SS. Cyril and Methodius University - Skopje

Faculty of computer science and engineering

Forecasting "Amazon" Time Series Prediction Models for Stock prices

Course: Introduction to Data Science

Anastazija Kovachevikj

Code Link

Skopje, 2023

Contents

6.1. Introduction

Can we predict amazon stock prices? And, if so, how do models perform for predicting such an event? Our dataset contains information about Amazon's stock prices over the last four years. From this time series dataset, we will extract features that we will later use for training our predictors. Throughout the project, we will experiment with different prediction models, as well as various methods of dividing the data for those models. Using the previously mentioned data and the company's ESG score, the project's primary objective is to train and evaluate predictors.

6.2. Data preparation and analysis

The data for this project was obtained from yahoo! finance for dates spanning between January 1st 2020 through August 1st 2023. To get more familiar we should mention that the dataset is consisted of seven attributes:

- **Date -** date of trading day
- **Open –** first recorded price on that trading day
- **High -** highest recorded price on that trading day
- **Low –** lowest recorded price on that trading day
- **Close -** last recorded price on that trading day
- **Adj Close -** final price of the stock whilst considering external factors that might impact it
- **Volume -** total number of shares traded on that trading day and total activity and liquidity of the stock for that day

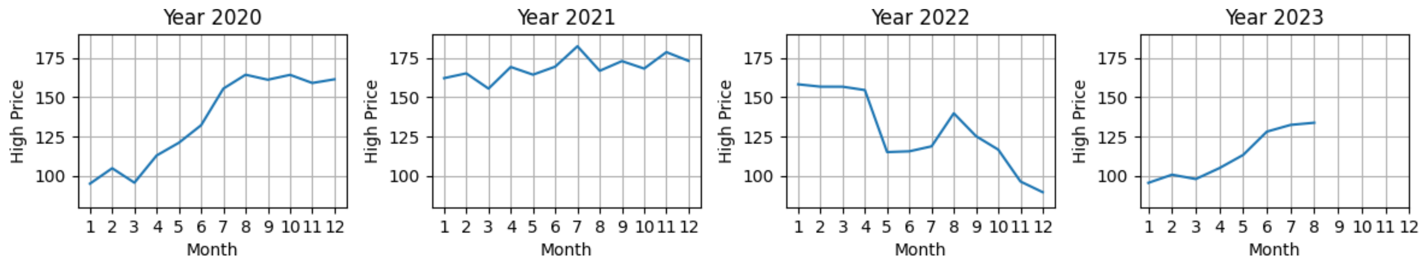


Figure 1: High Price value throughout the years

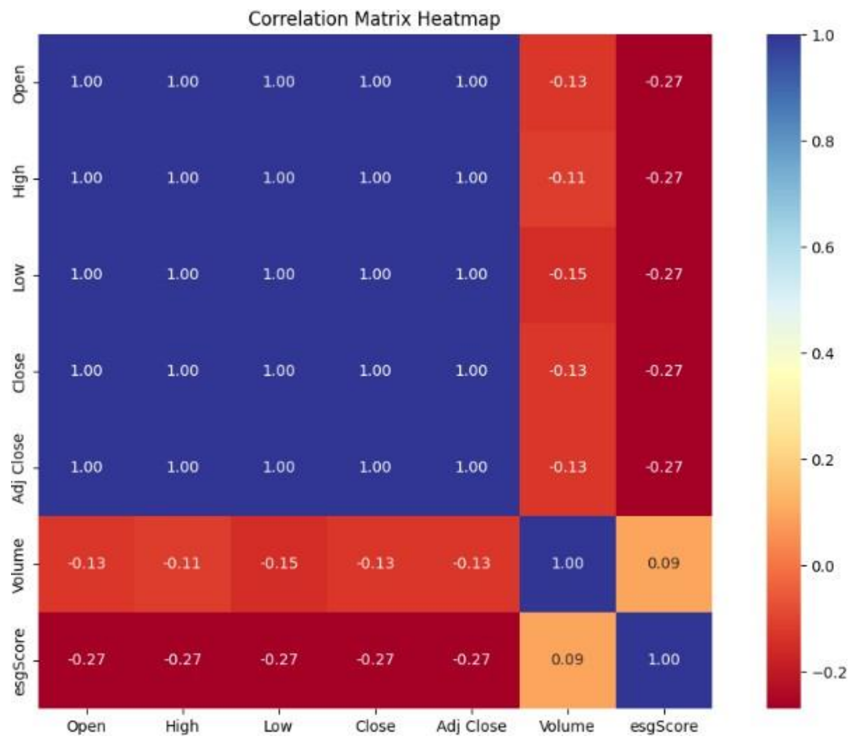
It is worth mentioning is that this dataset had no missing values, so there was no need for any imputation method.

The remaining data was collected from the Yahoo! Finance API for ESG scores. From this we had 4 extra columns, esgScore, governanceScore, socialScore and environmentScore. Unfortunately, the API had data only going up to August 1st 2022, which meant that we had to impute the other missing values. We decided to impute the missing values with the mean of all ESG scores after observing that the scores do not vary much over time. Additionally, we dropped all columns except esgScore, because they added no value to our prediction.

While analyzing the data, when dropping the attribute Date, we found that all attributes except from Volume and esgScore demonstrate a correlation coefficient of one. This rare correlation score indicates a perfect linear relationship between our attributes.

Such a finding suggests that the attributes are tightly connected and that each attribute consistently points to the same direction or follows identical trends as the others. This means that changes in one attribute are perfectly mirrored by corresponding changes in another. Also, one attribute's prediction is completely matched with a prediction of another.

Given that such a perfect correlation is an extremely rare occurrence, we should also ask ourselves if there are any outliers, sampling biases or patterns that we have missed.



	Open	High	Low	Close	Adj Close	Volume	esgScore
Open	1.00	1.00	1.00	1.00	1.00	-0.13	-0.27
High	1.00	1.00	1.00	1.00	1.00	-0.11	-0.27
Low	1.00	1.00	1.00	1.00	1.00	-0.15	-0.27
Close	1.00	1.00	1.00	1.00	1.00	-0.13	-0.27
Adj Close	1.00	1.00	1.00	1.00	1.00	-0.13	-0.27
Volume	-0.13	-0.11	-0.15	-0.13	-0.13	1.00	0.09
esgScore	-0.27	-0.27	-0.27	-0.27	-0.27	0.09	1.00

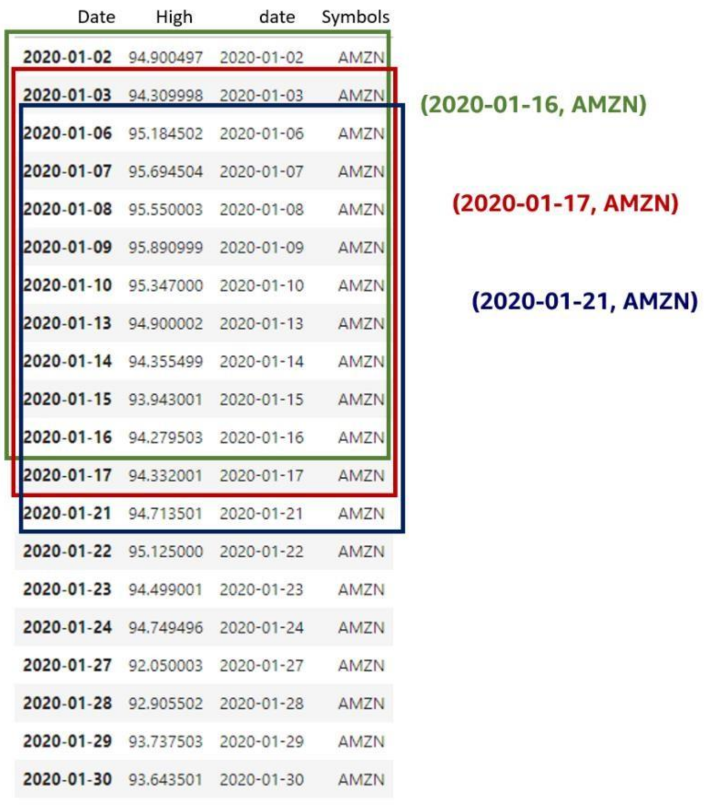
Figure 2: Correlation Matrix

6.3. Time series & Tsfresh

Because the dataset contains stock price information presented as time series data, we will now explain its concepts.
Time series data consists of observations recorded at specific time intervals, resulting in a chronologically ordered sequence of data points. In our case the stock prices are captured as a sequence of values over time which forms a time series.
Working with this type of data involves identifying and understanding patterns which usually requires specialized methods needed to capture and interpret the information accurately. For such purposes in our case, we will use the tsfresh library.

3.1 Tsfresh

We use the tsfresh library to implement rolling windows along with feature extraction. Rolling windows is a technique employed in time series analysis that involves moving a fixed-size window through the time series data. At each position of the window, feature extraction is performed to capture relevant characteristics within that window, meaning for each row it takes N past values. This method allows extraction of time-dependent patterns and trends that would otherwise be missed when looking at the complete time series.
The resulting extracted features from these rolling windows can then be used in various analytical tasks, such as developing prediction models or detecting specific occurrences within the time series.


(2020-01-16, AMZN)
(2020-01-17, AMZN)
(2020-01-21, AMZN)

Date	High	date	Symbols
2020-01-02	94.900497	2020-01-02	AMZN
2020-01-03	94.309998	2020-01-03	AMZN
2020-01-06	95.184502	2020-01-06	AMZN
2020-01-07	95.694504	2020-01-07	AMZN
2020-01-08	95.550003	2020-01-08	AMZN
2020-01-09	95.890999	2020-01-09	AMZN
2020-01-10	95.347000	2020-01-10	AMZN
2020-01-13	94.900002	2020-01-13	AMZN
2020-01-14	94.355499	2020-01-14	AMZN
2020-01-15	93.943001	2020-01-15	AMZN
2020-01-16	94.279503	2020-01-16	AMZN
2020-01-17	94.332001	2020-01-17	AMZN
2020-01-21	94.713501	2020-01-21	AMZN
2020-01-22	95.125000	2020-01-22	AMZN
2020-01-23	94.499001	2020-01-23	AMZN
2020-01-24	94.749496	2020-01-24	AMZN
2020-01-27	92.050003	2020-01-27	AMZN
2020-01-28	92.905502	2020-01-28	AMZN
2020-01-29	93.737503	2020-01-29	AMZN
2020-01-30	93.643501	2020-01-30	AMZN

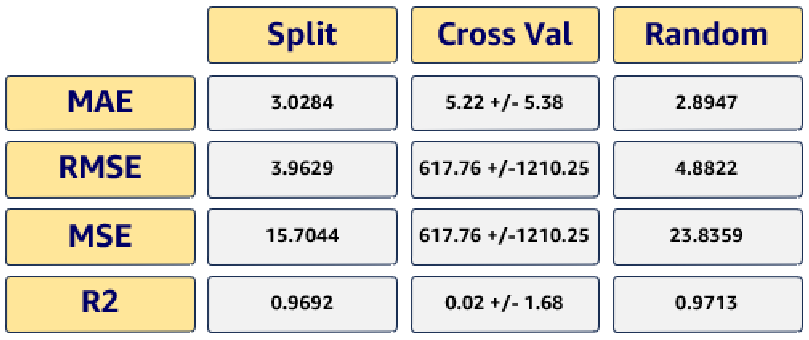
Figure 3: Rolling windows (with smaller window size)

6.4. Training Models

In the following section, we will take a closer look at the models and tools use for our data training. Specifically in our case we used LinearRegression, RandomForest and SVMs for regression (SVR). In the beginning we were unsure what is the most suitable approach for sampling the data because we wanted to choose one that would maintain representativeness while ensuring effective model training. Because of this we test 3 different ways of data splitting: split by years, cross-validation and random data split.

4.1 Linear Regression

For Linear Regression we used all three sampling techniques. First, we split data giving the train set all data for 2020 and 2021 and the test set is given all data for the 2022 and 2023. The second technique was cross-validation, which divides the dataset in multiple equal subsets. Then for each subset is used as a test set and the others are used as train sets. It selects a subset to be used for testing and all others are used for training. This is done for every subset we have; in our case it iterates five times. Lastly, for random sampling we split the data in N segments. Each segment is randomly split into train/test (ration 70-30), and then it is either appended to the final train or final test set. It is also worth mentioning that this random dataset is saved so that the same one is used for all other models.
After performing all previously mentioned methods we got the following results, shown in ***figure 4*.** From this, we can conclude that cross-validation is the least optimal method in terms of both accuracy and consistency when compared to the other techniques.



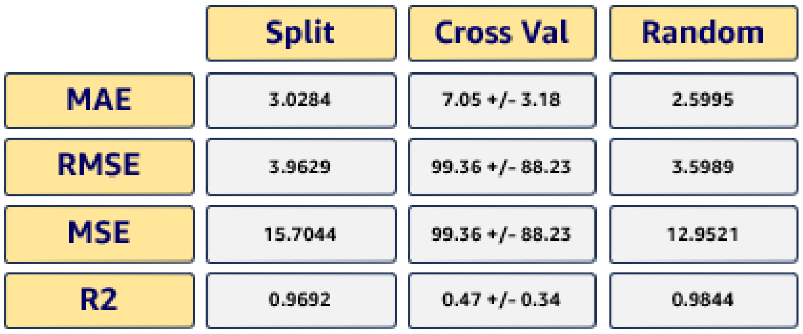
	Split	Cross Val	Random
MAE	3.0284	5.22 +/- 5.38	2.8947
RMSE	3.9629	617.76 +/-1210.25	4.8822
MSE	15.7044	617.76 +/-1210.25	23.8359
R2	0.9692	0.02 +/- 1.68	0.9713

Figure 4: Evaluation scores - Linear Regression

Next, we will talk about the second model we used and then we will discuss and compare the performance of both.

4.2 Random Forest

Random Forest is a machine learning technique that creates an ensemble of decision trees. It works by training multiple decision trees on different subsets of the data and combining their predictions for more accurate and reliable results. The randomness incorporated in the process prevents overfitting and allows the method to be used, in our case, for regression. For our model we used 100 estimators, which are essentially individual decision trees within the ensemble. We performed the three methods of sampling that we previously mentioned and got the results shown in ***figure 5***. Once again, the results for cross-validation showed the worst performance, so we decided that for the following models we will no longer sample data this way.



	Split	Cross Val	Random
MAE	3.0284	7.05 +/- 3.18	2.5995
RMSE	3.9629	99.36 +/- 88.23	3.5989
MSE	15.7044	99.36 +/- 88.23	12.9521
R2	0.9692	0.47 +/- 0.34	0.9844

Figure 5: Evaluation scores – Random Forest

4.3 Comparison of Linear Regression and Random Forest

If we look at figure 6 and figure 7, we can see a pattern of how each model and sampling technique performs. Cross-validation is not a good option for our predictors, as we have already determined, therefore we will now concentrate on the other two methods.

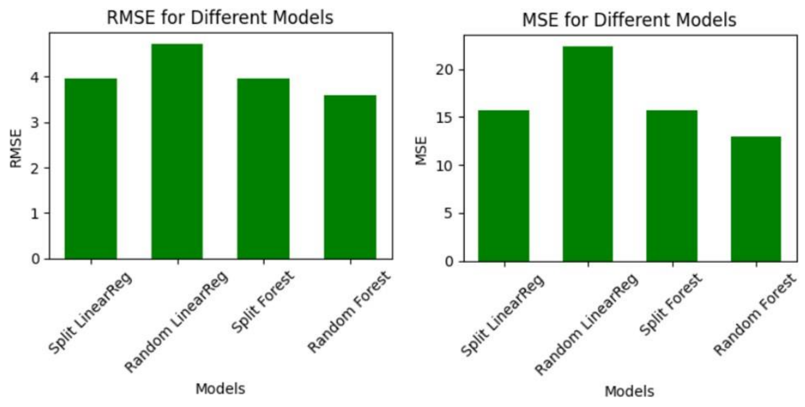


Figure 6: Histogram of RMSE and MSE

Although split by year sampling and random sampling initially appear to have similar results, in our opinion, split by year sampling is ineffective. For instance, if we choose a fixed date and the stock prices are rising exponentially, our model would never be trained with the highest prices, making it unrepresentative of all the data. In our opinion random sampling is the favorable option because it can deal with potential biases and ensure a more balanced representation of the data. Because of this we did the rest of the models only with random sampling in hopes of getting the best outcome.

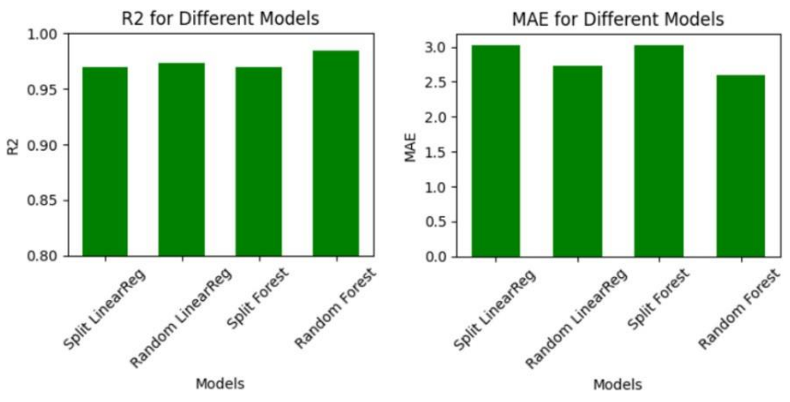
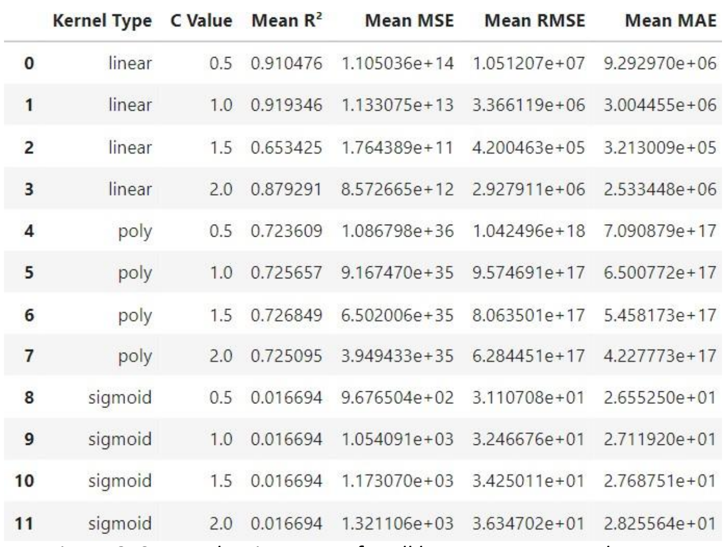


Figure 7: Histogram of R2 and MAE

4.4 SVRs

Support Vector Regression (SVR) is a machine learning algorithm that is used for regression tasks. It is based on the principles of Support Vector Machines (SVM), which were originally designed for classification. SVR applies SVM techniques to regression, where the goal is to predict continuous numerical values instead of categorical labels. We tried with different hyperparameter, kernel type an C value, which is the penalty constant for mispredictions. After iterating through all values for the hyperparameters we got the results shown in ***figure 8.*** The best R2 score is when we use linear kernel with C value of 1.0; nevertheless, despite the good r2 score all other metrics perform poorly.



	Kernel Type	C Value	Mean R²	Mean MSE	Mean RMSE	Mean MAE
0	linear	0.5	0.910476	1.105036e+14	1.051207e+07	9.292970e+06
1	linear	1.0	0.919346	1.133075e+13	3.366119e+06	3.004455e+06
2	linear	1.5	0.653425	1.764389e+11	4.200463e+05	3.213009e+05
3	linear	2.0	0.879291	8.572665e+12	2.927911e+06	2.533448e+06
4	poly	0.5	0.723609	1.086798e+36	1.042496e+18	7.090879e+17
5	poly	1.0	0.725657	9.167470e+35	9.574691e+17	6.500772e+17
6	poly	1.5	0.726849	6.502006e+35	8.063501e+17	5.458173e+17
7	poly	2.0	0.725095	3.949433e+35	6.284451e+17	4.227773e+17
8	sigmoid	0.5	0.016694	9.676504e+02	3.110708e+01	2.655250e+01
9	sigmoid	1.0	0.016694	1.054091e+03	3.246676e+01	2.711920e+01
10	sigmoid	1.5	0.016694	1.173070e+03	3.425011e+01	2.768751e+01
11	sigmoid	2.0	0.016694	1.321106e+03	3.634702e+01	2.825564e+01

Figure 8: SVR evaluation scores for all hyperparameter values

6.5. PCA

PCA is a dimensionality reduction technique used to reduce a high-dimensional dataset into a lower-dimensional space while maintaining as much relevant information as possible. PCA identifies principal components, which are linear combinations of the original features that capture the most variance in the data. By projecting data points onto these principal components, PCA reduces the data's complexity and helps visualize patterns.

For our data we wanted to see which numbers explained at least 90% of variance. We iterated through number of components starting from 1 up to 11 and got the results shown in figure 8. In the end we decided to continue with just 3 components because the difference in explained variance did not increase very much.

We then used the PCAdata to train our linear model and got the following results:

- Mean Squared Error (MSE): 54.2423
- Root Mean Squared Error (RMSE): 7.3649
- Mean Absolute Error (MAE): 5.8671
- R-squared (R^2): 0.9355

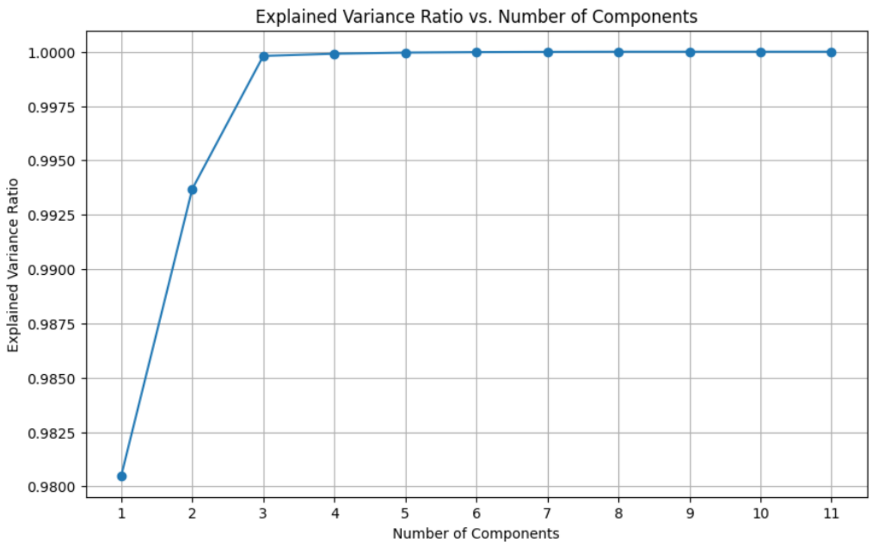


Figure 9: Explained Variance by each component

6.6. Conclusion

To summaries, all of this testing with numerous models has revealed that the random forest with random sampling has the greatest performance across all evaluation metrics. To improve our project, we would like to identify other variables that we can incorporate into our dataframe in addition to the stock price and esg score that will help the regression. Also we would like to explore more with use the of PCA in other models like Neural Networks or Naïve Bayes.