Golden Insights: Unlocking the Secrets of Gold Price Prediction

Done by:
Javier Tay Yu Xiang U2330451H
Jared Ong Kai Ze U2330650L
Ernest Then Shi Sheng U2331141L

Table of contents

01

Motivation

“Gold is the most stable asset”

FAKE

- Gold prices have fluctuated dramatically over the past century, with several large swings in both directions.
- Inflation, geopolitical tensions, supply and demand, and mining and refining costs influence the price of gold.

Invest Smarter

In this uncertain economy, it is difficult to make the right decisions when investing. Hence, we seek to uncover valuable insights into market trends, empowering investors with more informed decision-making capabilities in the dynamic world of finance

Problem Definition

- Find out which stock is the best predictor of gold price
- Classify whether to buy gold or not

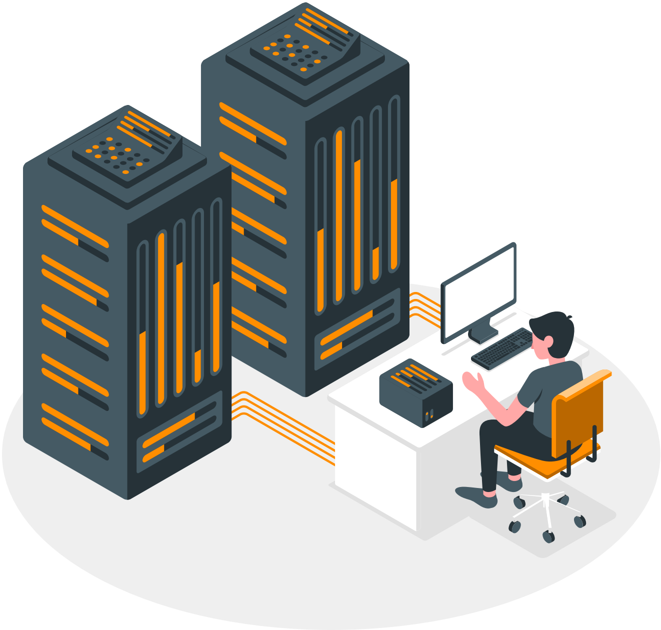
Our Dataset

- We used this dataset on the various US stock prices and volumes. https://www.kaggle.com/datasets/saketk511/2019-2024-us-stock-market-data
- This dataset offers an intricate exploration of market dynamics spanning five years (2019-2024)
- It represents a valuable dataset for dissecting trends and patterns within the global markets.

```
[ ] # Importing the dataset
    stocks = pd.read_csv('Stock Market Dataset.csv')
    stocks.head()
```

	Unnamed: 0	Date	Natural_Gas_Price	Natural_Gas_Vol.	Crude_oil_Price	Crude_oil_Vol.	Copper_Price	Copper_Vol.	Bitcoin_Price	Bitcoin_Vol.	...	Berkshire_Price	Berksl
0	0	02-02-2024	2.079	NaN	72.28	NaN	3.8215	NaN	43,194.70	42650.0	...	5,89,498	
1	1	01-02-2024	2.050	161340.0	73.82	577940.0	3.8535	NaN	43,081.40	47690.0	...	5,81,600	
2	2	31-01-2024	2.100	142860.0	75.85	344490.0	3.9060	NaN	42,580.50	56480.0	...	5,78,020	
3	3	30-01-2024	2.077	139750.0	77.82	347240.0	3.9110	NaN	42,946.20	55130.0	...	5,84,680	
4	4	29-01-2024	2.490	3590.0	76.78	331930.0	3.8790	NaN	43,299.80	45230.0	...	5,78,800	

5 rows × 39 columns



02

Exploratory Data Analysis

Steps Taken for EDA

C V C S

Cleaning

Find and drop null values

Visualise

Plot graphs to visualise the data

Correlation

Find out which stocks to use

Statistics

Find out the stats for each stock

Data Cleaning

```
stocks.isnull().sum()
```

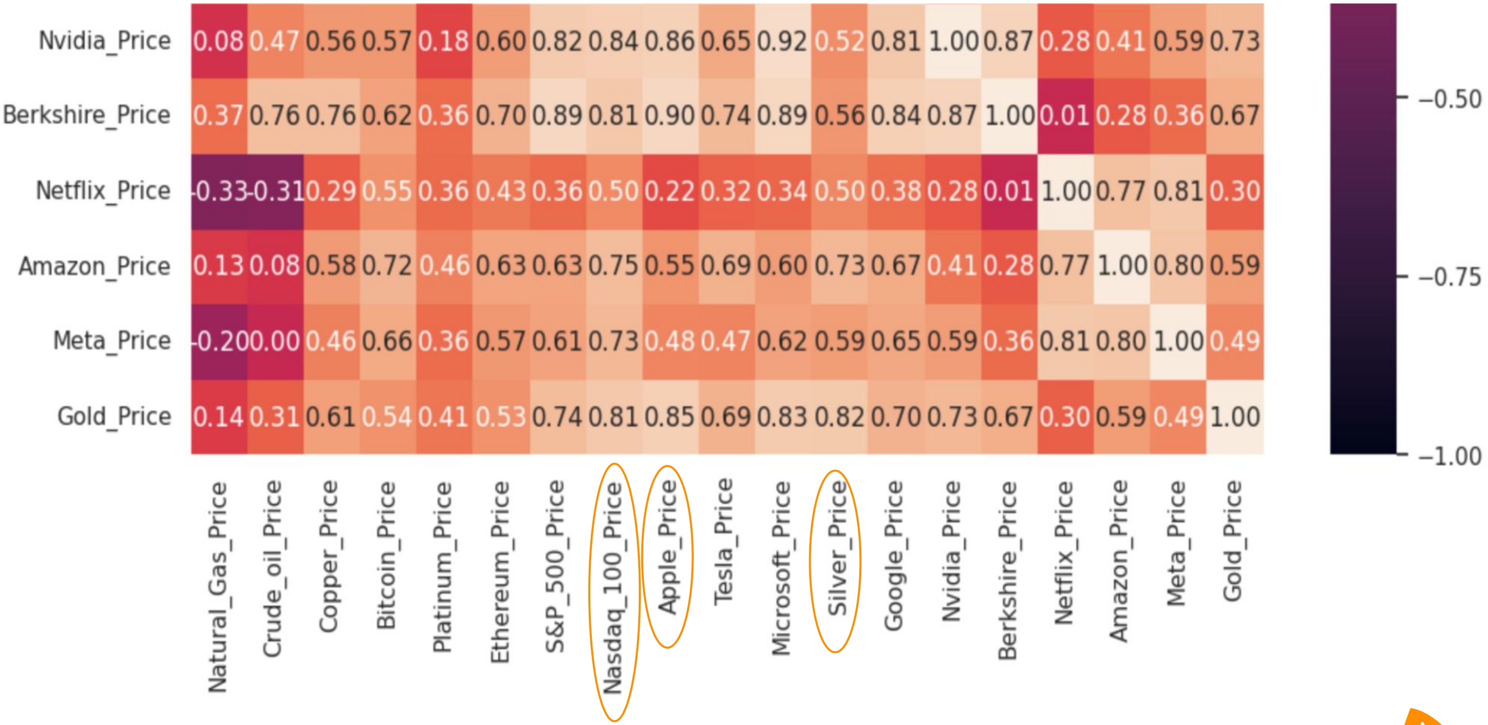
```
Date                 0
Natural_Gas_Price    0
Natural_Gas_Vol.     4
Crude_oil_Price      0
Crude_oil_Vol.      23
Copper_Price         0
Copper_Vol.         37
Bitcoin_Price        0
Bitcoin_Vol.         0
Platinum_Price       0
Platinum_Vol.      607
```

Since the null values only exist in the `Vol.` columns, we can just remove them because we are only interested in using the `Price` to predict the `Gold_Price`.

```
[ ] # New dataframe with prices only

    stocks_price = stocks[["Date", "Natural_Gas_Price", "Crude_oil_Price", "Copper_Price", "Bitcoin_Price", "Platinum_Price", "Ethereum_Price", "S&P_500_Price", "Nasdaq_100_
    stocks_price.head()
```

	Date	Natural_Gas_Price	Crude_oil_Price	Copper_Price	Bitcoin_Price	Platinum_Price	Ethereum_Price	S&P_500_Price	Nasdaq_100_Price	Apple_Price	Tesla_Price	Micros
0	02-02-2024	2.079	72.28	3.8215	43,194.70	901.6	2,309.28	4,958.61	17,642.73	185.85	187.91	
1	01-02-2024	2.050	73.82	3.8535	43,081.40	922.3	2,304.28	4,906.19	17,344.71	186.86	188.86	
2	31-01-2024	2.100	75.85	3.9060	42,580.50	932.6	2,283.14	4,848.87	17,137.24	184.40	187.29	
3	30-01-2024	2.077	77.82	3.9110	42,946.20	931.7	2,343.11	4,924.97	17,476.71	188.04	191.59	

Preliminary exploration



Statistics

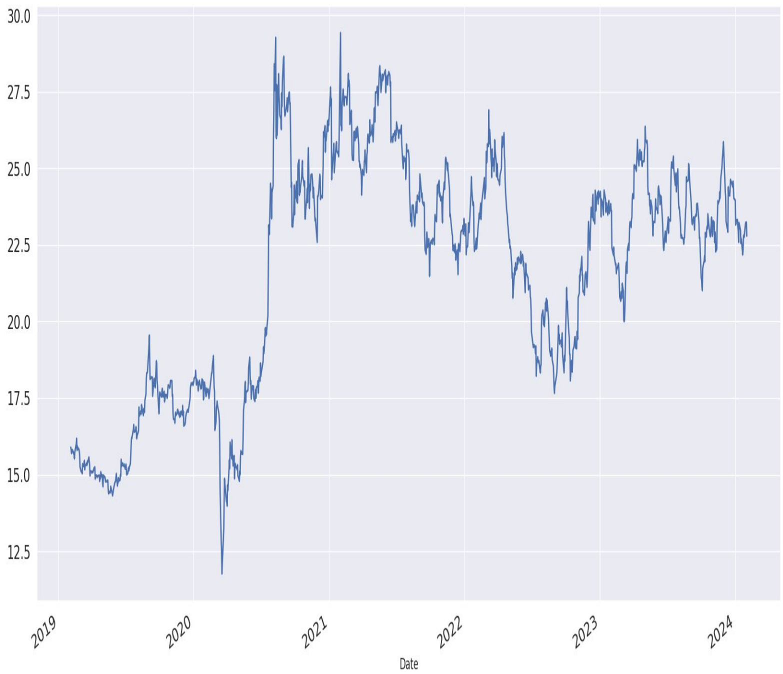
```
[ ]  predictorprices_final.describe()
```

	Nasdaq_100_Price	Apple_Price	Silver_Price	Gold_Price
count	1243.000000	1243.000000	1243.000000	1243.000000
mean	12037.318101	125.566533	21.588977	1759.246742
std	2887.069742	46.114122	3.859288	203.258901
min	6904.980000	42.360000	11.772000	1272.000000
25%	9298.730000	79.505000	17.998500	1669.600000
50%	12381.170000	136.760000	22.758000	1804.200000
75%	14563.250000	162.915000	24.512000	1912.800000
max	17642.730000	198.110000	29.418000	2089.700000

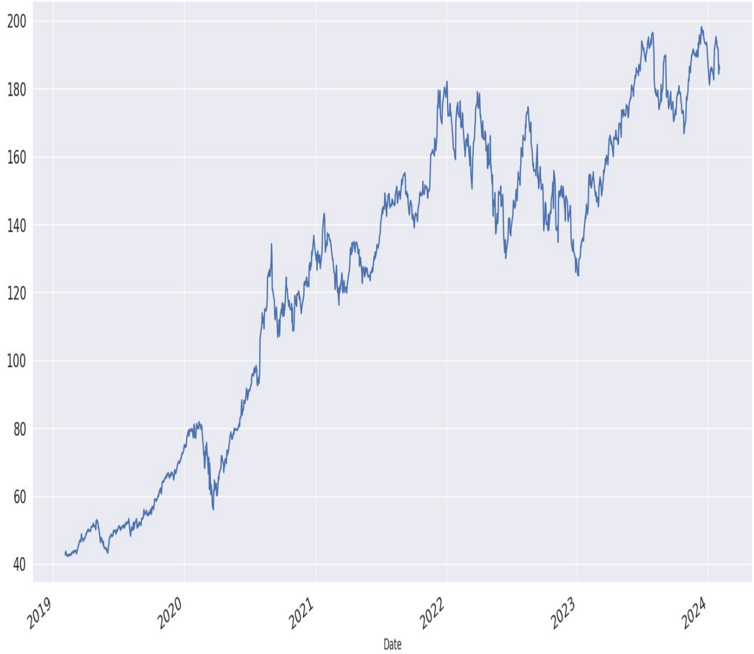
Graphs of Gold Price

Graphs of Nasdaq 100 Price

Graph of Silver Price



Graph of Apple Price



03

Core Analysis

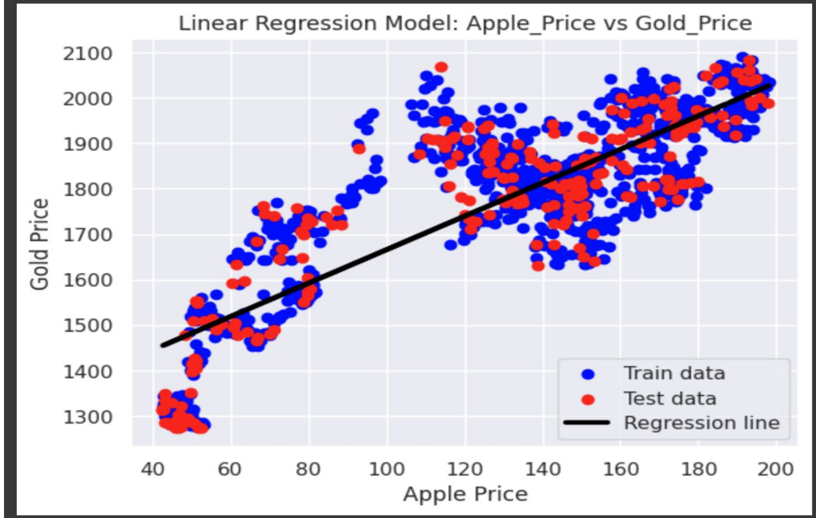
Machine Learning Techniques

	Description
Linear Regression	Find relationship between the stocks by fitting a straight line to the observed data, aiming to minimize the difference between predicted and actual values.
SARIMA	Time series forecasting model that combines autoregression, differencing, moving average, and seasonal components to make predictions.
Random Forest	Build multiple decision trees and combine predictions to make more accurate and robust predictions.

Linear Regression using Apple Prices

```python
# test_size 0.2 such that its a 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(nasdaq100, goldprice, test_size = 0.20)

# Create the Linear Regression model
linreg = LinearRegression()
linreg.fit(X_train,y_train)

# Coefficients of the Linear Regression line
print('Intercept of Regression \t: b = ', linreg.intercept_)
print('Coefficients of Regression \t: a = ', linreg.coef_)
print()

y_train_pred = linreg.predict(X_train)
y_test_pred = linreg.predict(X_test)
```

```
Intercept of Regression         : b =  [1297.68446525]
Coefficients of Regression      : a =  [[3.6806378]]
```

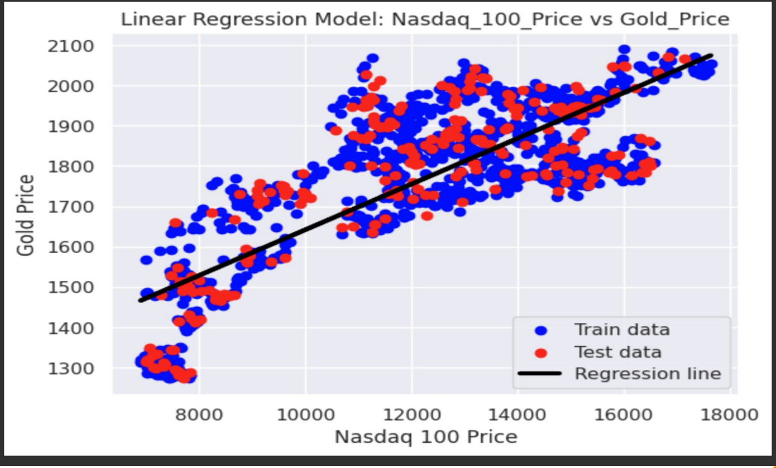



```
Goodness of Fit Metrics (Train set):
R^2 on Train set: 0.7114325131072741
Mean Squared Error on Train set: 11508.064487779746

Prediction Accuracy Metrics (Test set):
R^2 on Test set: 0.7495468958688507
Mean Squared Error on Test set: 11703.359884106838
```

Linear Regression using Nasdaq 100

```python
# test_size 0.2 such that its a 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(nasdaq100, goldprice, test_size = 0.20)

# Create the Linear Regression model
linreg = LinearRegression()
linreg.fit(X_train,y_train)

# Coefficients of the Linear Regression line
print('Intercept of Regression \t: b = ', linreg.intercept_)
print('Coefficients of Regression \t: a = ', linreg.coef_)
print()

y_train_pred = linreg.predict(X_train)
y_test_pred = linreg.predict(X_test)
```

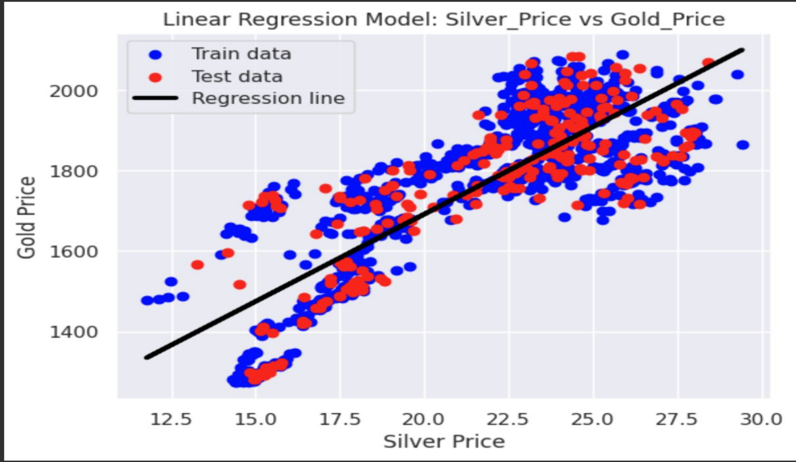
```
Intercept of Regression          : b =   [1074.29260561]
Coefficients of Regression       : a =   [[0.05671053]]


Goodness of Fit Metrics (Train set):
R^2 on Train set: 0.6561344691362083
Mean Squared Error on Train set: 14121.17934273839

Prediction Accuracy Metrics (Test set):
R^2 on Test set: 0.6249975698420803
Mean Squared Error on Test set: 15801.16430601258
```



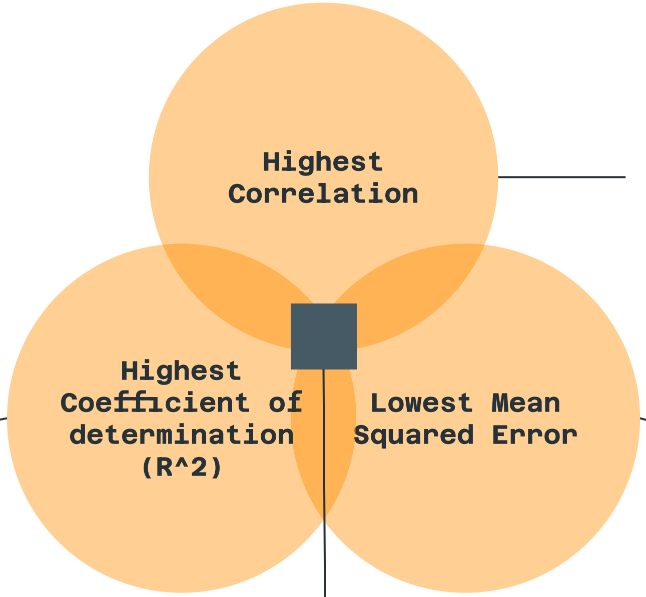
Linear Regression using Silver

```python
# test_size 0.2 such that its a 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(silver, goldprice, test_size = 0.20)

# Create the Linear Regression model
linreg = LinearRegression()
linreg.fit(X_train,y_train)

# Coefficients of the Linear Regression line
print('Intercept of Regression \t: b = ', linreg.intercept_)
print('Coefficients of Regression \t: a = ', linreg.coef_)
print()

y_train_pred = linreg.predict(X_train)
y_test_pred = linreg.predict(X_test)
```

```
Intercept of Regression         : b =  [823.01271139]
Coefficients of Regression      : a =  [[43.42303426]]

Goodness of Fit Metrics (Train set):
R^2 on Train set: 0.6704656418845341
Mean Squared Error on Train set: 13861.224407809319

Prediction Accuracy Metrics (Test set):
R^2 on Test set: 0.6550195239895384
Mean Squared Error on Test set: 13162.492876458999
```



Apple is the Winner!



Apple prices has a strong linear relationship with gold prices.

Indicates that a large percentage of the variability in the gold price is accounted for by Apple prices

Indicates that the predictions made are close to the actual values. Hence, minimal deviation from the true values.

Combining these results, Apple is the best predictor

SARIMA Model

What is SARIMA?

Seasonal Autoregressive Integrated Moving Average

Autoregressive Component

Relationship between an observation and lagged observations from previous time steps.

Integrated Component

Removes trends and seasonality by differencing the time series.

Moving Average

Relationship between an observation and the residual error from lagged observations.

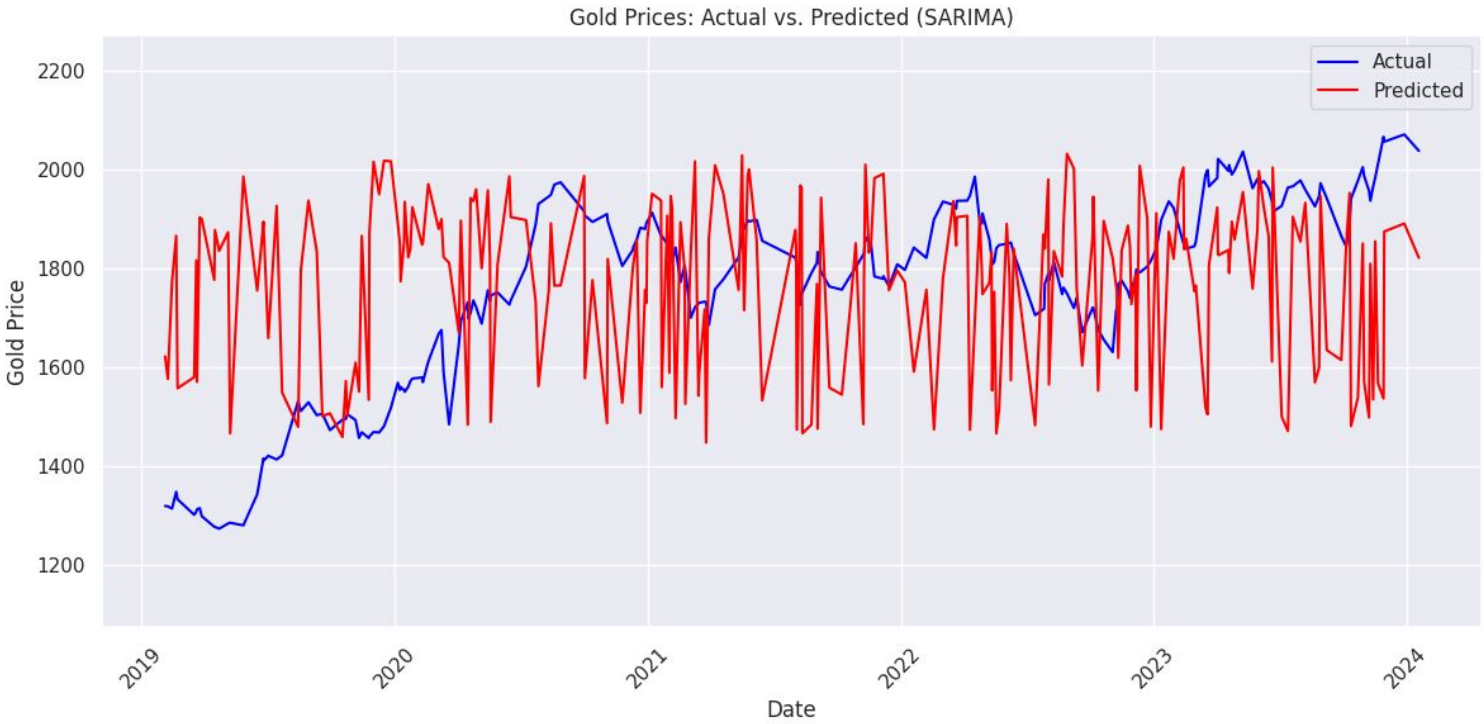
Seasonal Component

Include seasonal autoregressive, integrated, and moving average terms, addressing recurring patterns over fixed intervals.

Forecasting

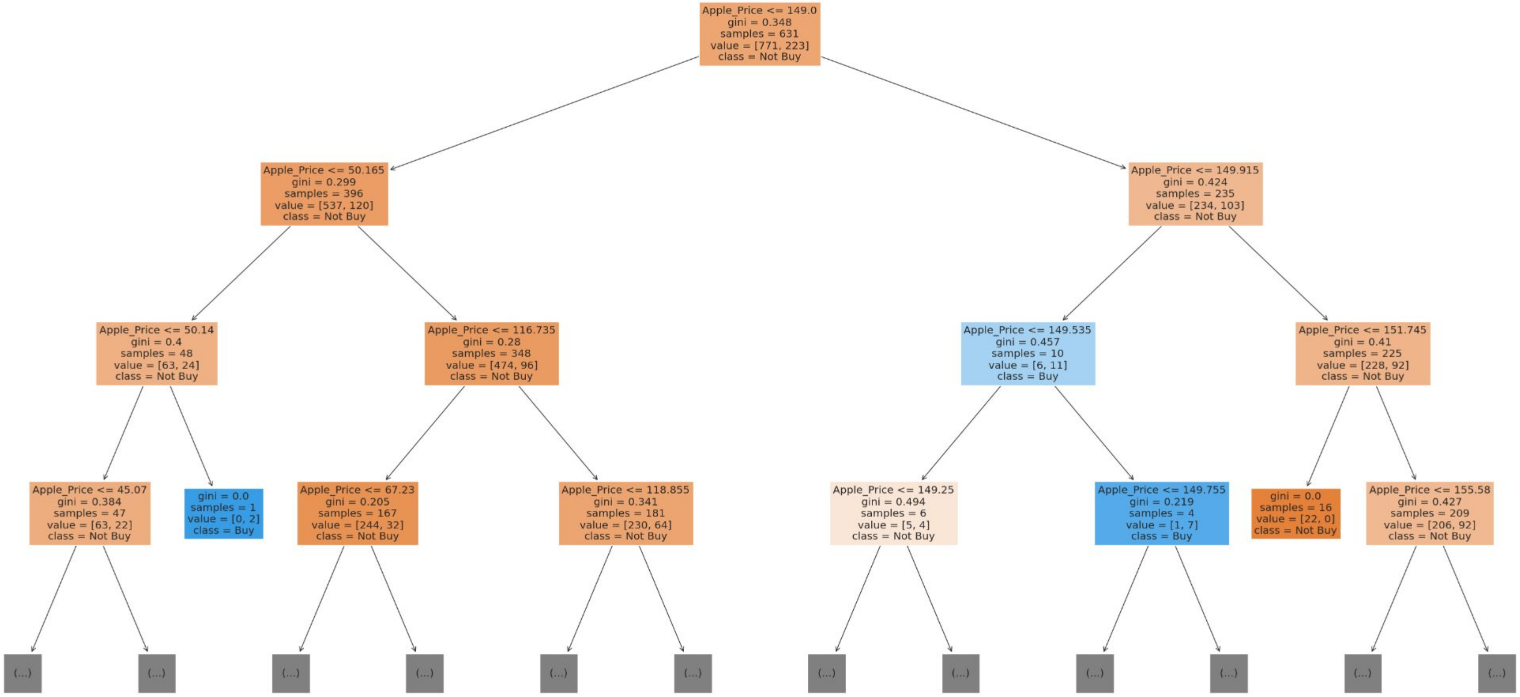
Forecasts future values based on historical data and estimated parameters, extrapolating captured patterns into the future.

Prediction Using SARIMA

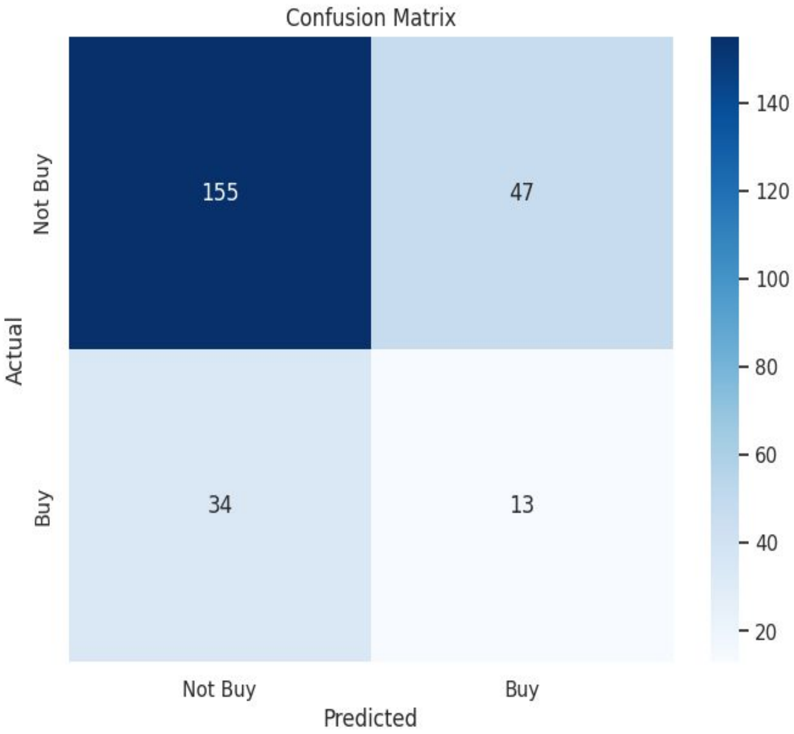


Buy or Not?

Random Forest Tree



Confusion Matrix




```
Accuracy: 0.6746987951807228
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.77      0.79       202
           1       0.22      0.28      0.24        47

    accuracy                           0.67       249
   macro avg       0.52      0.52      0.52       249
weighted avg       0.71      0.67      0.69       249

Confusion Matrix:
[[155  47]
 [ 34  13]]
```

04

Conclusion

Outcome of Analysis

Comparison between stock prices of Apple, Silver and Nasdaq 100

Apple Stock Prices	Silver Prices	Nasdaq 100 Prices
Correlation = 0.8488	Correlation = 0.8171	Correlation = 0.8062
MSE = 11703	MSE = 13162	MSE = 15801
R^2 = 0.7495	R^2 = 0.6550	R^2 = 0.6250

Data Driven Insights

Predictable

- Using our prediction models like the random forest, we have a relatively high accuracy.
- There are multiple stocks that we can use to predict gold prices.
- Thus, it is entirely possible to predict future gold prices.

Not Perfect

- The prediction models are not 100% accurate. This means that we should not entirely depend on it.
- There are very high risks of investing in gold as the fluctuations are hard to predict.
- We should all be wary when investing. There is no “fast way” to become rich.

Thank You!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Storyset**