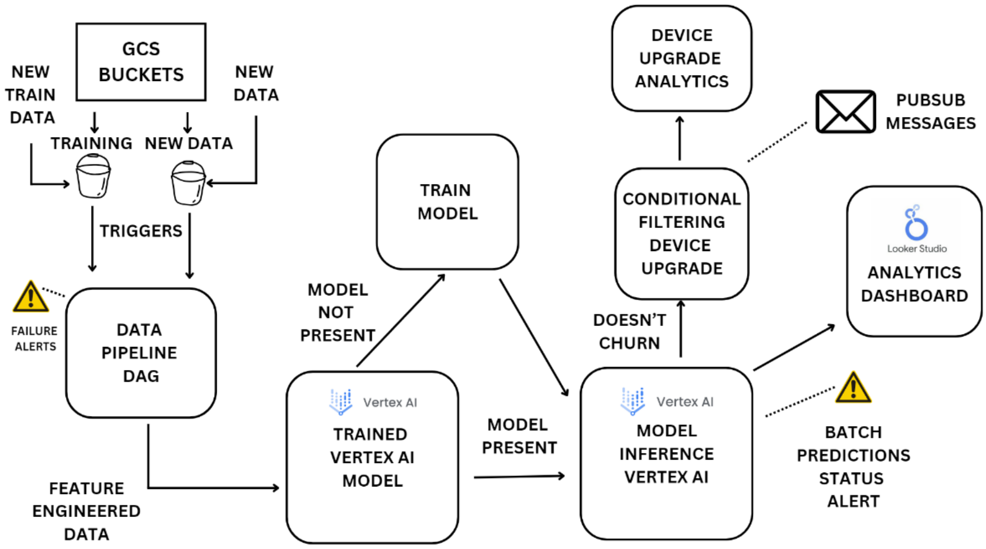
Telecom Customer Churn and Mobile Device Upgrade Prediction

Project Workflow and Screenshots



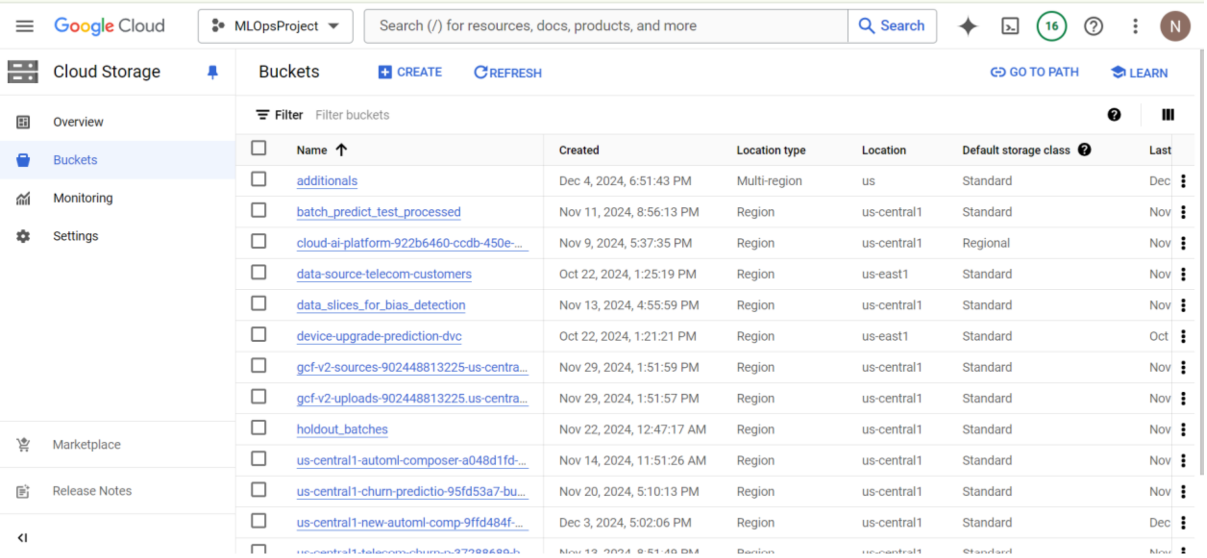
Business Use Case: Telecom companies have a pressing need to analyze dynamic customer behavior in order to send timely offers and prevent customer churn. This project addresses the need by designing a GCP based end-to-end pipeline for predicting customer churn and filtering customers who would upgrade their device to send notifications.

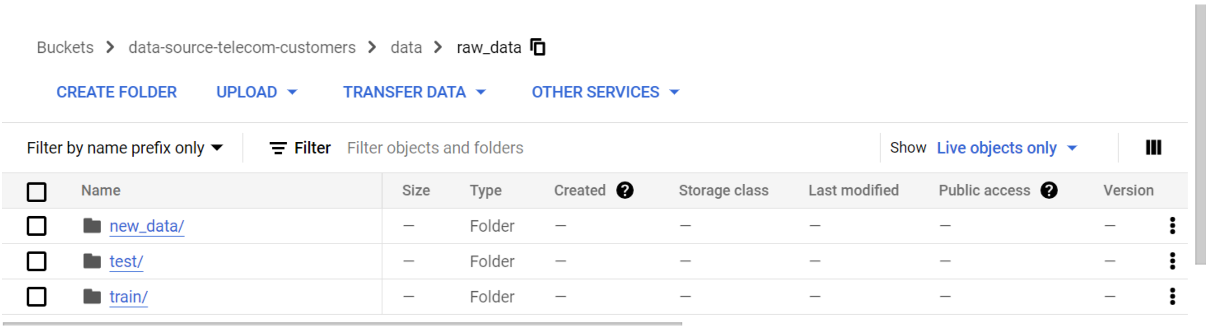
As the telecom companies receive customer data in new **batches**, this pipeline gets triggered to perform churn prediction, device upgrade estimation, and real-time customer analytics on interactive dashboards. We designed an interconnected DAG system with triggers to orchestrate the flow as new data gets ingested.

GCP Project Service Account Permissions:

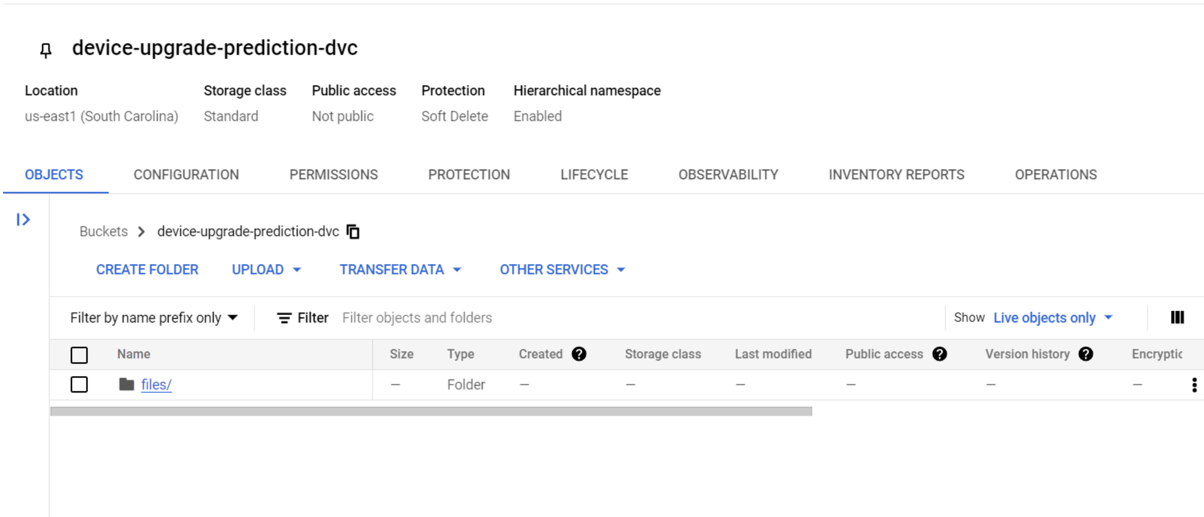
mlops-project-service-account@axial-rigging-438817-h4.iam.gserviceaccount.com	mlops-project-service-account	Cloud Build Logging Service Agent Composer Worker Logs Configuration Writer Logs Viewer Monitoring Editor Monitoring Viewer Owner Storage Admin Storage Object Viewer	10251/10303 excess permis

GCS Buckets for Data:





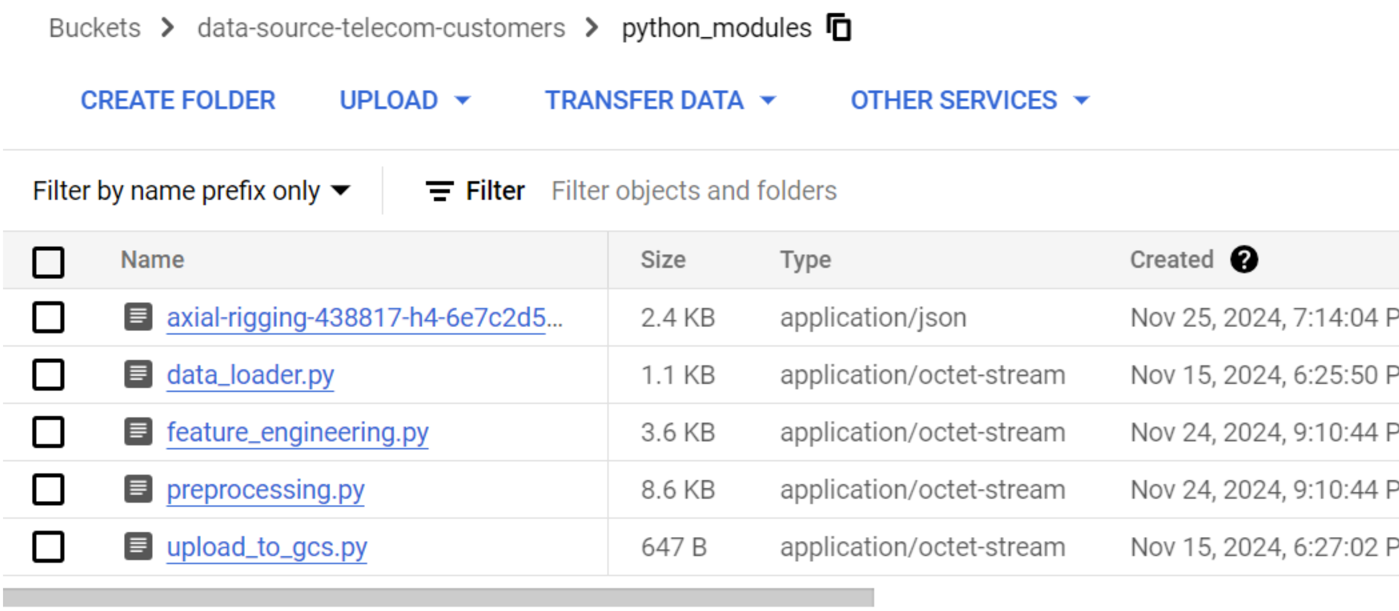
DVC integration for data version control:



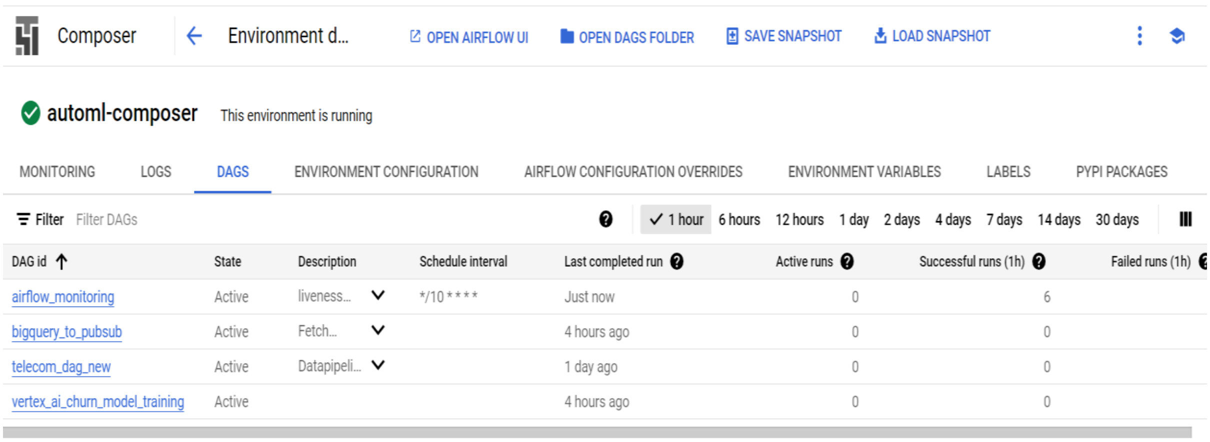
DVC logs cache files stored in a bucket.

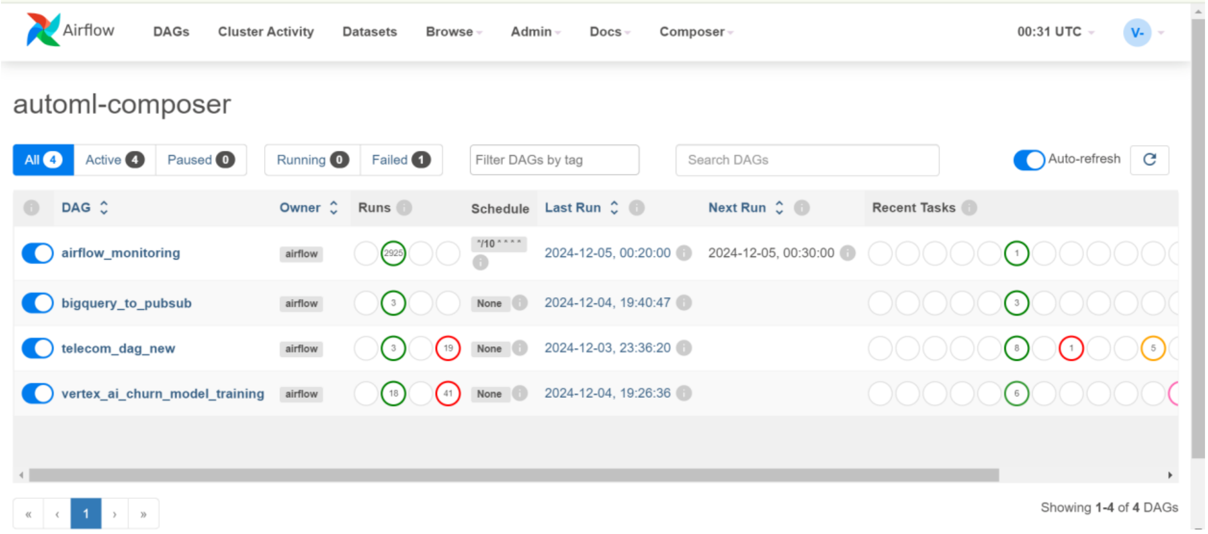
Python modules loaded to GCS:

Custom made python modules that are used by DAG tasks. DAG access these modules from the bucket.

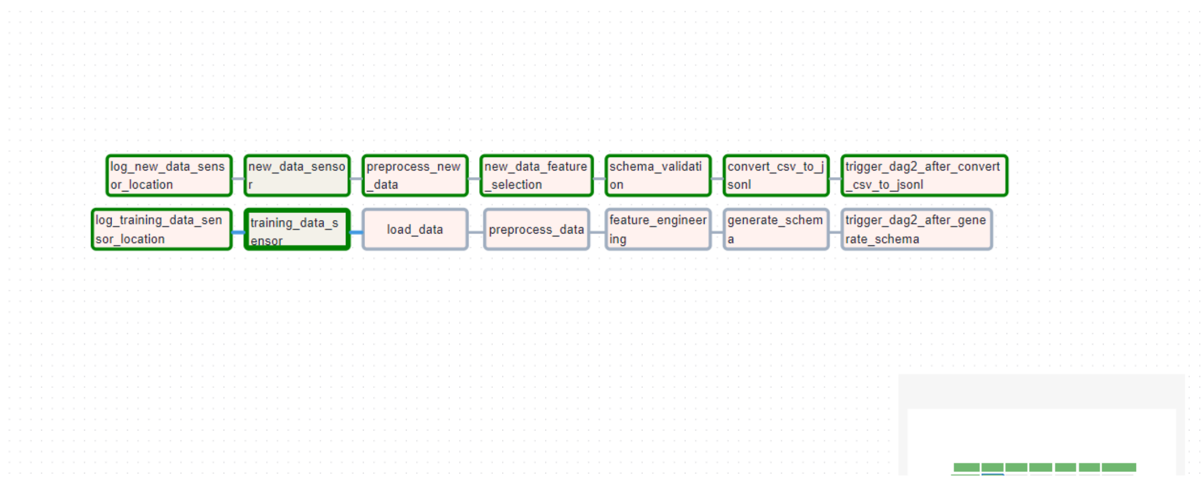


DAGs in Composer environment:

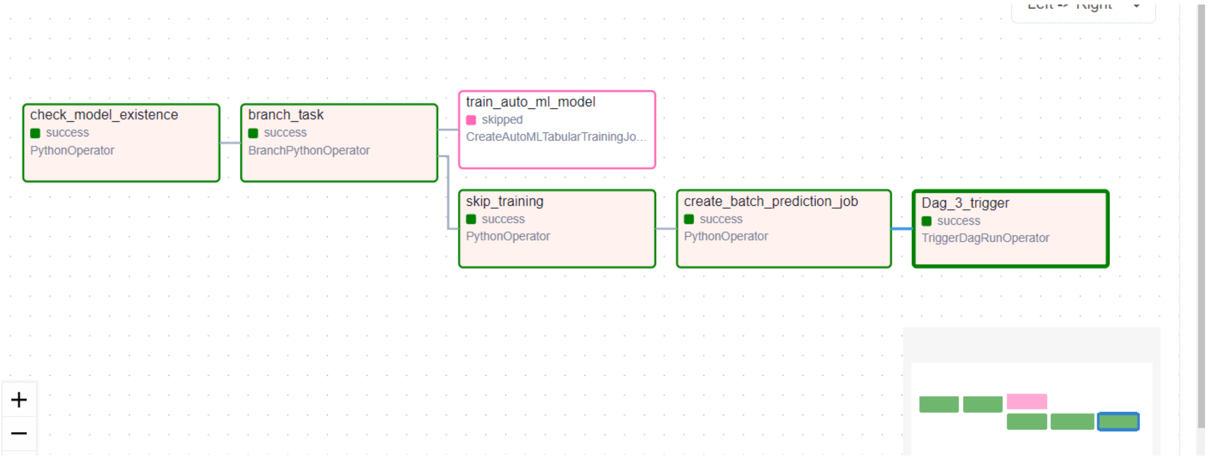




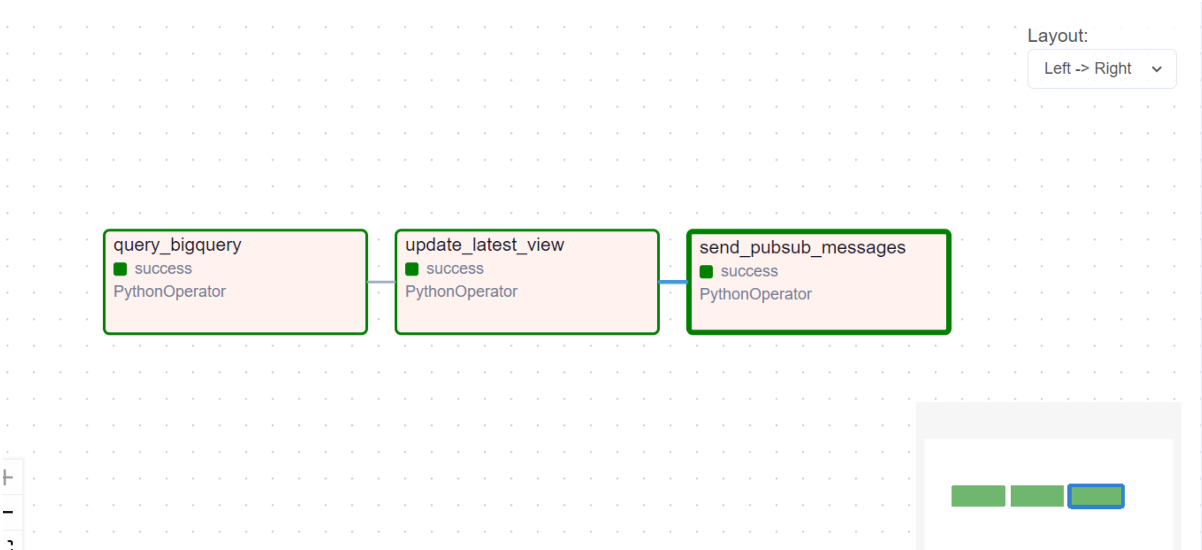
Data Pipeline DAG – gets triggered with incoming new data



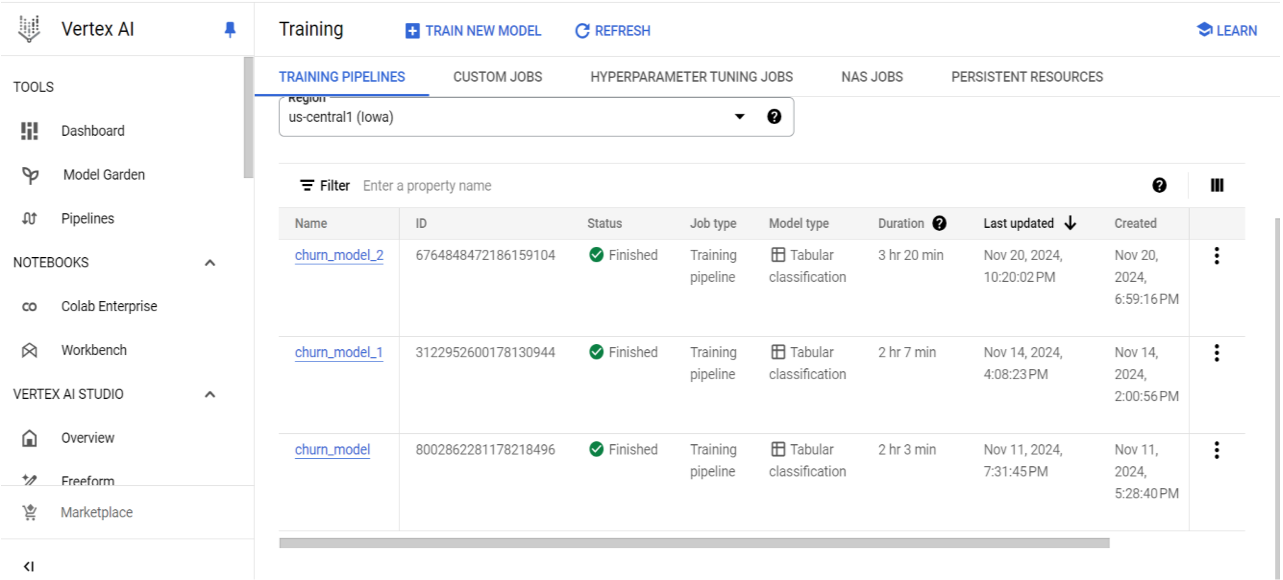
Vertex AI DAG - gets triggered when data DAG runs successfully; makes a call to Vertex AI model (Inference if model is present, training if model is not present already)



Front end DAG - gets triggered when vertex AI DAG runs successfully; Stores predictions in Big Query, updates Looker dashboard, filters customers likely to upgrade and sends messages Pub/Sub



Vertex AI model:



Latest and best model version – churn_model_2.

Notifications: Email alerts received after training completion and every batch prediction completion.

On Sat, Nov 9, 2024 at 9:26 PM Vertex AI <noreply-vertexai@google.com> wrote:

Hello Vertex AI Customer,

Vertex AI finished training model "temp_churn".
Additional Details:
Operation State: Succeeded
Resource Name:
projects/902448813225/locations/us-central1/trainingPipelines/6226524830447960064

To continue your progress, go back to your training pipeline using
https://console.cloud.google.com/vertex-ai/models?authuser=1&hl=en&project=axial-rigging-438817-h4

Sincerely,
The Google Cloud AI Team

Vertex AI finished batch prediction using model "bias_detection_currentequip_slice1" Inbox ×

Vertex AI <noreply-vertexai@google.com> Thu, Nov 14, 5:51 PM
to me

Hello Vertex AI Customer,

Vertex AI finished batch prediction using model "bias_detection_currentequip_slice1".
Additional Details:
Operation State: Succeeded
Resource Name:
projects/902448813225/locations/us-central1/batchPredictionJobs/9134017853601611776

To continue your progress, go back to your predictions using
https://console.cloud.google.com/vertex-ai/models/locations/us-central1/models/2993104846782988288/versions/1/batch-predictions?referrer=search&project=axial-rigging-438817-h4

Sincerely,
The Google Cloud AI Team

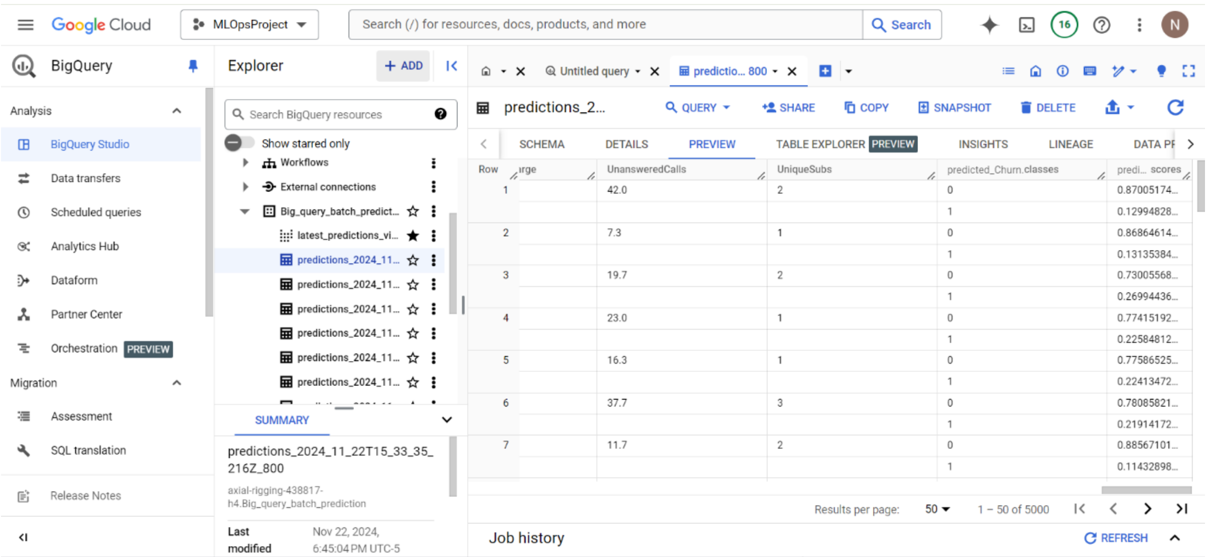
Vertex AI model inferences stored on Big Query for every batch:

```python
# Task: Create batch prediction job
def create_batch_prediction_job(**kwargs):
    log = get_task_logger("create_batch_prediction_job")
    try:
        model_name = kwargs['ti'].xcom_pull(task_ids='check_model_existence', key='model_name')
        if not model_name:
            raise ValueError("No model found for batch prediction.")

        log.info("Initializing Vertex AI platform for batch prediction...")
        aiplatform.init(project=PROJECT_ID, location=LOCATION)

        gcs_input = f"gs://{BUCKET_NAME}/latest_best_features_for_churn.jsonl"
        bigquery_output_prefix = "axial-rigging-438817-h4.Big_query_batch_prediction"

        log.info("Creating batch prediction job...")
        batch_prediction_job = aiplatform.BatchPredictionJob.create(
            job_display_name="batch_prediction_job",
            model_name=model_name,
            gcs_source=gcs_input,
            predictions_format="bigquery",
            starting_replica_count=10,
            max_replica_count=20,
            machine_type="c2-standard-30",
            bigquery_destination_prefix=bigquery_output_prefix
```



Model Performance:

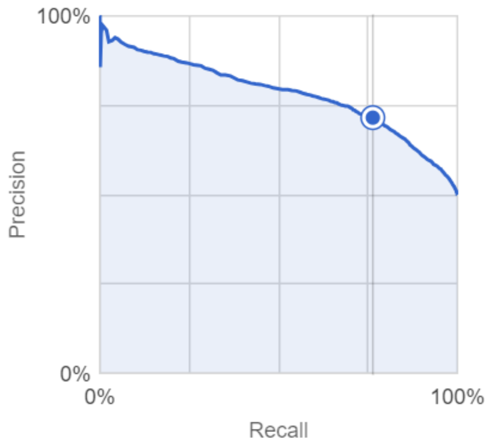
Evaluation details

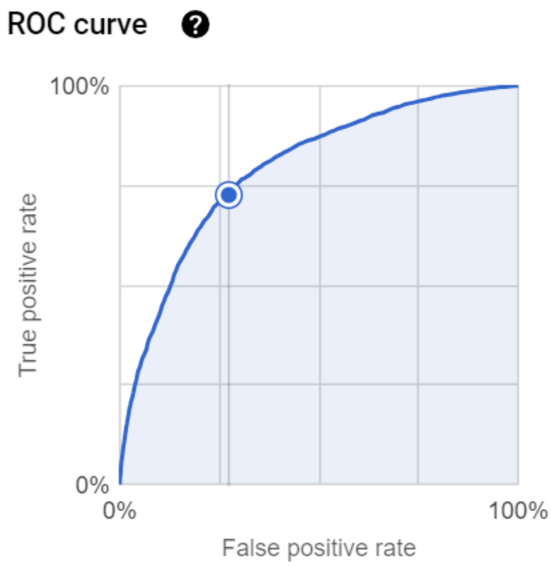
Confidence threshold 0.5

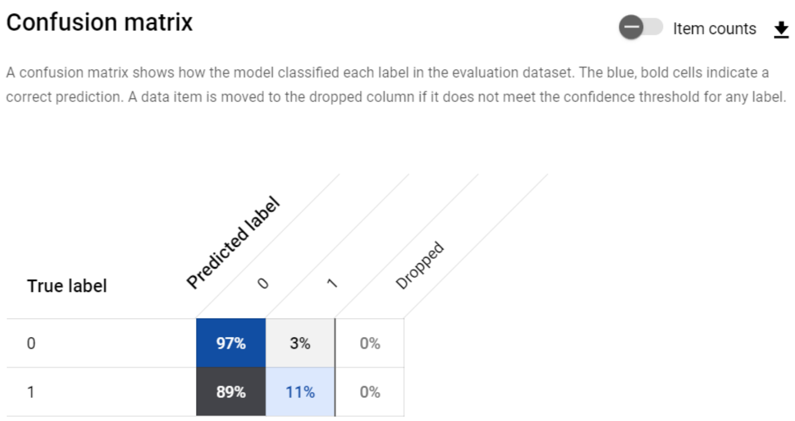
All labels

PR AUC	0.782
ROC AUC	0.793
Log loss	0.551
Micro-average F1	0.72627234
Macro-average F1	0.5111234
Micro-average precision	72.6%
Micro-average recall	72.6%

Precision-recall curve






Confusion matrix

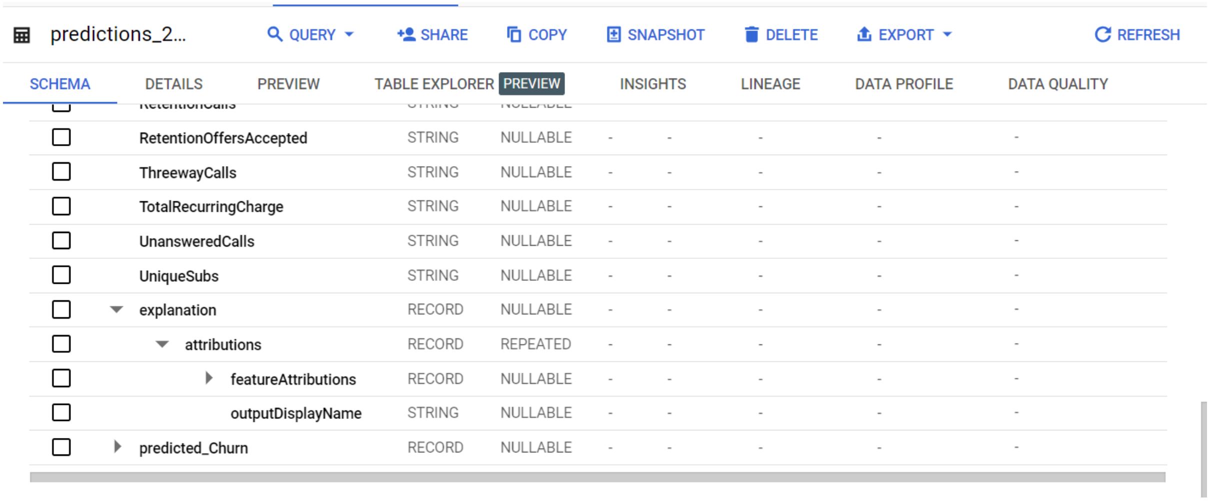
Item counts

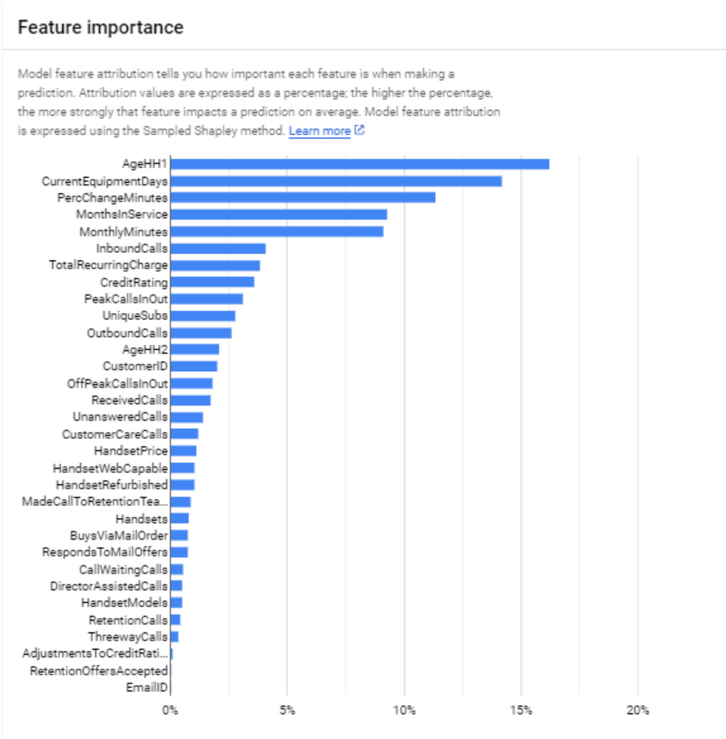
A confusion matrix shows how the model classified each label in the evaluation dataset. The blue, bold cells indicate a correct prediction. A data item is moved to the dropped column if it does not meet the confidence threshold for any label.

True label \ Predicted label	0	1	Dropped
0	97%	3%	0%
1	89%	11%	0%

Feature Sensitivity Analysis:

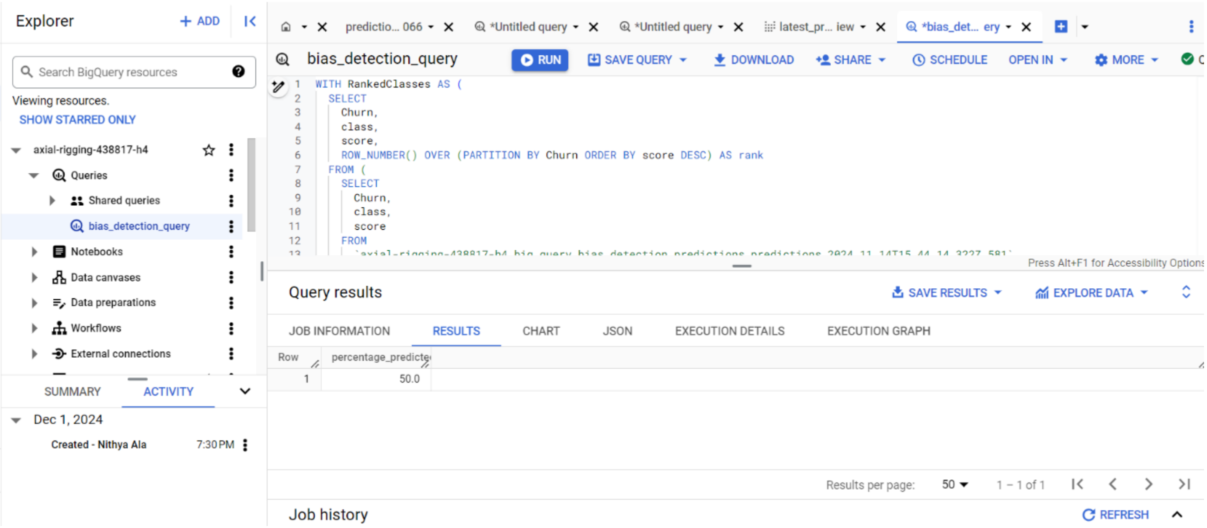
Explanations stored in big query



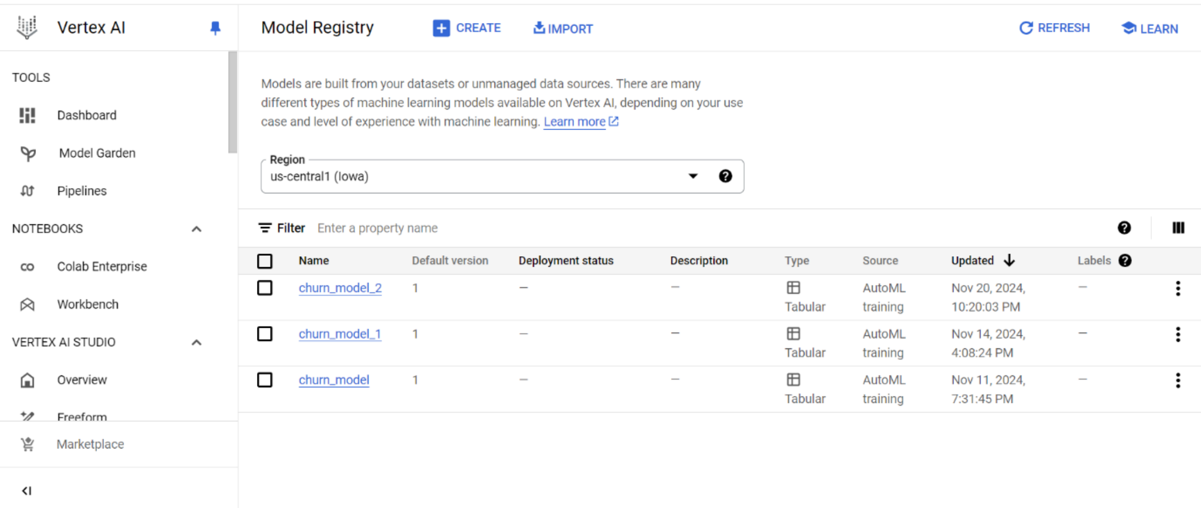


Bias Detection:

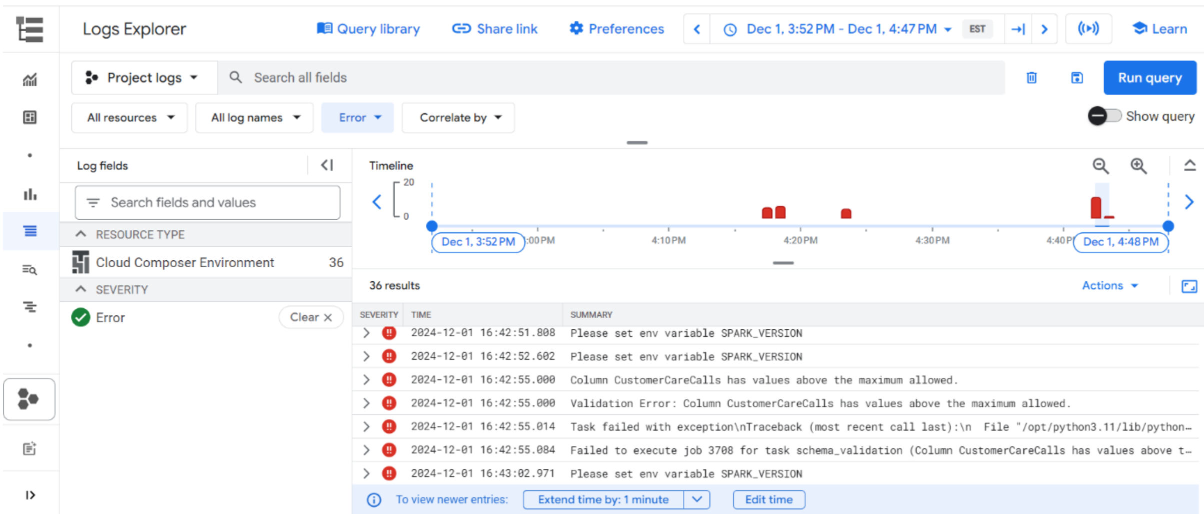
Bias detection performed on the most relevant column 'CurrentEquipmentDays'. Data sliced into 3 slices, predictions stored in big query and queried to analyze the model's performance on different slices. The model performance is similar for all slices so no bias was detected.



Model Registry:



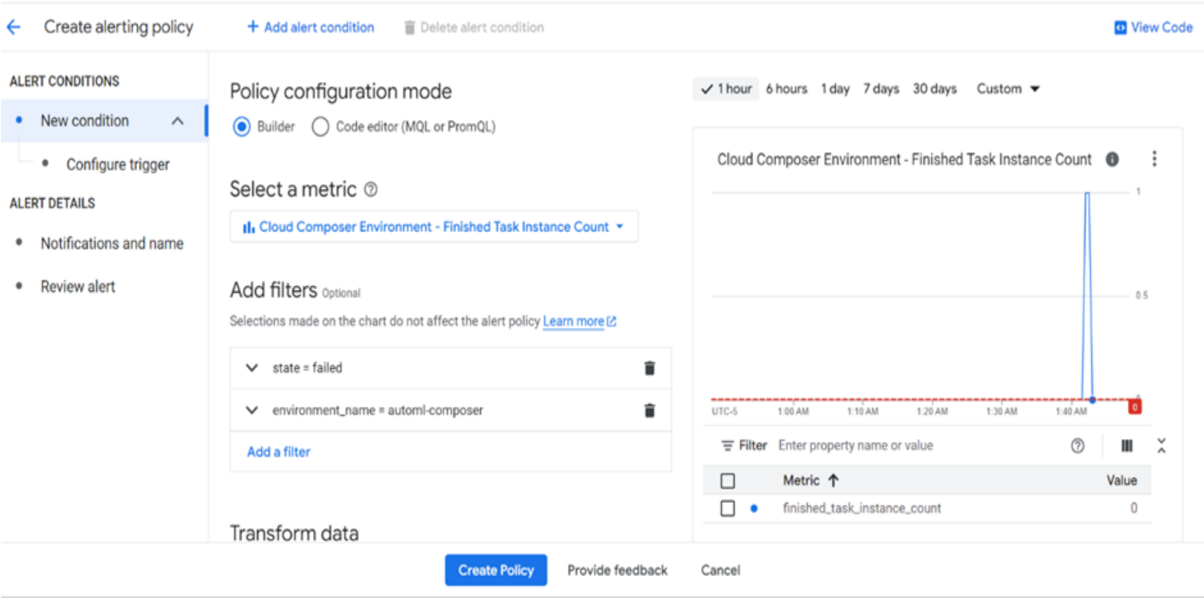
Logging and Alerts:

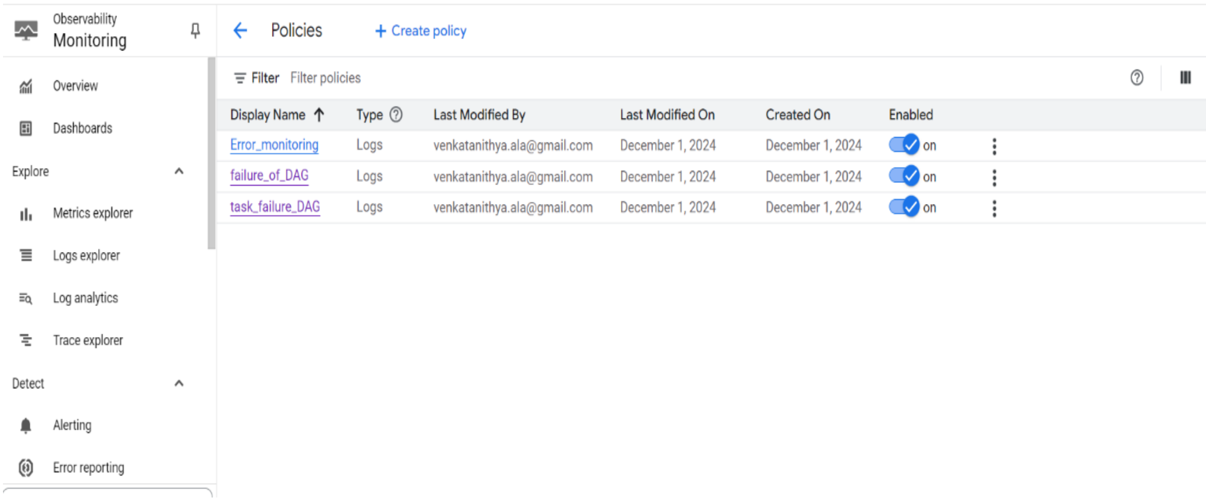


Utilizing log explorer for analyzing logs.

Alert policies set up to monitor task failures and DAG failures.

Email and SMS alerts set up.





Policy details · Enabled · Edit · Copy · Delete · JSON · 1 hour · 6 hours · 1 day · 1 week · 1 month · 6 weeks

failure_of_DAG

Conditions
Policy violates when ANY condition is met

Severity
Error

Configurations

Log query
resource.type="cloud_composer_environment" resource.labels.location="us-central1" resource.labels.environment_name="automl-composer" log_name="projects/axial-rigging-438817-h4/logs/airflow-scheduler" severity=ERROR

Notification rate limit
One notification per 5 minutes

Incident autoclose duration
1 day

Logs · Severity: Default · Filter · Search all fields and values

SEVERITY	TIMESTAMP	SUMMARY
!	2024-11-29 11:12:14.527 EST	airflow-scheduler Marking run <DagRun bigquery_to_pubsub @ 2024-11-29 16:07:07.814909+00:00: manual__2024-11-29T16:07:07.814909+00:00, state:running, queued_at: 202…
!	2024-11-29 11:39:06.829 EST	airflow-scheduler Marking run <DagRun bigquery_to_pubsub @ 2024-11-29 16:33:49.737569+00:00: manual__2024-11-29T16:33:49.737569+00:00, state:running, queued_at: 202…
!	2024-11-29 12:11:06.733 EST	airflow-scheduler Marking run <DagRun bigquery_to_pubsub @ 2024-11-29 17:05:57.776741+00:00: manual__2024-11-29T17:05:57.776741+00:00, state:running, queued_at: 202…
!	2024-12-01 01:47:06.035 EST	airflow-scheduler Marking run <DagRun telecom_dag_new @ 2024-12-01 06:41:33.700044+00:00: manual__2024-12-01T06:41:33.700044+00:00, state:running, queued_at: 2024-1…
!	2024-12-01 02:05:45.052 EST	airflow-scheduler Marking run <DagRun telecom_dag_new @ 2024-12-01 07:00:13.094020+00:00: manual__2024-12-01T07:00:13.094020+00:00, state:running, queued_at: 2024-1…
!	2024-12-01 15:31:11.861 EST	airflow-scheduler Marking run <DagRun telecom_dag_new @ 2024-12-01 20:25:41.311394+00:00: manual__2024-12-01T20:25:41.311394+00:00, state:running, queued_at: 2024-1…

task_failure_DAG

Conditions
Policy violates when ANY condition is met

Severity
Error

Configurations

Log query
resource.type="cloud_composer_environment" resource.labels.location="us-central1" resource.labels.environment_name="automl-composer" log_name="projects/axial-rigging-438817-h4/logs/airflow-worker" severity=ERROR

Notification rate limit
One notification per 5 minutes

Incident autoclose duration
1 day

Logs · Severity: Default · Filter · Search all fields and values

SEVERITY	TIMESTAMP	SUMMARY
!	2024-12-01 16:42:55.014 EST	airflow-worker Task failed with exception\nTraceback (most recent call last):\n File "/opt/python3.11/lib/python3.11/site-packages/airflow/models/taskinstance.py", …
!	2024-12-01 16:42:55.084 EST	airflow-worker Failed to execute job 3708 for task schema_validation (Column CustomerCareCalls has values above the maximum allowed.; 168842)
!	2024-12-01 16:43:02.971 EST	airflow-worker Please set env variable SPARK_VERSION
!	2024-12-01 16:47:56.962 EST	airflow-worker Please set env variable SPARK_VERSION
!	2024-12-01 16:47:59.431 EST	airflow-worker Column CustomerCareCalls has values above the maximum allowed.
!	2024-12-01 16:47:59.432 EST	airflow-worker Validation Error: Column CustomerCareCalls has values above the maximum allowed.
!	2024-12-01 16:47:59.437 EST	airflow-worker Task failed with exception\nTraceback (most recent call last):\n File "/opt/python3.11/lib/python3.11/site-packages/airflow/models/taskinstance.py", …

11 of 1,024

[ALERT - Error] task_failure_DAG for Cloud Composer Environment with {environment_name=automl-composer, location=us-central1, project_id=axial-rigging-438817-h4} Inbox ×

Google Cloud Alerting Sun, Dec 1, 4:42 PM (21 hours ago)
VIEW INCIDENT Log alert fired Error Cloud Composer Environment with a log matching the query has appeared Start time Dec 1, 2024 at 9:42PM UTC (less than 1 sec

Google Cloud Alerting <alerting-noreply@google.com> Sun, Dec 1, 4:48 PM (21 hours ago)
to me

Google Cloud · VIEW INCIDENT

Log alert fired · Error

Cloud Composer Environment with a log matching the query has appeared

Start time
Dec 1, 2024 at 9:42PM UTC (~5 minutes ago)

Policy
task_failure_DAG

Project
axial-rigging-438817-h4

Condition
Log match condition

environment_name : automl-composer · location : us-central1 · project_id : axial-rigging-438817-h4

2:27

50303

Yesterday • 4:42 PM

[ALERT - Error] task_failure_DAG for Cloud Composer Environment with {environment_name=automl-composer, location=us-central1, project_id=axial-rigging-438817-h4} ALRT-Error task_failure_DAG Log match condition

Unread

[ALERT - Error] task_failure_DAG for Cloud Composer Environment with {environment_name=automl-composer, location=us-central1, project_id=axial-rigging-438817-h4} ALRT-Error task_failure_DAG Log match condition

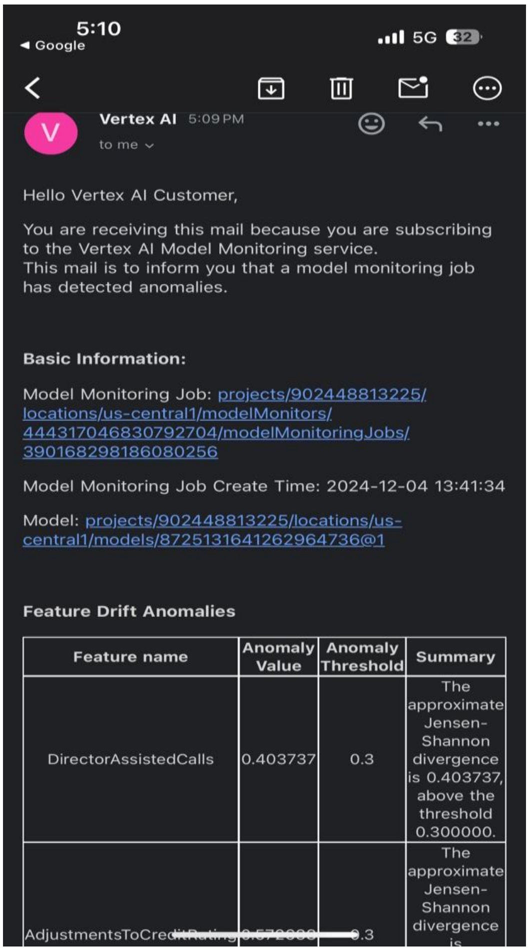
[ALERT - Error] failure_of_DAG for Cloud Composer Environment with {environment_name=automl-composer, location=us-central1, project_id=axial-rigging-438817-h4} ALRT-Error failure_of_DAG Log match condition

4:48 PM

Text message

Model Monitoring:

Model monitoring jobs set up and email alerts set up.



Drift detection jobs set up for analyzing feature drift, email alerts set up if drift score goes over the threshold.

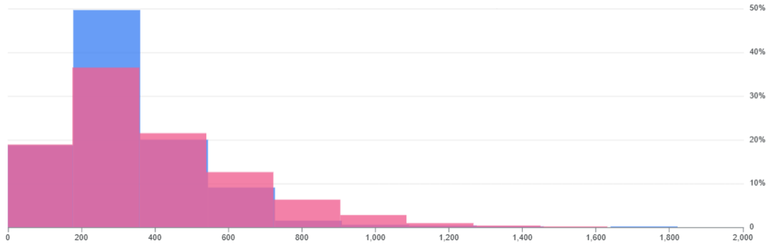
Features	Baseline	Target	Alerts	Drift score	Threshold	Metric
AdjustmentsToCreditRating				0.58344	0.3	Jensen-Shannon
AgeHH1			–	0.00102	0.3	Jensen-Shannon
AgeHH2			–	0.00532	0.3	Jensen-Shannon
BuysViaMailOrder			–	0.02425	0.3	L-Infinity
CallWaitingCalls			–	0.03946	0.3	Jensen-Shannon
CreditRating				0.49963	0.3	L-Infinity
CurrentEquipmentDays			–	0.02856	0.3	Jensen-Shannon
CustomerCareCalls			–	0.19007	0.3	Jensen-Shannon
CustomerID			–	0.13037	0.3	Jensen-Shannon
DirectorAssistedCalls				0.3237	0.3	Jensen-Shannon
EmailID			–	0.16255	0.3	L-Infinity
HandsetModels			–	0.0295	0.3	Jensen-Shannon
HandsetPrice				0.61517	0.3	Jensen-Shannon
HandsetRefurbished			–	0.01126	0.3	L-Infinity
HandsetWebCapable			–	0.04381	0.3	L-Infinity
Handsets			–	0.05549	0.3	Jensen-Shannon

INPUT FEATURE DRIFT | OUTPUT PREDICTION DRIFT | FEATURE ATTRIBUTION DRIFT | CONFIGURATION

Run status	Run time	Target	Baseline
Succeeded	Dec 5, 2024, 6:30:01 PM	BigQuery table View details	GCS object View details

CurrentEquipmentDays: Target vs Baseline

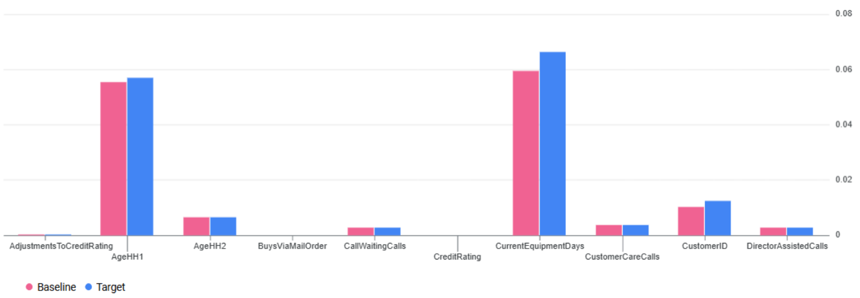
Snapshot of distribution job ran Dec 5, 2024, 6:30:00 PM



Statistics	Baseline	Target
Data points	51047	20015
Null	0%	0%
Mean	380.5444590279546	314.08463652260804
Min	-5	-5
Median	329	287
Max	1812	1823
Std Deviation	253.79720238700855	194.5371424855205
Zeros	0.04%	0.03%

Run status	Run time	Target	Baseline
Succeeded	Dec 5, 2024, 6:30:01 PM	BigQuery table View details	GCS object View details

Target vs Baseline - Feature attribution score



Features	Alerts	Score	Baseline score	Deviation	Threshold
AdjustmentsToCreditRating	–	0.00029	0.00031	-0.00002	0.3
AgeHH1	–	0.057	0.05538	0.00162	0.3
AgeHH2	–	0.00648	0.00665	-0.00017	0.3
BuysViaMailOrder	–	0	0		0.3
CallWaitingCalls	–	0.00269	0.00268	0.00001	0.3
CreditRating	–	0	0		0.3
CurrentEquipmentDays	–	0.06634	0.05942	0.00692	0.3

Retraining criteria:

Our architecture supports **on-demand retraining**, triggered by the addition of new data to the 'training data' bucket. The decision to include new data for retraining is informed by schema validation and anomaly detection tasks, which are part of the DAGs processing the incoming data. We also consider input feature drift and model performance (model drift) to prepare new training data and place it in the designated bucket. This action automatically triggers the training DAG, initiating the retraining process. By centralizing operations on GCP with storage buckets, Cloud Composer, and Vertex AI, our system maintains streamlined and automated workflows, eliminating the need for traditional CI/CD pipelines.

CI/CD:

We determined that integrating with Git for CI/CD workflows related to code changes is unnecessary for our project as it is fully deployed on Google Cloud Platform (GCP), leveraging Vertex AI's AutoML for model training and predictions. Since Vertex AI AutoML handles the modeling process, no custom model code is required. We have implemented automatic triggers based on incoming data, ensuring that Dataflow pipelines and directed acyclic graphs (DAGs) in Cloud Composer are executed whenever new data is added to the GCP storage buckets. The logic for these DAGs is predefined and fixed, automating the entire workflow without requiring code modifications.

Front End – Looker Analytics Dashboard

Since the telecom companies receive data in big batches, a front end is designed to display analytics of the batches rather than per customer user input-prediction design.