Udacity MLND Capstone Project

Prepared by Conrad Gann, OCTOBER 5th, 2016

Sector Rotation in the SP500

A study of using PCA with Linear Regression to predict the SP500 using its composite sector ETF's

(December 31, 1998 - December 31, 2014)

1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	Cumulative 1999-2014
Technology 64.5%	Financials 24.13%	Consumer Discretionary 11.74%	Healthcare -1.48%	Technology 37.7%	Energy 31.83%	Energy 38.52%	Consumer Discretionary 17.49%	Energy 35.34%	Consumer Staples -17.12%	Technology 48.80%	Consumer Discretionary 25.66%	Utilities 14.81%	Financials 26.08%	Consumer Discretionary 40.87%	Utilities 24.36%	Energy 238.65%
Basic Materials 22.64%	Consumer Staples 24.02%	Basic Materials -0.0093%	Basic Materials -7.38%	Consumer Discretionary 36.26%	Utilities 19.37%	Utilities 12.71%	Utilities 16.98%	Basic Materials 19.79%	Healthcare -24.81%	Basic Materials 45.07%	Industrials 25.48%	Consumer Staples 10.85%	Consumer Dicretionary 21.58%	Healthcare 39.02%	Healthcare 23.34%	Consumer Dicretionary 176.17%
Industrials 20.54%	Energy 22.49%	Healthcare -1.1%	Financials -16.35%	Basic Materials 34.53%	Industrials 16.11%	Healthcare 5.07%	Energy 16.54%	Utilities 15.28%	Utilities -31.42%	Consumer Discretionary 38.02%	Energy 19.72%	Healthcare 10.13%	Healthcare 14.96%	Industrials 37.89%	Technology 15.70%	Healthcare 165.87%
Healthcare 20.11%	Utilities 17.93%	Financials -10.85%	Energy -16.37%	Industrials 29.97%	Consumer Discretionary 12.07%	Financials 3.73%	Financials 16.01%	Technology 14.62%	Consumer Discretionary -34.04%	S&P 500 23.45%	Basic Materials 16.43%	Consumer Discretionary 4.30%	S&P 500 13.41%	Financials 33.37%	Financials 13.13%	Industrials 130.35%
S&P 500 19.51%	Industrials 5.54%	Consumer Staples -11.07%	Consumer Discretionary -19.2%	Financials 27.86%	Basic Materials 11.39%	S&P 500 3.00%	Basic Materials 14.96%	Industrials 11.85%	S&P 500 -38.49%	Energy 19.34%	S&P 500 12.78%	Energy 1.29%	Technology 13.36%	S&P 500 29.60%	Consumer Staples 12.82%	Basic Materials 124.16%
Consumer Discretionary 18.9%	S&P 500 -10.12%	Industrials -11.36%	Consumer Staples -21.5%	S&P 500 26.38%	S&P 500 8.99%	Basic Materials 1.85%	S&P 500 13.62%	Consumer Staples 10.26%	Energy -39.80%	Industrials 18.66%	Financials 10.76%	Technology 1.03%	Industrials 12.30%	Energy 23.93%	S&P 500 11.39%	Consumer Staples 77.24%
Energy 15.91%	Healthcare -11.79%	S&P 500 -13.05%	S&P 500 -23.37%	Energy 23.38%	Financials 8.53%	Industrials 1.13%	Consumer Staples 12.15%	Healthcare 5.43%	Industrials -40.19%	Healthcare 17.02%	Consumer Staples 10.73%	S&P 500 0.00%	Basic Materials 12.06%	Technology 23.88%	Industrials 8.27%	S&P 500 67.47%
Financials 0.47%	Consumer Discretionary -17.61%	Utilities -15.54%	Industrials -25.67%	Utilities 21.83%	Consumer Staples 5.97%	Consumer Staples 0.91%	Industrials 11.43%	S&P 500 3.53%	Technology -42.20%	Financials 15.02%	Technology 9.86%	Industrials -3.21%	Consumer Staples 7.42%	Consumer Staples 23.15%	Consumer Discretionary 7.96%	Utilities 55.62%
Utilities -7.26%	Basic Materials -19.4%	Energy -19.55%	Utilities -31.68%	Healthcare 13.56%	Technology 3.58%	Technology -0.99%	Technology 11.29%	Consumer Discretionary -14.75%	Basic Materials -45.47%	Consumer Staples 10.89%	Healthcare 1.38%	Basic Materials -12.78%	Energy 3.31%	Basic Materials 23.12%	Basic Materials 5.11%	Technology 26.26%
Consumer Staples -15.82%	Technology -41.88%	Technology -23.35%	Technology -38.33%	Consumer Staples 9.23%	Healthcare 0.13%	Consumer Discretionary -7.48%	Healthcare 5.58%	Financials -21.26%	Financials -56.72%	Utilities 6.85%	Utilities 1.03%	Financials -18.50%	Utilities -2.95%	Utilities 8.73%	Energy -10.56%	Financials 4.54%
Average % Differential *(Best Performing - Worst Performing)*																
80.32%	66.01%	35.09%	36.85%	28.47%	31.70%	46.00%	11.91%	56.60%	39.60%	41.95%	24.63%	33.31%	29.03%	32.14%	34.92%	39.28%

I Definition

Ia: Project Overview

This Capstone project explores feature processing and dimensionality reduction using PCA then Linear Regression via a stock market prediction model. The goal of the model is to use PCA for dimensionality reduction and feature orthogonalization then linear regression to predict if the market will go up or down on the next trading day. The idea is that a hypothetical trader would use the model to go long the market if the model prediction is positive and short the market if it is negative. The expectation is that feature data this easy to acquire is unlikely to have any signal value on predicting up/down moves in the stock market.

The stock market is considered very efficient, making market timing strategies that work out of sample, ie that work after the estimation period, extremely difficult to find. If market timing strategies can be discovered, however, they can be very profitable because they are easy to trade with relatively low transaction costs.

Of course finding a good market prediction model is not easy to do. Most features have a high noise to signal ratio, and a limited number of data points (unlike the self driving taxi experiment where we can create data with a random number generator). Another hazard is the tendency of the modeller to look at out of sample performance and then tweek model parameters, causing out of sample metrics to be poisoned by out of sample human insights.

The features commonly used in stock prediction models usually include some form of historical returns. In this project, rather than predict individual stock returns we predict Exchange Traded Fund (ETF) returns.

Our proxy for the market is the "SPY" ETF which is trackes the SP500, a common benchmark index for stock investment strategies. The 500 stocks inside the index are classified under a hierarchy known as Global Industry Classification Standard or GICS. At the top level of this hierarchy are nine "Sectors", such as Technology and Financials. Since January 2000 these Sectors, which are weighted by the market cap of their component stocks, have been easy to trade via the Sector SPDR ETF's launched by State Street.

Ib: Problem Statement

The challenge for using historical returns as features for stock market prediction is that they violate nearly every assumption desired for evaluating a linear regression. While most of these violations are beyond the scope of this project, the collinearity of these features is something that machine learning can address with a dimensionality reduction technique called Principal Component Analysis or PCA. One of the key questions in a PCA Linear Regression model, how many dimensions should be used in the regression. Since there are 9 sector ETF's there are potentially 9 dimensions that could be passed to the regression.

Ic: Metrics

The golden grail of performance for hedge funds is risk adjusted return which is usually measured by return divided by standard deviation of return. The simplest metric for this is the ex-post sharpe ratio[1]. $\frac{Return - Benchmark}{Standard\ Deviation\ of\ Return}$. For hedge fund strategies, it is common to treat the asset Benchmark = 0 since the focus is on absolute return.

[1] In the hedge fund space, the benchmark return for a sharpe ratio is often set to zero if the strategy is intended to be uncorrelated to other investment benchmarks.

For this particular project, the risk component, which is used in the denominator of the Sharpe ratio, standard deviation of SPY, is the same for all variations of the model tested. Hence the only performance metric of the sharpe ratio that changes with different model configurations is return, which is the keyline metric for this paper.

Since returns are pre-processed to difference of log form I can sum the returns over a given time range, Return = $\Sigma\, Log\ Returns\ (T)$, to compute the total return. On average there are 252 trading days in a year so the annualized return shown in the performance tables is 252*Average Daily Return returned from the simulator.

Other reported metrics captured include :

1. rmse_ = np.sqrt(rmse(reg_out.predicted,reg_out.actual))
2. mae_ = mae(reg_out.predicted,reg_out.actual)
3. evs_ = evs(reg_out.predicted,reg_out.actual)
4. r2_ = r_squared(reg_out.predicted,reg_out.actual)
5. mean = periods_per_year*trade_out.strat_ret.mean()
6. stdev = (np.sqrt(periods_per_year))*trade_out.strat_ret.std()

Given the simple trading strategy deployed, the accuracy of predictions beyond the up/down component of the prediction, does not really matter. The model will be a success if it can get the up/down prediction right more than 50% of the time Thus while the rmse, mae, and evs metrics are reported they are not a focus for model evaluation.

II Data Analysis

IIa: Data Exploration

Here is a sample of the raw dataset as in comes in from quandl - before pre-processing.

	xlf	xly	xlp	xlb	xlu	xlk	xle	xli	xlv	spy
2000-01-03	16.3659	24.6966	15.7681	17.5866	15.0350	45.7724	19.8395	21.4162	24.0592	106.9868
2000-01-04	15.6505	23.9536	15.3241	17.4173	14.5815	43.4503	19.4660	20.8245	23.5133	102.8029
2000-01-05	15.5275	23.6590	15.5948	18.1369	14.9495	42.8052	19.9795	20.7317	23.3023	102.9869
2000-01-06	16.2094	23.9280	15.8872	18.6025	14.9238	41.3861	20.7497	21.0101	23.3768	101.3317

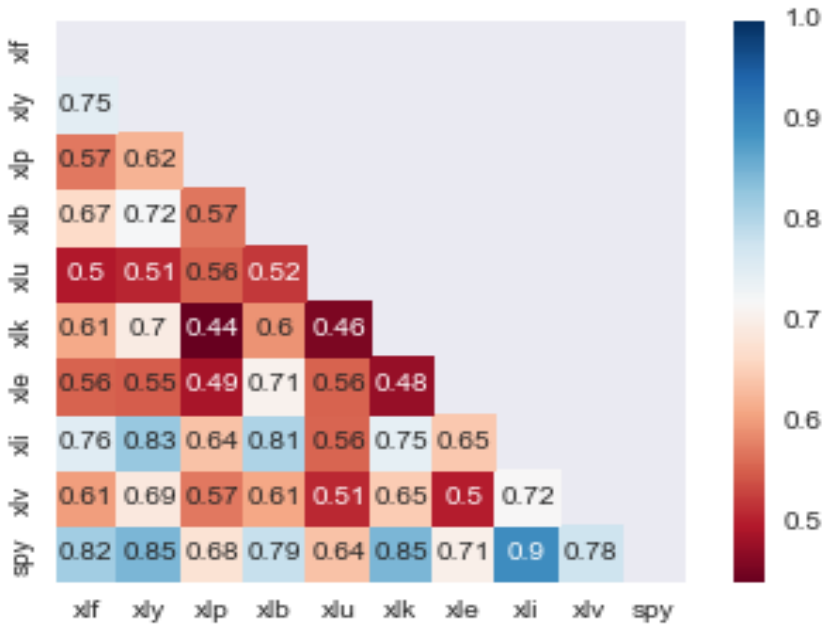
Here is a summary of the Target and Feature set. The most volatile sector is XLF and the least volatile sector is XLP.

Ticker	xlf	xly	xlp	xlb	xlu	xlk	xle	xli	xlv	spy
count	4,174	4,174	4,174	4,174	4,174	4,174	4,174	4,174	4,174	4,174
AnnRet	2.32	7.21	7.50	6.16	7.34	0.17	7.40	6.05	6.88	4.30
std	0.020	0.015	0.010	0.016	0.012	0.017	0.018	0.014	0.012	0.013
min	-0.191	-0.124	-0.062	-0.133	-0.089	-0.091	-0.156	-0.099	-0.103	-0.104
25%	-0.007	-0.006	-0.005	-0.008	-0.005	-0.007	-0.008	-0.006	-0.005	-0.005
50%	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
75%	0.008	0.008	0.005	0.009	0.007	0.007	0.010	0.007	0.006	0.006
max	0.273	0.093	0.067	0.132	0.114	0.149	0.153	0.102	0.114	0.136

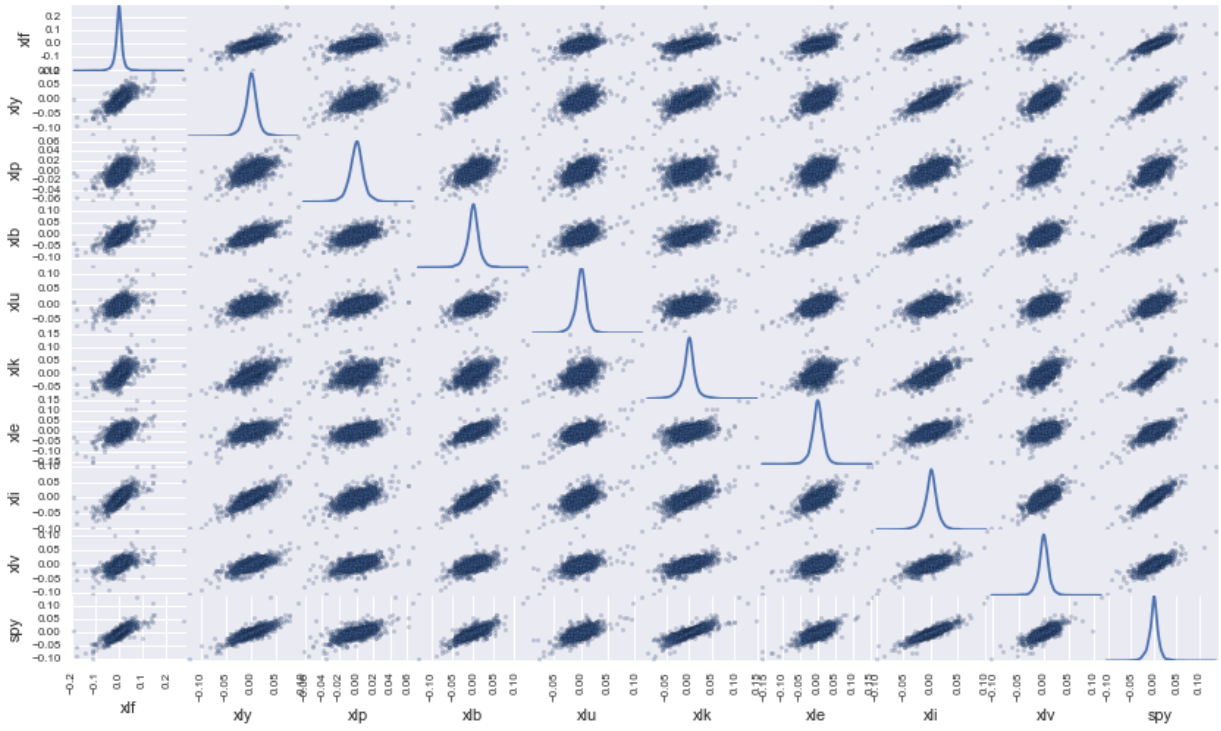
IIb: Data Visualization

To compare the returns for each sector visually, below is a chart with all returns indexed to have the same starting value on 2000-01-04. This date was picked because this was when the state street sector etf started trading. All nine sector ETF's had a gain over this period, however tech stocks (XLK) and financial stocks (XLF) were the poorest performers. Note that the start date is very near the peak of the technology boom, the aftermath of which pummelled tech stocks. Financial stocks (XLF) were strong performers going into the financial crises but have not fully recovered since.

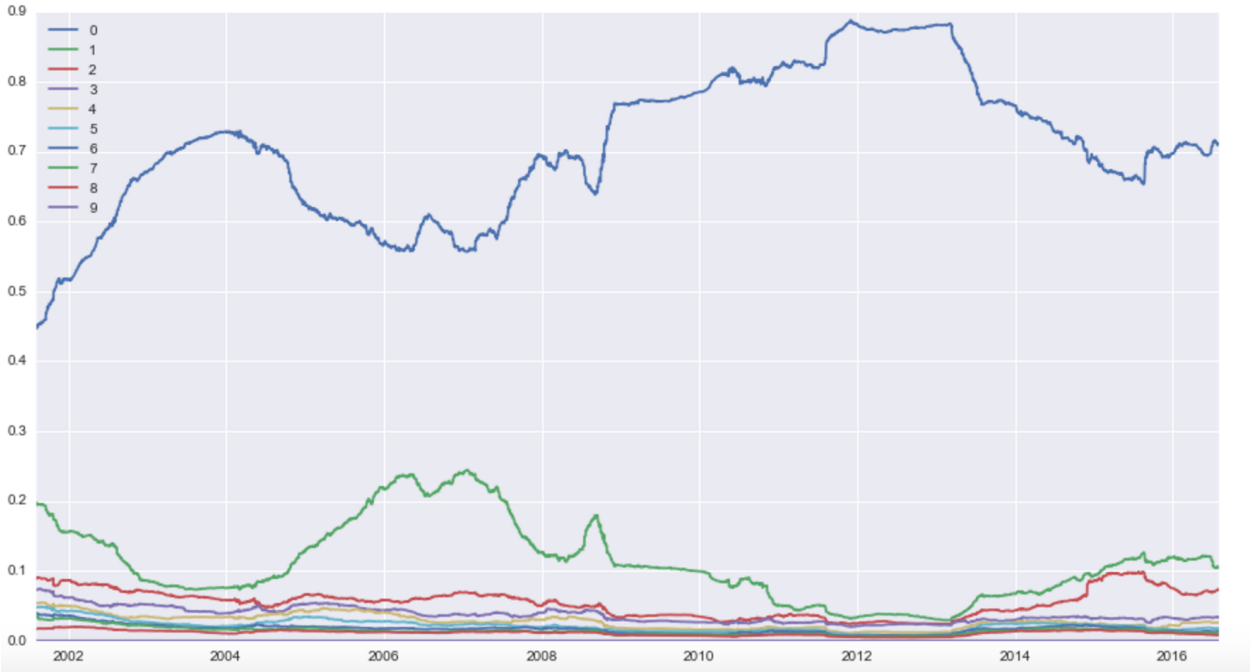
As we can see the ETF's are highly correlated with XLI "Industrials" being the most highly correlated with SPY at 0.90 and XLP "Consumer Staples" being the least correlated at .68.



Here is a scatter plot of each combination of the features. The colinearity of this feature set is illustrated by the up and to the right pattern in these plots.



Finally, within the PCA feature set, it is interesting to see how the proportion of explained variance by the first and second dimensions change over time. This graph uses the same 400 day training window used in the regressions.



IIc: Algorithms and Techniques

Two machine learning algorithms are used to generate one day forward predictions of SPY.

Principal Component Analysis (PCA) is an algorithm which takes a set of N features (in this case sector returns) and performs a linear transformation on them such that a new set of up to N features (the modeler can choose any number between 1 and N) is returned. These new features are sometimes referred to as PCA dimensions.

This new set of features has two desirable properties:

1. Each of the N new features is uncorrelated to all the other new features.
2. The new features can be ranked in terms of their ability to explain the overall variance of the original feature set.

Both of these properties can be useful for feeding features into the next step - linear regression.

Linear Regression is a modeling and prediction process that returns a set of coefficients which provide the best linear transformation between a set of features (PCA dimensions in this case) and a stated target (SPY returns). Best linear transformation in this case is the one which minimizes the squared difference between the regression's predictions and actual target values.

A focus of this report is to delve into the tradeoffs related to how many dimensions should be passed from PCA into the regression. As more dimensions are passed, each incremental dimension presumably adds some signal value from to the feature set but it also increases the level of overfit in the regression.

IId: Benchmark

On a daily basis, the probability that stocks will go up or down is very close to 50% either way, hence the benchmark metric for this paper is an annualized rate of return of zero.

To confirm, we did a test that goes long/short randomly or 50/50 on SPY. Using random darts to pick the long vs. short direction of the market in the simulator used for this report. As expected, after 100 random trials, the average return is close to of 0.52% per year (this gets closer to zero as # of trials increases).

IIIa Methodology - Data Preprocessing

All data for this study was sourced directly from Yahoo via Quandl's python API. The Features consist of SPY and 9 sector ETF's that collectively span all sectors in the SP500[1].

The data ranges from from 2000-01-04 to 2016-08-05 on a daily basis, a total of 4,174 data points.

Returns are computed as follows:

1. Prices used are adjusted for dividends via the adjusted_close field.
2. Returns are computed via difference of log prices - this dataframe is labeled dl

Next, three types of initial feature scaling options are explored:

1. DL = No scaling - just raw log returns
2. DLZ = A z-score of the difference of logs using a 252 day lookback window - this dataframe is labeled dlz.
3. DLE = Exponentially weighted moving average of DL using a 3-day half life.

The final preprocessing step is to set up the features and targets so that the features are properly lagging Targets by one day. For this I used patsy dmatrices. After using dmatrices for some time I discovered that it automatically inserts an Intercept column in the X variable. I choose to delete this column in the line after creating the X because I would rather control the use of an intercept in the regression explicitly in my simulator.

IIIb Methodology - Implementation

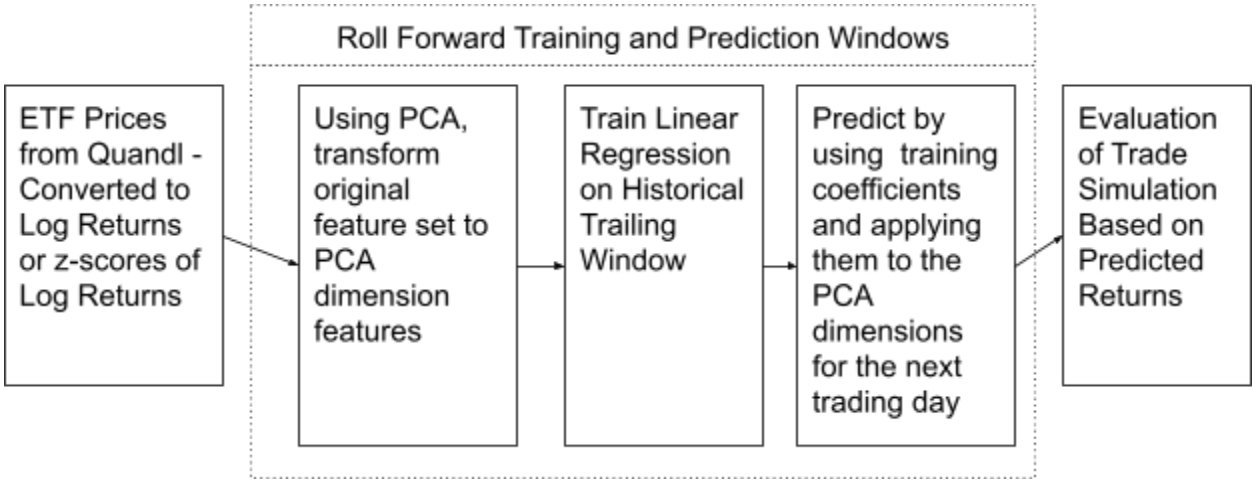
The methodology for implementing this study is designed to replicate the environment of a trader who needs the model re-trained every day and uses it to produce the next day's return. This hypothetical trader needs the ability to:

1. Measure the ETF returns
2. Update training sample for their PCA_Linreg model to make sure they have the most up to date model.
3. Unwind their prior day's position and trade into the market (just after the close SPY still trades) either long or short.

Key to the process is to use the most current market data while keeping all training for both PCA and linear regression modules limited to a historical time window ending just prior to date predicted.

To implement this we use a simulator that measures the out-of-sample trading sharpe for each choice of the number of dimensions to use. The simulator is designed to update its training sample on a fixed window, roll forward basis. The training window was fixed at 400 days.

The simulator uses an sklearn Pipeline to properly sequence 1) the PCA module and and 2) linear regression learner. The pipeline is called from a simulator that loops through a range of values, from top n=1 to top n=9, for the the n_components parameter of PCA. The simulator then computes out of sample trading results and stores the related performance metrics.



The PCA step uses the sklearn PCA dimensionality reduction module to generate linearly uncorrelated features that can be used in a regression. The key question at this point is how many dimensions (using the n_components parameter) should be passed from the PCA module as features in the Linear Regression. The simulator is design to range of over all combinations of top n dimensions where the top dimensions are ranked by their ability to explain the variance of the feature set.

The regression step uses sklearn linear regression which to train coefficients using an unrestricted Ordinary Least of Squares (OLS) linear estimator. I would have preferred to use the statsmodels linear regression estimator which has better summary output but it did not work in the pipeline framework.

The equations are set up so that the PCA dimension features are properly lagged:

$$Target_{T} = Constant + \beta * Top\ N\ PCA\ Dimension's_{T-1}$$

Predicted returns generated from the linear regression are then converted, via a trading rule, into out of sample strategy returns. The trading rule used is simple: if the prediction is for a gain in the SP500, the trading rule will go long the SP500 and vice versa if the prediction is for a loss.

The final scoring metric, return, for each test is based on applying a systematic trading strategy on these predictions.

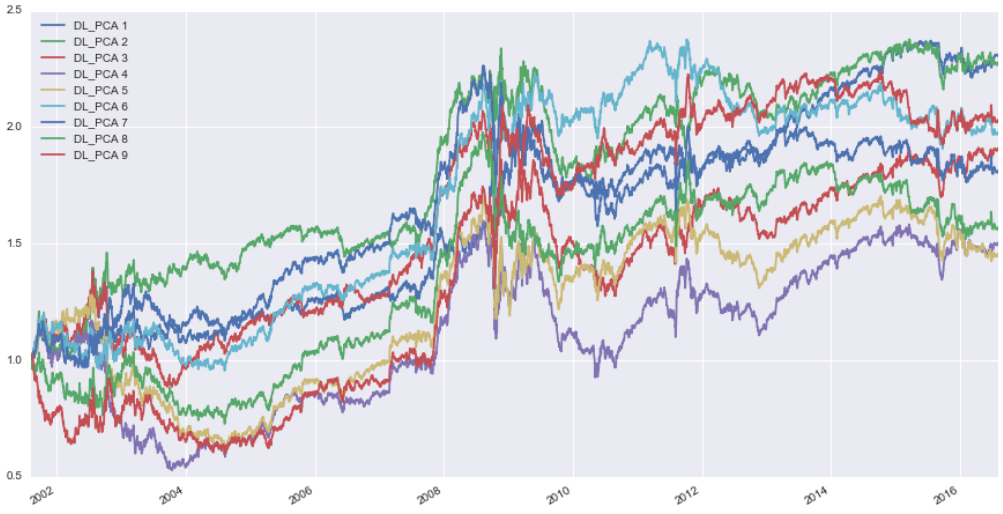
IIIc Methodology - Refinement

All the tests had the same target set - Log Returns on the SPY ETF which tracks the SP500. The focus for refining this model is to evaluate a) whether or not to z-score the feature set and b) how many PCA dimensions should be passed on to the regression.

Refinement - Results when using Log Returns only:

	DL_PCA 1	DL_PCA 2	DL_PCA 3	DL_PCA 4	DL_PCA 5	DL_PCA 6	DL_PCA 7	DL_PCA 8	DL_PCA 9
return	0.08738	0.0854	0.06088	0.03378	0.03063	0.06546	0.05432	0.03803	0.06884
z window	252	252	252	252	252	252	252	252	252
mae	0.008198	0.00821	0.008212	0.008225	0.008254	0.008275	0.008299	0.008321	0.008345
stdev	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965
beg_pred	2001-08-07	2001-08-07	2001-08-07	2001-08-07	2001-08-07	2001-08-07	2001-08-07	2001-08-07	2001-08-07
bench_std	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965	0.1965
R^2 OOS	-94.11	-48.52	-44.42	-38.8	-34.14	-29.08	-25.4	-23.46	-20.76
rmse	0.0124	0.01238	0.01241	0.01244	0.01249	0.01249	0.0125	0.01254	0.01255
end_pred	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04
bench_shp	0.2986	0.2986	0.2986	0.2986	0.2986	0.2986	0.2986	0.2986	0.2986
sharpe	0.4447	0.4347	0.3098	0.1719	0.1558	0.3331	0.2764	0.1935	0.3503
evs	-94.11	-48.52	-44.42	-38.8	-34.14	-29.08	-25.4	-23.46	-20.76
est window	400	400	400	400	400	400	400	400	400
bench_ret	0.05867	0.05867	0.05867	0.05867	0.05867	0.05867	0.05867	0.05867	0.05867

As expected, the only metric which varies from one test to the next is return (which is showing annualized return). There is very little variation in rmse or mae, and the variation in sharpe is driven by the variation in its return/numerator return component.



In looking at the out of sample trading performance of the non-z-scored feature set it is good to see that over the 16 year period the strategies finished with a gain. The best performance comes from passing just the first PCA dimension onto the linear regression.

Refinement - Results when using Z-scores on Log Returns:

Results when features are z-scored before PCA are generally better than without z-scoring but not in every case.

	DLZ_PCA 1	DLZ_PCA 2	DLZ_PCA 3	DLZ_PCA 4	DLZ_PCA 5	DLZ_PCA 6	DLZ_PCA 7	DLZ_PCA 8	DLZ_PCA 9
return	0.09709	0.1105	0.072	0.05638	0.06893	0.07939	0.06894	0.006223	0.06051
z window	252	252	252	252	252	252	252	252	252
mae	0.007983	0.007981	0.007999	0.008023	0.008043	0.008063	0.008081	0.008107	0.008108
stdev	0.1931	0.1931	0.1931	0.1932	0.1931	0.1931	0.1931	0.1932	0.1932
beg_pred	2002-08-12	2002-08-12	2002-08-12	2002-08-12	2002-08-12	2002-08-12	2002-08-12	2002-08-12	2002-08-12
bench_std	0.1931	0.1931	0.1931	0.1931	0.1931	0.1931	0.1931	0.1931	0.1931
R^2 OOS	-127.9	-60.7	-54.24	-47.42	-38.49	-35.27	-31.55	-29.33	-27.06
rmse	0.01219	0.01216	0.01219	0.01221	0.0122	0.01224	0.01225	0.01228	0.01228
end_pred	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04
bench_shp	0.4227	0.4227	0.4227	0.4227	0.4227	0.4227	0.4227	0.4227	0.4227
sharpe	0.5028	0.5721	0.3728	0.2919	0.3569	0.4111	0.3569	0.03221	0.3133
evs	-127.9	-60.7	-54.23	-47.42	-38.49	-35.27	-31.55	-29.33	-27.06
est window	400	400	400	400	400	400	400	400	400
bench_ret	0.08164	0.08164	0.08164	0.08164	0.08164	0.08164	0.08164	0.08164	0.08164

Z-scoring the features shifted the optimal configuration from passing one dimension to passing two dimensions on to the regression. The average annualized return of 11.05% is compelling however most of that return came in the 2008-2009 window which is characterized by high volatility.

Note the strange draw down when 8 z-scored dimensions are passed to the regression. This is concerning, suggesting that the model might be unstable to its parameter configurations.

Refinement - Results when using Exponential Weighted Moving Average on Log Returns:

The final feature scaling option explored is to adjust returns incorporate lags greater than one day by using an exponentially weighted moving average function set to a 3 day half life.

	DLE_PCA 1	DLE_PCA 2	DLE_PCA 3	DLE_PCA 4	DLE_PCA 5	DLE_PCA 6	DLE_PCA 7	DLE_PCA 8	DLE_PCA 9
return	0.04853	0.09621	0.1488	0.1039	0.1006	0.1246	0.09186	0.1044	0.1131
z window	252	252	252	252	252	252	252	252	252
mae	0.008202	0.008216	0.008222	0.008216	0.008246	0.008266	0.008265	0.008308	0.00835
stdev	0.1958	0.1958	0.1956	0.1958	0.1958	0.1957	0.1958	0.1958	0.1957
beg_pred	2001-08-22	2001-08-22	2001-08-22	2001-08-22	2001-08-22	2001-08-22	2001-08-22	2001-08-22	2001-08-22
bench_std	0.1958	0.1958	0.1958	0.1958	0.1958	0.1958	0.1958	0.1958	0.1958
R^2 OOS	-79.41	-61.27	-49.61	-33.41	-28.05	-26.79	-21.92	-19.83	-18.46
rmse	0.01244	0.01244	0.01246	0.01245	0.0125	0.01252	0.01249	0.01256	0.01261
end_pred	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04	2016-08-04
bench_shp	0.311	0.311	0.311	0.311	0.311	0.311	0.311	0.311	0.311
sim_time	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56

sharpe	0.2478	0.4914	0.7605	0.5309	0.5138	0.6368	0.4692	0.5333	0.5777
evs	-79.4	-61.26	-49.6	-33.41	-28.04	-26.79	-21.92	-19.83	-18.46
est window	400	400	400	400	400	400	400	400	400
bench_ret	0.0609	0.0609	0.0609	0.0609	0.0609	0.0609	0.0609	0.0609	0.0609

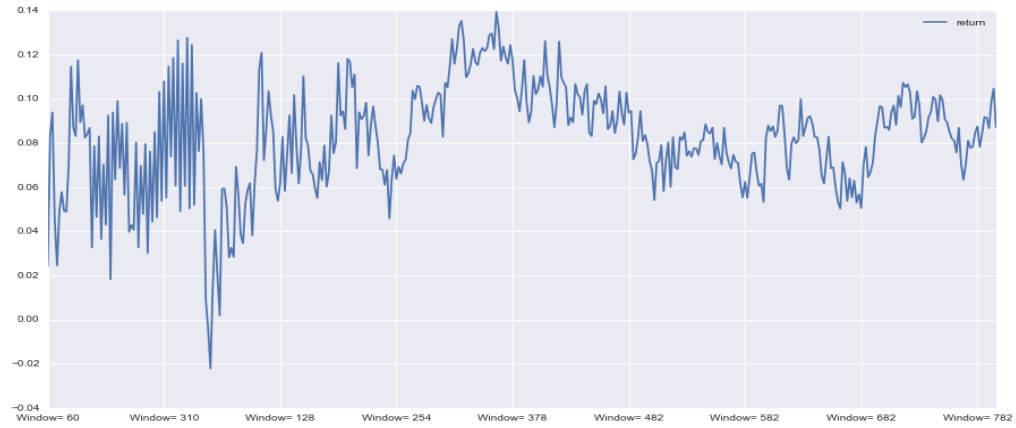
These are the best results yet! It is very interesting to see how the ewma smoothing of lags appears to greater explanatory power to the higher PCA dimensions. The best returns come with 3 dimensions which produces an annualized return of 14.88%.

Refinement - Testing for different Training Windows:

The initial window used for both PCA and regression was 400 days. With this window the best results were reached by using z-scored features and passing two dimensions from the PCA to the regression as features.

To refine the model further a test of the size of the trailing training window was conducted. The windows ranged from 60 to 800 days in increments of 2 days. The z-scored feature set was used and only 2 dimensions were passed on to the regression. Below is a graph of the average strategy return over a range of window sizes.

```
Peak results return    Window= 362
dtype: object with an average return of return    0.1396
dtype: float64
```



As we can see there is lots of volatility in the results for small windows, however, results are more consistently good for windows in the 300-400 range. Peak results are at 362 days which generated an average annual return of 13.96%. This is likely overfit, for my final model I decided to keep the window at 400 days.

IV Results

The final model recommended, DLE_PCA 3 uses z-scored features, a 400 day training window, and passes the three largest dimensions from PCA on as features in the linear regression (along with a constant term).

DLE_PCA 3 had an average annual return of 14.88% over the out-of-sample period from 2002-08-12 to 2016-08-04. This return is significantly higher than the non-pca benchmark return of 6.1% or the non-z-scored PCA with 2 dims of 8.54%.

	DL_PCA 2	DLZ_PCA 2	DLE_PCA 3	Raw_Features Benchmark
return	0.08472	0.09621	0.1488	0.06103
z window	252	252	252	252
mae	0.00821	0.008216	0.008222	0.008355
stdev	0.1957	0.1958	0.1956	0.1957
beg_pred	2001-08-07	2001-08-22	2001-08-22	2001-08-07
bench_std	0.1957	0.1958	0.1958	0.1957
R^2 OOS	-48.52	-61.27	-49.61	-22.07
rmse	0.01238	0.01244	0.01246	0.01254
end_pred	2016-08-04	2016-08-04	2016-08-04	2016-08-04
bench_shp	0.2974	0.311	0.311	0.2974
sim_time	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56	10/14/16 17:03:56
sharpe	0.4329	0.4914	0.7605	0.3118
evs	-48.52	-61.26	-49.6	-22.06
est window	400	400	400	400
bench_ret	0.05821	0.0609	0.0609	0.05821

Finally, The first two dimensions, which are used as features in the final model, shift a lot over time and were particularly volatile during the financial crisis.

V Conclusion

This Capstone project was intended to explore the use of PCA for converting potentially predictive features (sector returns) into a form more suitable as regression features for predicting the market.

The intuition is that PCA can take 9 highly correlated initial input features and reorganize them, through a linear transformation, into a set of 9 PCA dimensions which have two attractive properties: a) they are uncorrelated to each other and b) they can be sorted in terms of explained variance of the initial feature set. Both benefits help improve the regression by allowing it to fit to a reduced number of orthogonal features which contain almost as much information as the original feature set.

The challenge in applying machine learning tools to stock market prediction is that sklearn libraries are not well suited for financial time-series prediction. Unlike a standard random cross validation grid search, in finance the data must keep its time-series ordering, the training and prediction windows must be sequentially aligned with the prediction features always following after the training features.

Since Sklearn currently has no modules to do this properly, it is easy to make mistakes which cheat in the train/fit process by unintentionally looking into the future. To address this I built simulator from scratch which properly sets up the targets and feature sets for both training and prediction. The simulator also had to compute the returns of a strategy which follows the direction of the predicted returns. Finally, since I wanted to run 9

different regressions for each initial data type (dl or dlz) I needed a process to systematically capture the regression output and strategy returns so that they could be easily analyzed later.

The core idea of the simulator was to replicate what would be needed by a hypothetical trader who needs to use the model to go long the market if the model prediction is positive and short the market if it is negative. The expectation is that feature data this easy to acquire is unlikely to have any signal value on predicting up/down moves in the stock market.

Overall the keyline result of 11.05% when z-scored features are used and only two dimensions are passed to the regression is more favorable than I originally expected. As shown in the random darts test the benchmark return is zero. The standard deviation of 100 random trials is 5.02%, putting the tuned models returns of 11.05% about two standard deviations above no-signal zero return benchmark.

These results indicate that there might be some signal value for timing the SP500 by using the PCA on sector ETF returns to generate features for a linear regression. The results suggest that you only need to pass one to two dimensions from the PCA on to the linear regression. Dimensions 3,4,..9 explain very little variance of the feature set and adding them to the regression appeared to degrade the out of sample performance.

This model has numerous opportunities for further improvement. They include:

1. Add up/down confusion matrix and test classification learners.
2. Explore more feature scaling - add it to the pipeline. For example,trim extreme values from the feature set.
3. Add more history with the ewma function for weighted lagged regressors
4. Add new datasets to the feature set.
5. Explore adding lots of lags but controlling for overfitting by using bayesian priors on vector autoregressions, BVAR, to reduce spurious correlations in the covariance matrix.