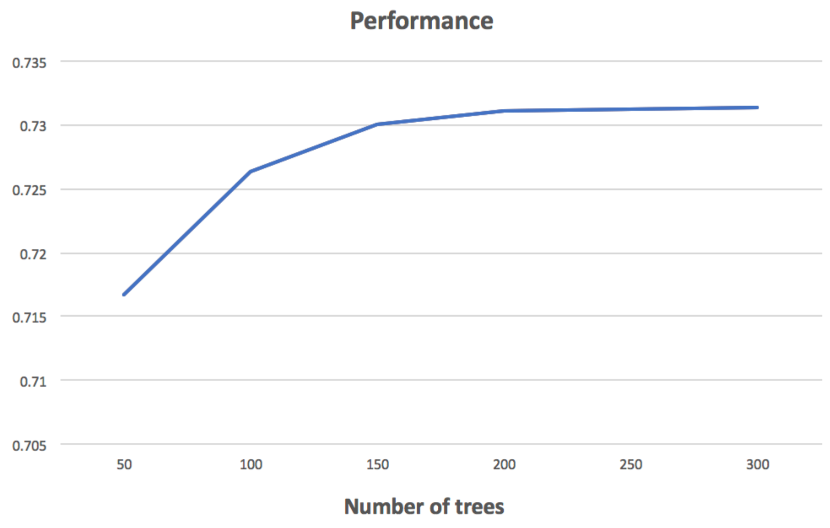
Random Forest

Wiki: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

In our project, we use random forest for multi-class classification. We used "RandomForestClassifier" from sklearn.ensemble to build random forest classifiers. The most import part we did in random forest is parameter tuning.

We take four parameter into consideration. They are:

a) n_estimators. It represents the number of trees in the forest. When we increase the number of trees in the forests, the performance was improved, but the speed of the code becomes slower. So we set a series of n_estimators to do experiments to find out the optimal n_estimators. This is the result of our experiment:

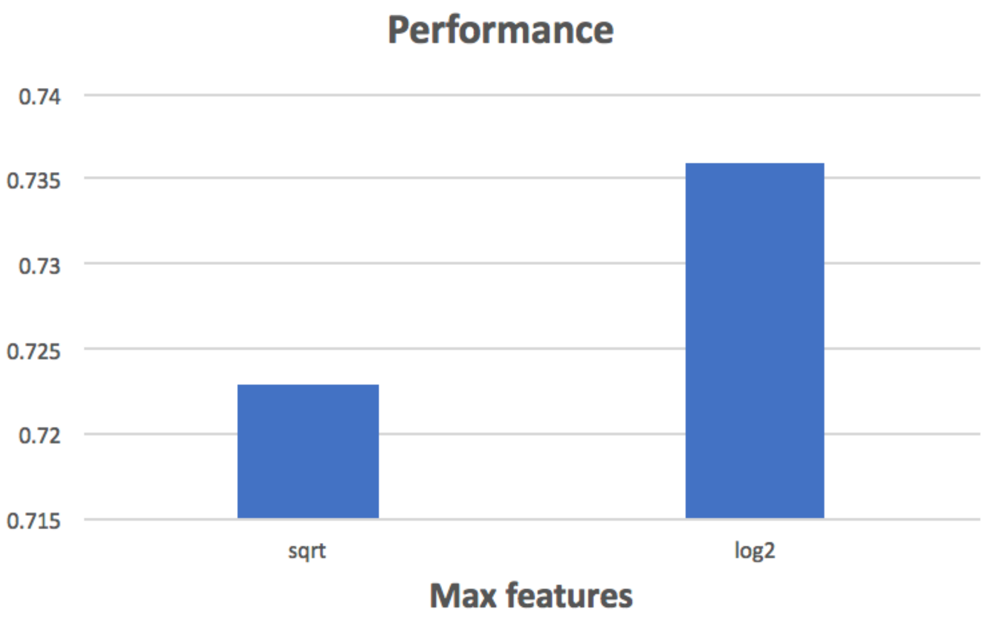


In order to obtain a good performance and a fast speed, we choose n_estimators = 200 as our optimal parameter.

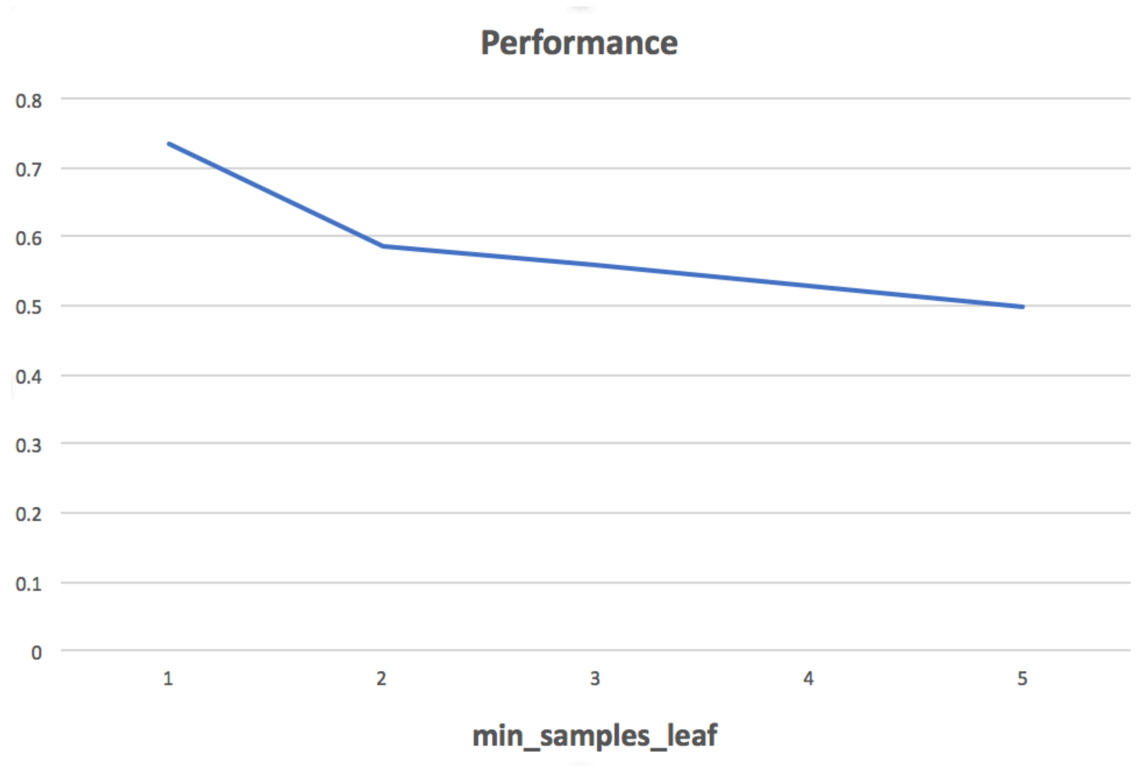
b) max_features. The number of features to consider when looking for the best split. There are different types of max_features we can choose. The most frequent used tree types are:

   a. "sqrt", then max_features=sqrt(n_features)
   b. "log2", then max_features=log2(n_features)
   c. None, then max_features=n_features

   If the max_features is too large, then the speed would be too slow. Also, reduce the maximum feature number it can have, avoids over-fitting. Empirical good default values are max_features=n_features for regression problems, and max_features=sqrt(n_features) for classification tasks max_features. In our experiment, we compared "sqrt" and "log2". It turns out that "log2" have better performance.

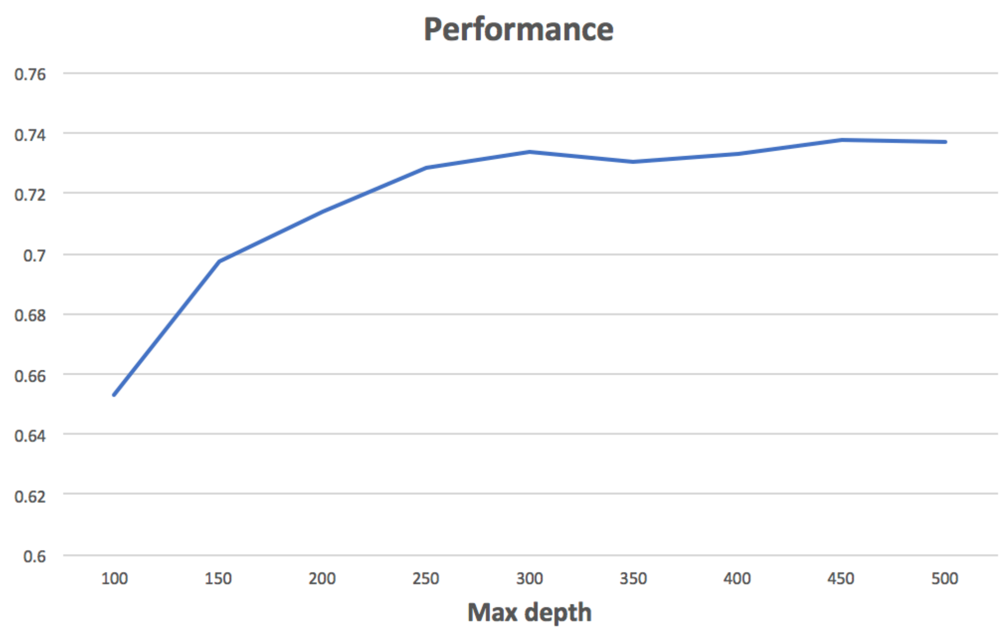


c) Min_samples_leaf. The minimum number of samples required to be at a leaf node. If int, then consider min_samples_leaf as the minimum number.



In our project, we set 1 to min_samples_leaf.

d) max_depth. The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. This is the result of our experiments:



In this figure, it shows that add the depth of tree can improve the performance. In our case, since we only have 200 trees in the forest, we can just set this parameter as default and let nodes expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

Overall, the best performace of Random forest is 73.7%.