Projekt ZUM - dokumentacja wstępna

Michał Bartnicki, Michał Piotrak

Temat zadania: **Analityczny - detekcja anomalii (kod - DAz11)**
Wybrany zbiór danych: **Credit Card Fraud Detection (Kaggle)**

Interpretacja tematu projektu

Realizowany projekt będzie miał charakter szczegółowego przeglądu rozwiązań problemu detekcji anomalii. Jest to obszar analizy danych, którego celem jest identyfikowanie obserwacji odbiegających znacznie od oczekiwanego wzorca. Takie podejście jest wymagane w m.in. obszarze ujawniania wszelkiego rodzaju oszustw, w tym przypadku dotyczących transakcji dokonanych za pomocą kart kredytowych.
Do rozwiązania problemu detekcji anomalii można wykorzystać różne metody, które możemy podzielić na dwie kategorie:

- metody nadzorowane - wykorzystują dane etykietowane, gdzie możemy wyróżnić podział na przykłady standardowe oraz anomalie.
- metody nienadzorowane - identyfikują anomalie na podstawie samej struktury zbioru danych, odległości między obserwacjami lub gęstości rozkładu danych. Te metody są użyteczne, gdy nie posiadamy pełnej wiedzy o występujących anomaliach, czyli w zbiorach bez etykiet.

Przeprowadzone badania w ramach projektu będą dotyczyły zarówno podejścia nadzorowanego, jak i nienadzorowanego. Dogłębna analiza tematu będzie wymagała również poruszenia aspektu niezrównoważenia klas, który wpływa na skuteczność klasycznych metod klasyfikacji. Poza tym, trzeba będzie przeprowadzić rozważania dotyczące wyboru miar oceny jakości modeli do przypadku silnie niezbalansowanych klas.

Lista algorytmów, które będą wykorzystane w eksperymentach

Metody nadzorowane:

Metoda	Klasa	Biblioteka
SVM z wagami klas	sklearn.svm.SVC* * w razie problemów z wydajnością i w zależności od rozkładu anomalii - sklearn.svm.LinearSVC	scikit-learn
Drzewo decyzyjne z wagami klas	sklearn.tree.DecisionTreeClassifier	scikit-learn
Las losowy	sklearn.ensemble.RandomForestClassifier	scikit-learn
XGBoost	xgboost.XGBClassifier	XGBoost

Metody nienadzorowane:

Metoda	Klasa	Biblioteka
OC - SVM	sklearn.svm.OneClassSVM	scikit-learn
Las izolacyjny	sklearn.ensemble.IsolationForest	scikit-learn
Lokalny czynnik odstający	sklearn.neighbors.LocalOutlierFactor	scikit-learn
DBSCAN	sklearn.cluster.DBSCAN	scikit-learn

Pozostałe algorytmy, które planujemy zastosować z krótkim opisem celu:

Metoda	Klasa	Biblioteka	Cel
Oversampling (SMOTE)	imblearn.over_sampling.SMOTE	Imbalanced-learn	zbalansowanie klas

Undersampling	imblearn.under_sampling.RandomUnderSampler	Imbalanced-learn	zbalansowanie klas
Walidacja krzyżowa	sklearn.model_selection.StratifiedKFold	scikit-learn	procedury walidacyjne

Plan badań

Cel eksperymentów

Nasza praca ma na celu porównanie skuteczności różnych algorytmów klasyfikacji w przypadku dwuklasowym, działając na silnie niezbalansowanym zbiorze danych. Ponadto dla powyższych rozwiązań zweryfikujemy pozytywny wpływ metod zapewniających zbalansowanie klas, gdzie w szczególności znaczenie to powinno mieć dla podejścia nadzorowanego. Chcemy także potwierdzić tezę, że algorytmy nadzorowane gwarantują wyższą skuteczność detekcji znanych wzorców anomalii niż algorytmy nienadzorowane, które z kolei powinny być lepsze w przypadkach gdy charakter anomalii jest zmienny.

Charakterystyka wybranego zbioru danych

Wybrany zbiór zawiera zanonimizowane informacje o transakcjach dokonanych za pomocą kart kredytowych przez europejczyków we wrześniu 2013 roku przez okres dwóch dni. Liczba transakcji: 284807, w tym 492 oznaczone jako fałszywe.
Transakcje są opisane przy pomocy następujących cech:

Cecha	Opis
Wektor cech [V1, …, V28]	Cechy transakcji po transformacji PCA
Time	Upływ czasu w sekundach pomiędzy pierwszą a określoną transakcją (bez transformacji PCA)
Amount	Kwota transakcji (bez transformacji PCA)
Class	Klasa transakcji, wartość ”1” - transakcja fałszywa (oszustwo), wartość ”0” - transakcja uczciwa

Opisywany zbiór jest wysoce niezbalansowany - przykłady określające transakcje fałszywe (które uznajemy za egzemplarze klasy reprezentującej przypadki nieprawidłowe) stanowią jedynie 0,172% wszystkich transakcji.

Z tego powodu modele uzyskane poprzez użycie metod nadzorowanych są narażone na zjawisko nadmiernego dopasowania. Żeby temu zapobiec użyjemy odpowiednich technik balansujących klasy, do których zaliczamy:

- undersampling - ograniczenie liczby przykładów klasy większościowej,
- oversampling - zwiększenie liczby przykładów klasy mniejszościowej, które możemy uzyskać poprzez zastosowanie algorytmu **SMOTE**, który generuje nowe próbki pomiędzy istniejącymi obserwacjami na podstawie ich lokalnej gęstości.

Poza tym, w celu przygotowania danych planujemy zrealizować następujące czynności:

- sprawdzić, czy mamy w następującym zbiorze jakiekolwiek braki danych i duplikaty,
- dokonać skalowania cech *Time* oraz *Amount* (w wyniku zastosowania transformacji PCA, wektor cech *[V1, …, v28]* powinien być uprzednio odpowiednio wyskalowany),
- pomimo zastosowania już transformacji PCA na części parametrów, zbadamy jeszcze korelacje pomiędzy nimi, w celu potencjalnej redukcji wymiarowości.

Parametry algorytmów, których wpływ będziemy badać

Dla algorytmów nadzorowanych zbadamy wpływ przekształceń typu undersampling i oversampling. Przetestujemy także różne wartości parametru *sampling_strategy,* czyli oczekiwany stosunek ilości transakcji klasy prawdziwej i fałszywej uzyskany po zastosowaniu resamplingu.

Ponadto, będziemy eksperymentować z różnymi parametrami samych algorytmów, wybór poniższych parametrów może ulec zmianie, wraz z postępem pracy i lepszym poznaniem oraz zrozumieniem działania badanych algorytmów.

Metoda	Parametry
SVM z wagami klas	*kernel* ● *linear*: może skutkować szybszym treningiem, ponieważ opiera się na liniowym rozdzieleniu klas ● *RBF* *C*, *class_weight* (*None* vs *balanced*)
Drzewo decyzyjne z wagami klas	*max_depth* (ustawić, aby zapobiec przerośnięciu drzewa), *class_weight* (*None* vs *balanced*)
Las losowy	*n_estimators* (ilość drzew, poszukujemy największej wartości z rozsądnym czasem treningu), *max_depth*, *class_weight*
XGBoost	*n_estimators*, *scale_pos_weight*, *learning_rate*, *max_depth*, *early_stopping_rounds* (może zapobiec przetrenowaniu)
OC - SVM	*kernel*, *nu* (większa wartość może skutkować większą ilością fałszywych alarmów, ale przetestujemy wartości inne niż domyślna 0.5), *gamma*
Las izolacyjny	*n_estimators*, *contamination*
Lokalny czynnik odstający	*n_neighbours*, *contamination*
DBSCAN	*eps*, *min_samples*

Planowane do wykorzystania procedury oceny modeli oraz miary jakości

Podstawą procedury oceny jakości modeli będzie k - krotna walidacja krzyżowa. Ze względu na silnie niezbalansowany zbiór danych zastosujemy technikę **Stratified k - fold**, która zapewnia, że proporcje klas w zbiorze treningowym i testowym są podobne do tych w całym zbiorze danych. Ze względu na to, że liczba anomalii jest mała, wartość parametru k nie powinna być duża i według naszej początkowej oceny optymalna jej wartość powinna wynosić co najwyżej 5.

Dokładność nie jest odpowiednią miarą jakości dla detekcji anomalii, ponieważ jej wysoka wartość może zostać osiągnięta poprzez prawidłowe klasyfikowanie przykładów klasy większościowej. Za główną miarę w tym zadaniu uznamy **odzysk**. Powie nam ona jak wiele, spośród wszystkich oszustw, potrafi wykryć model. Jest to ważne, aby jak najwięcej nielegalnych transakcji zostało odkrytych, nawet kosztem większej liczby transakcji fałszywie pozytywnych - które będą wiązać się z mniejszym kosztem - np. dodatkową pracą człowieka obsługującego alert o prawdopodobnym oszustwie, lub dodatkowym zatwierdzeniem transakcji przez użytkownika.
Jako miary pomocnicze zostaną użyte:

- **Precyzja -** określi, ile transakcji oznaczonych jako oszustwo jest faktycznie oszustwem. Ta miara jest mniej ważna w naszym przypadku, ale należy ją wziąć pod uwagę, aby nie obciążyć nadmiernie użytkownika / administratora fałszywymi alarmami.
- **Miara F** - powie o tym, jak zrównoważone względem siebie są precyzja i odzysk oraz pozwoli uniknąć skrajności modelu.
- **Krzywa Precision - Recall** - do ustalenia kompromisu między precyzją a odzyskiem.
- **Pole pod krzywą PR (Area Under the PR Curve, w skrócie AUPRC)** - wskaźnik opisujący krzywą PR, ogólnie im wyższa wartość tym lepsza jakość modelu.