In-Database Machine Learning on Bicycle Data from Munich

The difference between this approach to bicycle data analysis and typical approaches is, that this approach uses an Exasol database as the main storage and the main platform for executing analytical logic. Since the analytical functions provided by plain SQL are not sufficient for implementing all analytical tools needed, a framework extending the database for machine learning is used. This framework was developed by me during my master's thesis on the topic of "Extending SQL for Machine Learning".

As already mentioned, the column-based, cluster-based, and relational Exasol database was used for this approach. Exasol can already be extended for machine learning using so-called user-defined function (UDF) scripts. These execute code written in a scripting language, like Python. The framework created during my master thesis uses this functionality to provide a natural SQL integration for machine-learning algorithms. Thus, the framework creates an interface to access algorithms of the Python library *Scitkit-Learn* in SQL. The created models are stored in Exasol replicated file System BucketFS. The following graphic provides an overview of the framework:

The main contributions of the framework are the support for exploratory data analysis, increased scalability and cloud compatibility using Exasol clusters, increased efficiency, increased security, and simplification.

For more information regarding the framework and my master's thesis, contact me.

Munich was chosen because of its robust set of bicycle data. The created analyses are transferable to other cities. For all analysis, the data of the "Raddauerzählstellen" in Munich from the year 20172017 to the year 20222022 were used.

The goals for the analysis were to provide a proof-of-concept for the machine-learning framework and show the potential for bicycle data analysis.

Correlation between Rain and Bicycle Traffic

The problem statement for this analysis was, to determine the correlation between the amount of rain and the amount of bicycle traffic on a given day. For this purpose, the following plain SQL statement can be used:

```
SELECT CORR(RAIN , TRAFFIC)
```

```
FROM MUN.SUMPERDAY
```

When running this statement for the data of the “Raddauerzählstellen” starting from the year $2017$2017 to the year $2022$2022, the result is -0.18−0.18. Thus, a correlation between the amount of rain and bicycle traffic can be shown by using plain SQL functions.

Evaluation of Changes in Bicycle Traffic over the Years

The problem statement for this analysis was, to evaluate the changes in the amount of bicycle traffic between the years $2017$2017 and $2022$2022. For this purpose, the following plain SQL statement can be used:

```
SELECT "YEAR", AVG(TRAFFIC)
FROM MUN.SUMPERDAY
GROUP BY "YEAR" ORDER BY "YEAR"
```

The results of this statement are shown in the following diagram (which was created using *plotly*):

The diagram shows the overall increase in bicycle traffic over the years of data collection. The spike in the year $2020$2020 nicely compounds increased bicycle sales caused by lockdowns because of the Covid 19 pandemic.

Prediction of Bicycle Traffic by Daily Weather

The problem statement for this analysis was, to predict the bicycle traffic on a given day using the weather forecast for that day. For this purpose, the OpenWeather 5-day weather forecast was used.

To do this prediction, a machine-learning model is needed. For this application, I decided to use the decision-tree regressor algorithm of the *Scitkit-Learn* library. Thanks to the framework, the model can be trained in SQL using the following statement:

```
SELECT  ML.sklearn_tree_DecisionTreeRegressor_train
        (
        'trafficbyweather', -- Model name
        '{ "model_params": { "max_depth": 64 } }',
                    -- Model parameter
        "YEAR",     -- Feature 1
        "MONTH",    -- Feature 2
        "WEEKDAY",  -- Feature 3
        TEMP_MIN,   -- Feature 4
        TEMP_MAX,   -- Feature 5
        RAIN,       -- Feature 6
        TRAFFIC     -- Label
        ) -- Returns the model name
```

```
FROM    MUN.SUMPERDAY;
```

The created model is stored in BucketFS. Now, for the prediction, the following SQL statement can be used:

```
SELECT      ML.sklearn_tree_DecisionTreeRegressor_predict
            (
            'trafficbyweather', -- Model name
```

```
            '',                 -- Model parameter
            "TIMESTAMP",        -- Identifier
            "YEAR",             -- Feature 1
            "MONTH",            -- Feature 2
            "WEEKDAY",          -- Feature 3
            TEMP_MIN,           -- Feature 4
            TEMP_MAX,           -- Feature 5
            RAIN                -- Feature 6
            ) -- Emits tuples consisting of
              -- the identifier and the label
FROM        MUN.DAILYWEATHERPREDICTION
```

```
GROUP BY    iproc()
ORDER BY    IDENTIFIER DESC;
```

The output of the prediction can now be used to show the bicycle traffic prediction in addition to the weather forecast. To make this information easily understandable for a user, graphical output is advantageous. For this application, a GUI using *tkinter* was created.

The GUI displays the predicted bicycle traffic, and a comparison to previously recorded bicycle traffic, in addition to the weather forecast for each day. This comparison is split into five parts, the first representing the traffic between the 0%0% and the 20%20% quantile, the second between the 20%20% and the 40%40%, and so on.

The syntax for model creation and prediction shown here is not the final but an intermediate syntax. The final syntax was not implemented during my master's thesis due to time constraints.

- In-Database Machine Learning on Bicycle Data from Munich
  - Correlation between Rain and Bicycle Traffic
  - Evaluation of Changes in Bicycle Traffic over the Years
  - Prediction of Bicycle Traffic by Daily Weather

Expand allBack to topGo to bottom