Rafał Wójcik, 277366

Koncepcja projektu nr 6

Regresyjny las losowy

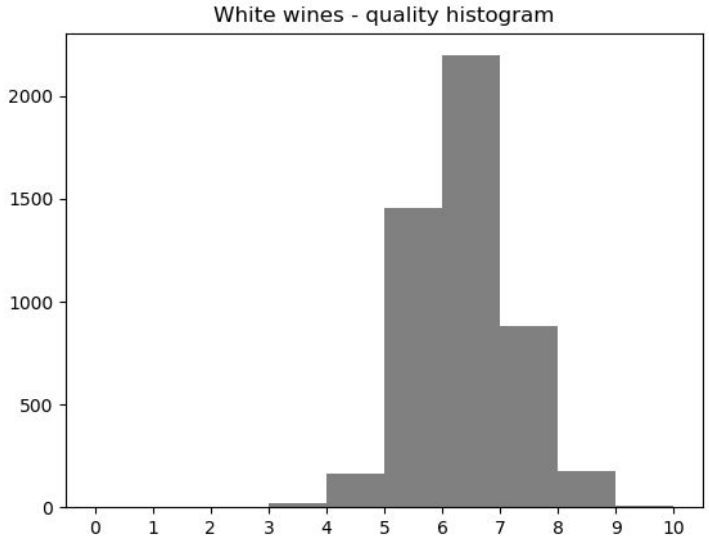
1. Treść projektu

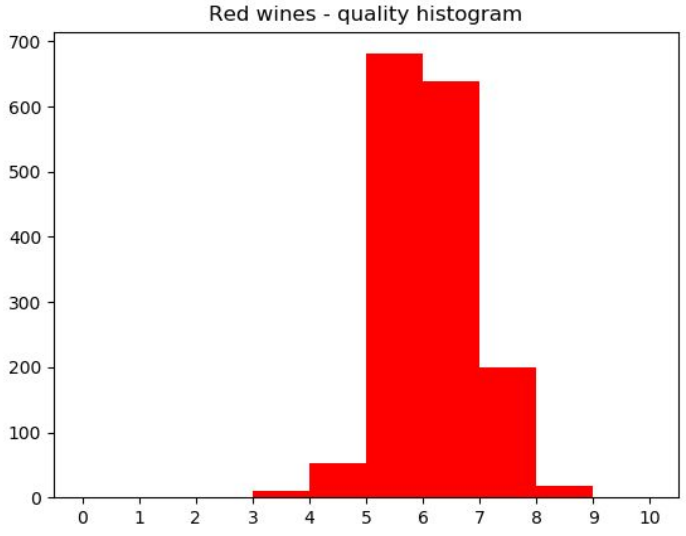
Zaimplementować zmodyfikowaną wersję algorytmu generowania lasu losowego regresji, w której do generowania kolejnych drzew losowane są częściej elementy ze zbioru uczącego, na których dotychczasowy model popełniał większe błędy. W eksperymentach należy wykorzystać zadanie Wine Quality.

2. Wstępna analiza danych

Zadanie zawiera dwa zbiory danych wiążące chemiczne właściwości win z ich jakością dla win czerwonych oraz białych. Wina posiadają 11 cech oraz przyporządkowaną im klasę oznaczającą jakość w skali od 0 do 10.

W pierwszym kroku przeanalizujmy rozkład win pod względem grup jakości:

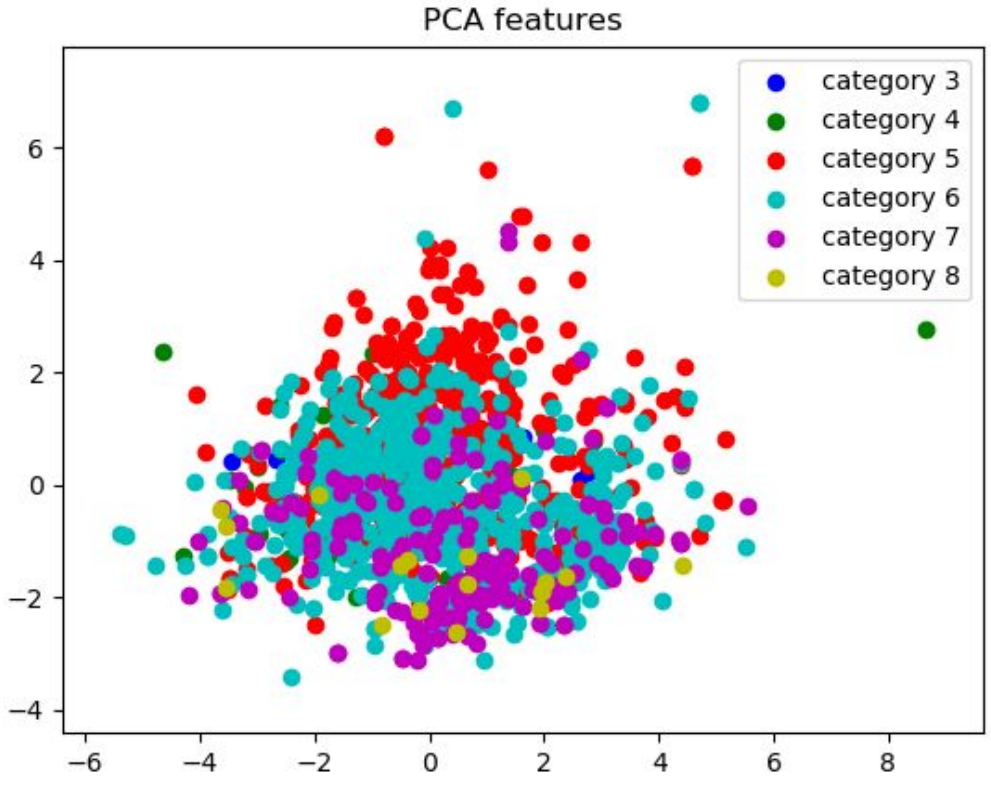


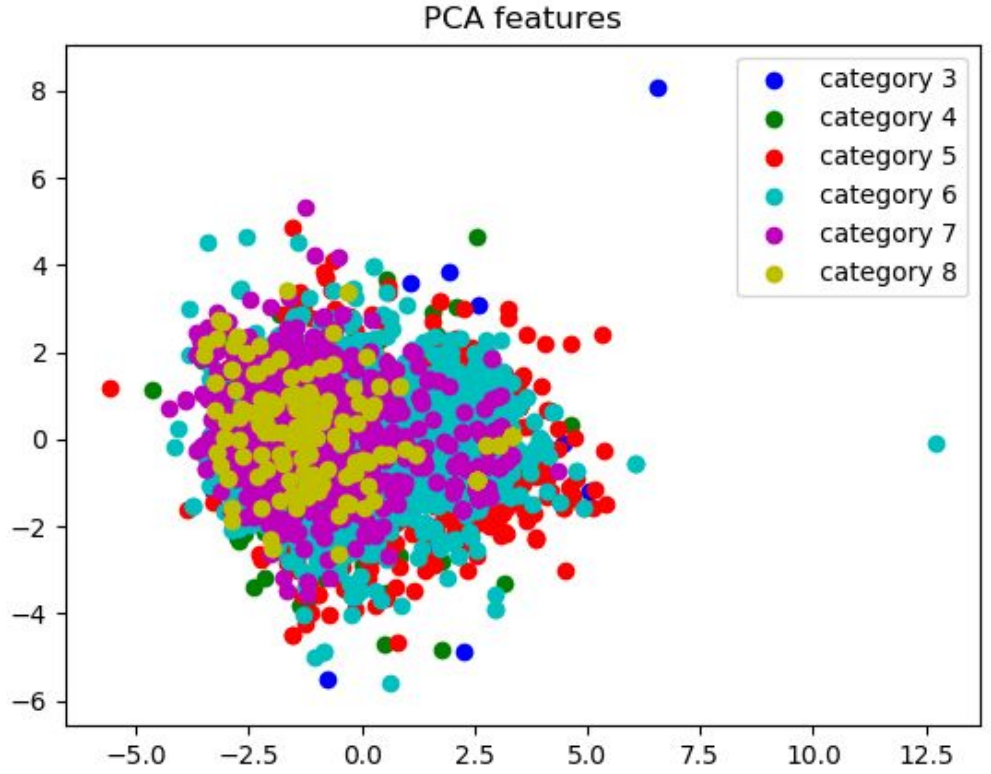


Jak widać zgodnie z opisem zawartym na stronie udostępniającej dane - jest znacznie mniej win wybitnych oraz słabych, a przeważają te znajdujące się w klasach 5-7.

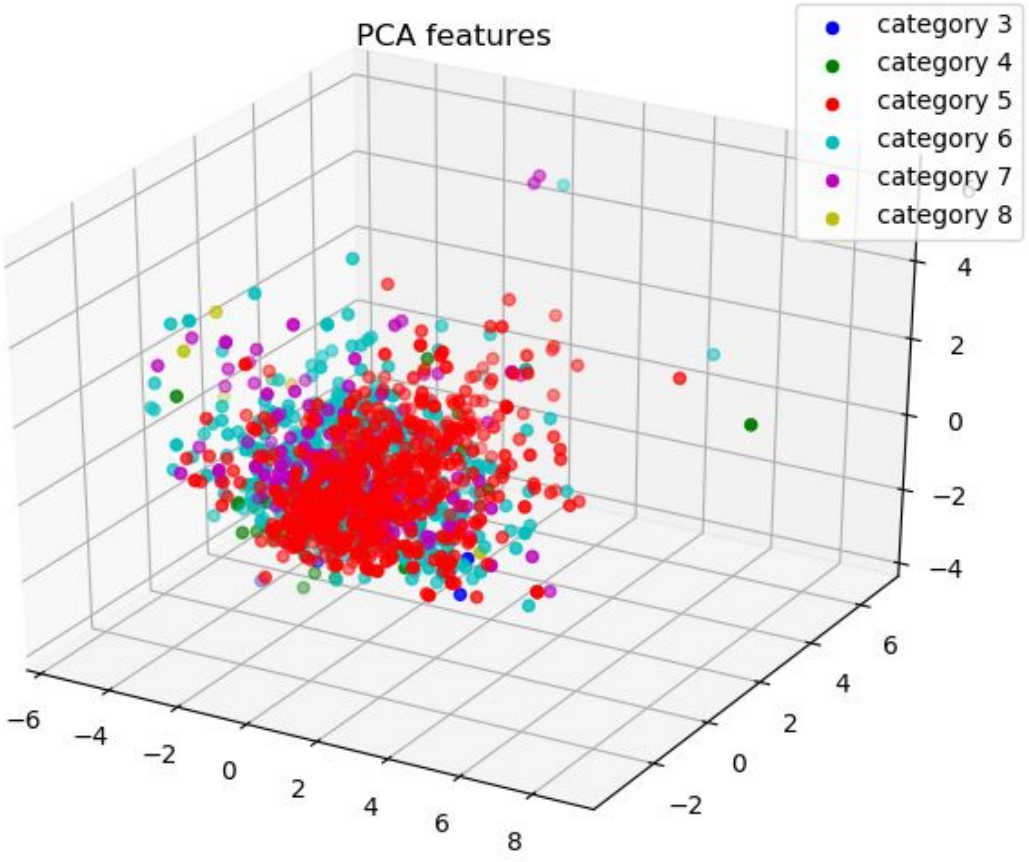
Na razie rozważać będziemy zbiory danych oddzielnie. Aby lepiej zrozumieć dane spróbujmy narysować je na wykresie. W celu zredukowania liczby cech skorzystamy

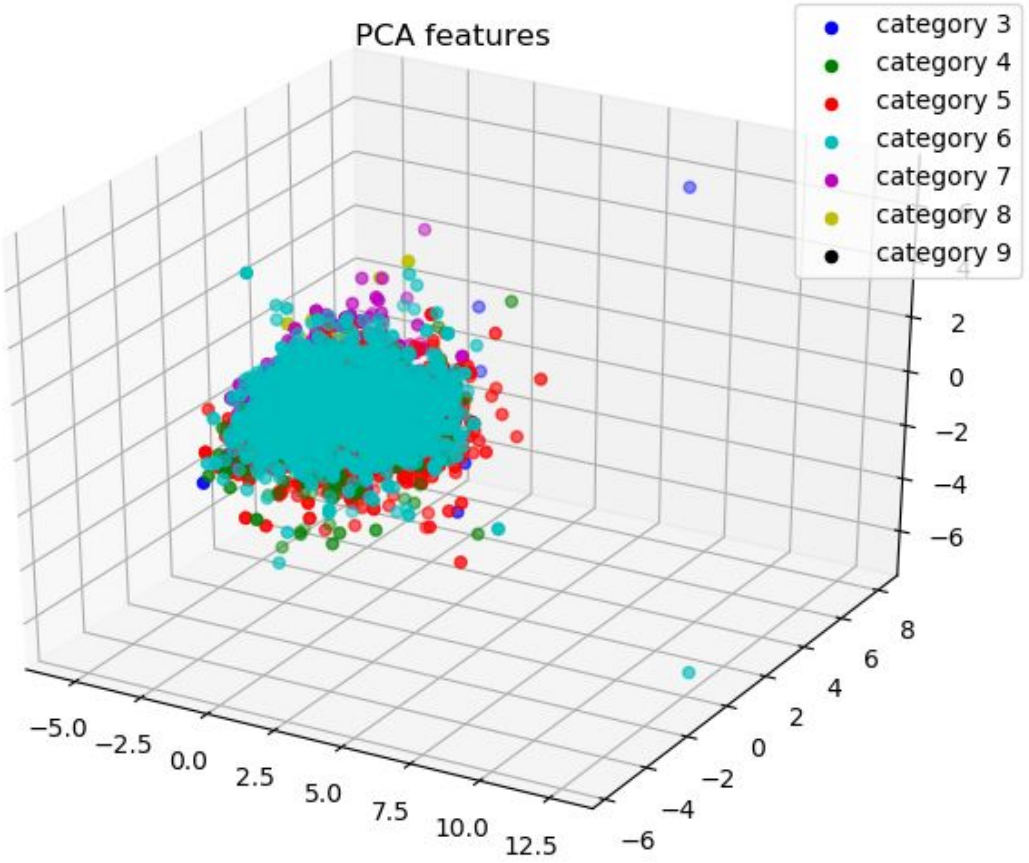
wstępnie z algorytmu PCA, który próbuje znaleźć taką podprzestrzeń rozciągniętą przez wektory będące kombinacją liniową pierwotnych zmiennych, aby odległości w rzucie ortogonalnym próbek na tę podprzestrzeń były jak najmniejsze. Zatem stara się znaleźć liniowe zależności między zmiennymi. Po sporządzeniu wykresów dla 2 cech o największej wariancji otrzymujemy (odpowiednio dla win czerwonych i białych):



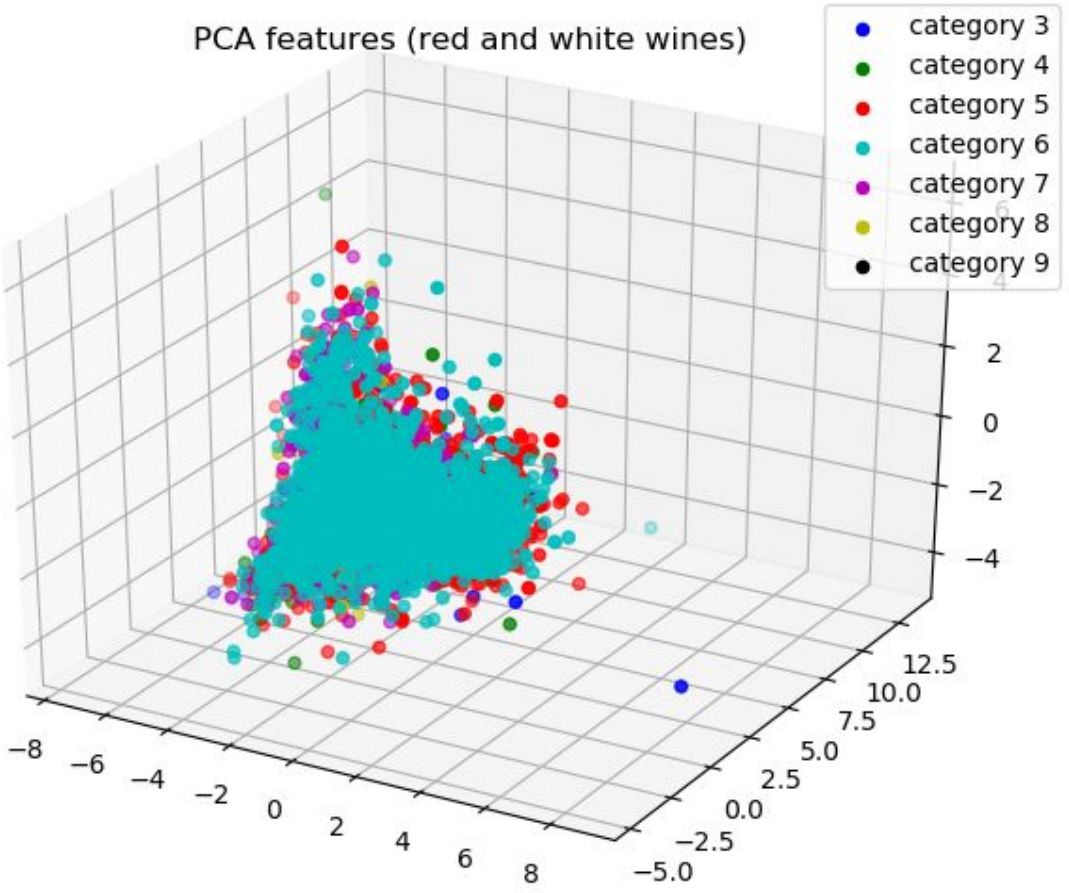


Natomiast dla wykresów zawierających o jedną cechę więcej, otrzymujemy następujące wykresy w przestrzeni trójwymiarowej:





Przeanalizujmy jeszcze wykres dla połączonych danych (wina czerwone + białe):



Tak więc okazuje się, że trudno dostrzec w danych wzorce, badając je jedynie pod kątem maksymalnie 3 cech otrzymanych z algorytmu PCA.

Powyższe wykresy wskazują jednak na pewne próbki wyraźnie odstające od innych. Mimo tego trudno dostrzec w nich jakieś powiązania klasowe. Z kolei po przyjrzeniu się danym z plików rzeczywiście niektóre próbki w ramach pewnych cech posiadają wartości lekko odstające (czasem nawet w wielu kolumnach jednocześnie).

Zbadajmy tym razem dane ściśle pod kątem znaczenia cech próbek. Sprawdźmy, czy wszystkie w równie istotny sposób wpływają na wynik. W tym celu użyjemy biblioteki sklearn i algorytmu opartego o klasyfikator Extra Trees (link do dokumentacji: http://scikit-learn.org/stable/modules/feature_selection.html#tree-based-feature-selection).

Po zastosowaniu algorytmu otrzymujemy procentowe wartości wpływu cechy na wynik:
0.07822734964819192, 0.09932466302436282, 0.08116938577860763,
0.08366978435806197, 0.08951895619957766, 0.08669808642672229,
0.08557114933894594, 0.08698026803106719, 0.07983210270460799,
0.08665724351539048, 0.14235101097446407.
Zatem na pierwszy rzut oka dane nie zawierają żadnych wyróżniających się cech.

3. Implementacja

```
n_trees       <- liczba drzew składających się na model
n_samples     <- liczba danych treningowych wybieranych dla każdego drzewa
n_features    <- liczba cech losowanych dla każdego z drzew
```

1. i := 0, M := null // M - zbiór drzew po i iteracjach
2. jeśli i >= n_trees: przejdź do kroku 9.
3. jeśli i = 0: S := zbiór n_samples losowo wybranych próbek, w innym wypadku: S := zbiór n_samples próbek, dla których błąd dla M był największy
4. wylosuj n_features cech
5. stwórz drzewo regresji D na podstawie danych składających się z próbek z pkt 3. z wyselekcjonowanymi cechami z pkt 4.
6. do zbioru drzew M dołącz drzewo D
7. i := i+1
8. przejdź do kroku 2.
9. zwróć model składający się z drzew z M

Dla zbioru drzew M odpowiedź dla danej próbki będzie liczona jako średnia po odpowiedziach dla każdego drzewa ze zbioru.

4. Strojenie parametrów

W trakcie uczenia dostępne będą 3 parametry strojenia: n_trees, n_samples oraz n_features (opisane w pkt. 3).