FINAL PROJECT REPORT

On

ANALYSIS OF CHEMICAL COMPONENTS IN COSMETICS

Prepared By

SUHANI PARMAR

TABLE OF CONTENTS

ABSTRACT

Understanding cosmetic product ingredients is essential for consumers, especially those with sensitive skin, as it helps them make informed purchasing decisions. This project tackles the complexity of ingredient lists by utilizing data science to develop a content-based recommendation system for cosmetics. By analysing the chemical components of 1,472 products, the system identifies patterns and visualizes ingredient similarities through advanced machine learning techniques like t-SNE. An interactive scatter plot was created using Bokeh, providing users with a clear and intuitive exploration tool for comparing products. Despite challenges such as data sparsity and ingredient variability, the project successfully demonstrates how data science can enhance consumer decision-making in the cosmetics industry.

1.1 About the Project

The "Analysis of Chemical Components" project aims to bridge the gap between complex ingredient information and consumer needs. It leverages machine learning to process ingredient data and create a user-friendly visualization of product similarities. This project is particularly significant for individuals with specific skin concerns, offering a scientific approach to choosing suitable cosmetics.

1.2 Objectives and Deliverables

Objectives:

- Develop a recommendation system based on cosmetic ingredient data.
- Apply machine learning techniques to analyze and reduce ingredient data dimensions.
- Create an interactive visualization to showcase product similarities.

Deliverables:

- A cleaned and pre-processed dataset of cosmetic ingredients.
- A dimensionality reduction model using t-SNE.
- An interactive scatter plot visualized with Bokeh, featuring product details accessible through a hover tool.

METHODOLOGY

2.1 Data Preprocessing

1. **Data Filtering**: The dataset was initially filtered to focus on a specific category, "Moisturizers," suitable for dry skin. This targeted approach ensured a relevant and manageable dataset for the analysis.
2. **Ingredient Tokenization**: Each product's ingredient list was tokenized into individual components. This step converted the textual ingredient data into structured tokens, which could be further analyzed.
3. **Document-Term Matrix (DTM)**: A DTM was constructed to represent the presence of specific ingredients across different products. Each row of the matrix represented a product, and each column corresponded to an ingredient.

2.2 Encoding Ingredients

1. **One-Hot Encoding**: A custom one-hot encoder was implemented to convert ingredient tokens into binary vectors. These vectors represented the presence or absence of specific ingredients in each product.
2. **Cosmetic-Ingredient Matrix**: The encoded vectors were aggregated into a matrix format, enabling further computations and analyses. This matrix served as the foundation for similarity evaluation.

2.3 Dimensionality Reduction

1. **t-SNE (t-distributed Stochastic Neighbor Embedding)**:
   - This technique was used to reduce the high-dimensional ingredient vectors into a two-dimensional space for visualization.
   - Parameters included:
     - n_components=2: To reduce dimensions to two.
     - learning_rate=200: To optimize convergence.
     - random_state=42: For consistent results across runs.

2.4 Visualization

1. **Interactive Scatter Plot**:
   - The Bokeh library was used to create an interactive scatter plot.
   - Each point on the plot represented a cosmetic product, with x and y coordinates derived from the t-SNE transformation.

2. **Hover Tool**:

- A hover feature was added to display product details, including:
    - Product Name
    - Brand
    - Price
    - Rank

2.5 Similarity Analysis

1. **Product Clustering**:
    - Products with similar ingredient profiles were identified by their proximity on the scatter plot.
2. **Validation**:
    - Two closely positioned products were compared to verify ingredient similarity, validating the accuracy of the recommendation system.

2.6 Packages Used

1. **Pandas**: Utilized for data manipulation and cleaning, including filtering and processing the dataset.

2. **NumPy**: Used for numerical computations, especially in constructing the Document-Term Matrix and encoding ingredient vectors.

3. **scikit-learn**:
    - Provided the t-SNE implementation for dimensionality reduction.
    - Enabled efficient transformation of high-dimensional data into a two-dimensional space.

4. **Bokeh**:
    - Used to create interactive visualizations, including scatter plots with hover tools for enhanced user interaction.

5. **Python Standard Libraries**:
    - Libraries like re and basic Python data structures were employed for tokenization and data parsing tasks.

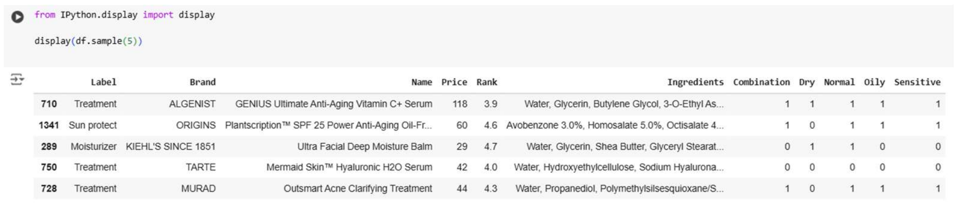
These packages collectively supported the entire workflow, from data preprocessing to visualization, ensuring an efficient and scalable analysis pipeline.

IMPLEMENTATION

Before delving into the specifics of the implementation process, it is important to highlight the systematic approach undertaken in this project. The implementation involved multiple steps, from data importation and preprocessing to advanced visualization and similarity analysis. Each stage was carefully executed to ensure the reliability and usability of the final product recommendation system.

3.1 Data Import and Exploration

1. **Loading the Dataset**:
   - The dataset was imported using Pandas and stored in a DataFrame. This provided a structured format to work with tabular data efficiently.
   - Sample rows and summary statistics were inspected to understand the structure and key features of the data.

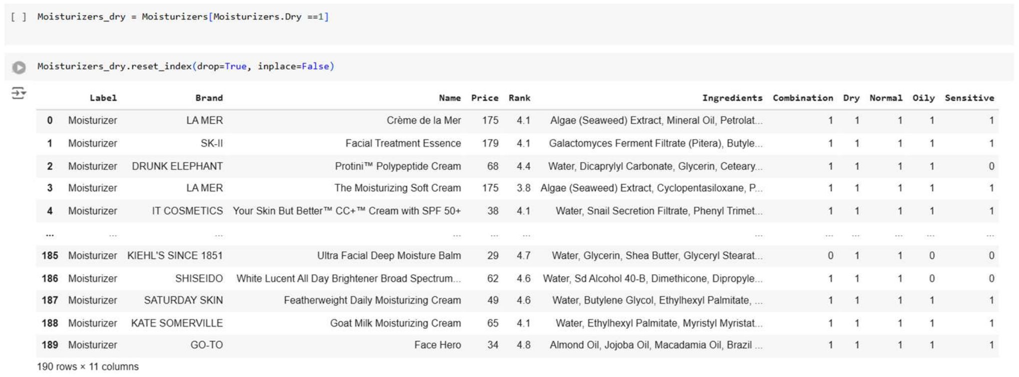


```
from IPython.display import display

display(df.sample(5))
```

	Label	Brand	Name	Price	Rank	Ingredients	Combination	Dry	Normal	Oily	Sensitive
710	Treatment	ALGENIST	GENIUS Ultimate Anti-Aging Vitamin C+ Serum	118	3.9	Water, Glycerin, Butylene Glycol, 3-O-Ethyl As...	1	1	1	1	1
1341	Sun protect	ORIGINS	Plantscription™ SPF 25 Power Anti-Aging Oil-Fr...	60	4.6	Avobenzone 3.0%, Homosalate 5.0%, Octisalate 4...	1	0	1	1	1
289	Moisturizer	KIEHL'S SINCE 1851	Ultra Facial Deep Moisture Balm	29	4.7	Water, Glycerin, Shea Butter, Glyceryl Stearat...	0	1	1	0	0
750	Treatment	TARTE	Mermaid Skin™ Hyaluronic H2O Serum	42	4.0	Water, Hydroxyethylcellulose, Sodium Hyalurona...	0	0	0	0	0
728	Treatment	MURAD	Outsmart Acne Clarifying Treatment	44	4.3	Water, Propanediol, Polymethylsilsesquioxane/S...	1	0	1	1	1

2. **Product Filtering**:
   - To focus on moisturizers suitable for dry skin, the dataset was filtered for the "Moisturizer" category and further refined to include only products targeting dry skin conditions.

3. **Index Reset**:
   - After filtering, the DataFrame index was reset to ensure clean and continuous indexing for further operations.



```
Moisturizers_dry = Moisturizers[Moisturizers.Dry ==1]
```

```
Moisturizers_dry.reset_index(drop=True, inplace=False)
```

	Label	Brand	Name	Price	Rank	Ingredients	Combination	Dry	Normal	Oily	Sensitive
0	Moisturizer	LA MER	Crème de la Mer	175	4.1	Algae (Seaweed) Extract, Mineral Oil, Petrolat...	1	1	1	1	1
1	Moisturizer	SK-II	Facial Treatment Essence	179	4.1	Galactomyces Ferment Filtrate (Pitera), Butyle...	1	1	1	1	1
2	Moisturizer	DRUNK ELEPHANT	Protini™ Polypeptide Cream	68	4.4	Water, Dicaprylyl Carbonate, Glycerin, Ceteary...	1	1	1	1	0
3	Moisturizer	LA MER	The Moisturizing Soft Cream	175	3.8	Algae (Seaweed) Extract, Cyclopentasiloxane, P...	1	1	1	1	1
4	Moisturizer	IT COSMETICS	Your Skin But Better™ CC+™ Cream with SPF 50+	38	4.1	Water, Snail Secretion Filtrate, Phenyl Trimet...	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...
185	Moisturizer	KIEHL'S SINCE 1851	Ultra Facial Deep Moisture Balm	29	4.7	Water, Glycerin, Shea Butter, Glyceryl Stearat...	0	1	1	0	0
186	Moisturizer	SHISEIDO	White Lucent All Day Brightener Broad Spectrum...	62	4.6	Water, Sd Alcohol 40-B, Dimethicone, Dipropyle...	1	1	1	0	0
187	Moisturizer	SATURDAY SKIN	Featherweight Daily Moisturizing Cream	49	4.6	Water, Butylene Glycol, Ethylhexyl Palmitate, ...	1	1	1	1	1
188	Moisturizer	KATE SOMERVILLE	Goat Milk Moisturizing Cream	65	4.1	Water, Ethylhexyl Palmitate, Myristyl Myristat...	1	1	1	1	1
189	Moisturizer	GO-TO	Face Hero	34	4.8	Almond Oil, Jojoba Oil, Macadamia Oil, Brazil ...	1	1	1	1	1

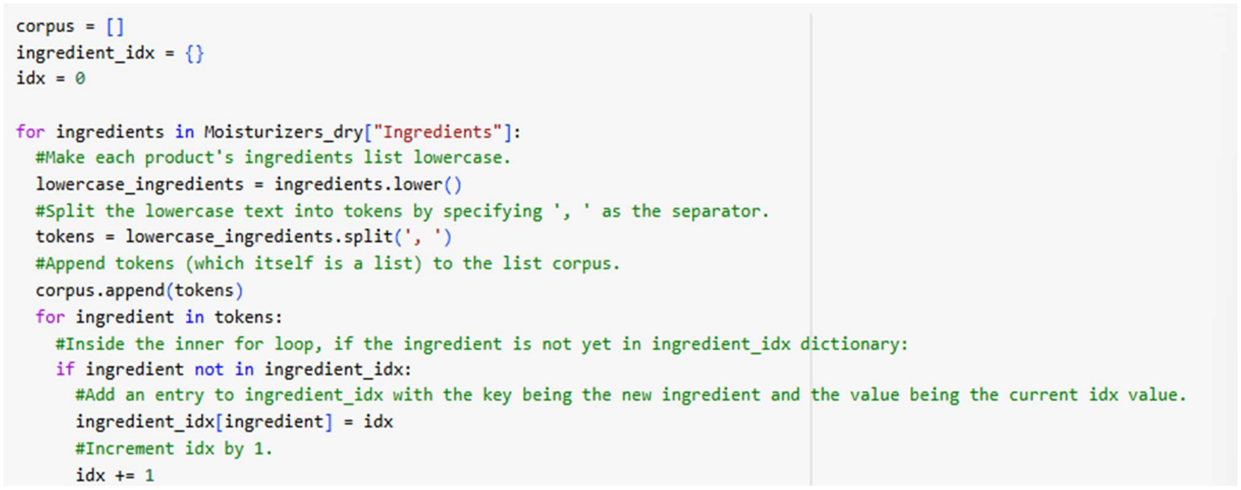
190 rows × 11 columns

3.2 Tokenization and Ingredient Processing

1. **Ingredient Cleaning**:
    - Ingredients lists were converted to lowercase to standardize the text and then split into tokens using a defined separator (,).

2. **Corpus Creation**:
    - A list of tokenized ingredients was created for all products, serving as the primary data structure for encoding.

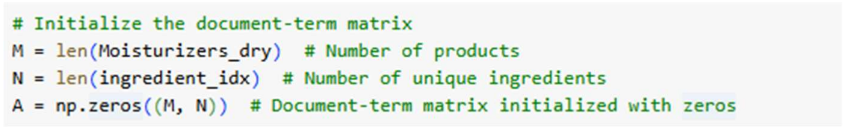
3. **Ingredient Indexing**:
    - A dictionary mapping each unique ingredient to a numerical index was built. This facilitated efficient matrix representation in later steps.



```
corpus = []
ingredient_idx = {}
idx = 0

for ingredients in Moisturizers_dry["Ingredients"]:
  #Make each product's ingredients list lowercase.
  lowercase_ingredients = ingredients.lower()
  #Split the lowercase text into tokens by specifying ', ' as the separator.
  tokens = lowercase_ingredients.split(', ')
  #Append tokens (which itself is a list) to the list corpus.
  corpus.append(tokens)
  for ingredient in tokens:
    #Inside the inner for loop, if the ingredient is not yet in ingredient_idx dictionary:
    if ingredient not in ingredient_idx:
      #Add an entry to ingredient_idx with the key being the new ingredient and the value being the current idx value.
      ingredient_idx[ingredient] = idx
      #Increment idx by 1.
      idx += 1
```

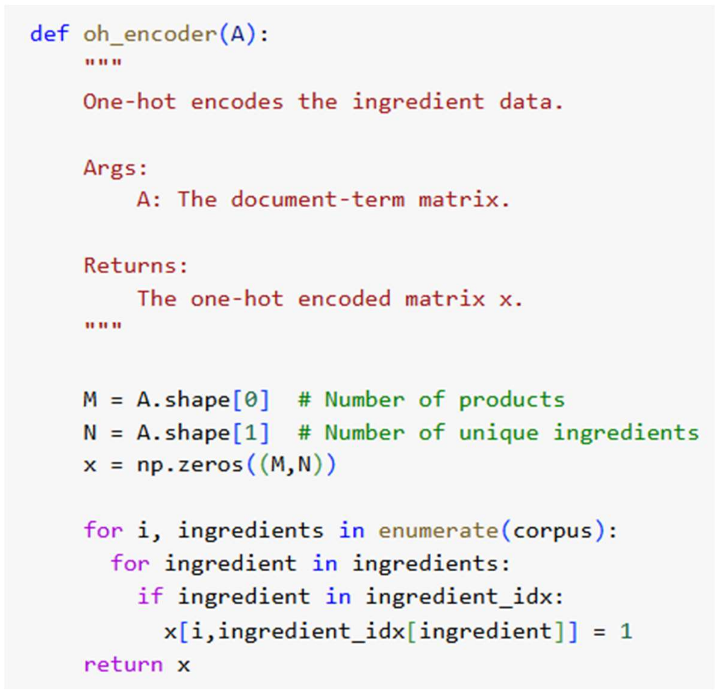
3.3 Constructing the Document-Term Matrix (DTM)

1. **Matrix Initialization**:
    - A matrix of zeros with dimensions corresponding to the number of products and unique ingredients was initialized.



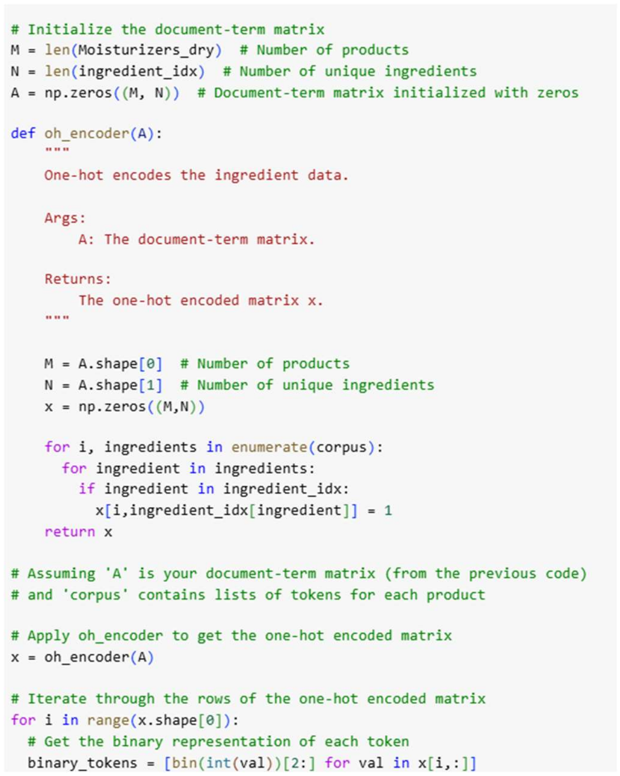
```
# Initialize the document-term matrix
M = len(Moisturizers_dry)  # Number of products
N = len(ingredient_idx)  # Number of unique ingredients
A = np.zeros((M, N))  # Document-term matrix initialized with zeros
```

2. **One-Hot Encoding**:
    - A custom function was developed to encode the presence of each ingredient in a product's list as a binary value (1 for presence, 0 for absence).
    - This function iterated over the tokenized ingredient lists and populated the matrix.



```python
def oh_encoder(A):
    """
    One-hot encodes the ingredient data.

    Args:
        A: The document-term matrix.

    Returns:
        The one-hot encoded matrix x.
    """

    M = A.shape[0]  # Number of products
    N = A.shape[1]  # Number of unique ingredients
    x = np.zeros((M,N))

    for i, ingredients in enumerate(corpus):
      for ingredient in ingredients:
        if ingredient in ingredient_idx:
          x[i,ingredient_idx[ingredient]] = 1
    return x
```

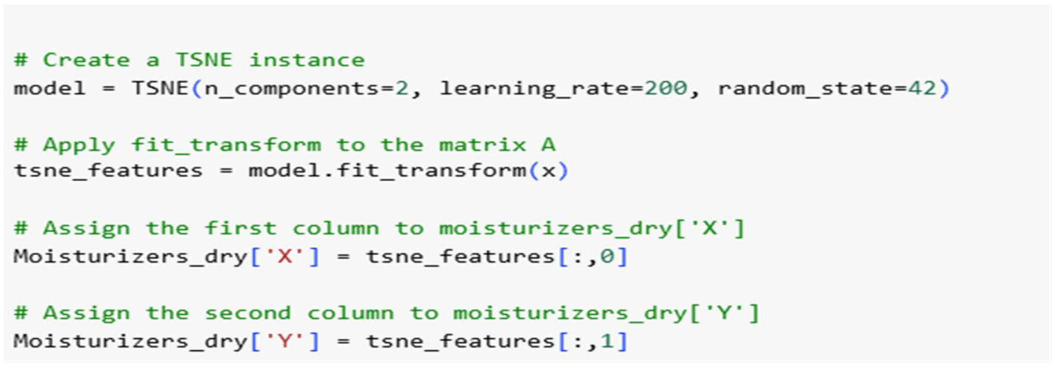
3. **Cosmetic-Ingredient Matrix**:
   - The completed DTM provided a structured and binary representation of product ingredient compositions.



```python
# Initialize the document-term matrix
M = len(Moisturizers_dry)  # Number of products
N = len(ingredient_idx)  # Number of unique ingredients
A = np.zeros((M, N))  # Document-term matrix initialized with zeros

def oh_encoder(A):
    """
    One-hot encodes the ingredient data.

    Args:
        A: The document-term matrix.

    Returns:
        The one-hot encoded matrix x.
    """

    M = A.shape[0]  # Number of products
    N = A.shape[1]  # Number of unique ingredients
    x = np.zeros((M,N))

    for i, ingredients in enumerate(corpus):
      for ingredient in ingredients:
        if ingredient in ingredient_idx:
          x[i,ingredient_idx[ingredient]] = 1
    return x

# Assuming 'A' is your document-term matrix (from the previous code)
# and 'corpus' contains lists of tokens for each product

# Apply oh_encoder to get the one-hot encoded matrix
x = oh_encoder(A)

# Iterate through the rows of the one-hot encoded matrix
for i in range(x.shape[0]):
  # Get the binary representation of each token
  binary_tokens = [bin(int(val))[2:] for val in x[i,:]]
```

3.4 Dimensionality Reduction Using t-SNE

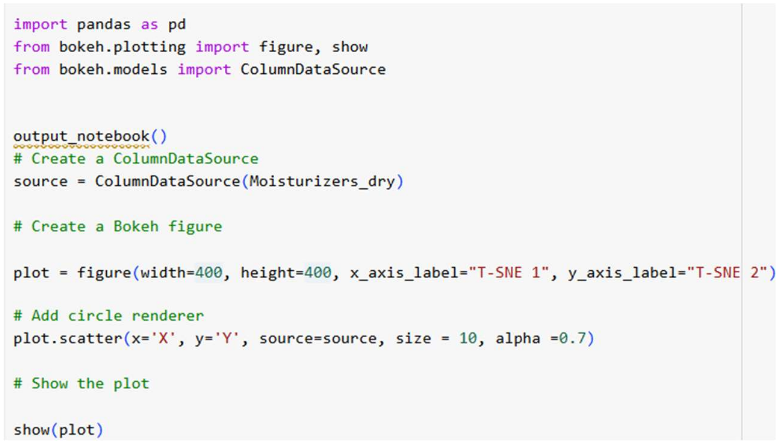
1. **t-SNE Configuration**:
   - The t-SNE algorithm was initialized with specific parameters:
     - n_components=2 to project data onto a two-dimensional plane.
     - learning_rate=200 for effective optimization.
     - random_state=42 to ensure reproducibility.
2. **Transformation**:
   - The high-dimensional ingredient data matrix was transformed into a 2D space using the fit_transform method of the t-SNE model.
   - The resulting coordinates were added as new columns (X and Y) in the filtered DataFrame.



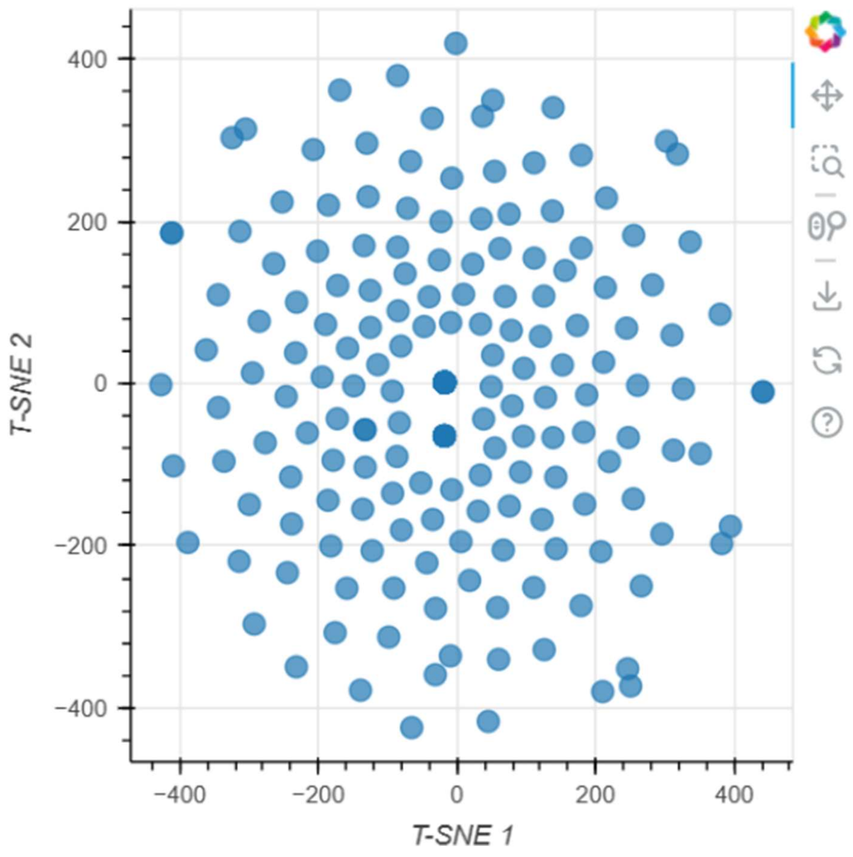
```
# Create a TSNE instance
model = TSNE(n_components=2, learning_rate=200, random_state=42)

# Apply fit_transform to the matrix A
tsne_features = model.fit_transform(x)

# Assign the first column to moisturizers_dry['X']
Moisturizers_dry['X'] = tsne_features[:,0]

# Assign the second column to moisturizers_dry['Y']
Moisturizers_dry['Y'] = tsne_features[:,1]
```

3.5 Visualization with Bokeh

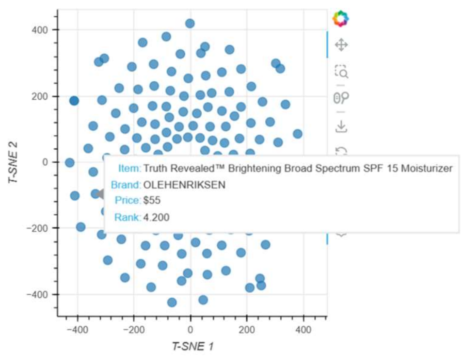
1. **Scatter Plot Generation**:
   - Using the Bokeh library, an interactive scatter plot was created, mapping the t-SNE-generated X and Y coordinates.



```
import pandas as pd
from bokeh.plotting import figure, show
from bokeh.models import ColumnDataSource


output_notebook()
# Create a ColumnDataSource
source = ColumnDataSource(Moisturizers_dry)

# Create a Bokeh figure

plot = figure(width=400, height=400, x_axis_label="T-SNE 1", y_axis_label="T-SNE 2")

# Add circle renderer
plot.scatter(x='X', y='Y', source=source, size = 10, alpha =0.7)

# Show the plot

show(plot)
```



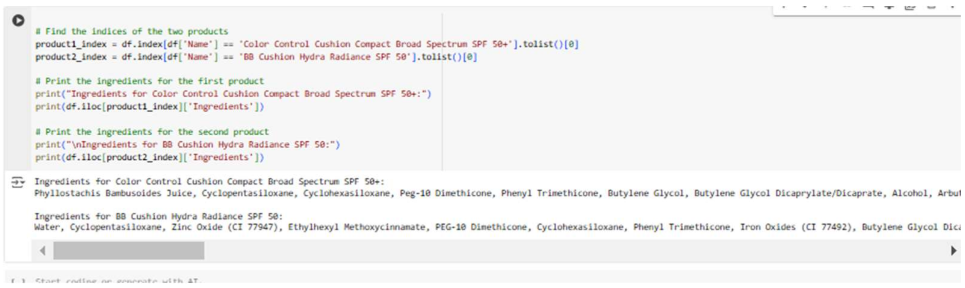
2. **Adding Hover Tools**:
    - Tooltips were implemented to display key product details, including:
        - Product Name
        - Brand
        - Price
        - Rank
3. **Interactive Features**:
    - Users could interact with the plot to explore ingredient similarities and related product information visually.



Item: Truth Revealed™ Brightening Broad Spectrum SPF 15 Moisturizer
Brand: OLEHENRIKSEN
Price: $55
Rank: 4.200

3.6 Product Comparison

1. **Identifying Similar Products**:
   - Two products positioned closely in the t-SNE plot were selected for comparison.
2. **Ingredient Analysis**:
   - The ingredient lists of these products were printed and manually reviewed to verify their similarity, ensuring the effectiveness of the recommendation system.



```python
# Find the indices of the two products
product1_index = df.index[df['Name'] == 'Color Control Cushion Compact Broad Spectrum SPF 50+'].tolist()[0]
product2_index = df.index[df['Name'] == 'BB Cushion Hydra Radiance SPF 50'].tolist()[0]

# Print the ingredients for the first product
print("Ingredients for Color Control Cushion Compact Broad Spectrum SPF 50+:")
print(df.iloc[product1_index]['Ingredients'])

# Print the ingredients for the second product
print("\nIngredients for BB Cushion Hydra Radiance SPF 50:")
print(df.iloc[product2_index]['Ingredients'])
```

```
Ingredients for Color Control Cushion Compact Broad Spectrum SPF 50+:
Phyllostachis Bambusoides Juice, Cyclopentasiloxane, Cyclohexasiloxane, Peg-10 Dimethicone, Phenyl Trimethicone, Butylene Glycol, Butylene Glycol Dicaprylate/Dicaprate, Alcohol, Arbut

Ingredients for BB Cushion Hydra Radiance SPF 50:
Water, Cyclopentasiloxane, Zinc Oxide (CI 77947), Ethylhexyl Methoxycinnamate, PEG-10 Dimethicone, Cyclohexasiloxane, Phenyl Trimethicone, Iron Oxides (CI 77492), Butylene Glycol Dica
```

Start coding or generate with AI.

3.7 Final Output

1. **Validation of Results**:
   - The visualization and ingredient comparisons provided qualitative evidence of the model's performance.
2. **Deliverable**:
   - An interactive tool enabling users to explore cosmetic products based on ingredient similarity was successfully developed.

CONCLUSION AND FUTURE SCOPE

The project provided valuable experience in applying data analytics to real-world problems, particularly in the field of data visualization. The project successfully utilized machine learning techniques such as t-SNE to reduce the dimensionality of cosmetic ingredient data and create an interactive visualization tool. By implementing a content-based recommendation system, the project enhanced the user's ability to explore product similarities, ultimately helping consumers make more informed decisions. The challenges of data sparsity and ingredient variability were addressed effectively, ensuring a robust and user-friendly system.

The hands-on training and mentorship throughout this project have significantly contributed to my understanding of data analytics and its practical applications. The skills gained in data preprocessing, dimensionality reduction, and visualization using tools like Python, Pandas, Bokeh, and t-SNE will be invaluable for my future endeavours in data science and analytics.

Future Scope:

There is considerable potential for expanding this project into a fully-fledged cosmetic recommendation platform. Future improvements could include:

1. **Integration of Additional Data Sources:** Incorporating data on user reviews, product effectiveness, and skin sensitivity could provide a more comprehensive recommendation system.
2. **Enhanced Recommendation Algorithms:** Moving beyond basic similarity analysis to implement more advanced machine learning models like clustering, classification, or deep learning to provide personalized recommendations.
3. **Real-Time Data Updates:** Implementing a live data pipeline to keep the product information and ingredient lists up to date, allowing users to access the most current products and trends.
4. **Mobile Application Development:** A mobile app version of this tool would make it even more accessible to consumers while they shop, allowing for real-time ingredient comparisons and recommendations.
5. **Expansion to Other Product Categories:** Similar methods could be applied to other consumer goods, such as food, skincare, or household products, creating a versatile recommendation system for a wide range of industries.

Overall, this project lays a strong foundation for building more sophisticated systems that help consumers navigate complex product choices, with ample opportunities for enhancement and growth in the future.