

# Dokumentacja

## Sarima

Metody w konstruktorze przyjmują dwie wartości:

*y*: szereg czasowy, dla którego dopasowany zostanie model. Szereg powinien mieć indeks miesięczny (MS - month start)

*horizon*: wartość większa od 0 lub None. W przypadku wartości liczbowej oznacza to, z jakich danych może co najwyżej korzystać model podczas wykonywania predykcji. Przykładowo, obliczając predykcję na marzec 2020 z horyzontem 1 miesiąc, model będzie wykorzystywał dane do lutego 2020. Jednak w przypadku horyzontu 2, model wykorzysta co najwyżej dane ze stycznia 2021. Wyjątkiem jest jedynie sytuacja, gdy dane te zostają przekazane jako szereg treningowy w konstruktorze.

Przykładowo, jako dane treningowe wykorzystano dane 01.01.2017-01.05.2019. Wykonując teraz predykcję dla 01.06.2019 z horyzontem 2 zostaną wykorzystane dane do 01.05.2019, dla 01.07.2019, także zostaną wykorzystane dane do 01.05.2019, natomiast dla 01.08.2019 zostaną już wykorzystane dane do 01.06.2019 (o ile zostały one przekazane w parametrach metody predict). Uzasadnieniem takiego postępowania jest założenie, że dane treningowe powinny stanowić minimalny zasób danych dostępnych dla modelu, a nowe napływające dane mogą go co najwyżej uzupełniać.

W przypadku wartości None, model wykorzystuje jedynie dane treningowe do wykonywania predykcji.

- `plot_acf` oraz `plot_pacf`

Obie powyższe metody rysują odpowiednie wykresy dla danych treningowych przekazanych w konstruktorze

- `hyperparameter_search_fit`

Przeszukuje wszystkie kombinacje parametrów p, d, q, P, D, Q z wybranym s oraz wybiera najlepsze hiperparametry w kontekście przekazanej metryki (dla której mamy minimalne wartości metryki). W przypadku mse (mean squared error) konieczne jest wydzielenie zbioru walidacyjnego. W tym celu wykorzystywany jest parametr „split\_fraction”. Przykładowo, dla domyślnej wartości 0.8, pierwsze 80% obserwacji zostanie przydzielonych do zbioru treningowego, natomiast pozostałe 20% do zbioru walidacyjnego na którym będzie szacowany błąd modelu.

Wybrane parametry służą do budowania modelu. Nie jest konieczne wywoływanie metody fit.

- fit

Drugi sposób po `hyperparameter_search_fit` na ustawienie odpowiednich parametrów aby móc wykonywać predykcje.

- predict

Metoda `predict` w dużej mierze wykorzystuje obiekt `test_data`. Jest to `pd.Series` z indeksem, dla którego zostanie wygenerowana predykcja. Szereg może zawierać `NaNy`. Jest to przydatne w sytuacji, kiedy chcemy obliczyć predykcję dla miesięcy przyszłych, dla których nie ma jeszcze danych. Dane tego szeregu wykorzystywane są jedynie w przypadku `horizon` ustawionego na wartość liczbową większą od zera w konstruktorze (czyli inną niż `None`). W takiej sytuacji wykonując predykcję dla pewnego momentu, algorytm będzie próbował uzupełnić dane treningowe o `test_data`, które poprzedzają bieżący moment o co najmniej tyle miesięcy ile wynosi `horizon`. Za każdym razem tworzony będzie nowy model, jednak wszystkie te modele zachowają te same parametry `order` oraz `seasonal_order`.

**Warto pamiętać, że zbiór treningowy nie jest nigdy ograniczany. Czyli wykonując predykcję na luty 2020, jeśli w danych treningowych był dostępny styczeń 2020 to zostanie on wykorzystany do wykonania predykcji nawet jeśli horyzont jest większy niż 1.**

W przypadku ustawienia `plot=True`, rysowany jest wykres porównujący predykcję wraz z przedziałami ufności do rzeczywistych wartości.

Metoda zwraca predykcję w formie `pd.Series` z indeksem jednakowym jak `test_data`

- analyse\_results

Zwraca różne przydatne informacje na temat modelu wytrenowanego na zbiorze treningowym przekazany w konstruktorze. Wyświetlany jest wykres diagnostyczny oraz wypisywane podsumowanie w terminalu.

## Prophet

Metody w konstruktorze przyjmują dwie wartości:

*y*: szereg czasowy, dla którego dopasowany zostanie model. Szereg powinien mieć indeks miesięczny (`MS - month start`)

**horizon:** wartość większa od 0 lub None. W przypadku wartości liczbowej oznacza to, z jakich danych może co najwyżej korzystać model podczas wykonywania predykcji. Przykładowo, obliczając predykcję na marzec 2020 z horyzontem 1 miesiąc, model będzie wykorzystywał dane do lutego 2020. Jednak w przypadku horyzontu 2, model wykorzysta co najwyżej dane ze stycznia 2021. Wyjątkiem jest jedynie sytuacja, gdy dane te zostają przekazane jako szereg treningowy w konstruktorze.

Przykładowo, jako dane treningowe wykorzystano dane 01.01.2017-01.05.2019. Wykonując teraz predykcję dla 01.06.2019 z horyzontem 2 zostaną wykorzystane dane do 01.05.2019, dla 01.07.2019, także zostaną wykorzystane dane do 01.05.2019, natomiast dla 01.08.2019 zostaną już wykorzystane dane do 01.06.2019 (o ile zostały one przekazane w parametrach metody predict). Uzasadnieniem takiego postępowania jest założenie, że dane treningowe powinny stanowić minimalny zasób danych dostępnych dla modelu, a nowe napływające dane mogą go co najwyżej uzupełniać.

W przypadku wartości None, model wykorzystuje jedynie dane treningowe do wykonywania predykcji.

**lb oraz ub:** kolejno dolne oraz górne ograniczenie na dane. Jeśli przykładowo przewidujemy zyski ze sprzedaży produktu, rozsądnym podejściem może być ustawienie lb na 0.

- **hiperparameter\_search\_fit**

Przeszukuje wszystkie kombinacje parametrów `seasonality_mode` oraz `changeoint_prior_scale`. W tym celu wykorzystuje mse (mean squared error) obliczane na wydzielonym zbiorze walidacyjnym, którego wielkość można kontrolować za pomocą parametru „`split_fraction`”. Przykładowo, dla domyślnej wartości 0.8, pierwsze 80% obserwacji zostanie przydzielonych do zbioru treningowego, natomiast pozostałe 20% do zbioru walidacyjnego na którym będzie szacowany błąd modelu. Wybierana jest ta kombinacja parametrów, która daje najmniejszą wartość błędu na zbiorze walidacyjnym.

Wybrane parametry służą do budowania modelu. Nie jest konieczne wywoływanie metody fit.

Z dokumentacji fbprophet:

**changeoint\_prior\_scale:** This is probably the most impactful parameter. It determines the flexibility of the trend, and in particular how much the trend changes at the trend changeoints. As described in this documentation, if it is too small, the trend will be underfit and variance that should have been modeled with trend changes will instead end up being handled with the noise term. If it is too large, the trend will overfit and in the most extreme case you can end up with the trend capturing yearly seasonality. The default of 0.05 works for many time series, but this could be tuned; a range of [0.001, 0.5] would likely be about right. Parameters like this (regularization penalties; this is

effectively a lasso penalty) are often tuned on a log scale.

**seasonality\_mode:** Options are ['additive', 'multiplicative']. Default is 'additive', but many business time series will have multiplicative seasonality. This is best identified just from looking at the time series and seeing if the magnitude of seasonal fluctuations grows with the magnitude of the time series (see the documentation here on multiplicative seasonality), but when that isn't possible, it could be tuned.

- fit

Drugi sposób po `hyperparameter_search_fit` na ustawienie odpowiednich parametrów aby móc wykonywać predykcje.

- predict

Metoda `predict` w dużej mierze wykorzystuje obiekt `test_data`. Jest to `pd.Series` z indeksem, dla którego zostanie wygenerowana predykcja. Szereg może zawierać `NaNy`. Jest to przydatne w sytuacji, kiedy chcemy obliczyć predykcję dla miesięcy przyszłych, dla których nie ma jeszcze danych. Dane tego szeregu wykorzystywane są jedynie w przypadku horizon ustawionego na wartość liczbową większą od zera w konstruktorze (czyli inną niż `None`). W takiej sytuacji wykonując predykcje dla pewnego momentu, algorytm będzie próbował uzupełnić dane treningowe o `test_data`, które poprzedzają bieżący moment o co najmniej tyle miesięcy ile wynosi horizon. Za każdym razem tworzony będzie nowy model, jednak wszystkie te modele zachowają te same parametry modelu ustawione podczas jego dopasowania.

**Warto pamiętać, że zbiór treningowy nie jest nigdy ograniczany. Czyli wykonując predykcję na luty 2020, jeśli w danych treningowych był dostępny styczeń 2020 to zostanie on wykorzystany do wykonania predykcji nawet jeśli horyzont jest większy niż 1.**

W przypadku ustawienia `plot=True`, rysowany jest wykres porównujący predykcję wraz z przedziałami ufności do rzeczywistych wartości.

Metoda zwraca predykcję w formie `pd.Series` z indeksem jednakowym jak `test_data`.

Parametr `verbose` kontroluje stopień logowania informacji. Domyślny 0 nie zwraca żadnych informacji do konsoli.