# Big query

## Basics

* Big query is data warehousing system not transactional
* Big query structure (Project 🡪 Dataset 🡪 Table)
* IAM role in big query can be applied at project, dataset “**CAN’T CONTROL AT TABLE LEVEL**”
* What if you want to share just two table to offshore in given data set. Solution is to create “VIEW” (Virtual table). You can have IAM role at View level.
* It also supports materialized view (BETA)
* You can make dataset public as well for all authenticated user
* LIMIT condition does not affect the cost. It fetches all the records & than just shows few records.

## How big query works

* In big query storage and compute are separated and connected with petabit network called Jupiter
* It stores data in columnar format (in colossus) contrary to traditional RDBMS system which is stored row format (record oriented)
* Big query has extremely fast performance on read but poor on write
* Big query does not support update operation on existing record

## Cost saving in big query

* How cost is calculated:
  + How many bytes read? (I/O operation)
  + How much passed between machine & next stage (Shuffle)
  + How many bytes written?
  + CPU work
* Caching query: Big query supports caching of a previous query which can save the cost. **Caching is per user only.**
* To import TB of data best performance is given by Avro- compressed format. This will reduce IO operation.
* Demoralize data (group table together with repeated data) when possible which is good for read performance & not write.
* Query optimization:
  + Avoid select \*
  + Filter early and big with where clause. This will reduce since less data is passed to next stage
  + Do biggest join first
* Partitioning:
  + consider a scenario where big retail chain dumps their sales data in big query to analyze further and create reports, most of the time they want to see last quarter sales. In this case partitioning data on the basis of Date will help so that big query won’t scan entire data.
  + You can partition by CREATION DATE or other TIMESTAMP or DATE column.
  + The new partition is introduced is Time unit partitioning. This will give us much granular control over partition. So now partition can be created on Hour to Year basis (PARTITION BY timestamp\_trunc(datetime, [HOUR/YEAR]))
* Clustering:
  + What if I want to partition on the basis of the text value, that’s when clustering comes into picture
  + Continuing previous scenario, let’s assume you want to query quarterly sales data for specific stores, so we can create clustering on the store name column.
  + We can cluster on multiple columns ex: store & product but order is really important (CLUSTER BY STORE, PRODUCT). In this case if user filter by STORE/ STORE+ PRODDUCT clustering will help but if user only filters by PRODUCT clustering won’t work.

## ML with big query

* How to train model:

CREATE MODEL `model name`

OPTIONS (MODEL\_TYPE=` `, INPUT\_LABLE\_COL = [‘Target col’]) AS

(SELECT \* FROM TABLE)

* How to check training statistics:

Click on model 🡪 evaluation tab

* How to evaluate on new data:

SELECT \* FROM

ML.EVALUATE (MODEL `model name`, SELECT \* FROM TABLE)

* How to predict on new data:

SELECT \* FROM

ML.PREDICT (MODEL `model name`, SELECT \* FROM TABLE)

* How to deploy model: Data pre-processing/Feature creation is a part of code. Input features should be exactly same of data in table.
  1. To deploy in local:
     1. Export model file in google cloud storage
     2. All TensorFlow based model will be exported as TensorFlow Saved Model
     3. AUTOML will be exported as container
     4. XG Boost will be deployed as custom prediction routine. We can download and edit those files in Python as well.
  2. To deploy on cloud:
     1. Export model to google cloud storage
     2. Use AI platform to deploy model for online prediction
     3. AUTOML models does not supports this
* Which all model types supported:
  1. Linear Regression
  2. Binary logistic regression
  3. Multi class logistic regression
  4. K-means clustering
  5. Matrix factorization for recommendation system
  6. Time series
  7. Boosted tree
  8. DNN
  9. AUTOML tables
  10. Also import previously trained TensorFlow models

## Limitation

* Can only export to cloud storage
* Can only export up to 1GB per file, but can be split to multiple files using wild card

(bq extract ‘projectid.dataset.table’ ‘gs://my-bucket/file-name-\*.json)

* Load multiple files using command line and not from UI
* Limited to 1000 table per dataset

# AI platform- Model explanation

## Overview

* Explanation (feature attribution) for each and every prediction helps you to understand model output for classification and regression
* Available for tabular data as well as image data
* **Limitation**: only supports model trained on TensorFlow 1.x if you are using keras than have to convert into estimators using model to estimator utility

## How it works

AI explanation offers two method. Both methods are based on concept of SHAP values.

* Sampled shapely:
  + For non-differentiable models such as ensemble models of tree
* Integrated gradients:
  + most suitable for differential models like deep neural network
  + especially useful for model having large feature space
  + it computes gradients of an output with respect to input. Multiplied element wise with input itself (Taylor approximation of predicted function at input)

# ML API # Vision

* **Label & web detection**: helps you to label data & web detection is step further with label it will find similar images across web & extract text where image is found to give extra information about that image.
* **OCR:** optical character recognition to extract text from images, give you coordinates where it is found as well as language of the text
* **Logo detection**: identify companies’ logos in the image
* **Landmark detection**: if image contains common landmark will also provide latitude and longitude of that landmark
* **Crop Hints**: helps you to crop the image to focus on particular subject
* **Explicit content detection**: provides likelihood ratings for the following explicit categories like adult, spoof, medical, violence & racy.
* **Face detection, document text detection (dense text or handwritten), Image properties, object localization**

# ML API # Video intelligence

* **Label detection**: it does label detection in two ways.
  + **At video level:** what is this video about
  + **At frame level**: what happening in every scene of video
* **Shot change detection**: if video changes from landscape pan to closeup it will detect that & give you timestamps for that
* **Explicit content detection**: identify inappropriate scene from video
* **Object detection and tracking**: ex tracking a car in video. It will give you all timeframe where car is present
* **Person detection with pose estimation**
* All other service which are present in Vision API is also available here.
* We can also mention region where model should run. Benefits from low latency.

# ML API # Cloud speech

* Pass an audio file and returns text transcription of that audio file
* It also supports speech timestamp which gives timestamp for each and every word of the text. It helps in audio search
* It supports Batch or streaming transcription
* **Profanity filtering**
* You can also translate text using cloud translation over 100 different languages

# ML API # Cloud NLP

* Extract entities
* Detect sentiment: sentiment can be detected at entity level.
* Analyze text:
  + Dependency parsing
  + Parsed label
  + POS tagging
  + Lemma
  + Morphology
* Classify content: 700 categories supported.