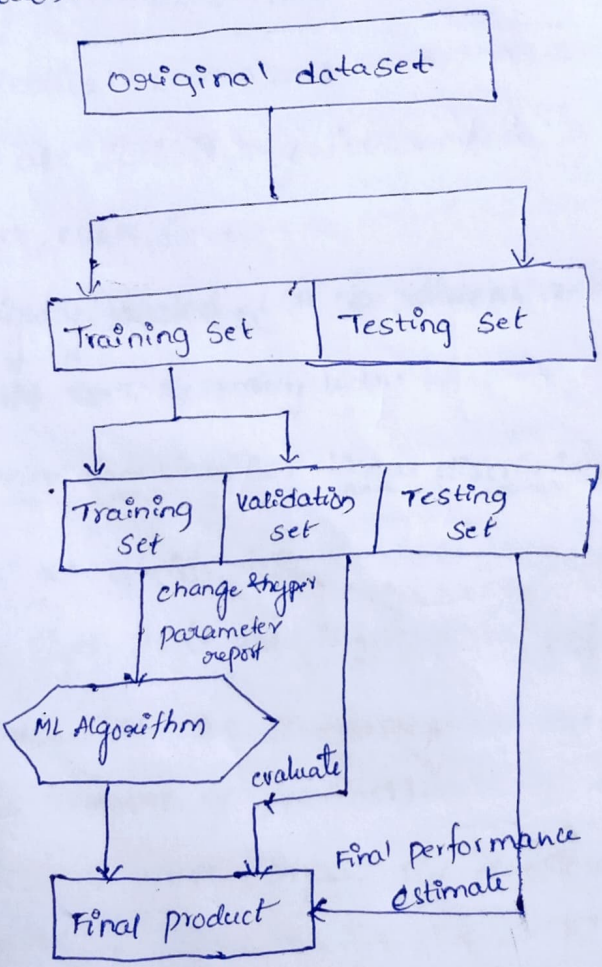
UNIT-IV

Learning best practices for model evaluation and hyper parameter tuning:

2b) Streamlining workflows with pipelines using k-fold Cross validation to access model performance:-

A popular approach for estimating the generalisation performance of machine learning model is Hold out cross validation.



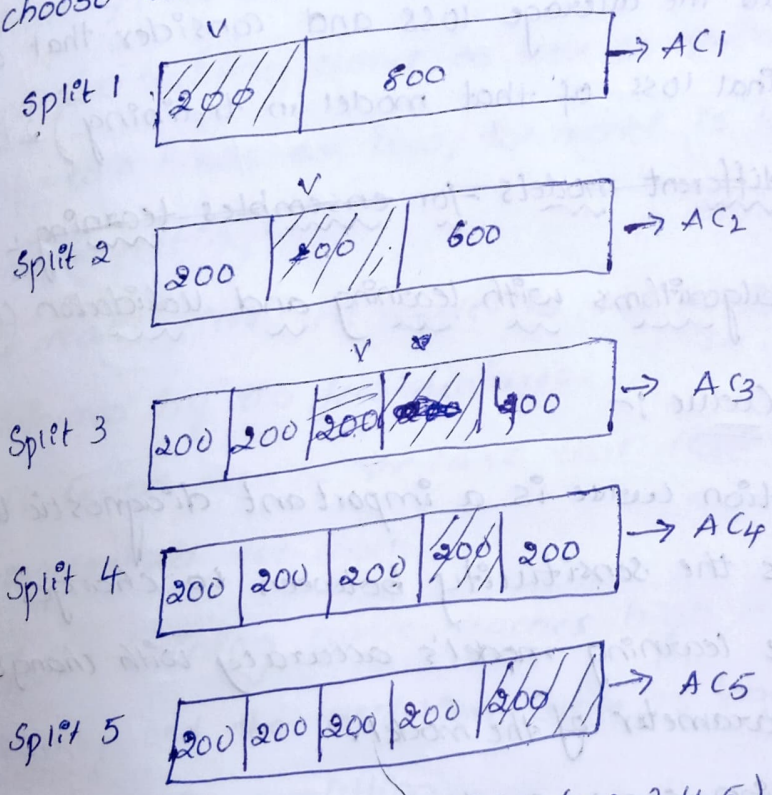
1. A better way of using the hold out method for model selection is to separate the data into 3 parts; (A training set, validation set and testing set).
2. The training set is used to fit the different models and the performance on the validation set is then used

for the model selection.

3. The advantage of having a test set that the model has not seen before during the training and model selection steps in that we can obtain a less biased estimate of it's ability to generalize in new data.

K-fold cross validation :-

- choose value for K=5 (5 to 10).



1. In K-fold cross validation we randomly split the training dataset into K-folds without replacements, where K-1 folds are used for the model training and one fold is used for performance evaluation. This procedure is repeated 'K' times. So that we obtain K-models and performance estimates.

Step 1 :- We split the data into test and train has the ratio of

Step 2: Define K-folds, K is an integer, (K=5) usually it holds b in 5 to 10. Now we will repeat the train data into trained and test data according to K folds. Here we train 'K-1' of data for training data [1-K-1] of data for testing each times.

Step 3: Calculate the Lacus of each K-fold cross validation

Step 4: Calculate the average loss and consider that as the final loss of that model in training) 2b)

~~Combining different models for ensembles learning~~:-

* Debugging algorithms with learning and validation curve:

Validation Curve:-

* A validation curve is a important diagnostic tool that shows the sensitivity between to changes in a machine learning model's accuracy with change in some parameter of the model.
* A validation curve is typically drawn between some parameter of the model and the model's score.
* Two curves are present in a validation curve - one for the training set score and one for the cross-validation score.
* By default, the function for validation curve, present in the scikit learn library performs 3-fold cross-validation.
* A validation curve is used to evaluate an existing model based on hyper parameters and is not used to tune a model.

* If we tune the model according to the validation curve/score the model may be biased towards the specific data against which the model is tuned.

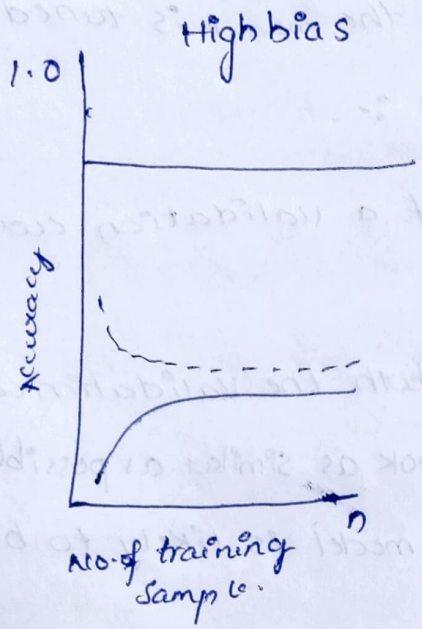
Interpreting a validation curve :-

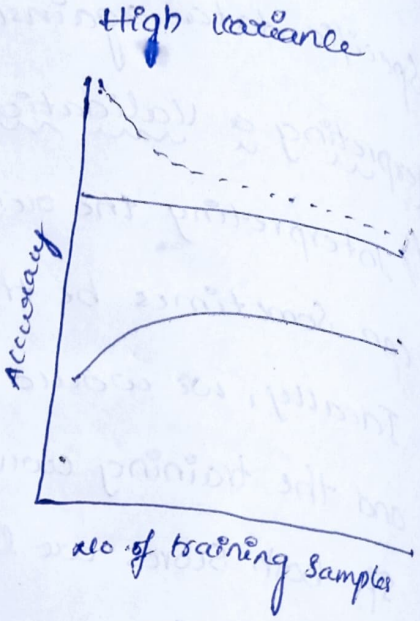
- Interpreting the results of a validation curve can sometimes be tricky.
- Ideally, we would want both the validation curve and the training curve to look as similar as possible.
- If both scores are low, the model is likely to be underfitting.
- It means either the model is too simple or it is informed by too few features.
- It could also be the case that the model is regularised too much.

+ If the training curve reaches high score relatively quickly and the validation curve is lagging behind, the model is overfitting.
- This means the model is very complex and there is too little data; or it could simply mean there is too little data.
- The value of the parameter where the training and validation curves are closest to each other.

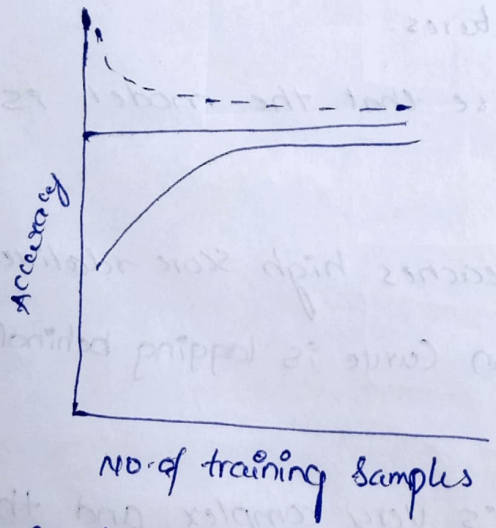
* Validation curve is a graphical technique that can be used to measure the influence of a single hyper parameter.
* The k-fold-cross validation is a technique to estimate the accuracy of the classifiers.

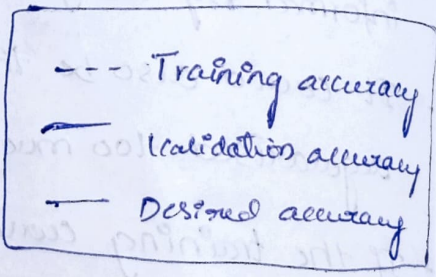
Ex: Diagnosing bias and variance problems with learning curves





Good bias-variance trade-off





* In the upper-left shows model with high bias. It has both low training & cross validation accuracy it goes underfitting.
* In the upper-right shows a model that suffers from high variance, which is indicated by the large gap between training and large gap between training and Cross validation accuracy it goes overfitting.

Fine-tuning machine learning models via grid search :-

* In machine learning, we have two types of parameters : those that are learned from the training data.
* For example, the weights in logistic regression, and the parameters of a learning algorithm that are optimized separately.
* The latter are the tuning parameters, also called as hyperparameters of a model.
* For example, the regularization parameter in logistic regression or the depth parameter of a decision tree.
* Validation curves to improve the performance of a model by tuning one of its hyperparameters. We will take a look at a popular hyperparameter technique called grid search.
* Grid search helps to improve the performance of a model by finding the optimal combination of those hyperparameter values.

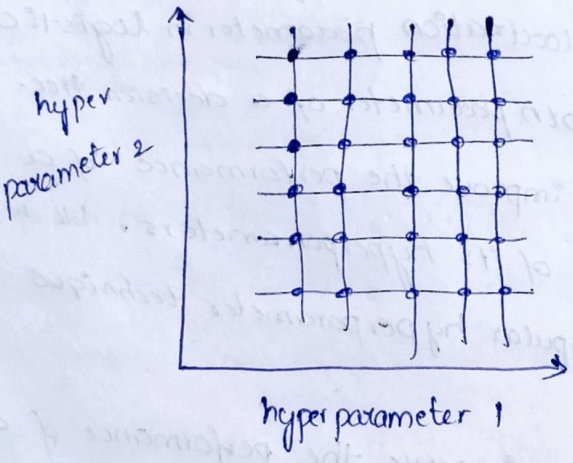
Tuning hyperparameters via grid search :-

- Hyper parameters are model parameters whose values are set before training.
- If the model has several parameters, we need to find the best combination of values of the hypaparameters searching in a multi dimensional space.

Grid search :-

- It is the simplest algorithm for hyperparameter using tuning.

- We divide the domain of the hyperparameters into discrete grid.
- The grid is calculated performance metrics using Cross-validation.
- The point of the grid that maximizes the average value in Cross validation, is the optimal combination of values for the hyper parameters.



- Grid search is an exhaustive algorithm that spans all the combinations, so it can actually find the best point in the domain.
- The draw back is very slow.
- Tuning the hyper parameters can be quite complex and expensive.

* Looking at different performance evaluation metrics

To evaluate the performance or quality of the model, different metrics are used and these metrics are known as performance metrics or evaluation metrics.

1. Performance Metrics for classification:-

The category or classes of data is identified

based on training data. The model learns from the given dataset and then classifies the new data into classes or group based on training.

performance of a classification model are :

- Accuracy
- confusion matrix
- precision
- Recall
- F-Score
- AUC - ROC.

1. Accuracy :- The accuracy metric is one of the simplest classification metrics to implement and it can be determined as the no. of correct predictions to the total no. of predicted.

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total no. of predictions}}$$

3(b) 2. Confusion Matrix :-

A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known.

The confusion matrix is simple to implement, but the terminologies used in this matrix might be confusing for beginners.

A typical confusion matrix for a binary classifier.

- It is divided into four terminologies :

1, True positive (TP) :- The prediction outcome is true, and it is true in reality, also.

2, True Negative (TN) :- The prediction outcome is false and it is false in reality also.

3, False positive (FP) :- The prediction outcomes are true, but they are in actuality.

4, False Negative (FN) :- The predictions are false and they are true in actuality.) & program

— precision :-

The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was actually correct.

$$\text{precision} = \frac{TP}{(TP+FP)}$$

- It is also similar to the precision matrix.

- Recall or Sensitivity :-

- It is also similar to the precision matrix.

- It aims to calculate the portion of actual positive that was identified incorrectly. It can be calculated as True positive or prediction. It is true positive and false negative.

$$\text{Recall} = \frac{TP}{TP+FN}.$$

1. Score :-

F-score or F1 score is a metric to evaluate a binary classification model on the basis of predictions that are made for the positive class. It is calculated with the help of precision and Recall. It is a type of single score that represents both precision and recall.

$$F_1\text{-score} = 2 * \frac{precision * recall}{precision + recall}$$

- AUC-ROC :-

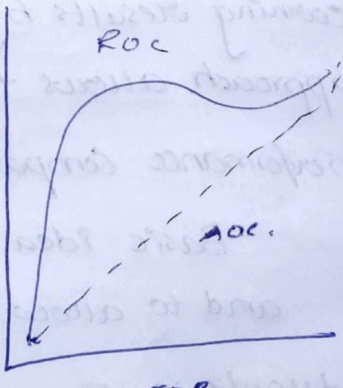
Some times we need to visualise the performance of the classification model on charts ; we can use the AUC-ROC curve. It is one of the popular and important metrics for evaluating the performance of the classification model.

ROC represents a graph to show the performance of a classification model at different threshold levels.

- True positive Rate (TPR)
- False positive Rate (FPR)

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{PF+TN}$$



-performance metrics for Regression:-

It is a supervised learning technique that aims to find the relationships between the dependent and independent variables.

Mean Absolute Error (MAE):- measures the absolute difference between actual and predicted value, absolute means taking a number as positive.

$$MAE = 1/N \sum |y - y'|.$$

Mean Squared Error :- one of the most suitable metrics for regression evaluation. measures the average difference between predicted values and the actual values.

$$MSE = 1/N \sum (y - y')^2$$

R squared score :- Also known as coefficient of determination which another popular metric used for regression model evaluation

$$R^2 = 1 - \frac{MSE\ (model)}{MSE\ (Baseline)}.$$

Adjusted R Squared :- It has limitation of improvement of a score on increasing the terms even through the model is not improving.

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1} \right) \times (1 - R^2) \right]$$

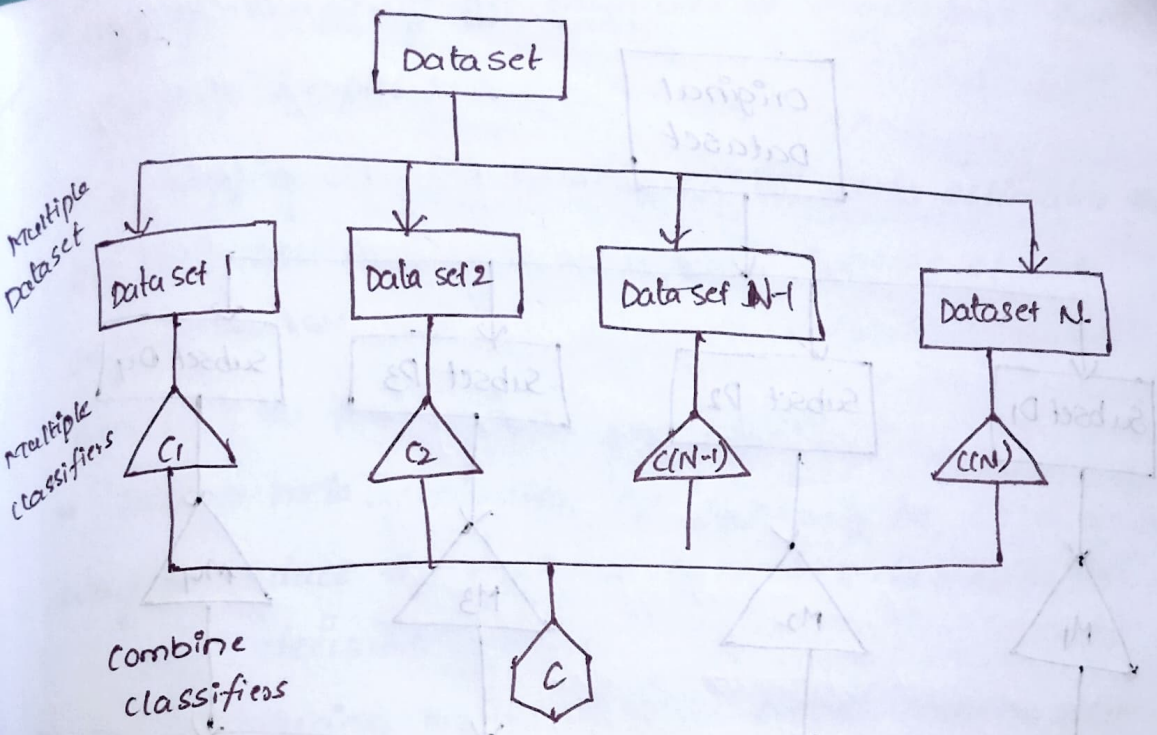
* Learning with ensembles :-

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model.

Basic idea is to learn a set of classifiers and to allow them to vote.

Advantage :- Improvement in predictive accuracy.

Disadvantage :- It is difficult to understand an ensembles of classifiers.



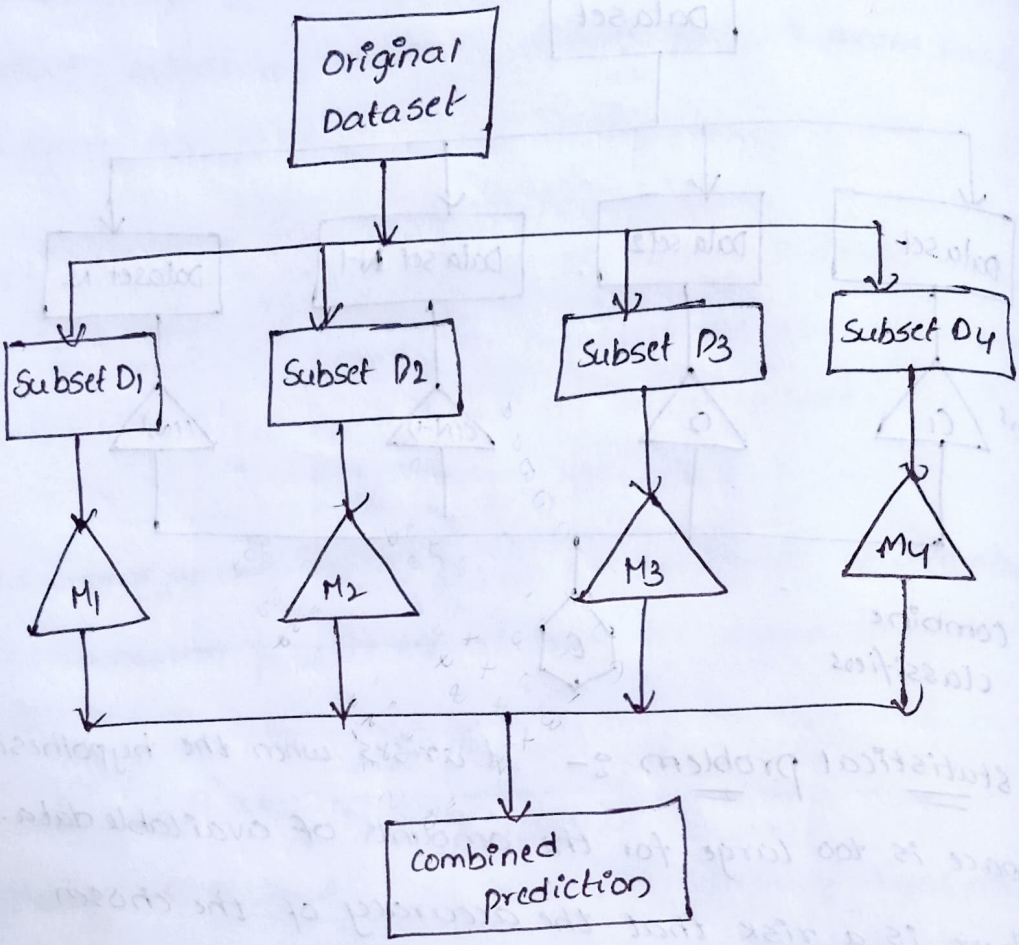
- Statistical problem :- It arises when the hypothesis space is too large for the amount of available data. There is a risk that the accuracy of the chosen hypothesis is low on unseen data.

Computational problem :- It arises when the learning algorithm cannot guarantees find the best hypothesis.

Representational problem :- It arises when the hypothesis Space does not contain any good approximation of the target class.

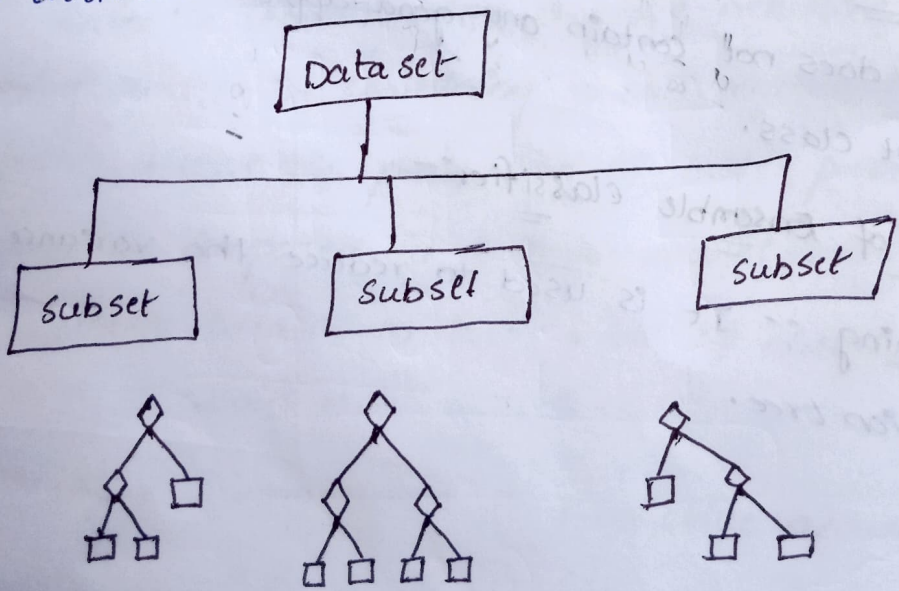
Types of Ensemble classifier :-

Bagging :- It is used to reduce the variance of a decision tree.



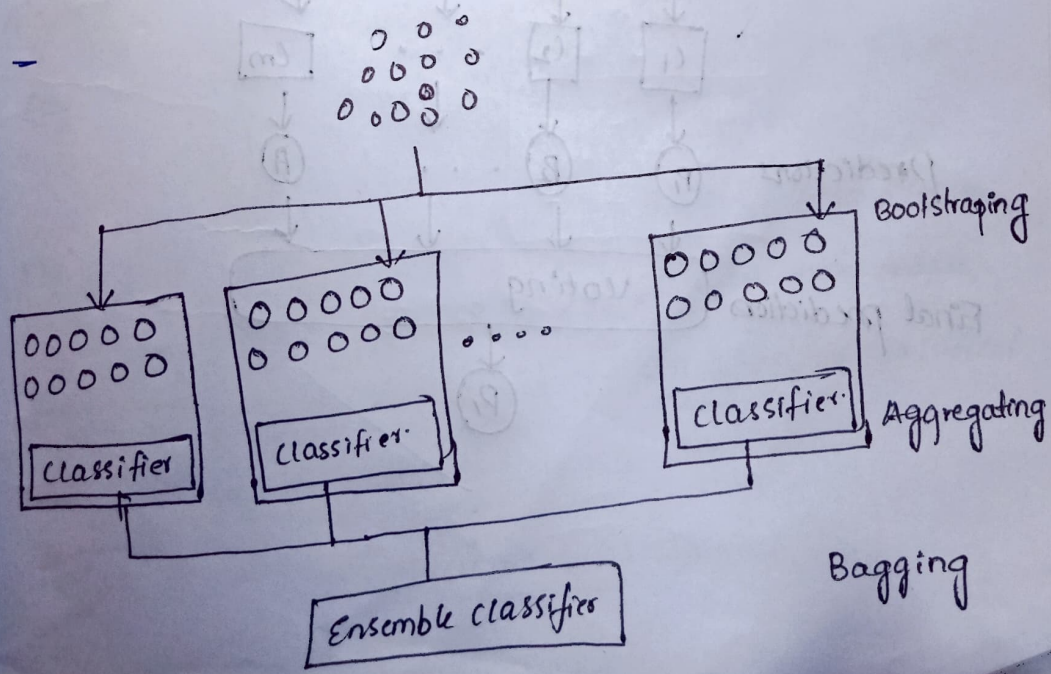
Random forest :-

- Extension over Bagging.
- Each classifier in the ensemble is a decision tree classifier and generated using random selection of attributes.



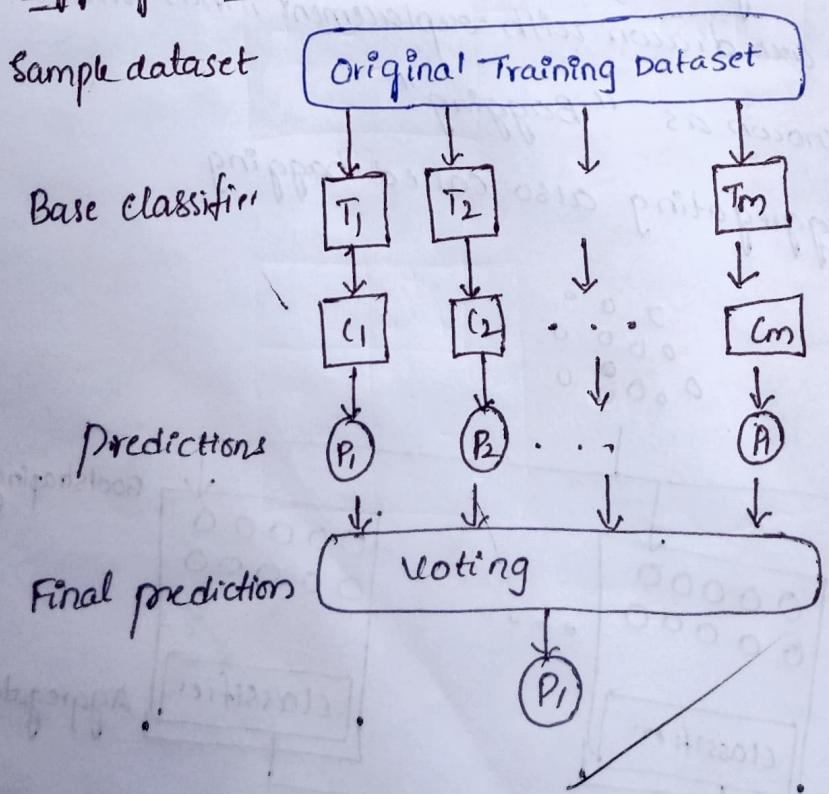
30) Bagging - building an ensembles of classifiers from bootstrap samples :-

* A bagging classifier is an ensemble meta estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual prediction to form a final prediction.
* Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator.
  Eg :- decision Tree.
* By introducing randomization into its construction procedure and then making an ensemble out of it.
* This algorithm encompasses several works from the literature.
* When random subsets of the dataset are drawn as random subsets of samples, then this algorithm is known as pasting.
* If samples are drawn with replacement, then the method is known as "Bagging".
* Bootstrap aggregating also called bagging



- Bagging leads to improvements for unstable procedu
- Bagging was shown to improve preimage learning.
- On the other hand it can midly degrade the performance of stable methods such as KNN.
- Each base classifier is trained in parallel with a training set which is generated by randomly drawing with replacement N.
- Training set for each of the base classifier is independent of each other.
- Many of the original data may be repeated in the resulting training set while others may be left out.
- Bagging reduces overfitting by averaging or voting, however this leads to an increase in bias

Bagging classifier :-



* leveraging weak learners via adaptive boosting :-

- Boosting is an ensemble modeleng technique that attempts to build a strong classifier from the number of weak classifiers.
- It is done by building a model by using weak model in series. Firstly model is built from the training data.
- Second model is built which tries to correct the errors present in first model.
- The procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum no. of models are added.
- "AdaBoost was the first really successful boosting algorithm developed for the purpose of binary tree classification.
- AdaBoost is short for Adaptive Boosting and is a very popular boosting technique that combines multiple "weak classifiers" into a single "strong classifier".

Algorithm :-

Step 1 : Initialize the dataset and assign equal weight to each of the data point.

step 2 : provide this is as input to the model and identify the wrongly classified data points.

Step 3 : Increase the weight of the wrongly classified data points.

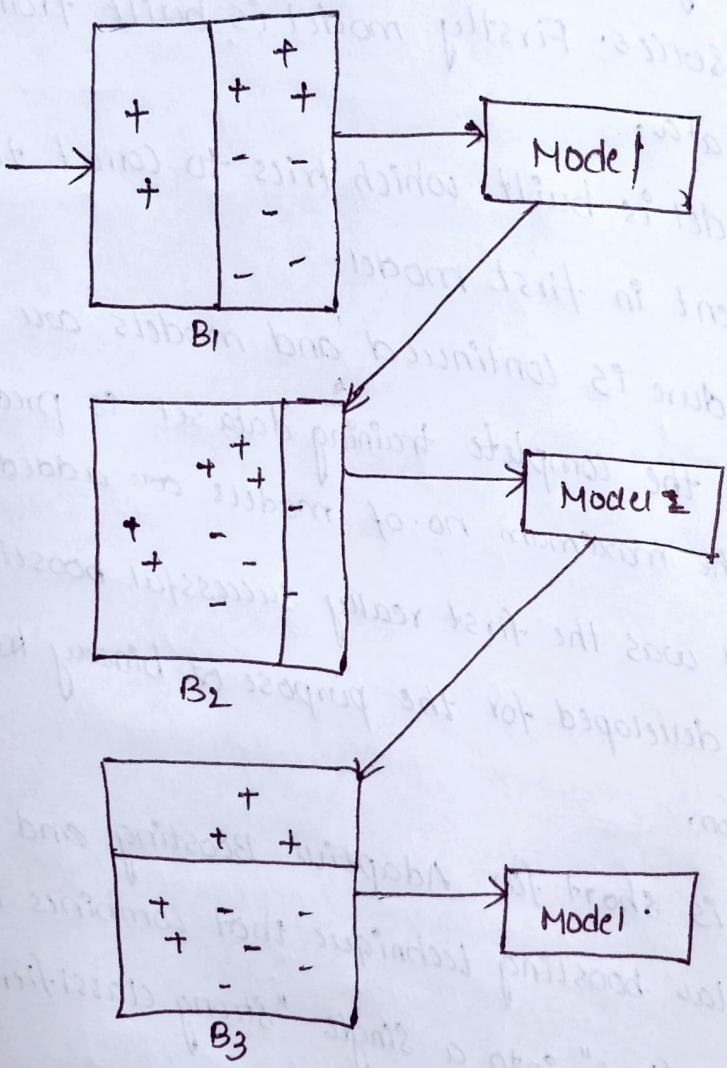
step 4 : if (got required results)

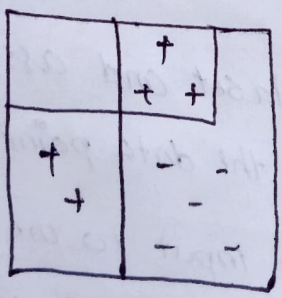
Go to Step 5

else

Go to Step 2.

Step 5 : End.





$B_4 = B_1 + B_2 + B_3$