GDG devfest Hellas 2020

Automated Machine Learning in 2020

Panagiotis Papaemmanouil
Data Scientist, Medoid AI

Medoid AI

About me,

- BSc **Mathematics**, AUTh
- MSc **Data and Web Science**, AUTh
- **Data Scientist** at Medoid AI
- Focusing on Financial Machine Learning

Medoid AI

Presentation Outline:

Automated Machine Learning (Auto ML) in 2020

Introduction

1. Motivation
2. Definition

Auto ML in Action

1. Feature Engineering
2. Model Selection
3. Hyperparameter Optimization
4. Neural Architecture Search

Tools & Frameworks

1. Open Source & Commercial tools
2. Code and GUI examples

Auto ML Case Study

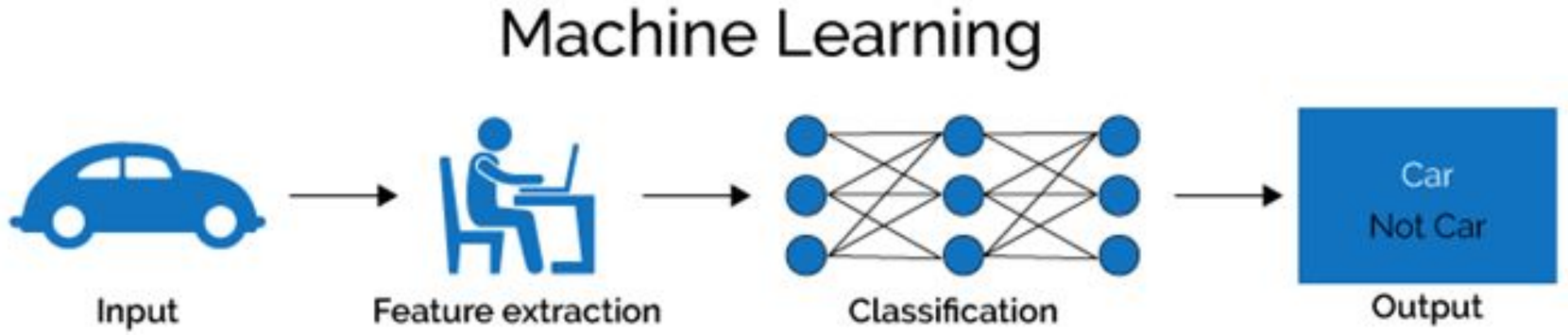
Google Cloud Platform (GCP),
Artificial Intelligence, Tables

Medoid AI

Introduction

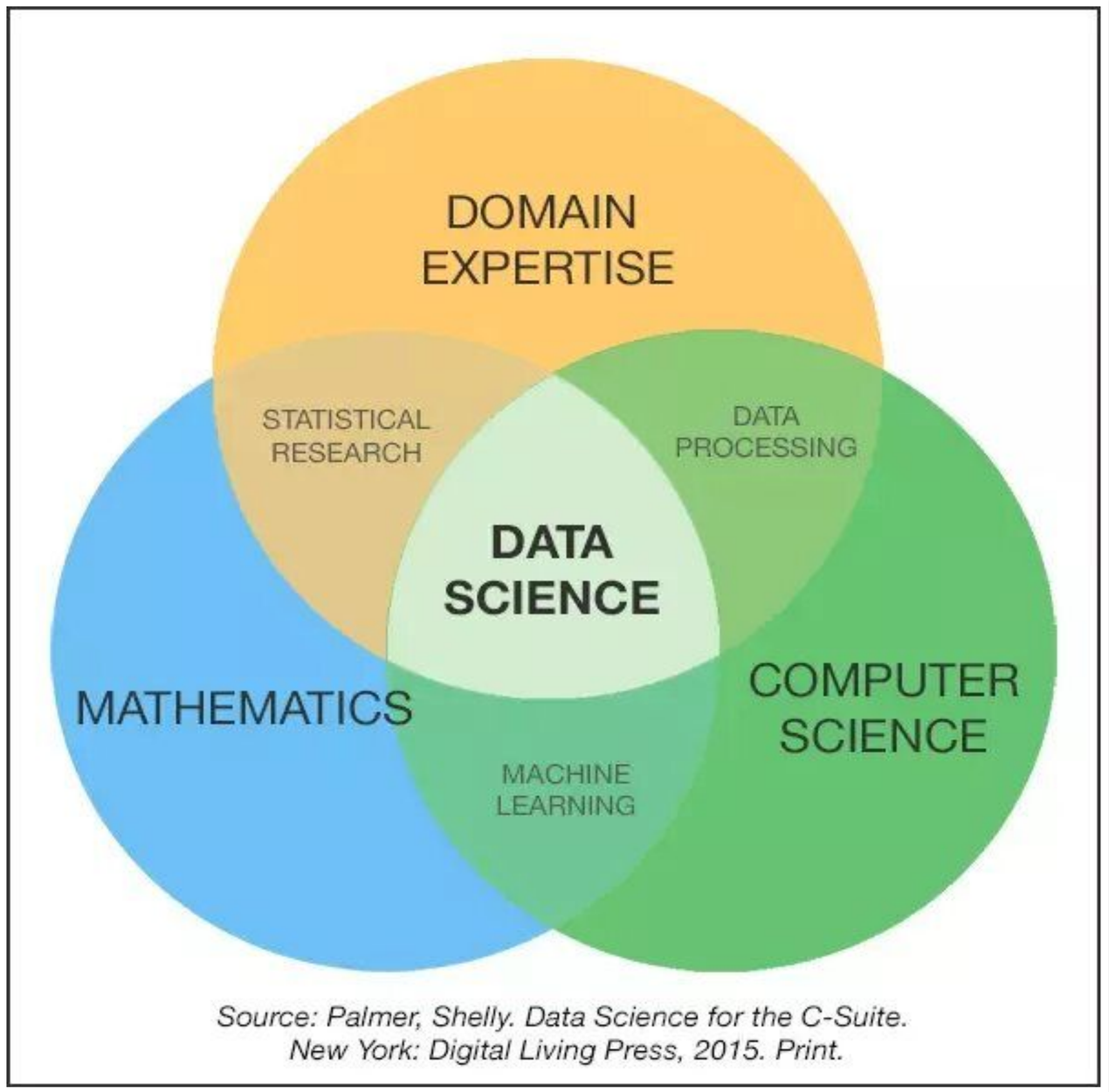
Machine Learning: “A field of study that gives computers the ability to learn without being explicitly programmed.”

1959, Arthur Samuel



Medoid AI

Full stack Data scientists are rare creatures!



Source: Palmer, Shelly. Data Science for the C-Suite. New York: Digital Living Press, 2015. Print.

- Knowledge of the business domain and business problems
- Knowledge of the data

- Ability to write code to gather data
- Ability to write code to explore/inspect data
- Ability to write code to manipulate data
- Ability to write code to extract insights
- Ability to write code to build models
- Ability to write code to implement models

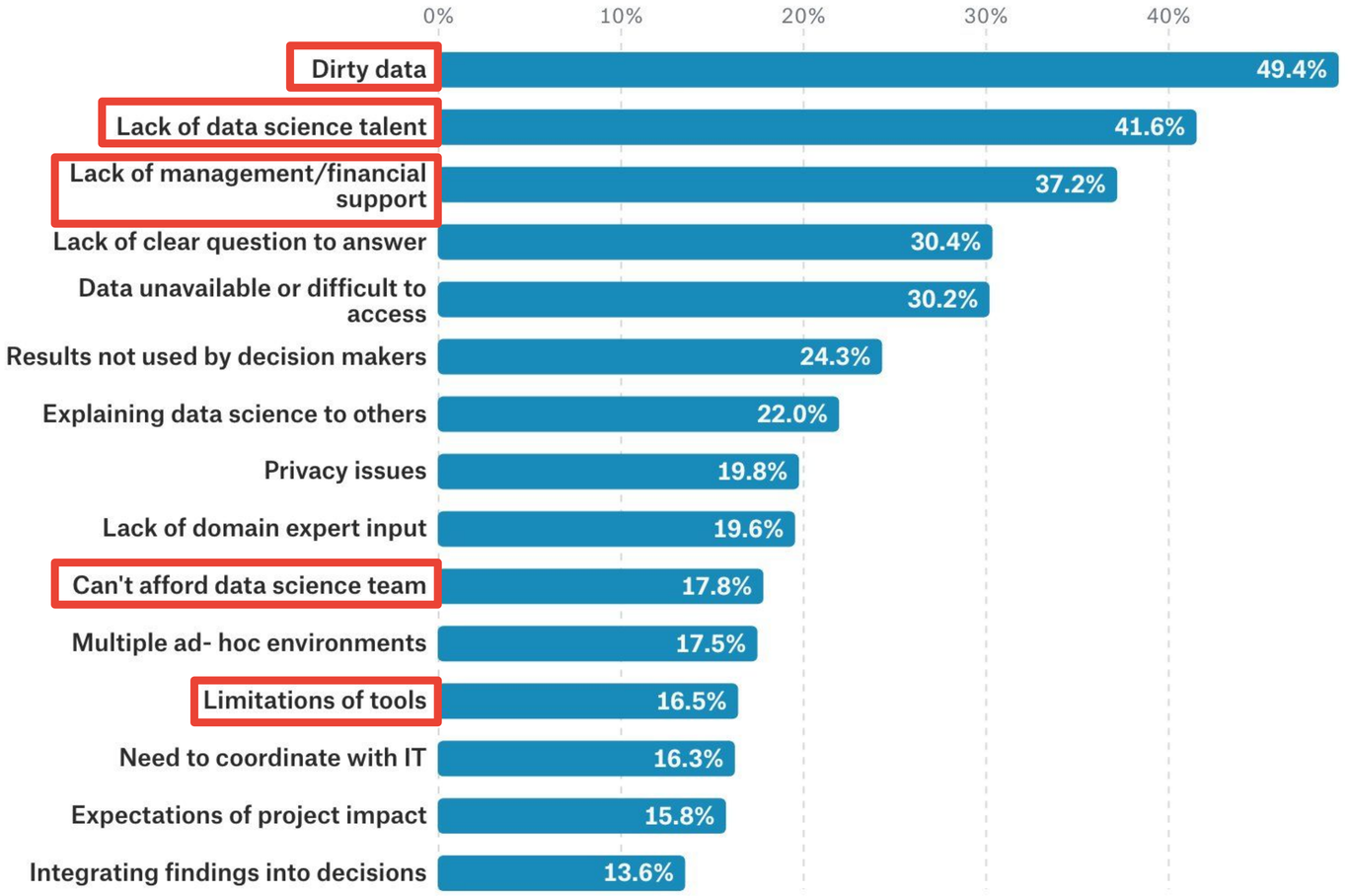
- Statistics
- Internal of Algorithms
- Practical Knowledge and experience
- Knowing how to interpret and explain models

Motivation

I. The Data Science case of Auto ML

1. ML pipeline building involves ***Repetitive***, ***Time consuming*** Tasks
2. Some tasks require a lot of **manual work**
3. ML is still somehow ***Difficult*** to build, deploy and maintain

II. The Business case of Auto ML



Data Science barriers or challenges in workplace

Medoid AI

Source: Kaggle surveys

The solution is

Automated Machine Learning

Medoid AI

“Automated Machine Learning (or Auto ML) is the process of automating the process of applying machine learning to real-world problems.”

*Thanks **wikipedia** for the so not obvious definition*

Gartner Said (at 2017) that “More Than

40%

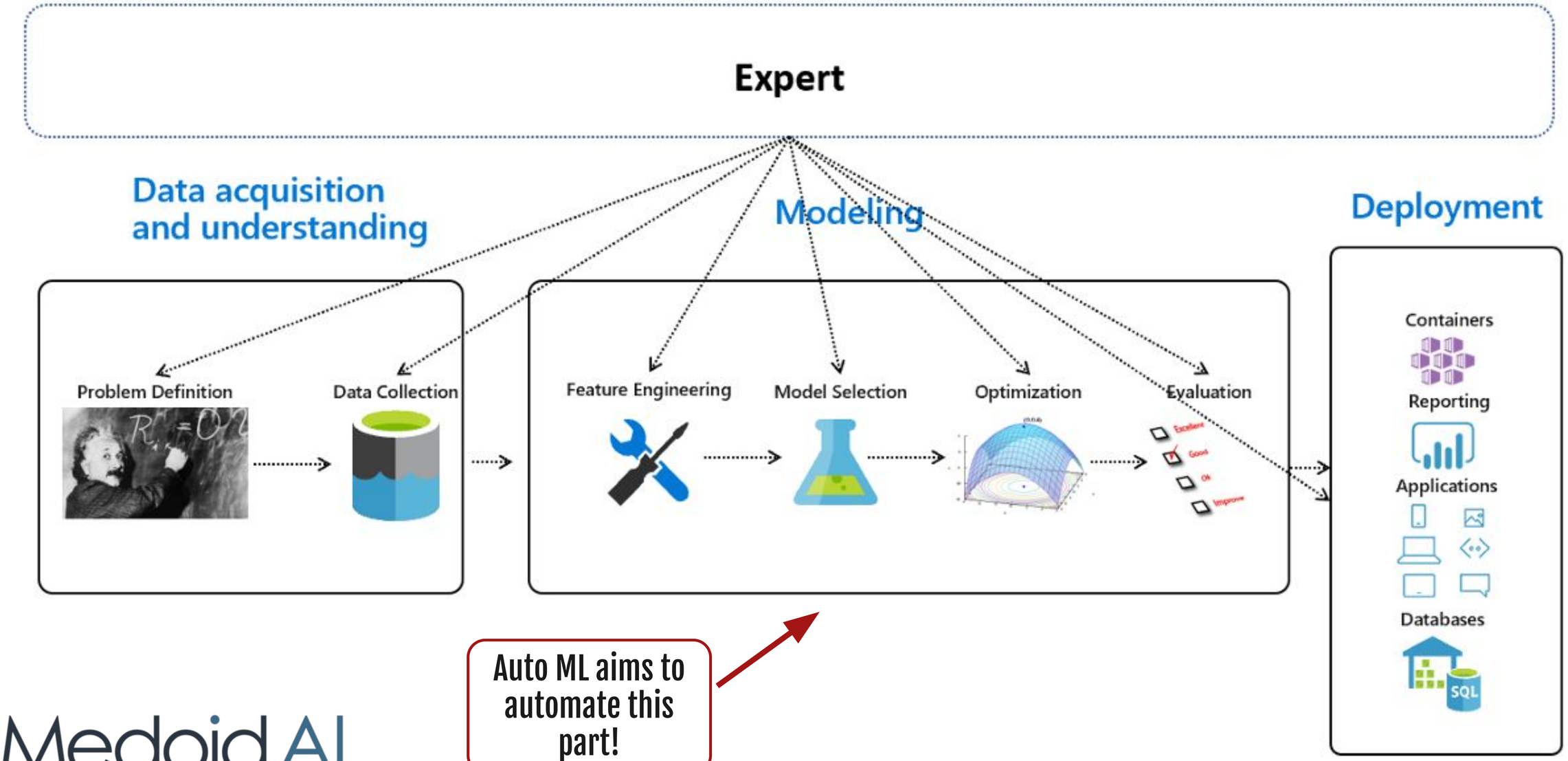
of Data Science Tasks Will Be Automated by 2020”

Source:https://www.gartner.com/en/newsroom/press-releases/2017-01-16-gartner-says-more-than-40-percent-of-data-science-tasks-will-be-automated-by-2020

Medoid AI

Auto ML in Action

Machine Learning project flowchart



Medoid AI

I. Feature Engineering

1. Data Cleaning

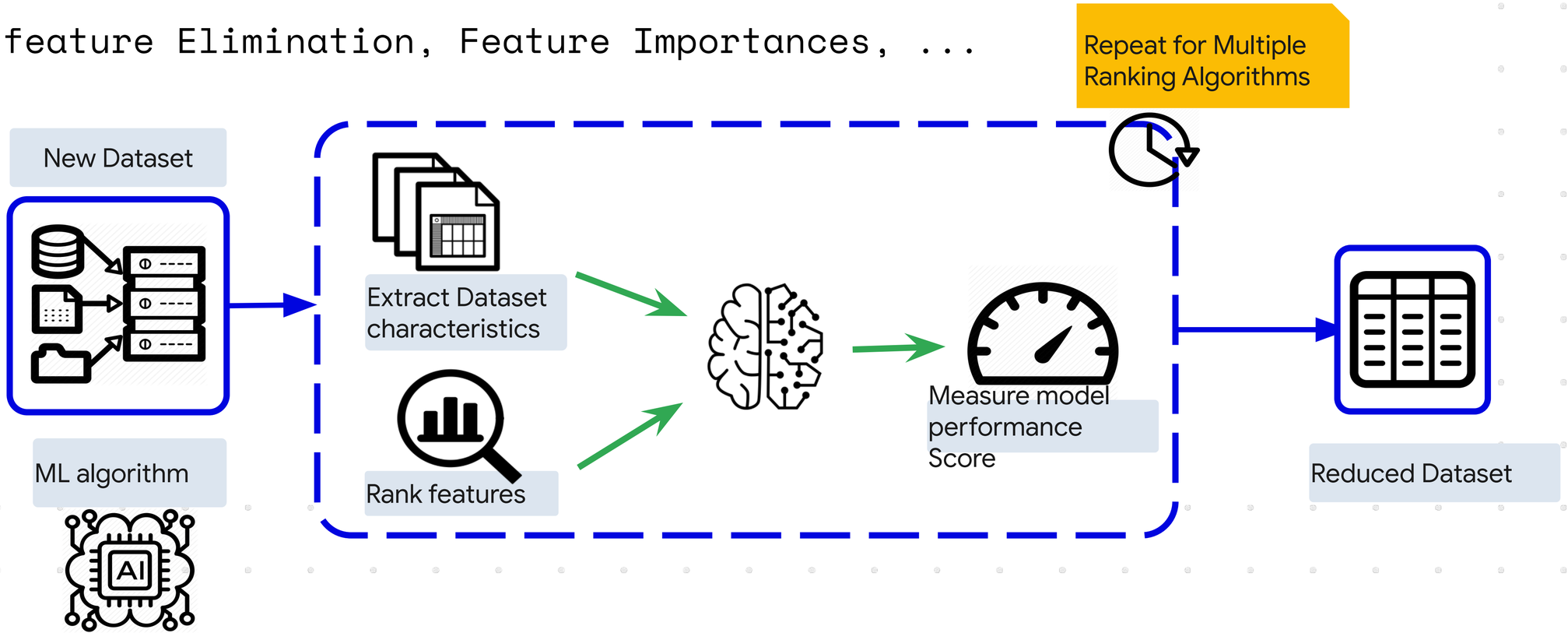
Data type identification, NA imputation, Scaling, Outliers, Categorical Encoding, ...

2. Feature Extraction

Deep Feature Synthesis

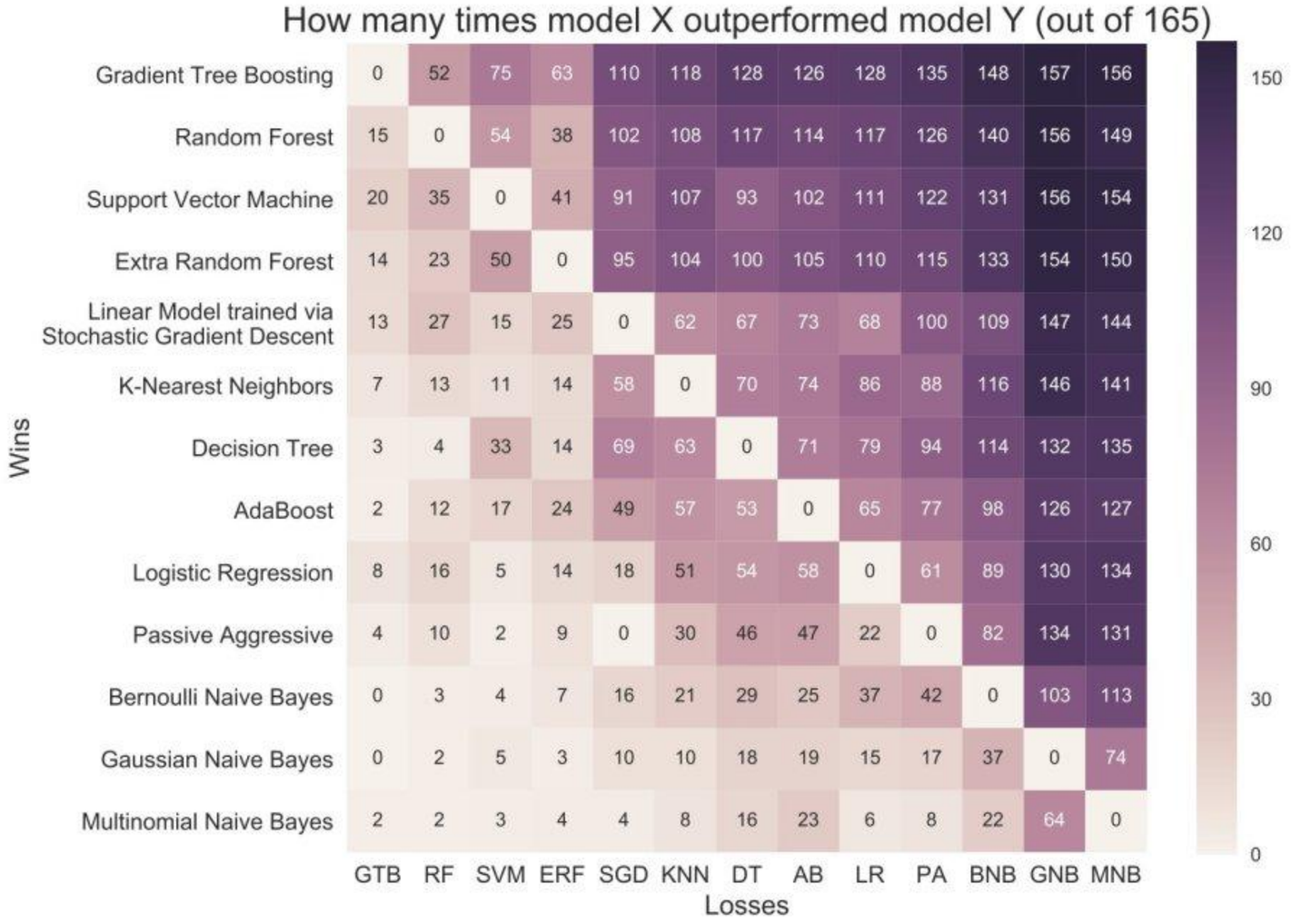
3. Feature Selection

Recursive feature Elimination, Feature Importances, ...



II. Model Selection: No free lunch Theorem (1/2)

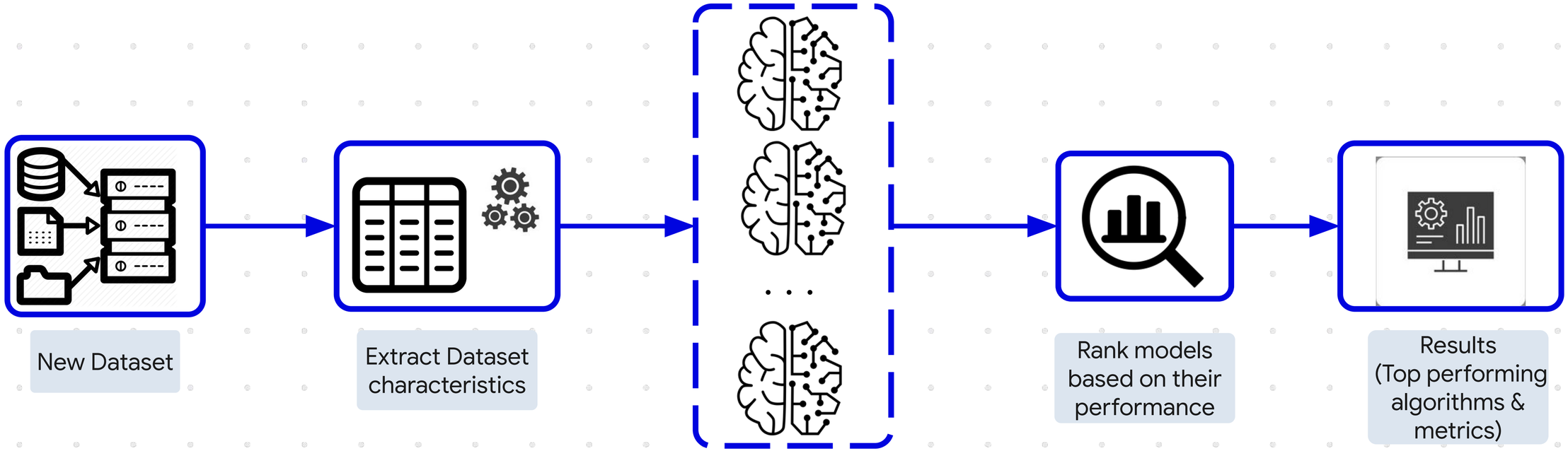
- Always check your **assumptions** before relying on a model or search algorithm.
- There is no ***"super algorithm"*** that will work perfectly for all datasets.



Source: sklearn documentation

II. Model Selection (2/2)

1. There is no best algorithm for all problems - *No free lunch theorem*
2. Best algorithm it is not intuitive
3. Complex models are not always optimal (bias-variance trade-off)

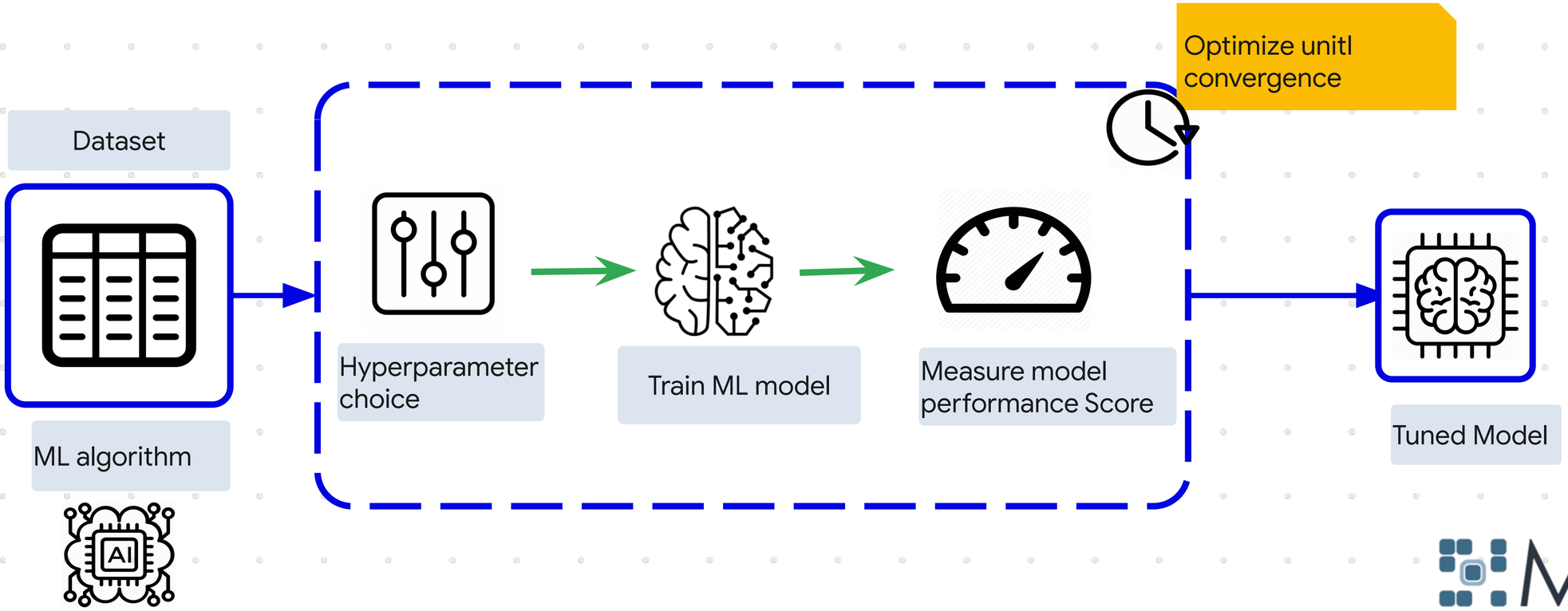


Medoid AI

III. Hyperparameter Optimization

1. Default parameters are almost always bad
2. Hyperparameter tuning requires a lot of repeating experiments
3. Selection of best hyperparameters set is based on experiments and heuristics

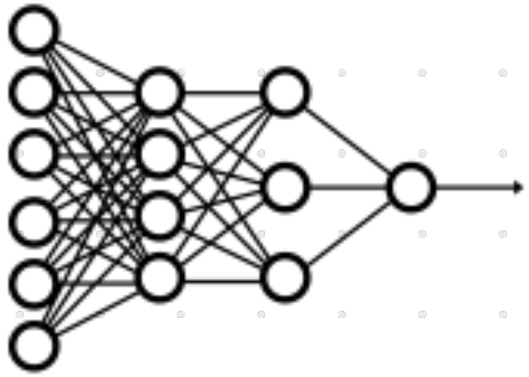
   (grid search, random search, bayesian optimization, evolutionary optimization, early stopping)



Medoid AI

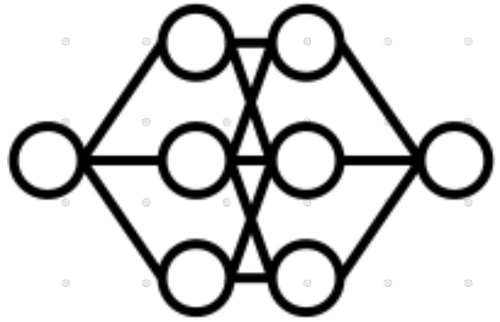
What about Deep Learning?

Automated Deep Learning:
Neural Architecture Search (NAS)



Approaches differentiate based on:

1. Search space
2. Search strategy
3. Performance estimation strategy



Medoid AI

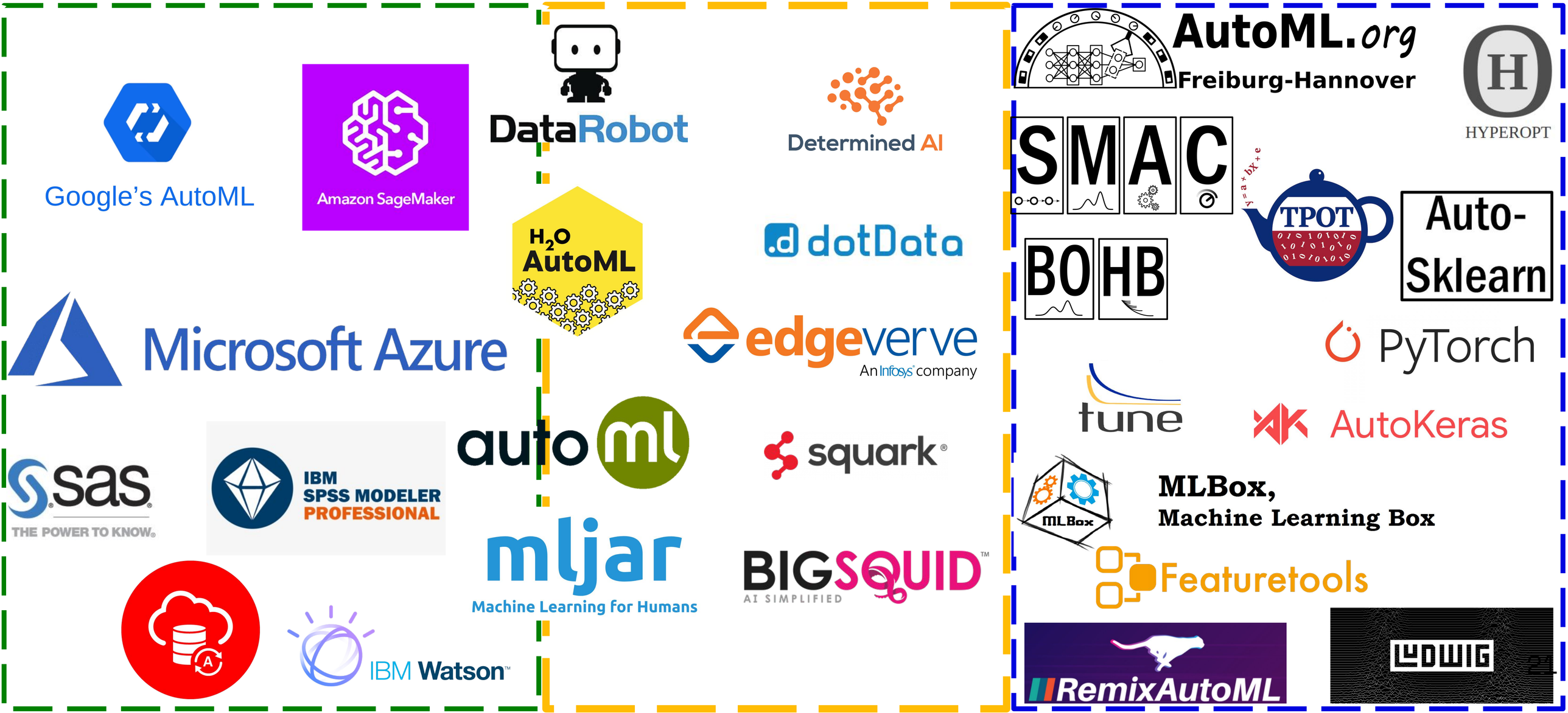
Tools & Frameworks

Tools & Frameworks Taxonomy

Tech Giants

Start-ups

Open Source



Notes

1. Some Tools aims to automate the **entire ML pipeline**, while others are focused on **specific tasks**
   (such as Features engineering, Model Selection or Hyperparameter optimization)

2. Some tools require **code**, other offers **GUI** and other supports **both** of them.

Auto ML: Code example (I)

```
from azureml.train.automl import AutoMLConfig

# task can be one of classification, regression, forecasting
automl_config = AutoMLConfig(task = "classification")

from azureml.core.dataset import Dataset
data = "data/path"
dataset = Dataset.Tabular.from_delimited_files(data)

automl_classifier=AutoMLConfig(
    task='classification',
    primary_metric='AUC_weighted',
    experiment_timeout_minutes=30,
    blocked_models=['XGBoostClassifier'],
    training_data=train_data,
    label_column_name=label,
    n_cross_validations=2)
```

Source: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-train

Auto ML: Code example (II)

H2O AutoML in R

```
library(h2o)
h2o.init()

train <- h2o.importFile("train.csv")

aml <- h2o.automl(y = "response_colname",
                  training_frame = train,
                  max_runtime_secs = 600)
lb <- aml@leaderboard
```

Simple API

Classification

```
from automl import Pipeline

est = Pipeline(task='classification',
               scoring='accuracy')
est.fit(X_train, y_train)
y_pred = est.predict(X_test)
```

Regression

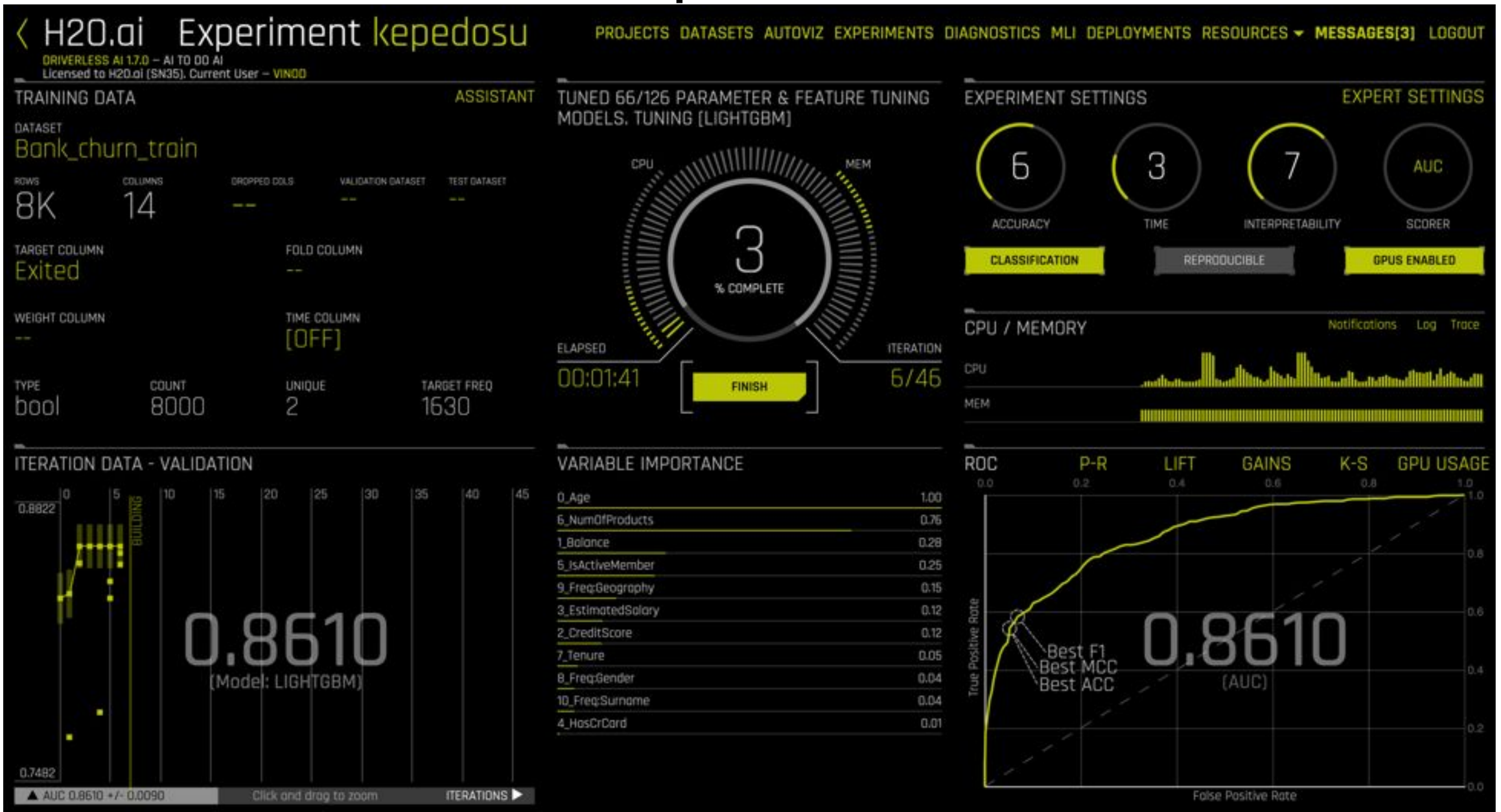
```
from automl import Pipeline

est = Pipeline(task='regression',
               scoring='neg_mean_squared_error')
est.fit(X_train, y_train)
y_pred = pipe.predict(X_test)
```

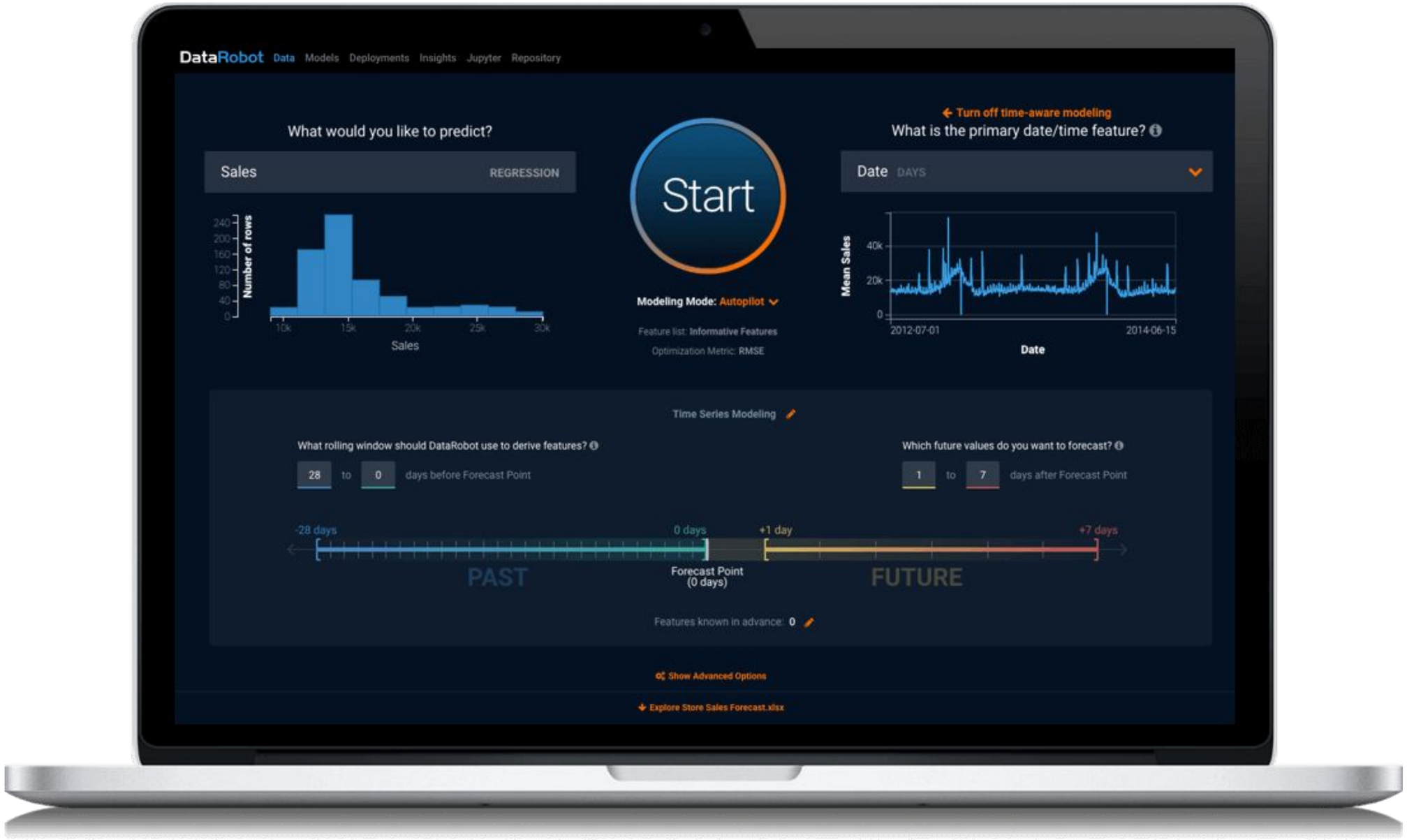
Auto-Sklearn

```
import autosklearn.classification as automl
classifier = automl.AutoSklearnClassifier()
classifier.fit(X_train, y_train)
predictions = classifier.predict(X_test)
```

Auto ML: GUI example (1/2)



Auto ML: GUI example (2/2)




THE #1 DATA SCIENTIST EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY MODEL'S TRAINING"
HEY! GET BACK TO WORK!
TRAINING!
OH. CARRY ON.

Case Study: Google Cloud Platform (GCP), AI, Tables

Google AI

Google Cloud Platform | Numerai - AutoML Tables | Search products and resources

Home
Storage
Compute Engine
Tables

PRODUCTS

ARTIFICIAL INTELLIGENCE
AI Platform
Data Labeling
Natural Language
Recommendations AI
Tables
Talent Solution
Translation
Vision
Video Intelligence

OTHER GOOGLE SOLUTIONS
Game Servers
Google Maps

OMMENDATIONS | CUSTOMIZE

ML Tables
017
O THIS PROJECT
ettings
e
rom the past 7 days
Trace
rting
errors. Have you set up Error Reporting?

Compute Engine
CPU (%)
No data is available for the selected time frame.
1.0 0.8 0.6 0.4 0.2 0
12 PM 12:15 12:30 12:45
Go to Compute Engine

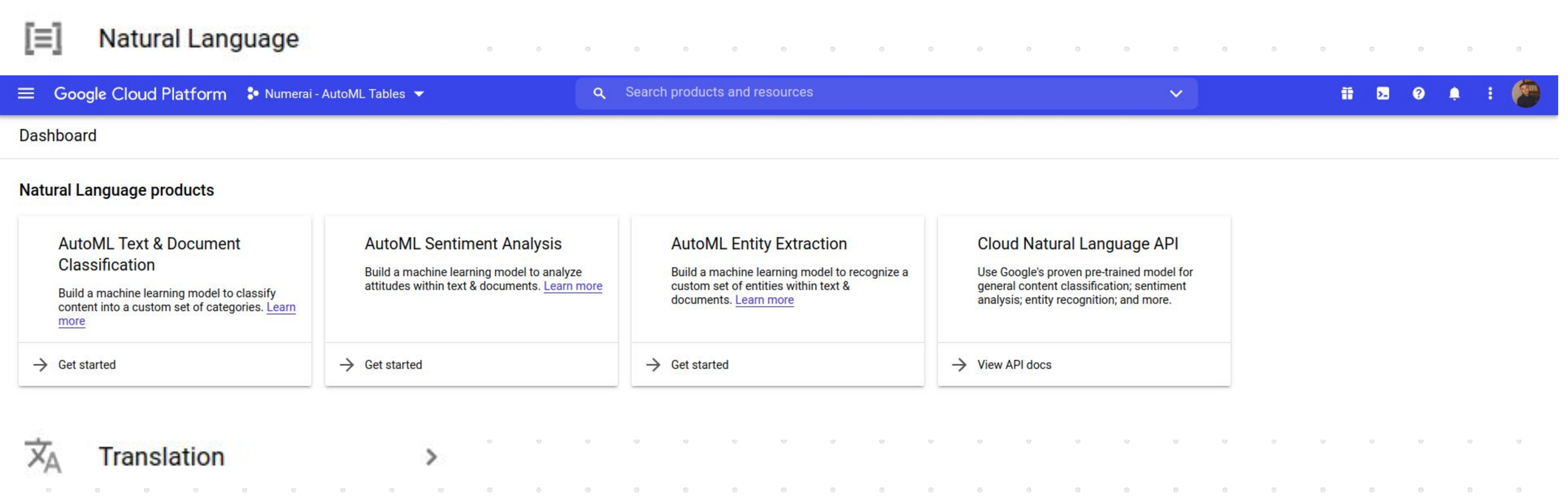
APIs
Requests (requests/sec)
0.5/s 0.4/s 0.3/s 0.2/s 0.1/s 0
12 PM 12:15 12:30 12:45
Requests: 0.150/s

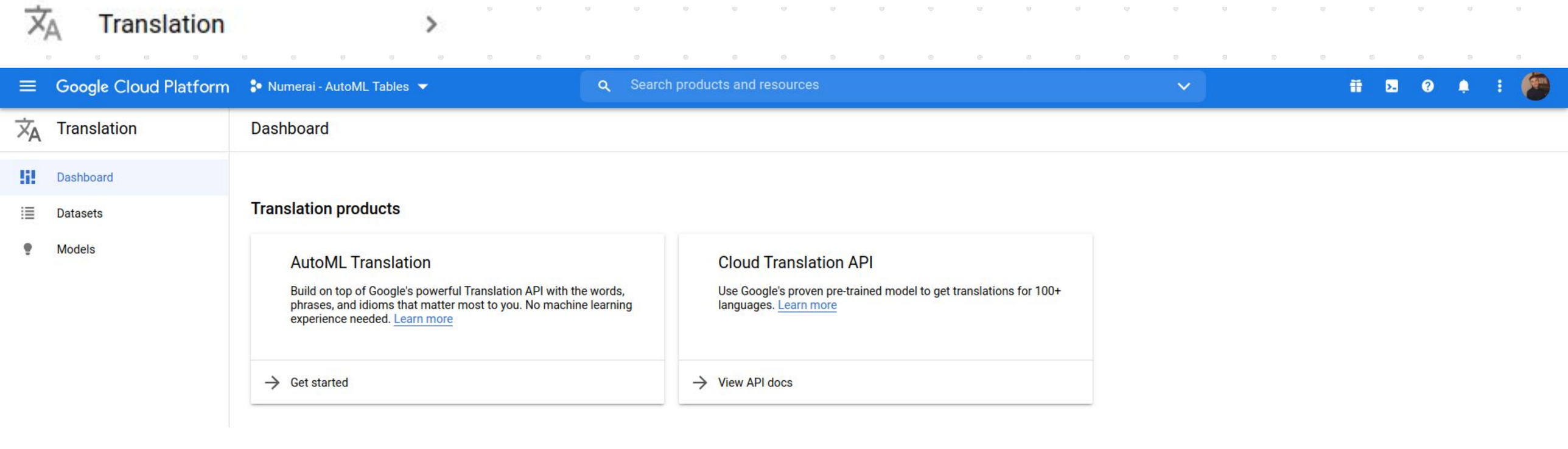
Google Cloud Platform status
All services normal
Go to Cloud status dashboard

Billing
Estimated charges EUR €0.00
For the billing period Sep 1 – 29, 2020
View detailed charges

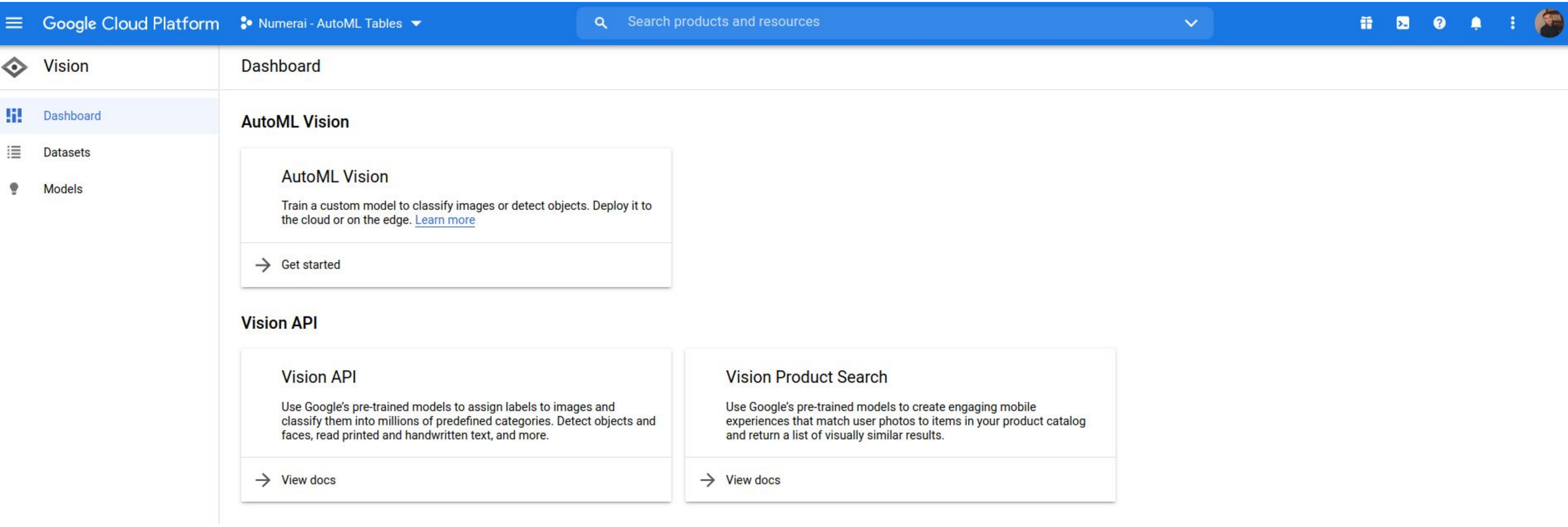
Monitoring
Set up alerting policies
Create uptime checks
View all dashboards
Go to Monitoring

Google's AutoML


Natural Language
Google Cloud Platform
Numerai - AutoML Tables
Search products and resources
Dashboard
Natural Language products
AutoML Text & Document Classification
Build a machine learning model to classify content into a custom set of categories. Learn more
Get started
AutoML Sentiment Analysis
Build a machine learning model to analyze attitudes within text & documents. Learn more
Get started
AutoML Entity Extraction
Build a machine learning model to recognize a custom set of entities within text & documents. Learn more
Get started
Cloud Natural Language API
Use Google's proven pre-trained model for general content classification; sentiment analysis; entity recognition; and more.
View API docs


Translation
Google Cloud Platform
Numerai - AutoML Tables
Search products and resources
Translation
Dashboard
Datasets
Models
Dashboard
Translation products
AutoML Translation
Build on top of Google's powerful Translation API with the words, phrases, and idioms that matter most to you. No machine learning experience needed. Learn more
Get started
Cloud Translation API
Use Google's proven pre-trained model to get translations for 100+ languages. Learn more
View API docs

Vision


Google Cloud Platform
Numerai - AutoML Tables
Search products and resources
Vision
Dashboard
Datasets
Models
Dashboard
AutoML Vision
AutoML Vision
Train a custom model to classify images or detect objects. Deploy it to the cloud or on the edge. Learn more
Get started
Vision API
Vision API
Use Google's pre-trained models to assign labels to images and classify them into millions of predefined categories. Detect objects and faces, read printed and handwritten text, and more.
View docs
Vision Product Search
Use Google's pre-trained models to create engaging mobile experiences that match user photos to items in your product catalog and return a list of visually similar results.
View docs

Video Intelligence

Google Cloud Platform | Numerai - AutoML Tables | Search products and resources

Dashboard

AutoML Video

AutoML Video Classification BETA

Train a custom model to classify shots and segments in your videos, no machine learning experience required. Learn more

→ Get started

AutoML Video Object Tracking BETA

Train a custom model using the API to detect and track objects in your videos with bounding boxes and labels. Deploy it to the cloud, then view prediction results in the Cloud Console.

→ Get started

Video API

Cloud Video Intelligence

Use Google's pre-trained models to detect over 20,000 entities and actions in your videos. You can also detect explicit content and shots, transcribe speech, recognize text, and more.

→ View API docs

Case Study: The Numerai dataset

NUMERAI

Numerai *is an AI-run, crowd-sourced hedge fund based in San Francisco.*

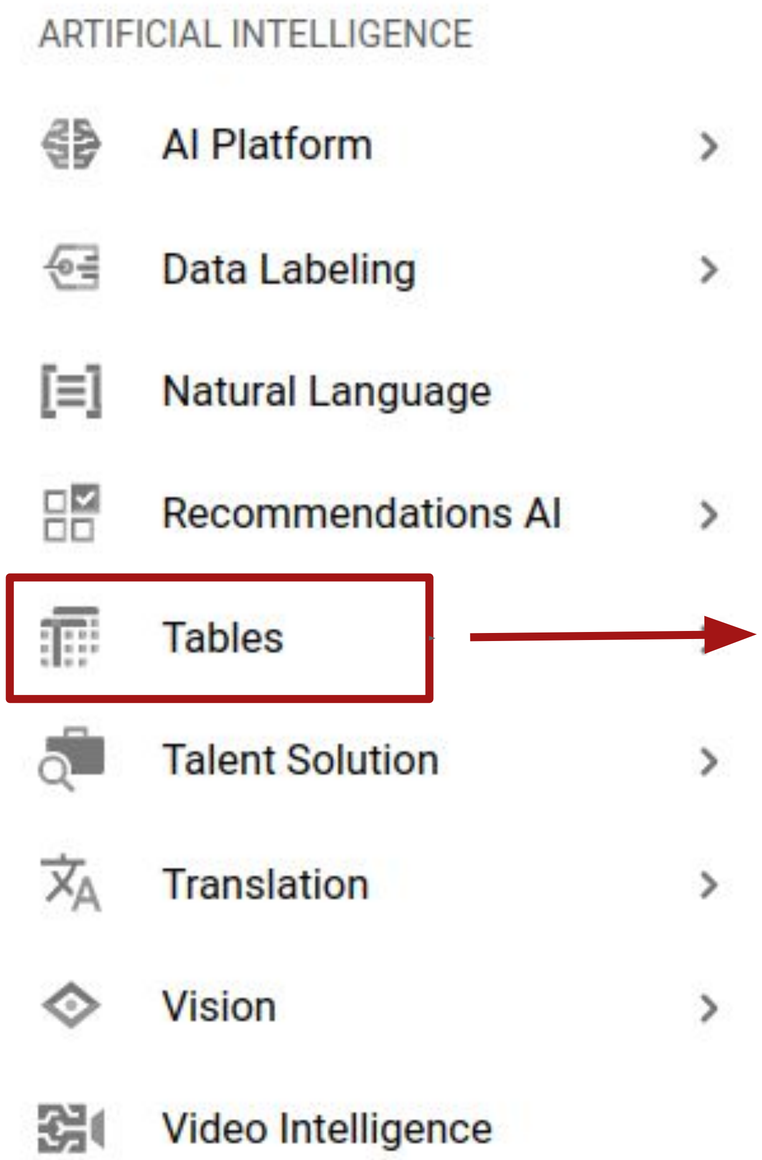
- Numerai's trades are determined by an AI, which is fueled by a network of thousands of anonymous data scientists
- Numerai hosts a weekly tournament, in which data scientists submit their predictions in exchange for the potential to earn some amount of USD and cryptocurrency called Numeraire.
- Predictions submitted by users steer Numerai's hedge fund together.

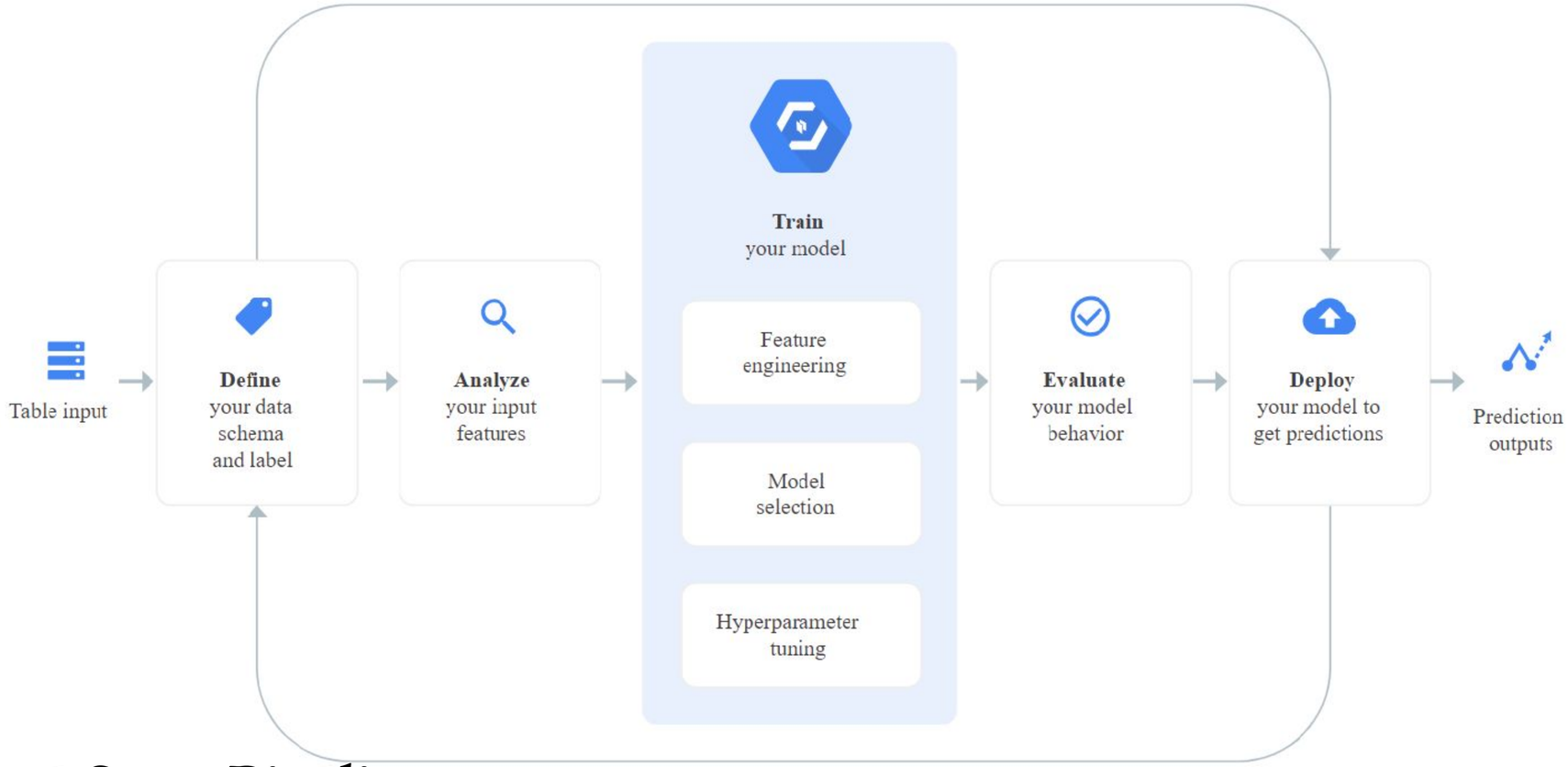
You could learn more here: https://numer.ai/learn

Case Study: GCP Tables (zero-code)

Google AI

Google's AutoML

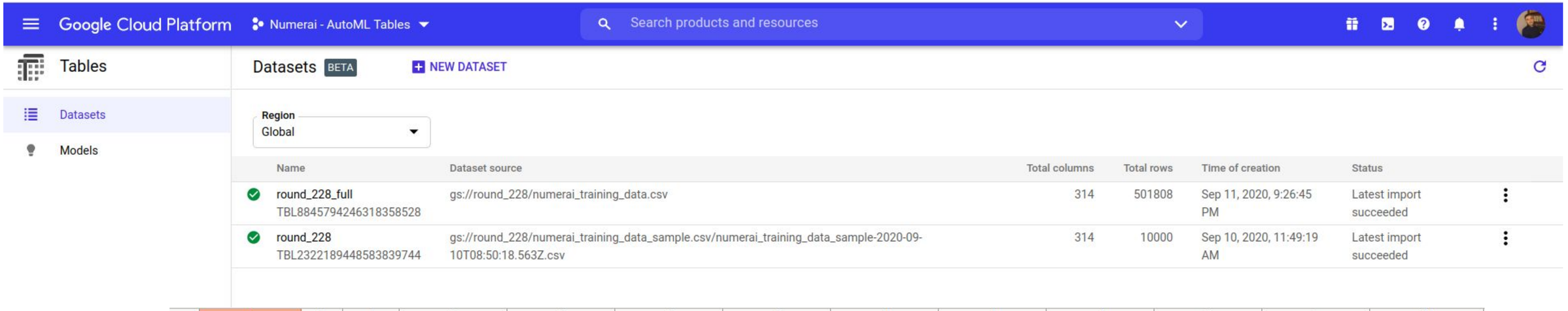




5 Steps Pipeline:

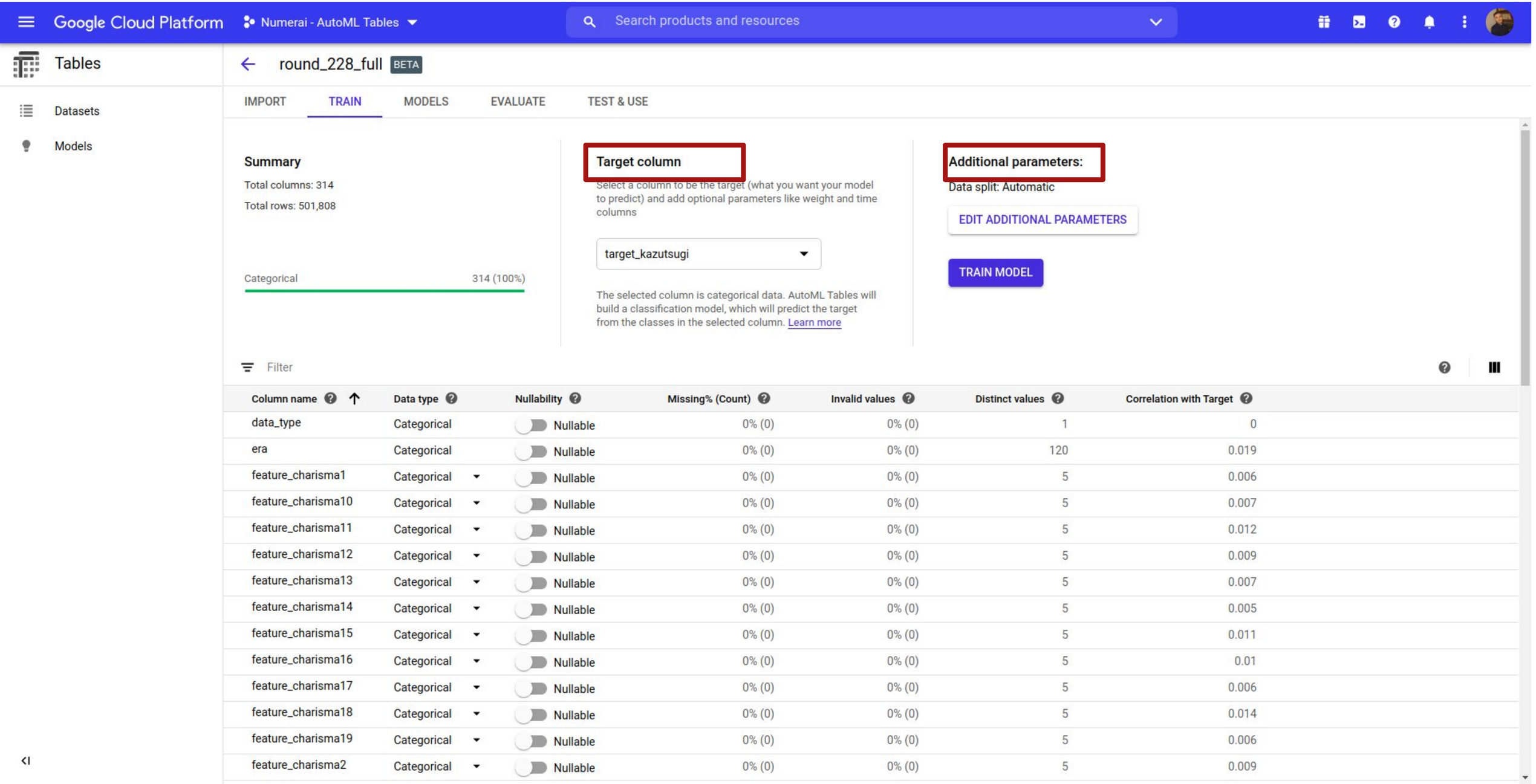
1. Import, 2. Train, 3. Models, 4. Evaluate, 5. Test & Use

Step 1: Import dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	era	data_type	feature_intelligence1	feature_intelligence2	feature_intelligence3	feature_intelligence4	feature_intelligence5	feature_intelligence6	feature_intelligence7	feature_intelligence8	feature_intelligence9	feature_intelligence10
2	n000315175b67977	era1	train	0	0.5	0.25	0	0.5	0.25	0.25	0.25	0.75	0.75
3	n0014af834a96cdd	era1	train	0	0	0	0.25	0.5	0	0	0.25	0.5	0.5
4	n001c93979ac41d4	era1	train	0.25	0.5	0.25	0.25	1	0.75	0.75	0.25	0	0.25
5	n0034e4143f22a13	era1	train	1	0	0	0.5	0.5	0.25	0.25	0.75	0.25	0.5
6	n00679d1a636062f	era1	train	0.25	0.25	0.25	0.25	0	0.25	0.5	0.25	0.25	0.5
7	n009aa2d32389eca	era1	train	0.5	0.5	0.25	0.25	0.75	0.75	0.75	0.5	0.25	0.5
8	n009ef1a5fe009b6	era1	train	0.5	0.25	0.25	0.75	1	1	1	0.75	0	0
9	n00ae5d51f55fb0f	era1	train	0.25	1	1	0.75	1	0.75	0.75	0.75	1	0.75
10	n00b0ac86d77aed7	era1	train	0.5	0.5	0.5	1	1	0.25	0.5	1	0.5	0.25
11	n00c63366aeaf76a	era1	train	0.5	1	1	0.25	0.75	0.25	0.25	0.75	0.5	0.5
12	n00c7211e58e4f4f	era1	train	0.5	0.5	0.5	0.25	0.75	0	0.5	0.5	1	1
13	n00d3e726a8f6e8d	era1	train	0.5	0.25	0.25	1	1	0.5	0.25	0.75	0.25	0.25
14	n00d8f184e5b24e8	era1	train	1	0	0	1	0.25	0.5	0.75	0.75	0.5	0.5
15	n00fde07e85a2898	era1	train	1	1	1	0.75	0.75	0.25	0.25	1	0.5	0.25
16	n011ecdf8e5491eb	era1	train	0.5	0.25	0.25	0.5	0.75	0.5	0.5	0.5	0.5	0.5
17	n0147510f8edb02e	era1	train	0.75	1	1	0.5	1	1	1	1	0.5	0.25
18	n014df3323bd4fb3	era1	train	0.25	0.25	0.25	0.5	0.25	0.25	0	0.25	0.75	1
19	n015d0492cd74e6b	era1	train	0.25	0.25	0.5	0.5	0.75	0.5	0.25	0	0	0.25
20	n0178f759dabe887	era1	train	0.75	0.75	0.75	1	0.5	0	0	1	0.5	0
21	n0191b1321d537a3	era1	train	0	0.5	0.75	0	1	0.5	0	0.5	0.25	0.25
22	n0198f7fdcdd0e2d	era1	train	0.75	1	1	1	0.5	0.75	1	0.25	0.75	1
23	n01deaabd87ca24a	era1	train	1	0	0	0.75	0	1	1	0.75	0.25	0.25
24	n020352f76d47ec3	era1	train	0.5	0.75	0.75	0.5	1	1	1	0.25	0.75	0.75
25	n020a02f7c902350	era1	train	0	1	1	0	0.25	0.5	0.25	0	1	1
26	n02140672cb2163f	era1	train	0.75	0.75	0.75	0.75	0.75	0	0	0.5	0.75	0.75
27	n021a6969fe1d565	era1	train	0.5	0.75	0.75	1	0.5	0	0	1	1	0.5
28	n0229492063ceac5	era1	train	0.75	0	0	0.5	0.25	0.5	0.25	0.5	0.75	0.75
29	n024741d3748f888	era1	train	0	0.75	0.75	0	0.5	0	0	0	0.5	0.75
30	n02535f41af5296b	era1	train	1	1	1	1	1	1	1	0.75	1	1
31	n0260a327dbf36b1	era1	train	0.25	0	0	0.75	0.75	0.5	0.5	0.75	0.5	0.25
32	n026ab0b8fabc480	era1	train	0.5	0.75	0.75	0	0.5	0	0	0.25	1	1
33	n028a235fc492eeb	era1	train	0.25	0	0	0.5	0.25	0	0	0.5	0.75	1
34	n02a3913d0b15efc	era1	train	0.75	1	1	0.25	0.75	0.75	1	0.5	1	0.75
35	n02ad35bb1903b21	era1	train	0.75	0.5	0.5	0.75	0	1	1	0.5	0.25	0.25
36	n02c9146bba8182c	era1	train	0.75	0	0	0.75	0.25	0.5	0.5	0.75	0	0
37	n03096b052026ca4	era1	train	1	0.25	0.25	0	0	1	1	0	0.5	0.75
38	n0311167dd5d7e64	era1	train	1	0.75	0.75	0.25	0.75	0.5	0.25	0.25	0	0.5

Step 2: Train 1/2 (Analyze dataset, Define target)



Column name	Data type	Nullability	Missing% (Count)	Invalid values	Distinct values	Correlation with Target
data_type	Categorical	Nullable	0% (0)	0% (0)	1	0
era	Categorical	Nullable	0% (0)	0% (0)	120	0.019
feature_charisma1	Categorical	Nullable	0% (0)	0% (0)	5	0.006
feature_charisma10	Categorical	Nullable	0% (0)	0% (0)	5	0.007
feature_charisma11	Categorical	Nullable	0% (0)	0% (0)	5	0.012
feature_charisma12	Categorical	Nullable	0% (0)	0% (0)	5	0.009
feature_charisma13	Categorical	Nullable	0% (0)	0% (0)	5	0.007
feature_charisma14	Categorical	Nullable	0% (0)	0% (0)	5	0.005
feature_charisma15	Categorical	Nullable	0% (0)	0% (0)	5	0.011
feature_charisma16	Categorical	Nullable	0% (0)	0% (0)	5	0.01
feature_charisma17	Categorical	Nullable	0% (0)	0% (0)	5	0.006
feature_charisma18	Categorical	Nullable	0% (0)	0% (0)	5	0.014
feature_charisma19	Categorical	Nullable	0% (0)	0% (0)	5	0.006
feature_charisma2	Categorical	Nullable	0% (0)	0% (0)	5	0.009

Step 2: Train 2/2 (Define train parameters)

Additional parameters

Data split

- (•) Automatic (default)
  80% of your data rows are for training, 10% for validation, and 10% for testing
- ( ) Manual

  Test or Train categorical columns not found. Please add one to use customized data split.
- ( ) Time column

Weight column

No columns available

SAVE CANCEL

Train your model

Model name *
round_228_full_20200927022216

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 budget doesn't include setup, preprocessing, and tear down. These steps usually don't exceed one hour total and you won't be charged for that time. 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).

313 feature columns *
All columns selected

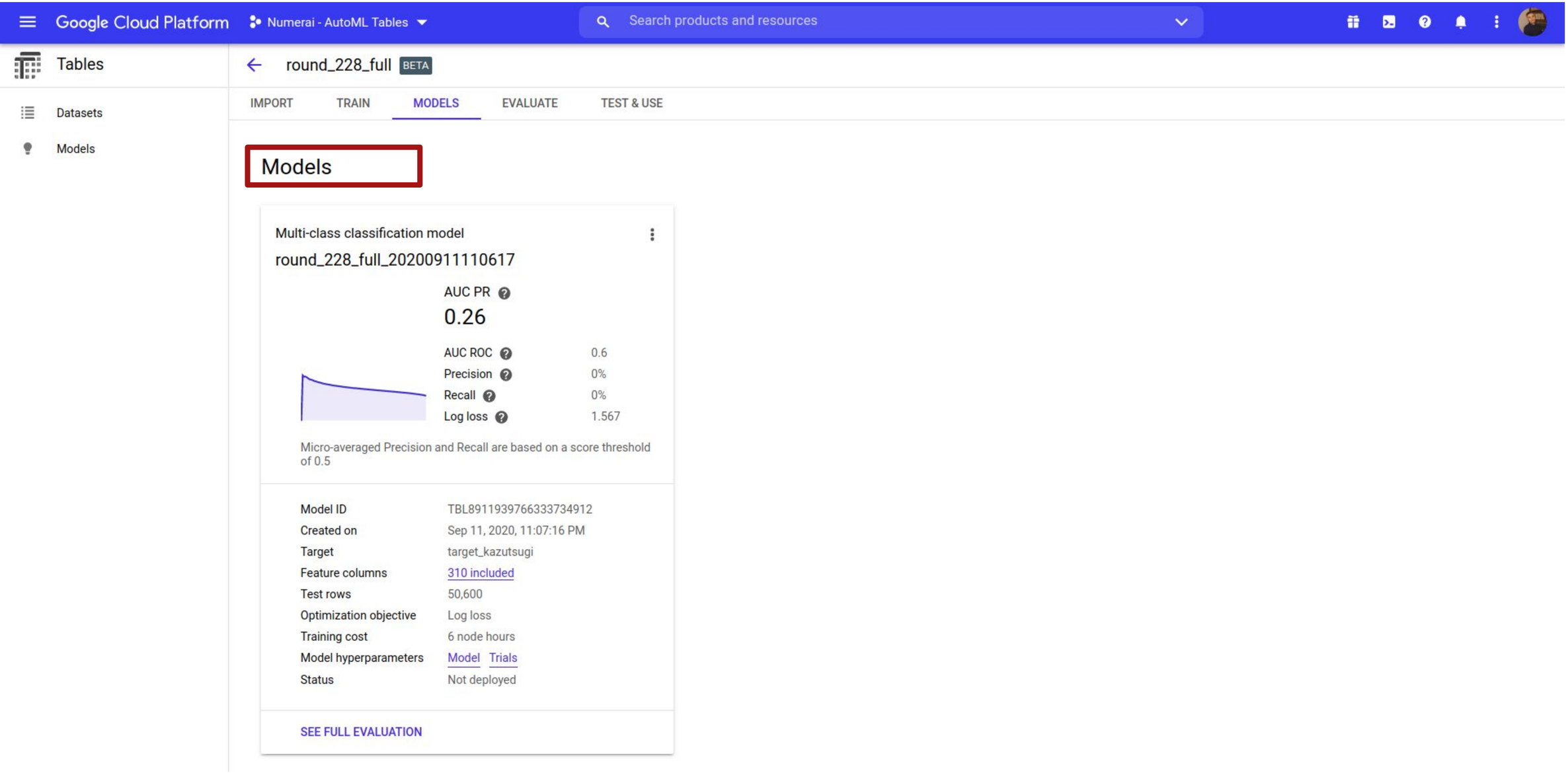
Summary

Model type: Multi-class classification model
Data split: Automatic
Target: target_kazutsugi
Input features: 313 features
Rows: 501,808 rows

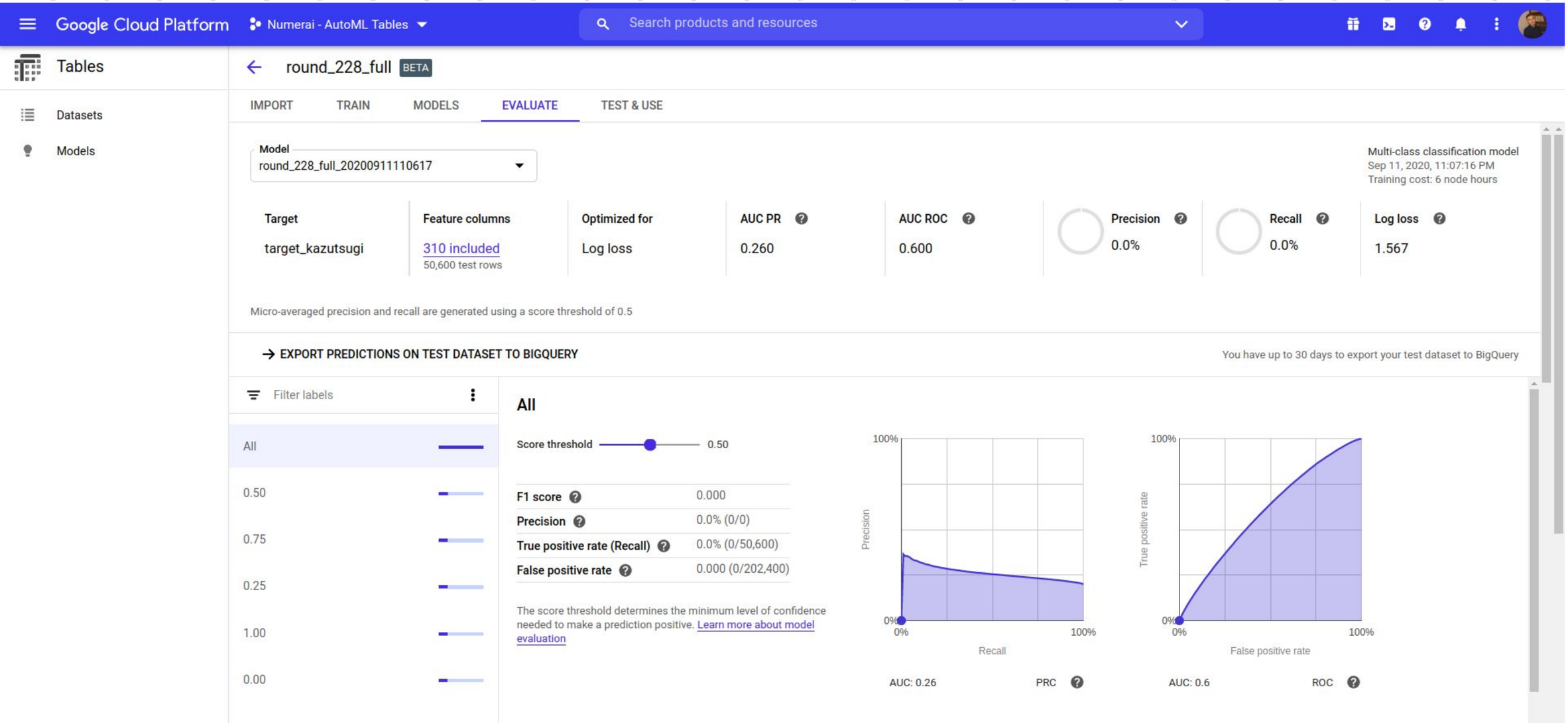
Advanced options

TRAIN MODEL CANCEL

Step 3: Models

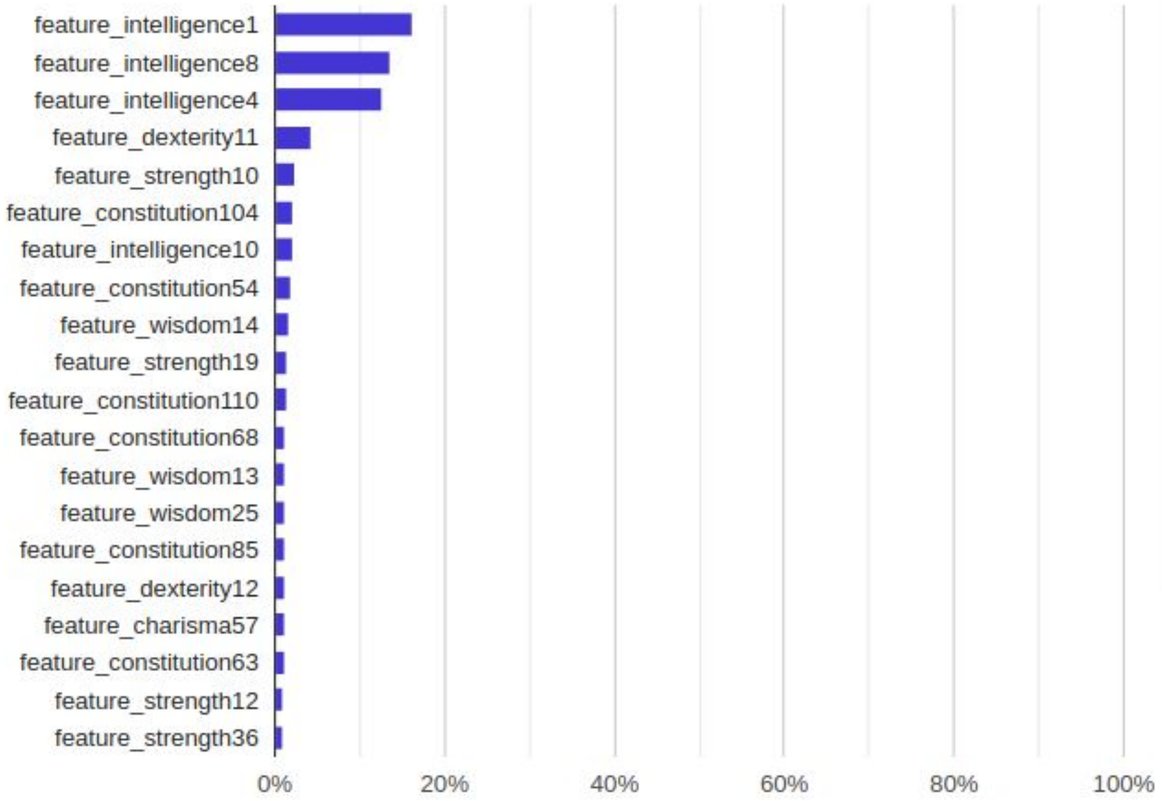


Step 4: Evaluate (1/2)



Step 4: Evaluate (2/2)

Feature importance

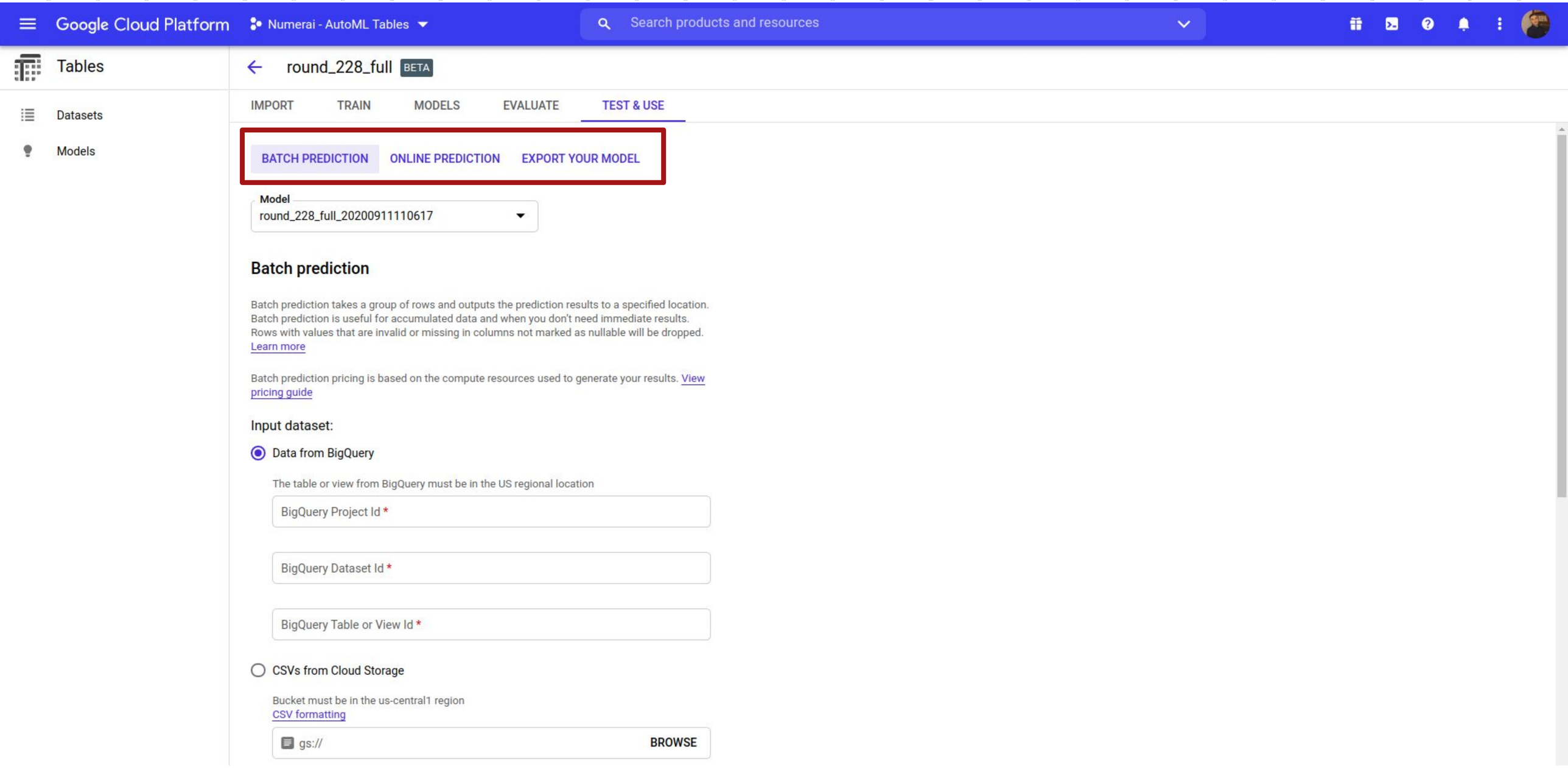


Confusion matrix

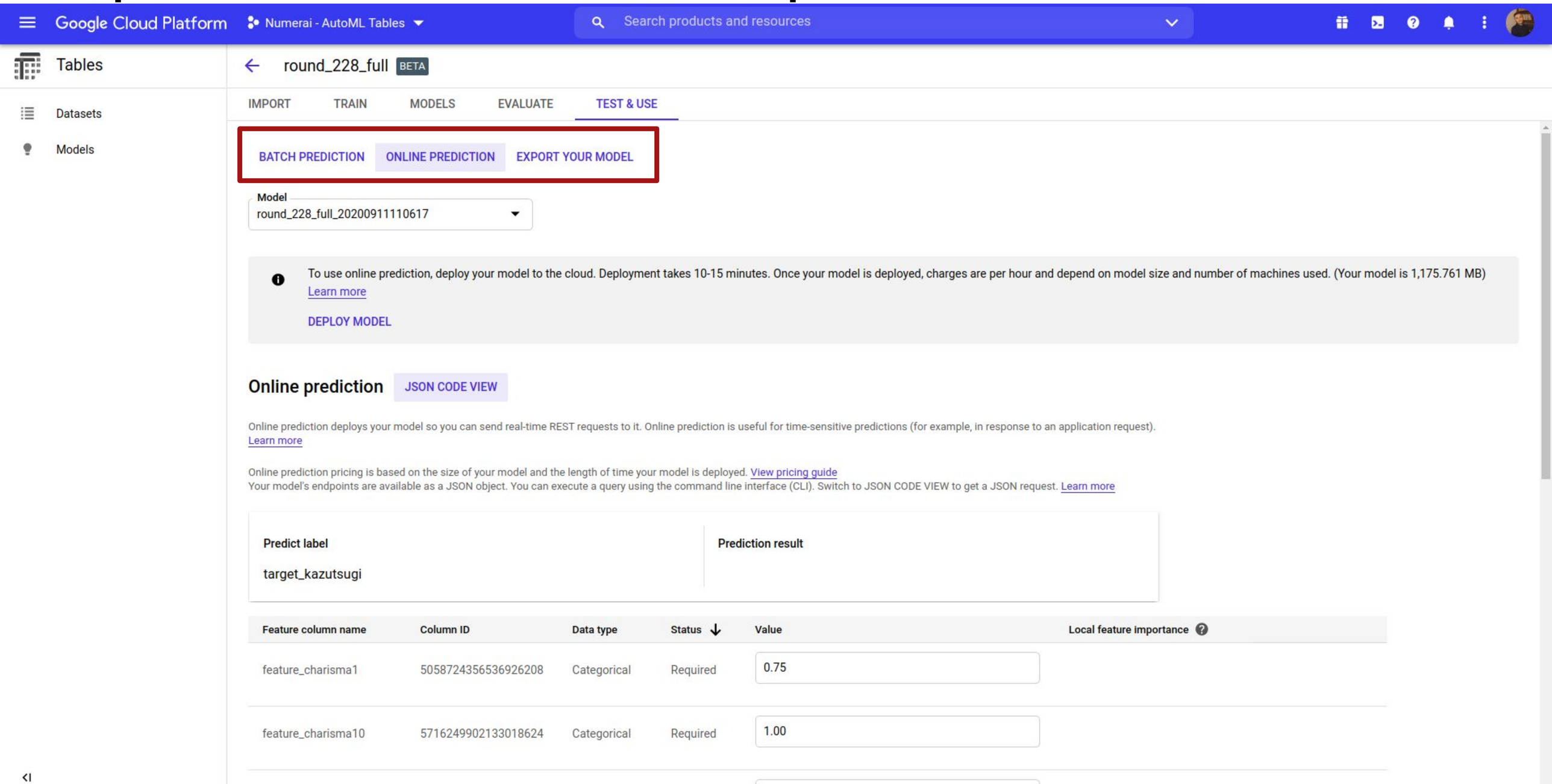
A confusion matrix helps you understand where misclassifications occur (which classes get "confused" with each other). Each row is a predicted class and each column is an observed class. The cells of the table indicate how often each classification prediction coincides with each observed class.

True labels \ Predicted labels	0.00	0.25	0.50	0.75	1.00
0.00	34%	9%	20%	10%	27%
0.25	20%	12%	37%	13%	18%
0.50	18%	12%	40%	13%	17%
0.75	20%	12%	37%	13%	19%
1.00	30%	9%	20%	10%	31%

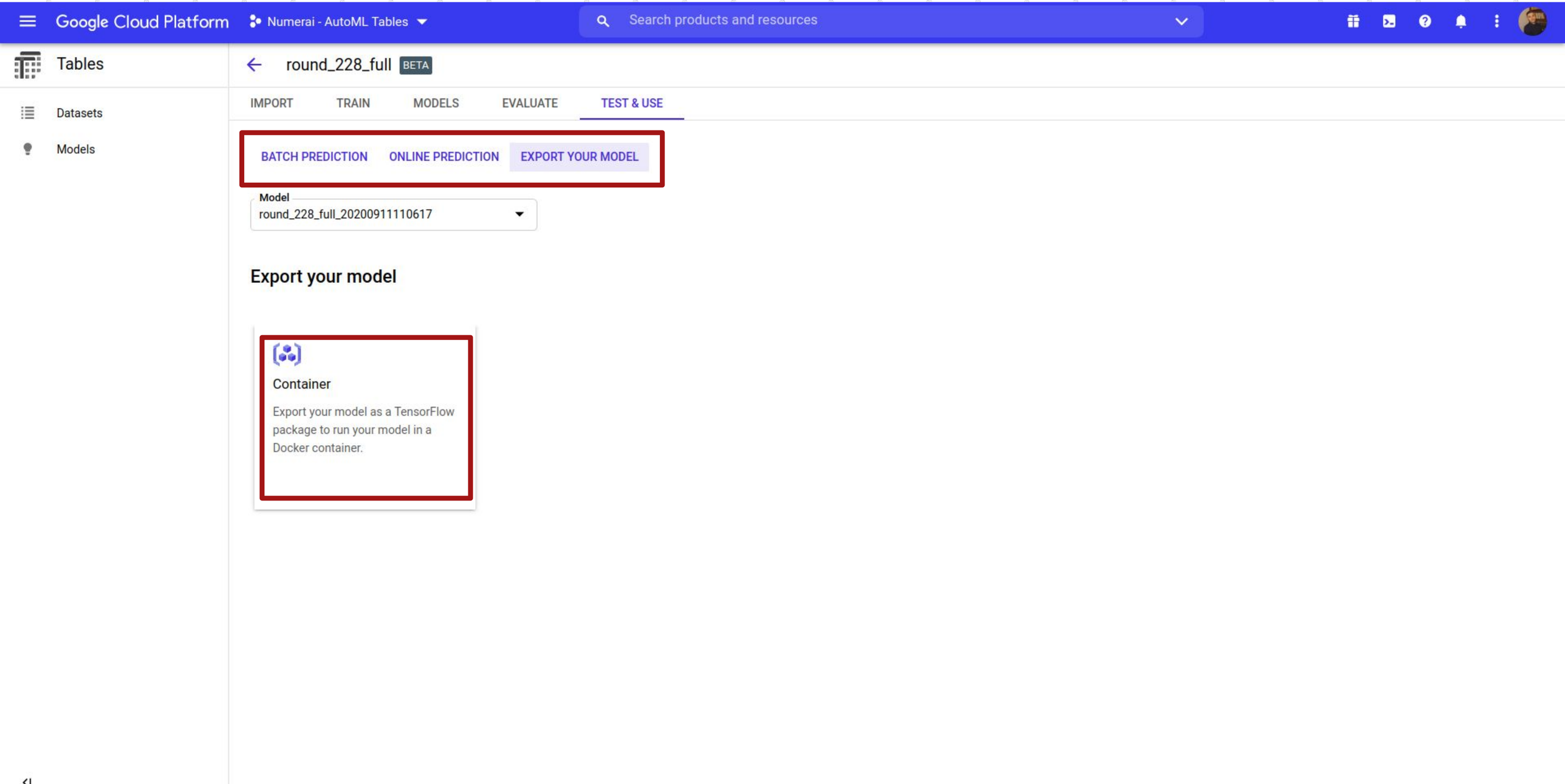
Step 5: Test & Use (Batch prediction)



Step 5: Test & Use (Online prediction)



Step 5: Test & Use (Export your model)



Closing Remarks

Advantages

1. **Fast experimentation.**
   Automatically search an entire space of candidate Machine learning pipelines and returns the best one.
   Shows what approaches are the best for the specific dataset and the specific task.

2. **Handles the boring stuff.**
   (hyperparameter tuning etc.) and helps the data scientist to focus on important things (business problem modeling)

3. **Democratization of Machine Learning.**
   No need for programming skills and/or understanding of Machine Learning
   (I'm not sure if this is an advantage!)

Disadvantages

1. Not fully automated
2. Not customized solutions
3. Not human competitive (yet)
   Kagglers are still state-of-the-art
4. There aren't any Reinforcement Learning applications.
5. Lack of domain knowledge.
   Poor feature extraction skills.

Conclusion.

~~“Now you can fake your Machine Learning expertise, without even understanding linear regression!”~~

“Use Auto ML tools to Free up time from boring and repetitive tasks, in order to focus in what actually matters.”

GDG

devfest

Hellas 2020

www.linkedin.com/in/panagiotis-papaemmanouil/

panagiotis.papaemmanouil[at]medoid.ai

www.medoid.ai/

www.linkedin.com/company/medoid-ai/

Medoid AI