IFT6390 - Kaggle Competition 2 Report

Jiarui Lu
Student ID: 20212244
`https://www.kaggle.com/lujiarui`

Xinyi He
Student ID: 20202160
`https://www.kaggle.com/hexinyialexia`

1 Introduction

This Kaggle competition is about the prediction of crop harvest on the subset of CropHarvest[2] dataset. In this binary classification task, each data point with feature $x \in \mathbb{R}^{18\times 12}$ is expected to be classified as *crop* (1) or *non-crop* (0) land. Machine learning models are implemented and train on a dataset of over $60,000$ labeled samples. In our attempt, random forest (RF), AdaBoost and XGBoost are implemented and tuned to predict the test label. Moreover, we normalize the features before learning/prediction and adopt 10-fold cross-validation on $F1$ score to determine best model. The best achieved performance is 0.99757 on Kaggle public leaderboard, 0.99779 on Kaggle private leaderboard both by RF.

2 Methods

2.1 Feature normalization

Before passing the input features to machine learning model, we observe the scale and value distribution of each dimension of features. Large variance exists among different features and thus we decide to perform normalization for each dimension. For any feature $x_j^{(i)} \in \mathbb{R}$ that belongs to sample $x^{(i)}$, it is converted into:

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j}, i = 1, 2, \ldots, N; j = 1, 2, \ldots, d, \tag{1}$$

where μ_j is the mean of the jth feature among the training set, σ_j is the standard deviation of the jth feature among the training set.

2.2 Predictive models

Due to the feature characteristic of CropHarvest, which does not have significantly graph/sequential/grid structure, no GNN/RNN/CNN is used in our study. Instead, we consider ensemble methods for their great performance on predictions of continuous data and good bias or variance reduction. Specifically, we implemented and tested the Random Forest (RF), AdaBoost and XGBoost algorithms.

RF performs bagging algorithms to generate a set of decision trees and perform majority vote to predict new label. Also, RF does random feature selection for predictors. These two characteristics largely reduce the variance of learned models; AdaBoost used weighted dataset from different training stage to obtain a set of weak learners (eg. decision trees) and make prediction based on weighted majority vote; XGBoost improves GBDT by adding a regularization term to the loss function for controlling the complexity of the model, the prediction is made by the weighted sum of the leaf nodes. Boosting algorithm enables it to have bias reduced largely.

Report of Kaggle 2 **CropHarvest** for IFT6390 A2021. Team name: **NO IDEA**.

Table 1: Best hyperparameter(s) for each model and average F1 score over cross-validation.

Model	Averaged F1
Random Forest	0.8892
AdaBoost	0.8375
XGBoost	0.8885

We implement these three methods based on Sci-Kit Learn [1], a popular machine learning package built for Python.

3 Results

As the experiment, we perform 10-fold cross validation (CV) with random split to evaluate the predictive performance for RF, and use F1-score as our metrics, defined as follow:

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}. \tag{2}$$

After 10-fold CV, the F1 performances of learned model on training set and validation set (a list of 10, respectively) are averaged respectively to make the final decision on best model selection. To be specific, we tune the max depth of decision trees from 5 to 95, with step-size 5 for Random Forest. The ensemble size or the number of RF/AdaBoost estimators is set to be 100. The base estimator used in AdaBoost is the decision trees with max depth $= 1$ by default in sci-kit learn.

For Kaggle submission, we use RF with *max_depth* $= 30$ and *max_depth* $= 65$ as our two submissions. The best achieved performance is 0.99757 on Kaggle public leaderboard, 0.99779 on Kaggle private leaderboard both by RF.

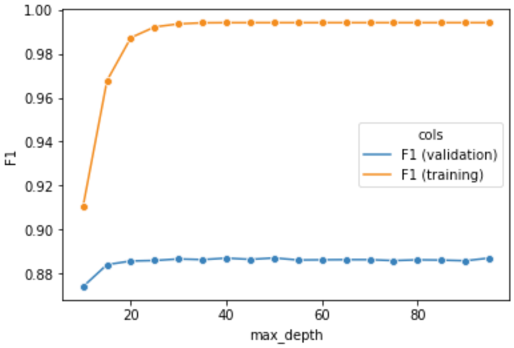

Figure 1: Moving performance with maximum depth for RF model. Best *max_depth*$= 65$.

4 Discussion

In this competition, we use three ensemble methods. Among them, Random Forest belongs to the Bagging algorithms; Adaboost and XGBoost belong to the boosting algorithms. Adaboost adjusts the weights according to the error rate and adds weak classifier. XGBoost keeps reducing the residuals, adding new trees, and building new models in the direction of the inverse gradient.

Random forest performs well on the test set, reduces the risk of overfitting since the results are averaged over all predictions and makes the model more robust to outliers. Its main drawbacks include the poor performance on imbalanced data (rare outcomes or rare predictors) and lack of an interpretable model.

AdaBoost very good use of weak classifiers for cascading, has a high degree of precision; Relative to the Random Forest Algorithm, AdaBoost fully considers the weight of each classifier;

Adaboost is sensitive to outliers, and outliers may have higher weights in the iterations, affecting the final prediction accuracy. Also data imbalance leads to a decrease in classification accuracy.

XGBoost is capable of parallel computing, pre-sorting the features, saving the structure, and reusing this structure in subsequent computations. XGBoost output importance of each feature, can be used for feature selection, fast to interpret. However, it is difficult to tune as there are too many hyperparameters, overfitting possible if parameters not tuned properly.

Eventually, after experimentation, we found that random forests worked best for this task. The reason is that Bagging is a technique to reduce the variance, while Boosting is a technique to reduce the bias. The bias of all three models are small, so reducing the variance becomes more important. Therefore the f1 score of Random Forest is the highest among the three.

In the future, as different data types their processing differs and need to be handled separately. Therefore if we have the knowledge of related aspects, we can perform better feature scaling and thus achieve better results.

Statement of contributions

This project is individually completed by team members **Jiarui Lu** and **Xinyi He** as required by IFT-6390 for graduate student. No collaboration to be stated.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] Gabriel Tseng, Ivan Zvonkov, Catherine Lilian Nakalembe, and Hannah Kerner. Cropharvest: A global dataset for crop-type classification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.