# KNN

Użyte cechy: Budżet, gatunek, kraj pochodzenia firm produkujących film i kraj powstania filmu; zostały wybrane zgodnie z naszą wiedzą ekspercką. Cechy posiadających wiele opcji jednocześnie (np. gatunek: film może być jednocześnie romansem i komedią) zostały rozbite na tabele, gdzie każda kolumna reprezentowała jedną z opcji (nasz przykładowy film miałby 1 w kolumnie romans i komedia a 0 w pozostałych kolumnach). Wszystkie cechy numeryczne (np. budżet) zostały znormalizowane z użyciem StandardScaler() z biblioteki sklearn.preprocessing. Jako, że wszystkie cechy były reprezentowane jako liczby mogliśmy zastosować dystans euklidesowy (np.sqrt(np.sum((datapoint1 – datapoint2) \*\* 2))) do obliczenia dystansu między poszczególnymi rzędami.

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznieW pierwszym podejściu szukaliśmy najbliższych sąsiadów na całym zbiorze danych (nie dzieląc go na poszczególnych studentów) zakładając, że jako, że należą do podobnej grupy społecznej (studenci jednej politechniki) ich opinie mogłyby być uznane jako wymienne. Uzyskaliśmy poniższy wynik:

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznieZdecydowanie lepsze wyniki udało się nam uzyskać po zastosowaniu takiej samej metody na danych podzielonych na oceny poszczególnych studentów:

Hiperparametrem używanym przez nas w uczeniu była liczba sąsiadów (k) dla algorytmu kNN. Dla zbioru zawierającego wszystkich studentów najlepsze k wynosiło 20 a dla studenta 1002 k wynosiło 11 MSE=5.5. Dla zbiorów podzielonych na różnych studentów określone zostały różne k. Najlepsze k było wybierane dla wartości, która dała najmniejszy średni błąd kwadratowy (mean squared error) na zbiorze testowym powstałym dla 5 różnych podziałów zbioru treningowego na treningowy i testowy (cross-validation).

# Drzewo decyzyjne i las losowy

Nasz algorytm drzewa decyzyjnego wybiera losową cechę z wszystkich dostępnych i sprawdza czy z jej pomocą (dla cech numerycznych porównuje czy jest większa niż kolejne 20 wartości między -0,9 a 0,9 a dla cech kategorialnych czy jest równa 0) może podzielić dostępne dane (trenowaliśmy na danych filmów dopasowanych do ocen pojedynczego studenta) na dwa zbiory w tym oba zawierające przynajmniej min\_size elementów w tym przynajmniej uniformity (ułamek) ma taką samą wartość oceny. Jeśli jednak taka kolumna nie zostanie znaleziona program podejmie próbę podziału zbioru na dwa niezależnie od wartości ocen. Przyrost informacji nie był mierzony. Pierwszy możliwy podział był realizowany. Proces był zatrzymywany, jeśli kolejny podział nie był możliwy.  
Losowość wbudowana w wybór kolumn pozwala nam tworzyć różne drzewa dla takich samych danych pozwalając nam tworzyć lasy losowe. Rezultat lasu losowego to najczęściej dawana odpowiedź przez pojedyncze drzewa (Counter(results).most\_common(1)[0][0]).  
Hiperparametrami jakich szukaliśmy były wymienione wyżej min\_size czyli minimalny rozmiar wydzielonego zbioru oraz uniformity czyli minimalny ułamek wydzielanego zbioru składający się z takich samych elementów. Dla lasów losowych dodatkowo szukaliśmy najlepszej ilości drzew.  
Najlepsze dla studenta 1002 to split\_ratio = 0.6, min\_split = 2, MSE: 4.6(1) a dla jego lasu losowego 10 drzew, split\_ratio = 0.6, min\_split = 2, MSE: 4.6(1).  
Szukaliśmy ich podobnie jak dla kNN: porównując wartości średniego błędu kwadratowego (mean squared error) otrzymanego na zbiorze walidacyjnym przez model uczony na zbiorze treningowym.

Obraz zawierający tekst, pismo odręczne, rysowanie, szkic

Opis wygenerowany automatycznieUtworzone drzewa były bardzo płytkie: z reguły składały się z 3 liści, jednak dla celów demonstracyjnych rozrysowaliśmy bardziej rozłożyste drzewo.

Mimo słabego wyglądu naszych drzew udało się nam uzyskać relatywnie dobre rezultaty:

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznieNiższy wynik naszego lasu losowego w porównaniu do drzewa decyzyjnego może być spowodowany niską (poniżej 50%) dokładnością co w połączeniu z prostotą naszych drzew mogło uwydatnić błędy w naszym lesie losowym.

# Podobieństwo między ludźmi

Do obliczenia podobieństwa między dwoma studentami użyliśmy podobieństwa Pearsona.   
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie  
Podobieństwo obliczane jest dla każdego studenta względem rozważanego studenta, ustawiani są w kolejności od najbardziej podobnego do najmniej podobnego. Szukając opinii danego studenta dla danego filmu studenci w tej liście są po kolei przeszukiwani i pierwszy posiadający ocenę danego filmu jest używany do przewidzenia tej oceny (jego ocena jest kopiowana).  
Hiperparametrem użytym w naszym algorytmie była minimalna ilość filmów jakie wspólnie musieli widzieć. Ze względu na brak wpływu niewspólnych filmów na wartość podobieństwa, by uniknąć sytuacji, gdzie gust dwóch studentów uznawany jest za identyczny, ponieważ oboje mają taką samą opinię na temat jedynego filmu, który oboje obejrzeli wprowadziliśmy minimum wspólnych filmów jakie studenci musieli zobaczyć zanim ich podobieństwo było rozważane.  
Szukaliśmy hiperparametru porównując porównując wartości średniego błędu kwadratowego (mean squared error) otrzymanego na zbiorze walidacyjnym przez model uczony na zbiorze treningowym.  
Najlepszy uzyskany parametr to 45.

Osiągnięto relatywnie słaby wynik:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Filtrowanie kolaboracyjne

Zgodnie z metodą przedstawioną na wykładzie szukaliśmy zestawu parametrów dla każdego studenta i cech każdego filmu szukając minimum różnicy między ocenami będącymi rezultatem działania na nich a faktycznej oceny użytkownika.  
Wartości te były z początku losowe:

i były zmieniane 50 razy w trakcie uczenia w pętli:  
Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Hiperparametrami były wartości K, alpha i beta.

K – cechy ukryte

α – szybkość uczenia – określa jak szybko zmieniać poszczególne parametry w czasie treningu

β – współczynnik regularyzacji – przeciwdziała przetrenowaniu i szumom

Najlepsze uzyskane to:

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie  
Uzyskaliśmy następujące wyniki:Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

# Wyniki i wnioski

To, jak dobre są nasze modele ocenialiśmy na zbiorze testowym.

Najlepszy wynik dokładności uzyskaliśmy dla drzewa decyzyjnego: 39,79%; jednak wszystkie metody, gdzie zadanie było podzielone na przewidywania dla pojedynczych studentów wszystkie uzyskały dokładność około 39%. Co ciekawe, wartość dokładności połączona z błędami o 1 ocenę (Dokładność +/- 1) była najwyższa dla filtrowania kolaboracyjnego: 78,06% i o ile była ona zbliżona do innych uzyskanych wartości nie jest do nich o tyle zbliżona do co dokładność.  
Uzyskane wyniki posortowane po Dokładność +/- 1:

|  |  |  |
| --- | --- | --- |
| Metoda | Dokładność [%] | Dokładność +/- 1 [%] |
| Filtr. kolaboracyjne | 38,9 | 78,06 |
| Podobieństwo | 35,82 | 71,77 |
| Drzewo decyzyjne | 39,79 | 70,89 |
| Las losowy | 39,53 | 70,8 |
| KNN - części | 39,09 | 69,69 |
| KNN - całość | 24,3 | 61,17 |

Nasze najgorsze wyniki dokładności to KNN dla wszystkich studentów jednocześnie: 24,3% oraz przy użyciu podobieństwa między ludźmi: 35,82%

Jak widać, nasze założenie o podobieństwie między studentami było nieprawidłowe; jednak analizując dodatkowo podobieństwo między gustami użytkowników w trakcie 3 zadania (omówione w dziale Podobieństwo między ludźmi) można osiągnąć lepszy choć nadal słaby wynik. Sugeruje to, że zarówno zbiór danych jak i zadanie przewidzenia ocen należałoby podzielić na podzadania obejmujące pojedynczych użytkowników.   
Jeśli jednak uogólnimy zadanie i skupimy się na metryce dokładności +/- 1 metody najlepszą metodą wydaje się filtrowanie kolaboracyjne jednak może być to również spowodowane dość nierównomiernym rozkładem ocen uzyskanym przez tę metodę:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Oceny | 0 | 1 | 2 | 3 | 4 | 5 |  |
| Dane treningowe | 1648 | 3304 | 6084 | 8297 | 6888 | 5999 | Ilość |
|  | 5,11% | 10,25% | 18,88% | 25,75% | 21,38% | 18,62% | Ułamek |
| Filtr. kolaboracyjne | 3 | 270 | 2010 | 4638 | 3097 | 722 | Ilość |
|  | 0,03% | 2,51% | 18,72% | 43,18% | 28,84% | 6,72% | Ułamek |
| Drzewo | 542 | 509 | 1815 | 3903 | 1942 | 2029 | Ilość |
|  | 5,05% | 4,74% | 16,90% | 36,34% | 18,08% | 18,89% | Ułamek |

Rozkład ocen z różnych źródeł, ułamek odnosi się do proporcji jaką dana ocena ma w całym zbiorze.