Team Deliverable 2

Team 4: Ahmed Farid Khan, Bennett Blanco, Dhruv Shah, Yu-Chin (Alyssa) Chen

---

Progress Summary:

After completing our ETL pipeline for the Strava project, we deployed Apache Superset on Google Compute Engine (GCE) to visualize our data stored in BigQuery. We set up a GCE VM instance to host Superset, selecting a machine type that meets Superset's performance needs. After configuring HTTP/S access and installing Docker and Docker Compose, we cloned the Superset repository and launched the application. Within Superset, we connected directly to BigQuery, allowing us to query the Strava data and create datasets for analysis.

Using Superset's SQL Lab, we built charts that were compiled into a dashboard displaying key performance metrics. To maintain accessibility, we created a firewall rule to open the required port and set the VM to be paused when not in use, reducing cloud costs. This setup allows for scalable and interactive data visualization, providing valuable insights from BigQuery in a flexible dashboard format.

In addition to data visualization, we developed several machine learning models, including Calories Burned Prediction, Grade Adjusted Pace, and Run Classification.

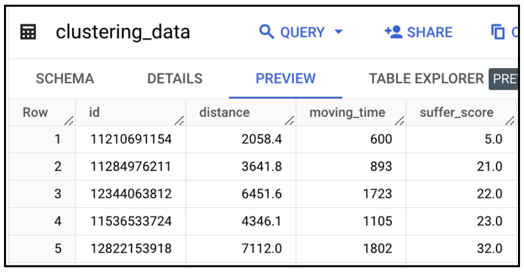
- **Calories Burned Prediction**: This model enables the prediction of potential calories burned, which could assist users in managing their energy output effectively. We tested linear regression, random forest regression, and XGBoost models, finding that linear regression performed best, achieving a Mean Absolute Error (MAE) of 24.89 and an R^2 of 0.99.
- **Grade Adjusted Pace Prediction**: This model predicts the moving time, allowing us to estimate both the pace and the grade-adjusted pace (GAP) for activities with varying elevation. Pace is calculated by dividing the moving time by distance, so by comparing the standard pace to GAP, we can better assess the impact of elevation changes on effort. After testing three models, the Random Forest model performed best, achieving a Mean Absolute Error (MAE) of 1.12 and an R^2 of 0.99.
- **Clustering to Categorize Types of Runs**: We applied clustering techniques to categorize different types of runs based on features such as distance, moving time, and suffer score. This model helps group runs into clusters, allowing users to understand the nature of their activities better and tailor their training accordingly. We used unsupervised learning methods such as K-means and hierarchical clustering to create meaningful categories for further analysis and recommendation. The runs were categorized into one of the following four types:

    - Low Intensity Run
    - Medium-Distance Steady Run
    - Marathon Prep

- ○ Long Tempo Run

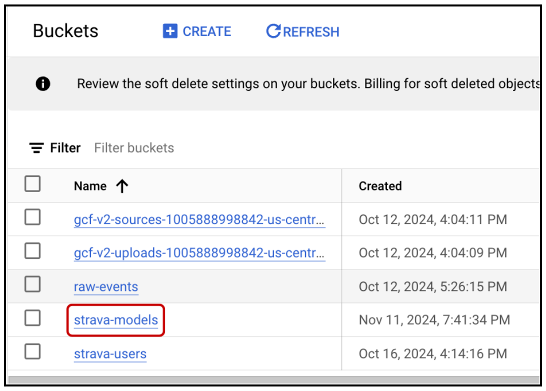
Based on the insights from each model, we decided that the clustering algorithm revealed the greatest information out of all of them and chose to deploy it in a Machine Learning pipeline. Here are the steps we undertook to achieve that:

- We wrote a cloud function to create a new BigQuery table which stored the data we needed for the clustering algorithm.



Row	id	distance	moving_time	suffer_score
1	11210691154	2058.4	600	5.0
2	11284976211	3641.8	893	21.0
3	12344063812	6451.6	1723	22.0
4	11536533724	4346.1	1105	23.0
5	12822153918	7112.0	1802	32.0

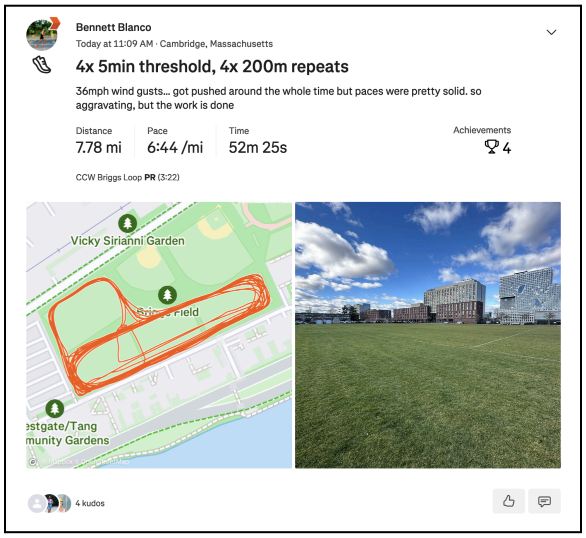
- Using this data, we created and saved the model inside Google Cloud Storage buckets with another Cloud Function.



Review the soft delete settings on your buckets. Billing for soft deleted objects

Name	Created
gcf-v2-sources-1005888998842-us-centr...	Oct 12, 2024, 4:04:11 PM
gcf-v2-uploads-1005888998842-us-centr...	Oct 12, 2024, 4:04:09 PM
raw-events	Oct 12, 2024, 5:26:15 PM
strava-models	Nov 11, 2024, 7:41:34 PM
strava-users	Oct 16, 2024, 4:14:16 PM

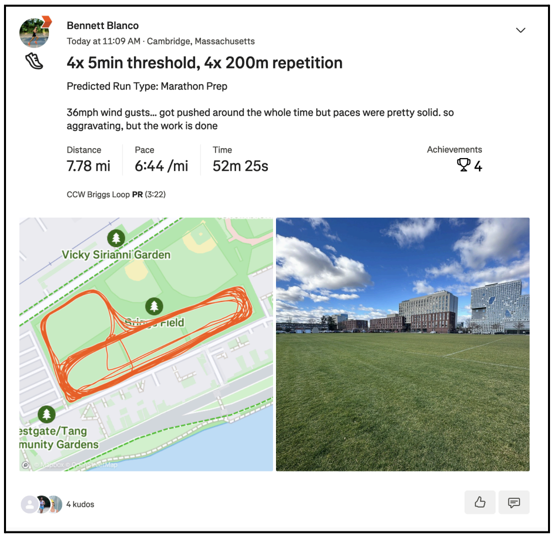
- Lastly, we wrote a third cloud function which predicts the type of run the Strava user just did and posts it to the Strava App using Strava's API.



Bennett Blanco
Today at 11:09 AM · Cambridge, Massachusetts
4x 5min threshold, 4x 200m repeats
36mph wind gusts... got pushed around the whole time but paces were pretty solid. so aggravating, but the work is done

Distance	Pace	Time	Achievements
7.78 mi	6:44 /mi	52m 25s	4

CCW Briggs Loop **PR** (3:22)



Bennett Blanco
Today at 11:09 AM · Cambridge, Massachusetts
4x 5min threshold, 4x 200m repetition
Predicted Run Type: Marathon Prep

36mph wind gusts... got pushed around the whole time but paces were pretty solid. so aggravating, but the work is done

Distance	Pace	Time	Achievements
7.78 mi	6:44 /mi	52m 25s	4

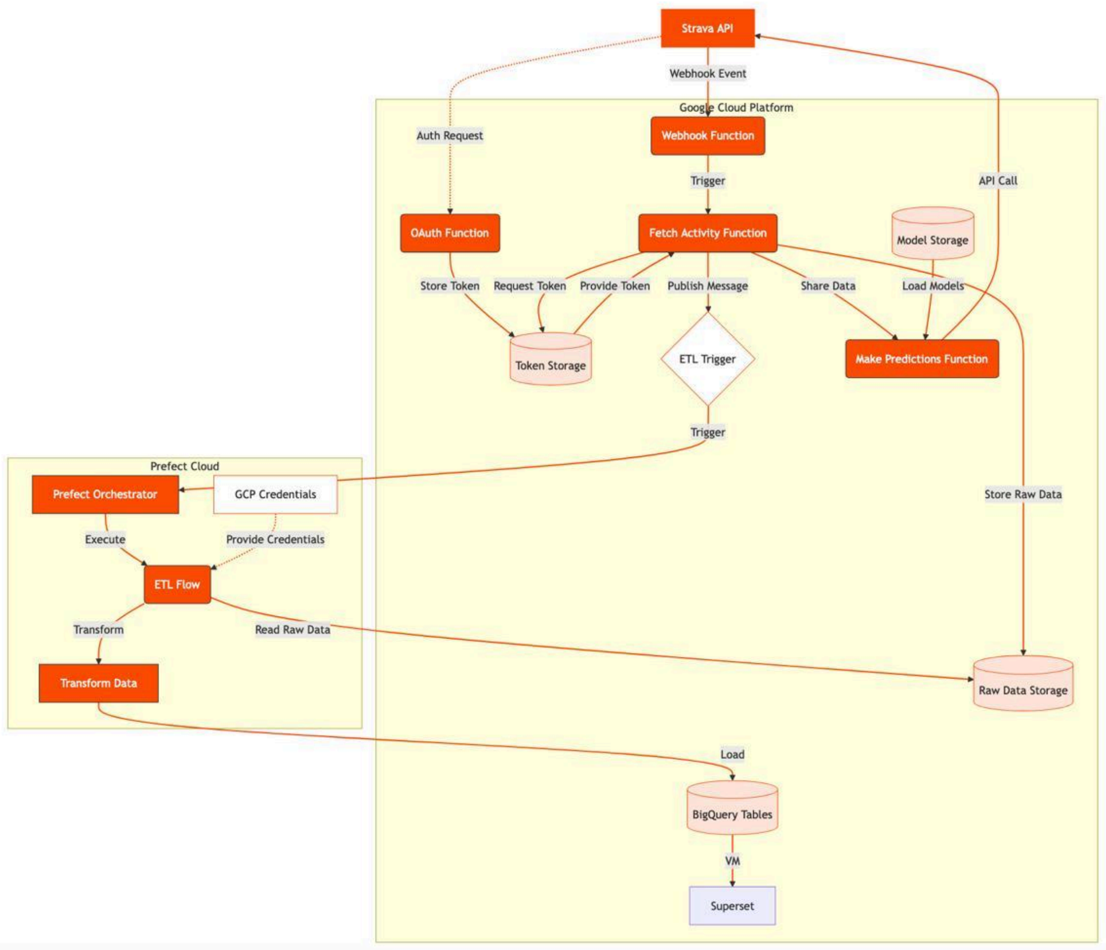
CCW Briggs Loop **PR** (3:22)

Changes Made:

We refined our ETL process to enhance data quality and streamline updates:

- **Adjusted Transformation Stage**: We modified the transformation stage of the ETL pipeline to eliminate unnecessary data processing steps, improving efficiency and reducing processing time.
- **Upsert Logic for ETL Load Stage**: To handle duplicate and updated records, we introduced upsert logic in the load stage of our ETL process. This ensures that when "update" events are received from webhooks, they overwrite existing records instead of creating duplicates, whereas "create" events add new records as usual.
- **Environment Variable for Webhook URL**: We moved the Prefect webhook URL to environment variables, enhancing security and making configuration management easier across different environments.

Updated Pipeline:



Future Steps:

- **Address memory issues for backfilling:** Since the Prefect memory is limiting our backfilling capacity, we are going to backfill the data outside the orchestration tool. Henceforth, the only limitation within the model will be the API limits, but we should be able to backfill the data over the space of a few days using this approach.

- **ML Dashboard**: Integrate machine learning models into an interactive dashboard to display predictions, trends, and insights in real-time. This will allow users to monitor key metrics and make data-driven decisions, with ML outputs directly accessible in a user-friendly format.
- **Generative AI Integration**: Explore the use of Generative AI to enhance data analysis and provide additional insights, such as generating personalized recommendations or performance summaries based on user activity patterns.
- **ML Workflow on Prefect**: Implement and automate machine learning workflows within Prefect, enabling efficient scheduling, monitoring, and error handling to streamline model deployment and maintenance.