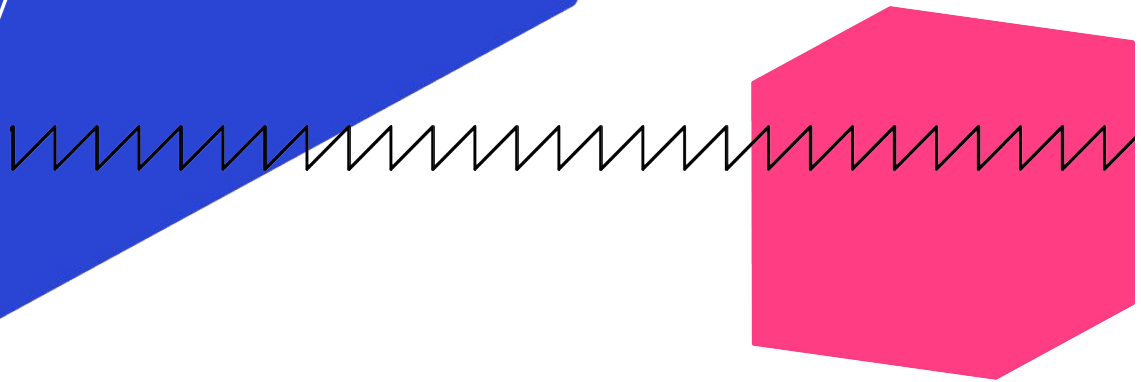
DSC Datathon 2022

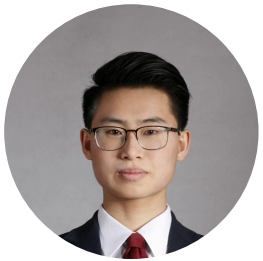
Peak.ai x Olist

Son, Sons & Company



Your Peak.ai Team: **Son, Sons & Company**

Sunny Son
Machine Learning Engineer



Shane Sun
Data Scientist

Morgan Xu
Software Engineer

Sunny Yang
Data Scientist

Problem Statement

Problem Statement | Data Preprocessing | Data Visualization | Data Analysis | Key Takeaways

3 DATA SCIENTISTS

2 DAYS AT NYU DSC

1 CHALLENGE

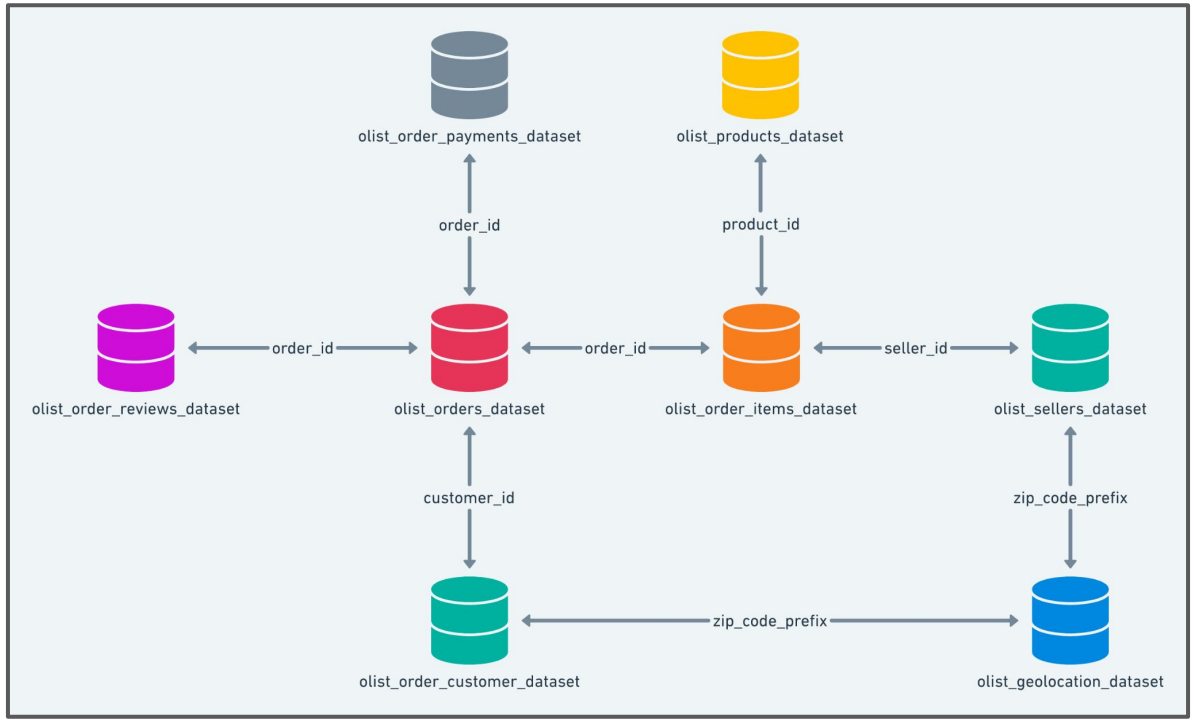
> Create a personalized product recommender for customers based on their purchase history

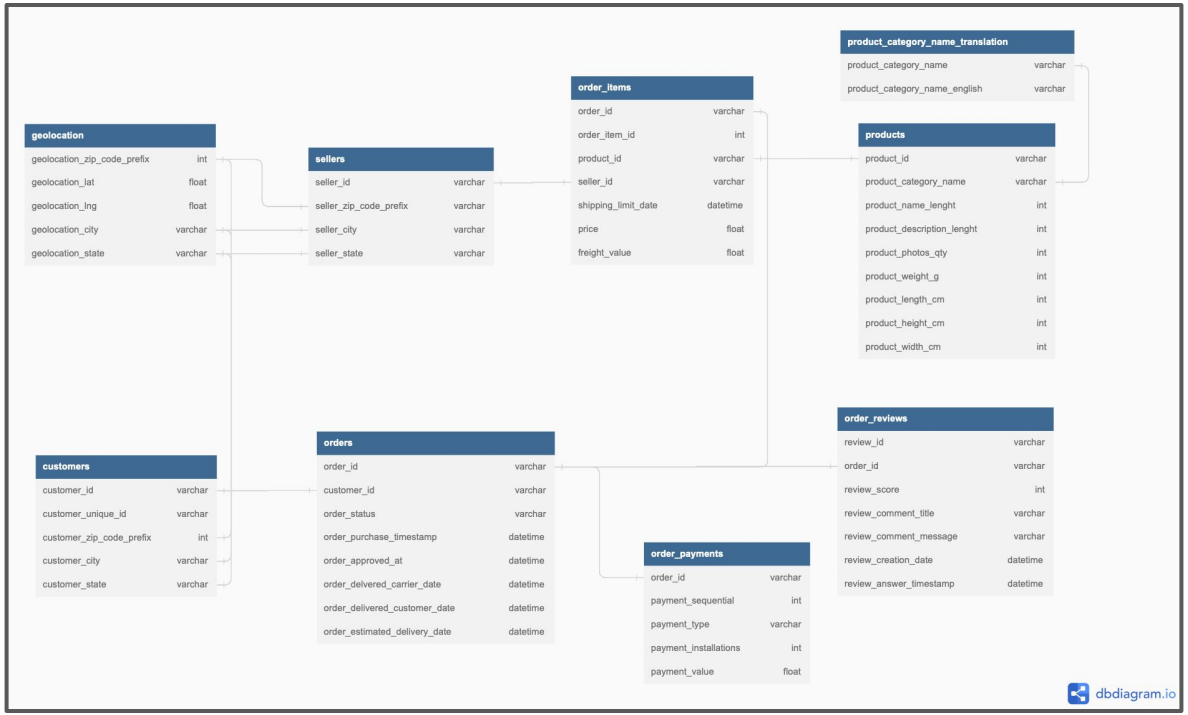
Olist Store

E-commerce solutions provider for small and medium enterprises

- Olist's primary value-add is a proprietary enterprise resource planning (ERP) software to manage payments, track order fulfillment, and perform basic accounting
- Store owners can build their own website with the assistance of Olist's integrations or sell directly on Olist's Marketplace

Olist has hired Peak to create a Product Recommendation System





Data Preprocessing

Problem Statement | Data Preprocessing | Data Visualization | Data Analysis | Key Takeaways

PREPARING OLIST'S DATA IN PYTHON

Packages Used: Pandas, Scikit Learn

Data Cleaning

We removed all missing, NaN, and corrupted data values across all eight tables.

Dataset Merging

We merged all the tables into one master table of 113,425 rows and 40 columns.

Data Imputing

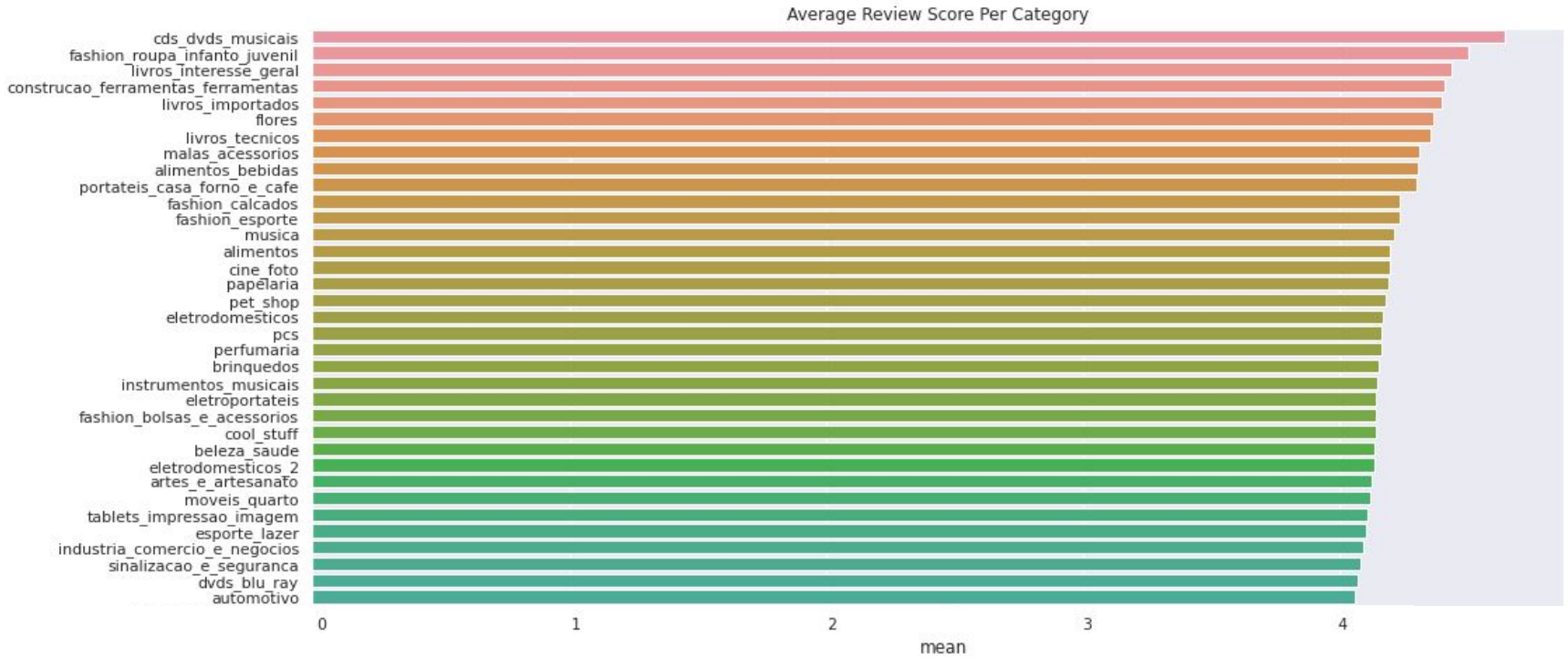
We used a KNN-Imputer on relevant numerical values such as reviews and product size

Data Visualization

Problem Statement | Data Preprocessing | Data Visualization | Data Analysis | Key Takeaways

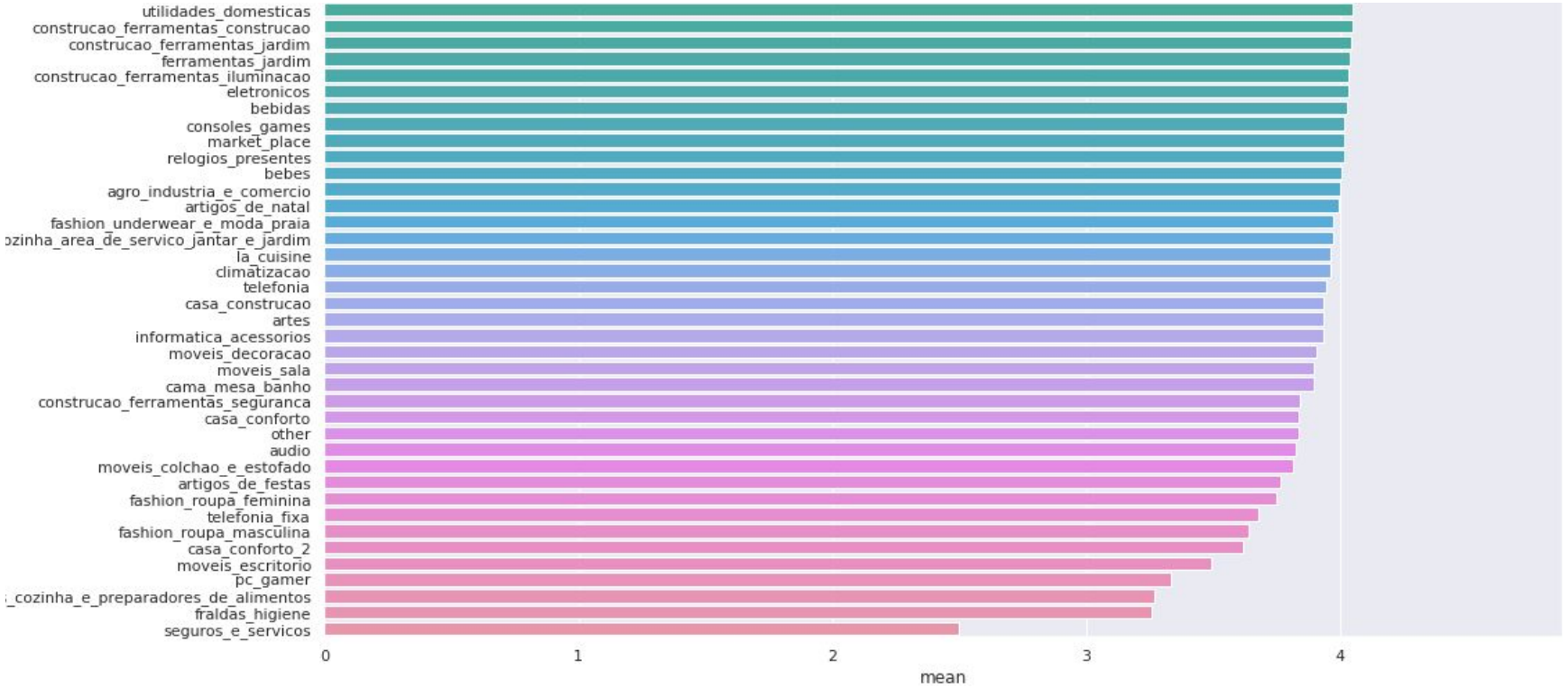
BAR GRAPH OF AVERAGE REVIEW SCORE PER CATEGORY

Packages Used: Seaborn



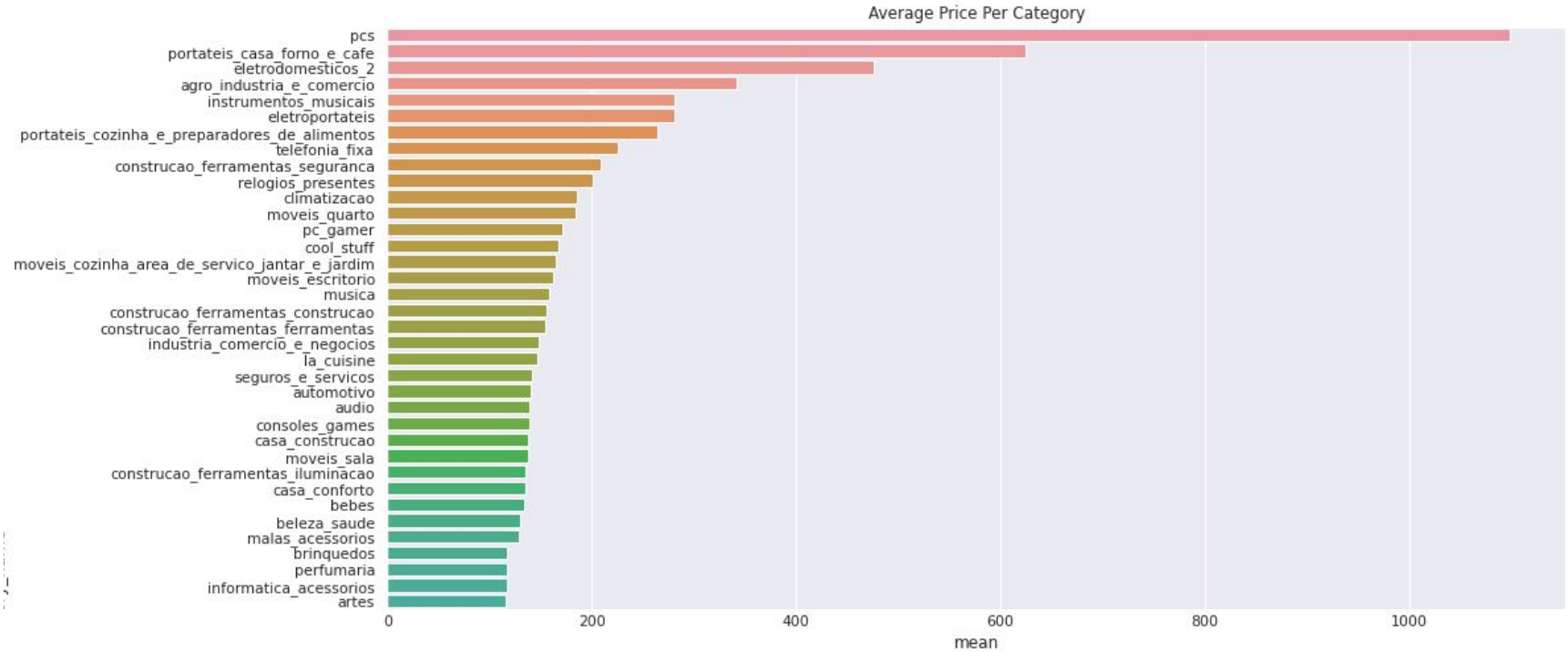
BAR GRAPH OF AVERAGE REVIEW SCORE PER CATEGORY Cont'd

Packages Used: Seaborn



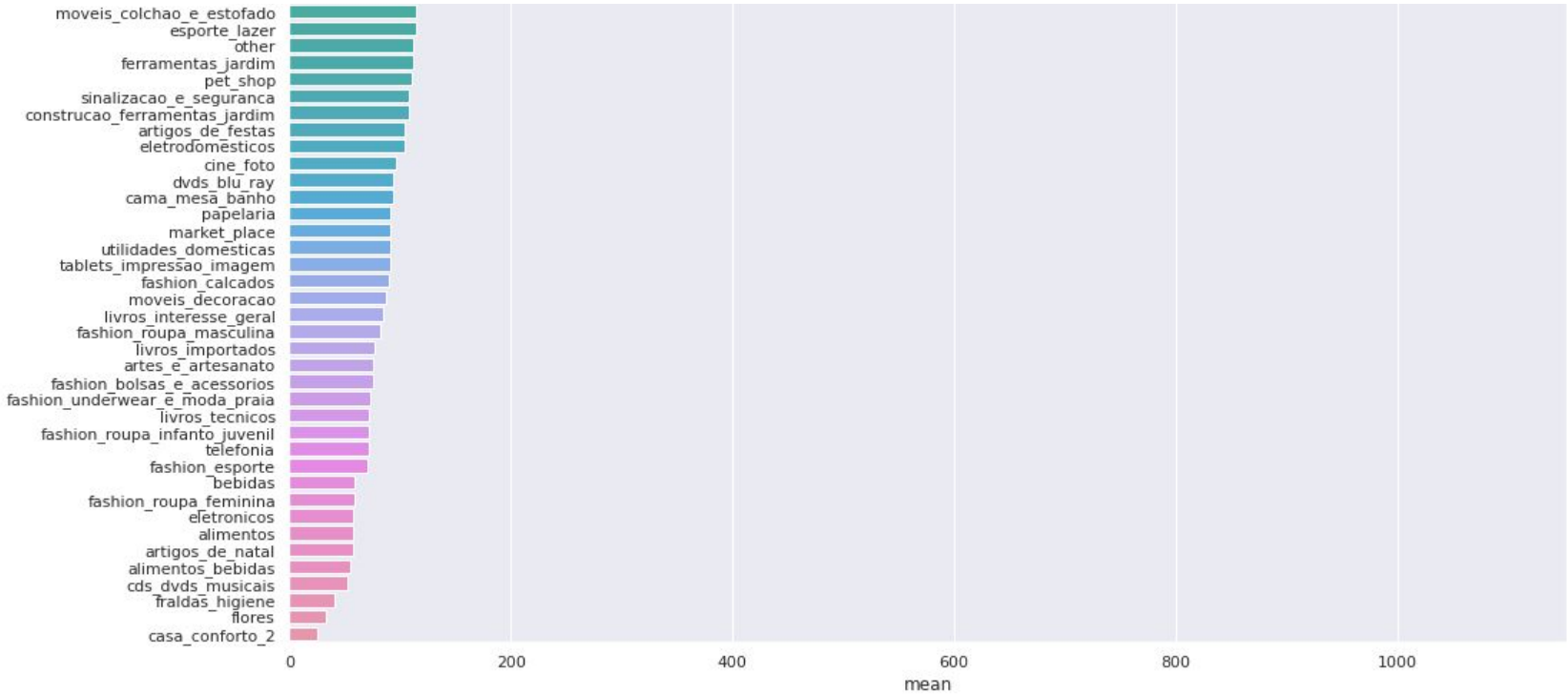
BAR GRAPH OF AVERAGE PRICE PER CATEGORY

Packages Used: Seaborn



BAR GRAPH OF AVERAGE PRICE PER CATEGORY Cont'd

Packages Used: Seaborn



Data Analysis

Problem Statement | Data Preprocessing | Data Visualization | Data Analysis | Key Takeaways

Our team used a

k-Nearest Neighbour model

to recommend products to shoppers

HOW WE CREATE OUR RECOMMENDATIONS

Packages Used: n/a

→ Look through the customer's past purchases

GIVEN Price, Shipping Time → "Features"

FIND A Similar Product_Unique_ID → "Label"

This combination of **GIVEN/FIND** creates a single "Datapoint"

INTRODUCING OUR FEATURES

Packages Used: Pandas, Numpy

Feature Spotlight: Average Review Score

The average review for a product is one of the best indicators of customer satisfaction. Higher rated products should be recommended more often.

Feature List: Fourteen Descriptors Of Product-Customer Fit

Average Price, Feight_value, Product_category_name, Product_name_length, Product_description_length, Product_photos_qty, Product_dimensions

SETTING UP OUR N-DIMENSIONAL SPACE

Packages Used: Scikit Learn, Numpy

Each Feature Represents A Dimension

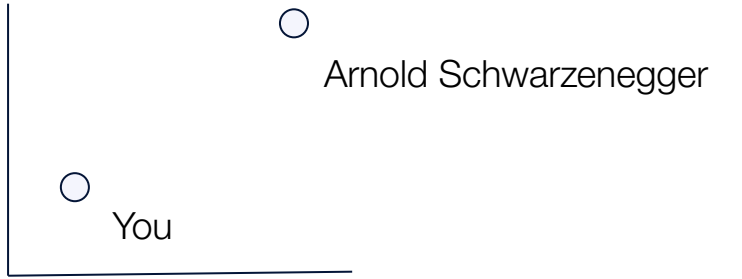
Travelling in a positive direction along a dimensional axis represents an increase in a feature value. The number of dimensions is equal to the number of features.

→ All the products available for purchase database occupy a point in this space.

TWO DIMENSIONAL SPACE

X-AXIS: Bodybuilding Skills

Y-AXIS: Acting Skills



SETTING UP OUR N-DIMENSIONAL SPACE Cont'd

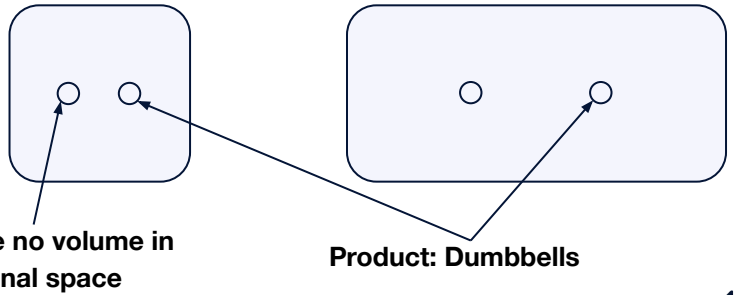
Packages Used: Scikit Learn, Numpy

Important Features Have Stretched Dimensions

Each feature is weighted according to their importance in influencing a customer's purchase decision. The absolute value of the distance between two points on a stretched axis will grow

→ More important features are stretched more

AXIS: "How Much You Recommend This Product"

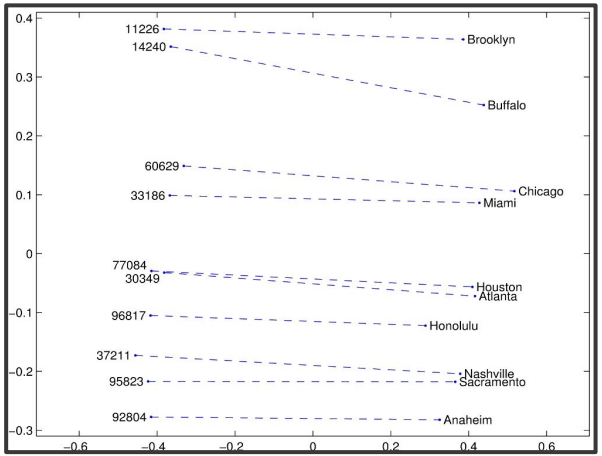


AXIS: "How Much Arnold Schwarzenegger Recommends This Product"

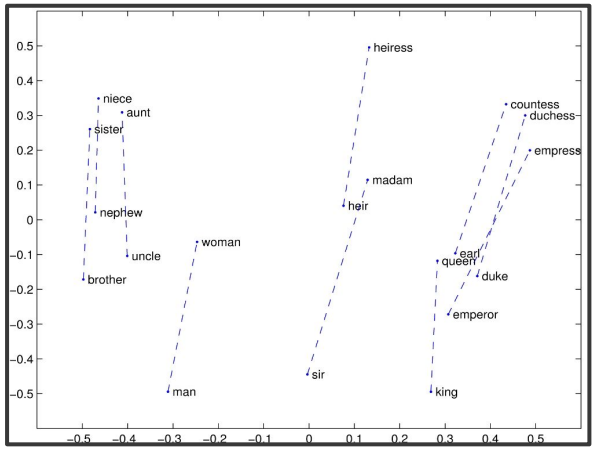
There’s still one more dimension we need to discuss: **Product Category**

GLOVE-50D EMBEDDING & COSINE SIMILARITY

Packages Used: Typing, Models, Numpy, Pandas, Re, Collections



city - zip code



man - woman

embedding dimensions

dictionary

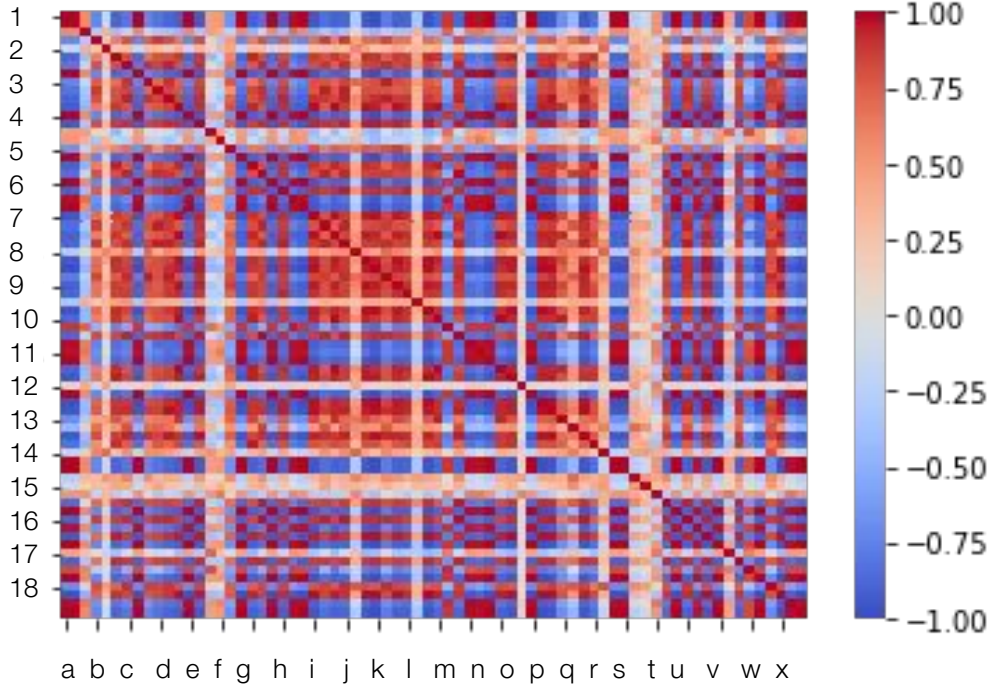
	emb_dim 1	emb_dim 2	...	emb_dim n
cat	⋱			⋮
dog		⋱		⋮
⋮			⋱	⋮
car	...	...	...	*car* × emb_dim n

example embedding

```
'perfumery': array([-0.45596  , -0.13718  , -0.23754  , -0.63402  ,
       -0.42744  ,  0.83699  ,  0.23567  , -0.078916 ,  0.5      ,
        0.08264  ,  0.2316   ,  0.77999  ,  0.47049  , -0.32597  ,
       -0.51283  ,  0.30945  ,  0.47881  ,  0.55381  , -0.24994  ,
       -0.47739  , -0.073831 ,  0.072328 , -0.10644  ,  0.25275  ,
        0.97813  , -0.38809  ,  0.29865  ,  0.29393  ,  0.35823  ,
       -1.1929   ,  0.071931 ,  0.034599 ,  0.027147 , -0.38162  ,
        0.73098  ,  0.062593 , -0.14562  , -0.38141  , -0.072509 ,
        0.20806  , -0.54812  , -0.21912  ,  0.51654  ,  0.64665  ,
       -0.65962  ,  0.10605  ,  0.17607  ,  0.29246  ,  0.0042172])
```

Heatmap of Similarities Between Categories

Packages Used: Matplotlib



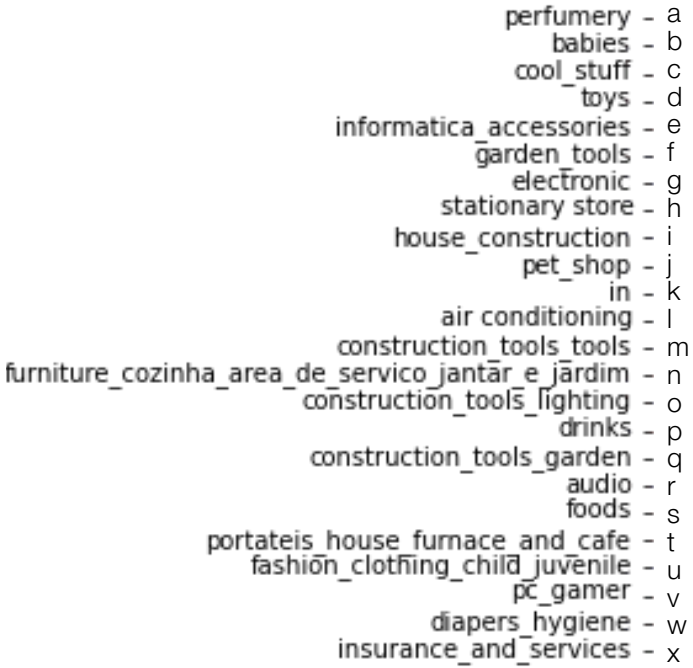
Cosine Similarity $\frac{\vec{a} \cdot \vec{b}}{|a||b|}$

NOTE: Axis ticks move up at different intervals for the x and y axes

Legend can be found on following slide

Heatmap of Similarities Between Categories (Graph Legend)

Packages Used: Matplotlib



Now we are ready to generate **product recommendations** based on purchase history

FINDING THE HIGHEST SIMILARITY

Packages Used: Scikit Learn

We find points that are located close together

Within our n-dimensional space, the distance between two points can be found with a slightly modified **Euclidean Distance** formula.

→ Based on a customer's purchase history, what are products (points) that are similar (close) to past purchases? **The close points are what we are recommending**

Euclidean Distance: $$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

FINDING THE NUMBER OF SIMILAR PRODUCTS

Packages Used: Scikit Learn

We find five similar products per past purchase

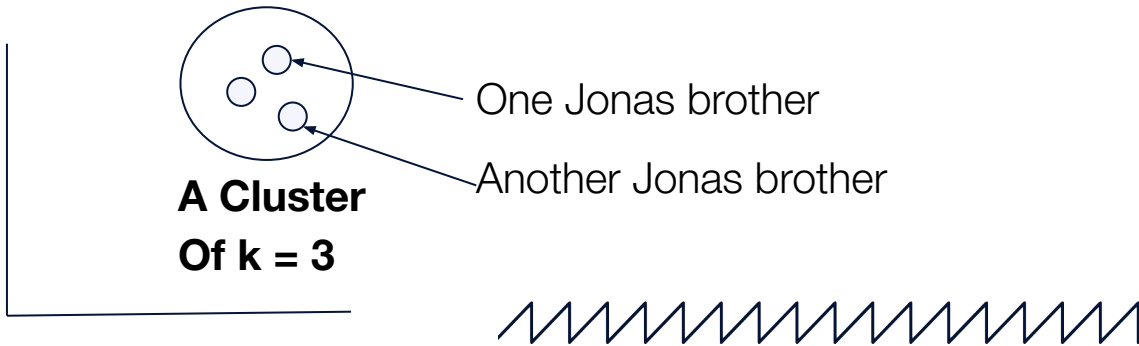
The variable k is the **number of new products that are similar to the purchased product**, including the purchased product itself. For k = 6, these six products create a "cluster" of five products we can choose to recommend.

→ If a recommend product has already been purchased, we **strike it from the cluster**

TWO DIMENSIONAL SPACE

X-AXIS: Singing Skills

Y-AXIS: Handsomeness



Key Takeaways

Problem Statement | Data Preprocessing | Data Visualization | Data Analysis | Key Takeaways

Thank You!

Sunny Son
Machine Learning Engineer

Shane Sun
Data Scientist

Morgan Xu
Software Engineer

Sunny Yang
Data Scientist

Son, Sons & Company