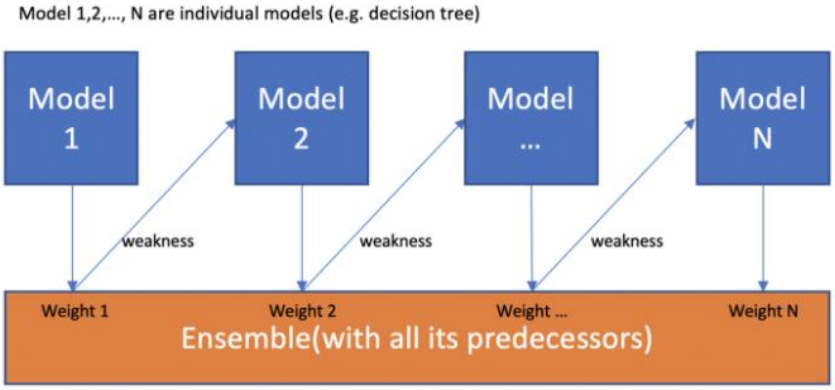
# Unit III Question Bank Answers

## 2 Marks

## Category – 1 (Easy):

**1. Name the commonly used Boosting algorithms?**

- **Boosting** is an ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones.


Model 1,2,..., N are individual models (e.g. decision tree)
Model 1
Model 2
Model ...
Model N
weakness
weakness
weakness
Weight 1
Weight 2
Weight ...
Weight N
Ensemble(with all its predecessors)

One is weak, together is strong, learning from past is the best

- To convert weak learner to strong learner following methods are used:
  1. Using average or weighted average.
  2. Consider prediction has a higher vote.
- Commonly used Boosting Algorithms:

  **1. AdaBoost (Adaptive Boosting):**

  In each iteration, AdaBoost identifies miss-classified data points, increasing their weights (and decrease the weights of correct points, in a sense) so that the next classifier will pay extra attention to get them.

  **2. Gradient Tree Boosting:**

  Gradient boosting algorithm is a machine learning technique to define loss function and reduce it. It is used to solve problems of classification using prediction models.

  **3. XGBoost**

  XG Boost is short for Extreme Gradient Boosting. XG Boost is upgraded implementation of Gradient Boosting Algorithm which is developed for high computational speed, scalability, and better performance.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**2. What are the different methods of ensembling?**

- Ensemble methods aim at improving predictability in models by combining several models to make one very reliable model. Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models.
- The most popular ensemble methods are **boosting, bagging, and stacking.**
  - ❖ **Bagging:**
  - ❖ It increases the accuracy of models through the use of decision trees, which reduces variance to a large extent. It is classified into two types, **i.e., bootstrapping and aggregation.**
    - ▪ **Bootstrapping** is a sampling technique where samples are derived from the whole population (set) using the replacement procedure.
    - ▪ **Aggregation** in bagging is done to incorporate all possible outcomes of the prediction and randomize the outcome.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

- **Boosting:** The technique combines several weak base learners to form one strong learner. Boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models.
- **Stacking:** This technique works by allowing a training algorithm to ensemble several other similar learning algorithm predictions.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

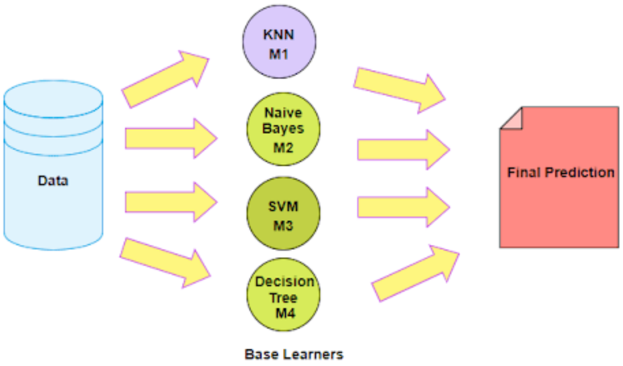
**3. What is averaging method in ensembling?**

- Model averaging is the simplest form of ensemble learning. Multiple models are trained on the same dataset and during prediction, and we take the average over multiple models.
- In the case of classification, the most common method for combining the predictions is to take votes from each model.
- For regression problems, we take the mean of the predictions from each model.
- This improves performance on the overall task by reducing overfitting.
- Frequently an ensemble of models performs better than any individual model, because the various errors of the models "average out."
- A limitation of this approach is that each model has an equal contribution to the final prediction made by the ensemble. Some models are known to perform much better or much worse than other models. This can result in fluctuations.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

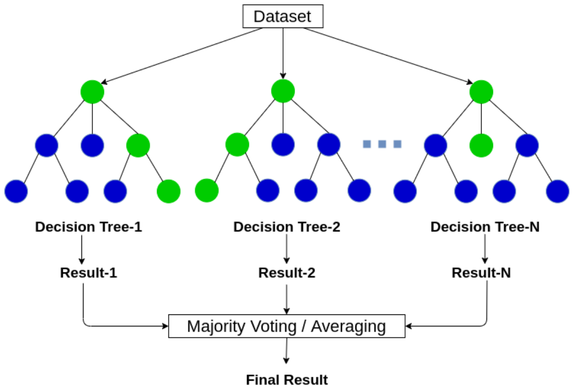
**4. Is Random Forest algorithm an ensemble method? Justify.**

- An Ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.



Ensemble Learning Method

- Random forest is a supervised ensemble learning algorithm as it combines the output of multiple (randomly created) Decision Trees to generate the final output.
- It is used for both classifications as well as regression problems. However, it is mainly used for classification problems.
- It is better than a single decision tree because it reduces the over-fitting by averaging the result.



**5. What is the package in SKlearn to use ensemble methods? Write the some popular ensemble functions.**

**Bagging:**

In scikit-learn, bagging methods are offered as a unified BaggingClassifier meta-estimator.

```
>>> from sklearn.ensemble import BaggingClassifier
>>> from sklearn.neighbors import KNeighborsClassifier
>>> bagging = BaggingClassifier (KNeighborsClassifier (), max_samples=0.5, max_features=0.5)
```

**Forests of randomized trees:**

The sklearn.ensemble module includes two averaging algorithms based on randomized decision trees: the RandomForest algorithm and the Extra-Trees method. Both algorithms are perturb-and-combine techniques specifically designed for trees.

**AdaBoost:**

AdaBoost can be used both for classification and regression problems:

For multi-class classification, **AdaBoostClassifier**.

For regression, **AdaBoostRegressor**

**Gradient Boosting:**

GradientBoostingClassifier supports both binary and multi-class classification.

GradientBoostingRegressor supports a number of different loss functions for regression which can be specified via the argument loss; the default loss function for regression is least squares ('ls').

## Category – 2 (Moderate)

**1. Does ensemble method give better performance? Justify.**

**(Study the definition and diagram of ensemble method from category-1 question 4)**

**Ensemble method improves robustness and performance:**

Fitting the model multiple times on the training datasets and combining the predictions using a summary statistic, such as the mean for regression or the mode for classification reduces the spread in the predictions made by the model. The mean performance will probably be about the same, although the worst- and best-case performance will be brought closer to the mean performance. In effect, it smooths out the expected performance of the model.

**2. What is the necessity of hyper parameter tuning?**

- **Hyperparameters:** A hyperparameter is a parameter that determines model architecture and cannot be learned from or estimated by training the data. They are externally given and are to be tuned in order to build a model to obtain the most optimal performance.
- **Hyperparameters are not model parameters.** Model parameters are learned during training
- The process of searching for the most optimal value of a hyperparameter is known as **hyperparameter tuning**. Using optimal values to build a model results in a better architecture of the model and improved model accuracy.

**3. Differentiate between hard voting and soft voting?**

- In **hard voting** (also known as majority voting), every individual classifier votes for a class, and the majority wins. Predict the class with the largest sum of votes from models
- In **soft voting**, every individual classifier provides a probability value that a specific data point belongs to a particular target class. Predict the class with the largest summed probability from models.

**4. Differentiate between bagging and boosting?**

- Bagging is a method of merging the same type of predictions. Boosting is a method of merging different types of predictions.
- Bagging decreases variance, not bias, and solves over-fitting issues in a model. Boosting decreases bias, not variance.
- In Bagging, each model receives an equal weight. In Boosting, models are weighed based on their performance.
- Models are built independently in Bagging. New models are affected by a previously built model's performance in Boosting.

**5. Differentiate between averaging and weighted averaging in ensembling.**

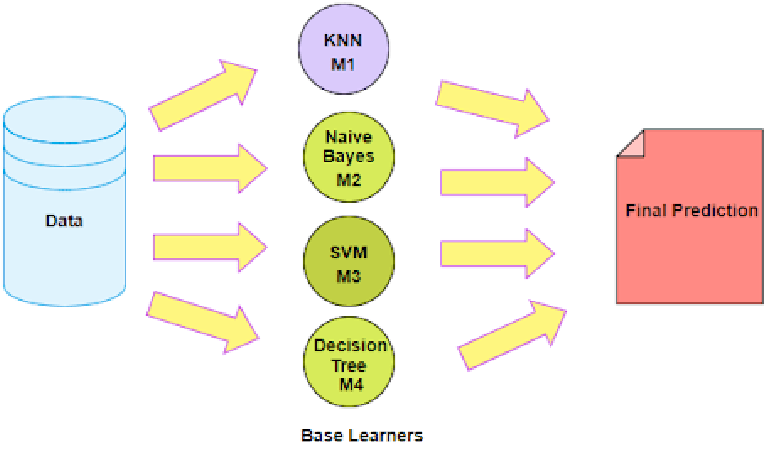
- **Model averaging** is an approach to ensemble learning where each ensemble member contributes an equal amount to the final prediction. In the case of regression, the ensemble prediction is calculated as the average of the member predictions. In the case of predicting a class label, the prediction is calculated as the mode of the member predictions.
- **A weighted ensemble** is an extension of a model averaging ensemble where the contribution of each member to the final prediction is weighted by the performance of the model. The model weights are small positive values and the sum of all weights equals one, allowing the weights to indicate the percentage of trust or expected performance from each model.

## 8 Marks:

**1. Broadly categorize ensemble methods with their suitability to different applications?**

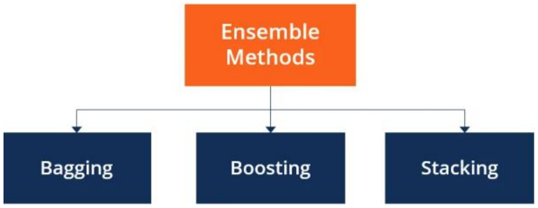
**Ensemble Methods:**

- Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly.



Ensemble Learning Method

- The most popular ensemble methods are boosting, bagging, and stacking.



- Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models.
- Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models.

**Categories of Ensemble Methods:**

Ensemble methods fall into two broad categories:

- **Sequential ensemble techniques** generate base learners in a sequence. The performance of the model is then improved by assigning higher weights to previously misrepresented learners.
- In **parallel ensemble techniques**, base learners are generated in a parallel format. Parallel methods utilize the parallel generation of base learners to encourage independence between the base learners. It significantly reduces the error due to the application of averages.

**Main Types of Ensemble Methods:**

- **Bagging:**
  - Mainly applied in classification and regression.
  - It increases the accuracy of models through the use of decision trees, which reduces variance.
  - The reduction of variance increases accuracy, hence eliminating overfitting
  - Bagging is classified into two types:
    - **Bootstrapping** is a sampling technique where samples are derived from the whole population (set) using the replacement procedure. The sampling with replacement method helps make the selection procedure randomized.

        - **Aggregation** in bagging is done to incorporate all possible outcomes of the prediction and randomize the outcome. Without aggregation, predictions will not be accurate, because all outcomes are not put into consideration.
    - Bagging is advantageous since weak base learners are combined to form a single strong learner that is more stable than single learners.
    - One limitation of bagging is that it is computationally expensive.
- **Boosting:**
    - Boosting is an ensemble technique that learns from previous predictor mistakes to make better predictions in the future.
    - Boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models.
    - Boosting takes many forms, which include gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost (Extreme Gradient Boosting).
    - **AdaBoost** makes use of weak learners that are in the form of decision trees.
    - **Gradient boosting** adds predictors sequentially to the ensemble.
    - **XGBoost** makes use of decision trees with boosted gradient, providing improved speed and performance.

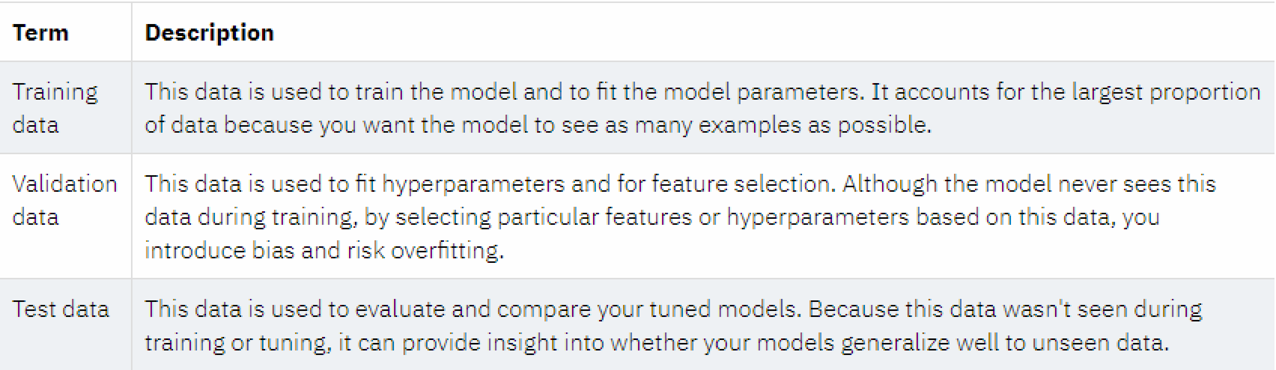
- **Stacking:**
    - Stacking, another ensemble method is often referred to as stacked generalization. This technique works by allowing a training algorithm to ensemble several other similar learning algorithm predictions. Stacking has been successfully implemented in regression, density estimations, distance learning, and classifications. It can also be used to measure the error rate involved during bagging.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**4. What are the different measures for evaluating a machine learning model?**

**<u>Model Evaluation:</u>**

- Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data.
- To properly evaluate a model, it should not be trained on the entire dataset.
- A typical train/test split would be to use **70% of the data for training and 30% of the data for testing.**
- To **evaluate the model** while still building and tuning the model, we create a third subset of the data known as **the validation set.**



| Term | Description |
|---|---|
| Training data | This data is used to train the model and to fit the model parameters. It accounts for the largest proportion of data because you want the model to see as many examples as possible. |
| Validation data | This data is used to fit hyperparameters and for feature selection. Although the model never sees this data during training, by selecting particular features or hyperparameters based on this data, you introduce bias and risk overfitting. |
| Test data | This data is used to evaluate and compare your tuned models. Because this data wasn't seen during training or tuning, it can provide insight into whether your models generalize well to unseen data. |

- A typical train/test/validation split would be to **use 60% of the data for training, 20% of the data for validation, and 20% of the data for testing.**
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

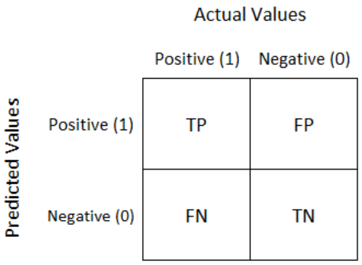
## Metrics:

### Classification metrics:

- When performing classification predictions, there are four types of outcomes that could occur.
- **True positives:** When you predict an observation belongs to a class and it actually does belong to that class.
- **True negatives:** When you predict an observation does not belong to a class and it actually does not belong to that class.
- **False positives:** When you predict an observation belongs to a class when in reality it does not.
- **False negatives:** When you predict an observation does not belong to a class when in fact it does.

### Confusion Matrix:

- A confusion matrix is a table that is often used to **describe the performance of a classification model.**



| | | Actual Values | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

**Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

**Precision**: Out of all the positive classes we have predicted correctly, how many are actually positive.

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

**Recall**: Out of all the positive classes, how much we predicted correctly. It should be high as possible.

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

**F-measure:** It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score.

### Regression metrics:

**Explained variance** compares the variance within the expected outcomes, and compares that to the variance in the error of our model.

$$EV\left(y_{true}, y_{pred}\right) = 1 - \frac{Var\left(y_{true} - y_{pred}\right)}{y_{true}}$$

**Mean squared error** is simply defined as the average of squared differences between the predicted output and the true output.

$$\text{MSE}\left(y_{true}, y_{pred}\right) = \frac{1}{n_{samples}} \sum \left(y_{true} - y_{pred}\right)^2$$

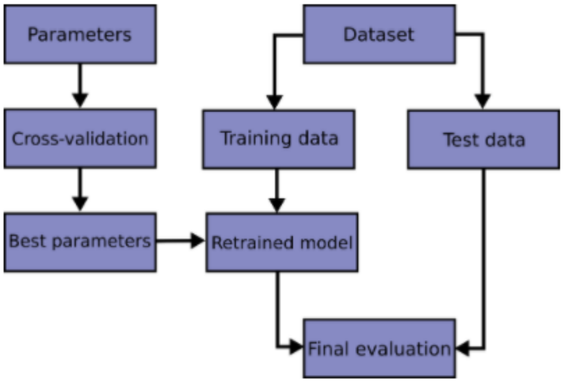
The **$R^2$ coefficient** represents the proportion of variance in the outcome that our model is capable of predicting based on its features.

$$R^2\left(y_{true}, y_{pred}\right) = 1 - \frac{\sum \left(y_{true} - y_{pred}\right)^2}{\sum \left(y_{true} - \bar{y}\right)^2}$$

$$\bar{y} = \frac{1}{n_{samples}} \sum y_{true}$$

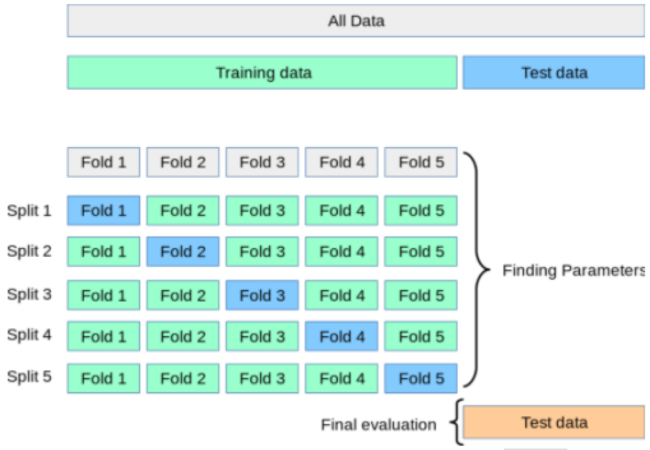
**2. What is cross validation dataset? When do you use it?**

- **Overfitting:**
  - Machine learning algorithms learn from examples. Overfitting is where a model can accurately make predictions for data that it was trained on but can't generalize to other data.
  - Overfitting is why you split your data into training data, validation data, and test data.
- Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
- The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.
- Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data.
- It is not used during the training of the model.
- It results in a less biased or less optimistic estimate of the model.



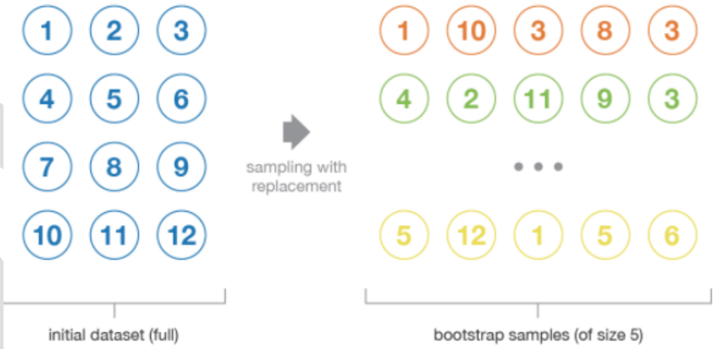
The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores.



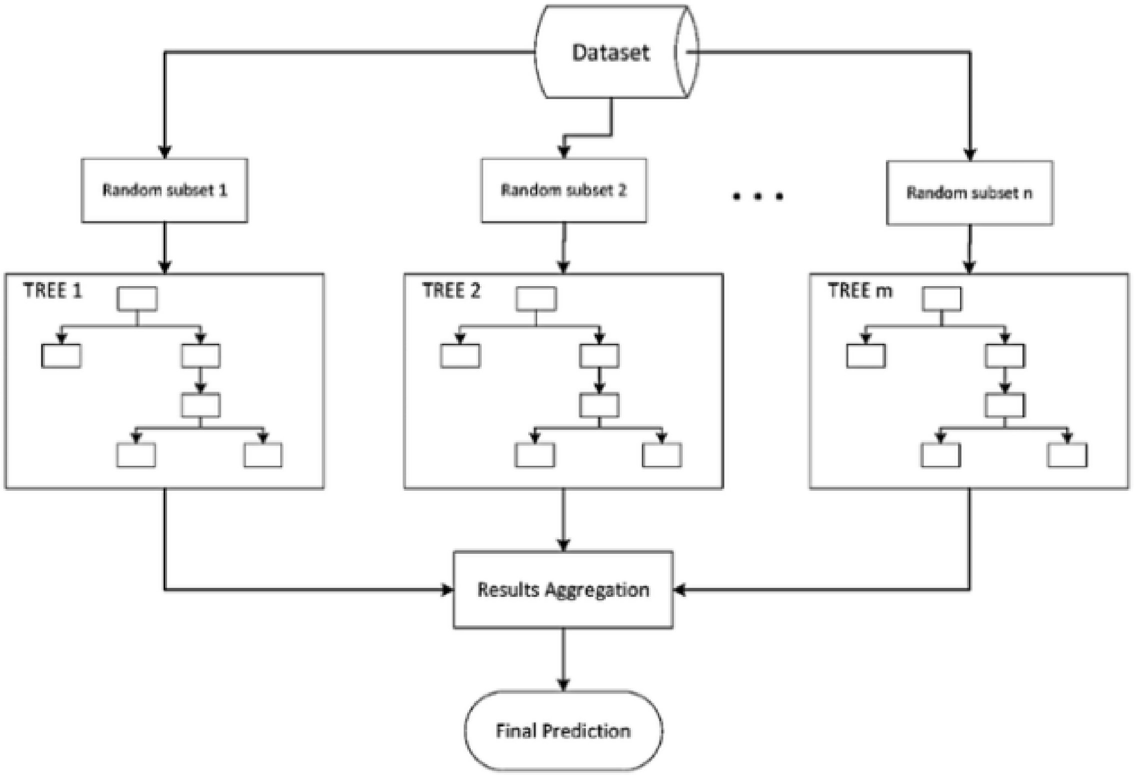
**3. When is bagging process preferred? Write the steps involved in it.**

- **Bagging:**
  - Bagging, also known as **bootstrap aggregating**, is the aggregation of multiple versions of a predicted model. Each model is trained individually, and combined using an averaging process. The primary focus of bagging is to achieve less variance than any model has individually.
  - **Bootstrapping:** It is the process of generating bootstrapped samples from the given dataset. The samples are formulated by randomly drawing the data points with replacement.



**Steps for Bagging:**

1. In Bagging, the bootstrapped samples are first created.
2. Then, either a regression or classification algorithm is applied to each sample.
3. Finally, in the case of regression, an average is taken over all the outputs predicted by the individual learners.
4. For classification either the most voted class is accepted (**hard-voting**), or the highest average of all the class probabilities is taken as the output (**soft-voting**).



**Mathematically, Bagging is represented by the following formula:**

$$\widehat{f_{bag}} = \widehat{f_1}(X) + \widehat{f_2}(X) + \cdots + \widehat{f_b}(X)$$

The term on the left hand side is the bagged prediction, and terms on the right hand side are the individual learners.

**When is Bagging used?**

Bagging works especially well when the learners are unstable and tend to overfit, i.e. **small changes in the training data lead to major changes in the predicted output**. It effectively reduces the variance by aggregating the individual learners composed of different statistical properties.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**Moderate Questions:**

**1. Why should variance problems be dealt with? What are the methods to overcome it?**

- A final machine learning model is one that is trained on all available data and is then used to make predictions on new data.
- A problem with most final models is that they suffer from variance in their predictions.
- **Variance** is the amount that the estimate of the target function will change if different training data was used.
- Each time a model is fitted, a slightly different set of parameters are obtained that in turn make slightly different predictions. Sometimes more and sometimes less skillful than what you expected.
- It is required to use the model to make predictions on real data where the answer is not known and the predictions are to be as good as possible.

**How to resolve variance problem?**

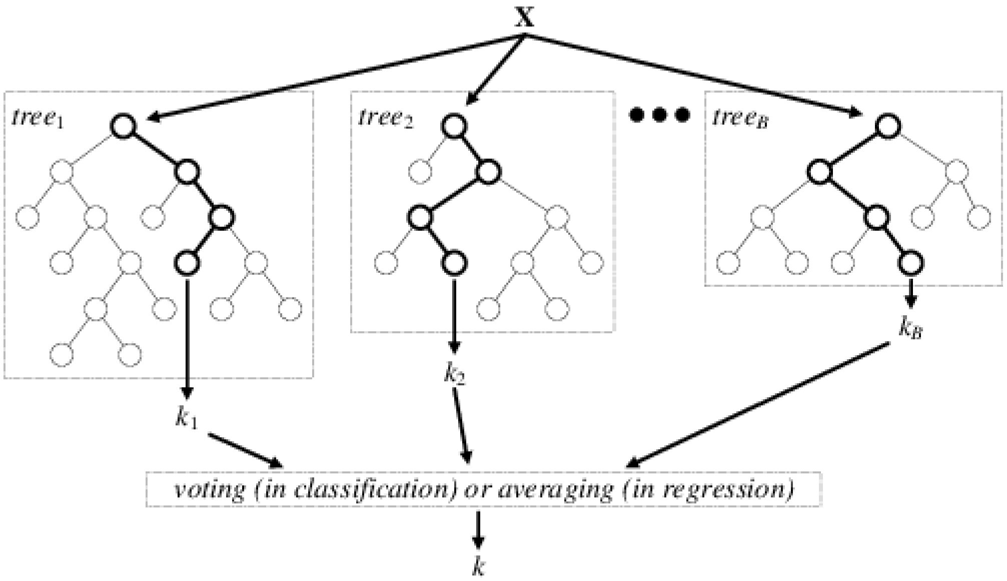
One way of resolving variance problem is to use cross-validation dataset.

(Write about cross validation mentioned in question 2 of **8 Marks EASY** section)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**2. How does ensemble approach improve model performance? What are the popular ensemble methods?**

- **How does ensemble approach improve model performance:**
  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.
- Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to **decrease variance** (bagging), **bias** (boosting), or **improve predictions** (stacking).
- **Popular Ensemble techniques:**
  **(Refer to 8 marks question 1 under Easy section)**
  **In addition to that:**
  **Random forests:**
  In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e. a bootstrap sample) from the training set. In addition, instead of using all the features, a random subset of features is selected, further randomizing the tree.
  As a result, the bias of the forest increases slightly, but due to the averaging of less correlated trees, its variance decreases, resulting in an overall better model.


X
tree₁
tree₂
tree_B
k₁
k₂
k_B
voting (in classification) or averaging (in regression)
k

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**3. What are the different metrics of measuring model performance?**

**(Refer to 8 marks easy question 4)**

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**Standard Questions:**

**1. How do you evaluate a machine learning model? Write different methods to improve the model performance?**

**How do you evaluate a machine learning model?**

**(Refer to 8 marks question 4 Model Evaluation Part)**

**Write different methods to improve the model performance?**

Methods used to improve model performance:

1. **Cross validation (Refer to 8 marks question 2 of Easy section, Diagram and steps for cv)**
2. **Hyperparameter tuning (GridSearch)**
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

- **Hyperparameters:** A hyperparameter is a parameter that determines model architecture and cannot be learned from or estimated by training the data. They are externally given and are to be tuned in order to build a model to obtain the most optimal performance.
- **Hyperparameters are not model parameters.** Model parameters are learned during training
- The process of searching for the most optimal value of a hyperparameter is known as **hyperparameter tuning**. Using optimal values to build a model results in a better architecture of the model and improved model accuracy.
- **Steps to follow for Hyperparameter tuning:**
    - Select the type of model we want to use like RandomForestClassifier, regressor or any other model
    - Check what are the parameters of the model
    - Select the methods for searching the hyperparameter
    - Select the cross-validation approach
    - Evaluate the model using the score

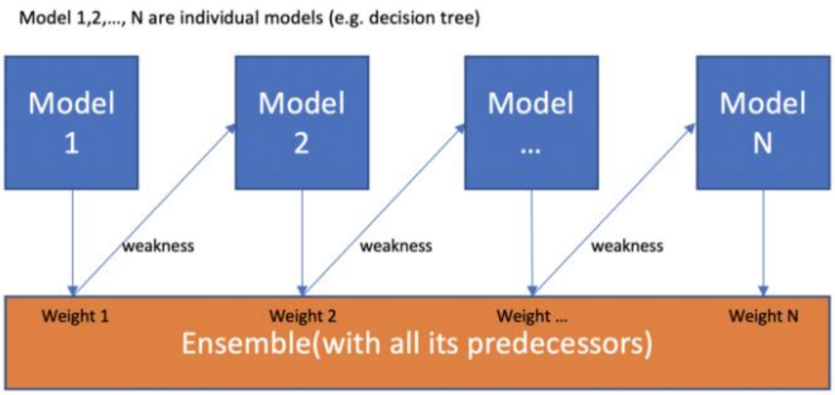
- **GridSearch:**
  Define a search space as a grid of hyperparameter values and evaluate every position in the grid.
  **Grid search for SVM:**
  **Hyperparameters of SVM:**
  **C:** Regularization Parameter
  **Kernel:** The main function of the kernel is to take low dimensional input space and transform it into a higher-dimensional space. It is mostly useful in non-linear separation problem.
  **Kernel values:** Linear, rbf, poly, sigmoid, precomputed
  **Degree:** Degree of the polynomial kernel and is ignored by other kernels.
  **Gamma:** Kernel coefficient for rbf, poly and sigmoid. (scaled,auto). Default value of gamma is 'scale'
  **coef0:** Independent term in kernel function. It is only significant in poly and sigmoid. Default value of coef0 is '0.0'.
  **Cross validation metric is used. Here number of folds is set to 5.**
- GridsearchCV is used. It builds a grid that comprises of model score for each hyperparameter.
- best_params_ is used to display the best values of each corresponding hyperparameter.
- The best accuracy score of the model is displayed using best_score_

**Grid Search : [https://youtu.be/HdlDYng8g9s](https://youtu.be/HdlDYng8g9s)**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

**2. Does boosting method gives better performance? Draw a diagram for boosting process.**

- **Boosting** is an ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones. It also minimizes a loss function.


Model 1,2,..., N are individual models (e.g. decision tree)
Model 1
Model 2
Model ...
Model N
weakness
weakness
weakness
Weight 1
Weight 2
Weight ...
Weight N
Ensemble(with all its predecessors)

One is weak, together is strong, learning from past is the best

- To convert weak learner to strong learner following methods are used:
    1. Using average or weighted average.
    2. Consider prediction has a higher vote.
- Commonly used Boosting Algorithms:

    **1. AdaBoost (Adaptive Boosting):**

    In each iteration, AdaBoost identifies miss-classified data points, increasing their weights (and decrease the weights of correct points, in a sense) so that the next classifier will pay extra attention to get them. A decision stump is used as a weak learner here. The exponential loss function is minimized in AdaBoost.

    **2. Gradient Tree Boosting:**

    Gradient boosting algorithm is a machine learning technique to define loss function and reduce it. It is used to solve problems of classification using prediction models.

    Gradient boosting involves three elements:
    1. A loss function to be optimized.
    2. A weak learner to make predictions.
    3. An additive model to add weak learners to minimize the loss function.

    **3. XGBoost**

    XG Boost is short for Extreme Gradient Boosting. XG Boost is upgraded implementation of Gradient Boosting Algorithm which is developed for high computational speed, scalability, and better performance.

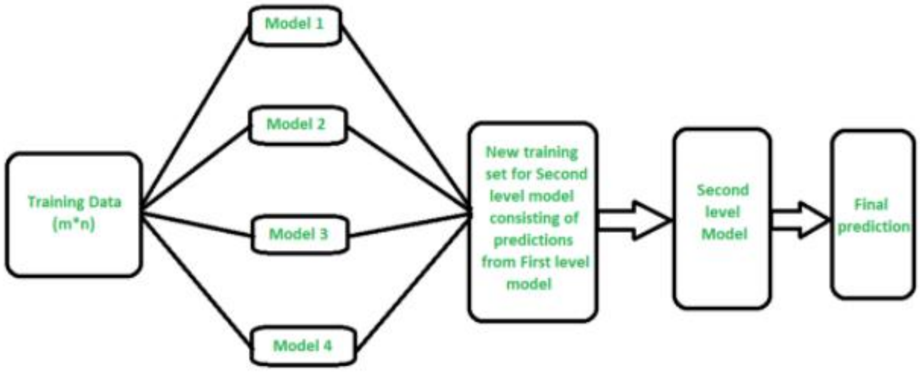
    The main features provided by XGBoost are:
    - Parallelly creates decision trees.
    - Implementing distributed computing methods for evaluating large and complex models.
    - Using Out-of-Core Computing to analyze huge datasets.
    - Implementing cache optimization to make the best use of resources.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**3. How does stacking process work. Explain with a diagram.**

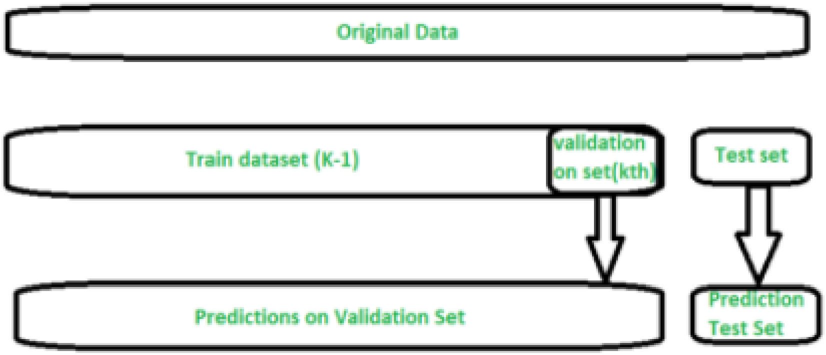
- The point of stacking is to explore a space of different models for the same problem.
- The idea is that you can attack a learning problem with different types of models which are capable to learn some part of the problem, but not the whole space of the problem.
- So, you can build multiple different 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.


Training Data (m*n)
Model 1
Model 2
Model 3
Model 4
New training set for Second level model consisting of predictions from First level model
Second level Model
Final prediction

**How stacking works?**

1. We split the training data into K-folds just like K-fold cross-validation.
2. A base model is fitted on the K-1 parts and predictions are made for Kth part.
3. We do for each part of the training data.
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.
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.


Original Data
Train dataset (K-1)
validation on set(kth)
Test set
Predictions on Validation Set
Prediction Test Set