Individual reflection (Xiaomeng Jia)

After reading the assessment together, we think this time we will something similar to the last project, but we do make lots of innovations and improvements. My individual reflection is written as follows:

1. Rationale behind the inference goal.

As we could determine the inference goal on ourselves, at first, we planned to make classifications and predictions upon the 'protocol_type', because we found that it contains three main types and it's a good attempt to work on it. After discussing, Long said that it's easy to find the protocol types by Wireshark, so it's meaningless to do that thing. Finally, we decided to do something similar to the last project, but we further classify the network flow into 4 types: smurf, neptune, normal and the rest are renamed as OTHER. By doing this, we could find how a model performs when it classifies and predicts a certain type and find out which one is the easiest to be recognized by machine.

2. Baseline model: Decision tree

Decision tree is an effective and practical classifier model. It is a tree structure, with the help of tree branch structure to classify a dataset. In the decision tree, each internal node represents the determination of an attribute, the branch of the node represents the determination result, and the leaf node represents a category. In this group project, Long and I use the ensemble algorithm based on decision tree classifier. We are very interested in the performance comparison between single classifier and ensemble classifier. In addition, Lu's model, Naïve Bayes, is a model based on conditional decision, which is similar to the algorithm of tree's branch decision, so we decided to use the decision tree as our baseline model.

3. Mathematics behind the model

3.1Introduction

As a new and highly flexible machine learning algorithm, random forest (RF) has a wide range of application prospects. Random forest is an algorithm that integrates multiple trees through the idea of ensemble learning. Its basic unit is the decision tree, and its essence belongs to a big branch of machine learning -- ensemble learning. Random forest seems easy to understand, in fact, its principle is relatively complicated. To fully understand its working principle, we need a lot of basic knowledge about machine learning. Let's talk about it briefly.

Firstly, we will introduce the sample selection method of random forest, Bagging: Bagging is the abbreviation of Bootstrap AGGregatING. Bagging is based on bootstrap sampling. Given a dataset containing N samples, first take a sample randomly as the training set, then put the sample back to the original dataset, repeat the process N times, and get the training set containing N samples. Some samples in the original data set

appear repeatedly in the training set, some never appear. For each tree, the training set is different, and it may contain a large number of repeated samples.

Therefore, we can say that Bagging + Decision tree = Random Forest. If we specify the number of trees in the forest, assuming T, we can get T decision tree classifiers. When the random forest is used to predict the prediction set, each tree has its own decision category for each sample, and the final category is generated by simple majority voting.

3.2 Key points in random forest

There are two key words in the name of random forest, one is "random", the other is "forest". We have introduced "forest" before, one is called a tree, then hundreds of trees can be called a forest, which is the embodiment of the main idea of random forest -- the ensemble learning. "Random" has two meanings. The first one is the randomness of samples, which has been mentioned before. It is generated randomly through bagging. The second is the randomness of features. Suppose that the feature dimension of each sample is M, specify a m much smaller than M, and randomly select m features from M features as a sub feature-set. In this way, the features in the training set corresponding to each tree are also random.

3.3 Model optimization

During the construction of each tree, since bootstrap sampling is carried out for the original dataset, about 1/3 of the samples do not participate in the generation of training set for the k-th tree. We call the remaining 1/3 the OOB (Out of Bag) samples of the k-th tree. For each sample, about 1/3 of the trees treat it as an OOB sample. These trees are used to predict the category of the OOB sample, and then a simple majority vote is taken as the classification result of the sample. Finally, the ratio of the number of misclassifications to the total number of samples in the original dataset is seen as the OOB misclassification rate of the random forest.

First, I determine an 'mtry' value to minimize the OOB error rate of the model. I make a 'for' loop function to generate a random forest in each loop. The loop variable is the number of features selected by each tree, increasing from 1 to M-1(M is the total number of features in the original dataset). Then I check the OOB error of each forest, find the smallest one (via the 'which.min' function), and use its 'mtry' value (14).

Next, I determine the number of trees in the forest to minimize OOB error. I set a large enough number of trees (1000) as the threshold and plot the graph of the forest. It is found that when 'nTree' is greater than 200, the error of the model tends to be stable. Increasing the number of trees will only increase the computational complexity of the model, but not affect the accuracy.

3.4 Analysis of variables' importance

After all the training and prediction work, Jia is interested in the importance of

each variable, that is, the impact of each variable on the results. The importance of each variable is measured by their 'MeanDecreaseGini'. The larger the value, the more important the variable is. It can be seen from the graph that the non-numerical variables in the original dataset are very important. Therefore, in order to make the model of other people in the group applicable to the dataset, it is a wise choice to deal with the non-numerical variables with dummy variables.

4. Relations to other methods.

Long and my methods are very similar, they are based on the decision tree ensemble algorithm. In the aspect of prediction accuracy, our accuracy is 99.9%, have better classification and prediction ability upon the dataset. In terms of dealing with variables, our tree structure algorithm can deal with non-numerical variables directly, which provides great convenience for code implementation. In the prediction of network flow types, random forest can be multi classified and predicted, while GBM can only make binary prediction, so Long needs to train four classifiers, and then make four binary predictions, which greatly increases the coding workload.

Lu's method is Naïve Bayes, which is an algorithm of classification and prediction based on conditional decision. In terms of prediction accuracy, our accuracy is 99%, and random forest is slightly higher (99.9%). In terms of dealing with variables, his model can’t deal with non-numerical variables directly, it must use dummy variable method to transform, and random forest does not need this step of work. In the prediction of network flow types, both random forest and Naïve Bayes can carry out multiple prediction, which is a very convenient prediction function. Only one classifier needs to be trained, four different classification and prediction results can be obtained.

Huang's method is multi-layer perceptron, which is an algorithm based on neural network structure. In terms of dealing with variables, similar to Lu's model, his model can't deal with non-numerical variables directly, so dummy variable method must be used for transformation. When predicting the network flow type, his model is similar to Long's model, which can't be multi classified, so he also needs to train four classifiers and make four times of binary prediction.

Based on the above analysis, the performance of random forest is better than the other three methods in classifying and predicting the network flow types of the dataset. Its ability to deal with non-numerical variables and multi classification is its biggest advantage.

5. The differences.

In this assessment, the inference goal is defined by ourselves, so we have more options to decide what to do, though we finally choose to do something similar to last project. At first, we planned to make classification and prediction of the protocol types, as our new inference goal. After discussing, we find it easy to find out which type it is (we can see the protocol type in Wireshark), then we decide to do something which

could not be seen explicitly.

In the last project, some used relatively simple methods, but we are required to use non-trivial methods this time, which I think is the most important thing. These non-trivial methods have more parameters and more complicated mechanisms, so it's a good attempt to optimize our models, during this process, we could attain insightful knowledge about the model. Besides, the comparison is made between the baseline model and ourselves, which give us a chance to find out how these methods of classification improved and understand the mechanism of more models.

6. Results and conclusions

At first, our team planned to use ROC curve as the visualization of the final result. However, Long and Huang's models can make multi classification and prediction, so if we use ROC curve, Lu and I will each provide one curve, while Long and Huang will each provide four curves, which is meaningless to make any comparison. Finally, we choose the confusion matrix as the presentation of the final results, from which we can clearly see the overall accuracy and sensitivity and specificity for each type. In terms of my random forest model, the overall accuracy is 99.9%, a high enough accuracy in practical use. I find it is better at predicting Smurf and Neptune, which suggests that these two types seem to have rather distinctive features. It should be noted that random forest makes lots of 'non-normal to normal' predictions and these misprediction mostly occur in 'OTHER' type, so I suppose if I classify the network flow into more types, the performance of my model would be enhanced.

7. Group dynamic

Learned from last project, this time I'm more willing to ask team members for help and happy to discuss with each other. Our group meet at least 4 times a week and we always work on this project in Huang's flat. Through this project, I deeply realized the importance of group cooperation, which greatly improved my learning efficiency, and also improved the team members' understanding of each model. In the report and my reflection, the information and analysis of others' model are all based on group discussions. For me alone, it's a really successful group project.