Market Research Dataset – Performance Assessment

Steven Schindler

Regression and Time Series analysis of market research dataset

Sewell, William; PhD

College of I.T., Western Governors University

August 04, 2023

Research Question

The Coca-Cola company went public in 1919 with 600,000 shares at 40$ per share (Nasdaq. (2020, December 6)). Over 100 years later the company holds a market share of 42% of the global soft drink market and reported revenue of $37.3 billion in 2020(tutor2u Economics. (n.d.)). This analysis attempts to make a predictive model of Coca-Cola stock prices of up to a year in advance. This analysis could then assist a stock trader with the best times to buy or sell Coca-Cola stock up to a year in advance.

Can a Regression and Time Series analysis make a predictive model for market research dataset of the Coca-Cola stock price? The null hypothesis is a predictive model for Coca-Cola Stock prices cannot be constructed with an accuracy of 70%. The alternative hypothesis is a predictive model for Coca-Cola Stock prices can be constructed with an accuracy of at least 70%.

Data Collection

The data includes 60 years of stock price history collected and shared from the website Kaggle. Kaggle is an online community platform for data scientists that publish open-source datasets (DataCamp. (n.d.))[1]. As stated above the data shared is 60 years of stock price history from Jan 1st, 1962, to Oct 25th, 2022. The main advantage of using Kaggle is that the data is already collected and presented in an easy CSV format. A disadvantage is that the data is not the most recent as it only goes to Oct 25th, 2022. The data is located here (Rahman, K. (n.d.)).

The fields in the data are listed in the table below:

Variable	Description	Data type
Date	Date from 1962-2022	datetime/continuous
Open	Open price at opening of the stock market	continuous
High	High price for that day	continuous
Low	Low price for that day	continuous
Close	Close price at closing of stock market	continuous
Volume	Volume Traded	continuous
Dividends	Dividends paid	continuous

Stock Splits	Stock Splits	continuous

Data Extraction and Preparation

The next step is to prepare the data for analysis using Python as the tool of choice. Python has many tools used for data science including the Pandas library. Pandas offers many advantages such as the ability to efficiently handle large datasets and provide the data in an easy representation (Patil, R. (n.d.)). A disadvantage of Pandas is the memory consumption as working with large data sets Pandas will use a large amount of memory (Patil, R. (n.d.)).

The first step of the cleaning process is to read the CSV file into an easily manipulated DataFrame, below shows that step as well as information about the data set.

```
In [1]: import pandas as pd

        df = pd.read_csv('Coca-Cola_stock_history.csv')
        df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15311 entries, 0 to 15310
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          15311 non-null  object
 1   Open          15311 non-null  float64
 2   High          15311 non-null  float64
 3   Low           15311 non-null  float64
 4   Close         15311 non-null  float64
 5   Volume        15311 non-null  int64
 6   Dividends     15311 non-null  float64
 7   Stock Splits  15311 non-null  int64
dtypes: float64(5), int64(2), object(1)
memory usage: 957.1+ KB
```

The info method shows the data type of the column and that there are 15,311 entries with the number of non-null values for each column. Using the isna() and the sum() methods which check for null values and counts the number respectively, then confirm that there are no null values.

```
: df.isna().sum()
```

```
: Date            0
  Open            0
  High            0
  Low             0
  Close           0
  Volume          0
  Dividends       0
  Stock Splits    0
  dtype: int64
```

Next convert the date column to a datetime object to give the date consistency of how it is formatted. Duplicate values are the next thing to check for using, the nunique() method which gives the number of unique values that are shown below.

```
[5]: df.nunique()

[5]: Date      15311
     Open      14855
     High      14547
     Low       14580
     Close     11105
     dtype: int64
```

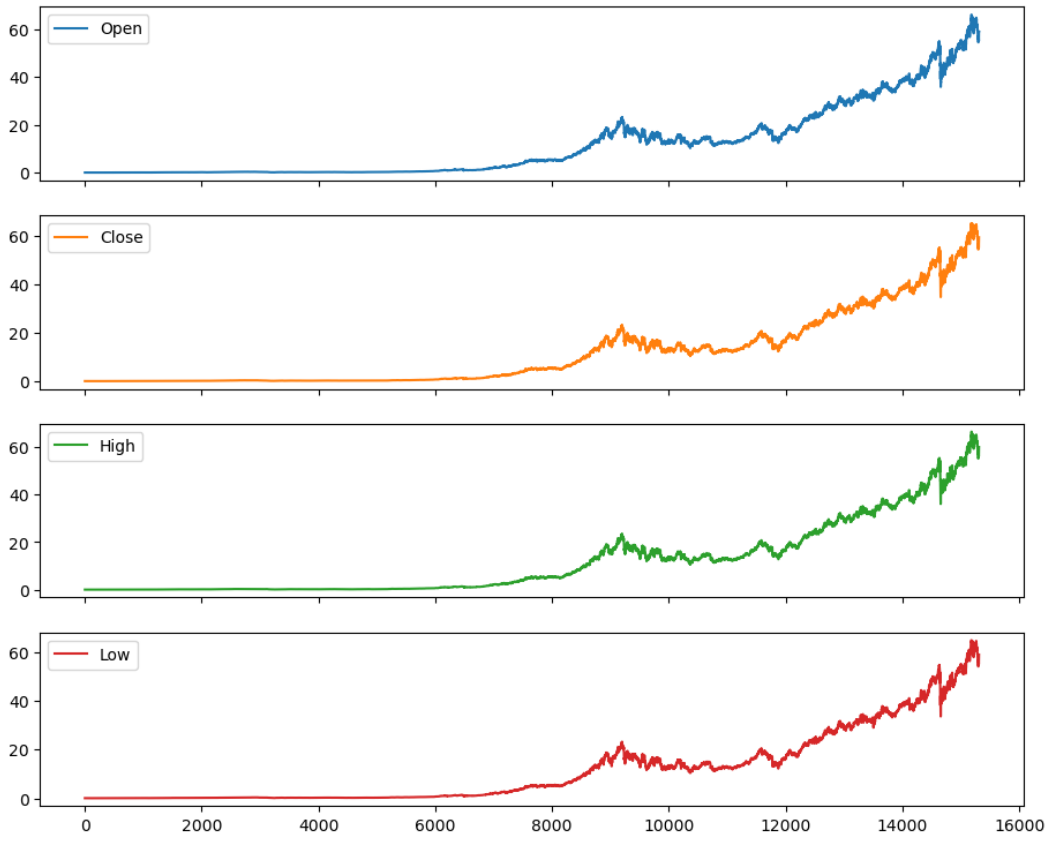
Date is the only column that has completely unique values which is alright though as the other columns could have prices that are the same such as, the high price for the day being equal to the close price of the day. There could also be dates where the values are the same but are on different dates such as, two Open dates that have the same value but are weeks apart. Checking for null values and duplicates as well as removing unnecessary columns the cleaning is now complete and the data is ready for analysis.

Analysis

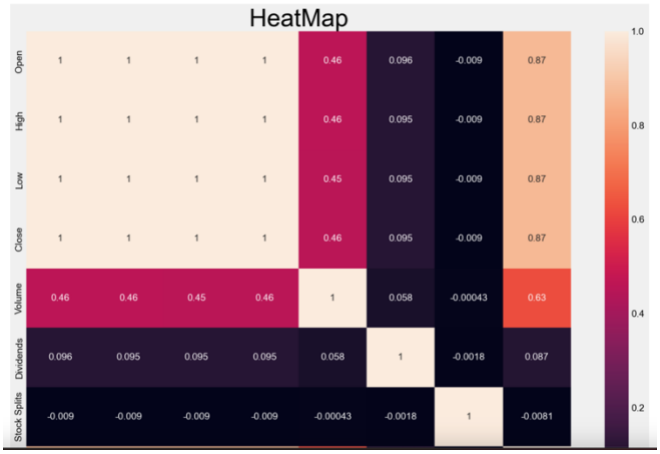
The two analysis techniques used are Multiple Linear Regression (MLR) and Autoregressive integrated moving average or Arima. MLR is a statistical technique that attempts to model the relationship between two or more explanatory variables and a response variable y fitting the data to a linear equation (Yale University Department of Statistics. (n.d.)). Arima is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends (Investopedia. (n.d.)). One advantage of regression is that it is simple to implement, and the coefficients are easy to interpret while a disadvantage is that it is susceptible to overfitting (GeeksforGeeks. (n.d.)).MLR is in the form of $Y = B_0 + B_1X_1 + B_2X_2 \ldots$ with B_0 being the y-intercept, B_1,$B_{2\ldots}$ being the coefficients of the X variables which are the independent variables, while Y is the dependent variable. One advantage of an Arima model is that it can handle a wide range of time series data if it is univariate, while a disadvantage is that because it

is univariate it cannot capture the interactions and dependencies between different variables (LinkedIn. (n.d.)).

The analysis begins with an exploratory look at some data visualizations.



We see from the picture above that Open, Close ,High and Low are very similar which may lead to multicollinearity. We see from the heatmap below as well as the VIF scores that Open, Close, High and Low have high multicollinearity.




```
6]: from statsmodels.stats.outliers_influence import variance_inflation_factor
X = df[['Close','Open','High','Low','Volume', 'Dividends','Stock Splits','date_delta']]

# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = X.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                   for i in range(len(X.columns))]

print(vif_data)
```

	feature	VIF
0	Close	44326.043243
1	Open	38947.564611
2	High	48447.965527
3	Low	41399.624343
4	Volume	4.382166
5	Dividends	1.018906
6	Stock Splits	1.000640
7	date_delta	8.926260

Because of the high multicollinearity the independent variables will be Date and Volume while the dependent variable will be Close for the Linear Regression analysis. Date has been converted to float as the number of days since the beginning of the data called delta_date.

```
df['date_delta'] = (df['Date'] - df['Date'].min())  / np.timedelta64(1,'D')
df['date_delta']
```

```
0            0.000000
1            1.000000
2            2.000000
3            3.000000
4            6.000000
            ...
15306    22206.166667
15307    22207.166667
15308    22210.166667
15309    22211.166667
15310    22212.166667
Name: date_delta, Length: 15311, dtype: float64
```

First the data is separated into a training data set and testing data set with the split at 80%:20% respectively.

```
x = df[['Volume','date_delta']]
y = df['Close']
```

```
train_x, test_x, train_y, test_y = train_test_split(x, y, test_size=0.20 , shuffle=False,random_state = 0)
```

Next, import the regression library from sklearn and fit the data to the regression model to extract the coefficients and the y-intercept.

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import confusion_matrix, accuracy_score
regression = LinearRegression()
regression.fit(train_x, train_y)
print("regression coefficient",regression.coef_)
print("regression intercept",regression.intercept_)
```

```
regression coefficient [-1.12313269e-07  1.22789933e-03]
regression intercept -4.699962581328602
```

From the picture above we now know the equation is $Y = -4.699 + -1.123X_1 + 1.227X_2$. Next using the predict method on testing data and comparing it to actual prices of Coca-Cola this dataframe is created:

```
In [75]: predicted=regression.predict(test_x)
         #print(test_x.head())
```

```
In [76]: dfr=pd.DataFrame({'Actual_Price':test_y, 'Predicted_Price':predicted})
         dfr.head()
```

Out[76]:

	Actual_Price	Predicted_Price
12248	19.236696	14.910077
12249	19.027758	15.974627
12250	19.140793	15.038287
12251	19.630617	14.410762
12252	19.654594	15.475013

Then running mean absolute error(MAE), mean square error(MSE), and root mean square error(RMSE) on the dataframe these values are returned:

```
print('Mean Absolute Error (MAE):', metrics.mean_absolute_error(test_y, predicted))
print('Mean Squared Error (MSE) :', metrics.mean_squared_error(test_y, predicted))
print('Root Mean Squared Error (RMSE):', np.sqrt(metrics.mean_squared_error(test_y, predicted)))
```

```
Mean Absolute Error (MAE): 19.46095411599748
Mean Squared Error (MSE) : 463.9558348904942
Root Mean Squared Error (RMSE): 21.539634047274205
```

Accuracy is calculated at -107.04

```
x2 = dfr.Actual_Price.mean()
y2 = dfr.Predicted_Price.mean()
Accuracy1 = (y2-x2)/y2 *100
print("The accuracy of the model is " , Accuracy1)
```

```
The accuracy of the model is  -107.04640220578617
```

With this low accuracy and poor MAE, MSE, and RMSE the NULL hypothesis cannot be rejected.

Data Summary and Implications

Sources

Nasdaq. (2020, December 6). A $120 Investment in Coca-Cola's IPO Would Be Worth This Much Money Now. Nasdaq. Retrieved August 05, 2023, from https://www.nasdaq.com/articles/a-%24120-investment-in-coca-colas-ipo-would-be-worth-this-much-money-now-2020-12-06

tutor2u Economics. (n.d.). Why Is Coca-Cola So Profitable? tutor2u Economics. Retrieved August 05, 2023, from https://www.tutor2u.net/economics/reference/why-is-coca-cola-so-profitable

DataCamp. (n.d.)[1]. What is Kaggle? Retrieved August 05, 2023, from https://www.datacamp.com/blog/what-is-kaggle

Rahman, K. (n.d.). Coca-Cola Stock - Live and Updated. Kaggle. Retrieved August 05, 2023, from https://www.kaggle.com/datasets/kalilurrahman/coca-cola-stock-live-and-updated?select=Coca-Cola_stock_history.csv

Patil, R. (n.d.). Exploring the Pros and Cons of Pandas. Medium. Retrieved August 06, 2023, from https://medium.com/@rohanjpatil63/exploring-the-pros-and-cons-of-pandas-1675463971d4

Yale University Department of Statistics. (n.d.). Linear Regression Analysis - Least Squares Estimation. Yale University Department of Statistics. Retrieved August 10, 2023, from http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm

Investopedia. (n.d.). Autoregressive Integrated Moving Average (ARIMA). Investopedia. Retrieved August 10, 2023, from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

GeeksforGeeks. (n.d.). ML | Advantages and Disadvantages of Linear Regression. Retrieved August 10, 2023, from https://www.geeksforgeeks.org/ml-advantages-and-disadvantages-of-linear-regression/

LinkedIn. (n.d.). What Are the Advantages and Disadvantages of ARIMA Models in Forecasting? Retrieved August 10, 2023, from https://www.linkedin.com/advice/3/what-advantages-disadvantages-arima-models-forecasting