VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Report on

Gold Price Prediction

Submitted in partial fulfillment of the requirements during VII Semester for the degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

Bachelor of Engineering
in
Information Science and Engineering

Submitted by

Krishna Daga 1RN16IS023

UNDER THE GUIDANCE OF

Dr. R Rajkumar

Assistant Professor

Dept. of ISE, RNSIT

ESTD:2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post

Bengaluru – 560 098

2021 – 2022

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ESTD:2001
An Institute with a Difference

CERTIFICATE

Certified that the internship work entitled ***Gold Price Prediction*** has been successfully completed by **Krishna Daga (1RN16IS023)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work for the said degree.

Dr. R Rajkumar	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal
Associate Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

I, **KRISHNA DAGA [USN: 1RN16IS023]** am a student of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the internship project entitled ***Gold Price Prediction*** has been carried out by me and submitted in partial fulfillment of the requirements for the *VII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

KRISHNA DAGA (1RN16IS023)

ABSTRACT

Gold has always occupied a predominant place in country's economies, and among populations. Owing to its characteristics, it is used as a hedging tool or can act as a safe haven in turmoil conditions. The aim of this project titled Gold Price Prediction is to explore the relationship of gold price with various explanatory variables that tend to be considered as indicators of financial and geopolitical crises. On the other hand, the study investigates the possibility of predicting gold price based on these variables.

Historically, gold had been used as a form of currency in various parts of the world. In present times, precious metals like gold are held with central banks of all countries to guarantee re-payment of foreign debts, and also to control inflation which results in reflectingthe financial strength of the country. Recently, emerging world economies, such as China, Russia, and India have been big buyers of gold, whereas USA, South Africa, and Australia are among the big seller of gold. Forecasting rise and fall in the daily gold rates, can help investors to decide when to buy (or sell) the commodity. But Gold prices are dependent on many factors such as prices of other precious metals, prices of crude oil, stock exchange performance, Bonds prices, currency exchange rates etc.

We in this project would forecast gold rates using the most comprehensive set of features and would apply various machine learning algorithms for forecasting and compare their results. We also identify the attributes that highly influence the gold rates. We would use **SPDRGold Trust (GLD) Exchange Traded Fund** data downloaded from https://finance.yahoo.com . We would try to predict Adjusted Close price of GLD ETF, the detail about the data would be described in the Dataset and Input section below.

ACKNOWLEDGEMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all his wisdom.

We place our heartfelt thanks to **Miss. Kavya TC** Professor and HOD, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Aman Upadhyay, Designation, NASTECH**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

Date:
Place: Bangalore

KRISHNA DAGA
USN-1RN16IS023

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The Goal of this project is to accurately predict the future adjusted closing price of Gold ETF across a given period of time in the future. For this project I have used different Machine Learning Algorithms and ensemble the solution model by combining top three performing models to predict the Adjusted closing price of the GLD ETF using a dataset of past prices system.

1.2 PROBLEM STATEMENT AND HIGHLIGHTS

The challenge of this project is to accurately predict the future adjusted closing price of Gold ETF across a given period of time in the future. The problem is a regression problem, because the output value which is the adjusted closing price in this project is continuous value. Various studies have been conducted by researchers to forecast gold rates using different machine learning algorithms with varying degrees of success but until recently the ability to build these models has been restricted to academics. Now with libraries like Scikit-learn anyone can build powerful predictive models. For this project I will use different linear, ensemble and boosting machine learning models to predict the adjusted closing price of the SPDR Gold Trust (GLD) ETF using a dataset of past prices from 18/11/2004 to 01/01/2019.

The Goal of this project is to accurately predict the future adjusted closing price of Gold ETF across a given period of time in the future. For this project I have used different Machine Learning Algorithms and ensemble the solution model by combining top three performing models to predict the adjusted closing price of the GLD ETF using a dataset of past prices.

Things I have learnt by completing this project:

- How to apply machine learning techniques on Time series Data.
- How to perform statistical analysis of Time series Data.
- How to collect and preprocess given data.
- How to ensemble different machine learning models and analyze model's performance.
- How to optimize Machine Learning models to increase accuracy and reduction in error.

1.3 METRICS

I would use Root Mean Square Error and R2 score as my evaluation metrics. Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are. The formula for calculating RMSE is given below

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

Where f is the Predicted or forecasted value and o is observed or Actual value. RMSE is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. R-squared is always between 0 and 100%

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean. So, when we apply both evaluation metrics to our benchmark and solution models, we would choose the one which has lower RMSE value and higher R2 score value.

1.4 HARDWARE REQUIREMENTS

- **Hardware** : Processor Intel dual core and above
- **Operating System** : Windows 7, Windows 8, Windows 10
- **Internet Connection** : Existing telephone lines, Data card or Any Wireless Network
- **Browser**: Google chrome latest version, IExplorer 10;
- **Performance**: The turn-around time of the project will be medium.

CHAPTER 2

WORKING MODEL OF PROJECT

ARCHITECTURE DIAGRAM

An architecture diagram describes what you're building, how stakeholders interact with it, and where constraints lie. Setting out this information helps you and your stakeholders navigate the project and brings clarity across the board.

Architecture diagram for given project is shown below:

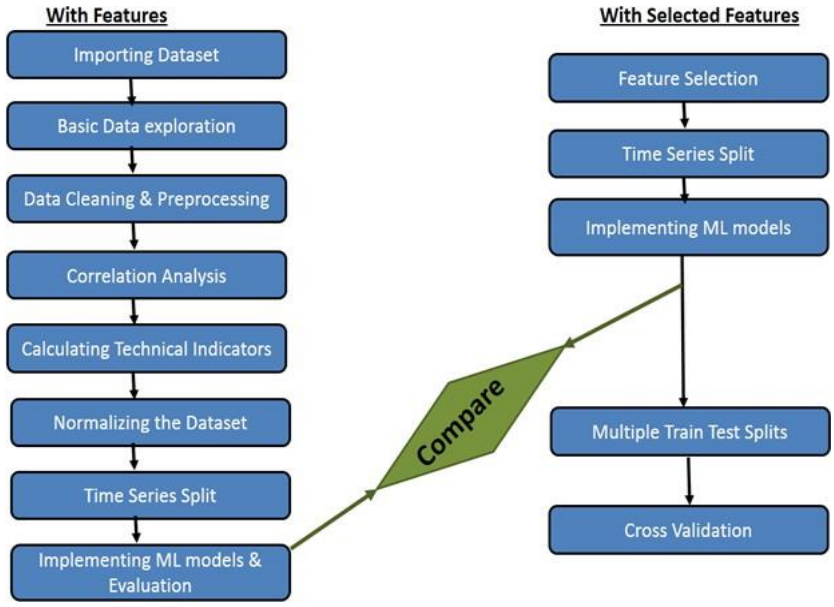

Fig 2.1 Architecture Diagram

Steps Followed –

1. Exploring Gold ETF closing prices
2. Perform statistical analysis
3. Perform Feature Engineering
4. Normalizing the Data
5. Splitting the dataset
6. Implement benchmark machine learning model and different solution models
7. Compare benchmark model with different machine learning models based on evaluation metrics described below.
8. Perform Feature Selection
9. Compare Feature selected models with full feature models.

CHAPTER 3

ANALYSIS

3.1 DATA EXPLORATION

Data for this study is collected from November 18th 2011 to January 1st 2019 from various sources. The data has 1718 rows in total and 80 columns in total. Data for attributes, such as Oil Price, Standard and Poor's (S&P) 500 index,

Dow Jones Index US Bond rates (10 years), Euro USD exchange rates, prices of precious metals Silver and Platinum and other metals such as Palladium and Rhodium, prices of US Dollar Index, Eldorado Gold Corporation and Gold Miners ETF were gathered. The historical data of Gold ETF fetched from Yahoo finance has 7 columns, Date, Open, High, Low, Close, Adjusted Close and Volume, the difference between Adjusted Close and Close is that closing price of a stock is the price of that stock at the close of the trading day.

Whereas the adjusted closing price takes into account factors such as dividends, stock splits and new stock offerings to determine a value. We would use Adjusted Close as our outcome variables which is the value we want to predict. 'SP_open', 'SP_high', 'SP_low', 'SP_close', 'SP_Ajclose', 'SP_volume' of S&P 500 Index, 'DJ_open', 'DJ_high', 'DJ_low', 'DJ_close', 'DJ_Ajclose', 'DJ_volume' of Dow Jones Index, 'EG_open', 'EG_high', 'EG_low', 'EG_close', 'EG_Ajclose', 'EG_volume' of Eldorado Gold Corporation (EGO), 'EU_Price', 'EU_open', 'EU_high', 'EU_low', 'EU_Trend' of EUR USD Exchange rate, 'OF_Price', 'OF_Open', 'OF_High', 'OF_Low', 'OF_Volume', 'OF_Trend' of Brent Crude oil Futures, 'OS_Price', 'OS_Open', 'OS_High', 'OS_Low', 'OS_Trend', of Crude Oil WTI USD, 'SF_Price', 'SF_Open', 'SF_High', 'SF_Low', 'SF_Volume', 'SF_Trend' of Silver Futures, 'USB_Price', 'USB_Open', 'USB_High', 'USB_Low', 'USB_Trend' of US Bond Rate data, 'PLT_Price', 'PLT_Open', 'PLT_High', 'PLT_Low', 'PLT_Trend' of Platinum Price, 'PLD_Price', 'PLD_Open', 'PLD_High', 'PLD_Low', 'PLD_Trend' of Palladium price 'RHO_PRICE' of RhodiumPrices'USDI_Price', 'USDI_Open', 'USDI_High', 'USDI_Low', 'USDI_Volume', 'USDI_Trend' of US dollar Index Price, 'GDX_Open', 'GDX_High', 'GDX_Low', 'GDX_Close', 'GDX_Adj Close', 'GDX_Volume' of Gold Miners ETF, 'USO_Open', 'USO_High', 'USO_Low', 'USO_Close', 'USO_Adj Close', 'USO_Volume' of Oil ETF USO are used as feature engineered variables.

Date	Open	High	Low	Close	Adj Close	Volume	SP_open	SP_high	SP_low	SP_close	...	GDX_Low	GDX_Close
2011-12-15	154.740005	154.949997	151.710007	152.330002	152.330002	21521900	123.029999	123.199997	121.989998	122.180000	...	51.570000	51.680000
2011-12-16	154.309998	155.369995	153.899994	155.229996	155.229996	18124300	122.230003	122.949997	121.300003	121.589996	...	52.040001	52.680000
2011-12-19	155.479996	155.860001	154.360001	154.869995	154.869995	12547200	122.059998	122.320000	120.029999	120.290001	...	51.029999	51.169998
2011-12-20	156.820007	157.429993	156.580002	156.979996	156.979996	9136300	122.180000	124.139999	120.370003	123.930000	...	52.369999	52.990002
2011-12-21	156.979996	157.529999	156.130005	157.160004	157.160004	11996100	123.930000	124.360001	122.750000	124.169998	...	52.419998	52.959999

SP_low	SP_close	...	GDX_Low	GDX_Close	GDX_Adj Close	GDX_Volume	USO_Open	USO_High	USO_Low	USO_Close	USO_Adj Close	USO_Volume
121.989998	122.180000	...	51.570000	51.680000	48.973877	20605600	36.900002	36.939999	36.049999	36.130001	36.130001	12616700
121.300003	121.589996	...	52.040001	52.680000	49.921513	16285400	36.180000	36.500000	35.730000	36.270000	36.270000	12578800
120.029999	120.290001	...	51.029999	51.169998	48.490578	15120200	36.389999	36.450001	35.930000	36.200001	36.200001	7418200
120.370003	123.930000	...	52.369999	52.990002	50.215282	11644900	37.299999	37.610001	37.220001	37.560001	37.560001	10041600
122.750000	124.169998	...	52.419998	52.959999	50.186852	8724300	37.669998	38.240002	37.520000	38.110001	38.110001	10728000

Fig 3.1 Data CSV File

3.2 EXPLORATORY VISUALIZATION

Below image shows the daily return of stock indexes with Gold ETF. In the image it is easily understood that price trend of Standards and Poors (S&P 500) Index and Dow Jones Index is somewhat same. But in some places Gold price shows reverse trend with respect to S&P and Dow Jones. To visualize the data I have used matplotlib library.

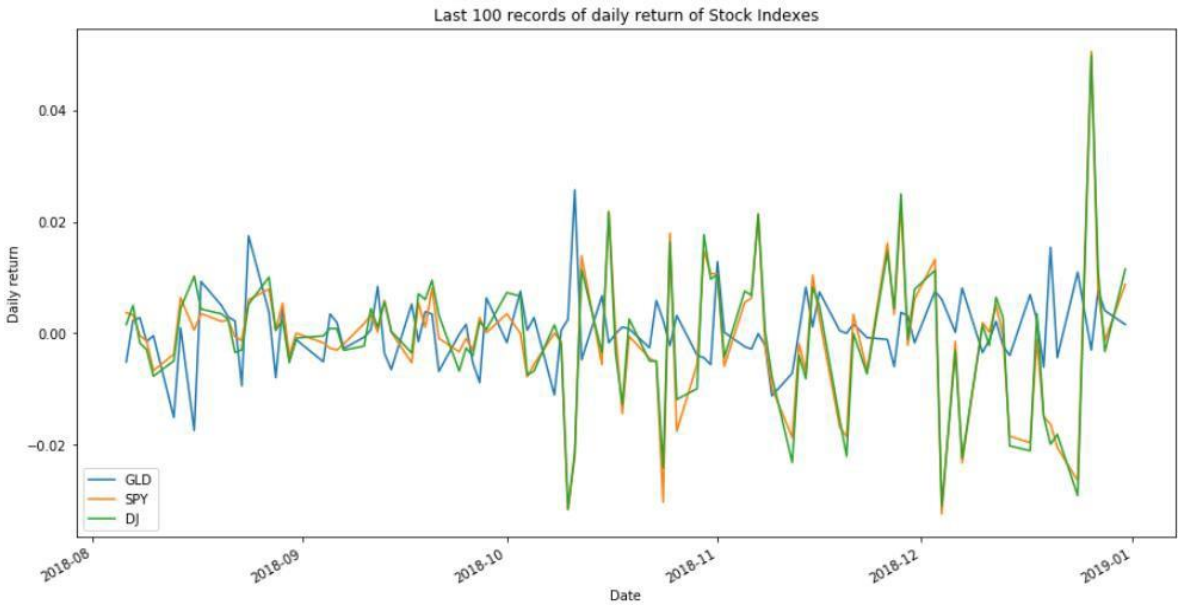

Fig 3.2 Exploratory Visualization of Stock Indexes

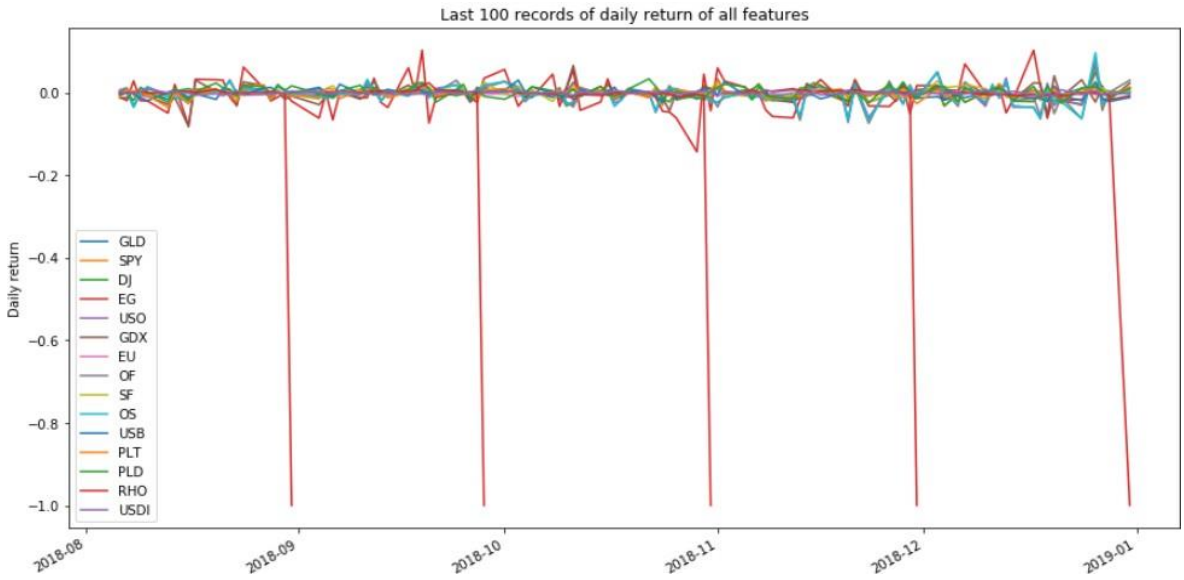

Fig 3.3 Exploratory Visualization of all features

3.3 TECHNICAL INDICATORS

MACD: The Moving Average Convergence-Divergence (MACD) is one of the most powerful and well known indicators in technical analysis. The indicator is comprised of two exponential moving averages (EMAs) that help measure momentum in a security. The MACD is simply the difference between these two moving averages plotted against a centerline, where the centerline is the point at which the two moving averages are equal.

DIF = EMA (close, 12) - EMA (close, 26)

MACD = EMA (DIF, 9) MACD is calculated by first calculating DIF which is the difference of 12 days of Exponential Moving Average (EMA) of Adjusted close price and 26 days EMA. After calculating DIF MACD is calculated which is 9 days EMA of DIF**.**

```
def calculate_MACD(df, nslow=26, nfast=12):
    emaslow = df.ewm(span=nslow, min_periods=nslow, adjust=True, ignore_na=False).mean()
    emafast = df.ewm(span=nfast, min_periods=nfast, adjust=True, ignore_na=False).mean()
    dif = emafast - emaslow
    MACD = dif.ewm(span=9, min_periods=9, adjust=True, ignore_na=False).mean()
    return dif, MACD
```

Fig 3.4 Calculation of MACD

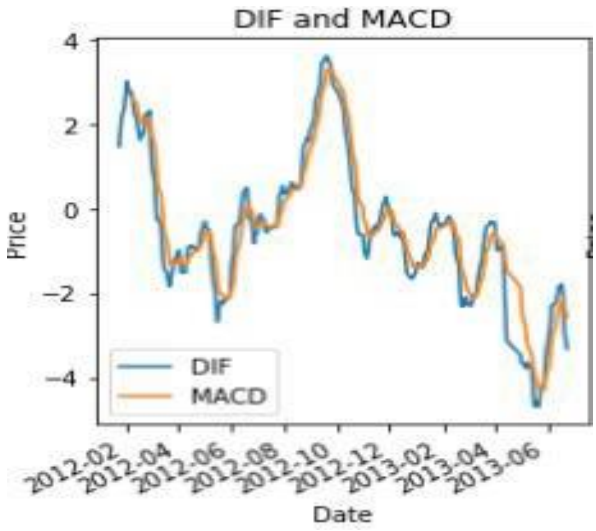

Fig 3.5 DIF and MACD Graph

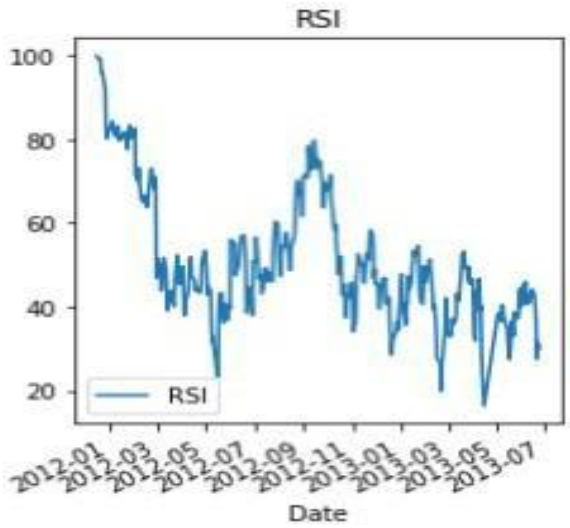

Fig 3.6 RSI Graph

RSI: The Relative Strength Index (RSI) is another well-known momentum indicators that's widely used in technical analysis. The indicator is commonly used to identify overbought and oversold conditions in a security with a range between 0 (oversold) and 100 (overbought). RSI can also be used to identify the general trend. The basic formula is:

RSI = 100 – [100 / (1 + (Avg of Upward Price Change / Avg of Downward Price Change))]

```
def calculate_RSI(df, periods=14):
    # wilder's RSI
    delta = df.diff()
    up, down = delta.copy(), delta.copy()

    up[up < 0] = 0
    down[down > 0] = 0

    rUp = up.ewm(com=periods,adjust=False).mean()
    rDown = down.ewm(com=periods, adjust=False).mean().abs()

    rsi = 100 - 100 / (1 + rUp / rDown)
    return rsi
```

Fig 3.7 Calculation of RSI

Simple Moving Average (SMA): Simply takes the sum of all of the past closing prices over a time period and divides the result by the total number of prices used in the calculation. SMA = (A1+A2+A3+…..An)/n Here An is the adjusted close price of period n, n is the number of total period in our case it is 15 days.

```
def calculate_SMA(df, peroids=15):
    SMA = df.rolling(window=peroids, min_periods=peroids, center=False).mean()
    return SMA
```

Fig 3.8 Calculation of SMA

Bollinger Bands: Bollinger Bands are a technical analysis tool developed by John Bollinger in the 1980s for trading stocks. The bands comprise a volatility indicator that measures the relative high or low of a security's price in relation to previous trades. Volatility is measured using standard deviation, which changes with increases or decreases in volatility. Bollinger Bands are comprised of three lines: upper, middle and lower band. The middle band is a moving average. The upper and lower bands are positioned on either side of the moving average band. Upper Band is 2 standard deviation above the middle band whereas lower band is 2 standard deviation below the middle band.

```
def calculate_BB(df, peroids=15):
    STD = df.rolling(window=peroids,min_periods=peroids, center=False).std()
    SMA = calculate_SMA(df)
    upper_band = SMA + (2 * STD)
    lower_band = SMA - (2 * STD)
    return upper_band, lower_band
```

Fig 3.9 Calculation of BB

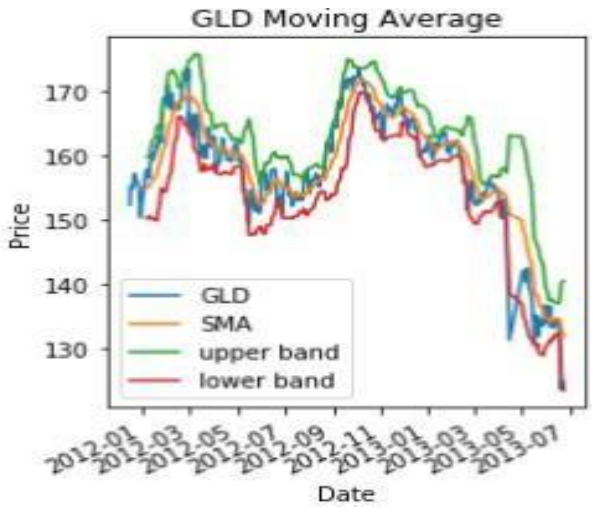

Fig 3.10 GLD Moving Average

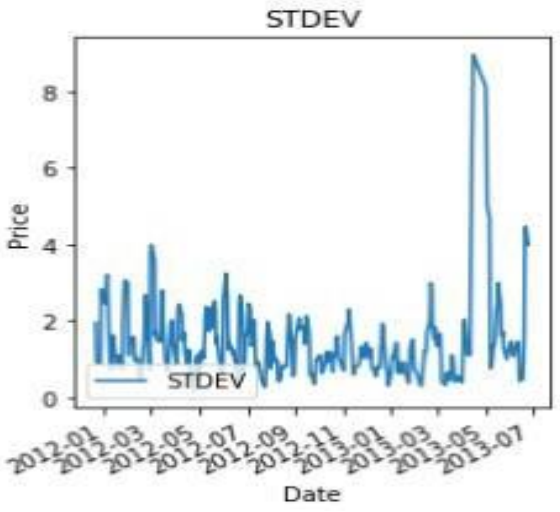

Fig 3.11 STDEV

Open – Close: It is calculated by taking difference of Open & Closing prices of stocks.

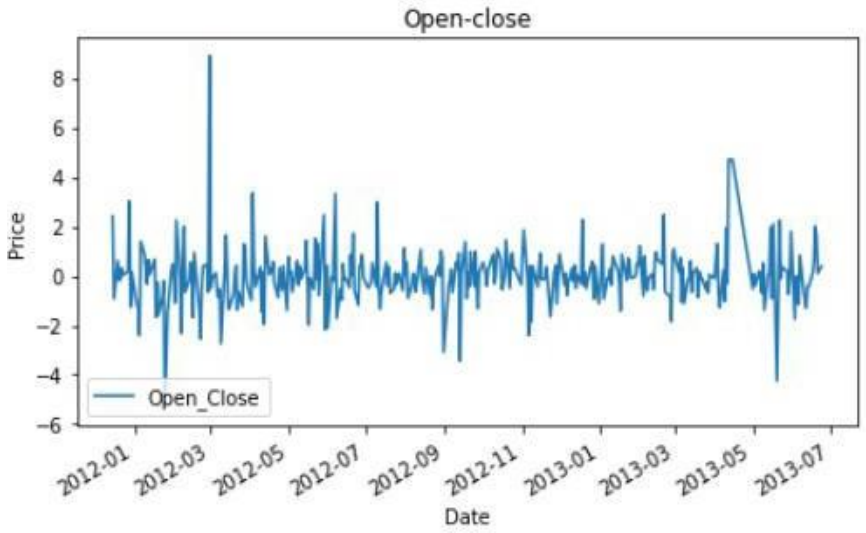

Fig 3.12 Open-Close

High – Low: It is calculated by taking difference between High and Low prices.

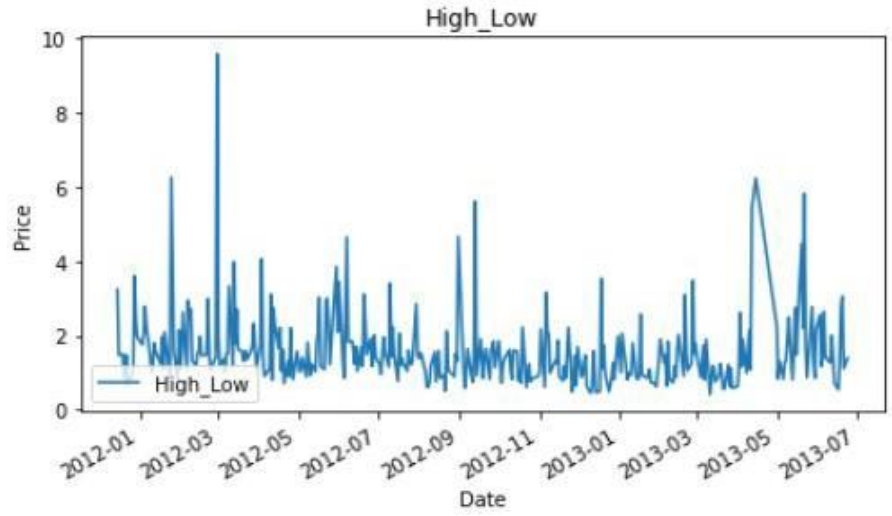

Fig 3.13 High-Low

3.4 SCATTERPLOTS

Below scatterplot shows relationship between Gold Adjusted price and different features such as Stock indexes SPY 500, Dow Jones Index, Prices of Platinum, Palladium, United States 10 year Bonds, Euro-USD Exchange rate etc.

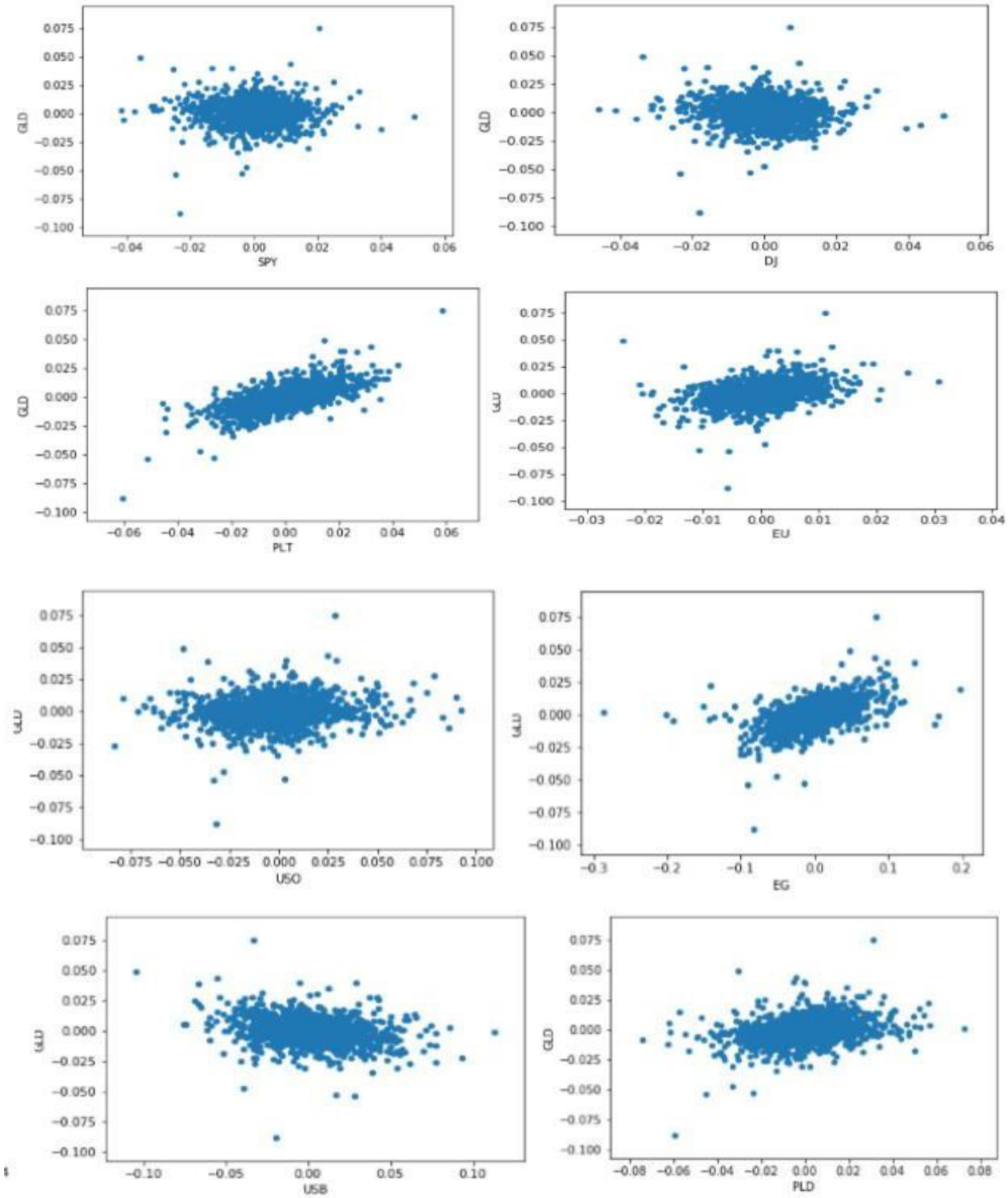

Fig 3.14 Scatterplots

3.5 STATISTICAL MEASURES

Kurtosis is a statistical measure that is used to describe the distribution. Whereas skewness differentiates extreme values in one versus the other tail, kurtosis measures extreme values in either tail. Distributions with large kurtosis exhibit tail data exceeding the tails of the normal distribution (e.g., five or more standard deviations from the mean). Distributions with low kurtosis exhibit tail data that is generally less extreme than the tails of the normal distribution. We have calculated mean, standard deviation and kurtosis of all the stock indexes and Gold rates in terms of Adjusted close price. Positive Kurtosis means more weights in the tailNegative Kurtosis It has as much data in each tail as it does in the peak.

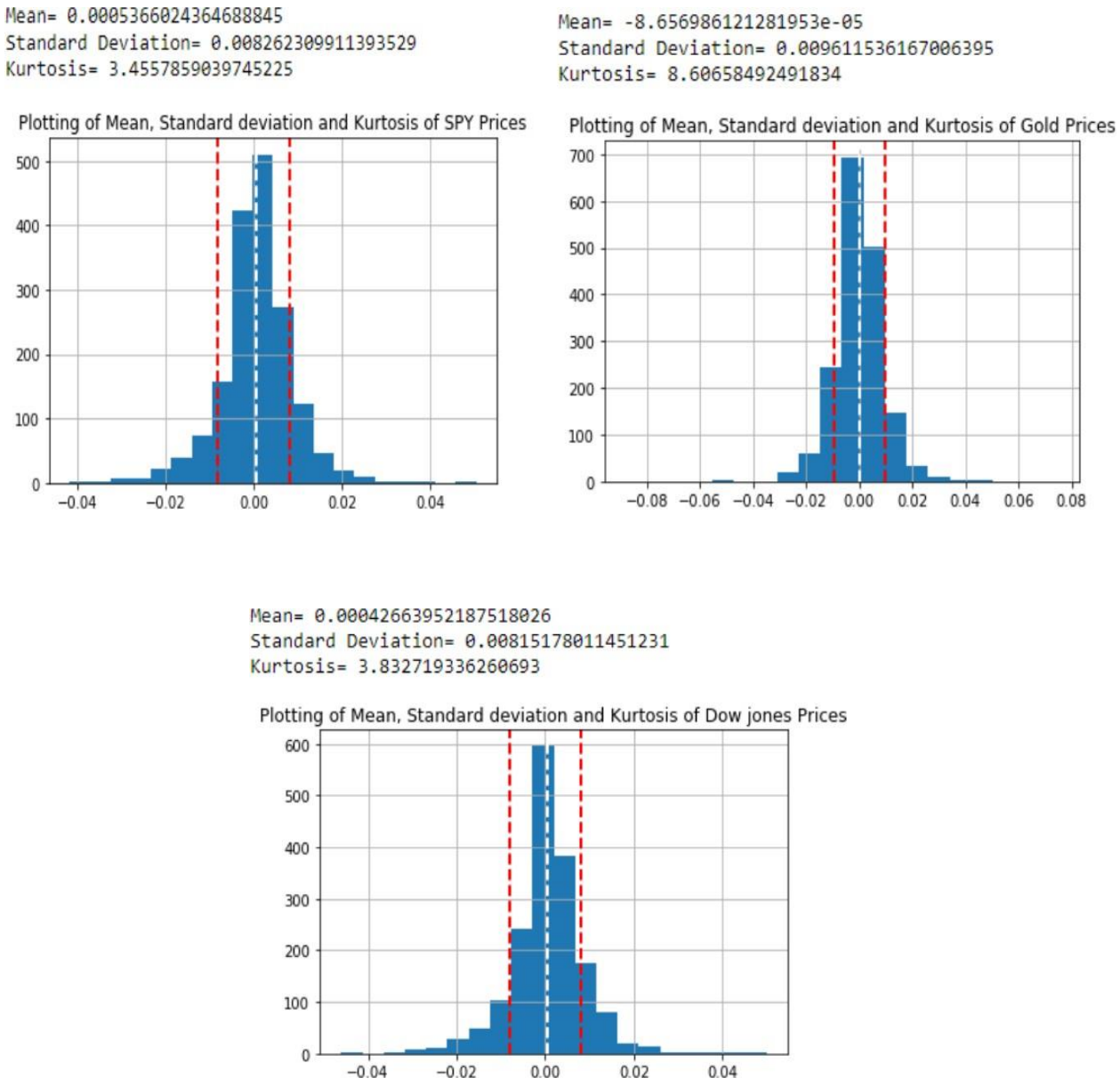

Fig 3.15 Mean, Standard deviation, Kurtosis

3.5 CORRERATIONAL ANALYSIS

In this step we will explore which feature variables are highly correlated with target variable which Adjusted Close price. First we have plotted correlation matrix. As there are so many features in the dataset so it is very difficult to visibly observe the correlation among variables.

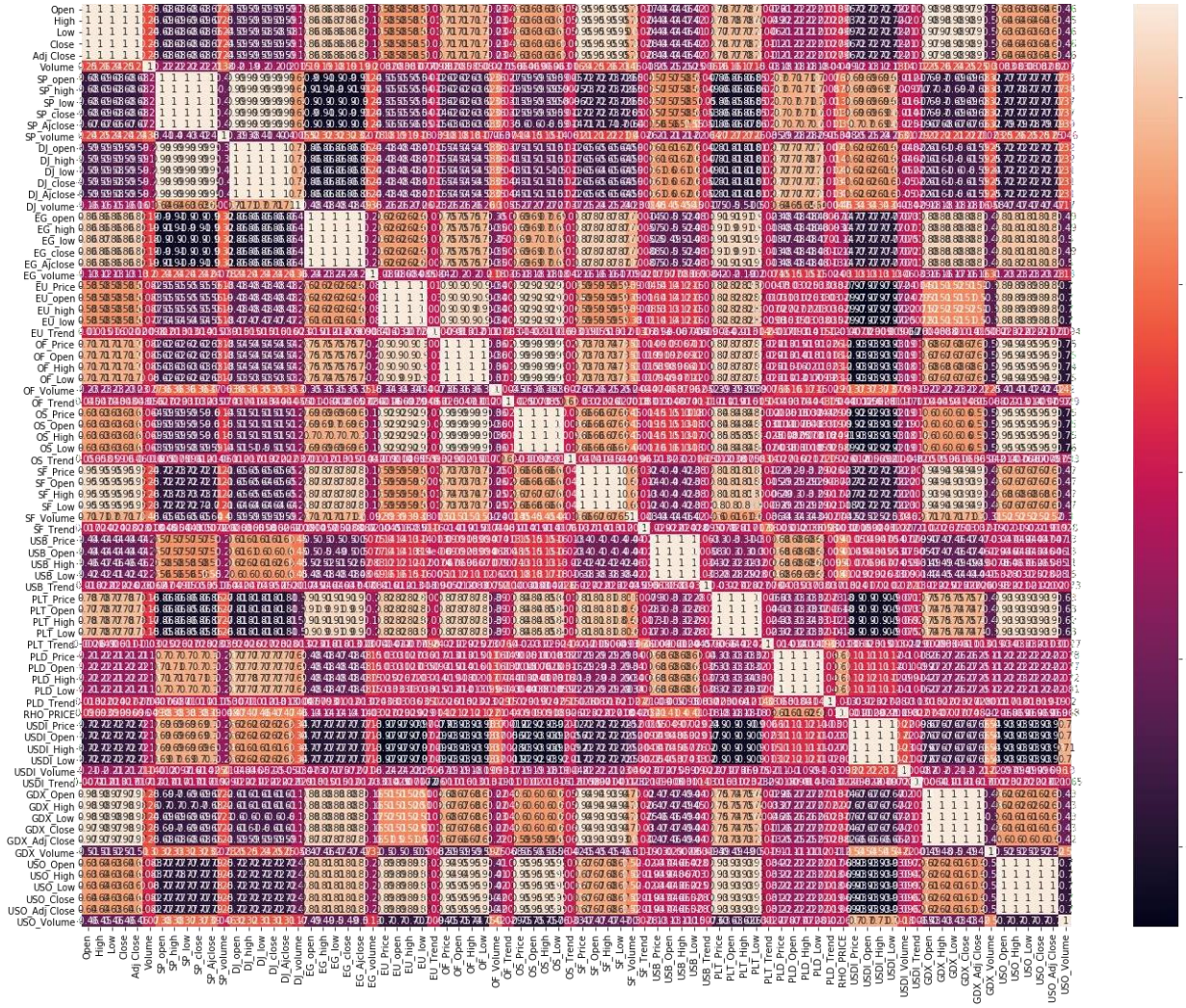

Fig 3.16 Correlational Analysis

Next we will calculate and plot correlation bar graph taking all the feature variables against target variable. Values in the range of -1 to +1 where negative value indicates that particular feature is negatively correlated with target variable and positive value indicates that feature is positively correlated with target variable and 0 value indicates no correlation exist between thatfeature and target variable.

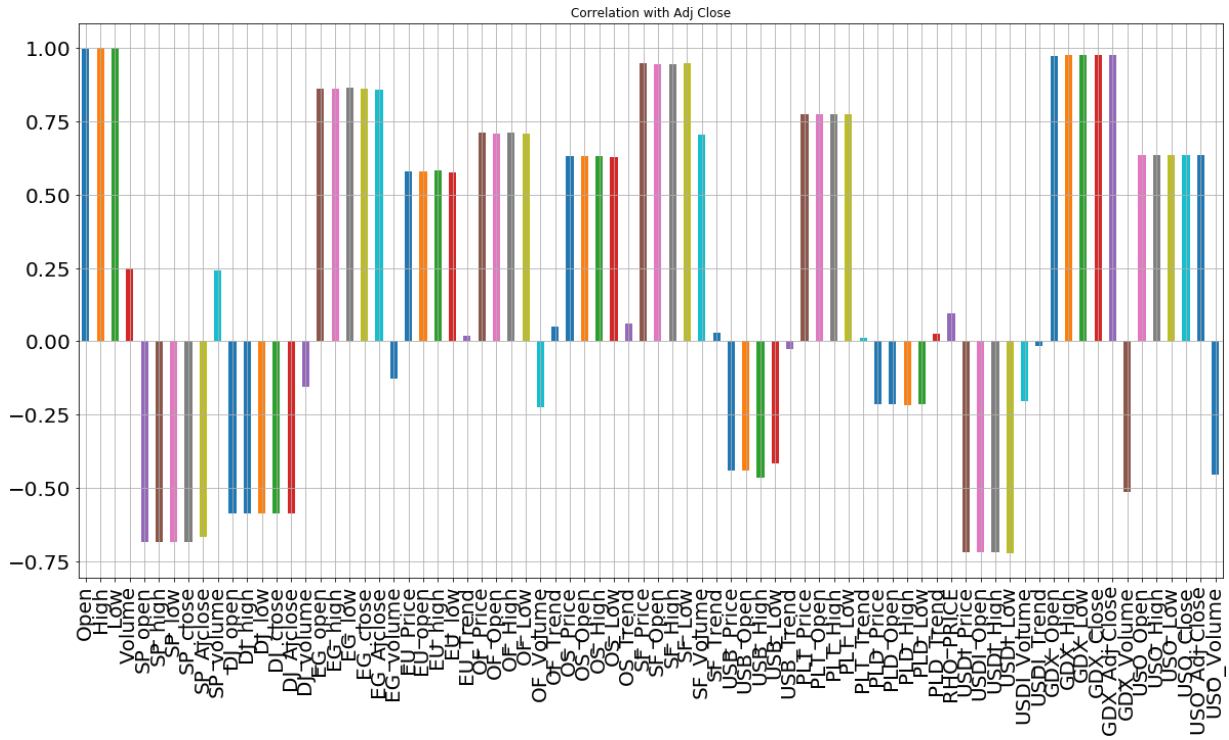

Fig 3.17 Correlation with Adj Close

Below image shows the summarized correlation between different features and target variable:

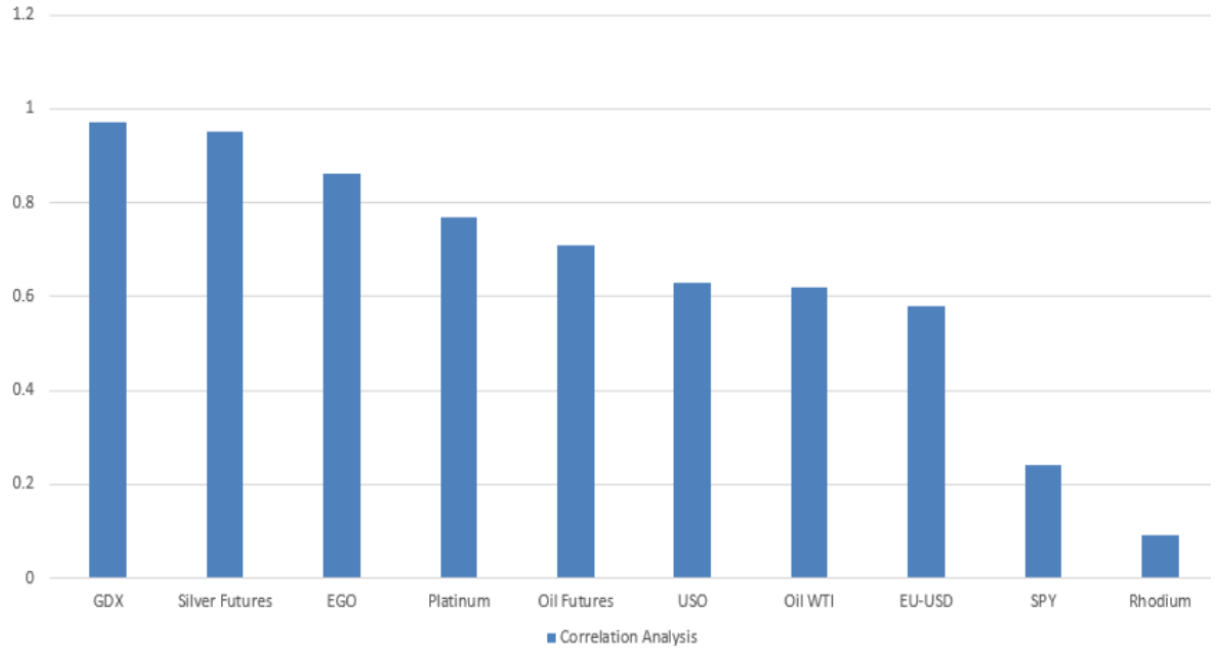

Fig 3.18 Summarized Correlation

CHAPTER 4

ALGORITHMS AND TECHNIQUES

4.1 ALGORITHMS

The goal of this project is to study time-series data and explore as many options as possible to accurately predict the Gold rates in future. We would use TimeSeriesSplit function in scikit-learn to split the data into training set and testing set, it could split the whole dataset into several packs and in each packs, the indices of testing set would be higher than training set. By doing this can prevent look ahead bias, which means the model would not use future data to train itself. I separate the last 90 days data as the validation dataset, to test the models by the data they have never seen. We have used different machine learning algorithms to see if they are able to accurately predict Gold prices.I have used 7 different machine learning algorithms
i.e. Support Vector Regressor (SVR), Random Forest Regressor, Lasso CV, Ridge CV, Bayesian Ridge, Gradient Boosting Regressor and Stochastic Gradient Descent (SGD) andthen we ensemble top three best performing algorithms and compare their performance with other algorithms.

- **Support Vector Regressor**

The model produced by Support Vector Regression depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction. I have used SVR with (kernel=’linear’)

- **Random Forest Regressor**

Random Forest is a supervised learning algorithm. It creates a forest and makes it somehow random. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random decision forests correct for decisiontrees habit of overfitting to their training set. I have used two parameters n_estimators=50 (default value =10), the number of trees in the forest and random_state=0, random_state is theseed used by the random number generator.

- **Lasso CV**

The Lasso is a linear model that estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent.

  - Performs L1 regularization, i.e. adds penalty equivalent to absolute value of the magnitude of coefficients.
  - Minimization objective = LS Obj + α * (sum of absolute value of coefficients).

Note that here 'LS Obj' refers to 'least squares objective', i.e. the linear regression objective without regularization. I have used three parameters n_alphas=1000, Number of alphas along the regularization path max_iter=3000, the maximum number of iterations and random_state=0. Here, the alpha parameter controls the degree of sparsity of the coefficients estimated.

- **Ridge CV**

Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares.

Here, $\alpha \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of α, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity. I have used the parameter gcv_mode='auto', flag indicating which strategy to use when performing Generalized Cross-Validation.

- **Bayesian Ridge**

Bayesian regression techniques can be used to include regularization parameters in the estimation procedure: the regularization parameter is not set in a hard sense but tuned to the data at hand. I have used default parameters in this algorithm.

- **Gradient Boosting Regressor**

Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. GradientBoostingRegressor supports a number of different loss functions for regression which can be specified via the argument loss; the default loss function for regression is least squares ('ls'). I have used following parameters: n_estimators=70, the number of boosting stages to perform. learning_rate=0.1,

learning rate shrinks the contribution of each tree by learning_rate, max_depth=4, the maximum depth limits the number of nodes in the tree, random_state=0, loss='ls', 'ls' refers to least squares regression.

- **Stochastic Gradient Descent**

SGD stands for Stochastic Gradient Descent: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). The regularizer is a penalty added to the loss function that shrinks model parameters towards the zero-vector using either the squared Euclidean norm L2 or the absolute norm L1 or a combination of both (Elastic Net). I have used following parameters: max_iter=1000, the maximum number of passes over the training data (aka epochs). tol=1e-3, the stopping criterion. If it is not None, the iterations will stop when (loss > previous_loss - tol), loss='squared_epsilon_insensitive', 'epsilon_insensitive' ignores errors less than epsilon and is linear past that; this is the loss function used in SVR. 'squared_epsilon_insensitive' is the same but becomes squared loss past a tolerance of epsilon. penalty='l1', the penalty (aka regularization term) to be used, alpha=0.1, Constant that multiplies the regularization term.

4.2 BENCHMARK MODEL

I would choose Decision Tree Regressor and use default arguments as the benchmark model. I would use the same data and features in my solution model detailed above. The result of the benchmark model on validation set is shown below:

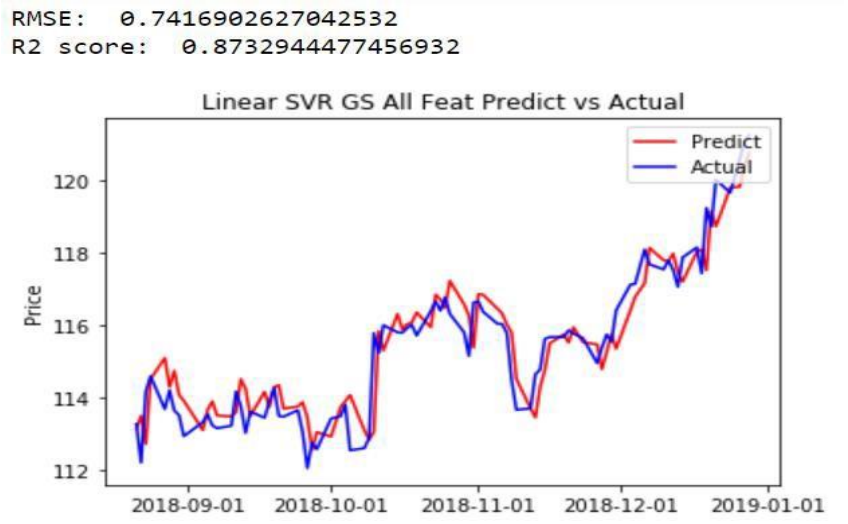

Fig 4.1 Benchmark Model

CHAPTER 5
IMPLEMENTATION

5.1 LANGUAGE USED

Python programming language

Machine learning is the kind of programming which gives computer the capability to automatically learn from data without being explicitly programmed. In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. Python is a perfect choice for the field of machine learning and data science. It is a minimalistic and intuitive language with a full-featured library line (also called frameworks) which significantly reduces the time required to get the results.

5.2 LIBRARIES USED

5.2.1 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high- level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions fromseveral other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is open- source software and has many contributors.

5.2.2 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

5.2.3 Seaborn

Seaborn is a library for statistical graphics. It builds on top of matplotlib and integrates closely with pandas data structures. Its plotting functions operate on

Data-frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

5.2.4 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits. Pyplot is a Matplotlib module which provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

5.2.5 Scikit Learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as ease of use, code quality, collaboration, documentation and performance. The library is focused on modeling data. It is not focused on loading, manipulating and summarizing data.

5.3 CODE SEGMENT

Importing Libraries

```
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestRegressor
from matplotlib.pyplot import figure
import seaborn as sns
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.dates as mdates
from sklearn import linear_model
from sklearn.model_selection import TimeSeriesSplit
from sklearn.svm import SVR
```

Fig 5.1 Importing Libraries

Result Validation

validate_result take a model and the model's name as input, it would use validation set to see the performance of model, and output the RMSE error and R2 score and also plot the results.

```
def validate_result(model, model_name):
    predicted = model.predict(validation_X)
    RSME_score = np.sqrt(mean_squared_error(validation_y, predicted))
    print('RMSE: ', RSME_score)

    R2_score = r2_score(validation_y, predicted)
    print('R2 score: ', R2_score)

    plt.plot(validation_y.index, predicted,'r', label='Predict')
    plt.plot(validation_y.index, validation_y,'b', label='Actual')
    plt.ylabel('Price')
    plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
    plt.gca().xaxis.set_major_locator(mdates.MonthLocator())
    plt.title(model_name + ' Predict vs Actual')
    plt.legend(loc='upper right')
    plt.show()
```

Fig 5.2 Validate result

Support Vector Regressor

The result of default parameter is not ideal, it didn't learn well how to predict price. So, I have tried to tune the hyper parameters using GridSearchCV to adjust the parameters to see whether we could improve the model performance. The parameters I have tuned using Gridsearch is shown below.

```
RMSE:  0.7416902627042532
R2 score:  0.8732944477456932
```

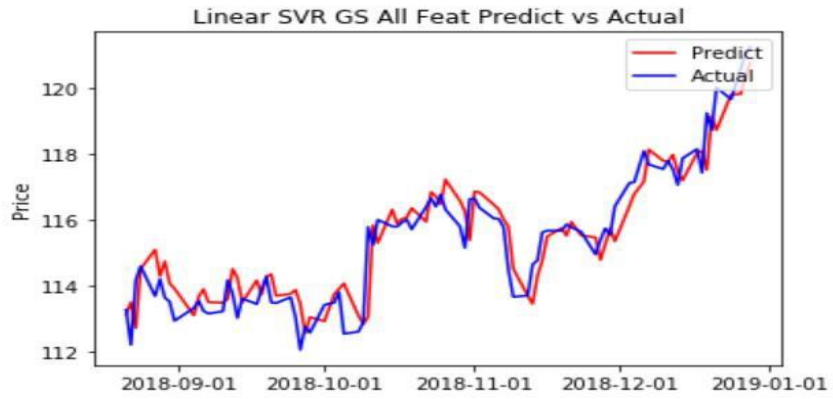

Fig 5.3 Linear SVR GS All Feat Predict vs Actual

Random Forest Regressor

After applying random forest regressor with n_estimators =50, we have achieved following result :

```
RMSE:  0.809652076705479
R2 score:  0.8490102869916533
```

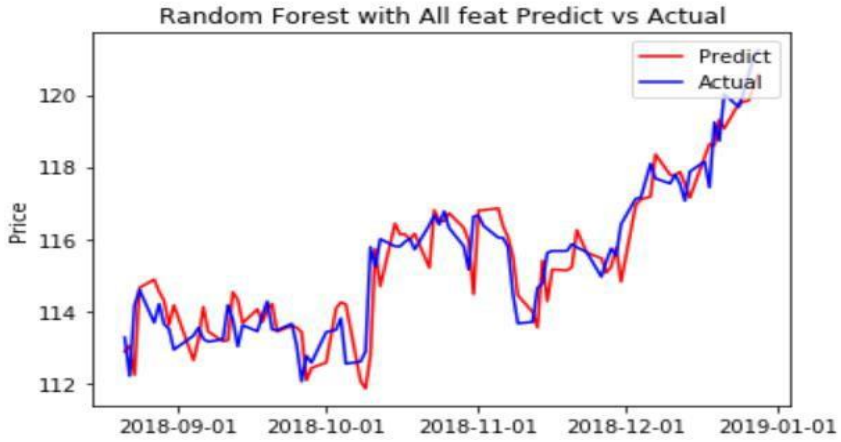

Fig 5.4 Random Forest All Feat Predict vs Actual

After applying random Forest regressor with default parameters we have tried to tune the hyper parameters using GridsearchCV to see if the performance would improve.Below is the result after applying GridsearchCV on Random forest.

```
RMSE:  0.8209975294625336
R2 score:  0.8447490766439336
```

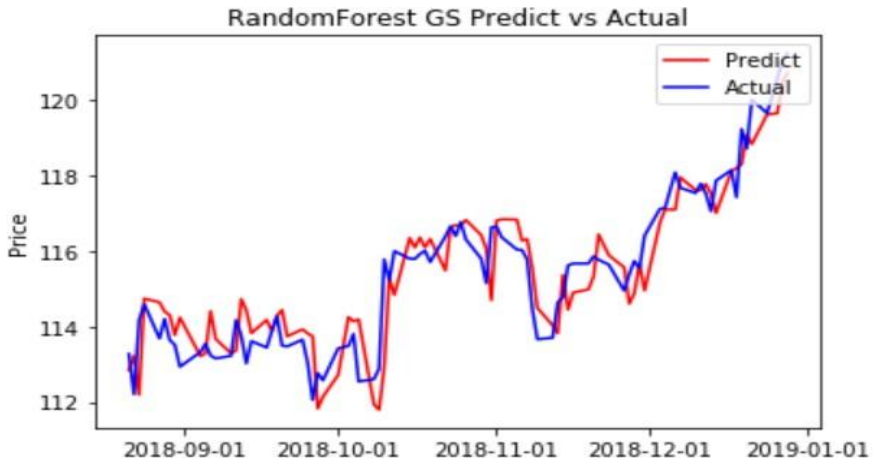

Fig 5.5 Random forest GS Predict vs Actual

Lasso CV

By default, Lasso CV performs Cross-Validation, so I have tried to tune some parameters i.e., n_alpha = 1000 perform better, and set the max_iter = 3000, because if the max_iter value is low, the model would reach the limit before convergence.

```
RMSE:  0.7159068365647187
R2 score:  0.8819506742756609
```

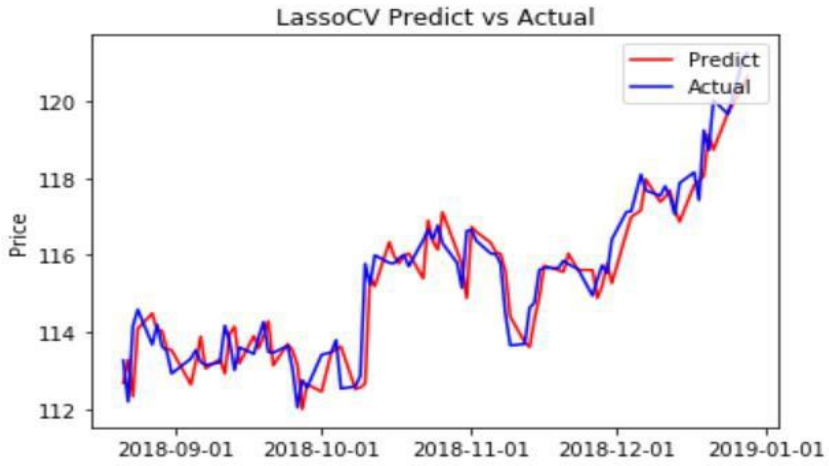

Fig 5.6 Lasso CV Predict vs Actual

Ridge CV

It performs generalized cross validation. I have used the parameter gcv_mode='auto', flag indicating which strategy to use when performing Generalized Cross-Validation.

```
RMSE:  0.7186272522528762
R2 score:  0.8810518047954405
```

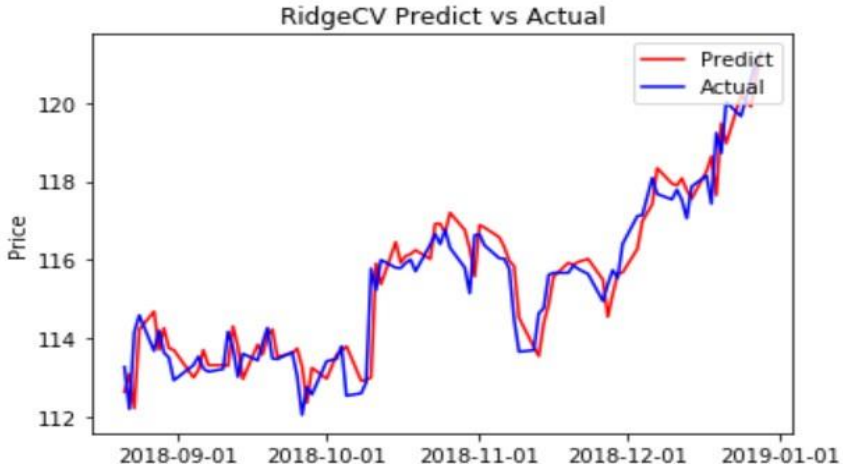

Fig 5.7 Ridge CV Predict vs Actual

Bayesian Ridge

Bayesian regression techniques can be used to include regularization parameters in the estimation procedure: the regularization parameter is not set in a hard sense but tuned to the data at hand. I have used default parameters in this algorithm.

```
RMSE:  0.7195639481910471
R2 score:  0.8807415162414403
```

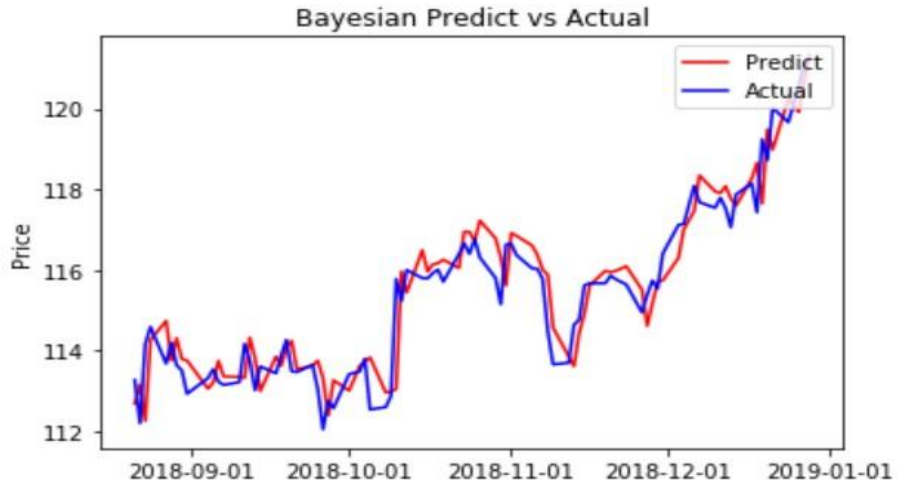

Fig 5.8 Bayesian Predict vs Actual

Gradient Boosting Regressor

Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. I have used following parameters: n_estimators=70, learning_rate=0.1, max_depth=4, random_state=0, loss='ls', 'ls' refers to least squares regression.

```
RMSE:  0.8094931831292773
R2 score:  0.8490695443986888
```

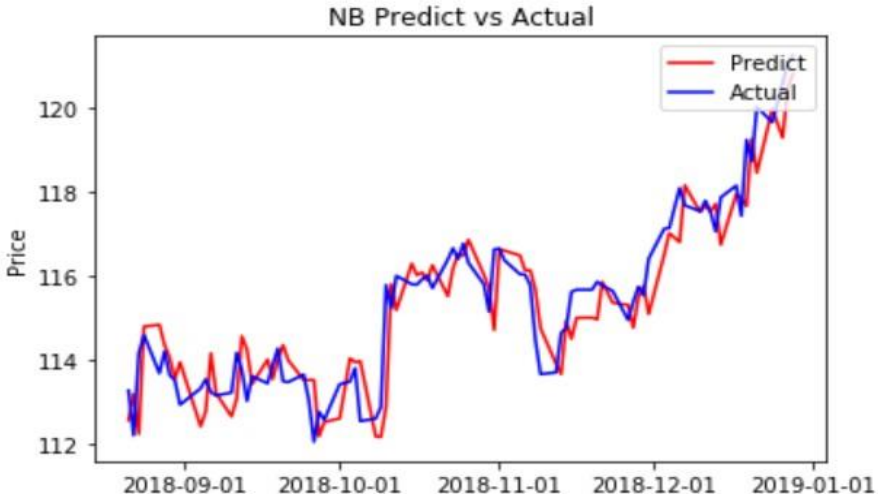

Fig 5.9 NB Predict vs Actual

SDG

SGD stands for Stochastic Gradient Descent: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). The regularizer is a penalty added to the loss function that shrinks model parameters towards the zero vector using either the squared euclidean norm L2 or the absolute norm L1 or a combination of both (Elastic Net). I have used following parameters : max_iter=1000, tol=1e-3, loss='squared_epsilon_insensitive', penalty='l1' and alpha=0.1

```
RMSE:  1.0294014616728233
R2 score:  0.7559268239699624
```

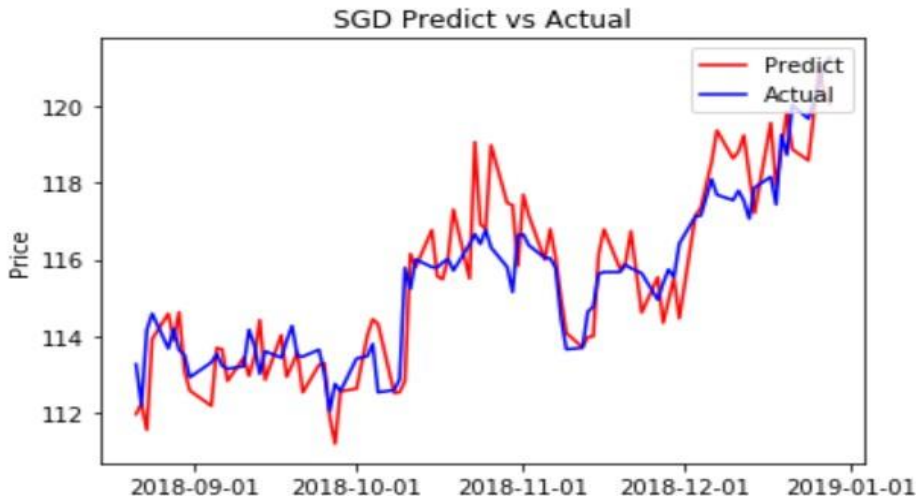

Fig 5.10 SGD Predict vs Actual

5.4 REVIEWS OF BENCHMARKS & SOLUTION MODELS

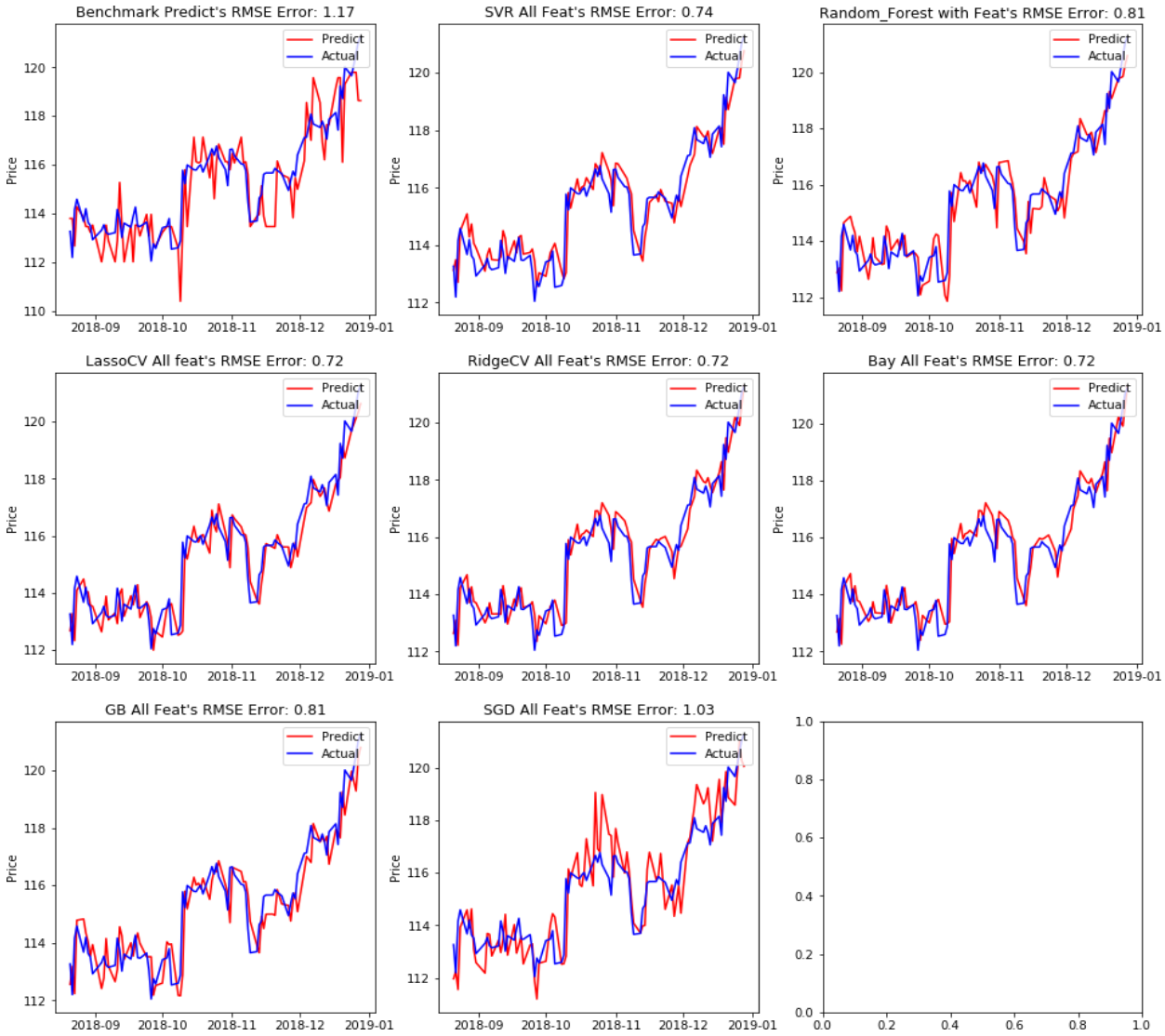

Fig 5.11 Review of Benchmark vs Solution Models

Comparison of RMSE Benchmarks and Solution Models

We can see that the RSME errors of models are close except SGD and benchmark model which is decision tree regressor, Benchmark model has highest RMSE error next SGD has second highest RMSE error whereas LassoCV, RidgeCV and Bayesian Ridge has nearly same performance having RMSE error in the range of 0.70 to 0.72.

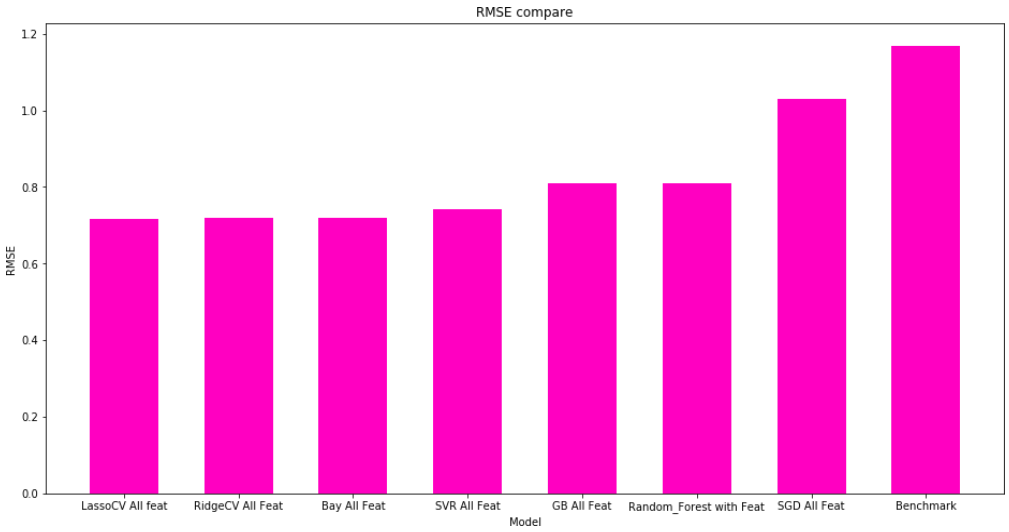

Fig 5.12 RMSE Benchmarks & Solution Models

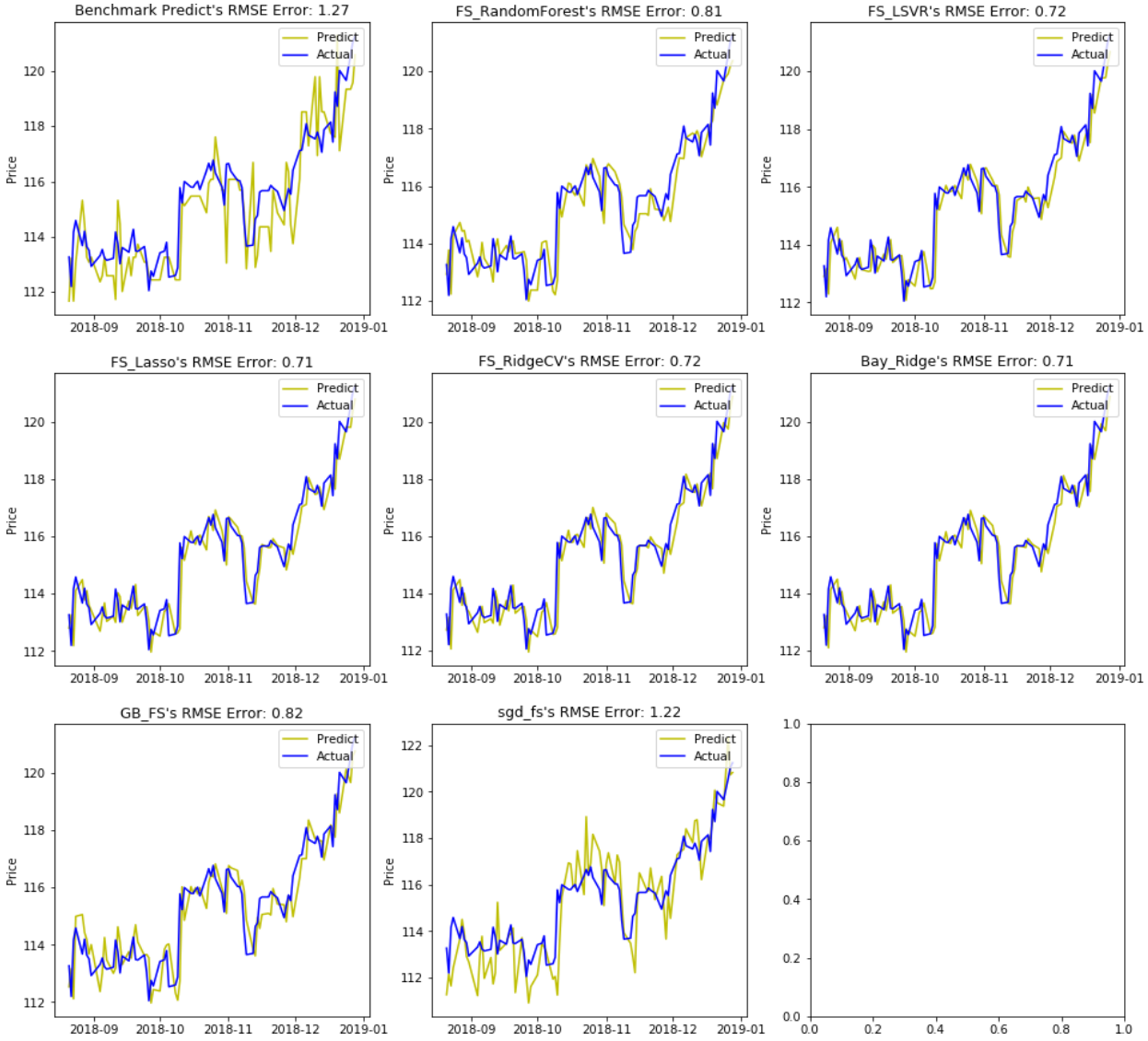

Fig 5.13 Review of RMSE of Features & Original

Comparison of RMSE of Feature selected and the Original Feature

Three Feature selected models performs better in RMSE error reduction & Feature selected Linear SVR is the best as it has RMSE of 0.716 & 0.741 in feature selected & all features model. Lasso cv and Bayesian Ridge performs slightly better and Ridge cv shows no improvement.

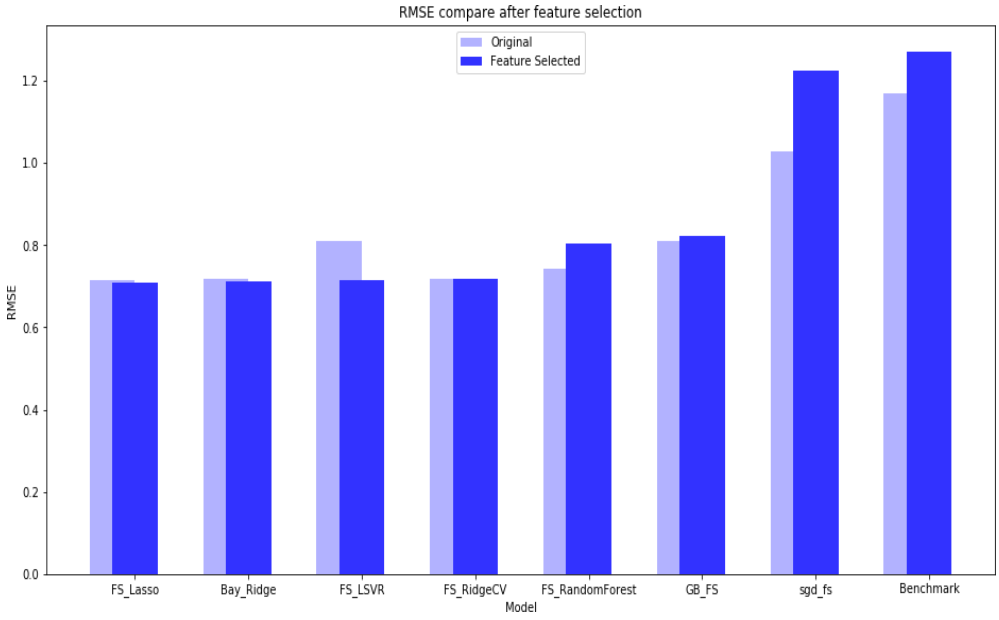

Fig 5.14 RMSE of Feature & Original Feature

CHAPTER 6

FUTURE ENHANCEMENT AND CONCLUSIONS

Free Form Visualization

I have already discussed all the important features of the datasets and their visualization in above sections. But in order to conclude my report I would choose ensemble model with original feature visualization. As I was very satisfied by seeing how close I have predicted the price with actual price and it has lowest Root Mean Square error of 0.699.

Ensemble Solution Model with Original features
RMSE: 0.699839121518396
R2 score: 0.8871901754597994

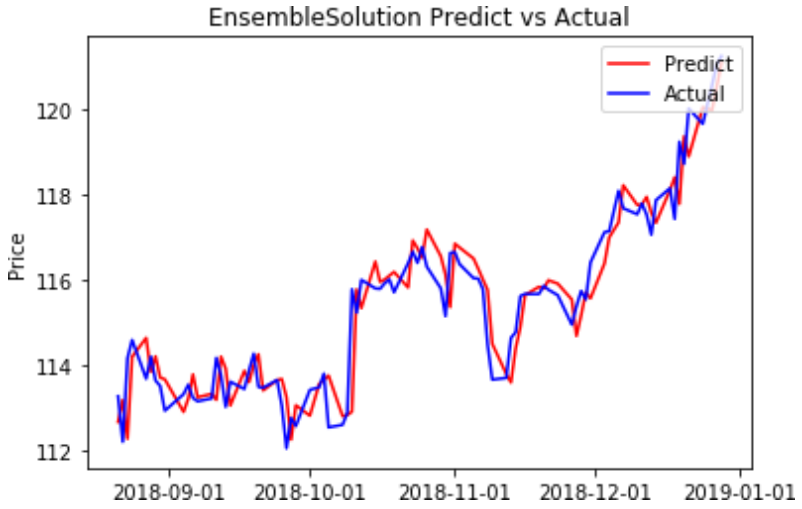

Fig 7.1 Free Form Visualization

Reflection

The process undertaken this project are as follows:

Set Up Infrastructure

- I Python Notebook
- Incorporate required libraries Sklearn, Numpy, Pandas, Matplotlib and Seaborn
- Git Project Organization

Dataset Preparation

- Incorporate data from different sources
- Process the data into pandas data-frame

- Normalize the data using Sklearn's MinMax Function
- Timeseries Split with n_splits=10

- Develop Benchmark Model
  - Set up basic decision tree regressor with default parameters as benchmark model.
- Develop Solution Model & improve using GridsearchCV
- Ensemble top three performing models
- Evaluate all the solution models and benchmark model & plot the result
- Refine the models using Feature selection
- Develop Benchmark, solution and ensemble models
- Evaluate and compare the results with original feature models
- Plot, analyze and describe the results for report.

It is almost practically impossible to get a model that can 100% predict the price without any error, there are too many factors on which gold prices depends I tried to capture as much factors as possible but still there are other factors such as CPI, political factors, disasters, financial breakdown that can affect the market. But this solution model performed really well in this project, which will help the trader to make better decision.

Reflection Future Scope of the Project

- There are still many technical indicators and feature variables which we have not included in our project, maybe there are some other indicators which we haven't explored would perform better.
- There are lots of Machine Learning algorithms which we haven't tried and may be neural network or LSTM would perform better than our solution.
- And last but not the least we have used data of around 14 years if we would increase the data, I think the performance of our solution models may be improved.

REFERENCES

- Udacity
- Kaggle
- Github
- Coursera
- Youtube
- Medium
- Wikipedia, the free encyclopedia