Sprawozdanie

Indukcyjne Metody Analizy Danych

Ćwiczenie 3. Wybrane metody klasteryzacji w oparciu o system R

Autor: Paweł Mielniczuk

**Spis treści:**

1. Opis działania algorytmów K-Means i PAM
2. Wprowadzenie do zbiorów danych
3. Opis implementacji
4. Analiza wyników
5. Analiza porównania wyników C4.5 i Naiwnego Bayes’a
6. Podsumowanie
7. **Opis działania algorytmów K-means i PAM**

**K-means –** jest to algorytm z dziedziny algorytmów uczenia nienadzorowanego. Stosowany jest do rozwiązywania problemów klasteryzacji. Za zadanie ma na celu utworzenia *k* klastrów, do których zostaną przydzielone i w ten sposób pogrupowane dane wejściowe.

Algorytm na początku ustawia *k* centroidów losowo rozmieszczonych w różnych lokalizacjach, po czym obliczana jest odległość każdej danej wejściowej od każdego z centroidów. Następnie wybierana jest najmniejsza odległość i w ten sposób przypisywana jest zależność do danego centroida.

Następnie dla każdego centroida wyliczany jest nowy środek. Brane są wszystkie dane, które należą do danego centroida i wyliczana jest średnia z ich wektorów. Każdy atrybut jest sumowany z innymi i dzielony przez liczbę instancji należących do tego centroida. Ta nowa średnia mówi o tym w jakie miejsce przesunie się centroid.

Kroki te powtarzane są dopóki żadne z danych wejściowych nie zmienią przypisania do innego klastra.

**K-medoids –** jest to algorytm podobny do k-means i jest pewnym jego ulepszeniem. K-means jest wrażliwe na instancje brzegowe, odbiegające mocno od innych. Wartości takie wpływają znacząco na średnią wartość i cechują się tym, że „przyciągają” do siebie średnią i zaburzają rozkład danych. W przypadku algorytmu PAM zamiast średniej używana jest najbardziej centralna instancja klastra, zwana medoidem.

Na początku zamiast wybierać losowo punkty, które tworzą środki naszych klastrów wybierane są losowo instancje, które stają się medoidami. Tak jak w przypadku k-means przydział do klastra dzieje poprzez wybranie najmniejszej odległości.

Całkowity koszt klasteryzacji obliczany jest jako suma odległości pomiędzy instancjami do ich środka klastra.

Wybranie nowego środka klastra dzieje się poprzez zamianę po kolei każdego z punktów należącego do danego klastra, nie będącego medoidem, z aktualnym medoidem. Jeżeli całkowity koszt klasteryzacji jest mniejszy od poprzedniego wtedy zamiana była dobrym posunięciem, jeżeli nie zmniejszyła się wtedy odwracana jest zamiana.

Jeżeli żaden medoid nie został zamieniony algorytm uważa się za zakończony.

1. **Wprowadzenie do zbiorów danych**

Podczas analizy i implementacji użyte zostały cztery zbiory danych. Zbiory podzielone są na dwie części. Pierwszą z nich są cechy, dokładnie wektor, cech oraz etykiety mówiące o przynależności wektora cech do konkretnej klasy.

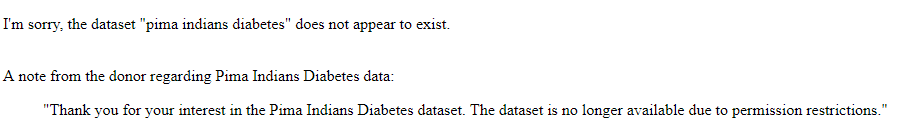
Wszystkie zbiory dostępne są do pobrania ze strony *https://archive.ics.uci.edu/ml/datasets.html*

Zbiory danych zostały ściągnięte i załadowane przy użyciu biblioteki *pandas* lub bezpośrednio załadowane za pomocą biblioteki *scikit-learn*.

Zbiory danych:

* Iris data set
* Wine data set
* Glass identification data set
* Pima diabetes data set

Ciekawostką jest, że w trakcie badania klasyfikatora i tworzenia sprawozdania ostatni ze zbiorów *Pima diabetes* został usunięty ze strony UCI ze przez ograniczenie uprawnień do udostępniania danego zbioru.



Rysunek 1 Wiadomość ze strony https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes mówiąca o braku dalszego dostępu do danego zbioru.

Poniżej zaprezentowano opis zbiorów. Opis ten pomoże w zrozumieniu danych, które będą analizowane. Dobre zrozumienie danych z którymi się pracuje jest niezbędną częścią do poprawnego przeprowadzenia badań.

**Zbiór Iris**

Jest to prawdopodobnie jeden z najbardziej znanych i podstawowych zbiorów danych przy problemach klasyfikacji i rozpoznawania wzorców.

Zbiór składa się ze 150 instancji, podzielonych na 3 równe zbiory po 50 klas każda.

Definicje atrybutów:

* Sepal – zielony płatek u dołu kielicha służący do ochrony kwiatu w trakcie kwitnięcia,
* Petal – płatek kwiatu, służący do przyciągania uwagi ptaków i insektów

Cechy zbioru zawierają cztery informacje:

1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm

Ostatnią, piątą kolumną jest klasa mówiąca o typie irysa. Możliwe są trzy klasy:

1. Iris Setosa
2. Iris Versicolour
3. Iris Virginica

**Zbiór Wine**

Zbiór ten został skonstruowany w wyniku analizy składu chemicznego win stworzonych w tym samym rejonie Włoch lecz przy użyciu trzech różnych odmian uprawnych.

Zbiór składa się ze 178 instancji.

Definicje atrybutów oraz cechy zbioru:

1. Alcohol – alkohol
2. Malic acid – kwas jabłkowy
3. Ash – popiół
4. Alkalinity of ash – alkaliczność popiołu
5. Magnesium – magnez
6. Total phenols – całkowita zawartość fenoli
7. Flavonoids – flawonoidy
8. Nonflavanoid phenols – fenole nieflawonowe
9. Proanthocyanidins – proantocyjanidyny
10. Color intensity, intensywność koloru
11. Hue – odcień
12. OD280/OD315 of diluted wines - OD280 / OD315 rozcieńczonych win
13. Proline – Proline

Pierwszy atrybut w pliku zawierającym dane jest identyfikatorem klasy od 1 do 3.

Rozłożenie instancji klas jest następujące:

* Klasa 1 – 59 instancji,
* Klasa 2 – 71 instancji,
* Klasa 3 – 48 instancji.

**Zbiór Glass identification**

Zbiór powstał poprzez analizę składu chemicznego badanego szkła aby określić typ powstałego szkła oraz jego przeznaczenie.

Zbiór składa się z 214 instancji podzielonych na 6 klas.

Rozłożenie instancji klas jest następujące:

* Klasa 1 – 70 instancji,
* Klasa 2 – 76 instancji,
* Klasa 3 – 17 instancji,
* Klasa 4 - 13,
* Klasa 5 - 9,
* Klasa 6 - 29.

Definicje atrybutów oraz cechy zbioru:

1. Id – numer porządkowy
2. Refractive index – współczynnik załamania światła
3. Sodium – sód
4. Magnesium – magnez
5. Aluminium – glin
6. Silicon – krzem
7. Potassium – potas
8. Calcium – wapń
9. Barium – bar
10. Iron – żelazo

**Zbiór Seeds**

Zbiór reprezentuje atrybuty 3 różnych typów zbóż.

Zbiór składa się z 210 instancji podzielonych na 3 klasy.

Rozłożenie instancji klas jest następujące:

* Klasa 1 (Kama) – 70 instancji,
* Klasa 2 (Rosa) – 70 instancji,
* Klasa 3 (Canadian) – 70 instancji,

Definicje atrybutów oraz cechy zbioru. Wszystkie atrybuty są miarami nasion zboża:

1. Area – pole
2. Perimeter – obwód
3. Compactness – ścisłość
4. Length of kernel – długość nasiona
5. Width of kernel – szerokość nasiona
6. Asymmetry coefficient – współczynnik asymetrii
7. Length of kernel groove - długość rowka nasiona

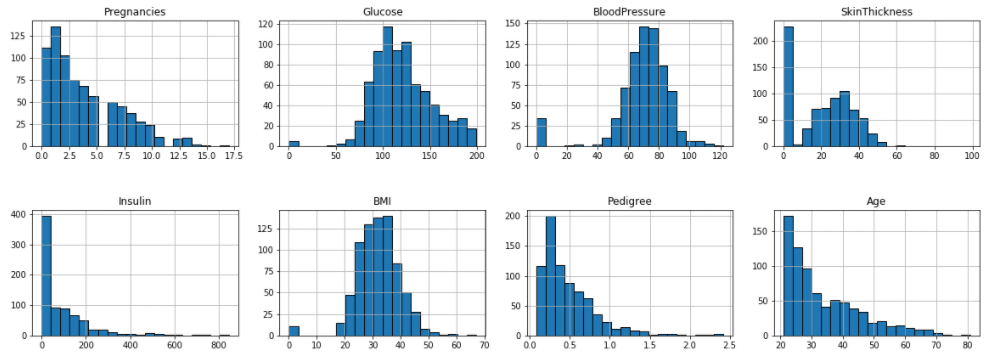
**Zbiór Pima diabetes**

Celem zbioru jest umożliwienie zdiagnozowania czy dany pacjent ma cukrzycę, bazując na diagnostykach zamieszczonych w cechach zbioru. Wszyscy pacjenci przebadani byli kobietami mającymi przynajmniej 21 lat oraz byli pochodzenia indiańskiego plemienia Pima.

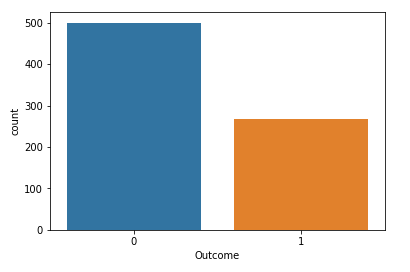
Zbiór składa się z 768 instancji posiadających dwie możliwe klasy 1 – oznaczające że zbadana osoba jest chora na cukrzycę, 0 – oznaczające że dana osoba nie jest chora na cukrzycę.

Definicje atrybutów oraz cechy zbioru:

1. Pregnancies – liczba ciąży
2. Glucose – poziom glukozy
3. Blood ressure – ciśnienie krwi
4. Skin thickness – grubość skóry
5. Insulin – poziom insuliny
6. BMI – body mass index
7. Diabetes pedigree function – funkcja pedigree
8. Age – wiek



Rysunek Histogramy atrybutów danych zbioru Pima diabetes



Rysunek Rozkład klas

1. **Implementacja**

Do implementacji użyty został skryptowy język programowania R oraz następujące biblioteki: Rweka, caret.

**Tworzenie modelu:**

Do stworzenia modelu drzewa została użyta metoda *train* z biblioteki *caret*. Metoda ta posiada wiele parametrów, po części opcjonalnych. Najważniejszymi jakie zostały użyte podczas tworzenia drzewa C4.5 to:

1. X – zbiór argumentów
2. Y – wektor klas
3. Method – metoda według, której ma zostać wytrenowany model
4. tuneLength – liczba różnych wartości, które będą wypróbowane dla każdego parametru algorytmu
5. trControl – obiekt, który mówi o tym w jaki sposób mają być obliczane wyniki, tutaj użyte w celu określenia krosswalidacji i liczby foldów.

Model ten zakłada, że atrybuty będą zawsze w rozkładzie normalnym i działa wtedy najlepiej. Jak widać na powyższych wykresach nie jest to jednak zawsze prawdą.

**Cross-validation:**

Często zdarza się, że mamy do czynienia z małymi zbiorami danych. Gdy taki zbiór podzielimy na zbiór treningowy i testowy, a czasem jeszcze walidacyjny nasze dane stają się zbyt małe aby poprawnie wyuczyć model. W takich przypadkach należy podjąć pewne kroki aby zapewnić, że wielkość zbioru będzie odpowiednio duża do wytrenowania modelu.

Jedną z nich jest kroswalidacja. Polega ona na podziale całego zbioru na określoną ilość podzbiorów, a następnie przeprowadzeniu na nich predykcji jak celny będzie nasz model.

Jednymi z podstawowych rodzajów walidacji modelu są:

* **K-Fold validation** – zbiór dzielony jest na K części. Następnie kolejno każdy z podzbiorów brany jest jako zbiór testowy, a pozostałe jako uczący. Analiza jest wykonywana tyle razy na ile części został podzielony zbiór. Po czym wszystkie wyniki się sumuje i uśrednia.
* **Stratyfikowana K-Fold validation** – zasada działania jest taka sama jak w przypadku zwykłego K-Fold validation z dodatkowym zachowaniem oryginalnych proporcji między klasami (labels) w podzielonych zbiorach.

Oprócz accuracy wyliczono także F1 score. Jest to średnia ważona precyzji oraz miary recall.

F1 = 2 \* (precision \* recall) / (precision + recall)

**Walidacja K-Fold**

Biblioteka caret pozwala na łatwe tworzenie kroswalidacji zwykłej poprzez metodę *trainControl*. Tam specyfikujemy metodę, która będzie użyta w kontrolowaniu uczena modelu, w przypadku kroswalidacji jest to „cv”. Dodatkowo ustalamy liczbę foldów za pomocą parametru *number*. Ostatnim z ważnych argumentów metody jest *summaryFunction*, w którym określamy po jakiej funkcji będzie liczona dokładność modelu.

**Stratyfikowana kroswalidacja**

Do policzenia dokładności drzewa użyta została kroswalidacja oraz kroswalidacja stratyfikowana. Jednakże biblioteka caret nie umożliwia w prosty sposób określenia kroswalidacji jako stratyfikowanej. Należy posłużyć się metodą *createFolds* gdzie określimy foldy oraz pamateru *index* z metody trainControl gdzie podamy nasze foldy i dzięki temu uzyskamy kroswalidację stratyfikowaną



Rysunek 4 Kroswalidacja stratyfikowana

**Parametry wykorzystane w analizie drzewa**

**Tune length –** jeden z automatycznych sposobów regulowania i dostrajania modelu w bibliotece Caret. Poprzez ustawienie parametru *tuneLength*, który przyjmuje jedynie liczby całkowite, ustalamy jaka liczba różnych wartości będzie użyta w każdym z hiperparametrów funkcji.

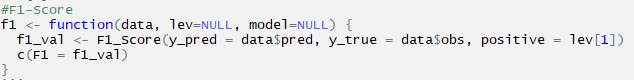
**Pruning confidence factor –** przyjmuje wartości w zakresie (0, 0.5>. Mała wartość tego parametru odpowiada za duży pruning, natomiast wysoka za niski. Wartości z zakresu (0.5,1) są dozwolone lecz nie spowodują żadnego pruningu. Wartość tego parametru jest używana do obliczenia górnej granicy błędu możliwego do posiadania w węźle lub liściu. Domyślna wartość to 0.25.

**Minimum number of instances –** parametr, który odpowiada za minimalną liczbę obserwacji, które muszą dotrzeć do liścia. Poprzez zwiększenie tej wartości zmniejszamy wielkość drzewa, a także zmiejszamy liczbę liści w drzewie.

**Dodatkowe parametry użyte podczas implementacji**

**TrainControl** *–* jest to jeden z podstawowych parametrów odpowiadający za to w jaki sposób kontrolujemy trenowanie modelu. Między innymi odpowiada za to jaką funkcję (accuracy, f1, precision) użyjemy do obliczania poprawności modelu.

**SummaryFunction** – parametr metody *trainControl* odpowiedzialny za określanie stopnia nauczenia modelu. Biblioteka Caret posiada jedynie dwie podstawowe funkcje (accuray i kappa). Nas interesuje jednak głównie F1-Score, na szczęscie możliwe jest napisanie własnej implementacji metody i jej przekazanie w tym parametrze.



Rysunek 5 Funkcja F1-Score

1. **Analiza wyników drzewa decyzyjnego**

W poniższych analizach parametrów, gdy parametry nie były zmieniane ich domyślne wartości wynoszą:

folds = 10,

tune length = 5,

confidence factor = 0.25,

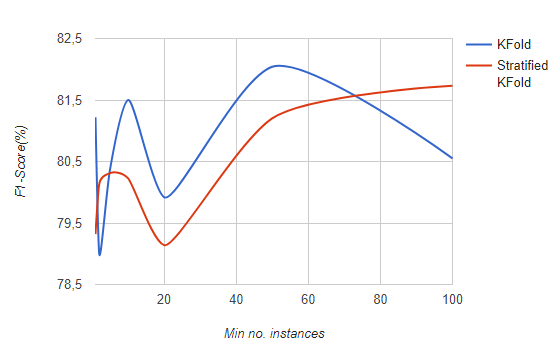
min no. Leaves = 2.

**Analiza Pima diabetes dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Min no. leaves** | **Liczba liści** | **Wielkość drzewa** | **Głębokość drzewa** | **Kfold F1-Score (%)** | **Stratified Kfold F1-Score(%)** |
| **1** | 24 | 47 | 10 | 81.22 | 79.32 |
| **2** | 20 | 39 | 9 | 79.02 | 80.12 |
| **5** | 13 | 25 | 8 | 80.36 | 80.31 |
| **10** | 15 | 29 | 7 | 81.50 | 80.23 |
| **20** | 10 | 19 | 5 | 79.92 | 79.14 |
| **50** | 9 | 17 | 5 | 82.04 | 81.20 |
| **100** | 3 | 5 | 2 | 80.55 | 81.73 |

Tabela Min no. leaves dla zbioru danych Pima diabetes

Wyniki kroswalidacji stratyfikowanej i kroswalidacji zwykłej były podobne, jednakże przy większej generalizacji modelu stratyfikowana kroswalidacja radziła sobie lepiej.

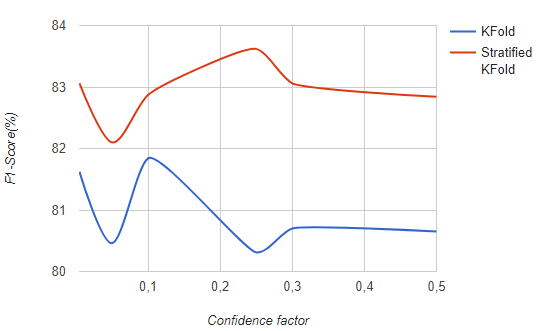


Rysunek Wykres F1-Score i Min no. leaves dla zbioru Pima diabetes

|  |  |  |
| --- | --- | --- |
| **Confidence factor** | **Kfold F1-Score (%)** | **Stratified Kfold F1-Score (%)** |
| 0.005 | 81.62 | 83.06 |
| 0.05 | 80.46 | 82.10 |
| 0.1 | 81.84 | 82.87 |
| 0.25 | 80.31 | 83.62 |
| 0.3 | 80.70 | 83.06 |
| 0.5 | 80.65 | 82.84 |

Tabela Confidence factor dla zbioru danych Pima diabetes

W przypadku badania confidence factor kroswalidacja stratyfikowana była zdecydowanie lepsza we wszystkich przypadkach i dawała zazwyczaj 1-2% lepsze wyniki.



Rysunek Wykres F1-Score i Confidence factor dla zbioru Pima diabetes

Najlepsze wyniki dla zbioru Pima diabetes uzyskano przy następujących parametrach:

KFold

|  |  |
| --- | --- |
| Min no. Instances | 50 |
| Confidence factor | 0.005 |
| **F1-Score** | **82.78%** |

Tabela Najlepszy winik przy kroswalidacji Kfold dla zbioru Pima diabetes

Stratified KFold

|  |  |
| --- | --- |
| Min no. Instances | 100 |
| Confidence factor | 0.5 |
| **F1-Score** | **83.25%** |

Tabela Najlepszy winik przy kroswalidacji Stratyfikowanej Kfold dla zbioru Pima diabetes

**Porównania najlepszych wyników C4.5 i Naiwnego Bayes’a**

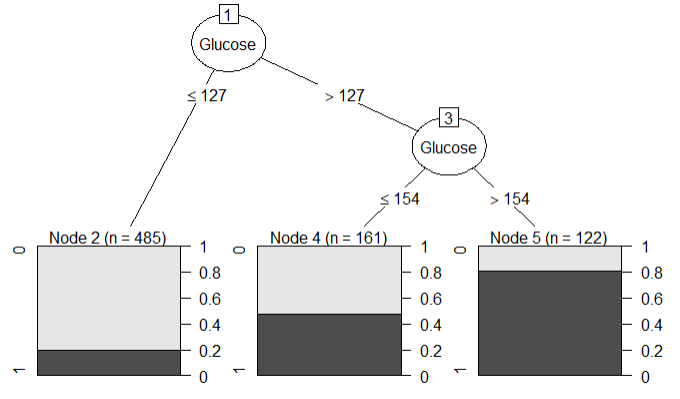
|  |  |  |
| --- | --- | --- |
| Model | C4.5 | Naive Bayes |
| Kroswalidacja | Stratyfikowana | Stratyfikowana |
| F1-Score | 83% | 75% |

Tabela Porównanie najlepszych wyników modelów C4.5 i Naive Bayes dla zbioru Pima diabetes

Najlepsze parametry w modelu C4.5 ustawione na M: 50, CF: 0.5.

Najlepsze parametry w modelu Naive Bayes ustawione na Dyskretyzacja: brak.

W przypadku porównania modelu drzewa decyzyjnego i modelu Naiwnego Bayes’a widać zdecydowaną przewagę modelu C4.5. Najlepsze wyniki uzyskane przed model C4.5 były lepsze o około aż 8% od modelu Naive Bayes.



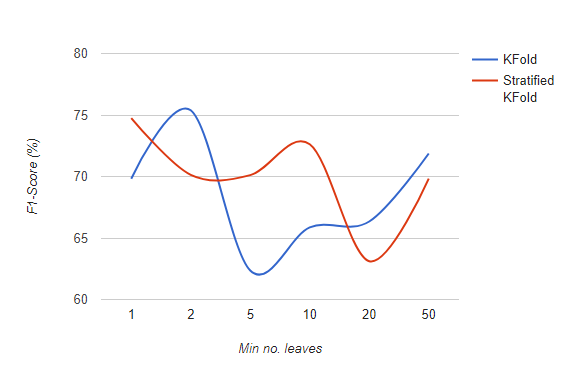
Rysunek Wizualizacja najlepszego modelu drzewa C4.5 dla zbioru Pima diabetes

**Analiza Glass dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Min no. leaves** | **Liczba liści** | **Wielkość drzewa** | **Głębokośc drzewa** | **Kfold F1-Score (%)** | **Stratified Kfold F1-Score (%)** |
| **1** | 27 | 53 | 9 | 69.82 | 74.75 |
| **2** | 30 | 59 | 10 | 75.38 | 70.13 |
| **5** | 16 | 31 | 8 | 62.36 | 70.11 |
| **10** | 9 | 17 | 6 | 65.86 | 72.62 |
| **20** | 4 | 7 | 3 | 66.34 | 63.11 |
| **50** | 3 | 5 | 2 | 71.87 | 69.83 |
| **100** | N/A | N/A | N/A | N/A | N/A |

Tabela Min no. leaves dla zbioru danych Glass

W przypadku zbioru Glass wyniki obu kroswalidacji były bardzo zbliżone do siebie. W przypadku dużej generalizacji przy zwiększeniu minimalnej liczby obserwacji w liściu celność modelu spadła ponad dwukrotnie. Jednakże przy małej generalizacji model osiągał bardzo wysokie wyniki.

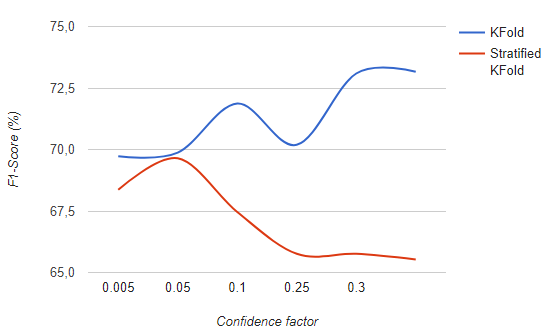
****

Rysunek Wykres F1-Score i Min no. leaves dla zbioru Glass

|  |  |  |
| --- | --- | --- |
| **Confidence factor** | **Kfold F1-Score (%)** | **Stratified Kfold F1-Score (%)** |
| 0.005 | 69.73 | 68.36 |
| 0.05 | 69.87 | 69.64 |
| 0.1 | 71.87 | 67.46 |
| 0.25 | 70.19 | 65.76 |
| 0.3 | 73.09 | 65.76 |
| 0.5 | 73.16 | 65.53 |

Tabela 7 Confidence factor dla zbioru danych Glass

W przypadku analizy confidence factor dla zbioru Glass uzyskano ciekawe wyniki. Kroswalidacja stratyfikowana była cały czas na tym samym poziomie 97.70%.

****

Rysunek 8 Wykres F1-Score i Confidence factor dla zbioru Glass

Najlepsze wyniki dla zbioru Glass uzyskano przy następujących parametrach:

KFold

|  |  |
| --- | --- |
| Min no. Instances | 1 |
| Confidence factor | 0.3 |
| **F1-Score** | **78.12** |

Tabela Najlepszy winik przy kroswalidacji Kfold dla zbioru Glass

Stratified KFold

|  |  |
| --- | --- |
| Min no. Instances | 10 |
| Confidence factor | 0.005 |
| **F1-Score** | **71.94** |

Tabela Najlepszy winik przy kroswalidacji stratyfikowanej Kfold dla zbioru Glass

**Porównania najlepszych wyników C4.5 i Naiwnego Bayes’a**

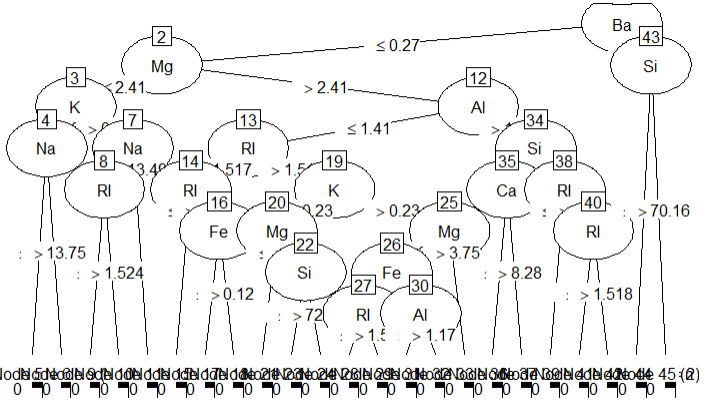
|  |  |  |
| --- | --- | --- |
| Model | C4.5 | Naive Bayes |
| Kroswalidacja | Kfold | Stratyfikowana |
| F1-Score | 78% | 57% |

Tabela Porównanie najlepszych wyników modelów C4.5 i Naive Bayes dla zbioru Glass

Najlepsze parametry w modelu C4.5 ustawione na M: 1, CF: 0.3.

Najlepsze parametry w modelu Naive Bayes ustawione na Dyskretyzacja: brak.

Wyniki modelu C4.5 znów przewyższały wyniki modelu Naiwnego Bayes’a. Jednakże wyniki były do siebie zbliżone. Kroswalidacja zwykła i stratyfikowana dawały podobne wyniki.



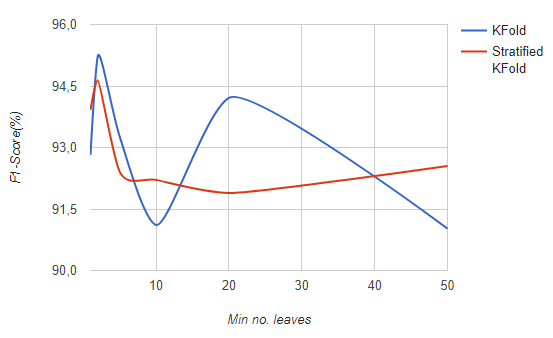
Rysunek Wizualizacja najlepszego modelu drzewa C4.5 dla zbioru Glass

**Analiza Wine dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Min no. leaves** | **Liczba liści** | **Wielkość drzewa** | **Głębokość drzewa** | **Kfold F1-Score (%)** | **Stratified Kfold F1-Score (%)** |
| **1** | 5 | 9 | 3 | 92.82 | 93.92 |
| **2** | 5 | 9 | 3 | 95.23 | 94.64 |
| **5** | 5 | 9 | 3 | 93.27 | 92.41 |
| **10** | 4 | 7 | 2 | 91.11 | 92.21 |
| **20** | 4 | 7 | 2 | 94.21 | 91.89 |
| **50** | 3 | 5 | 2 | 91.02 | 92.55 |
| **100** | N/A | N/A | N/A | N/A | N/A |

Tabela Min no. leaves dla zbioru danych Wine

W przypadku analizy zbioru Wine zwiększenie liczby minimalnych obserwacji do 100 w liściach nie było możliwe. Wyniki kroswalidacji zwykłej i stratyfikowanej były do siebie zbliżone, jednakże przy większej generalizacji kroswalidacja stratyfikowana rosła, a zwykła malała.

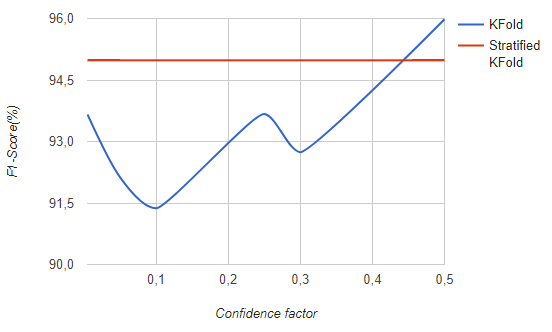


Rysunek Wykres F1-Score i Min no. leaves dla zbioru Wine

|  |  |  |
| --- | --- | --- |
| **Confidence factor** | **Kfold F1-Score (%)** | **Stratified Kfold F1-Score (%)** |
| 0.005 | 93.66 | 94.98 |
| 0.05 | 92.13 | 94.98 |
| 0.1 | 91.37 | 94.98 |
| 0.25 | 93.67 | 94.98 |
| 0.3 | 92.74 | 94.98 |
| 0.5 | 95.98 | 94.98 |

Tabela Confidence factor dla zbioru danych Wine

Podobnie jak w przypadku analizy Confidence factor dla zbioru danych Glass w przypadku zbioru Wine zaobserwowano podobne rezultaty przy kroswalidacji stratyfikowanej uzyskiwano cały czas takie same wyniki.



Rysunek 11 Wykres F1-Score i Confidence factor dla zbioru Wine

Najlepsze wyniki dla zbioru Wine uzyskano przy następujących parametrach:

KFold

|  |  |
| --- | --- |
| Min no. Instances | 2 |
| Confidence factor | 0.1 |
| **F1-Score** | **97.42** |

Tabela Najlepszy winik przy kroswalidacji Kfold dla zbioru Wine

Stratified KFold

|  |  |
| --- | --- |
| Min no. Instances | 1 |
| Confidence factor | 0.005 |
| **F1-Score** | **94.53** |

Tabela Najlepszy winik przy kroswalidacji Stratyfikowanej Kfold dla zbioru Wine

**Porównania najlepszych wyników C4.5 i Naiwnego Bayes’a**

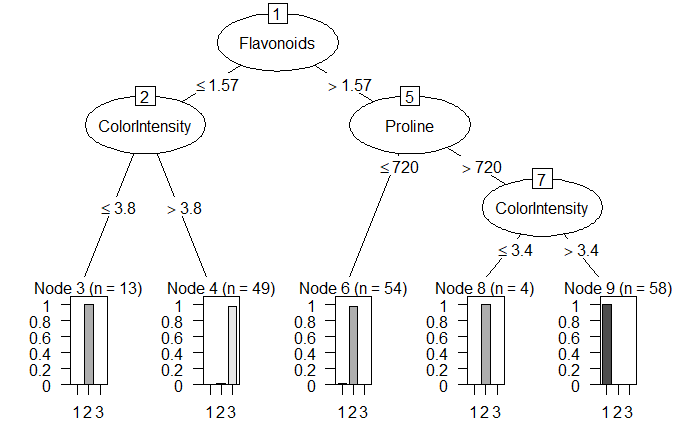
|  |  |  |
| --- | --- | --- |
| Model | C4.5 | Naive Bayes |
| Kroswalidacja | Kfold | Stratyfikowana |
| F1-Score | 97% | 95% |

Tabela Porównanie najlepszych wyników modelów C4.5 i Naive Bayes dla zbioru Wine

Najlepsze parametry w modelu C4.5 ustawione na M: 2, CF: 0.1.

Najlepsze parametry w modelu Naive Bayes ustawione na Dyskretyzacja: brak.

Podobnie jak w poprzednich zbiorach, model drzewa decyzyjnego sprawował się lepiej niż w przypadku modelu Naiwnego Bayes’a.



Rysunek Wizualizacja najlepszego modelu drzewa C4.5 dla zbioru Wine

1. **Podsumowanie**

Podczas analizy modelu drzewa decyzyjnego C4.5 uzyskano za każdym razem zbliżone lecz także lepsze wyniki niż w przypadku modelu Naiwnego Bayes’a. Największą różnicę wyników uzyskano dla zbioru Pima diabetes, który jest najbardziej liczny ze wszystkich sprawdzanych zbiorów, a jego wyniki polepszono o ponad 8%.

W modelu C4.5 dobre wyniki uzyskano zarówno w przypadku małej generalizacji przy dużej wielkości drzewa, jak i w przypadku dużej generalizacji gdzie drzewo było przycinane w wielu miejscach. W przypadku małej generalizacji zazwyczaj lepiej sprawowała się kroswalidacja zwykła, gdzie w przypadku dużej generalizacji stratyfikowana kroswalidacja zazwyczaj osiągałą lepsze wyniki.