Advanced Topics in Machine Learning

Programming Assignment 1

Task

Fit a regression tree for CARSEATS dataset (provided) and predict the sales value. Perform cross validation to compute the optimal tree complexity and then perform pruning accordingly. Use Bagging and Random forests approach to dataset and analyze the data

Software and Hardware details:

Language used : Python 2.7

OS : Windows 8

IDE: Pycharm

Dataset: https://github.com/selva86/datasets/blob/master/Carseats.csv

1. Preprocessing

The categorical variables were in string format. The other data were in integer/float format. So in Urban and US features , "Yes" was converted to 1 and "No" was converted to 0.Similary for SheveLoc "Bad" was converted to 0, "Medium" was converted to 1 and "Good" was converted to 2.

2. Splitting dataset

The dataset consists of 400 rows. 80% were used for training and 20 % for testing. To be precise **320 rows** were used for training and **80 rows** were used for testing.

3. Fitting the regression tree

The tree was built using recursive binary splitting technique. At any point, for splitting a top-down greedy approach was used to select the best feature to be split and the best split of it. The best feature ,spilt point was generated by randomly using 75 split points for each feature and calculating residual sum square error. The one with the minimum error is the best feature at that point. The tree continues to split until a **minimum no of leaf nodes is reached say 20**.The tree is stored as a class structure which has attributes like root to denote the list of rows the node contains, left and right child,mean value of sales feature in the list and flag value to denote if it is a leaf or not.

sample1

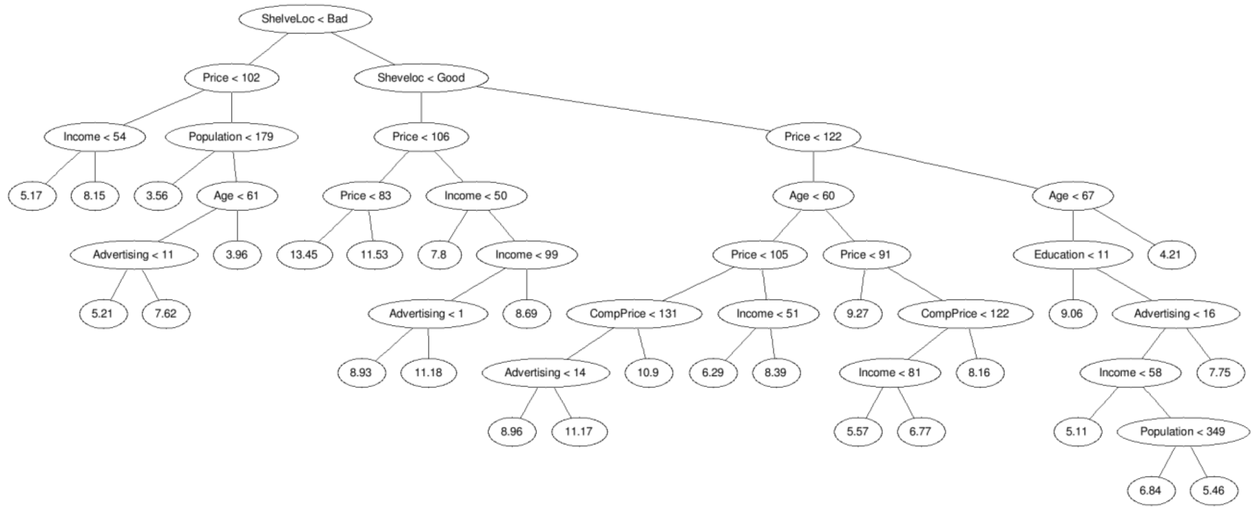
```
Converting into float

ShelveLoc < 1 Price < 102 Income < 54 5.17
Income >= 54 8.15
Price >= 102 Population < 179 3.56
Population >= 179 Age < 61 Advertising < 11 5.21
Advertising >= 11 7.62
Age >= 61 3.96
ShelveLoc >= 1 ShelveLoc < 2 Price < 106 Price < 83 13.45
Price >= 83 11.53
Price >= 106 Income < 50 7.8
Income >= 50 Income < 99 Advertising < 1 8.93
Advertising >= 1 11.18
Income >= 99 8.69
ShelveLoc >= 2 Price < 122 Age < 60 Price < 106 CompPrice < 131 Advertising < 14 8.96
Advertising >= 14 11.18
CompPrice >= 131 10.9
Price >= 106 Income < 51 6.29
Income >= 51 8.39
Age >= 60 Price < 91 9.27
Price >= 91 CompPrice < 122 Income < 81 5.57
Income >= 81 6.77
CompPrice >= 122 8.15
Price >= 122 Age < 67 Education < 11 9.06
Education >= 11 Advertising < 16 Income < 58 5.11
Income >= 58 Population < 349 6.84
Population >= 349 5.46
Advertising >= 16 7.75
Age >= 67 4.21
4.82068235245
Total_splits 26
Important Feautures:
ShelveLoc Price ShelveLoc
```

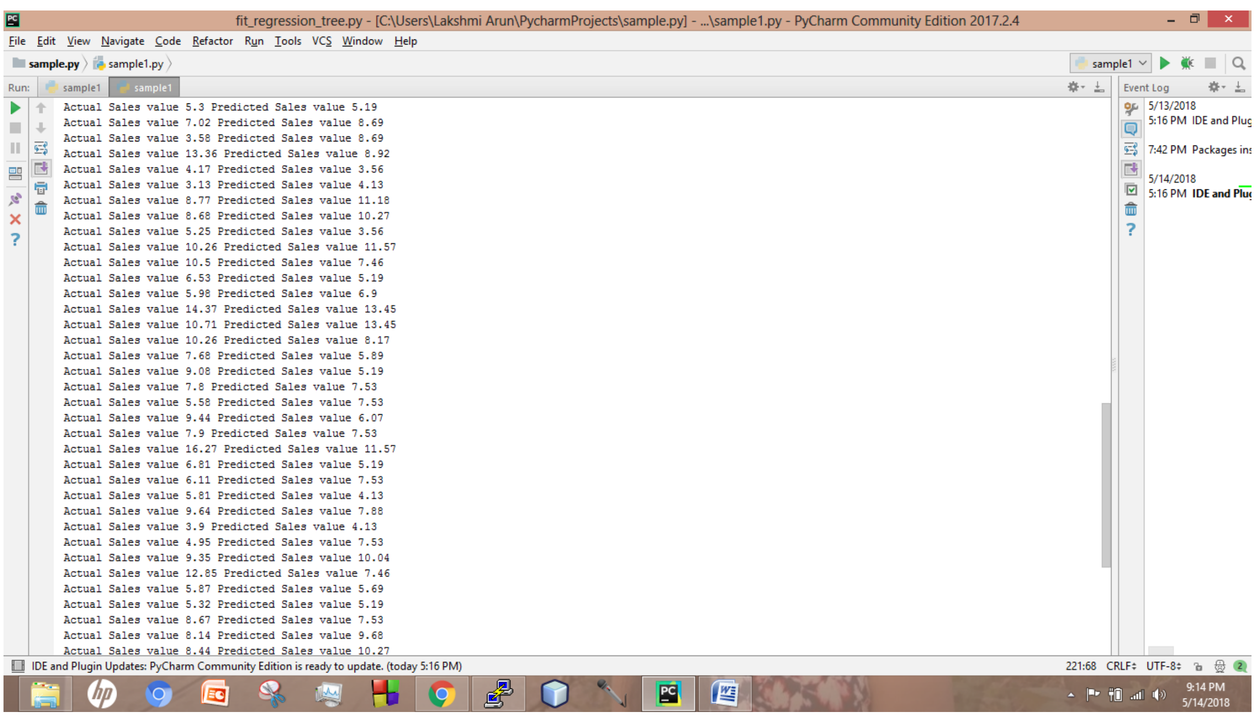
IDE and Plugin Updates: PyCharm Community Edition is ready to update. (today 5:16 PM)

The tree generated and leaf nodes mentions the mean value.

Generated Tree :

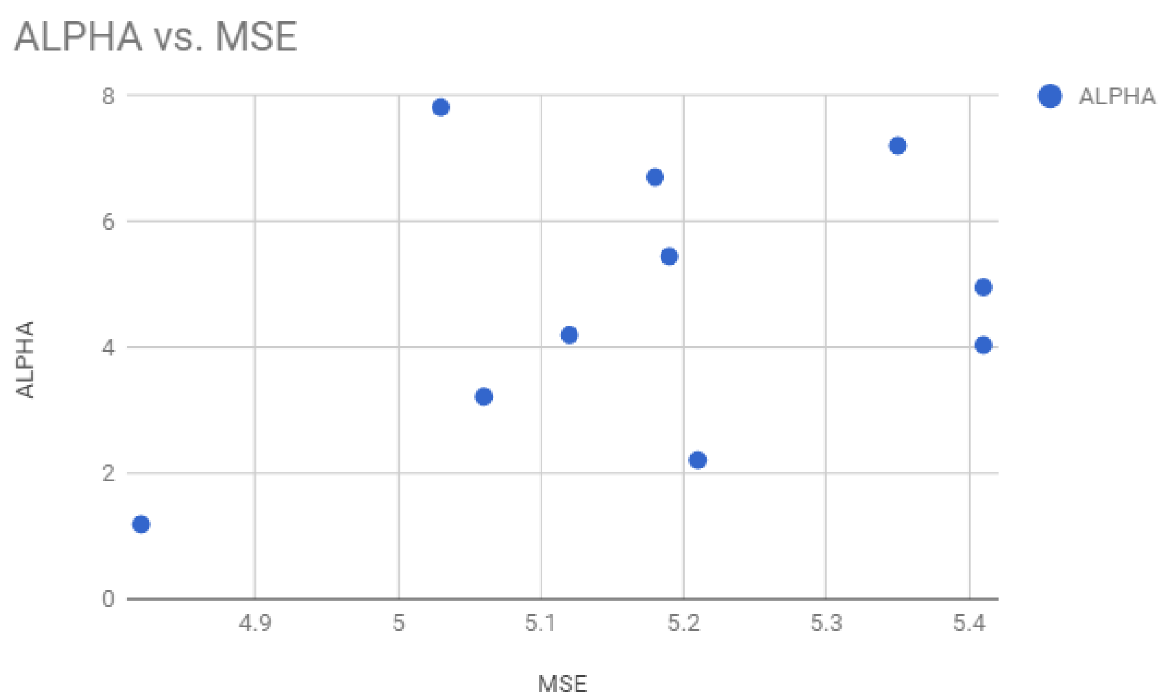


The MSE obtained is 4.82. The sales value predicted is the mean of the list that of the region where it belongs to according to the regression tree, below is the snapshot of the predicted sales value. Some values are close some are far apart



4.Cross Validation : Cross validation is performed on k folds, where **k say is 10** . At each time training is done on k -1 folds and testing is done on kth fold. The **size of each fold** is len/k which is **40** with our dataset. For each training set of dataset we use different m alpha randomly (uniformly random)generated within range 1,10 and generate m trees and test error is calculated as a function of alpha.

```
Cross validating
MSE  ALHA  Total_splits
5.35 7.21 9.5
5.41 4.04 12.6
5.18 6.71 9.9
5.21 2.21 16.5
5.12 4.2 12.3
5.06 3.22 14.0
4.82 1.19 22.5
5.41 4.96 11.4
5.03 7.82 8.8
5.19 5.45 10.7
```



5 Pruning

Pruning is done by cost complexity pruning. That is when we find the residual error we add with a penalizer (alpha*No leaf nodes till region m)

Using cross validation we estimated the alpha value to be 7.82 . If we prune the tree accordingly, we get

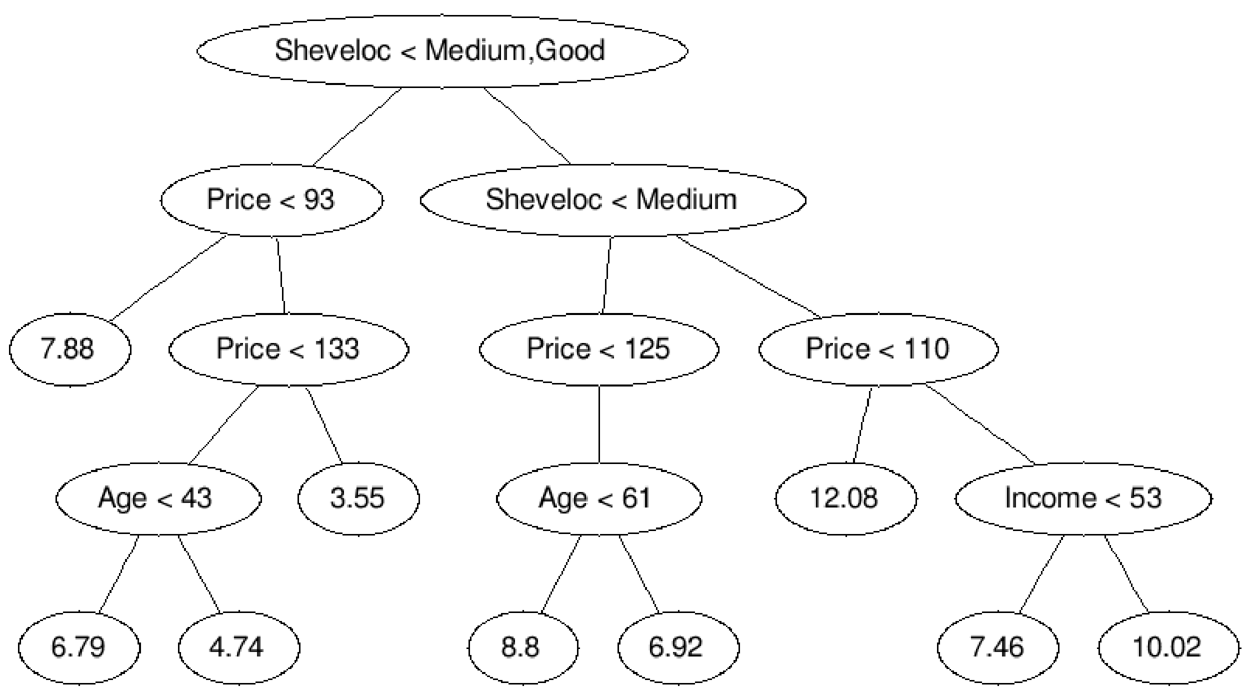
```
C:\Python27\python.exe "C:/Users/Lakshmi Arun/PycharmProjects/sample.py/sample1.py"
Loading Data

Converting into float

ShelveLoc < 1 Price < 93 7.88
Price >= 93 Price < 133 Age < 43 6.79
Age >= 43 4.74
Price >= 133 3.55
ShelveLoc >= 1 ShelveLoc < 2 Price < 110 12.08
Price >= 110 Income < 53 7.46
Income >= 53 10.02
ShelveLoc >= 2 Price < 125 Age < 61 8.8
Age >= 61 6.92
Price >= 125 5.79
4.98012660539
Total_splits 9
Important Feautures:
ShelveLoc
Price
ShelveLoc

Process finished with exit code 0
```

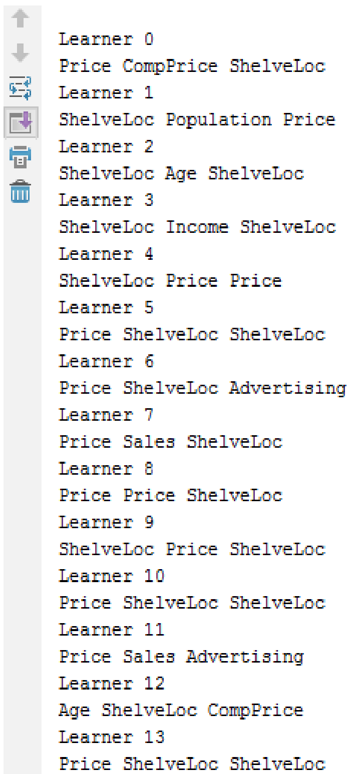
Tree Generated by pruning



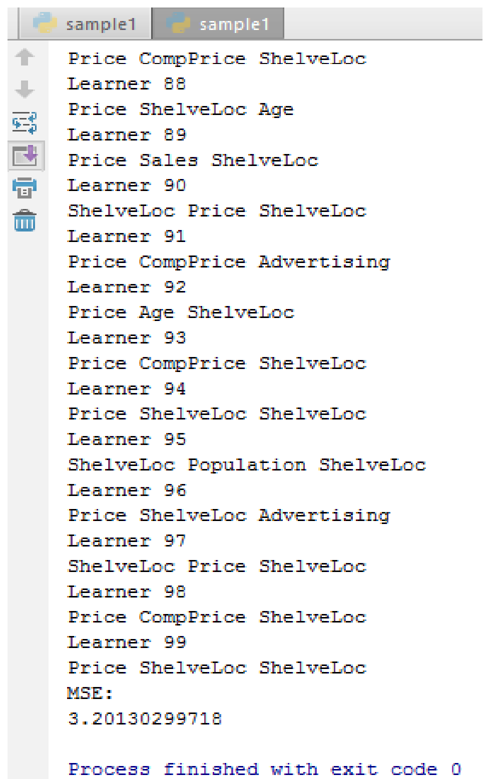
The MSE is around 4.98 . Previously our MSE was 4.7 without pruning. There is a slight increase in MSE but the complexity has been greatly reduced

6. Bagging :

Bagging is aggregating the predications of many weak learners. In my experiment my number of bag learners were 100. My sales value was predicted by aggregating (finding mean) of all the predicted values from 100 decision trees. Bagging works on sub-sampling of train data with replacement. I used the ration of 0.5 i.e each of bag learners will have (0.5*320 = 160) training data and tested on the common test data of 80 rows. The output lists the for each learner what are the important features and finally the MSE of the bagged trees. **Important Features are obtained by printing the first 3 features that are responsible for top-most split .** It can be noted that **Price and ShelveLoc** are the most important features.Next to that comes Comprice Age and Advertising. The MSE is obtained is 3.2 . That is good improvement which means bagging reduced the error rate.



```
Learner 0
Price CompPrice ShelveLoc
Learner 1
ShelveLoc Population Price
Learner 2
ShelveLoc Age ShelveLoc
Learner 3
ShelveLoc Income ShelveLoc
Learner 4
ShelveLoc Price Price
Learner 5
Price ShelveLoc ShelveLoc
Learner 6
Price ShelveLoc Advertising
Learner 7
Price Sales ShelveLoc
Learner 8
Price Price ShelveLoc
Learner 9
ShelveLoc Price ShelveLoc
Learner 10
Price ShelveLoc ShelveLoc
Learner 11
Price Sales Advertising
Learner 12
Age ShelveLoc CompPrice
Learner 13
Price ShelveLoc ShelveLoc
```



```
sample1 sample1
Price CompPrice ShelveLoc
Learner 88
Price ShelveLoc Age
Learner 89
Price Sales ShelveLoc
Learner 90
ShelveLoc Price ShelveLoc
Learner 91
Price CompPrice Advertising
Learner 92
Price Age ShelveLoc
Learner 93
Price CompPrice ShelveLoc
Learner 94
Price ShelveLoc ShelveLoc
Learner 95
ShelveLoc Population ShelveLoc
Learner 96
Price ShelveLoc Advertising
Learner 97
ShelveLoc Price ShelveLoc
Learner 98
Price CompPrice ShelveLoc
Learner 99
Price ShelveLoc ShelveLoc
MSE:
3.20130299718

Process finished with exit code 0
```

7.Random Forests:

Random forests is a technique which reduces the number of features considered for optimal step at a node for each bagging learner. Here, the number of features considered has been set to sqrt (num of total features) which is 3.13 in our dataset , on rounding it we 4. Reducing the features ensures few features are selected at random, so all features are given equal probability to considered for top-splits. The MSE obtained is **3.47** which is a little higher than normal bagging. The output is formated same like bagging,it lists the learners and important features for each learner and finally the MSE. Like bagging we have used

100 random learners .Again the important features are **SheveLoc and Price.** If we decrease the m(no of feautures) value to be very less then our MSE increases more.

```
C:\Python27\python.exe "C:/Users/Lak
Loading Data

Converting into float

Learner 0
Price CompPrice ShelveLoc
Learner 1
ShelveLoc Advertising Price
Learner 2
Price ShelveLoc CompPrice
Learner 3
CompPrice ShelveLoc Sales
Learner 4
ShelveLoc Population CompPrice
Learner 5
Age Price Income
Learner 6
Income Advertising Age
Learner 7
Age CompPrice ShelveLoc
Learner 8
Advertising Price Age
Learner 9
Advertising Price Population
Learner 10
Income ShelveLoc ShelveLoc
Learner 11
```

```
ShelveLoc Income ShelveLoc
Learner 89
Age Price ShelveLoc
Learner 90
Price Age ShelveLoc
Learner 91
CompPrice Price Age
Learner 92
CompPrice Income Age
Learner 93
ShelveLoc Price Price
Learner 94
Population Sales Education
Learner 95
Advertising ShelveLoc Income
Learner 96
Advertising Income ShelveLoc
Learner 97
Age ShelveLoc Price
Learner 98
Price Income Population
Learner 99
ShelveLoc Age Advertising
MSE:
3.47895912353

Process finished with exit code 0
```

Key - Observations

Model	MSE
Regression tree	4.82
Pruning	5.03
Bagging	3.13
Random Forests	3.47

The least MSE is obtained by bagging approach. Pruning increases the MSE a little and so does random forests when compared to bagging. By cross-validation the optimal tree complexity is found to be around 8.The important features for the car dataset are Shevloc and Price.

*Note : The trees and graphs were generated using graphvitz and excel respectively.