UNCOVERING ALLERGENS IN PERFUMES USING ANNOTATION AND MACHINE LEARNING

Submitted by:

Mukadas Akhtar(23046538)

Husnain Sultan (23042677)

Taha Mousavi (17012929)

OIM7507:Artificial Intelligence and Data Science

Dr. Emad Mohamed

University of Bradford

29th April 2024

**Introduction**

In our project, we developed a machine learning model using annotation.It is designed to assist consumers with allergies in selecting perfumes that are safe for them to use. Recognizing the challenges faced by individuals sensitive to specific ingredients commonly found in perfumes, our initiative focuses on identification of allergenic substances in various fragrance products, as no website offers a feature to help customers select perfumes based on their specific allergies. We began by targeting a known allergen, Eugenol, using data we scraped from two online retailers, to test and refine our model. Our goal is to expand this approach to include a comprehensive list of allergens such as Limonene, Linalool, and Benzyl Salicylate, thereby enhancing our model's utility and accuracy. This project not only aims to improve consumer safety but also enhances the shopping experience by providing personalized fragrance recommendations based on individual health needs.

### **Methodology**

### *Data Acquisition and Extraction Process*

We collected data on perfume names, prices, and the ingredients used. After compiling this information from the selected websites, we applied our machine learning model to detect the presence or absence of allergenic ingredients. The model also helped categorize perfumes by price, facilitating easier and more informed decision-making for consumers with allergies.

We sourced our dataset from two online retailers offering a diverse array of products, from medicines to perfumes, makeup, childcare items, and electronics. Initially, we opted for Boots.com due to its extensive perfume selection, which includes 849 options ranging from luxury brands to more affordable choices. We specifically selected this website because it provided the ingredients of the perfumes, a feature lacking on many other websites.

We compiled the URLs of all 849 perfumes and executed our script, which extracted the price, ingredients, and name of each perfume. The script was designed to automatically save this data into a CSV file. Through this process, we efficiently gathered comprehensive information on a wide variety of perfumes available at Boots.com.

This Python script started by Importing the necessary Libraries. The following libraries were imported: `requests` library for handling HTTP requests, `BeautifulSoup` from `bs4` for parsing HTML, and `csv` for handling CSV file operations. It also imports `drive` from `google.colab` for Google Drive access. Then we Mounted the Google Drive by using the `drive.mount('/content/drive')` command, allowing it to read from and write to files stored there. This is essential for saving the scraped data directly to a Google Drive folder.Next the URLs containing product information to be scraped were defined. We pasted all the perfume URLs in this function, enclosed in ‘’ and separated by commas. Then we specified the output CSV File Path where we wanted to save our extracted data in the Google Drive.Next, Headers were set for HTTP Requests, using the `headers` dictionary to simulate a web browser request.

Next, we set up our CSV file, defining the column titles as 'product\_name', 'ingredients', and 'price'. After setting up the basics, the code Sent HTTP Requests for each URL in the list and if the request was successful, it analyzed the HTML content using BeautifulSoup and extracted the product name, price, and ingredients. One necessary thing to consider while writing the code was that product name on the website was stored in a `h1` tag, price was in `div` tag and ingredients were in `h3` tag. We found this out by inspecting the Developer tools on the website by the ‘inspect’ feature and finding the right HTML element for each data point we wanted to extract.

Lastly, the script we made writes the extracted data to the CSV file using the writer. Additionally, the script printed the extracted information to the console for immediate viewing, as the code was running.

By the end of its execution, the script created a CSV file in Google Drive containing the extracted data from the specified webpages. This file was then used for further analysis in our report.

After executing the script and generating the CSV file, we discovered that the ingredients for 220 perfumes were not listed, resulting in the output “Ingredients not found” for these items. Consequently, we found an alternative source , Debenhams.com, which provided the ingredient details for the perfumes we were interested in. We then collected the URLs for these remaining 220 perfumes from the Debenhams website. To run our code on the Debenhams website ,we had to modify our original script, which was initially tailored for Boots.com.This was done to accommodate the different web structure of Debenhams.com. The product name was extracted from an ‘h2*’* tag.

The product price extraction was more complex, as it handled both minimum and maximum prices listed under a ‘div’ tag. The price was in this format to accommodate discounts or variable pricing based on quantity options. Moreover, the product description, which included the full name of the perfume,was extracted from an ‘h1’ tag. Once we ran the codes on the Boot and Debenhams website, we had data on 366 perfumes.

Here is a link to our code:[Web-scraping code.ipynb](https://colab.research.google.com/drive/15C36G5Kp0ZjNf4QD8MLFelEj275fwQlo?usp=sharing)

*Data cleaning and preparation*

On the Boots website, we identified 849 perfume listings. However, we excluded several pages from our analysis due to duplication, differing only in terms of product volume (e.g., 50ml versus 100ml options). After using our scraping script on the Boots site, we successfully extracted data for 700 perfumes, which we compiled into a CSV file. Notably, 220 of these entries lacked ingredient information, which led us to segregate these records into a separate Excel sheet for further attention.

Subsequently, we sourced the missing ingredient data for these 220 perfumes from Debenhams. When we applied our scraping script to these webpages, we managed to retrieve information for only 37 entries that were previously incomplete. The CSV file displayed the prices for these entries in a range format with a letter written with it, such as "A£50-A£150." This necessitated a formatting adjustment to standardize the price data across our dataset.

For the Debenhams website CSV file, the perfume names and brand names were initially recorded in two separate columns. We merged these columns to present the full name of each perfume, aligning with the format used in the Boots CSV file. This standardization facilitated the seamless merging of the two CSV files, encompassing both the original and the previously missing data.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **Findings and Analysis**

*Annotation Process*

The annotation process we used for our project was indirect or distant annotation. A breakdown of the process is as follows:

Indirect Annotation Approach

* Data Extraction:

We extracted perfumes data (product name, price, ingredients) from Boots and Debenhams website.

* Annotation Context:

The annotation was indirect, relying on the assumption that the listed ingredients on the website correspond to the actual chemical composition of the perfumes.

* Limitations:

This approach assumes that the website's ingredient lists are accurate and comprehensive, which may not always be the case.

Annotation Guidelines

* Web Source Integrity:

Annotation guidelines implicitly relied on the accuracy and completeness of the ingredient information provided on the websites.

* Handling Variability:

Guidelines included rules for interpreting ingredient formats (e.g., chemical names, common names) and handling variations in presentation.

For our project statement regarding annotation guidelines, the annotations used in this process are primarily indirect annotations derived from publicly available web data. The guidelines applied in this context focused on:

* Data Source Reliability:

Ensuring that the selected websites provided accurate and up-to-date information on perfume ingredients.

* Standardized Extraction:

Establishing methods to extract and interpret ingredient information consistently across different websites.

* Verification Protocol:

Implementing checks to validate the extracted data against trusted sources or chemical databases to ensure accuracy.

This approach allowed us to compile a dataset containing perfume-related information, including ingredients and prices, sourced from the web. However, it is important to note the limitations in this method, such as variations in data quality across different websites and the need for ongoing verification and validation of the extracted information for analytical purposes.

*Feature description of the data set.*

The dataset we extracted contains three primary features essential for analyzing perfume formulations. The first feature is Product Name (product\_name). The product\_name feature includes textual descriptions of each perfume, serving as a unique identifier for individual products. This can be useful for brand-specific analysis of what ingredients each brand tends to use. An example of this feature is "Jean Paul Gaultier Scandal Absolu 50ml".

The second feature is the ingredients (ingredients). The ingredients feature is crucial for the project as it lists the components used in each perfume. This data is presented as a comma-separated list, which includes compounds such as "AQUA (WATER)" and "ALCOHOL DENAT." Analyzing this data will allow for the extraction and categorization of ingredients, offering insights into common ingredient pairings, frequency of usage, and emerging trends in perfume formulations. We used the ingredients to study frequency, common pairings, and trends in perfume formulations.

The last feature is Price (price), which records the retail price of each perfume, denoted numerically with a preceding currency symbol of £. This information can be used to explore pricing strategies so see any correlation in presence of specific ingredients with price points, and understand market segmentation.

Together, these features provide a good foundation for in-depth analysis of perfume compositions and market dynamics. These features will help us analyze whether certain ingredients are associated with higher-priced products, possibly indicating premium components.

*Machine Learning Model*

By implementing and evaluating a basic machine learning model using the annotated data (extracted from web sources), we showcased the practical applications and benefits of the annotation process:

* Automated Data Extraction: The model demonstrates how annotations (web scraping) can facilitate automated data collection from diverse sources.
* Task Automation: Showcases how machine learning can leverage annotated data to perform tasks (e.g., product classification, ingredient prediction) that would otherwise be time-consuming or labor-intensive. Our machine learning code searches for perfumes that contain and do not contain a specified allergen (or ingredient) within their ingredients list. It sorts the results of perfumes by their price in descending order and prints out the number of perfumes that match the search criteria. For example, the script sets a specific search term ('Eugenol') and executes both functions to display perfumes that contain and do not contain 'Eugenol'. It prints results for both searches to the console.
* Proof of Concept: Provides tangible evidence of how annotations contribute to building useful models and applications in real-world scenarios. Our machine learning model will be used to identify allergen ingredients in perfumes and see if they are associated with higher-priced products. This script is useful for anyone needing to filter and analyze perfumes based on specific ingredients, particularly for users with allergies. It efficiently identifies suitable or unsuitable perfumes This approach can serve as a foundation for further exploration and refinement, highlighting the potential of annotation techniques in generating valuable datasets for machine learning applications.

Here is the basic machine learning code for the proof of concept that the annotation works:

[Machine Learning Final](https://colab.research.google.com/drive/1kOliKyAwVOdOVhf372oB9MrqOWsP4jhz?usp=sharing)

**Discussion of Responsible AI Practices**

To discuss how our project aligns with the principles of responsible artificial intelligence (AI), we can consider several key aspects that were highlighted during our tutorials. These aspects include transparency, safety and security, fairness, and accountability. Here’s how our project upholds each of these principles:

Our project ensures transparency by thoroughly documenting the data acquisition and processing methods. We have openly detailed how we scrape data from the Boots and Debenhams websites, specifying which elements are extracted and how they are handled. This transparency extends to our code, which is maintained with clear annotations and is available for review. We have also clearly stated how we are going to be using this data to help us classify perfumes based on their allergen properties.

Moreover, we tried to ensure that our data scraping methods were safe and secure. We tried to ensure that our script did not load on the website we were scraping. For this, our script processes URLs one at a time instead of requesting data from multiple URLs at the same time. Sequential processing helped us in reducing the load on the server, as each request is made only after the previous one has completed. This method is considered responsible because it shows our consideration for the efficiency of the web server. Overloading a server can lead to a variety of problems, including decreased performance and accessibility issues for other users. Additionally, we used the “/robots.txt” at the end of the URLs of the websites we wanted to scrape, just to check which areas of the site we could use. The robots.txt file is a crucial element in the context of web scraping and responsible practices. Our project adheres to privacy guidelines by only scraping publicly available data that does not involve personal user information.

Furthermore, our team remains accountable for the AI system’s performance and the outcomes of our project. We have implemented checks and balances to monitor the system’s accuracy and reliability, adjusting our models in response to any errors or inefficiencies identified. We are prepared to take corrective actions if the system produces errors or if unexpected outcomes occur, ensuring that our project remains ethical and effective in its application.

To promote fairness, our project avoids biased data selection by including a wide range of products from different brands and price points. We carefully selected diverse sources (e.g., Boots and Debenhams) to ensure a broad and representative dataset. When we were selecting perfumes, we chose different high end and cheaper brands, price ranges, and target genders so that there might be no bias in the data collected.

Lastly, we are accountable for the performance of the system and the project's results. Our models can be modified if there are any faults or inefficiencies found during evaluation. We are ready to make the necessary adjustments to ensure that our project stays morally and practically sound in its implementation in the event that the system generates faults or unexpected results.

**Conclusion**

In our project, we initially encountered challenges due to a lack of understanding that each website we scraped required different HTML elements to be targeted, such as `div`, `h1`, and `h2`. We mistakenly ran the same code across multiple websites, which led to errors because of the differing structures. To improve, we recognize the need to adapt our scraping scripts to handle the unique elements of each site more effectively.

Furthermore, we initially applied our machine learning algorithm to filter perfumes based on the presence of one specific allergen: Eugenol. Moving forward, we aim to expand our research to include a broader range of allergenic ingredients commonly found in perfumes, such as Limonene, Linalool, and Benzyl Salicylate. This will enable us to refine our filtering process, allowing consumers to select perfumes that accommodate their specific allergy sensitivities more effectively.

Additionally, although we aimed to expand our dataset by including more perfumes, we found it challenging to gather comprehensive ingredient information, as many websites do not disclose this detail. Looking forward, we plan to enhance our machine learning model and deploy it on websites to assist individuals with allergies. This development will enable users to select perfumes that are safe for them based on their ingredient profiles, thus making our project more impactful and user-friendly.