Implementacja algorytmu oversamplingu ADASYN

Klaudia Maciaszek - 259704 i Jakub Szuper - 259695

Politechnika Wrocławska

1 Wstęp i przegląd literatury

1.1 Wprowadzenie do tematu projektu, rys teoretyczny

ADASYN [1] - Adaptive Synthetic Sampling - Adaptacyjne syntetyczne próbkowanie

Klasyfikacja z niezbalansowanymi danymi może stanowić problem, ponieważ większość algorytmów używanych do klasyfikacji została zaimplementowana przy założeniu równej liczby przykładów dla każdej klasy. Rozwiązaniem tej komplikacji jest między innymi oversampling. Działanie to polega na sztucznym dodaniu danych, w celu zrównoważenia liczności próbek obu klas.

ADASYN wykorzystuje w swoim działaniu oversampling - idea tej metody polega na generowaniu sztucznych próbek z klasy o mniejszej liczności na podstawie stopnia niezbalansowania, ilości sąsiadów z klasy większej oraz wagi próbki. Algorytm ADASYN poprawia uczenie się, odnosząc się do rozkładu danych zmniejszając błąd wynikający z nierównowagi klas i adaptując się do danych warunków.

Celem projektu jest implementacja algorytmu oversamplingu ADASYN oraz wskazanie i porównanie z innymi metodami referencyjnymi.

1.2 Przegląd literatury

Istnieją także inne metody wykorzystujące oversampling [2].

Jedną z nich jest SMOTE - Synthetic Minority Oversampling Technique. Algorytm ten polega na [3] tworzeniu nowych próbek między istniejącymi na podstawie ich gęstości, które były wprowadzone jako dane wejściowe.

Alernatywną metodą jest ASMOTE [6] - Adaptive Synthetic Minority Oversampling Technique. Ta technika działa z dodatkową adaptacyjnością do rozmiaru mniejszej klasy, co może znacznie poprawiać wydajność modeli, dostarczając syntetyczny zestaw danych do szkolenia.

Kolejną metodą jest BorderlineSMOTE [4]. Opiera się ona na metodzie SMOTE. Wyróżnia się tym, że generowanie próbek odbywa się przy granicach decyzyjnych. W odróżnieniu od SMOTE, BorderlineSMOTE przepróbkowuje lub umacnia granicę wraz z punktami klasy mniejszościowej.

Następną metodą jest SMOTEBoost [5]. Algorytm ten łączy się z algorytmem SMOTE na zasadzie generowania próbek z klasy o mniejszej liczebności oraz procesu klasyfikacji opartego na algorytmie boostingowym. Tworzy on syntetyczne próbki zmieniając aktualizowane wagi oraz wyrównując nieprawidłowe rozkłady.

Z kolei algorytm RamOBoost [2] generuje nowe próbki na podstawie wag, określających trudność w klasyfikacji danego przykładu. Wagi próbek z klasy mniejszości zostają dostosowane zgodnie z ich rozkładem.

Metoda MAHAKIL [7] wykorzystuje pojęcie odległości Mahalanobisa [8], jako miara podobieństwa między dwoma obiektami. Dodatkowo algorytm ten może łączyć się z grupowaniem K-means, które polega na dzieleniu danych wejściowych na określoną liczbę klas.

W projekcie metodą, do której porównywane będą wyniki, będzie SMOTE oraz BorderlineSMOTE.

2 Metoda

Jednym z głównych etapów projektu było zaimplementowanie metody Adasyn, sugerując się przy tym istniejącymi już implementacjami i funkcjami. W języku Python stworzono klasę kompatybilną z danym stosem technologicznym korzystając z BaseEstimator z biblioteki scikit-learn. Wykorzystano wygenerowane dane syntetyczne, w celu wykonania danej implementacji, na którą składały się takie etapy jak znalezienie i stworzenie listy klasy mniejszościowej oraz użycie algorytmu NearestNeighbors, wykorzystując przy tym odopwiednie biblioteki.

Implementacja wstępnego eksperymentu opierała się na uruchomieniu zaimplementowanej metody zgodnie z odpowiednim protokołem eksperymentalnym. Wstępnymi wynikami było wypisanie w konsoli liczby klas mniejszościowej i większościowej oraz Accuracy dla danej metody.

3 Projekt eksperymentów

Celem realizowanych eksperymentów jest porównywanie jakości klasyfikacji zbiorów niezbalansowanych dla wybranych metod.

Do wykonania eksperymentów zostanie użyty język Python 3.10.11, korzystając z IDE PyCharm Community Edition 2022.3.2. Używanymi bibliotekami

będą: scikit-learn, imbalanced-learn oraz NumPy.

Na początku zostanie zaimplementowany algorytm ADASYN, a następnie zostaną zaimportowane algorytmy SMOTE oraz BorderlineSMOTE. Będa wykonane dwa eksperymenty - na danych syntetycznych oraz danych rzeczywistych. Kolejno zostaną obliczone wartości metryk oraz zostaną wykonane testy statystyczne.

Dodatkowo wykonany zostanie sprawdzian krzyżowy k-foldowy z liczbą foldów k=5 (wariant walidacji krzyżowej), który jest standardową procedurą oceny wydajności. Polega on na podzieleniu zbioru danych na zbiór uczący oraz testowy k-razy.

3.1 Metryki oceny wyników

Do oceny wydajności algorytmu zostanie wykorzystane kilka wskaźników oceny [9], takich jak Accuracy, Precision, F1 oraz Recall z biblioteki scikit-learn.

3.2 Testy statystyczne

Do przetestowania normalności rozkładu zostanie wykorzystany Test Shapiro-Wilka jako funkcja shapiro() z biblioteki SciPy.

3.3 Sposób generowania danych syntetycznych

Dane syntetyczne zostaną wygenerowane za pomocą funkcji make_classification z biblioteki scikit.learn. Zostanie wygenerowane 200 próbek. Liczba informatywnych atrybutów będzie wynosiła defaultowo - 2. Tak samo pozostałe parametry są ustawione domyślnie. Parametr 'weights' określa proporcje próbek przypisane do każdej klasy. Aby stworzyć niezbalansowany zbiór danych do tego parametru zostanie przypisane: weights=[0.1, 0.9].

3.4 Sposób pozyskania zbioru danych rzeczywistych

Dane rzeczywiste zostały przedstawione na stronie KEEL - Knowledge Extraction based on Evolutionary Learning [10] oraz udostępnione do pobrania na stronie Kaggle [11]. Został wybrany dataset związany z identyfikacją szkła. Liczba wszystkich przpyadków tego zbioru wynosi 214, Jest to zbiór niezbalansowany: zawiera 13,55% pozytywnych przypadków oraz 86,45% negatywnych przypadków. Atrybutami są pierwiastki chemiczne.

4 Wyniki eksperymentów

4.1 Implementacja oraz przeprowadzenie eksperymentów

Po zaimplementowaniu metody Adasyn, użyto zaimportowanych metod takich jak SMOTE i BorderlineSMOTE oraz gotową metodę ADASYN.

Wszystkie eksperymenty przeprowadzono na danych syntetycznych oraz danych rzeczywistych. Otrzymano wyniki dla każdych z wymienionych metryk (Accuracy, Precision, F1, Recall). Zapewniono powtarzalność wyników, co zrealizowano przy użyciu mechanizmów pseudolosowości opartych na zadanym ziarnie losowaści. W przypadku danych eksperymentów użyto random_state, który został wykorzystany przy walidacji krzyżowej k-foldowej (StratifiedKFold z bilbioteki scikit-learn) oraz przy make_classification. Otrzymano wyniki dla każdej metody i dla każdego kolejnego folda.

4.2 Implementacja kodu analitycznego

Kolejnym krokiem była prezentacja jakości według metryk, co przełożyło się na obliczenie dla każdego folda każdej metody wartości średniej oraz odchylenia standardowego.

Należało także przeprowadzić testy statystyczne, gdzie w przypadku projektu wykorzystano test Shapiro-Wilka (skorzystano z z funkcji shapiro() z biblioteki SciPy).

Końcowo wygenerowano wykresy, potrzebne do wykonania analizy dla każdej z metod.

Literatura

1. Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li, ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, 2008
2. Miss. Mayuri S. Shelke, Dr. Prashant R. Deshmukh, Prof. Vijaya K. Shandilya, A Review on Imbalanced Data Handling Using Undersampling and Oversampling Technique, 2017
3. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, 2002
4. Hui Han1, Wen-Yuan Wang1, and Bing-Huan Mao2, Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning, 2005
5. Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, Kevin Bowyer, SMOTEBoost: Improving Prediction of the Minority Class in Boosting
6. V. Tra, B. -P. Duong and J. -M. Kim, "Improving diagnostic performance of a power transformer using an adaptive over-sampling method for imbalanced data," in IEEE Transactions on Dielectrics and Electrical Insulation, vol. 26, no. 4, pp. 1325-1333, Aug. 2019, doi: 10.1109/TDEI.2019.008034.

7. Y. Zhang, T. Zuo, L. Fang, J. Li and Z. Xing, "An Improved MAHAKIL Oversampling Method for Imbalanced Dataset Classification," in IEEE Access, vol. 9, pp. 16030-16040, 2021, doi: 10.1109/ACCESS.2020.3047741.
8. Zygmunt Kaczmarek, Stanisław Czajka, Elżbieta Adamska, Propozycja metody grupowania obiektów jednoi wielocechowych z zastosowaniem odległości Mahalanobisa i analizy skupień
9. Rand Kouatly, Ietezaz Ul Hassan, Raja Hashim Ali, Zain Ul Abideen, Talha Ali Khan, Significance of Machine Learning for Detection of Malicious Websites on an Unbalanced Dataset
10. https://sci2s.ugr.es/keel/dataset.php?cod=143fbclid=IwAR3GC6uS1LGu9Wn2_-u7658PBNA1uD5ZbwSKPPCBYlusAU8T8qsQwMRzePYsub1
11. https://www.kaggle.com/datasets/baguspurnama/glass-imbalanced?resource=downloadfbclid=IwAR1-JgWLgFk64MhlOj-1TKF1rXldHU8AtxF0_QerPuOAMqyZhJlWpRAtDCQ