ENSEMBLE

Table of Contents

1. Introduction to Ensemble Learning
2. Basic Ensemble Techniques  
   2.1 Max Voting  
   2.2 Averaging  
   2.3 Weighted Average
3. Advanced Ensemble Techniques  
   3.1 Stacking  
   3.2 Blending  
   3.3 Bagging  
   3.4 Boosting
4. Algorithms based on Bagging and Boosting  
   4.1 Bagging meta-estimator  
   4.2 Random Forest  
   4.3 AdaBoost  
   4.4 GBM  
   4.5 XGB  
   4.6 Light GBM  
   4.7 CatBoost

 The ensemble is a method used in the machine learning algorithm. In this method, multiple models or ‘weak learners’ are trained to rectify the same problem and integrated to gain desired results. Weak models combined rightly give accurate models.

 Simple Ensemble Techniques

In this section, we will look at a few simple but powerful techniques, namely:

1. Max Voting
2. Averaging
3. Weighted Averaging

### **Max Voting**

The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a ‘vote’. The predictions which we get from the majority of the models are used as the final prediction.

For example, when you asked 5 of your colleagues to rate your movie (out of 5); we’ll assume three of them rated it as 4 while two of them gave it a 5. Since the majority gave a rating of 4, the final rating will be taken as 4. **You can consider this as taking the mode of all the predictions.**

The result of max voting would be something like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| 5 | 4 | 5 | 4 | 4 | 4 |

**Sample Code:**

Here x\_train consists of independent variables in training data, y\_train is the target variable for training data. The validation set is x\_test (independent variables) and y\_test (target variable) .

### **Averaging**

Similar to the max voting technique, multiple predictions are made for each data point in averaging. In this method, we take an average of predictions from all the models and use it to make the final prediction. Averaging can be used for making predictions in regression problems or while calculating probabilities for classification problems.

For example, in the below case, the averaging method would take the average of all the values.

i.e. (5+4+5+4+4)/5 = 4.4

model1 = tree.DecisionTreeClassifier()

model2 = KNeighborsClassifier()

model3= LogisticRegression()

model1.fit(x\_train,y\_train)

model2.fit(x\_train,y\_train)

model3.fit(x\_train,y\_train)

pred1=model1.predict\_proba(x\_test)

pred2=model2.predict\_proba(x\_test)

pred3=model3.predict\_proba(x\_test)

finalpred=(pred1+pred2+pred3)/3

### **Weighted Average**

This is an extension of the averaging method. All models are assigned different weights defining the importance of each model for prediction. For instance, if two of your colleagues are critics, while others have no prior experience in this field, then the answers by these two friends are given more importance as compared to the other people.

The result is calculated as [(5\*0.23) + (4\*0.23) + (5\*0.18) + (4\*0.18) + (4\*0.18)] = 4.41.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| weight | 0.23 | 0.23 | 0.18 | 0.18 | 0.18 |  |
| rating | 5 | 4 | 5 | 4 | 4 | 4.41 |

model1 = tree.DecisionTreeClassifier()

model2 = KNeighborsClassifier()

model3= LogisticRegression()

model1.fit(x\_train,y\_train)

model2.fit(x\_train,y\_train)

model3.fit(x\_train,y\_train)

pred1=model1.predict\_proba(x\_test)

pred2=model2.predict\_proba(x\_test)

pred3=model3.predict\_proba(x\_test)

finalpred=(pred1\*0.3+pred2\*0.3+pred3\*0.4)

## Advanced Ensemble techniques

The main purpose of using an ensemble model is to group a set of weak learners and form a strong learner.

## Bagging

Bagging is an acronym for ‘Bootstrap Aggregation’ and is used to decrease the variance in the prediction model.

## Bagging and Boosting: Differences

Bagging is a method of merging the same type of predictions. Boosting is a method of merging different types of predictions.

Bagging decreases variance, not bias, and solves over-fitting issues in a model. Boosting decreases bias, not variance.

In Bagging, each model receives an equal weight. In Boosting, models are weighed based on their performance.

Models are built independently in Bagging. New models are affected by a previously built model’s performance in Boosting.

Bagging is usually applied where the classifier is unstable and has a high variance. Boosting is usually applied where the classifier is stable and simple and has high bias.

## How are the main differences bagging and boosting?

Bagging is a technique for reducing prediction variance by producing additional data for training from a dataset by combining repetitions with combinations to create multi-sets of the original data. Boosting is an iterative strategy for adjusting an observation's weight based on the previous classification. It attempts to increase the weight of an observation if it was erroneously categorized. Boosting creates good predictive models in general.