E-commerce customer behaviour study

Using AI to study e-commerce customer data set and create recommendation.

01

Business Problem

Business problems and business goals to achieve

What we have:

An e-commerce company, to sales strategies to increase revenue for next half year.
Currently, we have available data set of customers, including:

- behaviour data
- Items properties
- category tree

What we have:

Behaviour data: events like clicks, add to carts, transactions, represent interactions that were collected over a period of 4.5 months. A visitor can make three types of events, namely “view”, “addtocart” or “transaction”

Items properties: describing unique items: for example: price over times of a items

Category tree: products category

What to do:

What recommender system we are trying to achieve:

- suggest products based on the items the customer is viewing .
- we use item-based collaborative filter and linear regression to help find any false positives on the possible recommendations given out.

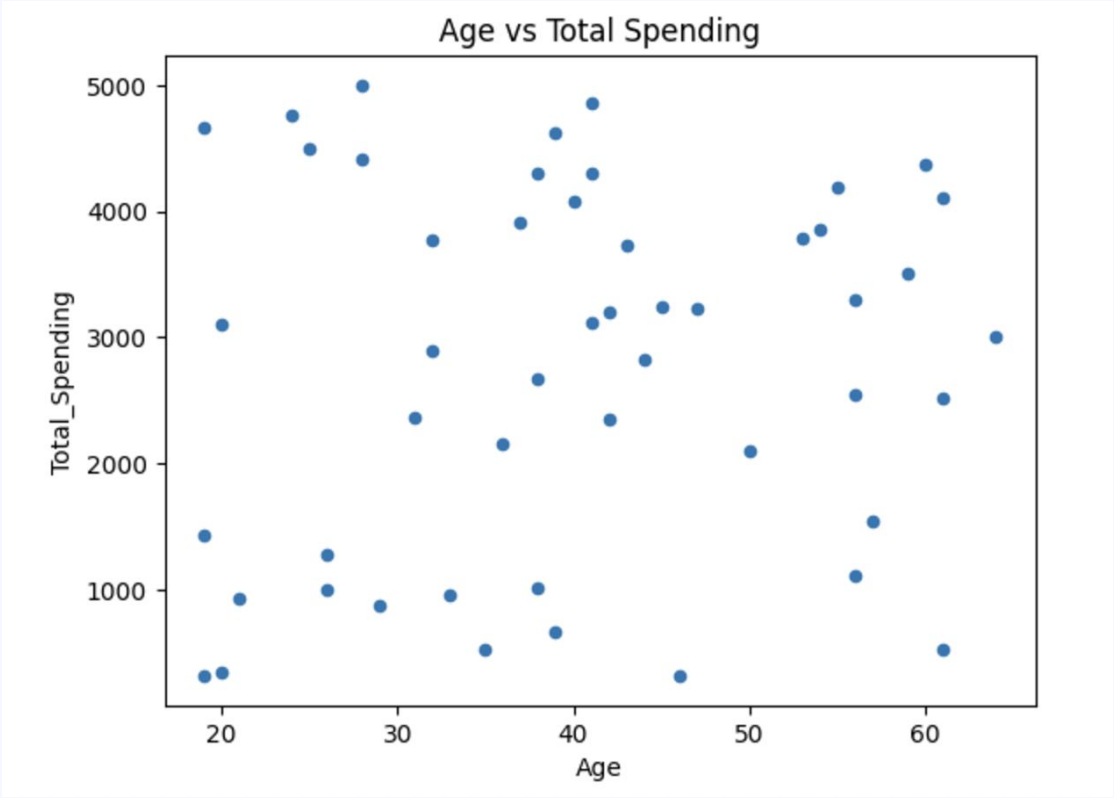
Steps:

1. Data Preprocessing:
   - Clean and normalize data.
   - Scale numerical features like Age for consistency.
2. Recommendation System Models: Collaborative Filtering:
   - Logistic Regression: Identify and reduce false positives in recommendations.
   - ecommend products based on item similarities.
   - Customer behaviour based on the data and do basic customer segmentation.
   - Visitors clustering
3. Evaluation Metrics :
   - Confusion Matrix: Visualize recommendation accuracy.
   - Root Mean Squared Error (RMSE) for collaborative filtering.
4. Visualization

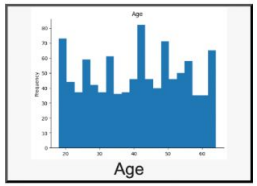
Data Cleaning

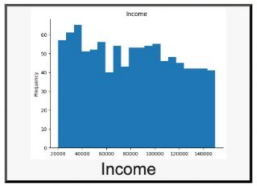
1. userId:
   - Unique identifier for each user.
   - Used for tracking preferences and interactions.
2. ProductId`:
   - Unique identifier for each product.
   - Maps purchases to specific items.
3. Age :
   - User demographic information for segmentation.
   - Helps group customers for tailored recommendations.
4. Interests`:
   - Captures user preferences (e.g., "electronics," "clothing").
   - Enhances recommendation accuracy through personalization.

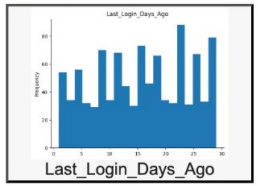
Data Visualise

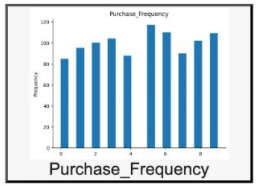


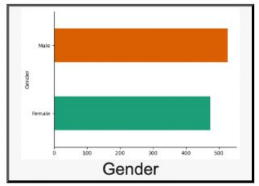
Next steps: Generate code with data | View recommended plots | New interactive sheet

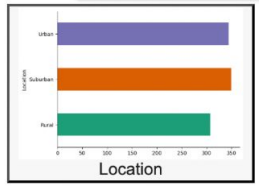

Age

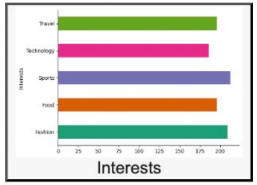

Income

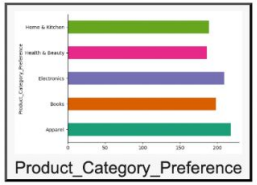

Last_Login_Days_Ago

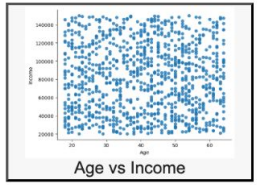

Purchase_Frequency

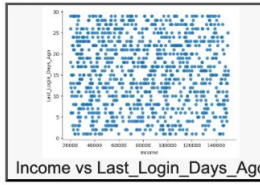

Gender

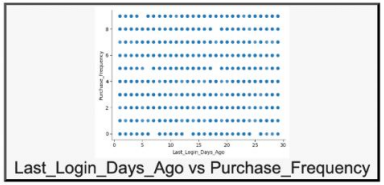

Location

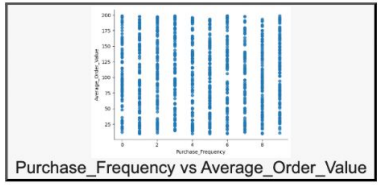

Interests

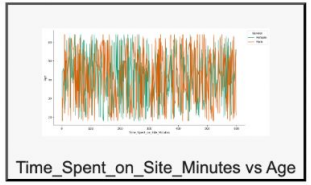

Product_Category_Preference

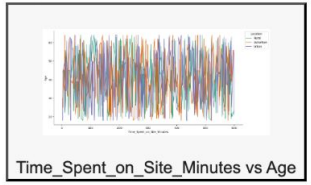

Age vs Income

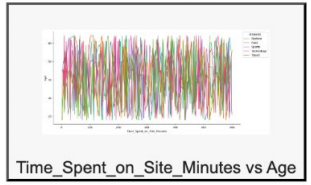

Income vs Last_Login_Days_Ago

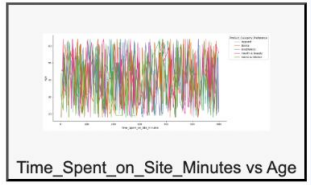

Last_Login_Days_Ago vs Purchase_Frequency

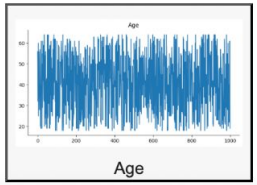

Purchase_Frequency vs Average_Order_Value

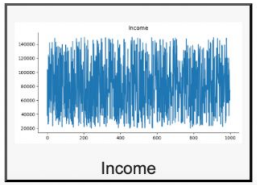

Time_Spent_on_Site_Minutes vs Age

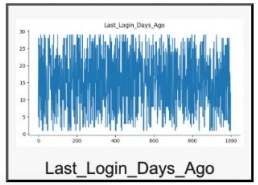

Time_Spent_on_Site_Minutes vs Age

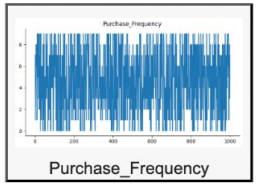

Time_Spent_on_Site_Minutes vs Age

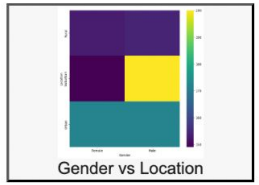

Time_Spent_on_Site_Minutes vs Age

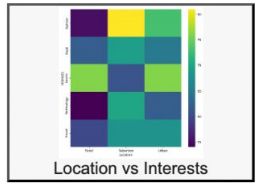

Age


Income


Last_Login_Days_Ago


Purchase_Frequency


Gender vs Location

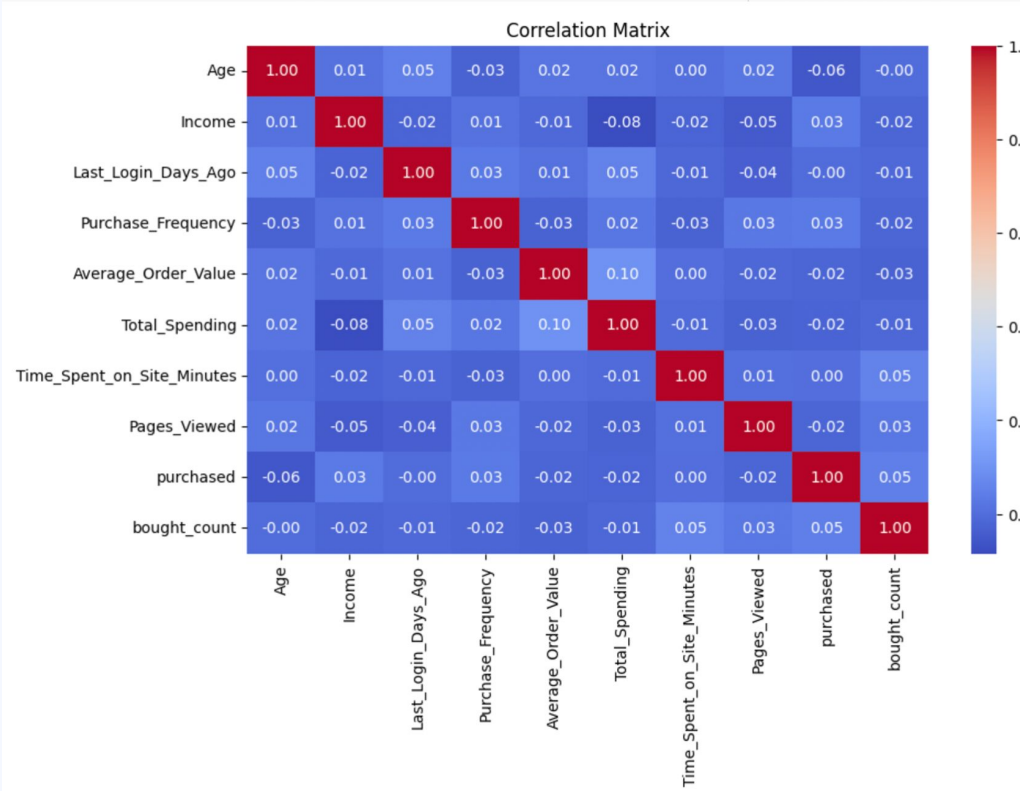

Location vs Interests

Regression results

Variables:

'Age', 'Income', 'Total_Spending',
'Pages_Viewed', 'Gender', 'Location',
'Interests', 'Newsletter_Subscription',
'Time_Spent_on_Site_Minutes',
'Purchase_Frequency',
'Average_Order_Value', 'visitor',
'bought_count'

Model Accuracy: 0.4967



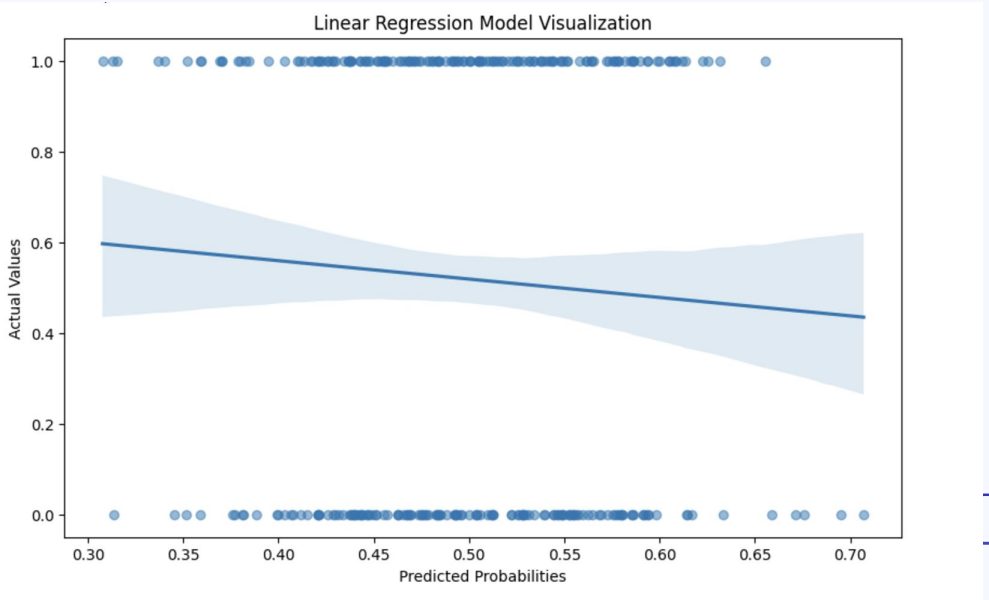
Regression results

With predicted values from the logistic regression model

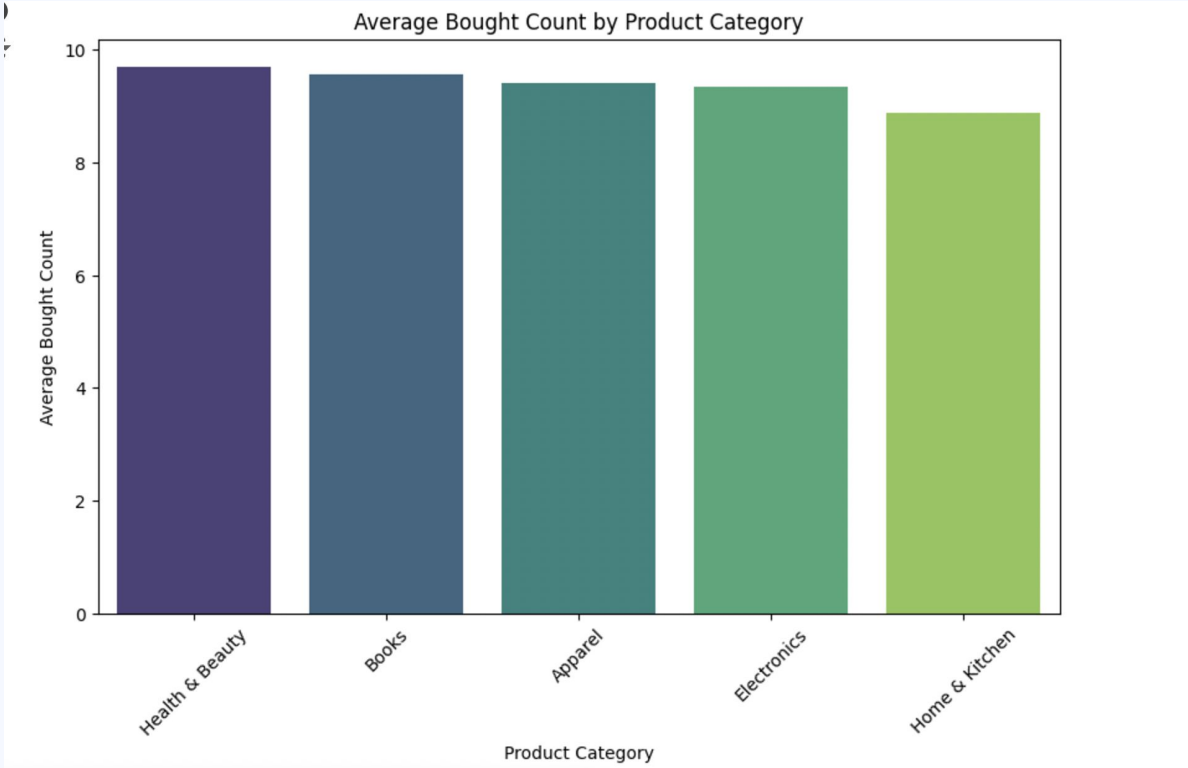
=> Calculate the predicted probabilities for visualization

```
y_pred_prob =
logreg.predict_proba(X_test)[:, 1]
```

```
Root Mean Squared Error: 0.5098643800002995
```

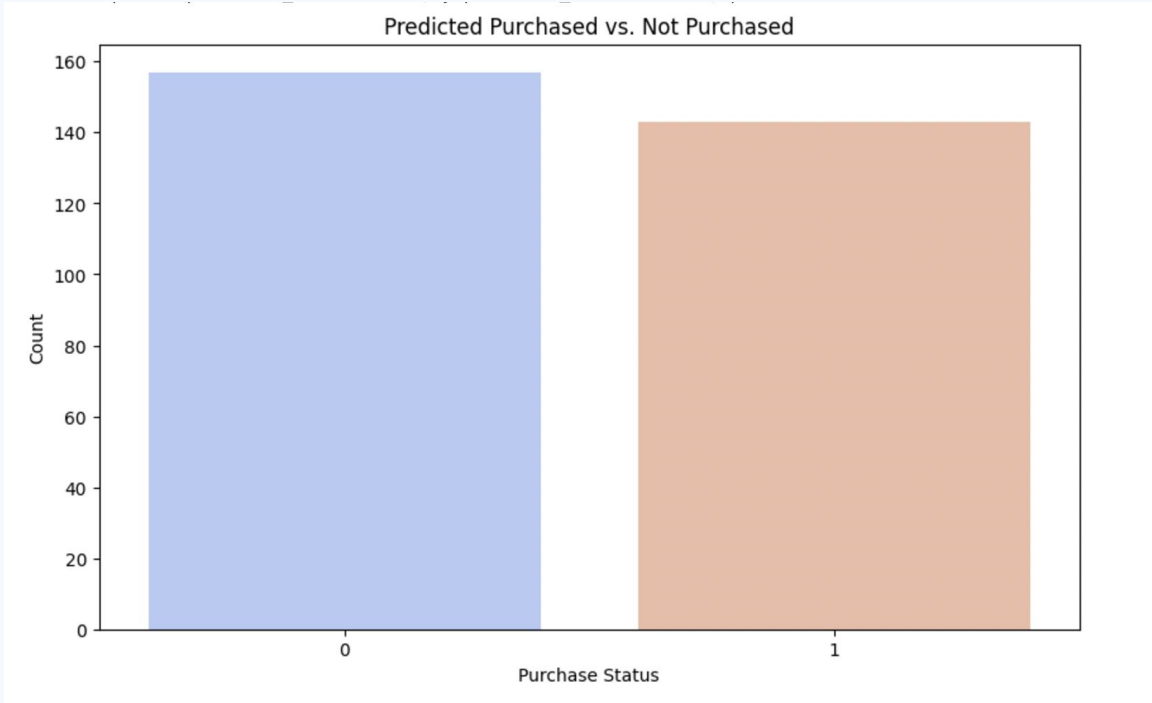


Ranking of Bought count



Aggregate Product Preferences from Predictions

Support Vector Machine: Use the model's predictions (y_pred) to aggregate the predicted purchase status



Item recommendation:

present to the visitor a list of the other items a customer previously bought along with what item the current visitor is viewing e.g. item number 80582

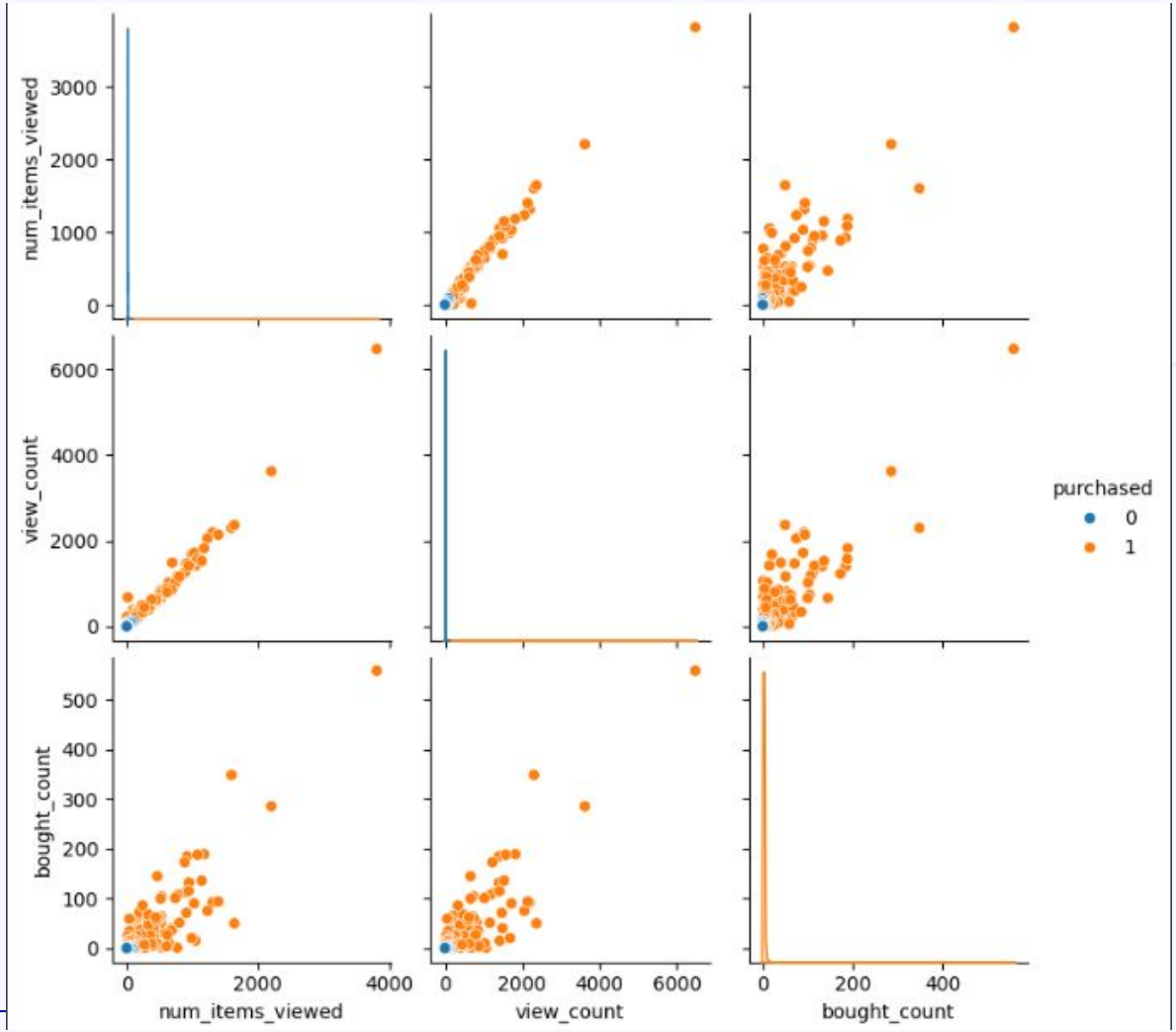
```
def recommender_bought_bought(item_id, purchased_items):


    recommender_list = []
    for x in purchased_items:
        if item_id in x:
            recommender_list += x

    #Then merge recommender list and remove the item id
    recommender_list = list(set(recommender_list) - set([item_id]))

    return recommender_list
```

```
[15] recommender_bought_bought(80582, purchased_items)
```

```
[105792, 200793, 12836, 380775, 15335, 400969, 25353, 302422, 237753, 317178]
```

Visitor clustering

Create new dataframe with new features.
Apply it to buying visitors/viewing visitors.(27821 for 70/30 split)
Combine both dataframes.
Plot it.


num_items_viewed
view_count
bought_count
purchased
0
1
0
1000
2000
3000
0
2000
4000
6000
0
100
200
300
400
500
0
2000
4000
0
2000
4000
6000
0
200
400

Thanks !

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**