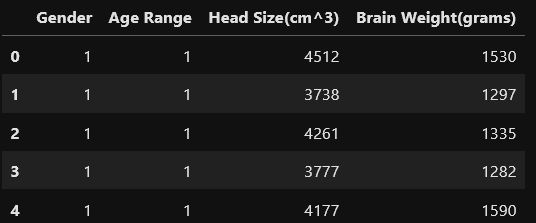
**Head Brain Dataset**

**Problem Defination:**

We are going to perform regression analysis on headbrain dataset. the input data is in csv format with 237 observations. it has total four columns (description is given below)

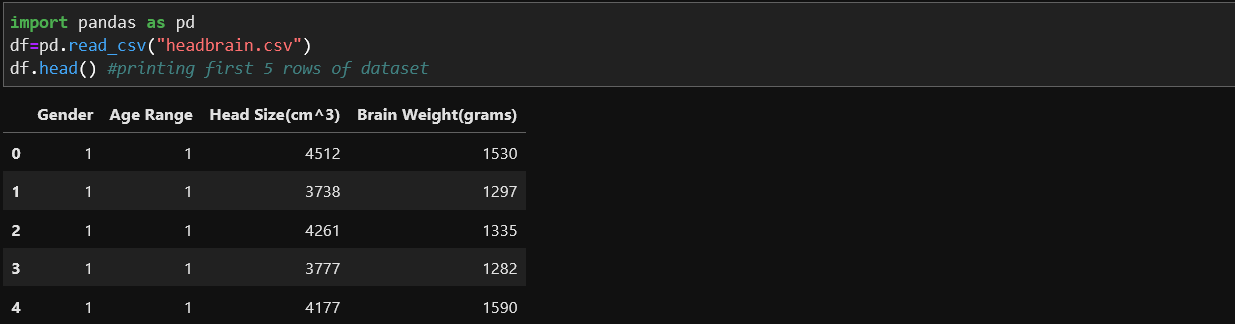
* Gender :1 for male, 2 for female
* Age Range : 1 for 20-46, 2 for above 46
* Head size : head size in cubic cm
* brain weight : brain weight in gm

Our target variable is "Brain Weight" which has continuous data so its a regression problem.



**Data Analysis:**

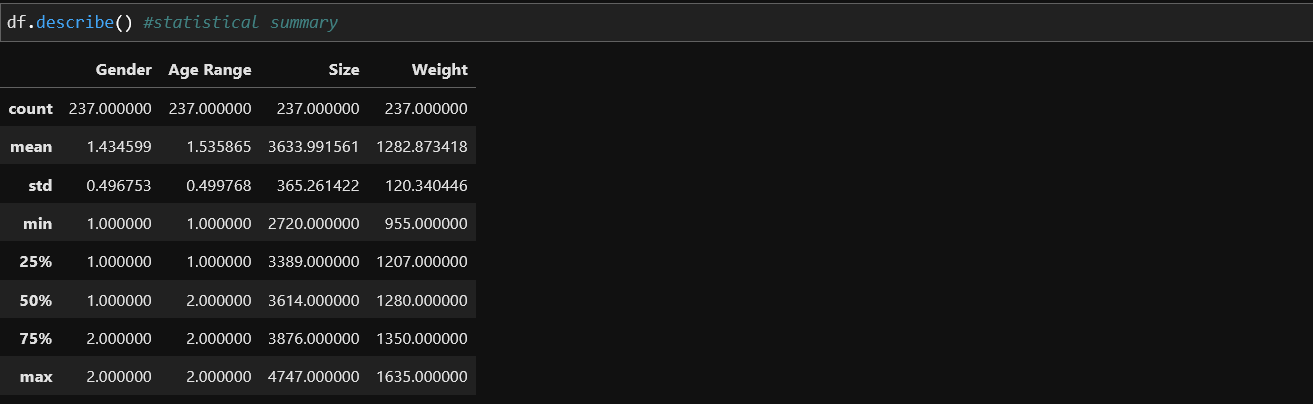
Loading the Dataset



Checking the shape of the dataset by using ‘df.shape’

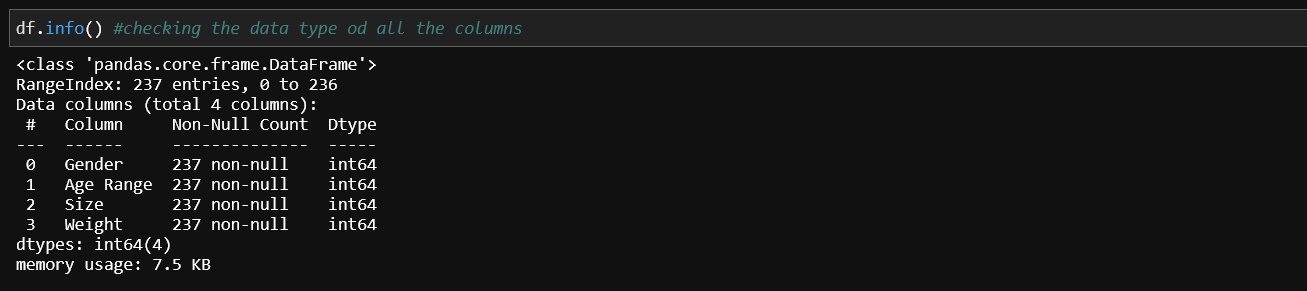
As output we get (237,4) which shows that the dataset has 237 rows and 4 columns.

Checking the Statical summary of the data using “df.discribe()” which provides us the following details of thr dataset



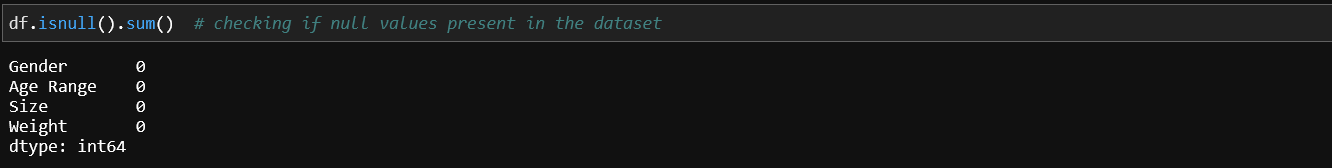
* Here we can see there are no missing values in the dataset
* No much difference observed between the mean and median so outliers may not present in the dataset.

Now we will check the datatype of the columns present in the dataset using df.info().



* Here we can see that no categorical data is present in the dataset therefore no encoding is required

Checking the Null values in the dataset using ‘df.isnull().sum()’ so that we can handle them by filling the null values using different methods.



* As shown in the output we can clearly see that there are no null values present in the dataset.

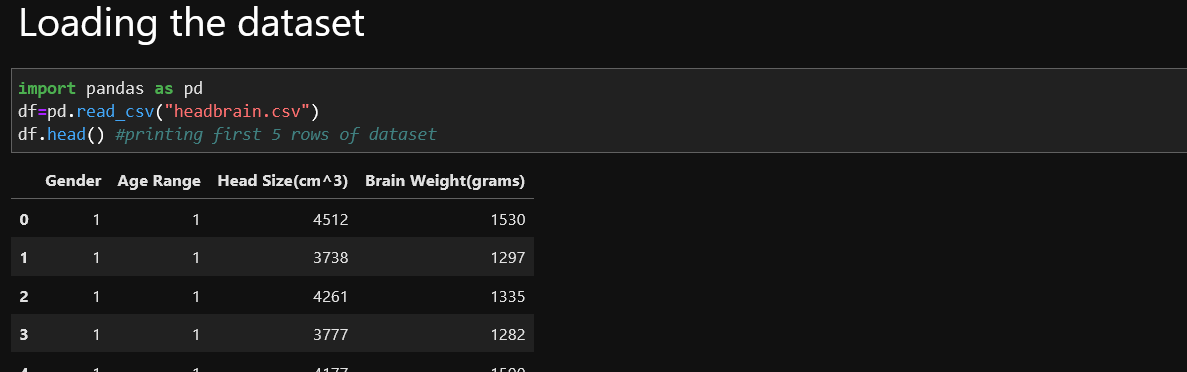
**EDA Concluding Remarks:**

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.

To share my understanding of the concept and techniques I know, I’ll take an example of white variant of [**Titanic** data set](https://archive.ics.uci.edu/ml/datasets/wine+quality) which is available on UCI Machine Learning Repository and try to catch hold of as many insights from the data set using EDA.

To starts with, I imported necessary libraries (for this example pandas, numpy, matplotlib and seaborn) and loaded the data set.



To take a closer look at the data took help of “.head()”function of pandas library which returns first five observations of the data set. Similarly “.tail()” returns last five observations of the data set.

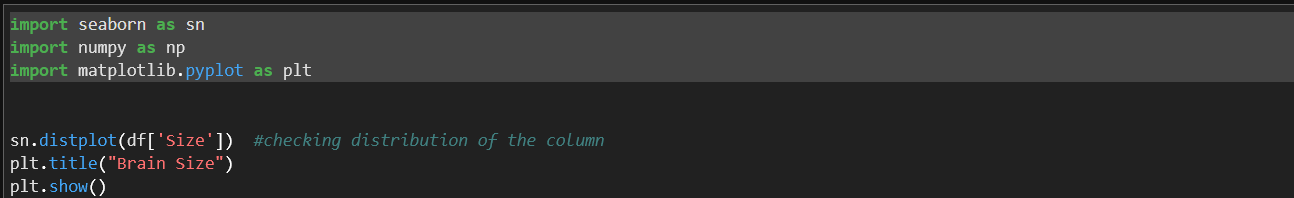
Let’s now explore data with beautiful graphs. Python has a visualization library, [Seaborn](https://seaborn.pydata.org/) which build on top of matplotlib. It provides very attractive statistical graphs in order to perform both [Univariate](http://www.statisticshowto.com/univariate/) and [Multivariate analysis](http://www.camo.com/multivariate_analysis.html)

**Univariate Analysis:**

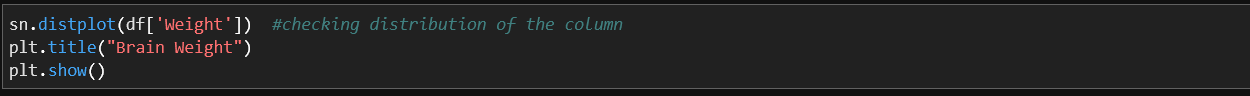
Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn't deal with causes or relationships (unlike regression ) and it's major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.

Importing the required files for analysis i.e. seaborn, matplotlib, pyplot etc.

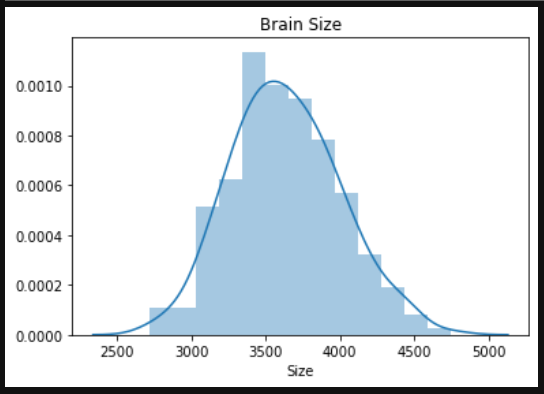
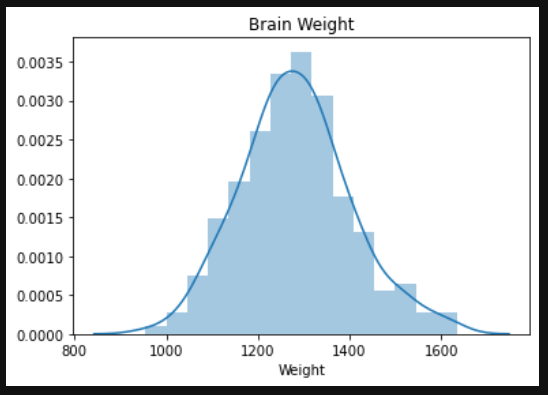
**DISTPLOT**



* Distplot for ‘Size’ Column.



* Distplot for ‘Weight’ Column.

**Brain Size Brain Weight**

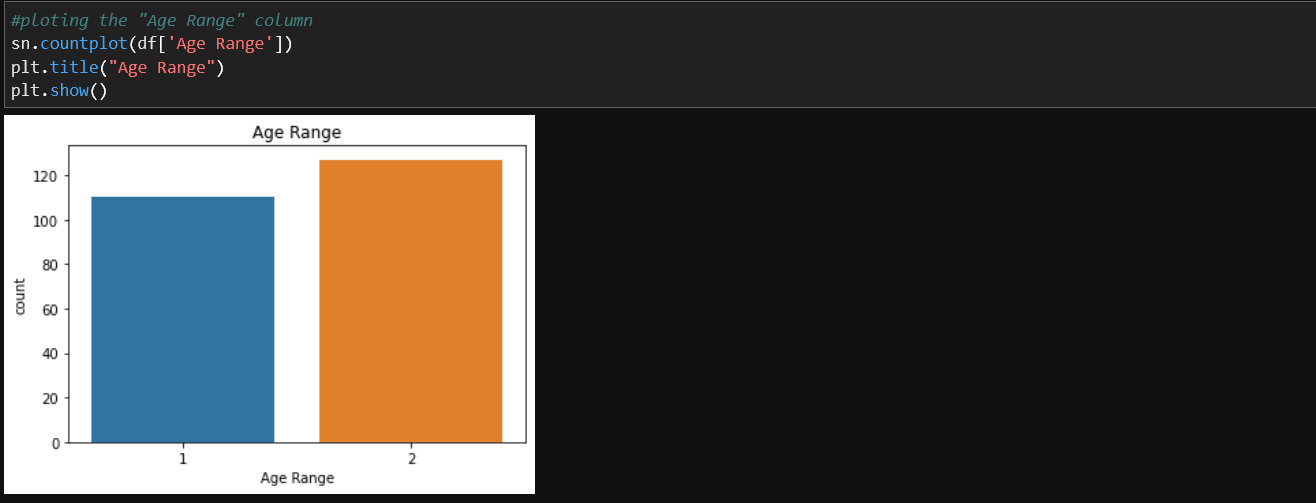
* As observed the data in both the columns are normally distributed.
* No skewness is present in the data.

**COUNTPLOT**

A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable. The basic API and options are identical to those for [barplot()](https://seaborn.pydata.org/generated/seaborn.barplot.html#seaborn.barplot), so you can compare counts across nested variables.

****

* As observed the Gender column has around 130 Males and about 100 females.



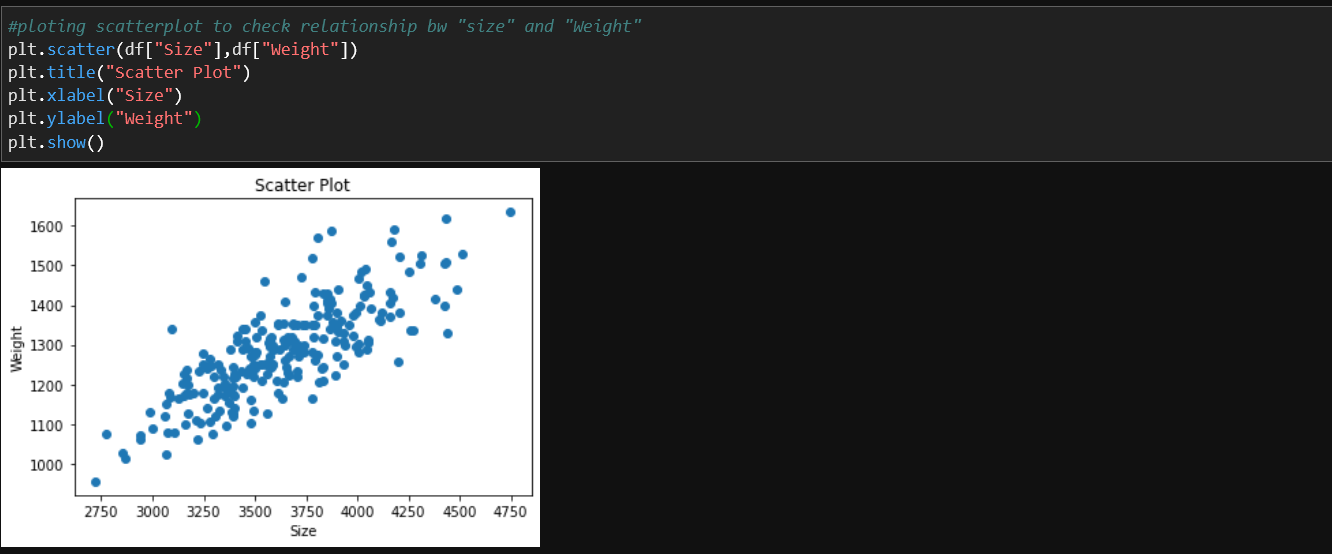
* The graph shows that the Majority of the age range is above 46.

**Bivariate Analysis:**

Bivariate analysis is one of the simplest forms of quantitative (statistical) analysis. It involves the analysis of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple hypotheses of association.

**SCATTERPLOT**

A scatter plot (also called a scatterplot, scatter graph, scatter chart, scattergram, or scatter diagram) is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data.



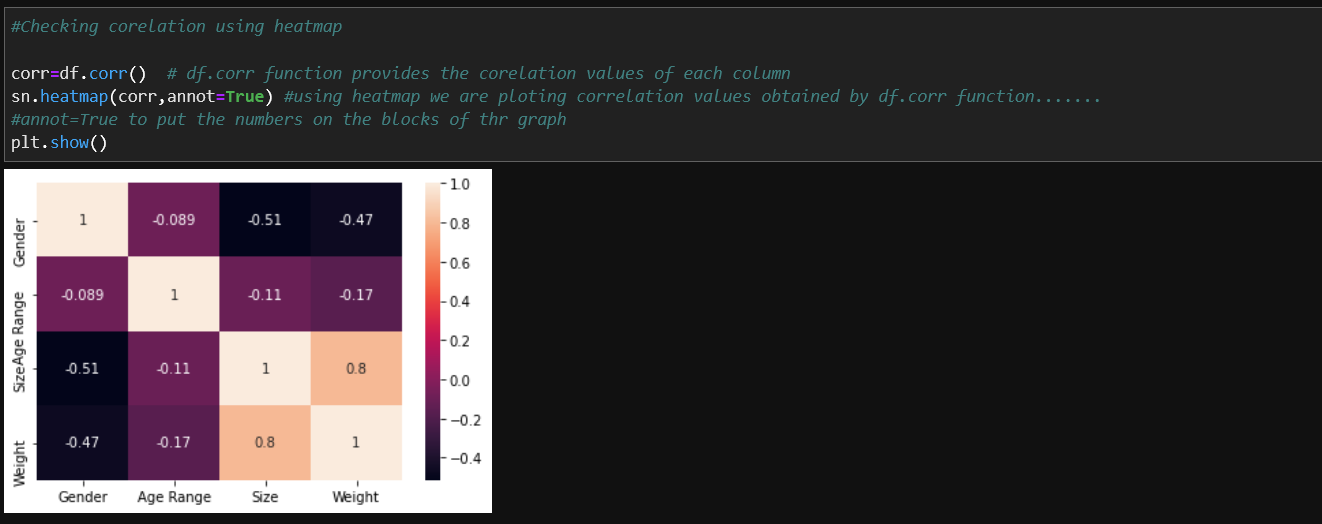
* We can see that there is a clear linear relationship (positive) between 'Size' and 'Weight' in above scatter plot.

**Multivariate Analysis:**

Multivariate analysis can be complicated by the desire to include physics-based analysis to calculate the effects of variables for a hierarchical "system-of-systems". Often, studies that wish to use multivariate analysis are stalled by the dimensionality of the problem. These concerns are often eased through the use of [surrogate models](https://en.wikipedia.org/wiki/Surrogate_model), highly accurate approximations of the physics-based code. Since surrogate models take the form of an equation, they can be evaluated very quickly. This becomes an enabler for large-scale MVA studies: while a [Monte Carlo simulation](https://en.wikipedia.org/wiki/Monte_Carlo_simulation) across the design space is difficult with physics-based codes, it becomes trivial when evaluating surrogate models, which often take the form of [response-surface](https://en.wikipedia.org/wiki/Response_surface_methodology) equations.

**HEATMAP**

A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors. The seaborn python package allows the creation of annotated heatmaps which can be tweaked using Matplotlib tools as per the creator's requirement.



* Very low correlation of 'Age Range' with target variable so we can drop this column
* Very high and positive correlation of "Size" with the target variable .
* High negative correlation of "gender" with the target variable but we will keep it as relation is strong.

Hence there in very low correlation of Age range with the target variable, it is ok to remove the Age Range Column.

* df.drop("Age Range",axis=1,inplace=True)

**Outliers Handling:**

An outlier is a data point in a data set that is distant from all other observation. A data point that lies outside the overall distribution of dataset. Many people get confused between Extreme values & Outliers.

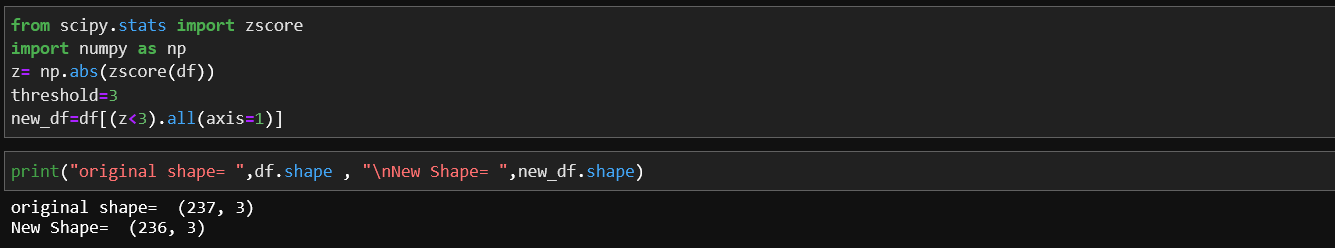
Checking the outliers in the data using Boxplot.

**BOXPLOT**

A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.



* There are outliers which can be removed using different methods, in this project we'll use inter quartile range to remove the outliers.



* After removing the ouliers we can see the difference between the shape of the dataframe before removing the outliers and after remving the outliers.
* 1 row is removed as outlier from the dataset.

**Pre-Processing Pipeline:**

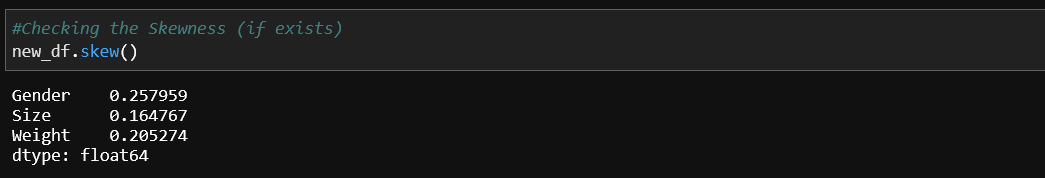
Data preprocessing is a predominant step in machine learning to yield highly accurate and insightful results. Greater the quality of data, the greater is the reliability of the produced results. Incomplete, noisy, and inconsistent data are the inherent nature of real-world datasets. Data preprocessing helps in increasing the quality of data by filling in missing incomplete data, smoothing noise, and resolving inconsistencies.

* Incomplete data can occur due to many reasons. Appropriate data may not be persisted due to a misunderstanding, or because of instrument defects and misfunctions.
* Noisy data can occur for a number of reasons (having incorrect feature values). The instruments used for the data collection might be faulty. Data entry may contain human or instrument errors. Data transmission errors might occur as well.

**Skewness**

Skewness refers to a distortion or asymmetry that deviates from the symmetrical bell curve, or normal distribution, in a set of data. ... A normal distribution has a skew of zero, while a lognormal distribution, for example, would exhibit some degree of right-skew.

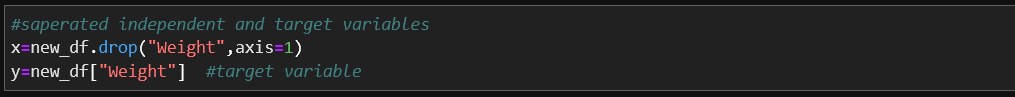
Checking the skewness in the data

****

* skewness is exceptable range (+/-0.5) so our data is not skewed.

**Model Training**

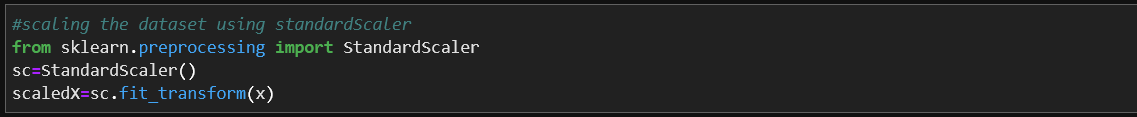
Saperating the independent and the target variable.



**Scaling**

Model can be bised to heigher values in dataset so its better to scale the dataset so that we can bring all the column in common range. there are two algorithms available for scaling. standardScaler and MinMaxScaler. we are going to use StandardScale here.

The idea behind StandardScaler is that it will transform your data such that its distribution will have a mean value 0 and standard deviation of 1. In case of multivariate data, this is done feature-wise (in other words independently for each column of the data).

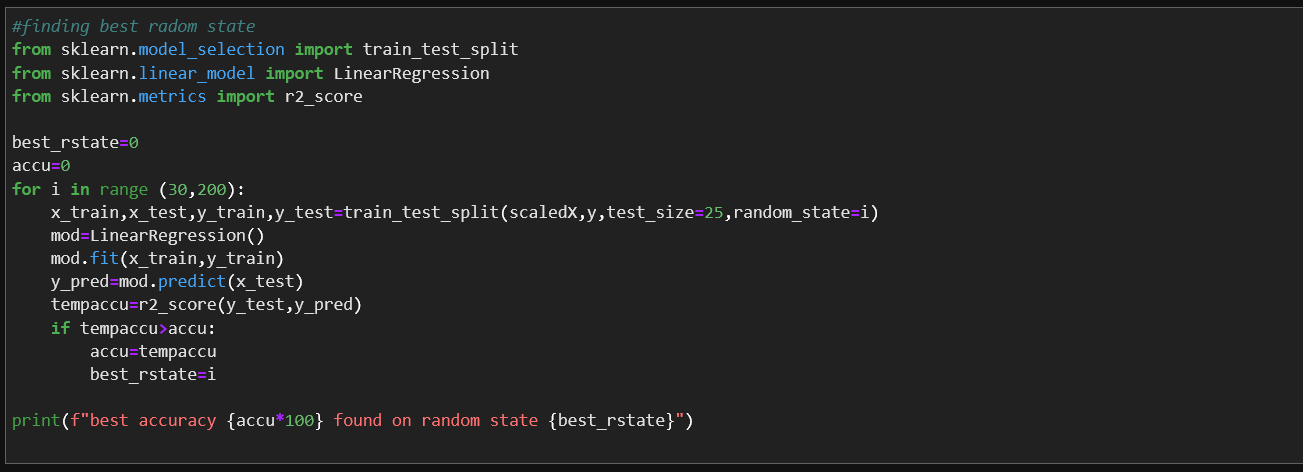
****

**Train-Test Split**

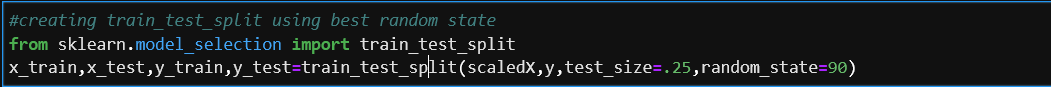
The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced

First we need to find the random state to perform train test split.



* best accuracy 84.66632012078443 found on random state 90
* we found best random state 90 and will be using it in train\_test\_split in next step.



**Building Machine Learning Models:**

Now, comes the fun part where we finally get to use the meticulously prepared data for model building. Depending on the data type (qualitative or quantitative) of the target variable (commonly referred to as the **Y** variable) we are either going to be building a classification (if **Y** is qualitative) or regression (if **Y** is quantitative) model.

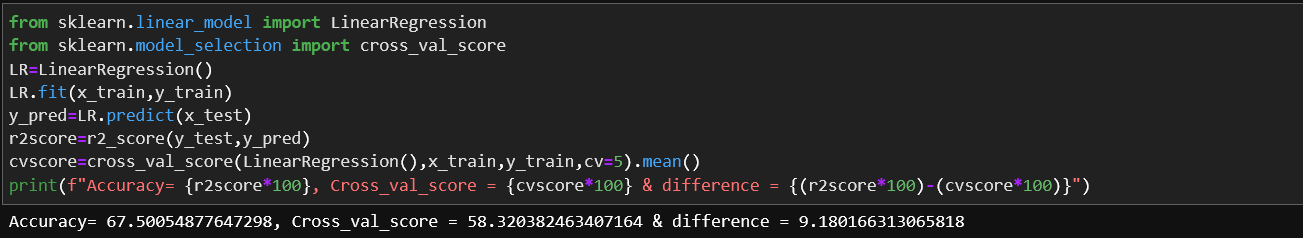
## Learning Algorithms

Machine learning algorithms could be broadly categorised to one of three types:

1. Supervised learning — is a machine learning task that establishes the mathematical relationship between input **X** and output **Y** variables. Such **X**, **Y** pair constitutes the labeled data that are used for model building in an effort to learn how to predict the output from the input.
2. Unsupervised learning — is a machine learning task that makes use of only the input **X** variables. Such **X** variables are unlabeled data that the learning algorithm uses in modeling the inherent structure of the data.
3. Reinforcement learning — is a machine learning task that decides on the next course of action and it does this by learning through trial and error in an effort to maximize the reward.

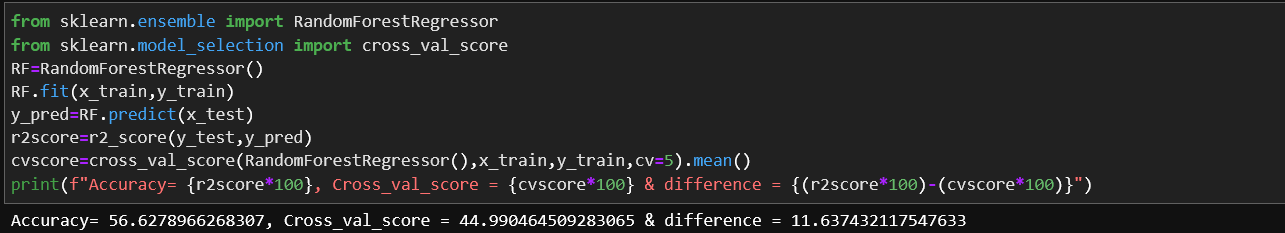
Finding the Best Model.

Linear Regression



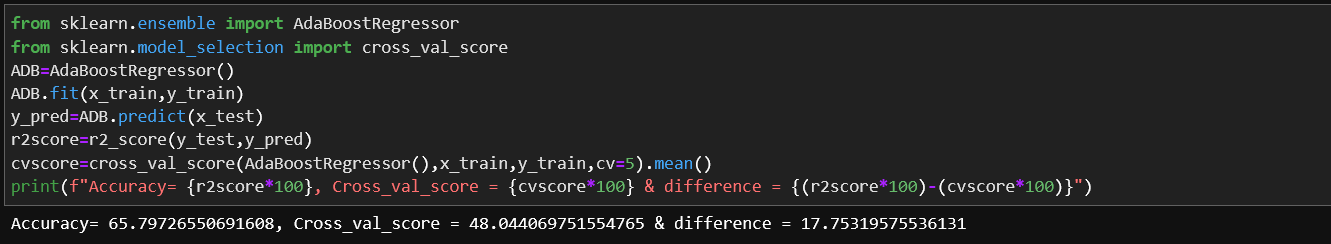
* Accuracy obtained using this method is 67.50%

RandomForestRegressor



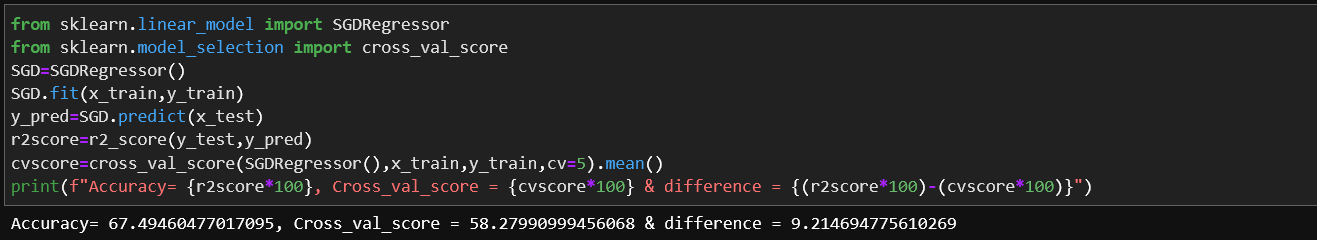
* Accuracy obtained using this method is 56.62%

AdaBooostRegressor



* Accuracy obtained using this method is 65.79%

SGDRegressor

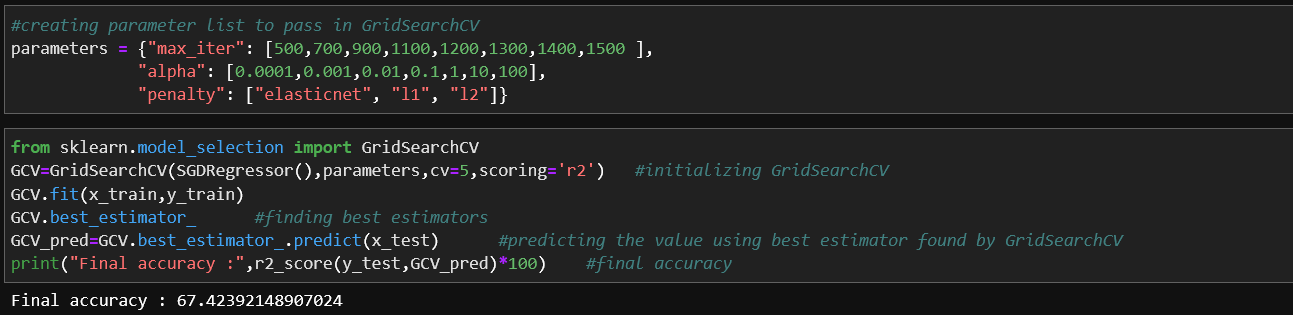


* Accuracy obtained using this method is 67.49%

LinearRegresson and SGDRegressor are best performing model with almost same accuracy and cross validation score. we can choose either of them as our best model so i am moving ahead with SGDRegressor .

**Hyperparameter Tuning**

Its a technique to find out the best parameter for our model to improve the accuracy.

****

* Model accuracy is not increase after 67% it is because of less data we need to collect more data to improve the accuracy further.

**Serialization**

****

**Concluding Remarks:**

After all the procedures from loading the dataset into the jupyter notebook to finding the best accuracy score we have completed our project with a accuracy score of 67.42%

we have got the 67.42% of accuracy with our models.