UIMP
Universidad Internacional Menéndez Pelayo

AEPIA
Spanish Association for Artificial Intelligence

Hortonworks Partner
ACADEMIC TRAINING

Pautas generales para las prácticas:

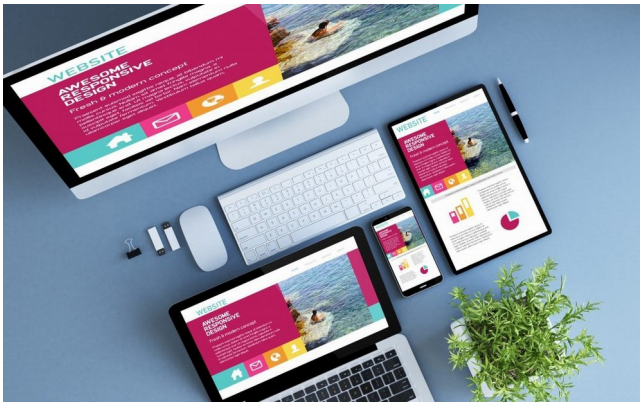
1. Las prácticas **deberán acompañarse de un breve documento README** que detalle cómo ejecutar la práctica y cómo configurar los parámetros si existen, así como cualquier otra particularidad que sea necesario conocer. Adicionalmente, si se desea explicar algún detalle de implementación, podrá adjuntarse opcionalmente un documento de diseño en el que se presente y justifique el diseño elegido.

2. El código debe seguir las **buenas prácticas** habituales: variables con nombres significativos, comentarios, eficiencia, etc.

Ejercicio de evaluación: Despliegue de modelos de Machine Learning y A/B testing

Descripción del proyecto

La empresa de venta online **silocompro.com** ha estado trabajando en la distribución de una nueva categoría de productos. Después de distribuirlos en sus tiendas físicas ha llegado el momento de distribuirlos también a través de su tienda online. En el departamento de tecnología es conocido que el canal de venta está saturado y se está mejorando la visualización. Sin embargo, la distribución de esta categoría de productos es una prioridad. Como solución temporal, se ha decidido que en la web se abra un espacio para sugerir productos de esta categoría. Pero esto debe ser solo mostrado a aquellos usuarios que tengan una predisposición a comprar estos productos, dado que se sospecha que introducir este cambio tendrá un efecto negativo en aquellos usuarios que no estén interesados, dado que para mostrar esta nueva categoría, incurrimos en el coste de oportunidad de no mostrar las categorías tradicionales de la empresa, afectando la oportunidad de conversiones de venta.



Estrategia

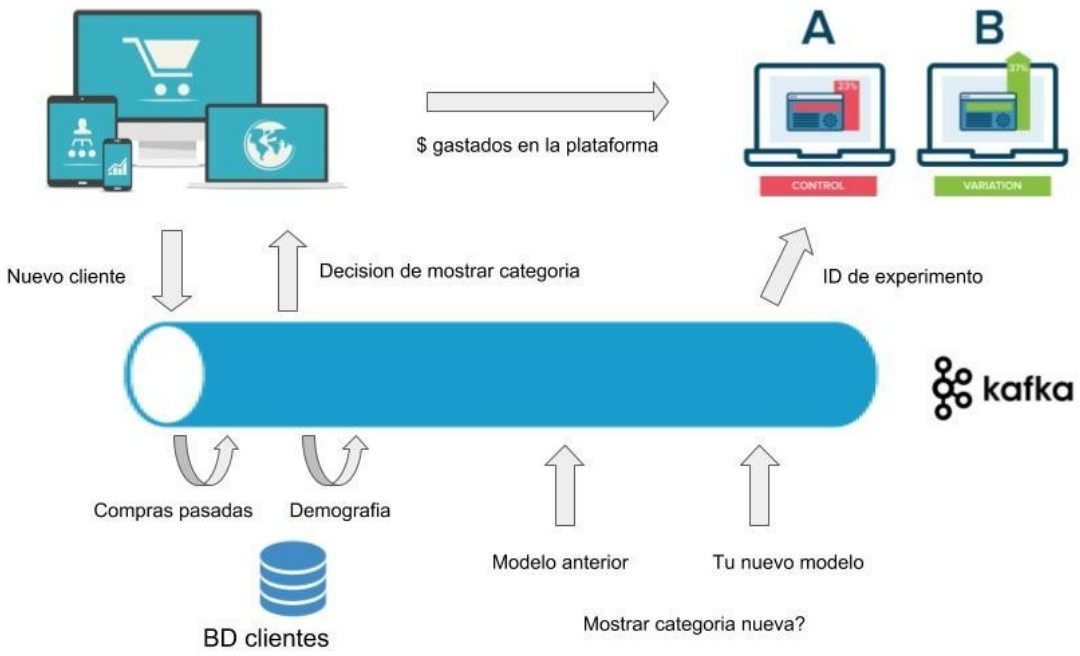
Para construir una solución de manera segura, se han recogido datos sobre aquellos clientes que han comprado el producto en tienda física y que están registrados en la tienda online. En una primera aproximación, se observó que hay una cierta tendencia a la compra en esta categoría dependiendo del sexo y edad. Este sistema se ha desplegado con cierto éxito, pero se sospecha que no es ideal ya que los ingresos medios han descendido 0.3 $/cliente después del despliegue, por lo que se ha hecho un rollback de este. Hay que tener en cuenta que los ingresos por cliente tienen una desviación típica de 15$.

Tu misión

Como científico de datos se te pide que demuestres una mejora de 0.7 $/cliente en ingresos con respecto al sistema original. Para ello cuentas con una nueva fuente de datos a mayores de la edad y sexo: el histórico de compra en otras categorías de los clientes en tienda. Se sospecha que existe un cross-selling entre ciertas categorías y la nueva. Para demostrar esta mejora, se te permite hacer un A/B testing entre la solución original y tu solución en una muestra de clientes.

Arquitectura

silocompro.com cuenta con una plataforma de A/B testing básica que se detalla en la siguiente figura



Se tomará una muestra de clientes en tiempo real. Para estos clientes, cuando entren a la plataforma se introduce un mensaje en kafka correspondiente a esa sesión de usuario con el id de cliente de este. Este topic de kafka es leído por la plataforma para enriquecerlo con la información demográfica y de histórico de compras pasadas, que son procesados por pipelines diferentes y escritos en otros dos topics de kafka separados.

El modelo original y los alternativos son asignados a muestras de usuarios separadas, que tienen que decidir si mostrar la categoría nueva al usuario o no. Una vez esta decisión es tomada, la web reacciona de manera automática ejecutando la decisión. Al terminar la sesión de usuario, la cantidad de dinero gastada por el usuario en la plataforma se registra, junto con la decisión e ID de sistema original. Una vez llegado al número de muestras correspondiente al test estadístico configurado en la plataforma, esta informa si el nuevo sistema mejora al anterior con el efecto deseado. Recordamos que negocio ha puesto como objetivo mejorar el sistema base en 0.7 $/ cliente de media.

Materiales

Para el desarrollo del proyecto, se te proporcionan los siguientes materiales:

- Un **conjunto de entrenamiento** sacado de los clientes en tienda con
  - histórico de compras pasadas
  - información de edad y sexo
  - si compro un producto de la nueva categoría o no
- **Acceso al sistema de A/B testing** que incluye
  - Web de configuración del experimento (http://bigdatamaster2019.dataspartan.com/). Una vez configurado un experimento en la web, esta proporciona un token para poder enviar predicciones al sistema final.
  - Acceso a las colas de kafka para lectura
    - **topic_demographic**: edad y sexo del cliente
    - **topic_historic**: indicadores de compras históricas del cliente en las diferentes categorías
  - ... y escritura
    - **topic_student_prediction**: topic en el que tu nuevo sistema determina si se muestra la nueva categoría o no.
  - La dirección de conexión a la cola de kafka está abierta en
    - bigdatamaster2019.dataspartan.com:19093
    - bigdatamaster2019.dataspartan.com:29093
  - Toda la información se lee/envía a kafka en json codificado en texto plano. Es responsabilidad del sistema final el parsear esta información y hacer join de los dos topics de entrada. El campo uuid corresponde al id de cliente ha de ser incluido en todos los mensajes, tanto de entrada como de salida del pipeline de predicción. Un ejemplo de formato de predicción en el topic_student_prediction es (valor de 1 significa mostrar la nueva categoría y 0 no mostrarla)
    - {“uuid”: 1234567, “value”: 1, “token”:“23u84023u4h0” }

Resultado esperado

Una vez configurado el experimento en la plataforma y obtenido el token, puede comenzar en cualquier momento a leer casos de entrada y generar predicciones. La plataforma los irá recogiendo y cuando llegue al final del experimento notificará si su sistema ha mejorado al sistema base o no. Es responsabilidad del data scientist el configurar el nivel de significancia, poder estadístico, efecto

y tamaño de muestra. La plataforma no dejará comenzar un experimento hasta que los parámetros introducidos sean coherentes.

Anexo: Esquema de datos de topics de entrada

A continuación se proporcionan mensajes de ejemplo para los dos topics de entrada en kafka. Los campos se corresponden con el sexo, edad y compras en distintas categorías del histórico de clientes que se dará como conjunto de entrenamiento.

topic_demographic

{'uuid': 404180, 'age': 41, 'man': 1, 'woman': 0}

topic_historic

{'uuid': 404180, 'products': {'cat17': 0, 'cat74': 0, 'cat42': 0, 'cat96': 0, 'cat13': 0, 'cat16': 1, 'cat11': 0, 'cat35': 0, 'cat60': 0, 'cat68': 0, 'cat10': 0, 'cat37': 0, 'u4': 0, 'cat29': 0, 'cat98': 0, 'cat40': 0, 'cat46': 1, 'cat2': 0, 'cat41': 0, 'cat44': 0, 'cat89': 0, 'cat53': 0, 'cat23': 1, 'cat21': 0, 'cat36': 0, 'cat8': 0, 'cat20': 0, 'cat67': 0, 'cat4': 0, 'cat81': 0, 'cat84': 0, 'cat27': 0, 'cat55': 0, 'u9': 0, 'cat34': 1, 'cat22': 1, 'cat71': 0, 'cat32': 1, 'cat3': 0, 'cat43': 0, 'cat52': 0, 'cat83': 0, 'cat7': 0, 'cat26': 1, 'cat15': 0, 'u3': 0, 'cat70': 0, 'u5': 0, 'cat24': 0, 'cat49': 1, 'cat1': 0, 'cat19': 0, 'u1': 0, 'cat77': 1, 'cat72': 0, 'cat91': 1, 'cat73': 0, 'cat57': 1, 'cat87': 1, 'cat75': 0, 'cat54': 0, 'cat33': 0, 'cat61': 0, 'cat59': 0, 'cat63': 0, 'cat9': 0, 'cat99': 1, 'cat76': 1, 'cat95': 0, 'u0': 0, 'cat86': 0, 'cat97': 0, 'cat58': 0, 'cat69': 0, 'cat88': 0, 'cat0': 0, 'cat56': 0, 'u2': 0, 'cat85': 0, 'cat25': 0, 'cat94': 0, 'u6': 1, 'cat47': 0, 'cat51': 0, 'cat80': 0, 'cat93': 0, 'cat5': 0, 'cat18': 0, 'cat79': 0, 'cat48': 0, 'cat78': 0, 'cat65': 0, 'cat14': 1, 'cat50': 0, 'cat30': 0, 'cat31': 0, 'cat92': 0, 'cat28': 0, 'cat38': 0, 'cat64': 0, 'cat62': 0, 'cat6': 0, 'cat39': 0, 'cat45': 0, 'cat66': 0, 'cat12': 1, 'cat82': 0, 'u7': 0, 'u8': 0, 'cat90': 0}}

Nota sobre implementación

El alumno debe programar el pipeline de su solución sobre Apache Flink 1.9 sobre Scala 2.12 y proporcionar todos los archivos de configuración necesarios para que se pueda compilar simplemente ejecutando “sbt assembly”. Se da libertad al alumno para implementar el aprendizaje del modelo a partir de los datos de entrenamiento utilizando el lenguaje que desee. Dicho modelo deberá ser importado desde el pipeline de Flink en formato PMML.