**PREDICTION OF GOLD PRICE USING MACHINE LEARNING IN PYTHON**

Done by

# Mr. Amey Makrand Kulkarni

Mentored by

# Ms. Jayakodi R and Mr. Vignesh M

In partial fulfillment of the requirement for POSTGRADUATE DIPLOMA IN DATA SCIENCE AND ANALYTICS



NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY [NIELIT] CHENNAI

(An Autonomous Scientific Society of Ministry of Electronics & Information Technology) Government of India

No. 25, Gandhi Mandapam Road, Chennai — 600025, Tamilnadu, India.

# DECLARATION



We hereby declare that the project work entitled

**PREDICTION OF GOLD PRICE USING MACHINE LEARNING IN PYTHON**

Done for

# NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY [NIELIT] CHENNAI

(An Autonomous Scientific Society of Ministry of Electronics & Information Technology) Government of India

No.25, Gandhi Mandapam Road, Chennai- 600025, Tamilnadu, India.

# Under the Guidance of

# Ms. Jayakodi R and Mr. Vignesh M

**NAME OF THE STUDENT Place:** Chennai

Mr. Amey Makrand Kulkarni **Date:** 10/03/2022

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to my guides Ms. Jayakodi and Mr. Vignesh for the support for the project work, for their patience, motivation and immense knowledge.

Sincere thanks to my Dr. Sanjeev Kumar Jha, Joint Director (Tech.), NIELIT Chennai, for helping us to do this course & the project during the COVID19 Pandemic situation & also for giving us many supports and valuable guidance during our study.

A special thanks to our lecturers Ms. V.Gayathri/Mr. Rakesh Varma T/Mr. Saran for constant support, guidance, help and encouragement to learn in better way.

My heartfelt gratitude and thanks to my mother Mrs. Neelima Kulkarni and my father Mr. Makarand Kulkarni for always helping me and supporting my education.

Finally heartfelt gratitude and thanks to my sister & friends for constantly being there by my side all the time and believing in my ability to pursue this course and project.

**POSTGRADUATE DIPLOMA IN DATA SCIENCE AND ANALYTICS**

**Project Synopsis**

**Name of the Student:** Mr. Amey Makarand Kulkarni

**Course:** P.G. Diploma in Data Science & Analytics **Institute/Company:** NIELIT, Chennai

**Guide Name:** Ms. Jayakodi R and Mr. Vignesh M

**Project Title:** “Prediction of Gold Price Using Machine Learning in Python.”

**Abstract:-**

Historically, gold was used for supporting trade transactions around the world besides other modes of payment. Various states maintained and enhanced their gold reserves and were recognized as wealthy and progressive states. In present times, precious metals like gold are held with central banks of all countries to guarantee re-payment of foreign debts, and also to control inflation. Moreover, it also reflects the financial strength of the country. Besides government agencies, various multi-national companies and individuals have also invested in gold reserves. In traditional events of Asian countries, gold is also presented as gifts/souvenirs and in marriages, gold ornaments are worn by ladies which enhances their beauty. In addition to the demand and supply of the commodity in the market, the performance of the world’s leading economies also strongly influences gold rates. We predict future gold rates based on some share prices which affect gold prices directly or indirectly using machine learning techniques.

**Keywords:** Gold, machine learning.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TITLE** | **PAGE NO.** |
| 1 | AIM & OBJECTIVES | 1 |
| 2 | FIGURES | 2-3 |
| 3 | INTRODUCTION | 6 |
| 4 | DATA & METHODOLOGY | 7-10 |
| 5 | LIBRARIES USED | 12-13 |
| 6 | WALKTHROUGH | 14-29 |
| 7 | RESULT | 30 |
| 8 | CONCLUSION | 31 |
| 9 | REFERENCES | 32 |

**AIM & OBJECTIVES**

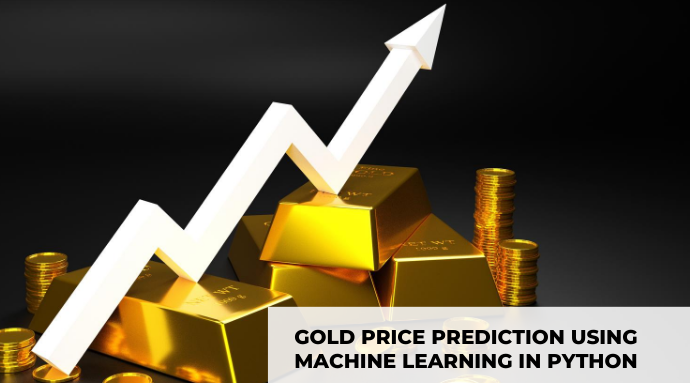
**AIM:** To predict Gold Price using Machine Learning in Python (using Random Forest Regressor.)

# OBJECTIVES:

* Gold Price dataset retrieval from [Kaggle](https://www.kaggle.com/jsphyg/weather-dataset-rattle-package).( <https://www.kaggle.com/altruistdelhite04/gold-price-data>)
* Random Forest is used for classification and regression but we will use only the random forest regressor.
* The dataset consists of 2290 records and 6 columns.



**Prediction Of Gold Price Using Machine Learning in Python**



**INTRODUCTION**

Historically, gold had been used as a form of currency in various parts of the world including USA. In recent times also, gold has maintained its value and has been used as a means for assessing the financial strength of a country. Big investors have also been attracted to this precious metal and invested huge amounts in it. Recently, emerging world economies, such as China, Russia, and India have been big buyers of gold, whereas USA, South Africa, and Australia are among the big seller of this commodity. Chinese and Indian traditional events also affect the price of the gold. In that time more money is poured for purchase of this commodity. Small investors also find this commodity for safe investment rather than alternate investment options, which bear in-built investment risks. Internal financial conditions of the aforementioned countries play a vital role for setting spot rates for gold. Governmental investments in gold are largely decided by their financial conditions, and interest rates, as they are indicators of the strength of their economy. When US interest rates become lower, more economic activity is witnessed in US, thus capital inflows in gold market are observed. Similarly, when interest rates lowered in China from 5.31 (2010) to 4.35 (2016), it bought gold aggressively.

The spot price is the current market price at which commodity is purchased or sold for immediate payment and delivery. It is differentiated from the futures price, which is the price at which the two parties agree to transact on future date. Gold spot rates are decided twice a day based on supply and demand in gold market. Fractional change in gold price may result in huge profit or loss for these investors as well as government banks. Forecasting rise and decline in the daily gold rates, can help investors to decide when to buy (or sell) the commodity. Various studies have been conducted by researchers to forecast gold rates, each of them insightful in their own right.

**DATA AND METHODOLOGY**

Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. Machine learning focusses on the development of computer systems that can access data and use it to learn for themselves. It is a method of data analysis that automates analytical model building. Using algorithms that learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

There are 3 main types of machine learning i.e. Supervised Learning, Unsupervised Learning and Reinforcement Learning. As a subset of Artificial Intelligence (AI), machine learning can be used to solve a myriad of problems such as fraud detection, web search results, credit scoring, customer segmentation, email spam filtering, etc.

**WHAT IS SUPERVISED LEARNING?**

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.

**

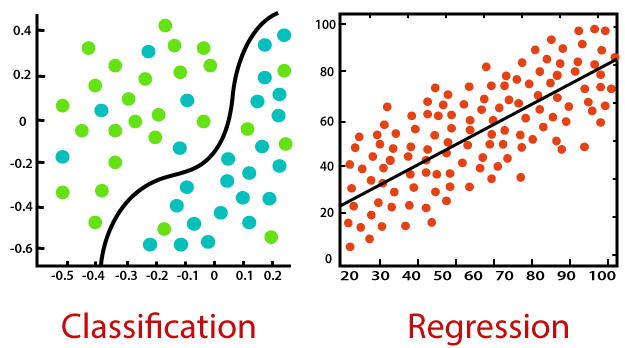
**HOW DOES SUPERVISED LEARNING WORKS?**

Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.

Supervised learning can be separated into two types of problems when data mining—classification and regression:

Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest, which are described in more detail below.

Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.



**WHAT IS UNSUPERVISED LEARNING?**

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

**DATASET**

The data was sourced from the Kaggle website consisting of ten years data from January 2008 to December 2018. It consists of the variables date, silver price, stock profit exchange, gold price, US dollar rate and United States oil ETF. The dataset consists of 2290 records.

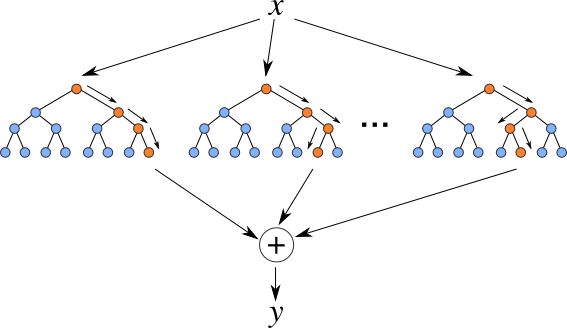
**MACHINE LEARNING ALGORITHMS**

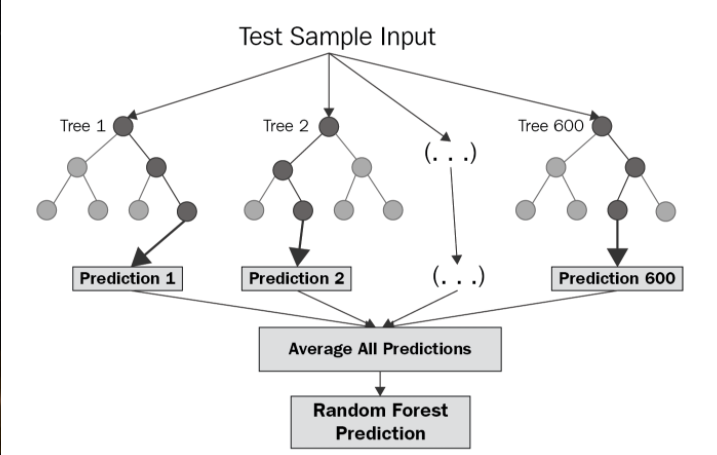
For developing the model, we use the algorithms such as Random Forest Regressor.

There are many algorithms such as Gradient Boosting or the Multiple linear regression but RF is thought to perform better because of its Decision Trees.

**RANDOM FOREST REGRESSION**

**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.





The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let’s walk through the steps:

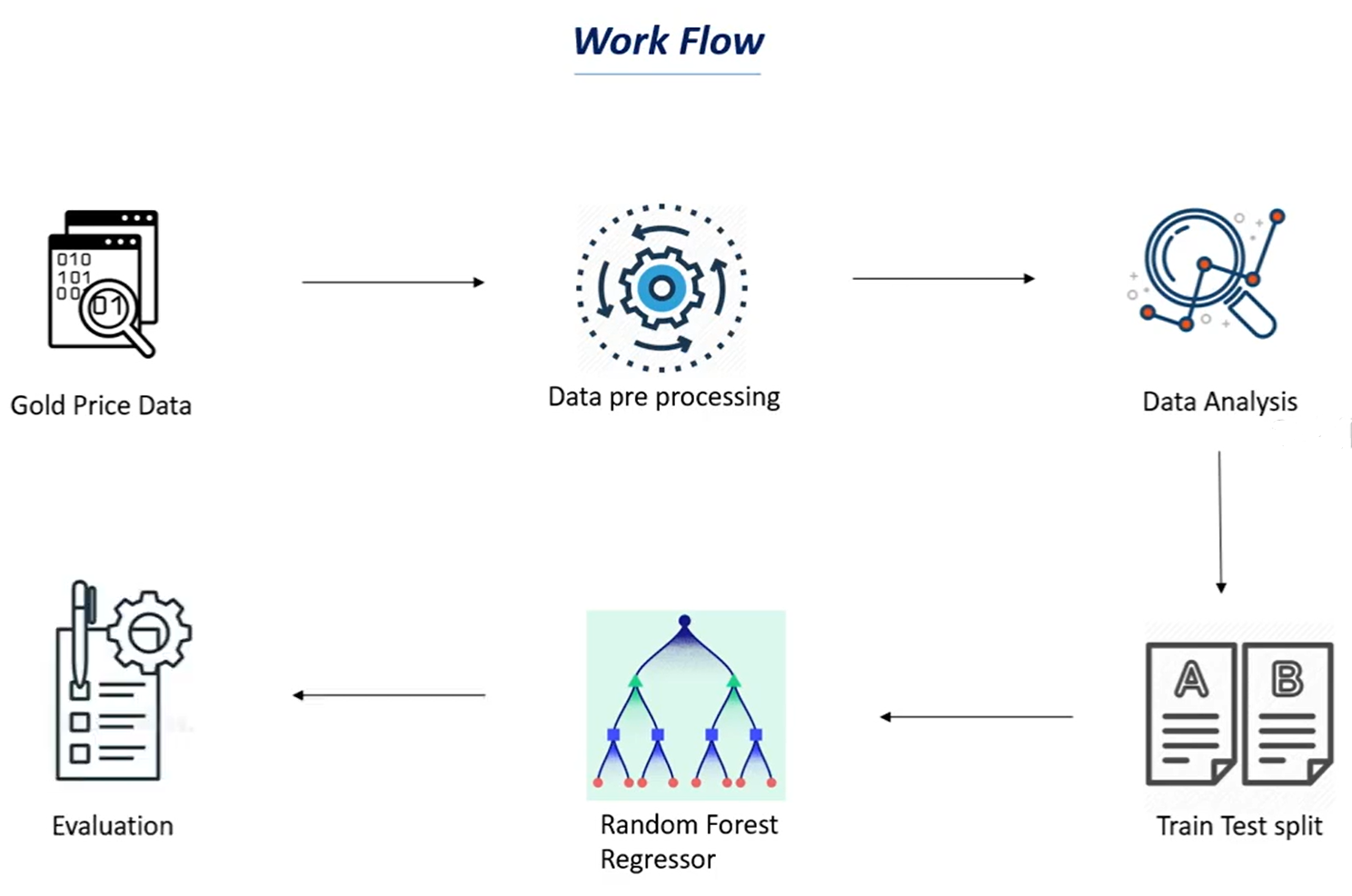
1. Pick at random *k* data points from the training set.
2. Build a decision tree associated to these *k*data points.
3. Choose the number *N*of trees you want to build and repeat steps 1 and 2.
4. For a new data point, make each one of your *N*-tree trees predict the value of *y* for the data point in question and assign the new data point to the average across all of the predicted *y*values.

A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships.

Now, the question arises why do we prefer random forests over decision trees. So, individual trees are more prone to overfitting but random forests can reduce this problem by averaging the predicted results from each tree.

The greater the number of trees will be in the forest, the higher the accuracy will be of the model and also it will reduce overfitting to a large extent.

**IMPLEMENTATION**



**Data Preprocessing:**

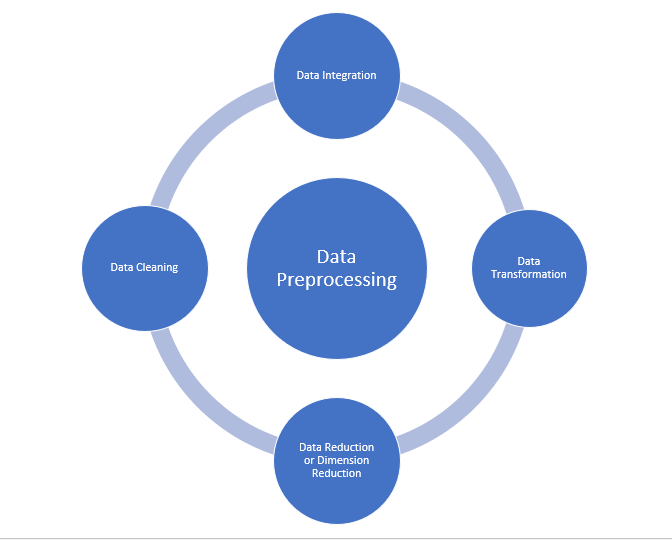
Data preprocessing is required when the data is incomplete, inconsistent or noisy.

**WHAT IS DATA PREPROCESSING?**

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

**STEPS TAKEN IN DATA PREPROCESSING**

1. **Data Cleaning**
2. **Data Integration**
3. **Data Reduction**
4. **Data Transformation**



**Data cleaning:**

Data cleaning is the process to remove incorrect data, incomplete data and inaccurate data from the datasets, and it also replaces the missing values. There are some techniques in data cleaning

**Handling missing values:**

* Standard values like “Not Available” or “NA” can be used to replace the missing values.
* Missing values can also be filled manually but it is not recommended when that dataset is big.
* The attribute’s mean value can be used to replace the missing value when the data is normally distributed  
  wherein in the case of non-normal distribution median value of the attribute can be used.
* While using regression or decision tree algorithms the missing value can be replaced by the most probable  
  value.

**Noisy:**

          Noisy generally means random error or containing unnecessary data points. Here are some of the methods to handle noisy data.

* **Binning:** This method is to smooth or handle noisy data. First, the data is sorted then and then the sorted values are separated and stored in the form of bins. There are three methods for smoothing data in the bin. Smoothing by bin mean method: In this method, the values in the bin are replaced by the mean value of the bin; Smoothing by bin median: In this method, the values in the bin are replaced by the median value; Smoothing by bin boundary: In this method, the using minimum and maximum values of the bin values are taken and the values are replaced by the closest boundary value.
* **Regression:** This is used to smooth the data and will help to handle data when unnecessary data is present. For the analysis purpose regression helps to decide the variable which is suitable for our analysis.
* **Clustering:** This is used for finding the outliers and also in grouping the data. Clustering is generally used in unsupervised learning.

But this dataset had no null values and also did not require feature scaling.

**WHAT IS FEATURE SCALING?**

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

**Why feature scaling is not required in Random Forest Regressor?**

Role of Scaling is mostly important in algorithms that are distance based and require Euclidean Distance.

Feature scaling is essential for machine learning algorithms that calculate distances between data.

Random Forest is a tree-based model and hence **does not require** feature scaling.

**Data Analysis:**

Here we choose the model for the regression task. We have chosen Random Forest Regression model.

**Training the Model:**

The model is trained by importing the required model and by passing the training data to it. The dataset is split into train and test data with test\_size=0.20. The linear model is imported from sklearn and the Random forest regressor .These models are trained by passing the train data. While training the model we also do Hyperparameter Tuning.

**What is Hyperparameter Tuning?**

When creating a machine learning model, you'll be presented with design choices as to how to define your model architecture. Often times, we don't immediately know what the optimal model architecture should be for a given model, and thus we'd like to be able to explore a range of possibilities. In true machine learning fashion, we'll ideally ask the machine to perform this exploration and select the optimal model architecture automatically. **Parameters which define the model architecture are referred to as hyperparameters and thus this process of searching for the ideal model architecture is referred to as hyperparameter tuning.**

**Evaluation:**

In the end we do the evaluation of the model like how well is it predicting the gold price and also we use metrics such as the r2\_score to get a score as to how accurate is the model.

**What is r2\_score?**

R2\_score is the coefficient of determiniation between the output variable and the independent variable and it varies between 0% to 100%.

# LIBRARIES USED:

**NumPy**

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific

computation. NumPy’s main object is the homogeneous multidimensional array. It is a table with same type elements, i.e., integers or string or characters (homogeneous), usually integers. In NumPy, dimensions are called axes. The number of axes is called the rank. There are several ways to create an array in NumPy like np.array, np.zeros, no.ones, etc. Each of them provides some flexibility. We can import numpy library by below command:

**Import numpy as np**

# Pandas

Pandas is a Python library that is used for faster data analysis, data cleaning, and data pre-processing. Pandas is built on top of the numerical library of Python, called numpy. Before you install pandas, make sure you have numpy installed in your system. We can import numpy library by below command:

**Import pandas as pd**

# Matplotlib

Matplotlib is a Library used for plotting graphs in the [Python programming](https://www.javatpoint.com/python-tutorial) [language](https://www.javatpoint.com/python-tutorial). It is used plot 2 - dimensional arrays. Matplotlib is built

on [NumPy](https://www.javatpoint.com/numpy-tutorial) arrays. It is designed to work with the border [SciPy](https://www.javatpoint.com/python-scipy) stack. It was developed by John Hunter in 2002.

The benefit of visualization is that user can have visual access to large amounts of the dataset. Matplotlib is a library which is consists of various plots such as histogram, bar, line, scatter, etc.

Matplotlib comes with a huge variety of plots. Plots are helpful for understanding patterns, trends and for making correlations. It has instruments for reasoning about quantitative information.

As matplotlib was the very first library of [data visualization](https://www.javatpoint.com/what-is-data-visualization) in python, many other libraries are developed on top of it or designed to work parallel to it for the analysis of the dataset. We can import matplotlib library by below command:

**import matplotlib.pyplot as pyplt**

# Seaborn

Seaborn is a library of Python programming basically used for making statistical graphics of the dataset. This library is built on top of the Matplotlib library. It is also integrated closely with Pandas, which is used for the data structure of Datasets.

Seaborn is very helpful to explore and understand data in a better way. It provides a high level of a crossing point for sketching attractive and informative algebraic graphics. We can import seaborn library by below command:

**import seaborn as sns**

If using jupyter notebook, it is important to set %matplotlib inline to show the visualizations on the notebook.

# train\_test\_split

The reason is that when the dataset is split into train and test sets, there will not be enough data in the training dataset for the model to learn an effective mapping of inputs to outputs. There will also not be enough data in the test set to effectively evaluate the model performance. We can import the library by below command:

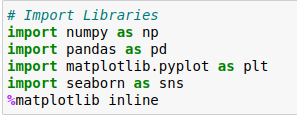
**From sklearn.model\_selection import train\_test\_split**

**WALKTHROUGH**

In this project, I used python3 and Jupyter notebook.

**IMPORT DEPENDENCIES**

The first thing that we have to do, is we have to import the necessary dependencies, that we will be using in the upcoming part of the program. Here in this project, we shall be using numpy, pandas, matplotlib, seaborn and sklearn.

****

**IMPORT THE DATASET**

We will use pandas module to read the csv file.

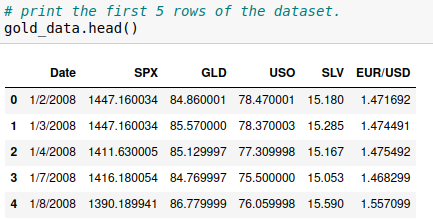


The dataset will get loaded in the form of a DataFrame.

**READING THE DATA FROM THE DATASET**

To get a look at how the data was stored in the variable, we use the command variable\_name.head(), to see the first five rows of the table.

This command prints the first five rows of the dataset.

****

There are columns with names like SPX,GLD and USO.

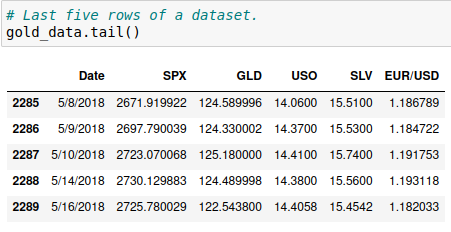
SPX stands for stock price exchange

GLD is our target variable which we want to predict using RF.

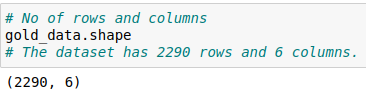
USO is US Oil ETF while SLV stands for silver price.

ETF is exchange traded fund: a type of investment fund that is traded on a stock exchange.

Like the head of the dataset we can also print the tail.



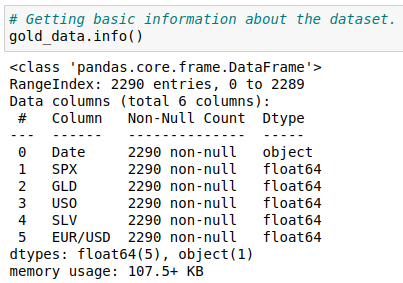
Now, let’s check how many rows and columns do we have in the dataset.



This tells us that there are 2290 rows and 6 columns.

Columns are called by different names in ML like variables or features so there are 6 features in this dataset including the target variable or feature.

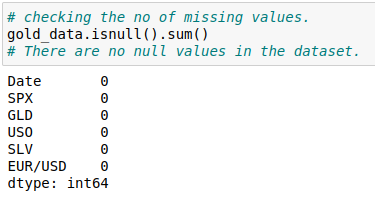
To get an overview of the dataset we use this command:



This gives us some basic info about the dataset like the datatypes, memory usage, again it gives us the no of rows and columns and also it tells us if there are any missing values.

We can see clearly that there are no missing values in this dataset.

But let’s check it with another command.



As we can see the zero values against each column telling us that there are no null values in this dataset.

**BUT WHAT TO DO IF WE HAVE NULL VALUES?**

There are three types of missing data:

**MCAR:** Missing Completely At Random. It is the highest level of randomness. This means that the missing values in any features are not dependent on any other features values. This is the desirable scenario in case of missing data.

**MAR**: Missing At Random. This means that the missing values in any feature are dependent on the values of other features.

**MNAR**: Missing Not At Random. Missing not at random data is a more serious issue and in this case, it might be wise to check the data gathering process further and try to understand why the information is missing. For instance, if most of the people in a survey did not answer a certain question, why did they do that? Was the question unclear?

**What to do with the missing values?**

Now that we have identified the missing values in our data, next we should check the extent of the missing values to decide the further course of action.

**Ignore the missing values**

* Missing data under 10% for an individual case or observation can generally be ignored, except when the missing data is a MAR or MNAR.
* The number of complete cases i.e. observation with no missing data must be sufficient for the selected analysis technique if the incomplete cases are not considered.

**Drop the missing values**

**Dropping a variable**

* If the data is MCAR or MAR and the number of missing values in a feature is very high, then that feature should be left out of the analysis. If missing data for a certain feature or sample is more than 5% then you probably should leave that feature or sample out.
* If the cases or observations have missing values for target variables(s), it is advisable to delete the dependent variable(s) to avoid any artificial increase in relationships with independent variables.

**Case Deletion**

In this method, cases which have missing values for one or more features are deleted. If the cases having missing values are small in number, it is better to drop them. Though this is an easy approach, it might lead to a significant decrease in the sample size. Also, the data may not always be missing completely at random. This may lead to biased estimation of parameters.

**Imputation**

Imputation is the process of substituting the missing data by some statistical methods. Imputation is useful in the sense that it preserves all cases by replacing missing data with an estimated value based on other available information. But imputation methods should be used carefully as most of them introduce a large amount of bias and reduce variance in the dataset.

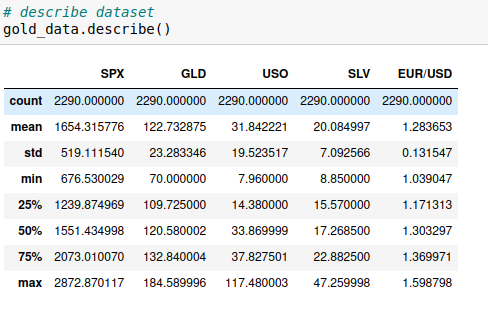
**Imputation by Mean/Mode/Median**

If the missing values in a column or feature are numerical, the values can be imputed by the mean of the complete cases of the variable. Mean can be replaced by median if the feature is suspected to have outliers.

**(What are outliers?**

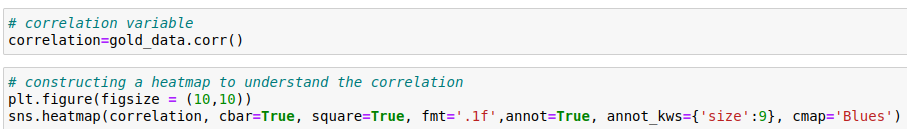
An outlier is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data.)

For a categorical feature, the missing values could be replaced by the mode of the column. The major drawback of this method is that it reduces the variance of the imputed variables. This method also reduces the correlation between the imputed variables and other variables because the imputed values are just estimates and will not be related to other values inherently.



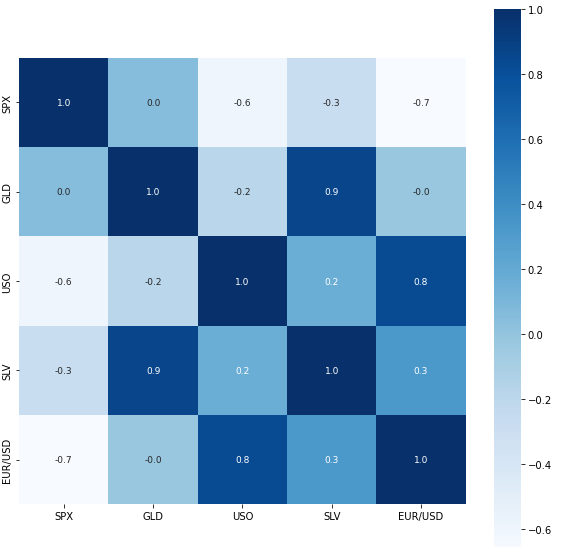
This describe command also gives some useful information like the mean, standard deviation and various percentile ranges.

Next we move on to find some correlation among the features.



**WHAT IS A HEATMAP?**

A heat map is a two-dimensional representation of information with the help of colors. Heat maps can help the user visualize simple or complex information. The correlation coefficient ranges from –1 to 1. When it is close to 1, it means that there is a strong positive correlation and when the coefficient is close to –1, it means that there is a strong negative correlation; like for example in the case of GLD that is the gold price we can see that it is negatively correlated to USO. Finally, coefficients close to zero mean that there is no linear correlation.



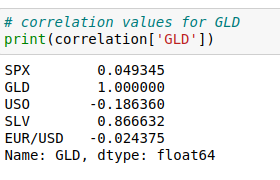
There are some parameter in the code sns.heatmap that help us to visualize data better.

**Explanation** 

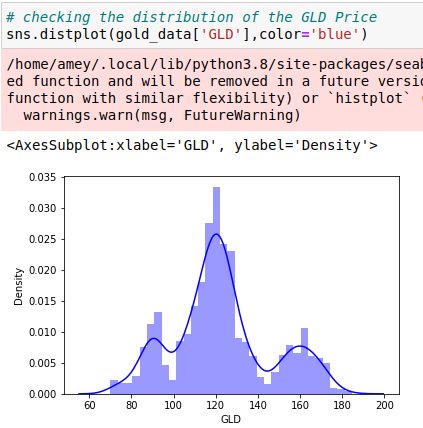
* annot is used to annotate the actual value that belongs to these cells.
* Annot\_kws is used to basically control the size of the names of the columns.
* cmap is used for the colour mapping you want like coolwarm, plasma, magma etc.
* fmt=’.1f’ means that we want the values in the heatmap to have 1 digit after the decimal.
* linecolor is used to set the colour of the lines separating the cells.

So our target variable as we know by now is the GLD price.

Therefore we can also see the correlation for the GLD specifically.



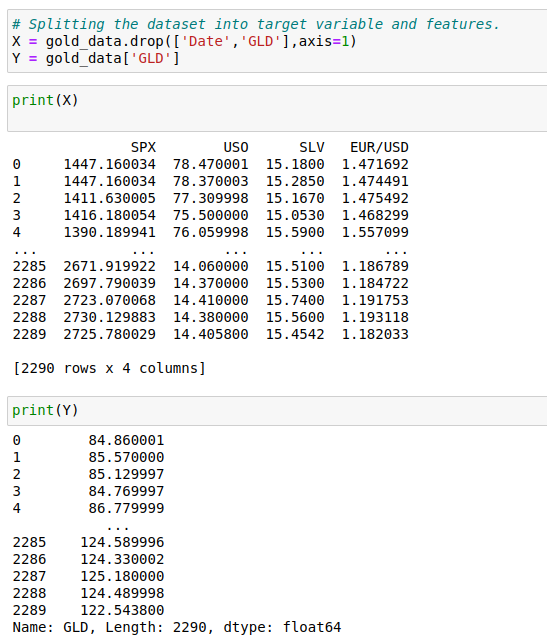
Next we look for the distribution of GLD price.



**WHAT IS A SNS DISTPLOT?**

Seaborn distplot lets you show a histogram with a line on it. This can be shown in all kinds of variations. We use seaborn in combination with matplotlib, the Python plotting module. A distplot plots a univariate distribution of observations.

**Let’s Split the data into target values and independent feature values**



Here, X is the feature variable, containing all the features like **SPX**, **USO**, **SLV**, etc., on which the price of gold depends, excluding the **GLD**and**Date**column itself.

Y, on the other hand, is the target variable, as that is the result that we want to determine i.e., the price of Gold. (It contains only the **GLD** column).

**Splitting X and Y into training and testing variables**

Now, we will be splitting the data into four variables, viz., X\_train, Y\_train, X\_test, Y\_test.

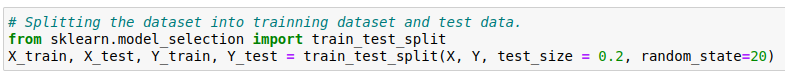
**BUT WHY DO WE SPLIT THE DATASET INTO TRAINING AND TESTING DATASET?**

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

It can be used for classification or regression problems and can be used for any supervised learning algorithm.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

By input element we mean the Y\_train set which is fed to the model for it to predict Y\_test.

****

Random\_state is the way in which the dataset is split. So if someone wants to get the exact split the way in which it is split in this case, he/she has to mention the same number i.e. 20.

Let’s understand the variables by knowing what type of values they store:

X\_train: This includes your all independent variables, that will be used to train the model, also as we have specified the test\_size = 0.2, this means 80% of observations from your complete data will be used to train/fit the model and rest 20% will be used to test the model.

Y\_train: This is your dependent variable which needs to be predicted by this model, this includes category labels against your independent variables, we need to specify our dependent variable while training/fitting the model.

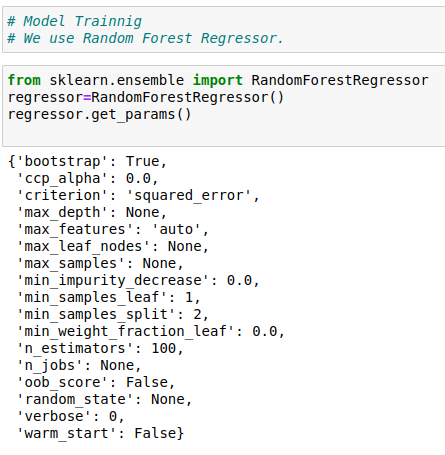
X\_test: This is remaining 20% portion of the independent variables from the data which will not be used in the training phase and will be used to make predictions to test the accuracy of the model.

Y\_test: This data has category labels for your test data, these labels will be used to test the accuracy between actual and predicted categories.

test\_size: represents the ratio of how the data is distributed among X\_train and X\_test (Here 0.2 means that the data will be segregated in the X\_train and X\_test variables in an 80:20 ratio). You can use any value you want. A value < 0.3 is preferred.

**Model Training: Random Forest Regressor**

Here we name our model ‘**regressor**‘.



Regressor.get\_params() gets us the various hyperparameters used in the Regressor.

Let’s understand what each of these mean.

**Bootstrap:*bool, default=True***

Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

**ccp\_alpha:*non-negative float, default=0.0***

Complexity parameter used for Minimal Cost-Complexity Pruning. The subtree with the largest cost complexity that is smaller than ccp\_alpha will be chosen. By default, no pruning is performed.

**Criterion *{squared\_error, absolute\_error, poisson}, default=squared\_error***

The function to measure the quality of a split. Supported criteria are “squared\_error” for the mean squared error, which is equal to variance reduction as feature selection criterion, “absolute\_error” for the mean absolute error, and “poisson” which uses reduction in Poisson deviance to find splits. Training using “absolute\_error” is significantly slower than when using “squared\_error”.

**max\_depth*int, default=None***

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.

**max\_features:*{“auto”, “sqrt”, “log2”}, int or float, default=”auto”***

The number of features to consider when looking for the best split.

**max\_leaf\_nodes:*int, default=None***

Grow trees with max\_leaf\_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity. If None then unlimited number of leaf nodes.

**max\_samples:*int or float, default=None***

If bootstrap is True, the number of samples to draw from X to train each base estimator.

**n\_estimators (*int), default=100***

The number of trees in the forest.

**n\_jobs:*int, default=None***

If the value of n\_jobs=1 it means 1 processor is dedicated to the processes and -1 means all available processors are dedicated.

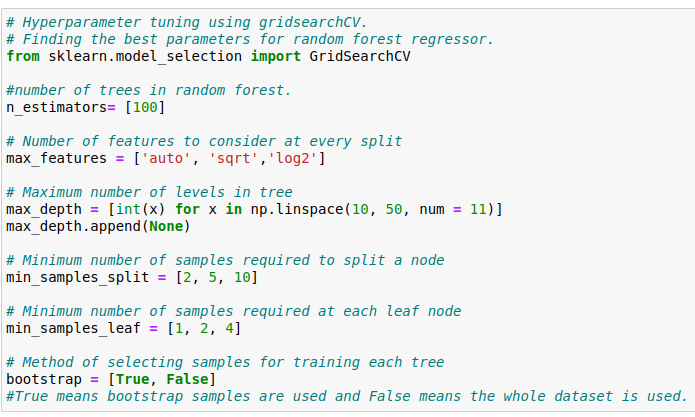
**oob\_score:*bool, default=False***

Whether to use out-of-bag samples to estimate the generalization score. Only available if bootstrap=True.

**Verbose:*int, default=0***

Controls the verbosity when fitting and predicting.

**HYPERPARAMETE TUNING**

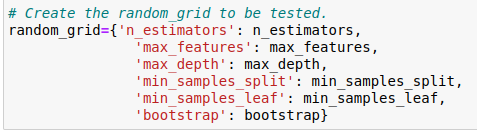


**WHAT IS GRIDSEARCHCV?**

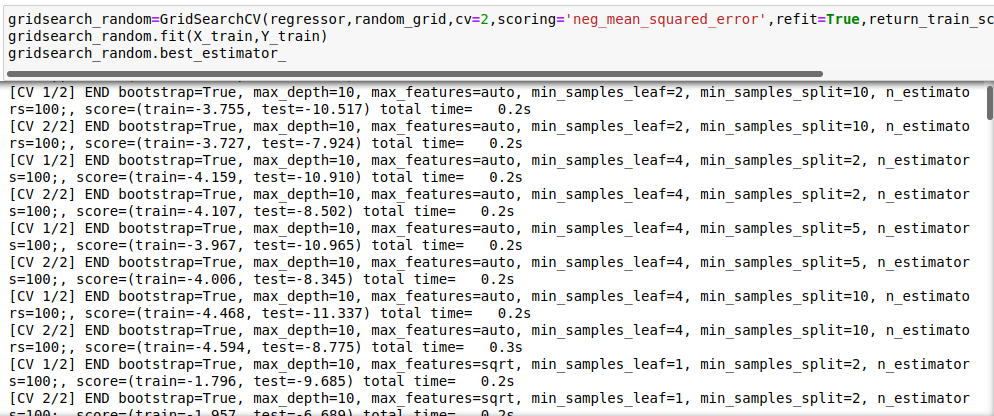
Grid search is arguably the most basic hyperparameter tuning method. With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results.

For example we would define a list of values for each hyperparameters that we want to tune and then run gridsearch on it.

Above we have defined the hyperparameters and their values and now we make a random grid containing all those parameters.



Now we run the gridsearch on the random grid.



After running the gridsearch on X\_train,Y\_train .



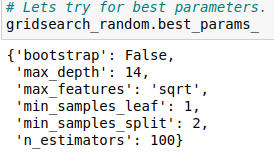
We get the best estimators for Random Forest Regressor.

**WHAT ARE BEST ESTIMATORS?**

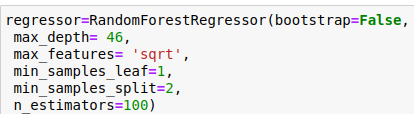
**best\_estimator:*estimator***

Estimator that was chosen by the search, i.e., estimator which gave highest score (or smallest loss if specified) on the left-out data.

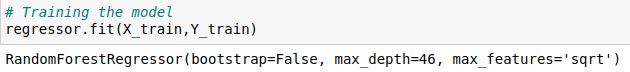
Or else we can also use best parameters command to find out the best parameters determined by the grid search.



So we fill our regressor with all these values.

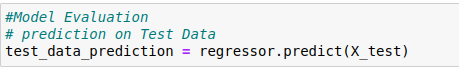


Now let us train the model, with our values containing the training dataset which are (X\_train, Y\_train).

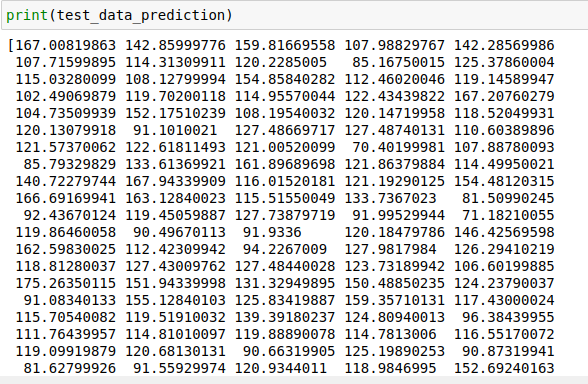


**MODEL EVALUATION**

Let’s now predict the values of the X\_test dataset using the predict () method.

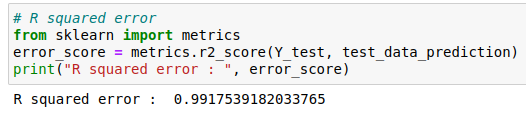


And print the test\_data\_prediction.



This the prediction done by our model which should be compared with Y\_test data.

Calculating the R-Squared error from the predicted value.



The R squared error comes out to be 0.991 which is an excellent score. We might also interpret this like our model is 99% accurate.

**COMPARING THE ACTUAL VALUES AND PREDICTED VALUES**

Now we compare the values visually.

We first convert the values of Y\_test into a list.



Now, plotting values of actual prices, versus the predicted prices to know, how close our predictions were to the actual prices:



We have plotted actual values against predicted values and we can see that the model works really good.

**RESULTS**

With the help of python machine learning, Random Forest Algorithm gives a R squared\_score of 99%.

**CONCLUSION**

With the help of the R squared error score, we can conclude Random Forest model is the best for prediction in regression problems

The important feature which can affect Gold Price is Silver Price (SLV).

The models we explore are tools of true utility replete with practical real-world applications.

**REFERANCES**

* 1. Iftikhar ul Sami, Khurum Nazir Junejo. ‘Predicting Future Gold Rates using Machine Learning Approach’.
  2. [Random Forest Regression. Random Forest Regression is a… | by Chaya Bakshi | Level Up Coding (gitconnected.com)](https://levelup.gitconnected.com/random-forest-regression-209c0f354c84)
  3. [Hyperparameter tuning for machine learning models. (jeremyjordan.me)](https://www.jeremyjordan.me/hyperparameter-tuning/)
  4. [What is Supervised Learning? | IBM](https://www.ibm.com/cloud/learn/supervised-learning)
  5. [Feature scaling - Wikipedia](https://en.wikipedia.org/wiki/Feature_scaling)
  6. [How to Deal with Missing Values in Your Dataset - KDnuggets](https://www.kdnuggets.com/2020/06/missing-values-dataset.html)
  7. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron.
  8. 1M.Sravani, 2Ch.Abhilash, 3T.Divya, 4Ch.Vasthav, 5D.Priyanka,”GOLD PRICE PREDICTION”, International Journal of Creative Research Thoughts (IJCRT), 2021.