# Projekt z przedmiotu Techniki eksploracji danych

Pierwszym etapem projektu był wybór danych dla trzech problemów – regresji, klasyfikacji binarnej oraz klasyfikacji wieloklasowej.

Do problemu regresji został wybrany zbiór **Concrete slump**. Na podstawie danych zbadano od czego zależy rozpływ betonu. Zbiór posiada 7 zmiennych objaśniających oraz 3 zmienne objaśniane, jednak na potrzeby analizy została wybrana jedna zmienna objaśniana – **FLOW**, która przyjmuje wartości rzeczywiste. W zbiorze wystąpiły wartości liczbowe, które posiadały przecinki – zostały one zamienione w funkcji na kropki, aby można było operować na wartościach.

Do problemu klasyfikacji binarnej został wybrany zbiór **Fertility**. Dane z tego zbioru badają jak konkretne czynniki wpływają na płodność. Zbiór posiada 9 zmiennych objaśniających oraz 1 zmienną objaśnianą. Zmienna objaśniana **Diagnosis** jest diagnozą – przyjmuje wartości „N" – normalna oraz „O" – zmieniona. Na potrzeby analizy wartości zmiennej objaśnianej zostały przekonwertowane na wartości binarne: 0 – diagnoza normalna, 1 – diagnoza zmieniona.

Do problemu klasyfikacji wieloklasowej został wybrany zbiór **Abalone**. Dane zebrane w zbiorze mają za zadanie przewidywać wiek uchowca na podstawie pomiarów fizycznych. Zbiór posiada 8 zmiennych objaśniających oraz 1 zmienną objaśnianą – **Rings**, która przyjmuje wartości całkowite, jest to wiek uchowca. Na potrzeby analizy zmienna opisująca płeć zwierzęcia została przekonwertowana na wartości całkowite: 2 – M, 1 – F, 0 – I (niemowlę).

Proces wczytania danych oraz ich przekształcanie wykonuje się w pliku „Dane.R". W pliku „Funkcje.R" znajduje się implementacja wszystkich funkcji, natomiast plik „Glowny.R" generuje wyniki i wykresy.

Pierwszym etapem analizy po wczytaniu danych była kroswalidacja. W tym celu powstały dwie funkcje - CrossValidTune_knn oraz CrossValidTune_drzewa, odpowiednio dla metody najbliższych sąsiadów oraz drzew decyzyjnych. Pierwsza funkcja została wykorzystana dla problemu regresji oraz klasyfikacji binarnej, natomiast druga dla klasyfikacji binarnej. Wyniki zostaną przedstawione poniżej.

**Regresja – CrossValidTune_knn, seed = 888, kfold = 1:3, partune = 1:2**

```
[1] "Regresja"
  kfold partune    MAE_t    MSE_t    MAPE_t    MAE_v    MSE_v    MAPE_v
1     1       1 19.52464 505.6933 0.3971730 15.37353 527.1850 0.4920711
2     2       1 17.90435 427.5186 0.5070116 19.70588 530.6618 0.3933275
3     3       1 17.46667 445.8513 0.5329126 21.46176 620.9674 0.4265378
4     1       2 17.57536 384.9535 0.4142641 13.90588 377.2512 0.4004082
5     2       2 16.15435 360.5053 0.4704803 14.63235 316.6654 0.3152917
6     3       2 15.46667 353.8722 0.4749830 18.25588 434.6990 0.3917094
```

Parametr kfold służy do podziału zbioru na zbiór treningowy oraz walidacyjny. Natomiast parametr partune w przypadku funkcji CrossValidTune_knn jest liczbą najbliższych sąsiadów. W powyższym przykładzie zostały wyliczone błędy MAE, MSE oraz MAPE zarówno dla zbioru dotyczącego problemu regresji w podziale na zbiór treningowy oraz walidacyjny. Parametr kfold przyjmuje wartości z przedziału 1:3, natomiast parametr partune 1:2, stąd mamy 6 kolumn wynikowych.

Błędy prognoz w przypadku regresji zmieniają się po zmianie parametrów kfold oraz partune. Błędy zarówno dla zbioru treningowego jak i walidacyjnego są bardzo wysokie. Analizując dane nie możemy zobaczyć pewnej zależności co do wyników. Dla zbioru testowego najkorzystniejsze błędy MAE oraz MSE (najniższe) występują dla najwyższych parametrów kfold oraz partune. Jednak w przypadku MAPE najniższy błąd pojawia się gdy parametry są niskie. Natomiast w przypadku zbioru walidacyjnego najniższe wartości błędów występują, gdy parametr partune jest najwyższy, lecz bez uwzględnienia parametru kfold.

**Klasyfikacja binarna – CrossValidTune_knn, seed = 888, kfold = 1:3, partune = 1:2**

```
[1] "Klasyfikacja binarna"
  kfold partune     AUC_t Czulosc_t Specyficzosc_t  Jakosc_t     AUC_v Czulosc_v Specyficzosc_v  Jakosc_v
1     1       1 0.9035714 0.9500000      0.8571429 0.9402985 0.9000000 1.0000000      0.8000000 0.9696970
2     2       1 0.9185824 0.9482759      0.8888889 0.9402985 0.9666667 0.9333333      1.0000000 0.9393939
3     3       1 0.8251366 0.9836066      0.6666667 0.9552239 0.9814815 0.9629630      1.0000000 0.9696970
4     1       2 0.9571429 1.0000000      0.1428571 0.9104478 0.9785714 1.0000000      0.4000000 0.9090909
5     2       2 0.9664751 1.0000000      0.2222222 0.8955224 0.9222222 0.9333333      0.3333333 0.8787879
6     3       2 0.8592896        NA             NA        NA 0.9722222 1.0000000      0.5000000 0.9090909
```

Dla przypadku klasyfikacji binarnej zostały obliczone miary – AUC, czułość, specyficzność oraz jakość zarówno dla zbioru treningowego jak i walidacyjnego. Oceniają one poprawność modelu klasyfikacyjnego. Im są wyższe (zbliżają się do jedności) tym klasyfikacja jest poprawniejsza. W przypadku zbioru treningowego najwyższe wartości AUC oraz czułości występują dla parametrów kfold = 1,2 oraz partune = 2. Natomiast w przypadku specyficzności dla zbioru treningowego możemy zauważyć spadek wartości wraz ze zwiększeniem wartości parametrów. Dla zbioru walidacyjnego wartości AUC mają podobne wartości dla wszystkich parametrów – we wszystkich przypadkach przyjmują wartości 0.9 lub więcej. Podobnie w przypadku czułości, gdzie trzy wartości przyjmują wartość 1. Natomiast jeśli weźmiemy pod uwagę specyficzność, jej wartość jest wyższa dla parametru partune = 1. W przypadku jakości wyniki dla kombinacji parametrów są zbliżone i przekraczają 0.87.

**Klasyfikacja binarna – CrossValidTune_drzewa, seed = 888, kfold = 1:4, partune = 1:8**

```
[1] "Głębokość drzewa"
[1] 1
[1] "Kfold"
[1] 2
        levelName Count                                  Prob Leaf
1 Root              100                             0.88, 0.12
2  ¦--Age <= 0.64    54  0.796296296296296, 0.203703703703704    *
3  °--Age >  0.64    46 0.978260869565217, 0.0217391304347826    *
[1] "Głębokość drzewa"
[1] 2
[1] "Kfold"
[1] 2
               levelName Count                                      Prob Leaf
1 Root                       100                                 0.88, 0.12
2  ¦--Age <= 0.64             54  0.796296296296296, 0.203703703703704    *
3  °--Age >  0.64             46 0.978260869565217, 0.0217391304347826
4       ¦--Sitting <= 0.56    14 0.928571428571429, 0.0714285714285714    *
5       °--Sitting >  0.56    32                                       1    *
```

W przypadku funkcji CrossValidTune_drzewa, parametr partune dotyczy głębokości drzewa (depth). Powyżej znajduje się przykład drzewa decyzyjnego dla klasyfikacji binarnej – odpowiednio dla

głębokości 1 oraz 2. Do budowy drzewa zostały wzięte zmienne Age oraz Sitting, zmienną objaśnianą jest Diagnosis.

**Oceny dla klasyfikacji binarnej, seed = 888, kfold = 1:4, zbiór treningowy**

```
  kfold   Entropy      Gini       SS
1     1 0.4897790 0.8094222 394621.2
2     2 0.6014354 0.7496889 190579.4
3     3 0.4897790 0.8094222 394621.2
4     4 0.5293609 0.7888000 302604.6
```

Powyżej zostały przedstawione wyniki oceny dla problemu klasyfikacji binarnej – Entropy, Gini, SS. Miary zostały obliczone dla zbioru treningowego – dla różnych parametrów kfold (1:4). Parametry te im mają niższe wartości tym lepiej dla modelu. W przypadku entropii dla kfold = 1 oraz 3 wartość jest najniższa. Natomiast wskaźnik Giniego jest najniższy dla kfold = 2 oraz 4. Miara SS jest bardzo wysoka dla wszystkich badanych wartości kfold, jednak najniższą wartość przyjmuje dla kfold = 2.

**Klasyfikacja binarna – modelOcena**

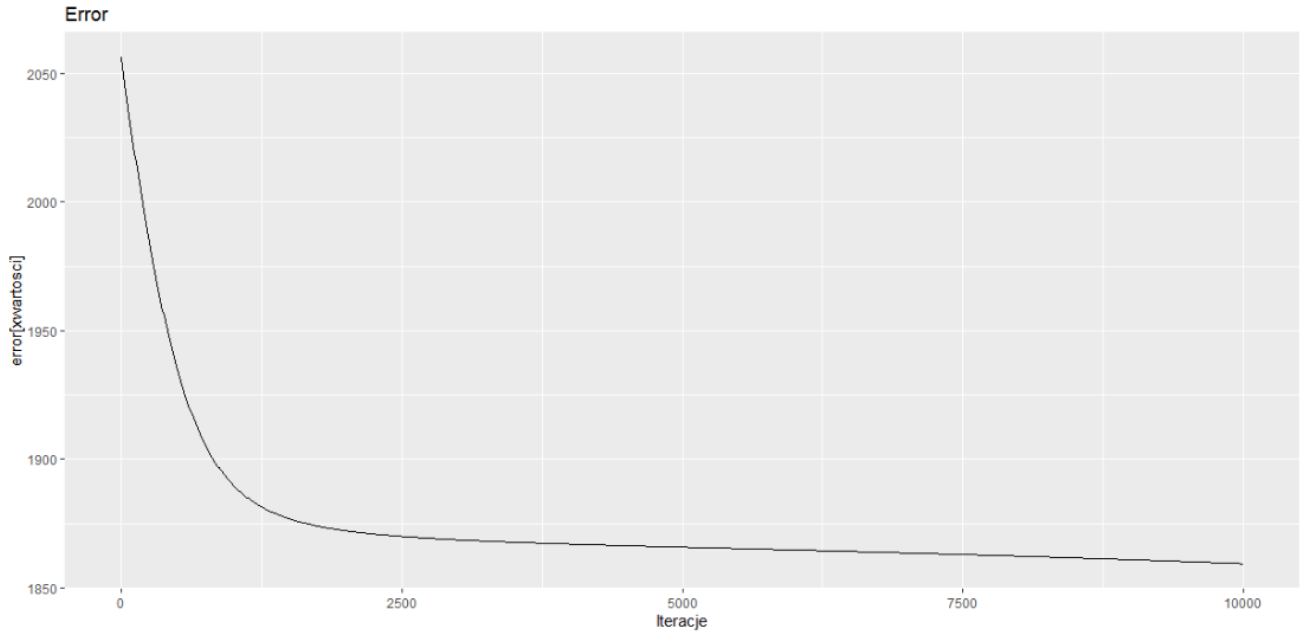
```
[[1]]
     y_hat
y_tar  0  1
    0 87  1
    1 10  2

[[2]]
    AUC funct       Czulosc Specyficznosc        Jakosc
    0.9086174     0.9886364     0.1666667     0.8900000
```

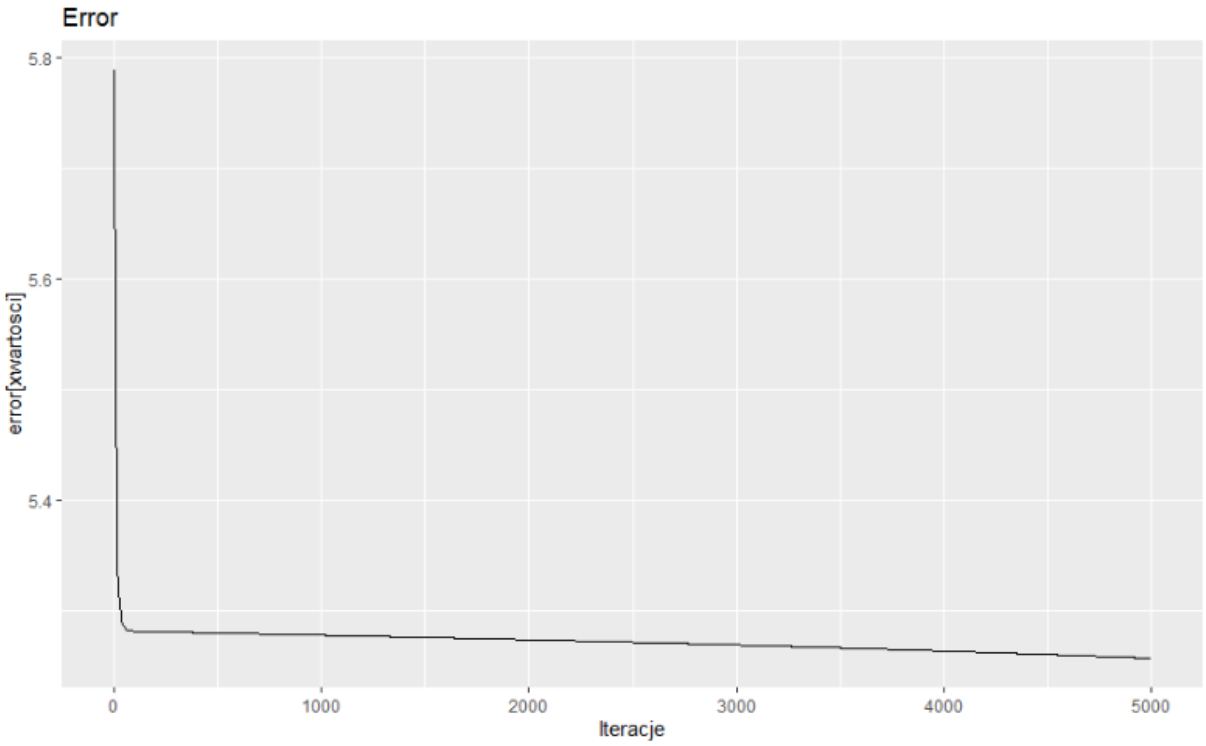
Powyżej zostały przedstawione wyniki funkcji modelOcena dla klasyfikacji binarnej – tym razem dla całego zbioru. AUC oraz czułość mają wysokie wyniki, bliskie jedności. Jakość również jest dosyć dobra – wynosi ok 0.9. Natomiast wartość specyficzności jest niska, wynosi poniżej 0.2.

Sieci neuronowe

**10000 iteracji – klasyfikacja wieloklasowa**




**5000 iteracji – klasyfikacja binarna**



**Metoda wektorów nośnych - regresja**