Wydział Matematyki i Nauk Informacyjnych

Politechniki Warszawskiej

Data mining

Projekt 2

Jakub Zbrzezny

Nr indeksu: 286689

7 czerwca 2020

1 Wprowadzenie.

Celem projektu jest przewidzenie, czy dane gospodarstwo dotyka bieda. Gospodarstwo jest objęte biedą, jeżeli wartość zmiennej *poor* wynosi "Poor" (zmienna *poor* występuje w zbiorach o nazwach kończących się na *labels* - jest to więc zmienna objaśniana). Natomiast zmiennymi objaśniającymi są wszystkie zmienne z odpowiadającego zbioru o nazwie kończącej się na *data* - jest ich bardzo dużo, więcej niż 100, są nimi na przykład: identyfikator gospodarstwa, zmienna określająca dzielnicę, zmienna wskazująca, czy jest elektryczność lub też zmienna określająca, jaki jest dochód.

2 Klasyfikatory.

Przygotowanie danych wygląda następująco:

Złączyłem zbior *train data*, *train labels* w jeden zbiór oraz *test data*, *valid labels* również w jeden zbiór.

Następnie łączyłem oba otrzymane zbiory, żeby czynniki były takie same, a następnie indeksowałem na zbiór treningowy oraz testowy.

Dodatkowo jeszcze sprawdziłem, gdy łączyłem zbiór treningowy z walidacyjnym.

2.1 1 klasyfikator.

1 klasyfikatorem, który zbadałem, jest *randomForest* z parametrem $ntree = 50$.

Robiłem model pełny (zmienną *poor* uzależniłem od reszty).

Prawdopodobieństwo, że predykcja zmiennej *poor* jest równa zmiennej *poor* ze zbioru testowego, wynosi 0.5006667 dla zbioru testowego, a dla walidacyjnego 0.7766667. Natomiast wartość klasyfikacji mierzonej jako AUC wynosi 0,499 dla testowego, a dla walidacyjnego 0,862.

2.2 2 klasyfikator.

2 klasyfikatorem, który sprawdziłem, jest *randomForest* z parametrem $ntree = 50$. Tym razem przeprowadziłem selekcję zmiennych funkcją *CMIM* z pakietu *praznik* z liczbą wybranych zmiennych równą 6.
W przypadku zbioru testowego zostały wybrane następujące zmienne:
cons 0901, *cons 0106*, *geo district*, *cons 0501*, *cons 0111*, *cons 0508*

Następnie po dopasowaniu modelu z 6 optymalnymi zmiennymi, prawdopodobieństwo, że wartość przewidywana zmiennej *poor* jest równa wartości zmiennej *poor* ze zbioru testowego, wynosi 0.5056667 dla testowego, oraz 0.7493333 dla walidacyjnego, czyli dla testowgo prawdopodobieństwo jest o 0,005 większe niż w przypadku modelu pełnego.
Wartość AUC dla testowego jest równa 0.504, zatem jest trochę wyższa o 0,005 od wartości AUC dla modelu pełnego. Natomiast dla walidacyjnego AUC jest równe 0.820. Zatem ten klasyfikator jest lepszy od 1, ponieważ otrzymane wartości są wyższe od wartości dla 1 klasyfikatora, ponadto ten klasyfikator działa szybciej, gdyż w modelu uzależniamy zmienną *poor* tylko od 6 zmiennych, a nie od 343.

Później jeszcze sprawdziłem klasyfikację z selekcją zmiennych dla $k = 12$, ale wtedy dla zbioru testowego wartość AUC wyszła mniejsza (była równa 0.500), chociaż dla zbioru walidacyjnego AUC było większe (wynosiło 0.84).

2.3 3 klasyfikator.

Kolejnym klasyfikatorem, który zbadałem, jest *randomForest*, ale z uwzględnieniem wybranych danych indywidualnych. Dane przygotowałem w następujący sposób: Wziąłem pierwsze 7 kolumn z *individual data*, następnie złączyłem ten zbiór ze zbiorem złączonym z treningowego i walidacyjnego funkcją *inner join* z pakietu *dplyr*. Zauważyłem, że w 6, 7 kolumnie były braki danych, więc te wiersze z brakami danych usunąłem, gdyż selekcja funkcją *CMIM* działa tylko wtedy, gdy nie ma braków danych. Selekcję zrobiłem z liczbą zmiennych równą 6. Otrzymałem następujące zmienne dla testowego:
cons 0801, *geo district*, *cons 0111*, *cons 0106*, *hld nbcellpho*, *cons 0508*
Otrzymane prawdopodobieństwo wynosi 0.6616667 dla testowego, a 0.754 dla walidacyjnego. Wartość AUC dla testowego jest równa 0.0.494 , a dla walidacyjnego jest bardzo mała w porównaniu z poprzednimi - wynosi tylko 0.494 (prawdopodobnie przez problemy, które wynikały z tego, że zbiór danych indywidualnych ma aż 56218 wierszy, a zbiór walidacyjny tylko 3000, a treningowy 6000).

2.4 4 klasyfikator.

Następnym klasyfikatorem jest *bagging* z parametrem $mfinal = 25$ na zbiorze złączonym z treningowego i walidacyjnego wraz z przeprowadzoną selekcją zmiennych funkcją *JMIM* z pakietu *praznik*. Otrzmałem następująceo zmienne dla zbioru testowego:

cons 0901, *hld dwateros*, *geo district*, *cons 0106*, *cons 0111*, *cons 0501*, *cons 0801*, *hld nbcellpho*, *cons 0401*, *hld cooking*, *cons 0803*, *cons 1204*

Otrzymane prawdopodobieństwo wynosi 0.498 dla testowego, a dla walidacyjnego 0.7403333, natomiast wartość AUC 0.492 dla testowego, a 0.82 dla walidacyjnego, zatem dany klasyfikator jest trochę gorszy od klasyfikatora *randomForest* z selekcją zmiennych bez danych indywidualnych.

2.5 5 klasyfikator.

Ostatnim klasyfikatorem, który sprawdziłem, jest klasyfikator z modelem opartym na funkcji *ranger* z parametrem $num.trees = 50$, wraz z selekcją zmiennych i danymi takimi jak poprzednio.

Otrzymane prawdopodobieństwo wynosi 0.503 dla testowego, a dla walidacyjnego 0.7466667, natomiast wartość funkcji AUC dla testowego jest równa 0.492, a dla walidacyjnego 0,821, czyli dany klasyfikator dla testowego ma obie wartości bardzo bliskie wartościom dla klasyfikatora *randomForest* z selekcją zmiennych, ale trochę niższe.

3 Podsumowanie eksperymentów i wybór końcowej metody.

Podsumowując, ogólnie wartości AUC klasyfikatorów nie różniły się mocno poza klasyfikatorem z danymi indywidualnymi i zbiorem testowym i walidacyjnym. Dla zbioru testowego prawdopodobieństwo, że wartość predykcji jest równa wartości zmiennej *poor* ze zbioru testowego wychodziło znacznie niższe niż w przypadku zbioru walidacyjnego. Tak samo było w przypadku wartości klasyfikacji mierzonej jako AUC. Generalnie wartość AUC oraz prawdopodobieństwo, że wartość predykcji jest równa wartości odpowiedniej zmiennej ze zbioru testowego, w przypadku użycia selekcji zmiennych, jest mniejsze niż w przypadku modelu pełnego ze wszystkimi zmiennymi, ale często jest znacznie szybsze, gdyż mniej zmiennych jest branych pod uwagę. Dlatego nierzadko lepszymi klasyfikatorami są klasyfikatory wraz z robieniem selekcji zmiennych.

Ostatecznie więc wybieram klasyfikator z modelem opartym na funkcji *randomForest* z selekcją zmiennych ze względu na to, że *randomForest*, ma najwyższą wartość AUC oraz jest najszybszy.