Project 3 - Analyze DEMA of DJIA over 15 years

Aliaksandr Panko

March 19, 2018

Contents

1 Objectives

1. Download data for Dow Jones Index (DJIA) for the last 15 years.

2. Compute daily percentage deviations of mid- price of DJIA from the 200 DAY Exponential Moving average (200DEMA) of close prices.

3. Prices above the 200-DEMA would get a positive value while those below would get a negative value

4. Whenever price is above 200-DEMA the market is generally considered to be in an up-move (and vice versa). Graphically represent the historical deviations and mark out clear periods of overall bullish and bearish regimes in DJIA.

5. For each of these regimes, clearly fit separate linear trend-lines and plot it on the same graph.
6. From the difference in relative slopes of the different trend lines, make a note about the relative shock during market regime shifts (e.g. – if a long bull market gave way to a sudden correction, the angle between the trend-lines of the two regimes would be high and would correspond to a violent regime-shift)
7. Using the trend-line of the last-but-one regime, forecast the generic direction of price pattern for the last-regime. Did the projected and actual regimes match?

2 Implementation

2.1 Data Download

Required data has been downloaded from Yahoo Finance using next python function:

```
from pandas_datareader import data as pdr
pdr.get_data_yahoo()
```

2.2 MA vs EMA

A **moving average (MA)** is a widely used indicator in technical analysis that helps smooth out price action by filtering out the "noise" from random price fluctuations. It is a trend-following, or lagging, indicator because it is based on past prices.

The 50-day and 200-day MAs are widely followed by investors and traders, with breaks above and below this moving average considered to be important trading signals.

Moving averages also impart important trading signals on their own, or when two averages cross over. A rising moving average indicates that the security is in an uptrend, while a declining moving average indicates that it is in a downtrend.

MA can be calculated using the next formula:

$$MA_i = \frac{\sum_{i=1}^{n} P_i}{n}$$

, where P_i - prices

An exponential moving average (EMA) is a type of moving average that is similar to a simple moving average, except that more weight is given to the latest data. It's also known as the **exponentially weighted moving average**. This type of moving average reacts faster to recent price changes than a simple moving average.

EMA can be calculated using the next formula:

$$EMA_i = ((P_i - EMA_{i-1}) \times k) + EMA_{i-1}$$

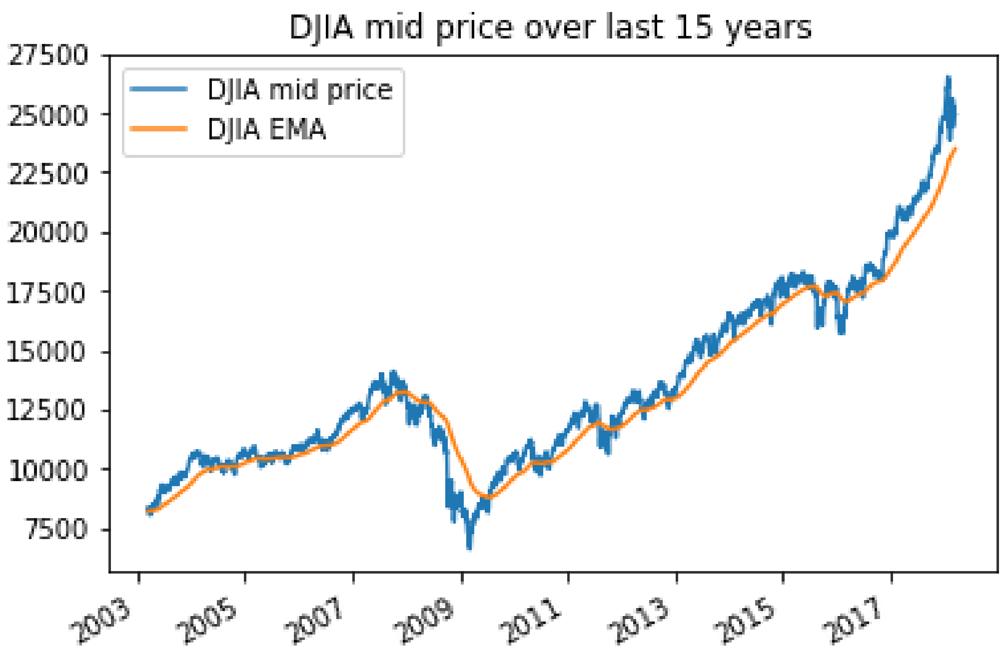
, where P_i is a current price, k is a multiplier that can be found by the formula below:

$$k = \frac{2}{Period + 1}$$

and initial value for EMA_i can be determined by the first price P_0

Amazing python function **pdf.ewm().mean()** calculates required exponentially weighted moving average.

DJIA mid price and exponentially weighted moving average are represented below:



2.3 Price Deviation

First I computed the **mid price** according to the next formula:

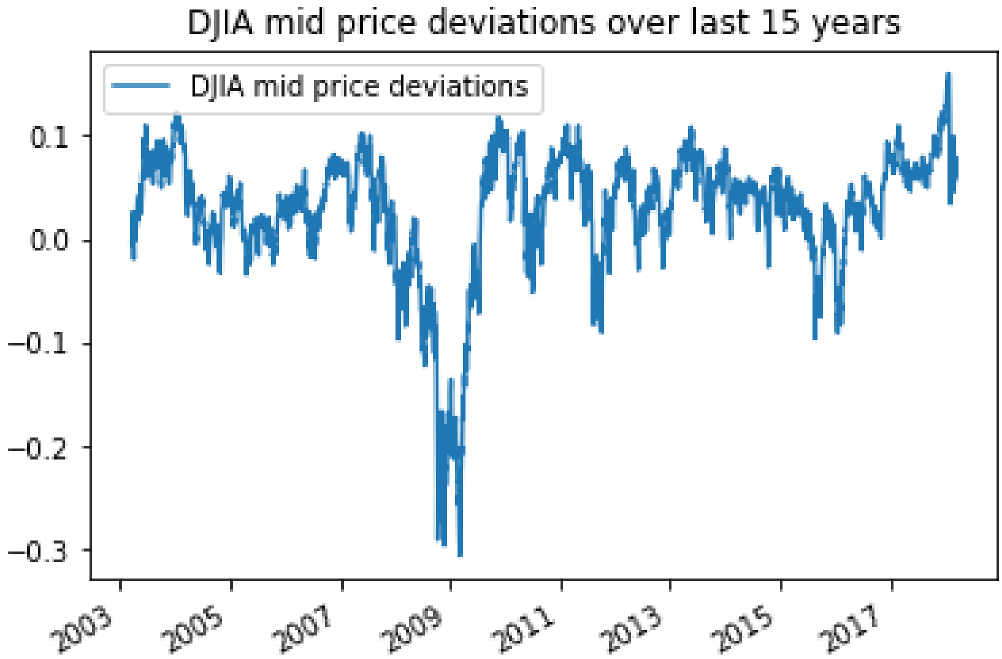
$$Mid = \frac{Low + High}{2}$$

The next step was to calculate daily percentage deviations of mid-price of DJIA from the 200 DAY Exponential Moving average (200DEMA) of close prices.

This was done using simple formula below:

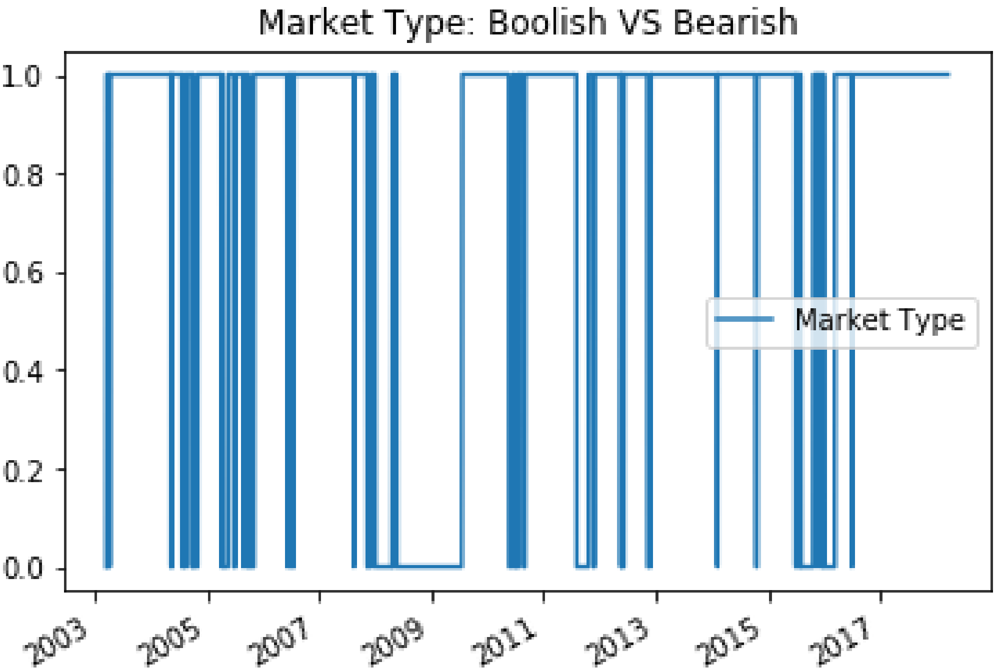
$$Deviation(\%) = \frac{Mid - EMA}{EMA}$$

DJIA mid price and exponentially weighted moving average are represented below:



2.4 Bullish and Bearish Periods

I represented Bullish and Bearish Periods using a signals as shown below:



Value 1 corresponds to Bullish market type and 0 - to Bearish

2.5 Trends

We need to analyze separately bearish and bullish periods and for each period determine a trend line. To do this I use the next algorithm:

1. Determine start and end date of a trend
2. Using **np.polyfit()** function and the prices during determined period find the trend line coefficients
3. Save found trend in a data frame
4. Later all trends are plotted using **plt.plot()** function

As a result we have:

Additionally I would like to mention that a figure size can be set up using **plt.figure(figsize=(10, 7))**

And it is much better to have original dates as x-axis then just some numbers. This can be done using **df.index[range].values** as **x** in **plt.plot(x,y)** Moreover **plt.gcf().autofmt_xdate()** helps to get beautiful date line.

2.6 Slopes vs Shocks

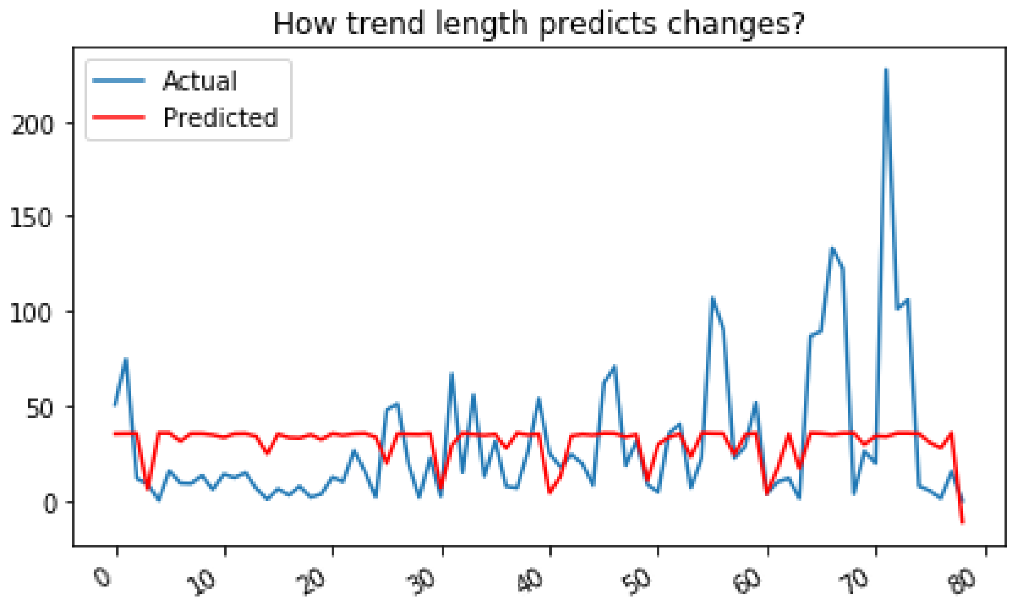
Let's check a claim that long trends change more dramatically than short ones. Mathematically speaking we need to check:

$$d_1 > d_2 \Rightarrow \Delta slope_1 > \Delta slope_2$$

To check this claim I decided to use regression

$$\Delta slope = \beta * length + intercept$$

The result show that there are no significant relationship between these two variables:



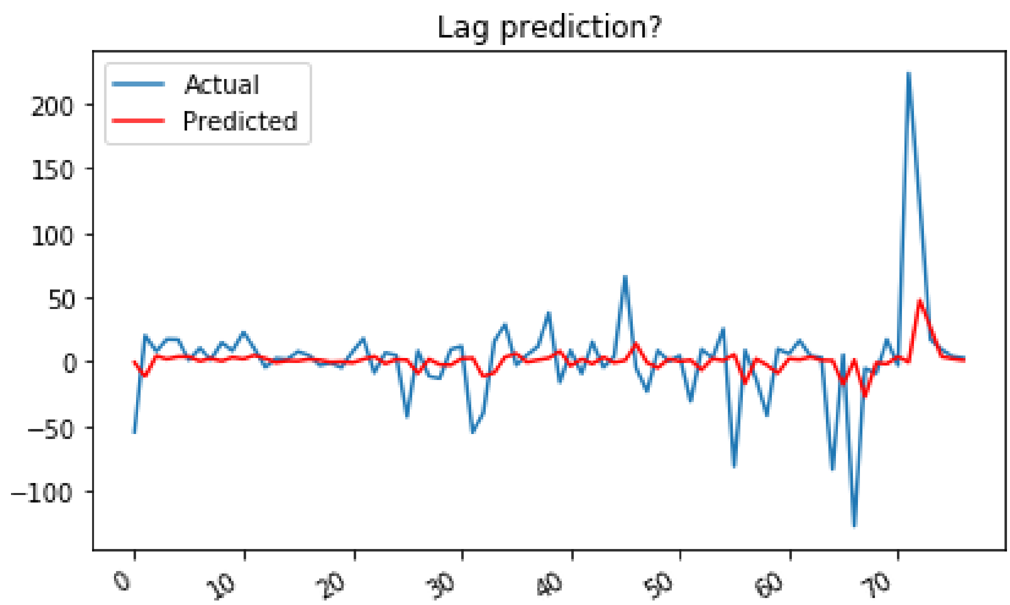

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.058
Model:                            OLS   Adj. R-squared:                  0.046
Method:                 Least Squares   F-statistic:                     4.748
Date:                Mon, 19 Mar 2018   Prob (F-statistic):             0.0324
Time:                        19:49:51   Log-Likelihood:                -398.12
No. Observations:                  79   AIC:                             800.2
Df Residuals:                      77   BIC:                             805.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         35.8334      4.850      7.389      0.000      26.177      45.490
x1            -0.1087      0.050     -2.179      0.032      -0.208      -0.009
==============================================================================
Omnibus:                       59.854   Durbin-Watson:                   1.226
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              276.793
Skew:                           2.396   Prob(JB):                     7.85e-61
Kurtosis:                      10.819   Cond. No.                         111.
==============================================================================
```

2.7 Prediction

In this section I tried to forecast the last slope using previous one. The prediction was made based on regression model AR(1)

$$\Delta slope_i = \beta * slope_{i-1} + intercept$$

As a result intercept in my model was insignificant, so I used model without it. Anyway the model showed very poor results:




```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.045
Model:                            OLS   Adj. R-squared:                  0.033
Method:                 Least Squares   F-statistic:                     3.594
Date:                Mon, 19 Mar 2018   Prob (F-statistic):             0.0618
Time:                        20:37:16   Log-Likelihood:                -391.04
No. Observations:                  77   AIC:                             784.1
Df Residuals:                      76   BIC:                             786.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.2125      0.112      1.896      0.062      -0.011       0.436
==============================================================================
Omnibus:                       55.007   Durbin-Watson:                   2.021
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              662.547
Skew:                           1.679   Prob(JB):                    1.35e-144
Kurtosis:                      16.972   Cond. No.                         1.00
==============================================================================
```

It is not difficult to calculate next value but with such model it has no much sense.

$$2.5636 * 0.21 = 0.54$$

and yes it is far from the real value