AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Praca dyplomowa

Klasyfikacja szeregów czasowych z wykorzystaniem funkcjonalnej regresji logistycznej w wariancie binarnym i wieloklasowym

Time series classification using functional logistic regression in binary and multiclass variants

Autor: *Jakub Poręba*
Kierunek studiów: *Automatyka i robotyka*
Opiekun pracy: *dr hab. inż. Jerzy Baranowski, prof. AGH*

Kraków, 2022

Spis treści

Bibliografia 22

1. Wstęp.

2. Cel i założenia pracy.

2.1. Schemat działania aplikacji

2.2. Publiczność i otwartoźródłowość

3. Słownik pojęć.

Kluczowe terminy i pojęcia budujące i opisujące regresję logistyczną i klasyfikację to:

- **zmienna** – dowolna cecha, liczba, ilość, która może być zmierzona lub określona, na przykład wiek, płeć, cena.
- **współczynnik –** dowolna liczba będąca mnożnikiem danej zmiennej niezależnej (wejścia), której towarzyszy. Określa jej wpływ i znaczenie dla zmiennej zależnej (wyjścia).
- ***outlier*** (z ang. wartość odstająca) – wartości w zbiorze danych wejściowych, które znacząco odbiegają od pozostałych.

4. Regresja logistyczna.

4.1. Pojęcie i zastosowania regresji logistycznej.

Jeżeli nie podano inaczej, informacje w tym podpunkcie pochodzą z [1, 2, 3]

Regresja logistyczna została pierwotnie opracowana na potrzeby biologii przez statystyków w celu opisu wzrostu populacji różnych gatunków na przestrzeni czasu – gwałtowny wzrost kończący się pewnym stopniem nasycenia. Pojęcie to zawiera w sobie całość procesu modelowania prawdopodobieństwa dla wyjścia dyskretnego, dychotomicznego, tj. dwustanowego, przykładowo prawda-fałsz, tak-nie itd. Jest zatem jednym z modeli klasyfikacji, inaczej niż wskazywałaby na to nazwa. Jest też zarazem jednym z najbardziej popularnych rozwiązań klasyfikacji binarnej. Pod kątem analizy rezultatów jest bardzo podobna do innych metod regresji. Regresja logistyczna charakteryzuje się jednak bardziej czasochłonnymi obliczeniami.

Jest to więc metoda statystyczna, która pozwala ocenić jak różne zmienne niezależne $x_1, x_2, \dots, x_k$ wpływają na dwustanową zmienną zależną Y, na przykład wpływ czasu poświęconego na naukę i oglądanie seriali (dwie zmienne niezależne) na zdanie egzaminu (zmienna zależna dwustanowa). Liczba praktycznych zastosowań regresji logistycznej jest zatem ogromna – sprawdzi się ona wszędzie tam, gdzie problem określony jest jako zależność binarnej predykcji od wielu zmiennych wejściowych. Pole zastosowania to przykładowo wspomniane nauki biologiczne i przyrodnicze czy też aplikacje związane z naukami społecznymi i analizą statystyk. Regresja wykorzystana w algorytmach uczenia maszynowego pozwala przewidywać prawdopodobieństwo wystąpienia zdarzeń na podstawie zgromadzonych danych historycznych - na przykład modele prognozy pogody analizujące dane meteorologiczne lub modele detekcji spamu na podstawie słów występujących w treści wiadomości.

O ile więc zmienna wyjściowa może przyjąć jedynie dwie wartości, wymagania wobec zmiennych wejściowych niezależnych w problemie regresji logistycznej są mniej rygorystyczne. Zmienna niezależna może należeć do jednej z trzech kategorii:

- **dane ciągłe**, mierzone na nieskończonej skali, mogące przyjąć dowolne wartości liczbowe, np. temperatura lub waga,
- **dane dyskretne kategoryczne**, przyjmujące jedną z określonych klas, na przykład ograniczony zbiór kolorów włosów,
- **dane dyskretne uporządkowane**, zawierające się w określonym, uporządkowanym przedziale, na przykład stopień zadowolenia z obsługi klienta w 10-stopniowej skali.

4.2. Model matematyczny.

Jeżeli nie podano inaczej, informacje w tym podpunkcie pochodzą z [3, 4, 5]

W typowych matematycznych analizach mamy do czynienia z ciągłą zmienną odpowiedzi Y i ciągłą zmienną objaśniającą X. W takich sytuacjach na ogół zakładamy relację:

$$E(Y|X) = \beta_0 + \beta_1 X$$

Równanie to reprezentuje regresję liniową, w której $E(Y|X)$ jest zmienną losową. Jeśli założymy, że zmienna Y może przyjąć jedynie dwie wartości – 0 lub 1 – wówczas $E(Y|X)$ będzie nazywane prawdopodobieństwem. Oznacza to zatem, że:

$$0 < \beta_0 + \beta_1 X < 1$$

Co jest jednak założeniem nie zawsze prawdziwym. Jeżeli jednak rozważymy $\ln\big(E(Y|X)\big)$, otrzymamy:

$$-\infty < \beta_0 + \beta_1 X < 0$$

W dalszym ciągu jest to przestrzeń ograniczona, lecz o wiele większa niż pierwotnie.

Rozważmy zatem iloraz szans Ω (z ang. *odds ratio*), będący stosunkiem prawdopodobieństwa wystąpienia danego zdarzenia w jednej grupie do prawdopodobieństwa jego wystąpienia w innej grupie. Przyjmuje on następującą postać:

$$\Omega = \frac{\Pi}{1 - \Pi}$$

gdzie Π oznacza prawdopodobieństwo sukcesu, w tym przypadku $\Pi = E(Y|X)$, a iloraz szans Ω mieści się w przedziale $(0, \infty)$.

Możemy zatem stworzyć ostateczny model regresji logistycznej, który przyjmie następującą postać:

$$\log(\Omega) = \log\left(\frac{E(Y|X)}{1 - E(Y|X)}\right) = \beta_0 + \beta_1 X$$

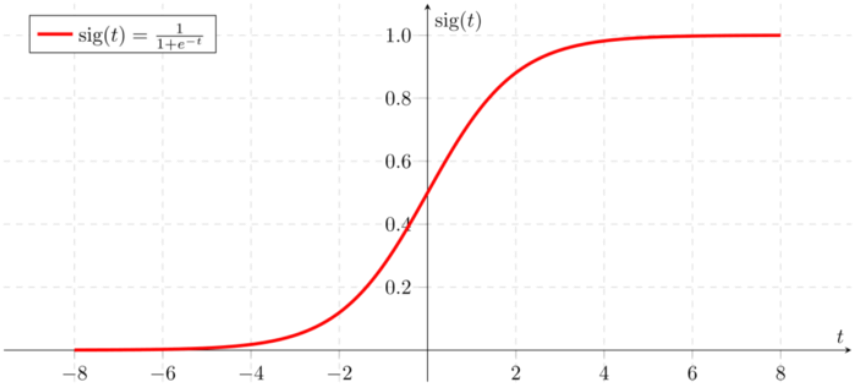
W tak zdefiniowanej przestrzeni otrzymujemy właściwy, nieograniczony zakres:

$$-\infty < \beta_0 + \beta_1 X < \infty$$

Model regresji w praktyce tworzony jest w oparciu o sigmoidę, funkcję logistyczną następującej postaci:

$$f(t) = \frac{e^t}{1 + e^t} = \frac{1}{1 + e^{-t}}$$

Można wywnioskować, że taka funkcja przyjmuje wartości z przedziału od 0 do 1 na całym przedziale, a konkretniej zmierza do zera dla kierunku ujemnego i do jedności dla dodatniej nieskończoności. Ograniczenie wartości do konkretnego zakresu, tu w szczególności do przedziału $\langle 0, 1 \rangle$, jest bardzo korzystne w kontekście statystyki, gdyż mamy pewność, że nigdy nie przekroczymy maksymalnego prawdopodobieństwa. To właśnie czyni regresję logistyczną tak popularną.



Rysunek 1. Wykres funkcji sigmoidalnej (źródło: [3])

Rezultatem hipotezy jest oszacowane prawdopodobieństwo, które pozwala określić pewność co do tego, czy dana wartość przewidywana jest zgodna z rzeczywistą dla konkretnych danych wejściowych.

W modelu regresji logistycznej zmienna t zawiera w sobie liniową kombinację zmiennych niezależnych, omówionych powyżej, będących w praktyce parametrami danych wejściowych.

$$t = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Można zatem zapisać:

$$f(t) = \frac{1}{1 + e^{-t}} = \frac{1}{1 + e^{-(\alpha + \sum_{i=1}^{k} \beta_i x_i)}}$$

gdzie:

α – stałe odchylenie, współczynnik przesunięcia prostej (ang. *bias*)

β_i – współczynnik udziału i-tej zmiennej niezależnej (x_i)

k – liczba zmiennych niezależnych

Przykładowo zatem dla dwóch zmiennych niezależnych ($k = 2$) funkcja przyjmie postać:

$$f(t) = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + \beta_2 x_2)}}$$

Tworząc model regresji logistycznej i przygotowując zbiór danych należy spełnić odpowiednie założenia:

- Zmienne $x_1, x_2, \ldots, x_k$ są liniowo niezależne i słabo lub wcale współliniowe. Zbyt duża korelacja zmiennych między sobą powoduje pogorszenie parametrów modelu.
- Zmienne niezależne są powiązane liniowo z różnicami logarytmicznymi $\left(\ln \frac{P}{1-P}\right)$
- Zbiór danych jest odpowiednio duży. Maksymalne prawdopodobieństwo jest słabsze niż metoda najmniejszych kwadratów, zatem rezultaty będą tym lepsze, im więcej próbek każdej ze zmiennych niezależnych będzie wprowadzonych.
- Zbiór danych nie posiada próbek odstających

- Wszystkie zmienne niezależne mają widoczny wpływ na predykcję. Należy odrzucić zmienne, których udział w zmiennej zależnej jest niewielki lub żaden.
- Zmienna zależna jest dychotomiczna.

5. Klasyfikacja wieloklasowa i regresja softmax.

Jeżeli nie podano inaczej, informacje w tym artykule pochodzą z [6, 7, 8].

W wielu problemach rzeczywistych mamy do czynienia z więcej niż dwoma możliwymi klasami. Przykładem może być grupa krwi (A, B, AB, 0) lub wyniki procesu rekrutacji na studia (w niektórych przypadkach może być ograniczona do klas „przyjęty" i „nieprzyjęty", lecz możliwa jest też trzecia opcja – „wpisany na listę rezerwową"). Chcielibyśmy więc poszerzyć model regresji logistycznej dla zastosowania w klasyfikacji wieloklasowej. Podejście takie jest faktycznie stosowane. Jednym z popularnych algorytmów dla danych kategorycznych, będącym uogólnieniem funkcji sigmoidy, jest regresja softmax.

Znormalizowana funkcja wykładnicza (powszechnie zwana softmax) zdefiniowana jest następująco:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

gdzie:

$\vec{z} = [z_1, z_2, \dots, z_K] \in \mathbb{R}^K$ – wektor danych wejściowych

$z_i \in \mathbb{R}$ – i-ty element wektora $\vec{z}$

$K > 1$ – wymiar przestrzeni wyjścia (liczba klas)

Najogólniej ujmując, rolą funkcji softmax jest wykorzystanie zdefiniowanego powyżej do znormalizowania wektora $\vec{z}$ (każdego jego elementu). W ten sposób funkcja ta zapewnia, że suma wszystkich elementów wektora $\vec{z}$ po normalizacji będzie wynosiła 1. Zamiast liczby Eulera, podstawą może być jakakolwiek liczba rzeczywista dodatnia.

Z powyższego powinno wynikać, że przy zastosowaniu liczby Eulera jako podstawy funkcja softmax powinna móc być sprowadzona do postaci analizowanej w poprzednim punkcie funkcji.

Niech dany będzie zbiór dwóch klas $[t, 0]$. Wówczas możemy obliczyć wartość znormalizowanej funkcji wykładniczej dla pierwszego elementu wyjściowego:

$$\sigma(\vec{z})_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} = \frac{e^{t}}{e^{t} + e^{0}} = \frac{e^{t}}{e^{t} + 1} = \frac{1}{1 + e^{-t}}$$

Otrzymaliśmy w ten sposób powszechnie znaną postać funkcji sigmoidy, przyjmującej jako argument wartość skalarną (w tym wypadku zmienną t). Dla formalności warto jeszcze sprawdzić warunek sumowania wszystkich elementów wektora. Dla drugiej klasy otrzymamy zatem:

$$\sigma(\vec{z})_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2}} = \frac{e^{0}}{e^{t} + e^{0}} = \frac{1}{e^{t} + 1}$$

Sumując obie wartości otrzymamy:

$$\sigma(\vec{z})_1 + \sigma(\vec{z})_2 = \frac{e^t}{e^t + 1} + \frac{1}{e^t + 1} = \frac{e^t + 1}{e^t + 1} = 1$$

Spełniony jest więc warunek sumy elementów, a zarazem sumy prawdopodobieństw, do jedności. Zatem funkcja sigmoidy jest szczególnym przypadkiem funkcji softmax dla $K = 2$.

Znormalizowana funkcja wykładnicza jest stosowana powszechnie w wielu dziedzinach, na przykład w uczeniu maszynowym i sieciach neuronowych, gdzie stanowi jedną z funkcji aktywacji neuronów. Odnosząc się do zastosowań opisanych w literaturze, warto nadmienić model wykrywania ryzyka raka przełyku oparty o dyskryminator bazujący na funkcji softmax [9], model globalny regresji softmax do rozpoznawania emocji na podstawie mowy [10] czy model nowej funkcji kosztu w celu lepszego wykorzystania optymalizacji gradientowej [11].

6. Szeregi czasowe.

Jeżeli nie podano inaczej, informacje w tym rozdziale pochodzą z [12, 13].

W matematyce pojęcie szeregów czasowych oznacza pewien zbiór punktów danych indeksowanych w porządku czasowym, chronologicznie. Najczęściej przyjmuje postać sekwencji danych złożonej z próbek równoodległych w czasie. Zapisać można go więc jako klasyczny zbiór:

$$X = \{X_1, X_2, \dots\} = \{X_t : t \in T\}$$

gdzie T oznacza zbiór indeksujący. Indeksami najczęściej są liczby naturalne lub chwile czasowe.

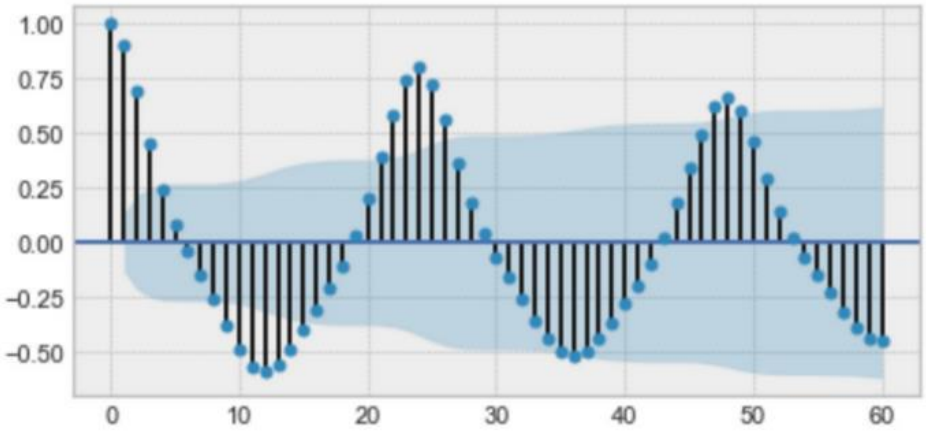
Przykładem może być wartość indeksów spółek giełdowych WIG20 odczytywana co poniedziałek w chwili zamknięcia sesji albo odczyt temperatury na danej stacji pomiarowej co godzinę. Szeregi czasowe stosowane są w wielu dziedzinach nauki, przykładowo w statystyce, przetwarzaniu sygnałów, ekonomii, astronomii czy prognozach pogody.

Szeregi czasowe są wykorzystywane podczas ich **analizy** w celu ekstrakcji danych znaczących statystycznie, wyszukiwania wzorców oraz danych charakterystycznych czy też zjawisk rzadkich. Na podstawie szeregu czasowego można określić na przykład jego trend, wahania sezonowe czy cykliczne. W przypadku tej pracy ma miejsce właśnie analiza szeregów czasowych. Osobną dziedziną jest **predykcja zjawisk**, w której bazując na posiadanych obserwacjach i ich analizie odpowiednie modele próbują przewidywać przyszłe przebiegi i zdarzenia. Przykładem takiego zastosowania jest próba przewidzenia wyników wyborów powszechnych w Szwecji w 2018 roku [13].

6.1. Wybrane parametry i pojęcia analizy szeregów czasowych.

- **Autokorelacja**

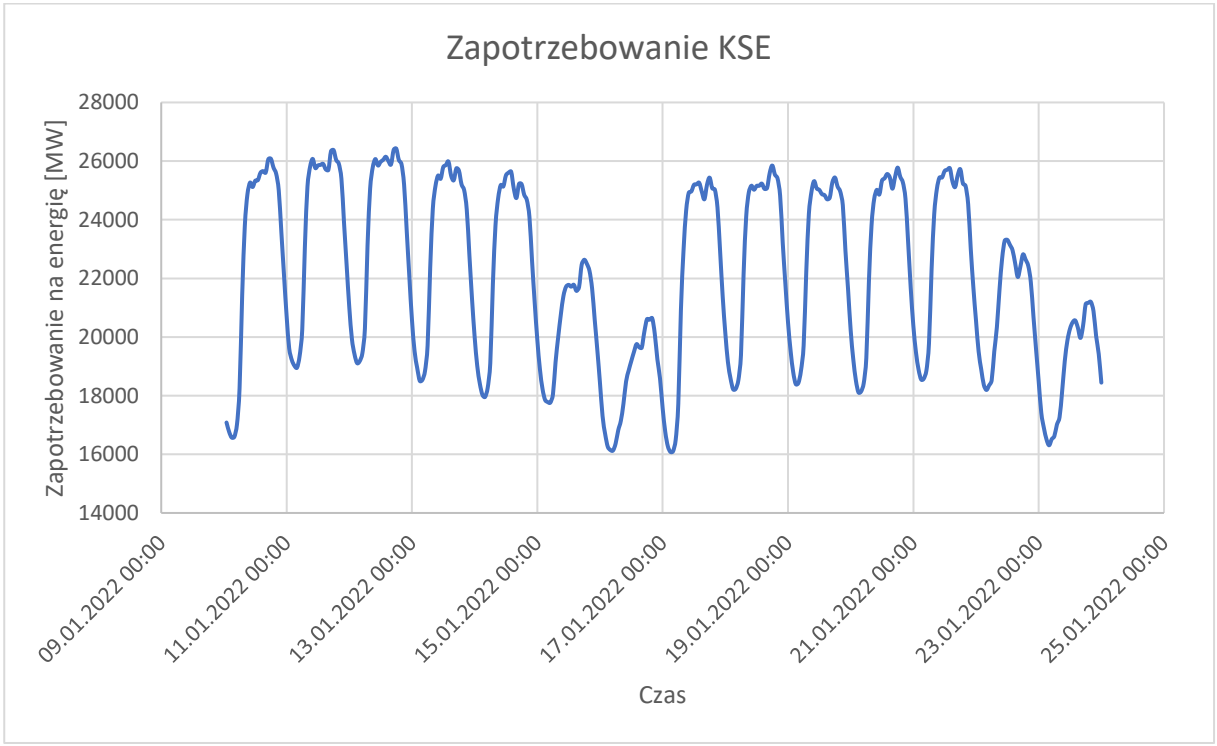
Nieformalnie ujmując, autokorelacja jest miarą podobieństwa między obserwacjami w ramach tego samego szeregu, ale przesuniętych w czasie. Przykładowo na wykresie poniżej można zauważyć, że próbka numer 1 i numer 24 są do siebie podobne, tak samo jak próbka 12 i 36 i tak dalej. Można na tej podstawie zauważyć, że to podobieństwo wystąpi dla każdej pary oddalonej od siebie o 24 próbki.



Rysunek 2. Przykład wykresu danych o wysokim współczynniku autokorelacji

- **Wahania sezonowe**

Pojęcie wahań sezonowych odnosi się do okresowych, powtarzających się wzorców zmian wartości. Przykładowo cykl zużycia energii jest powtarzalny – można zauważyć niższe zużycia nocą i wyższe w ciągu dnia.



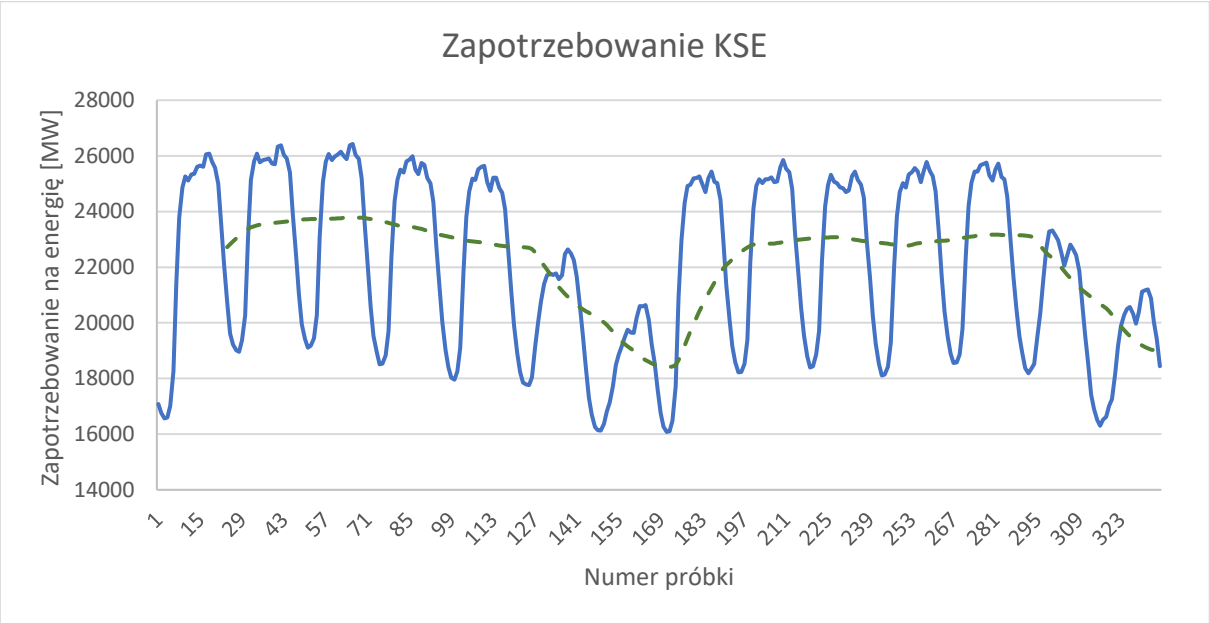
Rysunek 3. Wykres zapotrzebowania na energię elektryczną w Polsce w dniach 10-24 stycznia 2022

- **Stacjonarność**

Szereg czasowy jest stacjonarny wówczas, gdy jego własności nie zmieniają się w czasie. Oznacza to, że ma stałą średnią i wariancję. Przykładem stacjonarnego szeregu czasowego może być podany wyżej wykres zapotrzebowania na energię elektryczną (z pewnymi wahaniami dla weekendów). Szereg niestacjonarny to na przykład wykres ceny akcji na giełdzie.

- **Średnia krocząca**

Model średniej kroczącej wykorzystywany może być do predykcji kolejnych wartości szeregu. Pozwala na pewne przekształcenie wykresu szeregu czasowego zgodnie z zasadą, że każda kolejna próbka jest średnią ze wszystkich (lub pewnej liczby) przeszłych obserwacji. Pomimo swojej prostoty, często model ten sprawdza się zaskakująco dobrze w określeniu pewnego przybliżonego punktu startowego.



Rysunek 4. Wykres zapotrzebowania na energię elektryczną w Polsce w dniach 10-24 stycznia 2022 wraz ze średnią kroczącą

W przykładzie wykorzystano ponownie zapotrzebowanie na energię elektryczną. Zaznaczono średnią kroczącą z oknem 24 godzin. Stosowanie okna, czyli liczby przeszłych próbek branych pod uwagę, pozwala w pewien sposób wygładzić szereg czasowy bez utraty informacji o trendzie. Tak więc pomimo straty szczegółowych informacji, wciąż łatwo można zauważyć minima lokalne w okresie weekendów.

- **Wygładzanie wykładnicze**

Zasada działania wygładzania wykładniczego jest analogiczna do średniej kroczącej, z tym że dodatkowo uwzględnia się wagi przeszłych próbek, malejące wykładniczo – to znaczy, że im starsza (bardziej odległa w czasie) jest próbka, tym jej udział i znaczenie są mniejsze. W efekcie otrzymuje się ponownie wykres wygładzony, pozbawiony szumu. Podejście to jest wykorzystywane również do predykcji przyszłych wartości, zwłaszcza w przypadku szeregów czasowych bez wyraźnego trendu i wahań sezonowych.

Matematycznie definicja brzmi następująco:

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1-\alpha)s_{t-1}$$

gdzie:

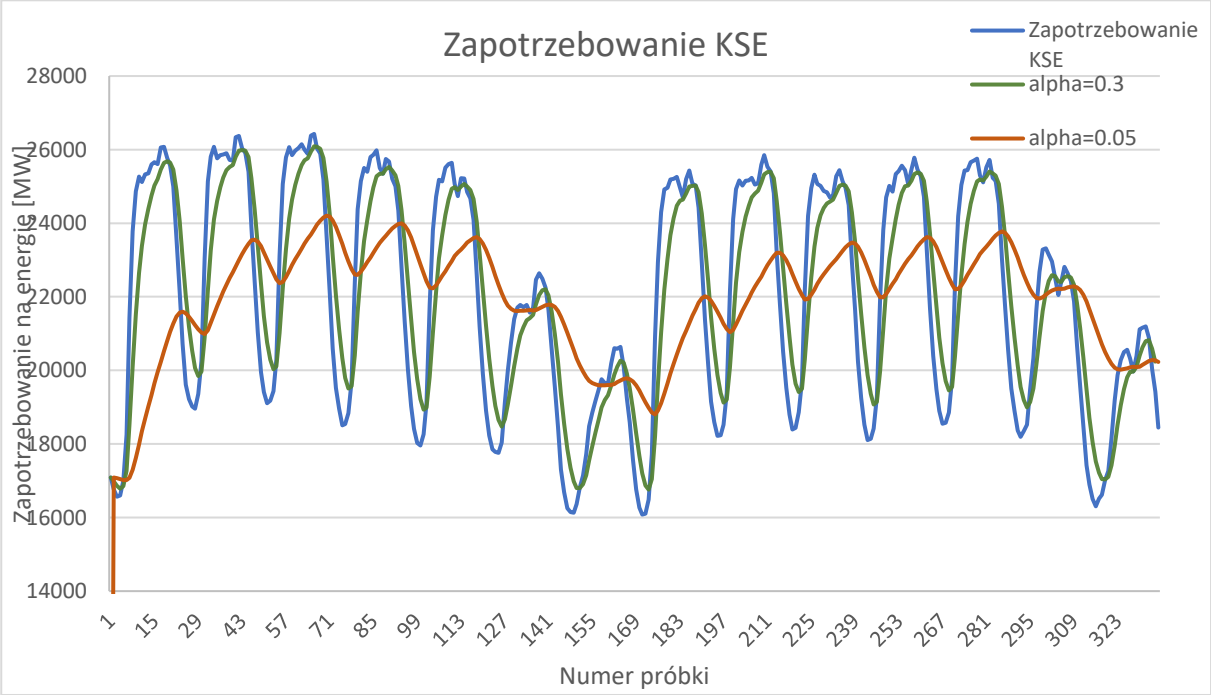
x_0 – początkowy wyraz szeregu pierwotnego

s_0 – początkowy wyraz nowego szeregu, wygładzonego wykładniczo

x_t, s_t – wartości wyrazu o indeksie t odpowiednio szeregu pierwotnego i nowego

$t > 0$ – indeks ze zbioru T poza zerowym

$\alpha \in (0, 1)$ – współczynnik wygładzania, definiujący jak szybko maleją wagi przeszłych próbek



Rysunek 5. Wykres zapotrzebowania na energię elektryczną w Polsce w dniach 10-24 stycznia 2022 wraz z przykładowymi przebiegami wygładzonymi wykładniczo dla różnych wartości α.

Im mniejsza wartość parametru α tym bardziej wygładzony przebieg. Jest to zgodne z intuicją, bowiem wraz ze zbliżaniem się parametru do zera, wzór zbiega do średniej kroczącej.

Stosowane są również metody podwójnego i potrójnego wygładzania wykładniczego, głównie w sytuacji, gdy występuje odpowiednio: trend i wahania sezonowe. Pojawiają się wówczas dodatkowe współczynniki β, γ o analogicznym znaczeniu. Przykładowo, w uproszczeniu, podwójne wygładzanie polega na wykorzystaniu wspomnianego wyżej wzoru na rezultacie jego pierwszego użycia. Matematycznie możemy zapisać:

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1-\alpha)(s_{t-1} + b_{t-1})$$
$$b_t = \beta(y_t - y_{t-1}) + (1-\beta)b_{t-1}$$

gdzie:

b – nowopowstały szereg

β – współczynnik wygładzania trendu

Pozostałe oznaczenia jak poprzednio.

7. Funkcjonalna analiza danych.

Jeżeli nie podano inaczej, informacje w tym rozdziale pochodzą z [15, 16, 17].

Funkcjonalna analiza danych jest relatywnie nową dziedziną w statystyce. Pierwsze złożone i kompletne opracowanie tego zagadnienia, autorstwa J.O. Ramsaya i B.W. Silvermana, pochodzi z 1997 roku [17].

Dane funkcjonalne mogą być zbierane na różne sposoby. Jednym z podejść jest wykorzystanie interpolacji do utworzenia funkcji na podstawie zebranych danych (w oparciu na przykład o szereg czasowy, wówczas będzie to funkcja zmienna w czasie). Metoda ta wynika ze skończonej, ograniczonej możliwości zbierania próbek danych przez fizyczne urządzenia. Dane funkcjonalne są wszelakiego pochodzenia – mogą to być nawet obrazy wykorzystane jako parametry modeli wykorzystywane w archeologii, co zostało opisane w [16].

W opisywanej pracy skupiono się na interpretacji danych funkcjonalnych jako funkcji ciągłych utworzonych na podstawie zbioru dyskretnych próbek uporządkowanych w czasie. Takie podejście zakłada, że pomimo skończonej rozdzielczości urządzeń pomiarowych sam mierzony proces jest ciągły, gładki, płynny. Dzięki temu, zamiast interpretować wiele posiadanych próbek jako osobne punkty, traktuje się je całościowo jako jedną obserwację.

7.1. Bazowy opis funkcjonalnej analizy danych

Rozważmy zbiór krzywych opartych o zbiór będący szeregiem czasowym:

$$\{\boldsymbol{x_i}(t), t \in T, i = 1, \dots, n\}$$

gdzie:

T – przedział czasowy z którego pochodzą dane pomiarowe

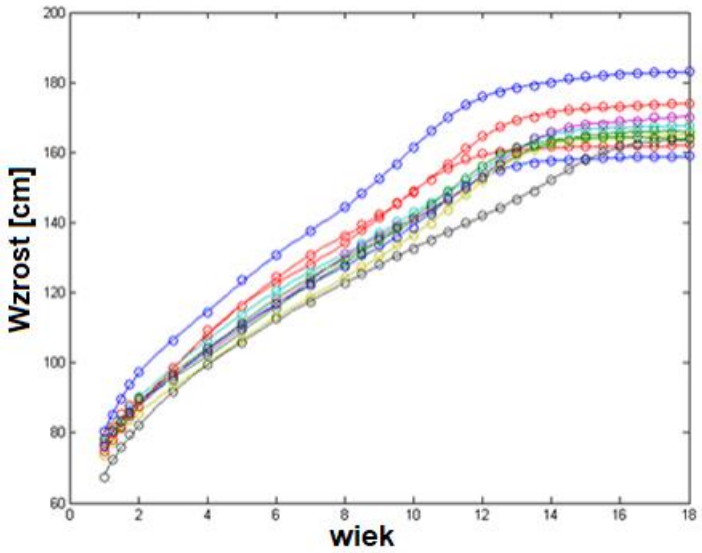
Obserwacje należą do przestrzeni Hilberta $L_2(T)$ z iloczynem skalarnym zdefiniowanym następująco:

$$\langle f, g \rangle = \int_T fg dt, \qquad \forall f, g \in L_2(T)$$

Wówczas, wektor $\boldsymbol{x_i} = [x_{i1}, \dots, x_{iN}]$ reprezentuje zbiór pomiarów dla i-tego szeregu czasowego złożonego z N punktów zebranych w przedziale czasowym T.

W podejściu funkcjonalnym wykorzystywanych jest wiele analogii do „klasycznego" odpowiednika analizy danych. Na przykład średnia dla danych funkcjonalnych zdefiniowana jest jako funkcja oparta na średnich wartościach n rozpatrywanych krzywych, liczona dla każdej chwili t:

$$\bar{x}(t) = n^{-1} \sum_{i=1}^{n} x_i(t)$$



Rysunek 6. Przykładowe krzywe danych funkcjonalnych - wzrost dziewczynek w zależności od ich wieku, oparte o dane z uniwersytetu Berkeley[1].

Dzięki założeniu ciągłości i gładkości funkcji możliwe jest również rozpatrywanie pochodnych. Dla przykładu na wykresie powyżej pozwala to na określenie szybkości wzrostu w zależności od wieku.

8. Wykorzystane technologie.

8.1. Python

Python to język programowania wysokiego poziomu o bardzo szerokim zastosowaniu. Jego ogromnymi atutami są przede wszystkim otwartoźródłowość, przejrzystość kodu, rozbudowane pakiety dodatkowych bibliotek i frameworków oraz rozwinięta i liczna społeczność. Język ten jest obecnie jednym z najbardziej popularnych. Jest stosowany w wielu dziedzinach, począwszy od podstaw nauki programowania, na rozwiązaniach sztucznej inteligencji kończąc. Ze względu na mnogość bibliotek statystycznych oraz dotyczących uczenia maszynowego, wieloplatformowość, a także wspomnianych wyżej cech, zdecydowano się na wybór Pythona jako głównego języka projektu.

Aby móc efektywnie tworzyć kod, zdecydowano się na wykorzystanie zintegrowanego środowiska programistycznego *PyCharm Professional*[2].

8.2. JavaScript

Jeżeli nie podano inaczej, informacje w tym podpunkcie pochodzą z [14].

O ile wybór Pythona dla rozwiązań obliczeniowych i backendowych jest dobrze uargumentowany, o tyle w celach tworzenia warstwy wizualnej aplikacji już nie. Z tego względu zdecydowano się wykorzystać JavaScript jako drugi język.

[1] https://www.jstor.org/stable/1125347
[2] https://www.jetbrains.com/pycharm/ Dostęp: 10.02.2022r.

JavaScript to skryptowy język programowania, w którym góruje podejście programowania funkcyjnego. Funkcje traktowane są jako obiekty, które można przechowywać w zmiennych jako referencje i przekazywać jak każde inne obiekty. Jest to podstawowy język skryptowy dla tworzenia stron internetowych. Stosowany jest jednak nie tylko w przeglądarkach, ale również w innych środowiskach, jak Adobe Acrobat czy wykorzystywany w projekcie Node.js.

Node.js to wieloplatformowe środowisko uruchomieniowe dla projektów JavaScript pozwalające deweloperom na budowanie aplikacji sieciowych i webowych, zarządzanie bibliotekami, automatyzację testów i wykonywanie innych okołoprojektowych zadań.

8.3. ReactJS[3]

React to biblioteka dla języka JavaScript służąca do tworzenia interfejsów użytkownika. Jest to framework deklaratywny, upraszczający tworzenie rozwiązań interaktywnych i responsywnych. Rola dewelopera została ograniczona do zaprojektowania kolejnych stron i widoków oraz opisaniu bazowej logiki, jaka ma stać za wyglądem. Całą resztą, czyli m.in. aktualizacją parametrów oraz odświeżaniem i ponownym renderowaniem komponentów zajmuje się już biblioteka w sposób automatyczny, lecz w pełni kontrolowalny. Idea oparcia na wspomnianych komponentach pozwala na izolowanie poszczególnych funkcjonalności i zamykanie ich w niezależnych fragmentach, które same zarządzają własnym stanem i mogą być łączone w zaprojektowany interfejs. Ważnym kierunkiem rozwoju biblioteki jest jej wersja natywna *React Native* pozwalająca na tworzenie natywnych aplikacji mobilnych działających niezależnie od platformy docelowej.

8.4. System kontroli wersji

Systemy kontroli wersji (z ang. *version control systems*) powstały w celu ułatwienia zarządzania projektami i śledzenia historycznych zmian. Każda modyfikacja, na przykład kodu źródłowego, jest zapisywana w historii i możliwa do odtworzenia lub wycofania w dowolnym momencie. Zapisywane są także informacje takie jak data, autor i opis zmian.

W trakcie pracy nad projektem związanym z pracą dyplomową wykorzystano system kontroli wersji **Git**. Jest to rozwiązanie darmowe, Open Source, oparte o architekturę rozproszoną, będące obecnie najbardziej popularnym zastosowaniem. Dzięki idei równoległych gałęzi (*branches*) pozwala na łatwą pracę nad wieloma zmianami równocześnie, co przekłada się na efektywne tworzenie projektu przez wiele osób. Przy pracy jednoosobowej cecha ta jest mniej istotna, lecz również wartościowa w kontekście elastyczności, separacji poszczególnych funkcji i fragmentów projektu czy właściwego wersjonowania kolejnych kroków milowych. System Git stanowi niezależne oprogramowanie, które może być stosowane kompatybilnie zarówno lokalnie jak i z wykorzystaniem zdalnych serwerów hostujących.

[3] https://pl.reactjs.org/ Dostęp: 10.02.2022r.

System Git jest wykorzystywany i dostarczany przez wiele serwisów. W pracy wykorzystano rozwiązanie **GitHub**, będące jednym z głównych dostawców narzędzia. Obecnie z usług serwisu korzysta ponad 4 miliony organizacji i firm[4]. Rozbudowana społeczność milionów użytkowników oraz mnogość poradników sprawia, że narzędzie jest łatwe do zapoznania się i wdrożenia do własnego projektu. W tym przypadku był on wykorzystywany od samego początku prac. Serwis umożliwia również automatyzację niektórych procesów, łatwe połączenie z systemami ciągłej integracji oraz z rozwiązaniami zewnętrznymi, jak wykorzystana w projekcie platforma Heroku.

8.5. Continuous Integration i Travis CI

Pojęcia *Continous Integration* (*CI*) oraz *Continouos Testing* stanowią obecnie nieodłączny element wielu projektów. Są to frazy, które skrywają całą metodologię automatyzacji procesów wersjonowania, wydawania, dokumentacji i wykonywania ciągłych testów (jednostkowych, integracyjnych, *end to end*…). Praktyki te mają na celu minimalizację błędów człowieka oraz standaryzację rozwoju oprogramowania, tak aby możliwe było wykonywanie powtarzalnych czynności tam, gdzie to konieczne. Takie podejście umożliwia łatwe śledzenie wszystkich zmian w kodzie, unikanie zmian przypadkowych i wczesne wykrywanie błędów. Założenie to jest obecnie jednym z podstaw efektywnego tworzenia oprogramowania [15].

Na rynku istnieje wiele rozwiązań CI. W trakcie pracy nad projektem wykorzystano narzędzie **TravisCI**[5], które jest jednym z popularniejszych serwisów oraz pozwala na darmowe wykorzystanie w indywidualnych, niekomercyjnych projektach. W ramach środowiska użytkownik otrzymuje dostęp do maszyny wirtualnej, którą można skonfigurować w pożądany sposób poprzez współdzielenie repozytorium projektu z serwisem oraz dodanie do niego pliku ustawień .travis.yml. W ramach tego pliku deweloper może skonfigurować parametry środowiska (na przykład wersję systemu), język programowania, etapy instalacji kolejnych wymaganych bibliotek, skrypt uruchamiający projekt oraz automatyzację testów i innych pożądanych zadań. Panel użytkownika wraz z konsolą dostarczają szczegółowe informacje o procesie budowania oraz ewentualnych błędach bądź ostrzeżeniach. Po każdym jednorazowym uruchomieniu tworzony jest raport z testów. Dzięki możliwości integracji z platformą GitHub skonfigurowane działania wykonywane są automatycznie po każdej aktualizacji wybranych gałęzi, co w konsekwencji pozwala na wykrycie błędów, w tym tych związanych z wieloplatformowością, na długo przed wydaniem oprogramowania. Zarazem, dzięki utworzeniu gotowej, sprawnej konfiguracji, użytkownik gwarantuje sobie działający proces wydawania wersji produkcyjnej w środowisku innym niż lokalne.

[4] Źródło: https://github.com/ Dostęp: 10.02.2022r.
[5] https://travis-ci.org/ Dostęp: 10.02.2022r.

8.6. Heroku

Heroku to platforma chmurowa działająca w modelu *„Platform as a Service”* (*PaaS*). Głównym dostarczanym produktem jest wirtualne środowisko pracy, udostępniane zarówno firmom jak i pojedynczym użytkownikom. Aplikacja posiada wiele gotowych rozwiązań i instrukcji oraz bogate wsparcie dla automatyzacji procesów, co pozwala na sprawne zarządzanie projektem i jego uruchamianie. Platforma wspiera natywnie wszystkie główne języki programowania, w tym także język Python wykorzystany podczas prac nad projektem. Szerokie wsparcie jest dostępne także pod kątem aplikacji webowych, co wpasowuje się bardzo dobrze w potrzeby projektu. Jedną z głównych cech platformy jest bardzo łatwa skalowalność. Dostępne zasoby oraz ich ustawienia można modyfikować za pomocą kilku kliknięć. Wersja darmowa oferuje jednak wystarczającą ilość pamięci i przestrzeni dyskowej, by uruchomić wersję testową wykonanej aplikacji. To, jak i fakt ogromnej popularności i zaufania społeczności było główny powodem wykorzystania usługi Heroku. Indywidualni użytkownicy jak i firmy i korporacje utworzyły w serwisie już ponad 13 milionów aplikacji[6].

8.7. Narzędzie dokumentacyjne

9. Wybrane metody konwersji szeregów czasowych na bazy funkcyjne.

10. Realizacja pracy

10.1. Repozytorium projektu

10.2. Zasada działania programu

10.3. Format danych wejściowych

10.4. Implementacja algorytmów

10.5. Uzyskane rezultaty i wizualizacje

10.6. Aplikacja webowa

11. Testy implementacji

11.1. Testy jednostkowe

11.2. Walidacja i scenariusze testowe

11.3. Testy aplikacji webowej

[6] Źródło: https://www.heroku.com/what Dostęp: 10.02.2022r.

12. Podsumowanie

Bibliografia

Bibliografia

[1] T. Edgar i D. Manz, Research Methods for Cyber Security, Cambridge, 2017.

[2] A. Subasi, Practical Machine Learning for Data Analysis Using Python, London: Academic Press, 2020.

[3] S. Swaminathan, „Logistic Regression - Detailed Overview," [Online]. Available: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc. [Data uzyskania dostępu: 11 09 2021].

[4] D. G. Kleinbaum, Logistic Regression: A Self-Learning Text, Nowy Jork: Springer, 1994.

[5] „An Introduction to Logistic Regression for Categorical Data Analysis," [Online]. Available: https://towardsdatascience.com/an-introduction-to-logistic-regression-for-categorical-data-analysis-7cabc551546c. [Data uzyskania dostępu: 15 02 2022].

[6] S. Yang, „Multiclass logistic regression from scratch," [Online]. Available: https://towardsdatascience.com/multiclass-logistic-regression-from-scratch-9cc0007da372. [Data uzyskania dostępu: 15 02 2022].

[7] X. Qiao, Z. Yang, Y. Wang, W. Nai, D. Li i Y. Xing, „Softmax Regression Based on Bacterial Foraging Optimization Algorithm with t-Distribution Parameters," IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC), Pekin, 2021.

[8] T. Wood, „Softmax Function," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer. [Data uzyskania dostępu: 17 02 2022].

[9] H. Rajaguru i S. K. Prabhakar, „An approach to classification of oral cancer using Softmax Discriminant Classifier," w *2nd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, 2017.

[10] Y. Sun i G. Wen, „Ensemble softmax regression model for speech emotion recognition," *Multimedia Tools and Applications,* pp. 8305-8328, Marzec 2017.

[11] S. Bruch, X. Wang, M. Bendersky i M. Najork, „An Analysis of the Softmax Cross Entropy Loss for Learning-to-Rank with Binary Relevance," *ICTIR '19: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval,* pp. 75-78, Wrzesień 2019.

[12] M. Peixeiro, „The Complete Guide to Time Series Analysis and Forecasting," [Online]. Available: https://towardsdatascience.com/the-complete-guide-to-time-

series-analysis-and-forecasting-70d476bfe775. [Data uzyskania dostępu: 23 02 2022].

[13] A. K. Rosenblad, „Accuracy of automatic forecasting methods for univariate time series data: A case study predicting the results of the 2018 Swedish general election using decades-long data series.," *Communications in Statistics: Case Studies & Data Analysis*, tom VII, pp. 475-493, 2021.

[14] [Online]. Available: https://developer.mozilla.org/pl/docs/Web/JavaScript. [Data uzyskania dostępu: 12 02 2022].

[15] W. Felidre, "Continuous Integration Theater," Porto de Galinhas, 2019.