Klasyfikatory

Magdalena Ośka
Paweł Okrutny
Michał Radziwanowski

Opis danych

400 klientów banku

Atrybuty:

`Income:` roczny dochód w $1,000.

`Limit:` Limit kredytowy

`Rating:` Rating kredytowy

`Cards:` Liczba kart kredytowych

`Age:` Wiek w latach

`Education:` Liczba lat edukacji

`Gender:` Kobieta/Mężczyzna (płeć)

`Student:` Tak/Nie (czy student)

`Married:` Tak/Nie (czy w związku małżeńskim)

`Ethnicity:`African American/Asian/Caucasian (pochodzenie etniczne)

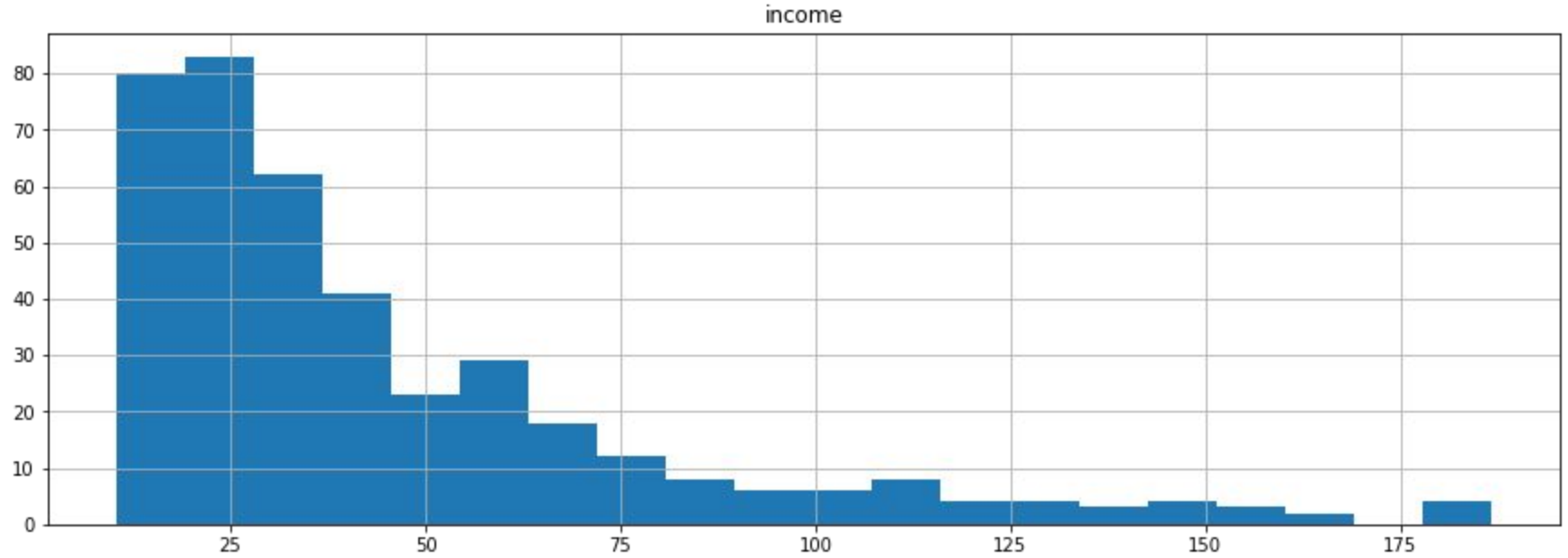
`Balance:` Średnie miesięczne saldo karty kredytowej w $

Przykład wartości danych

```
data.head()
```

	Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
0	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333
1	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903
2	104.593	7075	514	4	71	11	Male	No	No	Asian	580
3	148.924	9504	681	3	36	11	Female	No	No	Asian	964
4	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

Histogram dochodów



Większość klientów ma zdecydowanie niższe dochody.

Podział na grupy dochodowe / klasy ekonomiczne

The Poor: 0-18.000$ (0-5650.700 pln/msc)

Working Class: 18-30.000$ (5650.700-9.400 pln/msc)

Lower-Middle Class: 30-50.000$ (9.400-15.500 pln/msc)

Upper-Middle Class: 50-100.000$ (15.500-31.400 pln/msc)

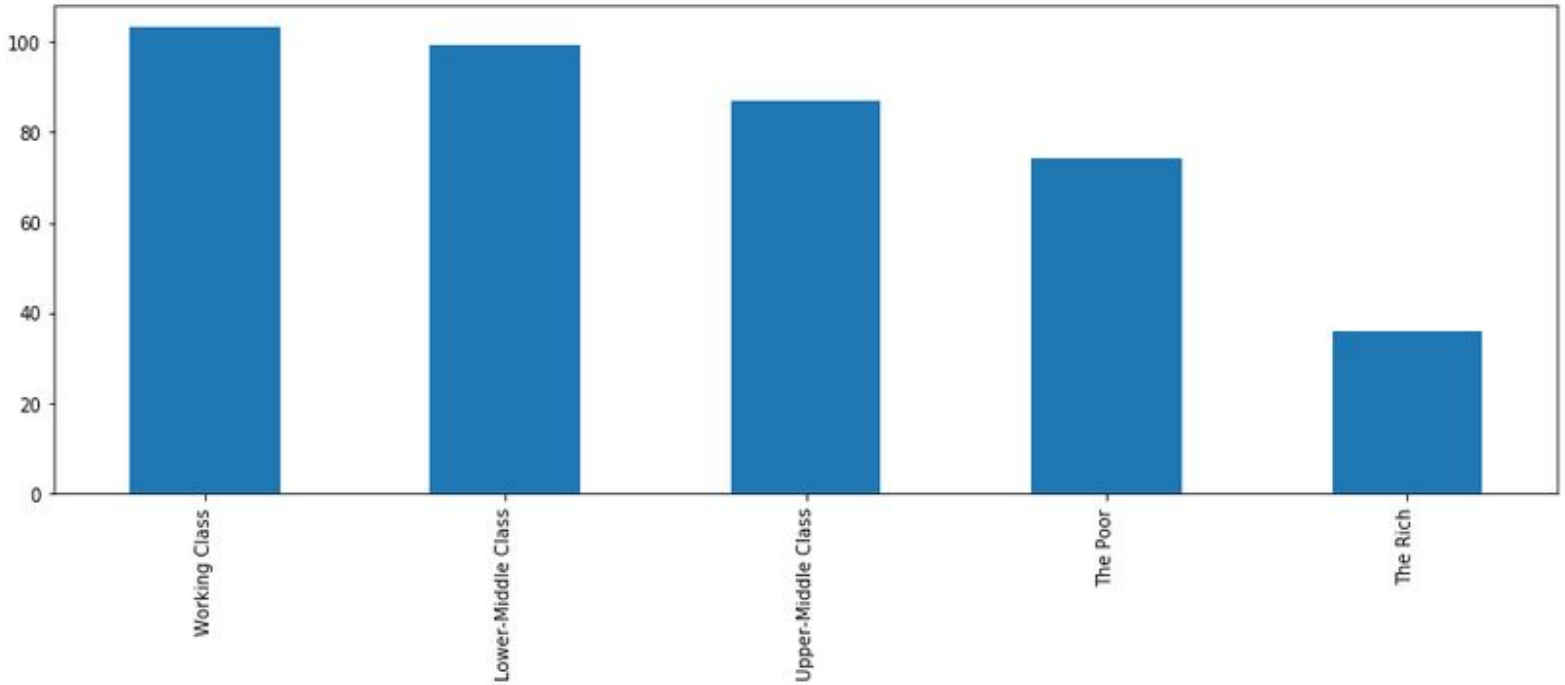
The Rich: 100.000$+ (31.400+ pln/msc)

Oparte na modelu klas w USA Leonarda Beeghley, 2004.

Przykładowe dane z przypisaną grupą dochodową

	income	limit	rating	cards	age	education	gender	student	married	ethnicity	balance	income_group
0	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333	The Poor
1	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903	The Rich
2	104.593	7075	514	4	71	11	Male	No	No	Asian	580	The Rich
3	148.924	9504	681	3	36	11	Female	No	No	Asian	964	The Rich
4	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331	Upper-Middle Class

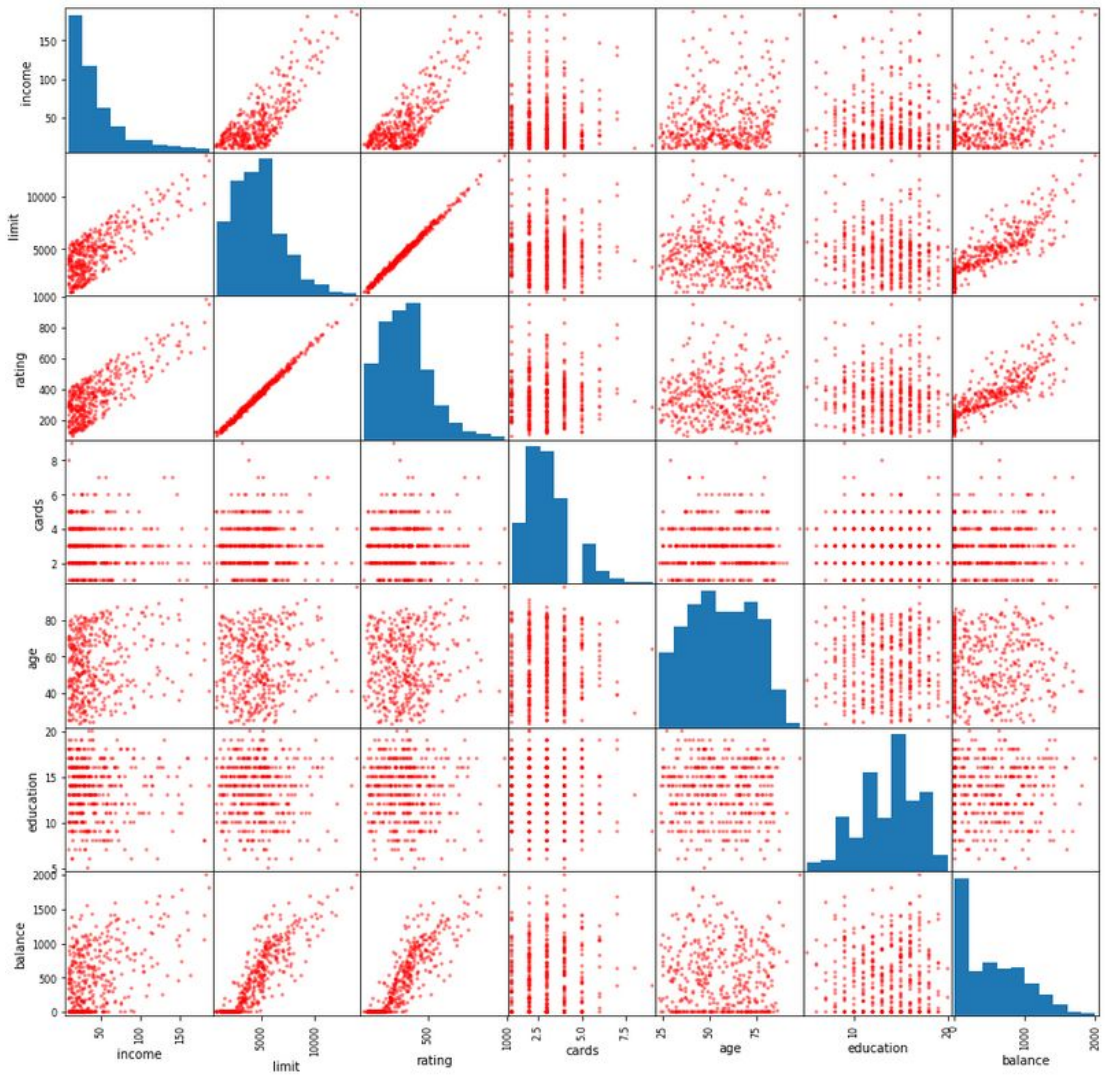
Liczebność grup w zestawie danych



Macierz rozrzutu

Zaobserwowane pozytywne korelacje:

- Rating i limit
- Limit i przychód
- Rating i przychód
- Limit i saldo
- Rating i saldo



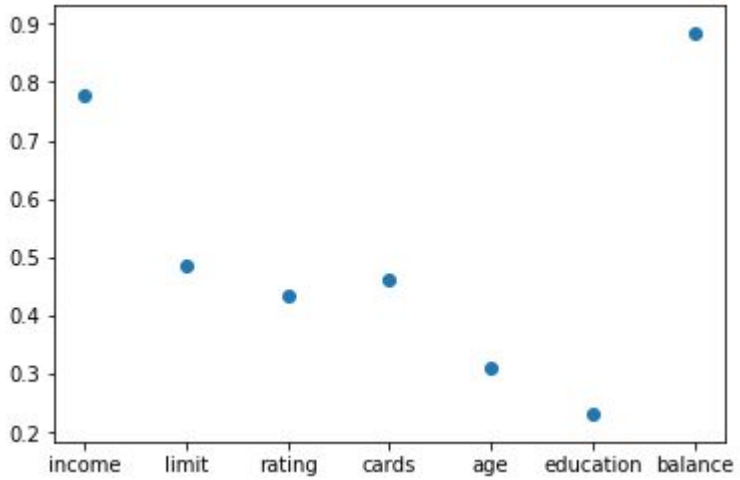
Podstawowe statystyki

```
data.describe()
```

	income	limit	rating	cards	age	education	balance
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	45.218885	4735.600000	354.940000	2.957500	55.667500	13.450000	520.015000
std	35.244273	2308.198848	154.724143	1.371275	17.249807	3.125207	459.758877
min	10.354000	855.000000	93.000000	1.000000	23.000000	5.000000	0.000000
25%	21.007250	3088.000000	247.250000	2.000000	41.750000	11.000000	68.750000
50%	33.115500	4622.500000	344.000000	3.000000	56.000000	14.000000	459.500000
75%	57.470750	5872.750000	437.250000	4.000000	70.000000	16.000000	863.000000
max	186.634000	13913.000000	982.000000	9.000000	98.000000	20.000000	1999.000000

Współczynniki zmienności

Klienci są najbardziej zróżnicowani pod względem dochodu (`income`) oraz salda (`balance`). Najmniej zróżnicowani są natomiast pod względem lat edukacji (`education`).



Klasyfikacja

Dane zostały podzielone:

- 80% danych to zbiór uczący `(train)`
- 20% danych to zbiór testowy `(test)`

Algorytm k najbliższych sąsiadów (k-nn)

`KNeighborsClassifier()`

Klasyfikacja ta polega na przypisaniu obiektu do klasy najczęściej występującej wśród jego najbliższych sąsiadów.

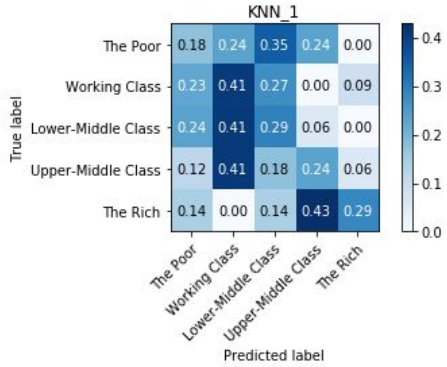
Gdzie **k** jest dodatnią liczbą całkowitą, zwykle małą, a sąsiedzi są wybierani przy pomocą funkcji odległości.

Jeśli k = 1, obiekt jest po prostu przypisywany do klasy jego najbliższego sąsiada.

Algorytm k najbliższych sąsiadów (k-nn)

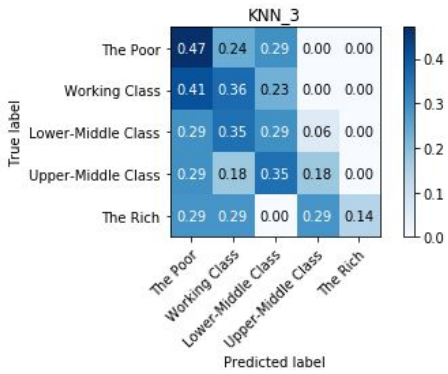
KNN_1

overall accuracy = 0.29



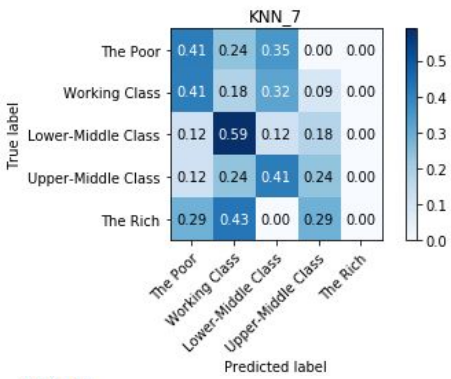
KNN_3

overall accuracy = 0.31



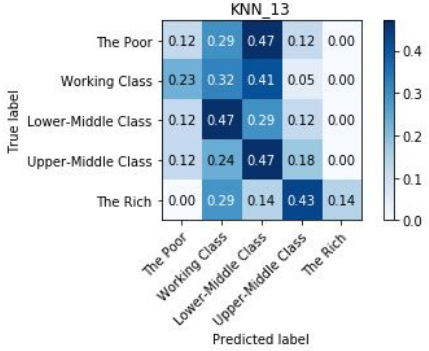
KNN_7

overall accuracy = 0.21



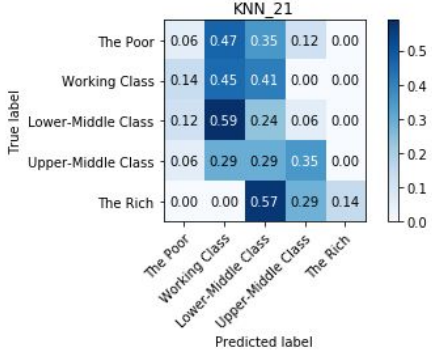
KNN_13

overall accuracy = 0.23



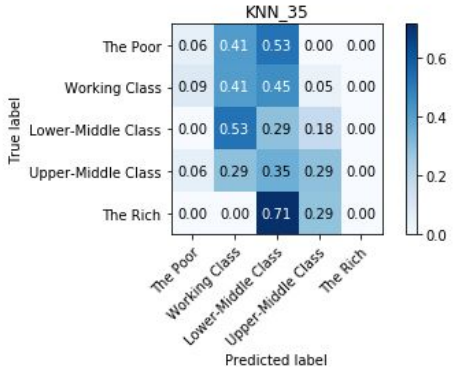
KNN_21

overall accuracy = 0.28



KNN_35

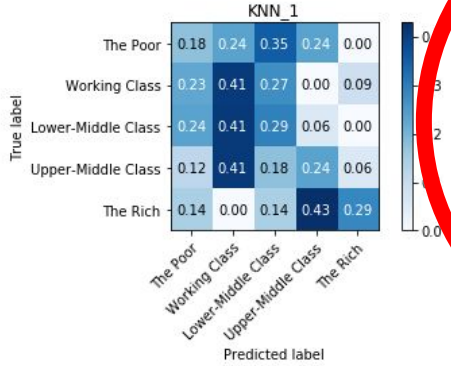
overall accuracy = 0.25



Algorytm k najbliższych sąsiadów (k-nn)

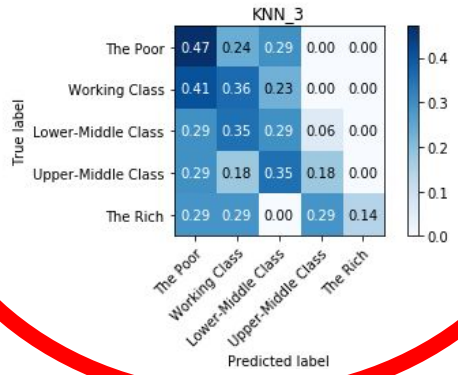
KNN_1

overall accuracy = 0.29



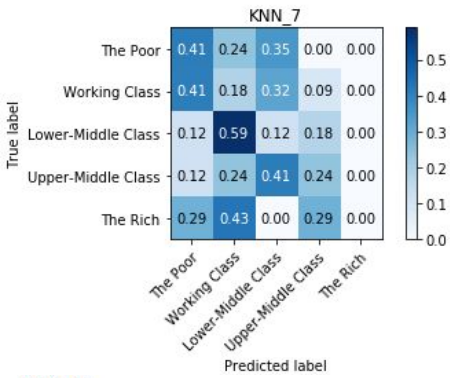
KNN_3

overall accuracy = 0.31



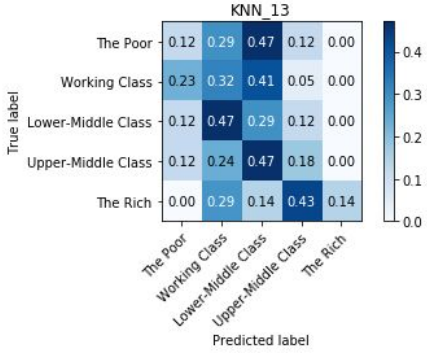
KNN_7

overall accuracy = 0.21



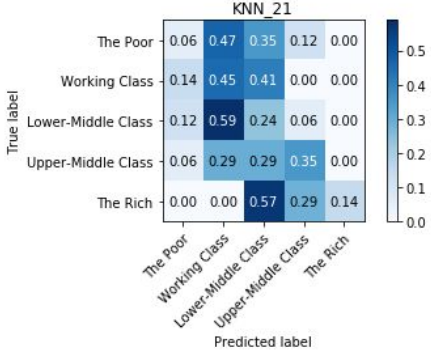
KNN_13

overall accuracy = 0.23



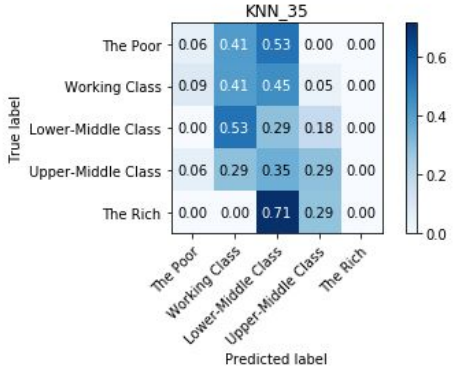
KNN_21

overall accuracy = 0.28



KNN_35

overall accuracy = 0.25



Algorytm k najbliższych sąsiadów (k-nn)

Słabo! Dlaczego?

Słaba wydajność KNN może być spowodowana dużą liczbą cech słabo skorelowanych z problemem klasyfikacji. Gdy tylko niewielka część cech daje jakąkolwiek informację, sąsiedztwo może być zdominowane przez rekordy, które są bliskie tylko na podstawie słabych cech.

Algorytm k najbliższych sąsiadów (k-nn)

Na przykład: mamy dane o ludziach: wiek, masa ciała, wzrost, płeć, kolor oczu i problem kategoryzacji “jest na emeryturze”. Jak mamy tyle cech, to skuteczność będzie słaba, bo wśród nich tylko wiek coś mówi o problemie. Ale jako, że jest dużo innych danych to ludzie o podobnym wyglądzie będą w sąsiedztwie najbliżej.

Gdy z cech zostawimy tylko wiek, to mamy skuteczność prawie 100%, bo jest to cecha mocno skorelowana z problemem.

Jak pokazano później, ograniczenie liczby atrybutów używanych do klasyfikatora k-nn poprawia wyniki, co czyni go jedną z najlepszych opcji dla tego problemu klasyfikacji.

Drzewa decyzyjne

`DecisionTreeClassifier()`

Celem jest stworzenie modelu, który przewiduje wartość zmiennej docelowej, ucząc się prostych reguł decyzyjnych na podstawie cech danych.

Algorytm próbuje rozwiązać problem za pomocą reprezentacji drzewa.

Każdy wewnętrzny **węzeł** drzewa odpowiada **atrybutowi**, a każdy węzeł **liścia** odpowiada etykiecie **klasy**. Ponieważ jest to zestaw decyzji binarnych, łatwo jest zwizualizować ten model jako drzewo (przykład za chwilę).

Drzewa decyzyjne

DT

accuracy = 0.62

domyślne

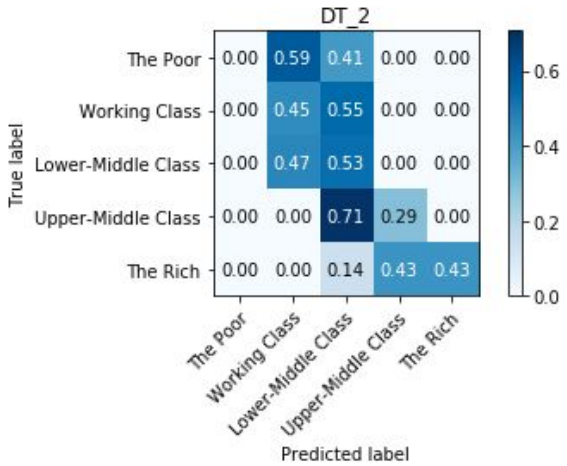
DT_E

accuracy = 0.57

kryterium entropii

DT_2

accuracy = 0.34

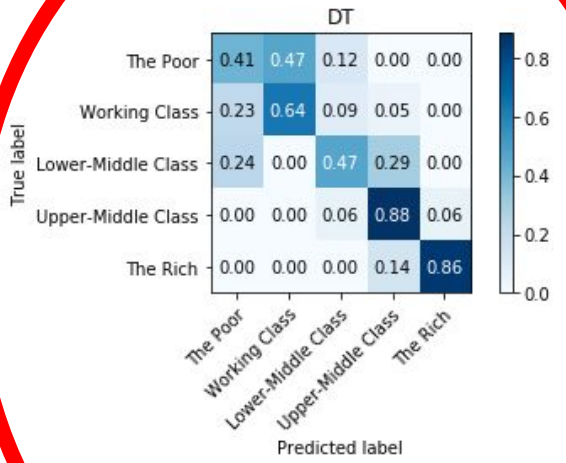


głębokość ograniczona do 2

Drzewa decyzyjne

DT

accuracy = 0.62



domyślne

DT_E

accuracy = 0.57

kryterium entropii

DT_2

accuracy = 0.34

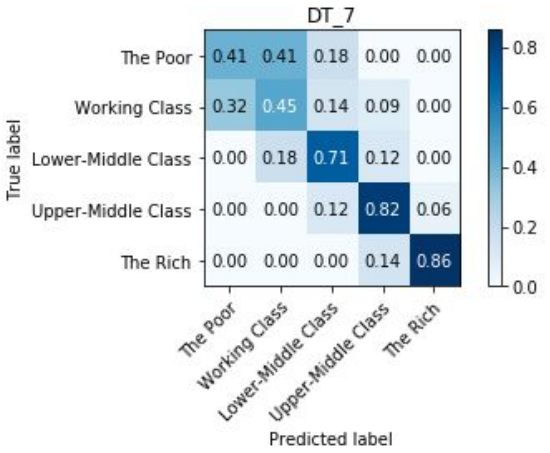
głębokość ograniczona do 2

Najlepsze accuracy = 0.62
Drzewo domyślne

Drzewa decyzyjne

DT_7

accuracy = 0.61



głębokość ograniczona do 7

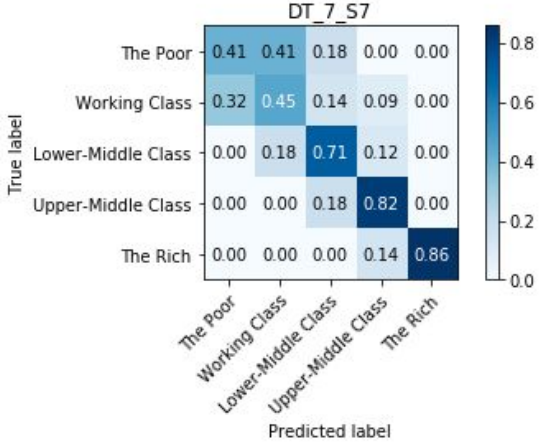
DT_E_7

accuracy = 0.55

głębokość ograniczona do 7 + kryterium entropii

DT_7_S7

accuracy = 0.61



głębokość ograniczona do 7 + minimalna liczba próbek wymaganych do podziału węzła równa 7

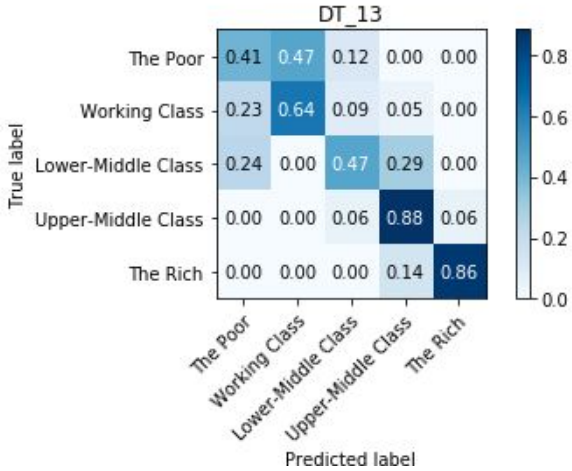
Najlepsze accuracy = 0.62
Drzewo domyślne

Drzewa decyzyjne

DT_7_S13
accuracy = 0.57

głębokość ograniczona do 7 + minimalna liczba próbek wymaganych do podziału węzła równa 13

DT_13
accuracy = 0.62



głębokość ograniczona do 13

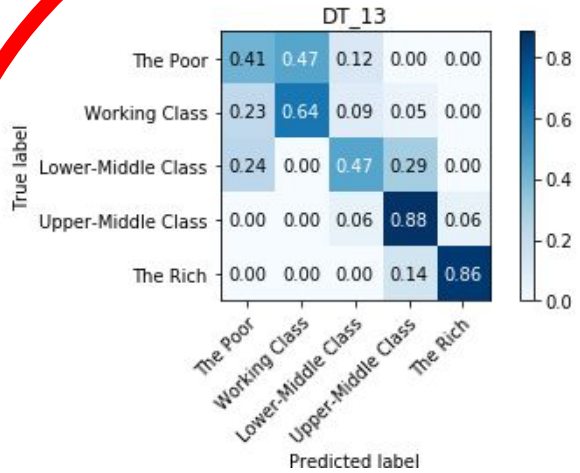
Najlepsze accuracy = 0.62
Drzewo domyślne oraz drzewo
o limicie głębokości 13

Drzewa decyzyjne

DT_7_S13
accuracy = 0.57

głębokość ograniczona do 7 + minimalna liczba próbek wymaganych do podziału węzła równa 13

DT_13
accuracy = 0.62



głębokość ograniczona do 13

Drzewa decyzyjne

Jak widać efektywność drzew decyzyjnych jest znacznie lepsza niż k-NN.

Ale czy może być jeszcze lepsza?

Drzewa decyzyjne

Jak widać efektywność drzew decyzyjnych jest znacznie lepsza niż k-NN.

Ale czy może być jeszcze lepsza?

Spróbujmy użyć metody z biblioteki scikit learn: `GridSearchCV.`

Grid Search służy do znalezienia optymalnych parametrów modelu.

GridSearchCV

Aby użyć powyższego narzędzia przekazujemy do tej metody następujący zestaw parametrów:

```
grid_param = {

    'max_depth': [None, 3, 5, 7, 11, 15, 21],

    'criterion': ['gini', 'entropy'],

    'min_samples_split': [2, 5, 7, 11, 15, 20, 30],

    'min_samples_leaf': [1, 3, 5, 7],

    'random_state': [42]

}
```

GridSearchCV

Aby użyć powyższego narzędzia przekazujemy do tej metody następujący zestaw parametrów:

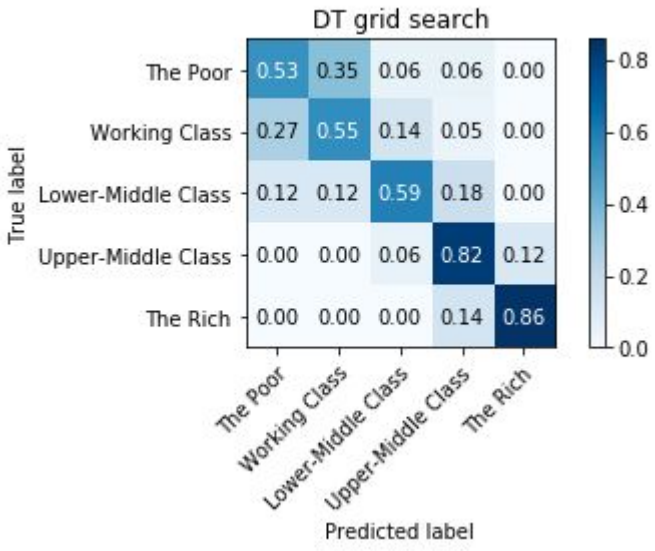
```
grid_param = {

    'max_depth': [None, 3, 5, 7, 11, 15, 21],

    'criterion': ['gini', 'entropy'],

    'min_samples_split': [2, 5, 7, 11, 15, 20, 30],

    'min_samples_leaf': [1, 3, 5, 7],

    'random_state': [42]

}
```

GridSearch przetestuje te parametry i zwróci najlepszy model.

Drzewa decyzyjne cd.

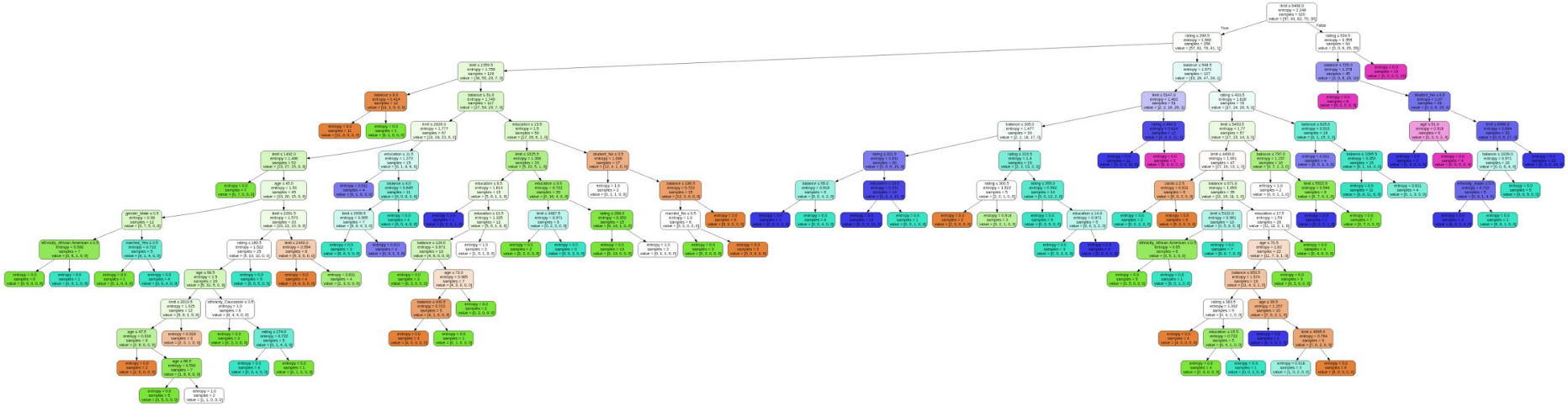
```
'max_depth': None

'criterion': 'entropy'

'min_samples_split': 5,

'min_samples_leaf': 1,

'random_state': 42
```

accuracy = 0.64



Udało się uzyskać dokładność **0.64**, która jest lepsza niż poprzednia, która była równa 0.62.

Drzewa decyzyjne



Barwa węzłów oznacza dominującą klasę.

Nasycenie rośnie wraz ze wzrostem dominacji danej klasy.

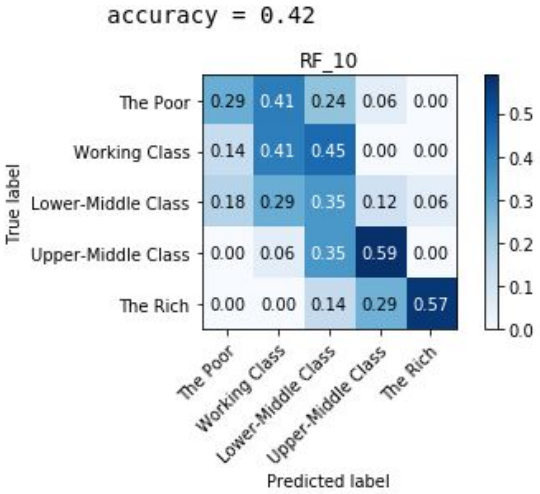
Random Forest (Las Losowy)

`RandomForestClassifier`

Algorytm tworzy las losowych drzew decyzyjnych.

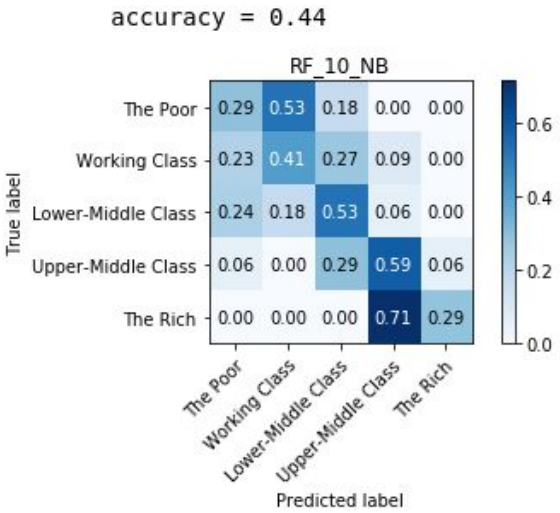
Drzewa te są tworzone na różnych podzbiorach danych oraz wykorzystuje się uśrednianie, aby poprawić dokładność predykcji oraz **kontrolować nadmierne dopasowanie.**

Random Forest

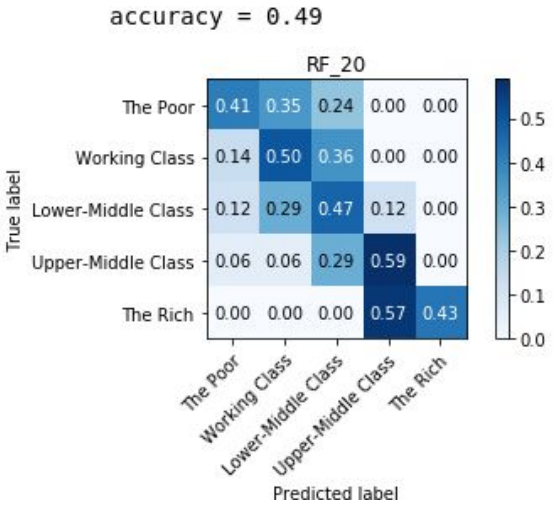


Liczba drzew = 10

Zbalansowane wagi klas



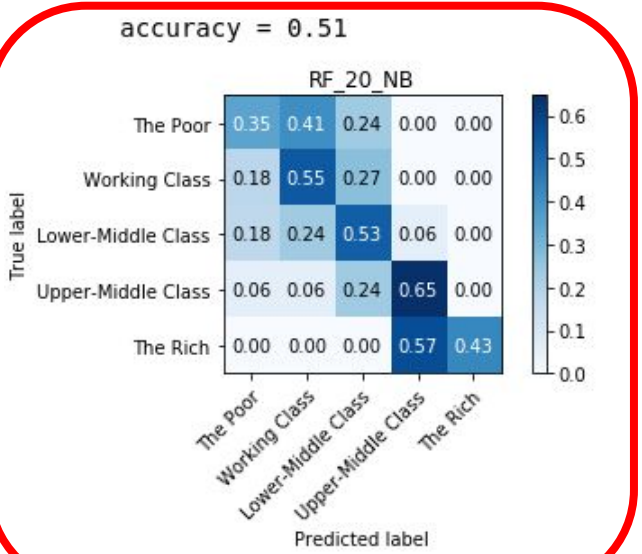
Liczba drzew = 10



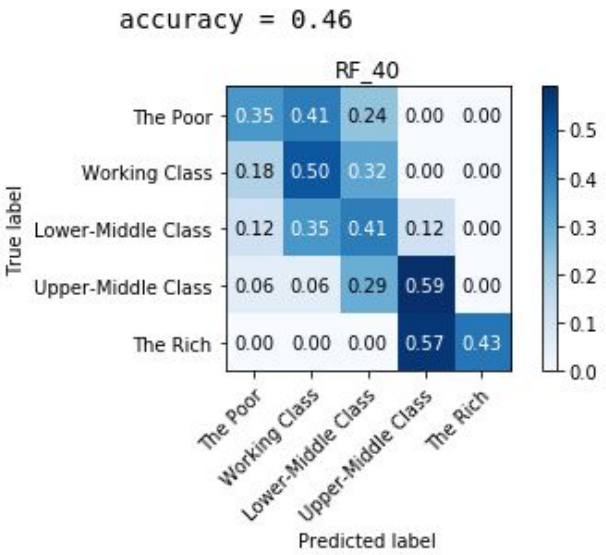
Liczba drzew = 20

Zbalansowane wagi klas

Random Forest



Liczba drzew = 20



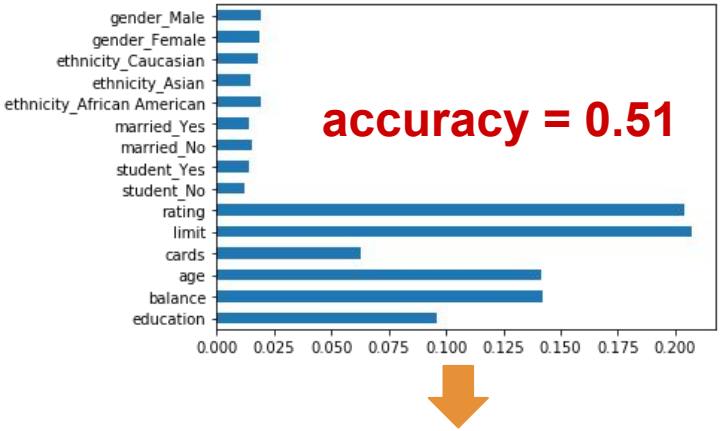
Liczba drzew = 40

Zbalansowane wagi klas

Najlepsza dokładność wynosi 0.51 dla lasu z liczbą drzew 20.

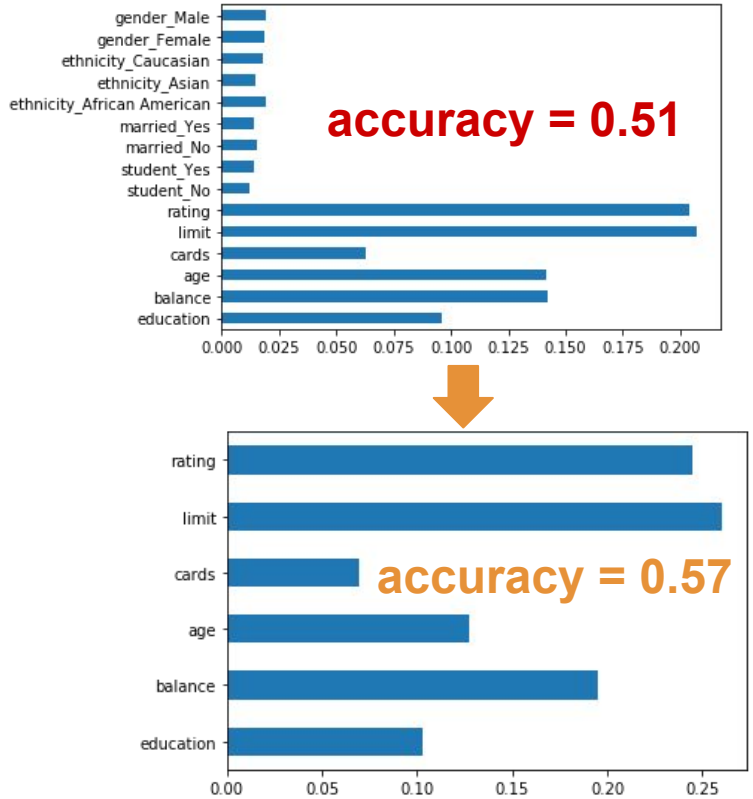
Spróbujmy więc ulepszyć ten model.

Random Forest



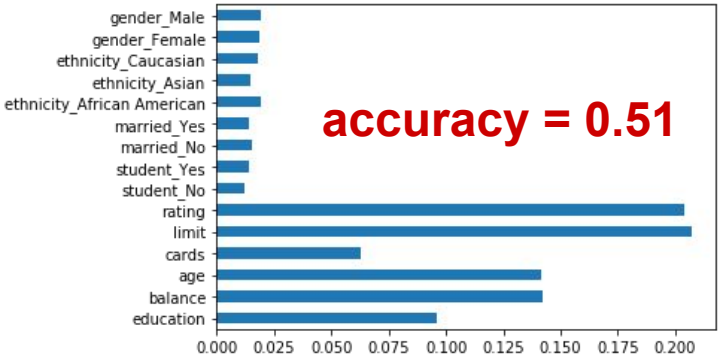
Klasyfikator Random Forest umożliwia śledzenie ważności cech, jak zostały one użyte podczas procesu uczenia się. Wizualizując znaczenie cech, można wybrać te, które są wspierają dla dokładność predykcji.

Random Forest

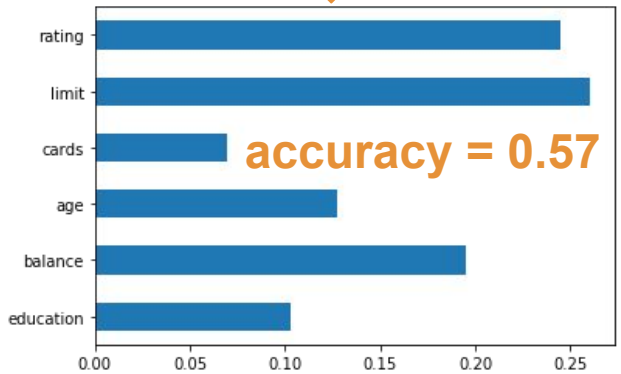


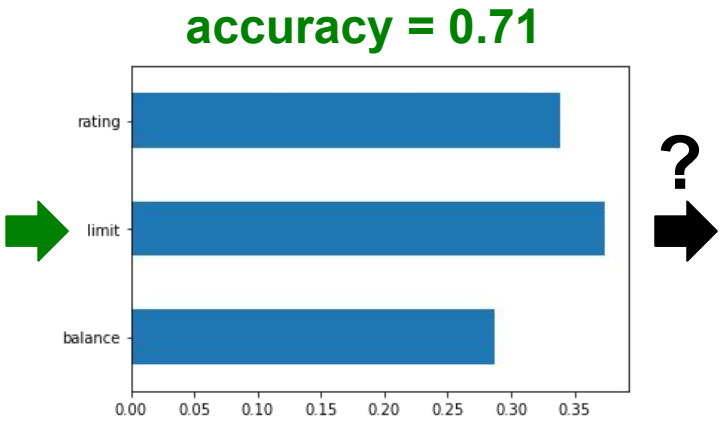
Klasyfikator Random Forest umożliwia śledzenie ważności cech, jak zostały one użyte podczas procesu uczenia się. Wizualizując znaczenie cech, można wybrać te, które są wspierają dla dokładność predykcji.

Random Forest

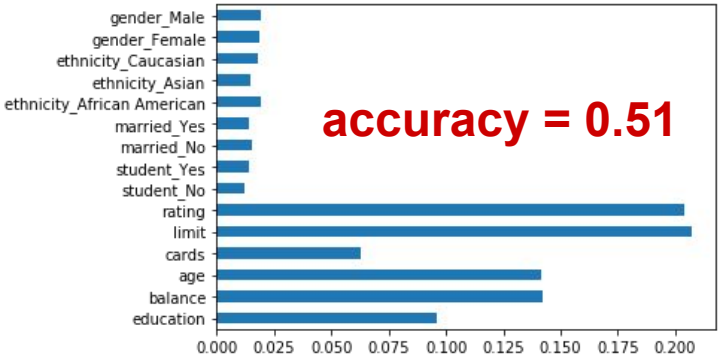


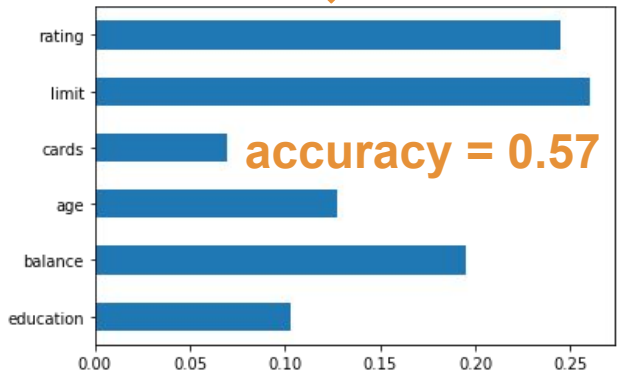
Klasyfikator Random Forest umożliwia śledzenie ważności cech, jak zostały one użyte podczas procesu uczenia się. Wizualizując znaczenie cech, można wybrać te, które są wspierają dla dokładność predykcji.



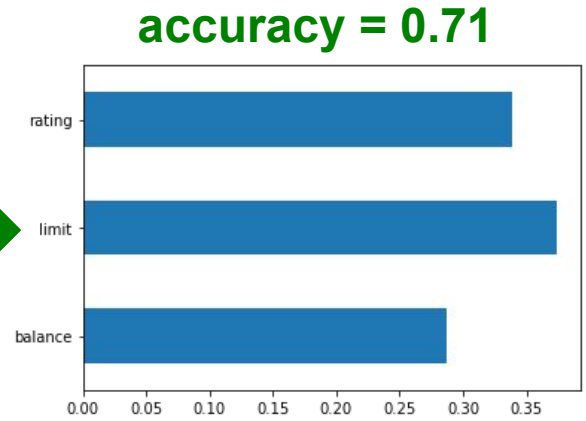


Random Forest





Klasyfikator Random Forest umożliwia śledzenie ważności cech, jak zostały one użyte podczas procesu uczenia się. Wizualizując znaczenie cech, można wybrać te, które są wspierają dla dokładność predykcji.



?

Ograniczenie do limit i rating powoduje już spadek dokładności do 0.39.

Regresja logistyczna

`LogisticRegression()`

W tym modelu prawdopodobieństwa opisujące możliwe wyniki pojedynczej próby są modelowane przy użyciu funkcji logistycznej:

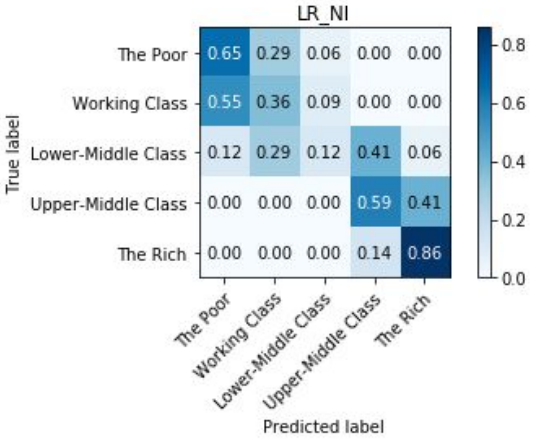
$$f(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

Która to może przyjmować dowolną liczbę o wartościach rzeczywistych i mapować ją do wartości między 0 a 1.

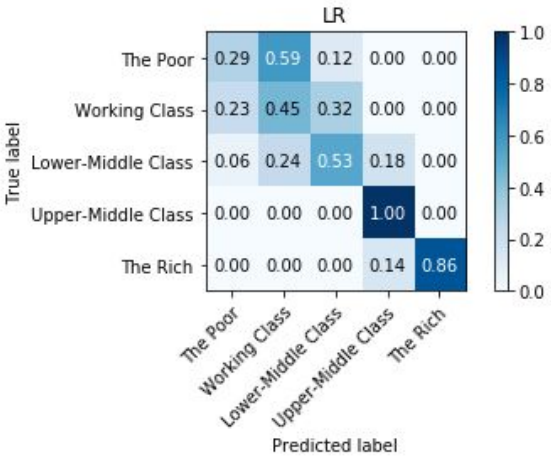
W problemach wieloklasowych funkcja logistyczna jest dopasowywana dla każdej klasy i dla danej wartości wybierane jest jedno o najwyższym prawdopodobieństwie.

Regresja logistyczna

accuracy = 0.46



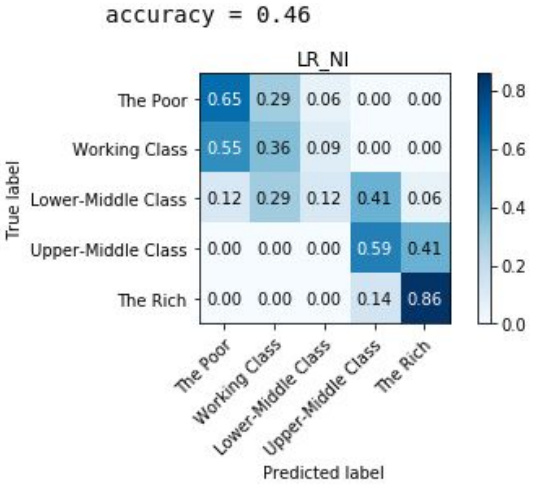
accuracy = 0.59

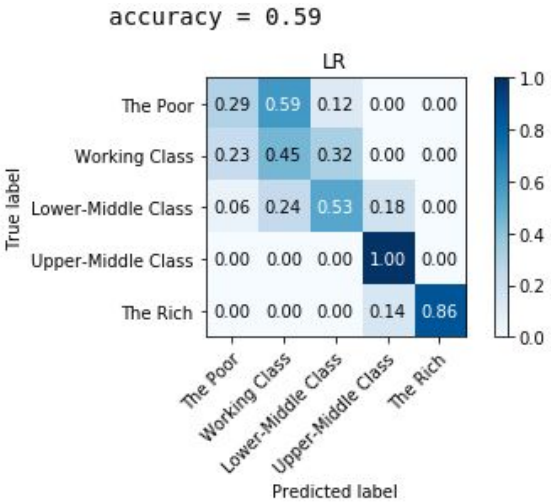


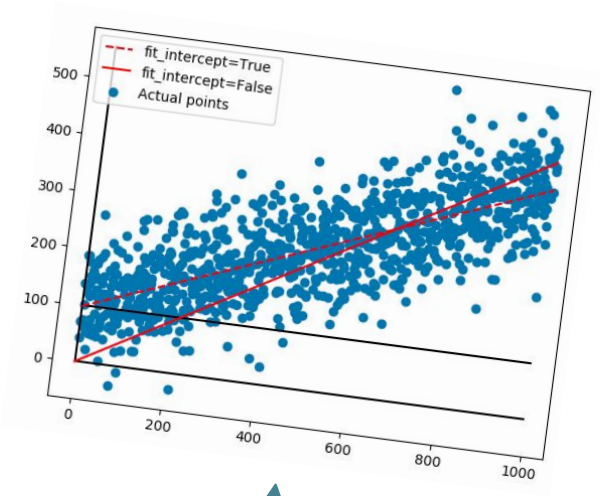
`fit_intercept = False`

`fit_intercept = True`

Regresja logistyczna





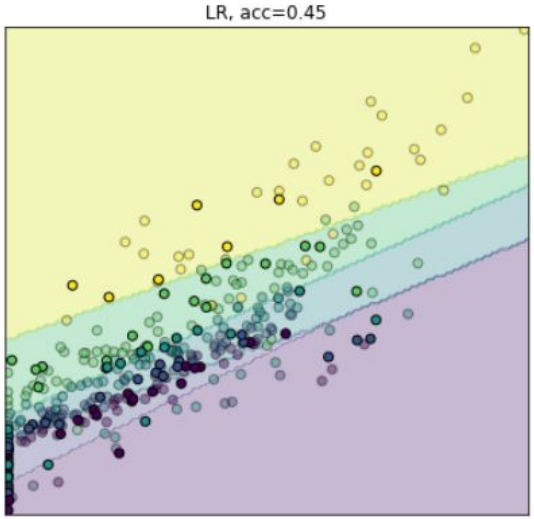


`fit_intercept = False`

`fit_intercept = True`

Decision Boundaries (Granice decyzyjne)

Wizualizacja kryteriów klasyfikacji jako granice decyzyjne - każdy **punkt** to **obiekt** z zestawu danych, a **osie** reprezentują dwie wybrane **cechy**. Granica decyzyjna dzieli punkty na **przestrzenie**, które są **klasami**, do których te punkty należą.

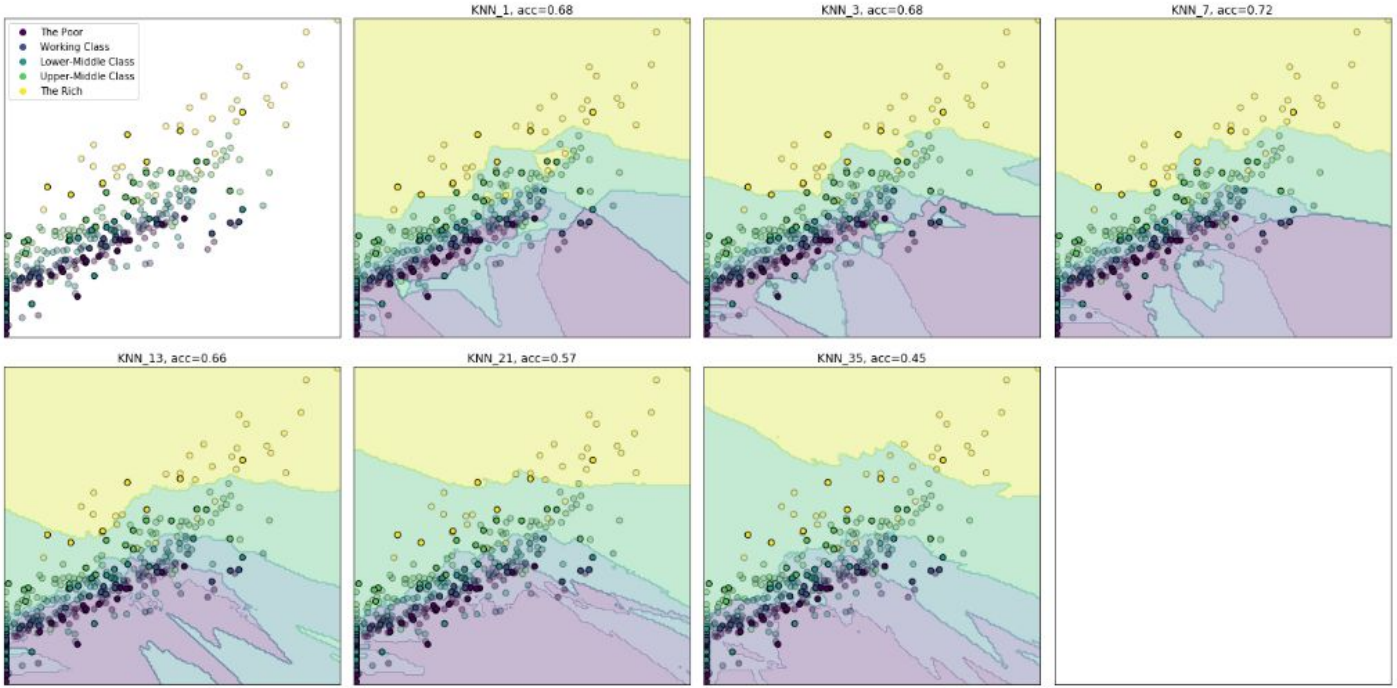


Kolory tła przedstawiają przestrzenie prawdopodobieństw klas.

Bardziej **transparentne** punkty oznaczają próbki z zestawu **treningowego**, natomiast bardziej **nasycone** pochodzą z próbki **testowej**.

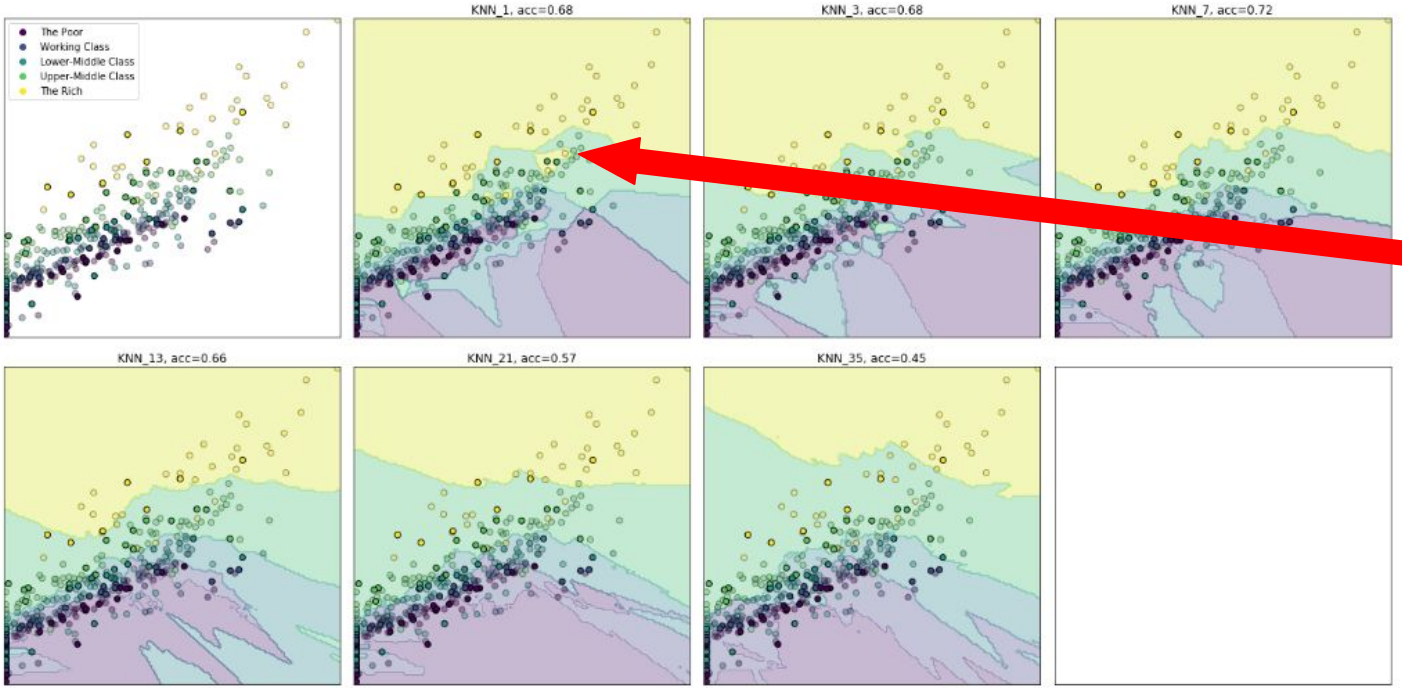
Decision Boundaries (Granice decyzyjne)

Wizualizacja kryteriów klasyfikacji k-NN dla cech Limit i Balance.



Decision Boundaries (Granice decyzyjne)

Wizualizacja kryteriów klasyfikacji k-NN dla cech Limit i Balance.

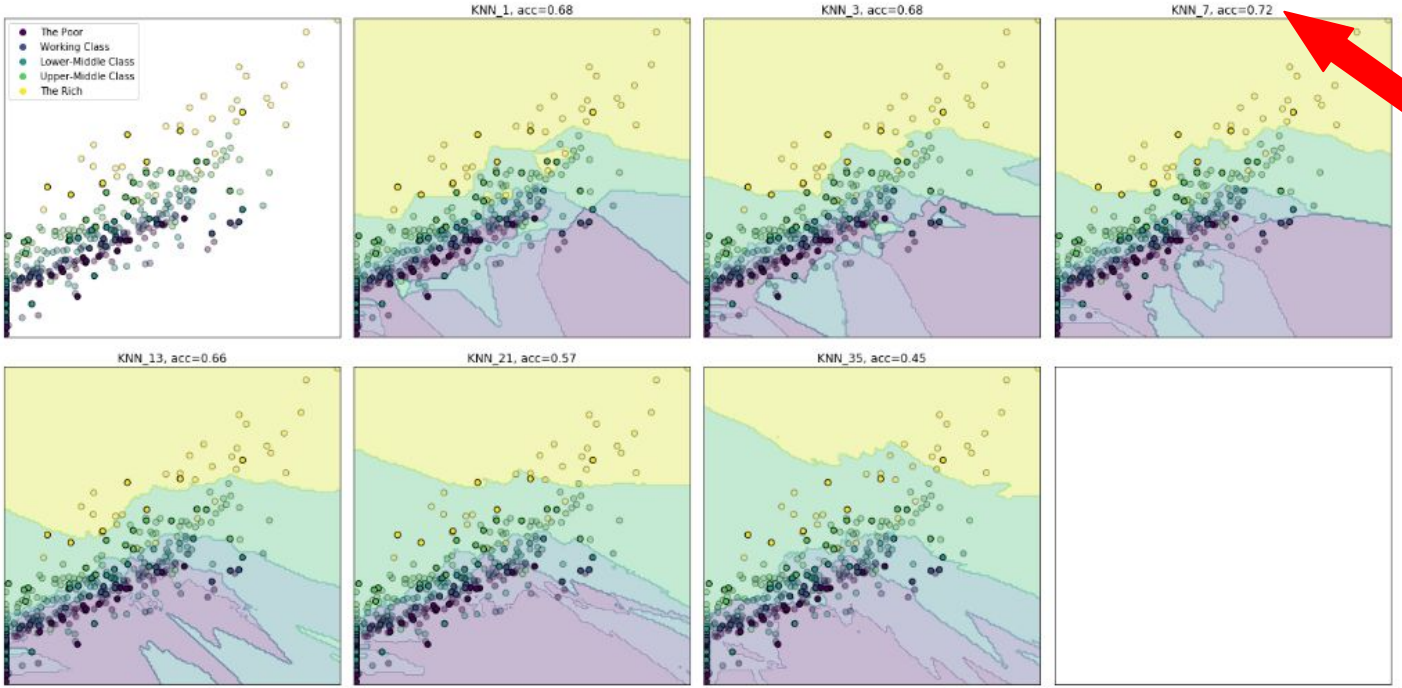


Przeuczenie

“Wyspy” z żółtymi klasami ze źle sklasyfikowanymi zielonymi punktami.

Decision Boundaries (Granice decyzyjne)

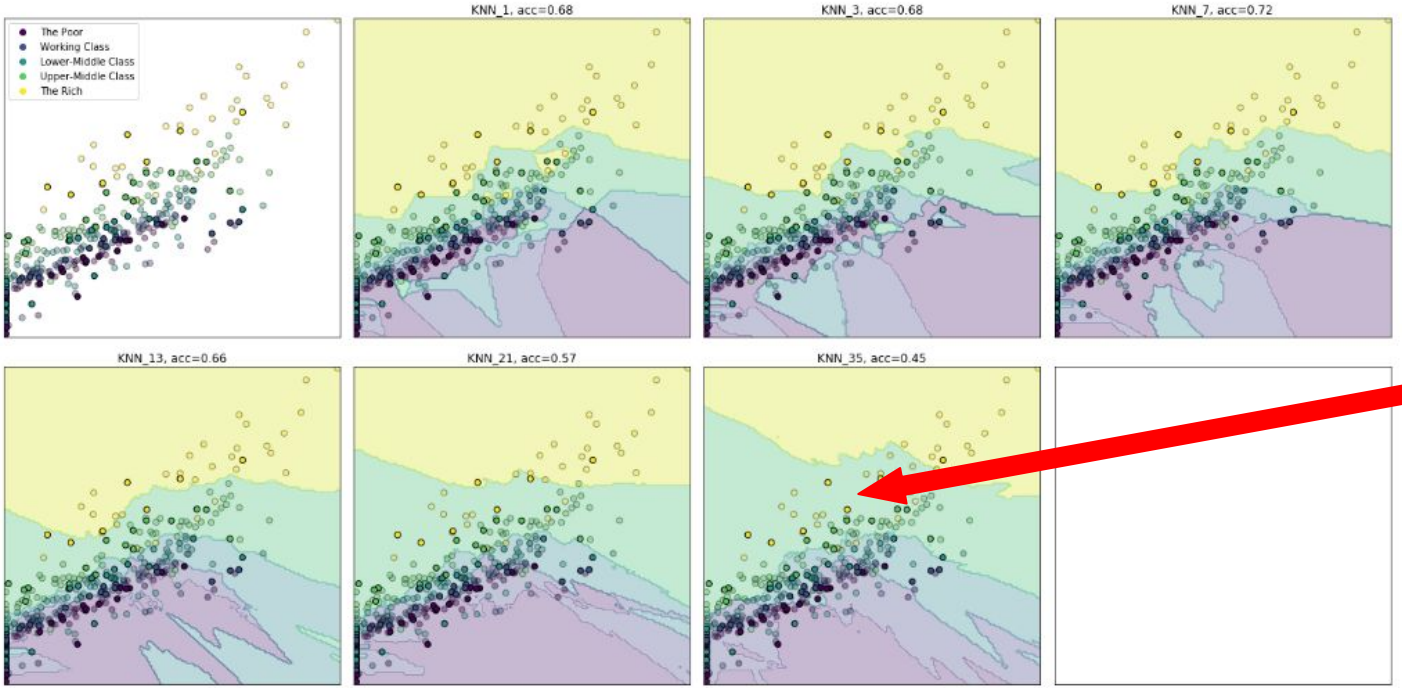
Wizualizacja kryteriów klasyfikacji k-NN dla cech Limit i Balance.



Najlepsza separacja

Decision Boundaries (Granice decyzyjne)

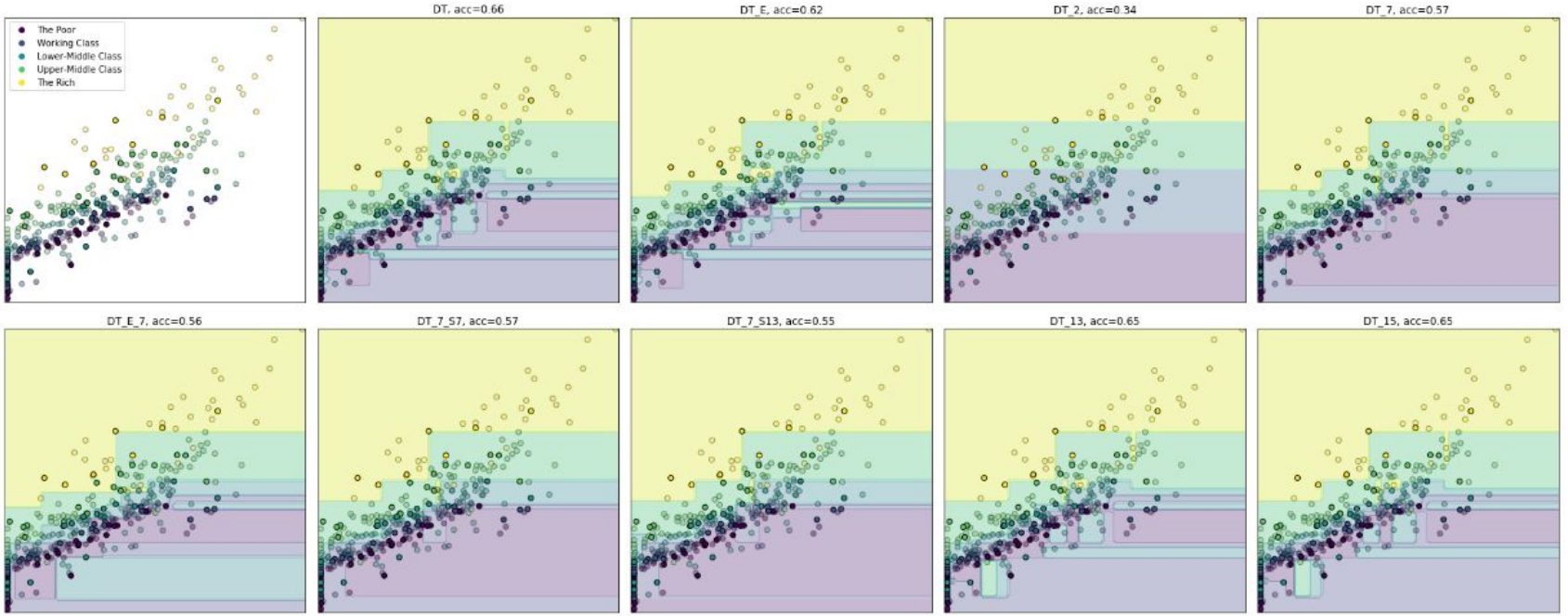
Wizualizacja kryteriów klasyfikacji k-NN dla cech Limit i Balance.



Wraz ze wzrostem k punkty rozproszone są coraz gorzej sklasyfikowane.

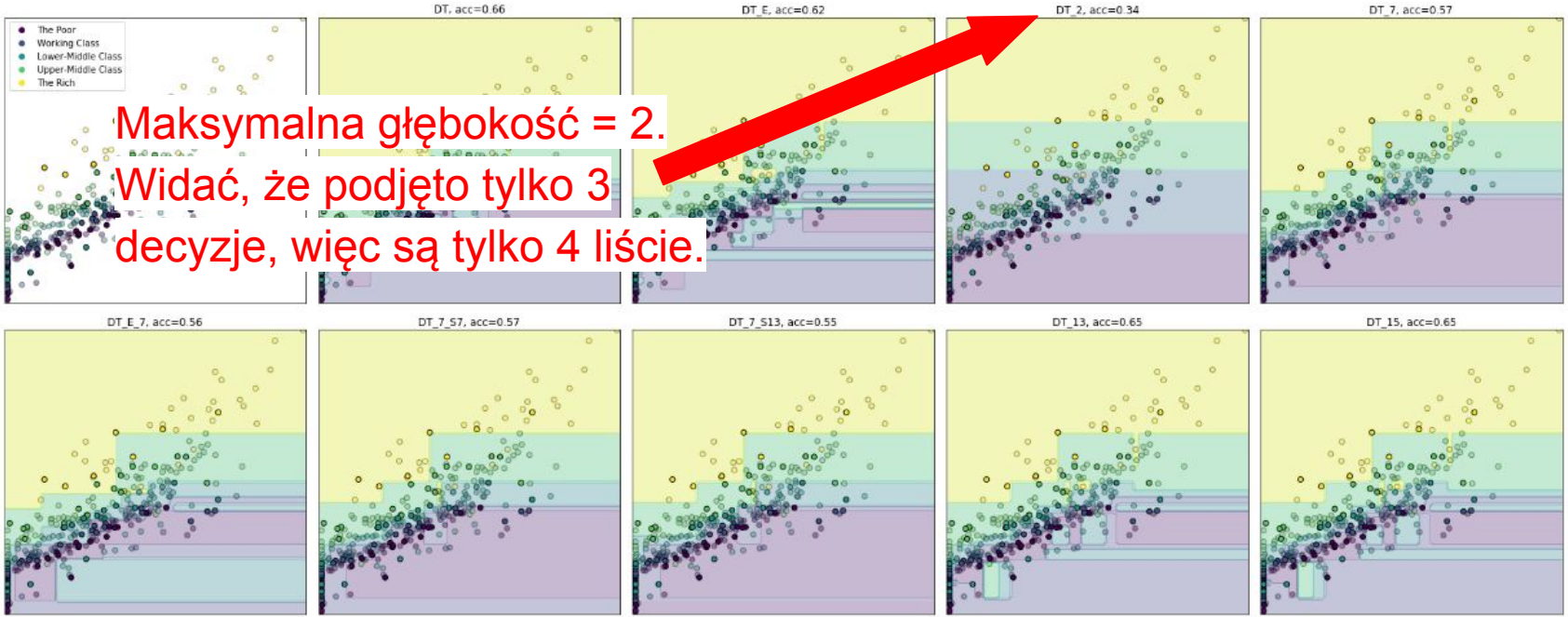
Decision Boundaries (Granice decyzyjne)

Wizualizacja kryteriów klasyfikacji Drzew Decyzyjnych dla cech Limit i Balance.



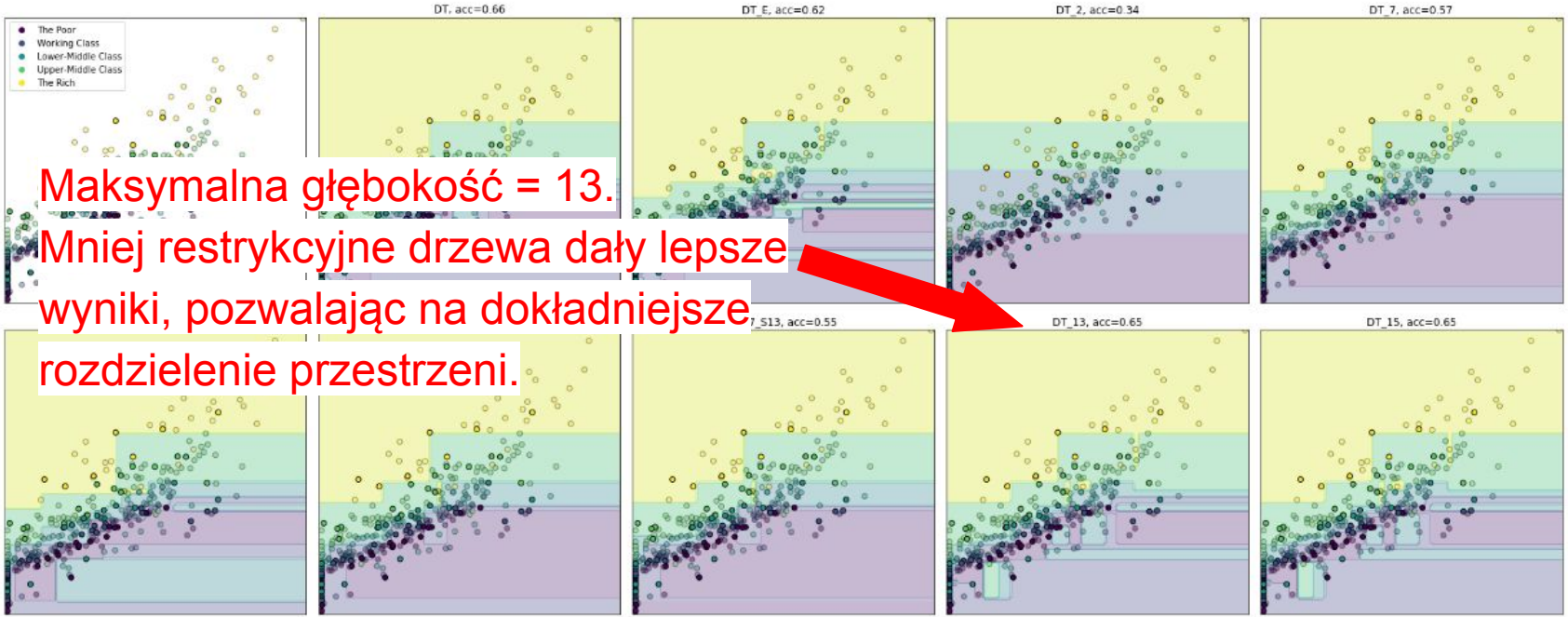
Decision Boundaries (Granice decyzyjne)

Wizualizacja kryteriów klasyfikacji Drzew Decyzyjnych dla cech Limit i Balance.



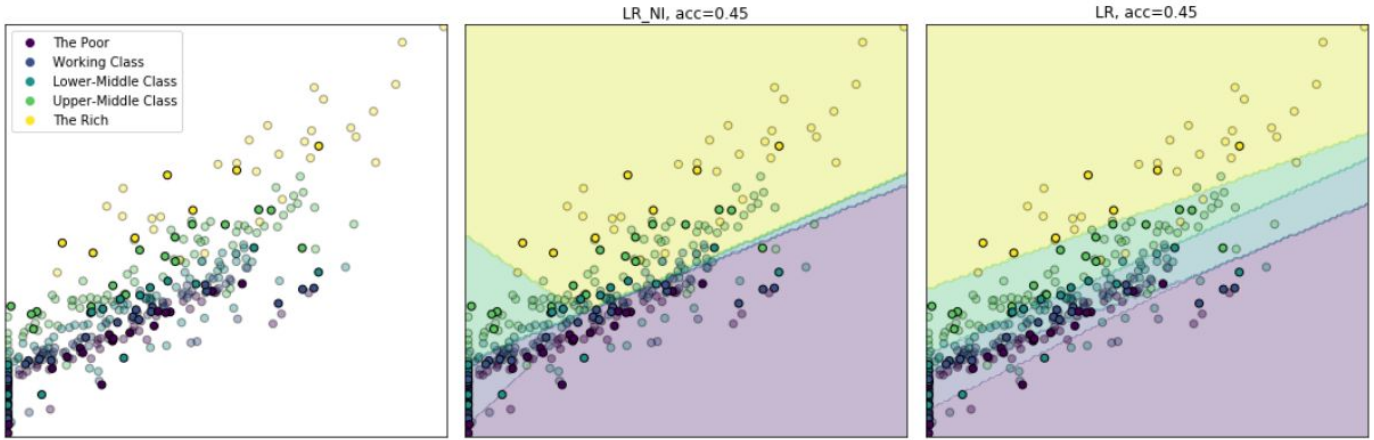
Decision Boundaries (Granice decyzyjne)

Wizualizacja kryteriów klasyfikacji Drzew Decyzyjnych dla cech Limit i Balance.



Decision Boundaries (Granice decyzyjne)

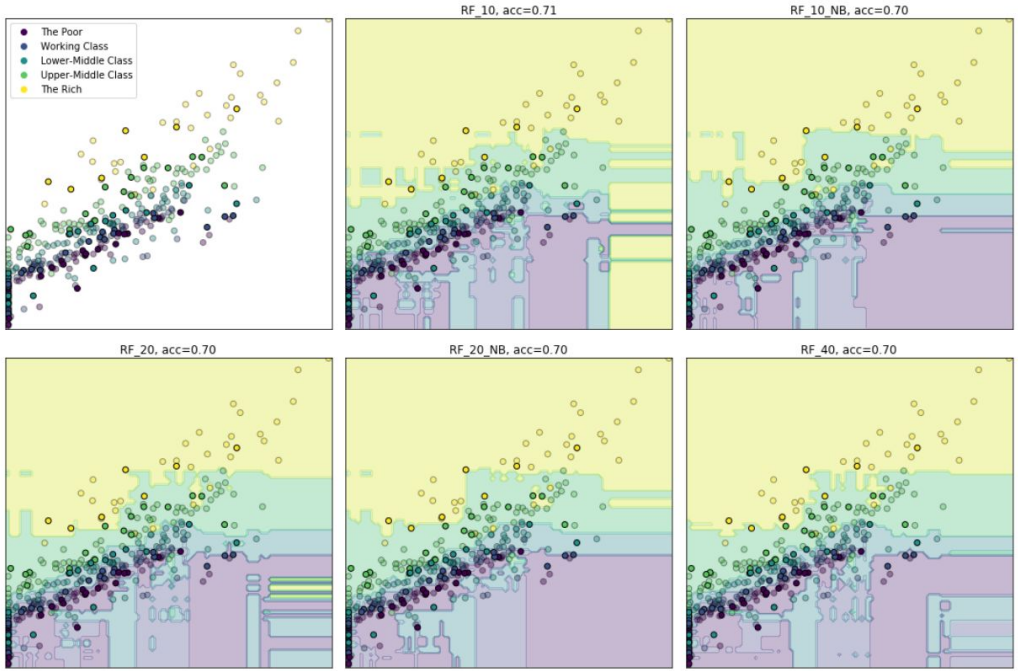
Wizualizacja kryteriów klasyfikacji Regresji Logistycznej dla cech Limit i Balance.



Regresji logistycznej trudno jest dostosować się do klas, które nie są rozdzielne liniowo. Przy dwóch cechach granicą decyzyjną jest linia, wzdłuż której funkcja logistyczna przyjmuje wartości równe 1/2 - jest to wartość progowa w środku funkcji logistycznej w zakresie od 0 do 1.

Decision Boundaries (Granice decyzyjne)

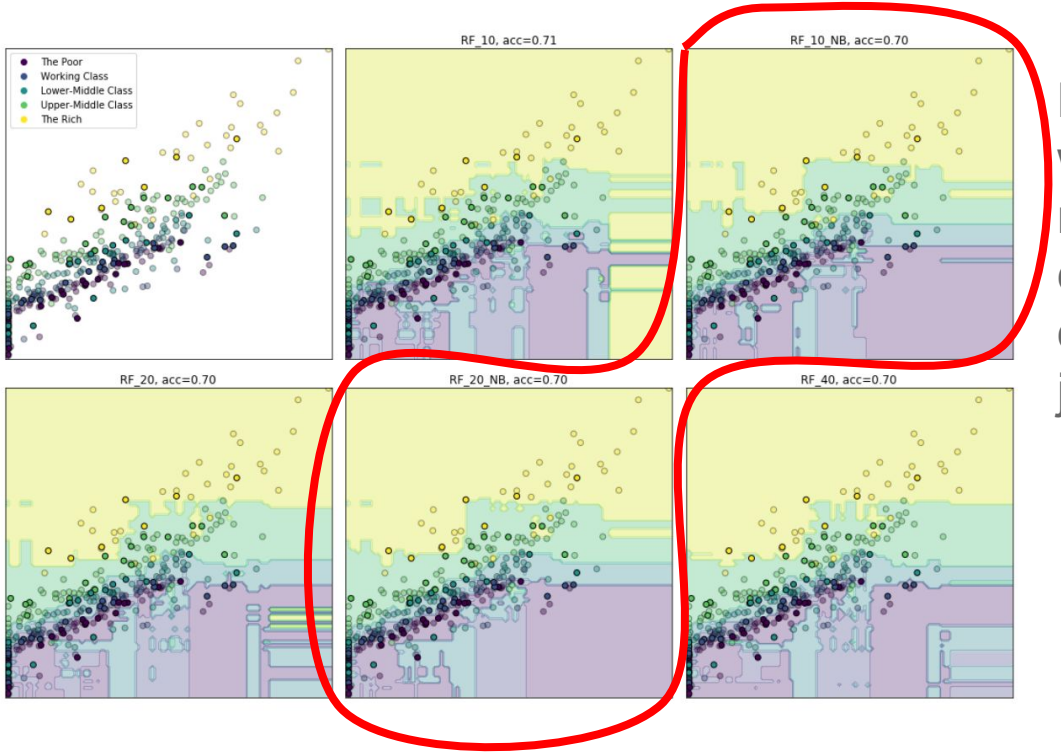
Wizualizacja kryteriów klasyfikacji Lasów Losowych dla cech Limit i Balance.



Widać, że losowe lasy generują wiele oddzielnych regionów.

Decision Boundaries (Granice decyzyjne)

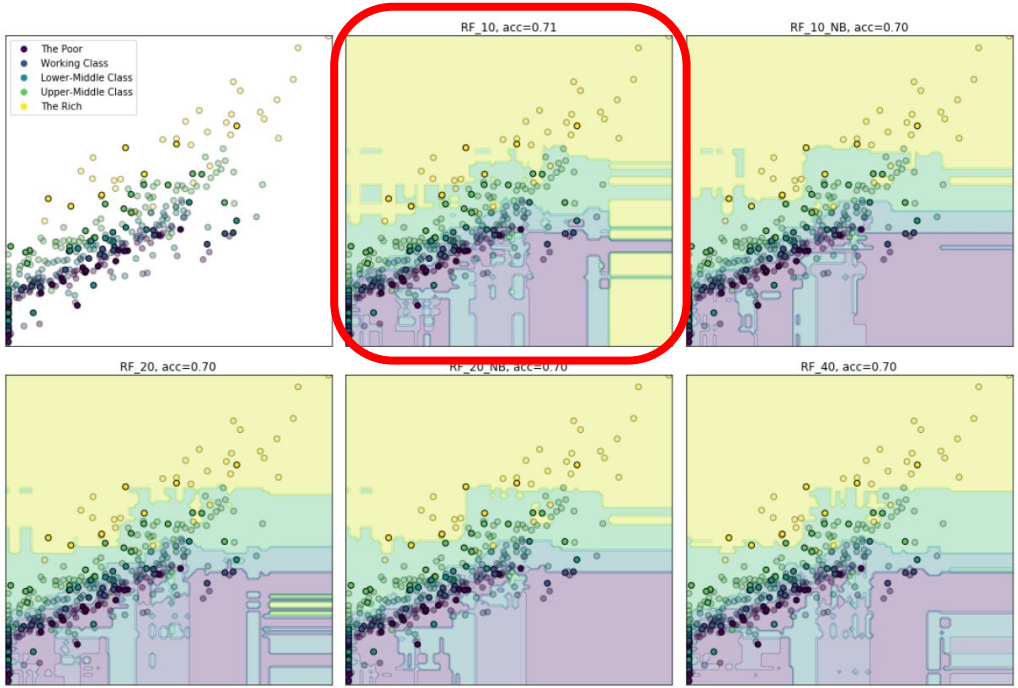
Wizualizacja kryteriów klasyfikacji Lasów Losowych dla cech Limit i Balance.



Losowe lasy bez równoważenia wag (NB) mają granicę decyzji najbardziej podobną do drzew decyzyjnych → mniej podziałów osi X - sugerując, że cecha osi Y jest dominująca.

Decision Boundaries (Granice decyzyjne)

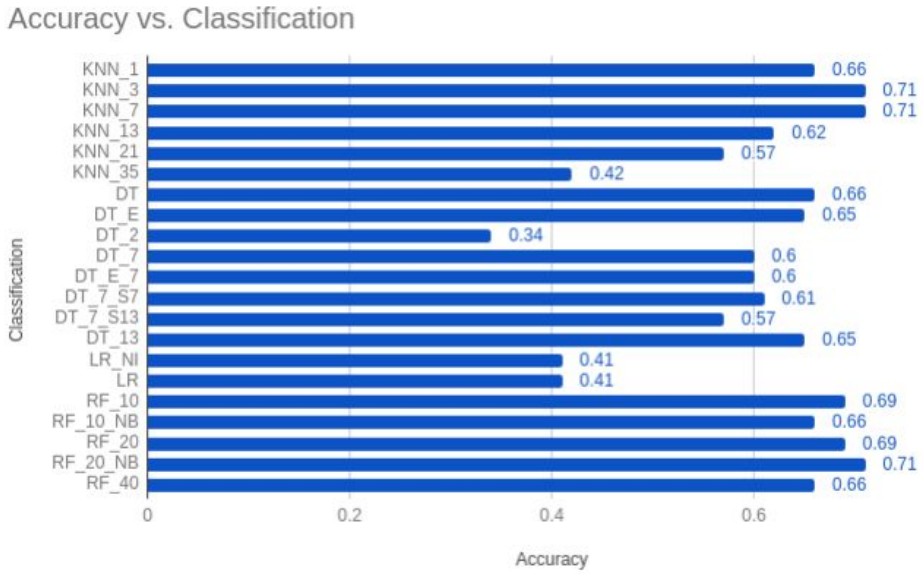
Wizualizacja kryteriów klasyfikacji Lasów Losowych dla cech Limit i Balance.



Jednakże wersja z równoważeniem wag osiąga lepszy wynik, co czyni go najlepszym ze wszystkich klasyfikatorów.

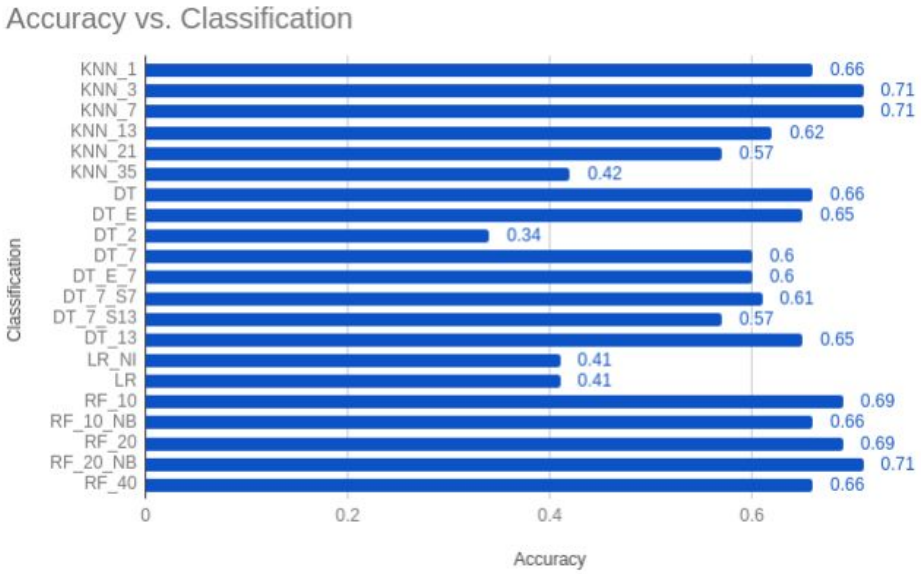
Najlepszy model

Nauczeni doświadczeniem przetestowaliśmy jeszcze raz wszystkie metody klasyfikacji ale w oparciu o tylko 3 najistotniejsze cechy: balance, limit, rating.



Najlepszy model

Nauczeni doświadczeniem przetestowaliśmy jeszcze raz wszystkie metody klasyfikacji ale w oparciu o tylko 3 najistotniejsze cechy: balance, limit, rating.



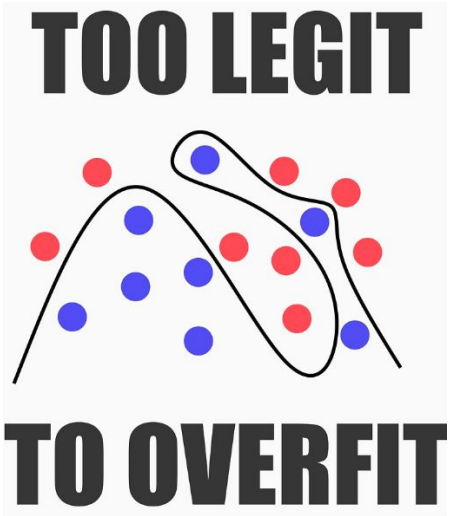
Jak widać najlepiej poradziły sobie klasyfikatory:

→ k-NN dla k=3 oraz 7

→ Random Forest dla 20 drzew, niezbalansowany

Wnioski

Tak wielka poprawa klasyfikacji k-NN pokazuje, że właściwa analiza danych oraz wybór cech może spowodować duże ulepszenie wydajności klasyfikatorów.



One more thing...

Sieci neuronowe (Neural Networks)

Użycie sieci neuronowej bez żadnej selekcji lub analizy daje modelowi około **70%** dokładności.

Używając jedynie wyznaczonych wcześniej najlepszych cech, dokładność sieci neuronowej wzrosła do **82%.**

Zdolność sieci neuronowych do adaptacji do danych i wydobywania ważnych funkcji oraz ich wartości okazuje się być silną przewagą nad zwykłymi metodami statystycznymi, zwłaszcza dla niedoświadczonych „naukowców”.

Dziękujemy za uwagę.