Foundations Of Machine Learning

Assignment-3

Hemanth K
ES19BTECH11003

REPORT ANALYSIS

4.a)For this, I have taken the reference of my submission from Assignment-1 i.e, Decision tree code and I have made a few changes and then started implementing the random forest code.

For random forest Initially I have bootstrapped the input data and then built the decision tree using that bootstrapped data and initially fixed the max_Depth=4 and n_features=2.

After construction of forest we will start predicting on validation data set using the function RF_predict and later we will calculate the accuracy using calculateAccuracy function defined above.

Finally the accuracy obtained using random forest is 0.9246376811594202

b)m(the number of features used for best split) is the number of trees you want to build before taking the maximum voting or averages of predictions. As we know, a higher number of trees give you better performance but makes your code slower. So we should choose as high value as our processor can handle because this makes your predictions stronger and more stable.

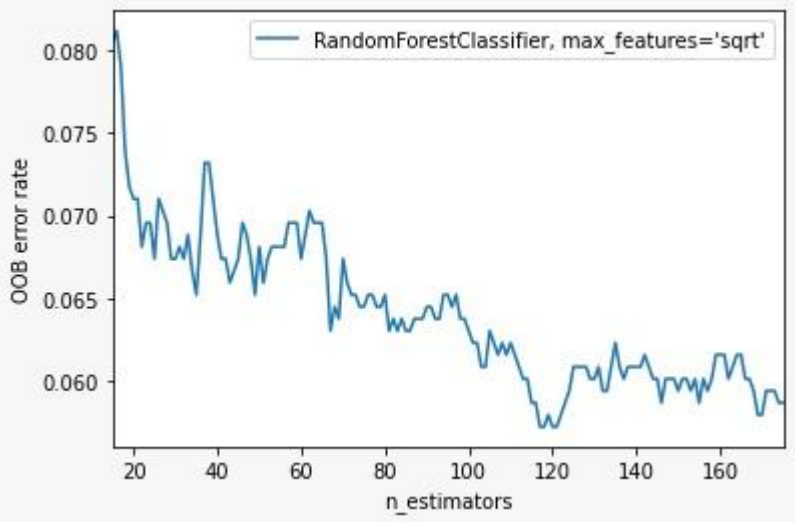
c)

What actually OOB is?

In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows:

Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree.Put each case left out in the construction of the kth tree down the kth tree to get a classification. In this way, a test set classification is obtained for each case in about one-third of the trees. At the end of the run, take j to be the class that got most of the votes every time case n was oob. The proportion of times that j is not equal to the true class of n averaged over all cases is the oob error estimate.

In Our case as n_estimators increases the OOB error rate slowly decreases and gradually tends to 0 for higher and higher values of n_estimators



5)**Pre-processing of data**

Initially I stored the training data and testing data into 2 variables ‘dataset’ and ‘dataset1’ and later I applied the filters so as to remove the unnecessary attributes.

Later I changed the target variable(here loan_status) to 0 if ‘fully paid’ and 1 if ‘charged off’ and removed all intermediate stages in both the data sets and also removed the attributes which always had the unique values in all data points from both test and train data sets.

After this we get the data shape as 82 attributes.

And now even if 85% of the data have the same values then also I removed those attributes as we know that it will be less dependent on these types of attributes which have mostly always the same value.

After this filter we finally have 46 attributes in both the data sets.

After observing few other attributes which I felt are of less priority

del_col_names = ["delinq_2yrs", "last_pymnt_d", "chargeoff_within_12_mths","delinq_amnt","emp_title", "term", "emp_title", "pymnt_plan","purpose","title", "zip_code",

"verification_status", "dti","earliest_cr_line", "initial_list_status", "out_prncp",
"pymnt_plan","delinq_amnt",
"chargeoff_within_12_mths", "total_rec_late_fee",
"out_prncp_inv", "issue_d"] #deleting some more columns

After this we end up with 28 attributes remaining

Later I have removed 3 other attributes and finally 25 attributes are used for fitting which are
features =
['loan_amnt','funded_amnt','funded_amnt_inv','int_rate','installment','grade','emp_length','home_ownership','annual_inc','loan_status','inq_last_6mths','open_acc','pub_rec','revol_bal','revol_util','total_acc','total_pymnt','total_pymnt_inv','total_rec_prncp','total_rec_int','recoveries','collection_recovery_fee','last_pymnt_amnt','pub_rec_bankruptcies','tax_liens']

I have also removed all the rows which have Nan values for the attributes. And finally ended up in 23815 rows for training data and 12985 rows for test data.

And also removed the % from interest_rate and revol_util so as to set for data fitting.

Finally extracting all the values of data attributes except ‘loan_status’ attribute data into X variable and storing data values of ‘loan_status’ attribute into Y variable.

Here I have also took the validation data set where 20% used for validation using the train_test_split function.

Now we use **GradientBoostingClassifier**(n_estimators=500,learning_rate=0.01,max_depth=3,subsample=0.5,validation_fraction=0.1,n_iter_no_change=20,max_features='log2',verbose=1) and fit using X_train and Y_train

And using roc_auc_score and predict_proba function we will find the accuracies for all the 3 (train,validation and test) sets which are as follows
Train :0.9981
Valid:0.9959
Test:0.9958

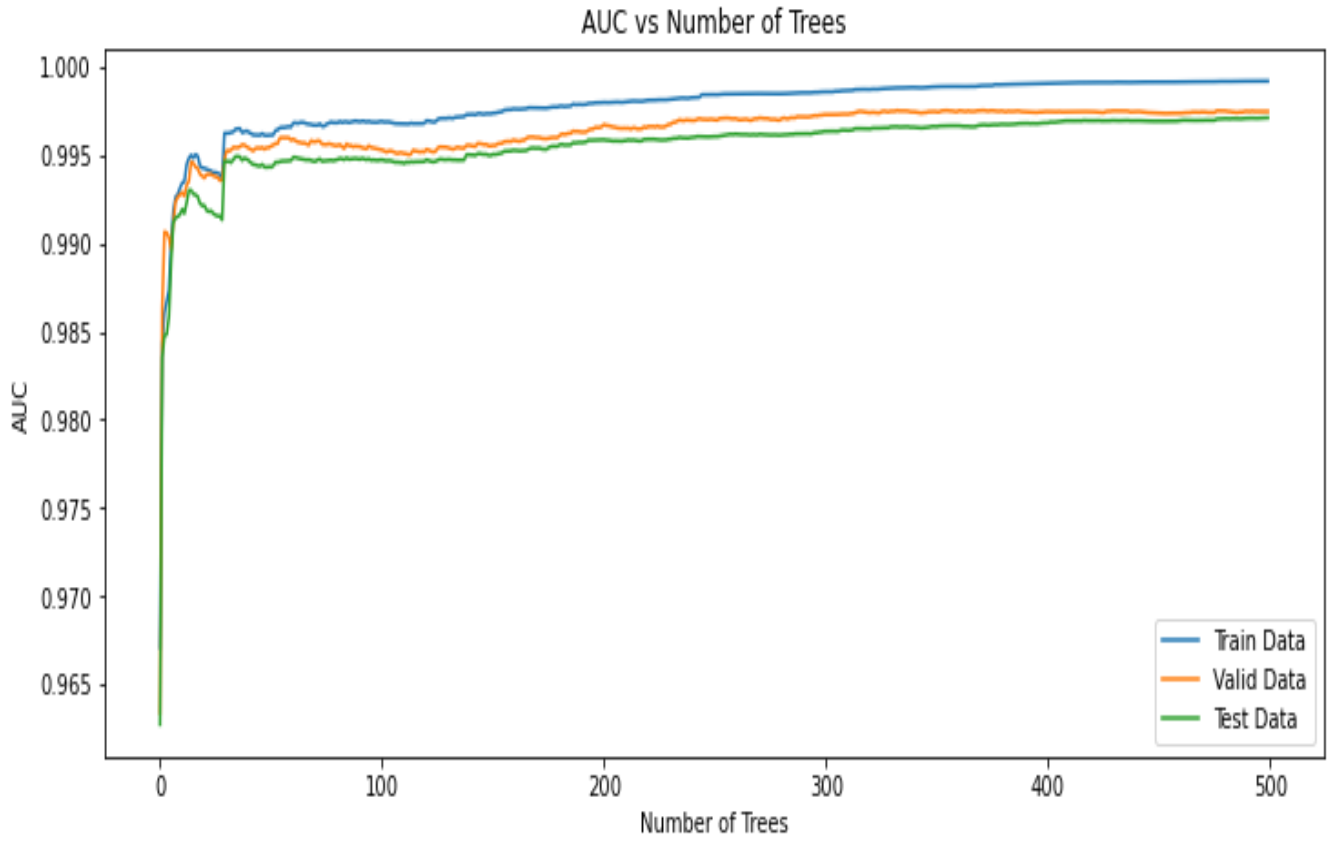
And the hyperparameters list based on the importance is

	Attribute	Importance
	recoveries	0.345874
	collection_recovery_fee	0.279556
	total_rec_prncp	0.124930
	funded_amnt	0.051760
	last_pymnt_amnt	0.044263
	total_pymnt	0.043996

total_pymnt_inv	0.034208
loan_amnt	0.022218
funded_amnt_inv	0.015094
installment	0.011675
total_rec_int	0.010984
int_rate	0.008355
grade	0.003635
revol_bal	0.001296
revol_util	0.000856
annual_inc	0.000405
inq_last_6mths	0.000274
total_acc	0.000264
open_acc	0.000129
pub_rec_bankruptcies	0.000083
home_ownership	0.000055
emp_length	0.000049
pub_rec	0.000043
tax_liens	0.000000

Now If we remove the top 3 attributes which are recoveries,collection_recovery_fee and total_rec_prncp the accuracies changes to

Train :0.9826
Valid:0.9775
Test:0.9790
b)As we know that as no_of_trees increases accuracy increases and so as to have a little clear vision I have plotted the accuracy with no_of_trees for all 3 data which are train,validation and test datasets.



As we can see clearly as the number of trees increases the accuracy increases but at a slower pace.

c)
Now we have classified the same data set files using DecisionTreeClassifier(criterion = "entropy",random_state = 10,max_depth=4).fit(X_train,Y_train)

And accuracy we obtained from this is 0.9862918752406623
And if we removed the 3 top most attributes from the attributes priority list from all the 3 data files and again fitted the model using the decision tree we get accuracy as 0.9126684636118598 which is less than the actual accuracy which we got earlier.