Mobile phone price front on the various factors like processor, RAM, internal memory etc.
Primary goal is to predict price in the dataset. Classify the mobile price range as 0,1,2 or 3 based upon the features given in the dataset.

Class 0 - low cost

Class 1 - medium cost

Class 2 - high cost

Class 3 - very high cost

Use SVM algorithm for classification.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn import preprocessing
```

```
%matplotlib inline
sns.set_palette("husl")
sns.set_style("whitegrid")
```

```
data = pd.read_csv('train.csv')
```

```
data.shape
```

```
(2000, 21)
```

```
data.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
battery_power    2000 non-null int64
blue             2000 non-null int64
clock_speed      2000 non-null float64
dual_sim         2000 non-null int64
fc               2000 non-null int64
four_g           2000 non-null int64
int_memory       2000 non-null int64
m_dep            2000 non-null float64
mobile_wt        2000 non-null int64
n_cores          2000 non-null int64
pc               2000 non-null int64
px_height        2000 non-null int64
px_width         2000 non-null int64
ram              2000 non-null int64
sc_h             2000 non-null int64
sc_w             2000 non-null int64
talk_time        2000 non-null int64
three_g          2000 non-null int64
touch_screen     2000 non-null int64
wifi             2000 non-null int64
price_range      2000 non-null int64
dtypes: float64(3), int64(18)
memory usage: 328.1 KB
```

```
data.head(5)
```

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	sc_w	talk_time	three_g	touch_screen	wifi	price_range
842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	0	1	1
1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	1	0	2
563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	1	0	2
615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16	8	11	1	0	0	2
1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8	2	15	1	1	0	1

EDA

```
df.shape
```

```
(2000, 21)
```

```
df.columns
```

```
Index(['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', 'sc_w', 'talk_time', 'three_g',
       'touch_screen', 'wifi', 'price_range'],
      dtype='object')
```

```
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
battery_power    2000 non-null int64
blue             2000 non-null int64
clock_speed      2000 non-null float64
dual_sim         2000 non-null int64
fc               2000 non-null int64
four_g           2000 non-null int64
int_memory       2000 non-null int64
m_dep            2000 non-null float64
mobile_wt        2000 non-null int64
n_cores          2000 non-null int64
pc               2000 non-null int64
px_height        2000 non-null int64
px_width         2000 non-null int64
ram              2000 non-null int64
sc_h             2000 non-null int64
sc_w             2000 non-null int64
talk_time        2000 non-null int64
three_g          2000 non-null int64
touch_screen     2000 non-null int64
wifi             2000 non-null int64
price_range      2000 non-null int64
dtypes: float64(3), int64(18)
memory usage: 328.1 KB
```

battery_power - Total energy a battery can store in one time measure in mAh

blue - Has bluetooth or not

clock_speed - speed at which microprocessor executes instructions

dual_sim - Has dual sim support or not

fc - Front Camera mega pixels

four_g - Has 4G or not

int_memory - Internal memory in Gigabytes

m_dep - Mobile depth in cm

mobile_wt - Weight of mobile phone

n_cores - Number of cores of processor

pc - Primary Camera mega pixels

px_height - Pixel resolution height

px_width - Pixel resolution width

ram - Random Access Memory in Megabytes

sc_h - Screen height of mobile in cm

sc_w - Screen width of mobile in cm

talk_time - The longest time that a single battery charge will last when you are

three_g - Has 3G or not

touch_screen - Has touch screen or not

wifi - Has wifi or not

price_range - 0 (low cost), 1 (medium cost), 2 (high cost), 3 (very high cost)

```
df.describe()
```

	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	sc_w	talk_time	three_g	touch_screen	wifi	price_range
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	1238.518500	0.495000	1.522250	0.509500	4.309500	0.521500	32.046500	0.501750	140.249000	4.520500	9.916500	645.108000	1251.515500	2124.213000	12.306500	5.767000	11.011000	0.761500	0.503000	0.507000	1.500000
std	439.418206	0.500035	0.816004	0.500035	4.341444	0.499662	18.145715	0.288416	35.399655	2.287837	6.064315	443.780811	432.199447	1084.732044	4.213245	4.356398	5.463955	0.426273	0.500035	0.500035	1.118314
min	501.000000	0.000000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000	1.000000	0.000000	0.000000	500.000000	256.000000	5.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000
25%	851.750000	0.000000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000	3.000000	4.000000	282.750000	874.750000	1207.500000	9.000000	2.000000	6.000000	0.000000	0.000000	0.000000	0.750000
50%	1226.000000	0.000000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000	4.000000	10.000000	564.000000	1247.000000	2146.500000	12.000000	5.000000	11.000000	1.000000	1.000000	1.000000	1.500000
75%	1615.250000	1.000000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000	7.000000	15.000000	947.250000	1633.000000	3064.500000	16.000000	9.000000	16.000000	1.000000	1.000000	1.000000	2.250000
max	1998.000000	1.000000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.000000	8.000000	20.000000	1960.000000	1998.000000	3998.000000	19.000000	18.000000	20.000000	1.000000	1.000000	1.000000	3.000000

```
df.isnull().sum()
```

```
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
dtypes: int64    0
```

```
df.corr()
```

	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	sc_w	talk_time	three_g	touch_screen	wifi	price_range
battery_power	1.000000	0.011482	-0.011756	0.041072	0.035199	0.014772	0.053349	0.031441	-0.000605	0.038313	0.033599	0.015148	0.041099	0.011482	-0.027189	0.033772	0.052510	0.011689	0.010351	0.027443	0.200723
blue	0.011482	1.000000	0.021419	0.035160	0.003593	0.013404	0.041177	0.004049	0.036161	0.004049	-0.006060	-0.041533	0.026351	0.013356	-0.030267	-0.021421	0.034160	0.027099	0.031213	-0.029959	0.020573
clock_speed	-0.011756	0.021419	1.000000	-0.012065	-0.027085	-0.043073	0.169454	0.006545	-0.014364	0.012772	-0.001212	-0.029959	-0.027562	-0.024658	-0.020189	-0.018786	-0.046435	-0.044054	-0.008740	-0.028017	-0.029727
dual_sim	0.041072	0.035160	-0.012065	1.000000	-0.029123	0.003187	-0.029133	0.001396	0.018723	-0.009964	0.030772	-0.016560	-0.029077	0.022420	-0.009476	-0.008672	0.014733	-0.015179	-0.022857	0.014083	0.017444
fc	0.035199	0.003593	-0.027085	-0.029123	1.000000	-0.016560	-0.029032	-0.001791	0.023618	-0.009390	0.645176	-0.022089	-0.029133	-0.009952	0.021421	0.043980	-0.011014	-0.012065	-0.028533	-0.009952	0.021998
four_g	0.014772	0.013404	-0.043073	0.003187	-0.016560	1.000000	0.024772	-0.012462	-0.009008	0.016274	-0.010810	0.027189	-0.022773	-0.029454	-0.016337	-0.029454	-0.046628	0.584246	-0.012065	-0.030551	0.014772
int_memory	0.053349	0.041177	0.169454	-0.029133	-0.029032	0.024772	1.000000	0.006886	-0.034214	-0.029959	-0.028310	0.029133	0.026351	0.033334	-0.028533	-0.021421	0.034160	0.027099	0.031213	-0.029959	0.044435
m_dep	0.031441	0.004049	0.006545	0.001396	-0.001791	-0.012462	0.006886	1.000000	0.039042	-0.028087	0.025263	0.023566	0.017814	0.026351	-0.009364	-0.027271	0.028087	-0.004404	0.023566	0.012350	0.000853
mobile_wt	-0.000605	0.036161	-0.014364	0.018723	0.023618	-0.009008	-0.034214	0.039042	1.000000	-0.029959	-0.028310	0.029133	0.026351	0.033334	-0.028533	-0.021421	0.034160	0.027099	0.031213	-0.029959	-0.030302
n_cores	0.038313	0.004049	0.012772	-0.009964	-0.009390	0.016274	-0.029959	-0.028087	-0.029959	1.000000	-0.028310	0.029133	0.026351	0.033334	-0.028533	-0.021421	0.034160	0.027099	0.031213	-0.029959	0.004399
pc	0.033599	-0.006060	-0.001212	0.030772	0.645176	-0.010810	-0.028310	0.025263	-0.028310	-0.028310	1.000000	0.029133	0.026351	0.033334	-0.028533	-0.021421	0.034160	0.027099	0.031213	-0.029959	0.033599
px_height	0.015148	-0.041533	-0.029959	-0.016560	-0.022089	0.027189	0.029133	0.023566	0.029133	0.029133	0.029133	1.000000	0.511454	0.498908	0.043037	0.023774	-0.010441	0.051824	0.030688	0.148086	0.148690
px_width	0.041099	0.026351	-0.027562	-0.029077	-0.029133	-0.022773	0.026351	0.017814	0.026351	0.026351	0.026351	0.511454	1.000000	0.470476	0.051454	0.030691	0.007246	0.070069	0.024213	0.165818	0.165171
ram	0.011482	0.013356	-0.024658	0.022420	-0.009952	-0.029454	0.033334	0.026351	0.033334	0.033334	0.033334	0.498908	0.470476	1.000000	0.043037	0.030691	0.007246	0.070069	0.024213	0.165818	0.917046
sc_h	-0.027189	-0.030267	-0.020189	-0.009476	0.021421	-0.016337	-0.028533	-0.009364	-0.028533	-0.028533	-0.028533	0.043037	0.051454	0.043037	1.000000	0.507454	-0.029078	-0.044099	0.023774	0.029960	0.022986
sc_w	0.033772	-0.021421	-0.018786	-0.008672	0.043980	-0.029454	-0.021421	-0.027271	-0.021421	-0.021421	-0.021421	0.023774	0.030691	0.030691	0.507454	1.000000	-0.029454	-0.029454	-0.029454	0.029454	0.038711
talk_time	0.052510	0.034160	-0.046435	0.014733	-0.011014	-0.046628	0.034160	0.028087	0.034160	0.034160	0.034160	-0.010441	0.007246	0.007246	-0.029078	-0.029454	1.000000	0.043272	0.011351	0.044711	0.021859
three_g	0.011689	0.027099	-0.044054	-0.015179	-0.012065	0.584246	0.027099	-0.004404	0.027099	0.027099	0.027099	0.051824	0.070069	0.070069	-0.044099	-0.029454	0.043272	1.000000	0.011351	0.044711	0.023611
touch_screen	0.010351	0.031213	-0.008740	-0.022857	-0.028533	-0.012065	0.031213	0.023566	0.031213	0.031213	0.031213	0.030688	0.024213	0.024213	0.023774	-0.029454	0.011351	0.011351	1.000000	0.029454	-0.028099
wifi	0.027443	-0.029959	-0.028017	0.014083	-0.009952	-0.030551	-0.029959	0.012350	-0.029959	-0.029959	-0.029959	0.148086	0.165818	0.165818	0.029960	0.029454	0.044711	0.044711	0.029454	1.000000	0.147511
price_range	0.200723	0.020573	-0.029727	0.017444	0.021998	0.014772	0.044435	0.000853	-0.030302	0.004399	0.033599	0.148690	0.165171	0.917046	0.022986	0.038711	0.021859	0.023611	-0.028099	0.147511	1.000000

```
sns.heatmap(df.corr)
```

```
```