**DSC 540**

**DATA PREPARATION**

**BELLEVUE UNIVERSITY**

**WEEK 7 & 8: PROJECT MILESTONE [1, 2 & 3]**

**Project Objective:** Perform at least 5 data transformation and/or cleansing steps to your flat file data. The below examples are not required - they are just potential transformations you could do. If your data doesn't work for these scenarios, complete different transformations. You can do the same transformation multiple times if needed to clean your data. The goal is to clean a dataset at the end of the milestone.

* Replace Headers
* Format data into a more readable format
* Identify outliers and bad data
* Find duplicates
* Fix casing or inconsistent values
* Conduct Fuzzy Matching

Make sure you clearly label each transformation (Step #1, Step #2, etc.) in your code and

describe what it is doing in 1-2 sentences

**Project Description**

Many travelers now have access to fantastic, straightforward, and practical lodging from Airbnb. Similar to that, it has provided several people with a means of generating additional income by marketing their homes as places for travelers to stay. To complete the project milestone I have selected 3 data sources named as (**Listings**, **Calendar**, and **Reviews** ) to query and design the Airbnb API and websites to easily provide the data to end users and facilitate easy booking and querying from UI.

**Data Sources**

To complete the project milestone I have selected the AIRBNB project and downloaded datasets from Tableau Community. The size of data set meets the project requirements minimum of 1000 rows across all datasets and total of 30 columns across the 3 datasets selected. All the data sources are uploaded to GITHUB link mentioned below.

<https://github.com/santoshgithub100/DSC540_TermEndProject>

**Listings.csv** : Summary information and metrics for listings in San Francisco (good for visualizations).

**Reviews.csv**: Summary Review data and Listing ID (to facilitate time-based analytics and visualizations linked to a listing).

**Calendar.csv**: details of Calendar data to facilitate rooms availability and customer bookings.

Entities and their association with other entities.

1. The Listing entity has 1: 1 relationship with Calendar entity
2. The Review entity has 1:1 relationship with Calendar and Listing entity.
3. The ListId is a primary key in the Listing entity and Foreign Key in the Reviews and Calendar entity.

**The AIRBNB entities relationship**



**API**

* All functions return [promises](https://github.com/wbinnssmith/awesome-promises)
* Contains the transactions at each Airbnb bookings – contains a transaction ID and store ID. The API response data format is pre-parsed JSON
* All the API will have error reporting and data validation
* ***getPublicListingCalendar***: Public availability and price data on a listing, count is the duration in months
* ***getCalendar***: Private calendar data regarding your listings. Reservations, cancellations, prices, blocked days
* ***getListingInfo***: To provide public facing data on any listing
* ***getListingDetails***: To get the Airbnb listing search
* ***getListingInfoHost***: To get the private data on one of customer listings
* ***getReviews***: To provide the reviews

**Website**

Provides a list of affordable homes for travelers with great reviews, things available in neighborhood for easy room booking and Additionally, if a traveler wants to find the cheapest listing available but with certain features he can prefer like 'free parking' etc., Thus providing an excellent, easy, and convenient place to stay during travel.

**Ethical implications of data wrangling specific to the data source and the steps completed (Milestone - 3)**

While designing websites and API we should be mindful of the content we create. The websites should not promote hate speech, discrimination, or any form of harmful or offensive content. So, it is very important to create a website that contribute positively to the online community and promote inclusivity, diversity, and respect. Depending on the circumstances and situation, the data source, and the particular methodology taken, data wrangling may have different ethical ramifications. The type of data and the analysis's objectives determine the particular ethical ramifications of data wrangling. Maintaining the confidence and respect of the people and communities from whom the data is derived is just as important as adhering to rules and regulations when it comes to ethical data wrangling. Responsible data science includes ethical data management and analysis as a fundamental component. The following are few challenges related to data wrangling:

* Data Privacy and Consent: When working with data, it's important to ensure that you have the necessary permissions and consents to use the data. This is particularly critical when dealing with personal or sensitive information. Ethical concerns may arise if data is collected, shared, or used without the knowledge and consent of the individuals involved. Additionally managing sensitive or private data and adhering to privacy regulations (e.g., GDPR) is crucial. Anonymization and data protection techniques must be applied correctly
* Data Quality: Ensuring that the dataset is accurate, complete, and reliable is one of the primary challenges. Cleaning, validating, and curating data can be time-consuming
* Data Security: Security is a top concern for APIs. Protecting against unauthorized access, data breaches, and attacks such as SQL injections or cross-site scripting requires robust authentication and authorization mechanisms. Data wrangling often involves data transfer, storage, and sharing. It's crucial to take measures to protect the data from unauthorized access, breaches, or leaks. Data security breaches can have serious ethical and legal implications.
* Data Anonymization and De-identification: If the dataset contains personal or sensitive information, it's essential to anonymize or de-identify the data to protect the privacy of individuals. Failure to do so could result in ethical and legal issues.
* Bias and Fairness: Data wrangling can introduce bias if not done carefully. For example, data cleaning or sampling methods may inadvertently favor certain groups or perspectives. It's crucial to be aware of and address bias to ensure fairness in data analysis and decision-making.
* Scalability: APIs should handle increasing loads and user traffic efficiently. Designing for scalability often involves load balancing, caching, and considering potential bottlenecks in the system.