Big Data Architecture
Big Data Machine Learning Bootcamp

Entregable

1. Selección del dataset

Trabajamos en el sector retail y mediante el análisis de datos queremos extraer conclusiones en los siguientes aspectos:

- Objetivo: conocer nuestros puntos fuertes y débiles, optimizar recursos, aumentar las ventas, entender el comportamiento del cliente, predecir la demanda y optimizar precios, detectar y prevenir la falsificación de productos, cross selling, up selling

- Qué hay que hacer: analizar información, procesarla y definir la estrategia a seguir para anticiparse a las expectativas del cliente, detectar fraude en la utilización de tarjetas de crédito, extraer conclusiones que ayuden a aumentar ventas, definir objetivos a corto y largo plazo.

Cómo: Procesado de datos a partir de: Datasets, APIs, ventas, fidelización, cookies, datos de compras presenciales y online, devoluciones, benchmarking, engagement, monitorización de marca, keywords, tasa de rebote, ROI, análisis de KPI´s, Google Trends, Google Analytics,

Resultado: adaptarnos según la información analizada. Los resultados nos dirán qué tenemos que cambiar, mejorar, estrategias de precios, definir estrategias a medida, establecer nuevos proyectos.

Dataset: nuestras dataset se encuentran en las siguientes webs:

Kaggle:
https://www.kaggle.com/manjeetsingh/retaildataset Data Analytics
https://www.kaggle.com/nazlisener/customer-segmentation-using-rfm: Customer Segmentation
https://www.kaggle.com/vijayuv/onlineretail: Online Retail

Toolbox Google:
https://datasetsearch.research.google.com/search?query=Retail%20spending%20in%20Europe%20in%202020%20by%20country&docid=L2cvMTFweDNmajNnbQ%3D%3D: Retail spending in Europe in 2020 by country

2. Big Data Architecture

- Seleccionamos Datasets estructurados en formatos XML, CSV, JSON.

Vemos que muchas no se adaptan a nuestras necesidades, son poco flexibles y no encontramos datos claros por ejemplo del sentimiento que genera la marca en el cliente o comentarios de noticias concretas así que buscamos diferentes APIS para complementar nuestras bases de datos y hacemos scraping para simplificar el proceso de extracción de datos, por lo que instalamos la siguiente librería: (este proceso se revisará todos los meses)

https://scrapy.org/

- Ahora buscamos APIS:

https://developer.twitter.com/ :buscamos datos en la API de Twitter para obtener Tweets del tema que queremos mediante filtrado por hashtag, perfiles de usuario y para generar Tweets automáticos por palabras seleccionadas por ejemplo “consulta”, “rebajas” “favoritos”.

Target.com: para búsquedas, reviews, detalles del producto

https://english.api.rakuten.net/apidojo/api/asos2/endpoints: ASOS API: productos, categorías, datos por país, descripción del producto, precio, descuentos….

https://apitracker.io/a/zalando: Zalando API: productos, categorías, datos por país, descripción del producto, precio, descuentos….

Almacenamos nuestras Datasets utilizando PostgreSQL y Elasticsearch

- Bases de datos PostgreSQL para almacenar los datos de las APIS de Twitter, Zalando y ASOS (con desarrollo en Python):

Proceso: Desde Google Cloud SQL:

Almacenamos los archivos Json de Twitter, Zalando y ASOS en Google Cloud Storage:

1. Entramos en la opción de Google Storage dentro de la consola
2. Creamos un bucket para esta información
3. Subimos los archivos seleccionados

Creamos una instancia de PostgreSQL con Cloud SQL.

1. Entramos con la opción de Cloud SQL dentro de la consola web.
2. Hacemos click en crear instancia
3. Seleccionamos PostgreSQL
4. Introducimos los datos de la instancia y hacemos click en opciones de configuración
5. Dentro de las opciones seleccionamos network e insertamos la información necesaria para acceder a la base de datos desde nuestro cliente
6. Creamos un usuario para conectarnos e insertar los datos
7. Insertamos los datos del usuario
8. Creamos la base de datos

Una vez terminada la creación del servicio SQL podemos entrar en “importar datos” y luego seleccionamos el archivo del dump que tenemos en storage, el formato y el usuario con el que hacer la importación.

Para comprobar que el import es correcto probamos la conexión haciendo chek y seleccionado SSH

La visualización de datos la realizaremos conectando PostgreSQL con Tableau: para detectar tendencias, pronósticos, explorar predicciones….

Se crearán “historias” para presentar la información obtenida al cliente quien tendrá acceso a los resultados.

- Para otras bases de datos utilizaremos Elasticsearch con codificación en Java o Python. Utilizamos Elasticsearch en bases de datos en las que necesitamos facilitar las búsquedas de texto: para realizar búsquedas, acceder a datos en tiempo real, almacenar logs, almacenar métricas, business analytics, para realizar búsquedas por varios campos a la vez (_all), para realizar búsqueda directa de palabras mediante filtrados.

Proceso: Desde Google Cloud:

1. En Compute Engine – VM instances hacemos click en créate Instance
2. Después rellenamos el formulario (configuración mínima 4GB de memoria y 2vCPUs
3. Esperamos a que se cree la instancia para poder entrar por SSH
4. Una vez dentro para poder lanzar comandos como administrador, instalamos wget
5. Luego instalamos las últimas versiones de Elastic y Kibana
6. Configuramos la conexión del IP
7. Configuramos kibana
8. Comprobamos que funcione
9. Ahora podemos cargar los datos

- Instalaremos Hadoop en la nube debido a su bajo costo y para fragmentar tareas en diferentes procesos y distribuirlos en diferentes clúster y así poder trabajar en paralelo y aprovechar las ventajas de los ficheros distribuidos HDFS y su procesamiento a través de los MapReduce jobs. Luego instalamos HIVE para solicitar, agrupar y analizar datos. Así podemos crear patrones y tendencias. Finalmente vamos a conectar Elasticsearch con Hadoop para trabajar con grandes ficheros de datos y para archivos de audio, vídeo e imágenes. Los resultados finales se presentarán al cliente mediante la visualización de datos en Kibana.

Proceso: Cluster de Hadoop en la nube:

1. En Google Cloud vamos a Compute Engine
2. Seleccionar un proyecto: crear
3. En Dataproc: crear agrupación
4. Crear un un clúster
5. Definir un clúster
6. Crear
7. Analizar la red VPC y el firewall
8. Añadimos en detalles de regla de cortafuegos para que nos permita entrar en las vistas de administración del clúster
9. Añadimos los puertos tcp
10. Buscamos la IP del nodo maestro buscando en Compute las instancias creadas (instancias de VM: SSH)
11. Entramos en el clúster haciendo check y conectando SSH

Ahora instalamos HIVE en el clúster de Hadoop

1. Instalamos las librerías necesarias para usar PyHIVE
2. Luego instalamos las librerías de Python para interactuar con HIVE
3. Accedemos a la base de datos de HIVE y hacemos una query
4. Cargamos los ficheros en HIVE
5. Conectamos Beeline a Hive

Conectamos Elasticsearch y Hadoop (ES – Hadoop)

Proceso: Ya tenemos creados nuestros clúster en Dataproc.

1. Descargamos los jars
2. Los metemos en Google Cloud Platform Storage
3. Modificamos el hive-site.xml para cargar los jars en la configuración de HIVE
4. Desde Beeline creamos una tabla que esté conecta a un índice de Elasticsearch

Se borrará el clúster cuando ya hayamos ejecutado el trabajo y no volvamos a necesitar ese clúster.

DIAGRAMA