Wstęp do uczenia maszynowego: Projekt I
Zbiór danych "sick"

Paweł Koźmiński, Paulina Przybyłek, Ada Gąssowska

21.04.2020

Dane "sick"

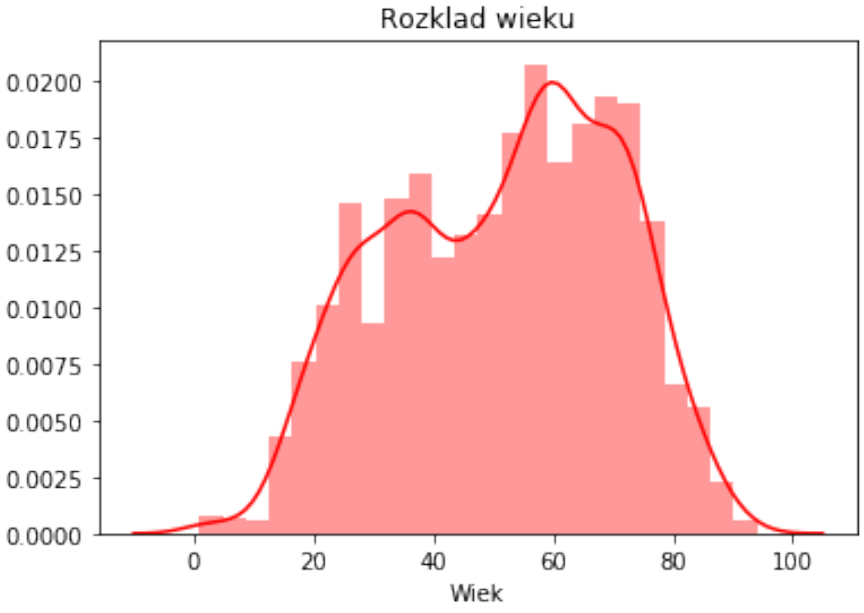
- Dane "Sick" to dane o 3772 pacjentach badanych na obecność choroby tarczycy.
- Ramka zawiera 30 kolumn, np. wiek, płeć, szpital w jakim przebywa pacjent, informacje medyczne (czy pacjent jest w ciąży, czy bierze leki, itp), poziomy wybranych hormonów oraz informację czy ma chorobę tarczycy czy jest zdrowy.
- Naszą zmienną celu jest właśnie kolumna określająca obecność choroby tarczycy ("Thyroid Disease").
- Dane są niezbalansowane - **chorobę tarczycy stwierdzono jedynie u około 6 procent pacjentów**

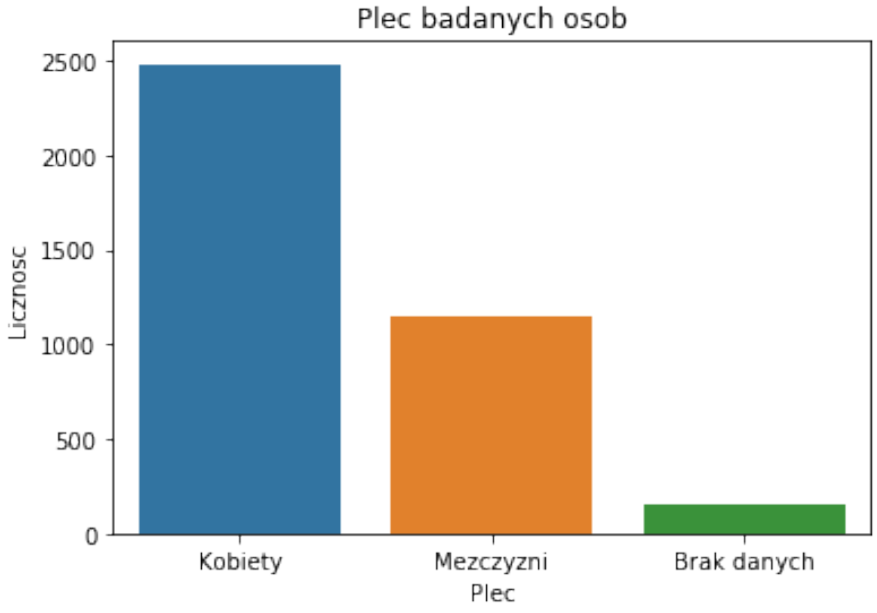
Dane numeryczne - wiek i hormony

	age	TSH	T3	TT4	T4U	FTI	TBG
count	3771.000000	3403.000000	3003.000000	3541.000000	3385.000000	3387.000000	0.0
mean	51.735879	5.086766	2.013500	108.319345	0.995000	110.469649	NaN
std	20.084958	24.521470	0.827434	35.604248	0.195457	33.089698	NaN
min	1.000000	0.005000	0.050000	2.000000	0.250000	2.000000	NaN
25%	36.000000	0.500000	1.600000	88.000000	0.880000	93.000000	NaN
50%	54.000000	1.400000	2.000000	103.000000	0.980000	107.000000	NaN
75%	67.000000	2.700000	2.400000	124.000000	1.080000	124.000000	NaN
max	455.000000	530.000000	10.600000	430.000000	2.320000	395.000000	NaN

- Wiek ma jeden dziwny rekord - 455 (na kolejnym slajdzie pominięty), TBG to same braki danych. Reszta wartości wiarygodna.

Rozkłady wieku i płci



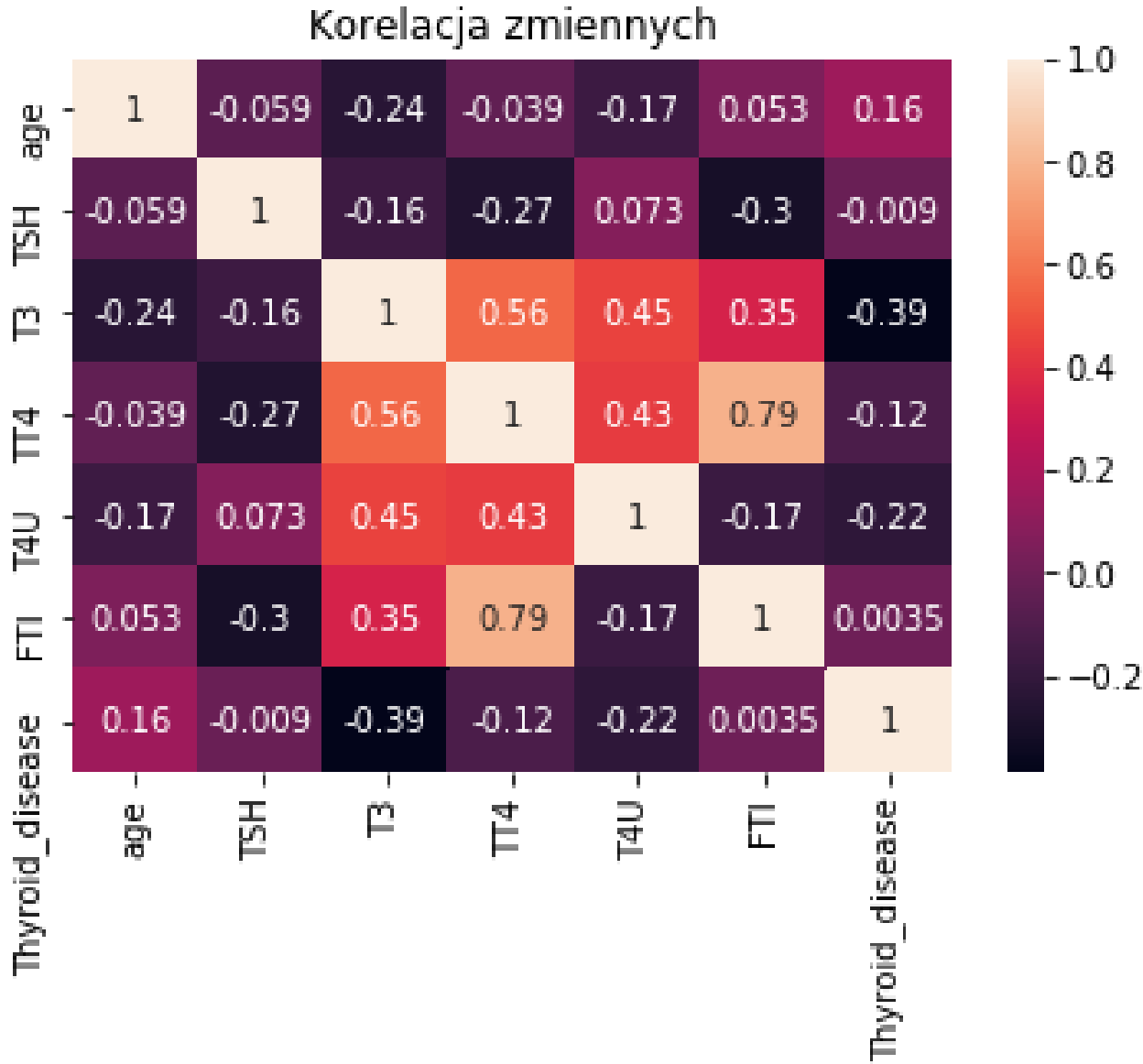


Procent braków danych w niepełnych kolumnach

age	0.03%
sex	3.98%
TSH	9.78%
T3	20.39%
TT4	6.12%
T4U	10.26%
FTI	10.21%
TBG	100%

W kolumnie TBG są same braki więc ją usuniemy. Na reszcie braków danych zastosujemy różne techniki imputacji.

Korelacje zmiennych numerycznych



Jedna znacząca korelacja - między FTI i TT4.

Inżynieria cech

- Zamiana boolean'ów na 1 i 0
- Usunięcie kolumny TBG
- Podział na zbiór treningowy (80%) i testowy (20%)
- Encoding zmiennych kategorycznych
- Imputacja danych

Encoding zmiennych kategorycznych

- **Kolumna sex** - na tej kolumnie wypróbowaliśmy:
  -zamianę na 0,1 i NaN;
  - Target Encoding i Oridinal Encoding traktując Missing jako oddzielną kategorię
- **Kolumna refferal source** - na tej kolumnie wypróbowaliśmy:
  - Target Encoding
  - One Hot Encoding

Imputacja brakujących danych

- Zmienną **sex** w ramce gdzie jest ona zakodowana numerycznie zaimputowaliśmy losowo z wykorzystaniem informacji o prawdopodobieństwie wylosowania konkretnej płci
- Zmienne **age, TSH, T3, TT4, T4U, FTI** zaimputowaliśmy średnią, medianą, modą, Knn Imputer'em oraz Iterative Imputer'em. Przed imputacją w kolumnie age na dwa sposoby poradziliśmy sobie z dziwnym wiekiem 455 - zamieniliśmy go na NaN i na 45.

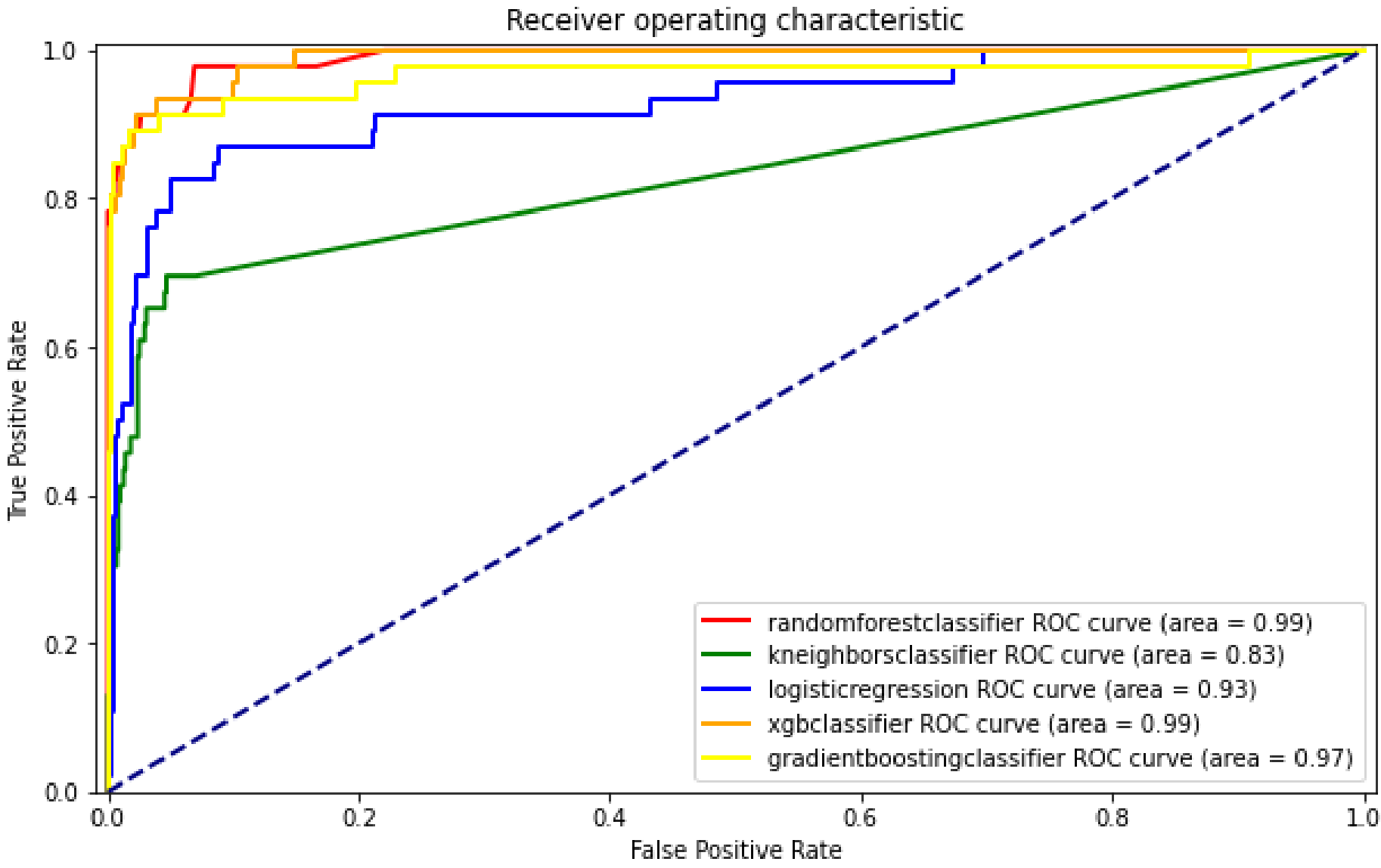
Pierwsze modele

- Na powstałych ramkach danych wypróbowaliśmy dwa wstępne modele - **XGBoost** oraz **Random Forest** (z domyślnymi parametrami).
- Na podstawie Score oraz macierzy konfuzji (najbardziej interesowało nas w nich przypadki False Negative - nie chcemy żeby chory pacjent zostal uznany za zdrowego), wybraliśmy ramkę danych do dalszego modelowania
- **Ostatecznie zmienna sex jest zakodowana na 0 i 1 a wartości są imputowane losowo, Refferal Source jest zakodowana Target encodingiem, wiek 455 zamieniony na NaN, a zmienne numeryczne zaimputowane są medianą.**

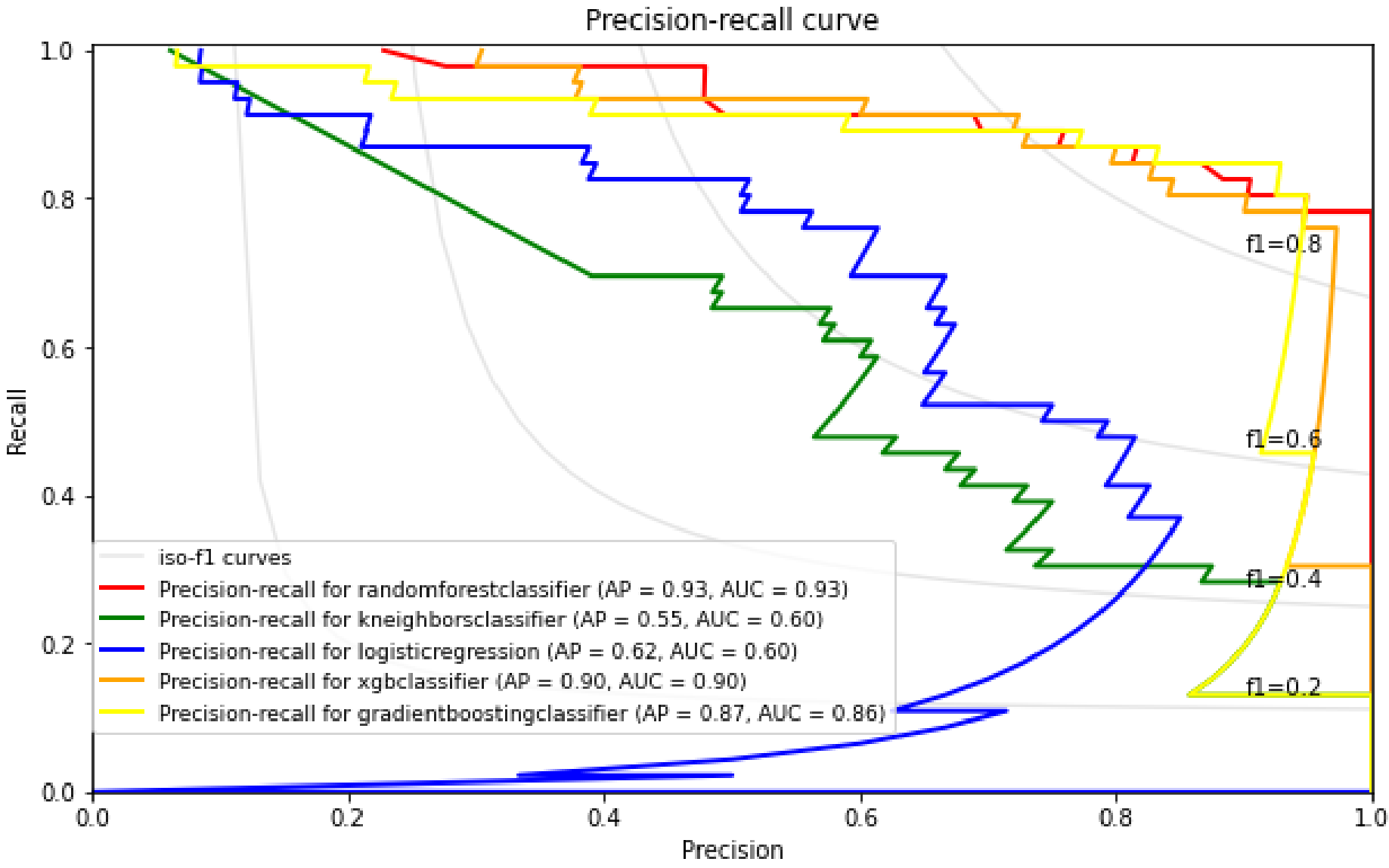
Szukanie optymalnych parametrów

- Stworzyliśmy funkcję **tuner**, która dla pięciu modeli (Regresja logistyczna, KNN, GradientBoosting, XGB oraz Random Forest) sprawdza czy standaryzacja dobrze wpływa na działanie modelu oraz dobiera najlepsze hiperparametry (sprawdzając zarówno grid search jak i random search).
- Sprawdziliśmy również jak na modele działa usunięcie jednej ze skorelowanych kolumn oraz usunięcie kolumn _measured, jednak w obu przypadkach modele radziły sobie gorzej, a czas działania skrócił się tylko nieznacznie

Krzywe ROC


Receiver operating characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
randomforestclassifier ROC curve (area = 0.99)
kneighborsclassifier ROC curve (area = 0.83)
logisticregression ROC curve (area = 0.93)
xgbclassifier ROC curve (area = 0.99)
gradientboostingclassifier ROC curve (area = 0.97)

Krzywe Precision-Recall



Najlepszy model

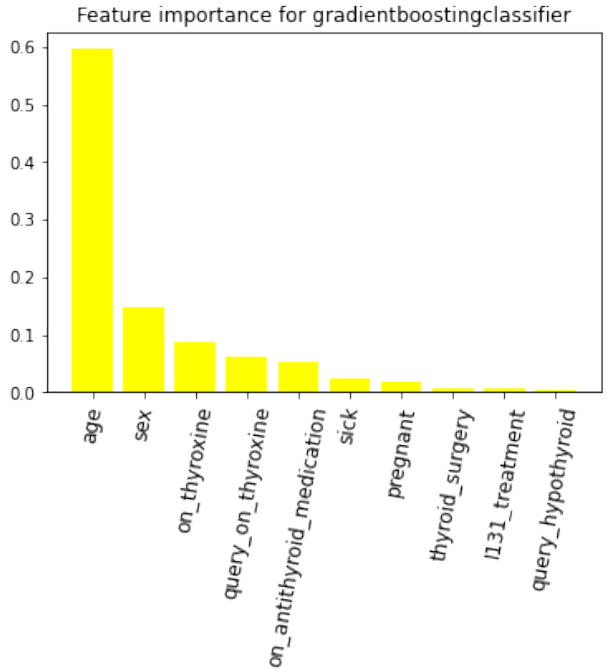
- Na podstawie miary krzywej ROC oraz krzywej Precission-Recall wynika, że najlepiej radzi sobie Random Forest z następującymi parametrami

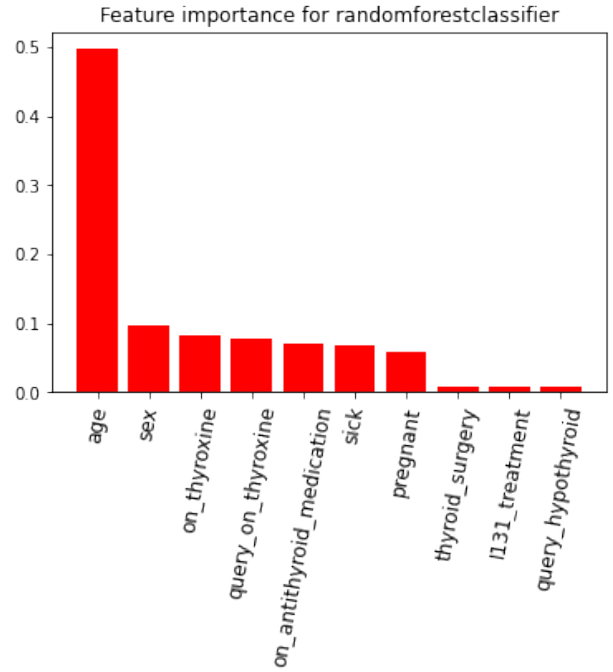
n_estimators:	879
max_features	'auto'
max_depth	14
criteration	'gini'

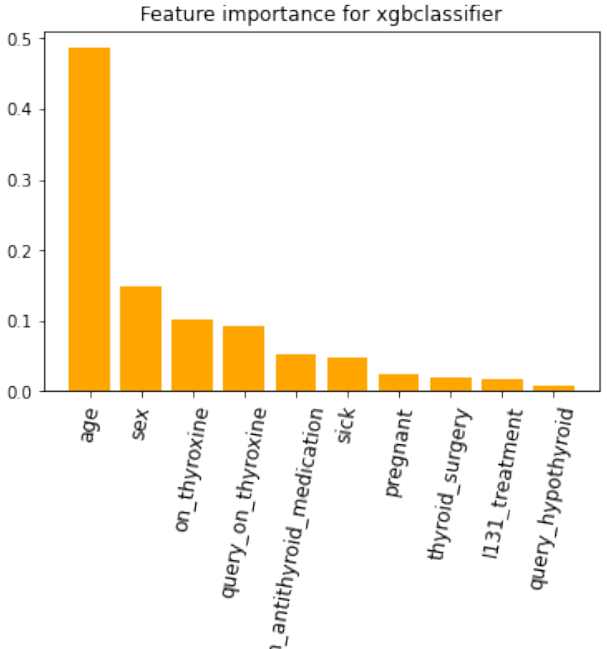
- Natomiast najlepszy Geometric Mean Score (0.919) uzyskał Gradient Boosting z następującymi parametrami

alpha:	0.00
learning_rate:	0.5
max_depth	3
min_samples_split	3
n_estimators	100

Feature Importance dla trzech najlepszych modeli



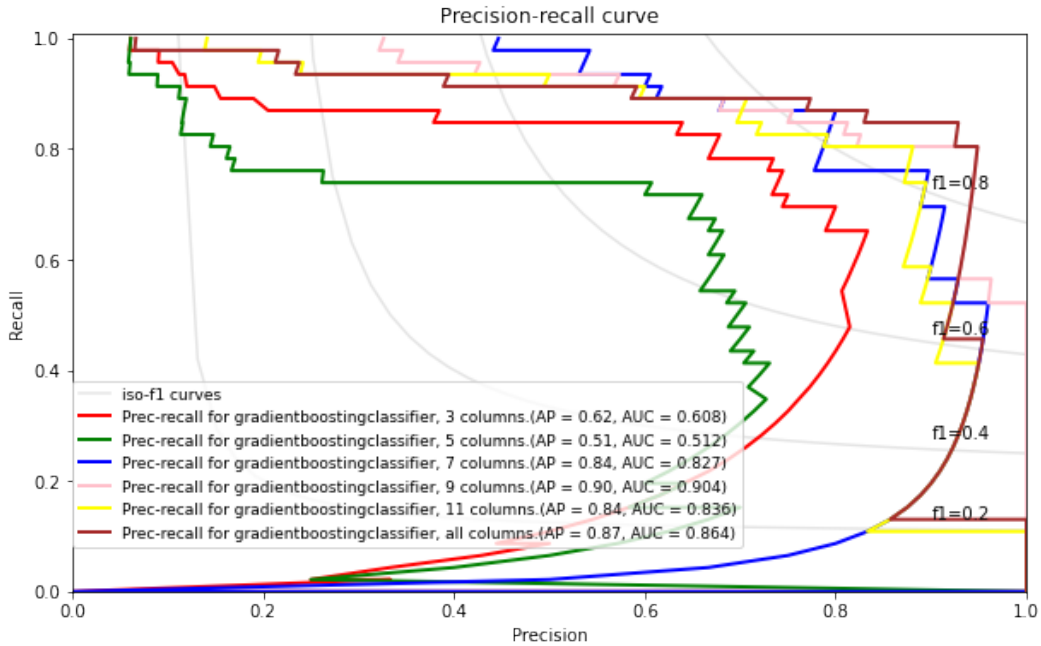


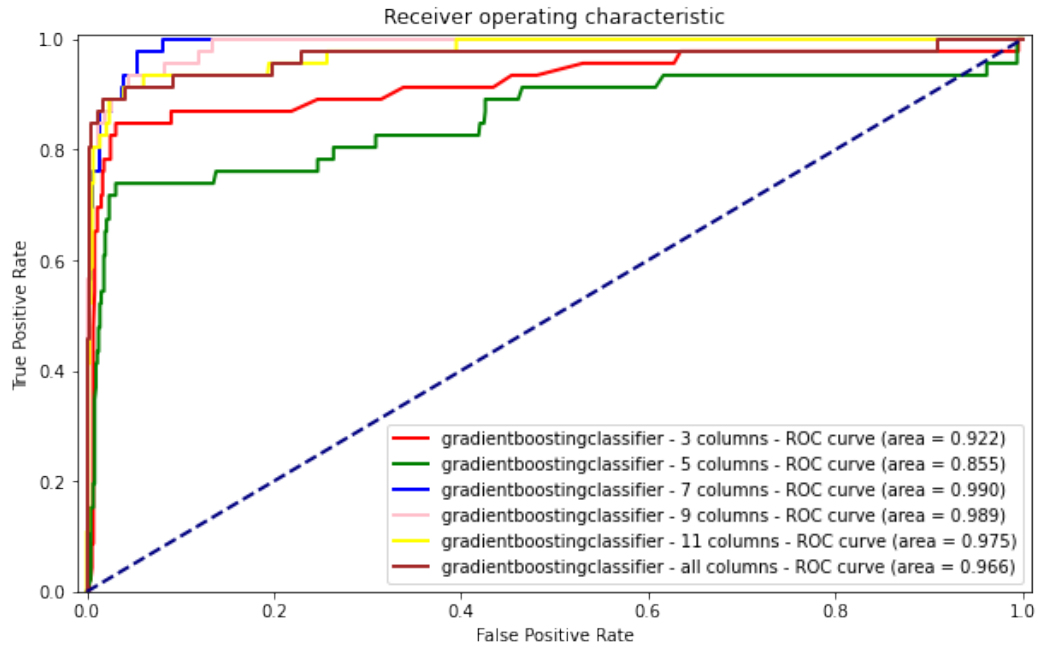


Modelowanie tylko na najbardziej znaczących kolumnach

- Dla trzech najlepszych modeli: GradientBoostingClassifier, RandomForestClassifier i XGBClassifier sprawdziliśmy jak poradzą sobie one po zostawieniu w ramce danych tylko częśći kolumn (tych najbardziej znaczących wg. feature importance).
- Sprawdziliśmy krzywe ROC i Precision-Recall oraz GM Score dla 3, 5, 7, 9, 11 i wszystkich kolumn.
- Dla RandomForest i XGB najlepsze wyniki miał model wykonany na pełnej ramce danych.
- Dla GradientBoosting'u na ramce danych z 7 kolumnami uzyskaliśmy najlepszy dotychczas uzyskany GM Score - 0.925.

Gradient Boosting - wykresy





AutoML - TPOT classifier

- Sprawdziliśmy równiez jak na naszej ramce poradzi sobie TPOT classifer z następującymi parametrami

generations:	5
verbosity:	2
n_jobs	-2

- Uzyskalismy tym samym GM Score 0.923 co jednak jest wynikiem nieznacznie niższym niż nasz Gradient Boosting Classifier uzyskał na 7 kolumnach.

Podsumowanie

- KnnClassifier i LogisticRegression uzyskały dosyć słabe wyniki na naszym zbiorze.
- XGBClassifier, RandomForestClassifier oraz Gradient Boosting Classifier uzyskały podobne - bardzo dobre wyniki.
- Najlepsze wyniki uzyskał **Gradient Boosting Classifier** po zostawieniu tylko 7 najbardziej znaczących według Feature Importance kolumn. Miał on wynik nieznacznie wyższy nawet od TPOT Classifier'a.