Experiment 2: Feature Selection by using Mobile Dataset

In [7]:
```
import pandas as pd
df=pd.read_csv('mobile_data.csv')
df.head()
```

Out[7]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_de
0	842	0	2.2	0	1	0	7	0.
1	1021	1	0.5	1	0	1	53	0.
2	563	1	0.5	1	2	1	41	0.
3	615	1	2.5	0	0	0	10	0.
4	1821	1	1.2	0	13	1	44	0.

5 rows × 21 columns

In [8]:
```
### Univariate selection
x=df.iloc[:,:-1]
y=df['price_range']
```

In [9]:
```
x.head()
```

Out[9]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_de
0	842	0	2.2	0	1	0	7	0.
1	1021	1	0.5	1	0	1	53	0.
2	563	1	0.5	1	2	1	41	0.
3	615	1	2.5	0	0	0	10	0.
4	1821	1	1.2	0	13	1	44	0.

In [10]:
```
y.head()
```

Out[10]:
```
0    1
1    2
2    2
3    2
4    1
Name: price_range, dtype: int64
```

In [11]:
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

SelectKBest selects the top features based on their scores using a statistical test, such as chi squared test or ANOVA F-test. The score measures the dependency between each feature and the target variable,and the K features with the highest scores with the highest scores are selected.

In [13]:
```
## Apply SelectKBest Algorithm
ordered_rank_features = SelectKBest(score_func=chi2,k=8)
ordered_rank_features
```

Out[13]:
```
SelectKBest
SelectKBest(k=8, score_func=<function chi2 at 0x0000029FF9D8F920>)
```

In [14]:
```
ordered_feature = ordered_rank_features.fit(x,y)
ordered_feature
```

Out[14]:
```
SelectKBest
SelectKBest(k=8, score_func=<function chi2 at 0x0000029FF9D8F920>)
```

In [15]:
```
dfscore=pd.DataFrame(ordered_feature.scores_,columns=['Score'])
dfcolumns=pd.DataFrame(x.columns)
#dfcolumns
```

In [16]:
```
features_rank = pd.concat([dfcolumns,dfscore],axis=1)
```

In [17]:
```
features_rank.columns=['Features','Score']
features_rank
```

Out[17]:

	Features	Score
0	battery_power	14129.866576
1	blue	0.723232
2	clock_speed	0.648366
3	dual_sim	0.631011
4	fc	10.135166
5	four_g	1.521572
6	int_memory	89.839124
7	m_dep	0.745820
8	mobile_wt	95.972863
9	n_cores	9.097556
10	pc	9.186054
11	px_height	17363.569536
12	px_width	9810.586750
13	ram	931267.519053
14	sc_h	9.614878
15	sc_w	16.480319
16	talk_time	13.236400
17	three_g	0.327643
18	touch_screen	1.928429
19	wifi	0.422091

In [29]:

```
features_rank.nlargest(10,'Score')
```

Out[29]:

	Features	Score
13	ram	931267.519053
11	px_height	17363.569536
0	battery_power	14129.866576
12	px_width	9810.586750
8	mobile_wt	95.972863
6	int_memory	89.839124
15	sc_w	16.480319
16	talk_time	13.236400
4	fc	10.135166
14	sc_h	9.614878

Correlation

In [32]:

```
import matplotlib .pyplot as plt
import seaborn as sns
corr = df.iloc[:,:-1].corr()
```

In [34]:

```
top_features = corr.index
plt.figure(figsize=(20,20))
sns.heatmap(df[top_features].corr(),annot=True)
```

Out[34]:

```
<Axes: >
```