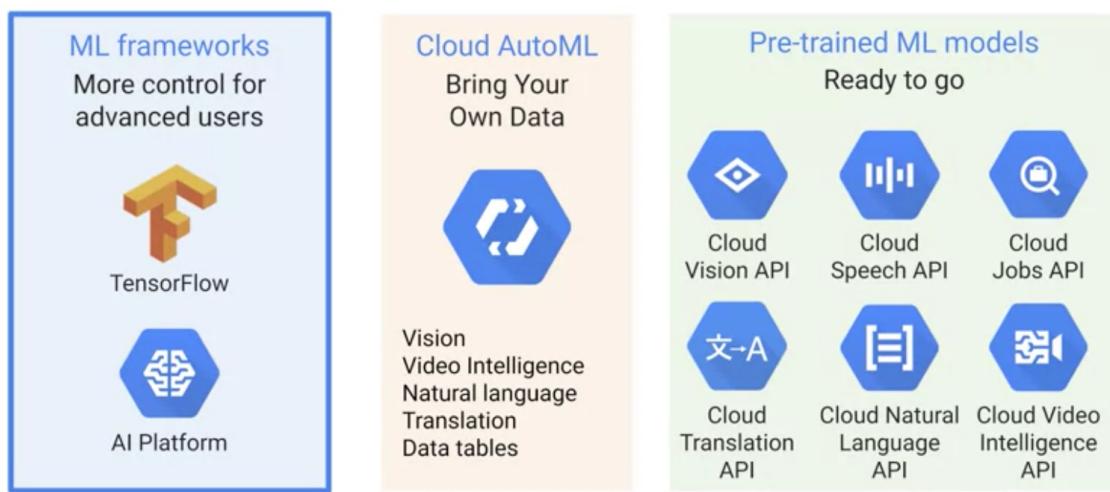
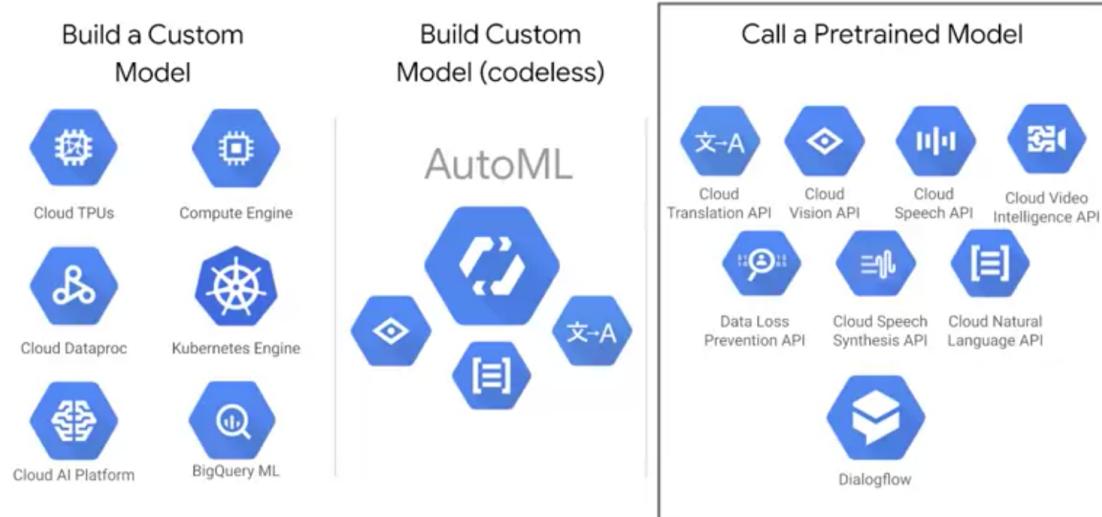


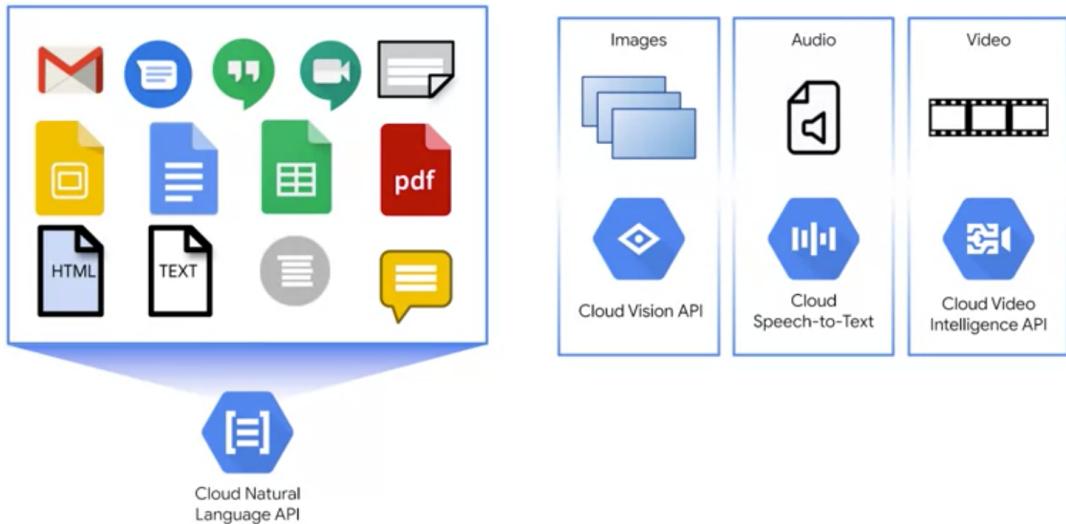
The GCP machine learning spectrum



For common ML tasks, consider Pretrained APIs

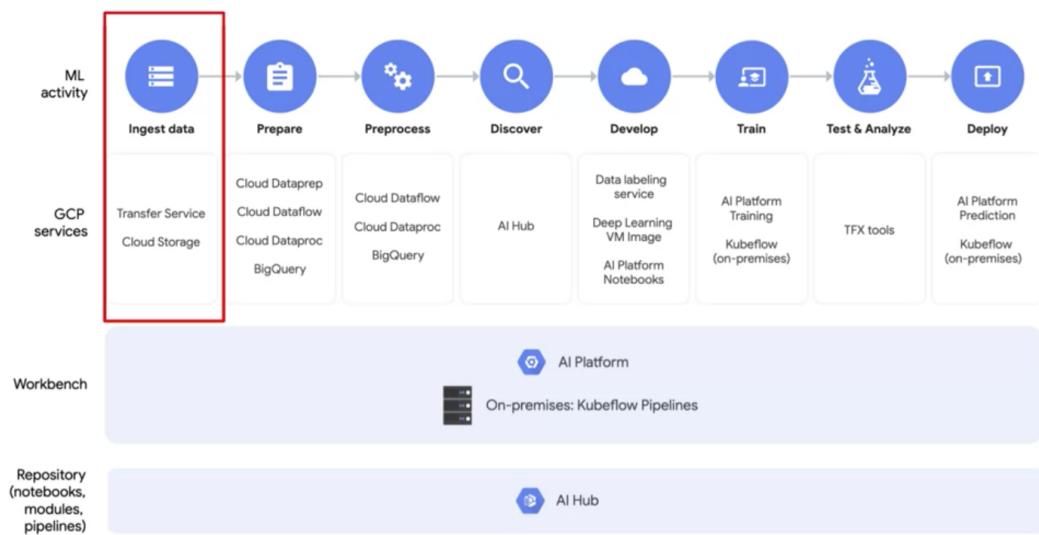


You process unstructured data by labeling it with AI



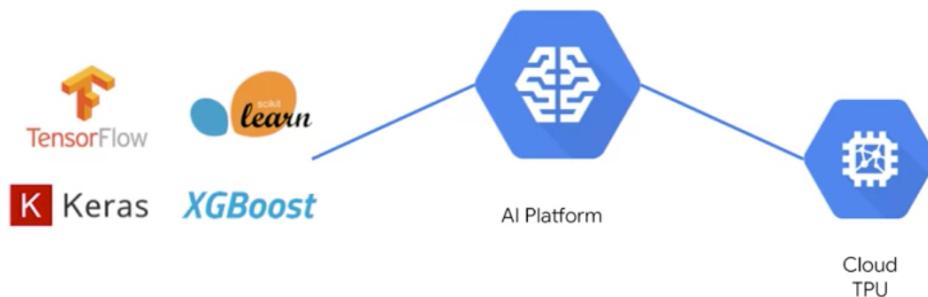
Phases of ML Projects

Bring data into your central Data Lake



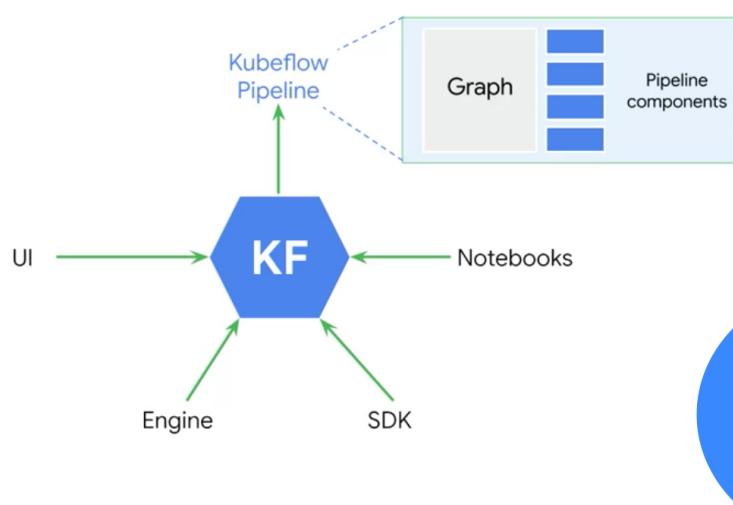
Ways to do custom ML on GCP

Cloud AI Platform is a fully managed service for custom machine learning models



- Scales to production
- Batching and distribution of model training
- Performs transformations on input data
- Hyper-parameter tuning
- Host and autoscale predictions
- Serverless - self-tuning - manages overhead

Kubeflow Pipelines package multi-step ML workflows



BigQuery ML empowers data analysts and scientists

Execute ML initiatives without moving data from BigQuery

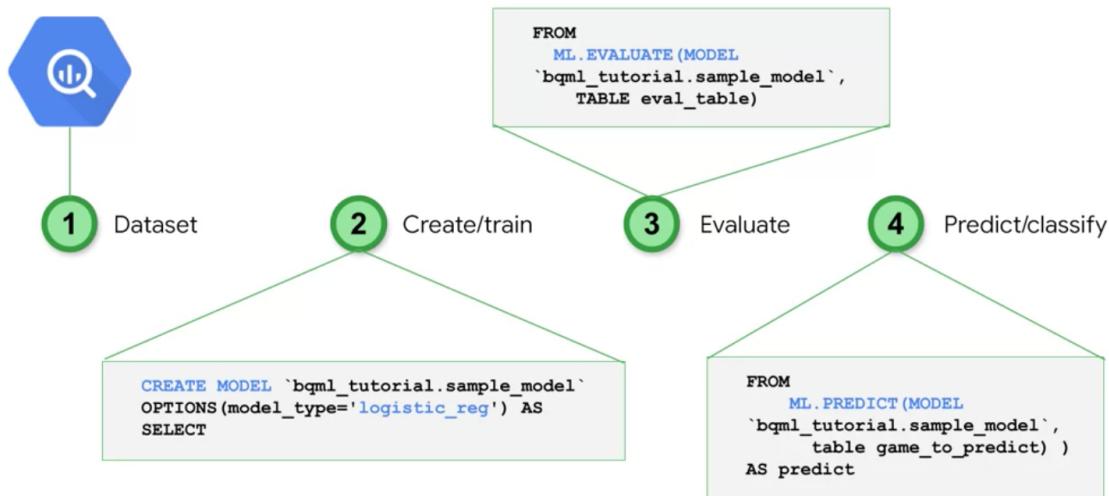


Iterate on models in SQL in BigQuery to increase development speed

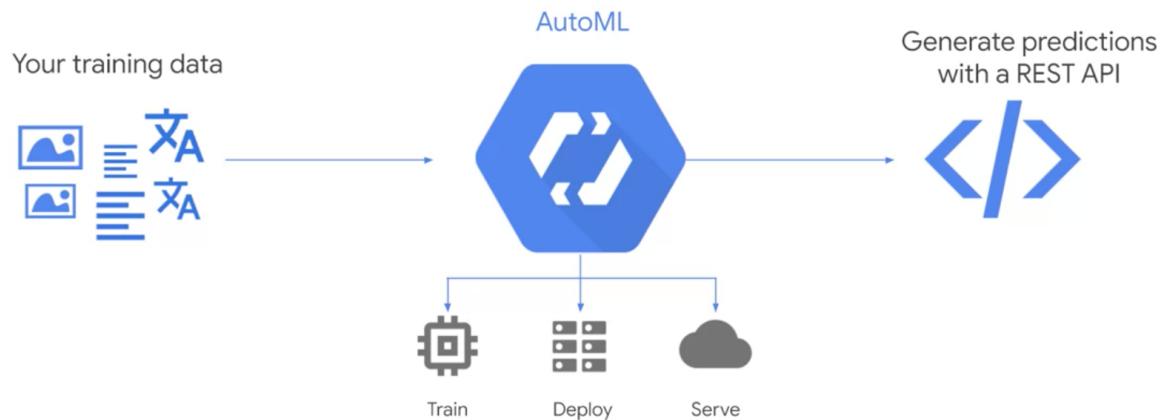
Automate model selection and hypertuning



Working with BigQuery ML

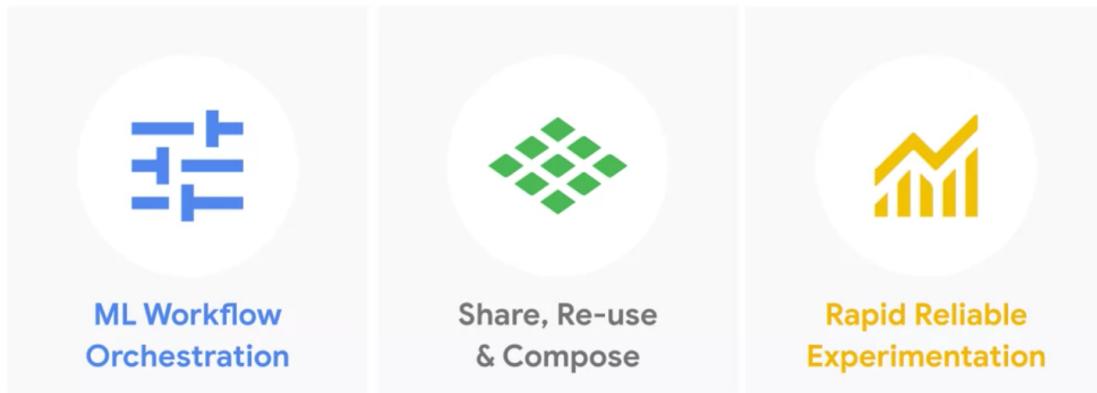


Train and serve custom models without any code
with Cloud AutoML

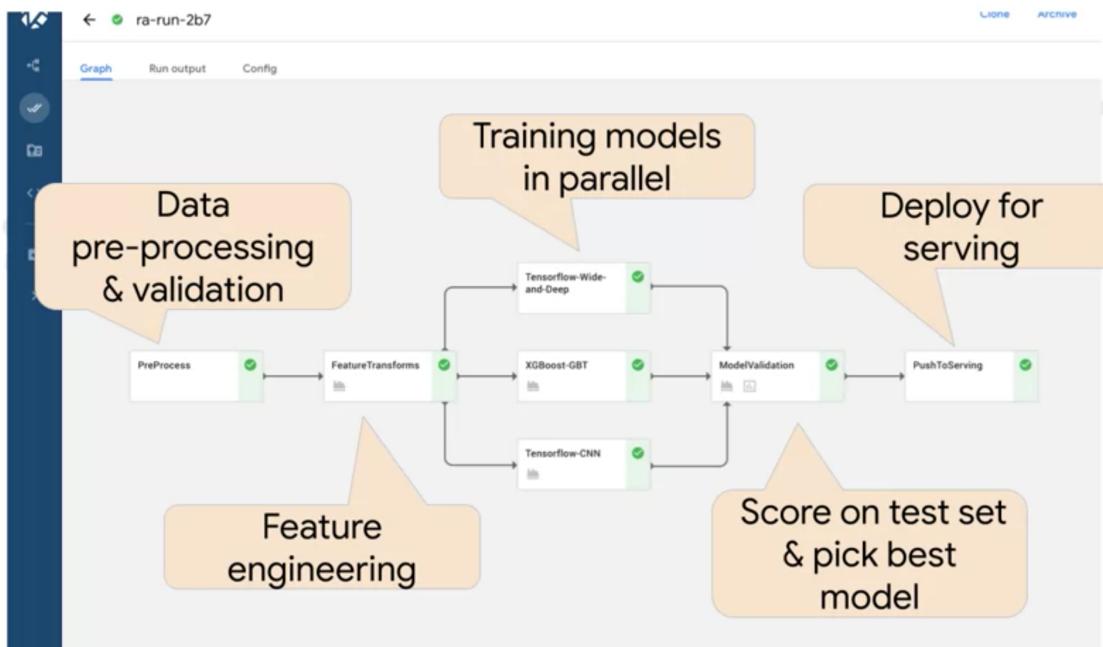


Kubeflow

Kubeflow Pipelines enable:



Visual depiction of pipeline topology



View all configs, inputs and outputs

The screenshot shows the configuration view for a run named "Simple XGBoost Classifier".

Run details:

| Status | Succeeded |
|-------------|-------------------------|
| Description | |
| Created at | 11/25/2018, 12:56:44 PM |
| Started at | 11/25/2018, 12:56:44 PM |
| Finished at | 11/25/2018, 1:16:37 PM |
| Duration | 0:19:53 |

Run parameters:

| | |
|------------|----------------------------------------------|
| output | gs://mlpipelines |
| project | foo2thebar |
| region | us-central1 |
| train-data | gs://ml-pipeline-playground/sfpd/train.csv |
| eval-data | gs://ml-pipeline-playground/sfpd/eval.csv |
| schema | gs://ml-pipeline-playground/sfpd/schema.json |
| target | resolution |
| rounds | 200 |
| workers | 2 |
| true-label | ACTION |

Author pipelines with an intuitive Python SDK

```
In [3]: import kfp.dsl as dsl

@dsl.pipeline(
    name='TFX Taxi Cab Classification Pipeline Example',
    description='Example pipeline that does classification with model analysis based on a public BigQuery dataset.'
)
def taxi_cab_classification(
    output,
    column_names=dsl.PipelineParam(
        name='column-names',
        value='gs://ml-pipeline-playground/tfx/taxi-cab-classification/column-names.json'),
    key_columns=dsl.PipelineParam(name='key-columns', value='trip_start_timestamp'),
    ...
    analyze_slice_column=dsl.PipelineParam(name='analyze-slice-column', value='trip_start_hour')):
    ...

    preprocess = dataflow_tf_transform_op(train, evaluation, schema, project, preprocess_mode, preprocess_module, trans
    training = tf_train_op(preprocess.output, schema, learning_rate, hidden_layer_size, steps, target, preprocess_modul
    analysis = dataflow_tf_model_analyze_op(training.output, evaluation, schema, project, analyze_mode, analyse_slice_c
    prediction = dataflow_tf_predict_op(evaluation, schema, target, training.output, predict_mode, project, prediction_
    deploy = kubeflow_deploy_op(training.output, tf_server_name)
```

Submit the run

```
In [16]: import kfp
from kfp import compiler

compiler.Compiler().compile(taxi_cab_classification, 'tfx.tar.gz')
run = client.run_pipeline(exp_id, 'tfx', 'tfx.tar.gz',
                          params={'output': 'gs://bradley-playground',
                                   'project': 'bradley-playground'})
```

Package & share pipelines as zip files

- Upload and execute pipelines via UI (in addition to API/SDK)
- Pipelines steps can be authored as reusable components

Run details

Pipeline: xgboost training - confusion matrix

Run name: product-recommender-model

Description (optional): Train XBG model for product recommendation application.

Run parameters

Specify parameters required by the pipeline:

| |
|----------------------------------------------|
| output |
| project |
| region |
| train-data |
| gs://ml-pipeline-playground/sfpd/train.csv |
| eval-data |
| gs://ml-pipeline-playground/sfpd/eval.csv |
| schema |
| gs://ml-pipeline-playground/sfpd/schema.json |
| target |
| resolution |
| rounds |
| 200 |
| workers |
| 2 |
| true-label |
| ACTION |

Kubeflow pipelines can be packaged and shared with other users.

AI Hub

AI Hub is a repository for AI assets

- Don't reinvent the wheel! Find and deploy ML pipelines

The screenshot shows the AI Hub web interface. On the left is a sidebar with navigation links: Upload, Home (which is selected), My assets, Scope (Public and Private), Category (Kubeflow pipeline, Notebook, Service, TensorFlow module, VM image, Trained model, Technical guide), and Data type. The main content area has two sections: "TensorFlow modules" and "Kubeflow pipelines".

TensorFlow modules: A banner at the top says "TensorFlow modules" with a subtext: "Modules let you train your models with smaller datasets, train faster, and improve your model's generalization". It includes "Explore modules" and "Learn more" buttons. Below the banner is a decorative illustration of orange 3D blocks and a small figure.

Kubeflow pipelines: This section contains three cards:

- Submitting a SparkSQL Job to Cloud Dataproc** (By Google): A Kubeflow Pipeline component to submit a SparkSQL job to Google Cloud Dataproc service.
- Data preparation by using the General Purpose Preprocessing component** (By Google): A component gives you a standard way of preprocessing datasets. Use it to read datasets and serve raw data serving using a standard process. The output of this component is in the TFRecord format.
- Batch predicting using Cloud Machine Learning Engine** (By Google): A Kubeflow Pipeline component to submit a batch prediction job against a trained model to Cloud ML Engine service.

At the bottom of the main content area, there is a "More modules" link.