Maximum Temperature Prediction

Author name: Yihang Zhou

email contact: zhouyiha@usc.edu

Date: 12/01/2020

1. Abstract

In this project, the goal is to predict maximum Temperature in next day. The data is from the Korea Meteorological Administration over Seoul, South Korea which is real-world. By splitting the data in four part: training data, validation data, testing data and unknown data, I do some preprocess to change category feature, handle missing feature and unknow data. Then I use 4 models to train and apply them to validation set or cross validation set. The results of MSE is: Lasso(1.9287), Ridge(1.9412), Random forest method 1(0.9833), Random Forest method 2(0.9652) and Adaboost(1.9107). Comparing the result of these four models, I choose to Random Forest method 2 as my final model and apply it to test set, the MSE result is 1.0860.

2. Introduction

2.1. Problem Type, Statement and Goals

This is a regression problem. In this project, I am trying to predict the maximum temperature in next day. The goal is to find the best model which has minimum MSE. Also, I want to get find the difference if I reduce some features. This project is not trivial by following reasons:

(1) Category feature
There is a category feature in the data set. I use three way to handle this category and see difference of each performance. Choose the best one to do following model training.

(2) High dimensionality of feature space
There are 23 features in this data set. If I use all the features, in some model, the computation cost will be high. Hence, reducing number of features is a

good way to lessen computation cost and may get better result by deleting some bad features.

(3) Nonlinear behaviors
Because of high dimension of the data set, I will use some non-linear model to train it and see the performance difference with linear model.

(4) The unknown data and miss features
The are some unknow data in the data set. I want to label them and comparing with just delete them. There are also some miss feature data I need to handle.

2.2. Overview of my Approach

Lasso, Ridge, Random Forest and Adaboost are used in this project. For Lasso and Ridge, I choose different parameters and calculate MSE to compare their performance. For Random Forest and Adaboost, by choosing number of trees and learning rate to calculate MSE and R2 score to compare their performance.

In addition, I use Random Forest to get a dimension reduction to see different number of features influence to the result.

3. Implementation

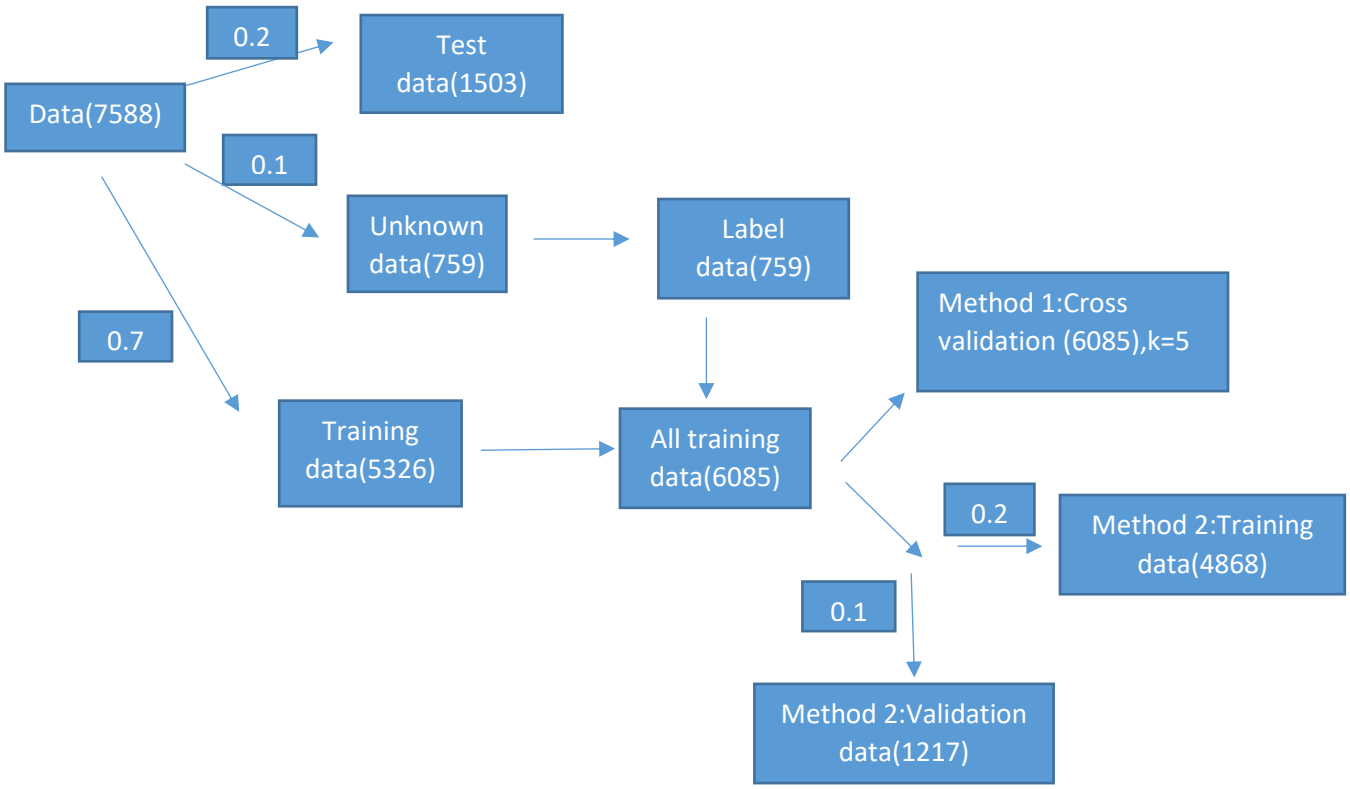
3.1. Data Set

There are 23 features which 22 features are numeric features and 1 is category feature. The label is Next_Tmax(next day maximum temperature). In the feature, the LDAPS is a model which get statistics by the Korea Meteorological Administration over Seoul, South Korea.

Feature Name	Type	description
1. station	Number	used weather station number: 1 to 25
2. Date	Category	Present day: yyyy-mm-dd ('2013-06-30' to '2017-08-30')
3. Present_Tmax	Number	Maximum air temperature between 0 and 21 h on the present day (Â°C): 20 to 37.6
4. Present_Tmin	Number	Minimum air temperature between 0 and 21 h on the present day (Â°C): 11.3 to 29.9
5. LDAPS_RHmin	Number	LDAPS model forecast of next-day minimum relative humidity (%): 19.8 to 98.5
6.	Number	LDAPS model forecast of next-day maximum relative humidity (%): 58.9 to 100

7. LDAPS_Tmax_lapse	Number	LDAPS model forecast of next-day maximum air temperature applied lapse rate (Â°C): 17.6 to 38.5
8. LDAPS_Tmin_lapse	Number	LDAPS model forecast of next-day minimum air temperature applied lapse rate (Â°C): 14.3 to 29.6
9. LDAPS_WS	Number	LDAPS model forecast of next-day average wind speed (m/s): 2.9 to 21.9
10. LDAPS_LH	Number	LDAPS model forecast of next-day average latent heat flux (W/m2): -13.6 to 213.4
11. LDAPS_CC1	Number	LDAPS model forecast of next-day 1st 6-hour split average cloud cover (0-5 h) (%): 0 to 0.97
12. LDAPS_CC2	Number	LDAPS model forecast of next-day 2nd 6-hour split average cloud cover (6-11 h) (%): 0 to 0.97
13. LDAPS_CC3	Number	LDAPS model forecast of next-day 3rd 6-hour split average cloud cover (12-17 h) (%): 0 to 0.98
14. LDAPS_	Number	CC4 - LDAPS model forecast of next-day 4th 6-hour split average cloud cover (18-23 h) (%): 0 to 0.97
15. LDAPS_PPT1	Number	LDAPS model forecast of next-day 1st 6-hour split average precipitation (0-5 h) (%): 0 to 23.7
16. LDAPS_PPT2	Number	LDAPS model forecast of next-day 2nd 6-hour split average precipitation (6-11 h) (%): 0 to 21.6
17. LDAPS_PPT3	Number	LDAPS model forecast of next-day 3rd 6-hour split average precipitation (12-17 h) (%): 0 to 15.8
18. LDAPS_PPT4	Number	LDAPS model forecast of next-day 4th 6-hour split average precipitation (18-23 h) (%): 0 to 16.7
19. lat	Number	Latitude (Â°): 37.456 to 37.645
20. lon	Number	Longitude (Â°): 126.826 to 127.135
21. DEM	Number	Elevation (m): 12.4 to 212.3
22. Slope	Number	Slope (Â°): 0.1 to 5.2
23. Solar radiation	Number	Daily incoming solar radiation (wh/m2): 4329.5 to 5992.9
Label		
24. Next_Tmax	Number	The next-day maximum air temperature (Â°C): 17.4 to 38.9

3.2. Dataset Methodology



In this process, I have 7752 data. After preprocess, I have 7588 data used for train. In different model, I use validation set or cross validation to calculate MSE or R2 score.

3.2.1 Cross Validation

In Lasso regression and Ridge regression, I use k-fold which k is 5 to do cross validation by training data(as method 1 show above). For every parameter, it will give me 5 results of its MSE. By calculating the mean of MSE and pick the minimum, I can get the best parameter in these two models.

3.2.2 Validation

In Random Forest regression and Adaboost regression, I split all training data to training data and validation data(as method 2 show above). Then I use training data to train the model with different number of trees or learning rate. I use validation data to make prediction and calculate MSE and R2 score to choose the best parameter.

In Random Forest, I also try to delete different number of features to see the performance by using validation set to calculate MSE and R2 score.

3.2.3 Test set

After I choose the final model, the test set is used to calculate the performance of my final model. It will just be used by once.

3.3. Preprocessing, Feature Extraction, Dimensionality Adjustment

3.3.1 Missing feature:

There are some missing features which are in same data which are not. I calculate the number of data which has missing feature which is 164 which is 2% of the total data set, so I decide just to delete them.

3.3.2 Category transform

There is one category in the data set which is date. I used 3 ways to change the category and use linear model to see each performance by using validation set.

Method 1: Delete category feature:

Consider that this feature may has no useful information, so I try to delete it.

Method 2: Use one-hot code:

This is a common way to transfer category to uncorrelated features. However, each date is different, if I transfer all the date, then the feature dimension would increase heavily, so consider the date is from June 30 to August 30 in 5 years and each day has 25 stations, I decide to change them in 61 categories which means each category present 25 stations for five year. This can help it focus on date changing influence in summer. After one-hot encoding for the category, data dimension will increase 61, I use Truncated SVD to reduce the new features to 3 and put it in training data.

Code using:

One-hot encoding: OneHotEncoder()

Dimension reduction: TruncatedSVD()

Use 1:61 to transform:

The reason I want to use 1:61 to substitute 61 categories is I think date has correlation with each other. For example, today maximum temperature may close with next day maximum temperature.

Result Showing

I use linear model to train the data after using different method to transform category feature. The results are below:

Method	MSE for validation set
Delete category (without standardize)	2.108
Delete category (standardize)	2.108
One-hot encoding(standardize)	2.127
Use 1:61 to transform(standardize)	2.094

As we can see, the minimum MSE is the last method. Hence, I decide use 1:61 to transform the category. Also, for data standardize, whether I standardize data, the result is almost the same. I think the reason is the magnitude of each feature does not have huge gap. Consider following non-linear training model, I decide use standardize data.

3.3.3 unknow data transform

There are unlabeled data in data set which is around 10% of the whole data set. I used SVM to predict unlabeled data and label them. Then put them in training data and calculating MSE by validation data in linear model.(the validation data is split before predict unknown data)

Method	MSE for validation set
add unknown data	2.093

We can see that the MSE decease a little bit, so I decide to add labeled unknown data in the training dada.

3.4. Training Process

3.4.1 Lasso regression

Model selection reasons and algorithms

Lasso regression is kind of linear regression which calculate squared error. The different is it adds a regularization term(l_1 norm) to give a penalty when choosing weight. It can help avoid overfitting because by this way it can decrease model complexity. It is also a good choice to do feature selection because it provides a sparer solution to weight(some features weight can shrink to 0). Hence, I choose this model to train data.

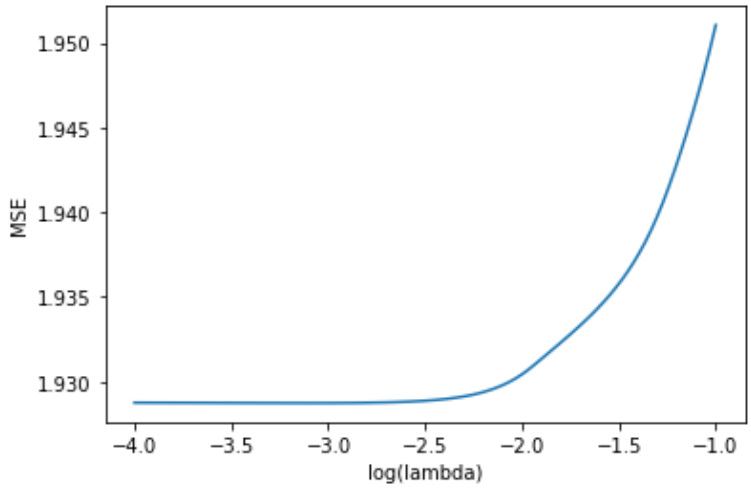
$$W^* = argmin(||y - Xw||_2^2 + \lambda||\mathrm{w}||_1)$$

(It is an object function, method to get weight)

Parameter selection

I use cross validation in this model. In every parameter, I use k-fold which k is 5 to split training data. Hence, each time it will use 4 parts data to train and 1 part data as validation set to calculate MSE. In this way, I can get 5 MSE in each parameter. Calculate mean of these 5 MSE and pick the minimum number in different parameter.

Lambda range (quantity:100)	Best lambda	MSE for cross validation set
10^(-4) to 10^(-1)	0.0008	1.9287



Analyze

In this model, I choose 100 different lambdas, so the number of hypothesis set is 100. From the best model, we can see that feature 6 and 19 have 0 coefficient. The dimension reduces 2 comparing with pre-processed feature space. The cross validation is used here to avoid overfitting. Choosing enough number of lambdas to avoid underfitting.

3.4.2 Ridge regression

Model selection reasons and algorithms

Different with Lasso regression, Ridge regression use l_2 norm in regularization term to find the best weight. It does not have the function to reduce feature dimension, but I can also give me a good analyze of data by weight constrain.

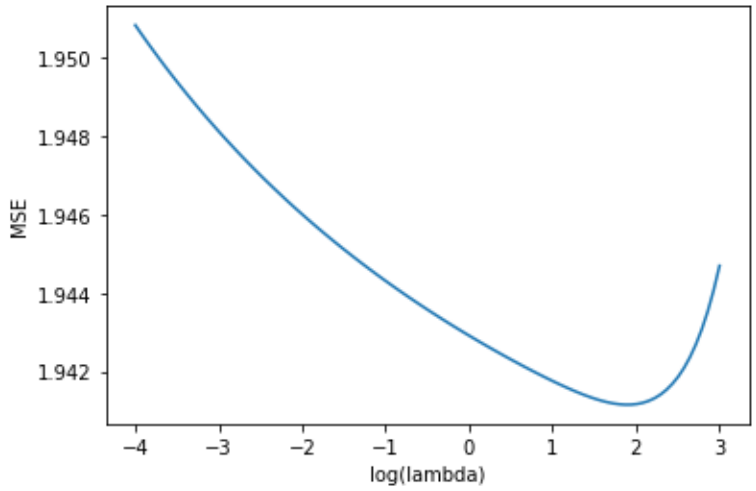
$$W^* = argmin(||y - Xw||_2^2 + \lambda||\mathrm{w}||_2^2)$$

(It is an object function, method to get weight)

Parameter selection

Same with Lasso regression model, I use cross validation here. I calculate mean of 5 MSE in each parameter and choose the minimum.

Lambda range (quantity:100)	Best lambda	MSE for cross validation set
10^(-4) to 1000	73.90	1.9412



Analyze

In this model, I choose 100 different lambdas, so the number of hypothesis set is 100. There is no dimension reduction here. Each feature has it own weight. By using cross validation and choose appropriate lambda here can reduce overfitting and underfitting.

3.4.3 Random Forest Regression

Method 1 : Use all features to train

Model selection reasons and algorithms

In linear regression model, I find it does not work so well. Hence, I want to try some non-linear regression model which is Random Forest to see its performance. This model is also called "bagging". In every tree, it selects parts of data and parts of features randomly to train. The final result in regression problem is average of all trees' result.

$$\hat{f}(x) = \frac{1}{B}\sum_{b}^{B} \hat{f}_b\,(\underline{x})$$

B is the total number of trees; $\widehat{f_b}(x)$ is prediction of each tree.

Parameter selection

I use validation in this model. The reason I do not use cross validation is Random Forest is kind of bootstrap aggregating problem. In each tree it does not use all data, so other data which does not use to train can be used to as validation set and evaluate each features importance. Hence, using cross validation is not necessary in Random Forest. From following, we can see that with in different number of trees, we calculate its MSE and choose the minimum one.

Number of trees range(150)	Best number of trees	R2-score	MSE for cross validation set
10 to 300	204	0.8954	0.9833

Analyze

I use 150 different trees, so the number of hypothesis sets are 150. There is no dimension reduction here. Each feature has its own importance. By choosing an appropriate number of trees and max depth of each tree can avoid overfitting and underfitting here.

Model 2: Reduce features to train:

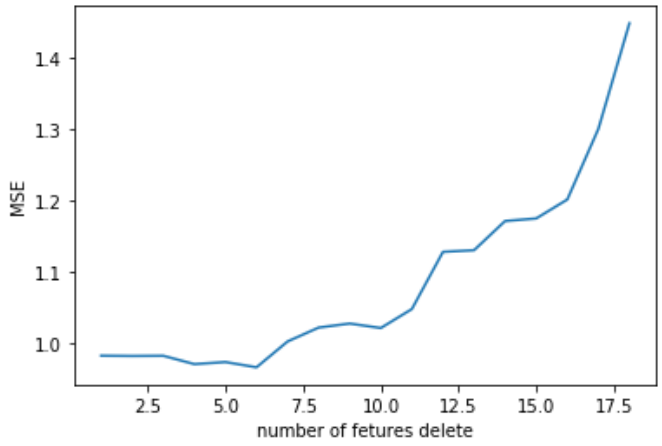
Model selection reasons and algorithms

In above random forest, I find it performs well in validation set. As we know Random Forest can show the importance of each feature, so I rank them and delete them from less important feature to most important feature. I want to see what the performance will be in different number of features. Following graph is the most 7 importance features.

Features name	rank	importance
7 LDAPS_Tmax_lapse	1	0.7007
3 Present_Tmax	2	0.0560
11 LDAPS_CC1	3	0.0290
13 LDAPS_CC3 - LDAPS	4	0.0238
9 LDAPS_WS - LDAPS	5	0.0216
14 LDAPS_CC4 - LDAPS	6	0.0205
8 LDAPS_Tmin_lapse	7	0.0172

Parameter selection

Same with Method I, the difference is every time I delete 1 feature, I will try 150 different trees to train and calculate MSE by validation set and choose the best tree. I delete feature 19 times. Following picture is minimum MSE for each deletion.



Trees range(150)	Number of features delete	Best number of tress	R2-score	MSE(validation set)
10 to 300	6	89	0.8958	0.9652

Analyze

I use 150 different trees and 19 deletion times, so the number of hypothesis sets are 2850 for validation with minimum MSE. . There is dimension reduction here. I reduce features 19 times. By reducing less importance features, choosing an appropriate number of trees and max depth of each tree can avoid overfitting and underfitting here.

3.4.4 Adaboost

Model selection reasons and algorithms

Adaboost is an adaptive basis function model. Different with random forest. It can classify the whole feature space. It is a "weak learner" that only need do better than chance. As it is also a non-linear model, I want to use it to see the performance difference between random forest and other linear regression model.

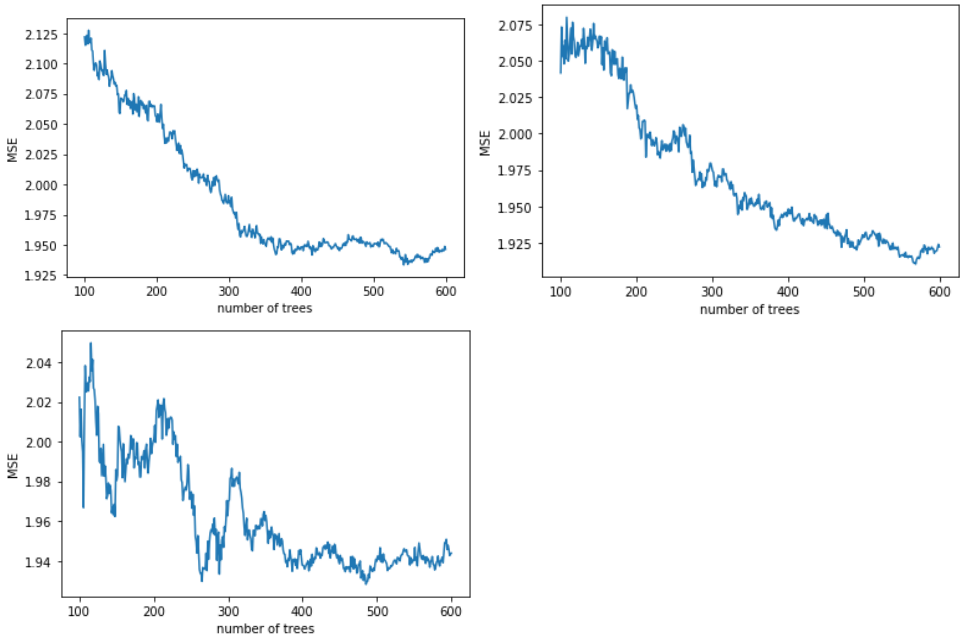
$$\hat{f}(\underline{x}) = f_0 + \sum_{m=1}^{M} \beta_m \emptyset(\underline{x}; \gamma_m)$$

$\emptyset(\underline{x};\ \gamma_m)$ is m_{th} base classifier; γ_m is parameters which are learn from data; β_m is importance of m_{th} base classifier; f_0 is initial value.

Parameter selection

I use validation set here to select parameter. The reason I do not use cross validation is because of high time computation. Adaboost need a long time to train the model. There are two parameters need to adjust, one is learning rate, one is number of estimators. There exists a balance between learning rate and

number of estimators. If learning rate is small, it needs more estimators to find the best model and will cost more time. If learning rate is too large, the number of estimators can choose small, so the model will be trained faster. However, it may hard to find global minimum because of the large learning rate. Following graph is what I got using three different learning rates in 450 different trees.



Estimator range(450)	Learning rate	Best number of tress	R2-score	MSE
100 to 600	3.4	542	0.7941	1.9334
100 to 600	3.5	568	0.7967	1.9104
100 to 600	3.6	147	0.7883	1.9282

Analyze

I use 450 different trees and 3 different learning rates, so the number of hypothesis sets are 1350 for validation with minimum MSE. There is no dimension reduction here. By choosing an appropriate number of estimators and learning rate can avoid overfitting and underfitting here.

3.5. Model Selection and Comparison of Results

From above model, I get following statistic:

	Best parameter		MSE	
Lasso	Lambda: 0.0008		cross validation: 1.9287	
Ridge	Lambda:73.90		cross validation: 1.9412	
Random Forest method 1	Number of trees: 204		R2-score: 0.8954	Validation: 0.9833
Random Forest method 2	Number of features delete: 6	Number of trees: 89	R2-score: 0.8958	Validation: 0.9652
Adaboost	Learning rate: 3.5	568	R2-score: 0.7967	Validation: 1.9104

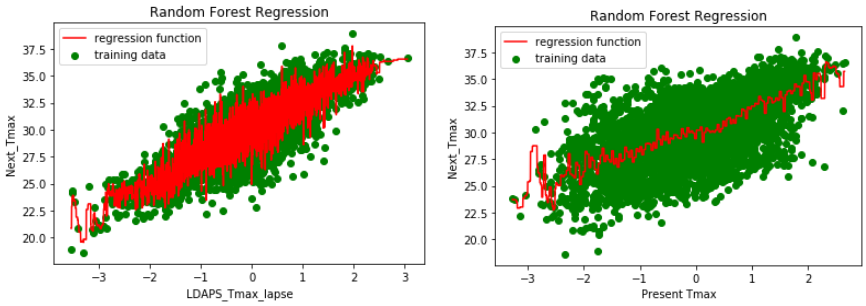
We can see that Random Forest method 2 has least MSE and R2-score. The reason I use R2-score here is I want to have another justification if MSE is similar.

For Lasso and Ridge, I thought Lasso and ridge can have much better performance than simple linear regression. However, the performance is just a little bit better than it. I think the reason is the data distribution not wide and do not change rapidly, so my object function in Lasso do not need large lambda to constrain the weight which can help reduce function degree. In ridge, the lambda is large because of one feature is highly relevant with output, so just give that feature enough weight, it can get a not bad MSE result.

The Random Forest perform well in this data set. This is what I expected as it a strong model which most time perform well than others. What I do not expected is method 2, compare with method 1, have a better result when some features are deleted. I think the reason is some features is bad which can mislead model weight choosing. Then when apply in validation set, the perform is not as good as training set.

The Adaboost performance just improve a little compare with Lasso and ridge. I think one reason is because of high computation consuming, I just try several learning rate which is not small(I try to use small learning rate like 0.1, but the MSE is still decreasing after 1000 iteration, it is hard find global minimum in this case), so my learning rate and number of estimators still need to be improved.

Following pictures is using best model to plot most importance two features. We can see that feature LDAPS_TMax_lapse has very high relevant with output. This is exactly the same with my feature importance list on above.



(x label is standardize)

4. Final Results and Interpretation

For trivial baseline, I compute the mean of all train label and calculate MSE with test set label. For non-trivial baseline, I use linear regression model.

	MSE for validation set
Trivial baseline	10.1327
Non-trivial baseline	2.1081

For the final result, I choose my Random Forest method 2 as final model and apply it to test set. The way is deleting same features on both training set and test set and use all the training data(include validation data) to train Random Forest model with 89 trees. Then use this model to predict test set output and calculate MSE with its true output.

	Number of feature deletion	Number of trees:	Test set MSE
Random Forest method 2	6	89	1.0806

From above, we can see it performs much better than our trivial baseline and non-trivial baseline. The reason why it has best performance I think in two ways. The original data distribution is a non-linear distribution, so use linear regression is hard to get best result. Another reason is dimension reduction. By reducing some not important features not only can help improve computation speed, but also can avoid overfitting. When some bad features which mislead model justification are delete, may even improve the performance.

If I want better performance, I think one way is to choose number of trees more precisely. As my method is from 10 to 300 choose 150 different trees, so for some number of tress which I do not choose may have better result. Another method is adjusting max depth, max features and some other parameters to get better performance.

Out of sample performance:

I transform this regression problem to 2 classification problem by choosing threshold. The threshold I choose is mean of test set's predicted output. If predicted output and true output in same side, we regard it as correct classify.

Hence, the in-sample error is 0.1005. We assume the tolerance δ is 0.1. We just use one hypothesis in test data, so M=1 and N=1053

$$E_{out}(h_g) \leq E_{test}(h_g) + \sqrt{\frac{1}{2N} ln \frac{2M}{\delta}}$$

$E_{out}(h_g)$ ≤0.1005+0.0377 =0.1382 with probability large than 0.9.

Hence, the out of sample error upper bound is 0.1382.

5. Summary and conclusions

From above, I find non-linear regression have better performance than linear regression. In non-linear regression, Random Forest in method 2 has the best performance. I find the best number of features deletions are 6 and trees are 89 which can give me a good test MSE result: 1.0806. This told me sometimes delete some features can get better result. It also shows me the importance of each features, which can help me have better understanding of this problem.

What I am interesting to do next is features selection. In method 2, I delete least important feature one by one and calculate MSE. However, it may exist some features combination which may have better result.

6. References

[1] Saptashwa Bhattacharyya, "Ridge and Lasso Regression: L1 and L2 Regularization", Sep26,2018.[Online].Available:https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

[2] Tony Yiu, "Understanding Random Forest", July 12,2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2