Machine Learning Nano Degree Capstone Project

August 2016

1 Problem Statement

Trading stocks is complicated and challenging for individuals and investment firms. For a profitable strategy, one needs to identify the price to buy the securities and the price to sell the securities in the future. Despite this, investors are constantly reviewing previous prices history and using it to influence their future investment decisions. In this project, we apply the time series momentum strategy in paper [1] to predict stock returns. It would be trivial to back out future stock prices from its returns. Momentum has been a main factor for stock returns. The assumption is that the winners in the past are likely to be winners in the future. From behavior finance's perspective, the fear and greed of investors' are the drivers for this phenomenon. Why would one bet on losers instead of winners?

Broadly speaking, momentum can be further divided into cross-sectional momentum and time-series momentum. Cross-sectional momentum literature finds that securities that outperform their peers in the past are likely to continue the relative outperformance in the future.

In contrast, time series momentum purely focuses on the securities' own past returns. Academic research states that the time series momentum reflects the behavior finance theories of investors' initial under-reaction and delayed over-reaction to new informations.

We build a stock return predictor based on time series momentum [1] and both regression and clustering machine learning algorithms. The trading strategy has been backtested in a broad space of asset classes, such as equity futures, commodity futures and treasury bond futures. Results show that classification algorithms delivers a high Sharpe ratio and little correlation to passive benchmarks.

2 Metrics

To evaluate the fitness of the regressional learning models, root mean squared error (RMSE) is used. It indicates the absolute fit of the model to the data. Lower RMSE values indicate better fit. We use F1 score to evaluate the clustering algorithms, which is a measure of the precision and recall of the algorithm.

To evaluate the trading strategies, we use Sharpe ratio [2] to measure the return per unit risk. Sharpe ratio is defined as

$$Sharpe ratio = \frac{Annualized\ return}{Annualized\ standard\ deviation} \tag{1}$$

We also report the annualized return and annualized standard deviation, max return and max drawdown to evaluate the strategy from different perspective.

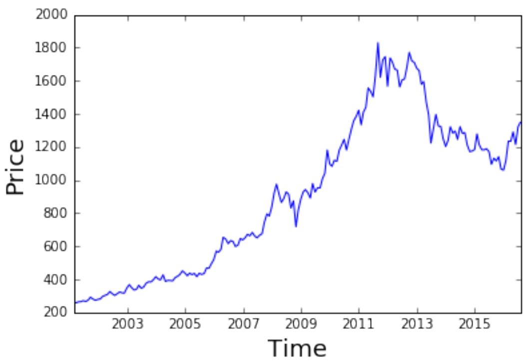
3 Data Exploration

We use various futures from different asset categories as the underlying security in our trading strategy including gold future, SPX future (ES1), tresury bond future (US1), oil future (CL1), currency pair (USDCAD), SPX Index and Yahoo US Equity. We summarize its descriptive statistics below. We observe that most futures have more positive returns than negative returns, though most of them have small Sharpe ratio. The exception is USDCAD, which have negative returns. However, we can trade CADUSD instead to make profits.

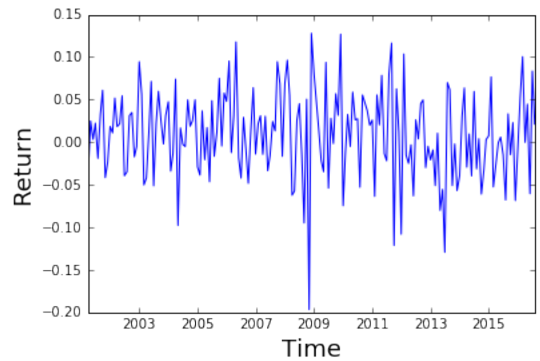
Asset	Gold	ES1	US1	CL1	USDCAD	SPX	Yahoo
Number of data points	185	185	185	185	185	185	185
Positive returns	103	112	100	102	84	113	100
Negative returns	82	73	85	83	101	72	84
Annualized returns	.11	0.04	0.03	0.03	-0.01	0.04	0.08
Standard deviation	0.17	0.15	0.11	0.32	0.096	0.149	0.42
Sharpe ratio	0.58	0.24	0.31	0.08	-0.11	0.24	0.18
Cumulative returns	403%	74.5%	65.2%	52%	-15.2%	75%	220%

4 Exploratory Visualization

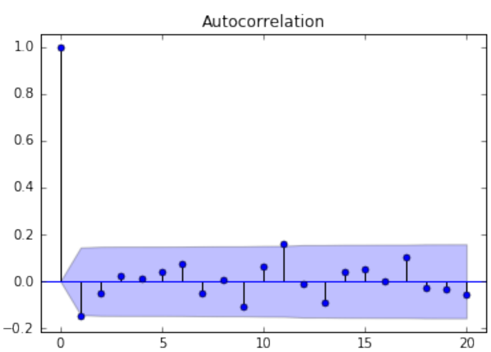
Historical prices and returns, autocorrelation plot and partial autocorrelation plot for selected securities are presented below. We observe that the price of most securities are increasing over time, though there are some big drawdowns sometimes. The returns of the securities are mean-reverting. Some lagged returns are highly correlated with current returns. Thus we use the past 12 months' monthly returns as the features of our predicting model. The mean of the returns is pretty arount 0 while the median has a high estimation error.



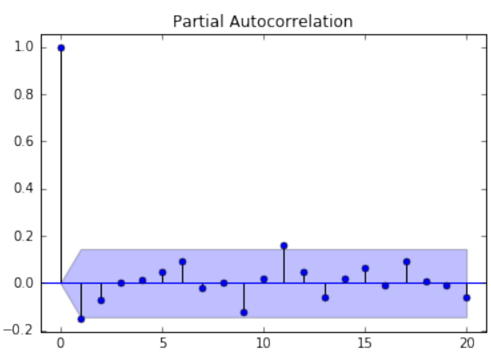
(a) Gold price



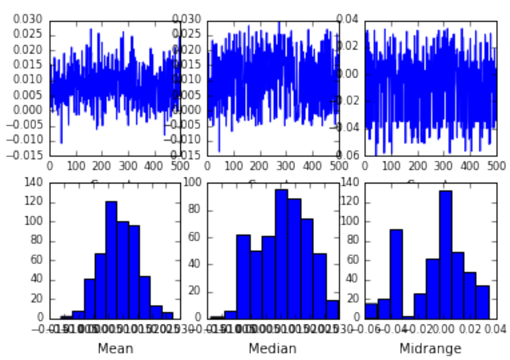
(b) Return of Gold



(c) Autocorrelation



(d) Partial autocorrelation



(e) Return of Gold

Figure 1: Gold data

From the features scatter plot, we could tell that the features are independent among each other.

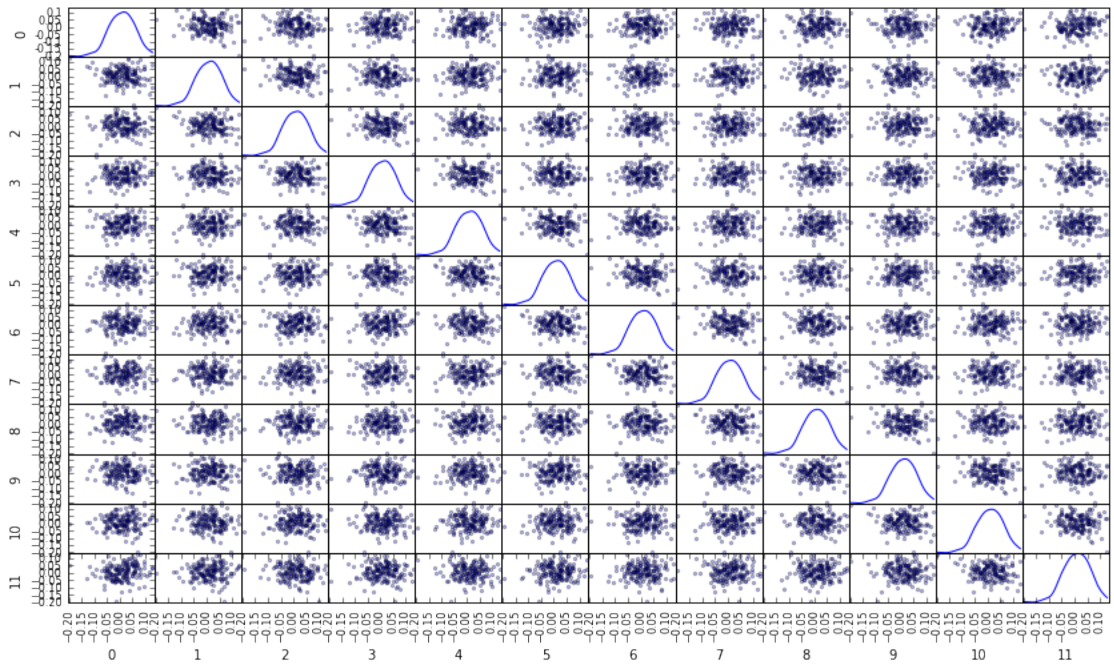
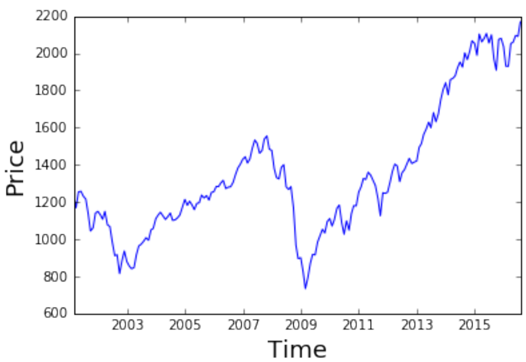


Figure 2: Feature scatter plot

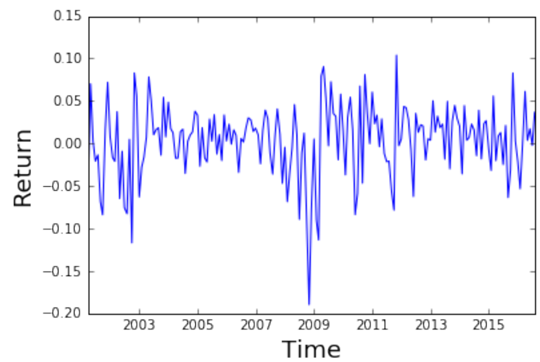
We show the price and return information of other securities' in future 3 and figure 4.

5 Algorithms and Techniques

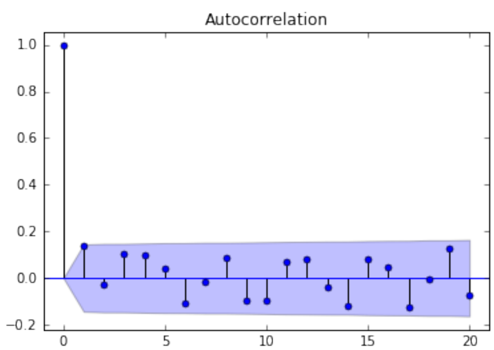
Even though all securities tested have positive returns over time, their Sharpe ratio are pretty low, meaning that investors need to tolerate a lot of risk in order to have the returns. Our objective is to leverage time series momentum theory and machine learning prediction models to identify the timing to long a security.



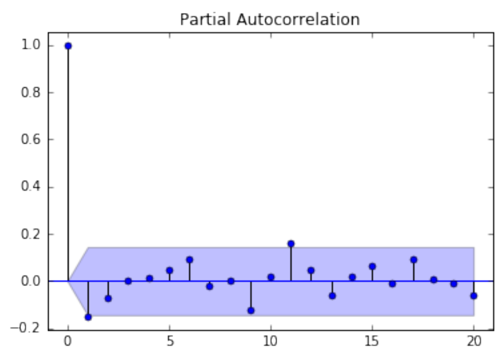
(a) ES future price



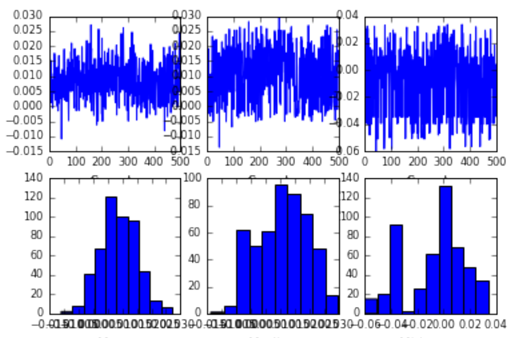
(b) ES future return



(c) Autocorrelation

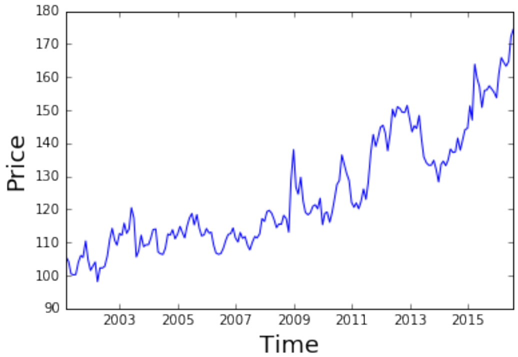


(d) Partial autocorrelation

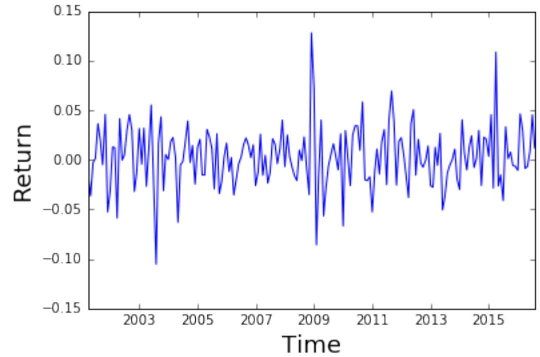


(e) Return of Es future distribution

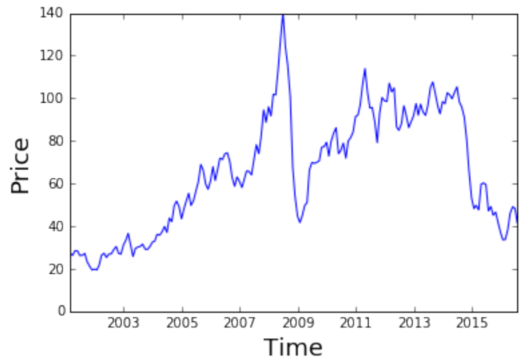
Figure 3: ES future data



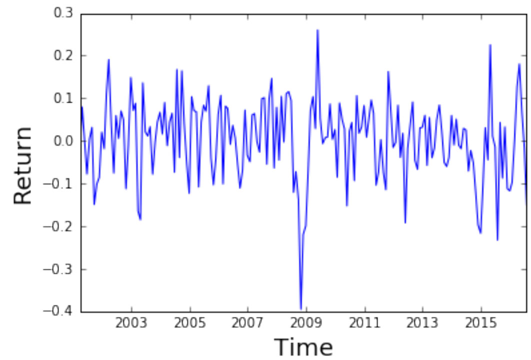
(a) US1 future price



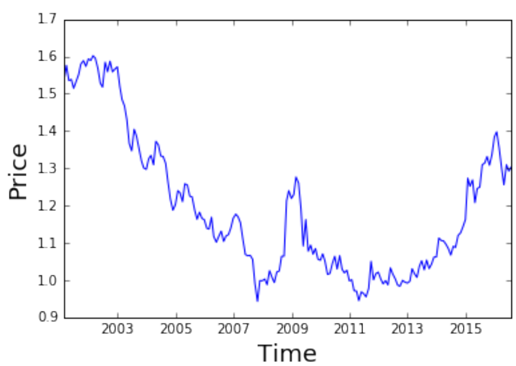
(b) US1 future return



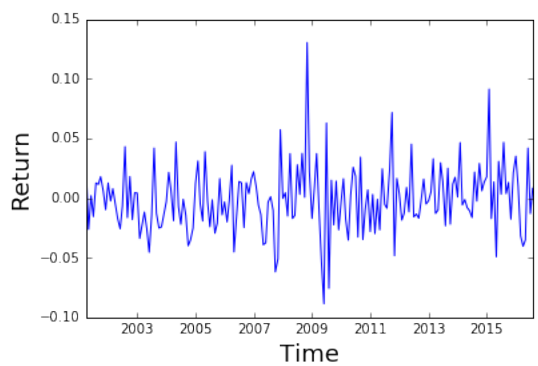
(c) CL1 future price



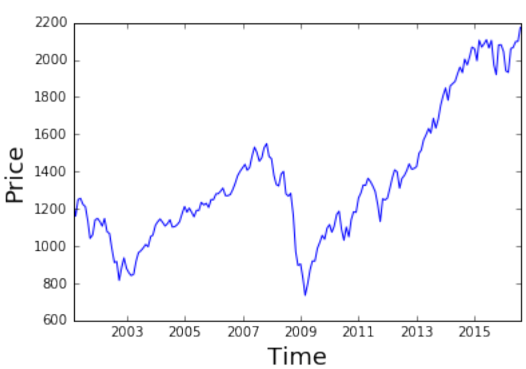
(d) CL1 future return



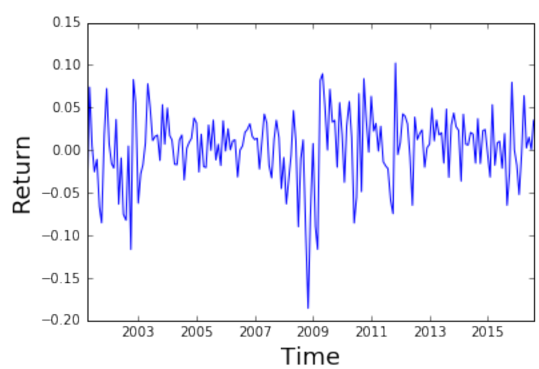
(e) USDCAD price



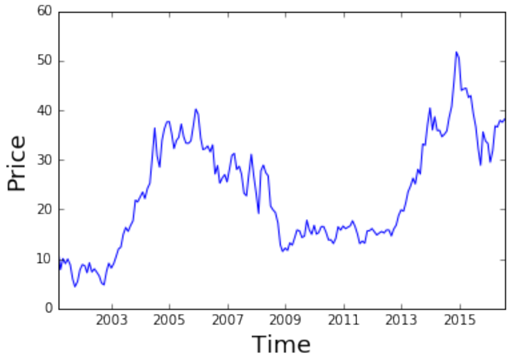
(f) USDCAD return



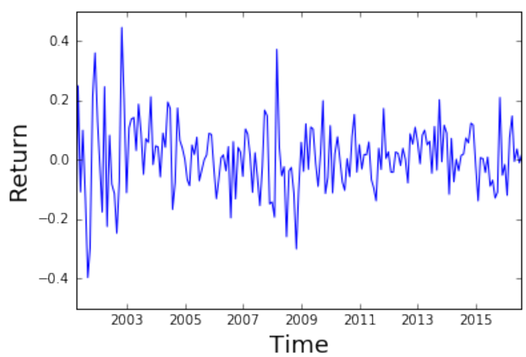
(g) SPX price



(h) SPX return



(i) Yahoo price



(j) Yahoo return

Figure 4: Other securites price and return

We use past 12 months' lagged returns to predict the return of next month. Both regression and classification models are used. The regression model is to predict the value of the future return. Long the security if the predicted value is positive, otherwise just held the cash. As what we need to know is only the sign of the future returns, we also applied classification models. Long the security if the predicted sign of future return is positive.

We use Linear regression, decision tree regressor and adaboost regressor and K neighbours classifier, Gaussian naive bayes and adaboost classifier.

Basically the predicting models take the view of following regressions:

$$r_t = \alpha + \beta_h r_{t-h} + \epsilon_t \tag{2}$$

and

$$sign(r_t) = \alpha + \beta_h r_{t-h} + \epsilon_t \tag{3}$$

We use linear regression as it is similar to what the paper [1] suggests. It is simple and explainable. We also believe decision tree regressor can add some nonlinearity to the data and thus may produce better result. Adaboost regressor is a ensamble learning model. It combines the merits of several model and thus we expect it to minimize overfitting effects and deliver good out of sample performance.

We also think to predict the sign of the returns may be better than predicting the values of the returns. So we also test classification models. K neighbours model is simple and non parametric, it does not make any assumption about the underlying data but it is sensitive to the local structure of the data. Naive bayesian classifier is based on Bayes theory with the assumption that all features are conditionally independent of each other, thus it also fits our dataset. For Adaboost model, we take advantage of its strength in reducing reducing bias and variance, converting weak classifiers to strong classifiers. Thus we also test it with our dataset.

We use a simple forward volatility model to determine the sizing of the portfolio as below:

$$\sigma_t^2 = 261 \sum_{i=0}^{\infty} w_i (r_{t-1} - \bar{r_t}^2) \tag{4}$$

where the weight function is selected to reflect the view the more recent variance are more correlated with future variance. And the sizing is $\frac{0.4}{\sigma_t}$ to minimize the overall volatility of the returns.

6 Benchmark

The benchmark for our algorithm is to long the security outright, meaning that we go long the security on the first day and sell it on the last day of the backtesting period. We also report the Sharpe ratio, annualized return, annualized standard deviation, max return and max drawdown of this passive strategy in Data Exploration section .

7 Implementation and refinement

The price data of the futures contract is downloaded from Bloomberg. We then calculate the log return of each date. Each monthly returns up to 12 months are the features of our model. And the dependent variable is the return for this month (regression model) or the sign of the return for this month (classification model)

We implement both regression and classification learning algorithm in our trading strategy in Python. Sklearn library is used in implementing the training model and cross validation. After the supervised model is trained, we compare the predictive return with a predefined threshold. We'll be making our bet only if the predictive return exceeds the threshold. This is to minimize noise.

```
def sup_trading_strat(predict_returns, y_test):
    total_returns = []
    for i in range(1, len(predict_returns)):
        vol = volatility_estimate(y_test[0:i])

```

```
        if predict_returns[i-1] >= threshold:
            total_returns.append(y_test[i] * 0.4 / vol)
        elif predict_returns[i-1] < -threshold:
            total_returns.append(-y_test[i] * 0.4 / vol)

    avg_ret = np.mean(total_returns) * 12
    avg_std = np.std(total_returns) * np.sqrt(12)
    sharpe = avg_ret/avg_std
    cum = np.cumsum(total_returns)
    return avg_ret, sharpe, cum
```

Listing 1: Trading strategy based on regression

8 Results

The results using both strategies are summarized in table 1 to 7. It shows that different models have different predicting power to different time series. For example, linear regression is good for ES future, while decision tree is good for SPX Index while adaboost regression generally have good performance across all securities. K neighbour model performs well on gold future. Gaussian naive bayes solves the Yahoo dataset well and adaboost works well on almost all securities. In general, the proposed strategies outperform the benchmark. By using the strategies, we can increase the Sharpe ratio to 0.8 to 1.8, meaning the strategies help investors make more returns while taking less risk.

Regressor	Linear regression	Decision tree	AdaBoostRegressor
RMSE	0.058	0.080	0.058
Avg return	0.053	0.067	0.06
Sharpe ratio	0.616	0.841	0.695

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.677	0.603	0.968
F1 score (out of sample)	0.509	0.627	0.577
Avg return	0.107	0.056	0.076
Sharpe ratio	1.484	0.706	1.075

Table 1: Trading strategy using gold

Regressor	Linear regression	Decision tree	AdaBoostRegressor
RMSE	0.043	0.062	0.038
Avg return	0.127	0.066	0.126
Sharpe ratio	1.448	0.836	1.314

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.802	0.733	0.987
F1 score (out of sample)	0.730	0.769	0.642
Avg return	0.086	0.067	0.059
Sharpe ratio	1.040	0.761	0.749

Table 2: Trading strategy using ES future

Regressor	Linear regression	Decision tree	AdaBoostRegressor
RMSE	0.035	0.038	0.036
Avg return	0.038	-0.048	0.023
Sharpe ratio	0.787	-0.593	0.454

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.689	0.721	0.985
F1 score (out of sample)	0.612	0.712	0.5
Avg return	-0.035	0.027	-0.01
Sharpe ratio	-0.459	0.372	-0.156

Table 3: Trading strategy using US1 future

Regressor	Linear regression	Decision tree	AdaBoostRegressor
RMSE	0.087	0.113	0.0832
Avg return	0.053	0.053	-0.009
Sharpe ratio	0.616	0.616	-0.193

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.606	0.687	1
F1 score (out of sample)	0.55	0.565	0.625
Avg return	0.031	-0.024	0.082
Sharpe ratio	0.57	-0.375	1.375

Table 4: Trading strategy using CL1 future

Regressor	Linear regression	Decision tree	AdaBoostRegressor
RMSE	0.026	0.046	0.0279
Avg return	-0.08	-0.011	-0.053
Sharpe ratio	-1.164	-0.099	-0.57

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.603	0.545	1
F1 score (out of sample)	0.438	0.4	0.457
Avg return	-0.036	-0.036	-0.107
Sharpe ratio	-0.77	-0.479	-1.8

Table 5: Trading strategy using USDCAD

Regressor	Linear regression	Gaussian Naive Bayes	AdaBoostRegressor
RMSE	0.042	0.056	0.038
Avg return	0.12	0.07	0.10
Sharpe ratio	1.32	0.93	1.47

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.82	0.73	0.98
F1 score (out of sample)	0.69	0.57	0.51
Avg return	0.06	0.03	0.06
Sharpe ratio	0.74	0.46	0.87

Table 6: Trading strategy using SPX

Regressor	Linear regression	Decision tree	AdaBoostRegressor
RMSE	0.109	0.153	0.095
Avg return	0.065	0.045	0.06
Sharpe ratio	1.03	0.727	0.94

Classifier	KNeighbour	Gaussian Naive Bayes	AdaBoostRegressor
F1 score (in sample)	0.73	0.74	1
F1 score (out of sample)	0.67	0.62	0.54
Avg return	0.078	0.096	0.075
Sharpe ratio	1.02	1.258	1.07

Table 7: Trading strategy using Yahoo

9 Conclusion

In this project, we study the time series momentum in various instruments. We apply both regression and classification models on two trading strategies that can enhance the Sharpe ratios. Results show that the proposed trading strategies significantly outperform the benchmark.

10 Reference

1. Moskowitz, Ooi, Pedersen: Time Series Momentum, Journal of Economics, 2010

2. https://en.wikipedia.org/wiki/Sharpe$_{r}atio$