Time series forecasting projects

Tom Rochette <tom.rochette@coreteks.org>

April 7, 2020 — c3292c86

1 Question

What are the general steps of a time series forecasting project?

2 Answer

2.1 Data profiling

Using a tool such as pandas-profiling, the dataset provided by the client is profiled and a variety of summary statistics produced, such as the min, mean, median, max, quartiles, number of samples, number of zeros, missing values, etc. are computed for numerical values. Other types of data also have their own set of properties computed.

These summary statistics allow you to quickly have a glance at the data. You will want to look for missing values to assess whether there's a problem with the provided data. Sometimes missing data can imply that you should use the prior value that was set. Sometimes it means that the data isn't available, which can be an issue and may require you to do some form of data imputation down the road.

2.2 Data analysis

Common things to look for in time series data are gaps in data (periods where no data has been recorded), the trend/seasonality/residual decomposition per time series, the autocorrelation and partial autocorrelation plots, distribution of values grouped by a certain period (by month, by week, by day, by day of the week, by hour), line/scatter plots of values grouped by the same periods.

2.3 Data cleanup

Data is rarely clean and ready to be consumed. This means many things: removing invalid values, converting invalid values or values out of range into a valid range, splitting cells that have multiple values in them into separate cells (e.g., "10 cm" split into "10" and "cm").

2.4 Data transformation

A variety of transformations can be applied to the cleaned data, ranging from data imputation (setting values where values are missing using available data), applying a function on the data, such as power, log or square root transform, differencing (computing the difference with the prior value), going from time zoned date time to timestamps, etc.

2.5 Feature generation

Common feature generation transformations are applied, such as computing lagged values on variables, moving averages/median, exponential moving averages, extracting the latest min/max, counting the number

of peaks encountered so far, etc. Feature generation is where you create additional information for your model to consume with the hope that it will provide it some signal it can make use of.

2.6 Establish a baseline

Before attempting to find a good model for the problem at hand you want to start with simple/naive models. The time series naive model simply predicts the future by using the latest value as its prediction.

2.7 Experiment

With a baseline established, you can now run a variety of experiments, which generally means trying different models on the same dataset while evaluating them the same way (same training/validation splits). In time series, we do cross-validation by creating a train/validation split where the validation split (i.e., the samples in the validation set) occurs temporally after the training split. The cross-validation split represents different points in time at which the models are trained and evaluated for their performance.

2.8 Performance analysis

After you've completed a few experiments you'll have a variety of results to analyze. You will want to look at your primary performance metric, which is generally defined as an error metric you are trying to minimize. Examples of error metrics are MAE, MSE, RMSE, MAPE, SMAPE, WAPE, MASE. Performance is evaluated on your validation data (out-of-sample) and lets you have an idea of how the model will perform on data it hasn't seen during training, which closely replicates the situation you will encounter in production.

2.9 Model selection

With many models and their respective primary metric computed, you can pick the one which has produced the lowest error on many cross-validation train/test splits.

2.10 Deployment

Once the model has been selected, it is packaged to be deployed. This generally implies something as simple as pickling the model object and loading it in the remote environment so it can be used to do predictions.

There are two modes of forecasting:

- Offline: Data used for forecasting is collected during a period of time and then a scheduled task uses this newly available data to create new forecasts. This is generally used for systems with large amounts of data where the forecasts are not needed in real-time, such as forecasting tomorrow's stock price, the minimum and maximum temperature, the volume of stocks that will be sold during the week, etc.
- Online: Data used for forecasting is given to the model and predictions are expected to be returned within a short time frame, on the order of less than a second to a minute.

Raw data is transformed and feature engineered, then given to the model to use to forecast.