

DATA ANALYTICS

FASE 3 | AULA 06 -

CLUSTERING COM BIGQUERY ML

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA?	17
REFERÊNCIAS.....	18

EMSE

O QUE VEM POR AÍ?

Bem-vindos e bem-vindas ao material de aula sobre clustering de dados no ambiente Google BigQuery!

O clustering é uma técnica de análise de dados que tem como objetivo agrupar itens similares com base em suas características. Essa abordagem permite identificar padrões, descobrir segmentos de clientes, entender comportamentos de mercado e tomar decisões estratégicas fundamentadas.

E quando se trata de trabalhar com grandes volumes de dados, o Google BigQuery se destaca como uma solução eficiente e escalável. Ele é um serviço de armazenamento e análise de dados totalmente gerenciado, oferecido pela Google Cloud Platform. O BigQuery permite a execução de consultas rápidas e complexas em grandes conjuntos de dados, proporcionando insights valiosos em tempo real.

Neste material de aula, exploraremos como utilizar o clustering de dados no ambiente Google BigQuery. Abordaremos os conceitos fundamentais do clustering, desde os algoritmos mais populares até as melhores práticas de pré-processamento de dados. Em seguida, mergulharemos nas funcionalidades do BigQuery, mostrando como aplicar esses conceitos em um ambiente prático e orientado a projetos.

Ao final desta aula, você terá adquirido conhecimentos sólidos para utilizar o clustering de dados no Google BigQuery, permitindo a você explorar informações valiosas e obter insights relevantes a partir de grandes conjuntos de dados. Estamos empolgados(as) para compartilhar esse conhecimento com você e vê-lo(a) se destacar nesse campo promissor da análise de dados. Vamos começar!

HANDS ON

Para encontrar os materiais utilizados no nosso Hands On, acesse o [github da disciplina](#).

Foque em entender tudo o que os códigos transmitem, replique-os em sua máquina local, e teste com muita dedicação para que o aprendizado seja definitivo!

EMANDA

SAIBA MAIS

Entendendo O Que É Clustering De Dados

Clustering, no contexto de Machine Learning, é uma técnica usada para agrupar objetos ou exemplos de dados similares em grupos, também conhecidos como clusters. O objetivo principal é encontrar padrões intrínsecos nos dados sem a necessidade de rótulos prévios ou supervisão.

O processo de clustering envolve a atribuição de pontos de dados a grupos com base em sua similaridade. A similaridade é medida usando uma função de distância ou similaridade, que quantifica quão próximos ou semelhantes dois pontos de dados estão entre si. Os pontos de dados que são mais próximos uns dos outros têm uma maior similaridade e são agrupados juntos.

Existem vários algoritmos de clustering disponíveis, cada um com suas próprias abordagens e suposições sobre a estrutura dos dados. Alguns dos algoritmos de clustering mais comuns são o K-means, o DBSCAN (Density-Based Spatial Clustering of Applications with Noise) e o Hierarchical Clustering.

O algoritmo K-means é um dos mais amplamente utilizados. Ele atribui pontos de dados à clusters de forma iterativa, minimizando a soma dos quadrados das distâncias entre os pontos e os centroides dos clusters. O número de clusters é especificado antecipadamente no K-means.

O DBSCAN, por outro lado, é um algoritmo baseado em densidade que agrupa pontos de dados em regiões densas do espaço. Ele não requer um número pré-definido de clusters e pode identificar automaticamente o número correto com base na densidade dos dados.

O Hierarchical Clustering, como o nome sugere, cria uma hierarquia de clusters, formando uma árvore de clusters. Ele pode ser aglomerativo (começando com clusters individuais e mesclando-os) ou divisivo (começando com um cluster único e dividindo-o em subclusters).

O clustering tem várias aplicações em Machine Learning e análise de dados. Pode ser usado para segmentação de clientes, agrupamento de documentos, análise de redes sociais, detecção de anomalias, entre outros. Essa técnica ajuda a identificar

grupos naturais nos dados, permitindo uma melhor compreensão e organização dos mesmos.

Colocando A Mão Na Massa Com Bigquery ML

O BigQuery tem vários conjuntos de dados de demonstração gratuitos. Neste exemplo específico, usaremos o conjunto de dados “London Bicycle Hire” para construir o agrupamento K-means.

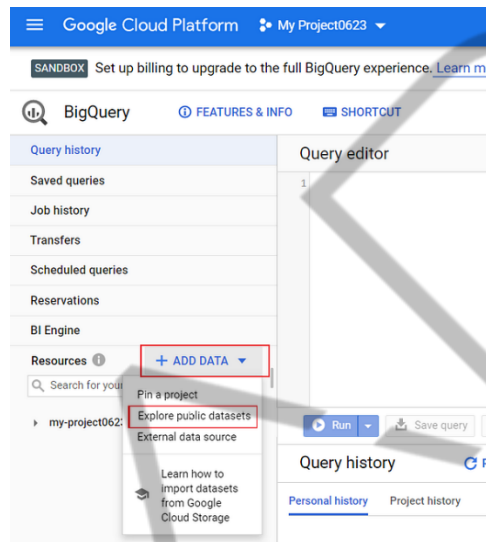


Figura 1 – Google Cloud Plataforma 1
Fonte: Kevin Bok (2020)

Primeiro, encontre “+ADD DATA” no painel esquerdo e clique em 'Explore public datasets'

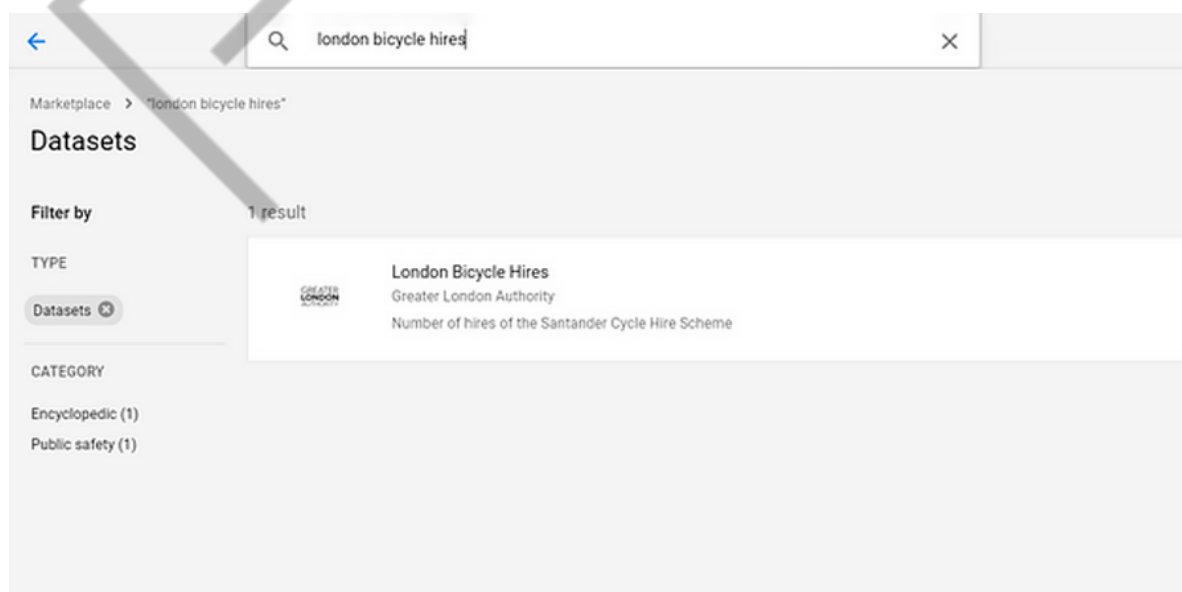


Figura 2 – Google Cloud Plataforma 2
Fonte: Kevin Bok (2020), adaptada por FIAP (2024)

Pesquise por “**London Bicycle Hires**” e clique em “**View dataset**”.

A partir daí, você pode ver o banco de dados “**bigquery-public-data**”, adicionado no canto inferior esquerdo. Se você ver london_bicycles ao rolar para baixo, então estamos prontos(as)!

Para criar um modelo de agrupamento K-means no BigQuery, precisamos criar um 'Dataset' que salve o modelo que iremos construir.

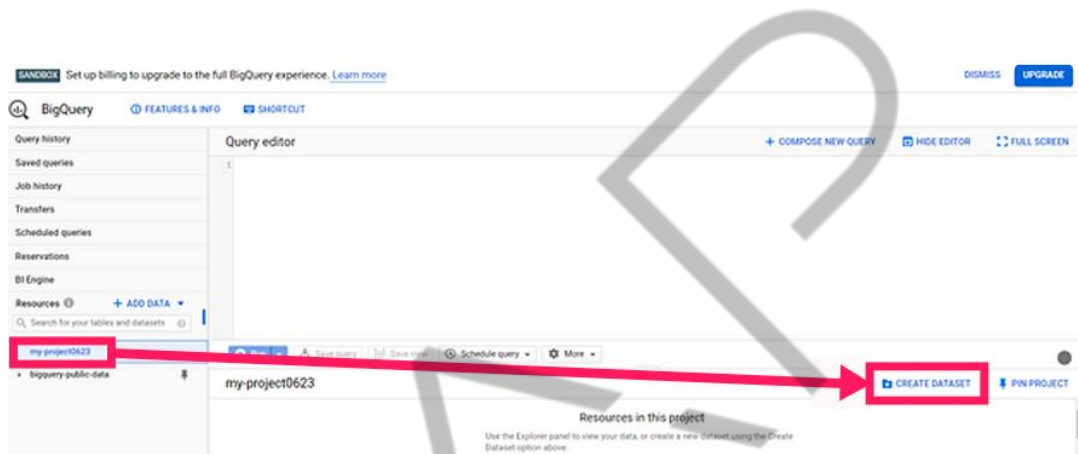


Figura 3 – Create Dataset 1
Fonte: Kevin Bok (2020), adaptada por FIAP (2024)

Para fazer isso, clique no **ID do projeto** que você criou na etapa 1 e, em seguida, clique em 'CREATE DATASET' no lado direito do monitor, conforme exemplificado na figura 3 – “Create Dataset 1”.

Figura 4 – Create Dataset 2
Fonte: Kevin Bok (2020), adaptada por FIAP (2024)

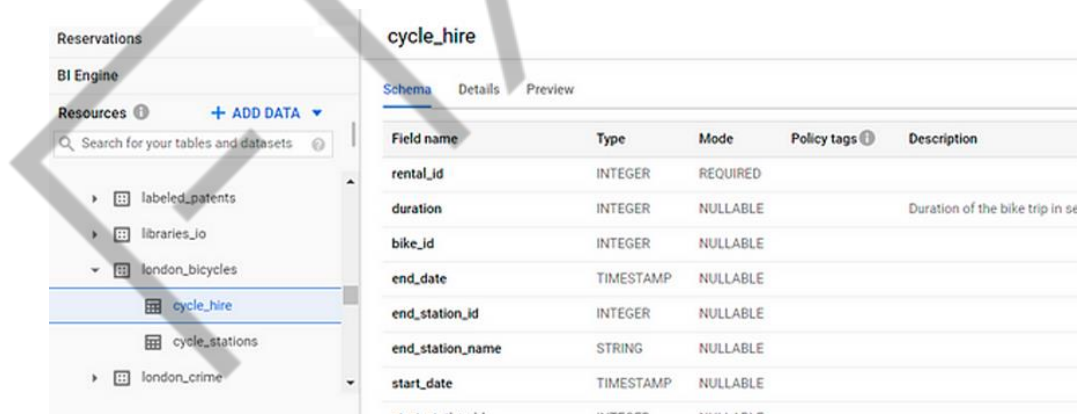
Coloque **k_means_tutorial** em Dataset ID e certifique-se de escolher 'EU' em Data location (os dados da bicicleta de Londres são armazenados na multirregião da EU (União Europeia), portanto, esse conjunto de dados também deve estar localizado na mesma região). Deixe as outras configurações como estão e clique em “Criar conjunto de dados”.

Como mencionado anteriormente, o dataset que vamos usar é **london_bicyclesdataset**, e vamos agrupar a estação de bicicletas por três das seguintes características:

- Duração do aluguel
- Número de aluguéis diários
- Distância da cidade

london_bicyclesdataset tem duas tabelas (**cycle_hire** e **cycle_stations**). Você pode clicar em cada conjunto de dados para ver as colunas de cada tabela.

cycle_hire: uma tabela de aluguel que tem **rental_id** e **bike_id** como sua chave, e tem informações de duração e estação inicial/final para cada aluguel de bicicleta

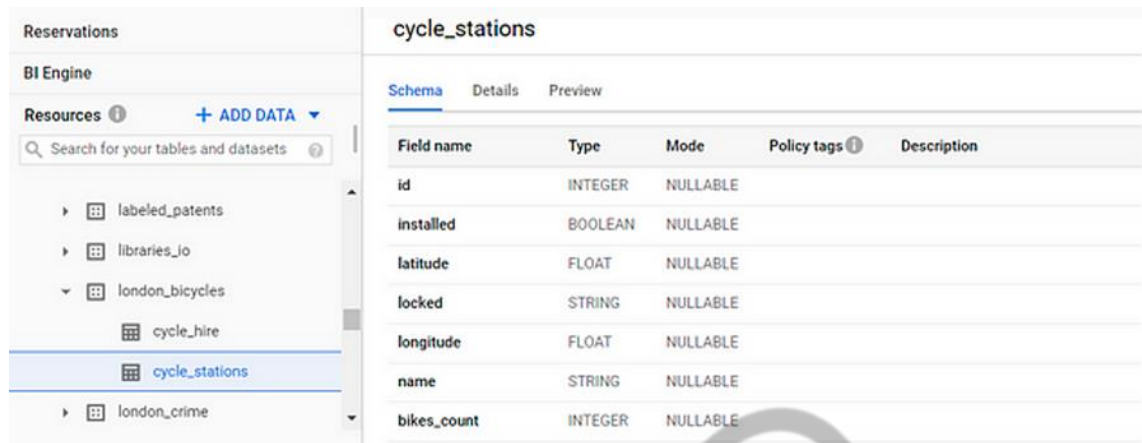


Field name	Type	Mode	Policy tags	Description
rental_id	INTEGER	REQUIRED		
duration	INTEGER	NULLABLE		Duration of the bike trip in se
bike_id	INTEGER	NULLABLE		
end_date	TIMESTAMP	NULLABLE		
end_station_id	INTEGER	NULLABLE		
end_station_name	STRING	NULLABLE		
start_date	TIMESTAMP	NULLABLE		
start_station_id	INTEGER	NULLABLE		

Figura 5 – cycle_hire

Fonte: Kevin Bok (2020), adaptada por FIAP (2023)

cycle_stations: dados para a estação de aluguel de bicicletas (longitude/latitude e número de bicicletas para cada estação).



Field name	Type	Mode	Policy tags	Description
id	INTEGER	NULLABLE		
installed	BOOLEAN	NULLABLE		
latitude	FLOAT	NULLABLE		
locked	STRING	NULLABLE		
longitude	FLOAT	NULLABLE		
name	STRING	NULLABLE		
bikes_count	INTEGER	NULLABLE		

Figura 6 – cycle_stations
Fonte: Kevin Bok (2020)

As variáveis que vamos usar para agrupar as estações são:

- Duração do aluguel.
- Número de aluguéis diários.
- Distância da cidade.

Passe o mouse sobre o projeto **k_means_tutorial** que criamos e copie e cole a consulta SQL abaixo no **editor de consultas** e clique em **“Executar”**.

```
with hs as (
select
h.start_station_name as station_name
,if (extract(dayofweek from h.start_date) = 7 or
extract(dayofweek from h.start_date) = 1,
"weekend","weekday") as isweekday
,h.duration
,st_distance(st_geogpoint(s.longitude, s.latitude), st_geogpoint(-0.1, 51.5)) / 1000 as
distance_from_city_center
from `bigquery-public-data.london_bicycles.cycle_hire` as h
join `bigquery-public-data.london_bicycles.cycle_stations` as s
on h.start_station_id = s.id
```

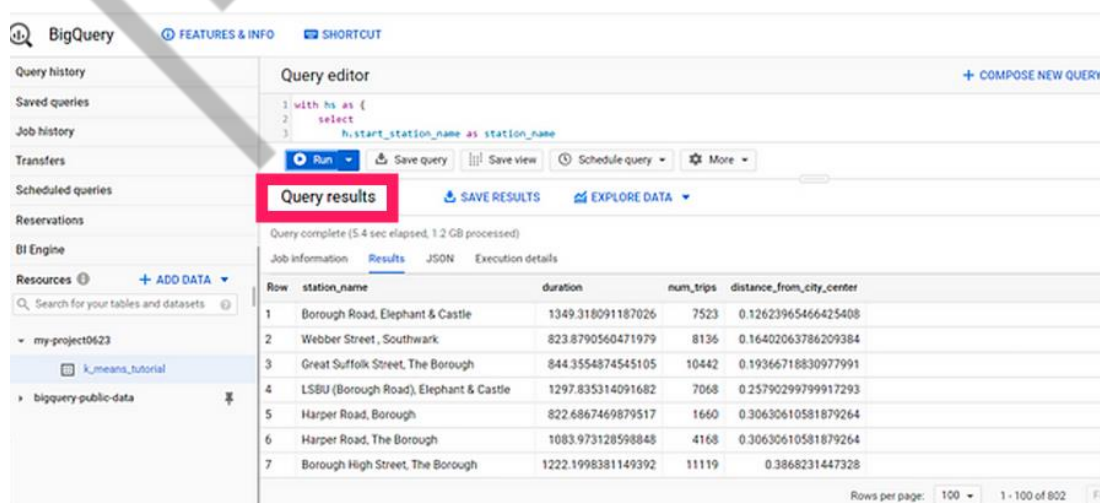
```

where h.start_date between cast('2015-01-01 00:00:00' as timestamp) and
cast('2016-01-01 00:00:00' as timestamp)
),
stationstats as (
select
station_name
,avg(duration) as duration
,count(duration) as num_trips
,max(distance_from_city_center) as distance_from_city_center
from hs
group by station_name
)

select *
from stationstats
order by distance_from_city_center

```

Depois disso, você deve ver algo como o que está na figura “Query results”, na seção “Resultados da consulta”.



Query complete (5.4 sec elapsed, 1.2 GB processed)

Row	station_name	duration	num_trips	distance_from_city_center
1	Borough Road, Elephant & Castle	1349.318091187026	7523	0.12623965466425408
2	Webber Street, Southwark	823.8790560471979	8136	0.16402063786209384
3	Great Suffolk Street, The Borough	844.3554874545105	10442	0.19366718830977991
4	LSBU (Borough Road), Elephant & Castle	1297.835314091682	7068	0.25790299799917293
5	Harper Road, Borough	822.6867469879517	1660	0.30630610581879264
6	Harper Road, The Borough	1083.973128598848	4168	0.30630610581879264
7	Borough High Street, The Borough	1222.1998381149392	11119	0.3868231447328

Rows per page: 100 1 - 100 of 802

Figura 7 – Query results

Fonte: Kevin Bok (2020), adaptada por FIAP (2023)

O que está sendo realizado nesta consulta SQL:

- Criamos duas tabelas temporárias 'hs' e 'stationstats' como subconsulta.
- 'hs' fornece as informações em cada linha para as estações em 2015 (nome, dia da semana, duração do aluguel, distância da cidade).
- Em seguida, na tabela 'stationstats', usamos funções agregadas para calcular algumas estatísticas importantes armazenadas na tabela 'hs'.

Agora que entendemos a logística básica do BigQuery, podemos criar um modelo de agrupamento K-means.

Podemos fazer um create model com model_type = 'kmeans' e treinaremos o modelo de agrupamento.

Copie e cole abaixo novamente e **clique em 'Executar'**.

```
create or replace model
k_means_tutorial.london_station_clusters
OPTIONS(model_type='kmeans',num_clusters=4) as
with hs as (
select
h.start_station_name as station_name
,if (extract(dayofweek from h.start_date) = 7 or
extract(dayofweek from h.start_date) = 1,
"weekend","weekday") as isweekday
,h.duration
,st_distance(st_geogpoint(s.longitude, s.latitude), st_geogpoint(-0.1, 51.5)) / 1000 as
distance_from_city_center
from `bigquery-public-data.london_bicycles.cycle_hire` as h
join `bigquery-public-data.london_bicycles.cycle_stations` as s
on h.start_station_id = s.id
where h.start_date between cast('2015-01-01 00:00:00' as timestamp) and
cast('2016-01-01 00:00:00' as timestamp)
```

```
),  
stationstats as (  
  select  
    station_name  
    , isweekday  
    , avg(duration) as duration  
    , count(duration) as num_trips  
    , max(distance_from_city_center) as distance_from_city_center  
  from hs  
  group by station_name, isweekday  
)  
select * except(station_name, isweekday)  
from stationstats  
order by distance_from_city_center
```

Esta consulta SQL difere da anterior em apenas **duas partes**:

- Adicionamos uma linha para criar um modelo no conjunto de dados `k_means_tutorial` que criamos acima e o resultado é armazenado como `london_station_clusters`.

```
create or replace model  
bqml_tutorial.london_station_clusters options (model_type='kmeans',num_clusters=4) as
```

- Adicionada uma nova coluna 'isweekday' na consulta 'groupby' na tabela `stationstats` e também adicionado `except (station_name, isweekday)`. Para isso, queremos ver como o dia da semana/fim de semana afeta a taxa de aluguel. Além disso, a consulta **Exceto (coluna)** exclui os nomes das colunas dentro dos parênteses.



Figura 8 – Go to model
 Fonte: Kevin Bok (2020), adaptada por FIAP (2023)

Uma vez feito, você deve ver algo como a tela da figura “Go to model”. Clique em 'Ir para o modelo'(go to model) e verifique os detalhes.

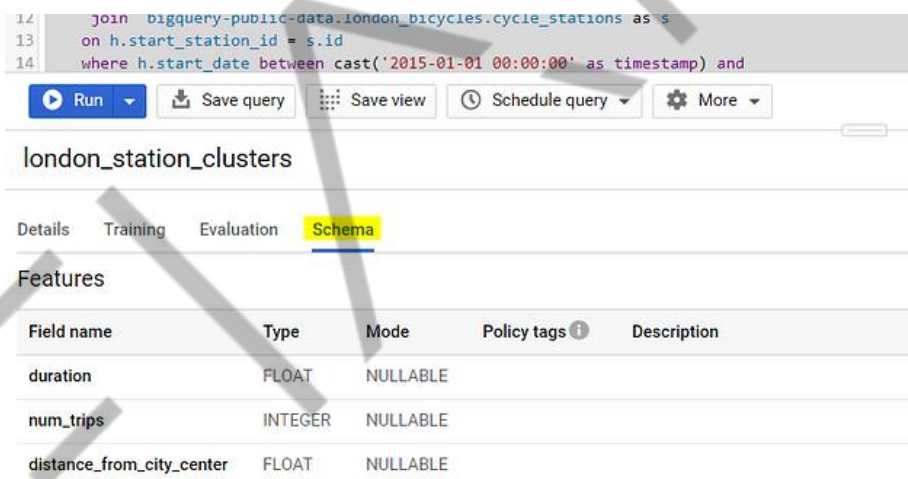


Figura 9 – london_station_clusters
 Fonte: Kevin Bok (2020)

Se você clicar em “Esquema” (Schema), poderá ver que o modelo é treinado usando três colunas (duration, num_trips, distance_from_city_center).

london_station_clusters

Details Training **Evaluation** Schema

Metrics

Davies-Bouldin index	1.2408
Mean squared distance	1.3185

Numeric features

This table shows the centroid value for each feature. Use the select menu to view more numeric features.

Select features (3/3) ▾







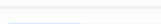
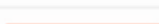
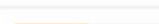



Centroid Id	Count	duration	num_trips	distance_from_city_center
1	268	 2,571.8349	 2,968.6269	 6.0658
2	643	 1,406.7382	 4,898.3250	 2.6291
3	154	 1,205.1628	 20,850.7597	 2.9461
4	539	 1,423.4450	 4,400.4323	 6.1634

Figura 10 – Evaluation

Fonte: Kevin Bok (2020), adaptada por FIAP (2023)

Se você clicar em “Avaliação” (Evaluation), poderá ver cada cluster. Temos 4 clusters desde que definimos num_clusters를 como 4 em create model, e vemos “valor do centroide”.

Vamos usar a função interna 'ml.predict' para encontrar o cluster ao qual uma determinada estação pertence.

```
with hs as (
```

```
select
```

```
h.start_station_name as station_name
```

```
,if (extract(dayofweek from h.start_date) = 7 or
```

```
extract(dayofweek from h.start_date) = 1,
```

```
"weekend","weekday") as isweekday
```

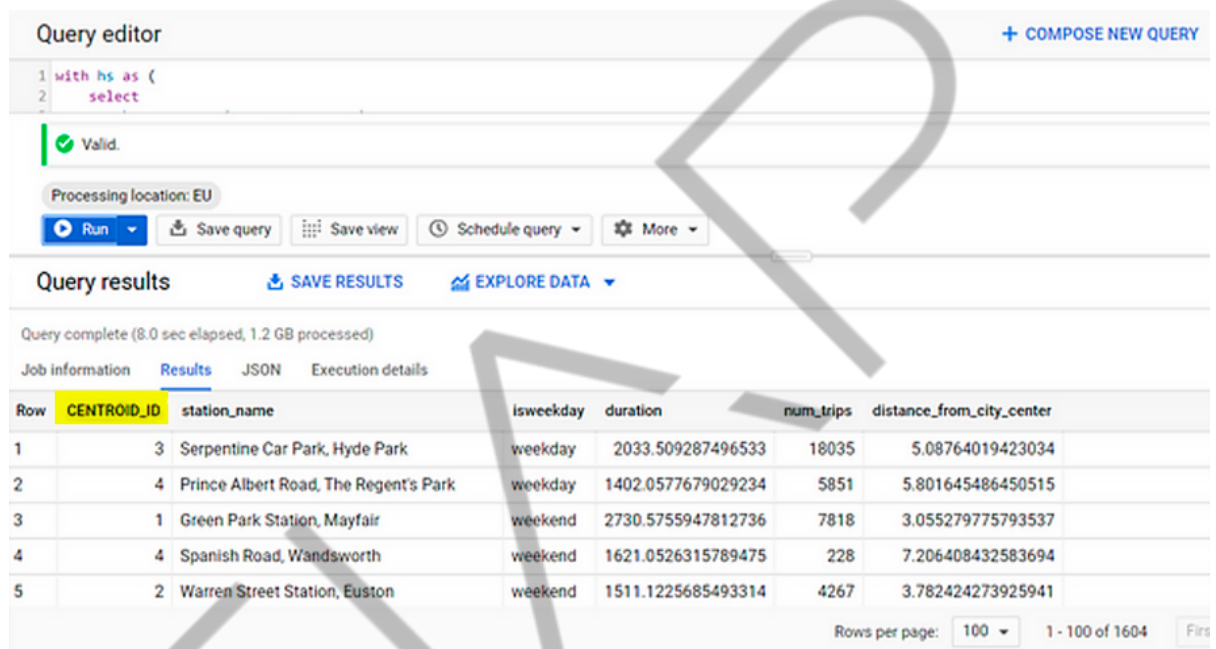
```
,h.duration
```

```
,st_distance(st_geogpoint(s.longitude, s.latitude), st_geogpoint(-0.1, 51.5)) / 1000 as
distance_from_city_center
from `bigquery-public-data.london_bicycles.cycle_hire` as h
join `bigquery-public-data.london_bicycles.cycle_stations` as s
on h.start_station_id = s.id
where h.start_date between cast('2015-01-01 00:00:00' as timestamp) and
cast('2016-01-01 00:00:00' as timestamp)
),
stationstats as (
select
station_name
, isweekday
, avg(duration) as duration
, count(duration) as num_trips
, max(distance_from_city_center) as distance_from_city_center
from hs
group by station_name, isweekday
)
select * except(nearest_centroids_distance)
from ml.predict(
model k_means_tutorial.london_station_clusters,
(
select *
from stationstats
)
)
```

A consulta acima tem duas partes:

- Modelo de agrupamento K-means que criamos.
- Dados do conjunto de teste (para previsão).

Adicionamos `except(nearest_centroids_distance)` para ver apenas os clusters previstos.



Query editor

1 with hs as (
2 select

Valid.

Processing location: EU

Run Save query Save view Schedule query More

Query results SAVE RESULTS EXPLORE DATA

Query complete (8.0 sec elapsed, 1.2 GB processed)

Job information Results JSON Execution details

Row	CENTROID_ID	station_name	isweekday	duration	num_trips	distance_from_city_center
1	3	Serpentine Car Park, Hyde Park	weekday	2033.509287496533	18035	5.08764019423034
2	4	Prince Albert Road, The Regent's Park	weekday	1402.0577679029234	5851	5.801645486450515
3	1	Green Park Station, Mayfair	weekend	2730.5755947812736	7818	3.055279775793537
4	4	Spanish Road, Wandsworth	weekend	1621.0526315789475	228	7.206408432583694
5	2	Warren Street Station, Euston	weekend	1511.1225685493314	4267	3.782424273925941

Rows per page: 100 1 - 100 of 1604

Figura 11 – CENTROID_ID
Fonte: Kevin Bok (2020), adaptada por FIAP (2023)

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, aprendemos sobre clustering de dados no ambiente Google BigQuery. O clustering é uma técnica de análise que agrupa itens similares com base em suas características, permitindo identificar padrões e tomar decisões estratégicas. O Google BigQuery é um serviço eficiente e escalável de armazenamento e análise de dados, que oferece consultas rápidas e insights em tempo real.

Exploramos os conceitos fundamentais do clustering, desde os algoritmos mais populares até as melhores práticas de pré-processamento de dados. Também conhecemos as funcionalidades do BigQuery, aplicando esses conceitos em projetos práticos.

Ao final da aula, você estará pronto e pronta para utilizar o clustering de dados no Google BigQuery, explorando informações valiosas e obtendo insights relevantes de grandes conjuntos de dados.

O que achou dessa disciplina? Conte-nos no Discord! Estaremos lá para responder eventuais dúvidas, interagir e muito mais.

REFERÊNCIAS

Basic Topic Clustering using TensorFlow and BigQuery ML, [s.d.]. Disponível em: <https://bigquery-lab.dimensions.ai/tutorials/05-topic_clusters/>. Acesso em: 20 jun 2023.

Create a k-means model to cluster London bicycle hires dataset, [s.d.]. Disponível em: <<https://cloud.google.com/bigquery/docs/kmeans-tutorial>>. Acesso em: 20 jun 2023.

ANASTACIO, BRUNO. **K-means: o que é, como funciona, aplicações e exemplo em Python**, 2020. Disponível em: <<https://medium.com/programadores-ajudando-programadores/k-means-o-que-é-como-funciona-aplicações-e-exemplo-em-python-6021df6e2572>> Acesso em: 20 jun 2023.

BOK, KEVIN. **K-Means Clustering in Google BigQuery ML**, 2020. Disponível em: <<https://medium.datadriveninvestor.com/k-means-clustering-in-google-bigquery-ml-b02907a961a8>>. Acesso em: 20 jun 2023.

CLUSTERING. Disponível em: <<https://scikit-learn.org/stable/modules/clustering.html>>. Acesso em: 20 jun. 2023.

Lakshmanan, LAK. **How to use K-Means clustering in BigQuery ML to understand and describe your data better**, 2019. Disponível em: <<https://towardsdatascience.com/how-to-use-k-means-clustering-in-bigquery-ml-to-understand-and-describe-your-data-better-c972c6f5733b>>. Acesso em: 20 jun 2023.

PALAVRAS-CHAVE

Palavras-chave: Machine Learning, BigQueryML, Big Query, SQL, Python, Kmeans, DBSCAN.

EMEND

The background is a dark blue field filled with numerous small, light blue dots. Overlaid on this are several large, wavy, translucent lines in shades of blue and yellow. A vertical line with a small 'x' at the bottom is on the left. A circle containing the number '7' is in the upper center. A hexagon is in the lower right.

POSTECH