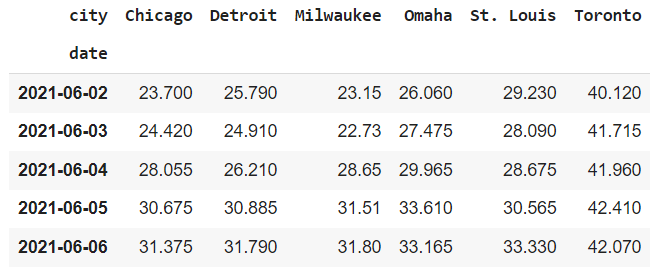
**Data Drift Monitoring for Simple Tabular data:**

In production, it is possible that a model encounters data with a different underlying distribution than it saw during training. This could lead to a case where the model is not properly creating predictions. Thus, it is important to monitors models and their data to ensure that they are functioning as expected.

In this case, the model being monitored is a simple example meant largely for demonstration purposes. This model takes the temperatures of 5 nearby cities (Detroit, Milwaukee, Omaha, Toronto, and St. Louis), and predicts the weather of Chicago. This is done using a simple SVM, and is not done with a time series perspective (the idea is that it could predict based on the temperature of the others). This scenario allows for the possibility of drift (as weather changes seasonally), as well as overall shifts.

**Example of Data Shift in Tabular Data:**

One change that was made to illustrate a data shift was switching data from Toronto to Phoenix, AZ. As would be expected, the distribution of temperatures is quite different between the two cities, especially over the dates used in this example (from March to early June). Here is a quick look at a sample from the data:



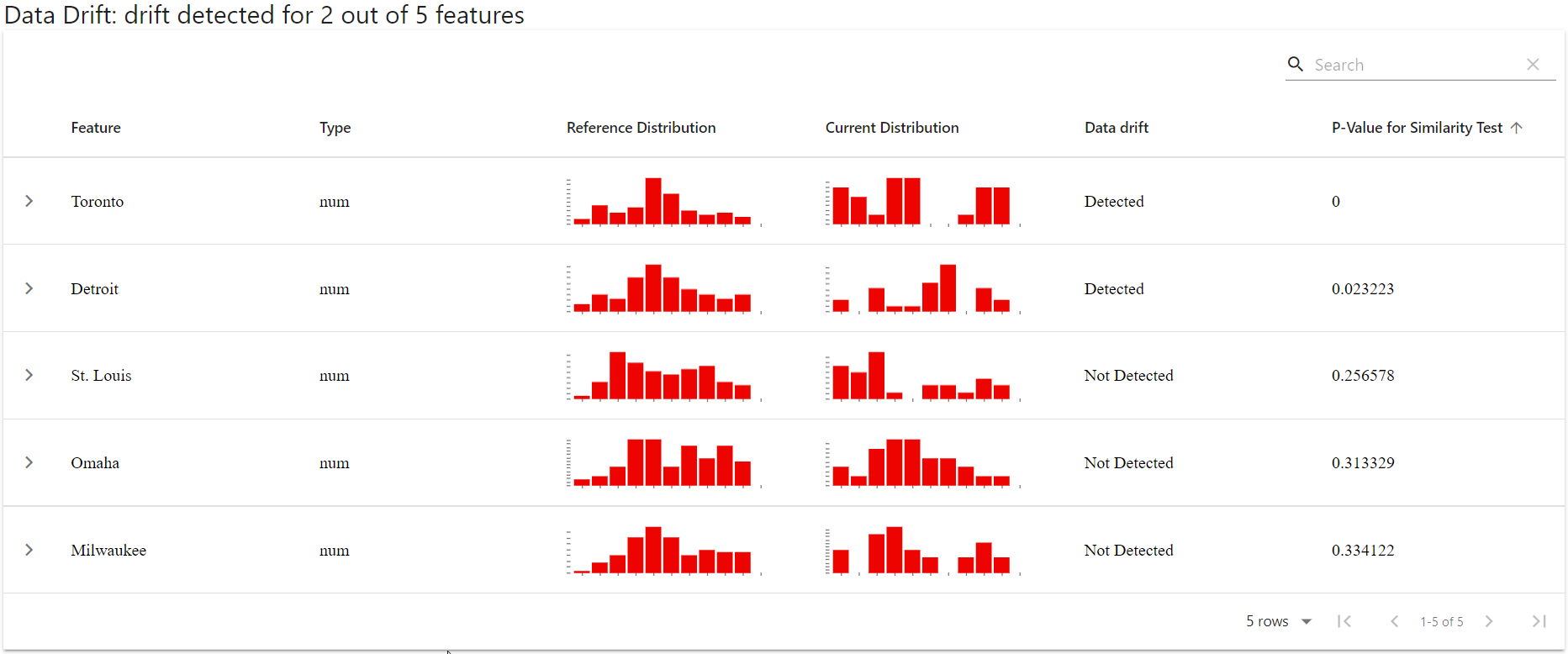
Notice how “Toronto” has significantly higher values than the other cities, as its value has been changed to that of Phoenix!

Evidently was used to monitor models – it is able to take an input dataframe of original training data, and an input dataframe of production training data, and create a dashboard with analytics about any differences.

The input was easy to use:



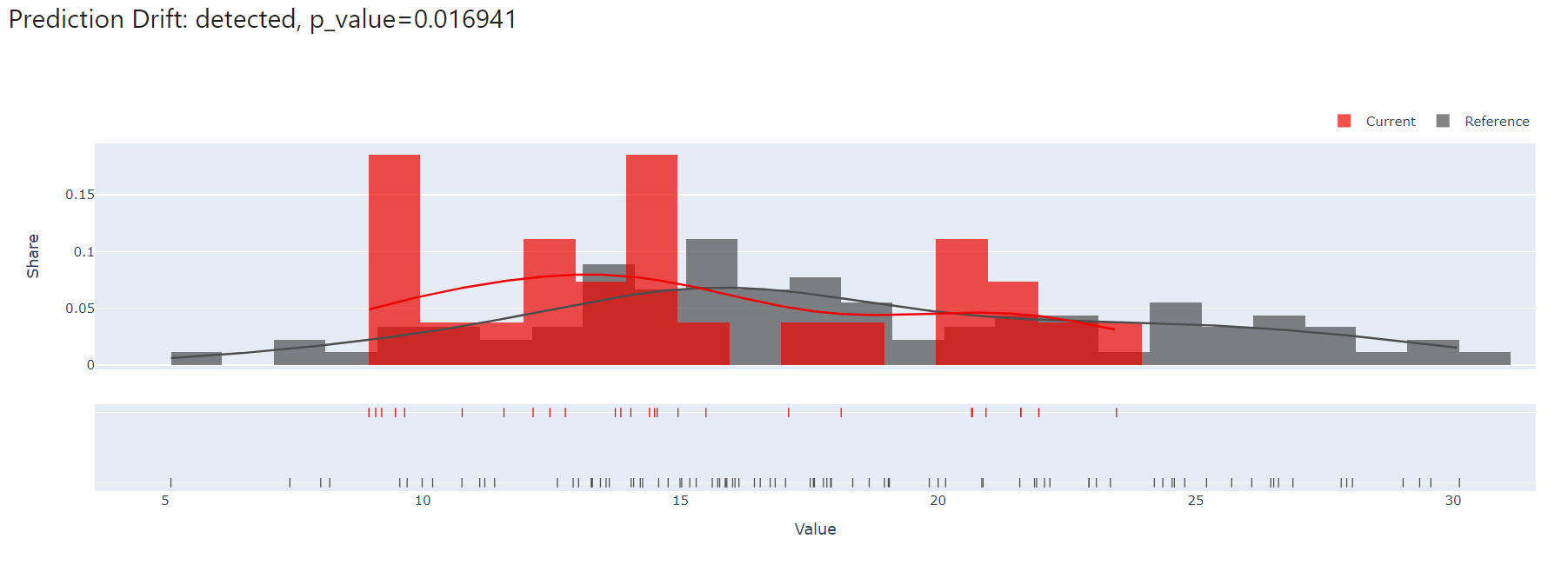
It can identify the input variables in each of the datasets (here df\_old corresponds to the original training data, and df corresponds to the data which the model is currently predicting). It can then match the variables across the datasets, and produce a report highlighting how there are differences:



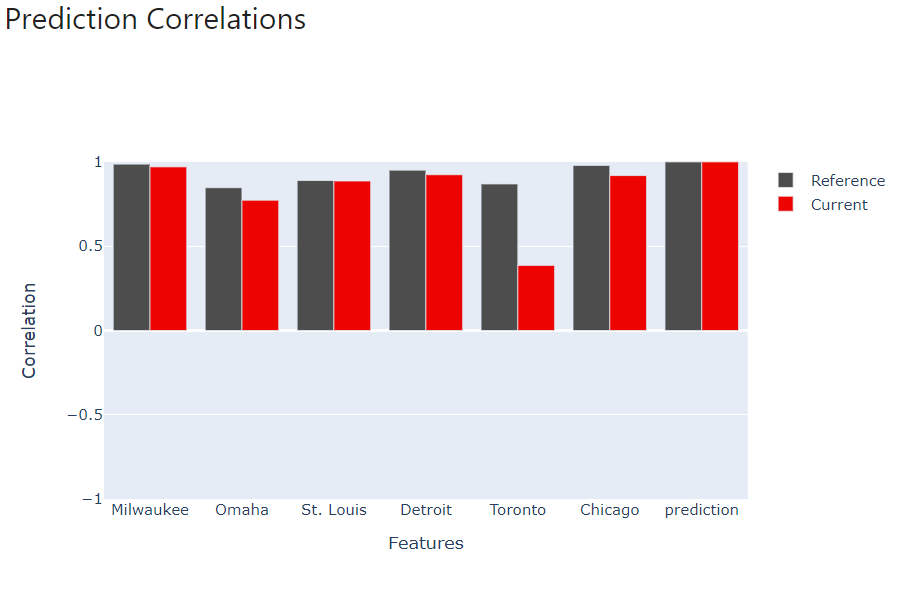
It is immediately clear that there is a drift in the distribution for “Toronto”, as there are now far more high values. Interestingly, a shift was also detected for Detroit, which is worth noting.

Evidently also offers capabilities for monitoring drift in the outcome variable as well, to understand if the models predictions are deviating from what was seen during training. This so called “Target Shift” can have serious business implications on a model in production, and should be watched for carefully.

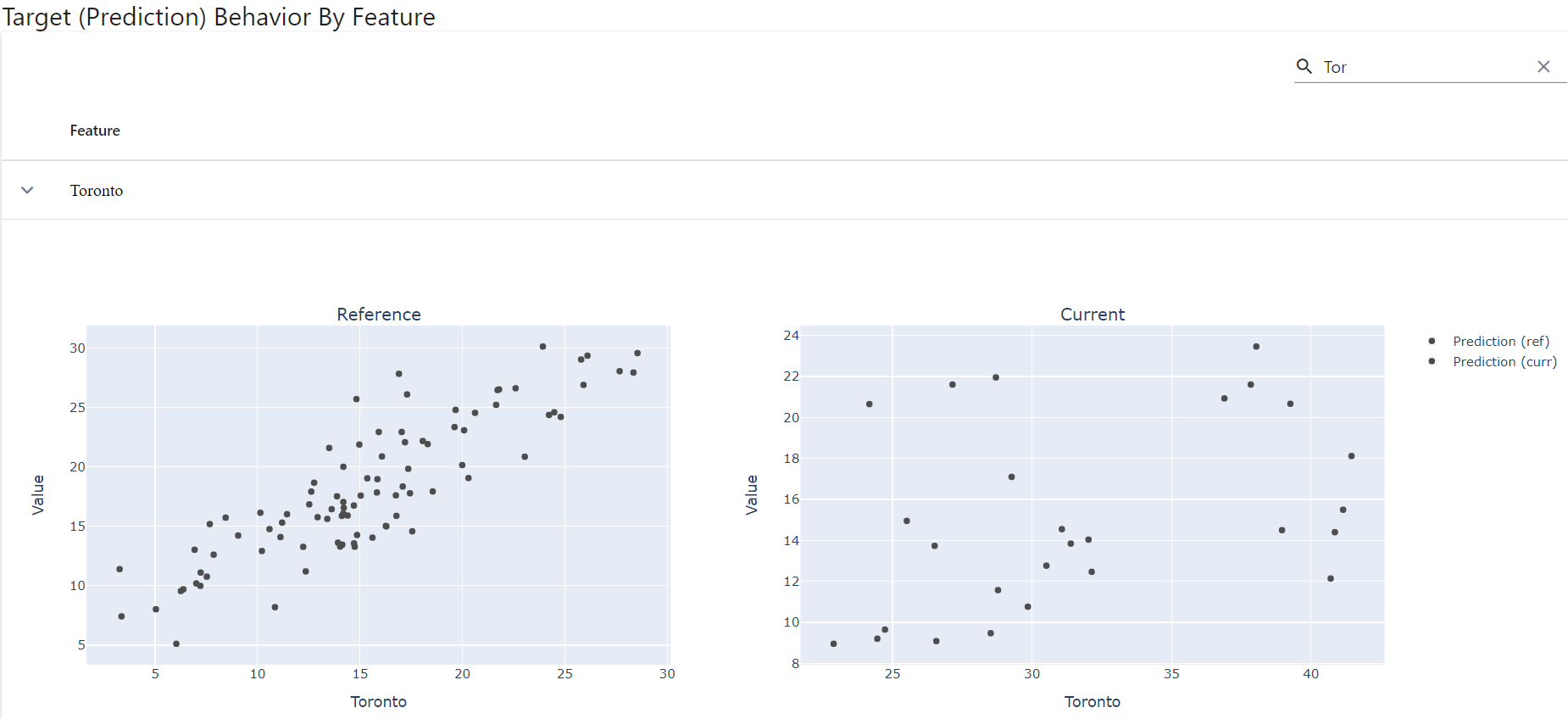
An example of this is shown below:



It can be seen how in the current dataset there is a shift in distrubiton. Further analysis is available, showing how predictions correlate with each of the features:



Evidently also showcases the prediction values, as well as the prediction behavior by feature (shown below):



Note the shift in axes.

**Use Cases for Evidently for Monitoring**:

While not providing an automated model monitoring system, and rather creating dashboards, Evidently is clearly useful in checking for data and target drift. Use cases may include early stages of model development, where there may not yet be a need to deploy a more complex constant monitoring solution. Additionally, models where data does not update frequently may be useful with evidently, as the dashboard creation can easily be built into a refresh process. Finally, evidently may be useful for small data science teams who may not want to license an expensive and complicated ML Ops solution. With its ease of use and strong visuals, Evidently is a great ML Ops package for monitoring drift.