Ensemble Model

Ques- What is Ensemble Model ?

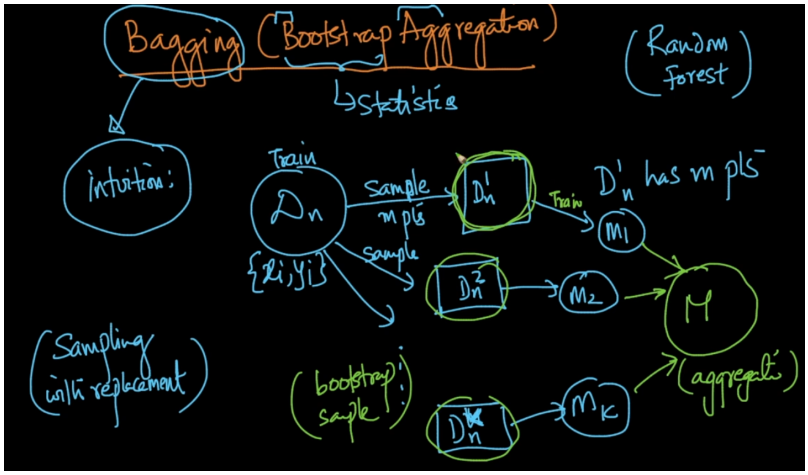
Ans- When multiple models used together to make more powerful model.

4 Types of Ensemble model are-

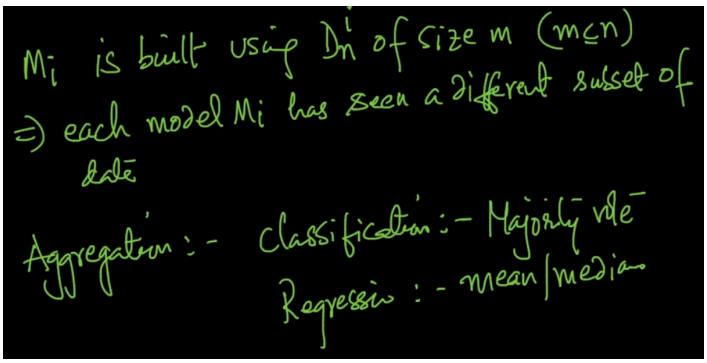
1. Bagging(Bootstrapped Aggregation)
2. Boosting
3. Stacking
4. Cascading

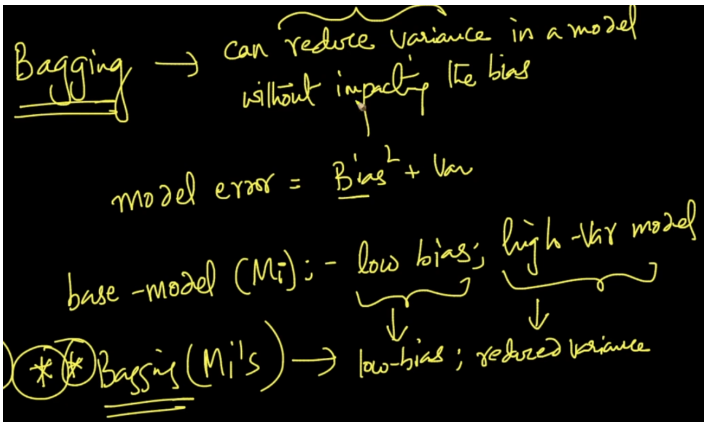
Bagging(Bootstrapped Aggregation)

- Bagging is used for both Classification and Regression.
- Here we take k sample from dataset and train k different model after that we aggregate them and select the majority one.
- **Sampling with replacement-** Take out m points from Dataset and that m points are deleted from our Dataset.

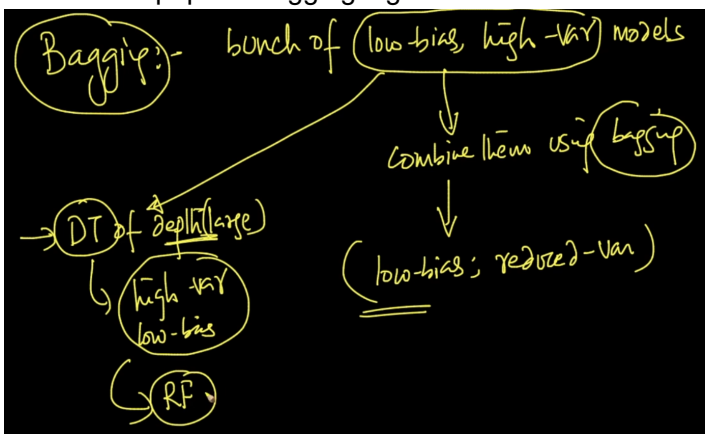


- In Bagging we trained sample with replacement with k model after getting k model we aggregate them which means for classification we pick the majority from models.





- If we have High variance and low bias models and combine them using Bagging then we get Low Bias and reduced variance.
- Example of High variance and low bias is Decision Tree with high depth.
- Random Forest is most used & popular bagging algo.



Random Forest

- Random forest is a supervised learning algorithm which is used for both classification as well as regression.
- It takes Decision trees as the base model for bagging and col sampling(Feature bagging).

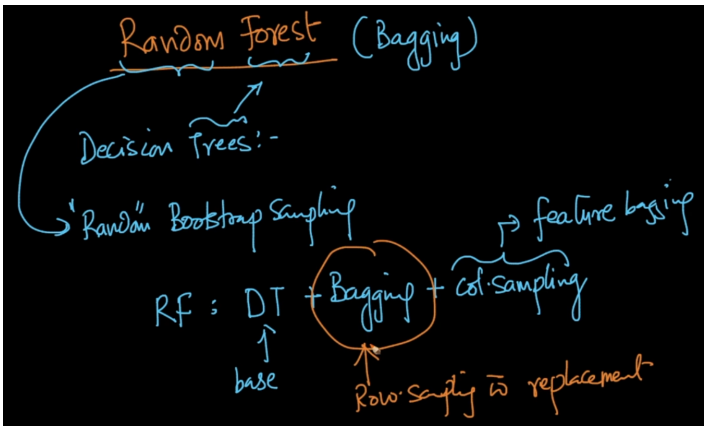
Random Forest = Decision Tree + Bagging + Column Sampling.

Step 1 − First, start with the selection of random samples from a given dataset.

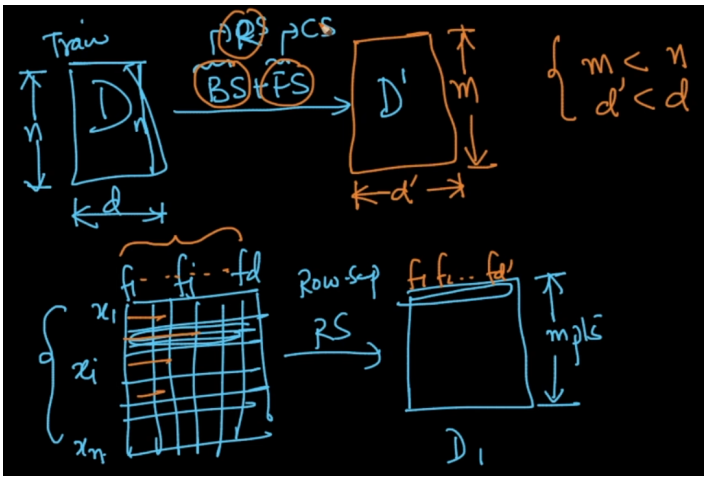
Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

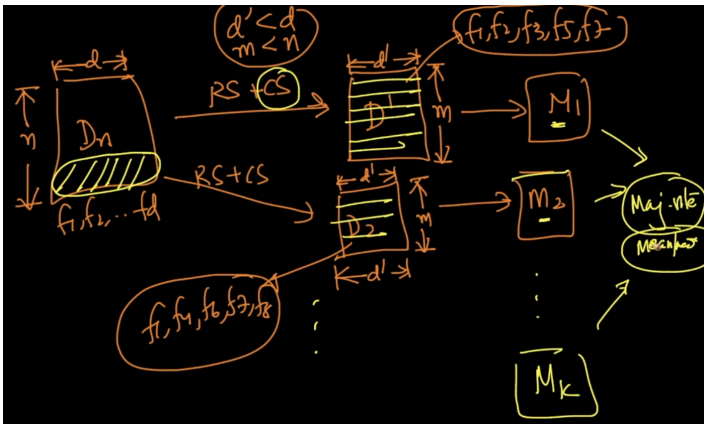


Implementation



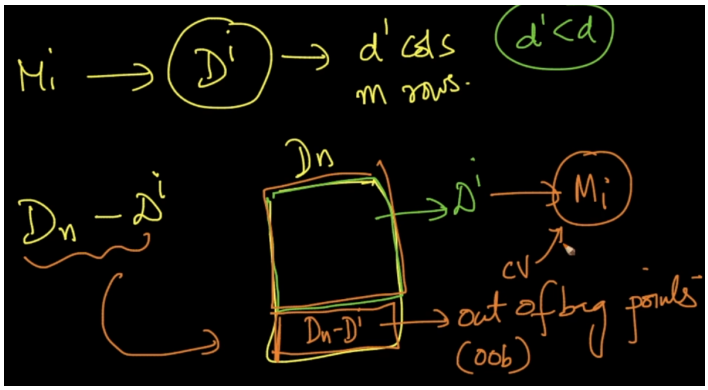
Explaination-

- The sampling of rows is done randomly with replacement.
- The selected sampled data set dimensions(or features) are also selected randomly.
- The model M1 is made on this data.
- The rows and columns are also sampled randomly in Random Forest.
- The above process is done for making several other models. The data sets will be more different than the bootstrap sample as the features selected are different.
- At the end we do majority vote.

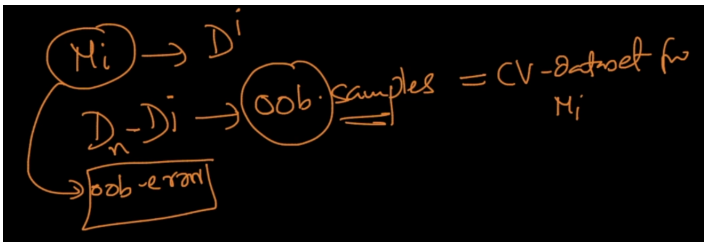


- The Mi's have no limit on the depth, making each models to over fit(high variance) does not impact the over all performance as the majority vote is done to reduce the high variance(over - fitting) of each models.

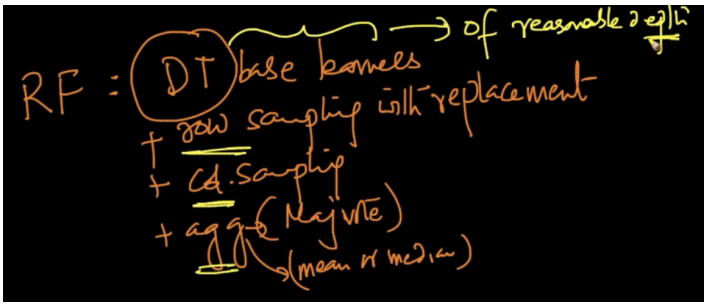
- The out of bag points are nothing but the points that are not included in the sample data for training the model.
- These OOB can be used for cross – validation



This error is called OOB error.



Summary -

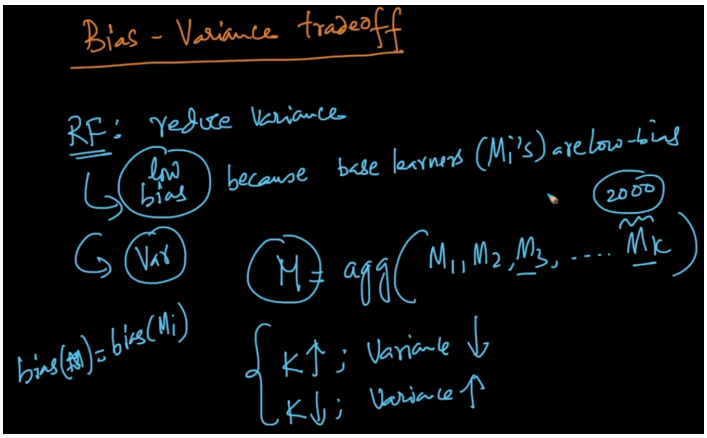


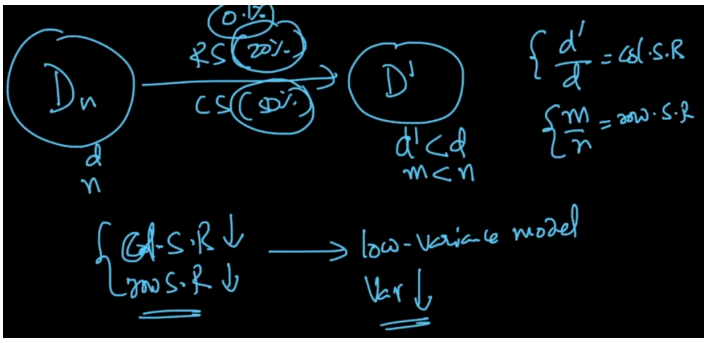
Advantage -

- It can handle large data set with higher dimensionality. It can handle thousands of input variables and identify most significant variables so it is considered as one of the dimensionality reduction methods.
- The model outputs Importance of variable, which can be a very handy feature (on some random data set).

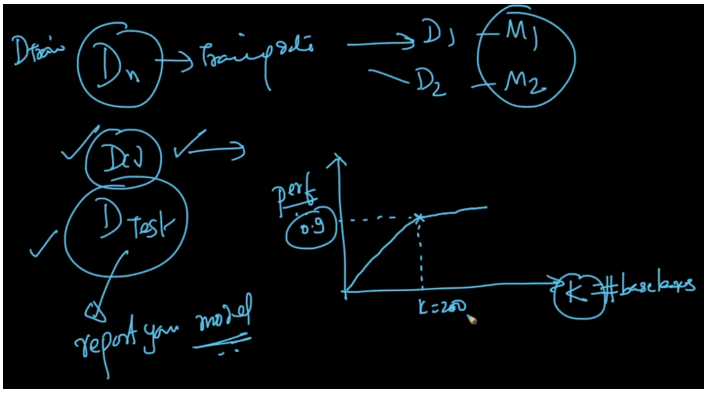
Bias Variance Tradeoff

- Random forest are said to have less bias because the base models i.e Decision Trees are low – bias.
- Variance is dependent on k, k is no. of decision tree to be trained.





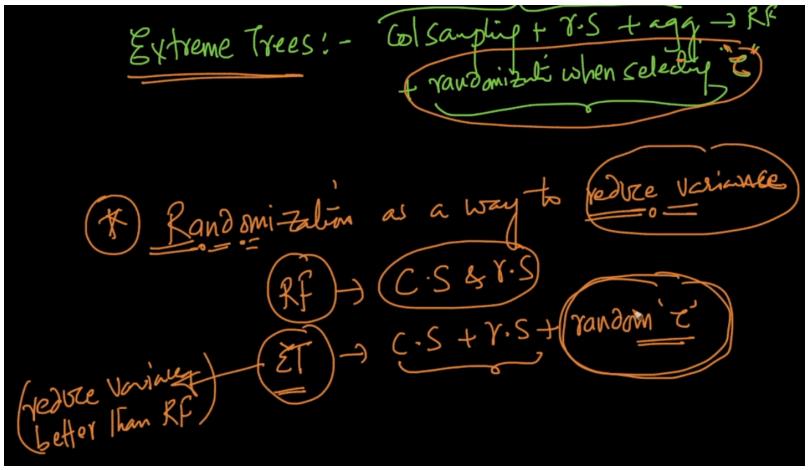
- Often times people tend to keep the column and row sampling rate constant and keep increasing the models.



- Hyper-parameter are -
  - k means no. of base learner i.e decision tree
  - Column Sampling Rate
  - Row Sampling Rate
- Cross – validation is used to choose the hyper parameter K(number of base learners).
- The number of base learners is the important hyper parameter than the row and column rate.

Extremely Randomized Tree

- In regular decision trees, we will try out all the values of the threshold and get the best possible value of the threshold for real – valued features, that increase the information gain.
- In randomized decision trees the some values of the bag are selected randomly and then we come up with a threshold. This is our final threshold of the base model.



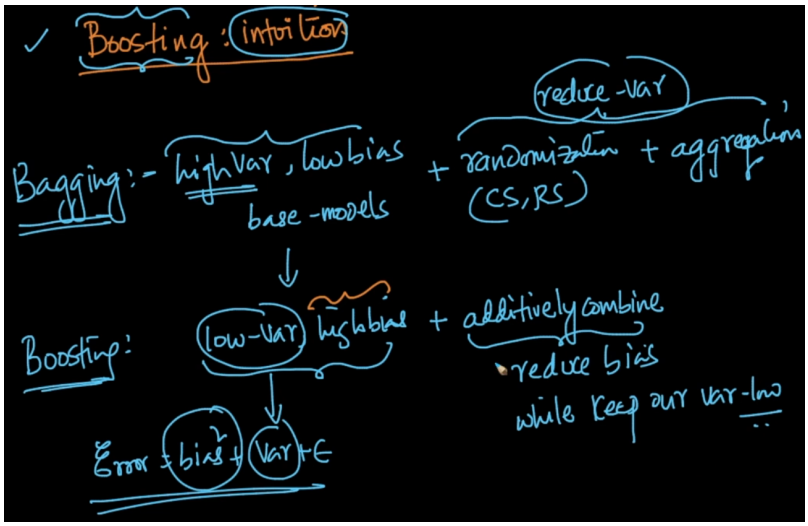
Random Tree Cases- DT vs RF

1. Random Forest = Decision Tree + Row Sampling + Column Sampling + aggregation. RS, CS and agg are done to reduce variance.
    - As Decision Tree is not work well large-Dimension, Categorical feature with many categories, one hot encoding then all such problem also exist in Random Forest bcz DT is the Base learner.
    - If DT fail to work well then RF also fail to work well.
    - But Bias-Vari tradeoff is not same as decision tree, bcz this depend on k, k is no. of base learner. We trained deep trees and we control the k but In DT bias-vari tradeoff work based on depth.
2. Feature Importance -
    - DT- Overall reduction in entropy or gini-entropy bcz of this feature at various level of DT only.
    - RF - Overall reduction in entropy or gini-entropy bcz of feature at various level of each of Mi's, here Mi is models.

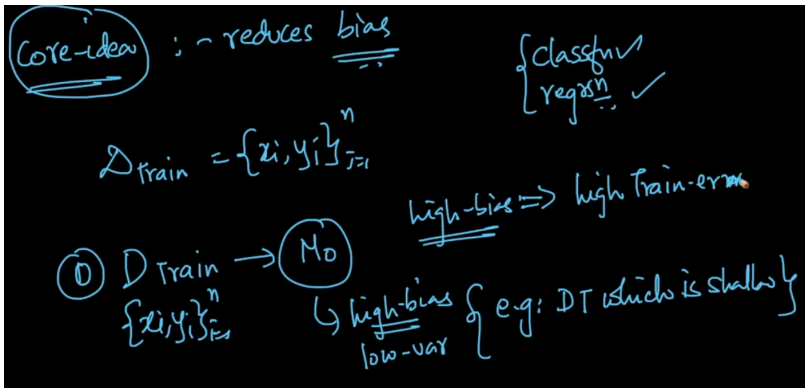
Boosting-

Situation - If a data point is incorrectly predicted by the first model, and then the next (probably all models), will combining the predictions provide better results? Such situations are taken care of by boosting.

Define - Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model. Instead of making high variance and low bias models in bagging, we use low variance and high bias models in Boosting with additive combine, that will reduce bias while keep out variance low.



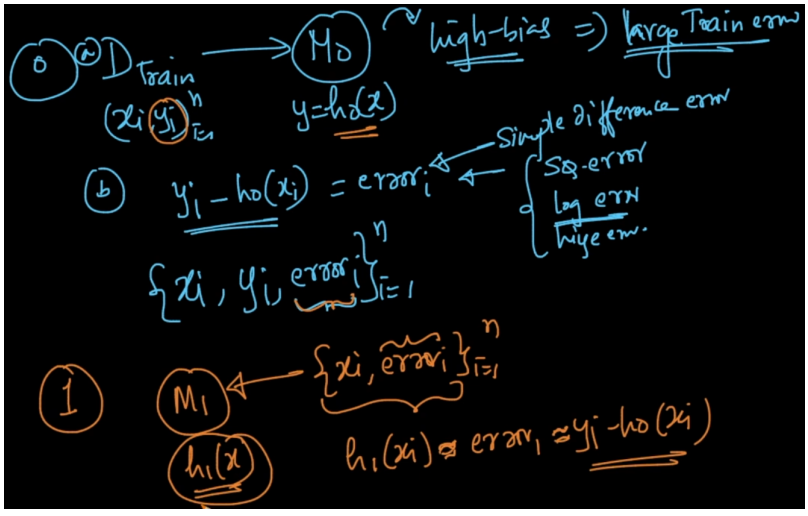
- High bias can be viewed as high training error.



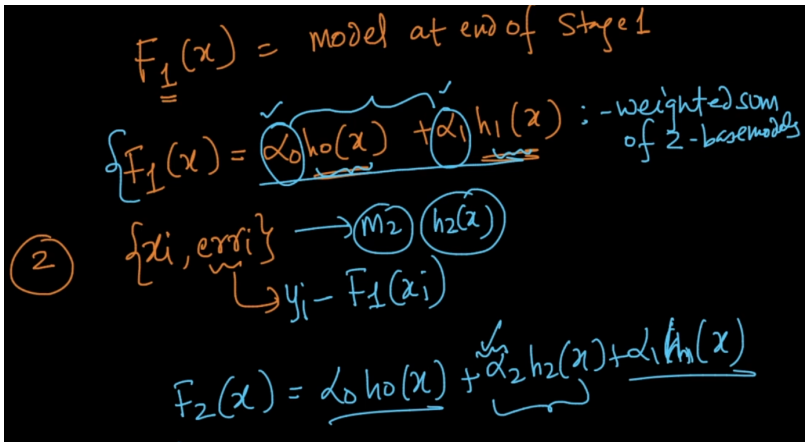
Step – 1: Train the model on all the data points. Make the model h(x).

Step – 2: Get the error on the points. Yi – h(xi). This is a simple error.

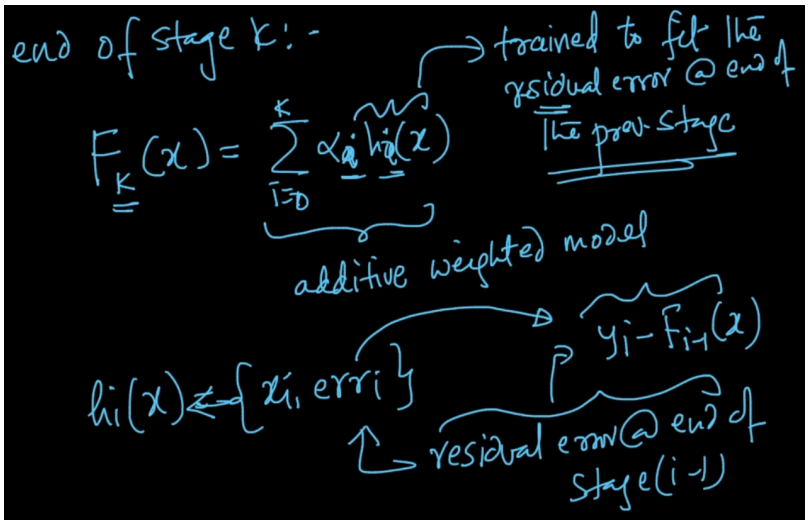
Step – 3: Train the model on the error that we got from the previous model.



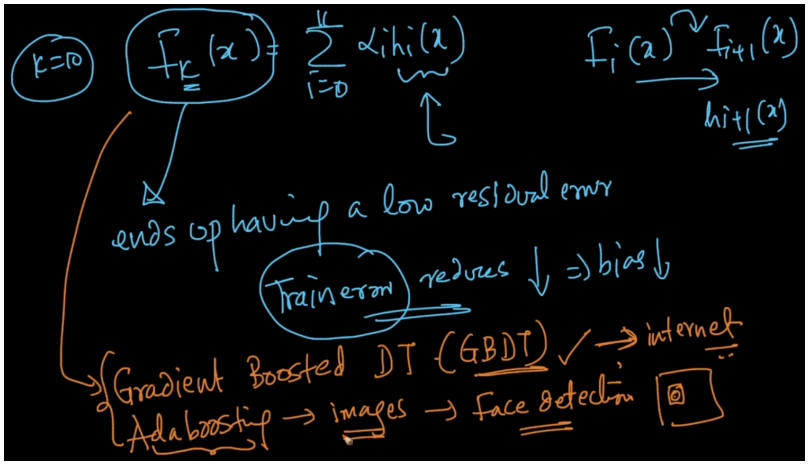
The models are made by additively combining the models.



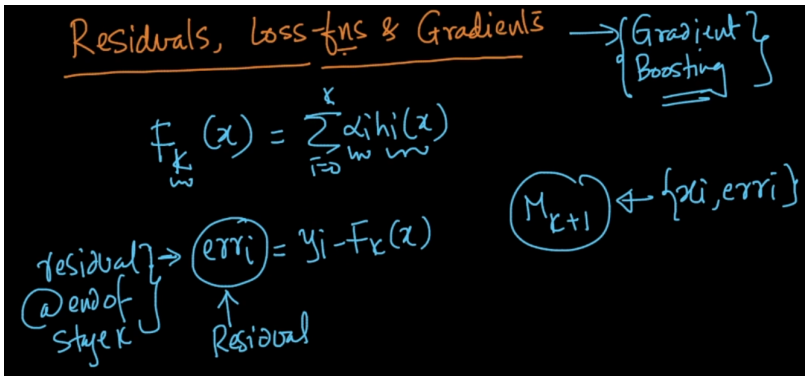
Generalized notation:



- The every model that is being constructed is to reduce the high error/bias. At every level the bias/error is reduced.

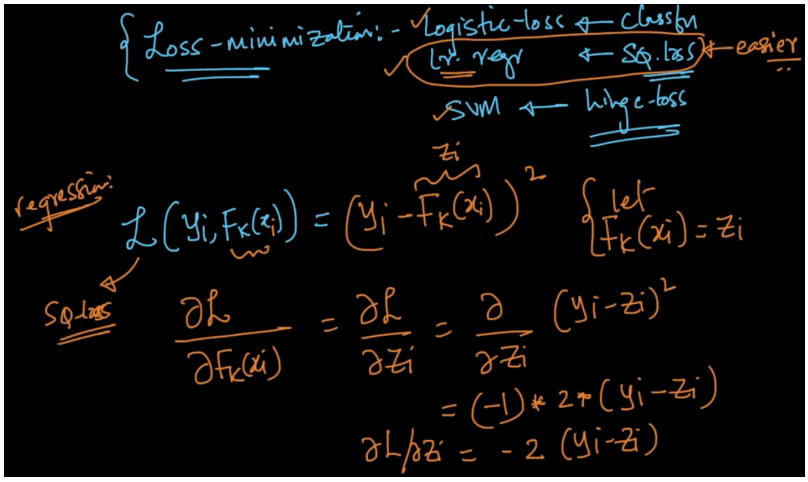


Residuals, loss-func and Gradients

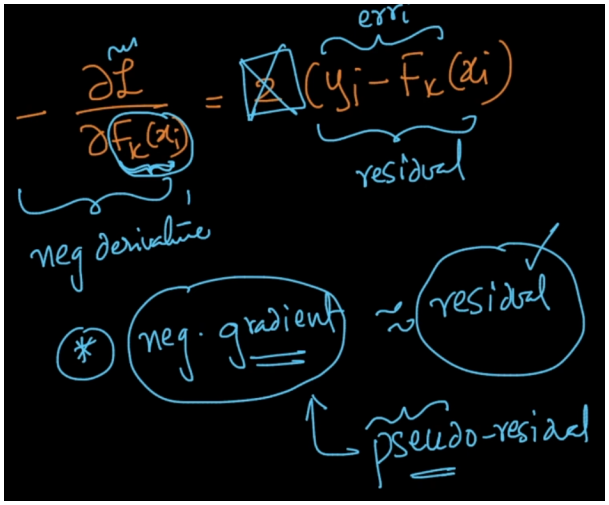


- Here the goal is to find the values and number of base learners. How to model the base learners.

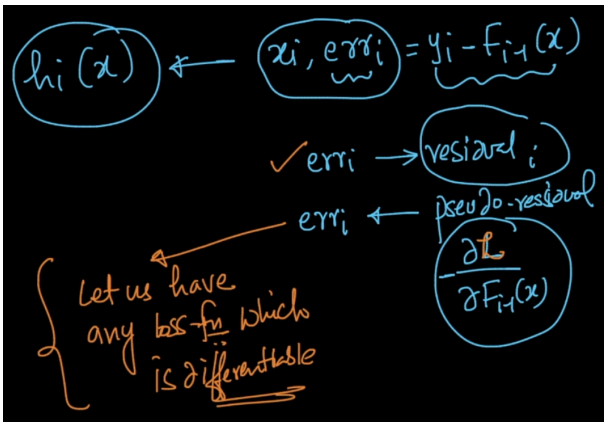
Loss Minimization - From One Base Learner



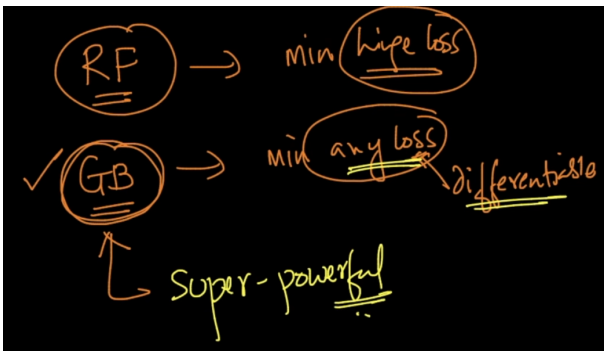
- At the end of the ‘K’ th base model.
- The error is called the residual error. The negative gradient can be taught like it is proportional to the residual error.



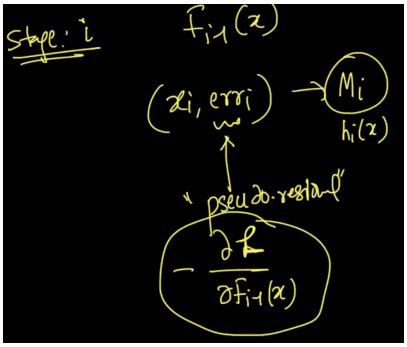
- This is often called as pseudo residuals.



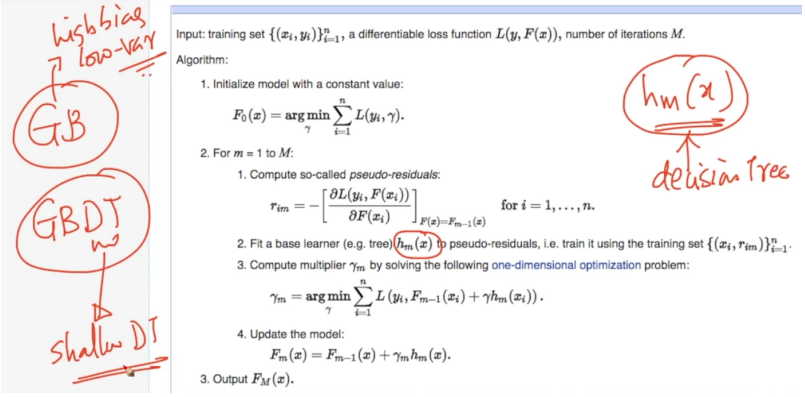
- Replacing the residuals with the pseudo residuals, will help in replacing with the any loss function which is differential.
- As long as the loss function is differential we can replace with any other loss function. This is the reason gradient boosting is so popular.



- At any stage ‘i’ we can take the pseudo residual of the actual error got from the previous level.



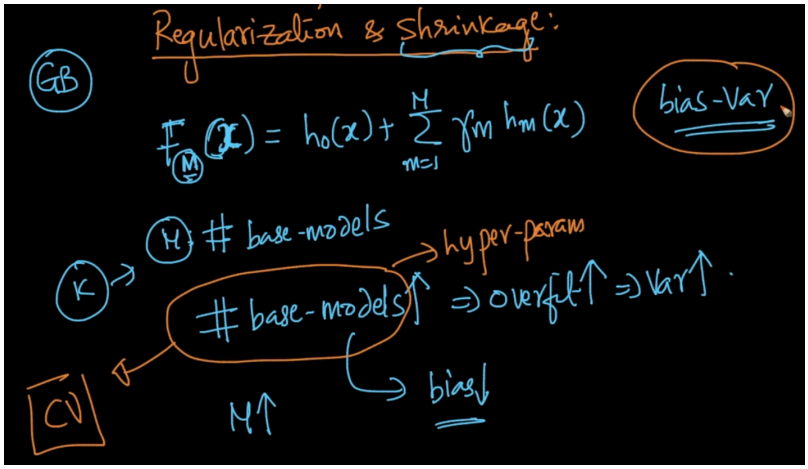
Gradient Boosting

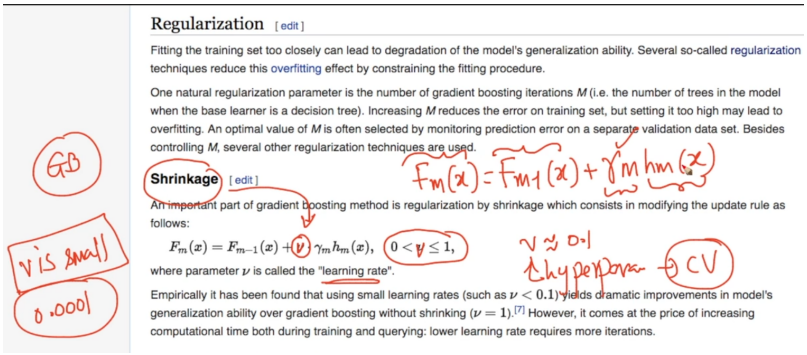


- Gradient boosting is the general idea. The each of the model must be high bias and low variance model. This can occur in case of decision trees with shallow depths.
- shallow depth means low depth.
- M is no. of Base model, same as k

Regularization & Shrinkage-

- As no. of base model increases it lead to Overfitting which result high Variance and low Bias.
- Here we can control Bias-Variance Trade-off.
- No. of Base model(M) calculated by CV.



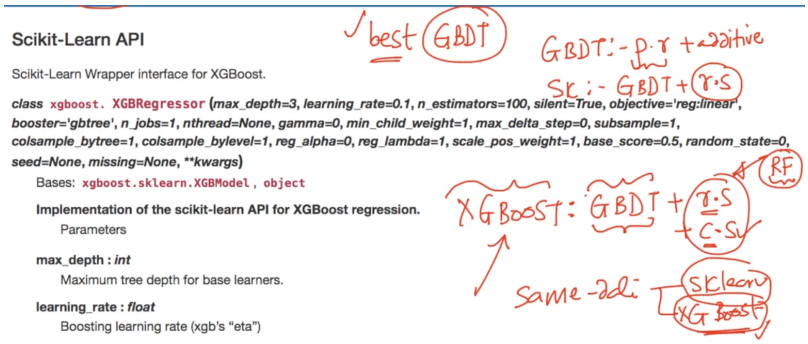


- Here the shrinkage can be tuned by using the variable ‘V’, which is called the learning rate. Lies between 0 and 1.
- Here ‘V’ and M are hyper parameter.
- If ‘V’ is very small then the chances of over fitting also reduces and variance also decreases.
- If ‘V’ increases then the chances of overfitting also increases.
- M is the number of base models.
- The ideal method is to find the best parameters using grid search.
- It is easy to over – fit for the gradient boosted trees by just incresing M.

XGBoost Boosting

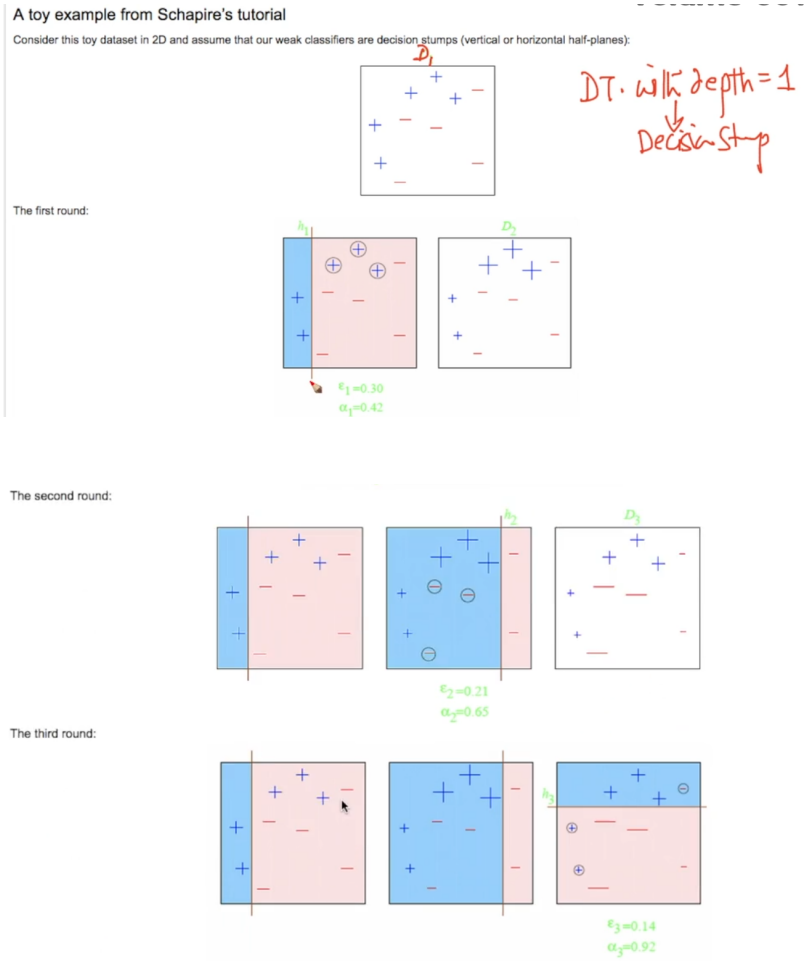
- sklearn implementation of Gradient Boosting is slow, so we use XGBoost Boosting algo.
- In XGBoost we use the GBDT + Row sampling + column sampling.

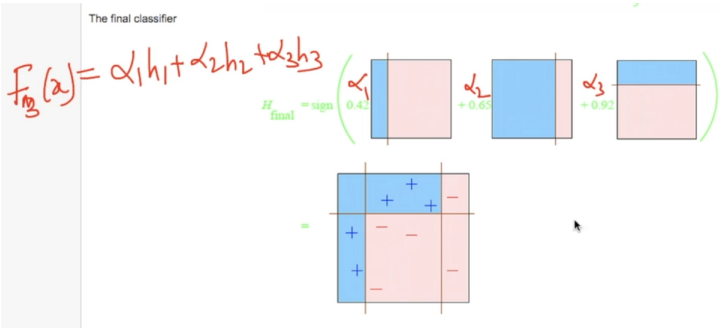
- XGBoost data sets perform well than the sklearn implementation due to several code optimizations.



AdaBoost -

- In AdaBoost we will getting errors in every poin of stage but we adapt them and increase the weight of the errors which is misclassified.

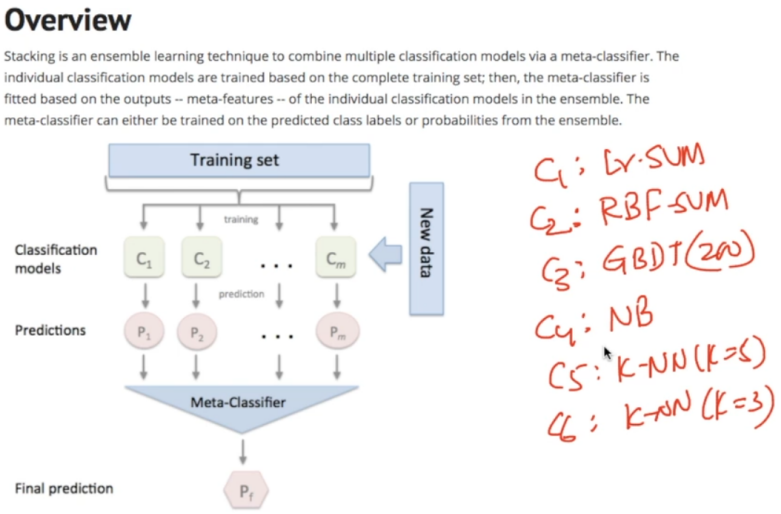




Stacking -

Refer as - http://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/ (http://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/)

- Stacking is an ensemble technique to combine multiple classification models via a meta-classifer.
- The base-level models are trained on a complete training set, then the meta-model is trained on the features that are outputs of the base-level model. The base-level often consists of different learning algorithms.
- If our models are different from one another it gives the best model.
- It mainly used in kaggle competition.
- It iused less than Boosting(GB) and Bagging(RF).
- It is not present in scikit-learn.



Implementation-

- In stacking the data is trained on several classification algorithms, the algos must be very different from each other. These are called the base models.(first – level classifiers). The more different the models are the better it works.
- Then again construct the data set on the predictions made by the base models for every point in the data set.
- Then on the new data set a classifier is trained to make the final output of the query point. The last classifier is called the meta classifier.

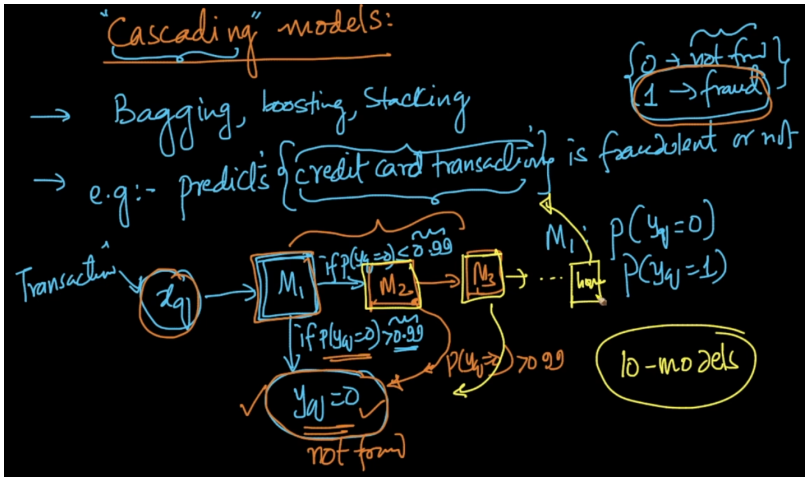
We can understand the process in the following steps -

1. We split the data into two parts viz, a training set and test set. The training data is further split into K-folds just like K-fold cross-validation.
2. A base model(e.g k-NN) is fitted on the K-1 parts and predictions are made for the Kth part.
3. This process is iterated until every fold has been predicted.
4. The base model is then fitted on the whole train data set to calculate its performance on the test set.
5. We repeat the last 3 steps for other base models.(e.g SVM,decision tree,neural network etc)
6. Predictions from the train set are used as features for the second level model.
7. Second level model is used to make a prediction on the test set.

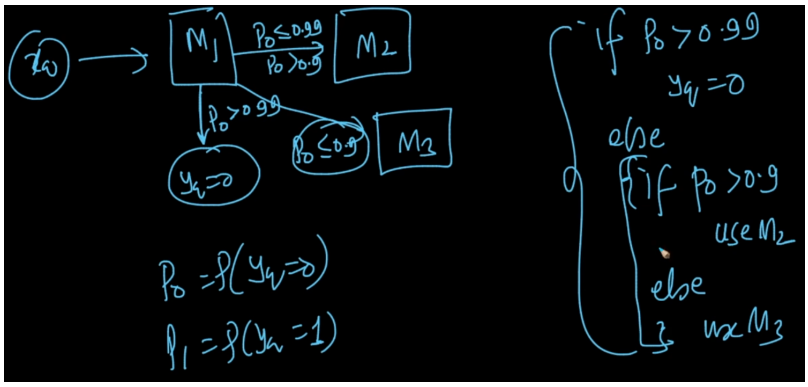
The outputs from the base models used as input to the meta-model may be real values in the case of regression, and probability values, probability like values, or class labels in the case of classification.

Cascading -

- Used when the cost of making a mistake is high.
- Example for understanding the cascading models. Every transaction is a vector, every fraud transaction can be done from various locations.
- Every model can give the class probabilities. (fraud and not – fraud). If probability of the point to be one class can be large and other class is low.
- At every stage we train an another classifier to check the certainty of the model.



- The points that don't give the accuracy that is being passed to the other model for achieving the accuracy and so on.. till there can be a specialist checking at the end.
- At every level the data points in the data set are reduced.
- The cascade can be more than one type.
- The down output of the model is considered to be the class of the model based on the probability of the model.(i.e, say the model outputs 0.99 as the probability score, then the model will confirm the class of the data point, the remaining points are further sent to the cascade model of the first model).



- Cascades models can be extensively used in fraud detection, medicine applications for maintaining the accuracy of the model.
- They can also be build up to 20 levels.
- If one transaction is lost then there will be huge loss

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