Szeregi czasowe 1 - wstęp

Szeregi czasowe, to sekwencje danych **uporządkowanych** ze względu na czas.

**Zmienne egzogeniczne** są to dodatkowe zmienne, które może dodać przy predykcji szeregu, które mogą pomóc w określeniu wyniku. Są to zmienne niezależne, czyli ta zmienna ma wpływ na szereg czasowy, ale nie na odwrót.

Przykład: W prognozowaniu sprzedaży lodów temperatura powietrza jest zmienną egzogeniczną, bo wpływa na sprzedaż, ale sprzedaż lodów nie wpływa na temperaturę.

**Wielowymiarowe szeregi czasowe (multivariate)** – zawierają wiele równoległych szeregów czasowych np. zyski i koszty. Które mogą zależeć od siebie lub nie.

**Zadania:**  
1. Regresja - Szeregi czasowe

Punkowa – przewidywanie przyszłych punktów

Przedziałowa – przewidywanie przedziału wartości jakie może model osiągnąć

2. Klasyfikacja - Szeregi czasowe

Klasyfikacja całego szeregu -> np. czy na podstawie danego szeregu aktywności można stwierdzić, że pacjent jest chory

3. Wykrywanie anomalii – ataki DDS

4. Segmentacja – np. wykrywanie kto mówi

5. Klasteryzacja – grupowanie szeregów

# Rodzaje modeli:

**Heurystyki:**

Średnia wartość, ostatnia wartość, mediana

Bardzo proste, dobre jako baseline.

Naiwna sezonowość -> skopiowanie wartości z poprzedniego sezonu (np. roku)

**Model statystyczne:**

ARIMA, ETS

**Proste modele deep learning:**

Kombinacja statystyk i prostego MLP

Dobre do trudnych danych i szeregów wielowymiarowych

Np. N-BEATS, N-HiTS, TSMixer, LSTM, GRU

**Złożone modele deep learning:**

Nowe I jeszcze nie tak sprawdzone

TFT, PatchTST, Chronos, TimesFM, TimeGPT

# Przygotowanie danych

Dane mają swoją **częstotliwość**, co, ile czasu są kolejne dane. Wszystkie metody wymagają pojedynczej częstotliwości – wszystkie dane w tych samych odstępach czasowych.

**Problemy:**

- Brakujące dane

- Za dużo danych, zbyt duża częstotliwość -> zbyt duża wariancja (szum)

- Intermittent – dużo zer

**Rozwiązania problemów:**

**Forward fill** (last value interpolation/ previous value interpolation) – uzupełnianie ostatnią wartością -> zastosowanie w regresji i klasyfikacji

**Interpolation** (linear, polynomial, ipt) – uzupełnienie na podstawie przeszłej i przyszłej wartości – zastosowanie tylko w klasyfikacji, wymaga wartości z przyszłości!

**Resampling –** agregowanie, żeby zmniejszyć częstotliwość, łączymy minuty w godziny

# Dekompozycja szeregów czasowych

Data = trend + seasonality + cycle + remainder

**Trend** (trend) – czy szereg na wykresie generalnie rośnie czy maleje, ogólny trend

**Sezonowość** (seasonality) – powtarzające się, okresowe zmiany (równe okresy) np. co tydzień sprzedaż spada do 0, bo niedziela niehandlowa (zawsze 7 dni).

**Cykl** (cycle) – powtarzające się, nierówne okresy – np. cykle koniunkturalne -> raz się gospodarka rozwija, a raz kurczy, trudno przewidzieć, kiedy

**Reszta** (remainder) – zmiany polityczne, klęski żywiołowe, ustawy

Często ciężko wykryć cykl, więc wtedy zakłada się, że dane składają się z 3 komponentów: data = ‘trend-cycle’ + seasonality + remainder. Łączy się trend z cyklem.

Obraz zawierający tekst, Czcionka, diagram, Wykres

Opis wygenerowany automatycznie

**Dekompozycja addytywna**

Niezależne elementy

-> trend + sezonowość + reszta

**Dekompozycja multiplikaty**

Zależne elementy, im większy jeden element, tym drugi też większy. Np. im większy sklep (trend ), to sezonowo przed black Friday też wzrasta (sezonowość )

Zamiana multiplikatywnego na addytywny za pomocą przekształcenia logarytmicznego.Obraz zawierający tekst, Czcionka, pismo odręczne, linia

Opis wygenerowany automatycznie

Multiplikatywna -> trend rośnie, to sezonowość też (coraz większe hopy)

Addytywna -> trend rośnie, ale piki takiej samej wielkości

Dekompozycja STL – algorytm

Rozdziela szereg na: trend + seasonality + remainder

Obraz zawierający tekst, linia, diagram, Czcionka

Opis wygenerowany automatycznie

# Cechy szeregu

**Stacjonarność silna**

Niezależnie jaki podzbiór danych, każdy z nich ma taki sam rozkład danych, taką samą średnią, odchylenie standardowe (idealne założenie, trudne do spełnienia)

Np. Biały szum gausowski

**Stacjonarność słaba** (weak/ wide-sense stationarity) – realna do spełnienia w praktyce

„ładny prosty szereg”

Średnia i autokowariancja są stałe, wariancja jest stała

Jak się narysuje szereg, to z grubsza wygląda wszędzie tak samo, nie ma trendu ani sezonowości.

Chcemy Homoskedastyczność – wariancja stała, cały czas taka sama.

1. Stacjonarny
2. Niestacjonarny, bo ma trend rosnący
3. Niestacjonarny, bo wariancja rośnie

Obraz zawierający tekst, Czcionka, linia, diagram

Opis wygenerowany automatycznie

**Autokowariancja –** kowariancja z opóźnieniem k kroków, miara powiązania przeszłych i przyszłych wartości.

**Autokorelacja –** autokowariancja / odchylenie standardowe

Wysoka autokowariancja dla k -> duża zależność przyszłych wartości od przeszłych

## Ocena stacjonarności

Manualna: wykresy **ACF**, **PACF** (dużo ludzi tak robi)

Testy statystyczne:

- **KPSS** (Kwiatkowski-Phillips-Schmidt-Shin) – najlepszy w praktyce

- **ADF** (Augmented Dickey-Fuller) – tradycyjny, trochę gorszy

# Transformacje – Stacjonaryzacja

1. **Trend** -> różnicowanie – zmiana wartości na delty

Zwykle stosuje się max dwa różnicowania. Trzeba potem pamiętać, żeby odwrócić dane. Żeby przewidywać wartości, a nie różnice.

Automatyczne usuwanie trendu:

Różnicowanie, dopóki nie wyjdzie testem KPSS, że szereg jest stacjonarny.

1. **Sezonowość m** -> sezonowe różnicowanie

Tak jak zwykłe różnicowanie, ale inne testy np. OCSB, usuwanie m wartości do tyłu, a nie tylko jedną

Można usunąć różne sezonowości, jeśli jest kilka na raz (dzień i rok)

1. **Niestabilna wariancja/kowariancja** -> log, sqrt, box-cox transformacje

Cel: stabilizacja szeregu/ wygładzenie

- Zmniejszenie heteroskedastyczności

- Rozkład wartości bliżej normalnego

- Zmienia relacje multiplikatywne w addytywne

- Stabilność numeryczna -> mniejsze wartości

**Transformacja Box-Cox:**

Obraz zawierający tekst, Czcionka, biały, linia

Opis wygenerowany automatycznie

Plusy:

- Model uczy się lambdy, dlatego często nowe transformacje np. dla lambda 0.25.

- Zautomatyzowany, często daje najlepsze wyniki.

Minusy:

- Wymaga dodatnich wartości (często dodaje się małe wartości, byleby nie było zer)

- Nie uwzględnia sezonowości

# Time series CV

**k-fold split –** podział na k kawałków po kolei. Modele neuronalne słabo radzą sobie w pierwszych foldach -> oszukiwanie, że gorzej sobie radzą w porównaniu z innymi metodami.

- Okno treningowe zawsze tego samego rozmiaru (np. 6 miesięcy)

- Foldy ten sam rozmiar. (fold=trening+test)

- Nie bierze długoterminowych trendów

- Bardziej stabilny dla krótkich przedziałów czasu.

**Exapanding window:**

1. **Początkowy rozmiar danych** – np. 50% danych to zbiór treningowy
2. Przewidywanie następnych wartości - **forecasting horizon** (ile wartości do przodu np. 3 miesiące)
3. Zwiększamy zbiór treningowy o **krok retreningu** (ile kroktów do przodu) (np. 7 dni) i znowu przewidujemy

Plusy: - elastyczny i praktyczny kontekst, dobry do testowania

- można elastycznie wybrać rozmiar horyzontu prognozowania oraz kroku retreningu (horyzont np. 3 miesiące, a retrening co 7 dni)

- dobry do modeli, które potrzebują więcej danych do nauki

Minusy: - wymagające obliczeniowo

# Metryki

**1. Klasyczne:** RMSE, MAE, GMAE(geometric), MdAE(median)

Czułe na outliery, których jest dużo w szeregach

2. **Relatywne do y:** MAPE, SMAPE (symetryczne)

Procentowe, dużo problemów ☹

3.**MASE**, interpretowalny, symetryczny, [0, 1], dobre 😊

# Analiza rezydułów

Błąd rezydualny – różnica pomiędzy wartością prawdziwą i przewidywaną: **Założenia dotyczące rezydułów idelanego modelu:**

Rezyduła powinny być białym szumem gassowskim:

1. Wycentrowane w 0, średnio model myli się tak samo w górę i w dół
2. Mają rozkład normalny (mierzy się testem Shapiro-Wilk lub Anderson-Darling)
3. Homoskedastyczność - brak autokorelacji – przyszłe wartości nie zależą od przeszłych (test Ljung-Box)

Szeregi czasowe 2 – modele statystyczne

# ARIMA

* Rodzina modeli autokorelacyjnych
* Założenie: model stacjonarny

1. Literka **I(d)** – różnicowanie, d – stopień
2. **AR(p)** – autoregresja – korelacja obecnych wartości od p poprzednich wartości. Początkowe wartości uznaje się za zera albo nie liczy się dla nich.
3. **MA(q) –** koryguje model, gdy model zaczyna mylić się w góre, to to, koryguje w dół, dopasowuje się (trening) ->

Zakłada się, że błędy są z rozkładu normalnego, więc wzór na MA(q) to ważona suma rozkładów normalnych, co można wyestymować za pomocą MLE

q-to liczba błędów wstecz

**! Nie mylić MA z:**

średnią ruchomą

eksponentalna ważona średnia ruchoma (EWMA)

**ARMA(p, q)**

Obraz zawierający tekst, Czcionka, paragon, zrzut ekranu

Opis wygenerowany automatycznie

**Trening:**

- typowo MLE, trenujemy stałą c, wagi: fi, theta

**-** metodaHannan-Rissanen: w pętli: trenujemy AR(p) jako regresję liniową, obliczamy błędy pomiędzy rzeczywistymi danymi i przewidzianymi, trenujemy MA(q) jako zwykłą regresję liniową -> potem zwykły MLE, albo powtarzanie metodę H-R

**Zalety:**

- Estymacja MLE

- przedziały ufności

- probabilistyczne prognozy

**AutoARIMA:**

Parametr d dobiera się testem statystycznym

a następnie p i q za pomocą grida

używa metryki AIC

**AIC**

Sprawdza jakość modelu bez zbioru walidacyjnego, liczy się tylko na zbiorze treningowym

Im niższy tym lepiej

Uwzględnia poziom skomplikowania modelu p+q, im prostszy tym lepiej

Niski koszt do policzenia

Wzorek na to, uwzględnia prawdopodobieństwo jak dobrze dane są dopasowane

Obraz zawierający tekst, Czcionka, biały, paragon

Opis wygenerowany automatycznie

NLL – negative log likelihood

**Stepwise selection**

Obraz zawierający zrzut ekranu, krąg, Wielobarwność, diagram

Opis wygenerowany automatycznie

Buduje się siatkę możliwych parametrów. Wartości p i q nie powinny być zbyt duże, bo będzie overfitting. Zaczyna się od sprawdzenia (0, 0), (0, 1), (1, 0), (2, 2) – bo one dają zwykle najlepsze wartości. Wybiera się ten z najniższym AIC -tak jak w przypadku niebieskiego: liczy się AIC dla wszystkich sąsiadów i wybiera ten z najniższym, a następnie dla jego sąsiadów liczy się AIC, aż wszyscy sąsiedzi będą mieć większe AIC niż środkowy.

**SARIMA**

1. Różnicowanie sezonowe
2. Różnicowanie zwykłe
3. Estymacja wzoru

+: uwzględnia sezonowość

-: koszty obliczeniowe 4 parametry do oszacowania/ 4D grid do przeszukiwania

ARIMA słaba do długich szeregów.

**SARIMAX**

Dodatkowe zmienne, czy jest weekend (0/1) – proste, statyczne (rozmiar sklepu) lub dynamiczne(ifweekend)

Dodawane przez feature engineering

**+**: zwykle rezultaty dużo lepsze, bo więcej wiedzy

**-**: koszt obliczeniowy, dodatkowa regresja i feature engineering

ARIMA domyślnie przewiduje jedną wartość do przodu. Założenie, że jeden krok do przodu przewidziany jest poprawny i dodajemy go do zbioru treningowego.

Model zawsze robi błąd - error compounding, bo opiera się na przeszłych wartościach.

**ARIMA plusy:**

+ Dużo wariantów

+ Dobrze radzi sobie z silnymi autokorelacjami.

**ARIMA minusy:**

- Założenie o stacjonarności

- Problemy ze złożonymi trendami i sezonowościami

- kosztowne obliczeniowo

- Tylko addytywne modele

- Potrzeba preprocessingu

# ETS (Exponential Smoothing/ Error trend seasonality)

Modeluje średnią, trend, sezonowość i rezydua

**Exponential weighting** – nowsze wartości ważniejsze niż wcześniejsze

Brak założenia o stacjonarności, nie trzeba preprocessingu

Dobrze działają dla małych danych, bardzo szybkie

**Model Holta (pojedyncze wygładzanie)**

Uwzględnia trend liniowy, prostą

**Damped trend (podwójne wygładzanie)**

Uwzględnia trend, tłumi się po jakimś czasie – jest coraz bardziej poziomy

Pomału przestaje rosnąć, zamiast rosnąć w nieskończoność

Lepszy do prognoz długoterminowych

Dodaje współczynnik tłumienia, coraz większy z czasem

Niebieski – Holt

Pomarańczowy – Damped trend

Obraz zawierający tekst, zrzut ekranu, Wykres, diagram

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

**Model Holt-Winters (portrójne wygładzanie)**

Uwzględnia trend i sezonowość

Krótkie sezonowości (miesięczne, roczne)

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Trening MLE

**ETS vs ARIMA**

ETS jest addytywny lub multiplikatywny

ARIMA jest tylko addytywny

ARIMA dobra do skomplikowanych szeregów

Wszystkie addytywne ETS można zapisać jako ARIMA, ale trzeba preprocessing

ETS lepszy do długich szeregów, bo ma damped trend

ETS dużo szybszy

ETS nie wspiera zmiennych egzogenicznych

**ETS gridsearch:**

**E**rror: additive (A), multiplicative(M)

**T**rend: no trend(N), additive(A), additive damped trend(Ad)

**S**easonality: no seasonality(N), additive(A), multiplicative(M)

Trend multiplikatywny raczej się nie robi, bo niestabilny.

Błąd addytywny - jeśli chcemy punkty. Błąd multiplikatywny – przedział ufności

**Przewidywanie oparte o regresję**

Zmienne dostępne w momencie przewidywania

**LightGBM** – najczęściej najlepsze, wymaga trochę więcej danych

Modele liniowe. LAD, Huber - prostrze

**Cechy związane z oknami przesuwnymi**

Statystyki ostatnich wartości: Średnia, odchylenia, kwantyle itd dla każdego okna czasowego – feature engineering

Potrzebne do nauki modeli regresji

Daty – **zmienne** **eventowe** – święta, kwartały

Uwzględnięcie konkretnych dat np. święta, weekend itp. w sprzedaży

**Cechy cykliczne:** 1 i 23 jako godziny są blisko, chociaż jako liczby daleko

Zakodowywanie dat i czasu

Naiwne podejścia:

One-hot-encoding miesiąca: gdy ważne czy to jest grudzień, czy miesiące zimowe

**Kodowanie cykliczne:** zakodowaniejako baza trygonometryczna, dobre do zakresów czasu, 1 i 23 godzina. Podział cechy na sinusa i cosinusa i liczenie kąta w radianach.

Obraz zawierający diagram, linia, zrzut ekranu, Wykres

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

**Cechy fourierowskie**

Sezonowość jako sinusy i cosinusy o różnych okresach

Sezonowość nie musi być liczbą całkowita, np. liczba dni w roku jest 365,25 i jeszcze trochę

Regresja harmoniczna

Sezonowość powtarzalna / periodyczna

Szereg z okresem m, tworzy parę sinusa i cosinusa

Kilka cech fourierowskich

**Prognozowanie oparte o regresję**

**Zalety:**

- Zakodowanie dodatkowych zmiennych

- często najlepsze wyniki z dobrym feature engineering

- złożone patterny, długie zależności czasowe

- wiele algorytmów do wyboru, ale lightGBM zwykle najlepszy

**Wady:**

Trzeba zrobić feature engineeering

MLOps complexity, pipeliny

LightGBM łatwo overfittuje

DeepLearning słaby, gdy małe zbiory danych, wolne, niestabilne, overfittują, duże wymagania obliczeniowe

Statystyczne: szybkie, zautomatyzowane, bliskie SOTA

# Wykład 3 – neuronowe metody do szeregów czasowych