Forecasting a Stock Price - IBM Advanced Data Science Capstone Project.

by Wany Fourreau.

Model Deployment

Forecasting a Stock Price using Linear Models such as LinearRegression, K-Nearest Neighbor(KNN), Support Vector Machine(SVM) and using RNN-LSTM as a Time Series Forecasting Model.

Disclaimer: The reader is not encouraged to invest in the stock market or attempt to trade based upon anything that is said or any models that the author has discussed in the paper. The equity market is a risky place and randomness is the best guarantee. This is simply an attempt to understand Advanced Data Science Modeling Process.

Introduction

A short understanding of my interpretation of market psyche.

Why should we try to predict a stock price?
In an attempt to minimize risk and maximize profit investors and traders alike want to have an edge. Within that philosophy predictive models are created to serve as forecasting indicators which are assumed to provide that edge. In the financial world or to be more precise, at a trading firm, the term often used is "technical analysis" for predicting short term moves, for example in a stock, an index, a commodity or others. The term "Fundamental analysis" is used as well but it is usually applied to understanding the health of a company and for long term investments. Technical analysis is often used as short-term indicators to give the trader an edge. In my personal view even when a model does not work a trader wants to have one to follow because it makes the transaction appear less random. It also serves as a tool to keep emotions out of the trade. After all trading is an emotional affair but do pretend I did not tell you that since this is a scientific project for a data science class.

The Modeling Aspect - The Edge

Predictive models not always but are often applied as short-term indicators. Many models are available in Machine Learning which can serve as stock market forecasting tools. What we must understand is no models can take the place of good governance over money management. Here however in my Advanced Data Science Capstone Project I would like to see how a few Machine Learning predictive models work in comparison to a stock's actual price performance. My goal is to submit a Linear Regression model, a K-Nearest Neighbor model, a Support Vector Machine model and an RNN-LSTM Time Series Forecasting model using the company Amazon.com stock's historical price data.

An overall understanding about the models I have selected.
I will make the answers as short and as simple as possible. If you desire deeper input please go to wikipedia.com or a data science/machine learning website like Towardsdatasience.com or machinelearningmastery.com

What is a Linear Regression Model?
To give you a simple answer. Linear regression makes an attempt at modeling the relationship between two variables by fitting a linear equation to observed data where one variable is considered to be an explanatory variable and the other is considered to be a dependent variable. It is the most basic and commonly used type of predictive analysis.

What is a K-Nearest Neighbor (KNN) Model?
Medium.com put it in the simplest term possible. KNN is a supervised learning classification and regression algorithm that uses nearby points in order to generate a prediction. It is said to be one of the most basic yet essential classification algorithms in Machine Learning.

What is a Support Vector Machine(SVM) model?
Similar to the KNN model, SVM is a supervised machine learning model that can be used for classification and regression. SVM is effective in high dimensional spaces. It is effective in cases where the number of dimensions is greater than the number of samples. It is memory efficient and is versatile.

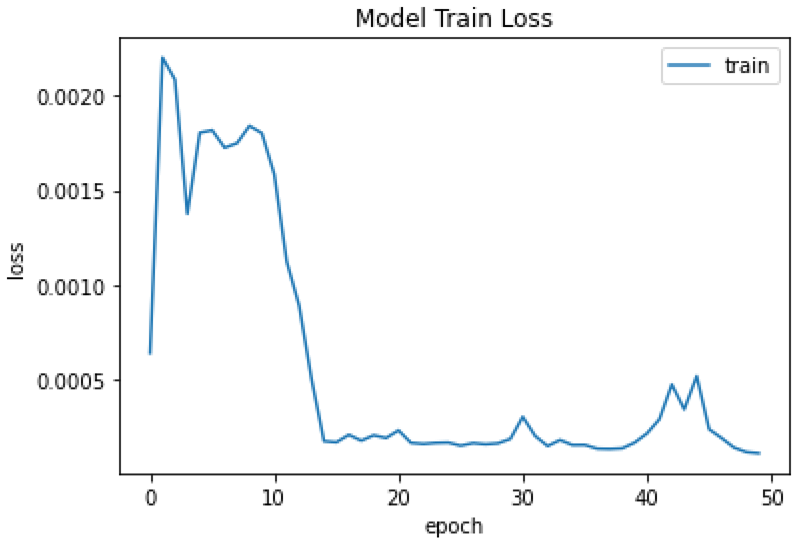
What is a RNN-LSTM Time Series Forecasting model?
RNN stands for recurrent neural network. LSTM stands for long short-term memory network. Since time series prediction problems are difficult types of predictive modeling problems, recurrent neural networks are used in deep learning to train very large architectures successfully.

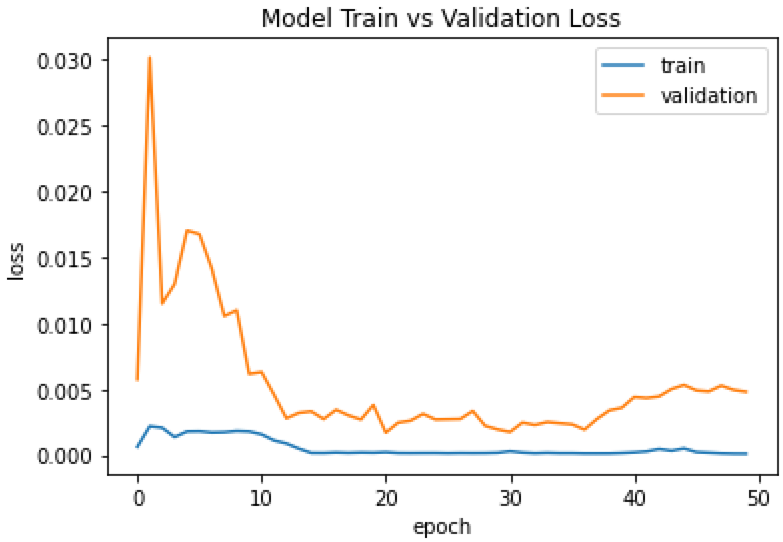
Model deployment.

I am recommending the RNN-LSTM model as one to deploy. I admit that it can be difficult to achieve precise predictions but by training a model that is not under-fitted or over-fitted, in my view this is a very impressive unsupervised model.

Originally when I first compiled the model based on 50 epochs it under-fitted as the plot below will show:

Post-training, plot the loss as:
Evaluation of how the model has trained over the epochs





It appears that the model has trained ok but there are a lot of ripples in the validation set. By the 40th epoch the validation and training loss over the epoch have reduced. However as I continued to rerunned the file to continue the work, the model has been taking longer to train and the loss over the epochs have been widening as you can see on the chart above.

To improve performance I decided to train the model with 100 epochs. Below is the result:

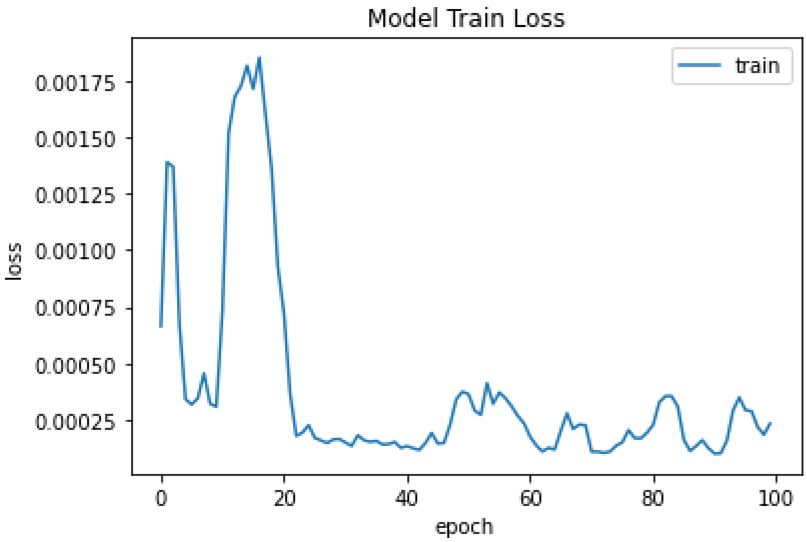
# Define the Lstm model for improved performance

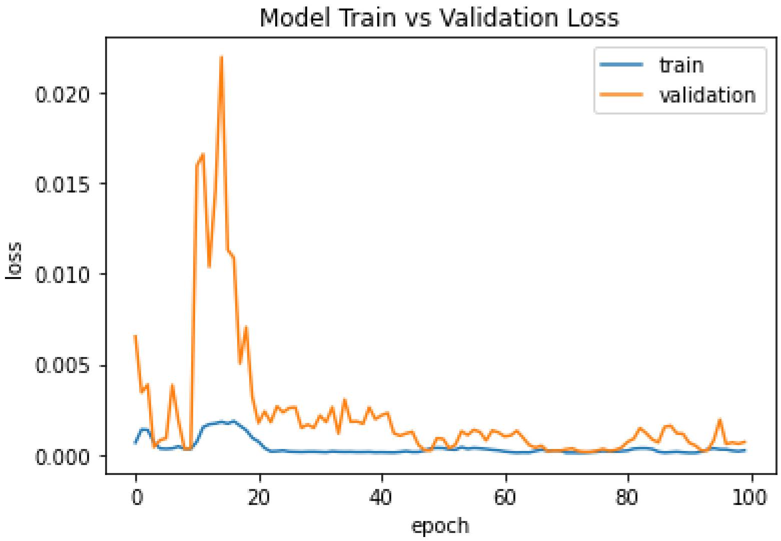
(Trying to fit the train & validation loss over epoch. Increased epochs from 50 to 100)

From the training data set I have written an automatic set up for a 20 percent validation split so later I could visualize how the losses fit over the epochs. After having done further research I increased the number of epochs to 100 from 50 hoping for a better fit. Look at the plots below.

Post-training, plot the loss as:

Evaluation of how the model has trained over the epochs. Now epoch equal to 100. Have the changes made a difference?

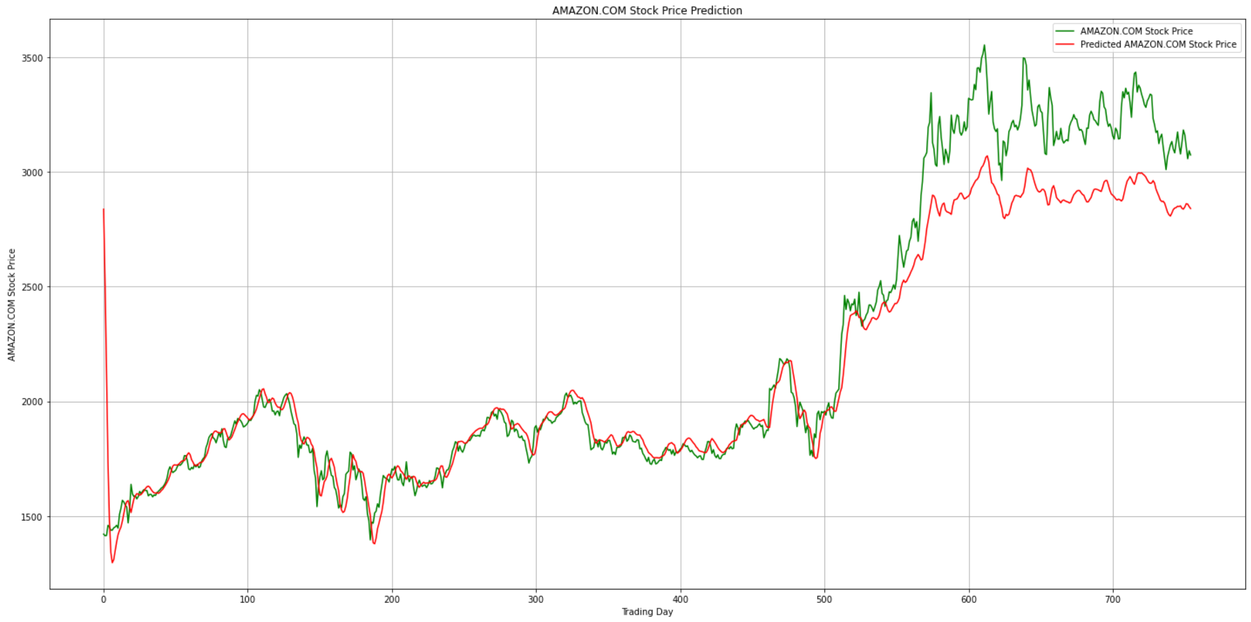




Yes the model has a better fit over the epoch. But can the model perform well against the test data set?

Model Evaluation & Visualizing the Results for the RNN-Lstm Models

Visualizing the results for the original model



Model Performance

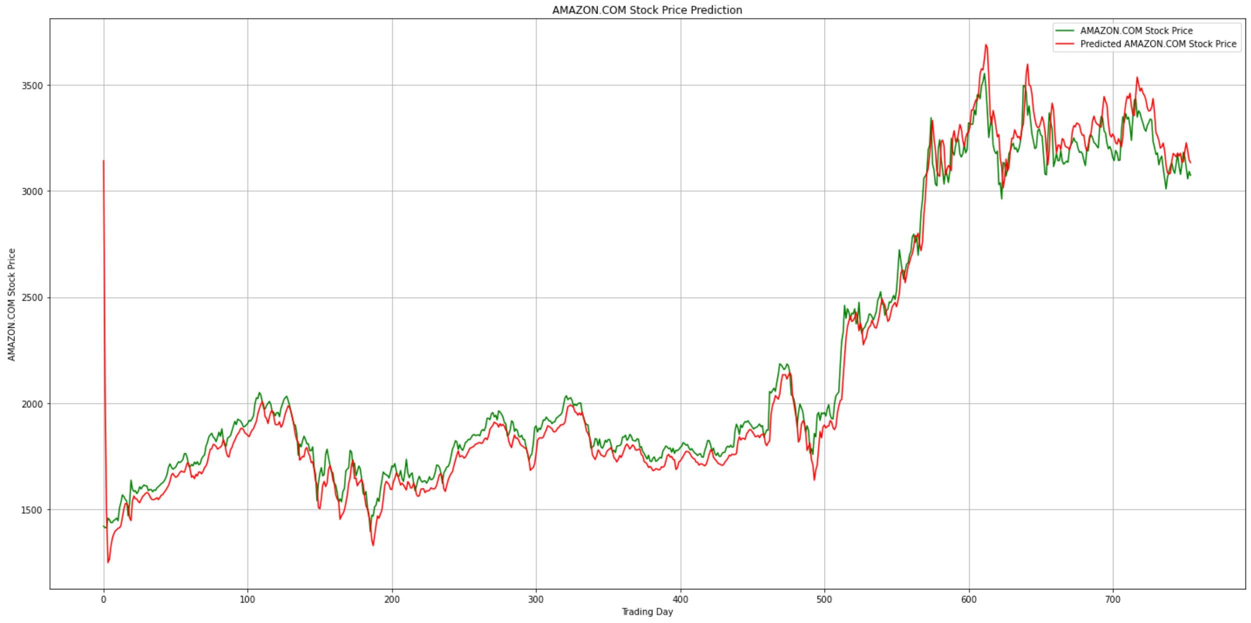
LSTM Model root mean square error 186.0728774117609

LSTM Model R2 score 0.911753314248435

LSTM Model Mean Absolute Error 116.32627308826572

Model Evaluation & Visualizing the Results for the RNN-Lstm Model Improved

Visualizing the results for the improved model



Looking at the visualization above the model's performance has improved.

Model Performance

LSTM Model root mean square error 104.88462191329695
LSTM Model R2 score 0.9719614280456601
LSTM Model Mean Absolute Error 68.07768667865273

In the chart of the original RNN-LSTM model the predicted stock price started to lag the actual stock price right after the covid-19 formal news in Mid-March. In the improved RNN-LSTM model the plot shows the model performed better after the covid-19 news.

To satisfy my curiosity and see how well the RNN-LSTM models will perform on more recent data. Below is the data and the charts of the redeployment:

Model redeployment.

Testing a New Never Seen Before Test Data on The RNN-LSTM-Time Series Forecasting Model and see the Result.

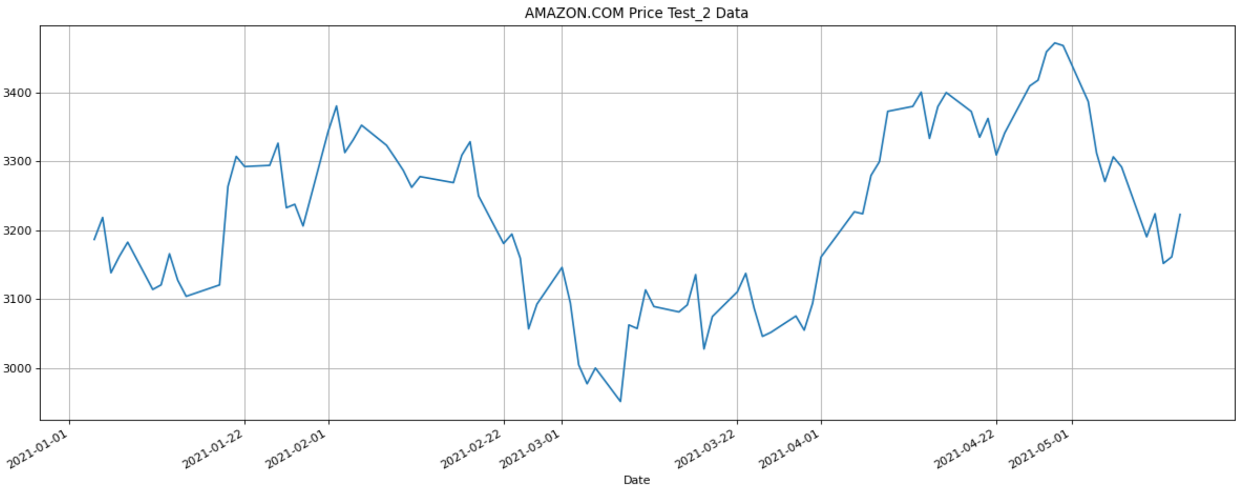
Model Deployment Test New

Here I would like to test an independent new set of data. Let us see how the model fairs. AMZN data from January 02, 2021 to May 15, 2021.

Prepare Test-2 data

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2021-01-04	3270.00000	3272.000000	3144.020020	3186.629883	4411400	0	0
2021-01-05	3166.01001	3223.379883	3165.060059	3218.510010	2655500	0	0
2021-01-06	3146.47998	3197.510010	3131.159912	3138.379883	4394800	0	0
2021-01-07	3157.00000	3208.540039	3155.000000	3162.159912	3514500	0	0
2021-01-08	3180.00000	3190.639893	3142.199951	3182.699951	3537700	0	0

(92, 7)

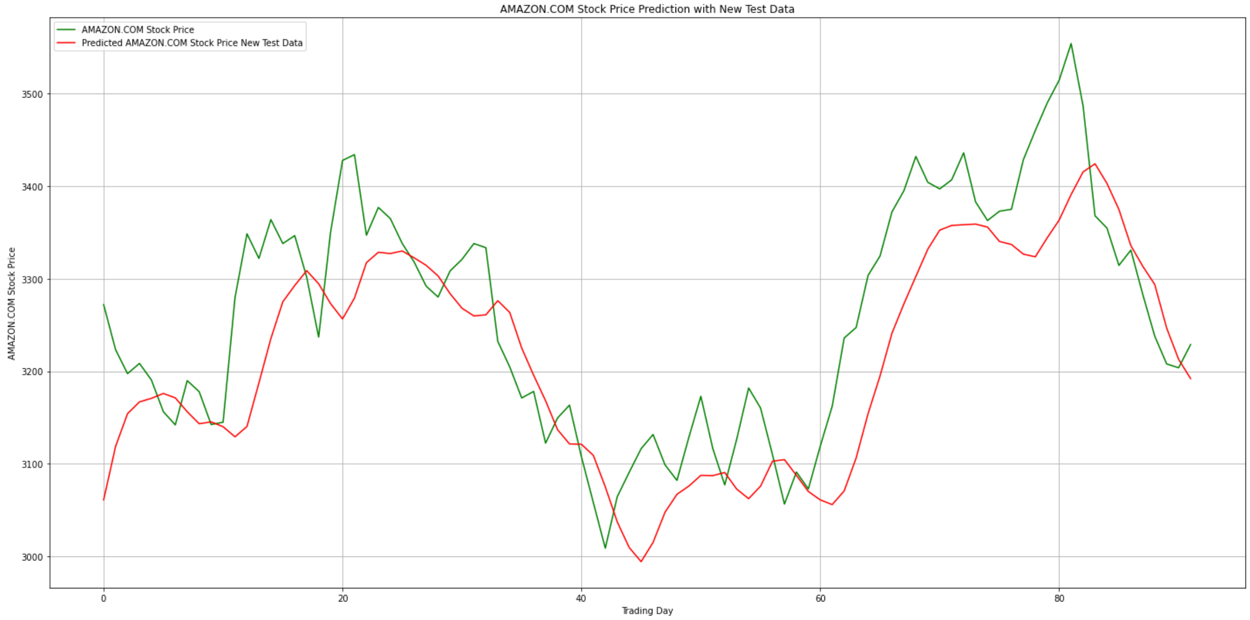


(92, 5)

Reshaping the data:
(92, 50, 1)

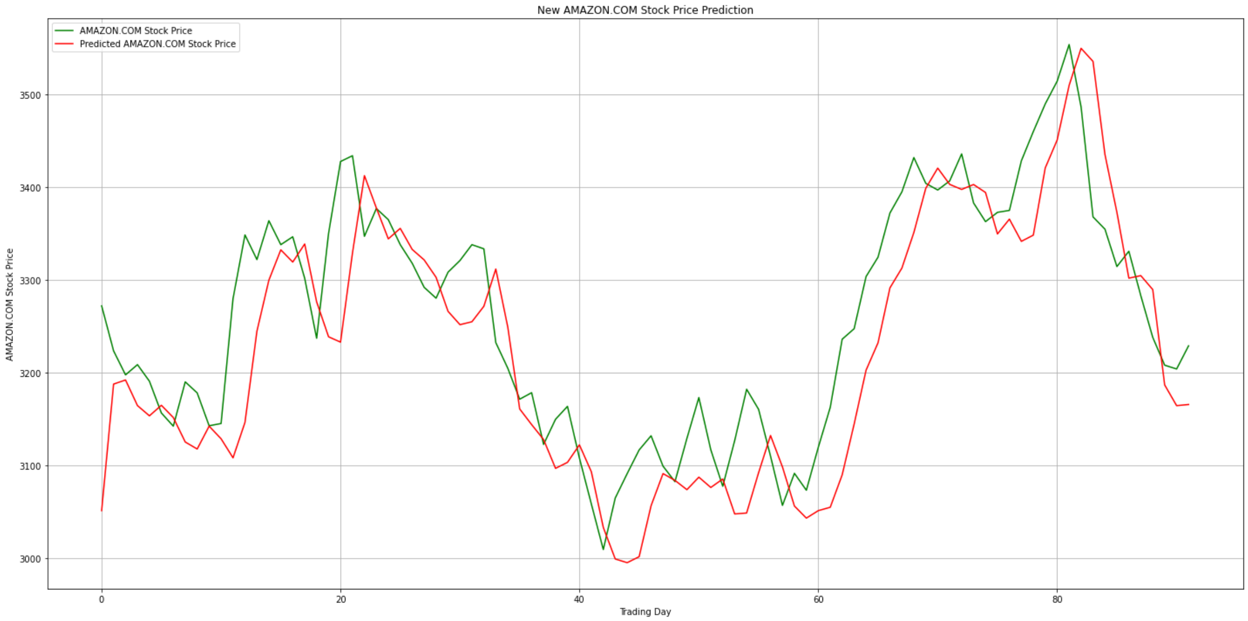
Making Prediction Using New Test Data

Visualizing The New Result Based on New Test Data - Original Model



Did I improve model performance?

Based on the visualization and the performance metrics Having a correctly fitted training model gave better results. See below.



Conclusion

As you can see above the models tested well with the new very recent test data. .

The End