MPWzD
Projekt zaliczeniowy

Dawid Ciochoń | Jakub Górowski | Adrian Gzyl

Treść prezentacji

1. Wprowadzenie - dane i temat
2. Zespół
3. Rozwiązanie
4. Rezultaty
5. Interpretacja
6. Wnioski

Dane i temat projektu

- Dane dotyczące sytuacji absolwentów szkół średnich (czy poszli na studia czy nie itp.) z lat 80 w USA
- Ponad 24 tys. badanych
- 6 lat badań
- Wśród badanych: absolwenci oraz uczniowie drugich klas szkół średnich, nauczyciele, dyrektorzy oraz rodzice

Przedstawienie danych (1)

- **gender** - oznacza płeć
- **ethnicity** - oznacza pochodzenie (afroamerykańskie, latynoskie, inne)
- **score** - rezultat testu przeprowadzonego na zakończenie drugiego stopnia edukacji (high school)
- **fcollege** - zmienna binarna oznaczająca, czy ojciec ankietowanej osoby ukończył edukację wyższą
- **mcollege** - zmienna binarna oznaczająca, czy matka ankietowanej osoby ukończyła edukację wyższą
- **home** - zmienna binarna oznaczająca, czy rodzina ankietowanej osoby posiada swój dom na własność
- **urban** - zmienna binarna oznaczająca, czy szkoła ankietowanego znajduje się na terenie zurbanizowanym
- **unemp** - stopa bezrobocia panująca w hrabstwie (okręgu - części składowej stanu) będącym miejscem zamieszkania osoby ankietowanej w roku 1980
- **wage** - stawka godzinowa za pracę fizyczną w fabryce w danym stanie w roku 1980 (wyrażone w USD)
- **distance** - odległość od domu ankietowanej osoby do najbliższej uczelni (wyrażona w 10 milach)
- **tuition** - średnie czesne wymagane przez uczelnie w danym stanie (wyrażone w 1000 USD)
- **education** - czas trwania edukacji osoby ankietowanej (wyrażony w latach)
- **income** - zmienna binarna oznaczająca, czy roczny dochód rodziny przekracza 25 tys. USD
- **region** - zmienna binarna oznaczająca region (zachód USA, czy inny)

Przedstawienie danych (2)

	score	unemp	wage	distance	tuition	education
count	4739.000000	4739.000000	4739.000000	4739.000000	4739.000000	4739.000000
mean	50.889029	7.597215	9.500506	1.802870	0.814608	13.807765
std	8.701910	2.763581	1.343067	2.297128	0.339504	1.789107
min	28.950001	1.400000	6.590000	0.000000	0.257510	12.000000
25%	43.924999	5.900000	8.850000	0.400000	0.484990	12.000000
50%	51.189999	7.100000	9.680000	1.000000	0.824480	13.000000
75%	57.769999	8.900000	10.150000	2.500000	1.127020	16.000000
max	72.809998	24.900000	12.960000	20.000000	1.404160	18.000000

Przedstawienie danych (3)

Dalsze kroki związane z danymi...

- usunięcie outlierów za pomocą metody bazującej na średniej i odchyleniu standardowym
- zamiana zmiennych tekstowych na zmienne liczbowe
- utworzenie zmiennej zależnej na podstawie zmiennej education (zawiera ona 38% “1”, zatem proporcje klas są zachowane)
- podział na zbiór treningowy i testowy w stosunku 70:30

Zespół

- Ponad 4 lata znajomości i wiele wspólnych projektów
- Dobra komunikacja w zespole
- Każdy członek naszego zespołu posiada doświadczenie komercyjne
- Wiedza z zakresu statystyki, ekonometrii, analizy danych oraz programowania

Rozwiązanie

Random forest - jest to wariacja techniki drzewa decyzyjnego - zamiast jednego, rozpatruje się w niej n drzew. Końcową decyzję o wartości zmiennej zależnej algorytm podejmuje na podstawie średniej (regresja) lub większości “głosów” na daną wartość (klasyfikacja). Technika ta charakteryzuje się odpornością na przeuczenie oraz problemy jakie można napotkać w zbiorze danych.

Parametry:

- n_estimators - liczba wykorzystanych w modelu drzew
- max_features - przyjmuje się, że dla klasyfikacji parametr ten powinien być równy pierwiastkowi z liczby zmiennych
- min_samples_leaf - określa minimalną wielkość węzła podziału obserwacji
- max_leaf_nodes - określa maksymalną liczbę liści jaką drzewo może posiadać.

Rozwiązanie

Decision tree - algorytm drzewa decyzyjnego należy do rodziny algorytmów uczenia nadzorowanego. Można go stosować zarówno do problemów klasyfikacji jak i regresji. W każdym węźle sprawdzany jest pewien warunek dotyczący danej obserwacji, i na jego podstawie wybierana jest jedną z gałęzi prowadząca do kolejnego wierzchołka. Klasyfikacja danej obserwacji polega na przejściu od korzenia do liścia i przypisaniu do tej obserwacji klasy zapisanej w danym liściu.

Parametry:

- criterion - kryterium podziału
- max-depth - maksymalna głębokość (liczbę poziomów) drzewa
- min-sample-leaf - minimalna liczba obserwacji w liściu
- min-samples-leaf - minimalna liczba obserwacji w liściu potrzebna do dokonania podziału

Rozwiązanie

SVM - zestaw metod nadzorowanego uczenia maszynowego, w którego skład wchodzą metody klasyfikacji, regresji i wykrywania outlierów. W projekcie zostanie wykorzystana metoda klasyfikacji SVC, gdyż nadaje się ona do przewidywania wartości zmiennej objaśnianej, która przyjmuje wartości 0 i 1.

Parametry:

- kernel - funkcja transformująca parametry wejsciowe (np. linear lub rbf)
- gamma - sposób dopasowania parametrów kernle'a

Rezultaty

Random forest

Accuracy (train): 0.748

Accuracy (test): 0.712

AUC (test): 0.668

Sensitivity: 0.49

Specificity: 0.85

Precision: 0.85

Decision tree

Accuracy (train): 1.0

Accuracy (test): 0.65

AUC (test): 0.629

Sensitivity: 0.55

Specificity: 0.71

Precision: 0.71

SVM

Accuracy (train): 0.728

Accuracy (test): 0.717

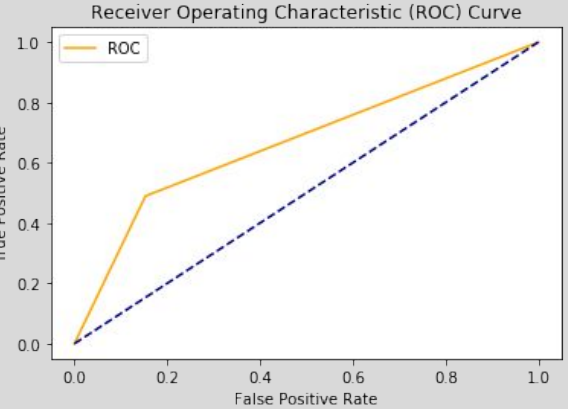
AUC (test): 0.678

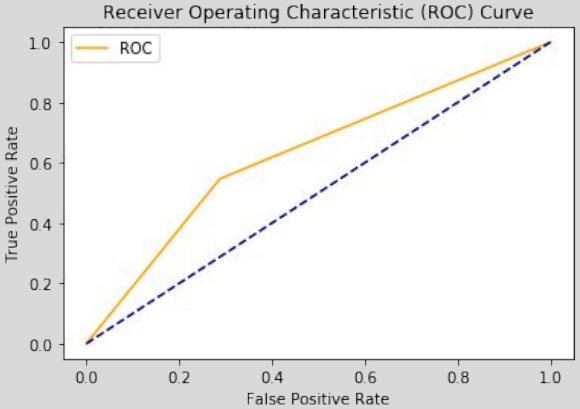
Sensitivity: 0.52

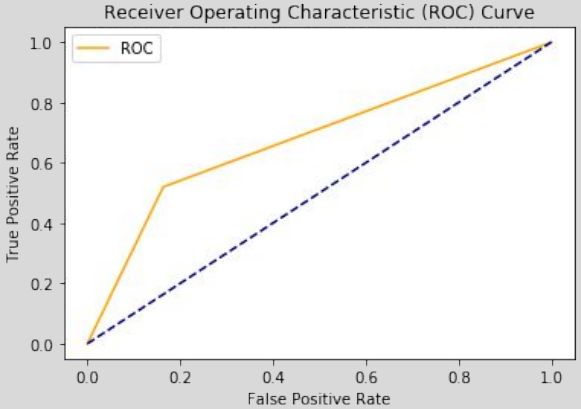
Specificity: 0.84

Precision: 0.84

Porównanie krzywych ROC (RF, DT, SVM)







Najlepsze kluczowe KPI (Accuracy oraz AUC) posiadał klasyfikator

SVM

7 min

Tyle czasu zajęło wykonanie tuningu klasyfikatora random forest na podstawie zadanej siatki hiperparametrów. Algorytm ten dla finalnej postaci osiągnął rezultaty zbliżone do SVM… które jednak nie wymagało *dostrojenia*

Interpretacja

- rozpatrywane dane skutecznie można klasyfikować wykorzystując hiperpłaszczyznę
- drzewo decyzyjne jest najbardziej czułe na przetrenowanie
- SVM jest najbardziej odpornym na przetrenowanie algorytmem
- liczba drzew ma pozytywny skutek na jakość rezultatów jedynie do pewnego momentu (najlepszy klasyfikator zawierał ich 200 - z możliwych 5000)
- nie powinno nakładać się ograniczenia na liczbę liści - najlepszy estymator posiadał największą możliwą ich liczbę
- drzewo decyzyjne ma tendencję do określania obserwacji jako 1 niż 0 - stąd też większa czułość i niższa specyficzność
- wszystkim rozpatrywanym klasyfikatorom bliżej jest do klasyfikatora losowego, niż idealnego (na podstawie AUC)

Napotkane problemy

W zasadzie jedynym problemem, jaki napotkał nasz zespół był konflikt języków programowania. Do tej pory znaczną część projektów dotyczących analizy statystycznej realizowaliśmy w języku R. Pomimo, że podobnie jak Python jest to język interpretowalny, to jednak występują różnice w sposobie np. dostępu do danych zgromadzonych w dataframe'ach, czy samym sposobie iteracji (w R iteruje się od 1). Z powodu przyzwyczajenia do języka R początkowe pisanie kodu obarczone było koniecznością korzystania z dokumentacji oraz stron takich jak stack overflow.

Intrygującym problemem była również specyfika Pythona, który wymagał aby wszystkie atrybuty posiadały wartość liczbową - z tego powodu konieczne było dokonanie niezbędnych przekształceń aż dla 8 zmiennych jakościowych.

W czasie prac nad projektem problemem okazało się również przestrzeganie wyznaczonych sprintów. Ze względu na specyfikę projektu oraz podział pracy każdy mógł pracować w preferowanym przez siebie czasie, przez co postępy pojawiały się nieregularnie - lecz wszystko co zostało zadeklarowane, znalazło się w końcowym rozwiązaniu (a nawet zostało ono wzbogacone)

Wnioski

Wykorzystanie nawet najprostszych technik uczenia maszynowego pozwala na uzyskiwanie lepszych rezultatów niż losowa klasyfikacja

Klasyfikatory opracowane na podstawie długiego procesu dopasowania nie zawsze są lepsze od tych bazowych

Lasy losowe w istocie stanowią udoskonalenie techniki drzew decyzyjnych i powinny być wykorzystywane jako ich zamienniki

Proces tuningu hiperparametrów powinien być prowadzony przy świadomości jak zmieniane parametry wpłynąć mogą na model