Politechnika Łódzka

Instytut Informatyki

**PRACA DYPLOMOWA MAGISTERSKA**

# Metoda oceny wiarygodności ofert na portalach ogłoszeniowych

Method of assessing the credibility of offers on e-commerce portals

**Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej**
**Promotor: dr inż. Łukasz Chomątek**
**Dyplomant: Kamil Kowalewski**
**Nr albumu: 239676**
**Kierunek: Informatyka Stosowana**
**Specjalność: Data Science**

Łódź, lipiec 2022

Instytut Informatyki

90-924 Łódź, ul. Wólczańska 215, budynek B9

tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: office@ics.p.lodz.pl

# Spis treści

# Streszczenie

Celem niniejszej pracy magisterskiej była analiza oraz przedstawienie dostępnych metod do określania wiarygodności ofert oraz rekomendacji ich użytkownikom. Po dokonaniu analizy zostały obnażone pewne aspekty, które autorska metoda oceny wiarygodności ofert na portalach ogłoszeniowych może poprawić lub wręcz wyeliminować. Po dokonaniu implementacji autorskiej metody oraz wybranej metody dostępnej w literaturze zostały przeprowadzone badanie, których celem było porównanie działania oraz skuteczność na wybranych zbiorach danych zawierających oferty z popularnego portalu ogłoszeniowego. Analiza ta przekazuję informację, czy oraz w jakich przypadkach metody te się wyróżniają lepszymi lub gorszymi wynikami uzyskanymi po ich użyciu.

# Rozdział 1

# Wstęp

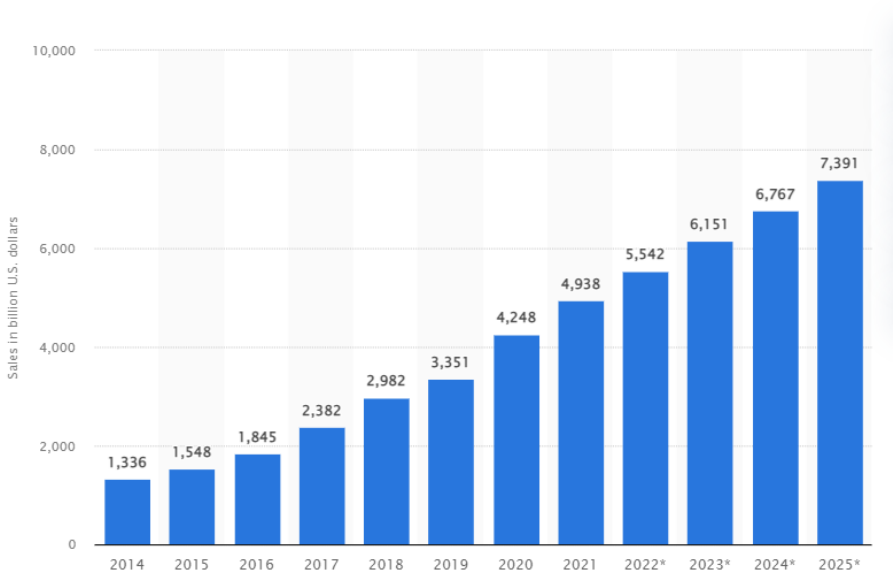
W czasach niesamowicie szybkiego rozwoju technologii, szczególnie tych związanych z informatyką, możliwość dokonywania różnych operacji w sposób zdalny poprzez wykorzystanie komputera jest niezwykle wygodne oraz pożądane przez obywateli wysoko rozwiniętych państw. Jedną z takich operacji są zakupy online na portalach ogłoszeniowych.

## 1.1 Problematyka

Problematyka dotycząca rekomendacji oraz określania wiarygodności ofert na portalach ogłoszeniowych jest rozległa i zdecydowanie nie jest trywialna. Ze względu na mnogość portali oraz ofert, które są umieszczane na nich, użytkownicy często mają do podjęcia naprawdę trudne decyzje w czasie zakupu. Jest to spowodowane głównie określeniem, które z ofert są godne zaufania, a które nie. Dla człowieka jest to proces żmudny oraz długotrwały co zaprzecza podstawowym założeniom handlu w internecie, który ma być szybki i wygodny. Całe szczęście istnieją sposoby na automatyzację procesu weryfikacji ofert. Niniejsza praca magisterska ma za zadanie przedstawić nową metodę i porównać ją z dostępnymi rozwiązaniami.

Celem potwierdzenia bardzo szybkiego rozwoju rynku e-commerce został zamieszczony wykres na rysunku 1.1, przedstawiający sprzedaż w zależności od roku. Jak łatwo zauważyć w przeciągu 7 lat - od roku 2014 do roku 2021 sprzedaż wzrosła o ponad 3,5 raza. Poprzez znak * zostały oznaczone prognozowane wartości dla

przyszłych lat. Jak widać, w 2025 roku prognozowany jest wzrost o 5,5 raza w stosunku do 2014 roku.




Rysunek 1.1: Sprzedaż detaliczna e-commerce na całym świecie w latach 2014-2025 [1]

## 1.2 Cel i założenia projektu

Celem niniejszej pracy magisterskiej było opracowanie autorskiej metody do określania wiarygodności ofert. W zakres prac można również włączyć analizę metod dostępnych w literaturze naukowej, implementację jednej z nich i dokonanie porównania z zaproponowaną metodą przez autora niniejszej pracy.

## 1.3 Definicje pojęć

**Portal ogłoszeniowy** - aplikacja webowa umieszczona w sieci Internet, która jest powszechnie dostępna. Zapewnia ona możliwość tworzenia konta użytkownika, które jest wymagana do sprzedaży i zakupu przedmiotów. Do przeglądania nie jest wymagane korzystanie z konta użytkownika. Przedmiot jest wystawiany przez sprzedającego natomiast zakupu dokonuje kupujący. Po zatwierdzeniu łączy ich swoista umowa kupna-sprzedaży, z której dwie strony są zmuszone się wywiązać.

**Oferta** - jeden z podstawowych obiektów. Zawiera ona w sobie sprzedawany produkt,

osobę, która go sprzedaję, liczbę sztuk, cenę, opinie innych użytkowników, którzy już dokonali jego zakupu i mieli ochotę wyrazić opinię na temat transakcji oraz produktu.

**Nieuczciwy sprzedawca** - sprzedawca, który umyślnie wprowadza w błąd kupującego poprzez podanie nieprawdziwych informacji na temat sprzedawanego produktu. W ten sposób oferta może stać się niewiarygodna.

## 1.4 Układ pracy

Pierwszy rozdział stanowi wstęp do tematyki portali ogłoszeniowych, rekomendacji ofert oraz określania ich wiarygodności. Drugi rozdział prezentuje przegląd literatury naukowej dostępnej w chwili opracowywania metody oraz tworzenia niniejszej pracy magisterskiej. Następny w kolejności rozdział przedstawia autorską metodę oceny wiarygodności ofert opracowaną oraz hipotezę badawczą, która w kolejnych rozdziałach będzie weryfikowana. Następnym, czwartym z kolei rozdziałem jest dotyczący środowiska eksperymentalnego, opisu implementacji autorskiej metody oraz całego programu wykorzystującego tę metodę. Piąty rozdział przedstawia wyniki przeprowadzonych eksperymentów oraz ich wizualizację. Ostatni rozdział zawiera podsumowanie i wnioski całej pracy magisterskiej oraz weryfikację postawionej hipotezy badawczej.

# Rozdział 2

# Przegląd literatury i wprowadzenie teoretyczne

Ze względu na dynamiczny wzrost, rozwój zagadnień i tematyki opisanej w rozdziale 1, naukowcy nie pozostają w tyle a wręcz wiodą prym w rozwijaniu systemów rekomendacji. Jest to dokonywane poprzez opracowywanie nowych metod i przedstawianie śmiałych wizji. Wiele z tych metod przedstawia nowatorskie pomysły oraz wykazuje się wyższą skutecznością niż rozwiązania dostępne w literaturze. Bardzo często jest to poparte badaniami w publikowanej literaturze. Wydawnictwa powszechnie zaliczane do grupy tych najbardziej prestiżowych przeprowadzają wnikliwe recenzje przed publikacją co w większości przypadków gwarantuje wysoką jakość przedstawianych treści. Nie mniej ze względu na mnogość dostępnych publikacji konieczna staje się weryfikacja ich zawartości celem wyboru takich prac, które przedstawiają zagadnienia najbardziej zbliżone do rozważanego problemu.

## 2.1 Systemy rekomendacji

Celem *Systemów rekomendacyjnych* [2] (ang. Recommendation Systems) jest automatyzacja doradztwa podczas podejmowania złożonych, wieloetapowych decyzji. Podstawą działania tych systemów są wcześniej dokonane wybory przez konkretnego użytkownika lub grupę użytkowników, do której można zaklasyfikować osobę lub podmiot, na rzecz którego wykonywana jest rekomendacja. Systemy te są rozwijane

i wykorzystywane w wielu liniach produktowych takich przedsiębiorstw jak np. Google[3] czy też Microsoft[4], celem zwiększenia zadowolenia użytkownika oraz zysków danej firmy. Nie jest to jedyna możliwość ich wykorzystania, gdyż pojęcie *Systemy rekomendacji* jest stosunkowo szerokie. Z najlepszej wiedzy autora niniejszej pracy magisterskiej wynika, że system, który dokonuje analizy przekazanych obiektów, a następnie zwraca użytkownikowi wskazania, czy też rekomendacje, co do wyboru jednego lub wielu z przekazanych obiektów, można zaliczyć do tej grupy systemów.

## 2.2 Określanie wiarygodności ofert oraz rekomendacja tych odpowiednich

Proces określania wiarygodności ofert należy do grupy procesów wieloetapowych, aby go przeprowadzić, warto pamiętać o podstawowych definicjach przedstawionych w sekcji 1.3. Na wiarygodność oferty ma wpływ wiele czynników takich jak stosunek proponowanej ceny do ceny rynkowej produktu, stosunek pozytywnych i negatywnych ocen oraz recenzji produktu. Nie mniej istotna jest opinia na temat sprzedawcy, która jest ugruntowana na podstawie informacji zwrotnych od klientów, którzy dokonywali transakcji u danego sprzedawcy.

Na podstawie ww. informacji oraz innych czynników jak obecność zdjęć produktu, czy też dokładny i szczegółowy opis, użytkownik wybranego systemu informatycznego nakierowanego na e-commerce, może stwierdzić, czy chce zamówić danych produkt, czy też nie. Nie jest tajemnicą, że proces ręcznej weryfikacji należy do dosyć żmudnych i czasochłonnych, co nie jest mile widziane w kontekście handlu elektronicznego.

Warto wspomnieć, że proces ten można zautomatyzować, jak większość procesów rekomendacji, na podstawie tożsamych informacji, jakie posiada potencjalna osoba dokonująca zakupu. Z tego powodu obecnie w literaturze istnieje pewna liczba rozwiązań proponowanych przez naukowców, aby zniwelować ten nieprzyjazny, dla przeciętnego człowieka, proces, który ma na celu podjęcie decyzji czy warto poważnie zainteresować się daną ofertą czy też lepiej jest kontynuować poszukiwania.

## 2.3 System określania wiarygodności oparty na analizie sementycznej treści recenzji

Artykuł zatytułowany ”Trustworthiness evaluation system in E-Commerce context”[5] pochyla się nad tematem tworzenia systemów podejmowania decyzji (ang. decision support system) oraz systemów reputacji zaufania (ang. Trust Reputation systems), często też oznaczane poprzez akronim *TRS*. Jego autorzy w początkowych fragmentach artykułu przedstawiają rozwój, zalety, wygodę zakupów poprzez zróżnicowane portale ogłoszeniowe. Na podstawie badań wskazują, że użytkownicy tych portali nadal wolą opierać się opiniach i doświadczeniach innych użytkowników, którzy zakupili towar z określonej oferty lub byli klientami danego sprzedawcy. Autorzy wskazują bardzo ciekawą kwestię, która nie jest poruszana bardzo często, a mianowicie to, że wiele parametrów takich jak bezpieczeństwo, jakość obsługi może być określana w bezpośredni sposób za pomocą metryk. W przypadku oceny wiarygodności ofert nie ma możliwości wykonania takiej oceny w sposób bezpośredni, aby otrzymać jeden konkretny wynik na podstawie wybranych parametrów.

Zwrócona jest uwaga na fakt, że znakomita większość gigantów w sektorze e-commerce zapewnia możliwości przesyłania opinii czy też recenzji, lecz nie jest w żaden sposób określana ich wiarygodność. Problem ten może występować z intencjonalnych lub nieintencjonalnych zachowań klientów, którzy wystawiają informację zwrotną niezgodną ze stanem rzeczywistym. Może być to kwestia pomyłki lub też celowego działania celem nieuczciwej walki z konkurencją bądź w celach zarobkowych na rzeczy właśnie innych sprzedawców stosujących zakazane praktyki w handlu. Rzeczą, o której warto wspomnieć, jest fakt, iż autorzy artykułu nie wzięli pod uwagę ważnej metodologii, jaką przyjęły wspominane już wielokrotnie portale ogłoszeniowe, a mianowicie, że opinię lub recenzje może wystawić użytkownik, który nie dokonał transakcji. Co więcej, w przeciwieństwie do opinii wystawianych w ramach np. Google Maps, jedna transakcja daje możliwość podzielenia się jedną opinią, a nie wieloma. Zabezpiecza to jeszcze bardziej przed osobami, które dla celów zarobkowych wystawiają oceny.

Rozwiązanie zaproponowane przez autorów opiera się na analizie semantycznej

zawartości tekstowej recenzji przekazanych przez użytkowników celem oceny produktu lub oferowanej usługi. Wspominają oni, że obecnie w literaturze było wiele prób tworzenia podobnych rozwiązań, lecz z wykorzystaniem podstawowych wartości numerycznych takich jak liczba transakcji klienta, zgodność w przedstawianych opinii, czy też data publikacji danej opinii lub recenzji.

Szczegółowe kroki proponowanej metody polegają na przeprowadzanie wspomnianej już analizy semantycznej, celem określania wydźwięku danej opinii np. czy jest pozytywna lub negatywna oraz w jakim stopniu. Uzyskane w ten sposób wyniki porównywane są z oceną wystawioną przez użytkownika. Dla przykładu może być to 5 gwiazdek. W przypadku znacznej rozbieżności, gdzie jej próg jest definiowany jako parametr, opinia taka jest oznaczana jako niewiarygodna. W kontekście wykorzystania tego w elektronicznym systemie sprzedażowym, wyglądałoby to w ten sposób, że użytkownik przesyła swoją informację zwrotną, dokonywana jest analiza zgodnie z przedstawionym algorytmem i tylko w przypadku pozytywnego jej wyniku jest ona dodawana do portalu. Skutkuje to jej upublicznieniem oraz przeliczeniem dotychczasowych statystyk na temat tego produktu.

W bardzo dokładny i przystępny został opisany sposób implementacji metody. Jedna z najważniejszych procesów, jakim jest analiza semantyczna, została określano w krokach takich jak wstępne przygotowanie tekstu poprzez tokenizację oraz inne popularne metody wstępnego przygotowania danych tekstowych. Kolejnym krokiem jest skorzystanie z *SentiWordNet* dostępnego w bibliotece *NLTK*, opisanej w sekcji 4.1.2.5, dostępnej dla języka Python, przedstawionego w sekcji 4.1.1.

W artykule został przedstawiony tylko przykładowy zbiór kroków, który ma na celu wystawienie opinii czy też recenzji razem z rysunkami obrazującymi ten proces. Niestety nie zostały przedstawione żadne badanie, z czego wynika, że nie został również przedstawiony żaden zbiór danych.

Co ważne przedstawioną pomysł oraz metodę można wykorzystać dwojako, pierwszym sposobem jest stworzenie zewnętrznego systemu, które określałby wiarygodność recenzji, a na podstawie tych właściwych wyliczał statystyki, dzięki czemu możliwe byłoby utworzenie dwóch grup ze zbioru produktów, tych nadających się do zakupu oraz nienadających się do zakupu. Drugim sposobem wykorzystania jest

implementacja tego mechanizmu przez właścicieli pewnego portalu ogłoszeniowego. We wspomnianym drugim przypadku autorzy przedstawiają użytkownikom, wcześniej przygotowaną pulę recenzji, aby ocenili oni ich poprawność oraz zgodność poprzez wybranie przycisku “Lubię To“ lub “Nie Lubię Tego“. Na tej podstawie byłoby określane intencje osoby przesyłającej opinie, co więcej system łączyłby to z wcześniejszym wynikiem o wiarygodności użytkownika na podstawie jego opinii.

## 2.4 Metoda określania wiarygodności ofert na podstawie opinii od wiarygodnych użytkowników

Artykuł zatytułowany ”A new reputation algorithm for evaluating trustworthiness in e-commerce context”[6] porusza tematykę określania wiarygodności ofert poprzez tworzenie systemów reputacji zaufania *TRS*. Zostały w nim przedstawione, w bardzo ciekawy sposób, definicje związane z domeną, jaką jest zaufania, wspomniane systemy *TRS* oraz czym charakteryzują się *TRS*, stworzone w solidnym i przemyślany sposób.

Następnie zostały przedstawione aktualne osiągnięcia z tej tematyki, dostępnej w literaturze naukowej. Autorzy artykuły wymieniają wiele prac, które skupiają się na propozycjach architektur systemów *TRS*, opartych na rozmaitych algorytmach obliczających ocenę reputacji w oparciu o sam produkt. Inne prace mocno ukierunkowują się na analizę semantyczną informacji zwrotnych, czy też recenzji od klientów, którzy zakupili ten oferowany produkt przez konkretnego sprzedawcę. Wspomniane jest, że pomimo starań, inne pozycje literaturowe nie rozważają wielu problemów, takich jak wiarygodność osób piszących komentarze, systematycznej aktualizacji ich wiarygodności, poprzez przechowywania wartości np. liczbowej, która mówiłaby o tym czy w recenzję lub opinię danego użytkownika można uwierzyć.

Niezwykle poruszającą kwestią jest przechodniość zaufania oraz jej związek z upływającym czasem i kolejnymi transakcja, czy też zakupami. Autorzy, powołując się na jedną z pozycji literaturowych, wskazują przedstawioną w niej przechodniość (tranzytywność) zaufania między agentami. Polega ona na tym, że skoro agent *A* ufa agentowi *B* natomiast nie ufa agentowi *C*, z powodu takiego, że agent *A* ufa bardziej agentowi *D* niż agentowi *B*. Agent *D* nie ufa *C* wiec agent *A* też mu nie ufa. Połączenia

te są skomplikowane i podczas ich aktualizacji trzeba zwracać szczególną ostrożność, aby przechodniość była zachowana.

Sama metoda, zaprojektowana i przedstawiona przez autorów, odbiega w dosyć znaczny sposób pod względem koncepcyjnym od istniejących rozwiązań. Na podstawie analizy recenzji lub opinii zamieszczonych przez danego użytkownika i ich wiarygodności, użytkownik ma dodawane lub odejmowane punkty od wartości liczbowej, określającej jego poziom wiarygodności. Zaproponowany zakres wartości to *[-5,5]*. Na rysunku 2.1 został zaprezentowany pseudokod algorytmu odpowiedzialnego za inkrementacja lub dekrementacje wartości wiarygodności użytkownika. Co warto dodać, w przypadku gdy wartość ta jest większa od *5*, jest ona ustawiana na wartość równą *5*.

```
Pseudo-code for the calculus of the trust degree of the user:
        function calculate_degree_trust_user () as double{
         list listinfos;
        Int idfeedback=get_idfeedback();
        Listinfos=get_infos_click (idfeedback);
        List listScore;
        Feedtrustworth= Listinfos[0];
        Userchoice= listinfos[1];
        Degree_trust_user=Listinfos[2];

        if (0<feedtrustworth<=1.5) and (userchoice="like")
        Or (-1.5=<feedtrustworth<=0) and (userchoice="dislike")
              Degree_trust_user+=0.25

        If (1.5<feedtrustworth<=2.5) and(userchoice="like")
        Or(-2.5=<feedtrustworth<-1.5) and (userchoice="dislike")
                    Degree_trust_user+=0.5

        If (2.5<feedtrustworth<=3.5) and (userchoice="like")
        Or(-3.5=<feedtrustworth<-2.5) and (userchoice="dislike")
                    Degree_trust_user+=0.75

        If (3.5<feedtrustworth<=5) and (userchoice="like")
        Or (-5=<feedtrustworth<-3.5) and (userchoice="dislike")
                    Degree_trust_user+=1
```

Rysunek 2.1: Algorytm określania i aktualizacji poziomu wiarygodności użytkownika przedstawiony w artykule [6]

Niestety pomimo ciekawego pomysłu, przedstawione rozwiązanie jest mocno niedopracowane i zawiera bardzo ogólną ideę. Wielokrotnie powtarzane jest sformułowanie, że w przyszłych pracach pojawią się dużo bardziej szczegółowe i dopracowane rozwiązania, jednak pomimo upływu czasu od publikacji nic takiego nie nastąpiło. Większość realnych problemów jest zasygnalizowane jako sugestia, jak np. wykorzystanie pewnych współczynników lub procentów do zmniejszania, lub

zwiększania poziomu wiarygodności użytkownika. Sama analiza semantyczna, również została przedstawiona w taki sposób, że trzeba ją wykonać, niestety brak jest informacji od autorów artykułu o adaptacji do przedstawionego algorytmu. Nie zostały przedstawione również żadne wyniki, które mogłyby świadczyć o przewadze przedstawionej metody w porównaniu do innych dostępnych rozwiązań. Jest to niezwykle ciekawy pomysł, który można rozwinąć w wielu kierunkach, lecz nie da się dokonać analizy wyników, czy kierunek założony przez odbiorców artykułu, jest odpowiedni i przynosi lepsze efekty.

## 2.5 Metoda kompleksowej analizy reputacji ofert na podstawie aspektowej analizy sentymentu

Artykuł zatytułowany ”Cross-Platform Reputation Generation System Based on Aspect-Based Sentiment Analysis”[7] skupia się na przedstawieniu komplementarnego systemu oceny wiarygodności, w którym zostały omówione dokładnie wszystkie aspekty. Warto zaznaczyć, że artykuł ten jest niezwykle aktualny, gdyż został opublikowany na początku pierwszego kwartału 2022 roku, natomiast autor niniejszej pracy magisterskiej przygotowuje ją w końcówce pierwszego kwartału oraz w drugim kwartale 2022 roku.

Autorzy artykułu w pierwszych sekcjach przedstawiają aktualny stan wiedzy w tej dziedzinie oraz fakt, że naprawdę dostępnych jest wiele prac naukowych poruszających ten temat i proponujących bardzo zróżnicowane metody. Tak jak już zostało wspomniane w ramach analizy pozostałych artykułów w sekcjach 2.3 oraz 2.4, wiekszość z nich opiera się na analizie opartej o wartości numeryczne, w nowszych pracach są opierane o analizę semantyczną. Niestety ze względu na ograniczenia wynikających z samej koncepcji ocen i recenzji, nie da się pozyskać więcej informacji.

Zdaniem autorów proponowane wcześniej metody, czy też algorytmy, bardzo mocno skupiały się na samym rdzeniu działania, natomiast były one przedstawione jako teoretyczne idee bez konfrontacji z realną ich implementacją i wykorzystaniem. Co więcej, pomijały one niezwykle kluczowe aspekty takie jak pobieranie i ekstrakcja cech z konkretnego portalu, czy też wizualizacja uzyskanych wyników, aby użytkownik

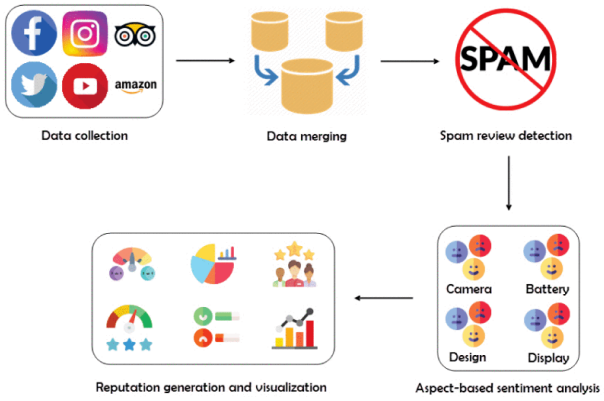
programu implementującego metody mógł w łatwy i świadomy sposób podjąć decyzję.

Zdecydowanie warto nadmienić, że artykuł ten porusza dużo szerszy obszar niż tylko portale ogłoszeniowe. Źródłem opinii mogą być oczywiście wspomniane portale ogłoszeniowe, natomiast niezwykle cenne są również opinii z takich miejsc jak platforma Facebook[8], Instagram[9], Twitter[10], YouTube[11] czy np. TripAdvisor[12]. Niestety w przypadku wspomnianych serwisów, jest możliwość wystawienia dowolnej liczby opinii, nawet w przypadku niebrania udziału w wydarzeniu czy też korzystania z danego serwisu czy też portalu. Jest to niezwykle niebezpieczne z punktu widzenia wiarygodności tych opinii i dalszego sugerowania się nimi przy podejmowaniu decyzji.

Ze względu na powyżej wspomniane kwestie autorzy zaproponowali następujący zestaw kroków wykonywanych przez system. Został on zaprezentowany na rysunku 2.2. Przechodząc do omówienia kroku określonego jako filtrowanie pozyskanych opinii, czy też recenzji (ang. Spam Filtering) warto podkreślić, że zostały zaproponowane dwa sposoby wykrywania. Pierwszym z nich jest wyliczanie podobieństwa między opiniami danej osoby. Zgodnie z przedstawionymi badaniami, większość osób wystawiająca fałszywe opinie dla korzyści finansowych, piszę je w bardzo podobny sposób, często korzystając z pewnego rodzaju szablonu, aby zwiększyć swoją wydajność, a co za tym idzie zwiększyć swój zysk. Drugim z nich jest wyliczanie wartości wystąpień recenzji danego użytkownika (ang. User Number of Reviews Frequency (UNRF)). Podstawą działania jest fakt, że nienaturalnym zachowaniem jest wystawianie zbyt dużej liczby recenzji przez jednego użytkownika dla wybranego obiektu, oferty czy też usługi. Na podstawie przedstawionych w artykule wzorów są wyliczane wartości (ang. score) dla obu sposób i na tej podstawie jest podejmowana decyzji, czy dany użytkownik jest osobą udostępniającą fałszywe recenzje i oceny, czy też nie. W przypadku uznania, że osoba udostępnia fałszywe informacje zwrotne są one usuwane w procesie filtracji.

Po przeprowadzeniu kroków wstępnych, opisanych powyżej, wykonywane jest proces analizy sentymentu opartej o aspekty. Polega on na określeniu znaczenia tekstu, a dokładnie to, czy jest pozytywny, czy też negatywny ze względu na przedstawione w tym tekście podmioty. Mogą one być określane przez przymiotniki o pozytywnych i negatywnych znaczeniach. Ważne jest to, aby brać te podmioty oraz ich określenia,

które faktycznie są związane z tym danym obiektem. Przykładem to obrazującym jest jakość robionych zdjęć przez aparat oraz jego wygląd. Jakości zdjęć często nie można określić przed zakupem, natomiast wygląd jak najbardziej, więc nie powinno być to brane pod uwagę przy wyliczaniu znaczenia danej wypowiedzi. Sami autorzy przedstawili to w bardzo ciekawy i dojmujący sposób, tłumacząc zaawansowane kwestie z tym związane.




Rysunek 2.2: Zestaw kroków przeprowadzanych przez proponowany system reputacji przedstawiony w artykule [7]

Przedstawione powyżej biegunowość jest jedną z wartości branych pod uwagę przy wyliczaniu reputacji, oprócz niej są jeszcze dwie. Pierwszą z nich, jest wartość popularności danej opinii, czy też recenzji, wyliczana na podstawie liczby polubień i udostępnień. Podstawą do brania tego pod uwagę jest fakt, że jeżeli ktoś polubił daną opinię to najprawdopodobniej się z nią zgadza lub uznał ją za przydatną w procesie podejmowania decyzji. Drugą wartością jest data wystawienia danej informacji zwrotnej, im jest starsza tym jest większe prawdopodobieństwo tego, że jest nieaktualna lub jej zawartość może nie do końca zgadzać się z teraźniejszym stanem. Powodem tego może być to, że np. sprzedawca poprawił lub pogorszył jakość usług, lub produktu. Możliwe jest też to, że zmienił np. opis, dzięki czemu teraz każdy klient jest dokładnie poinformowany o newralgicznych kwestiach.

Na podstawie tych trzech wartości jest obliczany wartość reputacji (ang. reputation score) i jest ona przedstawiona użytkownikowi w graficznej formie razem z podziałem wartości analizy sentymentu na aspekty. Co więcej, najbardziej wiarygodne opinie lub

recenzje są wyświetlane, celem potwierdzenia przedstawionych wartości liczbowych.

W artykule w bardzo przystępny sposób zostały przedstawione eksperymenty, wartości parametrów i hiperparametrów oraz uzyskane wyniki. Same zbiory danych wykorzystane do badań zawierają zagregowane opinie o np. hotelu z przedstawionych powyżej portali TripAdvisor, Facebook czy Twitter lub dotyczące filmu, pobrane z portalu IMDb oraz z Facebook i Twitter.

## 2.6 Potencjalne obszary ulepszeń obecnie dostępnych rozwiązań

Artykuły przedstawione w sekcjach 2.3, 2.4 oraz 2.5 zawierają bardzo ciekawe pomysły, często ich implementacje oraz wyniki. Bazują one na innych, wcześniej przedstawionych pracach i je rozwijają lub pokazują zupełnie inne podejście, mając na uwadzę dotychczasowy stan wiedzy w tej dziedzinie.

Autor niniejszej pracy dyplomowej po dogłębnych poszukiwaniach w dostępnych bazach literatury naukowej, przeczytaniu dziesiątek artykułów, które zostały wybrane po wstępnej selekcji na podstawie istotności w stosunku do tematu tej pracy, postanowił przedstawić potencjalne obszary ulepszeń.

Zdaniem autora niniejszej pracy dyplomowej można wykorzystać wszystkie dostępne informacje, jakie zapewnia dany portal ogłoszeniowy. Są to informacje w formie wartości liczbowych takich jak cena, średnia ocena klientów, oceny o sprzedawcy w różnych kategoriach takich jak szybkość dostawy, dokładność opisu. Co więcej, można przeprowadzić analizę semantyczną zawartości tekstowej, nie ograniczając się do zwykłej biegunowości, czyli czy ocena jest pozytywna lub negatywna. Istnieje możliwość badania emocji na podstawie słów użytych przez autora recenzji. Zebrane tak dane, można poddać procesowi grupowania poprzez skorzystanie z jednego z wybranych algorytmów zapewniających taką możliwość. Wynikiem grupowanie byłyby dwa klastry z ofertami o wysokiej lub niskiej reputacji. Podejście to rozwiązuje problem dobierania współczynników, które dla różnych zbiorów mogą być zdecydowanie inne, aby uzyskiwać zadowalające efekty.

# Rozdział 3

# Opracowana metoda

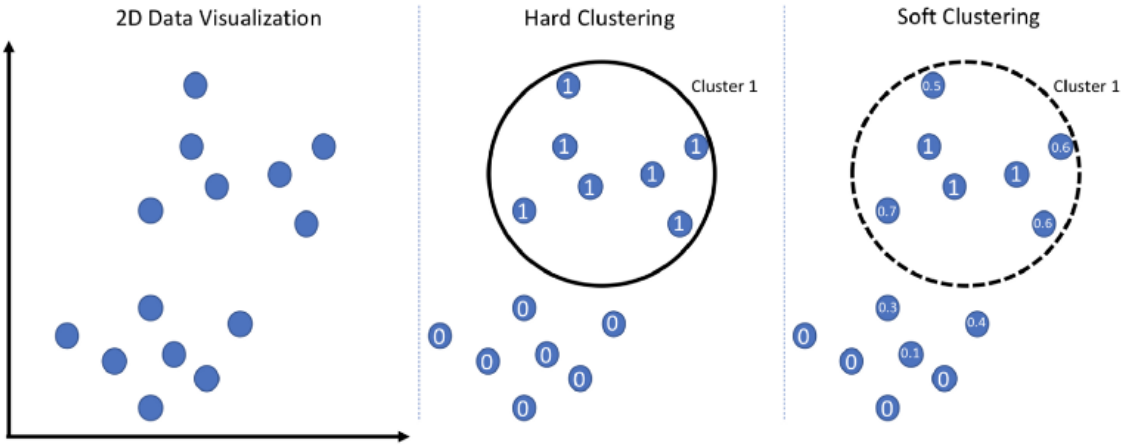
Po dokonaniu analizy dostępnych rozwiązań w literaturze naukowej oraz pozyskaniu aktualnej wiedzy w tematyce autor pracy zdecydował się na krok w postaci opracowania własnej metody, której celem jest określanie wiarygodności ofert. Zadanie to można było wykonać w bardzo zróżnicowany sposób natomiast autor zdecydował się na skorzystanie z algorytmu K-średnich oraz jej wersji opartej na logice rozmytej.

## 3.1 Algorytm K-średnich

Algorytm K-średnich[13] (ang. K-Means) należy do metody grupowania opartego na odległości. Ze względu na popularność nazwy z języka angielskiego, która jest bardzo często używana w języku polskim, autor w dalszej części niniejszej pracy będzie posługiwał się określeniem K-Means. Jego działania rozpoczyna się od podziału zbioru przypadku na $K$ skupień i rozpoczyna działanie od losowo wybranych $K$ środków skupień, które są możliwie jak najbardziej od siebie oddalone. W czasie kolejnych iteracji przypisuje obiekty do najbliższych skupień. Sama odległość jest wyliczana na podstawie wybranej metryki np. euklidesowej. Po dokonaniu alokacji obiektu są wyznaczane nowe środku skupień i na tej podstawie jest przypisywany kolejny obiekt. Kroki te są powtarzane do uzyskania stabilizacji lub gdy funkcje kryterialne nie osiągną swojego minimum.

## 3.2 Rozmyta wersja algorytm K-średnich

Algorytm rozmyty C-średnich[14] (ang. Fuzzy CMeans) różni się od algorytmu K-średnich, poza oczywiście inną nazwą parametru, *K* zmienione na *C*, faktem, że zamiast twardego klastrowania (ang. hard clustering) jest przeprowadzane miękkie klastrowanie (ang. soft clustering). Dla każdego z punktów jest określana wartość przynależności do danego klastra, jest to wartość z zakresu *[0,1]*. Im bliżej centroidu, tym wartość przynależności jest bliższa wartości 1. Różnica ta została zobrazowana na rysunku 3.1. Co więcej, analogicznie jak dla algorytmu K-Means w sekcji 3.1 autor pracy będzie się posługiwał nazwą C-Means.




Rysunek 3.1: Zobrazowanie różnicy między działaniem algorytmów K-Means i C-Means[15]

## 3.3 Warianty opracowanej metody

Wynikiem prac nad metodą oceny wiarygodności oferty okazały się dwie bliźniacze metody, które współdzielą wiele elementów. Pierwsza z nich wykorzystuję algorytm K-Means opisany w sekcji 3.1, natomiast druga z nich wykorzystuje algorytm C-Means przedstawiony w sekcji 3.2.

Krokami, jakie współdzielą, jest wstępne przetwarzanie danych oraz ekstrakcja nisko oraz wysoko poziomowych cech, na podstawie których tworzony jest wektor cech (ang. feature vector), po uprzednim wykonaniu normalizacji danych.

W wcześniej wspomnianym wektorze cech są umieszczane następujących cechy

przedstawione poniżej, natomiast algorytm nie ogranicza się do tych cech. Jest oczywiście możliwość wykorzystania innych istotnych cech w kontekście oceny wiarygodności oferty. Te przedstawione przez autora są jedynie propozycją oraz cechami, jakie zostały wykorzystane w ramach implementacji przedstawionej w sekcji 4.

- cena produktu
- czy istnieje możliwość zwrotu produkty
- liczba liter zawarta w opisie przedstawionej ofercie
- liczba punktów uzyskanych na podstawie przeprowadzonych transakcji i informacji zwrotnych od klientów
- procent informacji zwrotnych od klientów o charakterze pozytywny dla danego sprzedawcy
- rok dołączenia sprzedawcy
- bezwzględna liczba pozytywnych ocen sprzedawcy
- bezwzględna liczba neutralnych ocen sprzedawcy
- bezwzględna liczba negatywnych ocen sprzedawcy
- ocena dokładności opisu zamieszczonego w ofertach danego sprzedawcy
- ocena kosztu przesyłki produktów zawartych w ofertach danego sprzedawcy
- ocena czasu przesyłki produktów oferowanych przez danego sprzedawcę
- ocena komunikacji ze sprzedawcą
- wynik analizy semantycznej treści recenzji produktu zamieszczonej przez klientów, jej wynikiem jest wiele wartości odpowiadających emocjom przekazywanym w treści

Warto zaznaczyć, że sam sposób przeprowadzanie analizy semantycznej i jej wynik zależy od wybranej implementacji. Dokładny opis podejścia oraz implementacji zostanie przedstawiony w dalszej części niniejszej pracy.

Kolejnym krokiem, mając przygotowane wektory cech, jest wykonanie klasteryzacji (grupowanie). Zależnie od wybranego wariantu metody jest wykonywane to algorytmem K-Means lub C-Means z parametrem odpowiednio $K$, lub $C$ o wartości równej $2$. W wyniku tego procesu, z tak ustawionym parametrem, zawsze otrzymuje dwa klastry, czyli jawny i konkretny podział ofert ze względu na wartości znajdujące się w wektorze cech.

Niezwykle ważnym krokiem jest podjęcie decyzji, który z klastrów zawiera oferty bardziej wiarygodne. W przypadku proponowanego algorytmu warto wyraźnie zaznaczyć, że musi pozostać minimum jedna cecha, nisko lub wysoko poziomowa zależnie od implementacji, która nie trafi do wektora cech. Jest ona potrzebna do podjęcia wspomnianej decyzji o większej lub mniejszej wiarygodności ofert w danym klastrze. Proces ten można rozbudować i opierać na wielu cechach, jednak autor metody proponuje wykorzystanie parametry, jakim jest *średnia z wszystkich ocen produktu*. Na wybór tej propozycji miała wpływ implementacja przedstawiona w sekcji 4.

## 3.4 Hipoteza badawcza

Przedstawiona metoda wykaże się większą skutecznością niż ta proponowana w ramach artykułu przedstawionego w sekcji 2.4. Jest to spowodowane faktem, że bierze pod uwagę większą liczbę informacji, analiza semantyczna jest bardziej dogłębna oraz operacja grupowania sama dostosowuje się do konkretnego zbioru danych w przeciwieństwie do wartości progu, która decyduje o reputacji ofert.

# Rozdział 4

# Środowisko eksperymentalne

Celem przyświecającym autorowi niniejszej pracy magisterskiej było stworzenie środowiska eksperymentalnego opierającego się na standardach oraz technologiach uznawanych przez ekspertów jako wiodące. Co więcej, niezwykle ważne również było korzystanie z narzędzi w pełni darmowych lub otwartoźródłowych, aby w przyszłości odwzorowanie badań było zadaniem, które nie będzie sprawiać problemów.

## 4.1 Implementacja programu

### 4.1.1 Uzasadnienie wyboru języka programowania

Autor zdecydował się na wykorzystanie języka programowania Python[16] ze względu na jego popularność w dziedzinach informatyki związanych z uczeniem maszynowym, analizą danych czy też data science. Jest to technologia niezwykle wygodna oraz przystępna do przetwarzania oraz analizy danych, również tych złożonych. Jej rozwój następują w sposób dynamiczny, lecz zrównoważony, dzięki czemu jego wykorzystywanie oraz utrzymanie projektu napisanego we właśnie tej technologii nie stanowi problemu. Jednymi z głównych sponsorów są takie przedsiębiorstwa jak Google[3] czy Microsoft[4] co daje wysoki stopień pewności systematycznych aktualizacji, naprawy odnalezionych luk bezpieczeństwa oraz błędów w oprogramowaniu.

### 4.1.2 Wykorzystane biblioteki programistyczne

Aby efektywnie wykorzystać potencjał języka Python oraz jego możliwości, zostały wykorzystane popularne biblioteki programistyczne. Należą do nich chociażby NumPy[17] lub Pandas[18] oraz biblioteki opisane poniżej w ramach tej sekcji. Są one przede wszystkim bardzo znane, posiadają świetną dokumentację techniczną. Poziom i dokładność otestowania jest bardzo wysoki. Co więcej, tak samo, jak język programowania Python, są sponsorowane przez wiodące firmy z sektora IT. Dzięki temu wszelkie błędy są bardzo szybko naprawiane, a samo wydanie nowej wersji poprzedzają dogłębne testy.

#### 4.1.2.1 Biblioteka NumPy

Pierwszą ze wspomnianych bibliotek jest biblioteka NumPy. Jej główną funkcjonalnością jest N-wymiarowa tablica. Dzięki niej w bardzo przystępny sposób istnieje możliwość przechowywania danych w postaci macierzowej, które są tak popularne w przypadku zagadnień związanych z uczeniem maszynowym czy analizą danych. Oczywiście *N* jest parametrem, którego dolną granicą jest wartość *1*, natomiast górna granica to fizyczne ograniczenia sprzętowe. Dla przykładu tablica może być *100* wymiarowa, jeśli jest potrzeba przechowywania takich danych. Warto dodać, że biblioteka ta zapewnia bardzo duży zbiór operacji matematycznych, jakie można wykonywać oraz numeryczne narzędzia obliczeniowe takie jak transformaty Fouriera, czy funkcje wymagane do algebry liniowej. Całość zwieńcza fakt, iż rdzeń biblioteki został zaimplementowany w dobrze zoptymalizowanym kodzie napisanym w języku programowania *C*. Na listingu 4.1 został przedstawiony przykład użycia biblioteki Numpy.

```
1 >>> import numpy as np
2 >>> x = np.arange(4).reshape((2,2))
3 >>> x
4 array([[0, 1], [2, 3]])
5 >>> np.transpose(x)
6 array([[0, 2], [1, 3]])
```

Listing 4.1: Przykładowe użycie biblioteki NumPy

#### 4.1.2.2 Biblioteka Pandas

Biblioteka Pandas również ma za zadanie zapewniać infrastrukturę do przechowywania danych. W jej przypadku głównym obiektem jest *DataFrame*. Zapewnia on możliwość przechowywania oraz manipulacji danymi. Jest w nim również zintegrowane indeksowanie. Biblioteka ta jest szczególnie wygodna do operowania na danych dostarczonych w formatach *.csv*, *.tsv*, czy też *.xlsx* oraz wielu innych, które są mniej popularne. Dla każdego typu jest przygotowana w bibliotece określona metoda do wygodnego wczytywania zbioru danych. Są to np. *pd.read_csv()*, czy też *pd.read_excel()*, które zwracają wspomniany wcześniej obiekt *DataFrame*. Dostępnych jest wiele operacji, jakie możemy na nim wykonywać poprzez użycie dostarczonych w bibliotece funkcji. Założenia przyświecające autorom były takie, aby funkcjonalności były bardzo intuicyjne. Wszystkie operacje są dokonywane w sposób deklaratywny, jest to mocno zbliżone do kwerend wykonywanych w języku SQL (ang. Structured Query Language). Co więcej, dostępne są też analogiczne funkcję jak w SQL, a mianowicie *count()*, *groupby()*, *join()*. Dostarczane są również funkcje przeznaczone stricte do zadań z tematyki Data Science jak imputacje brakujących danych, usuwanie pustych rekordów, czy też konwersje do oczekiwane formatu np. zamiana dokładnej daty na porę roku co daje możliwość uzyskania lepszych wyników po dokonaniu analizy danych. Na listingu 4.2 został przedstawiony przykład użycia biblioteki Pandas.

```
>>> import pandas as pd
>>> df = pd.read_csv('data.csv')
>>> df
  Name  Score
0    a     90
1    b     80
2    c     95
>>> df.max()
Name      d
Score    95
dtype: object
>>> df.count()
Name     3
Score    3
dtype: int64
```

Listing 4.2: Przykładowe użycie biblioteki Pandas

#### 4.1.2.3 Biblioteka Matplotlib

Biblioteka Matplotlib[19] należy również do grona najbardziej popularnych oraz wręcz bazowych narzędzi w rękach programisty języka Python. Zapewnia ona możliwość bardzo zaawansowanej wizualizacji danych przechowywanych we wcześniej wspomnianych bibliotekach, a mianowicie NumPy oraz Pandas. Samo jej podstawowe użycie jest niezwykle łatwe oraz intuicyjne, dzięki czemu stała się wręcz podstawowym narzędziem do przedstawiania uzyskanych wyników. Jest ona oczywiście w pełni darmowa oraz ciągle rozwijana w dynamiczny sposób więc jej wykorzystanie nie niesie za sobą niebezpieczeństwa utraty wsparcia po pewnym czasie.

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  t = np.arange(0.0, 2.0, 0.01)
5  s = 1 + np.sin(2 * np.pi * t)
6
7  fig, ax = plt.subplots()
8  ax.plot(t, s)
9  ax.set(xlabel='time (s)', ylabel='voltage (mV)', title='Simple plot')
10 ax.grid()
11
12 fig.savefig("test.png")
13 plt.show()
```

Listing 4.3: Przykładowe użycie biblioteki matplotlib

#### 4.1.2.4 Biblioteka scikit-learn

Kolejną biblioteką z wymienianych jest scikit-learn[20]. Aby zapewnić deklarowane przez twórców funkcjonalności wykorzystuje ona w biblioteki takie jak NumPy oraz Pandas, które są de facto standardem i wykorzystywane są przez wiele innych bibliotek zapewniających implementacje różnych algorytmów uczenia maszynowego. Sam scikit-learn należy właśnie do grupy bibliotek zapewniających zróżnicowane algorytmy uczenia maszynowego. Są one pogrupowane ze względu na rodzaj, można wyróżnić następujące grupy algorytmów klasyfikacji, klasteryzacji (grupowania), regresji, redukcji wymiarowości oraz dotyczące preprocessingu. Samo wykorzystanie dostępnych funkcji oraz metod jest bardzo proste oraz intuicyjne ze względu na spójny

interfejs prezentowany przez klasy oraz bardzo sensownie dobrane domyślne wartości. Dzięki temu nawet bez dogłębnych badań, w krótkim czasie można uzyskiwać zadowalające wyniki. Co warto dodać dostarczana jest świetna dokumentacja, w której są również przykłady jak stosować konkretne metody i algorytmy, ich podstawy teoretyczne z bardzo dobrą wizualizacją. Wspierają to szybszy rozwój osoby z niej korzystającej oraz częściowo eliminuje możliwość dobrania złego algorytmu do problemu. Na listingu 4.4 został przedstawiony przykład użycia biblioteki scikit-learn.

```
>>> X = [[0], [1], [2], [3]]
>>> y = [0, 0, 1, 1]
>>> from sklearn.neighbors import KNeighborsClassifier
>>> neigh = KNeighborsClassifier(n_neighbors=3)
>>> neigh.fit(X, y)
KNeighborsClassifier(...)
>>> print(neigh.predict([[1.1]]))
[0]
>>> print(neigh.predict_proba([[0.9]]))
[[0.666... 0.333...]]
```

Listing 4.4: Przykładowe użycie biblioteki scikit-learn

#### 4.1.2.5 Biblioteka NLTK

Biblioteka NLTK[21] zapewnia możliwość przetwarzania oraz analizy języka naturalnego, stąd też jej nazwa (ang. Natural Language Toolkit). Dzięki niej dostępne są operacje wstępnego przetwarzania takie jak tokenizacji, usuwanie słów występujących w stop liście (ang. stop words), przeprowadzanie stemizacji (ang. stemming) lub lematyzacji (ang. lemmatization). Co więcej, można dokonywać wykrywania języka na podstawie podane fragmentu tekstu, czy też analizy sentymentu danej wypowiedzi lub tekstu. Informacją zwrotną analizy sentymentu jest stopień w jakim jest ona pozytywna, neutralna lub też negatywna. Co ciekawe jest udostępniona możliwość graficznej demonstracji np. analizy składniowej bądź dokonania tłumaczenia z konkretnego języka na wybrany.

#### 4.1.2.6 Biblioteka NRCLex

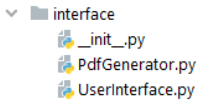
Biblioteka NRCLex[22] zapewnia jeszcze bardziej rozbudowaną analizę sentymentu. Na podstawie słów określa emocje zawarte w tekście, poprzez zwrócenie wartości od *0* do *1* w każdej z kategorii przedstawionych poniżej:

- strach (ang. fear)
- gniew (ang. anger)
- oczekiwanie (ang. anticipation)
- zaufanie (ang. trust)
- niespodzianka (ang. surprise)
- pozytywność (ang. positive)
- negatywność (ang. negative)
- smutek (ang. sadness)
- obrzydzenia (ang. disgust)
- radość (ang. joy)

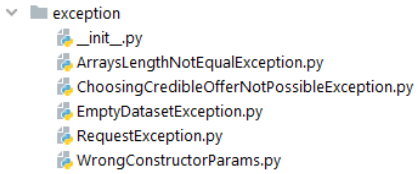
Co warto wspomnieć, suma wartości we wszystkich kategoriach jest równa *1*. Dodatkowo są zapewnione funkcjonalności dzielenia na słowa czy też zdania, natomiast zbiory danych do określania emocji są wykorzystywane z biblioteki NLTK opisanej w sekcji 4.1.2.5.

### 4.1.3 Struktura modułów i komponentów

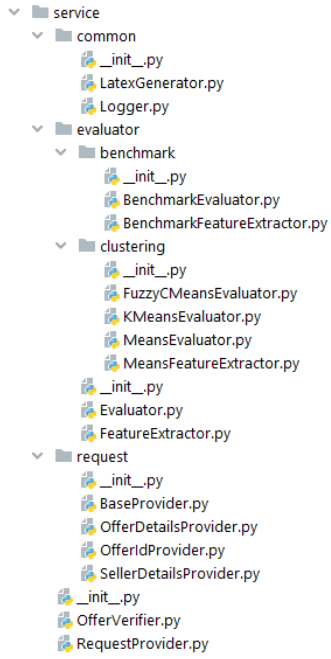
Program implementujący metodę przedstawioną w sekcji 3.3, został przygotowany zgodnie z popularnymi i uznawanymi przez wiele osób *dobrymi praktykami*. Autor zdecydował się wydzielić cztery pakiety. Są to *model*, *service*, *interface* oraz *exception*, przedstawione odpowiednio na rysunkach 4.4, 4.3, 4.1, 4.2. Przedstawiona poniżej infrastruktura jest wykorzystywana w kilku skryptach. Wykorzystywane są one do faktycznego wykorzystania programu dla użytkownika końcowego oraz skrypty do przeprowadzania badań na rzecz niniejszej pracy magisterskiej. W ramach każdego pakietu wymagany jest plik *__init__.py*, dzięki niemu interpreter wie, że jest to pakiet w języku *Python*.



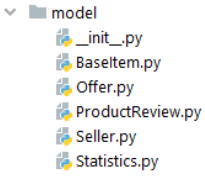

Rysunek 4.1: Zawartość pakietu *interface*




Rysunek 4.2: Zawartość pakietu *exception*




Rysunek 4.3: Zawartość pakietu *service*




Rysunek 4.4: Zawartość pakietu *model*

Opis plików przedstawionych na rysunku 4.1

- **PdfGenerator.py** - Generowanie raportów w formacie pdf
- **UserInterface.py** - Obsługa konsolowego interfejsu użytkownika

Opis plików przedstawionych na rysunku 4.2

- **ArraysLengthNotEqualException.py** - Klasa wyjątku informująca o nierównej długości tablicy
- **ChoosingCredibleOfferNotPossibleException.py** - Klasa wyjątku informująca braku możliwości wyboru wiarygodnych ofert
- **EmptyDatasetException.py** - Klasa wyjątku informująca o pustym zbiorze danych
- **RequestException.py** - Klasa wyjątku informująca o błędzie w przetwarzania żadania HTTP

- **WrongConstructorParamsException.py** - Klasa wyjątku informująca o błednych parametrach konstruktora

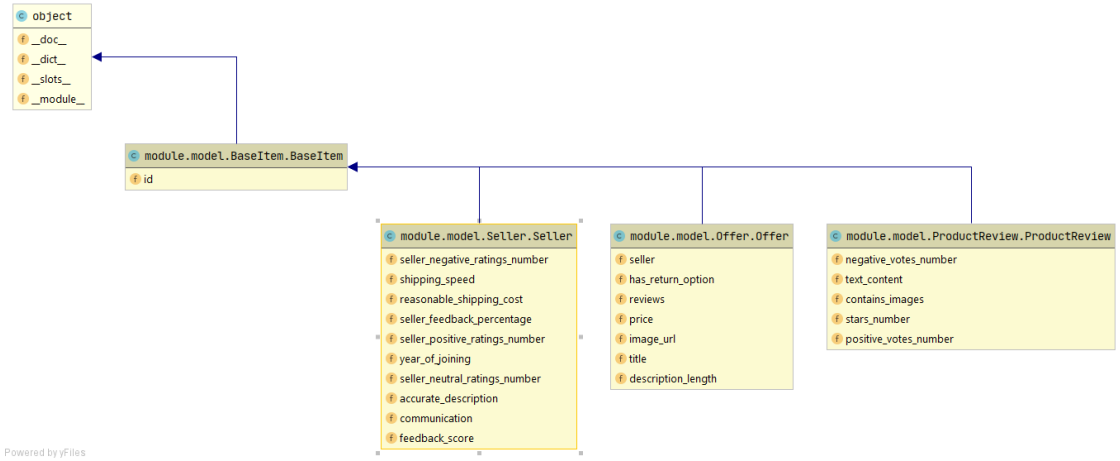
Opis plików przedstawionych na rysunku 4.3

- **LatexGenerator.py** - Generowanie raportów w składni technologii LaTex
- **Logger.py** - Logowanie
- **BenchmarkEvaluator.py** - Implementacja metody z literatury
- **BenchmarkFeatureExtractor.py** - Ekstracja cech dla metody z literatury
- **FuzzyCMeansEvaluator.py** - Implementacja autorskiej metody z wykorzystaniem algorytmu C-Means
- **KMeansEvaluator.py** - Implementacja autorskiej metody z wykorzystaniem algorytmu K-Means
- **MeansEvaluator.py** - Klasa bazowa dla *FuzzyCMeansEvaluator.py* oraz *KMeansEvaluator.py*
- **MeansFeatureExtractor.py** - Ekstracja cech dla autorskiej metody
- *Evaluator.py* - Klasa bazowa dla *MeansEvaluator.py* oraz *BenchmarkEvaluator.py*
- **FeatureExtractor.py** - Klasa bazowa dla *MeansFeatureExtractor.py* oraz *BenchmarkFeatureExtractor.py*
- **BaseProvider.py** - Klasa bazowa dla serwisów pobierających informacje o ofercie
- **OfferDetailsProvider.py** - Implementacja pobierania danych o ofercie
- **OfferIdProvider.py** - Implementacja pobierania identyfikatorów ofert dla wyszukiwanej frazy
- **SellerDetailsProvider.py** - Implementacja pobierania danych o sprzedawcy
- **OfferVerifier.py** - Implementacja weryfikacji wiarygodności ofert
- **RequestProvider.py** - Klasa agregująca funkjonalności*OfferDetailsProvider.py*, *OfferIdProvider.py*, *SellerDetailsProvider.py*

Opis plików przedstawionych na rysunku 4.4

- **BaseItem.py** - Klasa bazowa zawierająca unikalny identyfikator
- **Offer.py** - klasa modelu reprezentująca ofertę
- **OffersWrapper.py** - klasa modelu zapewniająca możliwość serializacji ofert do formatu *.json*
- **ProductReview.py** - klasa modelu reprezentująca recenzję oferty

- **Seller.py** - klasa modelu reprezentująca sprzedawcę
- **Statistics.py** - klasa modelu reprezentująca statystyki generowane przez program




Rysunek 4.5: Diagram UML klas zawartych w pakiecie model z pominięciem *Statistics*

## 4.2 Nałożone ograniczenia

W ramach realizacji implementacji autorskiej metody powstał program, w którym zostały nałożone pewne ograniczenia. Zostały one przedstawione poniżej:

- wymagany jest dostęp do sieci Internet
- program pobiera oferty tylko produktów oznaczone jako nowe
- program realizuje funkcjonalności tylko dla produktów katalogowych, nie ma możliwości analizy ofert z produktami unikalnymi jak dzieła sztuki
- forma zakupu jest ograniczona do *Kup Teraz* (ang. Buy Now)
- obsługiwana waluta to Dolar amerykański (USD)
- językiem w jakim obsługiwana jest analiza treści recenzji jest język angielski

## 4.3 Wstępne przygotowanie danych oraz ekstrakcja cech

Wyselekcjonowane informacje, które są pobierane przez program, poddawane są procesowi mapowanie na klasy modelu przedstawione w sekcji 4.1.3. Instancja klasy *Offer* agreguje w sobie instancję klasy *Seller* oraz listę instancji klasy *Review*. Wskutek tego otrzymujemy obiekt z danymi, na którym operuję program. Przed przekazaniem

tych danych do opracowanej metody jest dokonywane utworzenie obiektu klasy *DataFrame* dostępnego w bibliotece *Pandas* przedstawionej w sekcji 4.1.2.2. Do tego obiektu w pierwszym kroku trafiają następujące cechy oferty, przedstawione poniżej:

- cena produktu *(offer.price)*
- czy istnieje możliwość zwrotu produkty *(offer.has_return_option)*
- liczba liter w opisie przedstawionej ofercie *(offer.description_length)*
- liczba punktów uzyskanych na podstawie przeprowadzonych transakcji i informacji zwrotnych od klientów *(offer.seller.feedback_score)*
- procent informacji zwrotnych od klientów o charakterze pozytywny dla danego sprzedawcy *(offer.seller.seller_feedback_percentage)*
- data dołączenia sprzedawcy *(offer.seller.year_of_joining)*
- bezwzględna liczba pozytywnych ocen sprzedawcy *(offer.seller.seller_positive_ratings_number)*
- bezwzględna liczba neutralnych ocen sprzedawcy *(offer.seller.seller_neutral_ratings_number)*
- bezwzględna liczba negatywnych ocen sprzedawcy *(offer.seller.seller_negative_ratings_number)*
- ocena dokładności opisu zamieszczonego w ofertach danego sprzedawcy *(offer.seller.accurate_description)*
- ocena kosztu przesyłki produktów zawartych w ofertach danego sprzedawcy *(offer.seller.reasonable_shipping_cost)*
- ocena czasu przesyłki produktów oferowanych przez danego sprzedawcę *(offer.seller.shipping_speed)*
- ocena komunikacji ze sprzedawcą *(offer.seller.communication)*

Następnie dla kolumn nienumerycznych jest wykonywana operacja kodowania na wartości numeryczne za pomocą obiektu klasy *LabelEncoder* z biblioteki *scikit-learn*, przedstawionej w sekcji 4.1.2.4. Dzięki temu wspomniane cechy nienumeryczne mogą się znajdować w wektorze cech i wybrany algorytm, czy też metoda może na nich operować. Kolejny krok jest niezwykle ważny, gdyż jest to normalizacja wartości we wszystkich kolumnach. Zapewnia to przeskalowanie wartości w danej kolumnie z pierwotnego zakresu do wartości z zakresu od *0* do *1*.

W następnym etapie dokonywana jest analiza semantyczna treści każdej z recenzji danej oferty. Pierwszym jej krokiem jest weryfikacja czy zawartość tekstowa treści jest w języku angielskim, zgodnie z założeniami opisanymi w sekcji 4.2.

Kolejno jest wykonywane wstępne przygotowanie tekstu poprzez dokonanie kroków takich jak transformacja wszystkich liter do postaci małych liter, tokenizacja słów, usunięcie słów, które występują na stop liście lub zawierają znaki inne niż litery od *A* do *Z*. Po usunięciu wspomnianych słów przeprowadzana jest lematyzacji, czyli sprowadzenie słowa do jego formy podstawowej. Wszystkie te operacji zostały wykonane przy pomocy biblioteki *NLTK* opisanej w sekcji 4.1.2.5. Na listingu 4.5 została przedstawiona implementacja wstępne przetwarzanie tekstu.

```
1 def _prepare_text(self, text: str) -> str:
2         return list_to_string([
3             WordNetLemmatizer().lemmatize(x)
4             for x in word_tokenize(text.casefold())
5             if x.isalpha() and x not in self.stopwords
6         ])
```

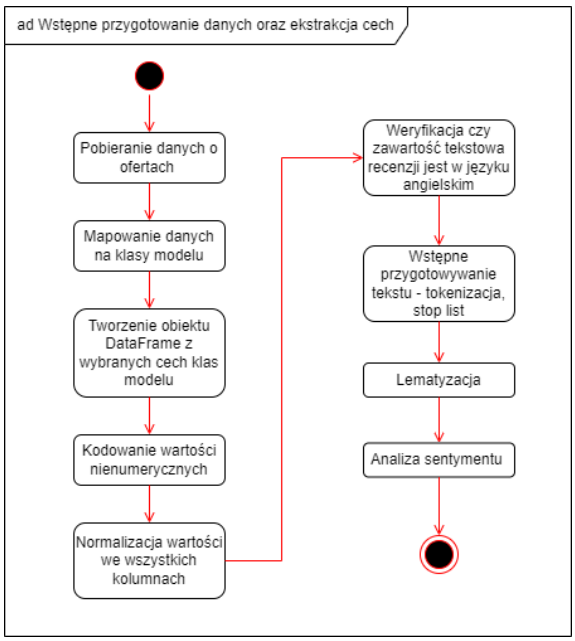
Listing 4.5: Implementacja wstępnego przetwarzania tekstu

Następnie jest przeprowadzana analiza sentymentu poprzez określenie emocji za pomocą biblioteki *NRCLex*, opisanej w sekcji 4.1.2.6. Z uzyskanej listy słowników (ang. dictionary), które zawierają konkretne wartości wyliczana jest średnia wartość danej emocji dla każdej z treści recenzji. Dzięki temu uzyskujemy uśrednione wartości emocji dla wszystkich recenzji. Co warto wspomnieć, biblioteka *NRCLex* zwraca wartości już po procesie normalizacji. Z tego powodu krok ten jest wykonywany jako ostatni, aby dwukrotnie nie dokonywać normalizacji. Uzyskane dziesięć wartości dla każdej z ofert jest dodawane do obiektu klasy *DataFrame*, który został już wcześniej zainicjalizowany. Na listingu 4.6 została przedstawiona implementacja ekstrakcji emocji z treści recenzji.

```
def _get_emotions_from_text_content(self) -> Tuple[List[str], List[List[float]]]:
        emotions_columns: List[List[float]] = []
        for offer in self.offers:
            reviews_emotions: List[Dict] = [
                NRCLex(self._prepare_text(review.text_content)).affect_frequencies
                for review in offer.reviews
            ]
            if len(reviews_emotions) == 0:
                emotions_columns.append(list(np.zeros(len(AFFECT_FREQUENCIES_KEY))))
                continue
            mean: pd.Series = pd.DataFrame(data=[emotions.values() for emotions in
                reviews_emotions]).mean()
            emotions_columns.append(mean.to_list())
        emotions_columns_ndarray = np.array(emotions_columns)
        rotated_emotions_columns = [
            list(emotions_columns_ndarray[:, index])
            for index in range(emotions_columns_ndarray.shape[1])
        ]
        return AFFECT_FREQUENCIES_KEY, rotated_emotions_columns
```

Listing 4.6: Implementacja ekstrakcji emocji z treści recenzji

Na rysunku 4.6 został przedstawiony diagram czynności wykonywanych w ramach wstępnego przetwarzania danych oraz ekstracji cech, które zostało opisane w tej sekcji.




Rysunek 4.6: Diagram czynności wstępnego przetwarzania danych oraz ekstracji cech

## 4.4 Implementacja autorskiej metody

Zgodnie z opisem przedstawionym w sekcji 3.3 autor opracował dwie bliźniacze metody, które różnią się sposobem klasteryzacji (grupowania). Pierwszą z nich wykorzystuje algorytm K-Means, natomiast druga algorytm C-Means, ich implemetacje zostały przedstawione odpowiednio na listingach 4.7 oraz 4.8. Do tego celu zostały wykorzystane biblioteki, w przypadku algorytmu K-Means została wykorzystana biblioteka *scikit-learn*, przedstawionej w sekcji 4.1.2.4, natomiast w drugim przypadku, jakim jest algorytm C-Means autor wykorzystał bibliotekę *fuzzy-c-means*[23].

```
1 class KMeansClusterizer(MeansClusterizer):
2
3     def perform_means_clusterization(self, dataset: pd.DataFrame) -> None:
4         k_means: KMeans = KMeans(
5             n_clusters=MeansClusterizer.K_PARAM, random_state=Clusterizer.RANDOM_STATE
6         )
7         self.cluster_labels: np.ndarray = k_means.fit_predict(dataset)
```

Listing 4.7: Implementacja wykorzystująca algorytm K-Means

```
1 class FuzzyCMeansClusterizer(MeansClusterizer):
2
3     def perform_means_clusterization(self, dataset: pd.DataFrame) -> None:
4         fcm = FCM(n_clusters=MeansClusterizer.K_PARAM)
5         fcm.random_state = Clusterizer.RANDOM_STATE
6         fcm.fit(dataset.to_numpy())
7         self.cluster_labels: np.ndarray = fcm.predict(dataset.to_numpy())
```

Listing 4.8: Implementacja wykorzystująca algorytm C-Means

Po dokonaniu klasteryzacji są przeprowadzane kroki wspólne dla obu algorytmów, a mianowicie jest to rozdzielanie obiektów klasy *Offer* na dwie listy, ze względu na przydzielony klaster podczas grupowania. Posiadając takie dwie listy dokonywany jest proces wyboru klastra z bardziej wiarygodnymi ofertami. Jest on oparty wartość *średniej z wszystkich ocen produktu*, implementacja została przedstawiona na listingu 4.9. Uzyskany w ten sposób zbiór informacji zawiera dwie listy ofert oraz dwie zmienne typu boolean, zawierające wartość Prawda (ang. True) lub Fałsz (ang. False), odpowiednio dla każdej z list o tym czy ofert zawarte w niej są wiarygodne, czy też nie.

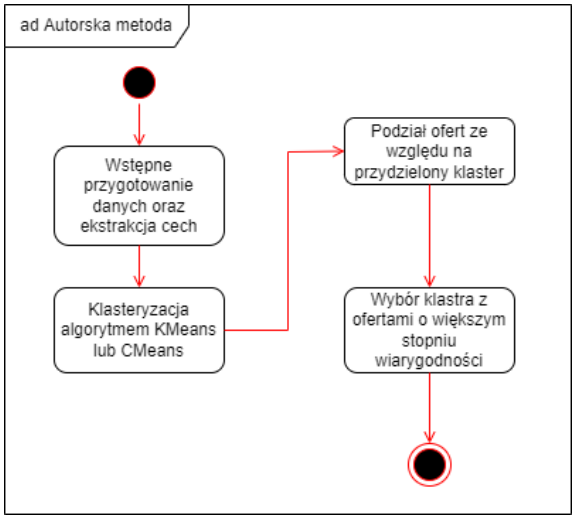
```
def _choose_list_with_more_credible_offers(
        self, combined_offers: Tuple[List[Offer], List[Offer]]
) -> Tuple[Tuple[List[Offer], bool], Tuple[List[Offer], bool]]:
    first_offers, second_offers = combined_offers
    first_average_stars_number: float = self._average_stars_number_for_offers_list(
        first_offers)
    second_average_stars_number: float = self._average_stars_number_for_offers_list(
        second_offers)
    if first_average_stars_number > second_average_stars_number:
        result = (first_offers, True), (second_offers, False)
    elif first_average_stars_number < second_average_stars_number:
        result = (second_offers, True), (first_offers, False)
    else:
        raise ChoosingCredibleOfferNotPossibleException()
    return result

def _average_stars_number_for_offers_list(self, offers: List[Offer]) -> float:
    offers_mean_sum = np.sum(
        [self._calculate_reviews_mean(offer.reviews) for offer in offers]
    )
    return round(offers_mean_sum / len(offers), 2)

def _calculate_reviews_mean(self, reviews: List[ProductReview]) -> int:
    return pd.Series([review.stars_number for review in reviews]).mean()
```

Listing 4.9: Implementacja wyboru bardziej wiarygodnego klastra

Na rysunku 4.7 został przedstawiony diagram czynności wykonywanych w czasie działania algorytmu autorskiej metody, które zostały opisane w tej sekcji.




Rysunek 4.7: Diagram czynności autorskiej metody

## 4.5 Implementacja wybranej metody z literatury

Do porównania autorskiej metody zostało wybrane rozwiązanie zaproponowane w artykule, opisane w sekcji 2.3. Wybór ten został podyktowany faktem, że zbiór dostępnych i wykorzystywanych informacji na temat oferty charakteryzuję się bardzo wysokim podobieństwem między dwiema metodami. Dzięki temu była możliwość bezpośredniego porównania na ofertach z portali ogłoszeniowych celem uzyskania możliwie jak najbardziej wiarygodnych wyników badań.

Pierwszym krokiem, jaki jest wykonywany, jest określenie czy podana wartość numeryczna ma odzwierciedlenie w znaczeniu semantycznym tekstu, a szczególnie w tym, czy opinia jest pozytywna, czy negatywna. Jest to dokonywane poprzez sprowadzenie wartości numerycznych do zakresu od *0* do *1*, następnie w wyniku analizy semantycznej uzyskuje wynik w jakim stopniu tekst ten jest pozytywny lub negatywny. Mając ten wynik, sprowadzany jest on też do zakresu od *0* do *1*. Na podstawie porównania uzyskanych wartości jest podejmowana decyzja czy opinia ta trafia do dalszej analizy. Współczynniki do porównywania wspomnianych wartości są parametryzowalne, dzięki czemu istnieje możliwość dopasowania celem uzyskania lepszych wyników. Następnie dla każdej z ofert wyliczana jest średnia z wartości numerycznych opinii, które trafiły do dalszej analizy. Implementacja przedstawionych kroków została przedstawiona na listingu 4.10.

Drugim krokiem jest wykonanie operacji w algorytmie, którego celem jest podział ofert na te wiarygodne oraz te niezasługujące na zaufania. Jest to dokonywane poprzez rozdział ofert na dwie grupy, na podstawie porównania wartości numerycznej z parametryzowanym progiem. Próg ten można dostosowywać do wybranego zbioru danych celem uzyskania możliwie jak najlepszych wyników. Wspomniana wartość numeryczna jest średnia opinii wyliczoną w kroku pierwszym dla każdej z ofert. Uzyskane tak wyniki można przekazać użytkownikowi w celu rekomendacji lub odradzenia przeprowadzania transakcji.

```
def calculate_score(self) -> BenchmarkFeatureExtractor:
    self._fix_not_valid_reviews()
    for offer in self.offers:
        normalized_stars_numbers = self._normalize_array(
            np.array([review.stars_number for review in offer.reviews]),
            MIN_MAX_REVIEW_VALUE[0], MIN_MAX_REVIEW_VALUE[1]
        )
        credible_reviews: List[ProductReview] = [
            review for stars_number, review in zip(normalized_stars_numbers, offer.
                reviews)
            if self._is_credible_review(stars_number, review.text_content)
        ]
        offer.reviews = credible_reviews
        score = pd.Series([review.stars_number for review in offer.reviews]).mean()
        self.dataset.append((offer, score))
    return self

def _is_credible_review(self, normalized_stars_number: int, text_content: str) -> bool:
    if text_content == "":
        return True
    polarity: float = TextBlob(self._prepare_text(text_content)).sentiment.polarity
    normalized_polarity: float = self._normalize_single_value_on_range(polarity, -1, 1)
    left_side = normalized_polarity - self.polarity_threshold
    right_side = normalized_polarity + self.polarity_threshold
    return left_side < normalized_stars_number < right_side
```

Listing 4.10: Implementacja algorytmu weryfikacji zgodności wartości numerycznej z wynikiem analizy semantycznej

```
def evaluate(self) -> Tuple[Tuple[Tuple[List[Offer], bool], Tuple[List[Offer], bool]],
    Statistics]:
    credible_offers: Tuple[List[Offer], bool] = ([], True)
    not_credible_offers: Tuple[List[Offer], bool] = ([], False)
    for offer, score in self.dataset:
        if score > self.credibility_threshold:
            credible_offers[0].append(offer)
        else:
            not_credible_offers[0].append(offer)
    result = (credible_offers, not_credible_offers)
    offers_count: int = len(credible_offers[0]) + len(not_credible_offers[0])
    return result, Statistics(offers_count)
```

Listing 4.11: Implementacja algorytmu oceny wiarygodności ofert na podstawie ocen

## 4.6 Maszyna wykorzystywana do eksperymentów

Do przeprowadzenia eksperymentów został wykorzystany komputer stacjonarny o specyfikacji przedstawione w tabeli 4.1. Systemem operacyjnym, jaki został wykorzystany na wspomnianej maszynie był *Microsoft Windows 10 Pro*[24] w wersji *10.0.19042 N/A Build 19042* o architekturze 64-bitowej.

Tabela 4.1: Specyfikacja maszyny do wykonywania eksperymentów

| | |
|---|---|
| Płyta główna | Gigabyte GA-EP45-DS4 |
| Procesor | Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz 2.67 GHz |
| Karta graficzna | NVIDIA GEFORCE GT 640 |
| Pamięć RAM | DDR2 Kingston 8GB 800mhz |
| Dysk twardy | Kingston SATA SSD A400 240GB 2,5” |

## 4.7 Zbiory danych wykorzystane do badań

W ramach przeprowadzonych badań, przedstawionych w rozdziale 5 zostały wybrane zbiory dane do eksperymentów. Zostały one stworzone na podstawie ofert z portalu eBay[25].

Portal ten w dosyć specyficzny i nie do końca konsekwentny sposób wyświetla dodawane opinie. Pomimo faktu wyszukiwania jednego, konkretnego i nowego produktu katalogowego, część ofert ma dokładnie takie same opinie, pomimo że są od innych sprzedawców. Niestety nie jest to wykonane konsekwentnie, gdyż można zauważyć pewne podgrupy ofert współdzielące te opinie, niektóre oferty mają opinie niepowtarzające się nigdzie indziej. Ze względu na fakt, że jednym z głównych elementów, jakie są analizowane są opinie autor niniejszej pracy zdecydował, że wybieranie do zbioru danych ofert z dokładnie takimi samymi oferta nie jest sensowne. Zdecydował on aby z każdej grupy ofert gdzie powtarzają się opinie wybrać zaledwie po dwie do trzech ofert.

Co więcej, taki zabieg pomoga przy przeprowadzaniu badań ponieważ każdą z ofert musi ocenić człowiek aby możliwa była weryfikacja skuteczności. W przypadku zbiorów danych zawierających np. setki ofert jest to proces bardzo czasochłonny i żmudny. Oczywiście w przypadku realnego użycia metody nie trzeba wykonywać

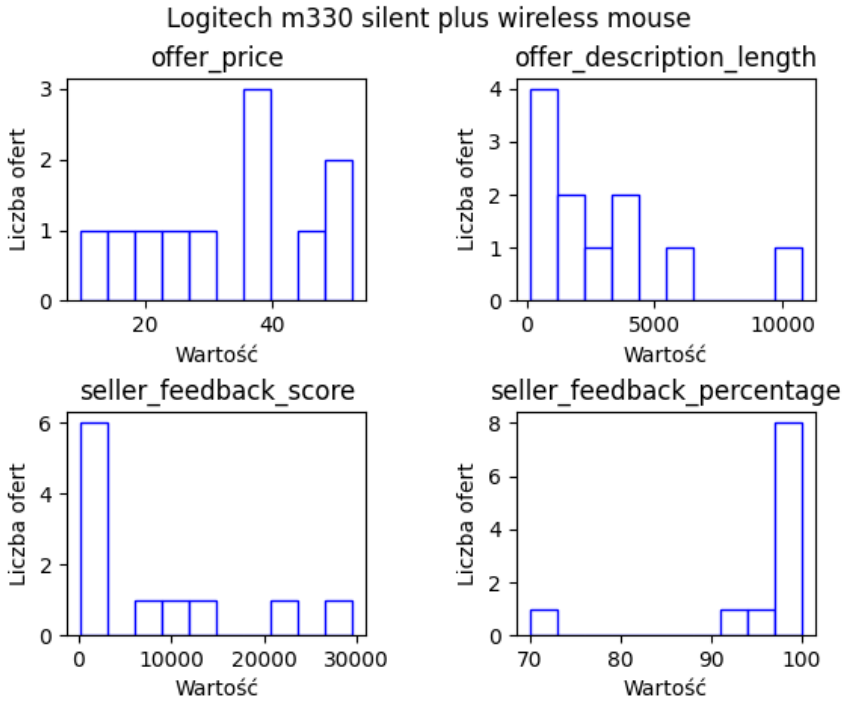
ręcznej oceny każdej oferty więc bez problemu można dokonywać weryfikacji setek ofert.

Z obserwacji autora niniejszej pracy wynika, że na tym portalu większość ofert produktów katalogowych ma pozytywne opinie. W przypadku wykorzystanych zbiorów danych opinie również są głównie pozytywne, co wpływa, że ich średnia jest zbliżona do wartości *4.5*. Nie mniej została podjęta decyzja o korzystaniu z tej cechy, gdyż w założeniach algorytmu jest to jedna z ważniejszych cech. Tym bardziej że metoda literaturowa wykorzystuję tę wartość do określania czy opinia jest wiarygodna. W poniższych sekcjach została przedstawiona charakterystyka każdego ze zbiorów.

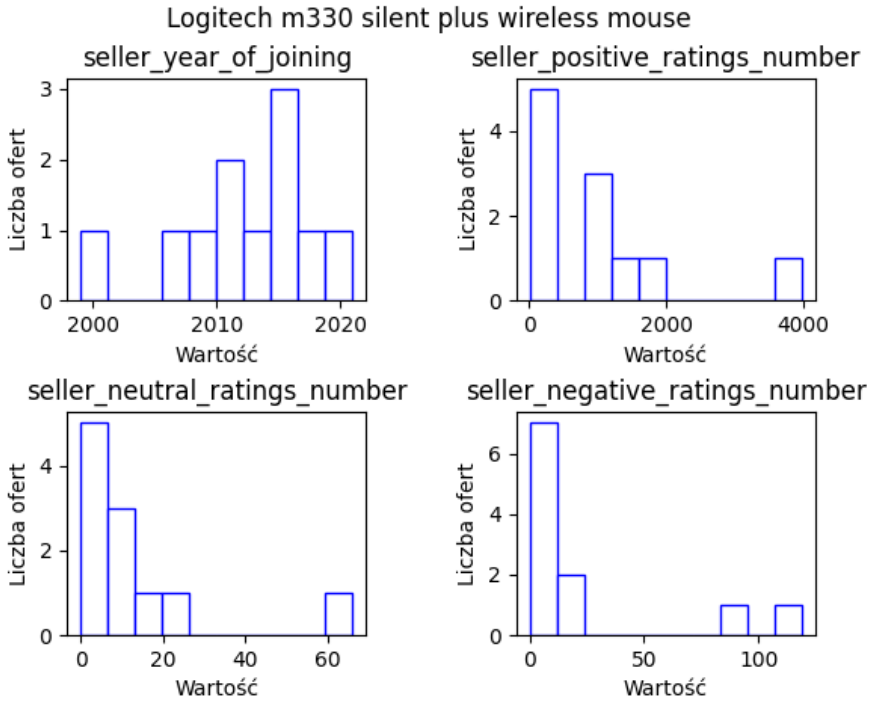
### 4.7.1 Zbiór danych #1

Tabela 4.2: Podstawowe informacje o zbiorze danych #1

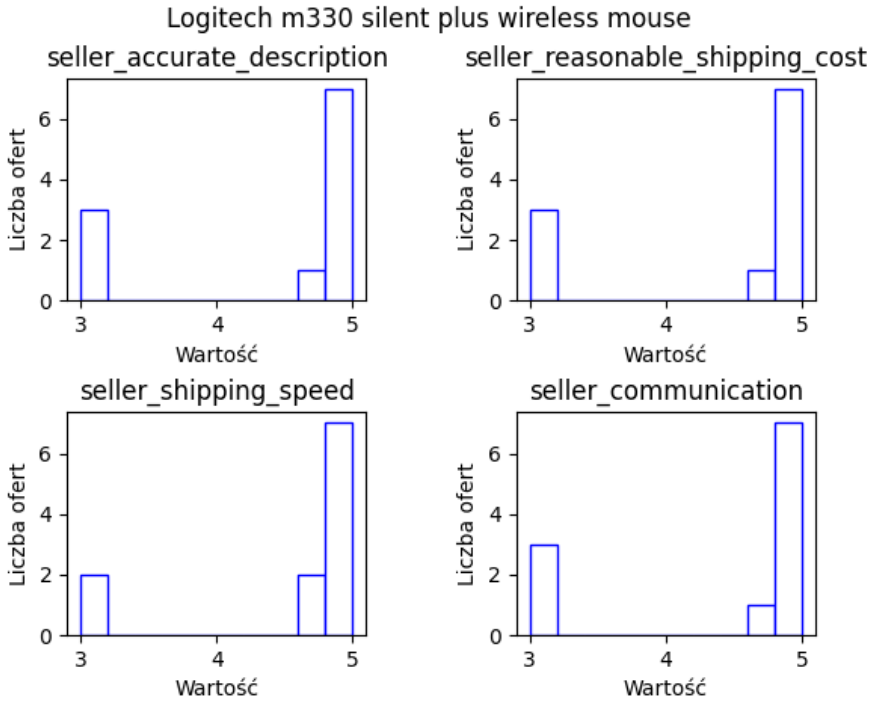
| Nazwa | Wartość |
|---|---|
| Nazwa katalogowa | Logitech m330 silent plus wireless mouse |
| Liczba ofert | 11 |
| Liczba ofert określona jako wiarygodna przez eksperta | 5 |
| Liczba ofert określona jako niewiarygodna przez eksperta | 6 |



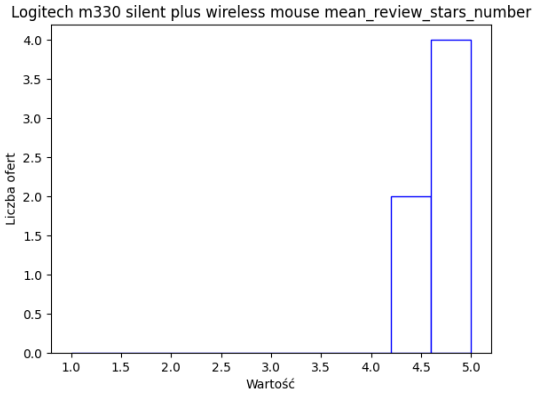

Rysunek 4.8: Histogramy wybranych cech oferty dla zbioru danych #1, cz. 1




Rysunek 4.9: Histogramy wybranych cech oferty dla zbioru danych #1, cz. 2




Rysunek 4.10: Histogramy wybranych cech oferty dla zbioru danych #1, cz. 3

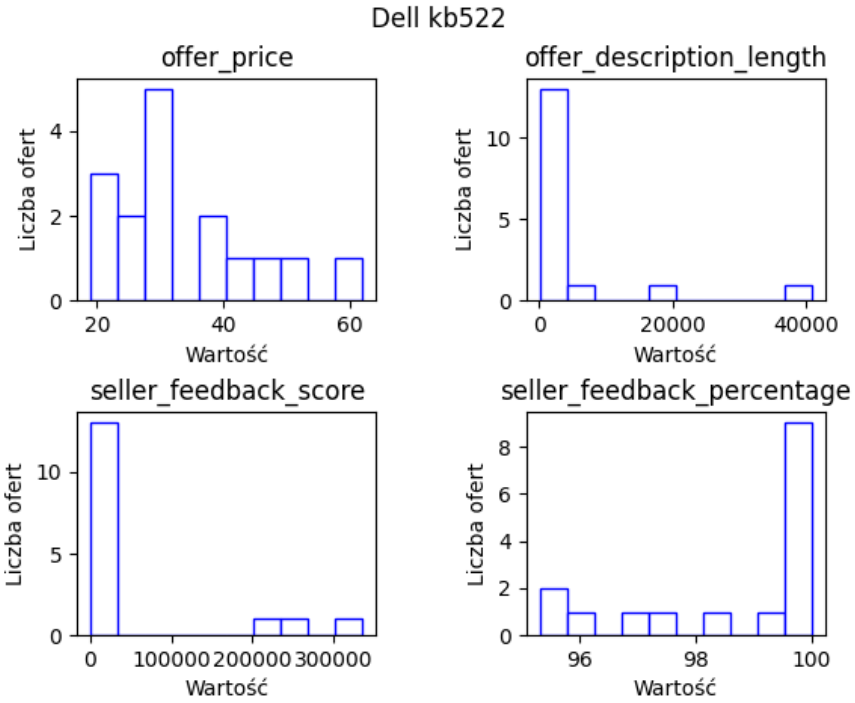



Rysunek 4.11: Histogram średniej wartości oceny dla oferty, zbiór danych #1

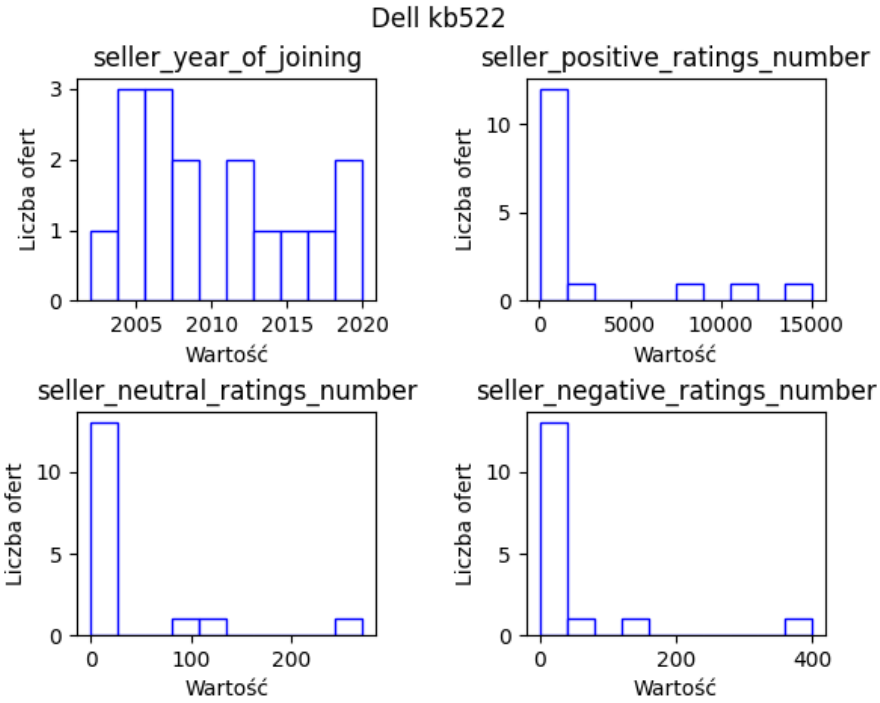
### 4.7.2 Zbiór danych #2

Tabela 4.3: Podstawowe informacje o zbiorze danych #2

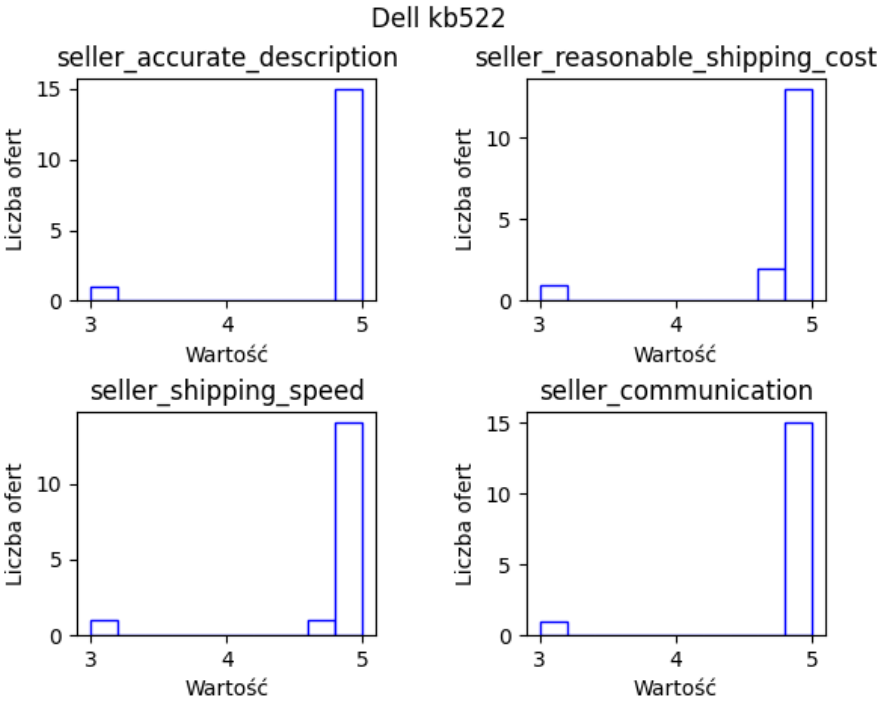
| Nazwa | Wartość |
|---|---|
| Nazwa katalogowa | Dell kb522 |
| Liczba ofert | 16 |
| Liczba ofert określona jako wiarygodna przez eksperta | 9 |
| Liczba ofert określona jako niewiarygodna przez eksperta | 7 |



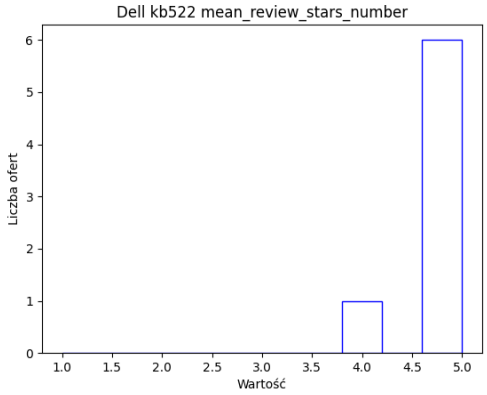

Rysunek 4.12: Histogramy wybranych cech oferty dla zbioru danych #2, cz. 1




Rysunek 4.13: Histogramy wybranych cech oferty dla zbioru danych #2, cz. 2




Rysunek 4.14: Histogramy wybranych cech oferty dla zbioru danych #2, cz. 3

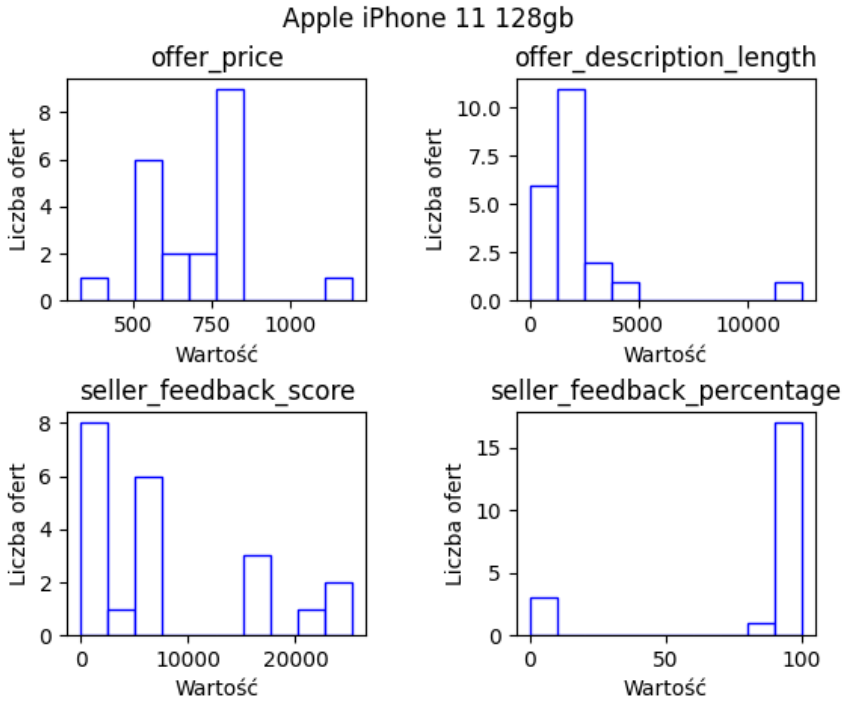



Rysunek 4.15: Histogram średniej wartości oceny dla oferty, zbiór danych #2

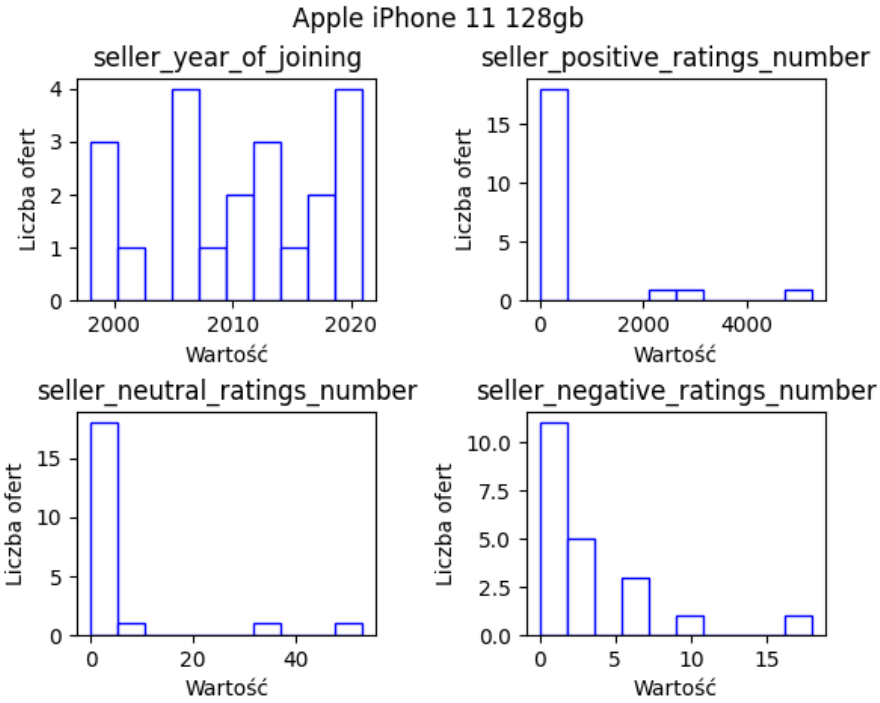
### 4.7.3 Zbiór danych #3

Tabela 4.4: Podstawowe informacje o zbiorze danych #3

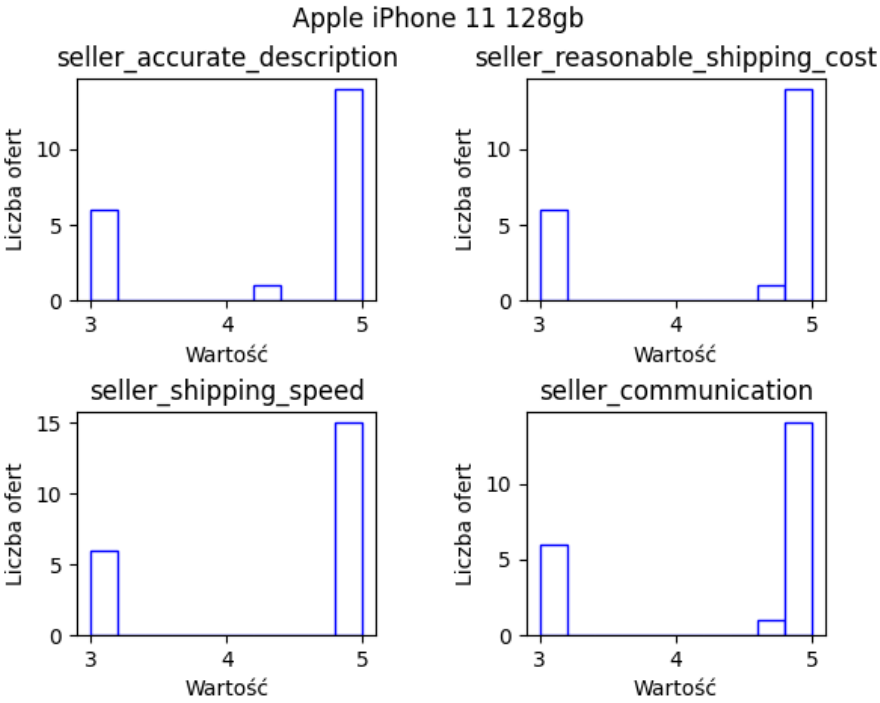
| Nazwa | Wartość |
|---|---|
| Nazwa katalogowa | Apple iPhone 11 128gb |
| Liczba ofert | 21 |
| Liczba ofert określona jako wiarygodna przez eksperta | 10 |
| Liczba ofert określona jako niewiarygodna przez eksperta | 11 |



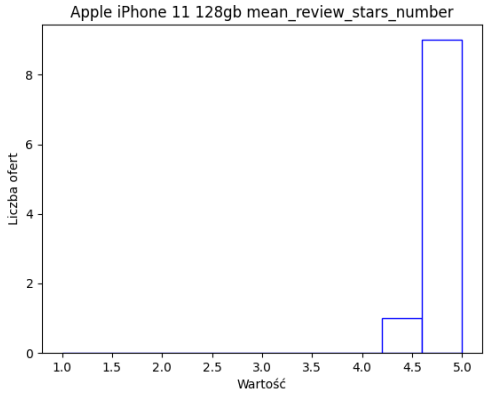

Rysunek 4.16: Histogramy wybranych cech oferty dla zbioru danych #3, cz. 1




Rysunek 4.17: Histogramy wybranych cech oferty dla zbioru danych #3, cz. 2




Rysunek 4.18: Histogramy wybranych cech oferty dla zbioru danych #3, cz. 3

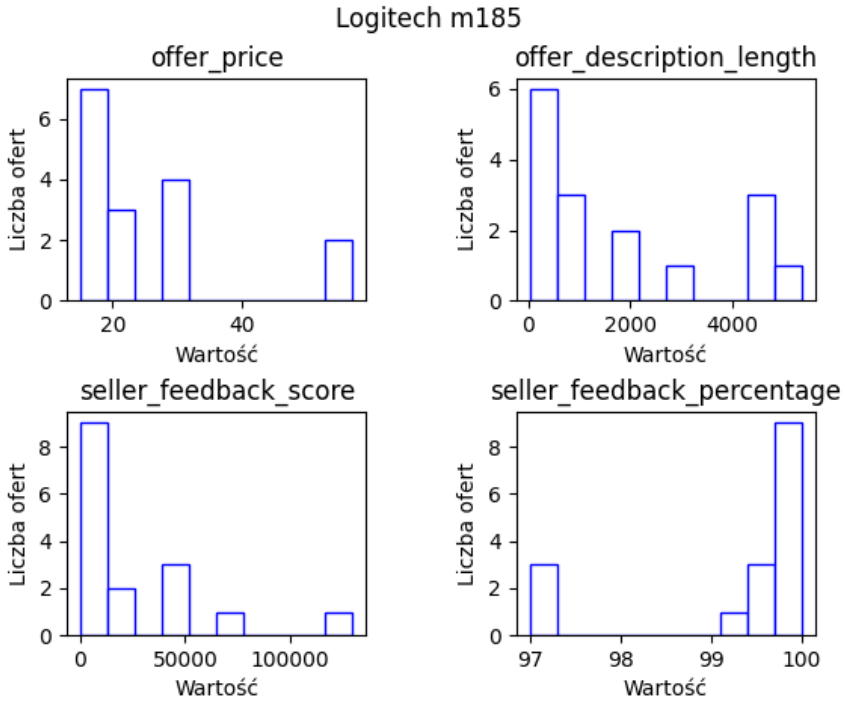



Rysunek 4.19: Histogram średniej wartości oceny dla oferty, zbiór danych #3

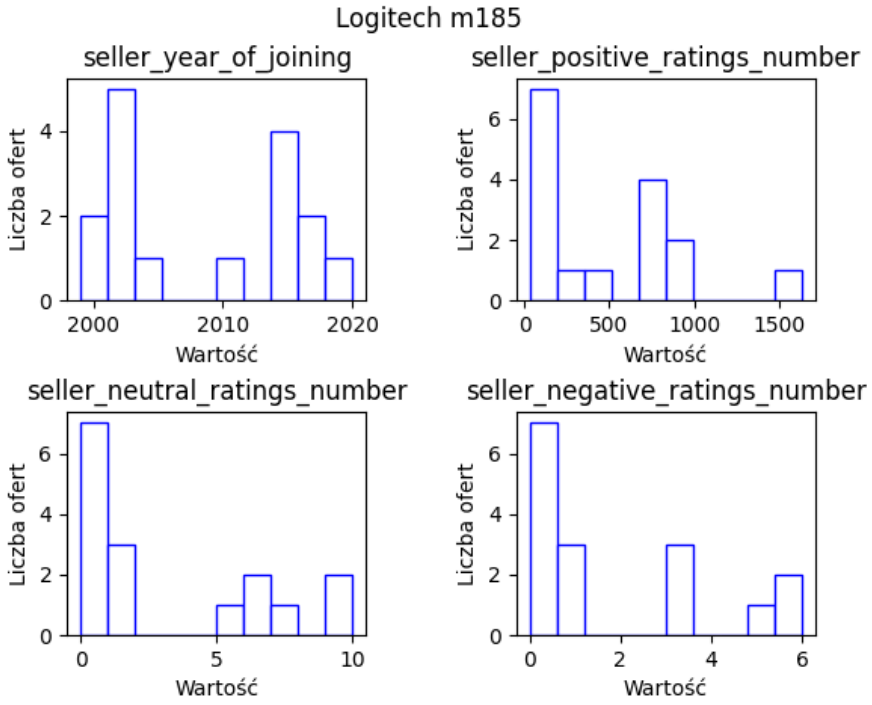
### 4.7.4 Zbiór danych #4

Tabela 4.5: Podstawowe informacje o zbiorze danych #4

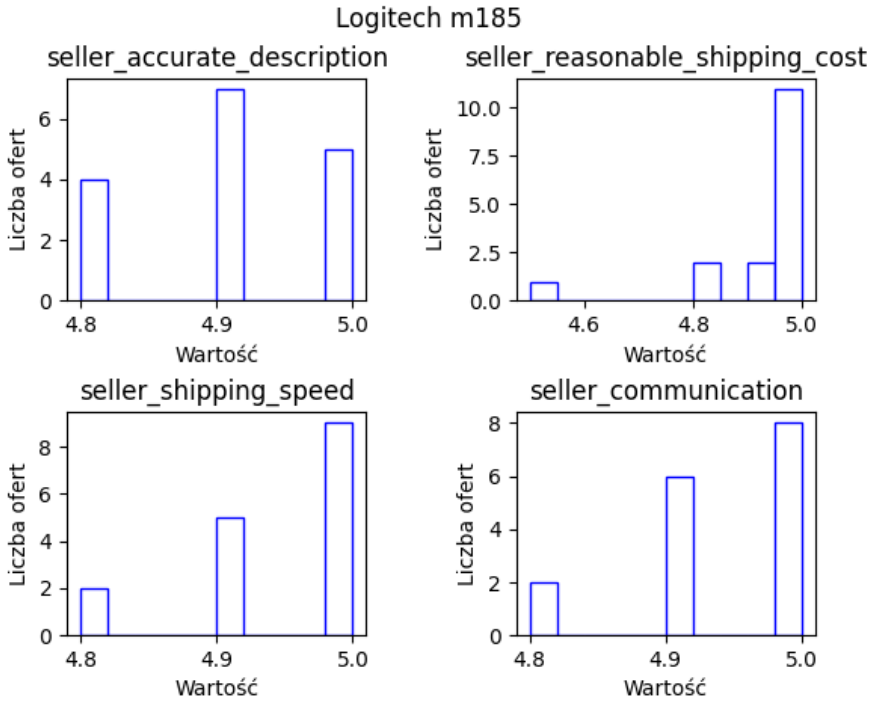
| Nazwa | Wartość |
|---|---|
| Nazwa katalogowa | Logitech m185 |
| Liczba ofert | 16 |
| Liczba ofert określona jako wiarygodna przez eksperta | 5 |
| Liczba ofert określona jako niewiarygodna przez eksperta | 11 |



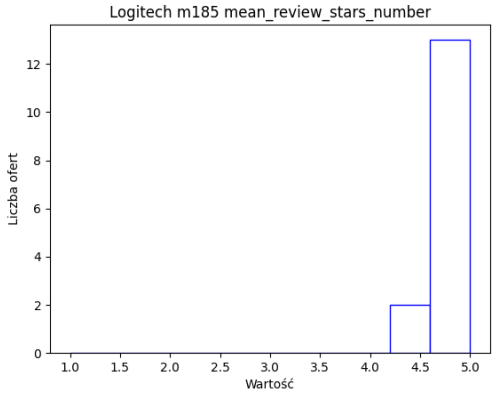

Rysunek 4.20: Histogramy wybranych cech oferty dla zbioru danych #4, cz. 1




Rysunek 4.21: Histogramy wybranych cech oferty dla zbioru danych #4, cz. 2




Rysunek 4.22: Histogramy wybranych cech oferty dla zbioru danych #4, cz. 3




Rysunek 4.23: Histogram średniej wartości oceny dla oferty, zbiór danych #4

### 4.7.5 Znaczące różnice między zbiorami danych

Jedną z ciekawych obserwacji jest to, że w przypadku zbiorów #2 oraz #4 istnieją wartości *seller_feedback_score*, które są stukrotnie większe w porównaniu do tych występujących w zbiorach #1 oraz #3.

Co więcej, w przypadku zbiorów #2 oraz #4 wartości *seller_feedback_percentage* występują w bardzo wąskim zakresie od około 95 do 100 natomiast w pozostałych zbiorach mamy duży szerszy zakres wartości.

Warto też zwrócić uwagę na zbiór #4 gdzie wartości *seller_accurate_description*, *seller_reasonable_shipping_cost*, *seller_shipping_speed*, *seller_communication* są tylko z wysokiego przedziału, czyli głównie od 4,8 do 5. Pozostałe zbiory posiadają dla tych pól również mniejsze wartości.

# Rozdział 5

# Wyniki eksperymentów

W ramach niniejszej pracy magisterskiej zostały przeprowadzone badania w formie eksperymentów. Celem tego było określenie skuteczności stworzonej metody w stosunku do zaimplementowanej metody z literatury opisanej w sekcji 4.5. Do tego celu zostały wykorzystane zbiory danych opisane w sekcji 4.7. Dla każdej oferty ze wszystkich zbiorów danych została przeprowadzona ręczna oceną wiarygodności oferty przez człowieka. Krokami, jakie zostały wykonane w ramach eksperymentów są:

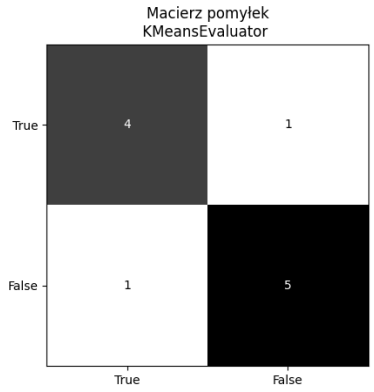
1. Wybór zbioru danych, w tym przypadku od #1 do #4
2. Przeprowadzanie eksperymentów dla każdego z poniższych metod celem oceny wiarygodności ofert
   (a) KMeansEvaluator - brak parametryzacji
   (b) FuzzyCMeansEvaluator - brak parametryzacji
   (c) BenchmarkEvaluator - parametry określone w nagłówku macierzy pomyłek
   (d) BenchmarkEvaluator - parametry określone w nagłówku macierzy pomyłek
3. W ramach każdego eksperymentu był mierzony czas wykonania oraz były uzyskiwane wyniki o liczbie ofert wiarygodnych oraz niegodnych zaufania
4. Uzyskane wyniki są porównywane z wartościami ręcznej oceny wiarygodności i na tej podstawie jest tworzona macierz pomyłek.

## 5.1 Zbiór danych #1

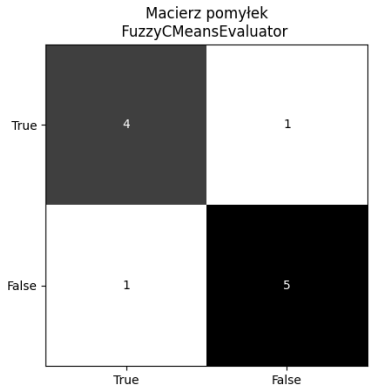
Do eksperymentów został wykorzystany zbiór przedstawiony w sekcji 4.7.1

Tabela 5.1: Statystyki dla zbioru danych #1

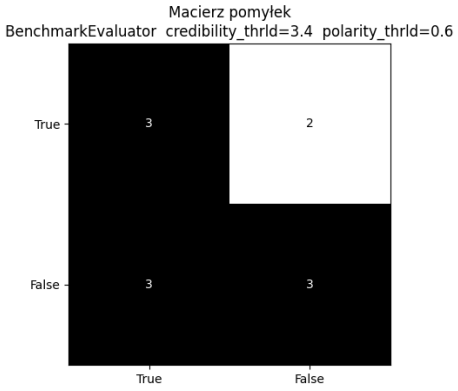
| Nazwa | Eksperyment #1 | Eksperyment #2 | Eksperyment #3 | Eksperyment #4 |
|---|---|---|---|---|
| Nazwa algorytmu | KMeansEvaluator | FuzzyCMeansEvaluator | BenchmarkEvaluator | BenchmarkEvaluator |
| Czas wykonania (s) | 6.125 | 2.485 | 2.422 | 2.328 |
| Liczba ofert określona jako wiarygodne | 5 | 5 | 6 | 4 |
| Liczba ofert określona jako niewiarogodne | 6 | 6 | 5 | 7 |



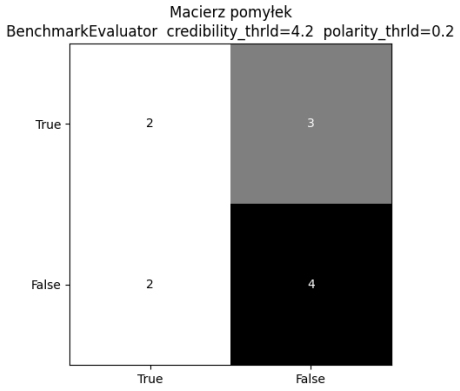

Rysunek 5.1: Macierz pomyłek dla zbioru danych #1, eksperyment #1




Rysunek 5.2: Macierz pomyłek dla zbioru danych #1, eksperyment #2




Rysunek 5.3: Macierz pomyłek dla zbioru danych #1, eksperyment #3




Rysunek 5.4: Macierz pomyłek dla zbioru danych #1, eksperyment #4

### 5.1.1 Podsumowanie uzyskanych wyników

Na podstawie powyższych wyników można stwierdzić, że dla tego zbioru danych najlepiej sprawdziła się autorska metoda, te same wyniki zostały uzyskane dla obu wariantów. Metody popełniły 2 błędy po jednym pierwszego i drugiego rodzaju. Czas wykonania był dla wariantu K-Means był znacznie wyższy niż dla C-Means.

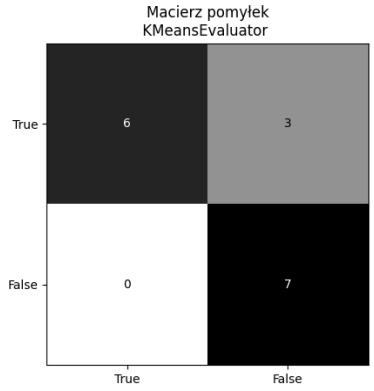
W przypadku metoda literaturowej to dla obu zestawów parametrów czasy wykonania były bardzo zbliżone. Jeżeli chodzi o wyniki to w eksperymencie #3 metoda popełniła 2 błędy pierwszego rodzaju i 3 błędy drugiego rodzaju natomiast w eksperymencie #4 metoda popełniła 3 błędy pierwszego rodzaju i 2 błędy drugiego rodzaju. Można stwierdzić, że skuteczność autorskiej metody jest akceptowalna.

## 5.2 Zbiór danych #2

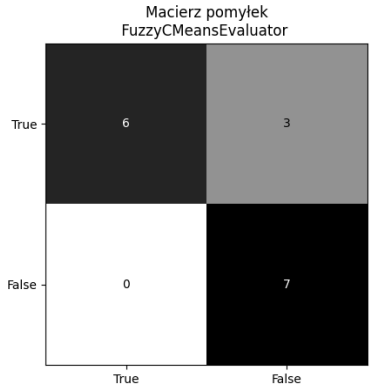
Do eksperymentów został wykorzystany zbiór przedstawiony w sekcji 4.7.2

Tabela 5.2: Statystyki dla zbioru danych #2

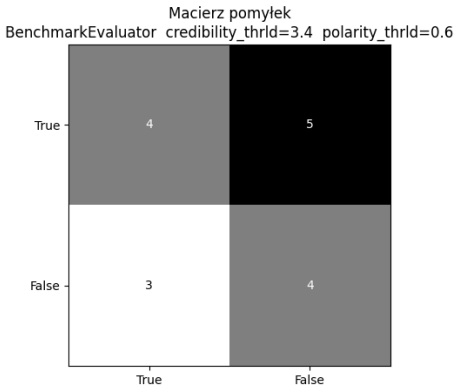
| Nazwa | Eksperyment #1 | Eksperyment #2 | Eksperyment #3 | Eksperyment #4 |
|---|---|---|---|---|
| Nazwa algorytmu | KMeansEvaluator | FuzzyCMeansEvaluator | BenchmarkEvaluator | BenchmarkEvaluator |
| Czas wykonania (s) | 4.553 | 0.615 | 0.641 | 0.585 |
| Liczba ofert określona jako wiarygodne | 6 | 6 | 7 | 4 |
| Liczba ofert określona jako niewiarogodne | 10 | 10 | 9 | 12 |



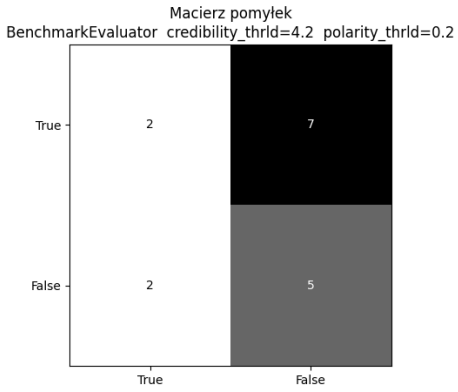

Rysunek 5.5: Macierz pomyłek dla zbioru danych #2, eksperyment #1




Rysunek 5.6: Macierz pomyłek dla zbioru danych #2, eksperyment #2




Rysunek 5.7: Macierz pomyłek dla zbioru danych #2, eksperyment #3




Rysunek 5.8: Macierz pomyłek dla zbioru danych #2, eksperyment #4

### 5.2.1 Podsumowanie uzyskanych wyników

Tak jak w przypadku zbioru danych #1 lepsze wyniki zwróciła autorska metoda, oba warianty zwróciły takie same wyniki, popełniając 3 błędy pierwszego rodzaju. Była za to siedmiokrotna różnica czasu wykonania na korzyść metody korzystającej z C-Means.

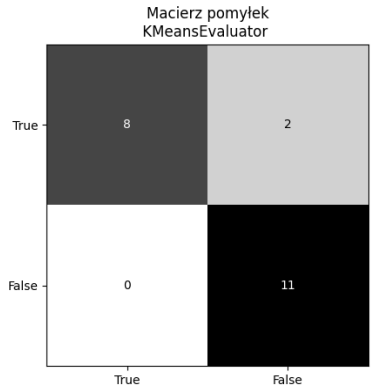
Metoda z literatury sprawdziła się gorzej, w eksperymencie #3 popełniła 5 błędów pierwszego rodzaju oraz 3 błędy drugiego rodzaju. W eksperymencie #4 popełniał 7 błędów pierwszego rodzaju i 2 błędy drugiego rodzaju, czyli sumarycznie 1 błąd więcej. Czasy wykonania były bardzo zbliżone. Można stwierdzić, że skuteczność autorskiej metody jest akceptowalna.

## 5.3 Zbiór danych #3

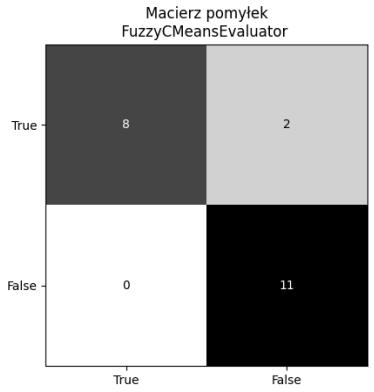
Do eksperymentów został wykorzystany zbiór przedstawiony w sekcji 4.7.3

Tabela 5.3: Statystyki dla zbioru danych #3

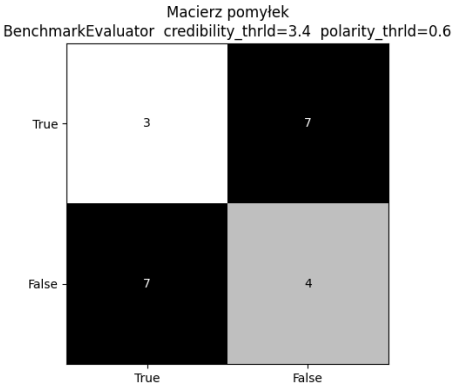
| Nazwa | Eksperyment #1 | Eksperyment #2 | Eksperyment #3 | Eksperyment #4 |
|---|---|---|---|---|
| Nazwa algorytmu | KMeansEvaluator | FuzzyCMeansEvaluator | BenchmarkEvaluator | BenchmarkEvaluator |
| Czas wykonania (s) | 5.875 | 2.11 | 2.0 | 1.922 |
| Liczba ofert określona jako wiarygodne | 8 | 8 | 10 | 8 |
| Liczba ofert określona jako niewiarogodne | 13 | 13 | 11 | 13 |



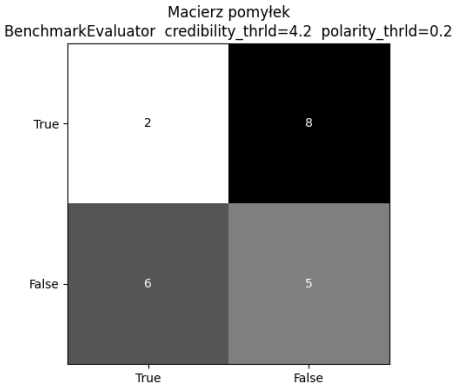

Rysunek 5.9: Macierz pomyłek dla zbioru danych #3, eksperyment #1




Rysunek 5.10: Macierz pomyłek dla zbioru danych #3, eksperyment #2




Rysunek 5.11: Macierz pomyłek dla zbioru danych #3, eksperyment #3




Rysunek 5.12: Macierz pomyłek dla zbioru danych #3, eksperyment #4

### 5.3.1 Podsumowanie uzyskanych wyników

Dla autorskiej metody uzyskane wyniki są analogiczne jak dla zbioru danych #2 jedynie liczba błędów pierwszego rodzaju zmniejszyła się o jeden i wynosi dwa. Czasy wykonania analogiczne do zbioru danych #1.

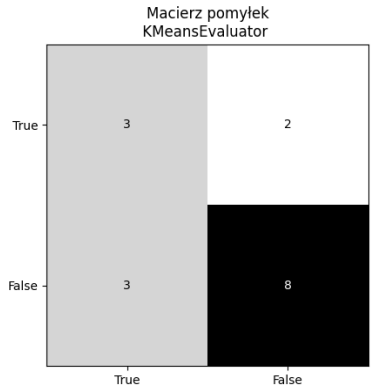
Metoda z literatury popełniła bardzo dużo błędów i wyniki nie są akceptowalne. Dla eksperymentu #3 było to 7 błędów pierwszego rodzaju i 7 błędów drugiego rodzaju natomiast dla eksperymentu #4 odpowiednio 8 i 6 błędów. Czasy wykonania bardzo zbliżone. Można stwierdzić, że skuteczność autorskiej metody jest akceptowalna.

## 5.4 Zbiór danych #4

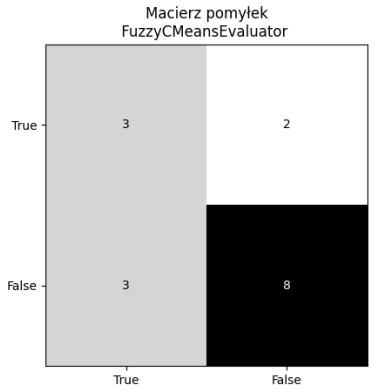
Do eksperymentów został wykorzystany zbiór przedstawiony w sekcji 4.7.4

Tabela 5.4: Statystyki dla zbioru danych #4

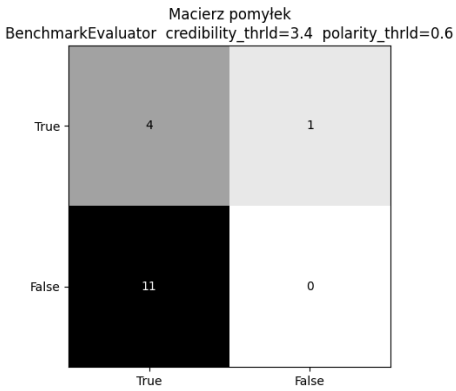
| Nazwa | Eksperyment #1 | Eksperyment #2 | Eksperyment #3 | Eksperyment #4 |
|---|---|---|---|---|
| Nazwa algorytmu | KMeansEvaluator | FuzzyCMeansEvaluator | BenchmarkEvaluator | BenchmarkEvaluator |
| Czas wykonania (s) | 13.266 | 10.189 | 8.907 | 8.469 |
| Liczba ofert określona jako wiarygodne | 6 | 6 | 15 | 10 |
| Liczba ofert określona jako niewiarogodne | 10 | 10 | 1 | 6 |



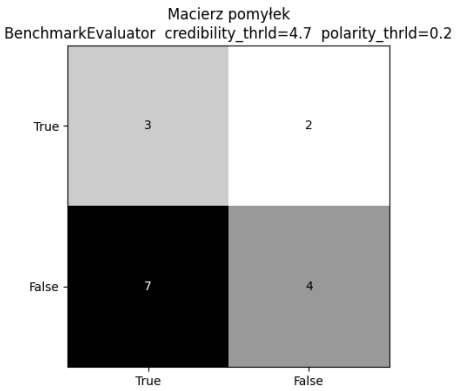

Rysunek 5.13: Macierz pomyłek dla zbioru danych #4, eksperyment #1




Rysunek 5.14: Macierz pomyłek dla zbioru danych #4, eksperyment #2




Rysunek 5.15: Macierz pomyłek dla zbioru danych #4, eksperyment #3




Rysunek 5.16: Macierz pomyłek dla zbioru danych #4, eksperyment #4

### 5.4.1 Podsumowanie uzyskanych wyników

Tak jak w poprzednich eksperymentach autorska metoda zwraca dla dwóch wariantów dokonała takich samych rekomendacji i popełniła taka samą liczbę błędów. Co więcej, występuje analogiczna różnica czasu na korzyść C-Means.

Metoda literaturowa uzyskała lepsze wyniki dla eksperymentu #4, gdyż popełniła mniej błędów a dokładnie 2 błędy pierwszego rodzaju oraz 7 błędów drugiego rodzaju. W przypadku eksperymentu #3 był tylko jeden błąd pierwszego rodzaju natomiast było aż 11 błędów drugiego rodzaju. Wynik metody literaturowej nie jest akceptowalny natomiast ten uzyskany autorską metodą zawiera błędy, lecz da się go zaakceptować.

## 5.5 Podsumowanie eksperymentów

Po przeanalizowaniu wyników uzyskanych w ramach eksperymentów dla wszystkich zbiorów można zauważyć tendencje, że opracowana metoda wykazuję się lepszą skutecznością niż ta zaimplementowana na podstawie literatury. Autor określił to po obserwacji macierzy pomyłek, metoda z literatury popełniała więcej błędów pierwszego i drugiego rodzaju. Oczywiście zaproponowana metoda również popełniała błędy i w niektórych przypadkach była ich niemała liczba. Nie mniej i tak było ich zauważalnie mniej niż w przypadku metody literaturowej. Warto dodać, że wartości określone przez człowieka są w większości przypadków poprawne, lecz człowiek też może podejmować różne decyzje stąd mogą występować rozbieżności. Ocenę człowieka nigdy nie można traktować jako prawdę absolutną lecz informacje te są niezwykle przydatne przy badaniu takich algorytmów i właśnie z nimi można porównywać uzyskane wyniki.

W przypadku metody literaturowej były przeprowadzane badania dla różnych wartości parametrów progów, które zostały opisane w sekcji 4.5, lecz do niniejszej pracy magisterskiej autor wybierał dwa najlepsze wyniki dla parametrów.

Po analizie autorskiej metody można zauważyć ciekawą kwestię, że dokonuje ona takich samych decyzji i popełnia takie same błędy. Warto jednak spojrzeć na różnice w czasach wykonania. Są one dosyć spore, w niektórych przypadkach nawet kilkukrotnie mniejsze, oczywiście na korzyść wersji wykorzystującej C-Means. Może wynikach to

z różnic w implementacji wykorzystanych bibliotek i modułów. Jedynym zadziwiającym przypadkiem są wyniki wykonania algorytmów dla zbioru danych #2 przedstawionego w sekcji 5.2, może to wynikać z kwestii sprzętowych lub jakieś ukrytej optymalizacji interpretera języka Python, na którą autor pracy nie ma wpływu.

W przypadku zbiorów danych o niewielkiej liczbie rekordów tak jak w przypadku przeprowadzanych eksperymentów, nie ma to praktycznie znaczenia w realnym użyciu. Duża różnica mogłaby być, gdyby zbiory miały, chociażby jeden milion rekordów. Nie mniej taka sytuacji raczej nie miałaby miejsca, gdyż na portalach ogłoszeniowych konkretny produkt katalogowy raczej nie występuję w tylu ofertach.

# Rozdział 6

# Podsumowanie i wnioski

Celem pierwszej części niniejszej pracy magisterskiej było wprowadzanie do tematyki dotyczącej zakupów przy użyciu internetowych portali ogłoszeniowych. Został przedstawiony fakt niesamowitego rozwoju tej formy sprzedaży na podstawie badań rynku z kilku poprzednich lat. Kolejne sekcje prezentowały niepodważalne zalety handlu w internecie, ale również nowe problemy i sytuacje, z którymi muszą się mierzyć ich użytkownicy.

Następne rozdziały prezentowały aktualny stan wiedzy w literaturze naukowej w obszarze dokonywania rekomendacji ofert, które można określić jako wiarygodne. Na podstawie ich analizy zostały przedstawione potencjalne obszary ulepszeń dostępnych rozwiązań. W oparciu o nie została opracowana autorska metoda oceny wiarygodności ofert na portalach ogłoszeniowych w dwóch wariantach. Została również przedstawiona hipoteza badawcza.

W kolejnym już czwartym z kolei, rozdziale zostało przedstawione środowisko eksperymentalne, dokładne informacje na temat implementacji autorskiej metody oraz wybranej metody z literatury. Co warto dodać zostały przedstawione i scharakteryzowane zbiory danych wykorzystane w badaniach. Co więcej, autor w szczegółowy sposób przedstawił sam proces przygotowywania danych i ekstrakcji cech, aby była możliwość wykorzystania ich w możliwie jak najlepszy sposób.

Następny rozdział przedstawia wykonane eksperymenty na wybranych zbiorach danych oraz uzyskane wyniki. Na ich podstawie można stwierdzić, że postawiona hipoteza badawcza okazała się prawdziwa, bo w przeprowadzonych eksperymentach

metoda wykazała się wyższą skutecznością niż wybrana metoda z literatury.

Tematyka dotycząca systemów rekomendacji nakierowanych na dokonywanie rekomendacji w branży e-commerce z pewnością w następnych latach będzie notowała duże przyrosty oraz będzie stawała się coraz bardziej popularna, tak samo z resztą jak same portale ogłoszeniowe. Zdaniem autora niniejszej pracy dyplomowej, przedstawiona autorska metoda jest kolejny krokiem naprzód w ulepszaniu wyspecjalizowanych systemów rekomendacji. Bardzo możliwe, że w najbliższych latach nastąpi swojego rodzaju przełom i same portale ogłoszeniowe będą zapewniać więcej informacji, na podstawie, których analiza i rekomendacje będą mogły być jeszcze bardziej wyrafinowane i złożone. Dzięki temu ich skuteczność będzie jeszcze większa a co za tym idzie, wygoda użytkowników takich portali wstąpi na zupełnie inny poziom.

# Spis rysunków

# Spis listingów

# Spis tabel

# Bibliografia


[1] statista.com. *Retail e-commerce sales worldwide from 2014 to 2025*. [online]. [dostęp: 01.03.2022]. Dostępny w Internecie, URL: `https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/`.

[2] Google LLC. *Recommendation Systems*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: `https://developers.google.com/machine-learning/recommendation`.

[3] Google LLC. *Google LLC*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: `https://www.google.pl/`.

[4] Microsoft Corporation. *Microsoft Corporation*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: `https://www.microsoft.com/pl-pl`.

[5] Harshal D. Dalvi, Abhijit Joshi i Narendra Shekokar. *Trustworthiness evaluation system in E-Commerce context*. IEEE, 2016. ISBN: 978-1-5090-3291-4.

[6] Hasnae Rahimi i Hanan El Bakkali. *A new reputation algorithm for evaluating trustworthiness in e-commerce context*. IEEE, 2013. ISBN: 978-1-4799-0324-5.

[7] Achraf Boumhidi, Abdessamad Benlahbib i El Habib Nfaoui. *Cross-Platform Reputation Generation System Based on Aspect-Based Sentiment Analysis*. IEEE, 2022. ISBN: 2169-3536.

[8] Inc. Meta Platforms. *Facebook*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: `https://www.facebook.com/`.

[9] Inc. Meta Platforms. *Instagram*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: `https://www.instagram.com/`.

[10] Inc. Twitter. *Twitter*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: https://twitter.com/.

[11] Google LLC. *YouTube*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: https://www.youtube.com/.

[12] TripAdvisor LLC. *TripAdvisor*. [online]. [dostęp: 07.03.2022]. Dostępny w Internecie, URL: https://www.tripadvisor.com/.

[13] Eric Oti i in. *Comprehensive Review of K-Means Clustering Algorithms*. T. 07. Sty. 2021, s. 64–69. DOI: 10.31695/IJASRE.2021.34050.

[14] James C. Bezdek, Robert Ehrlich i William Full. „FCM: The fuzzy c-means clustering algorithm". W: *Computers  Geosciences* 10.2 (1984), s. 191–203. ISSN: 0098-3004.

[15] Yufeng. *Fuzzy C-Means Clustering*. [online]. [dostęp: 12.03.2022]. Dostępny w Internecie, URL: https://towardsdatascience.com/fuzzy-c-means-clustering-with-python-f4908c714081.

[16] Python Software Foundation. *Python*. [online]. [dostęp: 16.03.2022]. Dostępny w Internecie, URL: https://www.python.org/.

[17] Numpy developers. *Numpy*. [online]. [dostęp: 16.03.2022]. Dostępny w Internecie, URL: https://numpy.org/.

[18] Pandas development team. *Pandas*. [online]. [dostęp: 16.03.2022]. Dostępny w Internecie, URL: https://pandas.pydata.org/.

[19] Matplotlib development team. *Matplotlib*. [online]. [dostęp: 16.03.2022]. Dostępny w Internecie, URL: https://matplotlib.org/stable/index.html.

[20] scikit-learn developers. *scikit-learn*. [online]. [dostęp: 16.03.2022]. Dostępny w Internecie, URL: https://scikit-learn.org/stable/index.html.

[21] NLTK development team. *NLTK*. [online]. [dostęp: 16.03.2022]. Dostępny w Internecie, URL: https://www.nltk.org/.

[22] Mark M. Bailey. *NRCLex*. [online]. [dostęp: 23.03.2022]. Dostępny w Internecie, URL: https://pypi.org/project/NRCLex/.

[23] Madson Dias. *fuzzy-c-means*. [online]. [dostęp: 12.05.2022]. Dostępny w Internecie, URL: https://pypi.org/project/fuzzy-c-means/.

[24] Microsoft Corporation. *Windows*. [online]. [dostęp: 20.05.2022]. Dostępny w Internecie, URL: https://www.microsoft.com/pl-pl/windows/.

[25] eBay Inc. *eBay*. [online]. [dostęp: 23.05.2022]. Dostępny w Internecie, URL: https://www.ebay.com/.