**Projekt zaliczeniowy**

**Techniki eksploracji danych**

**Zakres projektu**

Projekt miał na celu rozwiązanie problemów klasyfikacji i regresji przy pomocy własnych implementacji oraz funkcji z bibliotek R:

1. regresja – sieci neuronowe, drzewa decyzyjne, k-najbliższych sąsiadów
2. klasyfikacja binarna – sieci neuronowe, drzewa decyzyjne, k-najbliższych sąsiadów
3. klasyfikacja wieloklasowa – sieci neuronowe, drzewa decyzyjne, k-najbliższych sąsiadów

Zbadany też został wpływ hiper-parametrów na możliwości predykcyjne modeli oraz porównano wyniki z modelami uczonymi przy pomocy funkcji ‘train’ z biblioteki Caret. Funkcja ta pozwala na łatwą implementacji kroswalidacji dla modeli z bibliotek takich jak ‘nnet’ – Sieci Neuronowe, ‘rpart’ – Drzewa Decyzyjne czy ‘caret’ - KNN.

Algorytmy zbudowane własnoręcznie uwzględniały:

* sieci neuronowe – uogólnienie dla dowolnej liczby warstw i dowolnej liczby neuronów w każdej warstwie
* drzewa decyzyjne – możliwość przycięcia drzewa wykorzystując parametr ‘overfit’ i wartość ‘cf ‘
* knn – zaimplementowane zostało liczenie odległości Euklidesowej, Czebyszewa, Hamminga i Gower’a oraz wykonano zrównoleglenie predykcji dla każdego wariantu

Aby uśrednione statystyki poszczególnych modeli można było uznać za prawidłowe, wykonano w tym celu kroswalidację 10 razy (kFold = 10). Taka kroswalidacja została wykonana w przypadku algorytmów z własną implementacją oraz jako parametr kontrolny dla funkcji ‘train’.

Dane wykorzystane do projektu

1. Regresja:

https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

* Liczba obserwacji: 209
* Liczba atrybutów: 9
* Zmienna celu: publikowana relatywna wydajność (kolumna nr 9)
* Kolumna nr 2 i 10 zostały odrzucone – nr 2 to oznaczenia modeli, a nr 10 to estymowana wydajność

1. Klasyfikacja binarna:

http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

* Liczba obserwacji: 683
* Liczba atrybutów: 11
* Zmienna celu: diagnoza (M = malignant, B = benign) (kolumna nr 11)
* W kolumnie nr 7 pojawiły się znaki „?”

1. Klasyfikacja wieloklasowa:

https://archive.ics.uci.edu/ml/datasets/Yeast

* Liczba obserwacji: 1484
* Liczba atrybutów: 8
* Zmienna celu: lokalizacja białka (kolumna nr 10)
* Kolumna nr 1 to dana z bazy danych – nie brana pod uwagę. Dane posiadały specyficzny separator

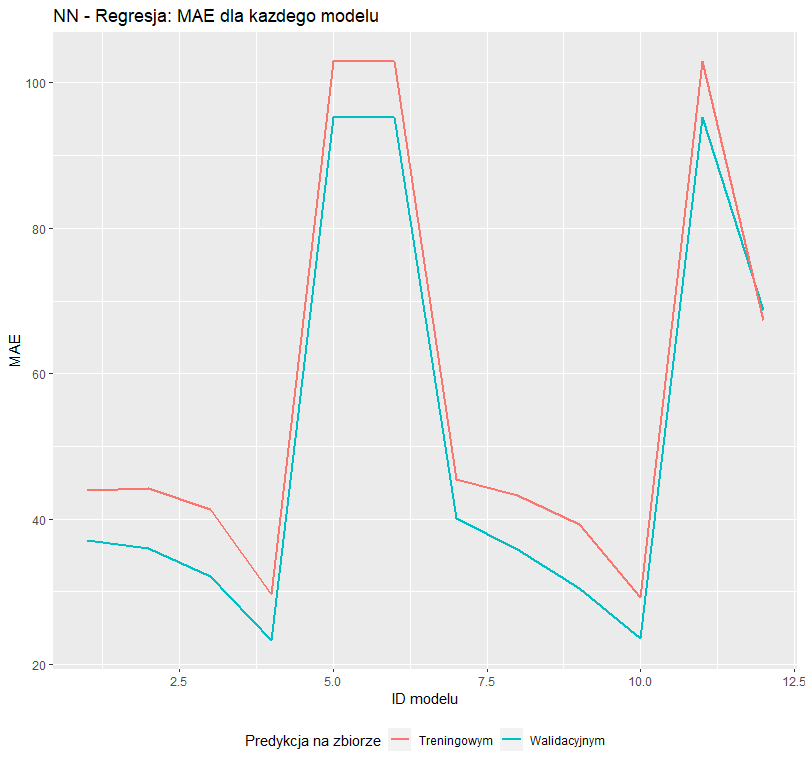
Wyniki dla poszczególnych algorytmów   
(wyniki dla walidacji)

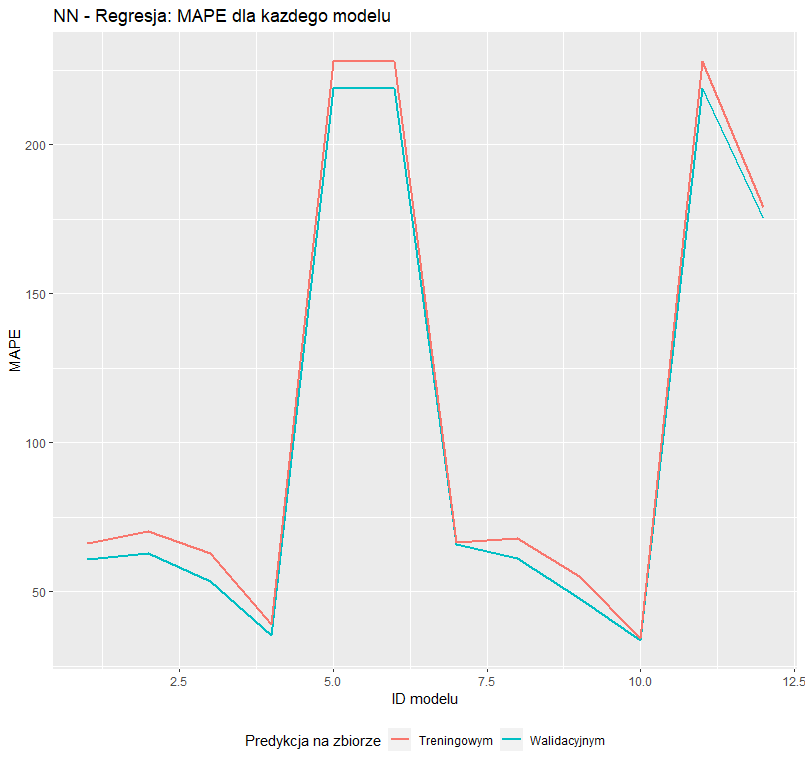
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Algorytm** | **Implementacja** | **Parametry** | **Ocena jakości dla zbioru Testowego** |
| **Regresja ( MAE )** | Sieci NN | Własna | h = (8,8), lr = 0.001,  *iter =* 80000 | 29.66 |
|  | Drzewa decyzyjne | Własna | depth = 6, minobs = 2 | 35.02 |
|  | KNN | Własna | k = 5 | 182.73 |
|  | Sieci NN | Biblioteka R | h = 8 | 0.061 |
|  | Drzewa decyzyjne | Biblioteka R | max depth = 3 | 51.41 |
|  | KNN | Biblioteka R | k = 3 | 32.75 |
| **Klasyfikacja Binarna (Jakość)** | Sieci NN | Własna | h = (8,8), lr = 0.01,  *iter =* 80000 | 0.9447 |
|  | Drzewa decyzyjne | Własna | depth = 9, minobs = 5, Gini, overfit = ‘prune’, cf = 0.02 | 0.9417 |
|  | KNN | Własna | k = 12 | 0.6388 |
|  | Sieci NN | Biblioteka R | h = 2 | 0.9634 |
|  | Drzewa decyzyjne | Biblioteka R | max depth = 3 | 0.9459 |
|  | KNN | Biblioteka R | k = 13 | 0.6762 |
| **Klasyfikacja Wieloklasowa (Jakość)** | Sieci NN | Własna | h = (8,8), lr = 0.1,  *iter =* 80000 | 0.5939 |
|  | Drzewa decyzyjne | Własna | depth = 9, minobs = 5, Gini, overfit = ‘prune’, cf = 0.022 | 0.5748 |
|  | KNN | Własna | k = 9 | 0.2782 |
|  | Sieci NN | Biblioteka R | h = 9 | 0.5837 |
|  | Drzewa decyzyjne | Biblioteka R | max depth = 8 | 0.5674 |
|  | KNN | Biblioteka R | k = 15 | 0.5929 |

Sieci Neuronowe (własna implementacja)  
**Regresja**

**h = list(c(8,8), c(8,8,8)), lr = c(0.0001, 0.001, 0.01), iter = c(20000, 80000)**

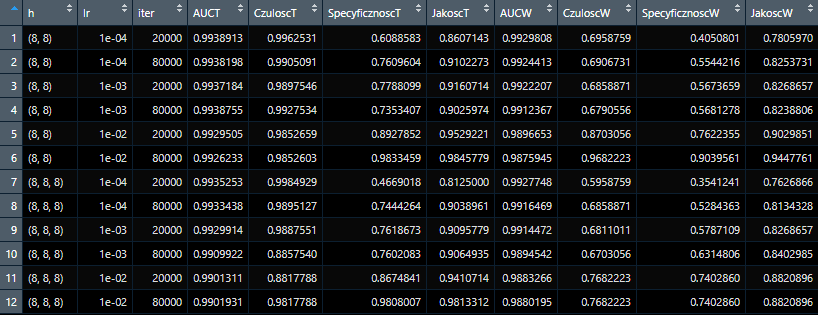


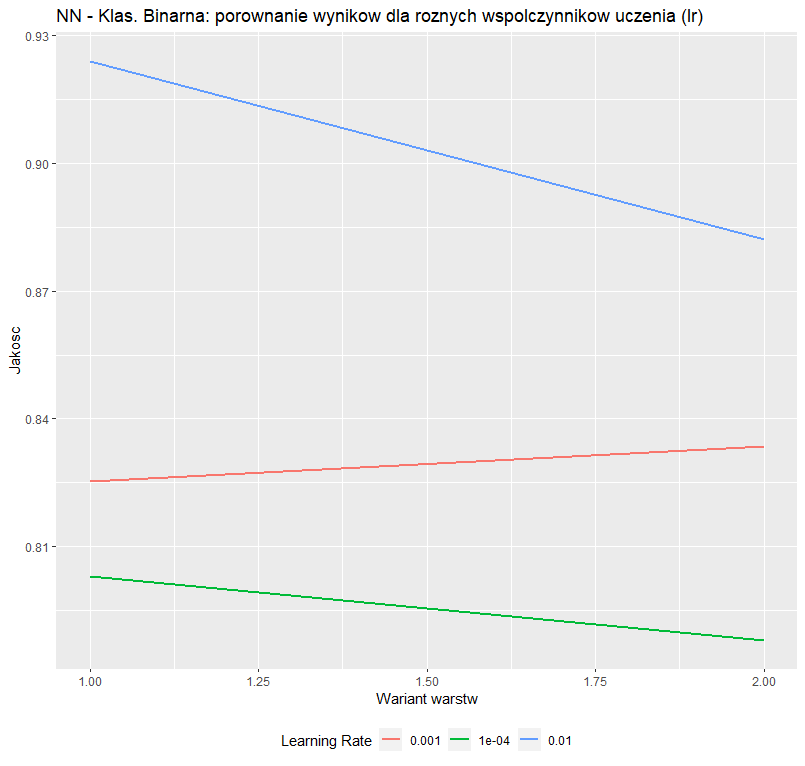




Sieci Neuronowe (własna implementacja)  
**Klasyfikacja Binarna**

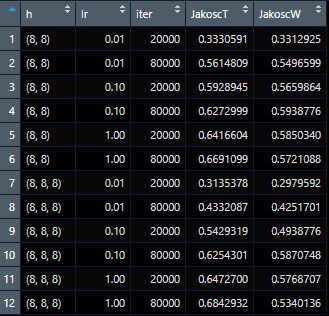
**h = list(c(8,8), c(8,8,8)), lr = c(0.0001, 0.001, 0.01), iter = c(20000, 80000)**

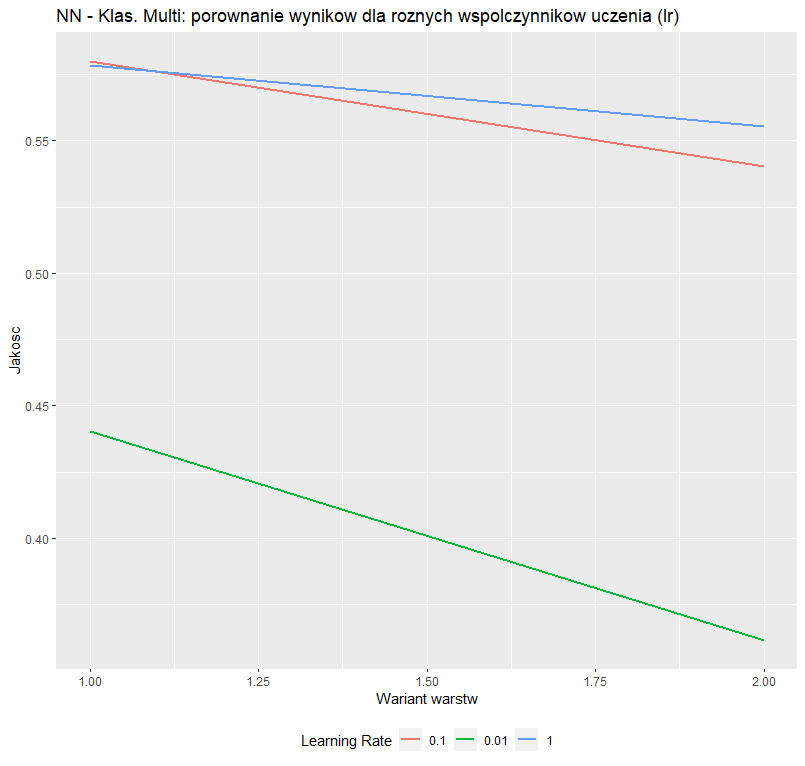




Sieci Neuronowe (własna implementacja)  
**Klasyfikacja Wieloklasowa**

**h = list(c(8,8), c(8,8,8)), lr = c(0.01, 0.1, 1.0), iter = c(20000, 80000)**

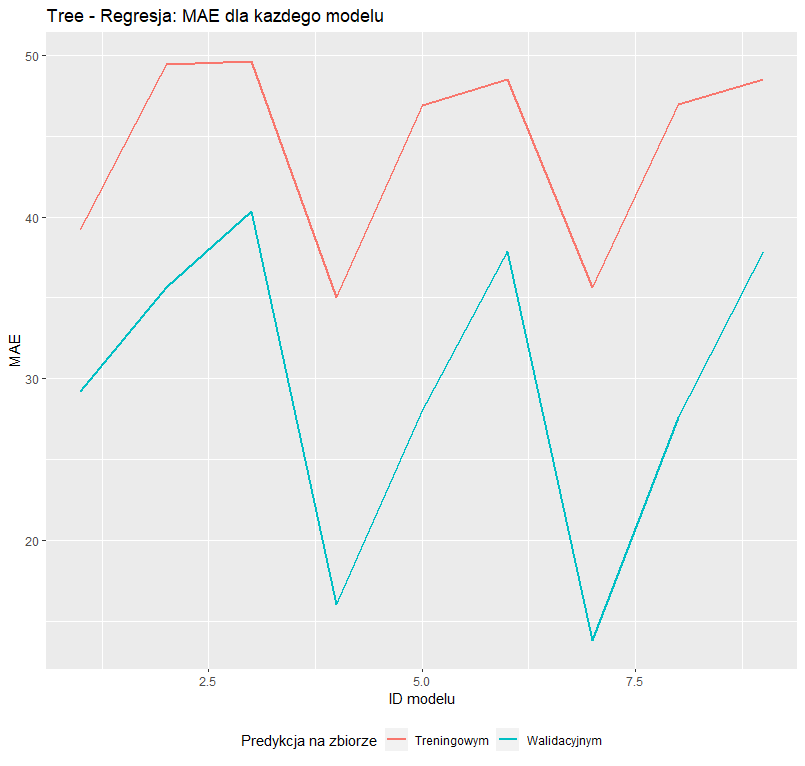


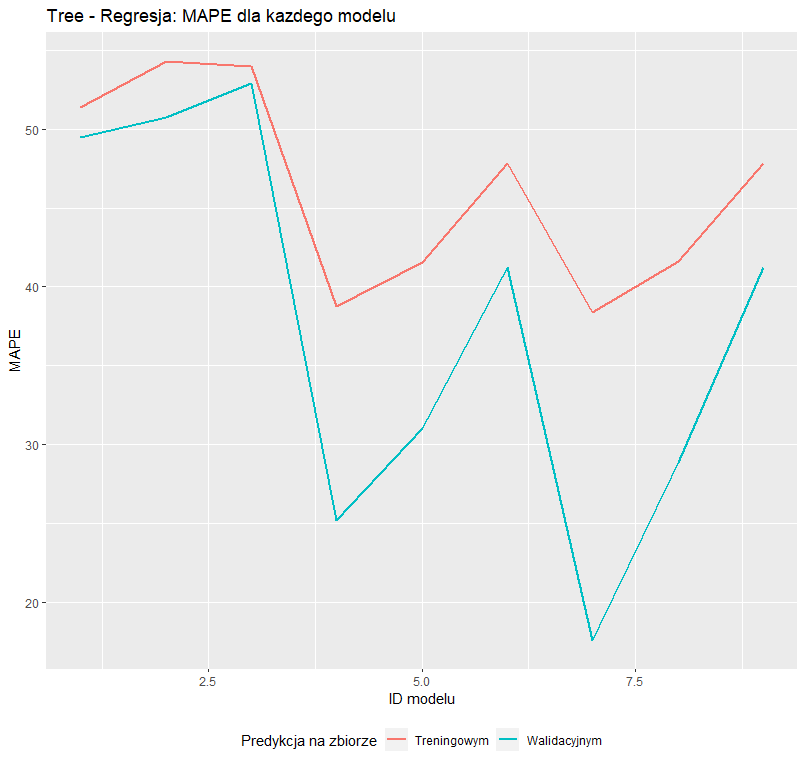


Drzewa Decyzyjne (własna implementacja)  
**Regresja**

**depth=c(3,6,9), minobs=c(2,5,10), type=c('SS'), overfit = c('none')**

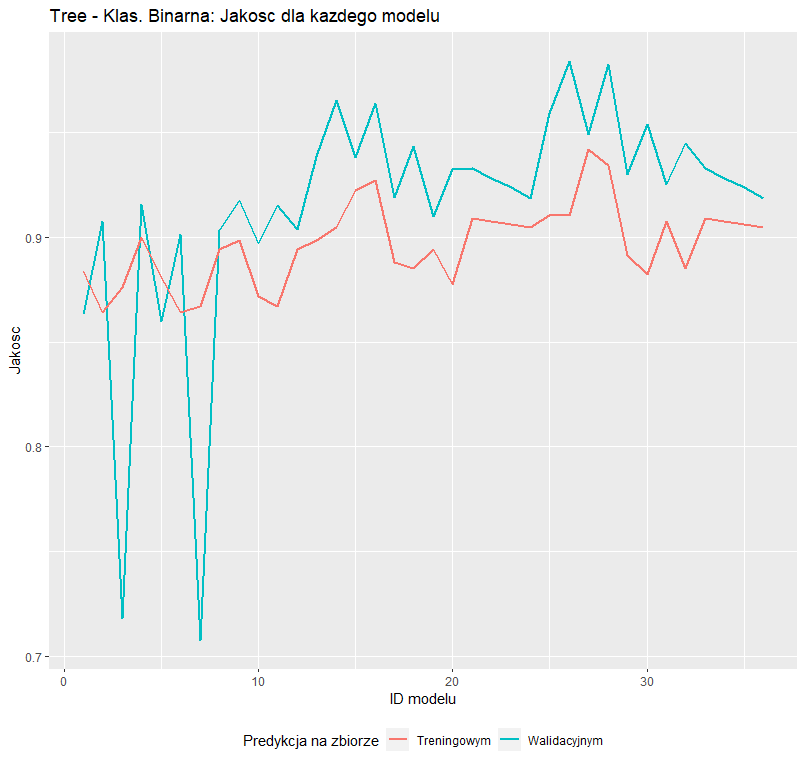
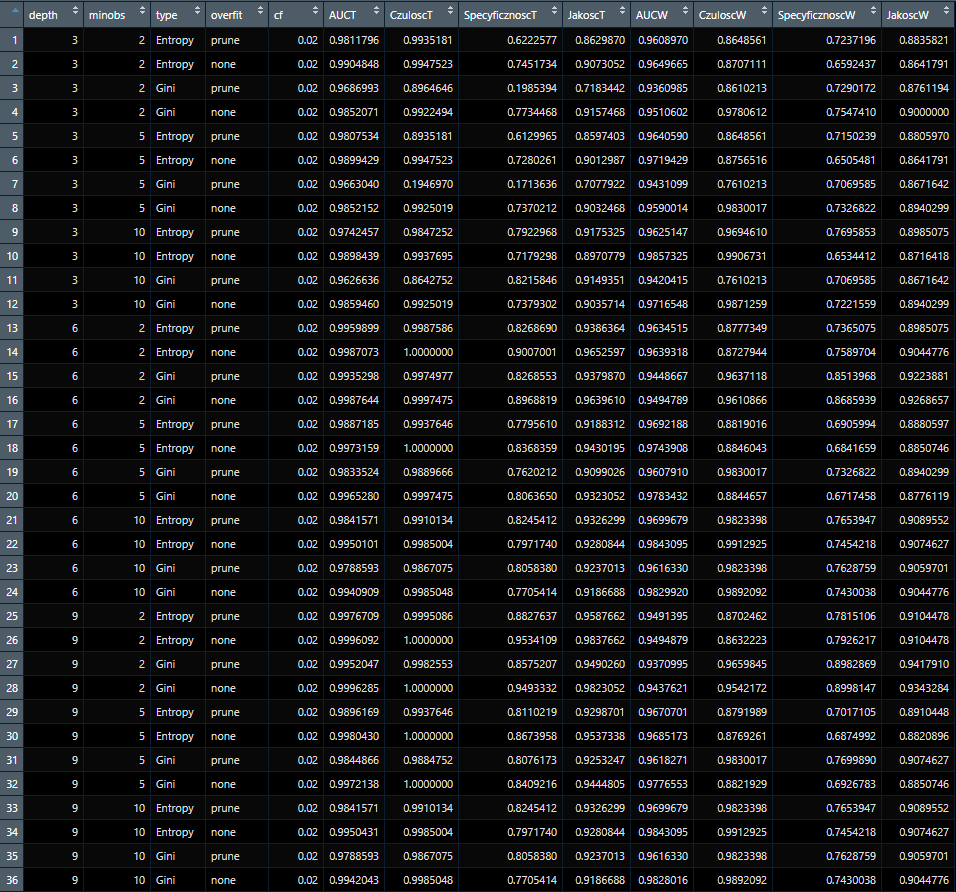






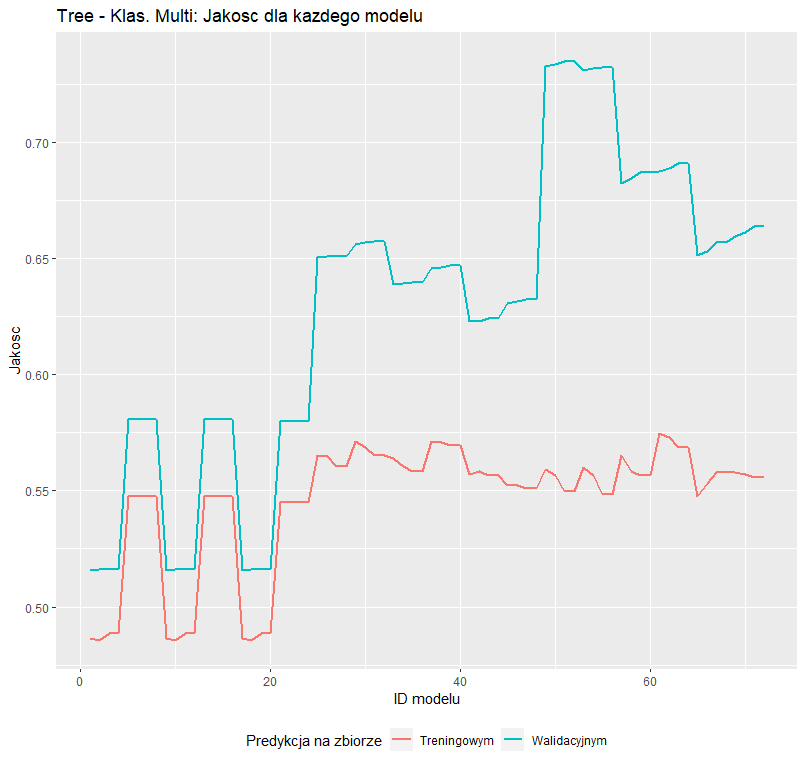
Drzewa Decyzyjne (własna implementacja)  
**Klasyfikacja Binarna**

**depth=c(3,6,9), minobs=c(2,5,10), type=c('Entropy', 'Gini'), overfit = c('none', 'prune'), cf=c(0.02)**

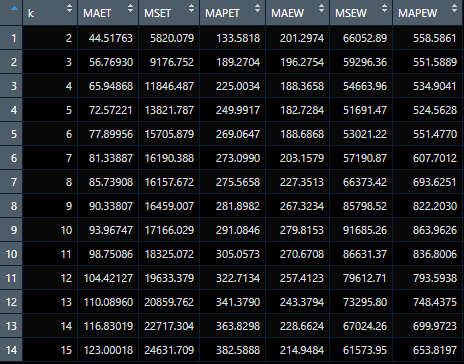


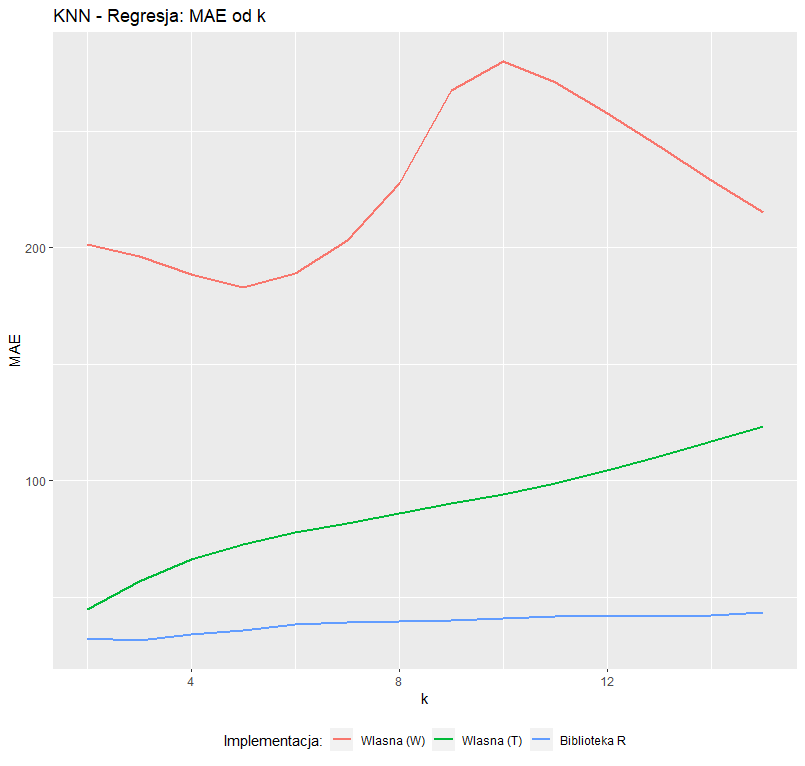
Drzewa Decyzyjne (własna implementacja)  
**Klasyfikacja Wieloklasowa   
depth=c(3,6,9), minobs=c(2,5,10), type=c('Entropy', 'Gini'), overfit = c('none', 'prune'), cf=c(0.05, 0.022)**

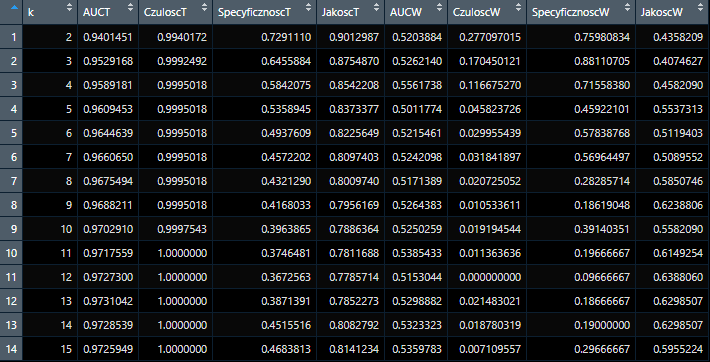


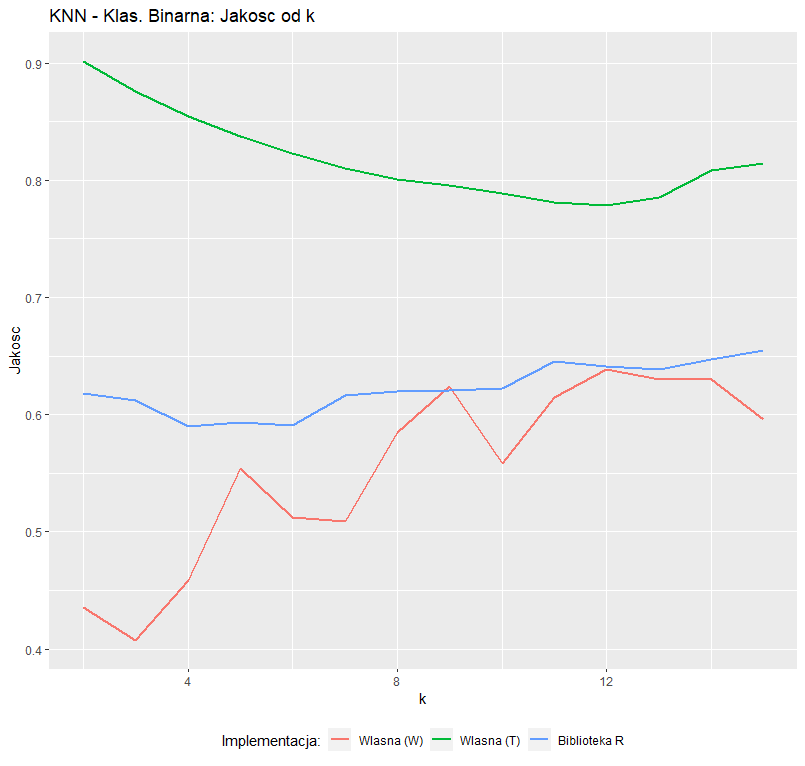
KNN (własna implementacja)  
**Regresja**





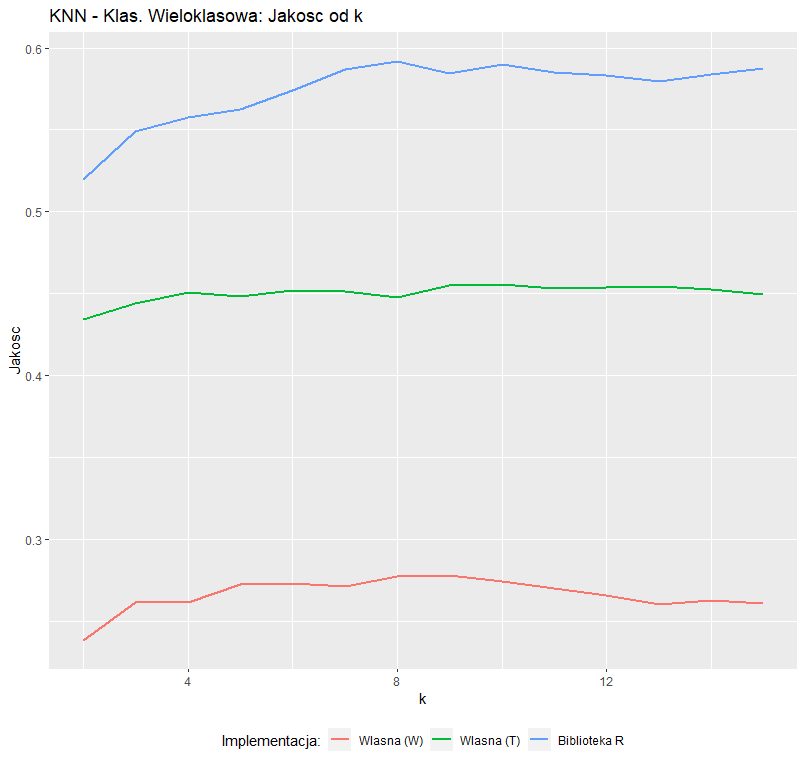
KNN (własna implementacja)  
**Klasyfikacja Binarna**





KNN (własna implementacja)  
**Klasyfikacja Wieloklasowa**





**Podsumowanie**

Implementacja własnych modeli Sieci Neuronowych, Drzew Decyzyjnych i KNN bazowała na ogólnej idei tych algorytmów. Zostały one uogólnione dla każdych danych – np. obliczenia różnych odległości dla KNN, oraz dla różnych parametrów – np. różna ilość warstw i liczba neuronów dla Sieci Neuronowej. Projekt ten ma na celu sprawdzenie owych implementacji dla różnych zbiorów i problemów. Aby sprawdzić czy wyniki są w sensownych granicach porównano wyniki z modelami z bibliotek R takimi jak ‘nnet’, ‘rpart’ i ‘caret’. W obu przypadkach wykonana została kroswalidacja aby statystyki wyników zostały odpowiednio uśrednione - pozwoliło to wybrać najlepsze modele.

W przypadku problemu regresji najlepszym modelem okazała się Sieć Neuronowa z biblioteki ‘nnet’. Osiągnęła ona najlepszy (czyli najmniejszy) wynik MAE = 0.061 dla h = 8. Z własnych implementacji najlepsza również była Sieć Neuronowa z dwiema warstwami, która osiągnęła wynik MAE = 29.66 . Pozostałe modele osiągały wyniki niedaleko własnej Sieci Neuronowej, aczkolwiek zaimplementowany własnoręcznie model KNN miał najlepsze MAE = 182.73 . Może to wynikać z liczenia odległości Gowera, ponieważ jedna z kolumn była faktorem. Problem klasyfikacji binarnej wyniki, dla własnych implementacji i z bibliotek, są bardzo bliskie. Różnica dokładności (jakości) pomiędzy poszczególnymi algorytmami jest niewielka. Najlepszym modelem okazała się Sieć Neuronowa z biblioteki R z wynikiem dokładności równym 0.9634. Własna implementacja nie była daleko, osiągnęła wynik 0.9447 . Przy tym problemie równie dobrze radziły sobie Drzewa Decyzyjne. W przypadku obu implementacji osiągnęły wynik ~ 0.94 . Tylko KNN, również w obu przypadkach osiągnęły niższe wyniki odpowiednio dla własnej implementacji - 0.6388, oraz dla modelu z biblioteki R – 0.6762 . Dane do problemu klasyfikacji wieloklasowej były największe ze wszystkich – niemalże 1500 wierszy. Jednakże tutaj ilość danych nie przełożyła się na dobre wyniki modeli. Tym razem najlepszym modelem ze wszystkich okazała się własna implementacja Sieci Neuronowej z wynikiem dokładności = 0.5939 . Modele z bibliotek osiągały wyniki bardzo podobne, w granicach 0.56 do 0.59, gdzie najlepszym algorytmem okazał się KNN z k = 15. Własne Drzewa Decyzyjne osiągnęły wynik 0.5748, natomiast KNN z k = 9 (najlepszy parametr) to dokładność na poziomie ~ 0.28 . Trzeba wziąć pod uwagę, że siatka parametrów była dość ograniczona i najlepsza kombinacja parametrów mogła zostać pominięta.