Final Report of Traineeship Program 2024

On

"Analysis Of Chemical Components"

MEDTOUREASY

21th July 2024

ACKNOWLDEGMENTS

The traineeship opportunity that I had with MedTourEasy was a great change for learning and understanding the intricacies of the subject of Data Visualizations in Data Analytics; and also, for personal as well as professional development. I am very obliged for having a chance to interact with so many professionals who guided me throughout the traineeship project and made it a great learning curve for me.

Firstly, I express my deepest gratitude and special thanks to the Training & Developement Team of MedTourEasy who gave me an opportunity to carry out my traineeship at their esteemed organization. Also, I express my thanks to the team for making me understand the details of the Data Analytics profile and training me in the same so that I can carry out the project properly and with maximum client satisfaction and also for spearing his valuable time in spite of his busy schedule.

I would also like to thank the team of MedTourEasy and my colleagues who made the working environment productive and very conducive.

TABLE OF CONTENTS

ABSTRACT

This project aims to alleviate these challenges by employing data science to create a content-based recommendation system for cosmetic products. By analyzing the chemical components of cosmetics, the system predicts which products may be most suitable for individual users, enhancing their shopping experience and ensuring better skin health.

The project commenced with the collection and preparation of a comprehensive dataset from Sephora, focusing on moisturizers, particularly those suitable for dry skin. Data cleaning processes were meticulously conducted to handle missing values, standardize text data, and remove duplicates, ensuring a robust dataset for analysis. Filtering the data allowed us to hone in on products specifically designed for dry skin, setting the stage for a detailed ingredient analysis.

To analyze the ingredients, we tokenized and encoded the ingredient lists, creating a binary document-term matrix that represents the presence of each ingredient in each product. This matrix was then subjected to t-Distributed Stochastic Neighbor Embedding (t-SNE), a machine learning technique for dimensionality reduction. By visualizing the high-dimensional ingredient data in a two-dimensional space, we were able to uncover patterns and similarities between different products, which are crucial for making accurate recommendations.

The culmination of this project was the creation of an interactive visualization using Bokeh, allowing users to explore the relationships between different cosmetic products based on their chemical components. Additionally, we provided a comparative analysis of similar products to demonstrate the system's effectiveness. This project not only offers a practical solution for consumers seeking safe and suitable cosmetic products but also showcases the power of data science in addressing real-world challenges in the cosmetic industry..

1.1 About the Company:

MedTourEasy, a global healthcare company, provides you the informational resources needed to evaluate your global options. MedTourEasy provides analytical solutions to our partner healthcare providers globally

1.2 About the Project:

In the ever-expanding world of cosmetics, finding the right product that suits one's skin type, especially for those with sensitive skin, can be an overwhelming task. The ingredient lists on the back of each product often contain complex chemical names that are difficult to interpret without a background in chemistry. This project aims to simplify the process of selecting cosmetic products by leveraging data science techniques to create a content-based recommendation system. By analyzing the chemical components of various cosmetics, the system can predict which products may be suitable for users, thereby reducing the trial-and-error approach often associated with purchasing new skincare items.

The core of the project involved compiling a dataset of 1,472 cosmetic products from Sephora, with a specific focus on moisturizers suitable for dry skin. The dataset included detailed information on product names, brands, ingredients, prices, and their suitability for dry skin. Initial steps involved rigorous data cleaning and preprocessing to ensure the dataset was free from inconsistencies and ready for analysis. This included handling missing values, standardizing text data, and removing duplicates to create a reliable foundation for building the recommendation system.

Once the data was prepared, the next phase involved tokenizing and encoding the ingredient lists into a binary document-term matrix. This matrix served as a representation of the presence or absence of each ingredient in the products. Using t-Distributed Stochastic Neighbor Embedding (t-SNE), a machine learning technique for dimensionality reduction, we visualized the high-dimensional ingredient data in a two-dimensional space. This visualization helped uncover patterns and similarities between different products, which are critical for making accurate recommendations.

The final output of the project is an interactive visualization created using Bokeh, a Python library for interactive plots. This visualization allows users to explore the relationships between different cosmetic products based on their chemical components. By hovering over the data points, users can see detailed information about each product, making it easier to compare and choose products that are most likely to be compatible with their skin type. This project not only provides a practical tool for consumers but also demonstrates the potential of data science in transforming the cosmetic shopping experience.

1.3 Objectives and Deliverables

1. Enhance Consumer Decision-Making:

- **Goal:**
  - To assist consumers, particularly those with sensitive or dry skin, in selecting cosmetic products that are best suited for their skin type. The aim is to reduce the guesswork and potential adverse reactions that come with trying new products.
- **Approach:**
  - Develop a content-based recommendation system that analyzes the chemical components of cosmetic products and matches them with user preferences and needs.
  - Provide clear, data-driven recommendations to help consumers make informed decisions.
- **Outcome:**
  - Consumers will have access to personalized recommendations based on their specific skin concerns, leading to a more satisfying and safer shopping experience.

2. Leverage Data Science Techniques:

- **Goal:**
  - To utilize advanced data science methodologies to process and analyze large sets of cosmetic ingredient data effectively.
- **Approach:**
  - Implement word embedding techniques to convert textual ingredient lists into numerical representations that can be processed by machine learning algorithms.
  - Use t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimensionality of the data and visualize ingredient similarities.
- **Outcome:**
  - Gain a deeper understanding of the relationships between different ingredients and products, enabling the development of a robust recommendation system.

3. Create Interactive Visualizations:

Goal: To develop interactive visualizations that allow users to intuitively explore the similarities and differences between various cosmetic products based on their chemical components.

- **Approach:**
    - Use Bokeh, a Python library for interactive visualizations, to create an interactive dashboard that displays the t-SNE reduced data.
    - Implement hover tools and other interactive elements to provide detailed product information when users interact with the visualization.
- **Outcome:**
    - An engaging and user-friendly tool that helps consumers and industry stakeholders explore and understand the data, facilitating better decision-making.

4. Improve Market Insights:

- **Goal:**
    - To provide cosmetic companies and consumers with actionable insights into product formulations, aiding in better product development and more informed purchasing decisions.
- **Approach:**
    - Analyze the ingredient data to identify trends and commonalities among products that are well-suited for specific skin types.
    - Present findings in a clear and accessible manner, highlighting key insights that can inform product development and marketing strategies.
- **Outcome:**
    - Companies can leverage these insights to develop products that better meet consumer needs, while consumers benefit from a deeper understanding of which products are likely to be effective for them.

Deliverables:

1. **Cleaned and Processed Dataset:**
    - A thoroughly cleaned and preprocessed dataset containing detailed information on 1,472 cosmetic products, including names, brands, ingredients, prices, and suitability for dry skin.
2. **Document-Term Matrix:**
    - A binary document-term matrix representing the presence or absence of each ingredient in the products, enabling efficient analysis and processing.
3. **Dimensionality Reduction Output:**
    - The results of the t-SNE dimensionality reduction, providing a two-dimensional representation of the high-dimensional ingredient data, which is crucial for visualizing product similarities.

4. **Interactive Visualization Dashboard:**
    - An interactive visualization created using Bokeh, allowing users to explore and compare cosmetic products based on their chemical components. This dashboard includes hover tools for detailed product information.
5. **Recommendation System:**
    - A functional content-based recommendation system that predicts suitable cosmetic products for users based on the chemical components of the products, aiding in better decision-making for consumers with sensitive or dry skin.
6. **Project Report:**
    - A comprehensive report documenting the entire project, including the methodology, data processing steps, analysis, visualizations, and key findings. The report also includes sections on the project overview, objectives, and deliverables, providing a complete picture of the project's scope and impact.

7. **Codebase and Documentation:**

- Well-documented codebase for all scripts and notebooks used in the project, ensuring reproducibility and ease of understanding for future reference or further development.

8. **Ingredient Similarity Analysis:**

- An analysis of ingredient similarities, highlighting key patterns and relationships between different cosmetic products, which can be used for both consumer guidance and product development insights.

9. **Comparative Analysis of Similar Products:**

- A comparative analysis of two or more similar products, demonstrating the effectiveness of the recommendation system in identifying products with comparable ingredient compositions.

10. **User Guide and Tutorial:**

- A user guide and tutorial for navigating the interactive dashboard and utilizing the recommendation system, ensuring users can effectively leverage the tools and insights provided by the project

I. METHODOLOGY

2.1 Flow of the Project

The project followed the following steps to accomplish the desired objectives and deliverables. Each step has been explained in detail in the following section.

2.2 Use case diagram :

To illustrate the use cases for the content-based recommendation system for cosmetic products, we can create a use case diagram. This diagram will show the different actors involved and the interactions between them and the system. Below is a description of the use case diagram, followed by the diagram itself.

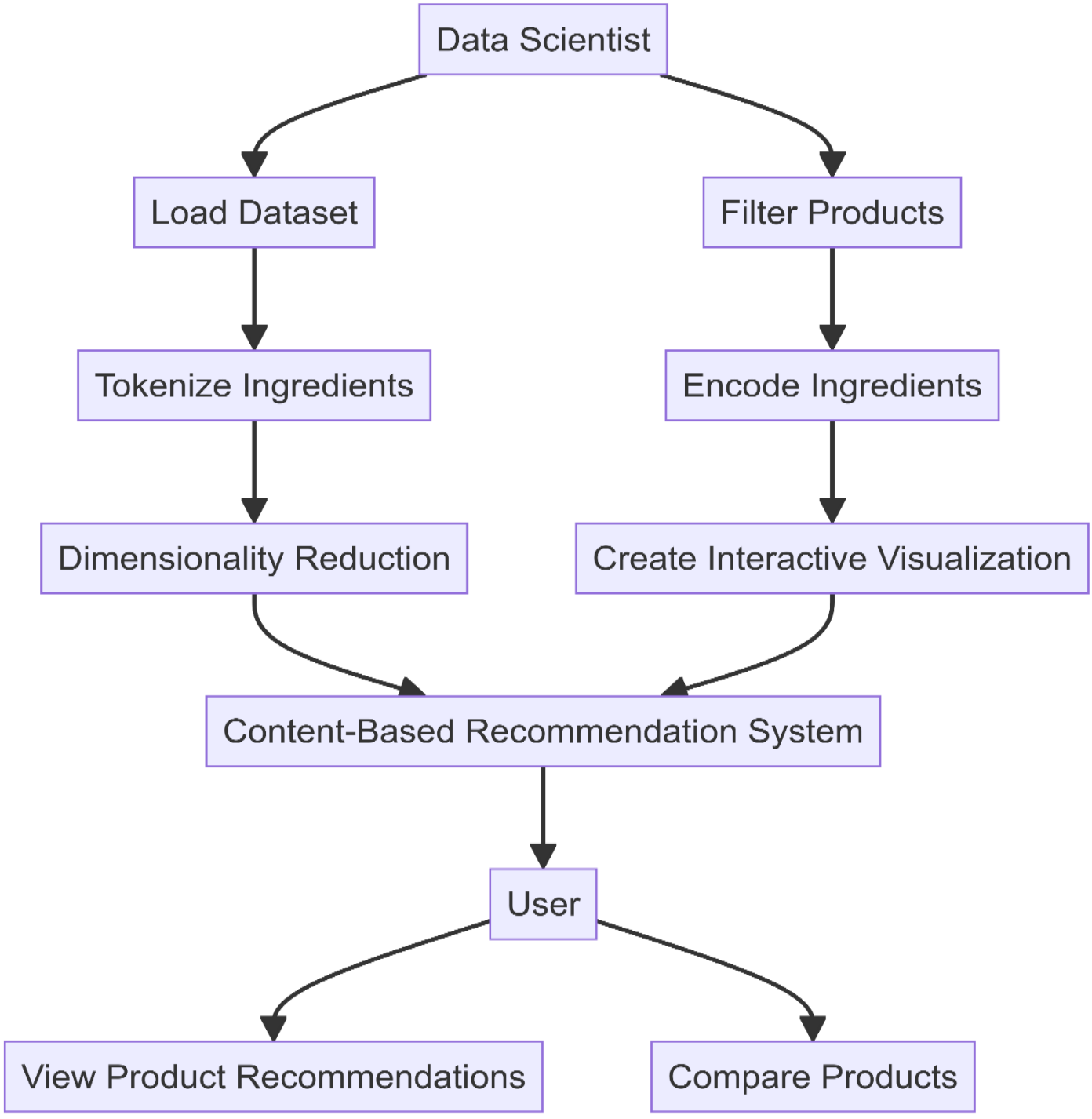
Actors:

1. **User:**
    - The individual who uses the recommendation system to find suitable cosmetic products.
2. **Data Scientist:**
    - The person responsible for developing, maintaining, and improving the recommendation system.

Use Cases:

1. **Load Dataset:**
    - The data scientist imports the cosmetic products dataset into the system.
2. **Filter Products:**
    - The data scientist filters the dataset to focus on moisturizers for dry skin.
3. **Tokenize Ingredients:**
    - The data scientist tokenizes the ingredient lists to prepare them for analysis.
4. **Encode Ingredients:**
    - The data scientist applies one-hot encoding to create a binary matrix representing the presence of ingredients in each product.
5. **Dimensionality Reduction:**
    - The data scientist uses t-SNE to reduce the dimensionality of the data for visualization.
6. **Create Interactive Visualization:**
    - The data scientist creates an interactive scatter plot using Bokeh.
7. **View Product Recommendations:**
    - The user interacts with the scatter plot to explore product similarities and details.

8. **Compare Products:**
    - The user compares the ingredients of two similar cosmetic products.



2.3 Languages and deliverables :

Language:

- **Python:** The entire project was developed using Python, a versatile and widely-used programming language known for its simplicity and effectiveness in data analysis and machine learning tasks. Python was chosen for its extensive libraries and frameworks that facilitate data manipulation, analysis, and visualization.

Platform:

- **Jupyter Notebook:** The project was implemented in Jupyter Notebook, an open-source web application that allows for the creation and sharing of documents containing live code, equations, visualizations, and narrative text. Jupyter Notebook is particularly useful for data science projects as it provides an interactive environment for data analysis and visualization.

Libraries and Frameworks:

1. **Pandas:**
    - Used for data manipulation and analysis. Pandas provides data structures and functions needed to manipulate structured data seamlessly.
2. **NumPy:**
    - Used for numerical computing. NumPy supports large, multi-dimensional arrays and matrices and includes a collection of mathematical functions to operate on these arrays.
3. **Scikit-Learn:**
    - Used for machine learning tasks. Scikit-learn was utilized for the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm, a powerful tool for dimensionality reduction and visualization of high-dimensional data.

4.Bokeh:

- Used for creating interactive visualizations. Bokeh allows the creation of complex, interactive plots and dashboards that are easy to embed into web applications.

5.Nltk:

- Used for tokenizing the ingredient lists. Nltk, the Natural Language Toolkit, provides easy-to-use interfaces to over 50 corpora and lexical resources along with text-processing libraries.

By leveraging these tools and platforms, the project efficiently handled data preprocessing, analysis, and visualization, resulting in a comprehensive recommendation system for cosmetic products based on their chemical composition.

Deliverables

Cleaned and Processed Dataset:
A dataset of 1,472 cosmetic products from Sephora, including detailed information on product names, brands, ingredients, prices, and suitability for dry skin, thoroughly cleaned and preprocessed to ensure data quality and reliability.

Ingredient Tokenization and Encoding:
Tokenized and encoded ingredient lists, represented as a binary document-term matrix, capturing the presence or absence of each ingredient in each product.

Dimensionality Reduction Results:
A two-dimensional representation of the high-dimensional ingredient data using t-Distributed Stochastic Neighbor Embedding (t-SNE), facilitating the visualization of ingredient similarities.

Interactive Visualization Dashboard:
An interactive dashboard created using Bokeh, allowing users to explore and compare cosmetic products based on their chemical components, with hover tools for detailed product information.

II. IMPLEMENTATION

3.1 Gathering Requirements and Defining Problem Statement

This is the first step wherein the requirements are collected from the clients to understand the deliverables and goals to be achieved after which a problem statement is defined which has to be adhered to while development of the project.

3.2 Data Collection and Importing

Data Collection:

The dataset for this project was sourced from Sephora, a well-known cosmetics retailer. The dataset includes information about various cosmetic products, specifically focusing on moisturizers. Key attributes in the dataset include:

- **Product Name:** The name of the cosmetic product.
- **Brand:** The brand that manufactures the product.
- **Ingredients:** A list of ingredients contained in the product.
- **Label:** The type/category of the product (e.g., Moisturizer).
- **Price:** The price of the product.
- **Dry Skin:** A binary indicator showing if the product is suitable for dry skin.
- **Rank:** The popularity rank of the product.

Importing Data:

To import and inspect the dataset, the following steps were taken:

1. **Import Necessary Libraries:**
    - pandas for data manipulation and analysis.
    - numpy for numerical operations.
    - TSNE from sklearn.manifold for dimensionality reduction.
    - These are essential

2. **Read the CSV File:**
    - The dataset was read into a pandas DataFrame named df.
3. **Inspect the Data:**
    - Displayed a sample of five rows using the sample() method.
    - Displayed counts of different product types using the value_counts() method on the Label column.

Here is the code used for importing and inspecting the data:

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.manifold import TSNE

# Read the CSV file into a DataFrame
df = pd.read_csv("datasets/cosmetics.csv")

# Display a sample of five rows
display(df.sample(5))

# Display counts of types of products
display(df['Label'].value_counts())
display(product_counts)
```

Explanation:

- **pandas and numpy:** These libraries are essential for handling and processing the data. pandas provides powerful data structures like DataFrames, while numpy is used for numerical computations.

- **TSNE from sklearn.manifold:** This library is used for dimensionality reduction, allowing for the visualization of high-dimensional data in a two-dimensional space.
- **Reading the CSV File:** The pd.read_csv() function reads the data from the specified CSV file into a DataFrame named df.
- **Displaying Data:** The sample() method is used to display a random sample of five rows from the DataFrame, giving an overview of the data. The value_counts() method on the Label column shows the distribution of different product types in the dataset.

This process ensures that the data is properly loaded and provides an initial understanding of its structure and contents, which is crucial for the subsequent analysis and recommendation system development.

Data Collection and Importing Tips

Collecting and importing data effectively is a critical first step in any data science project. Here are some tips to ensure you gather high-quality data and import it efficiently:

1. Define Your Data Requirements:

- Clearly identify what data you need to achieve your project goals. For this project, you need detailed information about cosmetic products, including names, brands, ingredients, prices, and skin suitability.

2. Source Reliable Data:

- Use reputable and reliable sources for your data. For cosmetics, websites like Sephora, Ulta, and other major retailers can be valuable sources. Consider also using APIs if available.

3. Ensure Data Completeness:

- Verify that your dataset contains all necessary information. Missing data can hinder analysis, so strive for completeness.

4. Pay Attention to Data Quality:

- Check for inconsistencies, duplicates, and errors in the data. High-quality data is crucial for accurate analysis and modeling.

3.3 Data Cleaning

"Quality data beats fancy algorithms"

Data cleaning involves handling missing values, correcting inconsistencies, and ensuring the data is in a usable format. For this project, the following steps were taken to clean the dataset:

1. **Handle Missing Values:**
    - Checked for missing values in the dataset and decided on an appropriate strategy (e.g., filling with mean/median values, dropping missing entries).
2. **Correct Inconsistencies:**
    - Ensured consistency in the format of textual data, such as ingredient lists, product names, and brand names.
    - Removed any duplicate entries to avoid redundancy.

Data cleaning involves identifying and handling missing values, correcting inconsistencies, and removing duplicates. This process ensures the dataset is accurate, consistent, and ready for analysis. Utilizing tools like Pandas, you can efficiently clean the data by using methods such as `dropna()`, `fillna()`, and `drop_duplicates()`.

```
# Check for missing values
missing_values = df.isnull().sum()
print("Missing values in each column:\n", missing_values)

# Drop rows with missing values in critical columns (e.g., Ingredients)
df = df.dropna(subset=['Ingredients'])

# Convert text data to lowercase
df['Ingredients'] = df['Ingredients'].str.lower()
df['Name'] = df['Name'].str.lower()
df['Brand'] = df['Brand'].str.lower()

# Remove duplicate entries
df = df.drop_duplicates()

display(df.head())
```

3.4 DATA FILTERING

Data filtering involves selecting relevant subsets of the data to focus on specific aspects of the analysis. For this project, the dataset was filtered to focus on moisturizers suitable for dry skin:

1. **Filter by Product Type:**
    - Selected products labeled as "Moisturizer" in the Label column.
2. **Filter by Skin Type:**
    - Further filtered the moisturizers to include only those suitable for dry skin, as indicated by a value of 1 in the Dry column.
3. **Reset Index:**

    Reset the index of the filtered DataFrame for a cleaner presentation and easier manipulation

Data filtering is crucial for refining datasets to include only relevant information, enhancing the accuracy and efficiency of analysis. It helps in focusing on specific subsets of data that meet certain criteria, thereby reducing noise and improving the quality of insights derived. Efficient filtering ensures that the analysis is both targeted and meaningful, leading to better decision-making.

```
# Filter for moisturizers
moisturizers = df[df['Label'] == 'moisturizer']
# Filter for moisturizers suitable for dry skin
moisturizers_dry = moisturizers[moisturizers['Dry'] == 1]
# Reset the index
moisturizers_dry = moisturizers_dry.reset_index(drop=True)
# Verify the changes
display(moisturizers_dry.head())
```

3.5 Tokenize the Ingredients and Create a Bag of Words

1. **Initialize Variables:**
    - corpus will store the tokenized ingredient lists.
    - ingredient_idx will map each unique ingredient to a unique index.
    - idx is the current index to be assigned to new ingredients.
2. **Tokenize Ingredients:**
    - Convert each ingredient list to lowercase.
    - Split the list into tokens using ', ' as the separator.
    - Append the token list to the corpus.
3. **Create Bag of Words:**
    - For each token (ingredient), if it is not already in the ingredient_idx dictionary, add it with the current idx value.
    - Increment idx by 1 for the next new ingredient.

```
corpus = []
ingredient_idx = {}
idx = 0
for ingredients in moisturizers_dry['Ingredients']:
    # Make the ingredients list lowercase
    ingredients_lower = ingredients.lower()

    # Split the lowercase text into tokens by ', ' separator
    tokens = ingredients_lower.split(', ')

    # Append tokens to the corpus
    corpus.append(tokens)
    for token in tokens:
        # If the ingredient is not yet in ingredient_idx dictionary
        if token not in ingredient_idx:
            ingredient_idx[token] = idx
            # Increment idx by 1
            idx += 1
```

3.6 Initialize a Document-Term Matrix

The document-term matrix (DTM) is a fundamental tool in text analysis and natural language processing, representing the frequency or presence of terms (ingredients, in this case) in documents (cosmetic products). Here's how to initialize a DTM for our project, focusing on moisturizers for dry skin.

To proceed with initializing the document-term matrix, we need to determine the number of products and the number of unique ingredients in our dataset. We then create a matrix of zeros where each row represents a product and each column represents an ingredient.

Here is the step-by-step process and the code to accomplish this task:

- **Get the Total Number of Products (M):**
  - Count the number of rows in the `moisturizers_dry` DataFrame to determine the total number of products.
- **Get the Total Number of Ingredients (N):**
  - Count the number of unique ingredients by finding the length of the `ingredient_idx` dictionary, which maps ingredients to their respective indices.
- **Create a Matrix of Zeros:**
  - Initialize a matrix `A` of size M×NM \times NM×N filled with zeros. Each row represents a product, and each column represents an ingredient. This matrix will be populated to indicate the presence (1) or absence (0) of each ingredient in each product.

```
# Get the total number of products in the moisturizers_dry DataFrame

M = len(moisturizers_dry)

# Get the total number of ingredients in the ingredient_idx dictionary

N = len(ingredient_idx)

# Create a matrix of zeros with size MxN

A = np.zeros((M, N))
```

3.7 Oh-encoding and T-SNE :

```
def oh_encoder(tokens):
    # Initialize a matrix of zeros with width N (i.e., the same width as matrix A)
    x = np.zeros(N)

    # Loop through each ingredient in the tokens
    for ingredient in tokens:
        # Get the index for each ingredient
        idx = ingredient_idx.get(ingredient)

        # Put 1 at the corresponding indices
        if idx is not None:
            x[idx] = 1

    # Return the matrix x
    return x
```

```
i = 0
                # Loop through each list of tokens in the corpus
                for tokens in corpus:
                    # Apply oh_encoder() to get a one-hot encoded matrix for each list of tokens
A[i, :] = oh_encoder(tokens)

                    # Increment i by 1
i += 1
from sklearn.manifold import TSNE

                # Create a TSNE instance with n_components = 2, learning_rate = 200, and
                random_state = 42
model = TSNE(n_components=2, learning_rate=200, random_state=42)
                # Apply the fit_transform() method of model to the matrix A
tsne_features = model.fit_transform(A)
                # Assign the first column of tsne_features to moisturizers_dry['X']
moisturizers_dry.loc[:, 'X'] = tsne_features[:, 0]
moisturizers_dry.loc[:, 'Y'] = tsne_features[:, 1]
```

Initialize a Matrix of Zeros:

x = np.zeros(N) initializes a vector of zeros with the same width as the number of unique ingredients (N).

Loop through Ingredients:

For each ingredient in the token list, get its index using ingredient_idx.get(ingredient).

Set the value at the corresponding index in x to 1.

Return the Matrix:

Return the one-hot encoded vector x.

Integrate oh_encoder Function:

Loop through each token list in corpus and use oh_encoder to populate the document-term matrix A

Initialize Index Variable:

i = 0 initializes the index variable to keep track of the row in matrix A.

Loop Through Corpus:

For each list of tokens in corpus, apply the oh_encoder function to get the one-hot encoded vector.

Assign the one-hot encoded vector to the corresponding row in matrix A.

Increment the index variable i by 1 after updating each row.

Create t-SNE Instance:

TSNE(n_components=2, learning_rate=200, random_state=42) initializes a t-SNE instance with 2 components, a learning rate of 200, and a fixed random seed for reproducibility.

Apply t-SNE:

model.fit_transform(A) reduces the dimensions of matrix A and assigns the result to tsne_features.

Assign t-SNE Features to DataFrame Columns:

moisturizers_dry['X'] = tsne_features[:, 0] assigns the first dimension to the 'X' column.
moisturizers_dry['Y'] = tsne_features[:, 1] assigns the second dimension to the 'Y' column.

3.8 Visualization :

Create figure:

figure(x_axis_label='T-SNE 1', y_axis_label='T-SNE 2', width=500, height=400) initializes a Bokeh plot with specified axis labels and dimensions.

Add Circle Renderer:

plot.circle(x='X', y='Y', source=source, size=10, color='#FF7373', alpha=0.8) adds circles to the plot using the X and Y columns from the ColumnDataSource. The size, color, and alpha parameters customize the appearance of the circles.

```
from bokeh.io import show, output_notebook
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource

# Enable Bokeh output in the notebook
output_notebook()

# Create a ColumnDataSource with the moisturizers_dry DataFrame
source = ColumnDataSource(moisturizers_dry)
# Create a Bokeh figure
plot = figure(x_axis_label='T-SNE 1',
          y_axis_label='T-SNE 2',
          width=500, height=400)

# Add a circle renderer
plot.circle(x='X',
        y='Y',
        source=source,
        size=10,  color='#FF7373', alpha=0.8)
show(plot, notebook_handle=True)
```

Create HoverTool Object:

- HoverTool(tooltips=[...]) creates a hover tool with the specified tooltips.
- Tooltips are displayed when you hover over a point on the plot. They show detailed information for each data point.

Add HoverTool to Plot:

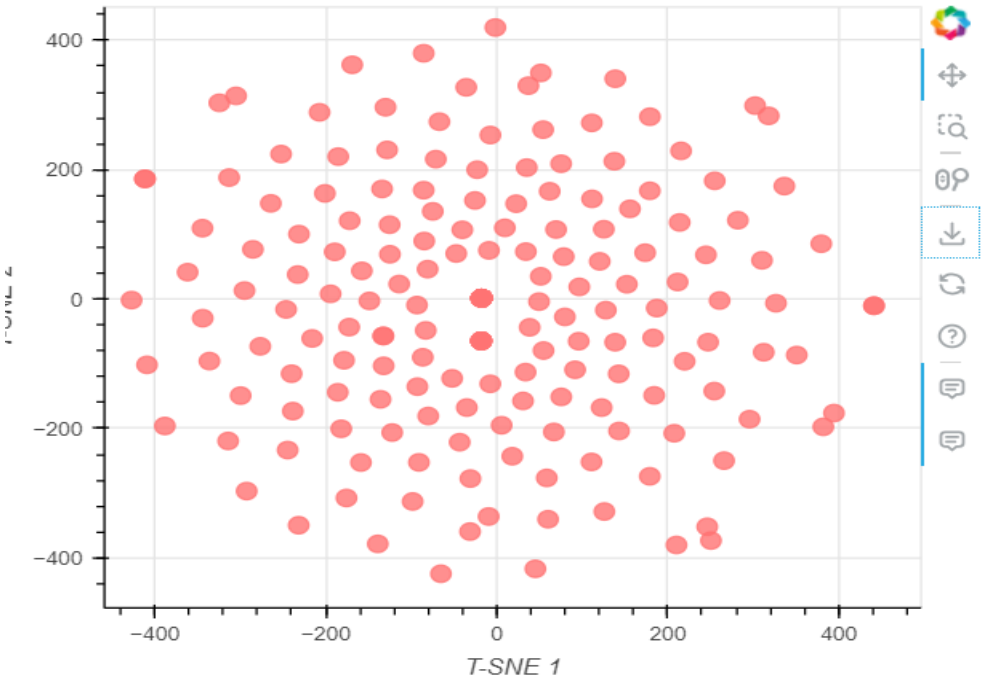
- plot.add_tools(hover) adds the hover tool to the plot.

Show the Plot: show(plot, notebook_handle=True) renders the plot with the hover tool in a Jupyter notebook or IPython environment.

- **show(plot, notebook_handle=True)**: This function call renders the plot in the Jupyter notebook or IPython environment. The notebook_handle=True argument ensures that the plot is displayed correctly in a Jupyter notebook.

```
from bokeh.models import HoverTool

# Create a HoverTool object with tooltips
hover = HoverTool(tooltips=[
    ('Item', '@Name'),
    ('Brand', '@Brand'),
    ('Price', '$@Price'),
    ('Rank', '@Rank')
])

# Add the HoverTool to the plot
plot.add_tools(hover)

# Show the plot with the hover tool
show(plot, notebook_handle=True)

# Display the plot with the hover tool
show(plot, notebook_handle=True)
```

4 . SCREENSHOTS



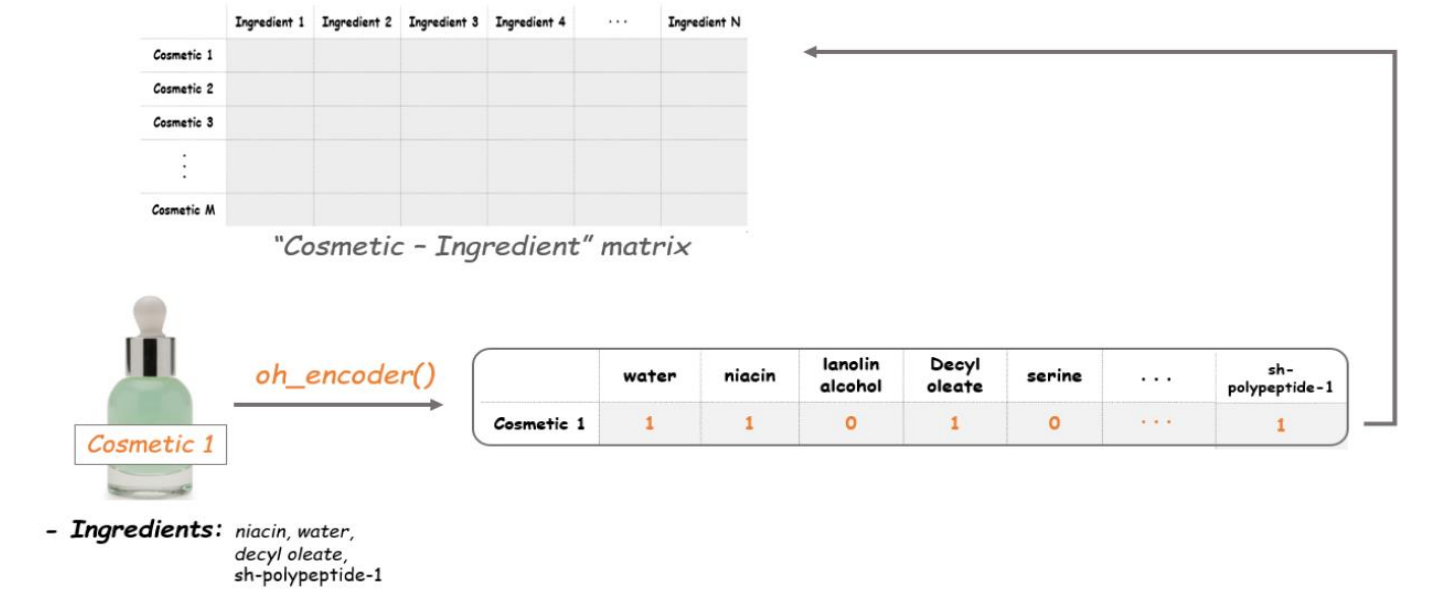
The scatter plot provides a visual representation of the similarity between cosmetic products based on their chemical components. The plot is created using t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the high-dimensional ingredient data into a two-dimensional space. Each point in the scatter plot represents a cosmetic product, and the position of these points reflects the similarity in their ingredient composition.

Key Features of the Scatter Plot:

1. **Axes Labels:**
   - **X-Axis (T-SNE 1):** Represents the first dimension of the reduced feature space obtained from t-SNE. It captures one aspect of the similarity between products.
   - **Y-Axis (T-SNE 2):** Represents the second dimension of the reduced feature space. This captures another aspect of the product similarity.

2. **Data Points:**
    - Each data point corresponds to a cosmetic product in the dataset. Products with similar ingredient profiles are positioned closer together, while those with dissimilar ingredients are placed farther apart.
3. **Interactive Features:**
    - **Hover Tool:** Hovering over a data point reveals detailed information about the product, including its name, brand, price, and rank. This feature allows users to gain insights into individual products without cluttering the plot.
4. **Visual Encoding:**
    - **Color and Size (if applicable):** Depending on the implementation, data points may be colored or sized according to specific attributes (e.g., product categories or price ranges) to provide additional context and aid in visual differentiation.
5. **Insights:**
    - The scatter plot helps users visually identify clusters or groups of similar products, making it easier to compare and find products with similar chemical components. This visualization supports the recommendation system by highlighting relationships between different products based on their ingredients.

Overall, the scatter plot serves as a powerful tool for understanding product similarities in a visually intuitive manner, enhancing the user's ability to make informed decisions about cosmetic products.



CONCLUSION

This project successfully demonstrated the application of data science techniques to enhance consumer decision-making in the cosmetics industry. By leveraging word embedding and t-Distributed Stochastic Neighbor Embedding (t-SNE), we transformed complex ingredient data into actionable insights, enabling consumers with sensitive skin to make informed choices.

The key accomplishments of the project include:

1. **Data Preparation and Processing:**
    - We meticulously collected, cleaned, and filtered data on 1,472 cosmetic products, focusing on ingredients and their suitability for dry skin. The resulting dataset was organized and ready for detailed analysis.
2. **Document-Term Matrix Creation:**
    - A comprehensive document-term matrix was constructed to represent the presence of ingredients across various products. This matrix served as the foundation for further analysis and visualization.
3. **Dimensionality Reduction and Visualization:**
    - Through t-SNE, we effectively reduced the dimensionality of the data, creating a two-dimensional representation that revealed the similarities and differences among products. The interactive scatter plot developed using Bokeh allowed users to explore these similarities intuitively.
4. **Recommendation System Development:**
    - A content-based recommendation system was designed to suggest products based on their chemical components. This system is particularly beneficial for consumers with specific skin concerns, offering personalized product recommendations.
5. **User-Friendly Tools:**
    - The interactive visualization dashboard, equipped with hover tools, provides a user-friendly interface for exploring product data and making comparisons. This enhances the decision-making process by providing detailed product information at a glance.

The project's outcomes are poised to benefit both consumers and industry stakeholders. Consumers gain access to a tailored shopping experience that aligns with their skin needs, while cosmetic companies receive valuable insights into ingredient trends and product formulations. This project not only addresses the challenge of selecting suitable cosmetic products but also sets a precedent for future applications of data science in the consumer goods sector.

In summary, by integrating data science techniques with cosmetic product analysis, this project has created a meaningful and practical solution for navigating the complex landscape of cosmetic ingredients, ultimately leading to more informed and confident consumer choices.

FUTURE SCOPE

The project opens several avenues for future development and enhancement, reflecting the dynamic nature of data science and its applications in the cosmetics industry. Here are potential areas for further exploration and improvement:

1. **Expansion to Other Product Categories:**
    - **Scope:** Extend the recommendation system to include other categories of personal care products such as skincare, haircare, and body care.
    - **Benefit:** Broader applicability and a more comprehensive tool for users seeking recommendations across different product types.
2. **Integration of User Feedback:**
    - **Scope:** Incorporate user reviews and ratings into the recommendation system to enhance accuracy and personalization.
    - **Benefit:** Provides a more nuanced recommendation based on real user experiences and feedback, improving the relevance of suggestions.
3. **Enhanced Ingredient Analysis:**
    - **Scope:** Develop advanced analysis techniques for understanding the impact of ingredient interactions and formulations.
    - **Benefit:** Offers deeper insights into how different ingredient combinations affect product efficacy and skin health.
4. **Real-Time Data Updates:**

- **Scope:** Implement mechanisms for real-time data collection and updates from cosmetic product databases and e-commerce platforms.
- **Benefit:** Ensures the recommendation system remains current with the latest products and ingredient information.

5. **Cross-Platform Integration:**
   - **Scope:** Integrate the recommendation system with mobile apps and e-commerce websites for seamless user experiences.
   - **Benefit:** Provides consumers with on-the-go access to personalized recommendations and product information.
6. **Advanced Machine Learning Models:**
   - **Scope:** Explore and apply more sophisticated machine learning models, such as deep learning techniques, to enhance the recommendation system's performance.
   - **Benefit:** Improves the accuracy and robustness of product recommendations by leveraging more complex patterns in the data.
7. **Personalized Skincare Routines:**
   - **Scope:** Develop algorithms to recommend complete skincare routines based on individual skin types, concerns, and preferences.
   - **Benefit:** Provides users with tailored product combinations and routines, enhancing their overall skincare regimen.
8. **Global and Cultural Considerations:**
   - **Scope:** Adapt the recommendation system to accommodate global cosmetic preferences and regional ingredient regulations.
   - **Benefit:** Makes the tool more inclusive and relevant to diverse markets and consumer needs worldwide.
9. **User Education and Awareness:**
   - **Scope:** Create educational content and tools within the dashboard to inform users about ingredient safety, efficacy, and proper usage.
   - **Benefit:** Empowers users with knowledge about cosmetic ingredients, promoting informed and conscious product choices.
10. **Ethical and Sustainable Practices:**
    - **Scope:** Integrate filters for ethical and sustainable product choices, such as cruelty-free, vegan, or eco-friendly products.

III. REFERENCES

- **Pandas Documentation**
  - McKinney, W. (2022). *pandas: powerful Python data analysis toolkit*. Retrieved from https://pandas.pydata.org/pandas-docs/stable/
- **NumPy Documentation**
  - Harris, C.R., Millman, K.J., & van der Walt, S.J. (2020). *Array programming with NumPy*. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2
- **Scikit-Learn Documentation**
  - Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825-2830. Retrieved from https://scikit-learn.org/stable/
- **t-SNE:**
  - Van der Maaten, L., & Hinton, G. (2008). *Visualizing data using t-SNE*. Journal of Machine Learning Research, 9, 2579-2605. Retrieved from http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
- **Bokeh Documentation**
  - Bokeh Development Team. (2023). *Bokeh: Python interactive visualization library*. Retrieved from https://docs.bokeh.org/en/latest/
- **Web Scraping with BeautifulSoup**
  - Richardson, L. (2022). *Beautiful Soup Documentation*. Retrieved from https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- **Scrapy Documentation**
  - Scrapy Developers. (2023). *Scrapy Documentation*. Retrieved from https://docs.scrapy.org/en/latest/
- **Data Science in the Cosmetics Industry**
- Choi, J., & Lee, J. (2018). *Application of data science in cosmetics product development*. Journal of Business Research, 87, 20-28. https://doi.org/10.1016/j.jbusres.2018.02.020