Ensemble learning

Wisdom of whole group is considered

Combining predictions of multiple machine learning models and taking final call.

Ensemble techniques have four aspects

① Diversity of opinion —

each member has different fact representation (eg weights) better than random chance. Every member has a different mapping function

② Independance -

each member must vote independantly

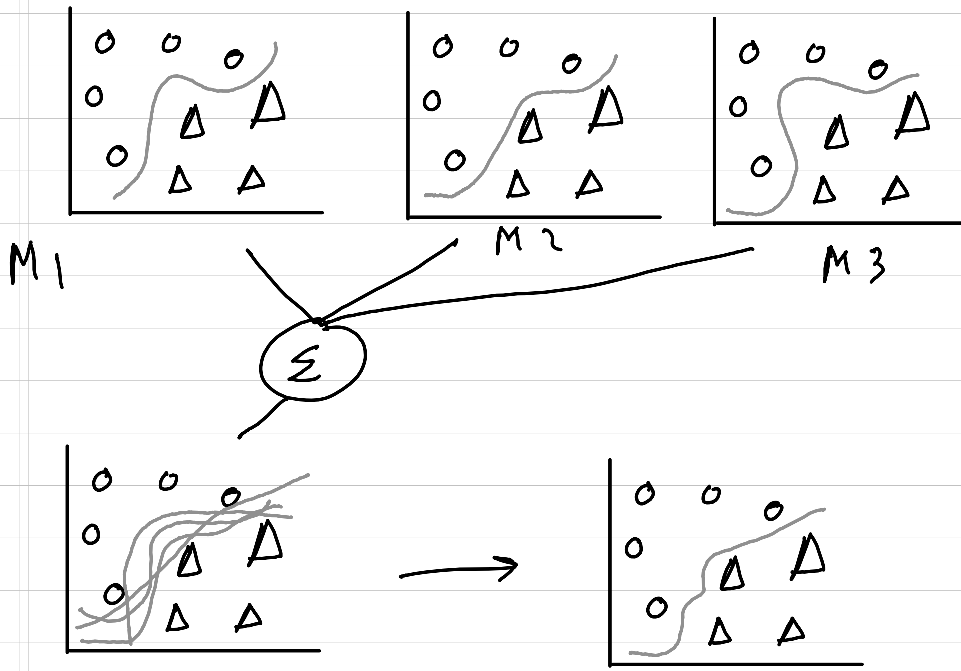
③ Decentralization -

Members are able to specialize on local data ie every member might have access to different parts of data.

④ Aggregation -

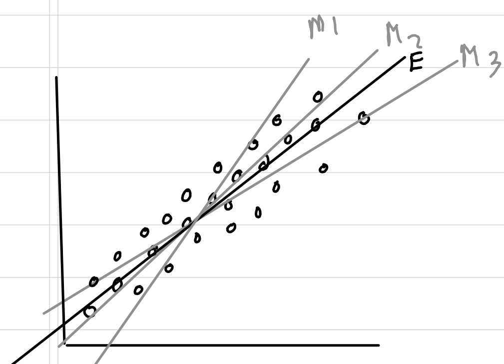
Mechanism to turn indivisual judgements to judgement of whole. (eg majority voting)

Graphical View



Combined predictions

Aggregated ensemble model-classification



Regression Ensemble

Ensemble methods might reduce the variance at cost of computing several models.

Sometimes ensemble methods may fail & a single model might give better performance

Ensemble methods must be used only if it is better than the best constituent member

Benefits of ensemble techniques →

1. performance : Better predictions and performance than a single model.

2. Robustness : Reduces the spread of model predictions & overfitting

3. Relaibility : Better to rely on multiple models for failure.

Empirical & theoritical evidence shows that some techniques like bagging reduce the variance while some like adaboost reduce both the bias as well as the variance.

Diversity in Ensemble Models.

Diversity means how different the models think. It is very important for the Model

If everything is same except random seed, then diversity won't be much, as the model will eventually learn same mapping

eg. If we make ensemble model of 5 decision trees on same data, then the models will all be the same.

Ways to increase diversity →

① Data Sample Manipulation →

Give different samples of dataset to different models.

Some will have few samples, others may have few other samples

Random subspace selection

② Input feature manipulation →

Provide different groups of input features to different models.

eg some have height, age, some have weight, age, gender, etc.

① & ② Together can be called as partitioning the search space and used in random forest.

③ Learning parameter manipulation →

Vary hyperparameters

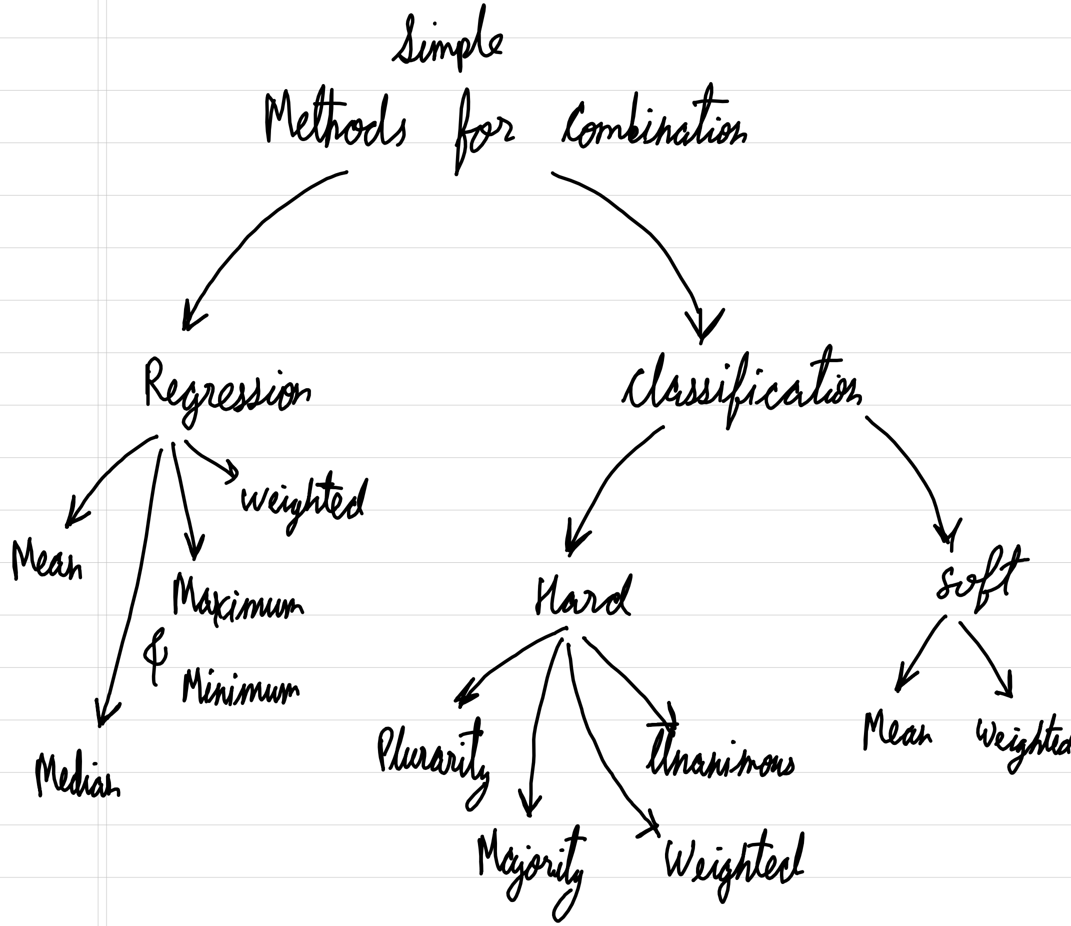
Vary starting point (eg initial clusters in k-Means)

Vary optimizing algorithm

④ Output representation manipulation →

Vary encoding

Modify representations in a manner which ensemble model learns something different.



Find the resultant value from the individual model outputs

Combining Classification Predictions

(A) Label Prediction

Discrete labels like "Red" "Green" "Blue" can be combined by Voting techniques

1. Plurality Voting → Majority wins
2. Majority Voting → The votes must be > 50% else no prediction
3. Unanimous Voting → All must vote same, else no decision
4. Weighted voting → weights attached to models by factors like accuracy & type of the model.

(B) Probability based prediction

Models may predict a probability for every class

$$\left.\begin{array}{l} \text{Red} : 0.75 \\ \text{Green} : 0.10 \\ \text{Blue} : 0.15 \end{array}\right\} \Sigma = 1$$

1. Vote using Mean probabilities
2. Vote using Sum probabilities
3. Vote using Weighted sum probabilities.

These are "soft voting" methods.

Dynamic classifier Selection

Suppose we have 3 models, decision tree (gini), decision tree (entropy) and kernalized SVM to classify buy car or not.

A student comes with fair credit score & middle age.

Model 1 : Buy
Model 2 : Not Buy
Model 3 : Buy.

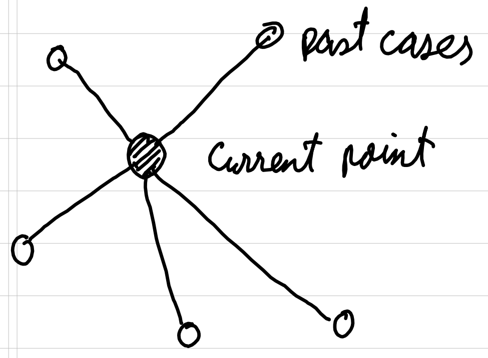
} Which model to give preference?

What if different models are good for different types of applications?

Based on the test data we can see that Model 2 has performed good (high accuracy) for cases similar to this case

Then we must give preference to Model 2.

In order to check cases similar to the case we have, we use k NN



Chose the best model based on the history of the models performance on such cases.

This can be thought of as giving a person a job he is good at.

This is one example of Dynamic Classifier Selection Local Accuracy DCS - LA

Estimate each classifiers' accuracy in local region of feature space sorrounding an unknown test sample and then use the decision of the most locally accurate classifier.

Dynamic classifier selection (DCS) partitions the input feature space in some way and assigning specific models to be responsible for making predictions for each partitions.

Accuracies can be of two types

Overall Local accuracy → for k samples the total accuracy is considered

Local Class Accuracy → estimated for every class in k samples.

eg accuracy of "Not give" of model 2 in k samples.

Dynamic Ensemble Selection

DES is extension of DCS that chooses a subset (instead of just one) model for making predictions.

Predictions are made by voting of models that perform well in the feature space.

One popular algorithm is KNORA by albert et al in 2008

K-Nearest Neighbour Oracle.

It is just like the DCS-LA algorithm with the exception that multiple algorithms are selected instead of just one.

Dynamic Ensemble Selection Library DESlib is an open source library that provides implementation of the DES algorithms

Combining Regression Preditions

1. Mean predicted value
2. Median predited value
3. Weighted Mean
4. Minimum predicted value
5. Maximum predicted value

} conservative

Model 1 : 99
Model 2 : 101
Model 3 : 98

Mean 99.33
Median 99
Min : 98 , Max : 101

Median Predicted value is more appropriate when distribution of predictions does not follow a gaussian distribution

Common Ensemble Methods

(A) Bagging

"Bootstrap Aggregation"

Training data is varied.

Diversity is ensured by the variations within the bootstrapped replicas on which each classifier is trained.

Relatively weak classifier (eg unpruned decision trees) whose decision boundaries measurably vary with respect to relatively small changes in training data

Each model gets own unique sample

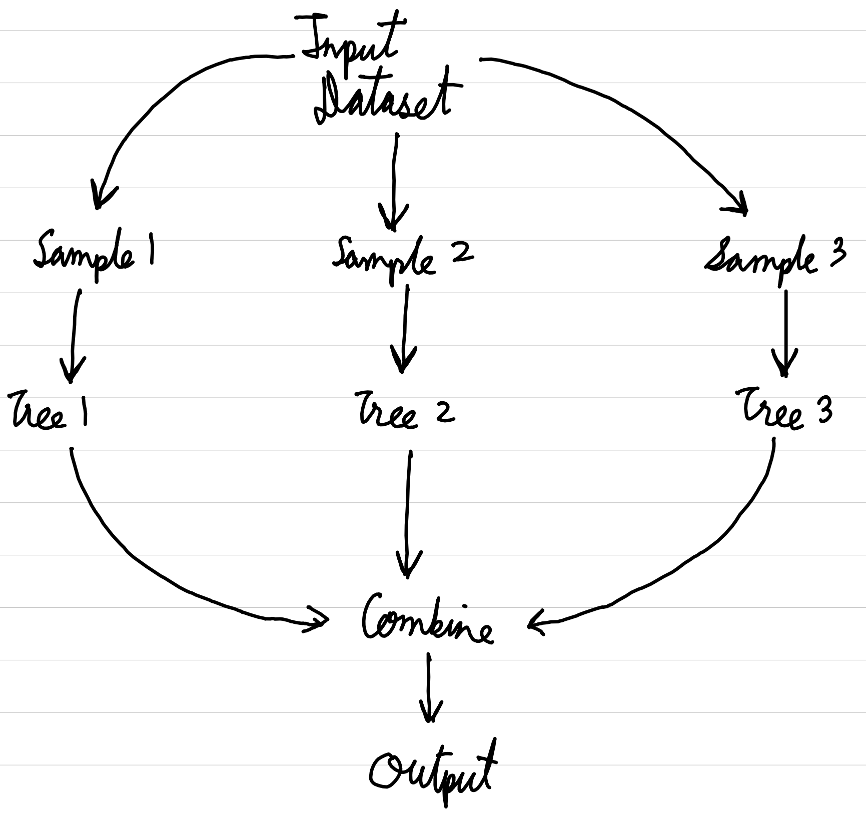
Each data sample may get selected zero, one or multiple times.

Simple voting or averaging is used

eg. Bagged decision trees (Canonical bagging)

Random Forest

Extra trees



(B) Boosting

Boosting seeks to change the training data to focus attention on examples that previous fit models have gotten wrong.

Each subsequent classifier increasingly focuses on instances misclassified by previously generated classifiers.

Training dataset is left unchanged, instead the learning algorithm is modified to pay more or less attention to specific examples based on weather they have been predicted correctly or incorrectly by previous members.

Correcting previous errors.

Bagging — Independent models
Boosting — Models built on top of one another.

Uses simple models called as weak learners (like decision trees) that make only a single or few decisions, each of which can barely do better than random guessing.

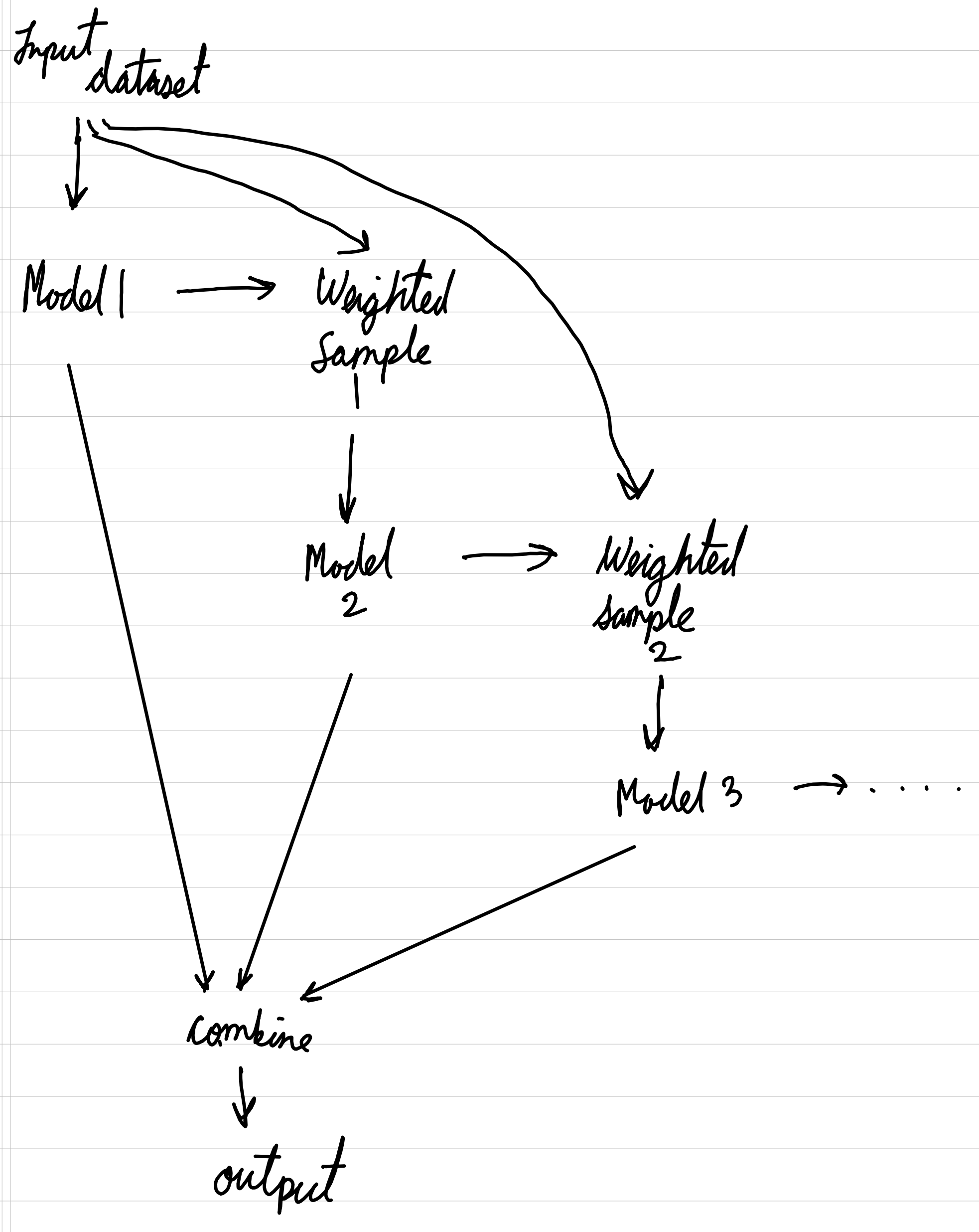
Change training data to give more importance to examples that are hard to predict

↓

Iteratively add ensemble models to correct predictions of models

↓

Combine predictions using weighted average of models



eg. AdaBoost
Gradient Boost
Stochastic Gradient Boost (XG Boost)

(c) Stacking Ensemble Learning

diverse group of model types & a model to combine

Complex algorithms are used to make predictions while a simple ML algorithm is used to combine them.

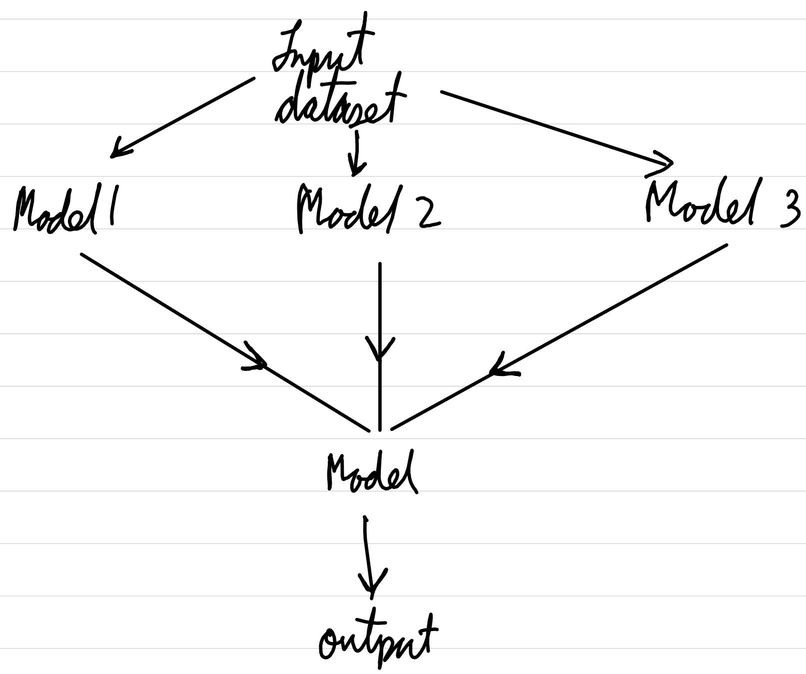
Unchanged training dataset

↓

Different ML algorithms on entire data

↓

ML model to learn how to best combine ensemble



stacked generalization

Blending ensemble

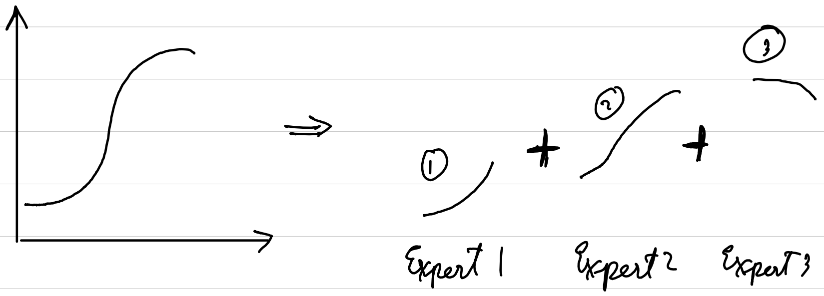
Super Learner ensemble

Mixture of experts ensemble

It is form of stacking with dynamic selection like input space partioning

Part of meta-learning

Divide task into subtasks, develop an expert for every group (Divide & rule)



1. Division of tasks into subtasks
2. Develop an expert for every subtask
3. Use a gating model to decide which model to use
4. Pool predictions and gating model output to make predictions

Dividing tasks is done by domain knowledge eg dividing an image into seperate elements like foreground, background, object, lines etc.

The gating model is usually a neural network. that takes the input pattern and outputs the contribution that each expert should have in making a prediction for the input.
MoE learns which portion of the feature space is learnt by each ensemble member.

The experts & the gating model are learnt together

Pooling is done by combining the weighted sum of all the classifier preditions with weight given by gating model confidence.