# Klasyfikacja spraw spornych Banku

Łukasz Waszak, Alior Bank SA

Michał Popielak, Alior Bank SA

LIPIEC 2024

Klasa $Y \sim \boldsymbol{X}$


## Cel – klasyfikacja zmiennej Y

- Y - Określenie szans na wygrania danej sprawy (wygra/przegra) – etykieta klasy (1/0).
- Zaklasyfikowanie/podział spraw (Y) do jednej z dwóch klas pod kątem konieczności tworzenia rezerwy (przepisy dot. rachunkowości UoR).

## Dane używane do klasyfikacji – X (dane z pozwu + dane z hurtowni)

- Dane ilościowe: Rozważano dane ilościowe z pozwu jak np. kwota pozwu oraz dane klienta z baz Banku jak np. kwota zobowiązań, wiek, ilość rachunków itp.
- Dane jakościowe: Typ produktu (14 kategorii) + typ pozwu (dane binarne, 13 zmiennych każda występująca w 2 kategoriach): „czy sprawa dot. TSUE”, „czy sprawa dot. funduszy inwestycyjnych”, „czy sprawa dot. kredytu darmowego”, „czy sprawa skarbowa”, „czy default” itp.

## Problemy klasyfikacji w analizowanym zagadnieniu

- Struktura prognozowanych klas (stosunek klasy 1 do 0) – algorytm SMOTE.
- Duża liczba zmiennych jakościowych – dodatkowe dane z hurtowni danych Banku nieużywane przez prawników.
- Pojawiające się nowe uwarunkowania prawne (zmienne w przyszłości vs przeszłość), które nie zostały uwzględnione na etapie budowania modelu – oparcie się na produktach (szczególnie w modelach regresyjnych).

Użycie algorytmu SMOTE do zbalansowania próby

```
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler, SMOTE
```

## Oprogramowanie i biblioteki

- Python

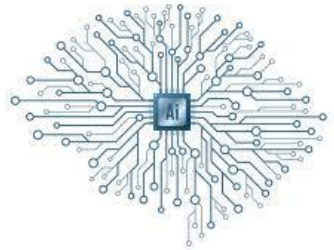

Biblioteki do oceny klasyfikacji

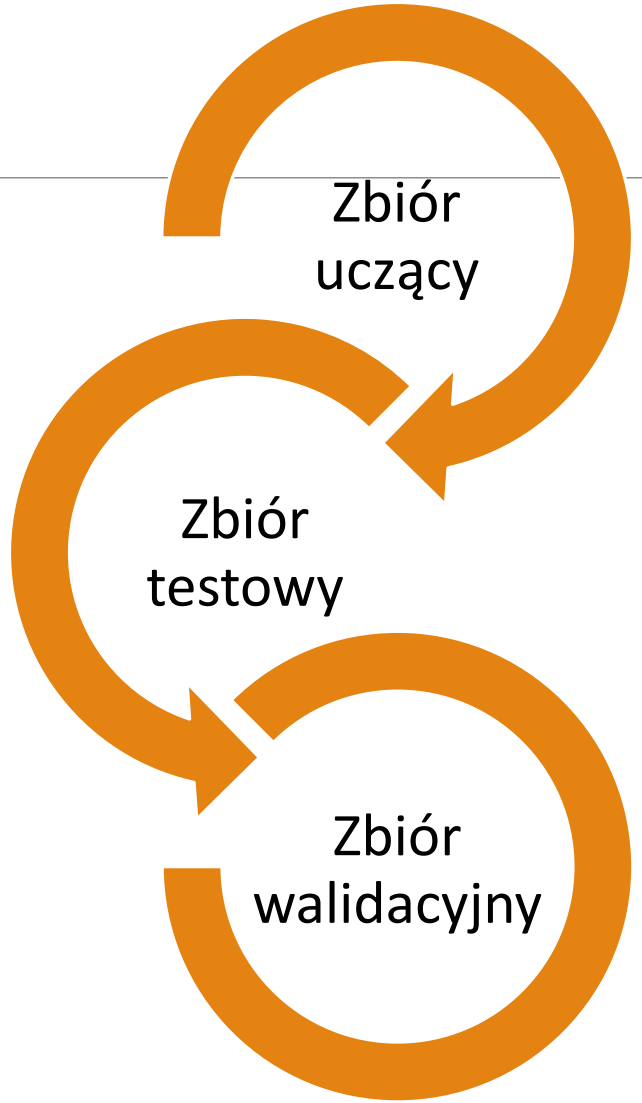
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score
```

Biblioteki do metod ML/AI


```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from xgboost import XGBClassifier
from sklearn.neural_network import MLPClassifier
```

# Model AI




## ❑ Model

- ❑ Do modelowania używamy rekordów/spraw, które mają status „Zamknięta” – tylko dla takich spraw nie zmienia się status przegrana/wygrana. Sprawy ze statusem „Otwarta” mogą zmieniać taki status.
- ❑ Zbiór rekordów do modelowania został podzielony na uczący i testowy (70:30).
- ❑ Model potrafi na zbiorze testowym identyfikować poprawnie 92-97% w zależności od klasy.
- ❑ Dla nowo zamykanych spraw (model walidacyjny) skuteczność modelu wynosi ok. 90% (vs 66% człowieka).

## Klasyfikacja spraw pod kątem wygrania/przegrania

W zakresie klasyfikacji spraw wygrana/przegrana, użyto drzewa i lasy losowe oraz ich modyfikacje, regresje logistyczną i sieć neuronową (j/n):

- Regresja logistyczna (accuracy 55%, LogisticRegression())
- Drzewo (accuracy 91%, DecisionTreeClassifier(max_depth = 9, random_state=42))
- Las losowy (accuracy 94%, RandomForestClassifier(n_estimators = 100))
- XTree (accuracy 94%, ExtraTreesClassifier(n_estimators=100))
- XGBoost (accuracy 95%, XGBClassifier())
- Sieć neuronowa (accuracy 70%, MLPClassifier(max_iter = 1000))

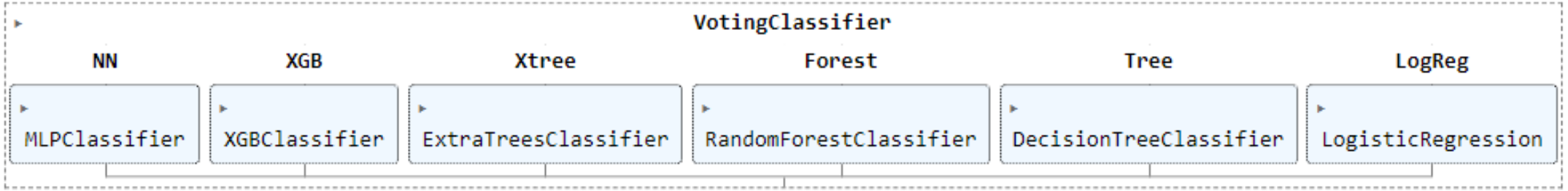
## Ocena jakościowa poszczególnych modeli – zbiór testowy

| Klasyfikator | Accuracy | Precision – 0 | Precision – 1 | F1-Score – 0 | F1-Score – 1 |
|---|---|---|---|---|---|
| Regresja logistyczna | 86% | 84% | 89% | 87% | 86% |
| Drzewo | 92% | 94% | 90% | 91% | 92% |
| Las | 94% | 95% | 93% | 94% | 94% |
| XTree | 94% | 95% | 93% | 94% | 95% |
| XGBoost | 94% | 96% | 93% | 94% | 94% |
| Sieć neuronowa | 67% | 63% | 73% | 71% | 62% |

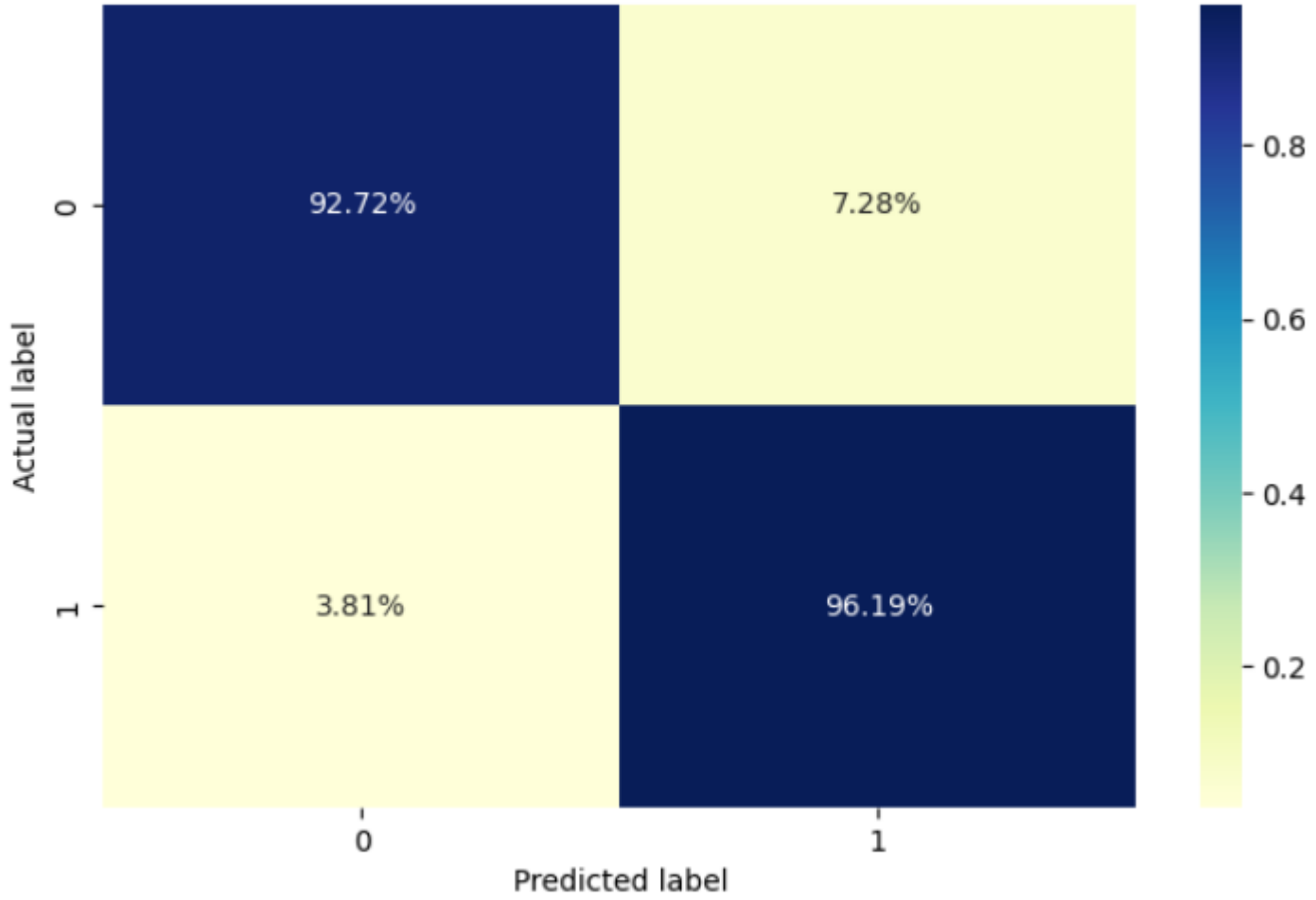
# Klasyfikator

```
wagi=[1, 4, 4, 3, 2, 1]
```

```
eclf = VotingClassifier(estimators=[('NN', clf6), ('XGB', clf5), ('Xtree', clf4), ('Forest', clf3), ('Tree', clf2), ('LogReg', cl
                        voting='soft', weights=wagi)
eclf.fit(X_train, y_train)
```

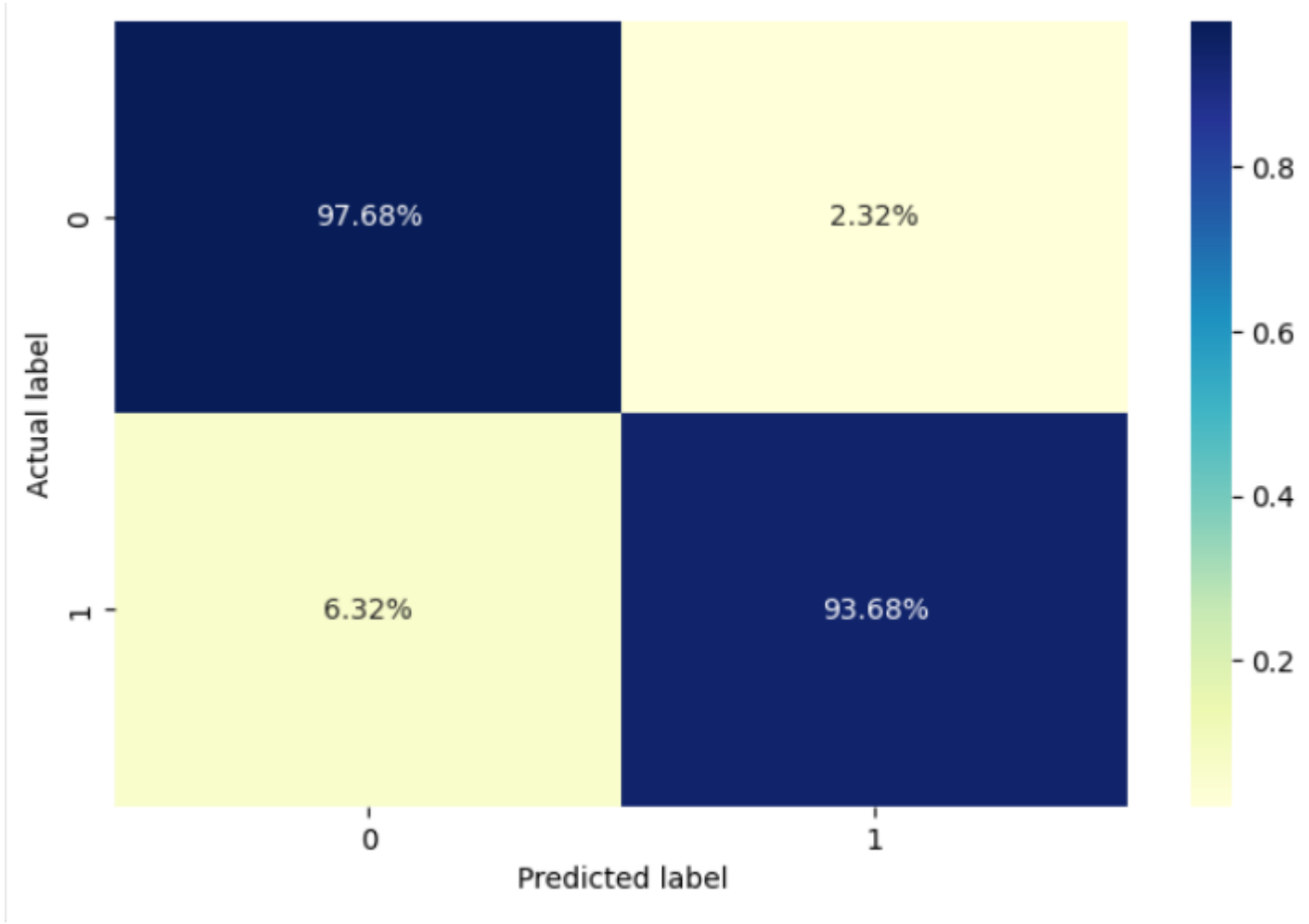


# Zbiór testowy – ocena modelu




| | 0 | 1 | accuracy |
|---|---|---|---|
| **precision** | 0.960169 | 0.930141 | 0.94458 |
| **recall** | 0.927169 | 0.961851 | 0.94458 |
| **f1-score** | 0.943381 | 0.945730 | 0.94458 |

# Cały zbiór – miary oceny modelu




| | 0 | 1 | accuracy |
|---|---|---|---|
| **precision** | 0.995513 | 0.737569 | 0.974166 |
| **recall** | 0.976767 | 0.936842 | 0.974166 |
| **f1-score** | 0.986051 | 0.825348 | 0.974166 |

# PODSUMOWANIE

## Wnioski

- Problemy z liczebnością w poszczególnych klas rozwiązano za pomocą algorytmy SMOTE.
- Model dla ponad 4 tys. spraw zamkniętych potrafi dokonywać automatycznej klasyfikacji z błędem na poziomie ok. 5%.
- Poziom skuteczności klasyfikacji modelu jest monitorowany od grudnia i charakteryzuje się poprawnością klasyfikacji na poziomie pomiędzy 80% a 95%.
- Na bazie powstałego modelu klasyfikacyjnego trwają dalsze prace dot. budowy innych modeli bazujących na wynikach zaprezentowanego.

## Bibliografia


- Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classification and Regression Trees”, Wadsworth, Belmont, CA, 1984.
- T. Hastie, R. Tibshirani and J. Friedman. “Elements of Statistical Learning”, Springer, 2009.
- L. Breiman, and A. Cutler, “Random Forests”, Springer, 2001
- P. Geurts, D. Ernst., and L. Wehenkel, “Extremely randomized trees”, Machine Learning, 63(1), 3-42, 2006.
- Classification methods in the diagnosis of breast cancer, Ł. Smaga, A. Ogloszka, Biometrical Letters Biometrical Letters 59(2), 2022.