Final Report

Colin Eberl-Coe, Alyssa Makhani, Andrew Zerwick

08 May 2014

1 Summary of the Data

MWD Trading, LLC provided us with data from four days of their trading in the market of Light Sweet Crude Oil futures. These data are manifestations of the price ladder, the visual tool that high-frequency traders implement to make sense of the market. As seen below, the blue corresponds to the bid side of the book, and the red corresponds to the offer side of the book.

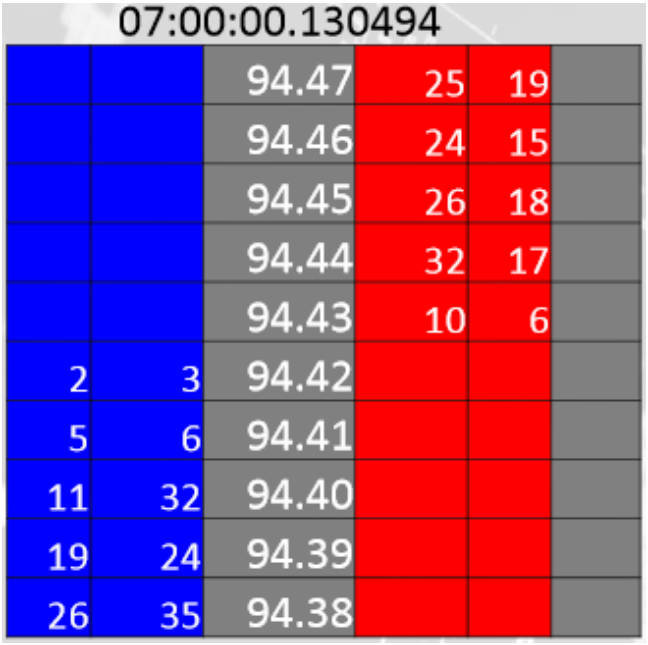

Figure 1: In this price ladder, depicted are the various offers and bids, their quantities, and the distribution of their quantities at 130 milliseconds past 7:00 AM.

For example, on the offer side, the lowest offer is \$94.43, of which there are 10 contracts, consisting of 6 different orders. Connecting these numbers to what actually happens on Wall Street, it could be that one party offered five contracts of \$94.43, and five other parties offered one contract each (5-1-1-1-1-1). The permutation is unimportant, as there is no way to (and no reason to) discern the distribution.

Therefore, the time series data given to us by MWD consisted of 36 variables such as `BidPrice 1-5`, `OfferPrice 1-5`, `BidNumberofOrders 1-5`, `OfferNumberofOrders 1-5`, `TradeQuantity`, and the most important, `MicroPrice`. As the trading day lasts 15 hours, each day of trading contained roughly 850,000 rows: in total, the size of the training dataset available to us was 3.7 million by 36; a veritable mountain of data to sift.

2 The Task

`MicroPrice` is the quantity of value to MWD, as they have found that it is a reasonably good predictor of the direction of the market. It is a weighted average of the inside bid and inside offer; "inside" referring to the lowest offer and the highest bid, corresponding to \$94.43 and \$94.42 in the above price ladder, respectively. Mathematically speaking,

$$\text{MicroPrice} = \frac{\text{BidPrice1} * \text{OfferQuantity1} + \text{OfferPrice1} * \text{BidQuantity1}}{\text{BidQuantity1} + \text{OfferQuantity1}}$$

The task set to us by MWD was to employ statistical models to predict this quantity in 1 second and in 60 seconds in the future. However, only the rows in which one of the four constituent variables of `MicroPrice` changes were to be used to make forecasts. This meant that the size of the training data was reduced to 1.2 million rows by 36 columns, still a workably large data set for statistical inference.

2.1 RMSE

But how to judge the predictive power of our models? After discussion with MWD, it was decided that the RMSE (Root Mean Square Error) would be used as the numerical criterion. RMSE is defined as

$$\text{RMSE} = \sqrt{\sum_{t=1}^{n} \left(\frac{\hat{x_t} - x_t}{n} \right)^2}$$

where $\hat{x_t}$ are the predicted values at time t and x_t are the actual values at time t. The first results of the RMSE were on the thousandths scale, so when the RMSE is reported hereafter, it is referring to RMSE $* 1000$.

2.2 Persistence Model

When attempting to predict a time-dependent process such as high-frequency stock trading, persistence is usually selected as a benchmark. In our case, persistence means that the prediction of `MicroPrice` for time $t+1$ and $t+60$ will be the `MicroPrice` at time t, the current time. Therefore, the RMSE values of the 1-second and 60-second persistence models provided the benchmark for our competition. Mathematically, persistence takes the form

$$\hat{y}_{t+1} = y_t$$
$$\hat{y}_{t+60} = y_t$$

where $\hat{y}_{t+1}$ and $\hat{y}_{t+60}$ are the forecasted values at time $t+1$ and time $t+60$, respectively, and y_t is the actual current `MicroPrice`. This proved to be a formidable target, as only two groups out of seven in the class managed to achieve lower errors than persistence. We were one of those groups, so our methodology and results will be discussed below.

3 Our Models

After trying clustering and neural networks with varying degrees of success throughout the semester, we decided to use linear models for two reasons: their simplicity, i.e. short computation time, and secondly for the reason that the quantity that we are trying to predict is a linear combination. Therefore, our models are both linear models, dictinct in number of predictor in order to represent the difference between the time horizons:

$$\hat{y}_1 = \beta_0 + \beta_1(MP_{current}) + \beta_2(\text{wtdmp}) + \beta_3(\text{mp5}) + \epsilon, \qquad \text{where } \epsilon \sim N(0, \sigma^2)$$
$$\hat{y}_{60} = \beta_0 + \beta_1(\text{wtdmp}) + \beta_2(\text{mp5}) + \epsilon, \qquad \text{where } \epsilon \sim N(0, \sigma^2).$$

3.1 One-Second Model

First, the 1-second model will be constructed. We will discuss how we came to use the predictors we did, and document the performance of the model using the Monday and Tuesday data as training and the Wednesday and Thursday data as testing.

3.1.1 Predictors

Current `MicroPrice` is obviously a predictor to be employed, as it has shown its predictive power in the persistence model. Then, as trades occur when the innermost bids and offers agree, MicroPrice is a predictor of that "inner space" between the middle rungs of the price ladder. Essentially, this "between-rung" information can be given by a variety of averages. How about the arithmetic mean? Using $\text{Mean}_{arith} = \frac{1}{2}$(`BidPrice1` + `OfferPrice1`) as the only predictor in a linear model to predict `MicroPrice1SecAhead`, the persistence score of 7.566 is not beaten, although Mean_{arith} is a good predictor on the 1-second timescale.

```
> RMSE          > Call:          >Pr(>|t|)
[1] 8.008678    > lm(s1~av.mp)   >2e-16 ***
```

Figure 2: RMSE, model call and p-value of t-test

So arithmetic mean on its own is too simplistic to explain the movement of `MicroPrice` better than persistence. Will predictive power be increased if a two-predictor model is used instead? Regressing `MicroPrice1SecAhead` on Mean_{arith} and current `MicroPrice`, the RMSE is significantly lower:

```
> RMSE          > Call:
[1] 7.727655    > lm(s1~av.mp + MicroPrice)
```

Figure 3: RMSE of two-predictor model

This is quite close to beating the benchmark score. How can it be pushed that vital bit lower? Considering the dynamic of the market, its direction will be persuaded by the "heavier" side of the book, so equal contributions from both inside prices is not quite appropriate. Therefore, the next step is to try a weighted average. Weighted MicroPrice, or `wtdmp`, will be similar to Mean_{arith}, differing in that it is a sum of weighted price terms. Explicitly,

$$\text{wtdmp} = \frac{\text{BidPrice1} * \text{BidQuantity1} + \text{OfferPrice1} * \text{OfferQuantity1}}{\text{BidQuantity1} + \text{OfferQuantity1}}$$

Is it reasonable to expect that this will have good predictive power? Based on the results of the t-test, there is zero chance that this predictor does not influence the response variable. Thus, to try to improve upon the arithmetic mean and current `MicroPrice` model, the two-predictor model of `wtdmp` and current `MicroPrice` is tested:

```
> result.fit     > Call:
[1] 7.676065     > lm(s1~ wtdmp + MicroPrice)
```

Figure 4: RMSE of two-predictor model

At this point, it behooved us to search for a third predictor, because subsequent attempts to lower the RMSE were unfruitful. We decided to use findings from the initial exploratory analysis, that is, incorporating more rungs of the price ladder. Essentially, even though the inside bid and offer drive the market, the bids higher in the book might influence where the market is heading. So, mp5 is an extended `MicroPrice` using as much of the price ladder as was made available by MWD. Mathematically,

$$\text{mp5} = \frac{\sum_{i=2}^{5}(OP_i)(BQ_i) + (OQ_i)(BP_i)}{\sum_{i=2}^{5}(OQ_i) + (BQ_i)}$$

where OP is OfferPrice, BQ is BidQuantity, OQ is OfferQuantity, and BP is BidPrice.

The definition of this predictor begs a few questions: why do the sums start at 2? and why use all five of the price rungs? To answer the first question, the $i = 1$th term of the above expression yields `MicroPrice`, which is already a predictor in the model: to avoid collinearity among predictors, the sums start at $i = 2$.

The reason for the inclusion of all of the available rows of the price ladder requires a look into our methodology. We started with `MicroPrice2`, which was the prescribed weighted sum of the best two inside bids and offers. Then, `MicroPrice3` was tested, and it performed better than `MicroPrice2`; similarly `MicroPrice4` performed better, until at last `MicroPrice5` was the predictor that, when included in the model, gave the lowest RMSE, lower than persistence, even.

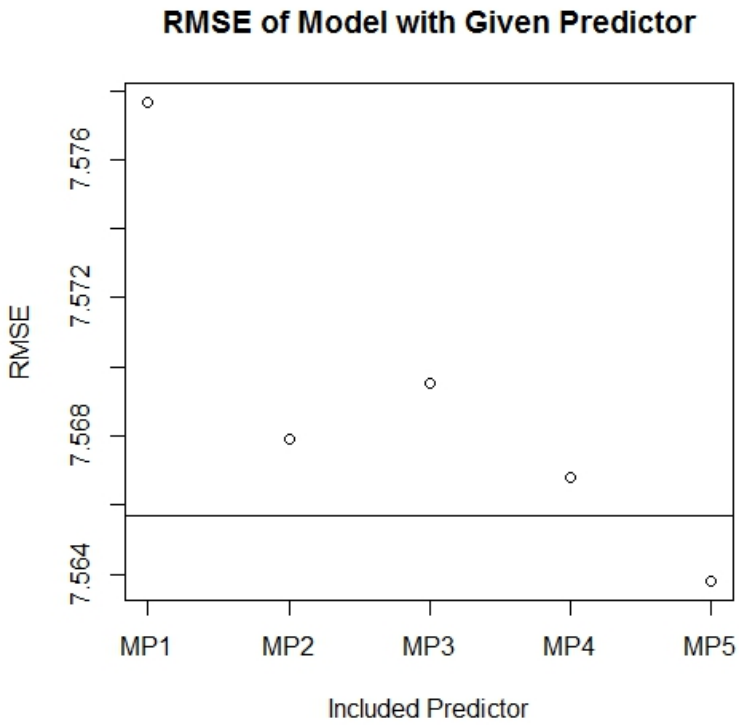

Figure 5: RMSE declines to the point that the model with MP5 beats persistence

We've arrived at a model that beats persistence on the 1-second timsecale based on the training data. It is yet to be seen whether this model will perform well on the held-out day's worth of data provided for the competition, but first we will look into the model used for the 60-second timescale.

3.2 Sixty-Second Model

As the larger monetary prize was awarded to the best model on the 1-second timescale, our efforts were primarily focused on refining that model. Therefore, we adopted the same predictors for the 60-second model, but with one modification. We found that the omission of current MicroPrice resulted in a lower RMSE:

4 Evaluation & Remarks

	Persistence	Our Models	Improvement
1 Sec Ahead	7.565705	7.563811	0.002894
60 Sec Ahead	42.61019	42.57	0.04

Our model was successful in beating the RMSEs for the persistence model one second ahead, but unsuccessful in beating persistence for 60 seconds ahead. This was a result of using Monday, Tuesday, Wednesday, and Thursday as the training data.

In our attempts to find a model that beat a persistence model, we also tried multiple forms of clustering, Neural Networks, Principle Component analysis to make the most effective predictive model. Overall, we found that linear models proved to be the most effective in forecasting microprice. With more time, our team would continue trying different linear models, altering the number of predictors while also performing different transformations on each of the raw predictors.