

**RATINGS PREDICTION**

Submitted by:

Ankita Prashant Chaudhari

**ACKNOWLEDGMENT**

References

1. Pang, B.; Lee, L. Opinion mining and sentiment analysis. Found. Trends Inf. Retr. 2008,2, 1–135.

2. Liu, B. Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. 2012,5, 1–167.

3. Khan, F.H.; Qamar, U.; Bashir, S. A semi-supervised approach to sentiment analysis using revised sentiment

strength based on Senti Word Net. Knowl. Inf. Syst. 2017,51, 851–872.

4.Khan, F.H.; Qamar, U.; Bashir, S. e SAP: A decision support framework for enhanced sentiment analysis and

polarity classiﬁcation. Inf. Sci. 2016,367, 862–873.

**INTRODUCTION**

The following article describes the application of a range of supervised and unsupervised machine learning models to a dataset of product reviews in an effort to predict rating value. The desired output to an input of a text review is a “star” rating on a continuum from 1 to 5. Because we wish to produce a continuum of sentiment rather than just polarity, this project could be categorized as a regression variation of sentiment analysis. Our goal is to produce the most versatile and accurate model that handles the widest range of mixed and polarized sentiments expressed in reviews. On top of trying multiple model types, we analyzed the impact of seven different embeddings, each of which had different built-in considerations for information like word index, directional significance, context, and frequency. These embeddings ranged from pre-trained word embedding models like BERT and Word2Vec, embedding schemes computed from scratch, and simpler encoding options like Bag of Words and TF-IDF. We trained models including supervised boosting models from Light GBM and CatBoost as well as three different deep learning networks which were more successful.

The article will first describe the curation of the dataset utilized to train and test the models, describing all review selection methodologies, pre-processing steps, and feature engineering. We will then dive into some of the differences behind different embedding techniques used, their context for our project, and their impact on model score. Then, the next section will describe the technical meat of the article — each attempted model and its corresponding score/lessons learned. There are also some unique quirks of this project that result in some unavoidable shortcomings, which will be discussed afterward. The article will then conclude with overall results, major findings, and recommendations for improvement and future work.

Potential applications include auto-generated suggestions for rating sentiment, falsified review detection, or other use cases where regression-type sentiment analysis is useful.

**Business Problem Framing**

Framing the correct problem in real-world data science projects may involve more processes than one would think. As much as it is worth noting that a data scientist’s role varies from company to company, problem framing is a challenge in organizations yet to become data-driven.

**Conceptual Background of the Domain Problem**

We obtained our data from the [Customer Reviews Dataset](https://s3.amazonaws.com/amazon-reviews-pds/readme.html) provided by Public Datasets, which contains official reviews from shoppers at. Here are the columns of the dataset along with a brief description of each: 'Uniq Id', 'Crawl Timestamp', 'Product Name', 'Product Rating', 'Review Title', 'Review Asin', 'Review Rating', 'Review Author', 'Review Date', 'Review Text', 'Source'.

**Review of Literature**

In conclusion, my thesis was proven to be correct. Combining the formerly known data about each user’s similarity to other users with the sentiment analysis of the review text itself, does help improve the model prediction of what rate the user’s review will get.

**Motivation for the Problem Undertaken**

We have a client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. The reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don’t have a rating. So, we have to build an application which can predict the rating by seeing the review.

**Analytical Problem Framing**

Mathematical/ Analytical Modeling of the Problem

**1.What is Statistical Modeling and How is it Used?**

Statistical modelingis the process of applying statistical analysis to a dataset. A statistical model is a mathematical representation (or mathematical model) of observed data.

When [data analysts](https://www.northeastern.edu/graduate/blog/what-does-a-data-analyst-do/) apply various statistical models to the data they are investigating, they are able to understand and interpret the information more strategically. Rather than sifting through the raw data, this practice allows them to identify relationships between variables, [make predictions](https://www.northeastern.edu/graduate/blog/predictive-analytics/) about future sets of data, and visualize that data so that non-analysts and stakeholders can consume and leverage it.

“When you analyze data, you are looking for patterns,” says Mello. “You are using a sample to make an inference about the whole.”

## Important Statistical Techniques in Data Analysis

Before any statistical model can be created, an analyst needs to collect or fetch the data on a database, clouds, social media, or within a plain excel file. To do this, analysts must also have a solid grasp of data structure and management, including how and where data is stored, fetched, and maintained. Those working in this field should thus share a passion for facts and data, and understand the basics of data manipulation, as well.

Once it comes time to analyze the data, there are an array of statistical models analysts may choose to utilize. According to Mello, most common techniques will fall into the following two groups:

* Supervised learning, including regression and classification models.
* Unsupervised learning, including clustering algorithms and association rules.

### Regression Models

Data analysts use **regression models** to examine relationships between variables. Regression models are often used by organizations to determine which independent variables hold the most influence over dependent variables—information that can be leveraged to make essential [business decisions](https://www.northeastern.edu/graduate/blog/data-driven-decision-making/).

“The most traditional regression models that have been used for a long time are logistic regression, linear regression, and polynomial regression,” Mello says. “These are the most common.”

Other examples of regression models can include stepwise regression, ridge regression, lasso regression, and elastic net regression.

### Classification Models

**Classification** is a process in which an algorithm is used to analyze an existing data set of known points. The understanding achieved through that analysis is then leveraged as a means of appropriately classifying the data. Classification is a form of machine learning that can be particularly helpful in analyzing very large, complex sets of data to help make more accurate predictions.

“Classification models are a form of supervised machine learning which is often used when the analyst needs to understand how they got to a certain point,” Mello says. “They give you more than just an output; [they give you] more information that you can use to explain the results of the prediction to your boss or stakeholder.”

Some of the most common classification models include decision trees, random forests, nearest neighbor, and  Naive Bayes.

There are also the neural networking models that are more used in AI. “These are very powerful models, and they can make accurate predictions very well,” Mello says, “but you typically cannot explain what is happening behind the scenes.”

**Types of the Model:**

Here it is the regression modeling technique use because in this dataset **Review\_Rating** is target variable and in this columns all continuous values are present so use regression models.

**Problem Definition:**

We have a client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. The reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don’t have a rating. So, we have to build an application which can predict the rating by seeing the review.

**Data Analysis:**

The first and foremost step involves importing necessary libraries and packages and loading the dataset as a pandas dataframe. The platform provides the training and testing datasets separately. Therefore, we load both datasets for further analysis.

**Data Loading and Visualisations:**

The first and foremost step involves importing necessary libraries and packages and loading the dataset as a pandas dataframe. Data visualization is the graphical representation of information and data. By using [visual elements like charts, graphs, and maps](https://www.tableau.com/learn/articles/data-visualization/glossary), data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Our eyes are [drawn to colors and patterns](https://www.tableau.com/learn/whitepapers/tableau-visual-guidebook). We can quickly identify red from blue, square from circle. Our culture is visual, including everything from art and advertisements to TV and movies. Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we [quickly see trends and outliers](https://www.tableau.com/reports/business-intelligence-trends). If we can see something, we internalize it quickly. It’s storytelling with a purpose. If you’ve ever stared at a massive spreadsheet of data and couldn’t see a trend, you know how much more effective a visualization can be.

**Importing libraries**

We will start by importing the libraries we will require for performing EDA. These include NumPy, Pandas, Matplotlib, and Seaborn.

### Reading data:

We will now read the data from a CSV file into a Pandas DataFrame in this there are two dataset first is train dataset and second is test dataset.

Install This project requires anaconda python, because below libraries already available.

• Numpy

• Matplotlib

• Seaborn

• Skit-learn

• Pandas

Also need to have software.

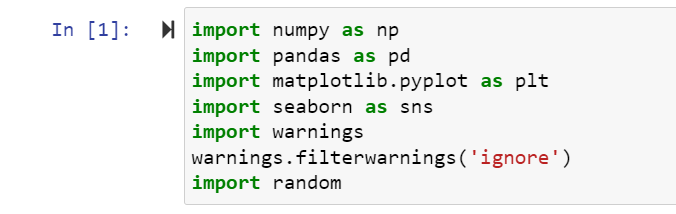
Install, run and execute a Jupyter notebook.

**EDA**:

There are no shortcuts in a machine learning project lifecycle. We can’t simply skip to the model building stage after gathering the data. We need to plan our approach in a structured manner and the exploratory data analytics (EDA) stage plays a huge part in that. I can say this with the benefit of hindsight having personally gone through this situation plenty of times. In my early days in this field, I couldn’t wait to dive into machine learning algorithms but that often left my end result hanging in the balance. I discovered, through personal experience and the advice of my mentors, the importance of spending time exploring and understanding my data.

## The Importance of Exploratory Data Analysis (EDA): There are no shortcuts in a machine learning project lifecycle. We can’t simply skip to the model building stage after gathering the data. We need to plan our approach in a structured manner and the exploratory data analytics (EDA) stage plays a huge part in that. I can say this with the benefit of hindsight having personally gone through this situation plenty of times. In my early days in this field, I couldn’t wait to dive into machine learning algorithms but that often left my end result hanging in the balance. I discovered, through personal experience and the advice of my mentors, the importance of spending time exploring and understanding my data.

**Loading dataset :**

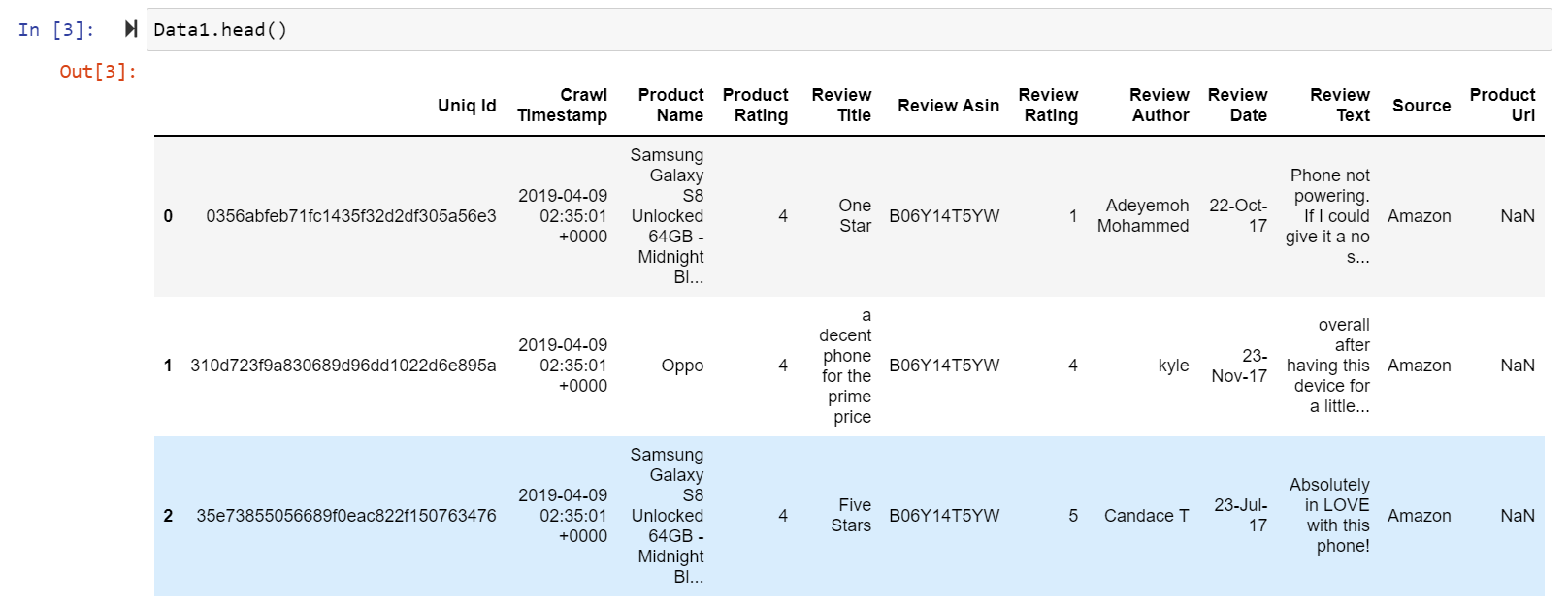
****

****

Here importing all necessary library and also load dataset.

**Data Sources and their formats**

Then need to read and load dataset. And Then this below code isusefor display all the columns in dataset. Let us have a look at how our dataset The structure and details of the data are given below in this there are dataset in which no of row and columns are present:



Here Review\_Rating is our Target Variable in dataset and in this dataset Review\_Rating are target variable and there was continues numeric values are present so this is **Regression Problem** so Need use Regression Algorithm.



Check the how much Columns and rows present in dataset using df.shape() is use to check the rows and columns count in this above dataset there are 2080 rows and 12 columns are present.

## Introduction

In Machine Learning, we use various kinds of algorithms to allow machines to learn the relationships within the data provided and make predictions based on patterns or rules identified from the dataset. So, regression is a machine learning technique where the model predicts the output as a continuous numerical value.

Regression analysis is often used in finance, investing, and others, and finds out the relationship between a single dependent variable(target variable) dependent on several independent ones. For example, predicting ratings, stock market or salary of an employee, etc are the most common regression problems.

### The algorithms we are going to cover are:

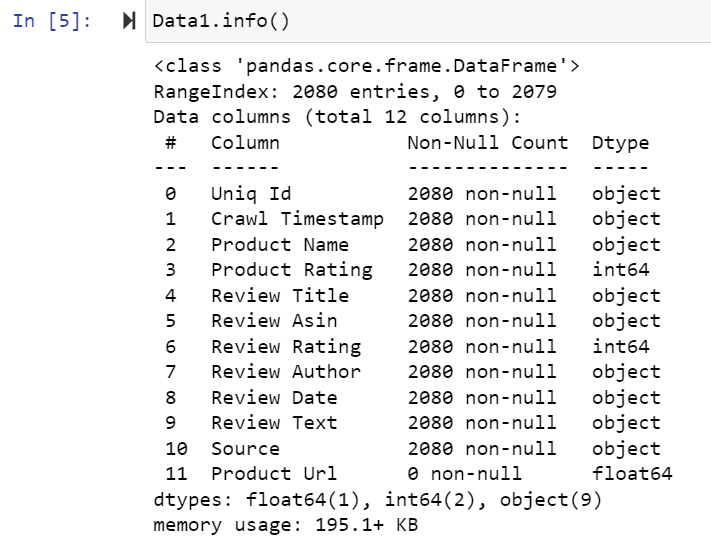
1. Linear Regression

2. Decision TreeRegressor

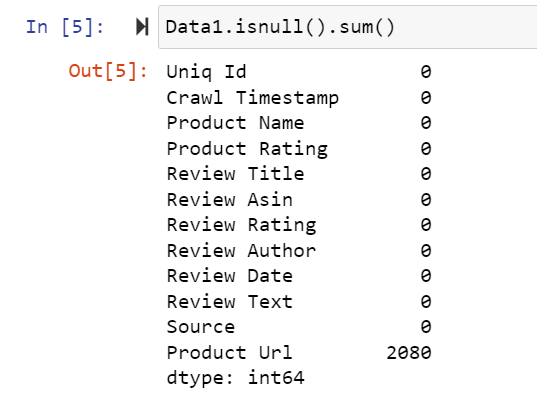
3. Support Vector Regression

4. Lasso Regression

5. Random Forest Regressor



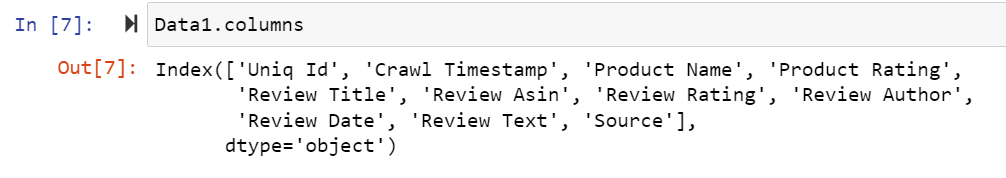
In above using df.info() is used to get the information all the columns in dataset and its data types. In above dataset there are 12 columns in dataset.



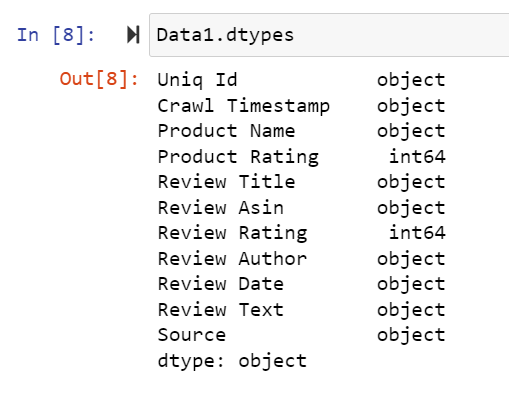
Check the how much missing values present in dataset using df.isnull() is use to check the missing values in dataset in this dataset there are missing values are present in this dataset product Url column.



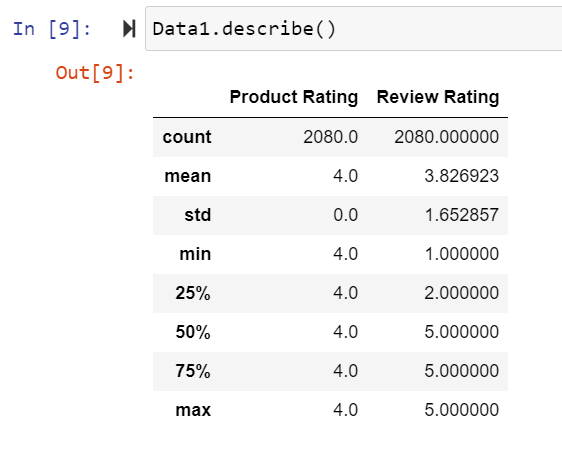
In above Product Url column is drop because all null values are present in this column.



In above df.colums is use to get all the column names in this dataset.



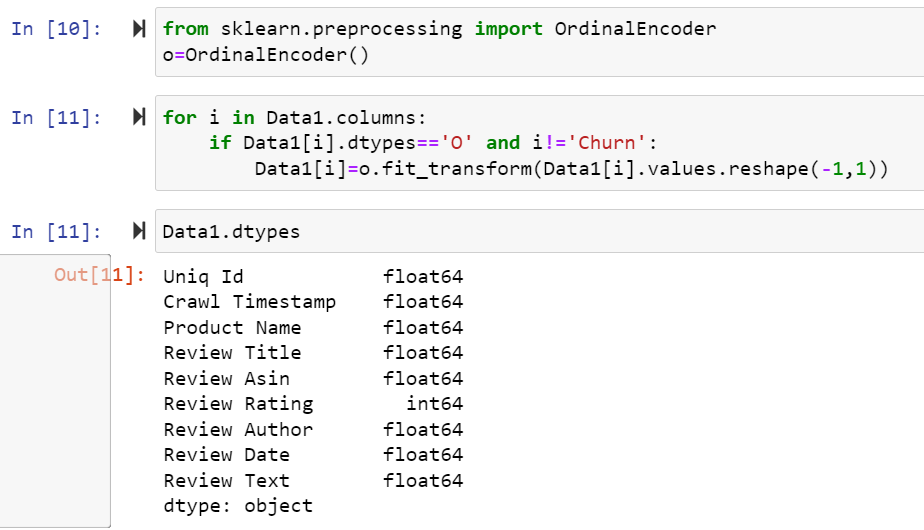
In above code df.dtypes is used to check the data types of all the column In above dataset.



Above describe the information of this dataset using df.describe().

**Data Preprocessing Done:**

Here we converted categorical data into numeric in dataset because without converting categorical data into numeric we cannot find out correlation of the dataset.

****

Here use label encoder to replace the categorical values into numeric in this dataset.

**State the set of assumptions (if any) related to the problem under consideration**

This study aims to analyse the accuracy of predicting house prices when using Multiple linear, Lasso, Ridge, Random Forest regression algorithms and Artificial neural network (ANN). Thus, the purpose of this study is to deepen the knowledge in regression methods in machine learning. In addition, the given datasets should be processed to enhance performance, which is accomplished by identifying the necessary features by applying one of the selection methods 2 to eliminate the unwanted variables since each house has its unique features that help to estimate its price. These features may or may not be shared with all houses, which means they do not have the same influence on the house pricing resulting in inaccurate output.

**Hardware and Software Requirements and Tools Used**

1. Software Requirements:

* 1. Coding Language: Python3, Python
  2. Coding software : Anaconda, Jupyter Notebook

1. Microsoft Office Word.
2. Snipping Tools (For Screenshots).
3. Microsoft Excel

**Non Functional Requirements:**

1: Platform Independent: The application would be platform independent if all the requirements are installed in the device.

2: Performance: The application should have better accuracy and should provide the information in less time.

3: Capacity: The capacity of the storage should be high so that large amount of data can be stored in order to train the model.

**Hardware Requirements:**

1 GB RAM.

200 GB HDD.

Intel 1.66 GHz Processor Pentium 4

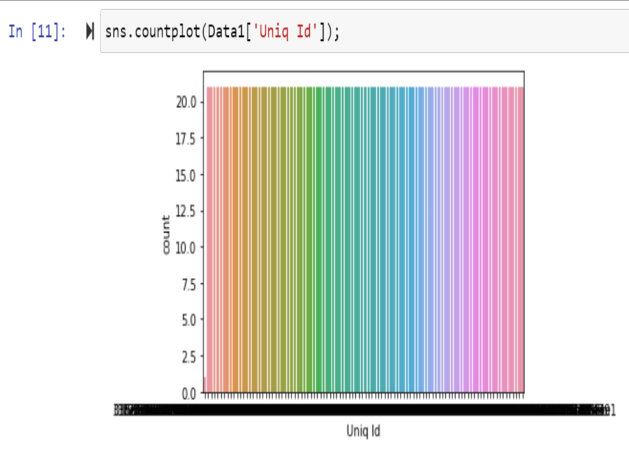
**Graphical representation**

### We will start with Univariate Analysis. We will be using a bar graph for this purpose.

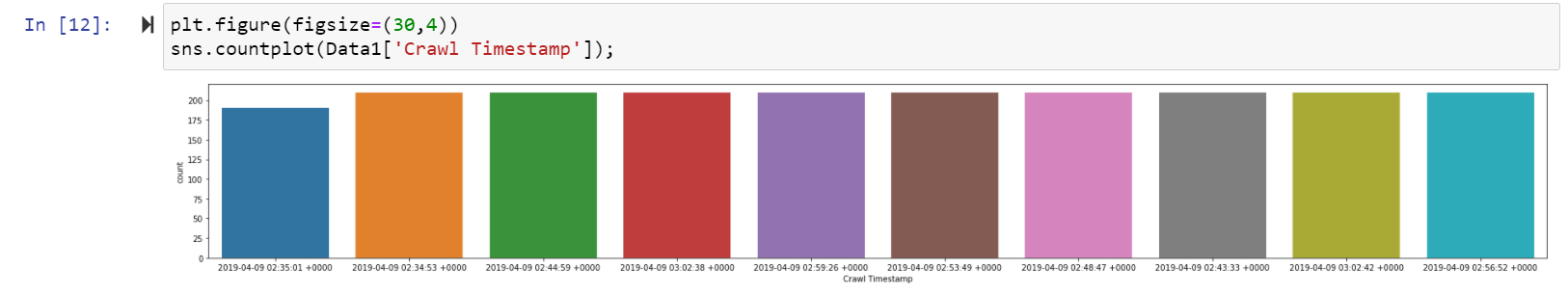
1. Univarient Analysis.
2. Bi Varient Analysis
3. Multivarient Analysis

**Univarient Analysis**:

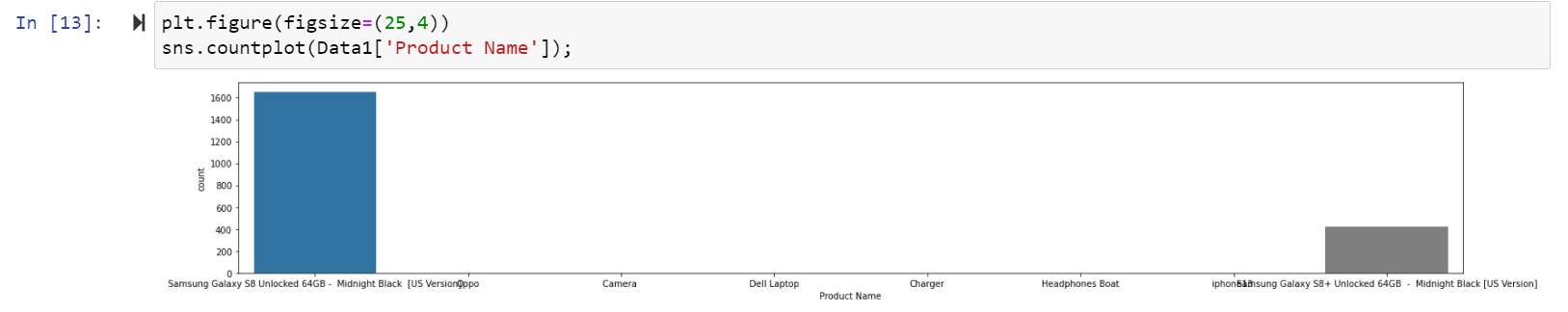
Univariate analysis **explores each variable in a data set**, separately. It looks at the range of values, as well as the central tendency of the values. It describes the pattern of response to the variable. It describes each variable on its own.



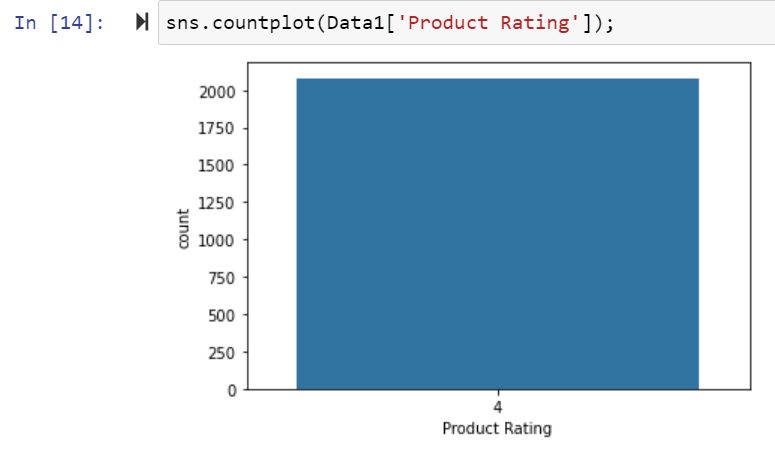
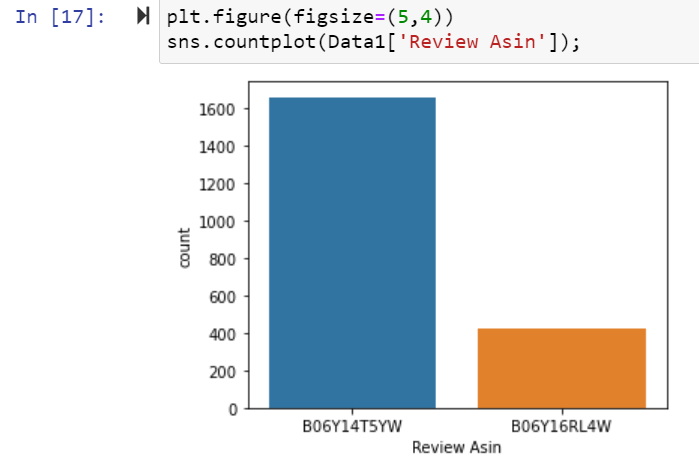
Univarient analysis of ‘Uniq Id’ column in dataset.



Univarient analysis of ‘Crawl Timestamp’ column in thisdataset.



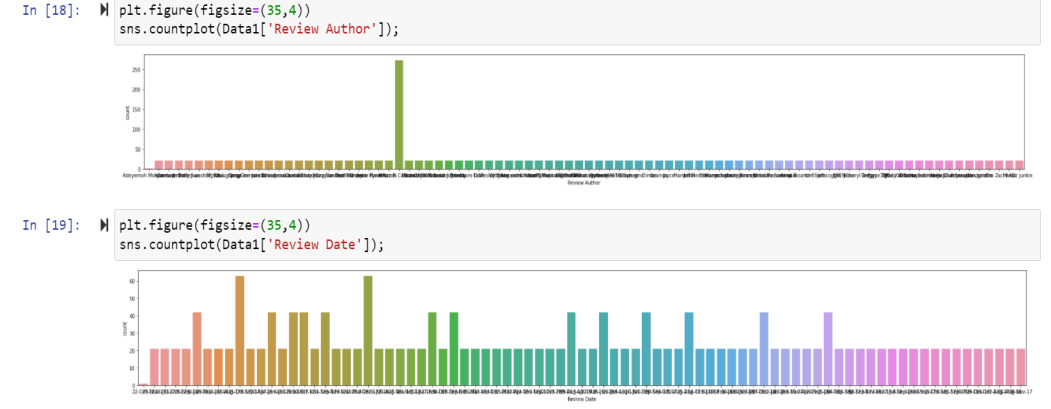
Univarient analysis of ‘Crawl Timestamp’ column in this dataset.

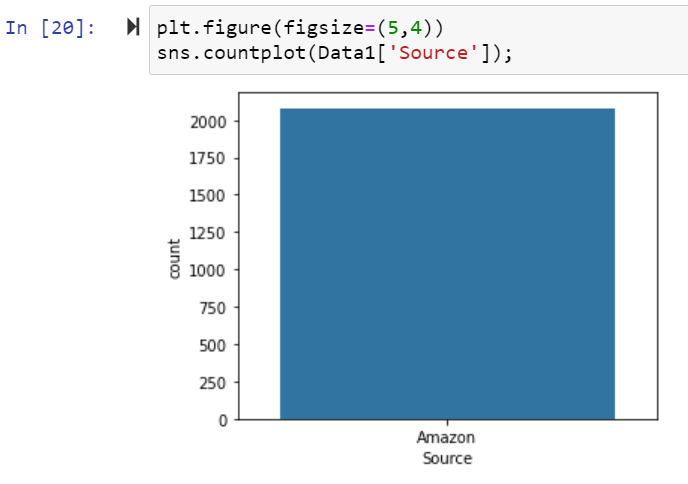
Univarient analysis of ‘Product Rating’ and ‘Review Asin’ column in this dataset.



Univarient analysis of ‘Review Rating’ column in this dataset.



Univarient analysis of ‘Review Author’ and ‘Review Date’ column in this dataset.

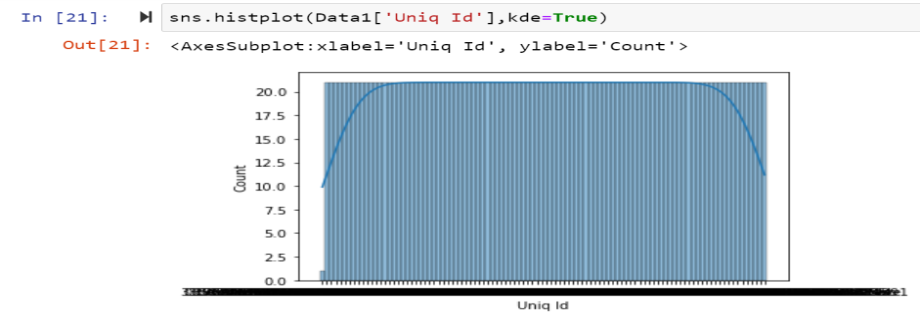


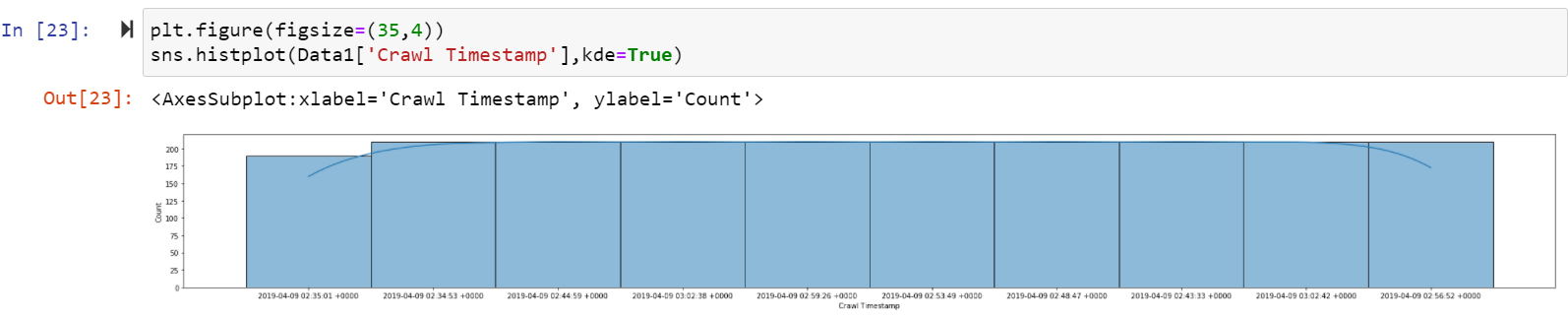
Univarient analysis of ‘Source’ column in this dataset.

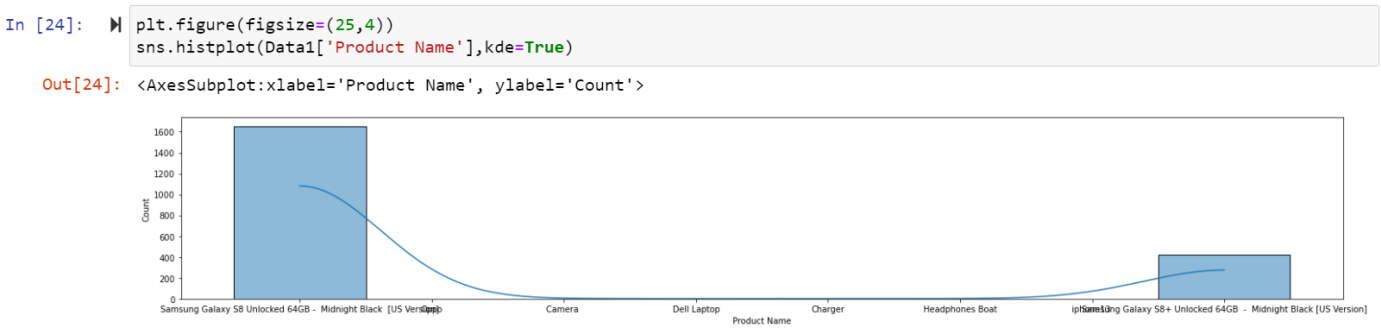
**Checking Distribution of columns in train datset they are normally distributed or not:**

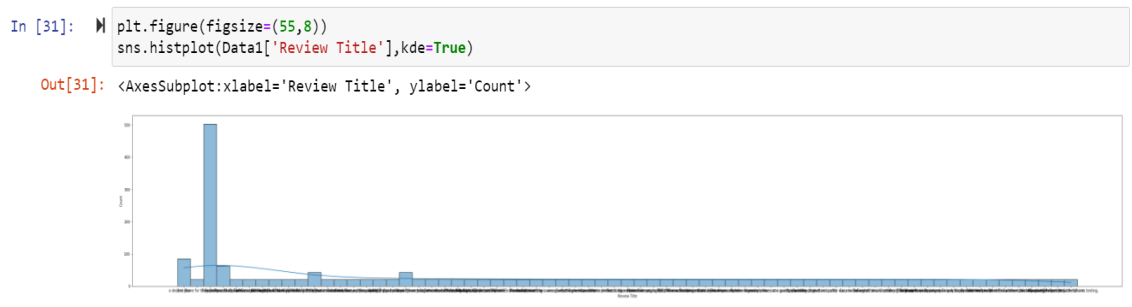
Now Check The all the columns data are normally Distributed or not:

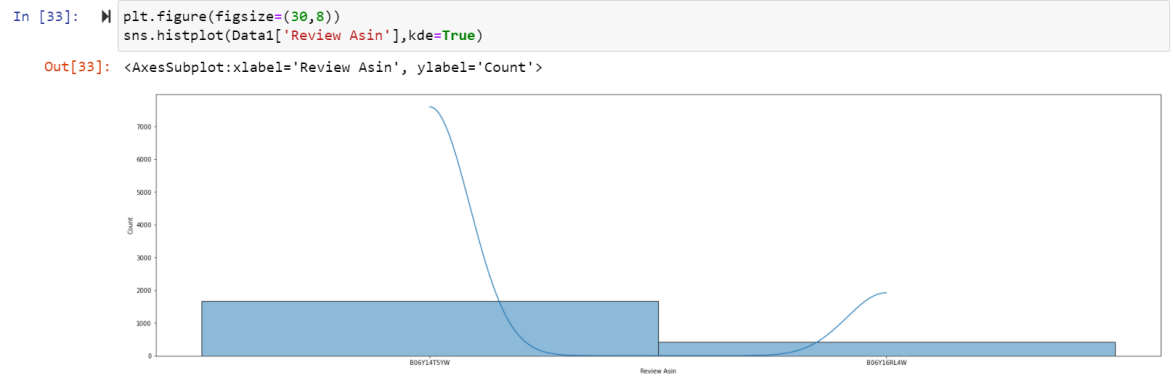
Normally distributed data, there is a constant proportion of data points lying under the curve between the mean and a specific number of standard deviations from the mean. Thus, for a normal distribution, almost all values lie within 3 standard deviations of the mean.

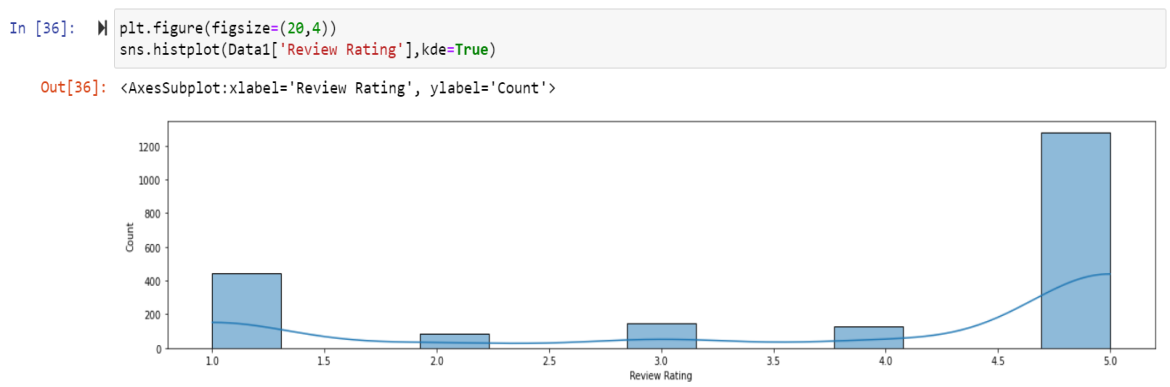


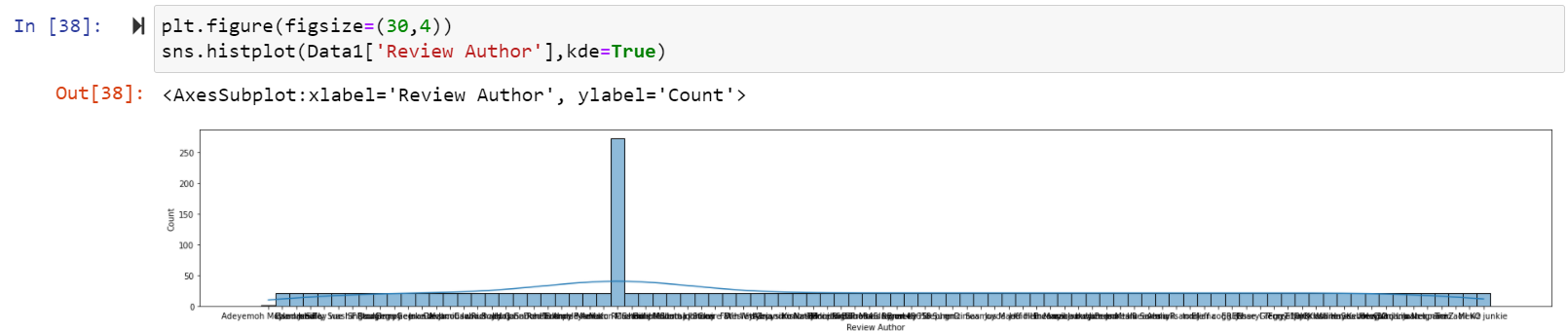


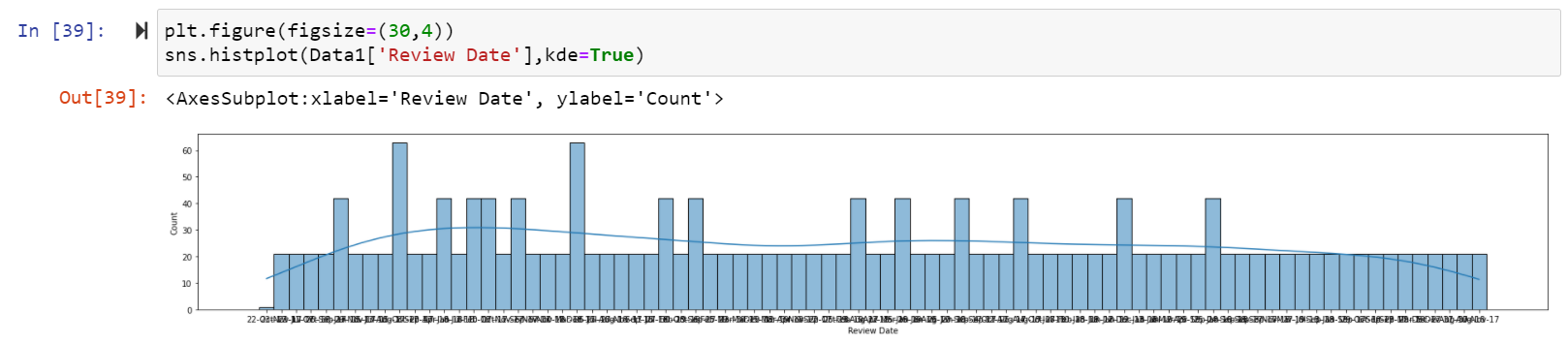


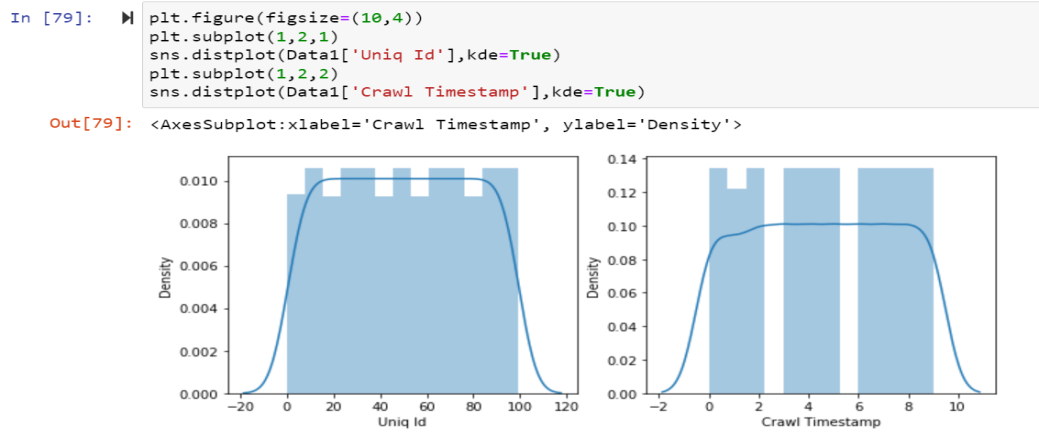


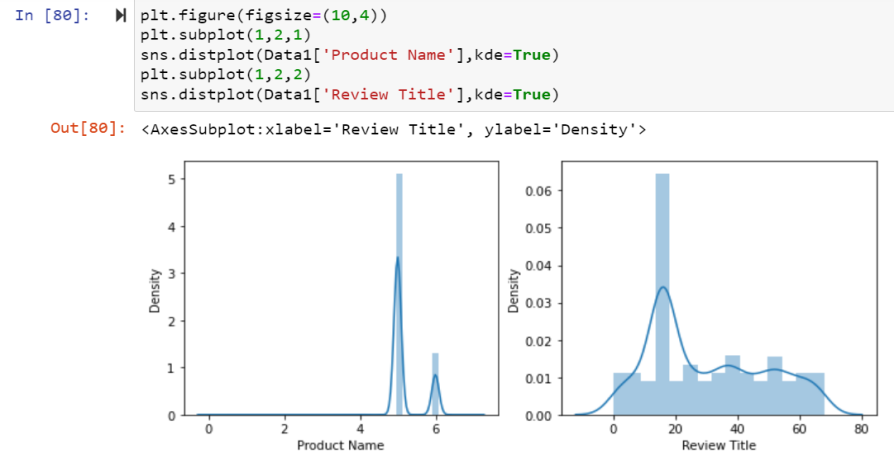




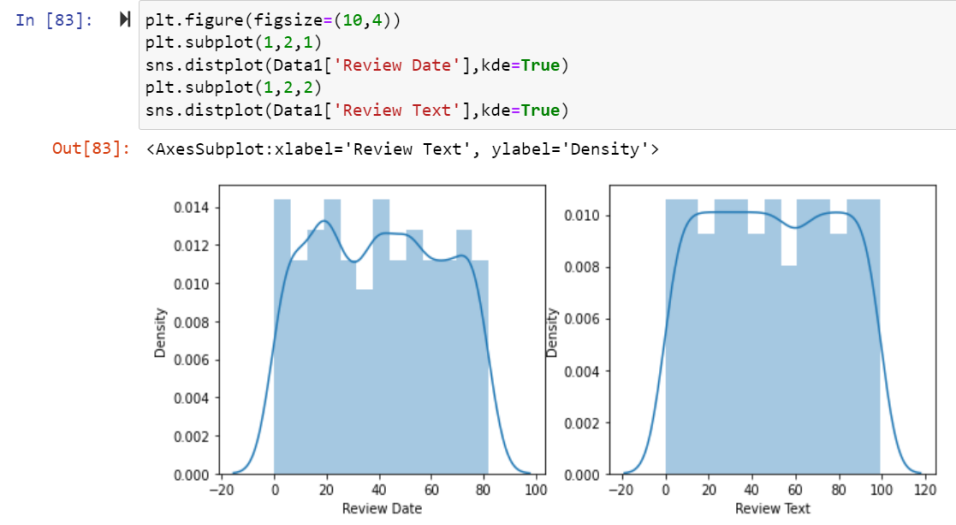


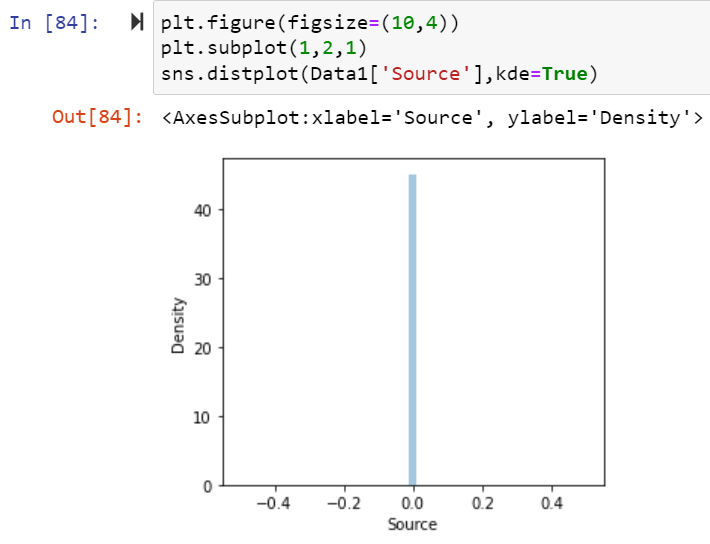






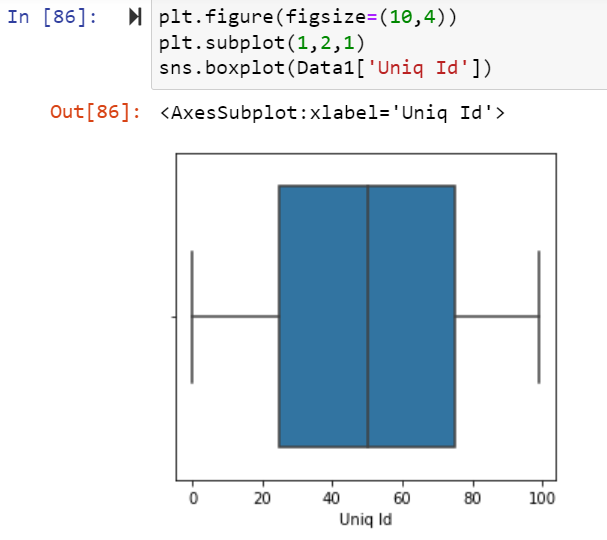
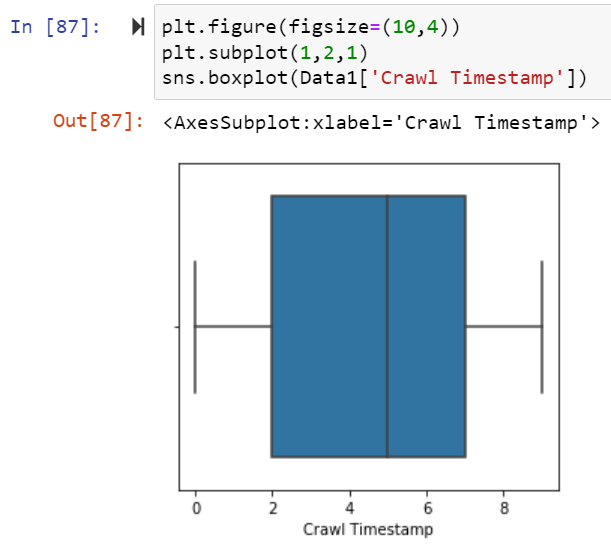


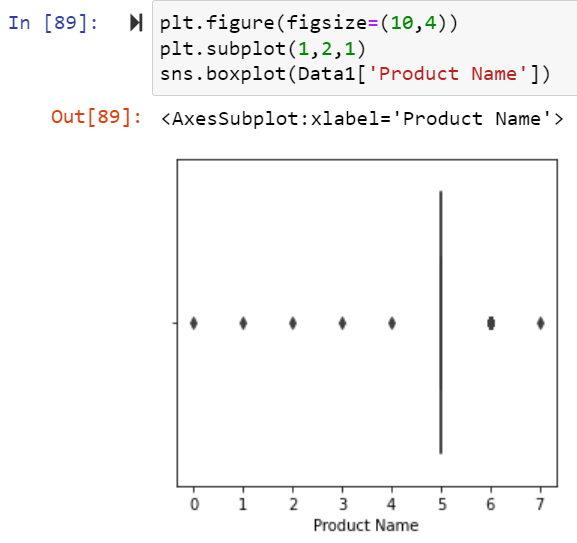
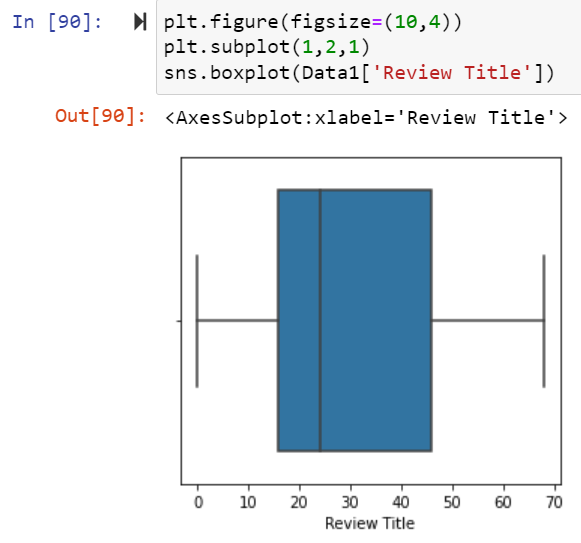


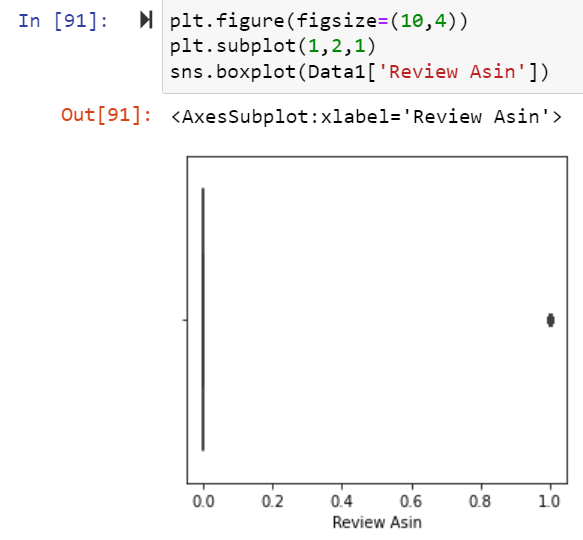
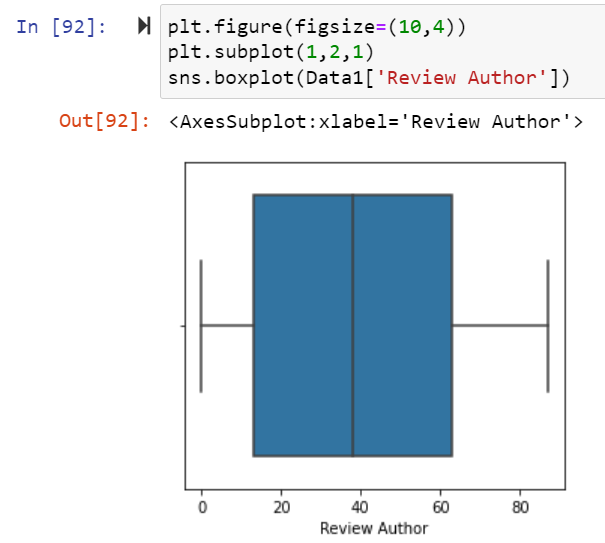


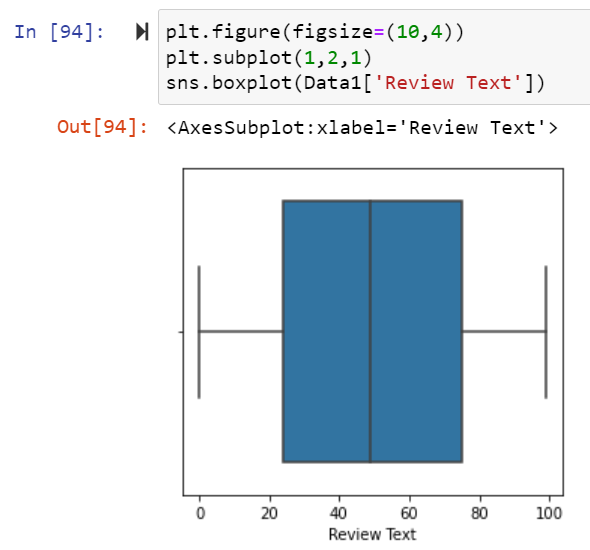
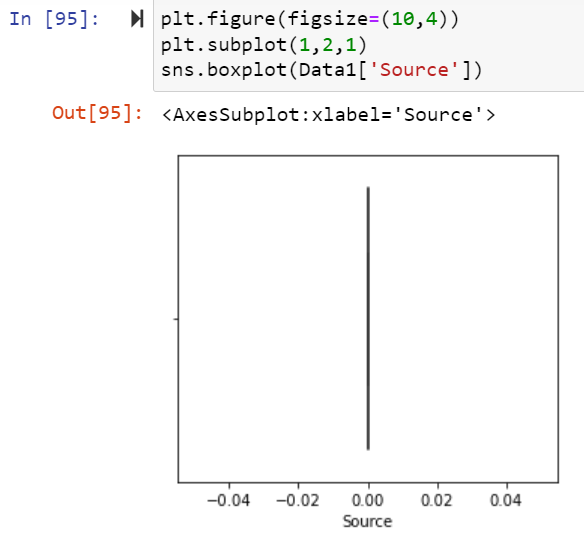
**Checking Outliers of this Dataset:**

An outlier is an **object(s) that deviates significantly from the rest of the object collection**. It is an abnormal observation during the Data Analysis stage, that data point lies far away from other values. An outlier is an observation that diverges from well-structured data.

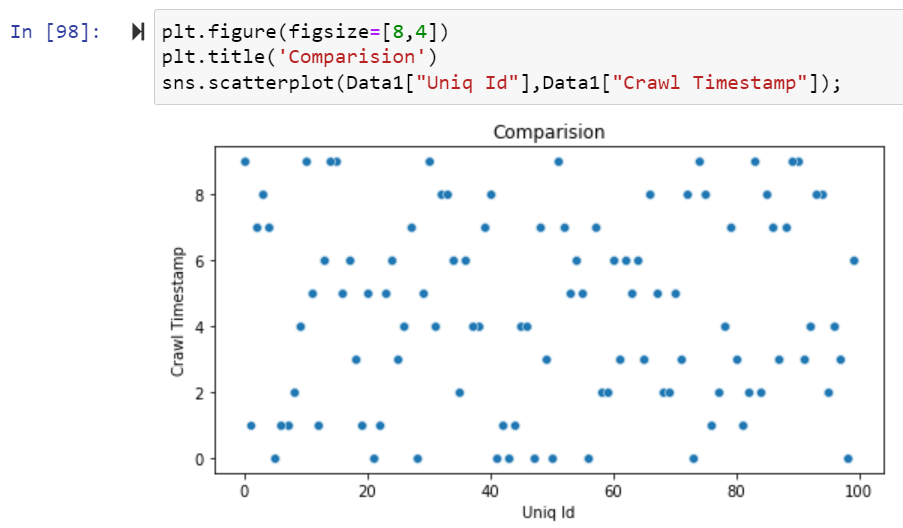
 

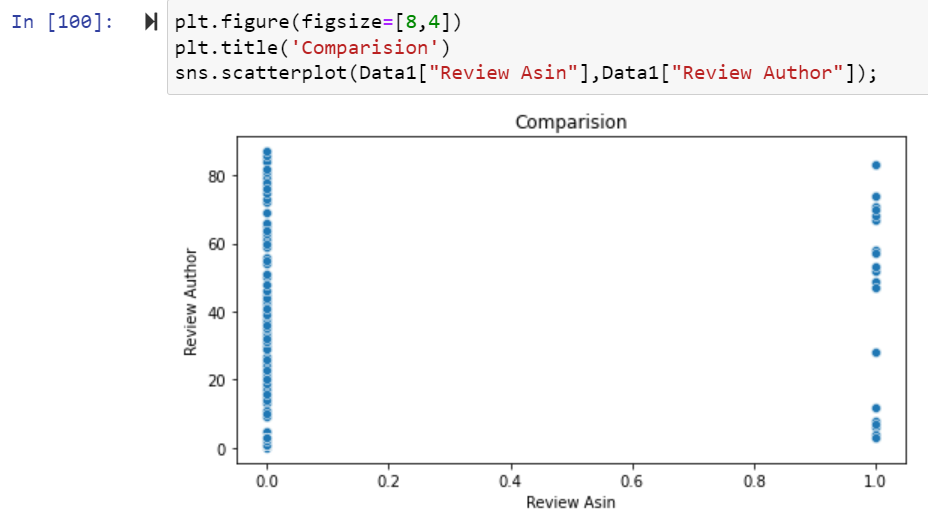
 

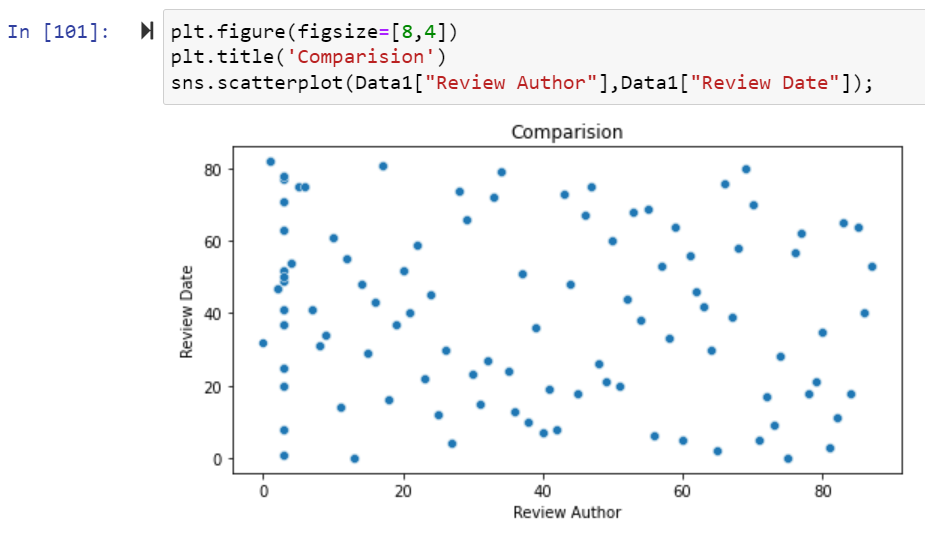
 

**Bi varient Analysis of this Dataset:**

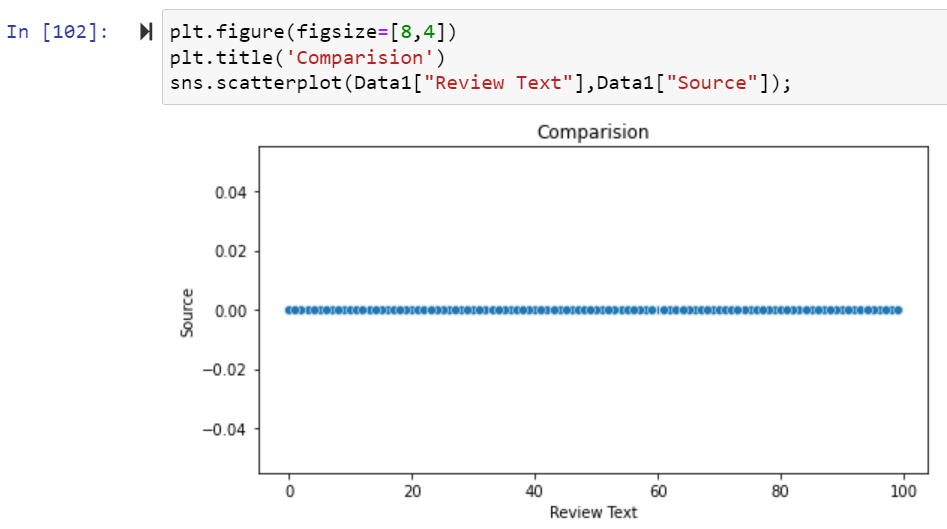
#### Scatter Plot:







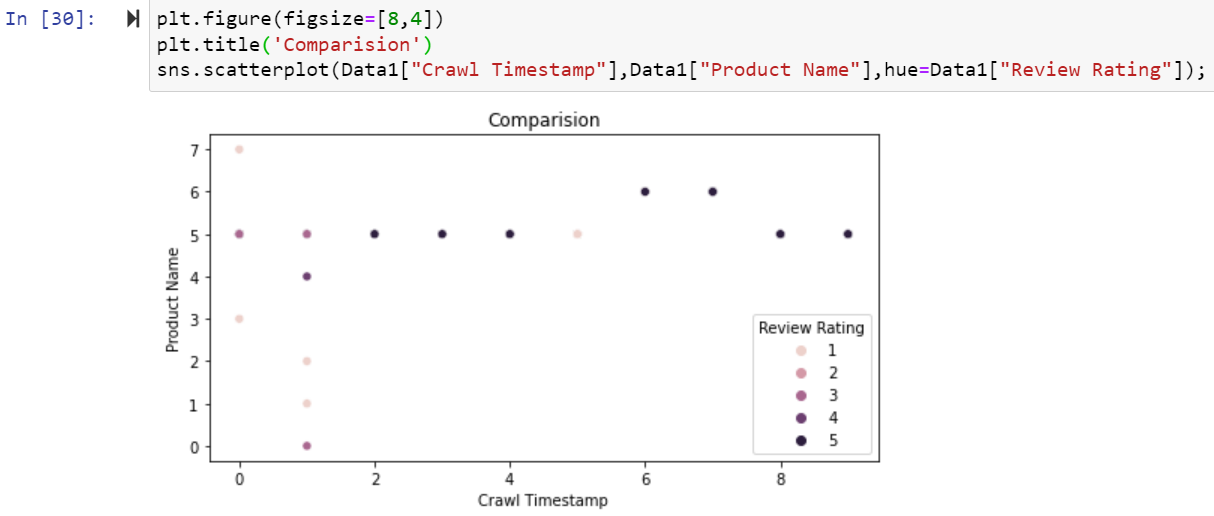
Here above Scatterplot shows the comparison between two columns.



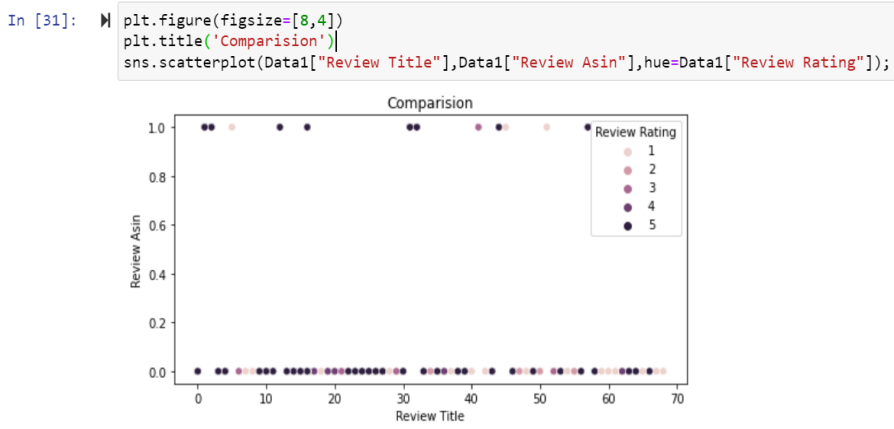
Here above Scatterplot shows the comparison between two columns.



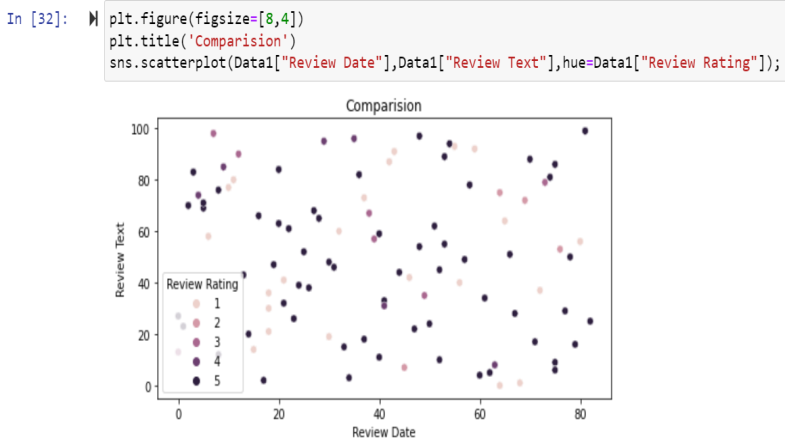
Here above Scatterplot shows the comparison between two columns.



Here above Scatterplot shows the comparison between two columns.



Here above Scatterplot shows the comparison between two columns.



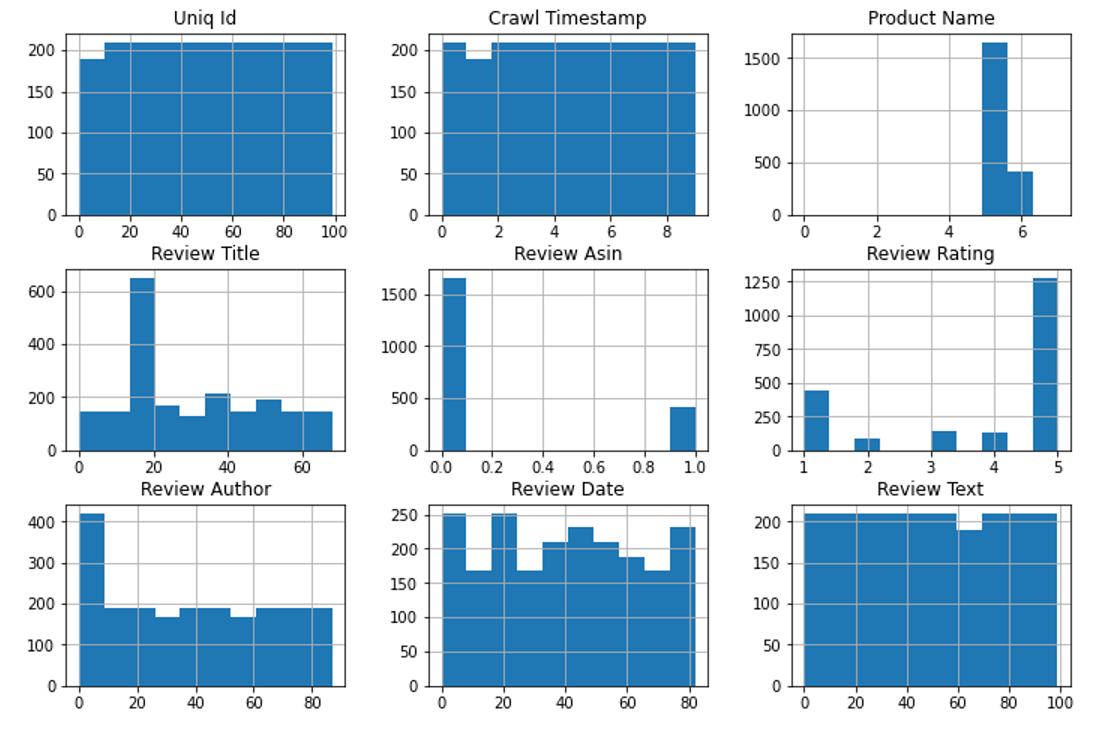
Here above Scatterplot shows the comparison between two columns.

**Multi Varient Analysis of Train Dataset:**

Multivariate analysis is used **to study more complex sets of data than what univariate analysis methods can handle**. ... Multivariate analysis can reduce the likelihood of Type errors. Sometimes, univariate analysis is preferred as multivariate techniques can result in difficulty interpreting the results of the test.

**HistPlot:**

Plot univariate or bivariate **histograms to show distributions of datasets**. ... Either a long-form collection of vectors that can be assigned to named variables or a wide-form dataset that will be internally reshaped. x, y vectors or keys in data. Variables that specify positions on the x and y axes.



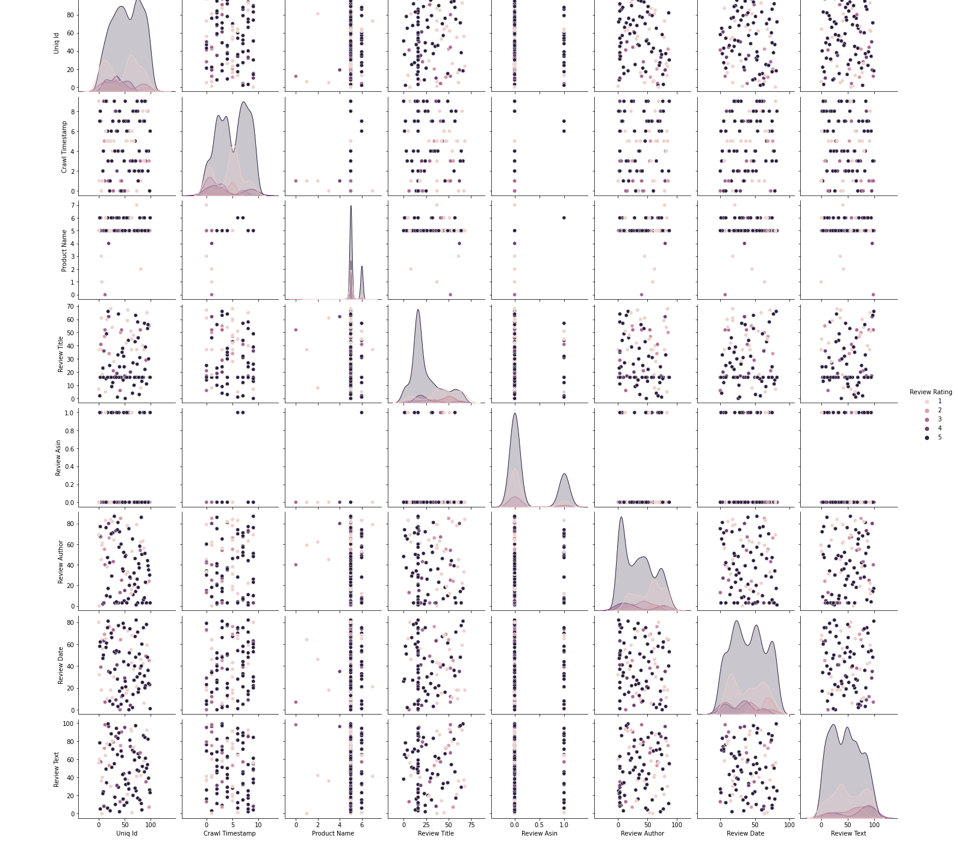
Above are the histplot of thisdataset.

**Pair Plot of this Dataset**:

Pair plot **visualizes given data to find the relationship between them where the variables can be continuous or categorical**. Plot pairwise relationships in a data-set. Pair plot is a module of sea born library which provides a high-level interface for drawing attractive and informative statistical graphics.

**sns.pairplot(train, hue='** **Review Rating')**

A pairs plot allows us to see both distribution of single variables and relationships between two variables . Pair plots are a great **method to identify trends for follow-up analysis** and, fortunately, are easily implemented in Python!



Above are the pair plot of this dataset.

**State the set of assumptions (if any) related to the problem under consideration**

Several Machine Learning models have been developed and deployed to filter out the unruly language and protect internet users from becoming victims of online harassment and cyberbullying.

**Hardware and Software Requirements and Tools Used**

1. Software Requirements:

* 1. Coding Language: Python3, Python
  2. Coding software : Anaconda, Jupyter Notebook

1. Microsoft Office Word.
2. Snipping Tools (For Screenshots).
3. Microsoft Excel

**Non Functional Requirements:**

1: Platform Independent: The application would be platform independent if all the requirements are installed in the device.

2: Performance: The application should have better accuracy and should provide the information in less time.

3: Capacity: The capacity of the storage should be high so that large amount of data can be stored in order to train the model.

**Hardware Requirements:**

1 GB RAM.

200 GB HDD.

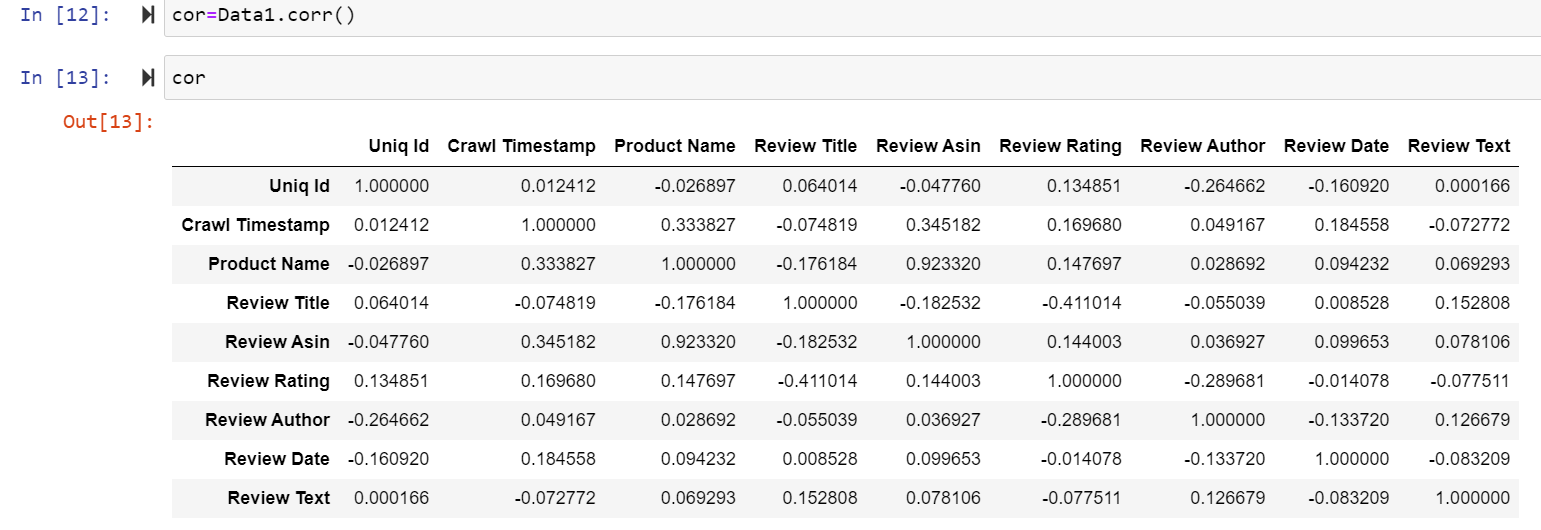
Intel 1.66 GHz Processor Pentium 4

**Visualizations:**

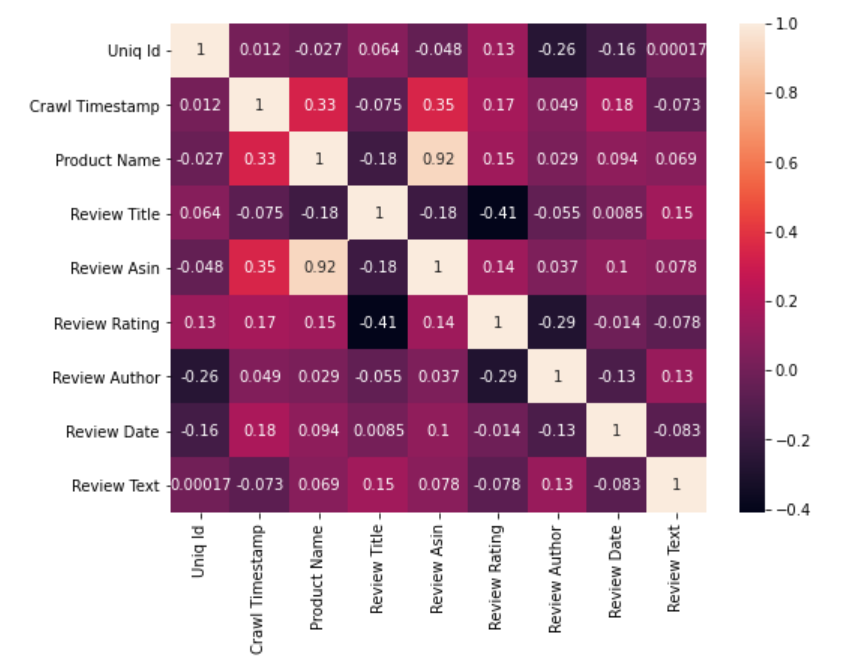
**Heatmap of Train Dataset:**

cor=df\_tr.corr()

cor



After that Checking Correlation of all independent columns with Target column in this dataset.



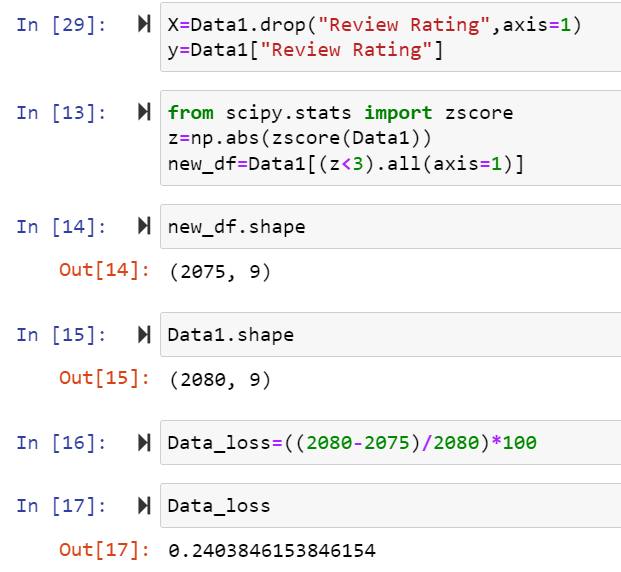
Heat map shows the correlation of every independent variable in dataset with target variable. Here above heatmap the every independent variable check correlation with this dataset.



Here we shows the correlation in another way in this above we show all columns are positively correlated with target variable.

**Z-score:**

Take your data point, subtract the mean from the data point, and then divide by your standard deviation. That gives you your Z-score. You can use Z-Score to determine outliers.One of the most commonly used tools in determining outliers is the Z-score. Z-score is just the number of standard deviations away from the mean that a certain data point is.In your future data science life, Z-scores are gonna be a really useful way to think about how usual or how unusual a certain data point is. And that’s going to be really valuable once we start making inferences based on our data. In this story, we will take a deep dive into our notebooks and learn how to detect outliers using Z-Score.



After that Cheking Data loss of the dataset after using preprocessing steps here using zscore method checking dataloss here after zscore there 10.17 is dataloss.

**IQR:**

The interquartile range rule is useful in detecting the presence of outliers. [Outliers](https://www.thoughtco.com/what-is-an-outlier-3126227) are individual values that fall outside of the overall pattern of a data set. This definition is somewhat vague and subjective, so it is helpful to have a rule to apply when determining whether a data point is truly an outlier—this is where the interquartile range rule comes in.

## What Is the Interquartile Range?

Any set of data can be described by its [five-number summary](https://www.thoughtco.com/what-is-the-five-number-summary-3126237). These five numbers, which give you the information you need to find patterns and outliers, consist of (in ascending order):

* The minimum or lowest value of the dataset
* The first quartile Q1, which represents a quarter of the way through the list of all data
* The [median](https://www.thoughtco.com/what-is-the-median-3126370) of the data set, which represents the midpoint of the whole list of data
* The third quartile Q3, which represents three-quarters of the way through the list of all data
* The maximum or highest value of the data set.

These five numbers tell a person more about their data than looking at the numbers all at once could, or at least make this much easier. For example, the [range](https://www.thoughtco.com/what-is-the-range-in-statistics-3126248), which is the minimum subtracted from the maximum, is one indicator of how spread out the data is in a set (note: the range is highly sensitive to outliers—if an outlier is also a minimum or maximum, the range will not be an accurate representation of the breadth of a data set).

Range would be difficult to extrapolate otherwise. Similar to the range but less sensitive to outliers is the interquartile range. The [interquartile range](https://www.thoughtco.com/what-is-the-interquartile-range-3126245) is calculated in much the same way as the range. All you do to find it is subtract the first quartile from the third quartile:

IQR = Q3 – Q1.

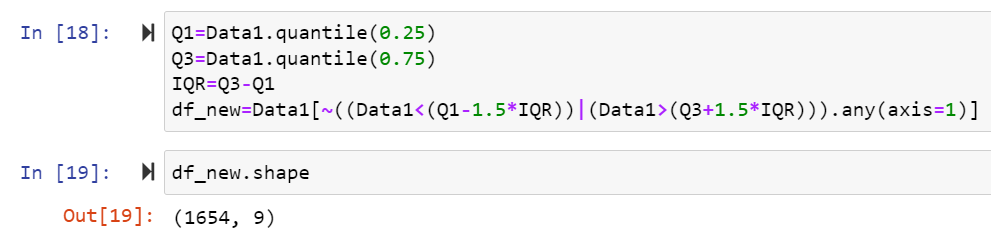
The interquartile range shows how the data is spread about the median. It is less susceptible than the range to outliers and can, therefore, be more helpful.

## Using the Interquartile Rule to Find Outliers

Though it's not often affected much by them, the interquartile range can be used to detect outliers. This is done using these steps:

1. Calculate the interquartile range for the data.
2. Multiply the interquartile range (IQR) by 1.5 (a constant used to discern outliers).
3. Add 1.5 x (IQR) to the third quartile. Any number greater than this is a suspected outlier.
4. Subtract 1.5 x (IQR) from the first quartile. Any number less than this is a suspected outlier.

Remember that the interquartile rule is only a rule of thumb that generally holds but does not apply to every case. In general, you should always follow up your outlier analysis by studying the resulting outliers to see if they make sense. Any potential outlier obtained by the interquartile.

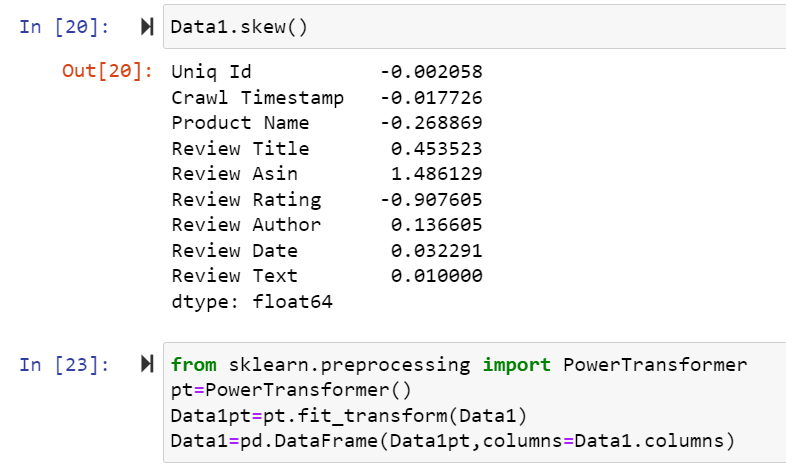


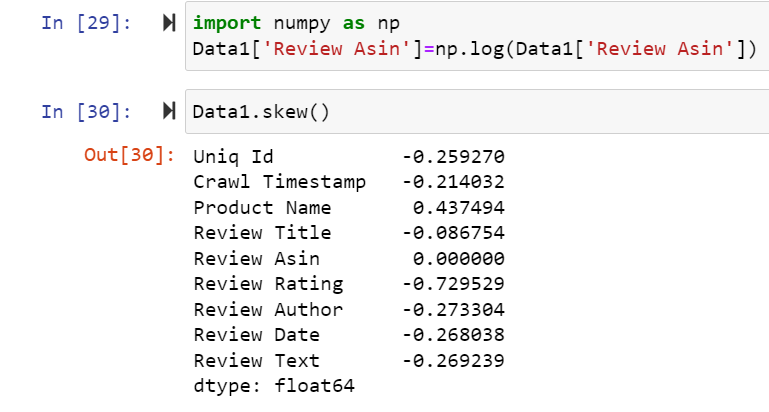
Here after Applying IQR method checking dataloss here using IQR method same dataloss.

**Skewness:**

The skewness is a measure of symmetry or asymmetry of data distribution, and kurtosis measures whether data is heavy-tailed or light-tailed in a normal distribution. Data can be positive-skewed (data-pushed towards the right side) or negative-skewed (data-pushed towards the left side).

**Checking skewness in dataset:**

****

****

To remove skewness use Power Transformer and log transform technique in dataset.

**Model/s Development and Evaluation**

**Identification of possible problem-solving approaches (methods)**

**Approach**

Importing the required libraries and reading the dataset.

Merging of the two datasets

* Understanding the dataset

1. Exploratory Data Analysis (EDA) –

* Data Visualization

1. Feature Engineering

* Duplicate value removal
* Missing value imputation
* Encoding of categorical variables
* Dropping of redundant feature columns
* Check for the outliners and removal of outliers.

1. Model Building

* Performing train test split
* Feature Scaling
* Lasso Regression
* Ridge Regression
* Decision Tree Regressor
* Random Forest Regressor
* AdaBoost Regressor
* KNeighbors Regressor

1. Model Validation

* R2 square error

1. Hypermeter Tuning (GridSearchCV)

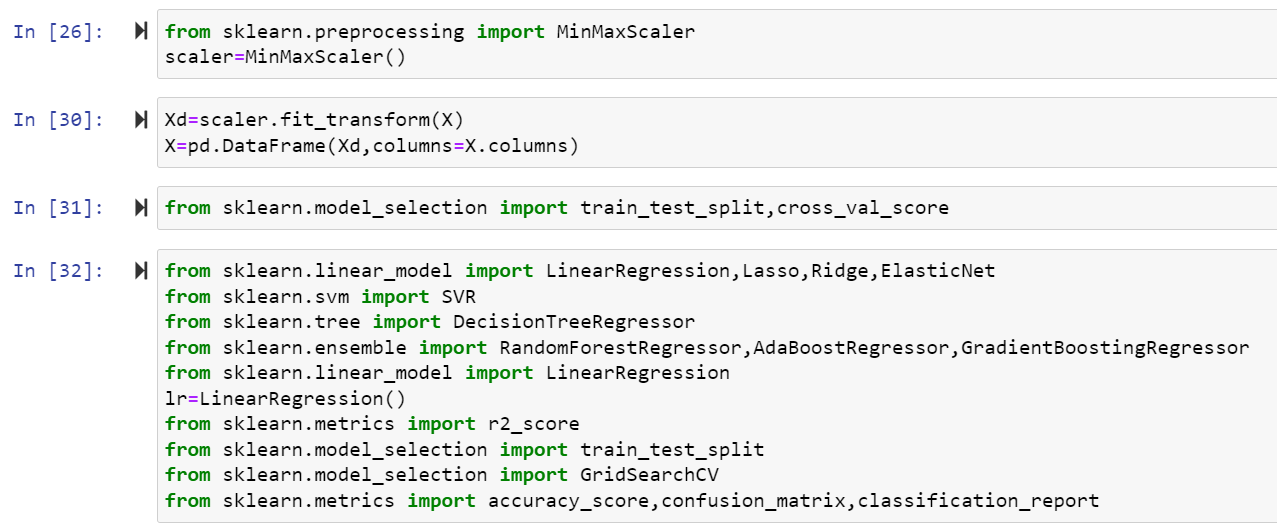
For Random Forest Regressor

1. Checking for Feature Importance
2. Creating the final model and making predictions

**Testing of Identified Approaches (Algorithms)**

* Performing train test split
* Feature Scaling
* Lasso Regression
* Ridge Regression
* Decision Tree Regressor
* Random Forest Regressor
* AdaBoost Regressor
* KNeighbors Regressor

**Building Machine Learning Models:**



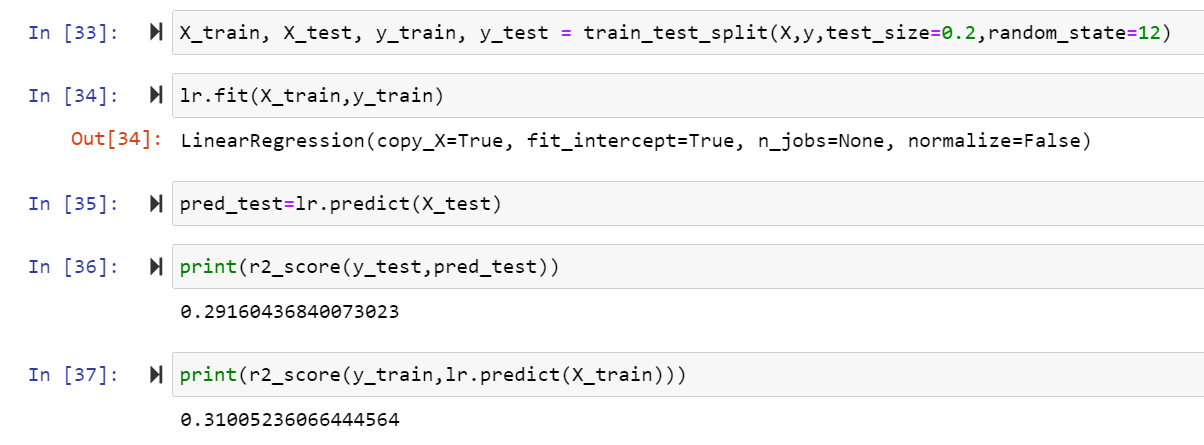
**Lasso Regression:**

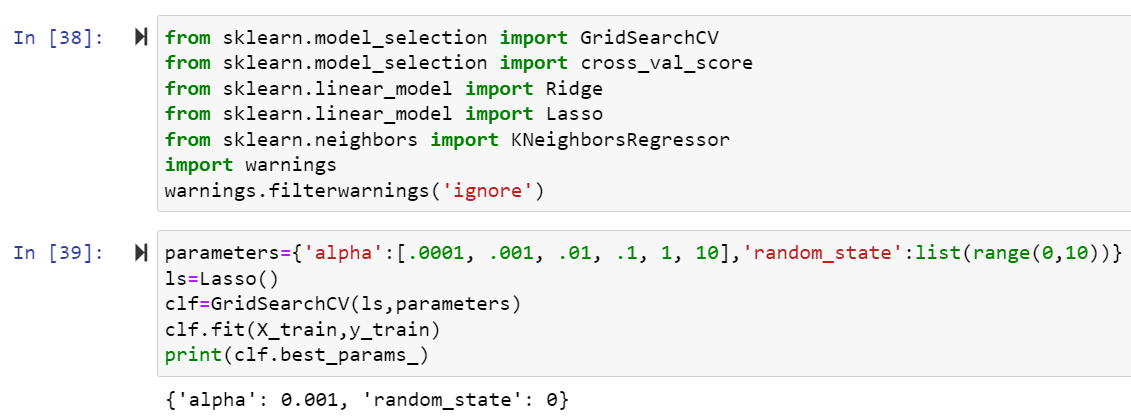
The word “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. It is a statistical formula for the regularisation of data models and feature selection.LASSO or L1 regularization is a technique that can be used to improve many models, including generalized linear models (GLMs) and Neural networks. LASSO stands for “**least absolute shrinkage and selection operator**.” However, you might wonder if the phrase or the acronym came first.

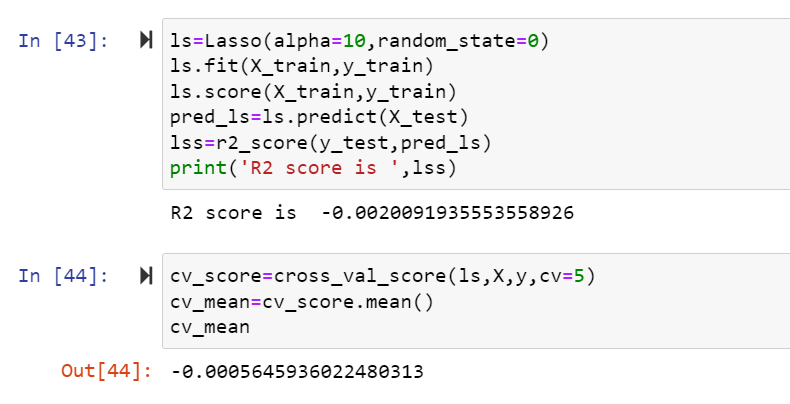
**What is Lasso Regression**?

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

Lasso Regression uses L1 regularization technique (will be discussed later in this article). It is used when we have more number of features because it automatically performs feature selection.







Here r2 score is -0.002 for dataset and cross validation score is -0.0005 in Lasso algorithm.

**Ridge Regression:**

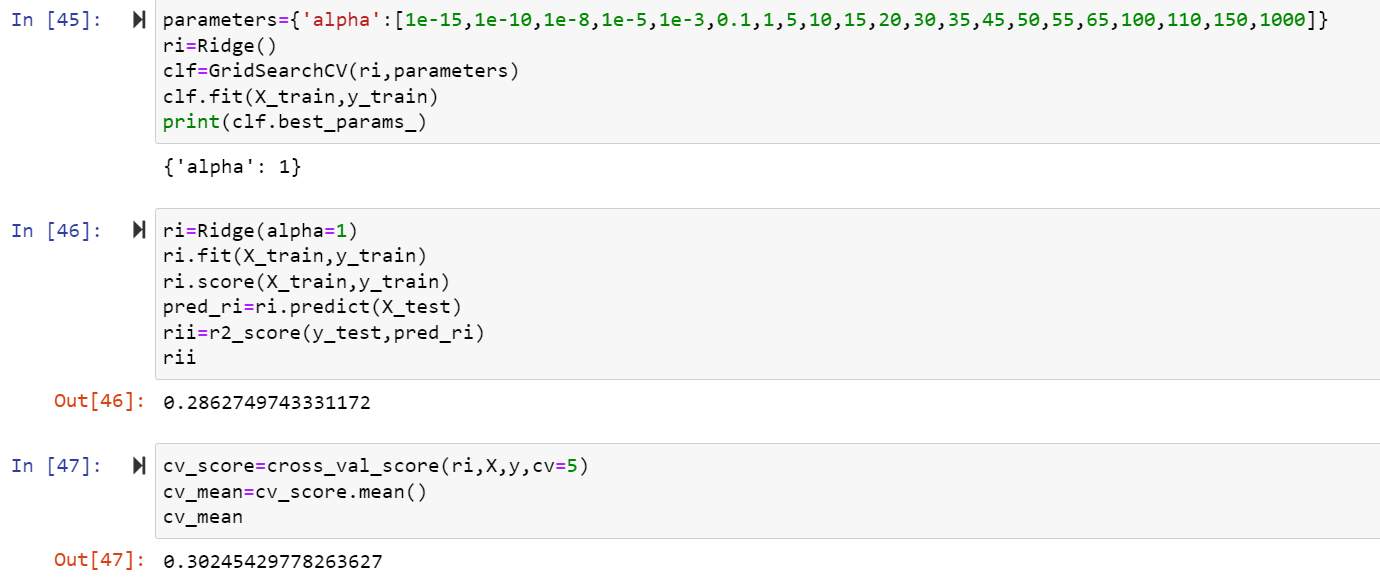
Ridge [regression](https://www.mygreatlearning.com/blog/what-is-regression/) is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

**Ridge Regression Models**

For any type of regression machine learning models, the usual regression equation forms the base which is written as:

Y = XB + e

Where Y is the dependent variable, X represents the independent variables, B is the regression coefficients to be estimated, and e represents the errors are residuals. Once we add the lambda function to this equation, the variance that is not evaluated by the general model is considered. After the data is ready and identified to be part of L2 regularization, there are steps that one can undertake.



Here r2 score is 0.28 for this dataset and 0.30 cross validation score Ridge algorithm.

**RandomForest Regression:**

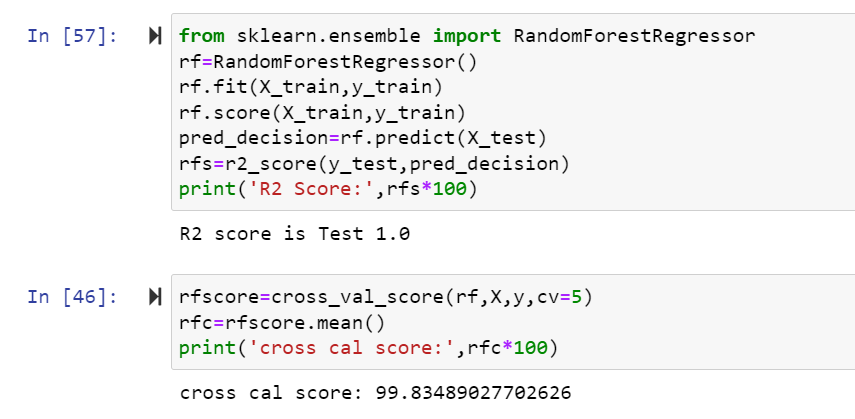
Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forest is an ensemble of decision tree algorithms.It is an extension of [bootstrap aggregation (bagging)](https://machinelearningmastery.com/bagging-ensemble-with-python/) of decision trees and can be used for classification and regression problems.

In bagging, a number of decision trees are created where each tree is created from a different bootstrap sample of the training dataset. A [bootstrap sample](https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/) is a sample of the training dataset where a sample may appear more than once in the sample, referred to as **sampling with replacement**.

Bagging is an effective ensemble algorithm as each decision tree is fit on a slightly different training dataset, and in turn, has a slightly different performance. Unlike normal decision tree models, such as [classification and regression trees](https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/) (CART), trees used in the ensemble are unpruned, making them slightly overfit to the training dataset. This is desirable as it helps to make each tree more different and have less correlated predictions or prediction errors.

Predictions from the trees are averaged across all decision trees resulting in better performance than any single tree in the model.



Here r2 score is 1.0 for this dataset and cross validation score is 99 in RandomForestRegressor algorithm.

**SVR Regression:**

Support Vector Regression (SVR) uses the same principle as SVM, but for regression problems. Let’s spend a few minutes understanding the idea behind SVR.

Recall the idea of SVR. Here, you give a set of input vectors and defined an output. The SVR then fits a model and tries to learn from those input vectors and finally predicts the response for a given new input vector. While working with time series data like stock prices, you need to determine which will be the "feature vector".

This is because time series data is time-dependent i.e. there will be a lot of past values and you can not take everything as feature vectors. For example, you have a stock price data set that contains the prices of a single stock from the previous six months. Now, based on this data you want to forecast the future price of the stock.

Here, you have to transform the past data to build some feature vectors. There are many ways you can do this i.e. averaging the past one month's prices or the current price of the stock divided by the moving average. This will minimize the input vectors and make it easier for the SVR to fit them. SVR features are unordered x-y pairs, so you can not get a model that considers time order. If you want to maintain the time order, you can build separate SVRs i.e. one for the past 10 days, one for the past 1 month, etc. and then take the average of all the predictions and forecast the values.

#### Advantages and Disadvantages of Support Vector Regression

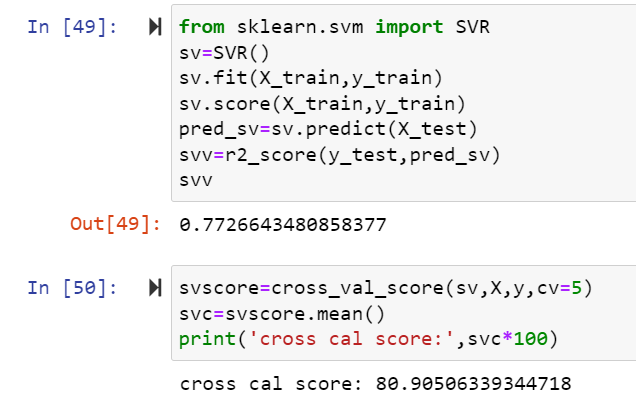
There are some key benefits to choose a support vector machine for regression tasks. There are some drawbacks as well. Let's talk about them-

**The key advantages are-**

* SVM works really well with high-dimensional data. If your data is in higher dimensions, it is wise to use SVR.
* For data with a clear margin of separations, SVM works relatively well.
* When data has more features than the number of observations, SVM is one of the best algorithms to use.
* As a discriminative model, it need not to memorize anything about data. Therefore, it is memory efficient.

**Some drawbacks are-**

* It is a bad option when the data has no clear margin of separation i.e. the target class contains overlapping data points.
* It does not work well with large data sets.
* For being a discriminative model, it separates the data points below and above a hyperplane. So, you will not get any probabilistic explanation of the output.
* It is hard to understand and interpret SVM as its underlying structure is quite complex.

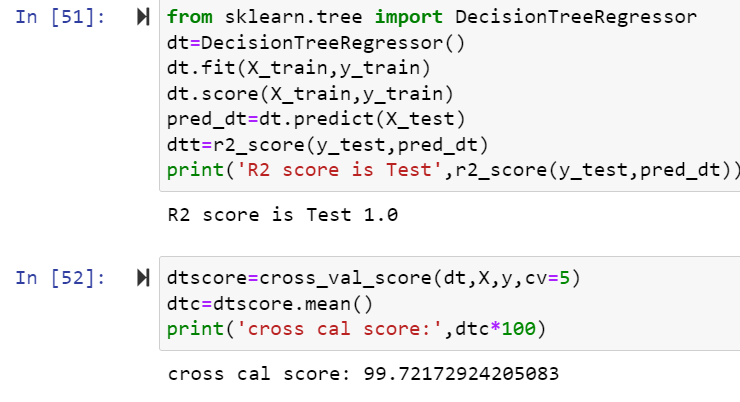


Here r2 score is 0.77 for this dataset and cross validation score is 80.90 in SVR algorithm.

**DecisionTreeRegressor:**

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

**Decision Tree** is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

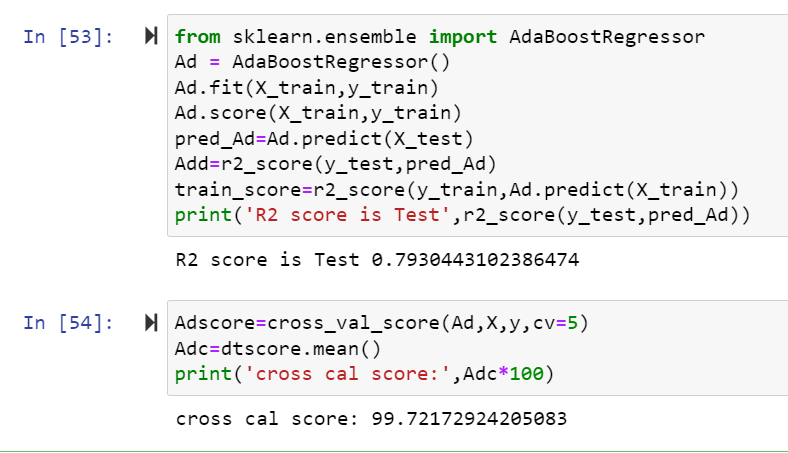


Here r2 score is 1.0 for this dataset and cross validation score is 99 in DecisionTreeRegressor algorithm.

**AdaBoostRegressor:**

Adaboost stands for Adaptive Boosting and it is widely used ensemble learning algorithm in machine learning. Weak learners are boosted by improving their weights and make them vote in creating a combined final model. In this post, we'll learn how to use [AdaBoostRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html" \t "_blank) class for the regression problem. AdaboostRegressor starts fitting the regressor with the dataset and adjusts the weights according to error rate. The tutorial covers:

* Preparing data
* Defining the model
* Predicting and checking the accuracy



Here r2 score is 0.79 for train dataset and cross validation score is 99 in AdaBoostRegressor algorithm.

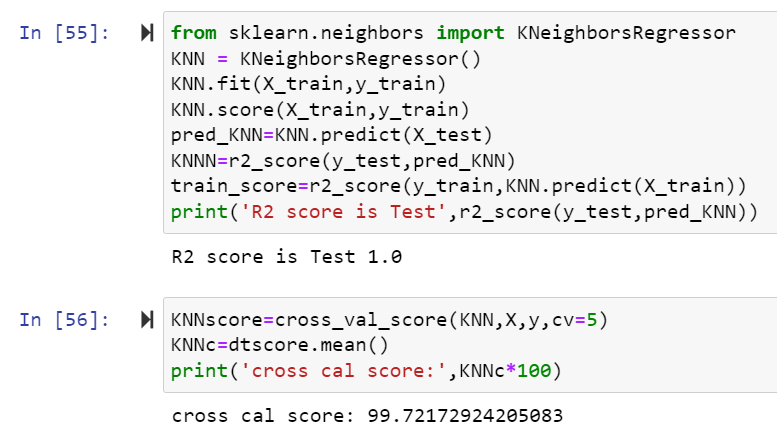
**KNN Regressor:**

As we saw above, KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses '**feature similarity**' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.

The kNN algorithm is one of the most famous [machine learning](https://realpython.com/learning-paths/machine-learning-python/) algorithms and an absolute must-have in your machine learning toolbox. Python is the go-to programming language for machine learning, so what better way to discover kNN than with Python’s famous packages [NumPy](https://realpython.com/numpy-tutorial/) and [scikit-learn](https://scikit-learn.org/stable/)!

Below, you’ll explore the kNN algorithm both in theory and in practice. While many tutorials skip the theoretical part and focus only on the use of libraries, you don’t want to be dependent on automated packages for your machine learning. It’s important to learn about the mechanics of machine learning algorithms to understand their potential and limitations.

At the same time, it’s essential to understand how to use an algorithm in practice. With that in mind, in the second part of this tutorial, you’ll focus on the use of kNN in the Python library scikit-learn, with advanced tips for pushing performance to the max.



Here r2 score is 1.0 for the dataset and cross validation score is 99 in KNN Regressor algorithm.

Here we check r2 score and after that check cross validation score of all the model the Random forest repressor is the best model because r2 score and cross. validation score is 0.99 So apply hyper parameter tuning on it.

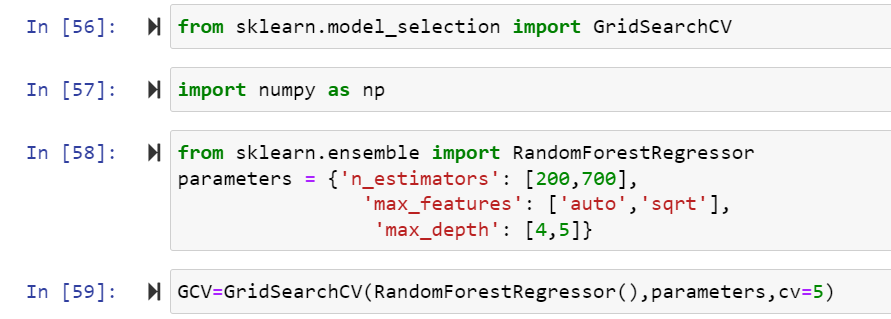
**GridSearchCv**:

GridSearchCV is **a library function that** is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

**Parameterlist:**

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as **Hyperparameters.**These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. Of course, you must select from a specific list of hyperparameters for a given model as it varies from model to model.

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as **Hyperparameters.**These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. Of course, you must select from a specific list of hyperparameters for a given model as it varies from model to model.

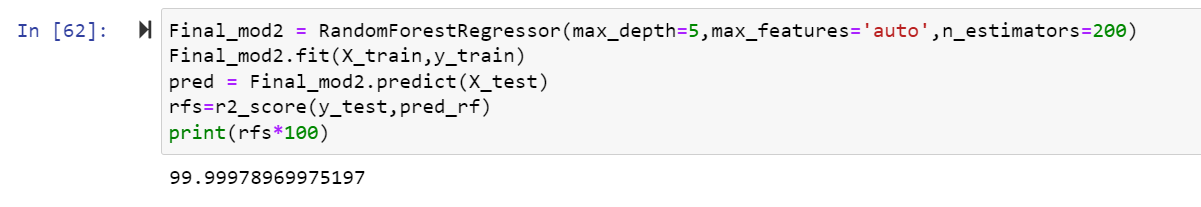


Here above are the parameter list of RandomForestRegressor model.

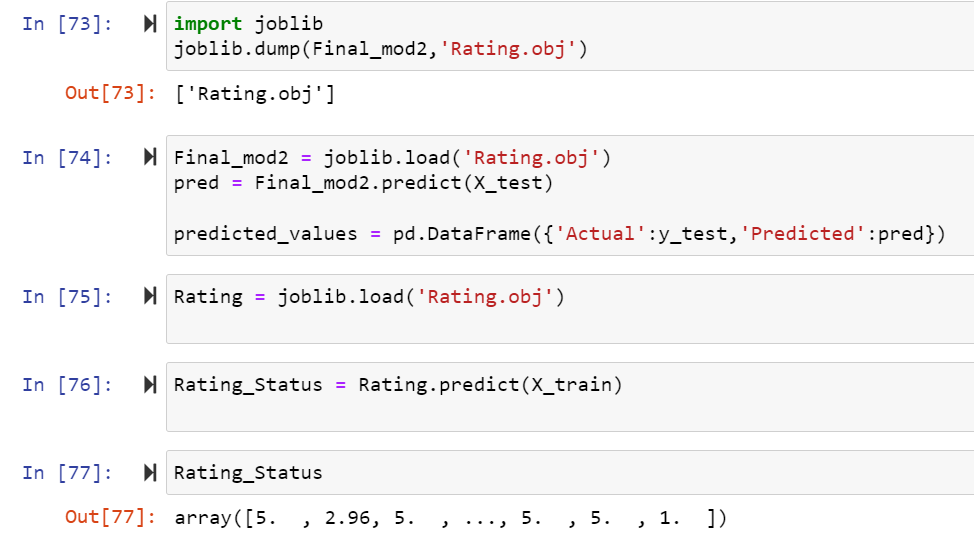
The above code block we have the following parameters  
max\_features: In this maximum features there are two values auto and auto,sqrt.

n\_estimator is 200 ,700 and in max\_depth 4,5.





Here after use different parameter list the best parameter list is select. Above are the best parameter list RandomForestRegressor. Put this parameters into the model so output is finally best score os RandomForestRegressor is 99 so it is the best score.



Finally Load the model and predict the values.

**Visualizations:**

For visualization purpose use heatmap for dataset: data visualization is a method of graphically representing numerical data where the value of each data point is indicated using colors. ... More importantly, heatmaps help to classify the sections that are performing sub-par and need optimization

**Interpretation of the Results:**

House price prediction project there are two dataset Train and Test dataset in Train dataset the SalePrice are the target variable was present and there was continues values are present so it regression problem need to use regression algorithm.

After that check the shapes of the dataset and check null values are present that that dataset and remove the null values .

Check the data types of the dataset and need to convert categorical columns into numeric.

And also the use zscore and IQR to check the data loss.

And check the skewness and remove the skewness using power transform and log transform method.

Use scaling technique.

And use different Regression models and depend on r2 score and cross validation score select Random forest repressor is the best model and apply hyper parameter tuning on it.

After that load that model and predict the values.

**CONCLUSION**

**Conclusion:**

In conclusion, my thesis was proven to be correct. Combining the formerly known data about each user’s similarity to other users with the sentiment analysis of the review text itself, does help improve the model prediction of what rate the user’s review will get.