```
  -- Create a cleaned table for modelling

CREATE or replace TABLE
stage1energy.dataset.table_clean
 as
select ML.LABEL_ENCODER(BuildingType) OVER () AS BuildingType,
ML.LABEL_ENCODER(PrimaryPropertyType) OVER () AS PrimaryPropertyType,
ML.LABEL_ENCODER(Neighborhood) OVER () AS Neighborhood, PropertyGFATotal, PropertyGFABuilding_s_,
ML.LABEL_ENCODER(ListOfAllPropertyUseTypes) OVER () AS ListOfAllPropertyUseTypes,
ML.LABEL_ENCODER(LargestPropertyUseType) OVER () AS LargestPropertyUseType,
LargestPropertyUseTypeGFA,
ML.LABEL_ENCODER(SecondLargestPropertyUseType) OVER () AS SecondLargestPropertyUseType,
ML.LABEL_ENCODER(ThirdLargestPropertyUseType) OVER () AS ThirdLargestPropertyUseType,
Electricity_kWh_, NaturalGas_therms_, TotalGHGEmissions
FROM `stage1energy.dataset.table_all_cols`
WHERE Electricity_kWh_>0
OR TotalGHGEmissions>0;




  -- Checking for anomalies in Y, NANs treatment (Examples of codes, NULLs are treated
automatically by BQ ML)
select min(Electricity_kWh_), min(TotalGHGEmissions)
from stage1energy.dataset.table_clean;

select count(*)
FROM `stage1energy.dataset.table_clean`
where BuildingType is null;

select distinct(BuildingType), count(BuildingType)
FROM `stage1energy.dataset.table_clean`
group by 1
order by 2 desc;
UPDATE `stage1energy.dataset.table_clean_test`
set SecondLargestPropertyUseType=
(select APPROX_TOP_COUNT(SecondLargestPropertyUseType, 1)[OFFSET(0)].value
from `stage1energy.dataset.table_clean_test`
where SecondLargestPropertyUseType is not null
limit 1)
where SecondLargestPropertyUseType is null;


UPDATE `stage1energy.dataset.table_clean_test`
set LargestPropertyUseTypeGFA=
cast((select PERCENTILE_CONT(LargestPropertyUseTypeGFA, 0.5) over()
from `stage1energy.dataset.table_clean_test`
where LargestPropertyUseTypeGFA is not null limit 1) as int)
where LargestPropertyUseTypeGFA is null;

UPDATE `stage1energy.dataset.table_clean_test_num`

set NaturalGas_therms_=
cast((select PERCENTILE_CONT(NaturalGas_therms_, 0.5) over()
from `stage1energy.dataset.table_clean_test_num`
limit 1) as int)
where NaturalGas_therms_=0;


-- Y1 ('TotalGHGEmissions')
-- Linear Regression model ('TotalGHGEmissions')

CREATE OR REPLACE MODEL
```

```
  stage1energy.dataset.emission_lr_model
OPTIONS
  ( model_type='LINEAR_REG',
    enable_global_explain=TRUE,
    input_label_cols=['TotalGHGEmissions'],
    max_iterations=15,
    DATA_SPLIT_METHOD = 'AUTO_SPLIT')
AS SELECT * except(Electricity_kWh_)
FROM stage1energy.dataset.table_clean;

-- Random Forest model ('TotalGHGEmissions')

CREATE OR REPLACE MODEL
  stage1energy.dataset.emission_rf_model
OPTIONS(MODEL_TYPE='RANDOM_FOREST_REGRESSOR',
        enable_global_explain=TRUE,
        NUM_PARALLEL_TREE = 50,
        TREE_METHOD = 'HIST',
        EARLY_STOP =TRUE,
        INPUT_LABEL_COLS = ['TotalGHGEmissions'],
        DATA_SPLIT_METHOD = 'AUTO_SPLIT')
AS SELECT * except(Electricity_kWh_)
FROM stage1energy.dataset.table_clean;

  -- Deep Neural Network (DNN) model ('TotalGHGEmissions')

CREATE OR REPLACE MODEL stage1energy.dataset.emission_dnn_model
OPTIONS(MODEL_TYPE='DNN_REGRESSOR',
        enable_global_explain=TRUE,
        ACTIVATION_FN = 'RELU',
        BATCH_SIZE = 16,
        DROPOUT = 0.1,
        EARLY_STOP = TRUE,
        HIDDEN_UNITS = [128, 128, 128],
        INPUT_LABEL_COLS = ['TotalGHGEmissions'],
        DATA_SPLIT_METHOD = 'AUTO_SPLIT',
        LEARN_RATE=0.001,
        MAX_ITERATIONS = 25,
        OPTIMIZER = 'ADAM')
AS SELECT * except(Electricity_kWh_)
FROM stage1energy.dataset.table_clean;

  -- Boosted Trees model ('TotalGHGEmissions') / Example of different codes to retrieve the
results of the modelling and make predictions

CREATE OR REPLACE MODEL
  stage1energy.dataset.emission_bt_model
OPTIONS
      ( MODEL_TYPE='BOOSTED_TREE_REGRESSOR',
        enable_global_explain=TRUE,
        BOOSTER_TYPE = 'GBTREE',
        NUM_PARALLEL_TREE = 1,
        MAX_ITERATIONS = 50,
        TREE_METHOD = 'HIST',
        EARLY_STOP = TRUE,
        INPUT_LABEL_COLS = ['TotalGHGEmissions'],
        DATA_SPLIT_METHOD = 'AUTO_SPLIT')
AS SELECT * except(Electricity_kWh_)
FROM stage1energy.dataset.table_clean;

-- evaluate the model on test data ('TotalGHGEmissions')

SELECT
```

```
  *
FROM
  ML.EVALUATE (MODEL `stage1energy.dataset.emission_bt_model`);


-- globally explain the model ('TotalGHGEmissions')
SELECT
  *
FROM
  ML.GLOBAL_EXPLAIN(MODEL `stage1energy.dataset.emission_bt_model`);

  -- the model prediction ('TotalGHGEmissions')
SELECT
  *
FROM
  ML.PREDICT (MODEL `stage1energy.dataset.emission_bt_model`,
    (
      SELECT * except(Electricity_kWh_)
      FROM stage1energy.dataset.table_clean
     )
  );

  -- explain the model prediction ('TotalGHGEmissions')
SELECT
*
FROM
ML.EXPLAIN_PREDICT(MODEL `stage1energy.dataset.emission_bt_model`,
  (
  SELECT * except(Electricity_kWh_)
  FROM stage1energy.dataset.table_clean
  ),
  STRUCT(3 as top_k_features));


-- Y2 ('Electricity_kWh_')
-- Linear Regression model ('Electricity_kWh_')

CREATE OR REPLACE MODEL
  stage1energy.dataset.electricity_lr_model
OPTIONS
  ( model_type='LINEAR_REG',
    enable_global_explain=TRUE,
    input_label_cols=['Electricity_kWh_'],
    max_iterations=15,
    DATA_SPLIT_METHOD = 'AUTO_SPLIT')
AS SELECT * except(TotalGHGEmissions)
FROM stage1energy.dataset.table_clean;

-- Random Forest model ('Electricity_kWh_')

CREATE OR REPLACE MODEL
  stage1energy.dataset.electricity_rf_model
OPTIONS(MODEL_TYPE='RANDOM_FOREST_REGRESSOR',
        enable_global_explain=TRUE,
        NUM_PARALLEL_TREE = 50,
        TREE_METHOD = 'HIST',
        EARLY_STOP =TRUE,
        INPUT_LABEL_COLS = ['Electricity_kWh_'],
        DATA_SPLIT_METHOD = 'AUTO_SPLIT')
AS SELECT * except(TotalGHGEmissions)
FROM stage1energy.dataset.table_clean;

  -- Deep Neural Network (DNN) model ('Electricity_kWh_')
```

```sql
CREATE OR REPLACE MODEL stage1energy.dataset.electricity_dnn_model
OPTIONS(MODEL_TYPE='DNN_REGRESSOR',
        enable_global_explain=TRUE,
        ACTIVATION_FN = 'RELU',
        BATCH_SIZE = 16,
        DROPOUT = 0.1,
        EARLY_STOP = TRUE,
        HIDDEN_UNITS = [128, 128, 128],
        INPUT_LABEL_COLS = ['Electricity_kWh_'],
        DATA_SPLIT_METHOD = 'AUTO_SPLIT',
        LEARN_RATE=0.001,
        MAX_ITERATIONS = 25,
        OPTIMIZER = 'ADAM')
AS SELECT * except(TotalGHGEmissions)
FROM stage1energy.dataset.table_clean;

  -- Boosted Trees model ('Electricity_kWh_') / Example of different codes to retrieve the
results of the modelling and make predictions

CREATE OR REPLACE MODEL
  stage1energy.dataset.electricity_bt_model
OPTIONS
      ( MODEL_TYPE='BOOSTED_TREE_REGRESSOR',
        enable_global_explain=TRUE,
        BOOSTER_TYPE = 'GBTREE',
        NUM_PARALLEL_TREE = 1,
        MAX_ITERATIONS = 50,
        TREE_METHOD = 'HIST',
        EARLY_STOP = TRUE,
        INPUT_LABEL_COLS = ['Electricity_kWh_'],
        DATA_SPLIT_METHOD = 'AUTO_SPLIT')
AS SELECT * except(TotalGHGEmissions)
FROM stage1energy.dataset.table_clean;

-- evaluate the model on test data ('Electricity_kWh_')

SELECT
  *
FROM
  ML.EVALUATE (MODEL `stage1energy.dataset.electricity_bt_model`);


-- globally explain the model ('Electricity_kWh_')
SELECT
  *
FROM
  ML.GLOBAL_EXPLAIN(MODEL `stage1energy.dataset.electricity_bt_model`);

  -- the model prediction ('Electricity_kWh_')
SELECT
  *
FROM
  ML.PREDICT (MODEL `stage1energy.dataset.electricity_bt_model`,
    (
      SELECT * except(Electricity_kWh_)
      FROM stage1energy.dataset.table_clean
     )
  );

  -- explain the model prediction ('Electricity_kWh_')
SELECT
*
```

```
FROM
ML.EXPLAIN_PREDICT(MODEL `stage1energy.dataset.electricity_bt_model`,
  (
  SELECT * except(Electricity_kWh_)
  FROM stage1energy.dataset.table_clean
  ),
  STRUCT(3 as top_k_features));
```