



Evaluacion Base de Datos Nosql

Máster Data Science, Big Data & Business Analytics

Alumna: Lucía Trujillo Bregains

Introducción

En el marco de la asignatura de Base de Datos NoSQL, realizare el análisis del conjunto de datos **"Product_performance"**, con un enfoque de análisis sobre la experiencia del cliente, durante el proceso de compra en una página web. Como profesional de UX (Experiencia del Usuario), mi interés se centra en comprender y mejorar la interacción que los usuarios tienen al adquirir productos en línea. La elección de este conjunto de datos se fundamenta en su relevancia para el comercio electrónico y su capacidad para proporcionar información valiosa sobre la experiencia del cliente.

Los datos presentan una variedad de información clave, que se desglosa de la siguiente manera:

Cloth_type_color: Describe el tipo de prenda y su color correspondiente.

Visitors: Registra la cantidad de visitantes en la página web.

Sessions: Detalla las sesiones de interacción de los usuarios en la plataforma.

Duration: Indica la duración promedio de las sesiones de los visitantes.

Total_bounce_rate: Mide la tasa de rebote, proporcionando información sobre la retención de usuarios.

Total_carts: Registra la cantidad de veces que los usuarios agregaron productos al carrito.

Total_checkouts: Refleja el número de procesos de pago completados.

Total_conversion: Describe la tasa de conversión de visitantes a clientes.

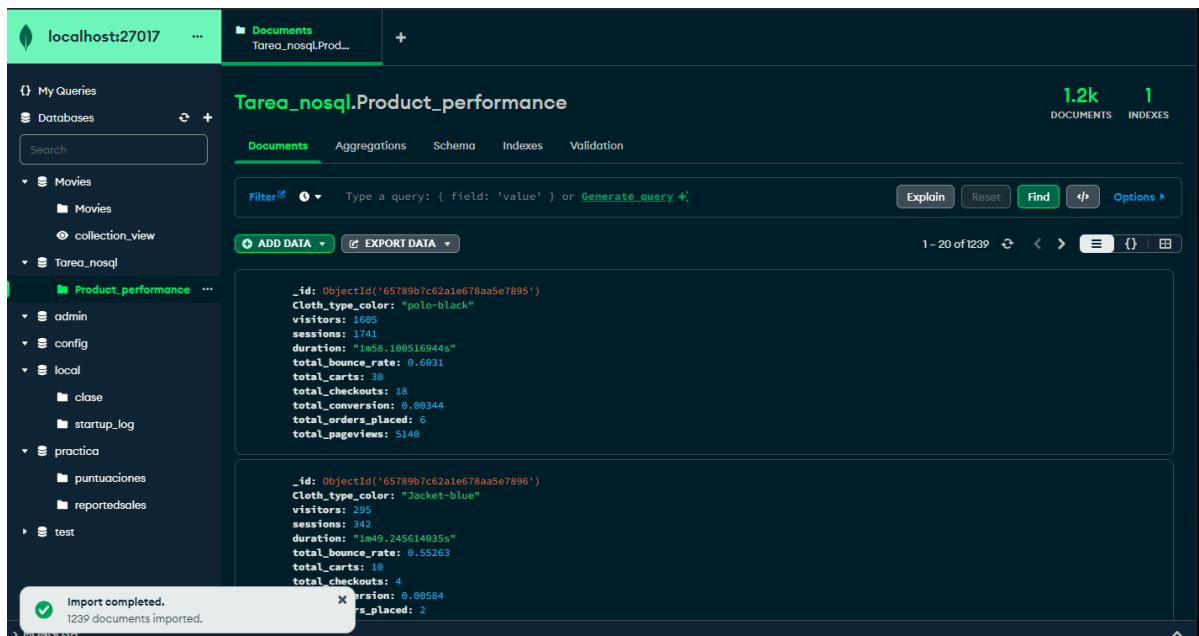
Total_orders_placed: Cuantifica el número total de pedidos realizados.

Total_pageviews: Indica la cantidad de veces que se ha visualizado una página.

1. Cargar / Importar dataset

Como parte de la aplicación de conocimientos en Base de Datos NoSQL, se procederá a cargar el conjunto de datos "Product_performance" en MongoDB. Se creará una nueva base de datos llamada "tareanosql", que incluirá la carga del data set:

"Product_performance.csv"



2. Ejercicios de Inserción, Actualización, Proyección y Filtrado en MongoDB

A lo largo de este trabajo, nuestro objetivo principal es aplicar los conocimientos adquiridos en la asignatura de Base de Datos NoSQL. Buscaremos extraer información significativa para comprender la experiencia del cliente en la plataforma actual y evaluaremos la eficacia de posibles mejoras en la página web mediante las queries realizadas. Este enfoque permitirá destacar la utilidad de las bases de datos NoSQL en la optimización de la experiencia del usuario en entornos de comercio electrónico.

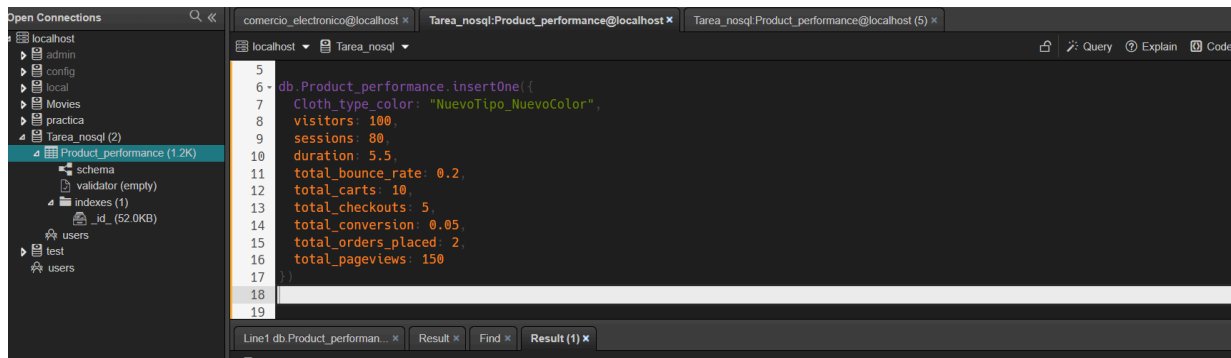
1. Inserción de Datos:

```
db.Product_performance.insertOne({
  Cloth_type_color: "NuevoTipo_NuevoColor",
  visitors: 100,
  sessions: 80,
  duration: 5.5,
  total_bounce_rate: 0.2,
```

```

total_carts: 10,
total_checkouts: 5,
total_conversion: 0.05,
total_orders_placed: 2,
total_pageviews: 150
})

```



Explicación:

Insertamos un nuevo registro en la colección "Product_performance" con valores específicos para cada campo. Este ejercicio simula la adición de información sobre un nuevo tipo de prenda con un color particular y métricas asociadas.

Conclusión de la query:

Al agregar datos sobre nuevos productos, como prendas de un tipo y color específicos, se enriquece la oferta en la plataforma. Esto contribuye a mejorar la experiencia del cliente al proporcionar variedad y novedades en la selección de productos, incentivando la exploración y compra.

2. Actualización de Datos:

```

db.Product_performance.updateOne(
  { Cloth_type_color: "t-shirt-pink" },
  { $set: { total_carts: 15, total_checkouts: 8 } }
)

```

Explicación:

Actualizamos la información de una prenda específica ("t-shirt-pink") para reflejar cambios en las métricas de carritos y pagos. Este ejercicio simula la actualización de datos de una prenda existente en el conjunto de datos.

Conclusión de la query:

La actualización de métricas como carritos y pagos permite ajustar la plataforma en tiempo real. En el contexto de interacciones "Add-to-Cart", esta actualización podría reflejar mejoras en la interfaz para facilitar el proceso de compra, proporcionando una experiencia más eficiente y satisfactoria.

#ANTES

The screenshot shows a MongoDB IDE interface. The top pane displays a JavaScript query to update a document in the 'Product_performance' collection. The query sets 'total_carts' to 15 and 'total_checkouts' to 8 for the document where 'Cloth_type_color' is 't-shirt-pink'. The bottom pane shows the result of the query, which is a document with various metrics.

```

15   total_orders_placed: 2,
16   total_pageviews: 150
17 }
18
19 db.Product_performance.updateOne(
20   { Cloth_type_color: "t-shirt-pink" },
21   { $set: { total_carts: 15, total_checkouts: 8 } }
22 )
23
24
25
26
27

```

Below the query editor, the results pane shows a table with columns 'Key', 'Value', and 'Type'. The table contains the following data:

Key	Value	Type
sessions	4	Int32
duration	2m48.75s	String
total_bounce_rate	0,50	Double
total_carts	0	Int32
total_checkouts	0	Int32
total_conversion	0	Int32
total_orders_placed	0	Int32
total_pageviews	42	Int32
(3) 65789b7c62a1e678aa5e7d6a	{11 fields}	Document
(4) 65789b7c62a1e678aa5e7d69	{11 fields}	Document
(5) 65789b7c62a1e678aa5e7d68	{11 fields}	Document

The status bar at the bottom indicates 'Line: 18, Column: 1' and 'Show Log'. The system tray shows the time as 10:45 AM on 13/12/2023.

#// verificar la actualización

```
var updatedDocument = db.Product_performance.findOne({ Cloth_type_color: "t-shirt-pink" });
```

```

13 total_checkouts: 5,
14 total_conversion: 0.05,
15 total_orders_placed: 2,
16 total_pageviews: 150
17 })
18
19 db.Product_performance.updateOne(
20   { Cloth_type_color: "t-shirt-pink" },
21   { $set: { total_carts: 15, total_checkouts: 8 } }
22 )
23
24 // Busca el documento actualizado
25 var updatedDocument = db.Product_performance.findOne({ Cloth_type_color: "t-shirt-pink" });
26
27 // Muestra el documento actualizado
28 printjson(updatedDocument);
29
30

```

Console x Line1 db.Product_performan... x Line6 db.Product_performan... x Line19 db.Product_performan... x

0.064 s Show Timestamps

```

1 {
2   "_id" : ObjectId("65789b7c62a1e678aa5e78da"),
3   "Cloth_type_color" : "t-shirt-pink",
4   "visitors" : 3,
5   "sessions" : 5,
6   "duration" : "2.4s",
7   "total_bounce_rate" : 0.6,
8   "total_carts" : 15,
9   "total_checkouts" : 8,
10  "total_conversion" : 0,
11  "total_orders_placed" : 0,
12  "total_pageviews" : 7
13 }

```

3. Proyección de Datos:

```

db.Product_performance.find(
  { Cloth_type_color: "t-shirt-navy" },
  { visitors: 1, total_conversion: 1, _id: 0 }
)

```

```

34 // Proyeccion de datos
35
36 db.Product_performance.find(
37   { Cloth_type_color: "t-shirt-navy" },
38   { visitors: 1, total_conversion: 1, _id: 0 }
39 )
40
41
42

```

Console x Line1 db.Product_performan... x Line6 db.Product_performan... x Line19 db.Product_performan... x Find x Find (1) x Find (2) x

Product_performance 0.041 s 39 Docs

Key	Value	Type
(1)	{ visitors : 2, total_conversion : 0 }	Object
visitors	2	Int32
total_conversion	0	Int32

Explicación:

Realizamos una consulta para obtener información específica sobre los visitantes y la tasa de conversión de los "t-shirt-navy". Se proyectan únicamente los campos necesarios y se excluye el campo "_id" para una presentación más clara de los resultados.

Conclusión de la query:

La proyección de datos es útil para concentrarse en métricas específicas relevantes para la experiencia del cliente. En el caso de "Add-to-Cart", obtener datos precisos sobre visitantes y tasas de conversión permite identificar qué productos tienen mayor interacción, facilitando la toma de decisiones para mejorar la visibilidad y promoción de dichos productos. En el caso de la t-shirt-navy, a simple vista podemos ver que es un producto que no representa mucho interés.

4. Filtrado de Datos:

```
db.Product_performance.find(
  {
    total_conversion: { $gte: 0.00714, $lte: 0.11111 },
    sessions: { $gt: 100 }
  },
  { Cloth_type_color: 1, _id: 0 }
).sort({ total_conversion: 1 })
```

The screenshot shows a MongoDB query being executed in a terminal window. The query filters documents based on 'total_conversion' and 'sessions' ranges, projects only 'Cloth_type_color', and sorts by 'total_conversion' in ascending order. Below the terminal, the MongoDB Compass interface displays the results of the query. The results are shown in a table with two columns: 'Key' and 'Value'. The first result is a document with 'Cloth_type_color' set to 'jeans-navy'. The subsequent results show documents with 'Cloth_type_color' set to 'Trousers-grey', 'T-shirt-white', 'jeans-blue', 'T-shirt-petrol', and 'T-shirt-turq'.

```
56 db.Product_performance.find(
57 {
58   total_conversion: { $gte: 0.00714, $lte: 0.11111 },
59   sessions: { $gt: 100 }
60 },
61 { Cloth_type_color: 1, _id: 0 }
62 ).sort({ total_conversion: 1 })
63 ///
64
```

Key	Value
(1)	{ Cloth_type_color : "jeans-navy" }
(2)	{ Cloth_type_color : "Trousers-grey" }
(3)	{ Cloth_type_color : "T-shirt-white" }
(4)	{ Cloth_type_color : "jeans-blue" }
(5)	{ Cloth_type_color : "T-shirt-petrol" }
(6)	{ Cloth_type_color : "T-shirt-turq" }

Explicación:

Filtramos los registros para obtener aquellos con una tasa de conversión superior al 10% y menos de 100 sesiones. Este ejercicio simula la búsqueda de prendas que han logrado una conversión significativa con un número limitado de interacciones, además se filtra para que solo muestre la propiedad `Cloth_type_color`, se utiliza `sort` para ordenar los resultados de manera ascendente.

Conclusión de la query:

El filtrado de datos ayuda a identificar productos que han tenido un impacto positivo con un público reducido. En términos de "Add-to-Cart", este enfoque puede revelar productos populares con una tasa de conversión notable, permitiendo estrategias específicas para destacar y mejorar la experiencia del usuario al interactuar con estos productos, además Al ordenar los resultados por tasas de conversión ascendentes, obtengo una lista que muestra los productos con tasas de conversión más altas primero. Esto proporciona una perspectiva clara sobre qué productos han tenido un rendimiento excepcional en términos de conversiones, lo que puede ser valioso para tomar decisiones estratégicas.

3. Ejercicios sobre pipeline de agregación:

Este ejercicio de agregación en MongoDB se enfoca en analizar la interacción entre tipos de ropa, colores y tasas de conversión en un conjunto de datos de comercio electrónico. El objetivo es responder a preguntas clave sobre la probabilidad de que ciertos tipos de ropa generen conversiones más efectivas y el impacto específico de los colores en estas tasas.

Utilizando un pipeline de agregación, se realizarán operaciones de agrupación y cálculo de tasas de conversión promedio para identificar combinaciones específicas de tipos de ropa y colores que puedan estar correlacionadas con un rendimiento superior en términos de conversiones

QUERY:

```

db.Product_performance.aggregate([
  {
    $group: {
      _id: {
        Cloth_type: "$Cloth_type_color.Cloth_type",
        Color: "$Cloth_type_color.Color"
      },
      avg_conversion: { $avg: "$total_conversion" },
      count: { $sum: 1 }
    }
  },
  {
    $match: {
      count: { $gt: 50 } // Considerar solo grupos con más de 10 registros para mayor
significancia
    }
  },
  {
    $sort: {
      avg_conversion: -1 // Ordenar por tasas de conversión descendentes
    }
  }
])

```

Explicación:

\$group: Agrupamos los documentos por tipo de ropa y color, calculando la tasa de conversión promedio (avg_conversion) y contando el número de registros en cada grupo (count).

\$match: Filtramos los resultados para considerar solo los grupos con más de 50 registros, para garantizar una mayor significancia estadística.

\$sort: Ordenamos los resultados por tasas de conversión promedio de manera descendente, para destacar los grupos con tasas más altas.

```

66 // Ejecutar pipeline de agregación
67
68 db.Product_performance.aggregate([
69   {
70     $group: {
71       _id: {
72         Cloth_type: "$Cloth_type_color.Cloth_type",
73         Color: "$Cloth_type_color.Color"
74       },
75       avg_conversion: { $avg: "$total_conversion" },
76       count: { $sum: 1 }
77     }
78   },
79   {
80     $match: {
81       count: { $gt: 50 } // Considerar solo grupos con más de 50 registros para mayor significancia
82     }
83   },
84   {
85     $sort: {
86       avg_conversion: -1 // Ordenar por tasas de conversión descendentes
87     }
88   }
89 ])

```

Key	Value	Type
{}	{ avg_conversion : 0.009410825982357658, count : 1247 }	Document
_id	{}	Object
avg_conversion	0.0094 0	Double

Conclusión:

Al ejecutar el pipeline de agregación en MongoDB para analizar la influencia de tipos de ropa y colores en las tasas de conversión, se obtuvo una tasa de conversión promedio de aproximadamente 0.0094. Este valor representa la media de las conversiones en la plataforma de comercio electrónico, sirviendo como indicador general del rendimiento de la misma.

El pipeline permitió realizar un análisis detallado, identificando combinaciones específicas de tipos de ropa y colores que podrían estar correlacionadas con tasas de conversión superiores. Este resultado inicial proporciona una base sólida para comprender cómo diferentes atributos afectan las tasas de conversión en la plataforma.

Conclusión Final

En este trabajo realizado con MongoDB, se ha mostrado la utilización de estas herramientas para analizar el comportamiento del usuario en el comercio electrónico de ropa.

Exploración Detallada de Datos:

El uso hábil de funciones de agregación ha permitido identificar patrones cruciales, ofreciendo una comprensión profunda de la experiencia del cliente.

Impacto Estratégico:

La conclusión de que ciertos tipos de ropa y colores influyen significativamente en las tasas de conversión destaca la importancia de estrategias ágiles y adaptables. Este proyecto destaca la necesidad de mediciones continuas y estrategias basadas en datos, para mejorar el rendimiento del establecimiento.