Data Science and Artificial Intelligence

Machine Learning

Bias and Variance

Lecture No. 2

By- SIDDHARTH SABHARWAL SIR

Recap of Previous Lecture

- Topic — Bias
- Topic — Variance
- Topic
- Topic
- Topic

Topics to be Covered

- Topic — Bias / Variance
- Topic — Feature selection technique
- Topic — Ensemble technique
- Topic
- Topic


YOUR MORNING
SETS UP THE
SUCCESS
OF YOUR DAY
Fazil Azmaan

Basics of Machine Learning

REVISION

Bias $\Rightarrow$ $Y - E(\hat{Y})$

- MSE $\Rightarrow$ $\underbrace{\text{Bias}^2 + \text{Variance}}_{\text{Controllable}} + \underbrace{\sigma^2}_{\text{Var of noise}}$

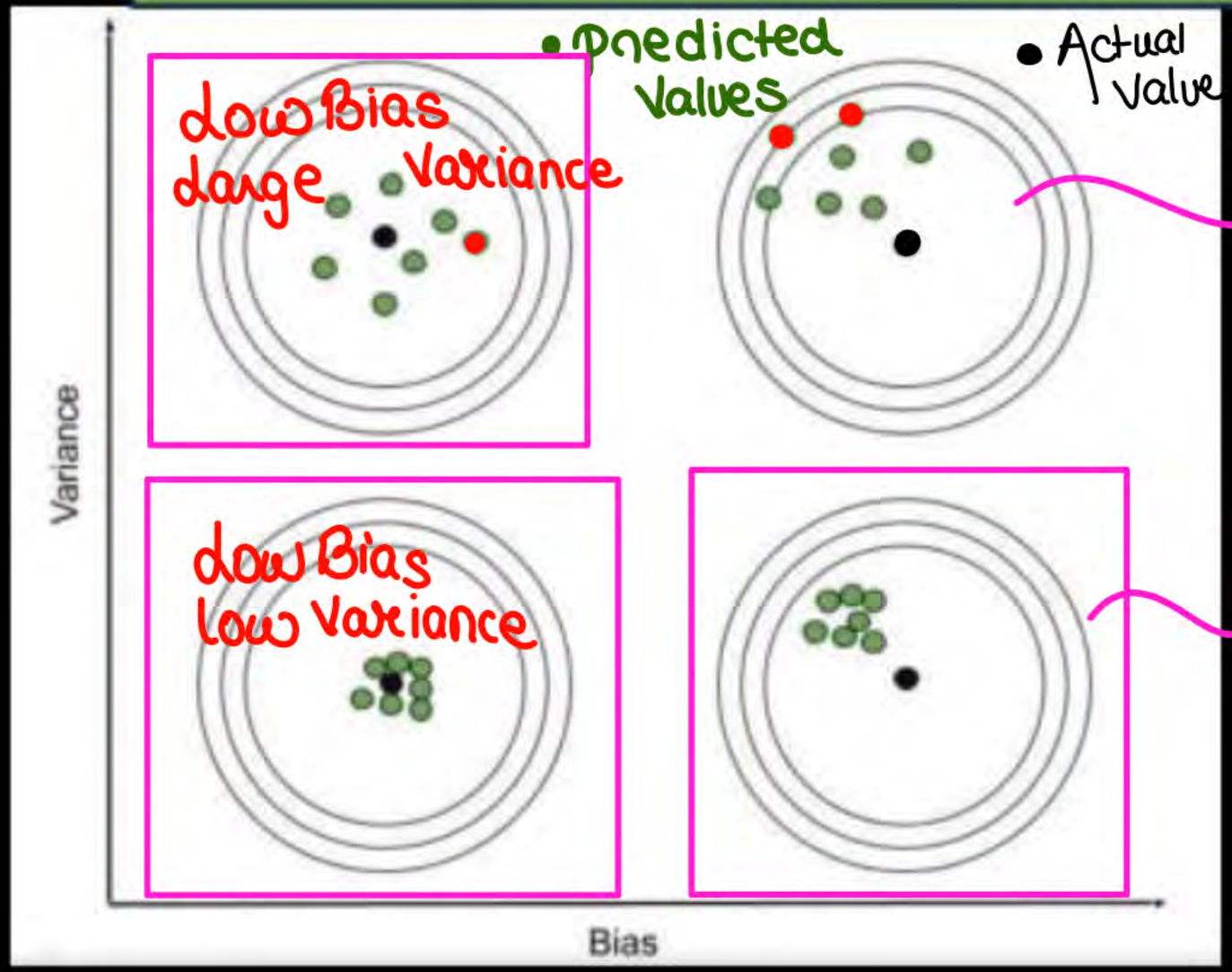
Basics of Machine Learning

Variance $\Rightarrow E\left(\left(E(\hat{Y})-\hat{Y}\right)^2\right)$

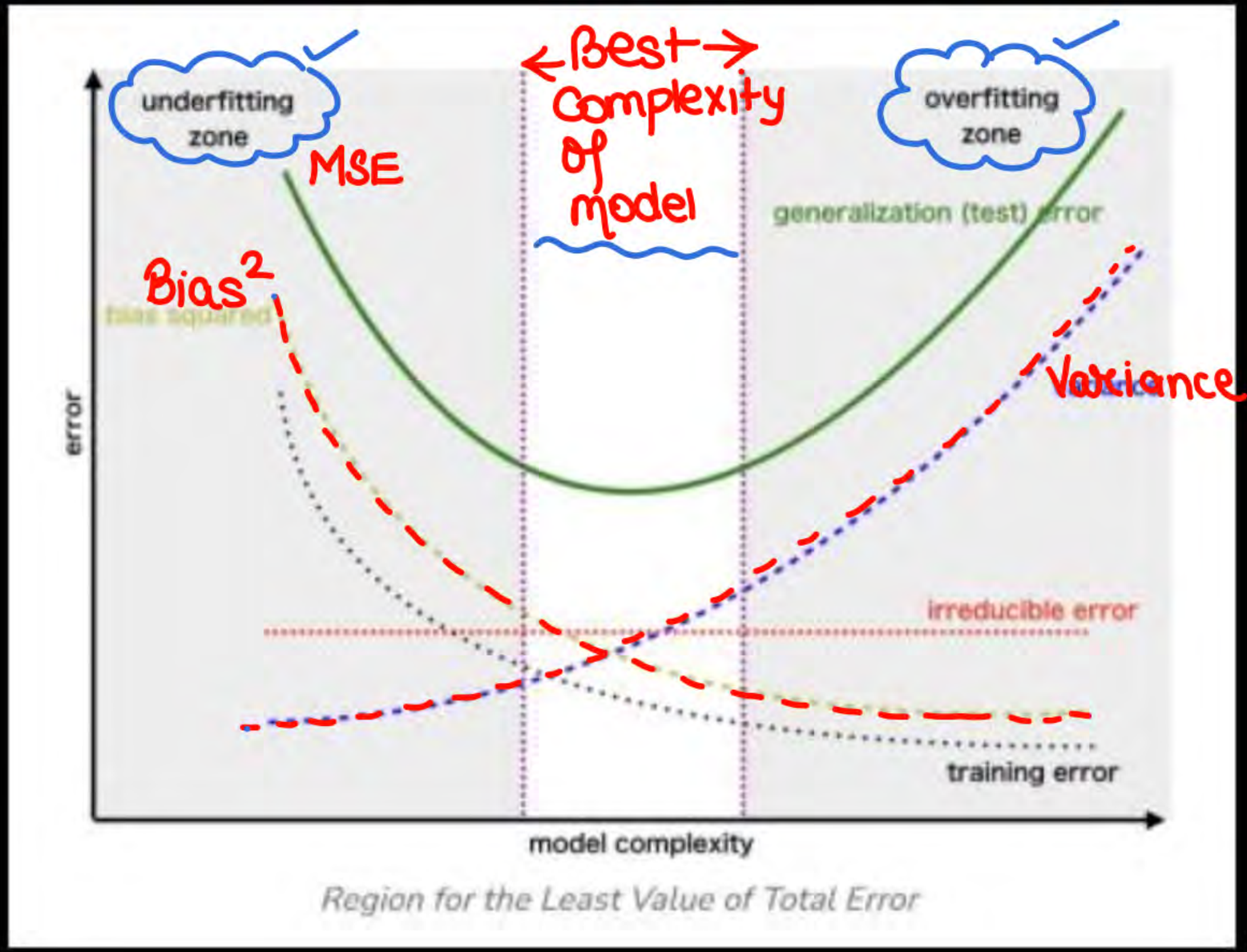
↳ Proxy to test error

Bias and Variance

Different Combinations of Bias-Variance



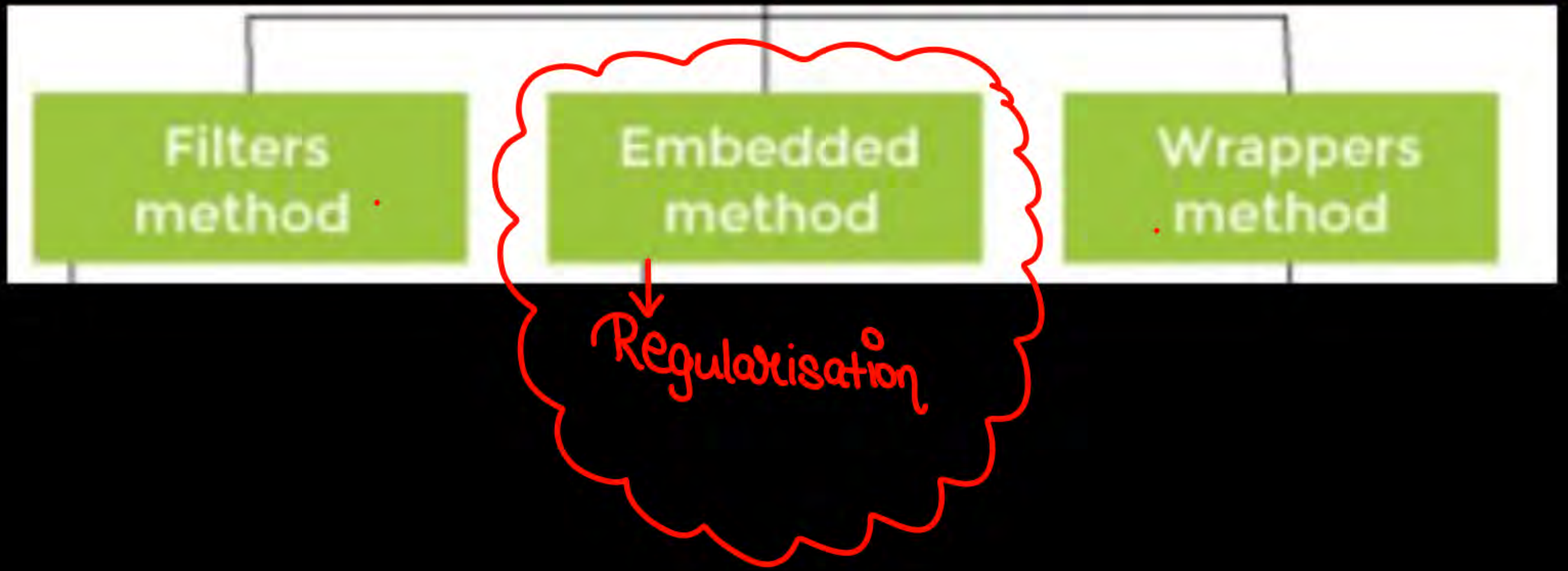
Bias and Variance



Region for the Least Value of Total Error

Only for ref

Feature Selection Methods



Bias and Variance

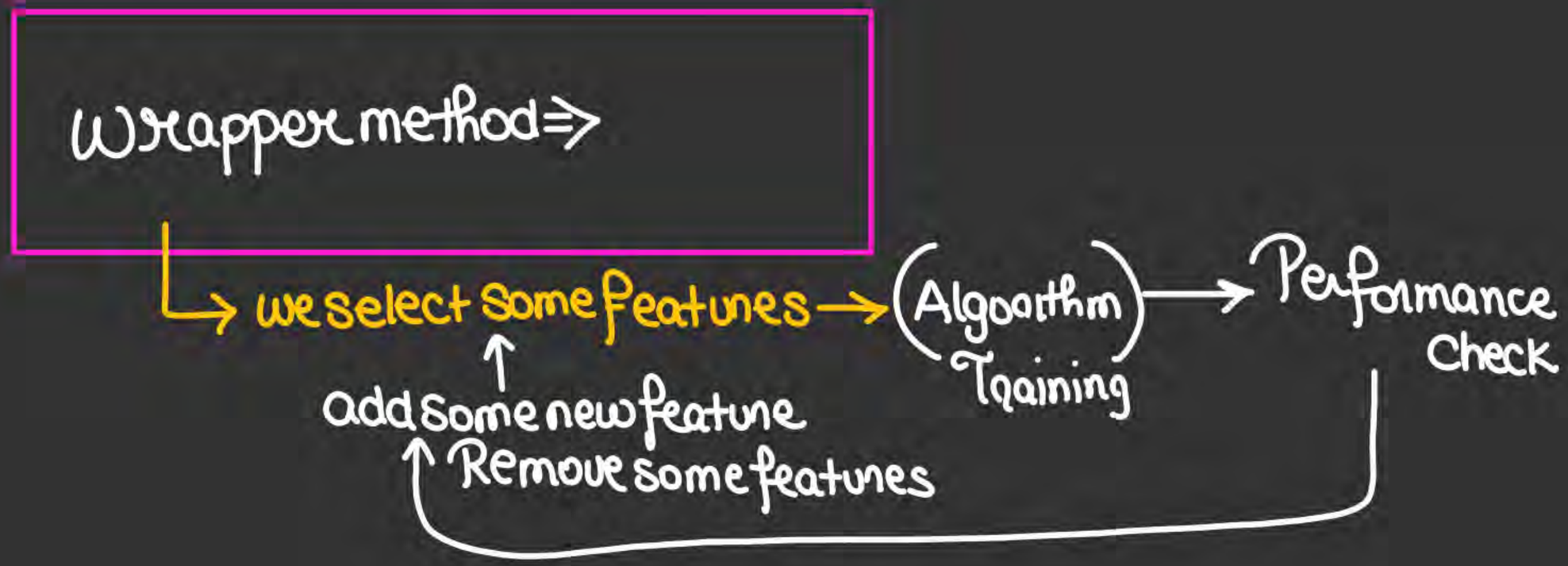
Why Feature selection ⇒ → Remove overfitting

The Role of Feature Selection

1. To reduce the dimensionality of feature space.
2. To speed up a learning algorithm.
3. To improve the predictive accuracy of a classification algorithm. → Algorithm become more generalised.
4. To improve the comprehensibility of the learning results.

⇓

→ Improve Quality of Results.

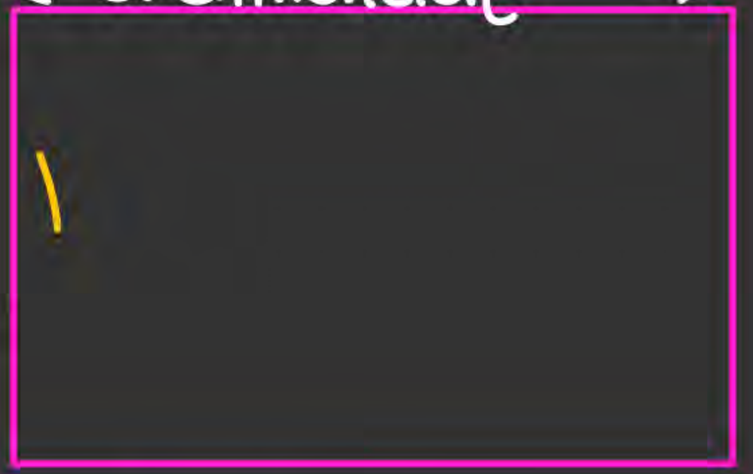

Wrapper method ⇒
we select some features
Algoarithm
Training
Performance
Check
Add some new feature
Remove some features

Wrapper method

↳ 1. Forward Stepwise Selection ⇒

Check performance $\Rightarrow \frac{1}{N}\sum(y_i - \hat{y}_i)^2$

← d dimension →



Step1 Take Single dimension → Train Algo → Check Performance

Best dimension Choosen as 1st dimension

Step2 ⇒ 1st dimension + Take Single dimension from data → Train Algo → Check Performance

Step 3 1st dimen 2nd dimen ____

Similarly choose 3rd ____ dimension

and finally we get best Combination of dimension for analysis.

- So the process will stop when the improvement on adding dimension < some threshold

Backward elimination method ⇒

- Take all dimensions → Train algo → Check Performance
- Now start eliminating dimensions one by one and check the performance of algo at each step, If change in performance < threshold then Remove the dimension else keep it.

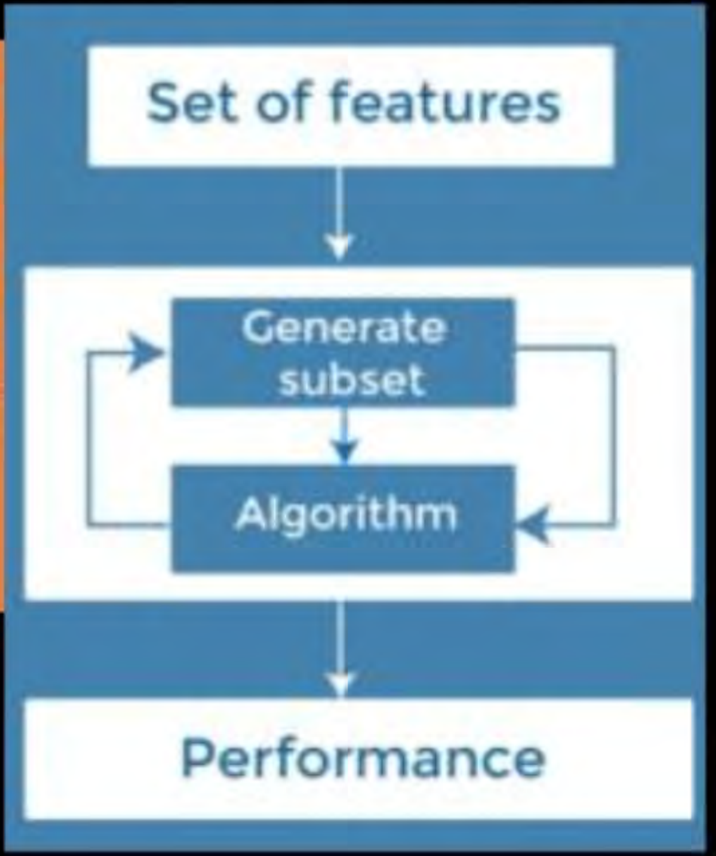
Exhaustive feature selection

- So we create all possible combination and then check performance of all all combination using algo.
- Find Best Combination


This will give best
Result but practically not Possible

Wrapper Methods

- Here selection of features is done by considering it as a search problem, in which different combinations are made, evaluated, and compared with other combinations. It trains the algorithm by using the subset of features iteratively.
- These are computationally extensive



Wrapper Methods

Some techniques of wrapper methods are: (Forward- and Backward-Stepwise Selection)

- Forward selection - Forward selection is an iterative process, which begins with an empty set of features. After each iteration, it keeps adding on a feature and evaluates the performance to check whether it is improving the performance or not. The process continues until the addition of a new variable/feature does not improve the performance of the model.
- Backward elimination - Backward elimination is also an iterative approach, but it is the opposite of forward selection. This technique begins the process by considering all the features and removes the least significant feature. This elimination process continues until removing the features does not improve the performance of the model.
- Exhaustive Feature Selection- Exhaustive feature selection is one of the best feature selection methods, which evaluates each feature set as brute-force. It means this method tries & make each possible combination of features and return the best performing feature set.

Filter method → Maths before using algorithm

→ So here the dimension are filtered out,
we calculate the correlation b/w the Y and dimension.
the dimension which are correlated with Y are
important, and rest are not important.

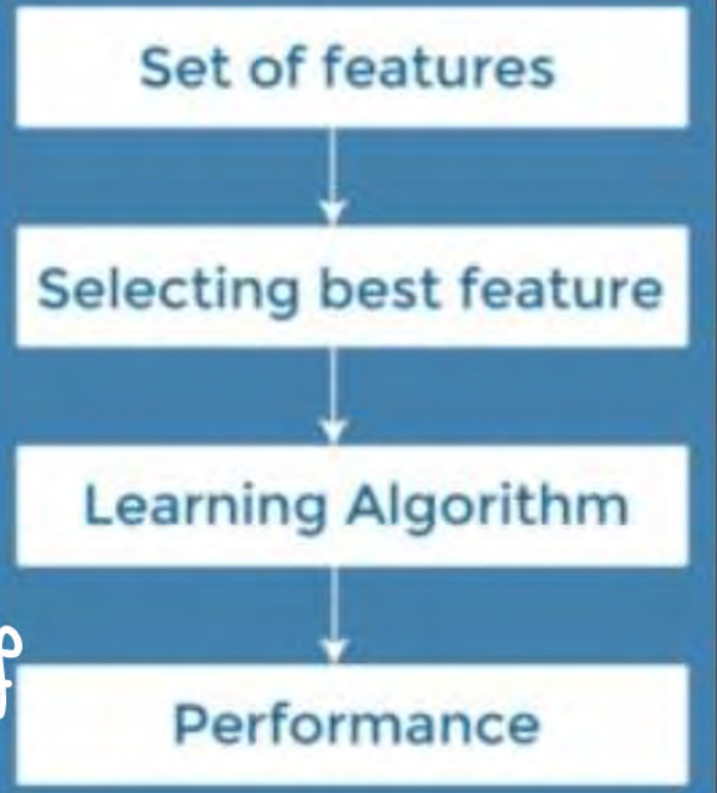
Bias and Variance

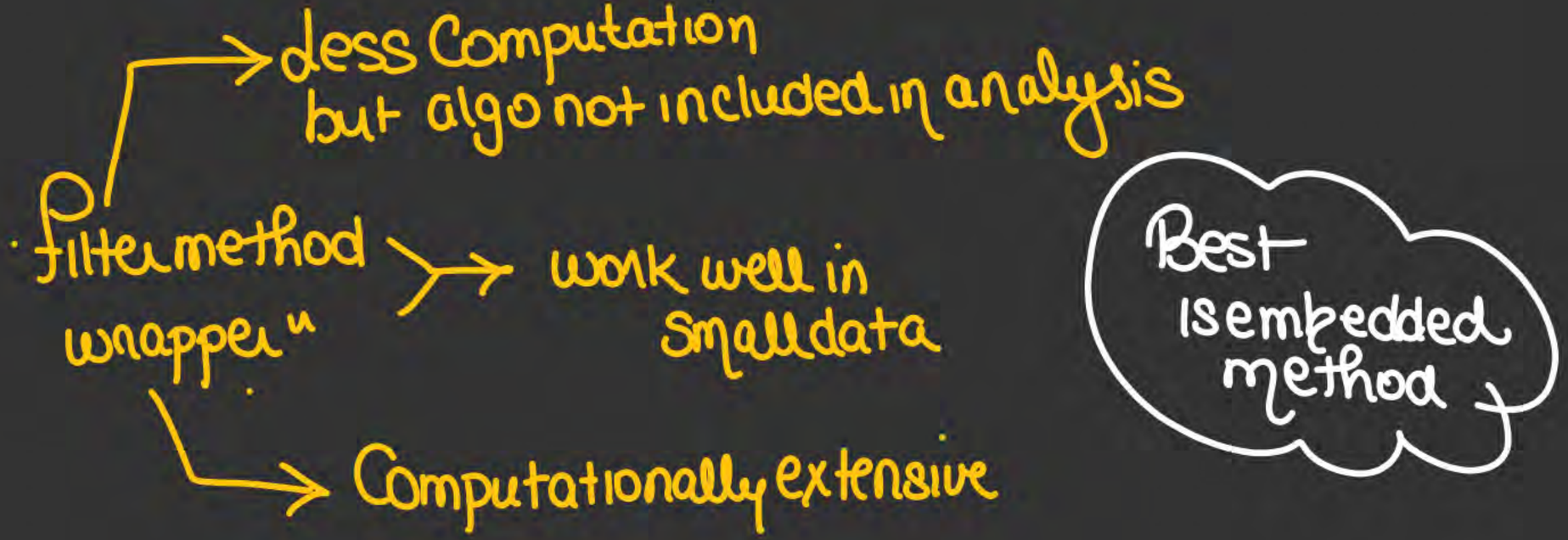
→ not as good as wrapper.

Filter Methods

- In Filter Method, features are selected on the basis of statistics measures. This method does not depend on the learning algorithm and chooses the features as a pre-processing step.
- Actually we find the features which are having maximum correlation with the output or label.
- The filter method filters out the irrelevant feature and redundant columns from the model by using different metrics through ranking.
- The advantage of using filter methods is that it needs low computational time and does not overfit the data.

CorrelationCoef



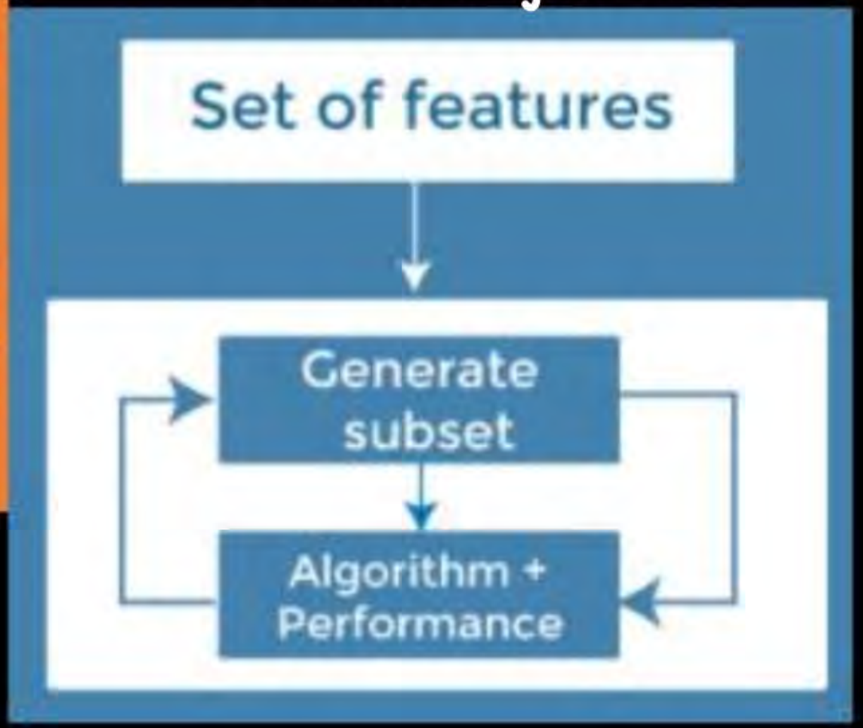

less Computation
but algo not included in analysis
filter method
wrapper"
work well in
small data
Computationally extensive
Best
is embedded
method

Bias and Variance

→ Best method,

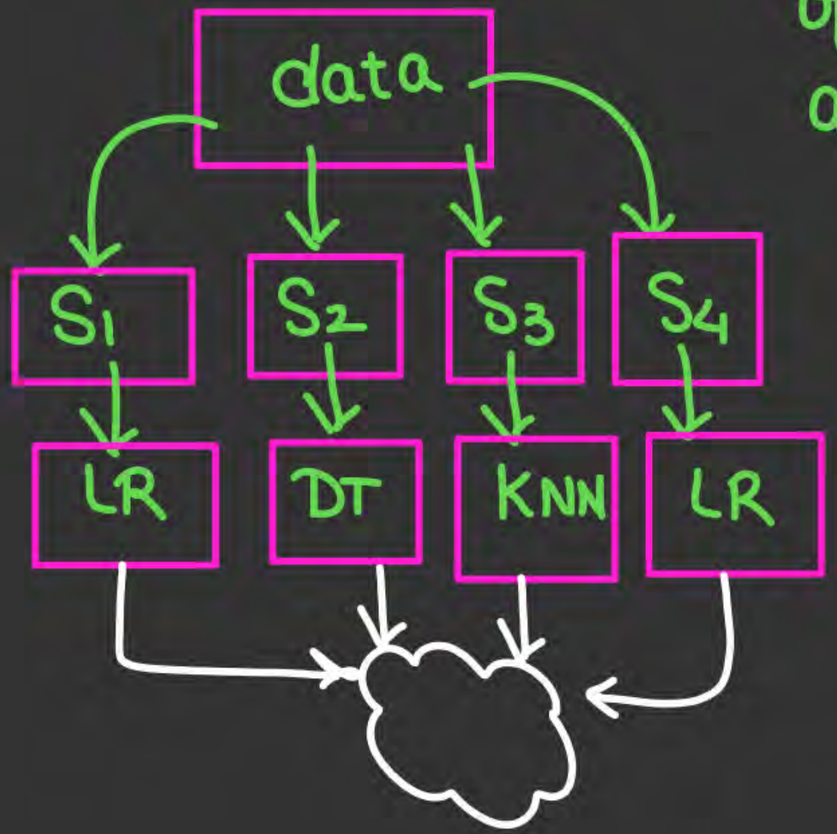
Embedded Methods → Regularisation $ Randomforest

- The above methods are used when the dataset is small. But when the dataset is large then we use Embedded methods
- Regularisation. + Randomforest → to find best λ.
- These methods are also iterative, which evaluates each iteration, and optimally finds the most important features that contribute the most to training in a particular iteration.



- Ensemble learning ⇒

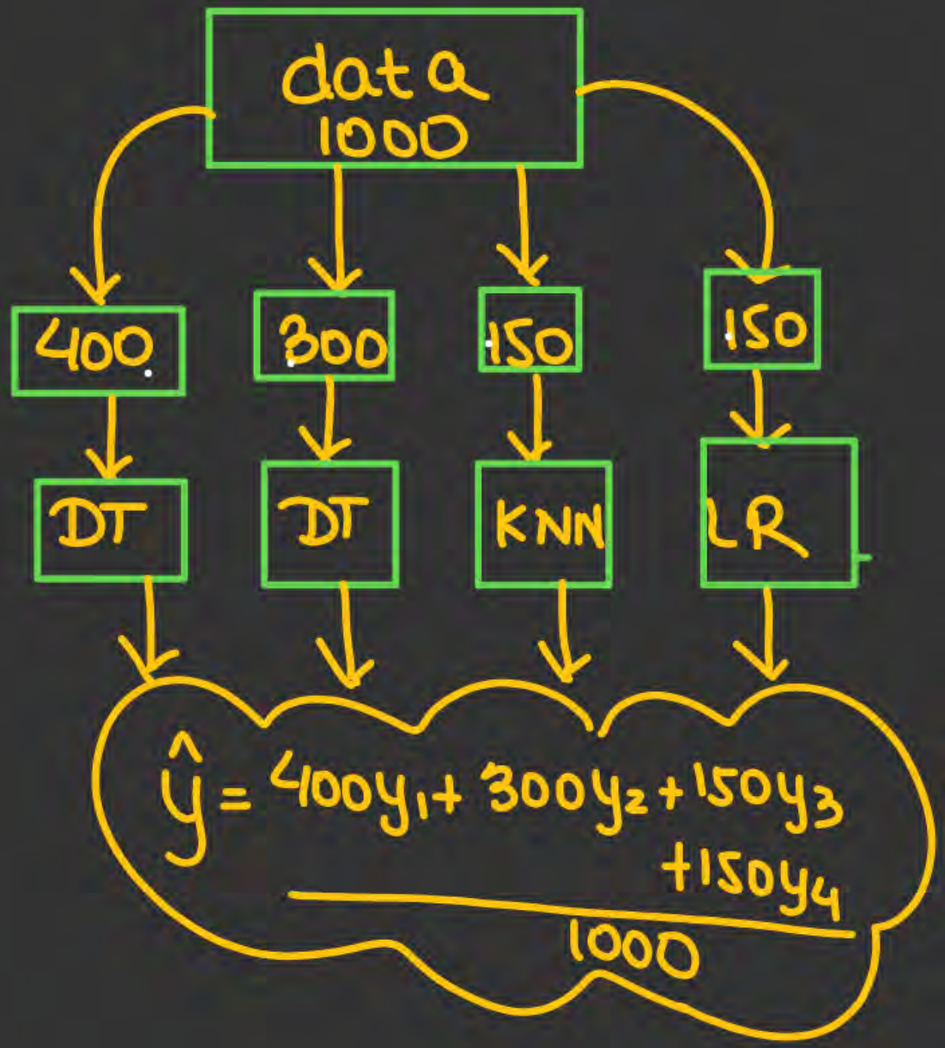
The method where Single model is not created, rather from subsets of data we create many models and final decision is taken by



final $\hat{Y}$ = avg of all $\hat{Y}$ → Reg.

final $\hat{Y}$ = majority voting → Class.

final $\hat{Y}$ = weighted avg $\hat{Y}$ → Reg.


data
1000
400
300
150
150
DT
DT
KNN
LR
$\hat{y} = \frac{400y_1 + 300y_2 + 150y_3 + 150y_4}{1000}$

- It may lead to underfit → To take care of this Subsets shd be large enough.

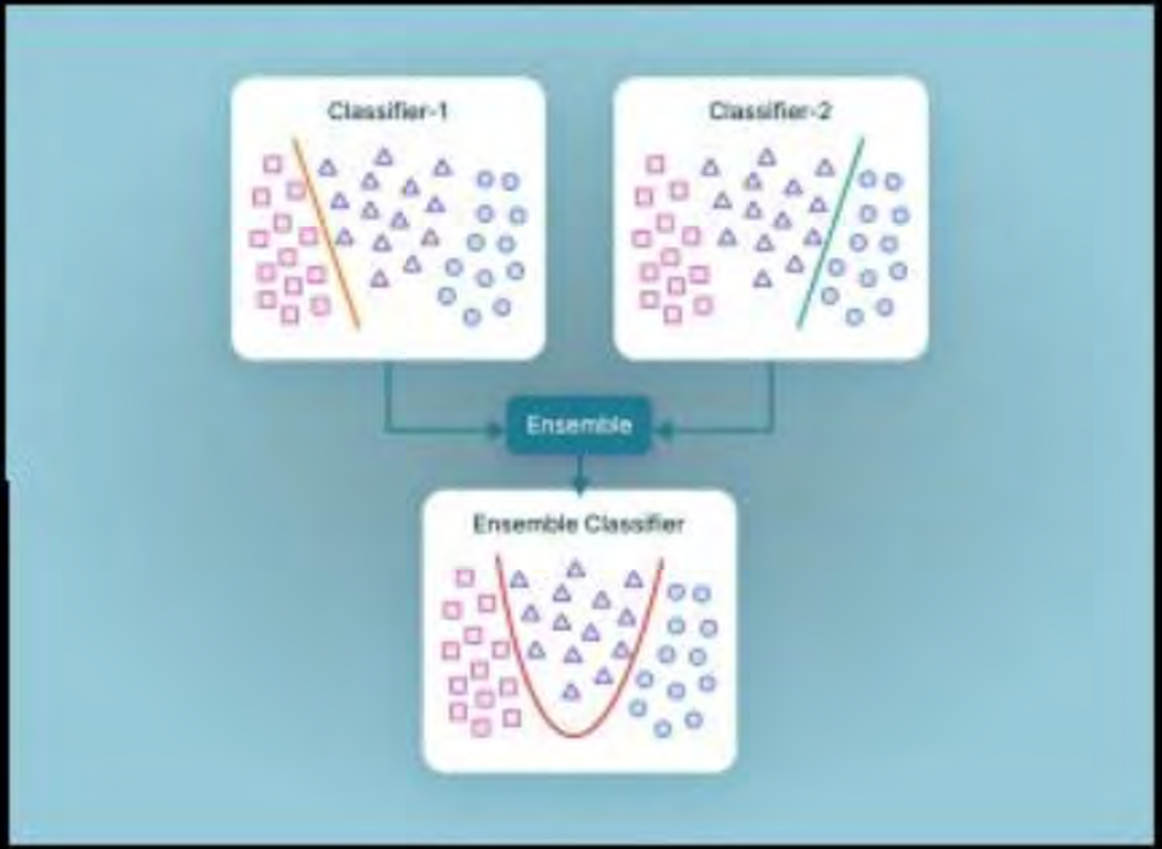
- Always better to create Subsets such that
  Propontion of all class of data in each subset
  ≠ Same as that of original data
  ↳ Stratified manner of distribution.

Bias and Variance

(ref, overview)

Ensemble learning

- Don't consult only one expert but consult many expert before taking the final decision.
- Ensemble learning helps improve machine learning results by combining several models.



- combine the outputs of diverse models to create a more precise prediction.

Few simple but powerful techniques, namely:

1. Max Voting
2. Averaging
3. Weighted Averaging

Bias and Variance

Ensemble learning

- This make the model more generalised and thus the test error and the train error so bias and variance decreases.

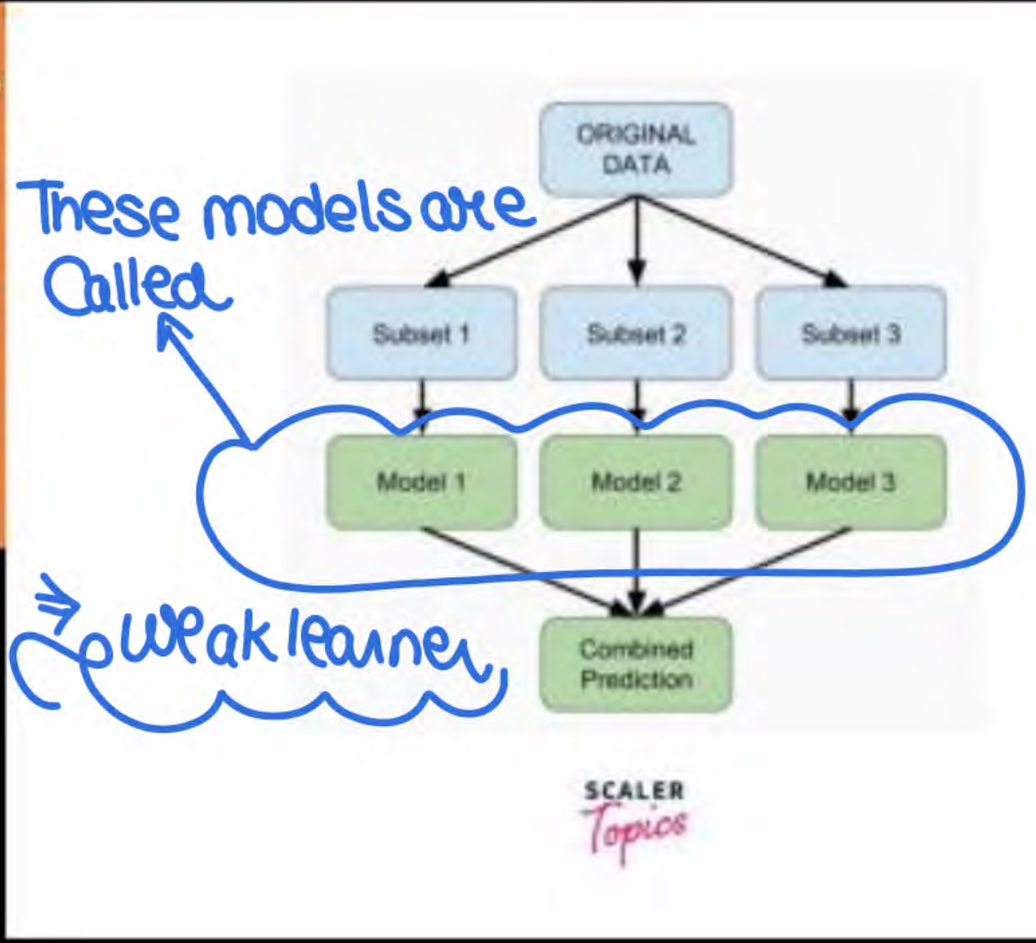
- Lets prove that the variance decreases...

- Becoz no model has access to whole data so no chance of overfitting

Bias and Variance

Ensemble learning

- All these models are called the base learners. → or weak learner
- These base learners can take different algorithms.
- And also we can give different training data to each of the model.
- These base learners are also called weak learners.



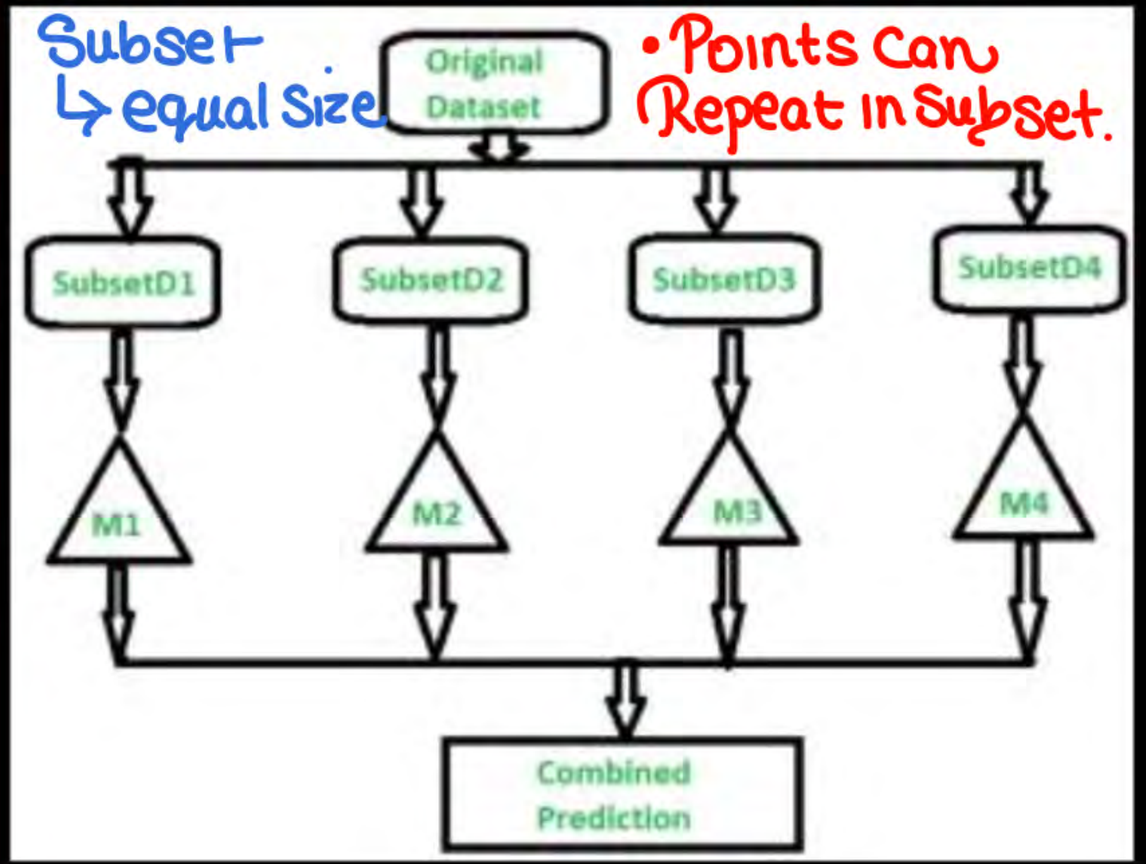
Bias and Variance

So whatever we have learnt abt Ensemble learning → Bagging Technique

Ensemble learning (bagging/Bootstraping)

❖ **Types of Ensemble Classifier – Bagging:**

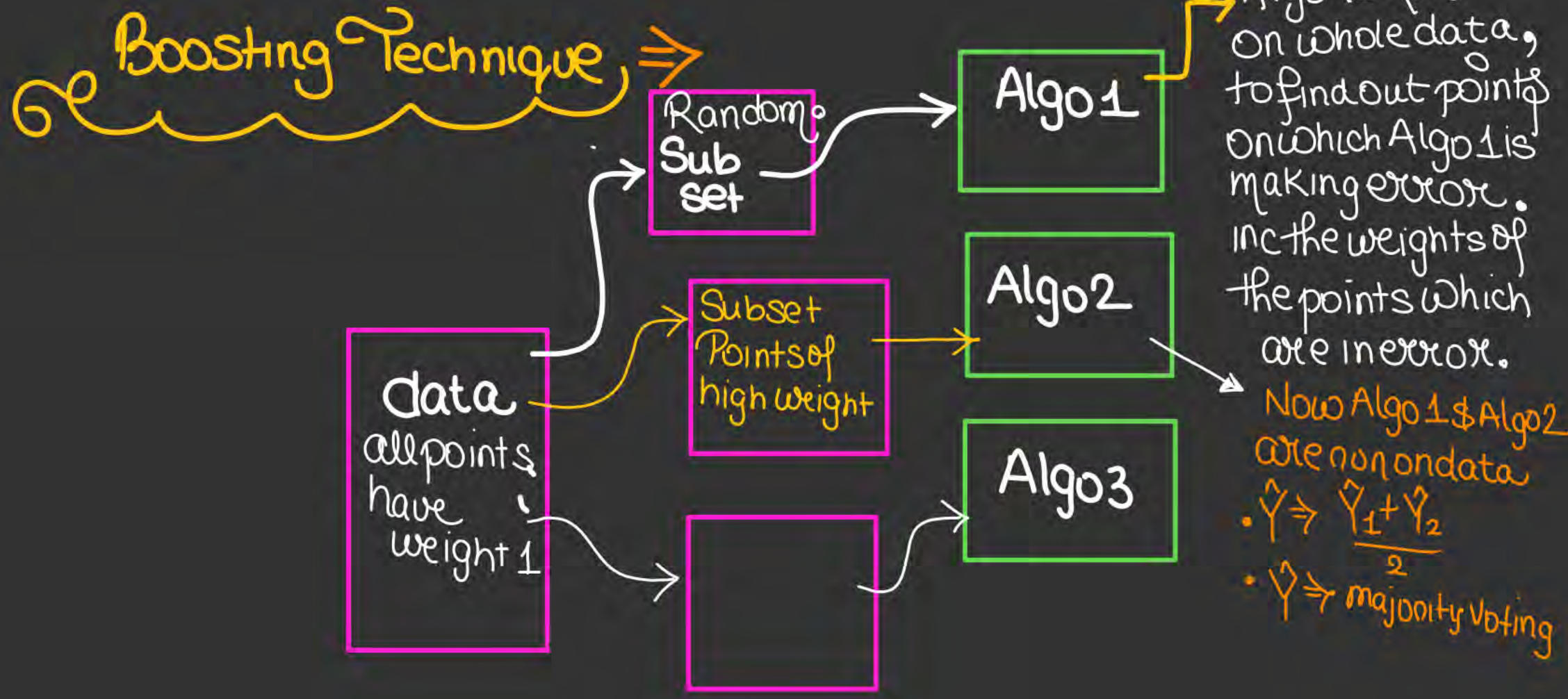
1. In Bootstrapping Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
2. But in Bagging we can create subset of different sizes
3. A base model is created on each of these subsets. (these are called the weak model)
4. Each model is learned in parallel from each training set and independent of each other.
5. The final predictions are determined by combining the predictions from all the models. ✔



Bias and Variance

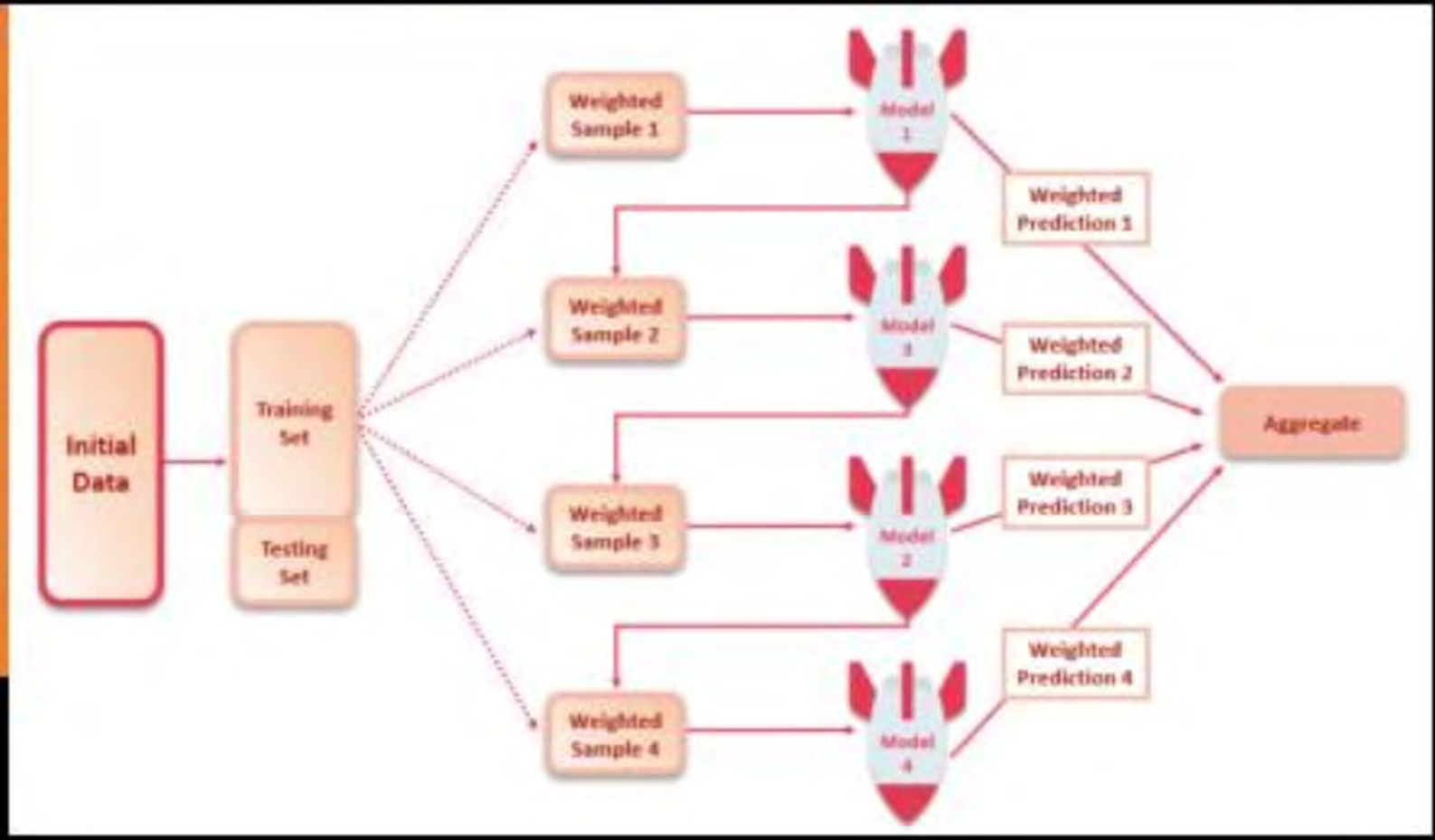
Ensemble learning

- Problem in DT is that it become too large on big dataset thus we use Ensemble learning here, So we break the training data into subsets and then train my model on these subsets.

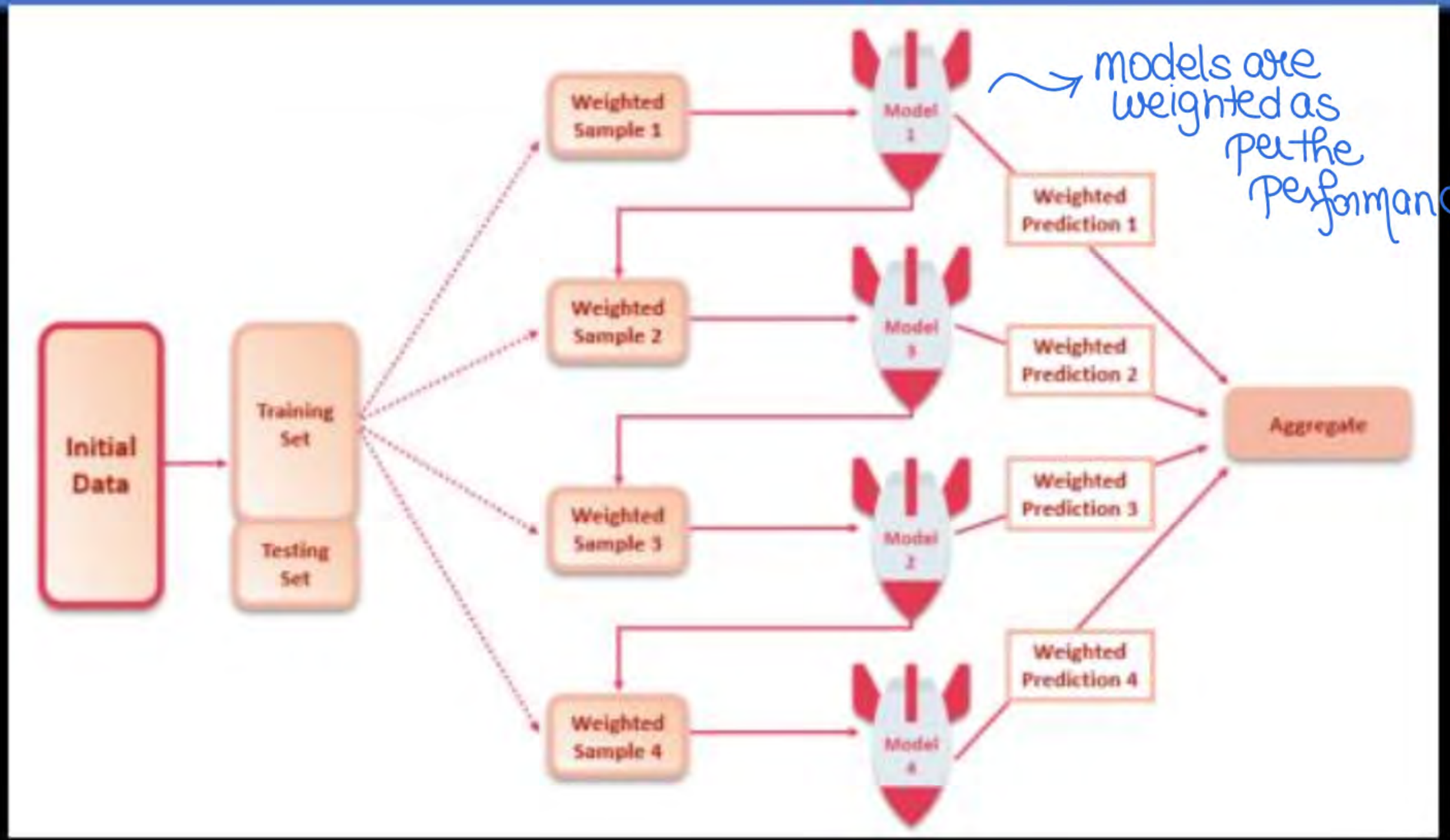

Boosting Technique ⇒
Random Sub set
Algo 1
Algo is Run on whole data, to find out points on which Algo 1 is making error. Inc the weights of the points which are in error.
data
all points have weight 1
Subset Points of high weight
Algo2
Now Algo 1 $ Algo 2 are run on data
• Ŷ ⇒ (Ŷ1 + Ŷ2) / 2
• Ŷ ⇒ majority voting
Algo3

Ensemble learning

- Types of Ensemble Classifier – Boosting:
- This is like Bagging.
- But this is not a parallel process rather a sequential process...
- Here we first learn a model and find the error on the data and then train next model where we have more error...



Ensemble learning - Boosting



Ensemble learning - Boosting

Sequential Method.

1. Samples generated from the training set are assigned the **same weight** to start with. These samples are used to train a homogeneous weak learner or base model.
2. The prediction error for a sample is calculated – **the greater the error, the weight of the sample increases**. Hence, the sample becomes more important for training the next base model.
3. The individual learner is weighted too – **does well on its predictions, gets a higher weight assigned to it**. So, a model that outputs good predictions will have a higher say in the final decision.
4. The weighted data is then passed on to the following base model, and steps 2) and 3) are repeated until the **data is fitted well enough to reduce the error below a certain threshold**.
5. When new data is fed into the boosting model, it is passed through all individual base models, and **each model makes its own weighted prediction**.
6. Weight of these models is used to generate the final prediction. The predictions are scaled and **aggregated to produce a final prediction**.

Random Forest Algorithm

Steps Involved in Random Forest Algorithm

- **Step 1:** In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.
- **Step 2:** Individual decision trees are constructed for each sample.
- **Step 3:** Each decision tree will generate an output.
- **Step 4:** Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression, respectively.

Bias and Variance

Random Forest Algorithm

Important Hyperparameters in Random Forest

- Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster.
- **Number of decision trees to be constructed**
- **Maximum number of features a tree can use**
- **Splitting thresholds**

Random Forest Algorithm

Key Benefits

- Reduced risk of overfitting: when there's a robust number of decision trees in a random forest, the classifier won't overfit the model since the averaging of uncorrelated trees lowers the overall variance and prediction error.
- Provides flexibility: Since random forest can handle both regression and classification tasks with a high degree of accuracy.
- Easy to determine feature importance: Random forest makes it easy to evaluate variable importance, or contribution, to the model. There are a few ways to evaluate feature importance. Gini importance and decrease in impurity are usually used to measure how much the model's accuracy decreases when a given variable is excluded.

Bias and Variance

Random Forest Algorithm

Key Challenges

- Time-consuming process: Since random forest algorithms can handle large data sets, they can be provide more accurate predictions, but can be slow to process data as they are computing data for each individual decision tree.
- Requires more resources: Since random forests process larger data sets, they'll require more resources to store that data.
- More complex: The prediction of a single decision tree is easier to interpret when compared to a forest of them.

Bias and Variance

Random Forest Algorithm

Decision trees	Random Forest
1. Decision trees normally suffer from the problem of overfitting if it's allowed to grow without any control.	1. Random forests are created from subsets of data, and the final output is based on average or majority ranking; hence the problem of overfitting is taken care of.
2. A single decision tree is faster in computation.	2. It is comparatively slower.
3. When a data set with features is taken as input by a decision tree, it will formulate some rules to make predictions.	3. Random forest randomly selects observations, builds a decision tree, and takes the average result. It doesn't use any set of formulas.

Random Forest Algorithm

Feature	Random Forest	Other ML Algorithms
Ensemble Approach	Utilizes an ensemble of decision trees, combining their outputs for predictions, fostering robustness and accuracy.	Typically relies on a single model (e.g., linear regression, support vector machine) without the ensemble approach, potentially leading to less resilience against noise.
Overfitting Resistance	Resistant to overfitting due to the aggregation of diverse decision trees, preventing memorization of training data.	Some algorithms may be prone to overfitting, especially when dealing with complex datasets, as they may excessively adapt to training noise.
Handling of Missing Data	Exhibits resilience in handling missing values by leveraging available features for predictions, contributing to practicality in real-world scenarios.	Other algorithms may require imputation or elimination of missing data, potentially impacting model training and performance.
Variable Importance	Provides a built-in mechanism for assessing variable importance, aiding in feature selection and interpretation of influential factors.	Many algorithms may lack an explicit feature importance assessment, making it challenging to identify crucial variables for predictions.
Parallelization Potential	Capitalizes on parallelization, enabling the simultaneous training of decision trees, resulting in faster computation for large datasets.	Some algorithms may have limited parallelization capabilities, potentially leading to longer training times for extensive datasets.

Maximum likelihood Estimation

What is MLE (lets see an example)

Maximum Likelihood Estimation (MLE) is a statistical method used to estimate the parameters of a probability distribution that best describe a given dataset. The fundamental idea behind MLE is to find the values of the parameters that maximize the likelihood of the observed data, assuming that the data are generated by the specified distribution.

THANK - YOU