

Computer Science

Extended Essay

Comparing SVR algorithm and Random Forest Regression algorithm in currency value prediction

To what extent can the efficiency of Support Vector Regression and Random Forest Regression be compared for predictive accuracy and computational complexity when predicting currency value based on gold price?

Word Count : 3919

Table of contents

Introduction	2
Background Information	3
Simple Linear-Regression	3
Support Vector Regression	4
Linear Support-Vector-Regression	4
Non-linear Support-Vector-Regression	8
Kernal Trick 15	8
Radial Basis Function	13
Random Forest Regression	14
Experiment Methodology	19
Dependent and Independent variables	19
Control variables	21
Procedure	21
Data	25
Metrics	27
Output (Graphs)	28
Training-set (SVR)	28
Training-set-smooth(SVR)	29
Testing-set	29
Training-set (RFR)	30
Training-set-smooth(RFR)	30
Testing-set	31
Comparing with metrics and Evaluation	32
Predictive Accuracy	32
Time complexity	35
Conclusion & Further Scope of Research	37
Bibliography	39
Appendix	43

Introduction

The currency value of a country determines its economic well-being and its value in the global market for trade, where gold prices affect ¹the currency value, posing as an essential commodity for trade. It is important for governments to predict currency values to take action accordingly to prevent factors like recession and boost the economy's GDP.

These predictions of the currency value can be done using machine learning algorithms such as support vector regression and random forest regression. Hence, I will be researching '**To what extent can the efficiency of Support Vector Regression and Random Forest Regression be compared for predictive accuracy and computational complexity when predicting currency value based on gold price?**'. Gold price of India will be the input value which predicts India's currency value (value of INR compared to 1 US dollar). This investigation is worth because it might bring a change and technological advancement in the governments economic decision as the machine learning algorithms can predict required values.

Support vector regression is a machine learning algorithm that was introduced in 1960s by Vladimir Vapnik² and used for regression tasks like predicting continuous numbers with inputs. It is an extension of the Support Vector Machine algorithm, which is primarily used for classification, assigning input points to a predicted class/category. SVR is designed to predict a

¹How Gold Affects Currencies." *Investopedia*, 16 May 2011,
www.investopedia.com/articles/forex/11/golds-effect-currencies.asp.

²Lesson 10: Support Vector Machines | STAT 508." *PennState: Statistics Online Courses*,
online.stat.psu.edu/stat508/lesson/10.

continuous target variable effectively such as real numbers rather than categorizing data into classes.

Random Forest Regression on the other hand, is a machine learning algorithm which was introduced in 2000s³. It combines the power of decision trees and ensemble learning techniques for regression tasks where an extension of the Random Forest algorithm is designed to predict a continuous target variable (regression), rather than classify data into categories like the SVM. This algorithm provides more advantages such as requiring minimal preparation of data, simplicity in visualisation and implementation.

Background Information

Simple Linear-Regression

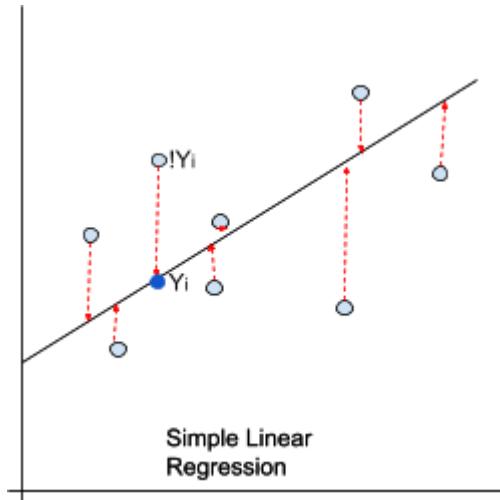


Figure 1.1

It is important to understand simple-linear-regression to get better understanding about support-vector-regression.

³"Random Forest Versus Logistic Regression: a Large-scale Benchmark Experiment." *BioMed Central*, 17 July 2018, bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5

As seen in the figure 1.1, the *simple linear regression*⁴(simply best-fit-line), is formed with the ordinary least squared method, where the linear equation is formed using this formula

$$Y_i = \beta_0 + \beta_1 X_i$$

5

Simple linear regression formula

which is the summation of ($y=mx+c$) linear equation of every point.⁶ The relationship between variables is found and the linear line is formed and positioned. Along with the distance between each actual data point($!Y_i$) and the predicted point(Y_i), the red line of all data points seen in figure 1.1 is minimized to get a simple linear regression with minimum errors.

Support Vector Regression

Linear Support-Vector-Regression

⁴"Simple Linear Regression." wwwjmp.com/en_in/statistics-knowledge-portal/what-is-regression.html.

⁵Agrawal, Anushka. "Simple Linear Regression Modeling-Part 1." *Medium*, 8 May 2021, medium.com/nerd-for-tech/simple-linear-regression-modeling-part-1-1ae3b59c6ab5.

⁶Simplilearn. "What is Simple Linear Regression in Machine Learning?" *Simplilearn.com*, 22 Aug. 2022, www.simplilearn.com/what-is-simple-linear-regression-in-machine-learning-article#.

"Numeracy, Maths and Statistics - Academic Skills Kit." *The Things We Do Here Make a Difference out There / Newcastle University*, www.ncl.ac.uk/webtemplate/ask-assets/external/mathss-resources/statistics/regression-and-correlation/simple-linear-regression.html.

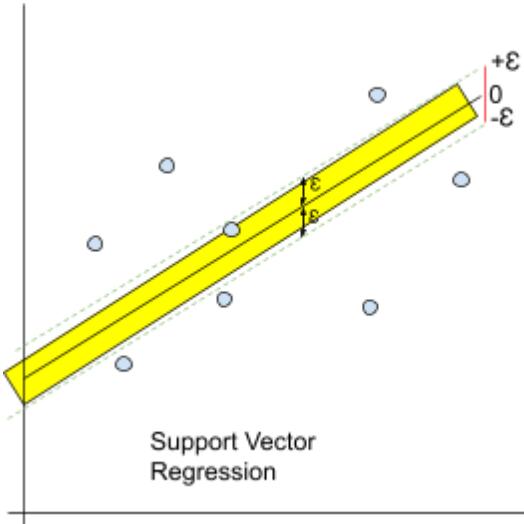


figure 1.2

In the figure above, the data points in 2d space (independent & dependent variable) are called **data points** and the data points in 3d space (2 independent & 1 dependent variable) are called **vectors**⁷.

Similar to simple-linear-regression, in the Support-Vector-Regression, there is a tube (*epsilon insensitive* tube (*yellow*)) formed instead of a single line. This tube is formed with the help of the simple linear equation (used in figure 1.1) formed in the middle (0), and has the width of $\text{epsilon}(\epsilon)$ ⁸ measured vertically like in figure 1.2. This formula is used in order to create the SVR equation with the tube:

⁷ "What is a Vector -- Computing for All." *Computing for All*, 30 July 2023, wwwcomputing4allcom/courses/introductory-data-science/lessons/what-is-a-vector/#..

⁸Singh, Navjot. "Support Vector Regression for Machine Learning." *Medium*, 18 June 2020, medium.com/@analyticsvidhya/support-vector-regression-for-machine-learning-843978ba6279.

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \xi_i^*) \rightarrow \min$$

9

This tube is also called the *margin of error* where the predicted values can deviate from the actual value if it's within this tube. If the values are there **inside** the tube and **not** accurate, they will **not** be considered as an error. The data points outside the tube (*slack variables*) are **counted** if they produce errors or are inaccurate compared to the actual value. This algorithm tries to cover all of the datapoints inside its epsilon width to predict *accurately*.

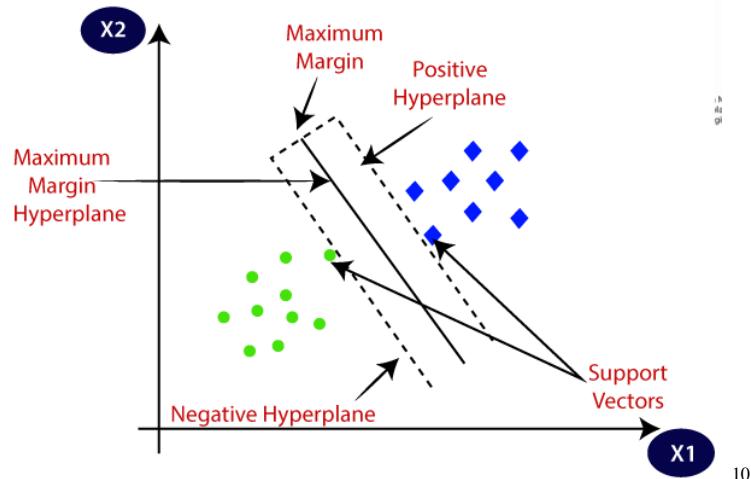


Figure 1.3

The points closer to the margin are called *support vectors* as seen in figure 1.3. They decide the width of the epsilon(margin) , the position and degree of slopeness, and most importantly,The width of the tube is the sum of distance between the support vector and regression line which is

⁹ ---. "Support Vector Regression for Machine Learning." Medium, 18 June 2020, medium.com/analytics-vidhya/support-vector-regression-for-machine-learning-843978ba6279.

¹⁰ "Support Vector Machine (SVM) Algorithm - Javatpoint." www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

the width of margin of error, the $\text{epsilon}(\epsilon)$. This margin of error can be formed with only 2 support vectors and no other data.

Hence, the *loss function*¹¹ which is the distance between the predicted and actual value is reduced. Similar to the ordinary least squared method, the loss function includes two components, one for errors outside the tube, and one for errors within the tube(which is ignored), where it minimizes the sum of errors within the epsilon. It keeps the margin (the tube width) as wide as possible where these points are giving flexibility (buffer) to the SVR by ignoring the error, making it differ and accurate from Simple-linear-regression.

The regularization parameter "C"¹² is used to balance the trade-off between minimizing errors and maximizing the margin width. This part of the positioning, and minimizing the error of the epsilon tube can be taken care of through techniques such as quadratic programming¹³.

In the figure above, (figure 1.2), the straight line is categorized as a classifier and not hyperplane ¹⁴because it is a 2D dataset/space where one dependent and one independent variable are present.

The ultimate goal for the SVR is to find the decision boundary. Additionally, the main **advantage** of SVR is that the algorithm takes the most extreme data point, classifies it as another variable, and trains it so that it transforms to the *extreme chances/prediction cases* making the

¹¹ "Support Vector Regression." SpringerLink, link.springer.com/chapter/10.1007/978-1-4302-5990-9_4#.

¹² "SVM 'C' parameter." Just a Moment., www.baeldung.com/cs/ml-svm-c-parameter#.

¹³"Quadratic Programming." Cornell University Computational Optimization Open Textbook - Optimization Wiki, optimization.cbe.cornell.edu/index.php?title=Quadratic_programming. Accessed 25 Jan. 2024.

¹⁴ "Separating Hyperplanes in SVM." GeeksforGeeks, 15 Sept. 2021, www.geeksforgeeks.org/separating-hyperplanes-in-svm/.

algorithm more accurate(at most times support vectors which decide the slope angle are the extreme data).

Non-linear Support-Vector-Regression

Kernal Trick¹⁵

When the data is not linearly separable, an optimal decision boundary is required for the algorithm to identify the difference between the data points as seen in the figure 2.1. which can be constructed with the help of higher dimensional space. These data points are taken and an extra dimension is added, which makes the data linearly separable, once it is converted back into 2d space it is non-linear SVR

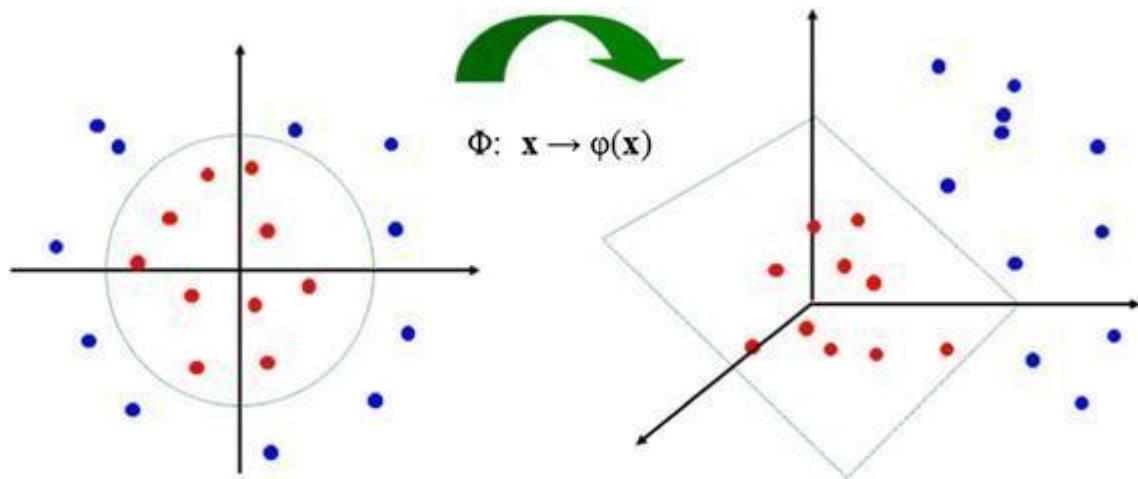


Figure 2.1

¹⁵Yadav, Suraj. "What is Kernel Trick in SVM ? Interview Questions Related to Kernel Trick." *Medium*, 29 Apr. 2023, medium.com/@Suraj_Yadav/what-is-kernel-trick-in-svm-interview-questions-related-to-kernel-trick-97674401c48d#.

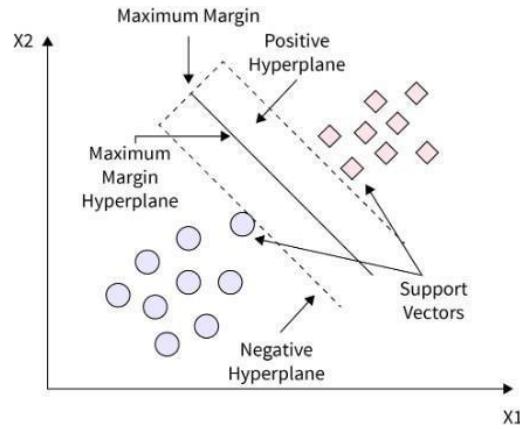


Figure 2.2

Lets assume the 2 types of data are not linearly separable, one set is in the middle of other as seen in figure 2.1. Hence the following formula of Kernal SVR(Radial basis function) is used.

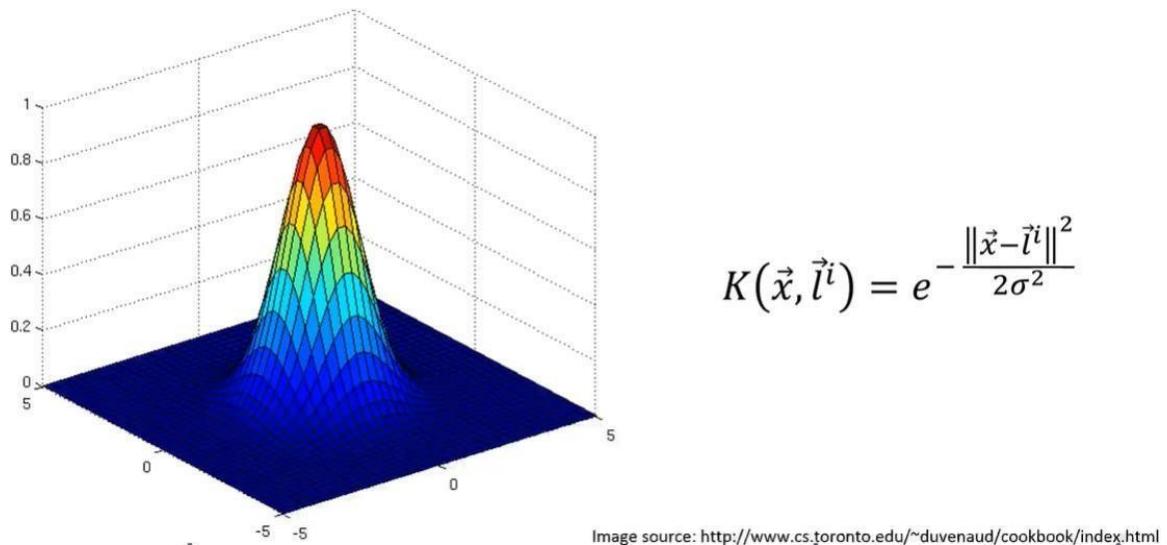


Figure 2.3

‘ x ’ are the points in the bottom dark blue plate which has only 2 dimensions at the bottom of the figure, ‘ l ’ is the landmark, ‘ i ’ is the serial number of the landmark. In this case, the ‘ l ’ is in the center of the plate, below the peak of the kernel (at 0,0 as seen in figure 2.3), ‘ σ ’ is a fixed

parameter. the numerator of the exponent is the distance between landmark ' l ' and the point ' x ' which is getting squared. The vertical axis represents the result of the calculation, with respective points in the plate (output). This formula states that as the distance between a data point and a landmark increases, the value of the radial basis function kernel decreases. When a data point is closer to a landmark, the kernel value increases, approaching the center (0,0) creating a peak or 'mountain' structure in the kernel's hypersurface as seen in the figure 2.3.

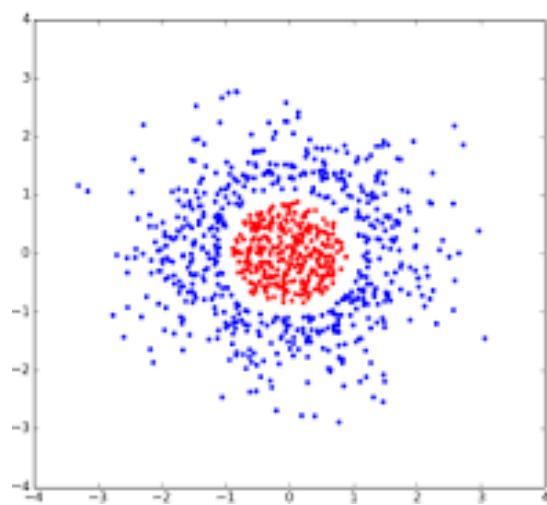
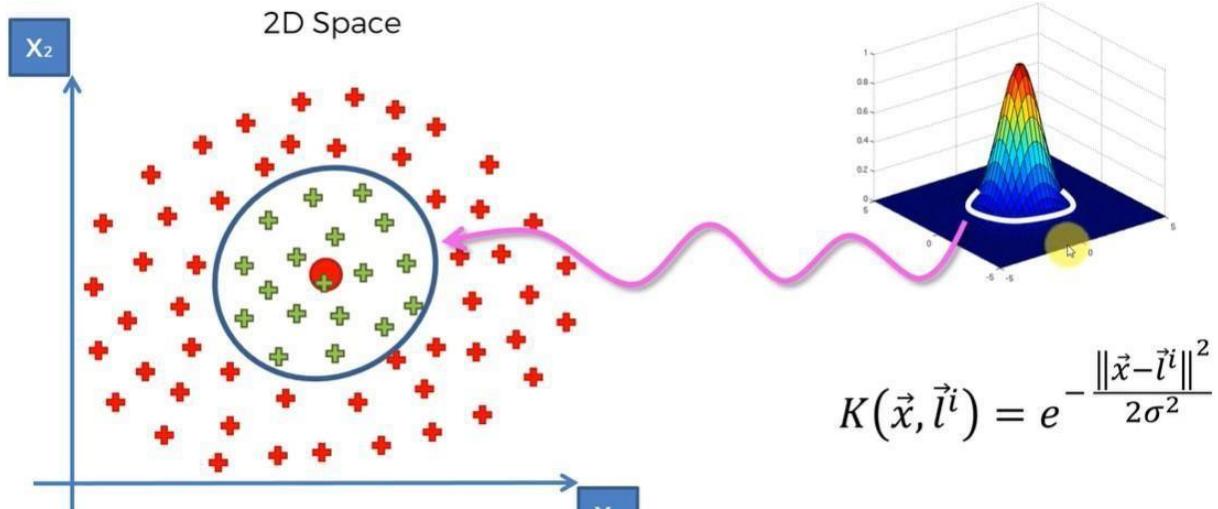


Figure 2.4

If we transform that hyperspace into a 2d space we get the above figure 2.4 and 2.5, Now the new landmark of this is the midpoint of the red variable data set in figure 2.4. After using the formula, a kernel is created like the one above(figure 2.5) allowing us to separate non-linear datasets. Since the boundary is created like this, any kernel value resulting from the formula close 0 is a red variable(any variable that has the value to the center(0,0)) and not close to 0 is a blue variable in figure 2.4. The circumference of this circle is decided by the ' σ ' value. This is how the non-liner-SVR is formed. To summarise for support vector regression which has data

points like this, the radial basis function is formed and the intersection between that and the hyper-line formed is the non-linear support vector regression.



16

Figure 2.5

When there is multidimensional space where there are more than two variables, the support vector points (closer to the margin tube) are not points as they are supposed to be visualized in a 2 dimensional diagram, those are turned into vectors due to the 3rd dimension and has direction & magnitude, as seen in the above figure 2.1.

The classifier, or the epsilon tube, will now be a hyperplane where the hyperplane above the regression line is called a positive hyperplane and the one below is a negative hyperplane as seen in the figure 2.2.

¹⁶ "Kernel Trick: A Non-linear Decision Boundary for Support Vector Machine Learning. | Taylor Ortiz Posted on the Topic | LinkedIn." *LinkedIn*, 20 Nov. 2023, www.linkedin.com/posts/anthonytortiz_using-the-kernel-trick-to-create-a-non-linear-activity-7132365409428656129-lkp/?trk=public_profile_share_view.

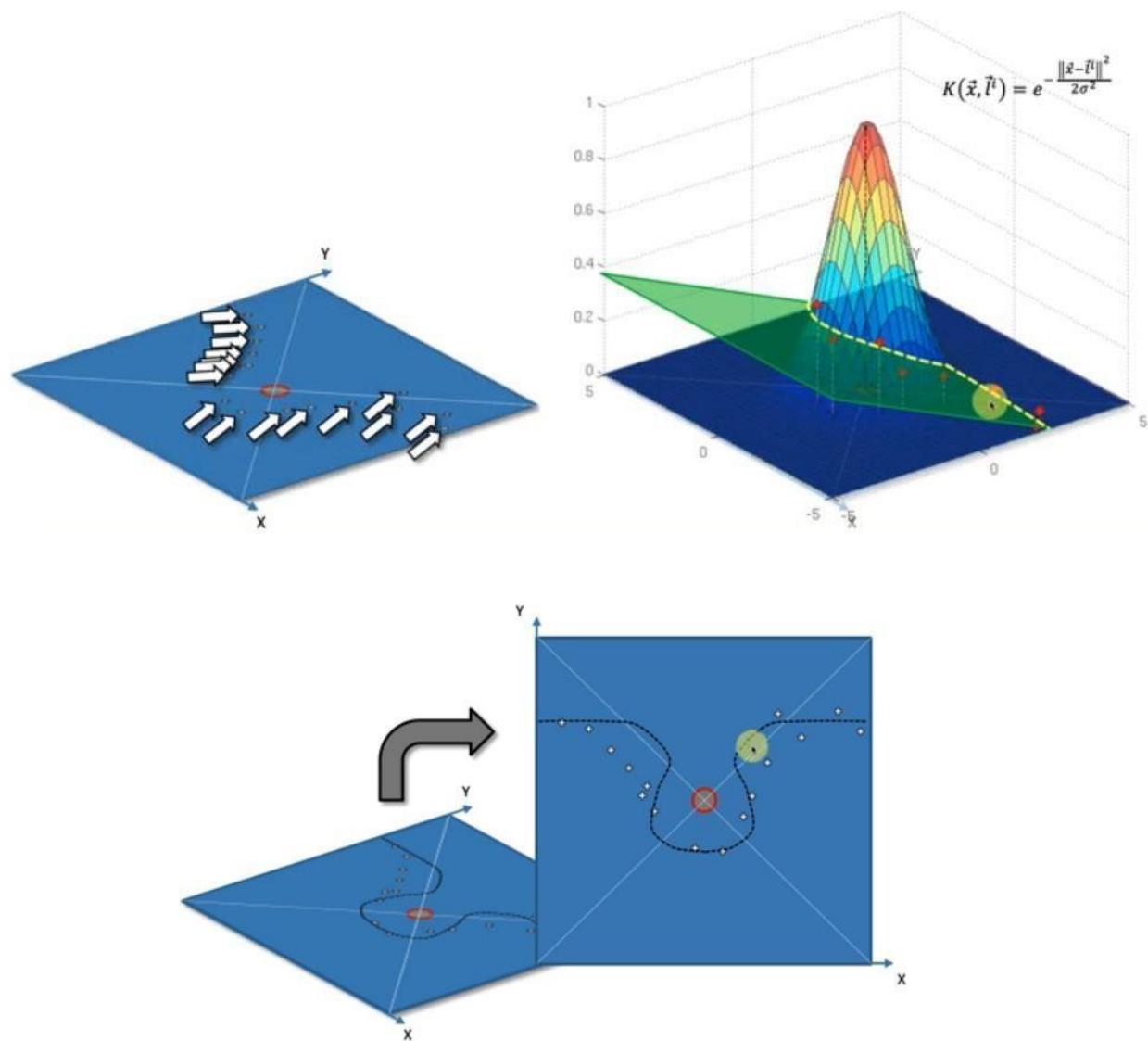


Figure 2.6

The above figure 2.6 is an representation of how the non-linear data is being converted into 3rd space and how the hyperlane is formed with the help of support vectors(the red dots)

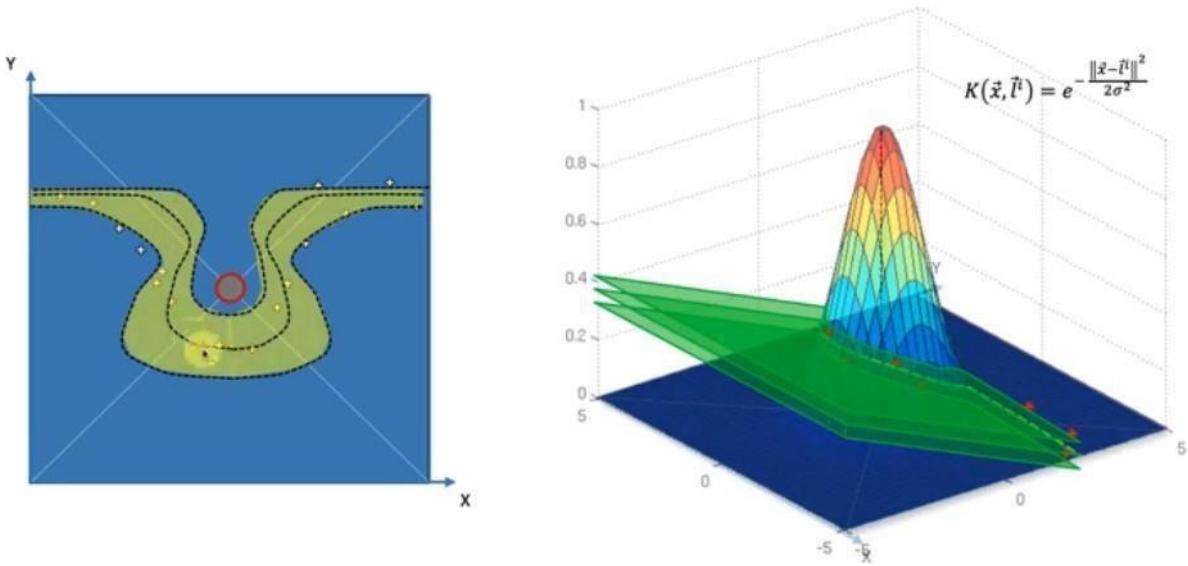


Figure 2.7

The top and bottom are the higher and lower epsilon like the ones in linear in the above figure 1.2, so initially the RBF function is bought and the intersection of the green plates in figure 2.7 is taken with the help of support vectors(the red dots near mountain structure , figure 2.7) by placing them in the correct position with right thickness (marginal error) and minimum error.

One **disadvantage** of this algorithm is requiring excessive processing power and time, especially when it converts the dimensions. This can be minimized by different types of *kernel functions*, we use *Kernals SVR (Radial basis function¹⁷)* in this essay.

Radial Basis Function

This RBF performs the same process of the kernel trick with implicitly adding an additional dimension. But this is not a physical dimension. Instead, the relationships between data points

¹⁷Sreenivasa, Sushanth. "Radial Basis Function (RBF) Kernel: The Go-To Kernel." Medium, 12 Oct. 2020, towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a.

are computed as if they were in a higher-dimensional space, which is the RBF and an **advantage** as it uses less computational resources(ex:time).

Random Forest Regression

Primarily, we need to understand Decision tree algorithm- CART¹⁸(classification and regression trees) Considering there are two independent variables and 1 dependent variables as seen in figure 3.1 :

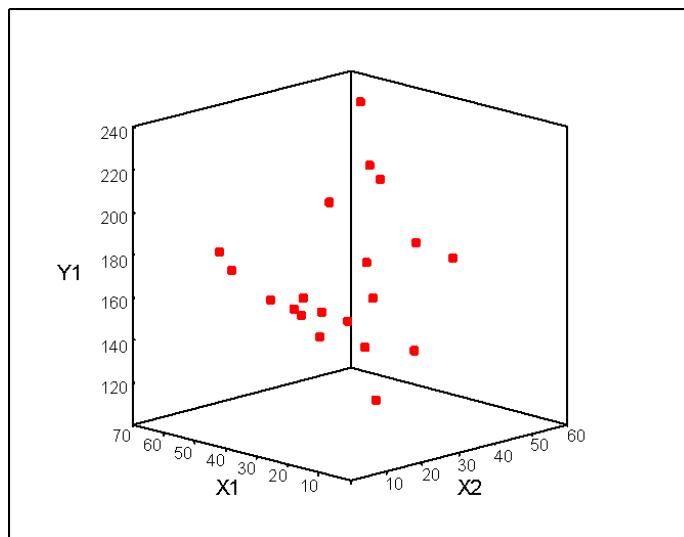


Figure 3.1

Now the algorithm takes the two independent variables and forms segments by randomly splitting the datas,(a split is called a *leaf*¹⁹) as seen in figure 3.2 :

¹⁸"CART (Classification And Regression Tree) in Machine Learning." GeeksforGeeks, 4 Dec. 2023, www.geeksforgeeks.org/cart-classification-and-regression-tree-in-machine-learning/.

¹⁹ Deepankar. "Decision Trees with CART Algorithm." Medium, 22 Apr. 2021, medium.com/geekculture/decision-trees-with-cart-algorithm-7e179acee8ff.

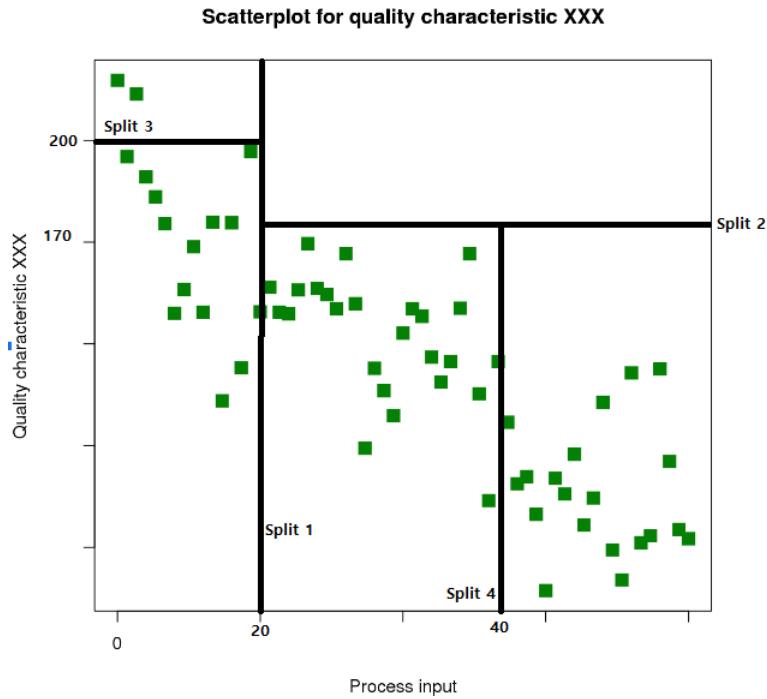


Figure 3.2

How the *leaves* (split) occurs is determined by the algorithm using the help of information entropy which is a mathematical concept with the following formula.

$$H = - \sum_{i=1}^N p_i \log(p_i)$$

Therefore, in this case, *information entropy* is used for using the correct leaf on the appropriate spot where the amount of information is maximized (the data points). Thus it adds value to the way it wants to perform segmentation (classifying). This process (algorithm) stops when there is less need for information to be added and once it cannot add any valuable information leaf, specifically when there is less than 5% of data points left, it stops reaching the terminal leaves (final splits).

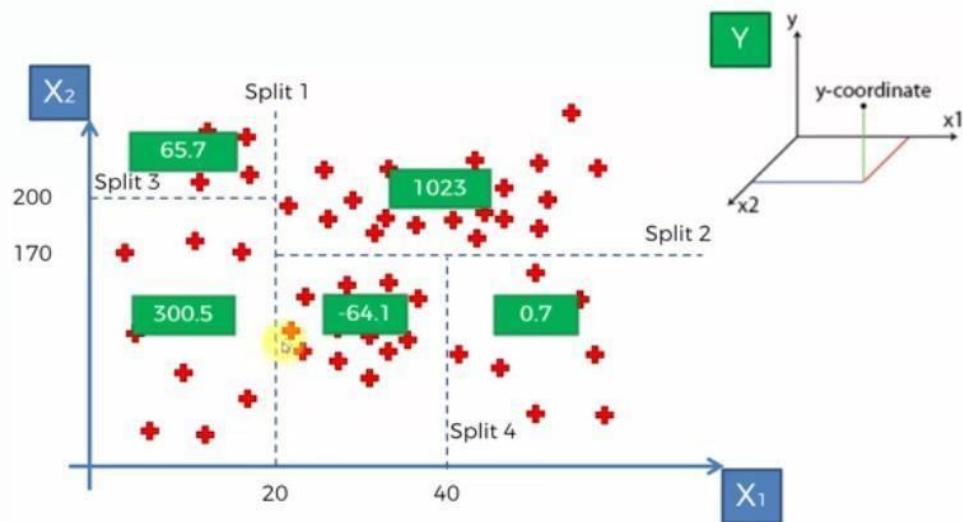


Figure 3.3

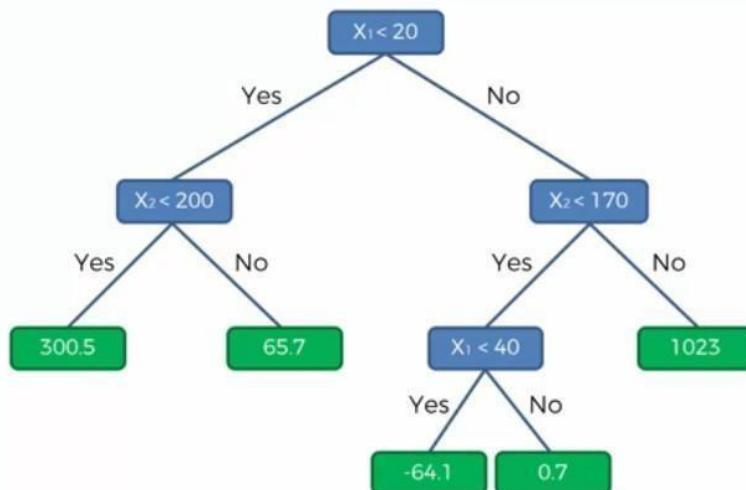


Figure 3.4

In the case of 2 independent variables , since there is a split, a decision tree is constructed ²⁰for the firure 3.3. In the above figure 3.4 the first considerable split happens at 20, so the first

²⁰ "DECISION TREES : CLASSIFICATION AND REGRESSION TREES (CART)." National Institute of Science Education and Research, www.niser.ac.in/~smishra/teach/cs460/23cs460/lectures/lec9.pdf.

decision of the tree is $X_1 < 20$, yes or no. Subsequently, there is a split at 170 of X_2 and as this split is only for X_2 s, which are greater than $X_2 > 20$, this falls under the right branch and forms a condition of $X_2 < 170$, yes or no. Then split 3 at $X_2 = 200$ occurs and as it is for less than $X_1 < 20$ it falls under the left branch. After the decision tree is constructed, the dependent variable is predicted by the average value of the terminal leaf (the segment) of the dependent variable which will be assigned to the data points falling under the terminal leaf. For example, when the X_1 is 30 and X_2 is 50, it falls under the bottom middle terminal leaf as seen in figure 3.5. Hence, it'll take the average of that and assign that value as predicted value as -65.3.

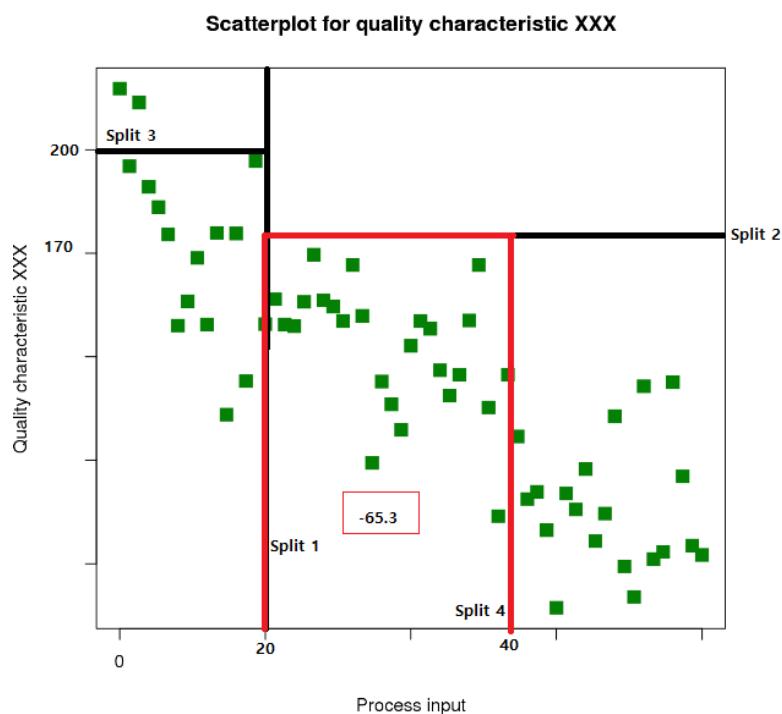


Figure 3.5

The key **advantage** of this algorithm is taking the average of efficiently split terminal leafs which gives more accurate results, rather than the entirety of datapoints like a few machine learning algorithms.

Using this decision tree algorithm, random forest regression is formed with *Ensemble learning*²¹(which is taking multiple algorithms or single algorithms multiple times to make the algorithm more powerful).

Step 1 : Picks a random number of data points from the data set

Step 2 : Build the decision tree associated with these random data points.

Instead of building the decision tree for the whole data set it builds only on chosen data points

Step 3: chooses the number of trees needed to build and repeats Steps 1 & 2(builds a lot of regression trees) with different data points in the set

Step 4: using all the trees built in step 3 the new data point(independent variable) is predicted by each tree, the average of all these predicted values of the data point of the trees is assigned and predicted by this algorithm

When the algorithm uses various numbers of predictions and takes the average, the accuracy improves, posing an **advantage**. Additionally, it's more **stable** because any change in the data set could affect only a particular or a group of trees(segments) associated to that point and not the other segments(hence other predictions wont affect).

²¹ Simplilearn. "What Is Ensemble Learning? Understanding Machine Learning Techniques." *Simplilearn.com*, 26 Mar. 2021, www.simplilearn.com/ensemble-learning-article.

Experiment Methodology

Dependent and Independent variables

Gold Price (Independent variable)	The input variable for the model to predict. Which is a market value of gold mineral. Here it is INR(indiancurrency) of 1 gram 24 karat gold. From the year 1925 to 2023
Size of data (Independent variable)	Is the number of data which the model gets trained. This is correlated with computer resources hence if the size increases, computer resources are used more leading to high time complexity
Hyper Parameters (Independent variable)	<p><u>SVR's Hyperparameter :</u></p> <ul style="list-style-type: none"> • Kernel Typ²²e: Specifies the type of kernel used in the algorithm in this case it is 'rbf²³(radial basis function). • Regularization Parameter (C)²⁴: Controls the compromises between achieving a low training and testing error. And maximizes the epsilon tube width

²²"Kernel Method." *Engati*, 6 2023, www.engati.com/glossary/kernel-method.

²³ Sreenivasa, Sushanth. "Radial Basis Function (RBF) Kernel: The Go-To Kernel." *Medium*, 12 Oct. 2020, towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a#.

²⁴ "What is the Influence of C in SVMs with Linear Kernel?" *Cross Validated*, stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel#.

	<p><u>RFR's Hyperparameter :</u></p> <ul style="list-style-type: none"> • Number of Trees (n_estimators)²⁵: Specifies the amount of decision trees in the random forest. Higher number of trees(Splits) generally leads to better performance, but it also increases computational complexity. • Minimum Samples Split (min_samples_split): Specifies the minimum number of samples data points required to split an segment(a internal node).
Predicted Currency Value (Dependent variable)	Represents the relative worth of a specific currency compared to another, in this case INR is compared to USD.
Perfomance metrics (Dependent variable)	<ul style="list-style-type: none"> • Mean-Squared-Error : Is the measure of the average squared difference between the predicted values and the actual values in a regression model. A lower MSE is considered as a better fit. • R-Squared : Is a statistical measure that represents the proportion of the variance in

²⁵ Bradley Boehmke & Brandon Greenwell. "Chapter 11 Random Forests | Hands-On Machine Learning with R." *Site Not Found · GitHub Pages*, 1 Feb. 2020, bradleyboehmke.github.io/HOML/random-forest.html.

	<p>the dependent variable that is derived by the independent variables in a regression model.</p> <p>It ranges from 0 to 1, with higher positive values close to 1 will indicate a better fit of the model to the data.</p> <ul style="list-style-type: none"> ● Execution Time : Execution time measures the amount of time it takes for a machine learning algorithm to process and generate predictions
--	--

Control variables

Hardware/Computational Resources	<p>Laptop - Dell 7570/S</p> <p>Processor - 8th gen core i5 - 8250</p> <p>Ram - 8GB</p> <p>Hard disc - 128GB SSD + 1TB</p> <p>Operation system - WIN 10</p>
Data Strucute and Size	<p>Is the type of representation of data, here it is array , which holds datas till 3 decimal places and numbers in thousands.</p>

Procedure

I will be using google colab as my development environment where I will use python and its library for various uses such as *Scikit* for data splitting, metrics calculation, *Numpy*(to work with arrays such as visiting each element in the array), *Matplotlib*(inside this library we use pyplot module) will allow to generate graphs and *Pandas* to import and create the dependent and dependant variable vector. And google sheets to export data sets in CSV format²⁶. My sources such as research papers/books/coding pages help me gain the knowledge to answer my RQ(ex: how the algorithms work) .The methodology and procedure is relevant to the RQ because it gives the overview of the investigation such as the types of variables/data.

Data Importing and Pre-Processing

1. Importing Data

The pandas ²⁷library read the data set and sends the data as output in data frame(which is the rows and columns the data is organised is called as data frame). So that the variable's output is stored in data frame and not in file format

2. Splitting independent(also known as features²⁸) and dependent variables into 2 entities using pandas library

- a. Matrix of features²⁹ - containing the possible input(independent) values
- b. Dependant variable Vector - containing only the possible ouput/prediction(dependent) values

²⁶ "What is a .CSV File and What Does It Mean for My Ecommerce Business?" *BigCommerce*, www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/.

²⁷ "Pandas Introduction." *W3Schools Online Web Tutorials*, www.w3schools.com/python/pandas/pandas_intro.asp#.

²⁸"Independent variable --> Features." *Attention Required! | Cloudflare*, statisticsbyjim.com/regression/independent-dependent-variables/.

²⁹ "How to Learn Machine Learning? – The Matrix of Features and The Target Variable." *Develando Blog*, 7 Dec. 2018,blog.develando.com/learn-machine-learning-matrix-features-dependent-variables-factor/.

3. Missing data

This process is done so that there is no error while the model is getting trained, 2 methods:

- i. Ignoring variables
- ii. Replacing missing variables with the average value of the column

4. Splitting Training and Test Set

$\frac{1}{3}$ of the data of dependent and independent variables are taken randomly using scikit's train_tes_split tool for test set, rest of the datas are kept for training set. This process gives 4 different data sets(test set's dependent and independent data, train set's dependent and independent data respectively)

5. Feature Scaling³⁰

The data set might have extreme data which may cause some complexity to the algorithm, feature scaling is standardising all the data into a given range(0 to 1 in this program), as the model might get confuse and takes more unwanted computational resources. Feature scaling is done after training and test set split, this is because information leakage might happen, leading to inaccurate predictions.

³⁰ Vashisht, Raghav. "When to Perform a Feature Scaling?" *Atoti Community*, 14 Feb. 2023, www.atoti.io/articles/when-to-perform-a-feature-scaling/#.

Building and Training the model

SVR - is build using scikit library and set the kernel type as ‘RBF’

RFR - is build and the limit of decision trees is set to 10

Then both the algorithm is trained with the test set

Testing and Visualisation

The predicted results and the actual results are then inversely transformed from the feature scaling and displayed with the help of test set datas. The datas will be splitted into 4 different size of (25,50,75 and 99) each size will be put as an input for both the models where $\frac{1}{3}$ of the data is set as testset and rest for training the model, where the timings and the 3 metric’s values are being tested for each data size.

Data

Currency	Gold	3	3.743	4.78	10.256	8.1	50.60
Value	Price(24krat)	3.32	4.405	4.75	11.187	8.38	54.00
2.76	1.875	3.32	5.105	4.75	11.935	8.96	43.20
2.75	1.843	3.32	5.293	4.77	11.975	8.74	48.60
2.75	1.837	3.32	6.2	4.77	9.7	8.19	68.50
2.74	1.837	3.32	8.387	4.76	6.325	8.13	93.70
2.76	1.843	3.32	8.862	4.77	7.175	7.86	133.00
2.77	1.805	3.32	9.587	4.78	7.175	8.66	167.00
2.97	1.818	3.32	9.417	4.78	7.175	9.46	164.50
3.8	2.306	3.31	9.918	4.76	7.18	10.1	180.00
3.14	2.405	3.61	9.805	6.36	8.38	11.36	197.00
2.64	2.881	4.79	7.681	7.5	10.25	12.37	213.00
2.71	3.018	4.79	7.306	7.5	16.20	12.61	214.00
2.67	2.993	4.78	7.775	7.5	17.60	12.96	257.00
2.68	3.174	4.75	7.918	7.5	18.40	13.92	313.00
2.73	3.604	4.76	9.081	7.49	19.30	16.23	314.00
3	3.743	4.79	9.062	7.59	20.20	17.5	320.00
3.32	4.405	4.78	9.538	7.74	27.85	22.74	346.60

25.92	433.40
30.49	414.00
31.37	459.80
32.43	468.00
35.43	516.00
36.31	472.50
41.26	404.50
43.06	423.40
44.94	440.00
47.19	430.00
48.61	499.00
46.58	560.00
45.32	585.00
44.1	700.00
45.31	1080.00
41.35	1250.00
43.51	1450.00
48.41	1850.00

45.73	2640.00
46.67	3105.00
53.44	2960.00
56.57	2800.65
62.33	2634.35
62.97	2862.35
66.46	2966.75
67.79	3143.80
70.09	3522.00
70.39	4865.10
76.38	4872.00
74.57	5267.00
81.35	5838.50

Metrics

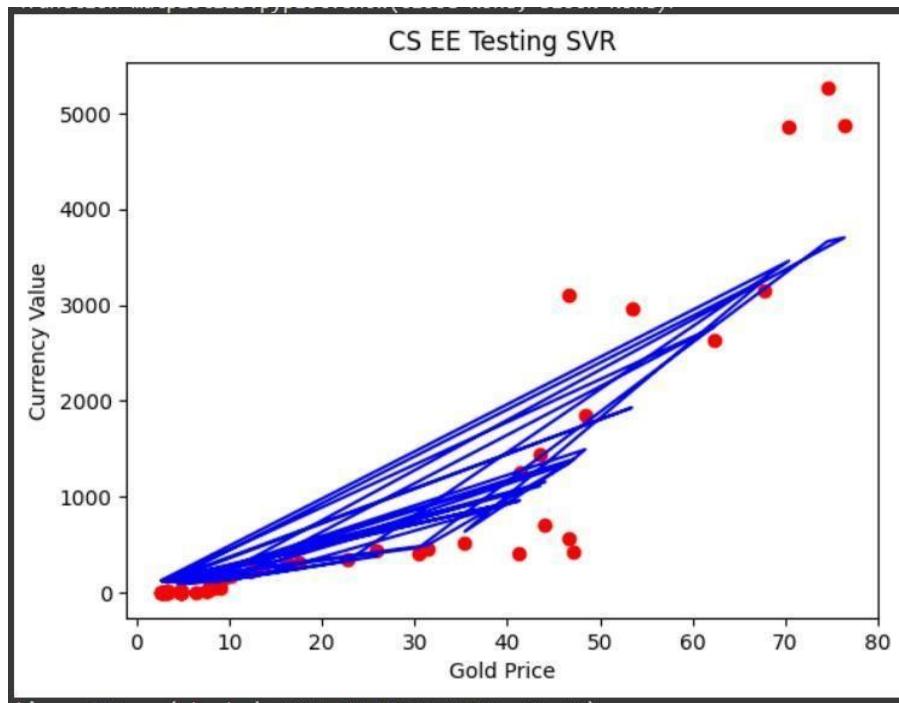
I will be researching my RQ using 4 metrics :

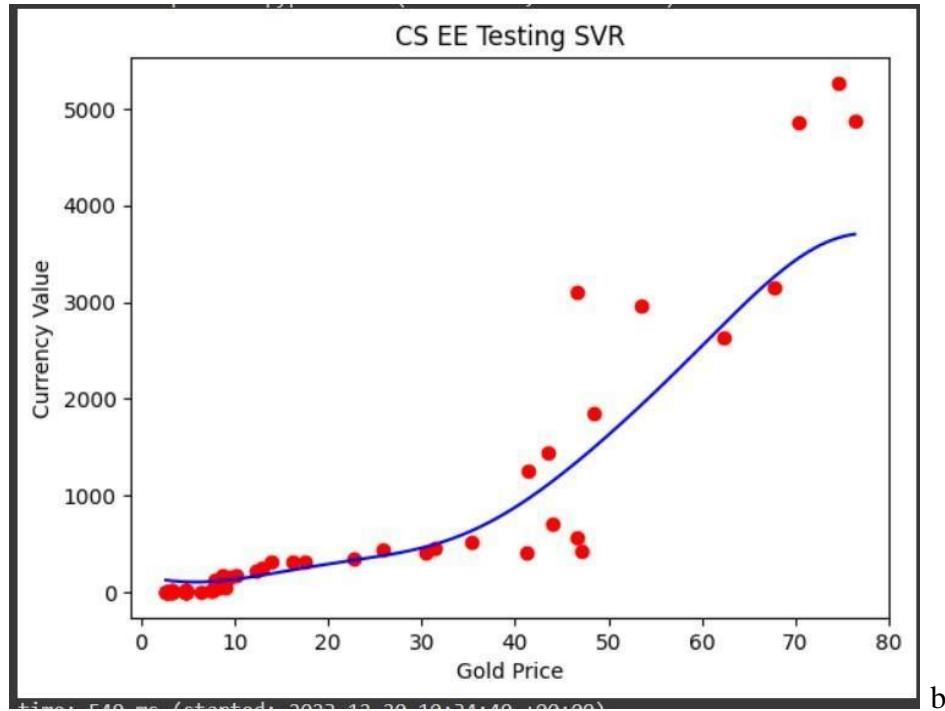
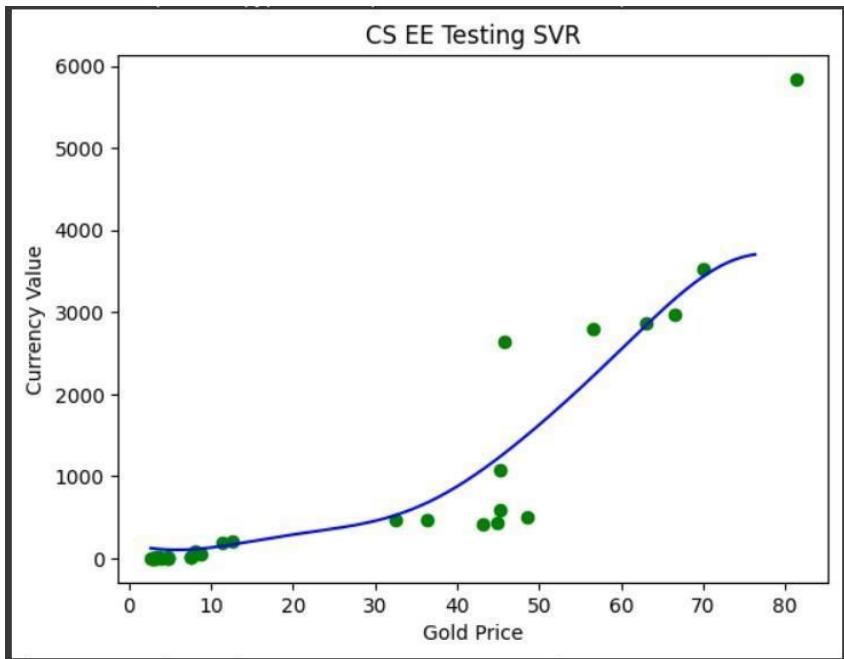
Rsquared (Predictive Accuracy)	<p>Is the metric which measures the variance between dependent and independent variable. Ranges between 0 and 1 where 1 is perfect fit and 0 is no relationship predicted between variables/ independent, and no predicted value is in the mean of actual value. If it is negative, it indicates that the regression model is a worse fit for the data than a simple horizontal line representing the mean of the dependent variable, possible reasons could be the model choice, data quality, or underlying assumptions of the analysis.</p>
Mean-Squared (Predictive Accuracy)	<p>The average squared difference between the predicted and the actual value which calculates the squared differences for each data point and averages them. Lower MSE values indicate better model performance, with zero representing a perfect fit , this is sensitive to outliers.</p>
Mean-Absolute-Error (Predictive Accuracy)	<p>The average absolute difference between predicted and actual value. calculates the absolute differences</p>

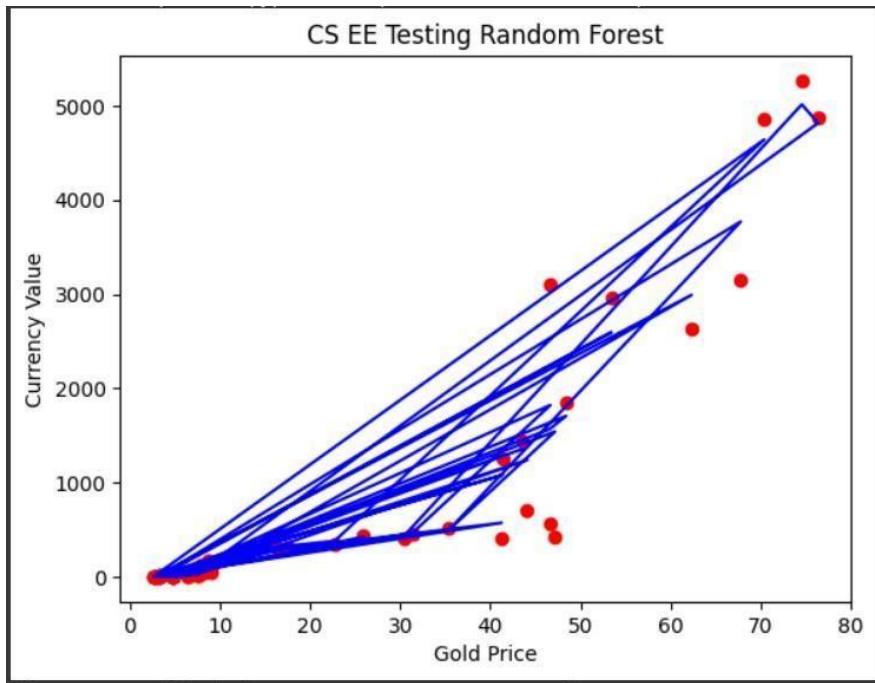
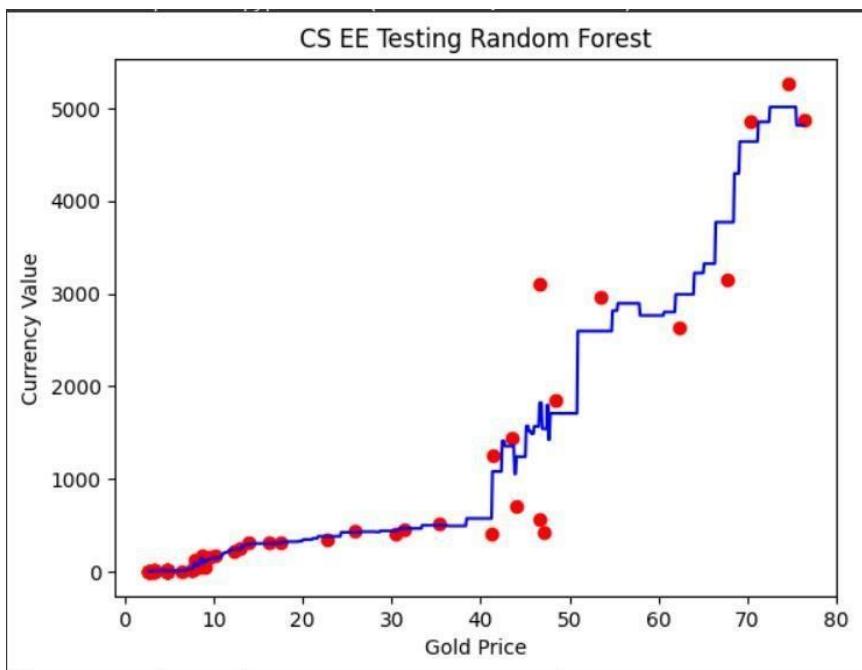
	for each data point, averages them Unlike Mean Squared Error (MSE), MAE is less sensitive to outliers, Lower MAE values indicate better model performance, with zero representing a perfect fit.
Time Complexity	Used to measure both training time and test time execution for different sizes of data in order to measure the computational complexity

Output (Graphs)

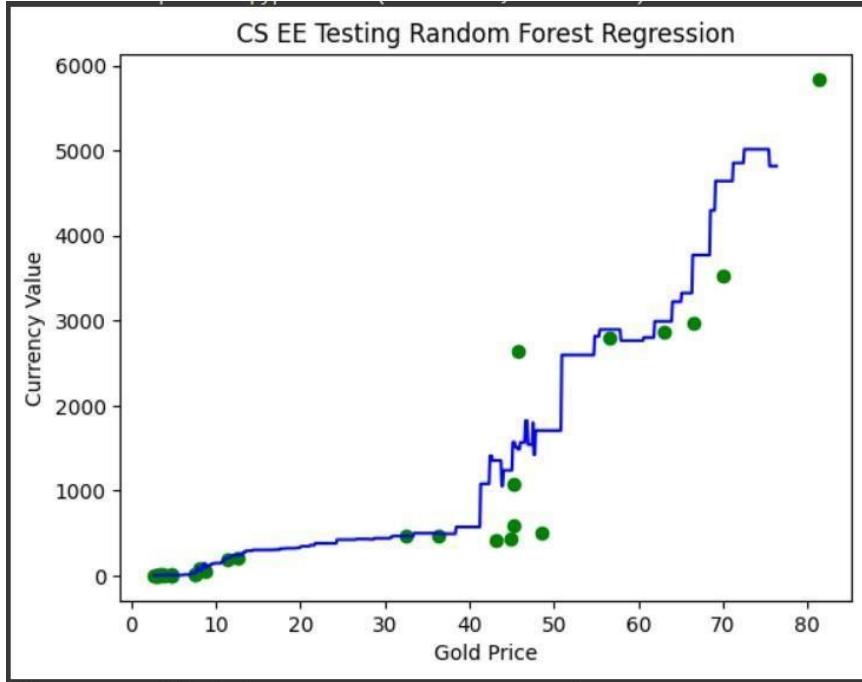
Training-set (SVR)



Training-set-smooth(SVR)Testing-set

Training-set (RFR)Training-set-smooth(RFR)

Testing-set



Blue line - Regression model, Red points - Training datas, Green points - Test data

According to the graphs, which is the graph for 99 data points where the RFR model is getting trained much better because this model is able to be lenient and achieve minimal distance between the data points compared to SVR. This is due to decision trees (the information entropy) the number of splits and spot of split takes place in the right spot and quantity leading to more leaves, which allows the model to train with the average of each leaf causing it to be more accurate. This helps in achieving minimal distance between green point and the regression line for the RFR.

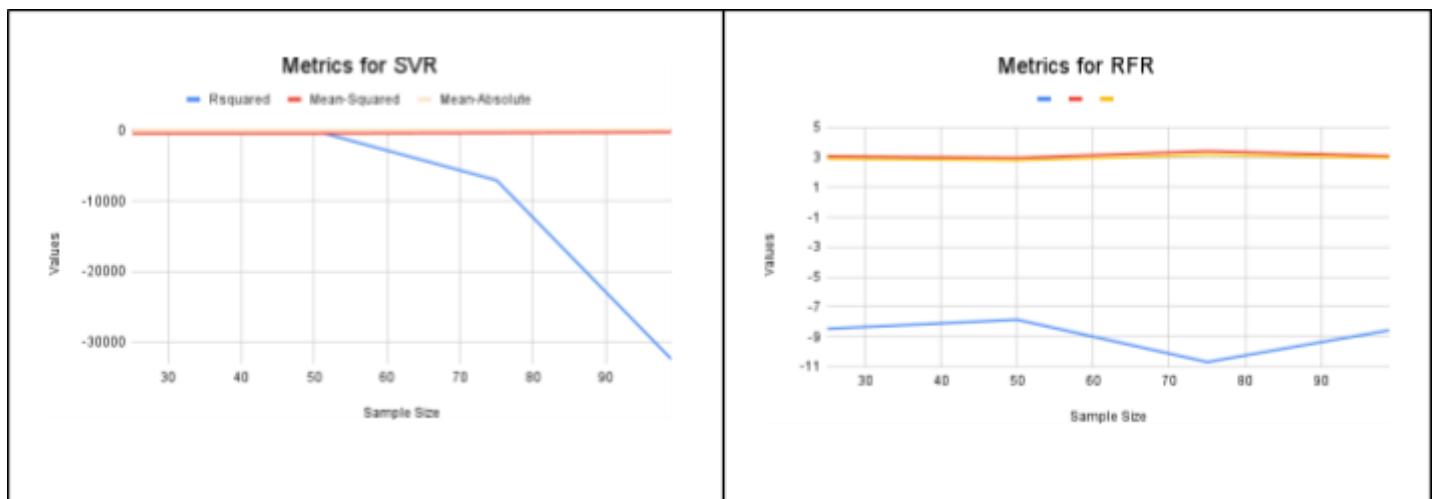
Whereas, the SVR is **not** very lenient with the fluctuations causing it to be more restrictive to predict accurate data. One possible reason could be due to the loss function(margin-of-error), it can ignore the data points within the epsilon tube as mentioned in the background information.

Comparing with metrics and Evaluation

Predictive Accuracy

Sample Size	R squared	Mean-Squared	Mean-Absolute
25(SVR)	-12.27	3.64	3.5
50(SVR)	-58.62	7.72	7.19
75(SVR)	-7062.47	84.04	80.28
99(SVR)	-32404.94	180.01	178.35
25(RFR)	-8.46	3.08	2.91
50(RFR)	-7.85	2.97	2.8
75(RFR)	-10.68	3.42	3.18
99(RFR)	-8.55	3.09	2.97

Table 1



The desirable value of a R^2 test is around 1 and positive. If it is a 0, then it explains that there is no variance in the target variable. If this value is negative, then it means it does not follow the trend of the data and model is worse. In the above table 1, SVR's R^2 value is -32404.94 which explains that there is very little/no correlation between variables in the context of SVR, The same can be seen in the plotted graph of Metrics for SVR and RFR above. And the other possible reason for a negative R^2 value might be the type of model because the model might not suit the data's trend which can be an error

Whereas, Random-Forest-Regression is still in negative but much closer to 0 than SVR which states that Random-Forest-Regression can figure a trend between the variables. The reason behind this, according to this research and as mentioned in the background knowledge, is that **even though** SVR trains in the extreme data-points, it cannot compete against RFR's Decision tree as it uses *Information entropy* for the right-splits and right-spots and takes the average of each leafs leading to accuracy one more reason for accuracy is stabilities. However, in RFR, change in one leaf might not affect another causing it to be more stable.

On the other hand, the Mean-Squared-Error value is desired ³¹to be closer to 0 symbolizing average squared difference between predicted and actual value. This metric is sensitive to outliers; any model that predicts more outliers contradicts this metric. According to the previous metric (R^2), SVR produces more outliers as there is a huge variance between the variables. Therefore, adding to the mean squared error with the value 180.02 states that SVR predicted

³¹ M, Padhma. "A Comprehensive Introduction to Evaluating Regression Models." *Analytics Vidhya*, 30 Nov. 2023, www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/#.

values are widely dispersed around its mean hence less accurate. Whereas, Random forest regression doesn't produce more outliers than the SVR since its R^2 value is in negative as well, however it is much closer to 0 than SVR, highlighting less variance from mean (this is in terms of predicted values). The effect of this makes the mean-squared matrix to be much closer to 0(3.09) causing RFR to be more accurate.

Similarly, even if we reject the outlier aspect, SVR produces higher mean-absolute³²-error compared to Random forest regression. Where in mean-absolute-error the desired value should be lower.

One major concept to note is that SVR is getting affected by the amount of data it needs to be trained and tested. As we see in the tabular column, with the data size increasing, the error metrics's values are also increasing, stating that it cannot find pattern in the variables hence leading to mispredictions making the predictions as the outliers compared to the actual predictions.

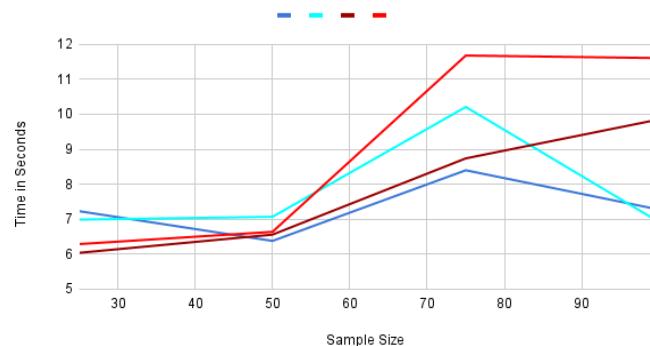
³² "What is a Good MAE Score? (simply Explained)." *Stephen Allwright*, 28 Aug. 2022, stephenallwright.com/good-mae-score/.

Time complexity

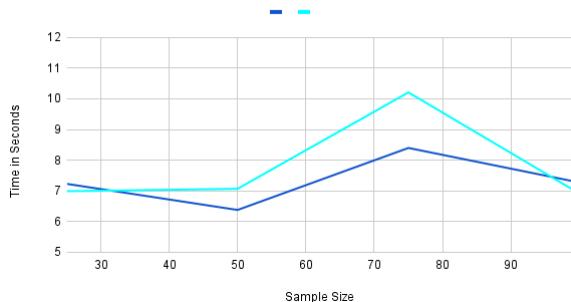
Sample Size	Training			Average	Predicting			Average
	Trial 1	Trial 2	Trial 3		Trial 1	Trial 2	Trial 3	
25(SVR)	7.49	7.24	6.95	7.23	7.41	6.32	7.23	6.99
50(SVR)	6.95	6.14	6.04	6.38	7.13	7.22	6.86	7.07
75(SVR)	5.94	7.75	11.5	8.4	8.53	6.7	15.4	10.21
99(SVR)	7.55	7.06	7.34	7.32	7.52	6.12	7.52	7.05
25(RFR)	6.21	5.96	5.95	6.04	6.33	6.23	6.32	6.29
50(RFR)	5.96	5.96	7.76	6.56	6.71	6.31	6.92	6.64
75(RFR)	5.85	7.87	12.5	8.74	7.04	13.8	14.2	11.68
99(RFR)	7.67	8.66	13.1	9.81	12.4	6.23	16.2	11.61

Table 2

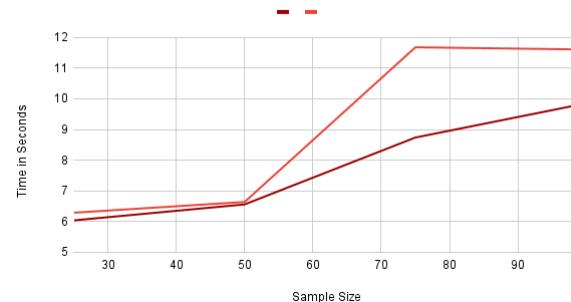
Time Complexity Graph



SVR Training & Test time complexity



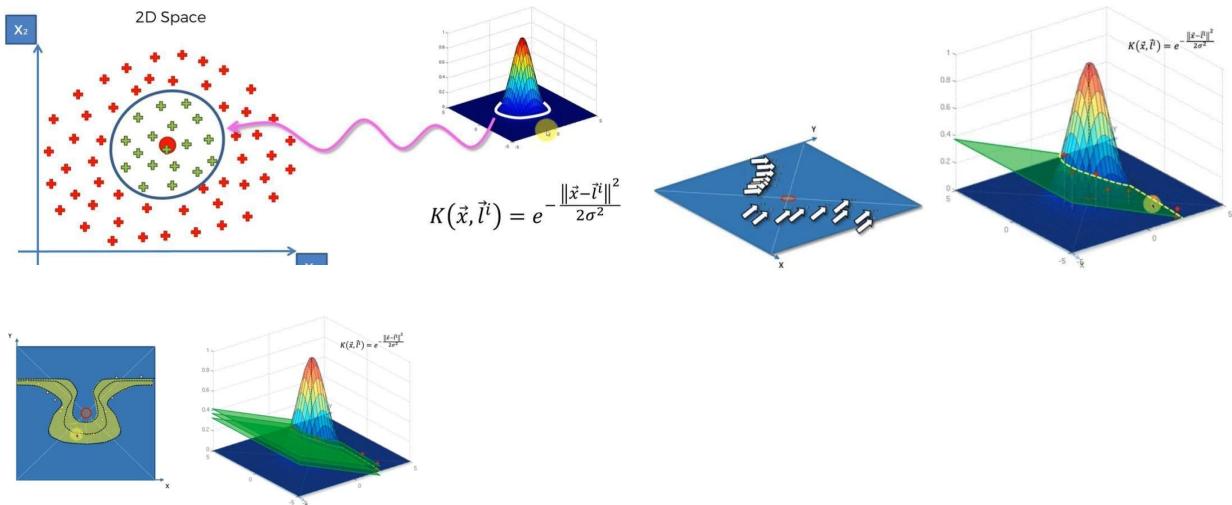
RFR Training and Test time complexity



... → Training result of RFR	... → Training result of SVR
... → Testing result of RFR	... → Testing result of SVR

Both the models take relatively high time for prediction compared to training. According to the above results(table 2 and time complexity graph of SVR and RFR), random forest regression takes higher time for both training and predicting. This is because of the computational complexity, as we saw how the Radial-basis-function kernel works in background information(RBF), we know that RBF reduces the execution time using this formula and without the 3rd dimension.

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$



Whereas Random-Forest-Regression doesn't have a function like this and as we know, Random-Forest-Regression has to perform decision tree function with more splits to ensure accuracy. However, that process may be more time consuming when compared to SVR and as the size of the data increases the number of splits also increases causing high time complexity, which can be a disadvantage for Random-Forest-Regression. All these analysis is found for the data set of size 99 data, where there are chances of the values to change as the data count increase, hence the sources of this investigation is trust only 80%.

Conclusion & Further Scope of Research

In conclusion, Answering the question **To what extent can the efficiency of Support Vector Regression and Random Forest Regression be compared for predictive accuracy and computational complexity when predicting currency value based on gold price?** That both the models have advantages and disadvantages that evaluate their efficiency. RFR has greater predictive accuracy, whereas SVR is faster thereby having less computational complexity which is mainly due to their key functions such as Information entropy and decision tree splits for RFR leading to lesser outliers in terms of predicted results and radial-basis-function for SVR leading to quicker execution time.

In both cases, the size of the data plays an important role. In SVR, as the size increases the error increases due lack of efficient training of data sets compared to RFR, whereas in RFR as the size of the data increases the execution time increases due to more splits for a greater size of data.

As we see in the above analysis R^2 metrics is negative for both the models , where it is not fitting the model well, the possible reason behind this could be the fact that currency value is compressed by many different variables ³³such as gold, agricultural goods, manufacturing/industrial goods and all the goods and services in a country. Hence, it may be hard to predict it with only gold prices, the future scope of this research could be testing both the models more realistic by adding more goods and services's prices which may turn R^2 value to positive as more variables may help algorithms train and predict better causing less variance between datas. And it can change the parameters such as "C" for SVR in order to analyze the accuracy when the constant changes.

³³CFI Team. "How is Currency Valued." *Corporate Finance Institute*, 6 Dec. 2022, corporatefinanceinstitute.com/resources/economics/how-is-currency-valued/.

Bibliography

"1 USD to INR in 1947 Till Now, Historical Exchange Rates Explained." *Blog-Best Foreign Exchange*, 10 Apr. 2023, www.bookmyforex.com/blog/1-usd-inr-1947-till-now. Accessed 4 Nov. 2023.

"6.3. Preprocessing Data." *Scikit-learn*, scikit-learn.org/stable/modules/preprocessing.html. Accessed 7 Sept. 2023.

Agrawal, Anushka. "Simple Linear Regression Modeling-Part 1." *Medium*, 8 May 2021, medium.com/nerd-for-tech/simple-linear-regression-modeling-part-1-1ae3b59c6ab5. Accessed 29 Sept. 2023.

Bradley Boehmke & Brandon Greenwell. "Chapter 11 Random Forests | Hands-On Machine Learning with R." *Site Not Found · GitHub Pages*, 1 Feb. 2020, bradleyboehmke.github.io/HOML/random-forest.html. Accessed 23 July 2023.

"CART (Classification And Regression Tree) in Machine Learning." *GeeksforGeeks*, 4 Dec. 2023, www.geeksforgeeks.org/cart-classification-and-regression-tree-in-machine-learning/.

CFI Team. "How is Currency Valued." *Corporate Finance Institute*, 6 Dec. 2022, corporatefinanceinstitute.com/resources/economics/how-is-currency-valued/.

"DECISION TREES : CLASSIFICATION AND REGRESSION TREES (CART)." *National Institute of Science Education and Research*, www.niser.ac.in/~smishra/teach/cs460/23cs460/lectures/lec9.pdf.

Deepankar. "Decision Trees with CART Algorithm." *Medium*, medium.com/geekculture/decision-trees-with-cart-algorithm-7e179acee8ff. Accessed 22 Apr. 2023.

Education Ecosystem (LEDU). "Understanding K-means Clustering in Machine Learning." *Medium*, 12 Sept. 2018, towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1#:~:text=A%20cluster%20refers%20to%20a,the%20center%20of%20the%20cluster. Accessed 11 Oct. 2023.

Gandhi, Rohith. "Support Vector Machine — Introduction to Machine Learning Algorithms." *Medium*, 5 July 2018, towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. Accessed 25 Aug. 2023.

"Historical Gold Rate/Trend in India - Complete Information." *Compare & Apply for Credit Cards & Loan Online in India*, www.bankbazaar.com/gold-rate/gold-rate-trend-in-india.html. Accessed 5 July 2023.

"How Gold Affects Currencies." *Investopedia*, 16 May 2011, www.investopedia.com/articles/forex/11/golds-effect-currencies.asp. Accessed 7 Feb. 2023.

"How to Learn Machine Learning? – The Matrix of Features and The Target Variable." *Develando Blog*, 7 Dec. 2018, blog.develando.com/learn-machine-learning-matrix-features-dependent-variables-factor/. Accessed 31 Jan. 2023.

"Independent variable --> Features." *Attention Required! / Cloudflare*, statisticsbyjim.com/regression/independent-dependent-variables/.

"Just a Moment..." *Just a Moment..*, www.baeldung.com/cs/ml-svm-c-parameter#. Accessed 10 Feb. 2023.

"Kernel Method." *Engati*, 6 2023, www.engati.com/glossary/kernel-method. Accessed 10 Apr. 2023.

"Kernel Trick: A Non-linear Decision Boundary for Support Vector Machine Learning. | Taylor Ortiz Posted on the Topic | LinkedIn." *LinkedIn*, 20 Nov. 2023, www.linkedin.com/posts/anthonytortiz_using-the-kernel-trick-to-create-a-non-linear-activity-7132365409428656129-Ikkp/?trk=public_profile_share_view. Accessed 29 Nov. 2023.

"Lesson 10: Support Vector Machines | STAT 508." *PennState: Statistics Online Courses*, online.stat.psu.edu/stat508/lesson/10. Accessed 25 Mar. 2023.

M, Padhma. "A Comprehensive Introduction to Evaluating Regression Models." *Analytics Vidhya*, 30 Nov. 2023, www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/#. Accessed 14 Oct. 2023.

"Measures of Worth, Inflation Rates, Saving Calculator, Relative Value, Worth of a Dollar, Worth of a Pound, Purchasing Power, Gold Prices, GDP, History of Wages, Average Wage." *Measuring Worth - Relative Worth Comparators and Data Sets*, www.measuringworth.com/datasets/exchangeproxy/result.php?year_source=1925&year_result=2023&countryE%5B%5D=India. Accessed 13 Nov. 2023.

"Medium." *Medium*,

medium.com/Analytics-vidhya/support-vector-regression-for-machine-learning-843978ba279. Accessed 20 Mar. 2023.

"Numeracy, Maths and Statistics - Academic Skills Kit." *The Things We Do Here Make a Difference out There*, Newcastle University,

www.ncl.ac.uk/webtemplate/ask-assets/external/math-resources/statistics/regression-and-correlation/simple-linear-regression.html. Accessed 2 Sept. 2023.

"Pandas Introduction." *W3Schools Online Web Tutorials*,

www.w3schools.com/python/pandas/pandas_intro.asp#. Accessed 26 Nov. 2023.

"Quadratic Programming." *Cornell University Computational Optimization Open Textbook - Optimization Wiki*, optimization.cbe.cornell.edu/index.php?title=Quadratic_programming. Accessed 25 Feb. 2023.

"Random Forest Versus Logistic Regression: a Large-scale Benchmark Experiment." *BioMed Central*, 17 July 2018, bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5. Accessed 13 Apr. 2023.

"Separating Hyperplanes in SVM." *GeeksforGeeks*, 15 Sept. 2021,

www.geeksforgeeks.org/separating-hyperplanes-in-svm/. Accessed 3 Oct. 2023.

SHAH, AJIT. "Gold Price Chart - Gold Price History for 95 Years." *TaxGuru*, 8 Feb. 2023,

taxguru.in/income-tax/gold-price-chart-gold-price-history.html#google_vignette. Accessed 11 Oct. 2023.

"Simple Linear Regression." www.jmp.com/en_in/statistics-knowledge-portal/what-is-regression.html.

Accessed 3 May 2023.

Simplilearn. "What Is Ensemble Learning? Understanding Machine Learning Techniques."

Simplilearn.com, 26 Mar. 2021, www.simplilearn.com/ensemble-learning-article. Accessed 7 Mar. 2023.

---. "What is Simple Linear Regression in Machine Learning?" *Simplilearn.com*, 22 Aug. 2022,

www.simplilearn.com/what-is-simple-linear-regression-in-machine-learning-article#. Accessed 1 Oct. 2023.

Singh, Navjot. "Support Vector Regression for Machine Learning." *Medium*, 18 June 2020,

medium.com/Analytics-vidhya/support-vector-regression-for-machine-learning-843978ba6279. Accessed 15 Aug. 2023.

"Sklearn.metrics.r2_score." *Scikit-learn*, scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html. Accessed 7 Sept. 2023.

Sreenivasa, Sushanth. "Radial Basis Function (RBF) Kernel: The Go-To Kernel." *Medium*, 12 Oct. 2020, towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a. Accessed 27 Oct. 2023.

"Support Vector Machine (SVM) Algorithm - Javatpoint." *Www.javatpoint.com*, www.javatpoint.com/machine-learning-support-vector-machine-algorithm. Accessed 27 Aug. 2023.

"Support Vector Regression (SVR) Using Linear and Non-Linear Kernels in Scikit Learn." *GeeksforGeeks*, 30 Jan. 2023, www.geeksforgeeks.org/support-vector-regression-svr-using-linear-and-non-linear-kernels-in-scikit-learn/. Accessed 26 Aug. 2023.

"Support Vector Regression." *SpringerLink*, link.springer.com/chapter/10.1007/978-1-4302-5990-9_4#. Accessed 15 Mar. 2023.

Vashisht, Raghav. "When to Perform a Feature Scaling?" *Atoti Community*, 14 Feb. 2023, www.atoti.io/articles/when-to-perform-a-feature-scaling/#. Accessed 19 May 2023.

"What is a .CSV File and What Does It Mean for My Ecommerce Business?" *BigCommerce*, www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/.

"What is a Good MAE Score? (simply Explained)." *Stephen Allwright*, 28 Aug. 2022, stephenallwright.com/good-mae-score/. Accessed 20 Aug. 2023.

"What is a Vector -- Computing for All." *Computing for All*, 30 July 2023, wwwcomputing4all.com/courses/introductory-data-science/lessons/what-is-a-vector/#. Accessed 15 Aug. 2023.

"What is the Influence of C in SVMs with Linear Kernel?" *Cross Validated*, stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel#. Accessed 17 June 2023.

Yadav, Suraj. "What is Kernel Trick in SVM ? Interview Questions Related to Kernel Trick." *Medium*, 29 Apr. 2023, medium.com/@Suraj_Yadav/. Accessed 9 Aug. 2023.

Appendix

Screenshots of the code in google colab (Python) and the result of matrixs

Support Vector Regression (SVR)

Libraries

```
[ ] import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

time: 751 µs (started: 2023-12-20 15:45:47 +00:00)
```

Dataset

```
[ ] dataset = pd.read_csv('Dataset.csv')
x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1:].values

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 1/3, random_state = 0)

print(x_train)
print(y_train)
print(x_test)
print(y_test)
```

Feature Scaling

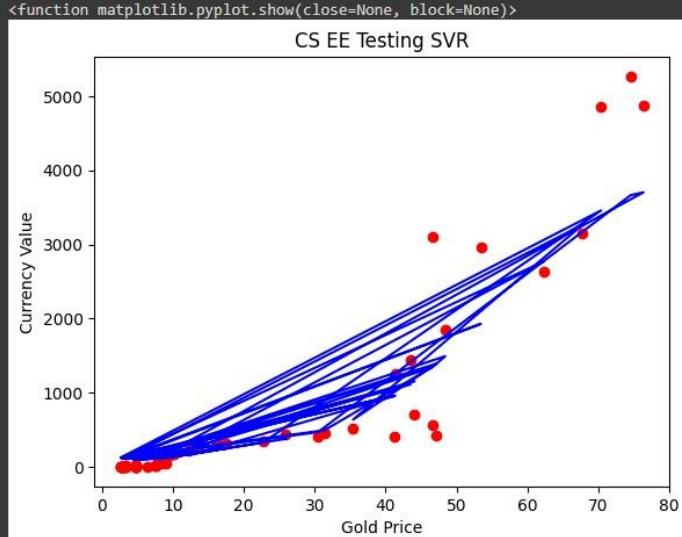
```
[ ] from sklearn.preprocessing import StandardScaler  
sc_xtrain = StandardScaler()  
sc_ytrain = StandardScaler()  
sc_xtest = StandardScaler()  
sc_ytest = StandardScaler()  
x_train = sc_xtrain.fit_transform(x_train)  
y_train = sc_ytrain.fit_transform(y_train)  
x_test = sc_xtest.fit_transform(x_test)  
y_test = sc_ytest.fit_transform(y_test)  
print(x_train)  
print(y_train)  
print(x_test)  
print(y_test)  
[-0.12155543]
```

Training the SVR model with dataset

```
[ ] !pip install ipython-autotime  
%load_ext autotime  
from sklearn.svm import SVR  
regressor = SVR(kernel = 'rbf')  
regressor.fit(x_train, y_train)
```

Visualising training set

```
[ ] plt.scatter(sc_xtrain.inverse_transform(x_train),sc_ytrain.inverse_transform(y_train),color = 'red')
plt.plot(sc_xtrain.inverse_transform(x_train), sc_ytrain.inverse_transform(regressor.predict(x_train).reshape(-1,1)), color = 'blue')
plt.title('CS EE Testing SVR')
plt.xlabel('Gold Price')
plt.ylabel('Currency Value')
plt.show
```

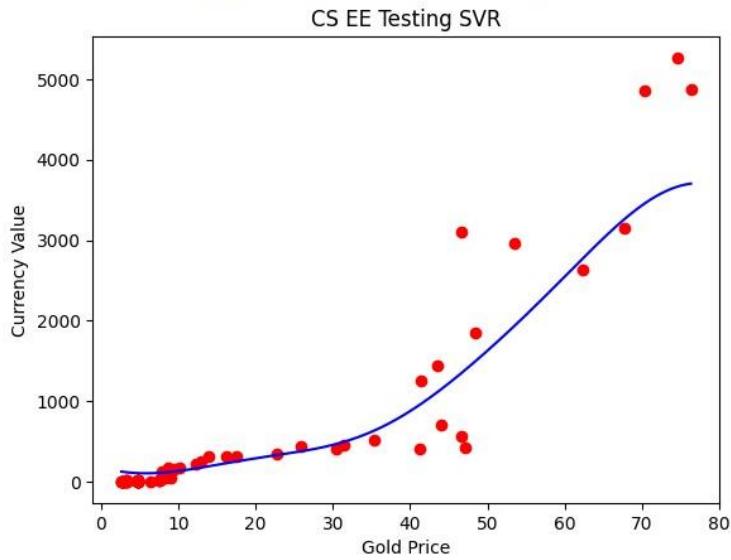


time: 296 ms (started: 2023-12-20 15:45:59 +00:00)

Higher resolution and smooth curve training set

```
[ ] x_grid = np.arange(min(sc_xtrain.inverse_transform(x_train)), max(sc_xtrain.inverse_transform(x_train)),0.1)
x_grid = x_grid.reshape((len(x_grid),1))
plt.scatter(sc_xtrain.inverse_transform(x_train),sc_ytrain.inverse_transform(y_train),color = 'red')
plt.plot(x_grid,sc_ytrain.inverse_transform(regressor.predict(sc_xtrain.transform(x_grid)).reshape(-1,1)),color = 'blue')
plt.title('CS EE Testing SVR')
plt.xlabel('Gold Price')
plt.ylabel('Currency Value')
plt.show
```

<function matplotlib.pyplot.show(close=None, block=None)>



time: 292 ms (started: 2023-12-20 15:45:59 +00:00)

Visualizing test set

```
[ ] !pip install ipython-autotime
%load_ext autotime
x_grid = np.arange(min(sc_xtrain.inverse_transform(x_train)), max(sc_xtrain.inverse_transform(x_train)),0.1)
x_grid = x_grid.reshape((len(x_grid),1))
plt.scatter(sc_xtest.inverse_transform(x_test),sc_ytest.inverse_transform(y_test),color = 'green')
plt.plot(x_grid,sc_ytrain.inverse_transform(regressor.predict(sc_xtrain.transform(x_grid)).reshape(-1,1)),color = 'blue')
plt.title('CS EE Testing SVR')
plt.xlabel('Gold Price')
plt.ylabel('Currency Value')
plt.show
```

Requirement already satisfied: ipython-autotime in /usr/local/lib/python3.10/dist-packages (0.3.2)

Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-packages (from ipython-autotime) (7.24.0)

Evaluating SVR Performance

```
[ ] from sklearn.metrics import r2_score
y_pred = sc_ytrain.inverse_transform(regressor.predict(sc_xtrain.transform(x_test)).reshape(-1, 1))
print("R-squared : "+str(r2_score(y_test, y_pred)))
from sklearn.metrics import mean_absolute_error
print("Mean Absolute : "+str(mean_absolute_error(y_test, y_pred)))
from sklearn.metrics import mean_squared_error
print("Mean Squared : "+str(mean_squared_error(y_test, y_pred, squared=False)))
```

R-squared : -32404.941257466246
Mean Absolute : 178.35025629007987
Mean Squared : 180.01650273646095
time: 5.37 ms (started: 2023-12-20 15:46:08 +00:00)

Random Forest Regression

Training the Random Forest Regression Model in the dataset

```
[ ] !pip install ipython-autotime
%load_ext autotime
from sklearn.ensemble import RandomForestRegressor
RFRegressor = RandomForestRegressor(n_estimators = 10, random_state = 0)
RFRegressor.fit(x_train, y_train)

Requirement already satisfied: ipython-autotime in /usr/local/lib/python3.10/dist-packages/ipython/autotime.py
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-packages/ipython/__main__.py
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages/setuptools/_distutils/__init__.py
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages/jedi/_compatibility.py
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages/decorator.py
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages/pickleshare.py
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-packages/traitlets/_version.py
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages/prompt_toolkit/_version.py
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages/pygments/_version.py
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages/backcall/_version.py
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages/matplotlib_inline/_version.py
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages/pexpect/_version.py
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages/parso/_version.py
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages/ptyprocess/_version.py
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages/wcwidth/_version.py
The autotime extension is already loaded. To reload it, use:
%reload_ext autotime
<ipython-input-124-d5cfb578b9ac>:5: DataConversionWarning: A column-vector
  RFRegressor.fit(x_train, y_train)

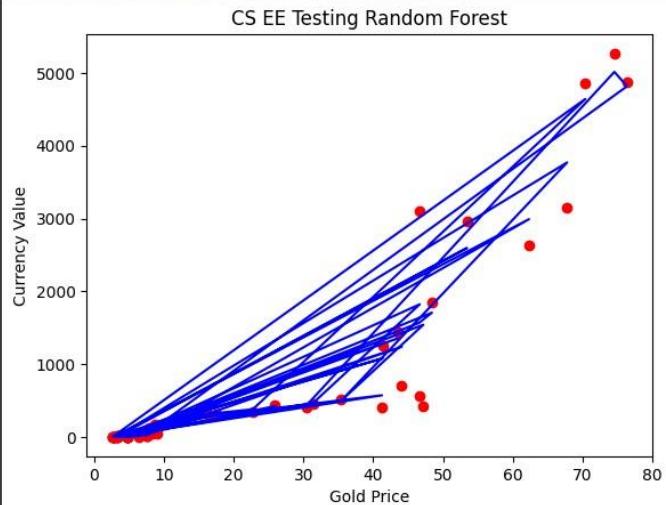
▼      RandomForestRegressor
RandomForestRegressor(n_estimators=10, random_state=0)

time: 14.8 s (started: 2023-12-20 15:46:08 +00:00)
```

Visualising Training set

```
[ ] plt.scatter(sc_xtrain.inverse_transform(x_train),sc_ytrain.inverse_transform(y_train),color = 'red')
plt.plot(sc_xtrain.inverse_transform(x_train), sc_ytrain.inverse_transform(RFregressor.predict(x_train).reshape(-1,1)), color = 'blue')
plt.title('CS EE Testing Random Forest')
plt.xlabel('Gold Price')
plt.ylabel('Currency Value')
plt.show
```

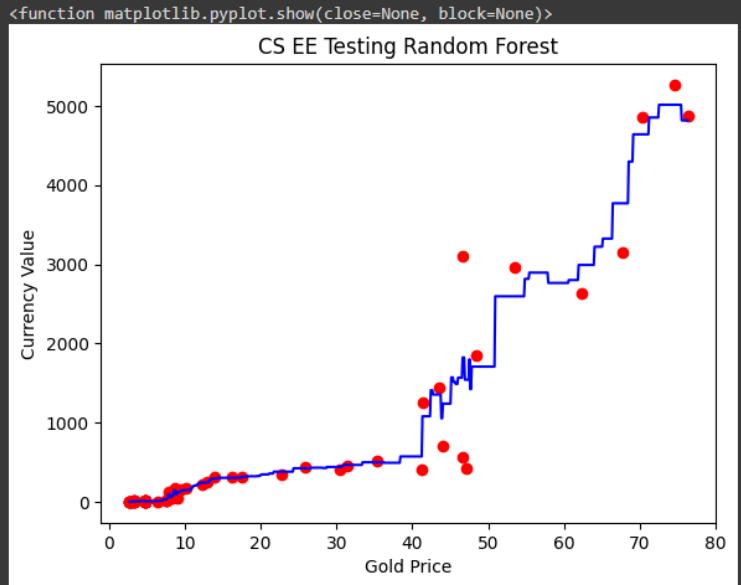
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
time: 798 ms (started: 2023-12-20 15:46:23 +00:00)
```

Higher resolution

```
[ ] x_grid = np.arange(min(sc_xtrain.inverse_transform(x_train)), max(sc_xtrain.inverse_transform(x_train)),0.1)
x_grid = x_grid.reshape((len(x_grid),1))
plt.scatter(sc_xtrain.inverse_transform(x_train),sc_ytrain.inverse_transform(y_train),color = 'red')
plt.plot(x_grid,sc_ytrain.inverse_transform(RFRegressor.predict(sc_xtrain.transform(x_grid)).reshape(-1,1)),color = 'blue')
plt.title('CS EE Testing Random Forest')
plt.xlabel('Gold Price')
plt.ylabel('Currency Value')
plt.show
```



time: 852 ms (started: 2023-12-20 15:46:24 +00:00)

Visualising Test Set

```
[ ] !pip install ipython-autotime
%load_ext autotime
x_grid = np.arange(min(sc_xtrain.inverse_transform(x_train)), max(sc_xtrain.inverse_transform(x_train)), 0.1)
x_grid = x_grid.reshape((len(x_grid),1))
plt.scatter(sc_xtest.inverse_transform(x_test),sc_ytest.inverse_transform(y_test),color = 'green')
plt.plot(x_grid,sc_ytrain.inverse_transform(RFRegressor.predict(sc_xtrain.transform(x_grid)).reshape(-1,1)),color = 'blue')
plt.title('CS EE Testing Random Forest Regression')
plt.xlabel('Gold Price')
plt.ylabel('Currency Value')
plt.show()

Requirement already satisfied: ipython-autotime in /usr/local/lib/python3.10/dist-packages (0.3.2)
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-packages (from ipython-autotime) (7.34.0)
```

Evaluating Random Forest Regression Performance

```
[ ] from sklearn.metrics import r2_score
y_pred2 = sc_ytrain.inverse_transform(RFRegressor.predict(sc_xtrain.transform(x_test)).reshape(-1, 1))
print("R-squared : "+str(r2_score(y_test, y_pred2)))
from sklearn.metrics import mean_absolute_error
print("Mean Absolute : "+str(mean_absolute_error(y_test, y_pred2)))
from sklearn.metrics import mean_squared_error
print("Mean Squared : "+str(mean_squared_error(y_test, y_pred2, squared=False)))

R-squared : -8.549775999998541
Mean Absolute : 2.9718585792013172
Mean Squared : 3.090271185510835
time: 4.72 ms (started: 2023-12-20 15:46:38 +00:00)
```

T- Test for Mean Squared *

```
[ ] from scipy import stats

# Assuming y_true_sv and y_pred_sv are the true and predicted values for SVR
# Assuming y_true_rf and y_pred_rf are the true and predicted values for Random Forest

# Perform t-test
t_stat, p_value = stats.ttest_rel(mean_squared_error(y_test, y_pred), mean_squared_error(y_test, y_pred2))

# Check significance
if p_value < 0.05:
    print("There is a significant difference between SVR and Random Forest.")
else:
    print("There is no significant difference.")
```

Screenshots of the time complexity matrix

```
!pip install ipython-autotime
%load_ext autotime
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(x_train, y_train)

Collecting ipython-autotime
  Downloading ipython_autotime-0.3.2-py2.py3-none-any.whl (7.0 kB)
Requirement already satisfied: ipython in /usr/local/lib/python3.10/site-packages (5.2.2)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/site-packages (from ipython)
Collecting jedi>=0.16 (from ipython->ipython-autotime)
  Downloading jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
                                             1.6/1.6 MB 10.2 MB
Requirement already satisfied: decorator in /usr/local/lib/python3.10/site-packages (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/site-packages (0.7.2)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/site-packages (4.2.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,>=3.0.1,<3.1 in /usr/local/lib/python3.10/site-packages (3.0.10)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/site-packages (2.11.2)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/site-packages (0.2.0)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/site-packages (0.1.2)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/site-packages (4.3.0)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/site-packages (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/site-packages (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/site-packages (0.2.5)
Installing collected packages: jedi, ipython-autotime
Successfully installed ipython-autotime-0.3.2 jedi-0.19.1
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py: column_or_1d(y, warn=True)

  ▾ SVR
    SVR()

time: 7.49 ms (started: 2023-12-20 07:02:13 +00:00)
```

```
!pip install ipython-autotime
%load_ext autotime
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(x_train, y_train)
```

```
Requirement already satisfied: ipython-autotime in /usr/local/lib/python3.10/dist-packages/ipython_autotime-0.1.0-py3.10.egg
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-packages/ipython-8.10.0-py3.10.egg
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages/setuptools-59.2.0-py3.10.egg
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages/jedi-0.18.0-py3.10.egg
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages/decorator-4.4.2-py3.10.egg
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages/pickleshare-0.7.2-py3.10.egg
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-packages/traitlets-4.2.1-py3.10.egg
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1 in /usr/local/lib/python3.10/dist-packages/prompt_toolkit-3.0.10-py3.10.egg
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages/pygments-2.12.0-py3.10.egg
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages/backcall-0.2.0-py3.10.egg
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages/matplotlib_inline-0.1.2-py3.10.egg
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages/pexpect-4.3.1-py3.10.egg
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages/parso-0.8.3-py3.10.egg
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages/ptyprocess-0.7.0-py3.10.egg
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages/wcwidth-0.2.5-py3.10.egg
The autotime extension is already loaded. To reload it, use
  %reload_ext autotime
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py: column_or_1d(y, warn=True)
```

▼ SVR

SVR()

```
time: 7.24 s (started: 2023-12-20 07:03:03 +00:00)
```



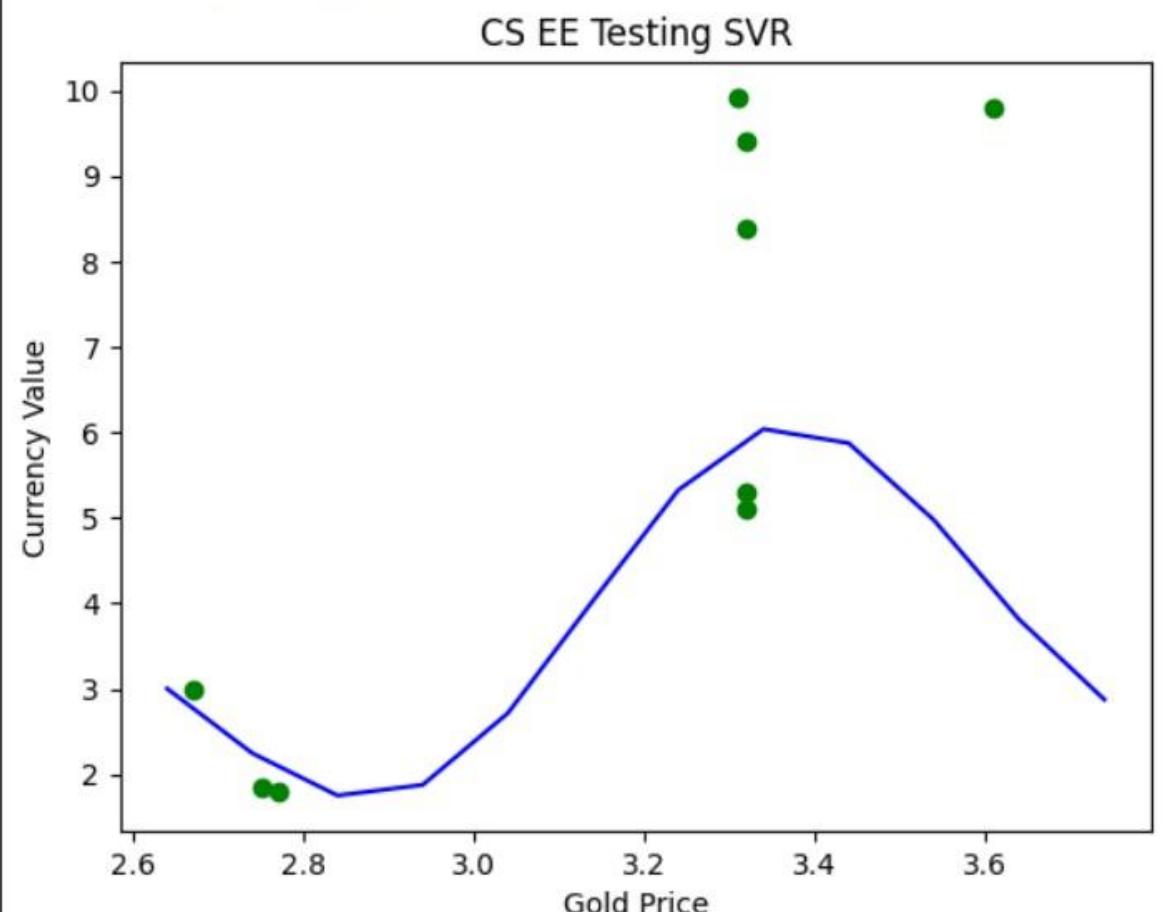
```
!pip install ipython-autotime  
%load_ext autotime  
from sklearn.svm import SVR  
regressor = SVR(kernel = 'rbf')  
regressor.fit(x_train, y_train)
```

```
Requirement already satisfied: ipython-autotime in /usr/local/lib/python3.10/  
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-pack  
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/  
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-p  
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-pa  
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-  
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/di  
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 i  
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-pac  
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-pac  
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10  
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-  
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.  
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/d  
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-pack  
The autotime extension is already loaded. To reload it, use:  
    %reload_ext autotime  
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: Dat  
    y = column_or_1d(y, warn=True)
```

```
▼ SVR  
SVR()
```

```
time: 6.95 s (started: 2023-12-20 07:03:50 +00:00)
```

```
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/site-packages/parso-0.8.3-py3.10.egg (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/site-packages/ptyprocess-0.7.0-py3.10.egg (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages/wcwidth-0.2.5-py3.10.egg (0.2.5)
The autotime extension is already loaded. To reload it, use:
%reload_ext autotime
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
time: 7.41 s (started: 2023-12-20 07:08:31 +00:00)
```

Gold Price

```
time: 6.32 s (started: 2023-12-20 07:09:25 +00:00)
```

Gold Price

```
time: 7.23 s (started: 2023-12-20 07:10:01 +00:00)
```

50 datas

```
time: 6.95 s (started: 2023-12-20 07:19:08 +00:00)
```

```
time: 6.14 s (started: 2023-12-20 07:19:37 +00:00)
```

```
time: 6.04 s (started: 2023-12-20 07:20:44 +00:00)
```

```
time: 7.13 s (started: 2023-12-20 07:21:26 +00:00)
```

```
time: 7.22 s (started: 2023-12-20 07:22:00 +00:00)
```

```
time: 6.86 s
```

75 datas

```
time: 5.94 s (started: 2023-12-20 07:30:26 +00:00)
```

```
time: 7.75 s (started: 2023-12-20 07:32:43 +00:00)
```

```
time: 11.5 s (started: 2023-12-20 07:33:12 +00:00)
```

```
time: 8.53 s (started: 2023-12-20 07:33:46 +00:00)
```

```
time: 6.7 s (started: 2023-12-20 07:34:38 +00:00)
```

```
time: 15.4 s (started: 2023-12-20 07:35:02 +00:00)
```

99datad

```
time: 7.55 s (started: 2023-12-20 07:41:40 +00:00)
```

```
time: 7.06 s (started: 2023-12-20 07:42:18 +00:00)
```

```
time: 7.34 s (started: 2023-12-20 07:43:12 +00:00)
```

Gold Price

```
time: 7.52 s (started: 2023-12-20 07:43:53 +00:00)
```

```
time: 6.12 s (started: 2023-12-20 07:45:03 +00:00)
```

```
time: 7.52 s (started: 2023-12-20 07:45:50 +00:00)
```

RFR

```
time: 6.21 s (started: 2023-12-20 07:10:56 +00:00)
```

```
time: 5.96 s (started: 2023-12-20 07:11:53 +00:00)
```

```
time: 5.95 s (started: 2023-12-20 07:12:37 +00:00)
```

```
time: 6.33 s (started: 2023-12-20 07:13:42 +00:00)
```

```
time: 6.23 s (started: 2023-12-20 07:14:41 +00:00)
```

```
time: 6.32 s (started: 2023-12-20 07:15:13 +00:00)
```

50

```
time: 5.96 s (started: 2023-12-20 07:23:23 +00:00)
```

```
time: 6.71 s (started: 2023-12-20 07:25:51 +00:00)
```

```
time: 6.31 s (started: 2023-12-20 07:27:02 +00:00)
```

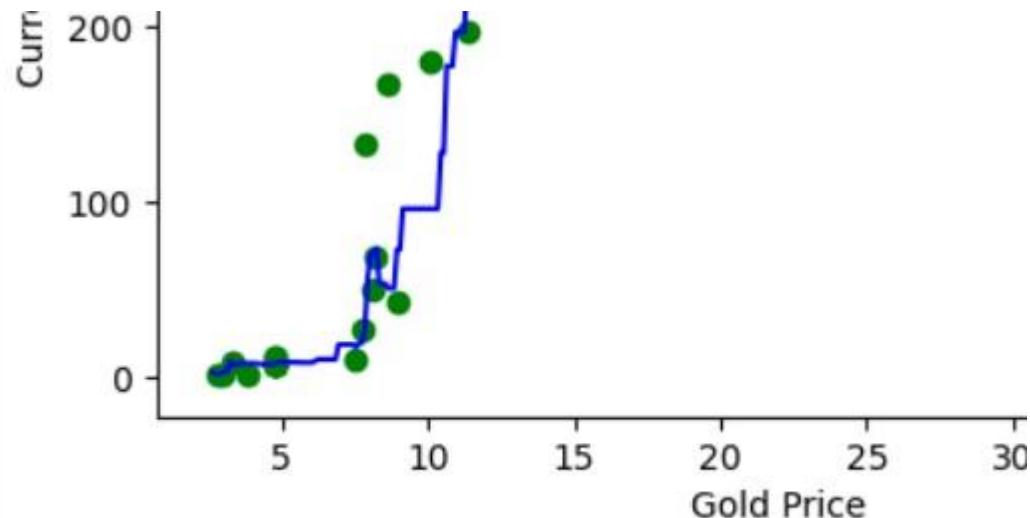
```
time: 6.92 s (started: 2023-12-20 07:27:34 +00:00)
```

75 datas

time: 5.85 s (started: 2023-12-20 07:36:04 +00:00)

time: 7.87 s (started: 2023-12-20 07:36:37 +00:00)

time: 12.5 s (started: 2023-12-20 07:37:19 +00:00)



time: 7.04 s (started: 2023-12-20 07:37:58 +00:00)

time: 13.8 s (started: 2023-12-20 07:38:23 +00:00)

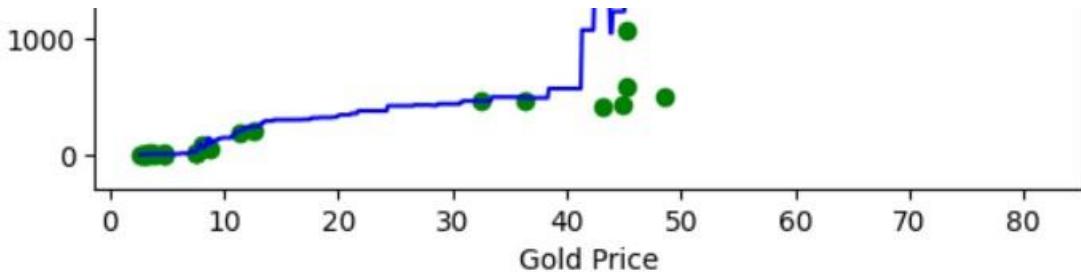
time: 14.2 s (started: 2023-12-20 07:38:54 +00:00)

99datas

time: 7.67 s (started: 2023-12-20 07:46:20 +00:00)

time: 8.66 s (started: 2023-12-20 07:47:53 +00:00)

time: 13.1 s (started: 2023-12-20 07:48:21 +00:00)



time: 12.4 s (started: 2023-12-20 07:48:54 +00:00)

time: 6.23 s (started: 2023-12-20 07:49:36 +00:00)

time: 16.2 s (started: 2023-12-20 07:49:59 +00:00)

Additional Resources and Notes :

Tools Used

Google colab - which consists of all the libraries required for the whole process such as plotting the data into a graph, error report, ect...

Dataset - datas of the stock prices

Github - to store the code in cloud along with google colab

General Disadvantages of ML :

Less accurate predictions due to model getting trained to improper

datasets Outliers

Size of data - show graphs with different sizes of dataset and evaluate on its

performance Missing datas

Categorical data of strings

If transformed into numerical in incorrect way such as random numbers then it will understand that it has an order and gets confused leading to incorrect prediction

External influencers of the data

Different units of data (such as money value and time(year))

SOLUTIONS FOR DISADVANTAGES

Feature scaling - (such as data sets without being featured scale talk about what is feature scaling and how is it important in the predictions refer to the feature scaling notes)

Taking care of missing data - (such as ignoring blank spaces [refer to the below notes for evaluation](#))

ML Process

Data Pre-Processing

- Importing data
- Cleaning data
 - Taking care of missing data
 - Separating input and output datas
- Feature scaling
- Splitting into training and test sets

Modeling

- Building the model
- Training the model
- Making the predictions

Evaluation

- Calculate performance metrics
- Make a verdict (good fitting model or not)

Splitting data into training and test set

You split the data for training so that the model recognizes the relation between the variables, hence 80% of the dataset is used to train the model in this course to understand accurate relation.

And 20% of the set is used to test the set, basically to check whether the predictions we made are correct

Feature Scaling

Is the process where the datas are brought to a 0-1 range for the datas to be formed into clusters(groups) for understanding of dataset well, and model to understand the datas well, for example if a database holds peoples weights which ranges between 15kg to 100kg those datas are transformed into 0-1 there can be outliers also in the transformation. This is done in 2 *in this course* ways as mentioned below

Feature scaling is always applied in columns

Normalization

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where

x_{scaled} is the new

transformed dataX - is the
data we are transforming
Xmin - is the minimum value in the dataset

Xmax - is the maximum value in the dataset

Standardization

$$x' = \frac{x - \bar{x}}{\sigma}$$

Where

x' is the new transformed data

x is the data we are transforming

\bar{x} is the average value in the dataset

σ is the standard deviation of the set

But this transforms the data set values to the range -3 to 3

Minor difference is standaridatison is subtracted by average and divided by standard deviation

So this is done so that the super powered datas such as below if they want cluser/group the purple person who gets 60k dollars at the age of 44 who would it make more sense to blue or red according to the age value it makes more sense if we cluster him with blue but compare to the difference of the variable 1 and 4 the numbers 10,000 and 8,000 are bigger so clustering purple guy with red guy makes more sense as the value is bigger(the purple is 1 year younger than the blue but 10,000 dollars lesser in oney value so obvi the money value is taken into count as the difference is bigger so tha age value is ignored) and their difference is small