Bank Credit Evaluation

Basia Seweryn, Michał Wietecki

24 kwietnia 2024

Streszczenie

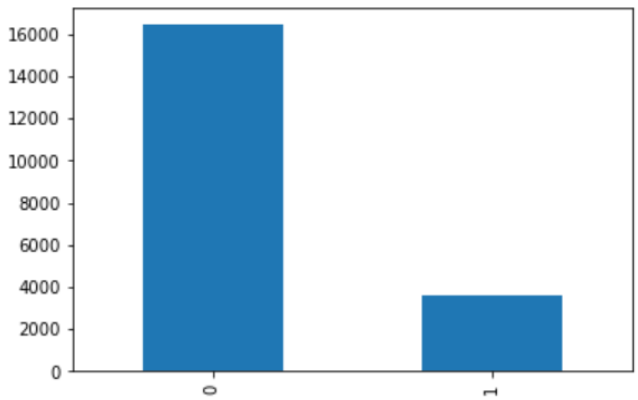
1 Dane

Nasze dane dotyczyły kredytów bankowych oraz tego czy zostały one spłacone bezproblemowo czy też nie. Dane zawierały 15 kolumn i około 20 000 rekordów. Kolumny oraz ich znaczenie zostały przedstawione poniżej:

'id': Identification number.
'TARGET': Binary variable indicating if the individual had difficulty in repaying the credit (1 for difficulty, 0 for no difficulty).
'CNT_CHILDREN': Number of children the individual has.
'AMT_INCOME_TOTAL': Total income of the individual.
'AMT_CREDIT': Credit amount requested by the individual.
'AMT_ANNUITY': Annuity of the loan.
'AMT_GOODS_PRICE': Price of the goods for which the loan is given.
'REGION_POPULATION_RELATIVE': Relative population of the region.
'DAYS_BIRTH': Age of the individual in days (negative value).
'DAYS_EMPLOYED': Number of days the individual has been employed (negative value).
'DAYS_REGISTRATION': Number of days the individual's registration was made relative to the current application.
'DAYS_ID_PUBLISH': Number of days since the individual published their ID.
'FLAG_WORK_PHONE': Binary flag indicating if the individual has a work phone (1 for yes, 0 for no).
'REGION_RATING_CLIENT': Region rating of the client.
'HOUR_APPR_PROCESS_START': Hour of the day when the loan application process started.

Rysunek 1: Informacja o danych

Wszystkie dane w zbiorze są numeryczne. Zmienna celu "Target"to kolumna binarna. Wartość 1 dla osób, które miały problem ze spłaceniem kredytu, 0 dla spłat bezproblemowych. Główny problem z danymi to ogromna różnica w liczności rekordów klasy 1 i klasy 0. Pzzedstawiono to za histogramie poniżej: Celem naszego modelu było przewidzenie wartości zmiennej Target.

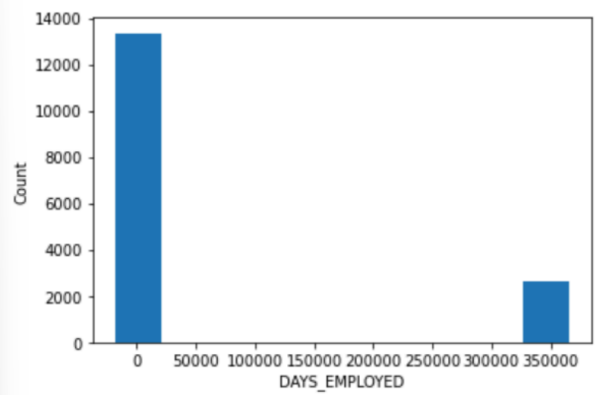


Rysunek 2: Histogram zmiennej Target

2 Preprocessing danych

2.1 Outliery i błędy systemowe

W zbiorze danych wszystkie kolumny były numeryczne oraz praktycznie niebrakowane, dlatego nie musieliśmy ich przekształcać. Danych wybrakowanych pojawiło się niewiele, rzędu kilkunastu, dlatego zostały one usunięte ze zbioru.
Podczas zapoznawania się z danymi, tworzenia wstępnych wykresów skrzypcowych etc. zauważyliśmy, że w wielu kolumnach pojawiają się bardzo wyodrębnione wartości, tzw. outliery (przykładowo zarobki roczne rzędu 100 milionów). Wartości te bardzo zaburzały naszą wstępną analizę oraz mogłyby przeszkadzać w dalszej pracy z modelami, dlatego część rekordów z wartościami ekstremalnymi została odrzucona.
Dodatkowo w kolumnie DAYS EMPLOYED zauważyliśmy ogromną nieprawidłowość, prawdopodobnie pewien błąd systemowy. Dla kilku tysięcy rekordów jako liczba dni zatrudnienia klienta została wpisana wartość 365243, co odpowiada około 1000 lat. Jest to ewidentnie błąd popełniony na przykład przy wprowadzaniu danych bądź eksploatacji danych. Rekordy te ostatecznie uznaliśmy za osoby niezatrudnione, a w miejsce błędnej wartości wpisaliśmy 0.



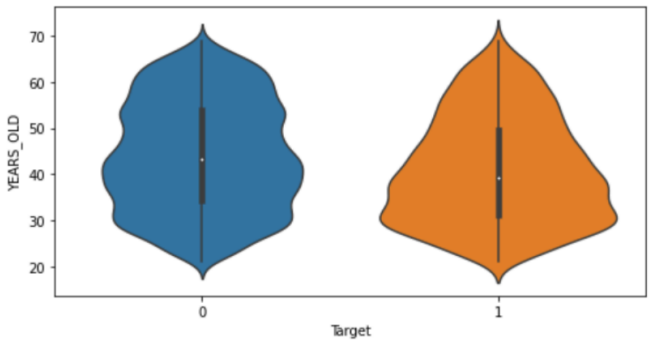
Rysunek 3: Histogram dla kolumny DAYS EMPLOYED

2.2 Normalizacja

W celu poprawnego działanie modeli znormalizowaliśmy wszystkie dane do wartości z zakresu [0,1]. Wykorzystaliśmy do tego $MinMaxScaler$, skalujący względem minimalnej i maksymalnej wartości (minimalna = 0, maksymalna = 1).

2.3 Badanie korelacji ze zmienną celu

W celu badania korelacji zmiennych ze zmienną celu głównie zwracaliśmy uwagę na wykresy skrzypcowe wszystkich zmiennych, rozróżniając dane dla klasy 0 oraz 1.
Dodatkowo tworzyliśmy macierze korelacji oraz sami wnioskowaliśmy, które kolumny mogą być przydatne (przykładowo kolumna dotycząca tego czy klient na telefon służbowy wydawała nam się bez znaczenia, tak też się okazało w dalszej analizie). Ostatecznie z 15 kolumn wejściowych postanowiliśmy zostawić 9, w tym między innymi: liczba dzieci, dochód roczny, wysokość kredytu i rocznej spłaty, zatrudnienie i wiek (zamieniony z dni na lata).
Dodatkowo stworzyliśmy dwie nowe kolumny: stosunek wysokości kredytu do pensji rocznej oraz stosunek rocznej spłaty kredytu do pensji rocznej.
Przykładowy wykres skrzypcowy zmiennej, która była kluczowa w przewidywaniu kolumny Target, został przedstawiony poniżej.



Rysunek 4: Wykres skrzypcowy dla zmiennej YEARS OLD w rozróżnieniu na klasy 0 i 1

3 Wstępne modelowanie

3.1 Niezbalansowanie danych

Z niezbalansowaniem danych probowaliśmy walczyć na kilka sposobów:

- szukanie modeli które lepiej radzą sobie z wykrywaniem klasy 1 - w praktycznym zastosowaniu naszego modelu ważniejsze jest wykrycie przypadków, gdzie klient będzie miał problem ze spłaceniem kredytu. Dlatego szukaliśmy modelu, który przewiduje odpowiednio dużo rekordów ze zmienną celu 1, nawet kosztem niższej metryki accuracy
- oversampling - sztuczne tworzenie nowych rekordów o zmiennej celu 1, różnymi strategiami
- badanie innych metryk niż accuracy

3.2 Dobór odpowiednich metryk

Ze względu na niezbalansowanie danych musieliśmy zastanowić się, które z metryk powinniśmy badać.

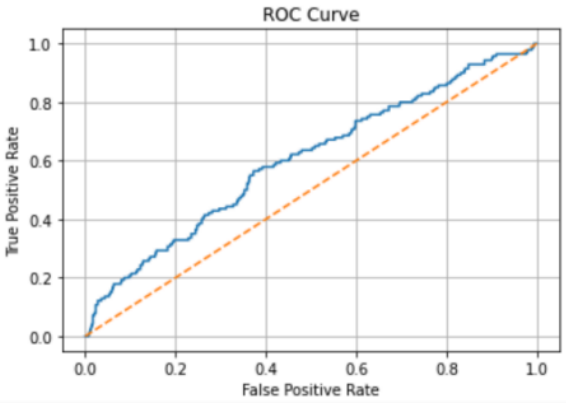
- accuracy- nieodpowiednia forma sprawdzenia modeli z powodu niezbalansowanych danych - przewidzenie klasy 0 dla każdego rekordu daje nam accuracy na poziomie 80%, co wydaje się wysokim wynikiem, ale nie ma zastosowania praktycznego
- PRECISION - miara tego jaka cześć rekordów z przewidziana klasa 1 jest przewidziana poprawnie. W naszym przypadku z praktycznego punktu widzenia jest to kluczowa metryka
- RECALL - jaka część jedynek została wykryta, kolejna kluczowa metryka dla naszych danych
- F1-Score - średnia harmoniczna dwóch powyższych
- liczba rekordów z przewidzianymi klasami 0 i 1 w porównaniu do danych wejściowych
- ROC Score i wykres ROC curve - stosunek liczności poprawnie przewidzianych klasy 1 do niepoprawnie przewidzianych W analizie poszczególnych modeli zwracaliśmy uwagę na wszystkie z powyższych, z wyjątkiem accuracy. (oczywiście również patrzyliśmy na nią, ale nie był to wyznacznik oceny modeli)

3.3 Sprawdzanie modeli

Do sprawdzania modeli używaliśmy głównie naszej funkcji 'model check'.

	precision	recall	f1-score	support
0.0	0.87	0.62	0.73	647
1.0	0.24	0.56	0.34	140
accuracy			0.61	787
macro avg	0.56	0.59	0.53	787
weighted avg	0.76	0.61	0.66	787

```
Wartość: 0.0, Liczba wystąpień: 464
Wartość: 1.0, Liczba wystąpień: 323
For the StackingClassifier
The TEST accuracy is 0.6124523506988564
The ROC score for TEST data is 0.5935802605431663
```



Rysunek 5: Przykładowy wynik funkcji 'model check'

3.4 Bazowe modele

Wiele bazowych modeli, które testowaliśmy na naszych danych radziło sobie z nimi bardzo niezadowalająco. Większość z nich dla wszystkich bądź prawie wszystkich rekordów przewidywało klasę 0.
Modele bazowe, które dały nam wyniki odchodzące od losowych, a nawet dość zadowalające to między innymi:

- LogisticRegression z argumentem class weight = 'balanced'
- GaussianNB
- RandomForestClassifier z argumentem class weight ='balanced'
- XGBClassifier
- KNeighboursClassifier
- DecisionTreeClassifier z argumentami max depth=5,class weight = 'balanced',min samples split=10

Warto zauważyć, że kilka z powyższych modeli ma argument $class_weight$ ustawiony na 'balanced'. Wówczas modele lepiej radziły sobie z naszymi bardzo niezbalansowanymi danymi.

4 Złożone modelowanie i model końcowy

4.1 Złożone modelowanie

Z użyciem powyższych modeli budowaliśmy różne modele złożone: VotingClassifiery oraz StackingClassifiery.
Modele złożone zależnie od parametrów dawały trochę lepsze wyniki niż modele bazowe. Najlepszy wynik uzyskaliśmy w StackingClassifierze z estymatorami wymienionymi wyżej oraz Regresją Logistyczną jako estymator finalny. Z tym modelem pracowaliśmy dalej, zmieniając parametry, próbując używać różnych technik oversamplingu oraz manipulowaniem thresholdu. To właśnie ta ostatnia strategia dała nam poprawę wyników modelu, dlatego próbowaliśmy zmieniać threshold w zakresie około [0.50,0.70].

4.2 Model końcowy

Najbardziej obiecujący model udało nam się uzyskać przez ustawienie thresholdu na 0.58 w StackingClassifierze z Regresją Logistyczną jako model finalny. Poniżej znajdują się wyniki dla zbioru testowego oraz zbioru walidacyjnego.

```
Accuracy with new threshold: 0.7580286168521463
              precision    recall  f1-score   support

         0.0       0.85      0.85      0.85      2581
         1.0       0.32      0.32      0.32       564

    accuracy                           0.76      3145
   macro avg       0.59      0.59      0.59      3145
weighted avg       0.76      0.76      0.76      3145

Wartość: 0, Liczba wystąpień: 2584
Wartość: 1, Liczba wystąpień: 561
For the StackingClassifier
The TEST accuracy is 0.6155802861685215
The ROC score for TEST data is 0.5898168833345698
```

Rysunek 6: Wyniki modelu końcowego dla danych testowych

```
Accuracy with new threshold: 0.7522236340533672
              precision    recall  f1-score   support

         0.0       0.84      0.86      0.85       647
         1.0       0.28      0.26      0.27       140

    accuracy                           0.75       787
   macro avg       0.56      0.56      0.56       787
weighted avg       0.74      0.75      0.75       787

Wartość: 0, Liczba wystąpień: 660
Wartość: 1, Liczba wystąpień: 127
For the StackingClassifier
The TEST accuracy is 0.6124523506988564
The ROC score for TEST data is 0.5935802605431663
```

Rysunek 7: Wyniki modelu końcowego dla danych walidacyjnych

4.3 Kroswalidacja modelu końcowego

Podjęliśmy również próbę dodatkowego poprawienia wyników dla modelu końcowego za pomocą kroswalidacji. Pierwszym podejściem do tego było zastosowanie kroswalidacji dla poszczególnych modeli używanych w StackingClassifierze. Jako pierwszy na celownik został wzięty DecisionTree użyty jako składowa. Gdy czasochłonne próby ledwo co poprawiły model uznaliśmy, że lepszym podejściem będzie spróbowanie zastosowania kroswalidacji na całym StackingClassifierze. Jednak ten pomysł również porzuciliśmy, ponieważ często długi czas wywoływania kończył się jedynie otrzymaniem błędów.

5 Cele biznesowe

Jeśli chodzi o zastosowania to widzimy tutaj dwie możliwości. Stworzenie takiego typu modelu mogłoby być przydatne w dwóch biznesach:

1. **bankowość** - dość oczywiste zastosowanie naszego modelu, jednak warto zaznaczyć tu, że dla banku prawdopodobnie priorytetem będzie unikanie dawania kredytu osobom, które mogą być ryzykowne jeśli chodzi o jego spłacanie. Dlatego tutaj z metryk recall i precision, bardziej kluczowa wydaje się być metryka recall mierząca to ile osób objętych ryzykiem zostało wykrytych.

2. **ubezpieczalnia** - tutaj natomiast mimo, że tematyka modelu prawdopodobnie nie byłaby związana tylko z finansami, ubezpieczalnia chce znaleźć klientów z małym ryzykiem i to właśnie im sprzedać ubezpieczenie.

Dlatego tutaj ważniejsza wydaje się być metryka precision, ponieważ ważne jest aby jak największa liczba podjętych klientów faktycznie przyniosła ubezpieczalni zyski. Jeżeli ubezpieczenie zostanie przedane osobie, która była zakwalifikowana jako klient "nieryzykowny", a potem okazałoby się, że jednak jest, to przyniosło by to ubezpieczalni straty.