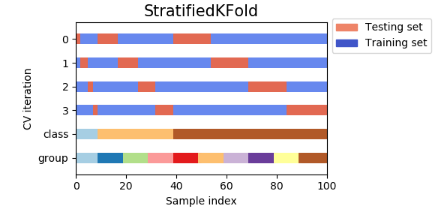
**Random Forest:**

One of the most important parameters in Random Forest is the number of trees (n\_estimators), and the number of features (max\_features): number of features to consider at every split.

To evaluate different values for each hyperparameter, we use the function GridSearchCV, we use a random sample from the grid, using K-Fold CV with each combination of values. The parameters of the estimator used to apply these methods are optimized by cross-validated search over parameter settings. GridSearchCV implements a “fit” and a “score” method. It also implements “predict”, “predict\_proba”, “decision\_function”, “transform” and “inverse\_transform” if they are implemented in the estimator used[[1]](#footnote-1).

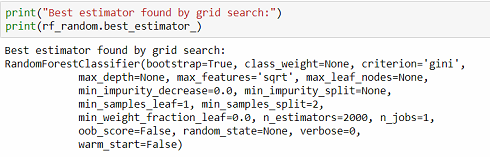
The data is split into train/test sets. The dataset is split into **n\_splits** consecutive folds without shuffling. Each fold is used once as a validation while k-1 remaining folds form the training set. [StratifiedKFold](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html" \l "sklearn.model_selection.StratifiedKFold" \o "sklearn.model_selection.StratifiedKFold) is a variation of *k-fold* which returns *stratified* folds: each set contains approximately the same percentage of samples of each target class as the complete set[[2]](#footnote-2).



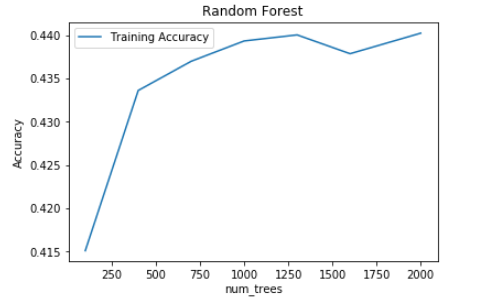
We test number of trees (n\_estimators) between [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000] and max\_features between ‘log2’ and ‘sqrt’, where max\_features=log2(number of features) or max\_features=sqrt(number of features)

1. **CIFAR-10**

The best parameter estimated by [StratifiedKFold](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html#sklearn.model_selection.StratifiedKFold) was 2000 number of trees and max\_features=’sqrt’.



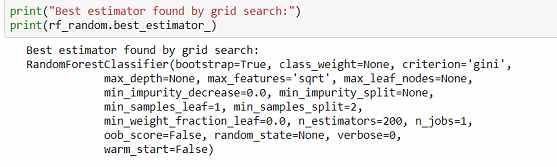
If we take some values of n\_estimators and max\_features=’sqrt’, and use the testing set, we can see the relationship of the Accuracy and number of trees chosen following:



The best accuracy obtained with the test set was 2000 number of trees. After the application of the algorithm with 2000 number of trees we obtained an accuracy of 44.95%

1. **Adult Database**

Using the function GridSearchCV to evaluate the algorithm with the hyperparameters number of tress and max\_features, we see the best estimator was found when n\_estimators=200 and max\_features='sqrt'.



Checking the accuracy using the training set with max\_features='sqrt' and different values for n\_estimators, the best value was obtained using n\_estimators=1600, the second bigger accuracy was obtained with n\_estimators=200



When we used these hyperparameters with our test set, we obtained an accuracy of 99.91%

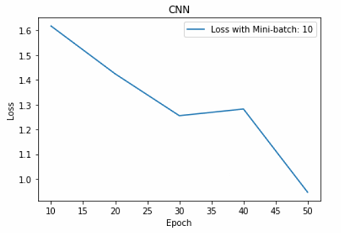
**CNN:**

1. **CFAR-10**

In order to do the performance analysis for the CNN algorithm over CFAR-10, we’re going to use the hyperparameters epochs and mini-batch size. The values used for epochs are [10,20,30,40,50] and for the mini-batch size are [10,50,100,150].

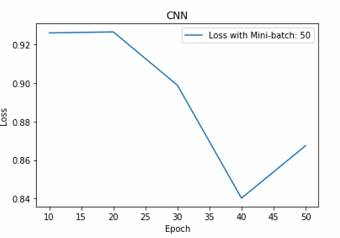
* With Mini-batch: 10

Final Loss: 0.9462



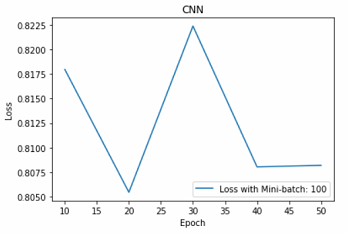
* With Mini-batch: 50

Final Loss: 0.8674



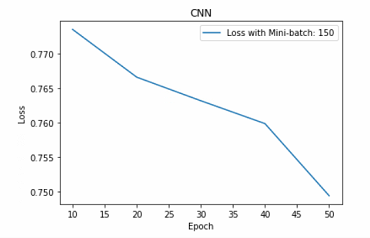
* With Mini-batch: 100

Final Loss: 0.8081



* With Mini-batch: 150

Final Loss: 0.7994



In general, we can see that while epoch increases, the loss value decreases. When the mini-batch value is bigger, the algorithm optimization will progress faster to the lower loss, so in this case, the best hyperparameter values for our model are Mini-batch=150 and Epoch=50. We obtained an accuracy of 65% using this model with the test set.

1. https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html [↑](#footnote-ref-1)
2. https://scikit-learn.org/stable/modules/cross\_validation.html [↑](#footnote-ref-2)