Bagging

September 25, 2018

Name : Shushil Kumar Ravishankar
Reg : 16BCE1259
lab : L39-L40
Topic: Bagging algorithms(Random Forest)

Bootstrap Aggregation or bagging involves taking multiple samples from your training dataset (with replacement) and training a model for each sample.
The final output prediction is averaged across the predictions of all of the sub-models.
The three bagging models covered in this section are as follows:
1)Bagged Decision Trees
2)Random Forest
3)Extra Trees
Random Forest :
1)Random forest is an extension of bagged decision trees.
2)Samples of the training dataset are taken with replacement, but the trees are constructed in a way that reduces the correlation between individual classifiers.
3)Specifically, rather than greedily choosing the best split point in the construction of the tree, only a random subset of features are considered for each split.
4)You can construct a Random Forest model for classification using the RandomForestClassifier class.

```
In [1]: import pandas as pd

In [2]: from sklearn import model_selection

In [3]: from sklearn.ensemble import RandomForestClassifier

        from sklearn.preprocessing import LabelEncoder

In [4]: data=pd.read_csv('clean_bmart.csv',sep=',')
        data.head()
```

Out[4]:

	Unnamed: 0	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	\
0	0	FDA15	9.30	Low Fat	0.016047	
1	1	DRC01	5.92	Regular	0.019278	
2	2	FDN15	17.50	Low Fat	0.016760	
3	3	FDX07	19.20	Regular	0.000000	
4	4	NCD19	8.93	Low Fat	0.000000	

```
                Item_Type  Item_MRP Outlet_Identifier  \
0                   Dairy  249.8092            OUT049
1             Soft Drinks   48.2692            OUT018
2                    Meat  141.6180            OUT049
3  Fruits and Vegetables  182.0950            OUT010
4               Household   53.8614            OUT013

   Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                       1999      Medium               Tier 1
1                       2009      Medium               Tier 3
2                       1999      Medium               Tier 1
3                       1998      Medium               Tier 3
4                       1987        High               Tier 3

         Outlet_Type  Item_Outlet_Sales
0  Supermarket Type1          3735.1380
1  Supermarket Type2           443.4228
2  Supermarket Type1          2097.2700
3      Grocery Store           732.3800
4  Supermarket Type1           994.7052
```

In [8]:
```
X=data.loc[(data['Outlet_Location_Type']=='Tier 1')|(data['Outlet_Location_Type']=='Ti
x=X.values[:,:]
y=X.values[:,10]
ley=LabelEncoder()
ley.fit(y)
y=ley.transform(y)
for i in [1,3,5,7,9,11]:
    en=LabelEncoder()
    en.fit(X.values[:,i])
    x[:,i]=en.transform(x[:,i])

x=x[:,[1,2,3,4,5,6,7,8,9,11,12]]
print (x)
print(y)
```

```
[[156 9.3 0 ... 0 1 3735.138]
 [659 17.5 0 ... 0 1 2097.27]
 [438 16.2 1 ... 0 1 1076.5986]
 ...
 [890 8.38 1 ... 0 1 549.285]
 [1348 10.6 0 ... 1 1 1193.1136]
 [50 14.8 0 ... 1 1 765.67]]
[0 0 1 ... 1 1 0]
```

In [17]:
```
seed = 7
num_trees = 100
max_features = 5
```

```
In [18]: kfold = model_selection.KFold(n_splits=10, random_state=seed)
         model = RandomForestClassifier(n_estimators=num_trees, max_features=max_features)
         results = model_selection.cross_val_score(model, x, y, cv=kfold)
         print(results.mean())
```

```
1.0
```

the mean estimate of classification accuracy= 1.0 this is classification with 100 trees and split points chosen from a random selection of 5 features using Random Forest Classification