Miniprojekt 2

Na początku stworzyłem moduł **division_and_scaling.py**, w którym dzielę w losowy sposób dane wejściowe na zbiór treningowy (2/3 obserwacji) i testowy (1/3 obserwacji) w taki sposób, żeby każda z klas miała w zbiorze treningowym swoje 2/3 obserwacji. Ponadto gdy badam regresją logistyczną to w module jest możliwość przeskalowania danych metodami: **min-max1, min-max2, standaryzacja.** Metaparametry skalowania, czyli średnia, odchylenie standardowe, minimum, maximum wyznaczam na podstawie zbioru treningowego, a następnie stosuję w procesie skalowania zarówno dla zbioru treningowego jak i testowego.

Następnie stworzyłem foldery odpowiadające implementowanym metodom:

- Naiwny klasyfikator bayesowski
- Regresja logistyczna

W folderze naiwnego klasyfikatora bayesowskiego zaimplementowałem następująca **regułę decyzyjną**:

y=T $\Leftrightarrow$

$$\log \frac{(\prod_{i=1}^{n} p(x_i|y=T))p(y=T)}{(\prod_{i=1}^{n} p(x_i|y=F))p(y=F)} = \sum_{i=0}^{n} \log\left(\frac{p(x_i \mid y=T)}{p(x_i \mid y=F)}\right) + \log\left(\frac{p(y=T)}{p(xy=F)}\right) \geq 0$$

Przy wyznaczaniu prawdopodobieństw zastosowałem wygładzenie Laplace'a

$$p(x_i = a \mid y = b) = \frac{\#\{x_i = a,\ y = b\} + 1}{\#\{y = b\} + 10}$$

W folderze regresji logistycznej zaimplementowałem algorytm **stochastycznego gradientu** szukający minimalizującego koszt wektora θ. Przed rozpoczęciem algorytmu uzupełniam daną macierz X kolumną wypełnioną 1.

Współczynnik θ_j w naszym algorytmie aktualizujemy natomiast w następujący sposób:

$$\theta_j \leftarrow \theta_j + \alpha \cdot \left(y^{(i)} - sig\left(\theta^T x^{(i)}\right)\right) \cdot x_j^{(i)}$$

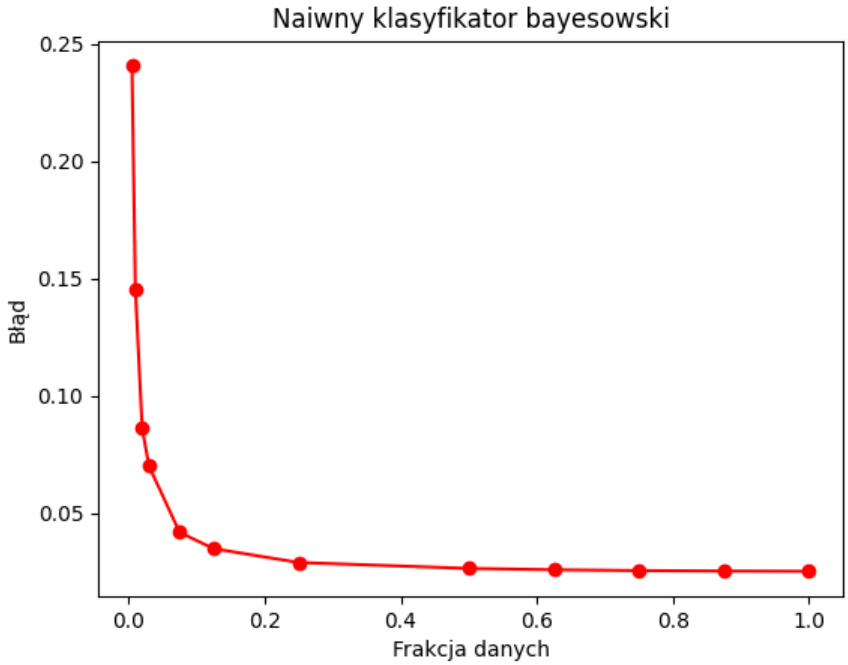
W folderze resources stworzyłem także foldery odpowiadającym powyższym metodom, w których zapisywałem wyniki / metaparametry obliczone w procesie uczenia się / testowania. Znajdują się tam też otrzymane wykresy zawierające uśrednione krzywe uczenia dla różnych podziałów obserwacji, a także wykresy średniej precyzji, czułości i miary F1.

Naiwny klasyfikator bayesowski

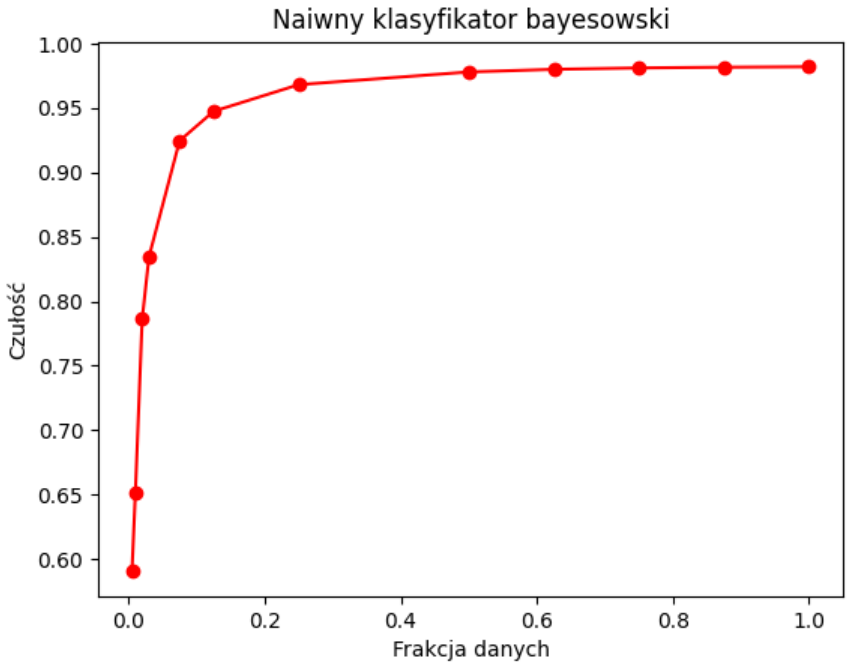
Z uwagi na szybkie działanie algorytmu podyktowane prostotą metody jaką jest naiwny klasyfikator bayesowski osiągane wyniki uśredniałem na 1000 przebiegach algorytmów dla losowo wybranych zbiorów treningowych i testowych.

Błędem w tej metodzie jest szacowane prawdopodobieństwo złej klasyfikacji, czyli $\frac{FN+FP}{TP+TN+FN+FP}$

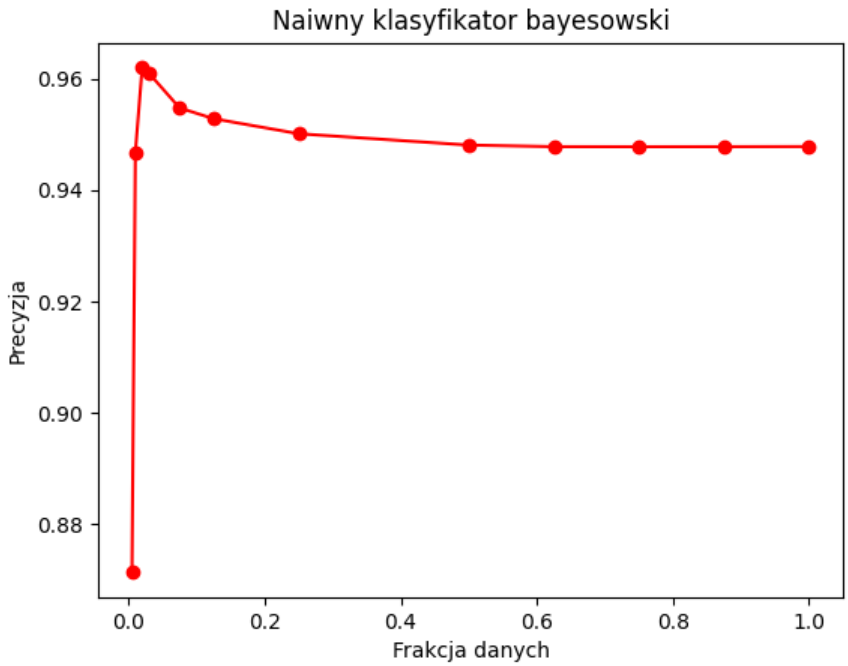
Analizując wykres krzywej uczenia można zauważyć bardzo niski błąd już dla małych frakcji zbioru treningowego – tzn. dla frakcji 0.005 algorytm osiągał średni błąd `0.24,` a dla frakcji 0.01 już tylko `0.145.` Wynika z tego, że algorytm nie potrzebował dużej liczby danych, a co więcej wystarczyło mu tylko kilka/kilkanaście obserwacji, do osiągnięcia dobrego poziomu klasyfikacji. Po osiągnięciu frakcji 0.125, metoda zwracała już średnio błąd tylko `0.035.` Dla kolejnych frakcji danych błąd już nie zmniejszał się znacząco - jedynie o setne części. Najlepszy wynik osiągnięto jednak ostatecznie dla frakcji 1.0 i był on równy `0.0253.`



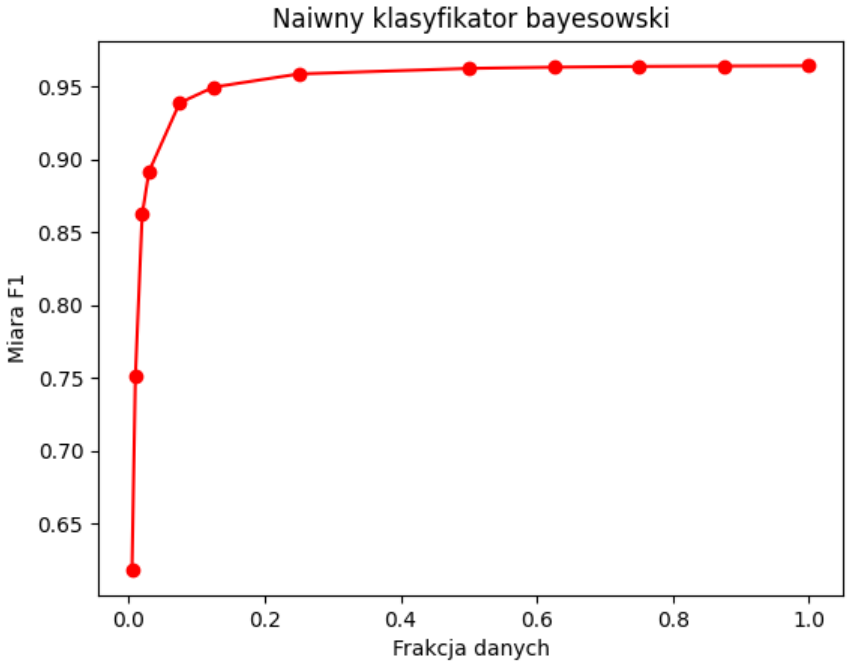
Czułość dla naszego modelu dla małych frakcji danych była średnio znacznie mniejsza niż precyzja - np. dla frakcji 0.005 precyzja to `0.871,` a natomiast czułość to `0.59.` Sytuacja zmienia się wraz ze wzrostem frakcji danych. Czułość znacząco rośnie i osiąga ostatecznie prawie 1.0 - `0.982,` co oznacza, że ta metoda wysoki poziom detekcji pozytywnych przypadków. Praktycznie nie ma przypadków FalseNegative, czyli złej klasyfikacji złośliwego raka piersi.



Precyzja początkowo też rośnie, jednakże swoje maksimum osiąga w frakcji 0.02, dla której wynosi średnio `0.962,` a potem nieznacznie spada osiągając ostatecznie we frakcji 1.0 wartość `0.947,` co jest też wysokim wskaźnikiem precyzji. Wynika z tego, że klasyfikator bayesowski ma całkiem niski poziom wszczynanie fałszywych alarmów



Miara F1 dla naiwnego klasyfikatora bayesowskiego prezentuje się natomiast następująco. Wykres wygląda podobnie do wykresu czułości z uwagi na charakter miary F1. Ostateczne wartości są jednak mniejsze.



Regresja logistyczna

W regresji logistycznej osiągane wyniki uśredniałem natomiast na 100 przebiegach algorytmów dla losowo wybranych zbiorów treningowych i testowych.

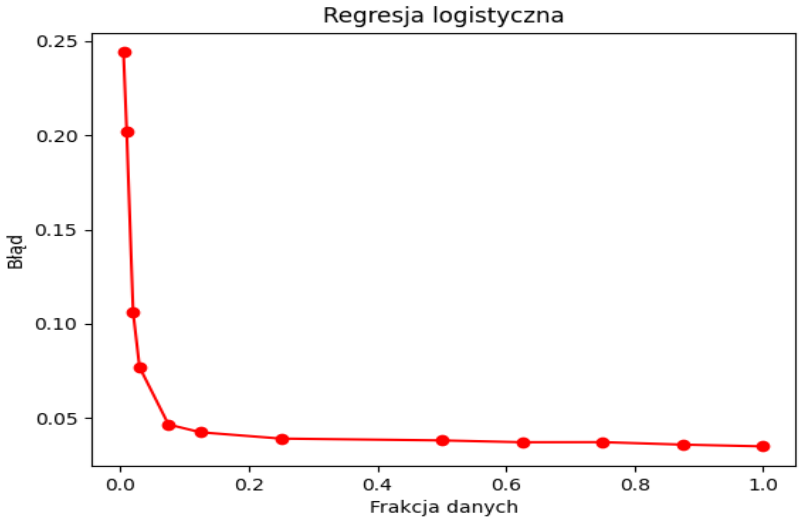
Podobnie jak w naiwnym klasyfikatorze bayesowskim błąd wykorzystywany przy ocenie modelu to $\frac{FN + FP}{TP+TN+FN+FP}$

Model regresji logistycznej badałem dla różnych skalowań danych początkowych:

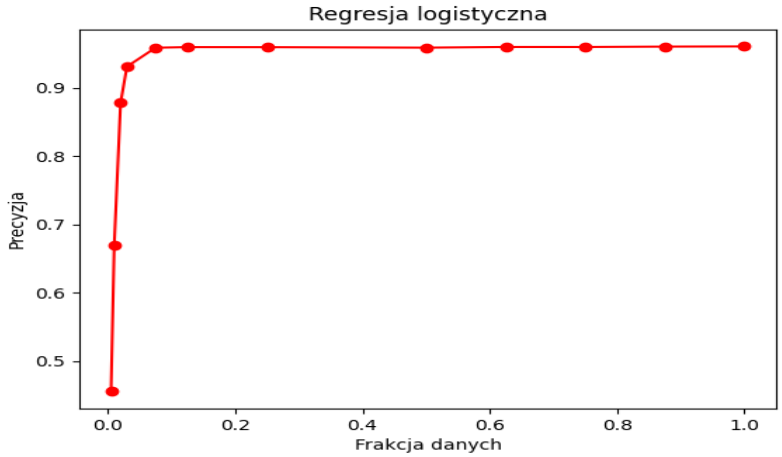
1. Min-max1

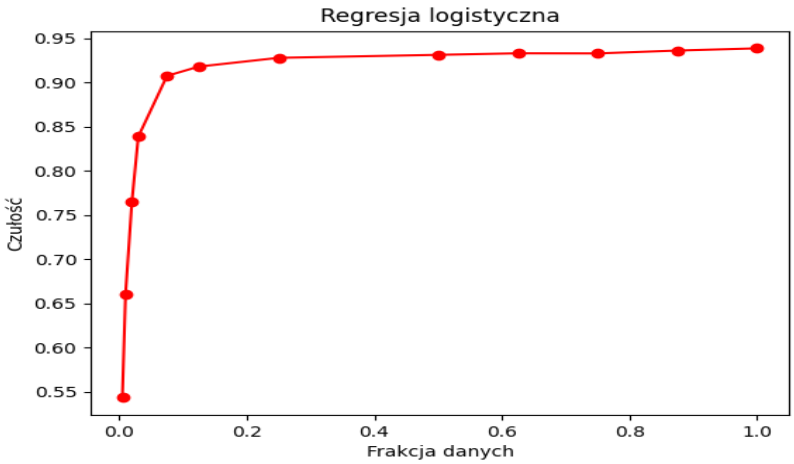
$$x^{(i)} \rightarrow \frac{x^{(i)} - \min(x)}{\max(x) - \min(x)}$$

Analizując wykres krzywej uczenia można zauważyć podobieństwo do wykresu otrzymanego z naiwnego klasyfikatora bayesowskiego (NKB). Już nawet na niewielkich frakcjach danych błąd jest poniżej 0.25. Przykładowo dla 0.005 mamy `0.243.` Jednakże model uczy się trochę wolniej niż w NKB, bo dla frakcji 0.01 otrzymujemy teraz `0.202.` Podobnie jak wcześniej po przekroczeniu frakcji 0.125, dla której dostałem wartość 0.046 otrzymywany błąd pozostawał na podobnym poziomie - zmniejszał się w nieznaczny sposób. Co ciekawe dla frakcji 1.0 osiągnięty błąd to `0.035,` czyli jest on większy niż dla NKB.

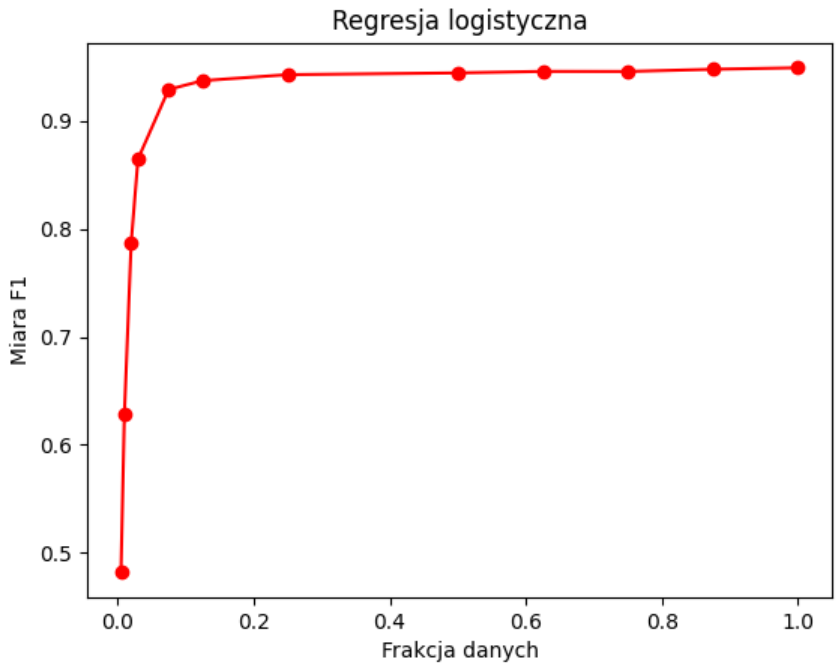


Wykres czułości i precyzji wygląda tym razem podobnie. Dla małych frakcji danych czułość i precyzja są w przedziale [0.5 - 0.6], a następnie rosną do wartości powyżej 0.9 dla frakcji większy niż 0.125. Co więcej dla małych frakcji danych przy liczeniu precyzji dostawałem ZeroDivisionError z czego wynika, że nie było wartości TP ani FP, co oznacza że klasyfikowaliśmy wszystko na N. Dla frakcji 1.0 precyzja wyniosła `0.9605,` a czułość `0.938,` czyli w tym przypadku precyzja była większa niż czułość, a także precyzja wyniosła więcej niż w NKB, a natomiast czułość o ok. 0.05 była gorsza niż NKB. Wynika z tego, że mniej alarmowaliśmy o fałszywych przypadkach choroby, ale mieliśmy niestety niższy poziom detekcji pozytywnych przypadków.





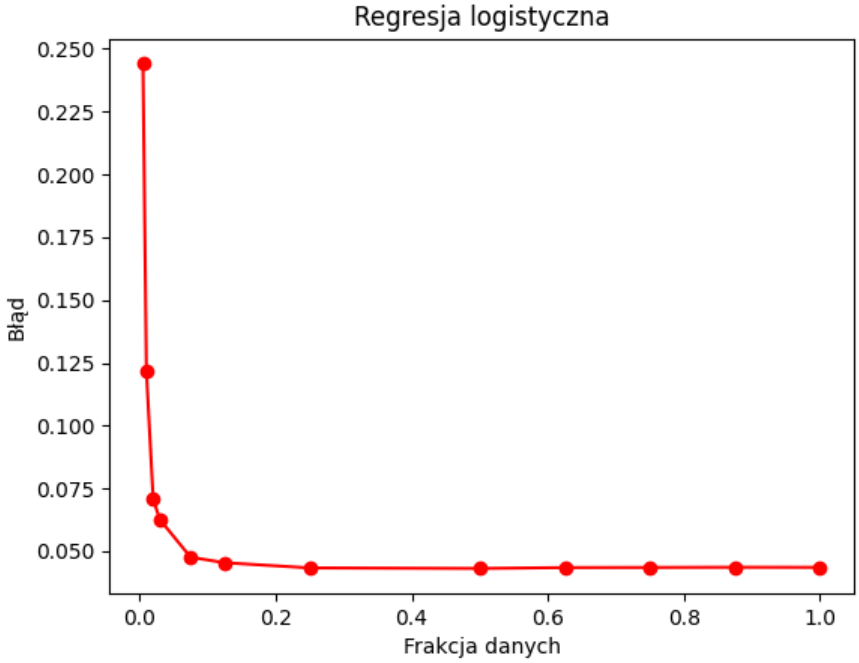
Wykres miary F1 wyglądał następująco:



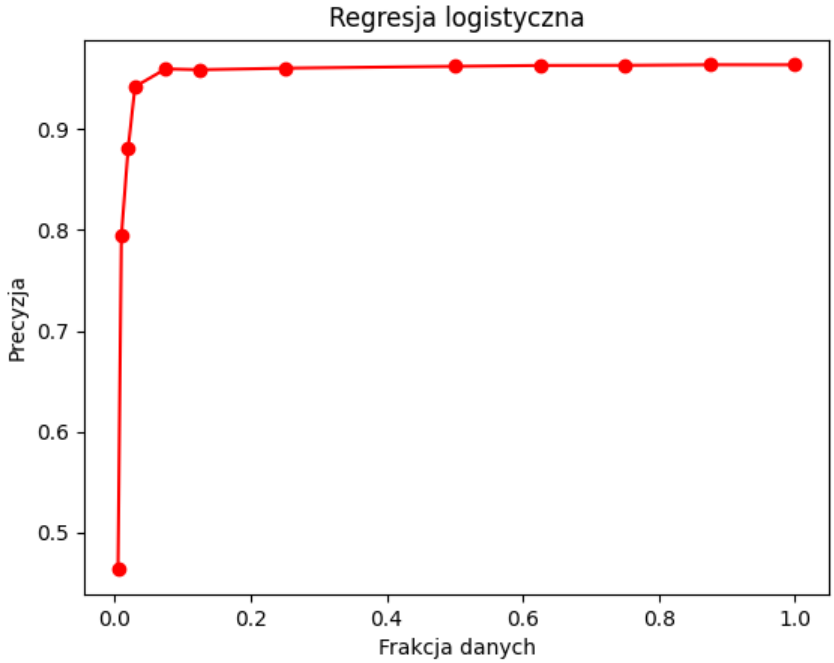
2. Min Max 2

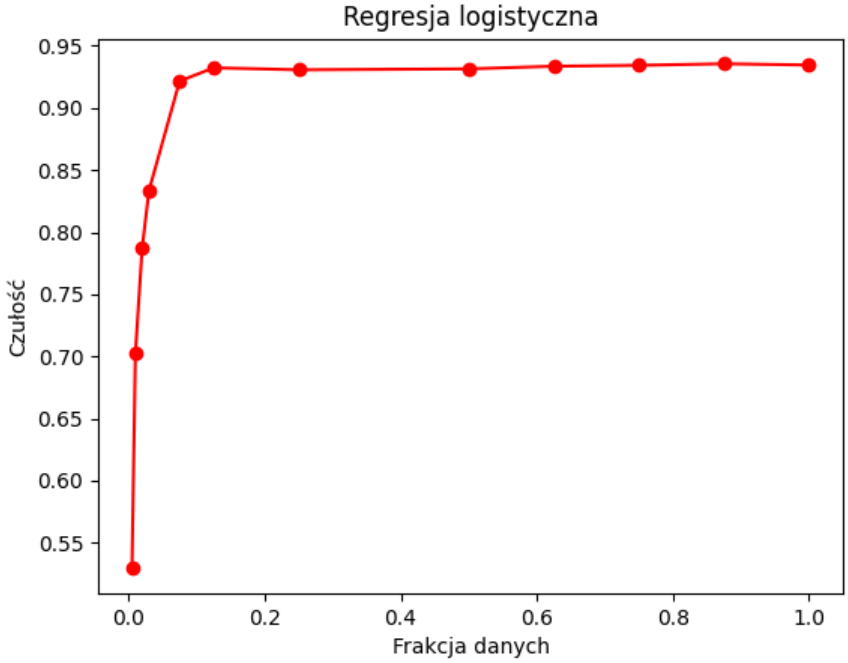
$$x^{(i)} \rightarrow \frac{x^{(i)} - \overline{x}}{\max(x) - \min(x)}$$

Wykres krzywej uczenia w tym przypadku jest bardzo zbliżony do wykresu regresji logistycznej z zastosowaniem skalowania min-max 1. Otrzymywane wyniki dla poszczególnych frakcji różnią się o niewielkie wartości często mniejsze niż 0.01. Najmniejszy błąd również jest dla frakcji 1.0, ale tym razem wynosi `0.0435` i jest trochę większy.

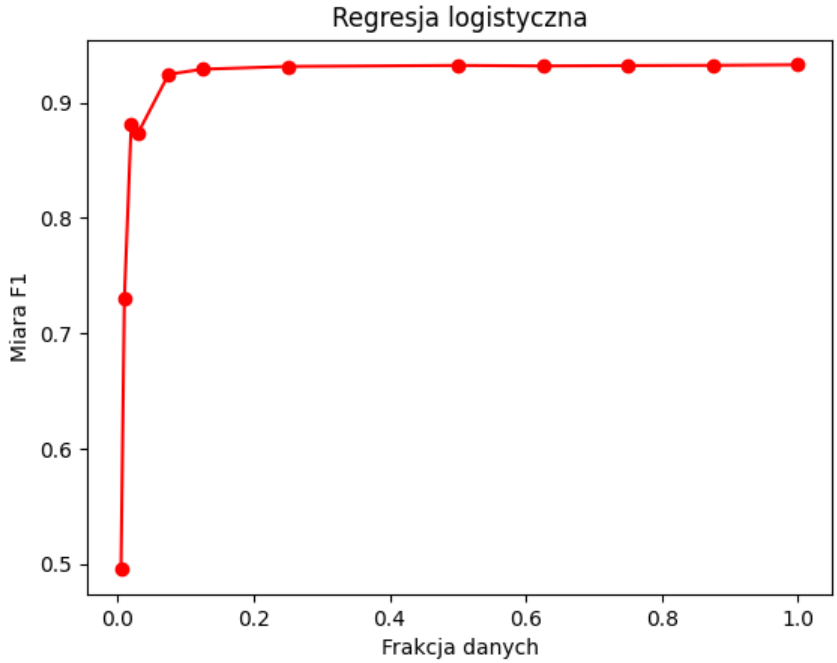


Kształt wykresu precyzji i czułości są też zbliżone co w min-max1. Warto jednak zauważyć, że precyzja dla małych frakcji danych jest wyższa niż wcześniej - tzn. dla 0.005 wynosi `0.739,` a ostateczny wynik dla 1.0 jest wyższy i wynosi 0.97. Wartości czułości są początkowo podobna co w min-max1, ale końcowy wynik jest jednak niższy i wynosi `0.9036.`





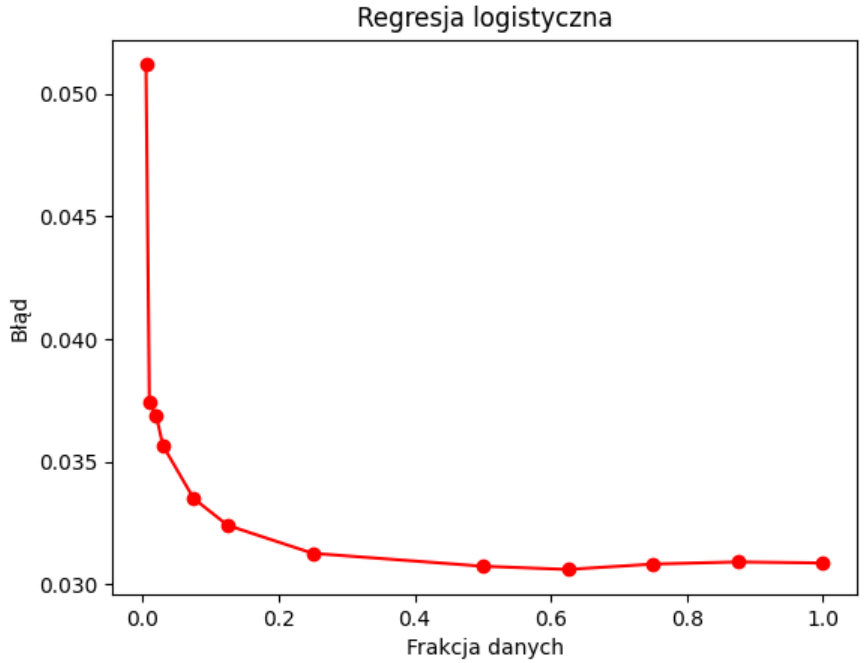
Wykres miary F1 wyglądał teraz następująco:



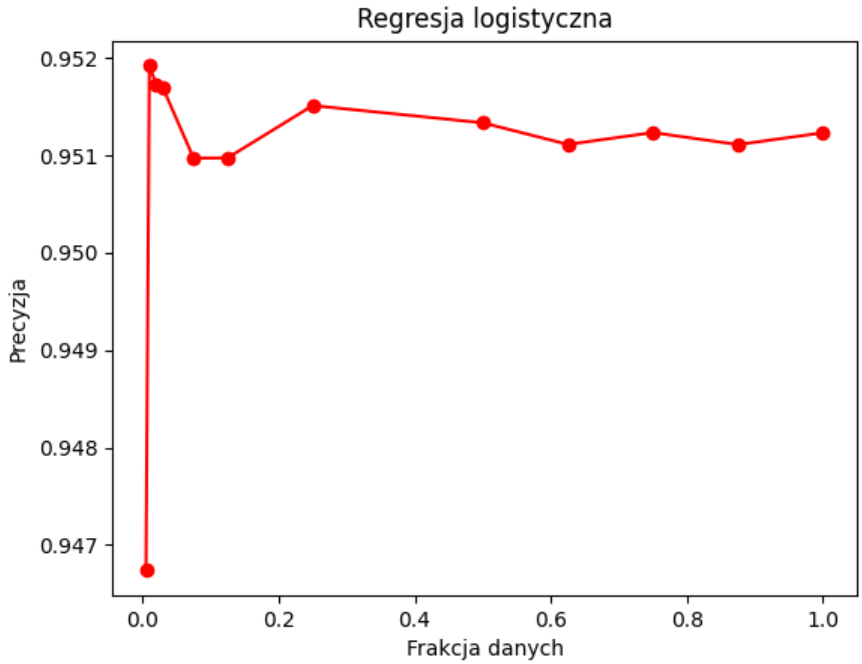
3. Standaryzacja

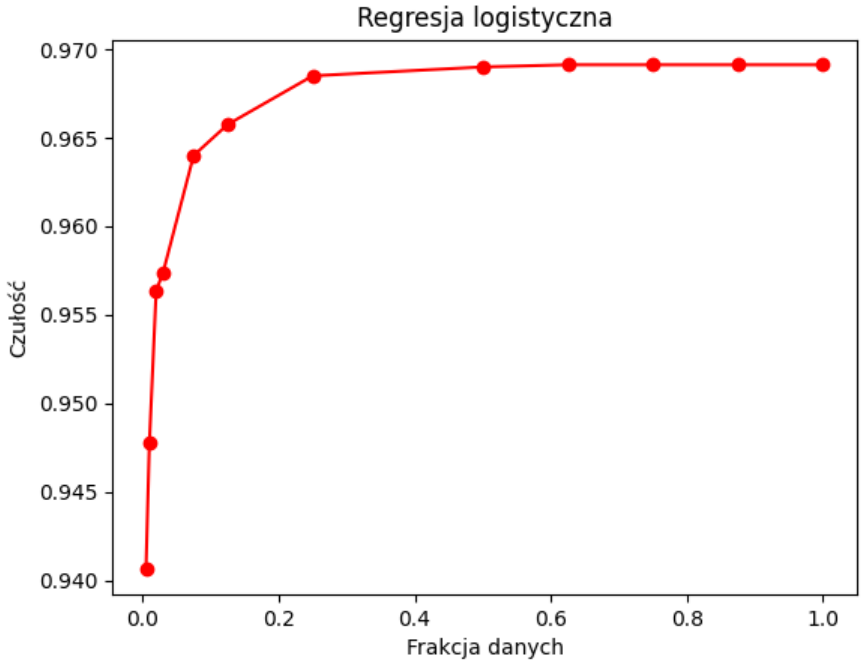
$$x^{(i)} \rightarrow \frac{x^{(i)} - \overline{x}}{s(x)}$$

Wykres krzywej uczenia w tym przypadku jest totalnie niespodziewany. Już dla frakcji danych 0.005 model uzyskuje błąd jedynie `0.051,` co jest bardzo niespodziewanym wynikiem. Sam kształt wykresu wygląda podobnie jak dla wszystkich wcześniejszych przypadków - tzn. największy spadek na początkowej frakcji 0.125, ale w związku ze świetnym początkowym wynikiem różnice pomiędzy największą i najmniejszą wartością są mniejsze niż 0.03. Najmniejszy błąd jest osiągany we frakcji 1.0 i wynosi `0.0284` , co jest jednak gorszym wynikiem niż w NKB pomimo znacznie lepszego startu niż wtedy.

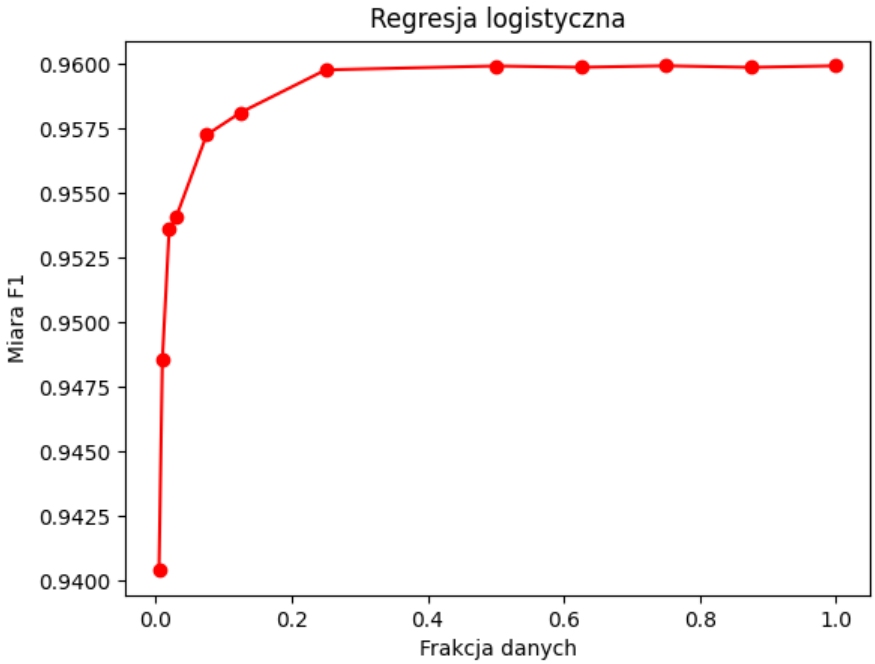


Precyzja i czułość z uwagi na niski błąd od samego początku wynoszą praktycznie tyle samo niezależnie od rozmiaru frakcji danych. Przykładowo różnice dla precyzji są mniejsze niż 0.001 co przekłada się na nieregularny wygląd wykresu. W przypadku czułości wartości zmieniają się o nieco większe wartości, ale cały czas są w okolicy 0.01. Największa precyzja to w przybliżeniu `0.951,` a czułość `0.9626,` co jest najlepszym wynikiem od czasu NKB.





Wykres miary F1 wynosi natomiast teraz



Podsumowując nieoczekiwanie naiwny klasyfikator bayesowski okazał się lepszy niż regresja logistyczna - miał mniejszy błąd, choć oczywiście różnica była niewielka (mniejsza niż 0.01), a także miał największą czułość spośród wszystkich rozważanych modeli. Niespodziewanie również stosując regresją logistyczną ze standaryzacją otrzymałem bardzo niski błąd wynoszący `0.051` już dla frakcji danych 0.005.