

**WOJSKOWA AKADEMIA TECHNICZNA**  
im. Jarosława Dąbrowskiego  
**WYDZIAŁ CYBERNETYKI**

---



**SPRAWOZDANIE**  
**Metody Eksploracji Danych**

Temat laboratorium: **FUNKCJA LOGISTYCZNA**

**INFORMATYKA**

(kierunek studiów)

**INŻYNIERIA SYSTEMÓW – ANALIZA DANYCH**

(specjalność)

Zespół:

**Michał ŚLĘZAK**  
**Szymon OLEŚKIEWICZ**

Prowadzący laboratorium:

**Dr inż. Romuald Hoffmann, prof.**  
**WAT**

---

**Warszawa 2025**



## Spis treści

Wstęp .....	4
Rozdział I. Podstawy teoretyczne.....	5
I.1. Funkcja logistyczna .....	5
I.2. Metoda doboru parametrów.....	6
I.3. Wykorzystane narzędzia .....	7
Rozdział II. Opis problemu .....	8
II.1. Treść zadania .....	8
Rozdział III. Implementacja modeli .....	9
III.1. Model logistyczny .....	9
III.2. Modele dla zagregowanych danych oraz podzielonych danych .....	11
Podsumowanie.....	18
Bibliografia .....	19
Spis tabel .....	20
Spis rysunków .....	20
Załączniki .....	21

## **Wstęp**

Celem niniejszego sprawozdania jest wykorzystanie funkcji logistycznej do wyznaczenia zależności sumarycznej liczby błędów w okresie eksploatacji oprogramowania badanego systemu operacyjnego.

## Rozdział I. Podstawy teoretyczne

### I.1. Funkcja logistyczna

Funkcja logistyczna to nieliniowa funkcja 1 zmiennej, zmienną tą jest zazwyczaj czas (t). Wykresem tej funkcji jest krzywa przypominająca kształtem literę „S” oraz należy do rodziny krzywych sigmoidalnych (kształtem przypominających właśnie literę „S”). Funkcja ta ma następujące właściwości – dla  $t > 0$  wartość funkcji logistycznej początkowo bardzo szybko rośnie a później tempo jej wzrostu maleje i końcowo stabilizuje się w pobliżu pewnej asymptoty. Klasyczna postać funkcji logistycznej jest pokazana poniżej.

$$y(t) = \frac{\alpha}{1 + \beta \cdot e^{-\gamma \cdot t}} \quad (1)$$

gdzie  $\alpha > 0, \beta > 0, \gamma > 0$ .

Parametry te oznaczają odpowiednio:

- $\alpha$  – poziom nasycenia odpowiadający asymptotej górnej. Jest to graniczna wartość, do której dąży funkcja przy  $t$  dążącym do nieskończoności. W kontekście zadania laboratoryjnego oznacza przewidywaną, całkowitą liczbę błędów jakie zawiera system.
- $\beta$  – parametr przesunięcia odpowiadający za położenie krzywej na osi OX, osi czasu. Wpływa na wartość początkową funkcji dla  $t = 0$ , czyli liczbę błędów wykrytych na początku eksploatacji systemu.
- $\gamma$  – współczynnik tempa wzrostu określający jak bardzo krzywa jest stroma. Im większy ten parametr, tym szybciej funkcja osiąga nasycenie.

W realizowanym zadaniu laboratoryjnym zastosowano sparametryzowaną postać funkcji logistycznej jak poniżej, z jawnym punktem przegięcia, który jest parametrem funkcji, w przeciwieństwie do poprzedniej postaci, gdzie punkt przegięcia jest obliczany z wartością innych parametrów – dla  $t = \ln \frac{\beta}{\gamma}$ .

$$y(t) = \frac{K}{1 + e^{-a(t-t_0)}} \quad (2)$$

gdzie:

- $K$  – odpowiada parametrowi alfa z klasycznego modelu. Oznacza poziom nasycenia, maksymalna wartość, którą funkcja może osiągnąć. W naszym przypadku jest to estymowana całkowita liczba błędów krytycznych w eksploatowanych systemie operacyjnym.
- $a$  – odpowiada parametrowi gamma z poprzedniego modelu. Określa współczynnik tempa wzrostu, który wpływa na to jak stroma jest krzywa.

Wysoka wartość tego parametru oznacza, że błędy są wykrywane bardzo często w krótkim okresie.

- $t_0$  – punkt przegięcia funkcji, jest to punkt, w którym funkcja osiąga połowę swojej maksymalnej wartości. Jest to punkt w czasie, w którym tempo przyrostu błędów było największe. Dla  $t < t_0$  liczba wykrywanych błędów krytycznych rośnie (wartość funkcji szybko rośnie), dla  $t = t_0$  funkcja osiąga połowę maksymalnej wartości funkcji i dla  $t > t_0$  liczba wykrywanych błędów maleje.

W celu zbadania jak dobrze wykres funkcji logistycznej dopasowuje się do sumarycznej liczby błędów można wykorzystać współczynnik determinacji, który obliczany jest na podstawie poniższego wzoru. Współczynnik ten przyjmuje wartości od 0 do 1. Im bliżej 1, tym nasz model jest bardziej dopasowany.

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (3)$$

## I.2. Metoda doboru parametrów

Ze względu na nieliniowy charakter rozpatrywanej funkcji logistycznej nie jest możliwe wyznaczenie jej parametrów za pomocą, np. Metody Najmniejszych Kwadratów, jak to miało miejsce w regresji liniowej. W celu estymacji parametrów modelu zastosowano Nieliniową Metodę Najmniejszych Kwadratów. Zadanie optymalizacyjne zdefiniowane jest jako minimalizacja sumy kwadratów reszt ( błędów) pomiędzy wartościami rzeczywistymi ( $y$  - skumulowana liczba błędów) a wartościami teoretycznymi modelu (to co zwraca nam funkcja po podstawieniu odpowiedniego  $t$ ), jak poniżej.

$$\min_{K,a,t_0} \sum_{i=1}^n \left( y_i - \frac{K}{1 + e^{-a(t_i - t_0)}} \right)^2 \quad (4)$$

Do obliczeń wykorzystano funkcję *curve\_fit* z biblioteki *SciPy* (*scipy.optimize*). Aby rozpocząć minimalizację, należy podać swego rodzaju punkt startowy (wektor parametrów początkowych  $p0$ ). W naszej implementacji, zastosowano następujące wartości początkowe, jak poniżej. Dla parametru  $K$  przyjęto wartość maksymalną z danych rzeczywistych, która jest de facto dolnym ograniczeniem asymptoty, tzn. system operacyjny musi mieć co najmniej tyle błędów, ile już ich wykryto. Dla parametru tempa wzrostu  $a$  przyjęto wartość 1, co jest standardową wartością domyślną. Dla parametru punktu przegięcia przyjęto medianę czasu trwania eksploatacji systemu, założono że punkt kulminacyjny sumarycznej liczby wykrytych błędów znajduje się w połowie badanego okresu, sugerując się wykresem skumulowanej liczby błędów.

### I.3. Wykorzystane narzędzia

W celu wyznaczenia zależności sumarycznej liczby błędów w okresie eksploatacji oprogramowania badanego systemu operacyjnego na podstawie wykładów oraz wiedzy własnej przygotowano program w języku *Python* wykorzystujący biblioteki:

- **Pandas,**
- **NumPy,**
- **stastmodel,**
- **scikit-learn,**
- **scipy,**

które również zostały wykorzystane do obliczeń oraz wizualizacji wyników.

## Rozdział II. Opis problemu

### II.1. Treść zadania

Przez cały okres eksploatacji pewnego systemu operacyjnego (OS) zbierano dane dotyczące liczby błędów krytycznych wykrytych w tym czasie w oprogramowaniu. Zebrane obserwacje w układzie miesięcznym przedstawiono w tabeli 3.

**Tab. 1. Liczba błędów krytycznych oprogramowania badanego OS w układzie miesięcznym**

Nr miesiąca	Liczba błędów								
1	1	21	9	41	3	61	2	81	8
2	0	22	0	42	2	62	19	82	0
3	0	23	0	43	1	63	7	83	6
4	0	24	3	44	6	64	2	84	5
5	0	25	1	45	3	65	5	85	10
6	0	26	1	46	0	66	1	86	0
7	0	27	0	47	1	67	12	87	5
8	0	28	1	48	1	68	4	88	1
9	0	29	0	49	0	69	6	89	2
10	0	30	2	50	0	70	4	90	1
11	0	31	10	51	14	71	7	91	2
12	0	32	0	52	1	72	2	92	1
13	0	33	16	53	4	73	2	93	0
14	0	34	0	54	1	74	3		
15	0	35	2	55	1	75	8		
16	0	36	2	56	7	76	4		
17	0	37	1	57	14	77	6		
18	0	38	1	58	6	78	3		
19	0	39	1	59	0	79	3		
20	1	40	0	60	1	80	5		

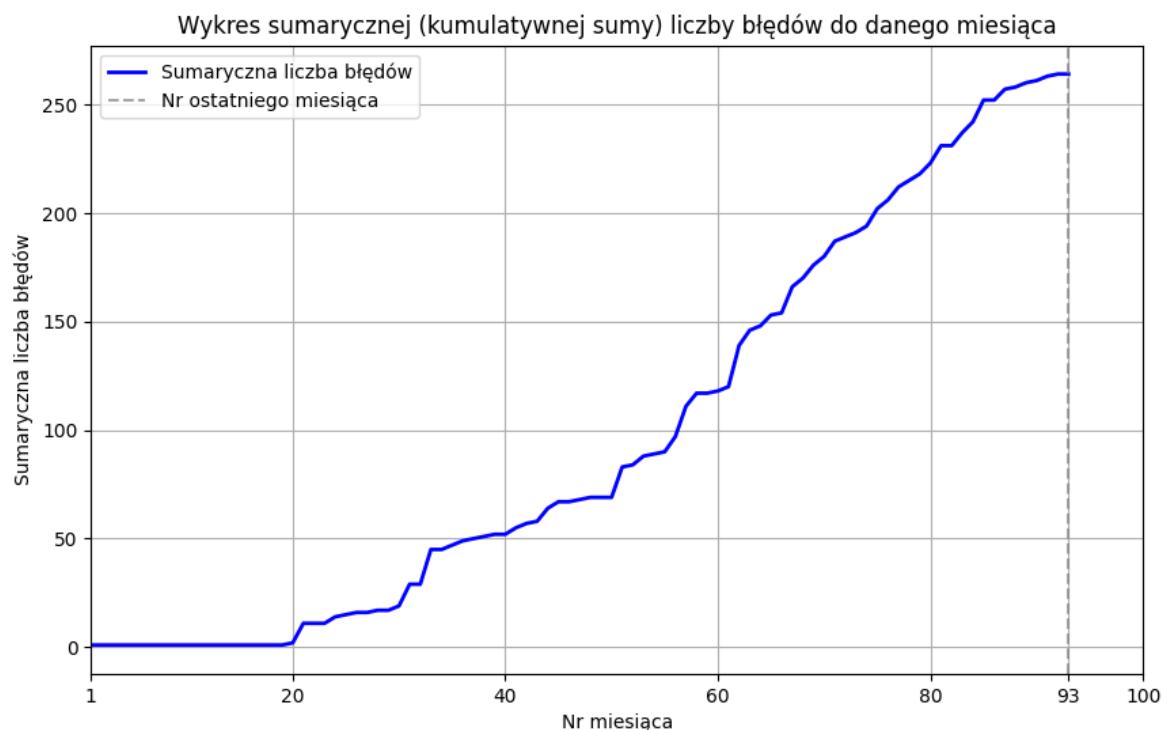
W zadaniu proszę:

1. Wyznaczyć zależność sumarycznej liczby błędów w okresie eksploatacji oprogramowania badanego systemu operacyjnego.
2. Na podstawie opracowanych modeli i przeprowadzonych obliczeń sformułować własne wnioski.
3. Wyniki analizy proszę zatrzymać w postaci sprawozdania, do którego proszę dodać jako załączniki wszystkie pliki z obliczeniami (obliczenia można przeprowadzić w dowolnie wybranym narzędziu)

## Rozdział III. Implementacja modeli

### III.1. Model logistyczny

W pierwszej kolejności przekształcono dane o miesięcznej liczbie wykrytych błędów krytycznych. Dodano do danych kolumnę przedstawiającą skumulowaną liczbę błędów do danego miesiąca. Przedstawiono to na poniższym wykresie.



Rys. 1. Wykres kumulatywnej sumy liczby błędów do danego miesiąca

Jak widać po wykresie i analizie sumarycznej liczby błędów krytycznych, wykres ukazuje charakterystyczny dla krzywych sigmoidalnych kształt litery „S”. W początkowej fazie eksploatacji systemu (do ok. 20 miesiąca) przyrost liczby błędów był bardzo mały, znikomy. Następnie, obserwujemy gwałtowny wzrost dynamiki wykrywania błędów krytycznych, co sugeruje fazę aktywnego użytkowania systemu operacyjnego i jego testy. Pod koniec badanego okresu, krzywa wyraźnie stabilizuje się, wykazuje tendencję do nasycenia, tj. spłaszczenia się, co oznacza, że tempo wykrywania nowych błędów spada. Uzasadnione jest zatem użycie modelu logistycznego. Warto także zauważyć, że punkt kulminacyjny wykrywania błędów następuje mniej więcej pośrodku badanego okresu, więc uzasadnione jest użycie później w doborze parametrów mediany czasu jako wartości początkowej dla algorytmu estymacji parametrów.

W kolejnym etapie rozwiązania, wyznaczono parametry funkcji logistycznej metodą opisaną we wstępnie teoretycznym: **Błąd! Nie można odnaleźć źródła**

**odwołania..** Wyznaczone parametry przedstawiono poniżej oraz obliczono współczynnik determinacji dopasowania modelu do danych.

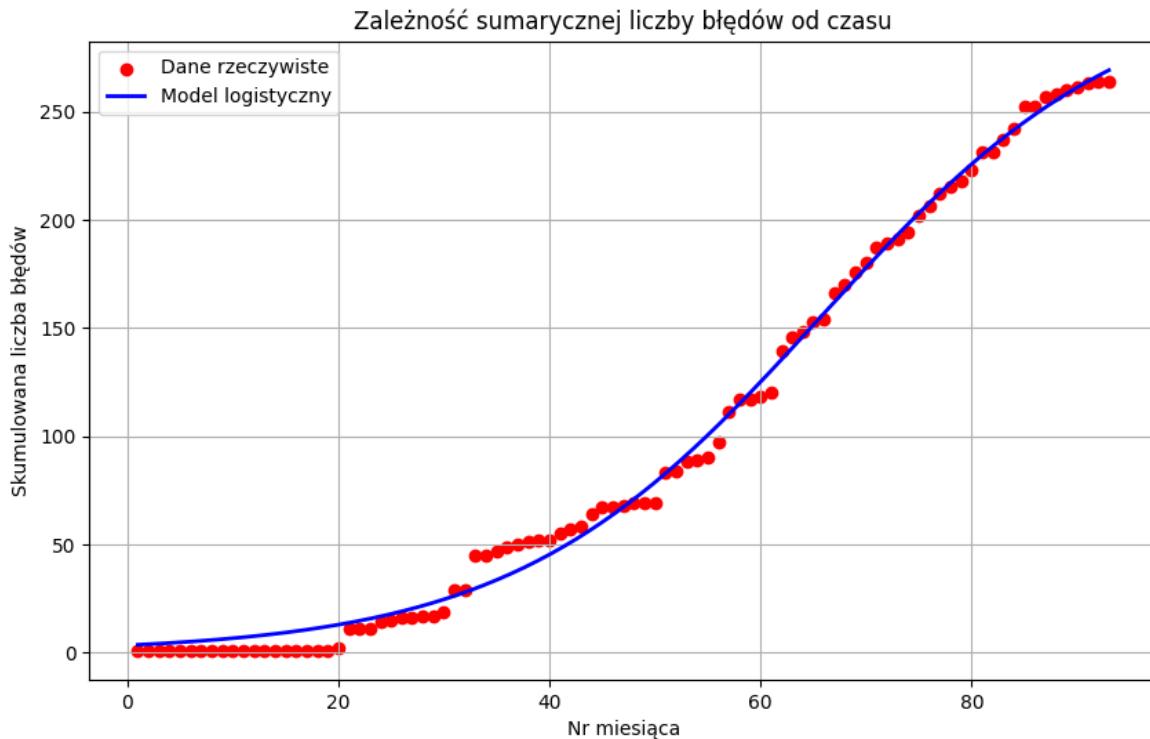
$$K \approx 310,38$$

$$\alpha \approx 0,069$$

$$t_0 \approx 65,73$$

$$R^2 \approx 99,6\%$$

Dopasowanie modelu do danych przedstawiono także na poniższym wykresie.



Rys. 2. Wykres zależności sumarycznej liczby błędów od czasu (nr miesiąca)

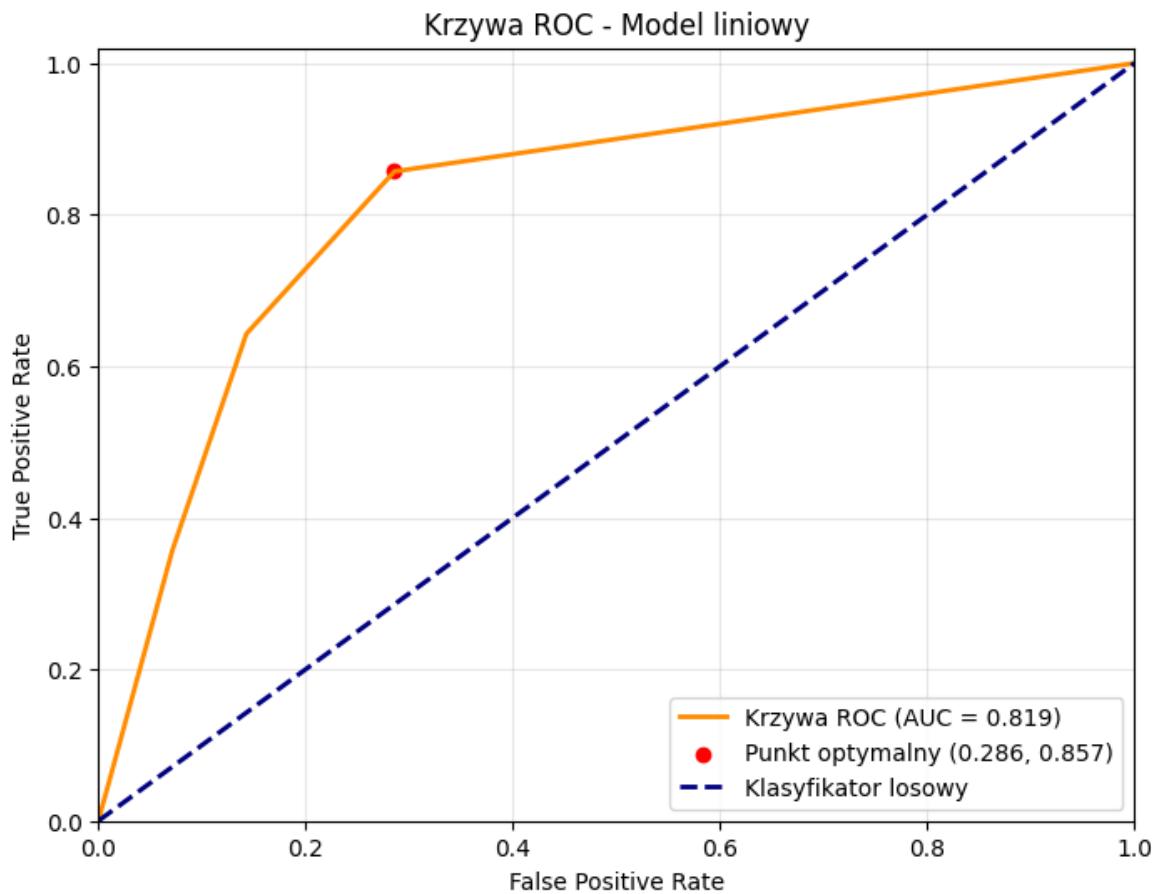
Nasz model logistyczny bardzo dobrze opisuje proces wykrywania błędów krytycznych w badanym systemie operacyjnym, świadczy o tym wysoki współczynnik determinacji równy w przybliżeniu 99,6%. Krzywa niemalże idealnie pokrywa się z danymi.

Parametr nasycenia wynosi ok. 310,38. Oznacza to, że teoretyczna, całkowita sumaryczna liczba błędów krytycznych, które mogą wystąpić w okresie eksploatacji systemu operacyjnego wynosi ok. 310. Biorąc pod uwagę, że w ciągu 93 miesięcy eksploatacji wykryto dotychczas 264 błędy, można wnioskować, że wykryto ok. 85% wszystkich przewidywanych błędów krytycznych.

Parametr tempa wzrostu wynosi ok. 0,069, co odzwierciedla umiarkowane tempo wzrostu sumarycznej liczby błędów. Wartość ta wskazuje na stabilny i dość długofalowy proces pojawiania się błędów krytycznych w czasie jego eksploatacji.

Punkt przegięcia wynosi ok. 65,73, co odpowiada mniej więcej końcowi 65. miesiąca eksploatacji systemu operacyjnego w badanym okresie. Po przekroczeniu tego miesiąca,

tempo wykrywania kolejnych błędów krytycznych, liczby kolejnych błędów zaczęło stopniowo maleć, co na wykresie objawia się przejściem krzywej w fazę spłaszczenia się.



Rys. 3. Wykres krzywej ROC dla modelu liniowego prawdopodobieństwa

Rysując krzywą ROC wyznaczono także optymalny próg klasyfikacji (0,507), który w przybliżeniu jest równy naszemu, odgórnie przyjętemu, równemu 0,5.

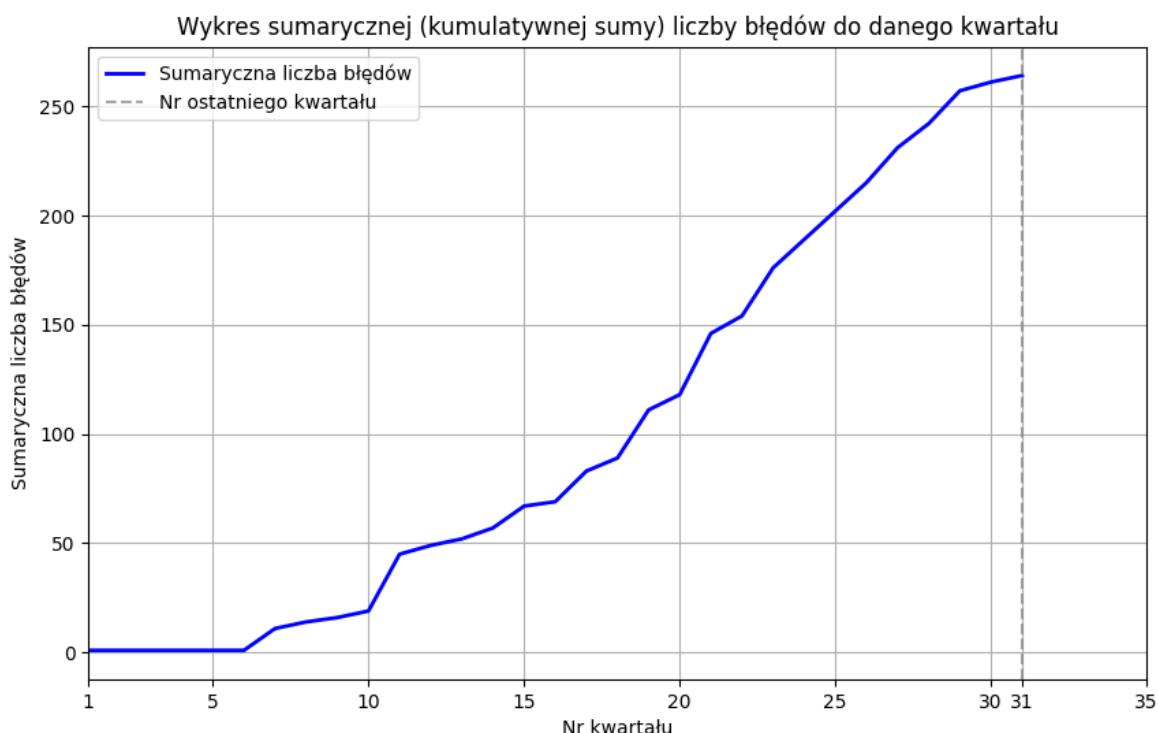
### III.2. Modele dla zagregowanych danych oraz podzielonych danych

W celu zbadania zależności oraz dopasowania sumarycznej liczby błędów do krzywej funkcji logistycznej, zdecydowano także na zbudowanie modeli dla:

1. Danych podzielonych na kwartały
2. Danych do 63. miesiąca (założenie, że jesteśmy w 63. miesiącu i nie mamy informacji o kolejnych)
3. Kwartałów z podziałem danych do 21. kwartału (założenie, że jesteśmy w 21. kwartale i nie mamy informacji o kolejnych)

### III.2.1. Model dla zagregowanych danych na kwartały

W pierwszej kolejności przekształcono dane o miesięcznej liczbie wykrytych błędów krytycznych, aby otrzymać dane kwartalne. Dodano do danych kolumnę przedstawiającą skumulowaną liczbę błędów do danego kwartału. Przedstawiono to na poniższym wykresie.



Rys. 4. Wykres kumulatywnej sumy liczby błędów do danego kwartału

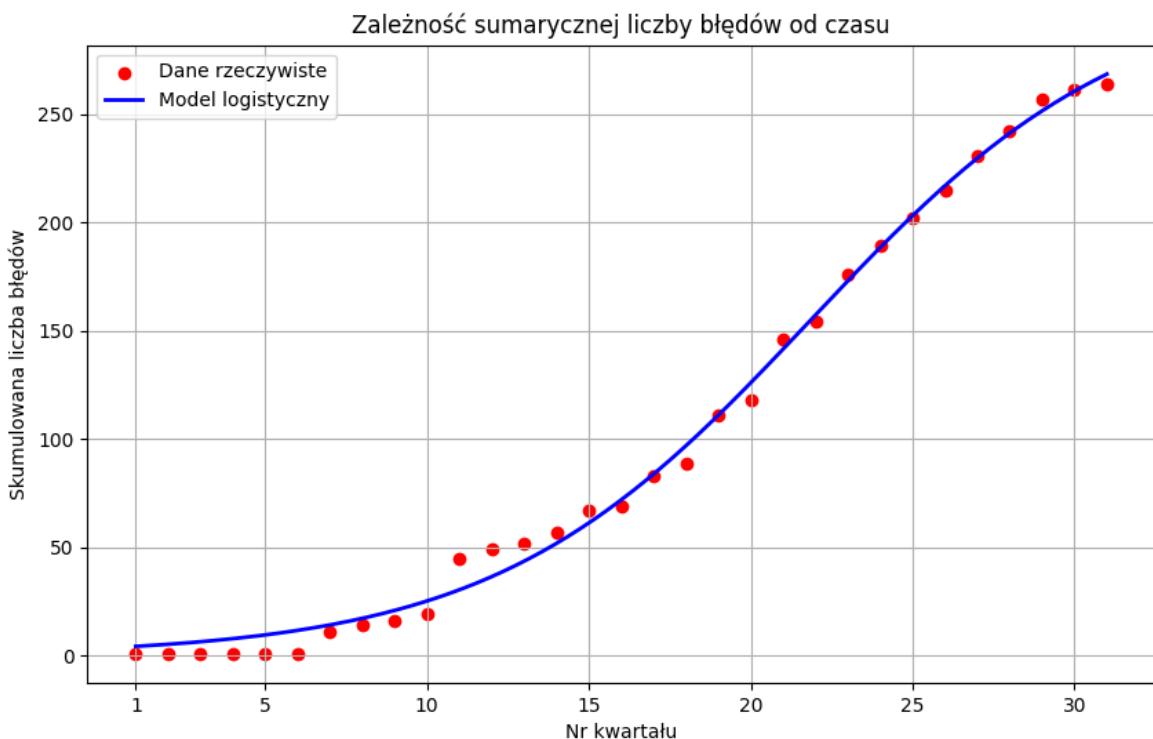
Jak widać po wykresie i analizie sumarycznej liczby błędów krytycznych, wykres ten również ukazuje charakterystyczny dla krzywych sigmoidalnych kształt litery „S”. W początkowej fazie eksploatacji systemu przyrost liczby błędów był bardzo mały, znikomy. Następnie, obserwujemy gwałtowny wzrost dynamiki wykrywania błędów krytycznych, co sugeruje fazę aktywnego użytkowania systemu operacyjnego i jego testy. Pod koniec badanego okresu, krzywa wyraźnie stabilizuje się, wykazuje tendencję do nasycenia, tj. spłaszcza się, co oznacza, że tempo wykrywania nowych błędów spada. Uzasadnione jest zatem użycie modelu logistycznego. Warto także zauważyć, że punkt kulminacyjny wykrywania błędów następuje mniej więcej pośrodku badanego okresu, więc uzasadnione jest użycie później w doborze parametrów mediany czasu jako wartości początkowej dla algorytmu estymacji parametrów.

W kolejnym etapie rozwiązania, wyznaczono parametry funkcji logistycznej metodą opisaną we wstępie teoretycznym: **Błąd! Nie można odnaleźć źródła odwołania..** Wyznaczone parametry przedstawiono poniżej oraz obliczono współczynnik determinacji dopasowania modelu do danych.

$$K \approx 309,76$$

$$\begin{aligned}a &\approx 0,204 \\t_0 &\approx 21,85 \\R^2 &\approx 99,6\%\end{aligned}$$

Dopasowanie modelu do danych przedstawiono także na poniższym wykresie.

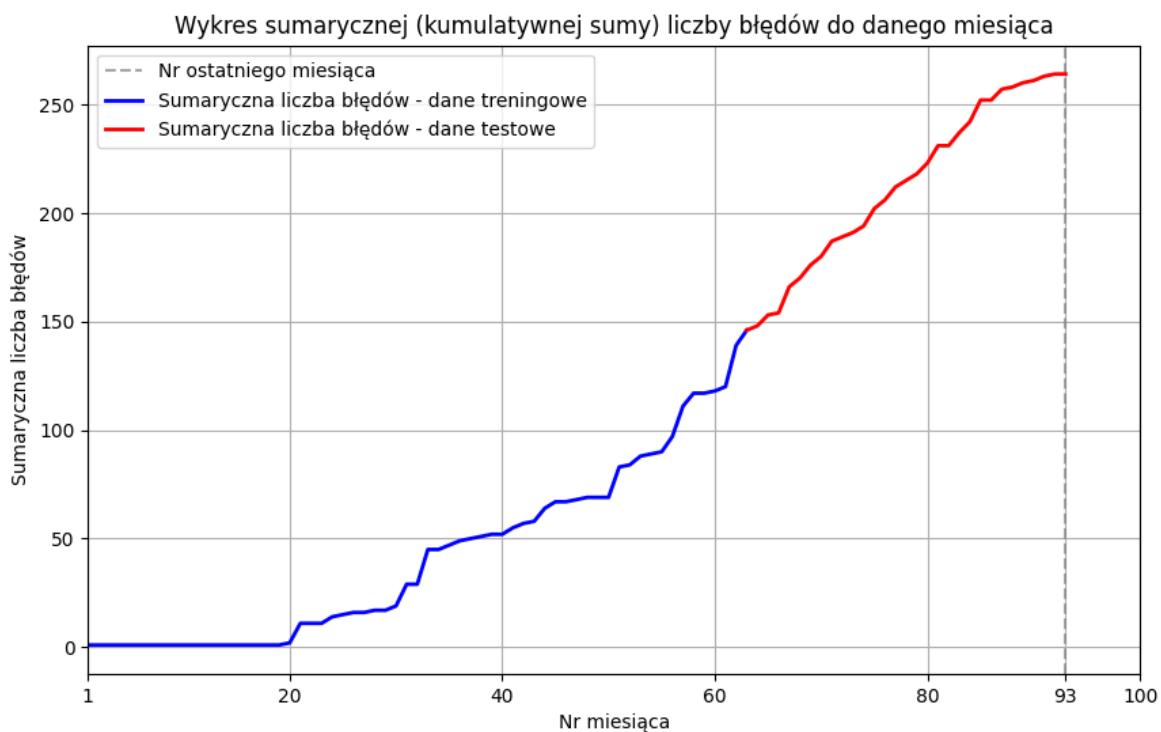


Rys. 5. Wykres zależności sumarycznej liczby błędów od czasu (Nr kwartału)

Wartości nasycenia (maksymalna, estymowana liczba błędów) oraz punkt przegięcia minimalnie zmalały natomiast tempo wzrostu lekko wzrosło. Model również bardzo dobrze opisuje proces wykrywania błędów w badanym systemie operacyjnym, minimalnie gorzej niż model z miesiącami oraz bez podziału zbioru danych ze względu na minimalnie mniejszą wartość nasycenia. Świadczy o tym współczynnik determinacji równy ok. 99,6%, co odpowiada pierwszemu zbudowanemu modelowi. Na wykresie widać również, że krzywa niemalże idealnie pokrywa się z danymi.

### III.2.2. Model dla podzielonych danych (do 63. miesiąca)

W drugiej kolejności przekształcono dane o miesięcznej liczbie wykrytych błędów krytycznych i do dalszych analiz użyto danych do 63. miesiąca włącznie. Dodano do danych kolumnę przedstawiającą skumulowaną liczbę błędów do danego miesiąca. Dane uwzględnione w tej analizie oraz dane testowe (te, o których nie wiemy, ze względu na założenie, że jesteśmy w 63. miesiącu) przedstawiono na poniższym wykresie odpowiednio kolorami niebieskim i czerwonym.



Rys. 6. Wykres kumulatywnej sumy liczby błędów do danego miesiąca z podziałem

Jak widać po wykresie i analizie sumarycznej liczby błędów krytycznych dla badanego okresu do 63. miesiąca, wykres nie ukazuje już jednoznacznie charakterystycznego dla krzywych sigmoidalnych kształtu litery „S”. W początkowej fazie eksploatacji systemu przyrost liczby błędów był bardzo mały, znikomy. Następnie, obserwujemy gwałtowny wzrost dynamiki wykrywania błędów krytycznych, co sugeruje fazę aktywnego użytkowania systemu operacyjnego i jego testy. Pod koniec badanego okresu, krzywa nie ukazuje już wyraźnego procesu stabilizacji, tak jak w poprzednich modelach. Niemniej, w celu eksperymentu i zbadania tego modelu dla tych danych też użyliśmy modelu logistycznego. W doborze parametrów użyto tych samych wartości początkowych jak w poprzednich modelach, tj. m.in. mediany czasu jako wartości początkowej dla algorytmu estymacji parametrów.

W kolejnym etapie rozwiązania, wyznaczono parametry funkcji logistycznej metodą opisaną we wstępie teoretycznym: **Błąd! Nie można odnaleźć źródła odwołania..** Wyznaczone parametry przedstawiono poniżej oraz obliczono współczynnik determinacji dopasowania modelu do danych.

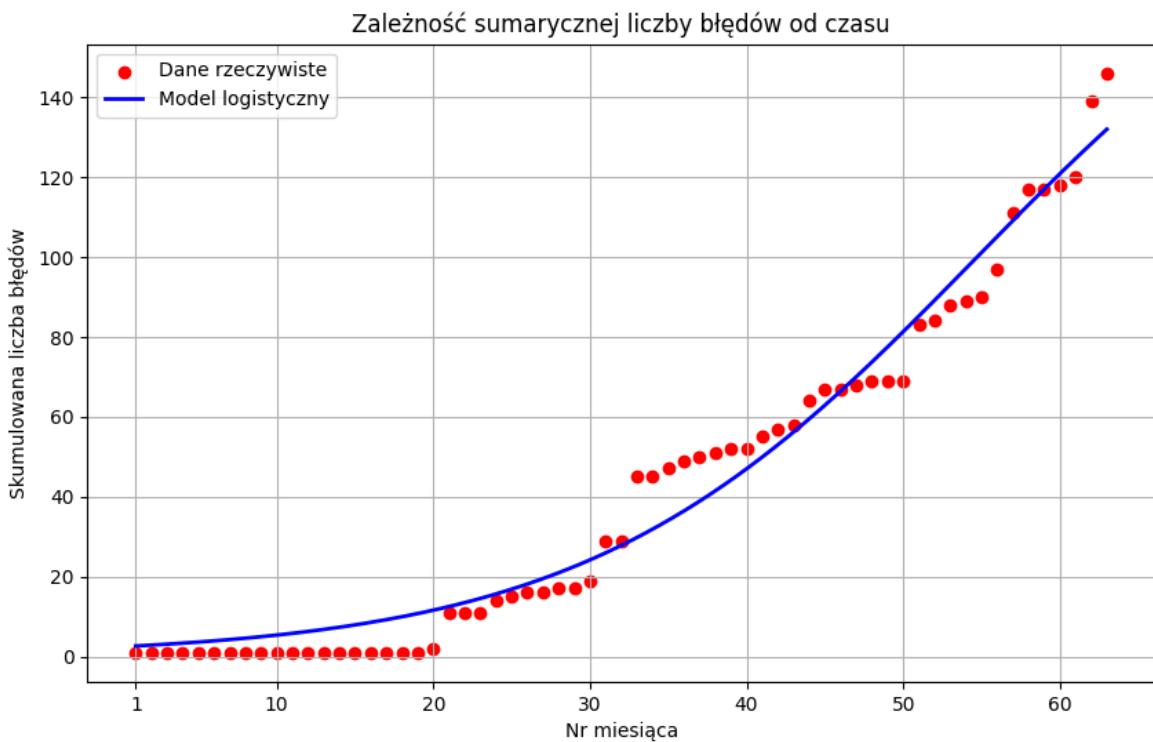
$$K \approx 199,28$$

$$a \approx 0,080$$

$$t_0 \approx 54,63$$

$$R^2 \approx 97,4\%$$

Dopasowanie modelu do danych przedstawiono także na poniższym wykresie.

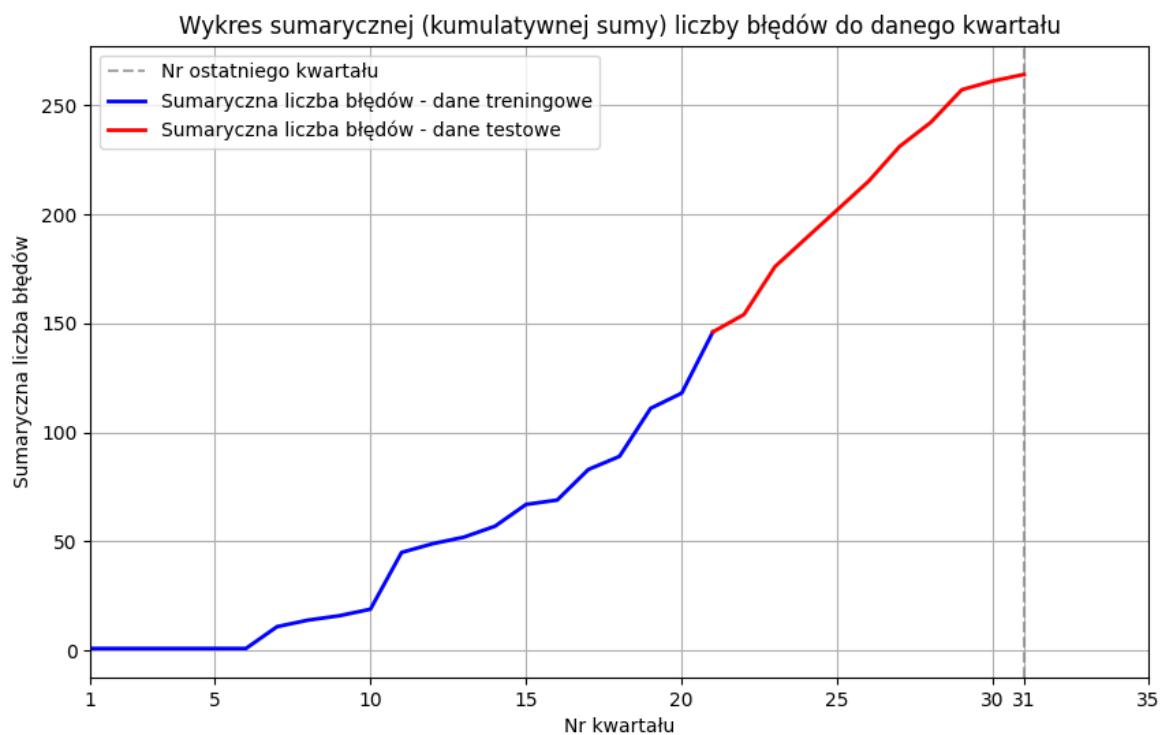


Rys. 7. Wykres zależności sumarycznej liczby błędów od czasu (Nr miesiąca) z podziałem

Wartości nasycenia oraz punkt przegięcia drastycznie zmalały. Tempo wzrostu wzrosło. Można było się tego spodziewać, gdyż zbiór rozpatrywany w tym modelu (zbiór treningowy) pomniejszył się ze względu na rozdzielenie danych (do budowy modelu użyto dane do 63. miesiąca). Mimo tego, model nadal bardzo dobrze opisuje dane, aczkolwiek to dopasowanie jest gorsze od 2 poprzednich modeli, ponieważ wynosi ok. 97,4%. Warto jednak zauważyć, że nie jest to dobry model gdyż maksymalna liczba błędów krytycznych badanego systemu operacyjnego jest bardzo zaniżona porównując ją z tą wyznaczoną dla poprzednich modeli dla pełnych danych. Przyjęcie tego modelu czy też zdecydowanie się na nie czekanie do tego co najmniej 93. miesiąca mogłoby skutkować błędnym założeniem maksymalnej liczby błędów, które wystąpiłyby w systemie, co z kolei mogłoby mieć poważne skutki w procesie wdrażania i późniejszej eksploatacji systemu operacyjnego.

### III.2.3. Model dla zagregowanych danych na kwartały z podziałem (do 21. kwartału)

W pierwszej kolejności przekształcono dane o miesięcznej liczbie wykrytych błędów krytycznych, aby otrzymać dane kwartalne a następnie podzielono dane na zbiór treningowy i testowy. W dalszych analizach tego modelu, uwzględniono dane treningowe, czyli dane do 21. kwartału, zgodnie z założeniem, że znajdujemy się w tym okresie i nie mamy informacji o kolejnych błędach. Dodano do danych kolumnę przedstawiającą skumulowaną liczbę błędów do danego kwartału. Przedstawiono to na poniższym wykresie.



Rys. 8. Wykres kumulatywnej sumy liczby błędów do danego kwartału z podziałem

Jak widać po wykresie i analizie sumarycznej liczby błędów krytycznych dla badanego okresu do 21. kwartału, wykres nie ukazuje już jednoznacznie charakterystycznego dla krzywych sigmoidalnych kształtu litery „S”. W początkowej fazie eksploatacji systemu przyrost liczby błędów był bardzo mały, znikomy. Następnie, obserwujemy gwałtowny wzrost dynamiki wykrywania błędów krytycznych, co sugeruje fazę aktywnego użytkowania systemu operacyjnego i jego testy. Pod koniec badanego okresu, krzywa nie ukazuje już wyraźnego procesu stabilizacji, tak jak w poprzednich modelach. Niemniej, w celu eksperymentu i zbadania tego modelu dla tych danych też użyliśmy modelu logistycznego. W doborze parametrów użyto tych samych wartości początkowych jak w poprzednich modelach, tj. m.in. mediany czasu jako wartości początkowej dla algorytmu estymacji parametrów.

W kolejnym etapie rozwiązania, wyznaczono parametry funkcji logistycznej metodą opisaną we wstępnie teoretycznym: **Błąd! Nie można odnaleźć źródła odwołania..** Wyznaczone parametry przedstawiono poniżej oraz obliczono współczynnik determinacji dopasowania modelu do danych.

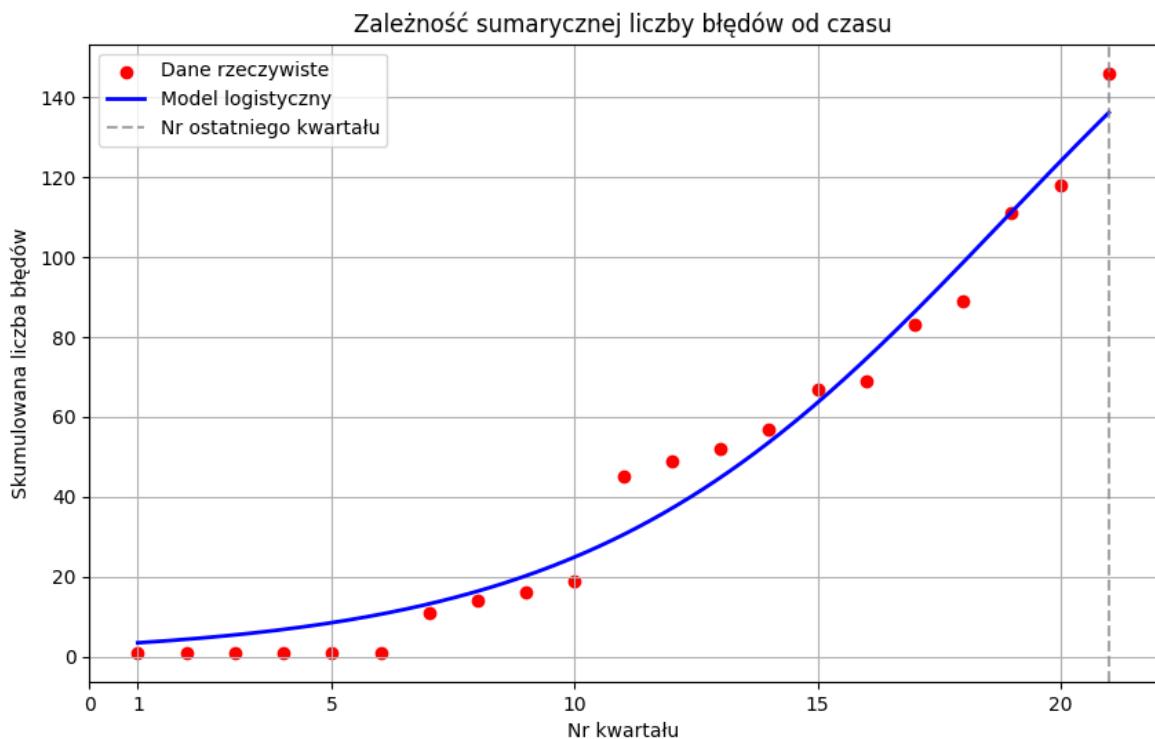
$$K \approx 219,05$$

$$a \approx 0,232$$

$$t_0 \approx 18,86$$

$$R^2 \approx 97,6\%$$

Dopasowanie modelu do danych przedstawiono także na poniższym wykresie.



Rys. 9. Wykres zależności sumarycznej liczby błędów od czasu (Nr kwartału) z podziałem

Tempo wzrostu wzrosło oraz punkt przegięcia zmalał w porównaniu z poprzednim modelem zagregowanym do kwartałów. Można natomiast zauważać ciekawy wynik, mianowicie, nasycenie wzrosło w porównaniu z modelem, w którym też zastosowano podział danych. Jak widać, agregacja danych do kwartałów zamiast pozostawienia miesięcy pomogła trochę polepszyć ten model. On jednak również nie jest najlepszy, bo maksymalna, estymowana liczba błędów (nasycenie) nadal jest bardzo zaniżona w porównaniu z modelem z pełnymi danymi bez podziału (pierwszy model, który zbudowano). Przyjęcie tego modelu czy też zdecydowanie się na nie czekanie do co najmniej 31. kwartału (lub 93. miesiąca w przypadku danych bez agregacji) także skutkowałoby błędnym założeniem maksymalnej, estymowanej liczby błędów, które wystąpiłyby w badanym systemie operacyjnym, co z kolei mogłoby mieć poważne skutki w procesie wdrażania i późniejszej eksploatacji systemu. Natomiast, samo dopasowanie danych do modelu jest nieco gorsze od poprzednich modeli bez podziału danych (ok. 99,6%), ale nieco lepsze od modelu bez agregacji danych do kwartałów, ale z podziałem danych (ok. 97,4%), gdyż wynosi ok. 97,6%.

## Podsumowanie

Proces pojawiania się błędów w badanym systemie operacyjnym przebiega zgodnie z modelem logistycznym, czego m.in. dowodzi charakterystyczny kształt krzywej – litera „S”. Początkowy powolny wzrost liczby błędów, następna faza intensywnego występowania błędów (przyrost liczby błędów) oraz końcowe wygasanie trendu wzrostowego są dość typowe dla oprogramowania. Choć system zbliża się do fazy nasycenia, parametr K sugeruje, że proces utrzymania tego systemu powinien być kontynuowany, gdyż system nie osiągnął jeszcze pełnego, teoretycznego poziomu bezawaryjności – nie osiągnął teoretycznej maksymalnej liczby błędów krytycznych, która jest reprezentowana przez obliczoną wartość K. Analiza dodatkowych zbudowanych modeli wykazała, że modele wymagają jak największej liczby danych, aby wiarygodnie estymować poziom nasycenia, który odpowiada estymowanej, maksymalnej liczbie błędów krytycznych badanego systemu operacyjnego. Zbyt wcześnie „ucięcie” danych sugeruje mniejszą maksymalną liczbę błędów systemu operacyjnego, co może w praktyce mieć poważne skutki dla użytkowanego systemu. Agregacja danych do kwartałów pozwoliła na zwiększenie, w przypadku modelu z podziałem danych, jakości estymacji nasycenia, w porównaniu do analogicznego modelu dla danych miesięcznych. Jednakże, analizując wszystkie zbudowane modele, trzeba zauważyć, że największą wartość biznesową ma model zbudowany dla całych danych w ujęciu miesięcznym, gdyż ma najwyższą wartość dopasowania danych do modelu oraz wartość nasycenia.

## Bibliografia

- [1] Hoffmann R.: „Metody eksploracji danych - Wykład 4. Regresja logistyczna Cz I slajdy”, Materiały dydaktyczne WAT, 2025.
- [2] [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve\\_fit.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html)
- [3] <https://www.youtube.com/watch?v=pNE1ufDXNSc&t=3s>
- [4] [https://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt\\_algorithm](https://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm)

**Spis tabel**

Tab. 1.	Liczba błędów krytycznych oprogramowania badanego OS w układzie miesięcznym.....	8
---------	--	---

**Spis rysunków**

Rys. 1.	Wykres kumulatywnej sumy liczby błędów do danego miesiąca.....	9
Rys. 2.	Wykres zależności sumarycznej liczby błędów od czasu (nr miesiąca).....	10
Rys. 3.	Wykres krzywej ROC dla modelu liniowego prawdopodobieństwa .....	11
Rys. 4.	Wykres kumulatywnej sumy liczby błędów do danego kwartału.....	12
Rys. 5.	Wykres zależności sumarycznej liczby błędów od czasu (Nr kwartału) ....	13
Rys. 6.	Wykres kumulatywnej sumy liczby błędów do danego miesiąca z podziałem 14	
Rys. 7.	Wykres zależności sumarycznej liczby błędów od czasu (Nr miesiąca) z podziałem 15	
Rys. 8.	Wykres kumulatywnej sumy liczby błędów do danego kwartału z podziałem 16	
Rys. 9.	Wykres zależności sumarycznej liczby błędów od czasu (Nr kwartału) z podziałem 17	

**Załączniki**

1. Plik notebook *Lab-2-Zadanie-2-Obliczenia.ipynb* – kompletny kod przeprowadzonych eksperymentów