

NUMPY AND PANDAS ASSIGNMENT

AMAZON PRODUCT DATASET

NAME: HATAGALE DHIRAJ BHAUSAHEB

PRN: 202401050070

ROLL NO: CC59

BATCH: CC3

1. FIND THE AVERAGE PRICE OF ALL PRODUCTS.

Solution: `df['price'].mean()`

2. FIND THE TOP 5 MOST EXPENSIVE PRODUCTS.

Solution: `df.sort_values('price', ascending=False).head(5)`

3. COUNT THE NUMBER OF PRODUCTS IN EACH CATEGORY.

Solution: `df['category'].value_counts()`

4. CALCULATE THE AVERAGE RATING FOR EACH CATEGORY.

Solution: `df.groupby('category')['rating'].mean()`

5. FIND THE PRODUCT WITH THE HIGHEST NUMBER OF REVIEWS.

Solution: `df.loc[df['number_of_reviews'].idxmax()]`

6. LIST ALL PRODUCTS THAT ARE CURRENTLY OUT OF STOCK.

Solution: `df[df['availability'] == 'Out of Stock']`

7. FIND THE AVERAGE DISCOUNT GIVEN BY BRAND.

Solution: `df.groupby('brand')['discount'].mean()`

8. FILTER PRODUCTS WITH PRICE ABOVE 1000 AND RATING ABOVE 4.5.

Solution: `df[(df['price'] > 1000) & (df['rating'] > 4.5)]`

9. FIND THE PERCENTAGE OF IN-STOCK VS OUT-OF-STOCK PRODUCTS.

Solution: `df['availability'].value_counts(normalize=True) * 100`

10. CREATE A NEW COLUMN FOR DISCOUNTED PRICE.

Solution: `df['discounted_price'] = df['price'] * (1 - df['discount'] / 100)`

11. CALCULATE THE TOTAL NUMBER OF UNIQUE BRANDS.

Solution: `df['brand'].nunique()`

12. FIND THE BRAND WITH THE MOST NUMBER OF PRODUCTS.

Solution: `df['brand'].value_counts().idxmax()`

13. CHECK IF THERE ARE ANY MISSING VALUES IN THE DATASET.

Solution: `df.isnull().sum()`

14. FILL MISSING RATINGS WITH THE AVERAGE RATING.

Solution: `df['rating'].fillna(df['rating'].mean(), inplace=True)`

15. LIST THE TOP 3 SELLERS WITH THE HIGHEST AVERAGE RATING.

Solution: `df.groupby('seller')['rating'].mean().sort_values(ascending=False).head(3)`

16. CALCULATE CORRELATION BETWEEN PRICE AND NUMBER OF REVIEWS.

Solution: `df['price'].corr(df['number_of_reviews'])`

17. GET THE DISTRIBUTION OF PRODUCT PRICES USING NUMPY HISTOGRAM.

Solution: `np.histogram(df['price'], bins=10)`

18. FIND STANDARD DEVIATION OF PRODUCT RATINGS.

Solution: `df['rating'].std()`

19. FILTER OUT PRODUCTS THAT BELONG TO THE 'ELECTRONICS' CATEGORY.

Solution: `df[df['category'] == 'Electronics']`

20. GROUP BY CATEGORY AND COUNT PRODUCTS WITH RATING > 4.

Solution: `df[df['rating'] > 4].groupby('category')['product_id'].count()`