Faculty of Mathematics and Information Science

WARSAW UNIVERSITY OF TECHNOLOGY

Automatyczne uczenie maszynowe

Przygotowanie modeli o największej zdolności predykcji dla sztucznie wygenerowanego zbioru danych

---

Szymon Pawlonka
Łucja Żukowska

WARSZAWA, 2023

1 Wstęp

Celem zadania było przygotowanie modeli o największej zdolności predykcji dla sztucznie wygenerowanego zbioru danych. Składał się on z dwóch podzbiorów o 500 numerycznych kolumnach - jeden zawierał 2000 wierszy wraz z przyporządkowaniem elementów do klas a drugi z 600 wierszy, ale bez odpowiednich przyporządkowań. Zdolność predykcji optymalizowano pod metrykę *balanced accuracy*. Podczas pracy należało przeanalizować dwie ścieżki

1. Zbudowanie własnego potoku uczenia wybranego algorytmu
2. Wykorzystanie rozwiązań automatycznego uczenia maszynowego.

Otrzymany zbiór podzielono, otrzymując zbiór treningowy oparty na 70% próbek i zbiór testowy składający się z 30% rekordów. Dodatkowo rozwiązanie było sprawdzane na próbce 5% danych docelowych, które ostatecznie miały sprawdzić poprawność modelu. Był to niewielki zbiór stanowiący 30 rekordów, więc stanowił mniejszą wartość odniesienia od zbioru testowego. Wszystkie eksperymenty przeprowadzono na ziarnie generatora równym 0.

2 Metody AutoML

W ramach tej ścieżki przetestowano trzy pakiety: AutoGluon, mljar-supervised oraz TabPFN.

Dla pakietu mljar-supervised przeprowadzono szybki trening na ograniczonym czasie 1 min w trybie *compete*. Jego wynik na zbiorze testowym był równy 0.8103. Po podwyższeniu limitu czasu do ośmiu godzin wyniki wewnętrznej walidacji krzyżowej były obiecujące, jednakże przez rzucane błędy biblioteka nie była w stanie podać predykcji dla zbioru testowego, co doprowadziło do zaniechania dalszych prób wykorzystania tego pakietu.

Dla biblioteki AutoGluon dokonano analogicznego testu, uzyskując po minucie wynik na poziomie 0.8336. Kolejna próba przeprowadzona na ograniczeniu czasu wykonania na osiem godzin i ustawionym trybie *quality* dała dokładność 0.8519, a na próbce danych docelowych 0.9667. Zdecydowano się również puścić ten sam algorytm dla zawężonej liczby kolumn wybranych na podstawie feature selection opartym na ExtraTrees. Więcej na

temat tej metody wyboru cech opisano w następnym rozdziale. Dzięki zastosowanej selekcji uzyskano wynik lokalny 0.9018, a na próbce danych docelowych 0.9333.

W przypadku TabPFN nie ma konieczności wprowadzenia ograniczenia czasowego z racji na jego szybkość wykonania. Z pełnym zestawem kolumn jego wynik predykcji dla zbioru testowego wyniósł 0.5434. Jako sieć neuronowa został on wytrenowany na zbiorach danych, które posiadają nie więcej niż 100 kolumn. Stąd wynika niedokładność jego predykcji dla zbioru danych otrzymanego w ramach tego zadania. Aby sprawdzić jego skuteczność dla mniejszej liczby kolumn przeprowadzono prosty algorytm wyboru kolumn (eng. *feature selection*) opierający się na zachłannym wyborze tych kolumn, które prowadzą do polepszenia wyniku. Ostatecznie algorytm wybrał zestaw 5 kolumn, który dla zbioru testowego dawał wynik predykcji na poziomie 0.8882. W przypadku próbki danych docelowych otrzymano dokładność 0.9667.

Mimo otrzymania wyższych wyników przez TabPFN z wybranym zestawem kolumn, ostatecznie zdecydowano się na wykorzystanie modelu wytrenowanego przez pakiet AutoGluon z racji na jego wewnętrzne mechanizmy zapobiegania przeuczenia oraz ze względu na wątpliwą możliwość prawidłowej predykcji przy wykorzystaniu zaledwie 1% kolumn zbioru danych. Ponadto wybrano wariant wytrenowany na okrojonym zbiorze cech z racji na wyższą zdolność predykcji na zbiorze testowym, który jest znacznie większy od próbki zbioru docelowego.

3 Własny potok uczenia

Na etapie wczesnych badań zauważono, że bazowe algorytmy uczenia maszynowego nie radzą sobie najlepiej dla zadanego zbioru danych. Przykładowo GradientBoosting osiągnął wynik na zbiorze testowym 0.7004, RandomForest 0.6776, SVM 0.6078, KNN 0.6832, drzewa decyzyjne 0.7328, MLP 0.5305. Optymalizacja hiperparametrów metodą RandomSearch również nie przyniosła zadowalających rezultatów, osiągając przykładowo dla RandomForest 0.7006.

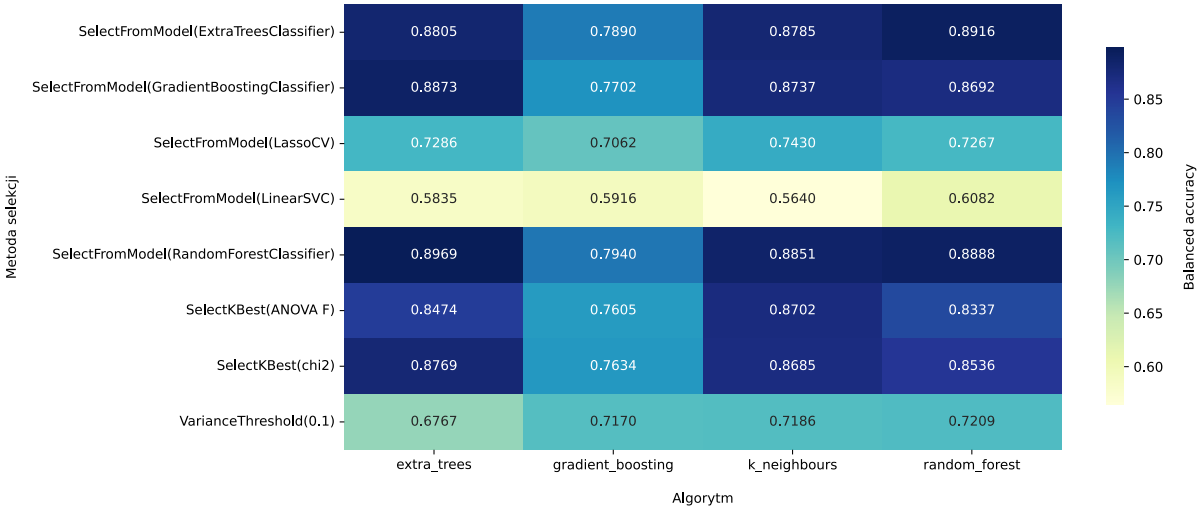
Z tego powodu zdecydowano się na ograniczenie zbioru rozważanych kolumn poprzez mechanizmy *feature selection*, aby odrzucić te dane, które utrudniają prawidłową predykcję. W ten sposób wybrano 8 metod wyboru kolumn, których wyniki przedstawiono w formie mapy ciepła 1. Jeżeli metoda umożliwiała podania liczby cech, to podano dla niej wartość 20, ponieważ we wcześniejszych testach była to średnio najbardziej optymalna wartość.

Warto odnotować, że najwyższy średni wynik wynoszący 0.8002 odnotował algorytm KNeighbors. Niewiele mniejsze średnie wyniki uzyskały RandomForest 0.7991 i ExtraTrees 0.7972.

Jednakże, najwyższy wynik uzyskał algorytm ExtraTrees dla metody selekcji opartej na RandomForest. Dodatkowo wykreślono zależności balanced accuracy od liczby wybranych cech, aby zoptymalizować liczbę wybranych cech dla dwóch najlepszych metod selekcji, czyli opartych na ExtraTrees i RandomForest. Analogiczne dane zebrano dla pozostałych algorytmów, jednak dla żadnej wartości parametru liczby wybranych cech nie uzyskano lepszego wyniku. Uzyskane w ten sposób wykresy zamieszczono w dodatku. Warto zaznaczyć, że wyniki metody selekcji opartej o ExtraTrees maleją wolniej ze wzrostem liczby kolumn niż wyniki selekcji opartej na RandomForest. Mimo, że najwyższy wynik 0.9034 na danych testowych uzyskał algorytm ExtraTrees ze zbiorem cech otrzymanych z metody selekcji opartej na RandomForest, to na próbce danych testowych uzyskał najgorszy wynik z dotąd badanych rozwiązań wynoszący 0.8667. Próby optymalizacji hiperparametrów, zarówno przy pomocy Random Search jak i Bayes Search, nie pomogły polepszyć otrzymanego wyniku w żadnym z badanych przypadków. Z tego powodu ostateczny potok nauczania składa się z zaledwie dwóch kroków: wyboru 18 najlepszych kolumn za pomocą metody selekcji opartej o ExtraTrees i samego algorytmu ExtraTrees. Na zbiorze testowym osiągnął on wynik 0.9019, a na próbce danych docelowych 0.9333.

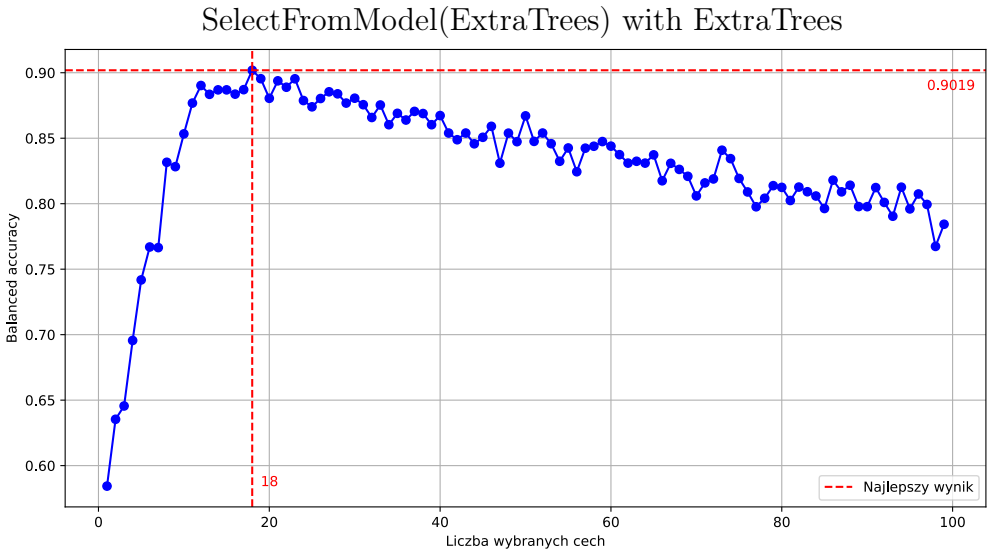
4 Wnioski

Ostatecznie zdecydowano się na rozwiązanie oparte o framework automatycznego uczenia maszynowego AutoGluon z zawężoną liczbą kolumn, który osiągnął na zbiorze testowym wynik balanced accuracy na poziomie 0.9018, zaś na próbce danych docelowych 0.9333. Własny potok nauczania jest zbudowany na połączeniu wyboru 18 najlepszych kolumn z algorytmem ExtraTrees. Wykorzystanie tej kombinacji zagwarantowało wynik na zbiorze testowym rzędu 0.9019, a na próbce danych docelowych 0.9333. Choć pakiety automatycznego uczenia maszynowego potrafią znajdować potoki, które dokonują bardzo dobrych predykcji dla zadanych zbiorów danych, to wciąż istnieje pole do poprawy, co widać w różnicy wyników AutoGluona na zbiorze testowym przed i po dokonaniu selekcji cech.

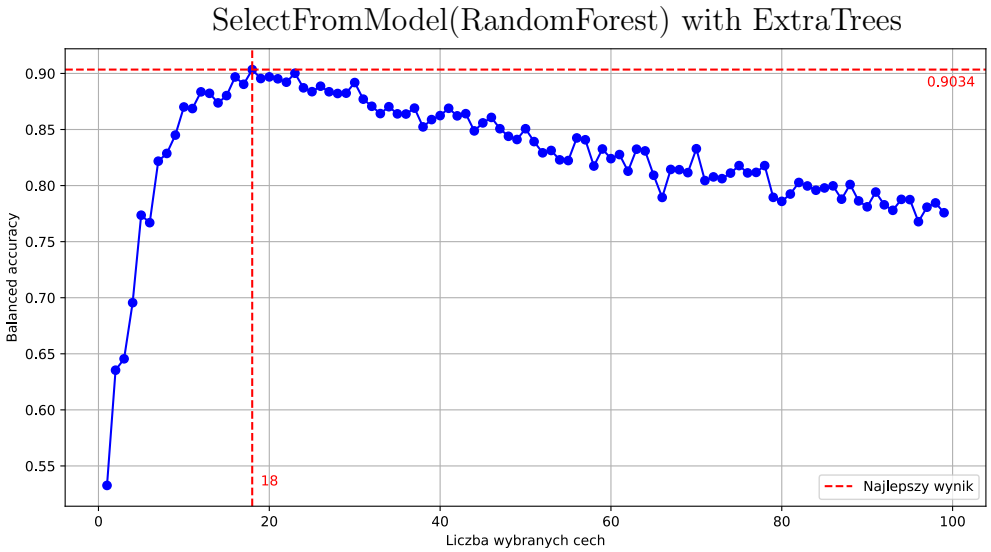


Rysunek 1: Mapa ciepła obrazująca wynik balanced accuracy w zależności od użytej metody wyboru cech oraz użytego algorytmu trenowania. W przypadku metody SelectFromModel wybierano zawsze 20 najlepszych kolumn. Liczba ta ma swoje poparcie w przeprowadzonych uprzednio testach ręcznych.

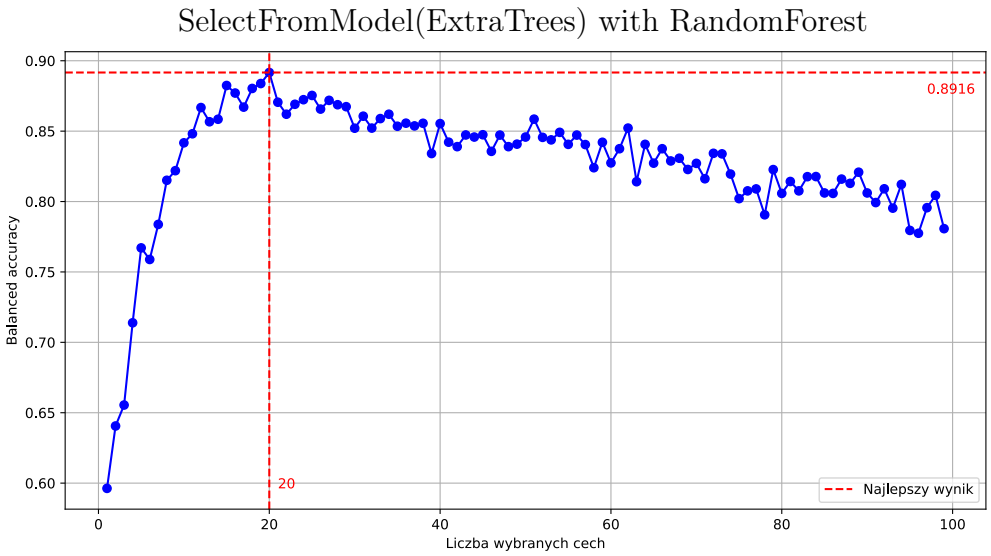
Dodatek



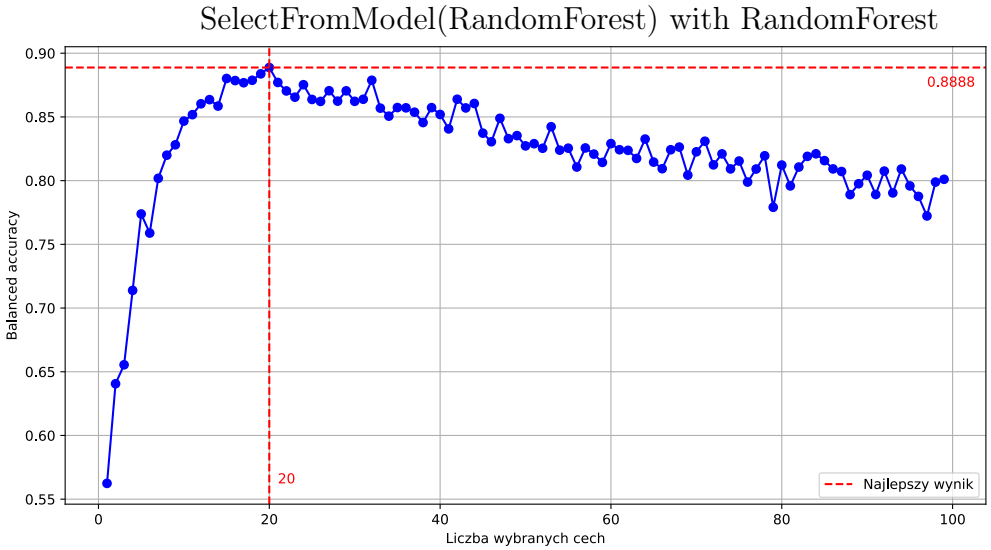
Rysunek 2: Wykres przedstawiający wynik balanced accuracy algorytmu ExtraTrees dla metody selekcji kolumn opartej na ExtraTrees w zależności od wybranej liczby cech zbioru treningowego.



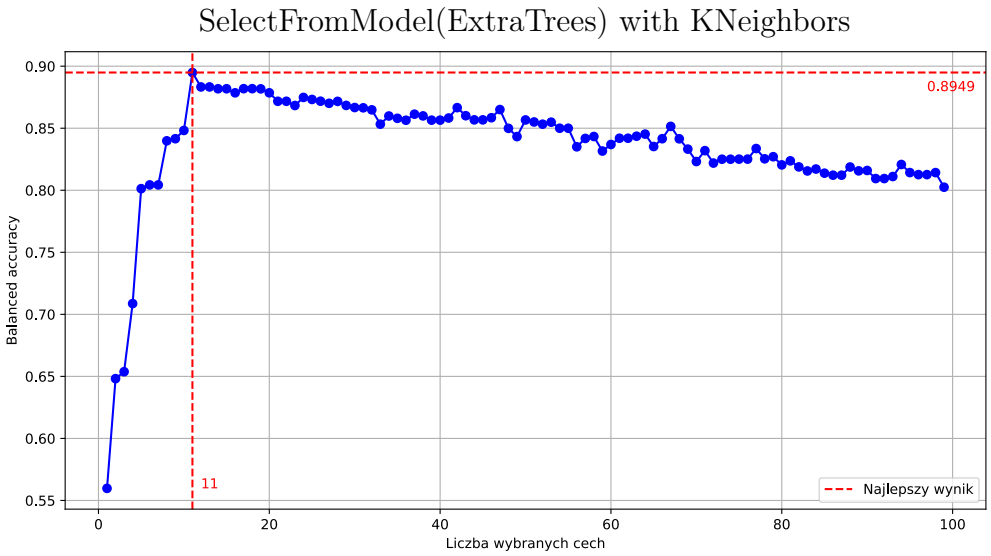
Rysunek 3: Wykres przedstawiający wynik balanced accuracy algorytmu ExtraTrees dla metody selekcji kolumn opartej na RandomForest w zależności od wybranej liczby cech zbioru treningowego.



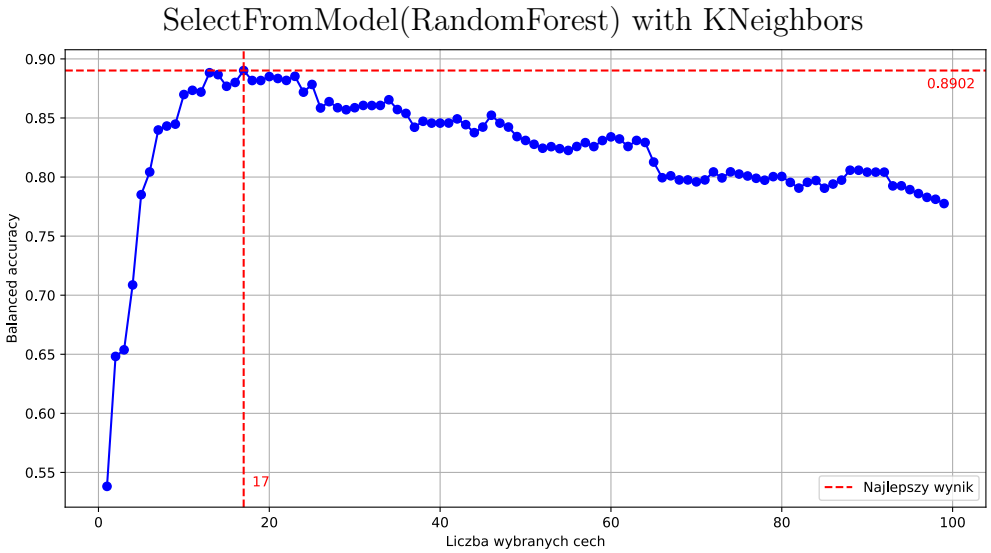
Rysunek 4: Wykres przedstawiający wynik balanced accuracy algorytmu RandomForest dla metody selekcji kolumn opartej na ExtraTrees w zależności od wybranej liczby cech zbioru treningowego.



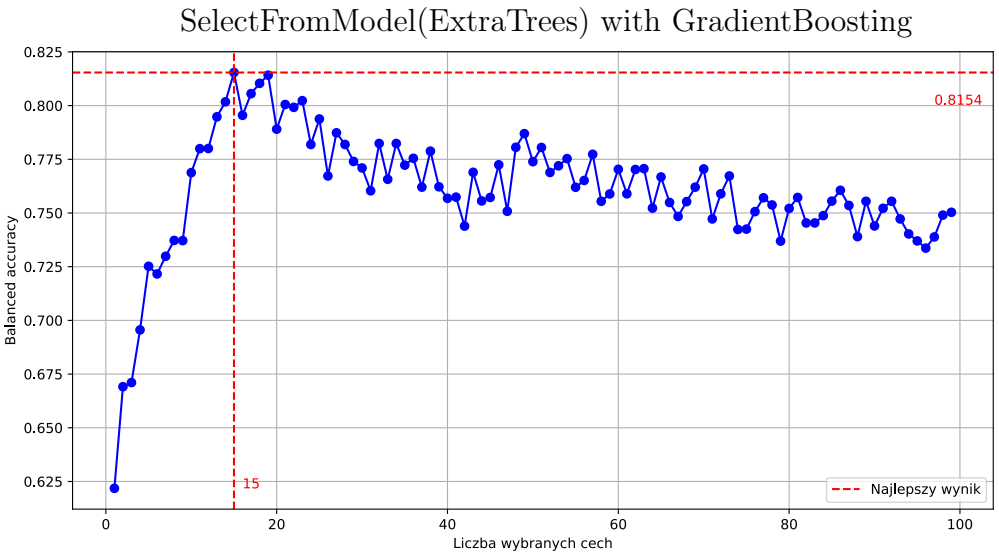
Rysunek 5: Wykres przedstawiający wynik balanced accuracy algorytmu RandomForest dla metody selekcji kolumn opartej na RandomForest w zależności od wybranej liczby cech zbioru treningowego.



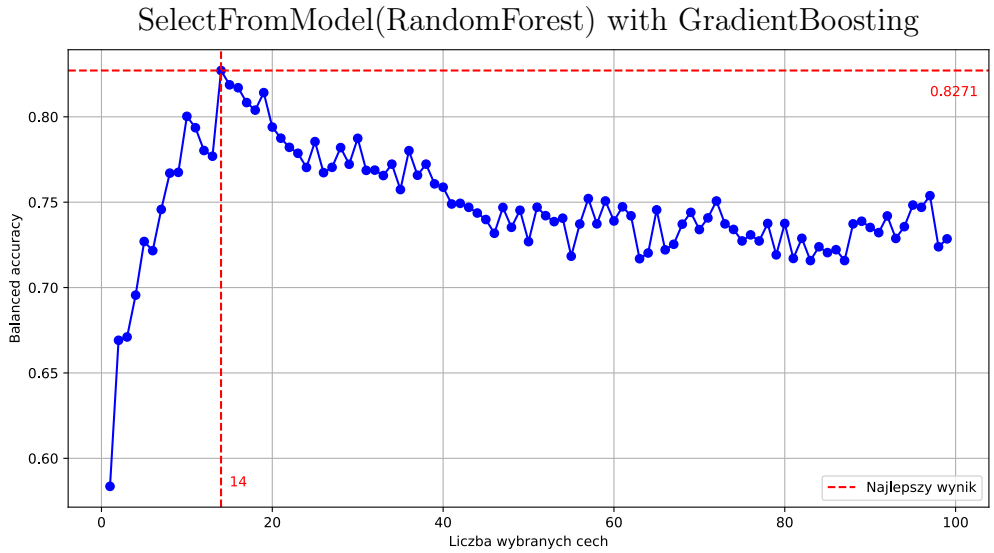
Rysunek 6: Wykres przedstawiający wynik balanced accuracy algorytmu KNeighbors dla metody selekcji kolumn opartej na ExtraTrees w zależności od wybranej liczby cech zbioru treningowego.



Rysunek 7: Wykres przedstawiający wynik balanced accuracy algorytmu KNeighbors dla metody selekcji kolumn opartej na RandomForest w zależności od wybranej liczby cech zbioru treningowego.



Rysunek 8: Wykres przedstawiający wynik balanced accuracy algorytmu GradientBoosting dla metody selekcji kolumn opartej na ExtraTrees w zależności od wybranej liczby cech zbioru treningowego.



Rysunek 9: Wykres przedstawiający wynik balanced accuracy algorytmu GradientBoosting dla metody selekcji kolumn opartej na RandomForest w zależności od wybranej liczby cech zbioru treningowego.