

Developing an application using pyqt

1.Installation:

To develop a PyQt5 application the required installations in ubuntu :

1. python3 should be installed by typing command in terminal.
2. install PyQt5 by typing command in terminal.
3. install numpy for numerical calculation in datascience by using command.
4. install pandas for data manipulation, visualization by using command.
5. install matplotlib for visualization (plotting) by using command.

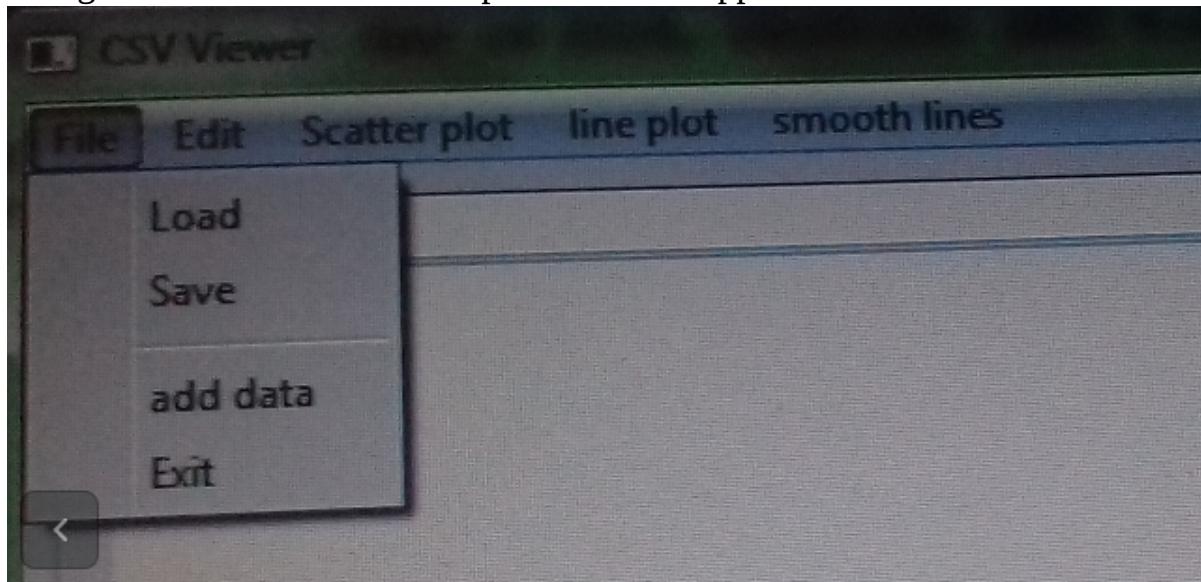
For your kind reference , To install all this follow below figure.

```
student@student-TravelMate-P243-M: ~
student@student-TravelMate-P243-M:~$ sudo apt-get install python3
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.5.1-3).
The following packages were automatically installed and are no longer required:
  libevent-core-2.0-5 libllvm4.0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
student@student-TravelMate-P243-M:~$ sudo apt-get install python qt5
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package qt5
student@student-TravelMate-P243-M:~$ sudo apt-get install numpy==1.9.3
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package numpy=
student@student-TravelMate-P243-M:~$ sudo apt-get install pandas
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package pandas
student@student-TravelMate-P243-M:~$ sudo apt-get install matplotlib
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package matplotlib
student@student-TravelMate-P243-M:~$
```

Here comes the application developing process.

2.Developing code:

Following functionalities should be present in the application.



1.Load a csv file using “Load” option available under “File” menu.

Description: After clicking Load button user can able to load the CSV file.

Code:

- i) By using `menubar.addMenu("File")` we can add “File” menu
- ii) By using `self.filemenu.addAction(QIcon.fromTheme("document-open"), "Load", self.loadCsv, QkeySequence.Open)` we can perform action.
- iii) By using `QFileDialog.getOpenFileName(self, "Open CSV", QDir.homePath() + "/Dokumente/CSV", "CSV (*.csv *.tsv *.txt)")` we can open csv file by using path provided.

2.Display the complete data from the loaded csv as a table:

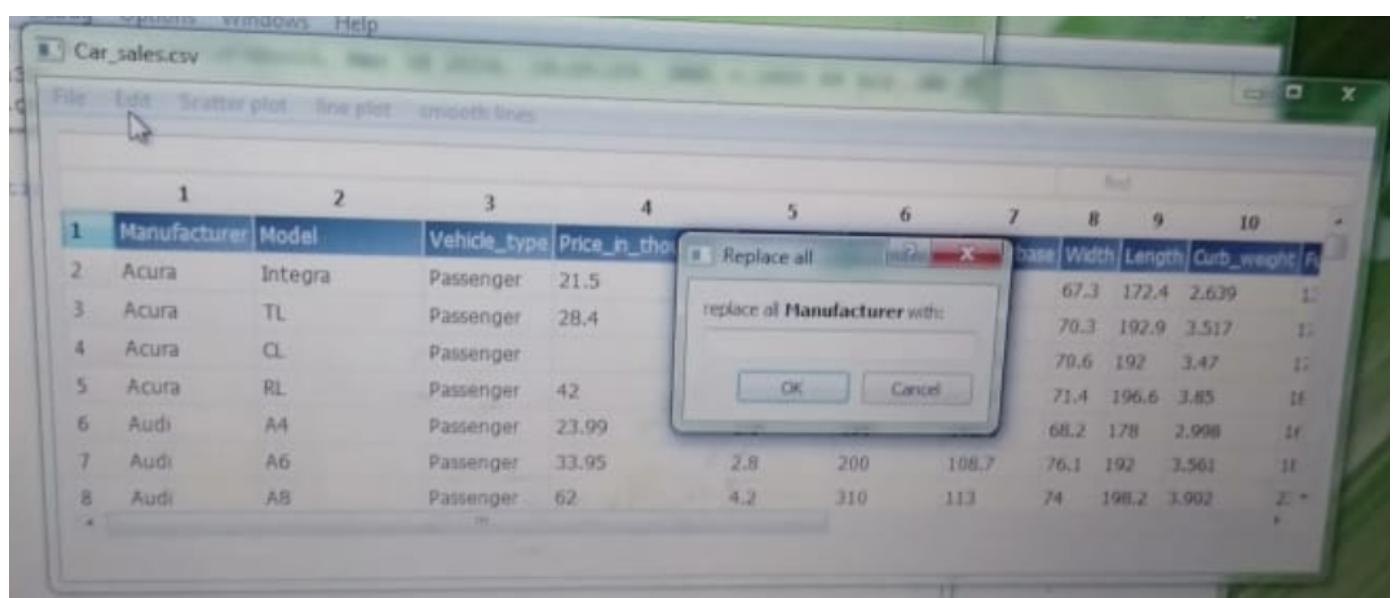
		3	4	5	6	7	8	9	
3	mid	RF	L-O2	L-BP	SURF-STBL	CORE-STBL	BP-STBL	COMFORT	ADM-DECS
3	mid	high	excellent	mid	stable	stable	stable	15	A
4	high	low	excellent	high	stable	stable	mod-stable	10	A
5	mid	low	good	high	stable	unstable	mod-stable	15	A
6	mid	mid	excellent	high	stable	stable	stable	10	A
7	high	low	good	mid	stable	stable	unstable	15	S
8	mid	low	excellent	high	stable	stable	mod-stable	5	S
9	high	mid	excellent	mid	unstable	unstable	stable	10	c

Description: After clicking Load button user can able to display the CSV file as a table.

Code:

- i) By using `QtWidgets.QtableView(self)` we can provide table structure.
- ii) By using `setStyleSheet(stylesheets(self)), setModel(self.model),
horizontalHeader().setStretchLastSection(True), setShowGrid(True)
, setGeometry(), dataChanged.connect(self.finishedEdit)`
we can set table dimensions as per our requirements.
- iii) By using `self.model.appendRow(items), self.tableView.
resizeColumnsToContents()` rows and columns to table.

3.Edit the existing data in the table using the ‘Edit data’ option under the “Edit” menu:

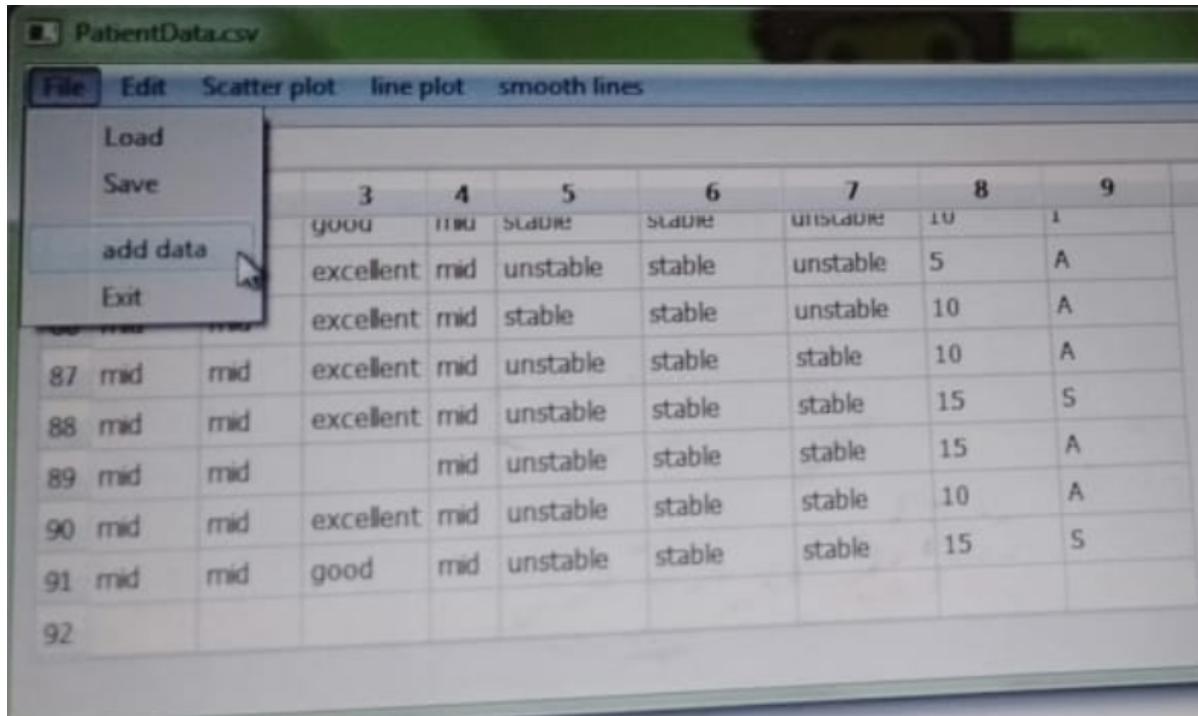


Description: After clicking Edit button user can able to Edit the data.

Code:

- i) By using `menubar.addAction("Edit")` we can add “Edit” menu
- ii) By using `self.editmenu.addAction(QIcon.fromTheme("edit-edit"),
"Edit", self.replaceThis)` we can perform action.
- iii) By using `QInputDialog.getText(self, "Replace all", "replace all " + mytext + " with:", QLineEdit.Normal, "", Qt.Dialog)`
we can edit the content.

4.Add new data to the table using ‘Add data’ option under “File” menu:



The screenshot shows a CSV file named "PatientData.csv" being edited. The 'File' menu is open, and the 'add data' option is highlighted with a mouse cursor. The main window displays a table with columns labeled 3 through 9 and rows numbered 3 to 92. The data includes various categories like 'good', 'mid', 'stable', 'unstable', and 'A' or 'S'.

			3	4	5	6	7	8	9
3	good	mid	stable	stable	unstable	10	1		
4	excellent	mid	unstable	stable	unstable	5	A		
5	excellent	mid	stable	stable	unstable	10	A		
6	mid	mid	excellent	mid	unstable	stable	stable	10	A
7	mid	mid	excellent	mid	unstable	stable	stable	15	S
8	mid	mid		mid	unstable	stable	stable	15	A
9	mid	mid	excellent	mid	unstable	stable	stable	10	A
10	mid	mid	good	mid	unstable	stable	stable	15	S
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									
51									
52									
53									
54									
55									
56									
57									
58									
59									
60									
61									
62									
63									
64									
65									
66									
67									
68									
69									
70									
71									
72									
73									
74									
75									
76									
77									
78									
79									
80									
81									
82									
83									
84									
85									
86									
87									
88									
89									
90									
91									
92									

Description: After clicking Edit button user can able to Edit the data.

Code:

- i)By using `self.editmenu.addAction(QIcon.fromTheme("add"), "add data", self.addRow)` We can perform action on data.
- ii)By using `self.tableView.setRowCount(count + 1)` by getting row count if row count is ==0 then we add columns by using `self.addColumn()`.

5.Select any number of columns from the displayed table:



The screenshot shows a CSV file named "Car_sales.csv" being edited. The 'Edit' menu is open, and the 'select columns' option is highlighted with a mouse cursor. The main window displays a table with columns labeled 3 through 10 and rows numbered 1 to 8. The data includes vehicle types like "Passenger" and numerical values for price and engine size.

		3	4	5	6	7	8	9	10
1	Price	Undo	Ctrl+Z						
2		Redo	Ctrl+Y						
3	Acura	TL	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length
4	Acura	CL	Passenger	21.5	1.8	140	101.2	67.3	172.4
5	Acura	RL	Passenger	28.4	3.2	225	108.1	70.3	192.9
6	Audi	A4	Passenger	42	3.5	210	114.6	71.4	196.6
7	Audi	A6	Passenger	23.99	1.8	150	102.6	68.2	178
8	Audi	A8	Passenger	33.95	2.8	200	108.7	76.1	192

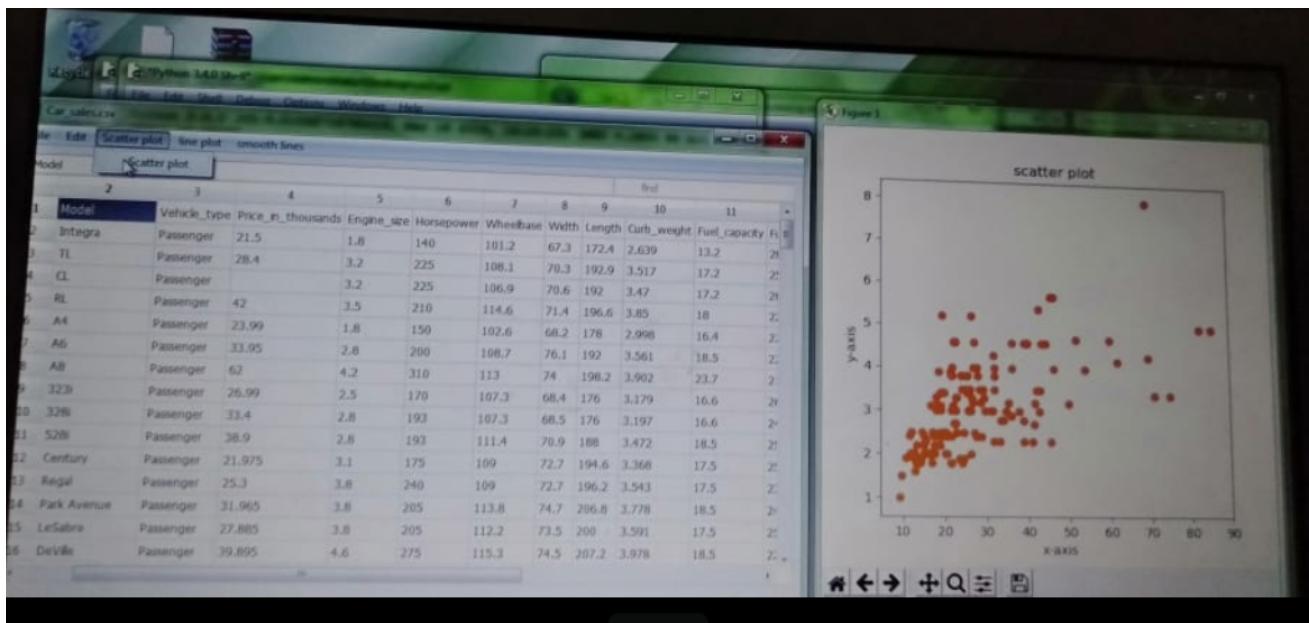
Description: After clicking select column button user can able to select column from the table ,if we want to select more than one column again column should be selected and then click on select column button.

Code:

- i)By using `self.editmenu.addAction(QIcon.fromTheme("Scatter"),"select columns",self.copyByContext)`We can perform action on data.
- ii)By using `QApplication.clipboard().setText (self.tableView.item(row,col).text())` we can select select the specified number of columns.

6.Plot the data from any two selected columns should be available as buttons as mentioned below:

- i)Plot scatter points:



Description: After selecting more than 2 columns one has to select the scatter plot button than we can able to plot the scatter plot.

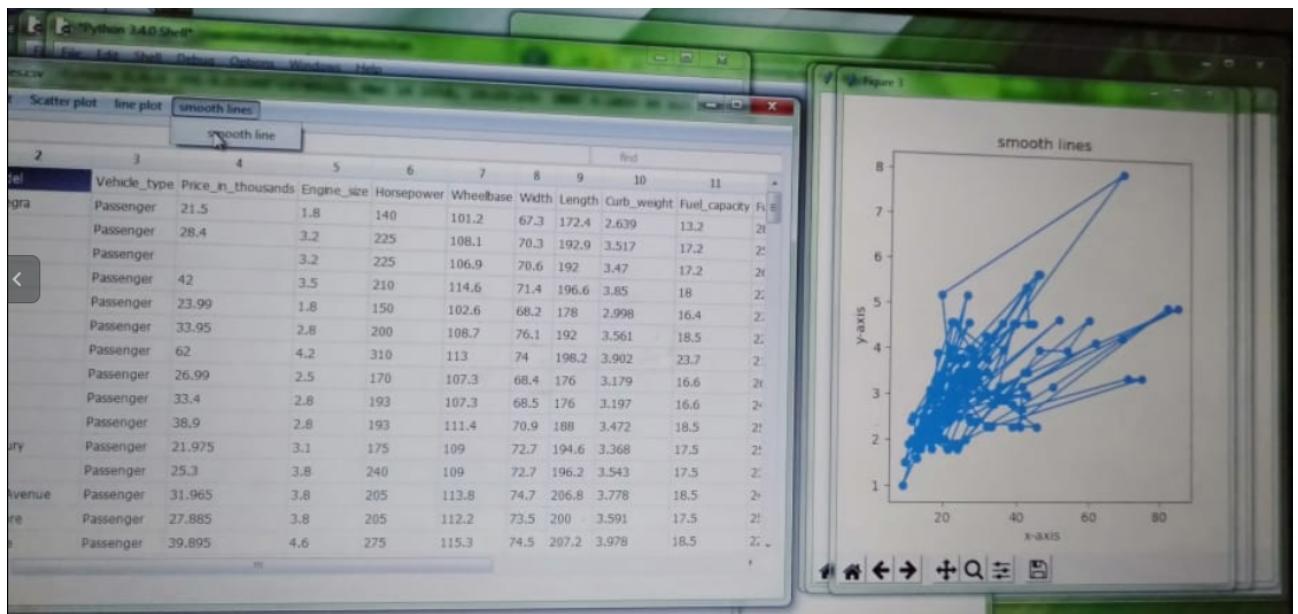
Code:

- i)By using `self.editmenu1=bar.addMenu("Scatter plot")` we can create the button.
- ii)By using `self.editmenu1.addAction(QIcon.fromTheme("Scatter","Scatter plot", self.plot))` we can perform action.

iii) By using

```
train_data = pd.read_csv(z)
plt.scatter(train_data[p[0][0]],train_data[p[0][1]]).show() we
can able to plot scatter plot.
```

ii) Plot scatter points with smooth lines:

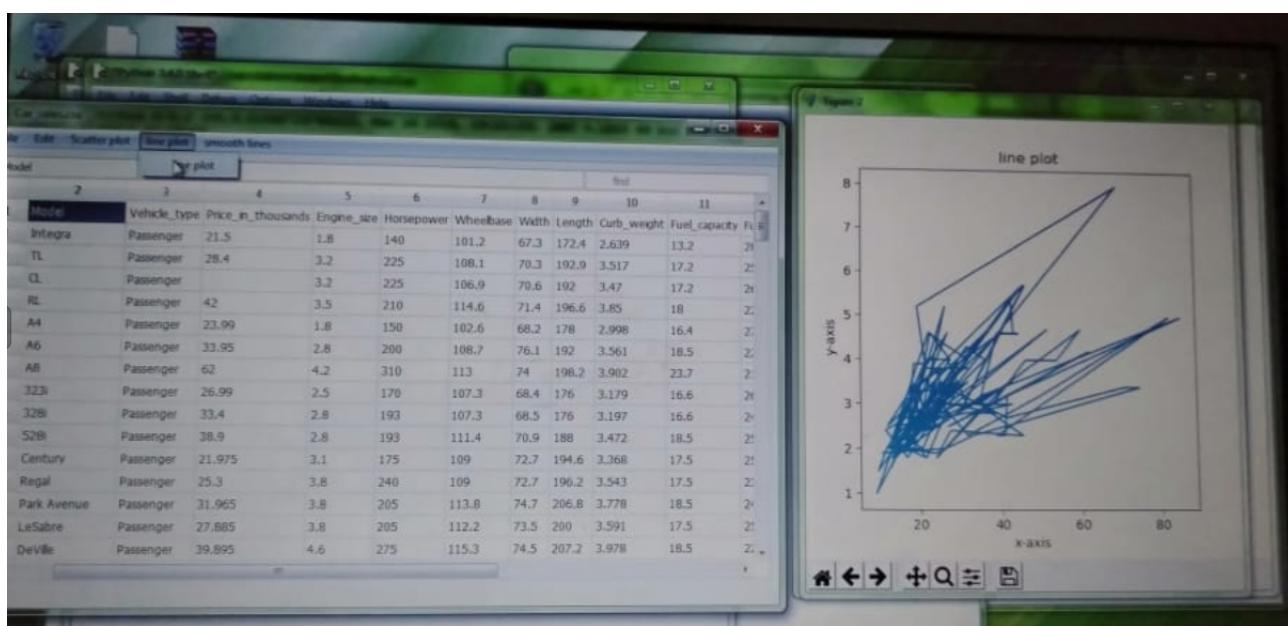


Description: After selecting more than 2 columns one has to select the smooth lines button than we can able to plot the smooth lines.

Code:

- i) By using `self.editmenu3=bar.addAction(QIcon.fromTheme("Scatter"), "smoothline", self.plot2)` we can create button.
- ii) By using `self.editmenu3.addAction(QIcon.fromTheme("Scatter"), "smoothline", self.plot2)` we perform an action.
- iii) By using `train_data = pd.read_csv(z)`
`plt.plot(train_data[p[0][0]],train_data[p[0][1]])` we can plot smooth lines.

iii) Plot lines:



Description: After selecting more than 2 columns one has to select the smooth lines button than we can able to plot the line plot.

Code:

- i) By using `self.editmenu2=bar.addAction(QIcon.fromTheme("Scatter"), "smoothline", self.plot2)` we can create button.
- ii) By using `self.editmenu3.addAction(QIcon.fromTheme("Scatter"), "smoothline", self.plot2)` we perform an action.
- iii) By using `train_data = pd.read_csv(z)
pplt.plot(x,y,'bo-')` we can plot smooth lines.

7.Click on any of the plot button. Plot should be generated accordingly in a new tab:

Description: For the above plots the new tab is generating.

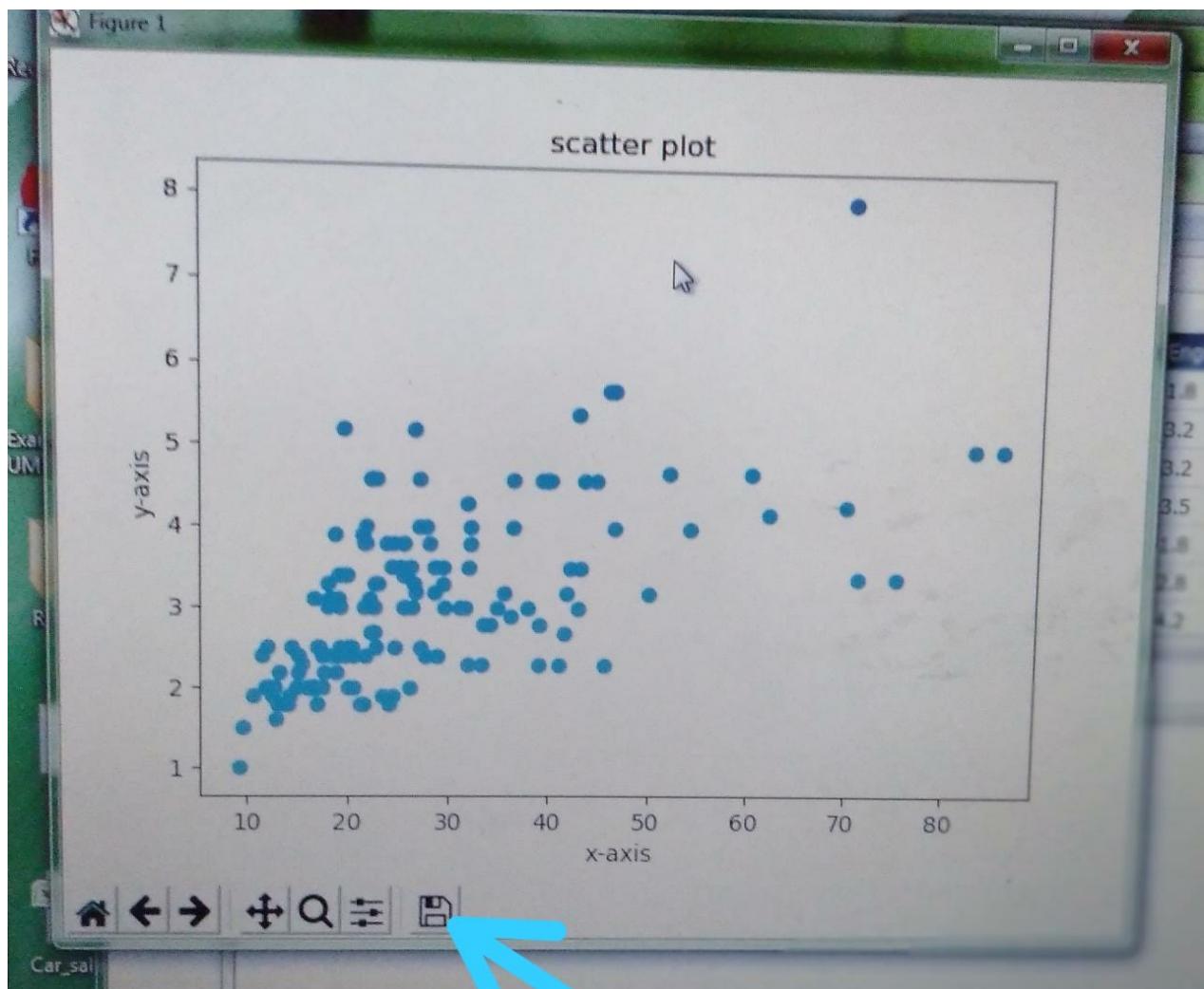
8.Label x-axis and y-axis accordingly.

9.Add a title to the graph.

Description: For the above plots x and y labels are used and title for the graph is given.

Code: By using `plt.title(' line plot')
plt.xlabel('x-axis')
plt.ylabel('y-axis')` we plotted the graph by using title and x,y labels.

10.Save the plot as .png file:



Description: By clicking on this save image button the image is saved.