Porównanie klasyfikatorów na wybranej bazie danych

Dominika Pienczyn

2020

1 Baza danych

Baza danych z której skorzystałam w projekcie to *Red Wine Quality*. Baza danych zawiera kolumny:

* stała kwasowość
* kwasowość lotna
* kwas cytrynowy
* cukier resztkowy
* chlorki
* wolny dwutlenek siarki
* całkowity dwutlenek siarki
* gęstość
* pH
* siarczany
* alkohol
* jakość

```
fixed acidity          float64
volatile acidity       float64
citric acid            float64
residual sugar         float64
chlorides              float64
free sulfur dioxide    float64
total sulfur dioxide   float64
density                float64
pH                     float64
sulphates              float64
alcohol                float64
quality                  int64
dtype: object
```

Rysunek 1: Typ danych znajdujących się w bazie.

Klasyfikacji dokonałam na kolumnie *quality*. Dane znajdujące się w tej kolumnie są liczbami całkowitymi.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

Rysunek 2: Min, max, średnia i częstość występowania poszczególnych odpowiedzi.

```
fixed acidity           False
volatile acidity        False
citric acid             False
residual sugar          False
chlorides               False
free sulfur dioxide     False
total sulfur dioxide    False
density                 False
pH                      False
sulphates               False
alcohol                 False
quality                 False
dtype: bool
```

Rysunek 3: Baza w żaden sposób nie została zmodyfikowana, wszystkie dane były kompletne..

2 Porównanie poznanych klasyfikatorów

2.1 Zestaw testowy i treningowy

```
print ('Zestaw treningowy:', X_train.shape,  y_train.shape)
print ('Zestaw testowy:', X_test.shape,  y_test.shape)
```

```
Zestaw treningowy: (1279, 11) (1279,)
Zestaw testowy: (320, 11) (320,)
```

3 Testowanie klasyfikatorów

3.1 k Najbliższych sąsiadów (dla wybranego k)

k - algorytm najbliższych sąsiadów *(ang. k - nearest neighbours, k-NN)*:

* prosty klasyfikator (ściślej: algorytm regresji nieparametrycznej używany w statystyce do prognozowania wartości pewnej zmiennej losowej)
* klasyfikacja nowych przypadków jest realizowana na bieżąco, tj. gdy pojawia się potrzeba klasyfikacji nowego przypadku

3.1.1 k Najbliższych sąsiadów (dla k = 6)

```
Wyniki dla k równego 6

Dokładność zestawu treningowego: 63.72 %

Dokładność zestawu testowego: 48.12 %

Macierz błędu dla k=6
 [[ 0  0  0  3  1  0]
 [ 0  1  5  2  0  0]
 [ 0  1 97 33  4  0]
 [ 0  0 68 50  9  1]
 [ 0  1 17 20  6  0]
 [ 0  0  1  0  0  0]]
```

Rysunek 4: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla k=6
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         4
           4       0.33      0.12      0.18         8
           5       0.52      0.72      0.60       135
           6       0.46      0.39      0.42       128
           7       0.30      0.14      0.19        44
           8       0.00      0.00      0.00         1

    accuracy                           0.48       320
   macro avg       0.27      0.23      0.23       320
weighted avg       0.45      0.48      0.45       320
```

Rysunek 5: Raport klasyfikacyjny dla k równego 6.

3.1.2 k Najbliższych sąsiadów (dla k = 5)

```
Wyniki dla k równego 5


Dokładność zestawu treningowego: 66.54 %

Dokładność zestawu testowego: 46.88 %

Macierz błędu dla k=5
 [[ 0  0  0  3  1  0]
 [ 0  0  5  3  0  0]
 [ 0  2 88 42  3  0]
 [ 0  1 63 56  7  1]
 [ 0  1 16 20  6  1]
 [ 0  0  1  0  0  0]]
```

Rysunek 6: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla k=5
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         4
           4       0.00      0.00      0.00         8
           5       0.51      0.65      0.57       135
           6       0.45      0.44      0.44       128
           7       0.35      0.14      0.20        44
           8       0.00      0.00      0.00         1

    accuracy                           0.47       320
   macro avg       0.22      0.20      0.20       320
weighted avg       0.44      0.47      0.45       320
```

Rysunek 7: Raport klasyfikacyjny dla k równego 5.

3.1.3 k Najbliższych sąsiadów (dla k = 2)

```
Wyniki dla k równego 2

Dokładność zestawu treningowego: 79.75 %

Dokładność zestawu testowego: 49.06 %

Macierz błędu dla k=2
 [[  0   3   0   1   0   0]
 [  0   1   5   2   0   0]
 [  0   5 103  26   1   0]
 [  1   9  67  43   8   0]
 [  0   4  12  18  10   0]
 [  0   0   1   0   0   0]]
```

Rysunek 8: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla k=2
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         4
           4       0.05      0.12      0.07         8
           5       0.55      0.76      0.64       135
           6       0.48      0.34      0.39       128
           7       0.53      0.23      0.32        44
           8       0.00      0.00      0.00         1

    accuracy                           0.49       320
   macro avg       0.27      0.24      0.24       320
weighted avg       0.50      0.49      0.47       320
```

Rysunek 9: Raport klasyfikacyjny dla k równego 2.

3.1.4 k Najbliższych sąsiadów (dla k = 1)

Rysunek 10: Macierz błędu, dokładność zestawu testowego i treningowego.

Rysunek 11: Raport klasyfikacyjny dla k równego 1.

3.2 Naive Bayes

Naiwny klasyfikator bayesowski *(ang. Naive Bayes)* - prosty klasyfikator probabilistyczny. Naiwne klasyfikatory bayesowskie są oparte na założeniu o wzajemnej niezależności predyktorów (zmiennych niezależnych). Często nie mają one żadnego związku z rzeczywistością i właśnie z tego powodu nazywa się je naiwnymi. Bardziej opisowe jest określenie – „model cech niezależnych". Ponadto model prawdopodobieństwa można wyprowadzić korzystając z twierdzenia Bayesa.

W zależności od rodzaju dokładności modelu prawdopodobieństwa, naiwne klasyfikatory bayesowskie można skutecznie „uczyć" w trybie uczenia z nadzorem. W wielu praktycznych aplikacjach, estymacja parametru dla naiwnych modeli Bayesa używa metody maksymalnego prawdopodobieństwa a posteriori; inaczej mówiąc, można pracować z naiwnym modelem Bayesa bez wierzenia w twierdzenie Bayesa albo używania jakichś metod Bayesa.

```
Naive Bayes

Dokładność zestawu testowego: 55.94 %

Macierz błędu dla Naive Bayes
 [[ 0  1  1  0  0  0]
 [ 0  2  1  3  0  0]
 [ 0  8 84 44  3  0]
 [ 0  5 33 70 18  2]
 [ 0  0  4 15 23  0]
 [ 0  0  0  1  2  0]]
```

Rysunek 12: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla Naive Bayes
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.12      0.33      0.18         6
           5       0.68      0.60      0.64       139
           6       0.53      0.55      0.54       128
           7       0.50      0.55      0.52        42
           8       0.00      0.00      0.00         3

    accuracy                           0.56       320
   macro avg       0.31      0.34      0.31       320
weighted avg       0.58      0.56      0.57       320
```

Rysunek 13: Raport klasyfikacyjny dla Naive Bayes.

3.3 Drzewa decyzyjne

Drzewa decyzyjne *(ang. Decision Tree)* - drzewa decyzyjne znajdują praktyczne zastosowanie w różnego rodzaju problemach decyzyjnych, szczególnie takich gdzie występuje dużo rozgałęziających się wariantów a także w warunkach ryzyka. Wiele algorytmów uczenia się wykorzystuje drzewa decyzyjne do reprezentacji hipotez. Zgodnie z ogólnym celem uczenia się indukcyjnego, dążą one do uzyskania drzewa decyzyjnego klasyfikującego przykłady trenujące z niewielkim błędem próbki i o możliwie niewielkim rozmiarze, w nadziei, że takie drzewo będzie miało również niewielki błąd rzeczywisty. Drzewa decyzyjne znajdują szerokie zastosowanie w problemach związanych z klasyfikacją i predykcją pojęć typu:

* diagnostyka medyczna,
* przewidywanie wydajności,
* i wiele więcej

Proces klasyfikacji z wykorzystaniem drzew decyzyjnych jest efektywny obliczeniowo, wyznaczenie kategorii przykładu wymaga w najgorszym razie przetestowania raz wszystkich jego atrybutów.

```
Drzewa decyzyjne

Dokładność zestawu testowego: 60.00 %

Macierz błędu dla Drzewa decyzyjnych
 [[ 0  0  1  0  1  0]
 [ 0  0  2  4  0  0]
 [ 1  9 89 32  7  1]
 [ 1  1 29 83 12  2]
 [ 0  0  5 14 20  3]
 [ 0  0  1  0  2  0]]
```

Rysunek 14: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla Drzewa decyzyjnych
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00         6
           5       0.70      0.64      0.67       139
           6       0.62      0.65      0.64       128
           7       0.48      0.48      0.48        42
           8       0.00      0.00      0.00         3

    accuracy                           0.60       320
   macro avg       0.30      0.29      0.30       320
weighted avg       0.62      0.60      0.61       320
```

Rysunek 15: Raport klasyfikacyjny dla Drzew decyzyjnych.

3.4 Random Forest

Las losowy *(ang. Random forest)* - lub losowy las decyzyjny (Random decision forest) – metoda zespołowa uczenia maszynowego dla klasyfikacji, regresji i innych zadań, która polega na konstruowaniu wielu drzew decyzyjnych w czasie uczenia i generowaniu klasy, która jest dominantą klas (klasyfikacja) lub przewidywaną średnią (regresja) poszczególnych drzew. Losowe lasy decyzyjne poprawiają tendencję drzew decyzyjnych do nadmiernego dopasowywania się do zestawu treningowego. Pierwszy algorytm losowych lasów decyzyjnych został stworzony przez Tina Kam Ho przy użyciu metody losowej podprzestrzeni, która w formule Ho jest sposobem na implementację podejścia „dyskryminacji stochastycznej" do klasyfikacji zaproponowanej przez Eugene'a Kleinberga.

```
Random forest

Dokładność zestawu testowego: 0.68125

Macierz błędu dla Random forest
 [[  0   0   5   0   0   0]
 [  0   0   3   2   0   0]
 [  0   1 102  24   2   0]
 [  0   0  29  92   8   0]
 [  0   0   1  25  24   0]
 [  0   0   0   1   1   0]]
```

Rysunek 16: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla Random forest
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.00      0.00      0.00         5
           5       0.73      0.79      0.76       129
           6       0.64      0.71      0.67       129
           7       0.69      0.48      0.56        50
           8       0.00      0.00      0.00         2

    accuracy                           0.68       320
   macro avg       0.34      0.33      0.33       320
weighted avg       0.66      0.68      0.67       320
```

Rysunek 17: Raport klasyfikacyjny dla Random Forest.

3.5 Support Vector Machines

Maszyna wektorów nośnych, maszyna wektorów podpierających *(ang. Support Vector Machine, SVM)* - abstrakcyjny koncept maszyny, która działa jak klasyfikator, a której nauka ma na celu wyznaczenie hiperpłaszczyzny rozdzielającej z maksymalnym marginesem przykłady należące do dwóch klas. Często wykorzystywana niejawnie w procesie rozpoznawania obrazów. Maszyna wektorów nośnych, korzystająca z jądra RBF jest w stanie klasyfikować nierozdzielne liniowo klasy. W przypadku, wystąpienia więcej niż jednej klasy maszynę wektorów nośnych zazwyczaj uczy się metodą OvR.

```
Support Vector Machines

--------------------------------------------------------------------------------------------------------------------------------------------------------
Dokładność zestawu testowego: 54.69 %

Macierz błędu dla Support Vector Machines
 [[ 0  0  5  0  0  0]
 [ 0  0  4  1  0  0]
 [ 0  0 95 34  0  0]
 [ 0  0 49 80  0  0]
 [ 0  0  3 47  0  0]
 [ 0  0  0  2  0  0]]
```

Rysunek 18: Macierz błędu, dokładność zestawu testowego i treningowego.

```
Raport klasyfikacyjny dla Support Vector Machines
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.00      0.00      0.00         5
           5       0.61      0.74      0.67       129
           6       0.49      0.62      0.55       129
           7       0.00      0.00      0.00        50
           8       0.00      0.00      0.00         2

    accuracy                           0.55       320
   macro avg       0.18      0.23      0.20       320
weighted avg       0.44      0.55      0.49       320


--------------------------------------------------------------------------------------------------------------------------------------------------------
```

Rysunek 19: Raport klasyfikacyjny dla Support Vector Machines.

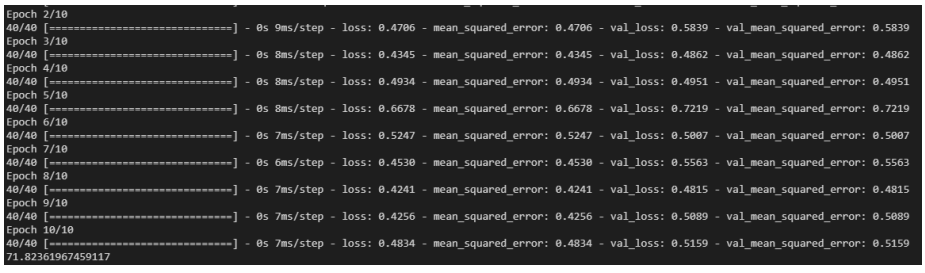
3.6 Sieci neuronowe

Sieć neuronowa *(ang. Neural Network)* - ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów przetwarzających, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa naturalnych neuronów, łączących je synaps, oraz układów nerwowych, w szczególności mózgu.

Czasem nazwą „sztuczne sieci neuronowe" określa się interdyscyplinarną dziedzinę wiedzy zajmującą się konstrukcją, trenowaniem i badaniem możliwości tego rodzaju sieci.

Rysunek 20: Sieci neuronowe

Rysunek 21: Ocena dla modelu losowego.



Rysunek 22: Ocena modelu sieci neuronowej.

4 Podsumowanie

Najlepsze wyniki otrzymałam za pomocą klasyfikatorów: *Random Forest, dokładność to* 71.56%. Na drugim miejscu znalazły się *Sieci neuronowe, dokładność to* 66.46%.

5 Reguły asocjacyjne

5.0.1 Co to są reguły asocjacyjne?

Reguły asocjacyjne przypominają reguły decyzyjne omawiane na poprzednim wykładzie. Tym razem jednak decyzja (prawa strona implikacji) nie jest z góry określona, tzn. nie wiemy, na którym atrybucie ma się opierać. Jest to przykład nauki bez nauczyciela (podobnie, jak w przypadku algorytmów grupowania): algorytm nie ma określonej z góry prawidłowej odpowiedzi, zamiast tego ma opisać wewnętrzne zależności między atrybutami.

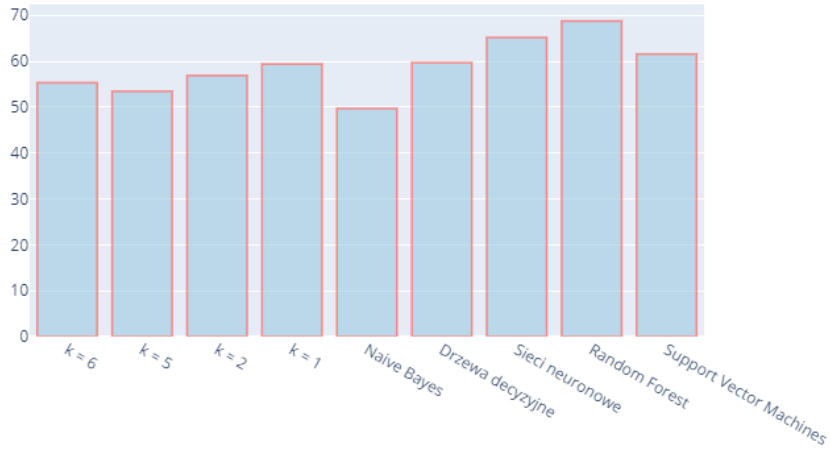
5.0.2 Czy szukanie reguły asocjacyjnych ma sens?

Wyszukiwanie reguł asocjacyjnych ma za zadanie zwiększyć wydajność systemów baz danych pod względem funkcjonalności umożliwiając zadawanie złożonych zapytań przykładowo w posta- ci: Znajdź wszystkie reguły, których konsekwencją jest sprzedaż produktu X. Znajomość reguł tego typu pozwoli podnieść sprzedaż produktu X.

5.0.3 Rodzaje reguł asosjacyjnych

* jednowymiarowe,
* wielowymiarowe,

6 Porównywanie klasyfikatorów na wykresie



Rysunek 23: Porównywanie klasyfikatorów na podstawie oceny modelu.