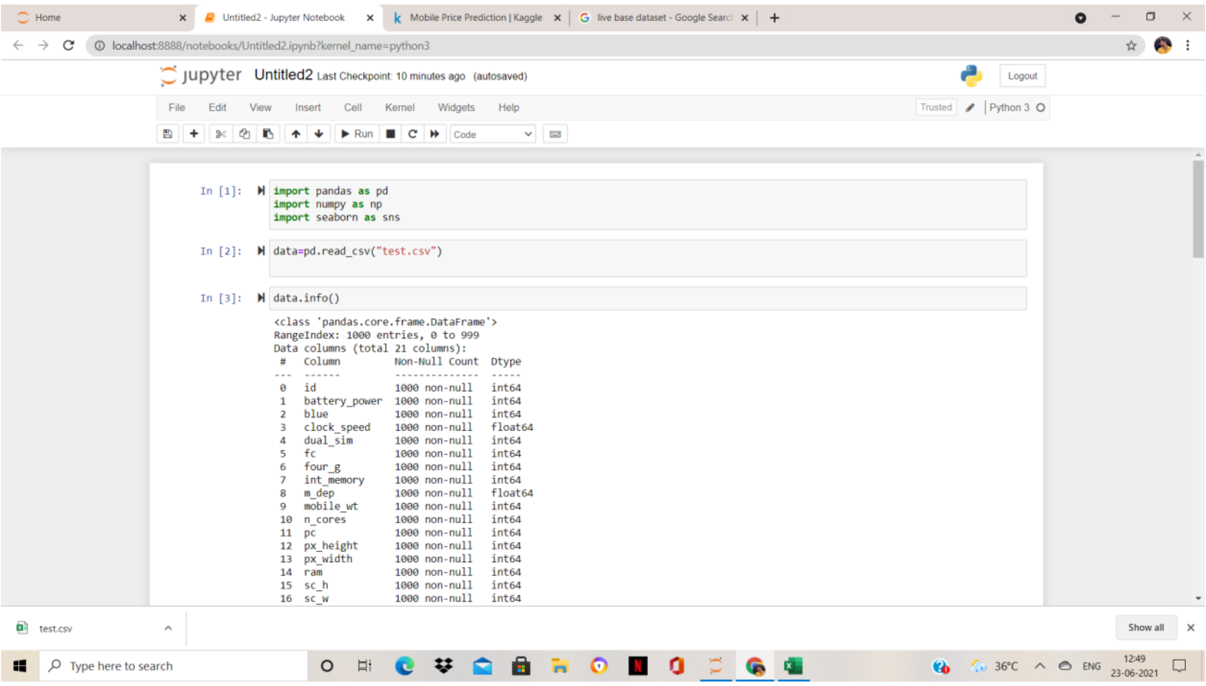
END SEMESTER- DWDM

-R.SRIJILL 18BCS034

Attributes and its types

1.battery_power -numeric 2 blue -binary3 clock_speed - numeric 4 dual_sim -binary 5 fc -numeric 6 four_g -binary 7 int_memory - numeric 8 m_dep - numeric 9 mobile_wt - numeric 10 n_cores - numeric 11 pc - numeric 12 px_height - numeric 13 px_width - numeric 14 ram - numeric 15 sc_h - numeric 16 sc_w - numeric 17 talk_time - numeric 18 three_g -binary 19 touch_screen -binary 20 wifi -binary

2.



```
 6   four_g         1000 non-null   int64
 7   int_memory     1000 non-null   int64
 8   m_dep          1000 non-null   float64
 9   mobile_wt      1000 non-null   int64
 10  n_cores        1000 non-null   int64
 11  pc             1000 non-null   int64
 12  px_height      1000 non-null   int64
 13  px_width       1000 non-null   int64
 14  ram            1000 non-null   int64
 15  sc_h           1000 non-null   int64
 16  sc_w           1000 non-null   int64
 17  talk_time      1000 non-null   int64
 18  three_g        1000 non-null   int64
 19  touch_screen   1000 non-null   int64
 20  wifi           1000 non-null   int64
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

In [4]:
```
data.head(5)
```

Out[4]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7	2
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6	0	7
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17	10	10
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10	0	7
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15	8	7

5 rows × 21 columns

In [5]:
```
!pip install sklearn
```

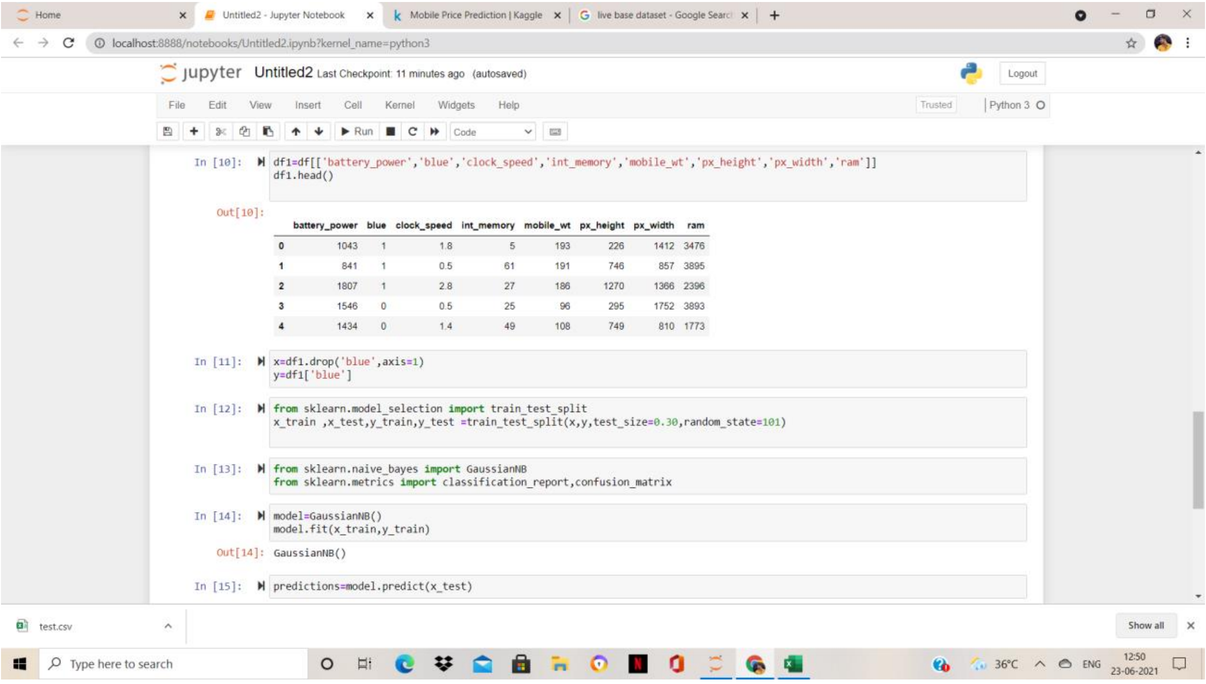
```
Collecting sklearn
  Downloading sklearn-0.0.tar.gz (1.1 kB)
Requirement already satisfied: scikit-learn in c:\users\jill\anaconda3\lib\site-packages (from sklearn) (0.23.2)
Requirement already satisfied: scipy>=0.19.1 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.5.
2)
Requirement already satisfied: numpy>=1.13.3 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.1
9.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklear
n) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\jill\anaconda3\lib\site-packages (from scikit-learn->sklearn) (0.17.
0)
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py): started
  Building wheel for sklearn (setup.py): finished with status 'done'
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl size=1321 sha256=85205ab8379b15bf8dda98315eb4f6d7122e
4804481489c09857aca9878b48db
  Stored in directory: c:\users\jill\appdata\local\pip\cache\wheels\22\0b\40\fd3f795caaa1fb4c6cb738bc1f56100be1e57da95849bfc
897
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0
```

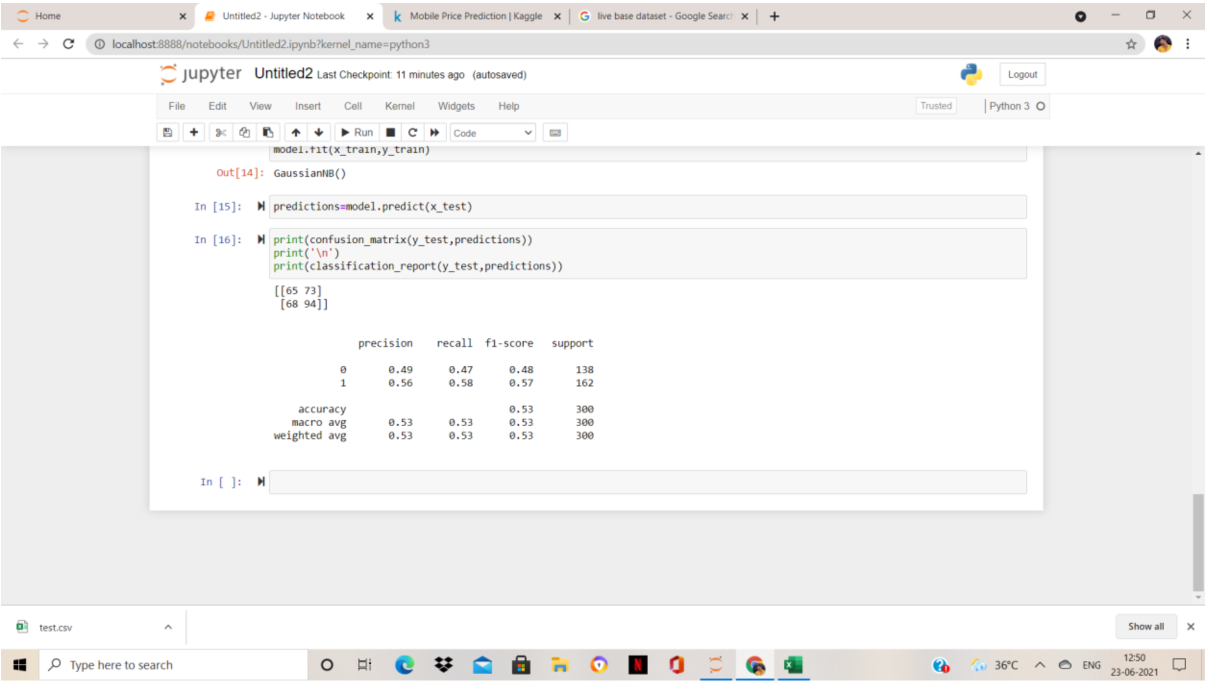
In [6]:
```
from sklearn import preprocessing
```

In [7]:
```
from sklearn.model_selection import train_test_split
```

In [8]:
```
df=pd.read_csv("test.csv")
```

In [10]:
```
df1=df[['battery_power','blue','clock_speed','int_memory','mobile_wt','px_height','px_width','ram']]
df1.head()
```





3. https://github.com/SRIJILL123/ENDSEM-18BCS034

4.

DWDM-ENDSEM

R.Srijill
18BCS034

Naive Bayes classifiers are a collection of classification Algorithms based on Bayes Theorem. It is not a single algorithm but a family of algorithm but a family of algorithm where all of them share a common principles.

$$P(A|B) = \frac{P(B)(A)\,P(A)}{P(B)}$$

Steps to implement Naive Bayes

1. Seporate By Class
2. Summarize Dataset
3. Summarize Data By class
4. Gaussian Probability Desity Function
5. Class Probabilities.

To calculate the probability of a even occuring we count how many times are event of interest can occur and dividing it by the Smaple space.

How to calculate Confusion Matrix :-

1. First you need to test dataset with its expected outcome Values.

2. Predict all the rows in the test dataset.

3. Calculate the expected Predictions and outcomes.

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{sensitivity} = \frac{Tp}{Tp + FN}$$

Predictions = model.Predict (x_File Name)

Accuracy is calculated as the total number of two correct predictions (TP+TN) divided by the total number of data set (P+N)