COMP4432 Machine Learning

Competition (Individual Work): Don’t Overfit II Competition

Hui Sze Ming 19050459d

1. Introduction

In this assessment, I have participated in this Kaggle competition “Don't Overfit! II”, which aims to avoid training a model that is too closely fit to a limited set of data and lose the generalization ability. In addition, the older version of the dataset in this competition will be used. The final scores of my best attempt are 0.724 (private score) and 0.732 (public score).

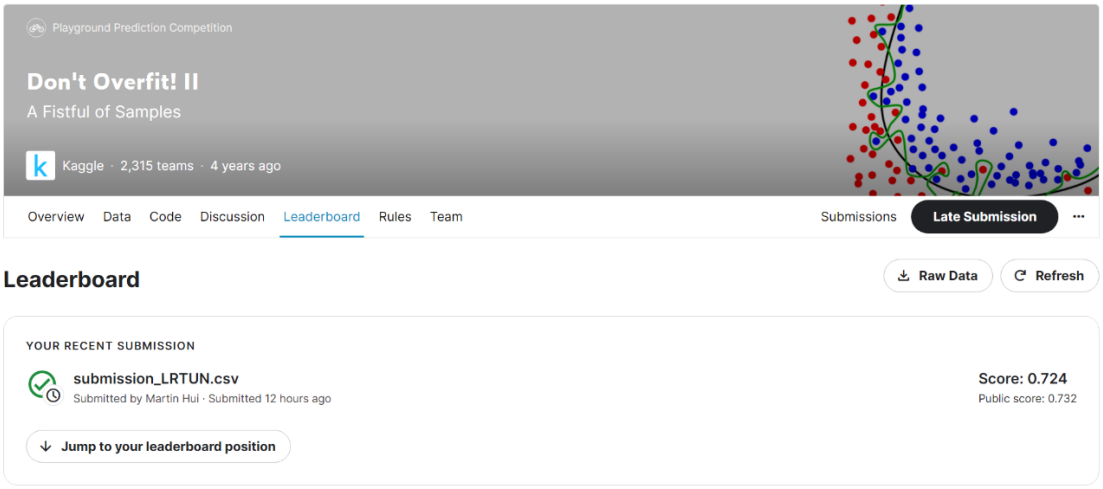

Figure 1. Final Score

2. Dataset

Before training the model and avoiding overfitting, a data overview might be helpful when building the models. The below figure shows the number of instances and the number of features corresponding to the training set and the testing set.

```
Data Overview of the train data:
Number of instances: 250
Number of features: 300


Data Overview of the test data:
Number of instances: 19750
Number of features: 300
```

Figure 2. Dataset Overview

As the figure shows, this data set is large with 300 features without semantic, it is hard to tell the importance of each feature and perform feature engineering. In addition, the testing dataset has much more instances comparing to the training dataset, there are not much data for the model to learn, so it may not be able to reflect the important characteristics in the whole data set and the unseen data. In result, it may behave poorly on generalization and the final model can be overfit to the training data easily.

3. Methodology and Results

(1) Feature Reduction

The first considered method is the feature reduction, which manual selecting and keeping some features, this approach may be useful when some features are not important. Nevertheless, this method will be effective if only having the domain knowledge.

Though that being said, feature reduction according to the calculated correlation still worth a try even the result score is expected to be bad because this approach may remove necessary features without noticing.

In the first submission, correlation between target and other features will be calculated, and features with minimum correlation lower than 0.005 will be

removed. On the other hand, the rest of the features will be training the Logistic Regression Classifier, which has been widely used for binary classification.

After the features reduction, there are only 13 features left for the training part which include the following columns: ['0', '2', '3', '4', '5', '10', '12', '13', '16', '19', '26', '31', '38']. After submission, the score is 0.515.

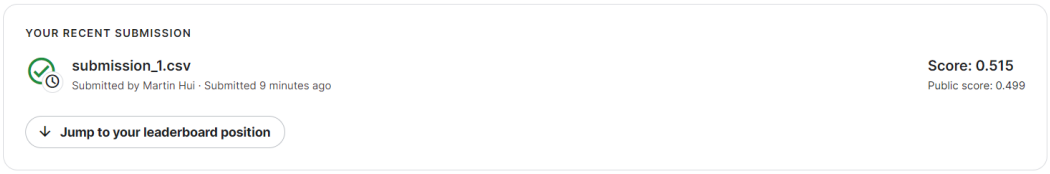

Figure 3. Result of Submission (feature reduction 1)

The reason of the low submission score may because of reducing too many features that are necessary to the model as predicted. Therefore, I had another attempt by removing features with minimum correlation lower than 0.0002, which remain 215 features. After submission, the score is 0.644.

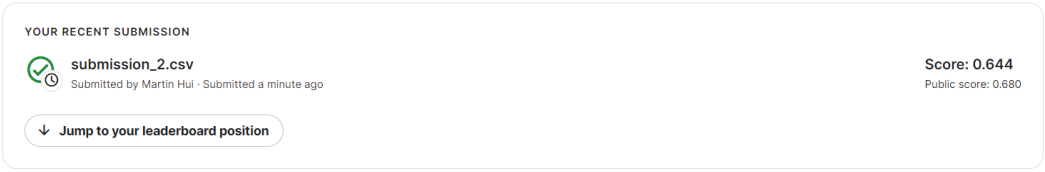

Figure 4. Result of Submission ((feature reduction 2)

In conclude, the data correlation may not be a good criterion for choosing the important features, it may also cause underfitting if we are removing too many features. Additionally, it indicates that feature reduction really requires domain knowledge from the expert in the corresponding field. In other words, it is found that using the correlation as the criterion of features reduction is bad, or feature reduction is not the best approach.

(2) Regularization

The second considered method is the regularization, which keeping all features and reduce their influence by setting smaller values to the parameter. In this part, both ridge regression and LASSO regression will be performed with the logistic regression classifier.

- $\|\theta\|_2$: Ridge Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

- $\|\theta\|_1$: LASSO Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}|\theta_j|\right]$$

Figure 5. Two Types of Regularization Regression[1]

As figure 4 has shown, Ridge Regression takes the square of the coefficients and LASSO Regression takes the magnitude. By introducing the penalty, it avoids training the overfitting model. After the submission, L1 regularization get the score of 0.701, L2 regularization get the score of 0.634.

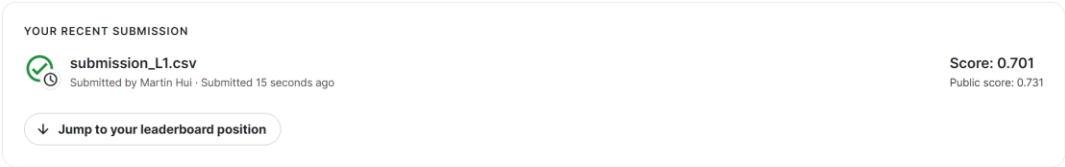

Figure 6. Result of Submission (LASSO)

Figure 7. Result of Submission (Ridge)

As the result, LASSO Regression performs better than Ridge Regression when training the logistic regression classifier in this case. Therefore, I conclusion here is

that LASSO Regression can reduce more unnecessary features comparing to Ridge Regression in this dataset which include many features.

(3) Cross Validation

The third considered method is the k-fold cross validation, which help to find the optimal model on the training set by randomly separating the training set into smaller group and use it for testing, it is expected to help avoiding overfitting. In this part, as having better performance in the last part, LASSO Logistic Regression Classifier will be used as the training model.

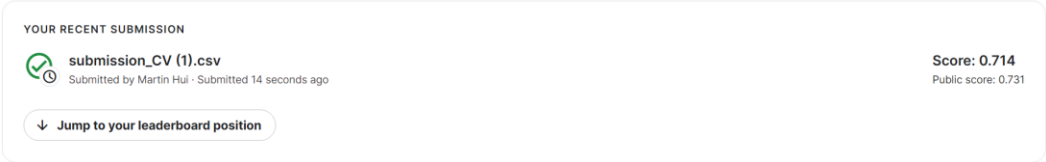

Figure 8. Result of Submission (5-Fold Cross Validation with LASSO)

As the result, cross validation helps to further improve the Logistic Regression Classifier(LASSO), which is as expected.

(4) Ensemble Method

The fourth consideration is the ensemble methods, which included the bagging and boosting. As learnt in class, Random Forest is based on bagging which is an ensemble of decision tree classifiers, and Voting Classifier (not included in class, combined Logistic Regression, Decision Tree and SVM in my attempt) is also one of the bagging method. On the other hand, XG Boost is based on gradient boosting which optimize tree boosting, and AdaBoost is also one of the boosting method. In this part, Random Forest and Voting Classifier are used for the bagging, XG Boost and AdaBoost are used for the boosting.

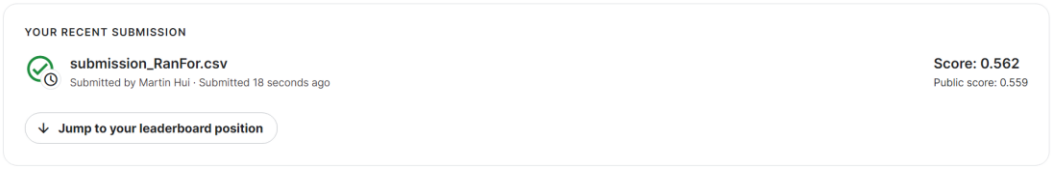

Figure 9. Result of Submission (Random Forest)

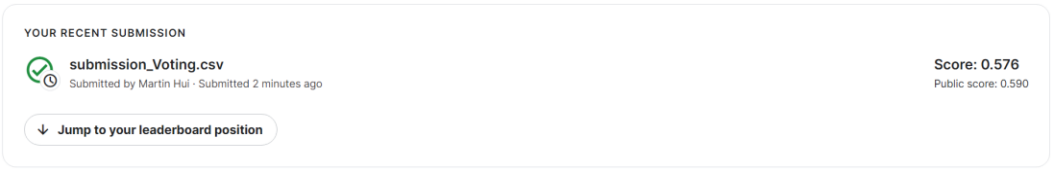

Figure 10. Result of Submission (Voting Classifier)

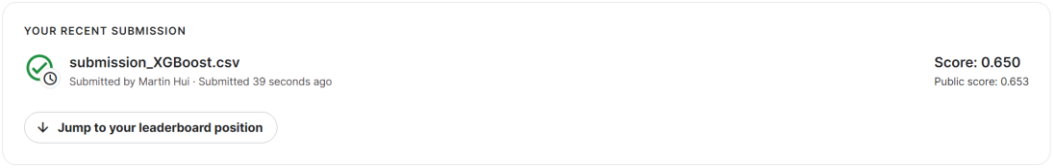

Figure 11. Result of Submission (XGBoosting)

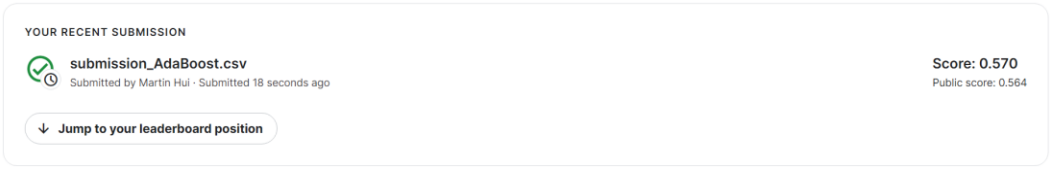

Figure 12. Result of Submission (AdaBoost)

As the result, XGBoosting performs better than Bagging, which means to avoid overfitting problem, it is more suitable to decrease the bias instead of variance in this dataset. However, it is unexpected that XGBoosting is getting lower score against 5-Fold Cross Validation in LASSO, because XGBoosting is supposed to be the Kaggle approach with higher score. In this case, my guess is that the instances in this dataset are linear separable which result better performance on the logistic regression classifier.

(5) Improvement on Logistic Regression with LASSO

After multiple attempts, it seems like using logistic regression(LASSO) is the best approach by far. Therefore, further tunning attempts have been performed.

The first attempt is to train the logistic regression(LASSO) with a lower C parameter 0.1 (previously set to 1) which adding more weight to complexity penalty and increase regularization.

Figure 13. Result of Submission (Tunned LASSO)

The second attempt is to train the logistic regression(LASSO) with 5-fold cross validation, also setting lower C parameters.

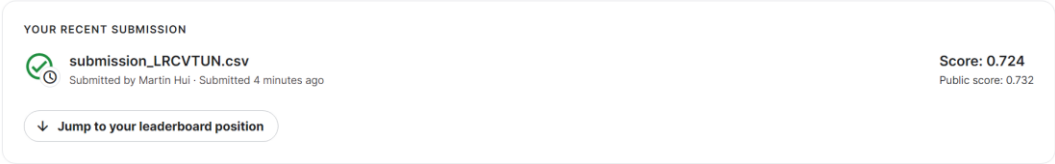

Figure 14. Result of Submission (Tunned 5-Fold Cross Validation LASSO)

As result, high regularization can be performed by adjusting the C parameter to 0.1, both attempts result in better score.

4. Conclusion

In conclusion, features reduction require domain knowledge of the dataset, and ensemble method did not perform as good as expected. On the other hand, L1 regularization has good performance especially after applying the k-fold cross validation and tuning the C parameters, which indicates that the dataset should be linear separable, regularization is effective against overfitting, and simple model with lesser parameters like logistic regression classifier can handle overfitting easier.

Beside of the technical part, after finishing this project I found that finding the best model and method require many attempts, and it is worth it. Originally, I thought XGBoost ensemble method should be the best approach against overfitting, which is not the case for my attempt, it turns out that Logistic Regression Classifier is the best method according to my attempts.

Moreover, I learnt that model tuning is important when building the model, a well-tuned model can achieve much better performance than the model using default parameters, like the C parameter in the logistic regression classifier. Although tuning a model require time and additional knowledge to the package, it is essential for optimization.

5. Reference

[1] Xiao HUNG (2023), "*Lecture4_Overfitting_Evaluation.pdf*", COMP4434 Big Data Analytics.