What's BigQuery?

its a petabyte-scale data analytics warehouse.

its a combination of two services -

* fast SQL query engine.
* managed storage for data.
* BigQuery uses a cache to store your pre-computed result.
* used parallel computing at scale.
* BigQuery uses googles colossus file system to store data.
* works with bulk data ingestion and streaming data ingestion
* BigQuery is most efficient when using BigQuerys own native storage.
* using the web UI of big query we can just run a part of code by using run selected by selecting the part and using run selected,
* also, if we select a database name SQL query and press ctrl + win we can get the schema of all the database it contains.
* you can preview data with preview tab in web UI instead of running a SQL cmd for it.
* you can work with external data sources such as google sheets to run the query directly but that is not recommended.
* but you can use cloud dataflow to transform ingested data.
* BigQuery natively supports arrays and structs as data types.
* BQ natively supports GIS(Geographical Information Systems) functions.
* BQ GeoViz is a geo-visual tool for analyzing geospatial data.

*Suppose,* you are dealing with data of low quality i.e. datasets having some fields as null so one can use **Cloud Dataprep** for a graphical view of data. One can build recipes which essentially consist of restrictions you can put on your data and can build a pipeline to transform data. the actual procession is done in **cloud dataflow**.

# BigQuery ML CheatSheet

1. Label: alias a column as 'label' or specify the column in OPTIONS using input\_label\_cols. you need to label afield as the label or pass an already exiting field as the label in input\_label\_cols.
2. Feature: Passed to the model as part of your SQL statement.
3. Model: an object stored in your BQ dataset.
4. Model Types: Linear Regression, Logistic Regression.
5. ML.FEATURE\_INFO: give you analytics of the features you passed to the model.
6. ML.TRAINING\_INFO: gives you the training progress.
7. ML.WEIGHTS: give you the weights of each column, the more close the weights are to -1 or 1 the better the column is.
8. ML.EVALUATE: evaluates your model performance.
9. ML.PREDICT: give the predictions on the data from your model.

**create a training dataset.**

**create/train your model.**

CREATE OR REPLACE MODEL <dataset>.<name>

OPTIONS(model\_type = '<type>') AS

<training\_dataset>

**evaluate**

FROM

ML.EVALUATE(MODEL 'model\_name', TABLE table\_name)

**predict/classify**

FROM

ML.PREDICT(MODEL 'model\_name', TABLE table\_name)

AS predict