Projekt 2
Perceptron
Algorytmy uczenia maszynowego

Olga Szatkowska

May 21, 2024

Contents

1 Cel projektu, opis problemu

W ramach projektu należało przeprowadzić analizę wybranego zbioru danych oraz dokonać przygotowania pod proces uczenia. Następnie, używając przygotowanego zbioru danych należało dokonać klasyfikacji korzystając z zaimplementowanych wcześniej algorytmów (k najbliższych sąsiadów, jeden vs reszta) oraz 3 wybranych, gotowych modeli z biblioteki scikit-learn.

2 Wizualizacja i opis wybrano zbioru danych

Wybrano zbiór danych dotyczący chorób serca dostępny na stronie UC Irvine Machine Learning Repository. Zbiór zawiera 14 atrybutów

- Wiek pacjenta
- Płeć
- Typ bólu klatki piersiowej
- Ciśnienie krwi w spoczynku w mm Hg na przyjęciu do szpitala
- Stężenie cholesterolu w surowicy w mg/dl
- Poziom cukru we krwi na czczo $>$ 120 mg/dl (1 = prawda; 0 = fałsz)
- Wyniki elektrokardiografii w spoczynku
  - 0 = oznacza normalny
  - 1 = mający nieprawidłowości ST-T (inwersje fali T i/lub uniesienia lub obniżenia ST $>$ 0,05 mV)
  - 2 = wskazujący na prawdopodobne lub pewne przerost lewej komory według kryteriów Estesa
- Maksymalne osiągnięte tętno
- Występowanie bólu dławicowego po wysiłku (1 = tak; 0 = nie)
- Depresja odcinka ST wywołana przez ćwiczenia w odniesieniu do stanu spoczynkowego
- Nachylenie szczytowego odcinka ST podczas wysiłku
  - 1 = w górę
  - 2 = płaski
  - 3 = w dół
- Liczba głównych naczyń (0-3) barwionych kontrastem fluoryzującym
- Status talasemii
  - 3 = normalna
  - 6 = ustalony defekt
  - 7 = nieodwracalny defekt
- Diagnoza choroby serca, gdzie 0 oznacza brak choroby, a wartości od 1 do 4 wskazują na obecność choroby o różnym stopniu nasilenia

2.1 Macierz korelacji

Macierz korelacji przedstawia zależności między dostępnymi w zbiorze danych zmiennym. Wartość bliska -1 oznacza, że im wyższa jest wartość jednej zmiennej, tym niższa dla drugiej, z kolei wynik bliski 1 oznacza, że obie wartości będą rosły lub malały synchronicznie. 0 wskazuje na całkowity brak związku pomiędzy zmiennymi. Na podstawie tego możemy wywnioskować, że większość zmiennych jest ze sobą powiązana. Zmienna **thalach**, czyli maksymalne tętno ma spory wpływ na wartości innych zmiennych.

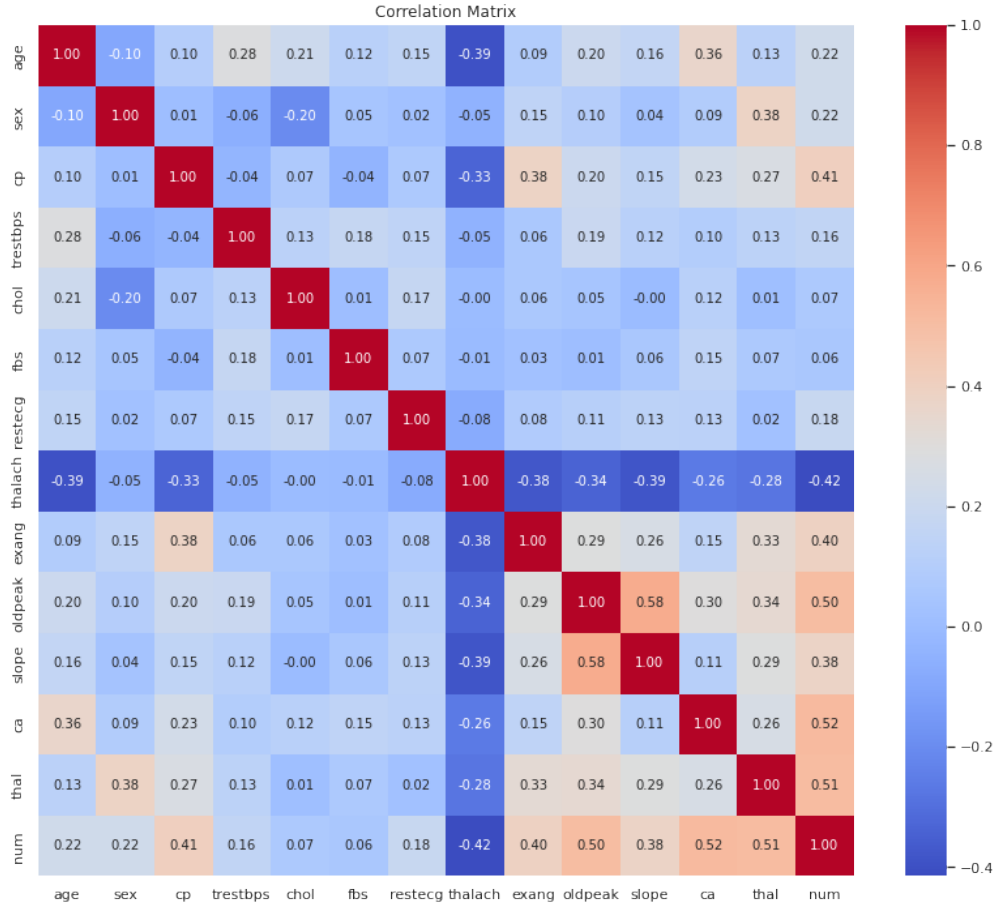


Figure 1: Macierz korelacji

2.2 Zależności między zmiennymi

Wykres pairplot, czyli wykres macierzowy, to narzędzie wizualizacji danych, które pokazuje dystrybucję zmiennych oraz zależności między nimi.

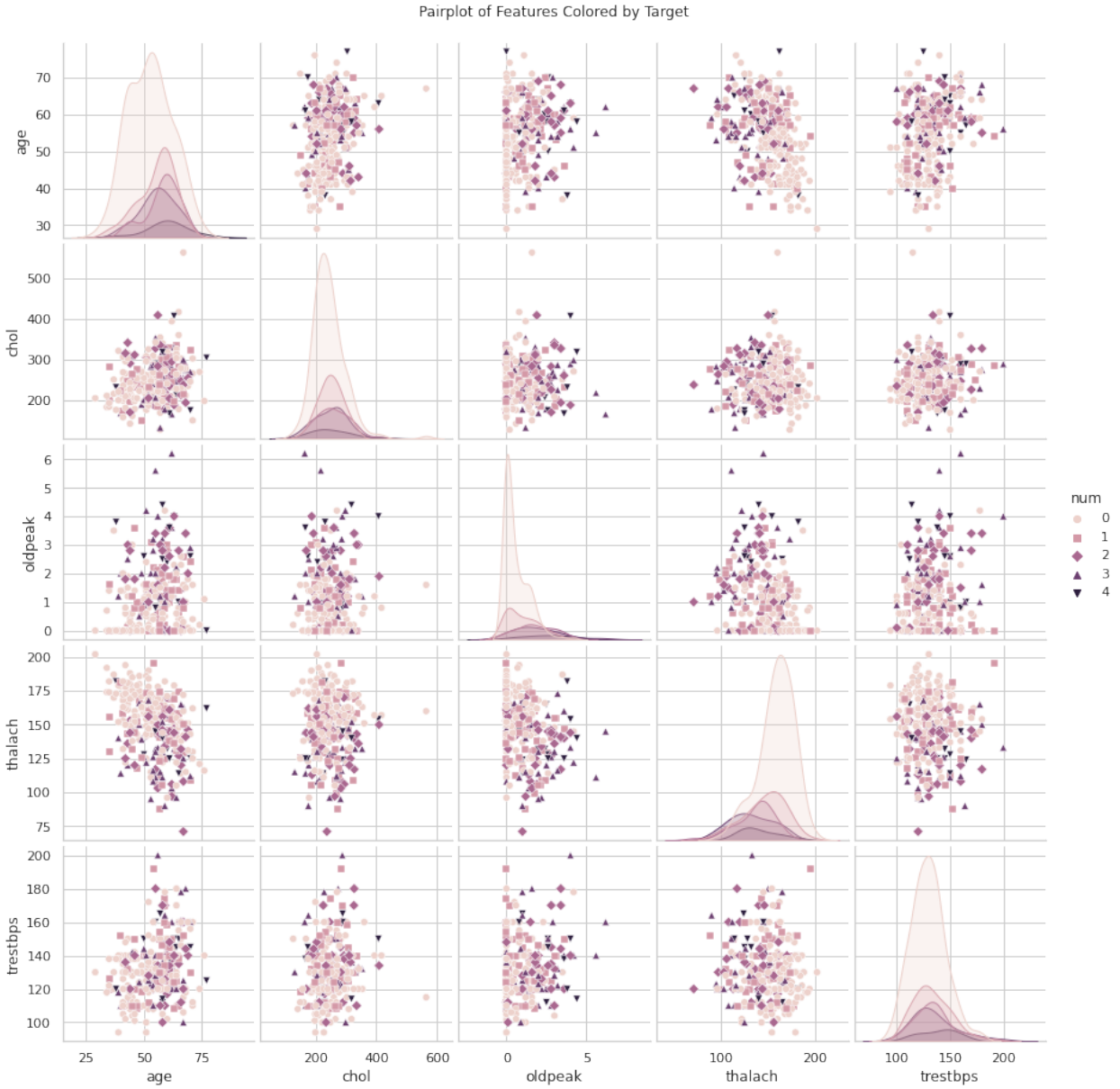


Figure 2: Zależności pomiędzy zmiennymi egzogenicznymi z wyłączenie zmiennych kategoryczny.

2.3 Rozkłady najważniejszych zmiennych

Poniższe wykresy przedstawiają częstotliwość zmiennych maksymalnego tętna oraz wieku w zbiorze danych. Dodatkowo, z ostatniego wykresu wynika, że większość zmiennych docelowych to 0.

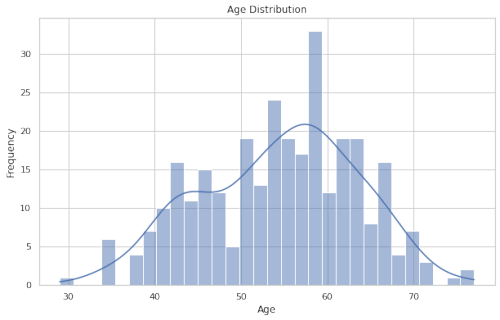


Figure 3: Rozkład wieku

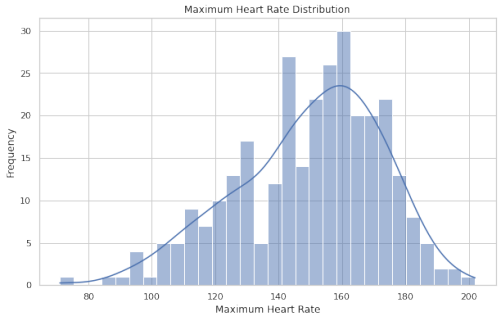


Figure 4: Rozkład maksymalnego tętna

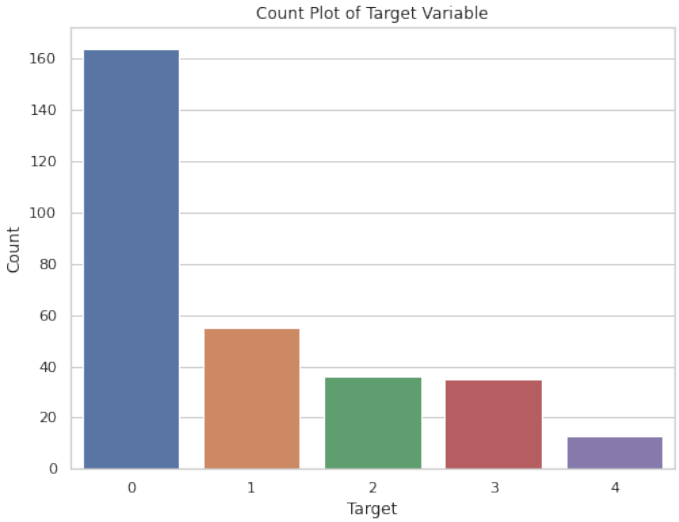


Figure 5: Rozkład zmiennej docelowej

3 Przygotowanie zbioru danych

Przed procesem uczenia należy dostosować zbiór danych. Należy go zeskalować aby uniknąć wartości odbiegających od normy oraz uzupełnić brakjące dane aby nie stracić danych.

3.1 Potok transformujący

W celu przygotowania zbioru danych przygotowano trzy potoki transformujące za pomocą biblioteki scikitlearn oraz modułu preprocessing. Było to konieczne ze względu na różnorodność wartości atrybutów w zbiorze danych. Wartości numeryczne zostały zeskalowane za pomocą skalera standardowego, który przekształca dane tak aby średnia była równa zero oraz brakujące wartości były uzupełnione metodą **mean**, która oblicza średnią wszystkich wartości uzupełnia brakujące wartości. Cechy kategoryczne, w których nie było brakujących wartości zostały pozostawione bez zmian. W przypadku brakujących wartości kategorycznych te zostały uzupełnione metodą **most frequent**, która znajduje najczęsciej występującą wartość i uzupełnia nią braki.

3.2 Zbalansowanie zbioru danych

Ze względu na dużą nierównowagę w zbiorze danych wykorzystano moduł **SMOTENC** z biblioteki imblearn. Zostało wygenerowane ponad 500 nowych próbek na podstawie zbioru danych. Ten moduł on na zdefiniowanie, które atrybuty są kategoryczne.

3.2.1 Rozkład zmiennych po zastosowaniu modułu SMOTENC

Każda klasa ma równy udział w zbiorze danych.

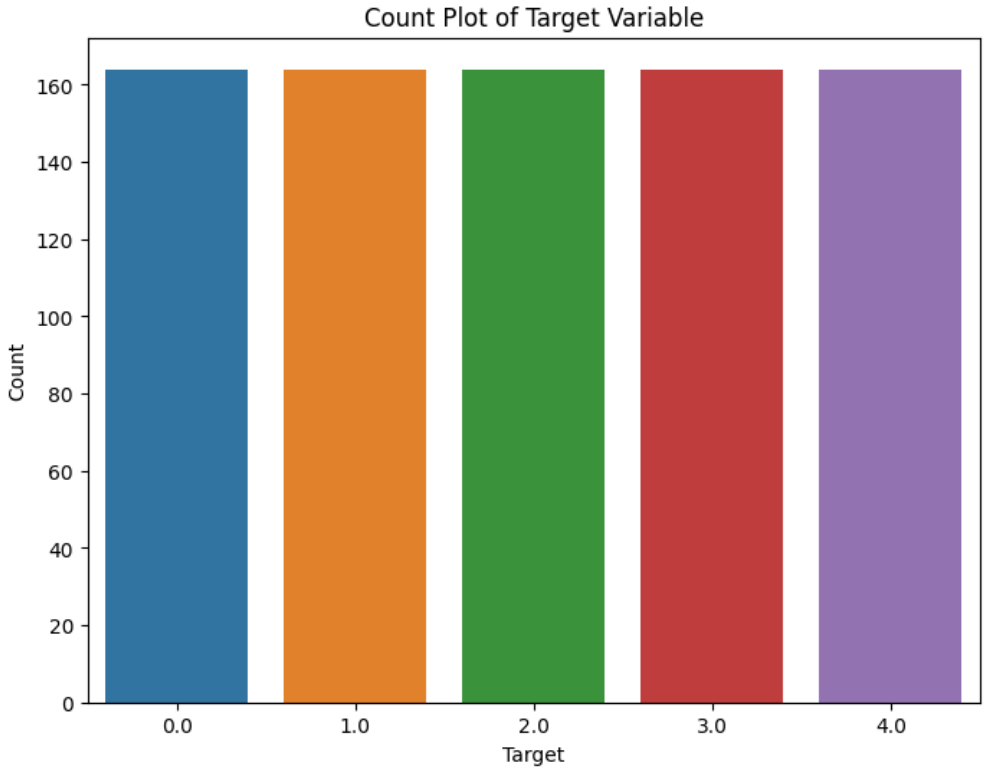


Figure 6: Rozkład zmiennych docelowych po zastosowaniu samplingu

3.2.2 Wykres macierzowy po zastosowaniu modułu SMOTENC

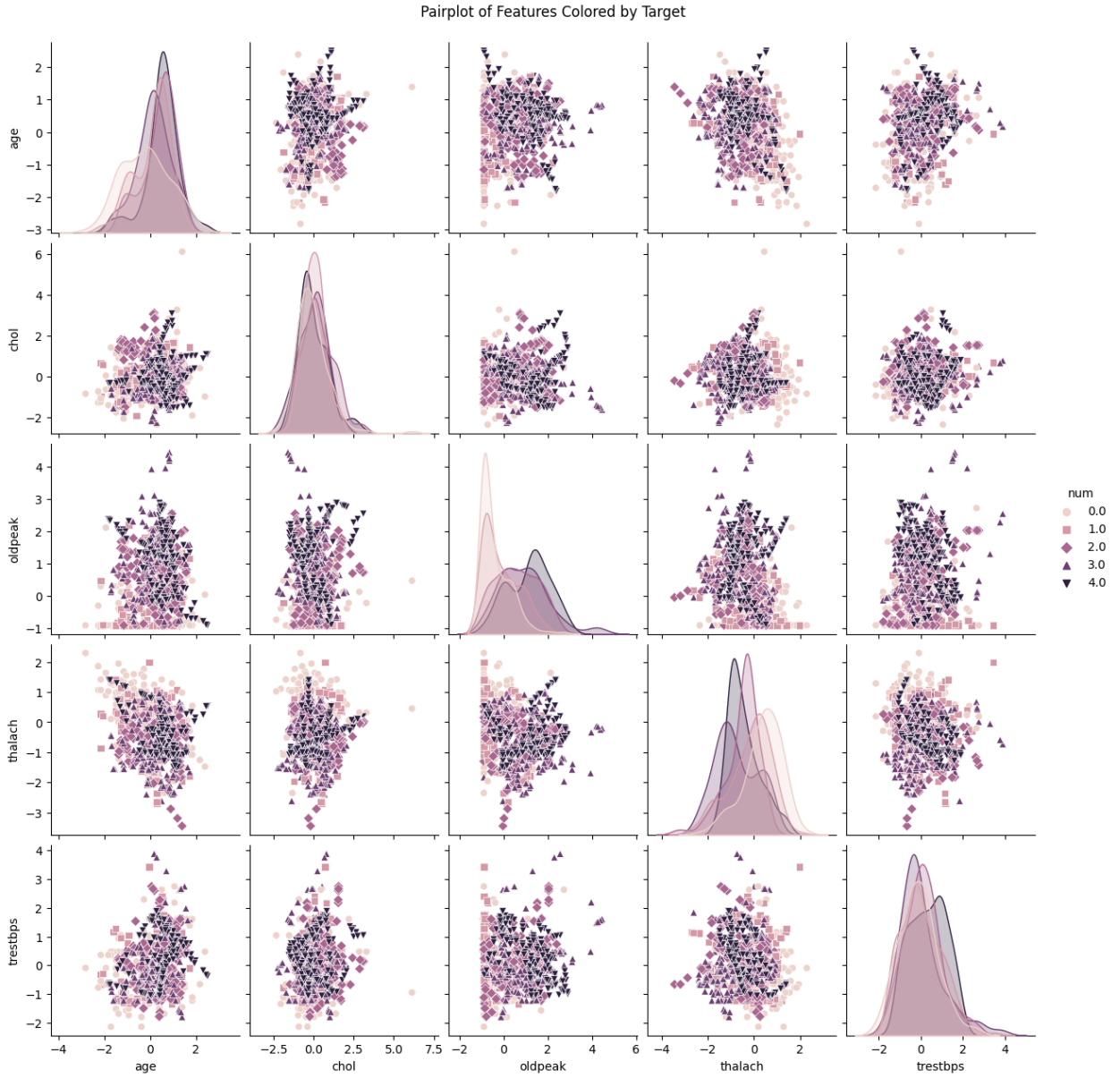


Figure 7: Zależności pomiędzy zmiennymi egzogenicznymi z wyłączenie zmiennych kategoryczny dla zbioru rozszerzonego

3.3 Podział zbioru danych

Zbiór danych podzielono na zbiory treningowy i testowy za pomocą gotowego modułu z biblioteki scikit-learn.

4 Zastosowane algorytmy

Zastosowano łącznie 5 algorytmów. Pierwsze dwa z nich zostały zaimplementowane podczas wcześniejszych zalgorytmów, a 3 pozostałe są gotowymi algorytmami pochodzącymi z biblioteki scikit-learn.

4.1 K najbliższych sąsiadów

Algorytm kNN polega na klasyfikacji na podstawie przestrzeni cech. Wybierane jest K najbliższych sąsiadów, a decyzja o klasie podejmowana jest na podstawię najczęściej pojawiającej się klasy.

4.2 One vs all

One vs all wymaga wyuczenia n modeli w celu klasyfikacji binarnej. Każdy z nich jest w stanie rozpoznać jedną klasę. Klasyfikacji dokonuję się wybierając model, który zaklasyfikował dane jako klasę oraz zwrócił najwyższy **confidence score**. Pozwala na liniową klasyfikację.

4.3 Drzewo decyzyjne

Ten algorytm dzieli zbiór danych na mniejsze grupy zadając pytania w celu klasyfikacji. Przykładem może być, czy atrybut X jest większy od 9. Ta klasyfikacja to suma funkcji nierówności. Korzeń drzewa reprezentuje cały zbiór danych i kierując się cechami, które najlepiej zróżnicują przykłady dzielimy je na mniejsze grupy. W zależności od odpowiedzi na każdym węźle przechodzimy dalej i ten proces jest kontynuowany aż do momentu klasyfikacji.

4.4 Las losowy

Ten algorytm jest rozwinięciem drzew decyzyjnych. Wykorzystuje zbiory drzew decyzyjnych do tworzenia silnego modelu poprzez łączenie słabszych klasyfikatorów. Las losowy tworzy wiele drzew decyzyjnych na podstawie różnych podzbiorów danych treningowych i na końcu łączy ich decyzje, aby uzyskać lepsze wyniki.

4.5 Perceptron wielowarstwowy

Perceptron składający się z warty wejściowej, kilku warstw ukrytych oraz warty wyjściowej. Podczas trenowania dane są przekazywane do sieci. Na końcu obliczany jest błąd klasyfikacji i dane są propagowane wstecz. Wagi zostają zaktualizowane. Dla każdej paczki danych te kroki są powtarzane, aż do uzyskania wystarczających wyników. Pozwala na nieliniową klasyfikację.

5 Wyniki

5.1 Wyniki na podstawie oryginalnego zbioru

5.1.1 Precyzja, trafność (recall), wskaźnik F1

KNN

- Precyzja: 0.2260145122278957
- Trafność: 0.47540983606557374
- Wskaźnik F1: 0.3063752276867031

Perceptron

- Precyzja: 0.5392661982825918
- Trafność: 0.47540983606557374
- Wskaźnik F1: 0.493140945043203

Drzewo decyzyjne

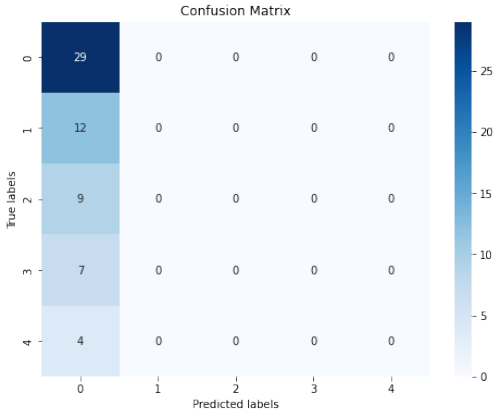
- Precyzja: 0.4816755989911728
- Trafność: 0.5081967213114754
- Wskaźnik F1: 0.4922762698199409

Las losowy

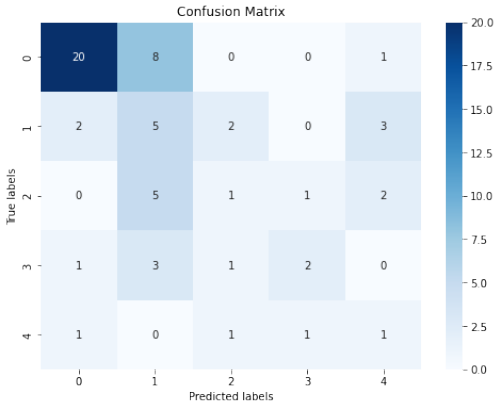
- Precyzja: 0.4045712484237074
- Trafność: 0.5081967213114754
- Wskaźnik F1: 0.44673279147724665

Perceptron wielowarstwowy

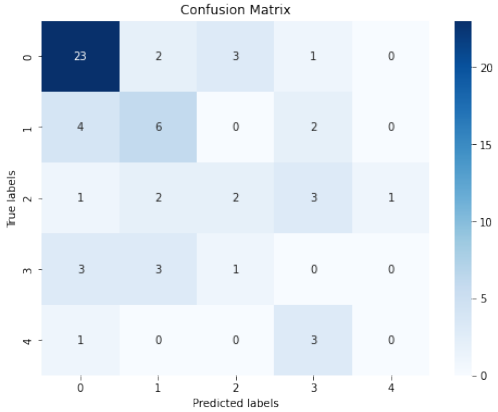
- Precyzja: 0.46147540983606555
- Trafność: 0.5245901639344263
- Wskaźnik F1: 0.4793017333036619



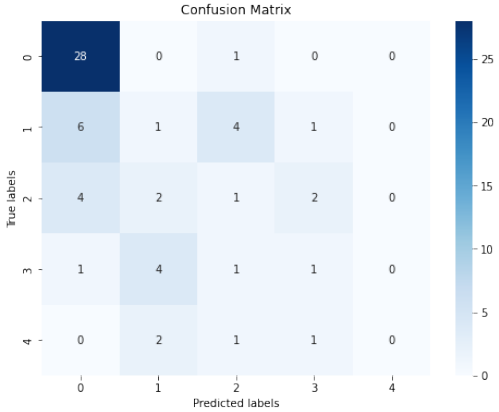
(a) kNN



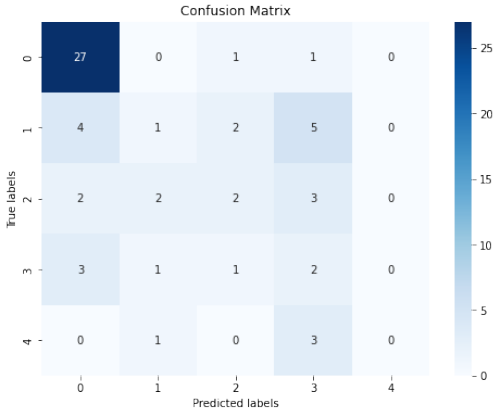
(b) Perceptron



(c) Drzewo decyzyjne



(d) Las losowy



(e) Perceptron wielowarstwowy

Figure 8: Porównanie różnych algorytmów

5.2 Wyniki na podstawie rozszerzonego zbioru

5.2.1 Precyzja, trafność (recall), wskaźnik F1

KNN

- Precyzja: 0.4849918964875923
- Trafność: 0.4268292682926829
- Wskaźnik F1: 0.3858632628819817

Perceptron

- Precyzja: 0.5520333743253398
- Trafność: 0.4573170731707317
- Wskaźnik F1: 0.42204852744500576

Drzewo decyzyjne

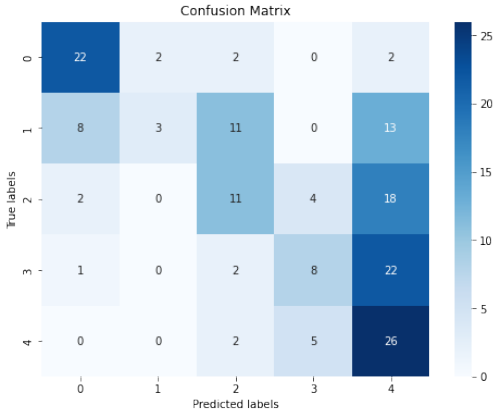
- Precyzja: 0.7259214115252188
- Trafność: 0.7073170731707317
- Wskaźnik F1: 0.7106648621064103

Las losowy

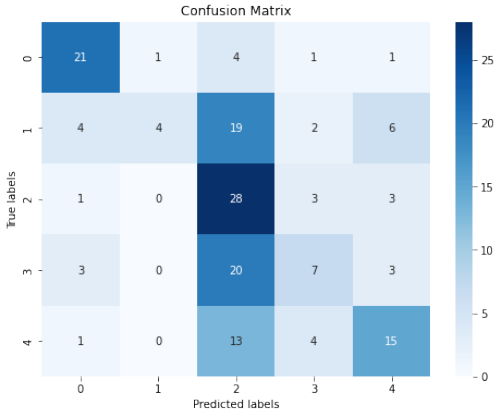
- Precyzja: 0.8056886659842466
- Trafność: 0.8048780487804879
- Wskaźnik F1: 0.8025951944397066

Perceptron wielowarstwowy

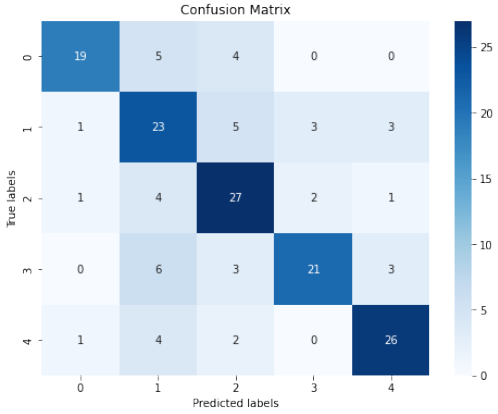
- Precyzja: 0.7354456389958637
- Trafność: 0.7439024390243902
- Wskaźnik F1: 0.7352297230459858



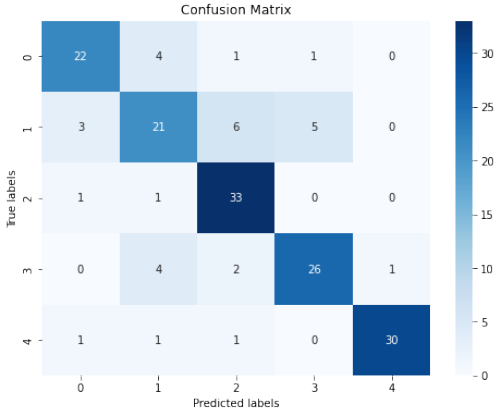
(a) kNN



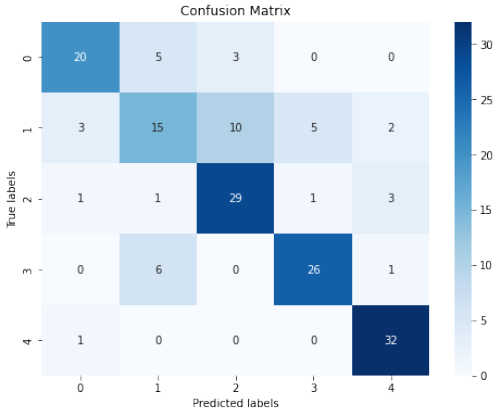
(b) Perceptron



(c) Drzewo decyzyjne



(d) Las losowy



(e) Perceptron wielowarstwowy

Figure 9: Porównanie różnych algorytmów

6 Wnioski

Oryginalny zbiór danych posiada zbyt duży udział klasy 0 aby modele były w stanie dokonać poprawnej klasyfikacji. Dodatkowo, wymaga on nieliniowej klasyfikacji ze względu na to, że większość atrybutów skupia się na średnich wartościach.

6.1 Modele z biblioteki scikit-learn

W przypadku rozszerzonego zbioru danych te modele klasyfikują z precyzją 0.7 - 0.8. Jest to zadowalający wynik. Wynika to z tego, że każdy z modeli jest w stanie dokonać nieliniowej klasyfikacji. Najlepiej radzi sobie perceptron wielowarstwowy ponieważ jest w stanie lepiej dopasować się do nieliniowych relacji w zbiorze dzięki wielu warstwom. Dodatkowo ten model dobrze radzi sobie z większa ilością cech w porównaniu do drzewa decyzyjnego.

6.2 kNN

Precyzja tego modelu oscyluje w okolicach 0.5. Wynika to z tego, że klasy nie tworzą klastrów i są przemieszane. Jest w stanie dobrze rozpoznać jedynie skrajne klasy, które leżą daleko od siebie,

6.3 One-vs-all

Precyzja tego modelu oscyluje w okolicach 0.5. Model klasyfikuje większość przypadków jako klasa 3 co wynika z tego, że klasy nie tworzą wyraźnych klastrów i model nie jest w stanie podzielić ich za pomocą funkcji liniowych.