AI-Powered Fashion Trend & Recommendation System

Summary Report

Overview

This project presents an **AI-powered solution** to forecast and visualize fashion search trends and recommend relevant products. By leveraging search queries, sales data, customer feedback, and product catalogs, the system extracts meaningful keywords, analyzes sentiment, forecasts demand trends using **Prophet**, and displays the results through an **interactive dashboard** built with **Dash**.

Approach & Assumptions

Problem Breakdown:

1. Trend Extraction

- Extract keywords from real-time search queries using NLP (tokenization, stopword removal).
- Track keyword frequency over time.

2. Relevance Scoring

- Combine:
    - Search frequency
    - Product sales count
    - Customer sentiment scores
- Normalize each and compute a weighted **composite score**.

3. Forecasting

- Apply **Facebook Prophet** to forecast each keyword's trend score for the next 7 days.
- Select **Top 20 keywords** based on projected scores.

4. Product Recommendation

- Match trending keywords to products via metadata (title, category, modifiers, keywords).
- Filter top matching products.

5. Dashboard

- Visualize:
  - Top 20 trending keywords (bar chart + table)
  - Matched products (interactive data table)
- Built using **Dash**, styled with Plotly and HTML components.

Assumptions:

- Search queries are clean and can be tokenized effectively.
- Customer sentiment is captured well via textual feedback.
- Product metadata is rich enough for keyword matching.
- Product IDs map consistently across datasets.

Tools, Libraries & Models Used

Category	**Tools / Libraries**
Language & IDE	Python, Google Colab
NLP & Sentiment	nltk, TextBlob
Forecasting	Prophet (Facebook/Meta)
Preprocessing	re, json, MinMaxScaler
Data Handling	pandas, numpy
Visualization	plotly, dash, dash_table
Storage / IO	Google Drive (CSV and JSON input/output)

Run Instructions

Dataset Structure:

All datasets should be stored in the following Google Drive path:

/MyDrive/HackFest_Dataset/

- search_trends.csv

- sales_data.csv
- customer_feedback.csv
- product_catalog.csv

Step-by-Step Execution

1: Mount Google Drive

```
{
        from google.colab import drive

        drive.mount('/content/drive')

}
```

- Enables reading/writing data from /content/drive/MyDrive/HackFest_Dataset/.

2: Import Required Libraries

```
{
        import pandas as pd

        import numpy as np

        import re

        import json

        import nltk

        from nltk.corpus import stopwords

        from textblob import TextBlob

        from sklearn.preprocessing import MinMaxScaler

        from prophet import Prophet

        from nltk.tokenize import TreebankWordTokenizer


        import pandas as pd

        import json
```

import re

}

- Imports tools for NLP, forecasting, data handling, sentiment analysis, and visualization.

3: Generate_trending_keywords() Function

Processes and forecasts keyword trends.

Steps:

- Load all datasets.
- Clean and tokenize search queries.
- Aggregate keyword frequency per date.
- Map products by productId and categorize.
- Compute average sentiment scores from feedback using TextBlob.
- Normalize features (search, sales, sentiment).
- Compute **composite score**:
    - ➢ score = 0.5 * normalized_search + 0.3 * normalized_sales + 0.2 * normalized_sentiment
- Forecast each keyword's score using **Prophet**.
- Select top 20 keywords for the next 7 days.
- Save results to: toptrendingkeywords.json.

4: generate_recommendations() Function

Maps trending keywords to matching products.

Steps:

- Load:
    - toptrendingkeywords.json
    - product_catalog.csv
- Merge metadata (title, category, modifiers, keywords) into a searchable text field.

- Match products using regular expression search.
- Save output to: recommended_products.csv.

5: Run Main Functions

```
{
    generate_trending_keywords()
    generate_recommendations()
}
```

- Produces two outputs:
    - Trending keywords with forecasts
    - Matched product recommendations

6: Install Plotly & Dash

```
{
        !pip install dash plotly
}
```

- Required for the visualization dashboard.

7: Visualization via Dash App

Key Components:

```
{
        import dash
        from dash import html, dcc, dash_table
        import plotly.express as px
        import pandas as pd
        import json
}
```

Features:

- **Top 20 Keywords Tab**:
    - Bar chart of forecasted scores.
    - Table listing keywords and scores.
- **Product Recommendations Tab**:
    - Interactive product table with forecast keyword scores.
- **UI**: Simple, tabbed layout using Dash components.

Run:

```
{
        if __name__ == '__main__':
            main()  # Starts Dash server
}
```

Outputs

File	Description
toptrendingkeywords.json	Contains top 20 forecasted keywords with their scores.
recommended_products.csv	Contains products matched to keywords, along with scores.
Dash App	Visualization dashboard with interactive keyword and product insights.

Dashboard Preview (UI Tabs)

- **Trending Keywords**
    - Bar chart of keyword scores
    - Table view of keywords + scores
- **Products in High Demand**
    - Product listings per keyword
    - Filtered table with metadata + match score

Future Enhancements

- Use **BERT embeddings** for fuzzy keyword-product matching.
- Integrate **real-time data streams** for live trend updates.
- Improve the dashboard with **filters, search, and graphs over time**.
- Add **product recommendation engine** using collaborative filtering.