Feature Selection:

Top reasons to use feature selection are:

- *It enables the machine learning algorithm to train faster.*
- *It reduces the complexity of a model and makes it easier to interpret.*
- *It improves the accuracy of a model if the right subset is chosen.*
- *It reduces overfitting.*

Different Methods for Feature Selection:

1] ***Filter Method****:*

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods.

Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

- ***Pearson's Correlation:*** *A Pearson's correlation is used when you want to find a linear relationship between two variables. Its value varies from -1 to +1. Pearson's correlation is given as:*

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- ***LDA:*** *Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.*

- ***ANOVA:*** *ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.*
-
- ***Chi-Square:*** *It is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.*

  One thing that should be kept in mind is that filter methods do not remove multicollinearity. So, you must deal with multicollinearity of features as well before training models for your data.

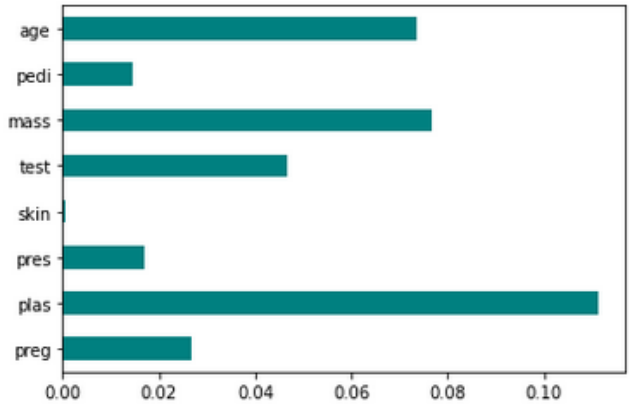
Some Additional Filter Methods:

1] Information Gain

Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.

```
from sklearn.feature_selection import mutual_info_classif
import matplotlib.pyplot as plt
%matplotlib inline

importances = mutual_info_classif(X, Y)
feat_importances = pd.Series(importances, dataframe.columns[0:len(dataframe.columns)-1])
feat_importances.plot(kind='barh', color = 'teal')
plt.show()
```



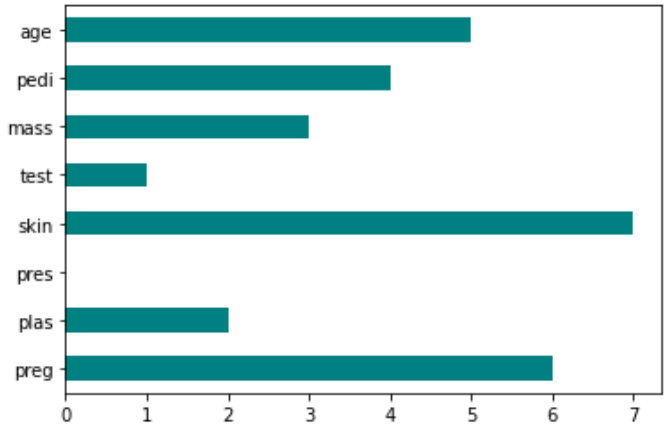
2] Fisher's Score

Fisher score is one of the most widely used supervised feature selection methods. The algorithm which we will use returns the ranks of the variables based on the fisher's score in descending order. We can then select the variables as per the case.

```
from skfeature.function.similarity_based import fisher_score
import matplotlib.pyplot as plt
%matplotlib inline

# Calculating scores
ranks = fisher_score.fisher_score(X, Y)

# Plotting the ranks
feat_importances = pd.Series(ranks, dataframe.columns[0:len(dataframe.columns)-1])
feat_importances.plot(kind='barh', color = 'teal')
plt.show()
```



3] Variance Threshold

The variance threshold is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e., features that have the same value in all samples. We assume that features with a higher variance may contain more useful information, but note that we are not taking the relationship between feature variables or feature and target variables into account, which is one of the drawbacks of filter methods.

```
from sklearn.feature_selection import VarianceThreshold

# Resetting the value of X to make it non-categorical
X = array[:,0:8]

v_threshold = VarianceThreshold(threshold=0)
v_threshold.fit(X)  # fit finds the features with zero variance
v_threshold.get_support()
```

```
array([ True,  True,  True,  True,  True,  True,  True,  True])
```

2] *Wrapper Method:*

In wrapper methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset. The problem is essentially reduced to a search problem. These methods are usually computationally very expensive.

Some common examples of wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc.

- ***Forward Selection:*** *Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.*
- ***Backward Elimination:*** *In backward elimination, we start with all the features and remove the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.*
- ***Recursive Feature elimination:*** *It is a greedy optimization algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination.*

One of the best ways for implementing feature selection with wrapper methods is to use the Boruta package that finds the importance of a feature by creating shadow features.

It works in the following steps:

1. *Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).*
2. *Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.*
3. *At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z-score than the maximum Z-score of its shadow features) and constantly removes features which are deemed highly unimportant.*
4. *Finally, the algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs.*

4. Embedded Methods:

Embedded methods combine the qualities' of filter and wrapper methods. It's implemented by algorithms that have their own built-in feature selection methods.

Some of the most popular examples of these methods are LASSO and RIDGE regression which have inbuilt penalization functions to reduce overfitting.

- *Lasso regression performs L1 regularization which adds a penalty equivalent to the absolute value of the magnitude of coefficients.*
- *Ridge regression performs L2 regularization which adds a penalty equivalent to the square of the magnitude of coefficients.*

5. Difference between Filter and Wrapper methods

1. *The main differences between the filter and wrapper methods for feature selection are:*
2. *Filter methods measure the relevance of features by their correlation with dependent variables while wrapper methods measure the usefulness of a subset of features by actually training a model on it.*

3. *Filter methods are much faster compared to wrapper methods as they do not involve training the models. On the other hand, wrapper methods are computationally very expensive as well.*
4. *Filter methods use statistical methods for evaluation of a subset of features while wrapper methods use cross validation.*
5. *Filter methods might fail to find the best subset of features in many occasions but wrapper methods can always provide the best subset of features.*
6. *Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using a subset of features from the filter methods.*

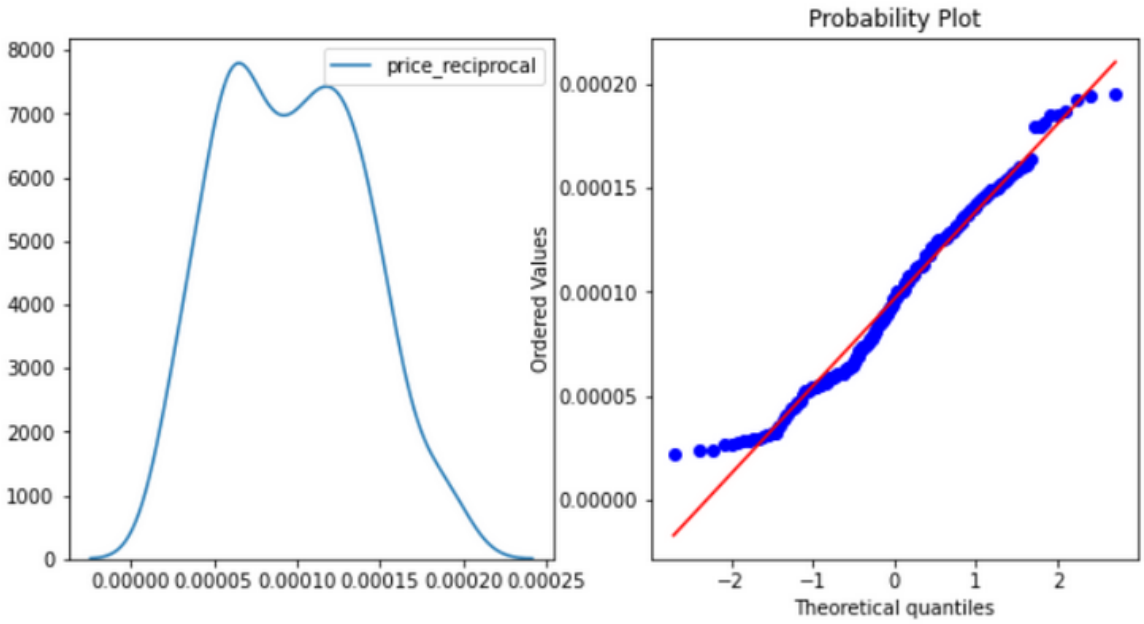
Feature Transformation:

We Use Feature Transformation when our data is not normally distributed or data is positive or negative skew.

We can check data distribution using Histogram or kde plot or Q-Q plot AKA probability plot.

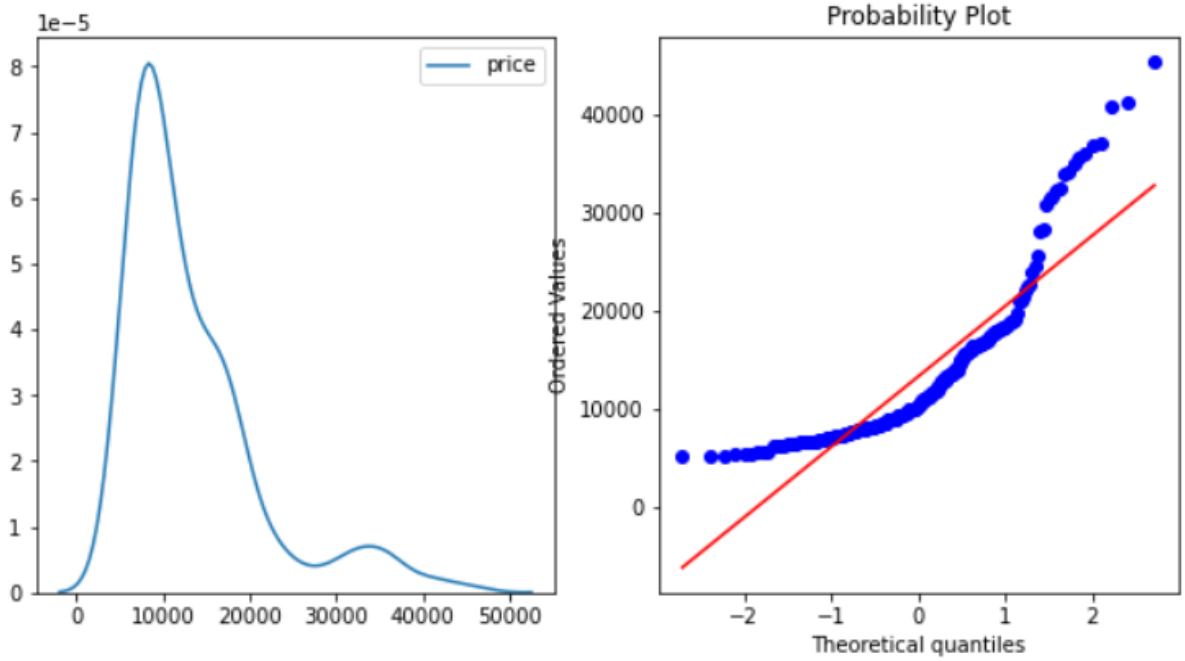
Check Normally DIstributed Data using kde and QQ plot:

A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, we should see the points forming a roughly straight line.




```
#function to return plots for the feature
def normality(data,feature):
    plt.figure(figsize=(10,5))
    plt.subplot(1,2,1)
    sns.kdeplot(data[feature])
    plt.subplot(1,2,2)
    stats.probplot(data[feature],plot=pylab)
    plt.show()
```

Check Not Normally DIstributed Data using kde and QQ plot:



Feature Transformation Methods:

1] <u>Logarithmic Transformation:</u>

This will convert the Price value to its log value i.e log(Price)

ex.- *cp['price_log']=np.log(cp['price'])*

2] <u>Reciprocal Transformation</u> :

This will inverse values of Price i.e1/Price

3] <u>Square Root Transformation</u> :

This transformation will take the square root of the Price column i.e sqrt(Price).

4] <u>Exponential Transformation</u>:

The exponential value of the Price variable will be taken.

5] <u>Box-Cox Transformation</u> –:

The Box-Cox transformation is defined as:

Ex- *cp['price_Boxcox'],parameters=stats.boxcox(cp['price'])*

6]QuantileTransformer:

This method transforms the features to follow a uniform or a normal distribution.

Detecting and Treating Outliers:

What is Outlier:

Outliers means something unusual in comparison to the others in a group.
Similarly, an Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.

Why do they occur?

An outlier may occur due to the variability in the data, or due to experimental error/human error.

They may indicate an experimental error or heavy skewness in the data(heavy-tailed distribution)

What do they affect?

In statistics, we have three measures of ***central tendency*** *namely Mean, Median, and Mode. They help us describe the data.*

Mean is the accurate measure to describe the data when we do not have any outliers present.

Median is used if there is an outlier in the dataset.

Mode is used if there is an outlier AND about ½ or more of the data is the same.

'Mean' is the only measure of central tendency that is affected by the outliers which in turn impacts Standard deviation.

Detecting Outliers:

If our dataset is small, we can detect the outlier by just looking at the dataset. But what if we have a huge dataset, how do we identify the outliers then? We need to use visualization and mathematical techniques.

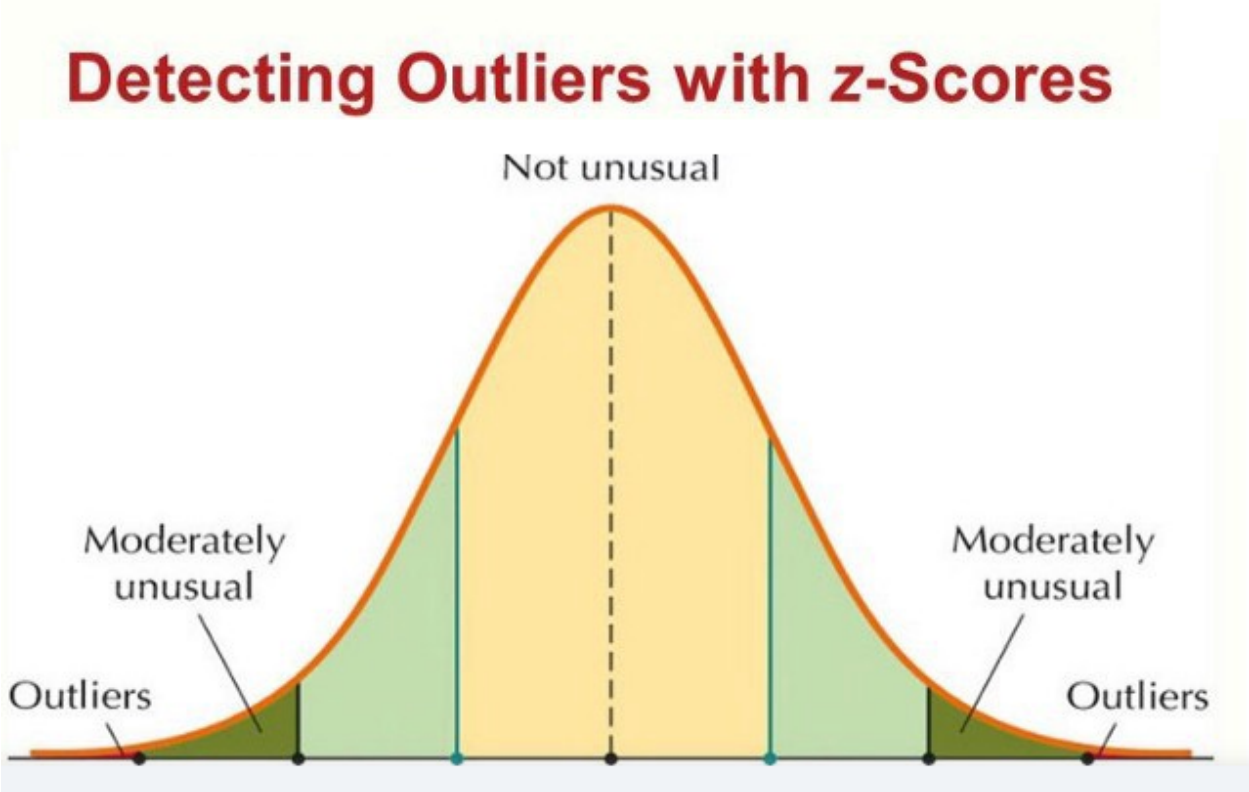
Below are some of the techniques of detecting outliers

- *Boxplots*
- *Z-score*
- *Inter Quantile Range(IQR)*

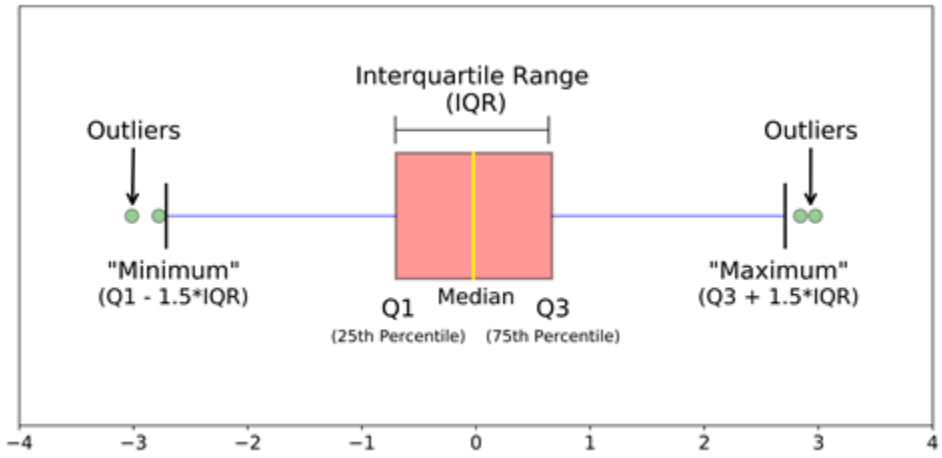
Detecting outliers using the Z-scores

Criteria: *any data point whose Z-score falls out of 3rd standard deviation is an outlier.*

define a threshold value of 3 and mark the data points whose absolute value of Z-score is greater than the threshold as outliers.



Detecting outliers using the Interquartile Range(IQR)



IQR to detect outliers

Criteria: *data points that lie 1.5 times of IQR above Q3 and below Q1 are outliers.*

- *Sort the dataset in ascending order*
- *calculate the 1st and 3rd quartiles(Q1, Q3)*
- *compute IQR=Q3-Q1*
- *compute lower bound = (Q1–1.5*IQR), upper bound = (Q3+1.5*IQR)*
- *loop through the values of the dataset and check for those who fall below the lower bound and above the upper bound and mark them as outliers*

5. Handling Outliers

Till now we learned about detecting the outliers. The main question is WHAT do we do with the outliers?

Below are some of the methods of treating the outliers

- *Trimming/removing the outlier*
- *Quantile based flooring and capping*
- *Mean/Median imputation*

1] Trimming/Remove the outliers

In this technique, we remove the outliers from the dataset. Although it is not a good practice to follow.

2] Quantile based flooring and capping(Winsorization)

In this technique, the outlier is capped at a certain value above the 90th percentile value or floored at a factor below the 10th percentile value.

***New array:** [15, 20.7, 18, 7.2, 13, 16, 11, 20.7, 7.2, 15, 10, 9]*

The data points that are lesser than the 10th percentile are replaced with the 10th percentile value and the data points that are greater than the 90th percentile are replaced with 90th percentile value.

3] Median imputation:

As the mean value is highly influenced by the outliers, it is advised to replace the outliers with the median value

***4] QuantileTransformer:**This method transforms the features to follow a uniform or a normal distribution. When data is transformed normally outlier automatically comes in that distribution.*

***5] Discretization:** In this technique, by making the groups we include the outliers in a particular group and force them to behave in the same manner as those of other points in that group. This technique is also known as **Binning**.*

There are also some algorithms to detect and impute Anomaly also called as Anomaly Detection Algorithms:

1] Isolation Forest

2] OneClassSVM

3] Gaussian Mixture Model