Website Visitor Analyzer

Overview:

The project involved developing an application to rank website visitors based on the activities they undertake on a website. The ultimate aim was to be able to identify, with high levels of precision, visitors who were likely to convert. To arrive at this conclusion, a rule-based scoring engine was to be used along with an AI model. Very broadly, Google Analytics was to be used to record all the activities performed by a user on a site. This data was then to be passed through a rule-based engine and an AI model to arrive at a score which was a tentative reflection of the chances of conversion. This outcome can be of very high relevance to companies desiring to improve their website experience, bring more predictability in user segmentation and outreach campaigns.

So the primary functions of the application were to connect to Google Analytics, fetch the data, apply custom rules on the data to arrive at constituent scores and then let the AI model run its algorithms to predict a final score. Besides these, The AI model was also expected to provide a feedback loop suggesting which events and parameters were proving to be more effective in predicting outcomes better, and therefore, influence the constituent scores in the future.

Client details:

Name: Confidential | Industry: Media | Location: UK

Technologies:

Python, FastAPI, PyTorch, Pandas, seaborn, NumPy, Vue.js, Postgres, SQLAlchemy, Docker, Google Cloud Run, Google BigQuery, Google Pub/Sub, Google Cloud Scheduler, Google Cloud Build, Google Secret Manager, Google Container Registry, Pandas, OKTA (SAML), Terraform (IaC), Jest, pytest

Website Visitor Analyzer

Project Description:

The project was mainly categorized into two different phases:

1. Application Overview

We have used Vue js as our frontend framework as per client requirements. And for authentication and authorization purposes we have used **OKTA.** The user will be redirected to the OKTA page to log in and return to the app after successful authentication. This application supports functionality like Creating and Managing users/roles, dashboards and graphs, client onboarding and configurations, Rule Management, Defining Buckets/Segments, etc.

Python was used at the backend with the new Lightweight API framework FastAPI. All the REST APIs were written in FastAPI / secured with Okta Filters and for Database connectivity SQLAlchemy was used as ORM Framework and PGSQL was used for Database.

The AI Model suggests a score based on the events and event parameters of a particular user. It is built on top of Google Analytics data which supports the latest GA4 as well as the legacy GA360. The model is built using PyTorch AI Library and the dataset preparation and messaging is done using Pandas whereas analysis of correlation between events and their parameters is done using the Seaborn library.

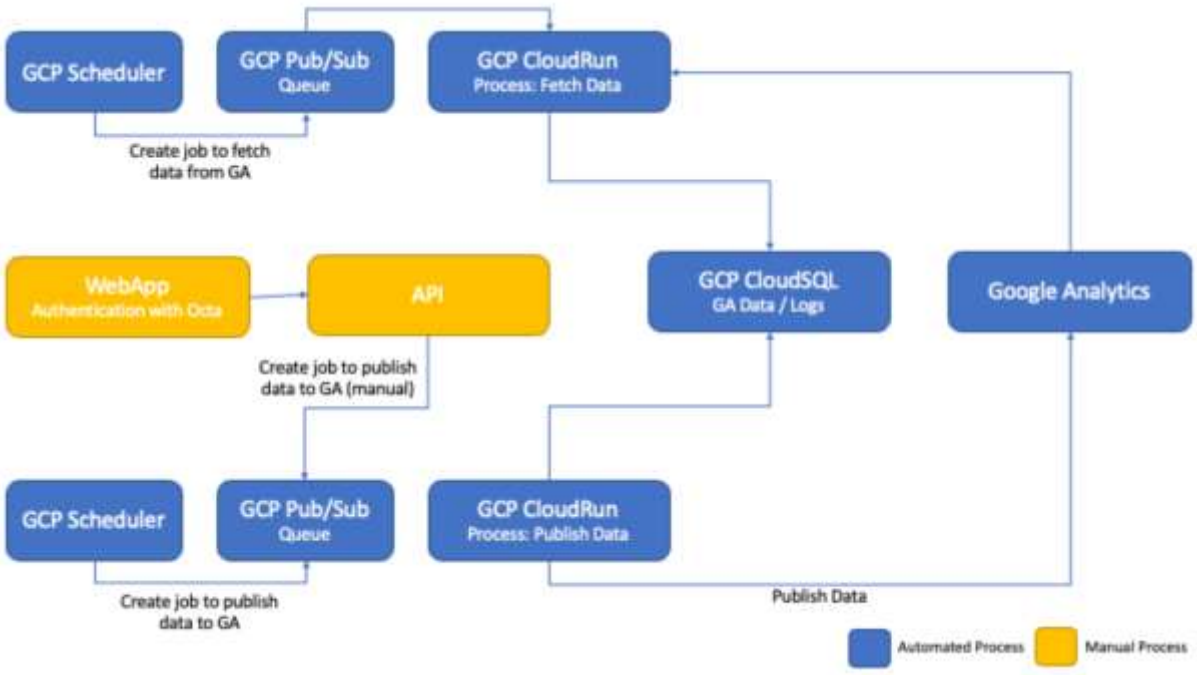
2. Data Import/Export Background Script overview:

This is the core engine of the application which is Multithreaded and supports Batch Executions of data. The primary job of this script is to connect with BigQuery and Fetch Analytics data then apply rules to it, generate segments and scores, and update back to GA. This script can be triggered manually and with a Cloud Scheduler as well. The Script runs on a Serverless Cloud Run container with an invocation using a Pub/Sub message.

Website Visitor Analyzer

Cloud Architecture:

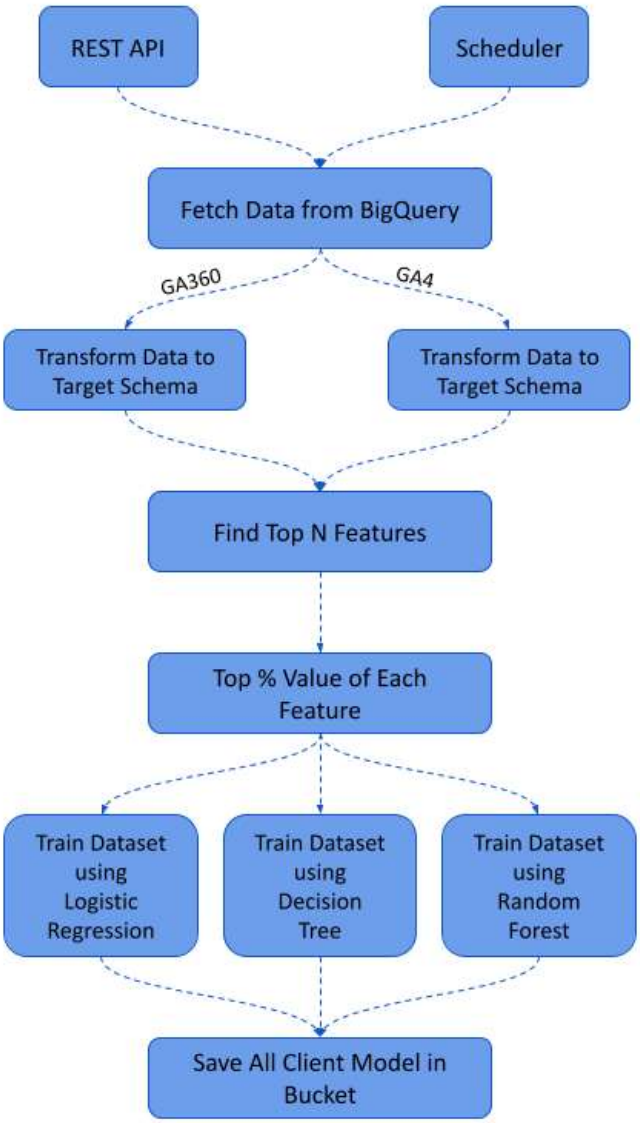
The Application is designed to work on cloud infrastructure with Google Cloud Platform. It uses GCP services like Cloud Run, Cloud Build, Cloud Scheduler, Container Registry, Cloud SQL, Google BigQuery, Google Cloud Pub/Sub, and Google Secret management.



Website Visitor Analyzer

Model Architecture:

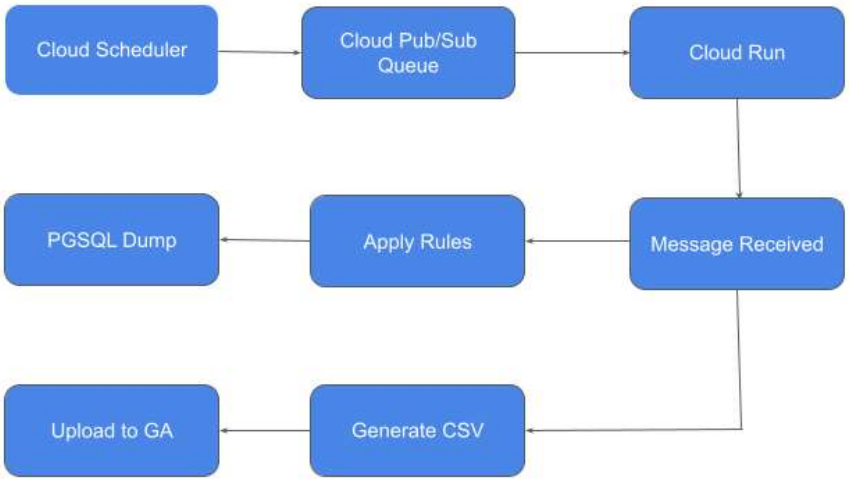
The model architecture involves the preparation of a dataset from GA4/GA360 data, The events and event parameters need to be standardized to a specific format and then all the data has to be analyzed for removal of unnecessary features and errors. The architecture is designed to train the dataset with machine learning algorithms like Logistic Regression, Decision Trees, and Random Forest, and based on the accuracy of each of the 3 models of the algorithm, the final model will be saved on cloud storage for further usage.



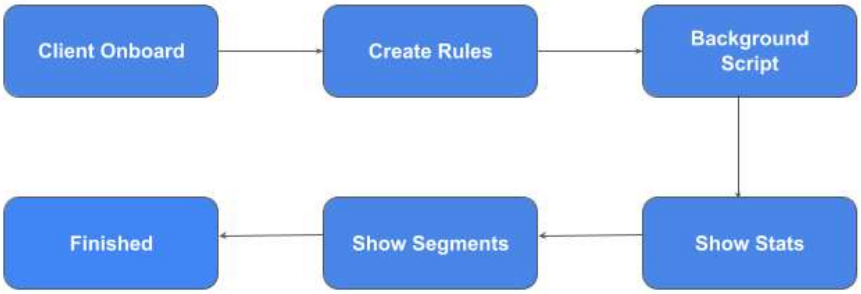
Website Visitor Analyzer

Broadly, the process involves two steps:

1. Data Import/Export script overview:



2. Application Roadmap:



Website Visitor Analyzer

Variables impacting the performance of the application:

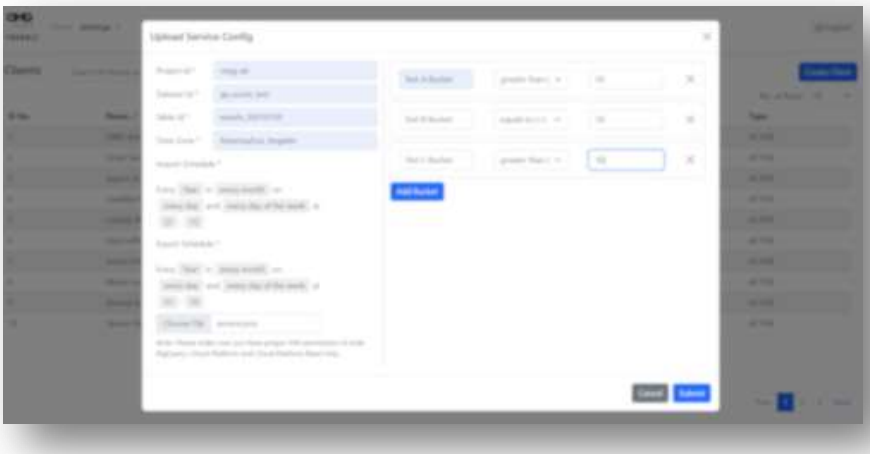
- Huge size of data import from BigQuery (in Millions)
- Application of all the rules and scores to a huge dataset
- GPU type and no of CUDA cores
- GPU Memory
- Size of CPU
- System Memory size
- Network Bandwidth
- Batch Size of each dataset
- Parallel Scoring
- Parallel Pub/sub trigger

Key measures to overcome optimization challenges:

- Splitting the dataset size while querying to a specific limit and use pagination in the query to pass the data so the system can manage the huge dataset.
- Normalization of Google Analytics data of various customers which includes GA4 and GA360 as well.
- Use Python multiprocessing techniques to apply scores to reduce the time taken by the script while scoring one by one.
- Reduce the size of the pub/sub trigger to rectify the parallel error of the system and the cost of computing.
- Handle parallel execution of multiple clients' data.
- Optimize applications to have less memory footprint.

Website Visitor Analyzer

Screenshots



Website Visitor Analyzer