Applied Data Science Capstone

adel

TOPICS

- DATASET STOCK DATA
- ANALYSIS AUTOMOBILE
- DATASET DASHBOARD WITH PLOTLY AND DASH HOUSING PRICE
- CREATE A
  CODING SUMMARY

INTRODUCTION

- Inthis presentation, we will explore insights from various datasets, accompanied by captivating Plotlygraphs and their dashboard.
- Initiation of course (2 JAN 2024)

Lets learn

- Courses like data science offer invaluable skills and knowledge that empower individuals to:
- Analyze complex data sets to extract meaningful insights.
- Make informed decisions based on data-driven evidence.
- Develop predictive models to forecast future trends and outcomes.
- Utilize advanced statistical techniques to uncover patterns and correlations.
- Communicate findings effectively through data visualization and storytelling.
- Navigate and manipulate large datasets using cutting-edge tools and technologies.

Question 1 -Extracting Tesla Stock Data Using yfinance

Question 2 -Extracting Tesla Revenue Data Using Webscraping-

Question 3 -Extracting GameStop Stock Data Using yfinance-

Question 4 -Extracting GameStop Revenue Data Using Webscraping-

Question 5 -Tesla Stock and Revenue Dashboard -

Question 6 -GameStop Stock and Revenue Dashboard

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is Tesla and its ticker symbol is `TSLA`.

```
tesla = yf.Ticker("TSLA")
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `tesla_data`. Set the `period` parameter to `max` so we get information for the maximum amount of time.

```
tesla_data = tesla.history(period="max")
```

Reset the index, save, and display the first five rows of the `tesla_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

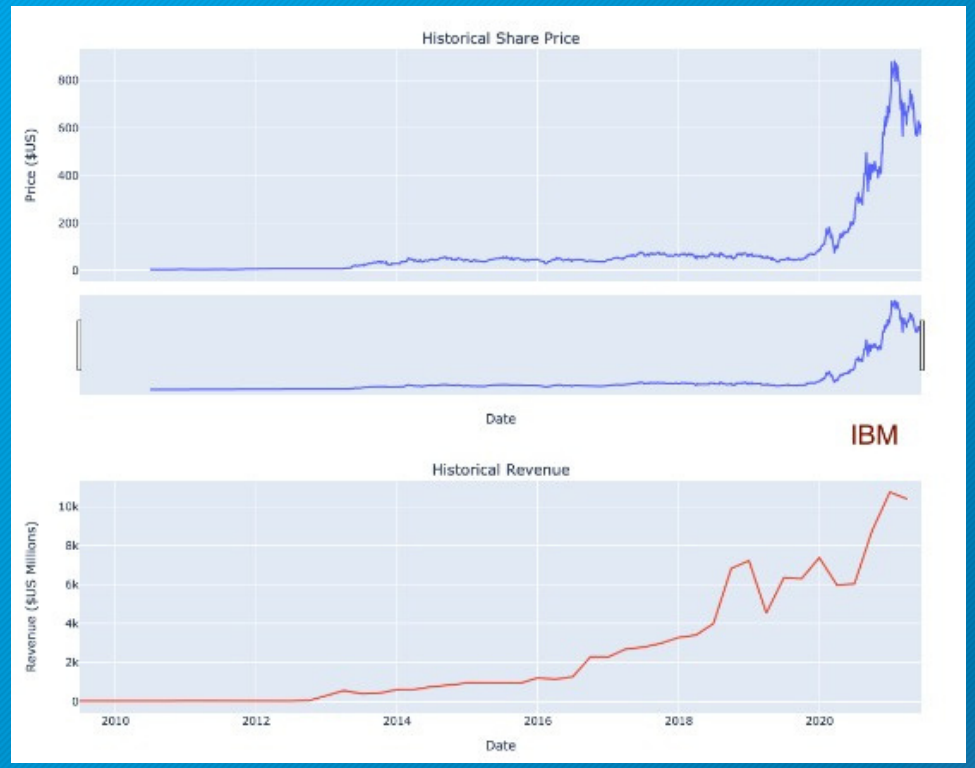
	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
1	2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2	2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
3	2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
4	2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0

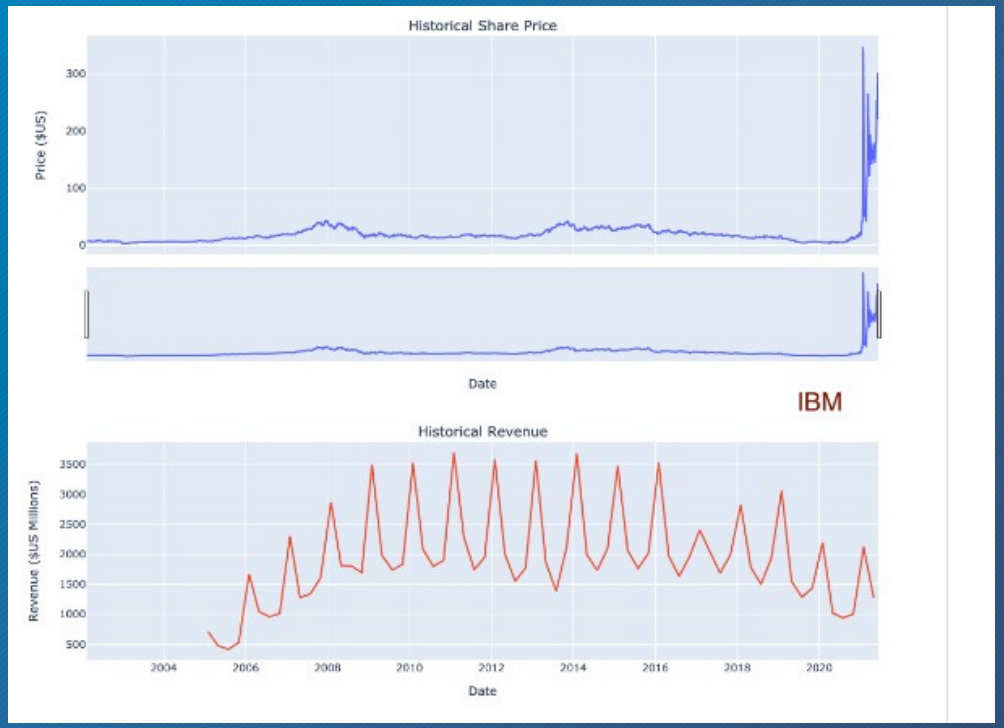
Top 5 revenue from upward and downward

	Date	Revenue
59	2006-01-31	1667
60	2005-10-31	534
61	2005-07-31	416
62	2005-04-30	475
63	2005-01-31	709

	Date	Revenue
41	2010-09-30	31
42	2010-06-30	28
43	2010-03-31	21
45	2009-09-30	46
46	2009-06-30	27

GRAPH

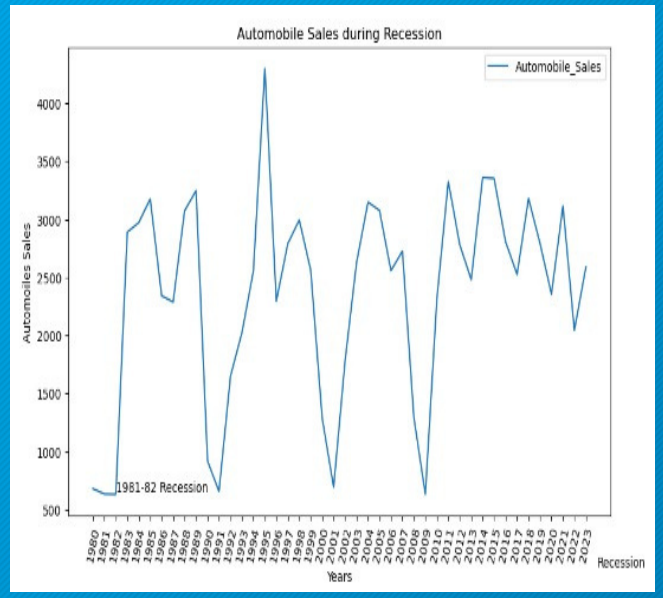

Historical Share Price
Price ($US)
800
600
400
200
0
Date
IBM
Historical Revenue
Revenue ($US Millions)
10k
8k
6k
4k
2k
0
2010
2012
2014
2016
2018
2020
Date


Historical Share Price
Price ($US)
300
200
100
0
Date
IBM
Historical Revenue
Revenue ($US Millions)
3500
3000
2500
2000
1500
1000
500
2004
2006
2008
2010
2012
2014
2016
2018
2020
Date

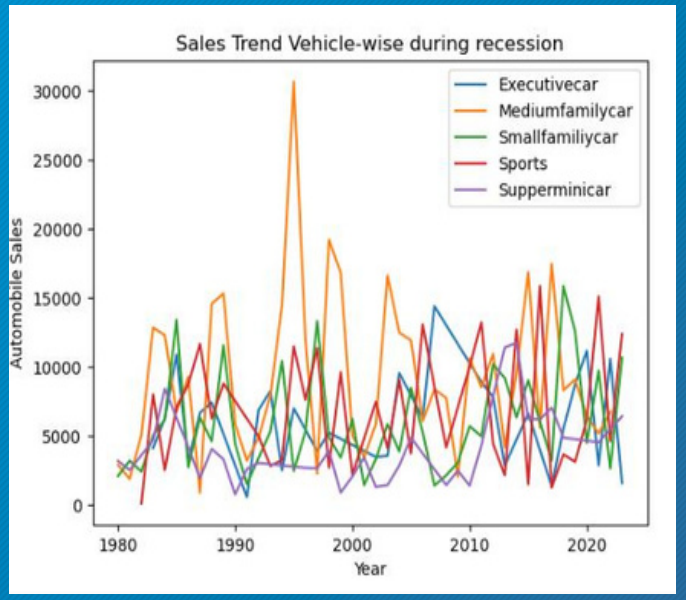
- TASK 1.1: Develop a *Line chart* using the functionality of pandas to show how automobile sales fluctuate from year to year. **(1 point)**
- TASK 1.2: Plot different lines for categories of vehicle type and analysethe trend to answer the question "Is there a noticeable difference in sales trends between different vehicle types during recession periods?" **(1 point)**
- TASK 1.3: Use the functionality of SeabornLibrary to create a visualization to compare the sales trend per vehicle type for a recession period with a non-recession period. **(1 point)**
- TASK 1.4: Use sub plotting to compare the variations in GDP during recession and non-recession period by developing line plots for each period. **(2 points)**
- TASK 1.5: Develop a Bubble plot for displaying the impact of seasonality on Automobile Sales. **(1 point)**
- TASK 1.6: Use the functionality of Matplotlibto develop a scatter plot to identify the correlation between average vehicle price relate to the sales volume during recessions. **(1 point)**
- TASK 1.7: Create a pie chart to display the portion of advertising expenditure of XYZAutomotivesduring recession and non-recession periods. **(1 point)**
- TASK 1.8: Develop a pie chart to display the total Advertisement expenditure for each vehicle type during recession period. **(1 point)**
- TASK 1.9: Develop a line plot to analysethe effect of the unemployment rate on vehicle type and sales during the Recession Period.

AUTOMOBILE DATASET FINDINGS

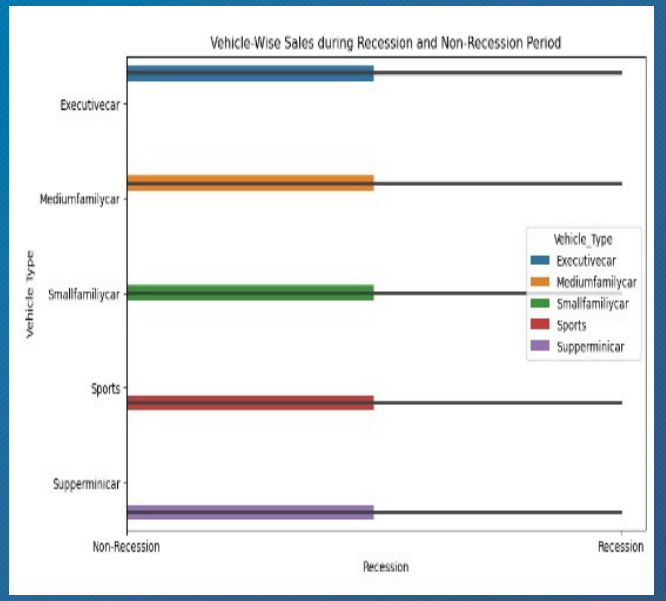
1



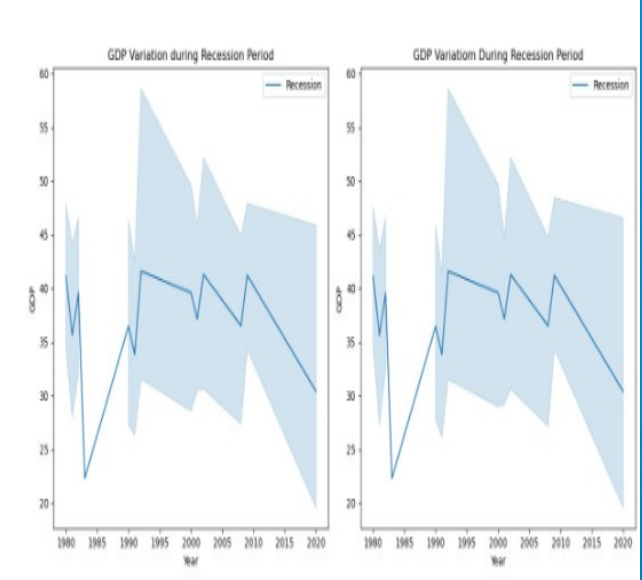
2



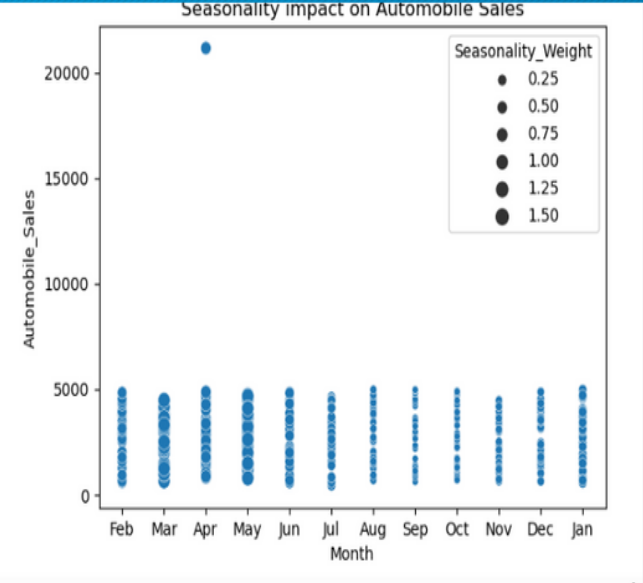
3



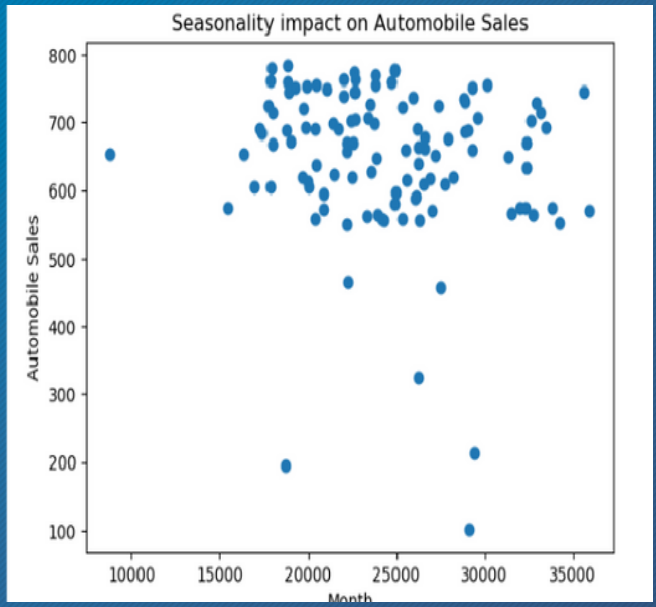
4


GDP Variation during Recession Period
Recession
GDP Variation During Recession Period
Recession
Year
Year

5


Seasonality impact on Automobile Sales
Seasonality_Weight
0.25
0.50
0.75
1.00
1.25
1.50
Automobile_Sales
Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan
Month

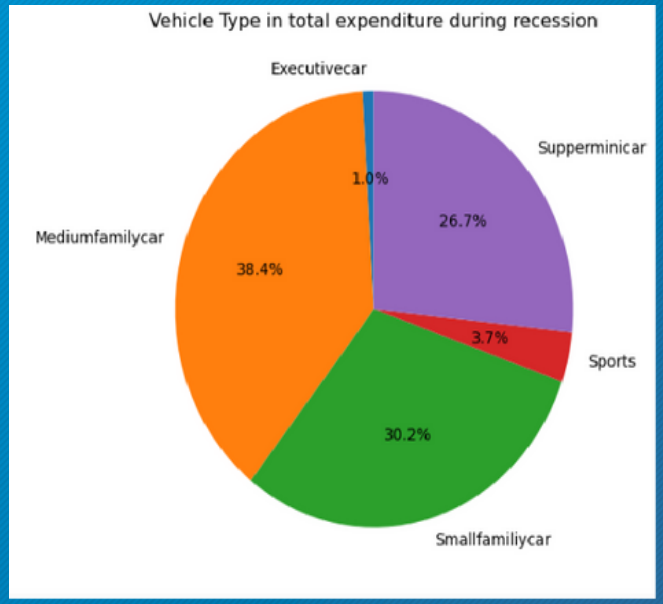
6


Seasonality impact on Automobile Sales
Automobile Sales
Month

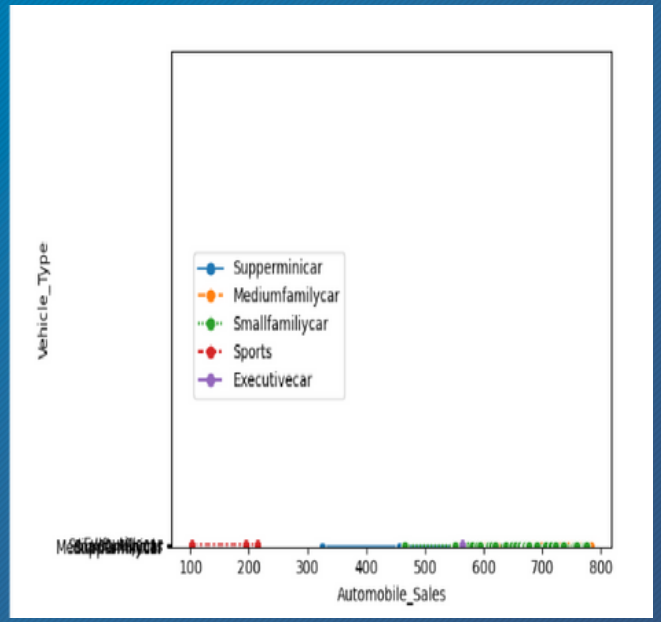
7


Advertising expenditure of XYZAautomotive
Non-Recession
0.0%
100.0%
Recession

8


Vehicle Type in total expenditure during recession
Executivecar
Supperminicar
1.0%
26.7%
Mediumfamilycar
38.4%
3.7%
Sports
30.2%
Smallfamiliycar

9


Vehicle_Type
Supperminicar
Mediumfamilycar
Smallfamiliycar
Sports
Executivecar
100
200
300
400
500
600
700
800
Automobile_Sales

DASHBOARD

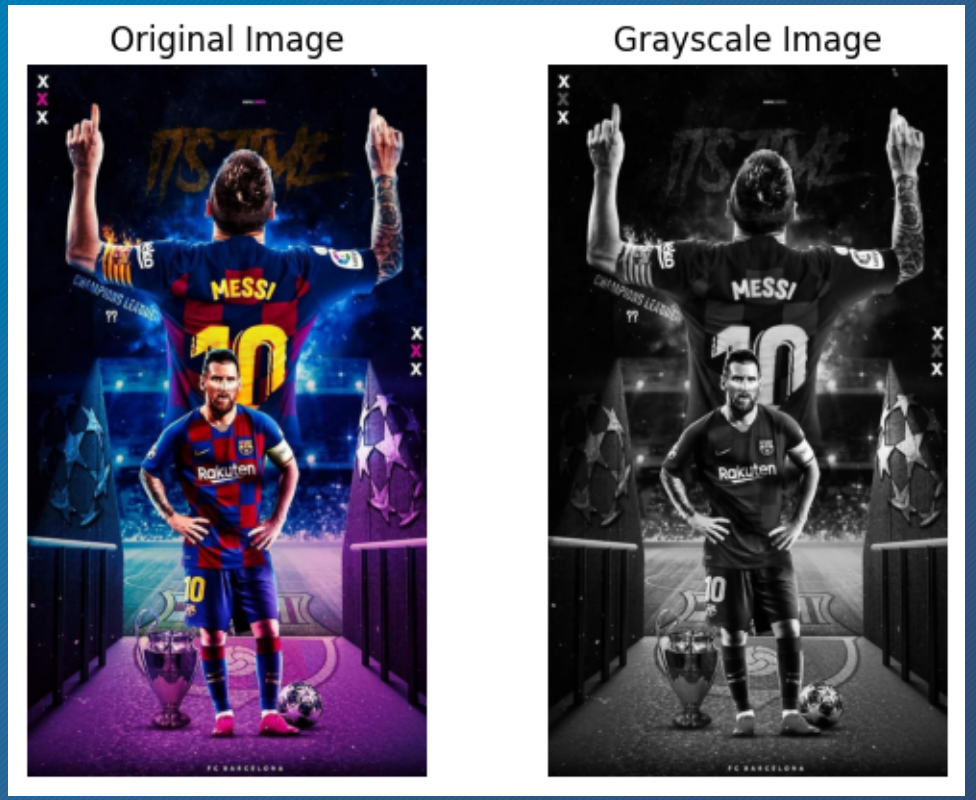

Average Automobile Sales fluctuation over Recession Period
Automobile_Sales
Year
Total Ad Expenditure Share by Vehicle Type during Recessions
Mediumfamilycar
Smallfamilycar
Supermnicar
Sports
Executivecar
38.4%
0.854%
Average Sales of Vehicle Types during Recessions
Automobile_Sales
Executivecar
Mediumfamilycar
Smallfamilycar
Sports
Supermnicar
Vehicle_Type
Vehicle-wise Sales by Unemployment Rate during Recessions
Automobile_Sales
Vehicle_Type
Mediumfamilycar
Smallfamilycar
Supermnicar
Executivecar
Sports

Examples

```
[5]: from google.colab.patches import cv2_imshow

[7]: import numpy as np
import cv2 as cv
img = cv.imread('/content/b0663c934150e3a0ce1ae37f3d2ce3db.jpg')
assert img is not None, "file could not be read, check with os.path.exists()"
res = cv.resize(img,None,fx=0.5, fy=0.5, interpolation = cv.INTER_CUBIC)
#OR
height, width = img.shape[:2]
                res = cv.resize(img,dsize=(int(0.5*width), int(0.5*height)), interpolation = cv.
↪INTER_CUBIC)cv2_imshow(res)
```

Examples

```
[14]:  import cv2
       from matplotlib import      pyplot  as  plt
      #Color conversions
#Result1


       image = cv2.imread('/content/b0663c934150e3a0ce1ae37f3d2ce3db.jpg')


       gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    plt.subplot(1, 2, 1)
    plt.imshow(cv2.cvtColor(image,
    cv2.COLOR_BGR2RGB)) plt.title('Original Image')
    plt.axis('off')

    plt.subplot(1, 2, 2)
    plt.imshow(gray_image, cmap='gray')
    plt.title('Grayscale Image')
    plt.axis('off')

    plt.show()
```



Examples

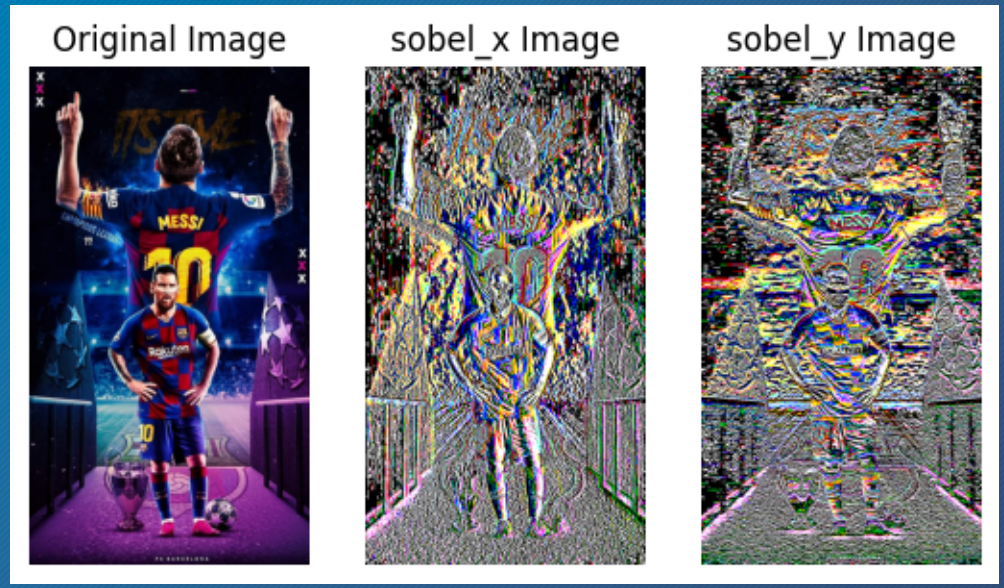
```
[23]: #Color conversions
#Result2

 image = cv2.imread('/content/b0663c934150e3a0ce1ae37f3d2ce3db.jpg')
yuv_image = cv2.cvtColor(image, cv2.COLOR_BGR2YUV)

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2YUV))
plt.title('YUV Image')
plt.axis('off')

plt.show()
```

Examples

```
[23]: #Color conversions
#Result2

 image = cv2.imread('/content/b0663c934150e3a0ce1ae37f3d2ce3db.jpg')
yuv_image = cv2.cvtColor(image, cv2.COLOR_BGR2YUV)

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2YUV))
plt.title('YUV Image')
plt.axis('off')

plt.show()
```

Examples

```
[42]: #Linear and non-linear filters
#result2

  image = cv2.imread('/content/b0663c934150e3a0ce1ae37f3d2ce3db.jpg')

sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)
sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)

plt.subplot(1, 3, 1)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3))
plt.title('sobel_x Image')
plt.axis('off')
plt.subplot(1, 3, 3)
plt.imshow(cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3))
plt.title('sobel_y Image')
plt.axis('off')

  WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
  WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
[42]: (-0.5, 719.5, 1279.5, -0.5)
```

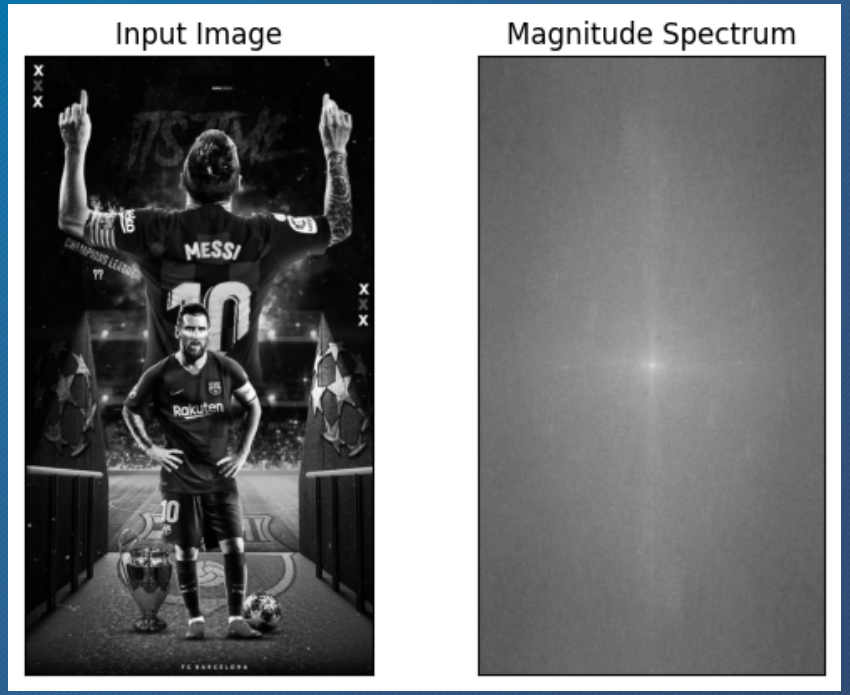


Examples

[46]: *#Fourier transforms*
#result1

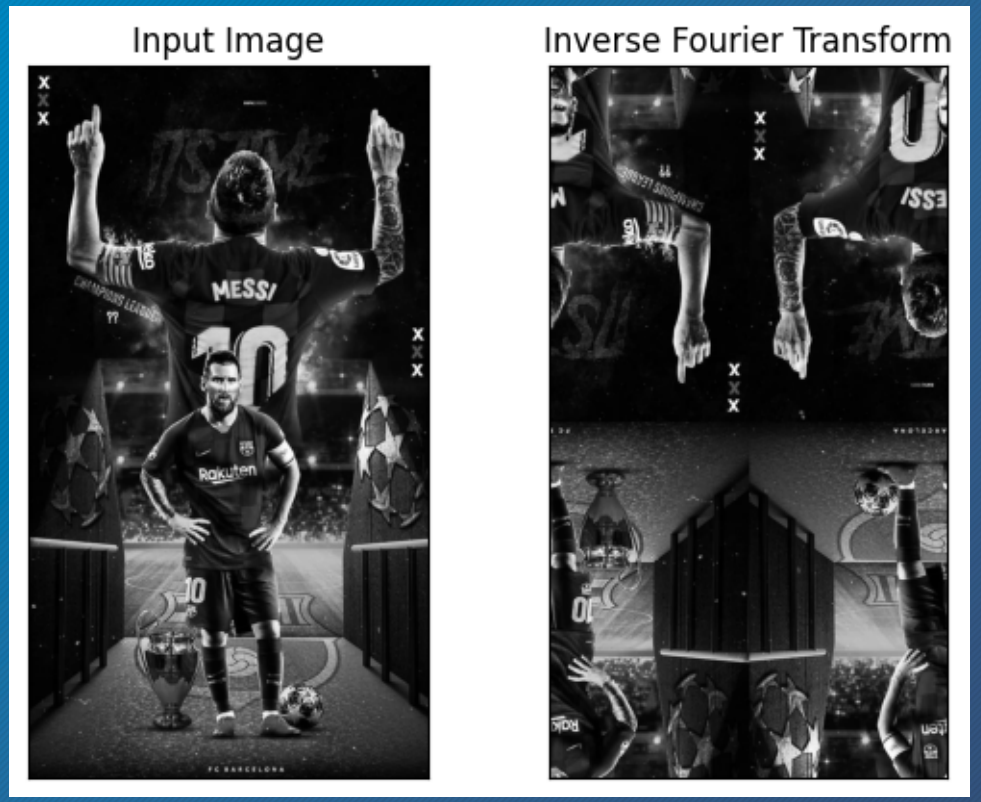
```
import cv2
import numpy as np
import matplotlib.pyplot as plt


image = cv2.imread('/content/b0663c934150e3a0ce1ae37f3d2ce3db.jpg', cv2.
↪IMREAD_GRAYSCALE)


if image is None:
    raise ValueError ("Image not found")


dft = cv2.dft(np.float32(image), flags=cv2.DFT_COMPLEX_OUTPUT)


dft_shift = np.fft.fftshift(dft)


magnitude_spectrum = 20 * np.log(cv2.magnitude(dft_shift[:, :, 0], dft_shift[:,␣
↪:, 1]))


plt.subplot(121), plt.imshow(image, cmap='gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(magnitude_spectrum, cmap='gray') plt.title('Magnitude
Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-
packages (4.8.0.76)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-
packages (from opencv-python) (1.23.5)
```



Examples

```
dft_shift = np.fft.fftshift(dft)


idft = cv2.idft(dft_shift)
idft_shift = np.fft.fftshift(idft)
magnitude_idft = cv2.magnitude(idft_shift[:, :, 0], idft_shift[:, :, 1])


plt.subplot(121), plt.imshow(image, cmap='gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(magnitude_idft, cmap='gray')
plt.title('Inverse Fourier Transform'), plt.xticks([]), plt.yticks([])
plt.show()
```



- "In closing, it's crucial to recognize that every data analysis comes with its own set of limitations and potential biases."
- "While our findings provide valuable insights, it's essential to remain mindful of the inherent flaws and uncertainties in the data."
- "As we conclude, let's not overlook the importance of critically evaluating our results and acknowledging any potential limitations or flaws in our analysis."
- "In the journey of data exploration, it's imperative to embrace the imperfections and uncertainties that come with the territory."
- "As we draw the curtains on our presentation, let's remember that transparency about the limitations and flaws in our analysis is key to maintaining integrity and trust in our findings."
- "As we wrap up, I encourage everyone to approach data analysis with a healthy dose of skepticism, recognizing that even the mostrigorous methodologies may have their flaws."
- "In the grand scheme of data analysis, understanding and addressing the flaws in our approach is just as important as uncoveringinsights."
- "In the final moments of our presentation, let's not forget to acknowledge the limitations inherent in our data and analysis,asthey serve as valuable lessons for future research."
- "As we conclude, let's remember that the pursuit of knowledge through data analysis is an ongoing journey marked by continuous learning and improvement."
- "In wrapping up our discussion, let's remain humble in the face of uncertainty and committed to refining our methods to mitigateflaws and strengthen the validity of our findings."

THANK YOU