SUPPORT VECTOR MACHINE(SVM)

In [27]:

```
import numpy as np
import pandas as pd
from sklearn.datasets import make_classification
from sklearn import svm
from sklearn.model_selection import train_test_split
```

In [28]:

```
df=pd.read_csv('C:/Users/USER/Desktop/petrol_consumption.csv')
df
```

Out[28]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consump
0	9.00	3571	1976	0.525	
1	9.00	4092	1250	0.572	
2	9.00	3865	1586	0.580	
3	7.50	4870	2351	0.529	
4	8.00	4399	431	0.544	
5	10.00	5342	1333	0.571	
6	8.00	5319	11868	0.451	
7	8.00	5126	2138	0.553	
8	8.00	4447	8577	0.529	
9	7.00	4512	8507	0.552	
10	8.00	4391	5939	0.530	
11	7.50	5126	14186	0.525	
12	7.00	4817	6930	0.574	
13	7.00	4207	6580	0.545	
14	7.00	4332	8159	0.608	
15	7.00	4318	10340	0.586	
16	7.00	4206	8508	0.572	
17	7.00	3718	4725	0.540	
18	7.00	4716	5915	0.724	
19	8.50	4341	6010	0.677	
20	7.00	4593	7834	0.663	
21	8.00	4983	602	0.602	
22	9.00	4897	2449	0.511	
23	9.00	4258	4686	0.517	
24	8.50	4574	2619	0.551	
25	9.00	3721	4746	0.544	
26	8.00	3448	5399	0.548	
27	7.50	3846	9061	0.579	
28	8.00	4188	5975	0.563	
29	9.00	3601	4650	0.493	
30	7.00	3640	6905	0.518	
31	7.00	3333	6594	0.513	
32	8.00	3063	6524	0.578	
33	7.50	3357	4121	0.547	

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consump
34	8.00	3528	3495	0.487	
35	6.58	3802	7834	0.629	
36	5.00	4045	17782	0.566	
37	7.00	3897	6385	0.586	
38	8.50	3635	3274	0.663	
39	7.00	4345	3905	0.672	
40	7.00	4449	4639	0.626	
41	7.00	3656	3985	0.563	
42	7.00	4300	3635	0.603	
43	7.00	3745	2611	0.508	
44	6.00	5215	2302	0.672	
45	9.00	4476	3942	0.571	
46	7.00	4296	4083	0.623	
47	7.00	5002	9794	0.593	

In [29]:

```
classes = 4
X,t= make_classification(100, 5, n_classes = classes, random_state= 40, n_informative = 2,
```

In [30]:

```
X_train, X_test, y_train, y_test=  train_test_split(X, t , test_size=0.50)
```

In [31]:

```
model = svm.SVC(kernel = 'linear', random_state = 0, C=1.0)
```

In [32]:

```
model.fit(X_train, y_train)
```

Out[32]:

```
SVC(kernel='linear', random_state=0)
```

In [33]:

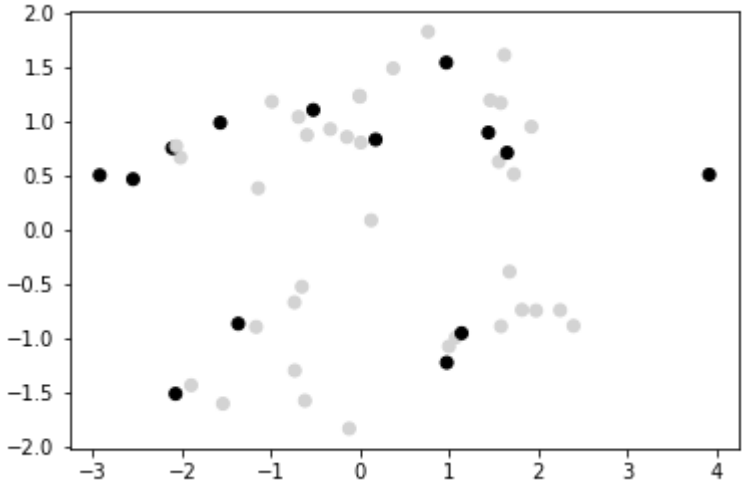
```
y=model.predict(X_test)
y2=model.predict(X_train)
```

In [34]:

```
import matplotlib.pyplot as plt
color = ['black' if c == 0 else 'lightgrey' for c in y]
plt.scatter(X_train[:,0], X_train[:,1], c=color)
```

Out[34]:

```
<matplotlib.collections.PathCollection at 0x17a00341490>
```



In [11]:

```
from sklearn.metrics import accuracy_score
score =accuracy_score(y, y_test)
print(score*100)
score2 =accuracy_score(y2, y_train)
print(score2*100)
```

```
84.0
96.0
```