EDS Activity 1:

Name: Arjun Pawashe
Roll No- CS5-60
PRN- 202401100112
Dataset: Amazon Product Dataset 2023
Google Collab Link: https://colab.research.google.com/drive/1r-8-k6iOewpmuaTfJI6wFY1g-8LRpe2B?usp=sharing

Activity 1.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

Pandas operations

1. Display basic info (columns, datatypes, missing values)
2. Convert "discount_price" and "actual_price" from string (₹ symbol) to numeric
3. Find average discount price
4. Find top 3 highest rated furniture items
5. Create a new column: "Discount_Percentage" based on actual vs discount price
6. Same price cleaning: remove ₹ and convert to numeric
7. Find item with highest number of ratings
8. Find average rating value
9. Count how many items have rating greater than 4.0
10. Group by 'sub_category' and count items

```
[14] import pandas as pd
     import numpy as np

     furniture_df = pd.read_csv('Furniture.csv')
     cricket_df = pd.read_csv('Cricket.csv')

     #1
     print(furniture_df.info())
     print("-----------------------------------------------------------------")

     #2
     furniture_df['discount_price'] = furniture_df['discount_price'].replace('₹', '', regex=True).replace(',', '', regex=True).astype(float)
     furniture_df['actual_price'] = furniture_df['actual_price'].replace('₹', '', regex=True).replace(',', '', regex=True).astype(float)
     print("-----------------------------------------------------------------")

     #3
     avg_discount_price = furniture_df['discount_price'].mean()
     print(avg_discount_price)
     print("-----------------------------------------------------------------")

     #4
     furniture_df['ratings'] = pd.to_numeric(furniture_df['ratings'], errors='coerce')
     top3_furniture = furniture_df.sort_values('ratings', ascending=False).head(3)
     print(top3_furniture[['name', 'ratings']])
     print("-----------------------------------------------------------------")


     #5
     furniture_df['Discount_Percentage'] = ((furniture_df['actual_price'] - furniture_df['discount_price']) / furniture_df['actual_price']) * 100
     print(furniture_df[['name', 'Discount_Percentage']].head())
     print("-----------------------------------------------------------------")

     #6
     cricket_df['discount_price'] = cricket_df['discount_price'].replace('₹', '', regex=True).replace(',', '', regex=True).astype(float)
     cricket_df['actual_price'] = cricket_df['actual_price'].replace('₹', '', regex=True).replace(',', '', regex=True).astype(float)
     print("-----------------------------------------------------------------")


     #7
     cricket_df['no_of_ratings'] = pd.to_numeric(cricket_df['no_of_ratings'].replace(',', '', regex=True), errors='coerce')
     most_rated_item = cricket_df.loc[cricket_df['no_of_ratings'].idxmax()]
     print(most_rated_item[['name', 'no_of_ratings']])
     print("-----------------------------------------------------------------")


     #8
     cricket_df['ratings'] = pd.to_numeric(cricket_df['ratings'], errors='coerce')
     avg_rating_cricket = cricket_df['ratings'].mean()
     print(avg_rating_cricket)
     print("-----------------------------------------------------------------")


     #9
     high_rated_count = (cricket_df['ratings'] > 4.0).sum()
     print(high_rated_count)
     print("-----------------------------------------------------------------")
     #10
     subcategory_counts = cricket_df['sub_category'].value_counts()
     print(subcategory_counts)
     print("-----------------------------------------------------------------")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1320 entries, 0 to 1319
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   name            1320 non-null   object
 1   main_category   1320 non-null   object
 2   sub_category    1320 non-null   object
 3   image           1320 non-null   object
 4   link            1320 non-null   object
 5   ratings         1189 non-null   object
 6   no_of_ratings   1189 non-null   object
 7   discount_price  1279 non-null   object
 8   actual_price    1301 non-null   object
dtypes: object(9)
memory usage: 92.9+ KB
None
-----------------------------------------------------------------
-----------------------------------------------------------------
2534.9077482408134
```

EDS Activity 1:

Name: Arjun Pawashe
Roll No- CS5-60
PRN- 202401100112
Dataset: Amazon Product Dataset 2023
Google Collab Link: https://colab.research.google.com/drive/1r-8-k6iOewpmuaTfJI6wFY1g-8LRpe2B?usp=sharing

Numpy Oprations

1. Extract discount_price and actual_price into numpy arrays
2. Find mean and median of discount prices
3. Find how many items have discount greater than ₹500
4. Create a boolean mask where discount price < ₹1000
5. Calculate savings array = actual_price - discount_price
6. Extract ratings into numpy array
7. Calculate standard deviation of ratings
8. Find minimum rating and maximum rating
9. Find how many products have no_of_ratings greater than 100
10. Normalize the discount_price values (Min-Max normalization)

```
#1
furniture_discount_prices = furniture_df['discount_price'].to_numpy()
furniture_actual_prices = furniture_df['actual_price'].to_numpy()

#2
mean_discount = np.mean(furniture_discount_prices)
median_discount = np.median(furniture_discount_prices)
print(mean_discount)
print(median_discount)

#3
greater_than_500 = np.sum(furniture_discount_prices > 500)
print(greater_than_500)

#4
mask_under_1000 = furniture_discount_prices < 1000
print(mask_under_1000)

#5
savings = furniture_actual_prices - furniture_discount_prices
print(savings)

#6
cricket_ratings = cricket_df['ratings'].to_numpy()

#7
std_dev_ratings = np.std(cricket_ratings)
print(std_dev_ratings)

#8
min_rating = np.min(cricket_ratings)
max_rating = np.max(cricket_ratings)
print(min_rating)
print(max_rating)

#9
cricket_no_of_ratings = cricket_df['no_of_ratings'].to_numpy()
more_than_100 = np.sum(cricket_no_of_ratings > 100)
print(more_than_100)

#10
cricket_discount_prices = cricket_df['discount_price'].to_numpy()
cricket_discount_prices = cricket_discount_prices[~np.isnan(cricket_discount_prices)]  # remove NaNs

normalized_discount_prices = (cricket_discount_prices - np.min(cricket_discount_prices)) / (np.max(cricket_discount_prices) - np.min(cricket_discount_prices))
print(normalized_discount_prices)
```

```
nan
nan
923
[ True False  True ...  True False  True]
[ 955. 2300.  168. ...  620. 1009. 1000.]
nan
nan
nan
128
[0.00417098 0.00754326 0.00369768 ... 0.00754326 0.01940541 0.00801657]
```

Colab paid products - Cancel contracts here

0s completed at 11:08 PM