Data Science & Exploratory Data Analysis

A PRESENTATION OF INTERNSHIP SUBMITTED BY :

TOPIC: SALES INSIGHTS & CANCER PREDICTION

NAME : VRUSHABH BODARYA

ENROLLMENT NO. : 200020116035

INTERNAL GUIDE : VANITA DANDHWANI

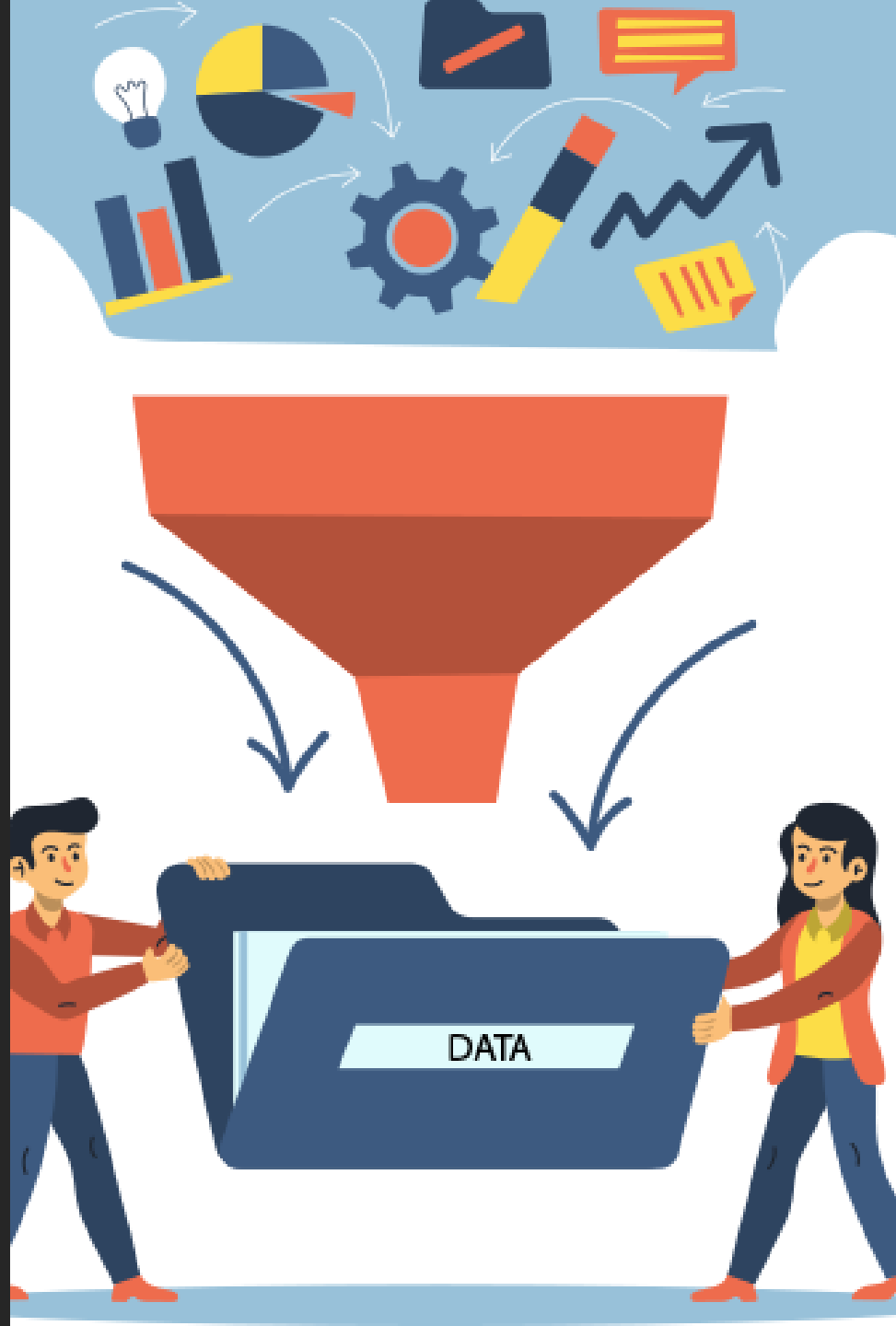
AT THE COMPANY : APPSTONE LAB TECHNOLOGIES LLP

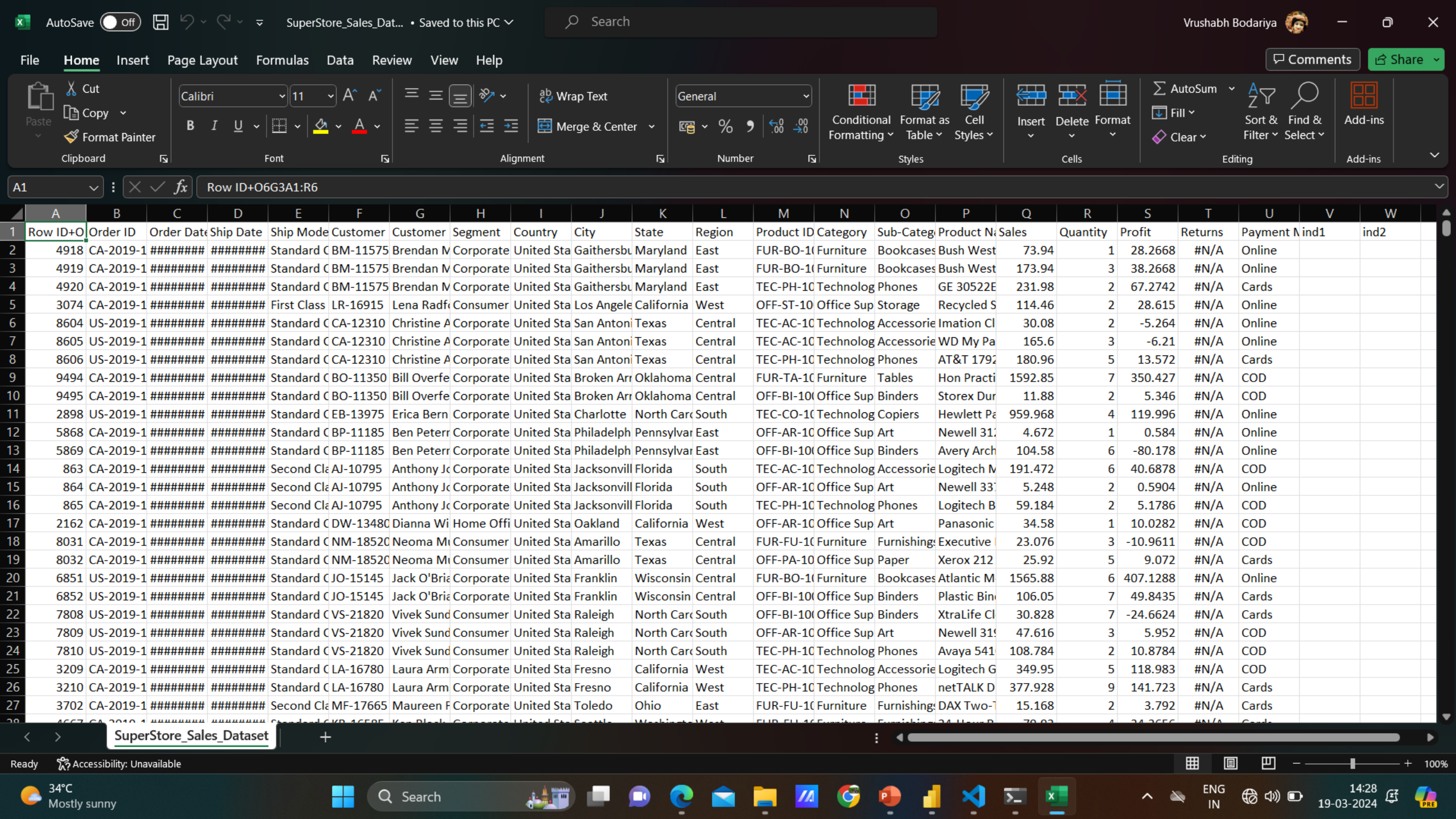
Disclose Sales Insights with Data Analysis, Power BI and Time Series Forecasting

- Welcome to our presentation on finding sales insights using data analysis, Power BI, and time series analysis.
- In today's competitive market, understanding sales trends and forecasting future sales is essential for business success.
- This presentation explores how we can leverage data analysis techniques, Power BI and time series analysis to gain valuable insights into sales data.

Data Collection & Preparation

- The first step in finding sales insights is collecting and preparing the data.
- Data sources may include sales transactions, customer demographics, marketing campaigns, and external factors like economic indicators.
- Cleaning, transforming, and integrating the data ensure its quality and consistency for analysis.





Row ID+O	Order ID	Order Date	Ship Date	Ship Mode	Customer	Customer	Segment	Country	City	State	Region	Product ID	Category	Sub-Categ	Product Na	Sales	Quantity	Profit	Returns	Payment M	ind1	ind2
4918	CA-2019-1	########	########	Standard C	BM-11575	Brendan M	Corporate	United Sta	Gaithersbu	Maryland	East	FUR-BO-1(	Furniture	Bookcases	Bush West	73.94	1	28.2668	#N/A	Online		
4919	CA-2019-1	########	########	Standard C	BM-11575	Brendan M	Corporate	United Sta	Gaithersbu	Maryland	East	FUR-BO-1(	Furniture	Bookcases	Bush West	173.94	3	38.2668	#N/A	Online		
4920	CA-2019-1	########	########	Standard C	BM-11575	Brendan M	Corporate	United Sta	Gaithersbu	Maryland	East	TEC-PH-1(	Technolog	Phones	GE 30522E	231.98	2	67.2742	#N/A	Cards		
3074	CA-2019-1	########	########	First Class	LR-16915	Lena Radf	Consumer	United Sta	Los Angele	California	West	OFF-ST-10	Office Sup	Storage	Recycled S	114.46	2	28.615	#N/A	Online		
8604	US-2019-1	########	########	Standard C	CA-12310	Christine A	Corporate	United Sta	San Antoni	Texas	Central	TEC-AC-1(	Technolog	Accessorie	Imation Cl	30.08	2	-5.264	#N/A	Online		
8605	US-2019-1	########	########	Standard C	CA-12310	Christine A	Corporate	United Sta	San Antoni	Texas	Central	TEC-AC-1(	Technolog	Accessorie	WD My Pa	165.6	3	-6.21	#N/A	Online		
8606	US-2019-1	########	########	Standard C	CA-12310	Christine A	Corporate	United Sta	San Antoni	Texas	Central	TEC-PH-1(	Technolog	Phones	AT&T 1792	180.96	5	13.572	#N/A	Cards		
9494	CA-2019-1	########	########	Standard C	BO-11350	Bill Overfe	Corporate	United Sta	Broken Arr	Oklahoma	Central	FUR-TA-1(	Furniture	Tables	Hon Practi	1592.85	7	350.427	#N/A	COD		
9495	CA-2019-1	########	########	Standard C	BO-11350	Bill Overfe	Corporate	United Sta	Broken Arr	Oklahoma	Central	OFF-BI-10(	Office Sup	Binders	Storex Dur	11.88	2	5.346	#N/A	COD		
2898	US-2019-1	########	########	Standard C	EB-13975	Erica Bern	Corporate	United Sta	Charlotte	North Car(	South	TEC-CO-1(	Technolog	Copiers	Hewlett Pa	959.968	4	119.996	#N/A	Online		
5868	CA-2019-1	########	########	Standard C	BP-11185	Ben Peterr	Corporate	United Sta	Philadelph	Pennsylvar	East	OFF-AR-1(	Office Sup	Art	Newell 31	4.672	1	0.584	#N/A	Online		
5869	CA-2019-1	########	########	Standard C	BP-11185	Ben Peterr	Corporate	United Sta	Philadelph	Pennsylvar	East	OFF-BI-10(	Office Sup	Binders	Avery Arch	104.58	6	-80.178	#N/A	Online		
863	CA-2019-1	########	########	Second Cla	AJ-10795	Anthony Jc	Corporate	United Sta	Jacksonvill	Florida	South	TEC-AC-1(	Technolog	Accessorie	Logitech M	191.472	6	40.6878	#N/A	COD		
864	CA-2019-1	########	########	Second Cla	AJ-10795	Anthony Jc	Corporate	United Sta	Jacksonvill	Florida	South	OFF-AR-1(	Office Sup	Art	Newell 33	5.248	2	0.5904	#N/A	Online		
865	CA-2019-1	########	########	Second Cla	AJ-10795	Anthony Jc	Corporate	United Sta	Jacksonvill	Florida	South	TEC-PH-1(	Technolog	Phones	Logitech B	59.184	2	5.1786	#N/A	COD		
2162	CA-2019-1	########	########	Standard C	DW-13480	Dianna Wi	Home Offi	United Sta	Oakland	California	West	OFF-AR-1(	Office Sup	Art	Panasonic	34.58	1	10.0282	#N/A	COD		
8031	CA-2019-1	########	########	Standard C	NM-18520	Neoma M(	Consumer	United Sta	Amarillo	Texas	Central	FUR-FU-1(	Furniture	Furnishings	Executive	23.076	3	-10.9611	#N/A	COD		
8032	CA-2019-1	########	########	Standard C	NM-18520	Neoma M(	Consumer	United Sta	Amarillo	Texas	Central	OFF-PA-10	Office Sup	Paper	Xerox 212	25.92	5	9.072	#N/A	Cards		
6851	US-2019-1	########	########	Standard C	JO-15145	Jack O'Bria	Corporate	United Sta	Franklin	Wisconsin	Central	FUR-BO-1(	Furniture	Bookcases	Atlantic M	1565.88	6	407.1288	#N/A	Online		
6852	US-2019-1	########	########	Standard C	JO-15145	Jack O'Bria	Corporate	United Sta	Franklin	Wisconsin	Central	OFF-BI-10(	Office Sup	Binders	Plastic Bin	106.05	7	49.8435	#N/A	Cards		
7808	US-2019-1	########	########	Standard C	VS-21820	Vivek Sund	Consumer	United Sta	Raleigh	North Car(	South	OFF-BI-10(	Office Sup	Binders	XtraLife Cl	30.828	7	-24.6624	#N/A	Cards		
7809	US-2019-1	########	########	Standard C	VS-21820	Vivek Sund	Consumer	United Sta	Raleigh	North Car(	South	OFF-AR-1(	Office Sup	Art	Newell 31	47.616	3	5.952	#N/A	COD		
7810	US-2019-1	########	########	Standard C	VS-21820	Vivek Sund	Consumer	United Sta	Raleigh	North Car(	South	TEC-PH-1(	Technolog	Phones	Avaya 541	108.784	2	10.8784	#N/A	COD		
3209	CA-2019-1	########	########	Standard C	LA-16780	Laura Arm	Corporate	United Sta	Fresno	California	West	TEC-AC-1(	Technolog	Accessorie	Logitech G	349.95	5	118.983	#N/A	COD		
3210	CA-2019-1	########	########	Standard C	LA-16780	Laura Arm	Corporate	United Sta	Fresno	California	West	TEC-PH-1(	Technolog	Phones	netTALK D	377.928	9	141.723	#N/A	Cards		
3702	CA-2019-1	########	########	Second Cla	MF-17665	Maureen F	Corporate	United Sta	Toledo	Ohio	East	FUR-FU-1(	Furniture	Furnishings	DAX Two-T	15.168	2	3.792	#N/A	Cards		

In [22]:
```
# Insert all require libraries
```

In [1]:
```
import pandas as pd
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
import seaborn as sns
```

In [6]:
```
# read csv file
walmart = pd.read_csv('SuperStore_Sales_Dataset.csv')
```

In [7]:
```
# provide top 5 raws of data
walmart.head()
```

Out[7]:

	Row ID+O6G3A1:R6	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Sub-Category	Product Name	Sales	Quant
0	4918	CA-2019-160304	01-01-2019	07-01-2019	Standard Class	BM-11575	Brendan Murry	Corporate	United States	Gaithersburg	...	Furniture	Bookcases	Bush Westfield Collection Bookcases, Medium Ch...	73.94	
1	4919	CA-2019-160304	02-01-2019	07-01-2019	Standard Class	BM-11575	Brendan Murry	Corporate	United States	Gaithersburg	...	Furniture	Bookcases	Bush Westfield Collection Bookcases, Medium Ch...	173.94	
2	4920	CA-2019-160304	02-01-2019	07-01-2019	Standard Class	BM-11575	Brendan Murry	Corporate	United States	Gaithersburg	...	Technology	Phones	GE 30522EE2	231.98	
3	3074	CA-2019-125206	03-01-2019	05-01-2019	First Class	LR-16915	Lena Radford	Consumer	United States	Los Angeles	...	Office Supplies	Storage	Recycled Steel Personal File for Hanging File ...	114.46	
4	8604	US-2019-116365	03-01-2019	08-01-2019	Standard Class	CA-12310	Christine Abelman	Corporate	United States	San Antonio	...	Technology	Accessories	Imation Clip USB flash drive - 8 GB	30.08	

Data Pre-processing

In [20]:
```
# removing ind1, ind2 column
df = walmart.drop(['ind1','ind2'],axis=1)
```

In [24]:
```
df.head()
```

Out[24]:

	Row ID+O6G3A1:R6	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Product ID	Category	Sub-Category	Pro N
0	4918	CA-2019-160304	01-01-2019	07-01-2019	Standard Class	BM-11575	Brendan Murry	Corporate	United States	Gaithersburg	...	East	FUR-BO-10004709	Furniture	Bookcases	Bookca Me
1	4919	CA-2019-160304	02-01-2019	07-01-2019	Standard Class	BM-11575	Brendan Murry	Corporate	United States	Gaithersburg	...	East	FUR-BO-10004709	Furniture	Bookcases	Bookca Me
2	4920	CA-2019-160304	02-01-2019	07-01-2019	Standard Class	BM-11575	Brendan Murry	Corporate	United States	Gaithersburg	...	East	TEC-PH-10000455	Technology	Phones	30522
3	3074	CA-2019-125206	03-01-2019	05-01-2019	First Class	LR-16915	Lena Radford	Consumer	United States	Los Angeles	...	West	OFF-ST-10003692	Office Supplies	Storage	Pers Fil Har F
4	8604	US-2019-116365	03-01-2019	08-01-2019	Standard Class	CA-12310	Christine Abelman	Corporate	United States	San Antonio	...	Central	TEC-AC-10002217	Technology	Accessories	Imation USB driv

Data Cleaning

Checking For Duplicates

In [31]:
```
# Using Conditional Statement

if df.duplicated().sum()>0:
    print('duplicates are present')
else:
    print('duplicates are not exist')
```

```
duplicates are not exist
```

In [32]:
```
df.duplicated()
```

Out[32]:
```
0       False
1       False
2       False
3       False
4       False
        ...
5896    False
5897    False
5898    False
5899    False
5900    False
Length: 5901, dtype: bool
```

In [33]:
```
df.duplicated(keep=False).sum()
```

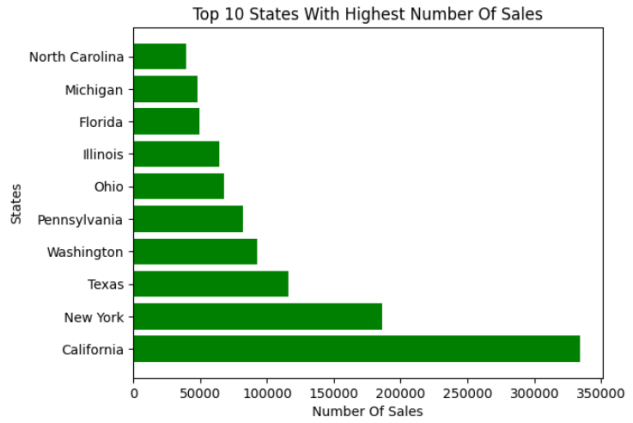
Out[33]:
```
0
```

Exploratory Data Analysis And Power BI

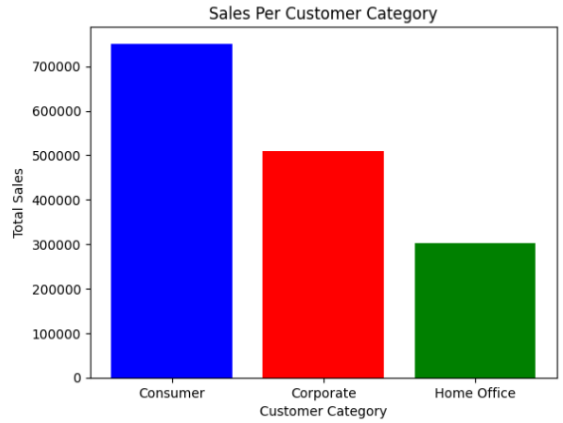
- Power BI is a powerful business analytics tool for visualizing and exploring data.
- EDA techniques such as data visualization, filtering, and drill-down analysis help uncover patterns and trends in sales data.
- Interactive dashboards and reports provide stakeholders with actionable insights in real-time.

In [111]:
```
# create Bar graph

plt.barh(Top_State_Sales['State'],Top_State_Sales['Sales'],color='green')
plt.title("Top 10 States With Highest Number Of Sales")
plt.xlabel("Number Of Sales")
plt.ylabel("States")
plt.show()
```



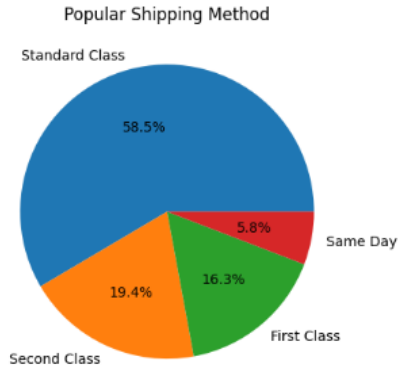
In [48]:
```
plt.bar(sales_per_category['Segment'],sales_per_category['Sales'],color=['blue','red','green'])
plt.title('Sales Per Customer Category')
plt.xlabel('Customer Category')
plt.ylabel('Total Sales')
plt.show()
```



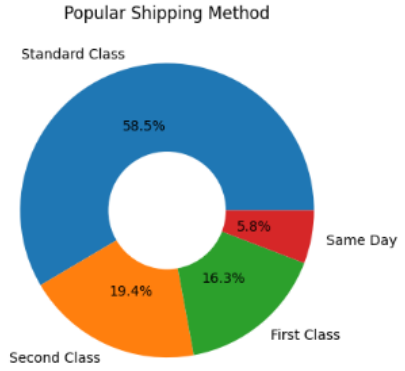
In [55]:
```
df['Ship Mode'].value_counts(normalize=True)
```

Out[55]:
```
Ship Mode
Standard Class    0.584816
Second Class      0.194374
First Class       0.162515
Same Day          0.058295
Name: proportion, dtype: float64
```

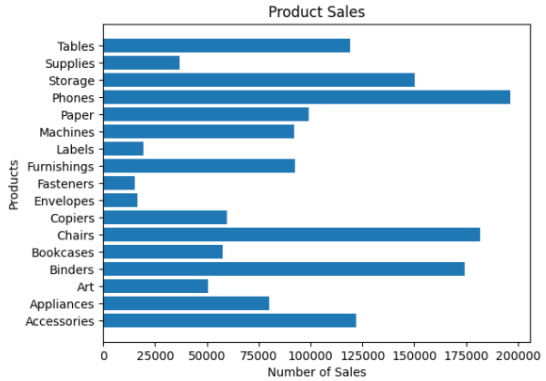
In [59]:
```
# create a pie chart
plt.pie(Shipping_mode['count'],labels=Shipping_mode['Ship Mode'],autopct='%1.1f%%')
plt.title('Popular Shipping Method')
plt.show()
```



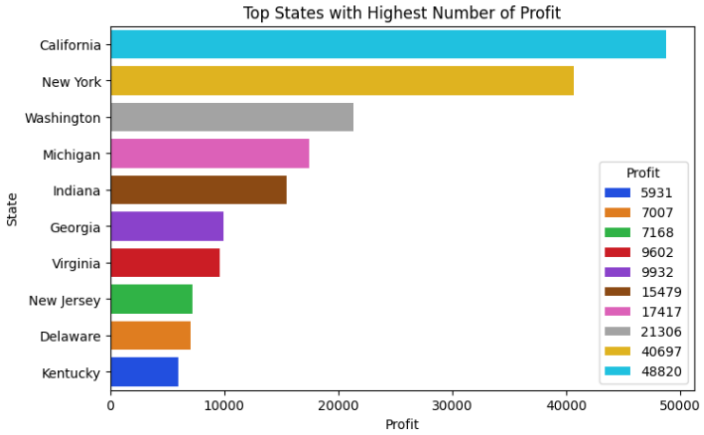
In [63]:
```
# create a donut-pie chart
plt.pie(Shipping_mode['count'],labels=Shipping_mode['Ship Mode'],autopct='%1.1f%%',radius=1)
plt.pie([1],radius=0.4,colors=['w'])
plt.title("Popular Shipping Method")
plt.show()
```



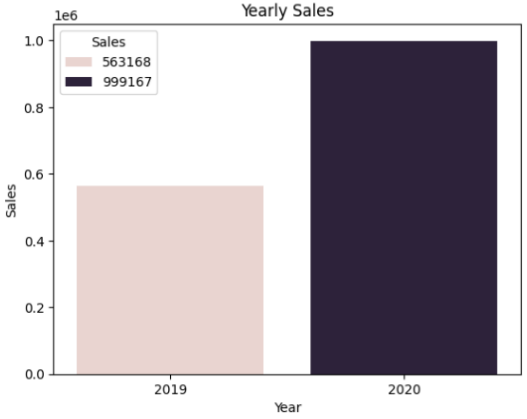
```
In [140]: # plotting bargraph
plt.barh(product_sales['Product'],product_sales['Sales'])
plt.title('Product Sales')
plt.ylabel('Products')
plt.xlabel('Number of Sales')
plt.show()
```



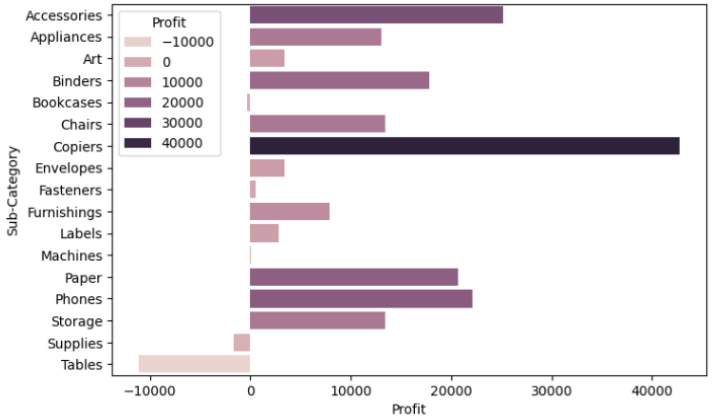

```
In [187]: plt.figure(figsize=(8,5))
sns.barplot(x='Profit',y='State',data=top_state,palette='bright',hue='Profit')
plt.title("Top States with Highest Number of Profit")
plt.show()
```

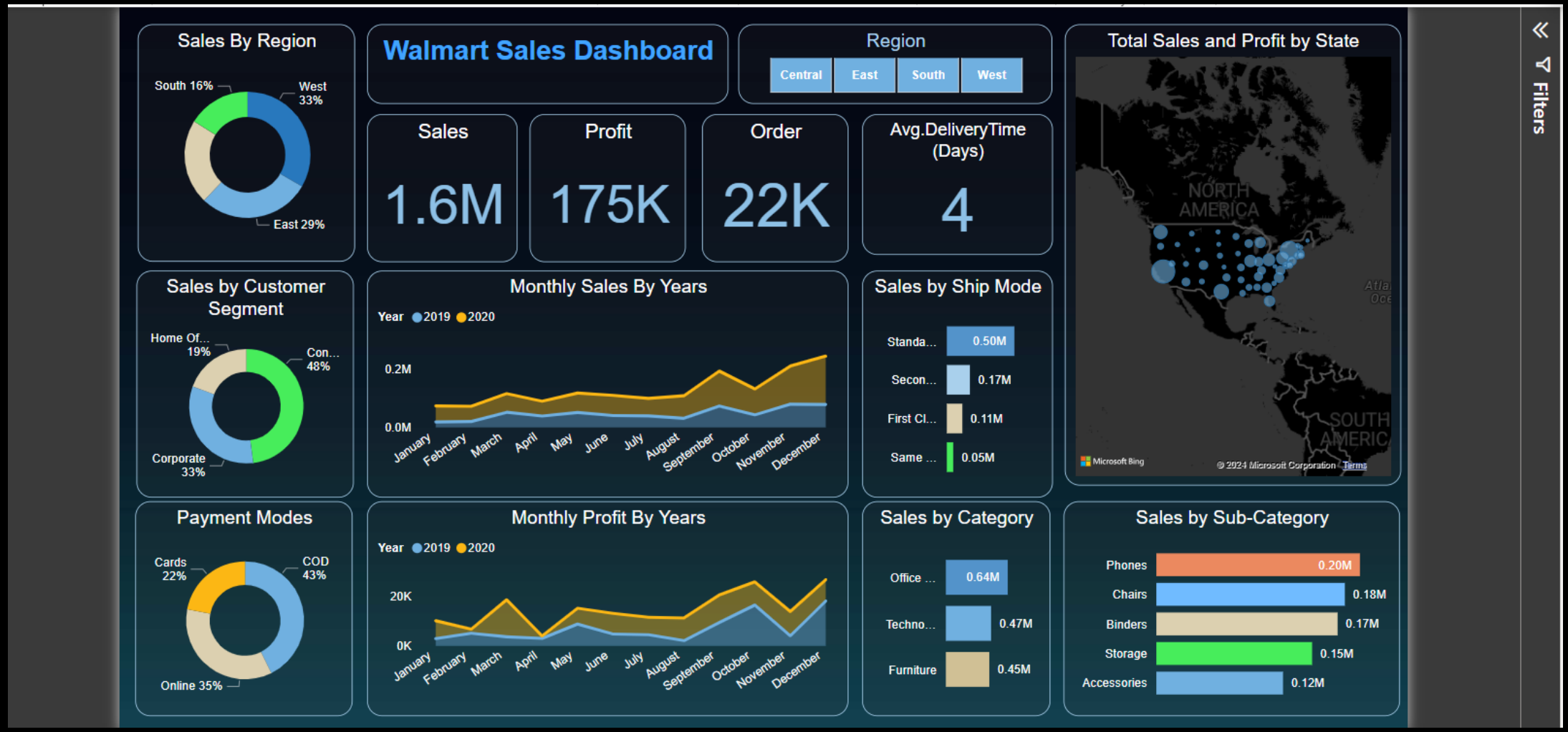



```
In [157]: # plotting bar graph based on yearly sales
sns.barplot(x='Year',y='Sales',data=yearly_sales,hue='Sales')
plt.title('Yearly Sales')
plt.show()
```




```
In [201]: plt.figure(figsize=(8,5))
sns.barplot(x="Profit",y="Sub-Category",data=product_profit,hue="Profit")
plt.show()
```



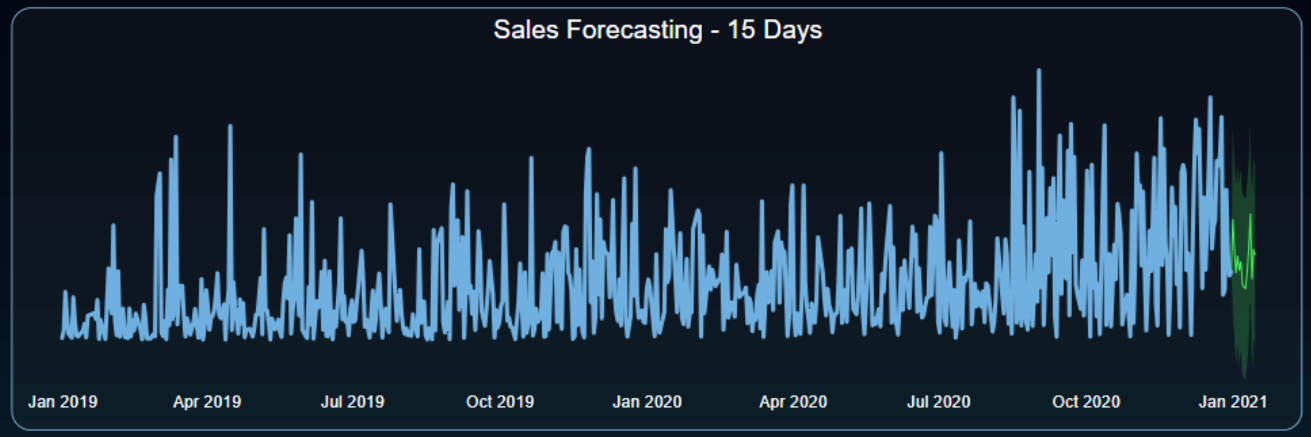

Walmart Sales Dashboard
Region
Central
East
South
West
Sales By Region
South 16%
West 33%
East 29%
Sales
1.6M
Profit
175K
Order
22K
Avg.DeliveryTime (Days)
4
Total Sales and Profit by State
NORTH AMERICA
SOUTH AMERIC
Microsoft Bing
© 2024 Microsoft Corporation Terms
Sales by Customer Segment
Home Of... 19%
Con... 48%
Corporate 33%
Monthly Sales By Years
Year 2019 2020
0.2M
0.0M
January February March April May June July August September October November December
Sales by Ship Mode
Standa... 0.50M
Secon... 0.17M
First Cl... 0.11M
Same ... 0.05M
Payment Modes
Cards 22%
COD 43%
Online 35%
Monthly Profit By Years
Year 2019 2020
20K
0K
January February March April May June July August September October November December
Sales by Category
Office ... 0.64M
Techno... 0.47M
Furniture 0.45M
Sales by Sub-Category
Phones 0.20M
Chairs 0.18M
Binders 0.17M
Storage 0.15M
Accessories 0.12M
Filters


GLOBAL SALES DASHBOARD
Quarter
Select all
Qtr 1
Qtr 2
Qtr 3
Qtr 4
Country
All
Year
All
Month
All
Orders
178K
Sales
13M
Profit
1M
Ship Days
4
Sales by Country
Sales by Sub-Category
Phones 13.5%
Copiers 11.94%
Chairs 11.8...
Book... 11.6%
Stora... 8.91%
Ap... 7....
T... ...
A... 5...
Fur... 3....
Sales by Region
Western ... 13.7%
C... ...
Ocea... 8.7%
Sout... 6.85%
Easter... 6.76%
Ea... 5....
Sout... 4.88%
Centr... 3.96%
East... 2.4...
Sales by Segment
Consum... 51.48%
Corporate 30.25%
Home ... 18.27%
Sales by Month and Year
Year 2012 2013 2014 2015
2M
1M
0M
January February March April May June July August September October November December
Sales by Ship Mode
Standard Class 7.6M
Second Class 2.6M
First Class 1.8M
Same Day 0.7M
Profit by Month and Year
Year 2012 2013 2014 2015
0.2M
0.1M
0.0M
January February March April May June July August September October November December
Sales by Category
Technology 4.7M
Furniture 4.1M
Office Supplies 3.8M
Sales by Market
Asia Pacific 4.0M
Europe 3.3M
USCA 2.4M
LATAM 2.2M
Africa 0.8M
Atlantic Ocean
Indian Ocean
ANTARCTICA
Microsoft Bing
© 2024 Microsoft Corporation Terms

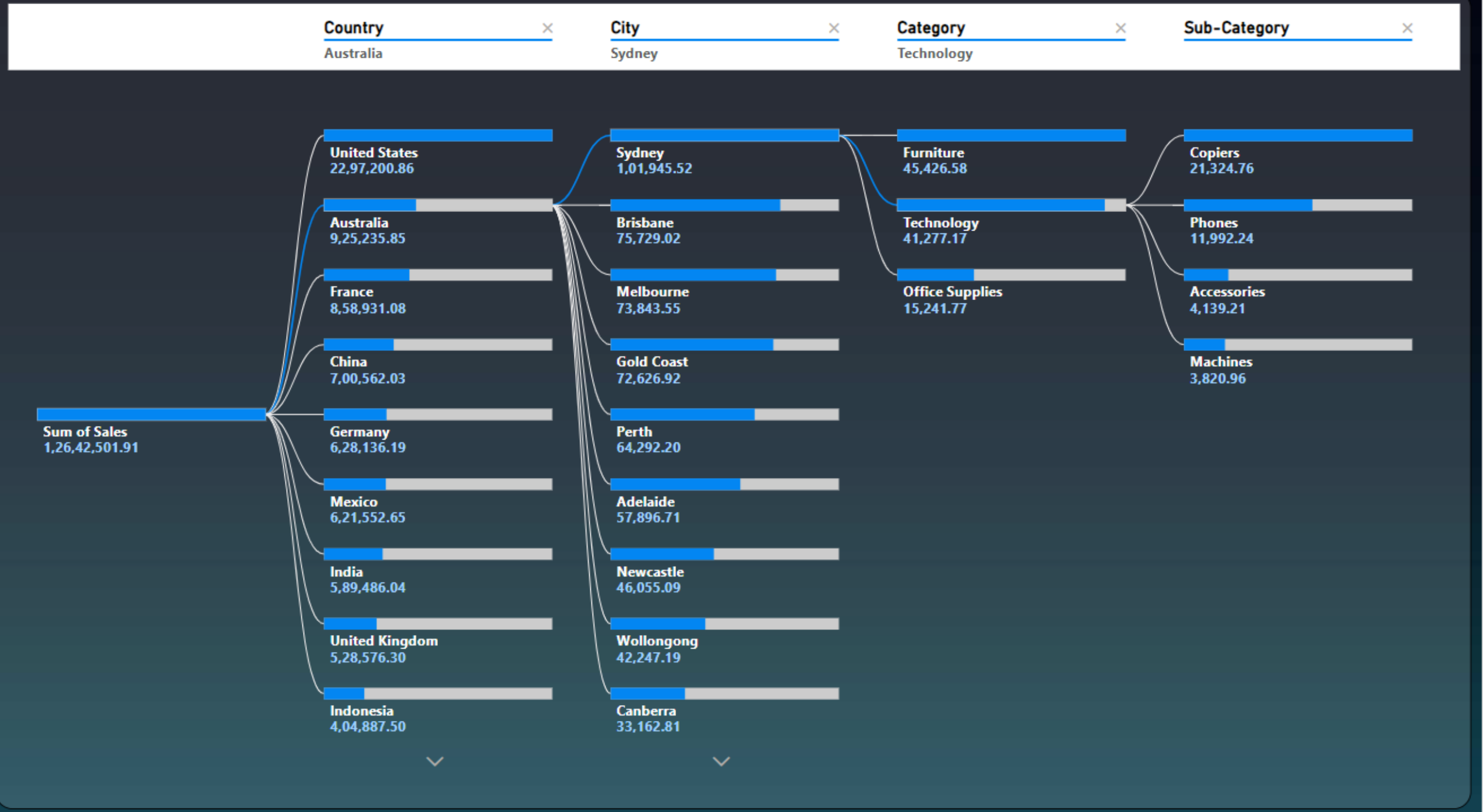
Time-Series Analysis For Sales Forecasting

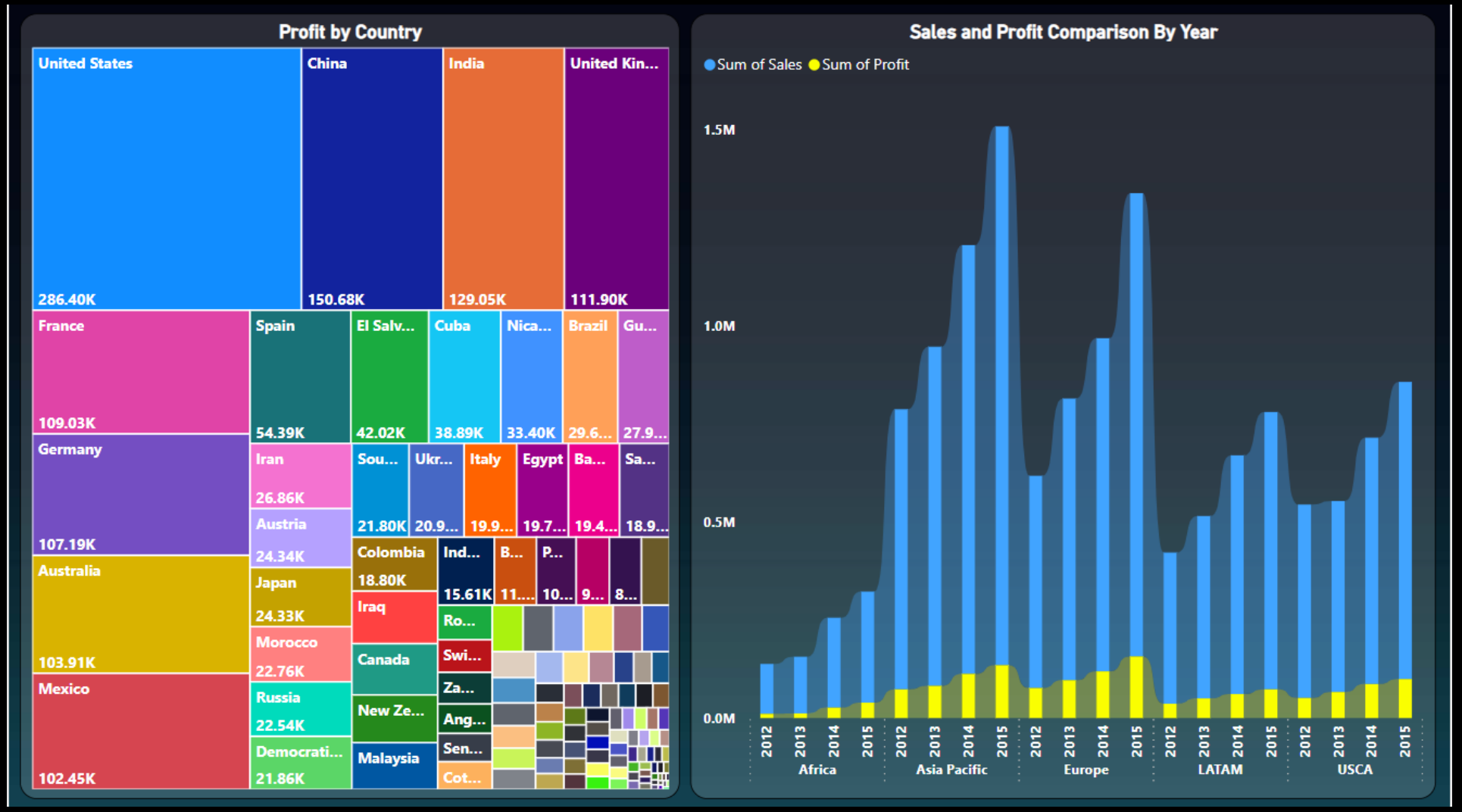
- Time series analysis is a statistical technique used to analyze temporal data and make predictions.
- By analyzing historical sales data, businesses can forecast future sales trends and demand patterns.
- Time series models such as ARIMA (Auto Regressive Integrated Moving Average) and Exponential Smoothing are commonly used for sales forecasting.

Walmart Sales Forecast - 15 Days



Filters


Country
Australia
City
Sydney
Category
Technology
Sub-Category
Sum of Sales
1,26,42,501.91
United States
22,97,200.86
Australia
9,25,235.85
France
8,58,931.08
China
7,00,562.03
Germany
6,28,136.19
Mexico
6,21,552.65
India
5,89,486.04
United Kingdom
5,28,576.30
Indonesia
4,04,887.50
Sydney
1,01,945.52
Brisbane
75,729.02
Melbourne
73,843.55
Gold Coast
72,626.92
Perth
64,292.20
Adelaide
57,896.71
Newcastle
46,055.09
Wollongong
42,247.19
Canberra
33,162.81
Furniture
45,426.58
Technology
41,277.17
Office Supplies
15,241.77
Copiers
21,324.76
Phones
11,992.24
Accessories
4,139.21
Machines
3,820.96


Profit by Country
United States
286.40K
China
150.68K
India
129.05K
United Kin...
111.90K
France
109.03K
Spain
54.39K
El Salv...
42.02K
Cuba
38.89K
Nica...
33.40K
Brazil
29.6...
Gu...
27.9...
Germany
107.19K
Iran
26.86K
Sou...
21.80K
Ukr...
20.9...
Italy
19.9...
Egypt
19.7...
Ba...
19.4...
Sa...
18.9...
Australia
103.91K
Austria
24.34K
Colombia
18.80K
Ind...
15.61K
B...
11....
P...
10...
9...
8...
Japan
24.33K
Iraq
Ro...
Mexico
102.45K
Morocco
22.76K
Canada
Swi...
Russia
22.54K
Za...
New Ze...
Ang...
Democrati...
21.86K
Malaysia
Sen...
Cot...
Sales and Profit Comparison By Year
Sum of Sales
Sum of Profit
1.5M
1.0M
0.5M
0.0M
2012
2013
2014
2015
Africa
Asia Pacific
Europe
LATAM
USCA

Actionable Insights and Decision Making

- The insights generated from data analysis and time series forecasting empower businesses to make informed decisions.
- By understanding sales trends and forecasting future sales, businesses can optimize pricing strategies, allocate resources efficiently, and identify growth opportunities.

Conclusion

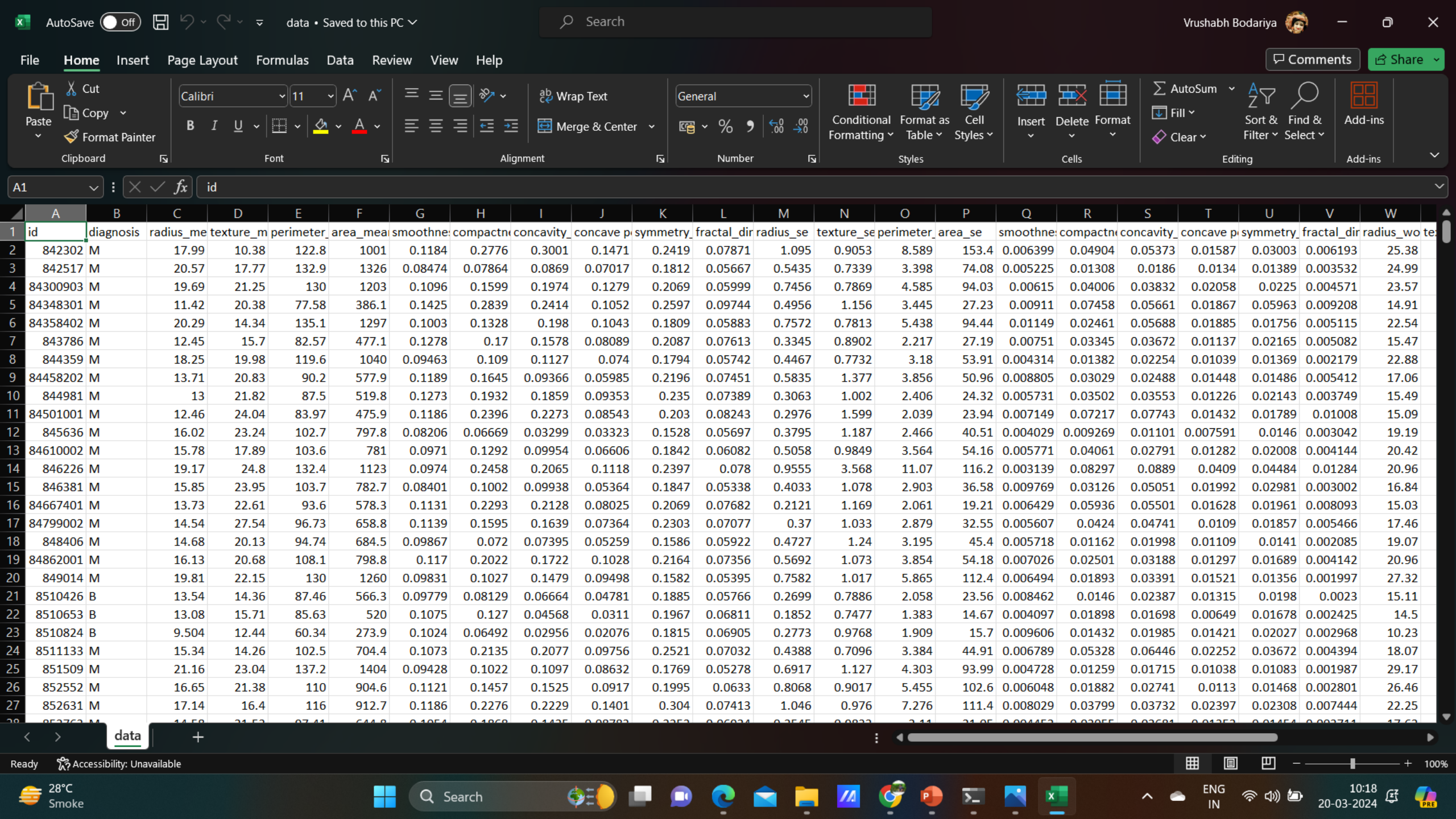
- Data analysis, Power BI and time series analysis are powerful tools for uncovering sales insights and forecasting future sales.
- By harnessing the power of data-driven insights, businesses can gain a competitive edge, drive growth, and enhance overall performance.
- Continued investment in data analytics capabilities and technology is essential for staying ahead in today's dynamic business environment.

Breast Cancer Prediction Using Data Science

- Welcome to the presentation on Breast Cancer Prediction Using Data Science.
- Breast cancer is a prevalent disease affecting women worldwide, making early detection crucial.
- This presentation explores the application of data science techniques for predicting breast cancer.

Data Collection

- Gathering relevant data is crucial for building an accurate predictive model.
- Data sources may include patient demographics, medical history, genetic factors, and imaging results.
- High-quality, diverse datasets enhance the reliability and effectiveness of predictive models.


AutoSave Off
data • Saved to this PC
Search
Vrushabh Bodariya
File Home Insert Page Layout Formulas Data Review View Help
Comments Share
Paste Cut Copy Format Painter
Calibri 11
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Cell Styles
Insert Delete Format
AutoSum Fill Clear
Sort & Filter Find & Select
Add-ins
Clipboard Font Alignment Number Styles Cells Editing Add-ins
A1
id
id diagnosis radius_me texture_m perimeter_ area_mea smoothne compactn concavity_ concave p symmetry_ fractal_dir radius_se texture_se perimeter_ area_se smoothne compactn concavity_ concave p symmetry_ fractal_dir radius_wo
842302 M 17.99 10.38 122.8 1001 0.1184 0.2776 0.3001 0.1471 0.2419 0.07871 1.095 0.9053 8.589 153.4 0.006399 0.04904 0.05373 0.01587 0.03003 0.006193 25.38
842517 M 20.57 17.77 132.9 1326 0.08474 0.07864 0.0869 0.07017 0.1812 0.05667 0.5435 0.7339 3.398 74.08 0.005225 0.01308 0.0186 0.0134 0.01389 0.003532 24.99
84300903 M 19.69 21.25 130 1203 0.1096 0.1599 0.1974 0.1279 0.2069 0.05999 0.7456 0.7869 4.585 94.03 0.00615 0.04006 0.03832 0.02058 0.0225 0.004571 23.57
84348301 M 11.42 20.38 77.58 386.1 0.1425 0.2839 0.2414 0.1052 0.2597 0.09744 0.4956 1.156 3.445 27.23 0.00911 0.07458 0.05661 0.01867 0.05963 0.009208 14.91
84358402 M 20.29 14.34 135.1 1297 0.1003 0.1328 0.198 0.1043 0.1809 0.05883 0.7572 0.7813 5.438 94.44 0.01149 0.02461 0.05688 0.01885 0.01756 0.005115 22.54
843786 M 12.45 15.7 82.57 477.1 0.1278 0.17 0.1578 0.08089 0.2087 0.07613 0.3345 0.8902 2.217 27.19 0.00751 0.03345 0.03672 0.01137 0.02165 0.005082 15.47
844359 M 18.25 19.98 119.6 1040 0.09463 0.109 0.1127 0.074 0.1794 0.05742 0.4467 0.7732 3.18 53.91 0.004314 0.01382 0.02254 0.01039 0.01369 0.002179 22.88
84458202 M 13.71 20.83 90.2 577.9 0.1189 0.1645 0.09366 0.05985 0.2196 0.07451 0.5835 1.377 3.856 50.96 0.008805 0.03029 0.02488 0.01448 0.01486 0.005412 17.06
844981 M 13 21.82 87.5 519.8 0.1273 0.1932 0.1859 0.09353 0.235 0.07389 0.3063 1.002 2.406 24.32 0.005731 0.03502 0.03553 0.01226 0.02143 0.003749 15.49
84501001 M 12.46 24.04 83.97 475.9 0.1186 0.2396 0.2273 0.08543 0.203 0.08243 0.2976 1.599 2.039 23.94 0.007149 0.07217 0.07743 0.01432 0.01789 0.01008 15.09
845636 M 16.02 23.24 102.7 797.8 0.08206 0.06669 0.03299 0.03323 0.1528 0.05697 0.3795 1.187 2.466 40.51 0.004029 0.009269 0.01101 0.007591 0.0146 0.003042 19.19
84610002 M 15.78 17.89 103.6 781 0.0971 0.1292 0.09954 0.06606 0.1842 0.06082 0.5058 0.9849 3.564 54.16 0.005771 0.04061 0.02791 0.01282 0.02008 0.004144 20.42
846226 M 19.17 24.8 132.4 1123 0.0974 0.2458 0.2065 0.1118 0.2397 0.078 0.9555 3.568 11.07 116.2 0.003139 0.08297 0.0889 0.0409 0.04484 0.01284 20.96
846381 M 15.85 23.95 103.7 782.7 0.08401 0.1002 0.09938 0.05364 0.1847 0.05338 0.4033 1.078 2.903 36.58 0.009769 0.03126 0.05051 0.01992 0.02981 0.003002 16.84
84667401 M 13.73 22.61 93.6 578.3 0.1131 0.2293 0.2128 0.08025 0.2069 0.07682 0.2121 1.169 2.061 19.21 0.006429 0.05936 0.05501 0.01628 0.01961 0.008093 15.03
84799002 M 14.54 27.54 96.73 658.8 0.1139 0.1595 0.1639 0.07364 0.2303 0.07077 0.37 1.033 2.879 32.55 0.005607 0.0424 0.04741 0.0109 0.01857 0.005466 17.46
848406 M 14.68 20.13 94.74 684.5 0.09867 0.072 0.07395 0.05259 0.1586 0.05922 0.4727 1.24 3.195 45.4 0.005718 0.01162 0.01998 0.01109 0.0141 0.002085 19.07
84862001 M 16.13 20.68 108.1 798.8 0.117 0.2022 0.1722 0.1028 0.2164 0.07356 0.5692 1.073 3.854 54.18 0.007026 0.02501 0.03188 0.01297 0.01689 0.004142 20.96
849014 M 19.81 22.15 130 1260 0.09831 0.1027 0.1479 0.09498 0.1582 0.05395 0.7582 1.017 5.865 112.4 0.006494 0.01893 0.03391 0.01521 0.01356 0.001997 27.32
8510426 B 13.54 14.36 87.46 566.3 0.09779 0.08129 0.06664 0.04781 0.1885 0.05766 0.2699 0.7886 2.058 23.56 0.008462 0.0146 0.02387 0.01315 0.0198 0.0023 15.11
8510653 B 13.08 15.71 85.63 520 0.1075 0.127 0.04568 0.0311 0.1967 0.06811 0.1852 0.7477 1.383 14.67 0.004097 0.01898 0.01698 0.00649 0.01678 0.002425 14.5
8510824 B 9.504 12.44 60.34 273.9 0.1024 0.06492 0.02956 0.02076 0.1815 0.06905 0.2773 0.9768 1.909 15.7 0.009606 0.01432 0.01985 0.01421 0.02027 0.002968 10.23
8511133 M 15.34 14.26 102.5 704.4 0.1073 0.2135 0.2077 0.09756 0.2521 0.07032 0.4388 0.7096 3.384 44.91 0.006789 0.05328 0.06446 0.02252 0.03672 0.004394 18.07
851509 M 21.16 23.04 137.2 1404 0.09428 0.1022 0.1097 0.08632 0.1769 0.05278 0.6917 1.127 4.303 93.99 0.004728 0.01259 0.01715 0.01038 0.01083 0.001987 29.17
852552 M 16.65 21.38 110 904.6 0.1121 0.1457 0.1525 0.0917 0.1995 0.0633 0.8068 0.9017 5.455 102.6 0.006048 0.01882 0.02741 0.0113 0.01468 0.002801 26.46
852631 M 17.14 16.4 116 912.7 0.1186 0.2276 0.2229 0.1401 0.304 0.07413 1.046 0.976 7.276 111.4 0.008029 0.03799 0.03732 0.02397 0.02308 0.007444 22.25
data
Ready Accessibility: Unavailable
100%
28°C Smoke
Search
ENG IN
10:18 20-03-2024

Data Pre-processing & Data Cleaning

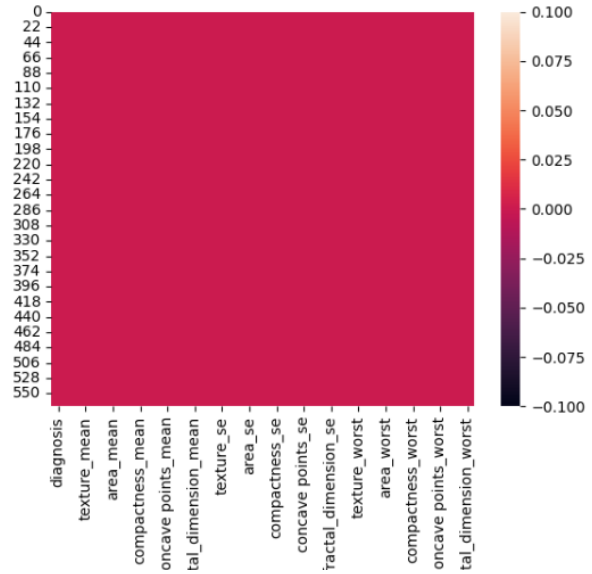
Handling Missing Values: Identify and handle missing values appropriately, either by imputation (replacing missing values with a calculated value) or deletion.

Removing Duplicates: Detect and remove duplicate entries in the dataset to ensure data integrity.

Correcting Inconsistent Data: Address inconsistencies such as typos, mislabeled categories, or erroneous entries to maintain data accuracy.

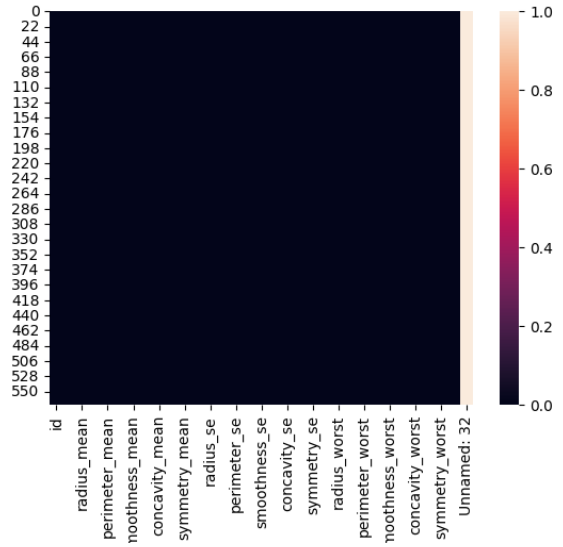
Dealing with Outliers: Identify and handle outliers that may skew the analysis results, either by removing them or transforming them.

```
In [14]: sns.heatmap(data.isnull())
Out[14]: <Axes: >
```



Data Pre-processing and Data Cleaning

```
In [9]: sns.heatmap(data.isnull())
Out[9]: <Axes: >
```




```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,classification_report
import pickle


def get_clean_data():
    data = pd.read_csv('data.csv')
    # clean the data
    data=data.drop(['Unnamed: 32','id'],axis=1)
    data['diagnosis'] = data['diagnosis'].map({'M':1,'B':0})
    print(data.info())
    return data

```

```
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
```

Model Training and Evaluation

- Machine learning models, such as logistic regression or linear regression, are trained on the preprocessed data.
- Model performance is evaluated using metrics like accuracy, precision.
- Streamlit can display these metrics in real-time, allowing users to understand the model's performance.

Evaluation of the model

In [43]:
```
from sklearn.metrics import accuracy_score
```

In [45]:
```
Accuracy = accuracy_score(y_test,y_predict)
```

In [46]:
```
Accuracy
```

Out[46]:
```
0.9824561403508771
```

In [47]:
```
#Classification report
from sklearn.metrics import classification_report
```

In [49]:
```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       108
           1       1.00      0.95      0.98        63

    accuracy                           0.98       171
   macro avg       0.99      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```

```
16
17  def create_model(data):
18      x=data.drop('diagnosis',axis=1)  # Independent variable
19      y=data['diagnosis'] #dependent variable
20
21      # scale the data
22      scaler = StandardScaler()
23      x=scaler.fit_transform(x)
24
25      # split the data
26      x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.4)
27
28      # train the data
29      model = LogisticRegression()
30      model.fit(x_train,y_train)
31
32      # test the model
33      y_predict = model.predict(x_test)
34      print('Accuracy of model :',accuracy_score(y_test,y_predict))
35      print('Classification report :',classification_report(y_test,y_predict))
36
37      return model,scaler
38
```

Normalize the data

In [27]:
```
# Scikit-Learn
from sklearn.preprocessing import StandardScaler

# create a scaler object
scaler = StandardScaler()

# fit the scaler to the data and transform the data
x_scaled = scaler.fit_transform(x)
```

In [28]:
```
x_scaled
```

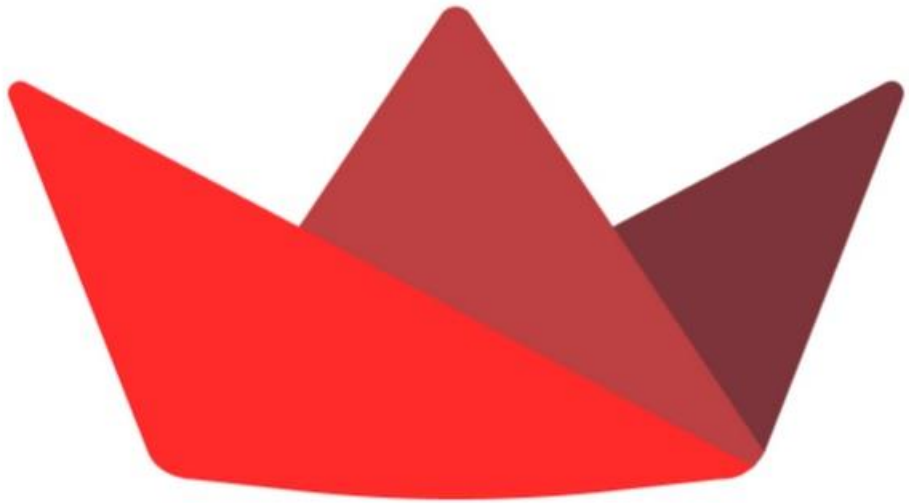
Out[28]:
```
array([[ 1.09706398, -2.07333501,  1.26993369, ...,  2.29607613,
         2.75062224,  1.93701461],
       [ 1.82982061, -0.35363241,  1.68595471, ...,  1.0870843 ,
        -0.24388967,  0.28118999],
       [ 1.57988811,  0.45618695,  1.56650313, ...,  1.95500035,
         1.152255  ,  0.20139121],
       ...,
       [ 0.70228425,  2.0455738 ,  0.67267578, ...,  0.41406869,
        -1.10454895, -0.31840916],
       [ 1.83834103,  2.33645719,  1.98252415, ...,  2.28998549,
         1.91908301,  2.21963528],
       [-1.80840125,  1.22179204, -1.81438851, ..., -1.74506282,
        -0.04813821, -0.75120669]])
```

```
None
Accuracy of model : 0.9736842105263158
Classification report :              precision    recall  f1-score   support

           0       0.97      0.99      0.98       148
           1       0.97      0.95      0.96        80

    accuracy                           0.97       228
   macro avg       0.97      0.97      0.97       228
weighted avg       0.97      0.97      0.97       228

PS E:\Internship\Breast Cancer Predictor>
```



Building The Streamlit Application

- Streamlit applications are built using simple Python scripts.
- Developers can add widgets like sliders, dropdowns, and buttons to create interactive elements.
- Streamlit's reactive framework automatically updates the interface based on user input or changes in the underlying data.

Go Run ··· Breast Cancer Predictor

main.py | app.py | data.csv | style.css

app.py > main

```python
import streamlit as st
import pickle
import pandas as pd
import plotly.graph_objects as go
import numpy as np

def get_clean_data():
    data = pd.read_csv('data.csv')
    # clean the data
    data=data.drop(['Unnamed: 32','id'],axis=1)
    data['diagnosis'] = data['diagnosis'].map({'M':1,'B':0})
    print(data.info())
    return data

```

```python
def add_predictions(input_data):
    model = pickle.load(open('E:\Internship\Breast Cancer Predictor\model.pkl','rb'))
    scaler = pickle.load(open("E:\Internship\Breast Cancer Predictor\scaler.pkl","rb"))

    # we convert input_data dictionary in array (in two dimensional)
    input_array = np.array(list(input_data.values())).reshape(1,-1)

    # Scale the input_array using scaler
    input_array_scaled = scaler.transform(input_array)

    st.subheader("Cell Cluster Prediction")
    st.write('The cell cluster is:')

    # now we make prediction using input_array_scaled
    prediction = model.predict(input_array_scaled)
    if prediction[0]==0:
        st.write("<span class='diagnosis benign'>Benign</span>", unsafe_allow_html=True)
    else:
        st.write("<span class='diagnosis malignant'>Malignant</span>", unsafe_allow_html=True)

    # probability values for prediction(benign or malignant)
    st.write('Probability of being Benign:',model.predict_proba(input_array_scaled)[0][0])
    st.write('Probability of being Malignant:',model.predict_proba(input_array_scaled)[0][1])

    st.write('This app can assist medical professionals in making a diagnosis, but should not be used as a substitute for a professional
```

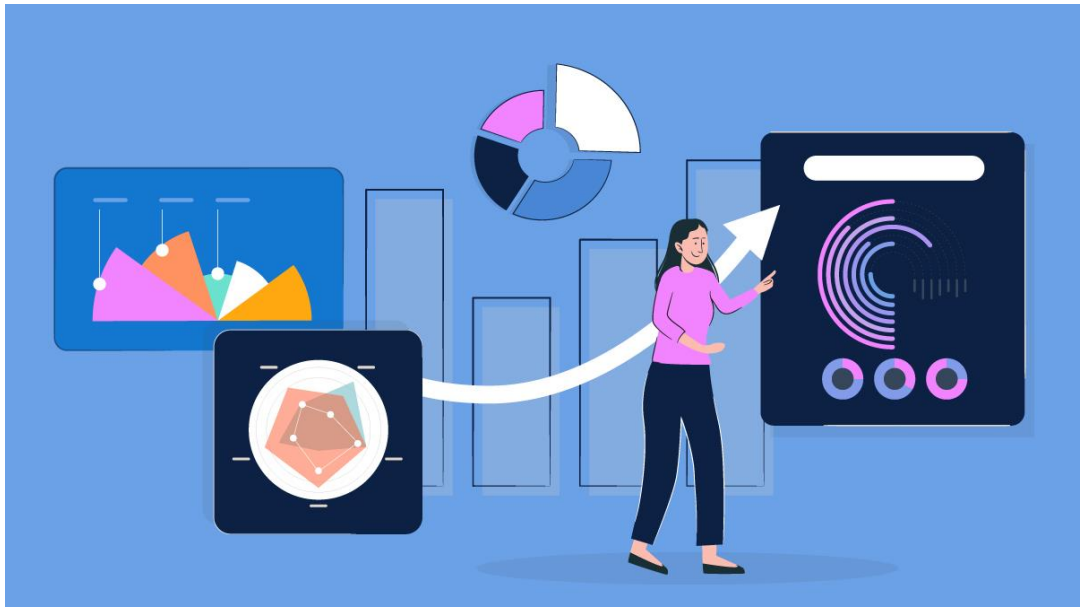
```python

data = get_clean_data()

slider_labels = [
    ("Radius (mean)", "radius_mean"),
    ("Texture (mean)", "texture_mean"),
    ("Perimeter (mean)", "perimeter_mean"),
    ("Area (mean)", "area_mean"),
    ("Smoothness (mean)", "smoothness_mean"),
    ("Compactness (mean)", "compactness_mean"),
    ("Concavity (mean)", "concavity_mean"),
    ("Concave points (mean)", "concave points_mean"),
    ("Symmetry (mean)", "symmetry_mean"),
    ("Fractal dimension (mean)", "fractal_dimension_mean"),
    ("Radius (se)", "radius_se"),
    ("Texture (se)", "texture_se"),
    ("Perimeter (se)", "perimeter_se"),
    ("Area (se)", "area_se"),
    ("Smoothness (se)", "smoothness_se"),
    ("Compactness (se)", "compactness_se"),
    ("Concavity (se)", "concavity_se"),
    ("Concave points (se)", "concave points_se"),
    ("Symmetry (se)", "symmetry_se"),
    ("Fractal dimension (se)", "fractal_dimension_se"),
    ("Radius (worst)", "radius_worst"),
    ("Texture (worst)", "texture_worst"),
    ("Perimeter (worst)", "perimeter_worst"),
    ("Area (worst)", "area_worst"),
    ("Smoothness (worst)", "smoothness_worst"),
    ("Compactness (worst)", "compactness_worst"),
    ("Concavity (worst)", "concavity_worst"),
    ("Concave points (worst)", "concave points_worst"),
    ("Symmetry (worst)", "symmetry_worst"),
    ("Fractal dimension (worst)", "fractal_dimension_worst"),
]

input_dict = {}

for label,key in slider_labels:
    input_dict[key]=st.sidebar.slider(
```



Visualizing Predictions

- Streamlit enables developers to visualize predictions using plots, charts, and tables.
- Interactive visualizations help users understand the factors influencing the predicted outcome.
- Streamlit's built-in support for popular plotting libraries like Matplotlib and Plotly makes it easy to create engaging visualizations.

```python
def get_scaled_values(input_dict):
    data = get_clean_data()

    X = data.drop(['diagnosis'], axis=1)

    scaled_dict = {}

    for key, value in input_dict.items():
        max_val = X[key].max()
        min_val = X[key].min()
        scaled_value = (value - min_val) / (max_val - min_val)
        scaled_dict[key] = scaled_value

    return scaled_dict
```

```python
def main():
    st.set_page_config(
        page_title='Breast Cancer Predictor',
        page_icon=':Female Doctor:',
        layout='wide',
        initial_sidebar_state='expanded'
    )

    with open("E:\Internship\Breast Cancer Predictor\style.css") as f:
        st.markdown("<style>{}</style>".format(f.read()), unsafe_allow_html=True)
    input_data = add_sidebar()

    #st.write(input_data)

    with st.container():
        st.title("Breast Cancer Predictor")
        st.write("Please connect this app to your cytology lab to help diagnose breast cancer from your tissue sample. This app predicts using a mac

    col1,col2 = st.columns([15,10])

    with col1:
        radar_chart = get_radar_chart(input_data)
        st.plotly_chart(radar_chart)
    with col2:
        add_predictions(input_data)


if __name__=='__main__':
    main()
```

```python
def get_radar_chart(input_data):

    input_data = get_scaled_values(input_data)

    categories = ['Radius', 'Texture', 'Perimeter', 'Area',
                  'Smoothness', 'Compactness',
                  'Concavity', 'Concave Points',
                  'Symmetry', 'Fractal Dimension']

    fig = go.Figure()

    fig.add_trace(go.Scatterpolar(
          r=[
            input_data['radius_mean'], input_data['texture_mean'], input_data['perimeter_mean'],
            input_data['area_mean'], input_data['smoothness_mean'], input_data['compactness_mean'],
            input_data['concavity_mean'], input_data['concave points_mean'], input_data['symmetry_mean'],
            input_data['fractal_dimension_mean']
          ],
          theta=categories,
          fill='toself',
          name='Mean Value'
    ))
    fig.add_trace(go.Scatterpolar(
          r=[
            input_data['radius_se'], input_data['texture_se'], input_data['perimeter_se'], input_data['area_se'],
            input_data['smoothness_se'], input_data['compactness_se'], input_data['concavity_se'],
            input_data['concave points_se'], input_data['symmetry_se'],input_data['fractal_dimension_se']
          ],
          theta=categories,
          fill='toself',
          name='Standard Error'
    ))
    fig.add_trace(go.Scatterpolar(
          r=[
            input_data['radius_worst'], input_data['texture_worst'], input_data['perimeter_worst'],
            input_data['area_worst'], input_data['smoothness_worst'], input_data['compactness_worst'],
            input_data['concavity_worst'], input_data['concave points_worst'], input_data['symmetry_worst'],
            input_data['fractal_dimension_worst']
          ],
          theta=categories,
          fill='toself',
          name='Worst Value'
    ))

    fig.update_layout(
      polar=dict(
        radialaxis=dict(
          visible=True,
          range=[0, 1]
        )),
      showlegend=True
    )
```

Deploy

Cell Nucleus Measurements

Radius (mean)
14.13
0.00 28.11

Texture (mean)
19.29
0.00 39.28

Perimeter (mean)
91.97
0.00 188.50

Area (mean)
654.89
0.00 2501.00

Smoothness (mean)
0.10
0.00 0.16

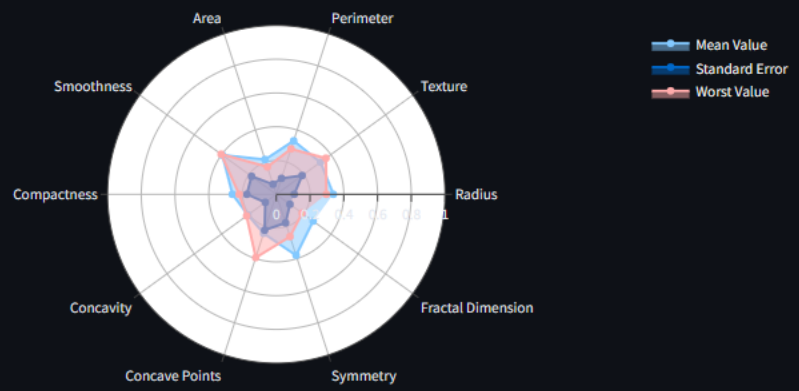
Compactness (mean)
0.10
0.00 0.35

Concavity (mean)
0.09
0.00 0.43

Concave points (mean)
0.05
0.00 0.20

Breast Cancer Predictor

Please connect this app to your cytology lab to help diagnose breast cancer from your tissue sample. This app predicts using a machine learning model whether a breast mass is benign or malignant based on the measurements it receives from your cytosis lab. You can also update the measurements by hand using the sliders in the sidebar.



Cell Cluster Prediction

The cell cluster is:

Benign

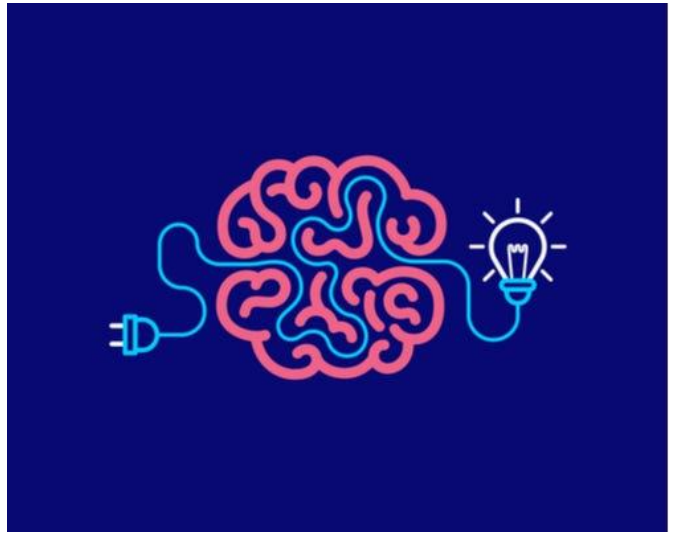
Probability of being Benign: 0.5596160667083172

Probability of being Malignant: 0.44038393329168274

This app can assist medical professionals in making a diagnosis, but should not be used as a substitute for a professional diagnosis.

Conclusion

- Streamlit provides a powerful platform for developing interactive web applications for breast cancer prediction.
- By combining data science techniques with Streamlit's intuitive interface, developers can create user-friendly tools for healthcare professionals and patients.
- Continued research and innovation in this field hold the promise of improving early detection and treatment of breast cancer.



Tools & Technology Used In Projects

- Visual Studio (Vs Code)
- Python (with Different libraries)
- Machine Learning
- Streamlit
- Power BI
- Jupyter Notebook
- Kaggle (for datasets)

vrushabhbodaryaait@gmail.com