SARIMA Model

1. Develop a Grid Search Framework

2. Case Study 1: No Trend or Seasonality

3. Case Study 2: Trend

4. Case Study 3: Seasonality

5. Case Study 4: Trend and Seasonality

1. Develop a Grid Search Framework

. This model has hyperparameters that control the nature of the model performed for the series, trend and seasonality, specifically:

* order: A tuple p, d, and q parameters for the modeling of the trend.
* seasonal order: A tuple of P, D, Q, and m parameters for the modeling the seasonality
* trend: A parameter for controlling a model of the deterministic trend as one of ‘n’, ‘c’, ‘t’, and ‘ct’ for no trend, constant, linear, and constant with linear trend, respectively.

We can start-off by defining a function that will fit a model with a given configuration and make a one-step forecast. The sarima forecast() below implements this behavior.

The function takes an array or list of contiguous prior observations and a list of configuration parameters used to configure the model, specifically two tuples and a string for the trend order, seasonal order trend, and parameter. We also try to make the model robust by relaxing constraints, such as that the data must be stationary and that the MA transform be invertible.

One important modification to the framework is the function used to perform the walk-forward validation of the model named walk forward validation(). This function must be updated to call the function for making an SARIMA forecast. The updated version of the function is listed below.

The only thing left to do is to define a list of model configurations to try for a dataset. We can define this generically. The only parameter we may want to specify is the periodicity of the seasonal component in the series, if one exists. By default, we will assume no seasonal component. The sarima configs() function below will create a list of model configurations to evaluate. The configurations assume each of the AR, MA, and I components for trend and seasonality are low order, e.g. off (0) or in [1,2]. You may want to extend these ranges if you believe the order may be higher. An optional list of seasonal periods can be specified, and you could even change the function to specify other elements that you may know about your time series. In theory, there are 1,296 possible model configurations to evaluate, but in practice, many will not be valid and will result in an error that we will trap and ignore.

To apply a SARIMA model, we will implemenet the Box-Jenkins Method:

Tabla

Descripción generada automáticamente con confianza media

Step 1: Identify

Is the time series stationary?

If not stationary, which transformation should we apply to make it stationary?

Is the time series seasonal?

If seasonal what is the seasonal period?

Which orders to use? (p for AR, q for MA)

From the decomposition above we can conclude:

* There is an upward trend on the price. Therefore, this time series is not stationary.
* From the seasonal component we can observe that the model is additive, since the seasonal component is similar (not getting multiplied) over the period of time.
* Also, we can observe on the seasonal component seasonality in sales with lower sales in January and higher sales in July.

ACF and PACF plots can help us find appropriate values for parameters p and q . However, the interpretation of these plots is not always clear. To obtain more assurance to our choices we can apply an empirical method. This method consists on fitting the ARIMA model for different values of p and q, and choosing the best value based on metrics such as AIC and BIC.

AIC (Akaike information criterion) is a metric which tells us how good a model is. Lower the value, better the model. The AIC also penalizes models which have lots of parameters. This means if we set the order too high compared to the data, we will get a high AIC value. This stops us overfitting to the training data.

BIC (Bayesian information criterion) is similar to AIC, therefore lower value means a better model. However, BIC penalizes additional model orders more than AIC. As consequence, BIC will sometimes suggest a simpler model.

LSTM model

The hyperparameters for the LSTM model will be the same five as the MLP; they are:

*  n input: The number of prior inputs to use as input for the model (e.g. 12 months).
*  n nodes: The number of nodes to use in the hidden layer (e.g. 50).
*  n epochs: The number of training epochs (e.g. 1000).
*  n batch: The number of samples to include in each mini-batch (e.g. 32).
*  n diff: The difference order (e.g. 0 or 12)