Chapter 6

The stacking ensemble approach

This chapter proposes the stacking ensemble approach for combining different data mining classifiers to get better performance. Other combination techniques like voting, bagging etc are also described and a comparative description showing how stacking is having advantages over others is also shown.

6.1 An introduction to combined classifier approach

In previous chapters, different classifiers that are very popular owing to their simplicity and accuracy were discussed. Since the last few years the researches in data mining are to another direction called meta learners. Here the scientific community tried to address the question" Whether a combined classifier model gives a better performance than selecting the single base level model with best accuracy?"

There are many answers to the above questions. Scientists tried to think about various possibilities in which classifiers can be combined and their performances are compared with the best among the base level classifiers from which they are made. In this research work, a study was conducted to find how ensemble of classifiers improves model performance. In the following section some of the most common model combining approaches that exist in the data mining, are analysed.

6.2 Popular ways of combining classifiers

In our day to day life, when crucial decisions are made in a meeting, a voting among the members present in the meeting is conducted when the opinions of the members conflict with each other. This principle of "***voting***" can be applied to data mining also. In voting scheme, when classifiers are combined, the class assigned to a test instance will be the one suggested by most of the base level classifiers involved in the ensemble. Bagging and boosting are the variants of the voting schemes.

Bagging is a voting scheme in which n models, usually of same type, are constructed. For an unknown instance, each model's predictions are

recorded [7, 9]. That class is assigned which is having the maximum vote among the predictions from models.

Boosting is very similar to bagging in which only the model construction phase differs. Here the instances which are often misclassified are allowed to participate in training more number of times. There will be n classifiers which themselves will have individual weights for their accuracies. Finally, that class is assigned which is having maximum weight [7, 47]. An example is Adaboost algorithm. Bagging is better than boosting as boosting suffers from over fitting. Over fitting is that phenomenon where the model performs well only for the training data. This is because it knows the training data better and it does not know much about unknown data.

There are 2 approaches for combining models. One of them uses ***voting*** in which the class predicted by majority of the models is selected, whereas in ***stacking*** the predictions by each different model is given as input for a meta level classifier whose output is the final class. Whether it is voting or stacking, there are two ways of making an ensemble. They are ***Homogenous ensemble*** where all classifiers are of same type and ***heterogeneous ensemble*** where the classifiers are different.

The basic difference between stacking and voting is that in voting no learning takes place at the meta level, as the final classification is decided by the majority of votes casted by the base level classifiers whereas in stacking learning takes place at the meta level.

6.3 Stacking framework-a detailed view

Stacking is the combining process of multiple classifiers generated by different learning algorithms $L_1 \ldots L_n$ on a single dataset. In the first phase a set of base level classifiers C_1, $C_2 \ldots C_n$ is generated. In the second phase a meta level classifier is developed by combining the base level classifier.

The WEKA data mining package can be used for implementing and testing the stacking approach. Meta learning is a separate area in the data mining domain and is usually a part of ensemble methods which are one of the hottest research fields. The following section analyses the impact of meta learning in data mining.

6.4 Impact of ensemble data mining models

A methodology on how models can be combined for customer behavior is described in [21]. Companies are eager to learn about their customer behavior using data mining technologies. But the diverse requirements of such companies make it difficult to select the most effective algorithm for the given problem. Recently, a movement towards combining multiple classifiers has emerged to improve classification results. In [21], a method for the prediction of the customer's purchase behavior by combining multiple classifiers based on genetic algorithm is proposed.

One approach in combining models is called the Meta decision trees, which deals with combining a single type of classifier called decision trees. [55] Introduces Meta decision trees (MDTs) as a novel method for combining multiple models. Instead of giving a prediction, MDT leaves specify which model should be used to obtain a prediction.

In this work, the focus is on how classifier performance can be improved using the stacking approach. While conventional data mining

research focuses on how the performance of a single model can be improved, this work focuses on how heterogeneous classifiers can be combined to improve classifier performance. It was also observed that this approach yields better accuracy in the domain of employment prediction problems. In [71], it is being observed that when one prepares ensemble, the number of base level classifiers is not much influencing, and usually researchers select 3 or 7 at random depending on the type of applications. In this research, three classifiers namely decision tree, neural network, and Naive Bayes classifier were selected for making an ensemble. They have been tested individually as explained in chapter 5 and their ensemble is explained in this chapter.

There are many strategies for combing classifiers like voting, bagging and boosting each of which may not involve much learning in the Meta or combing phase. Stacking is a parallel combination of classifiers in which all the classifiers are executed parallel and learning takes place at the Meta level. To decide which model or algorithm performs best at the Meta level for a given problem, is also an active research area, which is addressed in this thesis. It is always a debate that whether an ensemble of homogenous or heterogeneous classifiers yields good performance. [36] Proposes that depending on a particular application an optimal combination of heterogeneous classifiers seems to perform better than the homogenous classifiers.

When only the best classifier among the base level classifiers is selected, the valuable information provided by other classifiers is being ignored. In classifier ensembles which are also known as combiners or committees, the base level classifier performances are combined in some way such as voting or stacking.

Research has shown that combining a set of simple classifiers may result in better classification in comparison to any single sophisticated classifier [18, 39, and 59]. In [17], Dietterich gave three fundamental reasons for why ensemble methods are able to outperform any single classifier within the ensemble — in terms of statistical, computational and representational issues. Besides, plenty of experimental comparisons have been performed to show significant effectiveness of ensemble. Assume there are several different, but equally good, training data sets. A classifier algorithm is biased for a particular input *x* if, when trained on each of these data sets, it is systematically incorrect when predicting the correct output for *x*. An algorithm has high variance for a particular input *x* if it predicts different output values when trained on different training sets. The prediction error of a learned classifier is related to the sum of the bias and the variance of the learning algorithm. Usually there is a trade-off between bias and variance. A learning algorithm with low bias must be "flexible" so that it can fit the data well. But if the learning algorithm is too flexible, it will fit each training data set differently, and hence have high variance. Mathematically, classifier ensembles provide an extra degree of freedom in the classical bias/variance trade off, allowing solutions that would be difficult (if not impossible) to reach with only a single classifier.

6.5 Mathematical insight into stacking ensemble

If an ensemble has M base models having an error rate e < 1/2 and if the base models' errors are independent, then the probability that the ensemble makes an error is the probability that more than M/2 base models misclassify the example. The simple idea behind stacking is that if an input–output pair

(x, y) is left out of the training set of h_i, after training is completed for h_i, the output y can still be used to assess the model's error. In fact, since (x, y) was not in the training set of h_i, $h_i(x)$ may differ from the desired output y. A new classifier then can be trained to estimate this discrepancy, given by $y - h_i(x)$. In essence, a second classifier is trained to learn the error the first classifier has made. Adding the estimated errors to the outputs of the first classifier can provide an improved final classification decision [38].

6.6 Stacking ensemble framework applied in this work

In this work, keeping the three base level classifiers as same, various meta level classifiers were tested and it was observed that multi response modal trees(M5') meta level classifier performed best among others. Although numerous data mining algorithms have been developed, a major concern in constructing ensembles is how to select appropriate data mining algorithms as ensemble components.

A challenge is that there is not a single algorithm that can outperform any other algorithms in all data mining tasks, i.e. there is no global optimum solution in selecting data mining algorithms although much effort is devoted to this area. An ROC (Receiver Operating Characteristics) analysis based approach was approved to evaluate the performance of different classifiers. For every classifier, its TP (True Positive) and FP (False Positive) are calculated and mapped to a two dimensional space with FP on the x-axis and TP on the y-axis. The most efficient classifiers should lie on the convex hull of this ROC plot since they represent the most efficient TP and FP trade off. The three base level classifiers decision tree (ROC=0.77, ACC=0.803), neural networks (ROC=0.76, ACC=0.797) and naive Bayes (ROC=0.79,

ACC=0.794) are the best choices among base level classifiers for this domain. It was clear that by combing classifiers with stacking using an algorithm namely multi response model trees (M5'), an accuracy of (82.2) was observed which is better than selecting best among base level classifier accuracy in this work.

The table 6.1 shows the summary.

	Base level classifiers			**Meta level classifiers**		
Classifier	**Decision tree**	**Neural Network**	**Naïve Bayes**	**Bagging**	**Regression**	**M5'**
Accuracy	80.3	79.7	79.4	79.2	78	**82.2**
Precision	0.57	0.62	0.61	0.75	0.78	**0.87**
Recall	0.58	0.61	0.65	0.9	0.85	**0.76**
ROC	0.77	0.76	0.79	0.88	0.88	**0.89**

Table 6.1: Summary performances with base & Meta level classifier

Multi response model trees: When decision trees are constructed, if linear regression principles are also adopted, for each of the m target classes, m regression equations are formed. This concept is adopted in an algorithm namely M5 by Quinlan [71]. Given a new example x to classify, LR_j (x) is calculated for all j, and the class k is predicted with maximum LR_k (x). In multi response modal trees, instead of m linear equations, m model trees are induced. Model trees combine a conventional decision tree with the possibility of linear regression functions at the leaves and also it is capable of dealing with continuous attributes. This representation is relatively perspicuous

because the decision structure is clear and the regression functions do not normally involve many variables. M5' algorithm uses this concept and gives a better performance when used at the Meta level of the stacking ensemble for this domain. This is illustrated in figure 6.1. The M5' algorithm is implemented in Weka package as M5P algorithm. The experimental set up in Weka for ensemble learning is shown in figure 6.2.

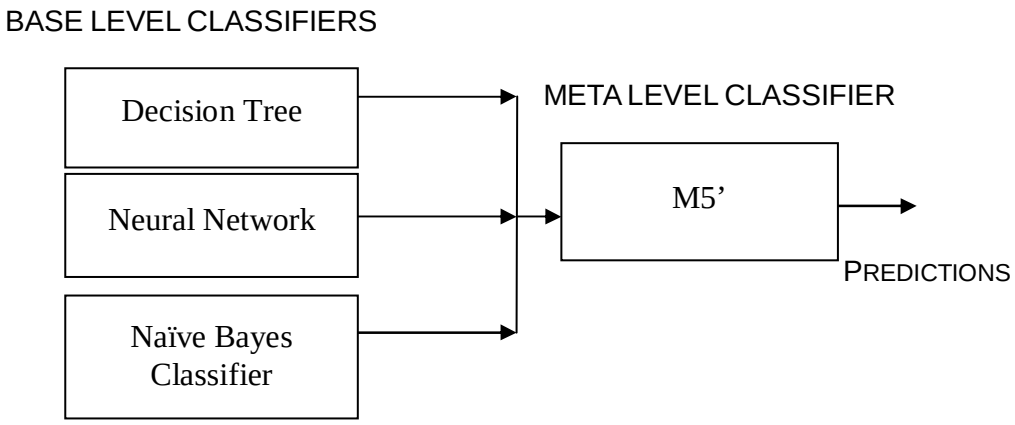

Fig 6.1: Model ensemble using stacking

Some Meta level algorithms expect only two class problems. So, in order to test those possibilities., the 4-class problem may be modelled as a 2- class problem, by combining attributes "Excellent" and "Good" as "Excellent" and "Average" and "Poor" as "Poor" . Then experiments on this two class problem showed that with the same base level classifiers, the voted perceptron algorithm was giving better performance in terms of accuracy (82%).

The Voted Perceptron algorithm: In the voted-perceptron algorithm, more information is stored during training and then it uses this elaborate information to generate better predictions on the test data. The algorithm is detailed below.

The information maintained during training was the list of all prediction vectors that were generated after each and every mistake. For each such vector, the number of iterations it survived was counted until the next mistake was made; this count was referred as the weight of the prediction vector. To calculate a prediction, the binary prediction of each one of the prediction vectors was computed and all these predictions were combined by a weighted majority vote. The weights used are the survival times described above. This makes intuitive sense as good prediction vectors tend to survive for a long time and thus have larger weight in the majority vote.

The algorithm:

Input: A labelled training set $<(x_1,y_1)\ldots.(x_m, y_m)>$ where $x_1 \ldots x_m$ are feature vector instances and $y_1 \ldots y_m$ are class labels to which the training instances have to be classified, T is the no of epochs.

Output: A list of weighted perceptrons $<(w_1,c_1)\ldots(w_k,c_k)>$ where $w_1 \ldots w_k$ are the prediction vectors and $c_1 \ldots c_k$ are the weights.

K=0

$w_1 = 0$

$c_1 = 0$

Repeat T times

For i = 1 to m

If (x_i, y_i) is misclassified:

$w_{k+1} = w_k + y_i x_i$

$c_{k+1} = 1$

$k = k + 1$

Else

$c_k = c_k + 1$

At the end, a collection of linear separators w_0, w_1, w_2, etc, along with survival times: c_n = amount of time that w_n survived is returned. This c_n is a good measure of the reliability of w_n. To classify a test point x, use a weighted majority vote: y'=sgn(S) where S is the sign function which returns output to a range as one of the values {0, 1,-1}. This is shown in equation (6.1).

$$\mathbf{Y'=}\ \operatorname{sgn}\left\{\sum_{n=0}^{N} c_n\ \operatorname{sgn}(w_n \cdot x)\right\} \qquad \textbf{- (6.1)}$$

6.7 Advantages of stacking ensemble methods

Stacking takes place in two phases. In the first phase each of the base level classifiers takes part in the j- fold cross validation training where a vector is returned in the form $<(y'_0 \ldots y'_m), y_j>$ where y'_m is the predicted output of the m^{th} classifier and y_j is the expected output for the same . In the second phase this input is given for the Meta learning algorithm which adjusts the errors in such a way that the classification of the combined model is optimized. This process is repeated for k-fold cross validation to get the final stacked generalization model. It is found that stacking method is particularly better suited for combining multiple different types of models. Stacked generalization provides a way for this situation which is more sophisticated than winner-takes-all approach [16, 39]. Instead of selecting one specific generalization out of multiple ones, the stacking method combines them by using their output information as inputs into a new space. Stacking then generalizes the guesses in that new space. The winner-takes-all combination approach is a special case of stacked generalization. The simple voting approaches have their obvious limitations due to their abilities in capturing only linear relationships. In stacking, an ensemble of classifiers is first trained

using bootstrapped samples of the training data, producing level-0 classifiers. The outputs of the base level classifiers are then used to train a Meta classifier. The goal of this next level is to ensure that the training data has accurately completed the learning process. For example, if a classifier consistently misclassified instances from one region as a result of incorrectly learning the feature space of that region, the Meta classifier may be able to discover this problem. Using the learned behaviours of other classifiers, it can improve such training deficiencies [16].

It is always an active research area that whether combining data mining models gives better performance than selecting that model with best accuracy among base level classifiers. In this research also, in pursuit for finding the best model suitable for this problem, this possibility was explored. In combining of the models usually the models in level 0(base level classifiers) are operated in parallel and combined with another level classifier called as Meta level classifier. In this work, using decision tree, neural network and Naive Bayes classifier as the base level classifiers, various Meta level classifiers have been tested and it was observed that multi response model tree Meta level classifier performed best among others. Although numerous data mining algorithms have been developed, a major concern in constructing ensembles is how to select appropriate data mining algorithms as ensemble components. As pointed out earlier numerous research works are being taken place in the field of ensemble of classifiers and many of them are proposing different types of classifiers at the base level and Meta level depending on the type of application. This research work is also giving light into the field of data mining research by proposing an efficient combination of base and Meta level classifiers for a social science problem like employment prediction.

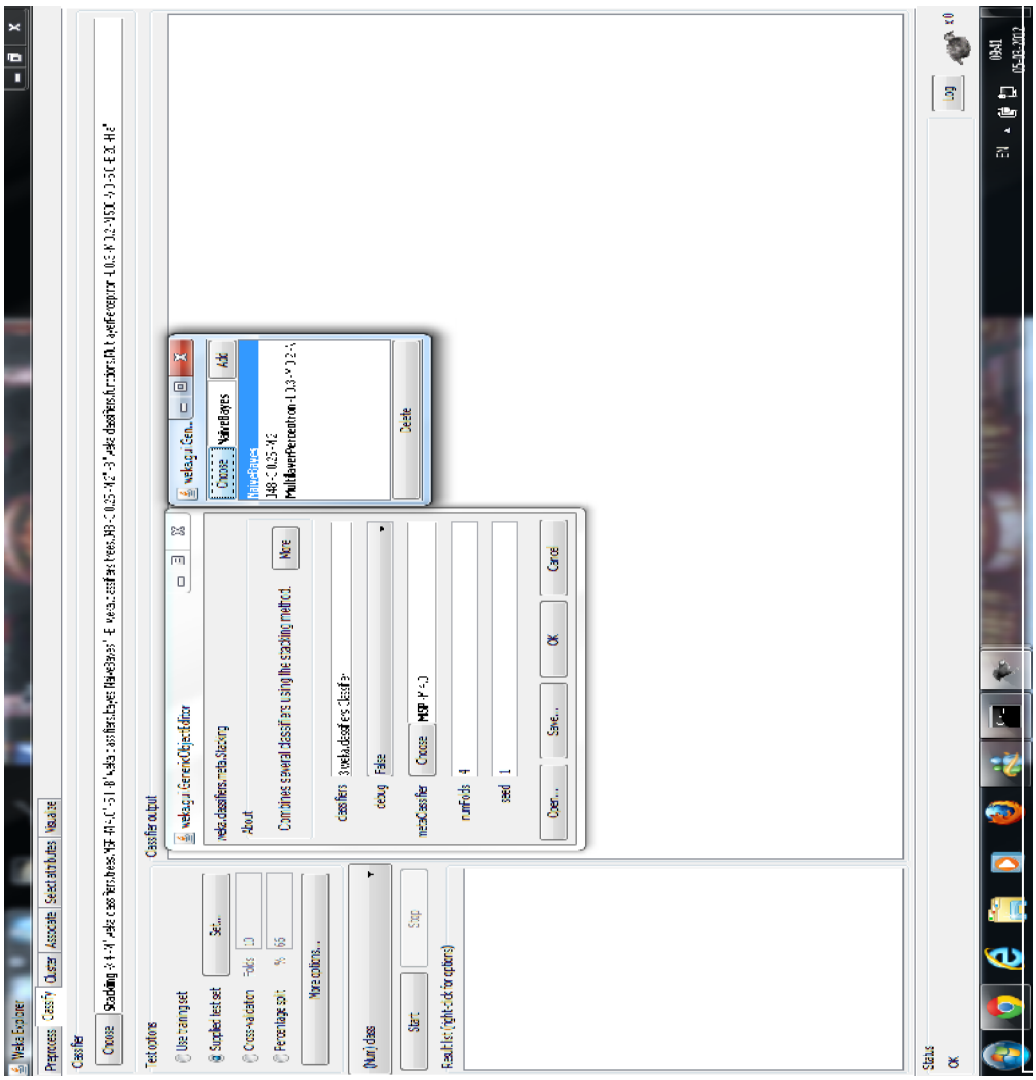

Figure 6.2: Weka Experimental set up for ensemble learning

6.8 Chapter summary

This chapter suggests the need for making an ensemble of classifiers and the various methods for making it. One of the hottest questions in front of data mining researchers today is "Whether a combined classifier model gives better performance than the best among the base level classifiers". This important question is tried to be addressed in this thesis. For exploring this there are two approaches-voting based techniques and stacking based techniques. The basic difference between stacking and voting is that in voting no learning takes place at the Meta level, as the final classification is by votes casted by the base level classifiers, whereas in stacking, learning takes place in the Meta level. A Meta level is the level at which the base level classifiers are combined using an algorithm. In this work, the base level classifiers are stacked with many meta learning algorithms like bagging, regression based algorithms etc, and it was observed that when multi response model tree algorithm is used at the Meta level the model is giving much better performance. Hence through this work, it is strongly suggested that a combined ensemble approach gives a better performance than selecting the best base level classifier, which also confirms few other research results that have happened in other functional domains [11, 25, and 38].