@[saikatmitra1998@gmail.com](mailto:saikatmitra1998@gmail.com)

Task 3

Task Intro   
In this task we have developed a machine learning model using Google Cloud Bigquery ML, and deployed the Model to our Exclusively designed Web application which is hosted on AppEngine in google cloud: the application is being developed using Bootstrap, Alpin js, and Flask for backend and API services for Data transfer between the application and Bigquery on GCP.  
For Training our Model we have prepared the data using query -

SELECT

CONCAT('T', LPAD(CAST(ROW\_NUMBER() OVER (ORDER BY c.TaskType, c.NameOfServiceEmployee) AS STRING), 4, '0')) AS TeamId,

c.TaskType,

c.NameOfServiceEmployee AS UniqueTeams,

AVG(c.AverageTimeToCompleteTheTask) AS AverageTimeToCompleteTheTask

FROM (

SELECT

TaskType,

NameOfServiceEmployee,

DistanceTakenToReachSite,

TimeTakenToReachSite,

AverageTimeToCompleteTheTask

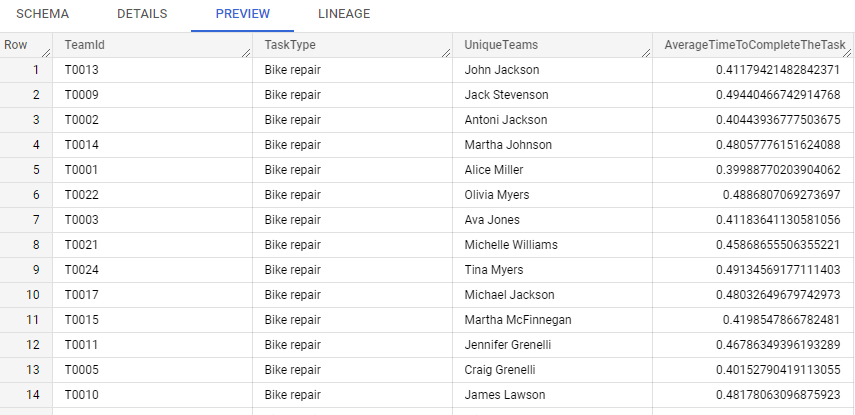
FROM MwwMsSAPData.CompletedTasksWithEmployeeTravellingInfo

GROUP BY TaskType, NameOfServiceEmployee, DistanceTakenToReachSite, TimeTakenToReachSite, AverageTimeToCompleteTheTask

) c

GROUP BY c.TaskType, c.NameOfServiceEmployee

**Output** -

  
  
And created a Table **MwwMsSAPData.Inputdata** out of it so that we can use this table in our model training later. For Training our model we developed two different types of models below -

**1. Model -**

-- Split the dataset into a training set and a test set

CREATE OR REPLACE TABLE MwwMsSAPData.Inputdata\_train AS

SELECT TaskType,

UniqueTeams,

AverageTimeToCompleteTheTask

FROM MwwMsSAPData.Inputdata

WHERE MOD(ABS(FARM\_FINGERPRINT(TeamId)), 10) < 8; -- 80% for training

CREATE OR REPLACE TABLE MwwMsSAPData.Inputdata\_test AS

SELECT

TaskType,

UniqueTeams,

AverageTimeToCompleteTheTask

FROM MwwMsSAPData.Inputdata

WHERE MOD(ABS(FARM\_FINGERPRINT(TeamId)), 10) >= 8; -- 20% for testing

-- Train the regression model

CREATE OR REPLACE MODEL MwwMsSAPData.TimePredictionModel

OPTIONS(model\_type='linear\_reg') AS

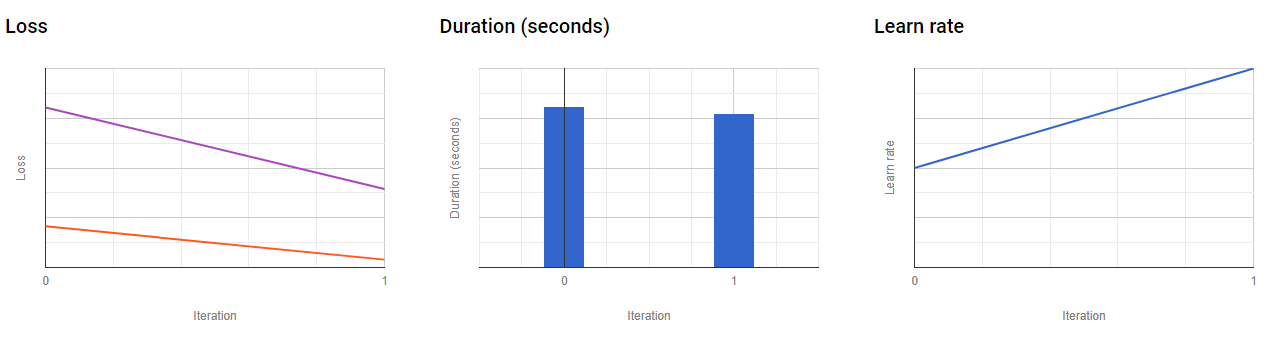
SELECT

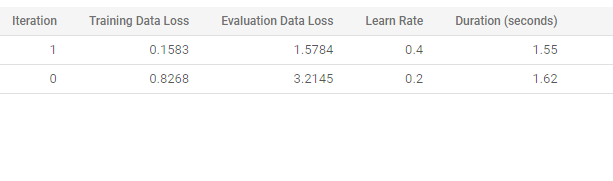
TaskType,

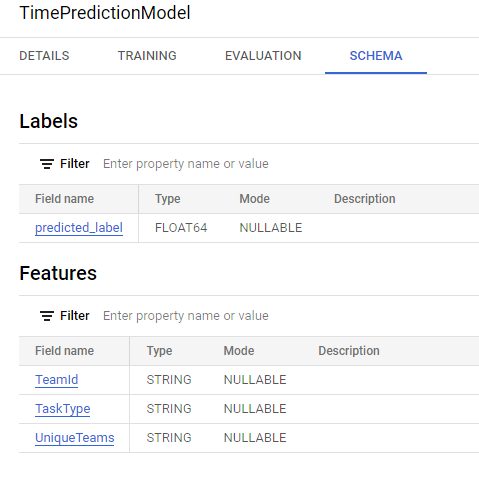
UniqueTeams,

AverageTimeToCompleteTheTask AS label

FROM MwwMsSAPData.Inputdata\_train;







[saikatmitra1998@gmail.com](mailto:saikatmitra1998@gmail.com)

**2. Model2 -**

-- Split the dataset into a training set and a test set

CREATE OR REPLACE TABLE MwwMsSAPData.Inputdata\_train2 AS

SELECT TaskType,

UniqueTeams,

AverageTimeToCompleteTheTask

FROM MwwMsSAPData.Inputdata

WHERE MOD(ABS(FARM\_FINGERPRINT(TeamId)), 10) < 8; -- 80% for training

CREATE OR REPLACE TABLE MwwMsSAPData.Inputdata\_test2 AS

SELECT

TaskType,

UniqueTeams,

AverageTimeToCompleteTheTask

FROM MwwMsSAPData.Inputdata

WHERE MOD(ABS(FARM\_FINGERPRINT(TeamId)), 10) >= 8; -- 20% for testing

-- Train the regression model

CREATE OR REPLACE MODEL MwwMsSAPData.TimePredictionModelwithoutteamid

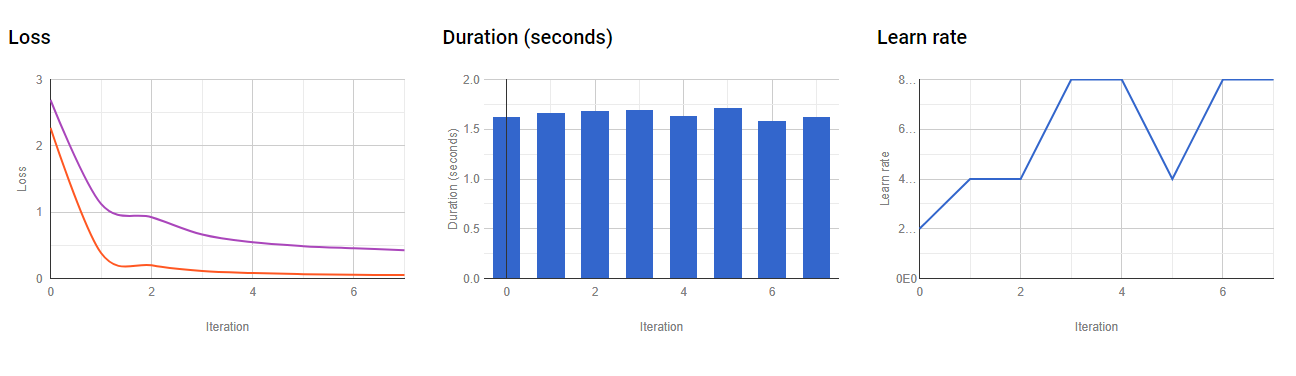
OPTIONS(model\_type='linear\_reg') AS

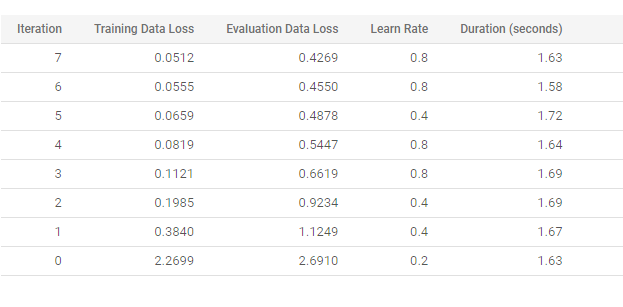
SELECT

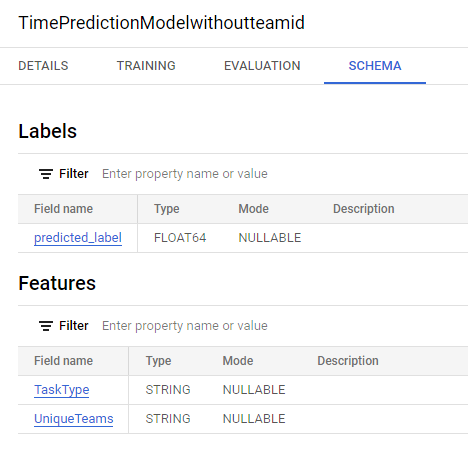
TaskType,

UniqueTeams,

AverageTimeToCompleteTheTask AS label

FROM MwwMsSAPData.Inputdata\_train2;  
  
  






[saikatmitra1998@gmail.com](mailto:saikatmitra1998@gmail.com) write about evaluation and delete pictures if you need to

**Application -**  The Application is designed to simulate the task creation with the help of a button to justify the model working and a predict button which query the machine learning model in big query and predicts the outputs sorted in best to worst sorting for selection.Deployement details [saikatmitra1998@gmail.com](mailto:saikatmitra1998@gmail.com)  
**FrontEnd-** HTML, Bootstrap

**Backend-** Flask, Bigquery API

**Infrastructure - Bigquery,AppEngine  
  
Architect Diagram -  
  
Application Dashboard -**