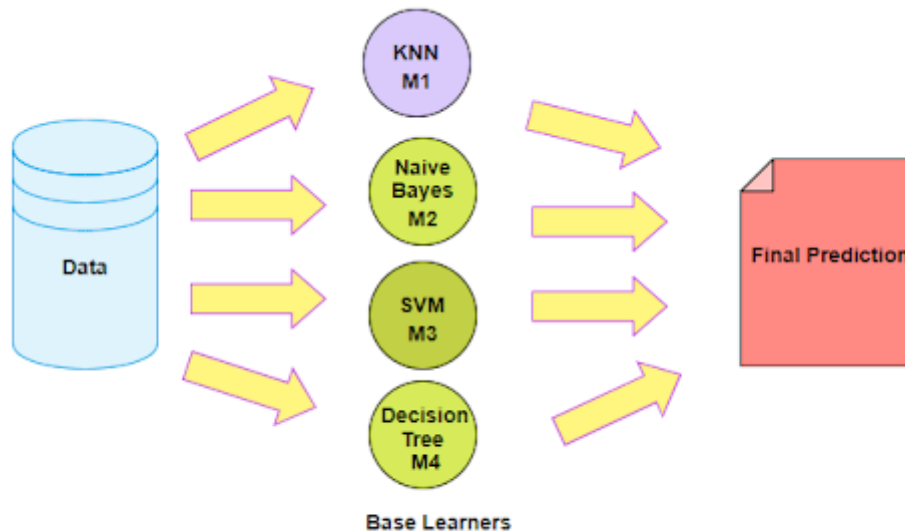


What is Ensemble Learning?

- Ensemble methods combine several trees base algorithms to construct better predictive performance than a single tree base algorithm.
- The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner, thus increasing the accuracy of the model.
- When we try to predict the target variable using any machine learning technique, the main causes of difference in actual and predicted values are noise, variance, and bias. Ensemble helps to reduce these factors (except noise, which is irreducible error).



- As you see in the picture here it combine 4 different algorithm, and by this 4 algorithm it predict the output. Now you can imagine here it combine different algorithm and predict so it more powerful than single model.

Now Let's Discuss About All Types Of Ensemble

Simple Ensemble

1. Max Voting
2. Averaging
3. Weighted Average

1. 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.

```
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(x_test)
pred2=model2.predict(x_test)
pred3=model3.predict(x_test)

final_pred = np.array([])
for i in range(0,len(x_test)):
    final_pred = np.append(final_pred, mode([pred1[i], pred2[i], pred3[i]]))

```

2. 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. $(5+7+8+3+5+4)/6 = 5.33$

```

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

```

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 $[(50.23) + (40.23) + (50.18) + (40.18) + (4*0.18)] = 4.41$.

```

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

- Now Discuss Advanced Ensemble Techniques
 1. Bagging
 2. Boosting
 3. stacking
 4. Blending
 5. cascading

1. Bagging

- It is also known as Bootstrapped Aggregation
- Bagging is mostly used to reduce the variance in a model. A simple example of bagging is the Random Forest algorithm.
- Bagging is one of the earliest, interesting and a very powerful ensemble algorithm.
- The core idea of bagging is to use bootstrapped replicas of the original dataset and use them to train different classifiers.
- Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.

How Bagging works on training dataset ?

Since Bagging resamples the original training dataset with replacement, some instance(or data) may be present multiple times while others are left out.

Original training dataset: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

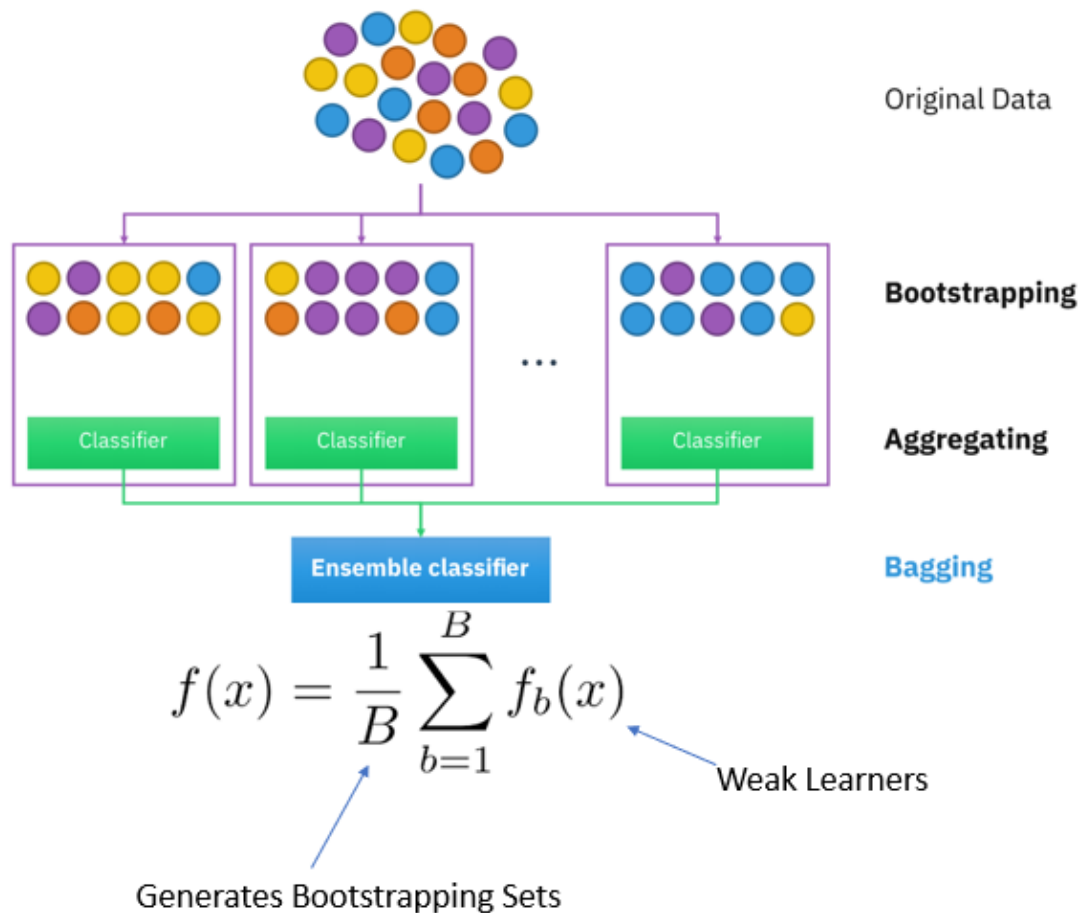
Resampled training set 1: 2, 3, 3, 5, 6, 1, 8, 10, 9, 1

Resampled training set 2: 1, 1, 5, 6, 3, 8, 9, 10, 2, 7

Resampled training set 3: 1, 5, 8, 9, 2, 10, 9, 7, 5, 4

Its Process Flow:

BAGGING LEARNING PROCEDURE

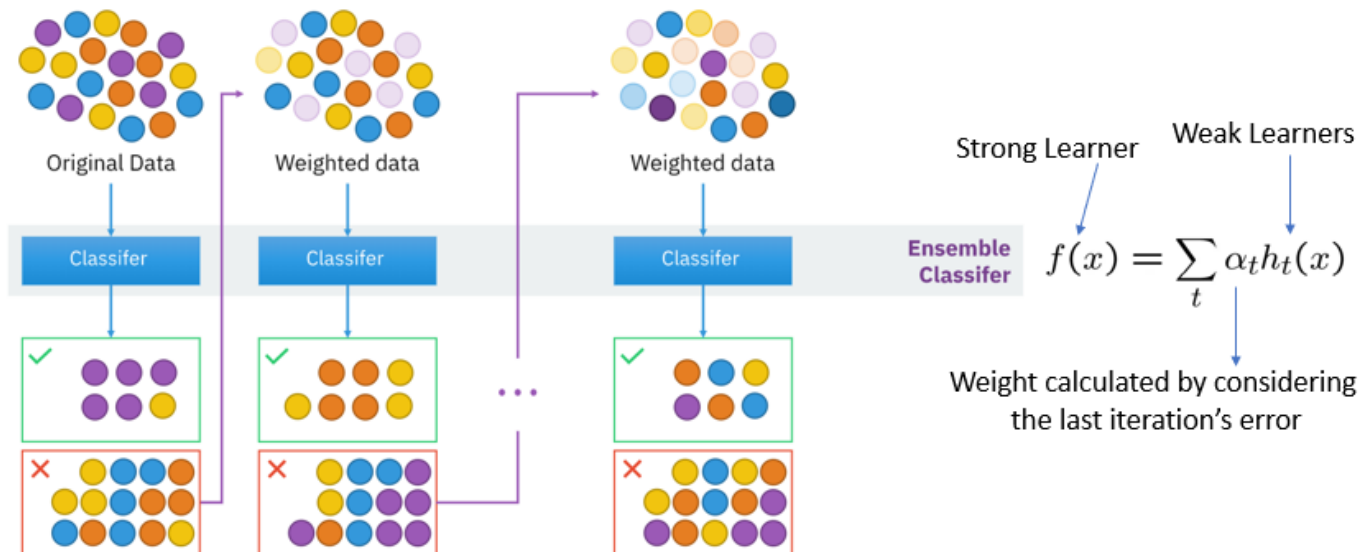


Eg. Random Forest Classifier

2. Boosting

- The second ensemble technique that we are going to discuss today is called Boosting.
- Boosting is used to convert weak base learners to strong ones. Weak learners generally have a very weak correlation with the true class labels and strong learners have a very high correlation between the model and the true class labels.
- AdaBoost was the first really successful boosting algorithm developed for the purpose of binary classification. AdaBoost is short for Adaptive Boosting and is a very popular boosting technique which combines multiple “weak classifiers” into a single “strong classifier”.

BOOSTING LEARNING PROCEDURE



Eg. AdaBoost ,GBM,XGBM,Light GBM,CatBoost

Similarities Between Bagging and Boosting –

- Both are ensemble methods to get N learners from 1 learner.
- Both generate several training data sets by random sampling.
- Both make the final decision by averaging the N learners (or taking the majority of them i.e Majority Voting).
- Both are good at reducing variance and provide higher stability.

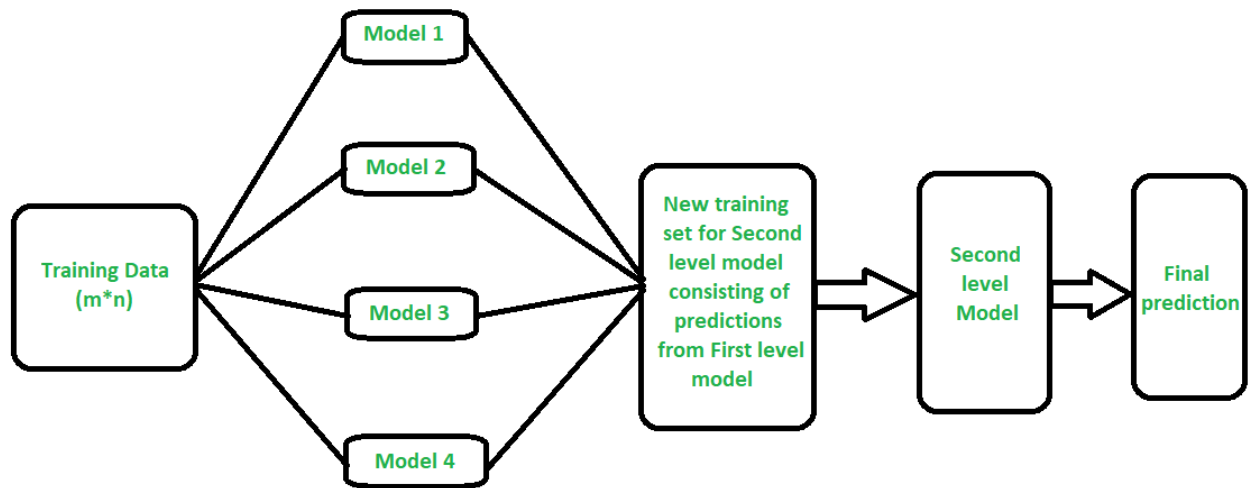
Where You Use bagging And Boosting

- If the difficulty of the single model is over-fitting, then Bagging is the best option.
- If the problem is that the single model gets a very low performance, Boosting could generate a combined model with lower errors as it optimises the advantages and reduces pitfalls of the single model.

3. Stacking

- Stacking is a way to ensemble multiple classifications or regression model.
- Stacking is an ensemble learning technique which is used to combine the predictions of diverse classification models into one single model also known as the meta-classifier.
- Stacking (sometimes called Stacked Generalization) is a different paradigm. The point of stacking is to explore a space of different models for the same problem.
- The idea is that you can attack a learning problem with different types of models which are capable to learn some part of the problem, but not the whole space of the problem. So, you can build multiple different learners and you use them to build an intermediate prediction, one prediction for each learned model.
- Then you add a new model which learns from the intermediate predictions the same target.
- This final model is said to be stacked on the top of the others, hence the name. Thus, you might improve your overall performance, and often you end up with a model which is better than any individual intermediate model. Notice however, that it does not give you any guarantee, as is often the case with any machine learning technique.

Stacking Procedure



How stacking works?

- We split the training data into K-folds just like K-fold cross-validation.
- A base model is fitted on the K-1 parts and predictions are made for Kth part.
- We do for each part of the training data.
- The base model is then fitted on the whole train data set to calculate its performance on the test set.
- We repeat the last 3 steps for other base models.
- Predictions from the train set are used as features for the second level model.
- Second level model is used to make a prediction on the test set.

Eg. Voting Classifier

4. Blending

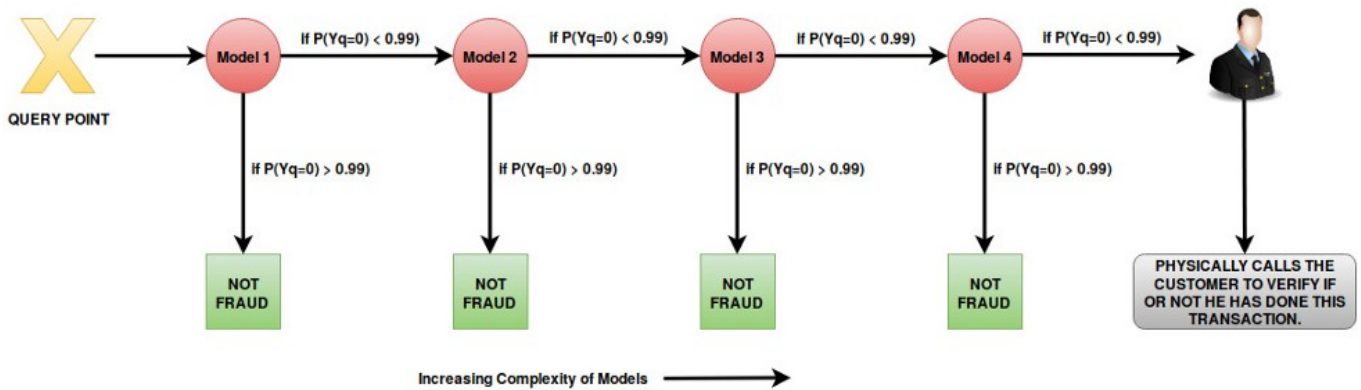
Blending is a similar approach to stacking.

- Blending is technique where we can do weighted averaging of final result.
- Blending is also an ensemble technique that can help us to improve performance and increase accuracy. It follows the same approach as stacking but uses only a holdout (validation) set from the train set to make predictions.
- Here is a detailed explanation of the blending process:
 - The train set is split into training and validation sets.
 - We train the base models on the training set.
 - We make predictions only on the validation set and the test set.
 - The validation predictions are used as features to build a new model.
 - This model is used to make final predictions on the test set using the prediction values as features.
- The difference between stacking and blending is that Stacking uses out-of-fold predictions for the train set of the next layer (i.e meta-model), and Blending uses a validation set (let's say, 10-15% of the training set) to train the next layer.

4. Cascading

- Cascading is one of the most powerful ensemble learning algorithm which is used by Machine Learning engineers and scientists when they want to be absolutely dead sure about the accuracy of a result.

- For example, suppose we want to build a machine learning model which would detect if a credit card transaction is fraudulent or not.
- If you think about it, it's a binary classification problem where a class label 0 means the transaction is not fraud & a class label 1 means the transaction is fraudulent.
- In such a model, it's very risky to put our faith completely on just one model. So what we do is build a sequence of models (or a cascade of models) to be absolutely sure about the fact that the transaction is not fraudulent.
- Cascade models are mostly used when the cost of making a mistake is very very high. I will try to explain cascading with the help of a simple diagram.



DIFFERENT STAGES OF QUERING CASCADE CLASSIFIERS IN A FOUR MODEL CASCADE SYSTEM

Look at the above diagram. Given that we have a transaction query point X_q , we will feed it to Model 1. Model 1 can be anything a random forest, or a logistic regression model or maybe a support vector machine. It can be anything! Basically what Model 1 does is that it predicts class probabilities to determine to which class do a given query point has higher chances of belonging to.

Let's say class label 1 means the transaction is fraudulent, and class label 0 means the transaction is not fraud. Typically, the predicted probabilities is given by this — $P(Y_q=0)$ and $P(Y_q=1)$, where Y_q is our actual class label. Now let's assume that $P(Y_q=0)$, i.e. the probability of the transaction to be not fraudulent is very high. If you think carefully, if $P(Y_q=0)$ is extremely high, we will say that the transaction is not fraud. Let's assume we have set a threshold of 99%. It means if and only if $P(Y_q=0) > 0.99$, we will declare the final prediction to be not fraudulent. However, if $P(Y_q=0) < 0.99$ we are not very sure if or not it's a fraudulent transaction although though there is a high chance that the transaction is not fraudulent. In such a case, when $P(Y_q=0) < 0.99$, we want to be really really sure that the transaction is not fraudulent. We need to be absolutely careful because if our model fails to detect a fraudulent transaction we might lose millions of dollars! So even when we are slightly unsure, we will train another Model 2. Model 2 does the same thing, it receives the query point and predicts $P(Y_q=0)$. Just like in stage 1, if $P(Y_q=0) > 0.99$, we will declare the transaction to be not fraudulent and terminate the loop. But again if we get $P(Y_q=0) < 0.99$, we aren' sure! Hence, we will pass the query point to another Model 3 in the cascade which does the same thing.

In a typical cascading system the complexity of models increases as we add more and more models to the cascade. Please note that all the models in a cascade are super powerful and has a very high accuracy on unseen data. However, it might happen that none of the models can give us a value of $P(Y_q=0) > 0.99$. In such a case, typically there is a human being who sits at the end of a cascade. This person will personally call the customer and ask him whether or not he has done the transaction. Now, we are absolutely certain that the transaction is not a fraud one when the customer says that he is the one who has done the transaction.

@@ Credit:- <https://medium.com/@saugata.paul1010/ensemble-learning-bagging-boosting-stacking-and-cascading-classifiers-in-machine-learning-9c66cb271674> (<https://medium.com/@saugata.paul1010/ensemble-learning-bagging-boosting-stacking-and-cascading-classifiers-in-machine-learning-9c66cb271674>)