Uczenie się maszyn - dokumentacja końcowa

1. Temat projektu:

Regresyjny las losowy - należy zaimplementować zmodyfikowaną wersję algorytmu generowania lasu losowego regresji, w której do generowania kolejnych drzew losowane są częściej elementy ze zbioru uczącego, na których dotychczasowy model popełniał większe błędy. W eksperymentach zostanie wykorzystane zadanie **Wine Quality**.

2. Opis algorytmów, które będą wykorzystane:

Algorytm CART

Do rozwiązywania problemów zarówno regresyjnych, jak i klasyfikacyjnych służy metoda CART. Metoda ta powstała w roku 1984. Opiera się na dwóch typach drzew:
- drzewa klasyfikacyjne (nieistotne w projekcie)
- drzewa regresyjne - służą do przewidywania wartości zmiennej docelowej, charakteryzuje je zmienna docelowa typu ciągłego

Konstrukcja algorytmu CART ma postać ciągu pytań, na które odpowiedzi determinują kolejne pytania, bądź kończą etap. W wyniku otrzymujemy strukturę drzewa, która w węzłach końcowych nie zawiera już pytań, lecz same odpowiedzi. Zasada działania algorytmu CART polega na tym, że rozszczepianie danych w węzłach opiera się na jednej zmiennej decyzyjnej, a podział zostaje zatrzymany, gdy odpowiedź na dane pytanie nie wyznacza kolejnego pytania. Metodologię CART charakteryzuje rekursywny podział binarny, a więc to, że węzeł macierzysty rozdzielany jest zawsze na dwa węzły potomne. Po dokonaniu podziałów, w każdej warstwie mierzony jest błąd klasyfikacji, a więc pewna niejednorodność kategorii wyniku.

Regresyjny las losowy (ang. *Regression forest*)

Algorytm buduje określoną liczbę drzew. W przypadku regresji zwracanym wynikiem jest średnia arytmetyczna wszystkich wartości zwracanych przez poszczególne drzewa. Każde kolejne drzewo jest budowane w oparciu o próbę bootstrapową. Polega ona na wylosowaniu ze zwracaniem *k* obserwacji z ciągu uczącego o długości *K*. Co więcej, w każdym węźle podział odbywa się za pomocą *m* wylosowanych cech (atrybutów) danej obserwacji. Liczba wylosowanych cech jest mniejsza niż liczba wszystkich dostępnych cech(najczęściej około sqrt z liczby wszystkich cech). Zatem w lesie losowym powstaje *n* drzew, składających się z k obserwacji (każde drzewo może mieć unikalny zestaw cech).

Drzewo decyzyjne wzmocnione (ang. Boosted decision tree) – regresyjne

Wariacja drzewa decyzyjnego. Wykorzystuje procedurę wzmacniającą AdaBoost (ang. Adaptive Boosting). Podobnie jak w przypadku lasu losowego algorytm losuje nowe ciągi uczące. Główna różnica polega na tym, iż każde kolejne budowane drzewo stara się kompensować błędy swojego poprzednika. Owo „ulepszanie” jest możliwe dzięki przypisaniu wag do obserwacji. W przypadku popełnienia błędu, zwiększana jest waga danego elementu i jest on częściej losowany w kolejnej iteracji.

3. Tworzenie pojedynczego drzewa

a) Reguła podziału: minimalizacja błędu średniokwadratowego – RMSE (ang. Root *Mean Squared Error*). Uzyskujemy to poprzez minimalizację średniej ważonej odchylenia standardowego obu synów.

b) Kryterium stopu: minimalna liczebność liścia, która mówi, że podział zbioru kończy się, gdy liczebność węzła zrówna się z minimalną przyjętą liczebnością. Przyjęta liczba nie może być zbyt mała, ponieważ doprowadzi to do nadmiernego dopasowania.

4. Tworzenie lasu

Las, w którym każde kolejne budowane drzewo wykorzystuje częściej dane na których dotychczasowy model popełniał największe błędy. Jest to połączenie wszystkich trzech algorytmów z pkt. 2.

Przy budowie pierwszego drzewa wybieramy losowo zestaw obserwacji, następnie dokonujemy sprawdzenia naszego modelu. Porównujemy otrzymane wyniki z rzeczywistymi. Na podstawie różnicy między wartością przewidzianą a rzeczywistą jesteśmy w stanie określić prawdopodobieństwo wybrania konkretnego testu do budowy następnego drzewa.

$$p = \frac{|ypredicted - y|}{sum\ of\ all\ errors}$$

Budujemy kolejne drzewo biorąc pod uwagę prawdopodobieństwa przy wyborze zestawu obserwacji. Dokonujemy sprawdzenia modelu. Liczymy prawdopodobieństwa. Powtarzamy do uzyskania określonej liczby drzew.

5. Dane wejściowe

W zadaniu “Wine Quality” występują dwa zestawy danych dotyczące win: białych i czerwonych. Dane wejściowe zawierają obiektywne obserwacje fizyczno-chemiczne np. PH i zmienną celu, która opiera się na subiektywnej ocenie (średnia ocena co najmniej 3 ekspertów). Każdy ocenił wino między 0 (bardzo złe) i 10 (bardzo dobre). Zestaw nie zawiera wartości prywatnych producentów takich jak typ winogron, nazwy producenta lub ceny. Wszystkie wartości są wartościami numerycznymi. Przykładowe wartości wierszy oraz dane statystyczne danych znajdują się poniżej. Brak wartości None value.

Wina czerwone:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

Wina białe:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5.877909
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0.885639
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3.000000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5.000000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6.000000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6.000000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9.000000

Obserwacje: znacznie większa liczba rekordów win białych, mała wariancja zmiennej celu w obu zestawach danych

6. Procedura eksperymentalna oraz uzyskane wyniki

Wszystkie testy wykonywane są za pomocą klasycznego lasu oraz wzmocnionego. Pracuję na dwóch zestawach danych. Z tego powodu dla każdego testu otrzymuję 4 wyniki. W każdym z testów liczę RMSE/MSE/MAE oraz accuracy. Accuracy określam, jako stosunek liczby dobrze przewidzianych wartości zmiennej celu do liczby wszystkich obserwacji w zbiorze testowym.

Przeprowadzone zostały następujące testy:

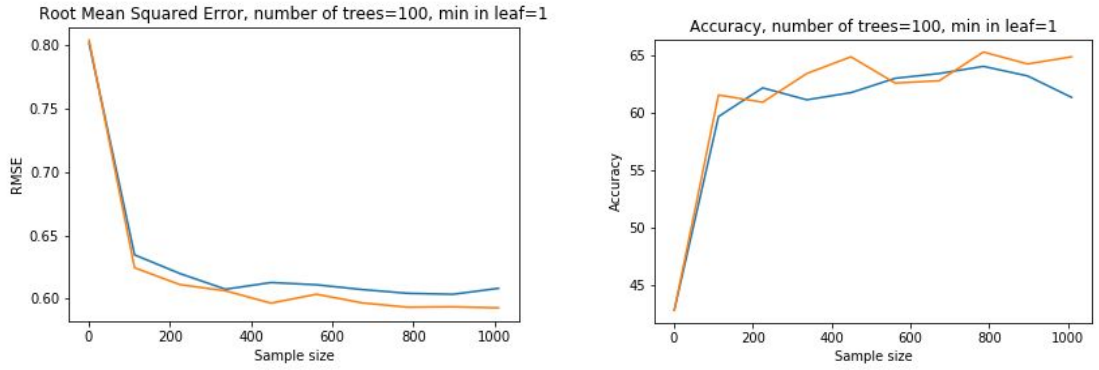
6.1 Ustalenie optymalnych wartości parametrów, które mogę zmieniać przy budowie lasu. Wynikiem obserwacji wykresy zmian RMSE oraz Accuracy.

Obserwuję, jak zmieniają się RMSE i Accuracy wraz ze zmianą parametrów takich jak:

- liczba obserwacji losowanych ze zbioru treningowego do nauki jednego drzewa, zakres zmian: 1:liczba obserwacji w zbiorze danych z krokiem 1/10 liczby obserwacji
- liczba drzew w naszym lesie, zakres zmian: 1:141 z krokiem 10
- minimalna liczba obserwacji w liściu, zakres zmian: 1:41 z krokiem 5

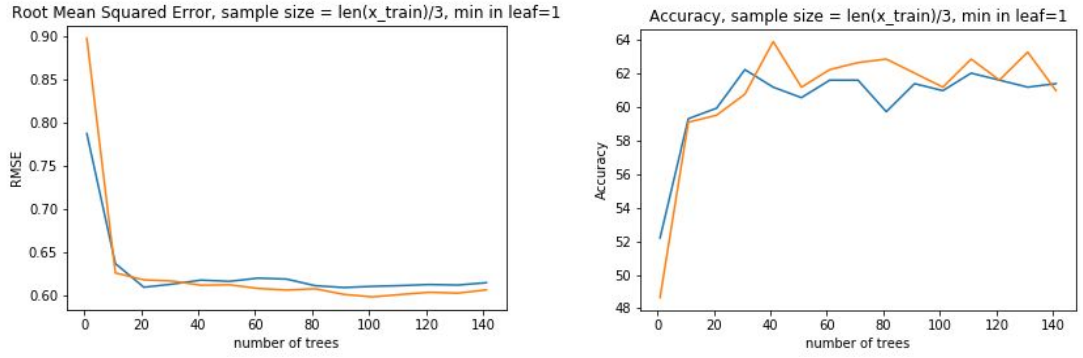
Badania przeprowadzane były na jednym losowym podziale zbioru danych dla obu typów lasu. Stosunek obserwacji trening:test 7:3.

Wina czerwone las losowy niewzmocniony(niebieski) vs wzmocniony(pomarańczowy)



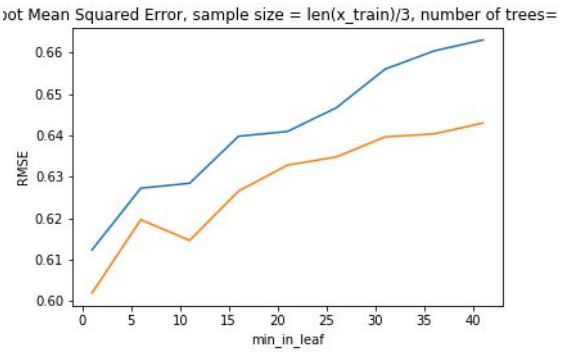
RMSE oraz Accuracy w zależności od liczby obserwacji do nauki jednego drzewa

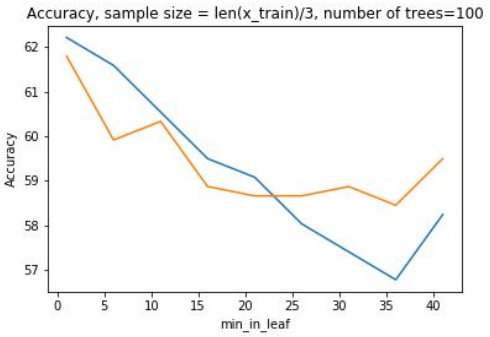
Obserwacje: delikatnie lepsze wyniki uzyskał las wzmocniony. Optymalną wartość liczby obserwacji wykorzystaną do nauki jednego drzewa ustalam na ⅓ zbioru testowego



RMSE oraz Accuracy w zależności od liczby drzew w lesie

Obserwacje: bardzo zbliżone wyniki uzyskane przez oba lasy. Optymalną wartość liczby drzew w lesie ustalam na 100.

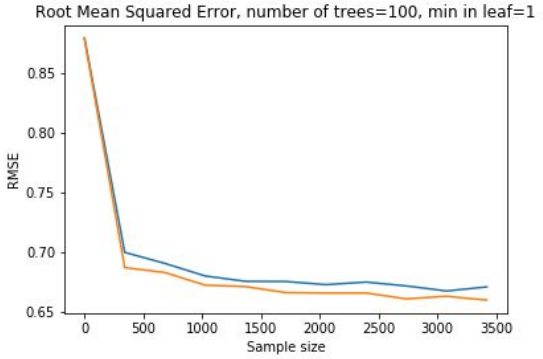


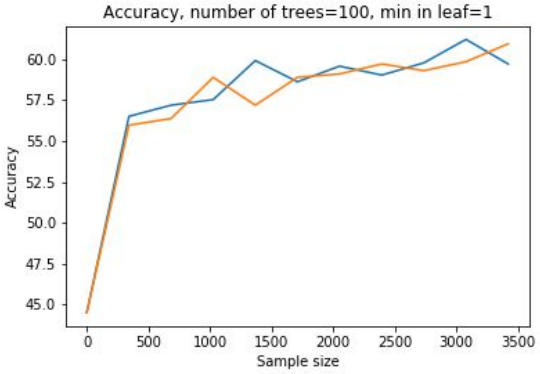


RMSE oraz Accuracy w zależności od minimalnej liczby obserwacji w liściu

Obserwacje: widoczne pogorszenie się wyników wraz ze wzrostem minimalnej liczby, las wzmocniony lepiej radził sobie w przypadku wzrostu

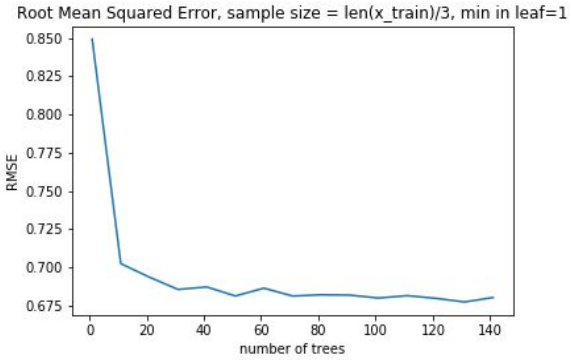
Wina białe las losowy niewzmocniony(niebieski) vs wzmocniony(pomarańczowy)

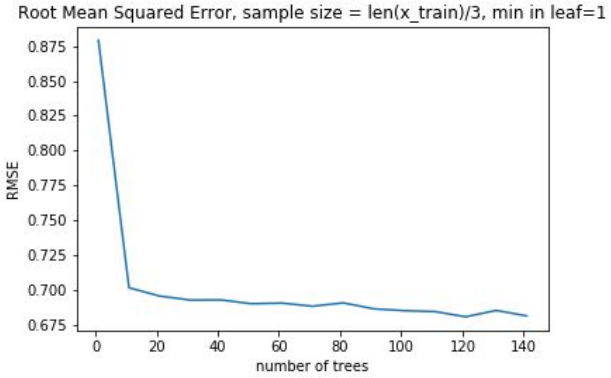




RMSE oraz Accuracy w zależności od liczby obserwacji do nauki jednego drzewa

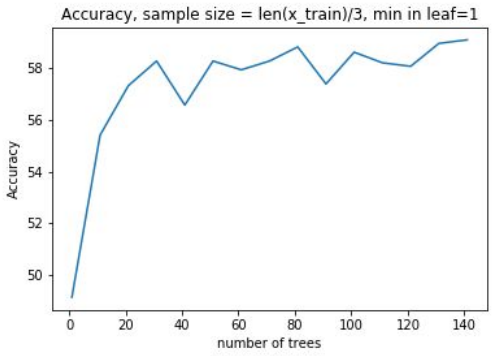
Niewzmocniony **Wzmocniony**

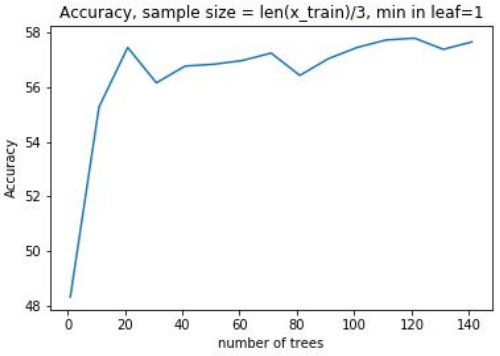




RMSE oraz Accuracy w zależności od liczby drzew w lesie

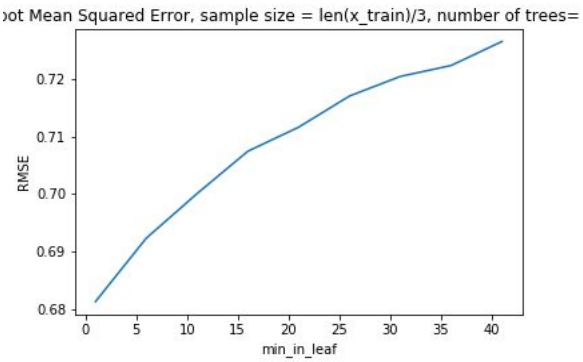
Niewzmocniony **Wzmocniony**

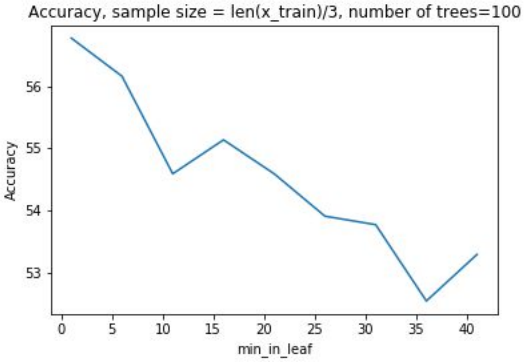




RMSE oraz Accuracy w zależności od minimalnej liczby obserwacji w liściu

Niewzmocniony **Wzmocniony**





RMSE oraz Accuracy w zależności od minimalnej liczby obserwacji w liściu

Obserwacje: obserwacje podobne do obserwacji zawartych w obserwacjach dla win czerwonych. Delikatnie słabsze wyniki RMSE oraz Accuracy niż dla win czerwonych.

Komentarz: Sample size jest to liczba obserwacji losowanych do nauczenia jednego drzewa, w przypadku lasu wzmocnionego każda obserwacja ma inne prawdopodobieństwo bycia wylosowanym.

Wnioski ad. 6.1

Widzimy, że model pracujący na mniejszym zbiorze danych (wina czerwone) otrzymał lepsze wyniki. Model, gdy pracował na bardziej licznym zbiorze zwiększyła się szansa, że w zbiorze testowym znajdzie się więcej przypadków, z którymi model wcześniej nie miał do czynienia. Obserwacjami zakłócającymi model są również outlier'y - obserwacje mocno różniące się od innych. Niestety w zbiorze mniej licznym mamy większe zagrożenie wystąpienia nadmiernego dopasowania. Walczyć z nimi można stosując różne techniki np. przycinanie drzewa.

Uzyskane wyniki nie są deterministyczne, ponieważ zostały uzyskane po jednorazowym losowym podziale test:trening. Jednakże jesteśmy w stanie zaobserwować jak zachowuje się model w przypadku zmian parametrów, a taki był cel tych testów.

6.2 Wykonanie 10 prób dla obu typów lasu i obu zestawów danych (dla optymalnych wartości parametrów budowy lasu) dla różnych wartości podziału test:trening (3:7)(1:1). Wynikiem testów jest średnia wartość RMSE/MSE/MAE oraz accuracy.

Optymalne wybrane parametry: liczba drzew 100, min liczba obserwacji w liściu 1, liczba obserwacji do nauki jednego drzewa ⅓ wszystkich obserwacji.

```
Default values: number of trees: 100, min in leaf 1, sample size len(x_train)/3
Default values tests for white not boosted 7:3
Mean Mean Absolute Error:  0.5303582380395901
Mean Mean Squared Error:  0.45609111205925223
Mean Root Mean Squared Error:  0.6753451799333821
Mean Accuracy:  57.385976855003406
Time of calculating:  207.96924138069153
```

Wina białe, las nie wzmocniony, podział test:trening 7:3

```
Default values tests for white  boosted 7:3
Mean Mean Absolute Error:  0.5232442483159367
Mean Mean Squared Error:  0.4388308887400218
Mean Root Mean Squared Error:  0.6624431211357107
Mean Accuracy:  59.97277059223962
Time of calculating:  2487.3505878448486
```

Wina białe, las wzmocniony, podział test:trening 7:3

```
Default values tests for red not boosted 7:3
Mean Mean Absolute Error:  0.49166777779552556
Mean Mean Squared Error:  0.39296932121357775
Mean Root Mean Squared Error:  0.6268726515119142
Mean Accuracy:  59.707724425887264
Time of calculating:  84.68111419677734
```

Wina czerwone, las nie wzmocniony, podział test:trening 7:3

```
Default values tests for red boosted 7:3
Mean Mean Absolute Error:  0.4728713151370174
Mean Mean Squared Error:  0.35695102211193275
Mean Root Mean Squared Error:  0.5974537824065831
Mean Accuracy:  64.92693110647181
Time of calculating:  685.7854497432709
```

Wina czerwone, las wzmocniony, podział test:trening 7:3

Obserwacje: Widzimy, że dla obu zestawów testowych las wzmocniony uzyskiwał delikatnie lepsze rezultatym, jednakże trenowanie oraz predykcja lasu wzmocnionego trwa o wiele dłużej. Stosując nasz algorytm cały proces wydłużał się około 10-krotnie.

Zmiana podziału test:trening z 7:3 na 1:1

```
Default values tests for white not boosted 1:1
Mean Mean Absolute Error:  0.5336712506711239
Mean Mean Squared Error:  0.464770775532133
Mean Root Mean Squared Error:  0.6817409885962065
Mean Accuracy:  57.370355247039605
Time of calculating:  172.31386709213257
```

Wina białe, las nie wzmocniony, podział test:trening 1:1

```
Default values tests for white boosted 1:1
Mean Mean Absolute Error:  0.5541096838622458
Mean Mean Squared Error:  0.49024235481263784
Mean Root Mean Squared Error:  0.7001730891805524
Mean Accuracy:  55.492037566353616
Time of calculating:  1648.5431907176971
```

Wina białe, las wzmocniony, podział test:trening 1:1

```
Default values tests for red not boosted 1:1
Mean Mean Absolute Error:  0.47532596756243695
Mean Mean Squared Error:  0.3828192787691632
Mean Root Mean Squared Error:  0.6187239115867135
Mean Accuracy:  61.82728410513142
Time of calculating:  67.569180727005
```

Wina czerwone, las nie wzmocniony, podział test:trening 1:1

```
Default values tests for red boosted 1:1
Mean Mean Absolute Error:  0.4929257539369566
Mean Mean Squared Error:  0.3784226573502787
Mean Root Mean Squared Error:  0.615160676043486
Mean Accuracy:  61.702127659574465
Time of calculating:  455.326265335083
```

Wina czerwone, las wzmocniony, podział test:trening 1:1

Obserwacje: Widzimy, że dla obu zestawów testowych wyniki są bardzo zbliżone. Dodatkowo trenowanie oraz predykcja lasu wzmocnionego trwa o wiele dłużej. Stosując nasz algorytm cały proces wydłużał się około 10-krotnie.

6.3 Wykonanie 10 prób predykcji wykorzystując **model lasu losowego regresji zawartego w sklearn** dla różnych wartości podziału test:trening (3:7)(1:1). Wynikiem testów jest średnia wartość RMSE/MSE/MAE oraz accuracy. Ten test pozwoli nam porównać wyniki otrzymane dzięki naszemu algorytmowi budowy lasu z algorytmem zawartym w sklearn. Potwierdzi poprawne działanie. Testy były wykonane na serwerze Colaboratory. Załączony plik: sklearn_tests.py

```
Red wines with sklearn test:trening 1:1
Mean Mean Absolute Error: 0.614294349979336
Mean Mean Squared Error: 0.3777125625
Mean Root Mean Squared Error: 0.614294349979336
Mean Accuracy: 64.3625
```

Wina czerwone, stosunek trening:test 1:1

```
Red wines with sklearn test:trening 3:7
Mean Mean Absolute Error: 0.5881822373770611
Mean Mean Squared Error: 0.3465886458333333
Mean Root Mean Squared Error: 0.5881822373770611
Mean Accuracy: 67.4375
```

Wina czerwone, stosunek trening:test 3:7

```
White wines with sklearn test:trening 1:1
Mean Mean Absolute Error: 0.6495570311224197
Mean Mean Squared Error: 0.4219951980400164
Mean Root Mean Squared Error: 0.6495570311224197
Mean Accuracy: 63.27888934258881
```

Wina białe, stosunek trening:test 1:1

```
White wines with sklearn test:trening 3:7
Mean Mean Absolute Error: 0.6342661529911572
Mean Mean Squared Error: 0.40251040816326533
Mean Root Mean Squared Error: 0.6342661529911572
Mean Accuracy: 66.00000000000001
```

Wina białe, stosunek trening:test 3:7

Wnioski ad. 6.3

Widzimy, że model pracujący na mniejszym zbiorze danych (wina czerwone) otrzymał lepsze wyniki. Zauważyć można również wzrost błędów oraz spadek accuracy w momencie zmiany podziału zestawu danych trening:test z 7:3 do 1:1. Model po zmianie stosunku miał mniejszą liczbę obserwacji do treningu oraz zwiększyła się szansa, że w zbiorze testowym znajdzie się więcej przypadków, z którymi model wcześniej nie miał do czynienia. Obserwacjami zakłócającymi model są również outlier'y - obserwacje mocno różniące się od innych.

7. Wnioski oraz porównania

Uważam, że model uzyskał satysfakcjonujące wyniki porównując je z uzyskanymi przy pomocy modelu z sklearn. Delikatnie lepsze wyniki sklearn mogą wynikać z wykorzystania skalowania predyktorów lub wykorzystania wiedzy na temat ważności cech zawartych w odpowiednich kolumnach(features importance).

Wykorzystanie wzmocnienia w lesie, z którego chcemy korzystać w naszym zadaniu uważam za nieefektywne. Mała wariancja zmiennej celu powoduje podobne błędy na zbiorze treningowym przy budowie lasu. Przypisane wagi są zbliżone do jednostajnego rozkładu. Cały proces znacznie wydłuża budowanie lasu. Przy każdym kolejnym dodanym drzewie następuję weryfikacja obecnego modelu. Znacznie zwiększamy koszt, praktycznie nie poprawiając modelu.

Otrzymany w zadaniu zestaw danych jest trudnym zestawem do pracy, ponieważ występuje straszna dysproporcja w ilości powtórzeń danej wartości zmiennej celu. W obu zestawach 75% wartości to 5 i 6 (zakres wartości 0-10). Brak wartości bliskich 0 oraz 10.

8. Otwarte kwestie wymagające późniejszego rozwiązania (wraz z wyjaśnieniem powodów, dla których ich rozwiązanie jest odłożone na później).

Brak określenia liczby drzew w lesie. Decyzja zostanie podjęta po analizie zachowania modelu dla różnej liczby drzew.

8.1 Późniejsze rozwiązanie (wyjaśnienie)

Tą kwestię badałem w teście 6.1 w którym testowałem różne parametryzacje lasu. Doszedłem do wniosku, że 100 drzew w lesie będzie optymalnym rozwiązaniem.

Sprawdziłem również, że w gotowych realizacjach lasu losowego wstępną liczbą drzew jest również 100.

9. Działanie programu

W przypadku uruchomienia pliku BoostedRegressionForest.py zostaną utworzone dwa lasy(wzmocniony i niewzmocniony) dla każdego zestawu danych z domyślnymi wartościami parametrów lasu. W przypadku uruchomienia pliku tests.py zostaną uruchomione wszystkie testy omówione w pkt. 6. Plik sklearn_tests.py zawiera testy przeprowadzone z wykorzystaniem lasu losowego dostarczonego przez sklearn.