Feature Selection in Python

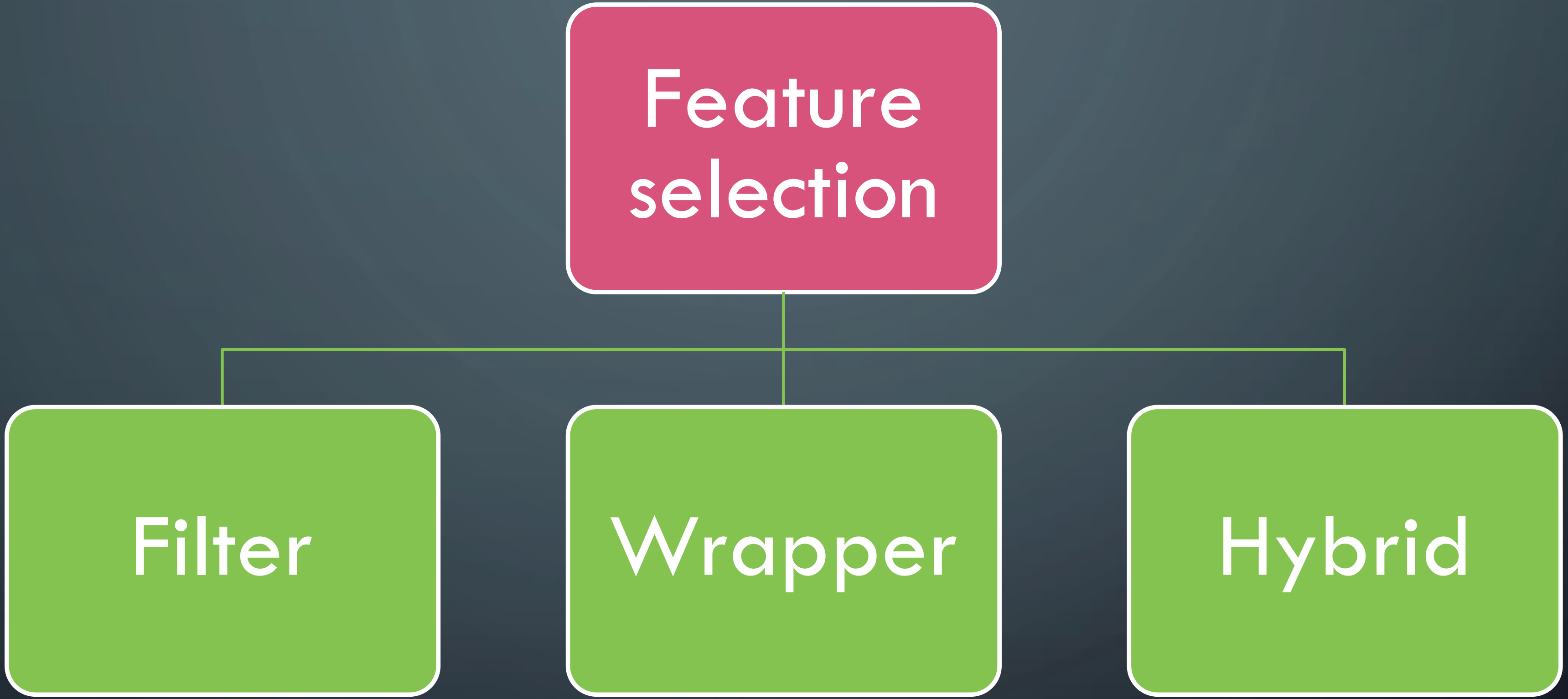
1. Poorya Mohammadi Nasab
2. Behzad Rabiei

Data Mining Course
Fall 2020

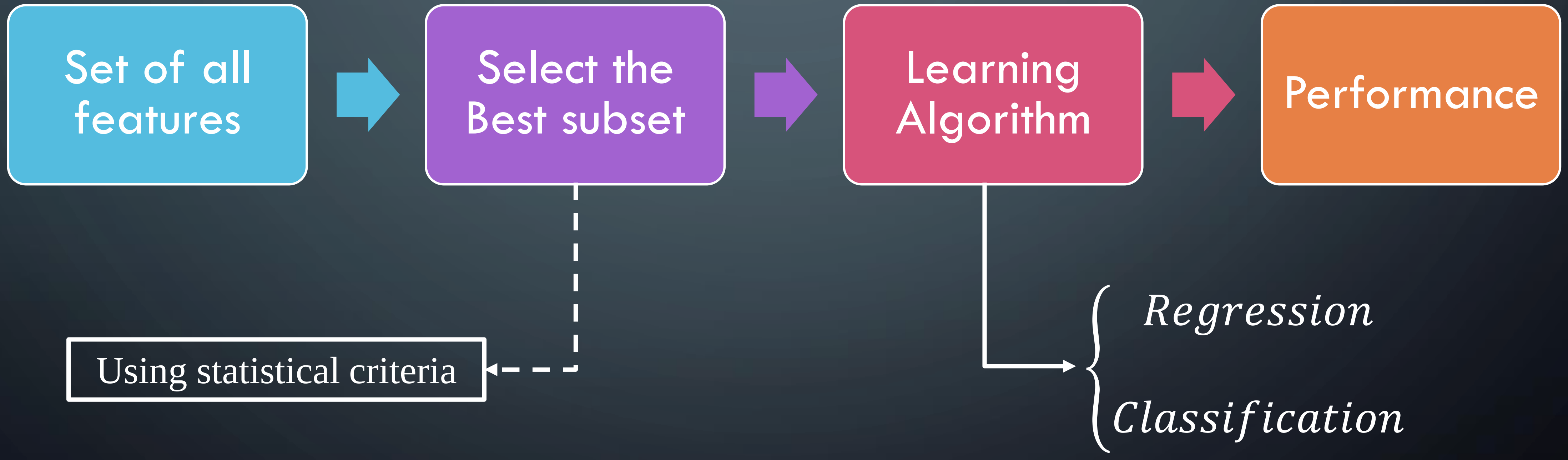
What is Feature selection

Feature selection (FS) is an important step in machine learning and has a significant impact on data mining performance. In recent years, datasets became very massive in number of attributes and instances. By feature, we mean each column in the table and dimension is the number of dataset columns and instance is the number of records. Feature selection algorithms try to remove redundant, irrelevant and noisy data to reduce the size of the dataset and increase classification accuracy.

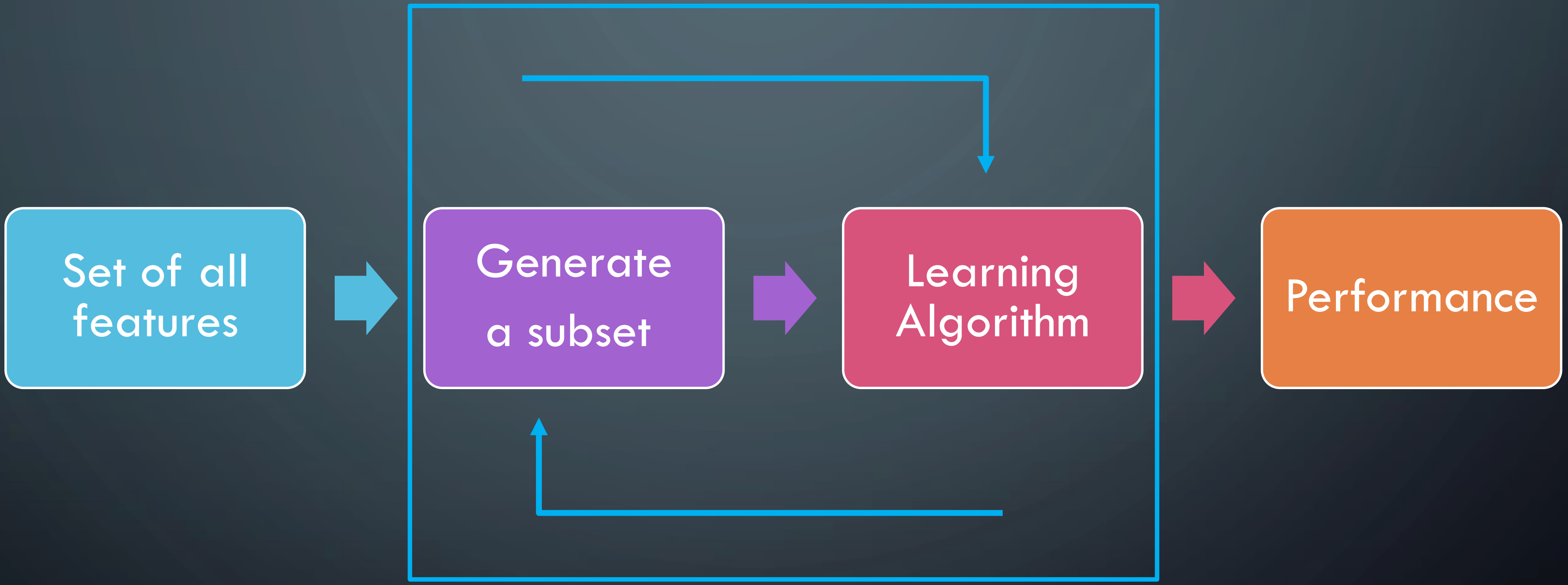
Various types of FS



Filter-Based Algorithms



Wrapper-Based Methods



Crow Search Algorithm (CSA)

CSA algorithm is a meta-heuristic algorithm proposed by Askarzadeh in 2016. The main inspiration of this algorithm came from crow search mechanism for hiding their food. They are known to be thieves, as they are stealing the food of the other birds. The four main principles of CSA are defined as follows:

1) Crows live in the flock's form
2) Crows keep in their memory their hiding places of foods
3) Each member of crow follows each other while doing thievery
4) Crows are very cautions against thievery; they protect their caches by a probability

Mobile Price Classification Dataset:

Battery Power (mAh)	*Bluetooth*	Processor Speed	*Dual Simcards*	Front Camera (mega pixels)
4G	Internal Memory (GB)	Mobile Depth (Cm)	Weight of mobile (gr)	Number of cores
Primary Camera (mega pixels)	Pixel Resolution Height	Pixel Resolution Width	RAM (MB)	Screen Height (Cm)
Screen Width (Cm)	Talk Time (h)	*3G*	*Touch Screen*	*WiFi*

Target Variable

Price range is the target variable with a value of:

- 0(low cost)
- 1(medium cost)
- 2(high cost)
- 3(very high cost)

1.Univariate Selection

Statistical tests can be used to select those features that have the strongest relationship with the output variable. The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features. In our example we use the chi-squared (chi²) statistical test for non-negative features to select 10 of the best features from the Mobile Price Range Prediction Dataset.

Using SelectKBest in Python

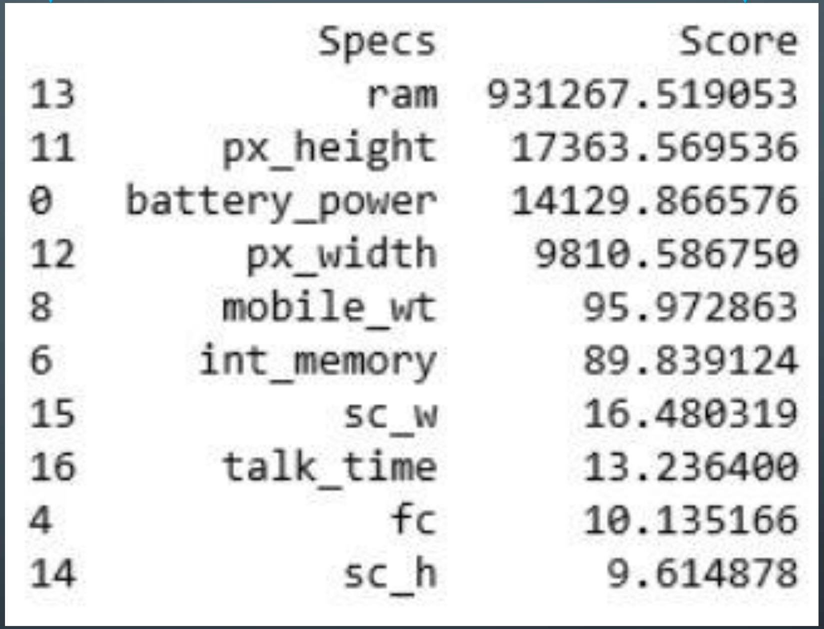
```
>>> import pandas as pd
>>> import numpy as np
>>> from sklearn.feature_selection import SelectKBest
>>> from sklearn.feature_selection import chi2

>>> data = pd.read_csv(‘train.csv’)
>>> X = data.iloc[:,0:20]   #independent columns
>>> y = data.iloc[:,-1]       #target column(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
>>> print(featureScores.nlargest(10,'Score'))  #print 10 best features
```

Output Data frame

Feature indexes

Calculated feature score using Chi2



```
             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 names

2.Feature Importance

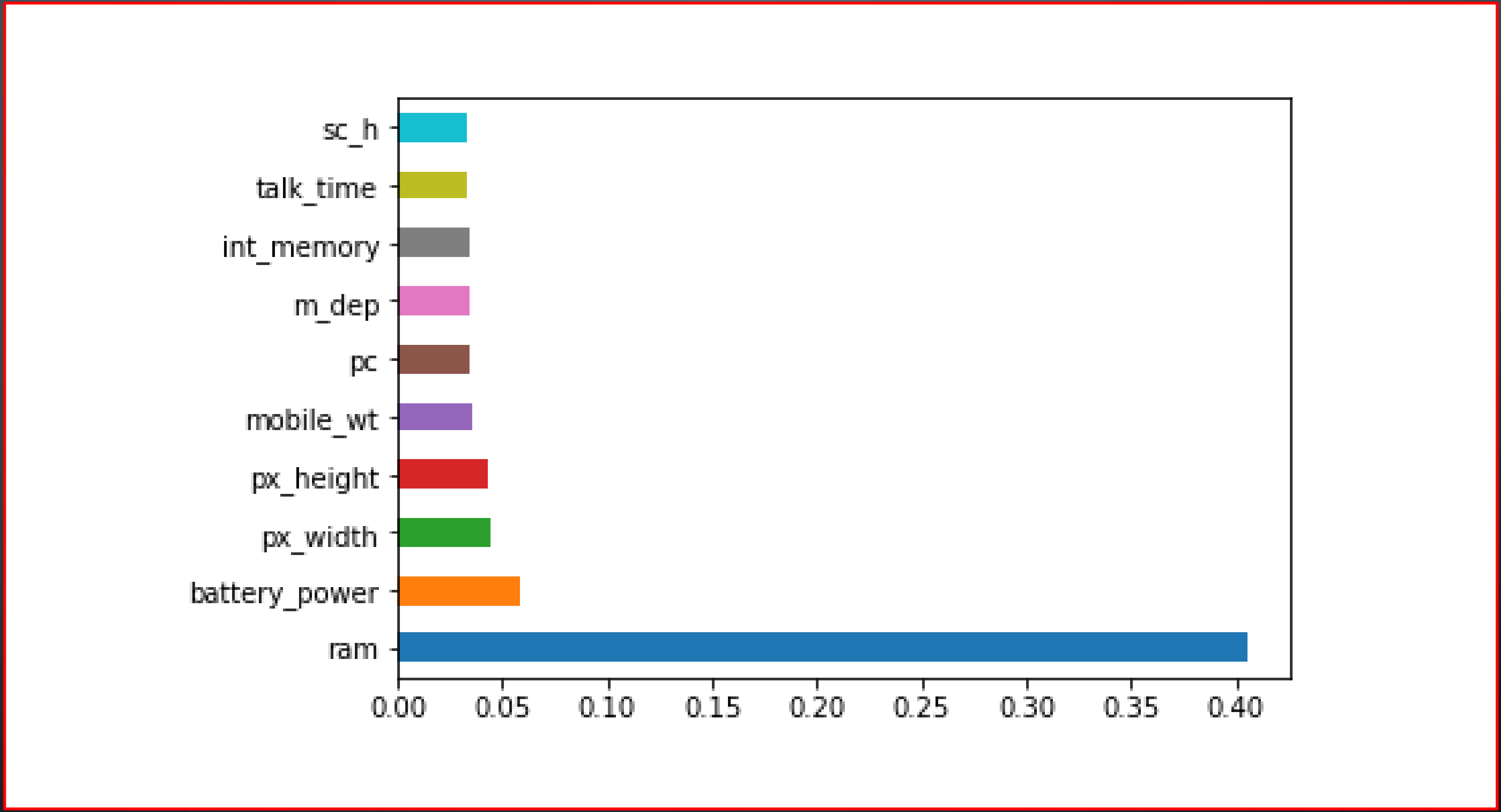
Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable. Feature importance is an inbuilt class that comes with Tree Based Classifiers, we will be using Extra Tree Classifier for extracting the top 10 features for the dataset.

Using Feature importance

```
>>> import pandas as pd
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> from sklearn.ensemble import ExtraTreesClassifier

>>> data = pd.read_csv(‘train.csv’)
>>> X = data.iloc[:,0:20]   #independent columns
>>> y = data.iloc[:,-1]      #target column(price range)
>>> model = ExtraTreesClassifier()
>>> model.fit(X,y)
>>> print(model.feature_importances_) #use inbuilt class feature_importances of tree based classifiers
    #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()
```

Plt Output



sc_h
talk_time
int_memory
m_dep
pc
mobile_wt
px_height
px_width
battery_power
ram
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40

3.Correlation Matrix with Heatmap

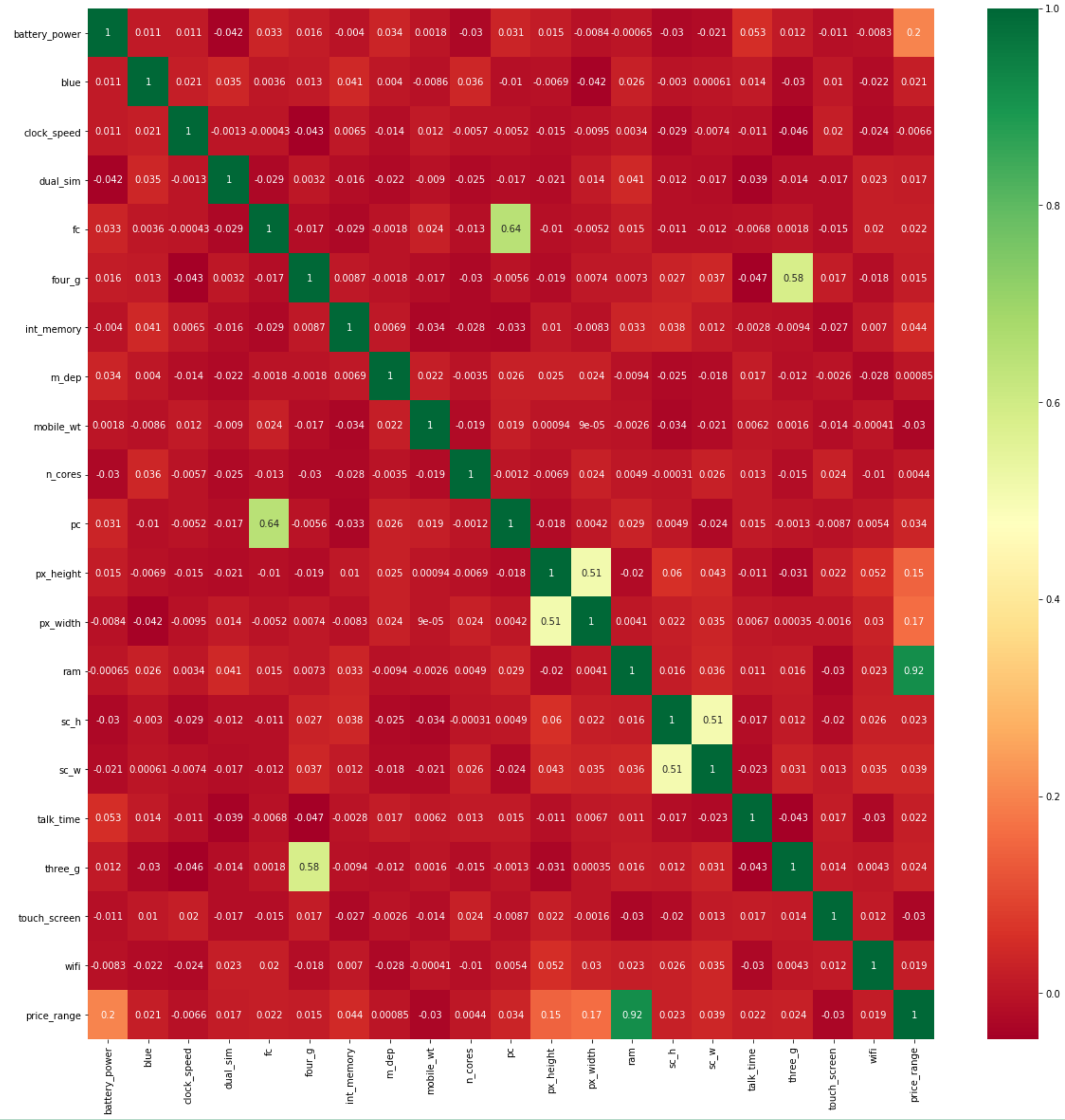
Correlation states how the features are related to each other or the target variable. Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable). Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the seaborn library.

Plotting Heatmap

```
>>> import pandas as pd
>>> import numpy as np
>>> import seaborn as sns

>>> data = pd.read_csv(‘train.csv’)
>>> X = data.iloc[:,0:20]   #independent columns
>>> y = data.iloc[:,-1]        #target column(price range)
       #get correlations of each features in dataset
>>> corrmat = data.corr()
>>> top_corr_features = corrmat.index
>>> plt.figure(figsize=(20,20))
       #plot heat map
>>> g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

Output



	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
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
px_width	-0.0084	-0.042	-0.0095	0.014	-0.0052	0.0074	-0.0083	0.024	9e-05	0.024	0.0042	0.51	1	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
three_g	0.012	-0.03	-0.046	-0.014	0.0018	0.58	-0.0094	-0.012	0.0016	-0.015	-0.0013	-0.031	0.00035	0.016	0.012	0.031	-0.043	1	0.014	0.0043	0.024
touch_screen	-0.011	0.01	0.02	-0.017	-0.015	0.017	-0.027	-0.0026	-0.014	0.024	-0.0087	0.022	-0.0016	-0.03	-0.02	0.013	0.017	0.014	1	0.012	-0.03
wifi	-0.0083	-0.022	-0.024	0.023	0.02	-0.018	0.007	-0.028	-0.00041	-0.01	0.0054	0.052	0.03	0.023	0.026	0.035	-0.03	0.0043	0.012	1	0.019
price_range	0.2	0.021	-0.0066	0.017	0.022	0.015	0.044	0.00085	-0.03	0.0044	0.034	0.15	0.17	0.92	0.023	0.039	0.022	0.024	-0.03	0.019	1