Zaawansowane metody uczenia maszynowego – projekt nr 1

Jadwiga Słowik

3 maja 2019

1 Cel projektu

Celem projektu jest oczyszczenie danych pochodzących z branży telekomunikacyjnej, a następnie na ich podstawie zbudowanie i przetestowanie czterech wybranych klasyfikatorów binarnych.

2 Opis zbioru danych

Dostarczony zbiór danych składa się z dwóch części: zbiór treningowy (wraz z etykietami) oraz zbiór testowy (bez etykiet). Nazwy zmiennych (cech/kolumn) mają jedynie numery porządkowe, nie niosą żadnej dodatkowej informacji o opisywanej cesze.

Pojedyncza obserwacja opisuje cechy konkretnego klienta (branży telekomunikacyjnej), natomiast klasa (etykieta) określa, czy klient przyjął proponowaną ofertę (wartość 1), czy nie (wartość 0).

Rozkład klas jest *niezbalansowany*. Liczność obserwacji o etykiecie 1 stanowi mniej niż 10% obserwacji.

3 Opis rozwiązania

Rozwiązanie niniejszego problemu zostało zaimplementowane przy pomocy języka *Python* i bibliotek: *numpy*, *pandas*, *sklearn*, *categorical_encoders*.

Analiza danych i implementacja rozwiązania znajduje się w `jupyter-notebooku` o nazwie `ml-project1-notebook.ipynb`.

Rozwiązanie składa się z dwóch głównych etapów, które kolejno zostaną opisane w kolejnych podrozdziałach.

3.1 Proces oczyszczania danych

Dostarczone dane zostały poddane następującym transformacjom:

1. Usuwanie wybranych kolumn
   Zostały usunięte kolumny, które zawierały więcej niż 30% braków danych. Ponadto, dla każdej pary kolumn został obliczony współczynnik korelacji (*Pearsona*). Następnie, na podstawie otrzymanych wyników, dla każdej kolumny została utworzona lista (innych) kolumn, które są skorelowane z daną kolumną w stopniu

większym niż 0.8. Na podstawie tak utworzonych list, zostały usunięte zbędne kolumny.

Następnie, dla każdej zmiennej (cechy) zostały wygenerowane histogramy. Dzięki temu, można było zauważyć, jak wygląda rozkład wartości dla każdej zmiennej. Zauważyłam, że w znacznej grupie wygenerowanych histogramów 0 przewyższa w liczności inne wartości. Bardzo możliwe, że jest to inny sposób oznaczenia braku danych, jednakże wcale nie musi to być prawda. Na przykład, jedna z kolumn mogłaby oznaczać aktualny bilans rachunku danego klienta (wartości ujemne – zadłużenie, wartości dodatnie – nadpłata).

2. Zastąpienie braków danych
   Braki danych dla danych ilościowych (w tym przypadku typu `float64`) zostały zastąpione wartością średnią w danej kolumnie, natomiast braki danych dla danych jakościowych (tym przypadku typu `object` i `int`) zostały zamienione przez wartość najczęściej występującą w obrębie danej kolumny (moda).

3. Kodowanie danych kategorycznych
   Niektóre algorytmy klasyfikacji mają problemy z obsługą zmiennych nieliczbowych. W tym celu, zmienne typu łańcuchowego zostały zakodowane przy pomocy algorytmu *factor / mean encoding*.

3.2 Proces klasyfikacji

Na przetworzone danych zostały zbudowane cztery następujące klasyfikatory:

- *Random forest*
- *XGBoost*
- *Naive Bayes*
- *QDA*
- Próba *SVM*
  Proces *kroswalidacji* został przerwany po ponad dwóch godzinach.

Ostateczny wybór „najlepszego" klasyfikatora został podjęty na podstawie wyników *kroswalidacji*. Dane treningowe zostały podzielone na 10 części i dla kolejno każdej części został wykonywany proces predykcji. Dla każdego etapu (*kroswalidacji*) były obliczane następujące metryki: *f1*, *precision*, *recall*. Warto zaznaczyć, że została pominięta metryka *accuracy*, z powodu faktu, że dany zbiór treningowy jest bardzo niezbalansowany.

Ostatecznie, został wybrany klasyfikator *XGBoost*, ponieważ daje największą czułość wraz z precyzją na wysokim poziomie. Dobre wyniki (ale nieco gorsze) zostały osiągnięte przez klasyfikator *Random forest*. Pozostałe dwa klasyfikatory (*QDA*, *Naive Bayes*) dawały znacznie gorsze wyniki. Dokładne tabelki z wynikami zostały zawarte w pliku `ml_project1_notebook.ipynb`.