BY:

Tanishka Shah

2nd Year CSE

Walchand Institute of Technology

	Month Starting	Open	High	Low	Close	Change %	Avg. Volume
0	Dec. 01, 2022	101.38	102.59	100.67	101.28	-0.17	21771536.0
1	Nov. 01, 2022	95.59	101.45	83.45	101.45	7.17	28294944.0

```
mp1.info()
```

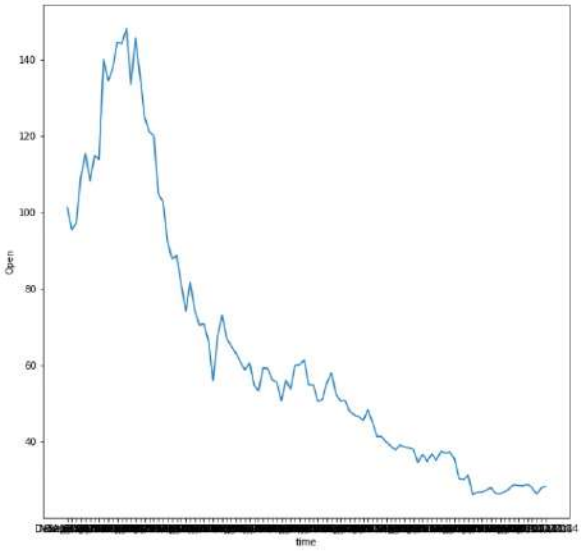
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Month Starting  106 non-null    object 
 1   Open            106 non-null    float64
 2   High            106 non-null    float64
 3   Low             106 non-null    float64
 4   Close           106 non-null    float64
 5   Change %        106 non-null    float64
 6   Avg. Volume     106 non-null    float64
dtypes: float64(6), object(1)
memory usage: 5.9+ KB
```

```
mp1.iloc[:,1:].describe()
```

	Open	High	Low	Close	Change %	Avg. Volume
count	106.000000	106.000000	106.000000	106.000000	106.000000	1.060000e+02
mean	63.558019	67.674151	60.285283	64.184623	1.432925	3.299223e+07
std	34.705462	36.999070	32.339007	34.598918	6.513333	9.553159e+06
min	26.240000	27.070000	24.380000	26.030000	-17.670000	4.321920e+05
25%	37.202500	38.947500	34.660000	37.412500	-2.090000	2.745647e+07
50%	54.925000	58.025000	50.665000	54.950000	1.150000	3.088268e+07
75%	79.640000	86.082500	72.647500	81.545000	5.690000	3.654029e+07
max	148.170000	152.100000	142.070000	148.270000	20.190000	6.490765e+07

```
plt.figure(figsize=(10,10))
plt.plot(mp1['Month Starting'], mp1['Open'])
plt.xlabel('time')
plt.ylabel('Open')
```

```
Text(0, 0.5, 'Open')
```



From the above graph we can infer that:

1. This graph (From latest to earliest) depicts the opening Market Trends of the stocks.
2. We can see a steady growth in this graph but we have seen a downward trend in the graph in the past few months.
3. This decline is probably due to covid surges and other industrial recession.

```
plt.figure(figsize=(10,10))
plt.plot(mp1['Month Starting'], mp1['Open'])
plt.plot(mp1['Month Starting'], mp1['Close'])
plt.bar(mp1['Month Starting'], mp1['Change %'])
plt.xlabel('time')
plt.ylabel('Open')
plt.ylabel('Close')
plt.ylabel('Change %')
```

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

```
mp1=pd.read_csv('/content/Google Dataset.csv')
mp1
```

	Month Starting	Open	High	Low	Close	Change %	Avg. Volume
0	Dec. 01, 2022	101.38	102.59	100.67	101.28	-0.17%	21771536.0
1	Nov. 01, 2022	95.59	101.45	83.45	101.45	7.17%	28294944.0
2	Oct. 03, 2022	97.22	105.10	91.90	94.66	-1.55%	27843110.0
3	Sep. 01, 2022	109.20	112.64	96.03	96.15	-11.91%	25381194.0
4	Aug. 01, 2022	115.53	123.26	108.80	109.15	-6.42%	18737451.0
...	...	...	...	...	...	...	...
101	Jul. 01, 2014	28.92	29.98	28.25	28.58	-0.64%	31411358.0
102	Jun. 02, 2014	28.03	29.12	26.94	28.76	2.75%	36121936.0
103	May. 01, 2014	26.35	28.39	25.16	27.99	6.31%	34808252.0
104	Apr. 01, 2014	27.93	30.24	25.14	26.33	-5.44%	64037909.0
105	Mar. 27, 2014	28.40	28.40	27.65	27.85	0.00%	432192.0

106 rows × 7 columns

```
mp1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Month Starting  106 non-null    object 
 1   Open            106 non-null    float64
 2   High            106 non-null    float64
 3   Low             106 non-null    float64
 4   Close           106 non-null    float64
 5   Change %        106 non-null    object 
 6   Avg. Volume     106 non-null    float64
dtypes: float64(5), object(2)
memory usage: 5.9+ KB
```

```
mp1.shape
```

```
(106, 7)
```

```
mp1.size
```

```
742
```

```
mp1.isnull().sum()
```

```
Month Starting    0
Open              0
High              0
Low               0
Close             0
Change %          0
Avg. Volume       0
dtype: int64
```

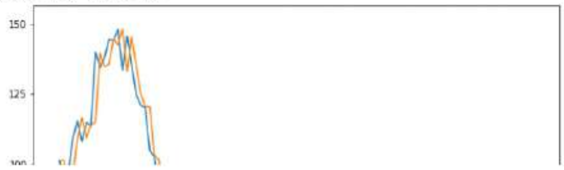
```
mp1['Change %'] = mp1['Change %'].str.replace('%','')
mp1
```

	Month Starting	Open	High	Low	Close	Change %	Avg. Volume
0	Dec. 01, 2022	101.38	102.59	100.67	101.28	-0.17	21771536.0
1	Nov. 01, 2022	95.59	101.45	83.45	101.45	7.17	28294944.0
2	Oct. 03, 2022	97.22	105.10	91.90	94.66	-1.55	27843110.0
3	Sep. 01, 2022	109.20	112.64	96.03	96.15	-11.91	25381194.0
4	Aug. 01, 2022	115.53	123.26	108.80	109.15	-6.42	18737451.0
...	...	...	...	...	...	...	...
101	Jul. 01, 2014	28.92	29.98	28.25	28.58	-0.64	31411358.0
102	Jun. 02, 2014	28.03	29.12	26.94	28.76	2.75	36121936.0
103	May. 01, 2014	26.35	28.39	25.16	27.99	6.31	34808252.0
104	Apr. 01, 2014	27.93	30.24	25.14	26.33	-5.44	64037909.0
105	Mar. 27, 2014	28.40	28.40	27.65	27.85	0.00	432192.0

106 rows × 7 columns

```
mp1['Change %'] = mp1['Change %'].astype({"Change %":'object', "Change %":'float64'})
mp1
```

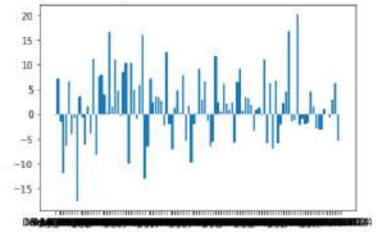
```
Text(0, 0.5, 'Change %')
```



1. This graph (From latest to earliest) depicts the opening Market as well as the closing market Trends of the stocks comparatively.
2. We can see a steady growth in this graph but we have also seen a downward trend in the graph in the past few months due to various circumstances
3. The stock analytics of opening and closing market trends is almost similar.
4. The blue bar plot (detailed be;ow) are decipeting the values of %change of stocks.

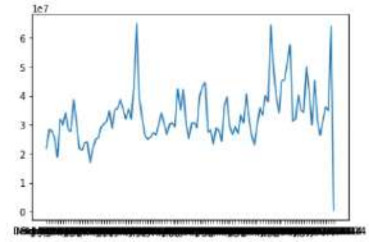
```
plt.bar(mp1['Month Starting'], mp1['Change %'])
```

```
<BarContainer object of 106 artists>
```



```
plt.plot(mp1['Month Starting'], mp1['Avg. Volume'])
```

```
[<matplotlib.lines.Line2D at 0x7fe855914400>]
```



1. data is not rightly skewed
2. it is highly unstable as mean deviation is very high

```
from sklearn import preprocessing
from sklearn import utils
#now lets try Linear Regression
#1.slicing-divide data into input(x),output(y)
y=mp1.iloc[:,5].values

#convert y values to categorical values
lab = preprocessing.LabelEncoder()
y = lab.fit_transform(y)
y
```

```
array([ 42,  84,  32,   2,  10,  81,  18,  37,   0,  69,  39,  11,  55,
        19,  96,   5,  86,  88,  71,  44, 100,  55,  94,  74,  40,  89,
        93,   3,  92,  75,  35,  76,  99,   1,   8,  85,  62,  70,  68,
        63,  24,  98,  27,   6,  52,  73,  46,  87,  17,  56,   4,  28,
        45,  90,  65,  82,  33,   9,  15,  97,  61,  47,  77,  58,  50,
        60,  14,  80,  91,  48,  67,  66,  57,  20,  49,  53,  41,  95,
        13,  78,   7,  83,  12,  26,  59,  72, 101,  31,  34, 102,  25,
        36,  29,  30,  72,  54,  23,  22,  21,  51,  43,  38,  64,  79,
        16,  43])
```

```
x=mp1.iloc[:,1:5].values
x
```

```
array([[101.38, 102.59, 100.67, 101.28],
       [ 95.59, 101.45,  83.45, 101.45],
       [ 97.22, 105.1 ,  91.9 ,  94.66],
       [109.2 , 112.64,  96.03,  96.15],
       [115.53, 123.26, 108.8 , 109.15],
       [108.34, 120.44, 104.76, 116.64],
       [114.93, 119.4 , 105.63, 109.37],
       [113.91, 123.14, 102.21, 114.04],
       [140.01, 144.04, 113.12, 114.97],
       [134.48, 144.16, 125.86, 139.65],
       [137.84, 152.1 , 124.76, 134.89],
       [144.48, 146.61, 124.64, 135.7 ],
       [144.21, 149.61, 140.25, 144.68],
       [148.17, 151.85, 142.07, 142.45],
       [133.55, 149.12, 131.17, 148.27],
       [145.65, 146.82, 133.  , 133.27],
       [135.48, 146.49, 134.18, 145.46],
       [124.85, 140.01, 124.85, 135.22],
       [121.1 , 127.8 , 119.14, 125.32],
       [120.14, 122.15, 111.5 , 120.58],
       [104.9 , 122.62, 104.84, 120.51],
       [102.83, 106.44, 100.5 , 103.43],
       [ 92.68, 107.63,  92.58, 101.84],
       [ 87.88,  96.74,  84.95,  91.79],
       [ 88.72,  92.36,  84.95,  87.59],
       [ 81.41,  90.9 ,  80.8 ,  88.04],
       [ 74.21,  84.35,  71.8 ,  81.05],
       [ 81.83,  86.66,  70.33,  73.48],
       [ 74.33,  82.96,  72.93,  81.71],
       [ 70.56,  79.35,  70.49,  74.15],
       [ 70.92,  73.8 ,  67.35,  70.68],
       [ 66.43,  72.05,  64.95,  71.45],
       [ 56.1 ,  68.  ,  53.99,  67.43],
       [ 67.58,  70.51,  50.68,  58.14],
       [ 73.1 ,  76.61,  63.55,  66.97],
       [ 67.08,  75.16,  67.08,  71.71],
       [ 65.05,  68.25,  63.95,  66.85],
       [ 63.25,  66.78,  63.03,  65.25],
       [ 60.95,  64.97,  58.12,  63.01],
       [ 58.85,  62.42,  58.16,  60.95],
       [ 60.7 ,  61.71,  57.01,  59.41],
       [ 54.9 ,  63.28,  54.69,  60.83],
       [ 53.28,  56.21,  51.25,  54.05],
       [ 59.4 ,  59.54,  55.01,  55.18],
       [ 59.21,  64.46,  58.75,  59.42],
       [ 56.25,  61.59,  56.17,  58.67],
       [ 55.62,  57.35,  54.3 ,  56.  ],
       [ 50.83,  55.87,  50.7 ,  55.82],
       [ 56.16,  56.23,  48.51,  51.78],
       [ 53.79,  54.78,  49.8 ,  54.72],
       [ 59.99,  60.5 ,  49.79,  53.84],
       [ 60.21,  60.65,  57.35,  59.67],
       [ 61.4 ,  62.83,  59.41,  60.91],
       [ 54.95,  63.69,  54.69,  60.86],
       [ 54.97,  59.31,  54.8 ,  55.78],
       [ 50.68,  55.54,  50.31,  54.25],
       [ 51.14,  54.71,  49.52,  50.87],
       [ 55.39,  58.85,  49.03,  51.59],
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=42)
print(x.shape)            #100%
print(x_train.shape)      #75%
print(x_test.shape)       #25%

print(y.shape)            #100%
print(y_train.shape)      #75%

print(y_test.shape)       #25%
```

```
(106, 4)
(79, 4)
(27, 4)
(106,)
(79,)
(27,)
```

```
#normalisation/scaling(done only for input)
#from sklearn.preprocessing import StandardScaler
#scaler = StandardScaler()
#x_train = scaler.fit_transform(x_train)
#x_test = scaler.fit_transform(x_test)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
model = make_pipeline(StandardScaler(), LogisticRegression())
```

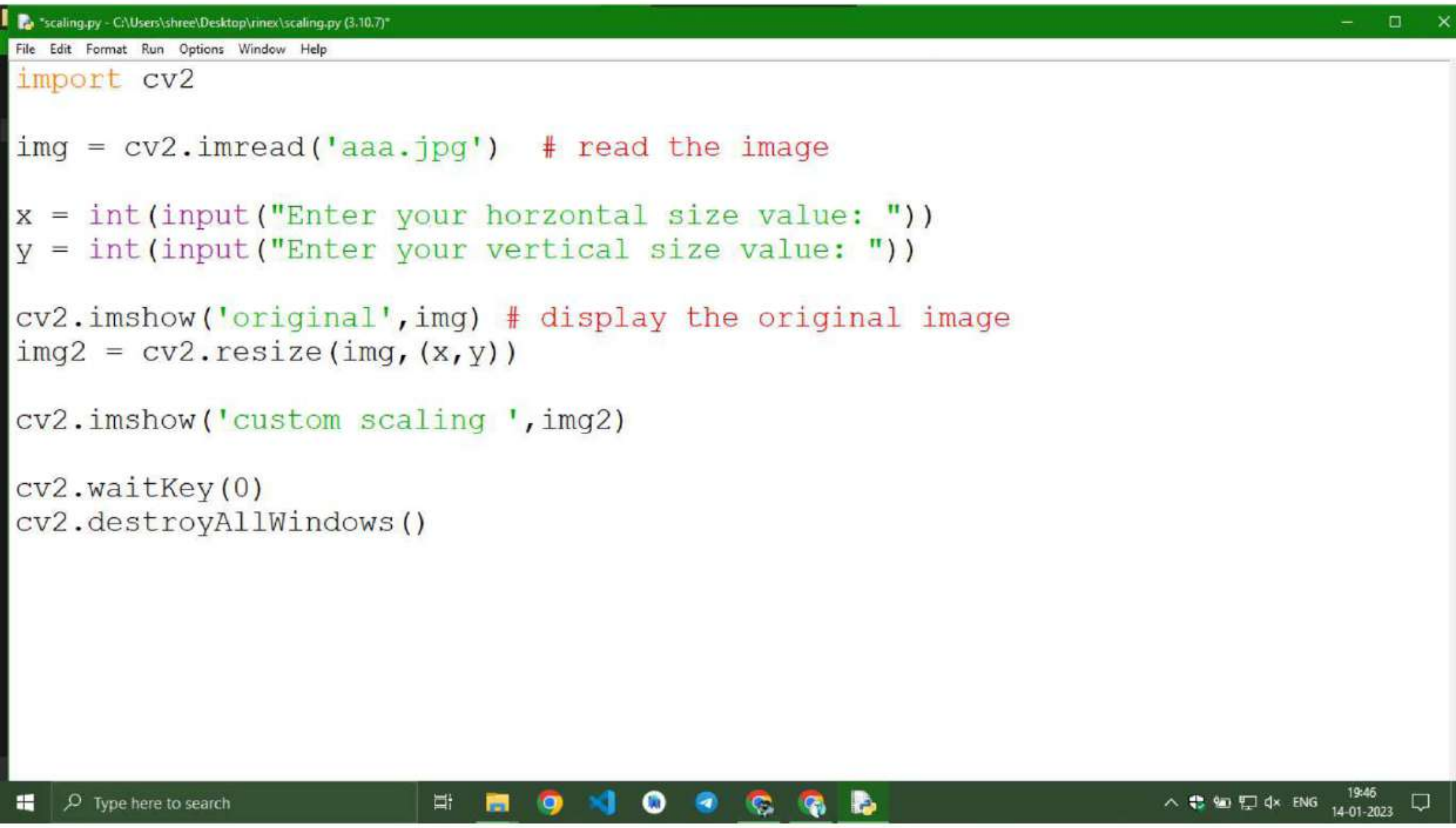
```
#fit the model
model.fit(x_train,y_train)
```

```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('logisticregression', LogisticRegression())])
```

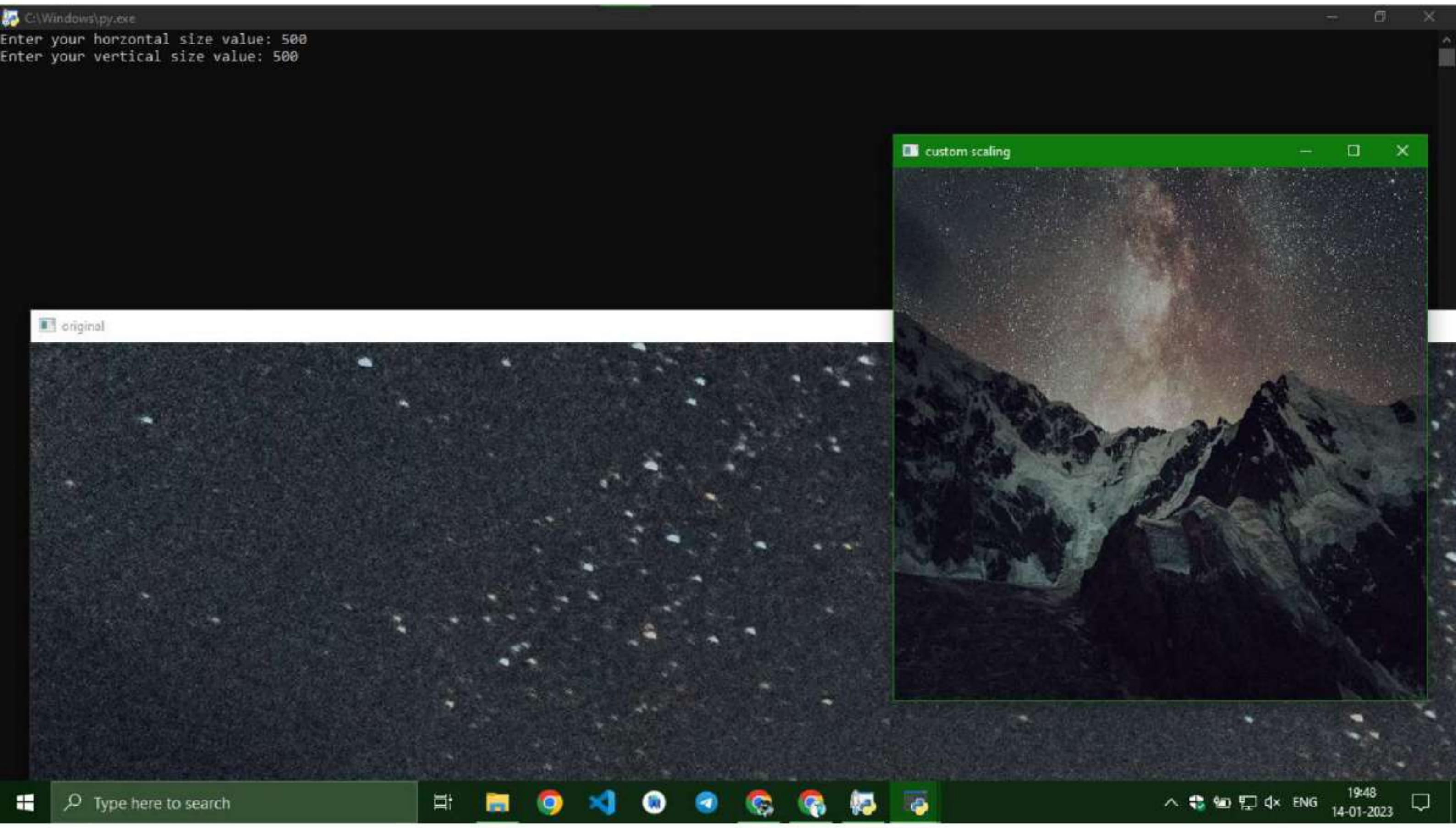
```
#predict the output
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
y_pred
```

```
array([ 72,  19,   0,  72,  72,  72,   3,   8,  72,  19,  72,  72,  44,
        72,  86,  72,  72,  72,  72,  70,  19,  72,  72,  89, 100,  92,
        72])
```

Creating an opencv project to change the dimensions of the image as defined by the user.


scaling.py - C:\Users\shree\Desktop\rinex\scaling.py (3.10.7)
File Edit Format Run Options Window Help
Type here to search
ENG
19:46
14-01-2023

```python
import cv2

img = cv2.imread('aaa.jpg')  # read the image

x = int(input("Enter your horzontal size value: "))
y = int(input("Enter your vertical size value: "))

cv2.imshow('original',img) # display the original image
img2 = cv2.resize(img,(x,y))

cv2.imshow('custom scaling ',img2)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



```
Enter your horzontal size value: 500
Enter your vertical size value: 500
```