

# 4033/5033: Final Project Report (4 Page)

Munkhbileg Munkhburen

## 1 Introduction

For this project I will work on time series data. One of the reason why I am really interested in Machine learning is to learn how to predict time series data. It makes me feel like I can see the future. Time series data is always in our everyday life. Weather, stock price, heart rate monitoring etc. In our normal supervised machine learning we have input and outputs. Time series data inputs are date and outputs are results. I always wondered about how to turn dates in input and predict future value. I will explore more about how to prepare our data for time series. In this project I chose apple's stock price data to experiment. The reason why I chose stock price is related with my finance background. While I was take stock trading strategy classes, I always wondered about how to use machine learning for stock price data.

When I talked with professor Chao Lan, he suggested that auto regressive is very good for time series data prediction. Also I will try to use normal regression for my machine learning model, then I will discuss about what is best way to predict time series data. We did not learned much about deep learning, but LSTM is one of the best way to predict the time series data.

## 2 Methodology

Autoregression is my main method for this project. Autoregression uses previous outputs to forecast future value. So the output from previous dates becomes the input for the autoregression model to predict next outputs. I will use mean squared error to measure the accuracy. Autoregression uses linear relationship same as linear regression. Autocorrelation and PACF is also used in model predict.

In our case we used Apple stock price data for Autoregression. For example lets divide out data for year to year. Each year apple releases iPhone during September and that is where Apple stock rises very high for the year and decreases eventually. We want to predict Apple's next year stock price using previous years results. We will need to find PACF for each year and will choose ones with high correlations for current years. If it have low correlation we don't want to include it. Here is the formula

$$x_n = \beta_0 + \beta_1 * x_{n-1} + \beta_2 * x_{n-2} \dots + \epsilon$$

$x_n$  is our current year and  $\beta$ 's are our correlation then it is multiplied by that years  $x$  which is value and  $\epsilon$  is the error

So above formula helps use to predict Apple's stock price using previous years. In real life we usually want to predict tomorrows value with high accuracy to make profit. So I chose each day as my period for my project.

## 3 Experiment

### 3.1 Data Preparation

I collected data using pandas framework called datareader. I collected Apple's stock from yahoo finance using DataReader.

```
In [73]: start = dt.datetime(2012,1,1)
end = dt.datetime(2021, 12, 12)
df = web.DataReader("AAPL", 'yahoo', start, end)
print(df.head())
```

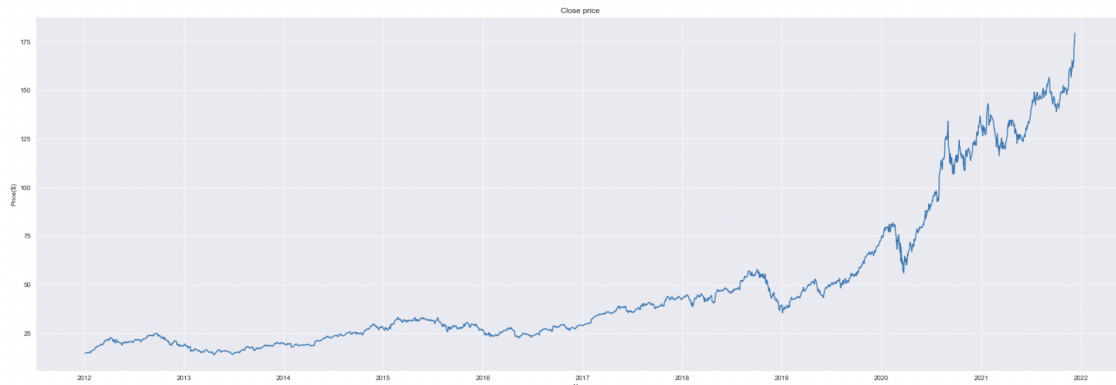
	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	14.732143	14.607143	14.621429	14.686786	302220800.0	12.591936
2012-01-04	14.810000	14.617143	14.642857	14.765714	260022000.0	12.659609
2012-01-05	14.948214	14.738214	14.819643	14.929643	271269600.0	12.800155
2012-01-06	15.098214	14.972143	14.991786	15.085714	318292800.0	12.933966
2012-01-09	15.276786	15.048214	15.196429	15.061786	394024400.0	12.913449

```
In [74]: df.describe()
```

```
Out [74]:
```

	High	Low	Open	Close	Volume	Adj Close
count	2503.000000	2503.000000	2503.000000	2503.000000	2.503000e+03	2503.000000
mean	48.810695	47.815294	48.306437	48.334997	2.157001e+08	46.565568
std	38.065806	37.189724	37.614968	37.656990	1.761320e+08	38.283345
min	14.271429	13.753571	13.856071	13.947500	4.100000e+07	12.134607
25%	23.874286	23.392500	23.643750	23.619107	1.033646e+08	21.155704
50%	32.532501	32.052502	32.240002	32.220001	1.520820e+08	29.395185
75%	53.958750	52.846251	53.412500	53.317499	2.660248e+08	52.008738
max	179.630005	174.690002	175.210007	179.449997	1.506120e+09	179.449997

I have Apple's stock price data from 2012/01/01 to 2021/12/12. Also I need only closing price as output. Then I will drop other columns. Here is my visualization and final data set prepared.



```
In [76]: df.drop("High", axis=1, inplace=True)
df.drop("Open", axis=1, inplace=True)
df.drop("Volume", axis=1, inplace=True)
df.drop("Adj Close", axis=1, inplace=True)
df.drop("Low", axis=1, inplace=True)
```

```
In [77]: df.head()
```

```
Out [77]:
```

	Close
Date	
2012-01-03	14.686786
2012-01-04	14.765714
2012-01-05	14.929643
2012-01-06	15.085714
2012-01-09	15.061786

## 3.2 Experiment Design

I used AutoReg from statsmodels to predict future value in Apple's stock price.

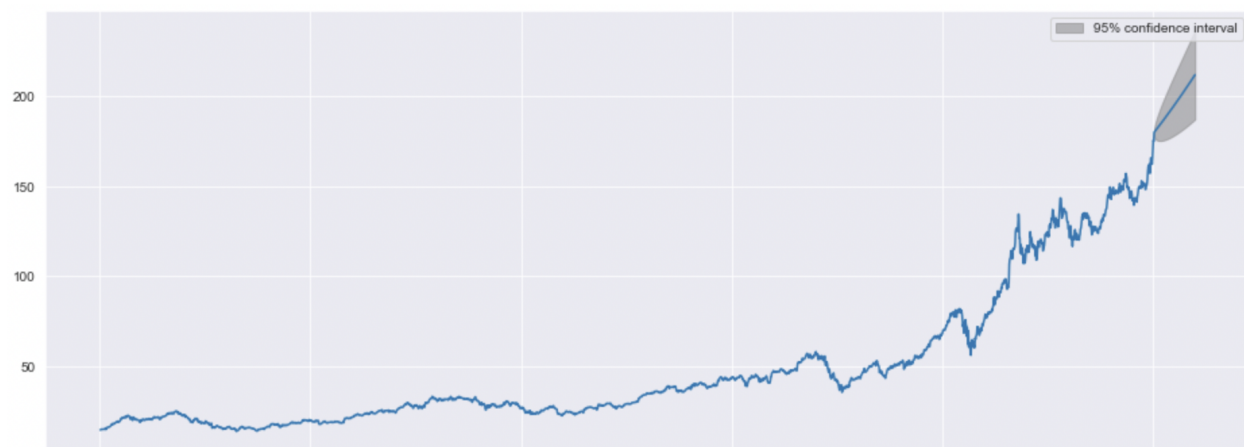


Fig. Apple Stock in Autoregression from Statsmodels

The grey area and straight line is my predicted future value of Apple's stock. This did not give me accurate enough prediction. But future value will be in gray are for .95 probability.

Usually time series data used past data to predict tomorrow's value. Instead of predicting just 1 value, I used 97 percent of data for training and 3 percent for testing. Then I used Autoregressive integrated moving average for calculating and measuring the accuracy.

Out[126]: [`matplotlib.lines.Line2D` at `0x7fce4a7a3d00`]



Fig. Apple Stock in Arima from pmdarima

Green line is my predicted and orange is given data. The mean squared error for this case is 137.

I chose my period for each and my testing value should be just tomorrow's value. But I chose more value for visualizing. When I chose my period for each day, previous day have very high correlation than others. So closer the day closer the correlation. So if my stock is decreasing for 3 days in a row, Autoregression will predict it will decrease forever. In above graph, my stock price was increasing for few months, so it predicted my stock price increase forever.

### 3.3 Results and Discussion

Autoregression work very good with time series data, but it is not well suited for stock price prediction. Autoregression have linear relationship. Stock price will never be linear. We need to find out when is price rising and when is price dropping. While I was experimenting, I made all the data for training input except the most recent one. Then I used Autoregression for predicting today's price using all the previous prices. I had very low mean squared error, but price difference in prediction was more than 3 dollars. Even 3 dollar matters lot in stock exchange when trading with high amount. When price follows linear relationship, Autoregression can capture it very well and predict accurately. If it is not linear we are not lucky.

Then I researched which method would be best for stock price data. Then I found LSTM which is deep learning method. I experimented little bit with it to visualize the prediction. I will deep dive into LSTM for next semester when I take advanced machine learning. It requires RNN which used in field of Deep learning.

**Out[70]: (18000, 19000)**



Fig. Apple Stock in LSTM

LSTM method gave me much more accurate prediction than Autoregression models.

My purpose of this project was to learn how to predict time series data using Autoregression. I just chose stock price for fun. Autoregression performs very well in linear data. In Apple stock prices there were small sections that follows linear relationship. When I do autoregression on those sections I almost got 0 mean squared error. If I have linear data set autoregression performs really well. Such as predicting number of times "All I want for Christmas is you" will be played this Christmas is very easy with Autoregression.

## References

- 1.) Lazzeri, Francesca. Machine Learning for Time Series Forecasting with Python. Wiley, 2021.
- 2.) AUFFARTH, BEN. MACHINE LEARNING FOR TIME-SERIES WITH PYTHON: Forecast, Predict, and Detect Anomalies with... State-of-the-Art Machine Learning Methods. PACKT PUBLISHING LIMITED, 2021.
- 3.) Jansen, Stefan. Machine Learning for Algorithmic Trading: Predictive Models to Extract Signals from Market and Alternative Data for Systematic Trading Strategies with Python. Packt Publishing, 2020.