Feature selection in machine learning

Feature selection in machine learning

- Why we need feature selection
- What are the benefits of feature selection
- How to select feature for different datasets
- What are different algorithm or technique for feature selection
- Different between feature selection and extraction
- Implementation of feature selection algorithm using python
- Is feature selection is part of data analysis or model classifier
- Discussion and challenge

Why we need feature selection

- High-dimensional data analysis is a challenge for researchers and engineers in the fields of machine learning and data mining. Feature Selection provides an effective way to solve this problem by removing irrelevant and Redundant data, which can reduce Computation time, improve learning accuracy, and facilitate a better understanding for the learning model or data.
- **Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model.**
- Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in.
- Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

What are the benefits of feature selection

- **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
- **Improves Accuracy**: Less misleading data means modeling accuracy improves.
- **Reduces Training Time**: fewer data points reduce algorithm complexity and algorithms train faster.
- To avoid the **Curse Dimensionality**,
- Feature selection technique can pre-process Learning Algorithm, and good feature selection results can improve learning accuracy, reduce learning time, and simplify Learning Results.
- A good feature selection method should have high learning accuracy but less computational overhead (time complexity and space complexity).

How to select feature for different datasets

- The feature measure or evaluation criteria plays an important role in feature selection which forms the basis of feature selection
- Different feature selection for supervised, unsupervised and semi supervised algorithm
- Supervised feature selection is often oriented to classification problem, and uses the relevance or correlation between the feature and the class label as its fundamental principle. The importance of the features can be evaluated by relevance measures. For a given dataset D = (X, C), with a feature set X and class label C, the supervised model aims to find an optimal feature subset that maximizes the classification accuracy
- Unsupervised feature selection method aim to cover the natural classification of data and improve the clustering accuracy by finding a feature subset based on either clustering or evaluation criteria. Unsupervised feature selection methods could be unsupervised filter or wrapper feature selection methods, depending on whether they rely on cluster algorithm.

Cont..

- Given the dataset $D = \{D_l, D_u\}$, where D_l is the sample set with class labels, and D_u is the sample set without class labels, semi-supervised learning model uses Du to improve the learning performance of learning model trained by Dl. Semi-supervised feature selection methods, which are mainly filter models, play an important role in semi-supervised learning. Score functions are applied in most semi-supervised feature selection methods and can be divided into four categories: variance score, Laplacian score Fisher score and Constraint score.

What are different algorithm or technique for feature selection

- According to their relationship with learning methods, feature selection methods can be classified into filter, wrapper, and embedded models.
- According to the evaluation criterion, feature selection methods can be derived from correlation, Euclidean distance, consistency, dependence, and information measure.
- According to the search strategies, feature selection methods can be divided into forward increase, backward deletion, random, and hybrid models.
- According to the type of the output, feature selection methods can be divided into feature rank (weighting) and subset selection models.

Here we will discuss about main three method:

Filter Method,

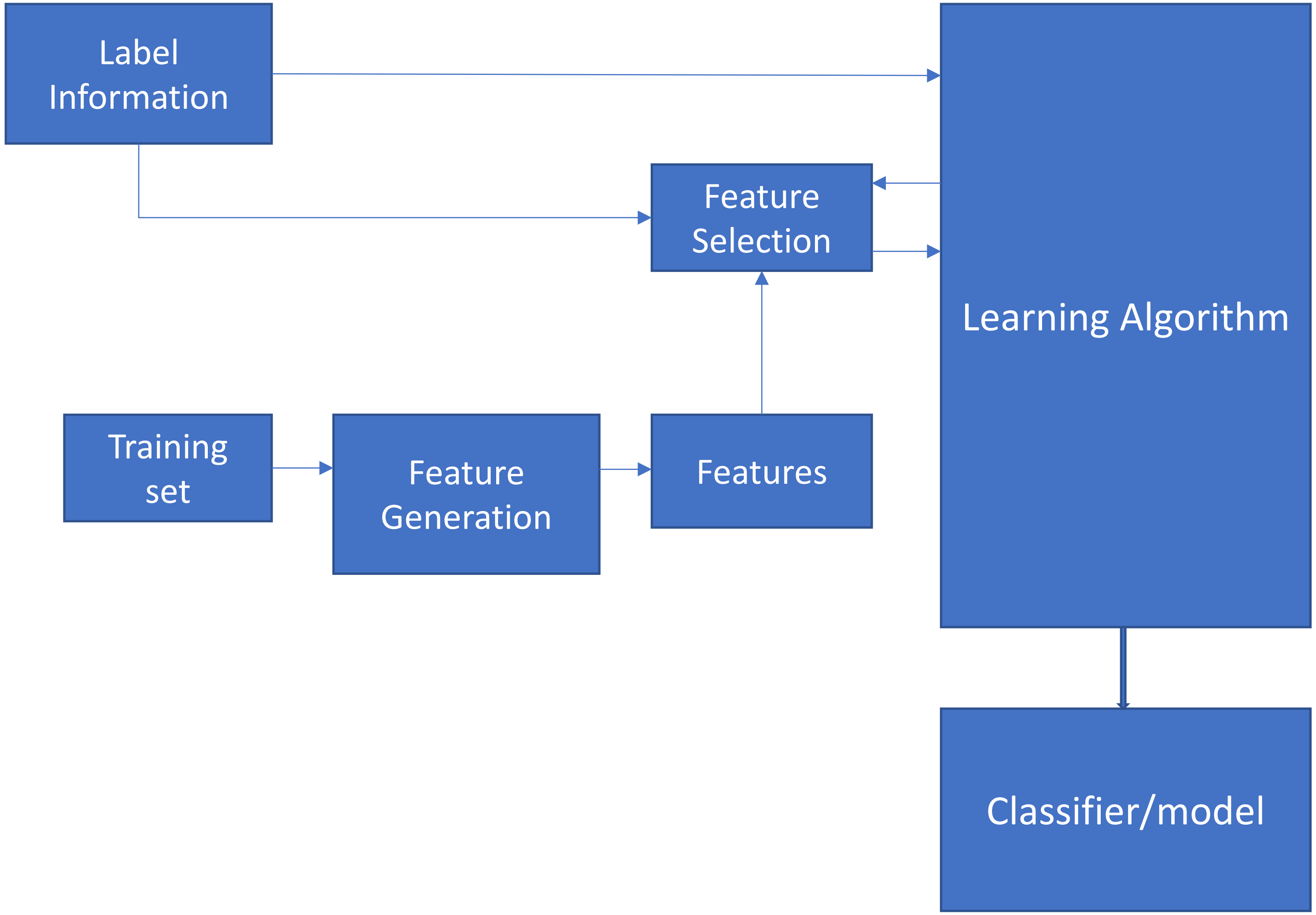
Wrapper Method

Embedded Method

Filter Method

- Filter Methods considers the relationship between features and the target variable to compute the importance of features
- Relying on the characteristics of data, filter models evaluate features without utilizing any classification algorithms [39]. A typical filter algorithm consists of two steps. In the first step, it ranks features based on certain criteria. Feature evaluation could be either univariate or multivariate
- In the univariate scheme, each feature is ranked independently of the feature space, while the multivariate scheme evaluates features in an batch way. Therefore, the multivariate scheme is naturally capable of handling redundant features. In the second step, the features with highest rankings are chosen to induce classification models.

Filter Method



A General Framework of Feature Selection for

Filter Method

- **Chi-Square Test/F-Tes**t: In general term, this method is used to test the independence of two events. If a dataset is given for two events, we can get the observed count and the expected count and this test measures how much both the counts are derivate from each other.

sklearn.feature_selection.f_regression

sklearn.feature_selection.f_classif

from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

- **Variance Threshold**: This approach of feature selection removes all features whose variance does not meet some threshold. Generally, it removes all the zero-variance features which means all the features that have the same value in all samples. Variance Threshold doesn’t consider the relationship of features with the target variable.

sklearn.feature_selection.VarianceThreshold

- **Information Gain**: Information gain or IG measures how much information a feature gives about the class. Thus, we can determine which attribute in a given set of training feature is the most meaningful for discriminating between the classes to be learned. Advantage of using mutual information over F-Test is, it does well with the non-linear relationship between feature and target variable. Sklearn offers feature selection with Mutual Information for regression and classification tasks.

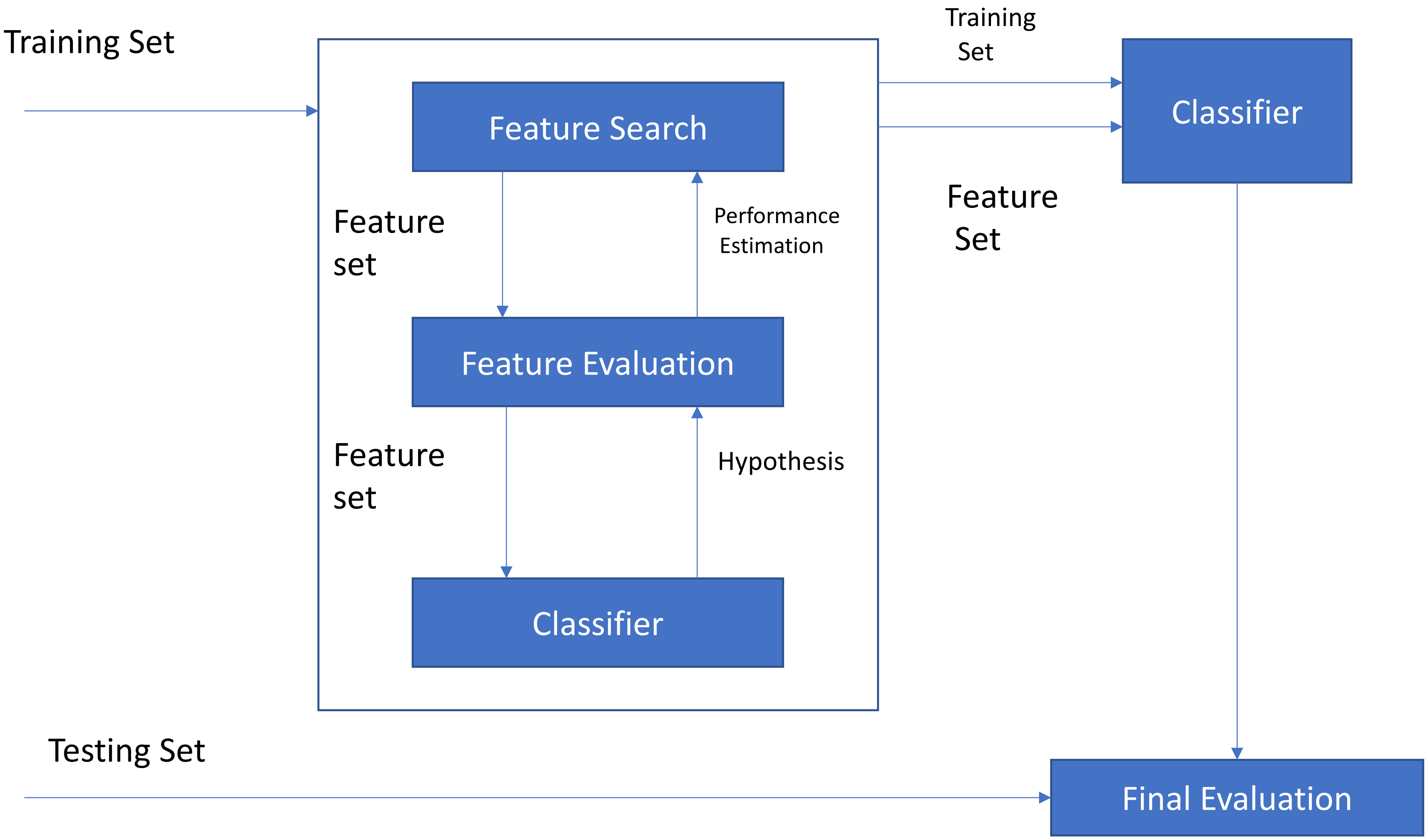
sklearn.feature_selection.mututal_info_regression
sklearn.feature_selection.mututal_info_classif

Wrapper Method

- Wrapper methods are based on greedy search algorithms as they evaluate all possible combinations of the features and select the combination that produces the best result for a specific machine learning algorithm.
- A downside to this approach is that testing all possible combinations of the features can be computationally very expensive, particularly if the feature set is very large.
- Feature selection search - how to search the subset of features from all possible feature subsets,
- Feature evaluation - how to evaluate the performance of the chosen classifier, and Induction Algorithm.

- Wrapper methods for feature selection can be divided into three categories: **Step forward feature selection**, **Step backwards feature selection** and **Exhaustive feature selection**

Wrapper Method



General Framework for Wrapper Methods of Feature Selection

Wrapper Method

- In the first phase of the step forward feature selection, the performance of the classifier is evaluated with respect to each feature. The feature that performs the best is selected out of all the features.
- In the second step, the first feature is tried in combination with all the other features. The combination of two features that yield the best algorithm performance is selected. The process continues until the specified number of features are selected.

from mlxtend.feature_selection import SequentialFeatureSelector

- Step backwards feature selection, as the name suggests is the exact opposite of step forward feature selection that we studied in the last section. In the first step of the step backwards feature selection, one feature is removed in round-robin fashion from the feature set and the performance of the classifier is evaluated.
- The feature set that yields the best performance is retained. In the second step, again one feature is removed in a round-robin fashion and the performance of all the combination of features except the 2 features is evaluated. This process continues until the specified number of features remain in the dataset.

from mlxtend.feature_selection import SequentialFeatureSelector

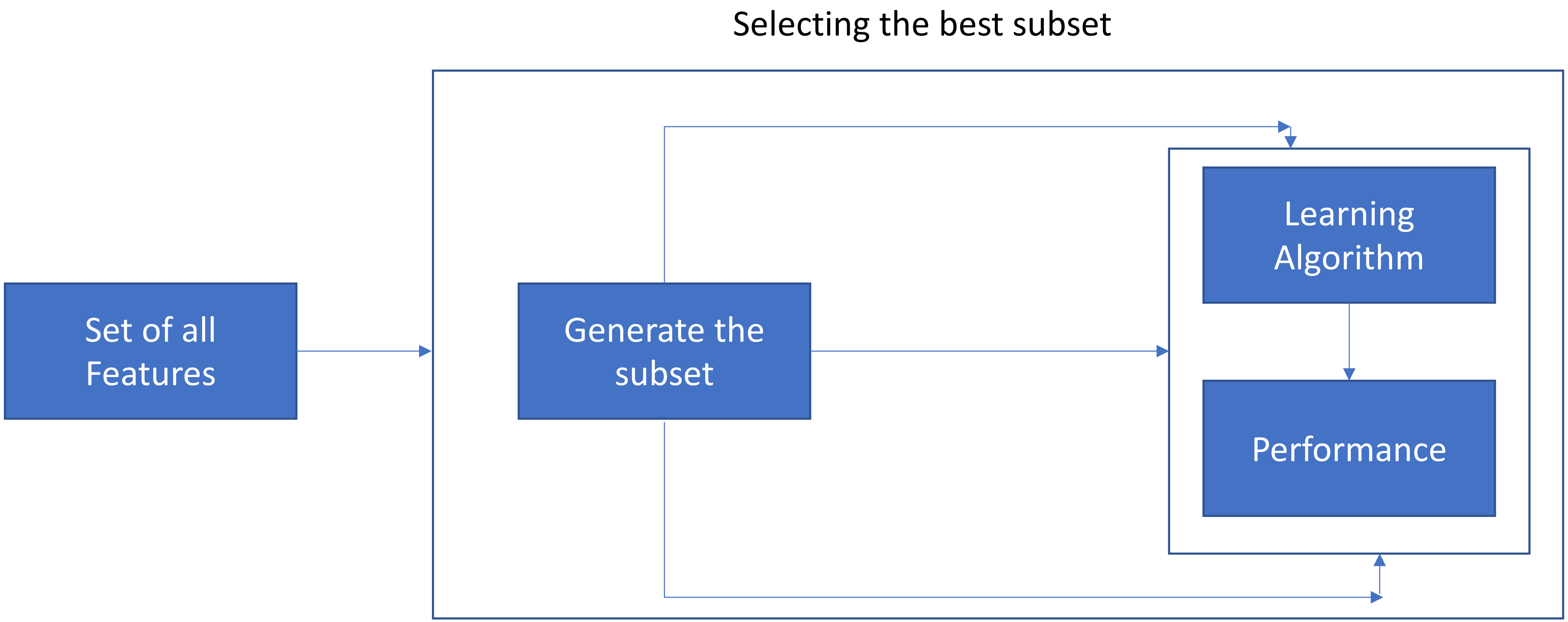
- In exhaustive feature selection, the performance of a machine learning algorithm is evaluated against all possible combinations of the features in the dataset. The feature subset that yields best performance is selected. The exhaustive search algorithm is the most greedy algorithm of all the wrapper methods since it tries all the combination of features and selects the best.
- A downside to exhaustive feature selection is that it can be slower compared to step forward and step backward method since it evaluates all feature combinations.

from mlxtend.feature_selection import ExhaustiveFeatureSelector

Embedded Method

- Embedded Models embedding feature selection with classifier construction, have the advantages of wrapper models - they include the interaction with the classification model and filter models - they are far less computationally intensive than wrapper methods.
- There are three types of embedded methods. The first are pruning methods that first utilizing all features to train a model and then attempt to eliminate some features by setting the corresponding coefficients to 0, while maintaining model performance such as recursive feature elimination using support vector machine .The second are models with a build-in mechanism for feature selection such as ID3 and C4.5. The third are regularization models with objective functions that minimize fitting errors and in the mean time force the coefficients to be small or to be exact zero. Features with coefficients that are close to 0 are then eliminated.

Embedded Method



General Framework for Embedded Methods of Feature Selection

Embedded Method

- L1 Regularisation Technique such as LASSO: Least Absolute Shrinkage and Selection Operator (LASSO) is a linear model which estimates sparse coefficients and is useful in some contexts due to its tendency to prefer solutions with fewer parameter values.
- Ridge Regression (L2 Regularisation): The L2 Regularization is also known as Ridge Regression or Tikhonov Regularization which solves a regression model where the loss function is the linear least squares function and regularization.
- Elastic Net: This linear regression model is trained with L1 and L2 as regularize which allows for learning a sparse model where few of the weights are non-zero like Lasso and on the other hand maintaining the regularization properties of Ridge.

Different between feature selection and extraction

- Both Feature extraction and feature selection are capable of improving learning performance, lowering computational complexity, building better generalizable models, and decreasing required storage.
- Feature extraction maps the original feature space to a new feature space with lower dimensions by combining the original feature space. It is difficult to link the features from original feature space to new features. Therefore further analysis of new features is problematic since there is no physical meaning for the transformed features obtained from feature extraction techniques.
- While feature selection selects a subset of features from the original feature set without any transformation, and maintains the physical meanings of the original features. In this sense, feature selection is superior in terms of better readability and interpretability. This property has its significance in many practical applications such as finding relevant genes to a specific disease and building a sentiment lexicon for sentiment analysis.

Cont..

- Typically feature selection and feature extraction are presented separately. Via sparse learning such as l1 regularization, feature extraction (transformation) methods can be converted into feature selection methods
- Dimensionality reduction is one of the most popular techniques to remove noisy (i.e. irrelevant) and redundant features.
- Dimensionality reduction techniques can be categorized mainly into feature extraction and feature selection. Feature extraction approaches project features into a new feature space with lower dimensionality and the new constructed features are usually combinations of original features.
- Examples of feature extraction techniques include Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Canonical Correlation Analysis (CCA). On the other hand, the feature selection approaches aim to select a small subset of features that minimize redundancy and maximize relevance to the target such as the class labels in classification.
- Representative feature selection techniques include Information Gain, Relief, Fisher Score and Lasso.

Implementation of feature selection algorithm using python

Here you will see three different method using python for feature selection.

1. Univariate Selection

2. Feature Importance

3.Correlation Matrix with Heatmap

First take a look on dataset and to apply the technique. You can download dataset from https://www.kaggle.com/iabhishekofficial/mobile-price-classification#train.csv

Description of variables in the above file

Cont..

Description of variables in the above file

battery_powerTotal energy a battery can store in one time measured in mAh

blueHas bluetooth or not

clock_speedspeed at which microprocessor executes instructions

dual_simHas dual sim support or not

fcFront Camera mega pixels

four_gHas 4G or not

int_memoryInternal Memory in Gigabytes

m_depMobile Depth in cm

mobile_wtWeight of mobile phone

n_coresNumber of cores of processor

pcPrimary Camera mega pixels

px_heightPixel Resolution Height

px_widthPixel Resolution Width

ramRandom Access Memory in Mega Bytes

sc_hScreen Height of mobile in cm

sc_wScreen Width of mobile in cm

Cont..

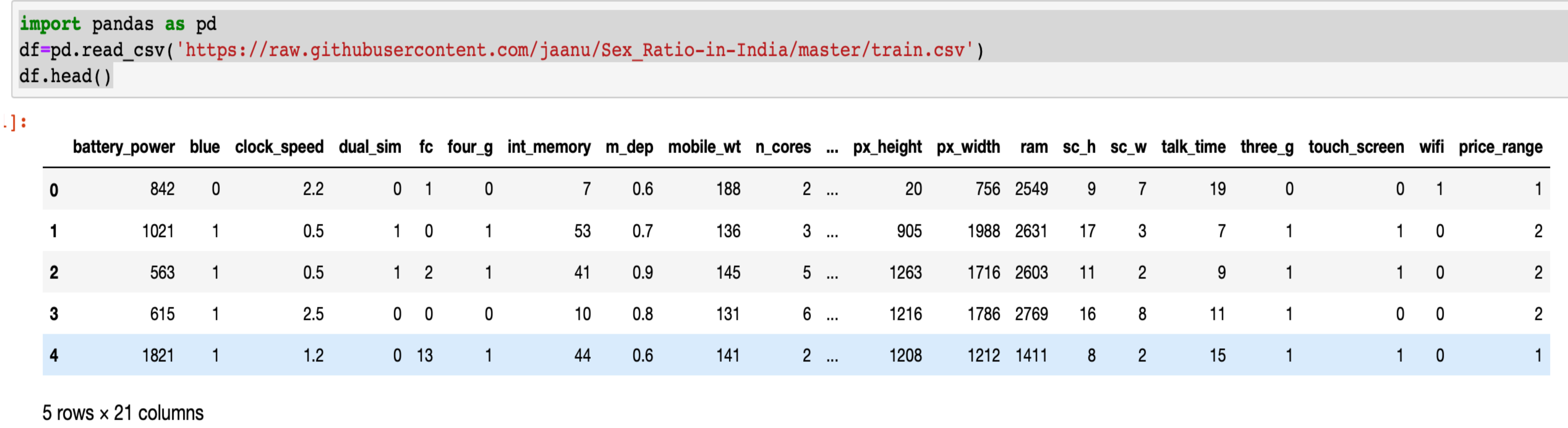
talk_timelongest time that a single battery charge will last when you are

three_gHas 3G or not

touch_screenHas touch screen or not

wifiHas wifi or not

price_rangeThis is the target variable with value of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost).



```
import pandas as pd
df=pd.read_csv('https://raw.githubusercontent.com/jaanu/Sex_Ratio-in-India/master/train.csv')
df.head()
```

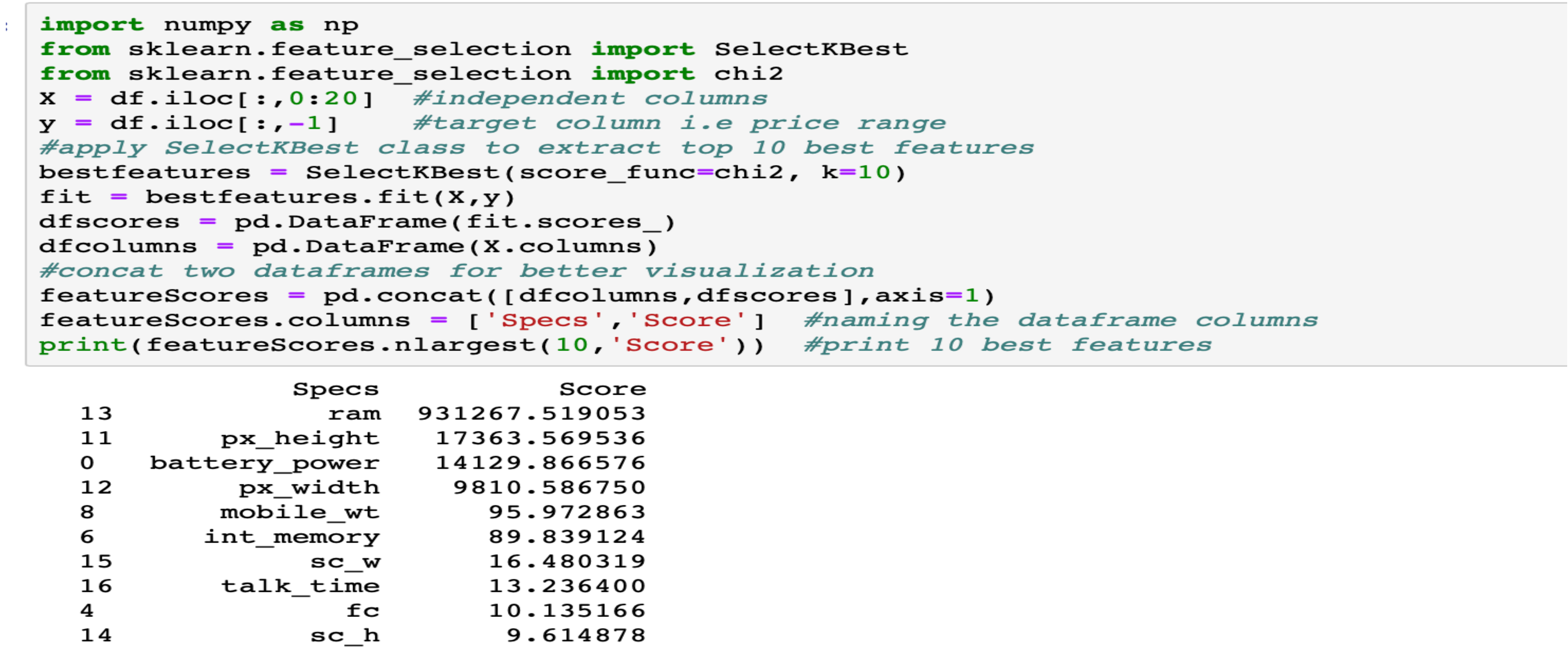
.]:

	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

Implementation of feature selection algorithm using python

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. uses the chi-squared (chi^2) statistical test for non-negative features to select 10 of the best features from the Mobile Price Range Prediction Dataset.

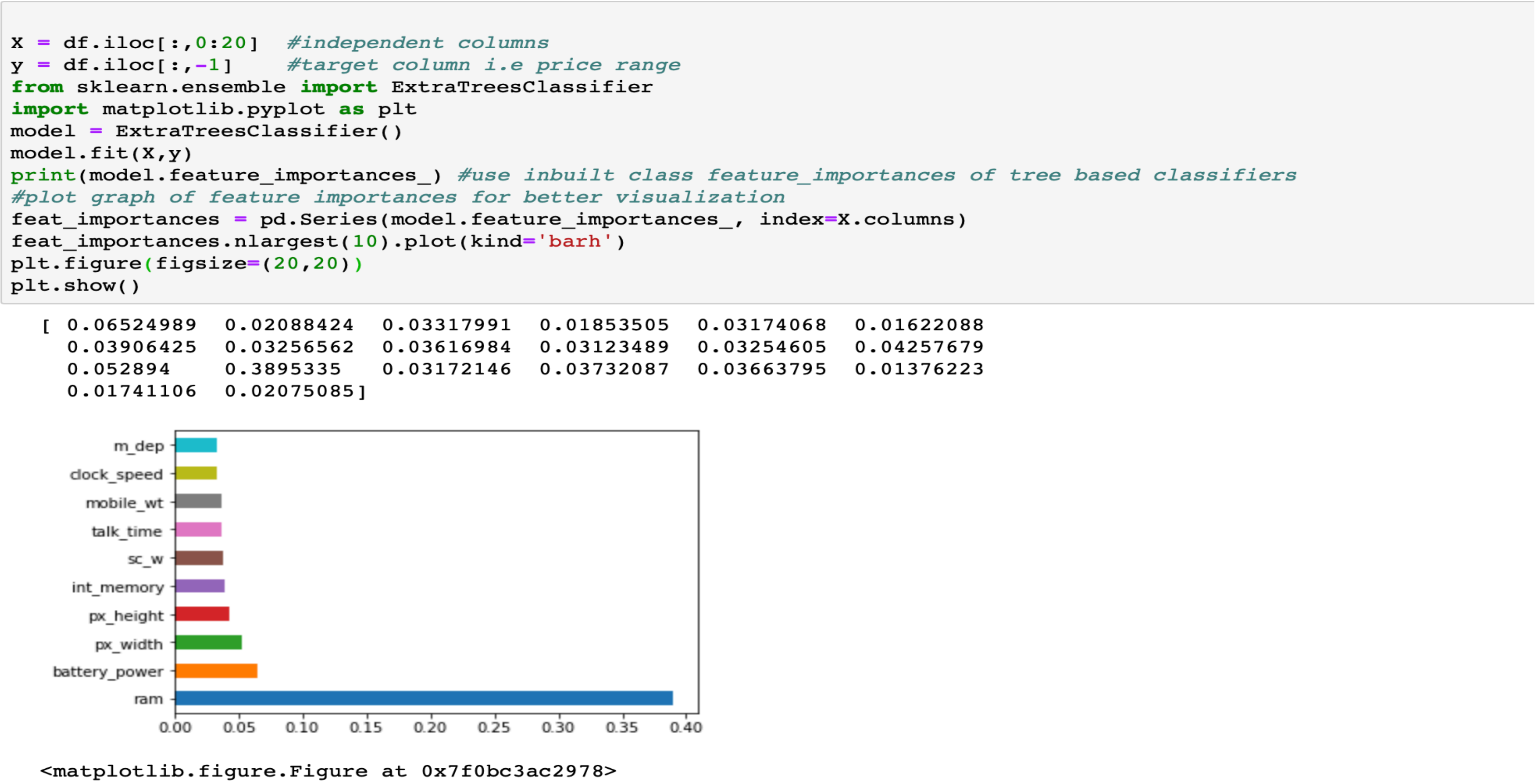


```
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
X = df.iloc[:,0:20]   #independent columns
y = df.iloc[:,-1]     #target column i.e price range
#apply SelectKBest class to extract top 10 best features
bestfeatures = SelectKBest(score_func=chi2, 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
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

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.



```
X = df.iloc[:,0:20]  #independent columns
y = df.iloc[:,-1]    #target column i.e price range
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
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.figure(figsize=(20,20))
plt.show()
```

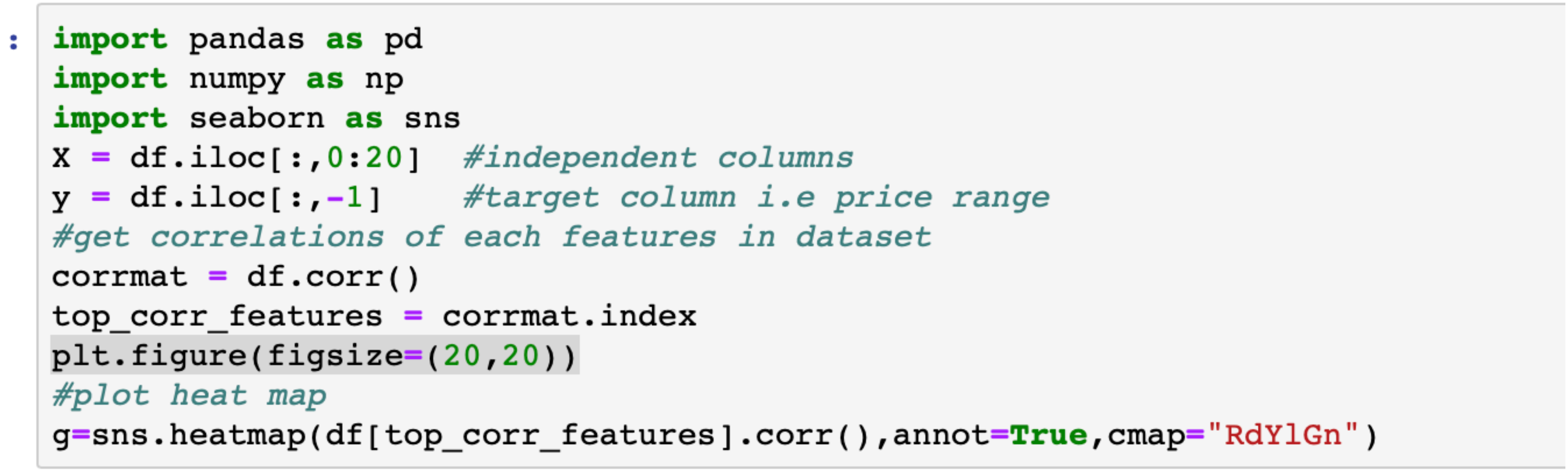
```
[ 0.06524989  0.02088424  0.03317991  0.01853505  0.03174068  0.01622088
  0.03906425  0.03256562  0.03616984  0.03123489  0.03254605  0.04257679
  0.052894    0.3895335   0.03172146  0.03732087  0.03663795  0.01376223
  0.01741106  0.02075085]
```

```
<matplotlib.figure.Figure at 0x7f0bc3ac2978>
```

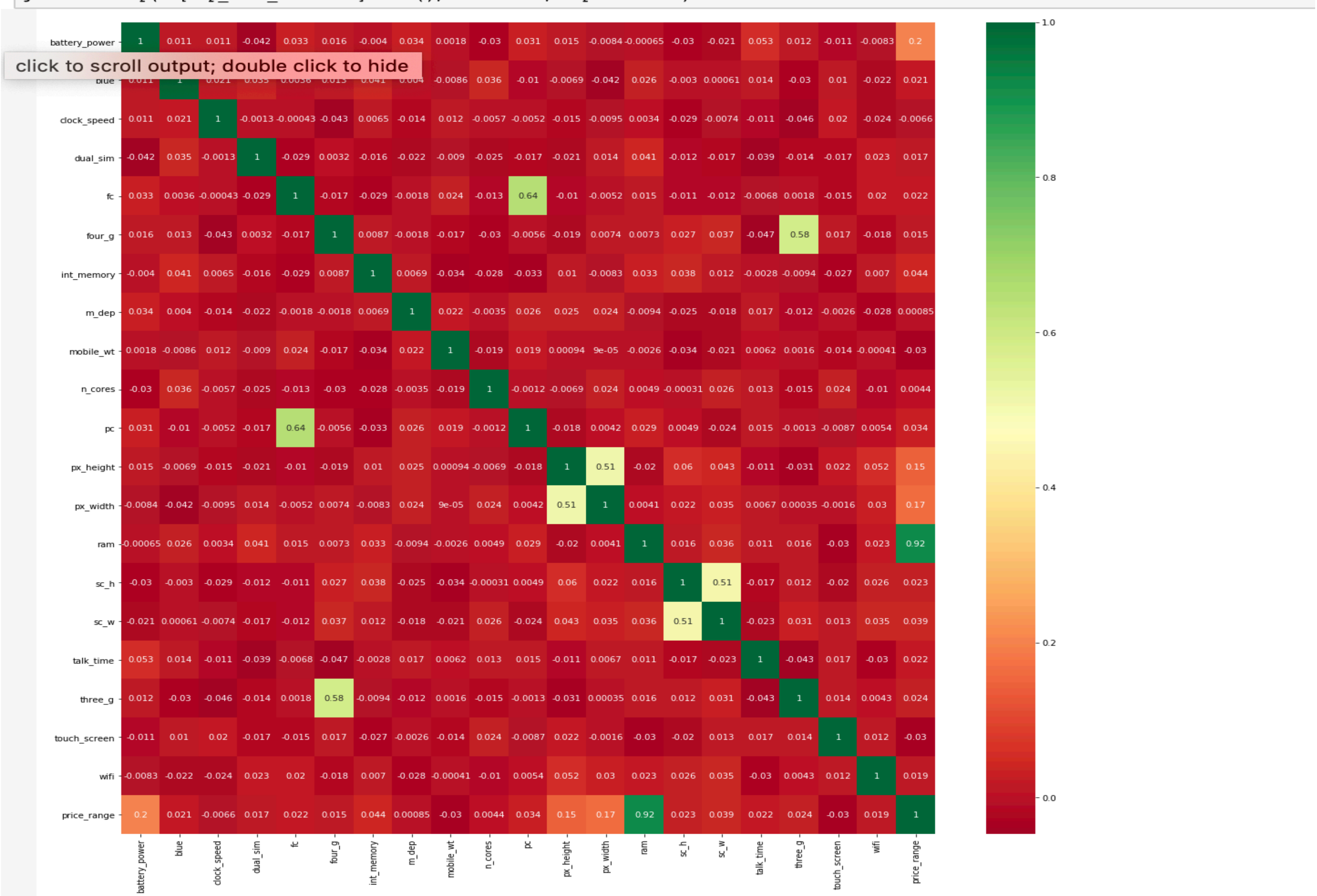
with Heatmap

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.



```
import pandas as pd
import numpy as np
import seaborn as sns
X = df.iloc[:,0:20]  #independent columns
y = df.iloc[:,-1]    #target column i.e price range
#get correlations of each features in dataset
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



click to scroll output; double click to hide

	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

Is feature selection is part of data analysis or model classifier

It is important to consider feature selection a part of the model selection process. If you do not, you may inadvertently introduce bias into your models which can result in overfitting.

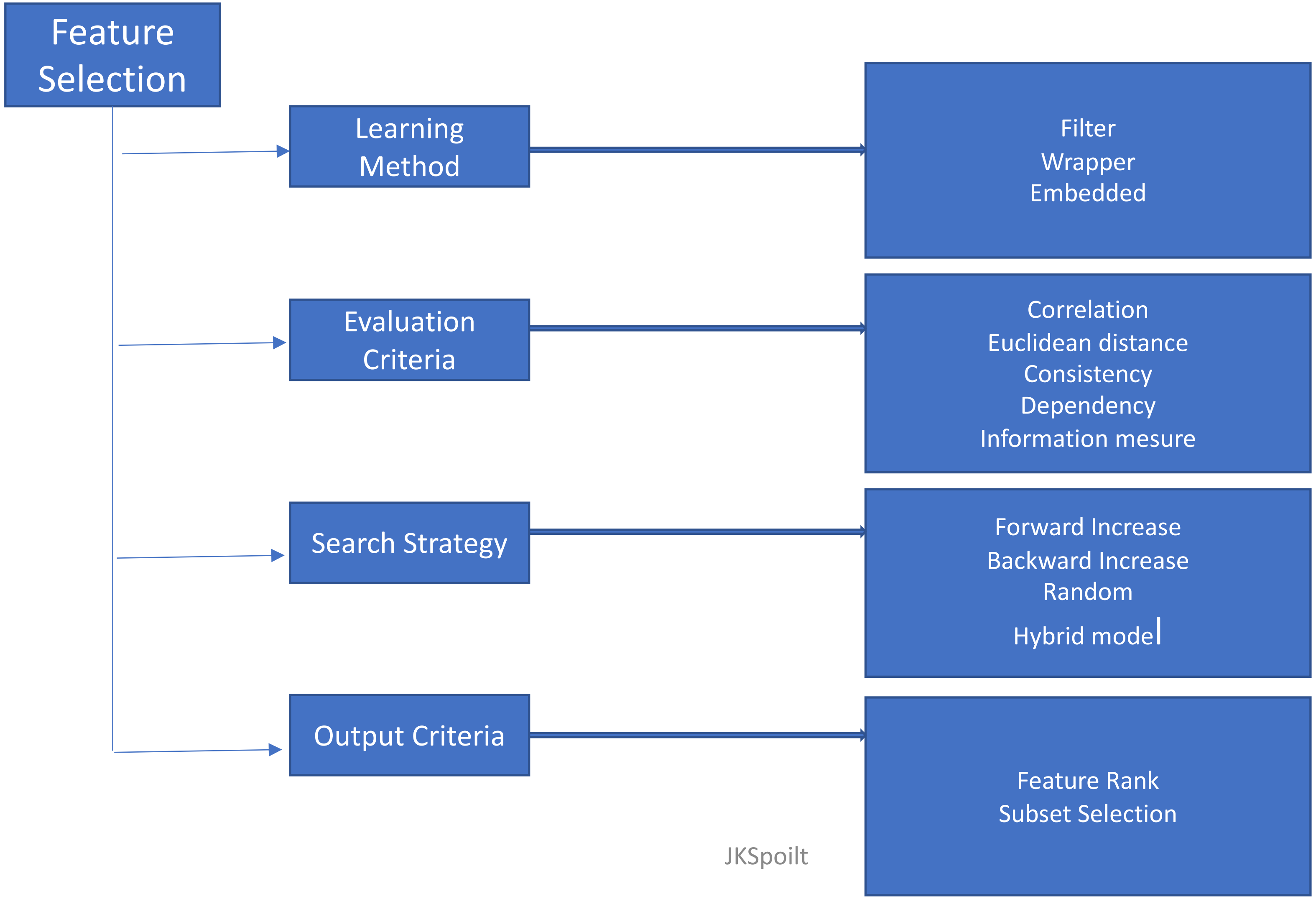
you must include feature selection within the inner-loop when you are using accuracy estimation methods such as cross-validation. This means that feature selection is performed on the prepared fold right before the model is trained. A mistake would be to perform feature selection first to prepare your data, then perform model selection and training on the selected features.

Challenge

- Filter models select features independent of any specific classifiers. However the major disadvantage of the filter approach is that it totally ignores the effects of the selected feature subset on the performance of the induction algorithm
- Wrapper models obtain better predictive accuracy estimates than filter models. However, wrapper models are very computationally expensive compared to filter models. It produces better performance for the predefined classifier since we aim to select features that maximize the quality therefore the selected subset of features is inevitably biased to the predefined classifier.

Conclusion

Feature Selection is a sub-topic of Feature Engineering



Feature Selection

Learning Method

Filter
Wrapper
Embedded

Evaluation Criteria

Correlation
Euclidean distance
Consistency
Dependency
Information mesure

Search Strategy

Forward Increase
Backward Increase
Random
Hybrid model

Output Criteria

Feature Rank
Subset Selection

Thank

☺ You