MMDS Challenge
Raport II

Damian Jackowski 134496
Patryk Olejniczak 114989
Dawid Wiśniewski 116912

17 stycznia 2018

1 Sformułowanie problemu

Zadanym problemem jest zaimplementowanie modelu predykcyjnego pozwalającego przewidzieć liczbę wyświetleń, liczbę odpowiedzi i rezultat wystawienia ogłoszenia (czy zostało sprzedane). Powyższy model może wykorzystać do rozwiązania postawionego problemu dane archiwalne wystawionych ogłoszeń oraz wyszukiwanych fraz na stronie OLX dla każdego miesiąca od listopada 2016 do września 2017 - są to tzw. dane treningowe na których można uczyć model. Powyższe trzy metryki ogłoszenia należy przewidzieć dla danych testowych z października 2017 (zbiór danych udostępniony na inaugurację konkursu), stycznia 2018 (opcjonalnie) oraz marca 2018 (finalny zbiór testowy).

Rozwiązania należy wysyłać na stronę konkursu (dataninija.olx.pl) gdzie zostają one poddane weryfikacji. Weryfikacja polega na sprawdzeniu miejsca w rankingu dla każdego parametru danego ogłoszenia. Każdy z trzech wymienionych wyżej parametrów (metryk) posiada własny ranking (wyższa wartość metryki to wyższe miejsce w rankingu). Weryfikacja polega na sprawdzeniu miejsca w rankingu dla każdego parametru ogłoszenia i obliczenia błędu oszacowania. Błąd ten obliczany jest następującym wzorem:

$L_{rank} = \frac{1}{\sum_{k<l} n_k n_l} \sum_{y_i<y_j} ([[f(x_i) > f(x_j)]] + 1/2[[f(x_i) = f(x_j)]])$

Wartość obliczona z powyższego wzoru mieści się w przedziale od 0 do 1 i oznacza jak bardzo błędne było oszacowanie (0 - idealne, 1 - bardzo złe). Wzór ten wykorzystywany jest dla rankingu każdego parametru. Ostatecznym wynikiem jest średni błąd oszacowania powyższych metryk.

2 Opis metody predykcyjnej

Zgodnie z instrukcją do zadania wykorzystano dwa modele predykcyjne: bazowy i konkursowy. Model bazowy to początkowa metoda predykcyjna wy-

brana intuicyjnie. Model konkursowy to po prostu ulepszony model bazowy, który poprawił wynik modelu bazowego w konkursie.

2.1 Model bazowy

Przyjęto, że do modelu bazowego użyte zostaną następujące kolumny z plików z danymi:

- category_id,
- city_id,
- user_id,
- paidads_id_index,
- has_phone,
- has_person

Każda z powyższych kolumn została wykorzystana jako osobna cecha. Dane z kolumn has_phone i has_person zmapowano do wartości t=1 oraz f=0. Ponadto wszystkie puste pola z powyższych kolumn wypełniono wartością -1.

Zdecydowano się na niezależne przewidywanie każdej z wymienionych w poprzednich rozdziałach metryk. Do przewidywania czy ogłoszenie zakończyło się sprzedażą postanowiono wykorzystać drzewa decyzyjne, predykcja liczby odpowiedzi również została wykonana przy użyciu drzewa decyzyjnego, a oszacowanie liczby wyświetleń wykonano metodą regresji liniowej. W celu programowej realizacji powyższych metod wykorzystano implementacje dostępne w pakiecie scikit–learn, tj. DecisionTree oraz LinearRegression. Powyższe rozwiązania wybrano metodą prób i błędów podczas której testowano implementację różnych algorytmów z pakiety scikit–learn. Przy wyborze algorytmów kierowano się wynikami z metod score powyższych implementacji (każdy algorytm ma metodę score zwracającą ocenę od $-\infty$ do 1) oraz wynikiem metody sklearn.metrics.auc dla testów sprzedano/niesprzedano.

Testy przeprowadzano na zbiorze treningowym dostarczonym przez organizatorów konkursu, który podzielono na następujące dwa podzbiory:

- podzbiór treningowy - od listopada 2016 do sierpnia 2017,
- podzbiór testowy - wrzesień 2017

Wynikiem przeprowadzonych testów było wybranie wykorzystanych algorytmów oraz ograniczenie zbioru uczącego - testy wykazały, że model dokonuje lepszego oszacowania gdy zbiór treningowy jest ograniczony do kilku miesięcy. Przyjęto, że model trenowany będzie na zbiorach od kwietnia do października 2017

2.2 Model konkursowy

Model konkursowy jest rozwinięciem modelu bazowego o następujące kolumny z plików z danymi:

- title,
- description,
- photo_sizes

Wartości z powyższych kolumn dodano jako kolejne cechy do zbioru uczącego i testowego, przy czym zmapowano je w następujący sposób: dla title i description obliczono liczbę użytych słów (przyjęto, że słowa oddzielone są spacją), a dla photo_sizes obliczono liczbę zdjęć.

3 Przetwarzanie danych

Jak już wcześniej wspomniano jedynymi szczególnymi krokami podjętymi podczas przetwarzania było mapowanie cech has_person i has_phone ze znaków t i f na odpowiednio 1 oraz 0. Kolejnemu przetworzeniu podlegały tytuły i opisy ogłoszeń gdzie postanowiono policzyć liczbę słów oraz rozmiary zdjęć, gdzie policzono liczbę dodanych zdjęć. Powyższe transformacje podjęto w celu zamiany wszystkich danych na wartości liczbowe akceptowane przez wykorzystane klasyfikatory.

4 Wyniki eksperymentalne

Wyniki eksperymentalne zaprezentowano w tabeli 1. Zastosowano tutaj następujące zbiory danych:

- uczący - kwiecień, maj, czerwiec, lipiec, sierpień 2017
- testowy - wrzesień 2017

W przypadku wysyłania danych na stronę konkursu zbiorem testowym jest zbiór za październik 2017. W związku z tym, dla tego przypadku rozszerzono także zbiór uczący o wrzesień 2017.

	zbiór walidacyjny				**test**
	L_v	L_r	L_s	L_{avg}	L_{avg}
model bazowy	0.50076	0.43147	0.42477	0.45233	0.46917
model konkursowy	0.5007	0.431	0.42477		0.44548

Tablica 1: Wyniki eksperymentalne.

5 Podsumowanie

Otrzymane wyniki konkursowe są lepsze od wyniku pliku z samymi zerami. Różnica jest nieznaczna, ale biorąc pod uwagę, że jest to średnia trzech parametrów z miliona ogłoszeń to raczej mało prawdopodobne jest aby była to wartość przypadkowa. Potwierdzeniem tego faktu może być to, że wykonano sześciokrotnego sprawdzenia danych na stronie konkursu (przy różnych zbiorach uczących) i za każdym razem wynik był lepszy niż ocena pliku z samymi zerami.

Reasumując, uzyskano dwa modele lepsze od samych zer. Widząc wyniki innych grup autorzy pracy są świadomi, że wiele jeszcze można poprawić w modelu konkursowym. W powyższych modelach nie wykorzystano stemmingu oraz lematyzacji. Nie użyto także danych dotyczących wyszukiwanych fraz na stronie internetowej. W celu rozbudowy modelu można także próbować dodawać kolejne cechy z plików dotyczących ogłoszeń. Inną możliwością poprawy oszacowania dokonywanego przez model jest także sprawdzenie innych algorytmów. Autorzy próbowali co prawda użyć między innymi implementacji SGDClassifier z pakietu sklearn, niestety czas przetwarzania tego klasyfikatora był tak długi dla zastosowanego modelu, że zrezygnowano z próby jego wykorzystania. Z kolei inne klasyfikatory uzyskiwały gorsze wyniki zwracane przez ich metodę score.

Należy tutaj zaznaczyć, że podczas prac nad modelem autorzy niniejszego tekstu nie wiedzieli, że organizator konkursu udostępnił funkcję obliczającą średni błąd. Właście dlatego skorzystano z metod score i auc. Prawdopodobnie to w połączeniu z być może błędną lub niedokładną interpretacją wyników powyższych metod spowodowało dobranie takich algorytmów.