Predicting Oil Prices

Finance and Commerce

Jeffrey Hale (SUID# 06246920)

Motivation:

My aim is to predict the future pricing of oil based on various factors to model supply and demand, historical daily oil / futures pricing, and historical daily correlated indicators related to the price of oil.

Most financial machine learning projects I have seen in the past are focused on equity markets – with the end goal being speculation for financial gain. My goal is to predict commodity pricing for better planning of capital expenditures and estimation of future revenue from the point of view of the commodities supplier.

It is my belief that the commodity cycle is driven by the mismatch of the capital investment cycle with the demand cycle, leading to periods of oversupply and shortage. I am using only publicly available data for this project, so I have chosen the oil market as there are vast amounts of public research and data available. My model will attempt to predict the price of oil 30 days from the given date.

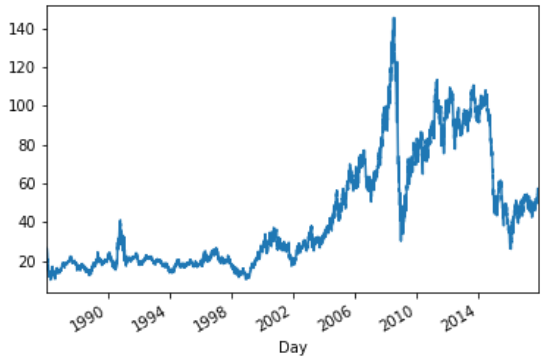
To this end, I built two models to predict oil pricing, one based on an autoregressive integrated moving average (ARIMA) and another based on a neural network (NN). We can see mixed results, where the ARIMA builds a simple lagging version of the input data and the neural network sees either overfitting of the training set or underfitting of both the training and test sets. The results indicate that without more sophisticated selection of the inputs, the neural network will lag the commonly used "technical analysis" done in the financial industry to determine trends.

Data:

The primary dataset is the WTI Crude Oil daily spot prices from January 1986 to November 2017. Other data considered are historical daily treasury rates, gold pricing, S&P500 pricing, oil company capital expenditures and various industry reports such as the OPEC monthly report.

The oil data consist of the daily and monthly WTI crude oil spot prices from January 1986 to November 2017.

Figure 1: WTI Daily Crude Oil Spot Prices ($USD)



This data includes daily values for all business days as well as regular monthly values. The daily data set includes 8042 values and the monthly data includes 382 data points.

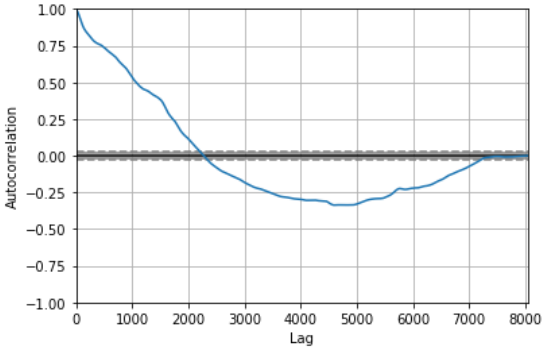
Other Features:

The features for the ARIMA are the historical oil prices themselves, while the features for the neural network that I tested vary. For the neural network, I added lagging oil price data as features for a given date, as well as trying out various combinations of correlated and uncorrelated historical data such as interest rates, gold prices, and capital expenditures of oil companies.

I also looked at some autocorrelation data to see if there was an amount of lagging data that could correlate to the current pricing to add as features. This autocorrelation data can be used to determine how many lagging data points to include for both the ARIMA and the

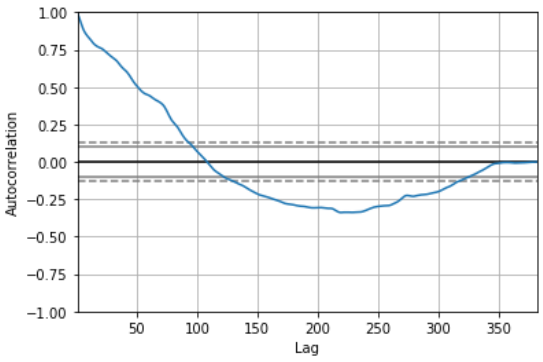
neural network model. I expected that seasonal trends could be better predicted by relying on past patterns.

Figure 2: Autocorrelation
WTI Daily Crude Oil Spot Prices



For the daily data, we can see that roughly the last 2,000 days have positively correlated data, with the last 1,000 days having significaltly (greater than 0.5) correlation. Since this data is based on a five day business week, 1,000 days roughly translates to four years of data.

Figure 3: Autocorrelation
WTI Monthly Crude Oil Spot Prices ($USD)



For the monthly data, we can see that roughly the last 100 months have positively correlated data, with the last 50 monhts having significaltly (greater than 0.5) correlation. Fifty months is just over four years of data, so this lines up with what we can see from the daily autocorrelation.

Method / Experiments:
I used two different models to determine the most accurate way of predicting future oil prices. First, I have chosen a simple Autoregressive Integrated Moving Average (ARIMA) model to get a baseline to compare the others against. This model seems to be fairly standard for use in extrapolating time series data and I have been able to quickly run it to see the results.

For this model, the cost is measured as the mean squared error (MSE) to determine it's effectiveness.

Figure 4: Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

The other factors to consider in an ARIMA model are:

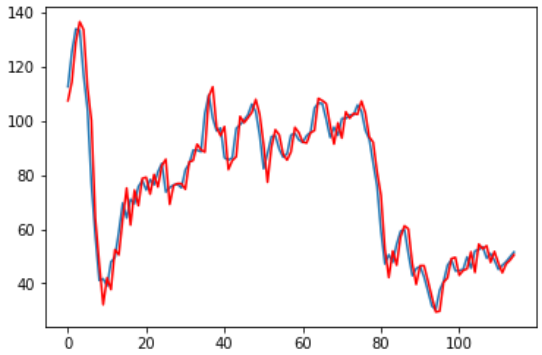
- *p* - the order (number of time lags)
- *d* - the degree of differencing
- *q* - the order of the moving average

I tried varying values of these variable, and came to see that setting (*p*, *d*, *q*) = (1, 1, 0) gave me the lowest mean squared error. I had expected that using a large value for *p*, such as the 50 found from autocorrelation to provide the most accurate prediction, but I found that numbers higher than 1 for *p* just slowed down the computation and did not decrease the cost.

Given that I am primarily concerned with the price of oil 30 days away from a given date, I will consider the monthly ARIMA prediction model here, which predicts the price of oil one month later.

Using 70% of the data for the training set and 30% for the test set, the ARIMA model for the monthly data can achieve mean squared error of 38.098 for the test set monthly data when attempting to predict the next month.

Figure 5: ARIMA – Monthly Oil Predictions (*p* = 1, MSE 39.572)



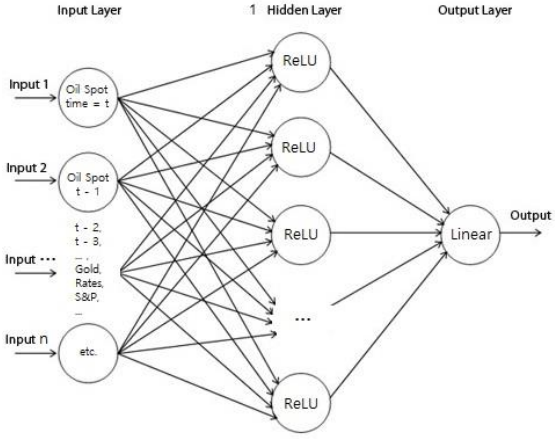
Other values of *p* fared worse than *p* = 1:

Table 1: ARIMA model *p* vs MSE

p (lagging months)	MSE
0	45.814
1	38.098
2	38.963
3	39.061
4	39.562
5	39.572
6	39.015
12	43.752

The other model I used was a shallow neural network with one hidden layer. The inputs to this network are initially the oil data plus lagging oil data, with eventual addition of other features such as gold pricing, US treasury interest rates, and oil company capital expenditures.

Figure 6: Neural Network with 1 hidden layer



For the activation of the hidden layer units, I used a ReLU function, which is equal to its input for all values greater than zero and equal to zero for all negative values.

Figure 7: The ReLU function

$$f(x) = x^+ = \max(0, x)$$

For the single unit output layer, I used a linear function to produce an actual future oil price, or something like the percentage increase expected for the oil price.

The cost function I used for the neural network was the same as that of the ARIMA model, the mean squared error as seen in figure 4.

I found that the result was much more accurate and interesting when I used the percentage gain/loss after 30 days rather than the actual value of the oil price. Unfortunately, this end result cannot be directly compared to the ARIMA model, but when using the predicted oil price itself I did not achieve anything approaching the ARIMA model.

I toyed around with learning rated and number of units in the hidden layer. I found that I needed a very small learning rate, otherwise I would not get convergence and I settled on a learning rate of 0.000001.

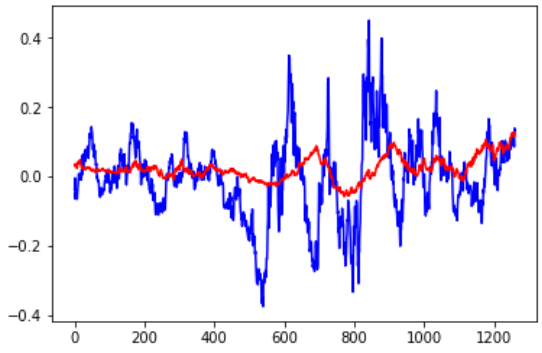
As for the number of units in the hidden layer, I tried anything from 10 to 500. I found that too many hidden units only attributed to more overfitting and slowed down processing without adding to the eventual accuracy of the test set.

As with the ARIMA model, I trained on 70% of the data and tested on the remaining 30% of the data.

I achieved mean squared error of 0.0068 with 200 units in the hidden layer and a learning rate of 0.000001. The result does not follow the full swings of the actual price movement, but achieves a more smoothed, softer version,

generally with swings in the same direction of the actual price movement as seen in figure 8.

Figure 8: Shallow Neural Net
MSE: 0.0068



Discussion:
It is still my belief that commodity cycles like that of the oil market are driven by the mismatch of the capital investment cycle with the demand cycle, leading to periods of oversupply and shortage. In this project, I attempted to model this mismatch by supplying a neural network with some of the indicators of future supply and demand in order to try to predict the effect of this expected mismatch on future oil pricing, but it seems that better indicators are needed other than the inputs I came up with.

Future:
In the future, I would like to explore using the neural network approach more in depth. I would like to try a deeper network as well as adding more complicated and nuanced features such as the word counts of key words in the monthly OPEC reports.

OPEC has published monthly reports starting in January 2001 that have many hard datasets as well as a wealth of text. I intend to use word count data, and after calculating the Pearson correlation coefficient over the selected words I will add the word counts of words that are highly correlated with oil pricing.

Other features to be considered in the future are Oil inventory and production levels, Refining Capacity, Rig Utilization, as well as indicators for demand, including: Economies of major oil consuming countries, mainly US and China: GDP Growth, Manufacturing, Shipping, Transportation, Employment, and Military spending.

I also want to experiment with the output of my model, for instance instead of predicting just the oil price or the percentage of increase/decrease, using SoftMax buckets, where the output of my model will give the most likely percentage move range on a given future date, i.e. (-6% to -2%), (-2% to +2%), (2% to 6%), etc., to increase the accuracy of the model.

I would also like to try some kind of hidden Markov model as these models have shown to do well predicting time series data.

Processing power permitting, I would like to try implement a deep recurrent neural network model and believe that this model may hold the potential to accurately predicting supply/demand imbalances as it can keep a memory of all factors that influenced the price in the past at a given time.

Contribution:
As this is a one-person team, all work is done by Jeffrey Hale (SUID# 06246920).

Prior Research:
I leaned on prior research of modelling of commodity pricing, including former Stanford student Megan Potoski who modelled the future price of gold in her CS 229 project.
Megan Potoski. *Predicting Gold Prices.* Stanford, 2013.
http://cs229.stanford.edu/proj2013/Potoski-PredictingGoldPrices.pdf

I also utilized the following guide on ARIMA models in python:
https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Data Sources:

Daily/Monthly WTI Crude Oil Spot prices: https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=pet&s=rwtc&f=d

Exxon historical Capex: Exxon Financial Reporting https://www.sec.gov/cgi-bin/browse-edgar?CIK=xom&owner=exclude&action=getcompany

Historical Gold Pricing https://www.gold.org/data/gold-price

US GDP https://www.bea.gov/national/index.htm#gdp

China GDP https://data.worldbank.org/country/china https://data.worldbank.org/indicator/NY.GDP.MKTP.CD?locations=CN

S&P 500 https://www.investing.com/indices/us-spx-500-historical-data

Treasury yields: Treasury.gov https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yieldAll

EIA Weekly Status Report (2011-) https://www.eia.gov/petroleum/supply/weekly/archive/#2016-2017

OPEC Monthly Market Report (2001-) http://www.opec.org/opec_web/en/publications/338.htm

IEA Monthly Report (1990-) https://www.iea.org/oilmarketreport/tables/