CSE 515T Project Status Report

Summary

My project is starting from reproducing a paper (attached in the same repo). I did several modifications due to there are typos and confusing notations. In general, I like his data features and his simple model. I also experimented LSTM besides his linear regression, but I don't have time to include LSTM part into this report. I also made a visualization tool to show the predicting price dynamically as time goes, which I plan to show in presentation. The future work would be experimenting with GP to predict a price interval instead of a single price.

2. BTC Price Data

This dataset is collected from GDAX order book by a python crawler I wrote myself. It was running on my AWS instance for one month and this is how I collected my dataset. I am focusing on 4 features: timestamp, last trade price, sum of top 20 bid order volume, and sum of top 20 ask order volume. Here I attached a picture of a GDAX (Coinbase) to demonstrate the bid and ask volume. The data was acquired at the interval of every 2 seconds.

Each data point contains timestamp, price, bid volume, ask volume.

In[7]:=
```
$TimestampIndex = 1;
$PriceIndex = 2;
$BidVolumeIndex = 3;
$AskVolumeIndex = 4;
```

In[13]:=
```
$PriceData = BTC Price Data ;
```

In[∘]:=
```
$PriceData
```

Out[∘]=

Tue 20 Mar 2018 11:45:50	8542.33	77.5961	45.5803
Tue 20 Mar 2018 11:46:00	8541.88	43.6552	43.4328
Tue 20 Mar 2018 11:46:10	8540.93	46.804	38.138
Tue 20 Mar 2018 11:46:20	8540.93	68.045	24.9296
Tue 20 Mar 2018 11:46:30	8540.92	65.2949	24.8595
Tue 20 Mar 2018 11:46:40	8540.93	64.9794	28.3992
Tue 20 Mar 2018 11:46:50	8541.16	91.3614	16.4277
Tue 20 Mar 2018 11:47:00	8545.27	121.182	15.6874
Tue 20 Mar 2018 11:47:10	8550.	115.784	27.5744
Tue 20 Mar 2018 11:47:20	8549.99	120.403	25.014
Tue 20 Mar 2018 11:47:30	8550.	123.143	21.7505
Tue 20 Mar 2018 11:47:40	8548.99	77.7999	52.3667
Tue 20 Mar 2018 11:47:50	8540.94	39.9577	58.7764
Tue 20 Mar 2018 11:48:00	8540.23	28.0957	63.8217
Tue 20 Mar 2018 11:48:10	8540.05	25.7413	69.0411
Tue 20 Mar 2018 11:48:20	8540.05	31.6011	68.2489
Tue 20 Mar 2018 11:48:30	8540.05	44.2359	47.7615
Tue 20 Mar 2018 11:48:40	8540.05	42.0424	43.3803
Tue 20 Mar 2018 11:48:50	8540.05	42.1052	47.7809
Tue 20 Mar 2018 11:49:00	8540.06	44.0974	50.9071

showing 1–20 of 79 582

Filter out missing data points.

In[12]:=
```
$PriceData = Select[FreeQ[#, Missing] &] @ $PriceData;
```

3. Splitting the data into three parts

I split the dataset into 3 parts, the pattern data is used to do k-cluster means to find the most frequent patterns, the training data is used to train the model and the testing data is going to be used in testing stage.

In[14]:=
```
{$PricePatternData, $PriceTrainingData, $PriceTestingData} =
   Partition[Normal @ $PriceData, UpTo[Ceiling[Length[$PriceData] / 3]]];
```

3. Model

To predict label y given associated observation x, we can utilize the conditional distribution of y given x given as follows:

$$P(y \mid x) = \sum_{k=1}^{T} P(y \mid x,\ T=k)\, P(T=k \mid x)$$

$$= \sum_{k=1}^{T} P(y \mid x,\ T=k)\, P(x \mid T=k)\ P(T=k)$$

$$= \sum_{k=1}^{T} P_k(y)\, P(\epsilon = (x - s_k))\, \mu_k =$$

$$\sum_{k=1}^{T} P_k(y) \exp\left(\tfrac{-1}{2} \|x - s_k\|_2^2\right) \mu_k \ (3)$$

We want to utilize empirical data as proxy for estimating conditional distribution of y given x given in (3), Specifically, given n data points $(x_i,\ y_i)$, 1 <= i <= n, the empirical conditional probability is:

$$P_{\text{emp}}(y \mid x) = \frac{\sum_{i=1}^{n} \mathbb{I}(y=y_i) \exp\left(\frac{-1}{4} \|x-x_i\|^2\right)}{\sum_{i=1}^{n} \exp\left(\frac{-1}{4} \|x-x_i\|^2\right)} \ (4)$$

Estimation of the conditional expectation of y, given observation x:

$$E_{\text{emp}}(y \mid x) = \frac{\sum_{i=1}^{n} y_i \exp\left(\frac{-1}{4} \|x-x_i\|^2\right)}{\sum_{i=1}^{n} \exp\left(\frac{-1}{4} \|x-x_i\|^2\right)} \ (6)$$

This estimation can be viewed equivalently as a linear estimator, let vector X(z) $\in \mathbb{R}^n$ be such that $X(z)_\alpha = \exp\left(\frac{-1}{4} \|z - x_\alpha\|^2\right)$ /Z(x) with $Z(x) = \exp\left(\frac{-1}{4} \|z - x_\alpha\|^2\right)$, and $y \in \mathbb{R}^n$ with i-th component being y_α then $\hat{y} = E_{\text{emp}}[y \mid z]$ is

$$\hat{y} = X_\alpha(z)\, y_\alpha \ (7)$$

(7) is used for predicting future variation in the price of Bitcoin.

5. Price series similarity function

We need to implement the following function:

$$s(a, b) = \frac{\sum_{z=1}^{M} (a_z - \text{mean}(a))\,(b_z - \text{mean}(b))}{M\, \text{std}(a)\, \text{std}(b)}. \qquad (1)$$

Note that we define std differently that in [XXX], because we believe there is a mistake made in equation for std right after [XXX, Equation 9].

In[15]:=
```
$STD[a_List] := Total[(a - Mean[a])^2] / Length[a]
```

In[16]:=
```
PriceListSimilarity::std0 = "std equals to zero. Price list similarity is set to ∞.";
PriceListSimilarity[a_List, b_List] /; Length[a] == Length[b] := Module[
    {res},
    If[$STD[a] == 0 || $STD[b] == 0,
        Message[PriceListSimilarity::std0];
        ((a - Mean[a]).(b - Mean[b]) + 1) / (Length[a] Sqrt[$STD[a] $STD[b] + 1]),
        ((a - Mean[a]).(b - Mean[b])) / (Length[a] Sqrt[$STD[a] $STD[b]])
    ]
]
```

6. Price change vectors

We need to find a vector of price changes starting at a given index *i* and going *d* indices into the past.

In[18]:=
```
$LastLengthIndexConsistentQ[data_List, last_Integer, length_Integer] :=
    last ≤ Length[data] && last - length + 1 ≥ 1
```

In[19]:=
```
PriceChangeList[data_List, last_Integer, length_Integer] /;
    $LastLengthIndexConsistentQ[data, last, length] :=
    data[[last - length + 1 ;; last, $PriceIndex]]
```

In[20]:=
```
DataChangeList[data_List, last_Integer, length_Integer] /;
    $LastLengthIndexConsistentQ[data, last, length] :=
    data[[last - length + 1 ;; last, 2 ;; 4]]
```

7. Price change prediction

Next, we implement [XXX, Equation 6] to predict the price change at a given time.

In[ ]:=
```
(* PriceListSimilarity[recentPriceChangeList, PriceChangeList[$PricePatternData, t, depth]
```

Note the following Equation 6 / 7 is abandoned

In[21]:=
```
ClearAll[PriceChangePrediction];
PriceChangePrediction[index_Integer, depth_Integer] /;
    $LastLengthIndexConsistentQ[$PriceTrainingData, index, depth] := Module[{
        recentPriceChangeList = PriceChangeList[$PriceTrainingData, index, depth],
        priceSum
    },
    priceSum[arg_] := Sum[
        arg[t] Exp[-Total[(recentPriceChangeList - PriceChangeList[$PricePatternData, t,
        {t, depth, Length @ $PricePatternData - depth}
    ];
    priceSum[$PricePatternData[[# + depth, $PriceIndex]] &]
    ─────────────────────────────────────────────────────
                        priceSum[1 &]
]
```

In[23]:=
```
MakeTrainingSet[depth_Integer] :=
    Table[DataChangeList[$PriceTrainingData, index, depth] → $PriceTrainingData[[index + 1
```

In[24]:=
```
PriceList[data_, model_, depth_] := Table[model[data[[index - depth;; index - 1, 2 ;; 4]]],
```

We use this historic time series and from it, generate 3 subsets of time-series data of three different lengths: S_1 of time-length 30 minutes, S_2 of time-length 60 minutes, S_3 of time-length 120 minutes. Now at a given point of time, to predict the future change δp, we use the historical data of 3 length: previous 20 minutes, 60 minutes, and 120 minutes.

In[25]:=
```
training180 = MakeTrainingSet[180]
```

Out[25]=
```
{ ⋯ 1 ⋯ }
```
large output | show less | show more | show all | set size limit...

In[26]:=
```
training360 = MakeTrainingSet[360];
training720 = MakeTrainingSet[720];
```

In[28]:=
```
p1 = Predict[training180]
```

In[ ]:=
```
p2 = Predict[training360, Method → "LinearRegression"]
```

Out[ ]=
```
PredictorFunction[ Input type: NumericalTensor (size: 360 × 3)
                   Method: LinearRegression ]
```

In[ ]:=
```
p3 = Predict[training720, Method → "LinearRegression"]
```

Out[ ]=
```
PredictorFunction[ Input type: NumericalTensor (size: 720 × 3)
                   Method: LinearRegression ]
```

In[∘]:=
```
Length /@ {training180, training360, training720, $PriceTrainingData}
```
Out[∘]=
```
{26 332, 26 152, 25 792, 26 513}
```

In[∘]:=
```
predP1 = PriceList[$PricePatternData, p1, 180]
```
Out[∘]=
```
{8527.05, 8567.42, 8537.64, 8512.53, 8515.35, 8529.25, 8518.78, 8529.66,
 8571.65, 8562.36, 8492.1, 8510.93, 8530.53, 8550.39, 8545.92, 8586.36,
 8544.6, 8568.62, 8538.67, … 26 295 …, 9011.02, 9033.39, 9006.34,
 9034.01, 9019.03, 9019.45, 9027.29, 9015.12, 9038.77, 9012.52, 9021.46,
 9020.58, 9030.12, 9011.04, 9009.55, 9029.21, 9031.73, 9017.32, 9038.31}
```
large output | show less | show more | show all | set size limit...

In[∘]:=
```
predP2 = PriceList[$PricePatternData, p2, 360]
```
Out[∘]=
```
$Aborted
```

In[∘]:=
```
predP3 = PriceList[$PricePatternData, p3, 720];
```

In[∘]:=
```
Differences[Length /@ {predP1, predP2, predP3}]
```
Out[∘]=
```
{-180, -360}
```

In[∘]:=
```
deltaP1 = Table[
  predP1[[360 + index]] - $PricePatternData[[720 + index, 2]], {index, 1, Length[predP3]}]
```
Out[∘]=
```
{-28.1871, 1.96292, -10.2679, -39.7582, 45.8628, 13.7248, -15.6196,
 7.02611, -69.6198, -5.04418, 29.4099, 21.8228, 3.68496, -17.9536,
 -19.8564, 12.7679, 25.6585, -58.2043, … 25 757 …, -2.08116, 8.73713,
 11.3227, 6.36002, 14.8596, -6.01196, 7.05375, 22.6428, 11.8996, 7.79631,
 34.8131, 14.9521, -11.8027, -4.10739, 41.0594, 11.2632, 11.6908, 14.1857}
```
large output | show less | show more | show all | set size limit...

In[∘]:=
```
deltaP2 = Table[
  predP2[[180 + index]] - $PricePatternData[[720 + index, 2]], {index, 1, Length[predP3]}]
```
Out[∘]=
```
{9.53842, 8.78072, 8.41194, 6.15173, 2.73472, 1.01112, 6.13385, 7.72086,
 6.43846, 5.78349, 9.22075, 7.68334, 7.23501, 5.84556, 5.70084, 2.27566,
 -1.36166, -11.1247, … 25 757 …, -2.93861, -1.01873, -0.0359387, 0.390862,
 -0.0166583, 0.179048, 0.253025, 0.707646, 0.4668, 1.29189, 1.85157,
 0.865075, 0.339871, -0.655324, -1.45163, -2.10745, -2.47959, -3.3806}
```
large output | show less | show more | show all | set size limit...

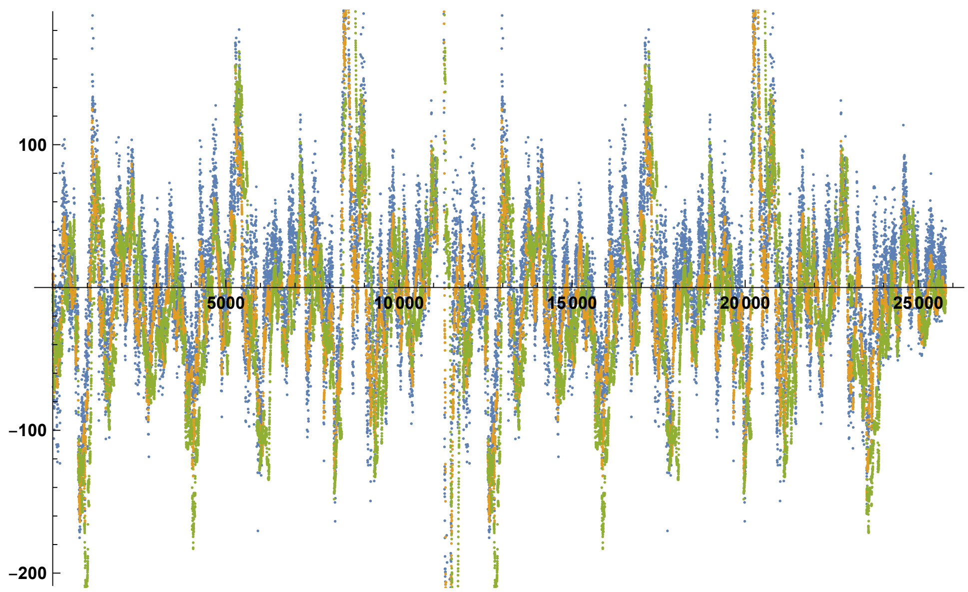
In[•]:=
```
deltaP3 =
  Table[predP2[[index]] - $PricePatternData[[720 + index, 2]], {index, 1, Length[predP3]}]
```

Out[•]=
```
{-0.683735, -5.52918, -7.9836, -10.0416, -14.2089, -16.2447, -9.36,
 -7.38688, -6.3297, -6.19108, -3.73042, -4.88817, -4.58851, -2.96713,
 -2.98615, -3.92471, -6.02153, ⋯ 25 759 ⋯, -12.4486, -11.4737, -11.3466,
 -11.4351, -11.6427, -11.2906, -11.1023, -11.1809, -11.241, -10.921,
 -10.6019, -10.6671, -11.3113, -11.426, -10.9113, -9.88373, -9.03295}
```

large output | show less | show more | show all | set size limit...

In[•]:=
```
ListPlot[{deltaP1, deltaP2, deltaP3}]
```

Out[•]=



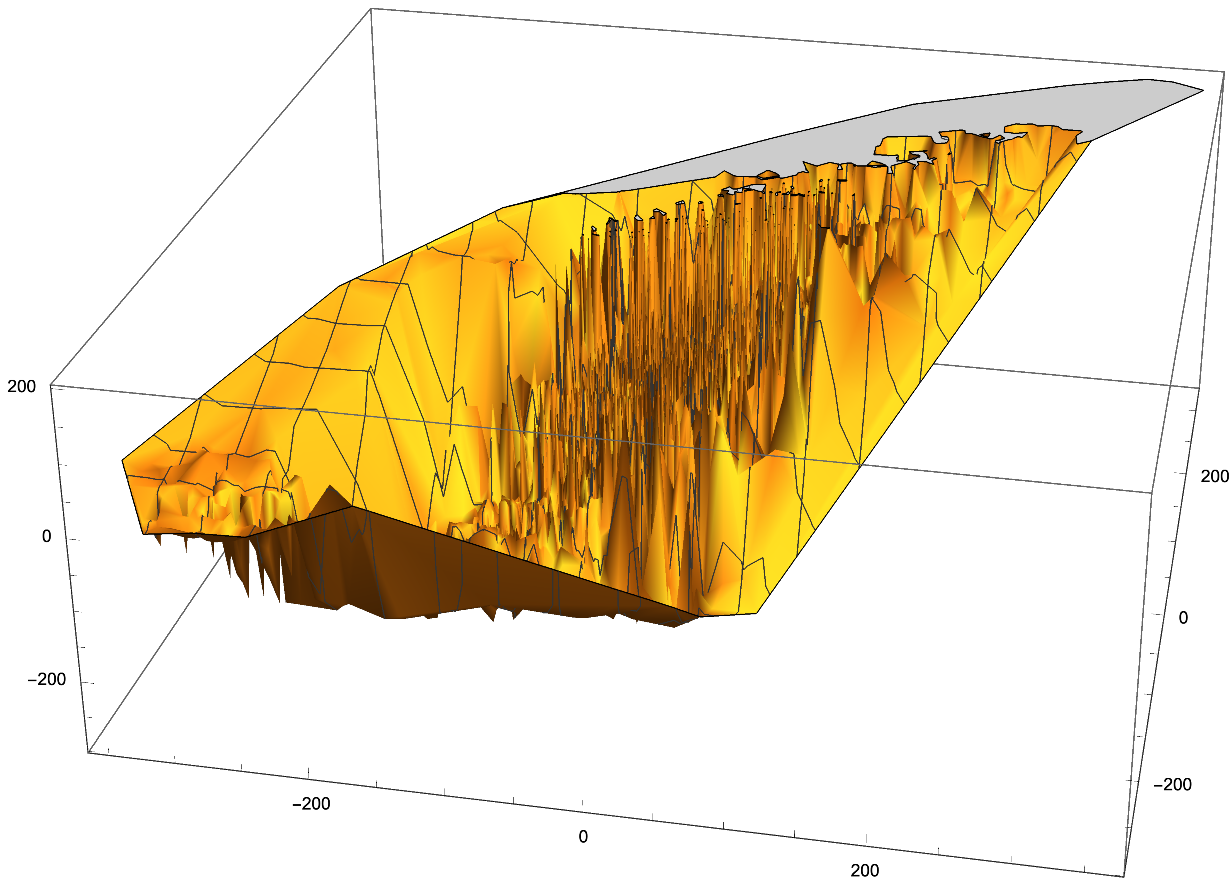
In[•]:=
```
deltaPall = Transpose[{deltaP1, deltaP2, deltaP3}]
```

Out[•]=
```
{{-28.1871, 9.53842, -0.683735}, {1.96292, 8.78072, -5.52918},
 {-10.2679, 8.41194, -7.9836}, {-39.7582, 6.15173, -10.0416},
 {45.8628, 2.73472, -14.2089}, {13.7248, 1.01112, -16.2447},
 ⋯ 25 782 ⋯, {-4.10739, -0.655324, -11.3113},
 {41.0594, -1.45163, -11.426}, {11.2632, -2.10745, -10.9113},
 {11.6908, -2.47959, -9.88373}, {14.1857, -3.3806, -9.03295}}
```

large output | show less | show more | show all | set size limit...

In[ ]:= ListPlot3D[deltaPall]

Out[ ]=



In[ ]:=

```
γ[{Vbid_, Vask_}] := (Vbid - Vask)/(Vbid + Vask)
```

In[ ]:= gammas = γ /@ $PricePatternData[[721 ;; -1, 3 ;; 4]]

Out[ ]=

```
{0.725366, 0.318722, 0.366907, 0.329451, 0.176506, -0.278668, -0.318445,
 -0.414788, -0.401086, -0.358285, 0.220797, 0.414739, 0.548472, 0.563386,
 0.618648, 0.666476, 0.591047, … 25 759 …, -0.779383, -0.682813, -0.74359,
 -0.782932, -0.775506, -0.71292, -0.665114, -0.640715, -0.682961, -0.664337,
 -0.64533, -0.711101, -0.774105, -0.724512, -0.746459, -0.73612, -0.703479}
```

large output | show less | show more | show all | set size limit...

In[∘]:=
```
priceTminus1 = $PricePatternData[[720 ;; -2, 2]]
```

Out[∘]=
```
{8811.34, 8811.67, 8814.46, 8815.27, 8817.09, 8820.9, 8822.99, 8816.64, 8815.05,
 8815.04, 8815., 8811.49, 8811.49, 8811.48, 8811.49, 8811.49, 8811.49, 8814.49,
 8822.99, 8823.8, 8823.99, 8823.99, 8826.7, 8830., 8838.2, 8852., 8858.88,
 8865.65, 8865.65, 8865.65, 8865.65, 8866.92, 8868.2, 8869.79, … 25 725 …,
 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01,
 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01,
 9019.01, 9019.01, 9017.18, 9016.1, 9016.1, 9016.1, 9016.1, 9016.09, 9016.09,
 9016.09, 9016.09, 9016.1, 9016.1, 9016.09, 9016.09, 9016.09, 9016.09, 9016.1}
```

large output | show less | show more | show all | set size limit...

In[∘]:=
```
priceT = $PricePatternData[[721 ;; -1, 2]]
```

Out[∘]=
```
{8811.67, 8814.46, 8815.27, 8817.09, 8820.9, 8822.99, 8816.64, 8815.05, 8815.04,
 8815., 8811.49, 8811.49, 8811.48, 8811.49, 8811.49, 8811.49, 8814.49, 8822.99,
 8823.8, 8823.99, 8823.99, 8826.7, 8830., 8838.2, 8852., 8858.88, 8865.65,
 8865.65, 8865.65, 8865.65, 8866.92, 8868.2, 8869.79, 8869.99, … 25 725 …,
 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01,
 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01, 9019.01,
 9019.01, 9017.18, 9016.1, 9016.1, 9016.1, 9016.1, 9016.09, 9016.09, 9016.09,
 9016.09, 9016.1, 9016.1, 9016.09, 9016.09, 9016.09, 9016.09, 9016.1, 9016.1}
```

large output | show less | show more | show all | set size limit...

In[∘]:=
```
deltaPfinal = priceT - priceTminus1
```

Out[∘]=
```
{0.334, 2.786, 0.814, 1.82, 3.806, 2.096, -6.352, -1.592, -0.014, -0.036,
 -3.512, 1.81899 × 10^-12, -0.006, 0.008, 0., 0., 3.002, 8.498, 0.808, 0.196,
 0., 2.706, 3.3, … 25 747 …, 0., 0., 0., 0., 0., 0., -1.832, -1.078, 0., 0.,
 -0.004, -0.006, 0., 0., 0., 0.006, 0.004, -0.01, 0., 0.002, -0.002, 0.01, 0.}
```

large output | show less | show more | show all | set size limit...

```
In[]:= train = Thread[Transpose[{deltaP1, deltaP2, deltaP3, gammas}] → deltaPfinal]

Out[]= {{-28.1871, 9.53842, -0.683735, 0.725366} → 0.334,
        {1.96292, 8.78072, -5.52918, 0.318722} → 2.786,
        {-10.2679, 8.41194, -7.9836, 0.366907} → 0.814,
        {-39.7582, 6.15173, -10.0416, 0.329451} → 1.82, … 25 786 …,
        {11.2632, -2.10745, -10.9113, -0.746459} → -0.002,
        {11.6908, -2.47959, -9.88373, -0.73612} → 0.01,
        {14.1857, -3.3806, -9.03295, -0.703479} → 0.}

       large output | show less | show more | show all | set size limit...

In[]:= δP = Predict[train, Method → "LinearRegression"]

Out[]= PredictorFunction[ Input type: NumericalVector (length: 4)
                          Method: LinearRegression ]
```

Test

```
In[]:= Dynamic[index]

Out[]= index

In[]:= deltaP1Test = PriceList[$PriceTestingData, p1, 180];

In[]:= deltaP2Test = PriceList[$PriceTestingData, p2, 360];

In[]:= deltaP3Test = PriceList[$PriceTestingData, p3, 720];

In[]:= Length /@ {$PriceTestingData, deltaP1Test, deltaP2Test, deltaP3Test}

Out[]= {26 511, 26 331, 26 151, 25 791}

In[]:= 26 511 - 720

Out[]= 25 791

In[]:= 26 331 - 25 791

Out[]= 540

In[]:= 26 151 - 25 791

Out[]= 360
```

In[◦]:=
```
δp1Test = $PriceTestingData⟦721 ;; -1, $PriceIndex⟧ - deltaP1Test⟦541 ;; -1⟧
```

Out[◦]=
```
{-14.5595, -38.1062, 15.1831, 26.613, -19.7064, 26.578, 31.8648,
 15.3631, 1.71009, -0.86805, 11.9301, 17.6021, 19.6861, 13.1928,
 -33.5427, -31.9699, 18.0138, 10.3629, ⋯ 25 755 ⋯, 5.96056, 23.1079,
 25.8463, 17.7197, 53.9127, 29.3717, 21.7732, 25.7362, 30.9587, 13.0376,
 23.5068, 15.969, 23.1096, 50.7115, 25.1954, 22.1403, 44.869, 47.7002}
```

large output | show less | show more | show all | set size limit...

In[◦]:=
```
δp2Test = $PriceTestingData⟦721 ;; -1, $PriceIndex⟧ - deltaP2Test⟦361 ;; -1⟧
```

Out[◦]=
```
{-2.33645, -3.06608, -3.98029, -4.1899, -4.26605, -4.51815, -5.36588,
 -4.68204, -4.28011, -4.69295, -4.50606, -4.00726, -4.18736, -5.10397,
 -5.60293, -5.01279, -5.06359, ⋯ 25 757 ⋯, -22.8491, -20.9785, -19.5088,
 -17.1816, -16.2532, -15.5136, -14.625, -13.7983, -13.2431, -13.2927,
 -12.9694, -13.124, -12.9005, -12.0756, -12.1889, -11.4195, -12.3113}
```

large output | show less | show more | show all | set size limit...

In[◦]:=
```
δp3Test = $PriceTestingData⟦721 ;; -1, $PriceIndex⟧ - deltaP3Test
```

Out[◦]=
```
{7.01814, 6.89076, 5.22294, 4.76333, 4.06558, 4.26855, 5.07335, 6.16188,
 6.10065, 5.69131, 5.80692, 5.3675, 5.70301, 5.26871, 6.5063, 4.72731,
 3.15596, 1.17579, ⋯ 25 755 ⋯, -30.7495, -21.0577, -22.8093, -23.0331,
 -18.2257, -17.1389, -7.86831, -5.73495, -6.94034, -9.27814, -10.2934,
 -12.488, -12.5144, -9.64263, -13.5119, -12.5748, -7.99635, -13.679}
```

large output | show less | show more | show all | set size limit...

In[◦]:=
```
γs = γ /@ $PriceTestingData[[721 ;; -1, 3 ;; 4]]
```

Out[◦]=
```
{-0.123044, -0.322994, -0.254758, -0.11234, -0.110771, -0.309963, -0.394384,
 -0.318886, -0.148253, 0.0127497, 0.237363, 0.178353, 0.168896, 0.315252,
 0.257211, 0.224532, 0.450943, ⋯ 25 758 ⋯, -0.366106, -0.235589, -0.203248,
 -0.199793, -0.15617, 0.0241072, 0.104638, -0.0726068, -0.0816476, -0.135407,
 -0.167947, -0.168719, -0.260714, -0.389705, -0.349147, -0.347284}
```

large output | show less | show more | show all | set size limit...

In[∘]:=
```
inputData = Transpose[{δp1Test, δp2Test, δp3Test, γs}]
```

Out[∘]=
```
{{-14.5595, -2.33645, 7.01814, -0.123044},
 {-38.1062, -3.06608, 6.89076, -0.322994},
 {15.1831, -3.98029, 5.22294, -0.254758}, {26.613, -4.1899, 4.76333, -0.11234},
 … 25 783 …, {25.1954, -12.0756, -13.5119, -0.260714},
 {22.1403, -12.1889, -12.5748, -0.389705},
 {44.869, -11.4195, -7.99635, -0.349147},
 {47.7002, -12.3113, -13.679, -0.347284}}
```

large output | show less | show more | show all | set size limit...

In[∘]:=
```
predictDeltaP = δP /@ inputData
```

Out[∘]=
```
{-0.164765, -0.492916, -0.494316, -0.232448, -0.10479, -0.635275, -0.794184,
 -0.599642, -0.221425, 0.123446, 0.546059, 0.391841, 0.374442, 0.712722,
 0.748632, 0.644594, 0.956675, … 25 757 …, -0.561925, -0.59348, -0.342023,
 -0.38892, -0.326365, -0.143709, 0.214199, 0.332388, -0.0191361, -0.0756104,
 -0.194829, -0.278005, -0.334286, -0.511111, -0.75584, -0.710758, -0.747613}
```

large output | show less | show more | show all | set size limit...

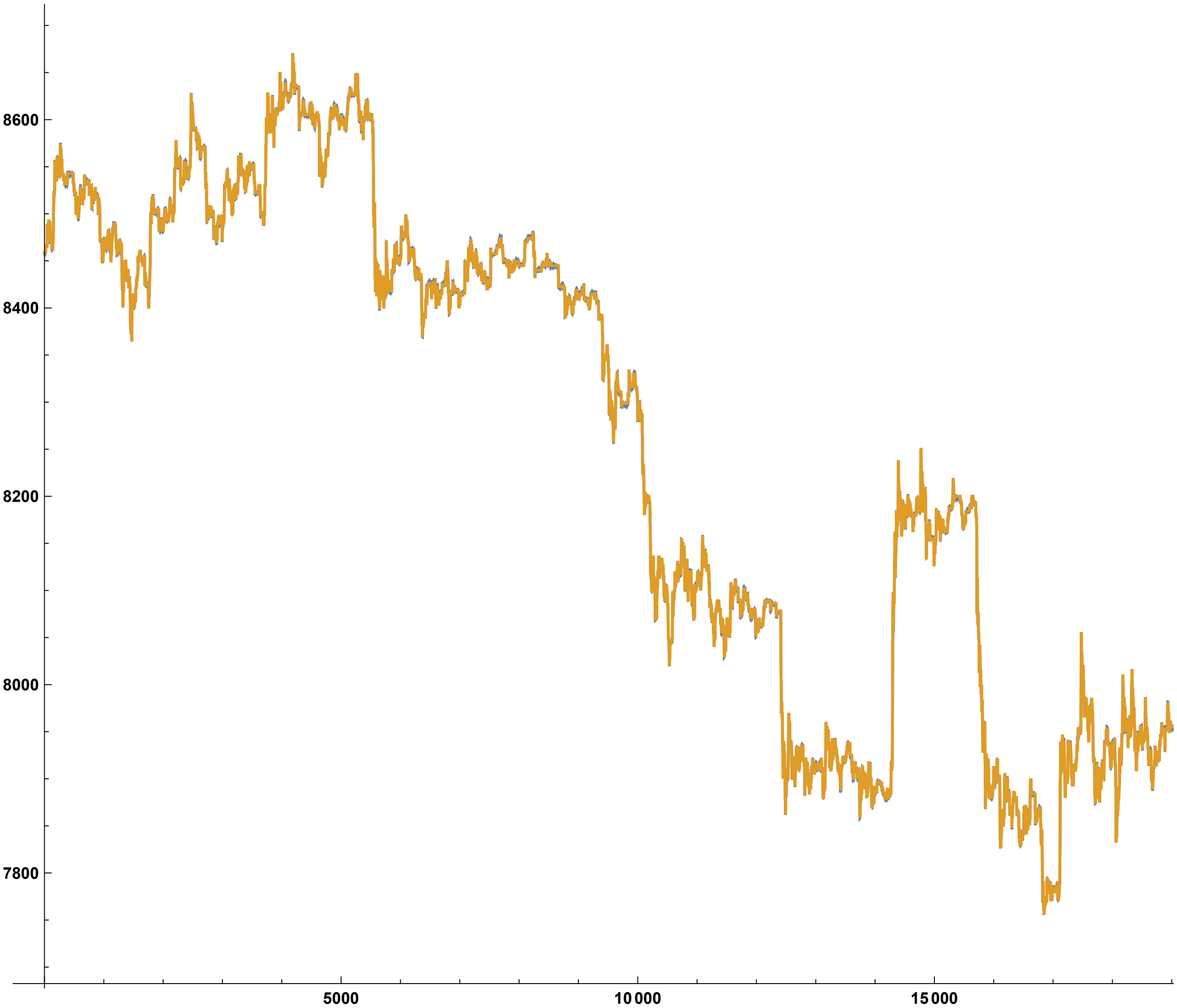
In[∘]:=
```
predictedP = $PriceTestingData⟦721 ;; -1, $PriceIndex⟧ + predictDeltaP
```

Out[∘]=
```
{8458.28, 8457.95, 8457.95, 8458.21, 8458.34, 8457.81, 8457.66,
 8457.85, 8458.23, 8458.56, 8459., 8458.84, 8458.82, 8459.16, 8459.2,
 8459.09, 8459.41, 8459.34, … 25 755 …, 7843.12, 7843.07, 7843.04,
 7843.29, 7843.24, 7843.3, 7843.48, 7843.83, 7843.96, 7843.6,
 7843.17, 7842.81, 7842.72, 7842.68, 7842.5, 7842.25, 7842.3, 7839.26}
```

large output | show less | show more | show all | set size limit...

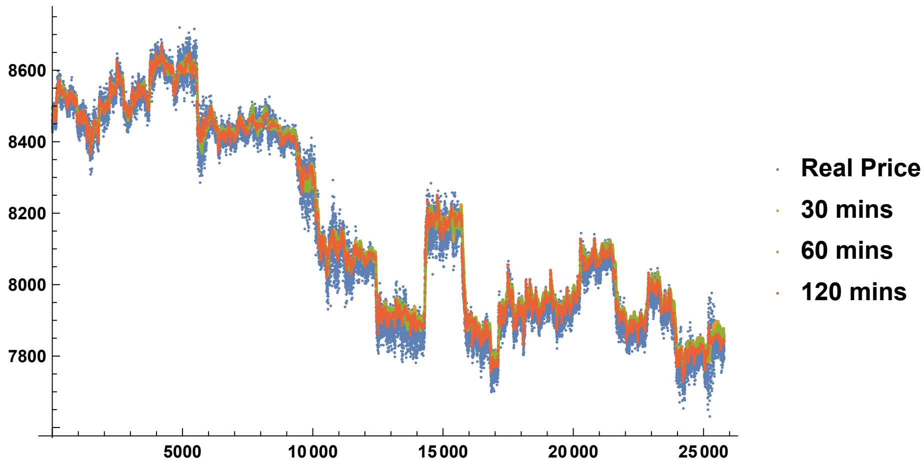
In[∘]:=
```
ListPlot[{predictedP, $PriceTestingData⟦720 ;; -1, $PriceIndex⟧}, Joined → True]
```

Out[∘]=



```
ListPlot[{deltaP1Test[[540 ;; -1]], deltaP2Test[[360 ;; -1]],
  deltaP3Test, $PriceTestingData[[720 ;; -1, $PriceIndex]]},
 PlotLegends → {"Real Price", "30 mins", "60 mins", "120 mins"}]
```

Out[ ]=



In[ ]:=

```
PriceList[data_, model_, depth_] := Table[model[data[[index - depth;; index - 1, 2 ;; 4]]],
```