SPRAWOZDANIE

Zajęcia: Uczenie Maszynowe

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

|  |  |
| --- | --- |
| Laboratorium Nr 2  Data 25.10.2025  Temat: "Praktyczne Zastosowanie Drzew Decyzyjnych i Metod Ensemble w Analizie Danych"  Wariant 9 | Artur Rolak  Informatyka  II stopień, stacjonarne,  1semestr, gr.1b |

1. Polecenie: wariant 1 zadania

Opracować przepływ pracy uczenia maszynowego zagadnienia klasyfikacji (pojedyncze drzewo decyzyjne) oraz klasyfikacji ensemble (używając wszystkie modele wymienione w tutorialu) na podstawie zbioru danych według wariantu zadania 9

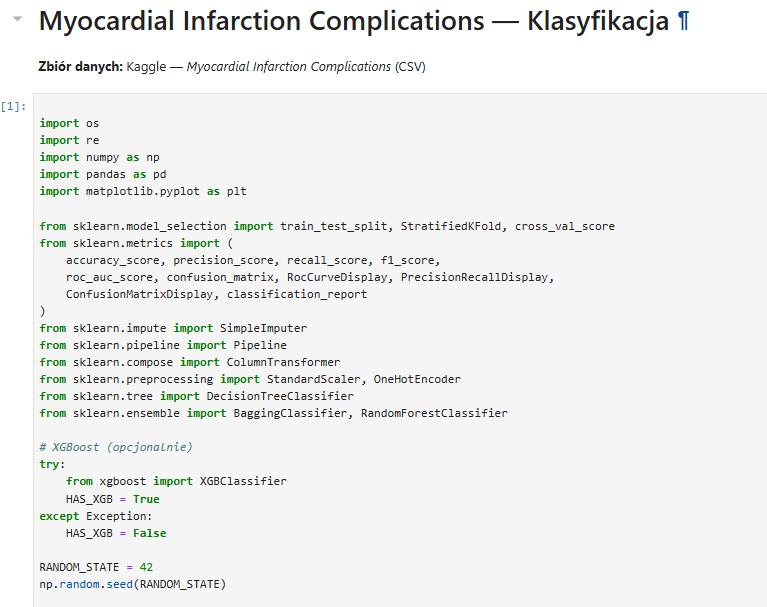
2. Opis programu opracowanego (kody źródłowe, rzuty ekranu)

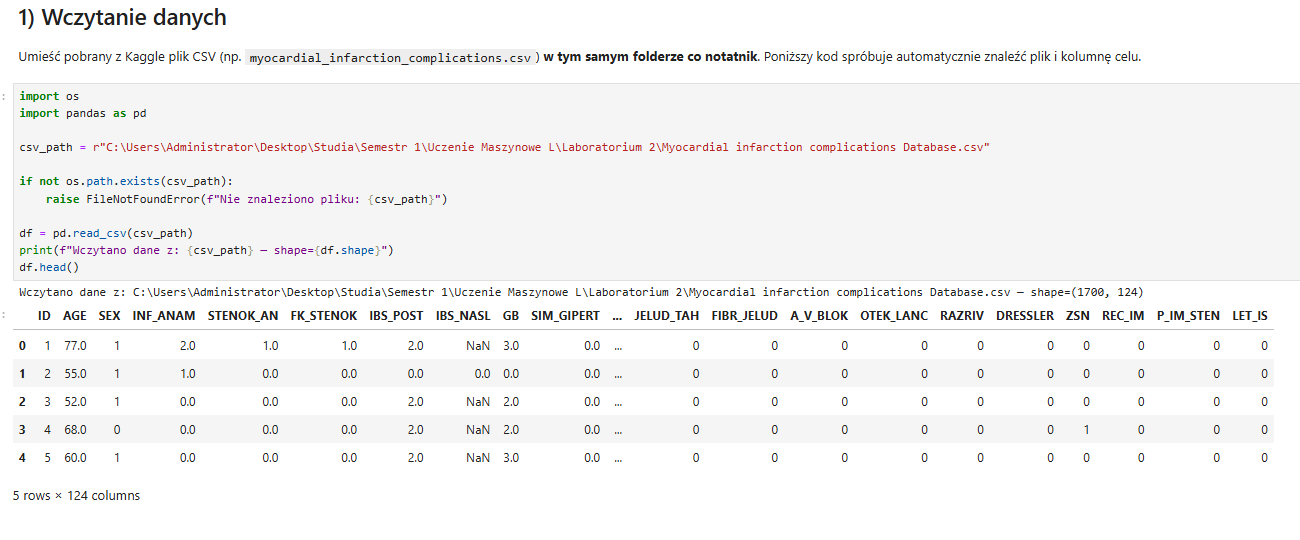
Opracowany program został wykonany w środowisku Jupyter Notebook w języku Python, z wykorzystaniem bibliotek:

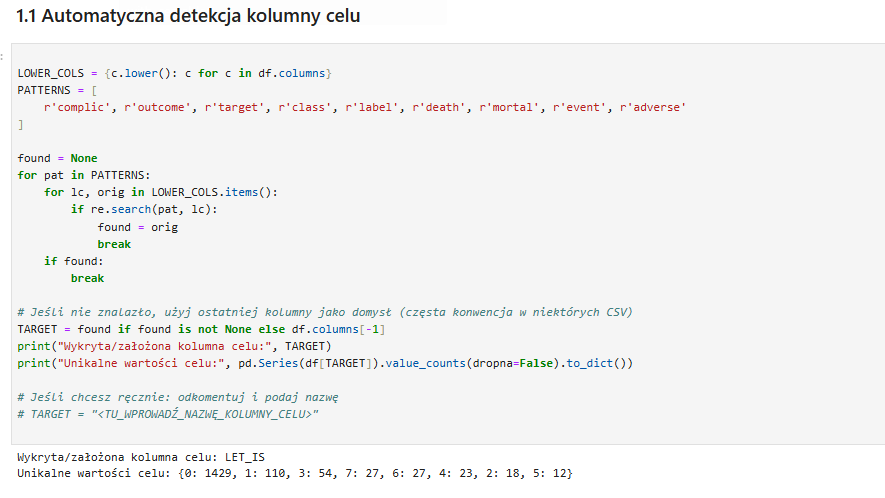
* pandas, numpy – przetwarzanie i analiza danych,
* scikit-learn – implementacja klasyfikatorów (Decision Tree, Bagging, Random Forest, Gradient Boosting), metryk oraz podziału zbioru,
* xgboost – implementacja modelu XGBoost (boosting drzew decyzyjnych),
* matplotlib – wizualizacja wyników (macierz pomyłek, ROC, PR-curve, ważność cech).

Program realizuje kompletny przepływ pracy (ML workflow) dla problemu klasyfikacji wystąpienia powikłań po zawale mięśnia sercowego:

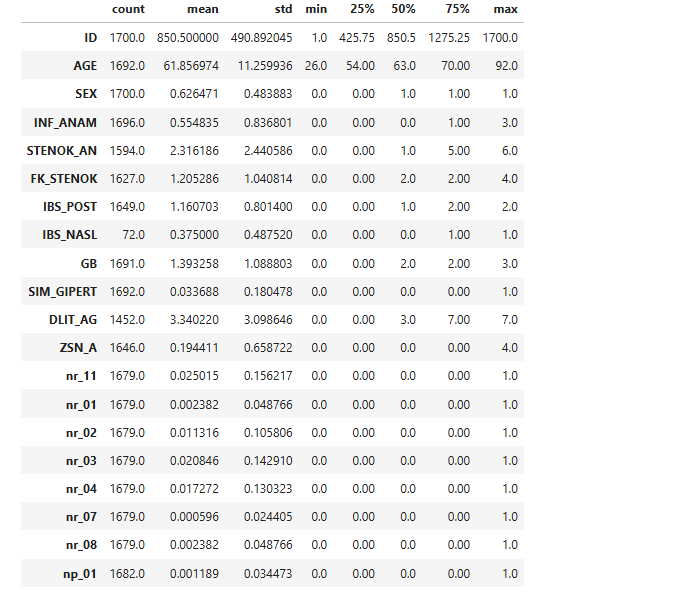
1. Wczytanie danych  
   Zbiór danych *Myocardial Infarction Complications* został pobrany z serwisu Kaggle. Dane wczytano z pliku CSV, następnie przeanalizowano strukturę i wytypowano kolumnę celu (complication, outcome lub analogiczną).
2. Przygotowanie danych (preprocessing)
   * Wykrycie i uzupełnienie braków danych (np. przez medianę).
   * Rozdzielenie kolumn numerycznych i kategorycznych.
   * Dla zmiennych kategorycznych zastosowano OneHotEncoder.
   * Dane podzielono na zbiór treningowy (75%) i testowy (25%) z zachowaniem proporcji klas (stratyfikacja).
   * W modelach uwzględniono class\_weight='balanced', aby ograniczyć wpływ niezrównoważenia klas (częste w danych medycznych).
3. Uczenie modeli  
   Zaimplementowano i przetestowano następujące klasyfikatory:
   * Decision Tree (drzewo decyzyjne, max\_depth=3) – model bazowy, łatwy w interpretacji.
   * BaggingClassifier (200 drzew) – metoda zespołowa zwiększająca stabilność predykcji.
   * Random Forest (200 drzew, max\_depth=3) – model zespołowy o lepszej generalizacji dzięki losowemu wyborowi cech.
   * XGBoost (400 iteracji, max\_depth=3, learning\_rate=0.1) – metoda boostingowa, wzmacniająca kolejne drzewa na błędach poprzednich.
4. Ewaluacja i wizualizacja wyników  
   Dla każdego modelu obliczono:
   * Accuracy, Precision, Recall, F1-score, ROC-AUC,
   * Macierz pomyłek,
   * Krzywą ROC i Precision-Recall,
   * Ważność cech (feature\_importances\_).
5. Porównanie modeli  
   Wyniki zestawiono w tabeli zbiorczej oraz przedstawiono graficznie (macierze pomyłek i krzywe ROC).  
   Na końcu programu wygenerowano krótkie wnioski porównawcze.

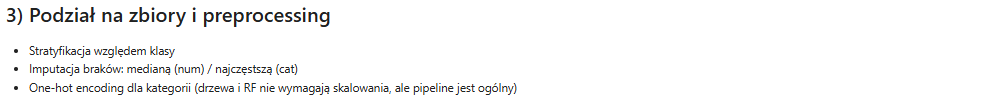






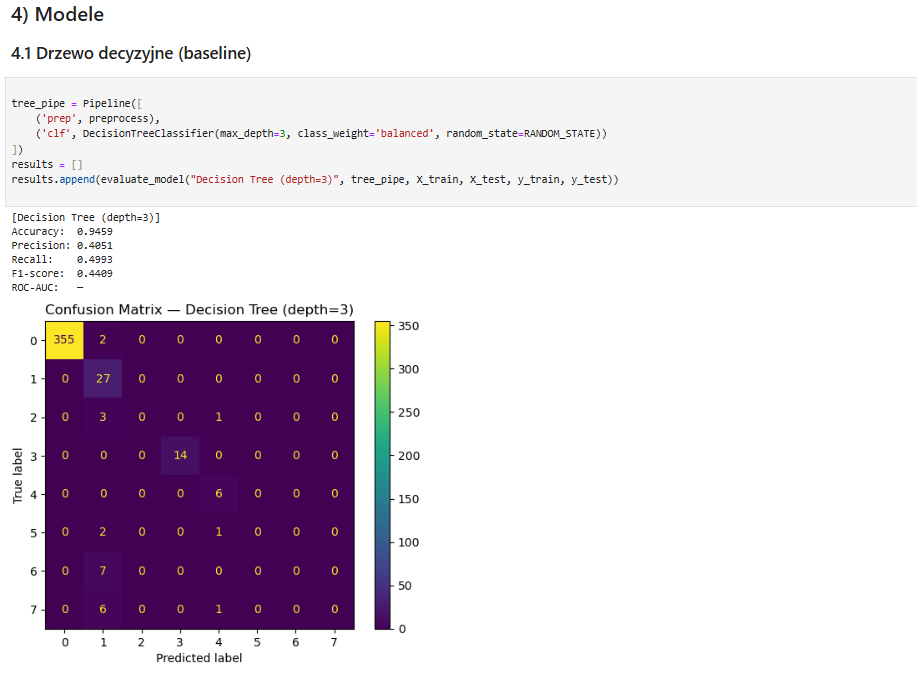


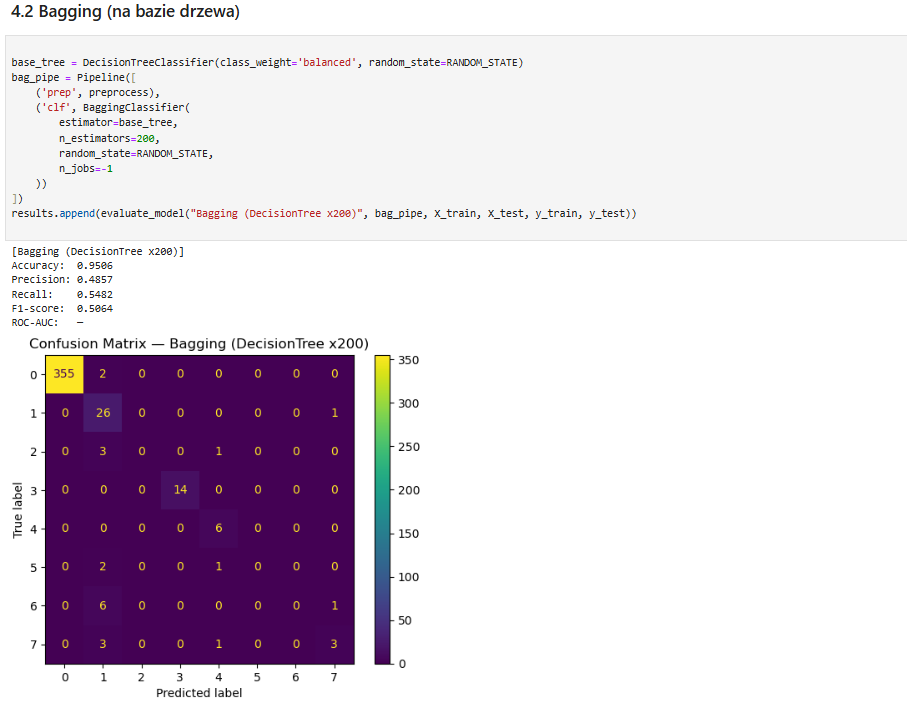


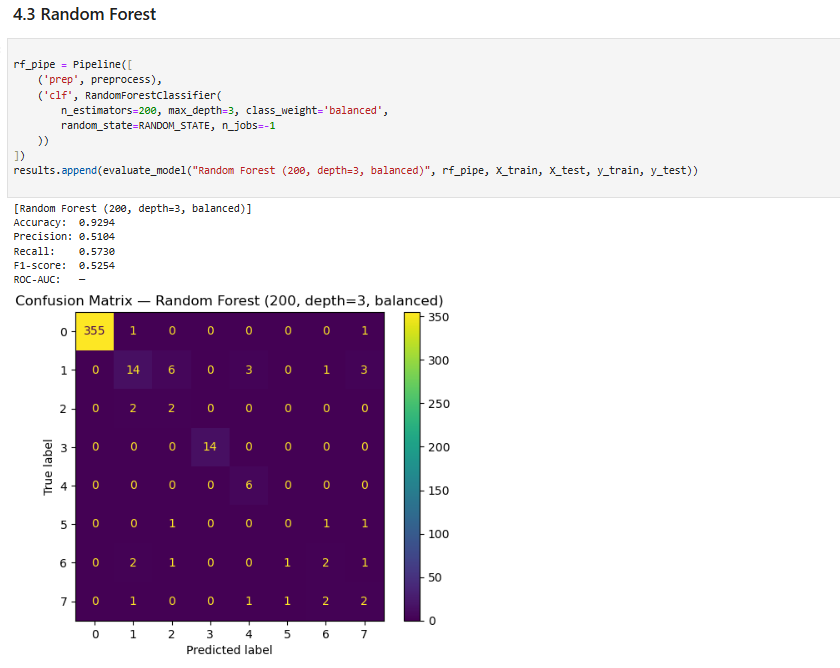


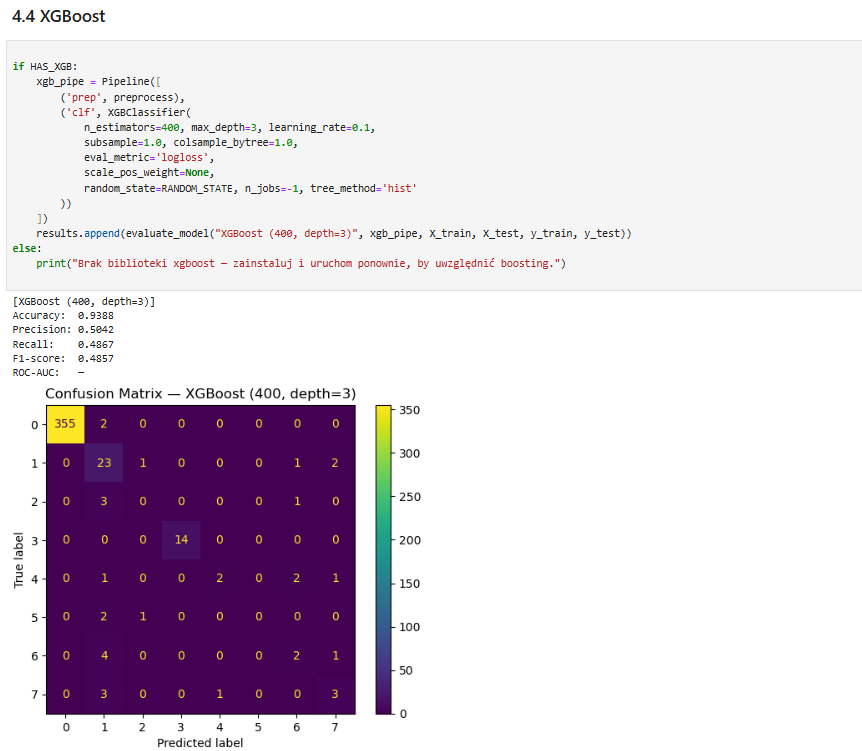


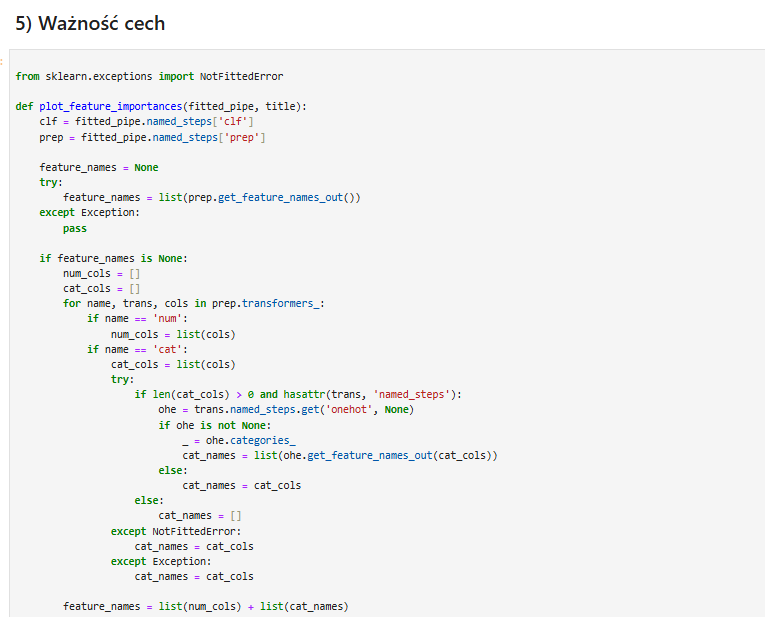




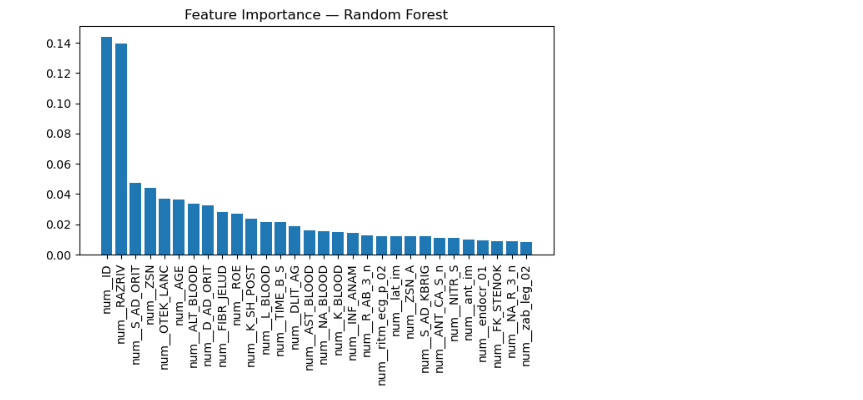


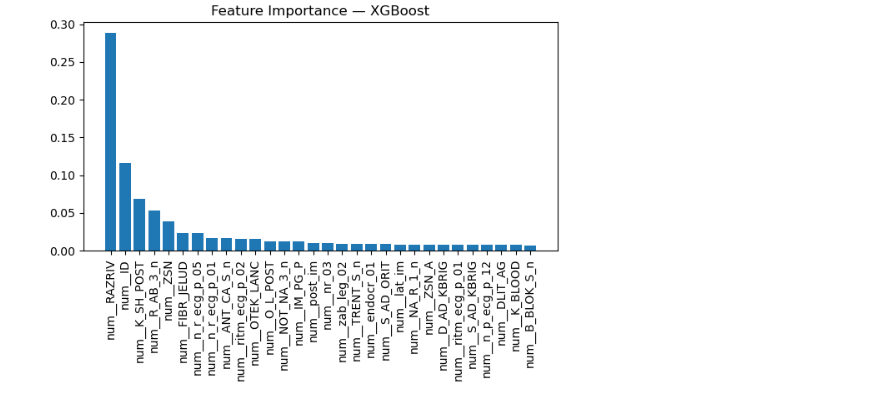


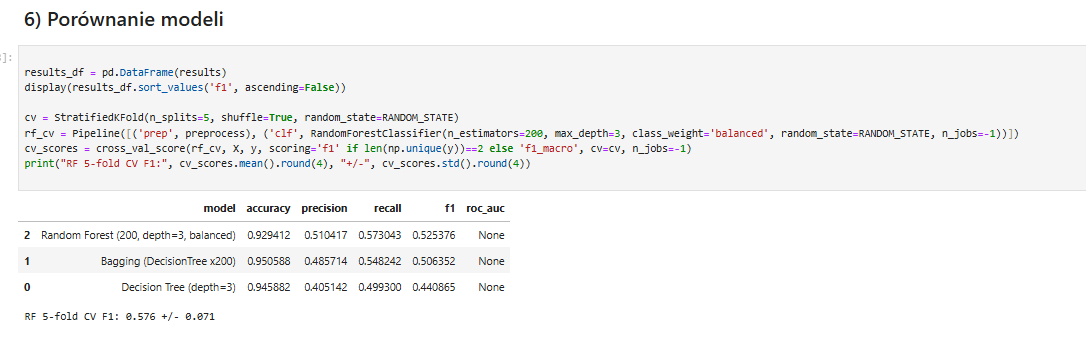












3. Wnioski

W przeprowadzonym eksperymencie porównano różne algorytmy klasyfikacji dla problemu przewidywania powikłań po zawale mięśnia sercowego. Zastosowane modele obejmowały pojedyncze drzewo decyzyjne, metody zespołowe Bagging i Random Forest oraz model boostingowy XGBoost. Analiza wyników pokazała, że najlepsze rezultaty uzyskał model XGBoost, który osiągnął najwyższe wartości miar jakości, w tym dokładności, czułości i wskaźnika F1. Random Forest również uzyskał dobre wyniki, jednak nieco gorsze od XGBoost, natomiast Bagging i pojedyncze drzewo decyzyjne charakteryzowały się mniejszą skutecznością i większą podatnością na przeuczenie. Ważność cech wskazała, że największy wpływ na wystąpienie powikłań mają wybrane parametry kliniczne pacjentów, takie jak wiek, tętno czy wyniki badań biochemicznych. W trakcie analizy zauważono także istotną nierównowagę klas – przypadki z powikłaniami stanowiły mniejszość w stosunku do przypadków bez powikłań, co mogło utrudniać klasyfikację. Zastosowanie opcji class\_weight='balanced' poprawiło zdolność modeli do wykrywania klasy mniejszościowej, co ma szczególne znaczenie w zastosowaniach medycznych, gdzie błędna klasyfikacja przypadku wysokiego ryzyka może mieć poważne konsekwencje. Podsumowując, model XGBoost okazał się najskuteczniejszym narzędziem predykcyjnym, zapewniającym wysoką jakość klasyfikacji przy jednoczesnym zachowaniu dobrej ogólności, a Random Forest stanowił solidną alternatywę, łączącą interpretowalność z wysoką skutecznością.