**Gold Rates Prediction Using Machine Learning**

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.

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 **SPDR Gold Trust (GLD) Exchange Traded Fund** data downloaded from [https://finance.yahoo.com](https://finance.yahoo.com/) in the date range of **18/11/2004 to 01/01/2019**. 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.

We have ensemble top three performing models to predict the Adjusted close price of Gold.

**Project Workflow:**

* **Importing Dataset**
* **Basic Data exploration**
* **Effect of index price on gold rates**
* **Calculating technical indicators**
* **Scaling or normalizing using minmaxscaler**
* **Time Series Spilt**
* **Creating machine learning model**
* **HyperParameter Tuning**

**Problem Highlights:**

\*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 Timeseries Data.
* How to perform statistical analysis of Timeseries Data.
* How to collect and preprocess given data.
* How to ensemble different machine learning models and analyze model's performance.
* How to optimise Machine Learning models to increase accuracy and reduction in error.

**Installations:**

This project requires Python 3.x and the following Python libraries should be installed to get the project started:

* [Numpy](http://www.numpy.org/)
* [Pandas](http://pandas.pydata.org/)
* [matplotlib](https://matplotlib.org/)
* [scikit-learn](https://scikit-learn.org/stable/)
* [Seaborn](https://seaborn.pydata.org/)

**About Data:**

The data file which I have used for this project is FINAL\_USO.csv. 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 target variables which is the value we want to predict.

**Evaluation Metrics:**

I would use **Root Mean Square Error(RMSE)** 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 .

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.