**Ensemble Learning Techniques to Improve Machine Learning**

**December 13, 2020**

* Topics:
* [Machine Learning](https://www.section.io/engineering-education/topic/machine-learning/)

AddThis Sharing Buttons

Share to Facebook

Share to TwitterShare to LinkedIn

We build machine learning models to provide the best possible predictions to a given problem. However, a single model may not make the best predictions and may be subject to errors such as variance and bias.

To reduce these errors and improve the predictions, we may combine multiple models into a single model. This is what is known as ensemble learning. We will explore ensemble learning techniques that can be used to improve the machine learning process.

**Contents**

* Introduction to ensemble learning.
* Ensemble learning benefits.
* Ensemble learning techniques.

**Prerequisites**

An understanding of machine learning is recommended. This [post](https://www.section.io/engineering-education/supervised-learning-algorithms/) can provide a good introduction or refresher to machine learning.

**Ensemble learning**

Ensemble learning refers to a method where many base models are combined to carry out the same task. These base models are usually referred to as weak learners.

Ensemble learning works on the principle that a weak learner predicts poorly when alone. But when combined with other weak learners, they create a strong learner. The strong learner performs much better than the lesser learners. We shall expound on weak learners later on.

**Why use ensemble learning?**

Compared to using a single model, it’s beneficial to use an ensemble for a couple of reasons:

**Performance**: As described in the previous section, the outcome of ensemble learning is a strong learner — the strong learner results from weak learners. As a result, the predictive capability of models is improved. Compared to a single model, better performance is achieved.

**Reduction of errors**: Prediction errors of machine learning models can be described through bias and variance. Bias refers to the difference between prediction and the actual outcome. Variance can be described as the responsiveness of a model to tiny fluctuations in the training set. It’s preferable to have a model with low bias and variance, but practically it’s a challenge to achieve.

Ensemble methods offer a way to lower prediction variance. For instance, through averaging of the performance of various models, the spread of performance is contained. The bagging technique assists in reducing the variance.

Boosting and stacking produce less biased models. We shall dive into these techniques in the next section. Plus, here’s a read on the relationship between [ensemble learning and bias and variance](https://medium.com/ml-research-lab/ensemble-learning-relation-with-bias-and-variance-431cdc0a3fc9).

The next section is split into two sections: basic and advanced ensemble learning techniques. We will cover three of each.

**Advanced ensemble learning techniques**

**Bagging**

The name bagging is derived from the words bootstrapping and aggregation. It combines the two to form one model. Bagging is a technique that works with the idea of uniting many models to produce a generalized result. However, a challenge is realized when training the models. When you train all the models on the same data then combine them, they are most likely to give very similar results.

To solve such a challenge, a sampling technique we may use is bootstrapping. It may be used to generate subsets of observations from the original dataset. The sampling is done with replacement. Sampling with replacement refers to when an example is randomly selected from a population then returned to the population. The subsets are of the same size as the original dataset. These subsets may be referred to as bags.

Here are the steps to the bagging process:

* Bags are generated through bootstrapping. As mentioned previously, sampling with replacement is done.
* After generating bags, we now carry out bootstrap aggregating (bagging). This method utilizes the bags to understand the dataset’s data distribution.
* What follows is the creation of base models on each of the bootstrapped subsets. It’s worth noting that the models run independently and in parallel to each other.
* The final step involves determining the final predictions by combining the results from all the models.

The image below will aid in the understanding of the bagging process.

*Bagging process*

[Source](https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/)

**Boosting**

Boosting involves the use of a collection of algorithms to convert weak learners into strong learners. A weak learner is one that labels examples marginally better than random guessing. An example is a basic decision tree. Weighted data is used on these weak learners. The weighting is specific for misclassified data. We shall revisit this in the next paragraph.

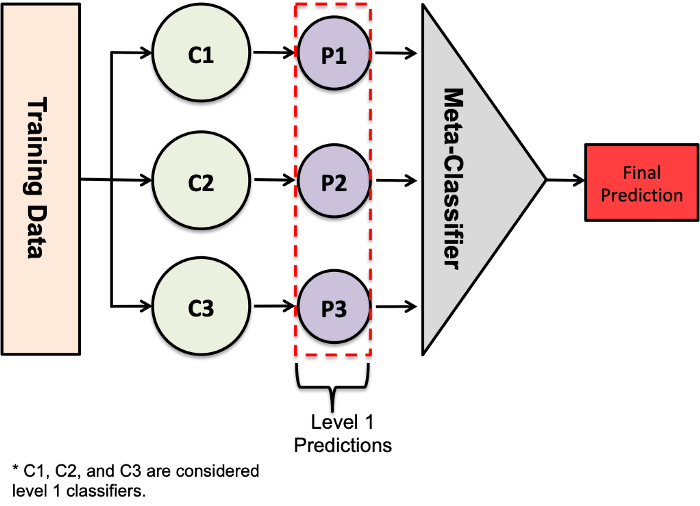
Boosting is sequentially done with every ensuing model aiming to minimize the error of the model before it. Let’s go through the process of boosting:

* Similar to bagging, a subset is produced from the original set.
* All datapoints start with weights of equal value.
* A base learner is then fitted to the generated subset.
* The learner is then used to generalize on the complete dataset.
* After making predictions, the outcome is compared with the actual values to calculate the errors.
* Once the errors are determined, the misclassified observations are assigned higher weights.
* The next step involves introducing a new model to make predictions on the same dataset. The goal of the new model is to remedy the errors of the model before it.
* The process is iterated with a new model that aims to iron out the previous model’s inefficiencies.
* This sequence continues until the strongest learner is realized. The final model becomes the weighted mean of all the previous models.

**Stacking**

We can describe stacking as a technique that uses many classifiers’ predictions as new features for training a meta-classifier. The many classifiers can be referred to as level one classifiers. We can also simply define a meta-classifier as a classifier that takes in the predictions of other classifiers.

Let’s use an image to explain this technique a bit more intuitively.



*The framework of a stacking classifier*

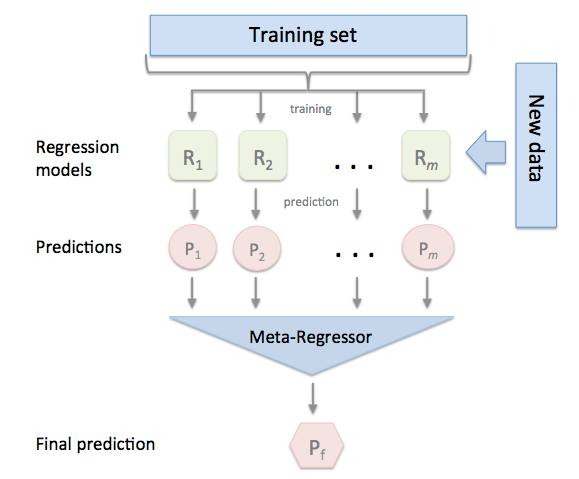
[Source](https://towardsdatascience.com/stacking-classifiers-for-higher-predictive-performance-566f963e4840)

From the image above, we note that we have three level-one classifiers (C1. C2, and C3). The classifiers are independently trained. After training, the classifiers make predictions. The predictions made are then used for the training of the meta-classifier. When stacking classifiers, it is advisable to have level one predictions from a subset of the training set that was not used to train the level one classifier.

The purpose of this is to avoid information leakage from what we’re attempting to predict (target) into the training set. To achieve this, we split the training set into two. The first half of the training set should be used to train the level-one classifiers. After the classifiers have been trained, we use them on the other half of the training data to make predictions.

Finally, we use the resultant predictions to train the meta-classifier. It’s worth noting that stacking can be used with regression models too. Like the approach with classification models, stacking puts together predictions of several regression models through a meta-regressor.

The image below gives a better description of the process.



*Stacking in regression*

[Source](https://rasbt.github.io/mlxtend/user_guide/regressor/StackingRegressor/#:~:text=Stacking%20regression%20is%20an%20ensemble%20learning%20technique%20to,of%20the%20individual%20regression%20models%20in%20the%20ensemble.)

Here is a technical [article on stacking and its implementation](https://towardsdatascience.com/stacking-classifiers-for-higher-predictive-performance-566f963e4840).

**Basic ensemble learning techniques**

**Max voting classifier**

This technique is similar to averaging (which shall be discussed later) but is ideal for classification problems. Its explanation is simple.

Several models make predictions, that are referred to as votes; each prediction counts as a vote. As is often the case with voting, a decision is often in favor of most votes. The same applies here. The prediction made by the majority of classifiers becomes the final prediction.

For instance, we attempt to predict car prices. The models involved end up giving a set of prices (in dollars) like 6000, 5000, 8000, 5000, 6000, 5000, 7000, 5000, and 7000. We note that most of the models predict $5000. Using max voting, 5000 becomes the final prediction.

**Averaging**

This is the easiest ensemble technique to define as it simply takes the average of all predictions made by the models. The average of the predictions is used to make the final prediction.

This technique can be used for both classification and regression problems. Consider the same scenario as in the previous section, where we want to predict car prices. Our models end up predicting $ 6000, 7000, 5000, 6000.

Since we’re averaging, the final prediction is:

6000+7000+5000+60004=6000

**Weighted averaging**

This technique is a variation of the averaging method. The difference is that, where averaging gives the models equal importance, weighted average gives greater importance to a model with greater predictive power. This importance is represented in terms of weights.

These weights may be represented as decimals whose total should be equal to 1. Here’s an example to make it clearer. Using the same example as the two previous sections, suppose we have four models to predict car prices. Their predictions are $ 5000, 8000, 7000 and 5000.

The respective weights would be 0.25, 0.1, 0.5 and 0.15.

The final prediction will be given by:

∑0.25∗50000.1∗80000.5∗70000.15∗5000=6300

**Wrapping up**

Ensemble methods allow us to average the performance of many models to generate one final model. This final model offers the best performance compared to individual models in the ensemble. We have discussed a few advanced ensemble techniques as well as a few simple ones. I hope it has been insightful.

Until next time, good luck!

**References and Further Reading**

**Which of the following will be the predicted category for these observations if you apply probability threshold greater than or equals to 0.5 for category “1” or less than 0.5 for category “0”?**

A.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| .70 | .80 | .75 | 1 |
| .50 | .64 | .80 | 1 |
| .30 | .20 | .35 | 0 |
| .49 | .51 | .50 | 0 |
| .60 | .80 | .60 | 1 |

B.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| .70 | .80 | .75 | 1 |
| .50 | .64 | .80 | 1 |
| .30 | .20 | .35 | 0 |
| .49 | .51 | .50 | 1 |
| .60 | .80 | .60 | 1 |

C.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| .70 | .80 | .75 | 1 |
| .50 | .64 | .80 | 1 |
| .30 | .20 | .35 | 1 |
| .49 | .51 | .50 | 0 |
| .60 | .80 | .60 | 0 |

D. None of these

**Solution: (B)**

Take the weighted average of the predictions of each model for each observation then apply threshold 0.5 you will get B answer.

For example, in first observation of models (M1,M2 and M3) the outputs are 0.70,0.80,0.75 if you take the weighted average of these three predictions, you will get 0.745 output (0.70 \* 0.4 + 0.80 \*0.3 + 0.75\* 0.3) which is above 0.5 that means this observation will belong to class 1.

**Suppose in binary classification problem, you have given the following predictions of three models (M1, M2, M3) for five observations of test data set.**

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| 1 | 1 | 0 |  |
| 0 | 1 | 0 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 1 |  |

**Which of the following will be the output ensemble model if we are using majority voting method?**

A.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

B.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

C.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

D. None of these

**Solution: (B)**

Take the majority voting for the predictions of each model for each observation.

For example for a first observation of models(M1,M2 and M3) the outputs are  1,1,0 if you take the majority voting of these three models predictions you will get 2 votes for class 1 that means this observation will belong to class 1.

**Q When using the weighted voting method, which of the following will be the output of an ensemble model?**

Hint: Count the vote of M1,  M2, and M3 as 2.5 times, 6.5 times and 3.5 times respectively.

A.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

B.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Output |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

C.

|  |  |  |  |
| --- | --- | --- | --- |
| M1 | M2 | M3 | Ouput |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |

D. None of these

**Solution: (C)**

Follow the steps in question number 20,21 and 22.