Project 4

A:

1. After we convert the features into scalars, the fitted value-actual value can be shown in figure 1 and the residuals-fitted value can be shown in figure 2.

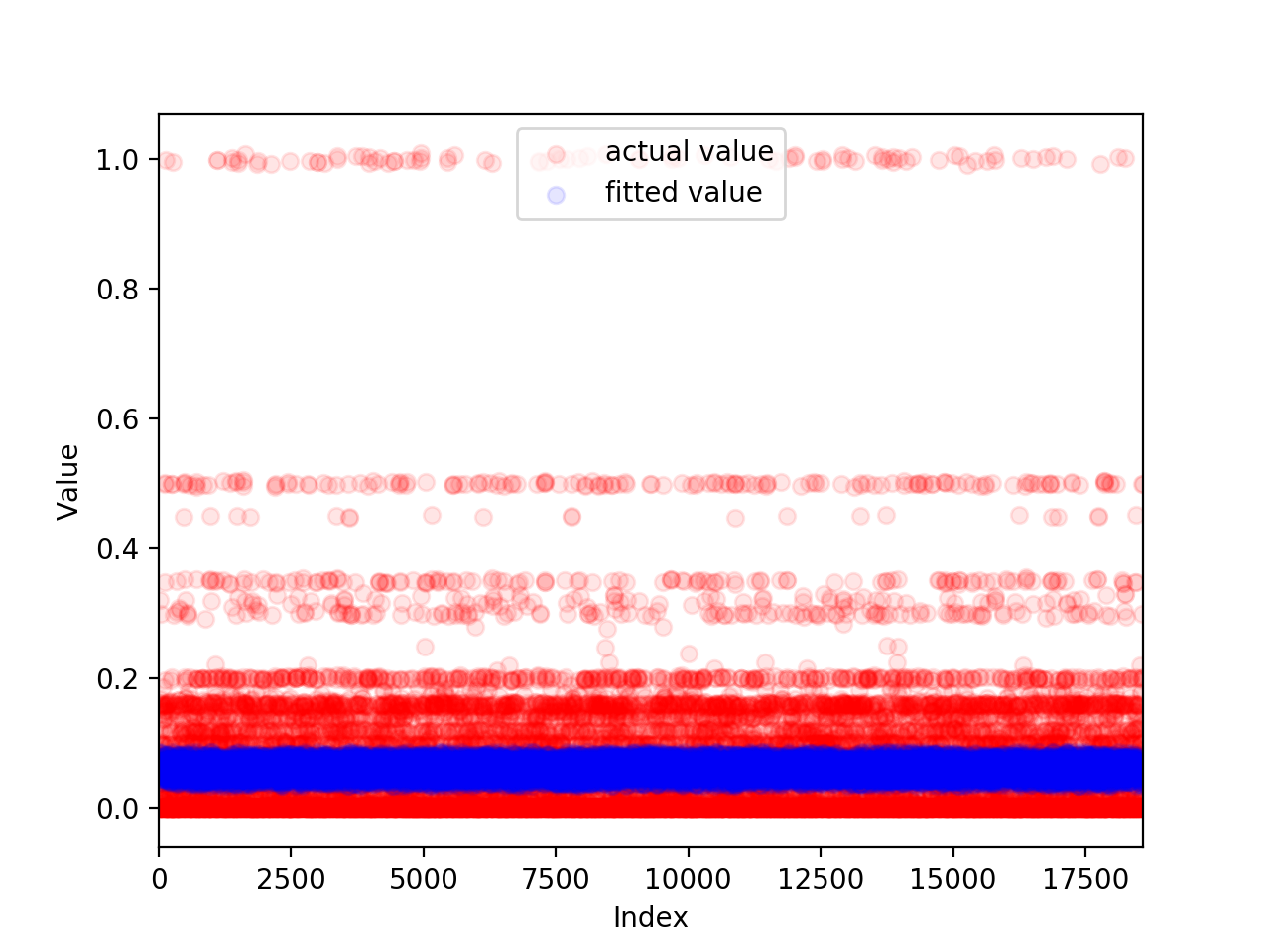


Figure 1 Fitted value-Actual value with linear model

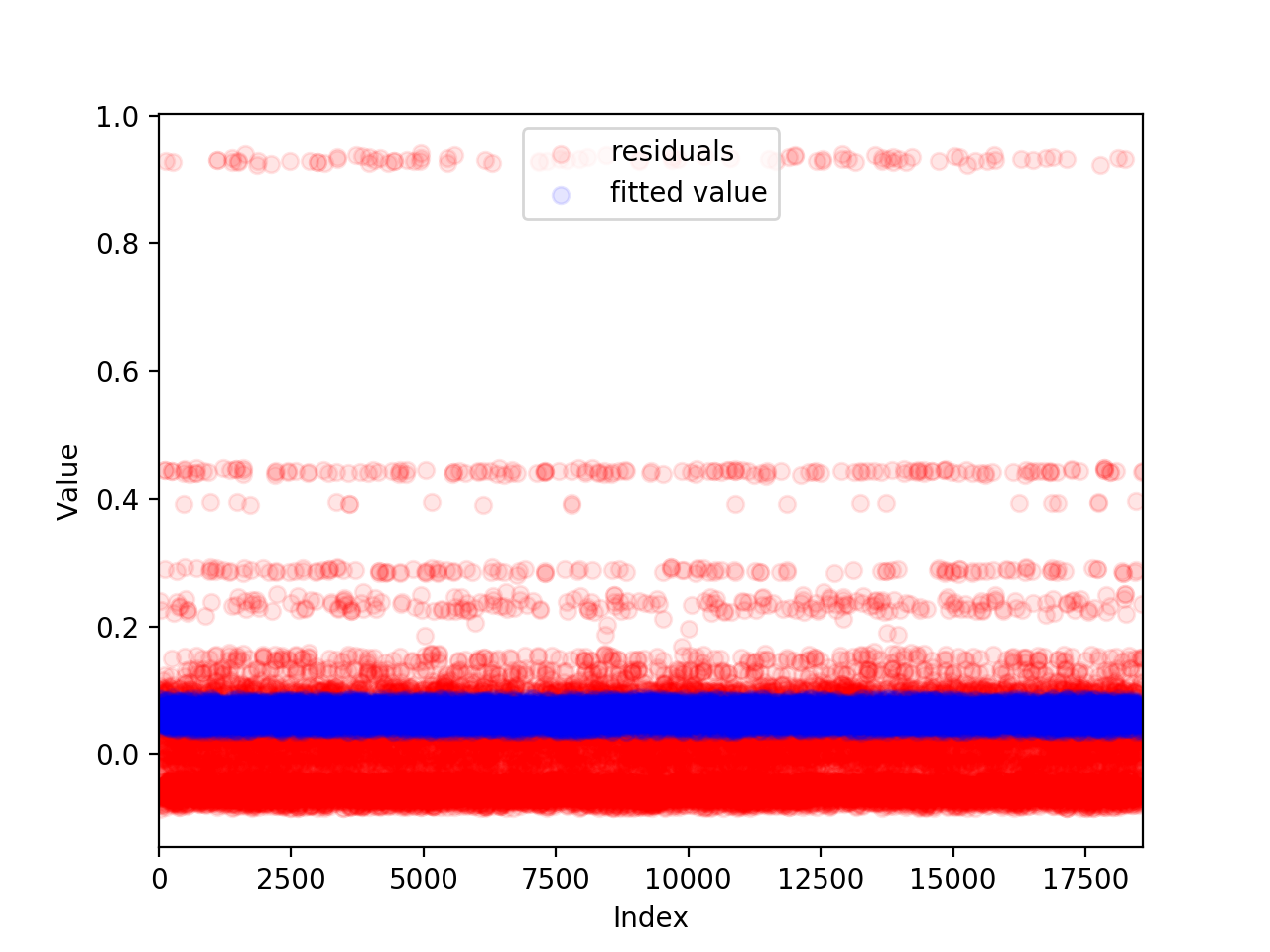


Figure 2 Residual value-Fitted value with linear model

From these two graphs, it is easy to tell that the result is not good. If we predict the back size precisely, the fitted value should be very close to actual values so that the line should lie on y1=y2 which means these two kinds of points should be covered by each other. However, according to the first graph, the line looks like they are not same at all, not even close. This means that no matter how actual value changes, the predicted value remain the same. Thus, it is pretty bad performance. The residual value is the difference between actual value and fitted value. Therefore, the second graph looks similar to the first one. In addition, the average test RMSE is 0.1036.

1. Considered the fact that there are 5 features that we used to predict back-up size. It is possible that these 5 features can have different contribution to prediction since they have different variance and mean. To eliminate this factor, one way is to standardize those features so that they will contribute to prediction equally. The result of standardize features can be shown in figure 3.

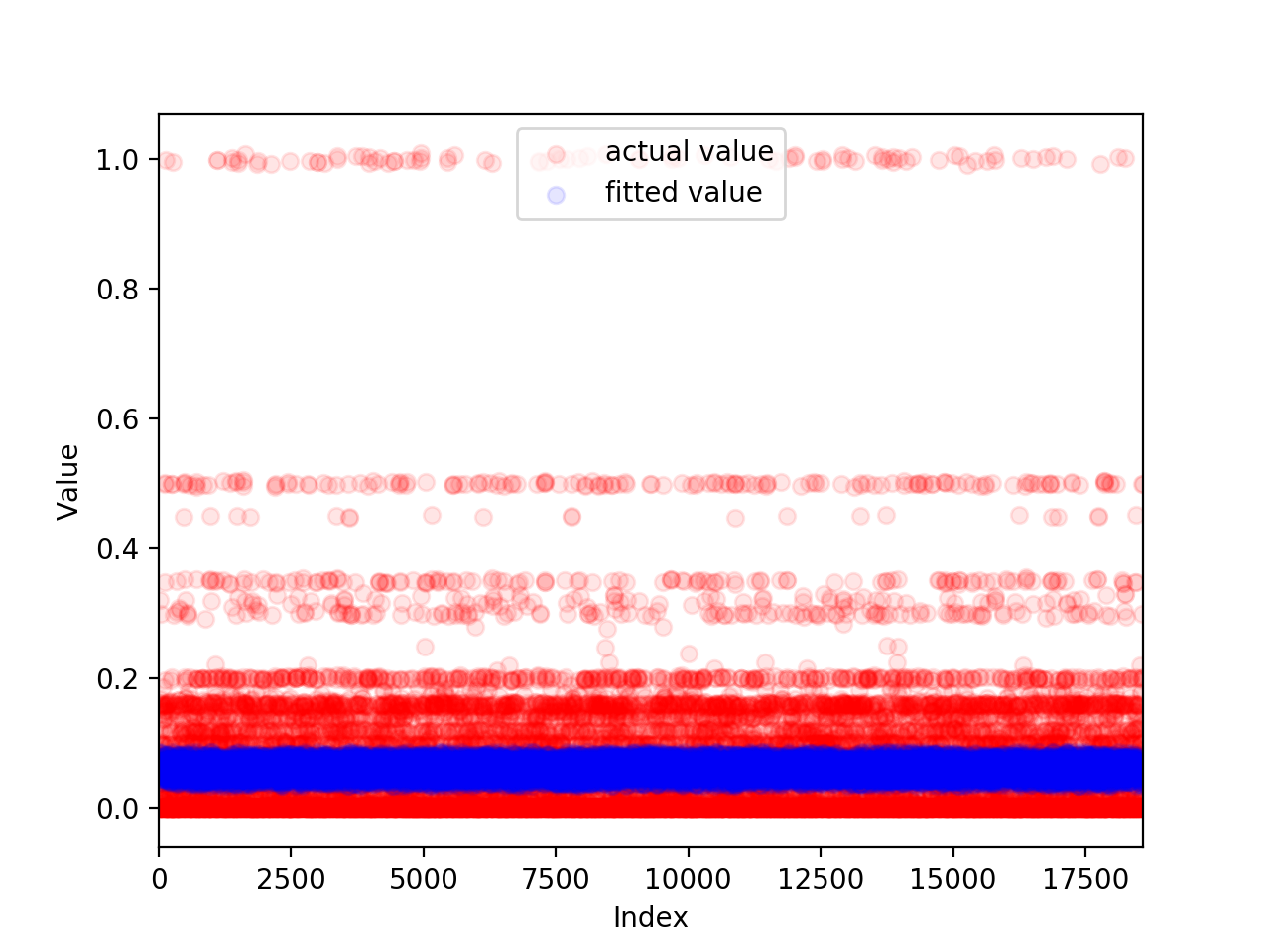


Figure 3 Fitted value-Actual value with linear model(Standardize features)

From graph above, it is obvious that the difference between this and figure 1 is too small to notice. And the fact is that the test RMSE is almost the same as the result of directly using scalar data. The performance of the model doesn’t get improved at all. Maybe standardization of features is not the main factor that influences the result.

1. As mentioned in question 2, the influence of each feature might be different on the result. So, it might be useful to find the most important features in the prediction. It will help us understand what will contribute to the result most and what we should pay less attention to. By using f\_regression and mutual information regression measure, we can get two array which show the score of each feature. In f\_regression, the result is [0.008, 38.816, 150.741, 26.139, 25.320]. From the result, the most important 3 features are backup start time, day of week and workflow id. In mutual information regression measure, the result is [0.0025, 0.2251, 0.2340, 0.7139, 0.4927]. This result shows that the most important 3 features are workflow id, filename and backup start time. Although these 2 measures give different result, it is obvious that both of them show that the first feature-week is much smaller than other features. Besides, the second feature and fifth feature have same magnitude and their difference is just usual compared to the difference with first feature. In conclusion, workflow id and backup start time are the most 2 important features and week is least important feature while the other two feature share similar importance to prediction.

Then we use [day of week, backup start time, workflow id] to predict the backup size, the result is shown in figure 4.

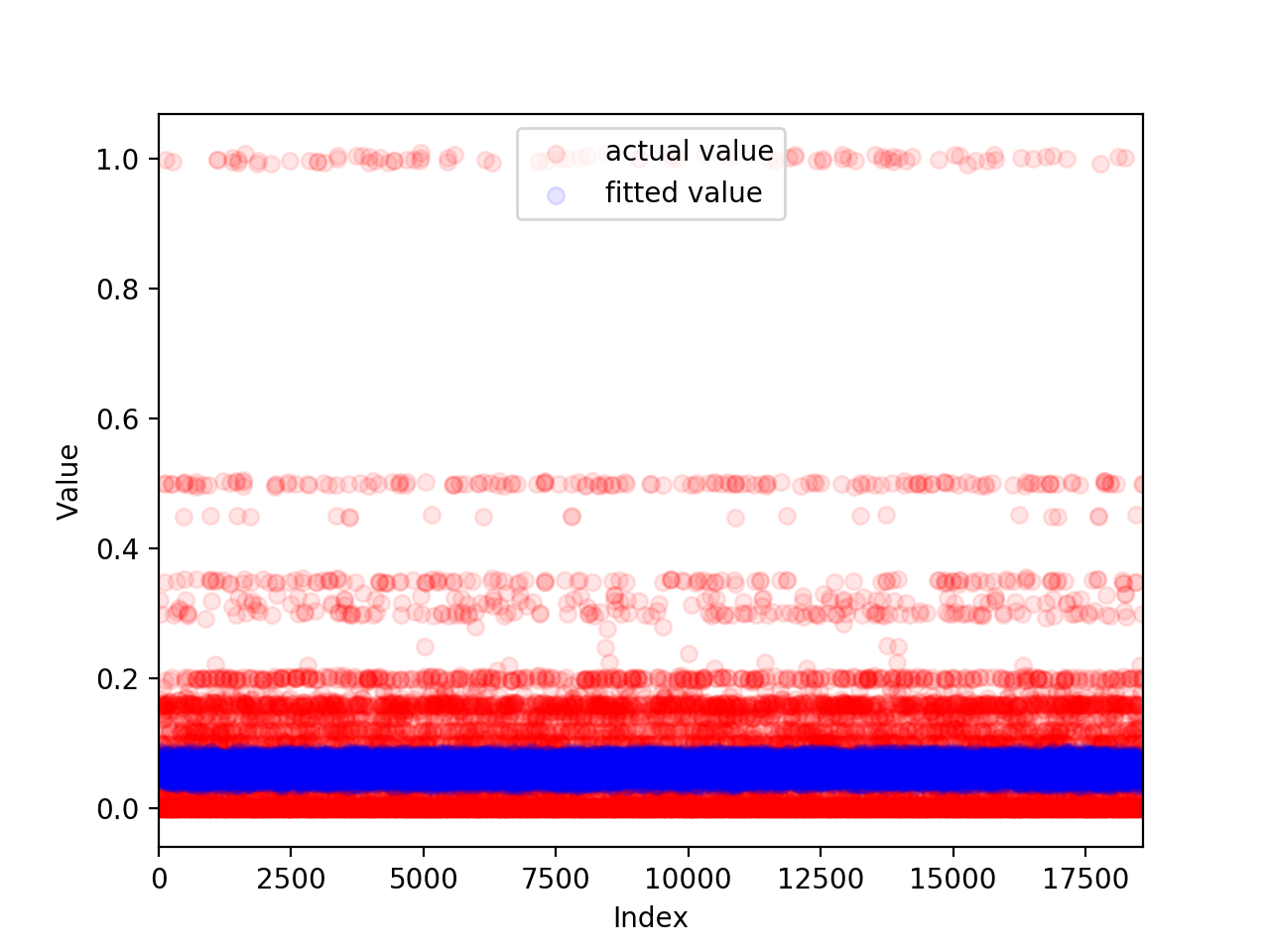


Figure 4 Fitted value-Actual value with selected features

The average RMSE is a little smaller than before, but the difference is too small to ignore. From the graph, the performance of selected features doesn’t get significant improvement and the line still looks like y=c. In other words, the most important features can improve the performance a little, but it can’t change the fact that this linear model has a bad performance on predicting this kind of data.

1. In the first 2 questions, we use scalar to describe the features. This definition of feature may not be exactly right. Thus, we introduce another method of encoding-one-hot-encoding. By using the combinations of different method of encoding 5 features, there are 32 different kinds of combinations. We use 1 to indicate scalar coding and 0 to indicate one-hot-encoding. And we use 1st bit to indicate week, 2nd bit to indicate day of week, 3rd bit to indicate backup start time, 4th bit to indicate workflow id, 5th bit to indicate filename. For example, 00001 means that we only use scalar encoding on week feature, others are encoded by one-hot-encoding, After applying linear model to each of 32 combinations, the average test RMSE and train RMSE of 32 combinations can be shown in figure 5.

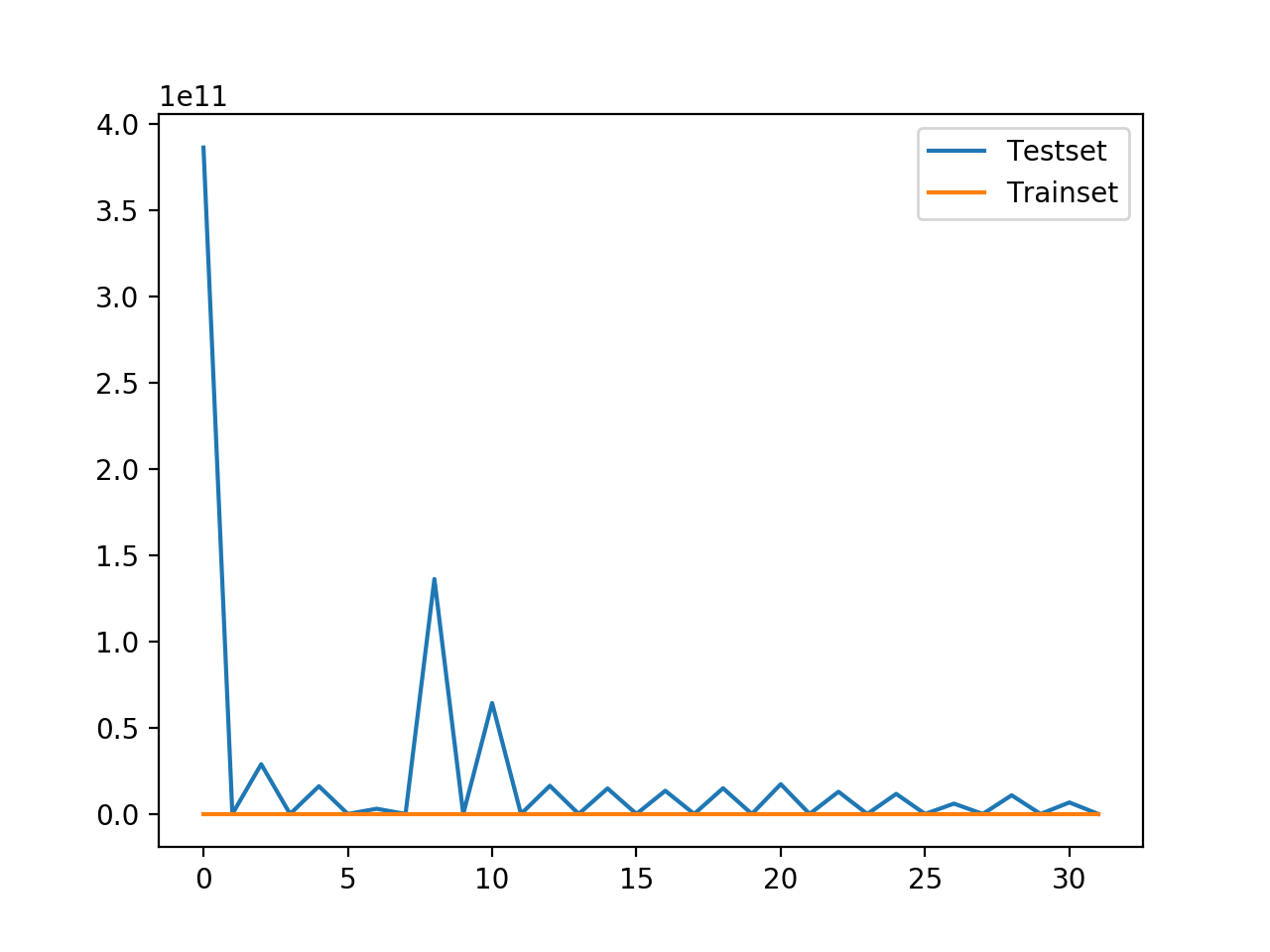


Figure 5 RMSE of trainset and testset of 32 combinations

From the graph above, it is interesting that odd number of combination get normal test RMSE which is less than 1 and close to 0. Nevertheless, all of even number of combination has extremely large test RMSE(>>1). The difference between even and odd number is whether we use one-hot-encoding on week feature or not. That means if we use one-hot-encoding on week, the RMSE will become incredible large. But when we set the shuffle parameter to be true and then the result shows in figure 6.

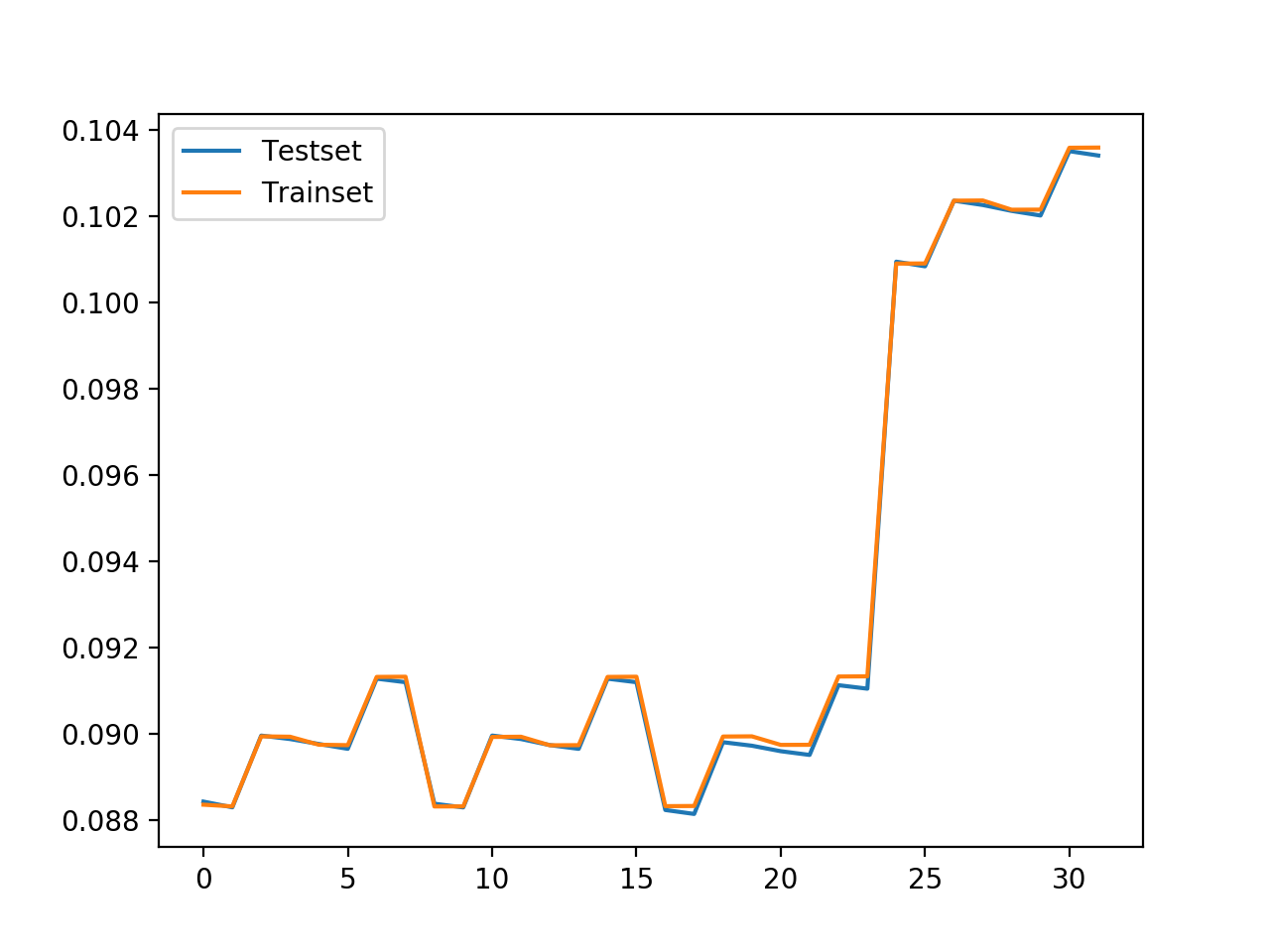


Figure 6 RMSE of trainset and testset of 32 combinations(shuffle=True)

From Figure 6, all of the RMSE of 32 combinations are reasonable now. The reason of this phenomenon might be that the change of choice of K-Fold. If we use shuffle=False which means that the testset and trainset are successive. The difference between i and i+1 point is very small while the backup size may be much larger compared to the input difference. Then the predicted value will be much larger due to this extremely large rate of output to input. However, if we select data point randomly, the similarity of these data points will be much smaller and then the predicted value will be normal. And the two successive points always share same week, then we can’t use one-hot-encoding on week feature because one-hot-encoding will enlarge the change of input if shuffle=True.

The least RMSE appears at combination number 17 which means we use one-hot-encoding on 2-4 features. This is exactly the same 3 features as what we conclude from f\_regression. By encoding these 3 most important features, the main characteristic will be highlighted so that the best result will be achieved. That is why the least RMSE appears at combination number 17.

1. In question 4, we found that there might be some significant increase in some specific combinations, to avoid this happen or decrease the influence of the one-hot-encoding on week, there are 3 linear models we can use to improve the performance- Ridge, Lasso, ElasticNet. By applying these 3 methods to each of 32 combinations while trying different parameter, we can find the best combination and most efficient parameter which fits one particular model.

From the result, the best combination of Ridge model will still be number 17, and the alpha is 0.9. The average test RMSE is 0.088. When it comes to Lasso model, the best combination will be number 20 with alpha=0.01 and the average test RMSE is 0.099. In ElasticNet model, we fixed the l1\_ratio to 0.5 and then sweep alpha from 0.01 to 1. Then the best combination is number 20 with alpha=0.01 as well and the average test RMSE is 0.094. It is interesting that in Lasso and ElasticNet model, the best combination is number 20 which is even number. That means these two model eliminate the influence of one-hot-encoding on week feature. ElasticNet and Lasso model share one similar property that they all includes || ||1 into their model while Ridge doesn’t. This might be the explanation to the sudden increase of RMSE when we apply one-hot-encoding on week feature. This dataset doesn’t fit for linear model since the data might not be linearly distributed. By taking norm 1 into prediction, the result of prediction might be more persuasive and general.