *skewness* en el set de datos, por la despreciabilidad de estos.

2. New dataframes

En este notebook se crearon los distintos csv que fueron realizados para la extracción de features, como fue mencionado anteriormente.

3. Feature engineering

En este notebook se agregan todos los features que se consideran que pueden ser pertinentes para el modelo. Para esto mismo se pensó agregar todos los features que se consideren que pueden ser útiles pero sabiendo que luego se somete a un proceso de selección.

4. Feature selection

En este notebook se utilizan distintas formas de seleccionar los features de manera de eliminar aquellos atributos que resulten ruidosos con el modelo de **Random Forest**. Se eligió dicho modelo por su popularidad para la selección de features ya que los árboles que crea el algoritmo toman distintos subconjuntos de atributos tomados al azar y arrojan distintos resultados. De esta manera, con cientos o miles de árboles el algoritmo adopta una amplia capacidad predictora de cada atributo.

Se utiliza como métrica el AUC debido a que es la utilizada por la plataforma de Kaggle para evaluar la eficiencia de los distintos modelos utilizados.

Las distintas formas de selección utilizadas se describen como sigue:

- **Cumulative importance**

El nombre del método fue inventado por el grupo de Trabajo y denota el método más intuitivo para la selección de features. Con el uso del algoritmo de **Random Forest** se parte de una lista de todos los features ordenados según importancia y se genera una lista de listas que agrega un feature a la vez. Por ejemplo siendo a,b,c features se parte de una lista como [a,b,c] y luego se obtiene [[a], [a,b], [a,b,c]]. El objetivo de este método es encontrar el *codo*, es decir, los features que incrementan el AUC local.

- **Forward Selection**

Con este método se comienza con ningún atributo y en cada paso se agrega el atributo que genere mejor resultado. Se agregan atributos siempre y cuando los resultados mejoren. El algoritmo termina cuando el resultado no se puede mejorar o cuando ya se han agregado todos los atributos.

- **Backward Selection**

Este método funciona a la inversa de **Forward Selection**. Se comienza con todos los atributos y se quita en cada iteración el atributo que aumente el resultado de la métrica. De esta manera el algoritmo termina cuando al quitar un atributo el resultado empeora o cuando ya no hay más atributos por quitar.

- **Stepwise Selection**

Este método es una variante que combina los dos métodos anteriores. En cada paso se considera agregar o quitar una variable de manera de aumentar el AUC local.

5. Submission framework

La principal idea de este notebook es definir una serie de funciones para armar las postulaciones de predicciones del trabajo práctico. Las mismas siguen los siguientes pasos:

- (a) Creación de la matriz **X** y el vector **y** para entrenar
- (b) Generación del split para obtener los sets de entrenamiento y de prueba

- (c) Ejecución del algoritmo de Machine Learning que devuelve un dataframe con `person` como índice y los `labels` como única columna.
- (d) Se obtienen las 3 medidas utilizadas como métrica para evaluar el rendimiento del algoritmo: precisión, auc y aucpr.
- (e) Se predicen las probabilidades
- (f) Se observa información relevante de la ejecución como la importancia de los features elegidos
- (g) Se guardan los resultados como csv para ser submiteados

6. Parameter tuning

El objetivo de este notebook es encontrar los hiperparámetros y los parámetros óptimos para cada algoritmo implementado. Para ello se utiliza el algoritmo de **Grid Search** que se encarga de probar todas las combinaciones de una lista de valores posibles para cada parámetro. Estos valores fueron investigados de diversas fuentes que serán mencionadas en la bibliografía debido a que es difícil fijar una frontera de valores sin una guía del área. Combinando las fuentes y el conocimiento de la funcionalidad de cada parámetro se llegó a los valores que se podrían considerar óptimos de los parámetros de los algoritmos.

6 Desarrollo

En el notebook principal llamado TP2 se realiza el desarrollo principal del Trabajo Práctico. A grandes rasgos en un primer lugar se utilizan los dataframes con todos los atributos seleccionados. Luego se definen y aplican los algoritmos de clasificación para realizar los entrenamientos y posteriores predicciones de conversiones. Finalmente se arman las postulaciones de labels.

La elección del algoritmo para realizar el *submit* se hace en base a todos los algoritmos y a combinaciones duales de ellos. Las combinaciones se realizan con el algoritmo **Voting Classifier** que es de la librería de **sklearn**. Para cada una de estas combinaciones se utiliza un set de features diferente a fin de elegir el que arroje mejor resultado. Dichos set de features se obtienen a partir de la selección que fue detallada previamente.

Finalmente, una vez elegido el algoritmo o el ensamble de algoritmos predilecto se entrenó con todo el dataframe (o mejor dicho **X train**) para enviar el submit.

7 Resultados obtenidos

Podríamos plotear kaggle? Tipo nuestro AUC local vs los distintos submits?

8 Conclusiones

9 Fuentes

- Parameter tuning para árboles de decisión:

<https://medium.com/@mohtedibf/indepth-parameter-tuning-for-decision-tree-6753118a03c3>

- Parameter tuning para random forest:
<https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e>
- Parameter tuning para xgboost:
<https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost->
- Parameter tuning para lightgbm:
<https://www.kaggle.com/sz8416/simple-bayesian-optimization-for-lightgbm>
<https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how>
- Parameter tuning para catboost:
<https://tech.yandex.com/catboost/doc/dg/concepts/parameter-tuning-docpage/>
- Parameter tuning para gradient boosting:
<https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient>
<https://medium.com/all-things-ai/in-depth-parameter-tuning-for-gradient-boosting-336>