Projekt 1, Wstęp do uczenia maszynowego

Jan Pogłód

1. Wstęp

Celem projektu było stworzenie najlepszego modelu klasyfikatora do predykcji danych na zbiorze *artificial*. Do stworzenia tego modelu był dostępny zbiór danych treningowych wraz z etykietami oraz zbiór testowy bez odpowiadającym próbkom etykietom. Poniżej przedstawiam kroki analizy danych wraz ze stworzeniem najlepszego modelu klasyfikatora oraz wyniki klasyfikacji mierzone miarą *balanced_accuracy*.

2. Analiza zbioru danych

Zbiór danych *artificial* jest przykładem zbioru zbalansowanego, a więc dla każdej z klas jakie w nim występują (klasa pierwsza **-1**, klasa druga **1**) mamy równomiernie rozłożony rozkład próbek. Zbiór ten składa się z *30* numerycznych zmiennych i *2000* obserwacji kompletnych obserwacji, a więc nigdzie nie możemy zaobserwować braku danych. Wartości w ramce treningowej dla wszystkich zmiennych są z przedziału *84-829*, średnia z wszystkich obserwacji jest równa *489*, a odchylenie standardowe wynosi w granicach *4.5-113*, uśrednione: *54*. Oto trzy przykładowe zmienne o najmniejszej średniej oraz trzy o największej:

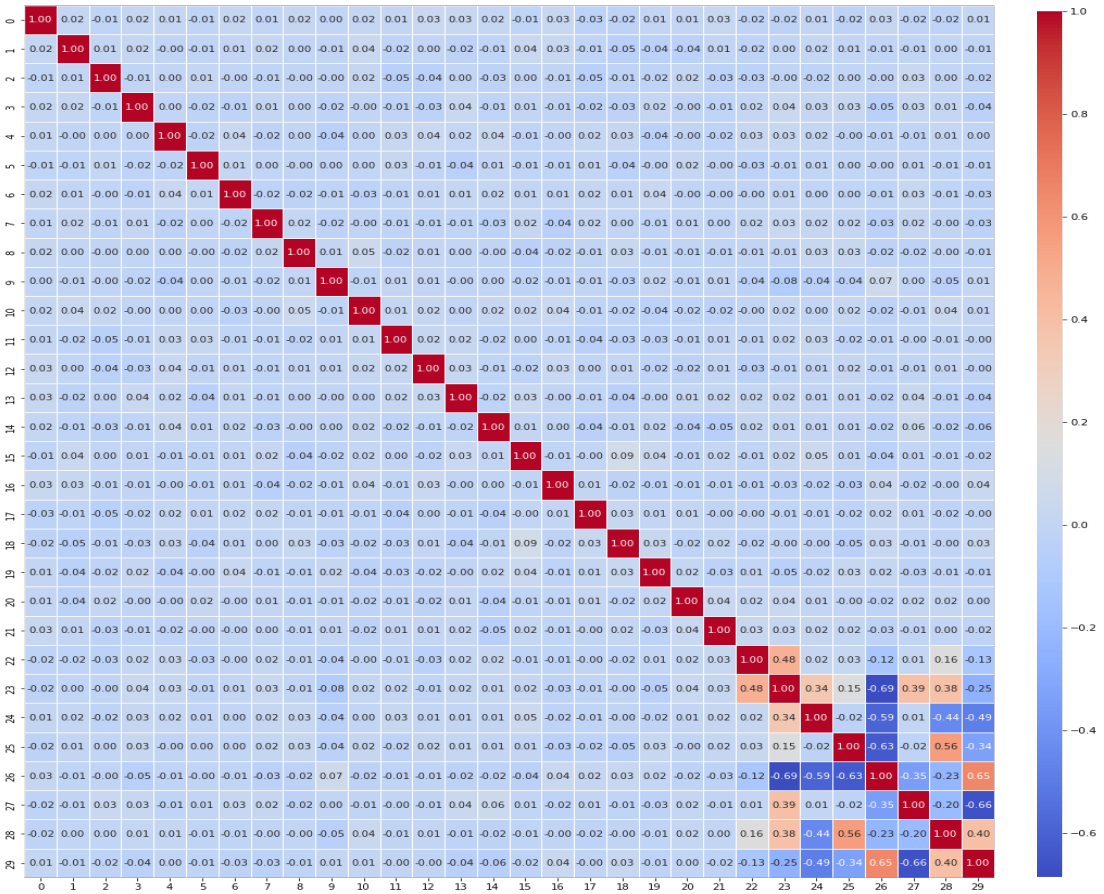
	Zmienna 1	Zmienna 2	Zmienna 3
Średnia	514.45	512.82	507.52
Odchylenie standardowe	113.47	96.95	53.68
Minimalna wartość	464	458	453
Maksymalna wartość	829	807	680
Mediana	513	512	507

Tabelka 1. Zestawienie wybranych statystyk dla 3 zmiennych o największej średniej z obserwacji

	Zmienna 1	Zmienna 2	Zmienna 3
Średnia	477.12	477.59	477.89
Odchylenie standardowe	4.54	5.17	7.79
Minimalna wartość	84	207	317
Maksymalna wartość	477	477	478
Mediana	493	495	505

Tabelka 2. Zestawienie wybranych statystyk dla 3 zmiennych o najmniejszej średniej z obserwacji

Jak widzimy zmienne są do siebie całkiem podobne, a ich statystyki nawet dla najbardziej granicznych nie różnią się znacząco. Zbiór testowy natomiast zawiera *600* obserwacji i nie znamy odpowiadających im etykiet. Średnie dla wszystkich obserwacji również są na podobnym poziomie *489.4*, a odchylenie standardowe w granicach *4.2-113*, uśrednione *30.59*. Za następny problem analizy danych postawiłem silność korelacji Pearsona między zmiennymi. Macierz korelacji dla wszystkich została przedstawiona poniżej.



Obraz 1. Heatmapa przedstawia siłę korelacji pomiędzy zmiennymi w ramce treningowej

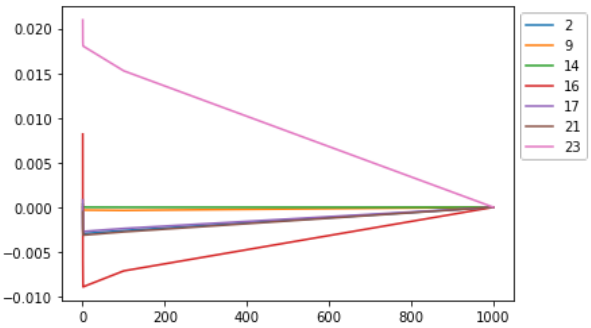
Z powyższej heatmapy możemy wysnuć kilka istotnych wniosków na temat treningowego zbioru danych. Widzimy z niej, że dla *22* pierwszych zmiennych ich wartości między sobą są ze sobą słabo skolerowane, głównie w granicach od *-0.06* do *0.06* w skali korelacji Pearsona. Jeżeli spojrzymy zaś na zmienne od *22* do *30*, to widzimy, że tam zmienne są już między sobą skolerowane umiarkowanie i osiągają korelacje aż na poziomie *-0,69*, czy *0,66*, co niemalże czyni je zmiennymi silnie skolerowanymi. Wnioskiem z całej powyższej heatmapy może być hipoteza, że niektóre ze zmiennych można pominąć przy tworzeniu i trenowaniu modeli do ostatecznej klasyfikacji danych, ponieważ wiele ze zmiennych może być mało istotne lub zaniżać ostateczny wynik na zbiorze testowym.

3. Selekcja zmiennych

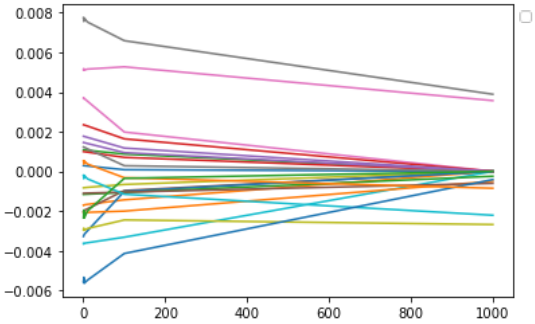
W celu najlepszej selekcji zmiennych dla ostatecznego modelu stworzyłem 5 różnych selektorów za pomocą różnych modeli klasyfikacji oraz porównałem ich wyniki, aby potwierdzić hipotezę, iż niektóre ze zmiennych mogą rzeczywiście nie nieść istotnych informacji dla modelu, a jedynie go przetrenować i nadmiernie dopasować się do zbioru treningowego. Poniżej przedstawiłem kolejne kroki tworzenia tych selektorów oraz na końcu wnioski i porównanie otrzymanych wyników.

3.1. Regresja logistyczna z regularyzacją L1

Jako pierwszy z modeli, którego użyłem do stworzenia selektora zmiennych był model regresji logistycznej z regularyzacją w normie L1, ponieważ posiada on zdolność, do zerowania współczynników dla mało istotnych zmiennych przy różnych wartościach parametru C. Za pomocą pięcio-krotnej walidacji krzyżowej z miarą *balanced_accuracy* przeprowadziłem metodą grid_search dobór najlepszych parametrów dla tego klasyfikatora i otrzymałem wówczas najlepszy parametr *C=0.01*. Dostałem, że dla takiego klasyfikatora aż 7 z współczynników przy zmiennych jest równe 0 przy dokładności czterech miejsc po przecinku. Poniżej po lewej stronie przedstawiłem, które z tych zmiennych model wyzerował, zaś po prawej stronie przedstawione są zmienne, które nie zostały wyzerowane.



Obraz 2. 7 zmiennych, które zostały wyzerowane przez klasyfikator regresji z najlepiej dobranymi parametrami

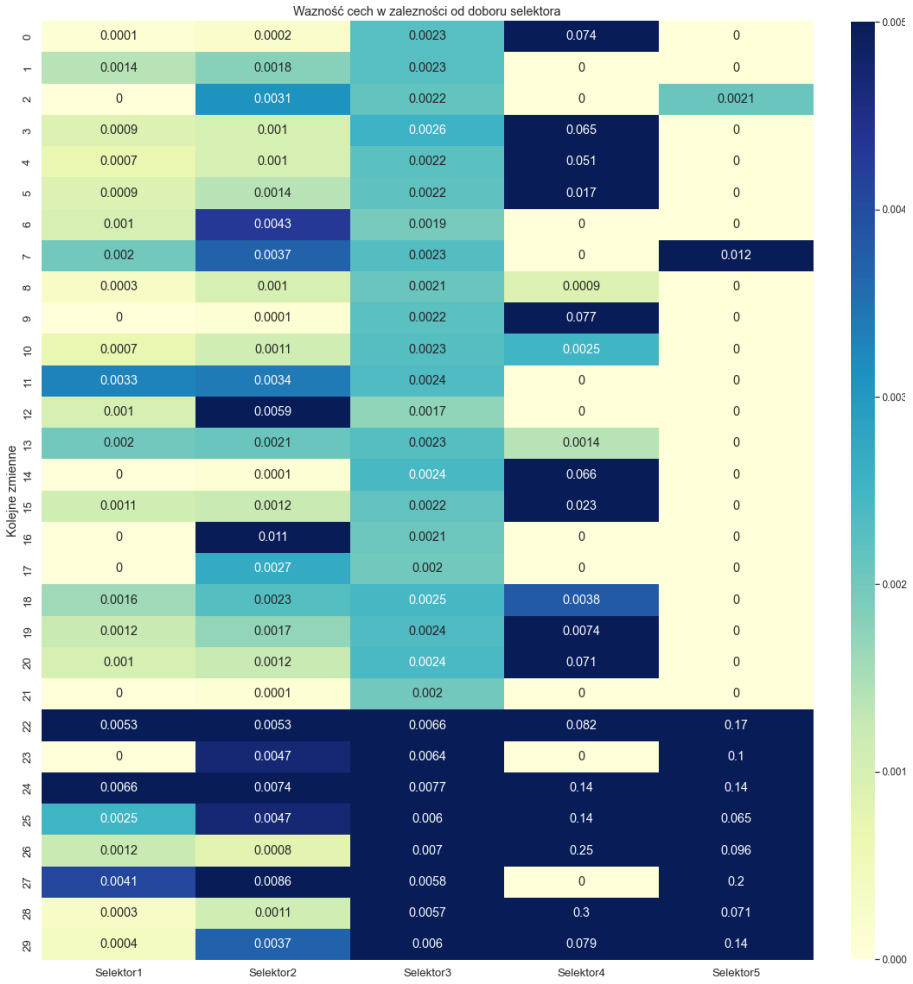


Obraz 3. Pozostałe zmienne, których model tak nie zakwalifikował

3.2. Pozostałe selektory zmiennych

- Jako drugie narzędzie do selekcji zmiennych użyłem ponownie regresji logistycznej, lecz tym razem z domyślnymi parametrami, a więc na przykład: z regularyzacją L2, *C=1 max_iter=100* i metodą optymalizacji Limited BFGS.
- Trzecim selektorem zmiennych został las losowy. Ponownie, jak i w kolejnych modelach użyłem pięcio-krotnej walidacji krzyżowej z miarą *balanced_accuracy*, która nadała mu optymalne parametry odpowiednio *max_depth=20, min_samples_leaf=2* i *n_estimators=200*.
- Aby stworzyć czwarty selektor zmiennych wykorzystałem model wektorów podpierających wraz ze znalezionymi optymalnie parametrami *C=1, kernel='rbf', gamma=0,1*
- Do stworzenia ostatniego z selektorów wykorzystałem zwykłe drzewo decyzyjne z parametrami *max_depth=5, min_samples_split=15*, które posłużyło mi jedynie do celów porównawczych.

Aby przetestować te selektory zmiennych stworzyłem pięć różnych modeli, za każdym razem używając tego samego klasyfikatora – Regresji logistycznej z regularyzacją *L2* i parametrem *C=0.001* dobranym metodą jak poprzednio. Do normalizacji danych użyłem funkcji *StandardScaler*(), która przekształca zmienne w ramce w taki sposób, aby miały średnią równą 0 i odchylenie standardowe, które wynosi 1. Poniżej przedstawiłem wyniki ukazujące miarę *balanced_accuracy* dla każdego z tego selektora i takiego samego klasyfikatora na zbiorze treningowym. Dalej przedstawiłem również jak kolejne selektory selekcjonują zmienne, a więc które zmienne według wyżej wymienionych selektorów są uznane za bardziej istotne dla modelu, a które za mniej. Poniższa heatmapa dostarcza nam bardzo dużo informacji o wszystkich zmiennych występujących w ramce danych treningowych, a przede wszystkim cechach każdego z selektorów i ilości mało istotnych cech.



Klasyfikator regresji logistycznej +	Balanced accuracy
Selektor1 (Regresja logistyczna L1)	61,75%
Selektor2 (Regresja logistyczna domyślna)	61,4%
Selektor3 (Las losowy)	60,25%
Selektor4 (SVC)	60,85%
Selektor5 (Drzewo decyzyjne)	59,85%
Bez selekcji (Sam model klasyfikacji)	60,5%

Tabelka 3. Zestawienie wyników na mierze balanced_accuracy dla kolejnych selektorów na zbiorze treningowym przy tym samym klasyfikatorze Regresji logistycznej z regularyzacją L2

Obraz 4.Heatmapa po lewej stronie przestawia wartość współczynnika odpowiedniego selektora przy kolejnej zmiennej. Im wartość współczynnika jest niższa tym zmienna została zakwalifikowana jako mniej istotna dla całości modelu, a im wartość ta jest większa, tym zmienna może nieść więcej informacji indywidualnych dla modelu i być przez to bardziej istotna

Na podstawie zależności – im ciemniejszy kolor tym zmienna jest bardziej istotna – można zauważyć przykładowo, że wszystkie selektory średnio uznają, że cechy *22-29* są bardzo istotne dla modelu, zaś szczególnie *21, 17* są bardzo mało istotne co potwierdził każdy z selektorów. Analizując całą heatmapę i wyniki selektorów na zbiorze testowym z modelem klasyfikacji regresji logistycznej, możemy teraz wybrać najlepszy selektor dla ostatecznego modelu. Widzimy, że wybranie selektora *5* lub selektora *4* nie będzie dobrą decyzją, ponieważ zbyt dużo współczynników zostało przez nie wyzerowane. Choć selektor z modelem wektorów podpierających wyliczył *19* niezerowych współczynników, to nie są ich wysokie wartości (szczególnie dla zmiennych przed *21*) potwierdzone przez pozostałe selektory, zaś model z drzewem decyzyjnym uznał, że należy wybrać jedynie *8* ostatnich zmiennych i dwie inne, co jest zbyt mało skutecznym wyborem w porównaniu z pozostałymi selektorami. W przeciwieństwie do nich, regresja logistyczna sprawdza się bardzo dobrze jako selektor zmiennych i w przypadku różnych sposobów regularyzacji i parametrów, wyniki są do siebie podobne. W porównaniu do nich widzimy, że las losowy również podobnie osiągnął wartości współczynników oraz co istotniejsze nie wyzerował on wszystkich współczynników, lecz przypisał każdej wagi, a ostatnie zmienne uznał za najbardziej istotne. Na podstawie tego badania, podjąłem decyzję, że jako selektor zmiennych wybiorę model lasu losowego oraz model z reularyzacją *L1* z ich optymalnymi parametrami oraz, że będę usuwał 10 zmiennych za pomocą funkcji SequentialFeatureSelector(), bo taką liczbę możemy dobrać analizując całą powyższą heatmapę.

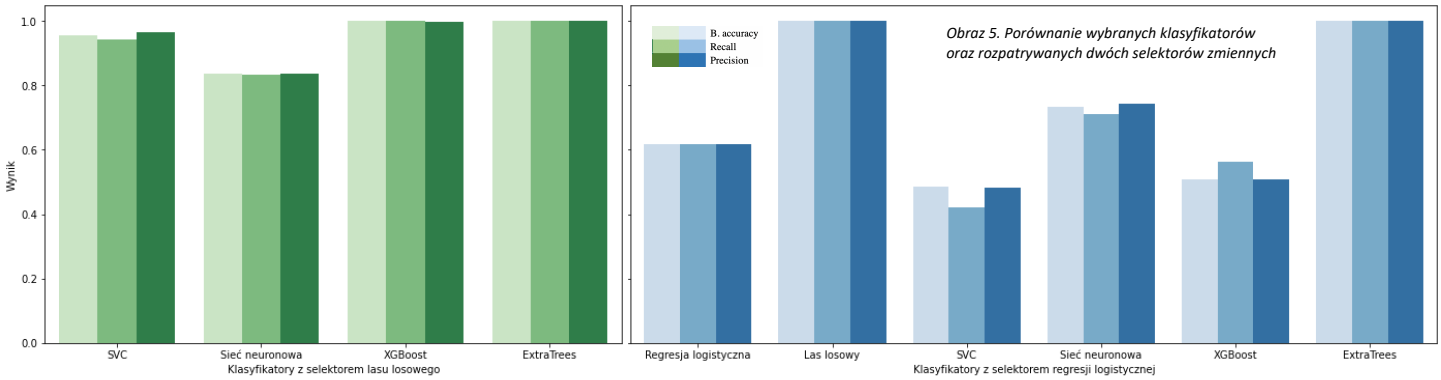
4. Kroki do osiągnięcia najlepszego modelu

Aby osiągnąć najlepsze wyniki predykcji na danych treningowych, stworzyłem sześć różnych modeli klasyfikacji i za pomocą pipelinów dodałem do każdego standaryzację danych, selektor zmiennych jako las losowy lub regresję wybrane jak wyżej z funkcją sekwencyjnego wyboru oraz grid-seach z 5-krotną walidacją krzyżową. Pierwsze trzy z modeli to regresja logistyczna L2, Las losowy i SVC z poprzedniego punktu. Oprócz nich dodałem również sieć neuronową – MLPClassifier, XGBClassifier z pakietu xgboost oraz klasyfikator extraTrees.

	Parametr 1	Parametr 2	Parametr 3
Sieć neuronowa	*alpha: 0.01*	*hidden_layer_sizes: (10,)*	*learning_rate_init: 0.1*
XGBClassifier	*learning_rate: 0.1*	*max_depth': 7*	*n_estimators: 200*
extraTrees	*max_depth: 20aa*	*min_samples_leaf:1*	*n_estimators: 200*

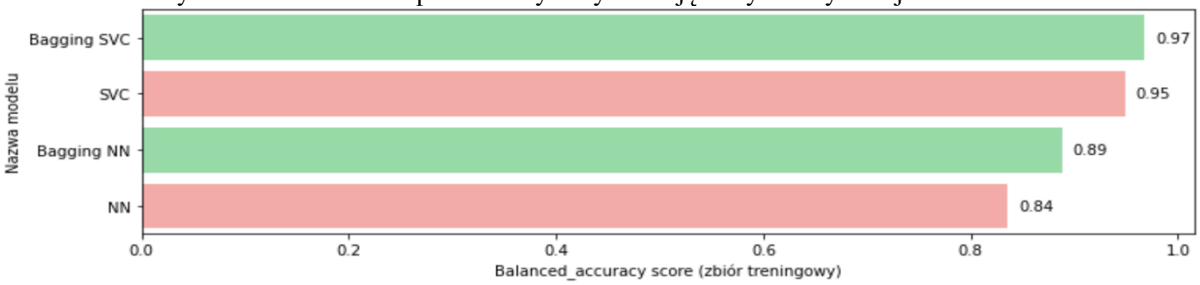
Tabelka 4. Optymalne parametry dobrane dla trzech nowych modeli klasyfikacji za pomocą funkcji grid_search

Dla powyższych sześciu modeli przetestowałem, jak zdolność predykcyjna na zbiorze treningowym zależy od doboru selektora wybierającego *20* najlepszych zmiennych. Okazało się, że las losowy (kolor zielony) sprawdza się znacznie lepiej jako selektor niż regresja logistyczna (kolor niebieski) w przypadku xgboost, SVC czy sieci neuronowej. Natomiast modele takie jak ExtraTrees czy las losowy wydają się bardzo dobrze wpasowywać w zbiór danych, co potwierdza ich pewny recall, precyzja i balanced_accuracy. Warto zaznaczyć, że modele które sprawdziły się dobrze na danych treningowych w przypadku użycia selektora lasu losowego to xgboost lub SVC.

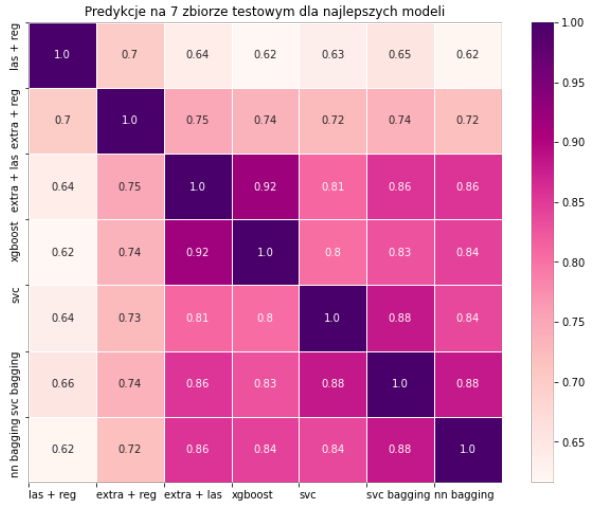


Obraz 5. Porównanie wybranych klasyfikatorów oraz rozpatrywanych dwóch selektorów zmiennych

Analizując powyższe wykresy stwierdziłem, że las losowy dobrze spełnia swoją rolę jako selektor zmiennych, a wytrenowane na nim modele sieci neuronowej i SVC mogłyby jeszcze zostać ulepszone za pomocą metody bagging, która dokonuje agregacji poprzez bootstrapowe wybieranie próbek. Przeprowadziłem bagging dla nich z *50* estymatorami i dostałem poniższe wykresy ukazujące wynik użycia tej techniki.



W następnym kroku siedem z klasyfikatorów, które okazały się być najlepsze na powyższych wykresach postanowiłem porównać ze sobą na zbiorze testowym. Wszystkie z tych modeli dokonały predykcji klas na tym zbiorze i metodą każdy z każdym przedstawiłem bliskość ich wyników predykcyjnych na poniższej heatmapie.



Widzimy na niej, że model lasu losowego oraz ExtraTrees z selektorem regresji logistycznej wydają się mieć względnie do innych modeli niższą zdolność do predykcji na zbiorze testowym, w przeciwieństwie do pozostałych pięciu, których wyniki predykcyjne były bliskie sobie i dokładne z minimalnie 80% prawdopodobieństwem. Postanowiłem zostawić jednak jeden model z innym selektorem dla zmniejszenia zależności i dla ostatecznych czterech modeli (bo poza svc, który nie dostał baggingu i sieci neuronowej mającej niskie wyniki na treningu – 0.89) przygotowałem klasyfikator głosujący z parametrem *voting = 'soft'*, aby połączył on otrzymane zdolności predykcyjne. W ten sposób powstał model VotingClassifier dla poniższych z czterech modeli:

- **SVC (po baggingu) z selektorem lasu losowego**
- **xgboost z selektorem lasu losowego**
- **ExtraTrees z selektorem lasu losowego**
- **ExtraTrees z selektorem regresji**

5. Podsumowanie

Dzięki powyższym krokom otrzymaliśmy klasyfikator, który będzie potrafił dokonać najlepszej predykcji na zbiorze testowym. Widzimy, że zastosowanie technik jak selekcja zmiennych, standaryzacja, bagging czy duża różnorodność modeli wzmacnia możliwości predykcyjne i jest niezbędna w problemie klasyfikacji. Za ich pomocą udało nam się otrzymać klasyfikator, który ostatecznie na 5% danych testowych osiągnął dokładność 93,(3)%.