Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Automatyki i Informatyki Stosowanej

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Systemy Informacyjno-Decyzyjne

Klasyfikacja użytkowników serwera WWW

Joanna Kolis
Numer albumu 295049

promotor
dr inż. Mariusz Kamola

WARSZAWA 2020

Klasyfikacja użytkowników serwera WWW

Streszczenie

Rola sieci Internet w pozyskiwaniu danych niezbędnych do prosperowania wielu sektorów zarówno biznesowych jak i naukowych nieustannie wzrasta. Dlatego bardzo istotny jest rozwój narzędzi pozwalających na wyodrębnianie informacji istotnych dla danego problemu. Jednym z ważniejszych zagadnień w zakresie eksploracji danych pochodzących z sieci www jest badanie sposobów użytkowania witryn internetowych przez użytkowników. Celem niniejszej pracy było znalezienie i wykorzystanie technik pozwalających na stworzenie modeli behawioralnych użytkowników serwera Wydziału Elektroniki i Technik Informacyjnych, które pozwolą na przewidywanie przyszłych aktywności. Wybrano do tego zadania metody bazujące na analizie skupień i redukcji wymiarowości. Dodatkowo przeprowadzono analizę szeregów czasowych i próbę prognozowania dziennej ilości zapytań. Na zakończenie badań posłużono się pewnymi analogiami do dziedziny przetwarzania języka naturalnego i zastosowano model konstruujący reprezentacje wektorowe danych.

Praca zawiera przegląd aktualnego stanu badań, podstawy teoretyczne wszystkich zastosowanych podejść oraz opis wykonywanych badań, a także wyniki przeprowadzonych eksperymentów.

Słowa kluczowe: eksploracja wykorzystania sieci, logi, modelowanie użytkowników, grupowanie, k-modes, redukcja wymiarowości, MCA, Word2Vec.

Classification of WWW server users

Abstract

The role of the Internet in obtaining data necessary for the prosperity of many sectors, both business and science, is constantly growing. Therefore, it is very important to develop tools that allow extracting information relevant to a given problem. One of the most important issues in the field of web mining is the analysis of users' usage of websites. The purpose of this work was to find and use techniques that allow creating behavioral models of the users of the Faculty of Electronics and Information Technology www server that will be able to predict future activities. Methods based on cluster analysis and dimensionality reduction were selected for this task. In addition, a time series analysis was performed and an attempt was made to forecast the daily number of queries. At the end of the study, some analogies to the field of natural language processing were used and a word embedding model was used.

The work contains a review of the current state of research, the theoretical basis of all approaches used and a description of the tests performed, as well as the results of conducted experiments.

Keywords: web usage mining, web logs, user modeling, clustering, k-modes, dimensionality reduction, MCA, Word2Vec

załącznik nr 3 do zarządzenia
nr 28 /2016 Rektora PW

Politechnika Warszawska

……..
miejscowość i data

……..
imię i nazwisko studenta
……..
numer albumu
…….…….
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wprowadzenie

1.1. Cel i zakres pracy

Celem niniejszej pracy jest analiza dziennika serwera www. Analiza ta ma doprowadzić do wyróżnienia typowych profili użytkowników na oraz do utworzenia modeli predykcyjnych aktywności poszczególnych grup. Dane do analizy pochodzą z serwera Wydziału Elektroniki i Technik Informacyjnych [1], w dalszej części pracy zwanego serwerem *studia*.

W rozdziale pierwszym zawarte jest krótkie wprowadzenie do problematyki tematu oraz przegląd aktualnego stanu wiedzy odnośnie procesu eksploracji wykorzystania sieci. W kolejnych rozdziałach przedstawiam metodologię i etapy przebiegu pracy zgodne z fazami wyróżnionymi w rozdziale 1.3. Rozdział drugi traktuje o przetwarzaniu danych w celu wyeliminowania wszelkich nieprawidłowości oraz wydobyciu jak największej ilości przydatnych informacji. Etap ten jest także potrzebny do przygotowania danych w sposób, w jaki będą je później przetwarzać kolejne algorytmy. Przedstawiona jest także charakterystyka danych otrzymanych do analizy. W rozdziale trzecim prezentuję podstawy teoretyczne zaproponowanych przeze mnie metod, wyjaśniam sposób ich działania oraz moją motywację do ich wyboru. Rozdział czwarty poświęcony jest przedstawieniu wykonanych przeze mnie badań oraz przeanalizowaniu otrzymanych rezultatów. Rozdział piąty stanowi podsumowanie wykonanych prac wraz z wnioskami i pomysłami na ulepszenie i kontynuację badań.

1.2. Wstęp do problematyki tematu

Gwałtownie rosnąca ilość źródeł informacji w internecie przy jednoczesnym braku uporządkowania oraz ciągły wzrost znaczenia sieci WWW w rozwoju wielu różnych branży, m.in. handlowej, usługowej, komunikacyjnej czy rozrywkowej, stwarza wysokie wymagania w zakresie metod i technologii pozwalających na uzyskiwanie i filtrowanie istotnych informacji. Ze względu na ograniczone zasoby sprzętowe i czasowe, analiza pozyskanych danych bez użycia zautomatyzowanych narzędzi stworzonych do tego celu jest praktycznie niemożliwa. Naprzeciw tym problemom wychodzi dziedzina eksploracji danych zwana eksploracją sieci (ang. *web mining*), czyli w najogólniejszym ujęciu przeszukiwanie sieci Internet i analiza pozyskanych danych w celu znalezienia użytecznych informacji [1]. *Web mining* można podzielić na trzy główne poddziedziny:

- eksploracja treści internetowych (ang. *web content mining*) - automatyczne wyszukiwanie i badanie źródeł informacji zawartych w sieci,
- eksploracja struktury sieci (ang. *web structure mining* - badanie powiązań w strukturze strony www, najczęściej za pomocą teorii grafów,
- eksploracja wykorzystania sieci (ang. *web usage mining*) - analizowanie wzorców nawigacji użytkowników.

[1] http://studia.elka.pw.edu.pl/

Największe zastosowanie w rzeczywistych problemach ma eksploracja wykorzystania sieci, bowiem pozawala na znalezienie wzorców zachowań użytkowników, które mogą następnie zostać użyte w celach:

- personalizacji - dostarczaniu użytkownikom indywidualnie dobranych ofert czy reklam,
- usprawnień działania systemu - optymalizacji transmisji i dystrybucji danych oraz równoważeniu obciążenia połączeń sieciowych,
- poprawienia bezpieczeństwa systemu - wykrywaniu prób włamań i ataków,
- modyfikacji strony - poprawieniu atrakcyjności witryny zarówno w zakresie wyglądu jak i funkcjonalności oraz optymalizacji struktury strony,
- analityki biznesowej (ang. *business intelligence*) - ewaluacji skuteczności kampanii reklamowych, wydajniejszego pozyskiwania klientów, maksymalizacji dochodów ze sprzedaży itp.

1.3. Eksploracja wykorzystania sieci

Eksploracja wykorzystania sieci skupia się na technikach, które są w stanie stworzyć portret behawioralny użytkownika, a następnie przewidywać jego przyszłe zachowania w czasie interakcji z witryną www. Standardowo proces eksploracji odbywa się w trzech fazach [2][3][4]:

- pozyskiwanie danych,
- *preprocessing*, czyli wstępne przygotowanie i obróbka danych,
- wykrywanie i analiza wzorców.

1.3.1. Pozyskiwanie danych

Dane wykorzystywane w eksploracji wykorzystania sieci mogą pochodzić z różnych źródeł; najczęściej rozważa się następujące trzy [5]:

- ze strony serwera:
  - — dzienniki serwera - tzw. logi, czyli pliki do których serwer dokonywał wpisu po każdym otrzymanym zapytaniu, uwzględniając zarówno informacje o samym zapytaniu jak i o kliencie, który je wykonał
  - — dane pozyskane od użytkownika, np. podczas procesu rejestracji, bądź wypełniania profilu
- ze strony klienta:
  - — pliki cookie, czyli ciągi tekstowe zapisywane przez przeglądarkę na żądanie serwera; mogą przechowywać informacje o trwającej sesji ale także o historii poprzednio odwiedzanych stron w ramach danej witryny
  - — zdalne agenty zaimplementowane w języku java lub javascript zagnieżdżone w stronach internetowych, które zbierają informacje o aktywności klienta
- ze strony serwera proxy - logi

1.3.2. Przetwarzanie i wstępna obróbka danych

Odpowiednie przygotowanie danych jest kluczowym etapem w procesie eksploracji, bowiem odkryte wzorce będą miały sens wyłącznie wtedy, gdy będą miały odniesienie w rzeczywistych modelach zachowań użytkowników. Aby dane dostarczone do analizy były jak najbardziej adekwatne, wstępne przetwarzanie danych odbywa się w kilku etapach [6]:

1. Czyszczenie danych - usunięcie danych niepoprawnych, redundantnych, eliminacja wszelkich szumów, bądź zapytań wykonanych bez wiedzy użytkownika (jak na przykład żądania zasobów typu .jpg, .gif czy .css), a także zapytan pochodzących od robotów internetowych.
2. Identyfikacja użytkowników - wyodrębnienie unikalnych użytkowników korzystających z witryny. Jest to kluczowe m.in. dla kolejnego etapu przetwarzania, czyli podziału na sesje. W celu identyfikacji użytkowników stosuje się wiele podejść [3], m.in:
    - przypisanie każdemu użytkownikowi unikalnego adresu IP pochodzącego z logów,
    - bazowanie na plikach cookie,
    - rozróżnianie użytkowników na podstawie rodzajów przeglądarek i wersji systemu operacyjnego w obrębie jednego adresu IP,
    - wykorzystanie pola user_id dostępnego w niektórych formatach logów.
3. Identyfikacja sesji - jest sposobem na odtworzenie wzoru nawigacyjnego użytkownika, bowiem sesję można zdefiniować jako ciąg zapytań wykonanych podczas pojedynczego dostępu do witryny. Celem podziału strumienia zapytań na sesje jest stworzenie logicznych grup referencji odnośnie zachowań dla każdego użytkownika [7]. Rozróżnia się trzy główne metody rekonstrukcji sesji [8]:
    - wyodrębnienie zestawu stron odwiedzonych w danym przedziale czasowym; najczęściej jest to 30 min,
    - bazowanie na czasie przeglądania pojedynczej strony - jeśli przekracza określony próg, kolejne zapytanie jest włączane do nowej sesji,
    - wykorzystanie topologii witryny i sprawdzenie, czy nadchodzące zapytanie jest osiągalne ze stron poprzednio odwiedzonych w danej sesji; jeśli żądana strona nie jest bezpośrednio połączona z żadną z nich, tworzy się nową sesję. Często zamiast bazować na topologii witryny w formie grafowej, korzysta się z pola *referrer* w logach serwera, które wskazuje URL strony, z której zostało wykonane dane zapytanie.

W przypadku braku dostępności danych pochodzących z mechanizmów śledzenia użytkownika, metody opisane w punktach 2. i 3. są narażone na szereg problemów [9] [10]:

- pojedynczy adres IP/kilka sesji - dostawcy usług internetowych zazwyczaj posiadają pulę serwerów proxy, za pośrednictwem których użytkownicy mają dostęp do in-

ternetu; potencjalnie kilku użytkowników korzystających z jednego serwera proxy może odwiedzać tę samą stronę w tym samym czasie.
- wiele adresów IP/pojedyncza sesja - niektórzy dostawcy usług internetowych lub narzędzia do ochrony prywatności losowo przypisują każde żądanie użytkownika do jednego z kilku adresów IP. W takim przypadku jedna sesja serwera może mieć wiele adresów IP,
- wiele adresów IP/pojedynczy użytkownik - użytkownik uzyskujący dostęp do Internetu z różnych komputerów będzie miał inny adres IP w zależności od sesji. Utrudnia to śledzenie powtarzających się wizyt tego samego użytkownika,
- wiele przeglądarek(systemów operacyjnych)/pojedynczy użytkownik - ponownie, użytkownik korzystający z więcej niż jednej przeglądarki bądź systemu operacyjnego, nawet na tym samym komputerze, pojawi się jako wielu użytkowników.

1.3.3. Wykrywanie i analiza wzorców

Na tym etapie przetworzone dane są analizowane w celu wydobycia wartościowych wzorców. Do wyszukiwania wykorzystuje się metody i algorytmy wywodzące się z innych dziedzin eksploracji danych, jak np. metody statystyczne i uczenie maszynowe. Najbardziej powszechne z nich wymienione zostały poniżej [3] [2]:

1. Analiza statystyczna - jest bardzo popularną techniką stosowaną do pozyskiwania wiedzy o użytkownikach danej witryny wykorzystującą statystyki opisowe (średnią, medianę, dominantę, częstotliwość itp.) w odniesieniu do różnych atrybutów. Stosuje się ją także to niskopoziomowego wykrywania błędów, takich jak znajdowanie najczęstszych nieautoryzowanych zapytań czy najczęściej występujących nieprawidłowych adresów URL. Analiza ta, pomimo swojej powierzchowności może prowadzić do usprawnień wydajności systemu, poprawy bezpieczeństwa czy wsparcia decyzji marketingowych.
2. Reguły asocjacyjne - pomagają znaleźć powiązania pomiędzy stronami, które najczęściej były odwiedzane wspólnie (w jednej sesji), mając na uwadze pewną graniczną wartość wsparcia dla reguły, przy czym strony te nie muszą być połączone hiperłączem.
3. Grupowanie (klasteryzacja) - metoda polegająca na podziale zbioru danych na grupy obserwacji, które różnią się między sobą, ale skupiają elementy jak najbardziej do siebie podobne. Wyróżnia się dwa typy klastrowania: grupowanie stron o podobnej treści oraz grupowanie użytkowników wykazujących podobne wzorce aktywności. Jest to podgrupa metod tzw. uczenia nienadzorowanego, zwanego też w niektórych źródłach klasyfikacją bezwzorcową [11].
4. Klasyfikacja (analiza dyskryminacyjna) - technika polegająca na przypisywaniu poszczególnych obserwacji do wcześniej zdefiniowanych klas. W dziedzinie eksploracji wykorzystania sieci jest to przydatne przy opracowywaniu profili użytkowników. Wymaga to wyodrębnienia i wyboru atrybutów, które najlepiej opisywać będą daną

klasę. Przykładowy profil mógłby wyglądać następująco: 40% użytkowników zamawiających produkt x, jest wieku 20-30 lat i żyje w południowej Polsce. Klasyfikacji można dokonać za pomocą algorytmów uczenia nadzorowanego, takich jak drzewa decyzyjne, maszyna wektorów nośnych (SVM) czy naiwny klasyfikator Bayesa

5. Wzorce sekwencji - technika polegająca na znajdowaniu pewnych wzorców sekwencji pomiędzy sesjami - czy po pewnym zestawie elementów zawsze następuje konkretny inny itp. Jest to podejście szczególnie przydatne w marketingu - pomaga przewidzieć przyszłe wzorce odwiedzin, dzięki czemu pozwala lepiej umieścić reklamy skierowane do konkretnych grup użytkowników

2. Przetwarzanie i wstępna obróbka danych

Wstępna obróbka danych może mieć duży wpływ na wydajność i efektywność zastosowanych później algorytmów, bowiem dane w surowej postaci niemal zawsze posiadają pewne niedoskonałości, szumy i braki. Zbiór danych należy scalić, oczyścić z danych niekompletnych, nieistotnych dla procesu modelowania i przekształcić do formatu ułatwiającego dalszy proces eksploracji. Niezbędne jest także zidentyfikowanie unikalnych użytkowników i sesji. Kroki te, a także szczegółowa specyfikacja danych, zostały szczegółowo opisane w kolejnych podrozdziałach. Dodatkowo na bieżąco stosowałam różne statystyki opisowe w celu zwizualizowania poszczególnych fragmentów i aspektów zbioru danych i aby lepiej zrozumieć jego strukturę.

2.1. Charakterystyka danych

Jednym z głównych źródeł danych służących do analizy ruchu na serwerze są jego dzienniki, czyli tak zwane logi. Główną zaletą logów w zastosowaniu do profilowania użytkowników jest to, że dane takie zostały sporządzone bez udziału żadnych czynników mogących wprowadzić stronniczość, a pozyskanie ich jest bardzo łatwe. Natomiast podstawową wadą jest to, że potrafią posiadać dużą ilość szumu i informacji zupełnie zbędnych, a wręcz przeszkadzających w prawidłowym modelowaniu użytkowników [12]. Logi muszą zatem zostać uważnie i w przemyślany sposób przygotowane do dalszej pracy. Dodatkowo nieścisłości wprowadzają techniki buforowania sieciowego, które w celu zmniejszenia opóźnień sieciowych, zapisują kopie żądanych stron przez pewien czas w pamięci podręcznej przeglądarki użytkownika, bądź na serwerze proxy. Jeśli żądana strona jest buforowana, serwer nie jest świadomy o nowym dostępie, który w konsekwencji nie jest wpisywany do dziennika.

2.1.1. Format danych

Jak już zostało wcześniej wspomniane, dane do analizy pochodzą z wydziałowego serwera studia. Są to logi z okresu wrzesień-grudzień 2018. Zapisy umieszczone zostały w 428 plikach tekstowych, w których każdy wiersz odpowiadał pojedynczemu zapytaniu do serwera. Format wpisów jest zbliżony do tzw. *Combined Log Format*, czyli ustandaryzowanego formatu plików tekstowych z logami. Przykładowy wiersz danych wygląda następująco:

```
5.184.hnU+2qCF/K0V.UENJp7w+cT - - [01/Sep/2018:12:01:43 +0200]
"GET /pub/18L/WEDT.A/ HTTP/1.1" 200 10512
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML,like Gecko) Chrome/68.0.3440.106 Safari/537.36
```

co odpowiada następującym informacjom:

```
adres_ip - - [data] ,,zapytanie'' kod_odpowiedzi rozmiar_odpowiedzi
[user_agent_string]
```

Gdzie:

- adres ip - adres IP klienta wysyłającego zapytanie; ostatnie 2 bajty są zaszyfrowane
- data – data wykonania zapytania w formacie:
  dzień/miesiąc/rok:godziny:minuty:sekundy±strefa czasowa
- zapytanie - składa się z użytej przez klienta metody, treści wysłanego zapytania oraz użytego protokołu
- kod odpowiedzi – kod odpowiedzi HTTP wysłany przez serwer w odpowiedzi na zapytanie
- rozmiar odpowiedzi - ilość bajtów zwróconych jako odpowiedź
- user agent string - nagłówek pozwalający przeglądarce na identyfikację typu programu klienckiego oraz używanego systemu operacyjnego

Zasadniczą różnicą pomiędzy powyższym formatem danych, a *Combined Log Format* jest brak pola *referer*, czyli adresu URL, z którego użytkownik został przekierowany wysyłając dane zapytanie. Rzeczą wartą odnotowania jest także format adresów IP, w których dwa ostatnie bajty zostały zaszyfrowane. Szyfrowanie odbywało się za pomocą algorytmu deterministycznego (bez dodatku soli), zatem poprawne będzie założenie, że adresy składające się z takich samych ciągów znaków w rzeczywistości odnoszą się do tego samego adresu IP.

Łącznie otrzymano 9197048 pojedynczych wpisów.

2.2. Wczytanie, konwersja i czyszczenie danych

Pierwszym krokiem było wczytanie logów i ich konwersja do formatu *.csv* oraz usunięcie wierszy zawierających nieprawidłowe dane (np. w niepoprawnym formacie) bądź z brakującymi wartościami. Wierszy takich było zaledwie 171, co stanowi niecałe 0.02 promila całych danych, zatem można uznać, że logi dostarczone zostały w stanie spójnym i kompletnym. Następnie dokonano konwersji daty na format rrrr-mm-dd HH:MM:SS±zone (np. 2018-09-23 00:00:29+02:00). Kolejno, z pola user_agent_string wydobyto informacje o używanej przeglądarce oraz systemie operacyjnym. Następnym krokiem było wydobycie z treści zapytania przedmiotu, jakiego dotyczy (o ile oczywiście zapytanie dotyczyło jakiegoś przedmiotu). Przeanalizowałam w tym celu możliwe adresy URL z serwera *Studia* odnoszące się do przedmiotów i przy pomocy wyrażeń regularnych wydobyłam ich nazwy z zapytań. Adresów wyszukiwałam ręcznie, bowiem nie udało mi się za pomocą żadnego programu zmapować wszystkich możliwych adresów, zatem wyniki analizy są obarczone błędem, który może wynikać z nie odnalezienia przeze mnie jakiegoś wzorca konstruowania adresów. Następnie usunęłam wszystkie zapytania, które nie były związane z żadnym

przedmiotem. Dodatkowo wyeliminowałam te przedmioty, do których odwoływano się mniej niż 10 razy i zawęziłam w ten sposób zbiór danych do 2331858 rekordów.

2.2.1. Boty internetowe

Jako że celem pracy jest modelowanie rzeczywistych użytkowników strony, kolejnym ważnym krokiem było wyeliminowanie wszelkich nie-ludzkich aktywności. Mowa tu głównie o wszelkich botach internetowych, zwanych także crawlerami, czyli programach zbierających informacje o strukturze i treści stron internetowych. Ich celem może być indeksowanie, poszukiwanie błędnych lub wygasłych linków, tworzenie baz stron internetowych oraz wiele innych. Aby wykluczyć zapytania wykonane przez boty, usunęłam z danych wszystkie adresy IP, które chociaż raz odniosły się do pliku *robots.txt*. Jest to specjalny plik, umieszczany na serwerze w celu poinformowania botów, które obszary serwisu może, a których nie powinien przeglądać. Dodatkowo, aby wyeliminować boty, które z jakiegoś powodu nie zażądały tego pliku, przy pomocy strony *User Agent String.Com* [2] stworzyłam listę najpopularniejszych crawlerów a także walidatorów i bibliotek, które można zidentyfikować przy użyciu user agent string i usunęłam wszystkie zapytania wykonane przez którykolwiek z nich.

2.2.2. Akcje

Ważną informacją potrzebną w procesie modelowania aktywności użytkowników jest także akcja jaka dotyczyła danego przedmiotu. Na podstawie analizy przeprowadzonej podczas ekstrakcji przedmiotów wyodrębniłam następujące działania, jakie użytkownik może podjąć na danym przedmiocie:

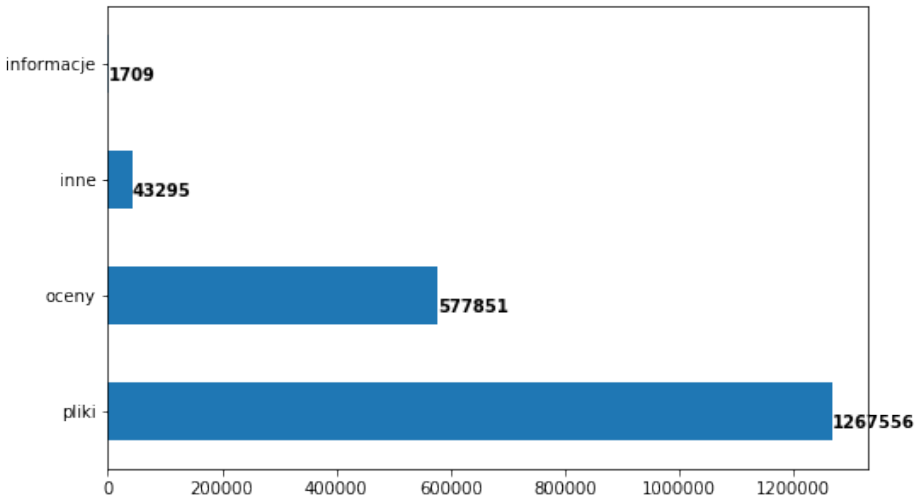
- wyświetlenie informacji o przedmiocie w systemie ERES
- wyświetlenie materiałów zamieszczonych przez prowadzącego przedmiot
- wyświetlenie ocen z przedmiotu
- inne - najczęściej odczytanie wyników ankiet

Początkowo podział uwzględniał także sprawdzenie stanu zapisu na dany przedmiot, jednak z uwagi na zmianę systemu do obsługi studiów i przeniesienie rejestracji na przedmioty z ERES na USOS, zapytania tego typu właściwie się nie pojawiały.

Jako że wszystkie wymienione wyżej akcje ograniczają się do odczytywania informacji z serwera, zawęziłam zbiór danych do zapytań wykonanych za pomocą metody GET, która służy właśnie do pobierania zasobów. W ten sposób usunęłam łącznie 6366864 rekordy, pozostawiając tym samym do analizy 1890411.

Rozkład wszystkich akcji w zbiorze danych został przedstawiony na Rysunku 1

[2] `http://www.useragentstring.com/pages/useragentstring.php`



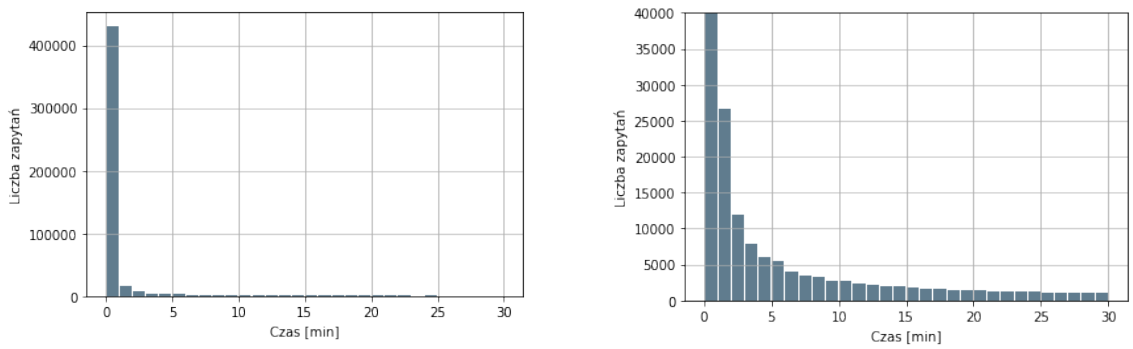
Rysunek 1. Liczba poszczególnych akcji

2.3. Identyfikacja użytkowników i podział na sesje

Do identyfikacji unikalnych użytkowników zastosowałam metodę opartą na rozróżnianiu ich poprzez adres IP, rodzaj przeglądarki i system operacyjny. Jeśli w ramach jednego adresu IP zapytania były wykonywane z różnych systemów operacyjnych bądź przeglądarek, zakładałam istnienie nowego użytkownika. Jak zostało wspomniane w rozdziale 1., nie jest to metoda pozbawiona błędów, jednak nie dysponowałam danymi pozwalającymi na bardziej wnikliwą analizę, jak np. pliki cookie. Za pomocą tej techniki wyodrębniłam 69618 unikalnych użytkowników.

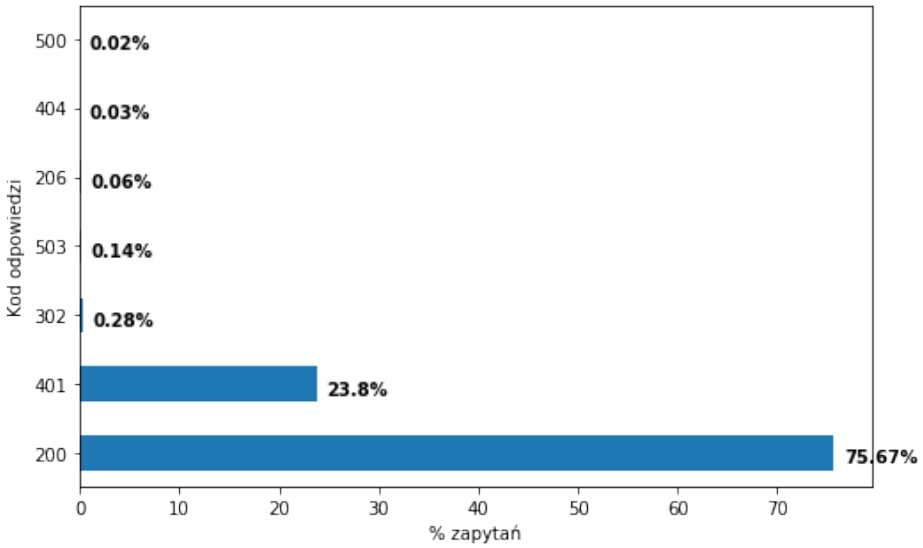
Następnym krokiem była identyfikacja sesji. Analizując sposoby wyodrębniania sesji przedstawione w rozdziale 1., zdecydowałam się zastosować nieco zmodyfikowane podejście oparte o czas przeglądania stron. Nie byłam bowiem w stanie skorzystać z metody opartej o pole referrer, gdyż brakuje go w logach z mojego zbioru danych, natomiast metoda polegająca na wyodrębnieniu sesji jako zestawu stron odwiedzonych w zadanym przedziale czasowym bazuje na zbyt sztywnym założeniu. Identyfikowałam zatem sesje jako ciąg zapytań, w którym odległości między kolejnymi zapytaniami od tego samego użytkownika są nie większe niż jakiś określony przedział czasu. Aby móc empirycznie wybrać taki przedział czasu sporządziłam rozkład czasu pomiędzy kolejnymi zapytaniami od tego samego użytkownika. Histogram w całości oraz z przyciętą skalą na osi y, w celu poprawienia czytelności zilustrowany został na Rysunku 2. Przedstawione wykresy nie pozwalają jednak jednoznacznie wyznaczyć takiego przedziału czasu, po którym ilość zapytań gwałtownie maleje.

Przy okazji chciałam zbadać jaki wpływ na rozkład odstępów czasowych mają zapytania niepoprawnie wykonane - czyli z kodem błędu świadczącym o błędzie występującym ze strony klienta, a zatem z przedziału 400-499 (w dalszej części pracy będę takie zapytania nazywała w skrócie błędnymi). Zapytań takich jest powiem ponad 24%, co widać



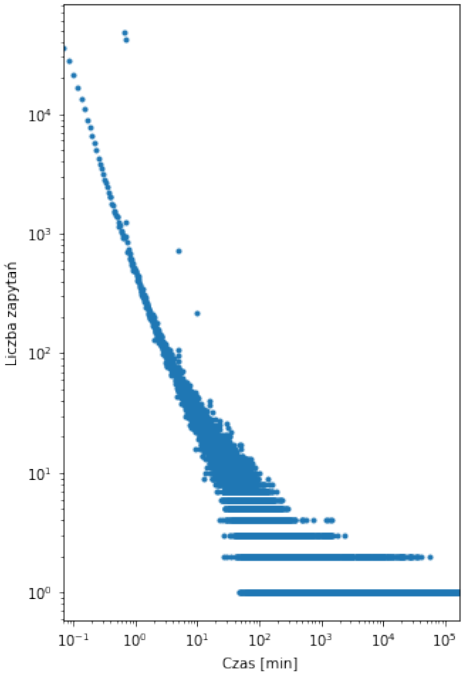
Rysunek 2. Rozkład czasu pomiędzy kolejnymi zapytaniami.

sporządzonym wykresie widocznym na Rysunku 3, który przedstawia w jakim procencie zapytań został zwrócony poszczególny kod odpowiedzi.

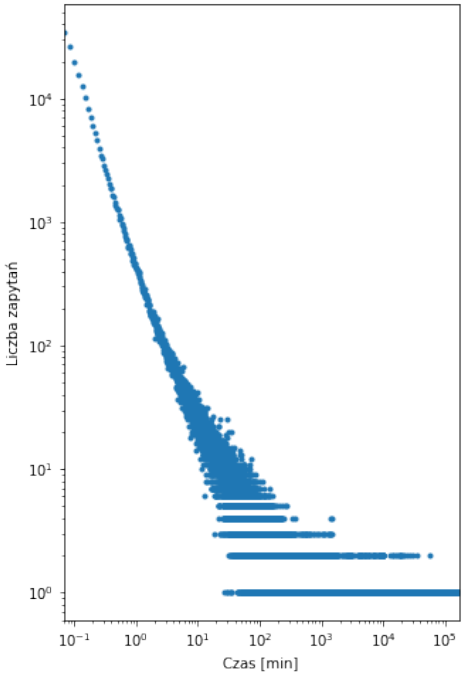


Rysunek 3. Stosunek ilości zapytań, w których zwrócone zostały poszczególne kody odpowiedzi

Zdecydowałam się porównać odstępy czasowe dla wszystkich zapytań oraz dla zapytań wyłącznie z kodem 200. Dane przedstawiłam w postaci punktowej na skali logarytmicznej na Rysunkach 4 i 5. Zapytania z kodami błędu z rodziny 400 wprowadziły tylko lekki szum widoczny na Rysunku 4 w postaci punktów poza głównym konturem wykresu, jednak z pewnością nie miały dużego znaczenia w wyglądzie rozkładu, który wykazuje cechy rozkładu potęgowego.
Jako że analiza wykonanego rozkładu nie pozwoliła wyróżnić wyraźnego odstępu czasu, dla którego ilość zapytań drastycznie maleje, jako wartość rozgraniczającą dwie sesje przyjęłam arbitralnie wartość **10 min**.

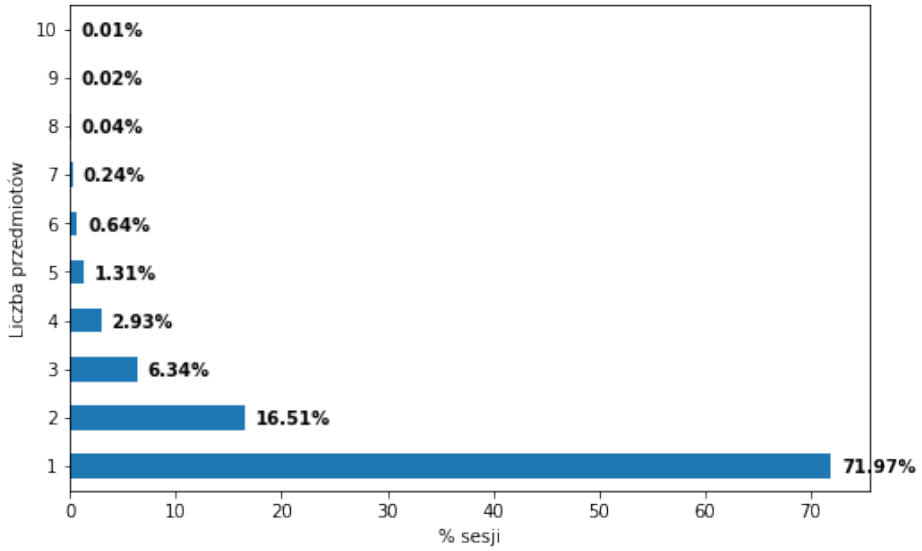


Rysunek 4. Czas pomiędzy kolejny zapytaniami dla wszystkich kodów odpowiedzi



Rysunek 5. Czas pomiędzy kolejny zapytaniami dla kodów odpowiedzi bez 401.

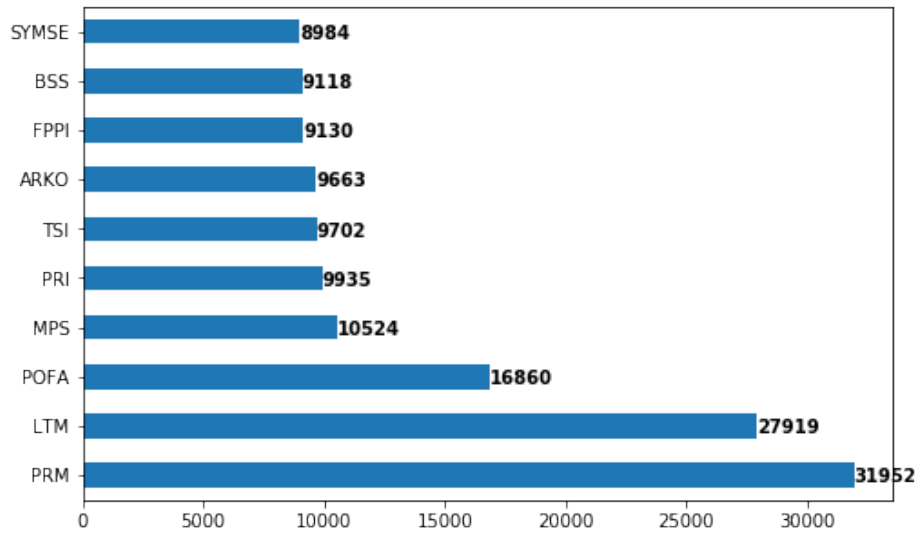
Przy okazji sporządzania powyższych wykresów dostrzegłam pewne wzorce zachowań, których źródłem z pewnością nie był człowiek. Do wzorców takich zalicza się na przykład wykonanie serii zapytań w czasie krótszym niż 1 sekunda. Zdecydowałam się wyeliminować wszystkie zapytania wykonane przez adresy IP, z których dokonano więcej niż 3 zapytań w ciągu jednej sekundy, ponieważ ciężko jest w sposób automatyczny znaleźć wszystkie schematy działań nie będące działaniami bezpośrednio ludzkimi, np zapytania wykonywane zawsze w określonym odstępie czasu, lub zawsze o określonej godzinie. Wyeliminowanie adresów IP, z których dokonywano podejrzanych aktywności ma na celu poprawienie jakości danych, jak i późniejszych predykcji. Spowodowało to jednak znaczne zmniejszenie rozmiaru zbioru danych, bowiem aż 641239 zapytań z 4660 unikalnych adresów IP zostało wykonanych w ciągu jednej sekundy. Stosując zasadę wspomnianą powyżej, usunęłam łącznie aż 1410425 rekordów.
Po wykonaniu powyższej modyfikacji podzieliłam cały zbiór danych na 103838 sesji. Przeanalizowałam także ilu przedmiotów zazwyczaj dotyczyły sesje: Jak widać, ponad 70% sesji dotyczyło tylko jednego przedmiotu.
Dodatkowo sporządzono statystyki opisowe dla przedmiotów, systemów operacyjnych i przeglądarek. Liczność poszczególnych atrybutów przedstawione jest w Tabeli 1, natomiast Rysunki 7, 8 oraz 9 przedstawiają częstości występowania w zbiorze danych poszczególnych wartości. Dla czytelności przedstawiono dziesięć najpopularniejszych wartości z każdej kategorii.



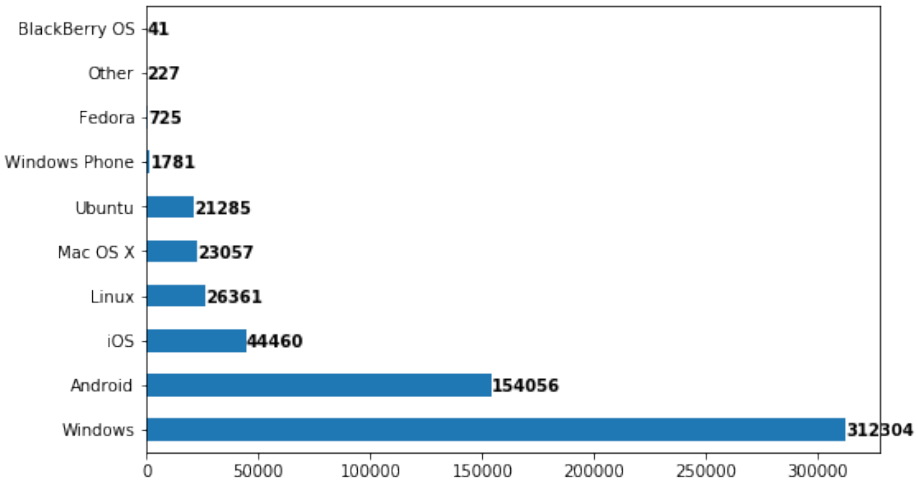
Rysunek 6. Procentowe przedstawienie sesji na podstawie liczby odwiedzonych w nich przedmiotów

atrybut	liczność
przedmiot	421
system operacyjny	19
przeglądarka	64

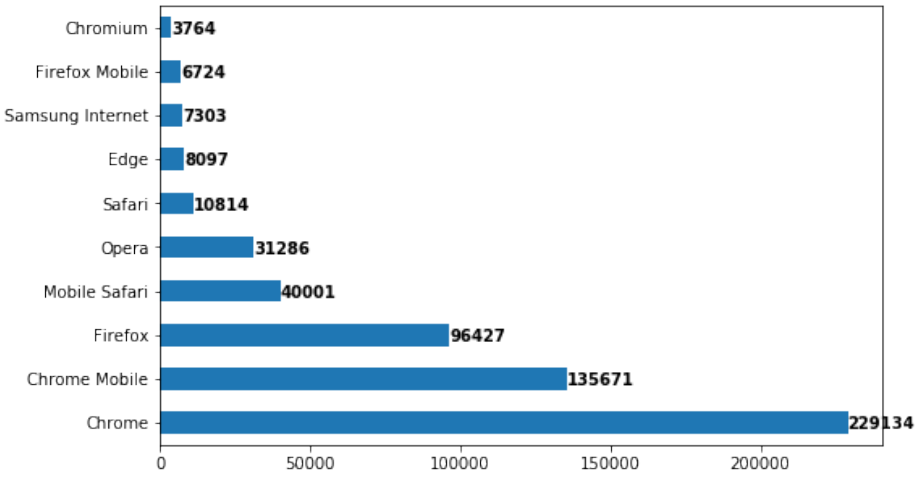
Tabela 1. Liczność atrybutów przedmiot, system operacyjny i przeglądarka



Rysunek 7. Najpopularniejsze przedmioty



Rysunek 8. Najpopularniejsze systemy operacyjne



Rysunek 9. Najpopularniejsze przeglądarki

Jak widać przedmioty są bardzo liczną kategorią, co może utrudniać proces wyszukiwania wzorców, a następnie ich analizy i interpretacji. Zgodnie z intuicją, najczęściej odwiedzane przedmioty są obowiązkowymi przedmiotami na studiach pierwszego stopnia dla kierunków Informatyka, Elektronika, Telekomunikacja oraz Automatyka i robotyka. Są to m.in podstawy programowania (PRM, PRI), logika i teoria mnogości (LTM), pola i fale (POFA), metody probabilistyczne i statystyka (MPS) czy teoria sygnałów i informacji (TSI). Nie jest zaskoczeniem także rozkład użytkowania systemów operacyjnych, gdzie najpopularniejszy jest system Windows oraz przeglądarek z wiodącym Chrome.

3. Wykrywanie i analiza wzorców - zastosowane podejścia

Podstawowym celem postawionym w niniejszej pracy było modelowanie użytkowników serwera WWW. Na podstawie przeglądu literatury przedstawionego w rozdziale 1., zdecydowałam się początkowo zastosować metody oparte na grupowaniu użytkowników na podstawie ich aktywności. Pierwszym kryterium dokonania podziału miało być źródło zapytań, a dokładniej system autonomiczny, czyli zbiór prefiksów IP, które są pod wspólną jurysdykcją i w których utrzymywany jest wspólny schemat trasowania, dzięki czemu do analizy można by dodać czynnik geograficzny. Niestety po przestudiowaniu sposobu określania systemu autonomicznego dla danego adresu IP, okazało się, że numery systemów autonomicznych zapisywane są w postaci 4-bajtowej, zatem prefiksy należące do różnych systemów autonomicznych mają często dużo węższe zakresy, niż dwa wspólne pierwsze bajty. Z tego powodu niemożliwe staje się jednoznaczne uzyskanie informacji o systemie autonomicznym dla adresu IP z zaszyfrowanymi dwoma bajtami.

Skupiłam się zatem na znalezieniu atrybutów sesji, które najlepiej ze sobą korelują, aby na tej podstawie dokonać podziału użytkowników. Początkowym założeniem była idea, aby wszystkie kroki prowadzące do wyróżnienia najczęściej współwystępujących atrybutów były wytłumaczalne i interpretowalne, dlatego poszukiwano metod, które nie opierają się na zastosowaniu sieci neuronowych. Jako że zbiór danych składa się z atrybutów przyjmujących dyskretne wartości z pewnego skończonego zbioru (tzw. zmienne kategoryczne), konieczne było znalezienie algorytmów, które będą w stanie takim rodzajem danych operować. Specyfika problemu zakłada, że algorytmy te będą działać w trybie klasyfikacji bez nadzoru, bowiem nie dysponujemy żadnymi etykietami, ani z góry określonymi profilami, które pozwoliłyby dokonać podziału. Poszukiwałam algorytmu, który albo będzie operował na danych kategorycznych i od razu je pogrupuje, albo przekształci je do postaci liczbowej, którą inne algorytmy będą potrafiły zinterpretować.

Mimo braku tej techniki w podstawowych metodach wykrywania wzorców, zdecydowałam się zacząć badania od analizy szeregów czasowych. Użyłam do tego modelu zdolnego do rozkładu szeregu czasowego na poszczególne składowe. Miałam na celu sprawdzenie czy model taki jest w stanie dopasować jakąś krzywą do dziennej liczby zapytań, a także przeprowadzić próbę prognozowania liczby zapytań w niedalekiej przyszłości.

Kolejne zastosowane przeze mnie podejście opiera się o analizę skupień, czyli zestaw metod grupujących elementy zbioru na podstawie metryki podobieństwa, w celu osiągnięcia jak najbardziej jednorodnych klas. Taka procedura umożliwia zredukowanie dużej liczby obserwacji do zdecydowanie mniejszej liczby kategorii i dzięki temu możliwe jest odnalezienie pewnych ukrytych wzorców.

Trzecie podejście nie jest techniką grupowania samą w sobie, jednak pozwala na zmniejszenie wymiarowości danych, przez co umożliwia przedstawienie ich jako punktów w dwuwymiarowej przestrzeni, które przejawiają skłonności do skupiania się w grupy podobnych obserwacji.

Czwarte podejście znacznie różni się od klasycznych technik eksploracji wykorzystania sieci, bowiem opiera się na analogii do metod stosowanych w przetwarzaniu języka naturalnego.

3.1. Wykorzystane technologie

Ze względu na charakterystykę problemu oraz moje doświadczenie w pracy z tym językiem, do implementacji wszystkich rozwiązań przedstawionych w niniejszej pracy użyłam języka programowania Python. Jest to język ogólnego przeznaczenia, który ma bogate zaplecze bibliotek służących do szeroko pojętych obliczeń naukowych i przetwarzania danych. Bardzo pomocnym narzędziem są także tzw. *notatniki* (ang. notebooks) - czyli środowiska obliczeniowe osadzone w formie aplikacji webowej, które służą za interpreter Pythona. Pozwalają one na niezależnie wykonywanie bloków kodu, bez potrzeby wykonywania całości skryptu od początku w celu powtórzenia części obliczeń, a także na bardzo przejrzystą dokumentację poszczególnych fragmentów. Do znacznej większości operacji na danych użyłam niezwykle pożytecznego pakietu *pandas*, który zapewnia szybkie i efektywne struktury danych oraz operacje na nich.

3.2. Analiza szeregów czasowych

Szereg czasowy jest to ciąg uporządkowanych danych, które próbkowane są z regularnym krokiem czasowym. Analiza szeregów czasowych polega na wyodrębnieniu i znalezieniu w danych znaczących wzorców i zależności opartych na zmiennej czasowej. Celem takiego procesu może być analiza sama w sobie jako jeden z etapów lepszego zrozumienia danych ale także predykcja przyszłych wartości opartych na zapisach historycznych. Jako że jednym z aspektów modelowania ruchu na serwerze WWW jest prognozowanie tego ilu zapytań HTTP możemy się spodziewać w zadanym przedziale czasowym, postanowiłam zastosować taką analizę. Do przygotowania jej użyłam procedury Prophet [13], która bazuje na dekompozycji szeregów czasowych do trzech głównych komponentów: trendu, sezonowości i wydarzeń nieregularnych. Użyty model addytywny łączy wymienione składowe w następujący sposób:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

gdzie:

- $g(t)$ to funkcja trendu, która modeluje nie-okresowe zmiany, czyli ogólna tendencję wzrostową lub spadkową (niekoniecznie liniową),

- $s(t)$ reprezentuje zmiany sezonowe (np. tygodniowe bądź dniowe),

- $h(t)$ odpowiada za efekty wydarzeń nieregularnych, jak np. świąt - te użytkownik musi sam wprowadzić,

- $\epsilon(t)$ to błąd uwzględniający wszelkie nietypowe zmiany, których model nie uwzględnił

3.3. K-Modes

Kolejną zastosowaną metodą był algorytm *k-modes* [14]. Jest to stosunkowo mało popularny algorytm, oparty na centroidach i będący rozszerzeniem algorytmu k-means (k-średnich) dla danych kategorycznych. W algorytmach opartych na centroidach, grupowanie polega na podzieleniu populacji na zadaną liczbę klas. Każda klasa, czy też inaczej klaster, jest reprezentowana przez wektor centralny. Dla k-means wektor taki jest obliczany jako średnia arytmetyczna współrzędnych wektorów przydzielonych do danego klastra. W przypadku k-modes do reprezentacji tendencji centralnych poszczególnych klastrów zamiast średnich używana jest moda (dominanta).
Niech $\boldsymbol{S}$ będzie zbiorem obserwacji opisywanych przez m dyskretnych atrybutów $A_1, ..., A_m$.

Definicja 1. *Modą zbioru* $\boldsymbol{S} = \{X_1, ...X_n\}$ *będzie wektor* $Q = [q_1, ..., q_m]$*, który minimalizuje*

$$D(\boldsymbol{S}, Q) = \sum_{i=1}^{n} d(X_i, Q_i) \tag{2}$$

Zatem modą q_i jest najczęściej występująca wartość w S jako $i - ty$ atrybut, co zostało zobrazowane w Tabeli 2

Tabela 2. Modą przedstawionego poniżej klastra obserwacji jest wektor $Q = [A, B, A]$

	A_1	A_2	A_3
1	A	B	A
2	A	C	A
3	C	B	B
Q	A	B	A

Miarą podobieństwa, którą minimalizuje się w celu dokonania przydziału obserwacji do klastrów jest tzw. miara odmienności, będąca sumą niedopasowań poszczególnych atrybutów pomiędzy obserwacjami, opisywana wzorem:

$$d(X, Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \tag{3}$$

gdzie

$$\delta(x_j, y_j) = \begin{cases} 0, & \text{jeśli } (x_j = y_j) \\ 1, & \text{w przeciwnym wypadku} \end{cases} \tag{4}$$

Funkcja kosztu dla zadania grupowania n obiektów w k klastrów zgodnie z powyższą notacją sprowadza się do:

$$P(W, \boldsymbol{Q}) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{li} d(X_i, Q_i) \tag{5}$$

z warunkami

$$w_{li} \in \{0, 1\}, \qquad 1 \leq l \leq k, \qquad 1 \leq i \leq n \tag{6}$$

$$\sum_{l=1}^{k} w_{li} = 1, \qquad 1 \leq i \leq n \tag{7}$$

gdzie $W = [w_{li}]$ jest macierzą podziału o rozmiarze $n \times k$.

Algorytm k-modes można zapisać w następujący sposób:

1. Wybierz k obiektów jako początkowe centra (mody) klastrów
2. Wyznacz miarę odmienności między każdym obiektem i każdym centroidem
3. Przydziel obiekt do klastra, dla którego miara 3 jest najmniejsza
4. Wyznacz nowe mody dla klastrów
    - jeśli mody się nie zmieniły, bądź wykonano już maksymalną założoną liczbę iteracji - zakończ
    - w przeciwnym wypadku idź do kroku 2.

Algorytm ten jest jest bardzo efektywny, wykazano bowiem eksperymentalnie [14], że osiąga zbieżność z liniową złożonością czasową. Nie jest jednak pozbawiony tych samych wad, z którymi borykają się wszystkie algorytmy oparte na centroidach. Jest to przede wszystkim fakt, że uzyskane rozwiązania są jedynie lokalnie optymalne, bowiem algorytmy te są wrażliwe na dobór punktów startowych, zatem jakość rezultatów zależy w pewnym stopniu od czynnika losowego. Istnieje kilka podejść odnośnie wyboru początkowych centroidów [15], jednak każda z nich jest obarczona pewnymi niedoskonałościami. Dodatkowym utrudnieniem jest konieczność ustalenia liczby klastrów a priori, co może prowadzić do błędnych interpretacji.

3.4. Multiple Correspondence Analysis

Aby znaleźć w danych wzorce, których istota może być subtelniejsza niż samo współwystępowanie przedmiotów, zdecydowałam się użyć metody pozwalającej na zmniejszenie wymiarowości danych. Metodą tą był algorytm MCA [16](ang. *Multiple Correspondence Analysis* - można przetłumaczyć jako *wielowymiarowa analiza korespondencji*). MCA jest używany do analizy zbioru obserwacji opisywanych przez zmienne nominalne i służy do znajdowania powiązań pomiędzy obserwacjami z wielowymiarowej perspektywy oraz badania podobieństw między zmiennymi i badania korelacji między kategoriami stosując redukcję wymiarowości. Pozwala to przedstawić dane wejściowe w takiej podprzestrzeni i z takim układem współrzędnych, aby jego osie maksymalizowały wariancję

przedstawianych danych. Metoda ta może być rozumiana jako analogiczna do analizy składowych głównych (PCA - ang. *Principal Component Analysis* [17]) lecz dla zmiennych kategorycznych zamiast ciągłych lub za rozszerzenie analizy korespondencji (CA - ang. *Correspondence Analysis* [18]) dla przypadku z więcej niż dwoma zmiennymi.

Najbardziej klasycznym sposobem przeprowadzania MCA jest zastosowanie CA do tzw. *indicator matrix* $\boldsymbol{Z}$, czyli macierzy kodującej w sposób binarny wszystkie możliwe wartości atrybutów dla rozważanych obiektów [19]. Zakładając, że zbiór danych składa się z K obserwacji opisywanych przez Q atrybutów, gdzie każdy atrybut może przyjąć J_q wartości, $\boldsymbol{Z}$ będzie rozmiaru $K \times J$. gdzie

$$J = \sum_{q=1}^{Q} J_q \tag{8}$$

Macierz taka posiada $N = K * Q$ niezerowych elementów. Co istotne, taki sposób kodowania zakłada, że każdy wiersz sumuje się zawsze do wartości Q. Dla mojego zbioru danych, przykładowe kodowanie dla $Q = 2$, $K = 6$, $J_1 = 4$ i $J_2 = 3$, a zatem $J = 7$ wyglądałoby w sposób przedstawiony w Tabeli 3:

Tabela 3. Kodowanie danych do postaci indicator matrix.

sesja	przedmiot	akcja
1	TASS	pliki
2	SNR	oceny
3	SOI	pliki
4	SNR	pliki
5	SOI	oceny
6	TASS	oceny

$\Longrightarrow$

	przedmiot			akcja		
	TASS	SNR	SOI	pliki	oceny	
1	1	0	0	1	0	Q=2
2	0	1	0	0	1	Q=2
3	0	0	1	1	0	Q=2
4	0	1	0	1	0	Q=2
5	0	1	0	0	1	Q=2
6	1	0	0	0	1	Q=2
	$L_1 = 2$	$L_2 = 3$	$L_3 = 1$	$L_4 = 3$	$L_5 = 3$	

Taki sposób przekształcenia nie umożliwia jednak przedstawienia sytuacji, w której w danej sesji odwiedzono więcej niż jeden przedmiot. Aby uwzględnić takie przypadki, nie wystarczy zakodować każdego przedmiotu jako nowej zmiennej binarnej, bowiem wtedy poszczególne wiersze tabeli nie będą sumować się do pożądanej wartości Q. Aby obejść ten problem, potraktowałam każdą wartość atrybutu przedmiot jako nową kategorię, która może przyjąć dwie wartości: tak lub nie, które mówią o tym czy w danej sesji przedmiot został odwiedzony czy nie. Przekształcenie tak sformułowanych danych wyglądałaby następująco (dla uproszczenia pomijam atrybut akcji):

Rozważając macierz $\boldsymbol{Z}$ wiersz po wierszu bądź kolumna po kolumnie, otrzymujemy wektory, które możemy interpretować jako współrzędne poszczególnych obserwacji bądź atrybutów w wysoko wymiarowej przestrzeni $\mathbb{R}^{J \times K}$, zatem możemy je rozważać jako zbiór punktów. Poniżej przedstawiam procedurę CA zastosowaną do macierzy $\boldsymbol{Z}$ mającą

Tabela 4. Poprawione kodowanie.

sesja	przedmiot
1	TASS
1	SNR
2	SOI
2	TASS

$\Longrightarrow$

	przedmiot						
	TASS		SOI		SNR		
	T	N	T	N	T	N	
1	1	0	1	0	0	1	Q=3
2	1	0	0	1	1	0	Q=3
	L_1=2	L_2=0	L_3=1	L_4=1	L_5=1	L_6=1	

na celu projekcję oryginalnych zbiorów punktów do przestrzeni o znacznie mniejszej wymiarowości, które następnie będzie można z łatwością zwizualizować.

3.4.1. Correspondence Analysis

Macierz $\boldsymbol{Z}$ należy przekształcić do macierzy korespondencji, zwanej też macierzą prawdopodobieństwa, bowiem wszystkie elementy sumują się do 1:

$$\boldsymbol{P} = \frac{1}{N}\boldsymbol{Z}, \quad \boldsymbol{P} = \{p_{ij}\} \tag{9}$$

oraz wyznaczyć tzw. masy dla rzędów i kolumn, czyli wektory przyjmujące jako wartości odpowiednio sumę zmiennych w poszczególnych wierszach lub kolumnach:

$$\begin{gathered} r_i = \sum_{j=1}^{K} p_{ij} \quad c_j = \sum_{j=1}^{J} p_{ij} \\ \text{to jest:} \quad \boldsymbol{r} = \boldsymbol{P}\mathbf{1} \qquad \boldsymbol{c} = \boldsymbol{P}^T\mathbf{1} \end{gathered} \tag{10}$$

gdzie $\mathbf{1}$ to wektor złożony z jedynek o długości dopasowanej do jego użycia; stąd pierwszy $\mathbf{1}$ ma wymiar $1 \times K$, a drugi $J \times 1$. Do dalszych obliczeń niezbędne będą także macierze diagonalne mas wierszy i kolumn:

$$\boldsymbol{D}_r = diag(\boldsymbol{r}) \quad oraz \quad \boldsymbol{D}_c = diag(\boldsymbol{c}) \tag{11}$$

Aby dokonać projekcji zbioru danych na podprzestrzeń o mniejszej wymiarowości należy najpierw wyznaczyć macierz ustandaryzowanych rezyduów:

$$\boldsymbol{S} = \boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{Z} - \boldsymbol{r}\boldsymbol{c}^T)\boldsymbol{D}_c^{-\frac{1}{2}} \tag{12}$$

a następnie wykonać dekompozycję według tzw. rozkładu według wartości osobliwych (SVD – ang. *Singular Value Decomposition*) [20]:

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T \tag{13}$$

gdzie $\boldsymbol{V}, \boldsymbol{U}$ to macierze ortogonalne: $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, natomiast $\boldsymbol{\Delta}$ to diagonalna macierz

wartości osobliwych, a $\mathbf{\Lambda} = \mathbf{\Delta}^2$ to macierz wartości własnych; $\mathbf{\Lambda} = \{\lambda_i\}$, gdzie wartości ułożone są w porządku malejącym.

Macierze te służą do wyznaczenia *factor scores*, czyli współrzędnych rzędów i kolumn w nowej przestrzeni, które wyrażają się następująco:

$$\boldsymbol{F} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{\Delta U} \quad oraz \quad \boldsymbol{G} = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{\Delta V} \tag{14}$$

Kolumny tak uzyskanych macierzy U i V są wektorami współrzędnych punktów reprezentujących odpowiednio obserwacje i atrybuty w kolejnych wymiarach. Każdemu i-temu wymiarowi przypisana jest wartość własna λ_i. Wartości te są proporcjonalne do ilości wariancji punktów początkowych wyrażonej (wyjaśnionej) przez dany wymiar. Suma wszystkich wartości własnych opisuje tzw. całkowitą inercję, będąca wielowymiarowym rozszerzeniem konceptu wariancji [21].

Możliwa jest także projekcja punktów, które nie zostały użyte do tworzenia nowych wymiarów. Są to tzw. dane uzupełniające (ang. *supplementary data*), a ich współrzędne są przewidywane tylko na podstawie wykonanej już analizy na pozostałych (tzw. aktywnych) zmiennych.

3.4.2. Interpretacja

Interpretacja wyników MCA zazwyczaj odbywa się za pomocą analizy odległości pomiędzy poszczególnymi punktami w dwu- lub trójwymiarowej przestrzeni, którą jednocześnie łatwo zwizualizować. Jednak co istotne, bliskość ma znaczenie jedynie dla punktów o takim samym charakterze, tj. wiersze z wierszami, a kolumny z kolumnami. Jeśli dwa punkty wierszowe położone są blisko siebie, to znaczy że charakteryzują się podobnymi wartościami kolejnych atrybutów. Jeśli chodzi o kolumny, należy rozróżnić dwa przypadki:

- bliskość punktów określających wartości różnych atrybutów oznacza, że te konkretne wartości mają tendencję to współwystępowania w obserwacjach
- bliskość punktów z zakresu tego samego atrybutu oznacza, że obserwacje, z którymi są powiązane wartości atrybutu opisywane przez te punkty są podobne

3.5. Word2Vec

W związku z brakiem satysfakcjonujących rezultatów dla poprzednio użytych metod, zdecydowałam się na zastosowaniu podejścia wykorzystującego sieci neuronowe. Jeśliby przeformułować zadanie postawione w niniejszej pracy do predykcji jednego atrybutu sesji na podstawie pozostałych, przywodzi to na myśl pewną analogię do technik stosowanych w przetwarzaniu języka naturalnego, bowiem modelowanie języka jest to zadanie polegające na przypisywaniu prawdopodobieństwa do sekwencji słów, tak aby rozkład ten był zgodny z faktycznym rozkładem słów w języku. Statystyczny model językowy można przedstawić za pomocą prawdopodobieństwa warunkowego wystąpienia kolejnego słowa,

biorąc pod uwagę wszystkie poprzednie [22] oraz korzystając z reguły łańcuchowej:

$$P(w_1 w_2 ... w_T) = \prod_{t=1}^{T} P(w_t | w_1 w_2 ... w_{t-1}) \tag{15}$$

W praktycznym modelowaniu przyjmuje się założenie Markova, że tylko n najbliższych słów będzie miało wpływ na wystąpienie w_t - zatem aproksymujemy prawdopodobieństwo wystąpienia słowa w danym kontekście:

$$P(w_1 w_2 ... w_T) = \prod_{t=1}^{T} P(w_t | w_1 w_2 ... w_{t-n+1}) \tag{16}$$

Klasyczny neuronowy model języka [22] maksymalizuje następującą funkcję celu:

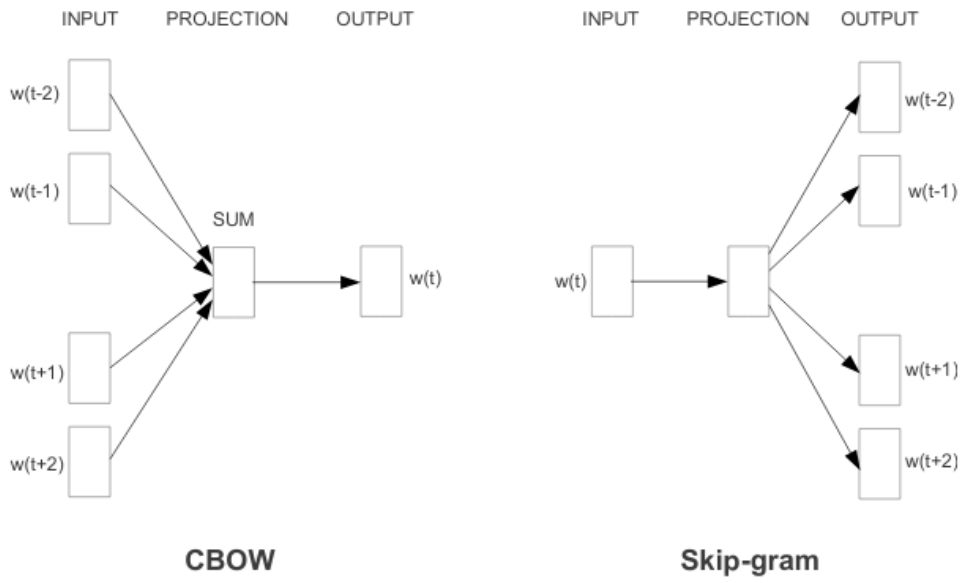
$$J_\theta = \frac{1}{T} \sum_{t=1}^{T} \log p(w_t | w_{t-1} ... w_{t-n+1}) \tag{17}$$

Tak jak w przypadku wszystkich dziedzin badawczych, w których korzysta się z metodologii uczenia maszynowego, tak również i w przetwarzaniu języka kluczową kwestią jest sposób przedstawienia danych i przekazania ich do algorytmu czyli w tym wypadku reprezentacja słów. Swego rodzaju przełomem w dziedzinie NLP było opracowanie tzw. reprezentacji dystrybucyjnej (ang. *word embeddings*). Nazwa ta wzięła się od semantyki dystrybucyjnej, która zakłada, że w dużych korpusach tekstowych znaczenie słowa można przejawia się poprzez konteksty, w jakich występuje - zatem podobne słowa występują w podobnym kontekście [23]. Metoda ta polega na takim doborze funkcji celu podczas trenowaniu modelu, aby model zmapował każde słowo do d-wymiarowej przestrzeni wektorowej, w której semantyczne podobieństwa między słowami są zachowane. Jedną z najbardziej popularnych metod tworzenia takich reprezentacji jest Word2Vec [24]. Autorzy w swojej publikacji prezentują dwie architektury płytkich (składających się z jednej warstwy ukrytej) sieci neuronowych, przedstawione na Rysunku 10, które są w stanie w sposób efektywny i przede wszystkim skuteczny tworzyć wektorowe reprezentacje dystrybucyjne. Obie architektury opisane pokrótce poniżej.

- Continuous Bag-of-Words (CBOW)
- Skip-gram

3.5.1. Continuous Bag-of-Words

Podczas gdy klasyczny model językowy jest w stanie brać pod uwagę tylko na poprzednie słowa w celu ich przewidywania, ponieważ jest ewaluowany na podstawie jego zdolności do przewidywania każdego następnego słowa w korpusie, model, który ma na celu wygenerowanie reprezentacji dystrybucyjnych, nie podlega temu ograniczeniu. Z tego powodu autorzy [24] używają tzw. okna, czyli kontekstu słowa w_t w postaci słów zarówno je poprzedzających, jak i po nim następujących, jak pokazano na Rysunku 10.



Rysunek 10. Architektury Word2Vec, źródło [24]

Co istotne kolejność słów zawartych w oknie nie ma znaczenia. Mając na względzie tę modyfikację, funkcję celu dla CBOW można zapisać następująco:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t | w_{t-n}...w_{t-1}\, w_{t+1}...w_{t+n}) \tag{18}$$

3.5.2. Skip-gram

Druga architektura może być traktowana jako przeciwieństwo CBOW. W tym wypadku, zamiast używać otaczających słów do przewidywania słowa centralnego, model skip-gram używa słowa centralnego do predykcji kontekstu w jakim się znalazło. Funkcja celu dla tak postawionego problemu formułuje się następująco:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n\leq j\leq n, \neq 0} \log p(w_{t+j} | w_t) \tag{19}$$

3.5.3. Adaptacja modelu do moich potrzeb

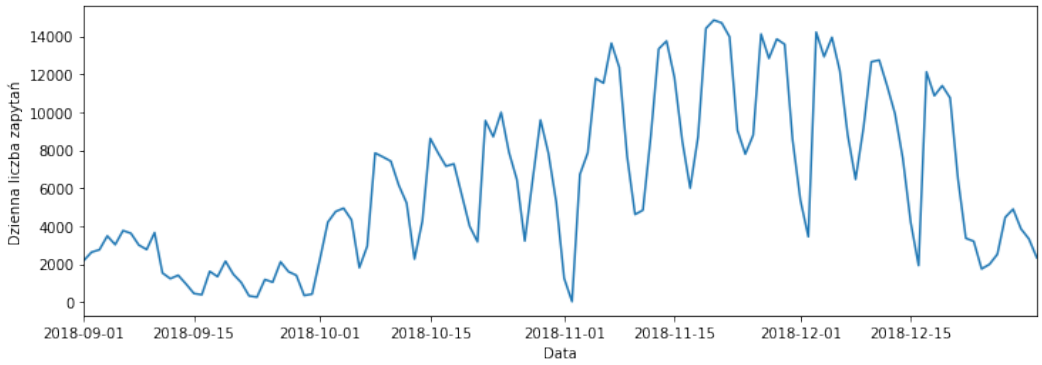
Aby móc skorzystać z mojego zbioru danych do wytrenowania modelu Word2Vec i otrzymania reprezentacji wektorowych, musiałam nanieść na niego pewną warstwę abstrakcji. Należało bowiem określić definicję słów oraz sposób konstruowania z nich zdań przekazywanych do modelu. Chcąc osiągnąć cel sprecyzowany jako przewidywanie jednego atrybutu sesji na podstawie atrybutów pozostałych, najrozsądniejszym wydawał się podział zbioru na zdania będące poszczególnymi rekordami i wybór wartości pojedyn-

czego atrybutu jako słowa. Dzięki temu podczas optymalizacji funkcji celu, model uczył się zależności pomiędzy wartościami atrybutów występującymi jednocześnie.

4. Wyniki

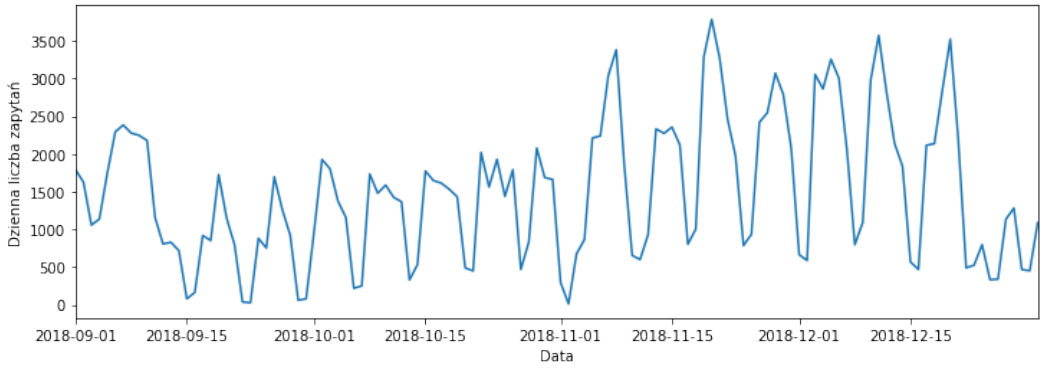
4.1. Prophet

Pierwszym krokiem analizy szeregów czasowych było wykreślenie dziennej aktywności użytkowników, widocznej na Rysunku 11



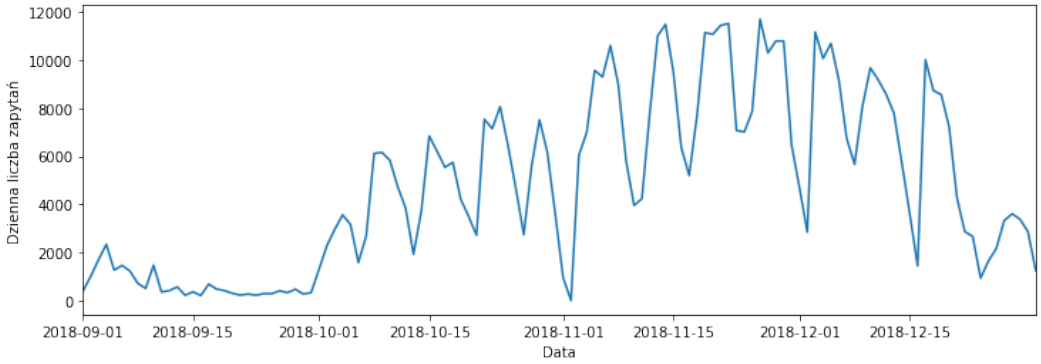
Rysunek 11. Wykres liczby zapytań w okresie wrzesień-grudzień.

Można zauważyć wyraźną okresowość tygodniową, zwłaszcza w okresie październik-grudzień oraz nieregularności i stosunkowo małą liczbę zapytań we wrześniu, co jest zgodne z intuicją. Biorąc pod uwagę duży udział zapytań niepoprawnych, sprawdziłam jak rozkładają się one w czasie (Rysunek 12).



Rysunek 12. Wykres liczby zapytań z kodem błędu z zakresu 400-499.

Jak widać ten rodzaj zapytań także wykazuje silną tygodniową okresowość z mniej więcej stałą intensywnością. Ciekawe są dwie "górki"w drugiej połowie września, które wyglądają niemal tak samo, co może wskazywać na nie-ludzką aktywność. Dodatkowo po odjęciu tych zapytań (Rysunek 13) główną różnicę można zauważyć właśnie we wrześniu, gdzie w miejscu tamtych górek, został praktycznie płaski rozkład. Aby jakość zarówno dopasowania, jak i predykcji była jak najwyższa, zapewniłam algorytmowi składnik $h(t)$ ze wzoru 1. Sporządziłam tabelę dni wolnych i świąt w okresie październik-grudzień (Tabela 5) i przekazałam ją do modelu.



Rysunek 13. Wykres liczby zapytań bez zapytań niepoprawnych.

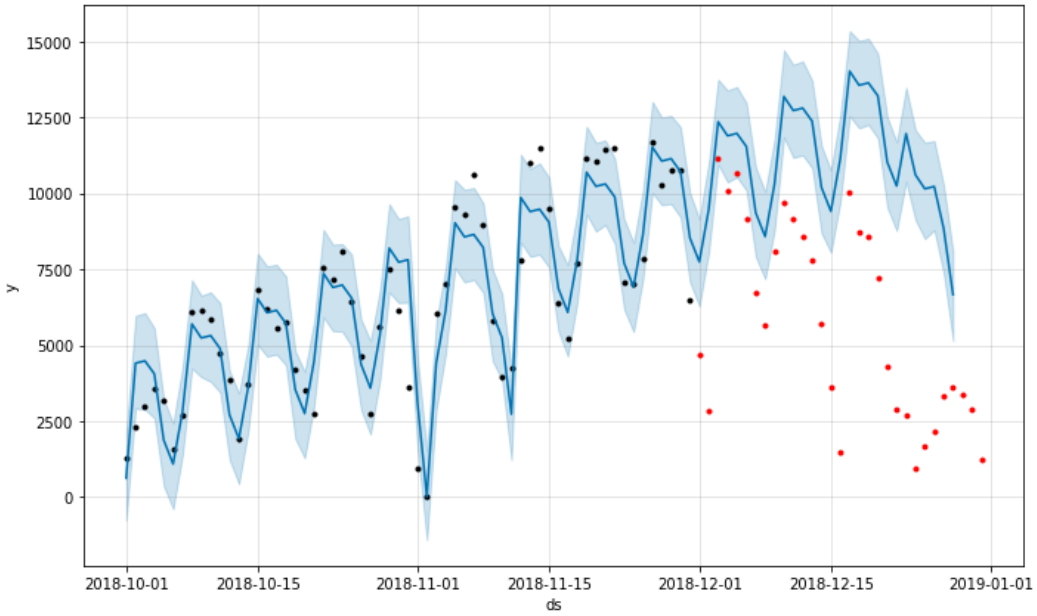
	wydarzenie	data
0	święto	2018-10-01
1	święto	2018-11-01
2	święto	2018-11-11
3	święto	2018-12-24
4	święto	2018-12-25
5	święto	2018-12-26
6	dzień_wolny	2018-11-02
7	dzień_wolny	2018-12-27
8	dzień_wolny	2018-12-28
9	dzień_wolny	2018-12-29
10	dzień_wolny	2018-12-30
11	dzień_wolny	2018-12-31

Tabela 5. Wydarzenia nieregularne w okresie październik-grudzień 2018

Z względu na to, że wrzesień nie wykazuje zbytniej regularności i wyraźnie różni się od kolejnych miesięcy, zdecydowałam się wykonać predykcje tylko biorąc pod uwagę miesiące październik i listopad z uwzględnieniem jedynie zapytań poprawnych. Prognoza została wykonana na okres kolejnych 31 dni. Wyniki przedstawione zostały na Rysunku 14.

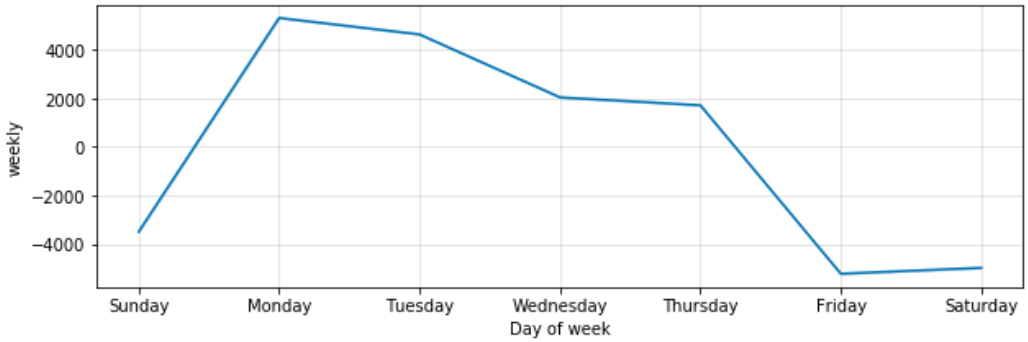
Czarne punkty są to dane przekazane do modelu, ciemna niebieska linia to dopasowanie wykonane przez prophet, jasnoniebieski obszar to przedział ufności ustalony na 80%, natomiast punkty czerwone to oryginalne dane z grudnia. Jak widać model bardzo dobrze poradził sobie z dopasowaniem do danych historycznych, a dzięki informacjom o świętach uwzględnił pik w okolicach 1 listopada. Dużo gorzej jednak wypada część prognozowana, gdzie model uwzględnił jedynie drobną tendencję spadkową w okolicach przerwy świątecznej, jednak ogólny poziom ilości zapytań został znacznie zawyżony. Okres, z którego pochodzą dane okazał się zdecydowanie za krótki, a tendencje zbyt wyraźne do wykonania wartościowej prognozy.

Przy okazji wykonywania predykcji sporządziłam także wykresy poszczególnych skła-

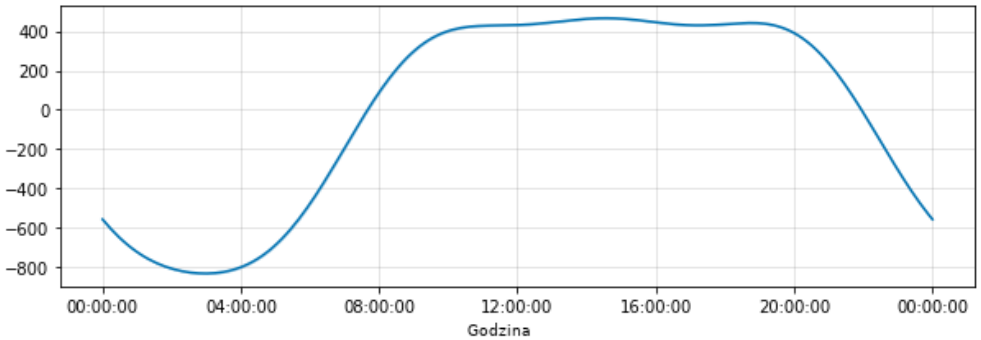


Rysunek 14. Dopasowanie wykonane przez prophet.

dowych konstruowanych przez model. Na Rysunkach 15 i 16 przedstawiłam kolejno składową tygodniową i dzienną.



Rysunek 15. Składowa tygodniowa.



Rysunek 16. Składowa dzienna.

Zgodnie z tym, co można było zauważyć na Rysunku 11, aktywność użytkowników przejawia okresowość tygodniową - podczas weekendów liczba zapytań drastycznie spada. Zaskoczeniem nie była także okresowość dzienna - poziom zapytań utrzymywał się na stałym poziomie w ciągu dnia i wyraźnie spadał w nocy.

4.2. K-modes

Jako pierwszy krok do wykonania grupowania na moim zbiorze danych zdecydowałam się sprawdzić, które przedmioty były wyszukiwane wspólnie najczęściej, aby móc to przełożyć na miarę podobieństwa poszczególnych sesji. W tym celu przekształciłam zbiór danych w sposób przedstawiony w Tabeli 6. Zakodowałam każdą sesję jako od-

sesja	przedmiot
1	TASS
2	SOI
3	SNR
4	SOI
5	PORR
6	TASS

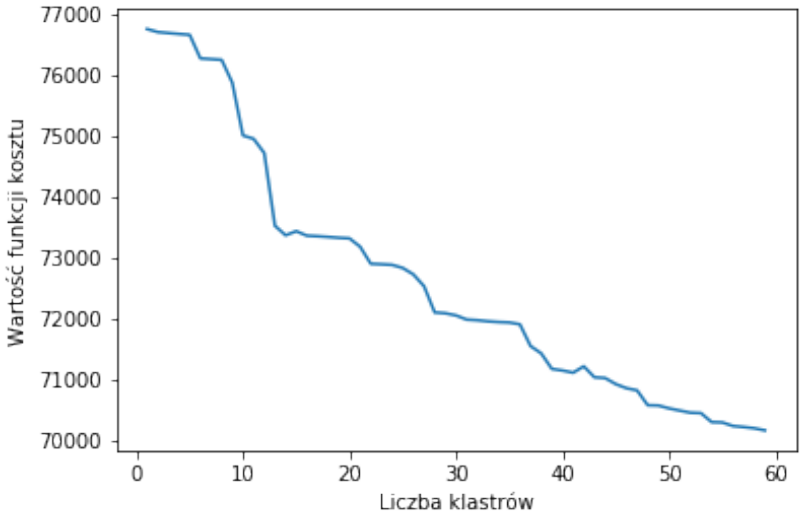
$\Longrightarrow$

	TASS	SOI	SNR	PORR
1	1	1	0	0
2	0	1	1	1
3	1	0	0	0

Tabela 6. Przekształcenie danych w celu zastosowania algorytmu k-modes.

dzielną obserwację posiadającą jako atrybuty wszystkie przedmioty. Umiejscowienie "1"w miejscu $\{i, j\}$, oznacza że w $i-tej$ sesji został odwiedzony $j-ty$ przedmiot. Uzyskano w ten sposób tabelę o rozmiarze 537×103838. Kolejnym krokiem było ustalenie na jaką liczbę k klastrów podzielić dane. Nie jest to zadanie trywialne, bowiem potrzebny jest kompromis pomiędzy prostotą, czyli liczbą klastrów, a efektywnością, czyli podziałem maksymalizującym jednorodność poszczególnych grup. Aby móc jakoś uzasadnić wybór liczby klastrów zastosowałam tzw. *elbow method*. Jest to metoda heurystyczna polegająca na wykonaniu wykresu zależności funkcji kosztu od ilości klastrów, a następnie znalezienie takiego k, dla którego dalsze zwiększanie liczby klastrów nie poprawia znacząco jakości grupowania, czyli miejsce, w którym funkcja dokonuje charakterystycznego przegięcia [25]. Jak widać na Rysunku 17, takie charakterystyczne miejsce występuje dla $k = 13$ i tę właśnie wartość wybrałam do przeprowadzenia grupowania. Patrząc na powyższy wykres warto zauważyć stosunkowo wysoką wartość funkcji kosztu, która wydaje się spadać zdecydowanie za wolno w odniesieniu do liczby klastrów. Sugeruje to, że dokonane grupowania mogą nie być zbyt wartościowe, bowiem w zbyt małym stopniu będą zbliżone do centroidów danych klastrów i będzie cechować je zbyt duża różnorodność, aby dało się na ich podstawie wyróżnić zestawy podobnych sesji.

Wykonanie grupowania dla $k = 13$ potwierdziło wcześniejsze zastrzeżenia. Fragment przykładowego wyniku działania algorytmu zamieściłam w tabeli 7 Z pierwszym klastrem algorytm poradził sobie bardzo dobrze, gdyż umieścił tam same angielsko-języczne



Rysunek 17. Elbow plot

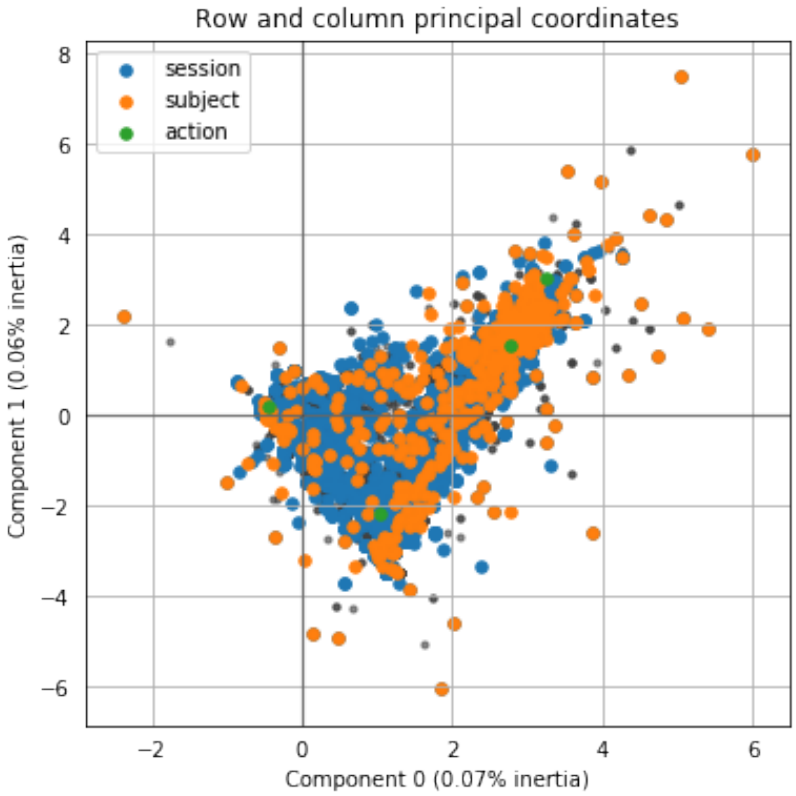
klaster 1:	EACAW EADCP EADIR EADS EANA EARIN EASP EBACK ECDS ECIRC ECIRS ECOAR ECOET ECOGR ECOHT ECOMM ECONE ECONT ECOPR ECOTE ECRYP ECULT EDABA EDAMI EDC EDCM EDCS EDDE EDICO EDISP EDRP EDSPA EDYCO EDYSY EEARE EECEL EELE EEVAL EFWA EGUI EIASR EIDMA EIMS EINIS EINNE EINST EINTE EIOD EIOT EISOC ELAC EMANA EMAR EMISY EMSMN ENEXT ENGON ENUME EOFT EOOP EOPSY EORI EPART EPCOS EPEFI EPFU EPHY EPNMEPRO EPRST EPRTE EQUMA EQUTH ESCS ESISM ESISY ESM ESOEN ESPTE ESPTR ESWIT ETASP ETEAM ETMAG ETSYS EUMTS EWAN EWNET EWSYS
klaster 2:	LTM
klaster 3:	TMO
klaster 4:	STP
Itd...	

Tabela 7. Przykładowy wynik grupowania algorytmem k-modes

przedmioty. Jednak kolejne 10 klastrów było jednoelementowych, a pozostałe przedmioty zostały przydzielone do zaledwie dwóch klastrów. Niestety mimo licznych prób eksperymentowania z liczbą klastrów, jak i sposobem inicjalizacji, nie udało mi się uzyskać bardziej znaczącego wyniku. Najprawdopodobniej taki sposób podziału jest spowodowany tym, że ponad 70% sesji odnosiło się do zaledwie jednego przedmiotu, przez co porównywanie przedmiotów na zasadzie analizy współwystępowania w danej sesji jest niewystarczające.

4.3. MCA

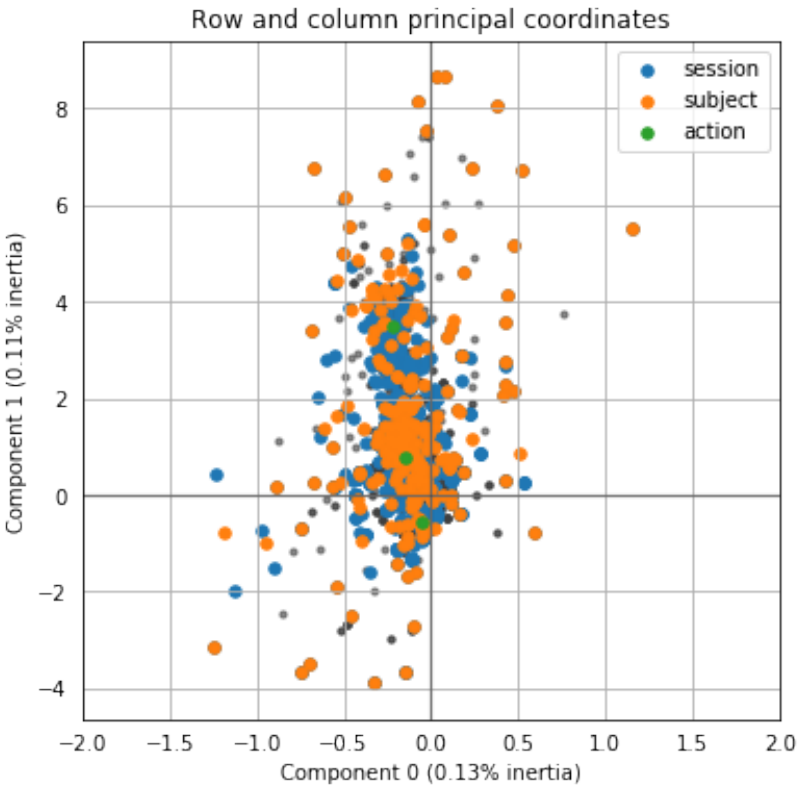
W celu uzyskania wyników łatwiejszych do interpretacji, a także ze względu na dużą złożoność obliczeniową, zastosowałam algorytm MCA do zbioru danych podzielonego na poszczególne miesiące. Taka metodologia pozwoliłaby na porównanie między sobą



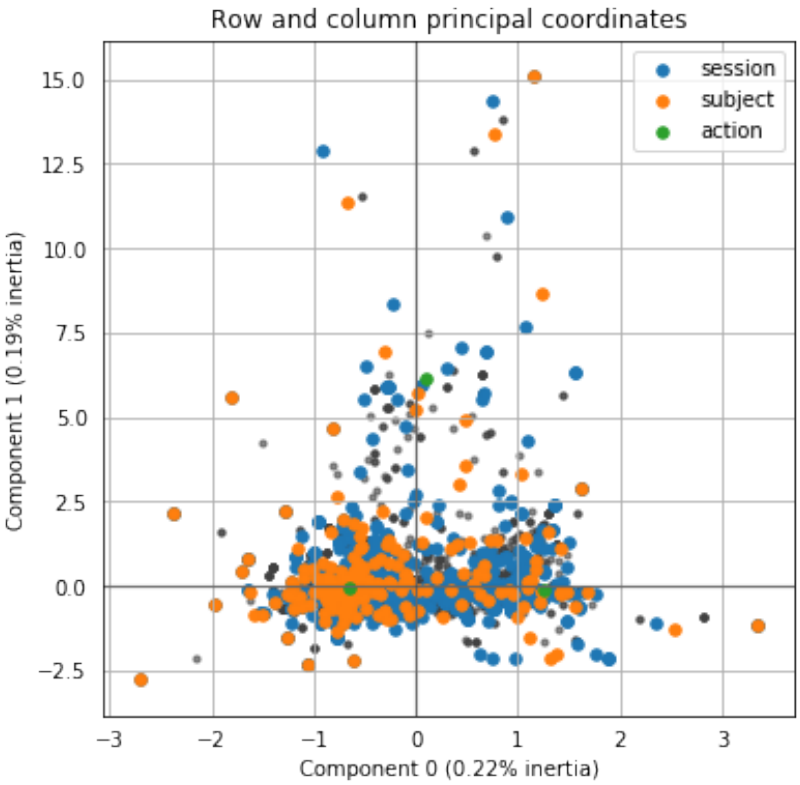
Rysunek 18. MCA wrzesień

skupisk wyszczególnionych w kolejnych miesiącach w celu weryfikacji poprawności i spójności grupowania, co niosłoby także informacje o dynamice akcji wykonywanych przez użytkowników. Projekcji dokonywałam przy pomocy ID sesji, przedmiotu i akcji, bowiem są te trzy atrybuty powinny nieść najwięcej informacji o zbiorze danych.

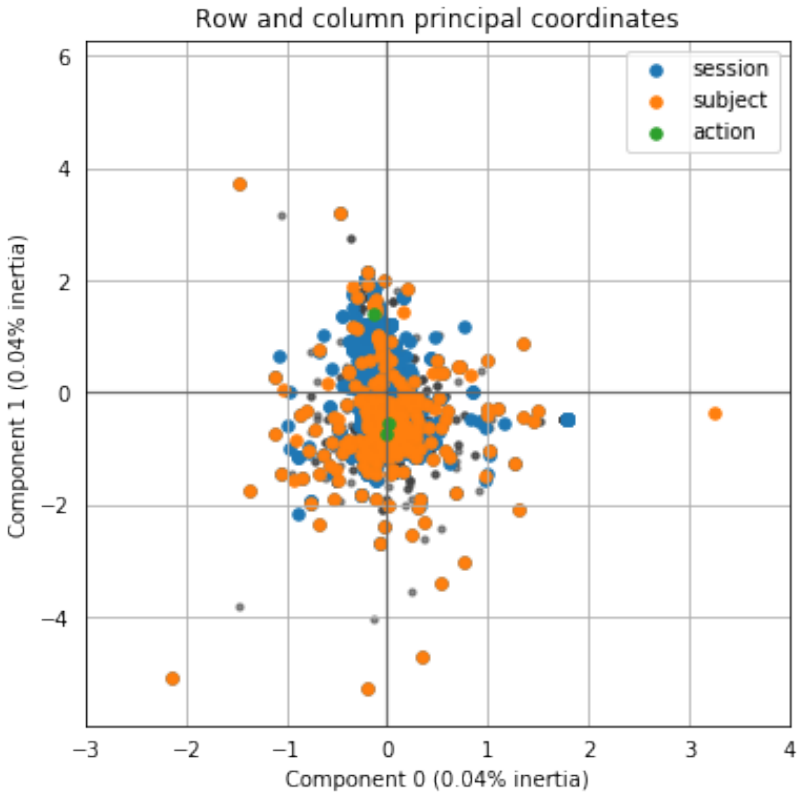
Niestety wyniki analizy okazały się bardzo niezadowalające. Na Rysunkach 18, 19, 20 i 21 przedstawiono rezultaty otrzymane dla pierwszych dwóch wyodrębnionych składowych w kolejnych miesiącach. Kolory punktów odpowiadają poszczególnym zmiennym, przy czym sesje traktowane były jako obserwacje, a przedmioty i akcje jako ich atrybuty, zgodnie z kodowaniem przedstawionym w rozdziale 4.4. Wszystkie projekcje przejawiają o wiele za niskie wartości inercji wyrażone przy pomocy dwóch pierwszych składowych, aby można było mówić o reprezentatywnym odwzorowaniu danych początkowych, bowiem nie sumują się nawet do 1%, podczas gdy autorzy [16] operują na wartościach rzędu 60-80%. Dodatkowo, nawet pomimo bardzo słabej jakości projekcji, obserwacje nie wykazują żadnych tendencji do tworzenia rozdzielnych skupisk, a jedynie pojedynczą chmurę z zagęszczeniem po środku. Być może jest to spowodowane ograniczeniem metody MCA do znajdowania jedynie zależności liniowych w danych. Wskazywać to może także na fakt, że serwer studia jest użytkowany w sposób niespecyficzny i aby odnaleźć pewne wzorce zachowań potrzebne są dodatkowe informacje.



Rysunek 19. MCA październik



Rysunek 20. MCA listopad



Rysunek 21. MCA grudzień

4.4. Word2Vec

Korzystając z metody Word2Vec przeprowadziłam szereg eksperymentów w celu otrzymania najbardziej satysfakcjonujących wyników. Eksperymenty te można podzielić na trzy kategorie:

1. Zmiana hiperparametrów modelu.
   Eksperymentowałam z rozmiarem warstwy ukrytej modelu, a zatem rozmiarem reprezentacji wektorowej, wielkością okna kontekstowego oraz ilością epok.
2. Zmiana atrybutów używanych podczas treningu.
   W celu dowiedzenia się, które atrybuty mają największy wpływ na jakoś przewidywania, konstruowałam zdania z różnych zastawów atrybutów. Ze względu na okresowość zapytań uwidocznioną podczas badania szeregów czasowych, do zbioru danych dołączyłam także dwie nowe zmienne: porę dnia i dzień tygodnia. Dzień podzieliłam na 4 pory:
   - noc - pomiędzy 00:00 a 6:00
   - rano - 6:00-12:00
   - popołudnie - 12:00-18:00
   - wieczór - 18:00-00:00

Wszystkie przedziały były domknięte lewostronnie. Łącznie zastosowałam 8 różnych kombinacji atrybutów.

3. Zmiana zmiennej docelowej, czyli zmiennej przewidywanej przez model.
   Skupiłam się na dwóch atrybutach: przedmiocie i akcji, bowiem te zmienne były kluczowe w procesie modelowania użytkowników na moim zbiorze danych. W procesie uczenia usuwałam wartości zmiennej docelowej ze zbioru treningowego i konstruowałam z nich zbiór testowy.

W celu oceny efektywności modelu Word2Vec zaimplementowałam własny mechanizm ewaluacji, bowiem standardową intencją tego podejścia jest samo otrzymanie reprezentacji wektorowej słów i na ogół nie używa się wytrenowanych modeli do predykcji. Ewaluacja ta polegała na sprawdzeniu, czy słowo (wartość atrybutu) przewidziane przez model odpowiada wartości rzeczywistej i na tej podstawie obliczałam metrykę *accuracy* (pol. *dokładność*). Dodatkowo podczas predykcji atrybutu przedmiot, ze względu na bardzo dużą liczność tej kategorii, obliczałam zmodyfikowaną wersję accuracy, znaną jako top-5 accuracy, która bierze pod uwagę czy wartość prawdziwa znajduje się w pięciu najbardziej prawdopodobnych wartościach wyjściowych modelu.

Eksperymenty zaczęłam od dobierania hiperparametrów modelu. Najkorzystniejszy ich zestaw został przedstawiony w Tabeli 8 i z jego wykorzystaniem przeprowadzałam eksperymenty z grup 2. i 3..

hiperparametr	wartość
rozmiar warstwy ukrytej	300
rozmiar okna	7
liczba epok	15

Tabela 8. Najlepsze wybrane hiperparametry dla modeli Word2Vec

Powyższe eksperymenty wykonywałam zarówno w architekturze CBOW jak i skip-gram. Zauważyłam, że nieznacznie lepiej sprawdza się model skip-gram, dlatego wszystkie zaprezentowane dalej wyniki będą odnosiły się właśnie do tej architektury.

W Tabeli 9 przedstawione zostały zbiorcze wyniki eksperymentów z grupy 2. i 3. Kolumna *zdanie* pokazuje jakie atrybuty były przekazane do treningu modelu; podczas predykcji odpowiednia zmienna docelowa była z tego zbioru usuwana. W celu poprawienia czytelności tabeli nazwy atrybutów zostały wpisane w formie skróconej, są to kolejno:

- ip - adres IP użytkownika
- kod - kod odpowiedzi HTTP serwera
- przeg - przeglądarka
- os - system operacyjny
- przedmiot
- akcja

- p_dnia - pora dnia
- d_tyg - dzień tygodnia
- sesja - numer ID przypisany danej sesji

atrybut	przedmiot		akcja
metryka	accuracy	acc_5	accuracy
zdanie			
ip, kod, przeg, os, przedmiot, akcja, p_dnia, d_tyg	0.5	0.86	0.79
ip, przeg, os, przedmiot, akcja, p_dnia, d_tyg	0.48	0.89	0.79
ip, kod, przeg, os, przedmiot, akcja	**0.59**	0.89	0.55
kod, przeg, os, przedmiot, akcja, p_dnia, d_tyg	0.12	0.35	0.82
ip, kod, przedmiot, akcja, p_dnia, d_tyg	0.53	**0.90**	**0.85**
ip, kod, przeg, os, przedmiot, akcja, p_dnia	0.48	0.86	0.74
ip, kod, przeg, os, przedmiot, akcja, d_tyg	0.47	0.86	0.72
sesja, przedmiot, akcja	0.5	0.84	**0.85**

Tabela 9. Wyniki dla modelu Word2Vec

Jak widać najlepsze wyniki accuracy zmiennej docelowej przedmiot, uzyskano dokonując predykcji na podstawie ip, kodu odpowiedzi, przeglądarki, systemu operacyjnego i akcji, jednak takie zdanie skutkuje najniższym wynikiem dla przewidywania akcji. Wartość top-5 accuracy najwyższa jest dla zestawu pomijającego system operacyjny i przeglądarkę, w tym przypadku najlepiej przewidywane były również akcje. Co ciekawe wprowadzenie zmiennych pory dnia i dnia tygodnia wcale nie przyczyniło się do wzrostu dokładności predykcji w przypadku obydwu zmiennych docelowych. Interesujący jest także wyjątkowo niski wynik dla zmiennej przedmiot w przypadku usunięcia ze zdania IP, a zupełny brak tej tendencji dla akcji. Dodatkowo usunięcie zamiast adresu IP przeglądarki i systemu operacyjnego przyczyniło się do bardzo wysokich wyników dla wszystkich pomiarów. Można zatem wysnuć wniosek, że sposób identyfikacji użytkowników za pomocą zarówno adresu IP, przeglądarki jak i systemu operacyjnego niekoniecznie w sposób optymalny odwzorowuje rzeczywisty podział użytkowników, bowiem wzorce zachowań wydają się być najsilniej skorelowane z adresem IP, a usunięcie dwóch pozostałych zmiennych wręcz poprawia jakość predykcji.

5. Podsumowanie

Niniejsza praca przedstawiała próby znalezienia technik pozwalających na tworzenie profili zachowań użytkowników, które mogą pozwolić na predykcję przyszłych aktywności poszczególnych grup. W tym celu przeprowadzono przegląd aktualnego stanu wiedzy z dziedziny eksploracji wykorzystania sieci, której jednym z podstawowych zadań jest modelowanie aktywności użytkowników witryny www. Zestawienie najczęściej używanych metod pozwoliło na wyrobienie sobie intuicji co do tego jakie techniki mogą okazać się przydatne.

Dane do analizy pochodziły z serwera studia - były to dzienniki serwera z okresu wrzesień - grudzień 2018. Początkowy etap prac skupił się na przetworzeniu danych w taki sposób, aby pozbyć się wszystkich informacji zbędnych i nie będących wynikiem bezpośredniej intencji użytkownika, a także zaobserwowanie jak największej liczby przydatnych zależności, które mogłyby przysłużyć się do bardziej wydajnego wykrywania ukrytych wzorców. Podzieliłam także zbiór na sesje przypisane do wydzielonych przeze mnie użytkowników.

Jako że jednym z początkowych założeń była łatwość interpretacji metody zastosowanej do wykrywania wzorców, pierwsze próby analizy danych przeprowadzałam za pomocą technik, których działanie nie opiera się o sieci neuronowe.

Poszukiwanie wzorców rozpoczęłam od analizy szeregów czasowych. Używając metodologii Prophet korzystającej z modelu dokonującego dekompozycji szeregów czasowych, próbowałam dokonać dopasowania krzywej do dziennej liczby zapytań oraz wykonać prognozowanie przyszłych wartości. O ile model poradził sobie z dopasowaniem do danych historycznych, to prognoza nie była w stanie przewidzieć zmiany dynamiki wykonywanych zapytań wraz z nadejściem przerwy świątecznej w grudniu. Jest to spowodowane zbyt krótkim przedziałem czasu, na którym model został dopasowany.

Kolejną wykorzystaną metodą była odpowiednik algorytmu k-średnich dla danych kategorycznych: k-modes. Technika ta miała dokonać grupowania sesji ze względu na współwystępowanie w nich przedmiotów. Jednak prawdopodobnie ze względu na fakt, że większość sesji dotyczyła zaledwie jednego przedmiotu, zależność taka była zbyt subtelna do wykonania wyraźnego grupowania, w efekcie czego jedyny klaster, któremu można by nadać znaczącą coś etykietę, składał się z przedmiotów angielskojęzycznych.

Następna technika miała pozwolić na znalezienie wzorców, które nie wynikają jedynie z prostego współwystępowania przedmiotu. Zastosowałam w tym celu metodę zmniejszania wymiarowości analogiczną do algorytmu PCA, z tym że przeznaczoną do analizy zmiennych kategorycznych - MCA. Jednak i w tym wypadku rezultaty nie były zadowalające, bowiem projekcja obserwacji na nowe składowe, nie niosła ze sobą wystarczająco dużo informacji, aby można było analizować ją jako pełnoprawną reprezentację danych początkowych.

Z powodu braku satysfakcjonujących wyników stosując metody wykorzystywane sze-

roko w standardowej eksploracji wykorzystania sieci, postanowiłam posłużyć się analogią do klasycznego modelu językowego i zastosowałam model wykorzystywany w przetwarzaniu języka naturalnego do konstruowania wektorowych reprezentacji tekstu. W tym celu zdefiniowałam zdanie jako zbiór wartości poszczególnych atrybutów dla danego zapytania i w ten sposób trenowałam model. Metoda ta przyniosła widoczne rezultaty - udało się bowiem na podstawie IP użytkownika, kodu odpowiedzi HTTP, przeglądarki, systemu operacyjnego i akcji przewidzieć jakiego przedmiotu dotyczy dane zapytanie z niemal 60% dokładnością, co przy tak licznej kategorii, jaką jest przedmiot, jest naprawdę dobrym wynikiem. Model poradził sobie także z predykcją akcji - aby zrobić to z 85% skutecznością, potrzebuje znać IP, kod odpowiedzi, przedmiot, porę dnia i dzień tygodnia, bądź jedynie ID sesji i przedmiot.

Mimo tego, że metoda Word2Vec pozwoliła na przewidywanie jednego atrybutu na podstawie pozostałych oraz ukazała pewne zależności pomiędzy atrybutami, zdecydowanie nie jest podejściem idealnym. Głównym problemem jest to, że model mimo pozornego rozróżnienia poszczególnych atrybutów, ze względu na wyuczone podobieństwa między nimi, nadal traktuje wszystkie wartości jako równoważne słowa. Nie zawiera zatem mechanizmu, który pozwala przewidywać jedynie wartości z danej kategorii. Skonstruowanie takiej metodologii mogłoby być dobrym pomysłem na kontynuację badań przedstawionych w powyższej pracy.

Bibliografia

[1] R. Cooley, B. Mobasher i J. Srivastava, „Web mining: Information and pattern discovery on the World Wide Web", grud. 1997, s. 558–567, ISBN: 0-8186-8203-5.

[2] M. J. Mughal, „Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview", *International Journal of Advanced Computer Science and Applications*, t. 9, czer. 2018.

[3] M. Aldekhail, „Application and Significance of Web Usage Mining in the 21st Century: A Literature Review", *International Journal of Computer Theory and Engineering*, t. 8, s. 41–47, lut. 2016.

[4] R. K. S. Anurag Kumar, „Web Mining Overview, Techniques, Tools and Applications: A Survey", *International Research Journal of Engineering and Technology (IRJET)*, t. 3, grud. 2016.

[5] K. Suneetha i R. Krishnamoorthi, „Identifying User Behavior by Analyzing Web Server Access Log File", *IJCSNS International Journal of Computer Science and Network Security*, t. 9, sty. 2009.

[6] J. Shanmugam i S. Rajagopalan, „A Survey on Web Personalization of Web Usage Mining", *International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 p-ISSN: 2395-0072*, t. Volume: 02, s. 6–12, mar. 2015.

[7] R. Cooley, B. Mobasher i J. Srivastava, „Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns", lip. 2000.

[8] D. S. T. Chitraa V., „A Novel Technique for Sessions Identification in Web Usage Mining Preprocessing", *International Journal of Computer Applications*, t. 34, list. 2011.

[9] M. Satya Prakash Singh, „Web Usage Mining Tools & Techniques: A survey", *International Journal of Advance Research, Ideas and Innovations in Technology*, t. 2, paź. 2016.

[10] J. Srivastava, R. Cooley, M. Deshpande i P.-N. Tan, „Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data.", *SIGKDD Explorations*, t. 1, s. 12–23, sty. 2000.

[11] J. F. Trevor Hastie Robert Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[12] G. Stermsek, M. Strembeck i G. Neumann, „A User Profile Derivation Approach based on Log-File Analysis", w *European Conference on Artificial Intelligence*, IOS Press, 2014.

[13] S. J. Taylor i B. Letham, „Forecasting at scale", *The American Statistician*, 2017.

[14] Z. Huang, „Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", *Data Min. Knowl. Discov.*, t. 2, s. 283–304, wrz. 1998.

[15] S. Khan i A. Ahmad, „Cluster center initialization algorithm for K-modes clustering", *Expert Systems with Applications*, t. 40, s. 7444–7456, grud. 2013.

[16] H. Abdi i D. Valentin, „Multiple Correspondence Analysis”, *Encyclopedia of Measurement and Statistics*, sty. 2007.

[17] I. Jolliffe, „Principal Component Analysis”, *Springer: Berlin, Germany*, t. 87, s. 41–64, sty. 1986.

[18] P. G. Heijden, A. Mooijaart i Y. Takane, „Correspondence analysis and contingency table models”, sty. 1994.

[19] M. Greenacre i O. Nenadic, „Computation of Multiple Correspondence Analysis, with Code in R”, *Department of Economics and Business, Universitat Pompeu Fabra, Economics Working Papers*, wrz. 2005. DOI: 10.2139/ssrn.847698.

[20] M. Greenacre, „Theory of Correspondence Analysis”, w *Correspondence Analysis in Practice*, CRC Press, 2007, s. 201–203.

[21] F. Husson i J. Josse, „Multiple Correspondence Analysis”, w. sty. 2014.

[22] Y. Bengio, R. Ducharme i P. Vincent, „A Neural Probabilistic Language Model”, t. 3, sty. 2000, s. 932–938.

[23] Z. S. Harris, „Distributional Structure”, *WORD*, t. 10, nr. 2-3, s. 146–162, 1954.

[24] T. Mikolov, G. Corrado, K. Chen i J. Dean, „Efficient Estimation of Word Representations in Vector Space”, sty. 2013, s. 1–12.

[25] M. Greenacre, *Statistics for Machine Learning*. Packt Publishing, 2017.

Spis rysunków

Spis tabel