**WINE QUALITY PREDICTION**

**Overview**

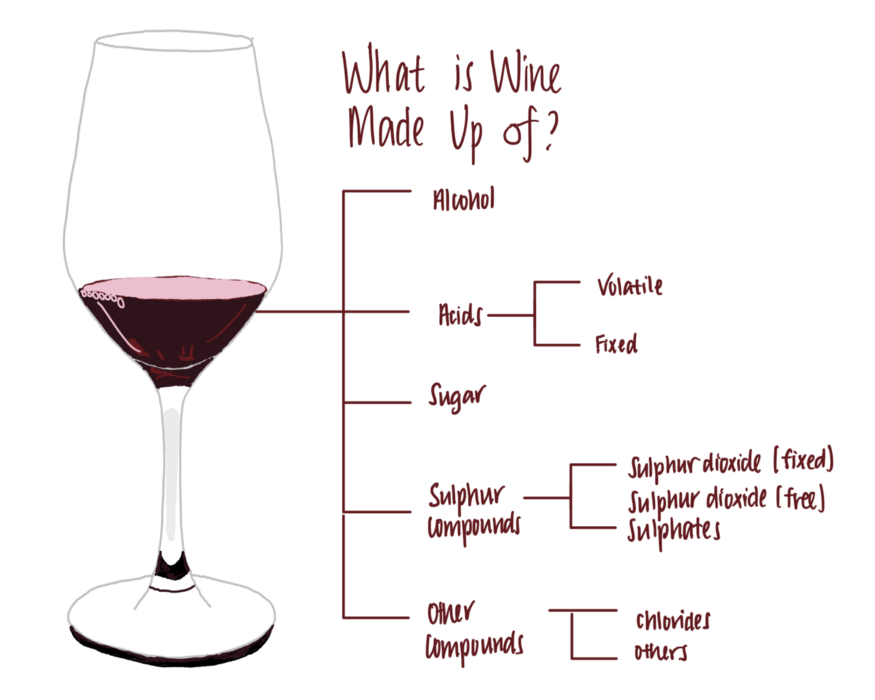
* **Basics understanding of Wine.**
* **Data description**
* **Importing modules**
* **Study dataset**
* **Handle null values**
* **VISUALIZATION**
* **Standardisation**
* **Applying model**
* **Save model**
* **Endnote**

# **Introduction**

As the quarantine continues, I’ve picked up a number of hobbies and interests… including WINE. Recently, I’ve acquired a taste for wines, although I don’t really know what makes a good wine. Therefore, I decided to apply some machine learning models to figure out what makes a good quality wine!

For this project, I used red wine quality dataset to build various classification models to predict whether a particular red wine is “good quality” or not. Each wine in this dataset is given a “quality” score between 0 and 10. For the purpose of this project, I converted the output to a binary output where each wine is either “good quality” (a score of 7 or higher) or not (a score below 7). The quality of a wine is determined by 11 input variables:

* **volatile acidity:**   Volatile acidity*is the*gaseous acids present in wine.
* **fixed acidity:**Primary fixed acids found in wine are tartaric, succinic, citric, and malic
* **residual sugar:**Amount of sugar left after fermentation.
* **citric acid:** It is weak organic acid, found in citrus fruits naturally.
* **chlorides:**Amount of salt present in wine.
* **free sulphur dioxide:**   So2 is used for prevention of wine by oxidation and microbial spoilage.
* **total sulphur dioxide**
* **pH:**In wine pH is used for checking acidity
* **density**
* **sulphates:**    Added sulphates preserve freshness and protect wine from oxidation, and bacteria.
* **alcohol:**   Percent of alcohol present in wine.



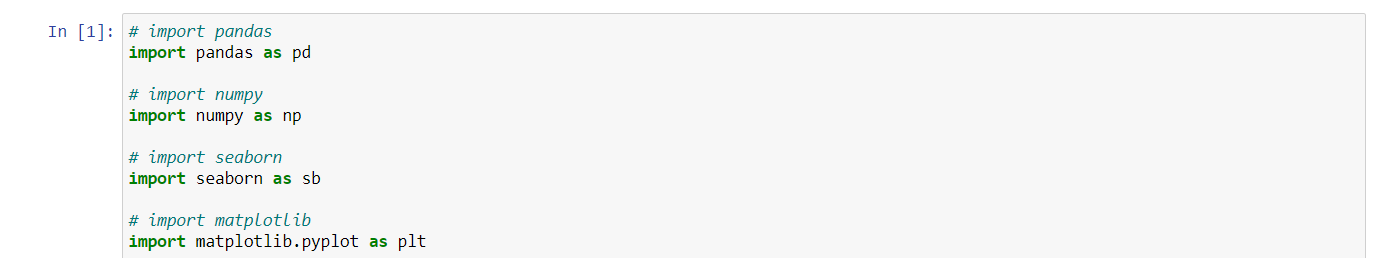
## Objectives

The objectives of this project are as follows

1. To experiment with different classification methods to see which yields the highest accuracy.
2. To determine which features are the most indicative of a good quality win.

### **Importing modules:**

Let’s import some basic library:



**NUMPY:**   
NumPy can be used to perform a wide variety of mathematical operations onarrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

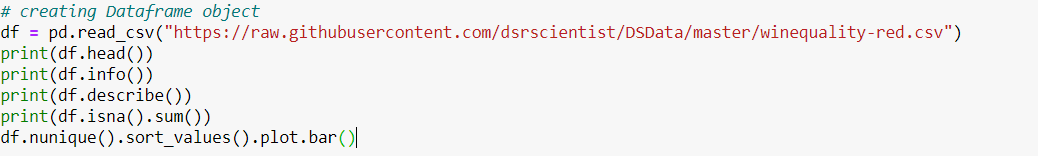
**PANDAS:**

Pandas is mainly used for data analysis**.**

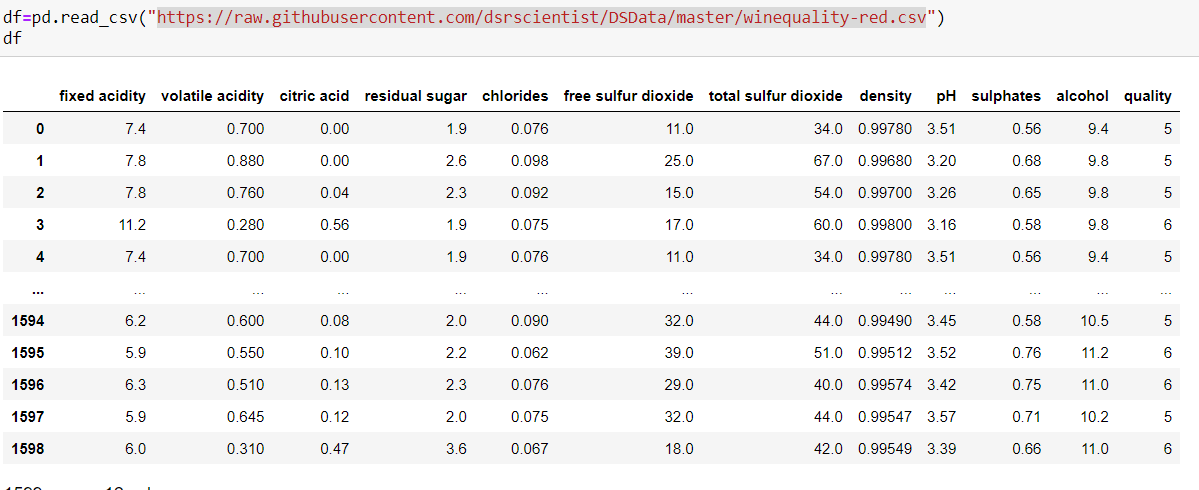
**SEABORN & MATPLOTLIB:**

Both the library is used for visualisation.

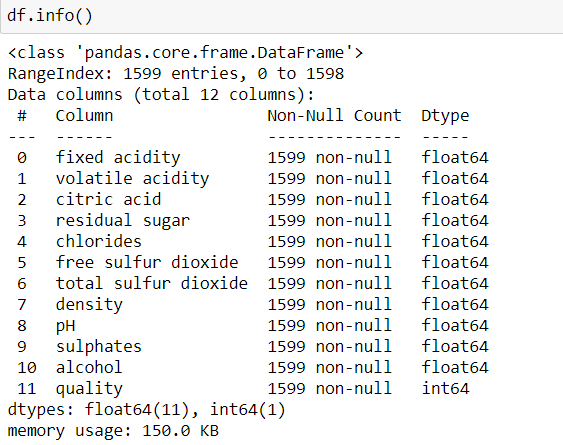
### **Study dataset:**



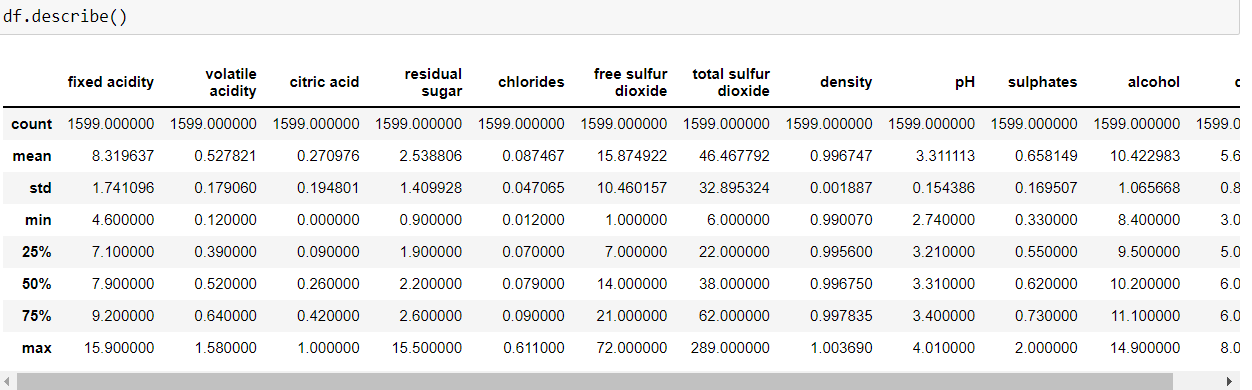
**OUTPUT :**

****

**INFORMATON:**

****

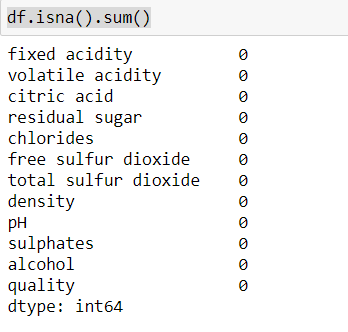
**DESCRIBE:**

****

From info we get the info about the datatype of respected columns which helps us in visualization to predict which graph is best fit for columns.

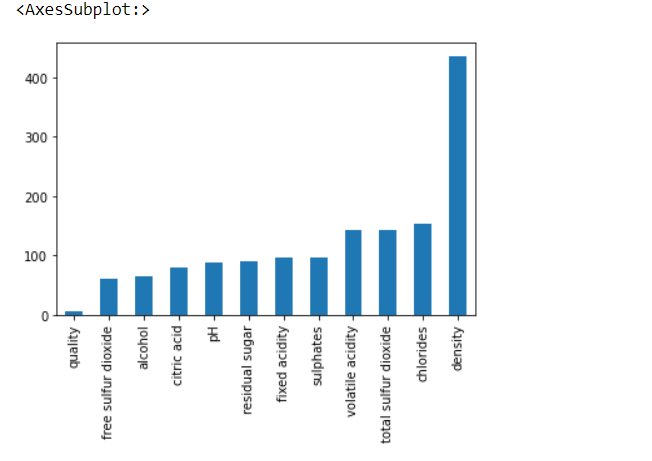
It describes the distribution of your data: 50 should be a value that describes the middle of the data, also known as median. 25, 75 is the border of the upper/lower quarter of the data. You can get an idea of how skew your data is. Note that the mean is higher than the median, which means your data is right skewed.

**NAN VALUE:**

****

**There is no nan value in the dataset.**

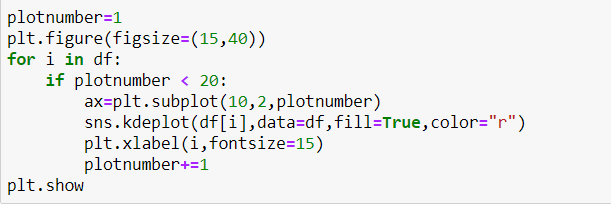
**UNIQUE VALUE IN COLUMN:**

****

**This plot gives info about the unique in dataset.**

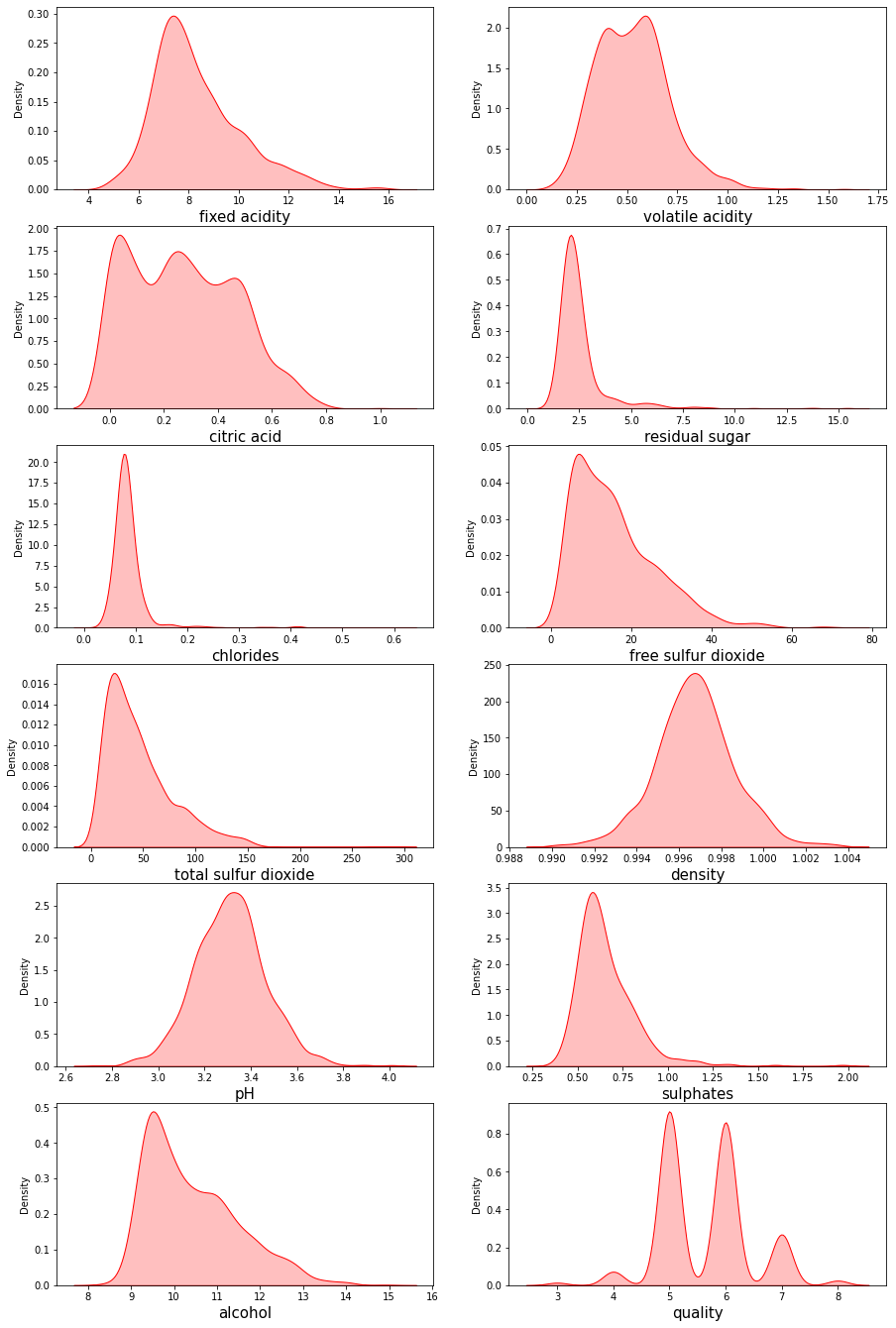
**VISUALIZATION:**

We know that the “image speaks everything” here the visualization came into the work, we use visualization for explaining the data.

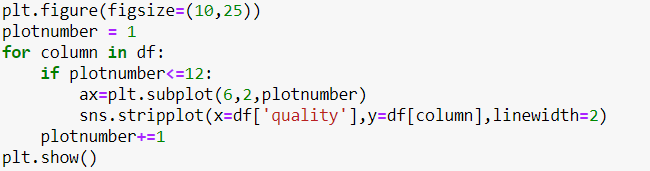
****

**OUTPUT:**

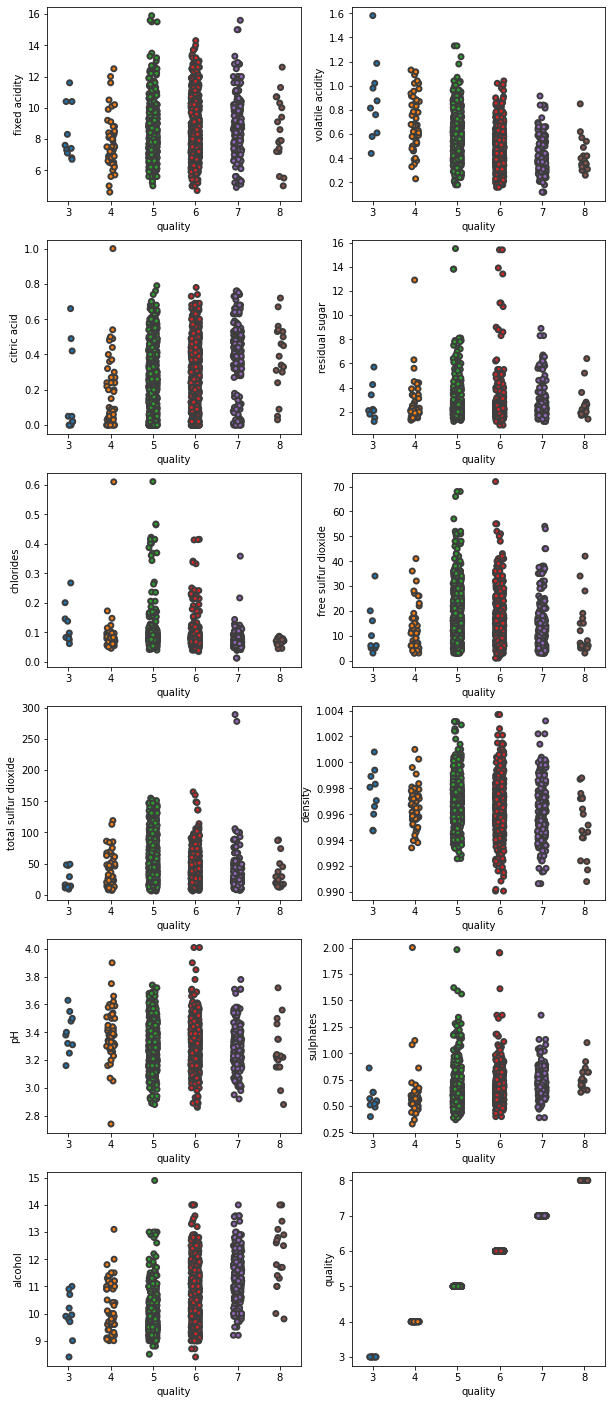
The below graph gives visuals about the skewness as well as about the distribution of data

**BIVARIATE ANALYSIS:**

As analysis is most important part in dataset through deep analysis, we can decide which column is contributing more to achieve our target.

****

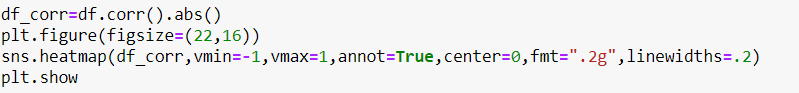
**OUTPUT:**

****

Through strip plot we can analyse which column is contributing how much in respected quality

**CORRELATION PLOT:**

When we performing any machine learning operations then we have to study the data features deep, there are many ways by which we can visualize each of the features easily. Now, we will perform a correlation on the data to see how many features are there they correlated to each other.

****

****

In correlation graph we see and study relation of feature and label and also study relation among feature and feature, if you think that why we have to discard those correlated, because relationship among them is equal they equally impact on model accuracy so, we delete one of them

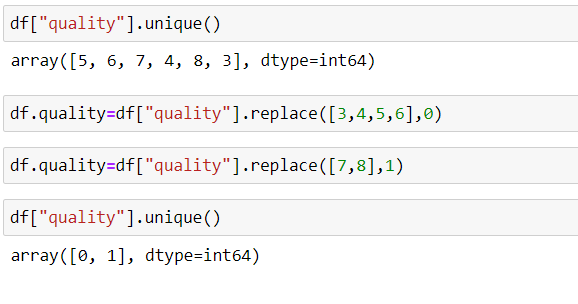
now, we have to find those features that are fully correlated to each other by this we reduce the number of features from the data**.**

**FEATURE ENGINEERING:**

What is a feature and why we need the engineering of it? Basically, all machine learning algorithms use some input data to create outputs. This input data comprises features, which are usually in the form of structured columns. Algorithms require features with some specific characteristics to work properly. Here, the need for **feature engineering** arises. I think feature engineering efforts mainly have two goals:

Preparing the proper input dataset, compatible with the machine learning algorithm requirements.

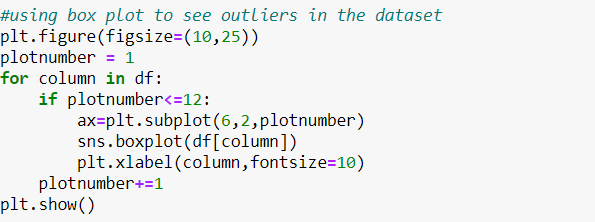
Improving the performance of machine learning models.

****

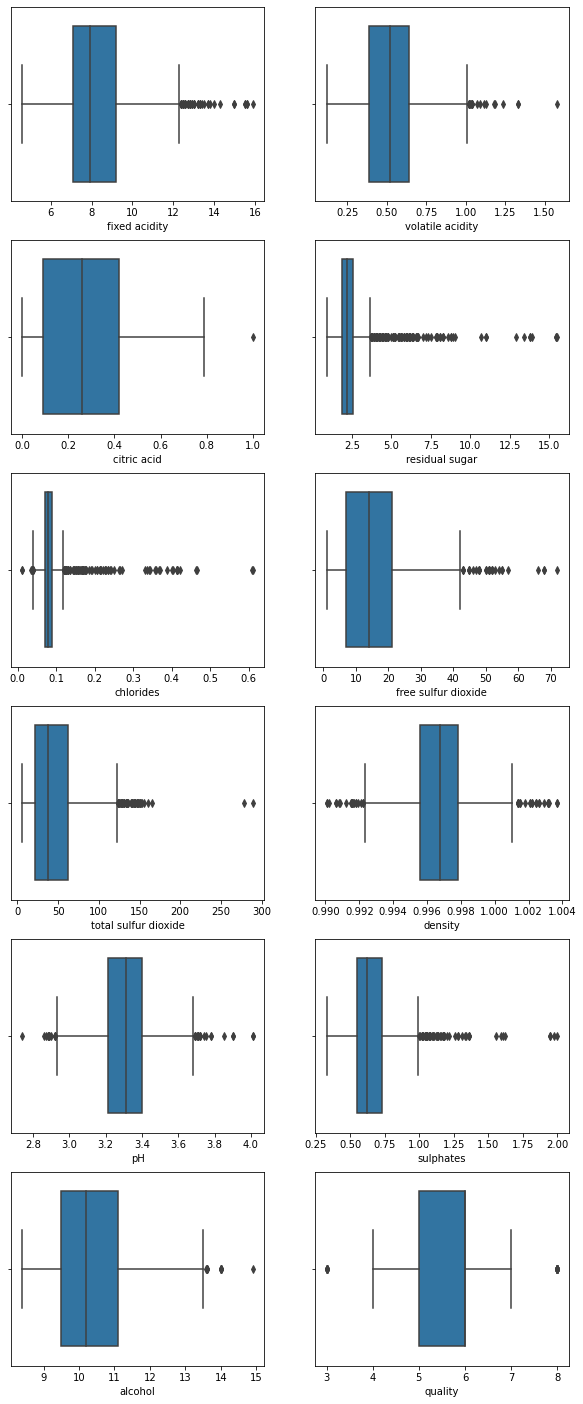
**OUTLIERS:**

Outliers in dataset increases the skewness in dataset, we remove outliers to remove skewness in dataset, but the most important thing we need to keep in mind while removing outliers is that there is not too much data loss as losing too much data is not a good practice, so while removing outliers we try to make our data loss less than 10%.

As visualisation is important part in machine learning we have a “boxplot” to see and analyse outlier in dataset.

****

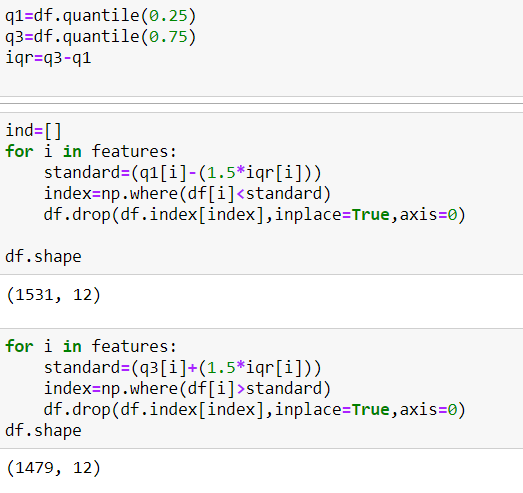
**OUTPUT:**

****

These black dots represent outliers. There are various methods to remove outlier:

* 1. **Z score**
  2. **Dot quartile**
  3. **Interquartile**

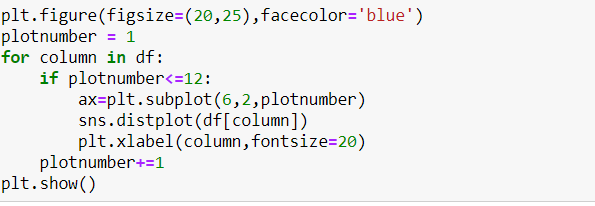
**INTERQUARTILE:**

****

This is how we apply interquartile to remove outliers in respected columns.

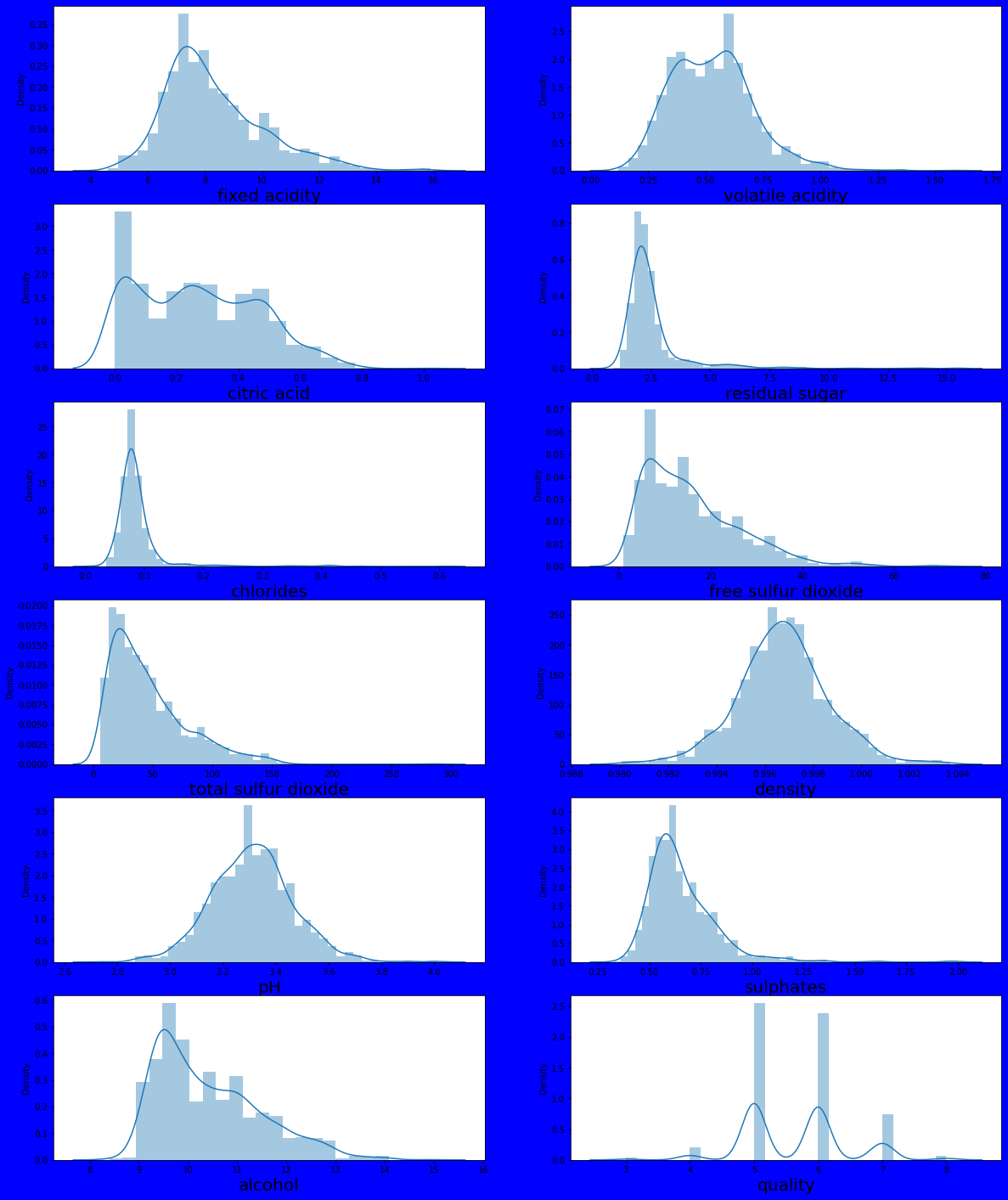
**SKEWNESS:**

To avoid overfitting and underfitting problem we remove skewness in dataset**.**

****

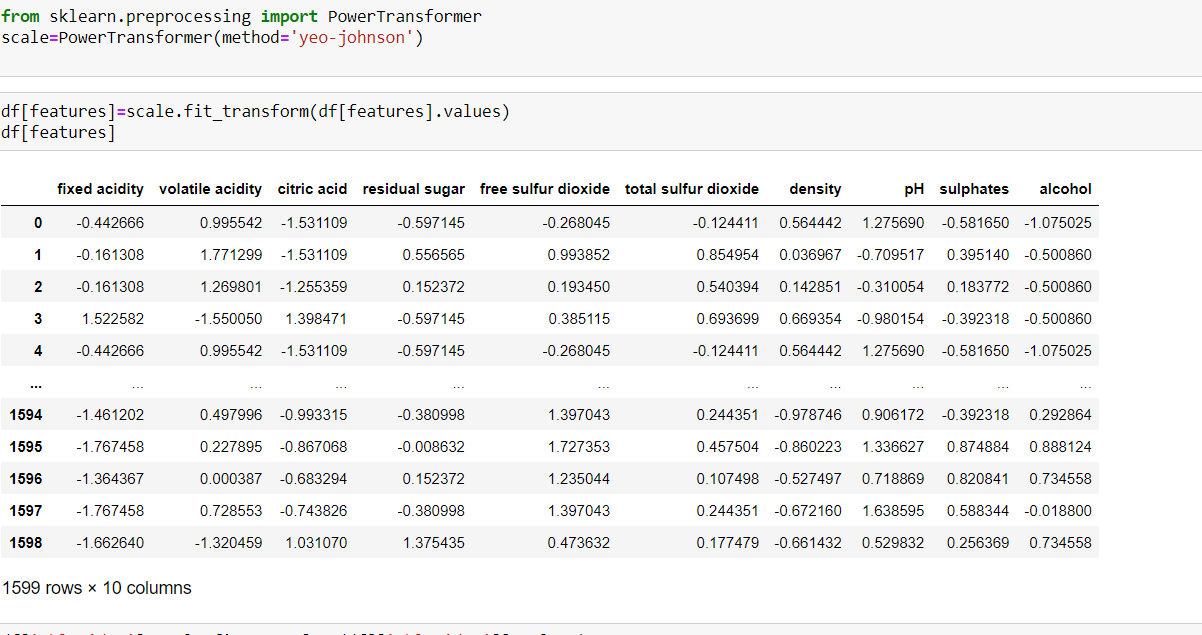
**OUTPUT:**

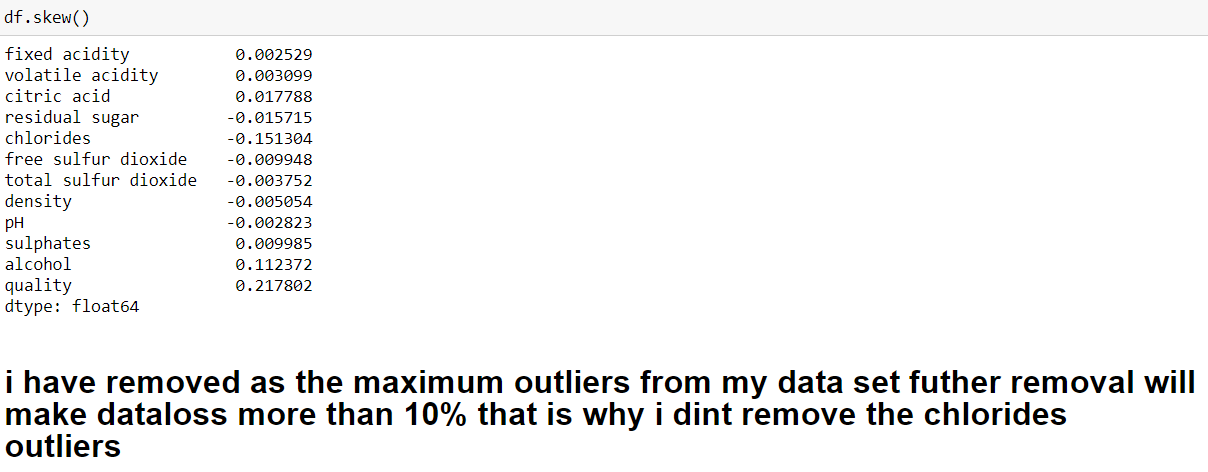
Through various plot we see and analyse skewness in dataset

****

**There are various method to remove skewness in dataset:**

* 1. **Log transform:** The logarithm, x to log base 10 of x, or x to log base e of x (ln x), or x to log base 2 of x, is a strong transformation with a major effect on distribution shape. It is commonly used for reducing right skewness and is often appropriate for measured variables. It can not be applied to zero or negative values.
  2. **Sqrt:** The square root, x to x^ (1/2) = sqrt(x), is a transformation with a  
     moderate effect on distribution shape: it is weaker than the logarithm  
     and the cube root. It is also used for reducing right skewness, and also  
     has the advantage that it can be applied to zero values. Note that the  
     square root of an area has the units of a length. It is commonly applied  
     to counted data, especially if the values are mostly rather small.
  3. **Cube root:** The cube root, x to x^ (1/3). This is a fairly strong transformation with a substantial effect on distribution shape: it is weaker than the logarithm. It is also used for reducing right skewness, and has the advantage that it can be applied to zero and negative values. Note that the cube root of a volume has the units of a length. It is commonly applied to rainfall data.
  4. **Power transformer:** Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. This is useful for modelling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired. Currently, Power Transformer supports the Box-Cox transform and the Yeo-Johnson transform. The optimal parameter for stabilizing variance and minimizing skewness is estimated through maximum likelihood. Box-Cox requires input data to be strictly positive, while Yeo-Johnson supports both positive or negative data. By default, zero-mean, unit-variance normalization is applied to the transformed data.





**BALANCING DATASET FOR GOOD PREDICTION:**

Suppose that you are working in a given company and you are asked to create a model that, based on various measurements at your disposal, predicts whether a product is defective or not. You decide to use your favourite classifier, train it on the data and you get a 96.2% accuracy   
Your boss is astonished and decides to use your model without any further tests. A few weeks later he enters your office and underlines the uselessness of your model. Indeed, the model you created has not found any defective product from the time it has been used in production.  
After some investigations, you find out that there is only around 3.8% of the product made by your company that are defective and your model just always answers “not defective”, leading to a 96.2% accuracy. The kind of “naive” results you obtained is due to the imbalanced dataset you are working with.

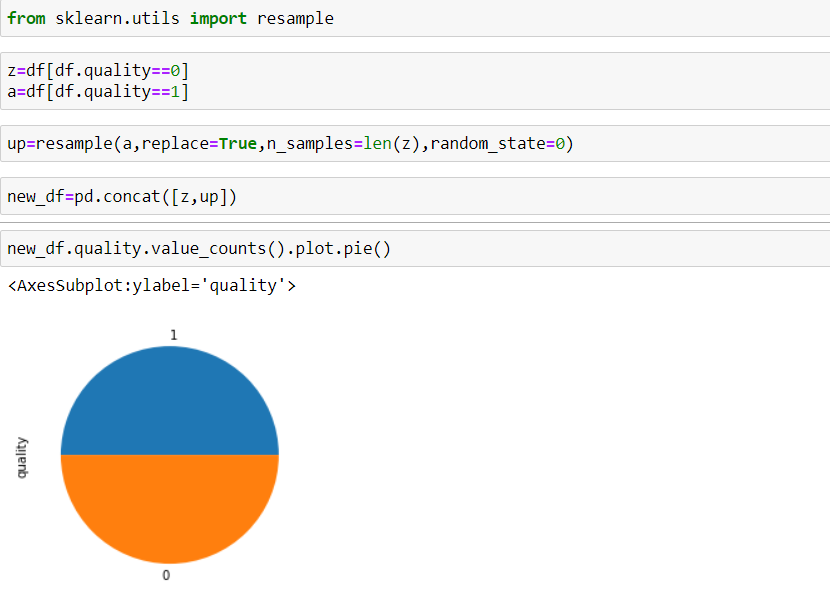
Apart from using different evaluation criteria, one can also work on getting different dataset. Two approaches to make a balanced dataset out of an imbalanced one are under-sampling and over-sampling.

**Under-sampling:**

Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. By keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modelling.

**Over-sampling:**

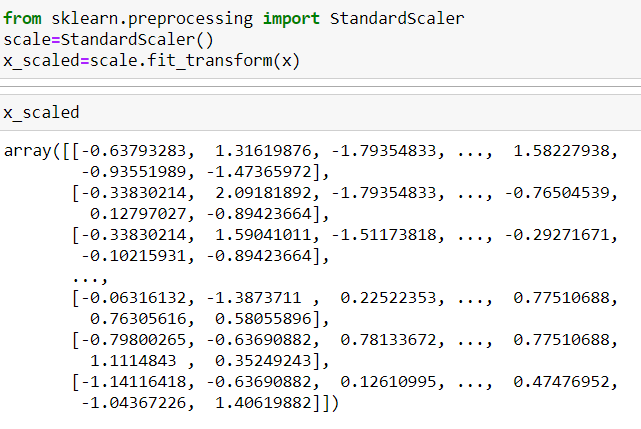
On the contrary, oversampling is used when the quantity of data is insufficient. It tries to balance dataset by increasing the size of rare samples. Rather than getting rid of abundant samples**.**

****

**STANDARDISATION**:

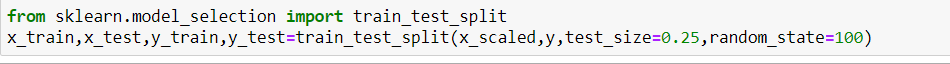
**Standardisation is a technique to convert whole dataset to one scale using various scaling feature**:

* 1. **STANDARD SCALER:** Standard Scaler follows Standard Normal Distribution (SND). Therefore, it makes mean = 0 and scales the data to unit variance.
  2. **MIN MAX SCALER:** Min Max Scaler scales all the data features in the range [0, 1] or else in the range [-1, 1] if there are negative values in the dataset. This scaling compresses all the inliers in the narrow range [0, 0.005].

**NORMALISTION**:

**SPLITTING DATASET**:

Now we split our dataset into two variables:

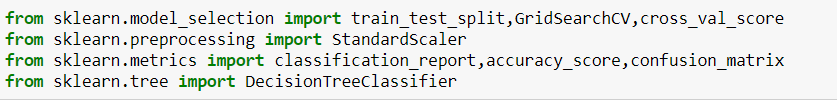
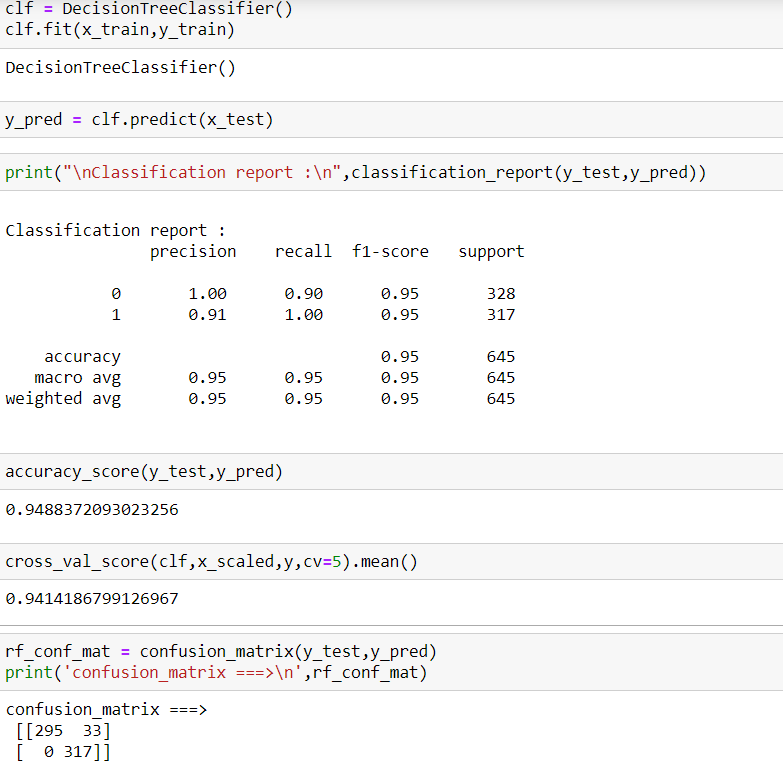


Here we split our dataset into two parts:

* 1. **Training Data:** used to fit the machine learning model.
  2. **Testing Data**: used to evaluate the fit machine learning model.
  3. **Test Size**: it can vary from 0-30% as per your choice.
  4. **Random State**: you can select it as per your choice

MODEL SELECTION:

This is the last step where we apply any suitable model which will give more accuracy.



## Evaluation of the classification model:

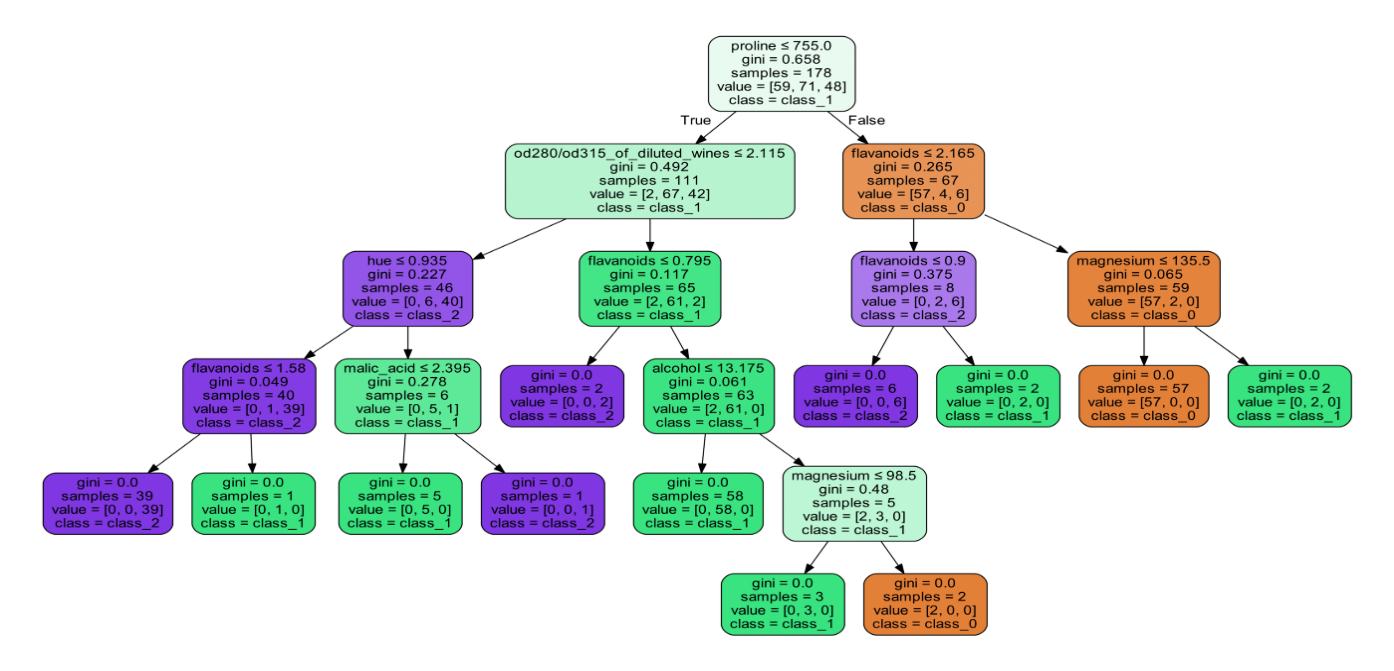
In machine learning once we have a result of the classification problem, how do we measure how accurate our classification is? For regression problem we have different metrices like R-squared score, Mean squared error etc. What are the metrices to measure the credibility of a classification model?

Metrices in a regression problem, the accuracy is generally measured in terms of the difference in the actual values and the predicted values. In a classification problem the credibility of the model is measured using the confusion matrix generated i.e., how accurately the true positives and true negatives were predicted. The different metrices used for this purpose are:

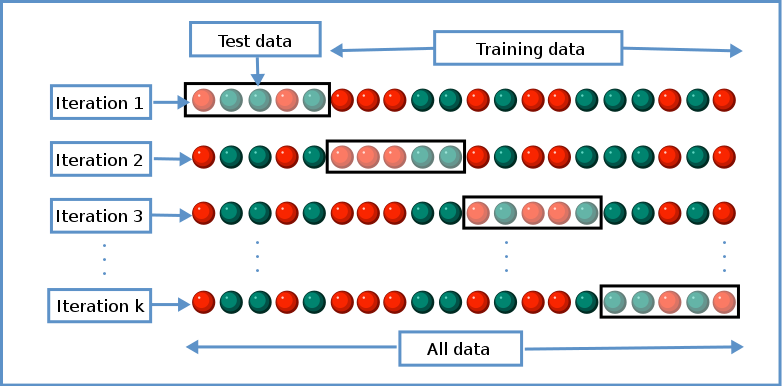
* Accuracy
* Precision
* Recall
* F1 Score
* Specificity
* AUC (Area Under Curve)
* ROC (Receiver Operator characteristics)

**DECISION TREE**:

Decision tree algorithm is one of the most versatile algorithms in machine learning which can perform both classification and regression analysis. It is very powerful and works great with complex datasets. Apart from that, it is very easy to understand and read. That makes it more popular to use. As the name suggests, this algorithm works by dividing the whole dataset into a tree-like structure based on some rules and conditions and then gives prediction based on those conditions. Let’s understand the approach to decision tree with a basic scenario. Suppose it’s Friday night and you are not able to decide if you should go out or stay at home. Let the decision tree decide it for you.



**CROSS VALIDATION SCORE**: Suppose you train a model on a given dataset using any specific algorithm. you tried to find the accuracy of the trained model using the same training data and found the accuracy to be 95% or may be even 100%. **What does it mean?** Is your model ready for prediction? The answer is no. Why? Because your model has trained itself on the given data i.e., it knows the data it has generalized over it well. But when you try to predict over a new set of data, it’s more likely to give you very bad accuracy because it has never seen the data before and thus it fails to generalize well over it. This is the problem of overfitting. To tackle such problem, **Cross-Validation** comes into the picture. Cross-validation is a resampling technique with a basic idea of dividing the training dataset into two parts (train and test). On one part(train) you try train the model and on the second part(test), the data which is unseen for the model you make the predict and check how well your model works on it. If the model works with good accuracy on your test data. It means that the model has not overfitted the training data and can be trusted with prediction whereas if it performs with bad accuracy then our model is not to be trusted and we need to tweak our algorithm.



**CONFUSION MATRIX**:



Where the terms have the meaning

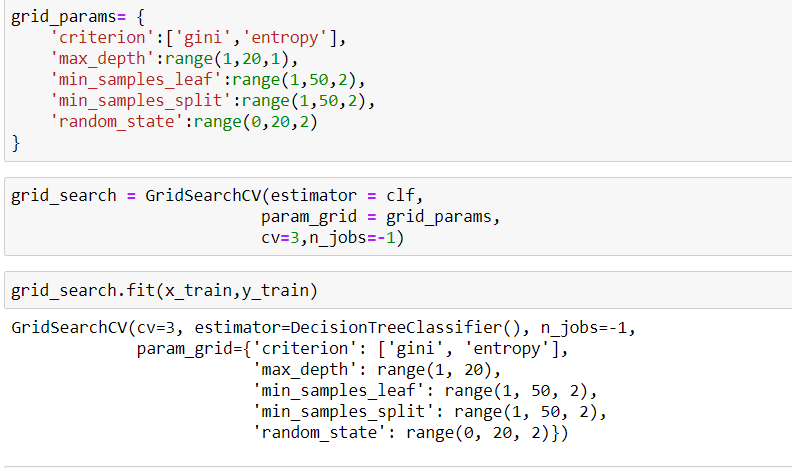
TP = A result that was predicted as positive by the classification model and also is positive

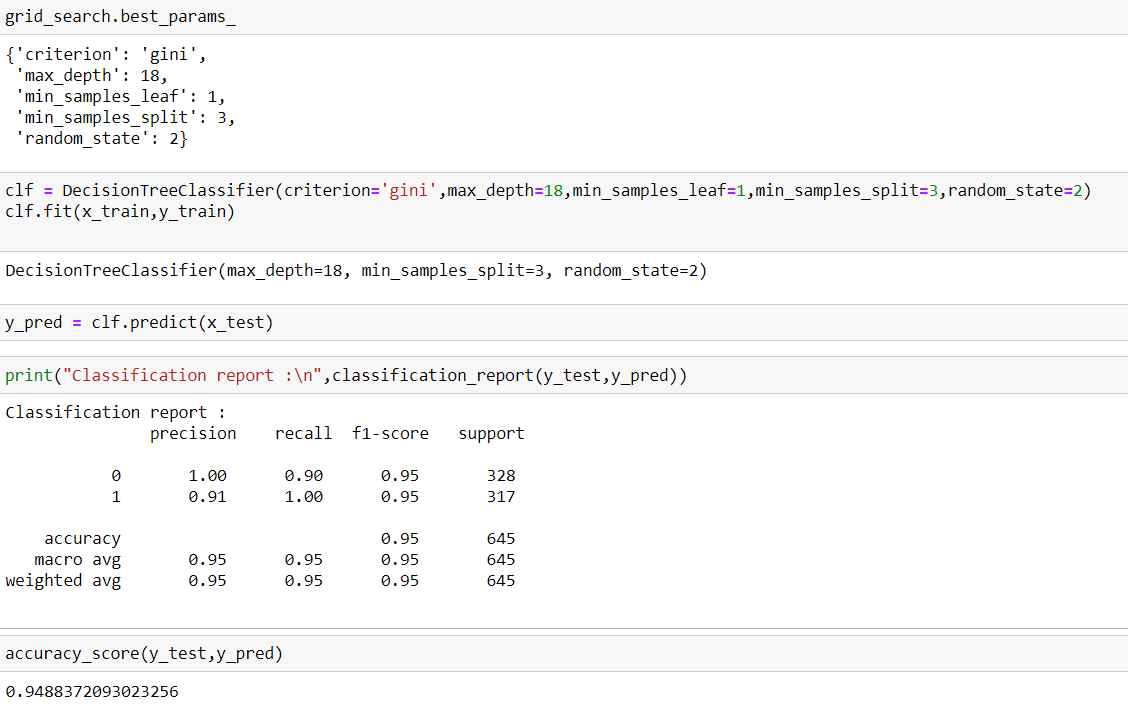
TN = A result that was predicted as Negative by the classification model and also is Negative

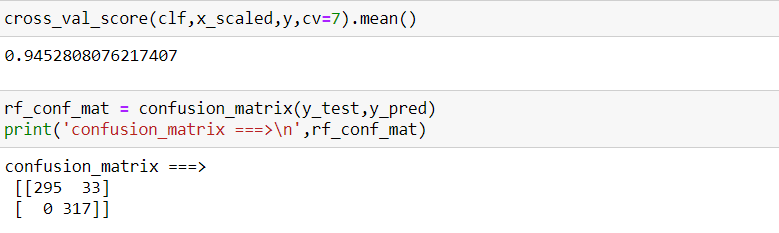
FP = A result that was predicted as positive by the classification model but actually it is Negative

FN = A result that was predicted as Negative by the classification model but actually it is positive

**HYPERPARAMETER TUNING**:





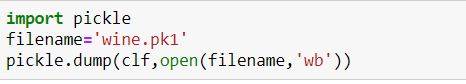


Hyper parameter tuning is done to find out best parameter for the best model with good accuracy score, so that we can increase the accuracy of the best model to more extent.

**AUC AND ROC CURVE:**



**SAVING BEST MODEL:**



After completing the model, the last and foremost step is to save the model.

**Endnote:**

wine is a very interesting topic I have deal with, I have got success in predicting good quality and bad quality wine as you can see in the above predicted model. Here are some of my assumptions based on above model which will help out in deciding which features t be more focused about while making a good quality wine.

**By looking into the details, we can see that good quality wines have higher levels of alcohol on average, as we can see it in heat map it is having a good correlation of almost 50 %.**

 Ingredients which contribute more for type of the wine. – Chlorides, sulphates, free sulphur dioxide, total sulphur dioxide, volatile. Acidity. • Ingredients which contribute more for quality of the wine. – Alcohol(major), density.

Data science is a trial-and-error process. The more you try the more you learn and more you succeed in your prediction. I am sure there are ways to improve this model and accuracy. Play around and see if you can get a better result.

I hope you like my article, I know its bit long, but I try to put each and every thing what I learn in my data science journey Infront of you. I try to put each and every information about tools and library I have used to make you understands better.

*I love everything that’s old, — old friends, old times, old manners, old books, old wine. — Oliver Goldsmith*

With these beautiful lines I am ending my article………….

**ADVERTISING SALES CHANNEL PREDICTION**

**Overview**

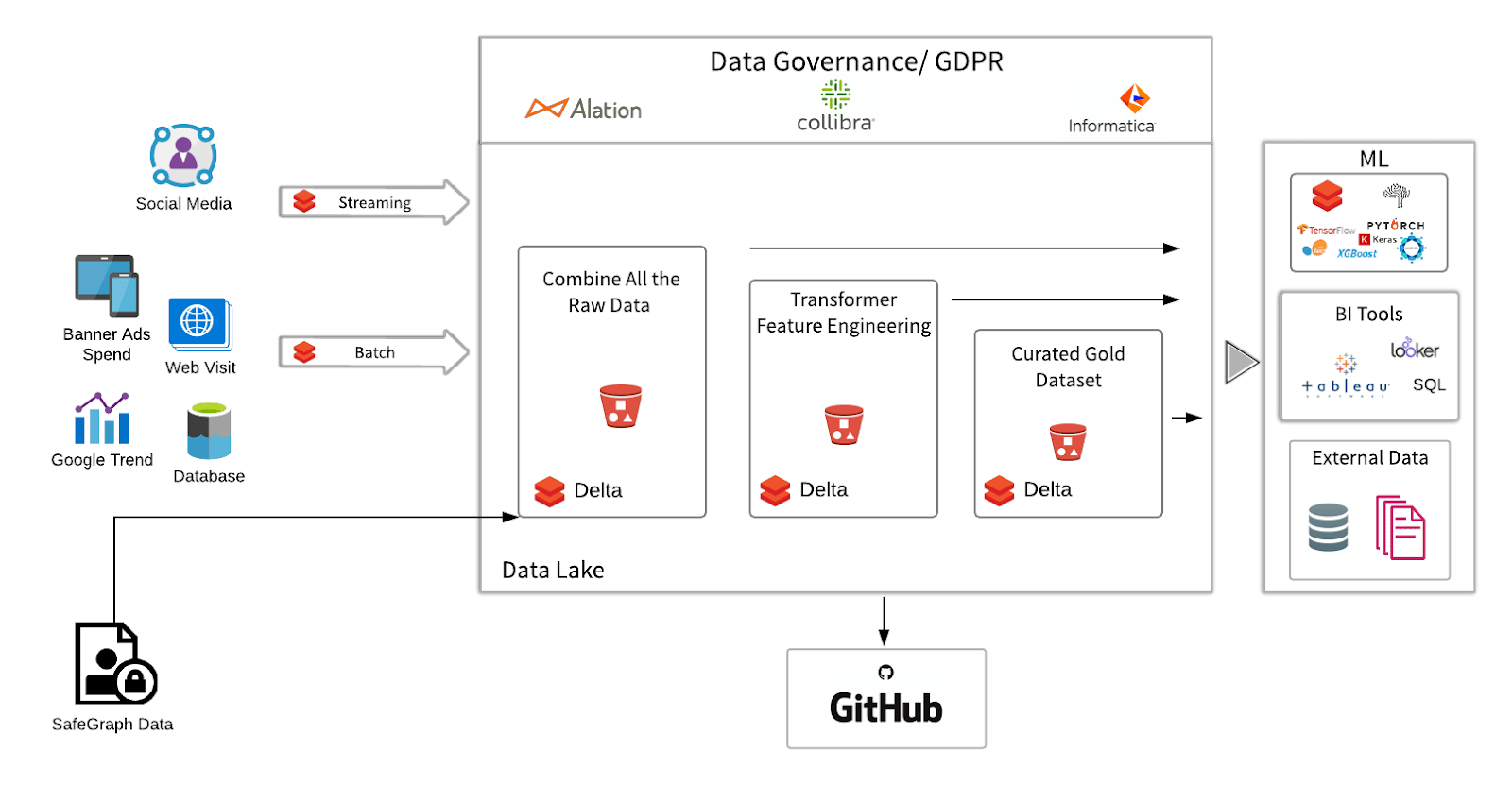
* **Basics understanding of Advertising Sales Channel Prediction.**
* **Data description**
* **Importing modules**
* **Study dataset**
* **Handle null values**
* **VISUALIZATION**
* **Standardisation**
* **Applying model**
* **Save model**
* **Endnote**

# **Introduction:**

What’s the first thing that we need to take care in our mind when we start a business is “SALES”. What’s all prospects we have to boost our product, although I am data science student so here, I am writing a blog for all those who are going to be entrepreneur in upcoming future. I am here writing about a regression problem that most people encounter in their life while doing a business, this article will help them out in deciding what all they can do to boost their business. Here I am writing an article about the most important prospects of every business is “SALES PREDICTION”.

Sales prediction means predicting how much of a product people will buy based on factors such as the amount you spend to advertise your product, the segment of people you advertise for, or the platform you are advertising on about your product.

Typically, a product’s sales are primarily dependent on how much you spend on advertising it, as the more people your product reaches, the more sales will increase as long as the quality of your product is good. Now in the section below, I will take you through a machine learning project on sales prediction using Python



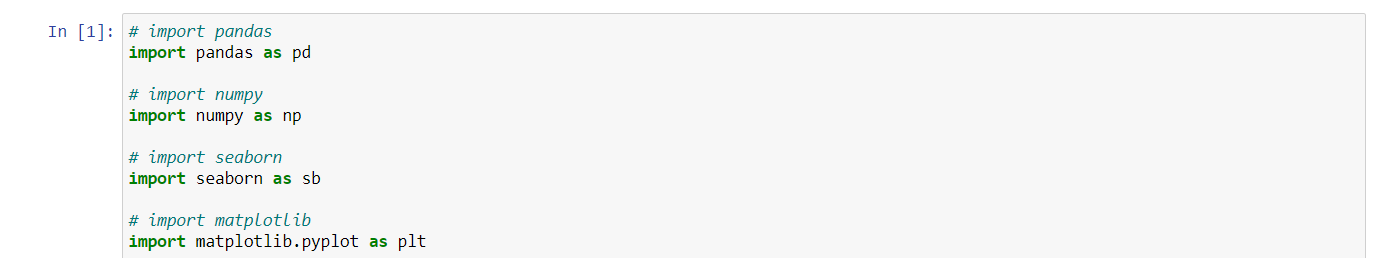
## Objectives:

The objective of this project are as follows:

1. To experiment with different regression methods to see which yields the highest accuracy
2. Build a model which predicts sales based on the money spent on different platforms for marketing.

### Importing modules**:**

Let’s import some basic library:



**NUMPY:**   
NumPy can be used to perform a wide variety of mathematical operations onarrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

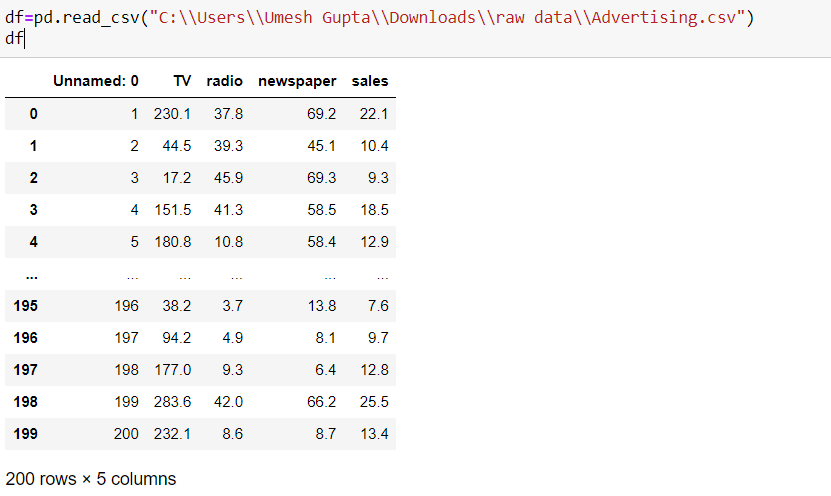
**PANDAS:**

Pandas is mainly used for data analysis**.**

**SEABORN & MATPLOTLIB:**

Both the library is used for visualisation

### Study dataset**:**



we have to predict the sales on the basis of TV, RADIO AND NEWSPAPER.

Unnamed is unwanted column as it is not giving any vital information about the target column.

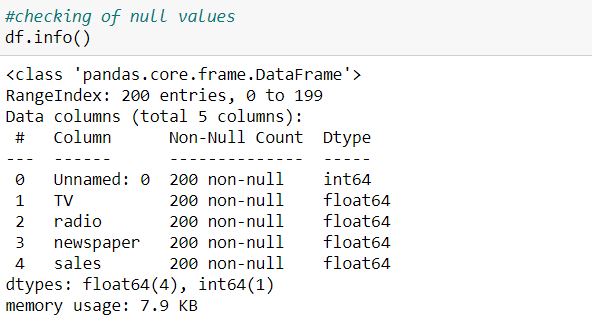
We see a bunch of columns with some values in them. Now, in every machine learning program, there are two things, **features** and **labels.**

 Features are the part of a dataset which are used to predict the label. And labels on the other hand are mapped to features.

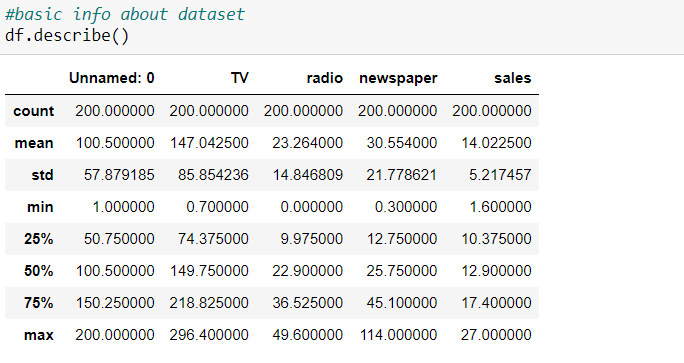
After the model has been trained, we give features to it, so that it can predict the labels.

So, if we analyse this dataset, since we have to predict the wine quality, the attribute quality will become our **label** and the rest of the attributes will become the **features**.

INFORMATON**:**



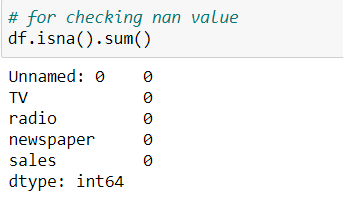
DESCRIBE**:**



From info we get the info about the datatype of respected columns which helps us in visualization to predict which graph is best fit for columns.

It describes the distribution of your data: 50 should be a value that describes the middle of the data, also known as median. 25, 75 is the border of the upper/lower quarter of the data. You can get an idea of how skew your data is. Note that the mean is higher than the median, which means your data is right skewed.

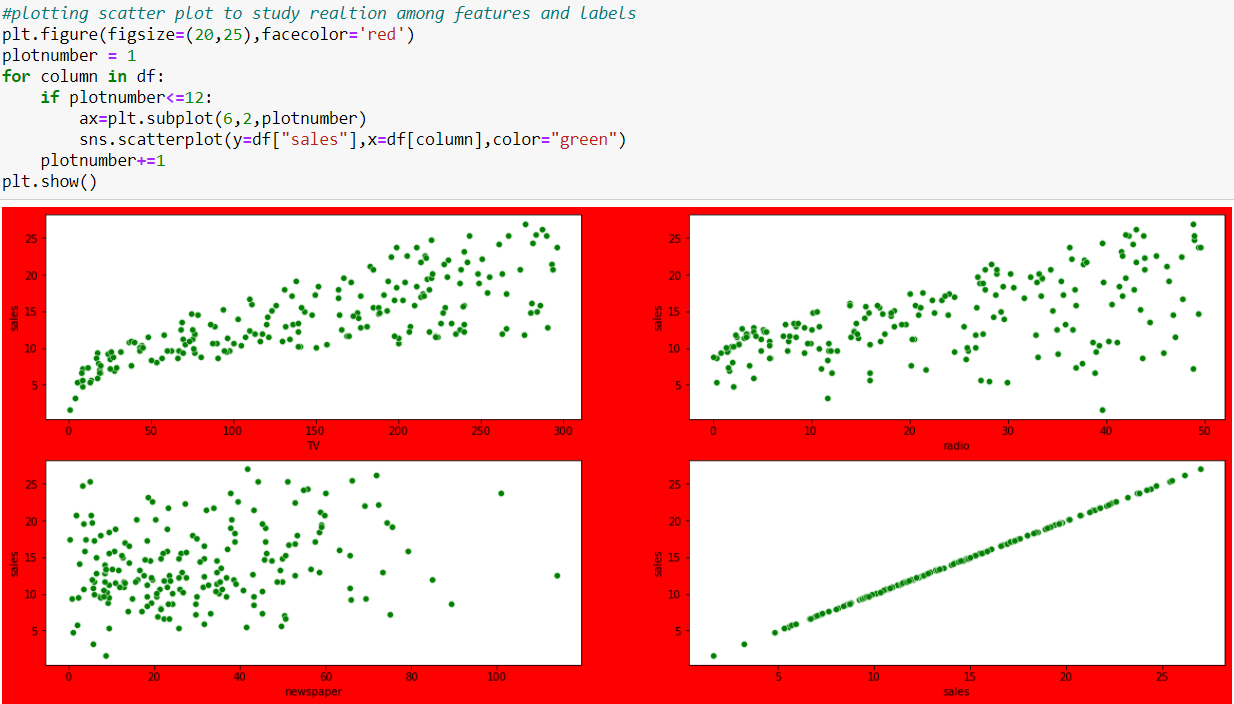
**NAN VALUE:**



We don’t leave nan value in data as it is chances that it makes data set overfit during prediction. Nan value is often fill with mean and median for continuous data and with mode for categorical column.

VISUALIZATION**:**

We know that the “image speaks everything” here the visualization came into the work, we use visualization for explaining the data.

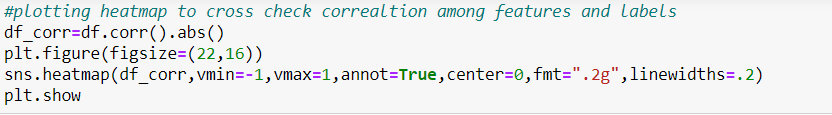


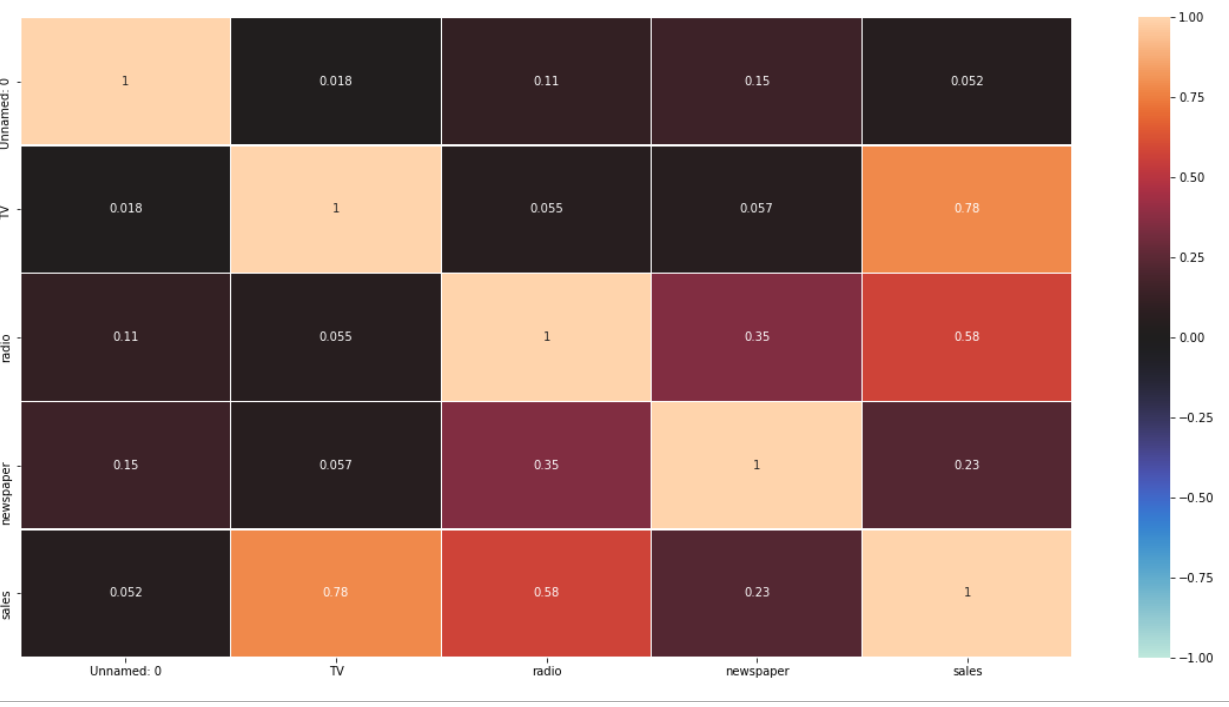
The above graph shows the relationship of target column(sales) with other features:

1. In first plot we can linear relationship of sales with T.V i.e., when the investment in tv advertisement increases sale also increases.
2. In second plot we find linear relationship of radio with sales i.e., when the investment in radio advertisement increases sale also increases.
3. In third plot we don’t find any linear relationship between newspaper and sales.
4. The above plot is also the bivariate analysis of the dataset.

CORRELATION PLOT**:**

When we performing any machine learning operations then we have to study the data features deep, there are many ways by which we can visualize each of the features easily. Now, we will perform a correlation on the data to see how many features are there they correlated to each other.

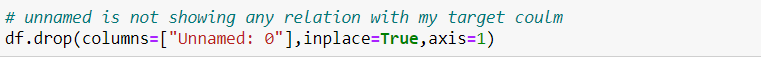




In correlation graph we see and study relation of feature and label and also study relation among feature and feature, if you think that why we have to discard those correlated, because relationship among them is equal they equally impact on model accuracy so, we delete one of them

now, we have to find those features that are fully correlated to each other by this we reduce the number of features from the data**.**

DROPPING COLUMN:

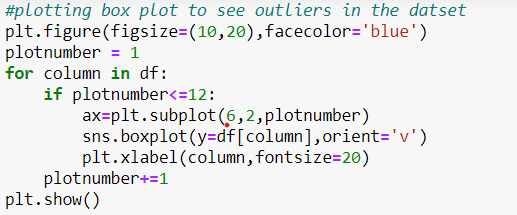


As unnamed is unwanted column and it also showing very least correlation with the sales column.

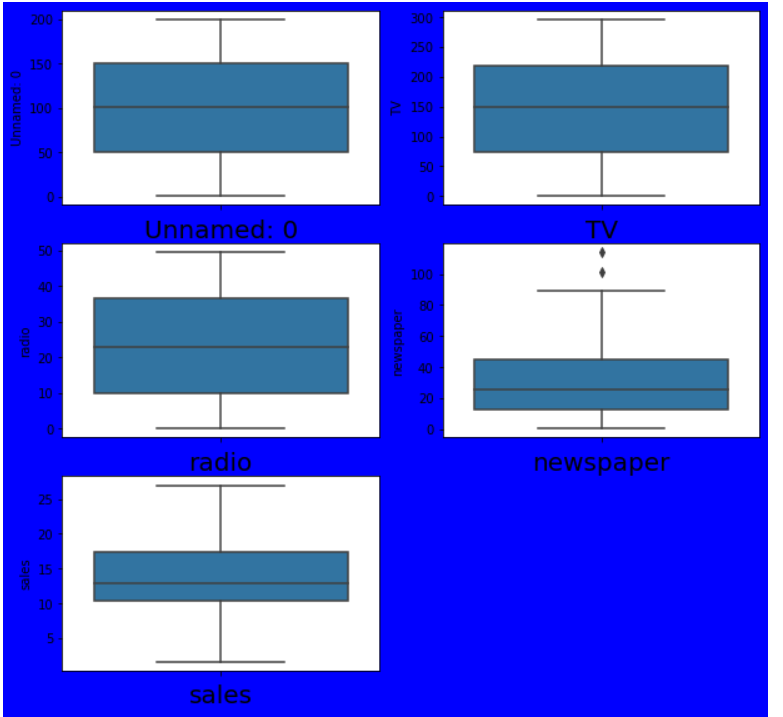
OUTLIERS**:**

Outliers in dataset increases the skewness in dataset, we remove outliers to remove skewness in dataset, but the most important thing we need to keep in mind while removing outliers is that there is not too much data loss as losing too much data is not a good practice, so while removing outliers we try to make our data loss less than 10%.

As visualisation is important part in machine learning we have a “boxplot” to see and analyse outlier in dataset.



OUTPUT**:**



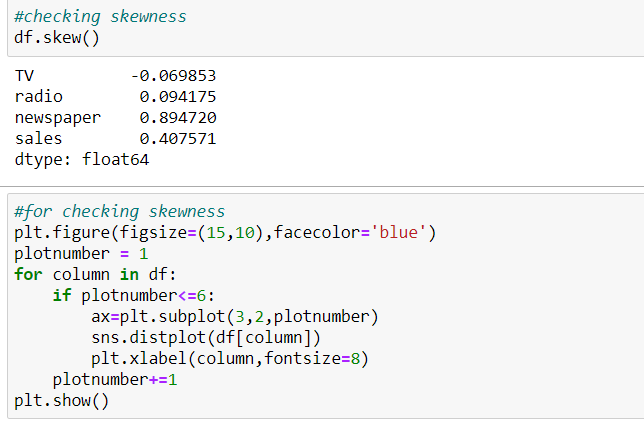
These black dots represent outliers. There are various methods to remove outlier:

* 1. **Z score**
  2. **Dot quartile**
  3. **Interquartile**

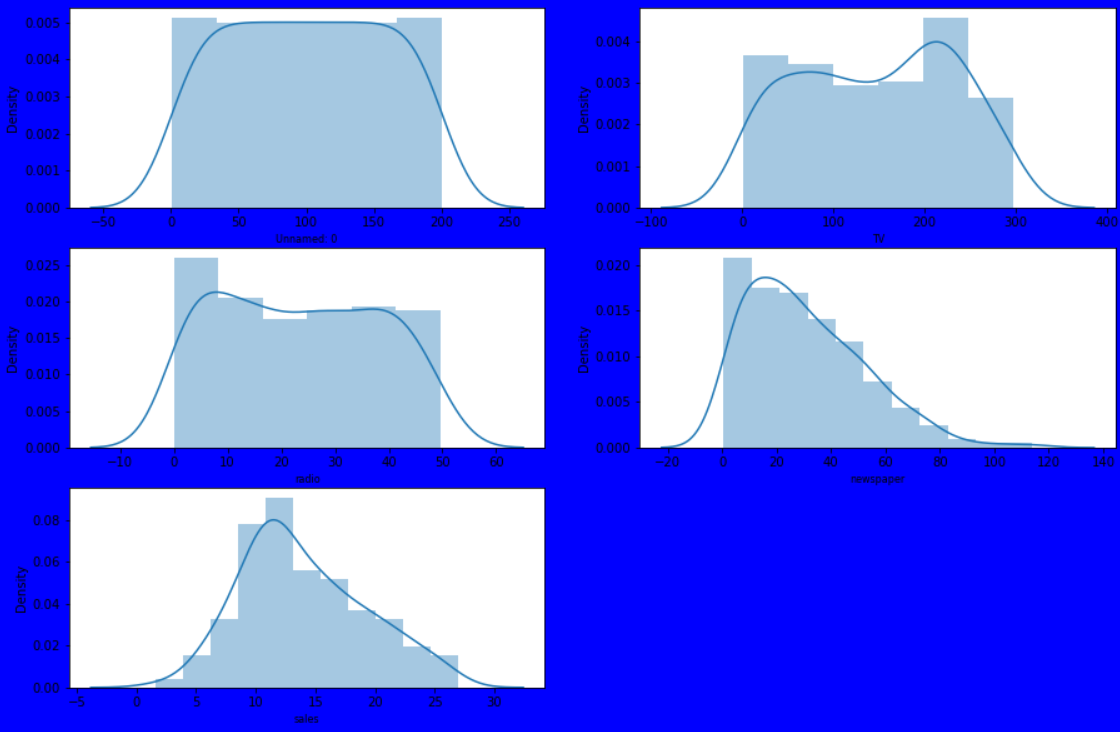
Sometime we keep the outlier in dataset just to prevent data loss, and also when the column is not having any relationship with the target variable so we avoid outlier and move forward with the dataset.

SKEWNESS**:**

To avoid overfitting and underfitting problem we remove skewness in dataset**.**



OUTPUT**:**



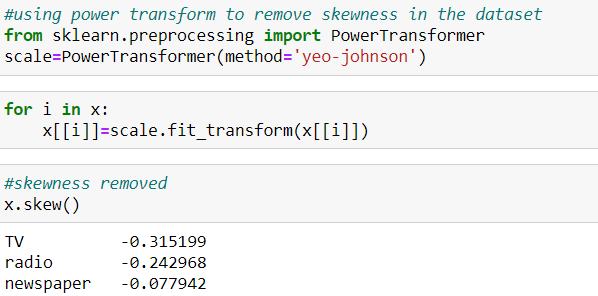
Through distribution plot we see the Gaussian Distribution of the column.

In newspaper plot we can see the slop is skewed in right direction.

We tried to bring the curve to the centre by applying various skewness removal technique.

Power transformer**:**

Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. This is useful for modelling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired. Currently, Power Transformer supports the Box-Cox transform and the Yeo-Johnson transform. The optimal parameter for stabilizing variance and minimizing skewness is estimated through maximum likelihood. Box-Cox requires input data to be strictly positive, while Yeo-Johnson supports both positive or negative data. By default, zero-mean, unit-variance normalization is applied to the transformed data.

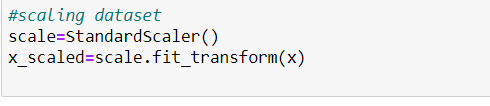


We don’t remove skewness for target column, now as our skewness is removed the dataset is good to go for model building.

STANDARDISATION:

Standardisation is a technique to convert whole dataset to one scale using various scaling feature:

* 1. **STANDARD SCALER:** Standard Scaler follows Standard Normal Distribution (SND). Therefore, it makes mean = 0 and scales the data to unit variance.

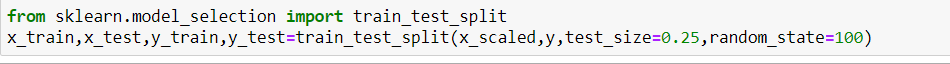
****

Most of the times, our dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidian distance between two data points in their computations, this is a problem.

If left alone, these algorithms only take in the magnitude of features neglecting the units. The results would vary greatly between different units, 5kg and 5000gms. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To supress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling

SPLITTING DATASET:

Now we split our dataset into two variables:

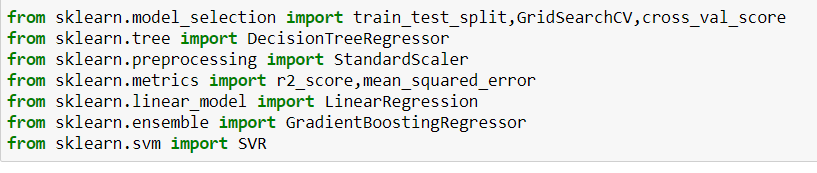


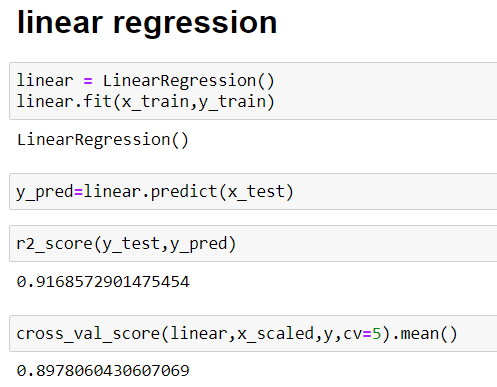
Here we split our dataset into two parts:

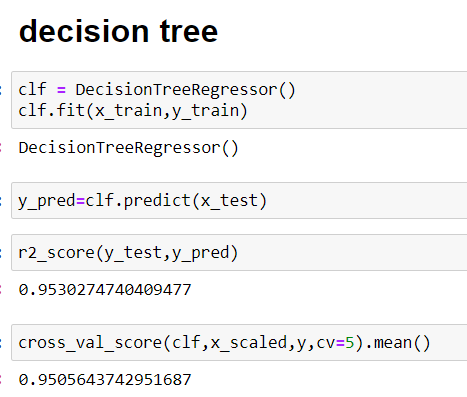
* 1. **Training Data:** used to fit the machine learning model.
  2. **Testing Data**: used to evaluate the fit machine learning model.
  3. **Test Size**: it can vary from 0-30% as per your choice.
  4. **Random State**: you can select it as per your choice

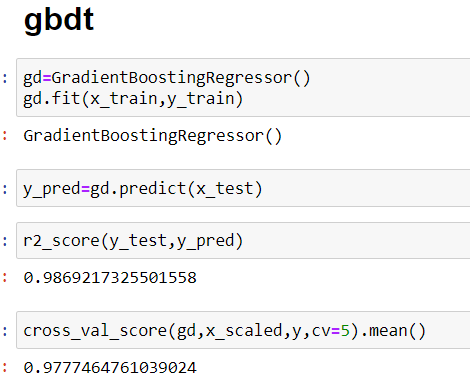
MODEL SELECTION:

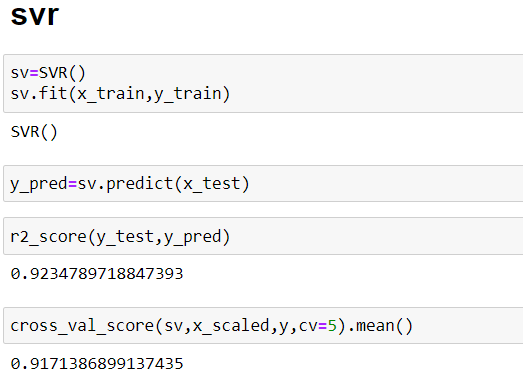
This is the last step where we apply any suitable model which will give more accuracy.









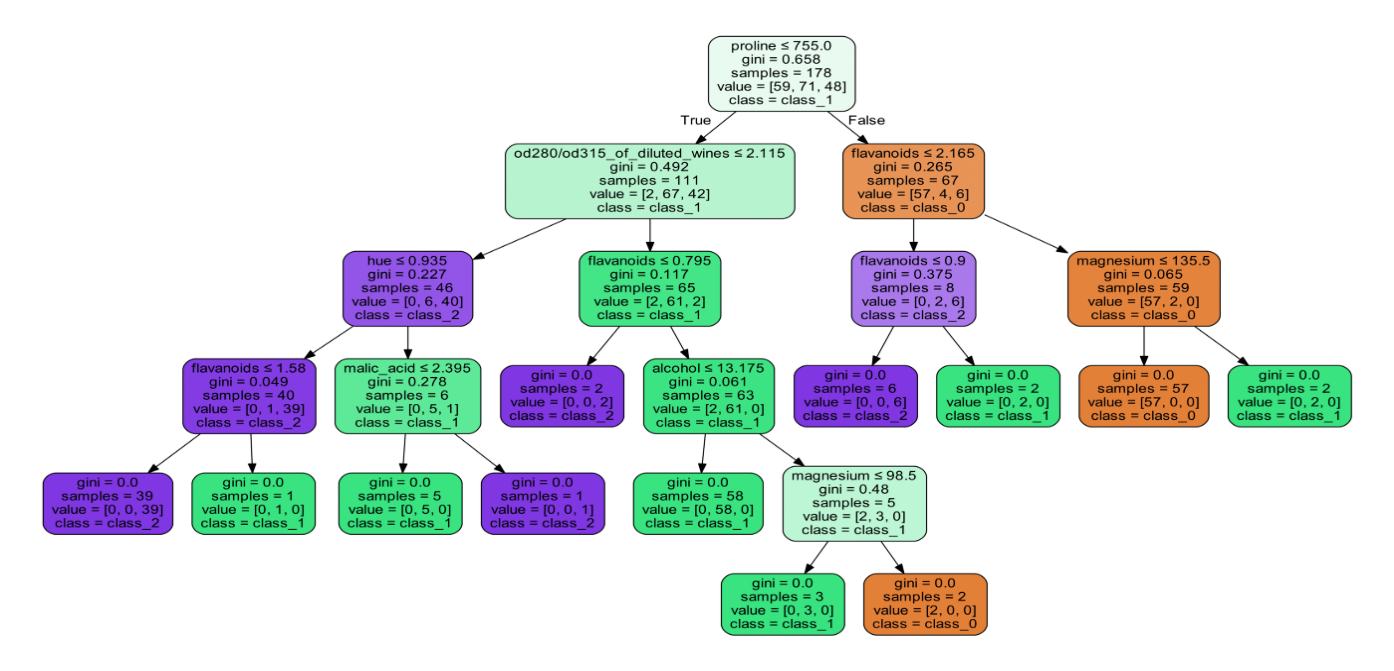


We select the best model on the basis of difference between r-2 score and cross validation score. on the basis of this Decision tree is my best model with r-2 score=95.3% and cross validation score=95%

## Evaluation of the classification model:

**DECISION TREE**:

Decision tree algorithm is one of the most versatile algorithms in machine learning which can perform both classification and regression analysis. It is very powerful and works great with complex datasets. Apart from that, it is very easy to understand and read. That makes it more popular to use. As the name suggests, this algorithm works by dividing the whole dataset into a tree-like structure based on some rules and conditions and then gives prediction based on those conditions. Let’s understand the approach to decision tree with a basic scenario. Suppose it’s Friday night and you are not able to decide if you should go out or stay at home. Let the decision tree decide it for you.



Although we may or may not use the decision tree for such decisions, this was a basic example to help you understand how a decision tree makes a decision. So how did it work?

* It selects a root node based on a given condition, e.g., our root node was chosen as time >10 pm.
* Then, the root node was split into child notes based on the given condition. The right child node in the above figure fulfilled the condition, so no more questions were asked.
* The left child node didn’t fulfil the condition, so again it was split based on a new condition.
* This process continues till all the conditions are met or if you have predefined the depth of your tree, e.g., the depth of our tree is 3, and it reached there when all the conditions were exhausted.

In machine learning once we have a result of the classification problem, how do we measure how accurate our classification is? For regression problem we have different metrices like R-squared score, Mean squared error etc. What are the metrices to measure the credibility of a classification model?

R-2 SCORE:

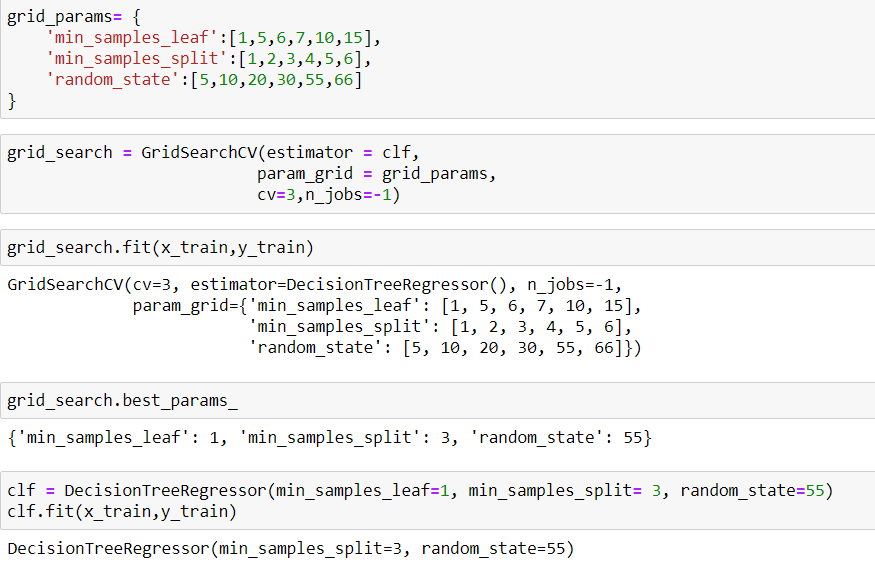
The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset. Simply, it is the difference between the samples in the dataset and the predictions made by the model.  If the value of the R2 is 1, it means that the model is perfect and if its value is 0, it means that the model will perform badly on an unseen dataset.

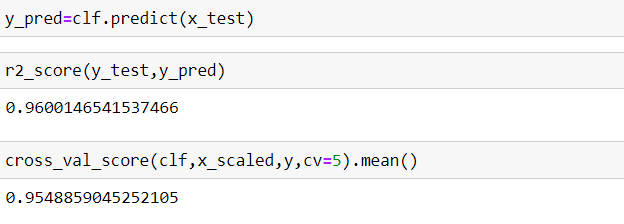
CROSS VALIDATION SCORE**:**

Suppose you train a model on a given dataset using any specific algorithm. you tried to find the accuracy of the trained model using the same training data and found the accuracy to be 95% or may be even 100%. **What does it mean?** Is your model ready for prediction? The answer is no. Why? Because your model has trained itself on the given data i.e., it knows the data it has generalized over it well. But when you try to predict over a new set of data, it’s more likely to give you very bad accuracy because it has never seen the data before and thus it fails to generalize well over it. This is the problem of overfitting. To tackle such problem**, Cross-Validation** comes into the picture. Cross-validation is a resampling technique with a basic idea of dividing the training dataset into two parts (train and test). On one part(train) you try train the model and on the second part(test), the data which is unseen for the model you make the predict and check how well your model works on it. If the model works with good accuracy on your test data. It means that the model has not overfitted the training data and can be trusted with prediction whereas if it performs with bad accuracy then our model is not to be trusted and we need to tweak our algorithm.

**HYPERPARAMETER TUNING**:

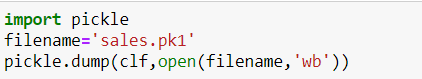
Hyper parameter tuning is done to find out best parameter for the best model with good accuracy score, so that we can increase the accuracy of the best model to more extent.





as after hyper para meter tuning, I can see that my model r-2 score has increase and reached to 96% and my cv score has also increased and reached to 95%

**SAVING BEST MODEL:**



After completing the model, the last and foremost step is to save the model.

ENDNOTE:

After running our four models, I used metric: R-squared and cross validation, to evaluate our model prediction performance. As we expected, Model 2 is the best in terms of metrics and cross validation with R-Squared: 95.3%, cross validation: 95%. It is reasonable that decision tree in Model 2 gives us superior “predictions”. However, from a perspective of “marginal impact” interpretation, model 2 may be the winners even though their performance measurements are behind just by a little margin. In the context of our business question focusing on the prediction of sales, Model 2 will be the best choice.