```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data_train=pd.read_csv("mobiledata.csv")
data_test=pd.read_csv("mobiledata.csv")
```

```
data_train.head()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208

5 rows × 21 columns

```
data_test.head()
```

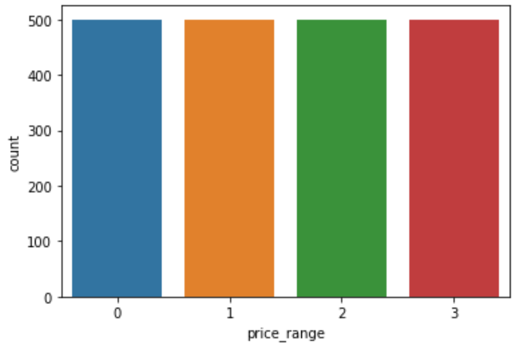
Automatic saving failed. This file was updated remotely or in another tab. Show diff

								m_dep	mobile_wt	n_cores	...	px_height
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208

5 rows × 21 columns

```
sns.countplot(data_train['price_range'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as
  warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7fe01333cee0>
```



```
data_train.shape,data_test.shape
```

```
((2000, 21), (2000, 21))
```

```
data_train.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
dtype: int64
```

```
data_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

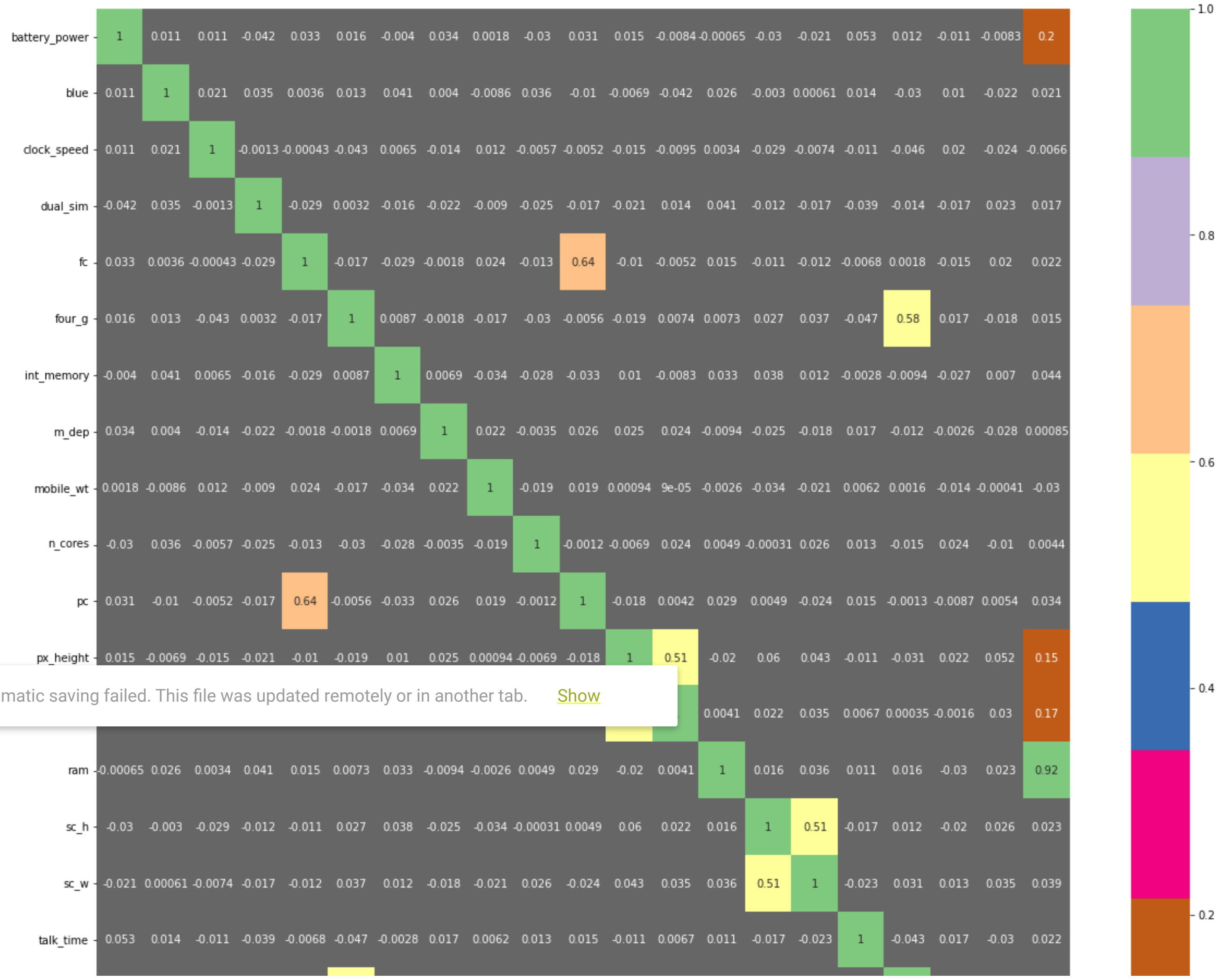
```
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

```
data_train.describe()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_core
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.00000
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.046500	0.501750	140.249000	4.52050
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.145715	0.288416	35.399655	2.28783
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000	1.00000
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000	3.00000
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000	4.00000
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000	7.00000
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.000000	8.00000

8 rows × 21 columns

```
import seaborn as sns
plt.figure(figsize=(20,20))
sns.heatmap(data_train.corr(),annot=True,cmap=plt.cm.Accent_r)
plt.show()
```

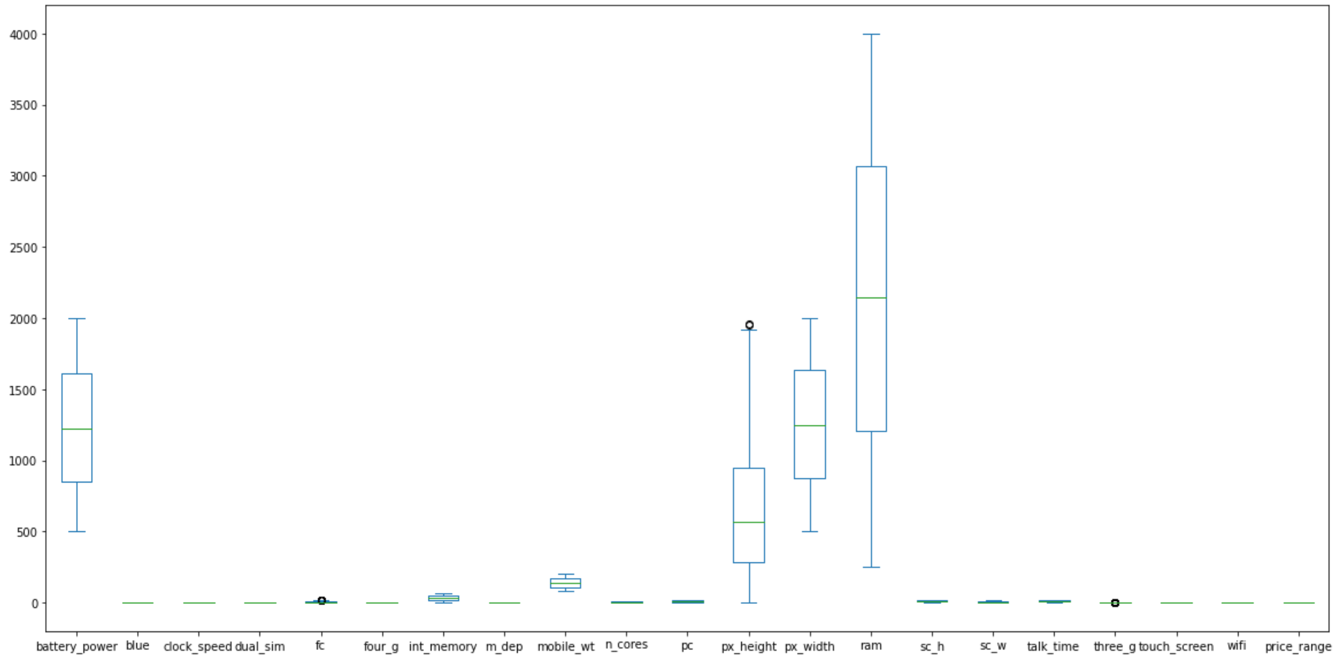


battery_power	1	0.011	0.011	-0.042	0.033	0.016	-0.004	0.034	0.0018	-0.03	0.031	0.015	-0.0084	-0.00065	-0.03	-0.021	0.053	0.012	-0.011	-0.0083	0.2
blue	0.011	1	0.021	0.035	0.0036	0.013	0.041	0.004	-0.0086	0.036	-0.01	-0.0069	-0.042	0.026	-0.003	0.00061	0.014	-0.03	0.01	-0.022	0.021
clock_speed	0.011	0.021	1	-0.0013	-0.00043	-0.043	0.0065	-0.014	0.012	-0.0057	-0.0052	-0.015	-0.0095	0.0034	-0.029	-0.0074	-0.011	-0.046	0.02	-0.024	-0.0066
dual_sim	-0.042	0.035	-0.0013	1	-0.029	0.0032	-0.016	-0.022	-0.009	-0.025	-0.017	-0.021	0.014	0.041	-0.012	-0.017	-0.039	-0.014	-0.017	0.023	0.017
fc	0.033	0.0036	-0.00043	-0.029	1	-0.017	-0.029	-0.0018	0.024	-0.013	0.64	-0.01	-0.0052	0.015	-0.011	-0.012	-0.0068	0.0018	-0.015	0.02	0.022
four_g	0.016	0.013	-0.043	0.0032	-0.017	1	0.0087	-0.0018	-0.017	-0.03	-0.0056	-0.019	0.0074	0.0073	0.027	0.037	-0.047	0.58	0.017	-0.018	0.015
int_memory	-0.004	0.041	0.0065	-0.016	-0.029	0.0087	1	0.0069	-0.034	-0.028	-0.033	0.01	-0.0083	0.033	0.038	0.012	-0.0028	-0.0094	-0.027	0.007	0.044
m_dep	0.034	0.004	-0.014	-0.022	-0.0018	-0.0018	0.0069	1	0.022	-0.0035	0.026	0.025	0.024	-0.0094	-0.025	-0.018	0.017	-0.012	-0.0026	-0.028	0.00085
mobile_wt	0.0018	-0.0086	0.012	-0.009	0.024	-0.017	-0.034	0.022	1	-0.019	0.019	0.00094	9e-05	-0.0026	-0.034	-0.021	0.0062	0.0016	-0.014	-0.00041	-0.03
n_cores	-0.03	0.036	-0.0057	-0.025	-0.013	-0.03	-0.028	-0.0035	-0.019	1	-0.0012	-0.0069	0.024	0.0049	-0.00031	0.026	0.013	-0.015	0.024	-0.01	0.0044
pc	0.031	-0.01	-0.0052	-0.017	0.64	-0.0056	-0.033	0.026	0.019	-0.0012	1	-0.018	0.0042	0.029	0.0049	-0.024	0.015	-0.0013	-0.0087	0.0054	0.034
px_height	0.015	-0.0069	-0.015	-0.021	-0.01	-0.019	0.01	0.025	0.00094	-0.0069	-0.018	1	0.51	-0.02	0.06	0.043	-0.011	-0.031	0.022	0.052	0.15
														0.0041	0.022	0.035	0.0067	0.00035	-0.0016	0.03	0.17
ram	-0.00065	0.026	0.0034	0.041	0.015	0.0073	0.033	-0.0094	-0.0026	0.0049	0.029	-0.02	0.0041	1	0.016	0.036	0.011	0.016	-0.03	0.023	0.92
sc_h	-0.03	-0.003	-0.029	-0.012	-0.011	0.027	0.038	-0.025	-0.034	-0.00031	0.0049	0.06	0.022	0.016	1	0.51	-0.017	0.012	-0.02	0.026	0.023
sc_w	-0.021	0.00061	-0.0074	-0.017	-0.012	0.037	0.012	-0.018	-0.021	0.026	-0.024	0.043	0.035	0.036	0.51	1	-0.023	0.031	0.013	0.035	0.039
talk_time	0.053	0.014	-0.011	-0.039	-0.0068	-0.047	-0.0028	0.017	0.0062	0.013	0.015	-0.011	0.0067	0.011	-0.017	-0.023	1	-0.043	0.017	-0.03	0.022

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
data_train.plot(kind='box',figsize=(20,10))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe0132c9b80>
```



```
x=data_train.drop('price_range',axis=1)
y=data_train['price_range']
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.1,random_state=101)
```

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

```
X_test=sc.transform(X_test)
test=sc.transform(X_test)
```

```
X_train
```

```
array([[-1.62737257, -0.98675438, -1.01271559, ..., -1.78222729,
        -1.00892875, -0.99888951],
       [-0.75199354,  1.01342342,  0.58093235, ..., -1.78222729,
         0.99115027, -0.99888951],
       [-0.20630271,  1.01342342,  0.70352065, ...,  0.56109566,
        -1.00892875,  1.00111173],
       ...,
       [ 0.69636086,  1.01342342, -0.03200917, ...,  0.56109566,
        -1.00892875, -0.99888951],
       [ 0.83733099, -0.98675438, -1.2578922 , ...,  0.56109566,
        -1.00892875,  1.00111173],
       [ 0.4144206 , -0.98675438, -0.39977408, ...,  0.56109566,
         0.99115027,  1.00111173]])
```

```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
dtc.fit(X_train,Y_train)
```

```
DecisionTreeClassifier()
```

```
pred=dtc.predict(X_test)
pred
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
                                                                    1,
                                                                    2,
       3, 1, 1, 3, 3, 1, 0, 0, 2, 3, 3, 2, 0, 3, 3, 3, 2, 2, 3, 1, 3, 1,
       0, 0, 0, 2, 1, 2, 3, 2, 2, 3, 3, 2, 0, 3, 0, 0, 2, 1, 2, 2, 2, 1,
       0, 0, 3, 3, 0, 2, 0, 3, 2, 0, 2, 3, 0, 2, 2, 3, 0, 3, 0, 0, 2, 0,
       1, 0, 3, 2, 2, 2, 1, 3, 2, 0, 3, 3, 2, 3, 1, 3, 3, 2, 1, 1, 0, 0,
       1, 1, 0, 2, 3, 0, 2, 3, 1, 3, 0, 1, 0, 0, 1, 3, 2, 0, 2, 1, 3, 2,
       3, 3, 2, 0, 3, 1, 2, 2, 2, 2, 1, 2, 1, 1, 3, 3, 1, 2, 0, 3, 1, 3,
       1, 2, 3, 1, 2, 1, 0, 1, 3, 3, 1, 2, 1, 3, 1, 0, 2, 3, 0, 3, 0, 0,
       3, 0])
```

```
from sklearn.metrics import accuracy_score, confusion_matrix
dtc_acc=accuracy_score(pred,Y_test)
print(dtc_acc)
print(confusion_matrix(pred,Y_test))
```

```
0.825
[[44  6  0  0]
 [ 6 36  6  0]
 [ 0  4 45  2]
 [ 0  0 11 40]]
```

```
from sklearn.svm import SVC
knn=SVC()
knn.fit(X_train,Y_train)
```

```
SVC()
```

```
pred1=knn.predict(X_test)
pred1
```

```
array([1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 3, 1,
       2, 3, 2, 2, 2, 2, 0, 0, 2, 3, 0, 0, 3, 0, 0, 0, 1, 1, 1, 1, 3, 2,
       3, 0, 2, 3, 3, 1, 0, 1, 2, 3, 2, 2, 0, 3, 2, 3, 2, 2, 3, 1, 3, 1,
       0, 1, 0, 2, 1, 2, 3, 2, 1, 3, 3, 2, 1, 2, 0, 0, 2, 2, 2, 2, 2, 1,
       0, 0, 3, 2, 0, 2, 0, 3, 2, 0, 2, 3, 0, 1, 3, 3, 0, 3, 0, 0, 2, 0,
       1, 0, 3, 2, 1, 1, 1, 3, 1, 0, 3, 2, 2, 3, 1, 2, 3, 2, 1, 1, 1, 0,
       0, 1, 0, 1, 3, 0, 2, 3, 1, 3, 0, 0, 0, 1, 1, 3, 2, 0, 2, 0, 2, 2,
       3, 2, 2, 0, 3, 2, 2, 2, 1, 2, 1, 2, 1, 0, 3, 3, 1, 2, 0, 3, 1, 3,
       2, 2, 3, 2, 1, 1, 0, 1, 2, 2, 2, 2, 0, 3, 1, 0, 2, 2, 0, 2, 0, 0,
       3, 0])
```

```
from sklearn.metrics import accuracy_score, confusion_matrix
svc_acc=accuracy_score(pred1,Y_test)
print(svc_acc)
print(confusion_matrix(pred1,Y_test))
```

```
0.88
[[46  3  0  0]
 [ 4 40  8  0]
 [ 0  3 52  4]
 [ 0  0  2 38]]
```

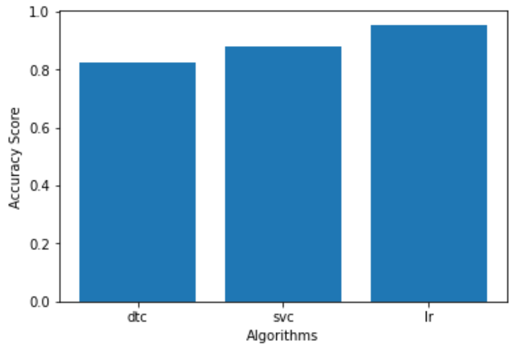
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(X_train,Y_train)

     LogisticRegression()
```

```
pred2=lr.predict(X_test)
pred2

     array([1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 3, 1,
            2, 3, 2, 2, 2, 2, 0, 0, 2, 3, 0, 0, 3, 0, 0, 0, 1, 1, 1, 2, 3, 2,
            3, 0, 1, 3, 3, 1, 0, 0, 3, 3, 3, 3, 1, 3, 2, 3, 2, 2, 3, 1, 3, 1,
            0, 0, 0, 2, 1, 2, 3, 2, 1, 3, 3, 2, 0, 2, 0, 0, 2, 1, 2, 2, 2, 1,
            0, 0, 3, 2, 0, 2, 0, 3, 2, 0, 2, 3, 0, 1, 3, 3, 0, 3, 0, 0, 2, 0,
            1, 0, 3, 2, 2, 1, 1, 3, 1, 0, 3, 2, 2, 3, 1, 2, 3, 2, 1, 1, 1, 0,
            0, 1, 0, 2, 3, 0, 2, 3, 1, 3, 0, 0, 0, 1, 1, 2, 2, 0, 3, 1, 2, 2,
            3, 2, 2, 0, 3, 2, 2, 2, 2, 2, 1, 2, 1, 1, 3, 3, 1, 2, 0, 3, 1, 3,
            2, 2, 3, 2, 2, 1, 0, 1, 3, 2, 1, 2, 0, 3, 1, 0, 2, 2, 0, 2, 0, 0,
            3, 0])
```

```
from sklearn.metrics import accuracy_score, confusion_matrix
lr_acc=accuracy_score(pred2,Y_test)
print(lr_acc)
print(confusion_matrix(pred2,Y_test))

     0.955
     [[49  1  0  0]
      [ 1 45  3  0]
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
plt.bar(x=['dtc','svc','lr'],height=[dtc_acc,svc_acc,lr_acc])
plt.xlabel("Algorithms")
plt.ylabel("Accuracy Score")
plt.show()
```



Colab paid products - Cancel contracts here

✓ 0s completed at 12:10 PM