

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv('train.csv')
df.head()
```

Out[2]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	0	1	1
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	1	0	2
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	1	1	0	2
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	1	0	0	2
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	1	0	1

5 rows × 21 columns

```
In [3]: X = df.iloc[:,0:20]
y = df.iloc[:, -1]
```

```
In [4]: y.head()
```

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

```
In [5]: from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

```
In [6]: best_features = SelectKBest(score_func = chi2, k=10)
fit = best_features.fit(X,y)
```

```
In [7]: dfscores = pd.DataFrame(fit.scores_)
```

```
In [8]: dfcolumns = pd.DataFrame(X.columns)
```

```
In [11]: featureScores = pd.concat([dfscores, dfcolumns], axis = 1)
featureScores.columns = ['Score', 'Specs']
```

```
In [12]: featureScores
```

Out[12]:

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

```
In [13]: print(featureScores.nlargest(10, 'Score'))
```

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

## Feature Importance Using ExtraTreesClassifier

```
In [14]: from sklearn.ensemble import ExtraTreesClassifier
```

```
In [15]: best_feature = ExtraTreesClassifier()
```

```
In [16]: best_feature.fit(X,y)
```

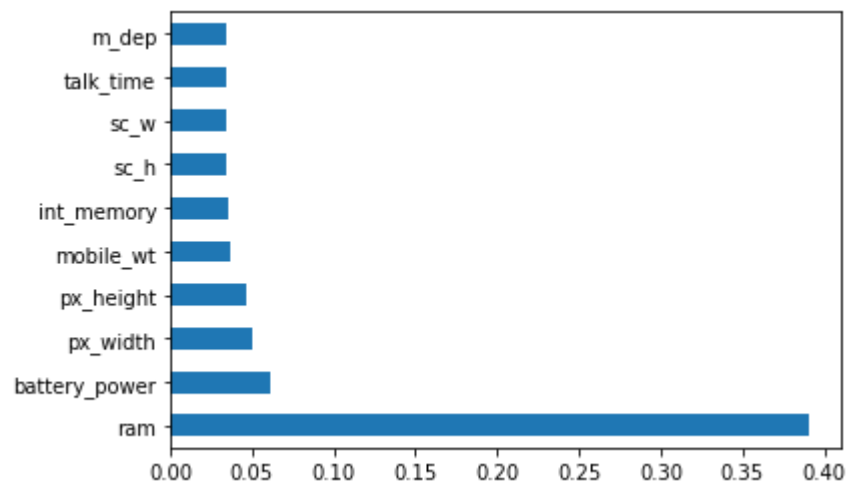
```
Out[16]: ExtraTreesClassifier()
```

```
In [17]: print(best_feature.feature_importances_)
```

```
[0.06083564 0.02023259 0.03271377 0.01942193 0.03267316 0.01724914
0.03513466 0.03411495 0.03647716 0.03372242 0.03398213 0.04597472
0.05011467 0.39069204 0.03440662 0.03437468 0.03428768 0.01430243
0.0188247  0.02046492]
```

```
In [18]: feat = pd.Series(best_feature.feature_importances_, index = X.columns)
```

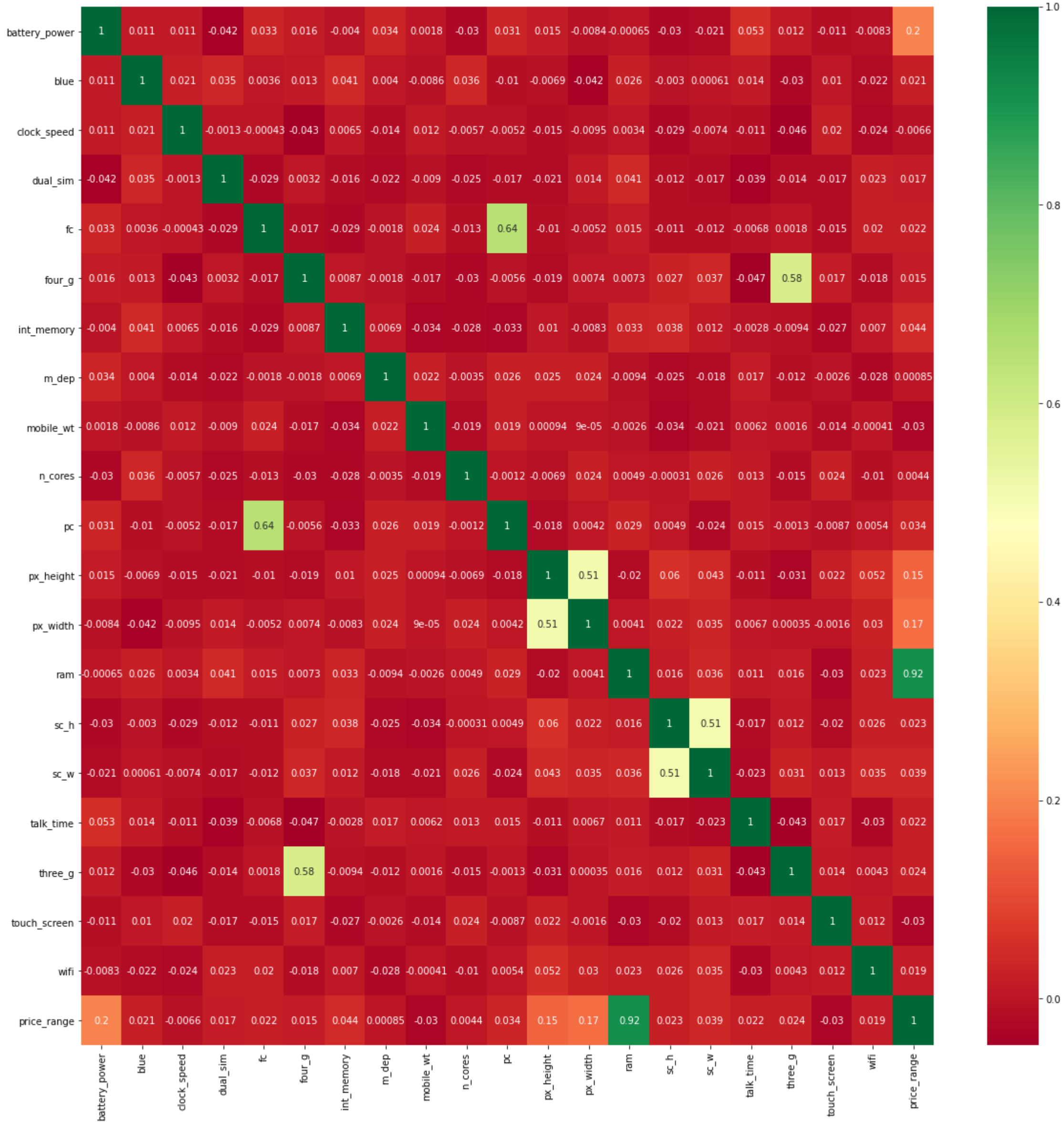
```
In [19]: feat.nlargest(10).plot(kind = 'barh')
plt.show()
```



## Confusion Matrices with Heatmap

```
In [22]: import seaborn as sns
cormat = df.corr()
plt.figure(figsize = (20,20))
top = cormat.index
g = sns.heatmap(df[top].corr(), annot = True, cmap = 'RdYlGn')
```

```
Out[22]: <AxesSubplot:~>
```



```
In [ ]:
```