Stock Market Forecasting

By Dinesh Pandey

This document explains about different methods that can be carried out or implemented.

I have included two source code and two CSVs in the same folder.

1) Data preparation, data pre-processing, ARIMA model and LSTM (Univariate model) (Source code)
2) Feature addition, LSTM (multi variate model) (Source Code)
3) CSV file with future close return that shows the missing data's (for observation).
4) Data count on each Ticker (for observation)

Please note that all the codes are labelled.

Tasks

Data processing

In this task, I combined all the provided data into one single file. Change the date into datetime format. I created data frame with 7 columns: "Tickers", "Date", "Open", "High", "Low", "Close", "Volume".

Step 1

I created **five** separate data frames ("Open", "High", "Low", "Close", "Volume". Moreover, I indexed each data frame by Date. **(Code Provided)**

Sample Output (df_close)

```
In [31]: df_close.head()
```

Out[31]:

Tickers	1AD	1AG	1AL	1PG	1ST	3DM	3DP	3PL	4CE	4DS	...	ZNO	ZNT	ZNZ	ZOZI	ZRL	ZTA	ZUSD	ZYB	ZYL
Date																				
2015-01-02	NaN	NaN	NaN	1.215	NaN	NaN	NaN	2.26	NaN	NaN	...	NaN	NaN	NaN	NaN	0.067	NaN	NaN	NaN	NaN
2015-01-05	NaN	NaN	NaN	1.265	NaN	NaN	NaN	2.07	NaN	NaN	...	NaN	NaN	4.40	NaN	0.070	NaN	NaN	NaN	NaN
2015-01-06	NaN	NaN	NaN	1.245	NaN	NaN	NaN	2.08	NaN	NaN	...	NaN	NaN	4.39	NaN	NaN	NaN	NaN	NaN	NaN
2015-01-07	NaN	NaN	NaN	1.250	NaN	NaN	NaN	2.05	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.003	NaN	NaN	NaN
2015-01-08	NaN	NaN	NaN	1.255	NaN	NaN	NaN	2.08	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 2773 columns

Step 2

In the task, I created two separate data frames (close return and future close return) **(Code Provided)**

$r_{t-1,\,t}$ (close return) = P_t / P_{t-1}

$r_{t,\,t+1}$ (future close return) = P_{t+1} / P_t

Sample Output (df_cr)

```
df_fcr= df_future_close_return-1

In [34]: df_cr.head()
```

Out[34]:

Tickers	1AD	1AG	1AL	1PG	1ST	3DM	3DP	3PL	4CE	4DS	...	ZNO	ZNT	ZNZ	ZOZI	ZRL
Date																
2015-01-02	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
2015-01-05	NaN	NaN	NaN	-0.958848	NaN	NaN	NaN	-1.084071	NaN	NaN	...	NaN	NaN	NaN	NaN	-0.955224
2015-01-06	NaN	NaN	NaN	-1.015810	NaN	NaN	NaN	-0.995169	NaN	NaN	...	NaN	NaN	-1.002273	NaN	NaN
2015-01-07	NaN	NaN	NaN	-0.995984	NaN	NaN	NaN	-1.014423	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
2015-01-08	NaN	NaN	NaN	-0.996000	NaN	NaN	NaN	-0.985366	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN

5 rows × 2773 columns

Step 3

I created a new data frame containing the ratio High/Low for each ticker each day in the same format. **(Code Provided)**

Sample Output

```
In [41]: df_hl_ratio.head()
```

Out[41]:

Tickers	1AD	1AG	1AL	1PG	1ST	3DM	3DP	3PL	4CE	4DS	...	ZNO	ZNT	ZNZ	ZOZI	ZRL	ZTA	ZUS
Date																		
2015-01-02	NaN	NaN	NaN	1.068966	NaN	NaN	NaN	1.055300	NaN	NaN	...	NaN	NaN	NaN	NaN	1.0	NaN	Na
2015-01-05	NaN	NaN	NaN	1.061224	NaN	NaN	NaN	1.111111	NaN	NaN	...	NaN	NaN	1.022727	NaN	1.0	NaN	Na
2015-01-06	NaN	NaN	NaN	1.041322	NaN	NaN	NaN	1.039604	NaN	NaN	...	NaN	NaN	1.006865	NaN	NaN	NaN	Na
2015-01-07	NaN	NaN	NaN	1.041667	NaN	NaN	NaN	1.050000	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	1.0	Na
2015-01-08	NaN	NaN	NaN	1.044534	NaN	NaN	NaN	1.014493	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Na

5 rows × 2773 columns

Exploratory Data Analysis

The first task that I did in this project is to observe the data, its time stamp.

There are data for stocks (tickers) (ASX) from **2015-01-02** to **2018-06-29** (**40 months**).

Data for the weekends and holidays are not available, which is obvious. Stock market are closed on those days. Holidays and weekends can also be provided as a feature to let model know about it.

- Some tickers are listed late.
- There are tickers with little to no data.
- There are missing values in between.
- Correlation between different features

- Coverage - are there missing values? If so, how could these be handled?

  It is important to have a deep understanding about data to find and fill the missing values. So, while handling missing value, I will find answers for these questions

  - Whether the missing data is happening at random?
  - Why the data is missing?
  - Do these missing values provide some useful information?
  - Will these be serious impact if I completely remove the row or column of data where there are missing values?

How to handle missing data?

Simple approach can be:

- Dropping the entire row and column wit missing data
- Imputing the missing values using statistical approach with the help of non-missing values
  - Mean, median, mode imputation
- Advance approach
  - We can build models such as multiple imputation using chained model (MICE) using Bayesian ridge or extremely random forest model.

In this project, I take the full-time stamp (2015-2018) to see the % of missing data. I found out that some tickers don't have values at all. It is obvious that we need many observations as the period of the maximum expected seasonality to create a nice prediction model. According to my understanding, with less than 2 years of daily data, we cannot create a nice model for time series prediction. Hence, in the project, if the missing values between the time stamp (2018 – 2015) is less then 2 years, I have not used it. Also, from my experience, if there are more than

25% of missing values in time series data, its prediction would be efficient. In this case, we should focus on collecting more data rather than wasting our time creating the model.

While doing this analysis, I should ignore counting leading NANs. Sime tickers might have listed in the ASX marker late.

I have filtered those tickers. I imputed missing values with the method ffill (forward fill). I did this because of the time constrain).

ARIMA, KATS from the Facebook can be a great open source tool to handle these things

- The quality of the data - are there any suspect values? If so, how should they be handled? can you verify these from independent sources?

  We can handle this from the data distribution. These might me some outliers and has to be fixed. We looked at the data and analyse weather the falls within the data distribution or not. Also, we can use box plot to see the outliers.

- **The time series structure of the data - how does this impact your analysis?**

  For time series, we should always see whether the data is stationary or not. The basic assumption about stationary is mean and variance doesn't change over time. If the data is not stationary, machine learning model cannot meet global optimal and will not work. In this case, we should first convert non-stationary data in to stationary.

  Different methods to find stationarity of data
  - Zero mean
  - Can be calculated from certain static value.
  - Can used some tool for fast processing.

  Data has seasonality, trend and pattern. We can use this information to see whether the data is stationary or not.

  We have identified the stationarity of data as demonstrated in code.

Forecasting

In real time, we can consider market trend as an additional feature to predict. For example, we can see what type of analysis there in the social media platform is. Based on its positive and negative sentiment value score, we can determine future closing price.

In our case, we can create some static variable such as what is the trend in the last week, months and year. We can even observe seasonality trend. With feature engineering, we can capture all these trends. During forecasting, we use all these features to create the model and compare the results.

In this project, I have tried to visualize using ARIMA model (Auto regressive integrated moving average). ARIMA and KATS has many useful readymade tools to check data's stationarity, seasonality etc.

Also, for prediction I have used ARIMA and neural network (LSTM) (both univariate and multi variate). During prediction, we have dependent variable (future close return) (Y) and independent variable (historical data time series). For the demonstration purpose, I have used last week's average, last week's max, last week's min, Close, last week's data change, lags until shift 10.

I have used training, validation and testing = 60, 20, 20. In other problems, we can choose randomization. But in timeseries, randomization is not suitable for time series. RMSE is used to check the performance of the model.

Simple trading strategy

I think we can treat this problem as a classification problem. In the share market, we will have three variables: BUY, SELL and HOLD.

If the difference is -ve and greater than 1% = BUY

If the differences in +ve and greater than 1% = SELL

If it's within the 1% range = HOLD

If we convert it into three variable, regression will convert to the classification problem.

Overfitting problem

For this I have drawn a residual plot. It can even be seen from training and testing error on each epoch. Based on these values, we can analyse whether the model is overfitted or under fitted (bias and variance)

Good models errors should be equally distributed in both +ve and -ve direction. (zero mean). Ex: if it is inclined towards +ve direction, it’s a positively biased models and it should be fixed with regularization techniques.