**Project Report On CS 178#Machine Learning and Data Mining**

You Li # *YOUL10*

Ruiqi Li # *RUIQL10*

Hanze Liu # *HANZEL1*

Kaggle Team Name: IZ\*ONE (Placed 56/293 when we submit)

Canvas Group # 100

Final Score: 0.74038

|  |  |  |  |
| --- | --- | --- | --- |
| Model Name | Training Performance | Validation Performance | Kaggle Result |
| Random Forest | 0.8735 | 0.7545 | 0.71797 |
| Gradient Boosting | 0.912 | 0.7320 | 0.73965 |
| Extreme Tree | 0.935 | 0.6957 | 0.71342 |
| Ada Boosting | 0.995 | 0.722 | 0.73644 |

Model Analysis:

# Please note that the Train-Validation split ratio of our data is 0.70-0.30 and they are all shuffled.

* Random Forest:
  + We developed a random forest model of bagged trees, and the algorithms are implemented by the built-in random forest method within the SKLearn library. We constructed the random forest and bagged 28 of them with various combinations on max\_depth(from 3-9) and min\_sample split(2,3,4,5), and weigh the result on average.
  + We estimated that the result should be around 0.7 as this bagged model obviously overfits. However, we got a really surprising result which is around 0.736. This shows that the potential improvement of this approach is really promising.
  + We have several hypotheses for this model’s disadvantage. First, we may have a small number of entry members (28 may not be enough). Second, the hyperparameters are set poorly and result in overfitting. We used grid-search at the very beginning, but the improvement is not very substantial.
* Gradient Boosting
  + We used XGBoost and grid search for this method, and it works pretty well. In order to train the model well(The most powerful weapon in our arsenal), we used GridSearchCV to train the set of parameters that can get the best result.
  + As the split ratio is already considerable enough, we set the subsample parameter anchored to 0.75 in order to avoid underfitting. We started by tuning the combination of max\_depth and min\_child\_weight, and then we tuned gamma, and then we tuned alpha and collapse\_by\_node. This model performs brilliantly and gets a validation roc score of 0.726.
  + In order to reduce the entropy, we also trained a couple of more GBDT using the full X and Y data. We then fixed the subsample parameter to 0.80, and reset the random seed for a couple of times. We bagged those learners along with those learners with split data. The bagged prediction is not very different compared to the one that used data splitting. But the outcome is really considerable.
* Extreme Tree:
  + The extreme tree is introduced as a random maker within our model. It is only designated to make the overall prediction more balanced, as the outliers it predicted can sometimes become a tiebreaker.
  + We didn’t train this model very carefully, and we only used 500 estimators and 200 depth. This is sufficient to become the tie-breaker, and it can reduce the overall overfit of our final prediction.
  + We bagged 12 extreme trees with various mean\_sample leaves (3,4,5,6), and we used 3 various maximization methods(log2 and float). The overall predicted outcome is only around 0.71342, which is not promising, but it’s sufficient for our purpose.
* Ada Boosting:
  + We used Ada Boosting just the same as how we tuned our XGBoost method. We used GridSearch to tune it, getting a result that is slightly weaker than our XGBoost method.
* Final Conclusion:
  + We eventually merged the outcome of those 4 learners by giving them a weight, which is “Random Forest:10, Gradient Boosting:14, Extreme Tree:4, Ada Boosting:8”. We used some grid search to tune the performance of this, and this is the best outcome we can get on Kaggle. I think this ratio is very reasonable as XGBoost and Random Forest method shows strong ability in prediction, and so does the rest of the models.
  + We think that we can also improve our data by normalizing each feature and in each entry. It is reasonable that the outliers in our data may influence our model’s robustness, and we shall normalize it carefully. However, due to our lack of proficiency in statistics, we don’t have many feasible measures to achieve it. We will do our best to achieve this in the future.