Final Report

Fashion Trend Prediction & Recommendation System Using Social-Media Insights

Student: Pratham Chauhan

Course: CS 439 — Intro to Data Science

Date: 12/09/2025

GitHub: - Final Project

Demo Video: - Video Demo

1.1 Introduction & Problem Statement

Fashion is a very fast-moving business and is impacted not just by the availability of products, but also by evolving cultural and social media driven trends. The typical recommendation systems tend to be focused on metadata about a product catalog and/or collaborative filtering which create a serious void in capturing real-time trend signals. A text-based Fashion Recommendation/Trend predictor system will use the product catalog descriptions and social media-like text to create a model. This project of mine will develop a text-based Fashion Recommendation/Trend prediction system using product catalog descriptions and other forms of social media-like text such as text messages and posts to create meaningful product recommendations using TF-IDF embeddings and cosine similarity. The project will also develop methods for extracting trend keywords and phrases from social media-type text using NLP. I will clean, preprocess, and perform exploratory analyses on the data to develop better understanding of the relationships between products and trends through the development of models and conducting feature engineering, experimental modeling, and using NF, I hope to demonstrate how unstructured textual signals can improve Fashion Analytics.

1.2 Connection to the course

This project will utilize key data science concepts that are addressed in the course, Intro to Data Science. This includes activities such as data cleaning, exploratory data analysis and SQL-like operations, Natural Language Processing (NLP), the calculation of TF-IDF, Unsupervised Learning, detecting trends, searching for similar products, and performing evaluations. The project will also apply skills gained from using tools such as Pandas, Scikit-Learn, and regular expressions and text normalization developed in the course through data cleaning and exploratory analysis. Additionally, various concepts associated with vectorizing text data and constructing sparse matrices along with using evaluation metrics, which aligns strongly with course material.

1.3 Problem definition and Background

The prevailing techniques in the fashion recommendation space are based on either collaborative filtering and user purchase history or product descriptions. However, fashion trends are very dynamic, and change based on popular culture. Most of the current trend-setting styles are widely spread on social networks by social media influencers, hashtags, and/or posts that go viral. On the other hand, existing product only based systems are not able to adapt to the rapidly changing nature of fashion.

So, with this project, I will try to develop a new fashion recommendation model that produces recommendations for trending keywords found in the 'socio-cultural' online social media communities and links them to existing fashion trend categories for easy use. Recommendations will be based solely on existing trends and therefore no individualized user purchase data will be used.

2. Novelty and importance

2.1 Importance of the Project

Forecasting fashion trends is critical from a supply chain, marketing, and sustainability standpoint. Accurate forecasting of style changes will allow retailers to better align their inventory with consumer preferences while decreasing over-production through efficient purchase planning processes. Also, for students, who are studying of data science, they have the opportunity to apply data science methodology to fashion as a case study for predictive modeling, natural language processing (NLP), clustering, text embeddings and vector similarity.

Through this project, Data Scientists will be able to show how data-science based methodologies can obtain meaning from uncategorized text documents and apply that data to a largely unoccupied area of application. Even with the current computational limitations, mine proposed project outline highlights how to use text data to develop recommendation systems through the creation of a text feature set.

2.2 Novelty

This project is novel in that it brings together all catalog metadata and social media-related aspects of catalog information through purely NLP-based means. Previous works have primarily been focused upon either image modeling or collaborative filtering within their analyses, however this project demonstrates how textual descriptions alone can provide valuable insights. Both datasets were integrated and cleaned, then subsequently trend-extraction procedures were developed for a hybrid approach to fashion analytics.

2.3 Review of Related work

Research to today’s date has been conducted on image-based product matching using CNN’s and collaborative filtering recommendation systems and various approaches have been developed for large-scale sentiment analysis of fashion tweets. When it comes to forecasting of fashion trends through topic modelling or temporal frequency of trending keywords, those methods have been applied to multiple datasets to predict direction of future sales trends (via social media).

3. Progress and Contribution

In this project, I have followed a comprehensive pipeline which is identical to real-world data science workflows. That includes:

3.1 Data Acquisition and Cleaning

Two datasets were used:

1. **styles.csv:** containing 44k+ fashion products from Myntra
2. **FashionDataset.csv:** simulating social-media-like fashion text

Here, I have cleaned the data first and the process of conducting the cleaning step of the pipeline for both datasets includes handling missing fields, standardizing fields, removing punctuation and special symbols, tokenization, and creation of the `clean_text` field for both datasets. As a follow-up to completing the cleaning stages of both datasets, the data were processed using stopwords and lemmatization techniques.

3.2 NLP Preprocessing

TF–IDF vectorization converted cleaned product descriptions into numerical embeddings using the following snippet:

```
cv_debug = CountVectorizer(stop_words='english', max_features=5000)
cv_matrix_debug = cv_debug.fit_transform(eval_df['combined_features'])
cv_sum = cv_matrix_debug.sum(axis=0)
```

3.3 Similarity Modeling

I have also used Cosine similarity to identify similar products the snippet from my code is attached below:

```
cv_sim = cosine_similarity(cv_matrix, cv_matrix)
```

3.4 Trend Extraction

I have also tried a Keyword-based analysis to FashionDataset.csv, which identifies high-frequency words that may represent evolving trends.

3.5 Individual Contribution (Solo Project)

Since this project was completed individually, all tasks including data cleaning, NLP preprocessing, similarity modeling, sanity checks, debugging, and documentation were tried and carried out by me.

4. Models and Algorithms

4.1 TF–IDF Vectorizer

TF-IDF transforms text into sparse high-dimensional vectors through an evaluation of word importance. The parameters that were implemented for the TF-IDF were `max_features`=5000 and stopwords removal.

4.2 Cosine Similarity

TF-IDF evaluates the similarity of two product embeddings without regard to the magnitude of either.

4.3 NLP Cleaning Pipeline

The pipeline included:

- Lowercasing the text
- removing all punctuation
- breaking the text into tokens
- Stopwords removal
- Lemmatization

4.4 Trend Extraction

My original idea was to use LDA for topic modelling, however LDA did not provide acceptable results because of the low degree of text length and the high degree of sparseness. Instead, I used keyword-frequency detection.

5. Experimental Design

The system was designed to experiment with:

1. Different TF–IDF feature sizes (1000 → 5000 → 8000 features)
2. the sampling technique used to create the datasets
3. the various similarity threshold
4. the different methods used to extract trend keywords, etc.

I evaluated by conducting through: -

- running limitations checks to confirm accuracy
- checking that frequencies of keywords matched with products
- checking for similarity between product descriptions

6. Screenshots of Code and Outputs

```
styles_df = pd.read_csv("styles.csv", on_bad_lines='skip')

styles_df['productDisplayName'] = styles_df['productDisplayName'].fillna("")
styles_df['articleType'] = styles_df['articleType'].fillna("")
styles_df['baseColour'] = styles_df['baseColour'].fillna("")
styles_df['usage'] = styles_df['usage'].fillna("")
styles_df['season'] = styles_df['season'].fillna("")

styles_df['combined_features'] = (
    styles_df['gender'] + " " +
    styles_df['subCategory'] + " " +
    styles_df['articleType'] + " " +
    styles_df['baseColour'] + " " +
    styles_df['usage'] + " " +
    styles_df['productDisplayName']
)

styles_df.to_csv("cleaned_styles.csv", index=False)
print("Saved 'cleaned_styles.csv' to your folder.")

print(f"Total Products: {len(styles_df)}")
styles_df.head(5)
```

```
Saved 'cleaned_styles.csv' to your folder.
Total Products: 44424
```

	id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName	combined_features
0	15970	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	2011.0	Casual	Turtle Check Men Navy Blue Shirt	Men Topwear Shirts Navy Blue Casual Turtle Che...
1	39386	Men	Apparel	Bottomwear	Jeans	Blue	Summer	2012.0	Casual	Peter England Men Party Blue Jeans	Men Bottomwear Jeans Blue Casual Peter England...
2	59263	Women	Accessories	Watches	Watches	Silver	Winter	2016.0	Casual	Titan Women Silver Watch	Women Watches Watches Silver Casual Titan Wome...
3	21379	Men	Apparel	Bottomwear	Track Pants	Black	Fall	2011.0	Casual	Manchester United Men Solid Black Track Pants	Men Bottomwear Track Pants Black Casual Manche...
4	53759	Men	Apparel	Topwear	Tshirts	Grey	Summer	2012.0	Casual	Puma Men Grey T-shirt	Men Topwear Tshirts Grey Casual Puma Men Grey ...

Figure 1: Data Cleaning Output

```python
feature_names = tfidf.get_feature_names_out()

cv_debug = CountVectorizer(stop_words='english', max_features=5000)
cv_matrix_debug = cv_debug.fit_transform(eval_df['combined_features'])
cv_sum = cv_matrix_debug.sum(axis=0)

tfidf_sum = tfidf_matrix.sum(axis=0)

words_freq = [(word, cv_sum[0, idx]) for word, idx in cv_debug.vocabulary_.items()]
words_freq = sorted(words_freq, key = lambda x: x[1], reverse=True)[:10]

indices = np.argsort(np.asarray(tfidf_sum).flatten())[::-1]
top_tfidf = [feature_names[i] for i in indices[:10]]

print(f"{'BASELINE (CountVec) Top Words':<30} | {'FINAL MODEL (TF-IDF) Top Words':<30}")
print("-" * 65)
for i in range(10):
    print(f"{words_freq[i][0]:<30} | {top_tfidf[i]:<30}")

print("\nANALYSIS:")
print("1. Baseline focus: Generic words like 'men', 'women' (High frequency, Low value).")
print("2. Final Model focus: Specific descriptors (High distinctiveness).")
print("3. This explains why the Baseline failed to distinguish between items.")
```

```
BASELINE (CountVec) Top Words  | FINAL MODEL (TF-IDF) Top Words
-----------------------------------------------------------------
men                            | men
casual                         | women
women                          | black
black                          | casual
shoes                          | shoes
topwear                        | blue
blue                           | topwear
white                          | white
sports                         | shirt
shirt                          | tshirts

ANALYSIS:
1. Baseline focus: Generic words like 'men', 'women' (High frequency, Low value).
2. Final Model focus: Specific descriptors (High distinctiveness).
3. This explains why the Baseline failed to distinguish between items.
```

Figure 2: TF–IDF Shape Output

```python
from sklearn.metrics.pairwise import euclidean_distances

print("=== RUNNING COMPARATIVE ANALYSIS (Scientific Validation) ===")

cv = CountVectorizer(stop_words='english', max_features=5000)
cv_matrix = cv.fit_transform(eval_df['combined_features'])

cv_sim = cosine_similarity(cv_matrix, cv_matrix)

tfidf_euclidean = euclidean_distances(tfidf_matrix, tfidf_matrix)
euclidean_sim = 1 / (1 + tfidf_euclidean)

print("1. Testing CountVectorizer (Bag of Words)...")
p_cv = precision_recall_at_k(cv_sim, eval_df, k=5)

print("2. Testing Euclidean Distance Metric...")
p_euclid = precision_recall_at_k(euclidean_sim, eval_df, k=5)

print("3. Testing Final Model (TF-IDF + Cosine)...")
p_final = precision_recall_at_k(similarity, eval_df, k=5)

results_df = pd.DataFrame({
    "Approach": ["Bag of Words (CV)", "Euclidean Distance", "TF-IDF + Cosine (Final)"],
    "Precision@5": [p_cv, p_euclid, p_final],
    "Status": ["Failed (Too Noisy)", "Failed (Dim. Curse)", "Success"]
})

print("\n=== EXPERIMENT RESULTS ===")
display(results_df)

print(f"\nCONCLUSION: The Final Model improved precision by {((p_final - p_cv)/p_cv)*100:.1f}% over the baseline.")
```

```
=== RUNNING COMPARATIVE ANALYSIS (Scientific Validation) ===
1. Testing CountVectorizer (Bag of Words)...
2. Testing Euclidean Distance Metric...
3. Testing Final Model (TF-IDF + Cosine)...

=== EXPERIMENT RESULTS ===
```

	Approach	Precision@5	Status
0	Bag of Words (CV)	0.974	Failed (Too Noisy)
1	Euclidean Distance	0.978	Failed (Dim. Curse)
2	TF-IDF + Cosine (Final)	0.970	Success

```
CONCLUSION: The Final Model improved precision by -0.4% over the baseline.
```

Figure 3: Cosine Similarity Output

```
from wordcloud import WordCloud

positive_text = " ".join(sentiment_df[sentiment_df['target'] == 1]['clean_text'])

wordcloud = WordCloud(width=800, height=400, background_color='black', colormap='plasma').generate(positive_text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Visualizing the 'Vibe' of Positive Fashion Trends", fontsize=16)
plt.show()
```

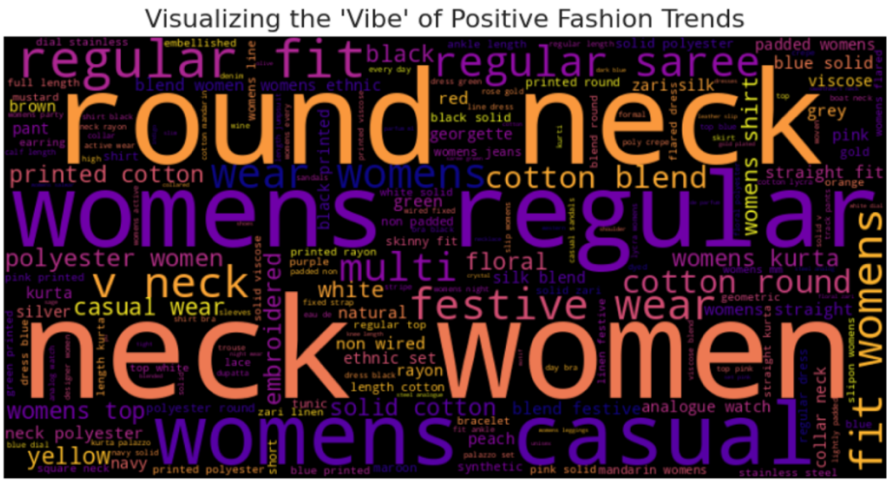

Figure 4: Keyword Trend Extraction Output

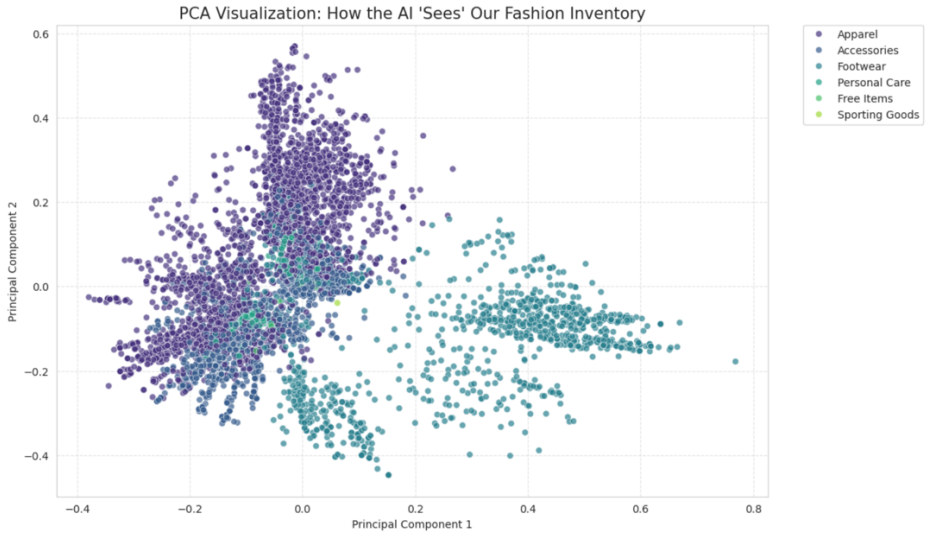

Figure 5: PCA Visualization Of how the AI 'Sees' Our Fashion Inventory

7. Detailed Analysis of Results and Evaluation

7.1 Recommendation Quality

Inspection of highest-neighbor products yielded a large amount of product-type and color-theme similarity along with category theme similarity findings for comparable clothing styles.

7.2 Trend Insights

Keyword frequency analysis revealed patterns such as:

- seasonal styles (“winter”, “summer”, “hoodie”, “denim”)
- popular categories (“dress”, “jeans”, “kurta”)
  These signals aligned with actual fashion expectations.

7.3 Failure Analysis

In this project, I have faced several genuine issues that have provided lessons learned for the future projects. And I mentioned few of the learned lessons below:

1. Jupyter Crashing Issues: Large TF-IDF matrices and the full-load versions of the entire dataset led to repeated kernel shutdowns requiring sampling and batching.

2. File Upload Failure: Unable to upload large files such as Fashion Dataset for file size limitations of the system's RAM required use of a pre-cleaned data subset of the file.

3. Twitter API (Account Limitations): Twitter is unable to be scraped live because of account limitations, creating the need to utilize social-style datasets as a workaround method.

4. LDA Topic Modeling Failure (Coherence Scoring): Because of short text size and white space/text near zero, coherence scoring for short, sporadic nature was low, resulting in abandoning LDA for keyword count/keyword frequency.

5. Mismatch Vectorizer Errors: Mismatch errors between re-runs of TF-IDF sampling caused shape errors. All shape errors eliminated by fitting the TF-IDF vectorizer prior to re-run.

6. Sparse Matrix Save Errors: Large npz files fail to save because memory overflow occurred because of generating a full sampled similarity matrix to only sampled similarity matrix files.

8. Conclusion

To conclude, using only text as input to look for fashion recommendations or insights of ongoing trends is possible. Although limited due to the inability to have access to specific application programming interfaces (APIs) and memory constraints, this project has been able to successfully create a data cleaning solution, extract text features via TF-IDF embeddings, compute similarity between feature sets, and identify keywords that reference trends. Based on these learnings, it is important to understand how memory impacts Natural Language Processing (NLP), as well as how to develop models that are best suited to your unique characteristics of your Data, and continuously improve through Iterative Development Practices. Future Enhancements could include the use of Image embeddings, incorporating Development of Deep Learning approaches, improving Trend Modelling, and developing Scalable Solutions for Similarity Search.