

## Univariate Selection


```
import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
data = pd.read_csv("/content/drive/MyDrive/mobile_train.csv")
X = data.iloc[:,0:20] #independent columns
y = data.iloc[:, -1]  #target column i.e price range

#apply SelectKBest class to extract top 10 best features
bestfeatures = SelectKBest(score_func=chi2, k=10)
fit = bestfeatures.fit(X,y)

dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score'] #naming the dataframe columns
```

featureScores

	Specs	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	

```
print(featureScores.nlargest(10,'Score')) #print 10 best features
```

	Specs	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

## Feature Importance

—

```
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X,y)
```

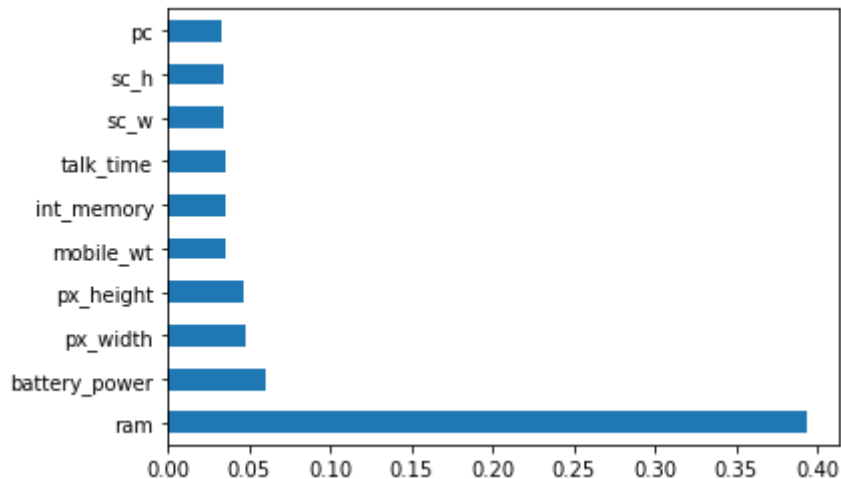
```
ExtraTreesClassifier()
```

```
print(model.feature_importances_) #use inbuilt class feature_importances of tree based cla
```

```
[0.0601292  0.02121841 0.03303608 0.01968605 0.03326206 0.01766871
 0.03546954 0.03270203 0.03576236 0.03246003 0.03347872 0.04659445
 0.04801884 0.39339916 0.03406896 0.03435857 0.03542379 0.0144744
 0.0184762  0.02031244]
```

```
#plot graph of feature importances for better visualization
```

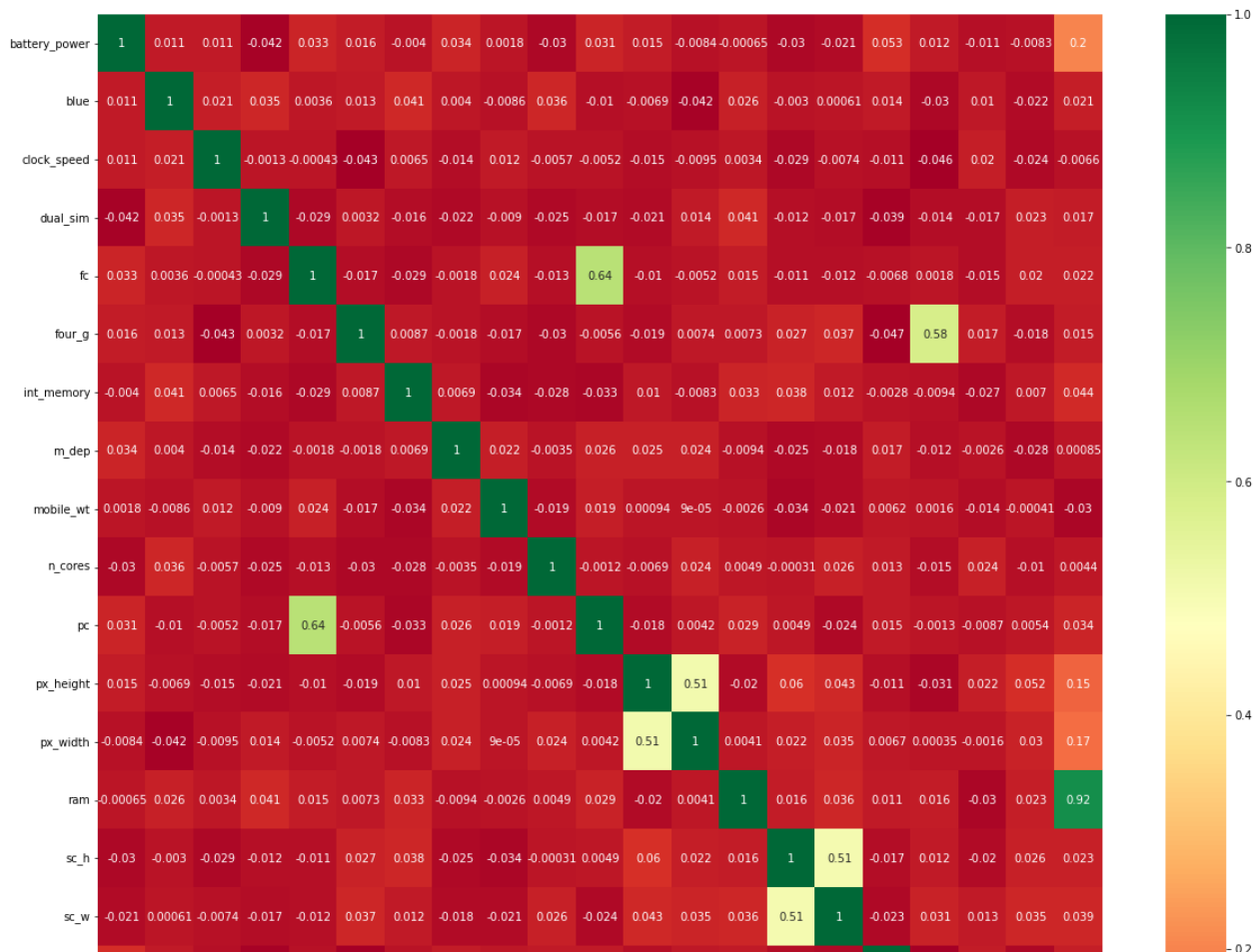
```
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```

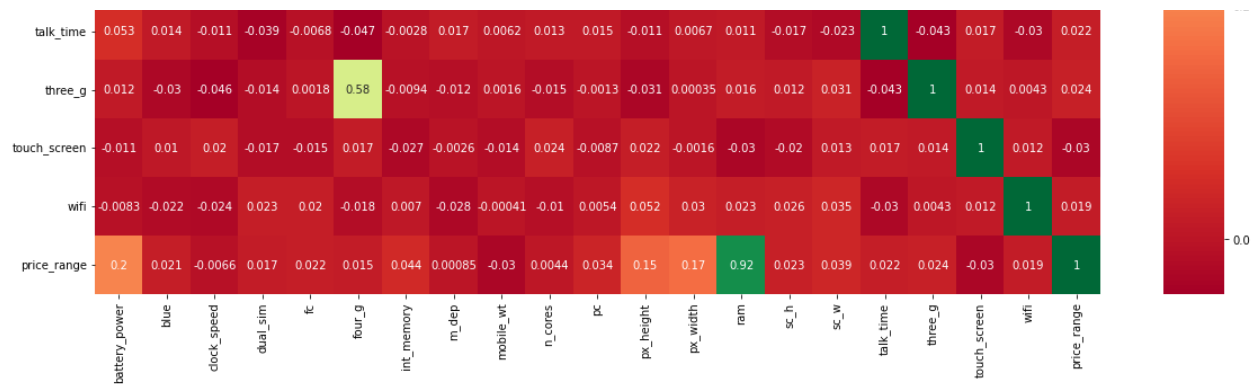


## Correlation Matrix with Heatmap

```
import seaborn as sns
#get correlations of each features in dataset
corrmat = data.corr()
```

```
top_corr_features = corrmatrix.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```







---

✓ 4s completed at 4:47 PM

