Wstęp do uczenia maszynowego - projekt 1

Karol Degórski i Adrian Kamiński

1 Opis problemu

Problem postawiony w pracy projektowej nr 1 polega na przewidywaniu jaką ocenę końcową (przyjmującą wartości od 0 do 20) otrzyma dany uczeń szkoły średniej w Portugalii za pomocą danych demograficznych oraz lifestylowych. Zgodnie z informacją zwartą w zadaniu problem ten można traktować zarówno jako problem klasyfikacji jak i regresji.

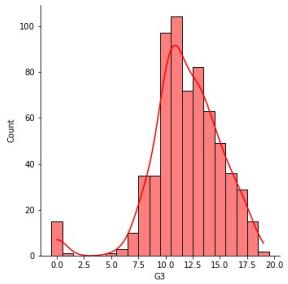
2 Opis zbioru danych

W pracy projektowej wykorzystaliśmy zbiór danych **School Grades**, pochodzący ze strony apispreadsheets.com. Dane zawarte w tym zbiorze dotyczą osiągnięć uczniów i ich ocen oraz zawierają ich dane demograficzne, społeczne i inne szeroko związane edukacją. Zawiera on informacje o uczniach dwóch szkół średnich w Portugalii: szkoły im. Gabriela Pereiry oraz im. Mousinho da Silveira. Informacje te zostały zebrane za pomocą raportów pochodzących ze szkół oraz z kwestionariuszy. Rozmiar tego zbioru nie jest duży, bo zawiera tylko 649 rekordów, natomiast ma relatywnie sporo kolumn, bo aż 33. Ponadto nie zawiera braków danych, co znacząco ułatwia modelowanie i preprocessing. W tabeli (3) w załącznikach prezentujemy opis cech zawartych w tym zbiorze danych.

Jak możemy zauważyć oraz doczytać w opisie danych zmienna celu, jaką jest ocena końcowa danego ucznia (G3), jest silnie skorelowana ze zmiennymi G1 (ocena po pierwszym okresie) oraz G2 (ocena po drugim okresie). Nie jest to dziwne ponieważ z reguły uczeń, na koniec powinien dostać ocenę podobną do tych po poszczególnych okresach (uczeń bardzo dobry, zwykle cały czas będzie osiągał bardzo dobre wyniki, a uczeń słaby, również nie poprawi się z dnia na dzień). Dlatego też warto rozważyć modelowanie oceny końcowej G3 zarówno z wykorzystaniem informacji o otrzymanych ocenach G1 i G2, jak i również z pominięciem tej informacji. Dzięki temu możemy przetestować jak dobrze nasze modele będą w stanie przewidywać ocenę końcową bazując tylko na informacjach demograficznych, społecznych oraz innych związanych ze szkołą (bez informacji o poprzednich ocenach). Zbudowanie takich modeli może być szczególnie przydatne dla szkół, które dopiero przyjmują uczniów i nie posiadają informacji o cząstkowych ocenach.

3 Opis preprocesingu zastosowanego w modelu

W celu lepszego zapoznania się ze zbiorem danych przeprowadziliśmy eksploracyjną analizę danych. Okazało się, że oceny G1, G2 i G3 mają rozkłady normalne i są one niemalże identyczne. Zmienna celu G3 tak jak wcześniej zauważyliśmy ma rozkład normalny (można to zauważyć na rysunku obok), ale jednak widać wyraźny spadek między wartościami 9, a 10. Biorąc pod uwagę fakt, że 10 punktów stanowi 50% możemy stwierdzić, że wynika to z chęci nauczycieli do podciągnięcia oceny, tak aby więcej uczniów osiągnęło co najmniej połowę. Zmienna ta zawiera również trochę wartości 0. Jako, że w ramce danych nie ma wartości `None`, być może w taki sposób zostały zastąpione. Jednakże mimo to, możemy uznać ten rozkład za normalny.



Ponadto zauważyliśmy, że nasz zbiór danych zawiera dużo zmiennych dyskretnych, które mogą być pomocne przy modelowaniu, ale wymagają użycia kodowania, o którym piszemy w dalszej części raportu. Na etapie eksploracyjnej analizy danych zauważyliśmy również, że informacja o nieobecnościach (absences) ma rozkład prawostronnie skośny oraz dużo wartości zerowych. Również okazało się, że kobiety mają więcej lepszych ocen G3 niż mężczyźni oraz, że uczniowie mieszkający w miastach mają wyższą ocenę G3. Kolejną ciekawą informacją jest to, że uczniowie ze szkoły Gabriela Pereira mają trochę lepsze oceny oraz że uczniowie, którzy zamierzają podjąć studia otrzymują przeważnie lepsze oceny.

Zbadaliśmy również korelacje między zmiennymi. Zauważyliśmy, że jest pewna korelacja również na zmiennych dyskretnych - na przykład całkiem widoczną korelację widać miedzy zmiennymi Medu - Fedu. Po uzyskaniu tych informacji mogliśmy zabrać się do przeprowadzenia kodowania zmiennych tekstowych na liczbowe oraz innej inżynierii cech. Wykorzystaliśmy dwa encodery: `LabelEncoder` i `OneHotEncoder`. Wykorzystaliśmy je na kolumnach zawierających dane tekstowe: porządkowe (ordinal - `LabelEncoder`) i nominalne (nominal - `OneHotEncoder`). Stworzyliśmy 4 różne zestawy danych, aby móc porównywać osiągane przez nie wyniki:

- *with_ exams* - zbiór zawierający wszystkie kolumny
- *without_ exams* - zbiór nie zawierający kolumn G1 i G2
- *with_ exams_ optimized* - zbiór zawierający kolumny G1 i G2, ale zoptymalizowany (niektóre kolumny usunięte oraz przeprowadzone grupowanie)
- *without_ exams_ optimized* - zbiór nie zawierający kolumn G1 i G2 i zoptymalizowany (niektóre kolumny usunięte oraz przeprowadzone grupowanie)

W zbiorach zoptymalizowanych informacja o nieobecnościach została pogrupowana w przedziały $[0, 4], (4, 8], (8, 12], (12, \infty)$. Informacja o wieku ucznia została przeskalowana z użyciem `MinMaxScaler`. Ponadto usunęliśmy kolumny: romantic, famrel, paid, famsup, Pstatus, famsize, które po eksploracyjnej analizie danych wydawały się nie wnosić dodatkowych informacji. Natomiast w zbiorach niezoptymalizowanych użyliśmy transformacji logarytmicznej dla zmiennej absences. Następnie dokonaliśmy podziału zbioru na treningowy i testowy, gdzie zbiór testowy stanowił 30% całości. Jednakże zdecydowaliśmy się też trenować modele na całym zbiorze i wykorzystywać pięciokrotną kroswalidację.

Kolejną ważną kwestią jest wybór metryki, do oceny działania naszego modelu. Zdecydowaliśmy się na miarę `RMSE`, czyli pierwiastek z błędu średniokwadratowego. Miara ta wydaje się być odpowiednią z uwagi na swego rodzaju ciągłość zmiennej celu, jaką jest ocena końcowa G3. Oceniając model najistotniejsze jest dla nas, czy przewidywana przez dany model ocena znacząco różni się od tej prawdziwej. Niekoniecznie ważne jest czy dokładnie przewidzieliśmy ocenę. Następnie przeszliśmy do trenowania wybranych modeli predykcyjnych.

4 Opis wyboru modelu oraz sposobu doboru hiperparametrów

Na początku potraktowaliśmy problem przewidywania oceny końcowej G3 jako problem klasyfikacji, dlatego też wykorzystaliśmy następujące modele:

- `las losowy`,
- `XGBoost`,
- `Stacking Classifier` - składający się z `XGBoosta` i `lasu losowego` z użyciem `LogisticRegression` jako finalnego estymatora,
- `Voting Classifier` - składający się z `XGBoosta` i `lasu losowego` z głosowaniem `soft`,
- `sieć neuronowa typu MLP`

Jednakże później doszliśmy do wniosku, że warto potraktować ten problem jako problem regresji i wykorzystaliśmy model regresji liniowej. W przypadku regresji liniowej, aby zachować fakt iż oceny są wartościami całkowitymi, nadpisaliśmy klasę `LogisticRegression` tak, by zaokrąglała otrzymane oceny (funkcją `numpy.round`).

Do wyboru hiperparametrów wykorzystaliśmy `RandomizedSearchCV` (z trzykrotną kroswalidacją oraz z 20 iteracjami). Ostatecznie wybrane hiperparametry przy modelach ztunningowanych można znaleźć w tabeli (4).

5 Podsumowanie

W tablicy (1) przedstawiliśmy wartości błędu `RMSE` uzyskane bez użycia kroswalidacji. Możemy zauważyć, że `Stacking Classifier` działa najlepiej dla zbioru danych zawierającym dane o ocenach G1 i G2. Możemy również zauważyć, że przeprowadzona przez nas optymalizacja dla tego modelu zmniejszyła wartość błędu o około 0,09. Nie jest to może jakieś znaczące usprawnienie, ale jednak nasz model mniej się myli i z większą dokładnością jesteśmy w stanie przewidywać ocenę końcową G3.

Z kolei dla zbiorów danych nie zawierających informacji o ocenach G1 i G2, najlepszym klasyfikatorem okazał się `Voting Classifier`. Biorąc pod uwagę powyższe wyniki okazuje się, że klasyfikatory wykorzystujące wyniki uzyskane z innych modeli i łączące je radzą sobie najlepiej.
Zwizualizowane działanie najlepszych modeli dla każdego ze zbiorów (basic) można znaleźć na rysunku (1)

Traktując problem jako problem regresji i wykorzystując jeden z najbardziej podstawowych modeli uczenia maszynowego jakim jest regresja liniowa również możemy szybko dojść do bardzo dobrych wyników (model szybko się uczy i nie wymaga doboru hiperparametrów). Czas wykonywania się poszczególnych etapów można znaleźć w tabeli (5).

Co ciekawe po jej wykonaniu okazało się, że to właśnie regresja linowa uzyskała najlepsze wartości miary RMSE ze wszystkich modeli, zarówno na zbiorach danych zawierających informacje o ocenach G1 i G2, jak i na zbiorach nie zawierających tej informacji. Z uwagi na fakt, że regresja jest jedynym z wykorzystanych przez nas modeli, który wykorzystuje informacje o naturalnej kolejności danych w klasach, mogliśmy się spodziewać, że to właśnie ten model będzie najbardziej optymalny.

Wyniki w postaci barplotów można znaleźć na rysunku (2) - wyniki bez kroswalidacji oraz na rysunku (3) - wyniki z kroswalidacją.

Model	Feature set	with_exams	without_exams	with_exams optimized	without_exams optimized
Las losowy	**basic**	1.609268	3.244720	1.581949	3.199359
	tunned	1.419642	3.256552	1.614041	3.233637
XGBoost	**basic**	1.320451	2.980277	1.264911	3.099214
	tunned	1.373747	3.402865	1.337813	3.418652
Stacking Classifier	**basic**	1.248589	3.066778	1.152478	3.027227
	tunned	1.632993	3.261272	1.609268	3.226493
Voting Classifier	**basic**	1.312660	2.977695	1.285022	2.972524
	tunned	1.362501	3.213752	1.341641	3.239183
Sieć neuronowa typu MLP	**basic**	1.755577	3.293350	1.860521	3.355057
	tunned	1.623545	3.039906	1.542559	3.154972
Regresja liniowa	**basic**	1.272994	3.000855	1.246534	3.011091

Tablica 1: Wartości RMSE modeli bez użycia kroswalidacji

W tablicy (2) prezentujemy wyniki uzyskane z użyciem kroswalidacji.

Model	Feature set	with_exams	without_exams	with_exams optimized	without_exams optimized
Las losowy	**basic**	1.727201	3.246242	1.797661	3.324900
	tunned	1.533115	3.122996	1.637920	3.091979
XGBoost	**basic**	1.469001	3.626783	1.436691	3.789112
	tunned	1.478754	3.285428	1.495278	3.221975
Stacking Classifier	**basic**	1.408419	3.154920	1.376968	3.123955
	tunned	1.670129	3.252998	1.584846	3.259000
Voting Classifier	**basic**	1.456462	3.587236	1.384069	3.561761
	tunned	1.379078	3.184158	1.519558	3.159756
Sieć neuronowa typu MLP	**basic**	1.908066	3.445904	1.981687	3.305271
	tunned	1.886850	3.028987	1.666282	3.088503
Regresja liniowa	**basic**	1.369499	2.767701	1.363295	2.740264

Tablica 2: Wartości RMSE modeli z użyciem kroswalidacji

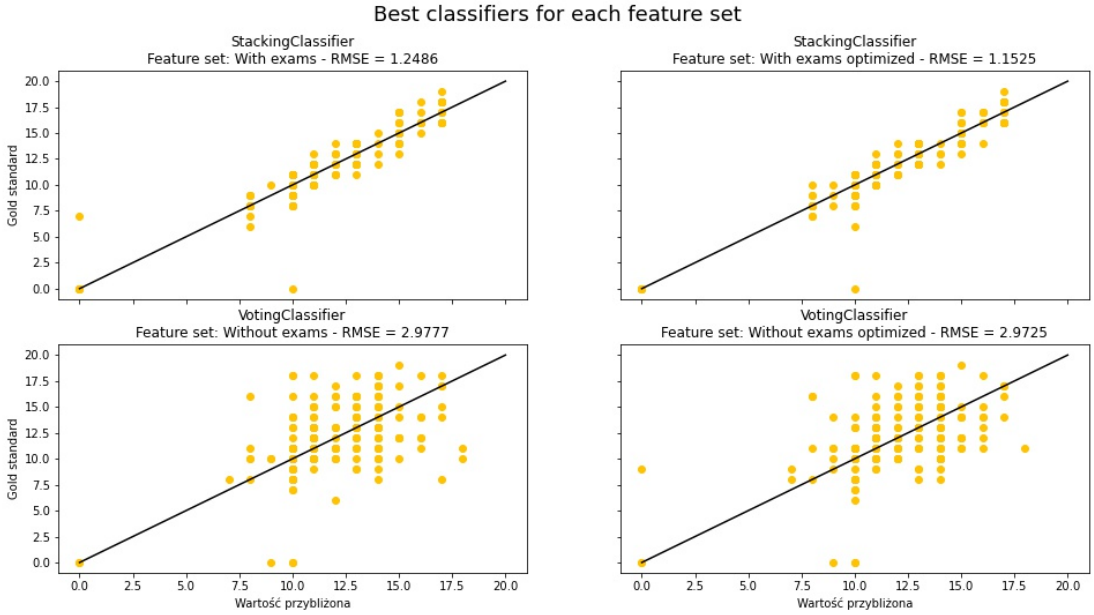
6 Załączniki

nazwa	typ	opis
school	string	szkoła ucznia - wartości binarne: "GP" Gabriel Pereira, "MS" Mousinho da Silveira
sex	string	płeć ucznia - wartości binarne: "F" żeńska, "M" męska
age	integer	wiek ucznia - wartości liczbowe od 15 do 22
address	string	adres studenta - wartości binarne: "U" miejski, "R" wiejski
famsize	string	rozmiar rodziny - wartości binarne: "LE3" mniej lub równo 3, "GT3" więcej niż 3
Pstatus	string	stan mieszkaniowy rodziców - wartości binarne: "T" mieszkają razem, "A" mieszkają osobno
Medu	integer	wykształcenie mamy ucznia - wartości liczbowe: 0 - brak, 1 - ukończona szkoła podstawowa do 4 klasy, 2 ukończona szkoła podstawowa do 9 klasy, 3 - średnie, 4 - wyższe)
Fedu	integer	wykształcenie taty ucznia - wartości liczbowe: 0 - brak, 1 - ukończona szkoła podstawowa do 4 klasy, 2 ukończona szkoła podstawowa do 9 klasy, 3 - średnie, 4 - wyższe)
Mjob	string	praca mamy ucznia - możliwe wartości: "at_home" w domu, "health" służba zdrowia, "services" usługi, "teacher" nauczyciel, "other" inna
Fjob	string	praca taty ucznia - możliwe wartości: "at_home" w domu, "health" służba zdrowia, "services" usługi, "teacher" nauczyciel, "other" inna
reason	string	powód wybrania tej szkoły - możliwe wartości: "home" blisko domu, "course" program nauki, "reputation" renoma, "other" inny
guardian	string	opiekun ucznia - możliwe wartości: "mother" mama, "father" tata, "other" inny
traveltime	integer	czas dojazdu z domu do szkoły - wartości liczbowe: 1 - do 15 minut, 2 - od 15 do 30 minut, 3 - od 30 minut do 1 godziny, 4 - ponad 1 godzinę
studytime	integer	tygodniowy czas nauki ucznia - wartości liczbowe: 1 - mniej niż 2 godziny, 2 - od 2 do 5 godzin, 3 - od 5 do 10 godzin, 4 - więcej niż 10 godzin
failures	integer	liczba wcześniej niezdanych przedmiotów - wartości liczbowe: 0, 1, 2, 3, lub 4 jeśli liczba ta jest większa równa 4)
schoolsup	string	informacja czy uczeń ma dodatkowe wsparcie w zakresie nauki - wartrtości binarne: tak, nie
famsup	string	informacja czy uczeń ma wsparcie w rodzinie w zakresie nauki - wartrtości binarne: tak, nie
paid	string	informacja czy uczeń chodzi na płatne zajęcia dodatkowe z zakresu szkolnych przedmiotów - wartrtości binarne: tak, nie
activities	string	informacja czy uczeń chodzi na zajęcia pozaszkolne - wartrtości binarne: tak, nie
nursery	string	informacja czy uczeń uczęszczał do przedszkol - wartrtości binarne: tak, nie
higher	string	informacja czy uczeń chce podjąć studia - wartrtości binarne: tak, nie
internet	string	informacja czy uczeń ma dostęp do internetu w domu - wartrtości binarne: tak, nie
romantic	string	informacja czy uczeń jest w związku - wartrtości binarne: tak, nie
famrel	integer	jakość relacji rodzinnych - wartości liczbowe: od 1 - bardzo złe do 5 - świetne)
freetime	integer	czas wolny po szkole - wartości liczbowe: od 1 - bardzo mało do 5 - bardzo dużo)
goout	integer	wyjścia ze znajomymi - wartości liczbowe: od 1 - bardzo mało do 5 - bardzo dużo)
Dalc	integer	dzienne spożycie alkoholu - wartości liczbowe: od 1 - bardzo niskie do 5 - bardzo wysokie)
Walc	integer	weekendowe spożycie alkoholu - wartości liczbowe: od 1 - bardzo niskie do 5 - bardzo wysokie)
health	integer	weekendowe spożycie alkoholu - wartości liczbowe: od 1 - bardzo niskie do 5 - bardzo wysokie)
absences	integer	liczba nieobecności w szkole - wartości liczbowe: od 0 do 93
G1	integer	ocena po pierwszym okresie - wartości liczbowe: od 0 do 20
G2	integer	ocena po drugim okresie - wartości liczbowe: od 0 do 20
G3	**integer**	**ocena końcowa - wartości liczbowe: od 0 do 20**

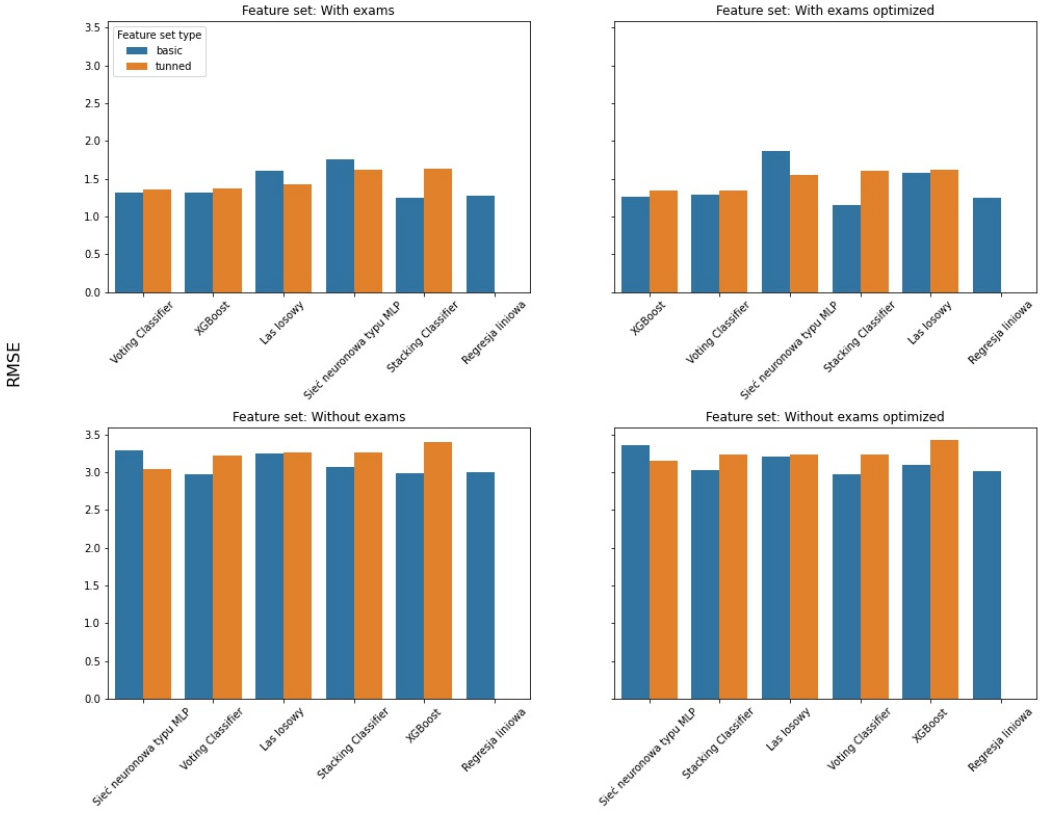
Tablica 3: Opis wszystkich cech zawartych w zbiorze danych

Model	Feature set	Hyperparameters
Las losowy	**With exams**	{'min_samples_split': 6, 'max_features': 10, 'max_depth': 6, 'criterion': 'entropy'}
	With exams optimized	{'min_samples_split': 6, 'max_features': 10, 'max_depth': 6, 'criterion': 'entropy'}
	Without exams	{'min_samples_split': 4, 'max_features': 7, 'max_depth': 3, 'criterion': 'entropy'}
	Without exams optimized	{'min_samples_split': 4, 'max_features': 7, 'max_depth': 3, 'criterion': 'entropy'}
XGBoost	**With exams**	{'booster': 'dart', 'learning_rate': 0.008250781339763612, 'max_without_exams_optimizedepth': 7}
	With exams optimized	{'booster': 'gbtree', 'learning_rate': 0.018289492658229912, 'max_without_exams_optimizedepth': 4}
	Without exams	{'booster': 'gblinear', 'learning_rate': 0.08146397256356941, 'max_without_exams_optimizedepth': 10}
	Without exams optimized	{'booster': 'gblinear', 'learning_rate': 0.05426333128918984, 'max_without_exams_optimizedepth': 10}
Sieć neuronowa typu MLP	**With exams**	{'activation': 'relu', 'alpha': 0.03899660137584881, 'batch_size': 60, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'momentum': 0.41893439668259447, 'solver': 'adam'}
	With exams optimized	{'activation': 'logistic', 'alpha': 0.013382821925482681, 'batch_size': 80, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive', 'momentum': 0.7110956862030287, 'solver': 'lbfgs'}
	Without exams	{'activation': 'tanh', 'alpha': 0.00010898794623297014, 'batch_size': 10, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'momentum': 0.4936352725997194, 'solver': 'sgd'}
	Without exams optimized	{'activation': 'tanh', 'alpha': 0.00010898794623297014, 'batch_size': 10, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'constant', 'momentum': 0.4936352725997194, 'solver': 'sgd'}

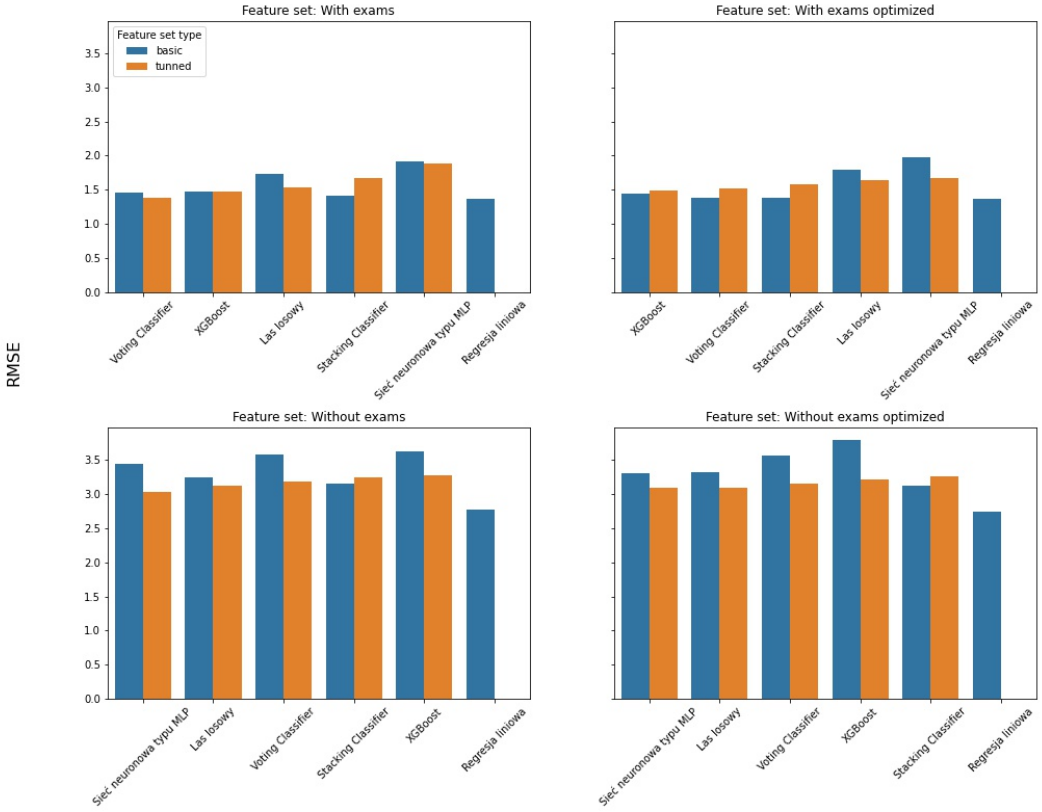
Tablica 4: Wybrane przez nas optymalne hiperparametry



Rysunek 1: Caption



Rysunek 2: Caption



Rysunek 3: Caption

Etap	Czas wykonania (MM:SS)
Data preprocessing	00 : 03
Fitting basic models	00 : 55
Tunning and fitting models	03 : 48
Cross validation	02 : 49
Sum	07 : 37

Tablica 5: Czas wykonania kolejnych etapów