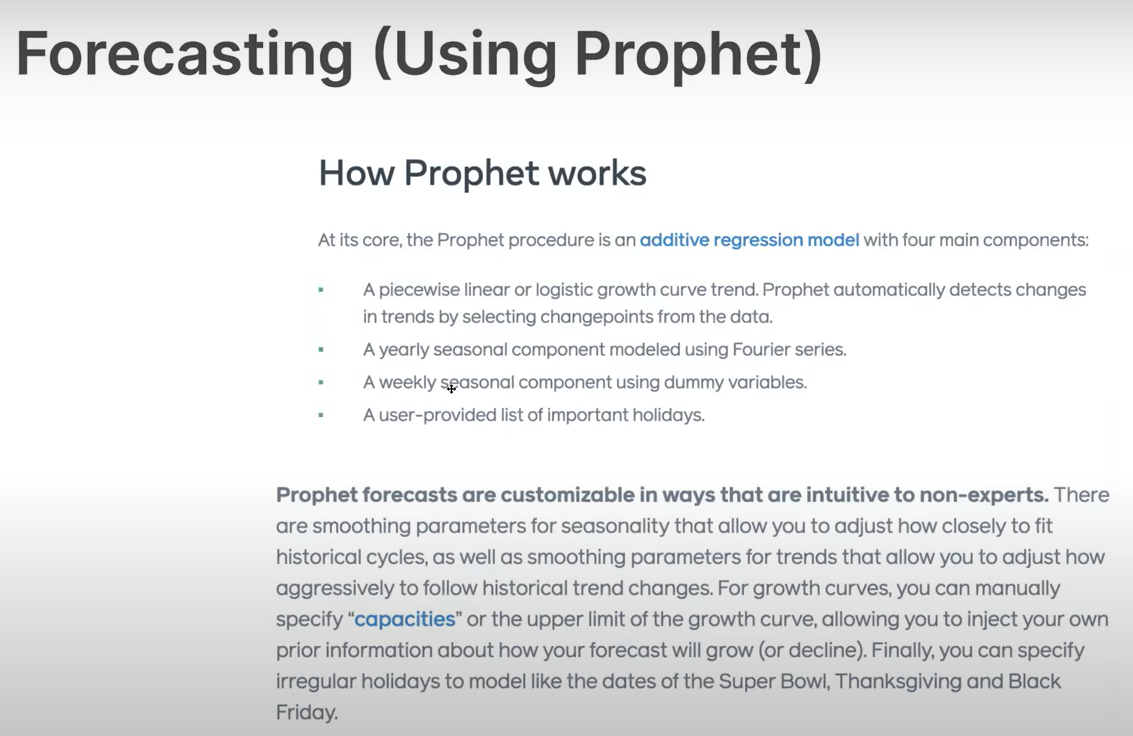
TECHNICAL NOTE ON THE FORECAST WITH PROPHET

Given that SparkML is not well suited for time series and after trials with several libraries for time series in analysis pyspark like flintTS, the complexity of their application and generalization due to the need of manual tuning has led us to try a faster and more autonomous approach with fb.prophet.



Prophet is an open-source forecasting software produced by Facebook’s data science team.

Models use trend, seasonality and user defined holidays. It uses a generalized additive model. The forecasting problem is treated as a curve-fitting exercise, which is different from time series models that explicitly account for the temporal dependence structure in the data.

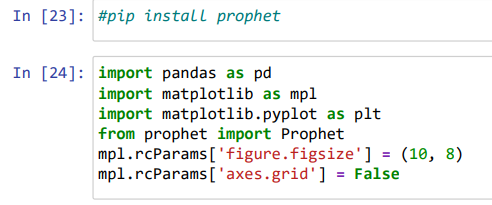
However, it is automatic, has easier to use and interpret parameters, fits faster, and allows different time spacing of data, missing data, etc.

It allows quick high-quality forecasts, and allows for finetuning so that experts can improve the results by adding specialized knowledge. It works better with time series that have strong seasonal effects.

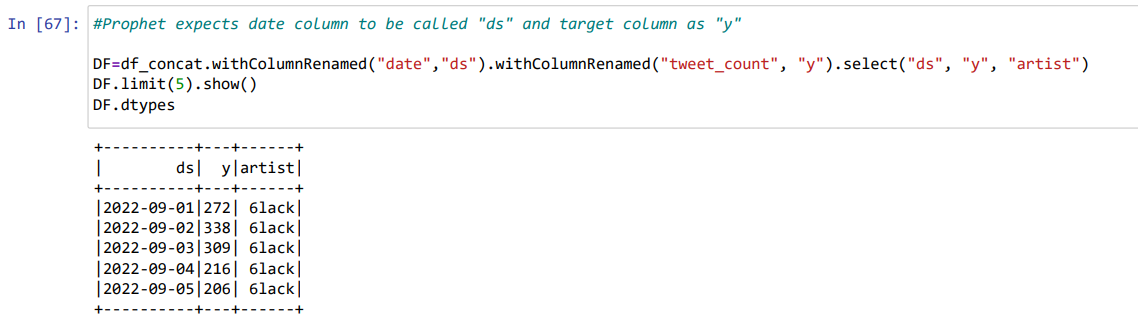
Can detect changes in trends and is robust with missing data.

The steps followed have been the following:

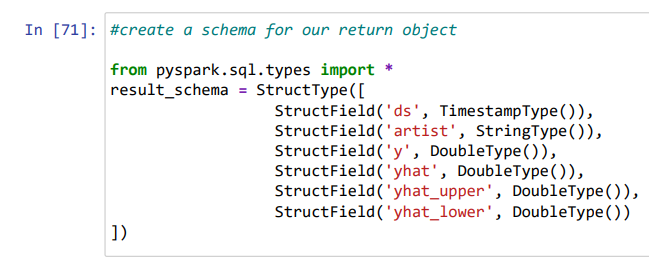
Installing and importing prophet from PyPI.



Prophet requires specific names in the data and target columns:



Create a schema for the returned object



As it can be seen below, a user defined function (UDF) has been created to

* Instantiate the model
* Configure the parameters
* Fit the model
* Configure the out of sample predictions
* And make the predictions

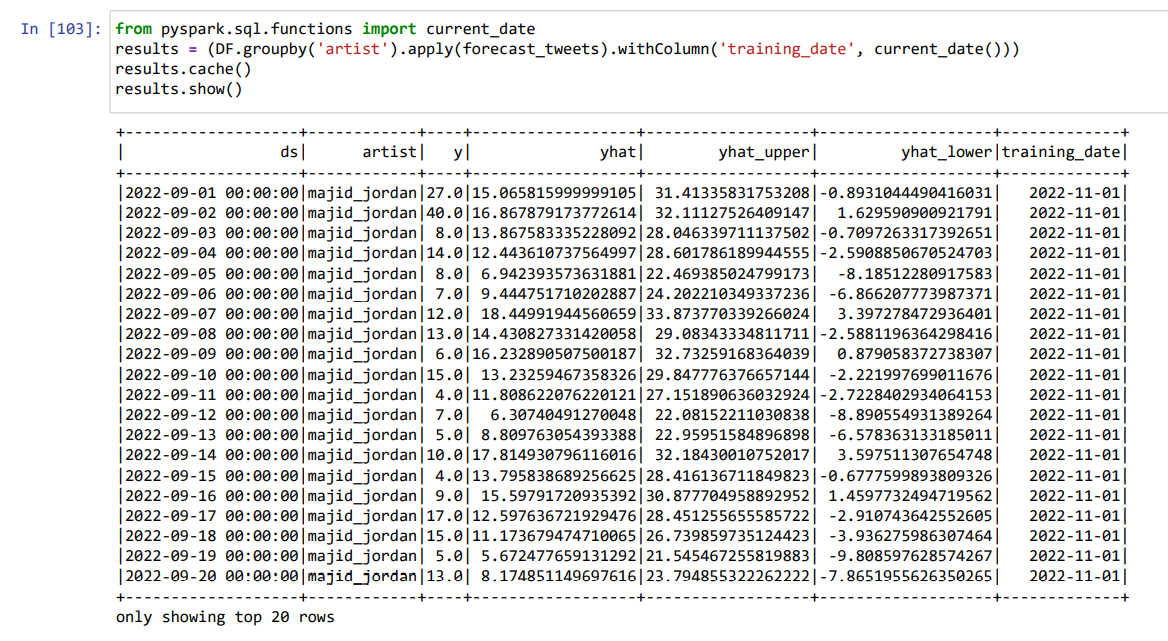
This allows its application easily and in a repetitive manner if required and facilitates the integration in the final version of the processing script.

A word on the initial configuration of the model is worth. Studies recommend a multiplicative model for the seasonality for events prone to viral effect like tweets. In our case, the application of that configuration led to a systematic exponential growth in tweets per day. While this effect is verified in more detail an additive model has been used because it projects a more realistic evolution of the daily tweets. This reduces the spikes but captures the trend.

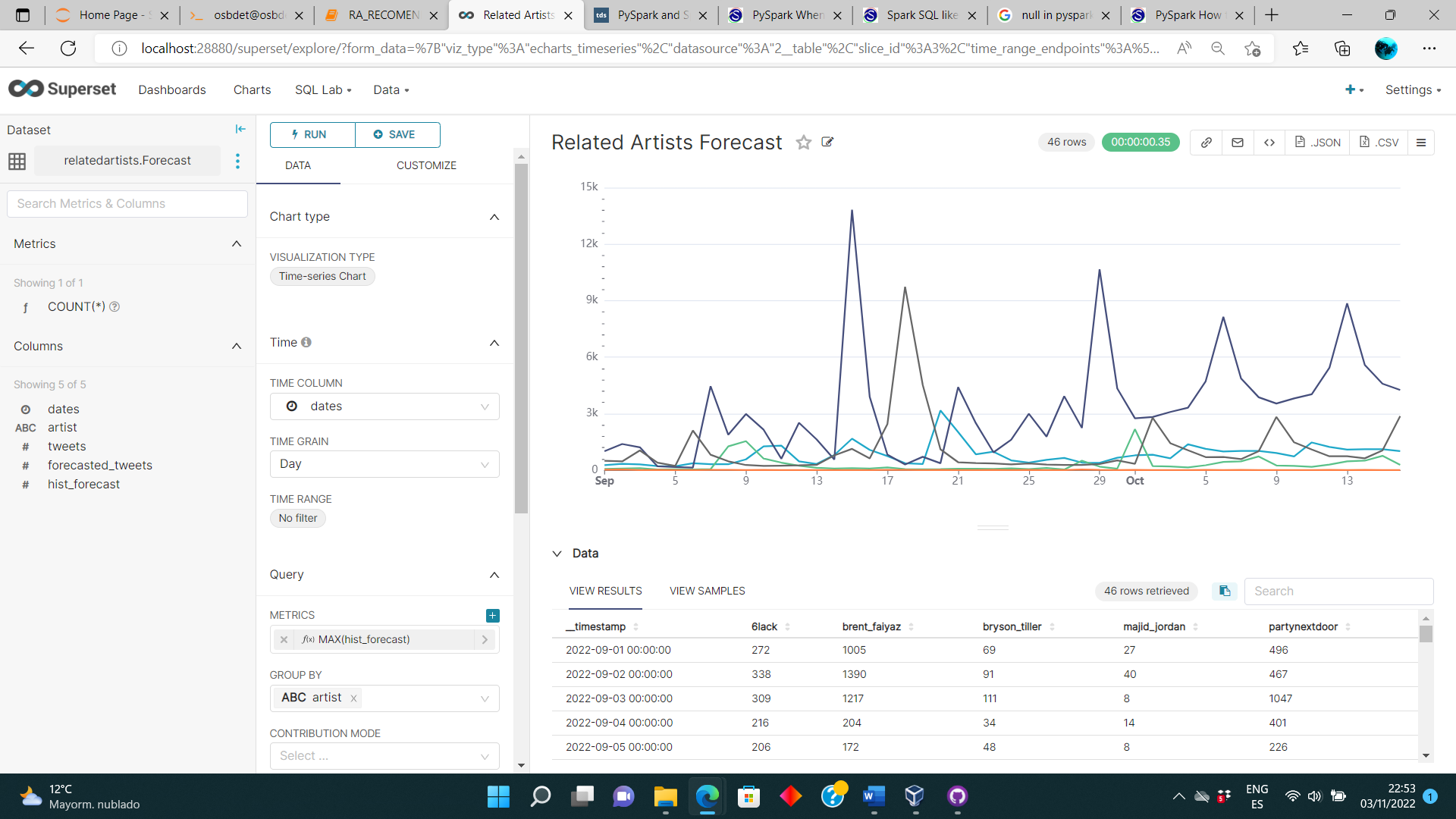
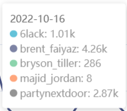
No external variables have been included however, other sources like google trend data could be used in the future if this allows a sounder forecast.



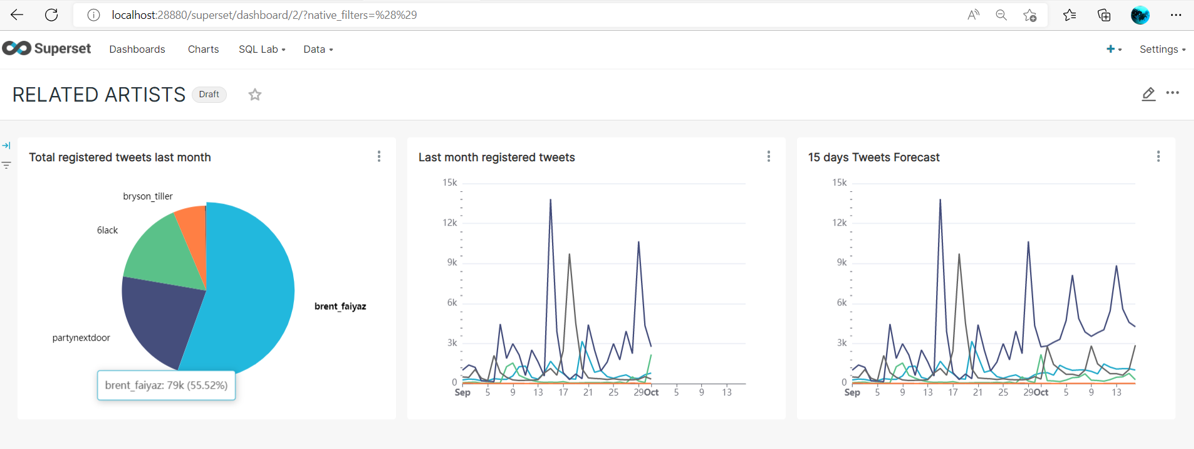
Finally, we apply the UDF to obtain the forecast and include the training date as a good practice.



After export to MariaDB as serving layer, some visualizations have been generated.



A provisional dashboard in Superset showing initial and forecasted series and the initial amount of tweets in a pie chart in the left can be seen below.



**Sources:**

[Quick Start | Prophet (facebook.github.io)](https://facebook.github.io/prophet/docs/quick_start.html)

[Time Series Forecasting With Prophet And Spark - Databricks](https://www.databricks.com/blog/2020/01/27/time-series-forecasting-prophet-spark.html)