CIS550 Final Project: Oeda Platform

Huifang Ye, Joyce An-jie Wang, Zhihui Zhang, Zhuoran Hu

Project Overview

1 Introduction

- Target Problem
- Project Goal
- Application Functionality

Project goals and target problems

Create a user-friendly, intuitive Business Intelligence or Business Analytics tool for e-commerce:

- People's shopping preferences
- Generate graphs and charts
- Analytics for sales and marketing
- Geographic insights

Application Functionality

Display trends

Display trending/featured visual analysis

Search

Search product, order, or payment information using multiple different search keywords and filters

Create reports

Users can generate a variety of charts, widgets, pivot, summary or tabular views on sales and geographical data based on their search input

Export

Users can export queried results as .csv files to their local device

Visualization

Provides visualization for geographic distribution of customer/sellers/sales data and display Brazilian city demographics

Market Analysis

Users can search and generate sales results and product analysis based on Brazilian city demographics

2 Overview of Datasets

- Brazilian E-Commerce Dataset
- Brazilian Cities

Overview of Datasets

Brazilian E-Commerce Dataset by Olist:

- Item
- Product
- Customer
- Payment
- OrderInfo
- Category
- Seller
- Review

A Brazilian ecommerce public dataset of orders made at the Olist store. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil.

Overview of Datasets

Brazilian Cities
- This dataset is a compilation of several publicly available demographic information about Brazilian Municipalities.

- There are in total 79 fields for each city, which includes city, state, resident population, Human Development Index (HDI), number of Pay TV users, GDP, number of companies by industries, Walmart stores, and etc.
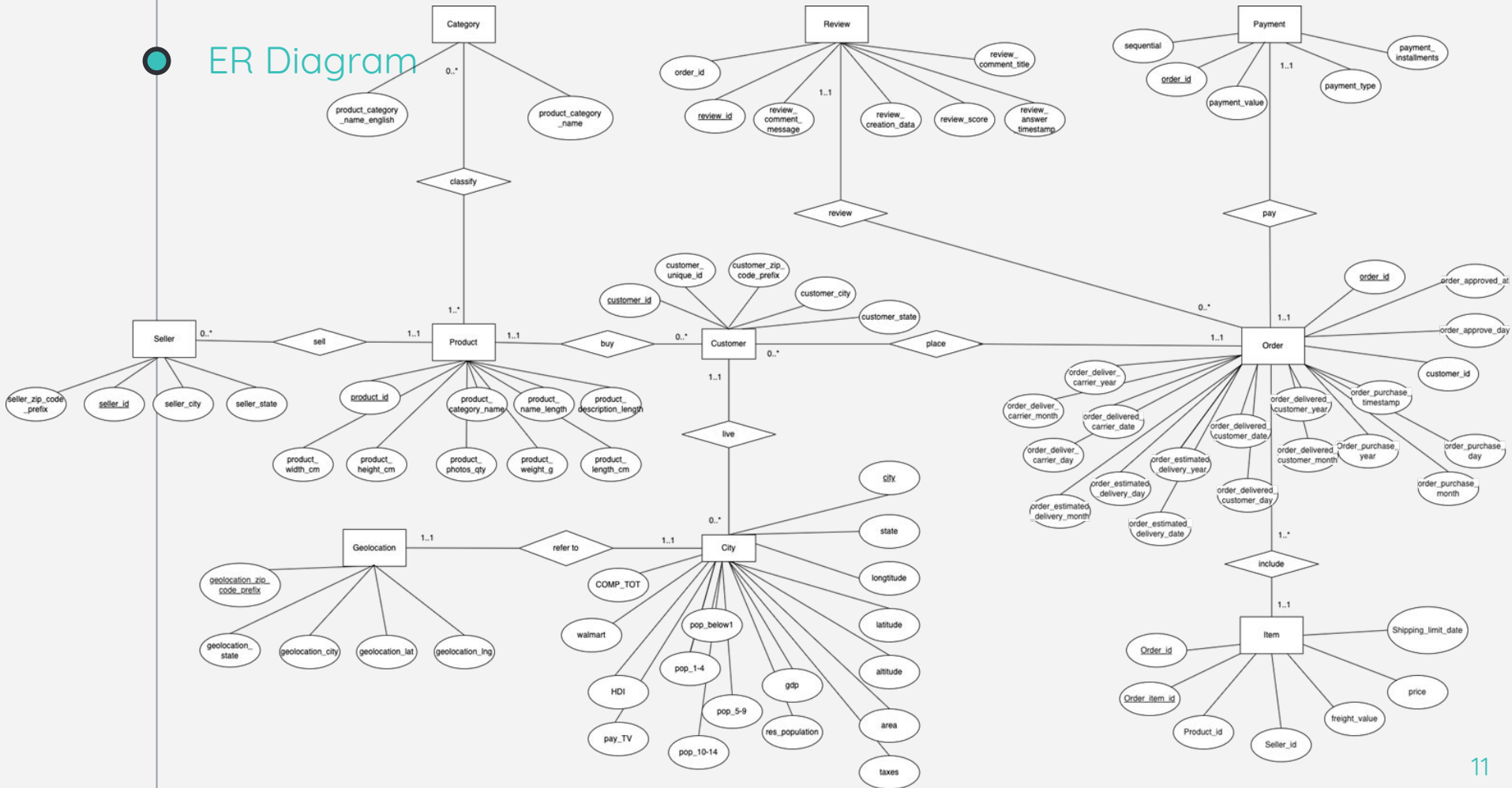
3 Data Cleaning

- Use Python (Pandas, PandaSQL) and Colab
- Use dropna(), DateTimeIndex, rename, and merge in pandas
- Use natural join, sort_values, nunique(), and groupby

4 Database

- Schema Design & ER Diagram
- Normal Form and Justification

ER Diagram

11

5 Complex Queries and Optimization

- Example of Complex Query
- Performance Evaluation
  - Runtime and Optimizations

Example: Complex Query

For each of the five cities, list the total number of Walmart stores, total number of orders (based on customer location) by year, total sales (based on product price) by year, and the most popular product category based on sales (in English) by year. The result is ordered by the number of Walmart stores.

```sql
WITH top_cities
     AS (SELECT city, walmart
         FROM City
         ORDER BY walmart DESC
         LIMIT 5),
    orders_products
     AS (SELECT O.order_id,
                O.order_deliver_customer_year AS year,
                C.customer_id,
                C.customer_city AS city,
                P.product_id,
                P.product_category_name,
                I.price
         FROM OrderInfo O
         JOIN Item I ON O.order_id = I.order_id
         JOIN Product P ON I.product_id = P.product_id
         JOIN Customer C ON O.customer_id = C.customer_id),
    total_orders
     AS (SELECT city, year, COUNT(DISTINCT order_id) AS count
         FROM orders_products
         WHERE city IN (SELECT city FROM top_cities)
         GROUP BY city, year),
    total_sales
     AS (SELECT city, year, SUM(price) AS sales
         FROM orders_products
         WHERE city IN (SELECT city FROM top_cities)
         GROUP BY city, year),
    top_product
     AS (SELECT city, year, c.product_category_name_english, SUM(price) AS sales
         FROM orders_products op
         JOIN Category c ON c.product_category_name = op.product_category_name
         WHERE city IN (SELECT city FROM top_cities)
         GROUP BY city, year, c.product_category_name_english)
SELECT tc.city AS City,
       tc.walmart AS 'Number of Walmart Stores',
       tto.year AS Year,
       tto.count AS 'Number of Orders',
       ts.sales AS Sales,
       tp.product_category_name_english AS 'Top Selling Product'
FROM top_cities tc
NATURAL JOIN total_orders tto
NATURAL JOIN total_sales ts
JOIN top_product tp ON tc.city = tp.city
WHERE tto.year = tp.year AND tp.sales >= ALL (SELECT sales
                                              FROM top_product tp
                                              WHERE tp.city = tc.city
                                              AND tp.year = tto.year)
ORDER BY tc.walmart DESC, tto.year;
```

Performance Evaluation

Query Description	Original	Optimized
Show the average rating and the number of review for each product category (type 1)	769 ms	214 ms
Show the average rating and the number of review for each product category (type 2)	577 ms	225 ms
Query the differences of total orders between 2016, 2017, and 2018 for each product category	812 ms	409 ms
Create a sales report for the top 5 cities with the most Walmart stores	934 ms	551 ms
Retrieve the top 10 products with highest average review scores	148 ms	86 ms
Get the differences in total, average, max, and min payment values by credit card users and bank ticket users from each state	442 ms	280 ms

6 Technical Challenges

- Set up Node.js in different systems
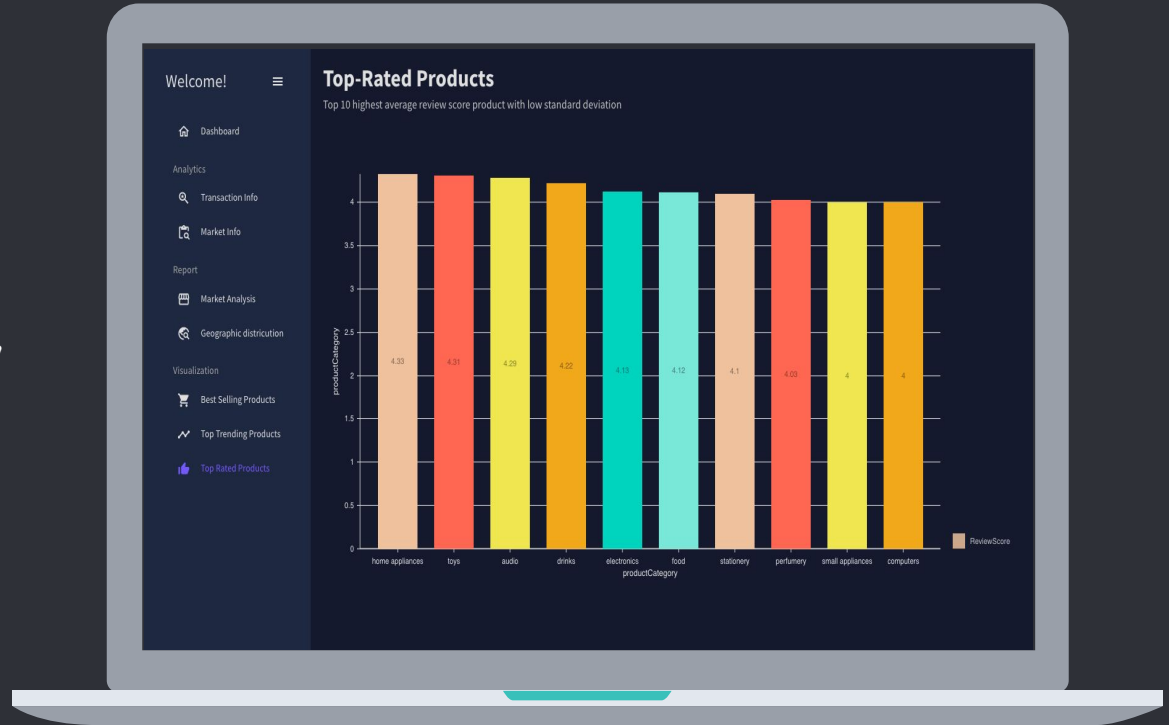- Read Timeout Error
- Connecting data to the frontend

7 Extra Credit Features

- Code Coverage (unit testing > 80%)
- Integration with APIs and colab

Live Demo

Website Overview

Thank you!

Questions and Comments