30.06.2023

Sztuczna Inteligencja i Inżynieria Wiedzy

Lab 4

Autor: Jakub Szwedowicz 243416

Prowadzący: dr inż. Piotr Syga

Zajęcia: czw 7:30

30.06.2023

Spis treści

1. eksploracja danych – przedstaw podstawowe dane statystyczne i uwagi dotyczące cech i etykiet zbioru danych. (10 punktów)

Z pliku glass.names można wyczytać kilka istotnych informacji.

Przede wszystkim poznajemy dane statystyczne, z których wynika, że w przetwarzanych danych nie występują braki.

```
8. Missing Attribute Values: None

Summary Statistics:
Attribute:   Min     Max      Mean     SD      Correlation with class
 2. RI:       1.5112  1.5339   1.5184  0.0030  -0.1642
 3. Na:      10.73   17.38    13.4079  0.8166   0.5030
 4. Mg:       0       4.49     2.6845  1.4424  -0.7447
 5. Al:       0.29    3.5      1.4449  0.4993   0.5988
 6. Si:      69.81   75.41    72.6509  0.7745   0.1515
 7. K:        0       6.21     0.4971  0.6522  -0.0100
 8. Ca:       5.43   16.19     8.9570  1.4232   0.0007
 9. Ba:       0       3.15     0.1750  0.4972   0.5751
10. Fe:       0       0.51     0.0570  0.0974  -0.1879
```

Powyższe potwierdza również prosta analiza przy użyciu pandas. Odczytać można, że dla każdej kolumny występuje 214 wartości „non-null”, gdzie wszystkich próbek jest właśnie 214 i dodatkowo zliczając „isna” wychodzi wynik 0

```
print(dataframe.info())
lack_of_data_name = 'N/D'
train_isna = pd.DataFrame(dataframe.isna().sum(), columns=[lack_of_data_name])
train_isna
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 214 entries, 0 to 213
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   RI      214 non-null    float64
 1   Na2O    214 non-null    float64
 2   MgO     214 non-null    float64
 3   Al2O3   214 non-null    float64
 4   SiO2    214 non-null    float64
 5   K2O     214 non-null    float64
 6   CaO     214 non-null    float64
 7   BaO     214 non-null    float64
 8   Fe2O3   214 non-null    float64
 9   Type    214 non-null    int64
dtypes: float64(9), int64(1)
memory usage: 16.8 KB
None
```

	N/D
RI	0
Na2O	0
MgO	0
Al2O3	0
SiO2	0
K2O	0
CaO	0
BaO	0
Fe2O3	0
Type	0

Z pliku dowiadujemy się również jakie pierwiastki występują oraz w jakich typach produktów szklanych występują:

```
7. Attribute Information:
   1. Id number: 1 to 214
   2. RI: refractive index
   3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as
                  are attributes 4-10)
   4. Mg: Magnesium
   5. Al: Aluminum
   6. Si: Silicon
   7. K: Potassium
   8. Ca: Calcium
   9. Ba: Barium
  10. Fe: Iron
  11. Type of glass: (class attribute)
      -- 1 building_windows_float_processed
      -- 2 building_windows_non_float_processed
      -- 3 vehicle_windows_float_processed
      -- 4 vehicle_windows_non_float_processed (none in this database)
      -- 5 containers
      -- 6 tableware
      -- 7 headlamps
```

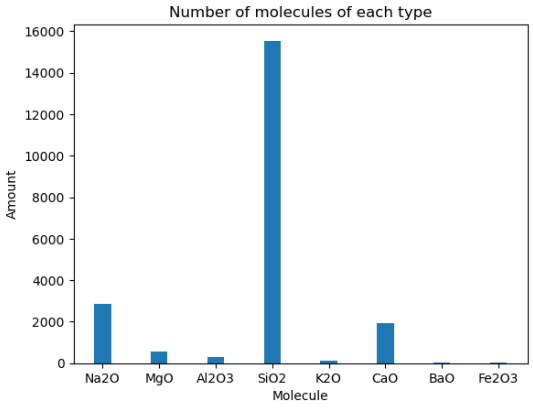
Poznajemy również cząsteczki, które będą tak naprawdę zliczane:

```
An original file donated by Vina Speihler

ID, N    -- numeric identifier of the instance
RI, N    -- refractive index
NA2O, N  -- Sodium oxide
MGO, N   -- magnesium oxide
AL2O3, N -- aluminum oxide
SIO2, N  -- silcon oxide
K2O,  N  -- potassium oxide
CAO, N   -- calcium oxide
BAO, N   -- barium oxide
FE2O3, N -- iron oxide
TYPE, N  -- An unknown, but must correspond to the types in the paper
CAMG, N  -- Unsure
```

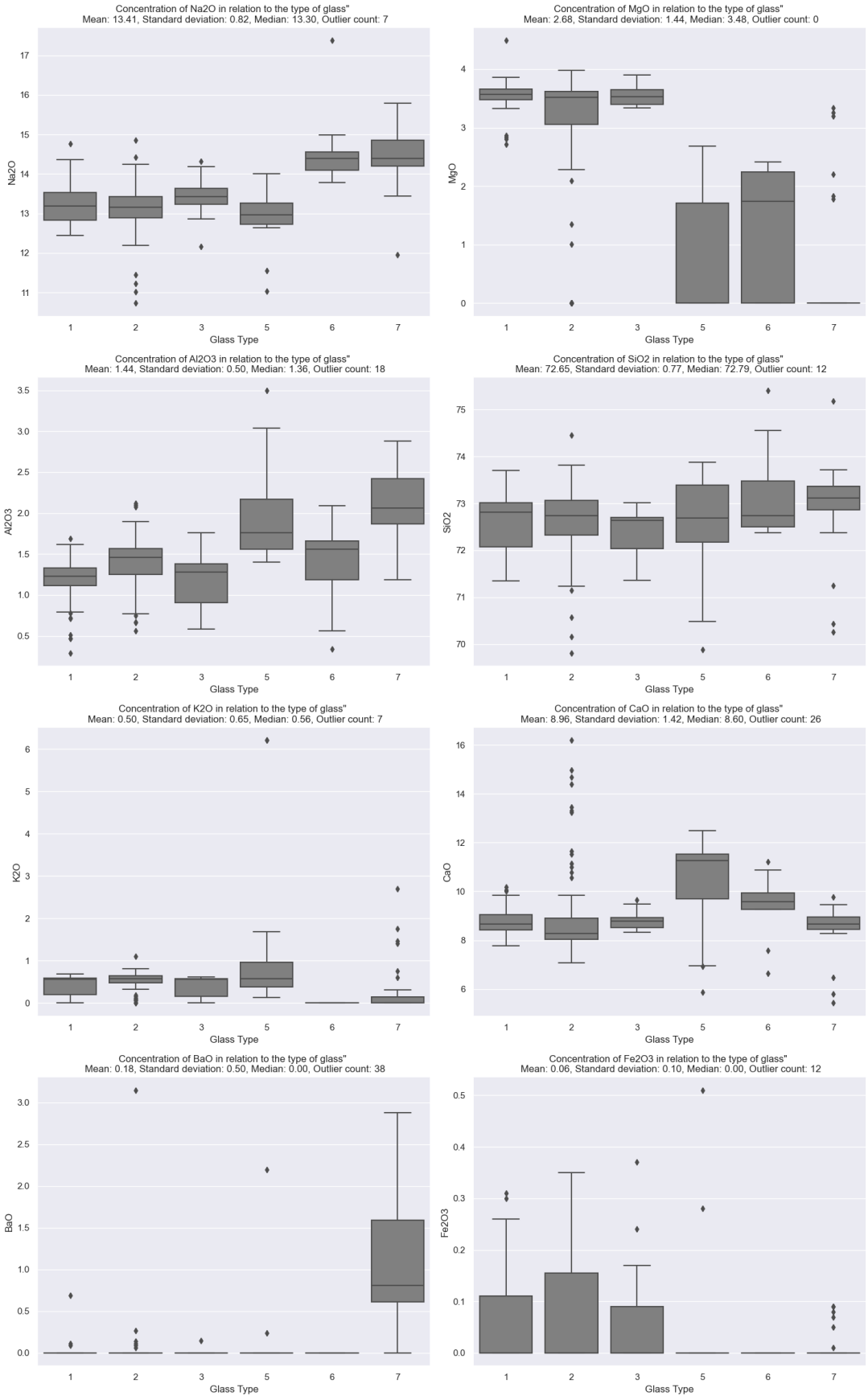
Wczytane z pliku dane można przedstawić za pomocą tabeli:

	Na2O	MgO	Al2O3	SiO2	K2O	CaO	BaO	Fe2O3
0	13.64	4.49	1.10	71.78	0.06	8.75	0.00	0.0
1	13.89	3.60	1.36	72.73	0.48	7.83	0.00	0.0
2	13.53	3.55	1.54	72.99	0.39	7.78	0.00	0.0
3	13.21	3.69	1.29	72.61	0.57	8.22	0.00	0.0
4	13.27	3.62	1.24	73.08	0.55	8.07	0.00	0.0
..	...	...	...	...	...	...	...	...
209	14.14	0.00	2.88	72.61	0.08	9.18	1.06	0.0
210	14.92	0.00	1.99	73.06	0.00	8.40	1.59	0.0
211	14.36	0.00	2.02	73.42	0.00	8.44	1.64	0.0
212	14.38	0.00	1.94	73.61	0.00	8.48	1.57	0.0
213	14.23	0.00	2.08	73.36	0.00	8.62	1.67	0.0

[214 rows x 8 columns]

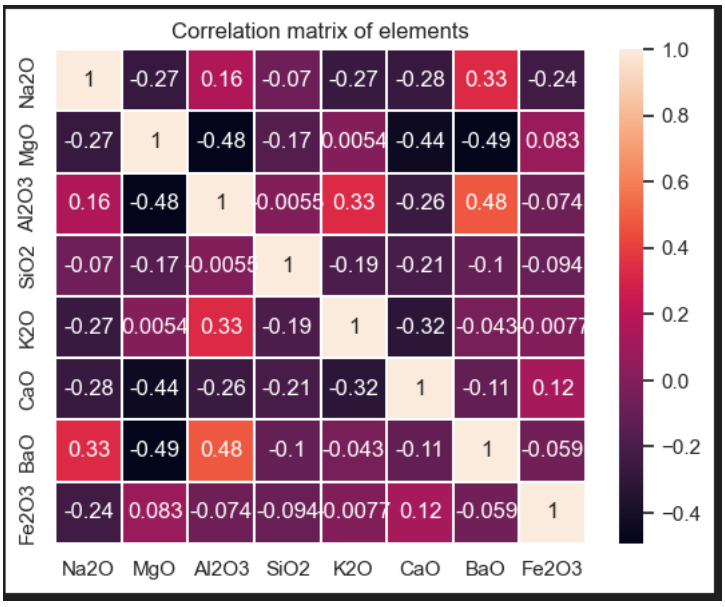


Ponadto przygotowane zostały wykresy pudełkowe przedstawiające cechy statystyczne poszczególnych cząsteczek. Można z nich wyczytać średnią, odchylenie standardowe, medianę oraz wartości odstające zdefiniowane jako te leżące poza przedziałem międzykwartalnym (IQR).



2. Przygotowanie danych - podziel dane na zestaw uczący i walidacyjny (alternatywnie użyj walidacji krzyżowej), zbadaj wpływ różnego typu przetworzenia danych na wyniki klasyfikacji (proponowane: normalizacja, standaryzacja, dyskretyzacja, selekcja cech, PCA) - czyli wykonaj porównanie wyników bez przetworzenia danych z rezultatami po ich przetworzeniu, wykorzystując co najmniej 2 metody różnego typu (osobno). (30 punktów) Bonus – usuń 5% wartości cech i przygotuj dane stosując metody radzenia sobie z brakującymi danymi. (5 punktów

Usunięto kolumny Id, type oraz Ri a następnie stworzono macierz korelacji przedstawiającą korelację występowania pomiędzy poszczególnymi cząsteczkami. Na podstawie rysunku można stwierdzić, że np. obecnośc BaO koreluje z występowaniem Al_2O_3.



Do podzielenia danych na zestawy uczące i walidacyjne napisana została klasa ModelsData, która będzie gromadzić:

- Modele (knn, SVC, itd.),
- Dane testowe wspólne dla modeli wartości funkcji: y_test, y _train
- Specyficzne dla każdego modelu zbiory argumentów funkcji X_train_scaled oraz X_test_scaled.
- Zestaw najlepszych scalerów dla każdego modelu, które dały najlepszy rezultat.

Przetestowano kombinacje: braku scalera, StandardScaler, ScandardScaler z PCA(i=1…8), PCA(i=1…8), MinMaxScaler(). Na tej podstawie uzyskano najlepsze kombinacje skalerów dla każdego modelu:

```
The best scaler for model DecisionTreeClassifier is [StandardScaler()] with accuracy 0.7906976744186046
The best scaler for model KNeighborsClassifier is [StandardScaler(), PCA(n_components=3)] with accuracy 0.813953488372093
The best scaler for model SVC is [StandardScaler()] with accuracy 0.7209302325581395
The best scaler for model LogisticRegression is [None] with accuracy 0.7209302325581395
The best scaler for model GaussianNB is [PCA(n_components=7)] with accuracy 0.6511627906976745
The best scaler for model RandomForestClassifier is [StandardScaler(), PCA(n_components=3)] with accuracy 0.8372093023255814
```

Gdzie w przypadku nie użycia scalerów, podstawowe modele (bez strojenia hiperapametrów) uzyskują niższe dokładności klasyfikacji danych:

```
The best scaler for model DecisionTreeClassifier is [None] with accuracy 0.7441860465116279
The best scaler for model KNeighborsClassifier is [None] with accuracy 0.6511627906976745
The best scaler for model SVC is [None] with accuracy 0.32558139534883723
The best scaler for model LogisticRegression is [None] with accuracy 0.7209302325581395
The best scaler for model GaussianNB is [None] with accuracy 0.5348837209302325
The best scaler for model RandomForestClassifier is [None] with accuracy 0.813953488372093
```

3. klasyfikacja – przetestuj klasyfikatory i zbadaj wpływ na wyniki: naiwny klasyfikator Bayesa oraz drzewo decyzjne używając przynajmniej 3 różnych zestawów hiperparametrów. (40 punktów) Bonus – Przetestuj (ze zrozumieniem!) bardziej zaawansowane algorytmy, takie jak Las losowy czy Klasyfikator wektorów nośnych (SVM, z ang. Support Vector Machines). (5 punktów)

Dla GNB użyto zestawu różnych wartości ”var_smoothing”. Hiperparametr ten sztucznie dodaje zdefiniowaną wartość do wariancji rozkładu. Zmienna ta zwiększa zatem wagę próbek bliżej średniej rozkładu. W zależności od charakteru badanego zjawiska – może to być porządeń gdy badane zmienne mają charakter rozkładu normalnego lub nie.

Dla drzewa decyzyjnego (które w sklearn jest binarne, gdyż jest przystosowane do danych numerycznych) modyfikowane są 4 parametry:

- criterion – kryterium którego algorytm używa do określenia w jaki sposób należy podzielić przetwarzane dane w każdym poziomie drzewa aby uzyskać najlepszą dokładność.
- max_depth – maksymalna wysokość drzewa. Istotnie wpływa na wymagania pamięciowe i obliczeniowe dla działania algorytmu gdyż liczba węzłów drzewa rośnie wykładniczo w stosunku do wysokości drzewa. Istotne również jest, że zbytnia głebokość drzewa może doprowadzić do zbytniego dopasowania modelu do danych (overfitting), co w rezultacie może bardzo pogorszyć klasyfikację danych spoza zestawu trenującego i testowego. Zbyt niska głębokość sprawi, że niewielkie różnice w próbkach będą powodować diametralnie różne wyniki klasyfikacji.
- min_samples_split, - zmienna ograniczająca od dołu liczbę wymaganych próbek w węźle aby dokonać podziału i rozrostu drzewa. Zbyt duża wartość może doprowadzić to uniewrażliwienia drzewa na małe zmiany wartości próbek, natomiast zbyt mała wręcz odwrotnie.
- min_samples_leaf – podobnie jak powyżej ale dotyczy liści.

```
Finding best models using non scaled data
Training model GaussianNB
Fitting 5 folds for each of 7 candidates, totalling 35 fits
Best score: 0.40369747899159664
Best parameters: {'var_smoothing': 1e-07}
Training model DecisionTreeClassifier
Fitting 5 folds for each of 232 candidates, totalling 1160 fits
Best score: 0.6781512605042017
Best parameters: {'criterion': 'entropy', 'max_depth': 17, 'min_samples_leaf': 2, 'min_samples_split': 2}
```

```
Finding best models using scaled data
Training model GaussianNB
Fitting 5 folds for each of 7 candidates, totalling 35 fits
Best score: 0.5144537815126051
Best parameters: {'var_smoothing': 1e-07}
Training model DecisionTreeClassifier
Fitting 5 folds for each of 232 candidates, totalling 1160 fits
Best score: 0.6376470588235295
Best parameters: {'criterion': 'entropy', 'max_depth': 14, 'min_samples_leaf': 2, 'min_samples_split': 4}
```

Wyniki dla różnych modeli (SVC, Regresji logistycznej, Losowego lasu klasyfikatorów, GNB oraz drzewa decyzyjnego) przedstawiają się następująco:

========== non Scaled ==========

Finding best models using non scaled data

Training model SVC

Best score: 0.6963025210084033

Best parameters: {'C': 256, 'class_weight': None, 'gamma': 'auto', 'kernel': 'rbf'}

Training model LogisticRegression

Best score: 0.5789915966386554

Best parameters: {'C': 0.3, 'dual': False, 'penalty': 'l2', 'solver': 'newton-cg', 'tol': 0.5}

Training model GaussianNB

Best score: 0.40369747899159664

Best parameters: {'var_smoothing': 1e-07}

Training model DecisionTreeClassifier

Best score: 0.6610084033613445

Best parameters: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 2}

Training model RandomForestClassifier

Best score: 0.7245977011494253

Best parameters: {'criterion': 'gini', 'max_depth': 90, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 3, 'n_jobs': -1}

========== Scaled ==========

Finding best models using scaled data

Training model SVC

Best score: 0.6611764705882354

Best parameters: {'C': 16, 'class_weight': None, 'gamma': 'scale', 'kernel': 'rbf'}

Training model LogisticRegression

Best score: 0.5966386554621849

Best parameters: {'C': 0.5, 'dual': False, 'penalty': 'l2', 'solver': 'sag', 'tol': 0.05}

Training model GaussianNB

Best score: 0.40369747899159664

Best parameters: {'var_smoothing': 5e-07}

Training model DecisionTreeClassifier

Best score: 0.6489075630252101

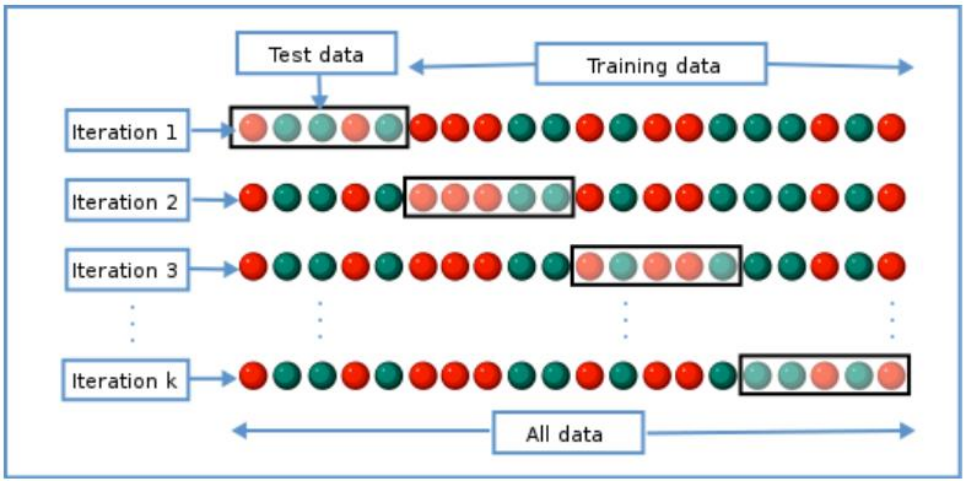
Best parameters: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 2}

Training model RandomForestClassifier

Best score: 0.7377011494252873

Best parameters: {'criterion': 'gini', 'max_depth': 90, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_jobs': -1}

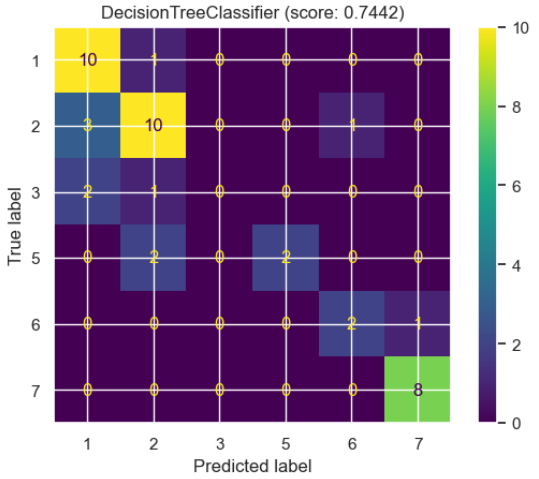
Same modele zostały wyznaczone przy użyciu GridSearchCV, który przeszukuje N-wymiarową przestrzeń hiperparametrów oraz dokonuje cross-validacji na zbiorze danych treningowych w celu określenia skuteczności modelu. Najpowszechniejszą formą cross-validacji jest K-fold Cross-Validation, która polega na iteracyjnym dzieleniu zbioru treningowego na k partycji, a następnie w każdej iteracji jedna partycja zostaje użyta jako zbiór testowy, a pozostałe k-1 jako zbiór treningowy.



K-Fold Cross Validation (Image by Gufosowa from WikiMedia)

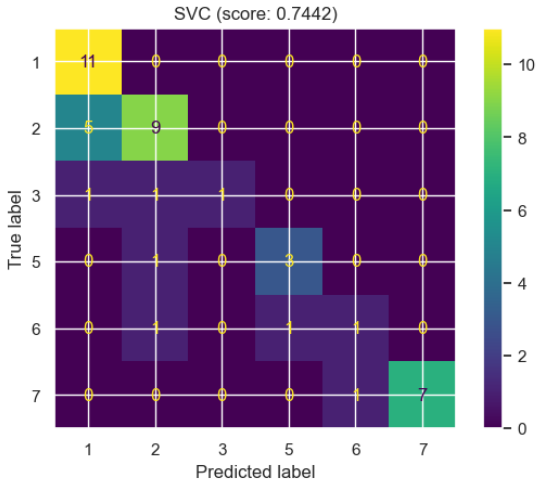
4. ocena klasyfikacji – do porównania wyników różnego typu przygotowania danych oraz wykorzystanego klasyfikatora użyj poznanych metryk oceny klasyfikacji i zinterpretuj wyniki. (20 punktów)

Dla danych zeskalowanych:



	precision	recall	f1-score	support
1	0.6667	0.9091	0.7692	11
2	0.7143	0.7143	0.7143	14
3	0.0000	0.0000	0.0000	3
5	1.0000	0.5000	0.6667	4
6	0.6667	0.6667	0.6667	3
7	0.8889	1.0000	0.9412	8
accuracy			0.7442	43
macro avg	0.6561	0.6317	0.6263	43
weighted avg	0.7080	0.7442	0.7130	43

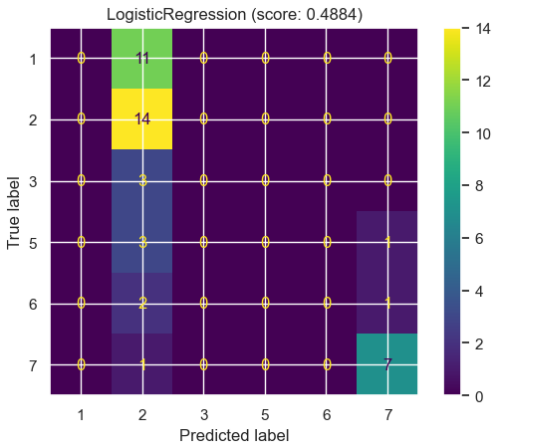
0.6417748917748918 0.2599944643511192



Report of confusion matrix

	precision	recall	f1-score	support
1	0.6471	1.0000	0.7857	11
2	0.7500	0.6429	0.6923	14
3	1.0000	0.3333	0.5000	3
5	0.7500	0.7500	0.7500	4
6	0.5000	0.3333	0.4000	3
7	1.0000	0.8750	0.9333	8
accuracy			0.7442	43
macro avg	0.7745	0.6558	0.6769	43
weighted avg	0.7702	0.7442	0.7326	43

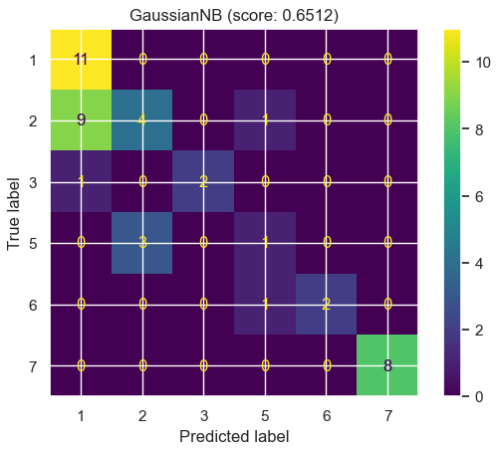
0.45346320346320346 0.09425406629795417



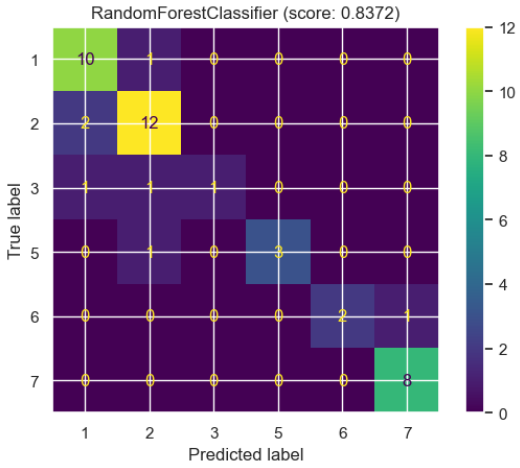
Report of confusion matrix

	precision	recall	f1-score	support
1	0.0000	0.0000	0.0000	11
2	0.4118	1.0000	0.5833	14
3	0.0000	0.0000	0.0000	3
5	0.0000	0.0000	0.0000	4
6	0.0000	0.0000	0.0000	3
7	0.7778	0.8750	0.8235	8
accuracy			0.4884	43
macro avg	0.1983	0.3125	0.2345	43
weighted avg	0.2788	0.4884	0.3431	43

0.4346320346320346 0.1633337406470414



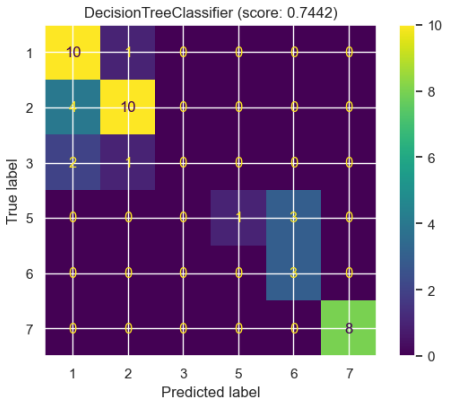

```
Report of confusion matrix
              precision    recall  f1-score   support

           1     0.5238    1.0000    0.6875        11
           2     0.5714    0.2857    0.3810        14
           3     1.0000    0.6667    0.8000         3
           5     0.3333    0.2500    0.2857         4
           6     1.0000    0.6667    0.8000         3
           7     1.0000    1.0000    1.0000         8

    accuracy                         0.6512        43
   macro avg     0.7381    0.6448    0.6590        43
weighted avg     0.6766    0.6512    0.6242        43

0.4396103896103896 0.22433642249242675
```



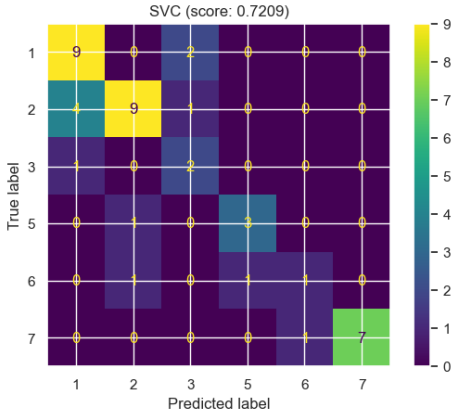

```
Report of confusion matrix
              precision    recall  f1-score   support

           1     0.7692    0.9091    0.8333        11
           2     0.8000    0.8571    0.8276        14
           3     1.0000    0.3333    0.5000         3
           5     1.0000    0.7500    0.8571         4
           6     1.0000    0.6667    0.8000         3
           7     0.8889    1.0000    0.9412         8

    accuracy                         0.8372        43
   macro avg     0.9097    0.7527    0.7932        43
weighted avg     0.8552    0.8372    0.8282        43

0.6876623376623376 0.2256325063527304
```

30.06.2023

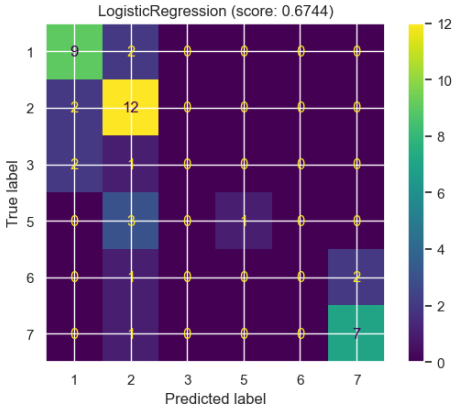
Dla danych niezeskalowanych:



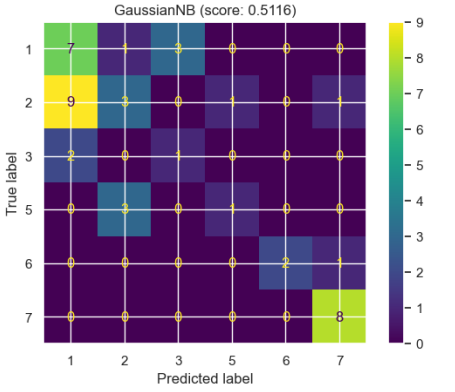

```
Report of confusion matrix
              precision    recall  f1-score   support

           1     0.6250    0.9091    0.7407        11
           2     0.8333    0.7143    0.7692        14
           3     0.0000    0.0000    0.0000         3
           5     1.0000    0.2500    0.4000         4
           6     0.5000    1.0000    0.6667         3
           7     1.0000    1.0000    1.0000         8

    accuracy                         0.7442        43
   macro avg     0.6597    0.6456    0.5961        43
weighted avg     0.7452    0.7442    0.7097        43

0.6123376623376624 0.29522698688018073
```



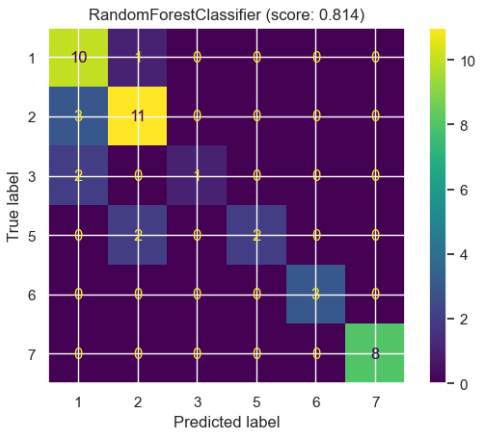

```
Report of confusion matrix
              precision    recall  f1-score   support

           1     0.6429    0.8182    0.7200        11
           2     0.8182    0.6429    0.7200        14
           3     0.4000    0.6667    0.5000         3
           5     0.7500    0.7500    0.7500         4
           6     0.5000    0.3333    0.4000         3
           7     1.0000    0.8750    0.9333         8

    accuracy                         0.7209        43
   macro avg     0.6852    0.6810    0.6706        43
weighted avg     0.7494    0.7209    0.7248        43

0.6225108225108225 0.29464082480943554
```




```
Report of confusion matrix
              precision    recall  f1-score   support

           1     0.6923    0.8182    0.7500        11
           2     0.6000    0.8571    0.7059        14
           3     0.0000    0.0000    0.0000         3
           5     1.0000    0.2500    0.4000         4
           6     0.0000    0.0000    0.0000         3
           7     0.7778    0.8750    0.8235         8

    accuracy                         0.6744        43
   macro avg     0.5117    0.4667    0.4466        43
weighted avg     0.6102    0.6744    0.6121        43

0.5878787878787879 0.2820584687909939
```




```
Report of confusion matrix
              precision    recall  f1-score   support

           1     0.5238    1.0000    0.6875        11
           2     0.5714    0.2857    0.3810        14
           3     1.0000    0.6667    0.8000         3
           5     0.3333    0.2500    0.2857         4
           6     1.0000    0.6667    0.8000         3
           7     1.0000    1.0000    1.0000         8

    accuracy                         0.6512        43
   macro avg     0.7381    0.6448    0.6590        43
weighted avg     0.6766    0.6512    0.6242        43

0.43008658008658013 0.26257507865881885
```

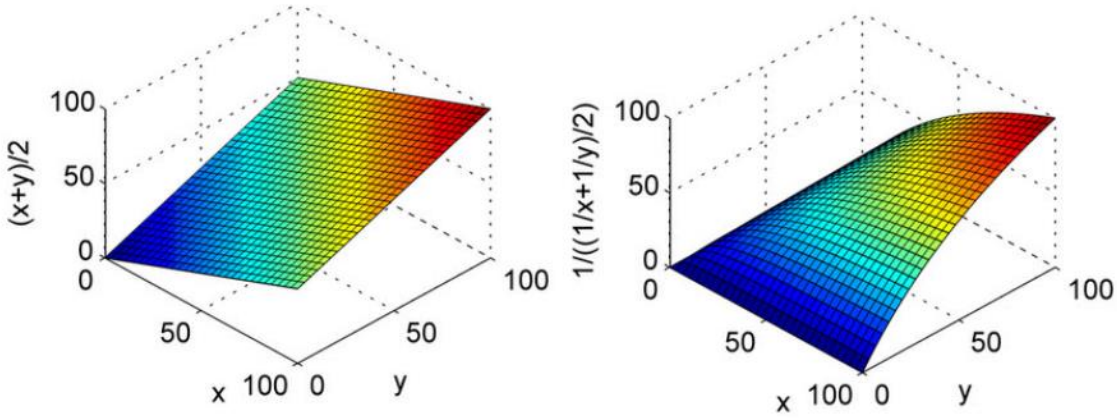


Report of confusion matrix

	precision	recall	f1-score	support
1	0.6667	0.9091	0.7692	11
2	0.7857	0.7857	0.7857	14
3	1.0000	0.3333	0.5000	3
5	1.0000	0.5000	0.6667	4
6	1.0000	1.0000	1.0000	3
7	1.0000	1.0000	1.0000	8
accuracy			0.8140	43
macro avg	0.9087	0.7547	0.7869	43
weighted avg	0.8450	0.8140	0.8053	43

Na podstawie powyższych rysunków można odczytać:

- precision – TP / (TP + FP) – wskazuje ile prawdziwie pozytywnych klasyfikacji dokonał model w stosunku do wszystkich pozytywnie udzielonych odpowiedzi. Metryka przydatna w sytuacji, gdy koszt udzielenia fałszywie pozytywnej klasyfikacji jest bardzo wysoki (np. przy diagnozie raka)
- recall – TP / (TP + FN) – mierzy zdolność modelu do określania pozytywnych przewidywań. Innymi słowy: im więcej razy model wskaże odpowiedź fałszywie negatywną (czyli taką, która powinna być prawdziwie pozytywna) tym mniejszy będzie wynikowy ułamek. Przydatna miara w przypadku gdy koszt pominięcia odpowiedzi pozytywnej jest wysoki i chcemy maksymalizować liczbę prawdziwie poprawnych odpowiedzi (np. wykrywanie pułapek - min)
- f1-score – średnia harmoniczna z precision oraz recall. Pozwala ona łatwiej zauważyć gdy któraś z wartości (precision lub recall) zbliża się do zera (wtedy cała średnia również dąży do 0)



Wykresy 2D dla średniej arytmetycznej (z lewej) i średniej harmonicznej (z prawej).

- Suport: Jest to liczba wystąpień danej klasy w zestawie danych. Ta metryka jest przydatna, wtedy gdy chcemy zrozumieć, jak dystrybuowane są dane i jak model może działać na różnych klasach. Dla przykładu, model może działać bardzo dobrze na klasie, która ma dużo próbek (wysokie wsparcie), ale nie tak dobrze na klasie, która ma tylko kilka próbek (niskie wsparcie).

Reasumując można zauważyć, że GNB gorzej sobie radzi z klasyfikacją szkła od drzewa decyzyjnego. Prawdopodobnie wynika to z faktu, że GNB zakłada wzajemną niezależność zmiennych (cząstek), co w przypadku inżynierii materiałowej jest nieprawdą. Ponadto GNB właśnie z racji na uproszczenia jakie stosuje algorytm w opisie modelu to jest on szybszy co może mieć swoje zastosowanie w systemach czasu rzeczywistego.

Na podstawie różnic w dokładności GNB oraz drzewa decyzyjnego dla danych przetworzonych i nieprzetworzonych przez scalery można zauważyć, że GNB jest bardziej wrażliwy na brak standaryzacji danych.