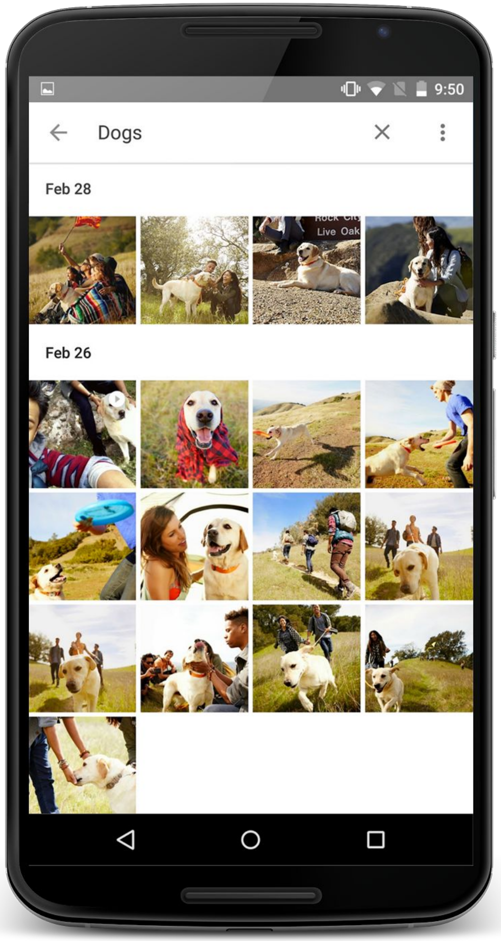
BigQuery ML

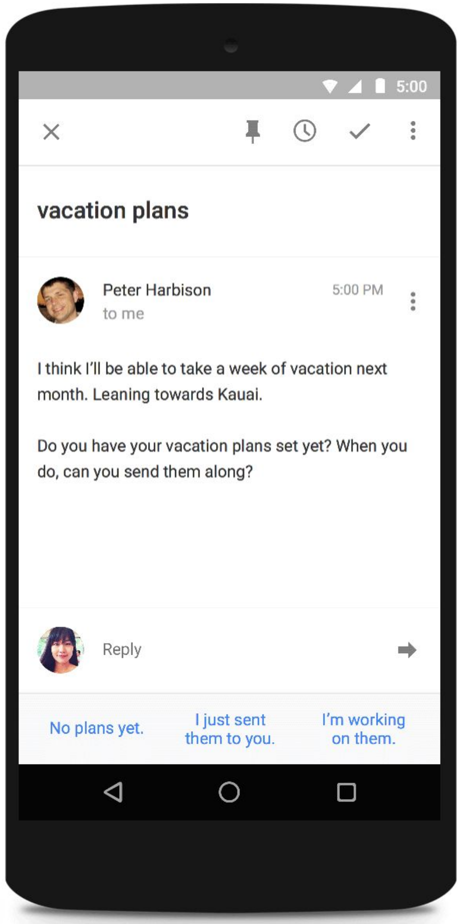
When you hear “AI or ML,” you probably think of:

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

BQML is a way to easily build machine learning models

Perception APIs
Google-trained models for your app

App Developers

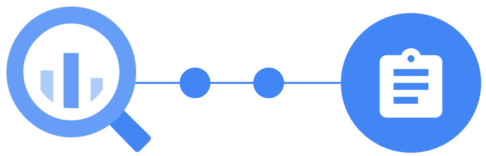
BQML
Easy to use, Familiar SQL

Data Analysts

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

Data Engineers / Data Scientists

Months to create and deploy an ML model



Export data

Tasks

Suboptimal models on small data in notebooks

ETL for model training

Infrastructure management for production models

ETL for batch predictions

Extensive data science training

Security & Compliance

Set up and maintain data access policies for laptops, notebooks, VM clusters

- But I already defined these for my data
- Has anyone managed to do this successfully?

Custom model without BQML

1. Identify a problem & gather data
2. Import & preprocess data
3. Split data
4. Build model
5. Train & evaluate
6. Deploy model
7. Make predictions

Custom model with BQML

1. Identify a problem & gather data
- Import & preprocess data
- Split data
- Build model
2. Train & evaluate
- Deploy model
3. Make predictions

Google Cloud

BigQuery ML

Execute ML models without moving data from BigQuery

Iterate on models in SQL in BigQuery to increase development speed

Automate common ML tasks and hyperparameter tuning

Working with BigQuery ML

1 Dataset

2 Create/train

```
CREATE MODEL `bqml_tutorial.sample_model`
OPTIONS(model_type='logistic_reg') AS
SELECT
```

3 Evaluate

```
FROM
  ML.EVALUATE(MODEL
`bqml_tutorial.sample_model`,
    TABLE eval_table)
```

4 Predict/classify

```
FROM
    ML.PREDICT(MODEL
`bqml_tutorial.sample_model`,
      table game_to_predict) )
AS predict
```

Supported BigQuery ML models

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

Table info

Table ID	nyc-tlc:yellow.trips
Table size	129.72 GB
Long-term storage size	129.72 GB
Number of rows	1,108,779,463

pickup_datetime	dropoff_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	rate_code	passenger_count
2010-03-04 00:35:16 UTC	2010-03-04 00:35:47 UTC	-74.035201	40.721548	-74.035201	40.721548	1	1
2010-03-15 17:18:34 UTC	2010-03-15 17:18:35 UTC	0.0	0.0	0.0	0.0	1	1
2015-03-18 01:07:02 UTC	2015-03-18 01:07:07 UTC	0.0	0.0	0.0	0.0	1	5
2015-03-09 18:24:03 UTC	2015-03-09 18:25:37 UTC	-73.93724822998047	40.758201599121094	-73.93726348876953	40.7581901550293	1	1
2010-03-06 06:33:41 UTC	2010-03-06 06:36:06 UTC	-73.785514	40.6454	-73.784564	40.648681	1	2
2013-08-07 00:42:45 UTC	2013-08-07 00:58:43 UTC	-74.025817	40.763044	-74.046752	40.78324	5	1
2015-04-26 02:56:37 UTC	2015-04-26 03:00:01 UTC	-73.98765563964844	40.77165603637695	-73.98755645751953	40.771751403808594	1	1
2015-04-29 18:45:03 UTC	2015-04-29 18:49:01 UTC	0.0	0.0	0.0	0.0	1	1
2010-03-11 21:24:48 UTC	2010-03-11 21:46:51 UTC	-74.571511	40.9108	-74.628928	40.964321	1	1
2013-08-24 01:58:23 UTC	2013-08-24 01:58:23 UTC	-73.972171	40.759439	0.0	0.0	5	4

Google Cloud

Taxi photo:
@alexanderredl
https://unsplash.com/photos/S9qxkJN0f4Q

Select data

Photo from Unsplash

```
SELECT
  fare_amount,
  pickup_longitude,
  pickup_latitude,
  dropoff_longitude,
  dropoff_latitude,
  passenger_count

FROM
  `nyc-tlc.yellow.trips`
```

Google Cloud

Build and train with CREATE MODEL

Photo from Unsplash

Google Cloud

```
CREATE OR REPLACE MODEL
  mydataset.model_linreg

OPTIONS(
  input_label_cols=['fare_amount'],
  model_type='linear_reg') AS

SELECT
  fare_amount,
  pickup_longitude,
  pickup_latitude,
  dropoff_longitude,
  dropoff_latitude,
  passenger_count

FROM
  `nyc-tlc.yellow.trips`
```

Evaluate with **ML.EVALUATE**

Photo from Unsplash

Google Cloud

```
SELECT
  *

FROM
  ML.EVALUATE(
    MODEL mydataset.model_linreg
    )
```

Use the model with ML.PREDICT

Photo from Unsplash

Google Cloud

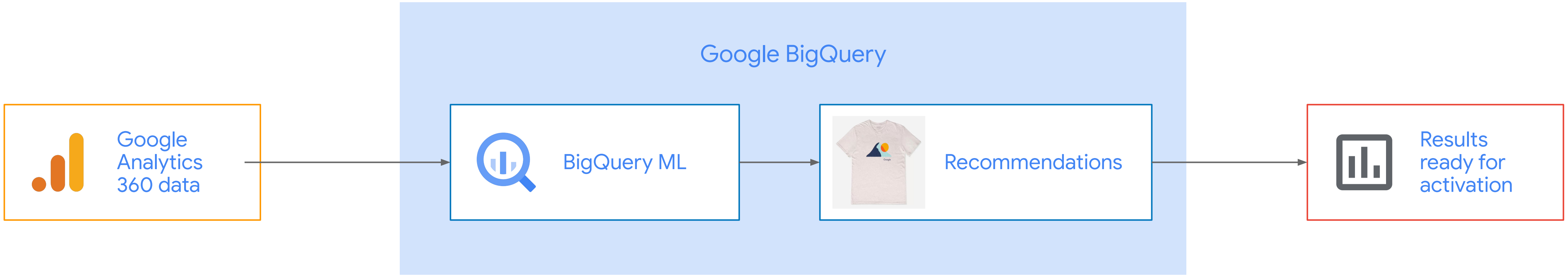
```
SELECT
  *

FROM
  ML.PREDICT(MODEL mydataset.model_linreg,
    (
    SELECT
      fare_amount,
      pickup_longitude,
      pickup_latitude,
      dropoff_longitude,
      dropoff_latitude,
      passenger_count
    FROM
      `nyc-tlc.yellow.trips`
  ))
```

How to build and deploy a recommendation system with BigQuery ML

Google Cloud

How to build a recommendation system with BigQuery ML



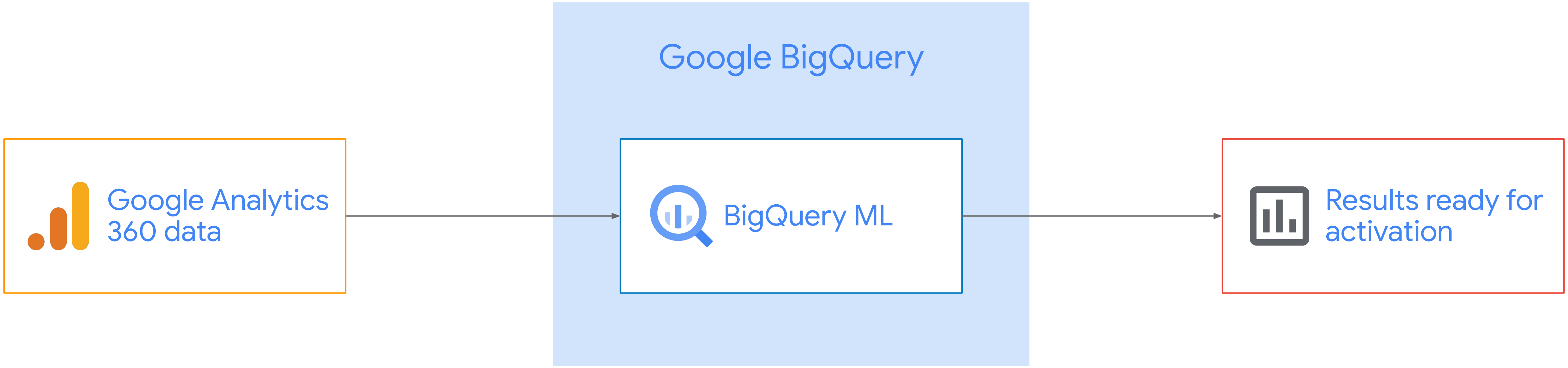
Show relevant products at the right moment, right context.

Businesses need to build recommendation systems

To improve conversions and clickthrough-rates.

And build customer loyalty.

How do you build a recommendation system with BigQuery ML?



Common use case:

- Prepare your training data in BigQuery
- Train a recommendation system with BigQuery ML
- Use the predicted recommendations in production

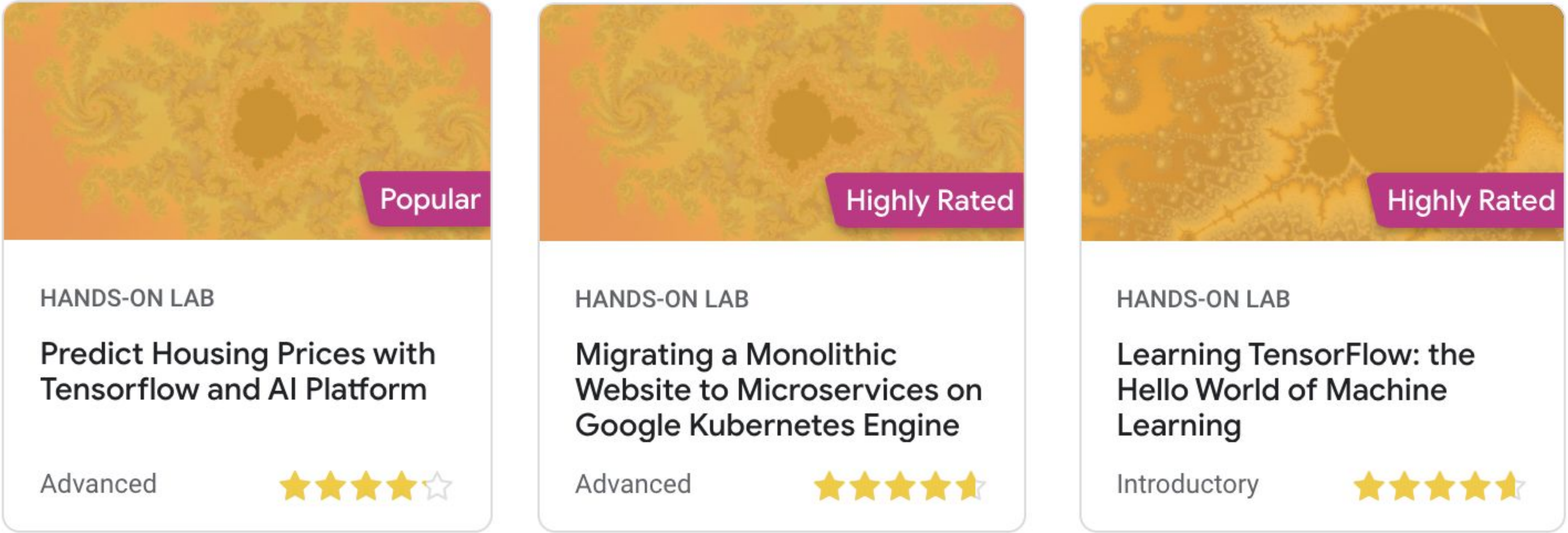
Preparing the training data

Google Cloud

How do you know if a user likes a product ?
item
service
news article
movie

Google Cloud

Explicit feedback



Implicit feedback

For example, amount of time spent viewing a product

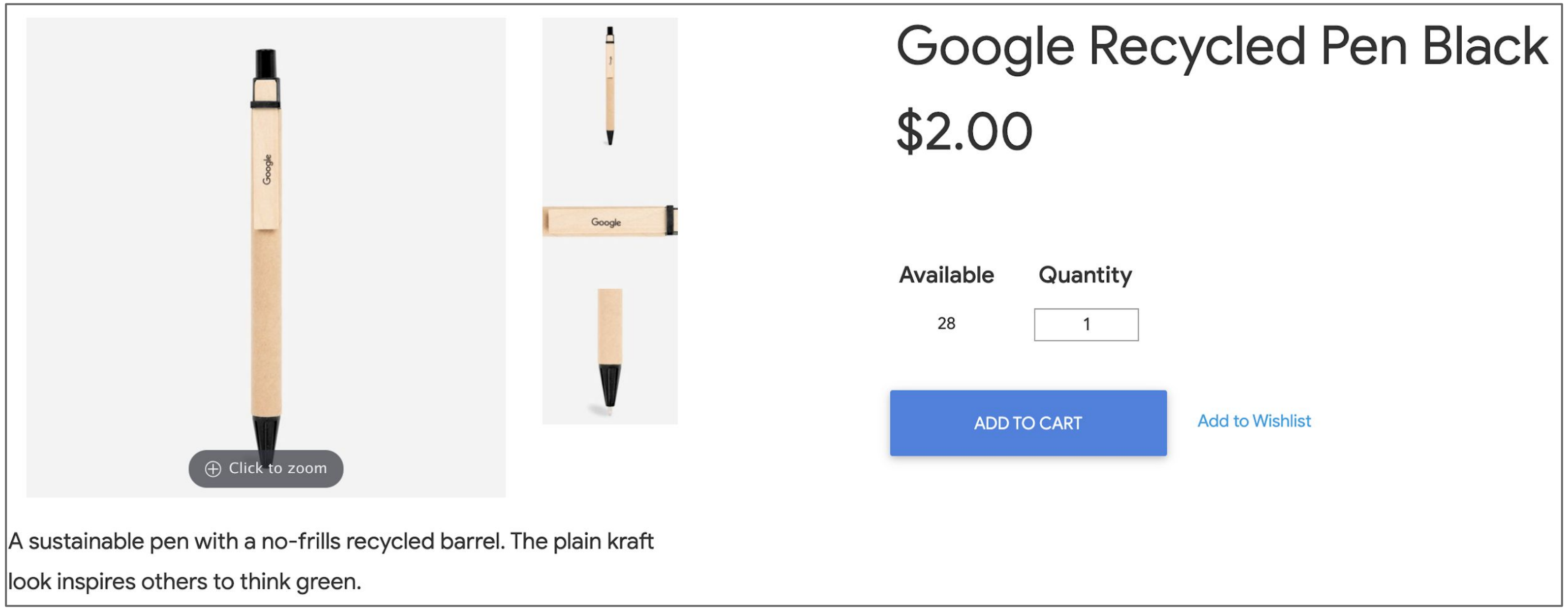


Image: googlemerchandisestore.com

Google

official merchandise store

New | Apparel | Lifestyle | Stationery | Eco-Friendly | Shop by Brand | Sale | Campus Collection

Brand of Item | Sort By

Hats

YouTube Twill Sandwich Cap Black

$13.00

Google Leather Strap Hat Black

$17.00

Google Leather Strap Hat Blue

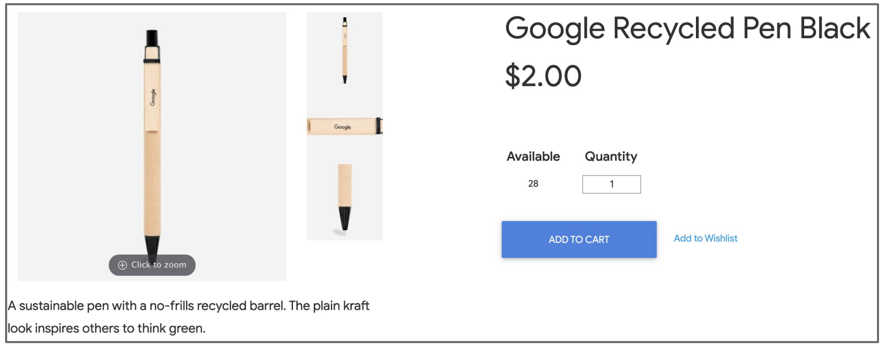
$17.00

YouTube Leather Strap Hat Black

~~$17.00~~ $11.90

Sample Google Analytics 360 data

from the Google Merchandise Store



Available on
BigQuery Public Datasets

Pre-processed training data

	visitorId	itemId	session_duration
0	5468761641774363851-1	GGOEGAAX0031	59
1	7186762836575002506-1	GGOEGAAX0031	5787
2	1827763305655925232-3	GGOEGAAX0031	6423
3	3863873694540855771-4	GGOEGAAX0031	6168
4	2046797444872582027-2	GGOEGAAX0031	6891
5	5885261202339404877-2	GGOEGAAX0031	85256
6	0097216552247524030-5	GGOEGAAX0031	39980
7	860497822069663220-1	GGOEGAAX0031	2486
8	8960897138106393989-1	GGOEGAAX0031	11641
9	9681217015581152227-1	GGOEGAAX0031	6398

https://console.cloud.google.com/marketplace/details/obfuscated-ga360-data/obfuscated-ga360-data?filter=solution-type:dataset

Google Cloud

Training a recommendation system using Matrix Factorization with BigQuery ML

Google Cloud

	visitorId	itemId	session_duration
0	5468761641774363851-1	GGOEGAAX0031	59
1	7186762836575002506-1	GGOEGAAX0031	5787
2	1827763305655925232-3	GGOEGAAX0031	6423
3	3863873694540855771-4	GGOEGAAX0031	6168
4	2046797444872582027-2	GGOEGAAX0031	6891
5	5885261202339404877-2	GGOEGAAX0031	85256
6	0097216552247524030-5	GGOEGAAX0031	39980
7	860497822069663220-1	GGOEGAAX0031	2486
8	8960897138106393989-1	GGOEGAAX0031	11641
9	9681217015581152227-1	GGOEGAAX0031	6398

Training data

```
SELECT
  *
FROM
  bqml.aggregate_web_stats
```

Build and train with CREATE MODEL

	visitorId	itemId	session_duration
0	5468761641774363851-1	GGOEGAAX0031	59
1	7186762836575002506-1	GGOEGAAX0031	5787
2	1827763305655925232-3	GGOEGAAX0031	6423
3	3863873694540855771-4	GGOEGAAX0031	6168
4	2046797444872582027-2	GGOEGAAX0031	6891
5	5885261202339404877-2	GGOEGAAX0031	85256
6	0097216552247524030-5	GGOEGAAX0031	39980
7	860497822069663220-1	GGOEGAAX0031	2486
8	8960897138106393989-1	GGOEGAAX0031	11641
9	9681217015581152227-1	GGOEGAAX0031	6398

Training data

```
CREATE OR REPLACE MODEL
  bqml.retail_recommender

OPTIONS(
  model_type='matrix_factorization',
  user_col='visitorId',
  item_col='itemId',
  rating_col='session_duration',
  feedback_type='implicit'
  ) AS

SELECT
  *
FROM
  bqml.aggregate_web_stats
```

To setup flat-rate pricing:
https://cloud.google.com/bigquery/docs/reservations-intro

Evaluate the model with **ML.EVALUATE**

```
SELECT
  *

FROM
  ML.EVALUATE(
    MODEL bqml.retail_recommender
  )
```

Row	mean_average_precision	mean_squared_error	normalized_discounted_cumulative_gain	average_rank
1	0.011463546060150412	1.0999974250389957E8	18.68240759482596	0.28084605495250364

Predict for a single user with ML.RECOMMEND

```
DECLARE MY_VISITORID STRING DEFAULT "6499749315992064304-2";

SELECT
  *
FROM
  ML.RECOMMEND(
     MODEL bqml.retail_recommender,
     (SELECT MY_VISITORID as visitorID)
  )
ORDER BY predicted_session_duration_confidence DESC

LIMIT 5
```

Row	predicted_session_duration_confidence	visitorId	itemId
1	35344.75190048335	0824461277962362623-1	GGOEGAAX0074
2	28775.356706353305	0824461277962362623-1	GGOEGBJC076099
3	26283.729992603585	0824461277962362623-1	GGOEAFKQ020499
4	24583.28072228624	0824461277962362623-1	GGOEGAAX0289
5	23462.16560910448	0824461277962362623-1	GGOEGOAB021699

Batch Predictions for all users with ML.RECOMMEND

Row	visitorId	itemId	predicted_session_duration_confidence
1	0000010278554503158-1	GGOEGBJC076099	33266.00563956515
2	0000010278554503158-1	GGOEADHB014799	23798.82355845534 4
3	0000010278554503158-1	GGOEYHPB072210	23317.03648749228
4	0000010278554503158-1	GGOEGEVA022399	23045.588017967602
5	0000010278554503158-1	GGOEGAAX0074	22231.48418727726
6	0000020424342248747-1	GGOEGBJC076099	31222.7514436289
7	0000020424342248747-1	GGOEYHPB072210	29845.991045227882
8	0000020424342248747-1	GGOEGODR017799	28248.607012765326
9	0000020424342248747-1	GGOEGEVA022399	28086.011190871777
10	0000020424342248747-1	GGOEGOBG023599	26348.173682193796

Google Cloud

How to use the predicted recommendations in production

Google Cloud

1. **Export recommendations for ad retargeting campaigns**

clientId	LikelyToBuyProductA
123	0.70
345	0.90

For product campaigns,
create a new column for "likelihood to purchase"
based on the predicted recommendations,
then import into Google Analytics to create new campaigns.

2. Connect with your CRM for personalized emails

"Hi Conchita, you might be interested in {A}, {B}, {C}"

Google Analytics 360 Integration for Marketing Cloud

https://trailhead.salesforce.com/en/content/learn/modules/google-analytics-360-integration-for-marketing-cloud

Google Cloud

How to build and deploy a demand forecasting solution with BigQuery ML

Google Cloud

Transportation

➔ Predict ticket sales

https://unsplash.com/photos/9qQTUYm4ss4

Media/Gaming

➔ Predict # active players/time
➔ Predict content viewership

https://unsplash.com/photos/4P0zdOSstqI

Telecommunications

➔ Predict network traffic

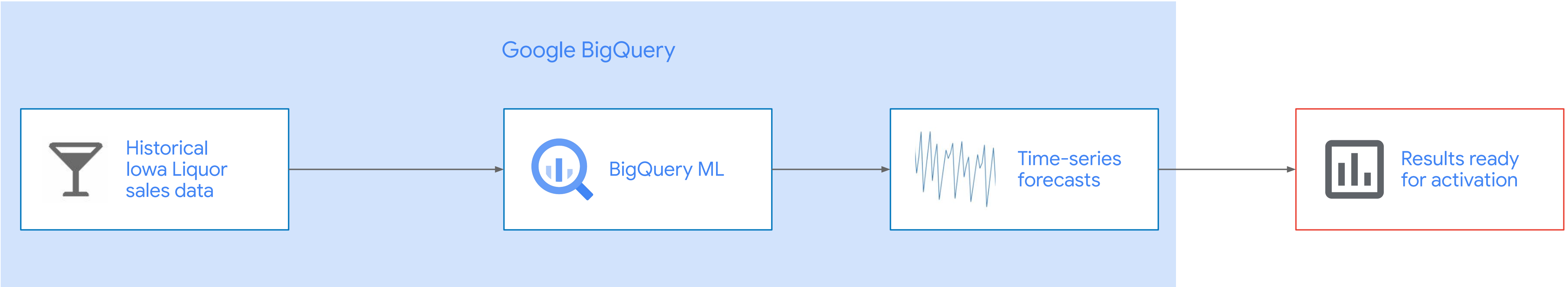
https://unsplash.com/photos/M5tzZtFCOfs

Google Cloud

Retail

https://unsplash.com/photos/nHRXNv2qeDE

Google Cloud



- Prepare the training data in BigQuery
- Train and evaluate a time-series model with BigQuery ML
- Visualize the forecasts in a dashboard
- Schedule and automate model retraining

Google Cloud

Preparing the training data

Iowa Liquor Sales data

Transactional data:

	invoice_and_item_number	date	store_number	item_description	bottles_sold	sale_dollars
0	S26345100049	2015-06-22	2528	99 Bananas Mini	4	35.640
1	S17774100094	2014-03-10	3385	Montezuma Triple Sec	192	612.480
2	S08249300002	2012-10-11	4719	Templeton Rye	30	813.900
3	S06239100038	2012-06-25	2638	Captain Morgan Original Spiced Barrel Tray Pack	150	3937.500
4	INV-20674100062	2019-07-17	2512	Everclear Alcohol	36	486.000

`` `bigquery-public-data.iowa_liquor_sales.sales` ``

Iowa Liquor Sales data, BigQuery Public Datasets
https://console.cloud.google.com/marketplace/details/iowa-department-of-commerce/iowa-liquor-sales

Training data

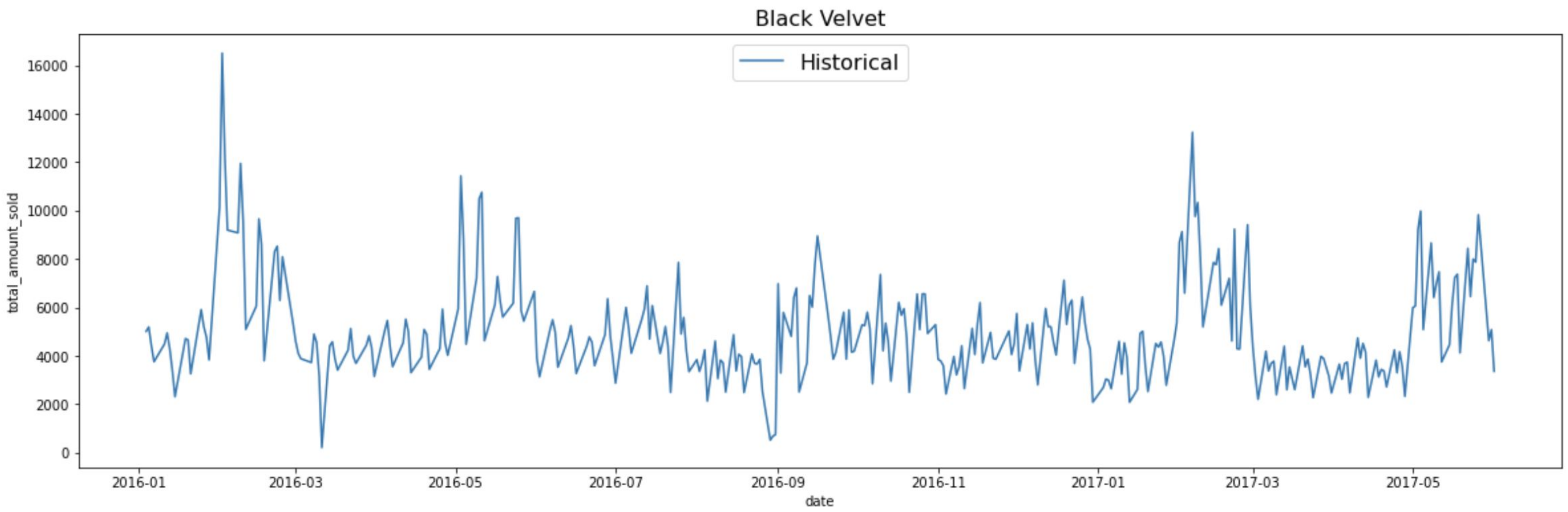
	date	item_name	total_amount_sold
0	2016-01-04	Black Velvet	5014
1	2016-01-05	Black Velvet	5193
2	2016-01-06	Black Velvet	4422
3	2016-01-07	Black Velvet	3760
4	2016-01-11	Black Velvet	4492
5	2016-01-12	Black Velvet	4945
6	2016-01-13	Black Velvet	4302
7	2016-01-14	Black Velvet	3394
8	2016-01-15	Black Velvet	2318
9	2016-01-19	Black Velvet	4714

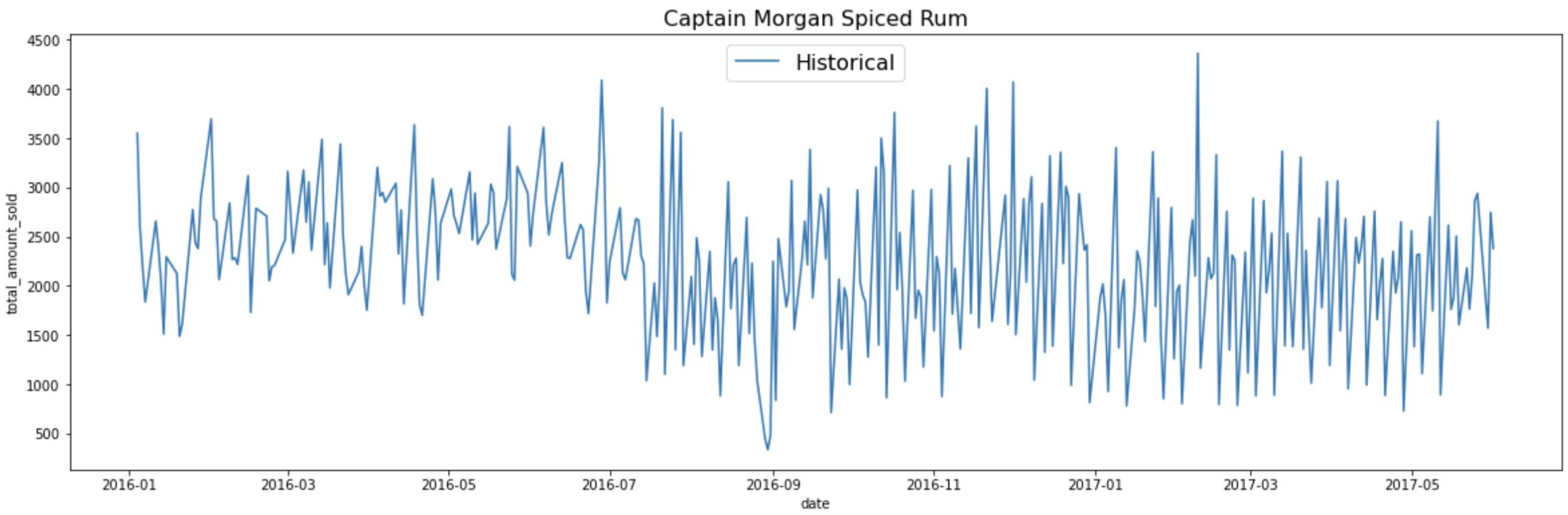
```
SELECT
  date,
  item_description AS item_name,
  SUM(bottles_sold) AS total_amount_sold

FROM
  `bigquery-public-data.iowa_liquor_sales.sales`

GROUP BY
  date, item_name

HAVING
  date BETWEEN DATE('2016-01-01') AND
               DATE('2017-06-01')
```





Google Cloud

Google Cloud

Creating the time-series model in SQL with BigQuery ML

Build and train with CREATE MODEL

	date	item_name	total_amount_sold
0	2016-01-04	Black Velvet	5014
1	2016-01-05	Black Velvet	5193
2	2016-01-06	Black Velvet	4422
3	2016-01-07	Black Velvet	3760
4	2016-01-11	Black Velvet	4492
5	2016-01-12	Black Velvet	4945
6	2016-01-13	Black Velvet	4302
7	2016-01-14	Black Velvet	3394
8	2016-01-15	Black Velvet	2318
9	2016-01-19	Black Velvet	4714

training_data

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

Build and train with CREATE MODEL

	date	item_name	total_amount_sold
0	2016-01-04	Black Velvet	5014
1	2016-01-05	Black Velvet	5193
2	2016-01-06	Black Velvet	4422
3	2016-01-07	Black Velvet	3760
4	2016-01-11	Black Velvet	4492
5	2016-01-12	Black Velvet	4945
6	2016-01-13	Black Velvet	4302
7	2016-01-14	Black Velvet	3394
8	2016-01-15	Black Velvet	2318
9	2016-01-19	Black Velvet	4714

training_data

```
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product




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

Build and train with CREATE MODEL

	date	item_name	total_amount_sold
0	2016-01-04	Black Velvet	5014
1	2016-01-05	Black Velvet	5193
2	2016-01-06	Black Velvet	4422
3	2016-01-07	Black Velvet	3760
4	2016-01-11	Black Velvet	4492
5	2016-01-12	Black Velvet	4945
6	2016-01-13	Black Velvet	4302
7	2016-01-14	Black Velvet	3394
8	2016-01-15	Black Velvet	2318
9	2016-01-19	Black Velvet	4714

training_data

```
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product

OPTIONS(
  MODEL_TYPE='ARIMA',
  TIME_SERIES_TIMESTAMP_COL='date',
  TIME_SERIES_DATA_COL='total_amount_sold',
  TIME_SERIES_ID_COL='item_name'

) AS

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

Build and train with CREATE MODEL

By specifying `TIME_SERIES_ID_COL`, this query will train multiple time-series models, one for every *item_name*

```
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product

OPTIONS(
  MODEL_TYPE='ARIMA',
  TIME_SERIES_TIMESTAMP_COL='date',
  TIME_SERIES_DATA_COL='total_amount_sold',
  TIME_SERIES_ID_COL='item_name'

) AS

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

Build and train with CREATE MODEL

```sql
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product

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
  iowaliquor.training_data
```

https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-time-series#holiday_region

Build and train with CREATE MODEL

Holiday regions:

'GLOBAL'

'NA', 'JAPAC', 'EMEA', 'LAC'

'AE', 'AR', 'AT', 'BE', ...

```
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product

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
  iowaliquor.training_data
```

https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-time-series#holiday_region

Build and train with **CREATE MODEL**

For each item:

BigQuery ML creates up to 42 candidate models using Auto ARIMA, and then the best model gets automatically chosen for you

```
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product

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
  iowaliquor.training_data
```

https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-time-series

Build and train with **CREATE MODEL**

Behind-the-scenes

- Pre-processing
- Holiday effects
- Seasonal and trend decomposition
- Trend modeling with ARIMA and auto-ARIMA

```
CREATE OR REPLACE MODEL
  iowaliquor.forecast_by_product

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
  iowaliquor.training_data
```

https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-time-series

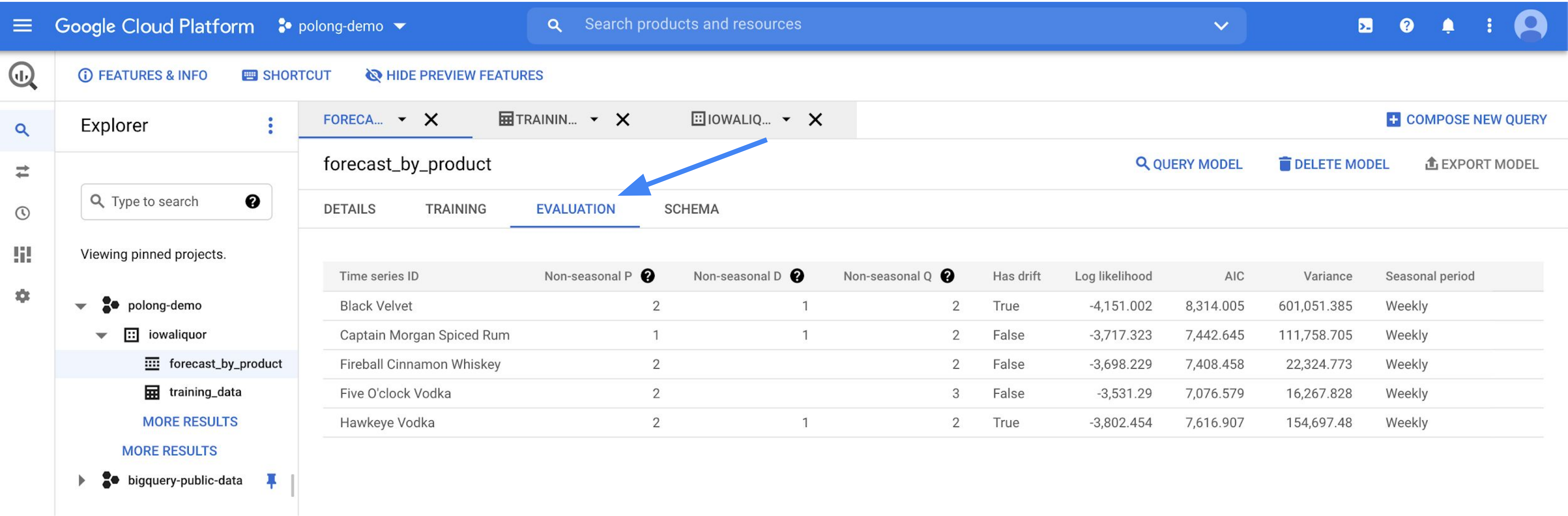
Google Cloud

Evaluating the time-series model with BigQuery ML

Evaluating the model

```
SELECT
  *
FROM
  ML.EVALUATE(MODEL iowaliquor.forecast_by_product)
```

Row	item_name	non_seasonal_p	non_seasonal_d	non_seasonal_q	has_drift	log_likelihood	AIC	variance	seasonal_periods
1	Black Velvet	2	1	2	true	-4151.002490901895	8314.00498180379	601051.3851864329	WEEKLY
2	Captain Morgan Spiced Rum	1	1	2	false	-3717.322548476585	7442.64509695317	111758.70450670805	WEEKLY
3	Fireball Cinnamon Whiskey	2	0	2	false	-3698.2287735141963	7408.457547028393	22324.77258276936	WEEKLY
4	Five O'clock Vodka	2	0	3	false	-3531.2895773450978	7076.5791546901955	16267.828082712951	WEEKLY
5	Hawkeye Vodka	2	1	2	true	-3802.4535522816786	7616.907104563357	154697.47994048957	WEEKLY



Time series ID	Non-seasonal P	Non-seasonal D	Non-seasonal Q	Has drift	Log likelihood	AIC	Variance	Seasonal period
Black Velvet	2	1	2	True	-4,151.002	8,314.005	601,051.385	Weekly
Captain Morgan Spiced Rum	1	1	2	False	-3,717.323	7,442.645	111,758.705	Weekly
Fireball Cinnamon Whiskey	2		2	False	-3,698.229	7,408.458	22,324.773	Weekly
Five O'clock Vodka	2		3	False	-3,531.29	7,076.579	16,267.828	Weekly
Hawkeye Vodka	2	1	2	True	-3,802.454	7,616.907	154,697.48	Weekly

Google Cloud

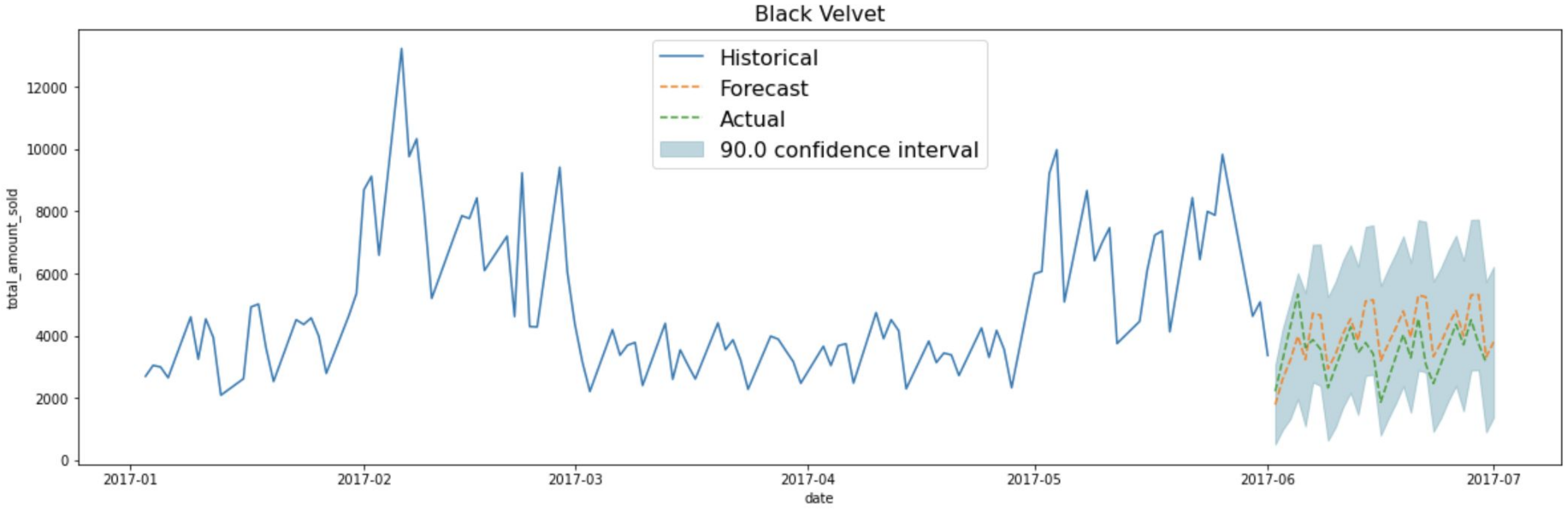
Google Cloud

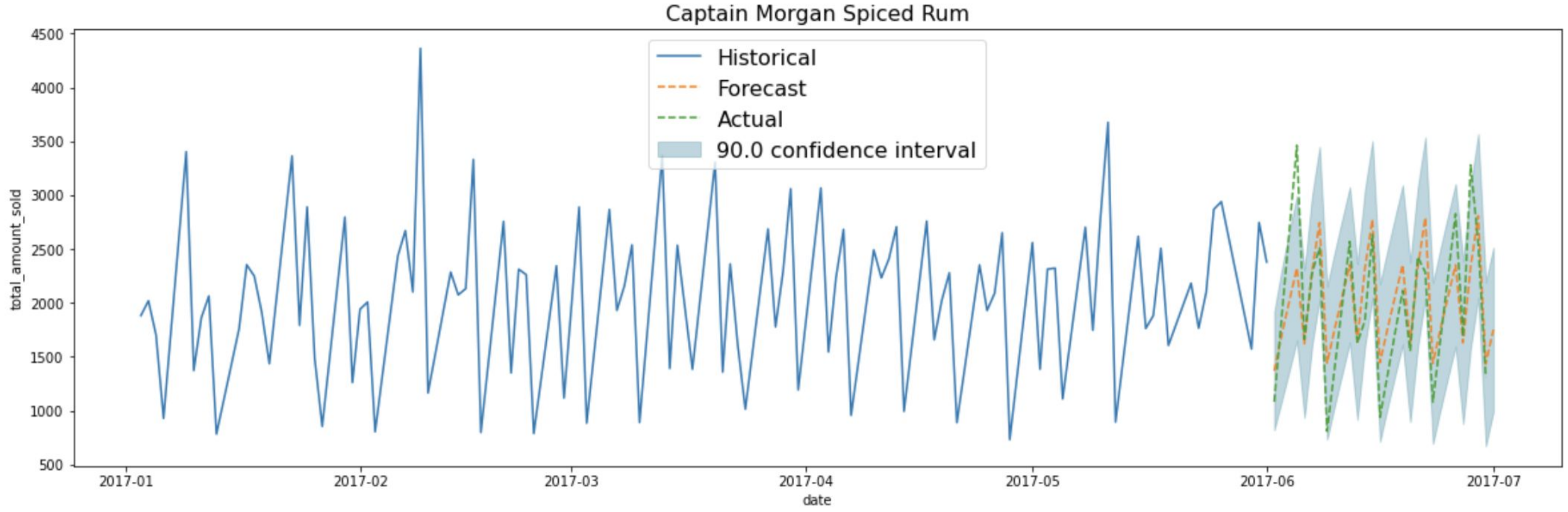
Predicting forecasted values with BigQuery ML

Making forecasts with ML.FORECAST

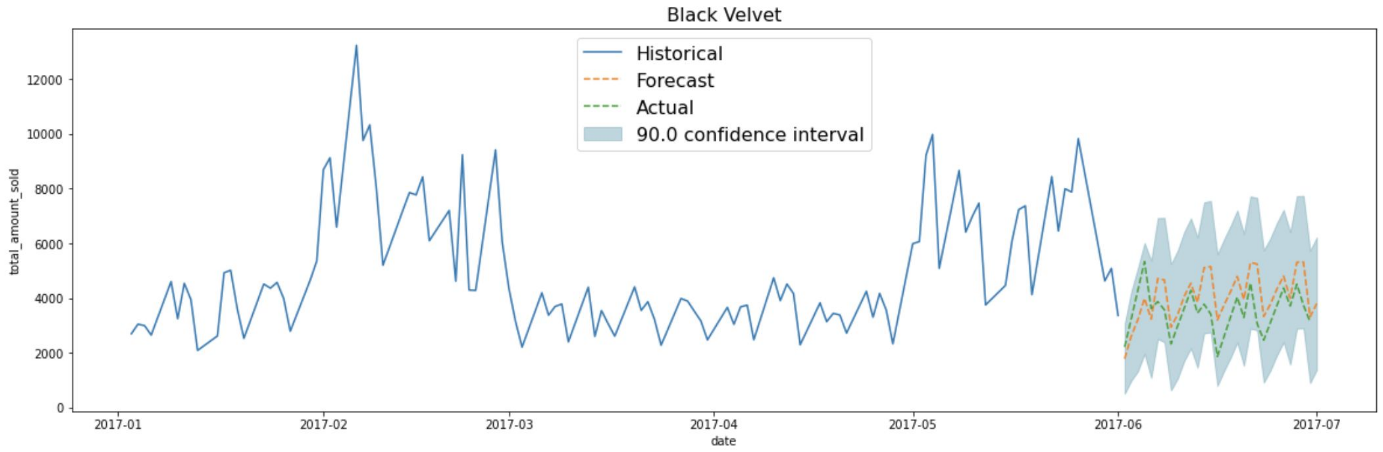
	item_name	forecast_timestamp	forecast_value	standard_error	confidence_level	prediction_interval_lower_bound	prediction_interval_upper_bound
0	Black Velvet	2017-06-02 00:00:00+00:00	1783.563	776.011	0.900	508.522	3058.603
1	Black Velvet	2017-06-03 00:00:00+00:00	2613.904	992.778	0.900	982.700	4245.109
2	Black Velvet	2017-06-04 00:00:00+00:00	3213.561	1146.609	0.900	1329.602	5097.521
3	Black Velvet	2017-06-05 00:00:00+00:00	3989.719	1232.544	0.900	1964.562	6014.876
4	Black Velvet	2017-06-06 00:00:00+00:00	3229.698	1303.835	0.900	1087.405	5371.992

Plotting the forecasts



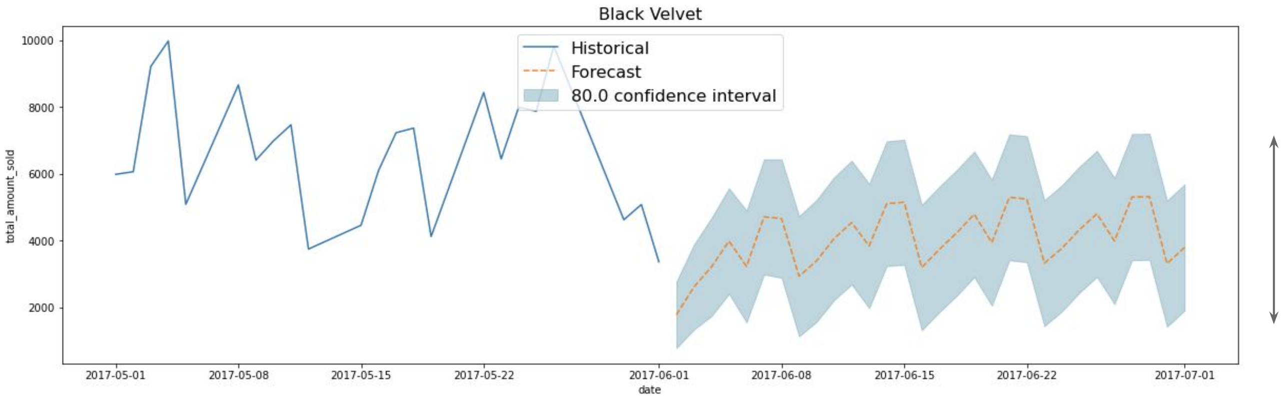


Making forecasts with ML.FORECAST

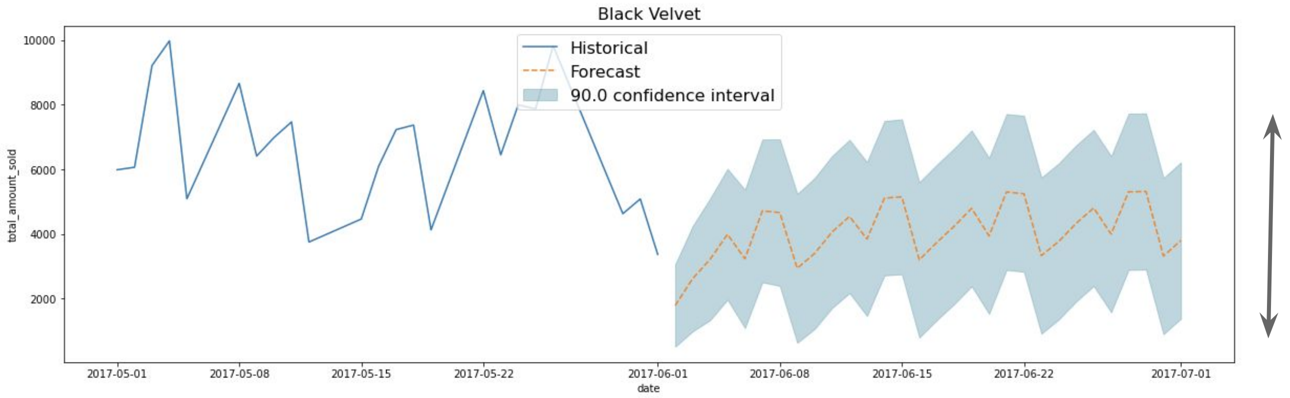



```
SELECT
  *
FROM
  ML.FORECAST(MODEL iowaliquor.forecast_by_product,
              STRUCT(30 AS horizon,
                     0.90 AS confidence_level)
             )
```

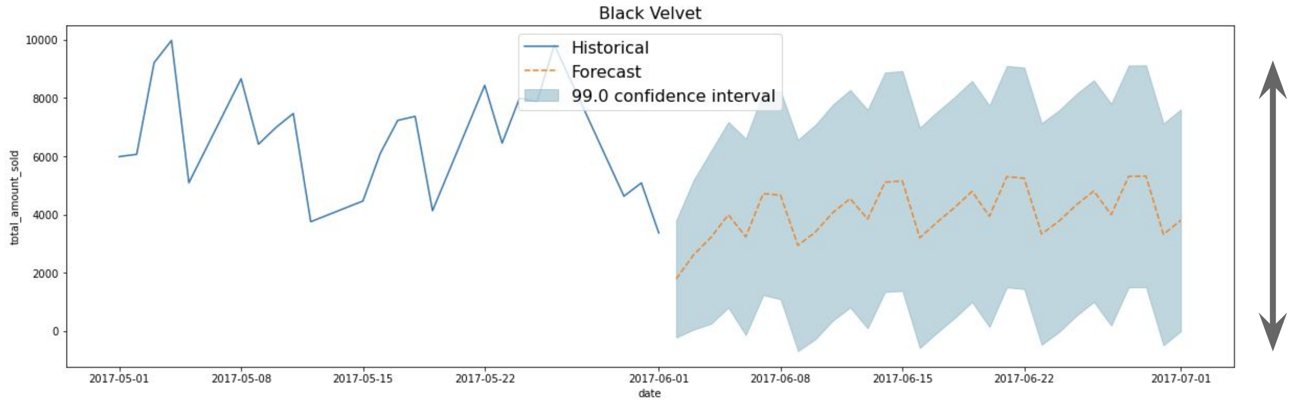
Confidence Levels



0.80 confidence interval



0.90 confidence interval

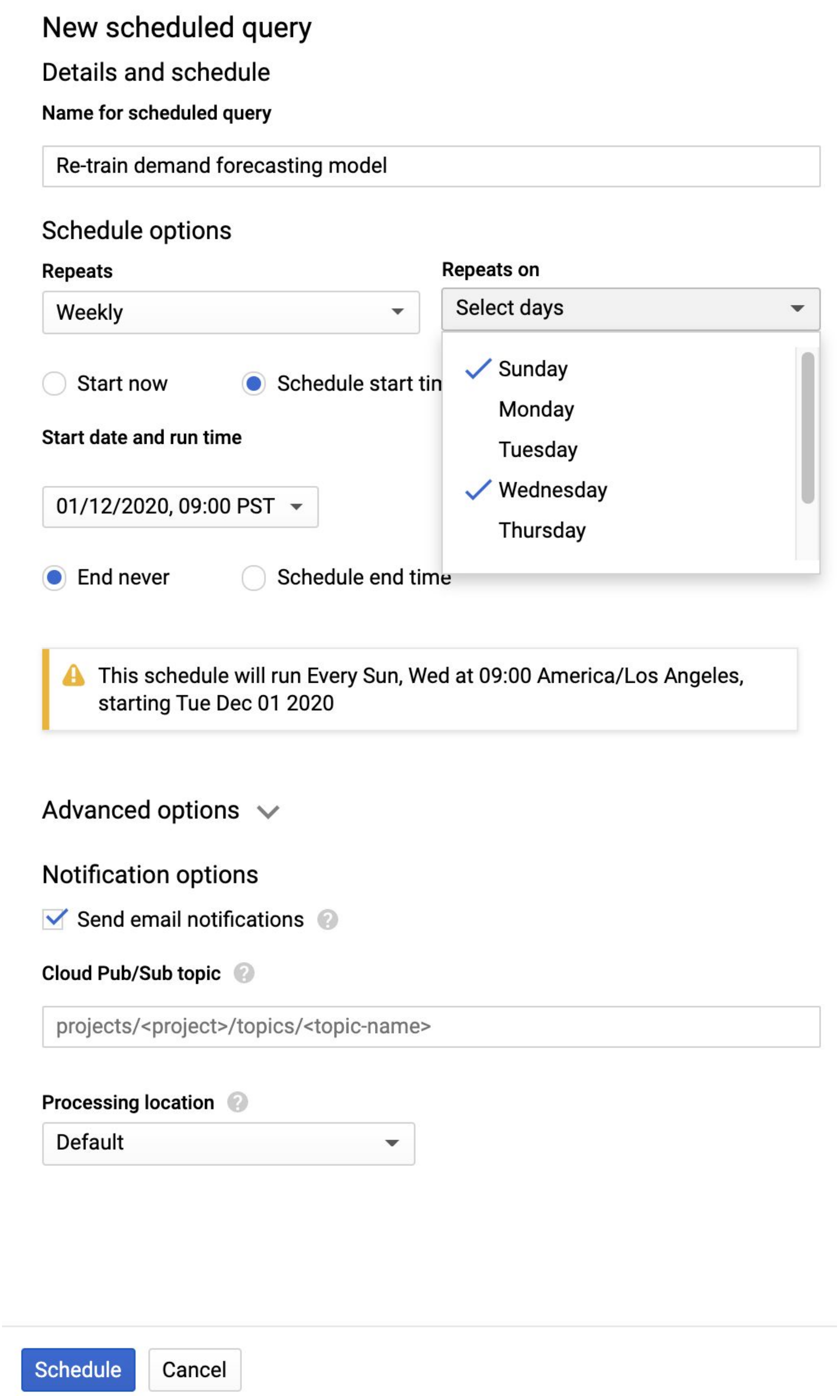


0.99 confidence interval

Scheduling and automating model retraining with scheduled queries

Model retraining

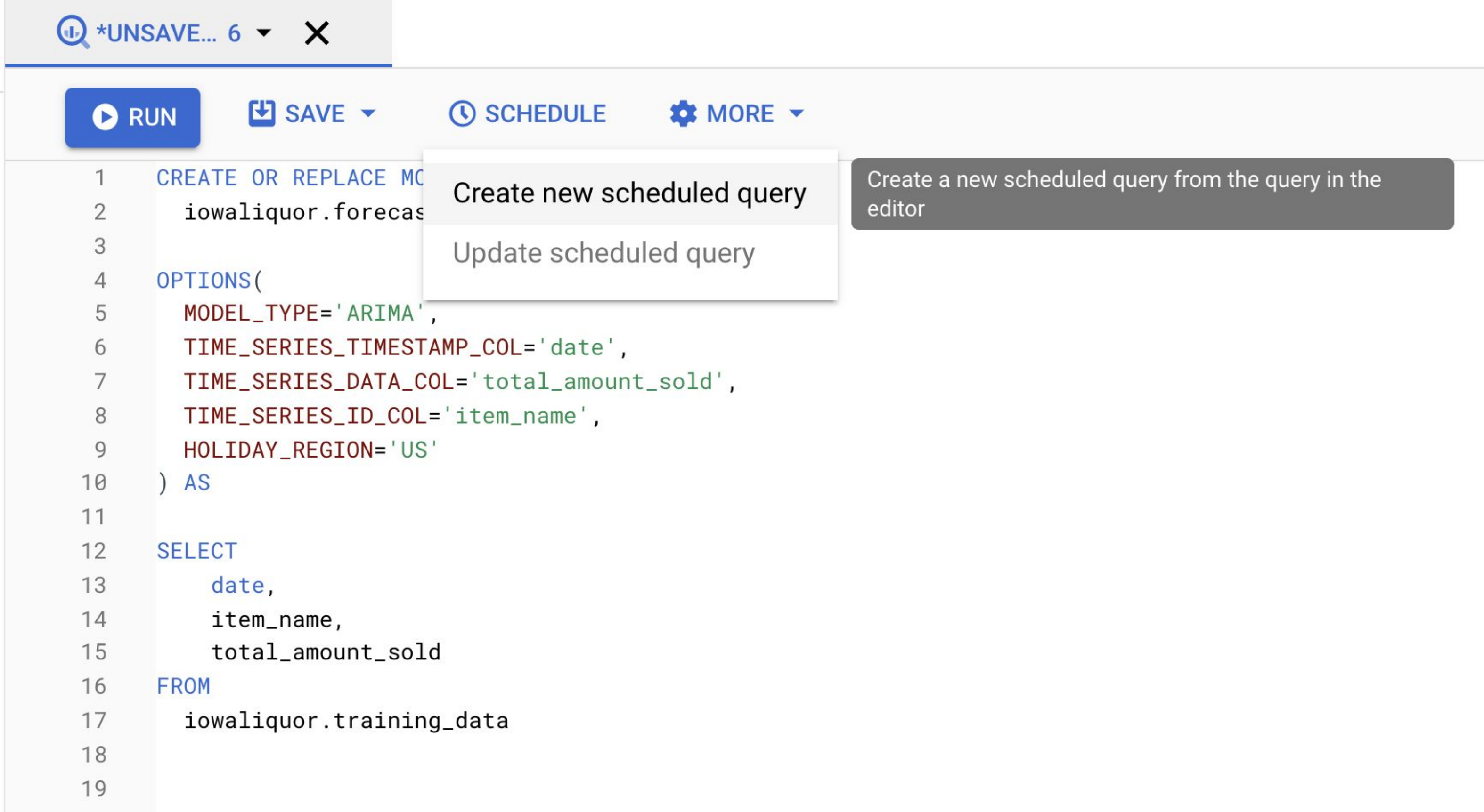
- Use Scheduled Queries in BigQuery



Google Cloud

Model retraining

- Scheduled Queries



Model retraining

- Scheduled Queries

BigQuery ← Scheduled queries + CREATE SCHEDULED QUERY LEARN

Filter transfer configs

	Display name	Source	Schedule (UTC)	Region	Destination data set	Next scheduled ↑	Actions
	Re-train demand forecasting model	Scheduled Query	every sun,wed 17:00	us		2 December 2020 at 17:00:00 UTC	

Google Cloud

Visualizing the forecasts in a dashboard with Data Studio

Preparing the data for visualization

Combine
historical data

with

forecasted values
and intervals

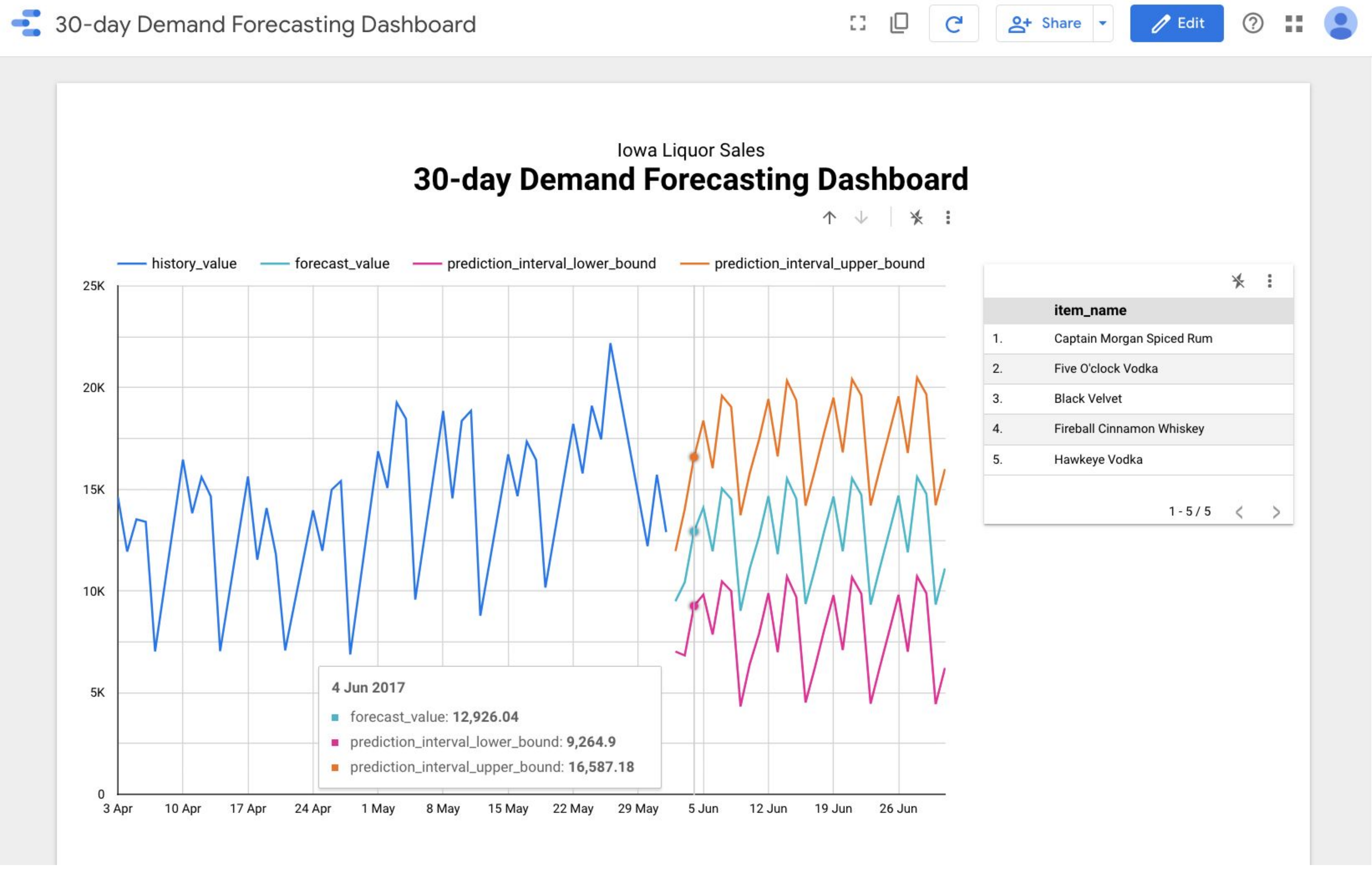
timestamp	item_name	history_value	forecast_value	prediction_interval_lower_bound	prediction_interval_upper_bound
2016-01-04	Black Velvet	5014	*null*	*null*	*null*
2016-01-05	Black Velvet	5193	*null*	*null*	*null*
2016-01-06	Black Velvet	4422	*null*	*null*	*null*
2016-01-07	Black Velvet	3760	*null*	*null*	*null*
2016-01-11	Black Velvet	4492	*null*	*null*	*null*

timestamp	item_name	history_value	forecast_value	prediction_interval_lower_bound	prediction_interval_upper_bound
2017-06-02	Black Velvet	*null*	1783.5628937512147	508.5223953525019	3058.6033921499275
2017-06-03	Black Velvet	*null*	2613.904133961403	982.6996381283686	4245.108629794438
2017-06-04	Black Velvet	*null*	3213.5614560309987	1329.601697639193	5097.521214422804
2017-06-05	Black Velvet	*null*	3989.719033749886	1964.5616454598683	6014.876422039903
2017-06-06	Black Velvet	*null*	3229.6982601347945	1087.4048354125644	5371.991684857025

Preparing the data for visualization

Use **`UNION ALL`**
to combine historical and forecasted values into a single query

```
CREATE OR REPLACE VIEW bqmlforecast.outputdata_datastudio
AS (
  SELECT
    date AS timestamp,
    item_name,
    total_amount_sold AS history_value,
    NULL AS forecast_value,
    NULL AS prediction_interval_lower_bound,
    NULL AS prediction_interval_upper_bound
  FROM
    bqmlforecast.training_data
  UNION ALL
  SELECT
    EXTRACT(DATE
    FROM
      forecast_timestamp) AS timestamp,
    item_name,
    NULL AS history_value,
    forecast_value,
    prediction_interval_lower_bound,
    prediction_interval_upper_bound
  FROM
    ML.FORECAST(MODEL bqmlforecast.arima_model,
      STRUCT(30 AS horizon, 0.9 AS confidence_level))
  ORDER BY timestamp
  )
```


30-day Demand Forecasting Dashboard
Share
Edit
Iowa Liquor Sales
30-day Demand Forecasting Dashboard
history_value
forecast_value
prediction_interval_lower_bound
prediction_interval_upper_bound
25K
20K
15K
10K
5K
0
3 Apr
10 Apr
17 Apr
24 Apr
1 May
8 May
15 May
22 May
29 May
5 Jun
12 Jun
19 Jun
26 Jun
4 Jun 2017
forecast_value: 12,926.04
prediction_interval_lower_bound: 9,264.9
prediction_interval_upper_bound: 16,587.18
item_name
1. Captain Morgan Spiced Rum
2. Five O'clock Vodka
3. Black Velvet
4. Fireball Cinnamon Whiskey
5. Hawkeye Vodka
1 - 5 / 5

ud

How time series models work in BigQuery ML

How do time series models work in BigQuery ML?

Multiple components used in time series model creation pipeline:

1. **Pre-processing**
   a. Missing value imputation
   b. De-duplication of timestamps
   c. Identifying spike anomalies/abrupt changes
2. **Holiday effects**
   a. Spike/dip anomalies during holidays will no longer be treated as anomalies
3. **Seasonal and trend decomposition**
4. Trend modeling with **ARIMA** and **Auto ARIMA**

You can forecast multiple products in the same query

Tackling high-value business problems using AutoML Tables

AutoML Tables

Start with **raw tabular data**

Channels

Geo locations

Promotions

Products

Sales

Month	Channel ID	Product ID	Geo ID	Volume
2018-1	INSTORE	102934	CA102	600
2018-1	INSTORE	112932	CA102	800
2018-1	WEB	901243	AZ203	300
...	...	...	...	...

Demand forecasting

Stock-out prediction

Price optimization

Customer lifetime value

Predict customer conversion / churn

Fraud prevention

and more...

- Build state-of-the-art models automatically
- Enriched treatment for a wide range of data primitives (#s, text, etc.)
- Gracefully handle datasets at BigQuery scale (currently up to 10TB)
- Code-less graphical UI for the full ML lifecycle

IMPORT SCHEMA ANALYZE TRAIN EVALUATE PREDICT

Import your data

AutoML Tables uses tabular data that you import to train a custom machine learning model. Your dataset must contain at least one input feature column and a target column. Optional columns can be added to configure parameters like the data split, weights, etc. Preparing your training data

Table from BigQuery

The table must be in the US regional location

BigQuery project ID *

BigQuery dataset ID *

BigQuery table ID *

CSV from Cloud Storage

The bucket containing the CSV must be in the us-central1 region. CSV formatting

gs:// **BROWSE**

IMPORT

IMPORT SCHEMA ANALYZE TRAIN EVALUATE PREDICT

Select a target

Select a column to be the target (what you want your model to predict) and add optional parameters like weight and time columns

Target column RESET

Deposit

The selected column is categorical data. AutoML Tables will build a classification model, which will predict the target from the classes in the selected column. Learn more

Additional parameters (Optional)

Before continuing, review your dataset schema to make sure each column has the appropriate data type and nullability setting

CONTINUE

Column name	Variable type	Nullability
Age	Numeric	Nullable
Job	Categorical	Nullable
MaritalStatus	Categorical	Nullable
Education	Categorical	Nullable
Default	Categorical	Nullable
Balance	Numeric	Nullable
Housing	Categorical	Nullable
Loan	Categorical	Nullable
Contact	Categorical	Nullable
Day	Categorical	Nullable
Month	Categorical	Nullable
Duration	Numeric	Nullable
Campaign	Categorical	Nullable
PDays	Numeric	Nullable
Previous	Numeric	Nullable
POutcome	Categorical	Nullable
Deposit Target	Categorical	Nullable

IMPORT SCHEMA **ANALYZE** TRAIN EVALUATE PREDICT

⚠ Not up to date. Click the “Continue” button on the Schema tab to regenerate statistics.

All features	17
Numeric	5
Categorical	12

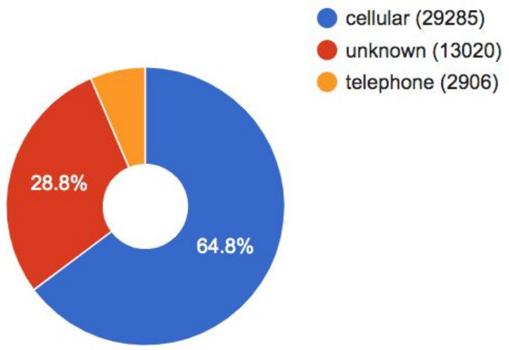
Filter instances

Feature name ↑	Type	Missing	Distinct values	Correlation with Target	Mean
Age	Numeric	0%	77	0.065	40.936
Balance	Numeric	0%	7,168	0.095	1,362.272
Campaign	Categorical	0%	48	0.083	---
Contact	Categorical	0%	3	0.144	---
Day	Categorical	0%	31	0.122	---
Default	Categorical	0%	2	0.028	---
Deposit	Categorical	0%	2	---	---
Duration	Numeric	0%	1,573	0.333	258.163
Education	Categorical	0%	4	0.071	---
Housing	Categorical	0%	2	0.117	---
Job	Categorical	0%	12	0.134	---
Loan	Categorical	0%	2	0.073	---
MaritalStatus	Categorical	0%	3	0.059	---
Month	Categorical	0%	12	0.245	---
PDays	Numeric	0%	559	0.181	40.198
POutcome	Categorical	0%	4	0.313	---
Previous	Numeric	0%	41	0.181	0.58

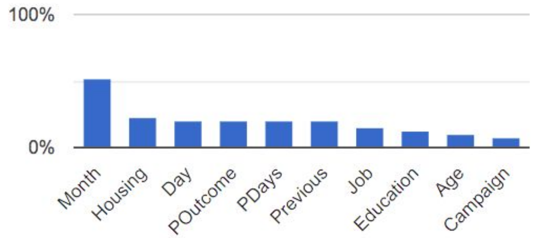
Rows per page: 50 1 – 17 of 17

Details

Distribution



Top correlated features to Contact



Train your model

Model name *

banking_20190410095716

Training budget

Enter a number between 1 and 72 for the maximum number of node hours to spend training your model. If your model stops improving before then, AutoML Tables will stop training and you'll only be charged for the actual node hours used. Training pricing guide

Budget * maximum node hours

Input feature selection

By default, all other columns in your dataset will be used as input features for training (excluding target, weight, and split columns).

16 feature columns *

All columns selected

Summary

Model type: Binary classification model

Data split: Automatic

Target: Deposit

Input features: 16 features

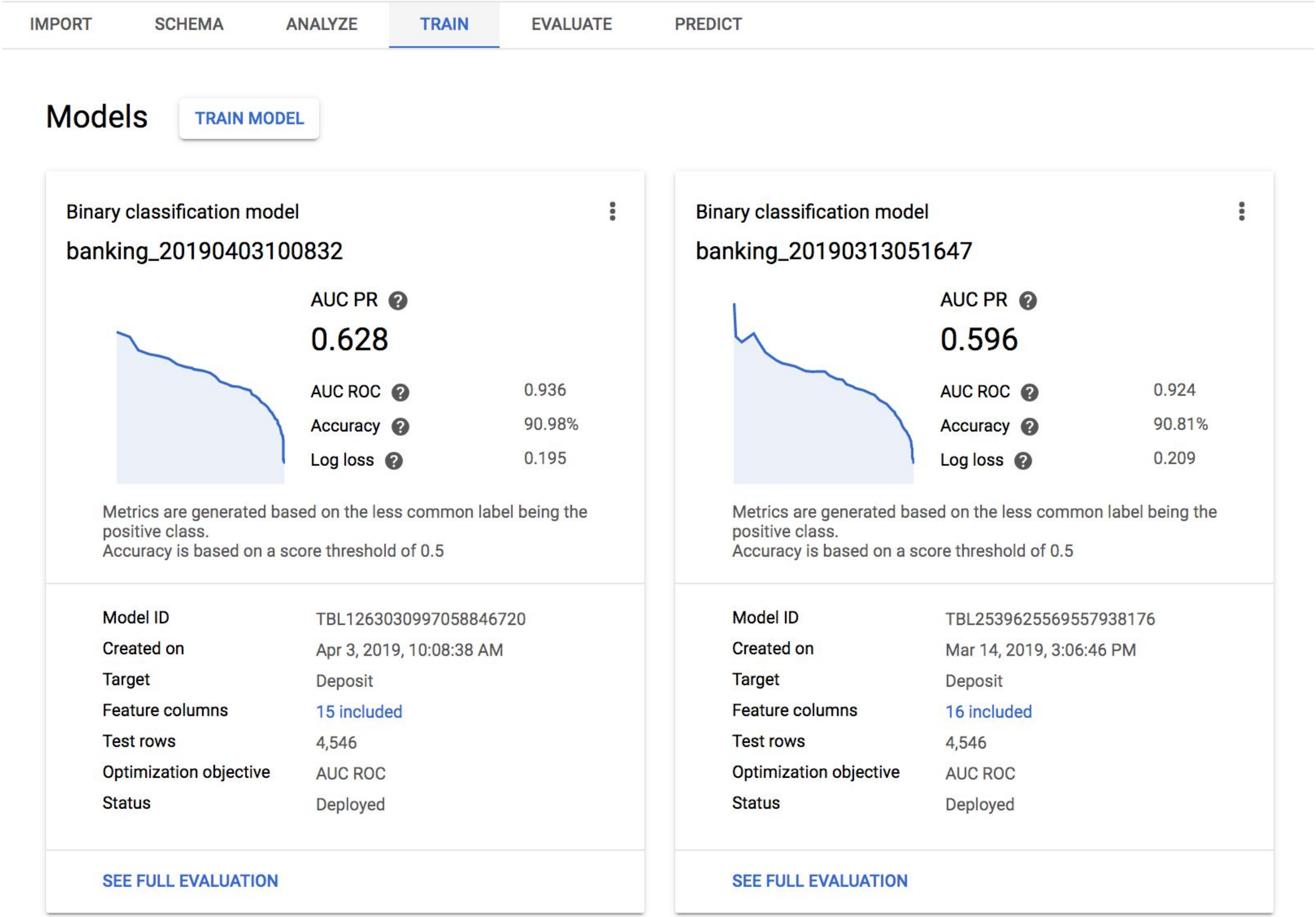
Blue Jeans Meeting

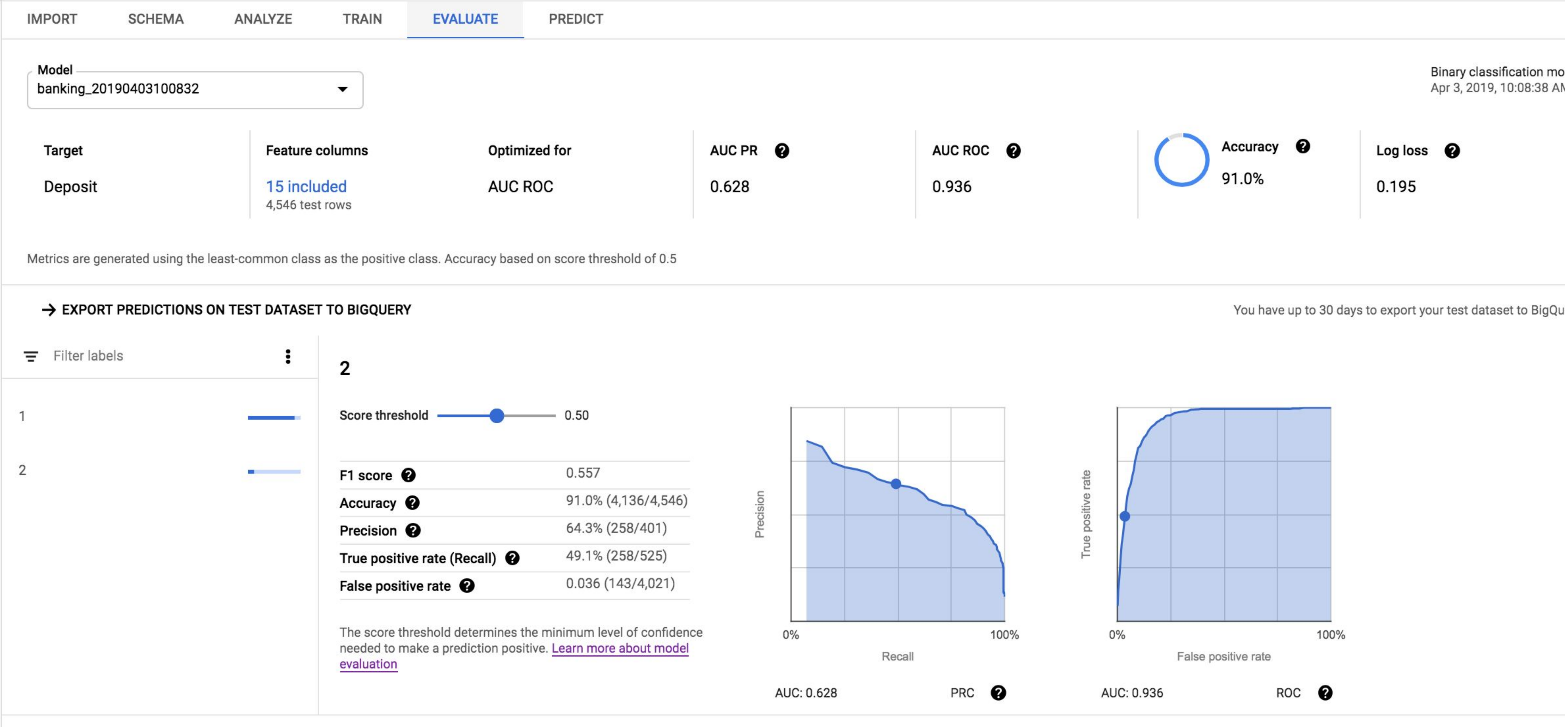
Rows: 45,211 rows

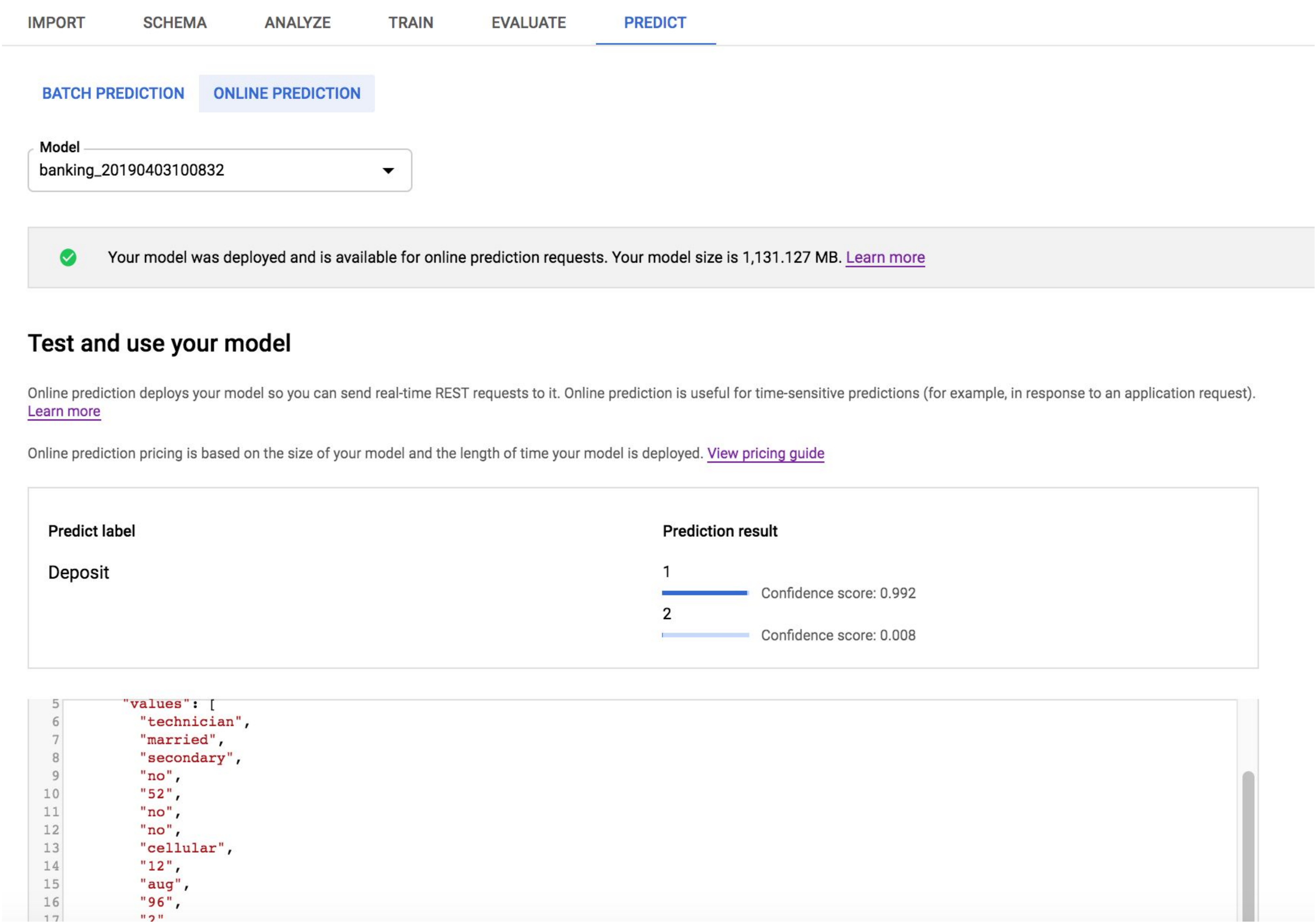
Optimization objective

Depending on the outcome you're trying to achieve, you may want to train your model to optimize for a different objective. Learn more

TRAIN MODEL CANCEL


IMPORT
SCHEMA
ANALYZE
TRAIN
EVALUATE
PREDICT
Models
TRAIN MODEL
Binary classification model
banking_20190403100832
AUC PR
0.628
AUC ROC 0.936
Accuracy 90.98%
Log loss 0.195
Metrics are generated based on the less common label being the positive class.
Accuracy is based on a score threshold of 0.5
Model ID TBL1263030997058846720
Created on Apr 3, 2019, 10:08:38 AM
Target Deposit
Feature columns 15 included
Test rows 4,546
Optimization objective AUC ROC
Status Deployed
SEE FULL EVALUATION
Binary classification model
banking_20190313051647
AUC PR
0.596
AUC ROC 0.924
Accuracy 90.81%
Log loss 0.209
Metrics are generated based on the less common label being the positive class.
Accuracy is based on a score threshold of 0.5
Model ID TBL2539625569557938176
Created on Mar 14, 2019, 3:06:46 PM
Target Deposit
Feature columns 16 included
Test rows 4,546
Optimization objective AUC ROC
Status Deployed
SEE FULL EVALUATION


IMPORT
SCHEMA
ANALYZE
TRAIN
EVALUATE
PREDICT
Model
banking_20190403100832
Binary classification mo
Apr 3, 2019, 10:08:38 AI
Target
Deposit
Feature columns
15 included
4,546 test rows
Optimized for
AUC ROC
AUC PR
0.628
AUC ROC
0.936
Accuracy
91.0%
Log loss
0.195
Metrics are generated using the least-common class as the positive class. Accuracy based on score threshold of 0.5
EXPORT PREDICTIONS ON TEST DATASET TO BIGQUERY
You have up to 30 days to export your test dataset to BigQu
Filter labels
1
2
2
Score threshold
0.50
F1 score
0.557
Accuracy
91.0% (4,136/4,546)
Precision
64.3% (258/401)
True positive rate (Recall)
49.1% (258/525)
False positive rate
0.036 (143/4,021)
The score threshold determines the minimum level of confidence needed to make a prediction positive. Learn more about model evaluation
Precision
0%
100%
Recall
AUC: 0.628
PRC
True positive rate
0%
100%
False positive rate
AUC: 0.936
ROC


IMPORT
SCHEMA
ANALYZE
TRAIN
EVALUATE
PREDICT
BATCH PREDICTION
ONLINE PREDICTION
Model
banking_20190403100832
Your model was deployed and is available for online prediction requests. Your model size is 1,131.127 MB. Learn more
Test and use your model
Online prediction deploys your model so you can send real-time REST requests to it. Online prediction is useful for time-sensitive predictions (for example, in response to an application request). Learn more
Online prediction pricing is based on the size of your model and the length of time your model is deployed. View pricing guide
Predict label
Deposit
Prediction result
1
Confidence score: 0.992
2
Confidence score: 0.008
"values": [
"technician",
"married",
"secondary",
"no",
"52",
"no",
"no",
"cellular",
"12",
"aug",
"96",

AutoML tables through BigQuery ML

```
CREATE MODEL
  `demo.return_buyer_model`
OPTIONS
  (model_type='automl_classifier',
    labels = ['will_buy_later'])
AS SELECT ...
```

```
SELECT
  fullVisitorId,
  predicted_will_buy_later,
  predicted_will_buy_later_probs
FROM ML.PREDICT(MODEL `demo.return_buyer_model`,
    (SELECT ...
```