Google Cloud

Predicting Visitor Purchases using BigQuery ML

Agenda

Introduction to BigQuery

- Fast SQL Query Engine
- Managed Storage for Datasets

Insights from Geographic data

Machine Learning on Structured Data

- Choosing the right model type
- Scenario: Predicting Customer Lifetime Value

Creating ML models with SQL

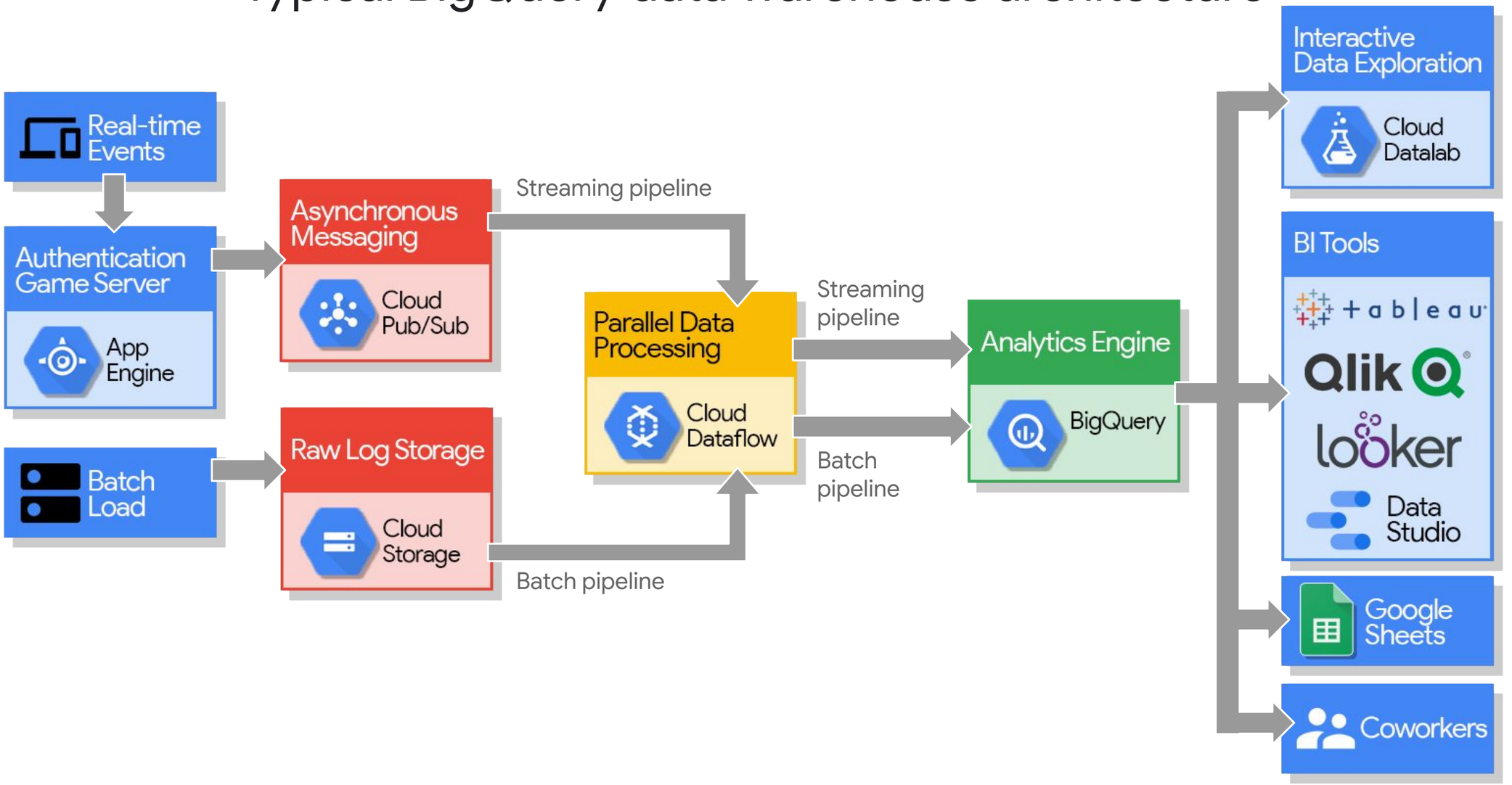
- Introduction to BigQuery ML
- ML project phases
- Key features walkthrough

BigQuery is a petabyte-scale fully-managed data warehouse

Google BigQuery

1. It’s serverless
2. Flexible pricing model
3. Data encryption and security
4. Geospatial data types & functions
5. Foundation for BI and AI

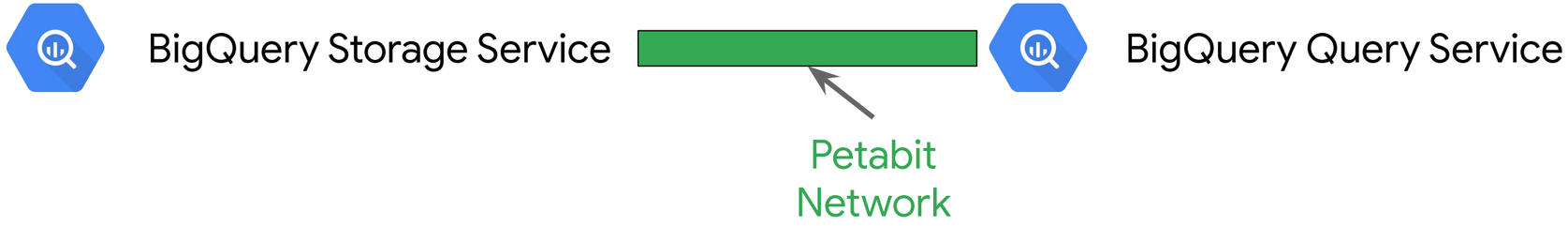
Typical BigQuery data warehouse architecture



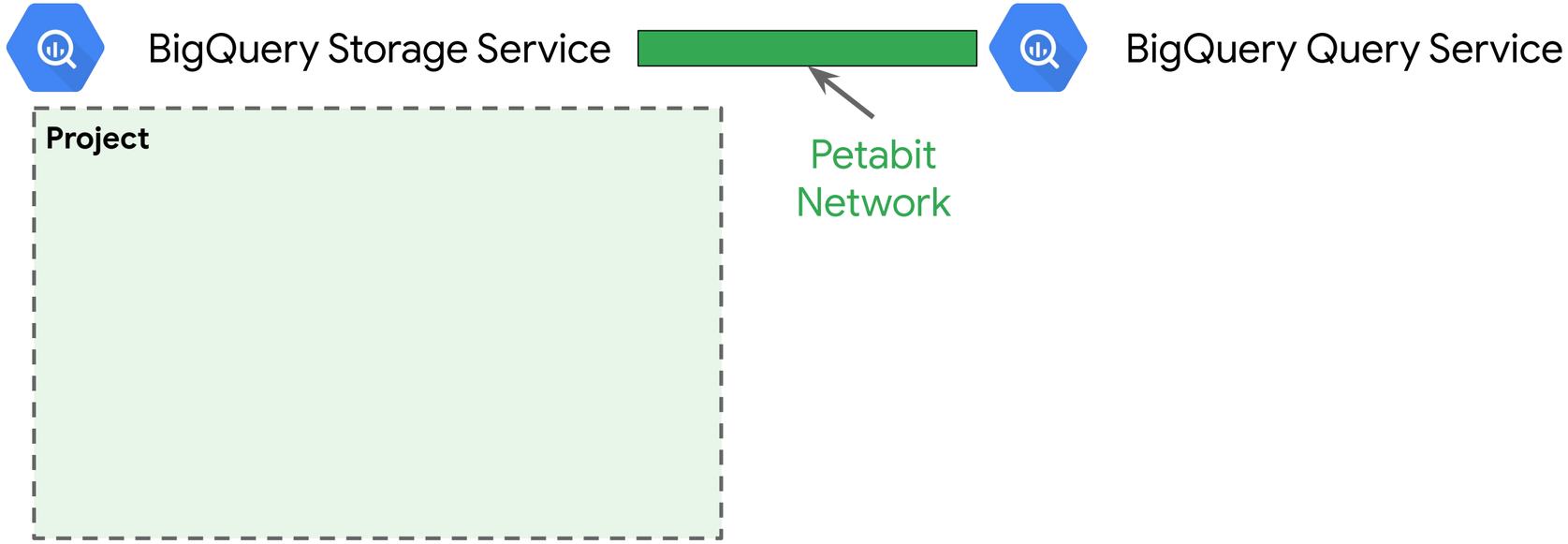
BigQuery is two services in one

1. **Fast SQL Query Engine**
2. Managed storage for datasets

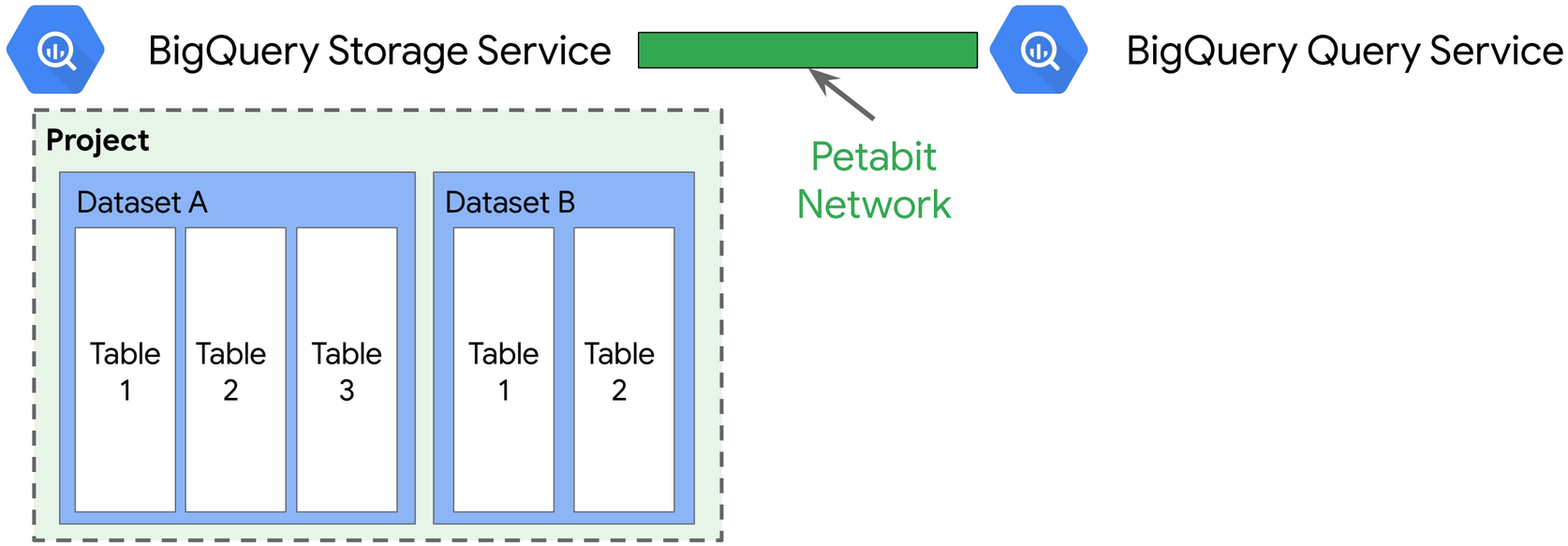
How does BigQuery work?



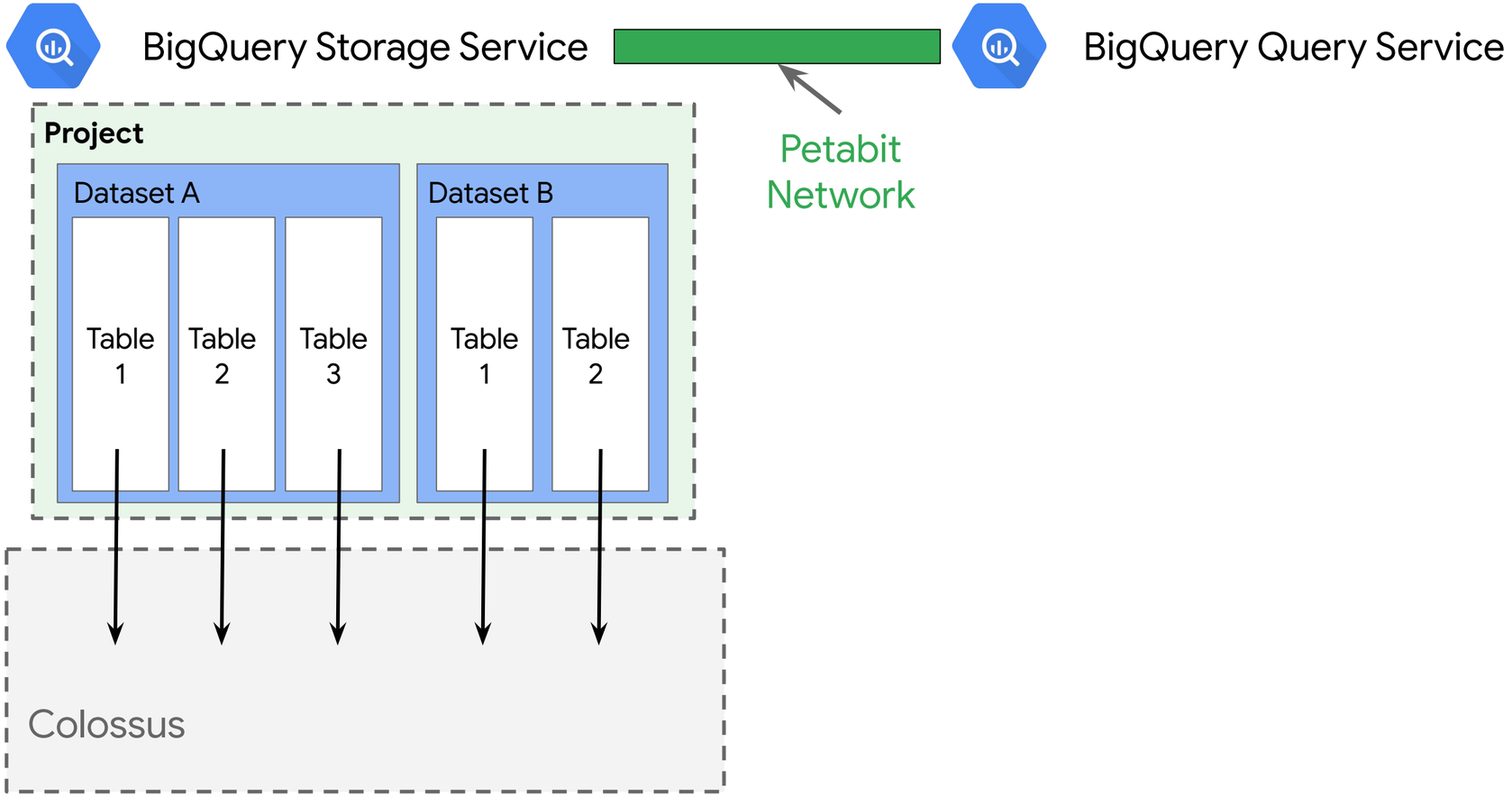
How does BigQuery work?



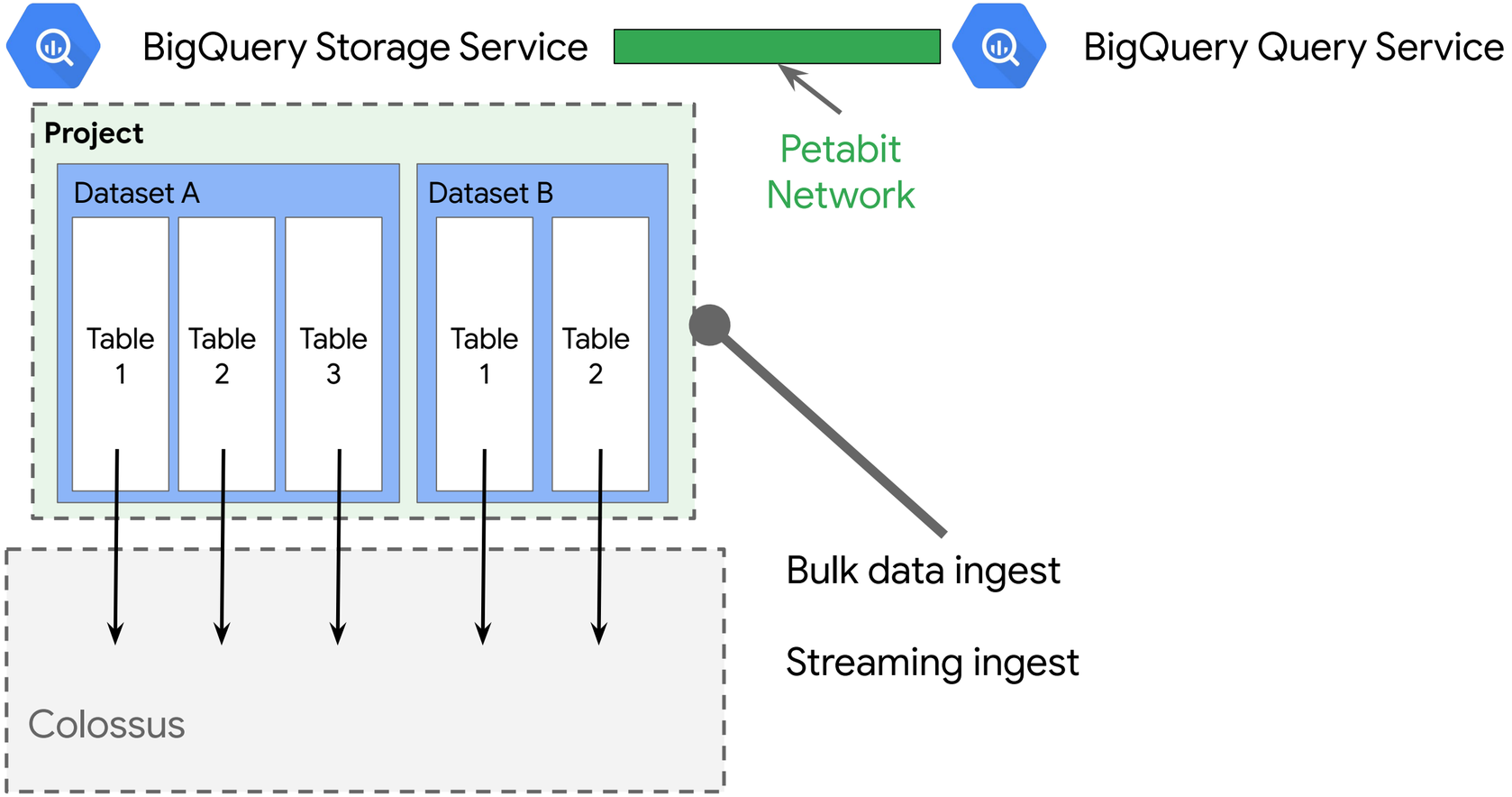
How does BigQuery work?



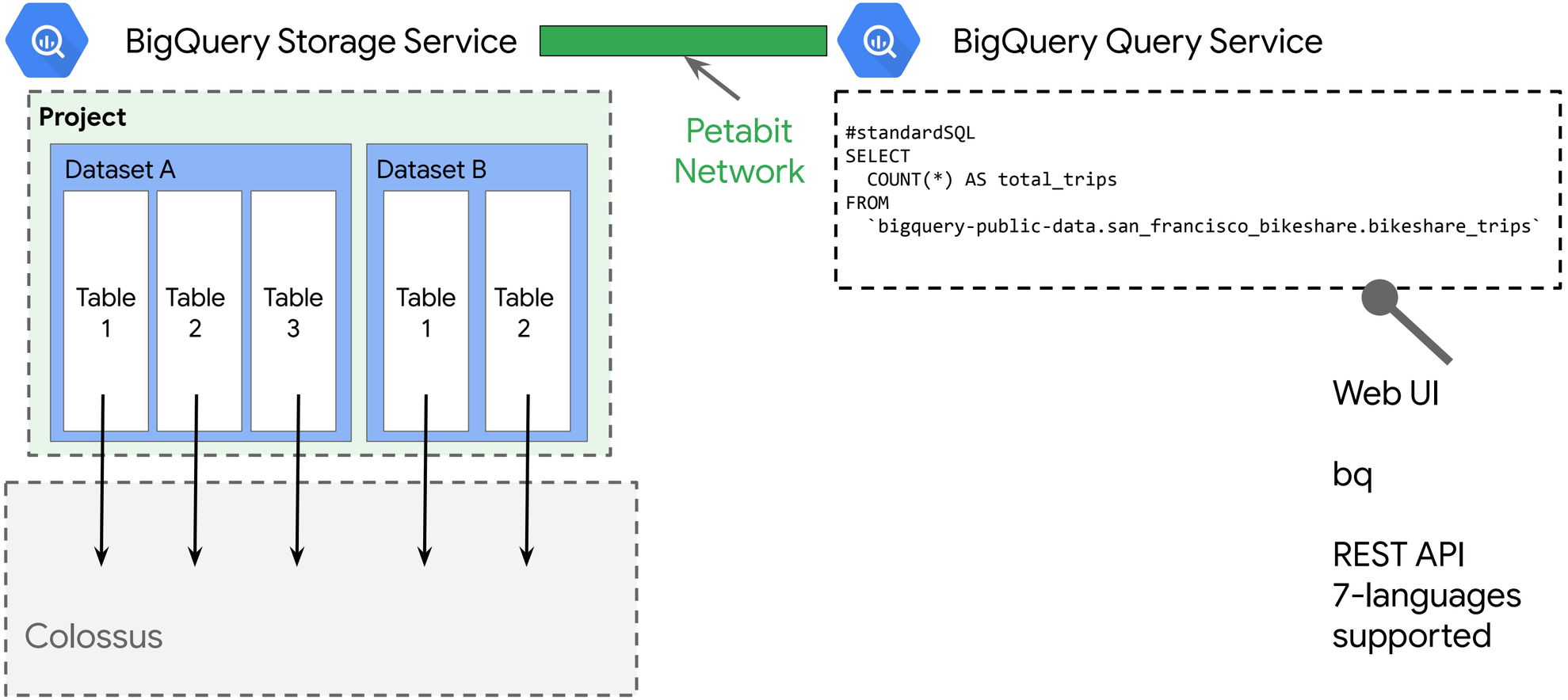
How does BigQuery work?



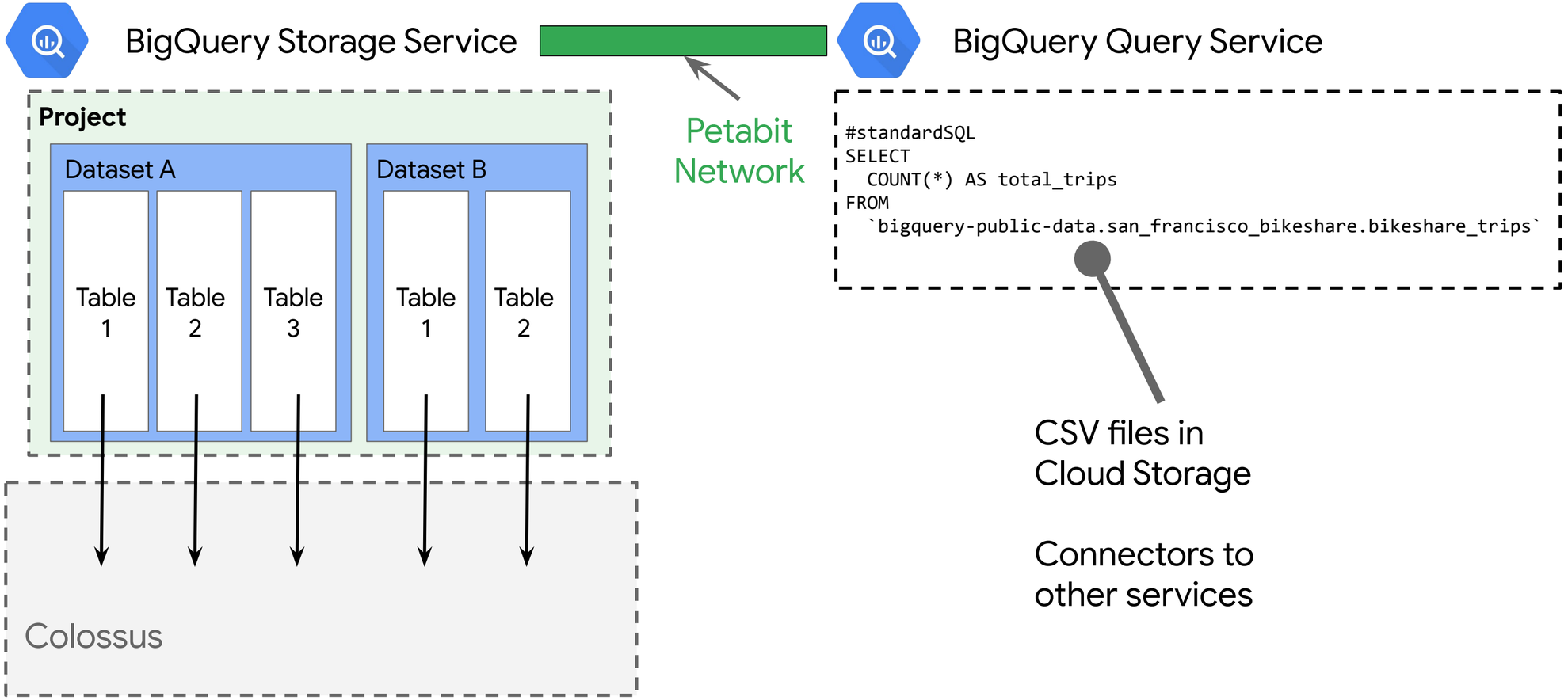
How does BigQuery work?



How does BigQuery work?



How does BigQuery work?



BigQuery supports standard SQL queries for analysis

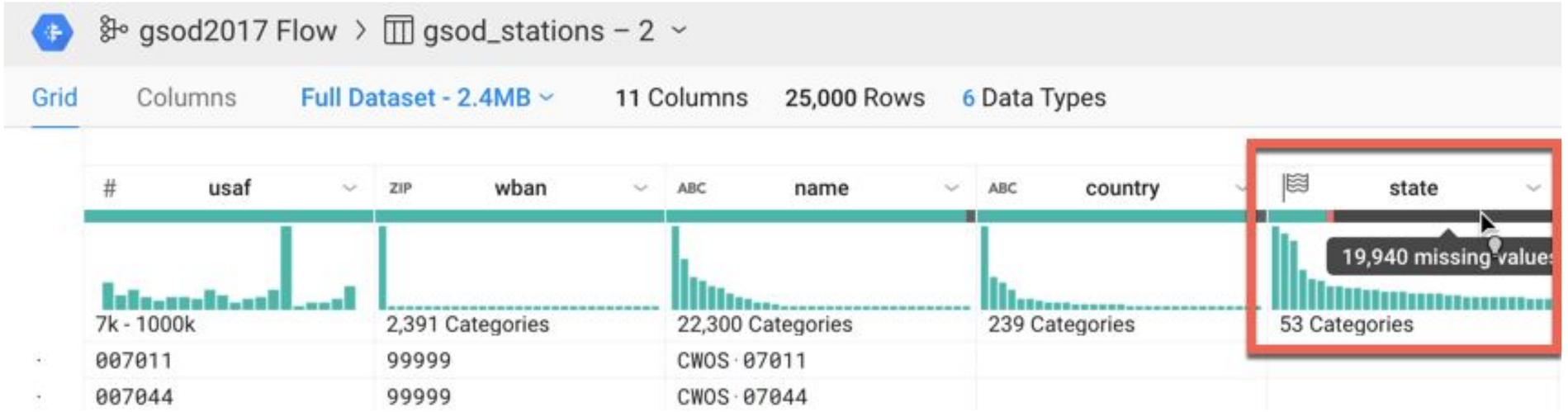
```
#standardSQL
SELECT
  COUNT(*) AS total_trips
FROM
  `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
```

Row	total_trips
1	1947419

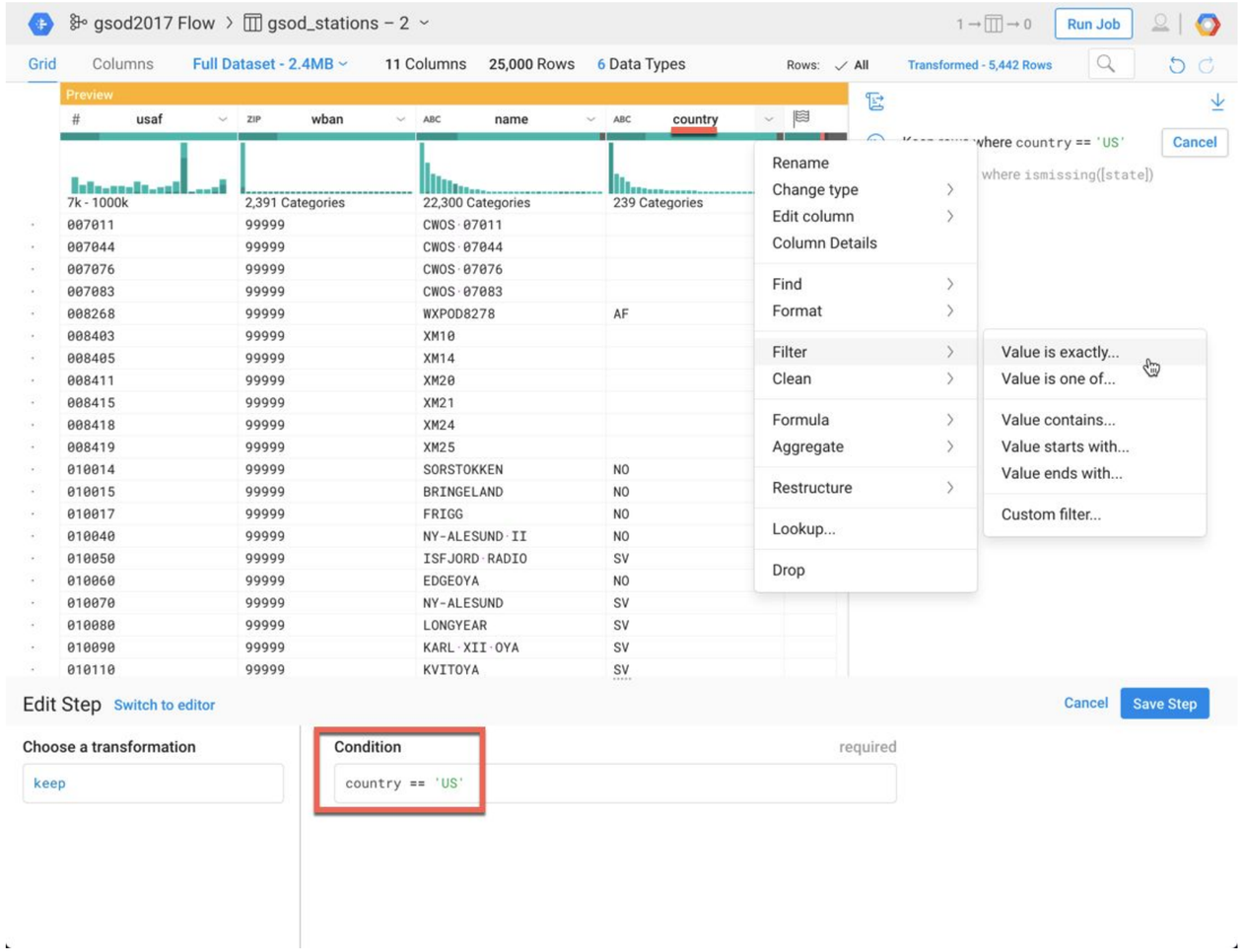
Common SQL operations include deduplication and cleansing

```
# Top 5 stations for casual bike share users
SELECT
  start_station_name,
  COUNT(*) AS total_trips
FROM
  `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`

WHERE c_subscription_type = 'Customer'
AND member_birth_year IS NOT NULL

GROUP BY start_station_name
ORDER BY total_trips DESC
LIMIT 5
```

Cloud Dataprep: Cleaning and transforming data with a UI



Common SQL operations include deduplication and cleansing



Schedule your queries to run with @run_time

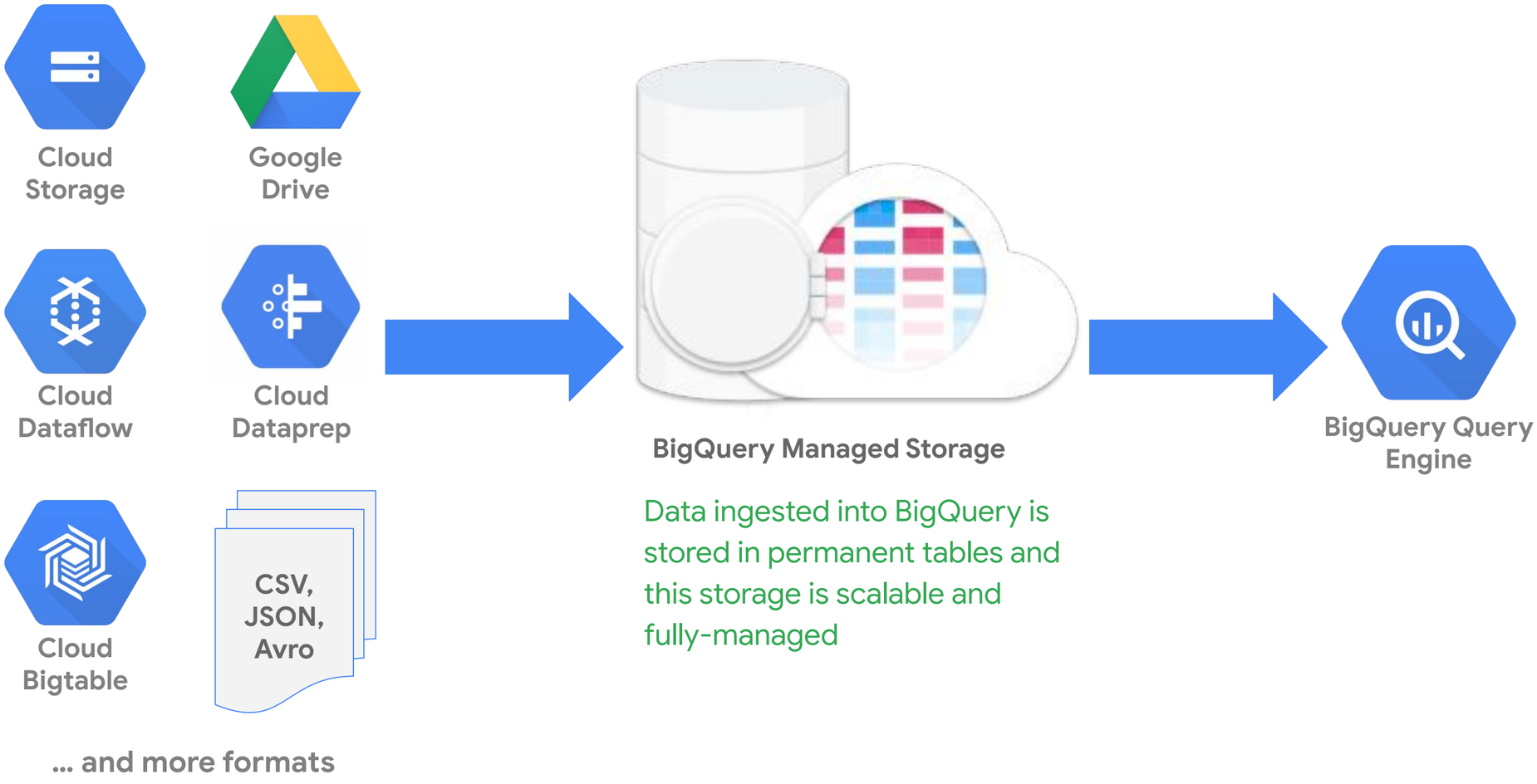
```
SELECT @run_time AS time,
  title,
  author,
  text
FROM `bigquery-public-data.hacker_news.stories`
LIMIT
  1000
```

BigQuery is two services in one

Google BigQuery

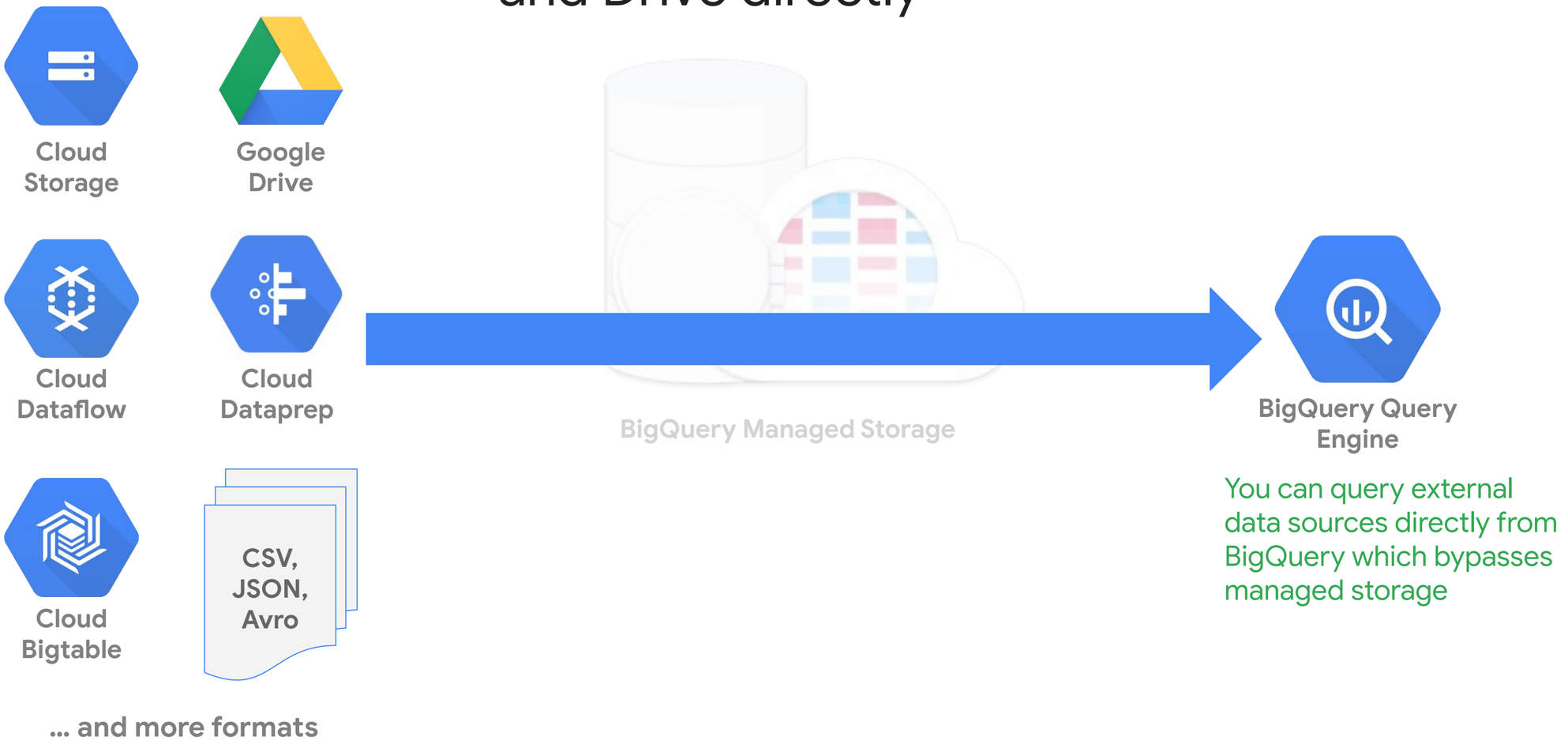
1. Fast SQL Query Engine
2. **Managed storage for datasets**

Use native BigQuery storage for the highest performance



Data ingested into BigQuery is stored in permanent tables and this storage is scalable and fully-managed

BigQuery can query external (aka federated) data sources in GCS and Drive directly



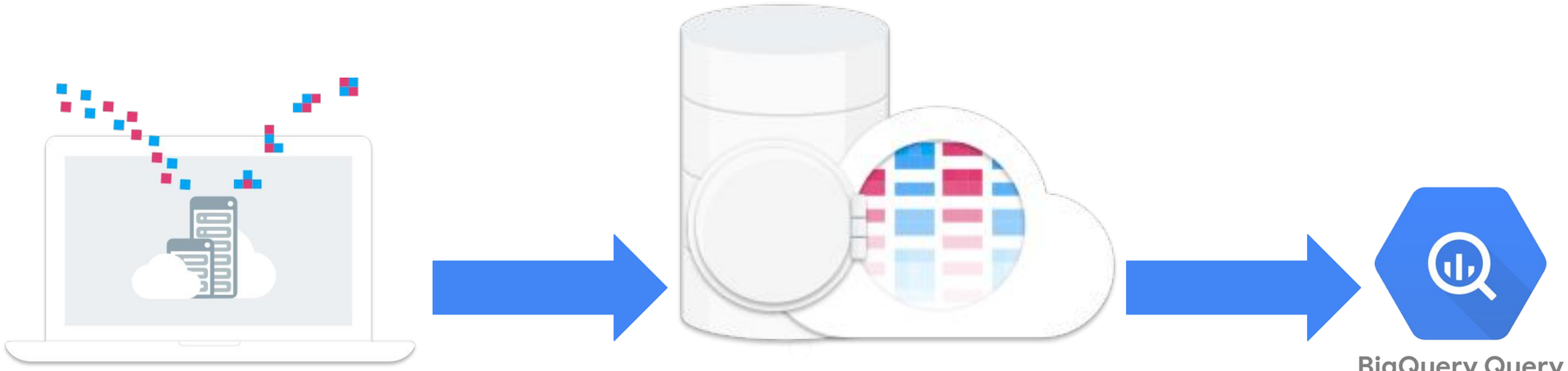
You can query external data sources directly from BigQuery which bypasses managed storage

Demo

Querying Google Sheets from BigQuery

Federated queries: Pros and Cons

Streaming records into BigQuery through the API



Streaming data allows you to query data without waiting for a full batch load

Compare streaming options for your use case

BigQuery

or

Cloud Dataflow

Challenge: Normalization vs Denormalization for reporting

Payments

Event Timestamps

Pickups

Dropoffs

Use STRUCTs and ARRAYS for consolidated reporting

Job information | Results | JSON | Execution details

Row	order_id	service_type	payment_method	event.status	event.time	pickup.latitude	pickup.longitude	destination.latitude	destination.longitude	total_distance_km
1	CR-6098	GO_CAR	CASH	DRIVER_FOUND	2018-12-31 01:45:15.667 UTC	-6.9228116	107.5930599	-6.912966	107.609604	2.9749999046325684
				COMPLETED	2018-12-31 02:06:56.894 UTC					
				PICKED_UP	2018-12-31 02:05:40.075 UTC					
				CREATED	2018-12-31 01:44:59.239 UTC					

BigQuery natively supports ARRAYs as data types

Use STRUCTs and ARRAYS for consolidated reporting

Job information | Results | JSON | Execution details

Row	order_id	service_type	payment_method	event.status	event.time	pickup.latitude	pickup.longitude	destination.latitude	destination.longitude	total_distance_km
1	CR-6098	GO_CAR	CASH	DRIVER_FOUND	2018-12-31 01:45:15.667 UTC	-6.9228116	107.5930599	-6.912966	107.609604	2.9749999046325684
				COMPLETED	2018-12-31 02:06:56.894 UTC					
				PICKED_UP	2018-12-31 02:05:40.075 UTC					
				CREATED	2018-12-31 01:44:59.239 UTC					

Schema | Details | Preview

Field name	Type	Mode
order_id	STRING	NULLABLE
service_type	STRING	NULLABLE
payment_method	STRING	NULLABLE
event	RECORD	REPEATED
event. **status**	STRING	NULLABLE
event. **time**	TIMESTAMP	NULLABLE
pickup	RECORD	NULLABLE
pickup. **latitude**	FLOAT	NULLABLE
pickup. **longitude**	FLOAT	NULLABLE

BigQuery natively supports ARRAYs as data types (and STRUCTs too)

Agenda

Introduction to BigQuery

- Fast SQL Query Engine
- Managed Storage for Datasets

Insights from Geographic data

Machine Learning on Structured Data

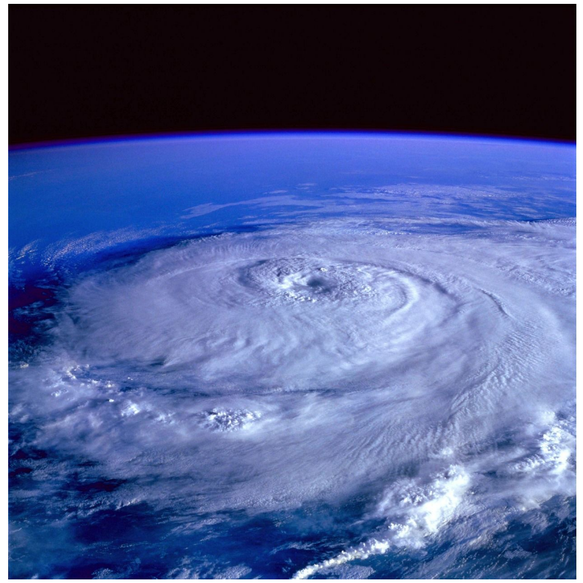
- Choosing the right model type
- Scenario: Predicting Customer Lifetime Value

Creating ML models with SQL

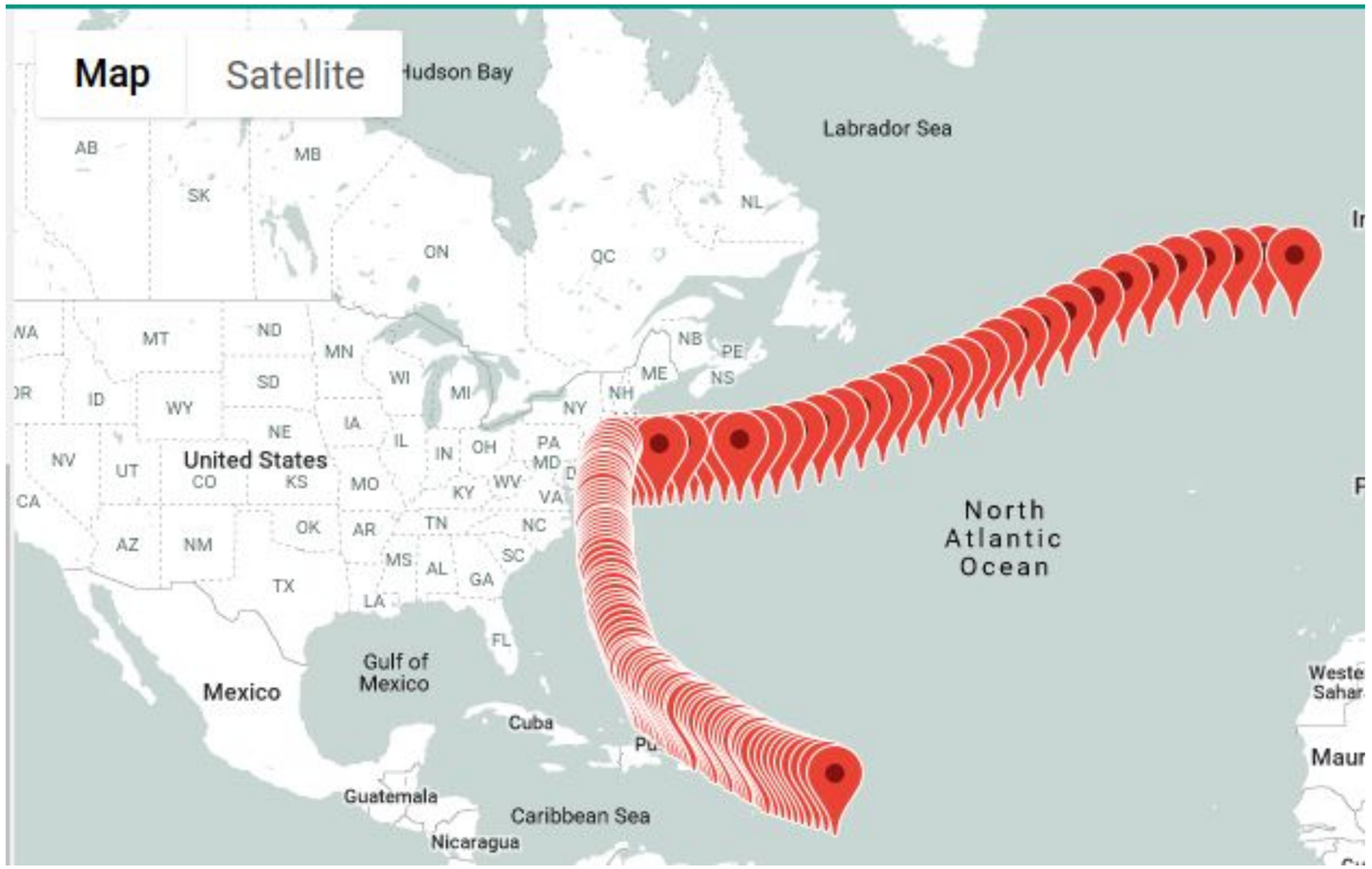
- Introduction to BigQuery ML
- ML project phases
- Key features walkthrough

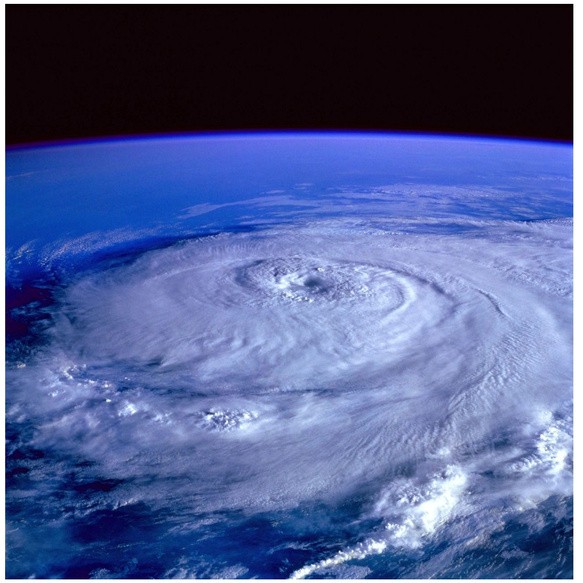
Unlock your geographic insights with BigQuery GIS functions

```
#standardSQL
SELECT
  ST_GeogPoint(longitude, latitude) AS point,
  name,
  iso_time,
  dist2land,
  usa_wind,
  usa_pressure,
  usa_sshs,
  (usa_r34_ne + usa_r34_nw + usa_r34_se + usa_r34_sw)/4 AS radius_34kt,
  (usa_r50_ne + usa_r50_nw + usa_r50_se + usa_r50_sw)/4 AS radius_50kt
FROM
  `bigquery-public-data.noaa_hurricanes.hurricanes`
WHERE
  name LIKE '%MARIA%'
  AND season = '2017'
  AND ST_DWithin(ST_GeogFromText('POLYGON((-179 26, -179 48, -10 48, -10 26,
-100 -10.1, -179 26))'),
    ST_GeogPoint(longitude, latitude), 10)
ORDER BY
  iso_time ASC
```



Unlock your geographic insights with BigQuery GIS functions





Agenda

Introduction to BigQuery

- Fast SQL Query Engine
- Managed Storage for Datasets

Insights from Geographic data

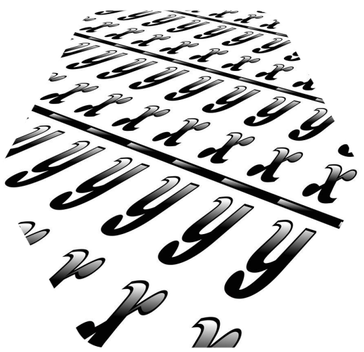
Machine Learning on Structured Data

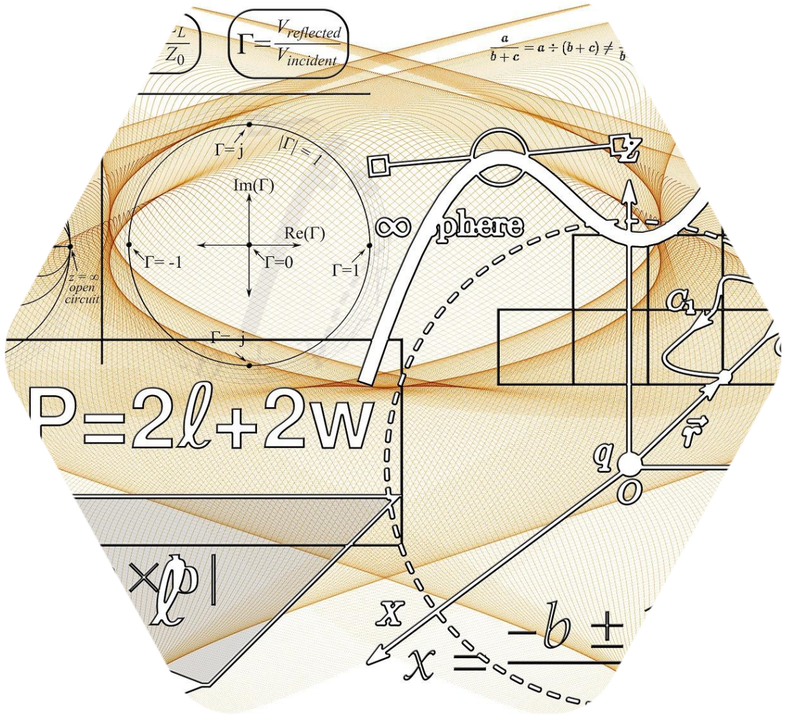
- Choosing the right model type
- Scenario: Predicting Customer Lifetime Value

Creating ML models with SQL

- Introduction to BigQuery ML
- ML project phases
- Key features walkthrough

The popular imagination of what ML is





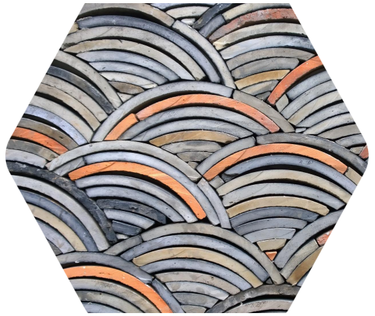
Lots of data

Complex mathematics in multidimensional spaces

Magical results

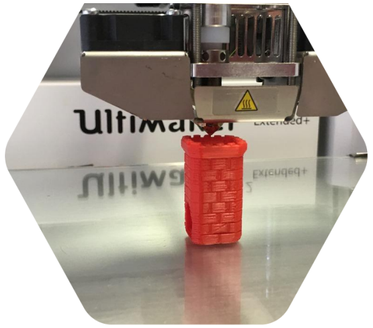
In reality, ML is...

Collect data



Organize data

Create model



Experiment a lot

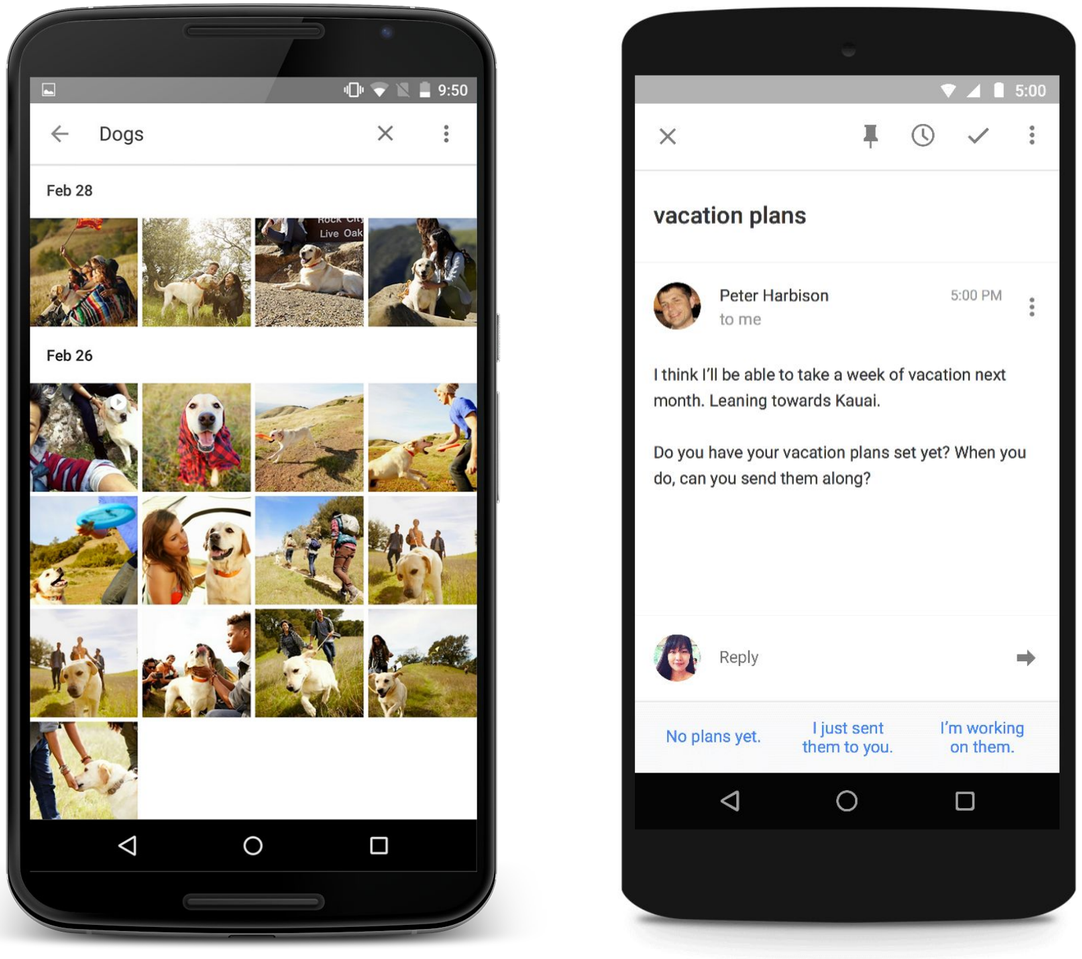
Magical results

When you hear “AI or ML,” you probably think of:

Image models

Sequence models

Neural Networks



The most common ML models at Google are those that operate on structured data

ML on structured data drives value

Type of network	# of network layers	# of weights	% of deployed models
MLP0	5	20M	61%
MLP1	4	5M	
LSTM0	58	52M	29%
LSTM1	56	34M	
CNN0	16	8M	5%
CNN1	89	100M	

Source (2017):
https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

Machine Learning =

Examples, not rules

A quick example

Predicting customer LTV **with a ML model**

Let’s predict the lifetime value of an ecommerce customer

ABC **hits_product_v2ProductName**

177 Categories

Google Men's Vintage Badge Tee Black
Google Women's 1/4 Zip Performance Pullover Two-Tone Blue
Google Men's Lightweight Microfleece Jacket Black
Google Men's Lightweight Microfleece Jacket Black
Google Women's Quilted Insulated Vest Black
Google Women's Lightweight Microfleece Jacket
Google Women's Lightweight Microfleece Jacket
Women's Weatherblock Shell Jacket Black
Men's Weatherblock Shell Jacket Black

Google Analytics provides us with aggregated site visit metrics

Results | Details

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions
1	7813149961404844386	79	1395	138	479.63	6245720000	67
2	7713012430069756739	2	514	6	1954.33	181940000	35
3	6760732402251466726	30	868	41	723.55	4812820000	34
4	5526675926038480325	1	466	1	7013.0	87960000	25
5	1957458976293878100	148	4303	284	796.46	77113430000	22
6	4983264713224875783	2	366	4	3807.5	74850000	21
7	2402527199731150932	28	559	31	906.61	3270100000	19

In ML terms, an instance (or observation) is a *row* of data

Results | Details

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days
1	6007196403211981721	8	147	11	772.5	null	null	7.5	2016-08-04	2017-07-15	345
2	7587138749751940102	9	94	9	312.33	24380000	1	1.0	2016-08-03	2017-07-14	345
3	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344
4	9557989866096732580	3	18	3	356.5	null	null	1.0	2016-08-03	2017-07-13	344
5	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343
6	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343
7	1957458976293878100	148	4303	284	796.46	77113430000	22	1.5	2016-08-04	2017-07-12	342
8	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340

What we are trying to predict for is the *label*

Results | Details

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days
1	6007196403211981721	8	147	11	772.5	null	null	7.5	2016-08-04	2017-07-15	345
2	7587138749751940102	9	94	9	312.33	24380000	1	1.0	2016-08-03	2017-07-14	345
3	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344
4	9557989866096732580	3	18	3	356.5	null	null	1.0	2016-08-03	2017-07-13	344
5	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343
6	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343
7	1957458976293878100	148	4303	284	796.46	77113430000	22	1.5	2016-08-04	2017-07-12	342
8	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340

Here we are predicting the lifetime revenue (number)

Labels could also be a discrete class of customer like “high value”

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days	label
1	7587138749751940102	9	94	9	312.33	24380000	1	1.0	2016-08-03	2017-07-14	345	High Value Customer
2	6007196403211981721	8	147	11	772.5	null	null	7.5	2016-08-04	2017-07-15	345	
3	9557989866096732580	3	18	3	356.5	null	null	1.0	2016-08-03	2017-07-13	344	
4	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344	
5	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343	High Value Customer
6	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343	
7	1957458976293878100	148	4303	284	796.46	77113430000	22	1.5	2016-08-04	2017-07-12	342	High Value Customer
8	9801276214964695322	79	462	106	219.44	null	null	1.5	2016-08-01	2017-07-07	340	
9	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340	
10	0084834161383601528	7	97	7	258.0	69260000	2	2.0	2016-08-04	2017-07-10	340	High Value Customer
11	928398408398925152	40	553	43	285.37	462190000	2	2.0	2016-08-02	2017-07-07	339	High Value Customer
12	351277725820061611	20	60	20	221.33	null	null	1.0	2016-08-05	2017-07-10	339	
13	4143624098732715494	6	13	7	52.5	null	null	1.0	2016-08-03	2017-07-08	339	
14	1927175312147751345	13	180	14	427.21	44970000	1	2.0	2016-08-03	2017-07-08	339	High Value Customer
15	1315772786660606104	28	272	36	340.3	279320000	3	21.25	2016-08-09	2017-07-14	339	High Value Customer

What you are trying to predict for (number or discrete class)
influences the model you will choose

The other columns of data are your potential feature columns

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days	label
1	7587138749751940102	9	94	9	312.33	24380000	1	1.0	2016-08-03	2017-07-14	345	High Value Customer
2	6007196403211981721	8	147	11	772.5	null	null	7.5	2016-08-04	2017-07-15	345	
3	9557989866096732580	3	18	3	356.5	null	null	1.0	2016-08-03	2017-07-13	344	
4	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344	
5	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343	High Value Customer
6	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343	
7	1957458976293878100	148	4303	284	796.4	77113430000	22	1.5	2016-08-04	2017-07-12	342	High Value Customer
8	9801276214964695322	79	1[illegible]	[illegible]	[illegible]	null	null	1.5	2016-08-01	2017-07-07	340	
9	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340	
10	0084834161383601528	7	97	7	258.0	69260000	2	2.0	2016-08-04	2017-07-10	340	High Value Customer
11	928398408398925152	40	553	43	285.37	462190000	2	2.0	2016-08-02	2017-07-07	339	High Value Customer
12	351277725820061611	20	60	20	221.33	null	null	1.0	2016-08-05	2017-07-10	339	
13	4143624098732715494	6	13	7	52.5	null	null	1.0	2016-08-03	2017-07-08	339	
14	1927175312147751345	13	180	14	427.21	44970000	1	2.0	2016-08-03	2017-07-08	339	High Value Customer
15	1315772786660606104	28	272	36	340.3	279320000	3	21.25	2016-08-09	2017-07-14	339	High Value Customer

Feature Columns

You will try to *model* the relationship between the features and your label

What if I don’t know where a new customer will fit?

Historical Training Data (Known LTV

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days	label
1	7587138749751940102	9	94	9	312.33	24380000	1	1.0	2016-08-03	2017-07-14	345	High Value Customer
2	6007196403211981721	8	147	11	772.5	null	null	7.5	2016-08-04	2017-07-15	345	
3	9557989866096732580	3	18	3	356.5	null	null	1.0	2016-08-03	2017-07-13	344	
4	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344	
5	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343	High Value Customer
6	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343	
7	1957458976293878100	148	4303	284	796.46	77113430000	22	1.5	2016-08-04	2017-07-12	342	High Value Customer
8	9801276214964695322	79	462	106	219.44	null	null	1.5	2016-08-01	2017-07-07	340	
9	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340	
10	0084834161383601528	7	97	7	258.0	69260000	2	2.0	2016-08-04	2017-07-10	340	High Value Customer
11	928398408398925152	40	553	43	285.37	462190000	2	2.0	2016-08-02	2017-07-07	339	High Value Customer
12	351277725820061611	20	60	20	221.33	null	null	1.0	2016-08-05	2017-07-10	339	
13	4143624098732715494	6	13	7	52.5	null	null	1.0	2016-08-03	2017-07-08	339	
14	1927175312147751345	13	180	14	427.21	44970000	1	2.0	2016-08-03	2017-07-08	339	High Value Customer
15	1315772786660606104	28	272	36	340.3	279320000	3	21.25	2016-08-09	2017-07-14	339	High Value Customer

Future Data (Unknown LTV)

17	7904807859681747547	3	42	3	1162.0	null	null	1.0	2016-08-05	2017-07-09	338	????????????????????
18	4405445121320750966	51	358	62	517.36	null	null	1.0	2016-08-08	2017-07-12	338	????????????????????
19	1419607020881916790	5	22	5	711.0	null	null	1.0	2016-08-12	2017-07-15	337	????????????????????
20	3862335714593915688	13	92	16	154.23	238000000	1	2.0	2016-08-09	2017-07-12	337	????????????????????

What if I don't know where a new customer will fit?

Historical Training Data (Known LTV

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days	label
1	7587138749751940102	9	94	9	312.33	24380000	1	1.0	2016-08-03	2017-07-14	345	High Value Customer
2	6007196403211981721	8	147	11	772.5	null	null	7.5	2016-08-04	2017-07-15	345	
3	9557989866096732580	3	18	3	356.5	null	null	1.0	2016-08-03	2017-07-13	344	
4	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344	
5	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343	High Value Customer
6	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343	
7	1957458976293878100	148	4303	284	796.46	77113430000	22	1.5	2016-08-04	2017-07-12	342	High Value Customer
8	9801276214964695322	79	462	106	219.44	null	null	1.5	2016-08-01	2017-07-07	340	
9	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340	
10	0084834161383601528	7	97	7	258.0	69260000	2	2.0	2016-08-04	2017-07-10	340	High Value Customer
11	928398408398925152	40	553	43	285.37	462190000	2	2.0	2016-08-02	2017-07-07	339	High Value Customer
12	351277725820061611	20	60	20	221.33	null	null	1.0	2016-08-05	2017-07-10	339	
13	4143624098732715494	6	13	7	52.5	null	null	1.0	2016-08-03	2017-07-08	339	
14	1927175312147751345	13	180	14	427.21	44970000	1	2.0	2016-08-03	2017-07-08	339	High Value Customer
15	1315772786660606104	28	272	36	340.3	279320000	3	21.25	2016-08-09	2017-07-14	339	High Value Customer

Future Data (Unknown LTV)

17	7904807859681747547	3	42	3	1162.0	null	null	1.0	2016-08-05	2017-07-09	338	??????????????????
18	4405445121320750966	51	358	62	517.36	null	null	1.0	2016-08-08	2017-07-12	338	??????????????????
19	1419607020881916790	5	22	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2016-08-12	2017-07-15	337	??????????????????
20	3862335714593915688	13	92	16	154.23	238000000	[illegible]	[illegible]	[illegible]	[illegible]	337	??????????????????

Infer or predict it with a model! →

DATA INSTEAD OF RULES

Your model will learn the weight to give each feature (-1, 1)

Historical Training Data (Known LTV

Row	fullVisitorId	distinct_days_visited	ltv_pageviews	ltv_visits	ltv_avg_time_on_site_s	ltv_revenue	ltv_transactions	avg_session_quality	first_visit	last_visit	ltv_days	label
1	7587138749751940102	9	[illegible]	9	312.33	24380000	1	[illegible]	[illegible]	[illegible]	345	High Value Customer
2	6007196403211981721	8	[illegible]	11	772.5	null	null	[illegible]	[illegible]	[illegible]	345	
3	9557989866096732580	3	[illegible]	3	356.5	null	null	[illegible]	[illegible]	[illegible]	344	
4	0720311197761340948	114	148	146	2118.0	null	null	1.0	2016-08-05	2017-07-15	344	
5	2742641486650042668	17	113	20	266.28	387000000	2	23.0	2016-08-02	2017-07-11	343	High Value Customer
6	0824839726118485274	127	3153	282	1520.0	null	null	26.0	2016-08-01	2017-07-10	343	
7	1957458976293878100	148	4303	284	796.46	77113430000	22	1.5	2016-08-04	2017-07-12	342	High Value Customer
8	9801276214964695322	79	462	106	219.44	null	null	1.5	2016-08-01	2017-07-07	340	
9	1950585318332186454	6	19	7	51.4	null	null	1.5	2016-08-05	2017-07-11	340	
10	0084834161383601528	7	97	7	258.0	69260000	2	2.0	2016-08-04	2017-07-10	340	High Value Customer
11	928398408398925152	40	553	43	285.37	462190000	2	2.0	2016-08-02	2017-07-07	339	High Value Customer
12	351277725820061611	20	60	20	221.33	null	null	1.0	2016-08-05	2017-07-10	339	
13	4143624098732715494	6	13	7	52.5	null	null	1.0	2016-08-03	2017-07-08	339	
14	1927175312147751345	13	180	14	427.21	44970000	1	2.0	2016-08-03	2017-07-08	339	High Value Customer
15	1315772786660606104	28	272	36	340.3	279320000	3	21.25	2016-08-09	2017-07-14	339	High Value Customer

0.5 0.4 0.1 0.1 0.1

Future Data (Unknown LTV)

17	7904807859681747547	3	42	3	1162.0	null	null	1.0	2016-08-05	2017-07-09	338	????????????????????
18	4405445121320750966	51	358	62	517.36	null	null	1.0	2016-08-08	2017-07-12	338	????????????????????
19	1419607020881916790	5	22	5	711.0	null	null	1.0	2016-08-12	2017-07-15	337	????????????????????
20	3862335714593915688	13	92	16	154.23	238000000	1	2.0	2016-08-09	2017-07-12	337	????????????????????

After the model is trained, you can see the relative importance of each field

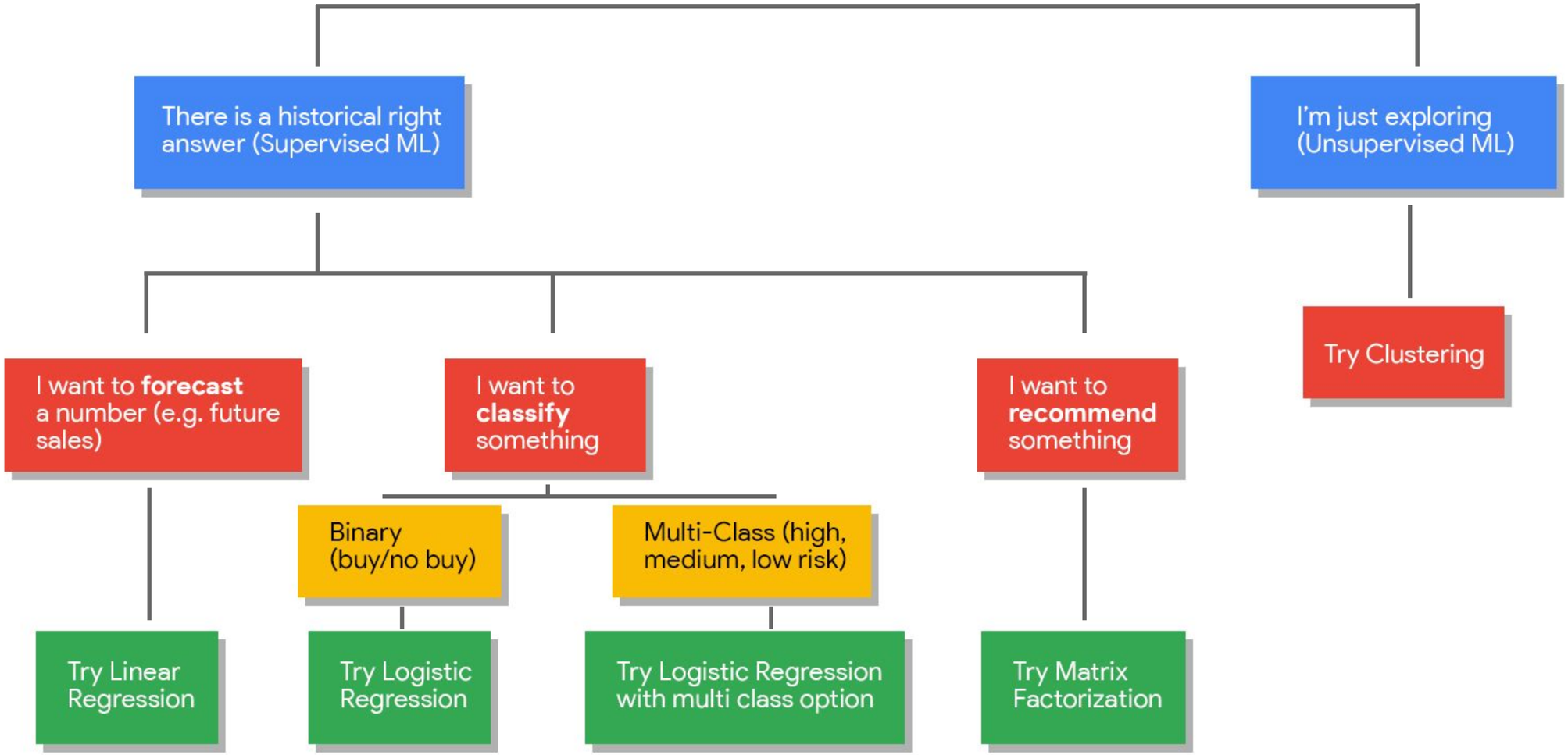
ML terminology review

- **Label** = the correct answer typically from historical data (can be number, string, etc.)
- **Feature** = other columns of data for the model to learn from
- **Model** = a computer-determined recipe to get from features to label
- **Model Types** = *(we will cover soon)*
- **Training** = showing the model lots of examples for it to learn the relationship
- **Weight** = Adjustable parameter of a model.
- **Evaluation** = how the model performs on a set of known labels it has not seen before in training
- **Prediction** = using a trained model to predict on unknown labels

Okay.. I've got data

What model should I use?

Choose the right model type for your structured data use case



Quiz: What model should you use if...

I want to predict the sales numbers for next month

1. Forecasting (linear regression etc..)
2. Classification (logistic regression etc..)
3. Recommendation (matrix factorialization etc..)
4. Unsupervised Learning (clustering etc..)
5. All of the above

Quiz: What model should you use if...

I want to predict the sales numbers for next month

1. **Forecasting (linear regression etc..)**
2. Classification (logistic regression etc..)
3. Recommendation (matrix factorialization etc..)
4. Unsupervised Learning (clustering etc..)
5. All of the above

Quiz: What model should you use if...

I want to predict if a user will buy or not buy

1. Forecasting (linear regression etc..)
2. Classification (logistic regression etc..)
3. Recommendation (matrix factorialization etc..)
4. Unsupervised Learning (clustering etc..)
5. All of the above

Quiz: What model should you use if...

I want to predict if a user will buy or not buy

1. Forecasting (linear regression etc..)
2. **Classification (logistic regression etc..)**
3. Recommendation (matrix factorialization etc..)
4. Unsupervised Learning (clustering etc..)
5. All of the above

It can take days to months to create an ML model

Demo

Creating a Machine Learning Model with SQL

Predicting Customer Return Purchases

BQML is a way to easily build machine learning models

Perception APIs
Google-trained models for your app

BQML (beta)
Easy to use, Familiar SQL

Training & Prediction
End to End Platform for your data & models

App Developers

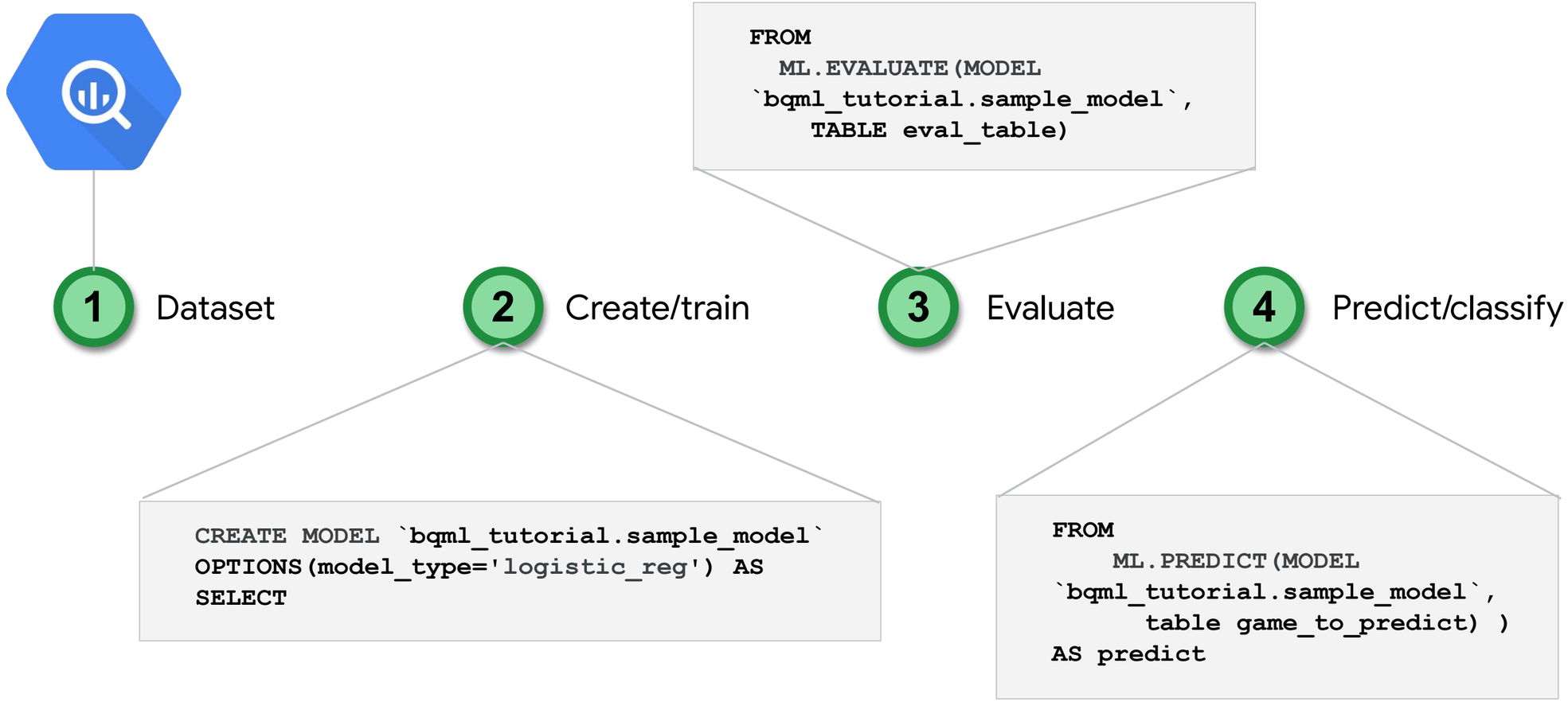
Data Analysts

Data Engineers / Data Scientists

Google Cloud

Cloud ML Engine

Working with BigQuery ML



Behind the scenes

With 2 lines of code:

- Leverages BigQuery's processing power to build a model
- Auto-tunes learning rate
- Auto-splits data into training and test

For the advanced user:

- L1/L2 regularization
- 3 strategies for training/test split: Random, Sequential, Custom
- Set learning rate

BigQuery ML

- Write Machine Learning models with SQL
- Experiment and iterate right where your data lives -- in BigQuery
- Build classification (binary and multi-class) and forecasting models
- Know ML? Inspect model weights and adjust hyperparameters too

Agenda

Introduction to BigQuery

- Fast SQL Query Engine
- Managed Storage for Datasets

Insights from Geographic data

Machine Learning on Structured Data

- Choosing the right model type
- Scenario: Predicting Customer Lifetime Value

Creating ML models with SQL

- Introduction to BigQuery ML
- ML project phases
- Key features walkthrough

BigQuery ML Cheatsheet

- **Label** = alias a column as ‘label’ or specify column in OPTIONS using input_label_cols
- **Feature** = passed through to the model as part of your SQL SELECT statement
  `SELECT * FROM ML.FEATURE_INFO(MODEL `mydataset.mymodel`)`
- **Model** = an object created in BigQuery that resides in your BigQuery dataset
- **Model Types** = Linear Regression, Logistic Regression (more coming)

```
CREATE OR REPLACE MODEL <dataset>.<name>
OPTIONS(model_type='<type>') AS
<training dataset>
```

- **Training Progress** = `SELECT * FROM ML.TRAINING_INFO(MODEL `mydataset.mymodel`)`
- **Inspect Weights** = `SELECT * FROM ML.WEIGHTS(MODEL `mydataset.mymodel`, (<query>))`
- **Evaluation** = `SELECT * FROM ML.EVALUATE(MODEL `mydataset.mymodel`)`
- **Prediction** = `SELECT * FROM ML.PREDICT(MODEL `mydataset.mymodel`, (<query>))`

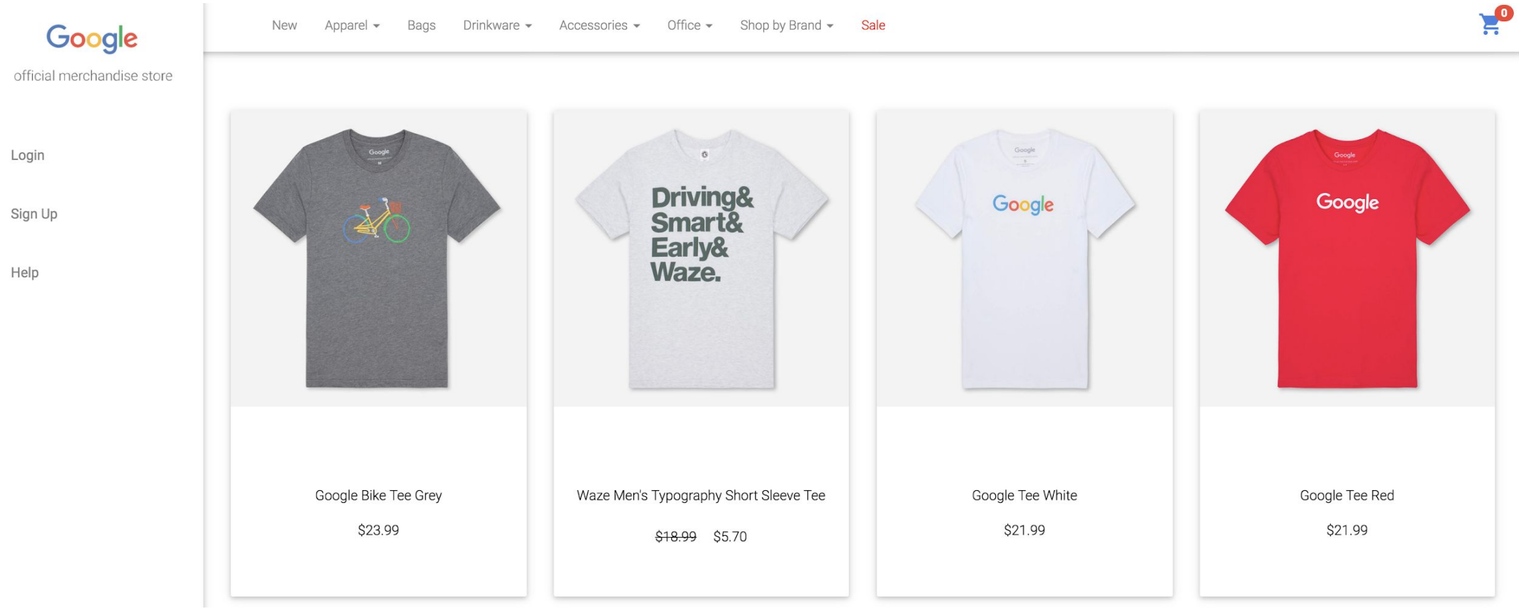
Choosing the right model options

```
CREATE OR REPLACE MODEL
  `mydataset.mymodel`
OPTIONS
  ( model_type='linear_reg',
  input_label_cols= 'sales'
  ls_init_learn_rate=.15,
  l1_reg=1,
  max_iterations=5 ) AS
```

Inspect what the model learned with ML.WEIGHTS

```
SELECT
  category,
  weight
FROM
  UNNEST((
    SELECT
      category_weights
    FROM
      ML.WEIGHTS(MODEL
`bracketology.ncaa_model`)
    WHERE
      processed_input = 'seed')) # try other
features like 'school_ncaa'
      ORDER BY weight DESC
```

Lab: Classify returning customers with BigQuery ML



Which ecommerce customers are likely to return and purchase?

BigQuery

Exploring the Dataset schema:

Google Analytics BigQuery Export schema

Lab

Predicting Visitor Purchases with BigQuery ML

- Explore the Google Analytics ecommerce dataset
- Create a ML training dataset
- Select a model to train
- Train, evaluate, and predict
- Improve ML model performance