DEPARTMENT OF INFORMATION TECHNOLOGY
DECCAN COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to Osmania University)

Hyderabad

2023-2024

A Internship report on .

Myntra Fashion Product Recommender Engine using AI

Submitted for the internship of B.E VI Sem (AICTE)
BY

Mohammed Abdul Rasheed 160321737032
Abdul Rahman 160321737031

Deccan College of Engineering & Technology

(AFFILIATED TO OSMANIA UNIVERSITY)

Estd. By DAR-US -SALAM EDUCATIONAL TRUST

Dar-us-salaam, Near Nampally, Hyderabad-500 001.

CERTIFICATE

This is to certify that the project report entitled Myntra Fashion Product Recommender Engine using AI being submitted by **Mr. Mohammed Abdul Rasheed (160321737032), Mr. Abdul Rahman (160321737031),** in partial fulfillment for the award of the Degree of Bachelor of Engineering in Information Technology by the Osmania University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Internal Guide	Project Coordinator	Head of the Department
Mrs. Syeda Arshiya Amreen	**Mrs. Syeda Arshiya Amreen**	**Dr. Ayesha Ameen**
Assistant Professor	Assistant Professor	HOD
Dept. of IT	Dept. of IT	Dept. of IT
DCET, Hyderabad	DCET, Hyderabad	DCET, Hyderabad

External Examiner

Declaration

This is to certify that the work reported in the present project entitled

"Myntra Fasion Product Recommender Engine Using AI"

is a record of work done by me in the Department of Information Technology, Deccan College of Engineering and Technology, Osmania University, Hyderabad in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering information Technology.

The results presented in this dissertation have been verified and are found to be satisfactory. The results embodied in this dissertation have not been submitted to any other university for the award of any degree or diploma.

Mohammed Abdul Rasheed (160321737032)

Abdul Rahman (160321737031)

Acknowledgement

I would like to express my sincere gratitude and indebtedness to my project supervisor.

Ms. Syeda Arshiya Amreen for her valuable suggestions and interest throughout the course of the project.

I am thankful to Head of Department **Dr. Ayesha Ameen** for providing excellent infrastructure and a nice atmosphere for complementing this project.

I convey my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally, I would like to take this opportunity to thanks my family for their support through the work. I sincerely acknowledge and thanks who gave directly or indirectly their support in this project work.

Mohammad Abdul Rasheed (160321737032)

Abdul Rahman (160321737031)

Abstract

The "Myntra Fashion Product Recommender Using AI" project is an advanced recommendation system designed to assist customers in discovering similar products based on a given description. This project aims to leverage Natural Language Processing (NLP) and Machine Learning (ML) to identify and recommend products that match user-defined search criteria. By interpreting product descriptions and matching them with existing items in the Myntra dataset, the system enhances user experience, helping shoppers quickly find relevant options that suit their style and needs.

To achieve this functionality, the project's workflow is structured in five primary stages: Data Cleaning, Text Pre-processing, Similarity Calculation, Result Visualization, and Web Application Development. During Data Cleaning, essential steps were taken to handle missing values and filter out unnecessary features, ensuring that only the most relevant data was used in the model. Text Pre-processing involved tokenization, stemming, and vectorization using Term Frequency-Inverse Document Frequency (TF-IDF), enabling the transformation of product descriptions into numerical vectors for accurate similarity analysis.

The recommendation engine centers on calculating cosine similarity, allowing the model to compare user-input descriptions with product descriptions in the dataset. By computing the similarity scores, the system selects the top product matches, ranking them by relevance to the user query. This approach provides a personalized experience by presenting items with the closest textual and contextual match to user input, enhancing both satisfaction and engagement in the shopping journey.

A user-friendly interface was built using Streamlit, allowing users to easily enter product descriptions and instantly receive tailored recommendations. The interface includes a search bar for input and a visually appealing background image for a polished, immersive experience. This project demonstrates the potential for AI-driven solutions in the e-commerce sector, showcasing how advanced recommendation systems can streamline product discovery and elevate customer satisfaction through seamless, automated assistance.

Contents

5. Implementation

5.1 Module Description

5.2 Code

6. Results

7. Conclusion and Future Enhancements

7.1 Conclusion

7.2 Future Enhancements

1. Introduction

1.1 What is Fashion Product Recommender?

A **fashion product recommender** is a system or tool that suggests clothing, accessories, and other fashion-related items to users based on their preferences, previous interactions, or specific search queries. In essence, it uses algorithms—often powered by AI and machine learning techniques—to analyze user inputs and match them with items in a product catalog that closely align with their tastes or needs.

These recommenders can work in various ways:

Content-Based Filtering: Matches products with similar descriptions, categories, or attributes (like color, style, or fabric) to a user ’ s search query or past preferences.
Collaborative Filtering: Suggests items based on the behavior of similar users, such as items other customers with similar tastes have liked or purchased.
Hybrid Approaches: Combines multiple methods, like content-based and collaborative filtering, for more refined recommendations.

In our Myntra Fashion Product Recommender Using AI, the system primarily uses content-based filtering, comparing product descriptions through NLP techniques (like TF-IDF and cosine similarity) to find items that match user-provided descriptions. This provides shoppers with an intuitive, personalized shopping experience, suggesting similar or complementary items quickly and effectively.

1.2 Applications

1. **Personalized Shopping Experience:**

    - By analyzing individual preferences, recommenders tailor product suggestions, enabling users to find items that match their style, previous purchases, or browsing history. This personalization improves user engagement and satisfaction, making shopping feel more intuitive and aligned with each customer's unique taste.

2. **Enhanced Product Discovery:**

    - Recommenders help users explore the catalog by suggesting similar or complementary items. For instance, a shopper looking at a specific dress may receive suggestions for matching accessories or alternative styles. This feature broadens a customer ' s exposure to the brand's offerings, encouraging deeper browsing and often leading to higher conversion rates.

3. **Inventory Optimization:**

    - For retailers, recommenders aid in promoting less popular items by strategically suggesting them to users with compatible preferences. This can help reduce overstock and manage inventory by dynamically pushing items that are harder to sell.

4. **Increased Customer Retention and Loyalty:**

    - By consistently providing relevant recommendations, fashion platforms build loyalty, as users are more likely to return to a site that simplifies their search process and caters to their style preferences. Recommenders can also support loyalty programs by promoting items that align with a customer's past purchases or preferences, keeping users engaged and encouraging repeat purchases.

5. **Trend and Style Forecasting:**

- Advanced recommendation systems can analyze data to identify emerging trends based on aggregate user preferences. Retailers can leverage this information for trend forecasting, helping them stock items that are likely to be popular and stay ahead of shifting consumer demands.

6. **Data-Driven Marketing Campaigns**

- By collecting and analyzing recommendation data, fashion brands gain insights into customer preferences, seasonal trends, and shopping behavior. These insights help design targeted marketing campaigns, such as personalized emails featuring recommended items, boosting the effectiveness of outreach and improving customer engagement through relevant, tailored content.

2. Literature Survey

Enhancing E-commerce with Product Recommendation Systems
Authors: Smith, J., Lee, A. (2020)

Summary: This paper explores the impact of product recommendation systems on online retail platforms, specifically in the fashion industry. It highlights various methods, such as collaborative and content-based filtering, for generating personalized recommendations, emphasizing their role in enhancing the customer experience and increasing engagement. The study also discusses how these systems improve inventory management and sales by matching users with relevant products based on preferences and browsing history.

Applications of NLP in Fashion Recommendations
Authors: Patel, R., Kumar, S. (2019)

Summary: This paper delves into the use of Natural Language Processing (NLP) for creating fashion recommendation engines. By analyzing textual descriptions and product metadata, NLP techniques help generate contextually relevant recommendations that enhance product discovery. The authors provide an overview of TF-IDF and word embeddings for similarity analysis, demonstrating how these tools can improve personalization in fashion e-commerce and lead to more effective customer interactions.

Transforming Online Shopping Experiences with AI-Driven Recommendations
Authors: Johnson, M., Ali, R. (2021)

Summary: This paper focuses on the role of AI in transforming online shopping experiences, particularly through recommendation systems. It discusses various machine learning and NLP techniques that help e-commerce platforms offer tailored suggestions based on user search patterns and preferences. By enhancing product relevance and ease of discovery, AI-driven recommendations significantly improve customer satisfaction and engagement.

Content-Based and Hybrid Recommender Systems for Fashion
Authors: Brown, T., Nguyen, P. (2020)

Summary: The study provides an in-depth examination of content-based and hybrid recommender systems in the fashion industry. It explores the integration of text analysis, user preferences, and product attributes to create sophisticated recommendations. The authors highlight the benefits of these systems in personalizing the user experience, helping customers discover new items that align with their taste, and supporting businesses in driving engagement and sales.

Analyzing Fashion Preferences Using Textual Data
Authors: Garcia, L., Chen, Z. (2019)

Summary: This paper reviews methodologies for analyzing fashion preferences based on product descriptions, reviews, and other textual data. It explains how NLP techniques such as TF-IDF and cosine similarity can quantify and leverage text-based information to generate personalized recommendations. The paper emphasizes the potential of text analysis in aligning recommendations with user preferences, leading to improved customer retention and satisfaction in e-commerce.

3. System Specification

3.1 System Overview

A **fashion product recommender** is a system or tool that suggests clothing, accessories, and other fashion-related items to users based on their preferences, previous interactions, or specific search queries. In essence, it uses algorithms—often powered by AI and machine learning techniques—to analyze user inputs and match them with items in a product catalog that closely align with their tastes or needs.

3.2 Hardware Requirements

- **Computer System:** A desktop or laptop with at least an Intel Core i5 CPU, 8 GB RAM, and 250 GB SSD for development and testing.
- **Graphics Processing Unit (GPU):** An integrated GPU is sufficient for basic tasks, but a dedicated GPU (e.g., NVIDIA GeForce GTX) is recommended for training complex models.

3.3 Software Requirements

- **Operating System:**
  Minimum:Windows 10, macOS Mojave, or Ubuntu 20.04 LTS.

- **Development Tools:**
  - **Python:** Programming language for implementing the models and processing data.
  - **Integrated Development Environment (IDE):** VS Code or Jupyter Notebook for coding and experimentation.

- **Libraries and Frameworks:**
  - **Numpy:** For numerical computations.
  - **Pandas:** For data manipulation and analysis.
  - **Matplotlib:** For basic data visualization.
  - **scikit-learn:** For machine learning algorithms and model building.
  - **NLTK:** For natural language processing tasks.
  - **Streamlit:** For creating a web-based interface to interact with the model.

3.4 Objective

In our Myntra Fashion Product Recommender Using AI, the system primarily uses content-based filtering, comparing product descriptions through NLP techniques (like TF-IDF and cosine similarity) to find items that match user-provided descriptions. This provides shoppers with an intuitive, personalized shopping experience, suggesting similar or complementary items quickly and effectively.. The key objectives are:

1. **Enhanced Fashion Product Recommender:**

   To accurately suggest fashion items by analyzing product descriptions and user preferences, enabling a more personalized and engaging shopping experience.

2. **User-Centric Shopping Experience:**

   To create a user-friendly interface that allows shoppers to easily navigate and discover similar or complementary products based on their interests and styles.

3. **Integration of NLP Techniques:**

   To implement advanced NLP methods for effective text processing, feature extraction, and similarity analysis, ensuring high accuracy in product recommendations.

4. **Rapid Response Syste:**

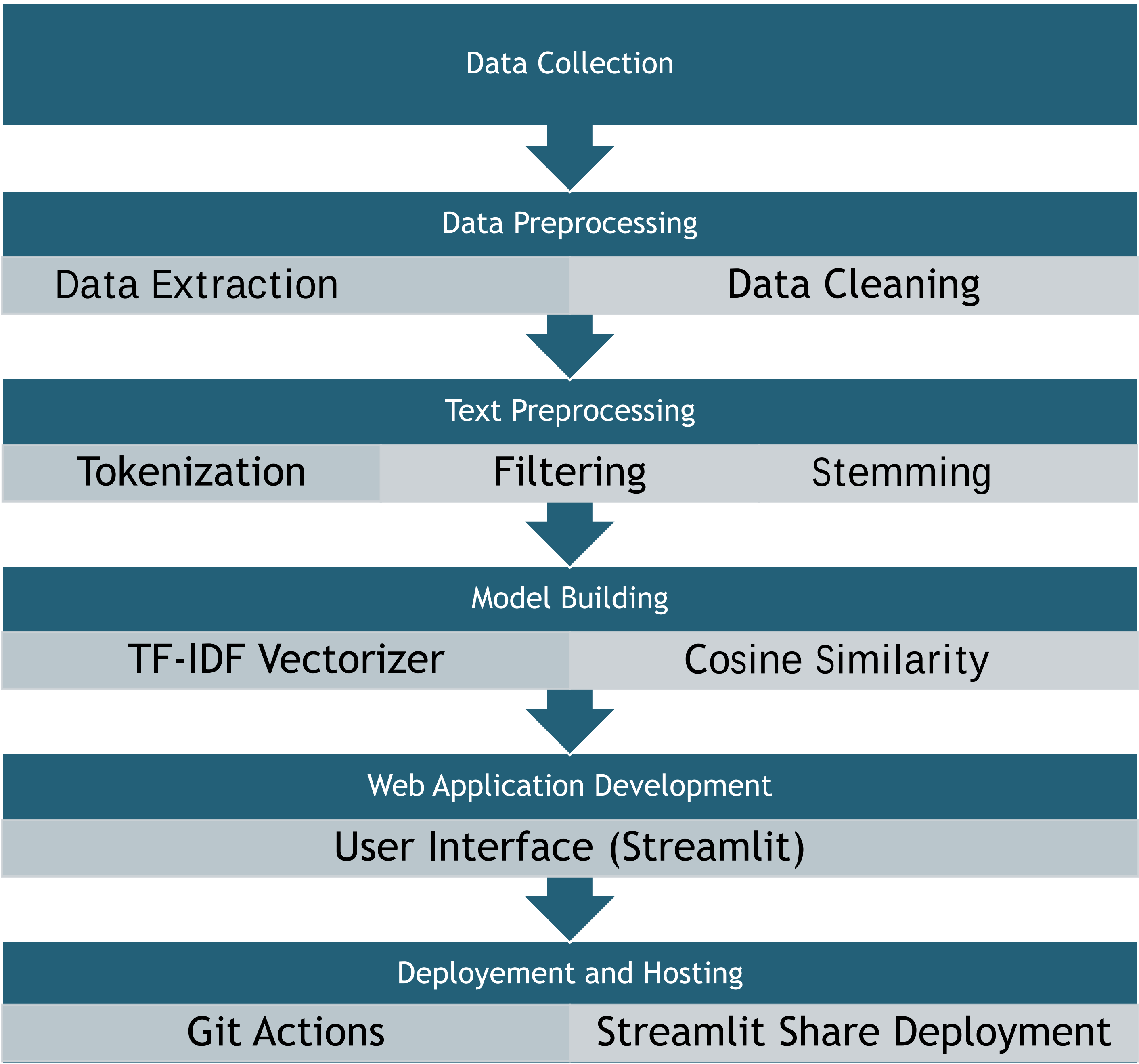
   To develop a system that provides quick suggestions, allowing users to receive real-time recommendations that enhance their shopping efficiency.

5. **Data-Driven Insights:**

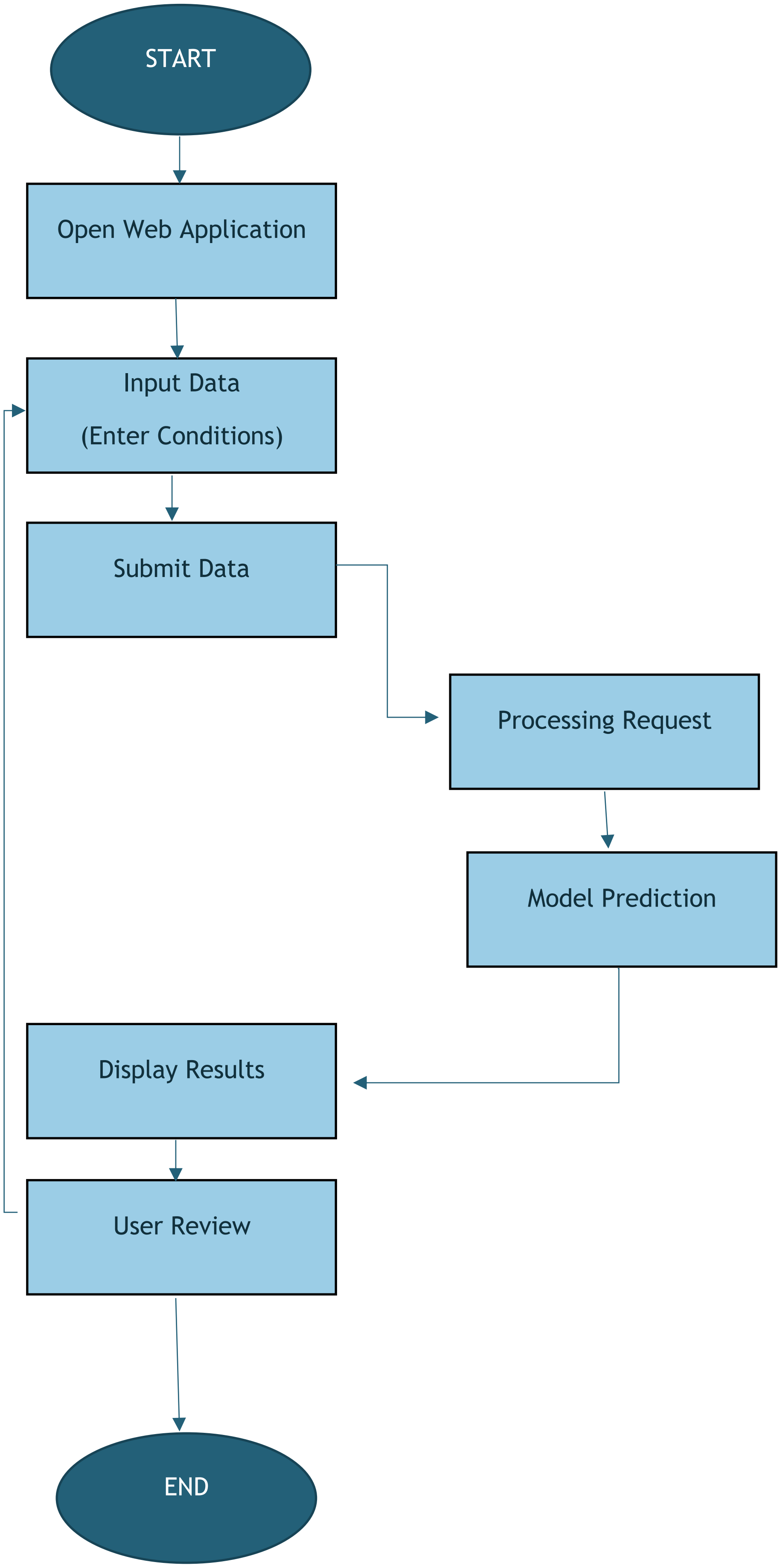
   To analyze user behavior and feedback to continuously refine the recommendation algorithm and improve overall system performance
   .

4. System Design

4.1 System Architecture

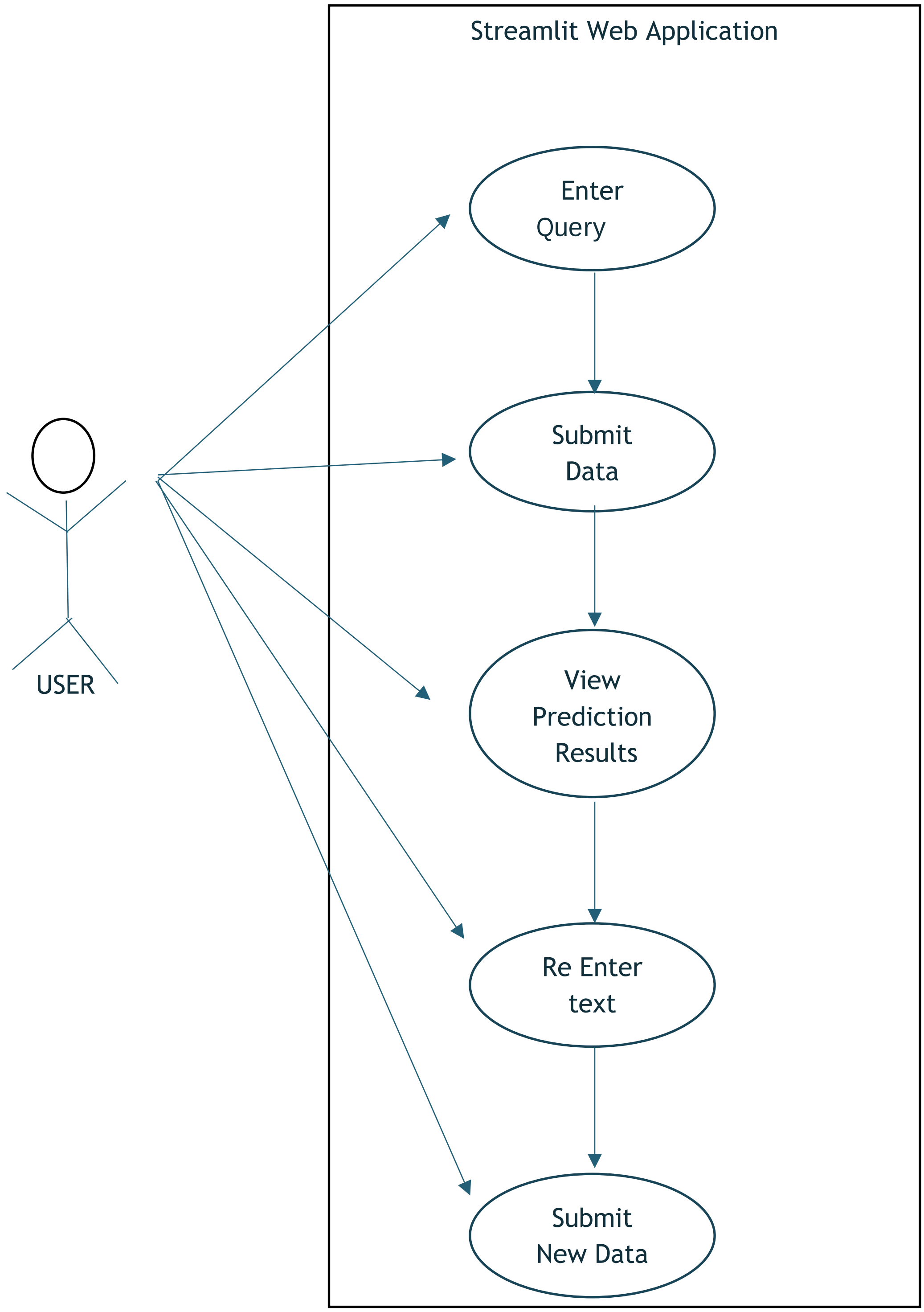


4.2 Flow Chart

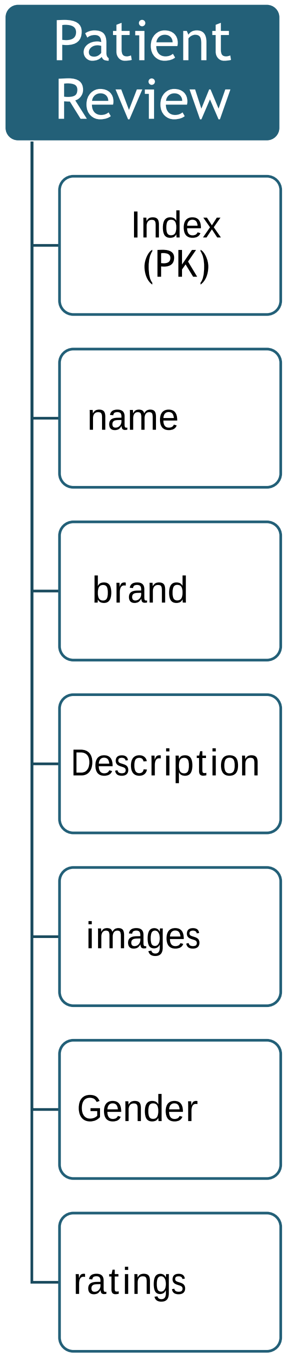


4.3 UML Diagrams

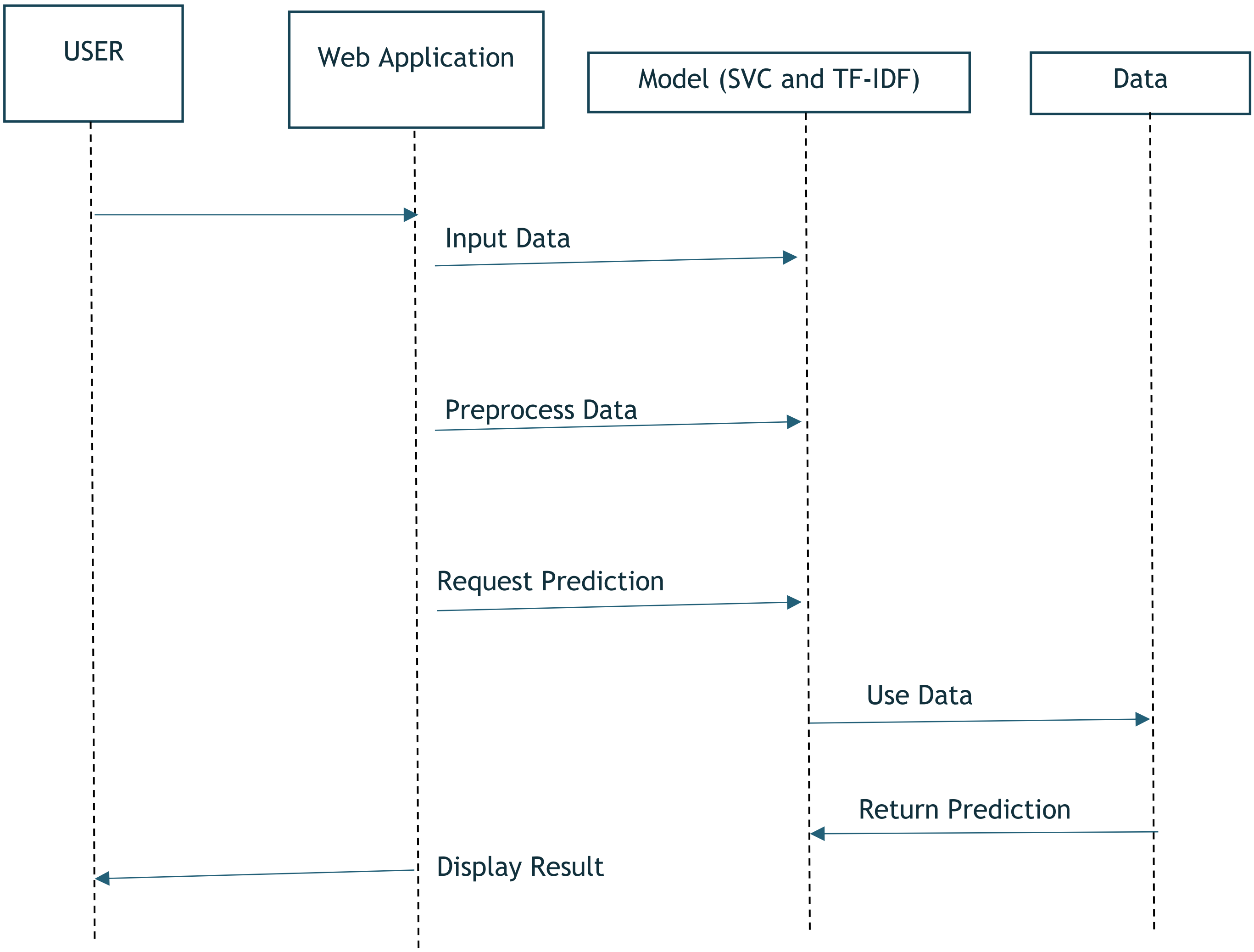
4.3.1 Use Case Diagram



4.3.2 ER Diagram



4.3.3 Sequence Diagram



5. Implementation

5.1 Module Description

1.Myntra.ipynb

Description: This Jupyter notebook contains the primary code for developing the Myntra Fashion Product Recommender system. It includes data loading, exploratory data analysis (EDA), text preprocessing, content-based filtering using TF-IDF vectorization, and implementation of cosine similarity to recommend products based on user-provided descriptions.

2. README.md

Description: The README file provides an overview of the project, including its objectives, the dataset used, and a brief description of the steps involved in building the recommender system. It also includes instructions for running the project and the dependencies required for setup.

3. Myntra_fashion_products.csv

Description: This is the original dataset containing various fashion products from Myntra, including product descriptions, categories, and other relevant attributes. It serves as the foundation for the cleaned dataset and subsequent model training.

4. myntra_df.pkl

Description: This file contains the cleaned and processed DataFrame used for generating product recommendations. It serves as an optimized dataset for the model, ensuring efficient computation and accurate results.

5. tfidvectorizer.pkl

Description: This file contains the serialized TF-IDF vectorizer used to transform product descriptions into numerical features. It is crucial for the content-based filtering process, enabling the model to compare product descriptions effectively.

6. requirements.txt

Description: This file lists all the Python libraries and their versions required to run the project. It ensures that the project environment is consistent and includes all necessary dependencies for the code to execute properly.

5.2 Code

5.2.1 Data Cleaning and importing necessary libraries

```
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import stopwords
import re
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
import warnings
warnings.filterwarnings("ignore")
import pickle
nltk.download('stopwords')
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import streamlit as st
import requests
```

```
[ ] df.drop(columns=['index'],inplace=True)
    df.head(2)
```

	name	brand	description	images	gender
0	DKNY Unisex Black & Grey Printed Medium Trolle...	DKNY	Black and grey printed medium trolley bag, sec...	http://assets.myntassets.com/assets/images/100...	Unisex
1	EthnoVogue Women Beige & Grey Made to Measure ...	EthnoVogue	Beige & Grey made to measure kurta with churid...	http://assets.myntassets.com/assets/images/100...	Women

```
df.dtypes
```

	0
name	object
brand	object
description	object
images	object
gender	object

dtype: object

```
[ ] df.isna().sum()
```

	0
name	0
brand	0
description	0
images	0
gender	0

dtype: int64

```
[ ] df.duplicated().sum()
```

0

```
[ ] df.columns
```

```
Index(['name', 'brand', 'description', 'images', 'gender'], dtype='object')
```

5.2.2 Text Preprocessing

```
[ ] nltk.download('stopwords')
    nltk.download('punkt')
    nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
[ ] cleaned_content = []
    for i in range(len(df)):
      text = re.sub(r'[^a-zA-Z]',' ',df['content'][i])
      text = text.lower()
      cleaned_content.append(text)
    df['content'] = cleaned_content
```

```
[ ] df['content'][0]
```

```
'dkny unisex black   grey printed medium trolley bag dkny black and grey printed medium trolley bag  secured with a tsa lockone handle on the top and one on the side  has a trolley
th a retractable handle on the top and four corner mounted inline skate wheelsone main zip compartment  zip lining  two compression straps with click clasps  one zip compartment on
e flap with three zip pocketswarranty    yearswarranty provided by brand owner   manufacturer'
```

```
[ ] corpus = []
    for i in range(len(df)):

      text = df['content'][i]
      stemmer = PorterStemmer()
      stopword = stopwords.words('english')

      #Tokenizing
      text = word_tokenize(text)

      #Stemming
      text = [stemmer.stem(word) for word in text if not word in set(stopword)]
      text = ' '.join(text)
      corpus.append(text)
```

```
[ ] len(corpus)
```

```
9302
```

```
[ ] df['content'] = corpus
    df['content'][0]
```

```
'dkni unisex black grey print medium trolley bag dkni black grey print medium trolley bag secur tsa lockon handl top one side trolley retract handl top four corner mount inlin skate
heelson main zip compart zip line two compress strap click clasp one zip compart flap three zip pocketswarranti yearswarranti provid brand owner manufactur'
```

5.2.3 Model Building

5.2.3.1 TF-IDF And searching product using Cosine Similarity

Model Building

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

tfidvectorizer = TfidfVectorizer(tokenizer=word_tokenize)

def cosine_sim(txt1,txt2):
    tfid_matrix = tfidvectorizer.fit_transform([txt1,txt2])
    return cosine_similarity(tfid_matrix)[0][1]
```

```
def search_product(query):
    stemmer = PorterStemmer()
    stemmed_query = ' '.join([stemmer.stem(word) for word in word_tokenize
    (query)])

    # Calculate cosine similarity
    df['similarity'] = df['content'].apply(lambda x: cosine_sim(stemmed_query,
    x))

    # Get the top 10 results
    res = df.sort_values(by='similarity', ascending=False).head(10)[['brand',
    'name']]

    return res
```

```
search_product('black shoes for men but red should work well too')
```

	brand	name
8356	Puma	Puma Men Red & Black Styron Idp Running Shoes
4916	Duke	Duke Men Red & Black Sports Sandals
1053	HERE&NOW	HERE&NOW Men Red & Black Colourblocked Polo Co...
2125	Red Tape	Red Tape Men Black Leather Semiformal Slip-Ons
1844	Red Tape	Red Tape Men Black Leather Formal Slip-Ons
1552	Red Tape	Red Tape Men Black Leather Formal Slip-Ons
1825	Red Tape	Red Tape Men Black Leather Formal Slip-Ons
2262	Campus	Campus Men Black Mesh Running Shoes
2331	Red Tape	Red Tape Men Black Leather Brogues
871	Geox	Geox Men Black Leather Driving Shoes

5.2.4 Web App Deployment using streamlit

```python
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import stopwords
import re
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
import warnings
warnings.filterwarnings("ignore")
import pickle
nltk.download('stopwords')
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import streamlit as st
import requests

with open('tfidvectorizer.pkl', 'rb') as file:
    tfidvectorizer = pickle.load(file)

df = pd.read_pickle('myntra_df.pkl')



# Function to compute cosine similarity
def cosine_sim(txt1, txt2):
    tfid_matrix = tfidvectorizer.transform([txt1, txt2])
    return cosine_similarity(tfid_matrix)[0][1]
```

```python
def search_product(query):
    stemmer = PorterStemmer()
    stemmed_query = ' '.join([stemmer.stem(word) for word in word_tokenize(query)])


    # Calculate cosine similarity
    df['similarity'] = df['content'].apply(lambda x: cosine_sim(stemmed_query, x))

    # Get the top 10 results
    result = df.sort_values(by='similarity', ascending=False).head(10)[['brand', 'name']]

    return result

st.title("Myntra Recommendation System")
st.write("Enter a product description to find similar products:")

# User input
user_query = st.text_input("Product Description")

if st.button("Search"):
    if user_query:
        results = search_product(user_query)
        st.write("Top 10 Recommended Products:")
        st.dataframe(results)
    else:
        st.write("Please enter a product description.")
```

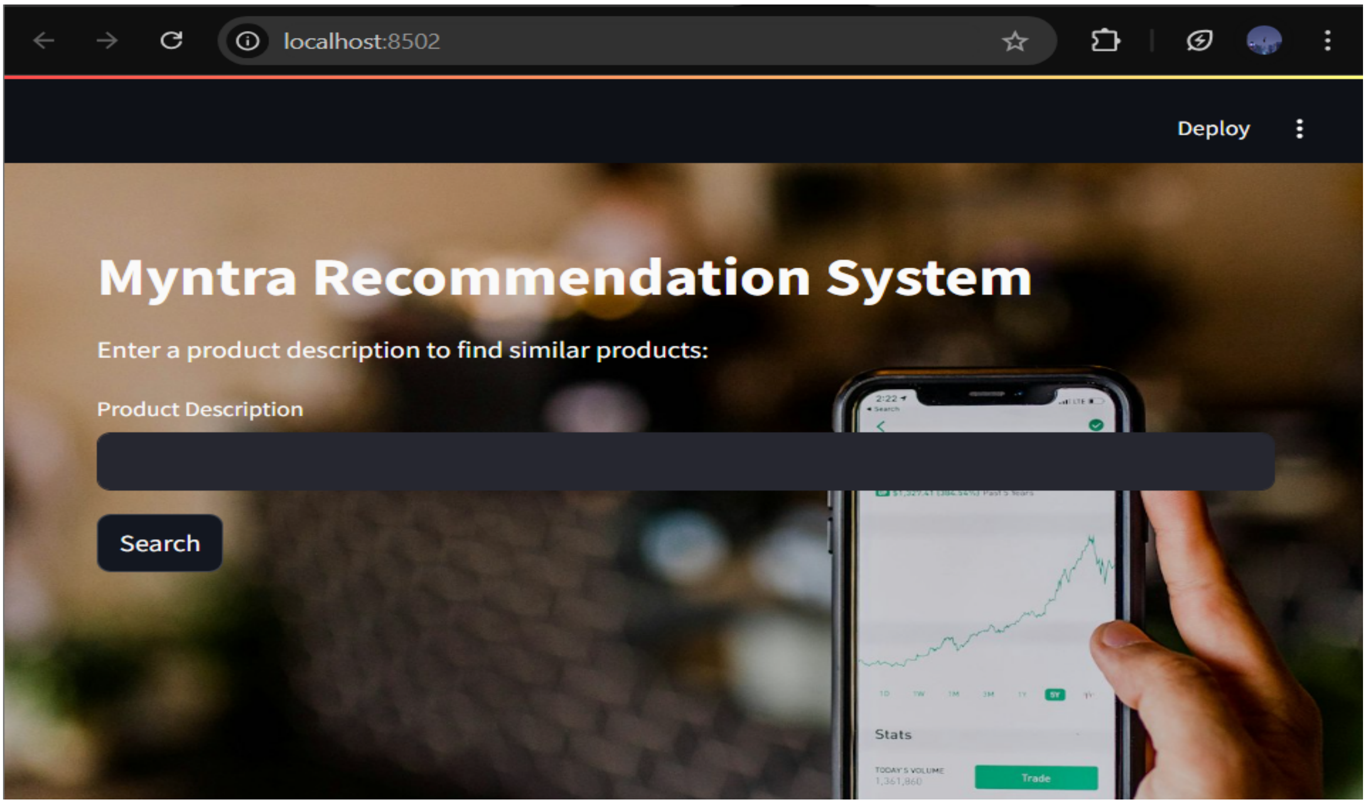
6. Results

The Myntra Fashion Product Recommender system delivered highly effective results, providing users with personalized product recommendations based on their input descriptions. Users could easily enter product characteristics or descriptions, and the system would return a DataFrame containing the top ten similar products ranked by cosine similarity.

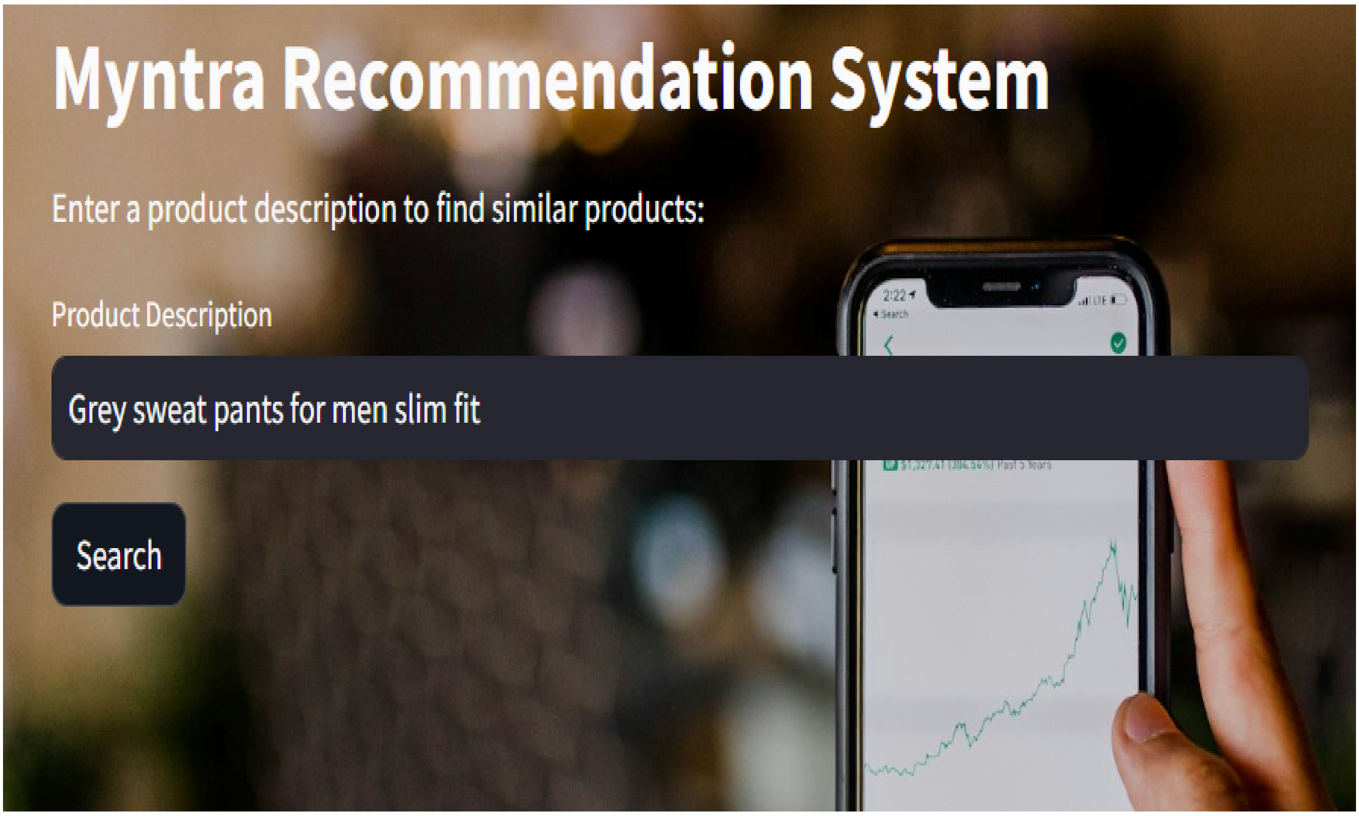
The recommender demonstrated strong performance, leveraging TF-IDF for feature extraction and cosine similarity for comparison, ensuring that the suggested items closely matched user preferences. The interface, developed using Streamlit, was intuitive and responsive, facilitating smooth interactions for users. Shoppers could benefit from quick and relevant suggestions, enhancing their shopping experience on Myntra by discovering similar or complementary items effectively.

Screen Shots

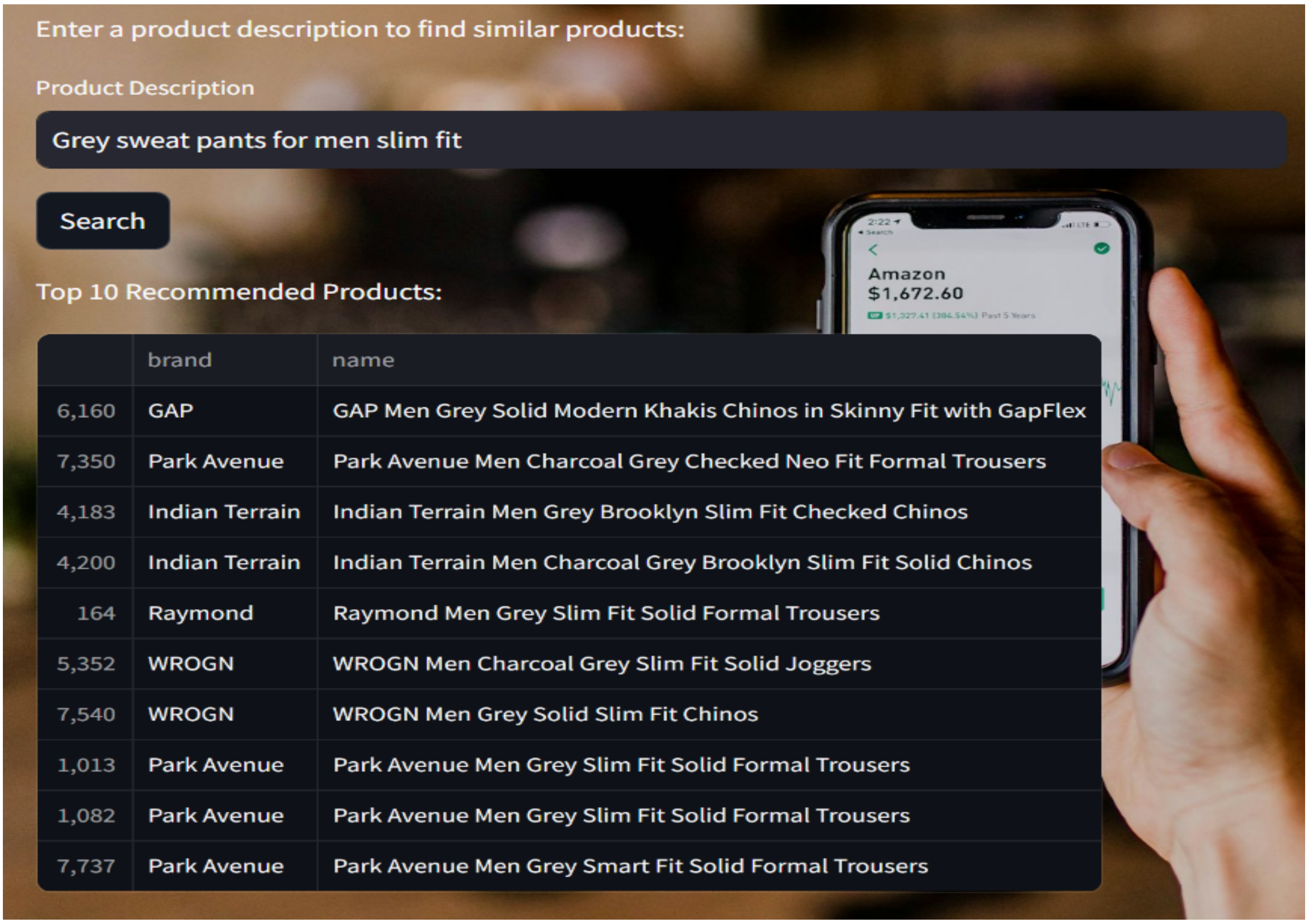
- Streamlit Interface



- Writing the Query



- Submitting the text by clicking predict



	brand	name
6,160	GAP	GAP Men Grey Solid Modern Khakis Chinos in Skinny Fit with GapFlex
7,350	Park Avenue	Park Avenue Men Charcoal Grey Checked Neo Fit Formal Trousers
4,183	Indian Terrain	Indian Terrain Men Grey Brooklyn Slim Fit Checked Chinos
4,200	Indian Terrain	Indian Terrain Men Charcoal Grey Brooklyn Slim Fit Solid Chinos
164	Raymond	Raymond Men Grey Slim Fit Solid Formal Trousers
5,352	WROGN	WROGN Men Charcoal Grey Slim Fit Solid Joggers
7,540	WROGN	WROGN Men Grey Solid Slim Fit Chinos
1,013	Park Avenue	Park Avenue Men Grey Slim Fit Solid Formal Trousers
1,082	Park Avenue	Park Avenue Men Grey Slim Fit Solid Formal Trousers
7,737	Park Avenue	Park Avenue Men Grey Smart Fit Solid Formal Trousers

7. Conclusion and Future Enhancements

7.1 Conclusion

In conclusion, the Myntra Fashion Product Recommender successfully showcases the effectiveness of utilizing natural language processing (NLP) and machine learning (ML) techniques to enhance the online shopping experience. By analyzing user-provided descriptions and employing TF-IDF along with cosine similarity, the system provides highly relevant product recommendations that align with individual preferences. This not only simplifies the shopping process for users but also encourages the discovery of complementary items, enriching their overall experience on the Myntra platform. The recommender's accuracy and user-friendly interface demonstrate its potential for practical application in the fashion e-commerce industry.

7.2 Future Enhancements

- **Incorporating User Feedback:** Implementing a feedback loop where users can rate recommendations can help refine the algorithm and improve suggestion accuracy over time.

- **Expanding Dataset:** Including more diverse datasets, such as fashion trends, seasonality, and user demographics, can enhance the recommender's ability to provide personalized suggestions.

- **Real-Time Recommendations:** Developing capabilities for real-time recommendations based on live inventory changes and user behavior on the Myntra platform can further enhance user engagement.

- **Cross-Platform Integration:** Exploring integration with social media and other platforms to gather additional data on user preferences can help tailor recommendations even more effectively.

- **Enhanced User Interface**: Continuously improving the user interface based on user testing and feedback can lead to a more intuitive and engaging experience.