Vertex AI Forecast

Highest accuracy forecasts with
state-of-the-art machine learning models

Mark Tenenholtz
@marktenenholtz

In NLP, it's really easy to consistently beat a traditional bag-of-words approach with a transformer.

In computer vision, it's really easy to consistently beat an SVM with a CNN.

In time-series forecasting, it's $*&%ing hard to consistently beat a seasonal rolling average.

5:00 AM · 09 Nov 22 · Typefully

Google Cloud

Google Cloud: Machine Learning tools for everyone

Business User
Insights and objectives

- Interactive BI
  Looker
- EDW in a Spreadsheet
  Connected Sheets
- Natural Language Query
  Data QnA

Data analyst
Query and analyze

- Endless EDW
  BigQuery
- Self-managed data pipelines
  Data Fusion, Dataflow
- Data models, catalog
  Looker, Data Catalog
- Machine learning in SQL
  BigQuery ML

Data engineer
Get clean, useful data

- Self-driving infra
  BigQuery, Dataflow, Composer
- Broad choice of tools/language
  Dataproc, Dataflow
- Data quality /lineage
  Vertex AI, BigQuery, Dataflow
- Real-time capabilities
  BigQuery, Dataflow

Data scientist
Models that work

- Portable notebooks
  Notebooks
- Model eval and selection
  Explainable AI, Tensorboard
- Point-and-click dev
  AutoML
- Collaboration
  Feature Store, Pipelines
- Managed models
  Forecast

ML developer
Intelligent apps

- Images, videos
  Vision, Video Intelligence
- Sentiment analysis, entity extraction
  NL, Translation
- Chatbots, voice commands
  Conversation
- Fleet routing, forecasting
  Optimization

ML engineer
Models in production

- Scalable model hosting
  Prediction
- ML CI/CD and orchestration
  Pipelines
- Provenance and lineage
  ML Metadata
- Improvements and retraining
  Model Monitoring

Vertex AI

Applications

- Vision and Video
- Conversation
- Language
- Structured Data

Core

- Workbench
- Data Labeling
- Deep Learning Env
- Experiments
- Metadata
- AutoML
- Training
- Explainable AI
- Feature Store
- Vizier (Optimization)
- Prediction
- Continuous Monitoring
- Pipelines
- AI Accelerators
- Hybrid AI

Structured Data Problem Types

Forecasting

How many products will be sold next month?

Sales	Date	Channel	SKU	...	Geo	Domain	Brand
$1,200	Jun-10-2020	Website	12345	...	US	Shoes	Nike
$1,350	Jun-11-2020	Email	54221	...	US	Shirts	Adidas
✓	**Jan-10-2021**	**Website**	**22345**	**...**	**CA**	**Shoes**	**Asics**

Classification

Will this product sell in 7 days?

Sold	...	ID	Geo	Domain	Title	Description	Tags	Brand
YES	...	104	US	Shoes	"Dark red..."	"Try this soft..."	["A, B, ..."]	Nike
NO	...	204	US	Shoes	"Women's..."	"Medium-size..."	["A, B, ..."]	Adidas
✓	...	**302**	**CA**	**Shoes**	**"Running..."**	**"All-terrain..."**	**["A, B, ..."]**	**Asics**

Regression

At which price will this product sell?

Price	...	ID	Geo	Domain	Title	Description	Tags	Brand
$52	...	104	US	Shoes	"Dark red..."	"Try this soft..."	["A, B, ..."]	Nike
$48	...	204	US	Shoes	"Women's..."	"Medium-size..."	["A, B, ..."]	Adidas
✓	...	**302**	**CA**	**Shoes**	**"Running..."**	**"All-terrain..."**	**["A, B, ..."]**	**Asics**

Forecasting Methods

Statistical

ARIMA, Exponential Smoothing, etc

- Theoretically grounded (>50 years of research)
- Mature tools and ecosystem
- Very popular
- Supported at Google via BQML ARIMA_PLUS

Neural Networks

CNN, RNNs, LSTMs

- New (~5 years)
- Active debate vs statistical methods
- Recently competed in prime competitions (M4, M5)
- Ecosystem is very raw
- Supported at Google via AutoML Forecast

	Statistical	Neural Networks
Data set		
Univariate series	🟢	🟡
Multivariate series	🟡	🟢
Few features	🟢	🔴
Many features	🔴	🟢
Patterns in Time Series		
Repeated patterns	🟢	🟢
Feature driven patterns	🔴	🟢
Other		
Cold starts	🔴	🟢
Short life cycle products	🔴	🟢

Method Fit: 🟢 great 🟡 ok 🔴 poor

Tree-based model risks

- Can’t extrapolate to values outside the training dataset.
- Doesn’t capture the sequences in the dataset as well as DNN models.
- Returns good evaluation metrics (illusion risk) but not always business value. Can have issues with under forecasting critical SKU’s.

Forecasting Methods

Neural Network builds a global model for all series, “cross-learns” from series.
It works well when

- there is lots of data (wide and long)
  - large number (100+) of time series (especially short series w/o data on repeated seasons)
  - Variable length time series, including short histories
  - lots of features
  - data has unstructured features (like product description text fields)
- series are strongly driven by features
- cold-starts

Statistical (autoARIMA) builds a separate model for each individual series. Typically retrained often.
It works well when

- series are strongly driven by seasonality and/or trends
- no limitation on dataset sizes, works equally well with
  - small number of series or univariate data
  - Long histories - enough to capture 2 or more seasonal
  - few or no features

Why use Deep Learning Models for Forecasting?

- Creates one "global" model for all time series
- Learns patterns **across** time series

Can use large number of drivers

✓ **Rich metadata**
(eg: product attributes, location attributes)

✓ **Historical factors**
(eg. inventory, weather)

✓ **Factors known in the future**
(eg: planned promotions/events, holidays)

Can model complex scenarios

✓ Cold start / new items

✓ Short product life cycles

✓ Burstiness, sparsity

✓ Unstructured data such as text descriptions

Vertex AI Forecast

1. Prepare Dataset — 2. Train Model(s) — 3. Forecast — 4. Visualize, Integrate

Retailer Data

- Past sales, inventory
- Product catalog
- Pricing and Promotions
- Marketing and Events
- Store information
- E-comm signals

Vertex AI Forecast

Data Preprocessing → **Modelling** → **Forecasts & Insights**

Validation + retail feature engineering

Proprietary Google models

Explainability

Data augmentation (soon)

Weather, holidays

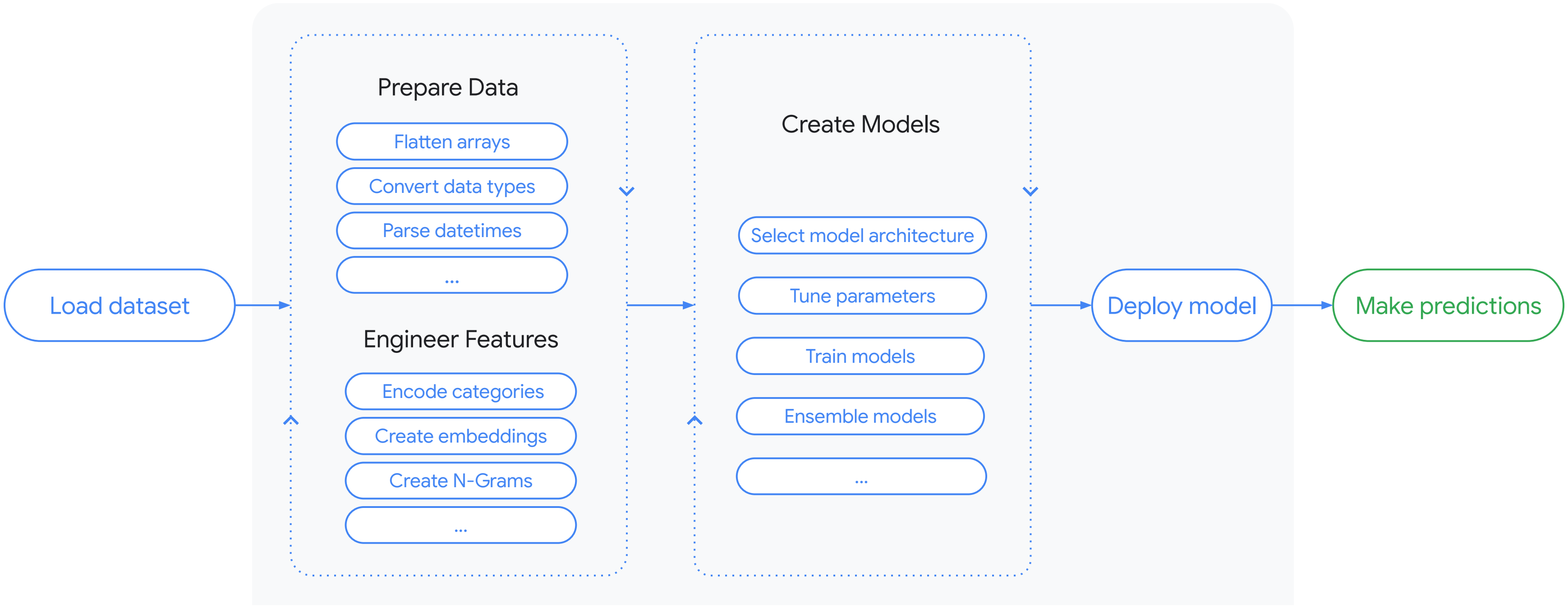
Integration

External visualization tools

Demand Planning Systems

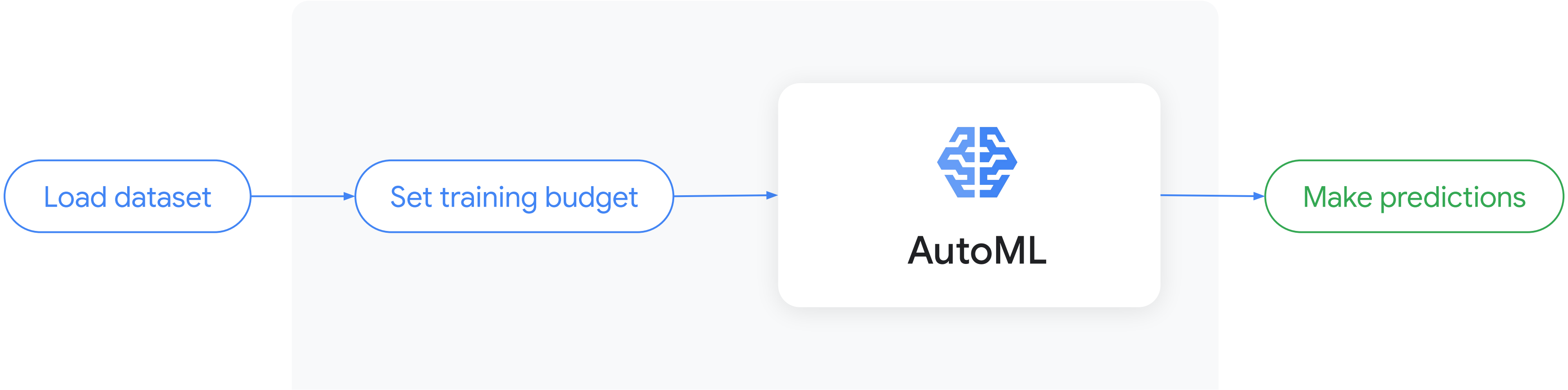
AutoML - Fastest path from data to value

Traditional Machine Learning Workflow

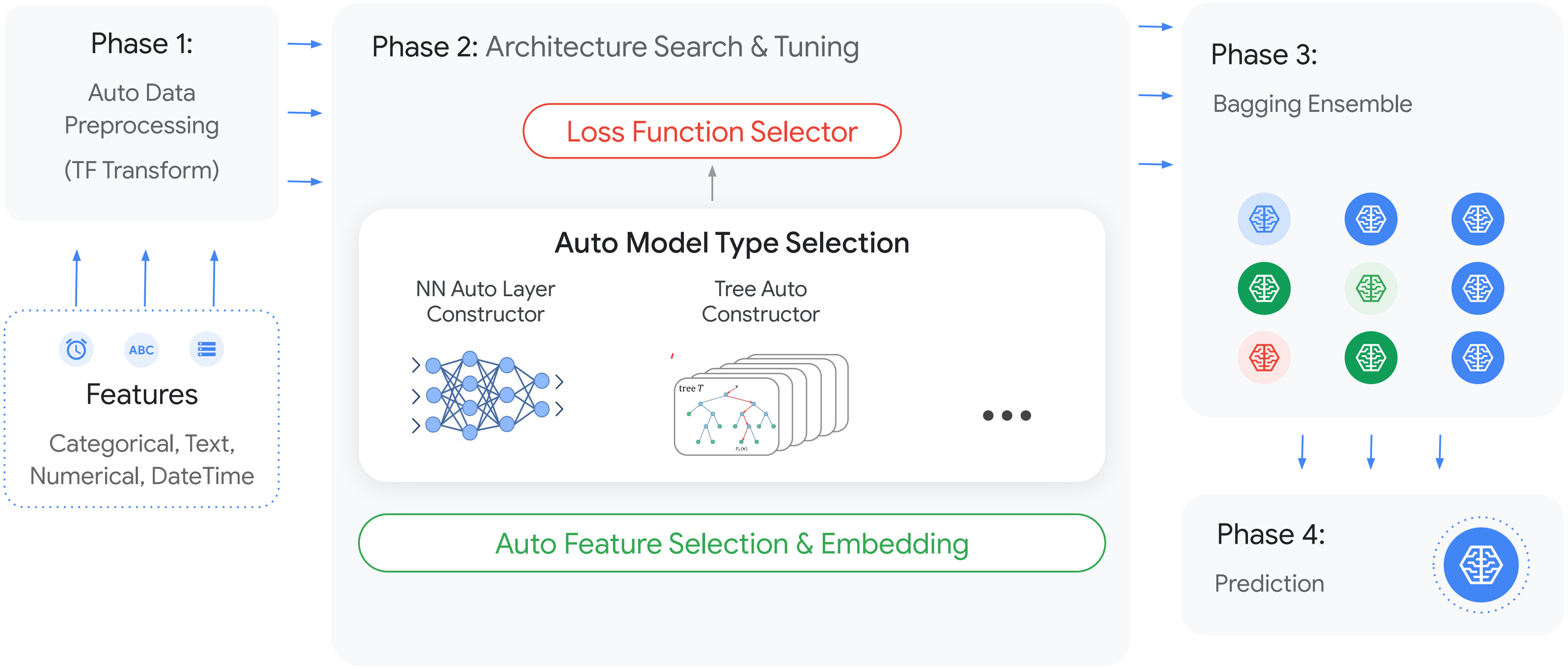


AutoML - Fastest path from data to value

AutoML Workflow



Powered by latest research from Google Brain



Automated feature engineering

Best practice transformations for all data types

Numbers: generate quantiles, log, z_score transforms

Datetime: extract year, month, day, weekday, categorize

Text: tokenize, generate n-grams, create embeddings

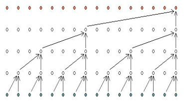
Arrays of categories: convert to lookup index, generate embeddings

Categories: one-hot encoding, grouping, embeddings

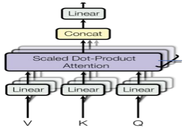
Nested fields: flatten, apply type transformations

Automated model architecture search

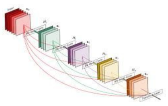
Evaluating Google's best model architectures for forecasting



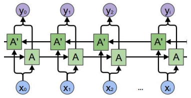
Convolutions



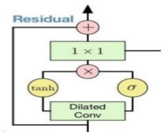
Attention



Skip Connections



LSTM



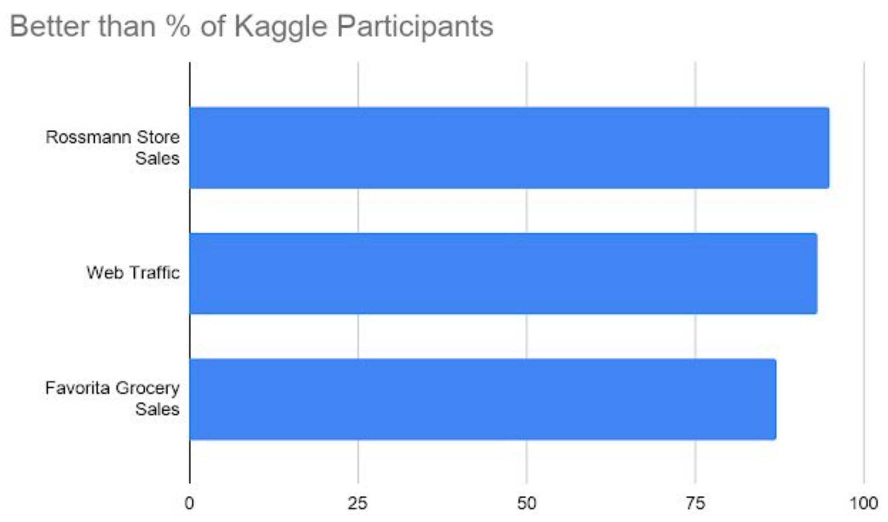
Gating

...

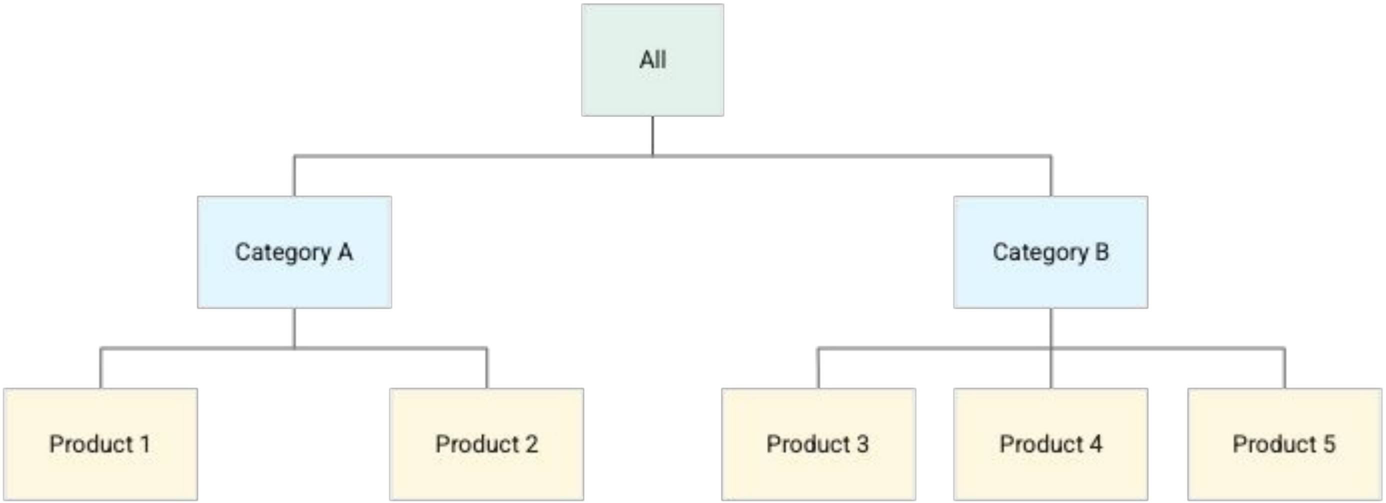
Benchmarks on Kaggle Datasets

Finished in **top 2.5%** (138 out of 5558 teams) in World's Top Forecasting competition.
M5: Estimate the sales of Walmart retail goods.

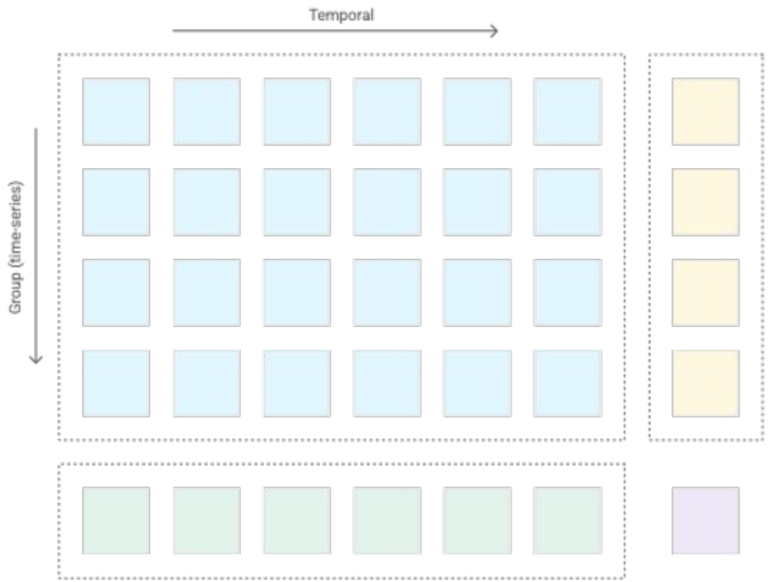
Consistently ranks in top 20% across a variety of datasets in different industries.



Hierarchical Forecasting



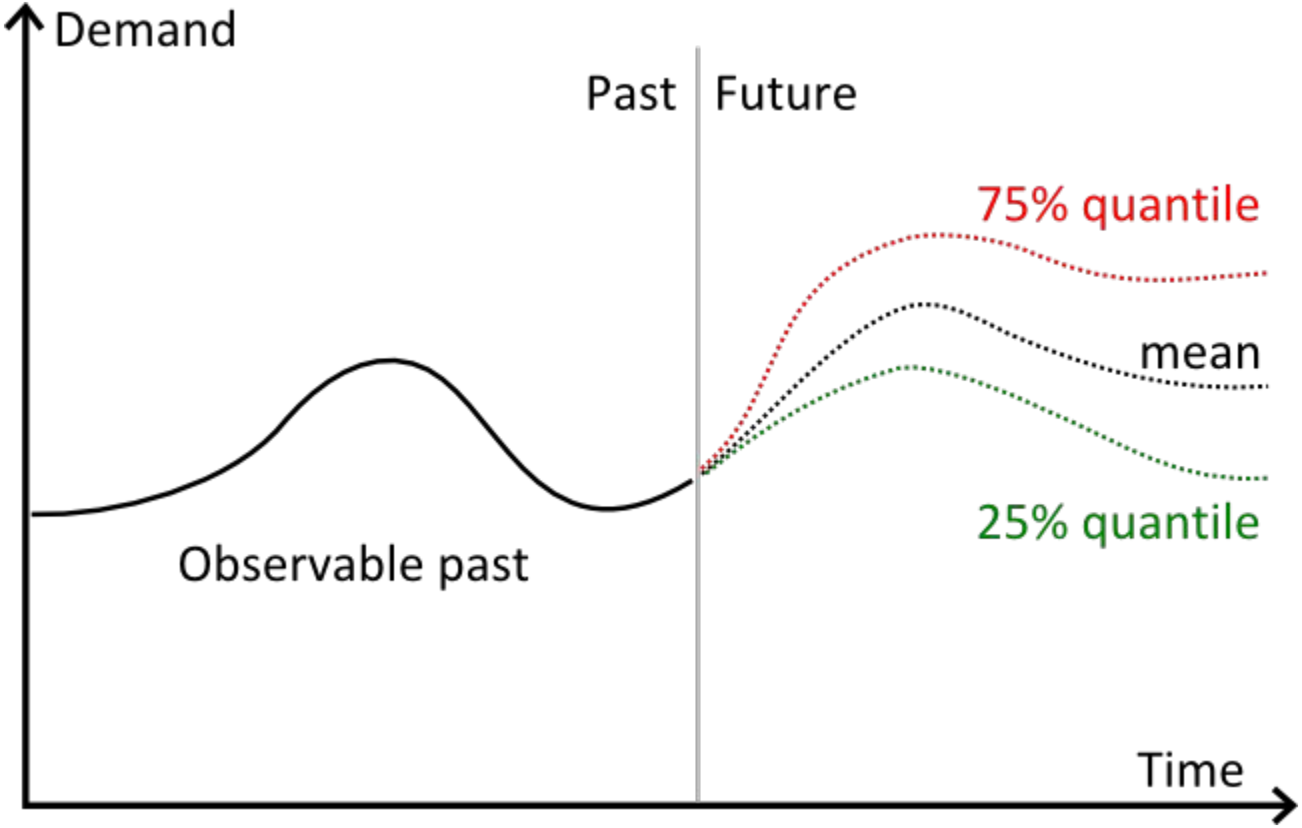
Hierarchical Forecasting



Hierarchical loss = 1 x loss
+ temporal total weight x temporal total loss
+ group total weight x group total loss
+ group temporal total weight x group temporal total loss

- **Reduce overall bias** to improve metrics over all time series (total sales).
- **Reduce temporal bias** to improve metrics over the horizon (season sales).
- **Reduce group level bias** to improve metrics over a group of time series (item sales)

Quantiles


Demand
Past
Future
75% quantile
mean
25% quantile
Observable past
Time

Business and Financial Evaluation

Given you have the baseline, AutoML and actuals you can run a financial analysis as traditional ML metrics fail to observe the financial impact of over forecasting vs under forecasting.

Baseline >= Actual >= AutoML

Baseline > AutoML > Actual

Actual > AutoML > Baseline

Actual >= Baseline >= AutoML

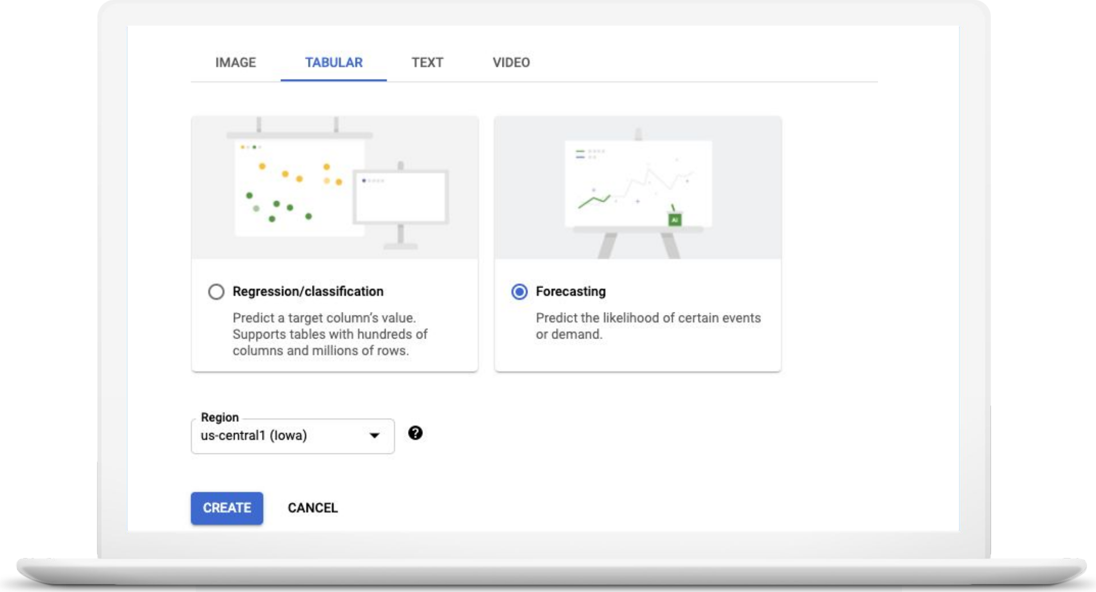
AutoML > Baseline > Actual

AutoML >= Actual >= Baseline

Gain = (AutoML-Baseline) * cost of under forecasting

Loss = (Baseline-AutoML) * cost of under forecasting

Variety of Interfaces



Intuitive Web UI

API + SDK

```
In [ ]: # The number of hours to train the model.
        model_train_hours = 1 #@param {type:'integer'}

        create_model_response = tables_client.create_model(
            model_display_name=MODEL_DISPLAY_NAME,
            dataset=dataset,
            train_budget_milli_node_hours=model_train_hours*1000,,
            exclude_column_spec_names=['fnlwgt','income'],
        )

        operation_id = create_model_response.operation.name

        print('Create model operation: {}'.format(create_model_respon
        se.operation))
```

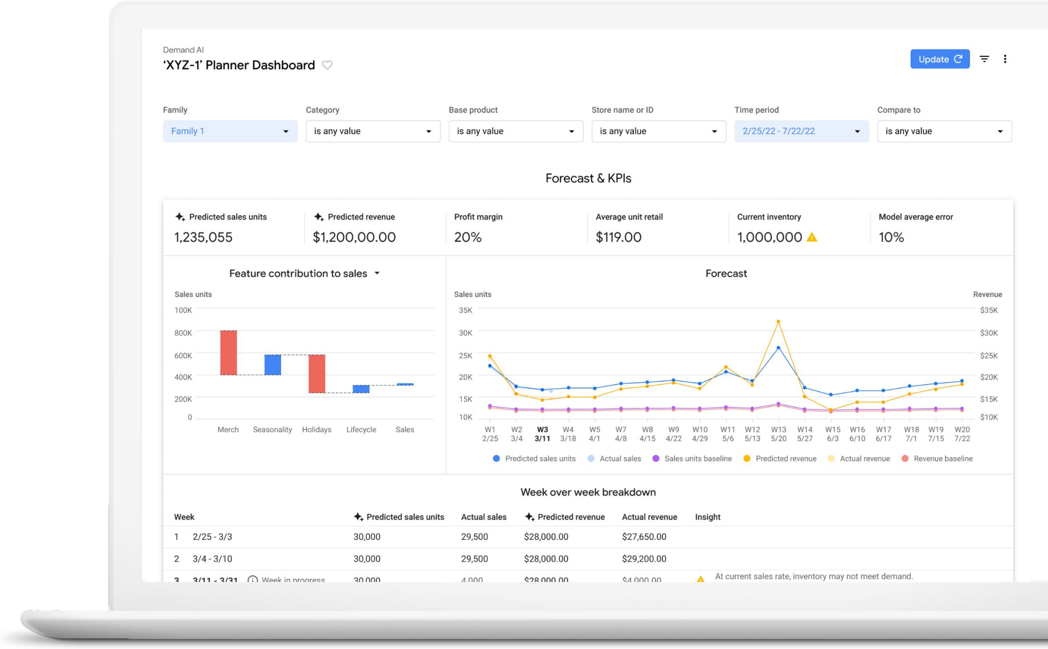
BigQuery SQL*

```
CREATE OR REPLACE MODEL project_id.mydataset.mymodel
       OPTIONS(model_type='AUTOML_REGRESSOR',
               input_label_cols=['fare_amount'],
               budget_hours=1.0)
AS SELECT
  (tolls_amount + fare_amount) AS fare_amount,
  pickup_longitude,
  pickup_latitude,
  dropoff_longitude,
  dropoff_latitude,
  passenger_count
FROM `nyc-tlc.yellow.trips`
WHERE ABS(MOD(FARM_FINGERPRINT(CAST(pickup_datetime AS STRING)), 100000)) = 1
```

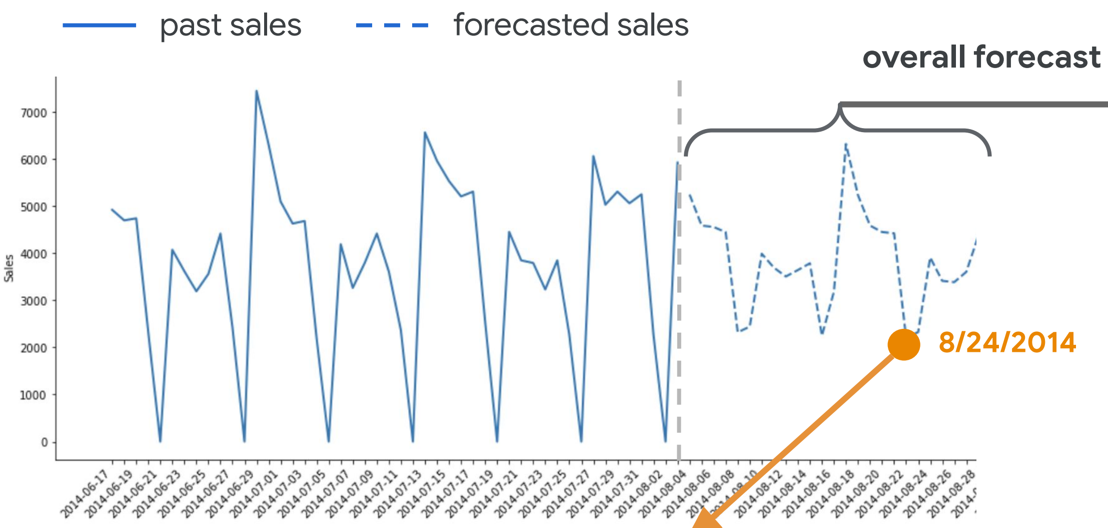
** currently only supports auto-Arima (no AutoML support)*

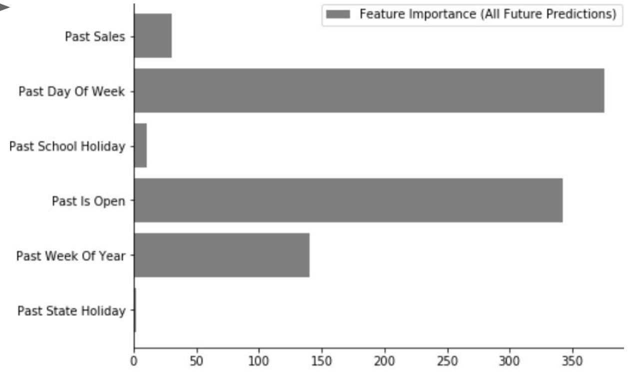
XAI for time-series / forecasting

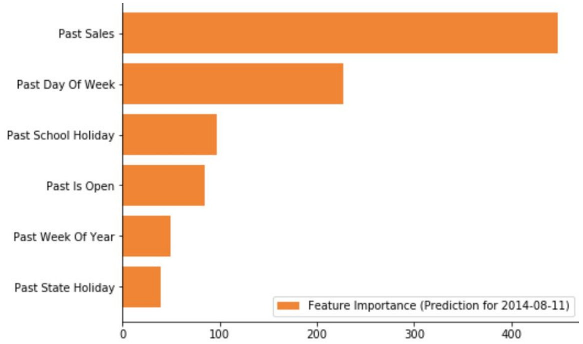
- Feature attributions for time-series models based on Sampled Shapley method
- **Forecasting models differ from normal tabular models** in the form of their inputs and the way we construct baselines.
    - 3-dimensional data (num_samples X num_historical_instances X num_features)
    - Attributions are generated per-feature, *per-historical* instance
    - Baselines are generated taking into account the time-ordering of samples
- **Aggregations.** These can be aggregated over historical instances to obtain aggregate feature-attributions.
    - E.g. for a retail sales forecasting use-case, feature-attribution aggregations can be by *product, location or time-slice*



Applying feature attributions to time-series







On **08/24/2014** the most important feature is

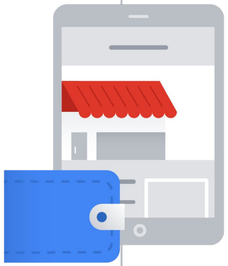
- "Past Sales".

Overall the most important features are

- Past day of the week
- Past is Open
  (whether the store was open)

Case Studies

Large US-based Apparel Retailer



Challenges

For this **high-quality, on-trend apparel retailer**, demand planning is key to business success. WIth disparate systems, the company recognized that it required a new approach to address challenges involving big data, granularity of predictions, and real-time demand signals.

Empowering business decisions with improved forecasting

Leveraging Vertex AI Forecast, the company saw increased accuracy for demand **and** labor forecasting while integrating new supply chain requirements into its forecasting capabilities.

2-3 month
reduction in model development time

$5M-$10M per year
savings from enhanced labor efficiency

$40M+
estimated additional revenue from improved supply chain and product allocation

Case Study 1:
Pre-season forecast for Apparel

- **Both Pre- and in-season:** forecast up to more than a year, based on data up to *three months* before the season starts
- **Cold start items:** Products relationships have to be learned automatically.
- **Fine granularity:** breakdown demand forecast to product variants (eg. color x size) down to the day x SKU level.

?

Retail Forecasting Primer: Item Demand

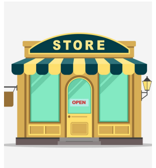
PRODUCT METADATA

- Summer dress
- Lightweight. Pattern.
- Long-sleeve. V-neck.
- 55% Linen. 45% Cotton
- Machine wash

DEMAND DRIVERS

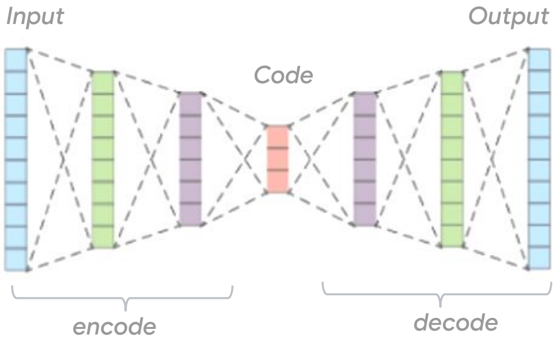
- Sales
- Price
- Competitor price
- Promotions/Events
- Holidays

STORE DATA



- Store description: Large, small, specialty
- Store location
- Foot traffic

AUTOML FORECAST



Machine learning models understand rich metadata, relationships between products and the joint effect of pricing, competition and product lifecycle.

MEDIUM HORIZON
12-16 month

- e.g. pre-season planning
- Buy/order planning
- New & Cold start items

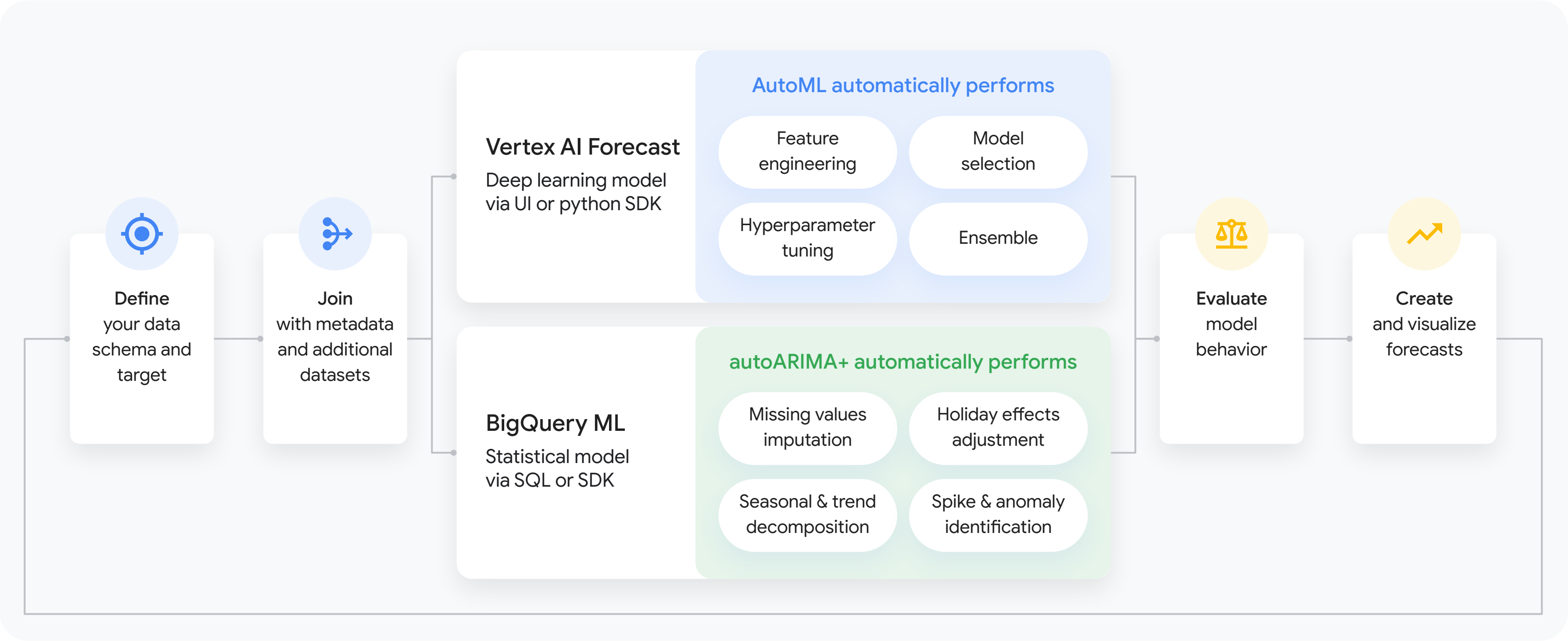
SHORT HORIZON
0-8 weeks

- in-season planning
- Replenishment, inventory
- Pricing, allocation
- SKU-Level

Vertex AI Forecast: Workflow & Demo

Forecasting Workflows on Google

Build high quality, scalable forecasting solutions with deep learning and statistical models from Google.



Easy to Get Started

Data

Load Dataset → Define Columns → Define Features

Generate Statistics

Run Validations

Supported sources:
- CSV
- BigQuery Table
- BigQuery View

Required inputs:
- Target
- Time Series Identifier
- Time

Required inputs:
- Type Transformations
- Time Dependence
- Availability at Prediction Time

Model

Setup Model → Train Model → Evaluate Model

Required inputs:
- Forecast Horizon
- Context Window
- Data Split
- Optimization Objective

Required inputs:
- Training Budget

Automated Feature Engineering

Auto Model Architecture Search

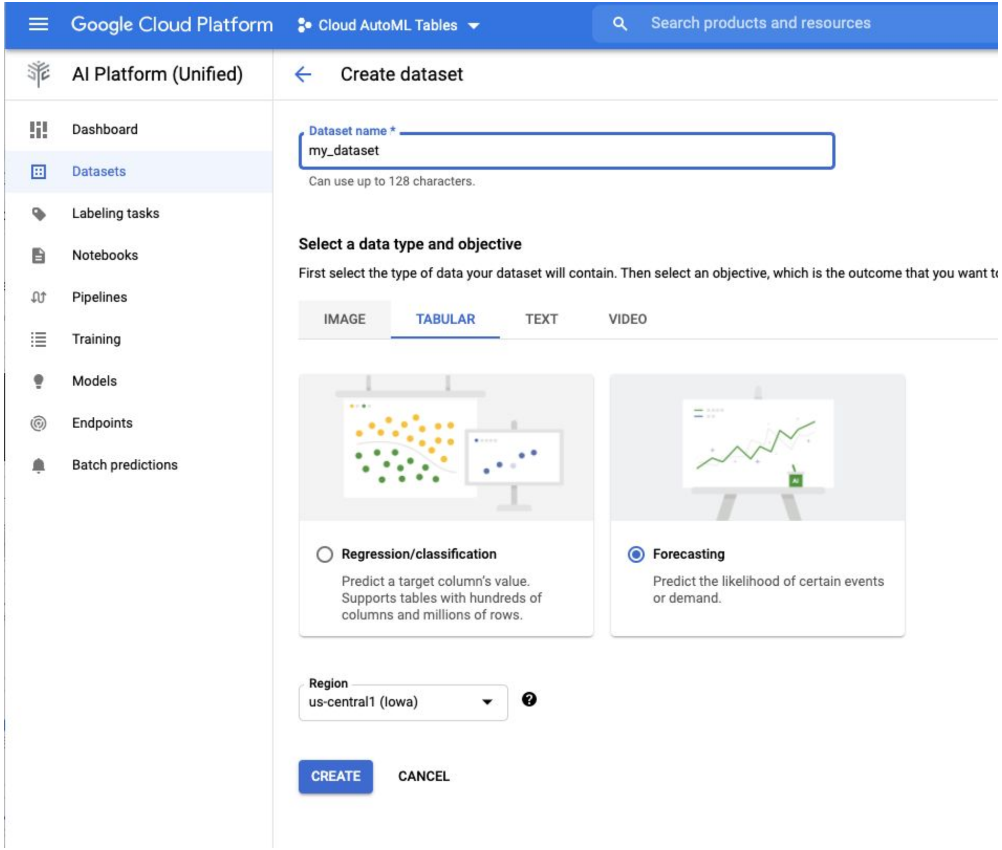
Prediction

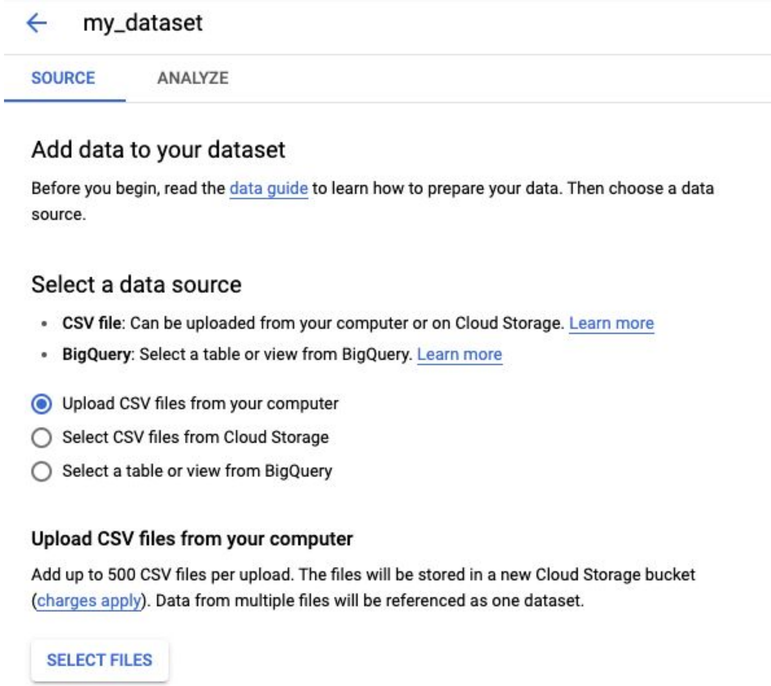
Create Request → Create Forecast

Required inputs
- Time
- *Feature Values*

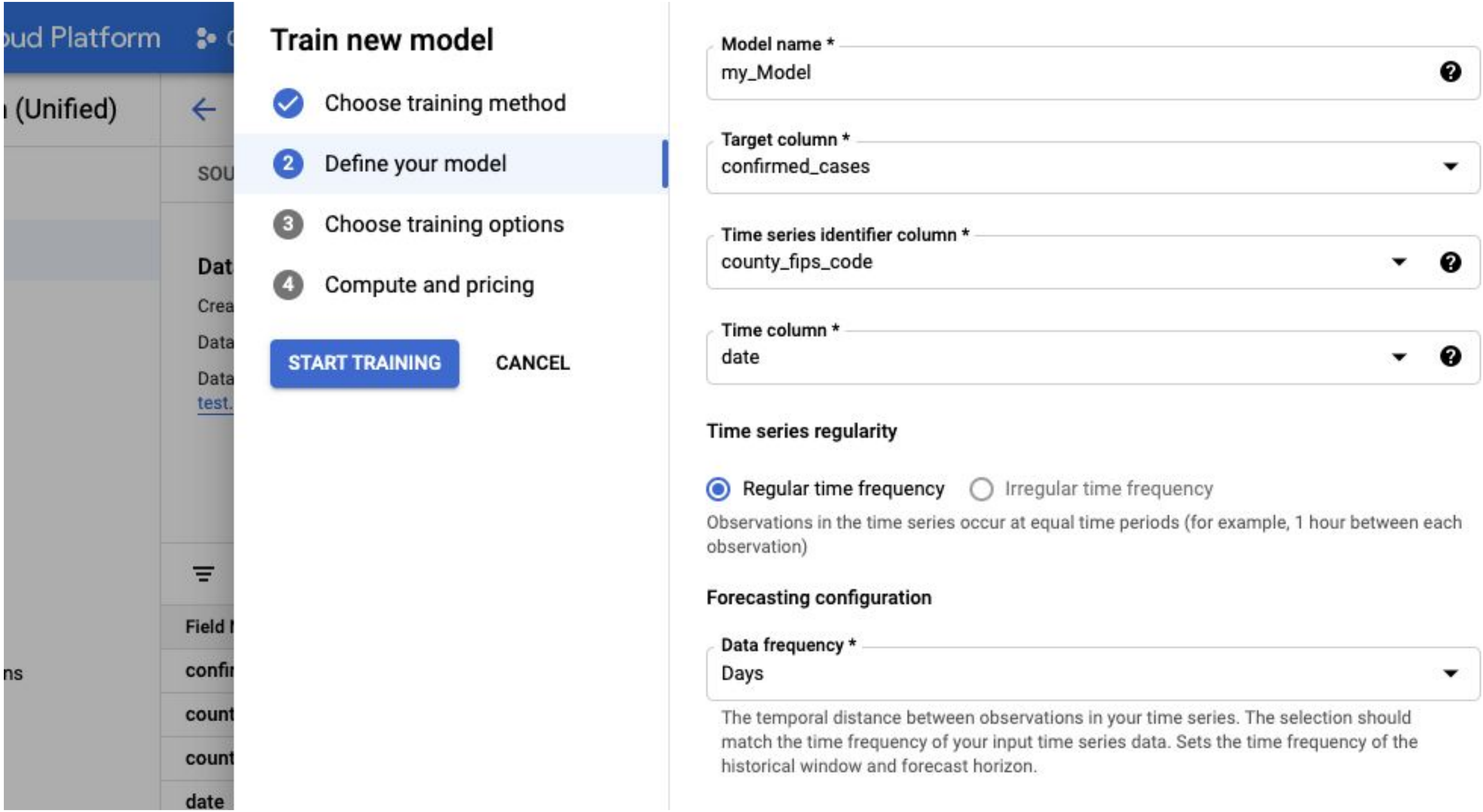
Output to
- CSV
- BigQuery

Demo: Create Dataset

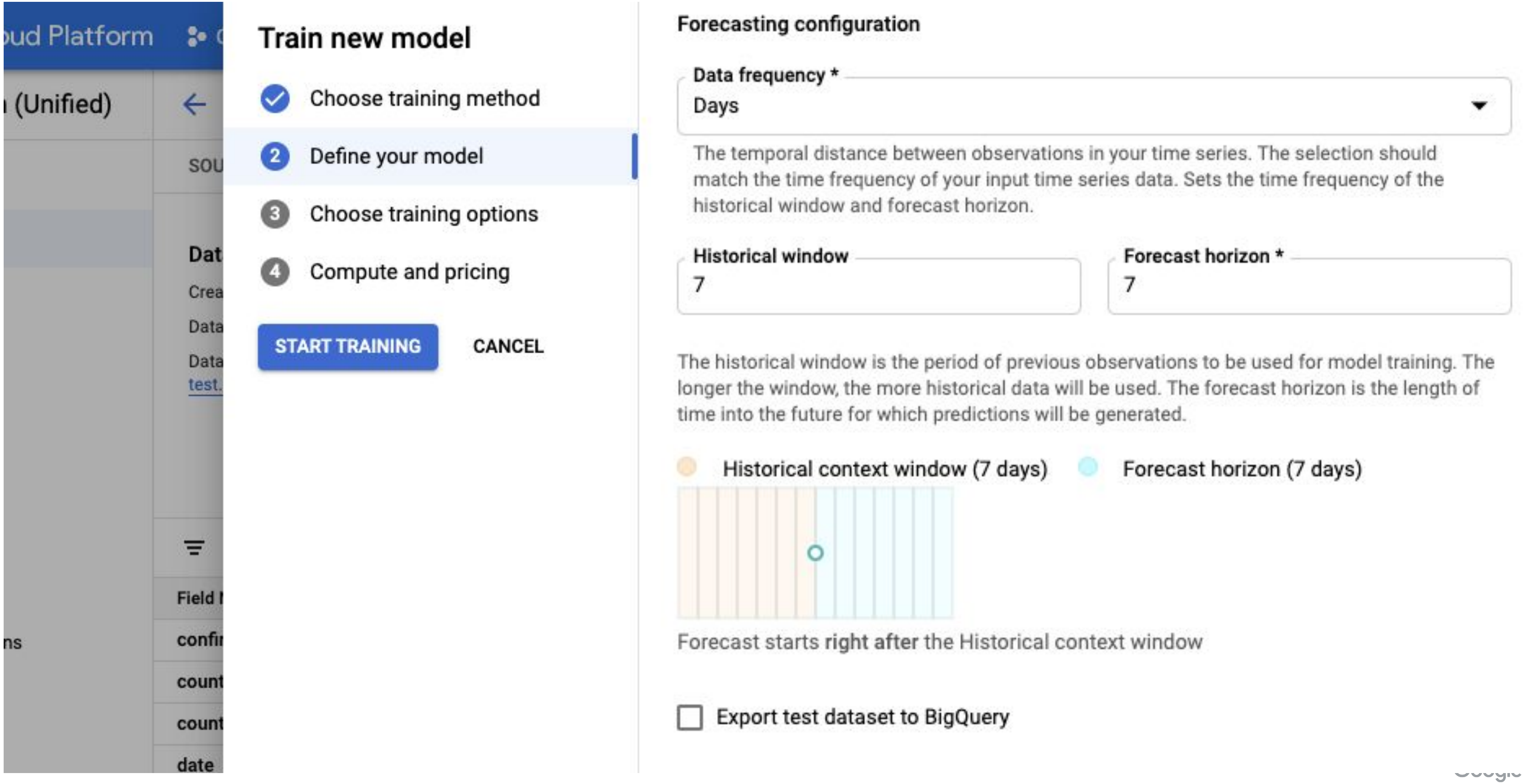




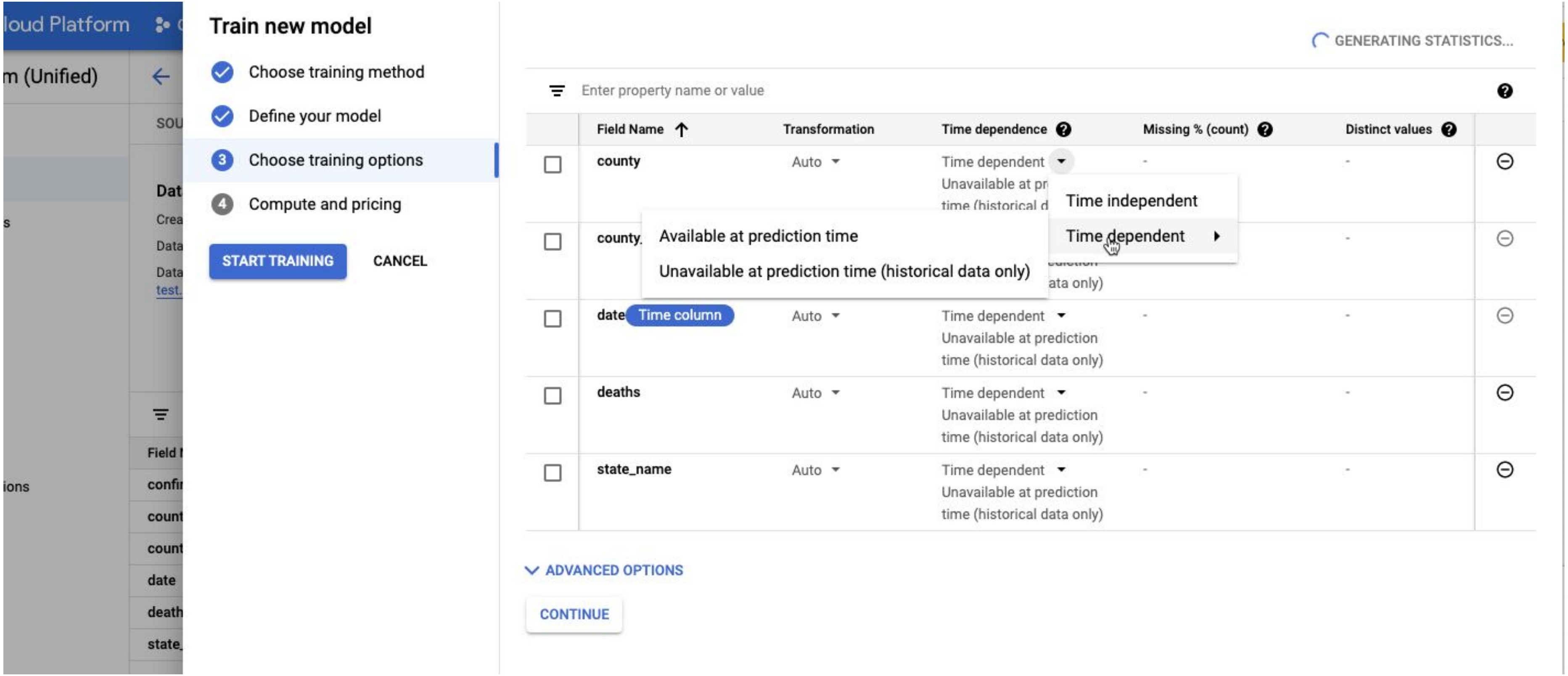
Demo: Training



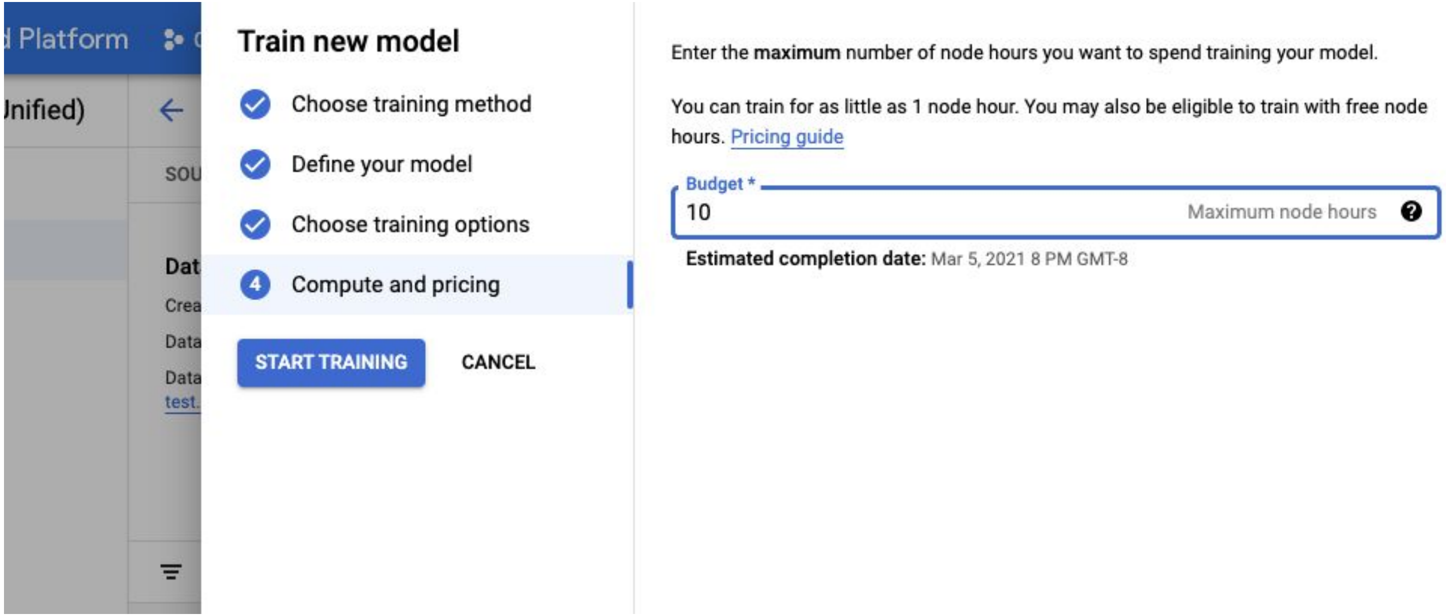
Demo: Training



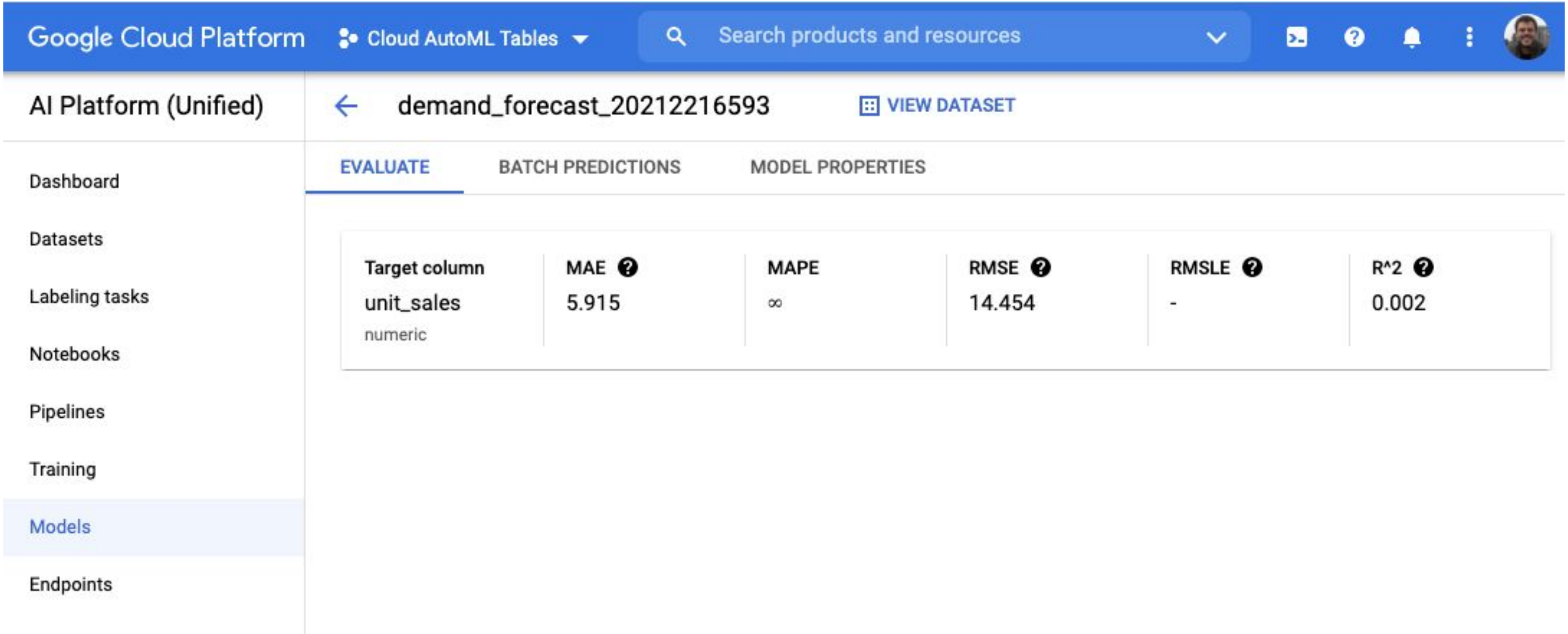
Demo: Training



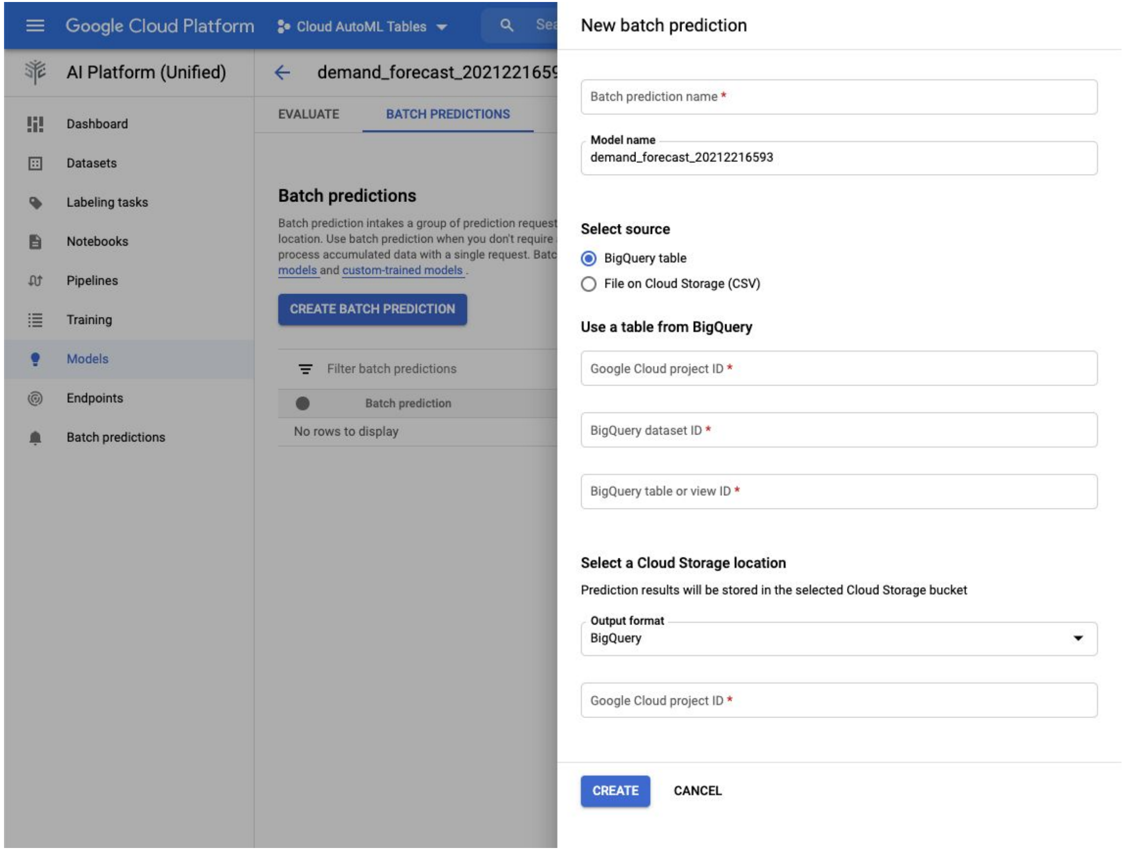
Demo: Training



Demo: Evaluate Model



Demo: Create Prediction



Questions