**Hackathon: Vertex AI**

**Ideas:**

The goal is to build models using AutoML, and use [Vertex AI Pipelines](https://cloud.google.com/vertex-ai/docs/pipelines/introduction) to orchestrate the process of training models with AutoML and deploy it to a Vertex AI Endpoint. I am hoping that by leveraging the automated tools in Vertex AI, we will be able to streamline the model development process, conduct robust feature engineering, deploy models efficiently to Vertex AI Endpoints, and MLOps.

What can AutoML do?

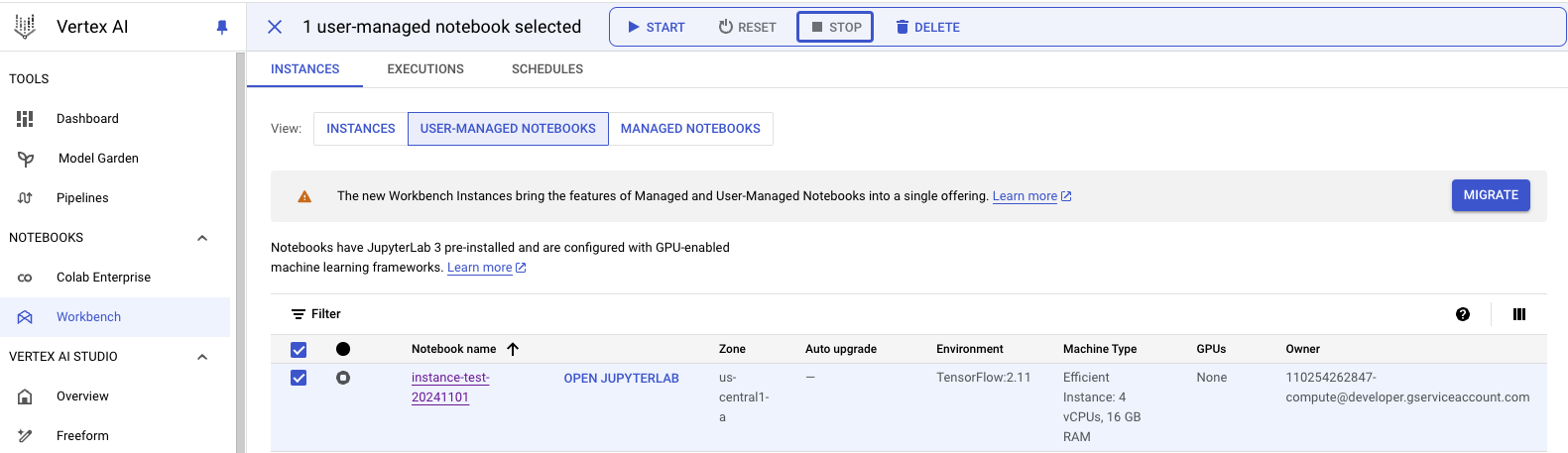
* Automated Feature Engineering:

The idea is once the data is there, the auto ML can process data (feature engineering) including scaling, standardization, one-hot encode. For timestamp data, AutoML will extract meaningful date and time temporal features.

* Model Diversity: Train a variety of model types such as neural networks and regression models.
* Model Evaluation and Optimization: We can define metrics for model evaluation and implement early stopping.
* Metric: One metric to be considered is the allocated training time, we tell it how many hours AutoML should keep trying and find the best model.
* Deployment: Deploy the best-performing model to a Vertex AI Endpoint

**Resources & References:**

* **Vertex AI MLOps GitHub Resources**:
  + [MLOps with Vertex AI](https://github.com/GoogleCloudPlatform/mlops-with-vertex-ai)
  + [Vertex AI Samples](https://github.com/GoogleCloudPlatform/vertex-ai-samples)
* **Pricing Information**:
  + [BigQuery ML Pricing](https://cloud.google.com/bigquery/pricing#bqml)
  + I signed up for a free trial and $300 credit. I have created a notebook in Vertex AI and will share it with everyone.



**Data Sources:**

* **Public BigQuery Datasets**:
  + [BigQuery Public Data](https://cloud.google.com/bigquery/public-data) (free BQ datasets we could use for this project).
* Alternatively we could also take sample data from the Kevala project.
* **CSV Data Loading**:
  + We can create a bucket in Cloud Storage, load their data, and integrate it with their Vertex AI environment.  
    [Loading CSV data from Cloud Storage](https://cloud.google.com/bigquery/docs/loading-data-cloud-storage-csv)

**Models:**

Model types are [here](https://cloud.google.com/vertex-ai/docs/training-overview). For this project, we are probably going to try forecasting models. Also open to discussions on exploring additional model types such as regression models and neural networks.

Example of using AutoMLForecastingTrainingJob

* Example: [AutoMLForecastingTrainingJob](https://cloud.google.com/python/docs/reference/aiplatform/latest/google.cloud.aiplatform.AutoMLForecastingTrainingJob)

Using batch prediction:

Reference: [Batch Prediction Job](https://cloud.google.com/python/docs/reference/aiplatform/latest/google.cloud.aiplatform.BatchPredictionJob)

**Pipelines:**

Use [Kubeflow](https://www.kubeflow.org/) Pipelines running on [Vertex AI Pipelines](https://cloud.google.com/vertex-ai/docs/pipelines/introduction) to orchestrate the process of training a custom model with AutoML and deploy it to a Vertex AI Endpoint.

Resources: [Vertex AI Pipelines Overview](https://cloud.google.com/vertex-ai/docs/training-overview#tabular)

**Endpoint Deployment**:

* Set up Vertex AI Endpoints and deploy the model (maybe with traffic percentage configuration?).
* References: [Endpoint Deployment](https://cloud.google.com/vertex-ai/docs/general/deployment)

**Model Evaluation**:

* Incorporate feature explanations to interpret model results and validate the model’s effectiveness.
* Reference: [Forecasting Explanations](https://cloud.google.com/vertex-ai/docs/tabular-data/forecasting-explanations)

**MLOps**:

* Reference:[MLOps](https://cloud.google.com/vertex-ai/docs/start/introduction-mlops)

**Next Steps**:

Karin to set up the development environment with a shared project and notebook. (I need to figure out the best way to share our work. One idea is to use the BigQuery ML plugin in dbt.)

Looking forward to great ideas and suggestions from you.