Milestone 1 Report

Team members: Ninglin Huang, Yu Wang, QiuyiZhang

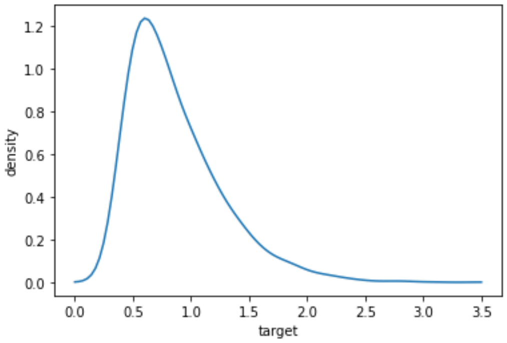
Data Exploration

1.Target

(1)Statistics for the target:

```
mean: 0.8672121212121211
std: 0.4050357160037629
range: 3.5
```

(2) Kernel density estimate of the distribution of the target values:



The KDE graph shows that values of target are mainly concentrated in the range of 0.5-1.0.

2.Features and how they relate to the target

(1) Statistics

A. mean:

```
acc_rate           -1.152606e+01
track              -1.267297e+01
m                   1.051280e+00
n                   6.038667e-02
current_pitch       6.293709e-01
current_roll        6.124848e-02
absoluate_roll     -1.100485e+01
climb_delta        -9.203636e-01
roll_rate_delta    -9.567273e-04
climb_delta_diff   -4.784242e-02
time1               2.187236e-02
time2               2.188558e-02
time3               2.188558e-02
time4               2.189782e-02
time5               2.189794e-02
time6               2.191770e-02
time7               2.191818e-02
time8               2.193745e-02
```

```
time9               2.193745e-02
time10              2.195418e-02
time11              4.390739e-02
time12              2.196885e-02
time13              2.196921e-02
time14              2.198194e-02
time1_delta        -1.320000e-04
time2_delta        -1.212121e-07
time3_delta        -6.266667e-05
time4_delta        -3.636364e-07
time5_delta        -8.072727e-05
time6_delta        -7.272727e-07
time7_delta        -1.774545e-05
time8_delta        -1.212121e-07
time9_delta        -9.890909e-05
time10_delta        4.848485e-07
time11_delta        8.999901e+00
time12_delta       -7.272727e-07
time13_delta       -9.309091e-05
time14_delta       -1.000000e+01
omega              -5.102791e-01
set                 2.198218e-02
```

B. std:

```
acc_rate            259.637258
track                25.675733
m                     0.320703
n                     0.118805
current_pitch         0.313628
current_roll          0.967274
absoluate_roll        4.140399
climb_delta          10.334136
roll_rate_delta       0.013203
climb_delta_diff      1.132179
time1                 0.006873
time2                 0.006906
time3                 0.006906
time4                 0.006914
time5                 0.006914
time6                 0.006924
time7                 0.006925
time8                 0.006933
time9                 0.006933
time10                0.006939
time11                0.013876
time12                0.006953
time13                0.006954
time14                0.006960
time1_delta           0.000695
time2_delta           0.000011
time3_delta           0.000462
time4_delta           0.000025
```

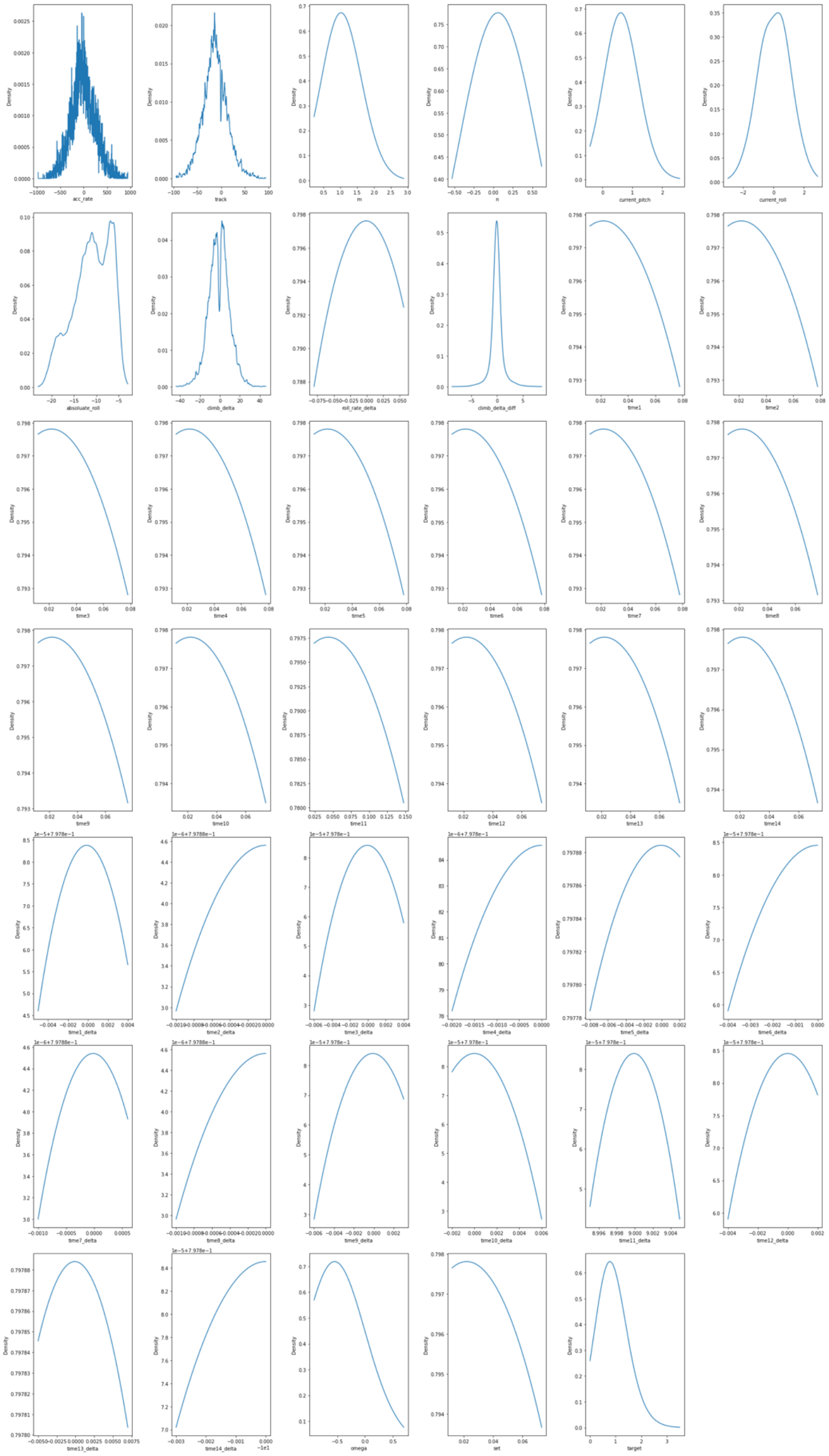
```
time5_delta           0.000491
time6_delta           0.000049
time7_delta           0.000113
time8_delta           0.000011
time9_delta           0.000615
time10_delta          0.000070
time11_delta          0.000631
time12_delta          0.000060
time13_delta          0.000613
time14_delta          0.000037
omega                 0.257113
set                   0.006961
```

C. range:

```
acc_rate            1921.000000
track                189.000000
m                      2.669835
n                      1.150000
current_pitch          3.000000
current_roll           5.900000
absoluate_roll        20.000000
climb_delta           90.000000
roll_rate_delta        0.136000
climb_delta_diff      17.500000
time1                  0.066000
time2                  0.066000
time3                  0.066000
time4                  0.066000
time5                  0.066000
time6                  0.066000
time7                  0.066000
time8                  0.064000
time9                  0.064000
time10                 0.062000
time11                 0.124000
time12                 0.062000
time13                 0.062000
time14                 0.061000
time1_delta            0.009000
time2_delta            0.001000
time3_delta            0.010000
time4_delta            0.002000
time5_delta            0.010000
time6_delta            0.004000
time7_delta            0.001600
time8_delta            0.001000
time9_delta            0.009000
time10_delta           0.008000
time11_delta           0.010000
time12_delta           0.006000
time13_delta           0.012000
time14_delta           0.003000
```

```
omega                          1.609438
set                            0.061000
```

(2) Kernel density estimate of the distribution for each feature:



(3) Correlation between each feature and the target:

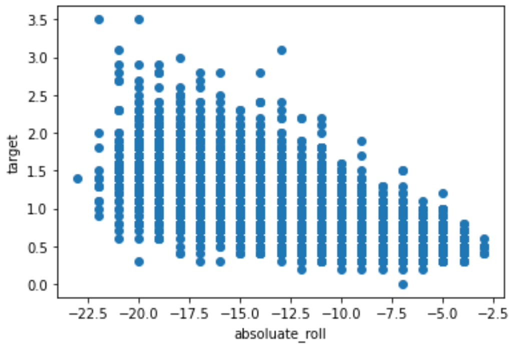
```
acc_rate             0.079642
track               -0.061289
m                    0.334420
n                    0.396974
current_pitch        0.308080
current_roll         0.106015
absoluate_roll      -0.704515
climb_delta         -0.056970
roll_rate_delta     -0.052774
climb_delta_diff    -0.073560
time1                0.641312
time2                0.640405
time3                0.640405
time4                0.640398
time5                0.640413
time6                0.638817
time7                0.638829
time8                0.636374
time9                0.636374
time10               0.633567
time11               0.633707
time12               0.630812
time13               0.630761
time14               0.628335
time1_delta          0.035307
time2_delta         -0.009047
time3_delta          0.016617
time4_delta         -0.019433
time5_delta          0.010641
time6_delta         -0.019433
time7_delta          0.032115
time8_delta         -0.017202
time9_delta          0.040202
time10_delta        -0.002875
time11_delta         0.027330
time12_delta        -0.008918
time13_delta         0.025826
time14_delta        -0.024296
omega                0.615920
set                  0.628272
target               1.000000
```

The 3 most correlated features:

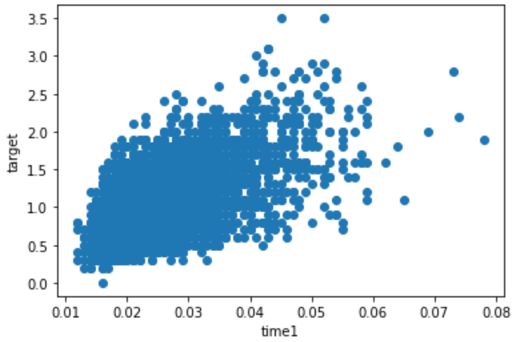
```
absoluate_roll    -0.704515
time1             0.641312
time5             0.640413
```

So the top3 features that correlated with target are: absoluate_roll, time1, time5.

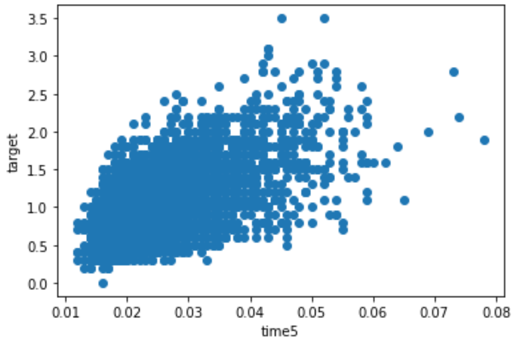
(4) Scatter plot for each of the 3 features found above between the feature and the target:



So absoluate_roll is negatively correlated with target.



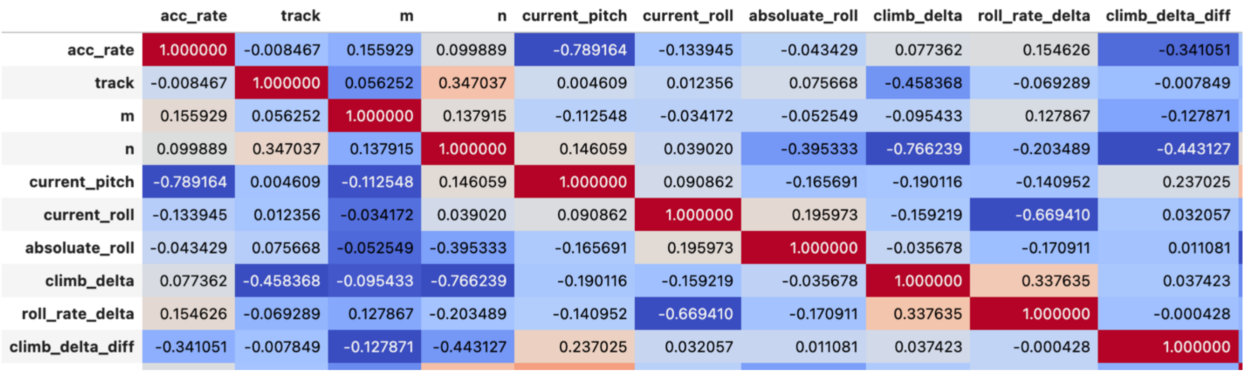
Time1 is positively correlated with target.



Time 5 is positively correlated with target.

3.Relationships between features

(1) Correlation matrix for the features:



	acc_rate	track	m	n	current_pitch	current_roll	absoluate_roll	climb_delta	roll_rate_delta	climb_delta_diff
acc_rate	1.000000	-0.008467	0.155929	0.099889	-0.789164	-0.133945	-0.043429	0.077362	0.154626	-0.341051
track	-0.008467	1.000000	0.056252	0.347037	0.004609	0.012356	0.075668	-0.458368	-0.069289	-0.007849
m	0.155929	0.056252	1.000000	0.137915	-0.112548	-0.034172	-0.052549	-0.095433	0.127867	-0.127871
n	0.099889	0.347037	0.137915	1.000000	0.146059	0.039020	-0.395333	-0.766239	-0.203489	-0.443127
current_pitch	-0.789164	0.004609	-0.112548	0.146059	1.000000	0.090862	-0.165691	-0.190116	-0.140952	0.237025
current_roll	-0.133945	0.012356	-0.034172	0.039020	0.090862	1.000000	0.195973	-0.159219	-0.669410	0.032057
absoluate_roll	-0.043429	0.075668	-0.052549	-0.395333	-0.165691	0.195973	1.000000	-0.035678	-0.170911	0.011081
climb_delta	0.077362	-0.458368	-0.095433	-0.766239	-0.190116	-0.159219	-0.035678	1.000000	0.337635	0.037423
roll_rate_delta	0.154626	-0.069289	0.127867	-0.203489	-0.140952	-0.669410	-0.170911	0.337635	1.000000	-0.000428
climb_delta_diff	-0.341051	-0.007849	-0.127871	-0.443127	0.237025	0.032057	0.011081	0.037423	-0.000428	1.000000

This is part of the heatmap of correlation matrix. The whole matrix is too large to show in this report. But you can see it in our Jupyter notebook.

(2) Interpret the matrix to see which features are highly correlated to each other:

The heat map shows the correlation coefficient between each pair of features. The color will indicate the strength and direction of the correlation (positive correlation will be shown as a red shade, negative correlation as a blue shade, and no correlation as white). The value in each cell of the heatmap represents the correlation coefficient between two corresponding features.

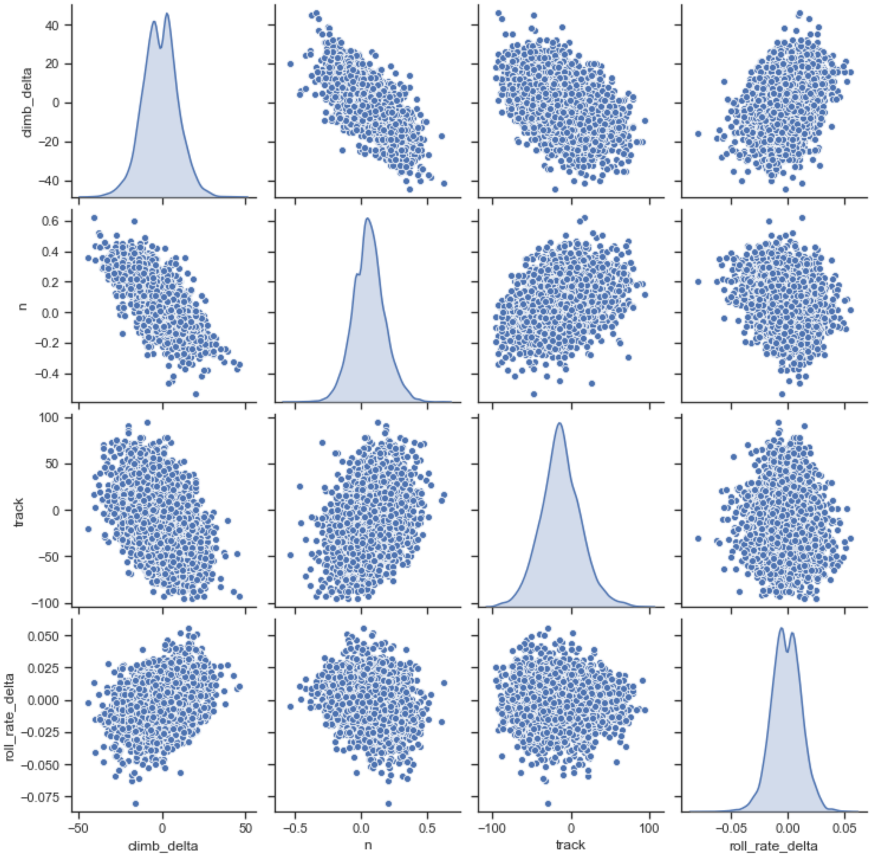
So from the heatmap we can get that there is a strong negative correlation between acc_rate and current_pitch, a strong negative correlation between track and climb_delta, a strong negative correlation between n and climb_delta, a strong negative correlation between current_roll and roll_rate_delta, a strong positive correlation between time1 and current_pitch, etc.

(3) Select 1 feature and take the 3 features that are most correlated to your chosen feature

A. Select ‘climb_delta’, and the top3 most correlated features are:

```
n                  - 0.766239
track              - 0.458368
roll_rate_delta    0.337635
```

B. Plot a 4x4 matrix of scatter plots:



The off-diagonal scatter plots show the relationship between each pair of features, while the diagonal plots show kernel density estimates for each corresponding feature. From the scatter plot, we can find that the climb_delta has a negative correlation with both n and track and has a positive correlation with roll_rate_delta. Besides, track has a positive correlation with n. And the other scatter plots don’t show clear patterns, so it suggests little or no correlation between these two features.

Baseline Models

1.Linear regression, kNN & random forest

At first, we pick linear regression, kNN and random forest as the model.

We implemented the 10-fold cross-validation to compute errors of the in-sample and out-of-sample. Since the problem is a regression problem, we choose to use mean squared error as loss function,which is the most common loss function.

(1)In-sample mean error：

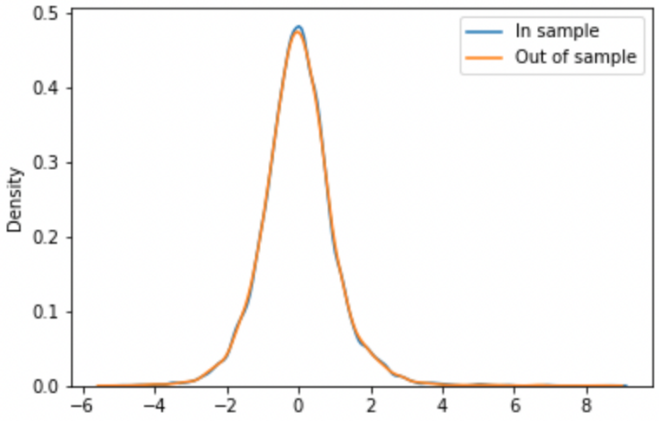
Linear regression 0.029564204643284676
knn 0.08284349629629773
rf 0.00037806420202020336

Given by the in sample error, all of these three model perform well and have a very sample mean square error. And comparing them to each other we could also find that the random forest have the smallest mean square error.

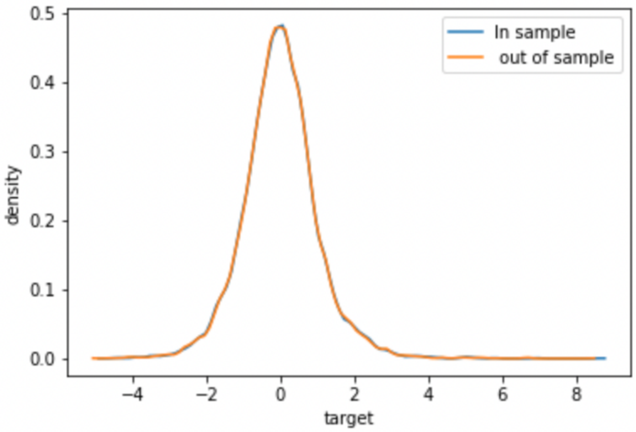
(2)Out-of-sample mean error：
Linear regression 0.03017786706300265
knn 0.12735898181818178
rf 0.0024899340606060593

Although the out of sample mean error is a little bit higher than the in sample mean error, it still have small sample mean error here.

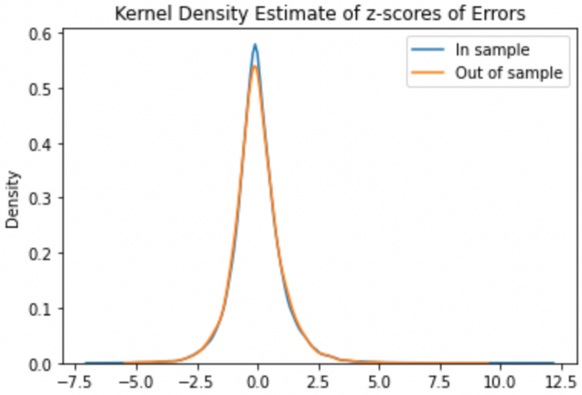
(3)kernel density estimation of z-score
For each model, kernel density estimation is performed on the z-score of each data point error as follows:
Linear model:



Knn:



Random Forest:



There is no big difference for in sample and out of sample error for the same model. And even though they all locate around 0, but we could still find the difference between this three model when we comparing the peak and the range of x axis. And we can find that the random forest model have the higher peak and other two model, which means it have a higher z score of error when comparing to two other model.

(4)In-sample t-test：

In-sample paired t-test p-values:

('LR', 'knn'): 1.0735217007824255e-18,

('LR', 'rf'): 3.6949448625998345e-14,

('knn', 'rf'): 6.6786815302998e-18,

Since the p value given by the in sample t test is pretty smaller, so that we could reject the null hypothesis that models are perform similarity and they are totally different from each other.

(5)Out-of-sample t-test：

```
Out-sample paired t-test p-values:
('LR', 'knn'): 1.7316322272982124e-11,
('LR', 'rf'): 3.864940749048801e-06,
('knn', 'rf'): 3.195124916575802e-10
```

Although the p-value increase a little bit compared to the case in sample t test, the p-value is still obviously smaller than the 0.05, so we could derived the same conclusion with the in sample t test.

2.Gaussian process, neural network & gradient boosting regression

Then we chose the other three models which are Gaussian process, neural network, and gradient boosting regression. Data were fitted, and in-sample and out-of-sample errors were calculated using 10-fold cross-validation. MSE is the loss function used in this case. It is a common assessment metric in regression problems. MSE calculates the mean squared error between predicted and actual values, allowing us to compare the performance of various models.

(1)In-sample mean error：

```
GPR 9.119034839800694e-21
MLP 0.045053909835082534
GBR 0.021287710818668808
```

The GPR model's in-sample error is close to zero, indicating that it fits the training data very well. However, the possibility of overfitting must be considered. MLP and GBR have higher errors, but they are still low. GPR outperforms the other two models on the training set.

(2)Out-of-sample mean error：

```
GPR 0.9115925878519997
```

MLP 0.04740220244348373

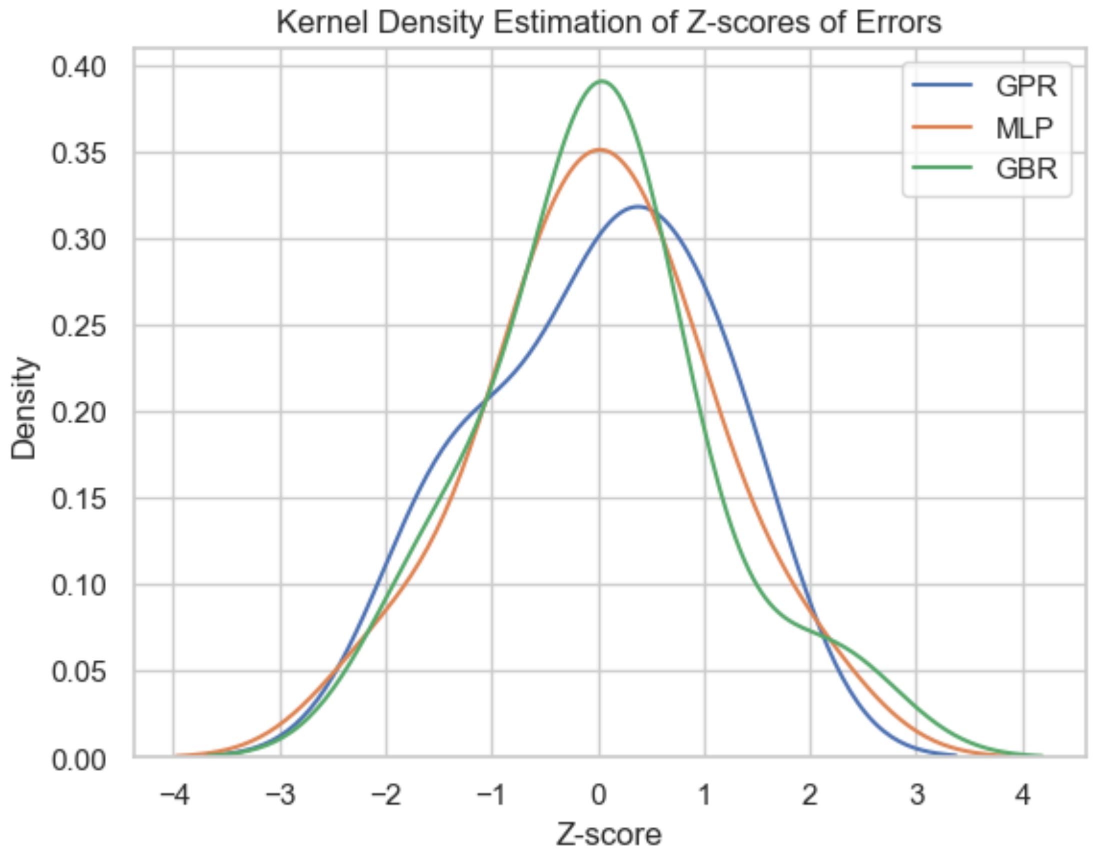
GBR 0.026084165983036194

GPR has a relatively high out-of-sample error, which may indicate that it does not generalize as well as MLP and GBR on the test set. Out-of-sample errors are lower in MLP and GBR, with GBR slightly outperforming MLP.

In summary, GPR suffers from over-fitting, which causes it to perform well in the sample but have a large out-of-sample error. MLP and GBR are better at generalization. The GBR model performs better when the errors inside and outside the sample are combined.

(3)kernel density estimation of z-score

For each model, kernel density estimation is performed on the z-score of each data point error as follows:



Explanation:

The highest peak in the distribution of kernel density estimates for GBR indicates that prediction errors are more concentrated within a certain range. This means that the GBR model's prediction error is more stable and consistent, which may result in better generalization performance.
The kernel density estimate distribution for GPR has the lowest peak and is skewed to the right. This indicates that the GPR model's forecast error distribution is more dispersed, and the overall error may be higher. As a result, the GPR model's generalization ability may be limited.
MLP kernel density estimates have a distribution that falls between GBR and GPR, with a peak in the middle. This indicates that the MLP model's prediction error distribution is between GBR and GPR, and its generalization performance may also be between the two.

(4)In-sample t-test：

```
In-sample paired t-test p-values:
('GPR', 'MLP'): 1.3223754506560932e-12,
('GPR', 'GBR'): 2.2834047383938924e-19,
('MLP', 'GBR'): 3.801393226797644e-10
```

In the sample t-test, the p-values of the three models are far less than 0.05, and there are significant differences. GPR outperforms MLP and GBR on the training set when combined with previous error data.

(5)Out-of-sample t-test：

```
Out-sample paired t-test p-values:
('GPR', 'MLP'): 3.810654196815402e-15,
('GPR', 'GBR'): 5.3241995567333776e-15,
('MLP', 'GBR'): 2.453692250225471e-06
```

The three models also differed significantly in the out-of-sample t-test.
GBR is a good choice when considering the comprehensive model's accuracy and generalization ability.