Łączenie technik Under-Sampling i Over-Sampling dla zbioru danych niezbalansowanych

Weronika Belniak 249048

Politechnika Wrocławska

Streszczenie W niniejszym badaniu przeanalizowano wpływ technik samplingu, takich jak SMOTE i NearMiss, na modele klasyfikacyjne stosowane na niezbalansowanych zbiorach danych. Przeprowadzono eksperymenty na dziesięciu losowo wybranych zbiorach, oceniając metryki dokładności (Accuracy) i czułości (Recall).

Słowa kluczowe: Imbalance Dataset; SMOTE; NearMiss

1 Temat Projektu

1.1 Wprowadzenie

Dane niezbalansowane stanowią powszechny problem w analizie danych, który może znacząco wpłynąć na skuteczność modeli uczenia maszynowego. W przypadku, gdy jedna klasa jest znacznie liczniejsza od drugiej, modele uczące mogą wykazywać tendencję do faworyzowania dominującej klasy, co prowadzi do niewłaściwego generalizowania się na rzeczywiste przypadki. W rezultacie, dokładność i skuteczność tych modeli może być znacząco obniżona, zwłaszcza w kontekście klasyfikacji lub detekcji rzadkich zdarzeń.

Problem danych niezbalansowanych występuje w wielu dziedzinach, gdzie różnice w liczności klas są powszechne. Jednym z najczęstszych przykładów jest medycyna, gdzie detekcja rzadkich chorób może być trudna ze względu na niewielką liczbę przypadków chorobowych w porównaniu do zdrowych pacjentów. Podobne problemy można napotkać w dziedzinach cyberbezpieczeństwa, czy w analizie danych finansowych.

Undersampling i oversampling to dwie główne metody radzenia sobie z problemem danych niezbalansowanych.

Undersampling polega na redukcji liczby przypadków z dominującej klasy, aby zrównoważyć proporcje między klasami. Może to być osiągnięte poprzez losowe usunięcie części przypadków z dominującej klasy lub wybór próbki danych z mniejszą liczbą przypadków tej klasy. Undersampling może prowadzić do utraty istotnych informacji zawartych w usuniętych danych, zwłaszcza gdy dane są już rzadkie.

Oversampling polega na zwiększeniu liczby przypadków w mniejszej klasie poprzez duplikację istniejących przypadków lub generowanie nowych syntetycznych przypadków. Metody oversamplingu, takie jak SMOTE (Synthetic Minority Over-sampling Technique), generują nowe przykłady poprzez interpolację między istniejącymi przykładami mniejszej klasy. Oversampling może prowadzić do nadmiernego dopasowania (overfitting), zwłaszcza gdy generowane są zbyt zbliżone do istniejących przykładów.

1.2 Studia literaturowe

Podczas studiów literaturowych zbierano oraz analizowano istniejące publikacje naukowe, książki, artykuły prasowe i inne źródła informacji powiązane z tematyką danych niezbalansowanych oraz metodami radzenia sobie z problemami z tym związanymi. Zebrane publikacje umieszczone zostały w Bibliografi.

Problem niezbalansowanych zbiorów danych dotyczy sytuacji, w której liczba przykładów w poszczególnych klasach w zbiorze danych jest znacząco różna. Głównym wyzwaniem jest to, że modele uczenia maszynowego uczące się na takich danych mogą wykazywać tendencję do faworyzowania dominującej klasy kosztem mniejszościowej, co prowadzi do niezadowalającej skuteczności w generalizacji na rzeczywiste przypadki.Rozwiązanie tego problemu wymaga zastosowania odpowiednich technik, takich jak oversampling i undersampling, aby zapewnić równowagę między klasami i poprawić skuteczność modeli uczenia maszynowego.

Recall (sensitivity), parametr nazywany również czułością wykorzystany zostanie do oceny skuteczności. Jest to metryka używana w ocenie wydajności modeli klasyfikacyjnych, szczególnie w przypadku niezbalansowanych zbiorów danych. Definiuje ona stosunek liczby poprawnie sklasyfikowanych pozytywnych przypadków (tj. True Positive, TP) do sumy wszystkich pozytywnych przypadków w zbiorze danych (TP oraz False Negative, FN).

Wzór na recall można zapisać jako:

$$\frac{TP}{TP + FN}$$

Wartość recall wyraża zdolność modelu do identyfikowania wszystkich pozytywnych przypadków w stosunku do wszystkich rzeczywiście występujących pozytywnych przypadków. Innymi słowy, im wyższa wartość recall, tym lepiej model radzi sobie z wykrywaniem pozytywnych przypadków.

SMOTE (Synthetic Minority Over-sampling Technique) to technika oversamplingu, która została zaproponowana jako rozwiązanie problemu niezbalansowanych zbiorów danych. SMOTE polega na generowaniu syntetycznych przykładów mniejszościowej klasy poprzez interpolację między istniejącymi przykładami tej klasy. Główna idea SMOTE polega na tym, że zamiast kopiowania istniejących przykładów

mniejszościowej klasy, tworzone są nowe przypadki poprzez łączenie cech sąsiadujących przykładów tej klasy. Procedura ta ma na celu zmniejszenie ryzyka nadmiernego dopasowania (overfitting), które może wystąpić przy prostym kopiowaniu przykładów.

NearMiss to technika undersamplingu, która jest stosowana do radzenia sobie z problemem niezbalansowanych zbiorów danych poprzez redukcję liczby przypadków dominującej klasy. Głównym celem NearMiss jest wybranie próbki przypadków dominującej klasy, która jest "blisko" przypadków mniejszościowej klasy.

2 Przygotowanie do eksperymentu

2.1 Opis problemu i algorytmiki

Problem danych niezbalansowanych zostanie rozwiązany z wykorzystaniem podejścia, polegającego na łączeniu technik oversampling i undersampling, czyli zastosowaniu oversamplingu do mniejszej klasy oraz undersamplingu do większej klasy, aby zrównoważyć proporcje między nimi. Wykorzystane zostaną techniki SMOTE oraz NearMiss, w sposób przedstawiony na rysunku Fig. 1.

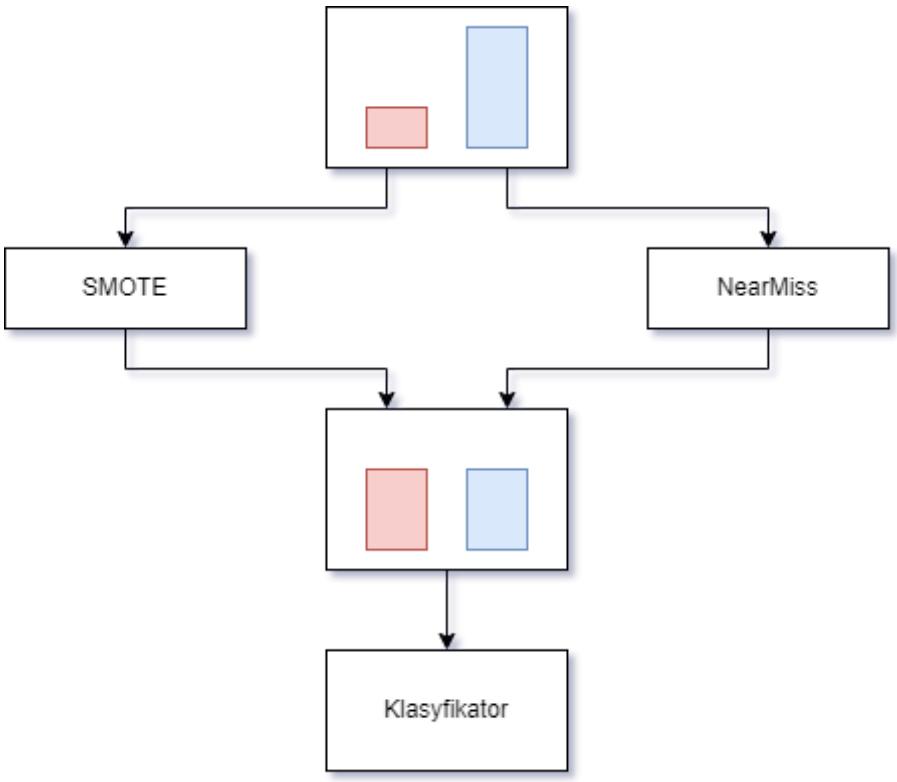
2.2 Plan eksperymentu

Badanie skupi się na porównaniu uzyskanych wyników w czterech przypadkach: na danych niezmodyfikowanych, zmodyfikowanych danych z użyciem techniki undersamplingu, zmodyfikowanych danych z użyciem techniki oversamplingu oraz zmodyfikowanych danych z użyciem metody łączącej oversampling i undersampling.

Dane wykorzystywane w trakcie przeprowadzania badania pochodzić będą ze strony KEEL. Wykorzystane zostanie 10 zbiorów danych. Każdy z wybranych zbiorów ma 2 klasy i charakteryzuje się danymi niezbalansowanymi.

W trakcie realizacji badań odpowiedzi na następujące pytania:

- Jak zmienia się wynik, mierzony metryką recall, w przypadku nieużycia żadnej techniki radzenia sobie z problemem danych niezbalansowanych, w porównaniu z zastosowaniem różnych technik?
- Jaki wpływ na wynik recall ma jedynie zastosowanie technik undersamplingu, czyli redukcji liczby przypadków dominującej klasy?
- Jakie rezultaty osiągniemy, stosując jedynie techniki oversamplingu, czyli zwiększania liczby przypadków mniejszościowej klasy?
- Jakie są efekty zastosowania kombinacji technik undersamplingu i oversamplingu, czyli równoczesne zmniejszanie liczby przypadków dominującej klasy i zwiększanie liczby przypadków mniejszościowej klasy?
- Jak różne proporcje oversampling i undersampling wpływają na metryki accuracy oraz recall?
- Które proporcje resamplingu (oversampling vs undersampling) dają najlepsze wyniki dla różnych typów modeli klasyfikacyjnych?



Rysunek 1: Diagram przedstawiający proces łączenia technik undersamplingu NearMiss i oversampingu SMOTE dla zbioru danych niezbalansowanych.

– Czy istnieją specyficzne proporcje resamplingu, które są szczególnie korzystne dla modelu drzewa decyzyjnego, k-NN czy Naive Bayes?

W ramach badania przeprowadzona zostanie analiza wpływu wspomnianych modyfikacji na parametry, takie jak metryka recall i czas treningu modelu. Celem jest zrozumienie, jak zastosowanie różnych technik, takich jak oversampling (np. SMOTE) i undersampling (np. NearMiss), wpływa na skuteczność modeli klasyfikacyjnych oraz na czas potrzebny do ich nauki. Ponadto, zbadany zostanie wpływ parametrów wejściowych używanych w technikach SMOTE i NearMiss.

W badaniu wykorzystana zostanie walidacja krzyżowa, czyli technika oceny modeli statystycznych i uczenia maszynowego, która jest używana do oszacowania wydajności modelu na nowych, niewidzianych danych.Głównym celem walidacji krzyżowej jest zapewnienie, że model będzie generalizował dobrze na niezależnym zbiorze danych, a nie tylko na danych treningowych.

Celem badania jest lepsze zrozumienie, jak techniki oversamplingu i undersamplingu wpływają na modele klasyfikacyjne w kontekście niezbalansowanych danych.

Techniki radzenia sobie z niezbalansowanymi danymi:

– Brak zastosowania techniki
– Oversampling
– Undersampling
– Kombinacja oversamplingu i undersamplingu

Protokół Badawczy:

Przygotowanie Danych:

– Przygotowanie zbioru danych treningowych i testowych.

Implementacja Modeli:

– Implementacja modeli klasyfikacyjnych, które będą używane w badaniu: Drzewa decyzyjne, Naiwny klasyfikator Bayesa oraz K najbliższych sąsiadów.

Przeprowadzenie Eksperymentu:

– Podział danych na zbiory treningowe i testowe.
– Wykorzystanie walidacji krzyżowej.
– Dla każdej techniki radzenia sobie z niezbalansowanymi danymi:
  • Trenowanie modelu na zbiorze treningowym.
  • Ocena wydajności modelu na zbiorze testowym za pomocą metryk accuracy i recall.

Analiza Wyników:

- Porównanie wyników wybranych metryk dla różnych technik radzenia sobie z niezbalansowanymi danymi.
- Identyfikacja najlepszych praktyk i optymalnych ustawień parametrów dla konkretnych przypadków problemów klasyfikacyjnych.

Raportowanie Wyników:

- Sporządzenie raportu zawierającego szczegółową analizę wyników eksperymentu oraz wnioski wyciągnięte na ich podstawie.

Projekt zostanie zrealizowany przy użyciu platformy Google Colab oraz języka Python, przy wykorzystaniu biblioteki scikit-learn do implementacji i testowania różnych technik radzenia sobie z problemem danych niezbalansowanych.

3 Przedstawienie wyników badań

Wyniki ewaluacji eksperymentalnej przedstawione są w formie tabel oraz wykresów. Badaniom poddane zostało 10 losowo wybranych zbiorów: ecoli-0-vs-1, glass0, glass1, haberman, vehicle1, vehicle2, wisconsin, yeast1, yeast3 oraz yeast4. W tabeli przedstawiono charakterystyki poszczególnych zbiorów.

Ilość danych w zbiorach				
Numer zbioru	Nazwa zbioru	Suma	Class 0	Class 1
1	ecoli-0-vs-1	220	143	77
2	glass0	214	144	70
3	glass1	214	138	76
4	haberman	306	225	81
5	vehicle1	846	629	217
6	vehicle2	846	628	218
7	wisconsin	683	444	239
8	yeast1	1484	1055	429
9	yeast3	1484	1321	163
10	yeast4	1484	1433	51

Tabela 1: Wybrane zbiory danych wraz z ich zawartością.

Analizie poddane zostały metryki Accuracy i Recall, gdzie metryka Accuracy mierzy ogólną poprawność klasyfikacji modelu, czyli stosunek liczby poprawnie sklasyfikowanych przypadków do liczby wszystkich przypadków, a metryka Recall mierzy zdolność modelu do poprawnego wykrycia wszystkich rzeczywistych pozytywnych przypadków (tzw. true positive) wśród wszystkich istniejących pozytywnych przypadków. Jest szczególnie przydatna w przypadkach, gdzie istnieje nierównowaga między klasami, a klasyfikacja pozytywna jest istotna.

3.1 Wyniki badań

Celem badania było uzyskanie wartości Accuracy i Recall dla wybranych modeli bez jakiejkolwiek modyfikacji danych oraz z użyciem różnych technik samplingu. Zastosowano 5-krotną weryfikację krzyżową w celu oceny modelu. Uzyskane wartości dla wybranych dziesięciu zbiorów danych są przedstawione w tabelach poniżej.

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	ecoli-0_vs_1	ecoli-0_vs_1	ecoli-0_vs_1	ecoli-0_vs_1
Classifier	DT	DT	DT	DT
Mean Accuracy	0,968	0,977	0,977	0,968
Std Accuracy	0,023	0,014	0,014	0,031
Mean Recall	0,970	0,986	0,986	0,962
Std Recall	0,046	0,018	0,018	0,062
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,982	0,982	0,982	0,982
Std Accuracy	0,009	0,009	0,009	0,009
Mean Recall	1,000	1,000	1,000	0,992
Std Recall	0,000	0,000	0,000	0,016
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,955	0,955	0,955	0,955
Std Accuracy	0,025	0,025	0,025	0,025
Mean Recall	0,992	0,992	0,992	0,992
Std Recall	0,016	0,016	0,016	0,016

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	glass0	glass0	glass0	glass0
Classifier	DT	DT	DT	DT
Mean Accuracy	0,822	0,808	0,808	0,813
Std Accuracy	0,048	0,059	0,059	0,043
Mean Recall	0,720	0,763	0,763	0,722
Std Recall	0,087	0,066	0,066	0,100
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,785	0,748	0,748	0,767
Std Accuracy	0,057	0,072	0,072	0,038
Mean Recall	0,681	0,803	0,803	0,786
Std Recall	0,087	0,055	0,055	0,033
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,640	0,650	0,650	0,645
Std Accuracy	0,064	0,054	0,054	0,057
Mean Recall	0,916	0,912	0,912	0,912
Std Recall	0,071	0,085	0,085	0,085

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	glass1	glass1	glass1	glass1
Classifier	DT	DT	DT	DT
Mean Accuracy	0,734	0,752	0,752	0,752
Std Accuracy	0,048	0,055	0,055	0,044
Mean Recall	0,597	0,618	0,618	0,664
Std Recall	0,130	0,080	0,080	0,092
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,799	0,794	0,794	0,790
Std Accuracy	0,035	0,049	0,049	0,026
Mean Recall	0,645	0,759	0,759	0,751
Std Recall	0,106	0,127	0,127	0,118
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,588	0,570	0,570	0,598
Std Accuracy	0,089	0,078	0,078	0,095
Mean Recall	0,902	0,878	0,878	0,902
Std Recall	0,075	0,080	0,080	0,075

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	haberman	haberman	haberman	haberman
Classifier	DT	DT	DT	DT
Mean Accuracy	0,660	0,614	0,614	0,624
Std Accuracy	0,062	0,068	0,068	0,074
Mean Recall	0,426	0,456	0,456	0,469
Std Recall	0,112	0,131	0,131	0,067
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,706	0,618	0,618	0,647
Std Accuracy	0,076	0,057	0,057	0,033
Mean Recall	0,297	0,504	0,504	0,596
Std Recall	0,142	0,138	0,138	0,109
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,748	0,735	0,735	0,739
Std Accuracy	0,047	0,024	0,024	0,037
Mean Recall	0,217	0,392	0,392	0,371
Std Recall	0,087	0,114	0,114	0,125

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	vehicle1	vehicle1	vehicle1	vehicle1
Classifier	DT	DT	DT	DT
Mean Accuracy	0,747	0,723	0,723	0,769
Std Accuracy	0,045	0,019	0,019	0,023
Mean Recall	0,515	0,524	0,524	0,606
Std Recall	0,070	0,080	0,080	0,045
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,729	0,690	0,690	0,687
Std Accuracy	0,023	0,024	0,024	0,038
Mean Recall	0,417	0,749	0,749	0,705
Std Recall	0,099	0,060	0,060	0,034
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,692	0,656	0,656	0,664
Std Accuracy	0,027	0,040	0,040	0,029
Mean Recall	0,648	0,700	0,700	0,691
Std Recall	0,075	0,068	0,068	0,082

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	vehicle2	vehicle2	vehicle2	vehicle2
Classifier	DT	DT	DT	DT
Mean Accuracy	0,968	0,967	0,967	0,966
Std Accuracy	0,010	0,017	0,017	0,019
Mean Recall	0,944	0,934	0,934	0,937
Std Recall	0,021	0,055	0,055	0,060
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,908	0,883	0,883	0,894
Std Accuracy	0,023	0,021	0,021	0,014
Mean Recall	0,776	0,895	0,895	0,905
Std Recall	0,072	0,043	0,043	0,057
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,808	0,758	0,758	0,800
Std Accuracy	0,046	0,055	0,055	0,034
Mean Recall	0,544	0,719	0,719	0,672
Std Recall	0,103	0,093	0,093	0,104

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	wisconsin	wisconsin	wisconsin	wisconsin
Classifier	DT	DT	DT	DT
Mean Accuracy	0,941	0,943	0,943	0,949
Std Accuracy	0,029	0,022	0,022	0,019
Mean Recall	0,894	0,911	0,911	0,933
Std Recall	0,045	0,026	0,026	0,026
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,977	0,980	0,980	0,977
Std Accuracy	0,013	0,014	0,014	0,013
Mean Recall	0,964	0,982	0,982	0,982
Std Recall	0,029	0,018	0,018	0,016
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,962	0,959	0,959	0,962
Std Accuracy	0,024	0,026	0,026	0,024
Mean Recall	0,972	0,972	0,972	0,977
Std Recall	0,029	0,029	0,029	0,021

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	yeast1	yeast1	yeast1	yeast1
Classifier	DT	DT	DT	DT
Mean Accuracy	0,701	0,697	0,697	0,703
Std Accuracy	0,033	0,020	0,020	0,019
Mean Recall	0,510	0,539	0,539	0,551
Std Recall	0,078	0,032	0,032	0,024
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,744	0,697	0,697	0,698
Std Accuracy	0,018	0,023	0,023	0,010
Mean Recall	0,429	0,675	0,675	0,661
Std Recall	0,055	0,039	0,039	0,036
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,319	0,326	0,326	0,311
Std Accuracy	0,023	0,021	0,021	0,022
Mean Recall	0,991	0,989	0,989	0,991
Std Recall	0,008	0,007	0,007	0,008

-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	yeast3	yeast3	yeast3	yeast3
Classifier	DT	DT	DT	DT
Mean Accuracy	0,935	0,933	0,933	0,940
Std Accuracy	0,015	0,011	0,011	0,013
Mean Recall	0,729	0,742	0,742	0,776
Std Recall	0,101	0,095	0,095	0,079
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,947	0,916	0,916	0,916
Std Accuracy	0,014	0,020	0,020	0,017
Mean Recall	0,686	0,847	0,847	0,821
Std Recall	0,066	0,047	0,047	0,055
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,307	0,334	0,334	0,321
Std Accuracy	0,072	0,080	0,080	0,079
Mean Recall	0,989	0,980	0,980	0,980
Std Recall	0,013	0,018	0,018	0,018
-	Original	SMOTE	NearMiss	SMOTE+NearMiss
File	yeast4	yeast4	yeast4	yeast4
Classifier	DT	DT	DT	DT
Mean Accuracy	0,951	0,935	0,935	0,933
Std Accuracy	0,011	0,021	0,021	0,022
Mean Recall	0,357	0,466	0,466	0,389
Std Recall	0,109	0,208	0,208	0,153
Classifier	kNN	kNN	kNN	kNN
Mean Accuracy	0,964	0,904	0,904	0,906
Std Accuracy	0,011	0,016	0,016	0,017
Mean Recall	0,122	0,675	0,675	0,626
Std Recall	0,090	0,089	0,089	0,034
Classifier	GNB	GNB	GNB	GNB
Mean Accuracy	0,172	0,197	0,197	0,193
Std Accuracy	0,050	0,052	0,052	0,050
Mean Recall	0,943	0,921	0,921	0,943
Std Recall	0,079	0,075	0,075	0,079

Tabela 2: Porównanie metryk Accuracy i Recall.

Zauważyć można, że wykorzystanie technik samplingu wpływa na badane metryki, zazwyczaj poprzez zmniejszenie wartości Accuracy, a zwiększenie metryki Recall.

Analizując uzyskane wyniki można także wyciągnąć wnioski odnośnie wykorzystanych modeli. Model GaussianNB w części przypadków uzyskuje metrykę Accuracy na poziomie niższym niż 0,5, co oznacza, że jest wynik jest gorszy niż wynik uzyskiwany przez klasyfikator losowy.

Przeprowadzono także parowe testy statystyczne, które miały na celu porównanie między sobą modeli i wskazanie w każdej z par, czy jeden z modeli jest statystycznie znacząco lepszy od drugiego.

ecoli-0-vs-1	DT	kNN	GNB
DT	0	0	0
kNN	1	0	1
GNB	1	0	0

glass0	DT	kNN	GNB
DT	0	1	0
kNN	0	0	0
GNB	1	1	0

glass1	DT	kNN	GNB
DT	0	0	0
kNN	1	0	0
GNB	1	1	0

haberman	DT	kNN	GNB
DT	0	1	1
kNN	0	0	1
GNB	0	0	0

vehicle1	DT	kNN	GNB
DT	0	1	0
kNN	0	0	0
GNB	1	1	0

vehicle2	DT	kNN	GNB
DT	0	1	1
kNN	0	0	1
GNB	0	0	0

wisconsin	DT	kNN	GNB
DT	0	0	0
kNN	1	0	0
GNB	1	1	0

yeast1	DT	kNN	GNB
DT	0	1	0
kNN	0	0	0
GNB	1	1	0

yeast3	DT	kNN	GNB
DT	0	1	0
kNN	0	0	0
GNB	1	1	0

yeast4	DT	kNN	GNB
DT	0	1	0
kNN	0	0	0
GNB	1	1	0

Z przeprowadzonych badań wynika, że uzyskane wyniki były rozbieżne dla każdego ze zbiorów wynik i nie można wskazać jednego najlepszego klasyfikatora dla wszystkich zbiorów.

Podczas badania skupiono się także na ocenie wpływu różnych proporcji podziału danych na wybrane metryki modeli klasyfikacyjnych, gdzie na jednej części zbioru danych stosuje się technikę undersamplingu, a na drugiej oversamplingu. Proporcje są przedstawione w postaci NearMiss:SMOTE. Celem jest zrozumienie, jak różne kombinacje tych technik wpływają na metryki Accuracy oraz Recall. Wyniki dla metryki Accuracy przedstawione zostały w poniższej tabeli.

-	ecoli-0-vs-1	ecoli-0-vs-1	ecoli-0-vs-1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	**0,973**	**0,982**	**0,955**
Std Accuracy	0,017	0,017	0,025
Ratio	2:8	2:8	2:8
Mean Accuracy	**0,964**	**0,986**	**0,955**
Std Accuracy	0,027	0,011	0,025
Ratio	3:7	3:7	3:7
Mean Accuracy	**0,982**	**0,982**	**0,955**
Std Accuracy	0,017	0,009	0,025
Ratio	4:6	4:6	4:6
Mean Accuracy	**0,982**	**0,982**	**0,959**
Std Accuracy	0,017	0,009	0,017
Ratio	5:5	5:5	5:5
Mean Accuracy	**0,982**	**0,977**	**0,959**
Std Accuracy	0,017	0,000	0,017
Ratio	6:4	6:4	6:4
Mean Accuracy	**0,955**	**0,982**	**0,964**
Std Accuracy	0,032	0,009	0,018
Ratio	7:3	7:3	7:3
Mean Accuracy	**0,968**	**0,977**	**0,959**
Std Accuracy	0,034	0,014	0,017
Ratio	8:2	8:2	8:2
Mean Accuracy	**0,955**	**0,982**	**0,964**
Std Accuracy	0,032	0,009	0,018
Ratio	9:1	9:1	9:1
Mean Accuracy	**0,950**	**0,982**	**0,968**
Std Accuracy	0,036	0,009	0,011

File	glass0	glass0	glass0
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	**0,794**	**0,767**	**0,645**
Std Accuracy	0,072	0,065	0,044
Ratio	2:8	2:8	2:8
Mean Accuracy	**0,846**	**0,776**	**0,640**
Std Accuracy	0,063	0,053	0,057
Ratio	3:7	3:7	3:7
Mean Accuracy	**0,841**	**0,771**	**0,645**
Std Accuracy	0,044	0,055	0,057
Ratio	4:6	4:6	4:6
Mean Accuracy	**0,780**	**0,790**	**0,650**
Std Accuracy	0,056	0,050	0,044
Ratio	5:5	5:5	5:5
Mean Accuracy	**0,804**	**0,776**	**0,645**
Std Accuracy	0,051	0,047	0,050
Ratio	6:4	6:4	6:4
Mean Accuracy	**0,804**	**0,752**	**0,636**
Std Accuracy	0,023	0,031	0,068
Ratio	7:3	7:3	7:3
Mean Accuracy	**0,789**	**0,725**	**0,659**
Std Accuracy	0,052	0,060	0,050
Ratio	8:2	8:2	8:2
Mean Accuracy	**0,780**	**0,674**	**0,654**
Std Accuracy	0,088	0,120	0,050
Ratio	9:1	9:1	9:1
Mean Accuracy	**0,728**	**0,659**	**0,663**
Std Accuracy	0,070	0,129	0,041
File	glass1	glass1	glass1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	**0,761**	**0,775**	**0,556**
Std Accuracy	0,037	0,056	0,076
Ratio	2:8	2:8	2:8
Mean Accuracy	**0,766**	**0,780**	**0,584**
Std Accuracy	0,069	0,061	0,084
Ratio	3:7	3:7	3:7
Mean Accuracy	**0,734**	**0,757**	**0,579**
Std Accuracy	0,011	0,047	0,072
Ratio	4:6	4:6	4:6
Mean Accuracy	**0,724**	**0,761**	**0,579**
Std Accuracy	0,045	0,052	0,068
Ratio	5:5	5:5	5:5
Mean Accuracy	**0,761**	**0,771**	**0,584**
Std Accuracy	0,069	0,054	0,070
Ratio	6:4	6:4	6:4
Mean Accuracy	**0,743**	**0,729**	**0,584**
Std Accuracy	0,072	0,048	0,060
Ratio	7:3	7:3	7:3
Mean Accuracy	**0,724**	**0,757**	**0,514**
Std Accuracy	0,031	0,052	0,036
Ratio	8:2	8:2	8:2
Mean Accuracy	**0,687**	**0,683**	**0,463**
Std Accuracy	0,064	0,075	0,049
Ratio	9:1	9:1	9:1
Mean Accuracy	**0,715**	**0,664**	**0,444**
Std Accuracy	0,076	0,064	0,041

File	haberman	haberman	haberman
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,650	0,628	0,748
Std Accuracy	0,096	0,038	0,042
Ratio	2:8	2:8	2:8
Mean Accuracy	0,654	0,638	0,742
Std Accuracy	0,081	0,053	0,033
Ratio	3:7	3:7	3:7
Mean Accuracy	0,608	0,621	0,748
Std Accuracy	0,085	0,057	0,047
Ratio	4:6	4:6	4:6
Mean Accuracy	0,634	0,644	0,702
Std Accuracy	0,066	0,040	0,048
Ratio	5:5	5:5	5:5
Mean Accuracy	0,634	0,611	0,719
Std Accuracy	0,074	0,063	0,025
Ratio	6:4	6:4	6:4
Mean Accuracy	0,579	0,637	0,716
Std Accuracy	0,044	0,043	0,027
Ratio	7:3	7:3	7:3
Mean Accuracy	0,618	0,650	0,722
Std Accuracy	0,091	0,023	0,063
Ratio	8:2	8:2	8:2
Mean Accuracy	0,644	0,660	0,719
Std Accuracy	0,059	0,058	0,044
Ratio	9:1	9:1	9:1
Mean Accuracy	0,582	0,650	0,716
Std Accuracy	0,036	0,086	0,053
File	vehicle1	vehicle1	vehicle1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,753	0,670	0,660
Std Accuracy	0,011	0,019	0,040
Ratio	2:8	2:8	2:8
Mean Accuracy	0,728	0,686	0,669
Std Accuracy	0,039	0,014	0,021
Ratio	3:7	3:7	3:7
Mean Accuracy	0,764	0,684	0,674
Std Accuracy	0,045	0,012	0,019
Ratio	4:6	4:6	4:6
Mean Accuracy	0,741	0,677	0,663
Std Accuracy	0,020	0,037	0,023
Ratio	5:5	5:5	5:5
Mean Accuracy	0,754	0,673	0,662
Std Accuracy	0,029	0,026	0,021
Ratio	6:4	6:4	6:4
Mean Accuracy	0,748	0,675	0,664
Std Accuracy	0,043	0,018	0,025
Ratio	7:3	7:3	7:3
Mean Accuracy	0,735	0,667	0,662
Std Accuracy	0,032	0,033	0,033
Ratio	8:2	8:2	8:2
Mean Accuracy	0,721	0,656	0,669
Std Accuracy	0,034	0,021	0,023
Ratio	9:1	9:1	9:1
Mean Accuracy	0,693	0,682	0,671
Std Accuracy	0,030	0,013	0,030

File	vehicle2	vehicle2	vehicle2
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,954	0,876	0,768
Std Accuracy	0,015	0,016	0,048
Ratio	2:8	2:8	2:8
Mean Accuracy	0,956	0,875	0,779
Std Accuracy	0,007	0,021	0,030
Ratio	3:7	3:7	3:7
Mean Accuracy	0,956	0,876	0,768
Std Accuracy	0,017	0,025	0,038
Ratio	4:6	4:6	4:6
Mean Accuracy	0,962	0,875	0,769
Std Accuracy	0,011	0,028	0,035
Ratio	5:5	5:5	5:5
Mean Accuracy	0,953	0,876	0,778
Std Accuracy	0,017	0,034	0,062
Ratio	6:4	6:4	6:4
Mean Accuracy	0,957	0,873	0,778
Std Accuracy	0,018	0,037	0,047
Ratio	7:3	7:3	7:3
Mean Accuracy	0,959	0,883	0,777
Std Accuracy	0,012	0,032	0,019
Ratio	8:2	8:2	8:2
Mean Accuracy	0,937	0,887	0,760
Std Accuracy	0,021	0,028	0,020
Ratio	9:1	9:1	9:1
Mean Accuracy	0,936	0,887	0,641
Std Accuracy	0,017	0,025	0,036
File	wisconsin	wisconsin	wisconsin
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,941	0,977	0,961
Std Accuracy	0,028	0,017	0,027
Ratio	2:8	2:8	2:8
Mean Accuracy	0,950	0,975	0,962
Std Accuracy	0,022	0,015	0,027
Ratio	3:7	3:7	3:7
Mean Accuracy	0,946	0,977	0,963
Std Accuracy	0,019	0,012	0,025
Ratio	4:6	4:6	4:6
Mean Accuracy	0,944	0,978	0,963
Std Accuracy	0,023	0,015	0,022
Ratio	5:5	5:5	5:5
Mean Accuracy	0,941	0,978	0,961
Std Accuracy	0,024	0,015	0,025
Ratio	6:4	6:4	6:4
Mean Accuracy	0,941	0,977	0,959
Std Accuracy	0,020	0,017	0,026
Ratio	7:3	7:3	7:3
Mean Accuracy	0,937	0,977	0,958
Std Accuracy	0,019	0,013	0,024
Ratio	8:2	8:2	8:2
Mean Accuracy	0,949	0,972	0,959
Std Accuracy	0,016	0,016	0,024
Ratio	9:1	9:1	9:1
Mean Accuracy	0,933	0,978	0,961
Std Accuracy	0,020	0,013	0,021

File	yeast1	yeast1	yeast1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,705	0,699	0,325
Std Accuracy	0,017	0,011	0,026
Ratio	2:8	2:8	2:8
Mean Accuracy	0,722	0,697	0,322
Std Accuracy	0,032	0,016	0,025
Ratio	3:7	3:7	3:7
Mean Accuracy	0,694	0,687	0,324
Std Accuracy	0,013	0,025	0,022
Ratio	4:6	4:6	4:6
Mean Accuracy	0,684	0,691	0,319
Std Accuracy	0,032	0,013	0,027
Ratio	5:5	5:5	5:5
Mean Accuracy	0,668	0,693	0,316
Std Accuracy	0,018	0,029	0,026
Ratio	6:4	6:4	6:4
Mean Accuracy	0,664	0,693	0,311
Std Accuracy	0,008	0,027	0,025
Ratio	7:3	7:3	7:3
Mean Accuracy	0,648	0,689	0,307
Std Accuracy	0,017	0,026	0,018
Ratio	8:2	8:2	8:2
Mean Accuracy	0,633	0,687	0,300
Std Accuracy	0,035	0,022	0,025
Ratio	9:1	9:1	9:1
Mean Accuracy	0,606	0,692	0,535
Std Accuracy	0,053	0,034	0,191
File	yeast3	yeast3	yeast3
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,934	0,916	0,325
Std Accuracy	0,008	0,020	0,064
Ratio	2:8	2:8	2:8
Mean Accuracy	0,935	0,911	0,320
Std Accuracy	0,021	0,014	0,079
Ratio	3:7	3:7	3:7
Mean Accuracy	0,928	0,914	0,338
Std Accuracy	0,021	0,018	0,092
Ratio	4:6	4:6	4:6
Mean Accuracy	0,929	0,920	0,357
Std Accuracy	0,021	0,015	0,102
Ratio	5:5	5:5	5:5
Mean Accuracy	0,931	0,917	0,329
Std Accuracy	0,021	0,017	0,084
Ratio	6:4	6:4	6:4
Mean Accuracy	0,923	0,920	0,340
Std Accuracy	0,014	0,019	0,080
Ratio	7:3	7:3	7:3
Mean Accuracy	0,920	0,918	0,334
Std Accuracy	0,027	0,022	0,097
Ratio	8:2	8:2	8:2
Mean Accuracy	0,908	0,919	0,461
Std Accuracy	0,018	0,021	0,255
Ratio	9:1	9:1	9:1
Mean Accuracy	0,890	0,907	0,520
Std Accuracy	0,024	0,023	0,342

File	yeast4	yeast4	yeast4
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Accuracy	0,932	0,901	0,206
Std Accuracy	0,016	0,023	0,056
Ratio	2:8	2:8	2:8
Mean Accuracy	0,931	0,900	0,178
Std Accuracy	0,021	0,025	0,048
Ratio	3:7	3:7	3:7
Mean Accuracy	0,926	0,893	0,210
Std Accuracy	0,012	0,030	0,064
Ratio	4:6	4:6	4:6
Mean Accuracy	0,937	0,904	0,238
Std Accuracy	0,015	0,029	0,083
Ratio	5:5	5:5	5:5
Mean Accuracy	0,927	0,901	0,230
Std Accuracy	0,016	0,031	0,087
Ratio	6:4	6:4	6:4
Mean Accuracy	0,929	0,904	0,225
Std Accuracy	0,014	0,022	0,104
Ratio	7:3	7:3	7:3
Mean Accuracy	0,930	0,893	0,229
Std Accuracy	0,017	0,027	0,085
Ratio	8:2	8:2	8:2
Mean Accuracy	0,893	0,873	0,299
Std Accuracy	0,030	0,028	0,232
Ratio	9:1	9:1	9:1
Mean Accuracy	0,883	0,869	0,582
Std Accuracy	0,022	0,058	0,346

Tabela 3: Porównanie wpływu łączenie technik samplingu na metrykę Accuracy.

Wyniki dla metryki Recall przedstawione zostały w poniższej tabeli.

File	ecoli-0_vs_1	ecoli-0_vs_1	ecoli-0_vs_1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,957	0,992	0,992
Std Recall	0,040	0,016	0,016
Ratio	2:8	2:8	2:8
Mean Recall	0,981	1,000	0,992
Std Recall	0,039	0,000	0,016
Ratio	3:7	3:7	3:7
Mean Recall	0,987	1,000	0,992
Std Recall	0,026	0,000	0,016
Ratio	4:6	4:6	4:6
Mean Recall	0,963	1,000	0,992
Std Recall	0,048	0,000	0,016
Ratio	5:5	5:5	5:5
Mean Recall	0,987	0,992	0,992
Std Recall	0,026	0,016	0,016
Ratio	6:4	6:4	6:4
Mean Recall	0,950	1,000	0,992
Std Recall	0,061	0,000	0,016
Ratio	7:3	7:3	7:3
Mean Recall	0,974	0,992	0,992
Std Recall	0,052	0,016	0,016
Ratio	8:2	8:2	8:2
Mean Recall	0,966	1,000	0,992
Std Recall	0,050	0,000	0,016
Ratio	9:1	9:1	9:1
Mean Recall	0,942	1,000	0,992
Std Recall	0,071	0,000	0,016

File	glass0	glass0	glass0
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,761	0,801	0,926
Std Recall	0,096	0,119	0,066
Ratio	2:8	2:8	2:8
Mean Recall	0,729	0,831	0,897
Std Recall	0,078	0,061	0,101
Ratio	3:7	3:7	3:7
Mean Recall	0,783	0,770	0,912
Std Recall	0,051	0,108	0,085
Ratio	4:6	4:6	4:6
Mean Recall	0,674	0,824	0,915
Std Recall	0,137	0,107	0,057
Ratio	5:5	5:5	5:5
Mean Recall	0,750	0,813	0,900
Std Recall	0,090	0,110	0,075
Ratio	6:4	6:4	6:4
Mean Recall	0,785	0,754	0,900
Std Recall	0,074	0,064	0,060
Ratio	7:3	7:3	7:3
Mean Recall	0,805	0,725	0,915
Std Recall	0,059	0,124	0,057
Ratio	8:2	8:2	8:2
Mean Recall	0,765	0,723	0,877
Std Recall	0,116	0,133	0,063
Ratio	9:1	9:1	9:1
Mean Recall	0,734	0,738	0,879
Std Recall	0,104	0,119	0,080
File	glass1	glass1	glass1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,640	0,742	0,862
Std Recall	0,044	0,121	0,060
Ratio	2:8	2:8	2:8
Mean Recall	0,720	0,714	0,897
Std Recall	0,032	0,142	0,041
Ratio	3:7	3:7	3:7
Mean Recall	0,621	0,698	0,886
Std Recall	0,093	0,105	0,060
Ratio	4:6	4:6	4:6
Mean Recall	0,675	0,728	0,752
Std Recall	0,137	0,099	0,227
Ratio	5:5	5:5	5:5
Mean Recall	0,678	0,734	0,809
Std Recall	0,091	0,083	0,086
Ratio	6:4	6:4	6:4
Mean Recall	0,643	0,673	0,721
Std Recall	0,063	0,114	0,211
Ratio	7:3	7:3	7:3
Mean Recall	0,625	0,715	0,563
Std Recall	0,091	0,102	0,275
Ratio	8:2	8:2	8:2
Mean Recall	0,715	0,695	0,327
Std Recall	0,070	0,122	0,219
Ratio	9:1	9:1	9:1
Mean Recall	0,729	0,684	0,260
Std Recall	0,062	0,105	0,108

File	haberman	haberman	haberman
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,479	0,488	0,392
Std Recall	0,128	0,119	0,115
Ratio	2:8	2:8	2:8
Mean Recall	0,481	0,558	0,377
Std Recall	0,070	0,031	0,130
Ratio	3:7	3:7	3:7
Mean Recall	0,447	0,522	0,422
Std Recall	0,105	0,110	0,107
Ratio	4:6	4:6	4:6
Mean Recall	0,553	0,486	0,381
Std Recall	0,062	0,140	0,121
Ratio	5:5	5:5	5:5
Mean Recall	0,492	0,523	0,397
Std Recall	0,113	0,075	0,118
Ratio	6:4	6:4	6:4
Mean Recall	0,491	0,606	0,393
Std Recall	0,149	0,055	0,126
Ratio	7:3	7:3	7:3
Mean Recall	0,534	0,580	0,463
Std Recall	0,078	0,030	0,125
Ratio	8:2	8:2	8:2
Mean Recall	0,657	0,584	0,430
Std Recall	0,103	0,056	0,113
Ratio	9:1	9:1	9:1
Mean Recall	0,669	0,588	0,521
Std Recall	0,057	0,076	0,057
File	vehicle1	vehicle1	vehicle1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,579	0,707	0,709
Std Recall	0,095	0,068	0,082
Ratio	2:8	2:8	2:8
Mean Recall	0,549	0,732	0,681
Std Recall	0,040	0,055	0,071
Ratio	3:7	3:7	3:7
Mean Recall	0,612	0,708	0,680
Std Recall	0,088	0,078	0,065
Ratio	4:6	4:6	4:6
Mean Recall	0,590	0,713	0,684
Std Recall	0,050	0,045	0,078
Ratio	5:5	5:5	5:5
Mean Recall	0,576	0,665	0,699
Std Recall	0,070	0,075	0,061
Ratio	6:4	6:4	6:4
Mean Recall	0,620	0,667	0,708
Std Recall	0,050	0,071	0,074
Ratio	7:3	7:3	7:3
Mean Recall	0,675	0,658	0,694
Std Recall	0,051	0,085	0,085
Ratio	8:2	8:2	8:2
Mean Recall	0,648	0,676	0,680
Std Recall	0,086	0,096	0,078
Ratio	9:1	9:1	9:1
Mean Recall	0,664	0,655	0,638
Std Recall	0,126	0,073	0,059

File	vehicle2	vehicle2	vehicle2
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,929	0,896	0,690
Std Recall	0,045	0,052	0,113
Ratio	2:8	2:8	2:8
Mean Recall	0,941	0,902	0,717
Std Recall	0,048	0,036	0,087
Ratio	3:7	3:7	3:7
Mean Recall	0,938	0,886	0,701
Std Recall	0,030	0,056	0,075
Ratio	4:6	4:6	4:6
Mean Recall	0,933	0,879	0,682
Std Recall	0,030	0,037	0,099
Ratio	5:5	5:5	5:5
Mean Recall	0,940	0,890	0,646
Std Recall	0,062	0,047	0,100
Ratio	6:4	6:4	6:4
Mean Recall	0,914	0,852	0,629
Std Recall	0,048	0,072	0,111
Ratio	7:3	7:3	7:3
Mean Recall	0,917	0,882	0,614
Std Recall	0,024	0,050	0,114
Ratio	8:2	8:2	8:2
Mean Recall	0,907	0,887	0,677
Std Recall	0,051	0,033	0,080
Ratio	9:1	9:1	9:1
Mean Recall	0,920	0,866	0,644
Std Recall	0,042	0,030	0,063
File	wisconsin	wisconsin	wisconsin
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,871	0,973	0,972
Std Recall	0,031	0,029	0,029
Ratio	2:8	2:8	2:8
Mean Recall	0,932	0,987	0,977
Std Recall	0,044	0,017	0,027
Ratio	3:7	3:7	3:7
Mean Recall	0,909	0,977	0,982
Std Recall	0,037	0,021	0,018
Ratio	4:6	4:6	4:6
Mean Recall	0,913	0,976	0,977
Std Recall	0,036	0,031	0,021
Ratio	5:5	5:5	5:5
Mean Recall	0,906	0,978	0,969
Std Recall	0,023	0,021	0,026
Ratio	6:4	6:4	6:4
Mean Recall	0,913	0,981	0,969
Std Recall	0,037	0,023	0,026
Ratio	7:3	7:3	7:3
Mean Recall	0,923	0,976	0,969
Std Recall	0,031	0,031	0,026
Ratio	8:2	8:2	8:2
Mean Recall	0,948	0,976	0,972
Std Recall	0,036	0,031	0,029
Ratio	9:1	9:1	9:1
Mean Recall	0,915	0,972	0,977
Std Recall	0,027	0,029	0,021

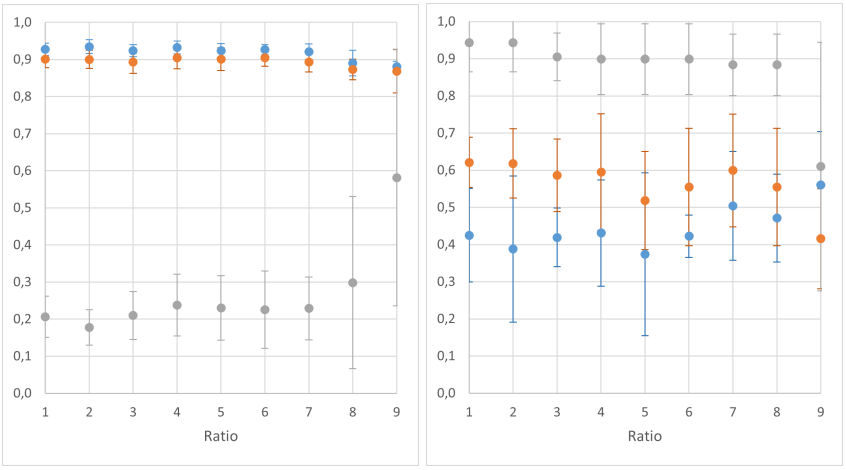
File	yeast1	yeast1	yeast1
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,532	0,678	0,991
Std Recall	0,067	0,038	0,008
Ratio	2:8	2:8	2:8
Mean Recall	0,573	0,671	0,989
Std Recall	0,064	0,036	0,007
Ratio	3:7	3:7	3:7
Mean Recall	0,551	0,657	0,991
Std Recall	0,034	0,035	0,008
Ratio	4:6	4:6	4:6
Mean Recall	0,595	0,631	0,994
Std Recall	0,055	0,045	0,008
Ratio	5:5	5:5	5:5
Mean Recall	0,555	0,621	0,994
Std Recall	0,066	0,049	0,008
Ratio	6:4	6:4	6:4
Mean Recall	0,572	0,607	0,991
Std Recall	0,028	0,078	0,008
Ratio	7:3	7:3	7:3
Mean Recall	0,586	0,592	0,998
Std Recall	0,063	0,059	0,005
Ratio	8:2	8:2	8:2
Mean Recall	0,617	0,596	1,000
Std Recall	0,035	0,072	0,000
Ratio	9:1	9:1	9:1
Mean Recall	0,610	0,574	0,723
Std Recall	0,037	0,047	0,232
File	yeast3	yeast3	yeast3
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,768	0,863	0,980
Std Recall	0,080	0,039	0,018
Ratio	2:8	2:8	2:8
Mean Recall	0,798	0,838	0,989
Std Recall	0,111	0,062	0,013
Ratio	3:7	3:7	3:7
Mean Recall	0,764	0,822	0,970
Std Recall	0,088	0,033	0,018
Ratio	4:6	4:6	4:6
Mean Recall	0,788	0,822	0,975
Std Recall	0,102	0,049	0,015
Ratio	5:5	5:5	5:5
Mean Recall	0,742	0,812	0,975
Std Recall	0,059	0,034	0,015
Ratio	6:4	6:4	6:4
Mean Recall	0,702	0,816	0,970
Std Recall	0,088	0,064	0,018
Ratio	7:3	7:3	7:3
Mean Recall	0,771	0,827	0,975
Std Recall	0,094	0,058	0,015
Ratio	8:2	8:2	8:2
Mean Recall	0,705	0,792	0,951
Std Recall	0,107	0,037	0,064
Ratio	9:1	9:1	9:1
Mean Recall	0,734	0,792	0,884
Std Recall	0,097	0,037	0,149

File	yeast4	yeast4	yeast4
Classifier	DecisionTreeClassifier()	KNeighborsClassifier()	GaussianNB()
Ratio	1:9	1:9	1:9
Mean Recall	0,424	0,621	0,943
Std Recall	0,085	0,068	0,079
Ratio	2:8	2:8	2:8
Mean Recall	0,443	0,618	0,943
Std Recall	0,151	0,093	0,079
Ratio	3:7	3:7	3:7
Mean Recall	0,363	0,586	0,904
Std Recall	0,119	0,098	0,064
Ratio	4:6	4:6	4:6
Mean Recall	0,470	0,595	0,899
Std Recall	0,110	0,157	0,095
Ratio	5:5	5:5	5:5
Mean Recall	0,359	0,518	0,899
Std Recall	0,204	0,132	0,095
Ratio	6:4	6:4	6:4
Mean Recall	0,407	0,555	0,899
Std Recall	0,054	0,158	0,095
Ratio	7:3	7:3	7:3
Mean Recall	0,542	0,599	0,884
Std Recall	0,121	0,152	0,083
Ratio	8:2	8:2	8:2
Mean Recall	0,504	0,555	0,884
Std Recall	0,164	0,158	0,083
Ratio	9:1	9:1	9:1
Mean Recall	0,600	0,416	0,610
Std Recall	0,133	0,135	0,335

Tabela 4: Porównanie wpływu łączenie technik samplingu na metrykę Recall.

3.2 Analiza wyników i wnioski

Dla lepszego zobrazowania zmian, uzyskane wyniki metryk Accuracy i Recall, dla jednego konkretnego zbiodu danych umieszczono na wykresie poniżej.



(a) Accuracy (b) Recall

Rysunek 2: Wykres wartości mierzonych metryk Accuracy (a) i Recall (b) dla przykładowego zbioru yeast4.

Wynikiem pierwszej części badania miało być ustalenie, który klasyfikator najlepiej sprawdza się dla każdego z analizowanych zbiorów danych. Dla każdego zbioru danych, najlepsze wyniki uzyskiwał inny model, co wskazuje na brak jednego najlepszego klasyfikatora dla wszystkich analizowanych zbiorów danych.

Zastosowanie technik samplingu miało wyraźny wpływ na analizowane metryki. W większości przypadków prowadziło to do zmniejszenia wartości dokładności (metryka Accuracy) modeli, co wynika z bardziej zrównoważonego reprezentowania klas i mniejszej dominacji klasy większościowej. Jednocześnie techniki samplingu, szczególnie SMOTE i NearMiss, przyczyniły się do zwiększenia wartości czułości (metryka Recall), poprawiając zdolność modeli do prawidłowego identyfikowania przypadków klasy mniejszościowej. Zastosowanie technik samplingu ma kluczowe znaczenie w kontekście klasyfikacji niezbalansowanych zbiorów danych, mimo że może to prowadzić do pewnego spadku dokładności, korzyści w postaci znacznie poprawionej czułości dla klasy mniejszościowej są istotne.

Badanie pokazało, że metryki zależą nie tylko od charakterystyki zbioru danych, ale także od proporcji zastosowanych technik oversamplingu i undersamplingu. Optymalne proporcje mogą różnić się w zależności od specyfiki danych oraz używanego klasyfikatora. Nie istnieje zależność, która pozwala na jednoznaczne stwierdzenie, że konkretna proporcja podziału danych jest najskuteczniejsza dla wszystkich badanych zbiorów.

References

1. R. Mohammed, J. Rawashdeh and M. Abdullah, 2020, Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results, 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2020, pp. 243-248, doi: 10.1109/ICICS49469.2020.239556.
2. Nutthaporn Junsomboon, Tanasanee Phienthrakul. 2017. Combining Over-Sampling and Under-Sampling Techniques for Imbalance Dataset. In Proceedings of the 9th International Conference on Machine Learning and Computing (ICMLC '17). Association for Computing Machinery, New York, NY, USA, 243–247
3. Ronaldo Prati, Gustavo Batista, Maria-Carolina Monard. 2009. Data mining with imbalanced class distributions: Concepts and methods. Paper presented at the IICAI. 359-376
4. Zhaozhao Xu, Derong Shen, Tiezheng Nie, Yue Kou. 2020. A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. Journal of Biomedical Informatics, Volume 107, 103465, ISSN 1532-0464
5. Nitesh Chawla, Kevin Bowyer and Lawrence Hall, W.Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357. 10.1613/jair.953
6. Tuong Le, Minh Thanh Vo, Bay Vo, Mi Young Lee, Sung Wook Baik, 2019, A Hybrid Approach Using Oversampling Technique and Cost-Sensitive Learning for Bankruptcy Prediction, Complexity, vol. 2019, Article ID 8460934, 12 pages, https://doi.org/10.1155/2019/8460934
7. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, 2012,A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 463-484
8. Anas Rulloh Alamsyah, S. Rahma, Nadira Sri Belinda, Adi Setiawan. 2022. SMOTE and Nearmiss Methods for Disease Classification with Unbalanced Data Case Study: IFLS 5. Proceedings of The International Conference on Data Science and Official Statistics. 2021. 10.34123/icdsos.v2021i1.240.
9. KEEL-dataset, https://sci2s.ugr.es/keel/imbalanced.php, last accessed 2024/04/15