NAME:VYSHNAVI VALLALA

HT.NO:2303A52429

BATCH:32

Question 1

Read the data with pandas and find features and target variables

Normalize the data with min-max scaling

Split the data into train and test.

```
import pandas as pd
df=pd.read_csv('/content/train.csv')
df
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222	1890	668	13
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915	1965	2032	11
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868	1632	3057	9
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336	670	869	18
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483	754	3919	19

2000 rows × 21 columns

```
y = df['price_range']
y
```

	price_range
0	1
1	2
2	2
3	2
4	1
...	...
1995	0
1996	2
1997	3
1998	0
1999	3

2000 rows × 1 columns

```
x=df.drop('price_range',axis=1)
x
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h
0	842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9
1	1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17
2	563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11
3	615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16
4	1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	794	1	0.5	1	0	1	2	0.8	106	6	14	1222	1890	668	13
1996	1965	1	2.6	1	0	0	39	0.2	187	4	3	915	1965	2032	11
1997	1911	0	0.9	1	1	1	36	0.7	108	8	3	868	1632	3057	9
1998	1512	0	0.9	0	4	1	46	0.1	145	5	5	336	670	869	18
1999	510	1	2.0	1	5	1	45	0.9	168	6	16	483	754	3919	19

2000 rows × 20 columns

Next steps: Generate code with x | View recommended plots | New interactive sheet

```
upd=x-x.min()/x.max()-x.min()
upd
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	
0	340.749249	0.0	1.533333	0.0	1.0	0.0	4.96875	0.4	107.6	0.875	2.0	20.0	255.74975	2292.93
1	519.749249	1.0	-0.166667	1.0	0.0	1.0	50.96875	0.5	55.6	1.875	6.0	905.0	1487.74975	2374.93
2	61.749249	1.0	-0.166667	1.0	2.0	1.0	38.96875	0.7	64.6	3.875	6.0	1263.0	1215.74975	2346.93
3	113.749249	1.0	1.833333	0.0	0.0	0.0	7.96875	0.6	50.6	4.875	9.0	1216.0	1285.74975	2512.93
4	1319.749249	1.0	0.533333	0.0	13.0	1.0	41.96875	0.4	60.6	0.875	14.0	1208.0	711.74975	1154.93
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1995	292.749249	1.0	-0.166667	1.0	0.0	1.0	-0.03125	0.6	25.6	4.875	14.0	1222.0	1389.74975	411.93
1996	1463.749249	1.0	1.933333	1.0	0.0	0.0	36.96875	0.0	106.6	2.875	3.0	915.0	1464.74975	1775.93
1997	1409.749249	0.0	0.233333	1.0	1.0	1.0	33.96875	0.5	27.6	6.875	3.0	868.0	1131.74975	2800.93
1998	1010.749249	0.0	0.233333	0.0	4.0	1.0	43.96875	-0.1	64.6	3.875	5.0	336.0	169.74975	612.93
1999	8.749249	1.0	1.333333	1.0	5.0	1.0	42.96875	0.7	87.6	4.875	16.0	483.0	253.74975	3662.93

:000 rows × 20 columns

Next steps: Generate code with upd | View recommended plots | New interactive sheet

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(upd,y,test_size=0.2,random_state=30)
print(x_train)
print(x_test)
print(y_train)
print(y_test)
```

```
      battery_power  blue  clock_speed  dual_sim    fc  four_g  int_memory  \
1572     393.749249   0.0    -0.166667       1.0   6.0     0.0    13.96875
1442     471.749249   0.0    -0.166667       1.0   5.0     0.0    50.96875
1516     579.749249   1.0     1.233333       1.0  13.0     1.0    39.96875
259     1057.749249   1.0     0.933333       1.0   6.0     1.0     3.96875
945      330.749249   0.0     0.433333       0.0   0.0     1.0    43.96875
...             ...   ...          ...       ...   ...     ...         ...
500      776.749249   1.0    -0.166667       1.0   1.0     0.0    51.96875
1837     481.749249   0.0    -0.066667       1.0   0.0     1.0    42.96875
941      241.749249   0.0    -0.166667       1.0   0.0     1.0    30.96875
421      676.749249   0.0     1.533333       1.0   6.0     0.0    54.96875
1829     563.749249   0.0     1.033333       1.0   4.0     1.0    45.96875
```

```
      m_dep  mobile_wt  n_cores    pc  px_height    px_width          ram  \
1572    0.1       83.6    6.875  14.0      126.0  1074.74975  2865.935968
1442    0.3      115.6    3.875   7.0      105.0    87.74975  1604.935968
1516   -0.1       20.6    5.875  17.0       83.0   315.74975  2199.935968
259     0.3       81.6    4.875  17.0      179.0  1058.74975  3095.935968
945     0.1       12.6    4.875   0.0      820.0   865.74975  3060.935968
...     ...        ...      ...   ...        ...         ...          ...
500    -0.1       23.6    6.875  16.0      581.0   249.74975   898.935968
1837   -0.1       49.6    2.875  11.0      948.0   704.74975  1795.935968
941     0.1       14.6    0.875   1.0     1587.0  1158.74975    59.935968
421     0.1        3.6    1.875  17.0       96.0   851.74975  2235.935968
1829    0.1       81.6    4.875   8.0     1188.0  1447.74975     8.935968

            sc_h  sc_w  talk_time  three_g  touch_screen  wifi
1572   1.736842   0.0        1.9      0.0           0.0   1.0
1442   7.736842  10.0       13.9      0.0           1.0   0.0
1516   1.736842   4.0        4.9      1.0           0.0   0.0
259    3.736842   1.0        0.9      1.0           1.0   1.0
945    2.736842   0.0       16.9      1.0           1.0   0.0
...         ...   ...        ...      ...           ...   ...
500    1.736842   1.0       17.9      0.0           0.0   1.0
1837  -0.263158   3.0        3.9      1.0           0.0   1.0
941   13.736842  10.0        1.9      1.0           0.0   0.0
421   10.736842  14.0        7.9      1.0           1.0   0.0
1829   2.736842   0.0        9.9      1.0           0.0   0.0

[1600 rows x 20 columns]
      battery_power  blue  clock_speed  dual_sim    fc  four_g  int_memory  \
1856    1284.749249   0.0     1.933333       0.0   6.0     0.0    11.96875
364      414.749249   1.0     1.933333       0.0   0.0     0.0    33.96875
1948     246.749249   0.0     0.733333       0.0   0.0     0.0    22.96875
1458     866.749249   0.0    -0.166667       0.0  10.0     1.0    39.96875
609      710.749249   0.0     1.933333       0.0   1.0     1.0    43.96875
...             ...   ...          ...       ...   ...     ...         ...
1163    1428.749249   1.0     1.333333       0.0  11.0     0.0    13.96875
572     1199.749249   1.0    -0.166667       0.0  13.0     1.0    43.96875
1105    1008.749249   1.0     1.833333       1.0  11.0     0.0    44.96875
1903     859.749249   1.0     0.733333       0.0   1.0     0.0    44.96875
481     1195.749249   1.0     0.033333       0.0   1.0     1.0    31.96875

      m_dep  mobile_wt  n_cores    pc  px_height    px_width          ram  \
1856    0.8       34.6    2.875  15.0      616.0   411.74975  2781.935968
364     0.1      114.6    2.875  12.0      188.0   491.74975  1213.935968
1948    0.8       29.6    5.875  14.0       88.0   208.74975  1717.935968
1458    0.0       10.6   -0.125  20.0      907.0   424.74975  1333.935968
```

Question 2

1. Read the data with pandas and describe the data
2. Find data type and shape of each column
3. Find the target and features
4. Find the null values (if yes fill the null values with '0' or mean of that column)
5. Normalize all the features
6. Split the data into train and test.

```
import pandas as pd
d=pd.read_csv('/content/train.csv')
d
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222	1890	668	13
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915	1965	2032	11
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868	1632	3057	9
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336	670	869	18
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483	754	3919	19

2000 rows × 21 columns

```
d.describe()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	..
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	.
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.046500	0.501750	140.249000	4.520500	.
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.145715	0.288416	35.399655	2.287837	.
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000	1.000000	.
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000	3.000000	.
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000	4.000000	.
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000	7.000000	.
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.000000	8.000000	.

8 rows × 21 columns

```
d.dtypes
```

	0
battery_power	int64
blue	int64
clock_speed	float64
dual_sim	int64
fc	int64
four_g	int64
int_memory	int64
m_dep	float64
mobile_wt	int64
n_cores	int64
pc	int64
px_height	int64
px_width	int64
ram	int64
sc_h	int64
sc_w	int64
talk_time	int64
three_g	int64
touch_screen	int64
wifi	int64
price_range	int64

```
d.dtypes
```

```
2000
```

```
x = d.drop('battery_power', axis=1)
y = d['battery_power']
print(x)
print(y)
```

```
      blue  clock_speed  dual_sim  fc  four_g  int_memory  m_dep  mobile_wt  \
0        0          2.2         0   1       0           7    0.6        188
1        1          0.5         1   0       1          53    0.7        136
2        1          0.5         1   2       1          41    0.9        145
3        1          2.5         0   0       0          10    0.8        131
4        1          1.2         0  13       1          44    0.6        141
...    ...          ...       ...  ..     ...         ...    ...        ...
1995     1          0.5         1   0       1           2    0.8        106
1996     1          2.6         1   0       0          39    0.2        187
1997     0          0.9         1   1       1          36    0.7        108
1998     0          0.9         0   4       1          46    0.1        145
1999     1          2.0         1   5       1          45    0.9        168

      n_cores  pc  px_height  px_width   ram  sc_h  sc_w  talk_time  three_g  \
0           2   2         20       756  2549     9     7         19        0
1           3   6        905      1988  2631    17     3          7        1
2           5   6       1263      1716  2603    11     2          9        1
3           6   9       1216      1786  2769    16     8         11        1
4           2  14       1208      1212  1411     8     2         15        1
...       ...  ..        ...       ...   ...   ...   ...        ...      ...
1995        6  14       1222      1890   668    13     4         19        1
1996        4   3        915      1965  2032    11    10         16        1
1997        8   3        868      1632  3057     9     1          5        1
1998        5   5        336       670   869    18    10         19        1
1999        6  16        483       754  3919    19     4          2        1

      touch_screen  wifi  price_range
0                0     1            1
1                1     0            2
```

```
2                1     0            2
3                0     0            2
4                1     0            1
...            ...   ...          ...
1995             1     0            0
1996             1     1            2
1997             1     0            3
1998             1     1            0
1999             1     1            3

[2000 rows x 20 columns]
0        842
1       1021
2        563
3        615
4       1821
        ...
1995     794
1996    1965
1997    1911
1998    1512
1999     510
Name: battery_power, Length: 2000, dtype: int64
```

```
d.isnull().sum()
```

	0
battery_power	0
blue	0
clock_speed	0
dual_sim	0
fc	0
four_g	0
int_memory	0
m_dep	0
mobile_wt	0
n_cores	0
pc	0
px_height	0
px_width	0
ram	0
sc_h	0
sc_w	0
talk_time	0
three_g	0
touch_screen	0
wifi	0
price_range	0

```
upd=x-x.min()/x.max()-x.min()
upd
```

	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h
0	0.0	1.533333	0.0	1.0	0.0	4.96875	0.4	107.6	0.875	2.0	20.0	255.74975	2292.935968	3.736842
1	1.0	-0.166667	1.0	0.0	1.0	50.96875	0.5	55.6	1.875	6.0	905.0	1487.74975	2374.935968	11.736842
2	1.0	-0.166667	1.0	2.0	1.0	38.96875	0.7	64.6	3.875	6.0	1263.0	1215.74975	2346.935968	5.736842
3	1.0	1.833333	0.0	0.0	0.0	7.96875	0.6	50.6	4.875	9.0	1216.0	1285.74975	2512.935968	10.736842
4	1.0	0.533333	0.0	13.0	1.0	41.96875	0.4	60.6	0.875	14.0	1208.0	711.74975	1154.935968	2.736842
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
1995	1.0	-0.166667	1.0	0.0	1.0	-0.03125	0.6	25.6	4.875	14.0	1222.0	1389.74975	411.935968	7.736842
1996	1.0	1.533333	1.0	0.0	0.0	36.96875	0.5	106.6	2.875	3.0	915.0	1464.74975	1775.935968	5.736842
1997	0.0	0.233333	1.0	1.0	1.0	22.96875	0.5	27.6	6.875	3.0	868.0	1131.74975	2800.935968	2.736842

Next steps: Generate code with upd View recommended plots New interactive sheet

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(upd,y,test_size=0.2,random_state=30)
print(x_train)
print(x_test)
print(y_train)
print(y_test)
```

```
      blue  clock_speed  dual_sim    fc  four_g  int_memory  m_dep  mobile_wt  \
1572   0.0    -0.166667       1.0   6.0     0.0    13.96875    0.1       83.6
1442   0.0    -0.166667       1.0   5.0     0.0    50.96875    0.3      115.6
1516   1.0     1.233333       1.0  13.0     1.0    39.96875   -0.1       20.6
259    1.0     0.933333       1.0   6.0     1.0     3.96875    0.3       81.6
945    0.0     0.433333       0.0   0.0     1.0    43.96875    0.1       12.6
...    ...          ...       ...   ...     ...         ...    ...        ...
500    1.0    -0.166667       1.0   1.0     0.0    51.96875   -0.1       23.6
1837   0.0    -0.066667       1.0   0.0     1.0    42.96875   -0.1       49.6
941    0.0    -0.166667       1.0   0.0     1.0    30.96875    0.1       14.6
421    0.0     1.533333       1.0   6.0     0.0    54.96875    0.1        3.6
1829   0.0     1.033333       1.0   4.0     1.0    45.96875    0.1       81.6

      n_cores    pc  px_height    px_width          ram       sc_h  sc_w  \
1572    6.875  14.0      126.0  1074.74975  2865.935968   1.736842   0.0
1442    3.875   7.0      105.0    87.74975  1604.935968   7.736842  10.0
1516    5.875  17.0       83.0   315.74975  2199.935968   1.736842   4.0
259     4.875  17.0      179.0  1058.74975  3095.935968   3.736842   1.0
945     4.875   0.0      820.0   865.74975  3060.935968   2.736842   0.0
...       ...   ...        ...         ...          ...        ...   ...
500     6.875  16.0      581.0   249.74975   898.935968   1.736842   1.0
1837    2.875  11.0      948.0   704.74975  1795.935968  -0.263158   3.0
941     0.875   1.0     1587.0  1158.74975    59.935968  13.736842  10.0
421     1.875  17.0       96.0   851.74975  2235.935968  10.736842  14.0
1829    4.875   8.0     1188.0  1447.74975     8.935968   2.736842   0.0

      talk_time  three_g  touch_screen  wifi  price_range
1572        1.9      0.0           0.0   1.0          2.0
1442       13.9      0.0           1.0   0.0          1.0
1516        4.9      1.0           0.0   0.0          1.0
259         0.9      1.0           1.0   1.0          3.0
945        16.9      1.0           1.0   0.0          3.0
...         ...      ...           ...   ...          ...
500        17.9      0.0           0.0   1.0          0.0
1837        3.9      1.0           0.0   1.0          1.0
941         1.9      1.0           0.0   0.0          0.0
```