Neuronowy model dawcy krwi

KWD

Mikołaj Baran, Łukasz Czuba

13.01.2022

Streszczenie

Dane pobrane z Centrum Transfuzji Krwi w Hsin-Chu w Tajwanie przekazane do publicznego dostępu 3 października 2008 roku opisują losowych dawców krwi z tego centrum poprzez 4 cechy. Problemem, jaki jest postawiony, jest problem klasyfikacji binarnej: rozstrzygnięcie na podstawie tych danych, czy dawca ten oddał krew w marcu 2007 roku w przypadku szczególnym, a w przypadku ogólnym predykcja, czy dawca przyjdzie oddać krew w zadanym terminie.

Spis treści

Rozdział 1

Wprowadzenie

3 października 2008 roku prof. I-Cheng Yeh z Departamentu Zarządzania Informacją Uniwersytetu Chung-Hua upublicznił dane uzyskane z Centrum Transfuzji Krwi w Hsin-Chu, łącząc je z etykietami w postaci pojedynczej klasy binarnej, opisującej czy dawca określony tymi danymi oddał krew w marcu 2007 roku, czy też nie, tym samym sprowadzając ten zbiór danych modelu RFM (Recency, Frequency and Monetary Value) używanego często w marketingu do identyfikacji najlepszych klientów. Stąd też dane te mogą posłużyć do rozwiązania problemu klasyfikacji binarnej.
Projekt ten ma na celu znalezienie rozwiązania tego problemu, kreując odpowiedni model wykorzystując analizę danych wejściowych oraz poznane narzędzia matematyczne i statystyczne, który następnie zostanie poddany ewaluacji i walidacji.

1.1 Opis problemu

Dane zgromadzone w zbiorze o nazwie *Blood Transfusion Service Center Data Set* zawierają 5 cech:

1. R (Recency - niedawność) - liczba miesięcy od ostatniej donacji dawcy
2. F (Frequency - częstotliwość) - suma wszystkich donacji dawcy
3. M (Monetary - wartość) - suma centymetrów sześciennych (mililitrów) krwi oddanej przez dawcę
4. T (Time - czas) - liczba miesięcy od pierwszej donacji dawcy
5. binarna zmienna reprezentująca, czy dawca oddał krew w marcu 2007 (wartość 1), czy też nie (wartość 0).

Zbiór danych zawiera 748 rekordów. Analiza struktury danych wejściowych wykazała, że są one w pełni kompletne, to znaczy zbiór nie zawiera żadnych

brakujących wartości. Wykazano również, że wszystkie dane, zgodnie z opisem, są typu całkowitoliczbowego (*int64*). Oznacza to, że nie wymagają one translacji na dane liczbowe.
Analiza zawartości danych wejściowych wykazała, że zarówno zakres wartości, jak i średnia wartość oraz odchylenie standardowe cechy *Monetary* są silnie większe od pozostałych - w przypadku średniej rząd tysięcy w porównaniu do rzędu dziesiątek u pozostałych, w przypadku wartości maksymalnej rząd dziesiątek tysięcy w porównaniu do rzędu dziesiątek u pozostałych cech. To wskazuje, że potencjalnie ta cecha mogłaby mieć zbyt duży wpływ na predykcje modelu, co czyni ją potencjalnym kandydatem do poddania normalizacji. Ponadto, zauważono, że zbiór danych jest nieco niezbalansowany ze względu na dość znaczącą przewagę etykiet o wartości 0 (76,2032%) w porównaniu do 1 (23,7968%). Jest to jednak w zupełności akceptowalne. Zakładamy bowiem, że odzwierciedla to realny stan rzeczy. Aby trenowany model trzymał się tego założenia, na etapie podziału zbioru danych na zbiór treningowy i testowy będziemy zwracać uwagę, by te proporcje zostały zachowane.
Podczas wizualizacji danych wejściowych zauważono silną korelację dodatnią cech *Monetary* i *Frequency*, co nie dziwi, gdyż oczywistym jest wniosek, że wraz ze wzrastającą liczbą donacji zwiększa się suma objętości oddanej krwi. Oznacza to, że jedną z tych cech można uznać za redundantną i usunąć ją ze zbioru danych. W celach badawczych w projekcie wykorzystane zostaną dwa zbiory: pełny oraz pozbawiony cechy *Monetary*, które ostatecznie zostaną porównane na etapie ewaluacji.

Rozdział 2

Opis metody

2.1 Wprowadzenie teoretyczne

Segmentacja RFM to metoda analizy danych wykorzystywana w marketingu polegająca na wartościowaniu klientów firmy na podstawie ich wcześniejszych interakcji z firmą. Analiza ta pozwala przewidywać zachowanie konkretnego klienta oraz opracować odpowiednią strategię marketingową wycelowaną w daną grupę klientów w celu zwiększenia szansy powodzenia reklamy, czyli innymi słowy skutecznego przekonania klienta do skorzystania z usług danej firmy. Skrót RFM (Recency, Frequency and Monetary Value) oznacza odpowiednio: okres, jaki upłynął od ostatniej interakcji klienta z firmą, częstotliwość, z jaką dany klient dokonuje interakcji oraz wartość danego klienta dla danej firmy.
Do stworzenia modelu rozwiązującego problem klasyfikacji binarnej doskonale pasuje narzędzie **regresji logistycznej**, czyli szczególny przypadek uogólnionego modelu liniowego. Charakteryzuje się ona tym, że zmienna zależna jest typu binarnego, a więc przyjmuje wyłącznie wartości 0 lub 1. Wartość ta jest jednak określana poprzez wyrażenie prawdopodobieństwa przydzielenia wartości 0 lub 1, co przekłada się na wyznaczenie prawdopodobieństwa wystąpienia zdarzenia opisanego przez zmienną zależną (tzw. prawdopodobieństwo sukcesu). Zastosowanie transformacji logit pozwala na linearyzację modelu regresji logistycznej i przedstawienie go w postaci regresji liniowej. W celu oszacowania modelu regresji logistycznej wykorzystuje się metodę największej wiarygodności (*maximum likelihood*) - poprzez obliczenia poszukiwane są takie wartości współczynników zmiennych niezależnych (predyktorów) wprowadzonych do modelu, że wiarygodność jest jak największa. W analizie regresji logistycznej określamy, czy dana zmienna niezależna wprowadzona do modelu ma wpływ na zmienną zależną, a więc czy jest w tym modelu istotna statystycznie. Ponadto, możemy określić parametr *exp(B)* (tzw. iloraz szans) dla danego predyktora, który określa, czy i w jakim stopniu (ile razy) wzrost wartości predyktora powoduje spadek lub

wzrost szansy wystąpienia analizowanego zdarzenia w stosunku do poziomu referencyjnego (niewystąpienie zdarzenia kodowane wartością 0).
Wzór na **równanie modelu regresji logistycznej**, w którym zmienna zależna przyjmuje dwie binarne wartości:

$$\mathbf{P(Y = 1|x_1, x_2, ..., x_k) = \frac{e_0^a + \sum_{i=1}^{k} a_i x_i}{1 + e^{a_0 + \sum_{i=1}^{k} a_i x_i}}}$$

gdzie:
$x_1, x_2, ..., x_k$ - zmienne niezależne
$P(Y = 1|x_1, x_2, ..., x_k)$ - warunkowe prawdopodobieństwo, że zmienna zależna Y przyjmie wartość równą 1 dla wartości zmiennych niezależnych $x_1, x_2, ..., x_k$
e - liczba Eulera 2,718
a_0 - stała
$a_1, a_2, ..., a_k$ - współczynniki regresji dla poszczególnych zmiennych niezależnych
Założenia, które muszą być spełnione dla modelu regresji logistycznej:

- zmienne objaśniające są ze sobą nieskorelowane
- logit prawdopodobieństwa zależy w sposób liniowy od zmiennych objaśniających.

Na poprawność i dokładność wnioskowania wpływa odpowiedni dobór zmiennych objaśniających - uwzględnione powinny być wyłącznie te, które wykazują istotny wpływ na zmienną zależną. Zarówno pominięcie wpływowej zmiennej niezależnej, jak i zbyt mała liczba obserwacji powodują spadek jakości analizy.

2.2 Badania symulacyjne

Dla omawianego zbioru danych stosunek wystąpienia zdarzenia (wartość 0) do niewystąpienia tego zdarzenia (wartość 1) wynosi 178:570, co oznacza, że bez względu na wszystko przypisując wartość 0 predykowanej krotce uzyskalibyśmy skuteczność modelu na poziomie 76,2032%. Analiza eksploracyjna danych wejściowych oraz wykorzystanie regresji logistycznej pozwoliło uzyskać model charakteryzujący się **skutecznością 75,4011%**. Zbiór danych został podzielony na podzbiór uczący i podzbiór testowy, przy czym podzbiór testowy zawiera 25% wszystkich danych. Macierz konfuzji wytrenowanego modelu dla zbioru danych testowych przedstawia się następująco:

$$\begin{bmatrix} 135 & 7 \\ 39 & 6 \end{bmatrix}$$

Oznacza to, że w 187 próbach predykcji model przewidział 135 razy poprawnie wartość 0 oraz 7-krotnie przewidział błędnie wartość 0, natomiast 6 razy poprawnie przewidział wartość 1 oraz 39-krotnie błędnie wskazał wartość 1 oznaczającą wystąpienie zdarzenia oddania krwi przez dawcę w marcu 2007 roku.
Model ten ma skuteczność niższą od bezwzględnego przydzielania zawsze wartości 0, co wskazuje, że zdecydowanie wymaga on optymalizacji wykorzystanych do jego budowy parametrów. Wykorzystane w tym modelu parametry były parametrami domyślnymi, a więc jako *solver* został wybrany algorytm 'lbfgs', a maksymalna liczba iteracji została ustawiona na 100. W celu znalezienia najlepszych parametrów wykorzystamy narzędzie kroswalidacyjne *GridSearchCV*. Jako dane wejściowe do funkcji GridSearchCV zostały wykorzystane parametry: pełna lista solverów dostępnych dla modelu regresji logistycznej ('newton-cg', 'lbfgs', 'liblinear','sag', 'saga'), jako maksymalna liczba iteracji liczby z zakresu $< 50; 1000 >$ z krokiem 50, natomiast jako metodę podziału dla kroswalidacji *cv* podano liczbę 10. Podanie liczby dla parametru cv dla estymatora będącego klasyfikatorem binarnym sprawia, że jako metodę podziału wykorzystuje się *StratifiedKFold* z liczbą podziałów wskazaną w parametrze.
Znalezione przez narzędzie GridSearchCV najlepsze z podanych parametrów to algorytm ***sag*** jako solver oraz wartość maksymalnej liczby iteracji jako **950**. Należy teraz zbudować model ze znalezionymi parametrami, wytrenować go i ewaluować poprzez narzędzia *accuracy_score* i macierz konfuzji, po czym porównać uzyskane wyniki z pierwotną wersją modelu. Okazuje się, że tak sparametryzowany model faktycznie uzyskuje znacząco lepsze wyniki, bowiem jego skuteczność została oceniona na poziom **80,7487%**, czyli o ponad 5% lepiej niż model z ustawieniami domyślnymi! Macierz konfuzji dla nowego modelu prezentuje się następująco:

$$\begin{bmatrix} 138 & 4 \\ 32 & 13 \end{bmatrix}$$

Oznacza to, że w 187 próbach predykcji model przewidział 138 razy poprawnie wartość 0 oraz 4-krotnie przewidział błędnie wartość 0, natomiast 13 razy poprawnie przewidział wartość 1 oraz 32-krotnie błędnie wskazał wartość 1 oznaczającą wystąpienie zdarzenia oddania krwi przez dawcę w marcu 2007 roku. Jest to zdecydowanie bardziej satysfakcjonujący rezultat. Można zauważyć wyraźną przewagę błędów po stronie odpowiedniej klasyfikacji sukcesu. Jak jednak wskazywaliśmy podczas analizy danych wejściowych, taki wynik nie może dziwić ze względu na silną przewagę etykiet o wartości 0 nad wartością 1 w całym zbiorze danych (76%). Można wyciągnąć z tego wniosek, że model ten częściej przewidzi skorzystanie klienta z usług firmy w przypadku, gdy w rzeczywistości z niej nie skorzysta niż sytuacji odwrotnej, co z punktu widzenia marketingu jest korzystne, gdyż

lepiej dotrzeć do klienta, który nie zdecyduje się ostatecznie na skorzystanie z usług firmy, niż stracić potencjalnego klienta poprzez pominięcie go w kampanii marketingowej.

Rozdział 3

Podsumowanie

Model regresji logistycznej jest odpowiednim narzędziem do predykcji zdarzeń w oparciu o dane w modelu RFM ze względu na swoją charakterystykę. Dane zawierające cechy RFM pozwalają na predykcję interakcji klienta z firmą, co jest powszechnie stosowane przez firmy w celach marketingowych. Odpowiednio przeprowadzona inżynieria cech oraz dalsza analiza eksploracyjna i wizualizacja pozwalają na odpowiednie przygotowanie ich przed podaniem modelom uczenia maszynowego. Analiza jakości uzyskanego modelu z wykorzystaniem stosownych metryk jest niezbędna, by ocenić przydatność modelu oraz konieczność czy opłacalność dalszej pracy nad jego rozwojem. Jeżeli rozwiązanie nie jest zadowalające, warto skorzystać z metod kroswalidacyjnych jak m.in. wykorzystana w tym projekcie metoda GridSearch, aby znaleźć parametry pozwalające zbudować skuteczniejszy model. Na koniec warto zadać sobie pytanie, czy uzyskany ostatecznie klasyfikator o skuteczności 81% jest wystarczający? Ocena pożyteczności modelu zależy od kontekstu jego wykorzystania. Gdy mamy np. do czynienia z branżą pojazdów autonomicznych taka skuteczność jest niedopuszczalna. W przypadku planowania działań marketingowych (co jest celem tego projektu) skuteczność taka może być uznana za zadowalającą, a klasyfikator ten może realnie wpłynąć na zwiększenie potencjalnych przychodów firmy.

Dodatek A

Kod programu

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

from numpy import mean
from numpy import std
from scipy.stats import sem

from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV

#wczytanie danych
transfusion_data = pd.read_csv("transfusion.data")

#analiza struktury danych
transfusion_data.info()

#analiza wartości danych
transfusion_data.describe()
sb.pairplot(transfusion_data, diag_kind="kde")

transfusion_data.rename(
    columns={'Recency (months)': 'recency', 'Frequency (times)': 'frequency',
```

```
    'Monetary (c.c. blood)': 'monetary', 'Time (months)': 'time',
    'whether he/she donated blood in March 2007': 'label'},
    inplace=True
)

#analiza balansu danych - stosunku poszczególnych klas w całym zbiorze
transfusion_data.label.value_counts(normalize=True)

#podział na zbiór pełny oraz pozbawiony cechy Monetary silnie dodatnio skorelowanej
#z cechą Frequency
transfusion_data_reduced = transfusion_data
transfusion_data_reduced.drop(columns=['monetary'])

X = transfusion_data.copy().drop(columns=['label'])
Xr = transfusion_data_reduced.copy().drop(columns=['label'])

Y = transfusion_data.label
Yr = transfusion_data_reduced.label

#podział danych na zbiory testowe i treningowe
X_train, X_test, y_train, y_test = train_test_split(
    X,
    Y,
    test_size=0.25,
    random_state=17,
    stratify=Y
)

Xr_train, Xr_test, yr_train, yr_test = train_test_split(
    Xr,
    Yr,
    test_size=0.25,
    random_state=17,
    stratify=Yr
)

#trening modelu
logistic_regression = LogisticRegression()

logistic_regression.fit(X_train, y_train)

acc = accuracy_score(y_test, logistic_regression.predict(X_test))
print("Model accuracy is {0:0.6f}".format(acc))
```

```
conf_matrix = confusion_matrix(y_test, logistic_regression.predict(X_test))
print(conf_matrix)

logistic_regression_r = LogisticRegression()

logistic_regression_r.fit(Xr_train, yr_train)

acc = accuracy_score(yr_test, logistic_regression_r.predict(Xr_test))
print("Model accuracy is {0:0.6f}".format(acc))

conf_matrix = confusion_matrix(yr_test, logistic_regression_r.predict(Xr_test))
print(conf_matrix)

#porównanie z modelem RandomForestClassifier

random_forest = RandomForestClassifier(n_estimators=700, random_state=101)

random_forest.fit(Xr_train, yr_train)

acc = accuracy_score(yr_test, random_forest.predict(Xr_test))
print("Model accuracy is {0:0.6f}".format(acc))

#GridSearch w celu znalezienia optymalnych parametrów dla modelu

max_iter = np.arange(50,1001, 50).tolist()
parameters=[{'solver':['newton-cg', 'lbfgs', 'liblinear','sag', 'saga'],
            'max_iter':max_iter}]
model = GridSearchCV(LogisticRegression(), parameters, cv=10)
model.fit(Xr,Yr)
print(model.best_params_)
model.best_estimator_

best_linear_regression = LogisticRegression(max_iter=950, solver='sag')

best_linear_regression.fit(Xr_train, yr_train)

acc = accuracy_score(yr_test, best_linear_regression.predict(Xr_test))
print("Model accuracy is {0:0.6f}".format(acc))

conf_matrix = confusion_matrix(yr_test, best_linear_regression.predict(Xr_test))
print(conf_matrix)
```