

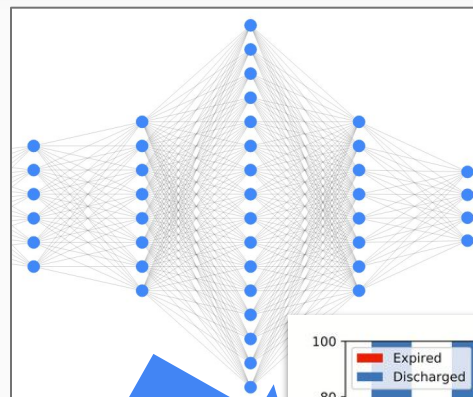
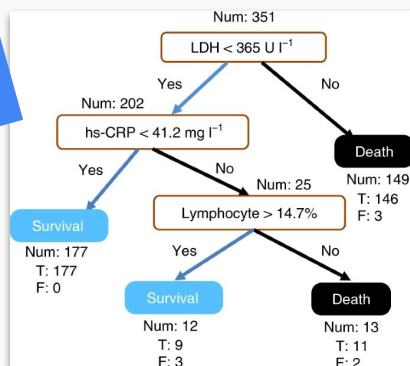
Sieci versus drzewa

Jakub Fołtyn, Kacper Grzymkowski, Konrad Komisarczyk

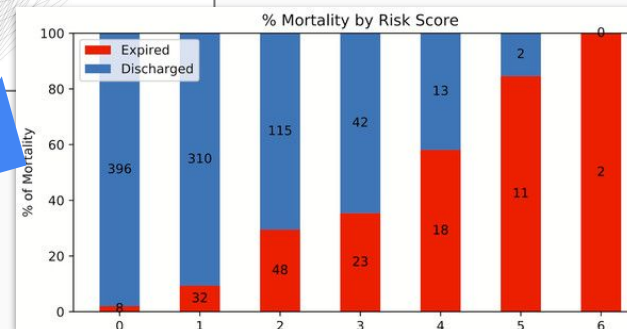
Artykuły

XGBoost

Yan et al. 2020



Li et al. 2020



Brak kodu

- Brak kodu dołączonego do artykułu
- Dokładny opis architektury

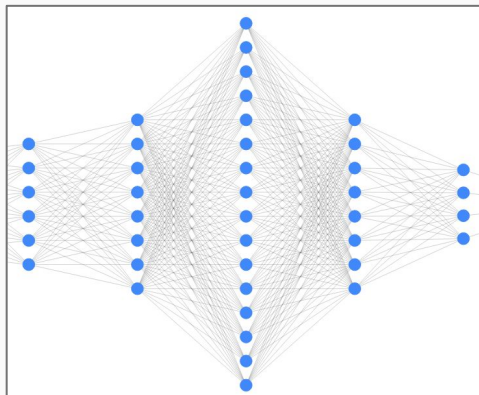


```
def make_article_layer(neurons, kernel_initializer="normal", kernel_regularizer="l2", **kwargs):
    return Dense(neurons, activation="relu", kernel_initializer=kernel_initializer, kernel_regularizer=kernel_regularizer, **kwargs)

def make_article_model():
    # The DNN model used five layers: an input layer and three hidden layers, respectively.
    model = Sequential()
    model.add(make_article_layer(6, kernel_initializer="normal", kernel_regularizer="l2"))
    model.add(make_article_layer(8, kernel_initializer="normal", kernel_regularizer="l2"))
    model.add(make_article_layer(16, kernel_initializer="normal", kernel_regularizer="l2"))
    model.add(make_article_layer(1, kernel_initializer="normal", kernel_regularizer="l2"))
    # [...] the sigmoid activation function and the "normal" normalization scheme were applied.
    model.add(Dense(1, activation="sigmoid", kernel_initializer="normal", kernel_regularizer="l2"))
    model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
    return model
```

Brak kodu

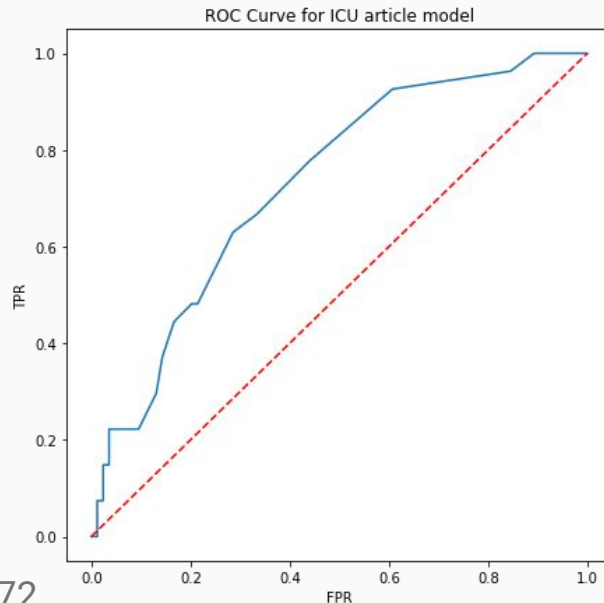
- Brak kodu dołączonego do artykułu
- Dokładny opis architektury



```
def make_article_layer(neurons, **kwargs):  
    return Dense(neurons, activation="tanh", kernel_initializer="normal", kernel_regularizer="l2", **kwargs)  
  
def make_article_model(layers):  
    # The DNN model used five layers of 4, 6, 10, 6, and 4 neurons respectively.  
    model = Sequential()  
    model.add(make_article_layer(4, **kwargs))  
    model.add(make_article_layer(6, **kwargs))  
    model.add(make_article_layer(10, **kwargs))  
    model.add(make_article_layer(6, **kwargs))  
    model.add(make_article_layer(4, **kwargs))  
    # [...] the sigmoid activation function and the "normal" normalization scheme were applied.  
    model.add(Dense(1, activation="sigmoid", kernel_initializer="normal", kernel_regularizer="l2"))  
    model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])  
    return model
```

Odtworzenie sieci

- Wyniki podobne do tych przedstawionych w artykule
- Gorsze, ale na akceptowalnym poziomie



Oryginalne AUC = 0.78

Odtworzone AUC = ~ 0.72

Problem ze stabilnością

Problem:

Nie działa, przynajmniej nie zawsze

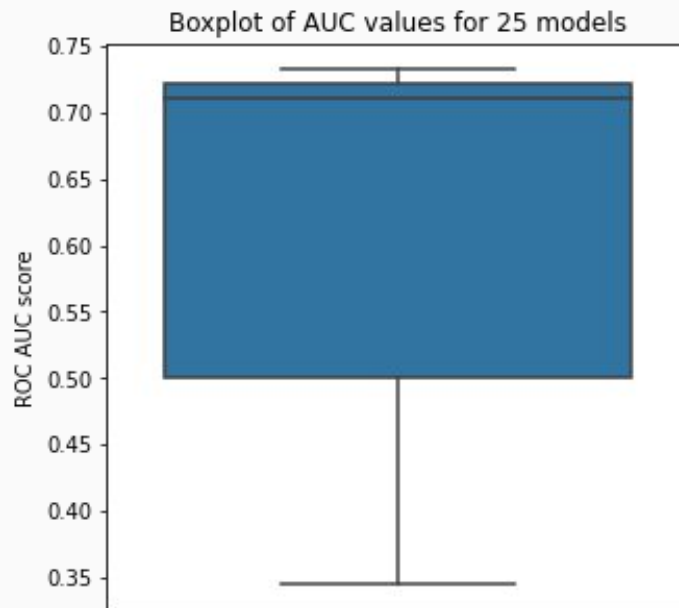
```
Training model #1  
ROC AUC: 0.7125220458553793  
Training model #2  
ROC AUC: 0.3734567901234568  
Training model #3  
ROC AUC: 0.7416225749559083  
Training model #4  
ROC AUC: 0.2702821869488536  
Training model #5  
ROC AUC: 0.3472222222222227
```

Problem ze stabilnością

Problem:

Nie działa, przynajmniej nie zawsze

```
Training model #1  
ROC AUC: 0.7125220458553793  
Training model #2  
ROC AUC: 0.3734567901234568  
Training model #3  
ROC AUC: 0.7416225749559083  
Training model #4  
ROC AUC: 0.2702821869488536  
Training model #5  
ROC AUC: 0.3472222222222227
```

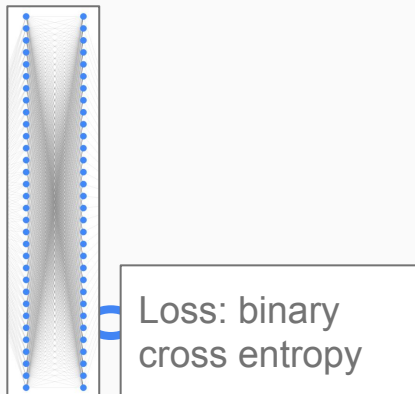


Modyfikacje

- Funkcja straty: Błąd średniokwadratowy - nietypowy wybór dla klasyfikacji

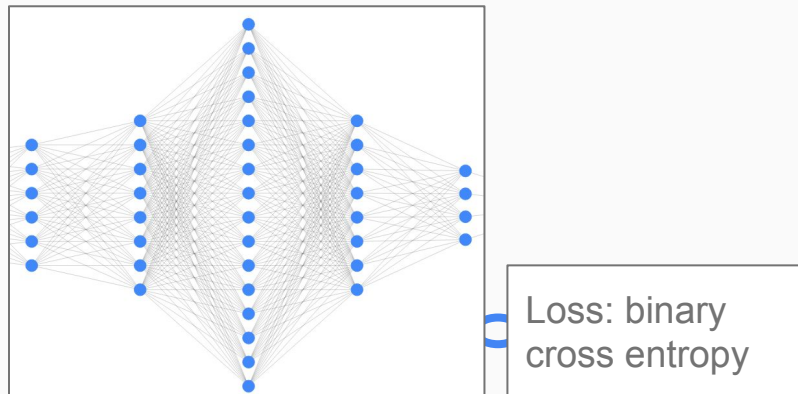
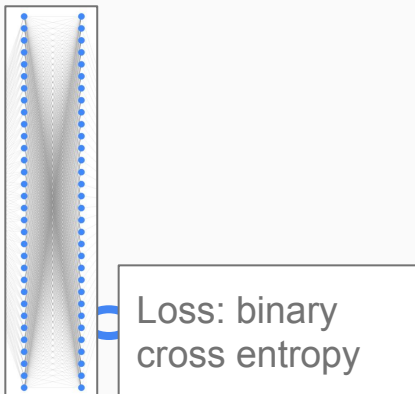
Modyfikacje

- Funkcja straty: Błąd średniokwadratowy - nietypowy wybór dla klasyfikacji
- Basic - 2 ukryte warstwy po 32 neurony



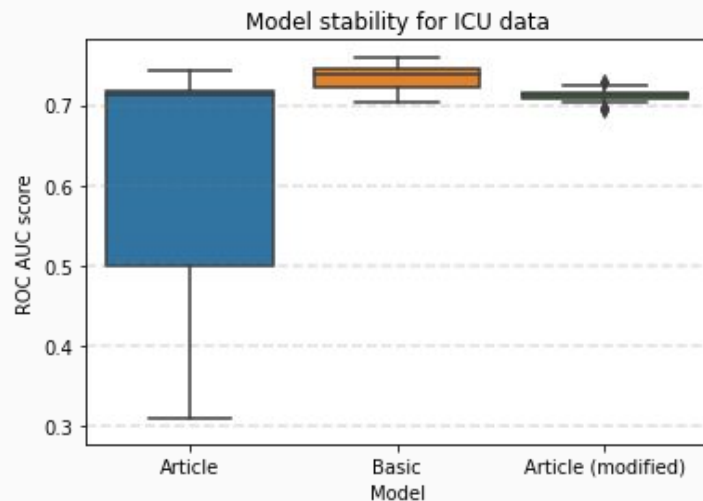
Modyfikacje

- Funkcja straty: Błąd średniokwadratowy - nietypowy wybór dla klasyfikacji
- Basic - 2 ukryte warstwy po 32 neurony
- Modified - artykułowa architektura z funkcją straty 'binary cross entropy'



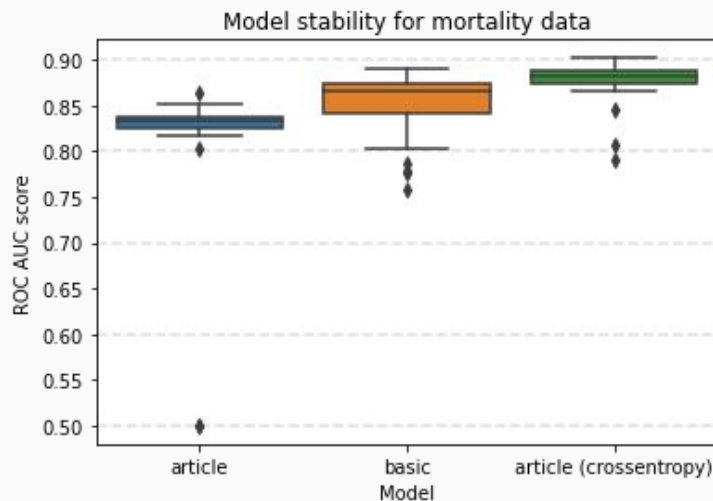
Poprawienie stabilności

Te modyfikacje znacznie poprawiły stabilność



Predykcja zgonu

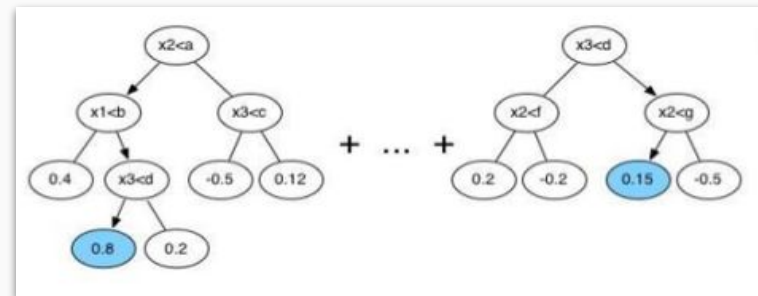
Podobny eksperyment przeprowadziliśmy dla zadania predykcji zgonu.



XGBoost

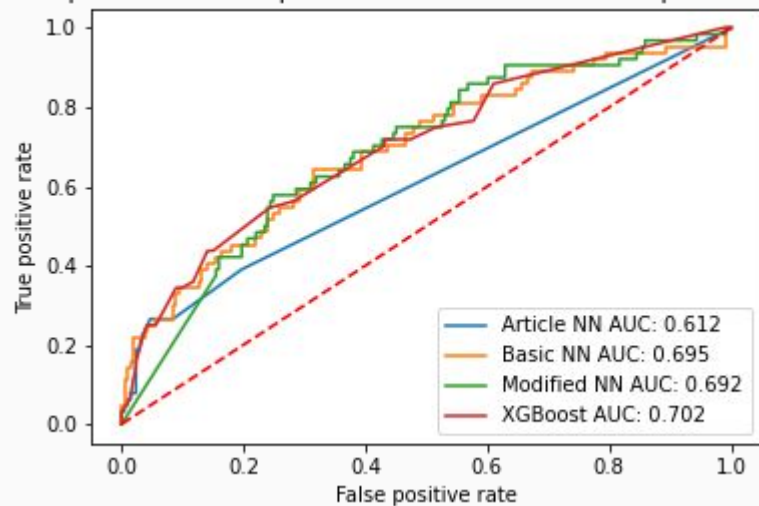
- Stworzyliśmy również modele XGBoost
- Znany i dobry algorytm.

XGBoost



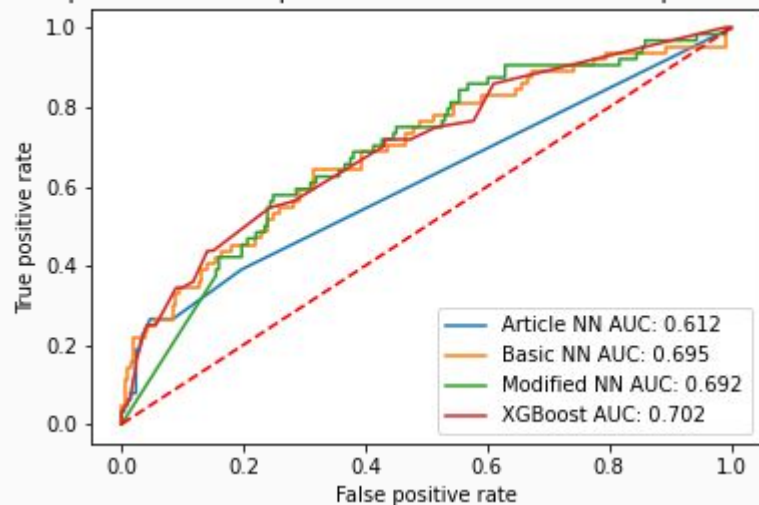
Porównanie efektywności

Comparison of model performance for ICU admission prediction task

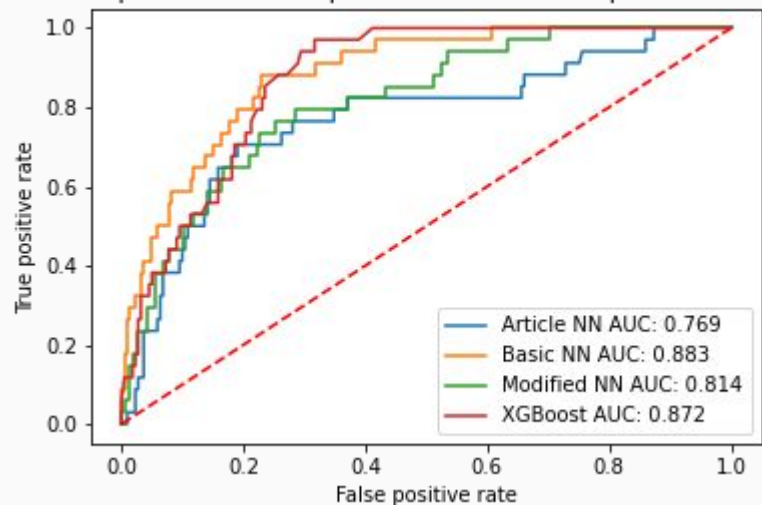


Porównanie efektywności

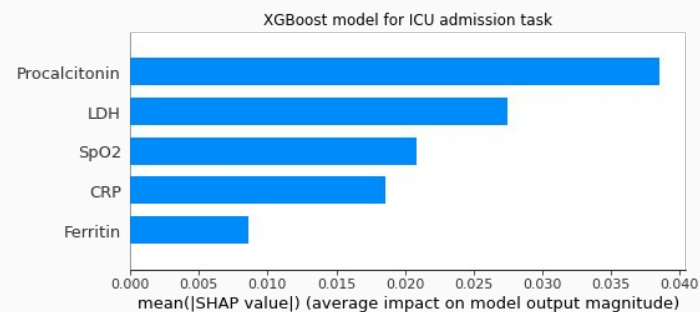
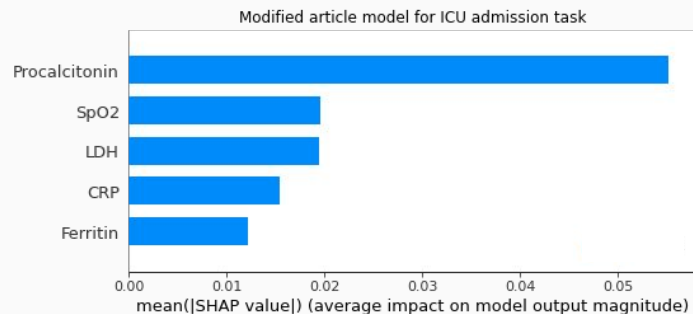
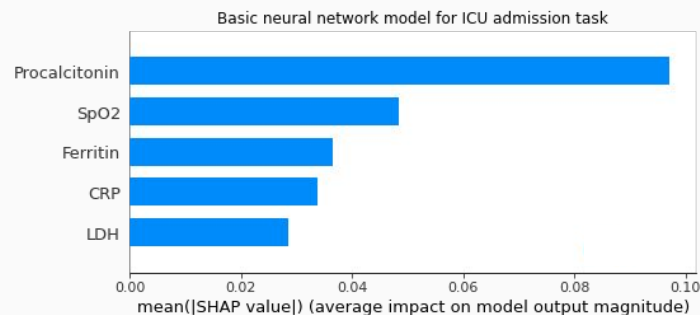
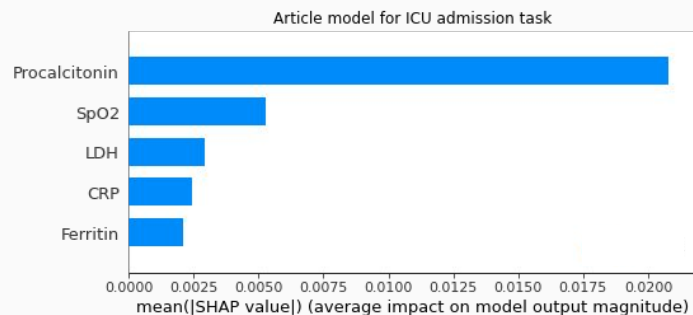
Comparison of model performance for ICU admission prediction task



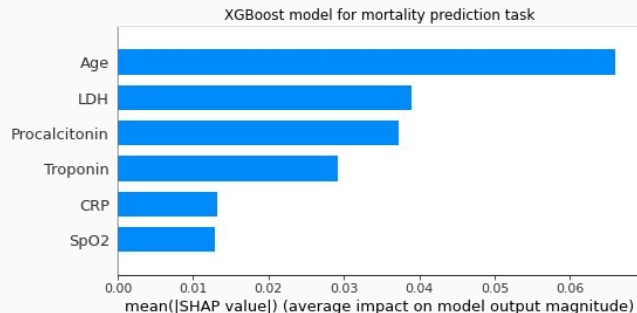
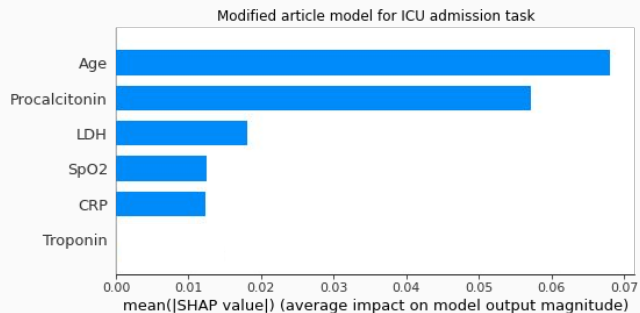
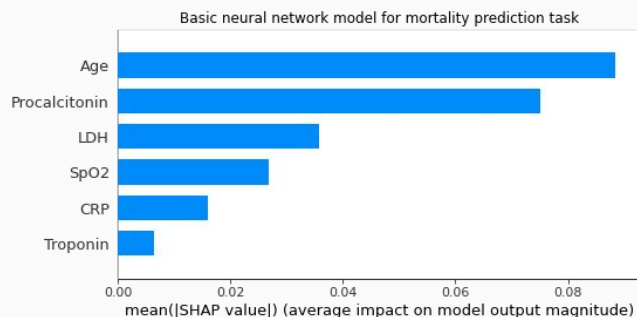
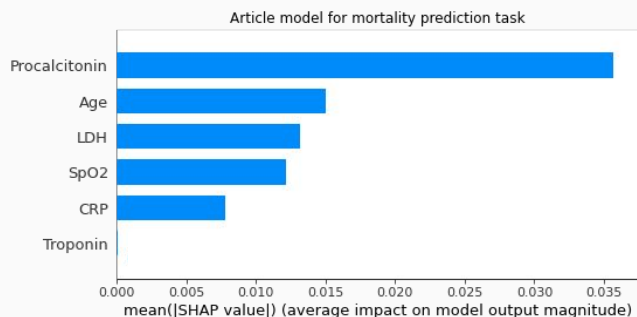
Comparison of model performance for death prediction task



Wyjaśnienia (ICU)

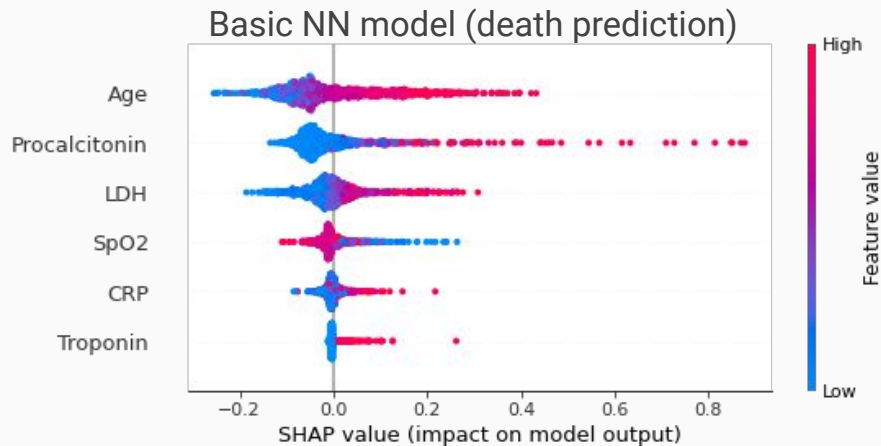


Wyjaśnienia (przewidywanie zgonu)



Dane

- Jedna cecha?
- Czy zadziała w innym szpitalu?
- Realia danych medycznych



Podsumowanie

Sieci, a modele drzewiaste:

- Oba algorytmy dobrze działają
- *Baseline* modele są ważne

Podsumowanie

Sieci, a modele drzewiaste:

- Oba algorytmy dobrze działają
- *Baseline* modele są ważne

Kryzys replikacji:

- Kod jest potrzebny w artykułach o tematyce ML