Machine Learning using GCP Tools for well trained models and quick Productionization

Diagram owner	@ Seethalakshmi Rengaraman
Status	DRAFT
Last date updated	10 May 2021
On this page	• Goals • Architecture • Deployment strategy • Rest API details • SLA • Future Considerations • Action Items • References and documentation

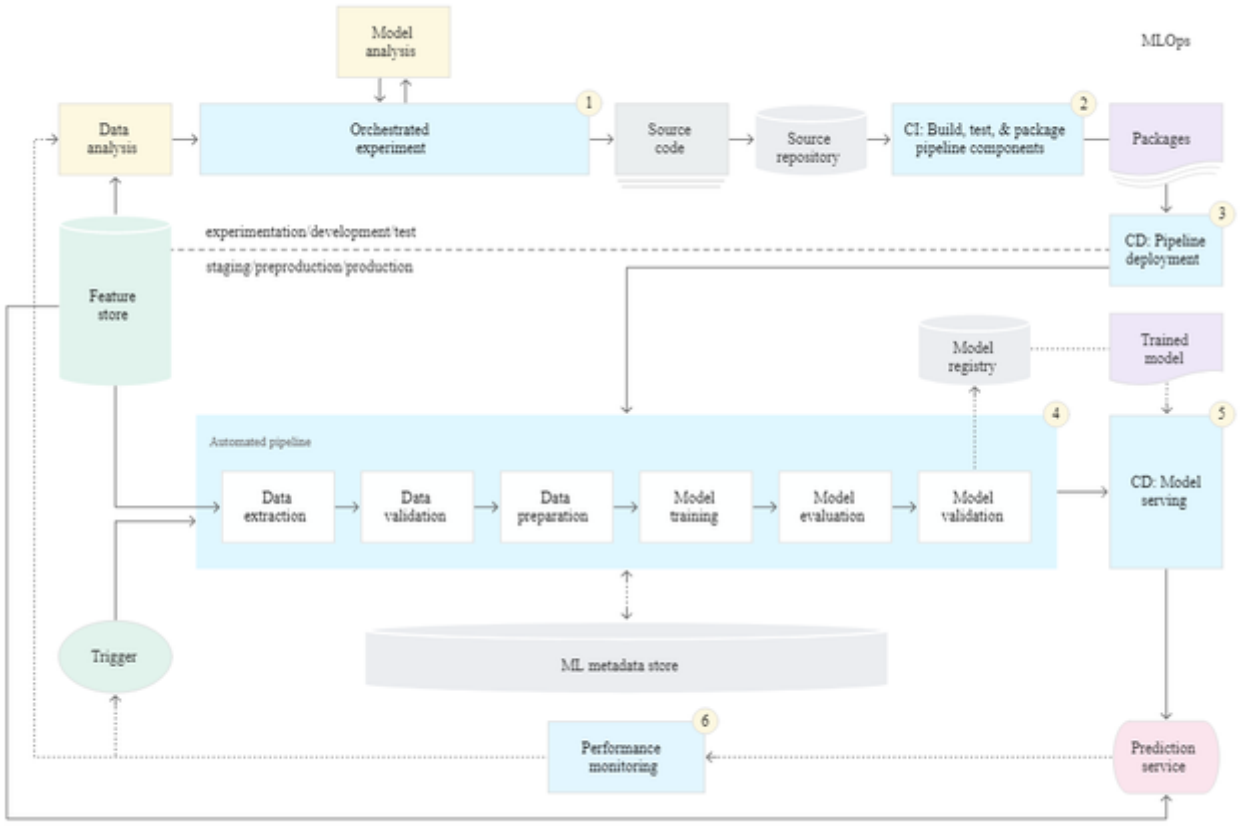
Pillars of a Well-Architected Framework

Name	Description
Operational Excellence	The ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.
Security	The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
Reliability	The ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.
Performance Efficiency	The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve
Cost Optimization	The ability to run systems to deliver business value at the lowest price point.

Goals

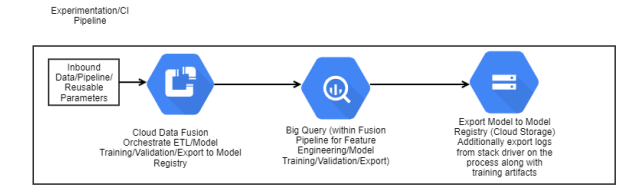
- **Data Engineering** - Ingest data from variety of sources in batch and streaming mode, process/transform, load into a data warehouse and further perform simple/complex transformations (secure ETL/ELT processing). Metadata management and visualization requirements as part of data analysis
- **Machine Learning** - Processed structured data with labels are trained with 95% efficiency with Auto-ML tables (can also experiment with other models) across variety of use cases within Omni, TMS, WM and other Manhattan Products on Cloud
- **Application Deployment** - Deploy trained model in High Availability secure containerized Kubernetes Cluster for serving through rest API with response time SLA of 30 ms per request, 1000 transactions/sec, 99% of the time across multiple customers
- **ML Ops**:
  - Automated CI/CD pipeline logging valuable information across the entire process
  - Develop failsafe mechanism to ensure zero downtime if predicted value couldn't be obtained owing to various conditions in production eg: Bad data
  - Collect feedback, Monitor/retrain models and ensure accuracy at regular time intervals
  - Develop feature stores to use across models
- **Overarching Infrastructure** - Determine necessary memory/processor requirements for Experimentation/Training/deployment phase and production service phase
- **Cost Effectiveness** - With respect to Infrastructure and IT maintenance cost

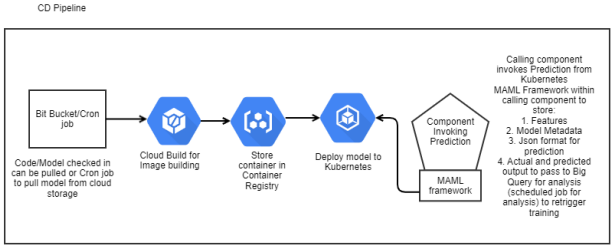
Architecture



Summary Option 1:

Cloud Fusion Pipeline(Data Integration/ETL/ELT/Feature Engineering/Model Training) Export model to Cloud Storage Trigger Cloud Build (Build and push image to Container Registry) Deploy to Kubernetes Cluster Calling component does necessary metadata/transformations (through MAML framework library within it) to invoke prediction from the Kube cluster Calling component stores predicted/actual output for monitoring and retriggering pipeline





Summary Option 2:

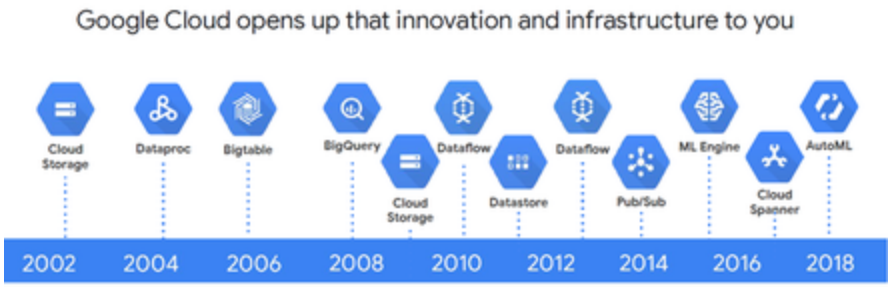
Cloud Fusion Pipeline(Data Integration/ETL/ELT/Feature Engineering/Import Data set and initiate model training in AI platform) Deploy model in AI platform Calling component does necessary metadata/transformations (through MAML framework library within it) to invoke prediction from AI platform AI platform can do continuous evaluation with us providing actual result or Calling component stores predicted/actual output for monitoring and retriggering pipeline

ML Ops CI/CD Pipeline Components	Option 1 Architecture	Option 2 Architecture	Deliverable at the end of the phase
Development and Experimentation	Cloud Fusion - For Data Integration/ETL/ELT /Initiating Feature Engineering, Model Training and export through Big Query Big Query - For Feature Engineering Big Query/AI Platform (Unified) - For Model Training and Evaluation (Model statistics) Export model to Cloud Storage to be containerized and deployed to Kubernetes Cluster	Cloud Fusion - For Data Integration /ETL/ELT/Initiating Feature Engineering through Big Query, Model Training and export through AI platform Big Query - For Feature Engineering AI Platform (Unified) - For Model Training and Evaluation What if Analysis, Feature Importance available for AI Platform deployment models Create Endpoint, deploy trained model within AI Platform (Unified)	Pipeline artifact along with required arguments for a reusable pipeline across models for ETL and Model Training and Evaluation stored in Cloud Storage Model exported to Cloud Storage (Option 1) Advantage with either option: **A single pipeline for ETL, Feature Engineering, Model Training and export using Cloud Data Fusion**
Pipeline continuous integration	Test Pipeline on Production data (integrated) Adjust Features/parameters, load test, add appropriate logs for monitoring and finalize infrastructure	Test Pipeline on Production data (integrated) Adjust Features/parameters, load test, add appropriate logs for monitoring and finalize infrastructure	Final Pipeline artifact, model, parameters, scripts stored in cloud storage Metadata for Model along with its version, model statistics, linked pipeline, features stored in framework library (present across all components requesting for online prediction) For eg: If component-order is requesting delivery date prediction, it will have the MAML framework library consisting of metadata for Delivery Date - model, pipeline, features, json transformation of features for prediction
Pipeline continuous delivery with Automated triggering	In Cloud Fusion, pipeline can be scheduled to run on Weekly/monthly basis for model re-training, evaluation. For cost effectiveness - Cloud Fusion can be spun up based on a cron job using Cloud Scheduler, pipeline imported and executed and Cloud fusion can be stopped post execution	In Cloud Fusion, pipeline can be scheduled to run on Weekly/monthly basis for collecting data, perform feature engineering before importing to AI platform to trigger re-training. For cost effectiveness - Cloud Fusion can be spun up based on a cron job using Cloud Scheduler, pipeline imported and executed and Cloud fusion can be stopped post execution	Installed CI/CD pipeline in production with automatic triggering of the pipeline periodically for new data collection/ETL and model re-training
Model continuous delivery	If model is evaluated to be better than previous version through model stats, one of the following deployment methods used: 1. Model checked into Bit Bucket/Git can trigger Cloud Build and deploy to GKE 2. Model exported to Cloud Storage and a cron job running on master node in GKE cluster can copy model to local, trigger cloud build and deploy to GKE (Traffic can be split across previous model version and new version) Calling component to have configuration if needed to be turned on for few customers vs not turned on for others	If model is evaluated to be better than previous version through model stats, one of the following deployment methods used: 1. Cloud Fusion Pipeline is used to wait until model training is complete to evaluate and deploy to AI platform endpoint 2. Once training is complete, the cron job used to initiate Cloud fusion can evaluate model and initiate deployment to endpoint (Traffic can be split across previous model version and new version) Calling component to have configuration if needed to be turned on for few customers vs not turned on for others	Model Prediction service is setup

Model Monitoring	1. Predicted result stored in transactional entity and actual result is subsequently obtained after event completion 2. Periodically import sample of predicted output and actual output into Big Query along with feature data to determine anomalies. If no anomalies/stats are good, no re-training, else trigger re-training 3. Model performance and logs also to be monitored and alerts raised (framework similar to other apps existing in Active)	AI Platform has option to do continuous evaluation on sample of data predicted (again requiring us to provide the actual output) and storing in Big Query. Based on evaluation metric, can re-trigger training Model performance and logs also to be monitored and alerts raised (framework similar to other apps existing in Active)	Model monitoring automation script /job to review prediction results, determine anomalies and trigger retraining
Optional Feature Store	Setup jobs for fetching and storing aggregated features to be used in prediction within MAML library (within calling component) No-sql store (Big Table/Firestore) necessity may arise if feature engineering done during training is complex to reduce performance bottleneck during prediction	Setup jobs for fetching and storing aggregated features to be used in prediction within MAML library (within calling component) No-sql store (Big Table/Firestore) necessity may arise if feature engineering done during training is complex to reduce performance bottleneck during prediction	Aggregated Features/calculated features stored in MAML library (within calling component)

Why Google?

Google has been specializing in Data processing from 2002 as seen below and leverages the best for its own heavy data stuff like google search /gmail/youtube etc which is now available to all. Its network availability is enormous.



Its customer base speaks for itself: https://cloud.google.com/customers

Twitter/Home Depot/Go Jek/Blume Global GCP solution analysis helped determine the following:

– Google can handle huge amount of data at required performance expectations through its tools

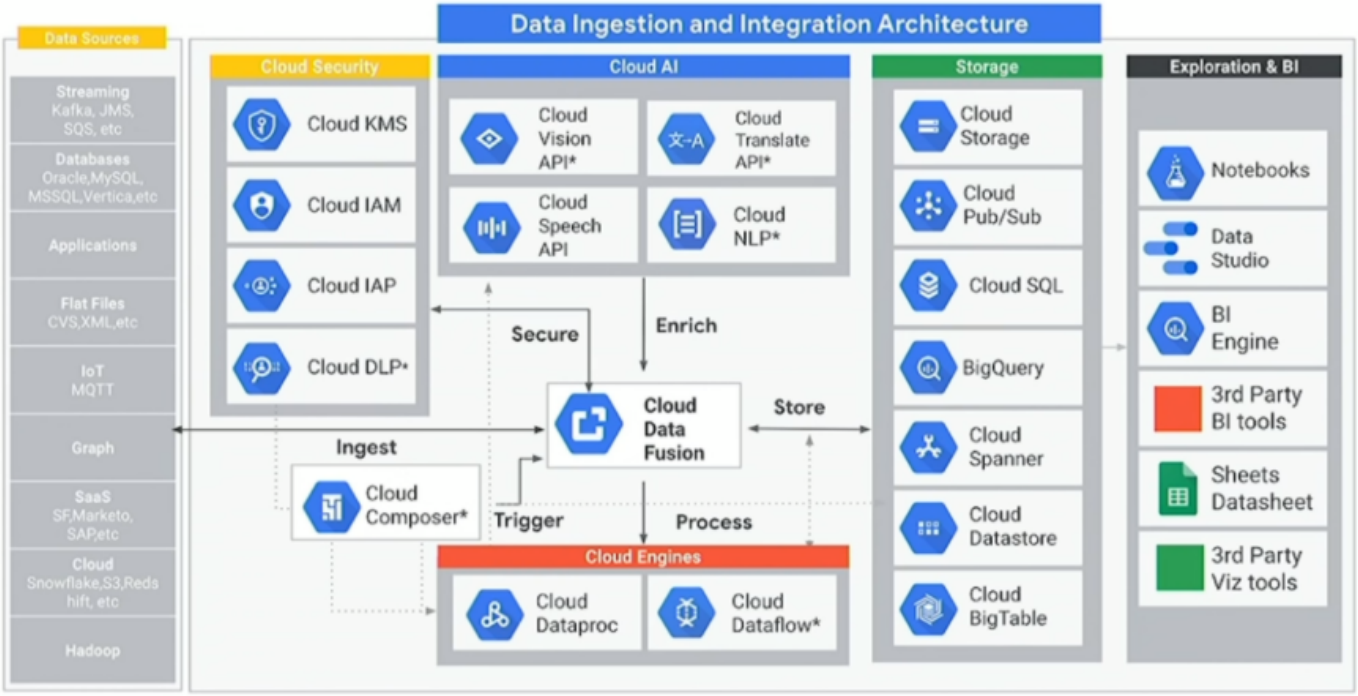
– Google doesn't promote only its tools and makes other competitive tools like DataBricks available within it

It is not a competitor to MA's customers.

Why Cloud Data Fusion?

A single pipeline for ETL, Feature Engineering, Model Training and export using Cloud Data Fusion as opposed to many other options where ETL/ELT pipeline is distinct from pipelines involving Feature Engineering/Model Training and Exporting model

Below picture says it all for all capabilities available with Cloud Data Fusion:



- **Open-source:** Its built on top of CDAP and it therefore enjoys a big community that keep on developing new connectors.
- **Accessible:** Great user interface requiring no coding for many integrations. It already comes with more than 100 connectors and good set of built in transformations that can be seamlessly applied to data and it is constantly growing.
- **Metadata:** search integrated datasets by technical and business metadata. Track lineage for all integrated datasets at the dataset and field level.
- **Flexible:** if you can't do something through the UI, Data Fusion is extensible and you can add your own code to it for transformations (custom plugins can be developed and included, also Cloud Data Fusion Hub has additional plugins/pre-built pipelines that can be used
- **GCP-native:** fully managed, GCP-native architecture unlocks the scalability, reliability, security and privacy guarantees of Google Cloud
- **Dataproc Cluster and Apache Spark based:** Internally it works out of Data proc cluster and Apache Spark based job submission that is already proven to work best with Big Data
- **Cheaper than many of its competitors:** Data Fusion has a basic version 120 hrs free per month. Per current requirements, if Data Fusion is used for Training only, it could be done using this version reducing cost. Idea is that either there will be a feature store to transform data or calling application will produce transformed data during production requests. Typical structured data models should work with this.

Why Big Query?

- **Simplicity and architectural integrations**
  - ANSI SQL compliant
  - No-op/serverless—ability to add storage and compute without getting into cycles of determining the right server type, procuring, installing, launching, etc.
  - Independent scaling of storage and compute
- **Reliability**
  - Reliability and availability: 99.9% monthly uptime
- **Scale**
  - Storage capacity: hundreds of PB
  - Query capacity: exabyte per month
  - Concurrency: 100+ queries with graceful degradation and interactive response
  - Streaming ingestion to support 100s of TB/day
- **Visualization and interactivity**
  - Mature integration with BI tools
  - Materialized views and query rewrite
- **Cost-efficient at scale**
- **BigQuery ML**
  - Simple ML Model creation, evaluation, prediction (supports all major models from Random forest, XG Boost, Time Series, recommendation based, custom tensor flow models, Auto-ML Regressor and classifiers). Allows to export models to be deployed outside
  - **Auto-ML Model advantages:**
    - It uses deep learning capabilities of neural networks similar to custom Tensor flow models (exported Auto ml model from Big query is saved as a tensor flow model)
    - When we kick off training, AutoML Tables automatically performs common feature engineering tasks for us and needn't be explicitly coded, including:
      - Normalize and bucketize numeric features.

    - Create one-hot encoding and embeddings for categorical features.
    - Perform basic processing for text features.
    - Extract date- and time-related features from Timestamp columns.
- Federated queries possible from other Cloud Databases

🚀 Deployment strategy

Deploy to Kubernetes clusters/AI platform. Performed load test using Locust on deployment to Kubernetes clusters enabling Auto scaling and there is ability to get response well within SLA limits for Big Query Trained model deployed to Kube clusters.

⌨️ Rest API details

🤝 SLA

Requests Per Second (RPS): 1000

Response Time: 30 ms, 99% of the time

Future Considerations

Leveraging Cloud Data Fusion streaming pipeline or Databricks on Google Cloud or use Data proc Spark Connectors to build Spark Streaming jobs if complex data transformations are required before request is submitted to model for prediction in production

✅ Action Items

	Action	Description	Owner	Due date	Jira ticket
1	☐			22 Mar 2021	
2					

🗂️ References and documentation

- Cloud Data Fusion - https://www.youtube.com/watch?v=kehG0CJw2wo
- Big Query - https://cloud.google.com/blog/products/data-analytics/google-cloud-a-leader-in-2021-forrester-wave-cloud-data-warehouse
- CI/CD Best Practices - https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning