PRACTICAL NO 1

AIM:- Introduction to Excel

a. Perform conditional formatting on a dataset using various criteria.
b. Create a pivot table to analyze and summarize data.
c. Use VLOOKUP function to retrieve information from a different worksheet or table.
d. Perform what-if analysis using Goal Seek to determine input values for desired output.

INTRODUCTION:-

Microsoft Excel is a versatile spreadsheet software essential for organizing, analyzing, and visualizing data. Its features make it a valuable tool for personal and professional tasks, particularly in business, finance, and data analysis.
This introduction covers some of the key features in Excel that help users handle complex data and extract valuable insights efficiently.

- **Conditional Formatting**: Highlight cells based on specific criteria (e.g., values above a threshold, duplicate entries, or trends). This makes it easier to identify key information or outliers at a glance.
- **Pivot Tables**: Summarize, analyze, and organize data dynamically. Drag-and-drop fields to reveal trends, comparisons, and patterns. Ideal for reporting and decision-making.
- **VLOOKUP Function**: A lookup tool to search a value in one column and retrieve related data from another column. Useful for cross-referencing and extracting information from large datasets.
- **Goal Seek**: Perform sensitivity analysis by calculating the input needed to achieve a target outcome. Frequently used in budgeting, forecasting, and decision-making.

These features empower users to handle complex data efficiently, transforming raw information into actionable insights. Mastering them streamlines workflows and supports better decision-making across personal and professional contexts.

OUTPUT:-

a. Perform conditional formatting on a dataset using various criteria.

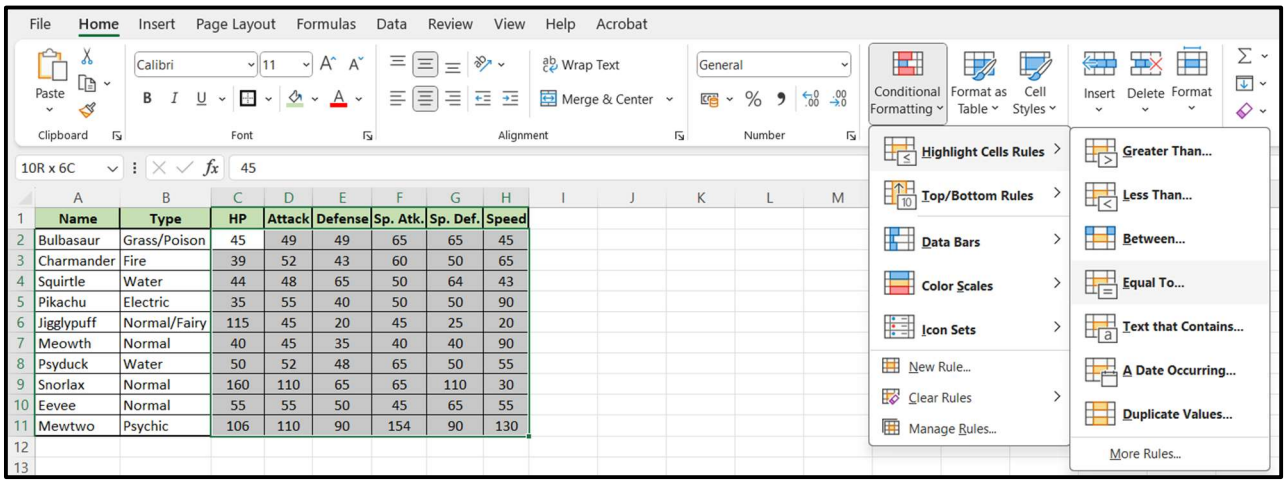
1) Equal to :-

 # With Numbers :-

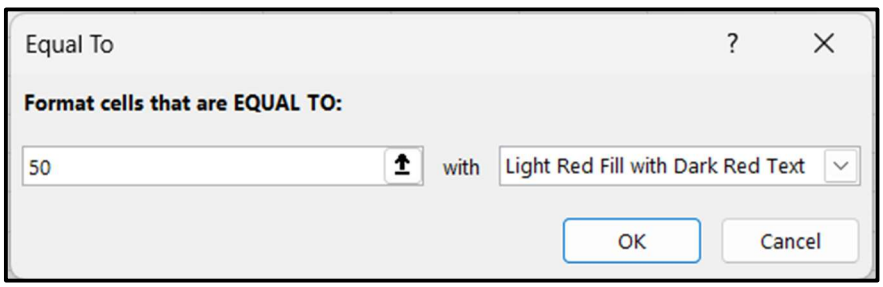
Step 1:- Open Excel and create a new excel sheet. Select the range **C2:H11** for all of the stat values.

	A	B	C	D	E	F	G	H	I
1	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 2:- Click on the Conditional Formatting icon , from Home menu. Select Highlight Cell Rules from the drop-down menu and select **Equal To..** from the menu.



Step 3:- Enter **"50"** into the input field. And select the appearance.



Step 4:- Now, the cells with values equal to **"50"** will be highlighted in red:

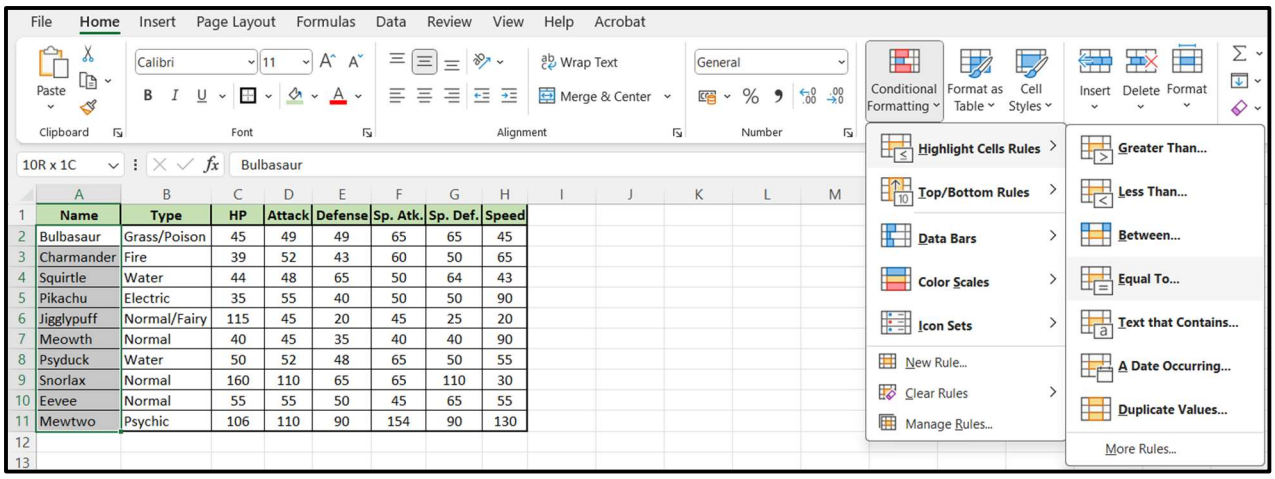
	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

 # With Text :-

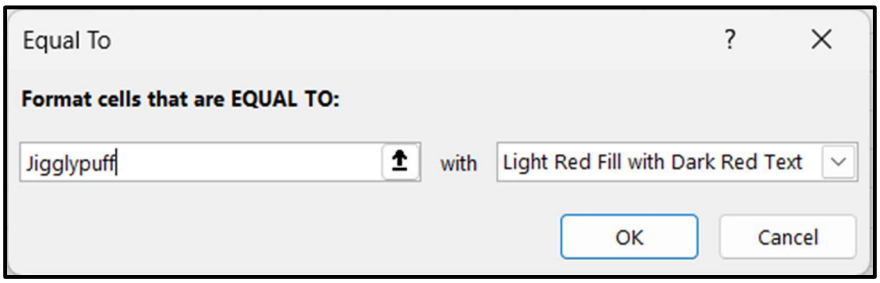
Step 1:- Open Excel and create a new excel sheet. Select the range **A2:A11** for all of the stat values.

	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 2:- Click on the Conditional Formatting icon , from Home menu. Select Highlight Cell Rules from the drop-down menu and select **Equal To..** from the menu.



Step 3:- Enter **"Jigglypuff"** into the input field. And select the appearance.



Step 4:- Now, the cells with values equal to **"Jigglypuff"** will be highlighted in red:

	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

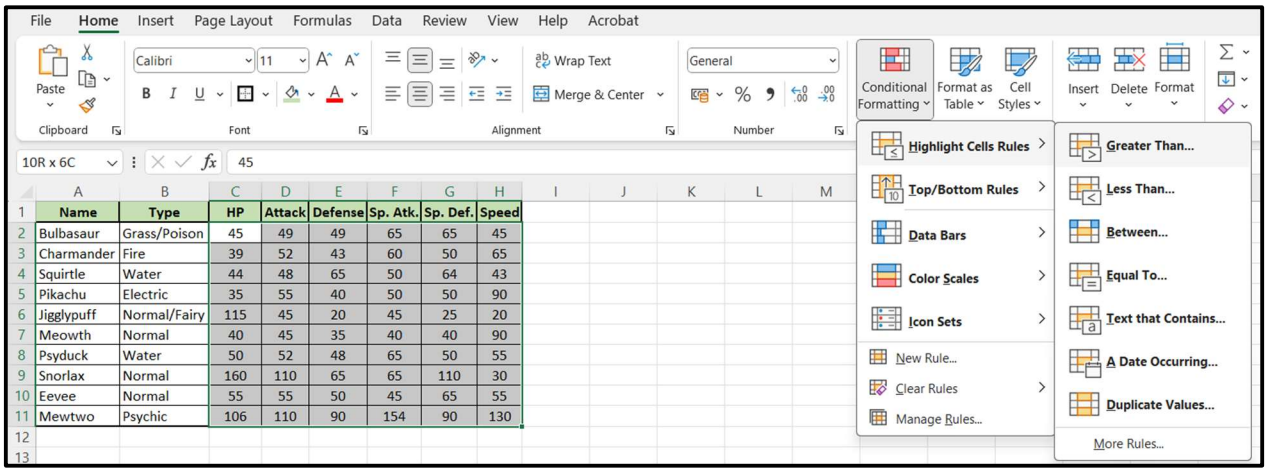
2) **Greater than :-**

 # With Numbers :-

Step 1:- Open Excel and create a new excel sheet. Select the range **C2:H11** for all of the stat values.

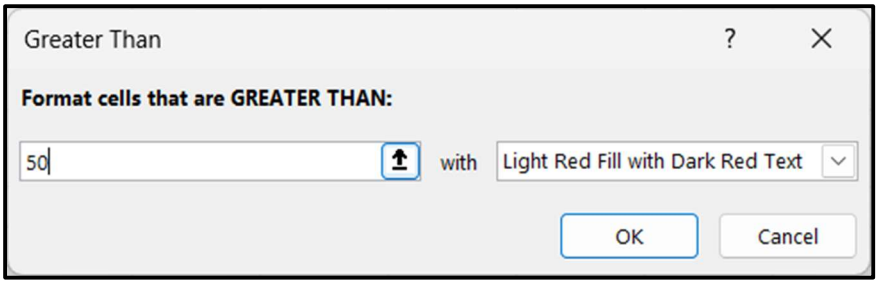
	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 2:- Click on the Conditional Formatting icon , from Home menu. Select Highlight Cell Rules from the drop-down menu and select **Greater Than..** from the menu.



	A	B	C	D	E	F	G	H
1	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45
3	Charmander	Fire	39	52	43	60	50	65
4	Squirtle	Water	44	48	65	50	64	43
5	Pikachu	Electric	35	55	40	50	50	90
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20
7	Meowth	Normal	40	45	35	40	40	90
8	Psyduck	Water	50	52	48	65	50	55
9	Snorlax	Normal	160	110	65	65	110	30
10	Eevee	Normal	55	55	50	45	65	55
11	Mewtwo	Psychic	106	110	90	154	90	130
12								
13								

Step 3:- Enter **"50"** into the input field. And select the appearance.



Step 4:- Now, the cells with values greater than **"50"** will be highlighted in red:

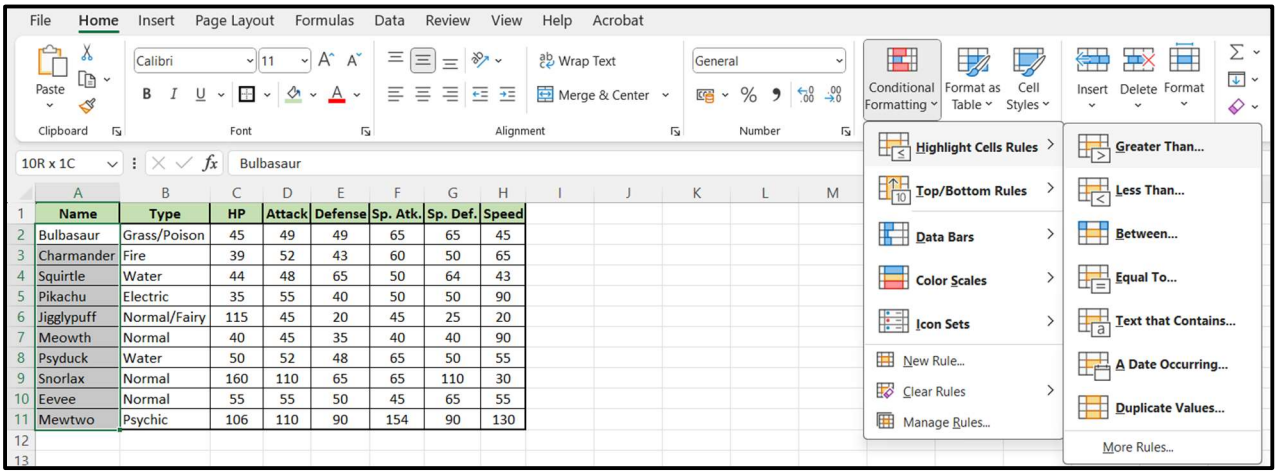
	A	B	C	D	E	F	G	H	I
1	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

 # With Text :-

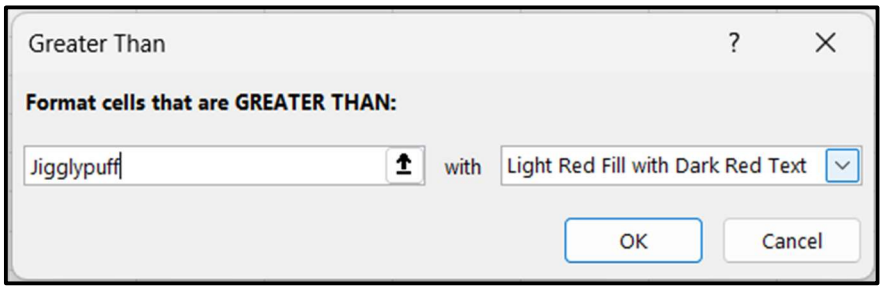
Step 1:- Open Excel and create a new excel sheet. Select the range **A2:A11** for all of the stat values.

	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 2:- Click on the Conditional Formatting icon , from Home menu. Select Highlight Cell Rules from the drop-down menu and select **Greater Than..** from the menu.



Step 3:- Enter **"Jigglypuff"** into the input field. And select the appearance.



Step 4:- Now, the cells with text values later in the alphabet than **"Jigglypuff"** will be highlighted in red:

	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

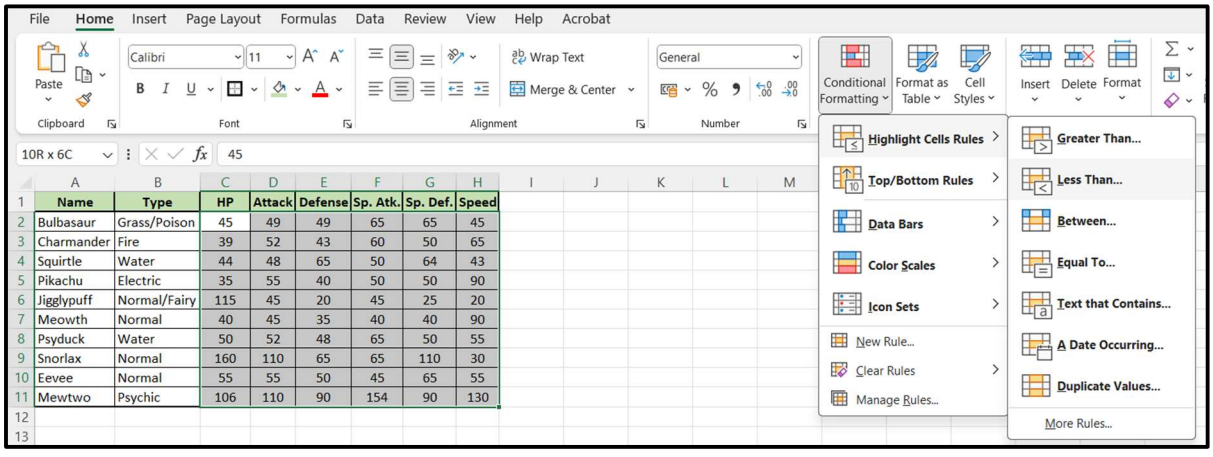
3) **Less than :-**

 # With Numbers :-

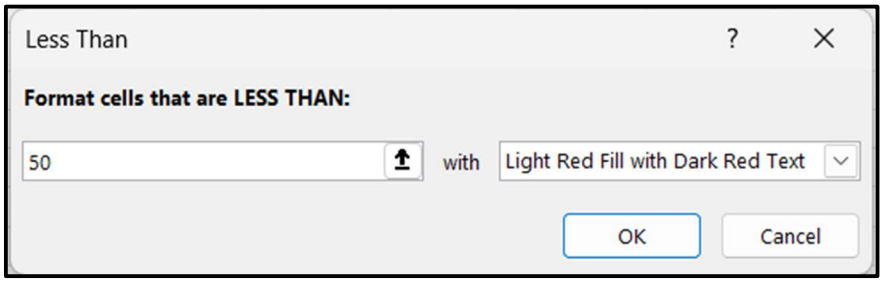
Step 1:- Open Excel and create a new excel sheet. Select the range **C2:H11** for all of the stat values.

	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 2:- Click on the Conditional Formatting icon , from Home menu. Select Highlight Cell Rules from the drop-down menu and select **Less Than..** from the menu.



Step 3:- Enter **"50"** into the input field. And select the appearance.



Step 4:- Now, the cells with values less than **"50"** will be highlighted in red:

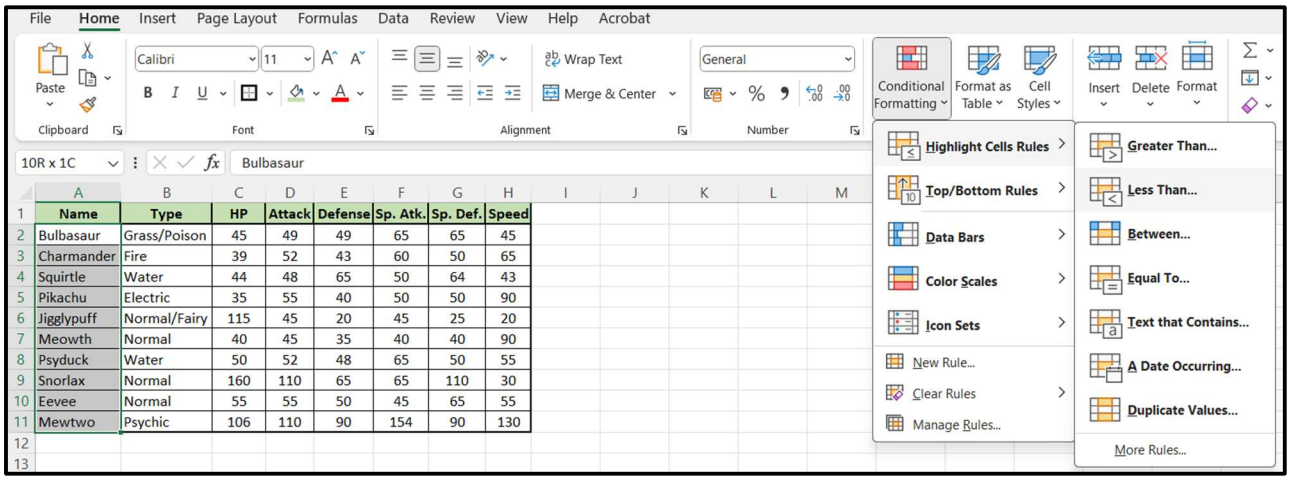
	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

 # With Text :-

Step 1:- Open Excel and create a new excel sheet. Select the range **A2:A11** for all of the stat values.

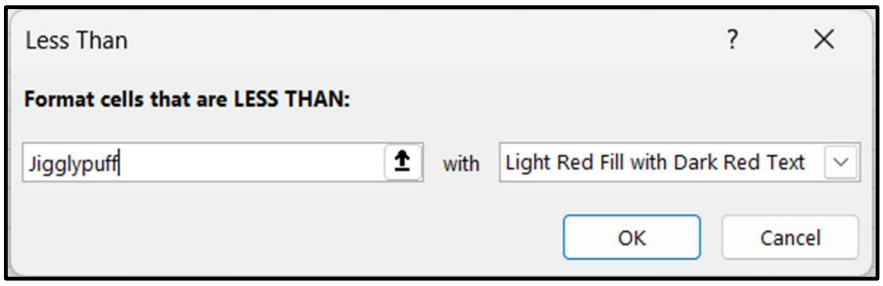
	A	B	C	D	E	F	G	H	I
1	**Name**	**Type**	**HP**	**Attack**	**Defense**	**Sp. Atk.**	**Sp. Def.**	**Speed**	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 2:- Click on the Conditional Formatting icon , from Home menu. Select Highlight Cell Rules from the drop-down menu and select **Less Than..** from the menu.



	A	B	C	D	E	F	G	H
1	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45
3	Charmander	Fire	39	52	43	60	50	65
4	Squirtle	Water	44	48	65	50	64	43
5	Pikachu	Electric	35	55	40	50	50	90
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20
7	Meowth	Normal	40	45	35	40	40	90
8	Psyduck	Water	50	52	48	65	50	55
9	Snorlax	Normal	160	110	65	65	110	30
10	Eevee	Normal	55	55	50	45	65	55
11	Mewtwo	Psychic	106	110	90	154	90	130
12								
13								

Step 3:- Enter **"Jigglypuff"** into the input field. And select the appearance.

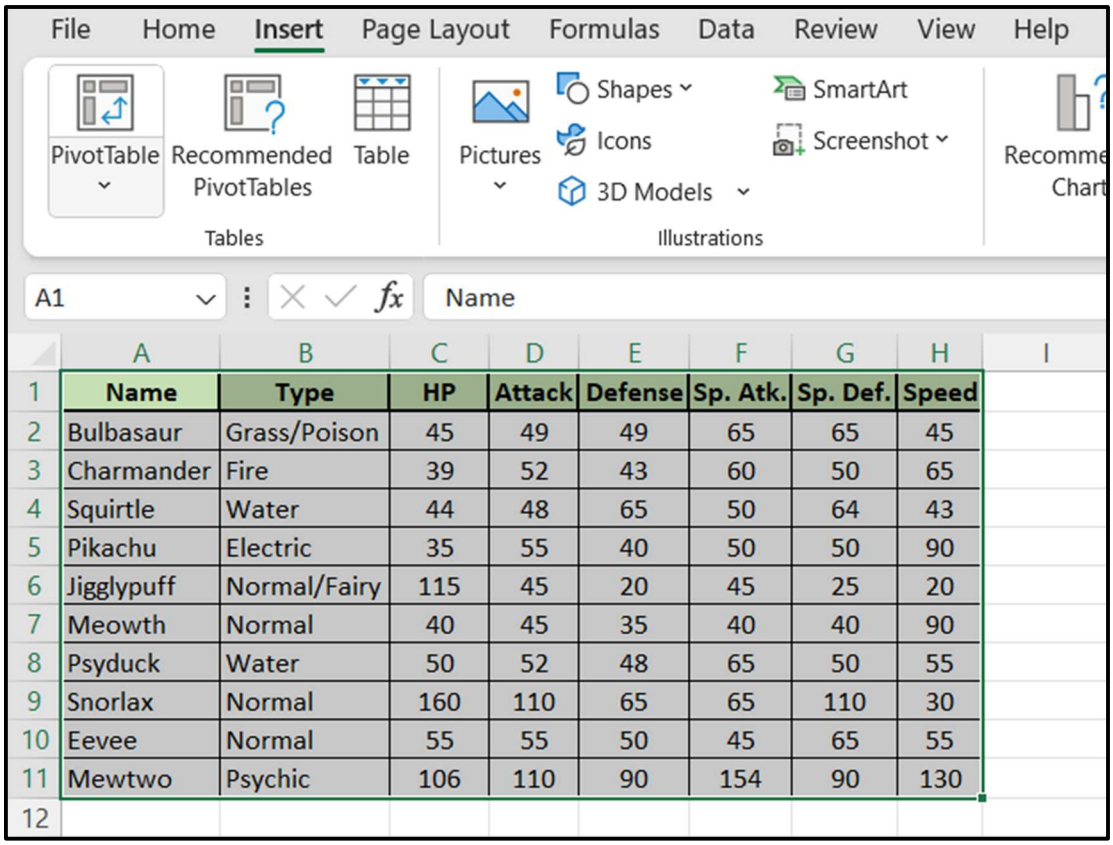


Step 4:- Now, the cells with text values earlier in the alphabet than **"Jigglypuff"** will be highlighted in red:

	A	B	C	D	E	F	G	H	I
1	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

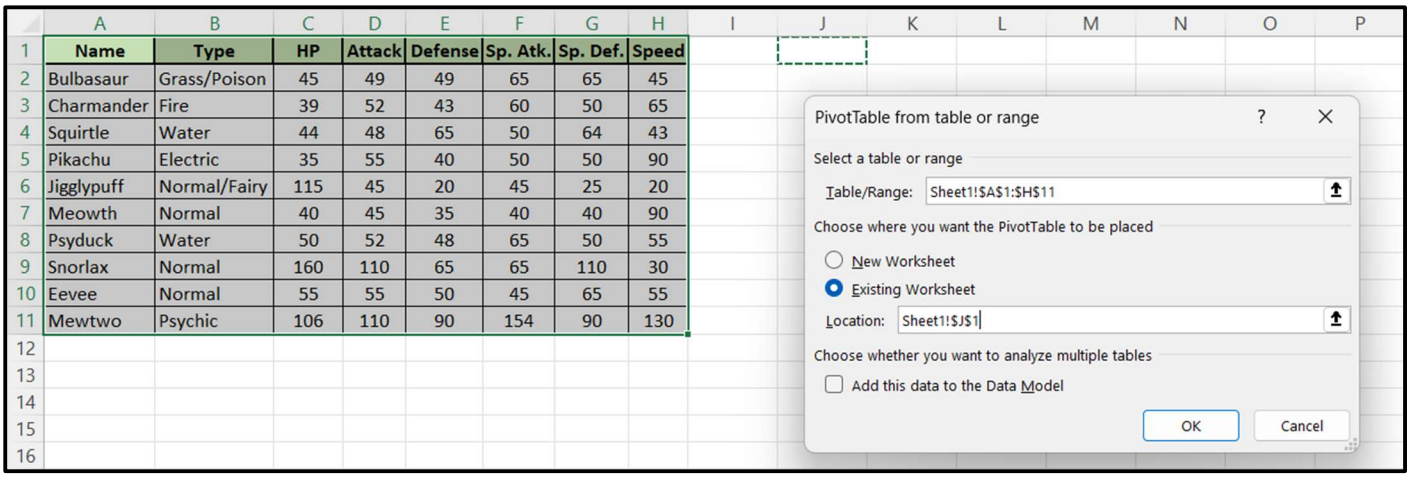
b. Create a pivot table to analyze and summarize data.

Step 1:- Open Excel and create a new excel sheet.
Step 2:- Select Table data A1:H11 and Click on the PivotTable , from Insert menu.

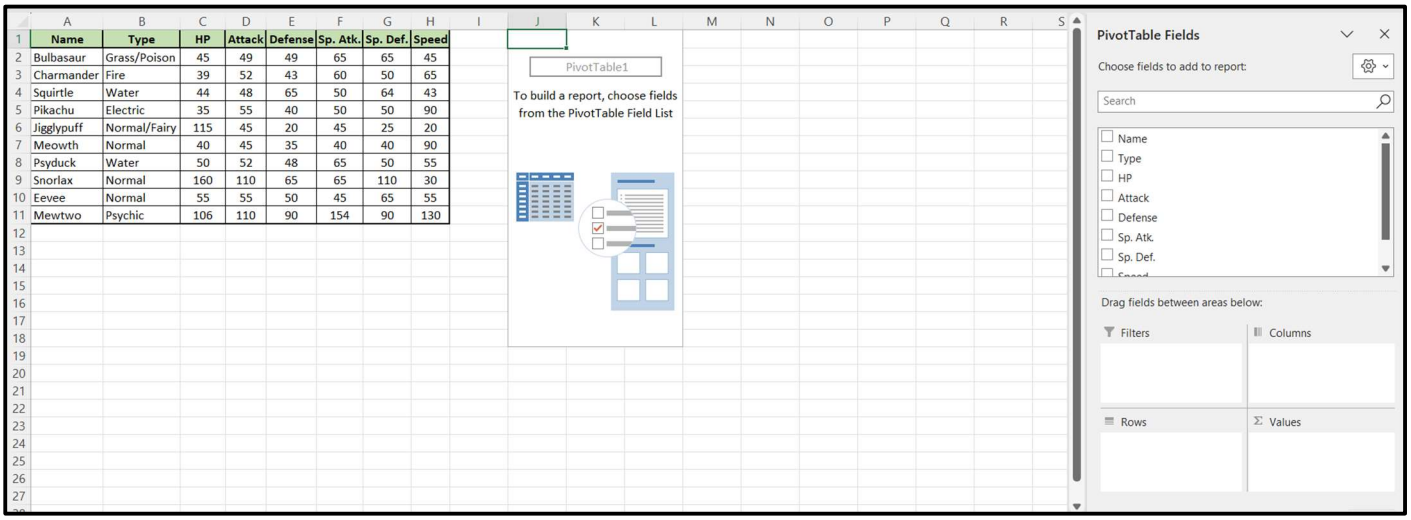


	A	B	C	D	E	F	G	H	I
1	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed	
2	Bulbasaur	Grass/Poison	45	49	49	65	65	45	
3	Charmander	Fire	39	52	43	60	50	65	
4	Squirtle	Water	44	48	65	50	64	43	
5	Pikachu	Electric	35	55	40	50	50	90	
6	Jigglypuff	Normal/Fairy	115	45	20	45	25	20	
7	Meowth	Normal	40	45	35	40	40	90	
8	Psyduck	Water	50	52	48	65	50	55	
9	Snorlax	Normal	160	110	65	65	110	30	
10	Eevee	Normal	55	55	50	45	65	55	
11	Mewtwo	Psychic	106	110	90	154	90	130	
12									

Step 3:- The following dialog box appears. The default location for a new pivot table is New Worksheet. Select Existing Worksheet then click on any cell from current sheet for location of Pivot Table.

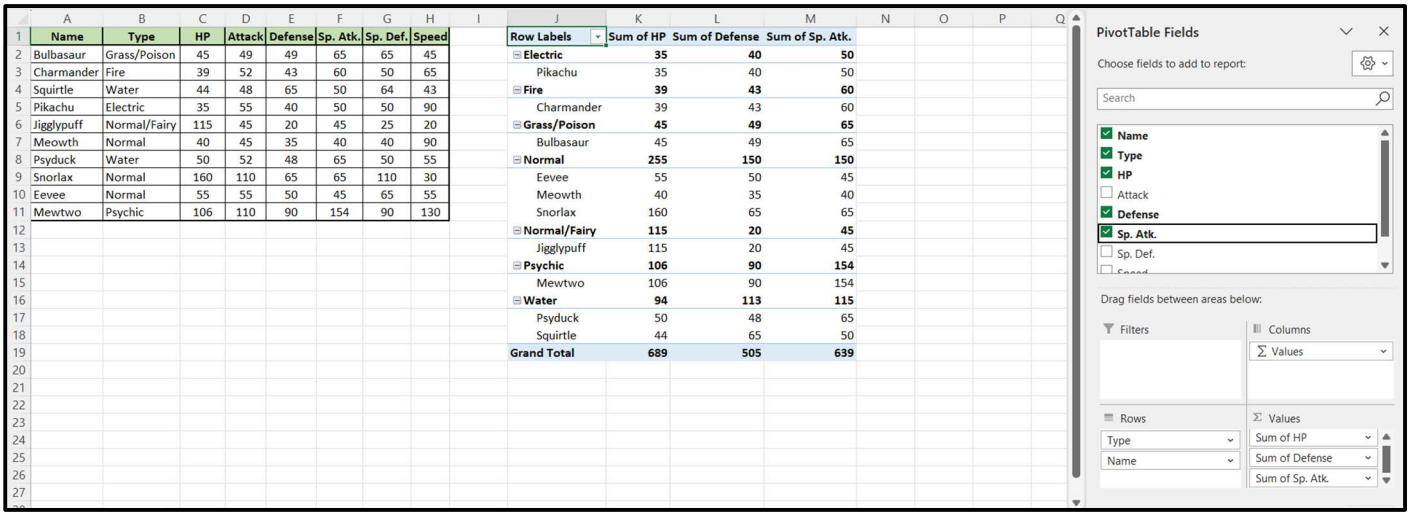


Step 4:- Click Ok. Then, it will create a pivot table.



Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed
Bulbasaur	Grass/Poison	45	49	49	65	65	45
Charmander	Fire	39	52	43	60	50	65
Squirtle	Water	44	48	65	50	64	43
Pikachu	Electric	35	55	40	50	50	90
Jigglypuff	Normal/Fairy	115	45	20	45	25	20
Meowth	Normal	40	45	35	40	40	90
Psyduck	Water	50	52	48	65	50	55
Snorlax	Normal	160	110	65	65	110	30
Eevee	Normal	55	55	50	45	65	55
Mewtwo	Psychic	106	110	90	154	90	130

Step 5:- Drag the following fields to the following areas.



Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed
Bulbasaur	Grass/Poison	45	49	49	65	65	45
Charmander	Fire	39	52	43	60	50	65
Squirtle	Water	44	48	65	50	64	43
Pikachu	Electric	35	55	40	50	50	90
Jigglypuff	Normal/Fairy	115	45	20	45	25	20
Meowth	Normal	40	45	35	40	40	90
Psyduck	Water	50	52	48	65	50	55
Snorlax	Normal	160	110	65	65	110	30
Eevee	Normal	55	55	50	45	65	55
Mewtwo	Psychic	106	110	90	154	90	130

Row Labels	Sum of HP	Sum of Defense	Sum of Sp. Atk.
Electric	**35**	**40**	**50**
Pikachu	35	40	50
Fire	**39**	**43**	**60**
Charmander	39	43	60
Grass/Poison	**45**	**49**	**65**
Bulbasaur	45	49	65
Normal	**255**	**150**	**150**
Eevee	55	50	45
Meowth	40	35	40
Snorlax	160	65	65
Normal/Fairy	**115**	**20**	**45**
Jigglypuff	115	20	45
Psychic	**106**	**90**	**154**
Mewtwo	106	90	154
Water	**94**	**113**	**115**
Psyduck	50	48	65
Squirtle	44	65	50
Grand Total	**689**	**505**	**639**

c. Use VLOOKUP function to retrieve information from a different worksheet or table.

Step 1:- Open Excel and create a new excel sheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed				
2	1	Bulbasaur	Grass/Poison	45	49	49	65	65	45				
3	2	Charmander	Fire	39	52	43	60	50	65		Search ID :-		
4	3	Squirtle	Water	44	48	65	50	64	43		Name :-		
5	4	Pikachu	Electric	35	55	40	50	50	90				
6	5	Jigglypuff	Normal/Fairy	115	45	20	45	25	20				
7	6	Meowth	Normal	40	45	35	40	40	90				
8	7	Psyduck	Water	50	52	48	65	50	55				
9	8	Snorlax	Normal	160	110	65	65	110	30				
10	9	Eevee	Normal	55	55	50	45	65	55				
11	10	Mewtwo	Psychic	106	110	90	154	90	130				
12													

Step 2:- Use VLOOKUP Function to find the name based on ID number.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ID	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed							
2	1	Bulbasaur	Grass/Poison	45	49	49	65	65	45							
3	2	Charmander	Fire	39	52	43	60	50	65		Search ID :-					
4	3	Squirtle	Water	44	48	65	50	64	43		=VLOOKUP(L3,A2:I11,2)					
5	4	Pikachu	Electric	35	55	40	50	50	90							
6	5	Jigglypuff	Normal/Fairy	115	45	20	45	25	20							
7	6	Meowth	Normal	40	45	35	40	40	90							
8	7	Psyduck	Water	50	52	48	65	50	55							
9	8	Snorlax	Normal	160	110	65	65	110	30							
10	9	Eevee	Normal	55	55	50	45	65	55							
11	10	Mewtwo	Psychic	106	110	90	154	90	130							
12																

VLOOKUP(**lookup_value**, table_array, col_index_num, [range_lookup])

Step 3:- After successfully using VLOOKUP function. The Function Return #N/A value.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed			
2	1	Bulbasaur	Grass/Poison	45	49	49	65	65	45			
3	2	Charmander	Fire	39	52	43	60	50	65		Search ID :-	
4	3	Squirtle	Water	44	48	65	50	64	43		Name :-	#N/A
5	4	Pikachu	Electric	35	55	40	50	50	90			
6	5	Jigglypuff	Normal/Fairy	115	45	20	45	25	20			
7	6	Meowth	Normal	40	45	35	40	40	90			
8	7	Psyduck	Water	50	52	48	65	50	55			
9	8	Snorlax	Normal	160	110	65	65	110	30			
10	9	Eevee	Normal	55	55	50	45	65	55			
11	10	Mewtwo	Psychic	106	110	90	154	90	130			

Step 4:- Let feed a value to it, type 5(cell:L3)

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	Name	Type	HP	Attack	Defense	Sp. Atk.	Sp. Def.	Speed			
2	1	Bulbasaur	Grass/Poison	45	49	49	65	65	45			
3	2	Charmander	Fire	39	52	43	60	50	65		Search ID :-	5
4	3	Squirtle	Water	44	48	65	50	64	43		Name :-	Jigglypuff
5	4	Pikachu	Electric	35	55	40	50	50	90			
6	5	Jigglypuff	Normal/Fairy	115	45	20	45	25	20			
7	6	Meowth	Normal	40	45	35	40	40	90			
8	7	Psyduck	Water	50	52	48	65	50	55			
9	8	Snorlax	Normal	160	110	65	65	110	30			
10	9	Eevee	Normal	55	55	50	45	65	55			
11	10	Mewtwo	Psychic	106	110	90	154	90	130			

d. Perform what-if analysis using Goal Seek to determine input values for desired output.

Step 1:- Set up the Excel Cells for Goal Seek as given below.

	A	B	C	D
1				
2		**Rate Per Annum**		Interest_rate
3		**No. of Monthly Payments**	360	NPERT
4		**Loan Amount**	5000000	Loan_Amount
5		**Type**	0	Type
6		**EMI**		EMI
7				

Step 2:- Use PMT function in the cell EMI(C6) to find the value.

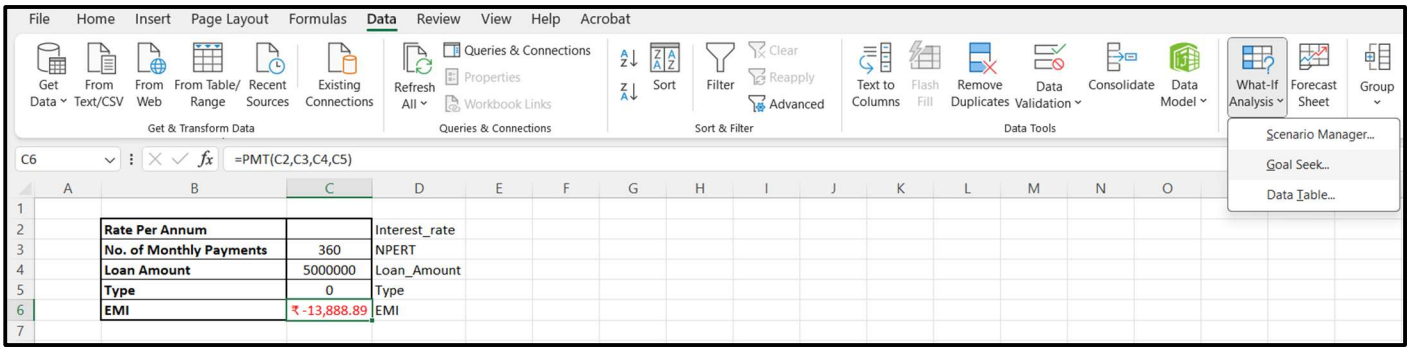
	A	B	C	D
1				
2		**Rate Per Annum**		Interest_rate
3		**No. of Monthly Payments**	360	NPERT
4		**Loan Amount**	5000000	Loan_Amount
5		**Type**	0	Type
6		**EMI**	=PMT(C2,C3,C4,C5)	

Step 3:- Value of EMI.

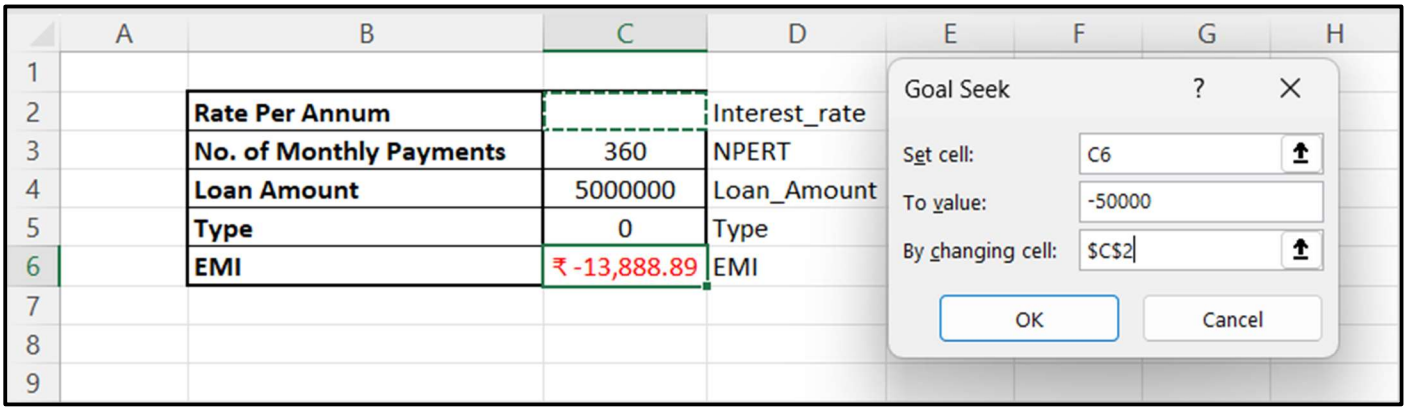
	A	B	C	D
1				
2		**Rate Per Annum**		Interest_rate
3		**No. of Monthly Payments**	360	NPERT
4		**Loan Amount**	5000000	Loan_Amount
5		**Type**	0	Type
6		**EMI**	₹ -13,888.89	EMI

Perform the Analysis with Goal seek.

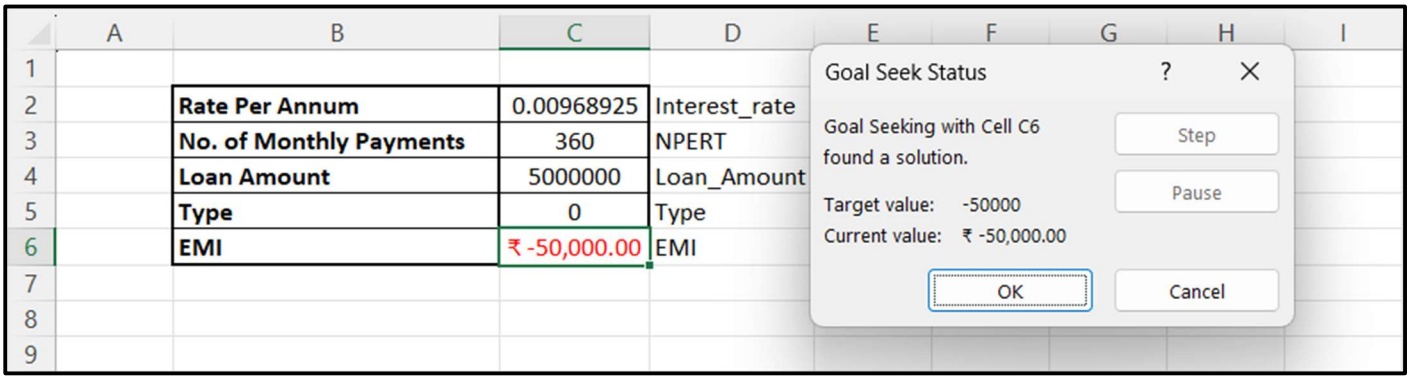
Step 4:- Click on the What if Analysis from the DATA Menu. Then select Goal Seek option.



Step 5:- Goal seek Dialog Box appear. Type cell EMI(C6) in Set cell box , Type -50000 to value box, Type cell Interest_Rate (C2) in the By changing cell box. And Click OK.



Step 6:- Goal Seek produces a result and Found the Solution Using C6 as shown below –



PRACTICAL NO 2

AIM:- Data Frames and Basic Data Pre-processing

a. Read data from CSV and JSON files into a data frame.
b. Perform basic data pre-processing tasks such as handling missing values and outliers.
c. Manipulate and transform data using functions like filtering, sorting, and grouping.

INTRODUCTION:-

Effective data handling is crucial in data analysis and machine learning. The Pandas library in Python provides powerful tools for managing and analyzing data using DataFrames, two-dimensional tables designed for efficient processing.

- **Reading Data**:
  - Use read_csv() to load data from CSV files.
  - Use read_json() to load data from JSON files.
  - These functions make it easy to import data for inspection and analysis.
- **Handling Missing Values**:
  - Fill missing values with specific values (e.g., column mean).
  - Remove rows or columns containing missing data to ensure accuracy.
- **Outlier Detection**:
  - Identify and handle outliers using techniques like the Interquartile Range (IQR) or Z-scores.
  - Address outliers by removing them or capping extreme values to a reasonable range.
- **Data Manipulation**:
  - Filter rows based on conditions.
  - Sort data in ascending or descending order.
  - Group data by categories to calculate aggregate statistics like sums or averages.

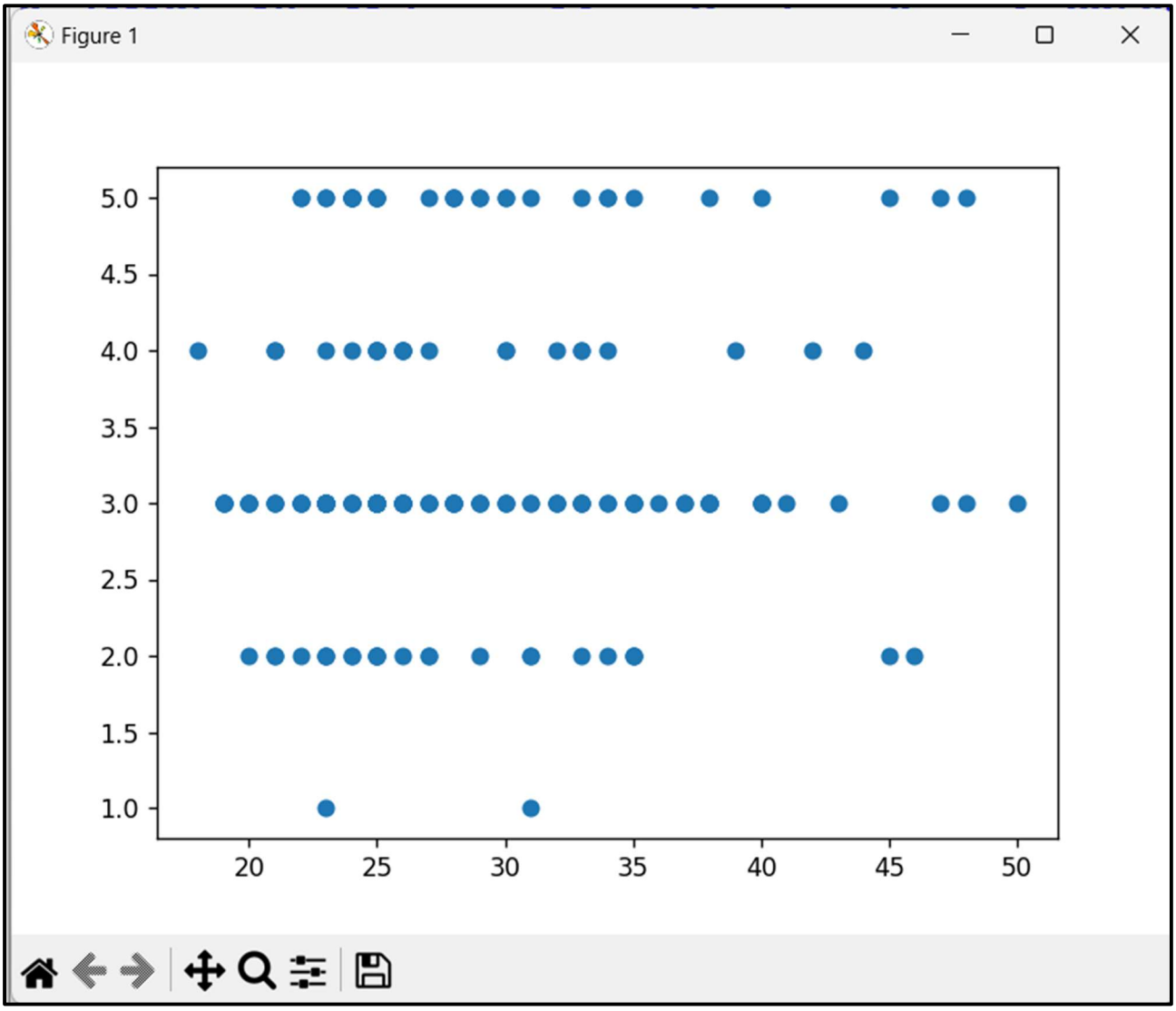
These tasks are essential for cleaning, transforming, and preparing real-world datasets for analysis and modeling. Mastering Pandas and its preprocessing techniques is key to efficient and accurate data analysis workflows.

CODE:-

```
import json
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

#creating a data frame
df=pd.read_csv("D:\Data Science\CardioGoodFitness.csv")
print(df.head())

#missing value and oulines
print(df.isnull().sum())

df1=df.dropna(axis=0)
print(df1.head())
print(df1.isnull().sum())

df['MaritalStatus']=df['MaritalStatus'].fillna('Single')
print(df['MaritalStatus'])
print(df.isnull().sum())
plt.scatter(x=df['Age'], y=df['Fitness'])
plt.show()

#sorting and grouping
sorted_df = df.sort_values(by='Age', ascending=False)
print(sorted_df)

mcount = df.groupby('MaritalStatus')['MaritalStatus'].count()
print(mcount)

#read data from json file
f=open("D:\Data Science\iris.json")
d=json.load(f)

df = pd.DataFrame(d)
print(df)
print(df.isnull().sum())

df1 = df.dropna(axis=0)
print(df1.head())

df['species']=df['species'].fillna('setosa')
print(df['species'])
print(df.isnull().sum())
```

OUTPUT:-

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

==========================================================================================
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0   TM195   18    Male         14        Single      3        4   29562    112
1   TM195   19    Male         15        Single      2        3   31836     75
2   TM195   19  Female         14     Partnered      4        3   30699     66
3   TM195   19    Male         12        Single      3        3   32973     85
4   TM195   20    Male         13     Partnered      4        2   35247     47
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0   TM195   18    Male         14        Single      3        4   29562    112
1   TM195   19    Male         15        Single      2        3   31836     75
2   TM195   19  Female         14     Partnered      4        3   30699     66
3   TM195   19    Male         12        Single      3        3   32973     85
4   TM195   20    Male         13     Partnered      4        2   35247     47
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
Squeezed text (181 lines).
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```



Squeezed text (181 lines).

```
MaritalStatus
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

Squeezed text (151 lines).

```
sepalLength    0
sepalWidth     0
petalLength    0
petalWidth     0
species        0
dtype: int64
   sepalLength  sepalWidth  petalLength  petalWidth species
0          5.1         3.5          1.4         0.2  setosa
1          4.9         3.0          1.4         0.2  setosa
2          4.7         3.2          1.3         0.2  setosa
3          4.6         3.1          1.5         0.2  setosa
4          5.0         3.6          1.4         0.2  setosa
```

Squeezed text (151 lines).

```
sepalLength    0
sepalWidth     0
petalLength    0
petalWidth     0
species        0
dtype: int64
```

PRACTICAL NO 3

AIM:- Hypothesis Testing

a. Formulate null and alternative hypotheses for a given problem.
b. Conduct a hypothesis test using appropriate statistical tests (e.g., t-test, chi-square test).
c. Interpret the results and draw conclusions based on the test outcomes.

INTRODUCTION:-

Hypothesis testing is a statistical method for making inferences about a population based on sample data. It is widely used in research, medicine, social sciences, and business decision-making.

- **Formulating Hypotheses**:
  - **Null Hypothesis (H_0)**: Assumes no effect, relationship, or difference (e.g., a drug has no impact).
  - **Alternative Hypothesis (H_1 or Ha)**: Proposes a significant effect, relationship, or difference (e.g., the drug improves health).
- **Selecting a Statistical Test**:
  - Choose the test based on the type of data and research question.
  - Common tests:
    - **t-test**: Compares means of two groups.
    - **Chi-square test**: Assesses associations between categorical variables.
- **Calculating Test Statistics**:
  - Compute a test statistic (e.g., t-value, chi-square statistic) to summarize the data.
  - Compare it to a critical value from the appropriate statistical distribution (e.g., t-distribution or chi-square distribution).
- **Interpreting Results**:
  - Evaluate the p-value, which measures evidence against H_0.
  - **If p-value < significance level (typically 0.05)**: Reject H_0; conclude the result is statistically significant.
  - **If p-value ≥ significance level**: Do not reject H_0; insufficient evidence to support H_1.

Hypothesis testing is essential for validating claims and making data-driven decisions. By evaluating the null and alternative hypotheses using statistical tests, researchers can determine whether observed patterns or effects are meaningful and significant.

CODE:-

```
# T-TEST
from scipy.stats import ttest_1samp
import numpy as np

ages = [45, 89, 23, 46, 12, 69, 45, 24, 34, 67]
print(ages)
mean = np.mean(ages)
print(mean)

t_test, p_val = ttest_1samp(ages, 30)
print("P-value is: ", p_val)

if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")

#CHI-SQUARE TEST
from scipy.stats import chi2_contingency

data = [[207, 282, 241], [234, 242, 232]]
stat, p, dof, expected = chi2_contingency(data)
alpha = 0.05
print("p value is " + str(p))

if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
```

OUTPUT:-

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11
Type "help", "copyright", "credits" or "lic

========== RESTART: D:/Data
[45, 89, 23, 46, 12, 69, 45, 24, 34, 67]
45.4
P-value is:  0.07179988272763561
We can accept the null hypothesis
p value is 0.10319714047309392
Independent (H0 holds true)
```

PRACTICAL NO 4

AIM:- ANOVA (Analysis of Variance)

a. Perform one-way ANOVA to compare means across multiple groups.
b. Conduct post-hoc tests to identify significant differences between group means.

INTRODUCTION:-

ANOVA is a statistical method used to compare the means of three or more groups to determine if there are statistically significant differences among them. It is widely applied in fields like psychology, medicine, and business to test hypotheses about the effects of different factors on a response variable.

- **Purpose of ANOVA**:
  - Compare means across multiple groups.
  - Test if at least one group mean is significantly different.
- **Types of ANOVA**:
  - **One-way ANOVA**: Tests one independent variable (factor) with multiple levels (groups).
  - Null hypothesis (H_0): All group means are equal.
  - Alternative hypothesis (H_1): At least one group mean differs.
- **Key Steps in One-Way ANOVA**:
  Divide data into groups based on the factor.
  Calculate the F-statistic: Compares variance between groups to variance within groups.
  Evaluate the p-value:
    - **p-value < 0.05**: Reject H_0; group means are significantly different.
    - **p-value ≥ 0.05**: Fail to reject H_0; no significant differences.
- **Post-hoc Tests**:
  - If ANOVA indicates significant differences, use post-hoc tests (e.g., Tukey's HSD) to determine which groups differ.
  - Post-hoc tests control for Type I errors in multiple comparisons.

ANOVA is a powerful tool for identifying and analyzing differences in group means. It provides insights into complex datasets, enabling researchers to make informed, data-driven decisions. Post-hoc tests further clarify specific group differences when needed.

CODE:-

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from scikit_posthocs import posthoc_tukey

np.random.seed(12)
races = ["asian","black","hispanic","other","white"]

# Generate random data
voter_race = np.random.choice(a= races,p = [0.05, 0.15 ,0.25, 0.05, 0.5],size=1000)
voter_age = stats.poisson.rvs(loc=18,mu=30,size=1000)

# Group age data by race
voter_frame = pd.DataFrame({"race":voter_race,"age":voter_age})
groups = voter_frame.groupby("race").groups

# Etract individual groups
asian = voter_age[groups["asian"]]
black = voter_age[groups["black"]]
hispanic = voter_age[groups["hispanic"]]
other = voter_age[groups["other"]]
white = voter_age[groups["white"]]

# Perform the ANOVA
print(stats.f_oneway(asian, black, hispanic, other, white))
print(posthoc_tukey(voter_frame,val_col="age",group_col="race"))
```

OUTPUT:-

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit
Type "help", "copyright", "credits" or "license()" for more information.

========== RESTART: D:/Data Science/datasciencepractical4.py ==========
F_onewayResult(statistic=1.7744689357329695, pvalue=0.13173183201930463)
              black     white  hispanic     asian     other
black      1.000000  0.999823  0.836230  0.584772  0.557072
white      0.999823  1.000000  0.497606  0.431914  0.389889
hispanic   0.836230  0.497606  1.000000  0.906173  0.900320
asian      0.584772  0.431914  0.906173  1.000000  1.000000
other      0.557072  0.389889  0.900320  1.000000  1.000000
```

PRACTICAL NO 5

AIM:- Regression and Its Types

a. Implement simple linear regression using a dataset.
b. Explore and interpret the regression model coefficients and goodness-of-fit measures.
c. Extend the analysis to multiple linear regression and assess the impact of additional predictors.

INTRODUCTION:-

Regression analysis is a statistical method for examining the relationship between a dependent variable (response) and one or more independent variables (predictors). It is widely used in economics, finance, biology, and social sciences for trend analysis, prediction, and establishing cause-and-effect relationships.

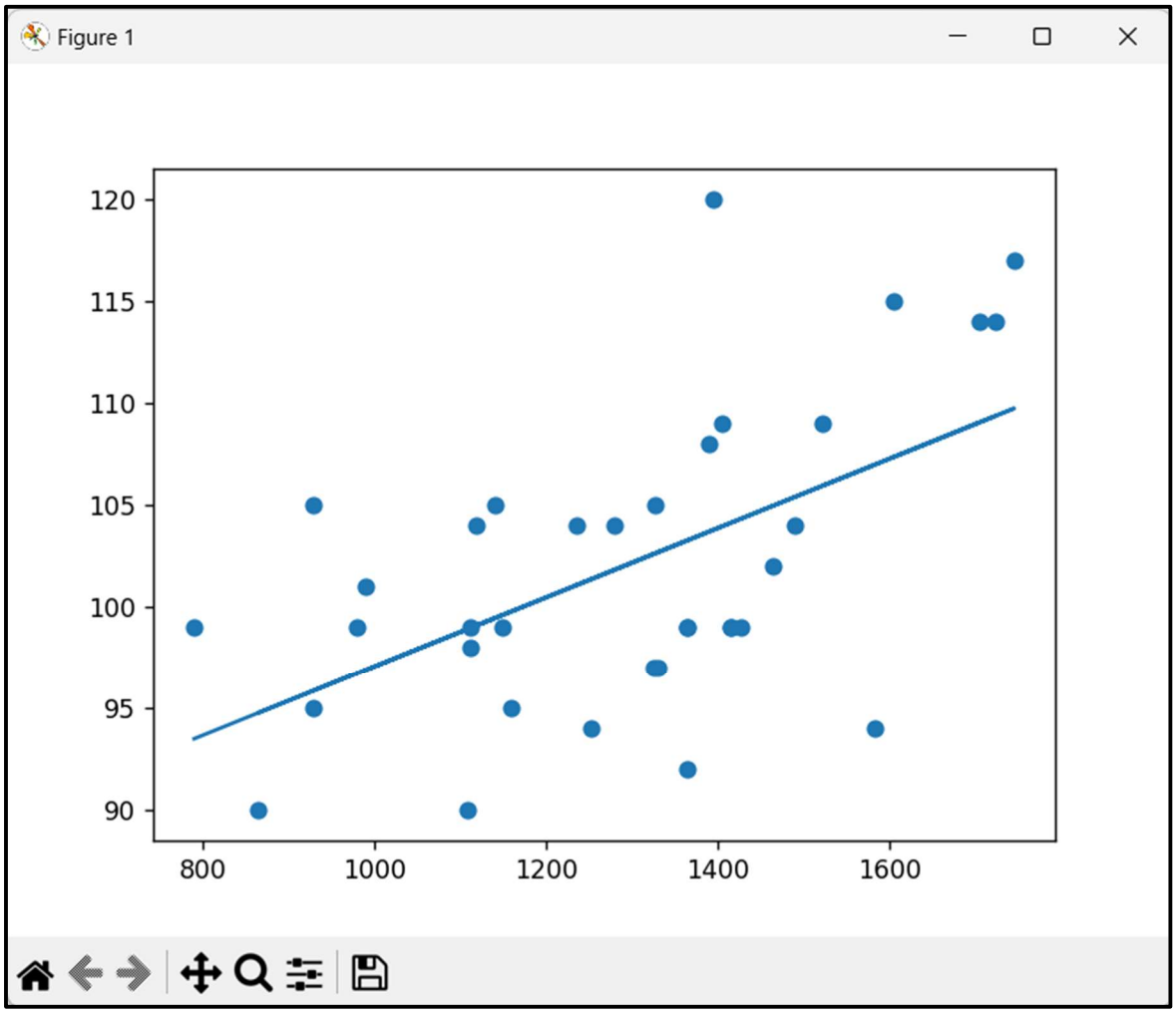
- **Types of Regression**:
  - **Simple Linear Regression**:
    - Models the relationship between one independent variable and the dependent variable.
    - Estimates coefficients (β_0 and β_1) to describe the relationship.
    - Enables prediction of the dependent variable (Y) based on the independent variable (X).
  - **Multiple Linear Regression**:
    - Extends the analysis to include multiple independent variables.
    - Provides a comprehensive view of factors affecting the outcome.
    - Assesses the marginal effect of each predictor while controlling for others.
- **Goodness-of-Fit**:
  - **R-squared**: Measures the proportion of variance in the dependent variable explained by the model.
  - **Higher R-squared**: Indicates a better model fit.
  - **Lower R-squared**: Suggests the model does not explain the data well.

Regression analysis is a crucial tool for understanding relationships, making predictions, and guiding decision-making. Simple linear regression is ideal for analyzing single predictors, while multiple regression captures complex interactions among multiple factors. By interpreting coefficients, assessing goodness-of-fit, and validating the model, regression offers valuable insights for data-driven decisions.

CODE:-

```
#Simple Linear Regression:
import pandas
import matplotlib.pyplot as plt
from scipy import stats

df = pandas.read_csv("D:\Data Science\DATA.csv")
x = df['Weight']
y = df['CO2']
slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):
   return slope * x + intercept

mymodel = list(map(myfunc, x))
plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()

#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:
predictedCO2 = myfunc(2300)
print(predictedCO2)

#Multiple Regression:
import pandas
from sklearn import linear_model

df = pandas.read_csv("D:\Data Science\DATA.csv")
X = df[['Weight', 'Volume']]
y = df['CO2']
regr = linear_model.LinearRegression()
regr.fit(X, y)

#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:
prediction_data = pandas.DataFrame([[2300, 1300]], columns=['Weight', 'Volume'])
predictedCO2 = regr.predict(prediction_data)
print(predictedCO2)
```

OUTPUT:-




```
================ RESTART: D:
119.15878810734196
[107.2087328]
```

PRACTICAL NO 6

AIM:- Logistic Regression and Decision Tree

a) Build a logistic regression model to predict a binary outcome.
b) Evaluate the model's performance using classification metrics (e.g., accuracy, precision, recall).
c) Construct a decision tree model and interpret the decision rules for classification.

INTRODUCTION :-

In the domain of machine learning, classification is a fundamental task aimed at predicting categorical outcomes based on input features. Logistic Regression and Decision Tree models are two widely used algorithms for binary classification problems, where the target variable can have only two possible outcomes. These models are essential due to their simplicity, interpretability, and effectiveness in a variety of applications such as fraud detection, medical diagnosis, and marketing.

Logistic Regression is a statistical method that models the probability of a binary outcome using a sigmoid function. It assumes a linear relationship between the independent variables and the log-odds of the dependent variable. Logistic Regression is computationally efficient, robust to noise, and provides probabilistic predictions, making it a preferred choice for many real-world scenarios. The model is evaluated using classification metrics such as accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic (ROC-AUC) curve to ensure its reliability.

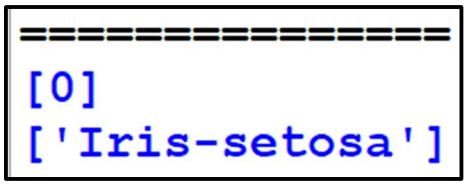
On the other hand, **Decision Trees** are non-parametric models that use a tree-like structure to represent decision-making processes. The model splits the data recursively based on feature values to create branches, with each leaf node corresponding to a specific class. Decision Trees are highly interpretable, as the decision rules can be easily visualized, making them useful for understanding how the classification decisions are made. They handle non-linear relationships and interactions between features effectively, but are prone to overfitting, especially when the tree is overly complex.

In this project, we aim to build and compare both Logistic Regression and Decision Tree models for a binary classification task. First, we will preprocess the data to ensure quality and consistency. Then, we will train the Logistic Regression model, evaluate its performance using classification metrics, and analyze its strengths and limitations. Next, we will construct a Decision Tree model, interpret its decision-making rules, and evaluate its performance. By comparing these two approaches, we can derive insights into their respective advantages and practical use cases. This study highlights the importance of model selection and evaluation in solving binary classification problems effectively.

CODE :-

```
import numpy
from sklearn import linear_model
import pandas as pd

X = numpy.array([3.78, 2.44, 2.09, 0.14, 1.72, 1.65, 4.92, 4.37, 4.96, 4.52, 3.69, 5.88]).reshape(-1,1)
y = numpy.array([0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
logr = linear_model.LogisticRegression()
logr.fit(X,y)

predicted = logr.predict(numpy.array([3.46]).reshape(-1,1))
print(predicted)

df = pd.read_csv("D:/Data Science/Iris.csv")
X1=numpy.array(df['SepalLengthCm']).reshape(-1,1)
y=df['Species']
logr = linear_model.LogisticRegression()
logr.fit(X1,y)

predicted = logr.predict(numpy.array([3.4]).reshape(-1,1))
print(predicted)
```

OUTPUT :-



PRACTICAL NO 7

AIM:- K-Means Clustering

a. Apply the K-Means algorithm to group similar data points into clusters.
b. Determine the optimal number of clusters using elbow method or silhouette analysis.
c. Visualize the clustering results and analyze the cluster characteristics.

INTRODUCTION:-

K-Means clustering is an unsupervised machine learning algorithm used to group similar data points into clusters. It identifies patterns in unlabeled data by minimizing variance within clusters and maximizing variance between them.

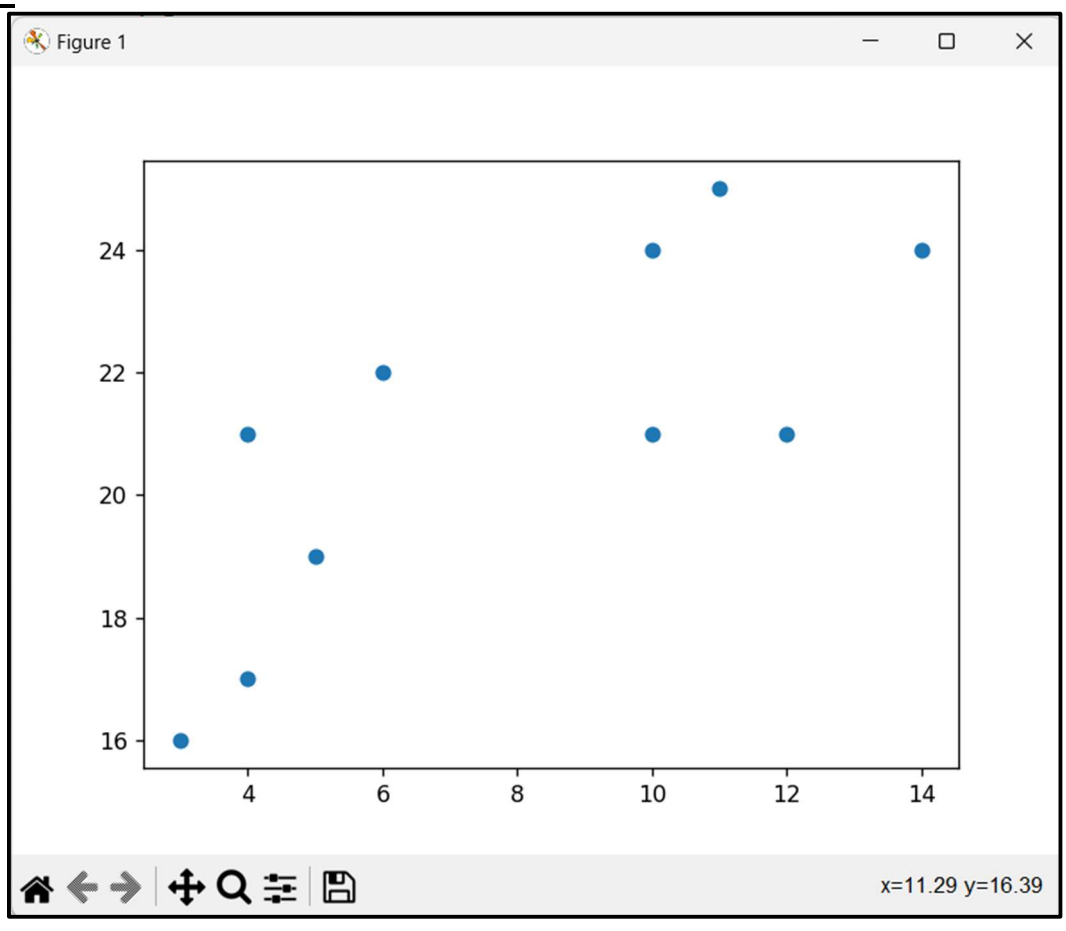
- **How K-Means Works**:
  - Assigns data points to the nearest cluster center (centroid).
  - Updates centroids based on the mean of assigned points.
  - Repeats until centroids stabilize (convergence).
  - Requires the number of clusters (K) to be predefined.
- **Determining Optimal K**:
  - **Elbow Method**:
    - Run K-Means with varying K values.
    - Plot sum of squared distances (inertia) against K.
    - Optimal K is at the "elbow" where the rate of inertia decrease slows.
  - **Silhouette Analysis**:
    - Measures similarity of data points within their cluster versus others.
    - Higher silhouette score indicates better clustering.
- **Analyzing and Visualizing Results**:
  - **Visualizations**:
    - Scatter plots and pair plots reveal cluster groupings and patterns.
  - **Cluster Interpretation**:
    - Examine centroids to identify shared features within clusters.
    - Useful for applications like customer segmentation or document grouping.

K-Means clustering is a powerful tool for discovering patterns in unlabeled data. By selecting the optimal K, visualizing results, and analyzing cluster characteristics, the algorithm provides actionable insights for decision-making and understanding data structure.

CODE:-

```
import matplotlib.pyplot as plt
x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]
plt.scatter(x, y)
plt.show()

from sklearn.cluster import KMeans
data = list(zip(x, y))
inertias = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()

kmeans = KMeans(n_clusters=2)
kmeans.fit(data)
plt.scatter(x, y, c=kmeans.labels_)
plt.show()
```

OUTPUT:-



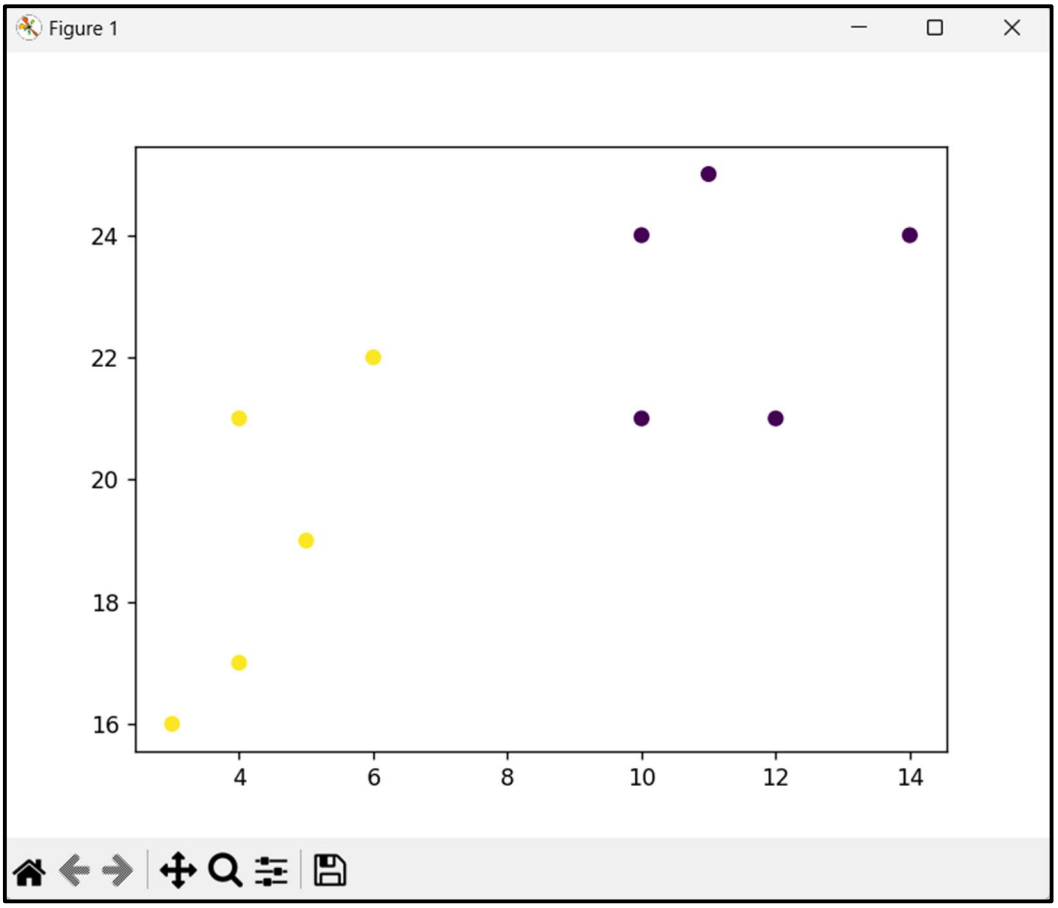


Figure 1
24
22
20
18
16
4
6
8
10
12
14

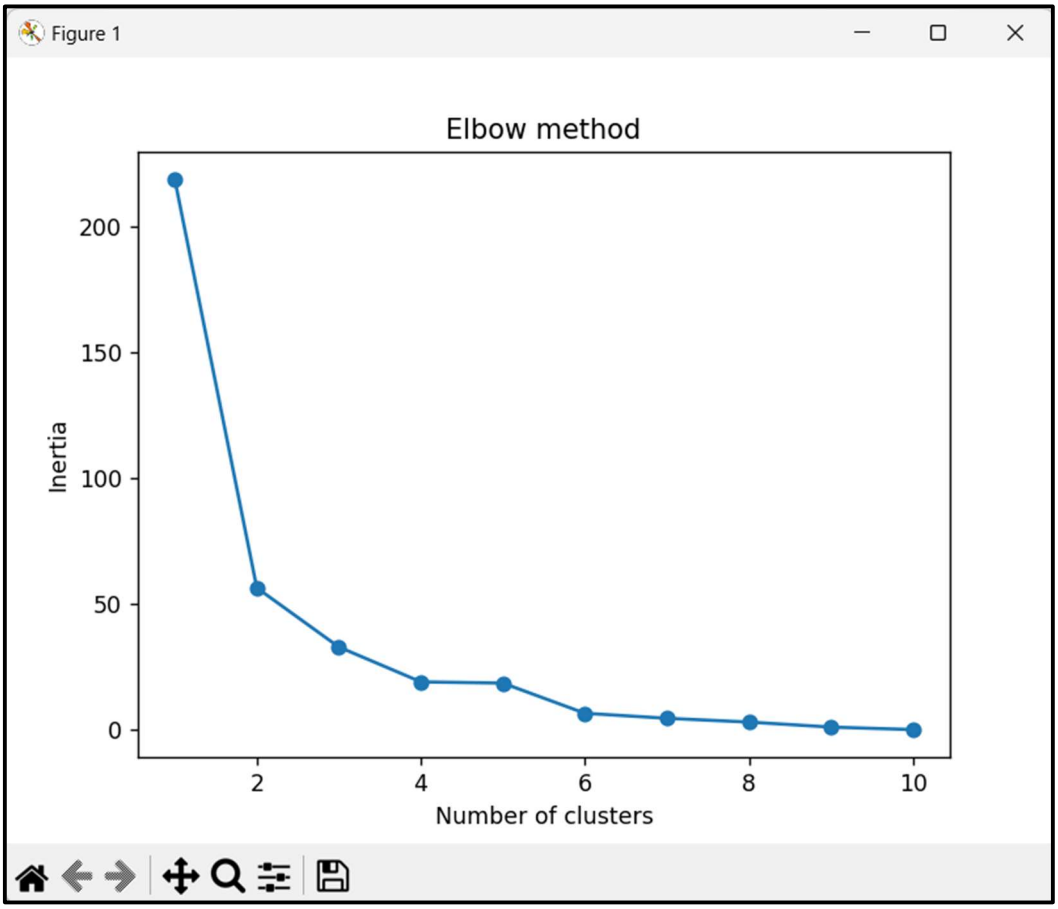


Figure 1
Elbow method
Inertia
200
150
100
50
0
2
4
6
8
10
Number of clusters

PRACTICAL NO 8

AIM:- Principal Component Analysis (PCA)

a) Perform PCA on a dataset to reduce dimensionality.
b) Evaluate the explained variance and select the appropriate number of principal components.
c) Visualize the data in the reduced-dimensional space.

INTRODUCTION :-

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique in data science and machine learning. It is particularly useful for high-dimensional datasets where multicollinearity among features and the curse of dimensionality can hinder the performance of machine learning algorithms. PCA transforms the original features into a set of linearly uncorrelated variables, known as principal components, which capture the maximum variance in the data.

The primary goal of PCA is to reduce the number of dimensions while preserving as much of the original data's variability as possible. This is achieved by finding the directions (principal components) along which the data varies the most and projecting the data onto these directions. Each principal component is a linear combination of the original features, with the first component capturing the largest variance, followed by the second component, and so on.

An essential step in PCA is evaluating the explained variance ratio, which indicates how much of the total variance is captured by each principal component. By analyzing the cumulative explained variance, we can determine the optimal number of components to retain while balancing dimensionality reduction and information preservation. This decision is critical in ensuring the reduced dataset retains sufficient information for subsequent analysis or modeling.

Once PCA is performed, the data is visualized in the reduced-dimensional space, often in two or three dimensions. These visualizations help uncover patterns, clusters, or separations that may not be apparent in the original high-dimensional space. PCA is also commonly used as a preprocessing step for machine learning algorithms to improve efficiency and mitigate overfitting.

In this project, we aim to apply PCA to a dataset to achieve dimensionality reduction. We will analyze the explained variance ratio to select the appropriate number of principal components and visualize the reduced-dimensional data. This study highlights the power of PCA in simplifying complex datasets while preserving meaningful information, making it an essential tool in exploratory data analysis and feature engineering.

CODE :-

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

dataset = pd.read_csv("D:/Data Science/wine.csv")
X = dataset.iloc[:, 0:13].values
y = dataset.iloc[:, 13].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1,
        stop = X_set[:, 0].max() + 1, step = 0.01),
        np.arange(start = X_set[:, 1].min() - 1,
        stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
        X2.ravel()]).T).reshape(X1.shape), alpha = 0.75,
        cmap = ListedColormap(('yellow', 'white', 'aquamarine')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
    c = ListedColormap(('red', 'green', 'blue'))(i), label = j)

plt.title('Logistic Regression (Training set)')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()
plt.show()
```

OUTPUT :-

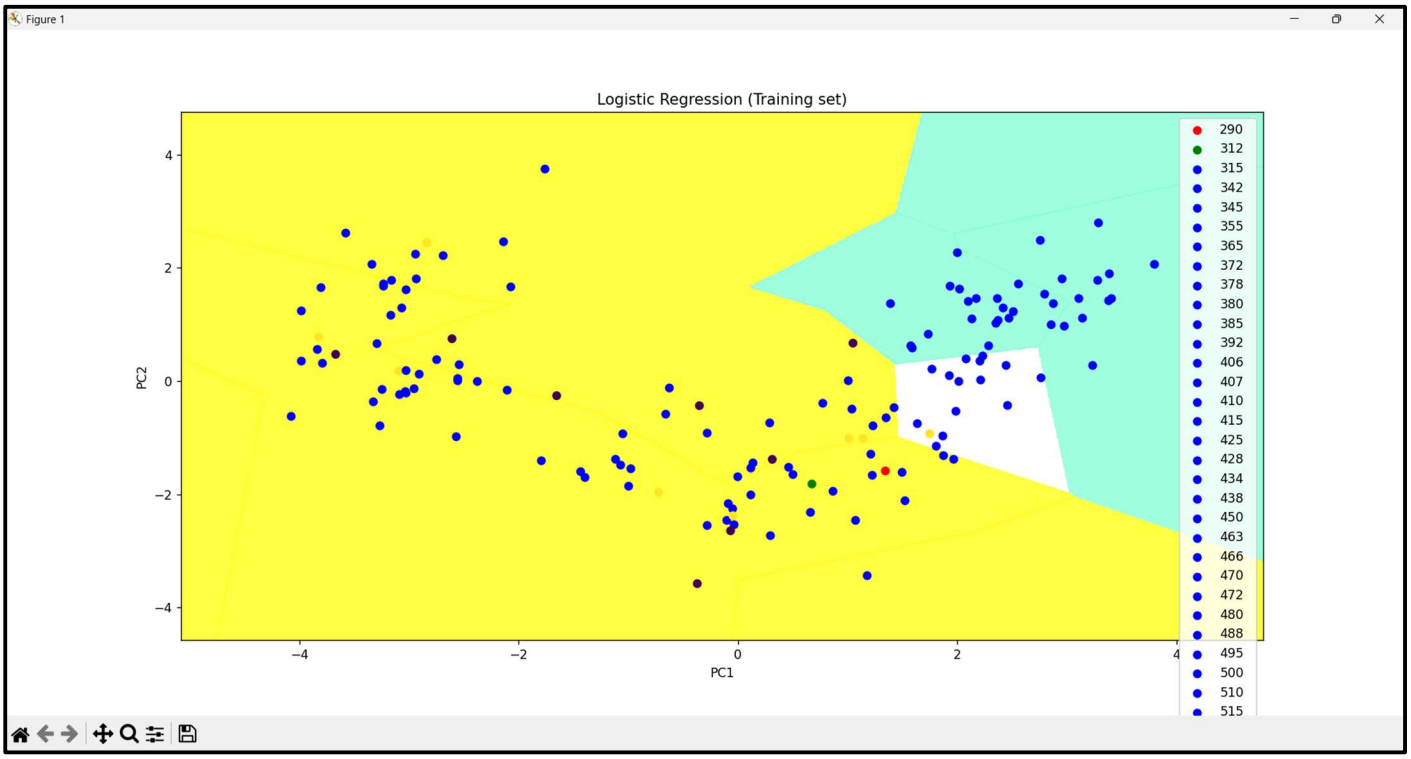


Figure 1
Logistic Regression (Training set)
PC2
PC1
4
2
0
−2
−4
−4
−2
0
2
4
290
312
315
342
345
355
365
372
378
380
385
392
406
407
410
415
425
428
434
438
450
463
466
470
472
480
488
495
500
510
515