**Predicting the Sales of Products**

****

In this blog, I will explain the process of machine learning model, Bivariate and univariate analysis on the Sales channel dataset, the dataset contain information of dollar amount spent for ads on TV, Radio and newspaper. I will build machine learning model and predict sales of products.

**Sales Channel Dataset:**

Below is the description of Sales Channel dataset.

TV: advertising dollars spent on TV for a single product in a given market (in thousands of dollars)

Radio: advertising dollars spent on Radio

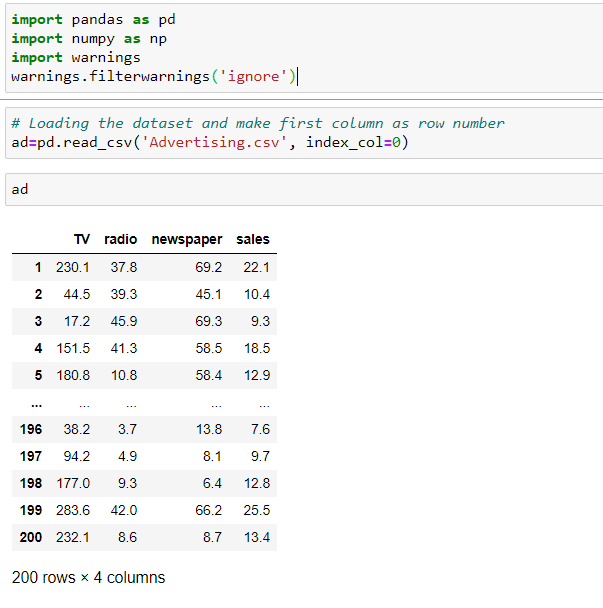
Newspaper: advertising dollars spent on Newspaper

Sales: sales of a single product in a given market (in thousands of widgets)

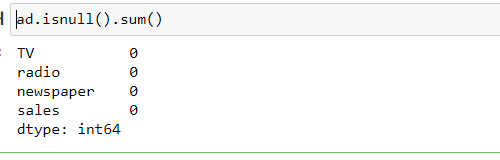
TYPE OF MACHINE LEARNING ALGORITHMS

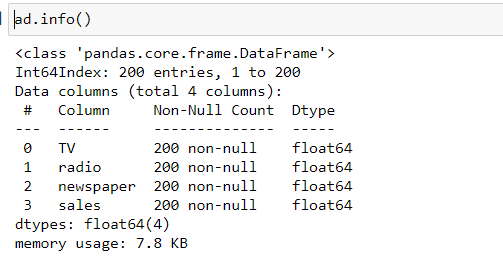
The target variable is sales which are continuous value so this is regression problem. I will build regression ML model.

**Importing important Libraries and Sales Channel dataset-**



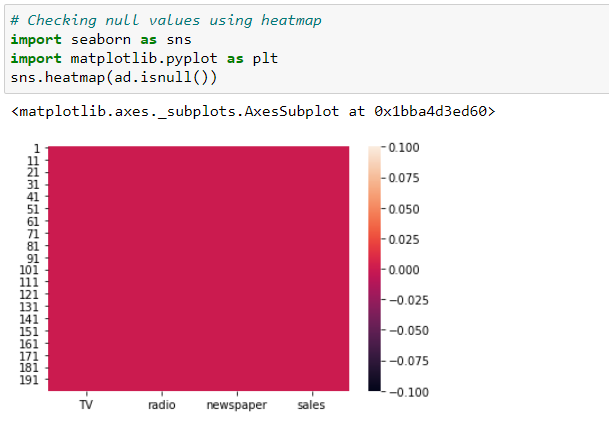
**Data exploring and analysis**



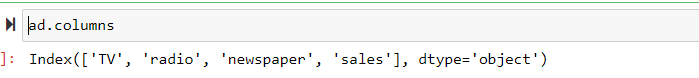


There are total 200 rows and 4 columns. There is no missing values and all the data in float64 data type. So no need to convert data type.

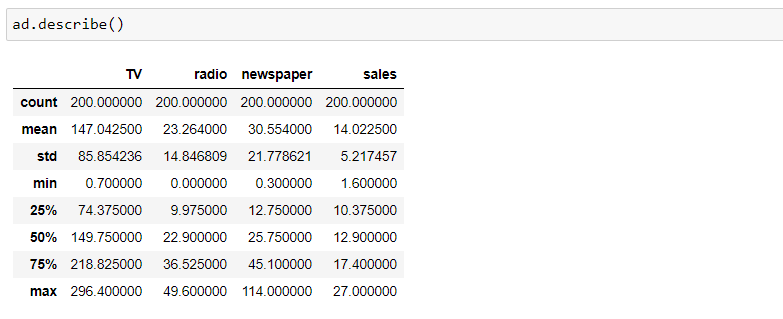
Checking null values using heatmap for better understating. Importing Graph libraries.



Checking dataset columns name.



**Summary Statistics**

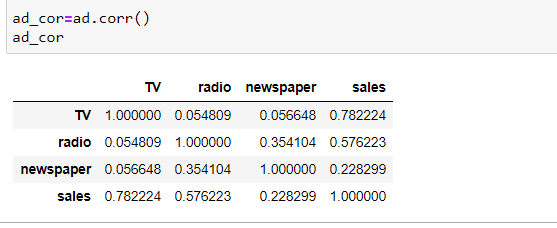


There is huge difference between 75% and maximum in column newspaper and sales, so possibly outliers present.

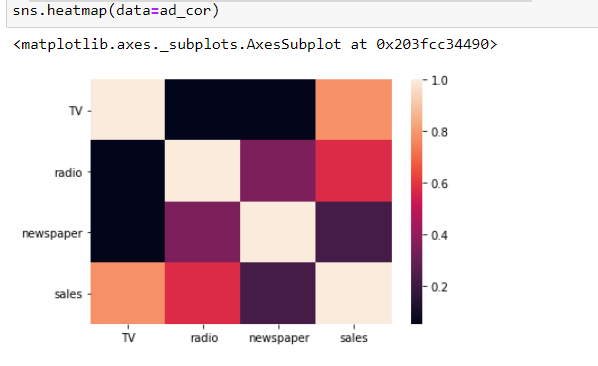
The minimum sales are 1.60 and maximum is 27.

Std is maximum in TV.

Checking correlation between two columns. Also Use heatmap to check correlation for better understanding.



We can see correlation between two columns using above data.

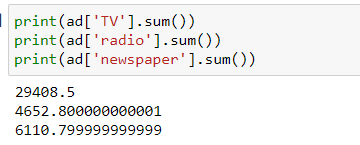


there is little high correlation between tv and sales, and very low correlation between newspaper and sales.

**Data Visualization**

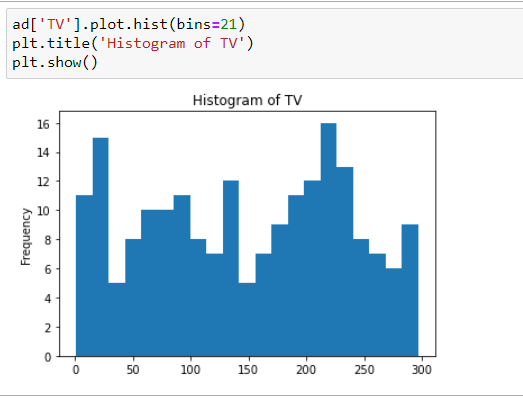
**Univariate Analysis**

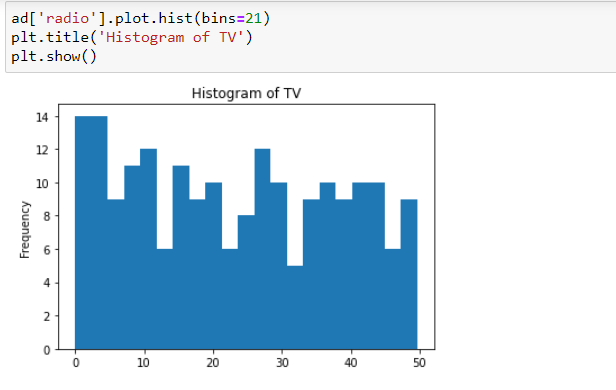
Checking total amount spent TV, radio and newspaper.

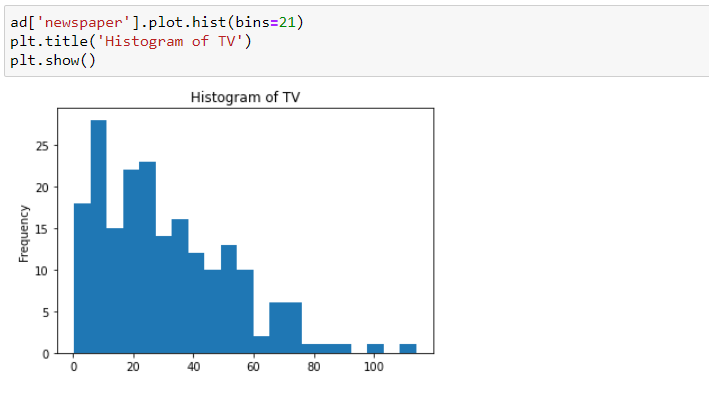


Maximum amount spent on TV, afterward radio and lowest amount spent on newspaper.

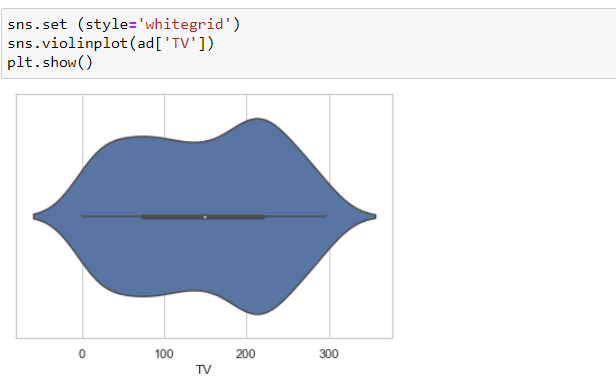
Now will use Histogram plot to visualize distributions of data. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range.



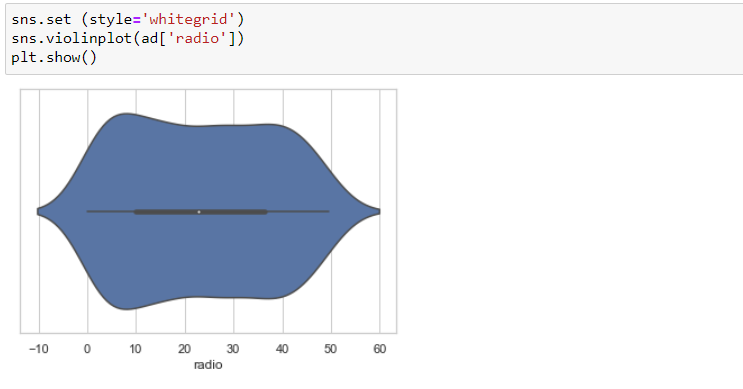




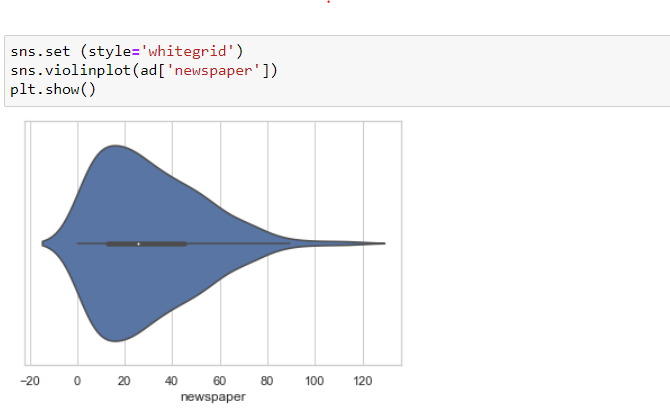
Now will use violin plot to visualise the distribution of the data and its probability density. This chart is a combination of a Box Plot and a Density Plot that is rotated and placed on each side, to show the distribution shape of the data.



Average 180 to 240 ((in thousands of dollars)) spent on TV advertising for most products.



Average 5 to 20 ((in thousands of dollars)) spent on radio advertising for most products.

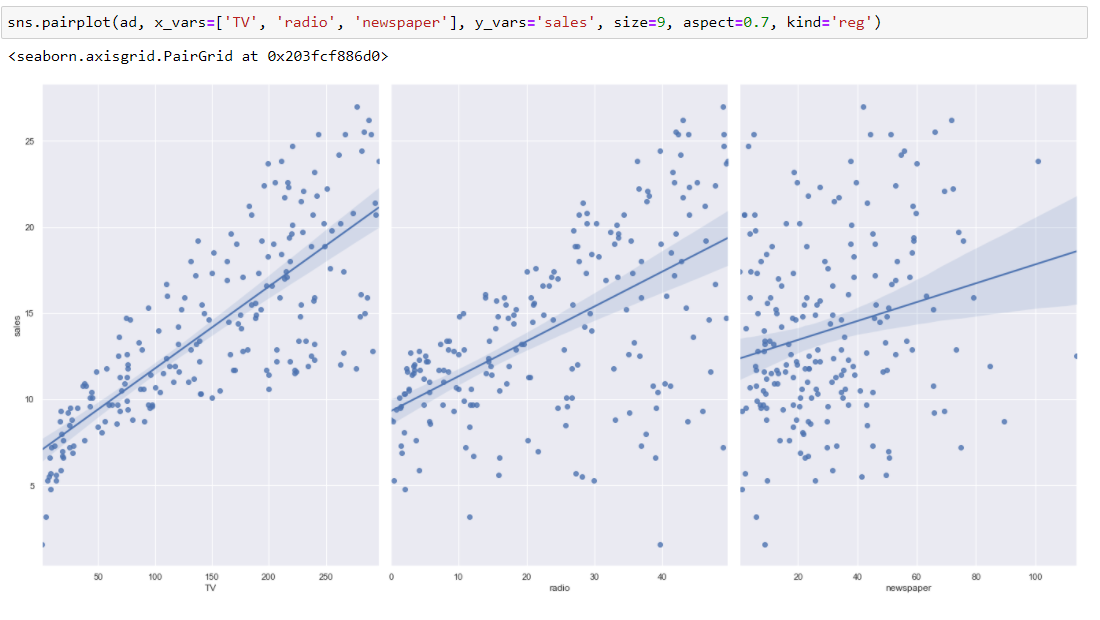


Average 10 to 30 ((in thousands of dollars)) spent on newspaper advertising for most products.

# Bivariate Analysis

Now will use pair plot to visualize relationships between each variable.

It produces a matrix of relationships between each variable in data for an instant examination of data. It can also be a great jumping off point for determining types of regression analysis to use.



There is positive relation between TV and sales, the graph shows more amounts spent on TV, the sales continuously increase.

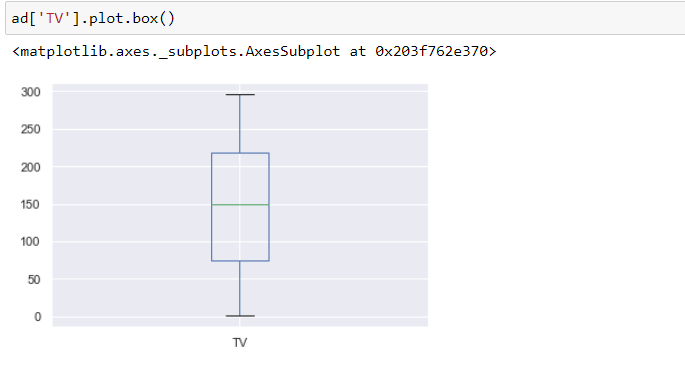
There is very low relation between radio and sales, the graph shows more amounts spent on radio, the sales not increase.

There is very low relation between newspaper and sales, the graph shows more amounts spent on radio, the sales not increase.

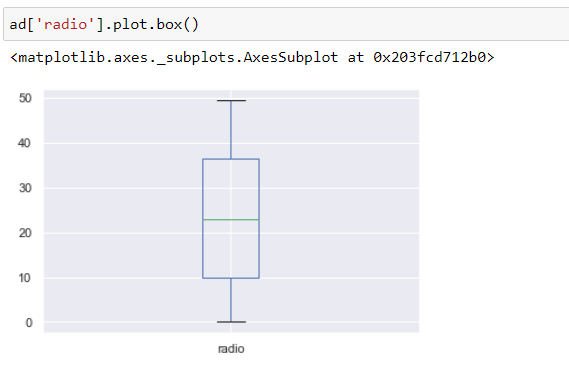
# Plotting Outliers

Now plot outliers using box plot.

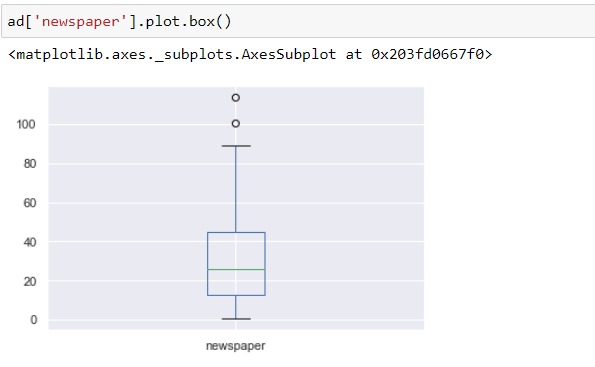
A boxplot is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”). It can also tell if data is symmetrical, how tightly data is grouped, and if and how your data is skewed.



No outliers present in TV column.



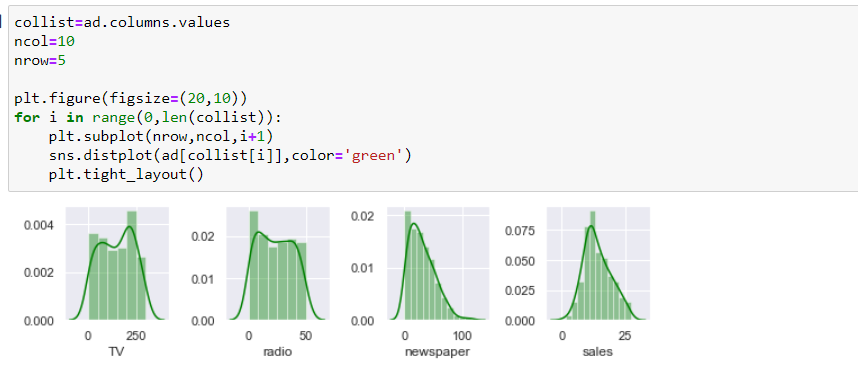
No outliers present in radio column.



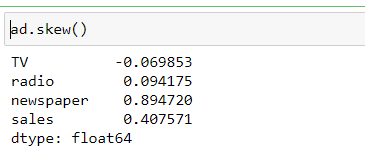
Possibly outliers present in newspaper column. We will remove outliers using Zscore value later on.

# Check the distribution of Skewness

Now checking distribution of skewness via for loop using distplot

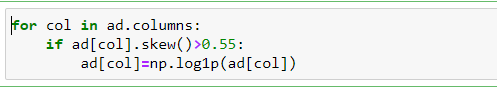


Checking skewness of data.

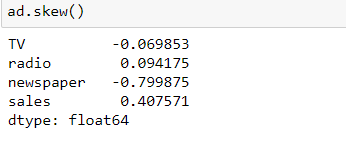


Now treating skewness via log method.

The log-transformation does remove or reduce skewness of data.



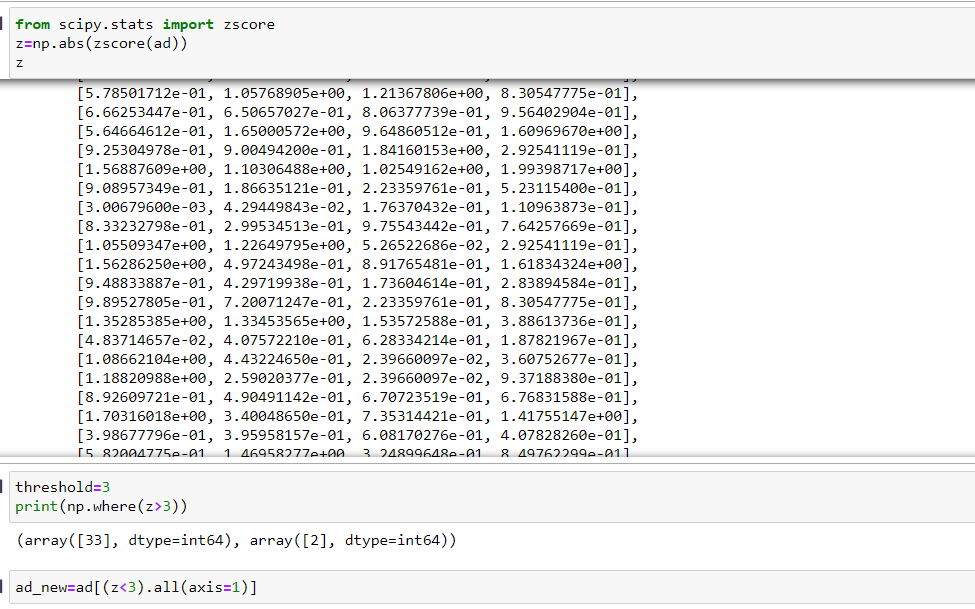
Checking skewness of data again.



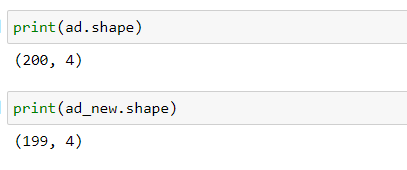
Now we can see the highly skewed data treated via log method.

**Removing Outliers**

Will remove outliers using Zscore value.



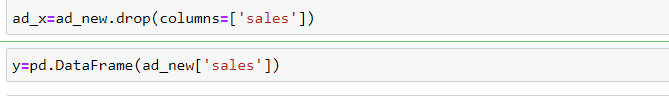
Now will check shape of data.



One row removed from dataset due to outliers.

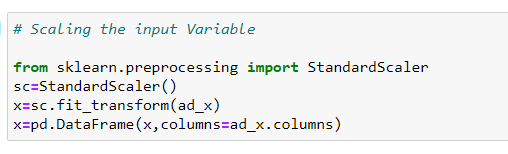
# Model Training

First will separating data into input and output variables. Later on will make input and output variables into train and test data.

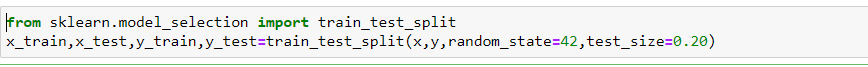


Now scaling the input Variable using Standard Scaler.

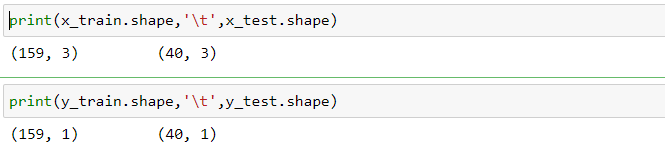
Standard Scaler transforms the data in such a manner that it has mean as 0 and standard deviation as 1. Standardization is useful for data which has negative values. It arranges the data in a standard normal distribution.



Now make input and output variables into train and test data. Will use 80% data for training and 20% data for testing purpose. Also importing train\_test\_split from sklearn.model\_selection.

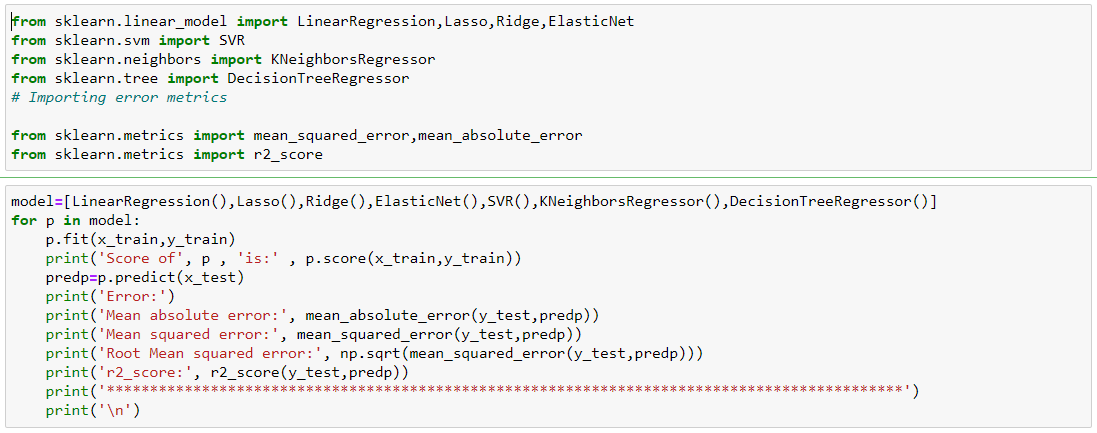


Now checking shape of train and test data.

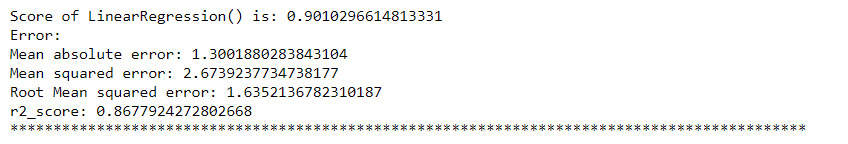


159 rows use for training the data and 40 rows use for testing data.

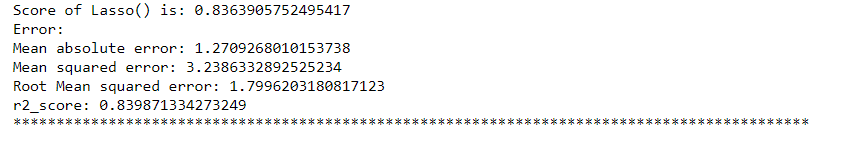
Now importing model libraries and using several algorithms via for loop. Will fit data into x\_train and y\_train and predict the x\_test data. Also will check mean absolute error, mean squared error, root mean squared error and r2 score.



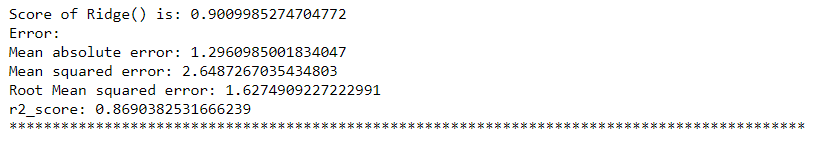
LinearRegression



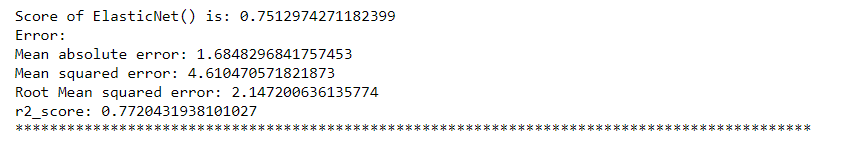
Lasso



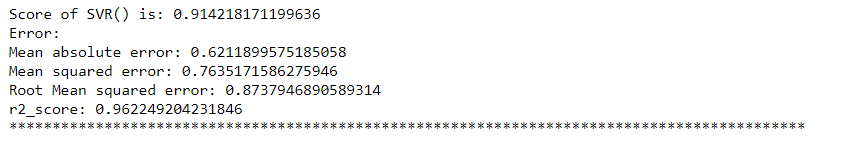
Ridge



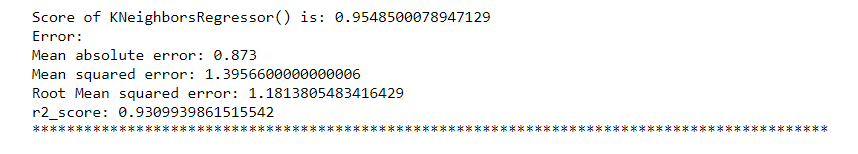
ElasticNet



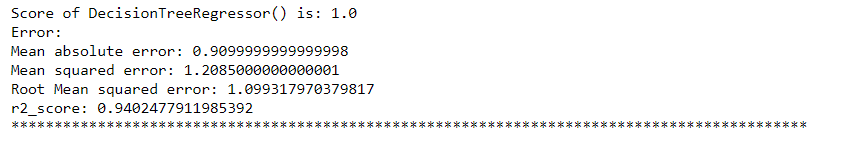
SVR



KNeighborsRegressor



DecisionTreeRegressor



The DecisionTreeRegressor provide best score and r2 score. Now use hyperparameter tuning to find out best result.

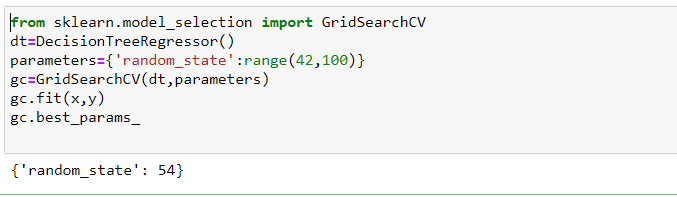
**DecisionTreeRegressor**

What us DecisionTreeRegressor?

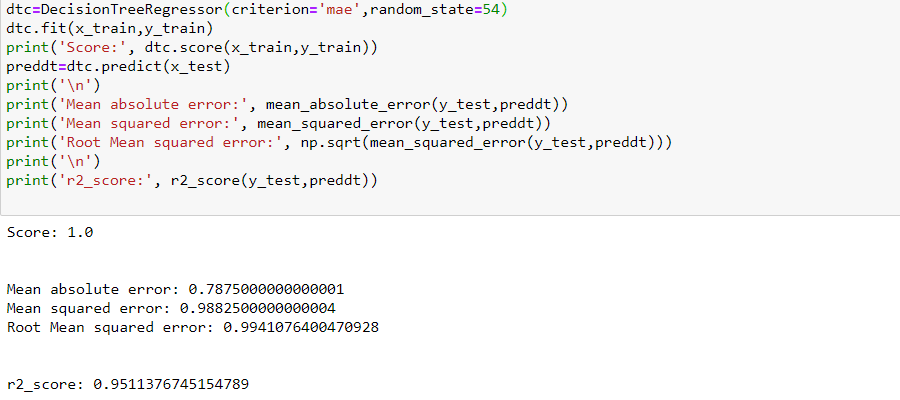
Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Will use GridsearchCV from sklearn to find out best parameter.

a hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters are derived via training



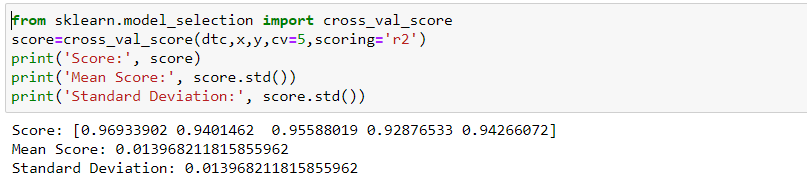
Above data shows random\_state 54 provide best result for DecisionTreeRegressor. Now again fit data into x\_train and y\_train and predict the x\_test data. Also will check mean absolute error, mean squared error, root mean squared error and r2 score.



We can see using 54 as random\_state our score and r2 score increase.

Now will use cross validate score of DecisionTreeRegressor to check if our model work properly.

cross\_val\_score returns score of test fold where cross\_val\_predict returns predicted y values for the test fold. For the cross\_val\_score , using the average of the output, which will be affected by the number of folds because then it may have some folds which may have high error (not fit correctly).



The score is between 92 to 96 which are good and our model work properly.

# Using Ensemble Technique to boostup score

Will use RandomForestRegressor and AdaBoostRegressor so boostup socre.

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods.

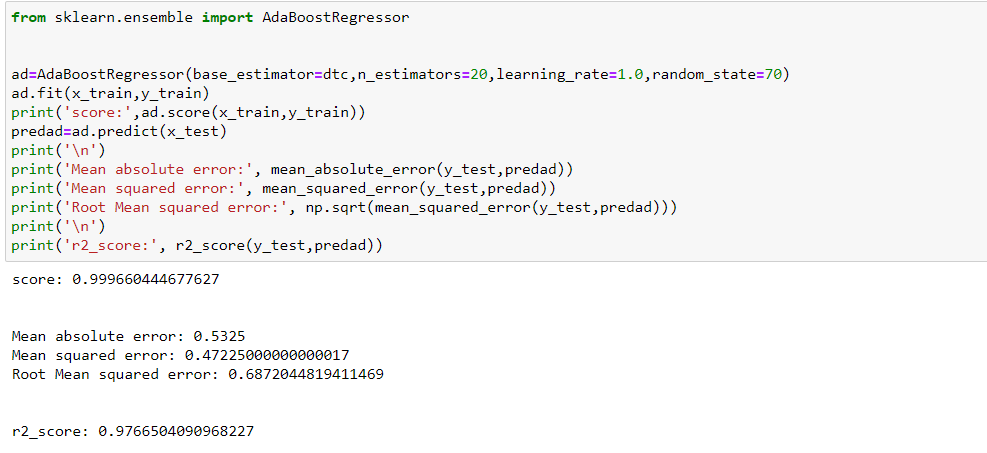


The randomforestregressor gave higher 2r score. Also lower the mean absolute error, mean squared error and root mean squared error.

RandomForestRegressor

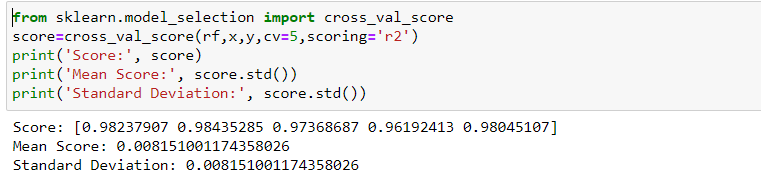
What is RandomForestRegressor?

A random forest regressor. A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.



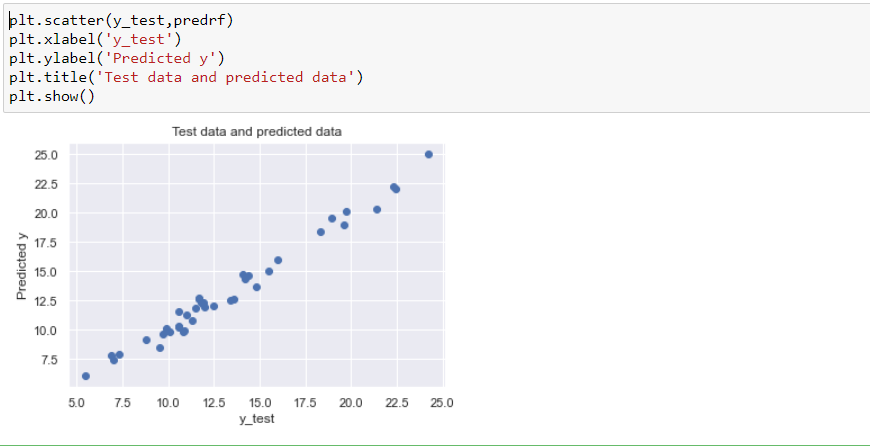
The AdaBoostRegressor also gave higher 2r score. Lower the mean absolute error, mean squared error and root mean squared error but not more than randomforestregressor. So will use randomforestregressor as final model.

Now use cross validate of RandomForestRegressor



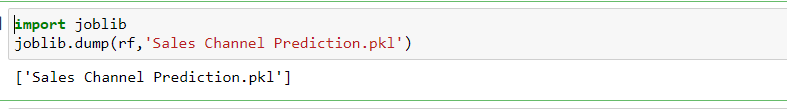
The score is between 96 to 98 which are good and our model work properly.

Now plotting scatter plot between test data and predicted data for RandomForestRegressor.



Above graph shows RandomForestRegressor model working fine and it is not under fitted and over fitted.

Saving the RandomForestRegressor as final model using joblib library.



**Summary**

We started with importing libraries and load dataset. After that check null values, see statistics summary, check correlation between two columns, performed univariate and bivariate analysis. During this process used seaborn and matplotlib libraries to do best visualizations. During the data pre-processing part, treated skewness via log method, removed outliers using zscore and scaling the input variables. Afterwards started training 9 different machine learning models, picked one of them (DecisionTreeRegressor) and applied hyperparameter tuning. Then use ensemble technics and find out best result with RandomForestRegressor. Then plot test data and predicted data and saved best model using joblib library.