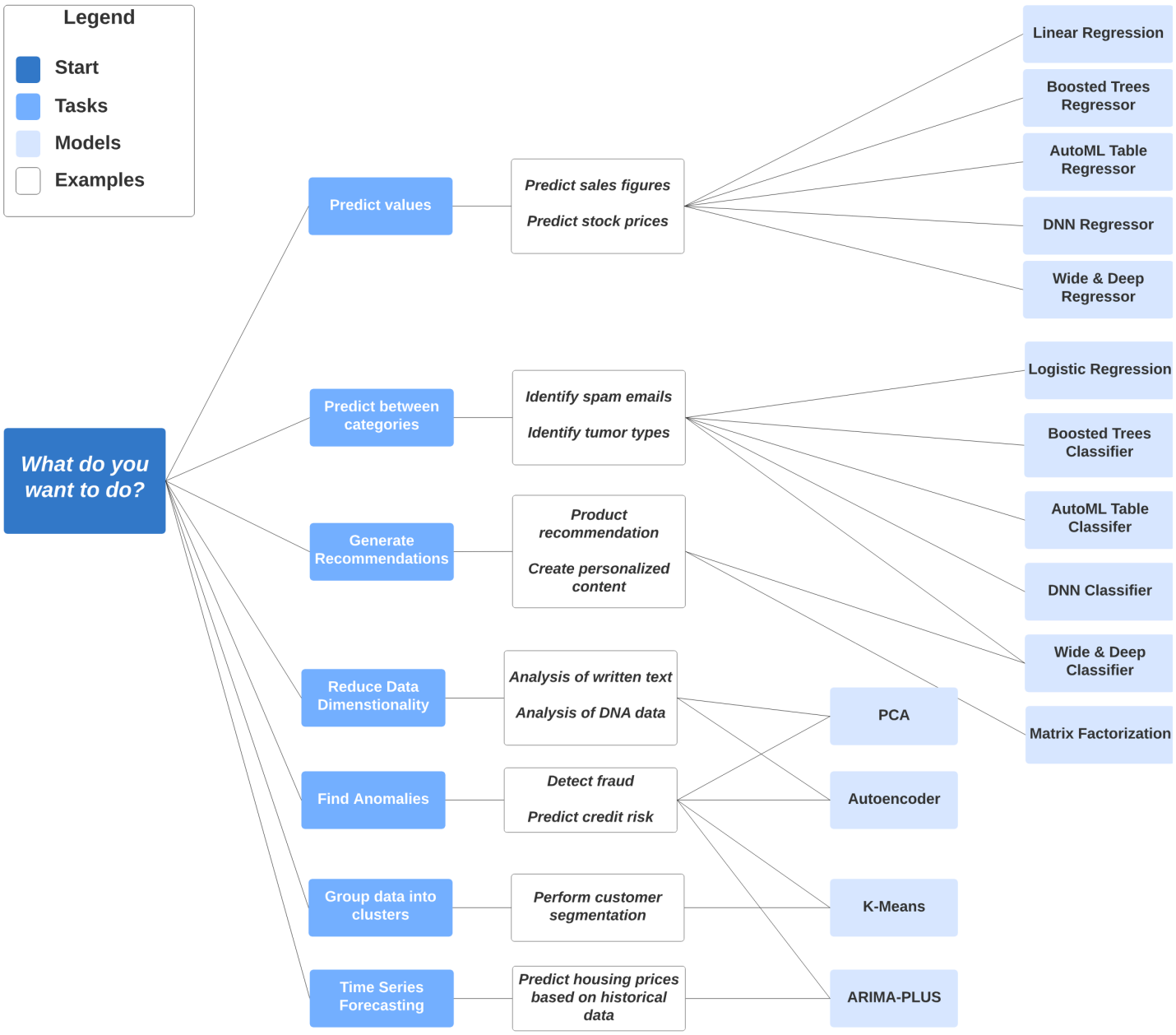
07 BQ ML - KirkYagami 👨‍💻🕵️

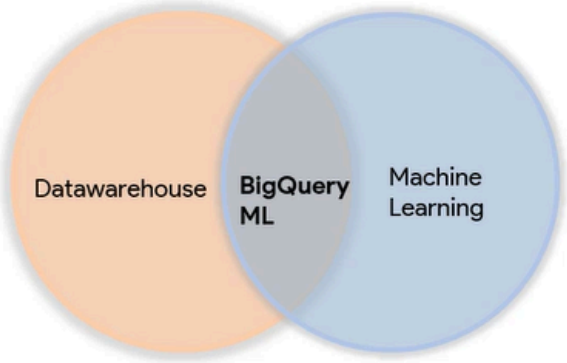


https://cloud.google.com/bigquery/docs/bqml-introduction#supported_models_in

What is BigQuery ML

"BigQuery ML lets you create and execute machine learning models in BigQuery using standard SQL queries."

"Democratizes Machine learning by letting Data analysts create, train, evaluate, and predict with ML models using existing SQL tools and skills."



What is BigQuery ML

```
CREATE OR REPLACE MODEL `bqml_tutorial.penguins_model`
OPTIONS
  (model_type = 'linear_reg',
    input_label_cols = ['body_mass_g']) AS
SELECT  * FROM  `bigquery-public-data.ml_datasets.penguins`
WHERE  body_mass_g IS NOT NULL
```

Create model query

```
SELECT  * FROM
  ML.EVALUATE (MODEL `bqml_tutorial.penguins_model`,
    ( SELECT  *  FROM
      `bigquery-public-data.ml_datasets.penguins`
    WHERE  body_mass_g IS NOT NULL) )
```

Model evaluation query

```
SELECT  *  FROM
  ML.PREDICT (MODEL `bqml_tutorial.penguins_model`,
    ( SELECT  *   FROM
      `bigquery-public-data.ml_datasets.penguins`
    WHERE  body_mass_g IS NOT NULL
      AND island = "Biscoe") )
```

Model prediction query

Challenges in ML and How BQ ML is addressing it

Challenge - ML requires trained professionals, data scientists having good knowledge of programming languages like Python or R

BQ ML – Create and Train the ML models simply with SQL.

Challenge – Data movement. Extract the data from Datawarehouse → Transform (Pandas, Datalab) → Build & Train models (TensorFlow)

BQ ML – Bring Machine Learning to Datawarehouse. Build & Train ML models on data where it sits.

Challenge – Performing standard ML tasks : Splitting data, Standardization of numeric features, Coding categorical features, Hyper parameter tuning etc.

BQ ML – All the ML tasks are managed internally by BigQuery.

Create model Query

```
CREATE MODEL | CREATE MODEL IF NOT EXISTS | CREATE OR REPLACE MODEL
model_name
TRANSFORM (select_list)
OPTIONS(model_option_list)
AS query_statement
```

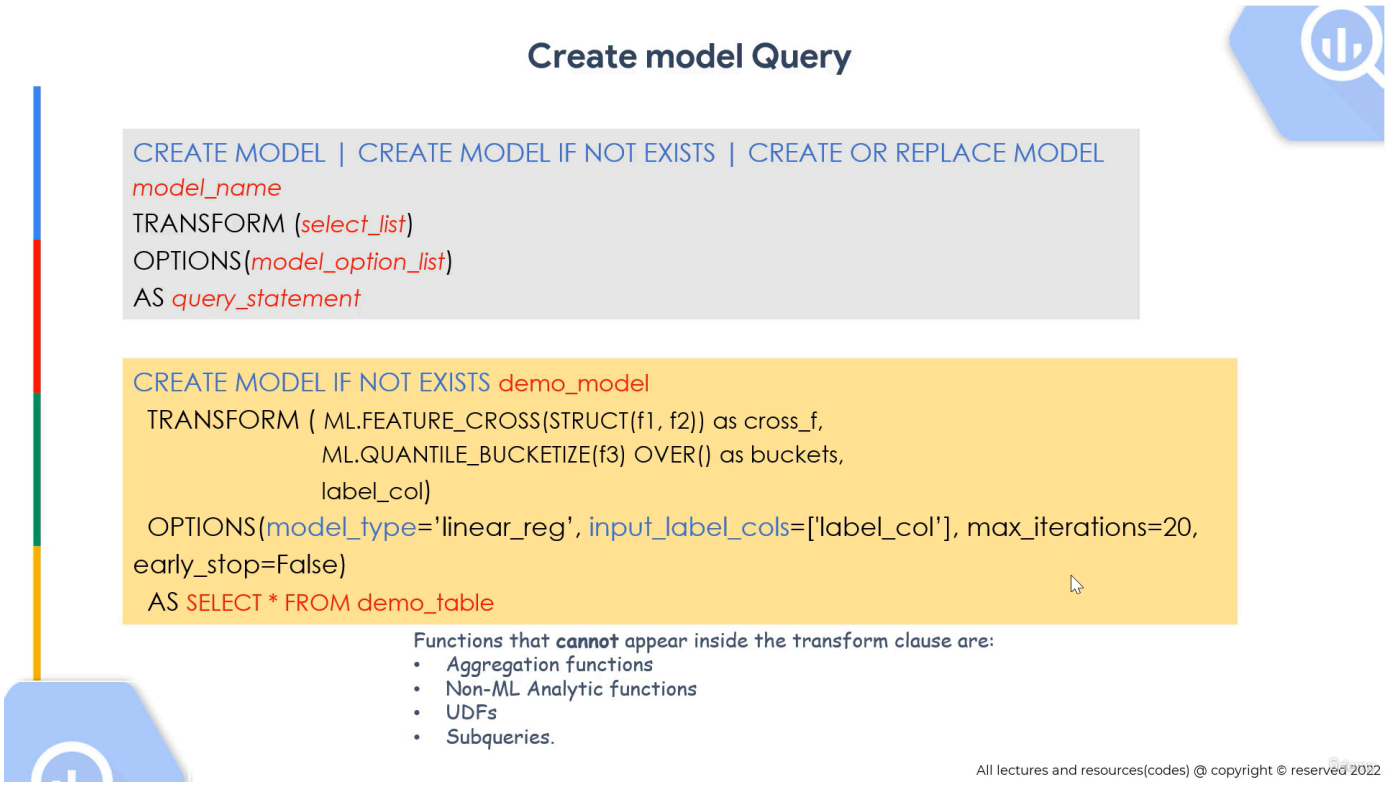
```
CREATE MODEL IF NOT EXISTS demo_model
  TRANSFORM ( ML.FEATURE_CROSS(STRUCT(f1, f2)) as cross_f,
                    ML.QUANTILE_BUCKETIZE(f3) OVER() as buckets,
                    label_col)
  OPTIONS(model_type='linear_reg', input_label_cols=['label_col'], max_iterations=20,
early_stop=False)
  AS SELECT * FROM demo_table
```

Model name must be:

- Unique per dataset.
- Can Contain up to 1,024 characters (upper or lower case alphabets, numbers and underscores).
- Not case-sensitive.

Create model Query

'LINEAR_REG’
'LOGISTIC_REG’
'KMEANS’
'PCA’
'MATRIX_FACTORIZATION’
'AUTOENCODER’
'AUTOML_REGRESSOR’
'AUTOML_CLASSIFIER’
'BOOSTED_TREE_CLASSIFIER’
'BOOSTED_TREE_REGRESSOR’
'DNN_CLASSIFIER’
'DNN_REGRESSOR’
'DNN_LINEAR_COMBINED_CLASSIFIER'
'DNN_LINEAR_COMBINED_REGRESSOR’
'ARIMA_PLUS’
'TENSORFLOW'



BigQuery ML advantages:

- Models can be built with SQL
- Data can stay within Data Warehouse:
  - Single tool used.
  - Less steps needed.
  - Can adhere to legal guidelines or requirements on data location.

BigQuery ML models can also be easily exported to a Docker container for use in online prediction.
Make sure to visit the BigQuery ML documentation online for the latest updates and how-to guides for all the major model types.

It is in US region.

```
SELECT * FROM
`bigquery-public-data.ml_datasets.penguins`
```

```
CREATE OR REPLACE MODEL bqml_example.mymodel
OPTIONS(model_type = 'linear_reg') AS
SELECT * FROM `bigquery-public-data.ml_datasets.penguins`
WHERE body_mass_g IS NOT NULL
```

```
SELECT * FROM ML.EVALUATE(MODEL `bqml_example.mymodel`,
  (SELECT * FROM `bigquery-public-data.ml_datasets.penguins`
   WHERE body_mass_g IS NOT NULL)
)
```

```
SELECT * FROM ML.PREDICT(MODEL `bqml_example.mymodel`,
  (SELECT * FROM `bigquery-public-data.ml_datasets.penguins`
   WHERE body_mass_g IS NOT NULL AND island = 'Biscoe')
)
```

Considerations

To leverage the entire GCP ecosystem for your data pipeline, consider using Dataflow to perform the processing and data transformations.
When it comes to BigQuery, understand the key factors that influence the performance and cost.

BigQuery Key Factors:

- Input and Output Size
- Shuffling
- Grouping
- Materialization
- Functions and UDFs

BigQuery query advice:

- Avoid selecting unnecessary columns.
- Filter as early as possible in the query.
- Put larger tables on the left of JOINs.
- ORDER BY should go last.
- When using GROUP BY, consider the amount of unique entries in the column.
- Take advantage of the Execution Details tab to get more details of compute used in your query.
- Take advantage of approximate functions, such as APPROX_TOP_COUNT instead of COUNT(DISTINCT).
  Note:
- BigQuery does periodically recluster data automatically for you at no extra charge.
- Use Views for common queries.
- Take advantage of Cloud Monitoring to understand the impacts of BigQuery utilization.

Consider scope of information needed:

- You may want to create smaller tables and only use the most recent data in the BigQuery data storage.
- Compare the costs of processing data from outside sources vs. storage costs within the BigQuery data warehouse.

Example Usage of BigQuery ML: Clustering with K-Means

In this example, we'll use the **K-Means clustering algorithm** on the **Iris dataset** from BigQuery's public datasets. The goal is to group the Iris flowers based on their petal and sepal characteristics.

Step 1: Load Dataset

```
SELECT * FROM `bigquery-public-data.ml_datasets.iris`
```

- **Dataset**: `bigquery-public-data.ml_datasets.iris`
- **Task**: Cluster the Iris flowers into groups based on their features (sepal length, sepal width, petal length, and petal width).

Step 2: Create and Train the K-Means Clustering Model

```
CREATE OR REPLACE MODEL bqml_example.iris_kmeans_model
OPTIONS(model_type = 'kmeans', num_clusters = 3) AS
SELECT
  sepal_length,
  sepal_width,
  petal_length,
  petal_width
FROM
  `bigquery-public-data.ml_datasets.iris`
```

- **Explanation**:
  - `model_type = 'kmeans'`: Specifies that we are using the K-Means clustering algorithm.
  - `num_clusters = 3`: We expect the model to cluster the flowers into 3 groups (as Iris dataset has three species).

Step 3: Evaluate the Clustering Model

```
SELECT *
FROM
  ML.EVALUATE(MODEL `bqml_example.iris_kmeans_model`)
```

- **Explanation**: The `ML.EVALUATE` function gives you metrics like the **mean squared distance** between points in each cluster, which helps assess how tightly grouped the clusters are.

Step 4: Use the Model to Make Predictions

```
SELECT *
FROM
  ML.PREDICT(MODEL `bqml_example.iris_kmeans_model`,
  (SELECT
    sepal_length,
    sepal_width,
    petal_length,
    petal_width
  FROM
    `bigquery-public-data.ml_datasets.iris`))
```

- **Explanation**:
  - `ML.PREDICT` assigns a cluster to each data point (flower), providing the cluster ID for each Iris flower based on the trained K-Means model.