Uniwersytet Rzeszowski

SKLEP Z ELEKTRONIKĄ

Bazy danych

Mateusz Hołyszko
2024-06-03

Spis treści

Wprowadzenie

Projekt dotyczy stworzenia sklepu z elektroniką funkcjonującego ze strony użytkownika. Główną funkcjonalnością projektu jest system „inteligentnego" polecania artykułów na podstawie: statystyki sprzedaży produktu, ocen i komentarzy użytkowników, historii zakupów użytkownika.

Rekomendacja zostanie przyznana na podstawie parametrów złożonych z współczynników wygenerowanych na podstawie trzech algorytmów:

Algorytmy Rekomendacyjne

Algorytm 1: Analiza sprzedaży i ocen

Algorytm oblicza wynik trafności na podstawie liczby sprzedanych produktów i średniej oceny użytkowników produktów w określonej kategorii. Algorytm porównuje dany produkt z innymi produktami w tej samej kategorii, ustanawiając punkty odniesienia dla sprzedaży i ocen. Wyższy wynik jest przypisywany produktom o dobrych wynikach sprzedaży i wyższych średnich ocenach, sygnalizując ich popularność i jakość. Takie podejście pomaga zapewnić, że polecane produkty są nie tylko popularne pod względem sprzedaży, ale także dobrze odbierane przez innych użytkowników.

Algorytm 2: Feature Matching

Algorytm przyznający parametr „feature" użytkownikom i produktom na podstawie recenzji (w przypadku produktu) i historii sprzedaży (w przypadku użytkownika). Parametr „feature" jest wyłuskany z komentarzy na podstawie algorytmu lematyzującego. Z komentarzy algorytm wydobędzie kilka słów kluczy i na podstawie słownika przypisze najbardziej pasujący parametr „feature". Przykładami takich parametrów mogą być: „energy-efficient", „affordable", „quiet" itp. Jeżeli parametr „feature" jest zgodny między użytkownikiem a produktem to współczynnik będzie wprost-proporcjonalnie większy.

Algorytm 3: Historia zakupów i dopasowanie kategorii

Algorytm ten oblicza współczynnik trafności na podstawie historii zakupów użytkownika, koncentrując się w szczególności na kategoriach produktów, które użytkownik kupił w przeszłości. Identyfikując wzorce zachowań zakupowych użytkownika, algorytm może przewidzieć, którymi kategoriami użytkownik może być zainteresowany. Prowadzi to do bardziej ukierunkowanych i spersonalizowanych rekomendacji, które są zgodne z wcześniejszymi zainteresowaniami i nawykami zakupowymi użytkownika.

Funkcjonalności Systemu

1. **Rejestracja i uwierzytelnianie użytkowników:**
    - Użytkownicy mogą tworzyć konta, logować się i wylogowywać.
2. **Profile użytkowników:**

- Każdy użytkownik ma swój profil zawierający historię zakupów, preferencje i zapisane przedmioty.

3. **Recenzje i oceny produktów:**
   - Użytkownicy mogą zostawiać recenzje i oceny zakupionych produktów.
   - Recenzje mogą być tekstowe, a oceny są liczbowe (np. 1-5 gwiazdek).
4. **Wyszukiwanie i filtrowanie:**
   - Użytkownicy mogą wyszukiwać produkty za pomocą słów kluczowych i filtrować wyniki na podstawie kategorii, przedziałów cenowych i innych atrybutów.
5. **Koszyk i płatność:**
   - Użytkownicy mogą dodawać produkty do koszyka.
6. **Silnik rekomendacji:**
   - System „inteligentny” sugeruje produkty użytkownikom na podstawie ich preferencji, historii zakupów i recenzji innych użytkowników.
   - Silnik rekomendacji wykorzystuje algorytmy „k-means” do obliczania rekomendacji na podstawie danych sprzedaży, ocen, recenzji.

Struktura Bazy Danych

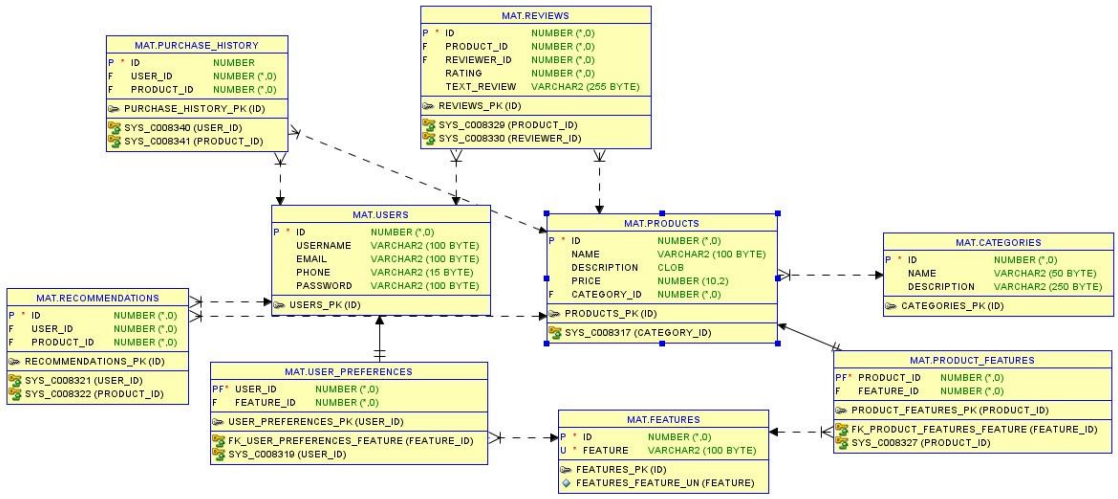

Figure 1 Erd bazy danych

1. **Tabela USERS:**
   - **ID:** Unikalny identyfikator użytkownika.
   - **USERNAME:** Nazwa użytkownika.
   - **EMAIL:** Adres email użytkownika.
   - **PHONE:** Numer telefonu użytkownika.
   - **PASSWORD:** Hasło użytkownika.
2. **Tabela PRODUCTS:**
   - **ID:** Unikalny identyfikator produktu.
   - **NAME:** Nazwa produktu.
   - **DESCRIPTION:** Opis produktu.
   - **PRICE:** Cena produktu.
   - **CATEGORY_ID:** Identyfikator kategorii produktu (klucz obcy).
3. **Tabela CATEGORIES:**

    - **ID:** Unikalny identyfikator kategorii.
    - **NAME:** Nazwa kategorii.
    - **DESCRIPTION:** Opis kategorii.
4. **Tabela REVIEWS:**
    - **ID:** Unikalny identyfikator recenzji.
    - **PRODUCT_ID:** Identyfikator recenzowanego produktu (klucz obcy).
    - **REVIEWER_ID:** Identyfikator recenzenta (klucz obcy).
    - **RATING:** Ocena produktu.
    - **TEXT_REVIEW:** Tekst recenzji.
5. **Tabela PURCHASE_HISTORY:**
    - **ID:** Unikalny identyfikator zakupu.
    - **USER_ID:** Identyfikator użytkownika (klucz obcy).
    - **PRODUCT_ID:** Identyfikator produktu (klucz obcy).
6. **Tabela USER_PREFERENCES:**
    - **USER_ID:** Identyfikator użytkownika (klucz obcy).
    - **FEATURE_ID:** Identyfikator preferencji użytkownika (klucz obcy).
7. **Tabela PRODUCT_FEATURES:**
    - **PRODUCT_ID:** Identyfikator produktu (klucz obcy).
    - **FEATURE_ID:** Identyfikator cechy produktu (klucz obcy).
8. **Tabela FEATURE_ID:**
    - Przechowuje kluczowe cechy które są przypisywane użytkownikom i produktom.

Szczegóły Implementacji

Procedury Składowane

Wszystkie operacje na bazie danych, takie jak tworzenie użytkowników, odczytywanie danych, aktualizacje i usuwanie, są wykonywane za pomocą procedur składowanych. Procedury te zostały zaimplementowane w SQL i mogą być wywoływane bezpośrednio z aplikacji Python. Poniżej znajduje się przykład procedury składowanej do zwracania produktów o zadanej kategorii:

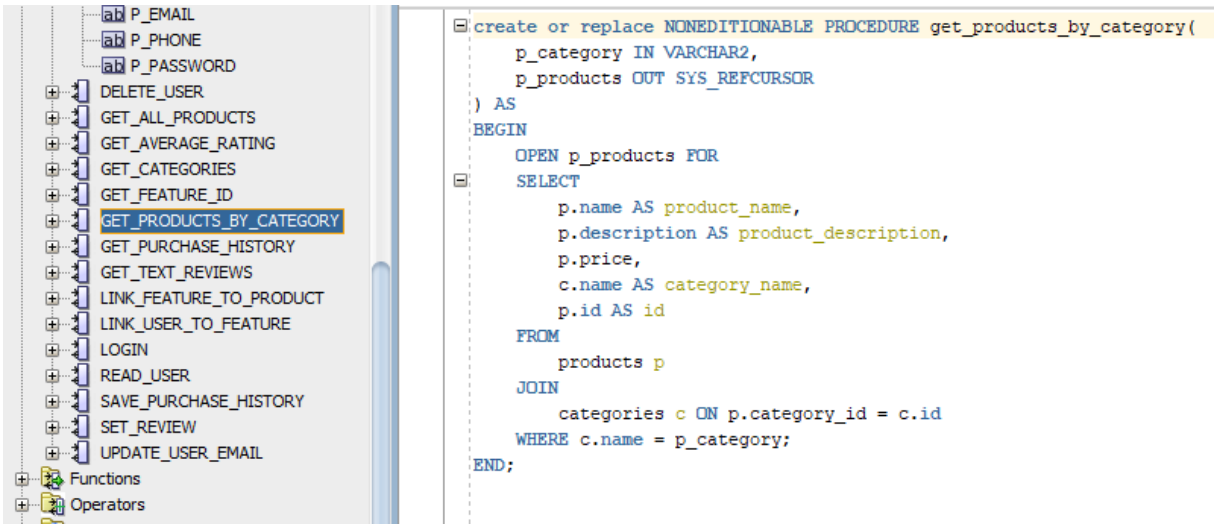

Figure 2 Procedura zwracająca produkty po kategorii

Klasa DBoperations w Pythonie

Klasa DBoperations w Pythonie jest odpowiedzialna za interakcję z bazą danych (głównie wywoływanie procedur). Poniżej znajduje się przykład implementacji metody create_user w tej klasie:

```python
def create_user(self, username, email, phone, password):
    try:
        self.cursor.callproc("create_user", [username, email, phone, password])
        self.connection.commit()
        print(f"User created successfully.")
    except cx_Oracle.IntegrityError as e:
        print("Failed to create user.")
        print(f"Error: {e}")
        self.connection.rollback()
```

Figure 3 Funkcja create_user w DBoperations

Funkcjonalności GUI

Interfejs użytkownika został zaprojektowany z użyciem biblioteki Tkinter. Umożliwia on użytkownikom interakcję z systemem poprzez przyjazny interfejs graficzny. Poniżej znajdują się przykłady okien w aplikacji:

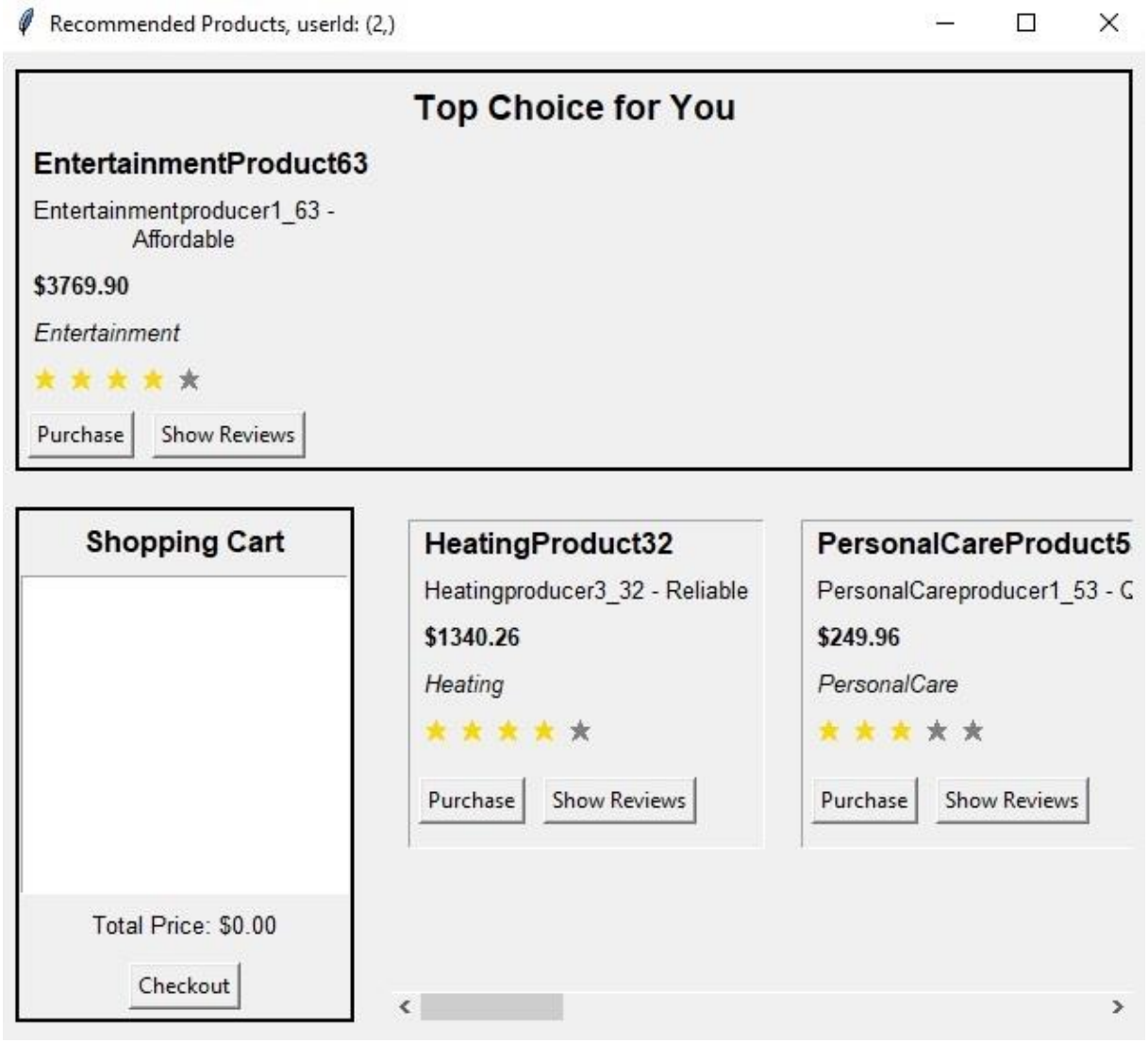

Figure 4 Okno rekomendacji

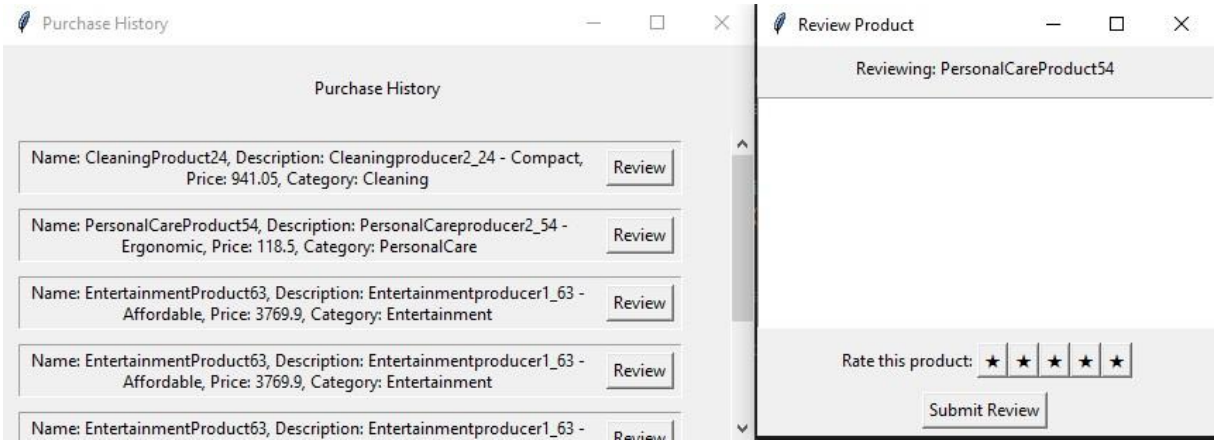

Figure 5 Okno histori zakupów i okno recenzji

Generowanie Danych Testowych

Generowanie danych testowych jest kluczowym krokiem w procesie testowania i walidacji systemu. W projekcie zastosowano skrypt Pythona do generowania przykładowych danych produktów i recenzji, które mogą być załadowane do bazy danych w celu symulacji rzeczywistych danych użytkowników i produktów. Skrypt generuje zapytania SQL do wstawiania produktów i recenzji na podstawie określonych parametrów, takich jak kategorie, firmy, przedziały cenowe i cechy produktów. Przykładowe zapytania są zapisywane do plików SQL, które można następnie zaimportować do bazy danych.

Algorytm K-means i Przetwarzanie Tekstu

Algorytm K-means jest jednym z algorytmów używanych do analizy tekstu w celu przypisania cech (features) do produktów i użytkowników. Proces ten obejmuje:

1. **Przetwarzanie Tekstu:**
   - Lematyzacja i usuwanie stop words (słów nieistotnych) za pomocą NLTK.
   - Tokenizacja tekstu i usunięcie znaków niealfabetycznych.
2. **Wektoryzacja TF-IDF:**
   - Obliczenie częstotliwości występowania terminów (TF) i częstotliwości dokumentów (DF).
   - Obliczenie macierzy TF-IDF, która reprezentuje teksty jako wektory cech.
3. **Klasteryzacja:**
   - Inicjalizacja centroidów na podstawie danych etykietowanych.
   - Iteracyjne przypisywanie tekstów do najbliższych centroidów i aktualizacja centroidów na podstawie średnich wektorów cech.

Obliczenie częstotliwości występowania terminów jest umożliwione przez zastosowanie słownika mapującego wiele przymiotników wskazujących do każdej cechy:

```
self.feature_keywords = {
    "Energy-Efficient": ["energy", "efficient", "power", "eco", "saving", "green", "consumption", "sustainable"],
    "Compact": ["compact", "small", "space", "sleek", "portable", "tiny", "minimal", "mini"],
    "Durable": ["durable", "sturdy", "robust", "long-lasting", "strong", "tough", "resilient", "hard-wearing"],
    "Innovative": ["innovative", "creative", "novel", "unique", "cutting-edge", "advanced", "original", "breakthrough"],
    "Ergonomic": ["ergonomic", "comfortable", "user-friendly", "easy-to-use", "handy", "intuitive", "well-designed"],
    "Quiet": ["quiet", "silent", "noise-free", "hushed", "low-noise", "soundless", "peaceful", "whisper-quiet"],
    "Smart": ["smart", "intelligent", "automated", "connected", "high-tech", "AI", "advanced", "clever"],
    "Stylish": ["stylish", "fashionable", "elegant", "chic", "sleek", "trendy", "modern", "aesthetic"],
    "Versatile": ["versatile", "multi-purpose", "adaptable", "flexible", "all-around", "universal", "multi-functional"],
    "High-Capacity": ["high-capacity", "large", "roomy", "spacious", "big", "substantial", "ample", "vast"],
    "Fast": ["fast", "quick", "speedy", "rapid", "swift", "high-speed", "efficient", "prompt"],
    "Safe": ["safe", "secure", "risk-free", "protected", "harmless", "trustworthy", "reliable", "riskless"],
    "User-Friendly": ["user-friendly", "easy-to-use", "accessible", "simple", "intuitive", "convenient", "straightforward"],
    "Affordable": ["affordable", "cheap", "inexpensive", "budget", "cost-effective", "economical", "low-cost", "reasonable"],
    "Reliable": ["reliable", "dependable", "consistent", "trustworthy", "steady", "unfailing", "proven", "stable"]
}
```

Figure 6 Słownik dla cech

Wykorzystane Biblioteki i Oprogramowanie

Projekt wykorzystuje szereg bibliotek i narzędzi programistycznych, które wspierają implementację i funkcjonowanie systemu:

1. **Python**: Język programowania używany do tworzenia backendu systemu, algorytmów rekomendacji oraz skryptów generujących dane testowe.
2. **Tkinter**: Biblioteka GUI dla Pythona, używana do tworzenia interfejsu użytkownika.
3. **cx_Oracle**: Biblioteka Pythona do łączenia się z bazą danych Oracle i wykonywania zapytań SQL.
4. **NLTK (Natural Language Toolkit)**: Biblioteka do przetwarzania języka naturalnego, używana do lematyzacji, tokenizacji i usuwania stop words.
5. **NumPy**: Biblioteka do obliczeń numerycznych, używana w implementacji algorytmu K-means.
6. **Oracle Database**: System zarządzania bazą danych, używany do przechowywania danych użytkowników, produktów, recenzji i historii zakupów.
7. **SQL Developer**: Narzędzie do zarządzania bazą danych Oracle, używane do tworzenia procedur składowanych i zarządzania strukturą bazy danych.