DSO 560 Group Project Part I

Due 11:59pm PST, April 30th

All available data will be uploaded to the **threadtogether** PostgreSQL database, available at

Host: **threadtogether.ychennay.com**
User**: dso560_student**
Password: **(in class)**
Port: **5432**

If your group would like to use the database to create temporary tables, please message me and I will create a group-specific user/password for your team to log in as.

Deliverables:

- **The same full_data table, with one column** for each of the N categories your group selected, indicating if the product matches a tag.
- **Github repository** (with instructor ychennay added as collaborator containing your group's product attribution tool/app code base).

Database Tables:

- **full_data**: contains all possible inventory items that **ThreadTogether** would potentially offer as products
- **tagged_product_attributes**: manually labelled attributes of the products by SME (subject matter experts) – can be joined to **full_data** via **product_id**
- **women's clothing reviews:** aggregated open source dataset of women's clothing reviews
- **categories (available in S3 bucket)**

Objectives:

1. **Attribution of web data** – TT will provide a dataset previously extracted from various retail websites, we would like to organize and extract meaningful product information from the web data.
2. **Attribution tool/app** – As a supplement to extracting features from current datasets, we propose the development of a text parsing app that would provide a repeatable solution and will require minimal human intervention for future datasets from retailers. The input to this app would be product descriptions, tags, and other metadata.

Recommended steps:

- **Decide which N – 2 categories your group would like to focus on**. You are required to focus on the proprietary attributes – style and occasion. Beyond that, your group should pick N - 2 other groups to analyze.

  Examples of categories:

    - Embellishments
    - Category
    - Prints
    - Material
- Join the **tagged_product_attributes** table with **full_data** and investigate the details, descriptions, and tags used for each category – your goal is to get a sense for the business logic and rules used in tagging a certain product category.
- **Build a model that will takes as input:**
    - product description (if any)
    - product name
    - product details (if any)
    - brand

And outputs the predicted attributes of this product. For example, if the category you are using is fit,

INPUT: • description: **Blush linen Button fastenings along front 100% linen; lining: 100% cotton Dry clean Designer color: Shell Imported** • brand: **Zimmermann** • brand_category: **Clothing / Jumpsuits / Full Length** **The actual clothing's product URL is here.**
OUTPUT: • **predicted fit: RELAXED**

Scoring Rubric	Points Available			
	1pt	**2pts**	**3pts**	**4pts**
The model passes 3 of the 4 test products by successfully returning the appropriate category attribute	No results return any appropriate matches	1-2 of the 4 tests return an appropriate match	3 of the 4 tests return an appropriate match	4 of the 4 tests return an appropriate match
Model shows some evidence of incorporating domain context (women's retail) – either via SME rules, or via embeddings. For example, a product in the pumps category will likely co-occur with open-toed / close-toed, pointed-toe.	No evidence of anything beyond a full string match	Basic cleaning/preprocessing evident, but the model itself relies entirely on simple regex/string match	Some basic business domain logic is encoded either via a word embedding scheme, or via preprocessing rules (ie. groupings)	Business domain logic is encoded either via a word embedding scheme, or via preprocessing rules (ie. groupings)
Code is documented with thought process easily visible to instructor/client to review	No (0 pts)		Yes (2 pts)	
Code is published to a Github repository with team members added as collaborators	No (0 pt)		Yes (1 pt)	
All group members have submitted 360 feedback reviews by deadline	No (0 pt)		Yes (1 pt)	