Abbey Dent

ECON 9000

Tom Lam

23 April 2019

Coinmarketcap scrapper

This assignment consists of two main parts. First, I have written a program to scrape data from coinmarketcap.com. Secondly, I used the data retrieved from my scrapper and performed some machine learning analysis on the it using some tools learned in class. The web scraping program I have written combines two tasks that together work to collect the targeted data in a meaningful and comprehensive manner – requesting and parsing. My program largely relied on the BeautifulSoup library to collect the data in a clear and concise way.

The first step in successfully running the scraping program I have written is running the request file. The coinmarket_resquest.py file request the html files from coinmarketcap.com. The request file contains several chunks of code with the heart of the code being the two for loops using indices i. The first for loop is requesting the html file from coinmarketcap.com/all/views/all which contains the name, symbol, market cap, price, circulating supply, volume, and percent changes for all the cryptocurrencies. For this particular program I set the loop to run 30 times with the sleep timer set to 21,600 seconds, so that the program would request the html files from coinmarketcap every 6 hours over a one-week period. The second for loop then uses a list of the currency names and the href links provided from the tables inside one of the html files collected earlier to loop through the top 500

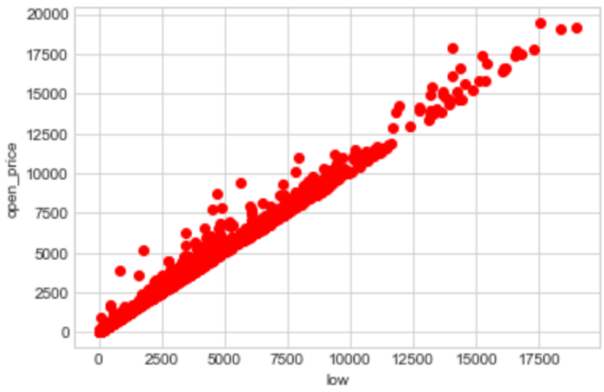
currencies, requesting the historical data for each one. For the historical html files I set the sleep timer to just 30 seconds due to the higher number of files that I needed to request. For the first loop mentioned I created and set the working directory to html_files and then created and changed the directory to historical_html_files for the second for loop mentioned which provided places for my computer to store the files and made the program cleaner and easier to execute. I chose only to scrape the historical data from the top 500 cryptocurrencies. The numbers for opening and closing price, day high and low, volume, and market cap begin approaching 0 for the lower range of the top 500 currencies, and for the currencies not far beyond the bottom 500 currencies, the numbers actually do reach 0. Because of this, I decided it was not worth the time and resources required to retrieve the data for the currencies beyond the top 500 to actually do so. Allowing the computer to request the bottom 1500+ currencies would have required significant time and effort from the computer to retrieve data that was virtually meaningless and not practically useful.

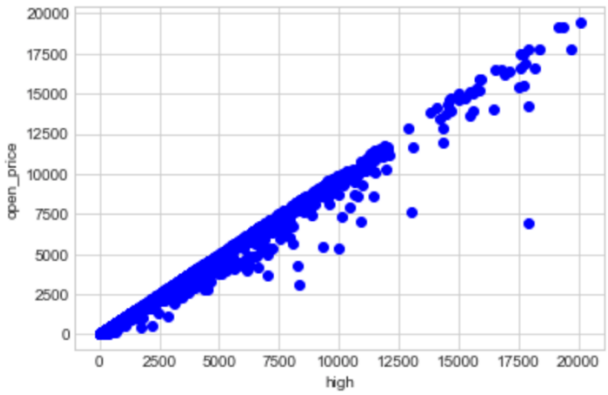
The next step in successfully running my web scraping is to now parse all of my html files and to then write the parsed files to a .csv file. I found the simplest and most easy way to do this was to write two separate parsers – one for the basic html files and one for the historical html files. The first parser I wrote parsed the basic html files by using a for loop that opened each file, read the file using BeautifulSoup, closing the file. From what it read with BeautifulSoup the loop was then able to create a table by using BeautifulSoup to find the table in the html code with the ID “currencies-all”, and then within that table find the tbody and finally find all the tr’s within that tbody. Each tr was essentially a row on the coinmarketcap website which contained all the desired information for each currency with every currency

being its own row. Nesting another for loop within the loop I was able to from each of those currency rows, extract the currency's short name, full name, market cap, price, circulating supply, volume, and percent change in the last 24 hours. I then appended each of these items into a data frame I had initialized earlier and finally converted the data frame into a .csv file. For the historical html files I used a similar approach, writing a for loop that opened each file, read it with BeautifulSoup and then closed it. I also used BeautifulSoup to find the table, the tbody within that table, and all the tr's within the tbody. Similarly to the first parser, I also nested a for loop into the loop which went through each row to pull the data I needed. Because each row did not correspond to a specific currency, retrieving the data was a little different here. Each tr contained tds that did not have specific classes and so for each variable I had to use indices to signify which td corresponded to which variable. For each variable I also used .text to let the program know to take the element within the text quotations for each td. As with the first parser, for this one I also appended each variable into my data frame which I had already initialized and then saved the data frame as a .csv file.

For the second part of this assignment, I sought to conduct some sort of machine learning analysis on the data I retrieved using some of the tools learned from class. Cryptocurrencies are different from stocks in that the market is open 24/7. This is interesting because the closing and opening times are only separated by one second. Because this is the case I would assume that unlike stocks, the relationship between the day's closing price and the current day's opening price should actually be positive rather than negative. This is an easy hypothesis to test with the regression techniques learned in class.

Before beginning my analysis, I first checked to make sure that my data seemed reasonable. I did this by testing the relationship between variables whose direction I could reasonably assume. I looked at the relationship between the variables low and open_price, and high and close_price. I could reasonably assume that these would both show positive relationships because if the currency opens at a higher price it makes sense that its day high would be greater and the same with low closing price and day low. I generated the scatterplots for these variables to verify that this is correct and as you can see both show positive relationships so I can continue with my analysis as my data seems to be accurate.





Now I can tackle my question about the relationship between open price and close price and see if my hypothesis is correct. I do this by regressing close price on open price to see how the closing price is effecting the opening price.

Out[55]:

OLS Regression Results

Dep. Variable:	open_price	**R-squared:**	0.993
Model:	OLS	**Adj. R-squared:**	0.993
Method:	Least Squares	**F-statistic:**	4.403e+07
Date:	Tue, 23 Apr 2019	**Prob (F-statistic):**	0.00
Time:	23:09:13	**Log-Likelihood:**	-1.5039e+06
No. Observations:	304342	**AIC:**	3.008e+06
Df Residuals:	304340	**BIC:**	3.008e+06
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1235	0.062	2.005	0.045	0.003	0.244
close_price	0.9970	0.000	6635.718	0.000	0.997	0.997

Omnibus:	221833.089	**Durbin-Watson:**	1.995
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	103172358308.570
Skew:	1.241	**Prob(JB):**	0.00
Kurtosis:	2855.372	**Cond. No.**	412.

My results show that the relationship between closing price and opening price is positive and significant at the 95% level with a t-stat of 6635.718. According to my results a 1 unit increase in the closing price is correlated with a .997 unit increase in the opening price. This is as I expected

as the closing price and opening price do not have the same effects on one another with cryptocurrencies as that they do with stocks.

The positive relationship between closing and opening price also would lead me to believe that a higher opening price will be associated with a higher day high. To test this intuition I regress opening price on day high and check my results.

Out[56]:

OLS Regression Results

Dep. Variable:	high	**R-squared:**	0.992
Model:	OLS	**Adj. R-squared:**	0.992
Method:	Least Squares	**F-statistic:**	3.805e+07
Date:	Tue, 23 Apr 2019	**Prob (F-statistic):**	0.00
Time:	23:18:48	**Log-Likelihood:**	-1.5418e+06
No. Observations:	304342	**AIC:**	3.084e+06
Df Residuals:	304340	**BIC:**	3.084e+06
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1461	0.070	2.093	0.036	0.009	0.283
open_price	1.0494	0.000	6168.231	0.000	1.049	1.050

Omnibus:	1260733.416	**Durbin-Watson:**	1.999
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	7312735338977.051
Skew:	114.061	**Prob(JB):**	0.00
Kurtosis:	24015.922	**Cond. No.**	412.

My results are as expected. Opening price has a positive relationship with day high and is significant at the 95% level with a t-stat of 6168.21. A 1 unit increase in opening price is correlated with a 1.0494 unit increase in day high. I also test the relationship between low and closing price by regressing day low on closing price. I expect the two to also have a positive relationship with lower day lows corresponding to lower closing prices

Out[57]:

OLS Regression Results

Dep. Variable:	close_price	**R-squared:**	0.995
Model:	OLS	**Adj. R-squared:**	0.995
Method:	Least Squares	**F-statistic:**	6.018e+07
Date:	Tue, 23 Apr 2019	**Prob (F-statistic):**	0.00
Time:	23:22:31	**Log-Likelihood:**	-1.4565e+06
No. Observations:	304342	**AIC:**	2.913e+06
Df Residuals:	304340	**BIC:**	2.913e+06
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1203	0.053	2.281	0.023	0.017	0.224
low	1.0534	0.000	7757.365	0.000	1.053	1.054

Omnibus:	988807.377	**Durbin-Watson:**	1.998
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	315097964072.414
Skew:	54.683	**Prob(JB):**	0.00
Kurtosis:	4986.597	**Cond. No.**	390.

Again, my results are as expected. Day low and closing price have a positive relationship which is significant at the 95% level with a t-stat of 7757.365. These results show that the relationships between prices with cryptocurrencies are different than with stocks. This assignment just shows one way in which machine learning analysis could be used here but the data generated from my scrapper could be used to examine far more relationships than mentioned above. This project simply shows one way in which machine learning analysis paired with a useful dataset can be an invaluable instrument in data analysis.