Cloud Computing

Triggered Training of Machine Learning model in Cloud

Project Report

Supervisor:	Prof. Dr. Christian Baun
Submitted by:	Nikhil Bajaj (1392872), Vedant Asawale (1392856) nikhil.bajaj@stud.fra-uas.de, vedant.asawale@stud.fra-uas.de
Submission:	2nd July 2022

Contents

Figures

Tables

1 Introduction

Cloud computing is a big shift from the traditional way businesses think about IT resources. Cloud-computing services cover a vast range of options now, from the basics of storage, networking and processing power, through to natural language processing and artificial intelligence as well as standard office applications. Pretty much any service that doesn't require you to be physically close to the computer hardware that you are using can now be delivered via the cloud. There are lots of service providers for cloud like Amazon Web Services, Microsoft Azure, Google Cloud, IBM cloud etc.Google cloud platform is Google's cloud offering. Similar to AWS and Azure, Google Cloud also offers services in various categories, including compute, storage, identity, security, database, AI and machine learning, virtualization, DevOps and many other technologies.In this project,we focused on Google Cloud and leveraged some of its services.

1.1 Motivation

The pace at which data is generated has exploded over the last few years. Along with that cloud services have proved to be a blessing with so many offerings at disposal for users.The world is moving or has already moved towards cloud computing. Scalability, reliability,availability and affordability are some of the important factors taken into consideration while moving from traditional on-prem infrastructure to cloud.Machine Learning is an area of computer science which requires tremendous computing resources which are most of the times not available with small-scale companies.However,using cloud services, its possible to train ML models without having to set up own infrastructure.In this project we tried to club these 2 cutting edge technologies to create a triggered ML pipeline wherein the pipeline retrains the ML model whenever new data is found. Triggers can be time threshold, data threshold or combination of both. We selected a music recommendation system as our use case as we felt that the underlying data in such a system keeps on changing as new songs are released frequently and the recommendation model should be continuously updated with this new data. However, such an idea of having a triggered ML pipeline can prove to be beneficial in many other business domains.

1.2 The Aim of the work

The aim of using Google cloud platform in our project is to understand the architecture of the cloud computing platform, to understand some of the key components used, test the features and understand some of the services. In our project, we used services such as Pub-Sub,Data-Flow,BigQuery,Google storage bucket, Google Colab and Grafana Cloud.

2 Architecture

The following is the architecture diagram of our project.

2.1 Architecture Diagram

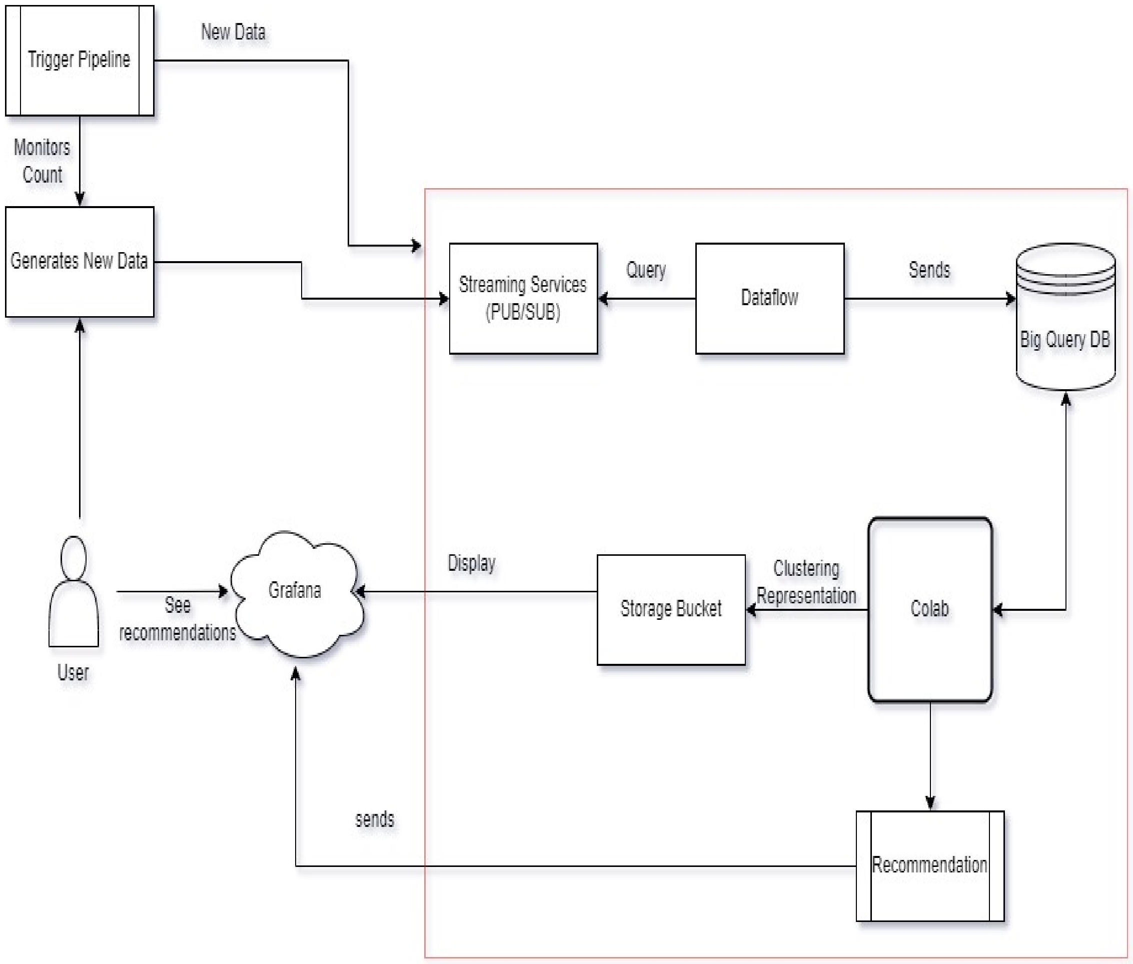


Figure 1 Architecture

2.2 Explanation

The data required by the ML model is initially loaded in the BigQuery tables. The code to train the model resides in Google Colab platform. Any new generated data is published to a Pub-Sub topic. A Data-Flow job is configured which monitors data published to the Pub-Sub topic and writes this new data to BigQuery tables. This data from BigQuery tables is continuously monitored by querying it after every specific interval. In case, if new data is found, the ML pipeline is triggered and Unsupervised ML model for music recommendation is retrained.The results of the new model are first stored in Google storage bucket and integrated with Grafana cloud for the user to monitor.

3 Fundamental

3.1 Google Cloud

Google Cloud [5] is a cloud computing service suite provided by Google. It provides various services like machine learning, data storage and analytics. The registration needs a credit card or bank details but Google does not debit money automatically. This bank details verification is made to stop the misuse of the resources by the bots and dummy account holders. The Google Cloud Platform (GCP) [3] is the public cloud platform from Google that serves as a collection of IaaS and PaaS services to provide consumers a highly stable and reliable on-demand service. Google cloud services are well established for the modern developers to develop stable applications and publish them across the globe. GCP provides many features as a service in the Big Data analytics, artificial intelligence, and containerization spaces. The GCP IaaS resources also have a series of physical hardware infrastructure like computers, hard disk drives, solid state drives, and networking that is contained within Google's globally distributed data centers. There are a lots of tools or features that are used to build up the GCP as a product or suite to the users.

3.2 Machine Learning

Machine Learning [7] is a part of artificial intelligence, where the problem is solved by considering the historical examples or previous examples. In contrast to artificial intelligence is machine learning. This is capable of finding hidden patterns within the data.The practical implementation of machine learning is done by algorithms.Usually, different machine learning algorithms can be divided into three categories: Supervised learning, unsupervised learning and reinforcement learning.

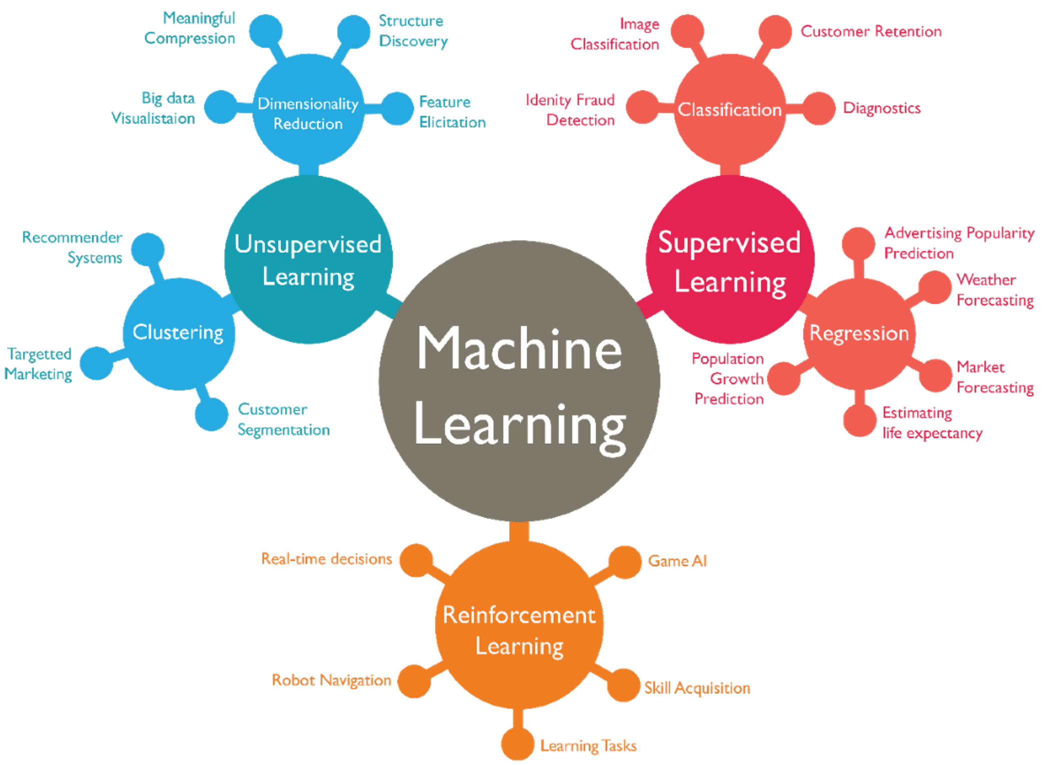


Figure 2 Types of Machine learning (https://techyexplorer.com)

3.2.1 Supervised Learning

Supervised Learning [7] is a method that attempts to estimate a relationship between input parameters (independent parameters) and output (target). The discovered relationship is called a mathematical method. Usually, the model describes phenomena hidden in the data set. Thus, the output of a data set can be predicted when the output is unknown.It is useful to distinguish between two main supervised models: Classification models (classification) and Regression models. The regression models map the input space in a real value domain. For example, a regressor may predict the demand for a particular product based on its characteristics. On the other hand, classifications map the input space into predefined classes.

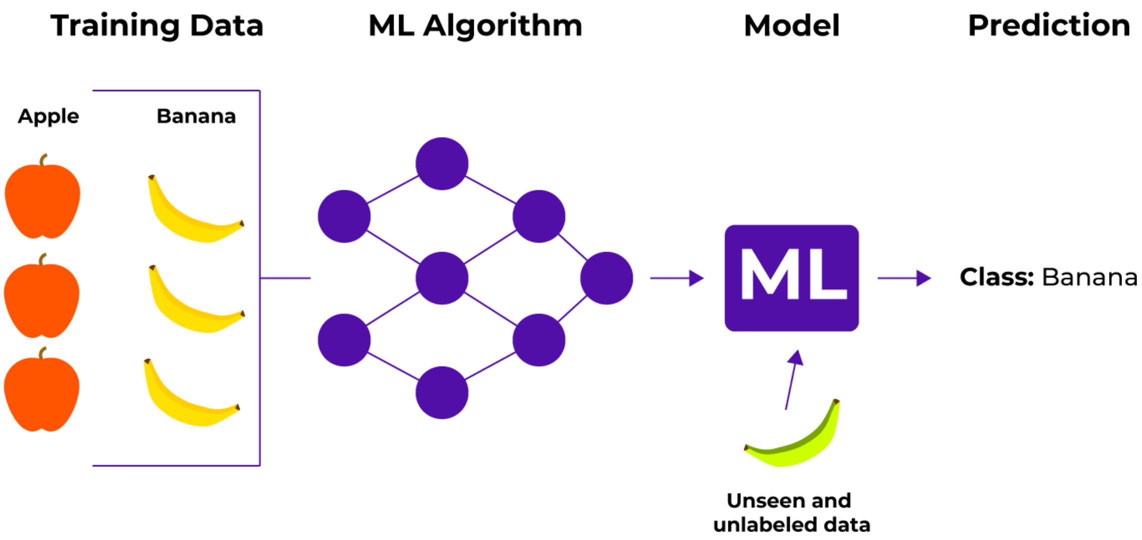


Figure 3 Demonstration of Supervised learning

3.3 Unsupervised Learning

In some pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering [6], or to determine how the data is distributed in the space, known as density estimation. To put forward in simpler terms, for a n-sampled space X1 to Xn, true class labels are not provided for each sample, hence known as learning without teacher.

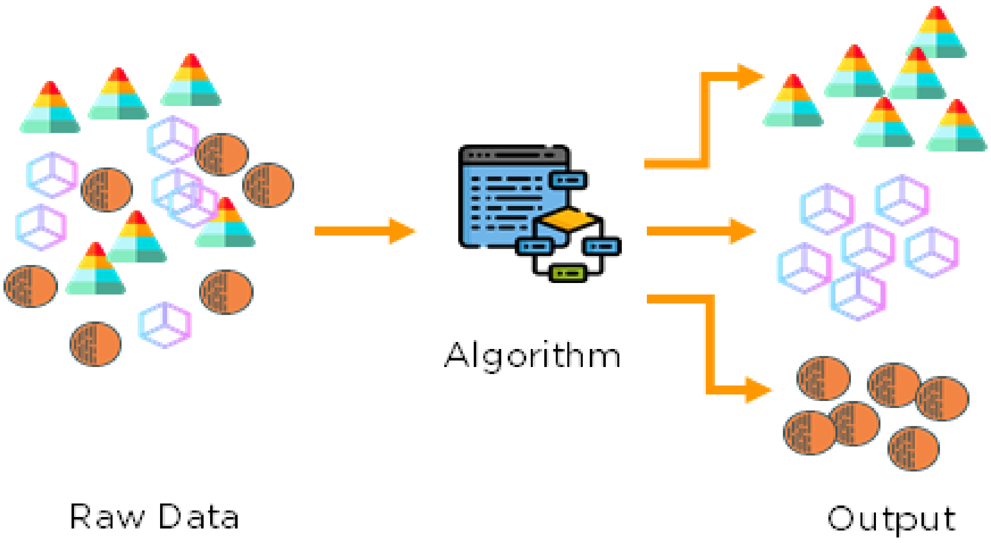


Figure 4 Demonstration of Unsupervised learning (https://medium.com)

3.3.1 K Means Clustering

K-Means [6] is one of the simplest unsupervised learning algorithms that solves the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). The main idea is to define k centres, one for each cluster. These centroids should be placed in a smart way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

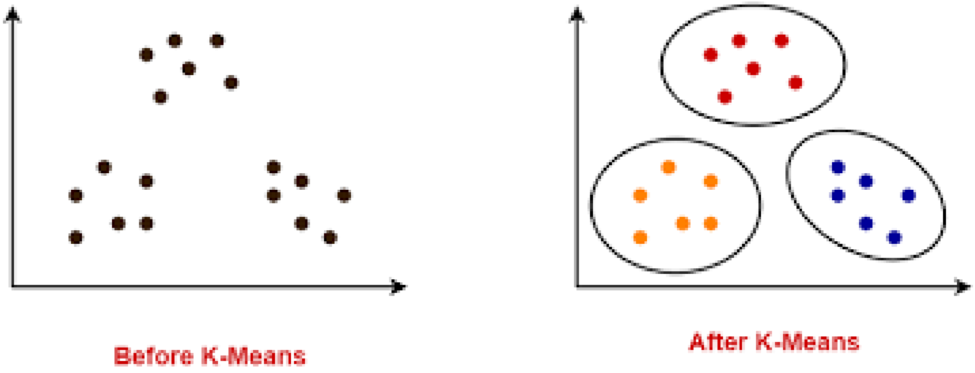


Figure 5 K-Means Clustering Demonstration

3.4 Recommendation Systems

Recommendation engines [1] are a subclass of machine learning which generally deal with ranking or rating products / users. Loosely defined, a Recommender System [8] is a system which predicts ratings a user might give to a specific item. These predictions will then be ranked and returned back to the user.

They're used by various large name companies like Google, Instagram, Spotify, Amazon, Reddit, Netflix etc. often to increase engagement with users and the platform. For example, Spotify would recommend songs similar to the ones you've repeatedly listened to or liked so that you can continue using their platform to listen to music.

3.4.1 Content Based Recommendation

Content based systems generate recommendations based on the users preferences and profile. They try to match users to items which they have liked previously. The level of similarity between items is generally established based on attributes of items liked by the user. Unlike most collaborative filtering models which leverage ratings between target user and other users, content based models focus on the ratings provided by the target user themselves. In essence, the content based approach leverages different sources of data to generate recommendations.

4 Cloud Services Used

Our project revolves around setting up a triggered Machine Learning[1] model for music recommendation system using multiple cloud services.For our project we used the free trial offered by google cloud. The free trial is valid for first time Google Cloud[2] users only. Registration requires country selection and payment profile creation. Payment profile can be created with your name, address and credit card or bank account. Once the registration is done we get a 90 days free trial with 300 USD credit. The free trial subscription ends after 90 days or if the credit is over before 90 days. Once the credit is over we have to purchase more credit or pay according to your usage. We have to pay for what we use and GCP has a price calculator for estimating our needs and usage.

Services	**Description**
Compute	App Engine - Infrastructure as a Service to run Microsoft Windows and Linux virtual machines.
Big Data (Big Query)	Big Query - Scalable, managed enterprise data warehouse for analytics. Cloud Pub/Sub - Scalable event ingestion service based on message queues. Cloud Dataflow - Managed service based on Apache Beam for stream and batch data processing.
Management Tools	Cloud Console - Web interface to manage Google Cloud Platform resources.
Storage and Databases	Cloud Storage - Object storage with integrated edge caching to store unstructured data.

Table 1 Table of the Google Cloud services used

4.1 Pub-Sub Service

Pub/Sub allows services to communicate asynchronously, with latencies on the order of 100 milliseconds.

Pub/Sub is used for streaming analytics and data integration pipelines to ingest and distribute data. It’s equally effective as a messaging-oriented middleware for service integration or as a queue to parallelize tasks.

Pub/Sub enables you to create systems of event producers and consumers, called publishers and subscribers. Publishers communicate with subscribers asynchronously by broadcasting events, rather than by synchronous remote procedure calls (RPCs).

Publishers send events to the Pub/Sub service, without regard to how or when these events are to be processed. Pub/Sub then delivers events to all the services that react to them. In systems communicating through RPCs, publishers must wait for subscribers to receive the data. However, the asynchronous integration in Pub/Sub increases the flexibility and robustness of the overall system.

4.2 Dataflow Service

Google Cloud Dataflow is a cloud-based data processing service for both batch and real-time data streaming applications. It enables developers to set up processing pipelines for integrating, preparing and analyzing large data sets, such as those found in Web analytics or big data analytics applications.

Cloud Dataflow can take data in publish-and-subscribe mode from Google Cloud Pub-/Sub middleware feeds or, in batch mode, from any database or file system. It agnostically handles data of varying sizes and structures using a format called PCollections, which is short for "parallel collections." The Google Cloud Dataflow service also includes a library of parallel transforms, or PTransforms, which allow high-level programming of often-repeated tasks using basic templates; in addition, it supports developer customization of data transformations.

4.3 BigQuery Table

Google BigQuery is a serverless, highly scalable data warehouse that comes with a built-in query engine. The query engine is capable of running SQL queries on terabytes of data in a matter of seconds, and petabytes in only minutes. You get this performance without having to manage any infrastructure and without having to create or rebuild index

4.4 Google Storage Bucket

Google Cloud Storage is a RESTful online file storage web service for storing and accessing data on Google Cloud Platform infrastructure. The service combines the performance and scalability of Google's cloud with advanced security and sharing capabilities. It is an Infrastructure as a Service (IaaS)

4.5 Google Colab

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

4.6 Grafana Cloud

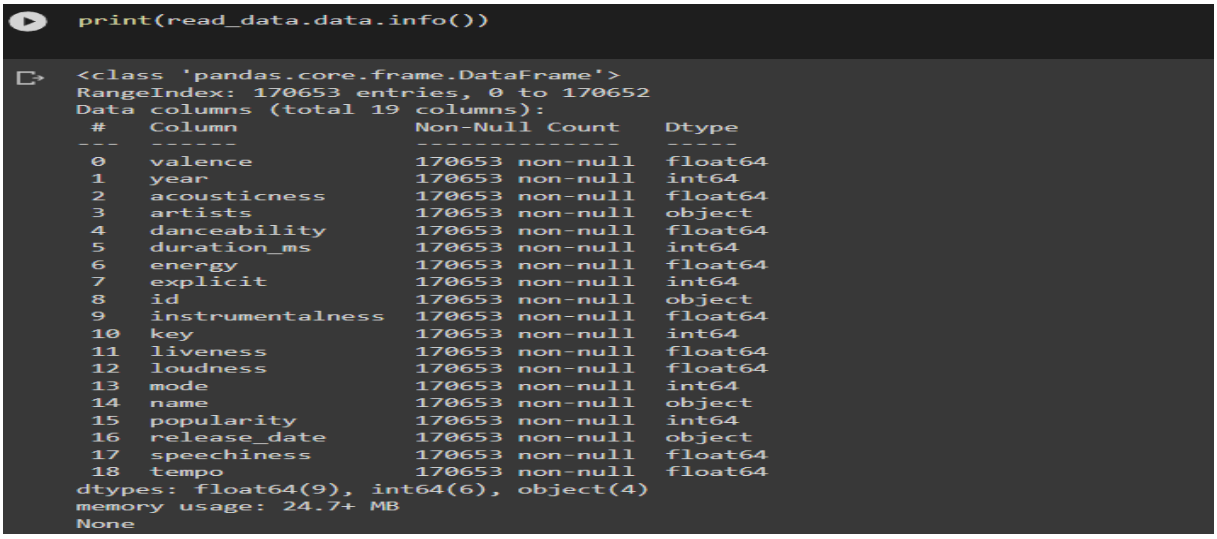
Grafana Cloud[4] is a highly available, performant, and scalable observability platform for your applications and infrastructure. It provides a centralized view over all of your observability data, whether the data lives in Grafana Cloud Metrics services or in your own bare-metal and cloud environments. With native support for many popular data sources like Prometheus, Elasticsearch, and Amazon CloudWatch, all you have to do to begin creating dashboards and querying metrics data is to configure data sources in Grafana Cloud.

5 Implementation Details

5.1 About Data

We have taken data from spotify which includes various features of songs.

The features of the songs are :
['valence', 'year', 'acousticness', 'danceability', 'durationms', 'energy', 'explicit', 'instrumentalness', 'key', 'liveness', 'loudness', 'mode', 'name', 'releasedate', 'popularity', 'speechiness', 'tempo']



```
print(read_data.data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170653 entries, 0 to 170652
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   valence           170653 non-null  float64
 1   year              170653 non-null  int64
 2   acousticness      170653 non-null  float64
 3   artists           170653 non-null  object
 4   danceability      170653 non-null  float64
 5   duration_ms       170653 non-null  int64
 6   energy            170653 non-null  float64
 7   explicit          170653 non-null  int64
 8   id                170653 non-null  object
 9   instrumentalness  170653 non-null  float64
 10  key               170653 non-null  int64
 11  liveness          170653 non-null  float64
 12  loudness          170653 non-null  float64
 13  mode              170653 non-null  int64
 14  name              170653 non-null  object
 15  popularity        170653 non-null  int64
 16  release_date      170653 non-null  object
 17  speechiness       170653 non-null  float64
 18  tempo             170653 non-null  float64
dtypes: float64(9), int64(6), object(4)
memory usage: 24.7+ MB
None
```

Figure 6 Song Data CSV1

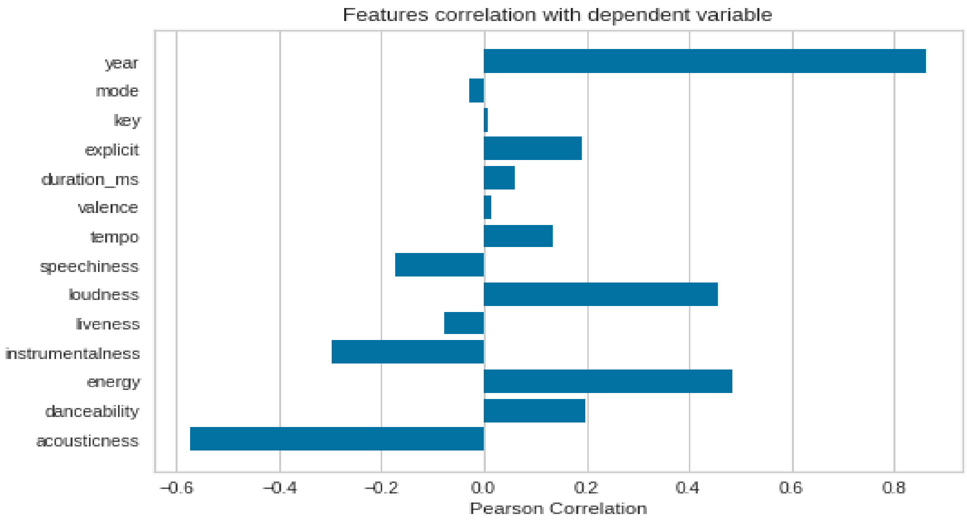
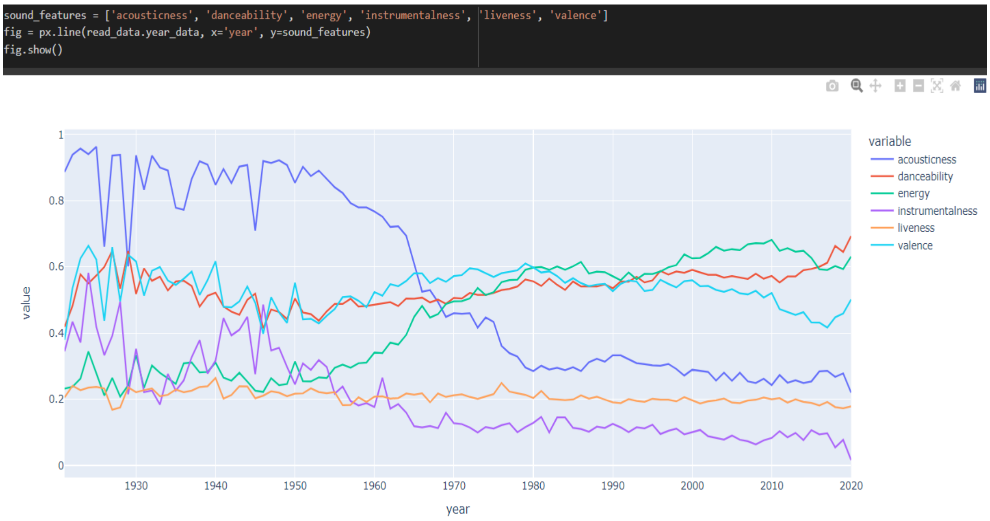


Figure 7 Correlation of Features

This is representation of the data given



```
sound_features = ['acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'valence']
fig = px.line(read_data.year_data, x='year', y=sound_features)
fig.show()
```

Figure 8 Value of Variables over the Year

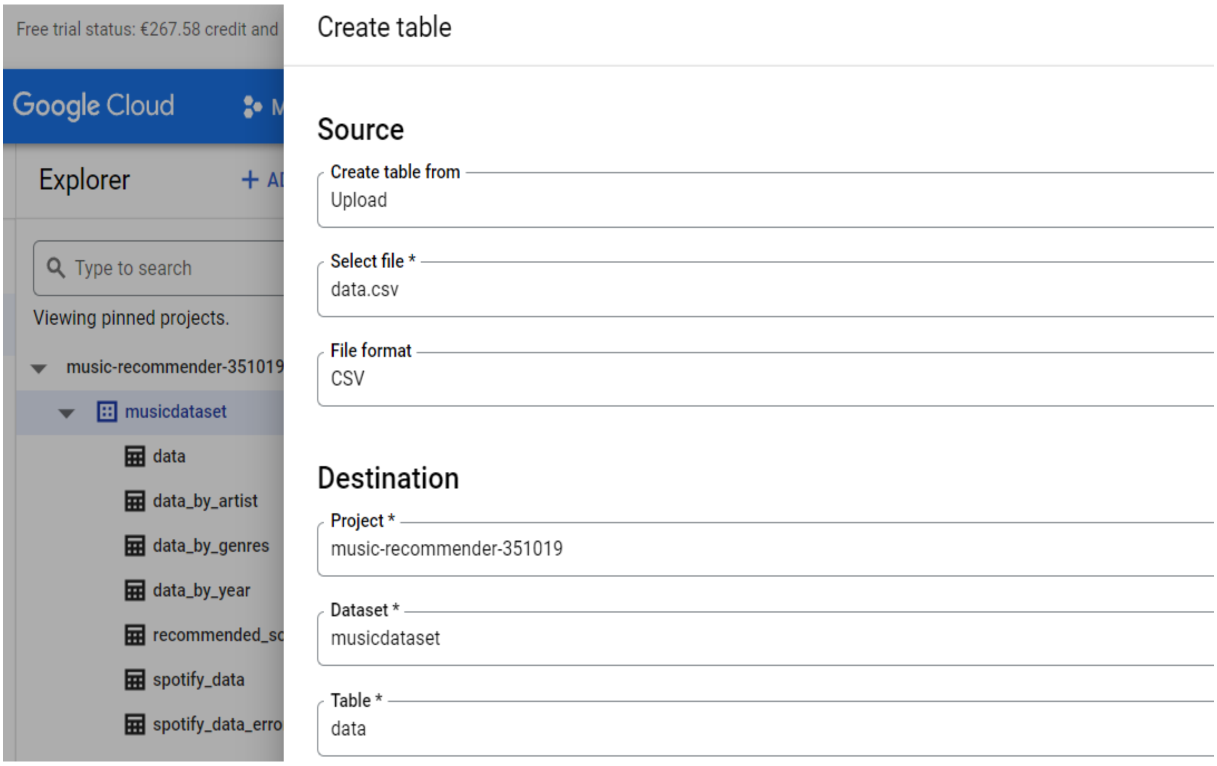
5.2 Google Cloud: Creating BigQuery Table

We used [2]Google BigQuery tables for storing and staging the datasets required for training the ML model.

5.2.1 Steps

(1) The first step is to create a Dataset which acts as a container for all your tables.

(2) For creating tables ,Bigquery provides us multiple options,we chose the "Upload option",which allows us to create a schema based on the data file which we upload.



Create table

Source

Create table from: Upload

Select file *: data.csv

File format: CSV

Destination

Project *: music-recommender-351019

Dataset *: musicdataset

Table *: data

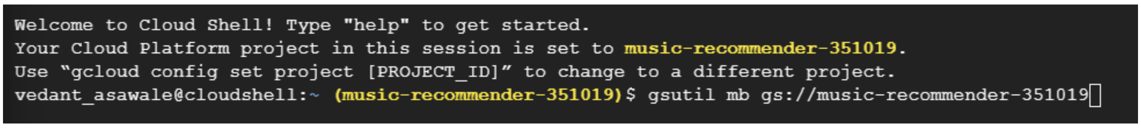
Figure 9 Creating BigQuery table

5.3 Google Cloud Storage: Creating Google Storage Bucket

A google storage bucket is used in our project as its required by the DataFlow job as a temporary storage location.Its also used later to store the visualisation output of our ML model to be rendered in Grafana.

5.3.1 Steps

(1) Storage bucket can be created via user interface or via cloud shell command line.We used the cloud shell to create the storage bucket.



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to music-recommender-351019.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
vedant_asawale@cloudshell:~ (music-recommender-351019)$ gsutil mb gs://music-recommender-351019
```

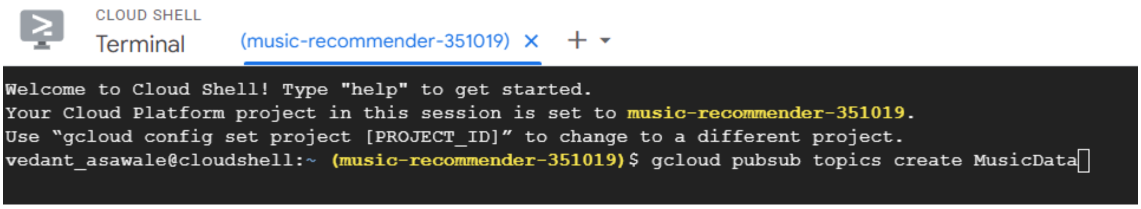
Figure 10 Creating Storage Bucket

5.4 Setting up ETL Pipeline using PubSub and DataFlow

An ETL pipeline has to be created so that new music data can be loaded to BigQuery in realtime.For this purpose,we first create a topic in pubsub and then create a dataflow job which reads data from the Pub-Sub and writes it to BigQuery.

5.4.1 Steps

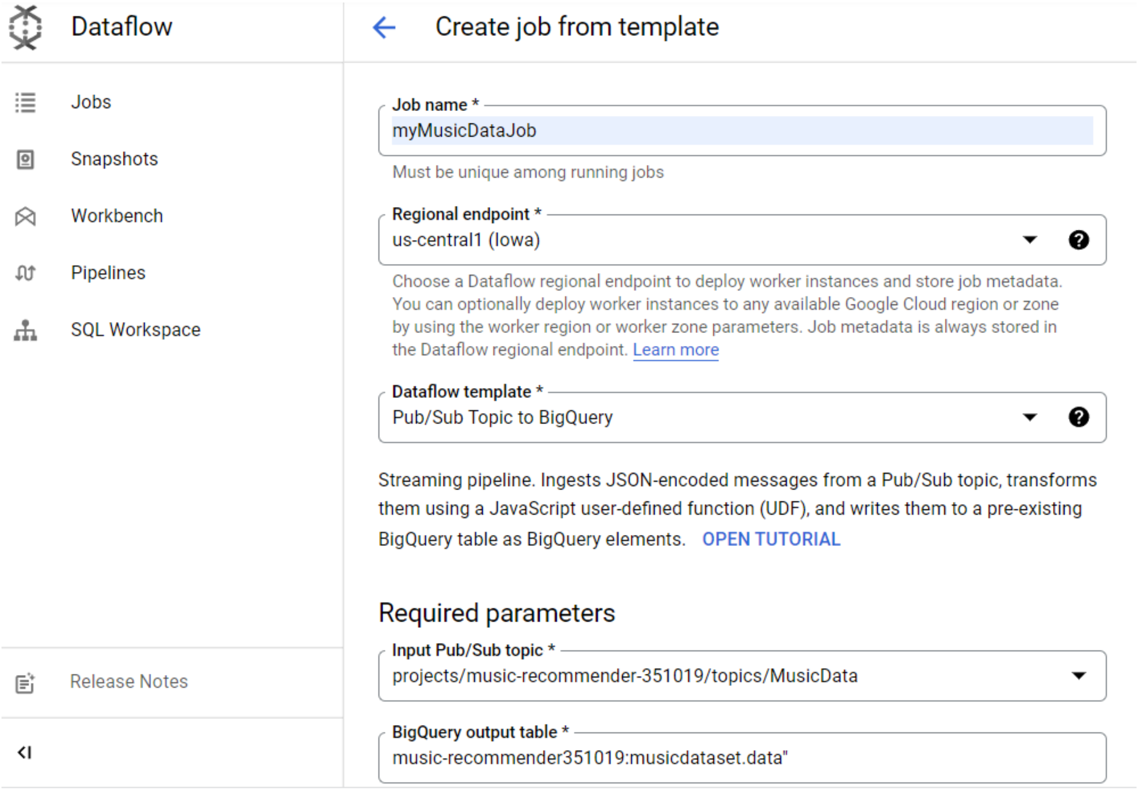
(1) The first step is to create a Pub-Sub Topic.



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to music-recommender-351019.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
vedant_asawale@cloudshell:~ (music-recommender-351019)$ gcloud pubsub topics create MusicData
```

Figure 11 Creating Pub-Sub Topic

(2) For creating the Dataflow job,we can use the PubSub-BigQuery template and provide it the pub-sub topic name , BigQuery table identifier and the storage bucket location for temporary storage.



Job name *
myMusicDataJob
Must be unique among running jobs

Regional endpoint *
us-central1 (Iowa)
Choose a Dataflow regional endpoint to deploy worker instances and store job metadata. You can optionally deploy worker instances to any available Google Cloud region or zone by using the worker region or worker zone parameters. Job metadata is always stored in the Dataflow regional endpoint. Learn more

Dataflow template *
Pub/Sub Topic to BigQuery

Streaming pipeline. Ingests JSON-encoded messages from a Pub/Sub topic, transforms them using a JavaScript user-defined function (UDF), and writes them to a pre-existing BigQuery table as BigQuery elements. OPEN TUTORIAL

Required parameters

Input Pub/Sub topic *
projects/music-recommender-351019/topics/MusicData

BigQuery output table *
music-recommender351019:musicdataset.data"

Figure 12 Creating Data-Flow Job

5.5 Creating Service Account and generating key

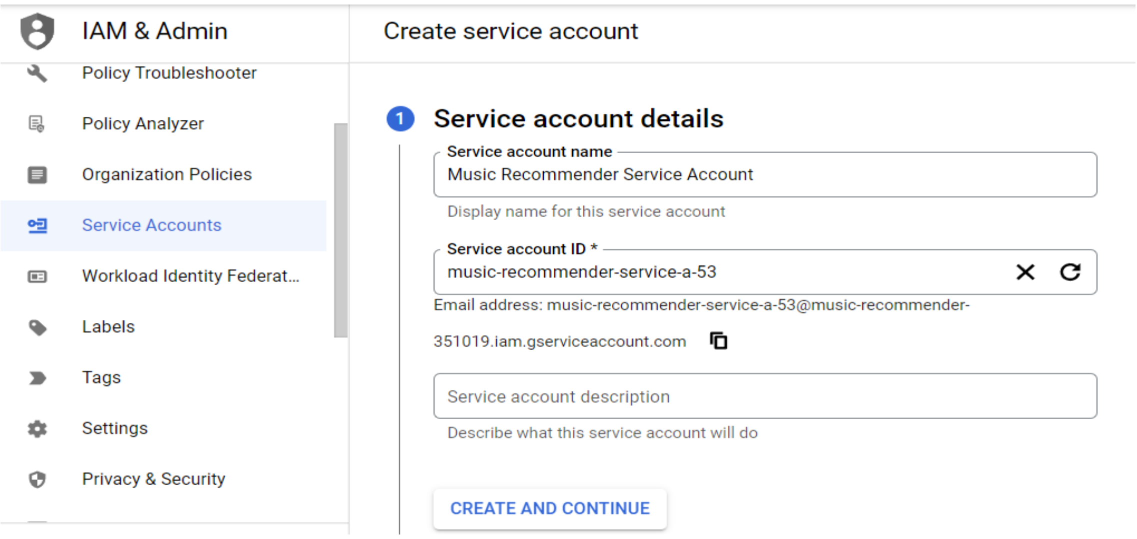
A GCP service account is a type of Google account proposed to interact with non-human users that requires authentication to be confirmed in order to fetch information over Google APIs.In other words,we require the service account so that we can use the Google APIs in our application.In our project,we needed the service account for publishing data to Pub-Sub,interacting with BigQuery tables and pushing data to Google storage bucket.

5.5.1 Steps

(1) The first step is to create a Service Account which can be done by navigating to IAM and Admin product panel ,selecting service accounts and clicking on "Create Service Account".

(2) The next step is to add the appropriate roles/permissions to the Service Account such as Pub-Sub Admin,BigQuery Admin etc so that we can use the respective APIs via the Service Account.

(3) The final step is to generate and download the key which can be used in our code for authentication purpose.



Service account details

Service account name: Music Recommender Service Account

Display name for this service account

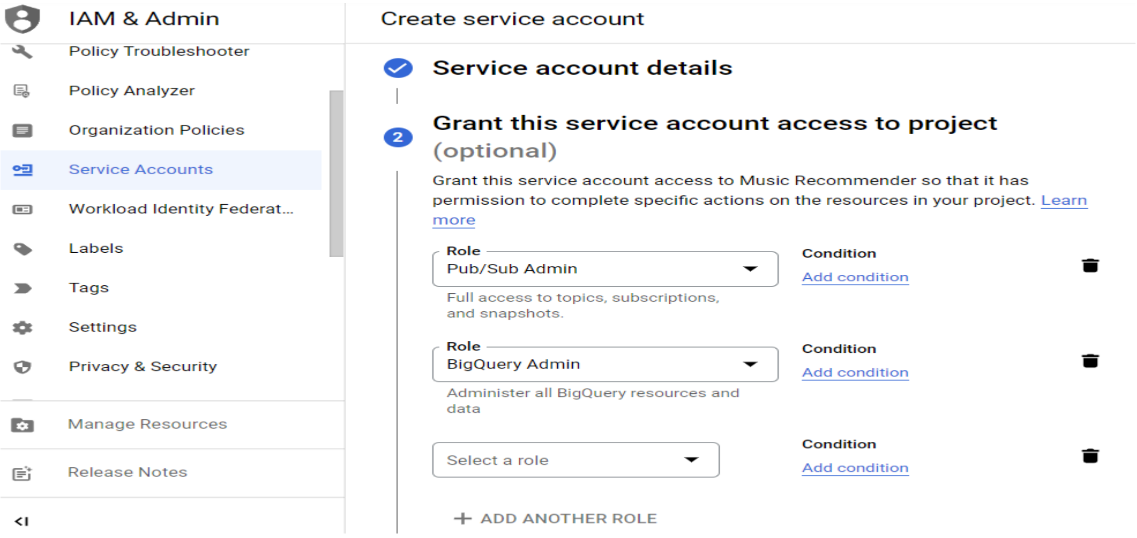
Service account ID *: music-recommender-service-a-53

Email address: music-recommender-service-a-53@music-recommender-351019.iam.gserviceaccount.com

Service account description

Describe what this service account will do

Figure 13 Creating Service Account



Service account details

Grant this service account access to project (optional)

Grant this service account access to Music Recommender so that it has permission to complete specific actions on the resources in your project. Learn more

Role: Pub/Sub Admin — Condition

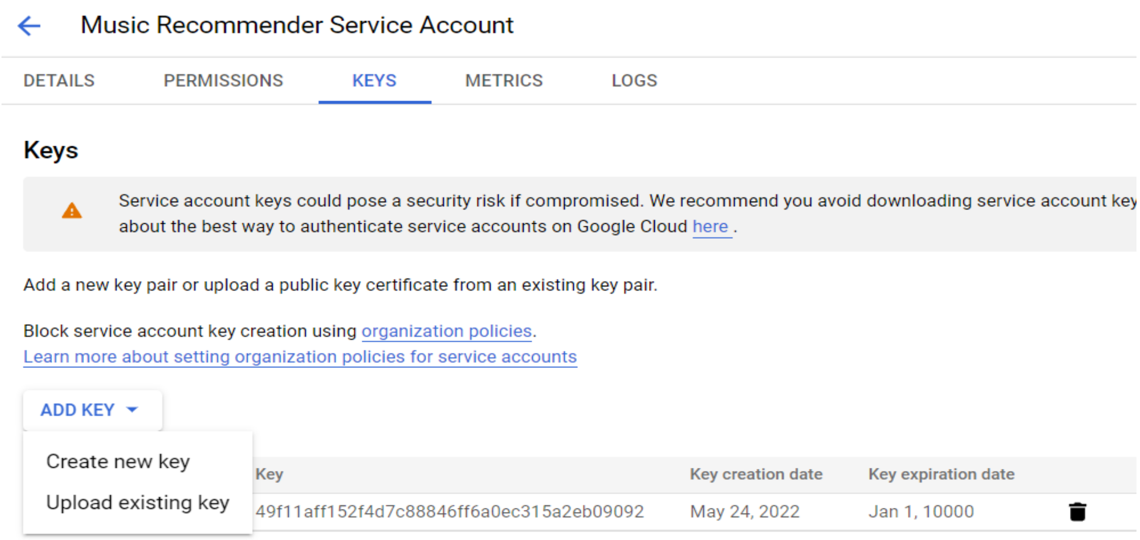
Full access to topics, subscriptions, and snapshots.

Role: BigQuery Admin — Condition

Administer all BigQuery resources and data

Condition

Figure 14 Grant Roles to Service Account



Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account key
about the best way to authenticate service accounts on Google Cloud here .

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

Key	Key creation date	Key expiration date
49f11aff152f4d7c88846ff6a0ec315a2eb09092	May 24, 2022	Jan 1, 10000

Figure 15 Generate Service Account Key

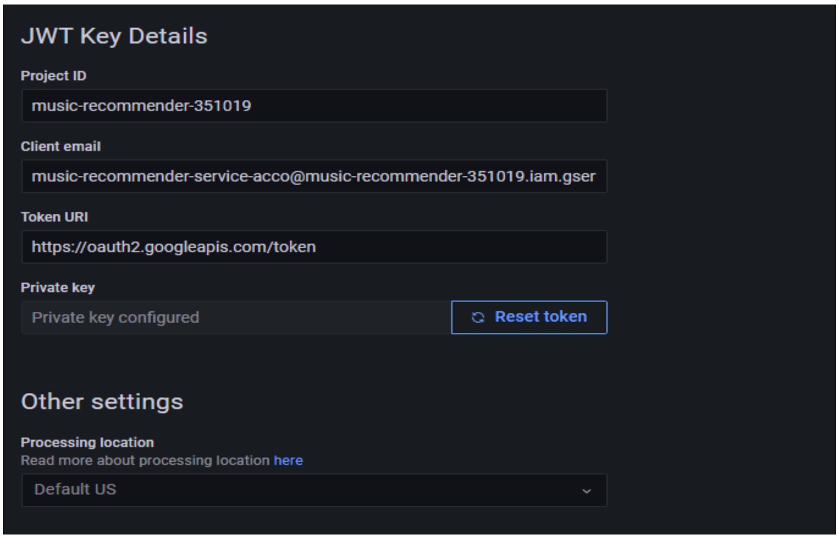
5.6 Setting up Grafana

5.6.1 Getting Started

1. Navigate to https://grafana.com/products/cloud/
2. Click Start for free on the banner.
3. Follow the instructions to finish setting up your account and access the Cloud Portal.
4. Choose an URL for your dashboard. 5. Add other users if required

5.6.2 Adding Data Source

1. Install Plugin for Big Query
2. Configure credentials by uploading jwt token



JWT Key Details

Project ID: music-recommender-351019

Client email: music-recommender-service-acco@music-recommender-351019.iam.gser

Token URI: https://oauth2.googleapis.com/token

Private key: Private key configured

Other settings

Processing location

Read more about processing location here

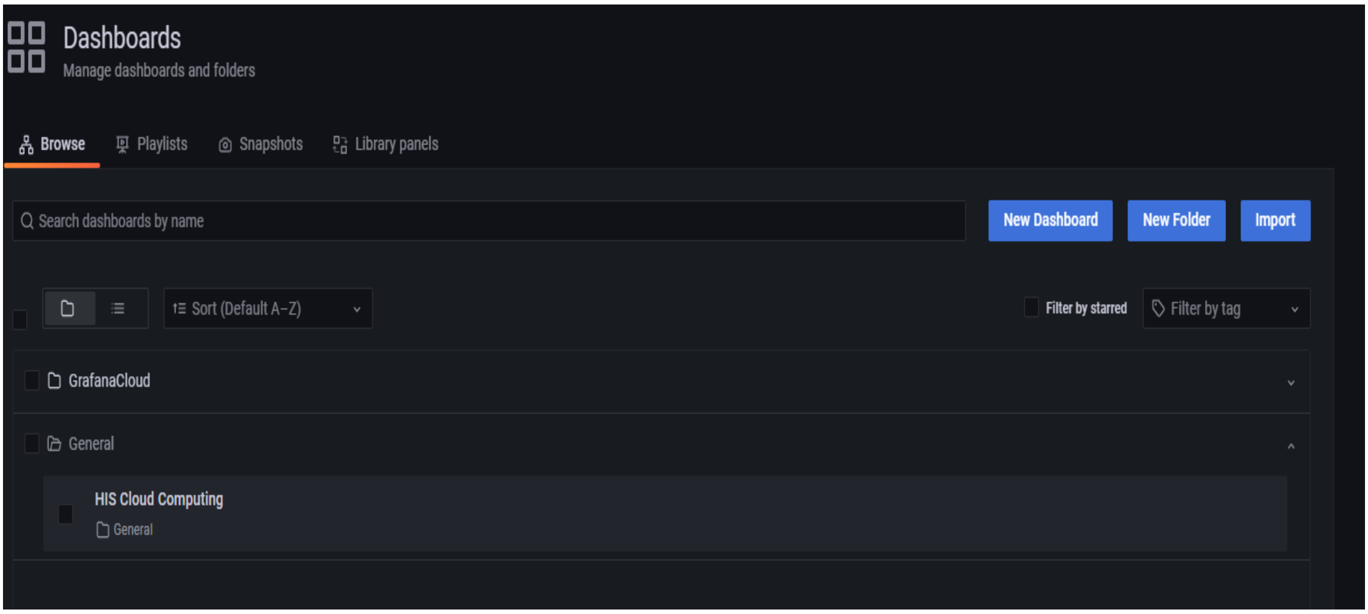
Default US

Figure 16 JWT Token Upload

5.6.3 Dashboard

1. Create a new dashboard

2. Add various panels and rows



Dashboards

Manage dashboards and folders

GrafanaCloud

General

HIS Cloud Computing

General

Figure 17 Creating a Dashoard

5.6.4 Display Data

1. Add a panel

2. Selection correct visualization of the data

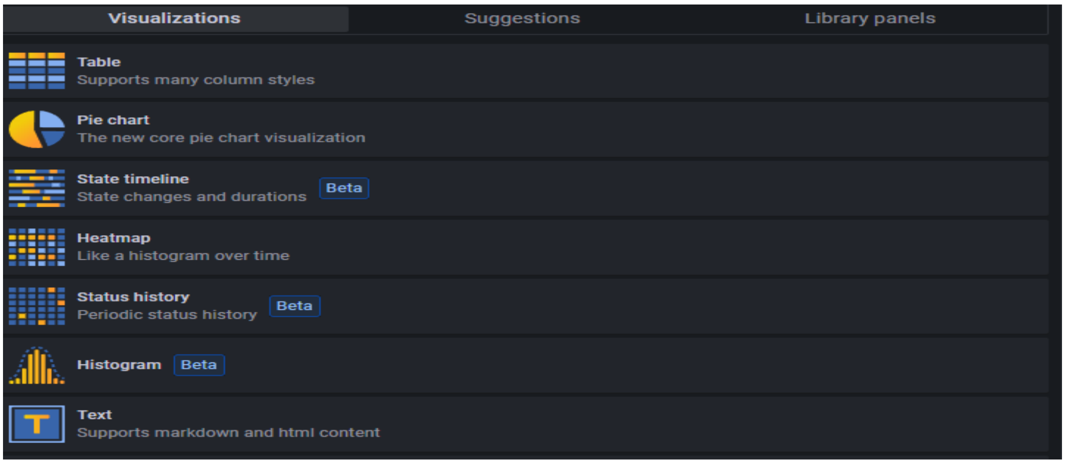


Table
Supports many column styles

Pie chart
The new core pie chart visualization

State timeline Beta
State changes and durations

Heatmap
Like a histogram over time

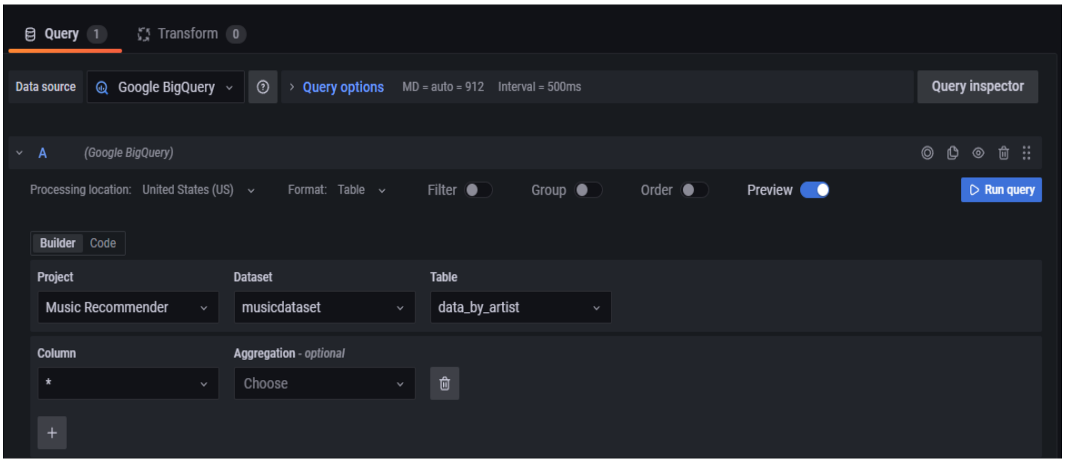
Status history Beta
Periodic status history

Histogram Beta

Text
Supports markdown and html content

Figure 18 Choose a visualization

3. Build the query by selecting Project, dataset, Table and columns



Data source: Google BigQuery

A (Google BigQuery)

Processing location: United States (US)

Format: Table

Project: Music Recommender

Dataset: musicdataset

Table: data_by_artist

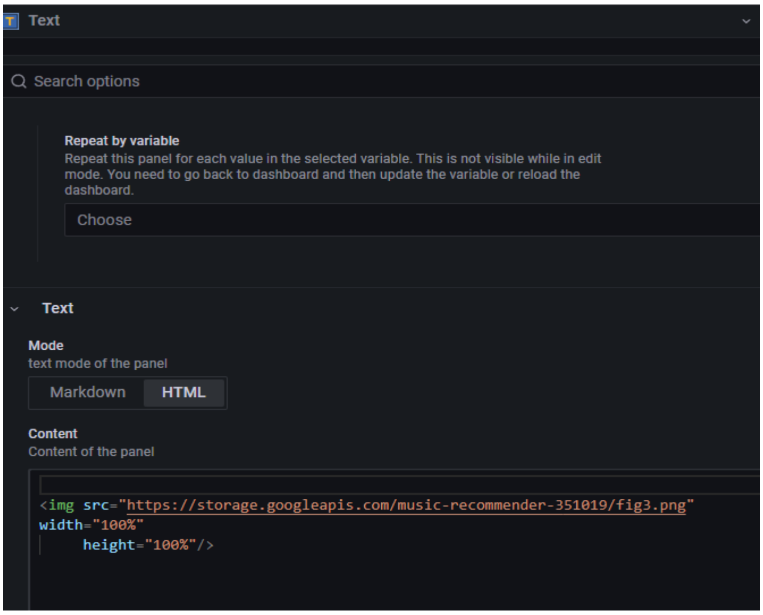
Column: *

Aggregation - *optional*: Choose

Figure 19 Build Query

4. Show the recommendations

5. To obtain Clustering Image from Bucket, add a panel of text. Add link of the Google Bucket to html src tag and image will be displayed



Text

Search options

Repeat by variable
Repeat this panel for each value in the selected variable. This is not visible while in edit mode. You need to go back to dashboard and then update the variable or reload the dashboard.

Choose

Text

Mode
text mode of the panel

Markdown **HTML**

Content
Content of the panel

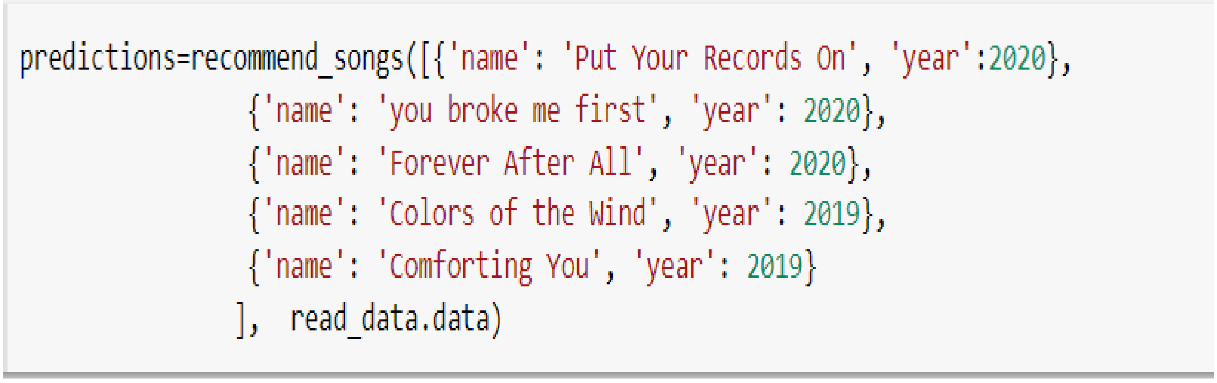
```
<img src="https://storage.googleapis.com/music-recommender-351019/fig3.png"
width="100%"
    height="100%"/>
```

Figure 20 Display Clustering Image

5.7 Code Sequence

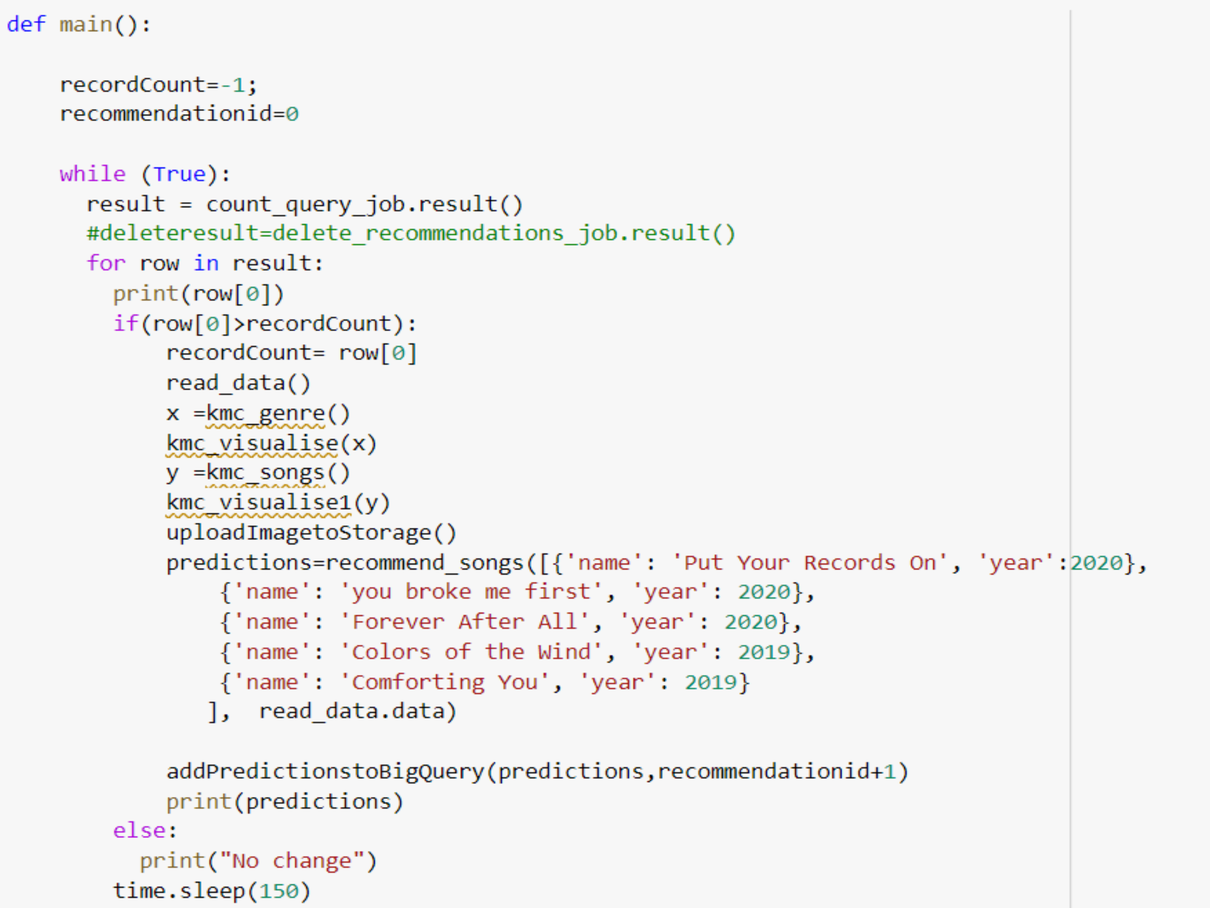
Our code is running in Google Colab. We have created a pipeline so that all the steps are performed in a single go. This pipeline has triggers attached. For now we have attached time trigger to it.
Working: Once the new data crosses a certain threshold, the pipeline or main function is called. This pipeline is checked every certain interval (depends on business use case).
In pipeline, the complete data is clustered using K-means. Then new recommendations are generated and shown on grafana to user.

Recommendations: The recommendations shown are content based which means we provide the history of the songs liked or listened by the user and based on the data, similar songs are recommended. The similar or recommended songs are figured out by using cosine similarity with the clusters obtained from K-means.



```
predictions=recommend_songs([{'name': 'Put Your Records On', 'year':2020},
                 {'name': 'you broke me first', 'year': 2020},
                 {'name': 'Forever After All', 'year': 2020},
                 {'name': 'Colors of the Wind', 'year': 2019},
                 {'name': 'Comforting You', 'year': 2019}
                ],  read_data.data)
```

Figure 21 Recommendation example



```
def main():

    recordCount=-1;
    recommendationid=0

    while (True):
      result = count_query_job.result()
      #deleteresult=delete_recommendations_job.result()
      for row in result:
        print(row[0])
        if(row[0]>recordCount):
            recordCount= row[0]
            read_data()
            x =kmc_genre()
            kmc_visualise(x)
            y =kmc_songs()
            kmc_visualise1(y)
            uploadImagetoStorage()
            predictions=recommend_songs([{'name': 'Put Your Records On', 'year':2020},
                {'name': 'you broke me first', 'year': 2020},
                {'name': 'Forever After All', 'year': 2020},
                {'name': 'Colors of the Wind', 'year': 2019},
                {'name': 'Comforting You', 'year': 2019}
               ],  read_data.data)

            addPredictionstoBigQuery(predictions,recommendationid+1)
            print(predictions)
        else:
          print("No change")
        time.sleep(150)
```

Figure 22 Code and Pipeline

6 Evaluation and Results

As shown in Figure 21 and Figure 22. Some songs from user (user data) are passed to the model. The main function is pipeline with the time trigger of 150s. Here after every 150seconds (for demonstration purpose), BigQuery table count is checked. Incase there are new songs in the database, code transfer to if loop and the model is retrained again and new recommendations are generated. We are using KMeans unsupervised learning algorithm for generating clusters, then using the user data and cosine similarity to recommend new songs to the user. Refer the Figure 23

The Grafana cloud https://hiscloud.grafana.net/ dashboard is being used to monitor the model stats, recommendations and cluster image generated. Refer the Figure 23

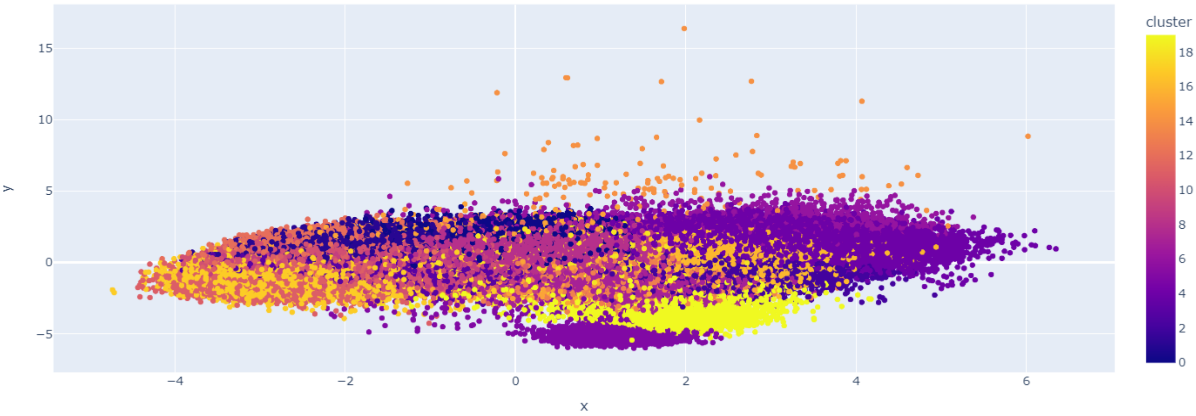
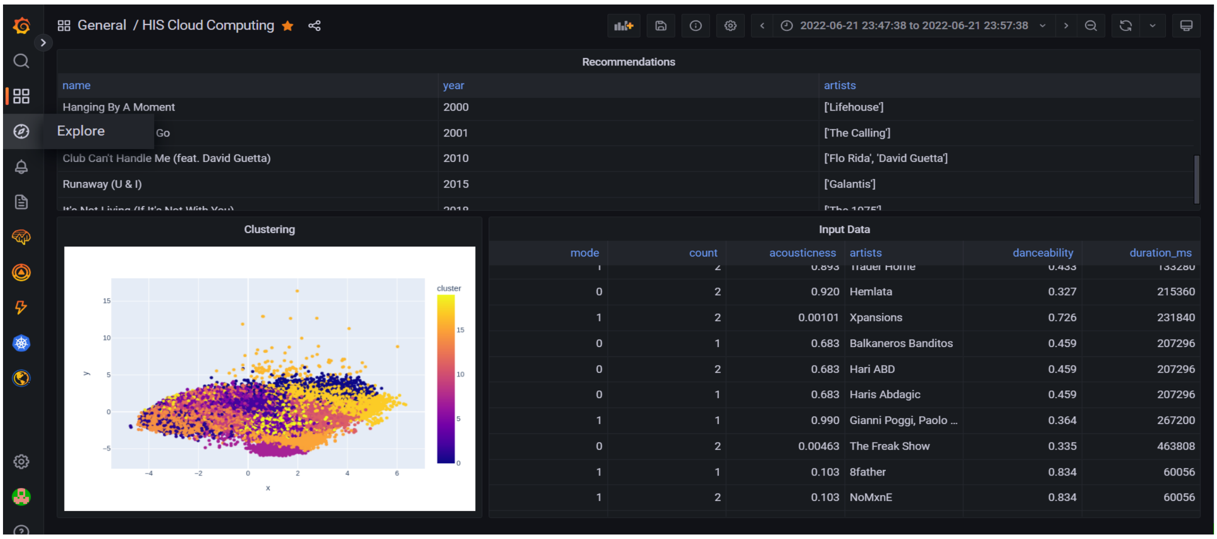


Figure 23 Generated Clusters



General / HIS Cloud Computing
2022-06-21 23:47:38 to 2022-06-21 23:57:38
Explore
Clustering
cluster

Recommendations

name	year	artists
Hanging By A Moment	2000	['Lifehouse']
Go	2001	['The Calling']
Club Can't Handle Me (feat. David Guetta)	2010	['Flo Rida', 'David Guetta']
Runaway (U & I)	2015	['Galantis']
It's Not Living (If It's Not With You)	2018	['The 1975']

Input Data

mode	count	acousticness	artists	danceability	duration_ms
1	2	0.893	Trader Home	0.433	133280
0	2	0.920	Hemlata	0.327	215360
1	2	0.00101	Xpansions	0.726	231840
0	1	0.683	Balkaneros Banditos	0.459	207296
0	2	0.683	Hari ABD	0.459	207296
0	1	0.683	Haris Abdagic	0.459	207296
1	1	0.990	Gianni Poggi, Paolo ...	0.364	267200
0	2	0.00463	The Freak Show	0.335	463808
1	1	0.103	8father	0.834	60056
1	2	0.103	NoMxnE	0.834	60056

Figure 24 Dashboard

7 Summary

In the following, the essentials of this work are summarized.

7.1 Summary

In this project, we understood that how an application can be created by leveraging services of Google Cloud Computing platform. We leveraged some of the important services of the google cloud like storage bucket, Pub-Sub, Data-flow, BigQuery and Google Colab. We implemented a trigger approach via which a ML model can be retrained whenever new data is available. Overall, we achieved our goal of getting hands on with some of the cloud services. Cloud computing can be considered as the future of computing world and sooner or later, most of the companies would move from the traditional on-prem infrastructure to cloud or at least use some of the cloud services.

References

[1] Rishabh Ahuja, Arun Solanki, and Anand Nayyar. “Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor”. In: *2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence)*. 2019, pp. 263–268. DOI: 10.1109/CONFLUENCE.2019.8776969.

[2] Shimon Ifrah. “Get Started with Google Cloud Platform (GCP)”. In: *Getting Started with Containers in Google Cloud Platform : Deploy, Manage, and Secure Containerized Applications*. Berkeley, CA: Apress, 2021, pp. 1–37. ISBN: 978-1-4842-6470-6. DOI: 10.1007/978-1-4842-6470-6_1. URL: https://doi.org/10.1007/978-1-4842-6470-6_1.

[3] Ioannis Karamitsos, Saeed Albarhami, and Charalampos Apostolopoulos. “Applying DevOps Practices of Continuous Automation for Machine Learning”. In: *Information* 11.7 (2020). ISSN: 2078-2489. DOI: 10.3390/info11070363. URL: https://www.mdpi.com/2078-2489/11/7/363.

[4] Grafana Labs. *Getting started with Grafana*. URL: https://grafana.com/docs/grafana/.

[5] Google LLC. *Google Cloud*. URL: https://cloud.google.com/docs.

[6] Sanatan Mishra. *Unsupervised Learning and Data Clustering*. URL: https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a.

[7] Ashish Patel. *Machine Learning*. 2018. URL: https://medium.com/ml-research-lab/machine-learning-algorithm-overview-5816a2e6303.

[8] Vatsal. *Recommendation Systems Explained*. URL: https://towardsdatascience.com/recommendation-systems-explained-a42fc60591ed.