GOLD PRICE PREDICTION USING SUPERVISED LEARNING.

Author: Meron Habtemichael

Course: Machine Learning

TABLE OF CONTENTS

1. INTRODUCTION

1.1. CONTEXT

Investments and Savings form an integral part of everyone's life where the task is getting more complicated over time. One of these common investment vehicles is Gold. It is mostly known as a store of value. It can be used to hedge against inflation as well. When some thoughts of investment cross the mind, people often find it challenging to do the analysis manually and predict how well some investments would perform.

The rising value of gold, its volatilities, increasing use cases, and fall in prices of other markets like capital markets and real estate markets have attracted more investors towards gold. This means there is a lot of statistical analysis and prediction to do. Here, Machine Learning falls into a perfect application. This project uses supervised learning models to predict the daily price of gold.

2. METHODOLOGY

2.1. DATASET

Most of the project's implementation effort was in data preparation and understanding. It is the most important and time-consuming task in a Machine Learning project lifecycle. Therefore, finding a quality dataset is a fundamental requirement to build any real-world AI application. Machine learning models heavily depend on the foundation of high-quality training data. Even the most powerful algorithms can be unusable with inappropriate or unprepared datasets. Over the past decade, we have had the privilege of having many corporations of open-source datasets that have motivated the AI community.

In this project, I used Kaggle's free available dataset to train a Machine learning model. One can find many similar websites/pages to download from. The larger the data set, the more time it takes for the model to process and predict. So, for this project, I have taken a medium-sized dataset, with instances of about 2500 samples.

2.2. DATA VISUALIZATION

Once the dataset is downloaded in CSV format, we will need the Pandas module to analyze, visualize, preprocess, and prepare it to be ready to be used for our Machine Learning models.

To prepare for analysis and model development, let us investigate our dataset.

2.2.1 Description and shape

The dataset originally includes 5 features and a target variable. Our mission is to predict the closing price of a Gold ETF at the end of a Trading Day. Given these features and the data type of each feature, the dataset was found to be appropriate. Let us see the following figure to see the datatype and description of the dataset.

```
df.shape    # Lets see and determine the shape of the dataset
```

```
(2547, 6)
```

```
df.isnull().sum()    # Check if we have missing values
```

```
Date           0
Close/Last     0
Volume        39
Open           0
High           0
Low            0
dtype: int64
```

```
df.dropna(inplace = True, axis = 0)     # Drop the null entries from the dataset
```

Fig 2.1 Description and Shape

From the figure above we can see we have missing values and then illuminated all the NULL values from the dataset. This is crucial. So, we have 2547 – 39 samples of time between 2022-10-28 and 2012-10-31. Each sample represents the Trading History of a day. The next step was to replace the default index with the 'Date' column. This makes the plotting much easier. Let us for example see our target variable (Closing price) plot in the following figure.

Fig 2.2 Closing price plot

From the above plot graph, we can see that Gold Price have been bullish in 2020 and 2021 mostly bearish from 2013 all the way until 2016. But for further visualization, let us quickly look at how it has performed in the last three months.

```
df.iloc[-90:].describe().astype(int)
```

	Close/Last	Volume	Open	High	Low
count	90	90	90	90	90
mean	1669	138715	1671	1679	1659
std	49	41561	48	47	49
min	1572	40829	1564	1582	1554
25%	1649	112521	1655	1663	1641
50%	1674	135184	1674	1684	1663
75%	1711	163687	1712	1718	1700
max	1751	241179	1751	1755	1746

Fig 2.3 Performance of the ETF in the past 90 days (about 3 months).

From the above figure, we can observe that the average price is 1669 which is more than what it is currently trading at 1648. The maximum volume traded in a single day was 241179, which is a substantial number compared to the most recent Trading Day.

Insights like this can help us to make better financial decisions just from visualization of the data. One can do many more things with this data like forecasting and alert notification. Machine Learning has been heavily used in this sector of finance since it first became popular. Let us get into Feature engineering our dataset so that then we can train our predicting models.

2.3. Feature Engineering

Feature engineering is more than selecting the appropriate features and transforming them. It is about preparing the dataset to be compatible with the algorithm and improving performance. In this project, I started first observing the correlation between each variable. Let us see the following figure containing a heatmap to see the existing correlation on the default given dataset.

```
he = df.drop(columns=['Date'], axis="1")
corr = he.corr()
corr
```

	Close/Last	Volume	Open	High	Low
Close/Last	1.000000	0.063957	0.998689	0.999336	0.999423
Volume	0.063957	1.000000	0.068945	0.072961	0.058990
Open	0.998689	0.068945	1.000000	0.999273	0.999144
High	0.999336	0.072961	0.999273	1.000000	0.998921
Low	0.999423	0.058990	0.999144	0.998921	1.000000

```
sns.heatmap(corr, annot=True)
```

```
<AxesSubplot:>
```

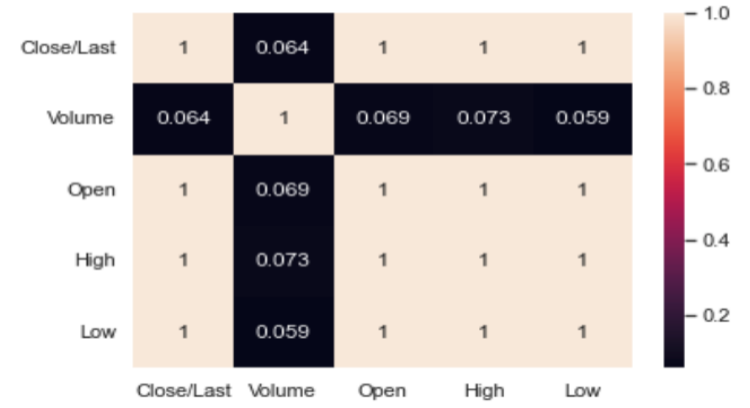

Fig 2.4 Correlation between variables

From the above figure, we can see that the data has a high correlation between each variable. This helps a lot in terms of performance and measuring the R**2 value (Which will be discussed in the results section.)

In this instance, there are more technical indicators that we can calculate and create as a feature, which can affect our model's performance for the better. This is known as **Feature Creation**. I created two features which are:

1. Day % Change: this is the price change percentage of each day with respect to the previous day. This is a quite common forecasting indicator. It is a good indicator of volatility. I could easily do it using Pandas. Look at the following figure.

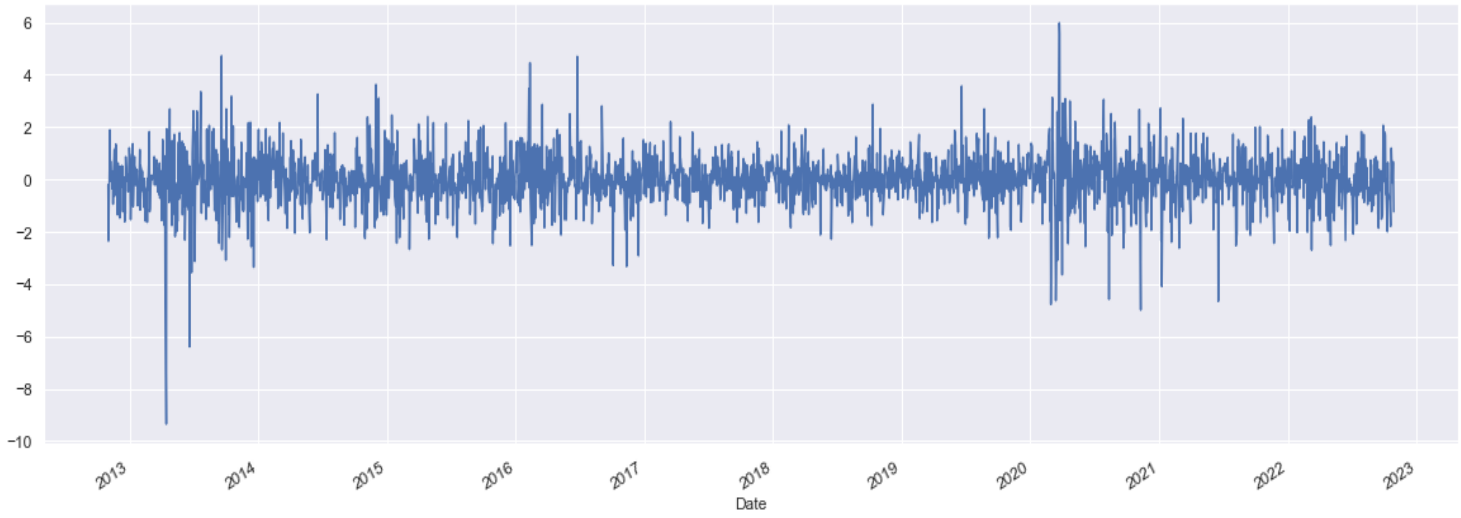

Fig 2.5 Daily percentage change plot

2. Moving Average (20): This is also a common technical analysis indicator that is calculated based on different magnitudes (for this case. 20). This moving average takes the last 20 day (about 3 weeks)'s closing price and averages them. Look at the following figure.

Fig 2.6 Moving Average (20) plot

After some additional few steps and dropping some empty values, we come to the last part, which is preparing the dataset so that we can train our models.

2.3.2. Encoding and Scaling/Standardizing features

The dataset does not include categorical features. The only feature we would need to encode is the 'Date' column. For this procedure, I have used Ordinal Encoder from Scikit-Learn to do the actual encoding. By default, the encoder does the assignment of numerical value in increasing order so this can easily be done in one step.

Standardizing the data values is one of the important practices done to increase the performance of a machine learning model. Standard-Scaler comes into play when the input dataset's characteristics differ between their ranges or when measured in different units of measure. So, in this case, I have applied this technique and influenced the variance. Finally, I divided the data set into a set of training and tests.

2.4. Training 3 ML models

After carefully preparing the dataset, and analyzing the underlying patterns and feature importance, the next step is obviously choosing a model and fitting the training data. I tried 7 models (which can be found in the source code) and they all worked very well. I closely analyzed

and fine-tuned 3 models i.e., linear-Regression, Random-Forest-Regressor and Gradient-Boosting-Regressor.

3. RESULTS

Here is the performance of models in the following table.

Model/Metrices	Mean absolute error	Mean squared error	R2_score
Linear Regression	2.970	17.75	0.99
Random-Forest-Regressor	4.14	37.76	0.99
Support-Vector-Regressor	58.52	8118.27	0.87

We can see that we have achieved an impressive performance.

4. DISCUSSION

Stock Price prediction using supervised learning helps to discover mistakes and changes in the price of a stock and the entire idea is to gain significant insights. Otherwise, manually staring at the [i]charts and trying to do technical analysis is a tiring and time-consuming task. Machine Learning models can help predict the stock price in a fast and more accurate way.

However, it is unreliable to fully see Gold's past price performance and predict future prices because ETFs are influenced by many factors. This study empathizes that this process is an extension of financial analysis, not a major decision-making indicator that individuals can depend on and make an income.

5. CONCLUSION

This study was conducted to understand the relationship between the gold price and its commonly associated features achieved from any exchange centers like NASDAQ or Yahoo finance. This specific dataset was taken from Kaggle and was just enough to train a few ML models. With some knowledge of finance, one can carefully create features (Indicators) and visualize the dataset using python tools like Pandas, Seaborn, and Matplotlib to gain some insights into financial markets. Machine Learning has been heavily used in the sector of Finance and dramatically changed things for the better.

6. REFERENCES

1. Dataset
2. Project on GitHub
3. Article on the internet, for General Understanding