**A MINOR PROJECT**

**On**

**SMARTWATCH PRICE PREDICTION USING MACHINE LEARNING AND DATA ANALYSIS**

Dissertation submitted in the partial fulfillment of the requirements

for the award of the degree of

# BACHELOR OF TECHNOLOGY

***By***

**DEPARTMENT OF INTERNSHIPS**

# KAJJAM VENKATA MANIKANTA CSWDA263

# PAVAN KUMAR

# GADDAM RAGHU VARMA CSWDA213

*Under the esteemed Guidance of*

# Er. Y V D CHANDRA SEKHAR

*Founder & Chief Executive Officer*

## *CS CODENZ*



**LEARN HERE, LEAD ANYWHERE**

**CS CODENZ**

**GUDIVADA – 521 323, ANDHRA PRADESH., INDIA**

**2022-2023**

**A MINOR PROJECT**

**On**

**SMARTWATCH PRICE PREDICTION USING MACHINE LEARNING AND DATA ANALYSIS**

Dissertation submitted in the partial fulfillment of the requirements

for the award of the degree of

**BACHELOR OF TECHNOLOGY**

***By***

**DEPARTMENT OF INTERNSHIPS**

## KAJJAM VENKATA MANIKANTA PAVAN ( CSWDA263 )

## KUMAR

## GADDAM RAGHU VARMA ( CSWDA213 )

*Under the esteemed Guidance of*

## Er. Y V D CHANDRA SEKHAR

*Founder & Chief Executive Officer*

### *CS CODENZ*



**LEARN HERE, LEAD ANYWHERE**

**CS CODENZ**

**GUDIVADA – 521 323, ANDHRA PRADESH., INDIA**

**2023-2024**

### CS CODENZ



#### CERTIFICATE

This is to certify that dissertation entitled **“SmartWatch Price Prediction using Machine Learning and Data Analysis”** submitted by **GADDAM RAGHU VARMA ( CSWDA213 )** in the partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** from **CS CODENZ** is a record of Bonafede work carried out by them under my guidance and supervision during the year 2022-2023. The result embodied in this dissertation have not been submitted by any other university or Institution for the award of any degree.

**Signature of the Supervisor**

**Er. Y V D CHANDRA SEKHAR**

Founder & CEO, CS CODENZ

#### DECLARATION

I **GADDAM RAGHU VARMA (CSWDA213)** declared that the dissertation report entitled **“SmartWatch Price Prediction using Machine Learning and Data Analysis”** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

##### Roll No Name Signature

CSWDA213 GADDAM RAGHU VARMA

Date:

Place:

**COs, POs and PSOs Mapping**

Subject Name : Major Project

Subject Code : PY42223

AcademicYear : 2023 - 2024

|  |  |  |
| --- | --- | --- |
| Subject Code |  | Course Outcomes |
| PR4204 | CO1 | Formulate solutions to computing problems using latest technologies and tools |
| CO2 | Work effectively in teams to design and implement solutions to computational problems and socially relevant issues |
| CO3 | Recognize the social and ethical responsibilities of a professional working in the discipline |
| CO4 | Apply advanced algorithmic and mathematical concepts to the design and analysis of software |
| CO5 | Devise a communication strategy (language, content and medium) to deliver messages according to the situation and need of the audience. |
| CO6 | Deliver effective presentations, extemporaneous or impromptu oral presentations. Setting up technical reports using technical tools. |

CO-PO-PSOs Mapping

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO 1 | 3 | 2 | - | 2 | 2 | - | - | - | - | - | - | - | 3 | - | - |
| CO 2 | 2 | 3 | - | 2 | 2 | - | - | - | - | - | - | - | 3 | - | - |
| CO 3 | 3 | 3 | - | 2 | 2 | - | - | - | - | - | - | - | 3 | - | - |
| CO 4 | 3 | 3 | - | 2 | 2 | - | - | - | - | - | - | - | 3 | - | - |
| CO 5 | 2 | 3 | - | 2 | 2 | - | - | - | - | - | - | 1 | 3 | - | - |
| CO 6 | 2 | 3 | 2 | 2 | 3 | - | - | - | 2 | 2 | 2 | 2 | 3 | - | - |
| **Avg** | **2.50** | **2.83** | **2.00** | **2.00** | **2.17** | **-** | **-** | **-** | **2.00** | **2.00** | **2.00** | **1.50** | **3.00** | **-** | **-** |

*Note: 1 – Good , 2 – Average, 3 - Excellent*

Signature of Student with Date Signature of Guide with Date

#### ACKNOWLEDGEMENT

This report dissertation could not have been written without the support of our guide **Er. Y V D Chandra Sekhar, Founder & CEO, CS CODENZ** who not only served as our superior but also encouraged and challenged us throughout our academic program our foremost thanks goes to his. Without his this dissertation would not have been possible. We appreciate him vast knowledge in many areas, and his insights, suggestions and guidance that helped to shape our research skills

It is needed with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We owe many thanks to many people who helped and supported us during the writing of this report

We are thankful to our project coordinator **Er. Y V D Chandra Sekhar,** Founder & CEO, CS CODENZ, for his continuous support

We express our sincere thanks to our respected for bet valuable suggestion and constant motivation that greatly helped us in successful completion of project We also take the privilege to express our heartfelt gratitude to **Er. Y V D Chandra Sekhar, Founder & CEO, CS CODENZ**

We are thankful to all faculty members for extending their kind cooperation and assistance Finally, we are extremely thankful to our parents and friends for their constant helped moral support

### Table Of Contents

Abstract…………………………………………………………………(i)

ProblemStatement………………………………………………………….1

ER Diagram....…………………………………………… . ………………2

Requirements……………………………………………………………….3

Description………………………………………………………………….4

Coding………………………………………………………………………5

Result …………………………………………………………………… . 46

Output…………………………………………………………………… . 48 Summary…………………………………………………………………. .51

### ABSTRACT

The Primary objective of this “**SMARTWATCH PRICE PREDICTION USING MACHINE LEARNING AND DATA ANALYSIS**” is to fulfill customer satisfaction and good trust along with challenges of the businesses. It includes data of smartwatch prices and real-time monitoring capabilities.

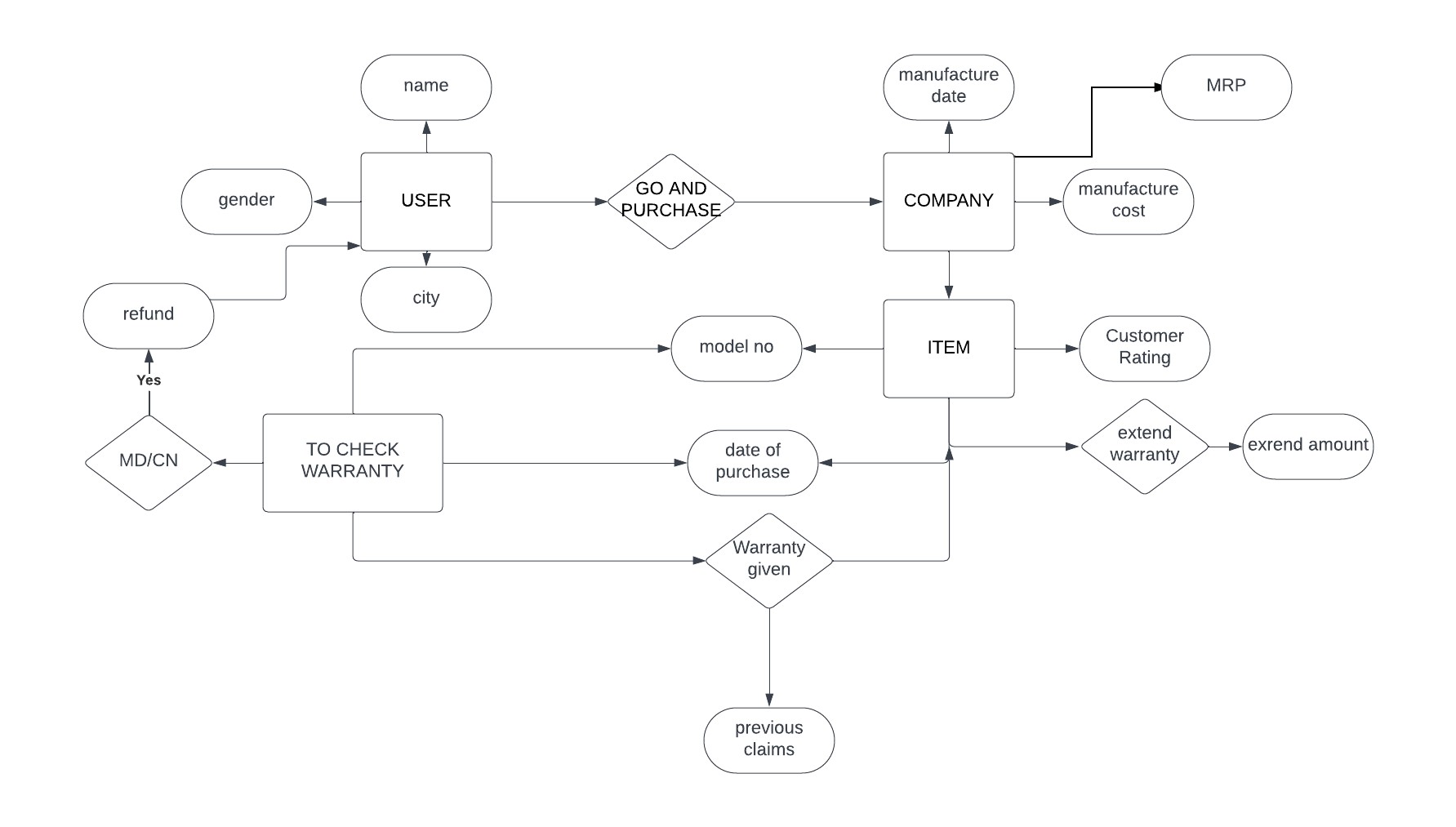
In today's dynamic consumer electronics market, smartwatches have emerged as a popular and rapidly evolving category of wearable technology. Predicting the prices of smartwatches is crucial for both consumers seeking the best deals and retailers optimizing their pricing strategies. This project presents a comprehensive framework for smartwatch price prediction, leveraging advanced machine learning techniques and data-driven insights.

**PROBLEM STATEMENT**

Smartwatch price prediction is a challenging task due to the rapid evolution of smartwatch technology and the diverse range of models available in the market. However, by using machine learning techniques, it is possible to develop models that can predict smartwatch prices with reasonable accuracy.

In the ever-evolving market of consumer electronics, specifically within the realm of smartwatches, consumers and retailers face a critical challenge: the lack of an efficient and accurate means to predict the prices of smartwatches.

**E-R DIAGRAM**



### REQUIREMENTS

#### HARDWARE REQUIREMENTS

* Personal Computer / Laptop with minimum RAM (4 GB), ROM (128 GB) and

Processor(i3)

* Good latency internet access

#### SOFTWARE REQUIREMENTS

* Basic Search Engine (Google)
* Google Colaboratory
* MICROSOFT WORD

#### FUNCTIONAL REQUIREMENTS

* Calculation
* Help in manipulating data and easy process.
* Graphical representation of Datasets

### DESCRIPTION

Data Analytics is a process of scrutinizing the data to obtain accurate results. In data analytics the main purpose is extracting the original data from data. In this data analytics we need to perform the major that is data munging.

#### DATA MUNGING

It is a process of transferring unstructured data into structured format. The goal is to make the data more usable and valuable for analytics or other purposes.

#### STEPS OF DATA ANALYSIS

1. Defining the Question
2. Collecting the data
3. Cleaning the data
4. Analyzing the data
5. Sharing your results
6. Embracing your failures
7. Summary

### CODE

#### 1) DEFINING THE QUESTION

The first step in any data analysis process is to define your objective. In data analytics **jargon**, this is sometimes called the ‘problem statement’. The problem at hand is to develop an efficient and accurate prediction of SmartWatches.

#### 2) COLLECTING THE DATA

**Pandas -** Helps to create a dataset and it is also a library in python.

**Pandas Package –** It’s a group of Panal Data’s which are used to analyze the labelled data and relational data.

**Series –** A series is a method of pandas and labelled data. Series are nothing but columns in Excel sheet.

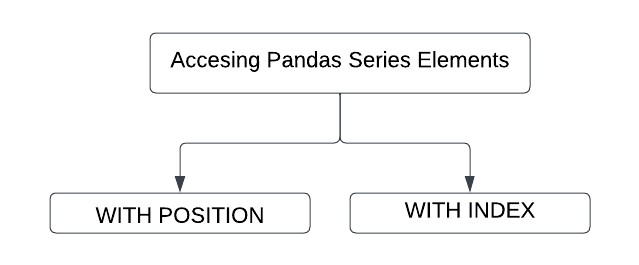
##### CREATING SERIES-

|  |
| --- |
| import pandas as pd a=[10,20,30,'a',50] b=pd.Series(a) print (b) |

1. 10
2. 20
3. 30
4. a 4 50 dtype: object

|  |
| --- |
| import pandas as pd  a=[10,20,30,40,50]  b=pd.Series(a)  print(b) |

1. 10
2. 20
3. 30
4. 40 4 50 dtype: int64 import pandas as pd a=[10,20,30,40,50] b=pd.Series(a) print(b)



##### ACCESSING WITH POSITION-

|  |
| --- |
| #SLICING import pandas as pd a=[10,20,30,40,50] b=pd.Series(a) print(b[-2:]) |

3 40 4 50 dtype: int64

|  |
| --- |
| #SLICING import pandas as pd a=[10,20,30,40,50] b=pd.Series(a) print(b[-4:-2]) |

1. 20 2 30 dtype: int64

|  |
| --- |
| import pandas as pd a=[10,20,30,40,50] b=pd.Series(a) print(b[2:]) |

1. 30
2. 40
3. 50

dtype: int64

##### ACCESSING WITH INDEX-

|  |
| --- |
| import pandas as pd  a=[10,20,'CS',40,50]  b=pd.Series(a,index=['!','@','#','$','%'])  print(b) print("-------------------")  print(b['@']) |

! 10

@ 20

# CS

$ 40 % 50 dtype: object -------------------

20

##### DATAFRAME

A data frame is a 2D data structure in which we store data in the form of tables. [rows x columns] We can create a table via Data Frame i.e., known as DATASET.

##### CREATING A DATASET-

|  |
| --- |
| #creating empty data set import pandas as pd a=pd.DataFrame() print(a) |

Empty DataFrame

Columns: []

Index: []

###### Creating data set using list-

|  |
| --- |
| #creating dataframe by using list import pandas as pd a=[10,20,30,40,50]    b=pd.DataFrame(a) print(b) |

0

1. 10
2. 20
3. 30
4. 40
5. 50

###### Creating data set using Dict-

|  |
| --- |
| #creating using DICT import pandas as pd x=[{'a':10,'b':20,'c':30}] y=pd.DataFrame(x) print(y) |

a b c

0 10 20 30

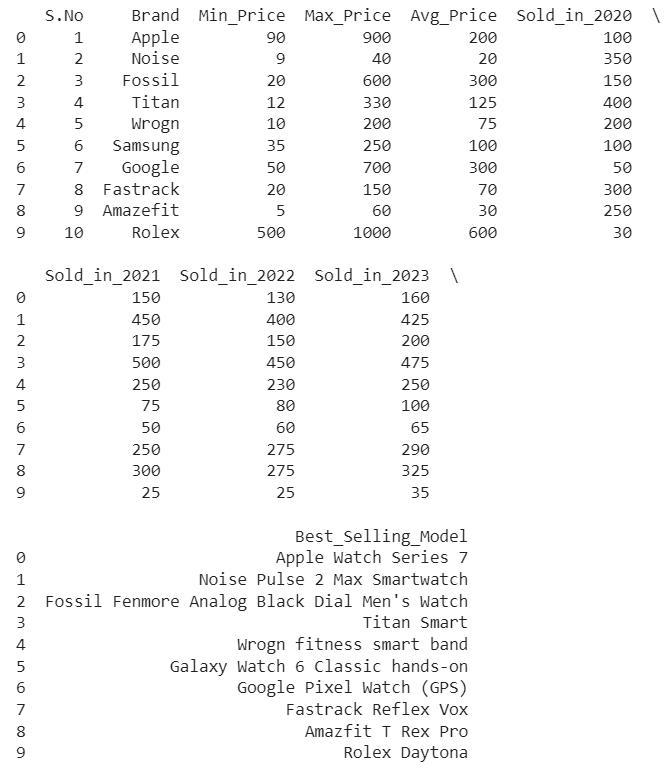
|  |
| --- |
| #creating using DICT  import pandas as pd x=[{'a':10,'b':20,'c':30}] y=pd.DataFrame(x) print(type(y)) |

<class 'pandas.core.frame.DataFrame'>

Creating dataset using Series

|  |
| --- |
| import pandas as pd a=[10,20,30,40] b=pd.Series(a) c=pd.DataFrame(b) print(c) |

|  |
| --- |
| #Import the Required libraries  import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  #Generating the Dataset using Pandas  data = { 'S.No': pd.Series([1,2,3,4,5,6,7,8,9,10]),  'Brand': pd.Series(['Apple','Noise','Fossil','Titan','Wrogn','Samsung','Google','Fastrack','Amazefit',  'Rolex']),  'Min\_Price': pd.Series([90,9,20,12,10,35,50,20,5,500]),  'Max\_Price': pd.Series([900,40,600,330,200,250,700,150,60,1000]),  'Avg\_Price': pd.Series([200,20,300,125,75,100,300,70,30,600]),  'Sold\_in\_2020': pd.Series([100,350,150,400,200,100,50,300,250,30]),  'Sold\_in\_2021': pd.Series([150,450,175,500,250,75,50,250,300,25]),  'Sold\_in\_2022': pd.Series([130,400,150,450,230,80,60,275,275,25]),  'Sold\_in\_2023': pd.Series([160,425,200,475,250,100,65,290,325,35]),  'Best\_Selling\_Model':pd.Series(['Apple Watch Series 7','Noise Pulse 2 Max Smartwatch',  "Fossil Fenmore Analog Black Dial Men's Watch",'Titan Smart','Wrogn fitness smart band',  'Galaxy Watch 6 Classic hands-on','Google Pixel Watch (GPS)','Fastrack Reflex Vox','Amazfit T Rex Pro',  'Rolex Daytona'])}  Data = pd.DataFrame(data)  print(Data) |



#### OPERATIONS ON DATASET

There are three operations we can perform on a dataset.

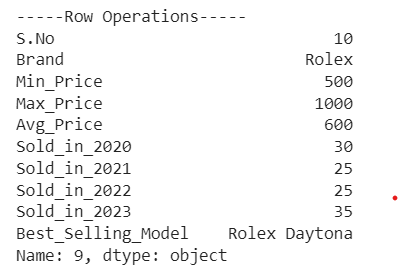
1. Row Operation
2. Column Operation
3. Selection Operation

ROW OPERATIONS-

* row selection
* row addition
* row deletion

##### ROW SELECTION-

|  |
| --- |
| #Performing the Row Operations  print('-----Row Operations-----')  #selecting a row and print the selected row  print(copy.loc[9]) |

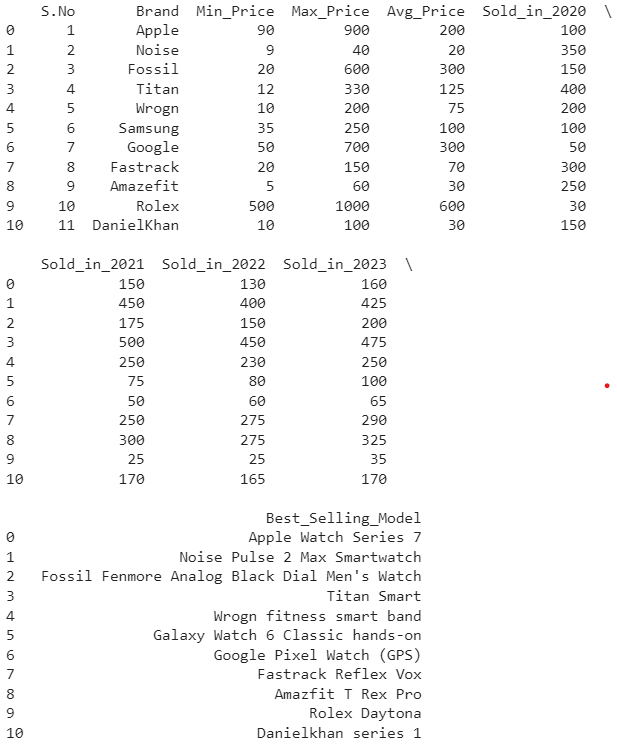


##### ROW ADDITION-

We can add the row for the dataset by using “.LOC()” method

But the row data must be the same comparing to the other rows.

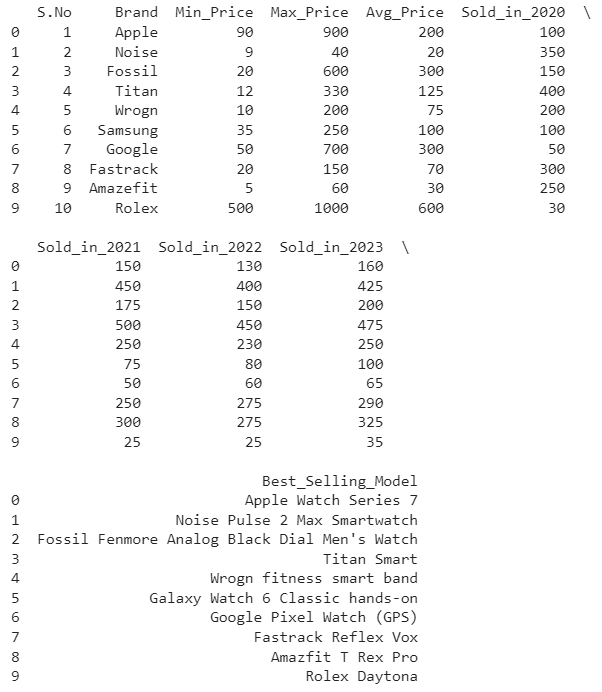
|  |
| --- |
| #Adding a Row and print the DataFrame with including added row  copy.loc[10] = [11,'DanielKhan',10,100,30,150,170,165,170,'Danielkhan series 1']  print(copy) |



##### ROW DELETION-

By using drop() method we can delete the row.

|  |
| --- |
| #Deleting a selected row and print the DataFrame  cop = copy.drop(10)  print(cop) |



###### COLUMN OPERATIONS

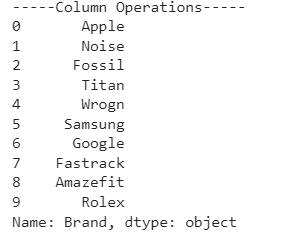
* Column Selection
* Column Addition
* Column deletion

###### COLUMN SELECTION-

We can select the column by using the column name and the data frame.

DataFrameObj.[‘column name’]

|  |  |
| --- | --- |
| |  | | --- | | #Performing Column Operations  print('-----Column Operations-----')  #Select a Column in a DataFrame and print the Selected Column  print(cop['Brand']) | |

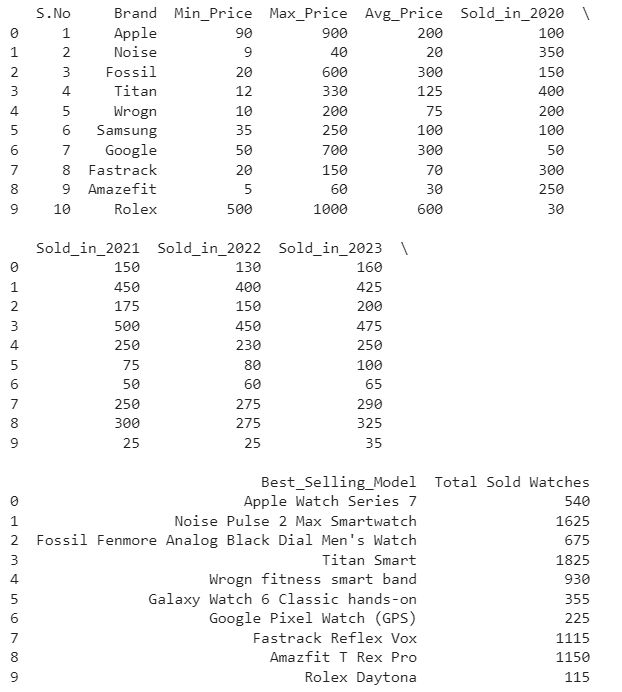


###### COLUMN ADDITION

We can add columns by dataframe[‘ ‘]

I want to add a column Total Sold Watches it can be done by adding 2 columns i.e., sold\_in\_2022 and sold\_in\_2023

|  |
| --- |
| #Adding a Column and print the DataFrame with newly added Column also  cop['Total Sold Watches'] = cop['Sold\_in\_2020']+cop['Sold\_in\_2021']+cop['Sold\_in\_2022']+cop['Sold\_in\_2023']  print(cop) |



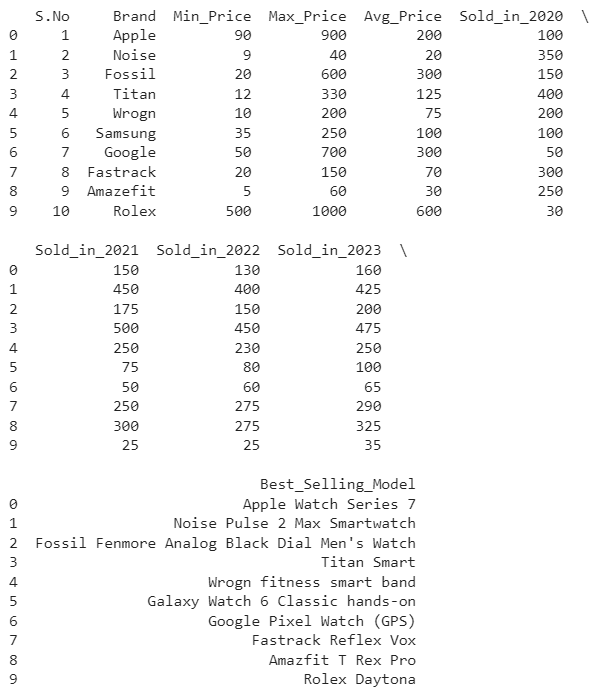
###### COLUMN DELETION –

To delete an entire column from a Pandas Series in Python, you can use the ’drop’ method or simply select the columns you want to keep.

|  |
| --- |
| #Deleting a column and print the DataFrame  del cop['Total Sold Watches']  print(cop) |

The taken dataset ‘Total Sold Watches’ column is removed .

The output of the code is given below.

****

**Reshaping the Data** : In Reshaping of data in the dataset there is no possibility to reshape based on our requirement.

There is only scope we can reshaping the data with the help of stack.

#Using stack()

copy = cop.stack()

print(copy)

0 S.No 1

Brand Apple

Min\_Price 90

Max\_Price 900

Avg\_Price 200

...

9 Sold\_in\_2020 30

Sold\_in\_2021 25

Sold\_in\_2022 25

Sold\_in\_2023 35

Best\_Selling\_Model Rolex Daytona

Length: 100, dtype: object

#Using Unstack()

cop = copy.unstack()

print(cop)

###### S.No Brand Min\_Price Max\_Price Avg\_Price Sold\_in\_2020 Sold\_in\_2021 \

###### 0 1 Apple 90 900 200 100 150

###### 1 2 Noise 9 40 20 350 450

###### 2 3 Fossil 20 600 300 150 175

###### 3 4 Titan 12 330 125 400 500

###### 4 5 Wrogn 10 200 75 200 250

###### 5 6 Samsung 35 250 100 100 75

###### 6 7 Google 50 700 300 50 50

###### 7 8 Fastrack 20 150 70 300 250

###### 8 9 Amazefit 5 60 30 250 300

###### 9 10 Rolex 500 1000 600 30 25

###### Sold\_in\_2022 Sold\_in\_2023 Best\_Selling\_Model

###### 0 130 160 Apple Watch Series 7

###### 1 400 425 Noise Pulse 2 Max Smartwatch

###### 2 150 200 Fossil Fenmore Analog Black Dial Men's Watch

###### 3 450 475 Titan Smart

###### 4 230 250 Wrogn fitness smart band

###### 5 80 100 Galaxy Watch 6 Classic hands-on

###### 6 60 65 Google Pixel Watch (GPS)

###### 7 275 290 Fastrack Reflex Vox

###### 8 275 325 Amazfit T Rex Pro

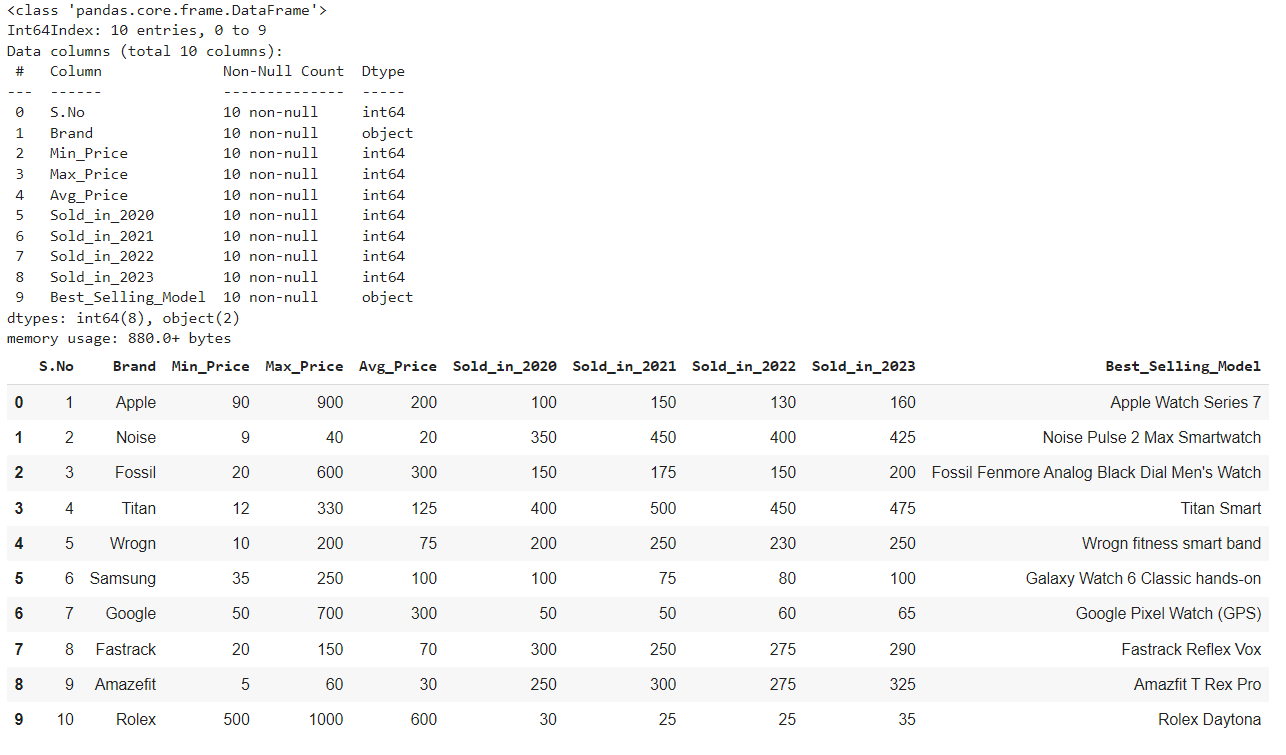
###### 9 25 35

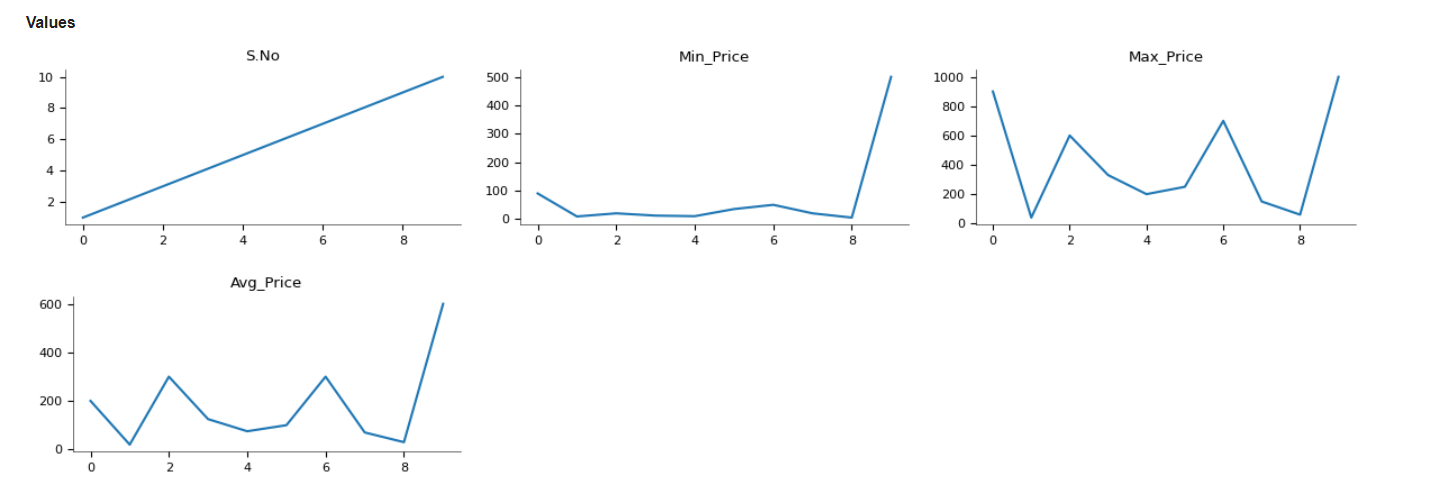
###### Rolex Daytona

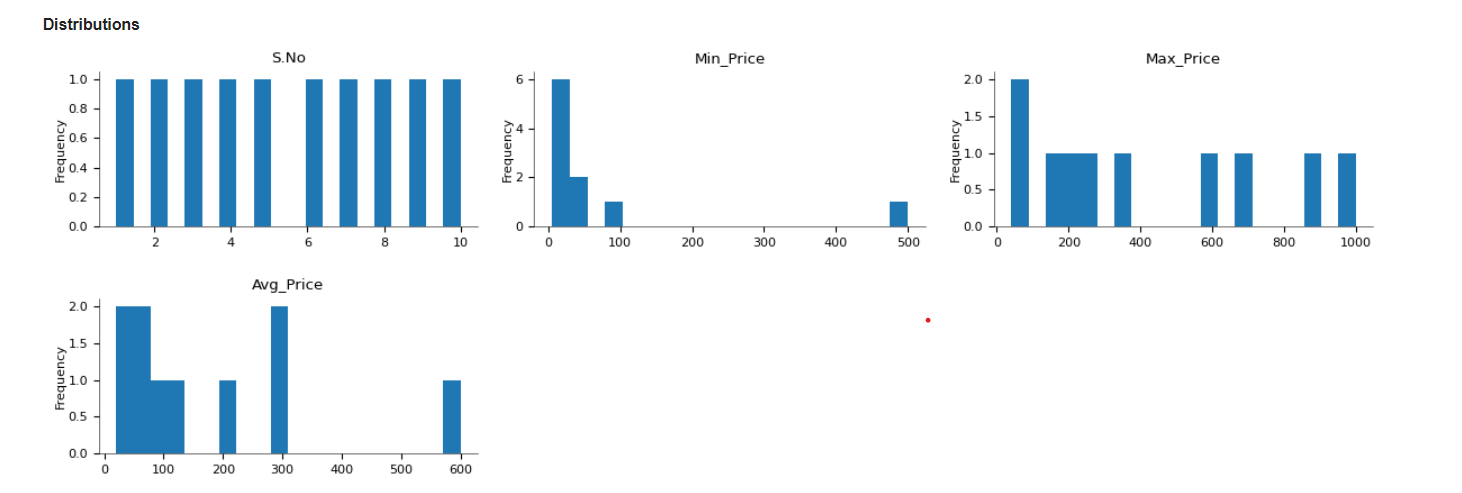
###### INFO( )

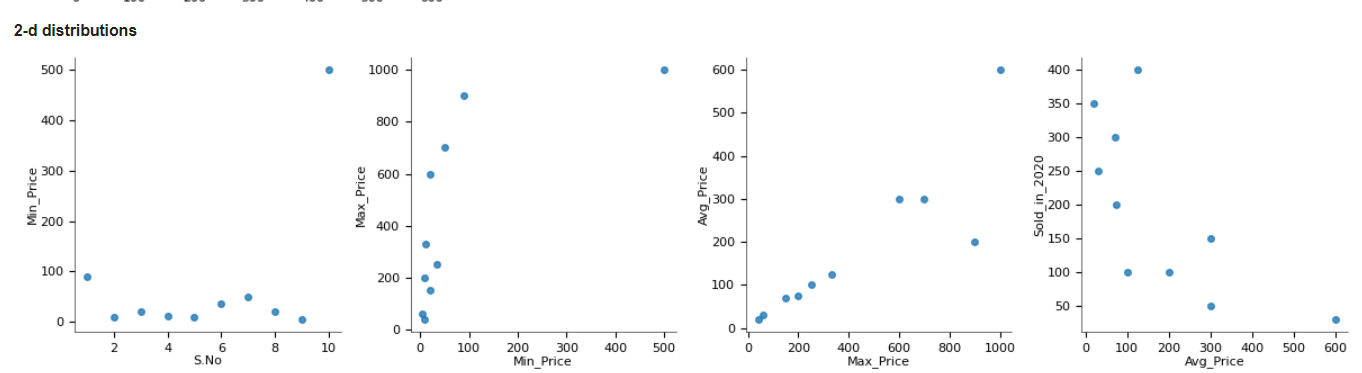
The info method provides a summary of the data including the data types of each column the number of non-null values.

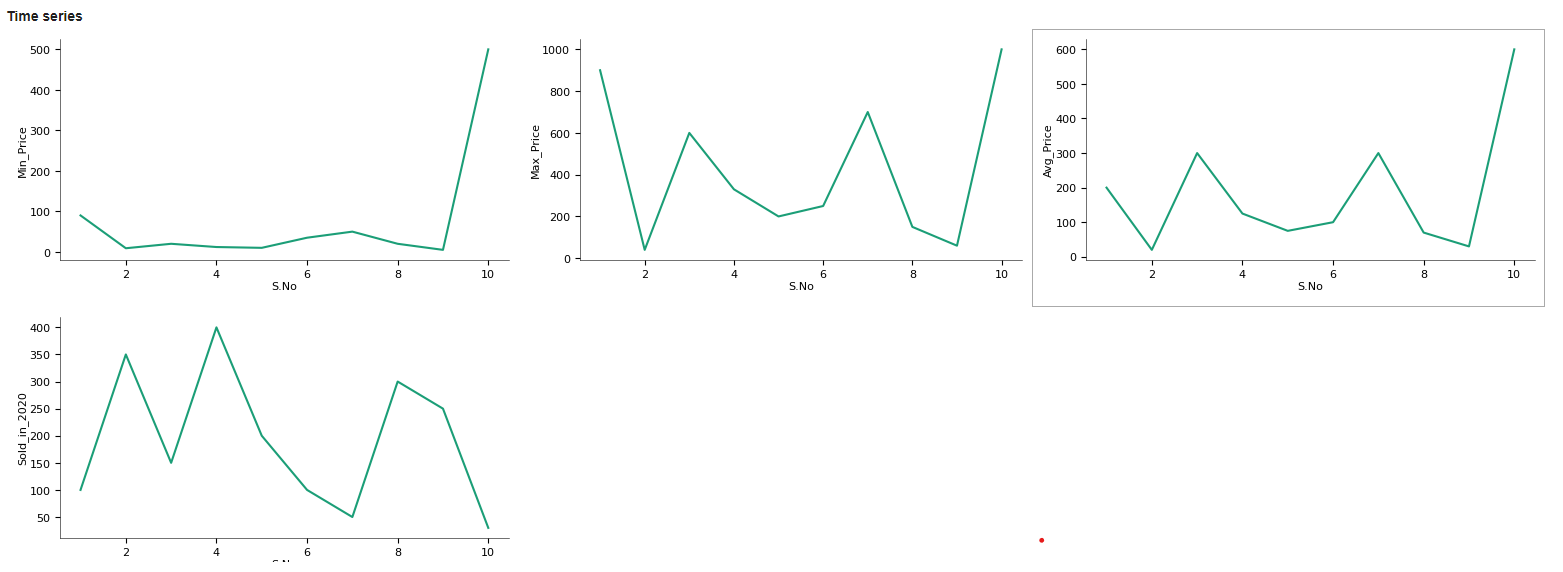
|  |  |
| --- | --- |
| |  | | --- | | #Info of the Data  cop.info()  cop | |

****









**Indexing the Data**

#Indexing the Data

print(cop.index)

Int64Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype='int64')

**SIZE OF DATA**

#Size of the Data

print(Data.size)

100

#Number of Dimensions in Data

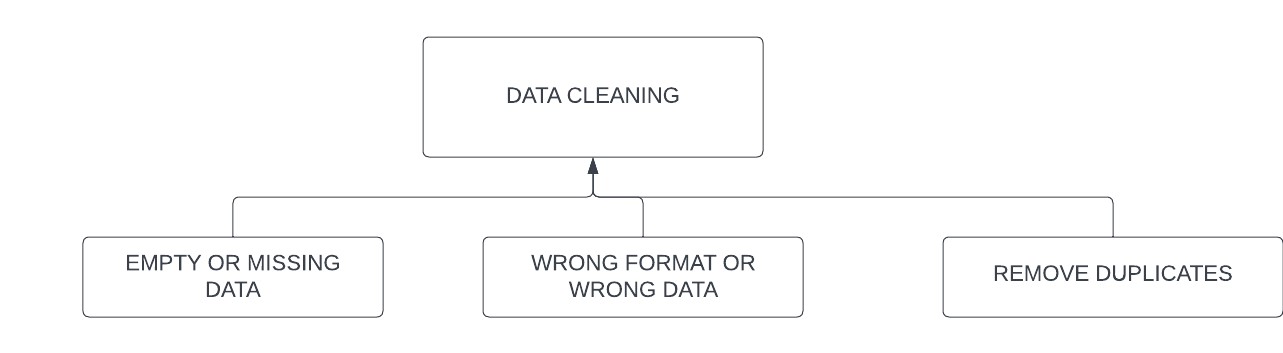
cop.ndim

2

**3) CLEANING THE DATA**

It is the process of remaining or replacing the NAN values.

**NAN** – Not A Null (or) Not A None



##### EMPTY OR MISSING DATA

Handling with Null Values :

Here we can perform operations on Dataset by loading numpy’s. In this process we can identify what are the missing values.

We can solve empty cells by using two methods.

* isnull( )
* notnull( )

#Import the Required libraries

import numpy as np

import pandas as pd

#Generating the Dataset using Pandas

data = { 'S.No': pd.Series([1,2,3,4,5,6,7,8,9,10]),

'Brand':pd.Series(['Apple','Noise','Fossil','Titan','Wrogn','Samsung','Google','Fastrack','Amazefit','Rolex']),

'Min\_Price': pd.Series([90,9,20,12,10,np.nan,50,20,5,500]),

'Max\_Price': pd.Series([900,40,600,330,200,250,700,150,60,1000]),

'Avg\_Price': pd.Series([200,20,300,125,75,100,300,70,30,600]),

'Sold\_in\_2020': pd.Series([100,350,150,np.nan,200,100,50,300,250,30]),

'Sold\_in\_2021': pd.Series([150,450,175,500,250,75,50,250,np.nan,25]),

'Sold\_in\_2022': pd.Series([130,400,150,450,230,80,60,275,275,25]),

'Sold\_in\_2023': pd.Series([160,425,np.nan,475,250,100,65,290,325,35]),

'Best\_Selling\_Model':pd.Series(['Apple Watch Series 7','Noise Pulse 2 Max

Smartwatch',"Fossil Fenmore Analog Black Dial Men's Watch",

'Titan Smart','Wrogn fitness smart band','Galaxy Watch 6 Classic

handson','Google Pixel Watch (GPS)',

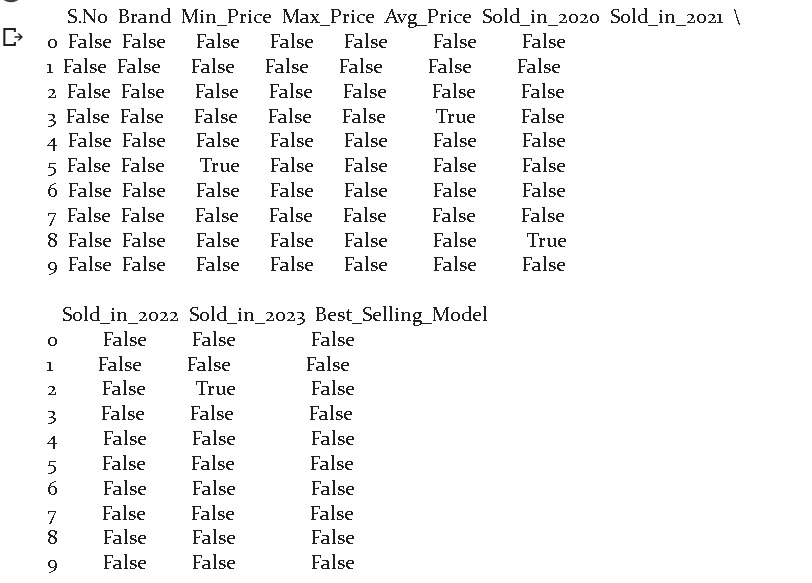
'Fastrack Reflex Vox','Amazfit T Rex Pro','Rolex Daytona'])}

Data = pd.DataFrame(data)

#Using isnull()

null = Data.isnull()

print(null)



Using Notnull():

#Using notnull()

notnull = Data.notnull()

print(notnull)

|  |
| --- |
| **Fillna():**    This method is used to fill the missing values with our required data.  We have 2 types in fillna( ) for the parameter method.  1.fillna(method=’pad’)  2.fillna(method=’bfill’)      #fillna()using pad parametre  fill = Data.fillna(method='pad')  print(fill)        #Checking Still any null values present in the Data  check = Data.isnull()  print(check)          **Analyzing the Data:** In the view of analyzing the data we perform the operations like    1.Statistical Operations  2.Computational Operations    **Statistical Operations** : In Statistical operations we have perform all operations which are related to mathematics. By performing these operations it gets proved on what dataset we generated.  The following are the Statistical Operations :  Mean( )  Mode( )  Median( )  Min( )  Max( )  Sum( )  Aggregate( )  Describe( )   * **Mean( ):** It performs the average operation for particular column in dataset we can perform the mean operation by using “ mean( ) ”.     # Mean of Data using mean()  mean = cop['Min\_Price'].mean()  print(mean)  75.1  # Mean of Data using mean()  mean = cop['Min\_Price'].mean()  print(mean)  423.0   * **Mode( ):** This method is used to return the output most repeated value   #Finding the Mode of the Data Using mode()  mode = cop['Avg\_Price'].mode()  mode  0 300  Name: Avg\_Price, dtype: object   * **Median( ) :** This method is used to return the mid value of the collection.   #Median of the Data using median()  median = cop['Max\_Price'].median()  median  290.0   * **Min( ) :** This method is used to return the minimum value in Selected column in the Dataset.     # Finding the Minimum value using min()  min = cop['Min\_Price'].min()  min  5   * **Max( ) :** This method is used to returns the maximum value in selected column of the Dataset.     # Finding the Maximum Value using max()  max = cop['Max\_Price'].max()  max      1000   * **Sum( ) :** This method is used to sum all the values in the collection.     # Sum all the prices in a column using sum()  sum = cop['Max\_Price'].sum()  sum  4230   * **Aggregate( ) :** This method is used perform 2 or more statistical operations at a time.   # Aggregating the Data using aggregate()  aggregate = cop.aggregate(['sum','min','max'])  aggregate |

##### COUNT( ) : This method is used to get the count of values in a column.

##### 

# Count the Number of Brand of watches using count()

count = cop['Brand'].count()

count

##### 

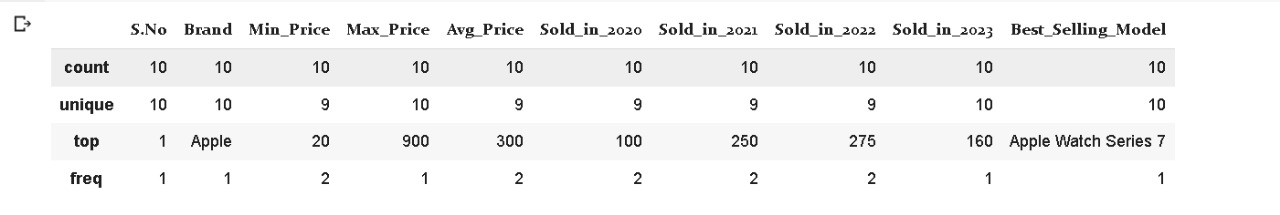
##### 10

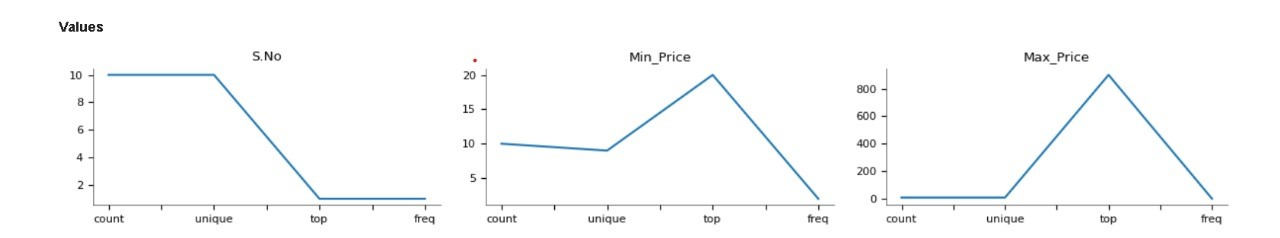
* **Describe( ) :** This method is used to get the details of dataset in mathematical way.

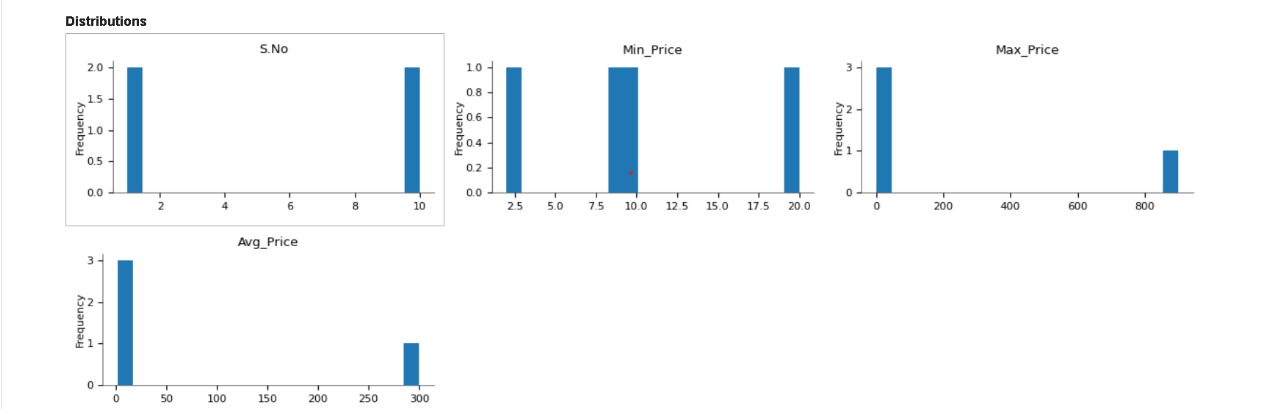
#Describing the Dataset using describe()

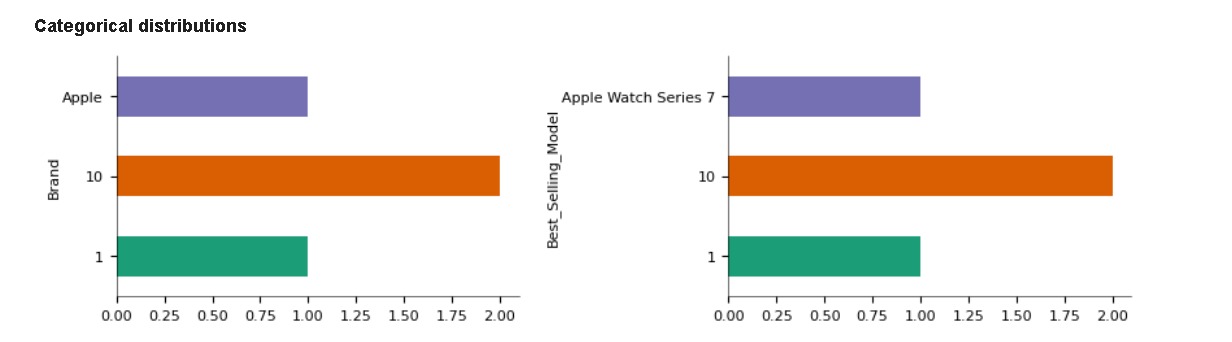
describe = cop.describe()

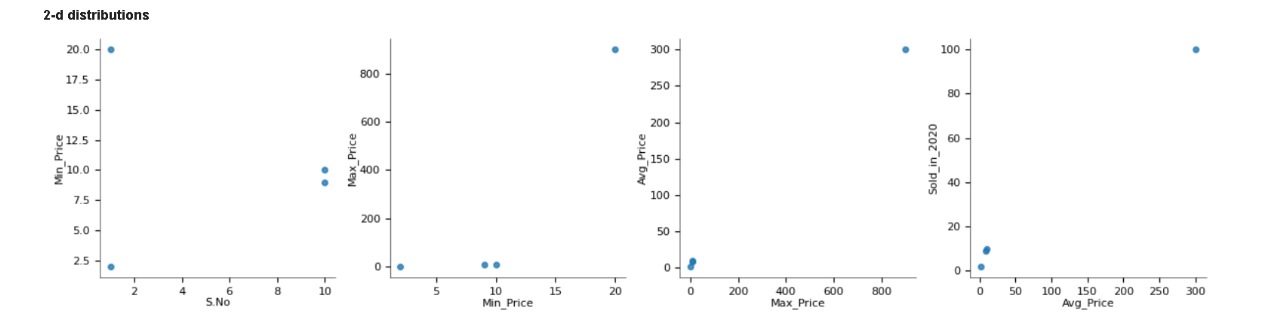
describe

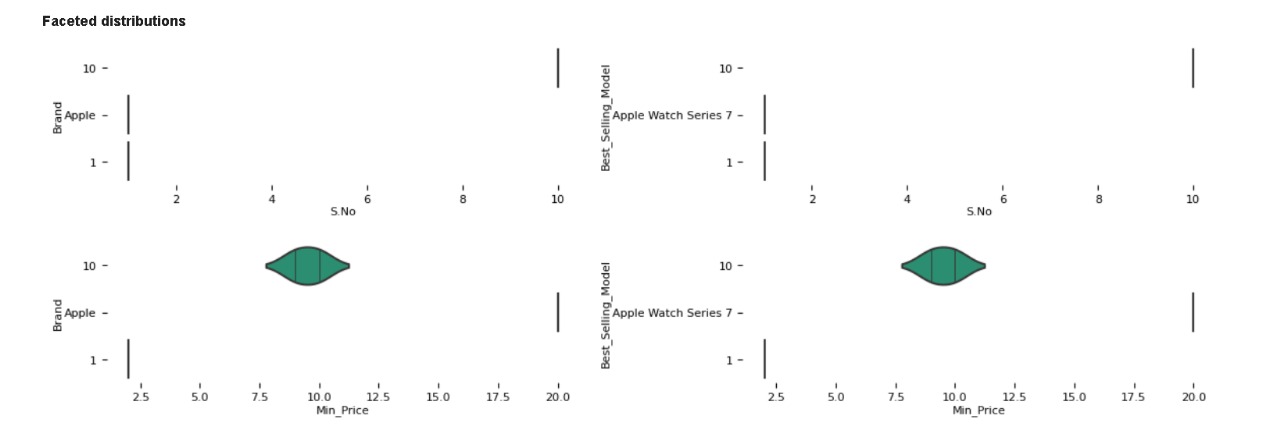


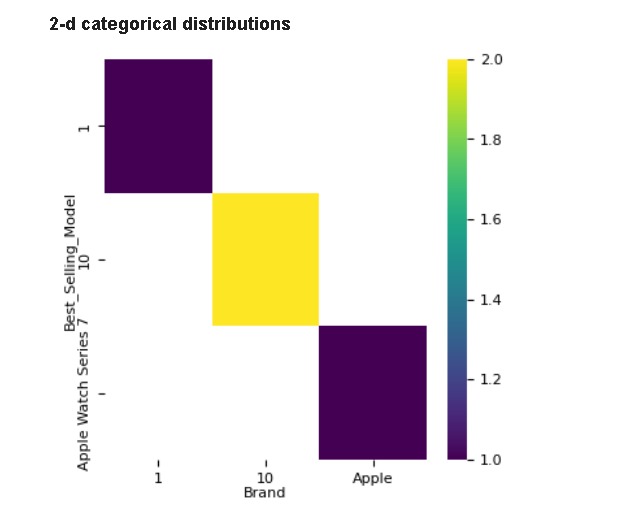












**Data Munging :**

It is process of filtering the data

**Head( ) :**  This method is used to get the records from top to bottom by default the head method returns the top 5 rows.

#Print the first 5 rows of Data using head()

head = cop.head()

print(head)

S.No Brand Min\_Price Max\_Price Avg\_Price Sold\_in\_2020 Sold\_in\_2021 \

0 1 Apple 90 900 200 100 150

1 2 Noise 9 40 20 350 450

2 3 Fossil 20 600 300 150 175

3 4 Titan 12 330 125 400 500

4 5 Wrogn 10 200 75 200 250

Sold\_in\_2022 Sold\_in\_2023 Best\_Selling\_Model

0 130 160 Apple Watch Series 7

1 400 425 Noise Pulse 2 Max Smartwatch

2 150 200 Fossil Fenmore Analog Black Dial Men's Watch

3 450 475 Titan Smart

4 230 250 Wrogn fitness smart band

**Tail( ) :** This method is used to get the records from bottom to up by default the tail method returns the bottom to up 5 rows.

#Print the last 5 rows of Data Using tail()

tail = cop.tail()

print(tail)

S.No Brand Min\_Price Max\_Price Avg\_Price Sold\_in\_2020 Sold\_in\_2021 \

5 6 Samsung 35 250 100 100 75

6 7 Google 50 700 300 50 50

7 8 Fastrack 20 150 70 300 250

8 9 Amazefit 5 60 30 250 300

9 10 Rolex 500 1000 600 30 25

Sold\_in\_2022 Sold\_in\_2023 Best\_Selling\_Model

5 80 100 Galaxy Watch 6 Classic hands-on

6 60 65 Google Pixel Watch (GPS)

7 275 290 Fastrack Reflex Vox

8 275 325 Amazfit T Rex Pro

9 25 35 Rolex Daytona

##### Rank( ) :

Gives a rank to the columns or entire data frame according to the ascending order or descending order for Numerical values , for alphabets it follows alphabetical order.

##### 

|  |
| --- |
| #Giving the Priority to values to Data using rank()  rank = cop.rank()  print(rank) |

##### S.No Brand Min\_Price Max\_Price Avg\_Price Sold\_in\_2020 Sold\_in\_2021 \

##### 0 1.0 2.0 9.0 9.0 7.0 3.5 4.0

##### 1 2.0 6.0 2.0 1.0 1.0 9.0 9.0

##### 2 3.0 4.0 5.5 7.0 8.5 5.0 5.0

##### 3 4.0 9.0 4.0 6.0 6.0 10.0 10.0

##### 4 5.0 10.0 3.0 4.0 4.0 6.0 6.5

##### 5 6.0 8.0 7.0 5.0 5.0 3.5 3.0

##### 6 7.0 5.0 8.0 8.0 8.5 2.0 2.0

##### 7 8.0 3.0 5.5 3.0 3.0 8.0 6.5

##### 8 9.0 1.0 1.0 2.0 2.0 7.0 8.0

##### 9 10.0 7.0 10.0 10.0 10.0 1.0 1.0

##### Sold\_in\_2022 Sold\_in\_2023 Best\_Selling\_Model

##### 0 4.0 4.0 2.0

##### 1 9.0 9.0 7.0

##### 2 5.0 5.0 4.0

##### 3 10.0 10.0 9.0

##### 4 6.0 6.0 10.0

##### 5 3.0 3.0 5.0

##### 6 2.0 2.0 6.0

##### 7 7.5 7.0 3.0

##### 8 7.5 8.0 1.0

##### 9 1.0 1.0 8.0

##### CORRELATION-

* It is a relation between two data column data members.
* We use method called corr( ) • It is scaled form of a covariance.
* Correlation values lies between (-1 to +1)

Attribute1.corr(attribute2)

**Types of correlations:**

we have three types of correlation.

* Positive – (0 to1)
* Negative – (0 to -1)
* No correlation – (0)

|  |
| --- |
| # Finding the Correlation between min and max price of watch using corr()  print(Data['Min\_Price'].corr(Data['Max\_Price'])) |

0.6864875223214113

**COVARIANCE-**

The covariance is the relation between two data members of two different columns.

• It is the measure of a correlation. • It lies between (-∞ to ∞).

|  |  |
| --- | --- |
| |  | | --- | | # Finding the Co-Variance between min and max price of watch using cov()  print(Data['Min\_Price'].cov(Data['Max\_Price'])) | |

#### 36521.88888888888

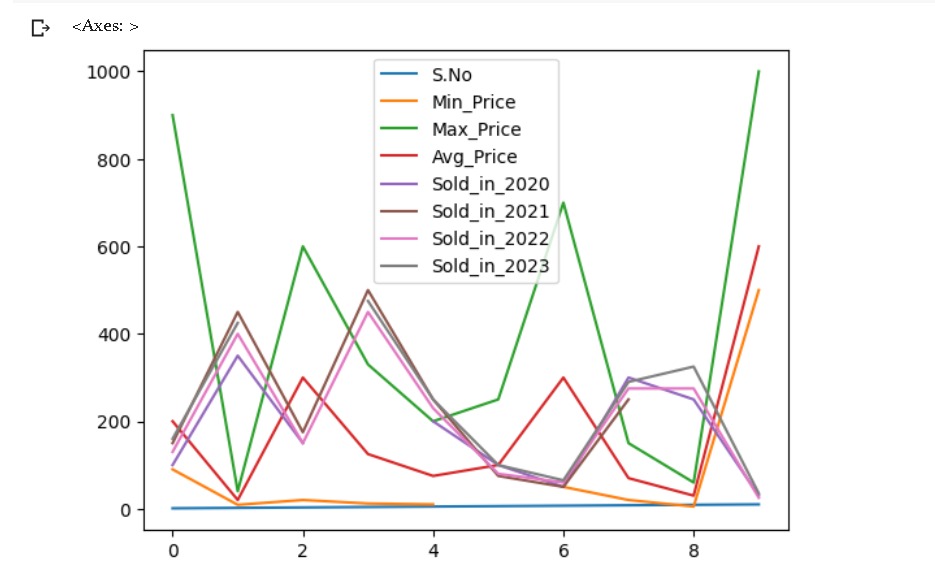
#### 5)VISUALIZING THE DATA & SHARING THE RESULT

Data visualization is a process of representing data in a graphical way. Here we can represent the below graph formats.

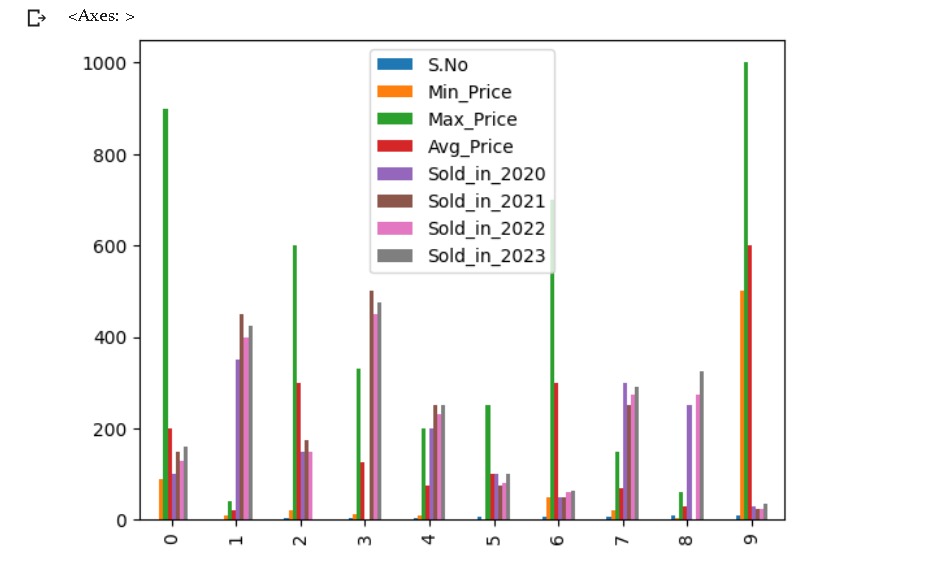
1. Line Graph
2. Bar Graph
3. Box Graph
4. KDE Graph
5. Area Graph
6. Histogram Graph

Here we need to use matplotlib module to represent graphs using the code.

|  |
| --- |
| # Line Graph  line\_Graph = Data.plot.line()  line\_Graph |



|  |
| --- |
| # Bar Graph  bar\_graph = Data.plot.bar()  bar\_graph |



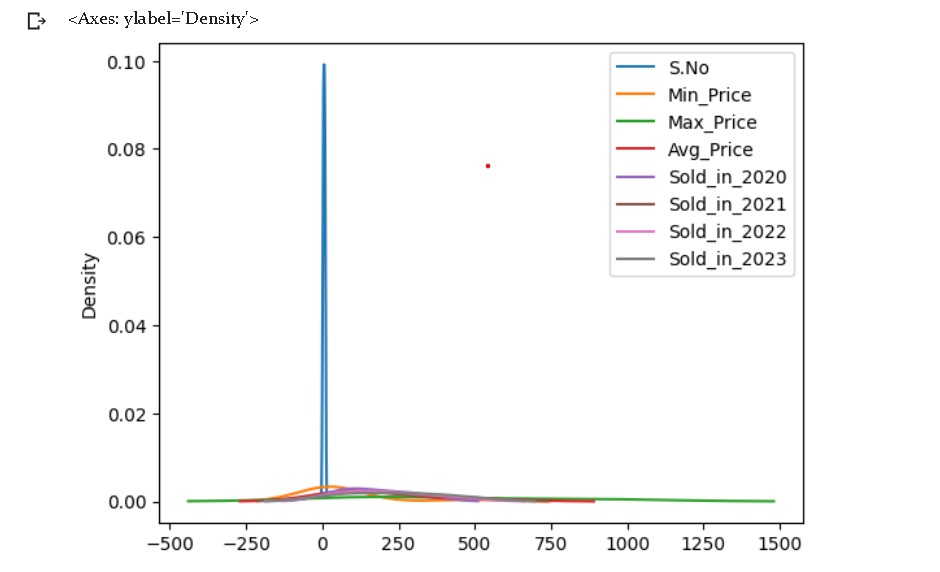
|  |
| --- |
| # Box Graph  Box = Data.plot.box()  Box |

|  |  |
| --- | --- |
|  |  |

# kde graph

kde = Data.plot.kde()

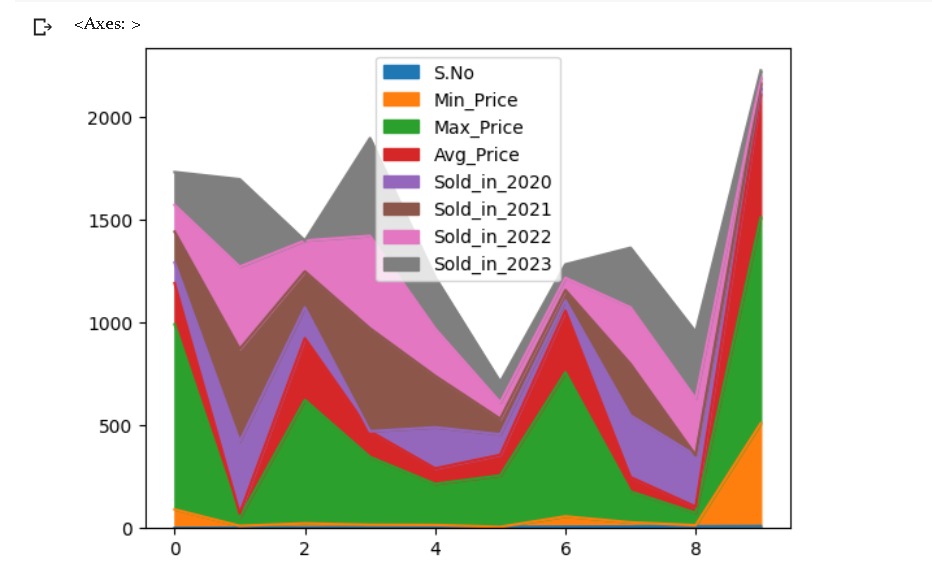
kde



# Area Graph

area\_graph = Data.plot.area()

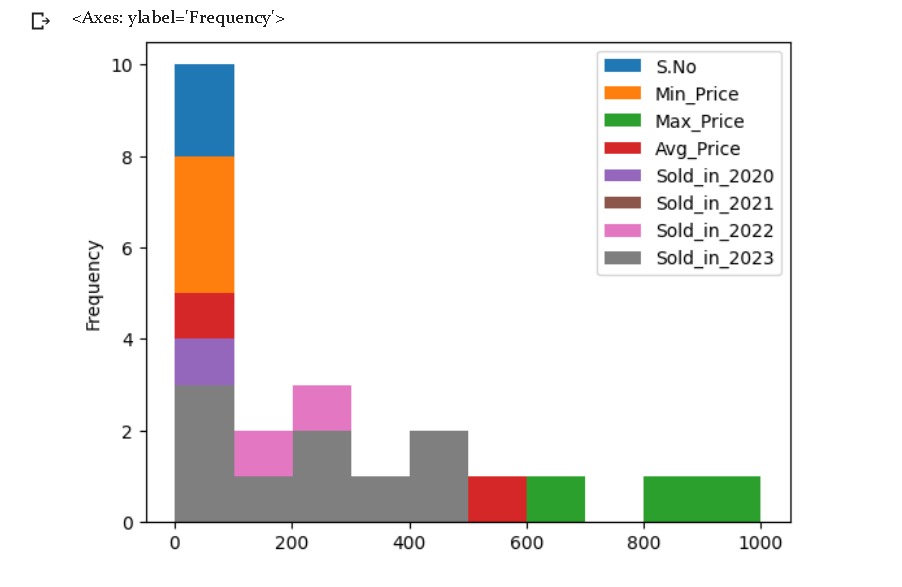
area\_graph



# Histogram

hist = Data.plot.hist()

hist



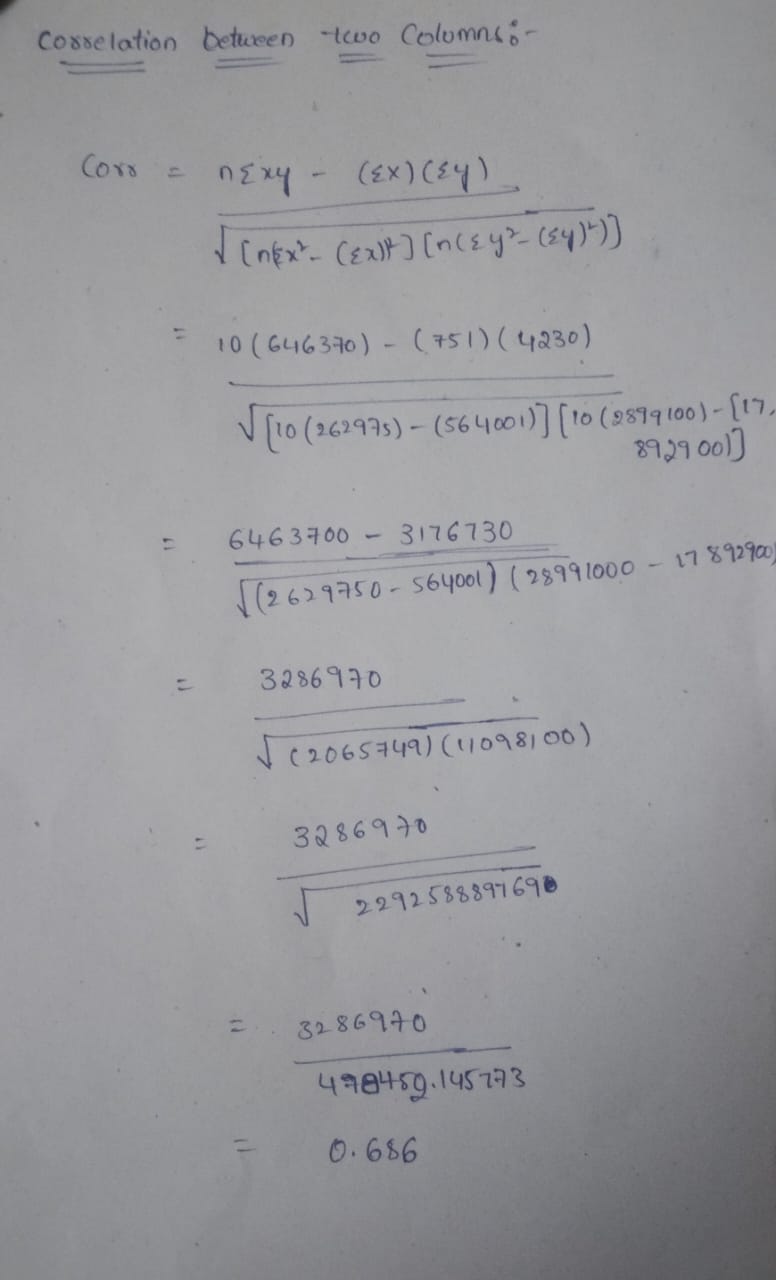
# scatter plot

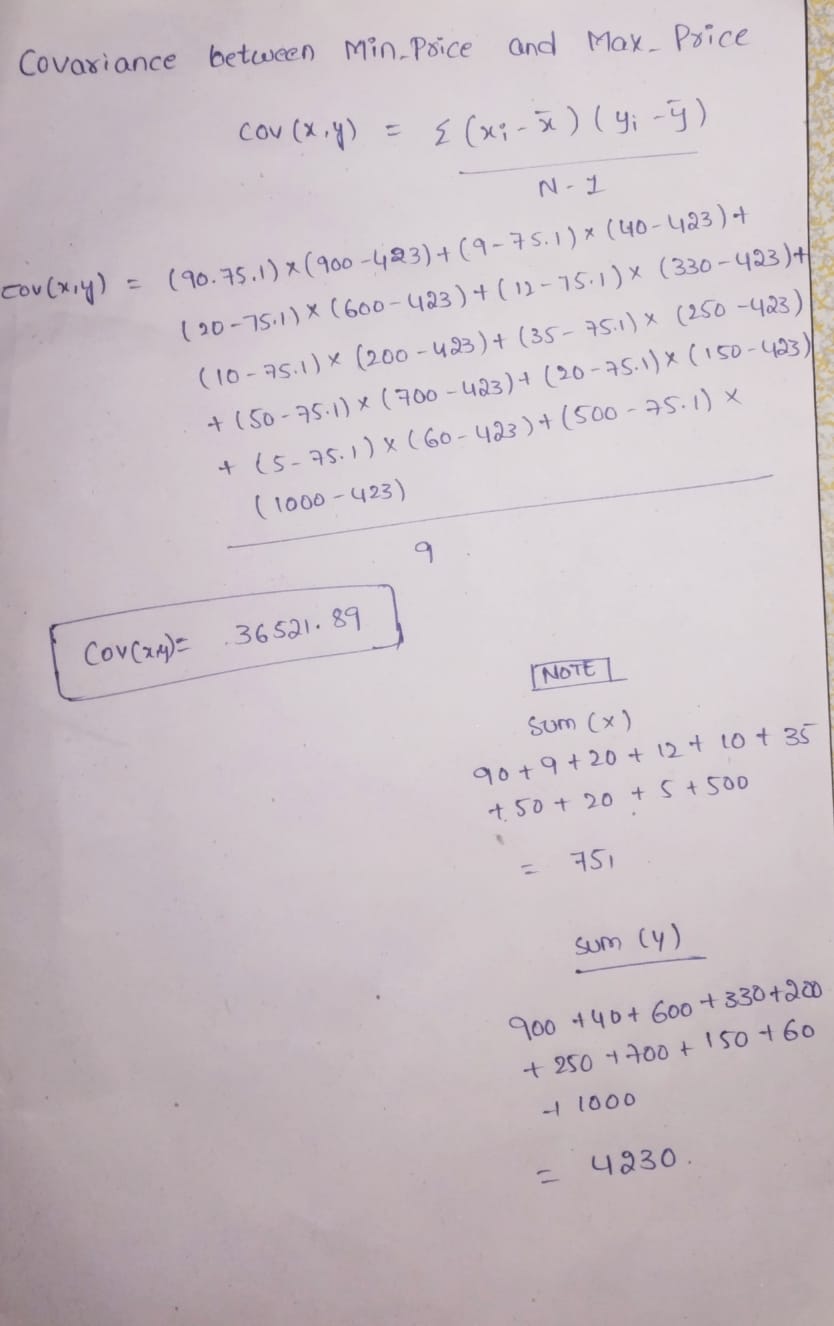
scatter = Data.plot.scatter('Min\_Price','Max\_Price')

scatter

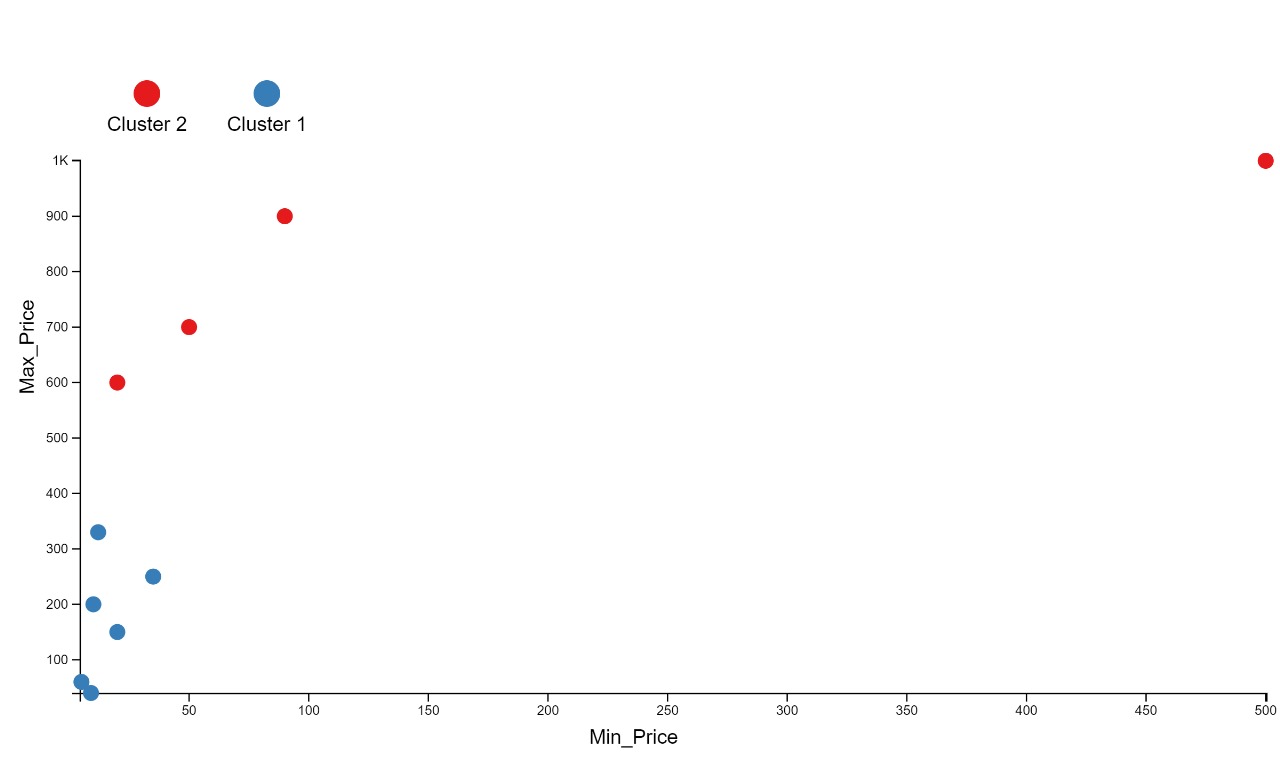
#### EMBRACING THE RESULTS

CORELATION AND CO-VARIANCE ARE MANUALLY CALCULATED

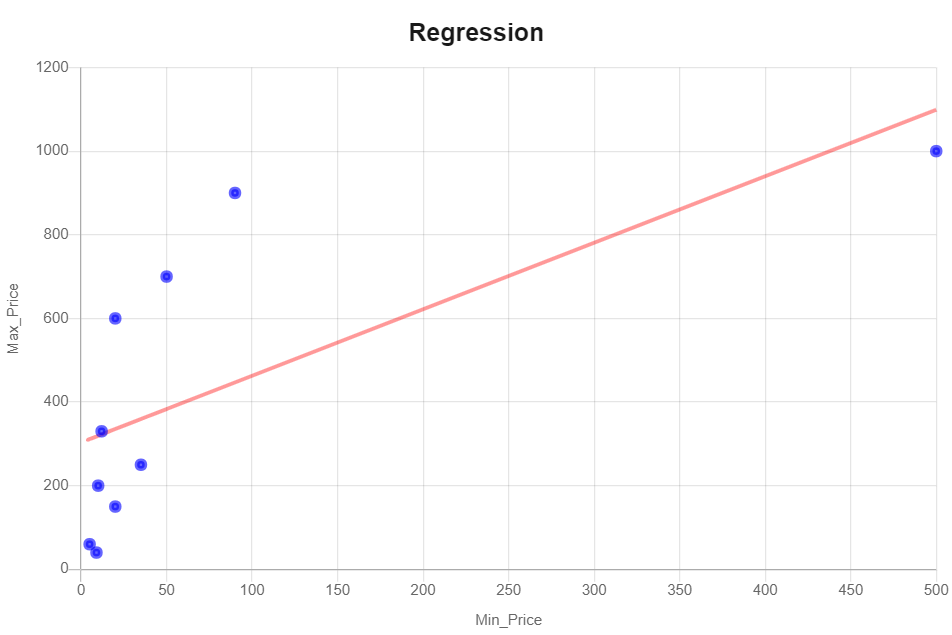


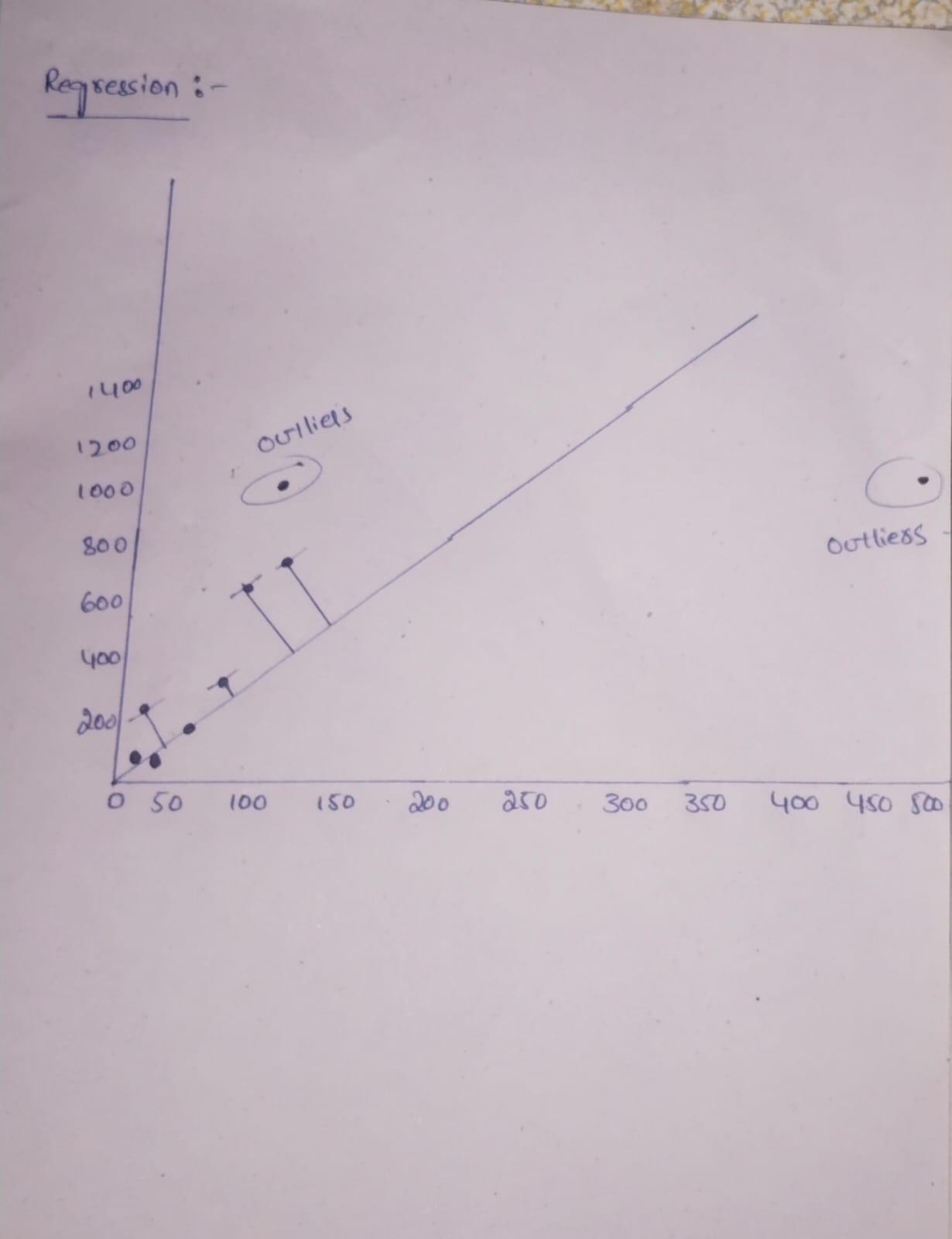


**Clustering :**



**Regression :**

****



#### SUMMARY

In this project, we aimed to develop a smartwatch price prediction using data analytics, with a focus on time series analysis techniques implemented in Python.

In summary, a smartwatch price prediction project involves data collection, preprocessing, modeling, and evaluation to create a predictive tool for estimating smartwatch prices, benefiting both consumers and businesses in the smartwatch industry.

**Benefits:**

* Consumers can make more informed purchasing decisions by estimating the fair price of a smartwatch.
* Businesses can optimize their pricing strategies to remain competitive in the market.
* Researchers and analysts can gain insights into factors affecting smartwatch prices and market trends.

**Challenges:**

* The accuracy of price predictions depends on the quality and quantity of data available.
* Smartwatch prices may be influenced by various external factors that are challenging to quantify.
* Ensuring the model's relevance over time requires continuous data updates and monitoring.