Using BigQuery in ML for NCAA Basketball

Here we are creating a BigQuery ML model for NCAA basketball using the public dataset. We are providing our query details and output screenshots.

First the input features of the model is queried using mbb_teams_games_sr table in the NCAA Basketball public dataset. The input features include the mean and standard deviation of previous game statistics for both home teams and away teams using different time windows. The time windows used are 10, 5 and 1 game before the current game. The team_id and season columns are used for one-hot encoding features. After generating the input features, we will generate training data.

After the training data has been generated, we will create a linear regression model. The model is used to predict the combined three-point field goal attempts based on the previous game statistics.

Below is a BigQuery to create a linear regression model. Here we have used all the other columns as features except columns related to the three point, game_id, season, schedule_date etc. which we want to predict later. These fields are all related to predict the overall game results.

```
CREATE OR REPLACE MODEL bqml_tutorial.ncaa_model OPTIONS(
  model_type='linear_reg', max_iteration=50) AS
SELECT
  * EXCEPT (
    game_id, season, scheduled_date,
total_three_points_made,
total_three_points_att),
total_three_points_att as label
FROM
  bqml_tutorial.wide_games
WHERE
  # remove the game to predict
  game_id != 'f1063e80-23c7-486b-9a5e-faa52beb2d83'
```

On querying it, we get the model:

Saved Query: BQML NCAA Basketball - wide_games [edited]

```
SELECT
  *
FROM
  ML.TRAINING_INFO(MODEL `bqml_tutorial.ncaa_model`)
```

Standard SQL Dialect

RUN QUERY | Save Query | Save View | Format Query | Schedule Query | Show Options | Query complete (1.5s elapsed, 0 B processed)

Results | Details | Download as CSV | Download as JSON | Save as Table | Save to Google Sheets

Row	training_run	iteration	loss	eval_loss	duration_ms	learning_rate
1	0	8	54.29042675264878	126.84568671743307	12501	0.05
2	0	7	64.04867194217444	126.88062827348686	9478	0.05
3	0	6	77.65029601811216	153.34659335309902	9900	0.05
4	0	5	96.77048342428301	155.06878018178773	9347	0.05
5	0	4	123.92071536616325	204.74935907634747	11366	0.05
6	0	3	163.00473166732914	212.59380546075826	11214	0.05
7	0	2	220.53564430745905	309.24688662799144	11890	0.05
8	0	1	309.6986255136534	339.867022050369	8354	0.05
9	0	0	524.7007876580622	613.319021000273	9866	0.2

Table | JSON

After creating the model, we will evaluate the performance of the model using the ML.EVALUATE function. Below is the query for the same.

WITH eval_table AS (

SELECT *, total_three_points_att AS label FROM `bqml_tutorial.wide_games`)

SELECT * FROM ML.EVALUATE(MODEL `bqml_tutorial.ncaa_model`, TABLE eval_table)

Below is the output we will get after executing the above query.

Saved Query: BQML NCAA Basketball - wide_games [edited]

```
WITH eval_table AS (
SELECT
  *,
  total_three_points_att AS label
FROM
  `bqml_tutorial.wide_games` )
SELECT
  *
FROM
  ML.EVALUATE(MODEL `bqml_tutorial.ncaa_model`,
    TABLE eval_table)
```

Standard SQL Dialect

RUN QUERY | Save Query | Save View | Format Query | Schedule Query | Show Options | Query complete (4.0s elapsed, 23.9 MB processed)

Results | Details | Download as CSV | Download as JSON | Save as Table | Save to Google Sheets

Row	mean_absolute_error	mean_squared_error	mean_squared_log_error	median_absolute_error	r2_score	explained_variance
1	8.430325827940533	89.2065416949254	0.061800726325130714	5.411970069525921	0.08601259240738124	0.42138196780608867

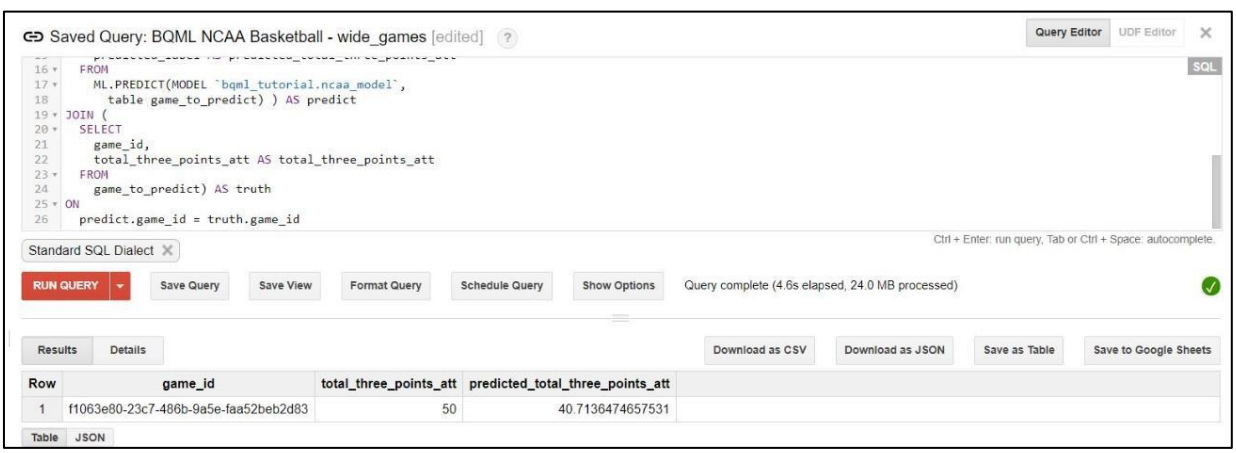
Table | JSON

Now that we have evaluated the model, the next step is to use the ML.PREDICT function to predict the total three point field goal attempts in the 2018 NCAA final game: Michigan versus Villanova.

Below is the query for the same.

```
WITH game_to_predict AS (
  SELECT * FROM `bqml_tutorial.wide_games` WHERE game_id='f1063e80-23c7-486b-9a5e-faa52beb2d83' )
SELECT truth.game_id AS game_id, total_three_points_att, predicted_total_three_points_att
FROM (
  SELECT game_id, predicted_label AS predicted_total_three_points_att
  FROM ML.PREDICT(MODEL `bqml_tutorial.ncaa_model`, table game_to_predict) ) AS predict
JOIN (
  SELECT game_id, total_three_points_att AS total_three_points_att
  FROM game_to_predict) AS truth ON predict.game_id = truth.game_id
```

Below is the output we will get after executing the above query.



The total_three_points_att value is the actual number of field goals that occurred in the final game – 50. The model's prediction is 40.71