UNIVERSIDAD AUSTRAL | INGENIERÍA

T.....1

Casuistica

Maestría en Explotación de Datos y Gestión del Conocimiento

Franco Lianza, Alexis Walker

23 December 2021

Índice

1 Introducción

blah blah

2 DataOps y MLOps: Cómo escalar el delivery de productos de datos sin morir en el intento

Ciclo de vida de todo proyecto de datos

Deuda técnica o metodológica, modelos que no llegan a producción

2008 - concepto de devops (entrega continuo) 2014 - reconciendo el valor de los datos, éstos como activos: dataops -> el universo no solo es el código sino que se suman los datos. 2015 - paper de google -> MLops: codigo datos y modelo.

2.1 Racionales

Las soluciones originadas a partir de los datos se tornaron críticas al estar cada vez más enbedidas en los cores. Tambien en parte por la alfabetizacion de las áreas usuarias, implicando mayor y más compleja demanda.

Problemas - una solucion implica nuevas necesidades - datos no integrados, no disponibles.. enumerar... - la naturaleza cambiante del producto de datos, siempre hay algo para mejorar

Se buscar cambiar la forma de trabajo del ususairo.

Soluciones - heroicas: laburando mucho, pero como el producto no termina nunca, vivimos sobre demandados - esperanza: todo funciona bien del vamos, . - sobrestimar: se sobre estima el trabajo para tener espacio a modo de cubrirse

Credibilidad en baja: - 51 - 87 - 80

2.2 Conceptos heredados de la industria del software

agile + devops + control estadistico de procesos

2.3 Ejemplo

desarrollo, gestion y gobierno

MLflow

3 Que hotel va primero para vos? Un ejemplo del desafio de usar Machine Learning para odernar hoteles en Despegar.com

Despegar.com se encontraba ante la incógnita de si era posible ordenar los hoteles que se mostraban a los usuarios de manera basado en la probabilidad de compra en tiempo real de manera de optimizar la conversión. La idea era generar un sistema que ante una búsqueda del usuario clasifique y ordene los resultados obteniendo algún tipo de score que pueda ser interpretado como probabilidad de compra.

3.1 Sorting

El sorting es un tipo de problema que consiste en la aplicación de algoritmos de machine learning cuyo objetivo es optimizar la ubicación de los items de los mas relevantes a los menos relevantes. Para esto, existen 3 enfoques distintos:

- ***Pointwise***: el modelo se aplica sobre cada ítem de manera individual, es decir se utilizan las características propias del ítem sin mirar a las del resto. El ordenamiento se da por el score individual de cada ítem.
- ***Pairwise***: el modelo se aplica sobre cada par de ítem, de forma que se comparan las características de cada uno a fin de obtener el mejor de ambos. El ordenamiento final es relativo a los scores de cada par.
- ***Listwise***: el modelo se aplica sobre la lista entera de items y busca optimizar alguna métrica de ordenamiento como Ganancia *Acumulada Descontada Normalizada (NDCG)* o la *Precisión Media Promedio (MAP)*.

Solo el ordenamiento *pointwise* provee scores que pueden ser interpretables como probabilidades, por lo que el proyecto utilizo este enfoque por sobre los otros.

Otro de los problemas encontrados durante la investigación sobre el sorting es el *sesgo de la posición*. En los análisis exploratorios realizados se encontró que la probabilidad de saber si un ítem es relevante para un usuario-contexto disminuye con la posición. Este sesgo complica el trabajo de entender cuanto influye las características del hotel y cuanto influye la posición en la que se muestra.

3.2 Diseno

El diseno del producto fue separado en 4 etapas: estimación del sesgo de la posición de los datos actuales, entrenamiento del modelo (feature engineering y modelado), post procesamiento de scores y evaluación offline.

3.2.1 Estimacion del sesgo

La posición de un ítem influye en la probabilidad de compra, es decir, es mas probable comprar items que aparezcan en los primeros resultados. Este sesgo debe ser medido de alguna forma para poder incorporarlo al modelo. El equipo de *Despegar.com* utilizaron el *algoritmo Expectation-Maximization* obteniendo un **ponderador** para utilizar en el modelo. Este ponderador puede ser interpretado como *la probabilidad de que el usuario haya visto el ítem.*

3.2.2 Entrenamiento del modelo

Los datos que se utilizaron para el armado del dataset consistían en características propias del contexto, del hotel, del contexto y hotel y finalmente, el target binario de si la persona compraba el hotel o no. La figura 1 muestra algunos ejemplos utilizados.

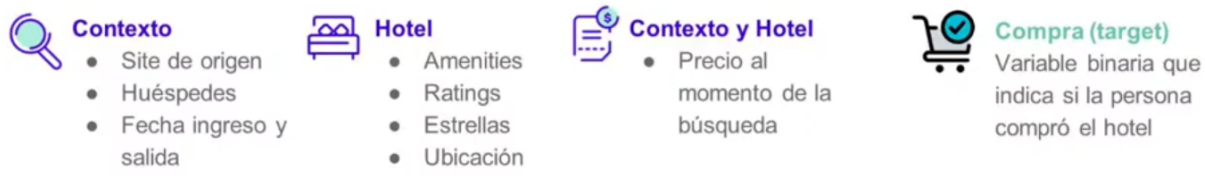

Fig. 1: Features utilizados

Algunas de estas features pueden ser obtenidas en tiempo real y otras tuvieron que pre-computarlas y dejarlas en un *feature store* para que se pudiesen consumir de manera online.

Para el modelado, se utilizo un *XGBoost por cada destino* por varias razones: por su potencia predictiva y por su portabilidad. El sistema transaccional de Despegar.com se encuentra programado en Scala, por lo que la elección de XGBoost les permitió programarlos en Python y luego ponerlos productivos en Scala.

La evaluación del modelado consitio en dos momentos: offline y online. La evaluación offline fue medida con métricas de clasificación (F-score) y métricas de probabilidad (Negative Log Loss). La online, mediante 3 KPIs definidos por el negocio: *tasa de conversión*, *margen por usuario* e *ingreso*. A su vez, el modelo fue deployado utilizando un esquema A/B para algunos de los destinos seleccionados.

3.2.3 Post procesamiento de scores

Luego del entrenamiento, los scores obtenidos fueron post procesados en función del enfoque.

En el *enfoque de clasificación*, se definió un umbral por el cual los hoteles eran clasificados en "relevantes" y "no relevantes". De esta manera se optimizaba de manera eficiente el f-score.

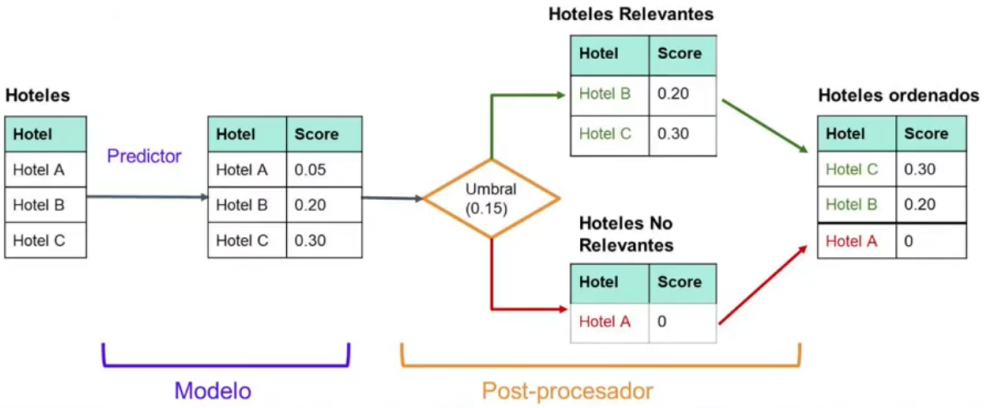

Fig. 2: Enfoque de Clasificacion

En el *enfoque de probabilidades*, se optimiza la métrica negative log loss (NLL) para medir diferencias entre el score (z) y el target de compra o no compra (y).

$$NLL = \sum_{i=-1}^{+N} y_i + \log z_i + (1 - y_i) \log(1 - z_i) \tag{1}$$

A su vez, se utilizaron *gráficos de calibración*. Estos gráficos permiten observar la relación de los scores medidos y la tasa de conversión. La idea general de esta herramienta es agrupar hoteles de manera similar y graficar las medidas. Resulto de gran importancia durante el desarrollo ya que se puede compartir con otras áreas de Despegar.com ya que son simples de interpretar: siempre y cuando se muestre una tendencia positiva, el modelo funciona correctamente.

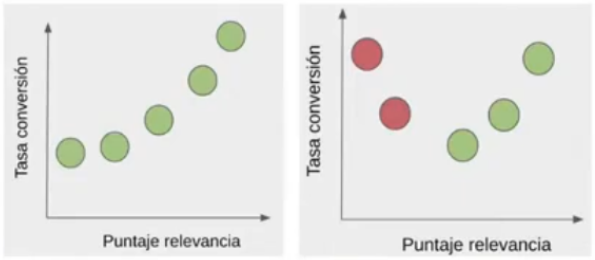

Fig. 3: Grafico de Calibración.

3.2.4 Evaluación

El equipo decidió realizar una POC con el enfoque de *clasificación* obteniendo resultados mixtos en los KPIs. En la siguiente iteración, se cambio al enfoque de *probabilidades*.

El enfoque de *probabilidades* resultó ser el mejor. La Figura 4 muestra los gráficos de calibración obtenidos por enfoque.

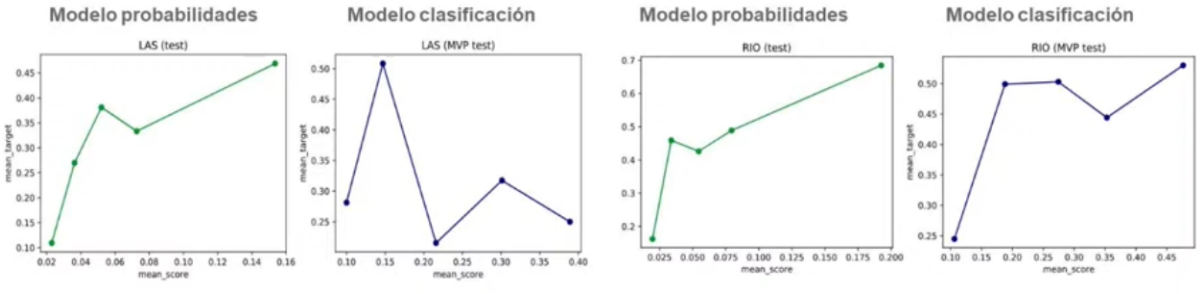

Fig. 4: Graficos de calibración obtenidos por enfoque.

En la Figura 5 se detalla el comportamiento de cada KPI en función de su enfoque.

Modelo\KPIs	Δ Mg/User	Δ GB	Δ CVR
POC	-	=	+
V1	+	+	+

Fig. 5: Resultados obtenidos por enfoque.