Contents

1 Procesamiento y análisis de datos

En esta sección se introduce brevemente el producto a analizar y las herramientas que se utilizaron para realizar el análisis y la predicción requerida.

1.1 Datos utilizados

Se estudiaron datos provistos por la empresa Trocafone, analizando un conjunto de eventos de web analytics de usuarios que visitaron www.trocafone.com, su plataforma de e-commerce de Brasil.

1.2 Sobre la empresa

Trocafone es un side to side Marketplace para la compra y venta de dispositivos electrónicos que se encuentra actualmente operando en Brasil y Argentina.

La empresa realiza distintas actividades que van desde la implementación de plataformas de trade-in (conocidos en la Argentina como Plan Canje), logística directa y reversa, reparación y recertificación de dispositivos (refurbishing) y venta de productos recertificados por múltiples canales (ecommerce, marketplace y tiendas físicas).

Para conocer más de su modelo de negocio, pueden visitar el siguiente artículo:
https://medium.com/trocafone/el-maravilloso-mundo-de-trocafone-5bdc5761856b

1.3 Lenguajes y librerías utilizados

- Se utlizó como lenguaje de programación **Python3**.
- Para las visualizaciones, se utilizaron las librerías **MatPlotLib** y **Seaborn**.
- Como editor se utilizó **Jupyter Lab** (o **Jupyter Notebook**)
- Para el manejo de DataFrames, se eligió **Pandas** como librería a utilizar.
- Se utilizaron algunas herramientas como std o argsort de la librería **Numpy**. calendar

- Se importaron diferentes métricas como "accuracy score", "f1 score", "precision score", "recall score", "roc auc score" de la librería **sklearn**.
- Se utilizaron diferentes busquedas de hiperparametros, entre ellas "GridSearch" y "RandomizedSearchCV" de **sklearn**.
- Para poder realizar cross-validation, se importó "StratifiedKFold" de **sklearn**.
- Se realizó Clustering mediante la funcion "KMeans" de la librería **sklearn**.
- Los siguientes clasificadores fueron importados: "xgboost", "lightgbm", "RandomForestClassifier", "CatBoostClassifier".
- Se utilizo como ensamble de clasificadores a "VotingClassifier".

1.4 Repositorio de Github

Para el trabajo en conjunto del equipo, se utilizo un repositorio en github, donde se encuentran todos los archivos necesarios del análisis y predicciones y este informe propiamente dicho.

Link: https://github.com/emabrea/7506-DATOS-TP2.git

2 Breve análisis del set de Datos

3 Featuring Engineering segunda entrega (Train 100%)

3.1 Features sobre acciones por rango de tiempo

3.1.1 Acciones en el último mes

- Visitas último mes
- Checkouts último mes
- Compras último mes
- Subscripciones último mes

3.1.2 Acciones en los últimos 15 días

- Visitas últimos 15

3.1.3 Acciones en la última semana

- Visitas última semana
- Checkouts última semana
- Compras última semana
- Campaña ultima semana

3.1.4 Acciones en los últimos 3 días

- Visitas últimos 3

3.2 Features de acciones del usuario

- Total visitas usuario
- Total checkout
- Total compras
- Búsqueda celular
- Días distintos
- última visita

3.3 Features de modelos

- Modelos distintos vistos

4 Clasificadores

5 Tuning

Grid Search y Random Search en KNN

En los diferentes algoritmos vamos a llamar parámetros a aquellos valores que el algo- ritmo encuentra a partir de los datos y vamos a llamar hiper-parámetros a aquellos datos que el algoritmo necesita para poder funcionar.

Llamaremos óptimos a los hiper-parámetros que logren para un set de Datos maximizar determinada métrica, en este caso, utilizamos la métrica ROC AUC.

Para encontrar los hiper-parámetros óptimos para un algoritmo pueden usarse dos métodos: Grid-Search o Random-Search.

5.1 Grid-Search

En un Grid-Search probamos todas las combinaciones posibles dentro de una lista de valores posibles para cada hiper-parámetro.

En este caso, debido a la cantidad de hiper-parámetros , se decidió comenzar en las listas de valores posibles para cada uno con un "paso" grueso.

Esto reduce en un principio la cantidad de combinaciones a ejecutar. Luego se podra refinar la busqueda en la zona donde resultaron optimos nuestros hiper-parametros iniciales.

Este proceso fue especialmente necesario cuando los hiper-parámetros tomaban valores reales.

Cuando la cantidad de hiper-parámetros es realmente muy grande la combinatoria a realizar puede resultar muy ineficiente, en estos casos puede recurrirse al método de Random-Search.

5.2 Random-Search

Como la cantidad de hiperparametros era elevada, posteriormente se recurrió al método de Random-Search,

En este método controlamos cuantas iteraciones realizamos de nuestro algoritmo y por cada iteración seleccionamos los valores de los hiper-parámetros al azar dentro de un rango preestablecido.

Cabe aclarar que este método no es tan preciso como un grid-search pero es mucho mas rápido, pudiendo invertir mayor parte del tiempo a la creación de Features.

5.3 Aplicación a nuestro algoritmo

En uno de nuestros primeros ensambles (constituídos por un Random Forest y un XGBoost) se aplico el método de Random Search con el fin de poder encontrar (u obtener una buena aproximación a ellos) de los hiperparametros:

Para XGBoost:

- Peso de

Para Random Forest:

- Peso de una de las clase 1 : La lista de este hiperparametro contenía los números del 1 al 10
- Criterio: Este hiper-parámetro podía ser : {'gini', 'entropy'}

Es muy importante aclarar que para la validación de hiper-parámetros se utilizó K-fold cross validation, con K= 10.

Es decir, de nuestro set de datos, el 10% fue usado para validar los hiper-parametros cada vez.

6 Ensamble

Los mejores algoritmos de ML suelen surgir de la combinación de varios algoritmos. Es muy raro que un solo algoritmo de ML logre mejores resultados que un ensamble. Esto pudo verse en las sucesivas entregas, donde utilizar un ensamble nos traía resultados mas satisfactorios que usar los algoritmos en soledad.

6.1 Combinando Algoritmos Diferentes

Estos procesos suelen ser la clave para obtener un mejor resultado, ya que permitieron aprovechar el poder expresivo de varios modelos completamente diferentes para obtener un resultado común. En nuestro caso, el problema a abordar era de clasificación.

6.1.1 Majority Voting

Tenemos varios clasificadores distintos para un cierto problema, cada uno de ellos produce un resultado y queremos obtener un resultado final. Una aproximación simple es ver cual es la clase que tiene mayoría entre todos los clasificadores.

En un primer momento, se decidió utilizar este tipo de ensamble debido a su simplicidad, pero nos encontramos que el mismo tiene sentido cuando la predicción es directamente la clase. Si la predicción es la probabilidad de cada clase entonces obtuvimos que otros métodos funcionan mejor.

Cuando la correlación entre los modelos es baja el resultado del ensamble, en general, mejora notablemente el resultado de cada modelo individual.

Una primera conclusión es que dados muchos clasificadores es conveniente elegir un conjunto que tenga buenos resultados y que estén muy poco correlacionados.

6.1.2 Averaging

Promediar el resultado de varios clasificadores es un método muy popular que funciona en muchos problemas distintos: regresión, clasificación (ya sea para predecir clases o probabilidades), etc. La idea principal es reducir el overfitting.

En general, una separación suave entre las clases es mejor que una separación muy irregular y promediar clasificadores logra esto.

Cuando promediamos clasificadores que predicen la probabilidad de las clases, hubo que prestar especial atención porque cada clasificador individual puede tener una calibración completamente diferente.

Uno puede dar probabilidades muy cercanas a 1s y 0s mientras que otro, a lo mejor, se mantiene dentro de un cierto rango. Una solución para esto es convertir cada probabilidad de un rango entre 1 y n, siendo n el total de personas a predecir. La persona con mayor probabilidad tiene 1, el segundo 2, etc, y el de menor probabilidad, n, sin importar el valor de las probabilidades. Si hacemos esto para todos los clasificadores podemos luego promediar los rangos y convertir estos promedios en un número entre 0 y 1 para la probabilidad final.

7 Conclusiones generales

Al finalizar el presente trabajo, nos propusimos destacar ciertos aspectos del mismo, de tal modo que permitan a la empresa tener una visión más clara de su progreso.

Los principales puntos a tratar son:

- Las ventas de celulares vienen aumentando con los meses, pero lentamente, aproximadamente 50 ventas más por mes.
- Las campañas y avisos publicitarios deben publicarse durante la semana, y no en el fin de semana. Además, dichas campañas deben mostrarse a partir de las 16:00, recordando que la mayor actividad ocurre a la noche.
- El día martes es un buen candidato para publicitar, tal vez porque los lunes son muy cargados de trabajo, y recién el martes uno tiene más libertad.
- Continuar con la campaña de subscripciones, luego de realizar una compra, pues en el último mes viene dando resultados.
- Si bien la marca más comprada es Samsung, deberían profundizar en la venta de iPhone, que son los más visitados, pues están de moda por ser considerados de alta gama. Esto no implica no seguir con los modelos de Samsung, simplemente abrir el panorama.
- Si bien los celulares de 16gb son los más vendidos, prestar atención con los de 32gb, que podrían ser los próximos protagonistas. A aquellos con 8gb, no brindar tanto soporte como antes.
- La condición "Bueno" sigue siendo la más visitada y comprada, tener esto en cuenta. Cada mes más usuarios compraron a partir de una campaña, pero como se vió en el análisis, la unica con resultados significativos es la de Google. Por lo tanto, creemos que que debería continuarse con esta modalidad, pero habría que evaluar la relación costo beneficio de las otras campañas publicitarias.
- Dar soporte especializado al navegador Google Chrome, tanto en su versión de escritorio como en su versión mobile, los otros no son tan significativos a nivel cantidad de usuarios.

- Por su parte, las tablets están en desuso, por lo tanto, enfocarse en los smartphones principalmente, que van a superar las computadoras de escritorio pronto, en términos de uso.
- Si bien la mayoría de los usuarios son de Brasil, se noto una creciente actividad en los Estados Unidos. Por lo tanto, una propuesta es traducir la página web al inglés, para promover la visita de personas de dicho país. También notamos un posible negocio futuro en Argentina, donde se debe hacer hincapié en la diferencia de precio con respecto a otros vendedores.
- Por último, se debe volver a hacer un análisis de datos en unos meses, y verificar el progreso de la empresa en base a las recomendaciones propuestas, para determinar el correcto camino.

References

[1] Trocafone website, www.trocafone.com.

[2] NumPy - NumPy, http://www.numpy.org/.

[3] Python Data Analysis Library, https://pandas.pydata.org/.

[4] Matplotlib: Python plotting — Matplotlib 3.0.0 documentation, matplotlib.org.

[5] seaborn: statistical data visualization — seaborn 0.9.0 documentation, seaborn.pydata.org

[6] Folium information,
https://github.com/python-visualization/folium

[7] GeoPy's documentation,
https://geopy.readthedocs.io/en/stable/