Skład zespołu:
Mateusz Winnicki

WSI ĆWICZENIE 6

Cel ćwiczenia

Tematem szóstych ćwiczeń jest regresja i klasyfikacja. Zadaniem do wykonania będzie zaimplementowanie algorytmu lasu losowego i przeprowadzenie klasyfikacji metodą k-krotnej walidacji krzyżowej oceny zakupu samochodu na podstawie jego specyfikacji. Dany zbiór tworzy 1728 obserwacji podzielonych na 4 klasy.

Przebieg ćwiczenia

Ćwiczenie rozpoczniemy od zbadania wykresów przynależności do określonej klasy w zależności od badanej cechy[1], np. ceny utrzymania samochodu.

We właściwym badaniu samego algorytmu lasu losowego, będziemy sprawdzać jakość modelu w zależności od liczby drzew oraz ich głębokości. W tym celu wyznaczymy macierze błędów dla poszczególnych przebiegów, ze szczególnym uwzględnieniem dokładności oraz precyzji i czułości w zależności od klasy. Badane statystyki będą porównywane pomiędzy średnimi wartością pochodzącą z walidacji krzyżowej a wynikami zbioru walidacyjnego na całym zbiorze uczącym.

Aby uruchomić program należy to zrobić z parametrami określającymi: lokalizację pliku z badanym zestawem danych, udział zestawu uczącego (np. dla wartości 0,80 otrzymamy podział na zestawy uczący i walidacyjny 80-20), liczba określająca na ile części dzielimy zestaw uczący przy wykorzystaniu walidacji krzyżowej, maksymalna głębokość drzew decyzyjnych oraz liczba drzew użytych w lesie losowym.

Dodanie do linii *-hist* włączy opcję wykreślenia histogramów wartości cech w zależności od klasy pojazdu. Wykresy pojawią się w folderze *plots*.

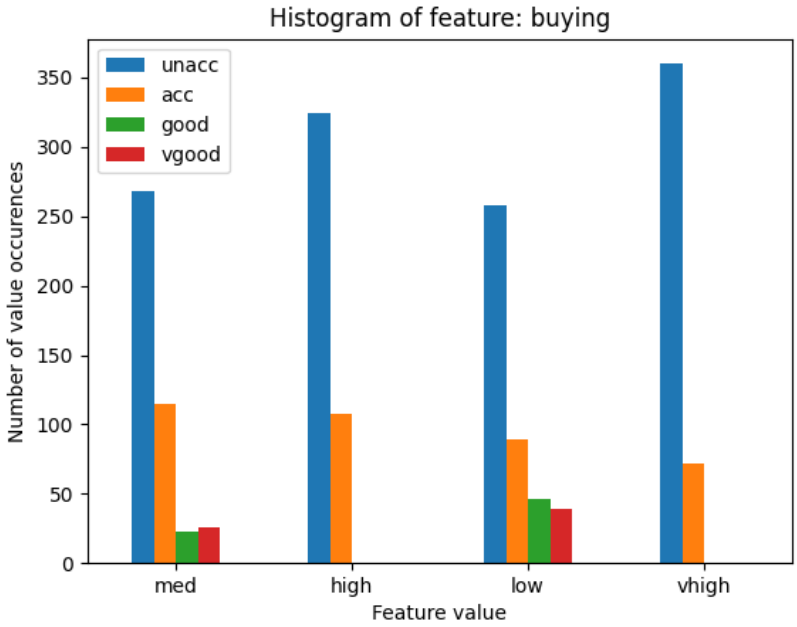
Przykład:
> *py .\classifier.py -f data/car.data -ts 0.8 -fn 4 -md 8 -tn 10 -hist*

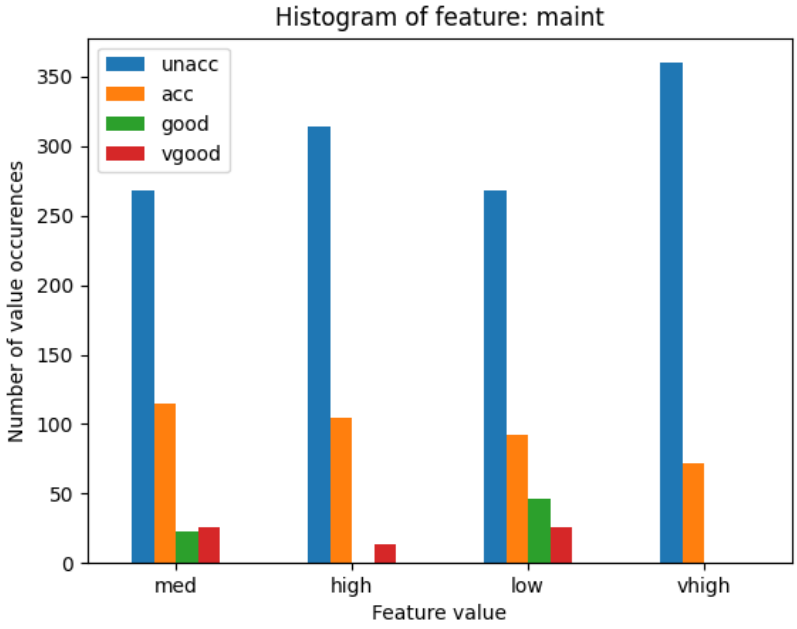
Moduły użyte w ćwiczeniu:

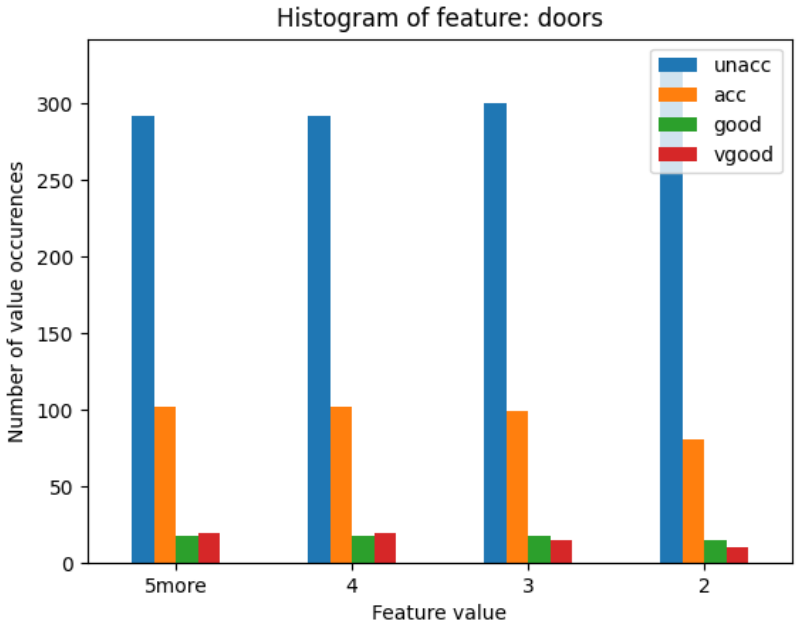
- *Numpy,*
- *Pandas,*
- *Matplotlib,*
- *Sklearn* (funkcja *train_test_split* do podziału zbioru na uczący i testowy),
- *Seaborn* (do ładnej prezentacji macierzy pomyłek w postaci heatmapy).

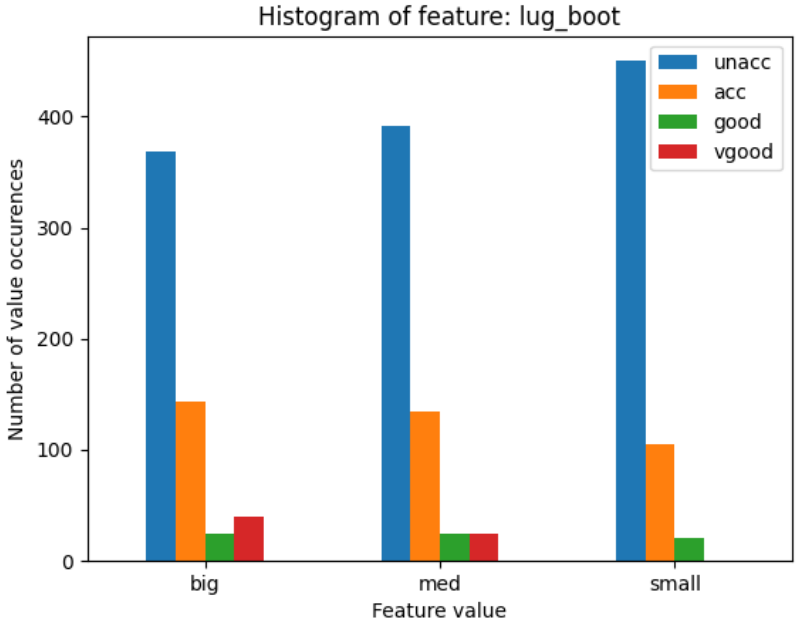
[1] Wartości cech na histogramach niestety nie są uporządkowane, poświęciłem jednak na funkcję rysującą zdecydowanie zbyt dużo czasu, żeby jeszcze dopracowywać ten drobny aspekt. Analiza danych będzie przeprowadzona pod wykresami przeze mnie, dlatego będzie to mniej wygodne w odczytywaniu dla autora, a nie recenzenta.

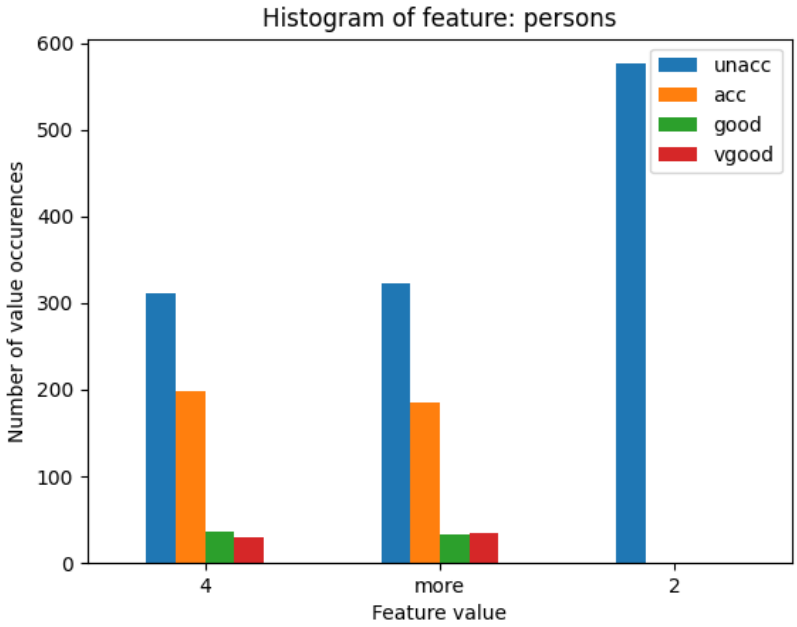
Wyniki ćwiczenia

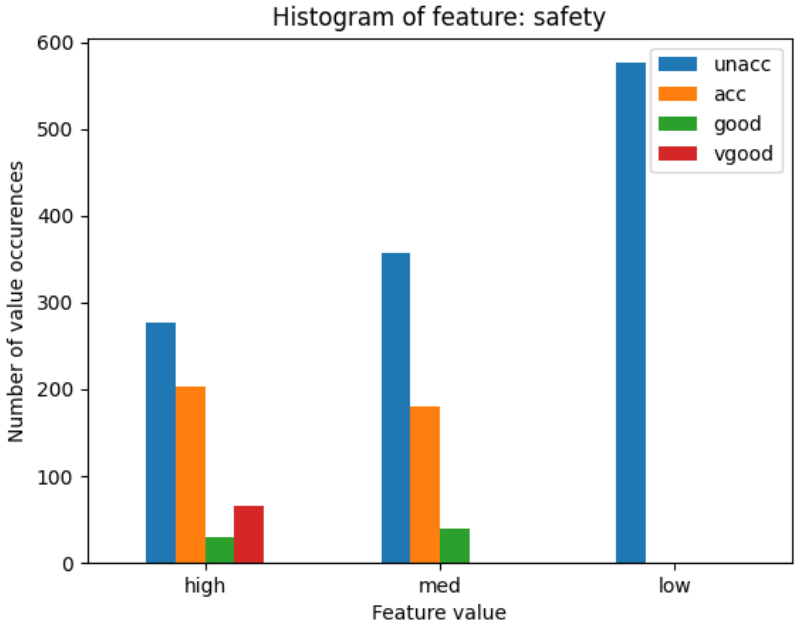





Histogram of feature: doors
unacc
acc
good
vgood
Number of value occurences
0
50
100
150
200
250
300
5more
4
3
2
Feature value


Histogram of feature: lug_boot
unacc
acc
good
vgood
Number of value occurences
0
100
200
300
400
big
med
small
Feature value


Histogram of feature: persons
Number of value occurences
Feature value
4
more
2
0
100
200
300
400
500
600
unacc
acc
good
vgood


Histogram of feature: safety
Number of value occurences
Feature value
high
med
low
0
100
200
300
400
500
600
unacc
acc
good
vgood

Z powyższych histogramów możemy na pierwszy rzut oka wysnuć dwie proste obserwacje – większość samochodów w zbiorze danych należy do klasy *unacc*, z kolei najmniej do klasy *good* i *vgood*.

Ze wzrostem ceny kupna i utrzymania, rośnie liczba samochodów zaklasyfikowanych jako *unacc* oraz spada liczebność pojazdów pozostałych klas. W badanym zbiorze danych nie ma żadnych aut klasy dobrej i bardzo dobrej, których cena zakupu miała wartość *high* i wyższą. W przypadku ceny utrzymania pojawiło się klika samochodów z etykietą *vgood* dla wartości *high*.

Podział klas w zależności od liczby drzwi jest w przybliżeniu taki sam dla każdej z wartości tej cechy, z małą tendencją wzrostową klasy *unacc* oraz spadkową klasy *acc* w przypadku samochodów 2-drzwiowych.

Im większy bagażnik, tym mniej pojazdów klasy *unacc* i więcej klasy *acc i vgood.* Samochody klasy *good* utrzymują liczebność ~30 egzemplarzy dla każdego rozmiaru bagażnika.

Samochody 2-osobowe zostały w całości zaklasyfikowane jako *unacc* (prawie 600 przypadków). Pojazdy 4-osobowe i większe, zostały podzielone bardzo podobnie – około 300 przypadków *unacc,* około 200 wystąpień klasy *acc* i po ~40 przypadków klasy *good* i *vgood*.

Dla cechy określającej bezpieczeństwo samochodu otrzymaliśmy podobny rozkład jak dla cechy określającej maksymalną liczbę jego pasażerów, z wyjątkiem klasy *vgood,* która w całości została nadana pojazdom o wysokim bezpieczeństwie.

Przykładowe drzewo decyzyjne budowane przez algorytm:

buying == vhigh

L maint == med

L persons == 2

L [unacc]

R [acc]

R maint == low

L [unacc]

R [unacc]

R safety == low

L buying == med

L [unacc]

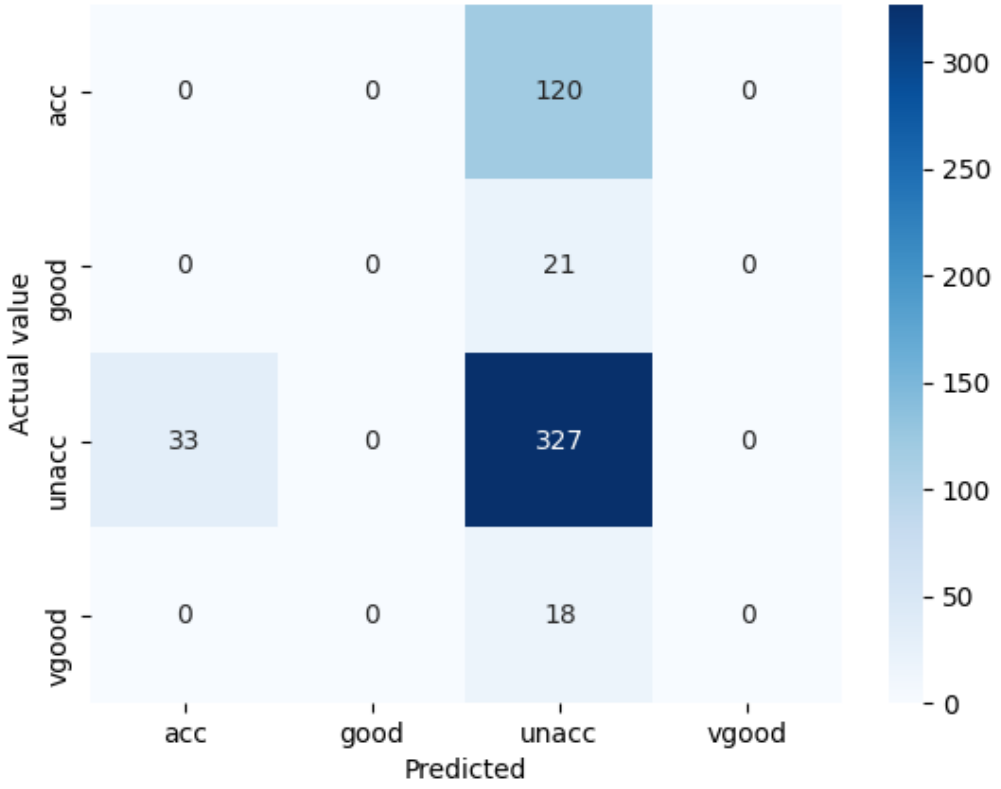
R [unacc]

R persons == 2

L [unacc]

R [acc]

Do selekcji atrybutów wykorzystałem indeks Giniego. Przy każdym podziale drzewa wykorzystywałem dwa losowe atrybuty. Mimo to przez dużą dominację klasy *unacc* otrzymywałem wiele drzew, których liście wskazywały właśnie na tą klasę, z czego mogą wynikać uzyskane w ćwiczeniu wartości precyzji i czułości poszczególnych klas.

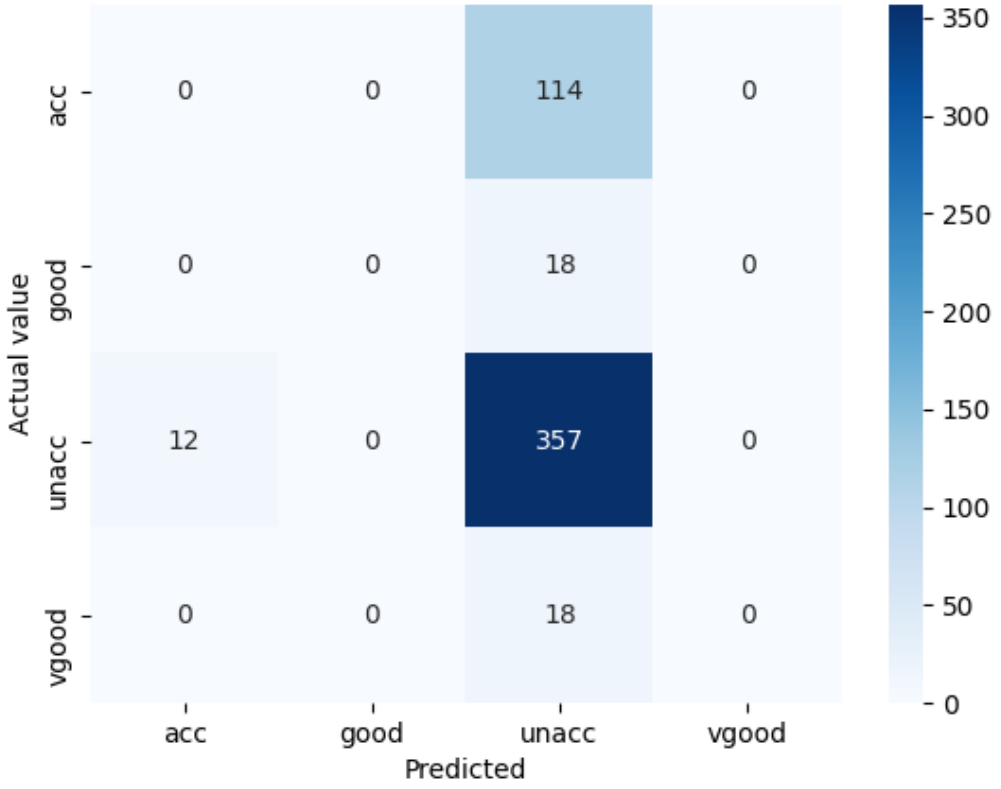


Rysunek 1 Tablica pomyłek dla przebiegu przy podziale 70-30, 10 drzew o maksymalnej głębokości 4

Średnia dokładność walidacji krzyżowej (3 podzbiory)	0,84
Średnia precyzja klasy *unacc*	0,69
Średnia czułość klasy *unacc*	0,96

Dokładność ewaluacji na zbiorze walidacyjnym	0,82
Średnia precyzja klasy *unacc*	0,67
Średnia czułość klasy *unacc*	0,91

Dla przebiegu ze stosunkowo niewielkim lasem 10 drzew, otrzymaliśmy wyniki o dużej dokładności. Czułość klasy *unacc* jest bardzo wysoka i wszystkie przypadki były jej przyporządkowywane. Parametry pozostałych klas były zerowe.

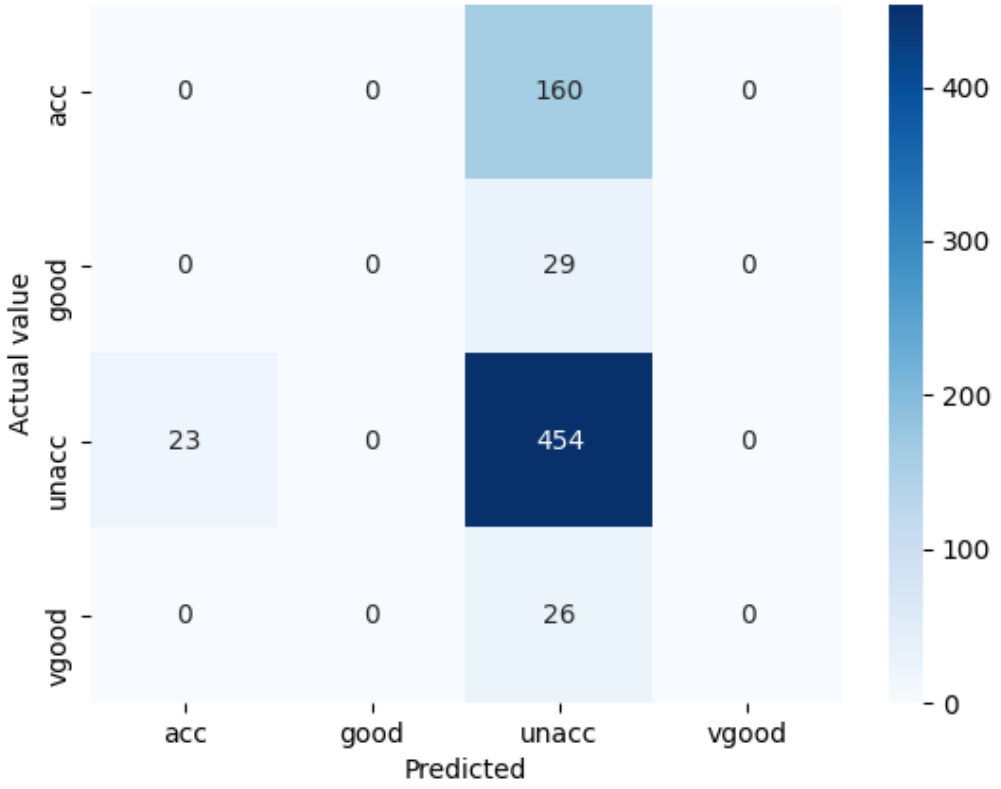


Rysunek 2 Tablica pomyłek dla przebiegu przy podziale 70-30, 10 drzew o maksymalnej głębokości 6

Średnia dokładność walidacji krzyżowej (3 podzbiory)	0,83
Średnia precyzja klasy *unacc*	0,69
Średnia czułość klasy *unacc*	0,96

Dokładność ewaluacji na zbiorze walidacyjnym	0,84
Średnia precyzja klasy *unacc*	0,71
Średnia czułość klasy *unacc*	0,96

Po zwiększeniu maksymalnej głębokości drzewa do 6 mierzone parametry wzrosły, jednak wynika to niestety z innego podziału danych – więcej wystąpień klasy *unacc* w zestawie testowym oznacza więcej praktycznie pewnych rozpoznać tejże klasy.

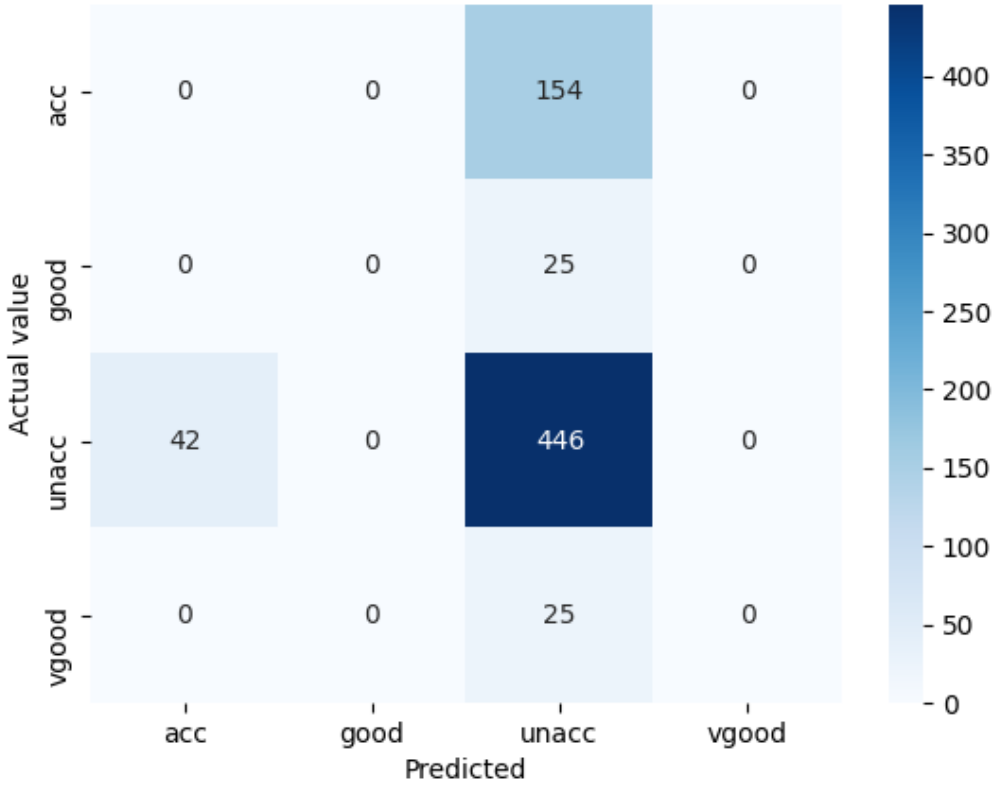


Rysunek 3 Tablica pomyłek dla przebiegu przy podziale 60-40, 50 drzew o maksymalnej głębokości 6

Średnia dokładność walidacji krzyżowej (3 podzbiory)	0,84
Średnia precyzja klasy *unacc*	0,69
Średnia czułość klasy *unacc*	0,93

Dokładność ewaluacji na zbiorze walidacyjnym	0,83
Średnia precyzja klasy *unacc*	0,68
Średnia czułość klasy *unacc*	0,95

Większa liczba drzew w lesie utrzymuje poziom poprzednich prób. Wynika to z faktu, że algorytm lasu losowego generuje jeszcze więcej drzew, które najprawdopodobniej będą „głosowały” na klasę *unacc*.

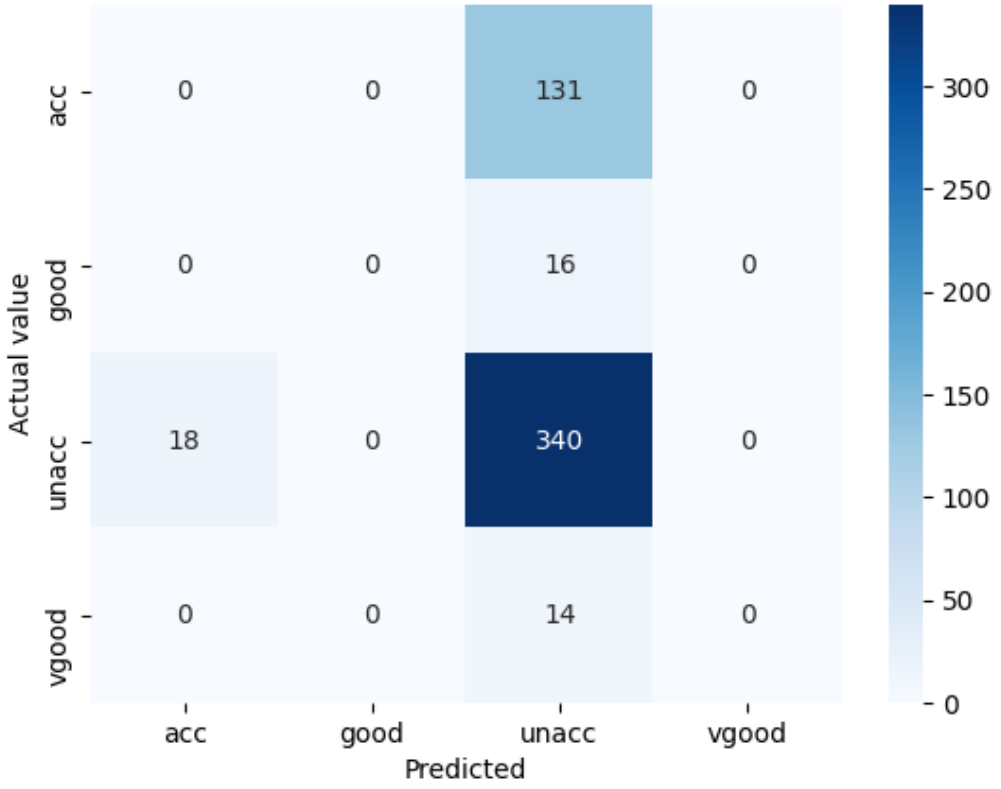


Rysunek 4 Tablica pomyłek dla przebiegu przy podziale 60-40, 50 drzew o maksymalnej głębokości 10

Średnia dokładność walidacji krzyżowej (3 podzbiory)	0,82
Średnia precyzja klasy *unacc*	0,67
Średnia czułość klasy *unacc*	0,91

Dokładność ewaluacji na zbiorze walidacyjnym	0,82
Średnia precyzja klasy *unacc*	0,68
Średnia czułość klasy *unacc*	0,91

Pogłębianie kreowanych drzew również nie pomaga naszemu klasyfikatorowi.



Rysunek 5 Tablica pomyłek dla przebiegu przy podziale 60-40, 50 drzew o maksymalnej głębokości 6

Średnia dokładność walidacji krzyżowej (5 podzbiorów)	0,84
Średnia precyzja klasy *unacc*	0,69
Średnia czułość klasy *unacc*	0,93

Dokładność ewaluacji na zbiorze walidacyjnym	0,82
Średnia precyzja klasy *unacc*	0,67
Średnia czułość klasy *unacc*	0,95

Podział zbioru trenującego na więcej podzbiorów, na których prowadzimy walidację krzyżową zwiększył czas wykonywania programu, jednak otrzymane metryki pozostają te same.

Podsumowanie

Otrzymany algorytm klasyfikacyjny lasu losowego nie spełnił do końca moich oczekiwań. Bardzo wysoka czułość jednej klasy oraz zerowe parametry pozostałych klas nie są raczej oczekiwanym zjawiskiem. Myślę, że dalsze manipulacje danymi wejściowymi nie doprowadziłyby do żadnego przełomu.

Zaimplementowany klasyfikator jest o tyle problematyczny, że gdybyśmy analogicznie stworzyli taki sam, dotyczący np. systemu alarmowego, który nadawałby nam sygnały: *włamywacz, nielubiana ciocia, domokrążca, zaproszony gość*, mógłby on informować nas za każdym razem, że powitamy przed drzwiami ostatnią klasę (z racji, że jest ona najpowszechniejsza i mamy na jej temat najwięcej informacji). Efekty raczej nie byłyby najszczęśliwsze.

Do implementacji należałoby wprowadzić elementy, które w znaczący sposób wpływałyby na losowość tworzonych drzew. Sposobem mogłaby być również rezygnacja z indeksu Giniego, który dzieli nam drzewa w sposób najlepszy z możliwych. Przy znacznej przewadze jednej klasy ponad pozostałymi, wiadomym jest, że najczęściej dokonujemy wtedy podziału na korzyść klasy dominującej. Tak jak wynikało to ze wstępnej analizy danych, 2-drzwiowe modele lub samochody o niskim bezpieczeństwie z miejsca prowadzą nas do klasy *unacc*.

Próbowałem wprowadzić metodę, która z zestawu uczącego losuje ze zwracaniem podzbiór wektorów cech, jednak efekt był ten sam – wynika to z mniejszej szansy na wybranie wektorów zaklasyfikowanych jako inne niż *unacc*.

Co do walidacji krzyżowej – nie dała ona nam w tym ćwiczeniu oszałamiających efektów, ponieważ nie wykorzystywaliśmy jej do dostrajania hiperparametrów, a jedynie porównywaliśmy jej wyniki ze zbiorem walidacyjnym. Nie wynikało z nich jednak nic szczególnego, ponieważ w podzbiorach wykorzystywanych w tym procesie nadal jedna klasa dominowała nad pozostałymi.