GA’s Stock Prediction - Final Project

Class: Introduction to Machine Learning

Gabriela Caetano de Jesus - GitHub
Aleksandra Ristic - GitHub

December 1, 2023

Table of contents:

December 1, 2023

Introduction

For an investor seeking profitability in the stock market, the essence lies in the precise timing of stock purchases and sales. An important part of stock investment revolves around accurate estimation of stock prices. Leveraging the prowess of advanced machine learning models, our approach involves an analysis of historical data sequences. This analytical prowess enables us to forecast the future components of the sequence with a high degree of accuracy, empowering investors to make strategic decisions that translate into financial gains.

Problem: A stock price prediction poses a substantial challenge, full of complexities that demand meticulous attention. Problems, such as data noise, market volatility, unpredictable events, and the inherent intricacy of financial markets collectively create a difficult task for accurate predictions. Data noise, stemming from various sources, obscures patterns within historical stock data, requiring advanced analytical approaches. Market volatility introduces rapid fluctuations, demanding models that can adapt swiftly. Unpredictable events, from geopolitical shifts to unforeseen disasters, further complicate the landscape. Navigating these challenges requires an understanding of financial market complexities, motivating the development of the machine learning model that overcomes these obstacles to provide investors with accurate and reliable stock forecasts.

Objective: The goal of this project is to develop an accurate and reliable closing daily stock price prediction using historical stock data, to assist investors, traders, and financial analysts in making informed decisions about buying or selling stocks in a dynamic market.

In the table below, we summarized all similar projects and source codes that we have been using for developing our stock prediction models

Past Works Done

Title	Dataset	Model 1	Model 2	Goal	Link
"Predicting Financial Time Series by us of Shapelets and Trend Lines While Exploring the Notion of concept drift within"	**35 Stocks**	**Shapelet + LSTM + MLP**	**Trend Lines + RNN + MLP**	How concept drift can be detected and avoided in the financial time series context?	https://projects.cs.uct.ac.za/honsproj/cgi-bin/view/2018/bodley_debruin_venter.zip/
"Stock Market Price Forecasting Using the Arima Model: an Application to Istanbul, Turkiye"	**Istanbul Monthly Stock Market Data 2009 to 2021 - 147 Observations**	**ARIMA** (AR and MA process)	-	Stock market price prediction in Istanbul.	https://dergipark.org.tr/en/download/article-file/2187949
"Time Series Forecasting using ARIMA Model A Case Study of Mining Face Drilling Rig"	**Total cost of mining face drilling rig (2009 to 2012)**	**ARIMA** (AR and MA process)	**ANN** (Artificial Neural Network)	Estimate the optimal replacement time of the face drilling rig.	https://www.diva-portal.org/smash/get/diva2:1266336/FULLTEXT01.pdf
"Predictive analysis on Multivariate, Time Series datasets using Shapelets"	**Batch dataset from a Chemical process**	**Shapelet + LTS**	**Shapelet + Time Warping**	Predicting the quality outcome of an industrial batch process.	https://cs229.stanford.edu/proj2016/report/ThakkarPredictive_Analysis_on_Multivariate_TimeSeries_datasets_using_Shapelets-report.pdf

December 1, 2023

Dataset

The dataset consisted of historical daily closing stock prices starting from August 20, 2004, to October 25, 2023, including a profile of 12 diverse technology companies. These companies, drawn from the dynamic tech sector, showcase the evolution of their stock values over nearly two decades. The dataset serves as a valuable resource for understanding market trends, assessing long-term performance, and conducting insightful analyses on the stock behavior of prominent tech entities. The data source, retrieved from https://finance.yahoo.com/, ensures reliability and accessibility, laying the foundation for comprehensive examinations of stock market dynamics within the specified time frame.

- Daily closing stock prices of 12 different tech companies, which are listed below:
- A time period: 08/20/2004 to 10/25/2023
- Historical Data Source: https://finance.yahoo.com/

- **AAPL**
- **ADBE**
- **AMZN**
- **CRM**
- **CSCO**
- **GOOGL**
- **INTC**
- **IBM**
- **MSFT**
- **NFLX**
- **NVDA**
- **ORCL**

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date	APPL	ADBE	AMZN	CSCO	CRM	GOOGL	IBM	INTC	MSFT	NFLX	NVDA	ORCL
2	8/20/2004	0.55	23.05	1.98	18.88	3.15	2.71	81.5	21.62	27.2	2.26	1.03	10.31
3	8/23/2004	0.56	22.82	1.97	19.18	3.13	2.74	80.93	21.89	27.24	2.27	1.05	10.3
4	8/24/2004	0.57	22.87	1.95	18.97	3.19	2.62	80.98	21.67	27.24	2.21	1.01	10.33
5	8/25/2004	0.59	23.12	2.02	19.32	3.24	2.65	81.33	21.95	27.55	2.24	1.06	10.4
6	8/26/2004	0.62	23.08	2.01	19.17	3.23	2.7	80.97	21.77	27.44	2.18	1.06	10.23
7	8/27/2004	0.61	23.21	2	19.47	3.29	2.66	81.2	22.02	27.46	2.16	1.08	10.29
8	8/30/2004	0.61	22.78	1.92	19.03	3.29	2.55	80.69	21.6	27.3	2.06	1.04	10.11
9	8/31/2004	0.62	22.93	1.91	18.76	3.25	2.56	80.97	21.29	27.3	1.99	1.04	9.97
10	9/1/2004	0.64	22.91	1.91	19.09	3.33	2.51	80.52	21.43	27.39	2.04	1.06	10.05
11	9/2/2004	0.64	23.42	1.96	19.3	3.47	2.54	80.85	21.63	27.62	2.12	1.1	10.29
12	9/3/2004	0.63	23.26	1.94	18.75	3.44	2.5	80.68	20.05	27.11	2.05	1.07	10.03
13	9/7/2004	0.64	23.62	1.93	19.05	3.36	2.54	81.23	19.89	27.36	2.2	1.05	10.08
14	9/8/2004	0.65	23.92	1.9	19.31	3.31	2.56	82.08	19.72	27.26	2.15	1.03	9.86
15	9/9/2004	0.64	24.48	1.9	19.93	3.29	2.56	82.64	20.17	27.28	2.15	1.13	9.93
16	9/10/2004	0.64	24.93	1.93	20.46	3.56	2.64	82.94	20.57	27.49	2.13	1.17	10.46

December 1, 2023

Methods

Shapelet Model

The Shapelet machine learning model represents a specialized approach for time series classification. It operates in the way of identifying significant sub-patterns, known as shapelets, within the time series data. These shapelets serve as markers, allowing for the comparison and classification of various time series based on their similarity to these patterns. The model excels in capturing localized patterns within the time series data, making it particularly effective in situations where discriminative features are not uniformly distributed across the entire sequence. This capability positions the Shapelet model as a valuable tool for accurate predictions and classification tasks in scenarios characterized by uneven feature distribution.

- The explanation of the code usage is located in the section How to use the code.

Arima Model

ARIMA, short for "autoregressive integrated moving average," is a statistical and econometric model designed for the analysis of events occurring over time. This model proves invaluable in comprehending historical data trends and forecasting future data points within a series. Particularly suited for metrics recorded at regular intervals, ranging from fractions of a second to daily, weekly, or monthly periods, ARIMA leverages past values to predict future points in a given time series. Its versatility extends to applications such as time series forecasting and the prediction of future stock prices, making it a widely employed tool in diverse fields reliant on temporal data analysis.

- The explanation of the code usage is located in the section How to use the code.

Clustering Model

Clustering is the technique of dividing the population or data points into several groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is a collection of objects based on similarity and dissimilarity between them.

K-means is a data clustering approach for unsupervised machine learning. It can separate unlabeled data into a predetermined number of disjoint groups of equal variance – clusters – based on their similarities. It's a popular algorithm thanks to its ease of use and speed on large datasets.

- The explanation of the code usage is located in the section How to use the code

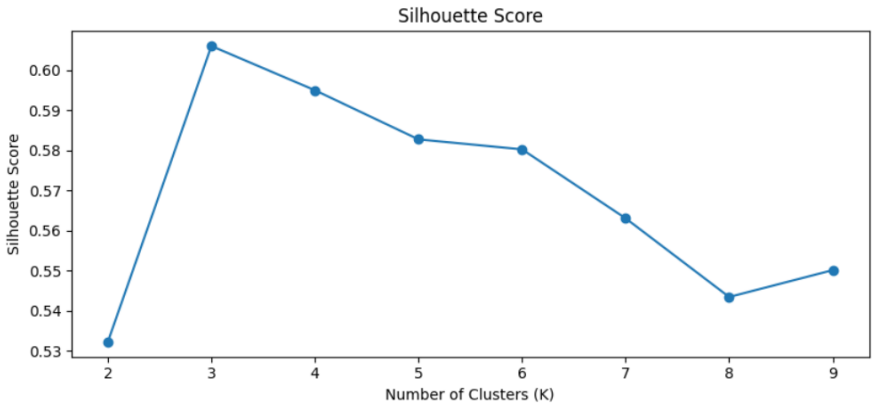
How to use the code:

Clustering Model

1. Change the ‘IBM’ for the company you want to see the stock predictions for.

```
# Extract the 'Price' column as the feature for clustering
prices = data['IBM'].values.reshape(-1, 1)

# Choose a range of potential values for K (number of clusters)
k_values = range(2, 10)  # Adjust the range as needed
inertia = []  # Sum of squared distances from data points to the centroids
silhouette_scores = []
```

2. Based on the Maximum Silhouette Score (figure 1), and change for the “optimal_k = value (figure 2).



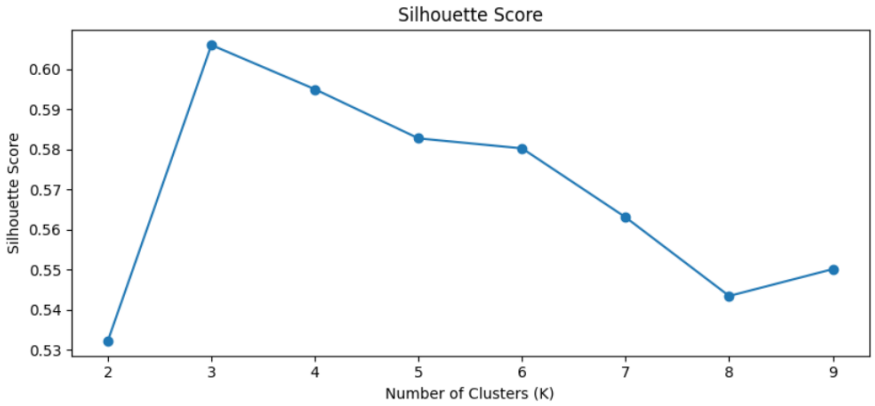

```
# Optimal number of clusters (K) based on maximum Silhouette Score
optimal_k = 3

# Create and fit the K-means model with the optimal K
kmeans = KMeans(n_clusters=optimal_k)
kmeans.fit(prices)
```

Shapelet Model

1. Change all the 'INTC' for the company you want to see the stock predictions.

```
[3]  # Load the dataset
     data = pd.read_csv('/content/final_dataset.csv')

     # Assume 'Date' is a datetime column
     data['Date'] = pd.to_datetime(data['Date'])

     # Sort the data by date
     data.sort_values('Date', inplace=True)

     # Use only 'Date' and 'IBM' columns
     data = data[['Date', 'INTC']]

     # Set 'Date' as the index
     data.set_index('Date', inplace=True)

     # Feature engineering: Creating a lag feature
     data['INTC_Lag1'] = data['INTC'].shift(1)

     # Drop missing values
     data.dropna(inplace=True)
```

2. Get the highest number from the Silhouette Score used for the same company on the Clustering Model. (figure 1) and change the “n_classes = 3” for the number (figure 2).




```
[5]  # Discretize the target variable for classification
     n_classes = 3  # You can adjust the number of classes based on your preference
     y_train_discrete = np.digitize(y_train, np.linspace(0, 1, n_classes + 1)[1:]) - 1
     y_test_discrete = np.digitize(y_test, np.linspace(0, 1, n_classes + 1)[1:]) - 1
```

3. Change the `'INTC'` for the same company you chose in the first step.

```
# Plot the original data and the predicted future direction as a bar plot
fig = go.Figure()

# Plot the original data
fig.add_trace(go.Scatter(x=test.index, y=test['INTC'], mode='lines', name='Original Data'))

# Create a bar plot for the predicted future values with different colors for increase and decrease
for timestamp, prediction, direction in zip(future_timestamps, future_predictions, directions):
    color = 'green' if direction == 1 else 'red'
    fig.add_trace(go.Bar(x=[timestamp], y=[prediction], marker_color=color, name='Predictions'))
```

ARIMA Model

1. Change the ‘IBM’ for the company you want to see the stock predictions for.

```
#Test for staionarity
df_close = stock_data['IBM']
def test_stationarity(timeseries):
    #Determing rolling statistics
    rolmean = timeseries.rolling(70).mean()
    rolstd = timeseries.rolling(100).std()
    #Plot rolling statistics:
    plt.figure(figsize=(15,9))
    plt.plot(timeseries, color='blue',label='Original')
    plt.plot(rolmean, color='red', label='Mean')
    plt.plot(rolstd, color='black', label = 'Std')
    plt.legend(loc='best')
    plt.title('Mean and Standard Deviation')
    plt.show(block=False)
test_stationarity(df_close)
```

2. Follow the results of the autoARIMA model, so you can find the best model.

```
# Model autoARIMA - Find the best model
model_autoARIMA = auto_arima(train_data, start_p=0, start_q=0,  # the auto-regressive (p) and moving average (q)
                      test='adf',       # Augmented Dickey-Fuller test - used to check for stationarity in time series data
                      max_p=3, max_q=3, # maximum p and q
                      m=1,              # frequency of series
                      d=None,           # let model determine 'd'
                      seasonal=False,   # No Seasonality
                      start_P=0,
                      D=0,
                      trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
print(model_autoARIMA.summary())
```

```
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=-20108.671, Time=0.31 sec
 ARIMA(1,1,3)(0,0,0)[0]             : AIC=-20123.341, Time=1.13 sec
 ARIMA(3,1,1)(0,0,0)[0]             : AIC=-20121.269, Time=0.52 sec
 ARIMA(3,1,3)(0,0,0)[0]             : AIC=-20133.352, Time=5.68 sec

Best model:  ARIMA(2,1,2)(0,0,0)[0]
Total fit time: 96.471 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 3860
Model:               SARIMAX(2, 1, 2)   Log Likelihood               10073.373
Date:                Sat, 02 Dec 2023   AIC                         -20136.747
Time:                        07:07:21   BIC                         -20105.456
Sample:                             0   HOIC                        -20125.635
```

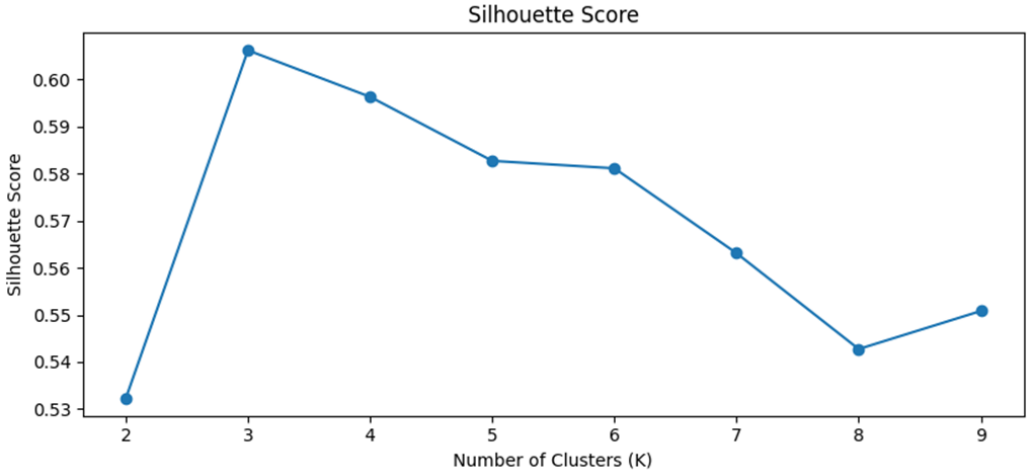
3. Initialize the ARIMA model with the above autoARIMA best model result.
4. Change the number of days (num_steps) you want to predict in the future.

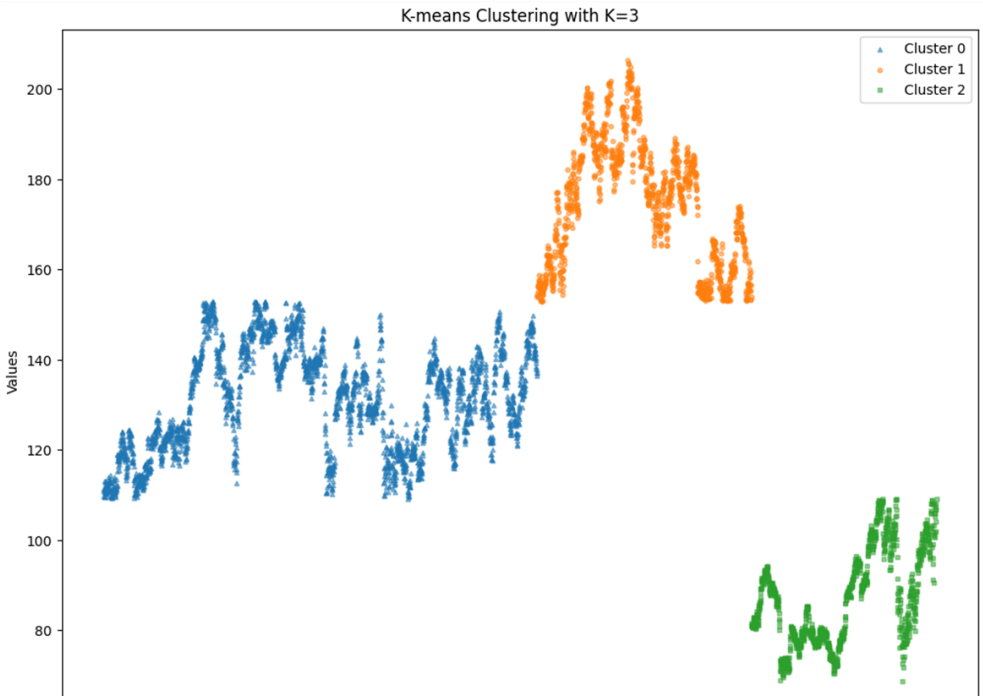
```
# Initialize and fit the ARIMA model
arima = ARIMA(test_data, order=(0, 1, 0))  # Adjust the differencing order
model = arima.fit()

# Calculate predictions for the next 5 years (approximately 1825 days)
num_steps = 966
predictions = []
```

Prediction Results

Clustering Model

According to the Silhouette Score, the optimal number of clusters (K) for the 'IBM' company' data, clustering is 3 (figure 1). The result of clustering is shown in figure 2, where we can see that data is clustered (grouped) by similar features, in this case - values. The lowest data has been captured as a Cluster 2 (green color) and it shows data with values 0 to 100. Cluster 0 (blue color) shows the data that falls into the value range of 100 to 150. The last data cluster is Cluster 1 (orange color) with a range of 150 and higher.





Shapelet Model

Our shapelet model predicts if future stock prices are going to increase or decrease. The way we imagined this prediction is to train and test the model into two categories: 0 (price will decrease) and 1 (price will increase). All future daily prices are predicted like that, if the price is going to increase tomorrow then the line will be green and if the price is decreasing the following day, it's going to be defined as a red line.

ARIMA Model

The ARIMA model is based on training, testing, and predicting future price movements based on historical data. In figure 1, we represented training data with a red line, testing data with a blue line, and predicted values with a green line for the next 2.5 years. Detailed price values are shown in figure 2, where each point is separately calculated and shown as daily stock value. In order to make a better visualization, we used bars with different colors to show if the future value is going to increase or decrease from today's value. The green color of the bars shows that the stock price is

going to increase for that specific day in the future, and the red color shows that it's going to decrease. Also, we create our plots to be dynamic, so by moving the mouse cursor it's possible to see the exact value for the specific future dates.

Conclusion

In conclusion, our commitment to enhancing investor decision-making in the stock market through advanced Machine Learning models remains steadfast. The significance of accurate stock price estimation cannot be overstated, and our approach, grounded in the analysis of historical data sequences, the power of sophisticated models to forecast future components with precision. However, the challenges inherent in stock price prediction, including data noise, market volatility, and unpredictable events, underscore the complexity of the task. Despite these challenges, our objective is clear: to develop a model that overcomes these obstacles and provides accurate closing daily stock price predictions.

As mentioned before, our approach to stock price prediction leverages three distinct models — Shapelet, ARIMA, and K-means clustering — to cater to different aspects of the complex forecasting landscape. The Shapelet model captures localized patterns, the ARIMA model leverages historical trends, and the clustering model offers insights into groupings based on similarity. Our dataset, spanning nearly two decades of daily closing stock prices for 12 tech companies, serves as a rich resource, ensuring the reliability and accessibility of our analyses. The efficacy of each model depends on the specific characteristics of the data and the nuances of the prediction task, emphasizing the need for a thoughtful and flexible approach in addressing the multifaceted challenges of stock price prediction.

As we move forward, the fusion of these models and methodologies represents a comprehensive approach to empower investors, traders, and financial analysts in navigating the dynamic stock market landscape.

Resources

https://www.scaler.com/topics/deep-learning/lstm-time-series/
https://medium.datadriveninvestor.com/time-series-prediction-with-lstm-f4a4cd0a5585
https://www.datacamp.com/tutorial/lstm-python-stock-market
https://github.com/Oliverdeb/time-series-analysis-ml-honours-thesis
https://www.analyticsvidhya.com/blog/2021/07/stock-market-forecasting-using-time-series-analysis-with-arima-model/