SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

Title of the Project

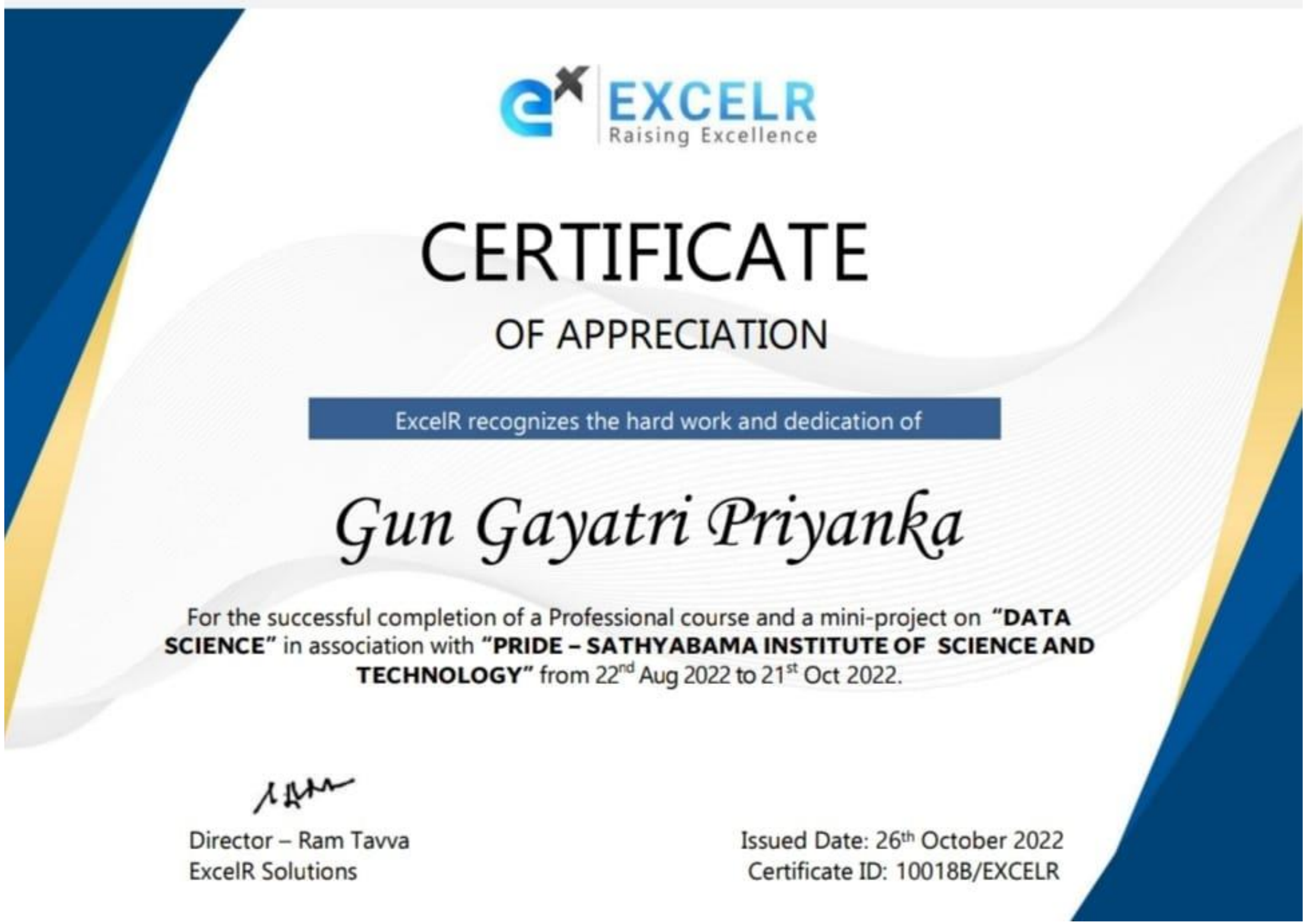
Project Supervisor: Dr. M Selvi
Name of the Student: Gayatri priyanka

Register Number: 40110372

Presentation Outline

- Course Certificate
- Introduction
- Objectives
- System Architecture / Ideation Map
- Module Implementation
- Application Snapshots
- Results and Discussions
- Conclusion & Future work
- References

Course Certificate



EXCELR
Raising Excellence

CERTIFICATE

OF APPRECIATION

ExcelR recognizes the hard work and dedication of

Gun Gayatri Priyanka

For the successful completion of a Professional course and a mini-project on **"DATA SCIENCE"** in association with **"PRIDE – SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY"** from 22nd Aug 2022 to 21st Oct 2022.

Director – Ram Tavva
ExcelR Solutions

Issued Date: 26th October 2022
Certificate ID: 10018B/EXCELR

Introduction

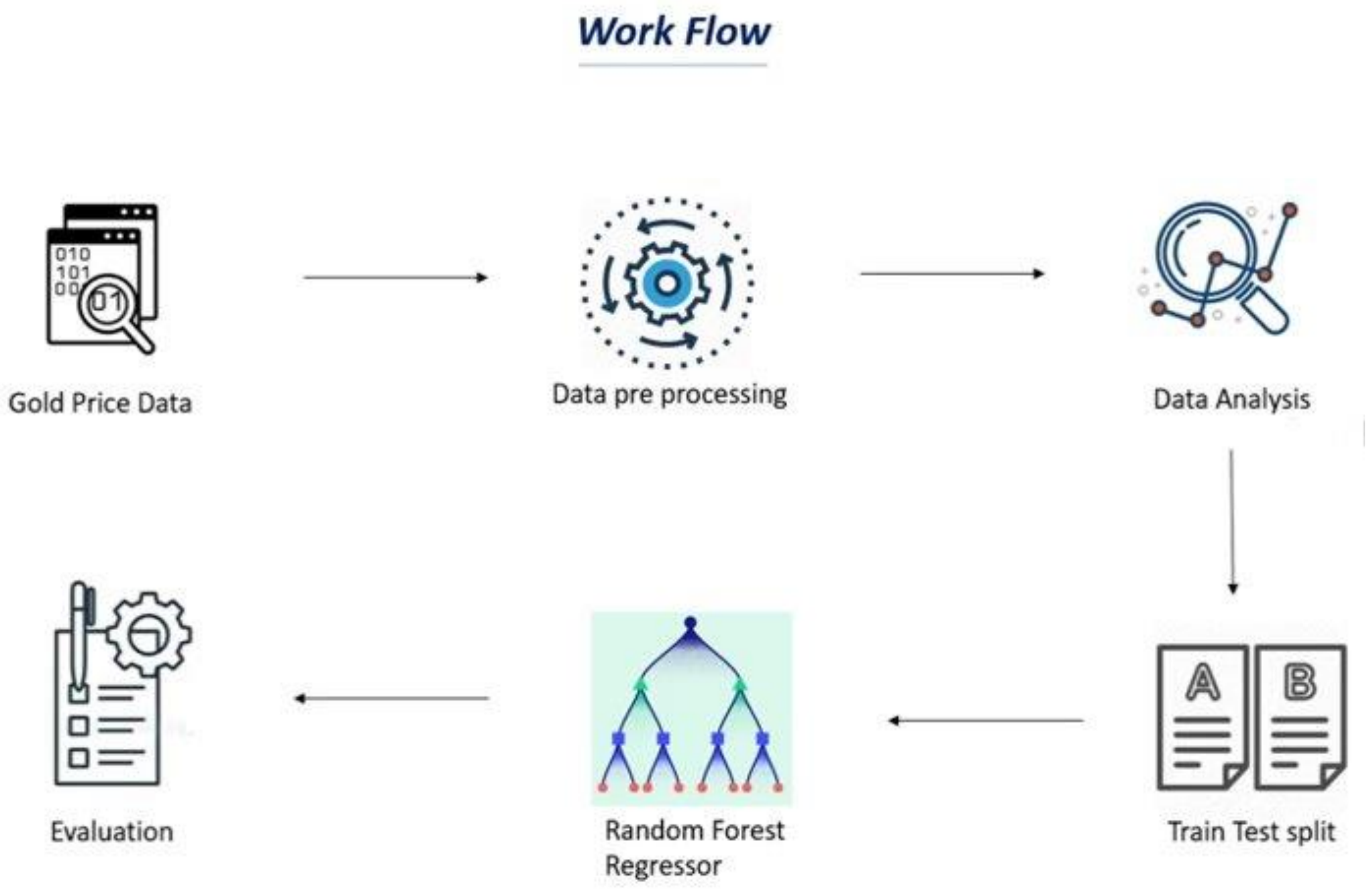
- Gold is one of the precious metals. It has been used as currency, for jewelry and other purposes. It is used as medium for money or exchange because of its limited supply and high value.
- It also reflects the country's economic strength and hence many companies and individuals started to invest in gold reserves. Due to its increasing value, many people considered gold as an attractive investment.
- Since gold is stored and accumulated over years, the influence of an year's production on its price is less. The price of gold depends on currency fluctuations and other economic variables. The raise of gold prices and fall of prices in other markets has attracted more investors to invest in gold market. These changes in the price of gold made the investments risky and a fear has been developed that these prices would decrease.

Objectives

The Project titled 'GOLD PRICE PREDICTION' predicts the gold's price based on:

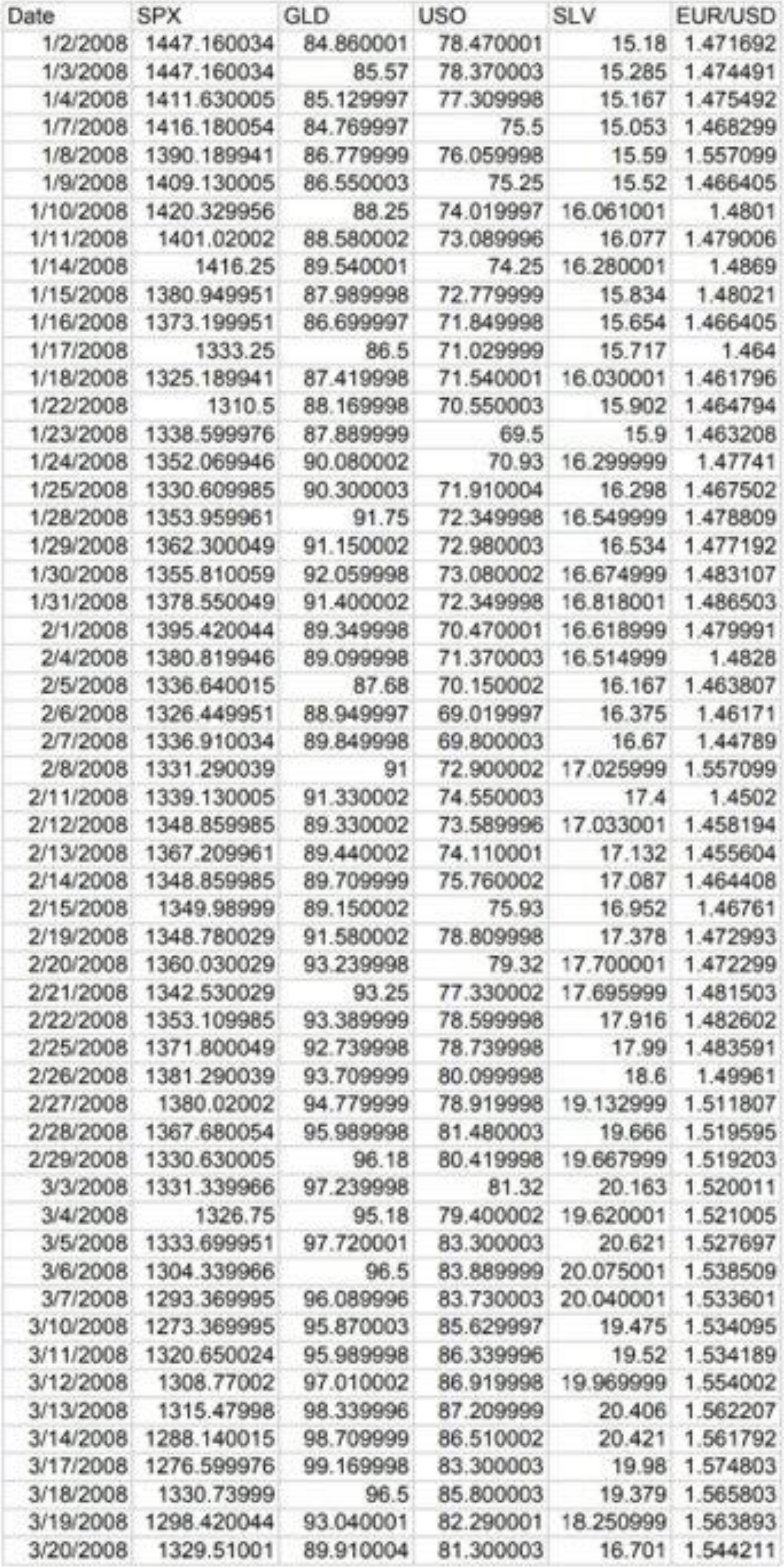
- Date — mm/dd/yyyy
- SPX — is a free-float weighted measurement stock market index of the 500 largest companies listed on stock exchanges in the United States.
- GLD — Gold Price
- USO — United States Oil Fund
- SLV — Silver Price
- EUR/USD — currency pair quotation of the Euro against the US

System Architecture/ Ideation Map



Gold price data

- gold prices are provided by several data feed providers, in the event that there is a failure in one data feed provider we switch to one of our other providers.
- This price is quoted in US dollars. Where the gold price is displayed in currencies other than the US dollar, it is converted into the local currency.
- Like all prices, the gold price reflects not only the inherent value of gold, but also the relative strength of the currency in which it is quoted. For example, the dollar price of gold may increase more in percentage terms than the Euro price of gold, to the extent that the change in price is a reflection of dollar weakness (in this case, against Euros) rather than an intrinsic change in gold market fundamentals.



Date	SPX	GLD	USO	SLV	EUR/USD
1/2/2008	1447.160034	84.860001	78.470001	15.18	1.471692
1/3/2008	1447.160034	85.57	78.370003	15.285	1.474491
1/4/2008	1411.630005	85.129997	77.309998	15.167	1.475492
1/7/2008	1416.180054	84.769997	75.5	15.053	1.468299
1/8/2008	1390.189941	86.779999	76.059998	15.59	1.557099
1/9/2008	1409.130005	86.550003	75.25	15.52	1.466405
1/10/2008	1420.329956	88.25	74.019997	16.061001	1.4801
1/11/2008	1401.02002	88.580002	73.089996	16.077	1.479006
1/14/2008	1416.25	89.540001	74.25	16.280001	1.4869
1/15/2008	1380.949951	87.989998	72.779999	15.834	1.48021
1/16/2008	1373.199951	86.699997	71.849998	15.654	1.466405
1/17/2008	1333.25	86.5	71.029999	15.717	1.464
1/18/2008	1325.189941	87.419998	71.540001	16.030001	1.461796
1/22/2008	1310.5	88.169998	70.550003	15.902	1.464794
1/23/2008	1338.599976	87.889999	69.5	15.9	1.463208
1/24/2008	1352.069946	90.080002	70.93	16.299999	1.47741
1/25/2008	1330.609985	90.300003	71.910004	16.298	1.467502
1/28/2008	1353.959961	91.75	72.349998	16.549999	1.478809
1/29/2008	1362.300049	91.150002	72.980003	16.534	1.477192
1/30/2008	1355.810059	92.059998	73.080002	16.674999	1.483107
1/31/2008	1378.550049	91.400002	72.349998	16.818001	1.486503
2/1/2008	1395.420044	89.349998	70.470001	16.618999	1.479991
2/4/2008	1380.819946	89.099998	71.370003	16.514999	1.4828
2/5/2008	1336.640015	87.68	70.150002	16.167	1.463807
2/6/2008	1326.449951	88.949997	69.019997	16.375	1.46171
2/7/2008	1336.910034	89.849998	69.800003	16.67	1.44789
2/8/2008	1331.290039	91	72.900002	17.025999	1.557099
2/11/2008	1339.130005	91.330002	74.550003	17.4	1.4502
2/12/2008	1348.859985	89.330002	73.589996	17.033001	1.458194
2/13/2008	1367.209961	89.440002	74.110001	17.132	1.455604
2/14/2008	1348.859985	89.709999	75.760002	17.087	1.464408
2/15/2008	1349.98999	89.150002	75.93	16.952	1.46761
2/19/2008	1348.780029	91.580002	78.809998	17.378	1.472993
2/20/2008	1360.030029	93.239998	79.32	17.700001	1.472299
2/21/2008	1342.530029	93.25	77.330002	17.695999	1.481503
2/22/2008	1353.109985	93.389999	78.599998	17.916	1.482602
2/25/2008	1371.800049	92.739998	78.739998	17.99	1.483591
2/26/2008	1381.290039	93.709999	80.099998	18.6	1.49961
2/27/2008	1380.02002	94.779999	78.919998	19.132999	1.511807
2/28/2008	1367.680054	95.989998	81.480003	19.666	1.519595
2/29/2008	1330.630005	96.18	80.419998	19.667999	1.519203
3/3/2008	1331.339966	97.239998	81.32	20.163	1.520011
3/4/2008	1326.75	95.18	79.400002	19.620001	1.521005
3/5/2008	1333.699951	97.720001	83.300003	20.621	1.527697
3/6/2008	1304.339966	96.5	83.889999	20.075001	1.538509
3/7/2008	1293.369995	96.089996	83.730003	20.040001	1.533601
3/10/2008	1273.369995	95.870003	85.629997	19.475	1.534095
3/11/2008	1320.650024	95.989998	86.339996	19.52	1.534189
3/12/2008	1308.77002	97.010002	86.919998	19.969999	1.554002
3/13/2008	1315.47998	98.339996	87.209999	20.406	1.562207
3/14/2008	1288.140015	98.709999	86.510002	20.421	1.561792
3/17/2008	1276.599976	99.169998	83.300003	19.98	1.574803
3/18/2008	1330.73999	96.5	85.800003	19.379	1.565803
3/19/2008	1298.420044	93.040001	82.290001	18.250999	1.563893
3/20/2008	1329.51001	89.910004	81.300003	16.701	1.544211

Example of some of the dataset taken for this regression.

DATA PREPROCESSING

- Data preprocessing is required when the data is incomplete, inconsistent or noisy. The data collected was noisy, so we performed outlier analysis and removed the noisy data. The data transformation is also done by performing normalization in which the data in each attribute is scaled between the range 0 to 1.

ANALYZING THE DATA.

WE SHALL ANALYZE THE DATA DEPENDING ON THE TERMS WE NEED:

- DATE — MM/DD/YYYY
- SPX — IS A FREE-FLOAT WEIGHTED MEASUREMENT STOCK MARKET INDEX OF THE 500 LARGEST COMPANIES LISTED ON STOCK EXCHANGES IN THE UNITED STATES.
- GLD — GOLD PRICE
- USO — UNITED STATES OIL FUND
- SLV — SILVER PRICE
- EUR/USD — CURRENCY PAIR QUOTATION OF THE EURO AGAINST THE US

TRAIN TEST SPLIT

- **Split the data into target values and feature values :**

-
  ```
  X = gold_data.drop(['Date','GLD'],axis=1)
  Y = gold_data['GLD']
  ```
- As there were no empty cells, we could readily begin with the table manipulations;
- 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)

TRAIN TEST SPLIT

- **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.
- X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state=2)
- Let’s understand the variables by knowing what type of values they store :
- X_train: contains a random set of values from variable ‘ X ‘
- Y_train: contains the output (the price of Gold) of the corresponding value of X_train.
- X_test: contains a random set of values from variable ‘ X ‘, excluding the ones from X_train(as they are already taken).
- Y_train: contains the output (the price of Gold) of the corresponding value of X_test.
- test_size: represents the ratio of how the data is distributed among X_trai 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

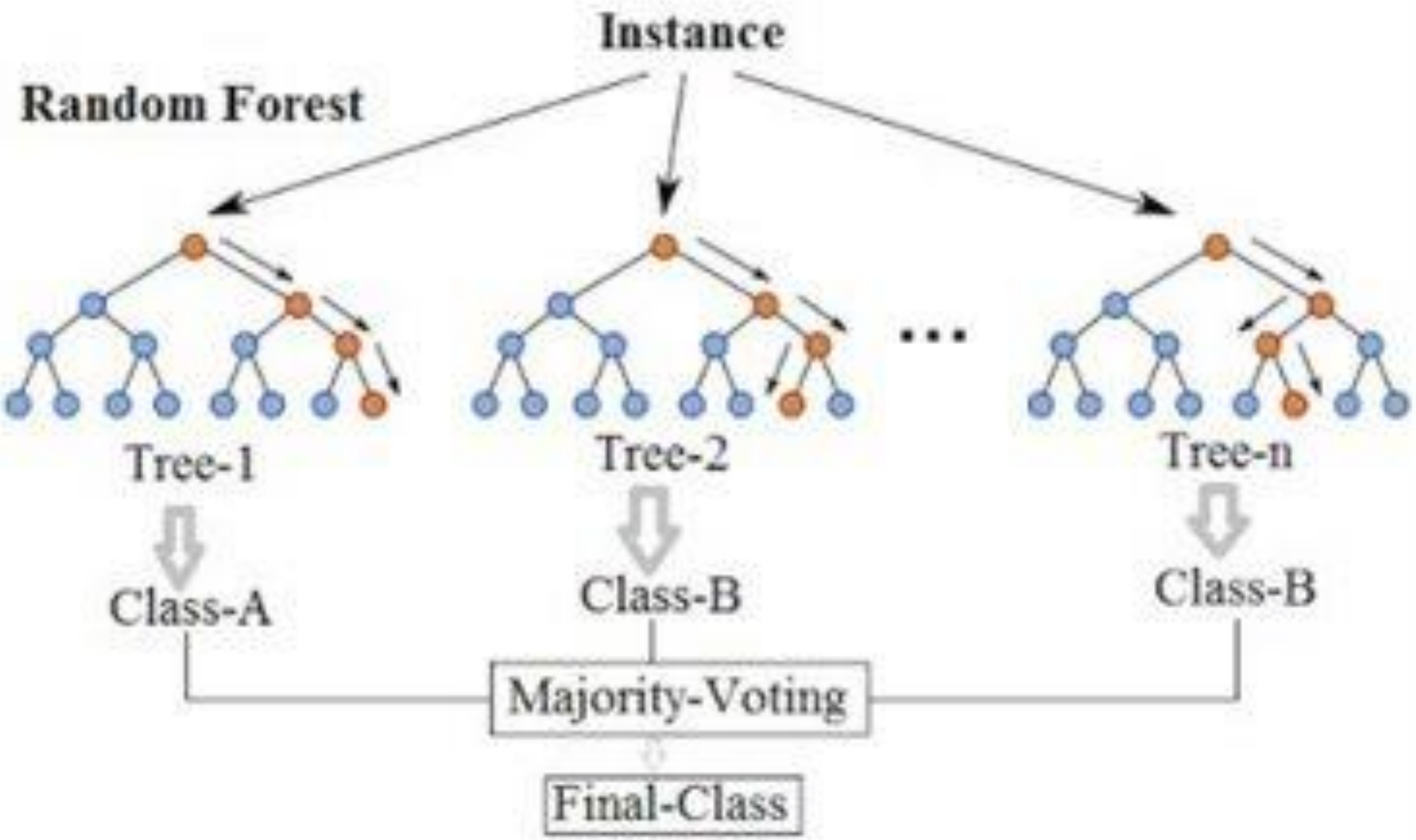
WHY RANDOM FOREST REGRESSION OVER LINEAR REGRESSION?

- **The greater number of trees in the forest leads to higher accuracy** and prevents the problem of overfitting. Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not.
- Linear Models have very few parameters, Random Forests a lot more. That means that **Random Forests will overfit more easily** than a Linear Regression.

RANDOM FOREST REGRESSION

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*
- As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

Random Forest Simplified



HOW DOES RANDOM FOREST ALGORITHM WORK?

- Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.
- The Working process can be explained in the below steps and diagram:
- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

WHY DO WE USE RANDOM FOREST?

- Miscellany: Each tree has a unique attribute, variety and features concerning other trees. Not all trees are the same.
- Immune to the curse of dimensionality: Since a tree is a conceptual idea, it requires no features to be considered. Hence, the feature space is reduced.
- Parallelization: We can fully use the CPU to build random forests since each tree is created autonomously from different data and features.
- Train-Test split: In a Random Forest, we don’t have to differentiate the data for train and test because the decision tree never sees 30% of the data.
- Stability: The final result is based on Bagging, meaning the result is based on majority voting or average.

ADVANTAGES

- RANDOM FOREST IS CAPABLE OF PERFORMING BOTH CLASSIFICATION AND REGRESSION TASKS.
- IT IS CAPABLE OF HANDLING LARGE DATASETS WITH HIGH DIMENSIONALITY.
- IT ENHANCES THE ACCURACY OF THE MODEL AND PREVENTS THE OVERFITTING ISSUE.

DISADVANTAGES

- ALTHOUGH RANDOM FOREST CAN BE USED FOR BOTH CLASSIFICATION AND REGRESSION TASKS, IT IS NOT MORE SUITABLE FOR REGRESSION TASKS.

MODEL EVALUATION

- Let’s now predict the values of the X_test dataset using the predict() method.
- test_data_prediction = regressor.predict(X_test)
- Calculating the R-Squared error from the predicted value. :
- error_score = metrics.r2_score(Y_test, test_data_prediction) print("R squared error : ", error_score)
- The output comes out to be: “R squared error:

DATA COLLECTION AND PROCESSING

- # loading the csv data to a Pandas DataFrame

```
gold_data = pd.read_csv
```

- # print first 5 rows in the dataframe

```
gold_data.head()
```

- # print last 5 rows of the dataframe

```
gold_data.tail()
```

- # number of rows and columns

```
gold_data.shape
```

- # getting some basic informations about the data

```
gold_data.info()
```

- # checking the number of missing values

```
gold_data.isnull().sum()
```

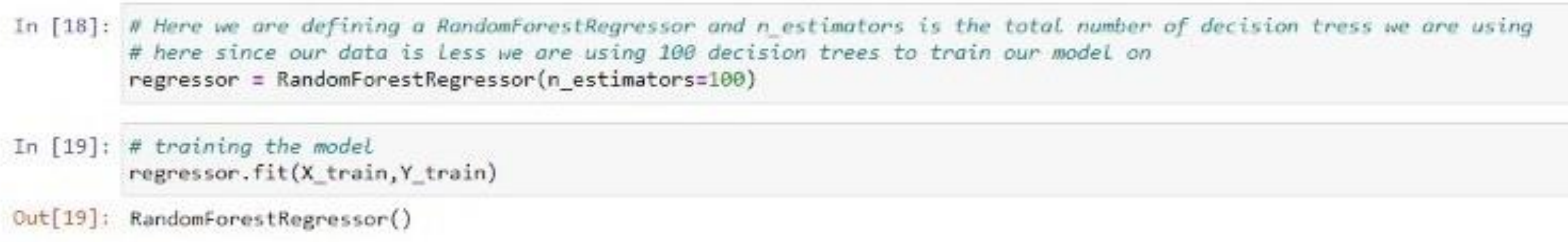
```
# getting the statistical measures of t
he data
gold_data.describe()
```

COMPARING THE ACTUAL VALUES AND PREDICTED VALUES

- Converting the values of Y_test into a list.
- `Y_test = list(Y_test)`
- Now, plotting values of actual prices, versus the predicted prices to know, how close ou predictions were to the actual prices :
-
  ```
  plt.plot(Y_test, color='blue', label = 'Actual Value')
  plt.plot(test_data_prediction, color='green', label='Predicted Value')
  plt.title('Actual Price vs Predicted Price')
  plt.xlabel('Number of values')
  plt.ylabel('GLD Price')
  plt.legend()
  plt.show()
  ```

Project Implementation

Assigning random forest to a variable and pass the regressor to the model and train model
Here,
n_estimator = how many trees we need in random forest model to make a prediction.
Now when after training the model we pass the parameters in which we have our data that has to be trained i.e x_train and y_train
After training lets test our model by passing the data that has to be tested i.e x_test now we get the result of values of absenteeism in hours for the tested data



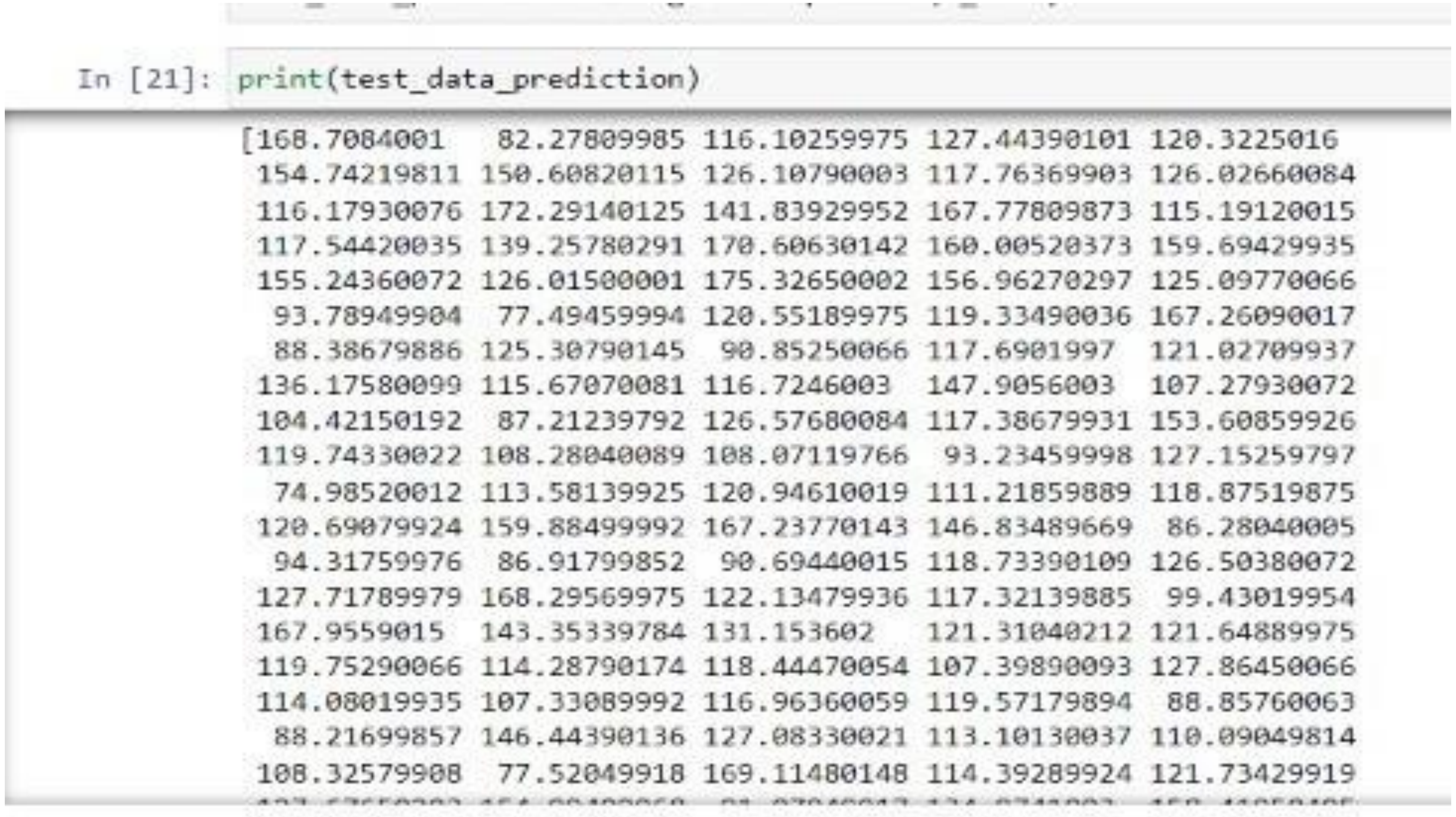
```
In [18]: # Here we are defining a RandomForestRegressor and n_estimators is the total number of decision tress we are using
         # here since our data is less we are using 100 decision trees to train our model on
         regressor = RandomForestRegressor(n_estimators=100)

In [19]: # training the model
         regressor.fit(X_train,Y_train)

Out[19]: RandomForestRegressor()
```

Sample Snapshot

The predicted results of the data which was sent for testing is as follows:



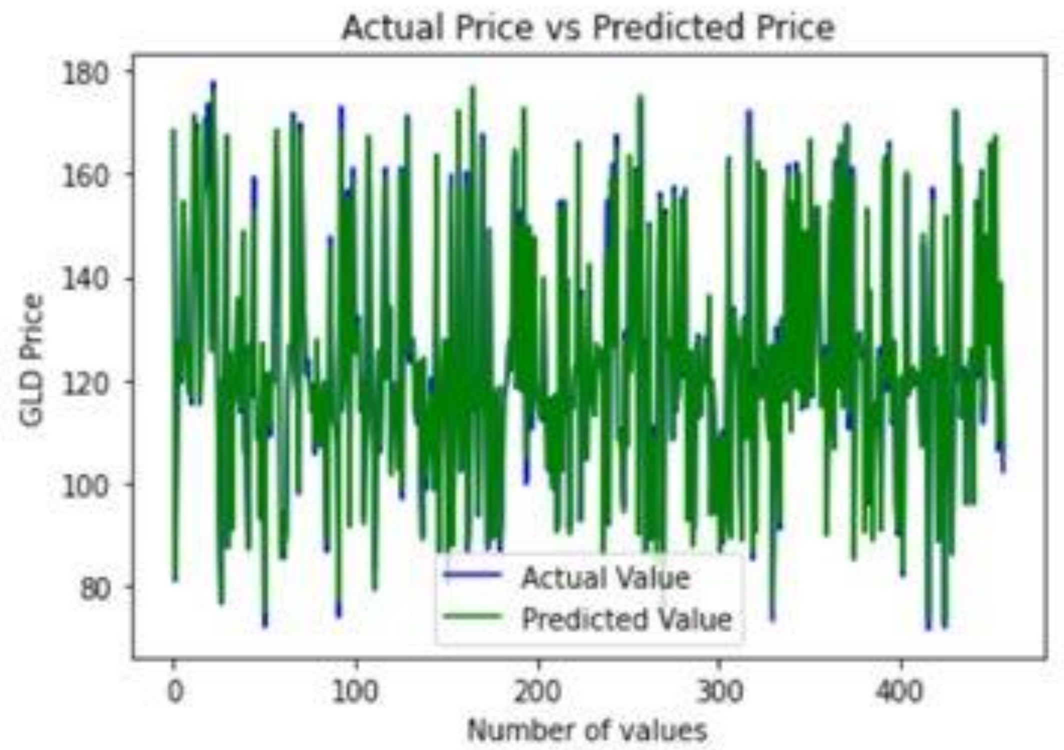
```
In [21]: print(test_data_prediction)
```

```
[168.7084001   82.27809985 116.10259975 127.44390101 120.3225016
 154.74219811 150.60820115 126.10790003 117.76369903 126.02660084
 116.17930076 172.29140125 141.83929952 167.77809873 115.19120015
 117.54420035 139.25780291 170.60630142 160.00520373 159.69429935
 155.24360072 126.01500001 175.32650002 156.96270297 125.09770066
  93.78949904  77.49459994 120.55189975 119.33490036 167.26090017
  88.38679886 125.30790145  90.85250066 117.6901997  121.02709937
 136.17580099 115.67070081 116.7246003  147.9056003  107.27930072
 104.42150192  87.21239792 126.57680084 117.38679931 153.60859926
 119.74330022 108.28040089 108.07119766  93.23459998 127.15259797
  74.98520012 113.58139925 120.94610019 111.21859889 118.87519875
 120.69079924 159.88499992 167.23770143 146.83489669  86.28040005
  94.31759976  86.91799852  90.69440015 118.73390109 126.50380072
 127.71789979 168.29569975 122.13479936 117.32139885  99.43019954
 167.9559015  143.35339784 131.153602   121.31040212 121.64889975
 119.75290066 114.28790174 118.44470054 107.39890093 127.86450066
 114.08019935 107.33089992 116.96360059 119.57179894  88.85760063
  88.21699857 146.44390136 127.08330021 113.10130037 110.09049814
 108.32579908  77.52049918 169.11480148 114.39289924 121.73429919
```

As we train and evaluate the model with improved conditions our model will get used to the different conditions and will able to give result more accurately.

Results and Discussion

By seeing the above fig plotting values of actual prices, versus the predicted prices to know, how close our predictions were to the actual prices . we can observe, that the actual prices and the predicted prices are almost the same, as the two graphs overlap each other. Thus, or model has performed extremely well.



Conclusion

- The main aim of this study is to predict the gold price that is influenced by the economic variables such as stock
- profit exchange, silver price, EUR/USD. In this study, we used the machine learning algorithms such as random forest to predict the price of gold accurately. Considering
- the results obtained, we conclude that the random forest model performed better than the other models.

- For future work, we can improve the results and predict the price more accurately by incorporating the other factors
- such as gold production, crude oil price, platinum price,inflation to the data and by using deep learning. As you saw in this project, we first train a machine learning model, then use the trained model for prediction. Similarly, any model can be made much more precise, by feeding a very large dataset, to get a very accurate score

References

- [1] V. K. F. B. Rebecca Davis, "Modeling and Forecasting of Gold Prices on Financia Markets," American International
- Journal of Contemporary Research, 2014.
- [2] Iftikharul Sami and Khurum Nazir Junejo, "Predicting Future Gold Rates using Machine Learning Approach",
- International Journal of Advanced Computer Science and Applications, 2017.
- [3] D Makala and Z Li, “Prediction of gold price with ARIMA and SVM”, Journal of Physics: Conference Series, 2021.
- [4] Navin, Dr. G. Vadivu, "Big Data Analytics for Gold Price Forecasting Based on Decision Tree Algorithm and Support
- Vector Regression (SVR)", International Journal of Science and Research (IJSR), 2013.
- [5] P. V. M. Vasava, P. G. M. Poddar, Sima P Patel, "Gold Market Analyzer using Selection based Algorithm",
- International Journal of Advanced Engineering Research and Science (IJAERS), 2016.
- [6] Megan Potoski,”Predicting Gold prices”, CS229, Autumn 2013.
- [7] Dr. Abhay Kumar Agarwal, Swati Kumari, "Gold Price Prediction using Machine Learning", International Journal of
- Trend in Scientific Research and Development (ijtsrd), 2020.

THANK YOU