5BIS/SE-Service Oriented Architecture:

Concept & Technology

Project Category

Performance Analysis of Gold Price

[5SE-Thanata-934]

Submitted Date: ……

Document History

Date	Version	By	Remarks
	1.0	**5SE-Thanata-934**	**Version1**

Contents at a Glance

Table of Contents

Acknowledgements

I have taken all the effort on this project with python language using flask framework. However, it would not be possible without the kind support and help of many individuals.

First and foremost, I would like to thank all the teachers of Faculty of Information Science Department (FIS) for their patience and supports on my project. Furthermore, I like to express my special thanks of gratitude to my teacher, Dr. Zu Zu Aung, Faculty of Information Science, Head of Software Engineering Major class, for her encouragement and endless support through the process on this project

Also, I appreciate Dr. Kyawt Kyawt San, Daw Lay Myat Myat Thein and Dr. Myint Myint Thein for their comments and guidelines for the improvement of my project where I came to know about so many things.

Last but not least, I thank to my friends, Myat Min Maung, Ei Ei Moe Pwint, Aye Thinzar Myo who helped me a lot in finalizing this project within the limited time frame. I also give my deepest love and appreciation to my parents for providing anything I need for this project as well as inspiration and motivation from them.

Abstract

The system is developed by using python language with flask framework. This system aims for the prediction of gold price especially to the merchants or investors for their business. The system estimates the price of the gold based on user's input which is the date.

Linear regression and Random Forest Regressor are one of the supervised machine learning algorithms, used in this system to predict the price of the gold. Before constructing the model, I did data cleaning, outlier detection, data visualization on the dataset. Then, I selected dependent variable Y in which it is the price of the gold where the value is computed based on the independent variables, day, month and year which are extracted from the provided dataset to construct the Linear Regression and Random Forest Models to estimate the price of the gold.

1.Introduction

1.1 Introduction

Machine Learning gives enterprises a view of trends in customer behaviour and business operational patterns, as well as supports the development of new products. Many of today's leading companies, make machine learning a central part of their operations. 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 reflecting the financial strength of the country. Forecasting rise and fall in the daily gold rates can help investors to decide when to buy or sell the commodity.

1.2 Objectives of the System

The objectives of the system are:

- To output the predicted price of gold with different models
- To understand how the linear regression and random forest models work and use them in Machine Learning processes.
- To evaluate which model is better for the system
- To get the knowledge of constructing a machine learning web application

1.3 Problem Statement

Predicting the price of gold is not easy as we think. In present time, the price movement of gold arise from a combination of many different factors such as demand, gold and fiat currencies, gold as a safe, supply of available gold haven asset. However, the price of the gold is predicted by using the Linear Regression and Random Forest Models in this system. Before predicting, the gold price data from the year 2000 to 2022 is applied with data visualization techniques. The extracted day, month and year columns are trained with Linear Regression and Random Forest Models to foreknow estimated price. The User Interface (UI) is built according to the extracted attributes and the user have to enter the valid inputs to achieve the required predicted price.

1.4 Requirement Analysis

Functional Requirements

Functional requirements are the product features or its functions that must be designed directly for the users and their convenience. They define the functionality of the software, which the software engineers have to develop so that the users could easily perform their tasks up to the business requirements.

Input Parameter

The user have to give the valid inputs of the date which are day, month and year.

Model

The system will train the dataset and construct the Linear Regression and Random Forest models to predict the gold as the final output.

Non-Functional Requirements

Non-functional requirements are the quality attributes, certain design or realization constraints or external interface that directly relate to the product. They act as an additional description of the functions of the product under development, which are important for stakeholders (users or developers).

Reliability/Availability

The system must be available to function 24/7 so that the users can rely on the system to predict the gold price for the business.

Performance

The system's performance is important as the user need to have the predicted price as quick as possible

Security

The system should have the tight security measures to protect the data in the dataset and the users.

Error Prevention

To prevent the error that can occurred within the system, the system will take precaution and error indicating signs to minimize the human error and system errors.

Recognition

The system having consistent user interface and matching with the real world will give the system interface recognition for the users.

Flexibility

Flexible user interface of a system is related with how the experienced and inexperience users can handle the interaction with the system.

Aesthetic and Minimalist Design

As complex design tends to produce complex user interface, the system will have a aesthetic and Minimalist Design interface.

Help and Documentation

User Interface can be made easier with the help of documentation and the users will be able to use the interface easier with a help or documentation of the system.

2. Project Specification

2.1 Project Charter

Project Name : Performance Analysis of Gold Price

Date : 20th Aug 2022

Project Goal	: To provide merchants or investors who are making the gold business to know the best predicted gold price for the day based on the input they give to the system.

Success Factors	: Estimated gold price based on user's inputs with use of Linear Regression and Random Forest Algorithm.

Scope	: day, month, year attributes of user's input and output final predicted price.

Schedule	: Requirement analysis	- 6 days
	Planning Phase	- 5 days
	Implementation Phase	- 15 days
	Execution Phase	- 3 days
	Testing	- 3 days
	Closing	- 1day
	Total	- about 1 month

Stakeholder	: Investors who are going to buy gold for business.

2.2 Data Preprocessing

Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process.

1. Importing Required Libraries.

Firstly, we need to load the required libraries for the system.

Importing Libaries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
# For models
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error
from math import sqrt
```

2. Loading Data

After importing the libraries, load the dataset which is in the csv format

Load the Data and Some Minor Cleaning

Importing the Data into a Pandas DataFrame for further analysis

```
df = pd.read_csv('dataset/Gold_Price_Data_Final_Final.csv')
```

3. Display the first five rows of the data using head() function

First 5 entries of the data ¶

```
df.head()
```

	Price	Year	Month	Day
0	281.0	2000	1	4
1	283.2	2000	1	5
2	281.4	2000	1	6
3	281.9	2000	1	7
4	281.7	2000	1	10

4. Display the last five rows of the data using tail() function

```
df.tail()
```

	Price	Year	Month	Day
5701	1970.5	2022	4	13
5702	1981.6	2022	4	14
5703	1978.5	2022	4	18
5704	1981.2	2022	4	19
5705	1952.1	2022	4	20

5. Check the null values from the dataset

Handling Null Values

```
df.isnull().any()
```

```
Price    False
Year     False
Month    False
Day      False
dtype: bool
```

6.Finding and removing outliers

Finding and Removing Outliers

```
from scipy import stats
```

```
z=np.abs(stats.zscore(df))
z
```

	Price	Year	Month	Day
0	1.474805	1.677273	1.584590	1.335567
1	1.470594	1.677273	1.584590	1.221453
2	1.474040	1.677273	1.584590	1.107339
3	1.473082	1.677273	1.584590	0.993225
4	1.473465	1.677273	1.584590	0.650884
...	...	...	...	...
5701	1.759744	1.735506	0.714194	0.308543
5702	1.780995	1.735506	0.714194	0.194429
5703	1.775060	1.735506	0.714194	0.262026
5704	1.780229	1.735506	0.714194	0.376139
5705	1.724517	1.735506	0.714194	0.490253

5706 rows × 4 columns

```
threshold=3
np.where(z>threshold)
```

```
(array([], dtype=int64), array([], dtype=int64))
```

```
df_no_outliers=df[(z<=3).all(axis=1)]
df_no_outliers
```

	Price	Year	Month	Day
0	281.0	2000	1	4
1	283.2	2000	1	5
2	281.4	2000	1	6
3	281.9	2000	1	7
4	281.7	2000	1	10
...	...	...	...	...
5701	1970.5	2022	4	13
5702	1981.6	2022	4	14
5703	1978.5	2022	4	18
5704	1981.2	2022	4	19
5705	1952.1	2022	4	20

5706 rows × 4 columns

```
df_no_outliers.shape
```

```
(5706, 4)
```

2.3 Model Building

After preprocessing, then we train the model with our clean data.

1. Train Linear Regression Model

Linear Regression

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error
```

```
linear_R = LinearRegression()
history = linear_R.fit(X_train, Y_train)
Y_pred_lr= linear_R.predict(X_test)
Y_pred_lr
```

```
array([ 253.19156482, 1649.35667232, 1352.53753174, ..., 1811.97798591,
       1588.94174428,  544.88402423])
```

2. Random Forest Model

Random Forest

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error,mean_squared_error
```

```
Random_F = RandomForestRegressor()
Random_F.fit(X_train, Y_train)
Y_pred_rf= Random_F.predict(X_test)
Y_pred_rf
```

```
array([ 279.8  , 1340.718, 1279.719, ..., 1748.567, 1272.67 ,  393.671])
```

3. Background Theory

3.1 Machine Learning

Machine Learning (ML) is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence (AI) that enables systems to learn and improve from experience without being explicitly programmed. It is based on the idea that systems can learn from data, identity patterns and make decisions with minimal human intervention. Machine Learning has proven to be one of the most game-changing technological advancements of the past decade. In the increasingly competitive corporate world, ML is enabling companies to fast-track digital transformation and move into an age of automation.

3.2 Different types of Machine Learning Algorithms

The focus of the field of machine learning is “learning”, there are many types that may encounter. The three main types of machine learning are supervised learning, unsupervised learning, and reinforcement learning. As with any method, there are different ways to train machine learning algorithms, each with their own advantages and disadvantages.

Supervised Learning

It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. Even though the data needs to be labeled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process.

Unsupervised Learning

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

Reinforcement Learning

Reinforcement learning is the training of machine learning models to make a sequence of decisions. It learns to achieve a goal in an uncertain, potentially complex environment. It is about learning the optimal behavior in an environment to obtain maximum reward. This optimal behavior is learned through interactions with the environment and observations of how it responds.

3.3 Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models are different based on the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. Linear regression is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable (Y). The variable you are using to predict the other variable's value is called the independent variable (X). This regression technique finds out a linear relationship between X (input) and Y (output). Mathematically, linear regression is represented as

$$\mathbf{y = a_0 + a_1 x + \varepsilon}$$

where :
y = Dependent Variable
x = Independent Variable
$\mathbf{a_0}$ = Intercept of the line
$\mathbf{a_1}$ = Linear Regression Coefficient
ε = Random Error

A linear line shows the relationship between the dependent and independent variables are called a regression line. That line presents two kinds of relationships which are Positive Linear Relationship and Negative Linear Relationship. When the dependent variable on the Y-axis increases and the independent variable on the X-axis increases, it is said to be Positive Linear Relationship. On the other hand, Negative Linear relationship is defined as the decreases in the dependent variable on the Y-axis and increases in independent variable on X-axis.

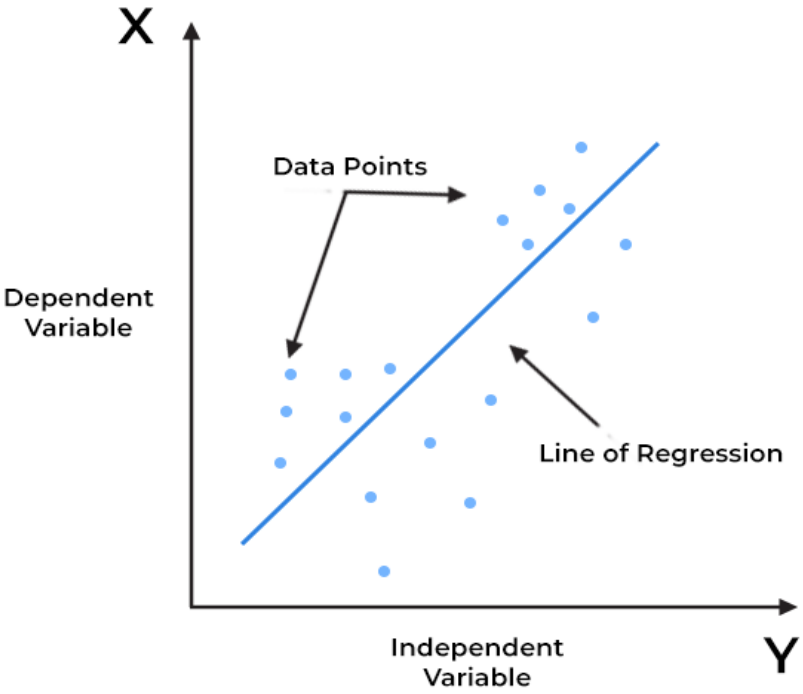


Figure 1 – Linear Regression

There are two main types of linear regression: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is used to estimate the relationship between two quantitative variables which are dependent and independent. It has one independent variable or predictor value X. Multiple linear regression refers to a statistical technique that uses two or more independent variables to predict the outcome of a dependent variable. It enables analysts to determine the variation of the model and the relative contribution of each independent variable in the total variance.

Finding the best fit line:

When working with linear regression, the main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so it is needed to calculate the best values for a_0 and a_1 to find the best fit line, and cost function is one of the solutions.

Cost function

The different values for weights or coefficient of lines (a0, a1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.

We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**.
For Linear Regression, the **Mean Squared Error (MSE)** cost function is used, which is the average of squared error occurred between the predicted values and actual values.

3.4 Random Forest Algorithm

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables (numerical value) as in the case of regression and categorical variables (non-numeric groups or categories) as in the case of classification. It performs better results for classification problems.

Working of Random Forest Algorithm

Random forest uses one of the ensemble techniques. Ensemble means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.
Ensemble uses two types of methods:

- Bagging - It creates a different training subset from sample training data with replacement & the final output is based on majority voting.
- Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy.

Among these two, Random Forest works on the Bagging principle.

Bagging

Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

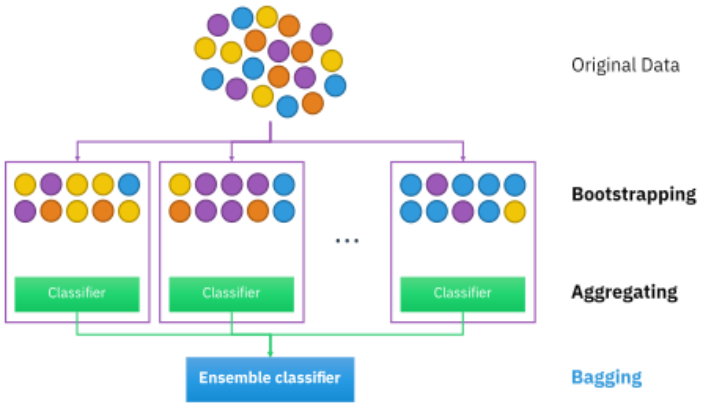


Figure 2 – Bagging

Steps involved in random forest algorithm

- Step 1: In Random forest n number of random records are taken from the data set having k number of records.
- Step 2: Individual decision trees are constructed for each sample.
- Step 3: Each decision tree will generate an output.
- Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

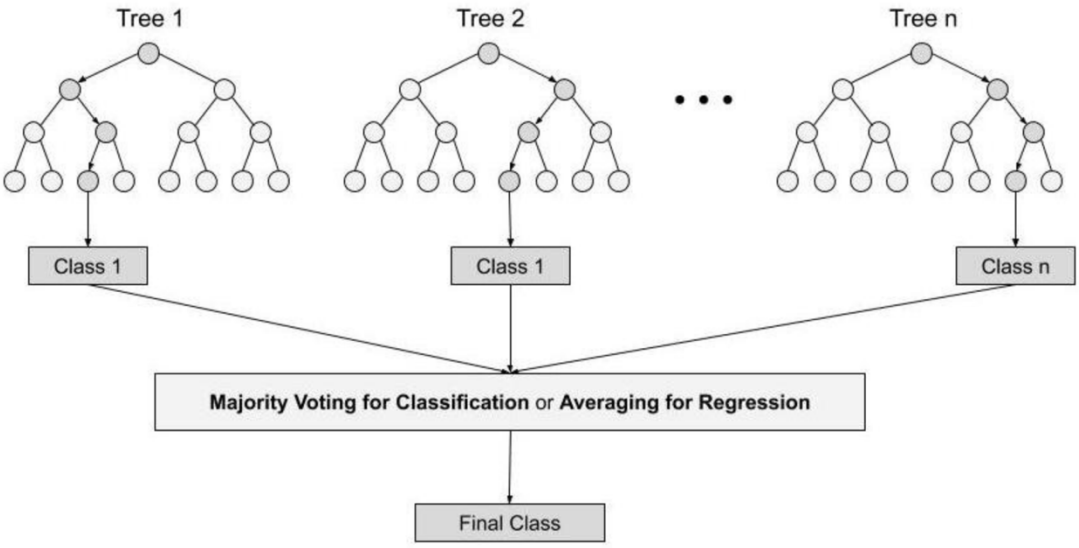


Figure 3 – Random Forest Algorithm

Important Features of Random Forest

1. **Diversity** - Not all attributes/variables/features are considered while making an individual tree, each tree is different.
2. **Immune to the curse of dimensionality** - Since each tree does not consider all the features, the feature space is reduced.
3. **Parallelization** - Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
4. **Train-Test split** - In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
5. **Stability** - Stability arises because the result is based on majority voting/ averaging.

Difference Between Decision Tree & Random Forest

Random forest is a collection of decision trees and still, there are a lot of differences in their behavior. Decision trees normally suffer from the problem of overfitting if it's allowed to grow without any control. A single decision tress is faster in computation. When a data set with features is taken as input by a decision tree it will formulate some set of rules to do prediction. On the other hand, Random Forest are created from subsets of data and the final output is based on average or majority ranking and hence the problem of overfitting is taken care of. It is comparatively slower. Random forest randomly selects

observations, builds a decision tree and the average result is taken and it doesn't use any set of formulas. Thus, random forest are much more better than decision trees only if the trees are diverse and acceptable.

Important Hyperparameters

Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster.

Following hyperparameters increases the predictive power of the random forest:

1. n_estimators– number of trees the algorithm builds before averaging the predictions.
2. max_features– maximum number of features random forest considers splitting a node.
3. mini_sample_leaf– determines the minimum number of leaves required to split an internal node.

Following hyperparameters increases the speed of the random forest:

1. n_jobs– it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor but if the value is -1 there is no limit.
2. random_state– controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and if it has been given the same hyperparameters and the same training data.
3. oob_score – OOB means out of the bag. It is a random forest cross-validation method. In this one-third of the sample is not used to train the data instead used to evaluate its performance. These samples are called out of bag samples.

4. Design

4.1 System Flow Diagram

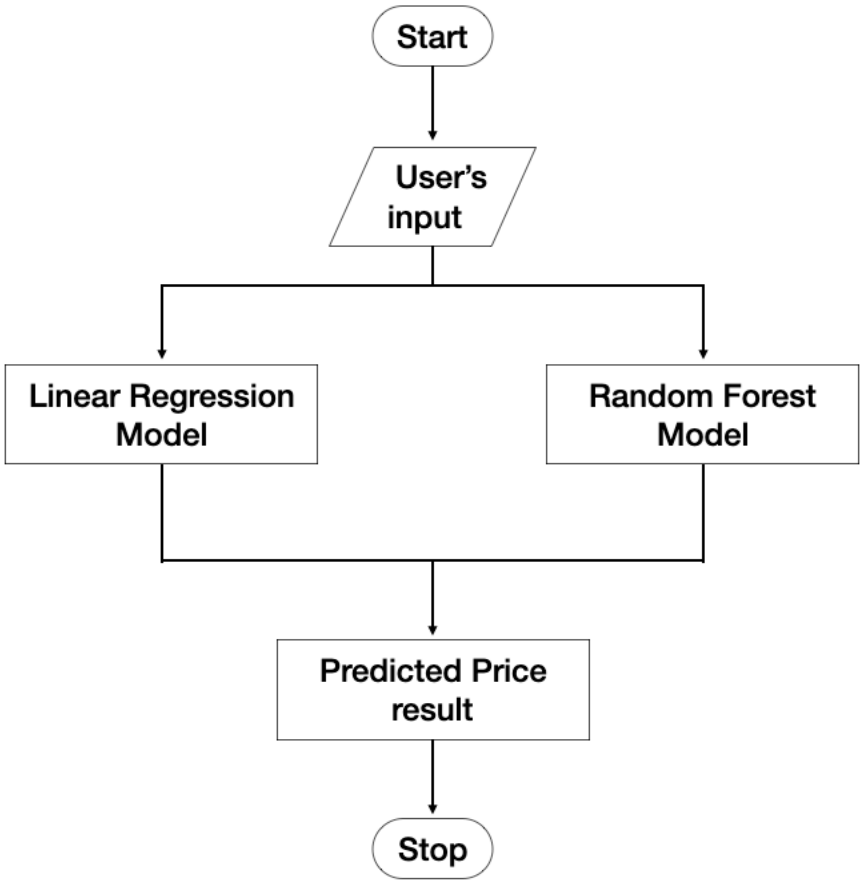


Figure 4 – System Flow Diagram of Performance Analysis of Gold Price System

4.2 Data Visualization

I used charts and bar graphs to visualize the price of the gold based on years and months by using matplotlib library.

1. Price based on months of a year with bar graph

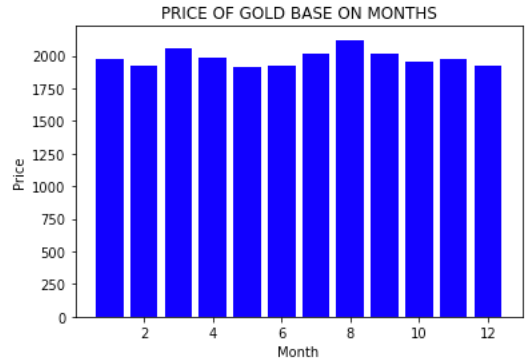
Visualization

Bar Graph

Price base on month of a year

```
plt.bar(df['Month'],df['Price'],color='blue')
plt.xlabel('Month')
plt.ylabel('Price')
plt.title('PRICE OF GOLD BASE ON MONTHS')
```

```
Text(0.5, 1.0, 'PRICE OF GOLD BASE ON MONTHS')
```

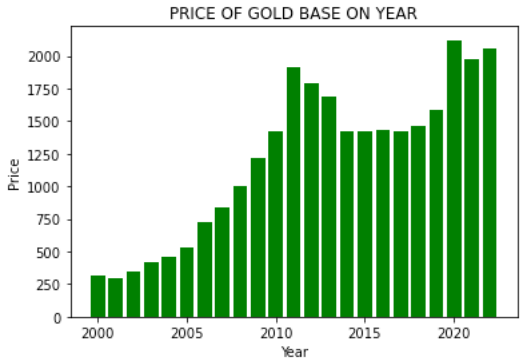


2. Price based on years from 2000 – 2022 with bar graph

Price base on the Year

```
plt.bar(df['Year'],df['Price'],color='Green')
plt.xlabel('Year')
plt.ylabel('Price')
plt.title('PRICE OF GOLD BASE ON YEAR')
```

```
Text(0.5, 1.0, 'PRICE OF GOLD BASE ON YEAR')
```



3. Accuracy and Mean Square Error for Linear Regression Model

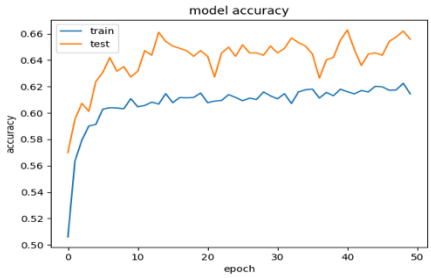
Model Evaluation For Linear Regression

Checking Accuracy

```
model_accuracy = linear_R.score(X_train, Y_train)
```

```
plt.plot(df['Price'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train'], loc='upper left')
```

```
<matplotlib.legend.Legend at 0x7fcd837927c0>
```



Mean Square Error

```
from sklearn.metrics import mean_absolute_error,mean_squared_error
mean_absolute_error(y_test,y_pred_df)
```

```
0.21435522699781953
```

```
mean_squared_error(y_test,y_pred_df)
```

```
0.1330631627486958
```

```
from math import sqrt
sqrt(mean_squared_error(y_test,y_pred_df))
```

```
0.36477823776740825
```

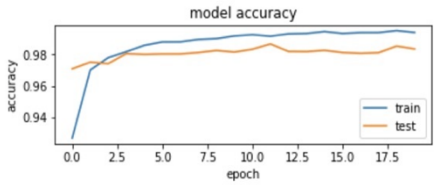
4. Accuracy and Mean Square Error for Random Forest Model

Checking Accuracy

```
accur_rf = Random_F.score(X_train, Y_train)
```

```
plt.subplot(2,1,1)
plt.plot(accur_rf)
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='lower right')
```

```
<matplotlib.legend.Legend at 0x7fcd83710640>
```



Mean Square Error

```
mean_absolute_error(y_test,y_pred_rfr)
```

```
0.18007575757575758
```

```
mean_squared_error(y_test,y_pred_rfr)
```

```
0.11990176767676768
```

```
sqrt(mean_squared_error(y_test,y_pred_rfr))
```

```
0.3462683463396094
```

4.3 Graphical User Interface of the System

The system's Home Page will be the first thing user will see when using the system in Fig 5. The user will be able to choose from two models which are Linear Regression and Random Forest. The user then give the required attributes to the system. After giving the day, month, year attributes with the valid inputs, the system will select the model that the user had chosen and predict the result with the give attributes in Fig 6 and 7.

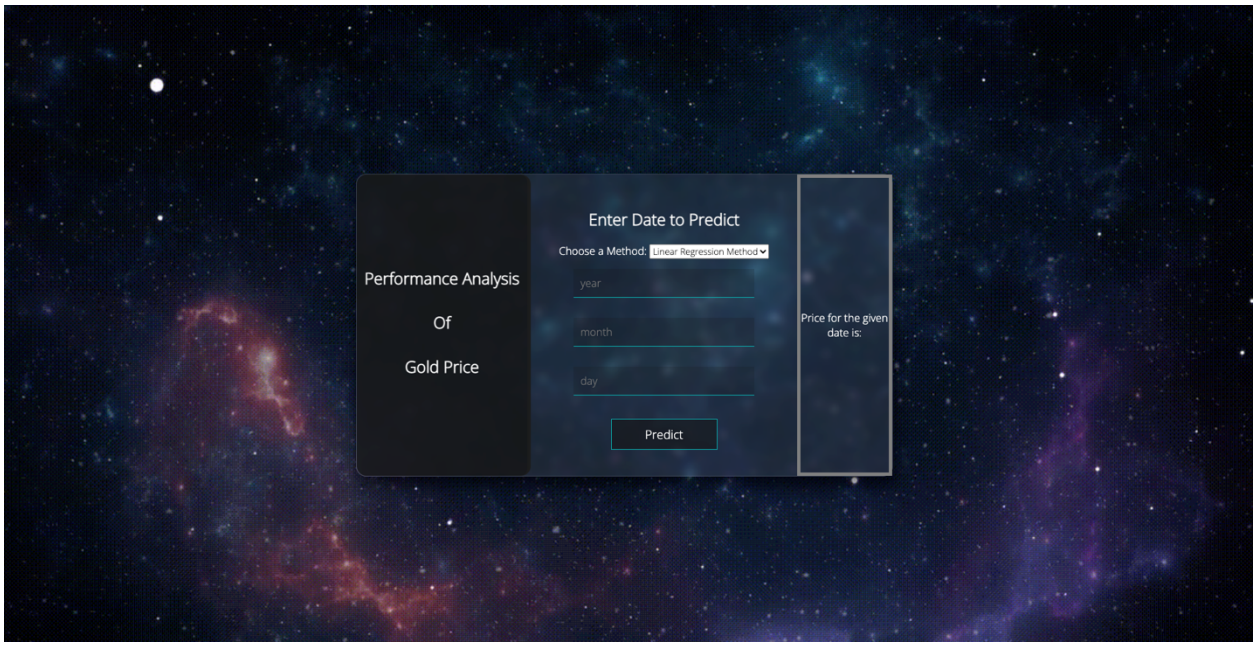


Figure 5 – Home Page

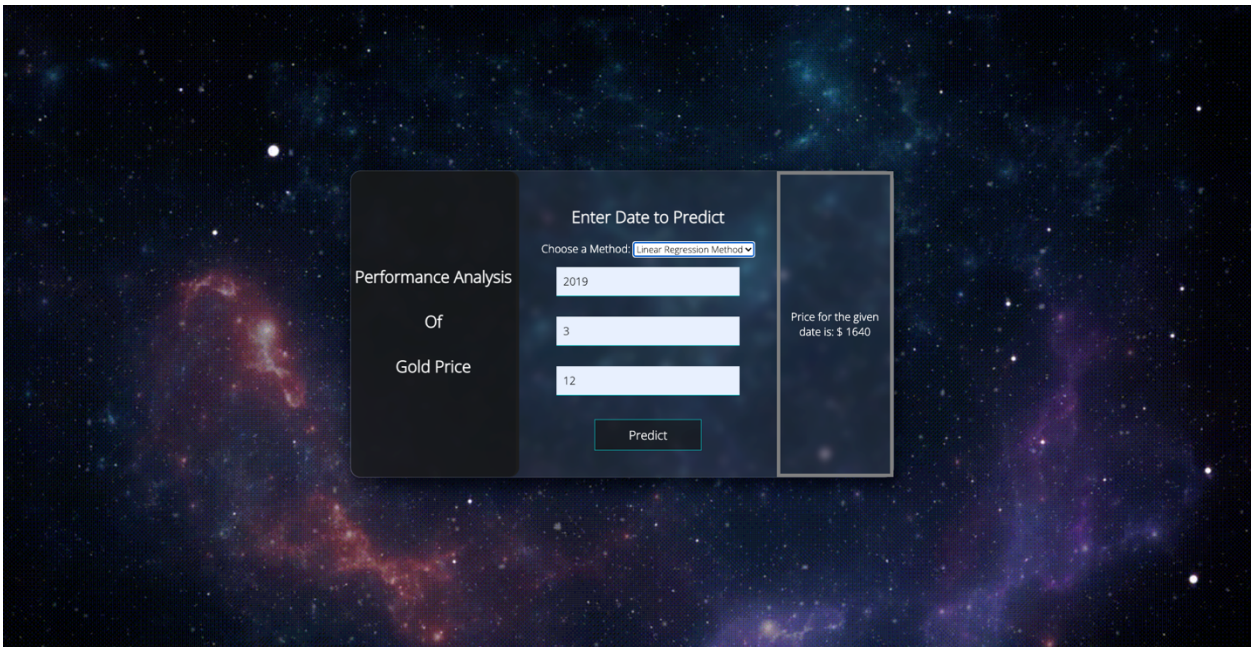


Figure 6 – Predicted result with Linear Regression Model

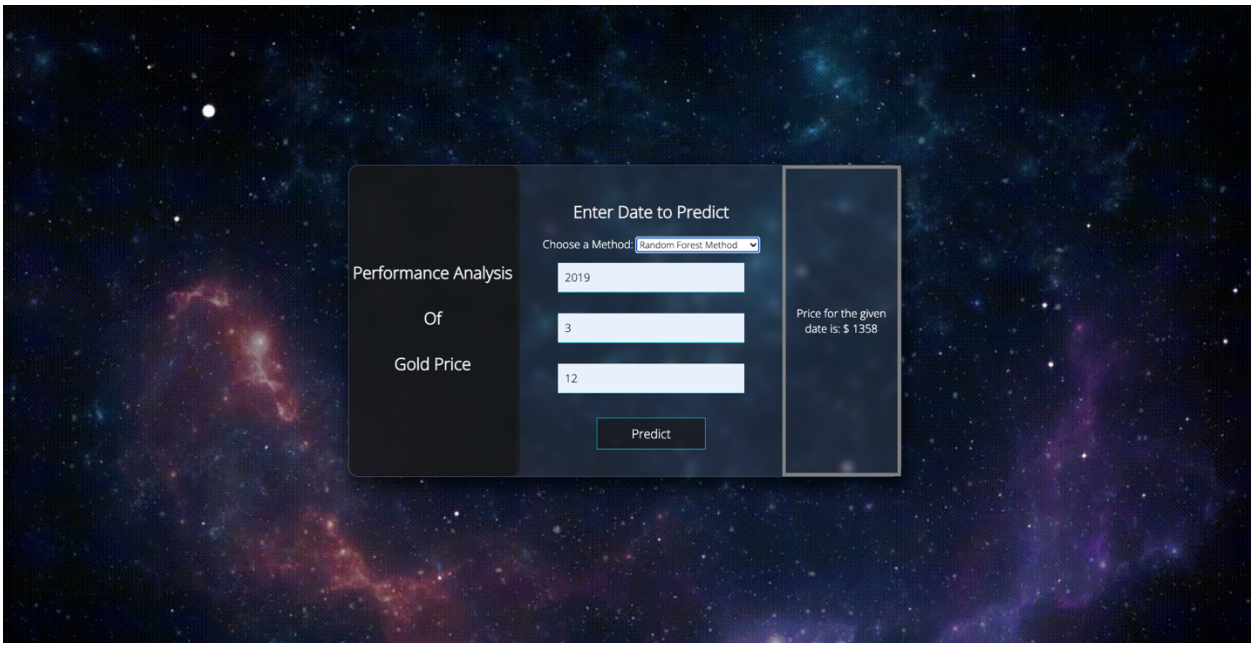


Figure 7 – Predicted result with Random Forest Model

5. Testing

5.1 Test Case and Specification

No.	1.
Category	Home Page
Normal/Error	Normal
Test Item	Display “Home Page” screen
Test Conditions	Access “ http://127.0.0.1:5000/ ” from browser
Expected Result	Home Page Screen and the required form will be displayed
How to check	Check browser screen. Confirm whether the screen layout follows the Interface Design. Confirm whether the screen shows up the expected result.

Table 1 – Test Case Specification for Home Page

No.	2.
Category	Home Page
Normal/Error	Normal
Test Item	Display result
Test Conditions	Access "http://127.0.0.1:5000/predict" from browser
Expected Result	The result is predicted with the selected model
How to check	Check browser screen. Confirm whether the screen layout follows the Interface Design. Confirm whether the screen shows up the expected result.

Table 2 – Test Case Specification for Result

6. Application Configuration

6.1 Hardware Requirements

- Processor: at least intel core i5
- RAM: at least 2.00GB
- Hard disk: 40GB
-

6.2 Software Requirements

- OS: Mac, Linux, Window
- IDE: Visual Studio Code, Atom, IDLE, Jyupter Notebook
- Internet Browser: Chrome, Safari, Firefox

6.3 Web Application URLS

Home Page

http://127.0.0.1:5000/

Predicted result

http://127.0.0.1:5000/predict

7.Conclusion

In conclusion, Performance Analysis of Gold Price System aims to help the merchants and investors to have a better decision while doing the gold business. The system shows the predicted price of the gold with two different models. Moreover, users can easily use this estimating system since all inputs can be directly referenced from from an online data repository company called Kaggle.

In this system, it is desired about how the system work, purpose of it, and how the Linear Regression and Random Forest Algorithms have worked in behind. I wish this system can help all the entrepreneurs who are about to do the gold business.

Tables and Figures

Figures

Tables