Business Analytics, Geo Analytics and Machine Learning with BigQuery

Alex Urcola
Kschool
Clase 1/2

¿Quién soy? : Alex Urcola

Alex Urcola
Senior Business Analytics Consultant
Google Spain

Agenda

01 Introduccion a BigQuery

02 Conceptos básicos de Standard SQL

Break (🎉🎉)

03 DataStudio como herramienta de Visualización

04 Ejercicios prácticos BigQuery y DataStudio

05 BigQuery GeoViz

Fin (🎉🎉)

Agenda

01 Introduccion a BigQuery

02 Conceptos básicos de Standard SQL

Break (🎉🎉)

03 DataStudio como herramienta de Visualización

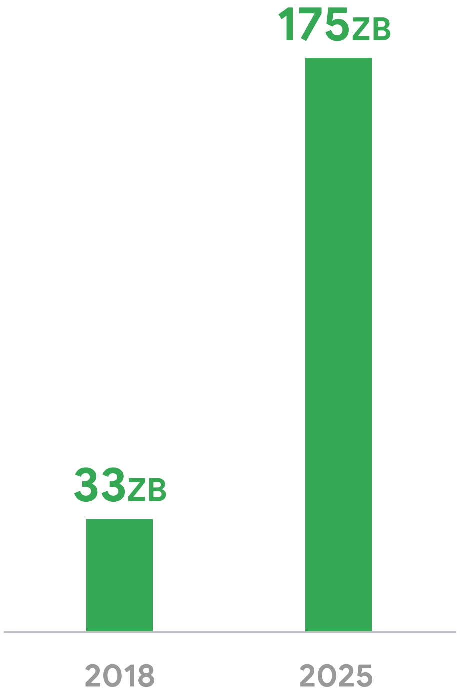
04 Ejercicios prácticos BigQuery y DataStudio

05 BigQuery GeoViz

Fin (🎉🎉)

Traditional data warehouses are melting with data growth

World datasphere will grow from
33 ZB in 2018
to 175 ZB by 2025 -IDC*



https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

Is your data warehouse ready for real-time data ?

“By 2025, more than a quarter of data created in the global datasphere will be real time in nature.”

IDC

Google BigQuery

Google Cloud Platform's
enterprise data warehouse
for analytics

Gigabyte- to **petabyte-scale**
storage and SQL queries

Encrypted, durable,
And highly available

GeoVizualizations
And geometric operations

Unique

Real-time insights from streaming data

Unique

Built-in **ML and GIS** for out-of-the-box
predictive insights

Unique

High-speed, in-memory **BI Engine**
for faster reporting and analysis

Unique

Modelos de ML de BigQuery

Classification

- Logistic regression
- DNN classifier (TensorFlow)
- XGBoost
- AutoML Tables

Other Models

- k-means clustering
- Time series forecasting
- Recommendation: Matrix factorization

Regression

- Linear regression
- DNN regressor (TensorFlow)
- XGBoost
- AutoML Tables

Model Import/Export

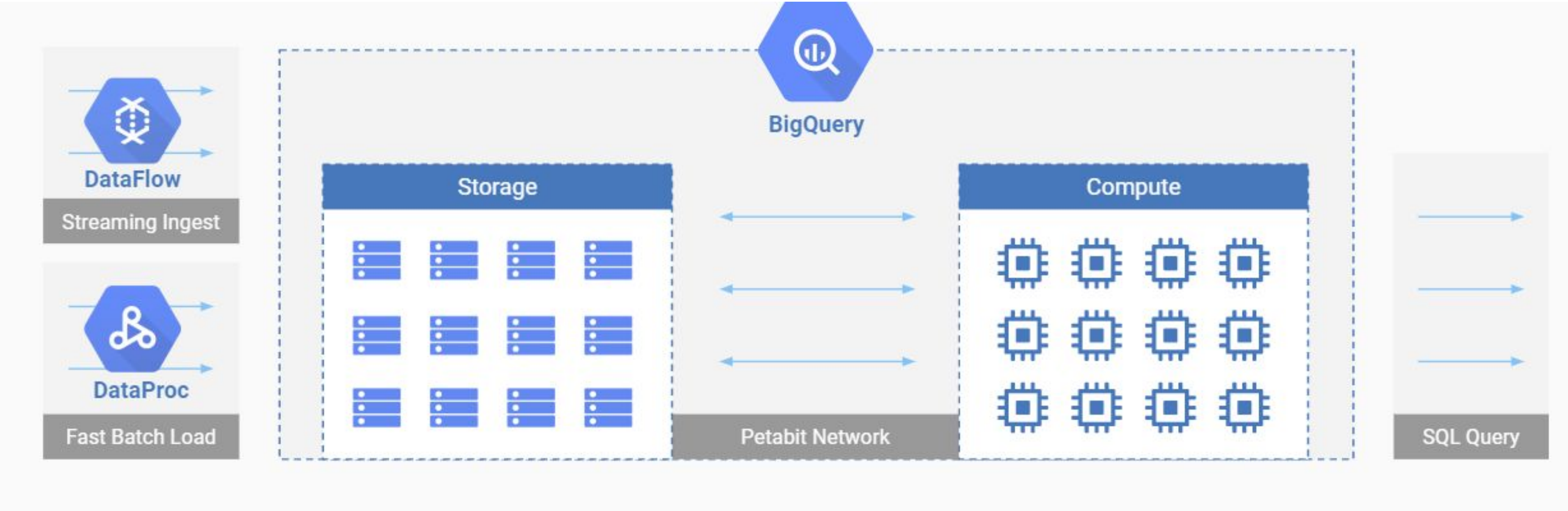
- TensorFlow models for batch and online prediction

BigQuery | Why is so powerful

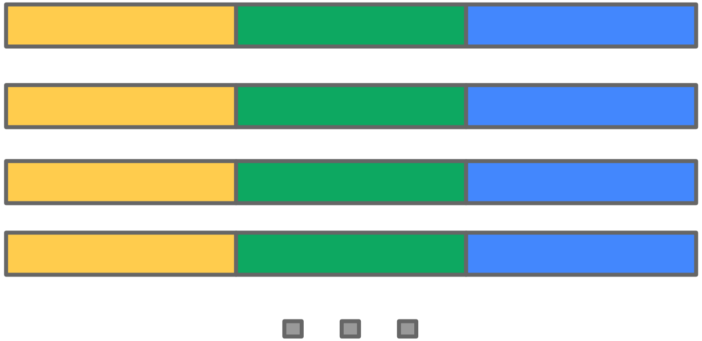
1. **Storage Differentiated from compute:** Permanent Storage Vs Temporal compute makes cheaper and faster
2. **Columnar Based storage:** Data model is based on columns vs registers making faster and cheaper
3. **Serverless:** Let BigQuery do the heavy lifting for you

BigQuery | Storage Vs Compute

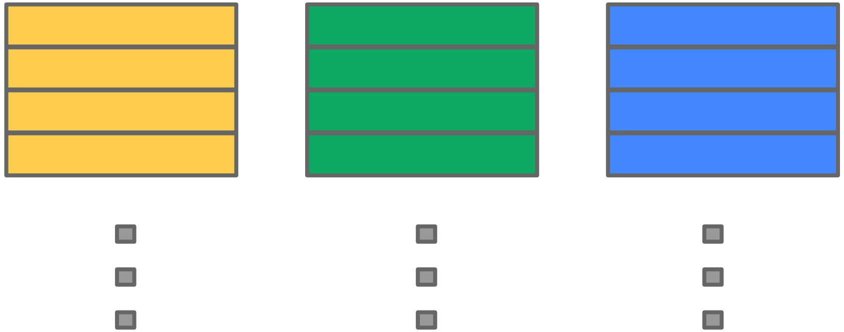
BigQuery = **Massively Parallel Processing** query with the petabit network and thousands of servers



BigQuery | Columnar based storage



Record-Oriented Storage



Column-Oriented Storage

BigQuery | Columnar based storage - Example

Table Definition

```
message Book {
 required string title,
 repeated string author,
 repeated group price {
   optional int64 discount,
   optional int64 usd,
   optional int64 eur,
 }
}
```

Field Records

```
Book1:
 author: "AAA"
 title: "firstTitle"
 price:
  discount: 0
  eur: 11
  usd: 12

Book2:
 author: "BBB"
 author: "CCC"
 author: "DDD"
 title: "secondTitle"

Book3:
 title: "thirdTitle"
 price:
  discount: 0
  eur: 11
 price:
  discount: 1
  eur: 11
```

BigQuery | Columnar based storage - Example

R & D Definition

```
Book1:
 author: "AAA"            R: 0, D: 1
 title: "firstTitle"      R: 0, D: 1
 price:
  discount: 0             R: 0, D: 2
  eur: 11                 R: 0, D: 2
  usd: 12                 R: 0, D: 2

Book2:
 author: "BBB"            R: 0, D: 1
 author: "CCC"            R: 1, D: 1
 author: "DDD"            R: 1, D: 1
 title: "secondTitle"     R: 0, D: 1
 (price):
  (discount: null)        R: 0, D: 0
  (eur: null)             R: 0, D: 0
  (usd: null)             R: 0, D: 0

Book3:
 title: "thirdTitle"     R: 0, D: 1
 (author: null)          R: 0, D: 0
 price:
  discount: 0            R: 0, D: 2
  eur: 11                R: 0, D: 2
  (usd: null)            R: 0, D: 1
 price:
  discount: 1            R: 1, D: 2
  eur: 11                R: 1, D: 2
  (usd: null)            R: 1, D: 1
```

Price.Eur column storage

```
compressed value, R, D

11 R: 0, D: 2
NULL R: 0, D: 0
11 R: 0, D: 2
11 R: 1, D: 2
```

BigQuery | Columnar based storage

Value: Stored Value
Repetition (r) the level of the nesting in the field path at which the repetition is happening
Definition (d) how many optional/repeated fields in the field path have been defined.

```
DocId: 10                r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country; }
    optional string Url; }}
```

```
DocId: 20                r2
Links
  Backward: 10
  Backward: 30
  Forward: 80
Name
  Url: 'http://C'
```

DocId

value	r	d
10	0	0
20	0	0

Name.Url

value	r	d
http://A	0	2
http://B	1	2
NULL	1	1
http://C	0	2

Links.Forward

value	r	d
20	0	2
40	1	2
60	1	2
80	0	2

Links.Backward

value	r	d
NULL	0	1
10	0	2
30	1	2

Name.Language.Code

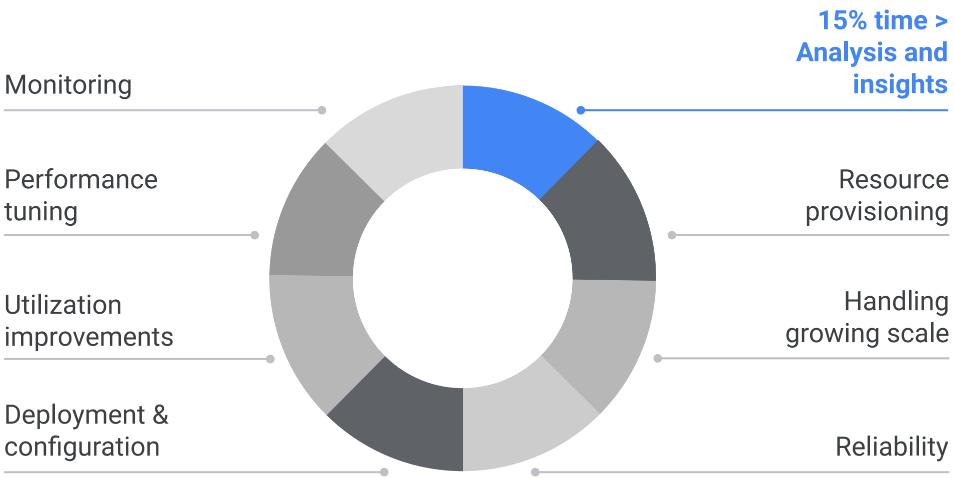
value	r	d
en-us	0	2
en	2	2
NULL	1	1
en-gb	1	2
NULL	0	1

Name.Language.Country

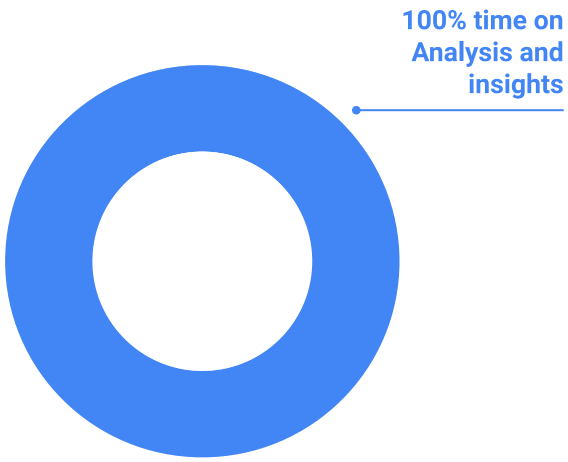
value	r	d
us	0	3
NULL	2	2
NULL	1	1
gb	1	3
NULL	0	1

BigQuery | Serverless data warehouse

Traditional data warehouses



BigQuery’s serverless analytics



BigQuery | Serverless data warehouse - Example

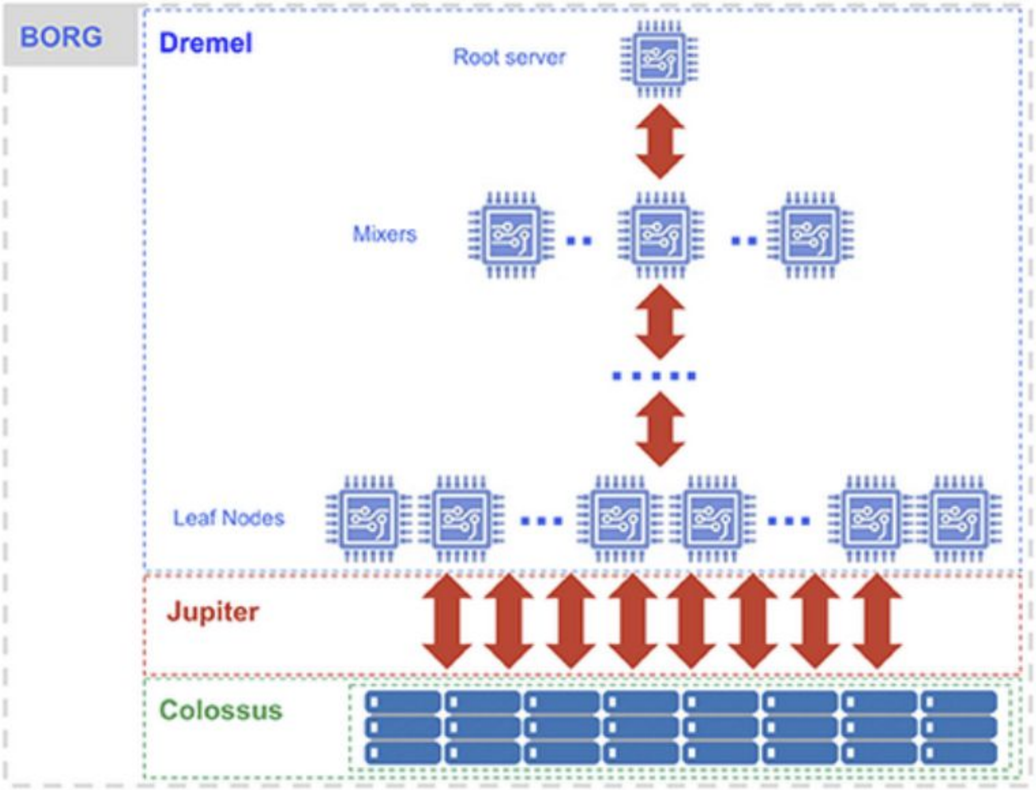
Workers responsible for aggregations

Network to Move Data leaf nodes to mixers

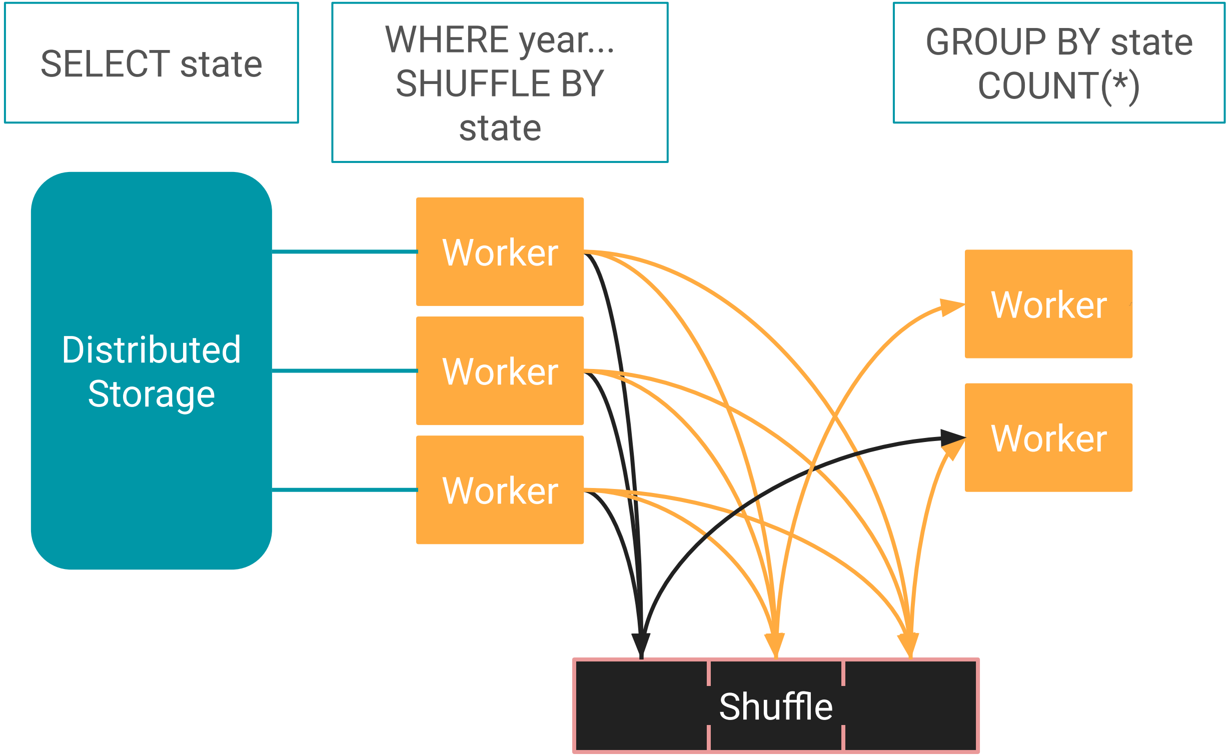
Workers that read and compute

Network to Move Data from Colossus to workers

Distributed Columnar Storage System



BigQuery | Serverless data warehouse - Example



Shuffle does not block future stages

BigQuery uses dynamic partitioning to distribute shuffle optimally

Substantial key-skew can still impact performance

Agenda

01 Introduccion a BigQuery

02 Conceptos básicos de Standard SQL

Break (🎉🎉)

03 DataStudio como herramienta de Visualización

04 Ejercicio práctico BigQuery y DataStudio

05 BigQuery GeoViz

Fin (🎉🎉)

¿Cuál es tu conocimiento de SQL?

Standard SQL | Query Basics

SELECT FROM

```
SELECT column
FROM `project.dataset.table`
```

Standard SQL | Query Basics

SELECT FROM

select origin **from** `chrome-ux-report.country_es.201907`

Standard SQL | Query Basics

Add limit clause to limit results

select origin **from** `chrome-ux-report.country_es.201907` **limit** 10

Standard SQL | Query Basics

Add order by clause to sort results

select origin **from** `chrome-ux-report.country_es.201907` **order by** origin **desc limit** 10

Standard SQL | Query Basics

Filter using Where clause

select origin **from**
`chrome-ux-report.country_es.201907`
where regexp_contains(origin,'bbva')
order by origin **desc limit** 10

Standard SQL | Intro to Functions

Cast Functions ([link]) to change date type

select cast(fare as string) **as** castedfare **from** `bigquery-public-data.chicago_taxi_trips.taxi_trips` **limit** 10

Standard SQL | Intro to Functions

Date Functions (link)

SELECT CURRENT_DATE() as the_date; ⟶ Current Date
SELECT EXTRACT(WEEK from CURRENT_DATE()) as the_date_day; ⟶ Extract Week
SELECT EXTRACT(WEEK(SUNDAY) from CURRENT_DATE()) as the_date_day; ⟶ Extract Week starting on sunday
SELECT
date(trip_start_timestamp,"Europe/Madrid") as date
from `bigquery-public-data.chicago_taxi_trips.taxi_trips`-->Extract date from a timestamp
SELECT DATE_ADD(CURRENT_DATE(), INTERVAL 5 DAY) as five_days_later; ⟶ Add dates
SELECT DATE_SUB(CURRENT_DATE(), INTERVAL 5 DAY) as five_days_ago; ⟶ substract dates
SELECT DATE_DIFF('2017-12-30', '2014-12-30', YEAR) AS year_diff;
SELECT DATE_DIFF('2017-12-30', '2014-12-30', DAY) AS day_diff;
SELECT PARSE_DATE("%x", "12/25/08") as parsed; ⟶ parse date 2008-12-25
SELECT FORMAT_DATE("%x", DATE "2008-12-25") as US_format; ⟶ parse_date 12/25/08
SELECT DATE_TRUNC(DATE '2008-12-25', month) as start_of_month;
SELECT TIMESTAMP_MILLIS(1230219000000) as timestamp; ⟶ change miliseconds to timestamp

Standard SQL | Intro to Functions

String Functions (link)

SELECT cast(CONCAT('1','2') as float64) as concated; ⟶ Concatenate string and convert to float

SELECT lower("APPLE") as lowered; ⟶ Lower capital letters

Standard SQL | Intro to Functions

Aggregation Functions ([link]())

```
SELECT sum(totrevenue) as revenue ,
    avg(totrevenue) as avg_revenue ,
    round(avg(totrevenue),2) as avg_revenue_round ,
    count(ein) as nonprofits,
    count(distinct ein) as nonprofitsdistinct,
    count(*) as total
    from `bigquery-public-data.irs_990.irs_990_2016`;
```

Standard SQL | Intro to Functions

Aggregation grouping Functions ([link]())

```
SELECT
ein as nonprofit,
sum(totrevenue) as revenue ,
    round(avg(totrevenue),2) as avg_revenue_round ,
    count(ein) as nonprofits,
    count(distinct ein) as nonprofitsdistinct,
    count(*) as total
    from `bigquery-public-data.irs_990.irs_990_2016`
Group by ein;
```

Standard SQL | Intro to Functions

Aggregation grouping Functions ([link]())

```
SELECT
ein as nonprofit,
sum(if(frgnofficecd = "Y",totrevenue,0)) as foreign_revenue,
sum(totrevenue) as revenue ,
    round(avg(totrevenue),2) as avg_revenue_round ,
    count(ein) as nonprofits,
    count(distinct ein) as nonprofitsdistinct,
    count(*) as total
    from `bigquery-public-data.irs_990.irs_990_2016`
Group by ein
```

Standard SQL | Intro to Functions

Statistical Functions (link)

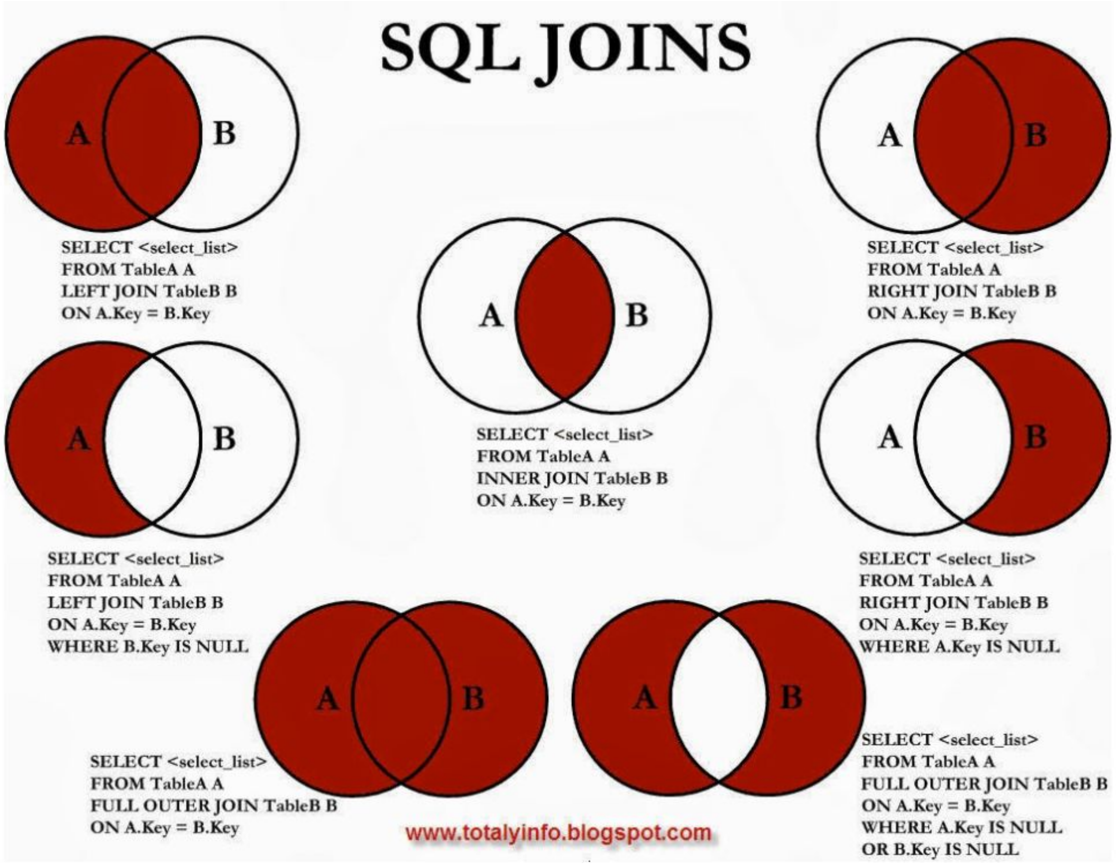
```
SELECT
stddev(noemplyeesw3cnt) as st_dev_employee_count,
avg(noemplyeesw3cnt) as avg_employee_count,
APPROX_QUANTILES(noemplyeesw3cnt, 100)[OFFSET(99)] AS employee_count_percentile_99,
APPROX_QUANTILES(noemplyeesw3cnt, 100)[OFFSET(90)] AS employee_count_percentile_90,
APPROX_QUANTILES(noemplyeesw3cnt, 100)[OFFSET(70)] AS employee_count_percentile_70,
APPROX_QUANTILES(noemplyeesw3cnt, 100)[OFFSET(50)] AS employee_count_percentile_50,
corr( totprgmrevnue, totfuncexpns) as corr_rev_expense,
approx_count_distinct(ein) as approx_nonprofits,
count(distinct ein) as nonprofits
    from `bigquery-public-data.irs_990.irs_990_2016`
    where frgnofficecd = "N"
```

Standard SQL | Intro to Joins

Joins (link)

```
SELECT
t1.ein as ein,
t1.noemplyeesw3cnt as nom_of_employees,
t2.state as state

    from `bigquery-public-data.irs_990.irs_990_2016` as t1
     INNER JOIN
`bigquery-public-data.irs_990.irs_990_ein`as t2
    USING(ein)
```

Standard SQL | Intro to Joins



Standard SQL | Intro to temporary tables

Temporary tables (link)

```
with WY_state as (select ein as filter from
`bigquery-public-data.irs_990.irs_990_ein` where state = "WY")
select
ein as ein,
noemplyeesw3cnt as nom_of_employees

    from `bigquery-public-data.irs_990.irs_990_2016`

    where ein in (select filter from WY_state)
```

Standard SQL | Intro to Functions

Analytical Functions - Rank (link)

firstname	department	startdate
Andrew	1	1/23/1999
Jacob	1	7/11/1990
Daniel	2	6/24/2004
Anna	1	10/7/2001
Pierre	1	2/22/2009
Ruth	2	6/6/1998
Anthony	1	11/29/1995
Isabella	2	9/28/1997
Jose	2	3/17/2013

`PARTITION BY department`

firstname	department	startdate
Andrew	1	1/23/1999
Jacob	1	7/11/1990
Anna	1	10/7/2001
Pierre	1	2/22/2009
Anthony	1	11/29/1995

firstname	department	startdate
Ruth	2	6/6/1998
Daniel	2	6/24/2004
Jose	2	3/17/2013
Isabella	2	9/28/1997

`ORDER BY startdate`

firstname	department	startdate
Jacob	1	7/11/1990
Anthony	1	11/29/1995
Andrew	1	1/23/1999
Anna	1	10/7/2001
Pierre	1	2/22/2009

firstname	department	startdate
Isabella	2	9/28/1997
Daniel	2	6/24/2004
Jose	2	3/17/2013
Ruth	2	6/6/2013

`RANK()`

firstname	department	startdate	rank
Jacob	1	7/11/1990	1
Anthony	1	11/29/1995	2
Andrew	1	1/23/1999	3
Anna	1	10/7/2001	4
Pierre	1	2/22/2009	5

firstname	department	startdate	rank
Isabella	2	9/28/1997	1
Daniel	2	6/24/2004	2
Jose	2	3/17/2013	3
Ruth	2	6/6/2013	4

Standard SQL | Intro to Functions

Analytical Functions - Rank (link)

```
SELECT firstname, department, startdate,
  RANK() OVER ( PARTITION BY department ORDER
BY startdate ) AS rank
FROM Employees;
```

Standard SQL | Intro to Functions

Navigation - Rolling average of 10 last employees (link)

```
SELECT firstname, department, startdate,
  Sum(salary) OVER ( PARTITION BY department
ORDER BY startdate asc ROWS BETWEEN 9
PRECEDING AND 0 FOLLOWING) AS
salary_rolloing_sum
FROM Employees;
```

Standard SQL | Intro to Functions

Navigation functions - Lead or next joined employee (link)

```
SELECT firstname, department, startdate,
  Lead(firstname) OVER ( PARTITION BY department
ORDER BY startdate asc) AS next_employee
FROM Employees;
```

Hands On Exercise K
Analytical functions

Standard SQL | Intro to Functions

Analytical Functions - Rank (link)

Extraer la ONG con más empleados de cada estado

```
with ranked as (

SELECT
t1.ein as ein,
t2.name as name,
t1.noemplyeesw3cnt as nom_of_employees,
t2.state as state,
rank() over (partition by state order by t1.noemplyeesw3cnt desc) as rank
    from `bigquery-public-data.irs_990.irs_990_2015` as t1
     INNER JOIN `bigquery-public-data.irs_990.irs_990_ein`as t2
    USING(ein)
    group by 1,2,3,4
    )

    select * from ranked where rank = 1
    order by nom_of_employees desc
```

Standard SQL | Intro to Functions

Navigation Functions - Rank (link)

Extraer el número de empleados de la siguiente ONG más grande

```
with ranked as (

SELECT
t1.ein as ein,
t2.name as name,
t1.noemplyeesw3cnt as nom_of_employees,
t2.state as state,
rank() over (partition by state order by t1.noemplyeesw3cnt desc) as rank,

lead(t1.noemplyeesw3cnt,1) over (partition by state order by t1.noemplyeesw3cnt desc) as next_num_employees
    from `bigquery-public-data.irs_990.irs_990_2015` as t1
     INNER JOIN `bigquery-public-data.irs_990.irs_990_ein`as t2
    USING(ein)
    group by 1,2,3,4
    )

    select * from ranked where state = 'CA' and rank = 1
    order by nom_of_employees desc
```

Standard SQL | Intro to Functions

Navigation Functions - Rank (link)

Extraer el número medio de empleados de las top 10 ongs

```
with ranked as (
SELECT
t1.ein as ein,
t2.name as name,
t1.noemplyeesw3cnt as nom_of_employees,
t2.state as state,
rank() over (partition by state order by t1.noemplyeesw3cnt desc) as rank,
lead(t1.noemplyeesw3cnt,1) over (partition by state order by t1.noemplyeesw3cnt desc) as next_num_employees,
Sum(t1.noemplyeesw3cnt) OVER ( PARTITION BY state ORDER BY t1.noemplyeesw3cnt asc  ROWS BETWEEN 9 PRECEDING AN
FOLLOWING)/10 AS employee_rolloing_sum
    from `bigquery-public-data.irs_990.irs_990_2015` as t1
     INNER JOIN `bigquery-public-data.irs_990.irs_990_ein`as t2
    USING(ein)
    group by 1,2,3,4
    )
    select * from ranked --where state = 'CA' --
    where rank = 1
    order by nom_of_employees desc
```

Standard SQL | Arrays & structs

Arrays (link)

Arrays are **ordered lists** of zero or more data values that must have the **same data type**

Standard SQL | Arrays & structs

Arrays (link)

select ['a','b','c'] as array_sample

Row	array_sample
1	a
	b
	c

with sample as (select ['a','b','c'] as array_sample)
select array_length(array_sample) as array_sample_length from sample

Row	array_sample_length
1	3

select ['a','b','c'] as array_sample, 'field' as field → BigQuery Creates Nested Field Structures

Results Details

Row	array_sample	field
1	a	field
	b	
	c	

Standard SQL | Arrays & structs

Unnest (link)

with table as (select ['a','b','c'] as array_sample, 'field' as field)

select array_sample,field from table,unnest(array_sample) as array_sample

Row	array_sample	field
1	a	field
2	b	field
3	c	field

Create array (link)

with table as (select 'a' as field union all select 'b' as field union all select 'c' as field)

select array_agg(field order by field desc) as array_created from table

Row	array_created
1	a
	b
	c

Standard SQL | Arrays & structs

Structs (link)

STRUCT are a container of ordered fields each with a type (required) and field name (optional).

You can store multiple data types in a STRUCT (even Arrays!)

Standard SQL | Arrays & structs

Structs (link)

select struct(35 as age, ['alicia','pedro'] as names) as info

Row	info.age	info.names
1	35	alicia
		pedro

select struct(35 as age, 'pedro' as names, ['p1','p2','p3'] as products) as info

Row	info.age	info.names	info.products
1	35	pedro	p1
			p2
			p3

Arrays of Structs:
select

[struct(35 as age, 'pedro' as names, ['p1','p2','p3'] as products), struct(30 as age, 'maria' as names, ['p1','p6','p8'] as products)] as info

Row	info.age	info.names	info.products
1	35	pedro	p1
			p2
			p3
	30	maria	p1
			p6
			p8

Standard SQL | Arrays & structs

Arrays & Structs - filter customers that bought p1 (link)

```
with table as (

select

[struct(35 as age, 'pedro' as names, ['p1','p2','p3'] as products),
struct(30 as age, 'maria' as names, ['p1','p6','p8'] as products),
struct(37 as age, 'juan' as names, ['p2','p7','p9'] as products)
] as info)
select
names
from table
, unnest(info) as info
where 'p1' in unnest(info.products)
```

Standard SQL | Declare variables

Declare and set variables (link)

```
DECLARE target_word STRING DEFAULT 'bespoke';
DECLARE corpus_count, num_palabra INT64;

SET (corpus_count, num_palabra) = (
  SELECT AS STRUCT COUNT(DISTINCT corpus), SUM(word_count)
  FROM `bigquery-public-data`.samples.shakespeare
  WHERE LOWER(word) = target_word
);

SELECT
  FORMAT('Found %d occurrences of "%s" across %d Shakespeare works',
        num_palabra, target_word, corpus_count) AS result;
```

Standard SQL | Run various ordered scripts

Run various scripts (link)

```
DECLARE x INT64 DEFAULT 10;
BEGIN
  DECLARE y INT64;
  SET y = x;
  SELECT y;

SELECT x;

END;
```

Standard SQL | if conditions

Run with if conditions (link)

```
DECLARE target_product_id INT64 DEFAULT 3;

IF EXISTS (

with products as ( select product_id,product_name from (select 1  as product_id, 'a' as
product_name UNION ALL
select 2  as product_id, 'b' as product_name UNION ALL
select 3  as product_id, 'c' as product_name ) )
SELECT target_product_id FROM products
        WHERE product_id = target_product_id) THEN
  SELECT CONCAT('found product ', CAST(target_product_id AS STRING));
ELSE
  SELECT CONCAT('did not find product ', CAST(target_product_id AS STRING));
END IF;
```

Standard SQL | Loops

Loops - Create loops in BigQuery (link)

```
DECLARE x INT64 DEFAULT 0;
LOOP
  SET x = x + 1;
  IF x >= 10 THEN
    LEAVE;
  END IF;
END LOOP;
SELECT x;
```

Agenda

01 Introduccion a BigQuery

02 Conceptos básicos de Standard SQL

Break (🎉🎉)

03 DataStudio como herramienta de Visualización

04 Ejercicio práctico BigQuery y DataStudio

05 BigQuery GeoViz

Fin (🎉🎉)

Agenda

Data Studio: Google's first* BI/Reporting tool available externally

Connect to all your data

Visualize with beautiful, informative reports

Share across the organization and around the world

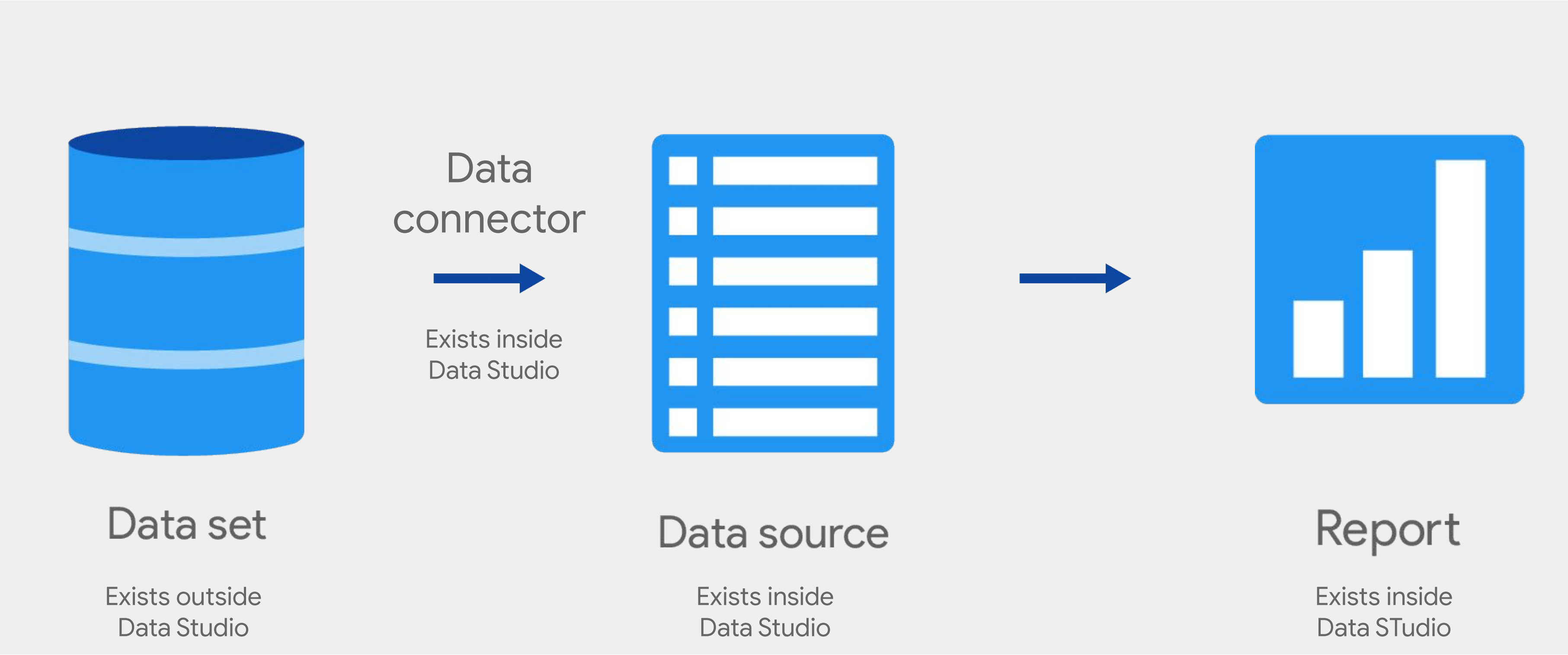
*Google just acquired Looker, but will continue to invest in Data Studio.

Why Data Studio?

Data Studio provides tools to create **beautiful reports** & perform **powerful ad-hoc analysis**.

PROS	CONS
• Large number of data connectors • Easy to maintain • New solution(ish) - constantly developed • Easy to use • Tech skills not mandatory • Easy sharing and collaboration • Free & globally available	• New(ish) solution - constantly developed • Limited formal support • (Somewhat) limited ability to customise (vs. other tools like Tableau)

Data entity relationship

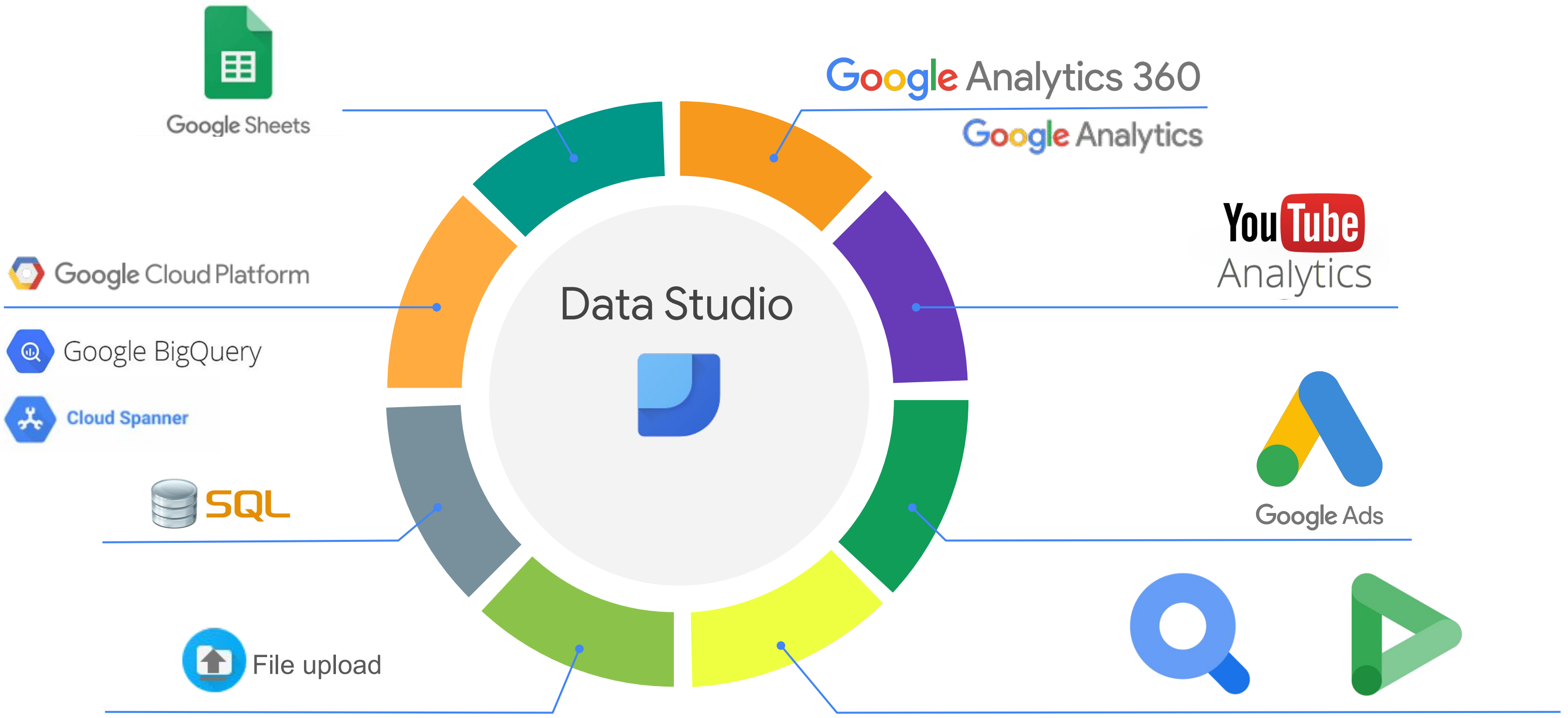


Data sources connect to underlying data sets

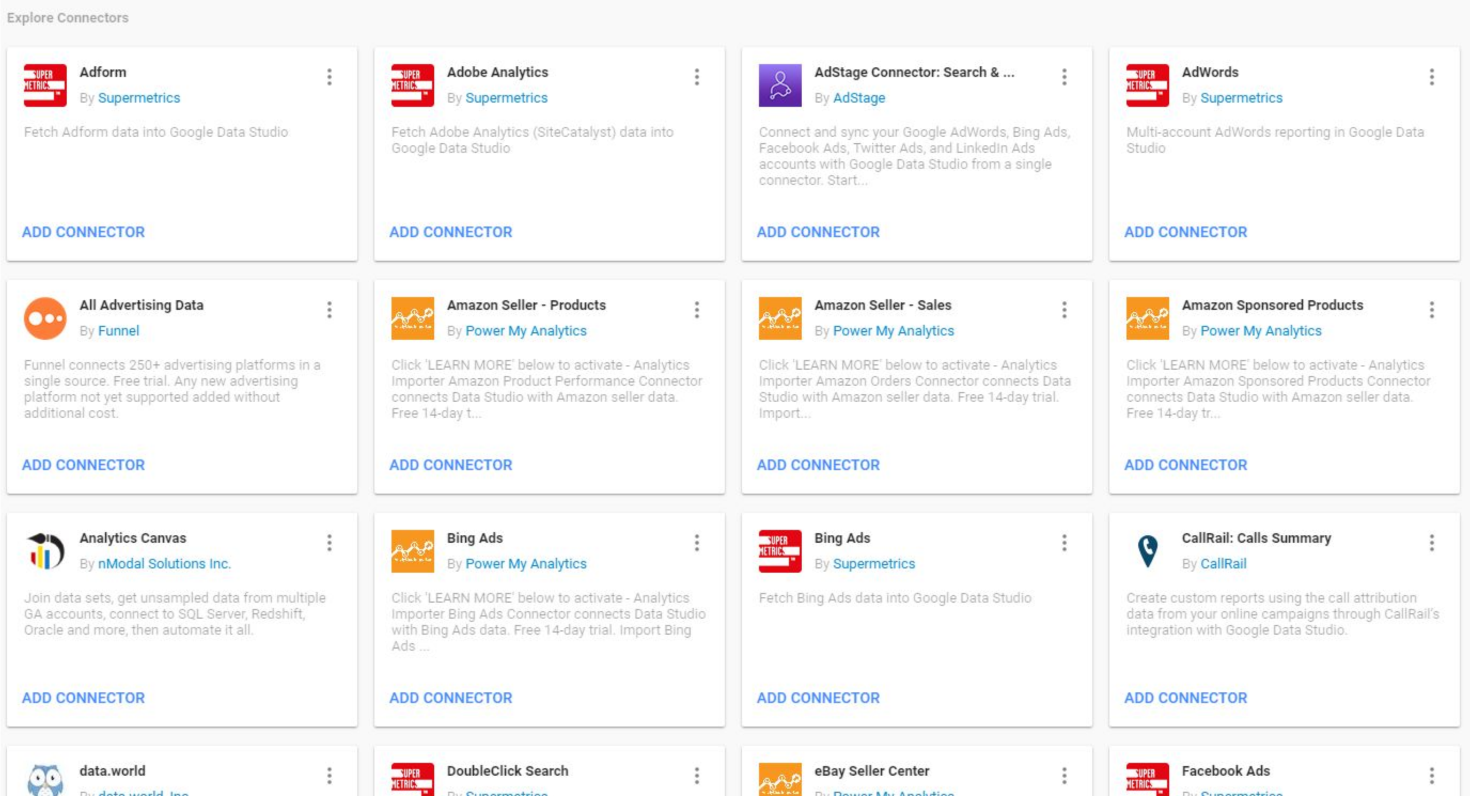
2 basic types of data sets:

Fixed schema	Flexible schema
We understand the data before we ingest it	No idea what the data is before we ingest it
• Google Ads • Search Ads 360 • Google Analytics • Search Console • YouTube Analytics • Display & Video 360	• BigQuery • File upload (CSV) • Google Cloud Storage • Google Sheets • SQL connectors (MySQL, PostgreSQL)

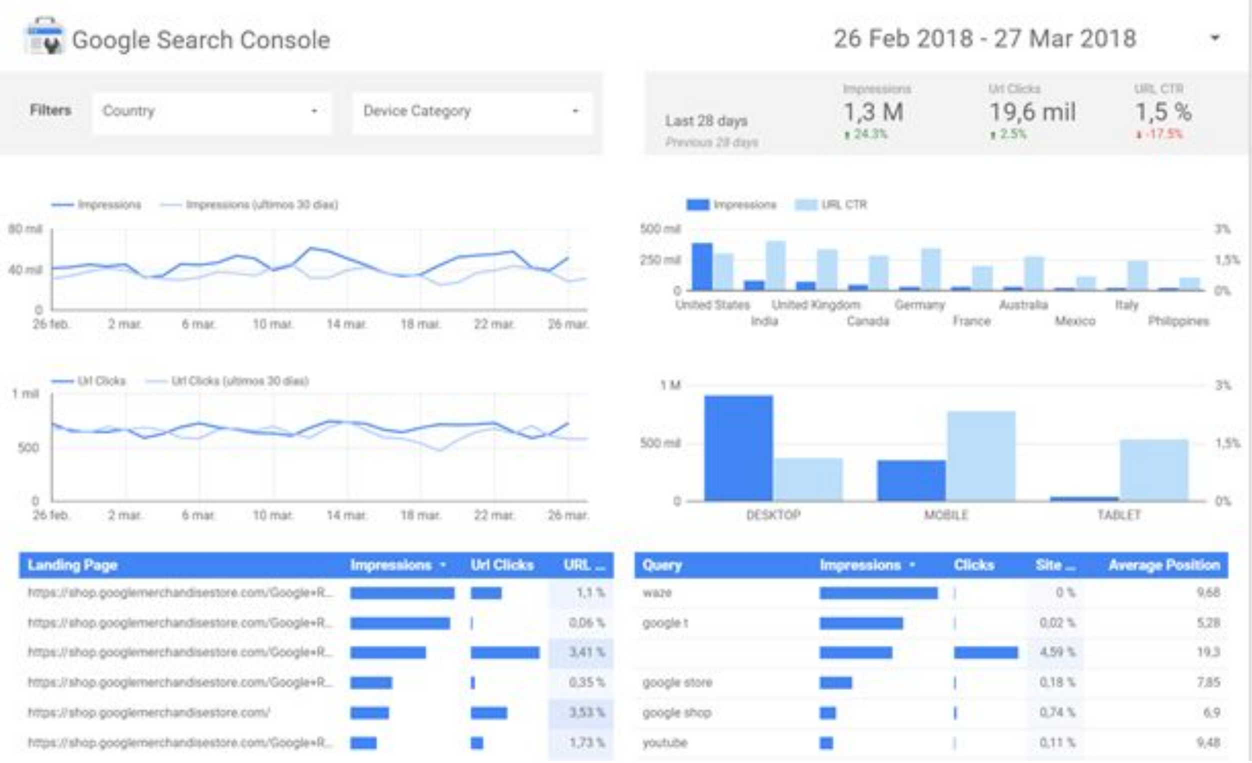
Data Studio connectors



... and more!

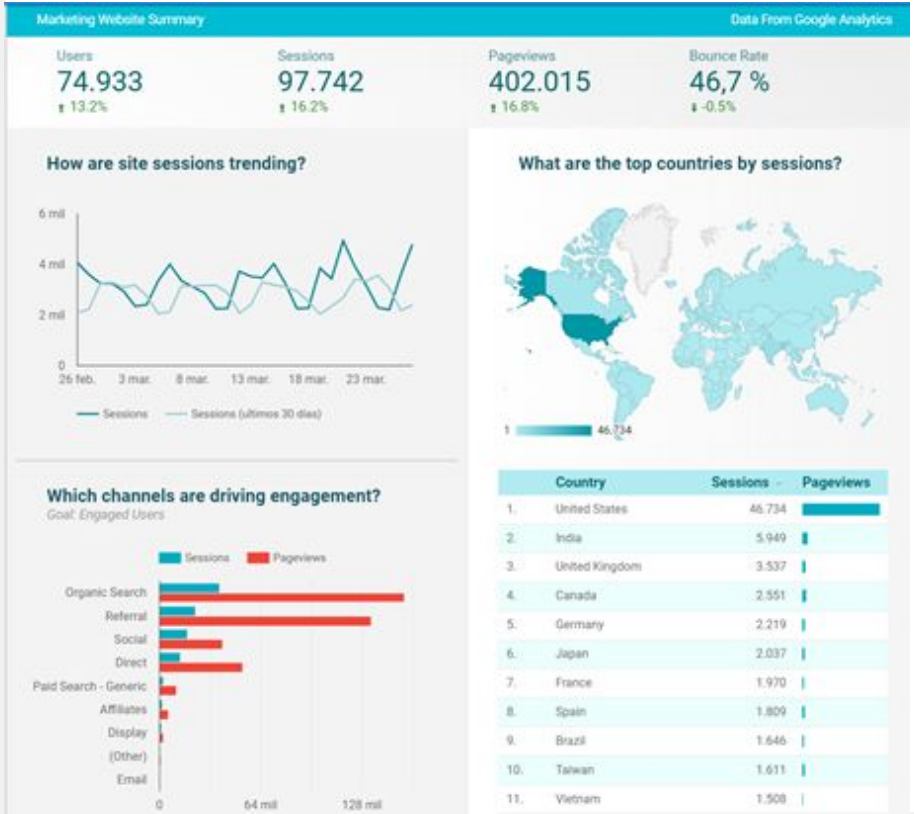

Explore Connectors
Adform
By Supermetrics
Fetch Adform data into Google Data Studio
ADD CONNECTOR
Adobe Analytics
By Supermetrics
Fetch Adobe Analytics (SiteCatalyst) data into Google Data Studio
ADD CONNECTOR
AdStage Connector: Search & ...
By AdStage
Connect and sync your Google AdWords, Bing Ads, Facebook Ads, Twitter Ads, and LinkedIn Ads accounts with Google Data Studio from a single connector. Start...
ADD CONNECTOR
AdWords
By Supermetrics
Multi-account AdWords reporting in Google Data Studio
ADD CONNECTOR
All Advertising Data
By Funnel
Funnel connects 250+ advertising platforms in a single source. Free trial. Any new advertising platform not yet supported added without additional cost.
ADD CONNECTOR
Amazon Seller - Products
By Power My Analytics
Click 'LEARN MORE' below to activate - Analytics Importer Amazon Product Performance Connector connects Data Studio with Amazon seller data. Free 14-day t...
ADD CONNECTOR
Amazon Seller - Sales
By Power My Analytics
Click 'LEARN MORE' below to activate - Analytics Importer Amazon Orders Connector connects Data Studio with Amazon seller data. Free 14-day trial. Import...
ADD CONNECTOR
Amazon Sponsored Products
By Power My Analytics
Click 'LEARN MORE' below to activate - Analytics Importer Amazon Sponsored Products Connector connects Data Studio with Amazon seller data. Free 14-day tr...
ADD CONNECTOR
Analytics Canvas
By nModal Solutions Inc.
Join data sets, get unsampled data from multiple GA accounts, connect to SQL Server, Redshift, Oracle and more, then automate it all.
ADD CONNECTOR
Bing Ads
By Power My Analytics
Click 'LEARN MORE' below to activate - Analytics Importer Bing Ads Connector connects Data Studio with Bing Ads data. Free 14-day trial. Import Bing Ads ...
ADD CONNECTOR
Bing Ads
By Supermetrics
Fetch Bing Ads data into Google Data Studio
ADD CONNECTOR
CallRail: Calls Summary
By CallRail
Create custom reports using the call attribution data from your online campaigns through CallRail's integration with Google Data Studio.
ADD CONNECTOR
data.world
DoubleClick Search
eBay Seller Center
Facebook Ads

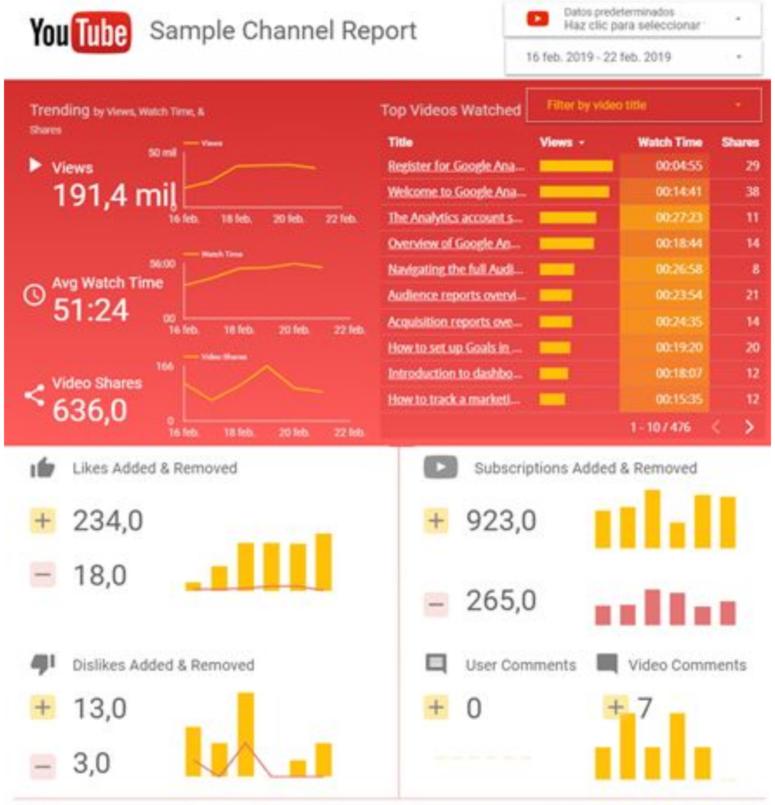
examples



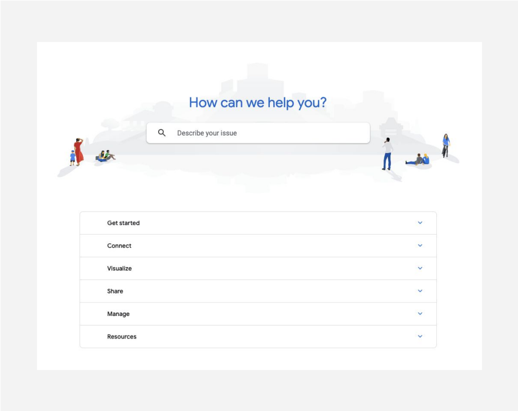
Google Search Console
26 Feb 2018 - 27 Mar 2018
Filters
Country
Device Category
Last 28 days
Previous 28 days
Impressions
1,3 M
24.3%
Url Clicks
19,6 mil
2.5%
URL CTR
1,5 %
-17.5%
Impressions
Impressions (ultimos 30 dias)
Url Clicks
Url Clicks (ultimos 30 dias)
URL CTR
United States
India
United Kingdom
Canada
Germany
France
Australia
Mexico
Italy
Philippines
DESKTOP
MOBILE
TABLET
Landing Page
Impressions
Url Clicks
URL ...
https://shop.googlemerchandisestore.com/Google+R...
https://shop.googlemerchandisestore.com/
Query
Clicks
Site ...
Average Position
waze
google t
google store
google shop
youtube

examples

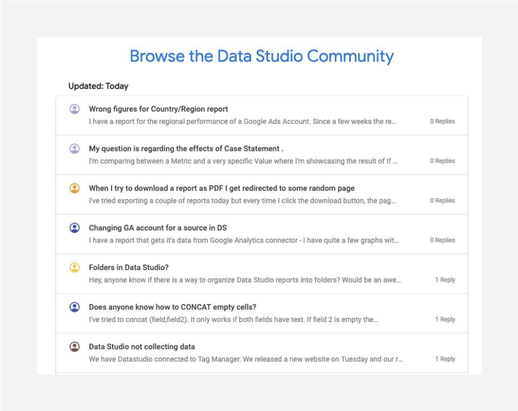

Marketing Website Summary
Data From Google Analytics
Users
74.933
13.2%
Sessions
97.742
16.2%
Pageviews
402.015
16.8%
Bounce Rate
46,7 %
-0.5%
How are site sessions trending?
6 mil
4 mil
2 mil
0
26 feb.
3 mar.
8 mar.
13 mar.
18 mar.
23 mar.
Sessions
Sessions (últimos 30 días)
What are the top countries by sessions?
1
46.734
Which channels are driving engagement?
Goal: Engaged Users
Sessions
Pageviews
Organic Search
Referral
Social
Direct
Paid Search - Generic
Affiliates
Display
(Other)
Email
0
64 mil
128 mil
Country
Sessions
Pageviews
1. United States 46.734
2. India 5.949
3. United Kingdom 3.537
4. Canada 2.551
5. Germany 2.219
6. Japan 2.037
7. France 1.970
8. Spain 1.809
9. Brazil 1.646
10. Taiwan 1.611
11. Vietnam 1.508


YouTube
Sample Channel Report
Datos predeterminados
Haz clic para seleccionar
16 feb. 2019 - 22 feb. 2019
Trending by Views, Watch Time, & Shares
Views
191,4 mil
Avg Watch Time
51:24
Video Shares
636,0
16 feb.
18 feb.
20 feb.
22 feb.
Top Videos Watched
Filter by video title
Title
Views
Watch Time
Shares
Register for Google Ana... 00:04:55 29
Welcome to Google Ana... 00:14:41 38
The Analytics account s... 00:27:23 11
Overview of Google An... 00:18:44 14
Navigating the full Audi... 00:26:58 8
Audience reports overvi... 00:23:54 21
Acquisition reports ove... 00:24:35 14
How to set up Goals in ... 00:19:20 20
Introduction to dashbo... 00:18:07 12
How to track a marketi... 00:15:35 12
1 - 10 / 476
Likes Added & Removed
+ 234,0
− 18,0
Subscriptions Added & Removed
+ 923,0
− 265,0
Dislikes Added & Removed
+ 13,0
− 3,0
User Comments
+ 0
Video Comments
+ 7

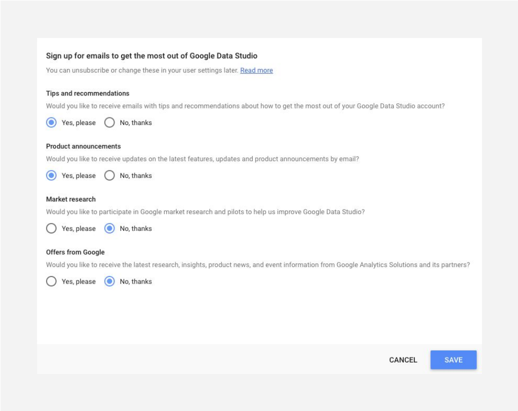
Further Data Studio resources



[Help center]



[Community forums]



Product updates
(opt-in in user settings)

Agenda

01	Introduccion a BigQuery
02	Conceptos básicos de Standard SQL
	Break (🎉🎉)
03	DataStudio como herramienta de Visualización
04	Ejercicio práctico BigQuery y DataStudio
05	BigQuery GeoViz
	Fin (🎉🎉)

Hands On Exercise K
Datastudio

BigQuery + DataStudio Exercise

Upload Raw Data

Manipulate in SQL

Visualize in DataStudio

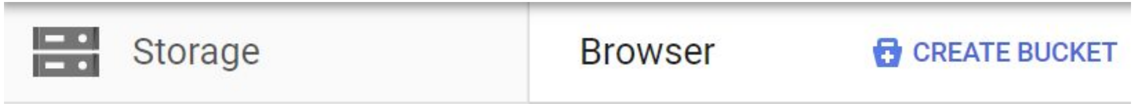
BigQuery + DataStudio Exercise

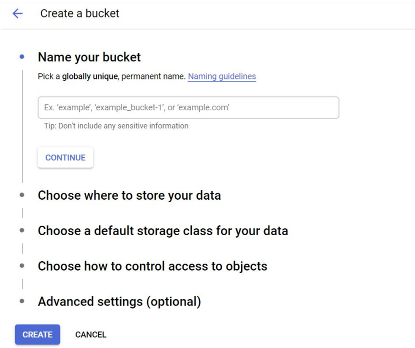
Upload Raw Data

Import Raw Data from Google Cloud Storage

Upload clase_jun_data_clase_2020 and clase_jun_currency_mapping_2020 to Google Cloud Storage

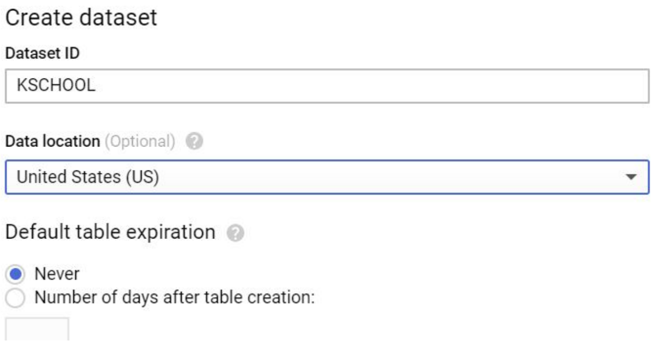
Create a Google Cloud Storage Bucket and upload both files





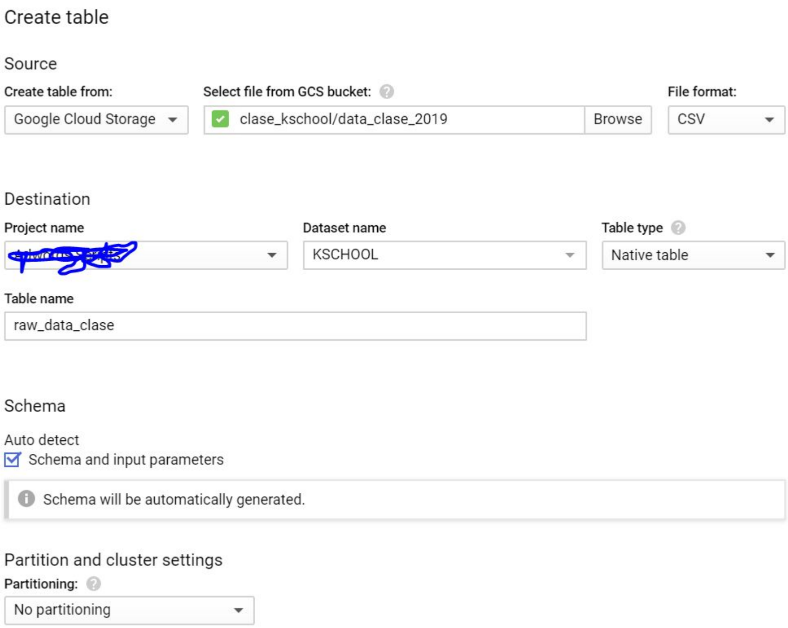
Import Raw Data from Google Cloud Storage

From BigQuery, Create a DataSet with the same region as the Cloud storage bucket



Import Raw Data from Google Cloud Storage

Create a table in the DataSet imported from Cloud Storage



BigQuery + DataStudio Exercise

Manipulate in SQL

BigQuery + DataStudio Exercise

Understand DataSet - Preview

Row	Network	Device	startofmonth	Campaign	CustomerId	Type	clicks	cost	impressions	conversions	ImpressionSum
1	Search Network	Computers	2016-02-01 00:00:00 UTC	CAMP1	C1	Brand	224	1566.3	2078	10	3533.4
2	Search Network	Computers	2016-01-01 00:00:00 UTC	CAMP1	C1	Brand	224	298.06	2422	8	5093.6
3	Search Network	Computers	2015-10-01 00:00:00 UTC	CAMP1	C1	Brand	258	283.8	2596	58	5453.6
4	Search Network	Computers	2016-03-01 00:00:00 UTC	CAMP1	C1	Brand	274	2279.94	2056	34	3437.8
5	Search Network	Computers	2015-08-01 00:00:00 UTC	CAMP1	C1	Brand	220	376.1	2168	84	4713.4
6	Search Network	Computers	2016-04-01 00:00:00 UTC	CAMP1	C1	Brand	126	159.58	1414	8	3206.2
7	Search Network	Computers	2015-05-01 00:00:00 UTC	CAMP1	C1	Brand	312	273.3	3244	40	6705.4
8	Search Network	Computers	2015-09-01 00:00:00 UTC	CAMP1	C1	Brand	224	294.94	2414	32	5188.6

BigQuery + DataStudio Exercise

Understand DataSet - Preview

Network: Search (Google Search Ads) Or Display (Google Display Banner Ads)
Device: Computer, Mobile or tablet where the ad was shown
StartOfMonth: First day of Month timestamp
Campaign: Campaign Associated with the Ad
CustomerId: Billing Customer Id for Google Ads
Type: Type and description of Campaign: Brand / Generic / Audiences
Clicks: number of Clicks driven by the Ads
Cost: Cost Associated to the ads
Impressions: total of impressions shown
Conversion: Goal actions achieved by ads
ImpressionSum = Position of the ad * Impressions

BigQuery + DataStudio Exercise

Create a Table that includes country, exchange rate and metrics to calculate YoY of 2016 Vs 2015

BigQuery + DataStudio Exercise

Solution

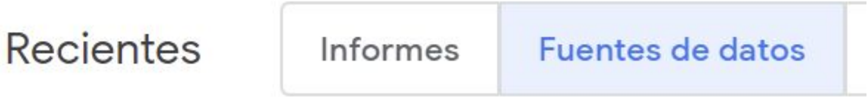
```
select
country, month,tipocambio,network,Device,type,campaign,currency, sum(clicks) as clicks, sum(cost)/tipocambio as costeur,sum(impressions) as impressions,
sum(conversions) as conversions,sum(impressionSum) as impressionSum,sum(cost2015)/tipocambio as cost2015,sum(cost2016)/tipocambio as cost2016,
sum(conversions2015) as conversions2015,sum(conversions2016) as conversions2016,sum(impressions2015) as impressions2015,sum(impressions2016) as
impressions2016,
sum(ImpressionSum2015) as ImpressionSum2015,sum(ImpressionSum2016) as ImpressionSum2016
 from (
select
t2.country as country,t2.TC as tipocambio,  t1.startofmonth as month, t1.Network as network, t1.Device as Device,
      t1.Type as type, t1.Campaign as campaign, t2.Currency as currency,
      sum(t1.clicks) as clicks, sum(t1.cost) as cost, sum(t1.impressions) as impressions, sum(t1.conversions) as conversions,
      sum(t1.ImpressionSum) as ImpressionSum,
      sum(if(extract(year from t1.startofmonth )= 2016,t1.cost,0)) as cost2016,
      sum(if(extract(year from t1.startofmonth )= 2015,t1.cost,0)) as cost2015,
      sum(if(extract(year from t1.startofmonth )= 2016,t1.conversions,0)) as conversions2016,
      sum(if(extract(year from t1.startofmonth )= 2015,t1.conversions,0)) as conversions2015,
      sum(if(extract(year from t1.startofmonth )= 2016,t1.impressions,0)) as impressions2016,
      sum(if(extract(year from t1.startofmonth )= 2015,t1.impressions,0)) as impressions2015,
      sum(if(extract(year from t1.startofmonth )= 2016,t1.ImpressionSum,0)) as ImpressionSum2016,
      sum(if(extract(year from t1.startofmonth )= 2015,t1.ImpressionSum,0)) as ImpressionSum2015
        from CLASE2.RAW as t1 INNER JOIN CLASE2.Mapping as t2
        on t1.CustomerId = t2.Accountid
        group by 1,2,3,4,5,6,7,8)
        group by 1,2,3,4,5,6,7,8
```

BigQuery + DataStudio Exercise

Visualize in DataStudio

BigQuery + DataStudio Exercise

Create DataSource in DataStudio from saved BigQuery Table



Select BigQuery as source

Select the table and click connect

BigQuery + DataStudio Exercise

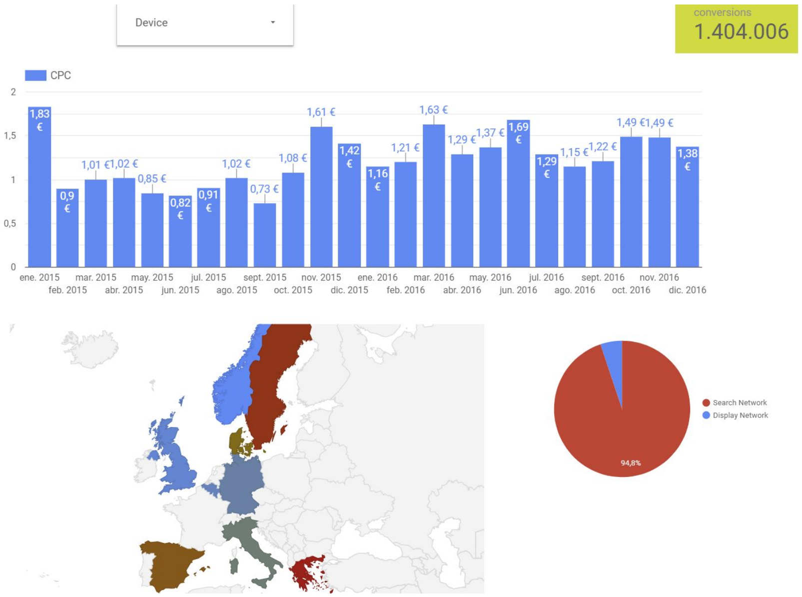
Adapt Fields and create calculated fields

EDITAR LA CONEXIÓN

Índice	Campo	Tipo	Agregación
1	Record Count	123 Número	Automática
2	country	País	Ninguna
3	month	Fecha (DDMMAAAA)	Ninguna
4	tipocambio	123 Número	Ninguna
5	network	ABC Texto	Ninguna
6	Device	ABC Texto	Ninguna
7	type	ABC Texto	Ninguna
8	campaign	ABC Texto	Ninguna
9	currency	ABC Texto	Ninguna
10	clicks	123 Número	Ninguna
11	costeur	123 Número	Ninguna

BigQuery + DataStudio Exercise

Create Report from DataSource



BigQuery + DataStudio Exercise

Share the report to collaborate or to share Visualizations

Action	Public	Can view	Can edit	Is owner
View data in the report (depending on data source credentials)	x	x	x	x
Copy the report		x	x	x
Prevent report copying				x
Share the report with others		x	x	x
Prevent report sharing				x
Modify the report			x	x
Use and modify data from added data sources			x	x
Add / remove data sources			x	x
Create / delete the report				x
Download data from the report	x	x	x	x
Prevent downloading data				x

Hands On Exercise K Means BQML

RFM model Classic but efficient

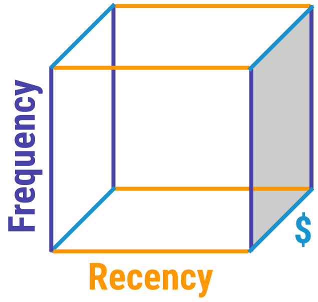
CRM extract example

Order date	Customer ID	Order Value
01-02-15	177237	$140
02-02-15	177323	$30
01-03-15	177237	$35

Recency and Frequency of purchases at user level

Monetary Value

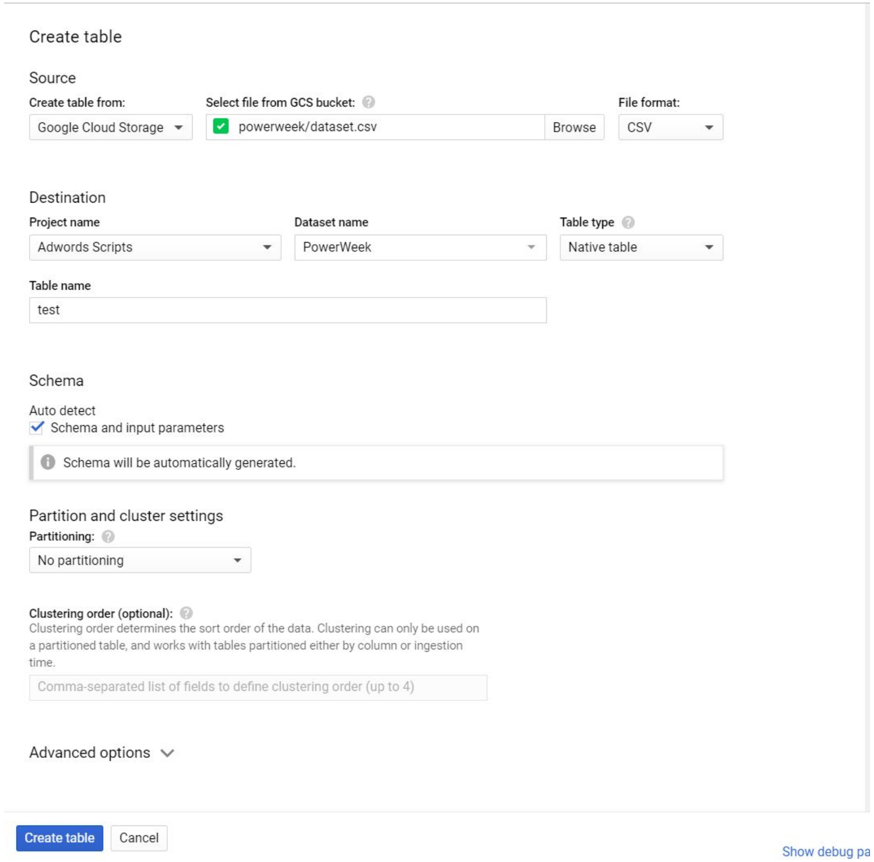
Customers scored with a LTV (RFM) model



Instructions

1. Upload CSV File into BigQuery about customer orders
2. Calculate in SQL Frequency, Recency, Monetization, average basket, average items, average time between purchases (Excluding returns).(Max date in dataset 2011-12-09)
3. Create a clustering using BigQuery Kmeans and decide strategy per cluster

Hands on exercise - Upload CSV to BQ



Hands on exercise - Transform Data to obtain RFM

```
SELECT

    cast(customer_id as string ) as id,
    sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,
    sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,
    sum(order_qty_articles )/count(distinct if(order_value >0,order_date,null))  as avg_items,
    date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,
    date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct if(order_value
>0,order_date,null)) as time_between

from `clase_feb_2021.orders`
group by 1
```

Row	id	monetary	frequency	avg_basket	avg_items	recency	time_between
1	12592	437.6	2	218.8	145.0	92	60.5
2	12399	1108.65	4	277.1625	294.5	119	35.5
3	12684	2283.63	7	326.23285714285714	283.57142857142856	7	29.428571428571427
4	12712	4241.630000000001	13	326.2792307692308	80.3076923076923	28	26.384615384615383

Hands on exercise - Kmeans Clustering

```
CREATE OR REPLACE MODEL clase_feb_2021.kmeans_model_5clusters OPTIONS(
  model_type='kmeans', num_clusters=5, distance_type='euclidean') AS
(
SELECT

   cast(customer_id as string ) as id,
   sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,
   sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,
   sum(order_qty_articles )/count(distinct if(order_value >0,order_date,null))  as avg_items,
   date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,
   date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct if(order_value
>0,order_date,null)) as time_between

from `clase_feb_2021.orders`
group by 1)
```

Hands on exercise - Define Strategy Per Cluster

Centroid Id	Count	monetary	frequency	avg_basket	avg_items_dnn	recency	time_between
1	410	881.6812	2.7390	321.9702	182.4925	193.7000	36.7018
3	1487	1,907.5553	4.8554	399.0776	246.4978	31.2771	38.1896
5	219	9,410.0584	20.0000	489.6593	283.8831	9.7580	20.1926
2	30	62,189.7347	19.4333	3,396.0492	2,177.0158	26.8333	28.4400
4	631	935.1316	2.7464	341.0379	207.1088	44.0951	98.2244

What would be the best number of cluster? Check next slide for code

```sql
DECLARE NUM_CLUSTERS INT64 DEFAULT 3; DECLARE MIN_ERROR FLOAT64 DEFAULT 1000.0; DECLARE BEST_NUM_CLUSTERS INT64 DEFAULT -1;
DECLARE MODEL_NAME STRING;
DECLARE error FLOAT64 DEFAULT 0;
WHILE NUM_CLUSTERS < 10 DO
SET MODEL_NAME = CONCAT('clase_feb_2021.kmeans_model_numcluster_', CAST(NUM_CLUSTERS AS STRING));
EXECUTE IMMEDIATE format("""
  CREATE OR REPLACE MODEL %s
    OPTIONS(model_type='kmeans',
            num_clusters=%d,
            standardize_features = true)
  AS SELECT
    cast(customer_id as string ) as id,
    sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,
    sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,
    sum(order_qty_articles )/count(distinct if(order_value >0,order_date,null))  as avg_items,
    date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,
    date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct
if(order_value >0,order_date,null)) as time_between
from `clase_feb_2021.orders`
group by 1;  """, MODEL_NAME, NUM_CLUSTERS);
EXECUTE IMMEDIATE format("""
    SELECT davies_bouldin_index FROM ML.EVALUATE(MODEL %s);
  """, MODEL_NAME) INTO error;
IF error < MIN_ERROR THEN SET MIN_ERROR = error;
SET BEST_NUM_CLUSTERS = NUM_CLUSTERS;
END IF;
SET NUM_CLUSTERS = NUM_CLUSTERS + 1;
END WHILE
```

Hands on exercise - Define Strategy Per Cluster

Centroid Id	Count	monetary	frequency	avg_basket	avg_items_dnn	recency	time_between
1	410	881.6812	2.7390	321.9702	182.4925	193.7000	36.7018
3	1487	1,907.5553	4.8554	399.0776	246.4978	31.2771	38.1896
5	219	9,410.0584	20.0000	489.6593	283.8831	9.7580	20.1926
2	30	62,189.7347	19.4333	3,396.0492	2,177.0158	26.8333	28.4400
4	631	935.1316	2.7464	341.0379	207.1088	44.0951	98.2244

2 **Top Customers (B2B) - Ad hoc offers and proactive approach**

5 **Top B2C Customers - Retain and Similar Audiences**

3 **New Potential Customers - Boost through coupons**

4 **Occasional Customers - Increase frecuency**

1 **Old Customers - Get them back**

Hands on exercise - Create Buckets to Activate GA

```
SELECT CENTROID_ID,id


FROM ML.PREDICT(MODEL `KSCHOOL.kmeans_model_5clusters`, (select
 cast(customer_id as string ) as id,
  sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,
  sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,
  sum( order_qty_articles)/count(distinct if(order_value >0,order_date,null))  as avg_items,,
  date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,
  date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct if(order_value
>0,order_date,null)) as time_between



 FROM `KSCHOOL.orders_dataset`
 group by 1
 ))
```

Hands On Exercise Recommender system BQML

Modelo Matrix Factorization - Collaborative filtering method

	Película 1	Película 2	Película n
Usuario 1	5	4	5
Usuario 2	1	2	5
Usuario n	5	1	3

Modelo Matrix Factorization - Creando factores que permiten reducir la dimensionalidad

2000 usuarios x 100 features = 200K entradas

1000 películas 100 x 1000 = 100K entradas

Matriz de usuarios y Rating (2000 x 1000 = 2M entradas)

Utilizaremos la tabla de Google Analytics para recomendar productos a posibles clientes (es una tabla nested)

⚠ 200 row per page limit reached due to duplicate values or complex results. Displaying 21 results to reflect this.

gory	hits.eventInfo.eventAction	hits.eventInfo.eventLabel	hits.eventInfo.eventValue	hits.product.productSKU	hits.product.v2ProductName	hits.product.v2ProductCategory	hits.pr
				GGOEGBFC018799	Electronics Accessory Pouch	Home/Electronics/	(not se
				GGOEGESB015199	Google Flashlight	Home/Electronics/	(not se
				GGOEGEVA022399	Micro Wireless Earbud	Home/Electronics/	(not se
				GGOEGCBB074199	Google Car Clip Phone Holder	Home/Electronics/	(not se
				GGOEGFKA022299	Keyboard DOT Sticker	Home/Electronics/	(not se
				GGOEGCBB074399	Google Device Holder Sticky Pad	Home/Electronics/	(not se
				GGOEGCBC074299	Google Device Stand	Home/Electronics/	(not se
				GGOEGEHQ072499	Google 2200mAh Micro Charger	Home/Electronics/	(not se
				GGOEGEHQ072599	Google 4400mAh Power Bank	Home/Electronics/	(not se
				GGOEGESB015099	Basecamp Explorer Powerbank Flashlight	Home/Electronics/	(not se
				GGOEGESC014099	Rocket Flashlight	Home/Electronics/	(not se
				GGOEGESQ016799	Plastic Sliding Flashlight	Home/Electronics/	(not se
				GGOEGAAX0313	Google Tri-blend Hoodie Grey	Home/Apparel/Men's/Men's-Outerwear/	(not se
				GGOEGAAX0358	Google Men's Zip Hoodie	Home/Apparel/Men's/Men's-Outerwear/	(not se
				GGOEGAAX0568	Google Men's Watershed Full Zip Hoodie Grey	Home/Apparel/Men's/Men's-Outerwear/	(not se
				GGOEGAAX0592	Google Men's Airflow 1/4 Zip Pullover Black	Home/Apparel/Men's/Men's-Outerwear/	(not se

Al no tener ratings, utilizaremos el tiempo en cada vista detallada de producto - Ejercicio

1. Obtener la suma de duración en tiempo (diferencia hits.time con respecto a la siguiente página vista) en todas las sesiones dedicada por cada usuario (fullvisitorid) en la vista detallada (hits.eCommerceAction.action_type = "2") de cada uno de los productos (hits.product.productSKU)

2. Cada usuario puede tener varias visitas (visitNumber) y varios hits (interacciones), consideramos un usuarion como la concatenacion de fullvisitorid y visitnumber

```sql
CREATE OR REPLACE TABLE DPLACEHOLDER10.aggregate_web_stats AS (
 WITH
  durations AS (
   --calculate pageview durations
   SELECT
    CONCAT(fullvisitorid,'-',
         CAST(visitNumber AS STRING),'-',
         CAST(hitNumber AS STRING) ) AS visitorId_session_hit,
         time as start_of_pageview,
    LEAD(time, 1) OVER (
     PARTITION BY CONCAT(fullvisitorid,'-',CAST(visitNumber AS STRING))
     ORDER BY
     time ASC ) - time AS pageview_duration,
     LEAD(time, 1) OVER (
     PARTITION BY CONCAT(fullvisitorid,'-',CAST(visitNumber AS STRING))
     ORDER BY
     time ASC ) as end_of_pageview_or_next_row,
   FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_201*`,
    UNNEST(hits) AS hit-- where fullvisitorid = '0006911334202687206'
  ),
  prodview_durations AS (
   --filter for product detail pages only  // vistas de detalles de productos
  SELECT
   fullvisitorid AS userId,
     CONCAT(fullvisitorid,'-',
         CAST(visitNumber AS STRING),'-',
         CAST(hitNumber AS STRING) ) AS visitorId_session_hit,
    productSKU AS itemId,
    IFNULL(dur.pageview_duration,
     1) AS pageview_duration,
   FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_201*` t,
    UNNEST(hits) AS hits,
    UNNEST(hits.product) AS hits_product
   LEFT JOIN
    durations dur
   ON
    CONCAT(fullvisitorid,'-',
          CAST(visitNumber AS STRING),'-',
          CAST(hitNumber AS STRING)) = dur.visitorId_session_hit
   WHERE
   eCommerceAction.action_type = "2"-- and  fullvisitorid = '2969418676126258798'
  ),
  aggregate_web_stats AS(
   --sum pageview durations by userId, itemId
   SELECT
    userId,
    itemId,
    SUM(pageview_duration) AS session_duration
   FROM
    prodview_durations
   GROUP BY
    userId,
    itemId )
    select * from aggregate_web_stats)
```

Script creación de la tabla con duración en páginas de producto de cada usuario

Entrenamiento del modelo - Es necesario reservar slots de entrenamiento - link

Entrenar el modelo (link)

```
      CREATE OR REPLACE MODEL
DPLACEHOLDER10.retail_recommender
      OPTIONS(model_type='matrix_factorization',
            user_col='userId',
            item_col='itemId',
            rating_col='session_duration',
            feedback_type='implicit'
            )
      AS
      SELECT * FROM
DPLACEHOLDER10.aggregate_web_stats;
```

Obtener recomendaciones

```
DECLARE MY_USERID STRING DEFAULT "0006911334202687206-1";

SELECT
  *
FROM
  ML.RECOMMEND(MODEL `DPLACEHOLDER10.retail_recommender`,
  (SELECT MY_USERID as userID)
              )
ORDER BY predicted_session_duration_confidence DESC
LIMIT 5;
```

Obtener top 5 recomendaciones por usuario

```sql
WITH predictions AS (
    SELECT
      userId,
      ARRAY_AGG(STRUCT(itemId,
                        predicted_session_duration_confidence)
                ORDER BY
                  predicted_session_duration_confidence DESC
                LIMIT 5) as recommended
    FROM ML.RECOMMEND(MODEL DPLACEHOLDER10.retail_recommender)
    GROUP BY userId
)

SELECT
  userId,
  itemId,
  predicted_session_duration_confidence
FROM
  predictions p,
  UNNEST(recommended)
```

Hands On Exercise time series ARIMA

Analizamos la tabla

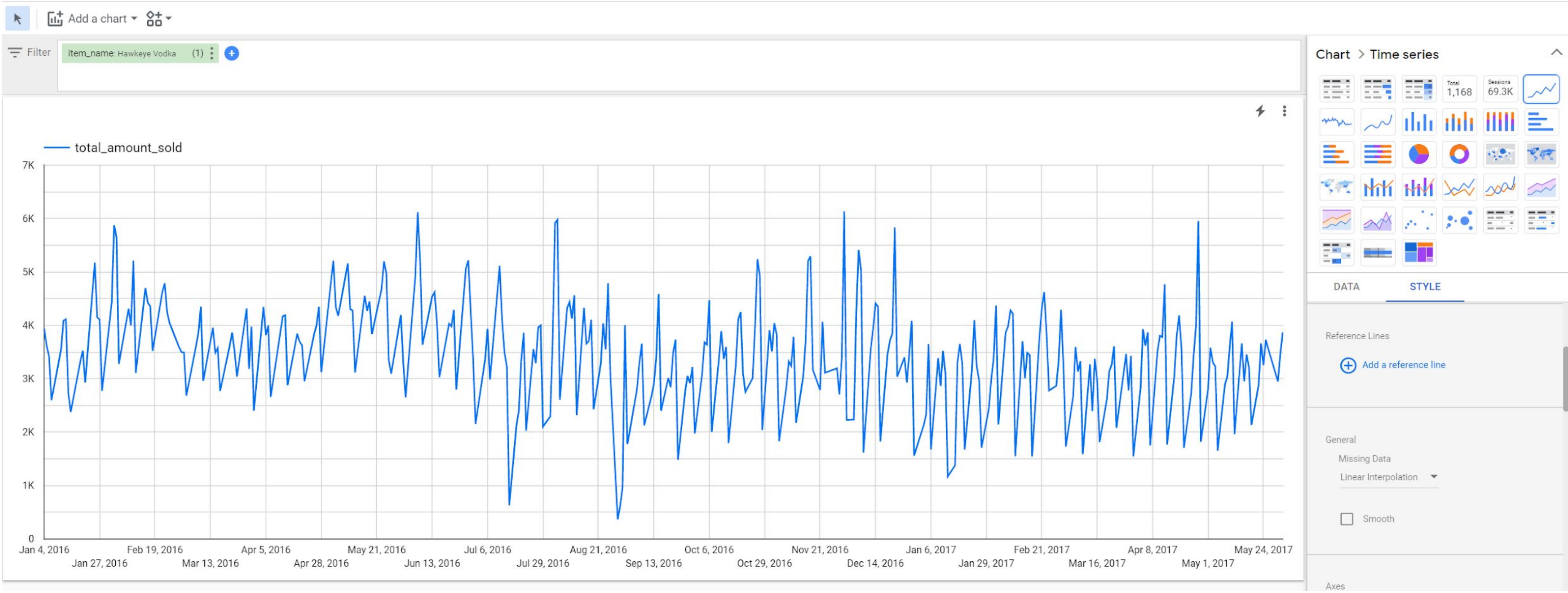
```
SELECT
    invoice_and_item_number,
    date,
    store_number,
    item_description,
    bottles_sold,
    sale_dollars
FROM
  `bigquery-public-data.iowa_liquor_sales.sales`
LIMIT
  5
```

Sacamos un histórico diario de los top 5 productos en ventas

```
CREATE OR REPLACE TABLE DPLACEHOLDER10.training_data AS (
    WITH topsellingitems AS(
         SELECT
            item_description,
            count(item_description) cnt_transactions
        FROM
            `bigquery-public-data.iowa_liquor_sales.sales`
        GROUP BY
            item_description
        ORDER BY cnt_transactions DESC
        LIMIT 5 #Top N
    )
    SELECT
        date,
        item_description AS item_name,
        SUM(bottles_sold) AS total_amount_sold
    FROM
        `bigquery-public-data.iowa_liquor_sales.sales`
    GROUP BY
        date, item_name
    HAVING
        date BETWEEN '2016-01-01' AND '2017-06-01'
        AND item_description IN (SELECT item_description FROM topsellingitems)
    );
```

```
SELECT  * FROM  DPLACEHOLDER10.training_data
```

Ejercicio, Analizar los históricos de ventas en DataStudio. Crear la serie temporal



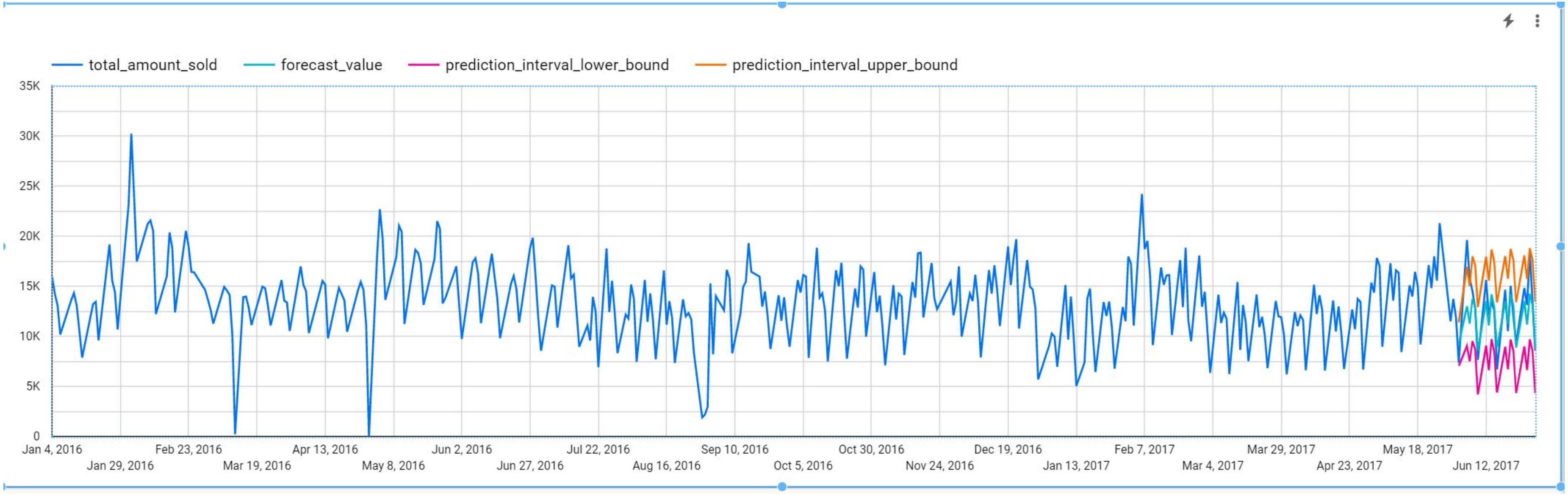
Generamos el modelo de series temporales

```
CREATE OR REPLACE MODEL `DPLACEHOLDER10.arima_model`

OPTIONS(
  MODEL_TYPE='ARIMA',
  TIME_SERIES_TIMESTAMP_COL='date',
  TIME_SERIES_DATA_COL='total_amount_sold',
  TIME_SERIES_ID_COL='item_name',
  HOLIDAY_REGION='US'
) AS

SELECT
    date,
    item_name,
    total_amount_sold
FROM
  `DPLACEHOLDER10.training_data`
```

Generamos la tabla para graficar histórico y predicción

```
CREATE OR REPLACE TABLE clase_feb_2021.predcitions_arima as (
with
topsellingitems AS(
         SELECT item_description,count(item_description) cnt_transactions
        FROM
            `bigquery-public-data.iowa_liquor_sales.sales`
        GROUP BY  item_description   ORDER BY cnt_transactions DESC LIMIT 5
    ),
forecasts as (
SELECT
      extract (date from forecast_timestamp AT TIME ZONE "UTC") as date,
item_name,forecast_timestamp,forecast_value,prediction_interval_lower_bound,prediction_interval_upper_bound
    FROM
      ML.FORECAST(MODEL clase_feb_2021.arima_model,
                  STRUCT(30 AS horizon,
                         0.9 AS confidence_level)
                 )),
 reals as (
SELECT
    Date, item_description AS item_name, SUM(bottles_sold) AS total_amount_sold
        FROM
    `bigquery-public-data.iowa_liquor_sales.sales`
GROUP BY
    date, item_name
HAVING
    date BETWEEN '2016-01-01'
            AND DATE_ADD('2017-06-01',
                             INTERVAL 31 DAY)
ORDER BY date)
    select t1.date as date,t1.item_name as item_name, total_amount_sold,
    forecast_value,prediction_interval_lower_bound,prediction_interval_upper_bound
        from reals as t1
        LEFT JOIN     forecasts as t2 using(date,item_name)
        where item_name in ( select item_description from topsellingitems))
```

Ejercicio, Ejecutar el gráfico en DataStudio



Hands On Exercise
DataStudio COVID

Instructions

1. Utiliza el dataset del COVID-19 (`bigquery-public-data.covid19_cds.global_cases_by_province`) para crear una tabla de BigQuery que tenga los casos diarios y acumulados, defunciones diarias y acumuladas y tasa de mortalidad ajustada (infectados 10 días antes vs defunciones 10 dias despues)

2. Sube la tabla de poblaciones por CCAA

3. Crea un dashboard con los datos en DataStudio

```
with

table_case_less1 as (
select  date,name,cases,deaths,recovered,tested,hospitalized
from `bigquery-public-data.covid19_cds.global_cases_by_province` where country = "Spain" and level = "state"
group by 1,2,3,4,5,6,7 order by 1 asc),

table_cases as (select date as  date,date_sub(date, interval 1 day) as date_less_one,name,cases,deaths,recovered,tested,hospitalized
from `bigquery-public-data.covid19_cds.global_cases_by_province`

where country = "Spain" and level = "state" group by 1,2,3,4,5,6,7,8 order by 1 asc),

base_table as (


select t1.date as date,t1.name as name,t1.cases as cumulated_cases , t2.cases as cumulated_cases_yesterday, t1.cases-if(t2.cases is null,0,t2.cases) as daily_cases,
(t1.cases-if(t2.cases is null,0,t2.cases))/nullif(if(t2.cases is null,0,t2.cases),0) as rate_daily_cases,
t1.deaths as cumulated_deaths , t2.deaths as cumulated_deaths_yesterday, t1.deaths-if(t2.deaths is null,0,t2.deaths) as daily_deaths,
(t1.deaths-if(t2.deaths is null,0,t2.deaths))/nullif(if(t2.deaths is null,0,t2.deaths),0) as rate_daily_deaths,
t1.recovered as cumulated_recovered , t2.recovered as cumulated_recovered_yesterday, t1.recovered-if(t2.recovered is null,0,t2.recovered) as daily_recovered,
from table_cases as t1

left join table_case_less1 as t2 on t2.date = t1.date_less_one and t1.name=t2.name

order by t1.date desc)

select date,t1.name as name,split(name,",")[offset(0)] as Geo,
cumulated_cases,daily_cases,rate_daily_cases,population,(daily_cases/population)*100000 as rate_daily_cases_per100mil,(cumulated_cases/population)*100000 as rate_cumulated_cases_per100mil
cumulated_deaths,daily_deaths,rate_daily_deaths,(daily_deaths/population)*100000 as rate_daily_deaths_per100mil,(cumulated_deaths/population)*100000 as rate_cumulated_deaths_per100mil,
cumulated_recovered,daily_recovered,
sum(daily_cases) OVER (partition by name order by date asc ROWS BETWEEN 9 PRECEDING AND 0 FOLLOWING) as moving_cases_pre_14_days,
sum(daily_deaths) OVER (partition by name order by date asc ROWS BETWEEN 0 PRECEDING AND 9 FOLLOWING) as moving_deaths_post_14_days,

from base_table as t1 left join clase_may.population as t2 using(name)
```

Iterate to find the best number of cluster

```
DECLARE NUM_CLUSTERS INT64 DEFAULT 3; DECLARE MIN_ERROR FLOAT64 DEFAULT 1000.0; DECLARE BEST_NUM_CLUSTERS INT64 DEFAULT -1; DECLARE MODEL_NAME STRING;

WHILE NUM_CLUSTERS < 8 DO

SET MODEL_NAME = CONCAT('ch09eu.london_station_clusters_', CAST(NUM_CLUSTERS AS STRING));

CREATE OR REPLACE MODEL MODEL_NAME OPTIONS(model_type='kmeans',num_clusters=NUM_CLUSTERS,standardize_features = true) AS SELECT * except(station_name) from ch09eu.stationstats;

SET error = (SELECT davies_bouldin_index FROM ML.EVALUATE(MODEL MODEL_NAME));

IF error < MIN_ERROR THEN SET MIN_ERROR = error;SET BEST_NUM_CLUSTERS = NUM_CLUSTERS;

END IF;

SET NUM_CLUSTERS = NUM_CLUSTERS + 1;

END WHILE
```

Agenda

01 Introduccion a BigQuery

02 Conceptos básicos de Standard SQL

Break (🎉🎉)

03 DataStudio como herramienta de Visualización

04 Ejercicio práctico BigQuery y DataStudio

05 BigQuery GeoViz

Fin (🎉🎉)

Analyze GIS data in BigQuery with familiar SQL

Accurate spatial analyses with **Geography** data type

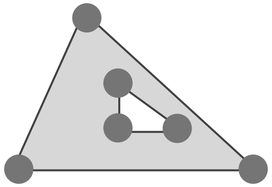
Support for core **GIS functions** – measurements, transforms, constructors, etc… – **using familiar SQL**

Data type:

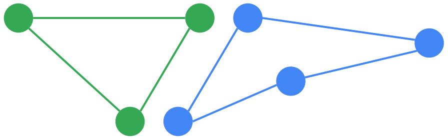
Point

Linestring

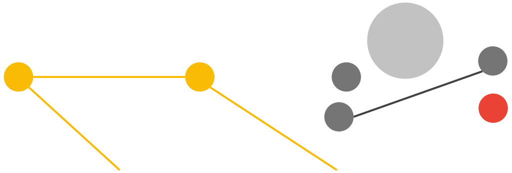
Polygon



Multi-polygon



Collections



Formats:

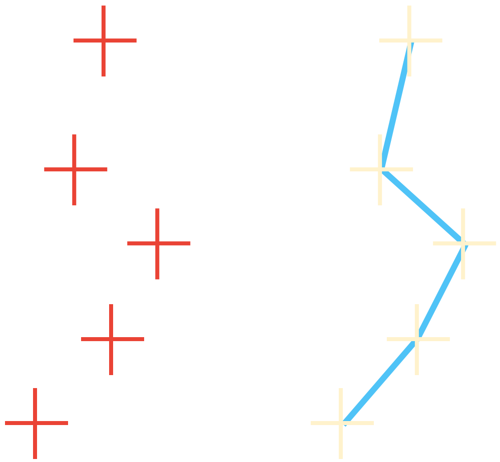
WKT
GeoJSON
WKB

Native SQL support for the most commonly used ST_* functions and geographic data types

Functions	Description
Constructors	Constructive operations build new geography literals from coordinates or existing geographies.
Transformations	Operations that return a single Geography from one or more distinct geographies (e.g., ST_Union)
Predicates	Predicate operations return true/false for some spatial relationship between two geometries. Most frequently used in filter clauses.
Accessors	Operations that let users navigate and select between multiple ways of handling a record based on its type, or select a particular element.
Measures	Measure operations compute some property of the geography such as perimeter, area, or distance to another geography.
Parsers	Operations that construct a Geography from raw coordinates or other geographies.
Formatters	Formatting operations return a geography converted into a standardized (usually string) format suitable for presenting in query results.

Constructors

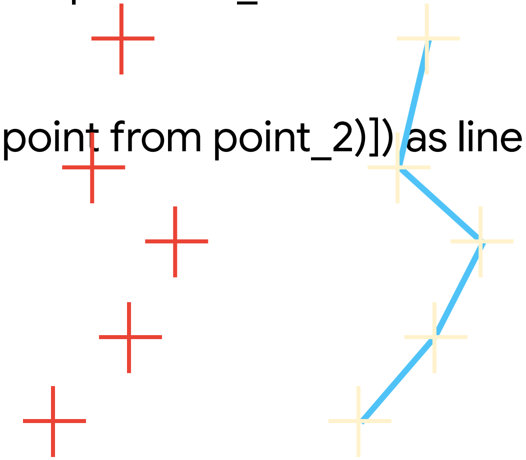
ST_GEOGPOINT(longitude, latitude)
ST_MAKELINE(geography_1, geography_2)
ST_MAKELINE(array_of_geography)
ST_MAKEPOLYGON(geography_expression)
ST_MAKEPOLYGON(geography_expression, array_of_geography)
ST_MAKEPOLYGONORIENTED(array_of_geography)



Build geographies from
coordinates or existing geographies

Constructors

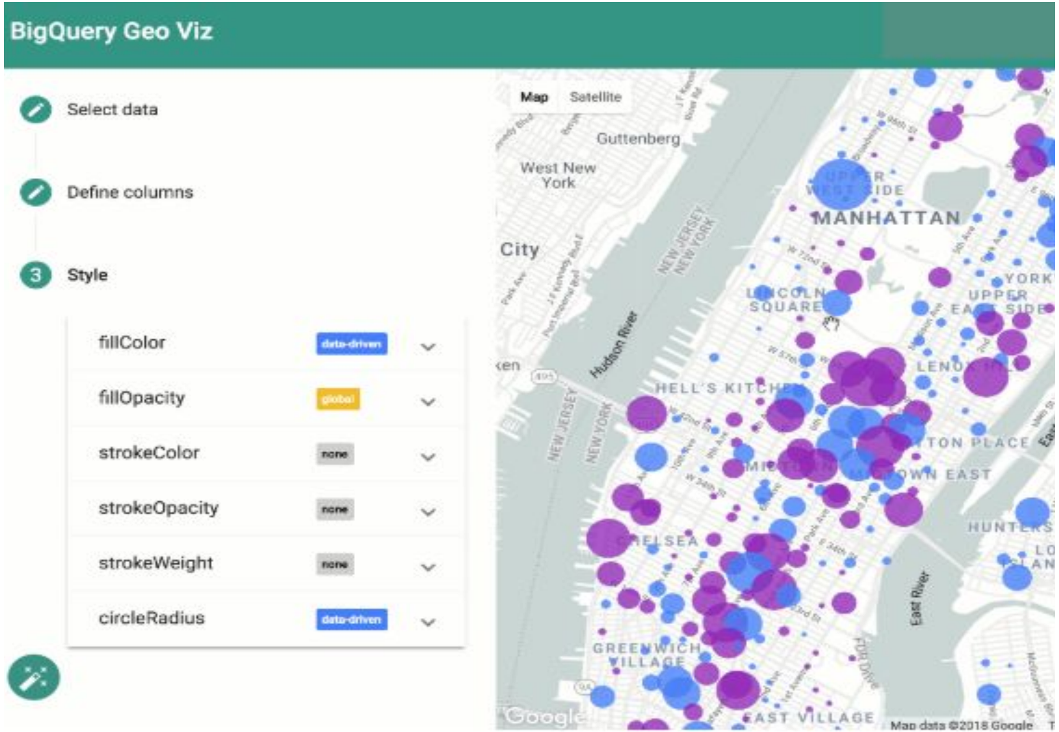
```
with
point_1 as (select ST_GEOGPOINT(longitude, latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name =
"Torino" limit 1 ),
    point_2 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name =
"Vercelli" limit 1)

    select ST_MAKELINE([(select point from point_1),(select point from point_2)]) as line
```



Build geographies from
coordinates or existing geographies

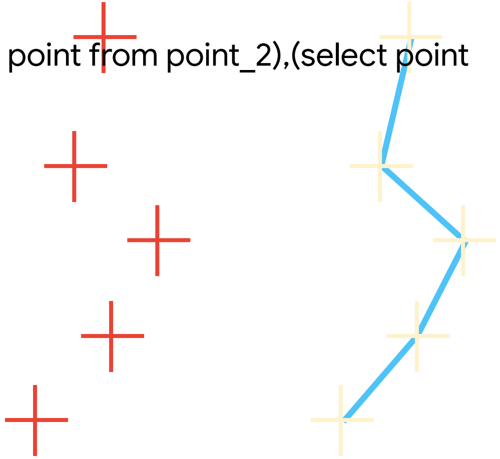
GEO VIZ UI

Test in GeoViz UI



Constructors

```
with
point_1 as (select ST_GEOGPOINT(longitude, latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Torino" limit 1 ),
    point_2 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Vercelli" limit 1),
    point_3 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Novara" limit 1)


    select ST_MAKEPOLYGON (ST_MAKELINE([(select point from point_1),(select point from point_2),(select point
from point_3)])) as triangle
```



Build geographies from
coordinates or existing geographies

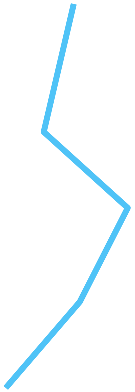
Parsers & formatters

```
ST_GEOGFROMGEOJSON(geojson_string)
ST_GEOGFROMTEXT(wkt_string)
ST_GEOGFROMWKB(wkb_bytes)
```

```
ST_ASGEOJSON(geography_expression)
ST_ASTEXT(geography_expression)
ST_ASBINARY(geography_expression)
```

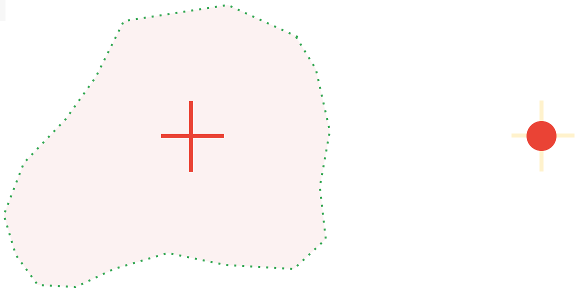
Create/export geographies
between formats

((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)),
((0 0 0, 0 1 0, 0 1 1, 0 0 1, 0 0 0)),
((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)),
((1 1 1, 1 0 1, 0 0 1, 0 1 1, 1 1 1)),
((1 1 1, 1 0 1, 1 0 0, 1 1 0, 1 1 1))



Transformations

```
ST_INTERSECTION(geography_1, geography_2)
ST_UNION(geography_1, geography_2)
ST_UNION(array_of_geography)
ST_UNION_AGG(geography)
ST_DIFFERENCE(geography_1, geography_2)
ST_CENTROID(geography_expression)
ST_CLOSESTPOINT(geography_1, geography_2[,spheroid=FALSE])
ST_BOUNDARY(geography_expression)
ST_SNAPTOGRID(geography_expression, grid_size)
```



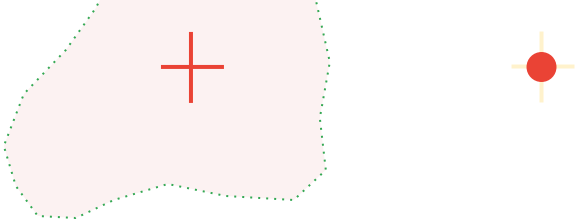
Create new geographies
with similar properties

Google Cloud

Transformations

```
with
point_1 as (select ST_GEOGPOINT(longitude, latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Torino" limit 1 ),
    point_2 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Vercelli" limit 1),
    point_3 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Novara" limit 1)


    select ST_MAKEPOLYGON (ST_MAKELINE([(select point from point_1),(select point from
point_2),(select point from point_3)])) as triangle,
     ST_CENTROID(ST_MAKEPOLYGON (ST_MAKELINE([(select point from point_1),(select point from
point_2),(select point from point_3)]))) as centroid
```

Create new geographies
with similar properties

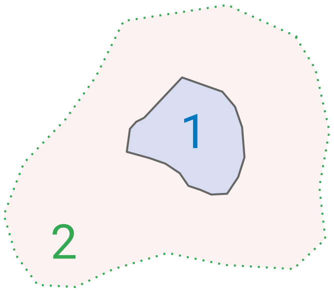


Predicates

```
ST_CONTAINS(geography_1, geography_2)
ST_COVEREDBY(geography_1, geography_2)
ST_COVERS(geography_1, geography_2)
ST_DISJOINT(geography_1, geography_2)
ST_DWITHIN(geography_1, geography_2, distance[, spheroid=FALSE])
ST_EQUALS(geography_1, geography_2)
ST_INTERSECTS(geography_1, geography_2)
ST_INTERSECTSBOX(geography, lng1, lat1, lng2, lat2)
ST_TOUCHES(geography_1, geography_2)
```

`ST_WITHIN(geography_1, geography_2)`

Filter geographies
(TRUE/FALSE)

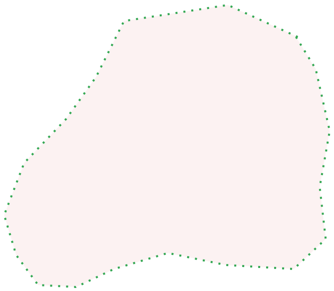


TRUE

Measures

```
ST_DISTANCE(geography_1, geography_2[, spheroid=FALSE])
ST_LENGTH(geography_expression[, spheroid=FALSE])
ST_PERIMETER(geography_expression[, spheroid=FALSE])
ST_AREA(geography_expression[, spheroid=FALSE])
ST_MAXDISTANCE(geography_1, geography_2[, spheroid=FALSE])
```

Compute measurements
of geographies



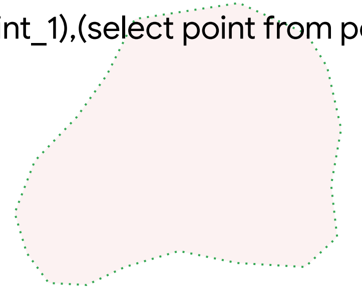
3967
(meters)

Google Cloud

Measures

```
with
point_1 as (select ST_GEOGPOINT(longitude, latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Torino" limit 1 ),
    point_2 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Vercelli" limit 1),
    point_3 as (select ST_GEOGPOINT(longitude,latitude) as point from
`bigquery-public-data.covid19_italy.data_by_province` where province_name = "Novara" limit 1)


    select ST_MAKEPOLYGON (ST_MAKELINE([(select point from point_1),(select point from point_2),(select
point from point_3)])) as triangle,
     ST_AREA(ST_MAKEPOLYGON (ST_MAKELINE([(select point from point_1),(select point from point_2),(select
point from point_3)]))) as area
```

Compute measurements
of geographies



3967
(meters)

Google Cloud

GEO VIZ UI

Insert Script in BigQuery Geo Viz

```
SELECT
  ST_GeogPoint(longitude, latitude)  AS WKT,
  num_bikes_available
FROM
  `bigquery-public-data.new_york.citibike_stations`
WHERE num_bikes_available > 30
```

GEO VIZ UI

Insert Script in BigQuery Geo Viz

with max_date as (Select max(date(date,"Europe/Madrid")) as da
from `bigquery-public-data.covid19_italy.data_by_province`) ,
base_table as (

select date(date,"Europe/Madrid") as
date,ST_GEOGPOINT(longitude, latitude) as point, confirmed_ca
from `bigquery-public-data.covid19_italy.data_by_province`)

select point,confirmed_cases from base_table as t1
inner join max_date using (date)

order by confirmed_cases desc

```
with max_date as (Select max( date(date,"Europe/Madrid")) as date
from `bigquery-public-data.covid19_italy.data_by_province`) ,
base_table as (

select date(date,"Europe/Madrid") as date,ST_GEOGPOINT(longitude,
latitude) as point, confirmed_cases from `bigquery-public-
data.covid19_italy.data_by_province` )

select point,confirmed_cases from base_table as t1
inner join max_date using (date)
order by confirmed_cases desc
```

Select Style Fill Color

fillColor **data-driven**

Fill color of a polygon or point. For example, "linear" or "interval" functions may be used to map numeric values to a color gradient.

Data-driven

Function
linear

Field
num_bikes_available

Domain

31	40	56

min: 31 max: 56

Range

Select Style Radious Circle

circleRadius data-driven

Radius of the circle representing a point, in meters. For example, a "linear" function could be used to map numeric values to point sizes, creating a scatterplot style.

Data-driven

Function
linear

Field
confirmed_cases

Domain

10000	30000	50000	100000	max: 191202

Range

5000	15000	20000	25000

Select Style Radious Circle

circleRadius **data-driven**

Radius of the circle representing a point, in meters. For example, a "linear" function could be used to map numeric values to point sizes, creating a scatterplot style.

Data-driven

Function
linear

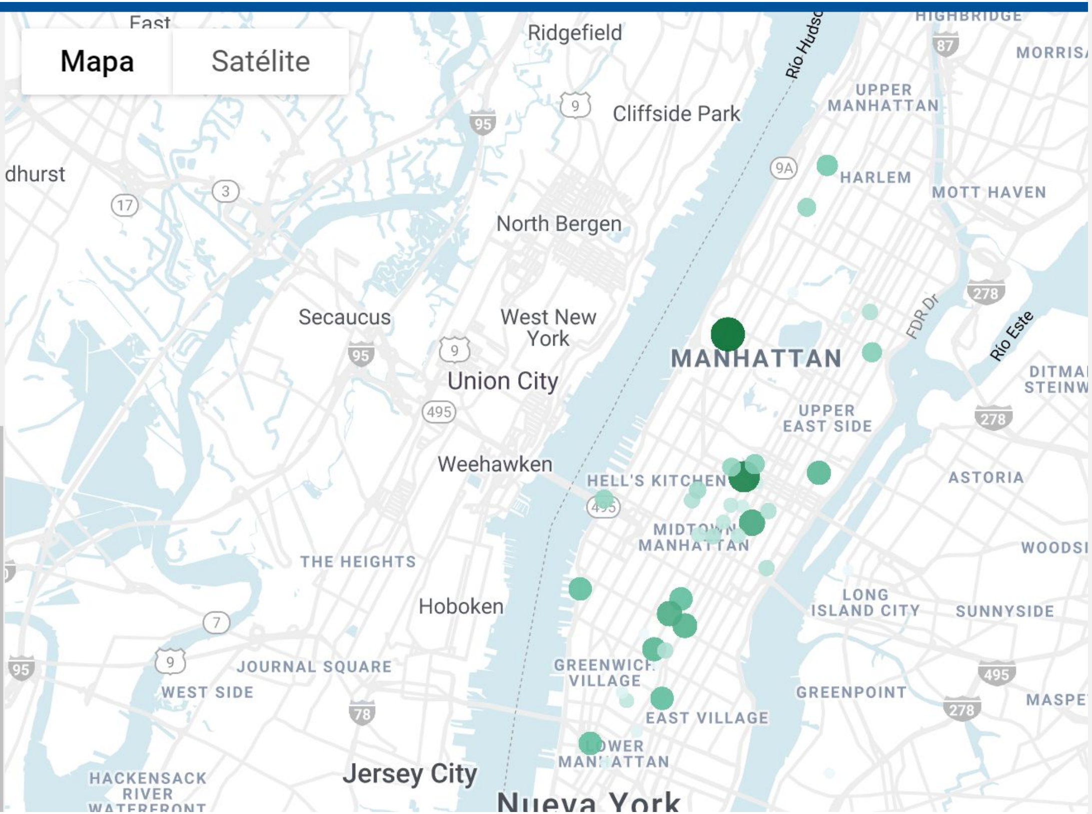
Field
num_bikes_available

Domain

31	40	56	max:
	31		56

Range

100	200	300



Select Style Fill Color

Muchas Gracias

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

Fin (🎉🎉)

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

Fin (🎉🎉)

ML For Analysts | What is Machine Learning

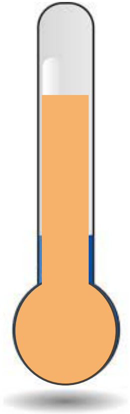
Huge amount of labeled data

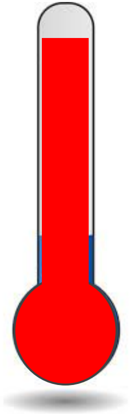
Learn from data (algorithm)

Predict new cases

ML For Analysts | How does Machine Learning work?



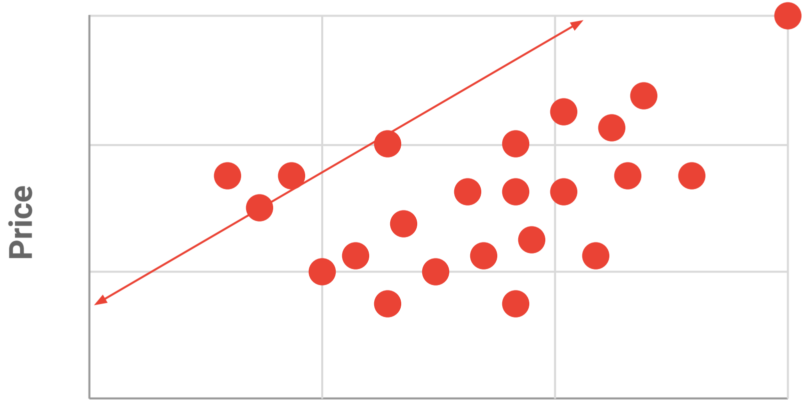



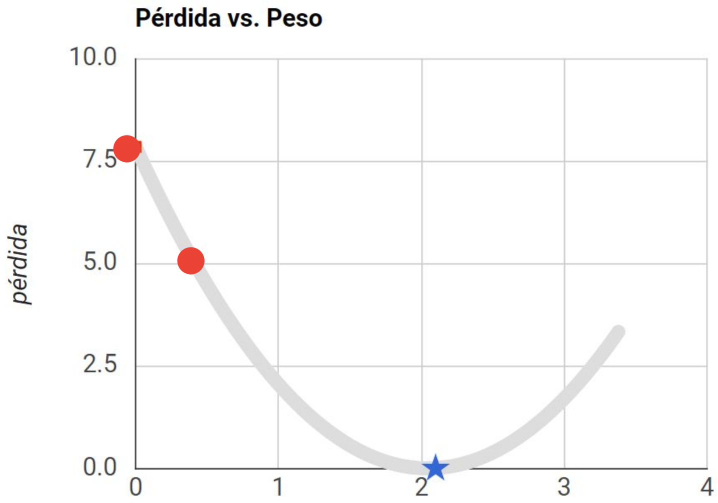

FOUND

Machine learning - How does it work ?

Proposed algorithm

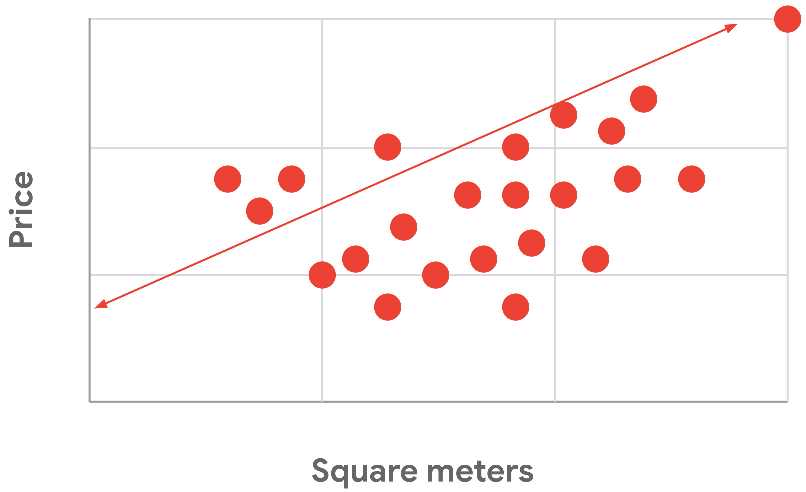


Loss function

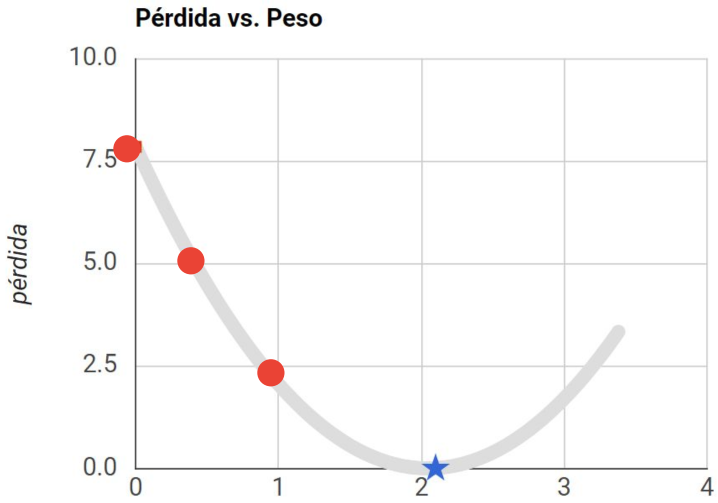


Machine learning - How does it work ?

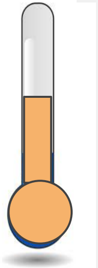
Proposed algorithm



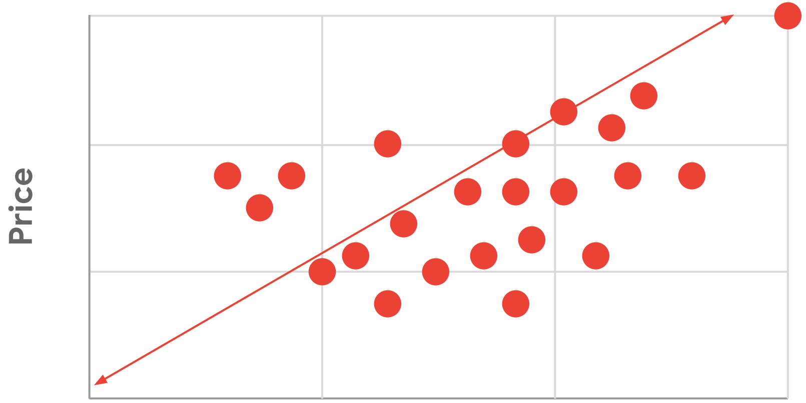
Loss function



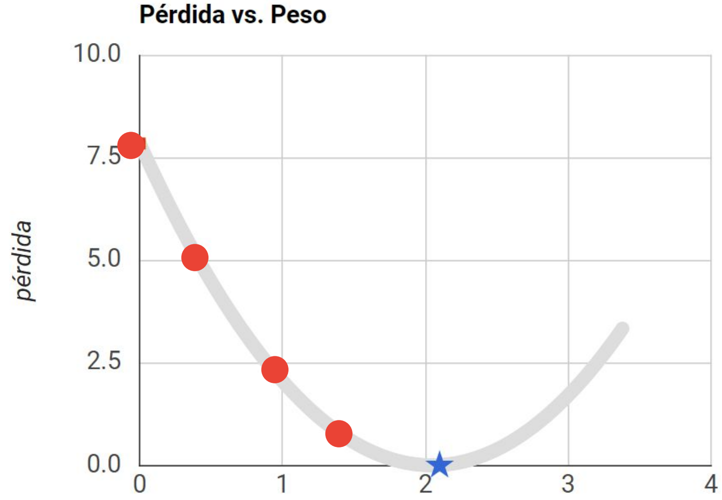
Machine learning - How does it work ?



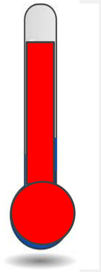
Proposed algorithm



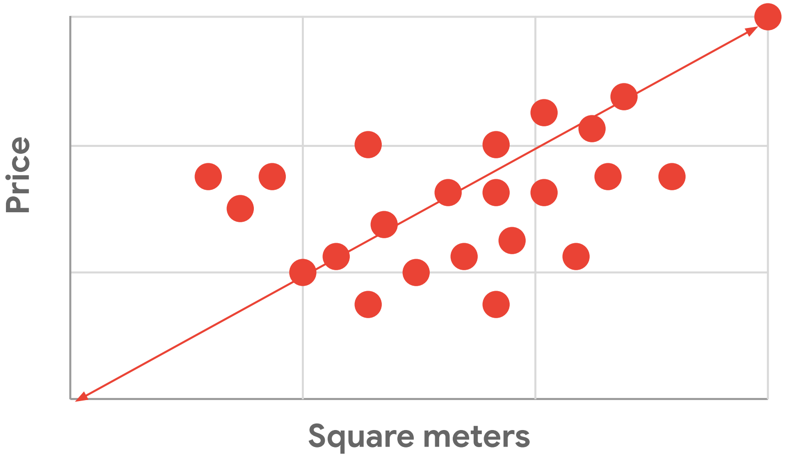
Loss function



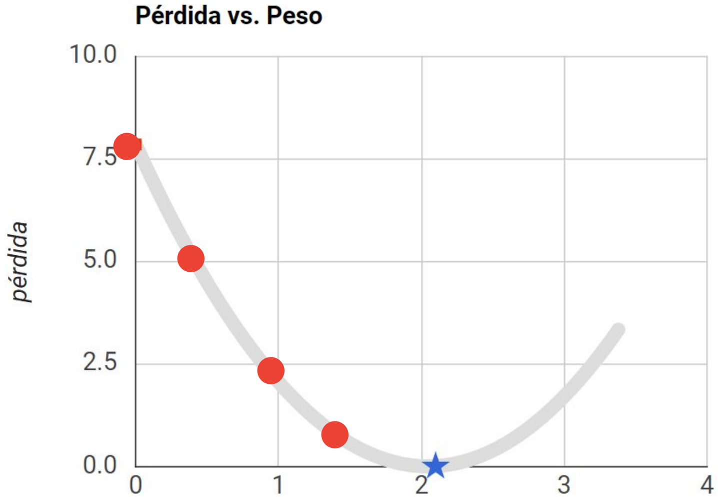
Machine learning - How does it work ?



Proposed algorithm



Loss function

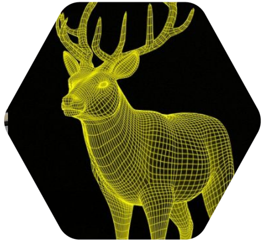


ML For Analysts | What is the process of ML?

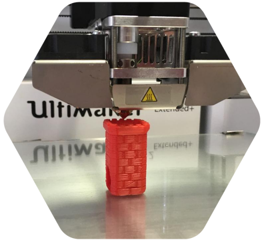
Define objectives

Collect data

Understand and prepare the data



Create the model



Refine the model

Serve the model

"

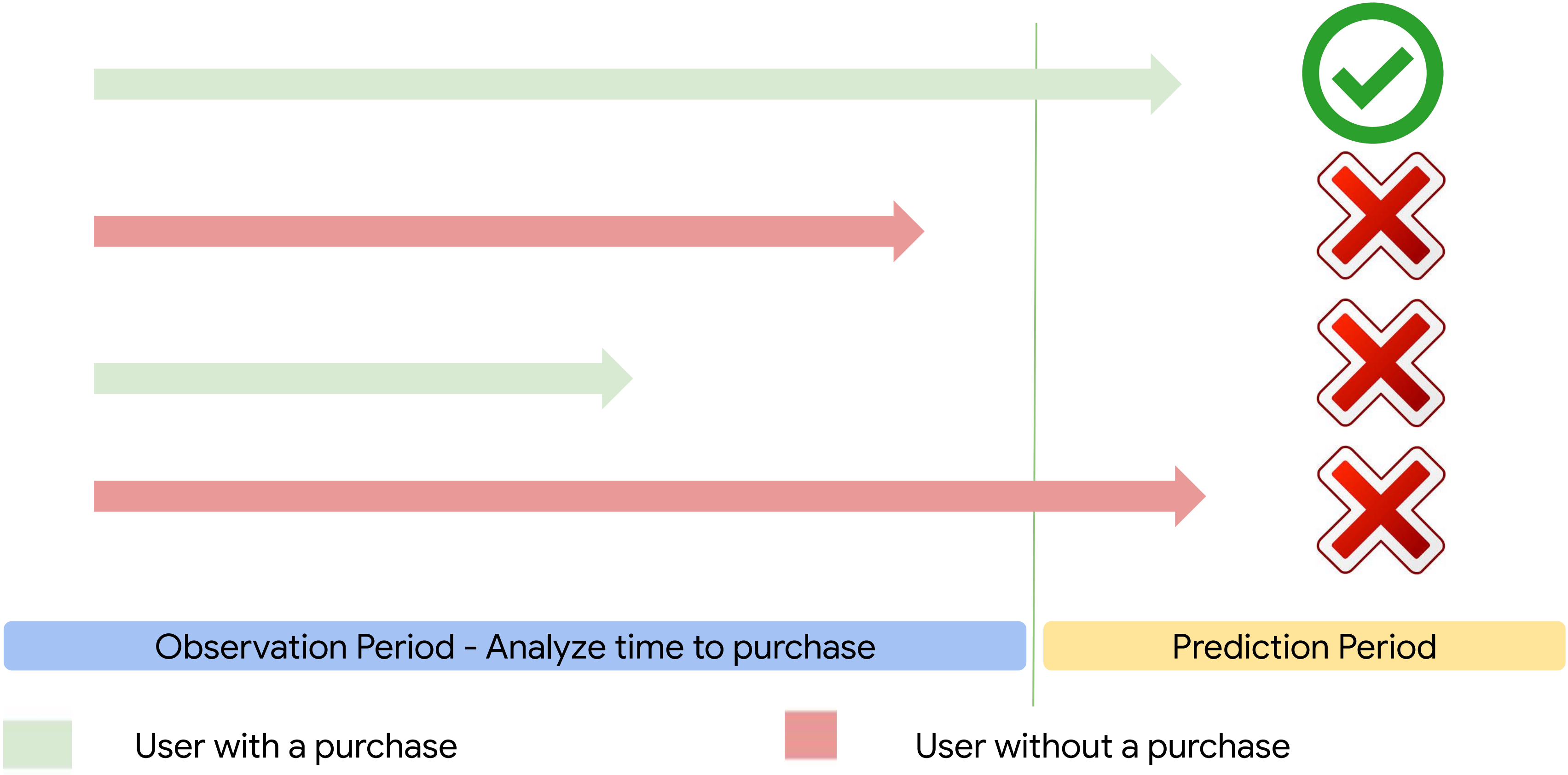
If your company isn't good at analytics,
it's not ready for AI

– Harvard Business Review , 2017

Collect
data

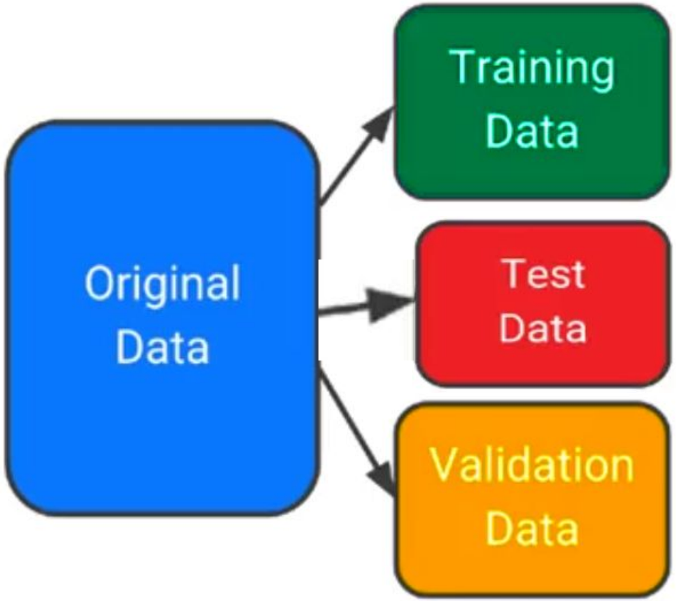
Understand
and
prepare the
data

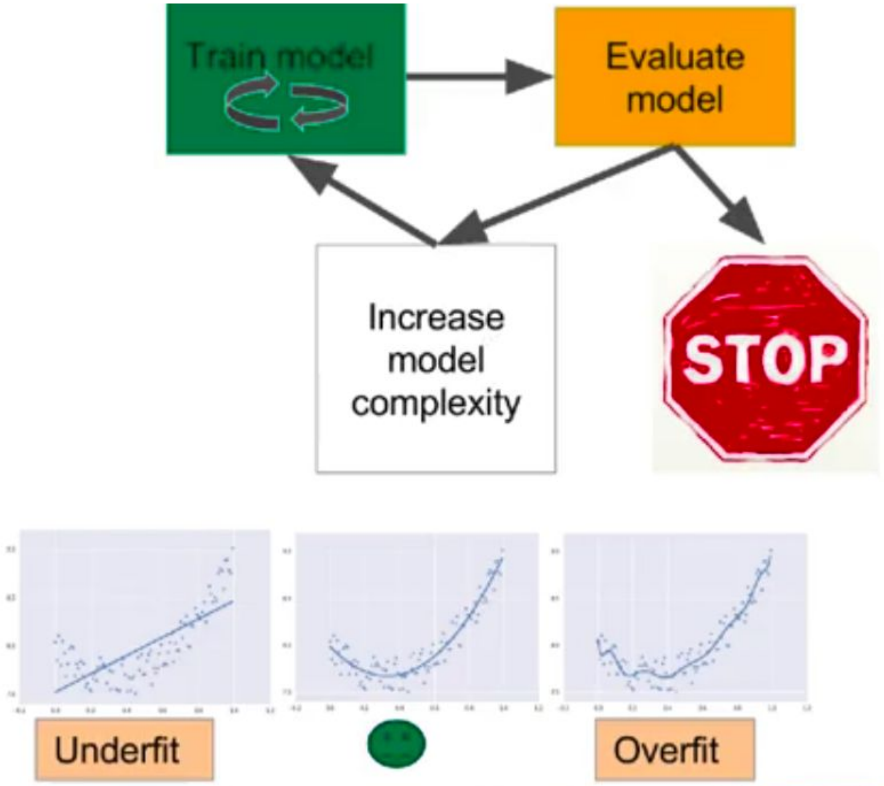
ML For Analysts | Model Design



ML For Analysts | Model Evaluation framework

s





ML For Analysts | How to evaluate a model

Classify the cats!

ML For Analysts | How to evaluate a model

		ML System Says	
		Cat	No Cat
Truth	Cat	True Positive #TP	False Negative #FN
	No Cat	False Positive #FP	True Negative #TN

ML For Analysts | How to evaluate a model

Predicted Categories by Model

ML For Analysts | How to evaluate a model

Accuracy: percentage of correct predictions

ML For Analysts | How to evaluate a model

Precision: percentage of correct predictions in positive labels

Accuracy when
ML says "cat"

TP + FP = 5

$$\text{Precision} = TP / (TP + FP) = 2 / 5 = 0.40$$

ML For Analysts | How to evaluate a model

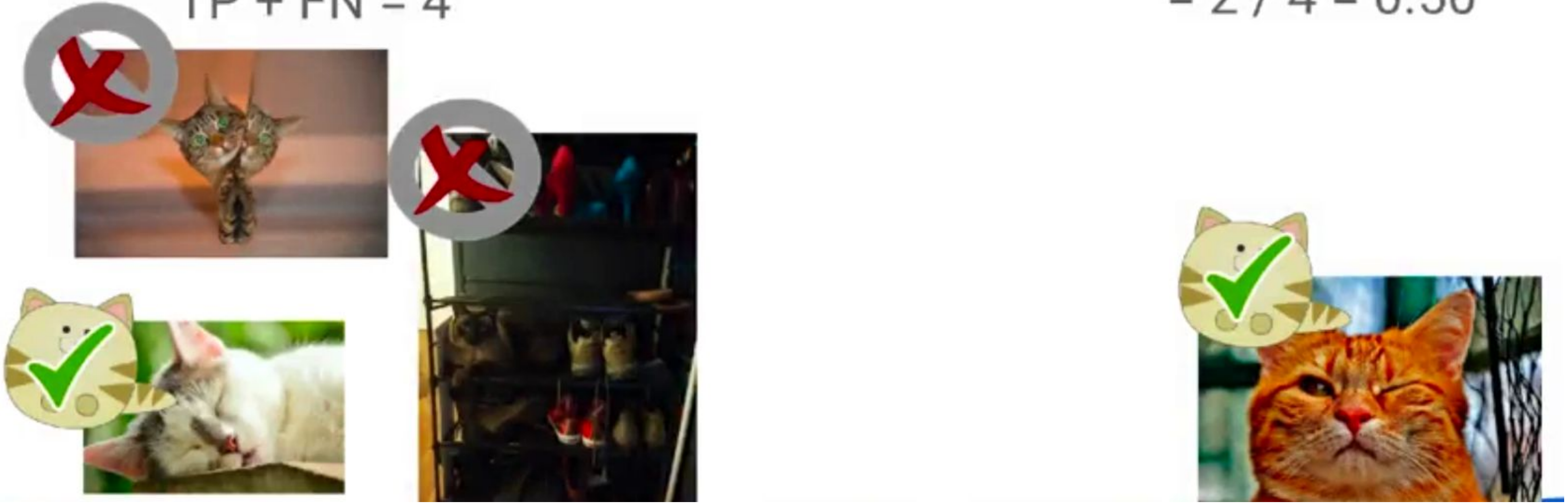
Recall: percentage of existing positive labels predicted by the model

Recall = fraction of cats ML finds

TP + FN = 4

Recall = TP / (TP + FN)

= 2 / 4 = 0.50



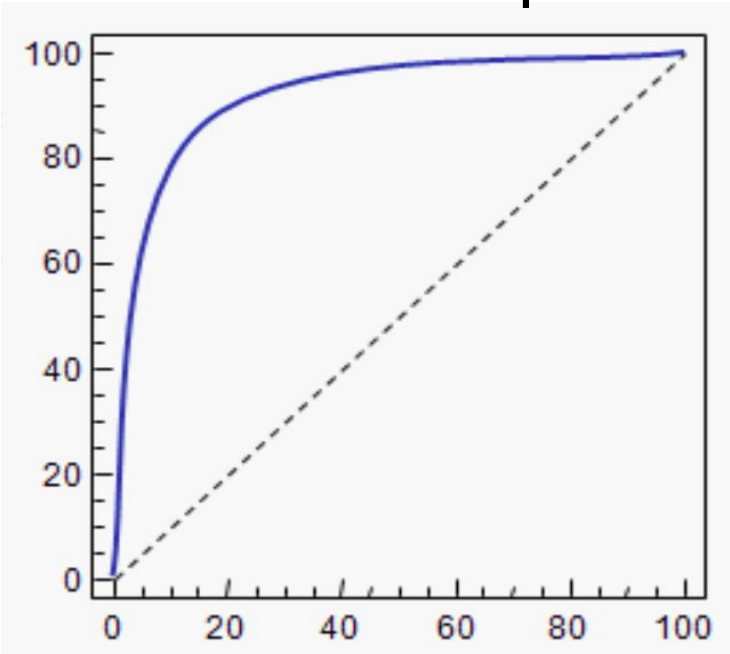
ML For Analysts | How to evaluate a model

ML For Analysts | How to evaluate a model

¿Qué métrica utilizarías para ver un modelo de detección de delincuencia

ML For Analysts | How to evaluate a model

ROC: Recall Vs False positive Rate



Recall TP/(TP +FN)

False Positive Rate: ML predicts Cat y it is not cat/ Total No Cats FP/ (FP + TN)

ML For Analysts | How to evaluate a model - Cost Benefit Matrix

- ➔ We want to send a promotion to customers about a new product
- ➔ We have ad related costs of making the offer visible of 5$
- ➔ We have an estimated profit of 100$ if the customer buys the product
- ➔ True Positive: Sending the offer to the customer that ends buying the product: Benefit: 100-5
- ➔ True Negative: Not Sending the offer to the customer that will not buy: Benefit: 0
- ➔ False Positive: Sending the offer to a customer that will not buy: Benefit: -5
- ➔ False Negative: Not sending the offer to a customer that would have bought: 0

ML For Analysts | How to evaluate a model - Cost Benefit Matrix

		Model Predictions		
		Model Predicts Purchase	Model Predicts No purchase	**TOTAL**
Real Results	Purchase	100	50	**150**
	No Purchase	400	2,000	**2,400**
	TOTAL	**500**	**2,050**	**2,550**

	Promotion Sent	Promotion Not send
Purchase	95	0
No Purchase	-5	0

Accuracy	82.35%	(100+2000)/2550
Recall	66.67%	(100/150)
Precision	20.00%	(100/500)
Average Conv Ra	5.88%	(150/2550)
Precision Lift	3.4	(20/ 5.88)

Cost Benefit Matrix	Promotion Sent	Promotion Not send
Purchase	9,500	0
No Purchase	-2,000	0
BENEFIT	**7,500**	

If we send the promotion to all the benefit would be

Benefit	14,250	(150*95)
Cost	-12,000	(2400*-5)
BENEFIT	**2.250**	

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

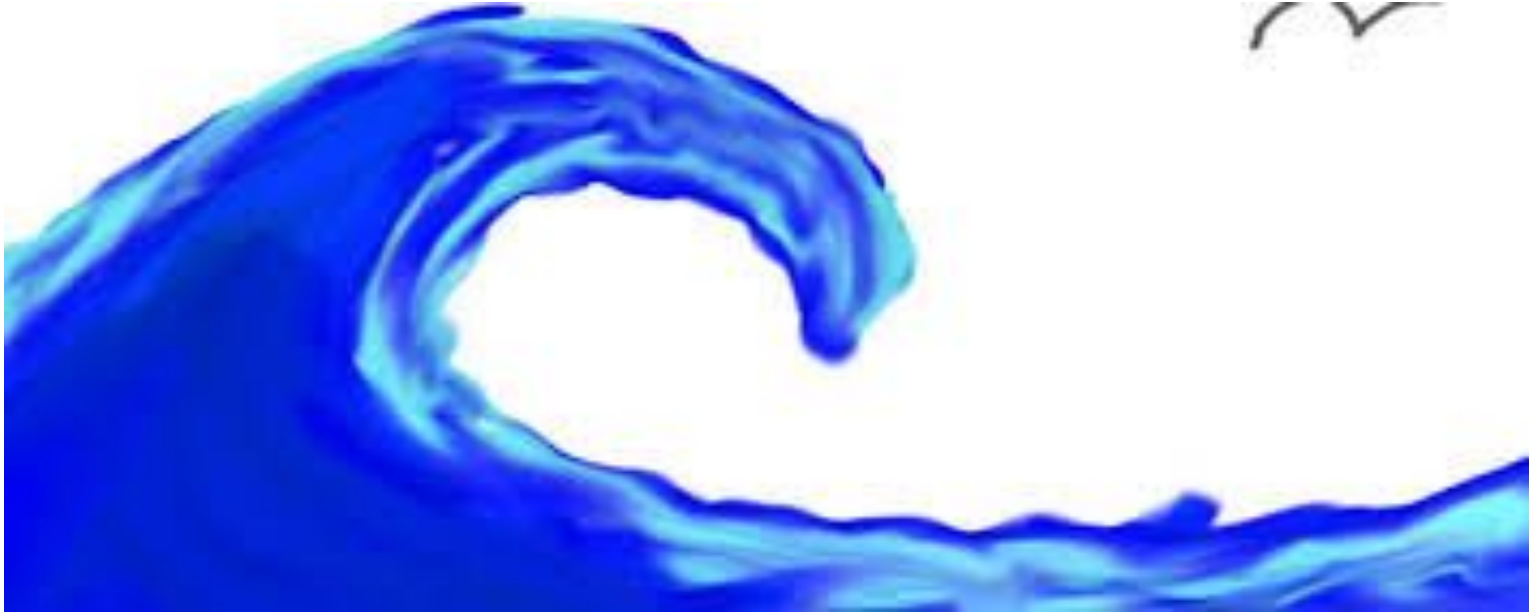
04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

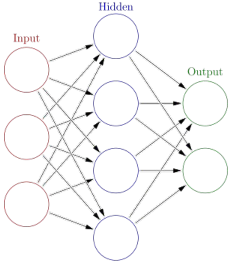
Fin (🎉🎉)

Digital revolutions that changed the world



What these revolutions have in common

ARPANET
THE FIRST INTERNET



1936 1969 1996 1980

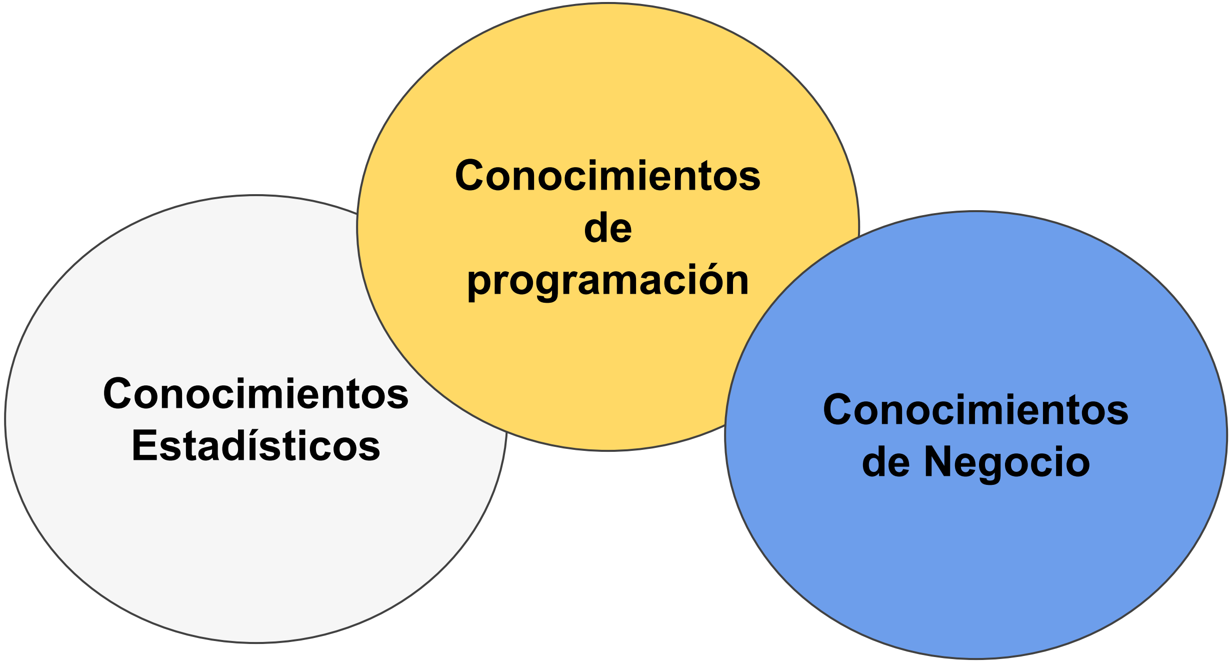
1964 1994 2007 2017

30 Years 13 Years 10 Years

“Today we’re evolving from a mobile first world to an A.I. first world”

- Sundar Pichai, CEO of Google

Being Data Scientist is very complex - Cloud make it easier



Google Cloud is democratizating ML for all audiences

	Novato	Medio	Avanzado	Skills necesarias
TensorFlow y CloudML Engine			●	• Requires programming and ML knowldge • Google Cloud ML makes production easier and scalable
BQML		●		• Requires basic ML knowledge • Train, evaluate and predict with SQL language
ML APIs/AutoML	●			• Basic programming knowledge required • Pre-trained models • Easy to deploy and no ML knowledge required

Machine Learning with SQL

1. **Execute** ML initiatives without moving data from BigQuery
2. **Iterate** on models in SQL in BigQuery to increase development speed
3. **Automate** common ML tasks and hyperparameter tuning

Modelos de ML de BigQuery

Classification

- Logistic regression
- DNN classifier (TensorFlow)
- XGBoost
- AutoML Tables

Other Models

- k-means clustering
- Time series forecasting
- Recommendation: Matrix factorization

Regression

- Linear regression
- DNN regressor (TensorFlow)
- XGBoost
- AutoML Tables

Model Import/Export

- TensorFlow models for batch and online prediction

BQML | Create Model Syntax (link)

```
{CREATE MODEL | CREATE MODEL IF NOT EXISTS | CREATE OR REPLACE MODEL}
model_name
[OPTIONS(model_option_list)]
[AS query_statement]

model_option_list:
 MODEL_TYPE = { 'LINEAR_REG' | 'LOGISTIC_REG' | 'KMEANS' | 'TENSORFLOW' }
 [, INPUT_LABEL_COLS = string_array ]
 [, OPTIMIZE_STRATEGY = { 'AUTO_STRATEGY' | 'BATCH_GRADIENT_DESCENT' | 'NORMAL_EQUATION' } ] ⟶
 [, L1_REG = float64_value ]
 [, L2_REG = float64_value ]
 [, MAX_ITERATIONS = int64_value ]
 [, LEARN_RATE_STRATEGY = { 'LINE_SEARCH' | 'CONSTANT' } ]
 [, LEARN_RATE = float64_value ]
 [, EARLY_STOP = { TRUE | FALSE } ]
 [, MIN_REL_PROGRESS = float64_value ]
 [, DATA_SPLIT_METHOD = { 'AUTO_SPLIT' | 'RANDOM' | 'CUSTOM' | 'SEQ' | 'NO_SPLIT' } ]
 [, DATA_SPLIT_EVAL_FRACTION = float64_value ]
 [, DATA_SPLIT_COL = string_value ]
 [, LS_INIT_LEARN_RATE = float64_value ]
 [, WARM_START = { TRUE | FALSE } ]
 [, AUTO_CLASS_WEIGHTS = { TRUE | FALSE } ]
 [, CLASS_WEIGHTS = struct_array ]
 [, NUM_CLUSTERS = int64_value ]
 [, DISTANCE_TYPE = { 'EUCLIDEAN' | 'COSINE' } ]
 [, STANDARDIZE_FEATURES = { TRUE | FALSE } ]

  [, MODEL_PATH = string_value ]
```

BQML | Evaluate Model Syntax (link)

```
ML.EVALUATE(MODEL model_name
            [, {TABLE table_name | (query_statement)}]

            [, STRUCT(<T> AS threshold)])
```

EXAMPLE

```
SELECT * FROM ML.EVALUATE(MODEL `mydataset.mymodel`, (

  SELECT custom_label, column1, column2

 FROM

   `mydataset.mytable`),

 STRUCT(0.55 AS threshold))
```

BQML | Evaluate Model Syntax ROC (link)

```
ML.ROC_CURVE(MODEL model_name

            [, {TABLE table_name | (query_statement)}]

            [, GENERATE_ARRAY(thresholds)])
```

EXAMPLE

```
SELECT  * FROM ML.ROC_CURVE(MODEL `mydataset.mymodel`,

    TABLE `mydataset.mytable`)
```

BQML | Evaluate Model Syntax Conf Matrix (link)

```
ML.CONFUSION_MATRIX(MODEL model_name
           [, {TABLE table_name | (query_statement)}]
           [, STRUCT(<T> AS threshold)])
```

EXAMPLE

```
SELECT * FROM  ML.CONFUSION_MATRIX(MODEL `mydataset.mymodel`,  (
    SELECT * FROM
      `mydataset.mytable`))
```

BQML | Predict Model Syntax (link)

```
ML.PREDICT(MODEL model_name,
          {TABLE table_name | (query_statement)}
          [, STRUCT<threshold FLOAT64> settings)])
```

EXAMPLE

```
SELECT  * FROM ML.PREDICT(MODEL `mydataset.mymodel`    (
    SELECT      label,column1,column2
    FROM
      `mydataset.mytable`))
```

BQML | Model Feature Info (link)

Feature information: `SELECT * FROM  ML.FEATURE_INFO(MODEL `mydataset.mymodel`)`

Feature Weights:
```
SELECT category,weight FROM UNNEST((SELECT category_weights FROM
ML.WEIGHTS(MODEL `mydataset.mymodel`) WHERE processed_input = 'input_col'))
```

Kmeans Centroids:
```
SELECT  * FROM ML.CENTROIDS(MODEL `project_id.dataset.model`)
```

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

Fin (🎉🎉)

CRMint

- A client-side web application for scheduling data flow process between GA 360 and Google ad products for solving business tasks: GA remarketing, LTV, etc.

- Data Pipeline workers
    - Data Import
    - Measurement Protocol
    - GA API data pull to BigQuery
    - Cloud Storage to BigQuery
    - BigQuery to Cloud Storage
    - BigQuery Query launcher
    - Cloud ML model predictor
    - GA audience launcher

‘Deploy CRM int‘

Link

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

Fin (🎉🎉)

‘Iberia ML to predict users probability to convert with BQML‘ - (link)

Iberia uses the power of Machine Learning to identify high value customers

6 pasos para conquistar Data Analytics y AI

STEP 1

Hacer las preguntas adecuadas

STEP 2

Research
Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

Capturar datos

STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados

6 pasos para conquistar Data Analytics y AI

STEP 1

Hacer las preguntas adecuadas

STEP 2

Research
Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

Capturar datos

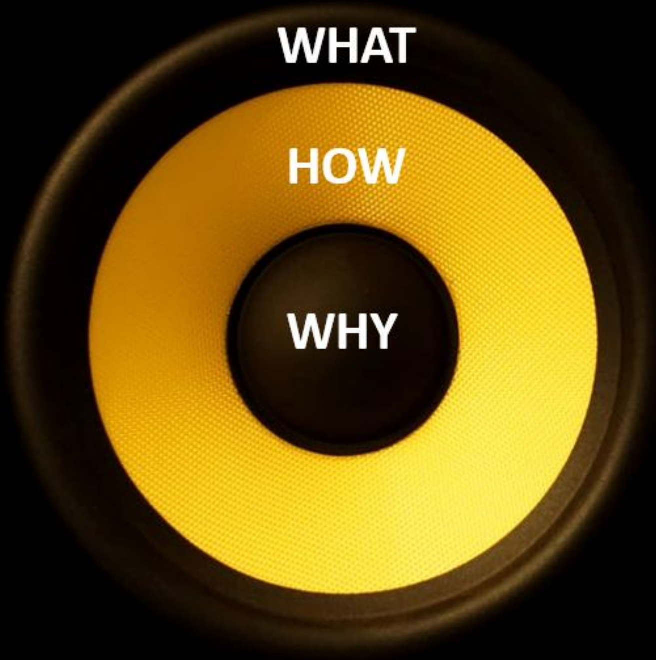
STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados

Simon Sinek´s Golden Circle



WHY: Propósito o razón de existencia de la empresa

Apple: "*We believe in challenging the status quo and doing things differently*"

HOW: El proceso o acciones para llevar a cabo el why

Apple: "*Our products are beautifully designed and easy to use*"

WHAT: El resultado, lo que hace la empresa

Apple: "*We make computers*"

"The Why"

IBERIA

Cada día es el primer día, el lema que nos abandera

Tenemos una trayectoria exitosa pero no nos conformamos con lo que somos, sino que nos esforzamos por superarnos y mejorar día tras día.

Esta es nuestra actitud y nuestra forma de afrontar cada día para ofrecer lo mejor de nosotros.

Nuestros valores

Qué hacemos y cómo lo hacemos

Afinidad: Es el vínculo cercano y emotivo, nuestra expresividad y manera de vivir y sentir las cosas que nos hacen relevantes también a Europa y al mundo entero.

Empuje: Es la capacidad de canalizar recursos y enfocar esfuerzos de manera eficaz y eficiente para alcanzar metas y consolidar el liderazgo en nuestro sector.

Talento: Es la profesionalidad que se nos reconoce y valora. El espíritu resolutivo, práctico, demostrado y experimentado que nos lleva a manifestar un gran desempeño de manera llana y natural.

“The how”- Link

Adaptar el mensaje para proveer una experiencia única de cara a la adquisición de clientes de alto valor ✓

Mejorar la eficiencia de las campañas de Marketing gracias al conocimiento de su comportamiento web ✓

Ofrecer una experiencia de compra única a través de la mejora continua en la captación de clientes a través del canal directo ✓

‘Iberia ML to predict users probability to convert with BQML‘ - (link)

1. Improve the average ROAS, by identifying and prioritizing high value customers

2. Tailor the message for high value customers

3. Find users similar to high value customers, using the power of Google Audiences

‘Iberia ML to predict users probability to convert with BQML‘ - (link)

Iberia, in collaboration with Gauss & Neumann and Google, analyzes the browsing data of web users, available in Google Analytics 360 and dumped in BigQuery , using an algorithm adapted to their business model and developed in BigQuery Machine Learning , which predicts both the probability of conversion, as the estimated transaction value.

“The what”- Link

Visita

Análisis

Comportamiento web

- Dispositivo
- Ubicación
- Estacionalidad
- Número visitas
- etc.

Scoring

85% Probabilidad — **Tier 1**

25% Probabilidad — **Tier 5**

60% Probabililidad — **Tier 2**

Activación

1. Audiences

Detectar características de clientes potenciales de alto valor

2. Estrategia personalizada

Ofrecer una estrategia adaptada para la captación de cada cliente

3. Personalización de mensajes

Ofrecer anuncios personalizados y una experiencia de compra única

Internal Stakeholders – ¿Apoyan el cambio?

- Convencer sobre el poder de los datos (más allá de la intuición)
- Vende en privado, comunica en público
- Explica el proceso y no sólo el resultado (Quick Wins, Expectativas)
- Muestra insights reveladores con los datos y genera engagement

ACCIONISTAS

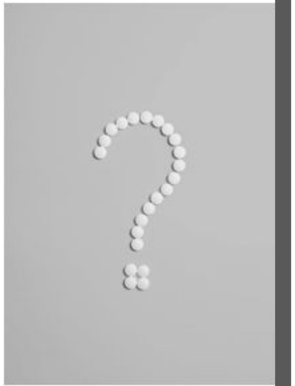
LEGAL & COMPLIANCE

C – LEVELS
(CEO, CFO, CDO, CPO,CIO, IT)

EMPLEADOS

6 pasos para conquistar Data Analytics y AI

STEP 1



Hacer las preguntas adecuadas

STEP 2

Research
Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

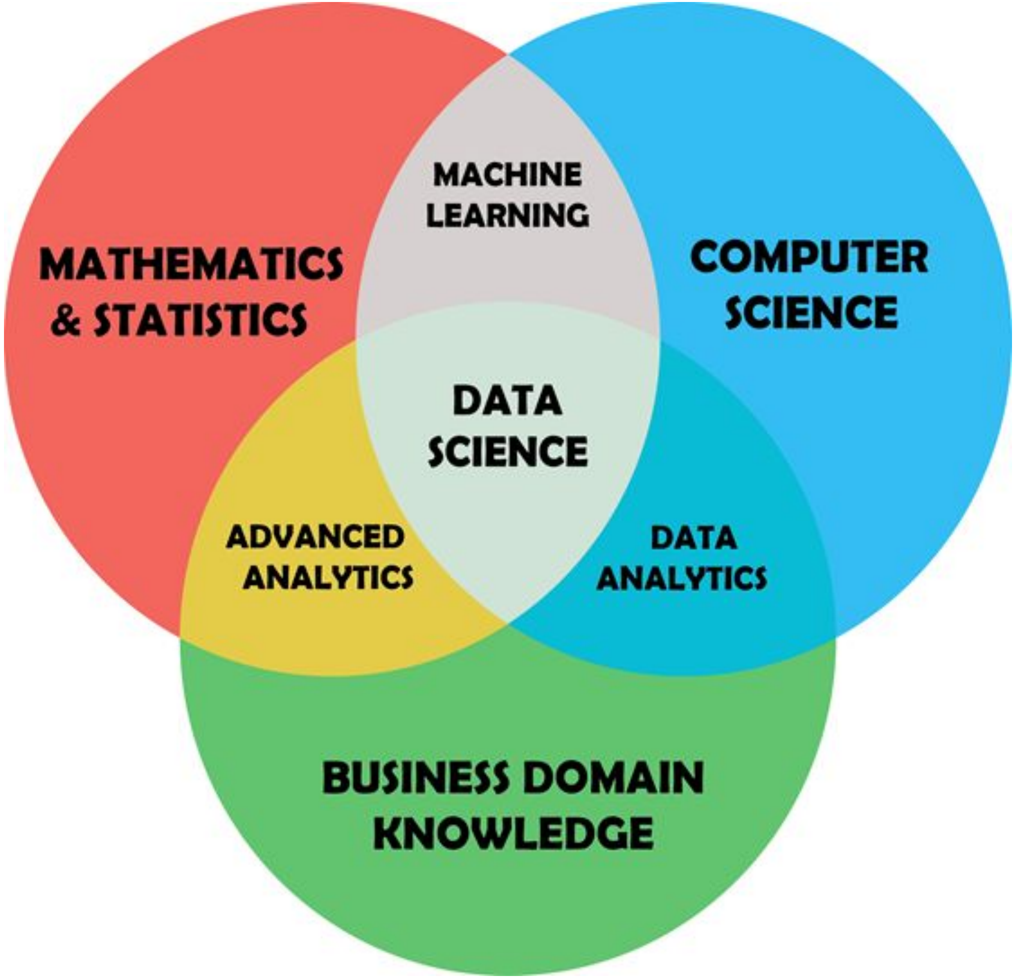
Capturar datos

STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados


MACHINE LEARNING
MATHEMATICS & STATISTICS
COMPUTER SCIENCE
DATA SCIENCE
ADVANCED ANALYTICS
DATA ANALYTICS
BUSINESS DOMAIN KNOWLEDGE

6 pasos para conquistar Data Analytics y AI

STEP 1

Hacer las preguntas adecuadas

STEP 2

Research
Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

Capturar datos

STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados

¿Qué **variables** de navegación **web** creéis que pueden ayudar a **predecir** si un usuario va a **comprar** un billete de avión?

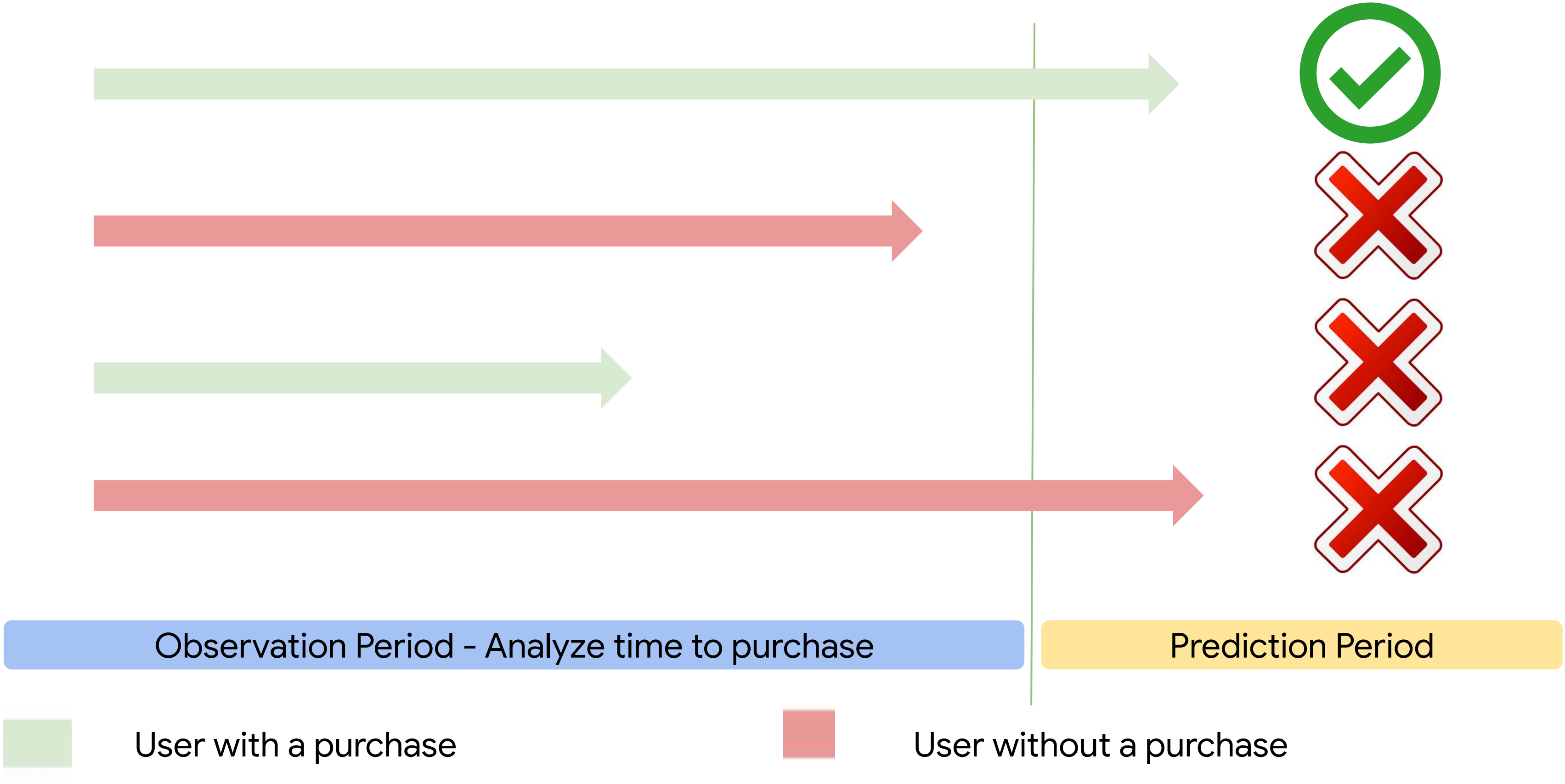
User profile features

- Internal Customer classification

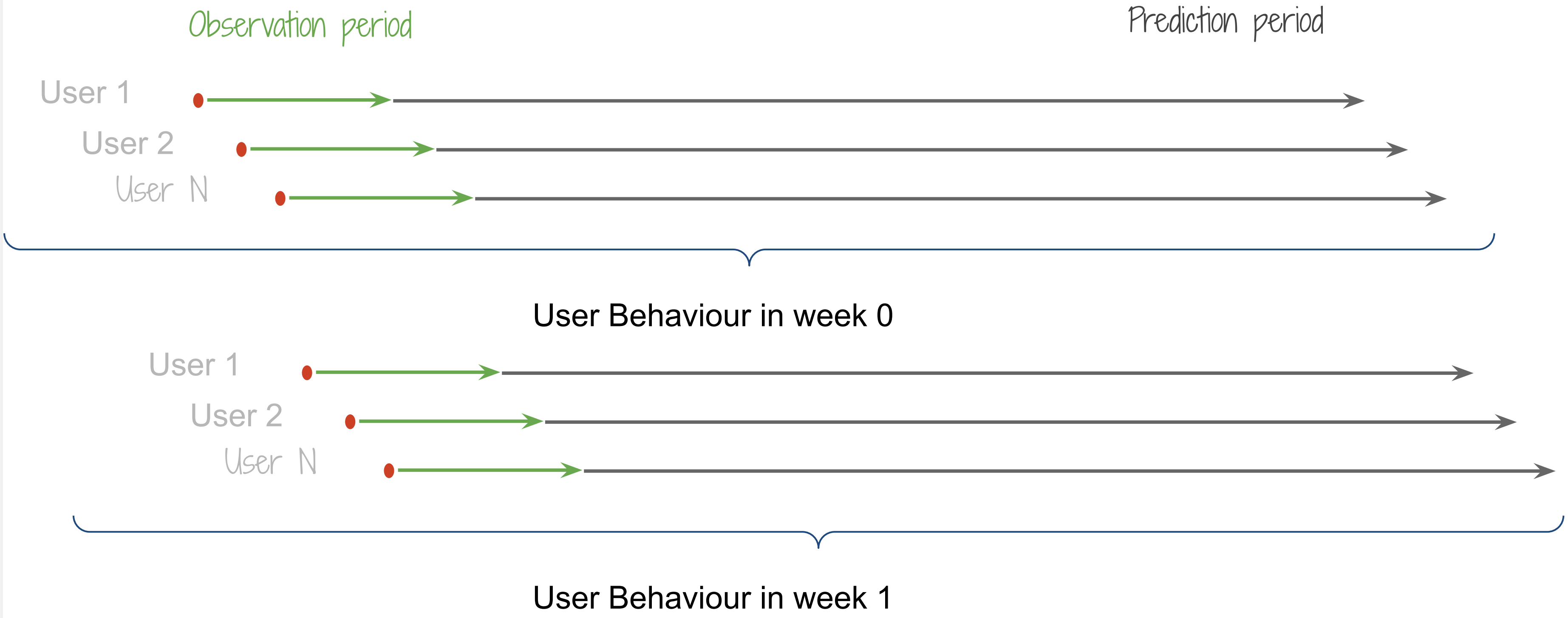
Transactional features

- Time on site

Model design - ‘**Keep calm and let the data talk**‘



Como incrementar el número de observaciones



6 pasos para conquistar Data Analytics y AI

STEP 1

Hacer las preguntas adecuadas

STEP 2

Research Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

Capturar datos

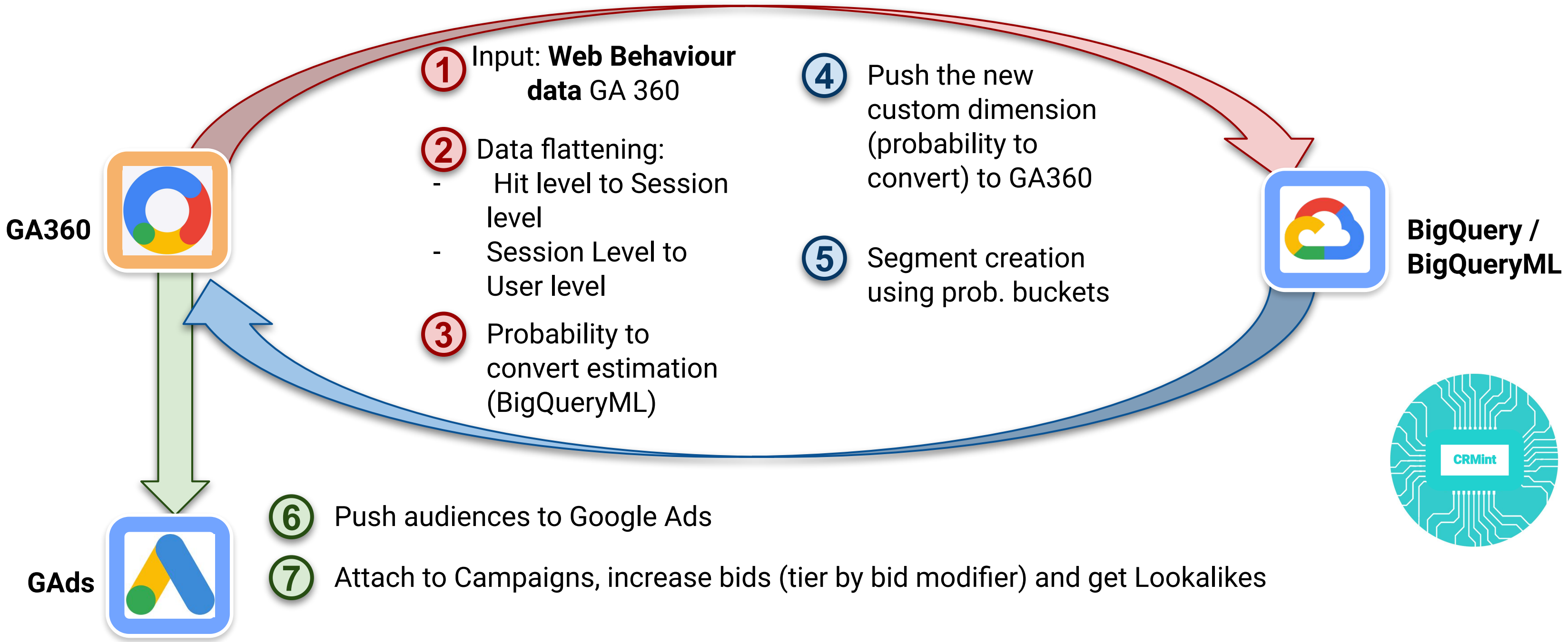
STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados

What - Architecture - ‘**BQML & CRMint as factors of time to market reduction**‘



Note: For the sake of simplicity, CRMInt is not showed, but it’s the central piece in the automatization of this process.

6 pasos para conquistar Data Analytics y AI

STEP 1

Hacer las preguntas adecuadas

STEP 2

Research
Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

Capturar datos

STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados

ML For Analysts | How to evaluate a model

		ML System Says	
		Cat	No Cat
Truth	Cat	True Positive #TP	False Negative #FN
	No Cat	False Positive #FP	True Negative #TN

Model design - ‘**Keep calm and let the business talk**‘

Consideration

-Custom Dimensions extracted from business are strong predictors of the outcome
- Feature cross make a huge difference in linear models (logistic regression) - be careful with overfitting

Extraction

-Make sure the you are capturing accurate data
- Use as much training volume as you can (include all month data, in different timeFrames, if not enough transactions use micro - conversions)

Analysis

- Avoid Signals correlated (to avoid colineality and improve explainability of the model)
-Avoid Signals that do not affect the model
-Use strategy for nulls (Imputation)
-Bucketize features to avoid outliers)

Modelization

-Use Regularization parameters (L1,L2)
-Use different hyperparameters (learn rate)
- Separate Training, evaluation, test data

6 pasos para conquistar Data Analytics y AI

STEP 1

Hacer las preguntas adecuadas

STEP 2

Research
Y Roadmap

STEP 3

Modelización y búsqueda de variables (Prueba de concepto)

STEP 4

Capturar datos

STEP 5

Analizar los datos

STEP 6

Presentar y actuar acorde los resultados

"El uso de algoritmos de **Machine Learning** nos ha permitido dar un salto cualitativo en la **optimización de nuestras campañas** mejorando el ROAS y reduciendo coste unitario."

Los resultados

Con la puesta en marcha de esta acción, Iberia mejora 3 veces el ROAS en términos de búsqueda genéricos como "vuelos" o "vuelos baratos" y 2,6 veces en genéricas relacionadas con rutas, del tipo "volar de Roma a Madrid". Además, el algoritmo pudo identificar el 20% de los usuarios que darán el 60% de las conversiones y del valor de la venta. Otro de los grandes logros es haber identificado, usando el poder de las Audiencias de Google, a un potencial de 2 millones de usuarios similares a los clientes de alto valor.

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

Fin (🎉🎉)

Agenda

01 Introducción ML para analistas

02 BigQuery Machine Learning

03 CRM int App engine Application

04 Caso Práctico Iberia

Break (🎉🎉)

05 Modelo Propensión a Compra según navegación Web

Fin (🎉🎉)

Hackathon Overview

01 Data Transformation in BigQuery GA 360

02 Train model through BigQueryML

03 Evaluate Model with BigQueryML

04 Test the model through BigQuery

05 Put into production with CRM int

Hackathon Overview

01 Data Transformation in BigQuery GA 360

02 Train model through BigQueryML

03 Evaluate Model with BigQueryML

04 Test the model through BigQuery

05 Put into production

1. Data Transformation- ‘**From Hit to User level**‘

```
----------------------DATA TRANSFORMATION----------------

with prediction_dates as (select date from `bigquery-public-data.google_analytics_sample.ga_sessions_*` where geoNetwork.country = "United States"   and
_TABLE_SUFFIX BETWEEN '20170601' AND '20170615' group by 1),          observation_dates as (select date from `bigquery-public-
data.google_analytics_sample.ga_sessions_*` where geoNetwork.country = "United States"   and _TABLE_SUFFIX BETWEEN '20170501' AND '20170531'  group by
1),       prediction_transactions as ( select fullvisitorId as idtran from `bigquery-public-data.google_analytics_sample.ga_sessions_*` where
geoNetwork.country = "United States"   and _TABLE_SUFFIX BETWEEN '20170601' AND '20170615' and date in (select date from prediction_dates) and
totals.transactions >= 1   -- and fullvisitorid in (select id from ids_more_visits)
      group by 1),
     basetable as (
           select * from       `bigquery-public-data.google_analytics_sample.ga_sessions_*` where geoNetwork.country = "United States"   and
_TABLE_SUFFIX BETWEEN '20170501' AND '20170531' and date in (select date from observation_dates)  -- and fullvisitorid in (select id from
ids_more_visits)
      ),
     lastday as ( select date from observation_dates group by 1 order by 1 desc limit 1 ),
     last2day as ( select date from observation_dates group by 1 order by 1 desc limit 2 ),
     last7day as ( select date from observation_dates group by 1 order by 1 desc  limit 7 ),
     last15day as ( select date from observation_dates group by 1 order by 1 desc  limit 15 ),
     source_table as (
         select concat(cast(visitStarttime as string),
         cast(fullvisitorId as string)) as id,
         date,
         visitStarttime,
         fullvisitorId,
         totals.pageviews,
         totals.timeOnsite,
         totals.transactions,
         totals.sessionQualityDim,
         device.operatingSystem,
         device.browser,
         device.deviceCategory,
         channelGrouping from basetable,
         unnest(hits) as hit
--where fullvisitorId --not in (select idobs from observation_transactions )
group by 1,2,3,4,5,6,7,8,9,10,11,12)
```

Hackathon Overview

01 Data Transformation in BigQuery GA 360

02 Train model through BigQueryML

03 Evaluate Model with BigQueryML

04 Test the model through BigQuery

05 Put into production

2. Model Training- ‘**Train a model with transformed data**‘

```sql
CREATE OR REPLACE MODEL `PARTNER_TRAINING.sample_model_1year_v2`
OPTIONS(model_type='logistic_reg',
learn_rate_strategy='constant',
--learn_rate_strategy='line_search',
data_split_method= 'random',
data_split_eval_fraction = 0.15,
learn_rate= 0.6,
l1_reg=0.15,
auto_class_weights = true
) AS
SELECT
  device,
  OS,
  visits,
  visits_lastday,
  visits_last2day,
  visits_last7day,
  pageviews,
  pageviews_lastday,
  pageviews_last2day,
  pageviews_last7day,
  pagedepth,
  pagedepth_lastday,
  pagedepth_last2day,
  pagedepth_last7day,
  avg_SQ,
  max_SQ,
  avg_QS_lastday,
  avg_QS_last2day,
  avg_QS_last7day,
  max_QS_lastday,
  max_QS_last2day,
  max_QS_last7day,
  max_timeOnSite,
  max_timeOnSite_lastday,
  max_timeOnSite_last2day,
  max_timeOnSite_last7day,
  avg_timeOnSite_lastday,
  avg_timeOnSite_last2day,
  avg_timeOnSite_last7day,
```

Hackathon Overview

01 Data Transformation in BigQuery GA 360

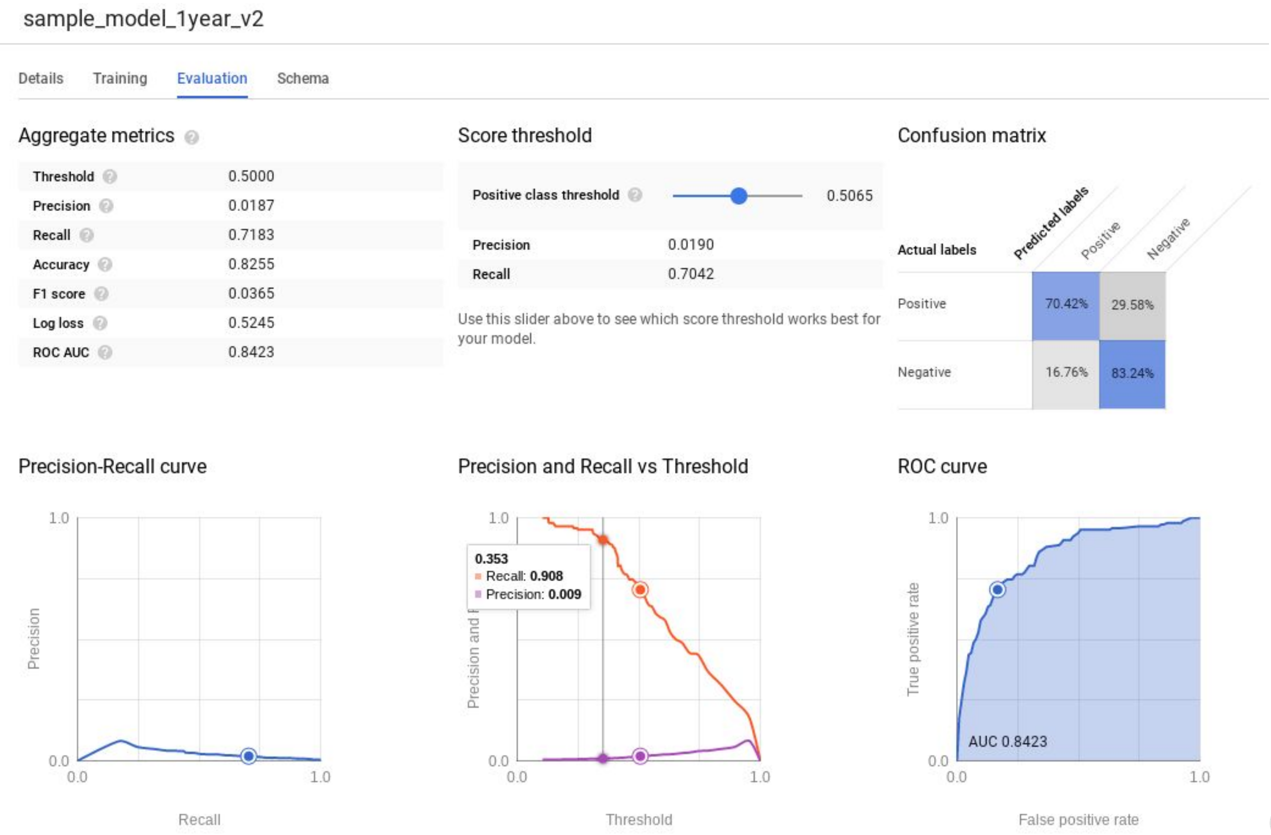
02 Train model through BigQueryML

03 Evaluate Model with BigQueryML

04 Test the model through BigQuery

05 Put into production

3. Evaluate the model- ‘**ok… but is my model good?**‘



Evaluate the model- ‘**ok… but is my model good?**‘

What are the weights of the model?

Categorical Variables

1	device	*null*	mobile	-0.6155886202312307
			tablet	-0.5900274731026636
			desktop	-0.05134029618402976
2	OS	*null*	Firefox OS	-0.3053143590223346
			Macintosh	0.3131824993832482
			Windows	-0.46318863608673144
			Samsung	-0.630697200703975
			(not set)	-0.8567187773904958
			Xbox	-0.8808040886937785
			Nintendo WiiU	-0.5661990188855418
			Android	-0.6416918546062974
			Chrome OS	0.145032012028166
			SunOS	-0.22136673673976623

Numerical Variables

visits	0.1949046354916802
visits_lastday	0.07850133858974218
visits_last2day	0.09740174861823442
visits_last7day	0.17540645382491488
pageviews	0.017399601722247648
pageviews_lastday	0.012125817223498321
pageviews_last2day	0.012985186854796061
pageviews_last7day	0.013371970147292253
pagedepth	0.023689508651841624
pagedepth_lastday	0.0015114166616272753
pagedepth_last2day	0.004517319861942276
pagedepth_last7day	0.01582458947055531

Evaluate the model- ‘**is there colineality?**‘

```
with devices as (

select clientId, case when OS in ('Android','iOS','     Windows Phone') then 1 else 0 end as
mobile_os,
case when device = 'mobile' then 1 else 0 end as mobile_device

from KSCHOOL.test_dataset )

select corr(mobile_os,mobile_device) as correlation_mobile from devices
```

Evaluate the model- ‘**ok… but is my model good?**‘ ⟶ Check Feature weights to export the model wherever you want

```
SELECT
    *
  FROM
   ML.WEIGHTS(MODEL `TRAININGBQML_GOOGLE.model_1`)
```

$$P = \frac{e^{\beta 0x + \beta 1x1 + \beta 2x2 + \ldots + \beta ixi}}{1 + e^{\beta 0x + \beta 1x1 + \beta 2x2 + \ldots + \beta ixi}}$$

Hackathon Overview

01 Data Transformation in BigQuery GA 360

02 Train model through BigQueryML

03 Evaluate Model with BigQueryML

04 Test the model through BigQuery

05 Put into production

4. Test the model- ‘**ok… but is my model generalizing?**‘

```
SELECT
  clientId,predicted_label,label
FROM
  ml.PREDICT(MODEL `google.com:travel-dashboards.PARTNER_TRAINING.sample_model_1year_v2`, (
SELECT
  *
FROM
  `google.com:travel-dashboards.PARTNER_TRAINING.20170601` )
  )--where label = 1
```

Evaluate the model- ‘**ok... but is my model good?**‘

ROC Recall Vs false positive rate

```
select sum(if(predicted_label=1 and label=1,1,0)) as ok, sum(label) as label,sum(predicted_label) as predicted_label,count(*) as registers from TRAININGBQML_GOOGLE.evaluation_test
```

		PREDICTION	
		Pos	**Neg**
ACTUAL	**Pos**	44	26
	Neg	2,487	13,443

$$\text{lift} = \frac{(TP/(TP+FP)}{(TP+FN)/(TP+TN+FP+FN)}$$

Positive observations	16%

Precision	2%
Recall	63%
Accuracy	84%
Total Lift	3.97

Hackathon Overview

01 Data Transformation in BigQuery GA 360

02 Train model through BigQueryML

03 Evaluate Model with BigQueryML

04 Test the model through BigQuery

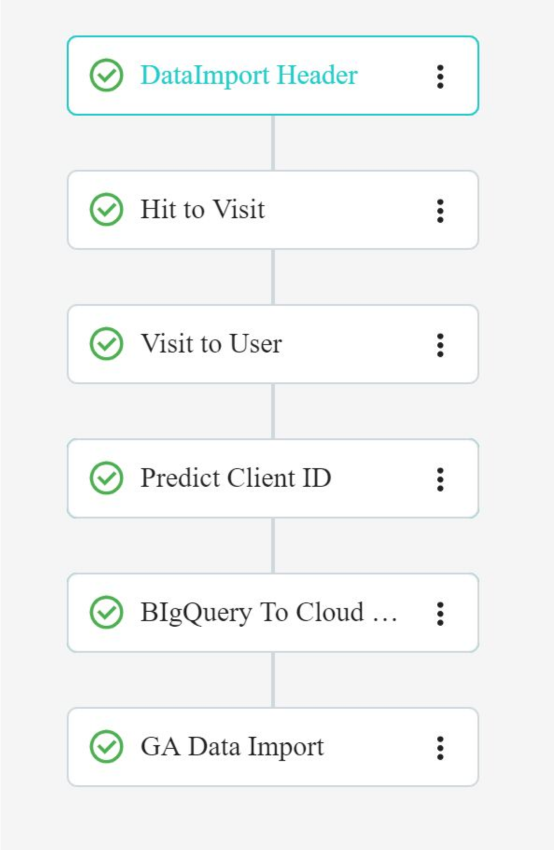
05 Put into production

5. Put into production- ‘**Let CRMint do the work for you**‘

5. Put into production- ‘**Deploy CRM int**‘

Link

5. Put into production- ‘**Create a Pipeline for audiences**‘



Create an empty table to simulate required fields in Data Import - BQ Launcher

Transform observation data of the last 30 days from hit to visit
BQ Launcher

Transform observation data of the last 30 days from Visit to Hit
BQ Launcher

Launch the ML predict to guess probabilities by clientId
BQ Launcher

Create a CSV from BQ to Cloud Storage (eliminating Headers)
BQ to Cloud Storage

Create a Data Import
Cloud Storage to Data Import

Thank you!