Capstone Project Report

Ahmed A. Akl

Machine Learning Engineer Nanodegree

March 12, 2019

Definition

Project Overview

Online shopping has a vital role in our daily lives due to its low cost, highly convenience, ease of use, and so forth. Selling an item involves a trade-off between higher price and shorter selling time. One area of research that has received little attention is the time-series variation of the relative pricing and how the relative product pricing changes over time.

The term time series refers to a series of data-points that indexed in time order, in other words, a sequence of observations taken sequentially in time. Thus, time series forecasting is the process of using a model to predict future values based on previously observed values. Data used for time series prediction is non-stationary e.g. economic, weather, stock price, and retail sales.

Most existing price prediction models for selling second hand items only focus on selling items in specific domain. For example, several work have studied car price prediction [1-3] and predicted the end price of online auctions for laptop and PDA categories [4, 5]. While these efforts tried to make the best deal based on individual profit maximization, no model is designed with the aim of predicting an adequate price for a product. Furthermore, most price prediction models rely only on limited number of features or online product reviews/descriptions; they don't consider other main features like different prices of different product categories.

For time series analysis, a recent research utilized the sentiment analysis of news, txt, with the time series analysis to forecast the e-commerce prices [6]. The proposed work is evaluated for only Apple mobile phone and the forecasted prices are for a new product not second-hand product.

Also, there is a related work that uses dataset consists of both images and contextual information to predict the houses prices, "House price estimation from visual and textual features", https://arxiv.org/pdf/1609.08399.pdf

Problem Statement

The output of the time series prediction problems is a scalar value, which is applicable in problems such as weather or stock prices.

Most of online shopping website have second hand products for selling. If we pick a bunch of products for a specific type with the same model, we will find that this model varies in price. It depends on the quality of the product. Then, if time series model used to predict the prices, it will predict the same price for all of these products. But in reality, they have different prices.

In this project, we aim to use machine learning to predict a range of prices at a given time point to be closer to reality instead of predicting a single price.

To solve the problem, we will build a machine learning model to predict the minimum price and another one for the maximum. We used machine learning model, Linear Regression, and deep learning model, LSTM, and time series statistical model, ARIMA.

After finishing the project, we will have a machine learning model that able to predict a range of values.

Metrics

The problem being addressed is divided into two sub-problems:

1. **Time series problem**:

In this sub-problem, we are predicting the price range for the expected prices following time points in future. we changed the ads' prices to follow the price index curve so that the prices have a hidden equation. This hidden equation will be the metric to decide the degree the model successes. Our target is to fit this curve using our machine learning model.

This problem will be handled with these methods

- Deep learning method, where our evaluation metric is MSE.
- Regression method, where our evaluation metric is R-squared.
- Time series statistical model ARIMA, where our evaluation metric is MSE.

And finally, state the method with the best results

2. **Regression problem:**

In this sub-problem, we are predicting a specific price in the predicted range resulting from the time series problem, which will be our model quality indicator. As regression problem, I suggest to use R-squared metric as an evaluation metric for this problem.

Analysis

Data Exploration

To predict a range at a given time point, I used a dataset that contains ads of second-hand products. Each second-hand item has a description, an image, a product type, and a price.

There are different categories of the products, the list of products is in ***Figure 1***.

We used **price indices** for different product types to mimic the price change of products through time**,** which is a normalized average of price relatives for a given class of goods or services during a given interval of time. Depend on this standard price index, we changed the ads prices for each product of 210 time point. Then, we distributed that product type ads (items) over the time points, such that each time point has at least 20 ads. Each time point has a minimum price and maximum price. We changed the ads' prices to follow the price index curve so that the prices have a hidden equation. This hidden equation will be the metric to decide the degree the model successes. In other words, fitting the price index curve is the model target.

product type	No_of_observations
furniture	11795
electronics	8992
generalforsale	7776
sportinggoods	6822
car & truck	6477
householditems	6392
autoparts	6331
musicalinstruments	5646
bicycles	5560
motorcycles/scooters	5527
baby & kidstuff	5427
appliances	5400
clothing & accessories	4601
antiques	4543
collectibles	3881
tools	2784
computers	2567
arts & crafts	2035
toys & games	2033
photo/video	1651
computerparts	1085
motorcycleparts	1008
farm & garden	997
cellphones	836

Figure 1 - Dataset: list of product types with the number of ads at each type

This problem is univariate time series problem, as the main feature is product price.

If we picked a single product type e.g. sporting goods and get its statistical properties, as shown in ***Table 1***.

Table 1 - statistical features of electronics product

count	mean	std	min	25%	50%	75%	max
11760	124.747	19.488	84.47	108.282	126.942	140.455	155.453

We also provide a box plot of sporting goods, ***Figure 2***, shows the data skews to maximum as the median (yellow line) is near to the maximum.

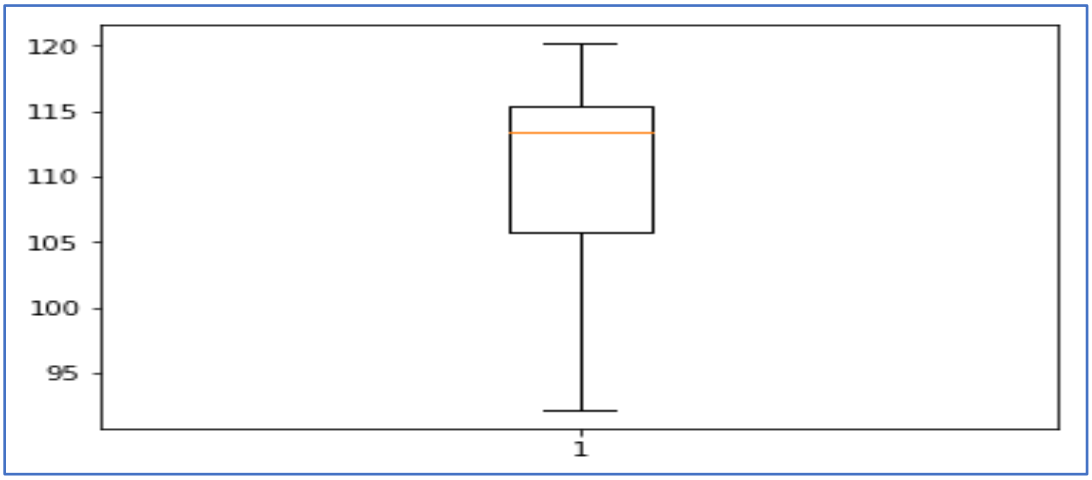

Figure 2 - Sporting Good Box Plot

During the process of downloading the data we deleted all null values to make the dataset more consistent.

Exploratory Visualization

In this section, we will discuss some of the characteristics of our dataset. We will show a sample of the dataset (one product) as it consists of more than product.

As shown in ***Figure 3***, the prices of the cars fall in a range of values. our target is to fit the minimum and the maximum curve of these values. The output of the fitting will be the range of values that we want to predict. X-axis represents the time point or the date and y-axis represent a range of prices of the used product.

In this figure, we show that the prices of the cars have upright trend which will be preprocessed in the preprocessing section to fit the statistical prediction models.

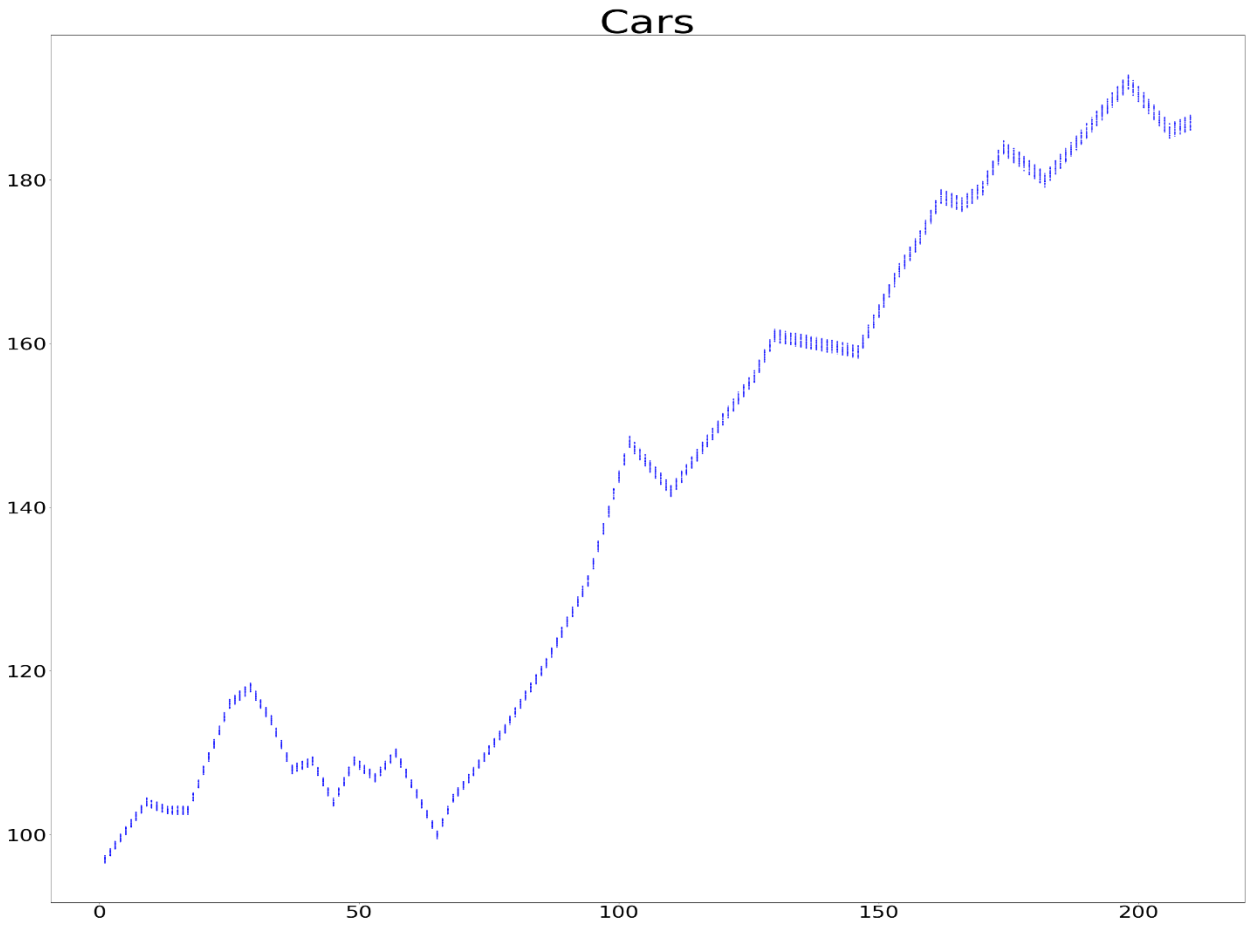

Figure 3 - Range of prices of Car Category

Algorithms and Techniques

The problem being addressed is divided into two sub-problems:

1. **Time series problem**:

In this part, the target is predicting price range for the expected prices following time points in future.

This problem will be handled with these methods

- **Deep Learning Method**,

  for DL method we use the most interesting deep learning model for sequence prediction, Recurrent Neural Network (RNN), especially the Long-Short Term Memory (LSTM) [7], which is able to learning long-term dependencies. LSTMs have the form of a chain of repeating modules of neural network, see **Figure 4**.

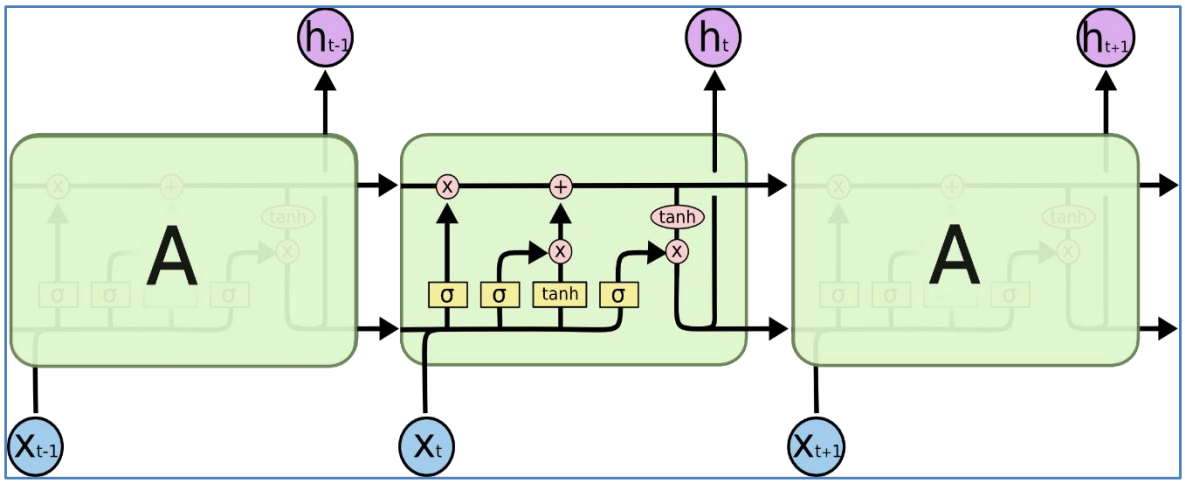

Figure 4- LSTM Architecture

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.

There are a set of hyperparameters must be set before start training LSTM network.

- Number of epochs (e.g. 100 epochs).
- Batch Size, number of time points for single training step.

Also, the network architecture consists of the following:

- Number of layers
- Layer types
- Layer parameters

- **Regression Method**,

  For the regression problem, we use the classical linear regression model. Linear regression is used to find a linear relationship between the target variable and one or more predictions. See the following equation,

$$Y = b_0 + b_1 X$$

  Where, Y is the predicted variable and X is the input variable b_0 and b_1 must be chosen so that they minimize the error.

- **Time Series Statistical Method**:

  For the time series statistical method, we used AutoRegression Integrated Moving Average (ARIMA) which is a class of statistical models for analyzing and forecasting

time series. ARIMA is a generalization of the simpler Autoregressive Moving Average by the adding the integration notion, where

AutoRegression (AR): model that uses the dependent relationship between an observation and some number of lagged observations.

Integrated (I): The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

Moving Average (MA): A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA has a set of parameters that should defined:

p: the number of the lagged observations.

d: the number of times the raw observations are differenced (degree of differencing).

q: the size of the moving average window.

2. **Regression problem:**

predicting a specific price in the predicted range resulting from the time series problem, which will be our model quality indicator. For this task we will use the standard linear regression.

Benchmark

To the best of our knowledge, we did not find a machine learning model that can predict a range of values.

Support Vector Regression model, which is an extension for Support Vector Classification (SVM) model. To test our model accuracy, we implemented support vector regression as a benchmark model to test the efficiency of our model. I test the model on our data and the results are stated, it shown in Justification section, as I compared it with ours.

Methodology

Data Preprocessing

Data preprocessing is an important task before starting our methods. In this project, we have to preprocess the data in two different methods to fit machine learning and statistical methods.

1- **Data Preprocessing for machine learning models:**

Before using machine learning, time series forecasting problems must be converted to supervised learning problems, from a sequence of points to pairs of input and output.

For this task we created a function that can convert a sequence to pairs of input and output.

For example: the following sequence is not suitable to for machine learning models.

Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Price	**59.383**	**60.592**	**60.753**	**61.744**	**62.285**	**64.868**	**65.953**	**66.629**	**68.798**

So that, the sequence is converted to be in this form as supervised pairs of data

X	**59.383**	**60.592**	**60.753**	**61.744**	**62.285**	**64.868**	**65.953**	**66.629**
Y	**60.592**	**60.753**	**61.744**	**62.285**	**64.868**	**65.953**	**66.629**	**68.798**

In this way the data is suitable for supervised learning.

Another preprocessing task, is **Normalization,** where the data is rescaled to fall in the range of (0,1).

2- Data preprocessing for statistical time series methods:

Before applying any statistical methods, you have to ensures that the data is stationary. Stationary data means that it has constant statistical properties over time, in other words, data has constant mean, constant variance and constant autocovariance over time.

For example, **Figure 5** is time series data for car & truck category.

Visually, we can see that the data has upright trend, which mean that the mean is not constant over time. But the implementation conducted some tests for data stationarity e.g. plotting rolling statistics (moving average) and Dickey and Fuller test.

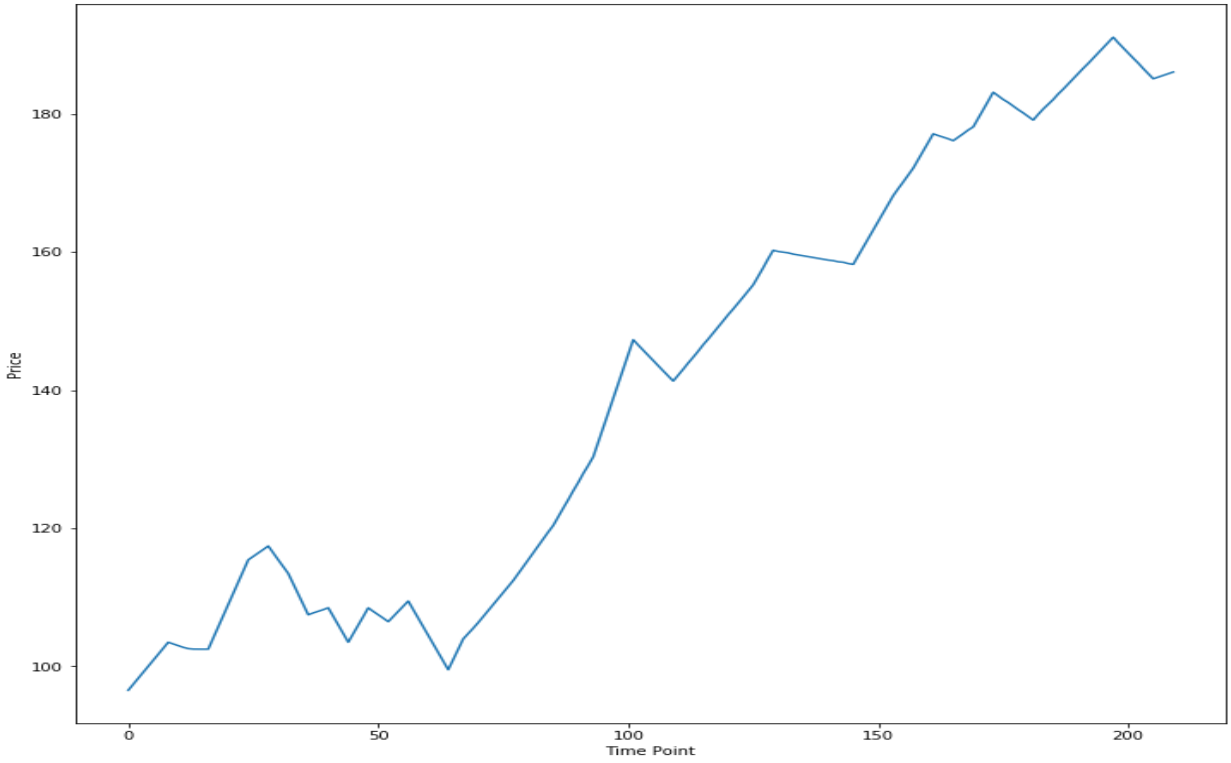

Figure 5- Cars data

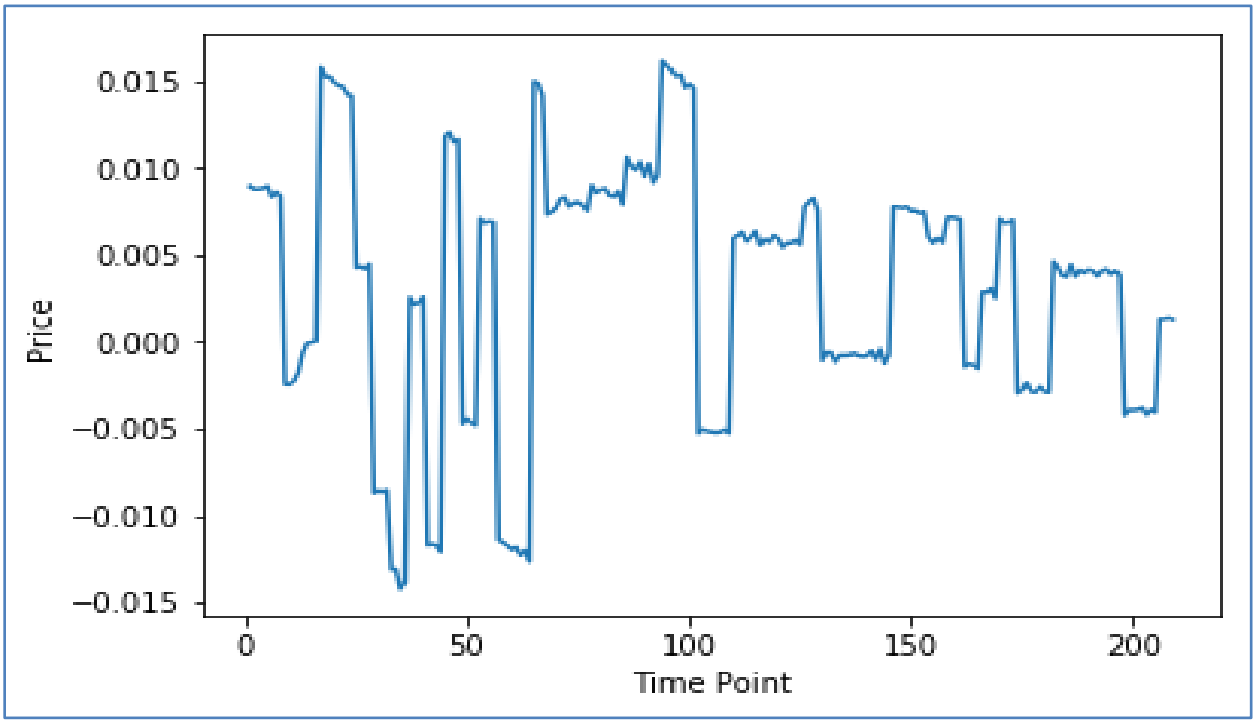

Figure 6 - Car Stationary Data

After testing the data stationary, we can see that the data is now stationary and suitable for applying statistical time series methods, as shown in **Figure 6**.

Implementation

In the implementation process, we used Jupyter notebook, a web application that allow creating and sharing documents that contain live code, equations and visualizations. We implemented all the models in Python 3.5, Keras, with TensorFlow backend, for deep learning models, scikit-learn library for machine learning models and statsmodels package for statistical models.

The learning process followed the following sequence.

1. The preprocessed dataset is divided into two datasets, the training set and the test set with percent 0.8 and 0.2 respectively.
2. We choose one-layer LSTM network with 4 units, and we trained it on the training data then test it performance on the testing data, then we adjust the number of units until we get the best value, as shown in Refinement section. We did the same with the other models ARIMA and linear regression models.
3. As our problem is a regression problem, where the output will be a real value, we will use Root Mean Square Error. **Root Mean Square Error**, is an estimator that measures the average of the squares of the errors, that is, the average squared difference between the estimated values and the given ones. The smaller the RMSE, the closer we are to finding the line of best fit.
4. The process of learning a model is divided into two steps: **The training process,** where the models are trained using the preprocessed training data, learn to fit the index price curve. **The testing process,** we test the model with unseen data, to measure the model performance and its capability to generalize for new data, which referred as the testing step.

One of the challenges that encounter me is in the implementation process is implementing ARIMA model. As it needs a lot of test and preprocessing to make the date suitable for training the model. For this process, I created a separate *Time_series_stationary_preprocessing.ipynb* to test the data stationarity.

Refinement

Optimizing machine learning model is an important task.

For example, *look-back* variable, or number of lagged values used for predicting the target variable, has a great impact on the predictions but increasing the variable value may cause overfitting.

In Linear Regression model, the initial state was *look-back = 1*, the model Root Mean Square Error (RMSE) for testing is 0.90. Then we finetuned the value to be 5, the RMSE is 0.47.

We also, tried some variations of Long – Short Term Memory (LSTM) network, as specified in the *LSTM.ipynb*, to get the best results that fit our problem. LSTM network has an important variable, the number of units in the LSTM layer, we started with 4 then increased the number and finally we got the best value to be 10 as it achieves the best RMSE.

Results

Model Evaluation and Validation

In this section, we will discuss our model and the final parameters used.

To evaluate a machine learning model a validation set used, where the target is validating the model to unseen or new data. Based on this data we finetune and adjust the model hyperparameters. The final architecture and hyperparameters were chosen because they performed the best among the tried combinations.

In case of LSTM, our final choice of LSTM variants is LSTM as regression achieves the best RMSE.

The final architecture of LSTM is:

- One LSTM layer with 10 units
- One output layer

The parameters of the LSTM are:

- Batch size = 1
- Loss function: mean square error
- Learning rate optimizer is Adam
- Epochs = 100

To help us understand the performance and the robustness of our forecasts, we compare predicted prices to real prices of the time series. As shown in ***Figure 7***, the model able to predict prices (green) that are too close to the real prices (blue). Base on this inference, we prove that the model is robust to be extended to more difficult tasks.

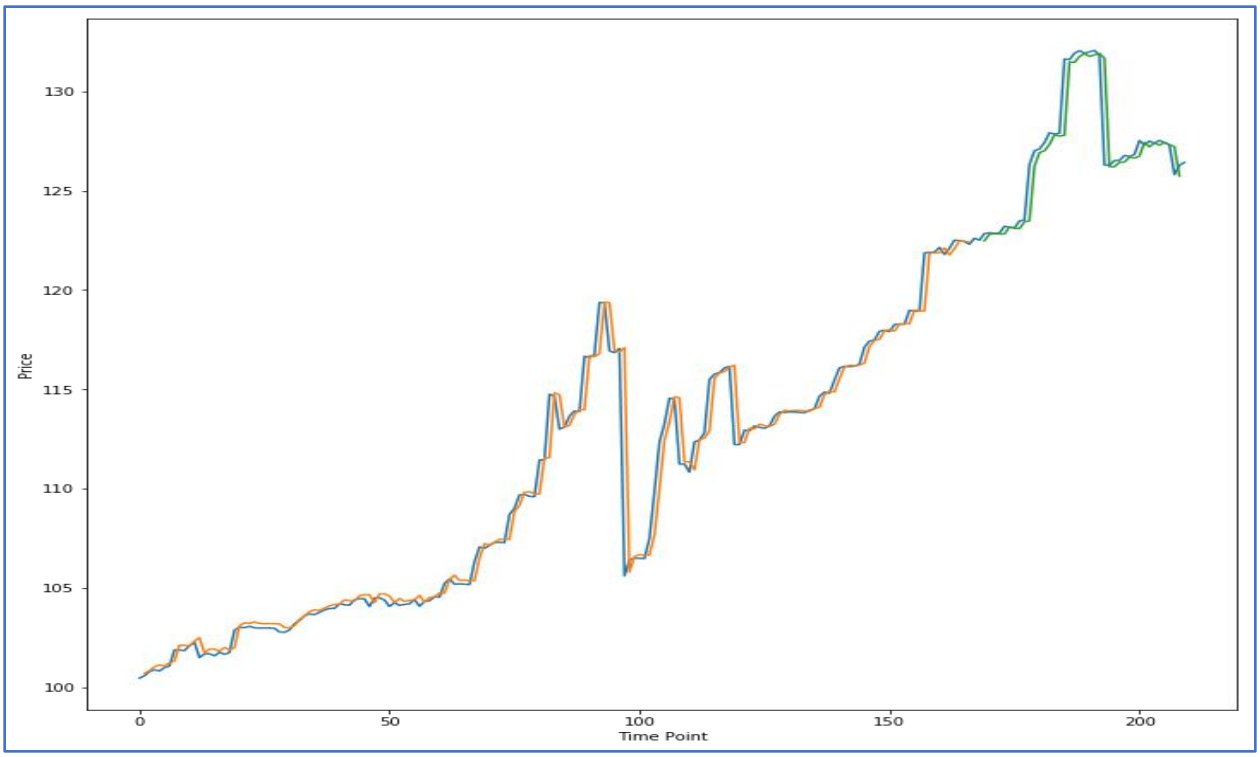

Figure 7 - antiques prediction accuracy

Justification

In this section, we will compare the results of our model with the benchmark model. In our experiments, we conducted the experiments with Linear Regression, LSTM, and ARIMA statistical model. The results of our model are better than the benchmark model, Support Vector Regression.

As shown in ***Figure 8***, the SVR is not able to fit the prices curve or even close to it. On the other hand, our models e.g. Linear Regression model has better performance in fitting the curve as shown in ***Figure 9***. The predicted prices of our model are close to the real prices. And the results of LSTM are much better in fitting the curve of prices. From these results, we can ensure that the model is robust to predict a range of prices, by predicting the minimum and the maximum correctly.

In summary, the application is useful in predicting the range of prices, but to solve the bigger and more complicated problems. The time series problems are complicated when the curve is seasonal or the problem is multivariant, where multiple variables are measured over time.

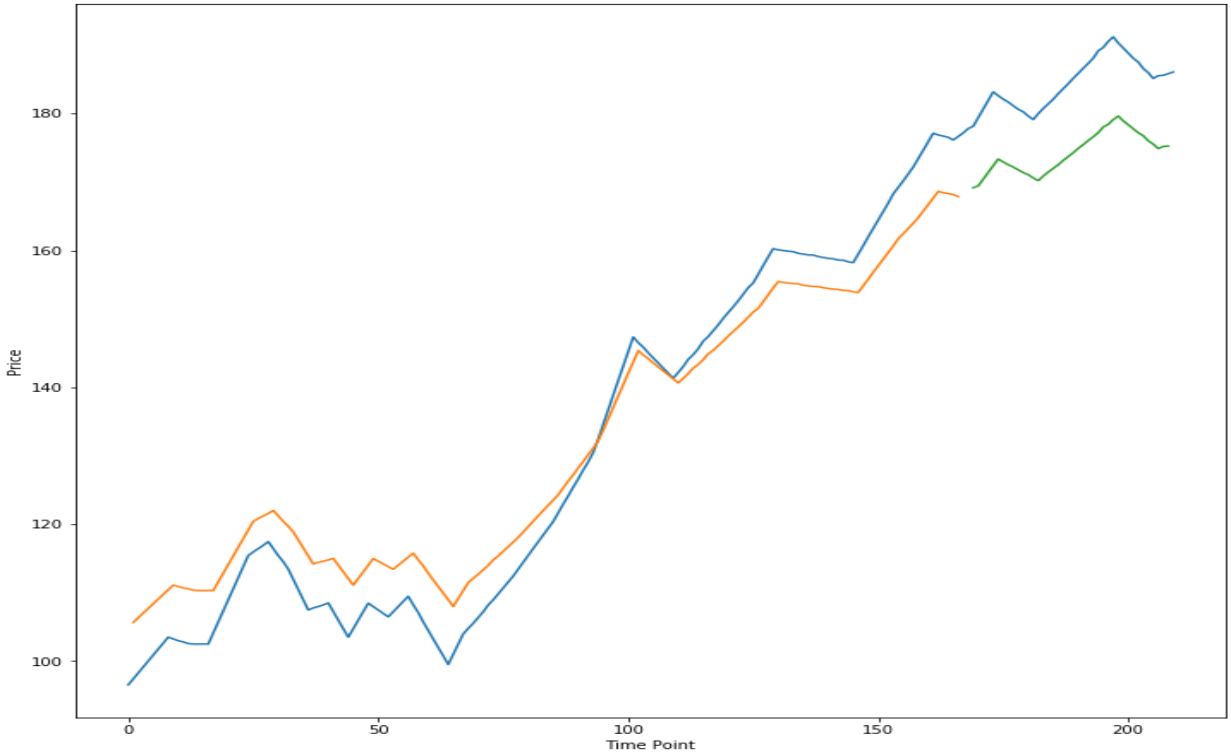

Figure 8 - SVR accuracy

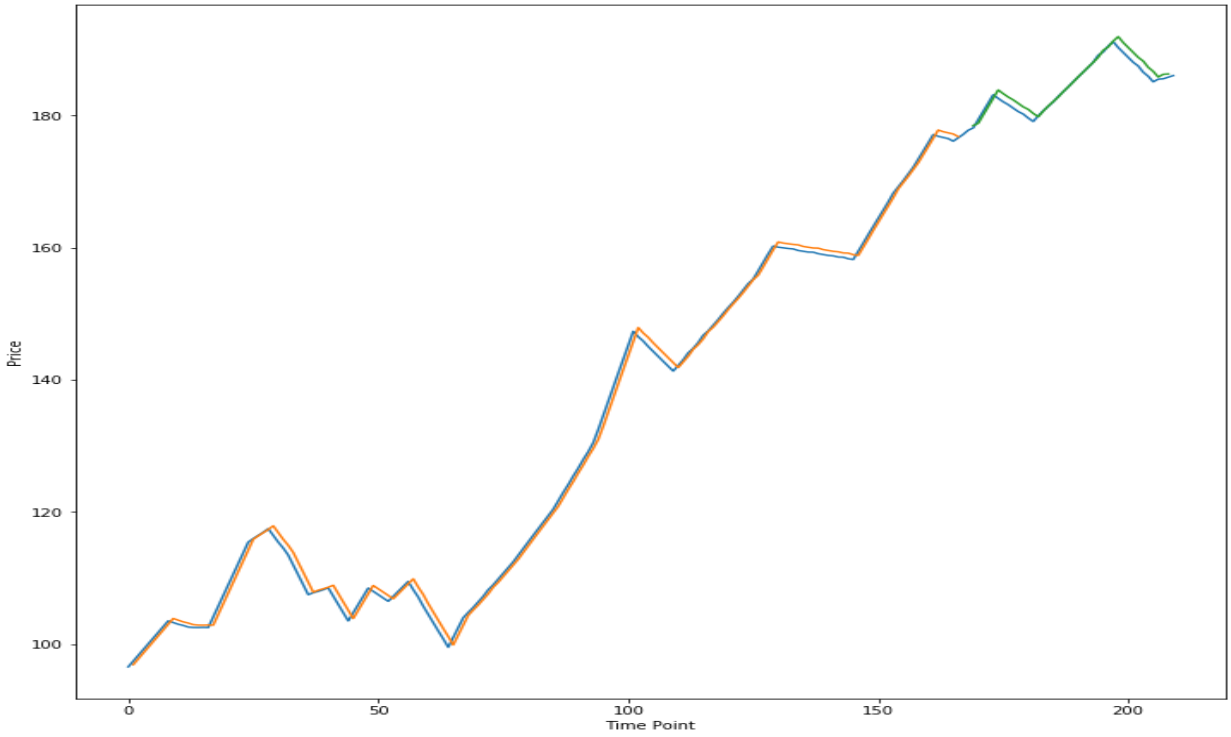

Figure 9 - Linear Regression Accuracy

Conclusion

Free-Form Visualization

In this section, we will show how is our model achieves better results.

To test the performance of our model, we chose a much harder curve. As shown in ***Figure 10***, the model able to fit and output prices that are close to the real prices.

The line plot (green line) is showing the observed values compared to the rolling forecast predictions. Overall, our forecasts align with the true values very well, showing an upward trend and downward trend of the prices.

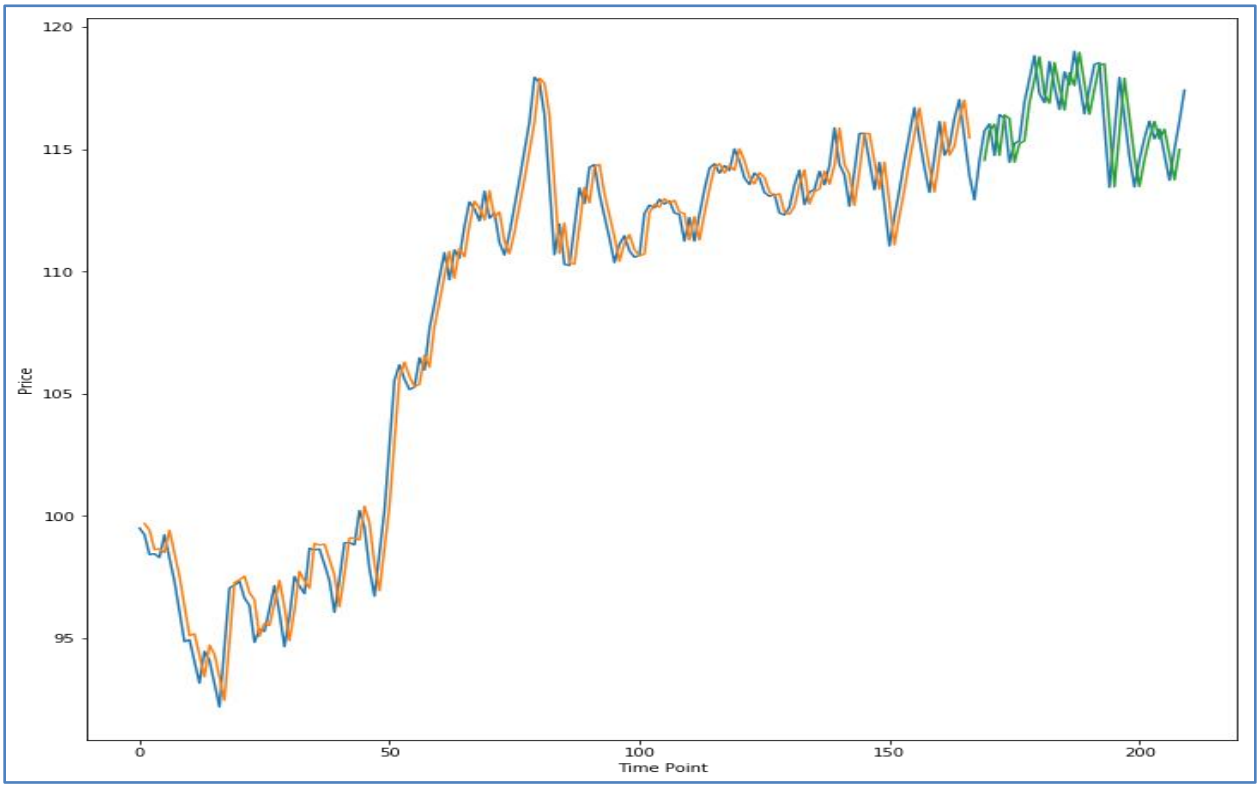

Figure 10 - Prediction Accuracy

Reflection

The process used for this project can be summarized using the following steps:

1. The problem is defined as we want to predict a range of valued for used products.

2. A dataset of used products is download and preprocessed as the prices of the products will follow the price index, clean it from null or duplicate values, as discussed in Data Exploration section.
3. We implemented a benchmark model, run it on the dataset, and the results are addressed.
4. Our models trained on the dataset, the process is repeated until the best results are obtained.

The dataset collection and adjusting is a difficult task and takes a lot of time to make it applicable for experiments. Dataset preparation is the most important part in machine learning field. Also, the implementation of statistical models such as ARIMA is hard and took a lot of time to adjust the data to it rather than the machine learning models.

Improvement

Predicting a range of prices is not an easy task. We used a simple method where we can predict the minimum and the maximum prices. But I think that there is a much intuitive methods that able to predict a range of values not just the max or min.

Also, the prices in the dataset are simple, just increasing or decreasing, where the models fitted it and achieve a good performance. I think more difficult dataset of prices will be interesting.

[1] N. Pal, P. Arora, D. Sundararaman, P. Kohli, and S. S. Palakurthy, "How much is my car worth? A methodology for predicting used cars prices using Random Forest," *arXiv preprint arXiv:1711.06970,* 2017.

[2] X. Zhang, Z. Zhang, and C. Qiu, "Model of Predicting the Price Range of Used Car," 2017.

[3] K. Noor and S. Jan, "Vehicle Price Prediction System using Machine Learning Techniques," *International Journal of Computer Applications,* vol. 167, 2017.

[4] R. Ghani, "Price prediction and insurance for online auctions," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 411-418.

[5] I. Raykhel and D. Ventura, "Real-time Automatic Price Prediction for eBay Online Trading," in *IAAI*, 2009.

[6] R. F.-Y. L. Kuo-Kun Tseng, Hongfu Zhou, Kevin Jati Kurniajaya, Qianyu Li, "Price prediction of e-commerce products through Internet sentiment analysis," *Electron Commer Res,* vol. October, 2017.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, pp. 1735-1780, 1997.