Optimizing Amazon Sales: A Comprehensive Analysis Using NumPy and Pandas

Introduction to Amazon Sales

In this presentation, we will explore **optimizing Amazon sales** through a comprehensive analysis using **NumPy** and **Pandas**. These powerful tools will help us to efficiently process data, uncover insights, and enhance sales strategies for better performance in the competitive Amazon marketplace.

Amazon Sales Analysis.ipynb

File Edit View Insert Runtime Tools Help Last edited on 28 June

Comment Share

+ Code + Text

Connect Gemini

Importing Required Libraries:

```
[ ] import numpy as np
    import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt
```

Importing the dataset:

```
[ ] sales_data = pd.read_csv('SALESDATA.csv', parse_dates=['DateKey','Invoice Date', 'Promised Delivery Date'])
    sales_data
```

	CustKey	DateKey	Discount Amount	Invoice Date	Invoice Number	Item Class	Item Number	Item	Line Number	List Price	Order Number	Promised Delivery Date	Sales Amount	Sales Amount Based on List Price	Sales Cost Amount	Sales Margin Amount	Sales Price	Sales Quantity	Sales Rep	U/M
0	10000481	2017-04-30	-237.91	2017-04-30	100012	NaN	NaN	Urban Large Eggs	2000	0.00	200015	2017-04-30	237.91	0.00	0.00	237.91	237.910000	1	184	EA
1	10002220	2017-07-14	368.79	2017-07-14	100233	P01	20910	Moms Sliced Turkey	1000	824.96	200245	2017-07-14	456.17	824.96	0.00	456.17	456.170000	1	127	EA
2	10002220	2017-10-17	109.73	2017-10-17	116165	P01	38076	Cutting Edge Foot-Long Hot Dogs	1000	548.66	213157	2017-10-16	438.93	548.66	0.00	438.93	438.930000	1	127	EA
3	10002489	2017-06-03	-211.75	2017-06-03	100096	NaN	NaN	Kiwi Lox	1000	0.00	200107	2017-06-03	211.75	0.00	0.00	211.75	211.750000	1	160	EA
4	10004516	2017-05-27	96627.94	2017-05-27	103341	P01	60776	High Top Sweet Onion	1000	408.52	203785	2017-05-28	89248.66	185876.60	0.00	89248.66	196.150901	455	124	SE
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65277	10017638	2018-03-21	505.78	2018-03-21	226497	P01	13447	High Top Oranges	8000	119.52	320895	2018-03-21	569.90	1075.68	239.95	329.95	63.322222	9	180	EA
		2018-03-		2018-03-				Landslide White												

Understanding Amazon Marketplace

The **Amazon marketplace** is vast and dynamic. To succeed, sellers must understand **consumer behavior**, **trends**, and **competition**. Utilizing data analysis tools like **NumPy** and **Pandas** enables sellers to extract valuable insights from sales data, driving informed decision-making.

Data Collection Techniques

Effective **data collection** is crucial for analysis. Sellers can gather data from **Amazon Seller Central**, **web scraping**, or utilizing **APIs.** This data forms the foundation for our analysis, allowing us to identify patterns and opportunities for sales optimization.

Assign: | New Tab | What is Linux? | Op... | AdBlock is now inst... | All B

Amazon Sales Analysis.ipynb

File Edit View Insert Runtime Tools Help Last edited on 28 June

Comment Share

+ Code + Text

Connect

Connect to a new runtime

```
[ ] sales_data01['Item Class'].value_counts()
```

```
P01    56977
PO1       16
Name: Item Class, dtype: int64
```

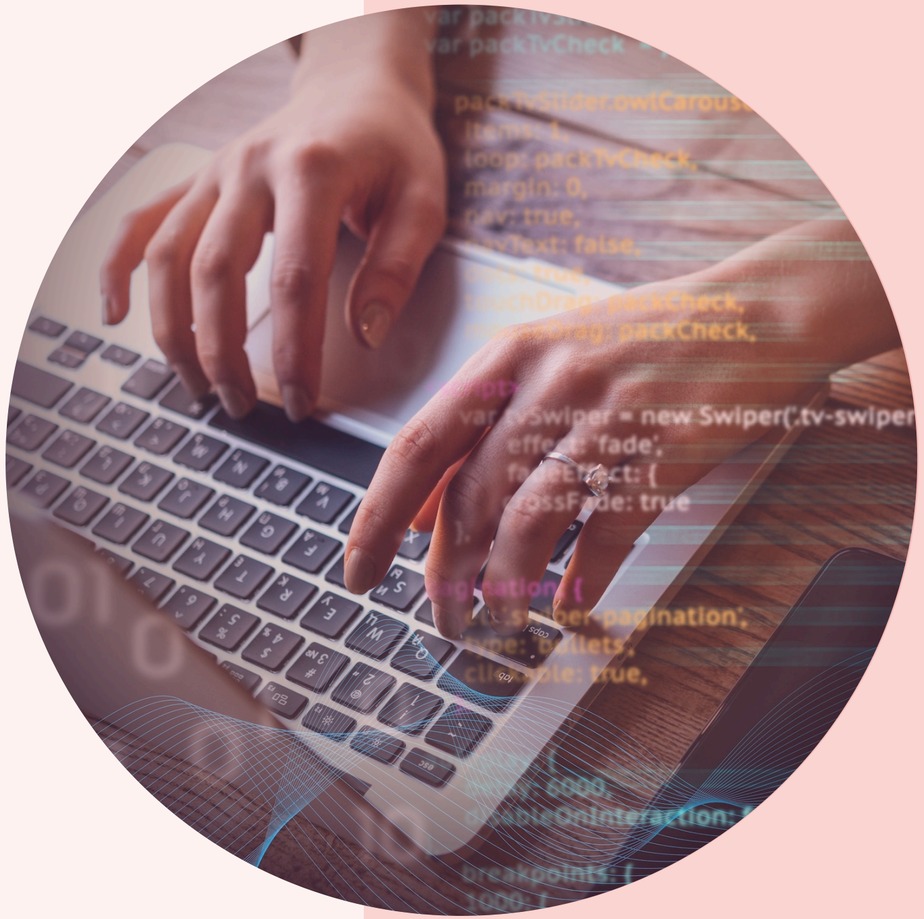
```
[ ] # Removing Null Values
    sales_data01.dropna(subset=['Discount Amount','Sales Price', 'Item Number'],inplace=True)
```

```
[ ] # Generating descriptive statistics
    sales_data01.describe()
```

	CustKey	Discount Amount	Invoice Number	Line Number	List Price	Order Number	Sales Amount	Sales Amount Based on List Price	Sales Cost Amount	Sales Margin Amount	Sales Price	Sales Quantity	Sales Rep
count	6.524100e+04	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000	65241.000000
mean	1.001770e+07	1857.310923	216292.785242	23725.043178	515.016834	180567.610122	2853.121051	4710.431974	1662.023052	1191.097999	283.250283	45.106712	137.421422
std	7.175846e+03	9039.535784	94982.018695	32669.565014	449.144896	67612.238675	15169.020896	20702.609533	9559.396917	5862.566647	250.448615	429.793733	26.644271
min	1.000045e+07	-255820.800000	100034.000000	1000.000000	0.000000	100838.000000	200.010000	0.000000	0.000000	-3932.930000	0.337341	1.000000	103.000000
25%	1.001272e+07	246.280000	117969.000000	3000.000000	181.560000	115281.000000	308.310000	561.040000	167.810000	129.890000	100.030000	2.000000	113.000000
50%	1.001966e+07	442.200000	222904.000000	12000.000000	325.190000	203695.000000	553.940000	999.750000	304.580000	246.480000	183.282857	3.000000	134.000000
75%	1.002351e+07	1001.500000	314325.000000	32000.000000	803.860000	218576.000000	1279.750000	2321.400000	688.550000	578.220000	448.220000	8.000000	160.000000
max	1.002758e+07	343532.660000	332842.000000	344000.000000	2760.700000	321532.000000	555376.000000	632610.160000	366576.000000	188800.000000	6035.000000	16000.000000	185.000000

```
[ ] # Creating Year, Month, Quarter, Day Columns in Sales_data01
    sales_data01['Invoice_Year'] = sales_data['Invoice Date'].dt.year
    sales_data01['Invoice_Month'] = sales_data['Invoice Date'].dt.month
    sales_data01['Invoice_Quarter'] = sales_data['Invoice Date'].dt.quarter
    sales_data01['Invoice_Day'] = sales_data['Invoice Date'].dt.day
```

```
[ ] sales_data01.info()
```



Using NumPy for Analysis

With **NumPy**, we can perform efficient numerical computations on sales data. This library allows for **array manipulations**, statistical calculations, and data transformations, which are essential for analyzing trends and forecasting sales performance effectively.

Leveraging Pandas for Dataframes

Pandas is instrumental in handling structured data with its **DataFrame** capabilities. It allows for easy data manipulation, cleaning, and exploration. By utilizing Pandas, we can visualize sales trends and generate insights that inform strategic decisions.

Assign: | New Tab | What is Linux? | Op... | AdBlock is now inst... | All Bookmarks

Amazon Sales Analysis.ipynb

File Edit View Insert Runtime Tools Help Last edited on 28 June

Comment Share

+ Code + Text

Connect Gemini

Observations:

- 1 Discount Amount is highly related to Sales Amount, Sales Cost Amount, Sales Amount Based on List Price & Sales Margin Amount and moderately related to Sales Quantity.
- 2- List Price highly related to sales price and has no relations with Sales amount, Sales cost amount, Sales amount based on list price & sales margin amount.
- 3- Sales quantity is moderately related to Sales amount, discount amount, sales margin amount.
- Their is no relation Between Sales Rep and Sales Amount, Sales Margin Amount.

```
[ ] # set style
    sns.set(style = 'darkgrid')
```

```
[ ] DaySalesInsights = sales_data01.copy()
    DaySalesInsights['Invoice_Date'] = pd.to_datetime(sales_data01['Invoice Date']).dt.date
```

```
[ ] top10sales = DaySalesInsights.groupby('Invoice_Date').sum().sort_values('Sales Amount', ascending = False)
    top10sales = top10sales.reset_index().head(10)
```

Yearly Sales Record:

```
[ ] Yearly_Sales =  sales_data02[['CustKey','Item','Invoice Date','Invoice_Year','Invoice_Month',
                                  'Sales Quantity', 'Sales Amount Based on List Price','Discount Amount',
                                 'Sales Amount', 'Sales Margin Amount','Sales Cost Amount','Sales Rep','U/M','List Price',
                                 'Sales Price']]
```

```
[ ] Yearly_Sales01 = Yearly_Sales.groupby('Invoice_Year').sum().reset_index()
```

Key Metrics for Optimization

To optimize sales, focus on key metrics such as **conversion rates**, **customer feedback**, and **inventory turnover**. Analyzing these metrics helps identify areas for improvement and informs strategies to enhance overall sales performance on Amazon.

Implementing Data-Driven Strategies

Once insights are gathered, it's essential to implement **data-driven strategies**. This may include adjusting pricing, enhancing product descriptions, or optimizing advertising campaigns. Continuous monitoring and adaptation are crucial for sustained success in the Amazon marketplace.

Conclusion and Future Steps

In conclusion, optimizing Amazon sales through **NumPy** and **Pandas** provides a competitive edge. By leveraging data analysis, sellers can make informed decisions and adapt to market changes. Future steps include ongoing analysis and refining strategies based on emerging trends.

[ ]
```
High_CostAmount = sales_data01.groupby('Item').sum().sort_values('Sales Cost Amount', ascending=False).reset_index()
High_CostAmount.index+=1
High_CostAmount=High_CostAmount.reset_index().rename(columns={"index":"Rank"})
```

Observations:

- Maximum sales quantities were generated in the unit of measure "EA".

Observations:

- Maximum mean level sales margin amounts were generated in "SE" U/M while minimum mean level of them in "PR" U/M.

Observations:

- As seen from the pairplot above, it is quite clear that list price is very strongly positively correlated with sales price.
- A strong correlation exists between sales amount, sales cost amount, sales margin amount, and sales amount based on list price.
- Sales amount, sales cost amount, and sales margin amount are strongly correlated with discount amount.
- Sales quantity is moderately positively correlated to sales amount, sales cost amount, sales margin amount and sales amount based on list price.
- Sales cost amount and sales margin amount are strongly positively correlated.
- Customer key and sales rep are weakly negatively correlated.
- Sales price and sales quantity are weakly negatively correlated.
- Sales quantity and sales rep are very weakly negatively correlated.
- Sales rep is very weakly negatively correlated to both discount amount and sales cost amount.
- Sales rep is very weakly positively correlated to sales margin amount, in fact we can claim them to be independent of each other as well.