# Random Forest and Ensemble Learning

Tushar B. Kute,
http://tusharkute.com

# Ensemble Learning

- An ensemble is a composite model, combines a series of low performing classifiers with the aim of creating an improved classifier.
- Here, individual classifier vote and final prediction label returned that performs majority voting.
- Ensembles offer more accuracy than individual or base classifier.
- Ensemble methods can parallelize by allocating each base learner to different-different machines.
- Finally, you can say Ensemble learning methods are meta-algorithms that combine several machine learning methods into a single predictive model to increase performance.
- Ensemble methods can decrease variance using bagging approach, bias using a boosting approach, or improve predictions using stacking approach.

# Real life examples

- Let's take a real example to build the intuition.
- Suppose, you want to invest in a company XYZ. You are not sure about its performance though.
- So, you look for advice on whether the stock price will increase by more than 6% per annum or not?
- You decide to approach various experts having diverse domain experience:

# The survey prediction

- Employee of Company XYZ:
  - In the past, he has been right 70% times.
- Financial Advisor of Company XYZ:
  - In the past, he has been right 75% times.
- Stock Market Trader:
  - In the past, he has been right 70% times.
- Employee of a competitor:
  - In the past, he has been right 60% times.
- Market Research team in the same segment:
  - In the past, he has been right 75% times.
- Social Media Expert:
  - In the past, he has been right 65% times.

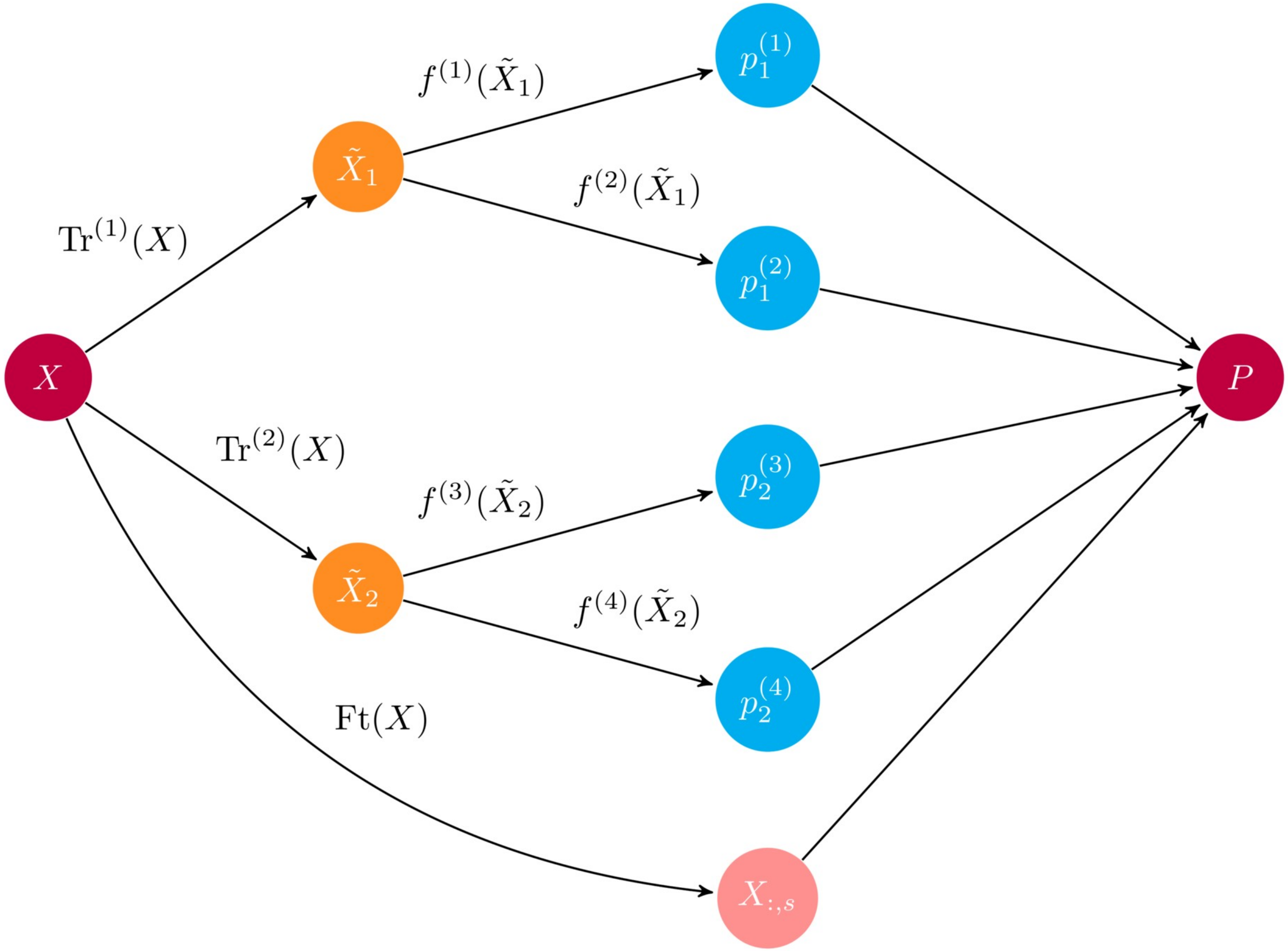
# Conclusion

- Given the broad spectrum of access you have, you can probably combine all the information and make an informed decision.
- In a scenario when all the 6 experts/teams verify that it's a good decision(assuming all the predictions are independent of each other), you will get a combined accuracy rate of 1 - (30% . 25% . 30% . 40% . 25% . 35%) = 1 - 0.07875 = 99.92125%
- The assumption used here that all the predictions are completely independent is slightly extreme as they are expected to be correlated. However, you can see how we can be so sure by combining various forecasts together.
- Well, Ensemble learning is **no** different.

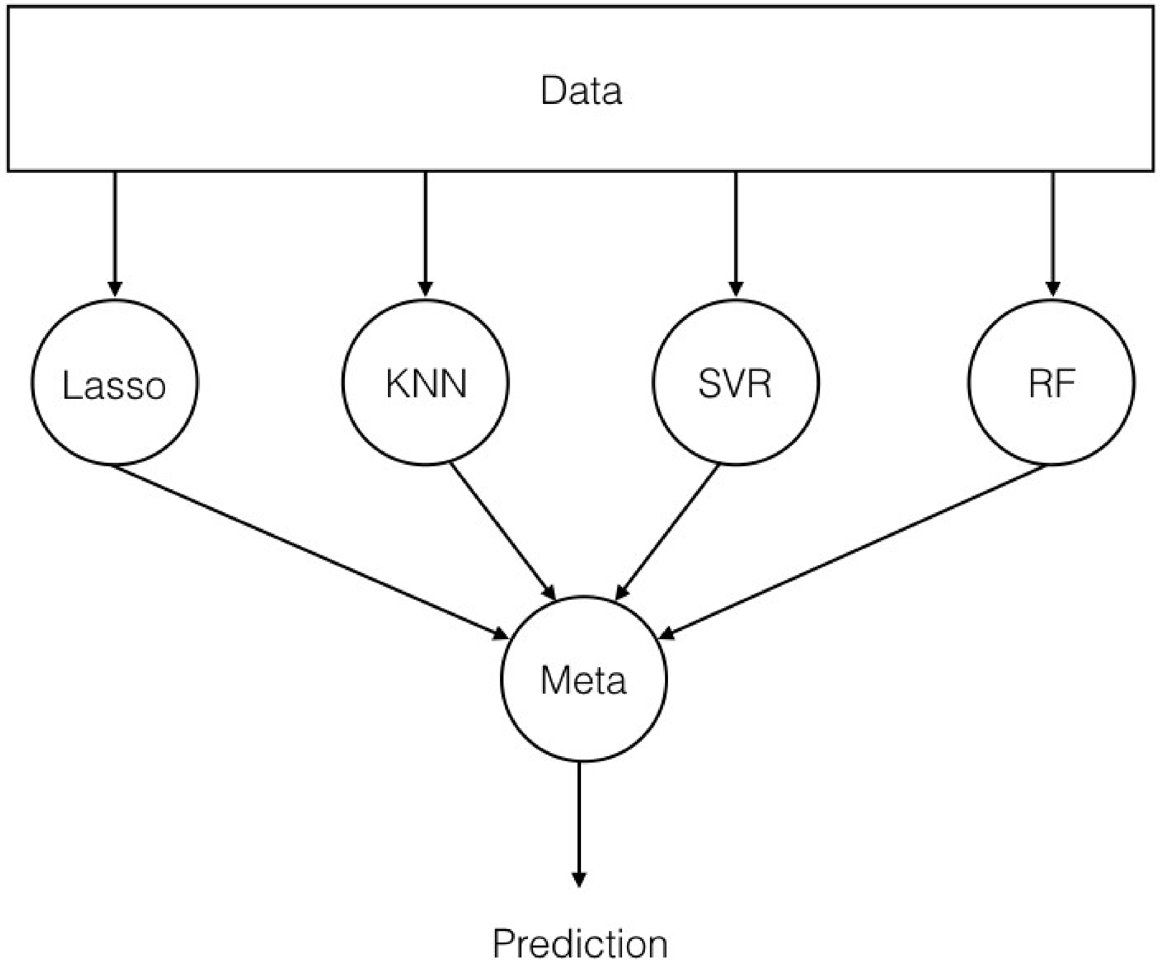
# Ensembling

- An ensemble is the art of combining a diverse set of learners (individual models) together to improvise on the stability and predictive power of the model.
- In our example, the way we combine all the predictions collectively will be termed as Ensemble learning.
- Moreover, Ensemble-based models can be incorporated in both of the two scenarios, i.e., when data is of large volume and when data is too little.

# Ensemble Schematic



# Basic Ensemble Structure



# Summary

- Use multiple learning algorithms (classifiers)
- Combine the decisions
- Can be more accurate than the individual classifiers
- Generate a group of base-learners
- Different learners use different
  - Algorithms
  - Hyperparameters
  - Representations (Modalities)
  - Training sets

# How models are different?

- Difference in population
- Difference in hypothesis
- Difference in modeling technique
- Difference in initial seed

# Why ensembles ?

- There are two main reasons to use an ensemble over a single model, and they are related; they are:
  – Performance: An ensemble can make better predictions and achieve better performance than any single contributing model.
  – Robustness: An ensemble reduces the spread or dispersion of the predictions and model performance.

# Model Error

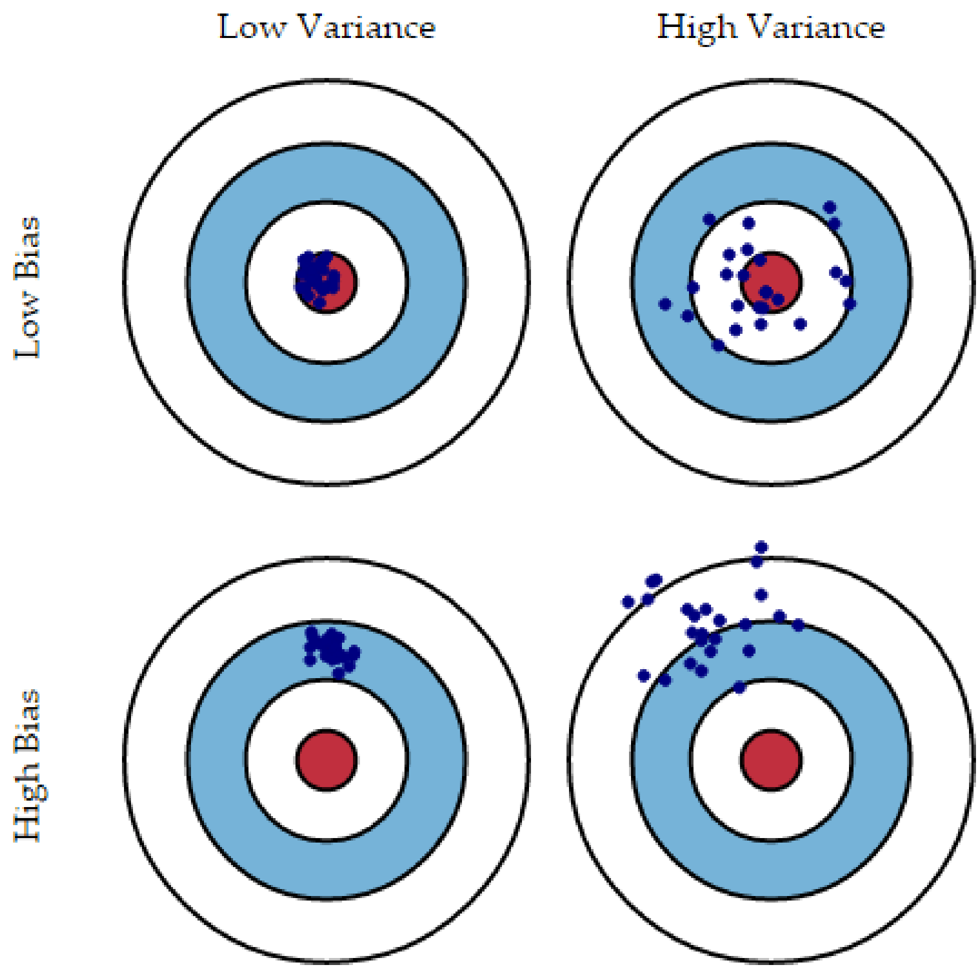
- The error emerging from any machine model can be broken down into three components mathematically. Following are these component:

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\hat{f}(x) - E[\hat{f}(x)]\right]^2 + \sigma_e^2$$

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

- **Bias error** is useful to quantify how much on an average are the predicted values different from the actual value. A high bias error means we have an under-performing model which keeps on missing essential trends.
- **Variance** on the other side quantifies how are the prediction made on the same observation different from each other. A high variance model will over-fit on your training population and perform poorly on any observation beyond training.

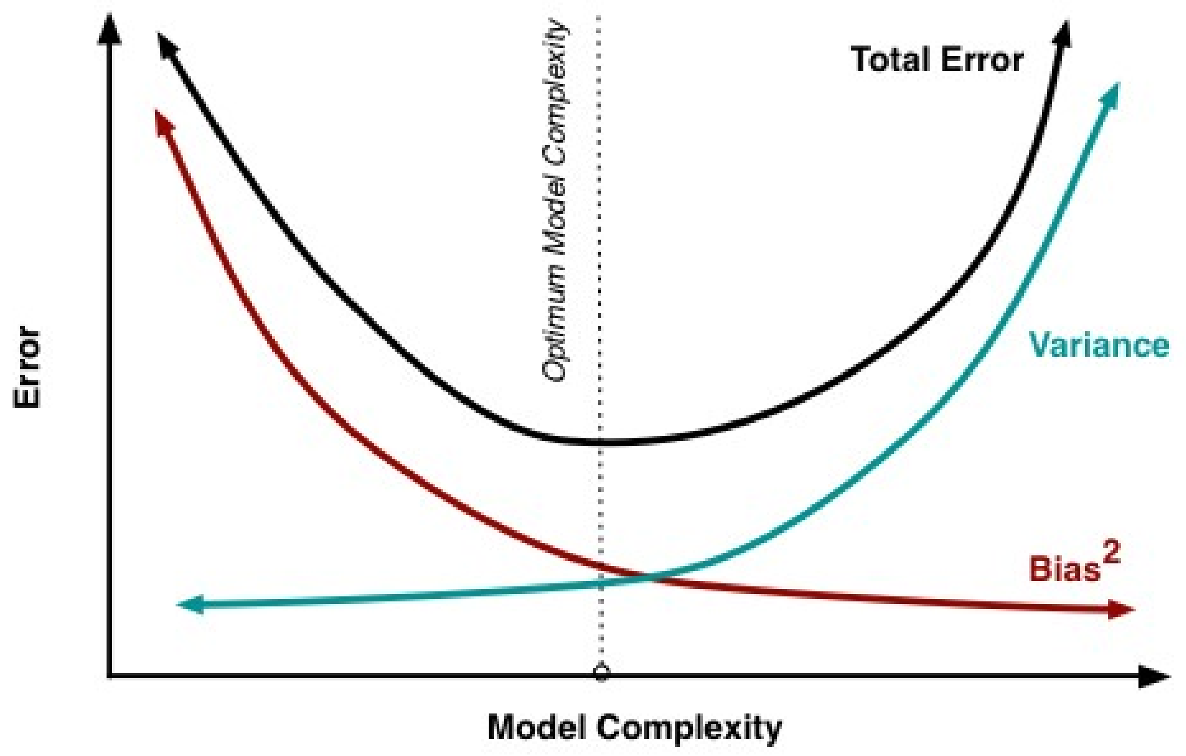
# Bias and Variance



# Bias – Variance Tradeoff

- Normally, as you increase the complexity of your model, you will see a reduction in error due to lower bias in the model. However, this only happens till a particular point.
- As you continue to make your model more complex, you end up over-fitting your model and hence your model will start suffering from high variance.
- A champion model should maintain a balance between these two types of errors. This is known as the trade-off management of bias-variance errors. Ensemble learning is one way to execute this trade off analysis.

# Bias – Variance Tradeoff



# Ensemble Creation Approaches

- Unweighted Voting (e.g. Bagging)
- Weighted voting – based on accuracy (e.g. Boosting), Expertise, etc.
- Stacking - Learn the combination function

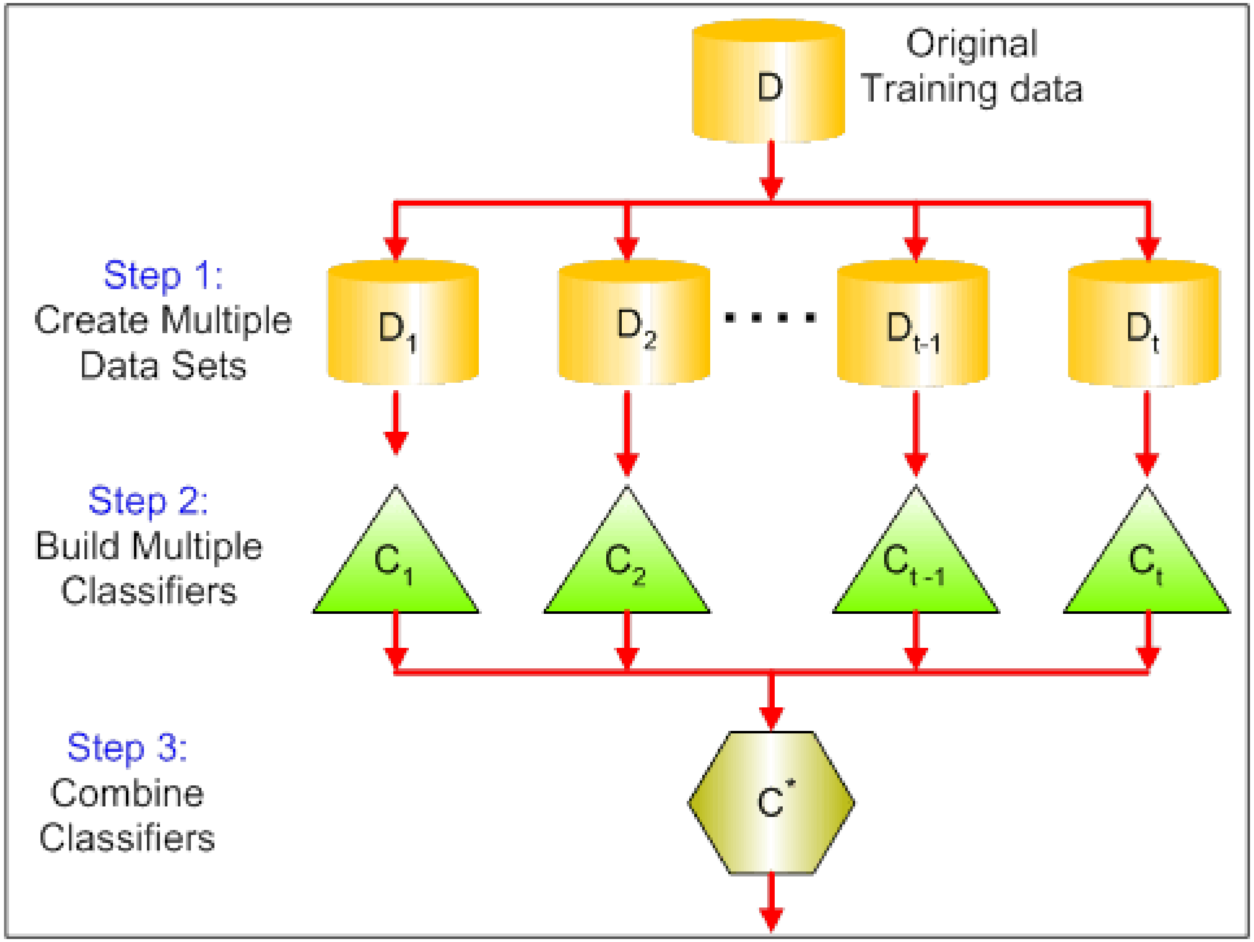
# Ensemble Learning Methods

- Bagging
- Boosting
- Stacking

# Bagging

- Bagging stands for bootstrap aggregation.
- It combines multiple learners in a way to reduce the variance of estimates.
- For example, random forest trains M Decision Tree, you can train M different trees on different random subsets of the data and perform voting for final prediction.
- Example:
  - Random Forest
  - Extra Trees.

# Bagging



Original Training data
D
Step 1:
Create Multiple Data Sets
$D_1$
$D_2$
$D_{t-1}$
$D_t$
Step 2:
Build Multiple Classifiers
$C_1$
$C_2$
$C_{t-1}$
$C_t$
Step 3:
Combine Classifiers
$C^*$

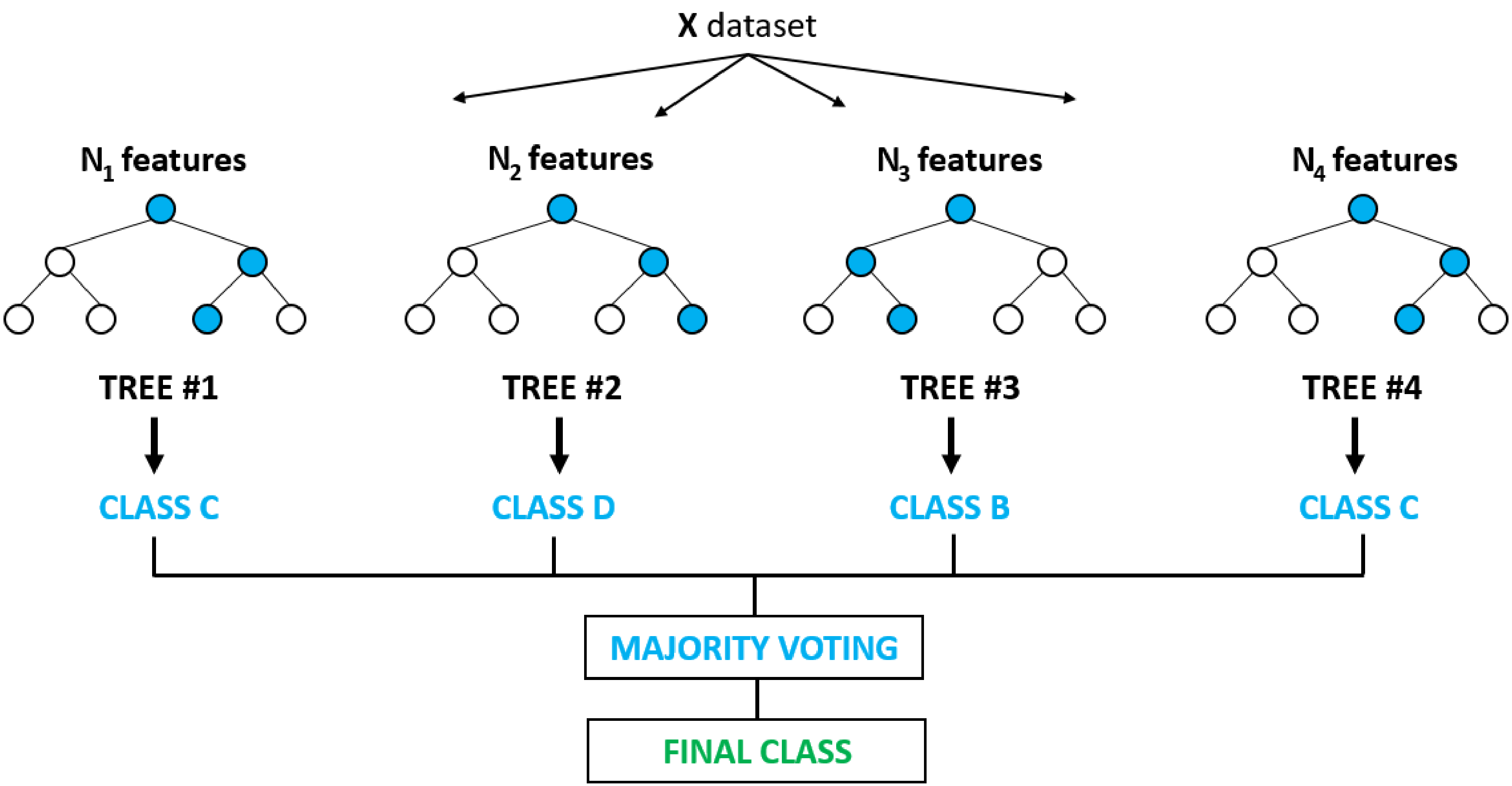
# Random Forest

- Random forest is a type of supervised machine learning algorithm based on ensemble learning.
- Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model.
- The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest".
- The random forest algorithm can be used for both regression and classification tasks.

# How it works ?

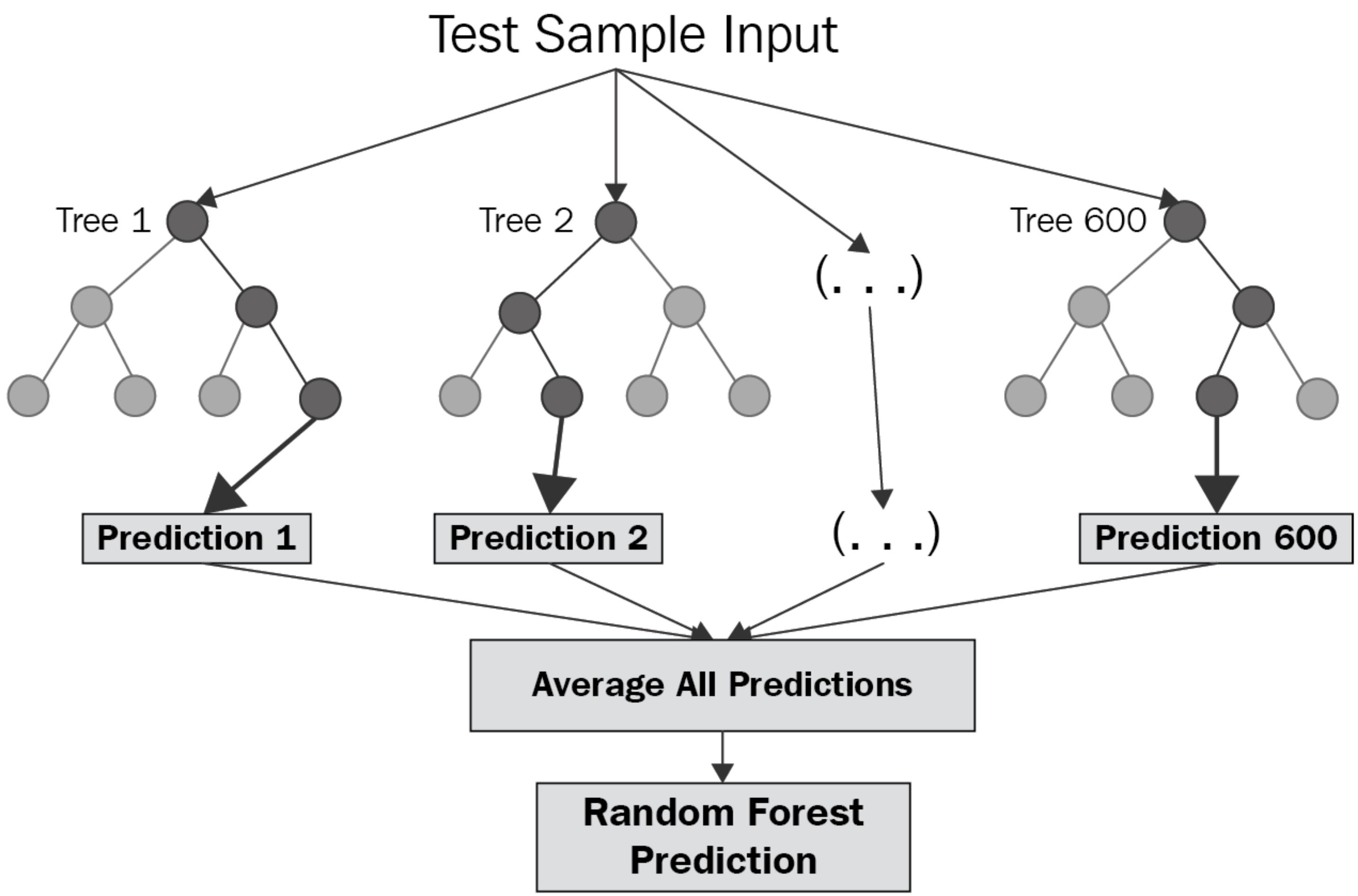
- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
- In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

# Majority Voting




Source: medium.com

# Regressor Output




Source: medium.com

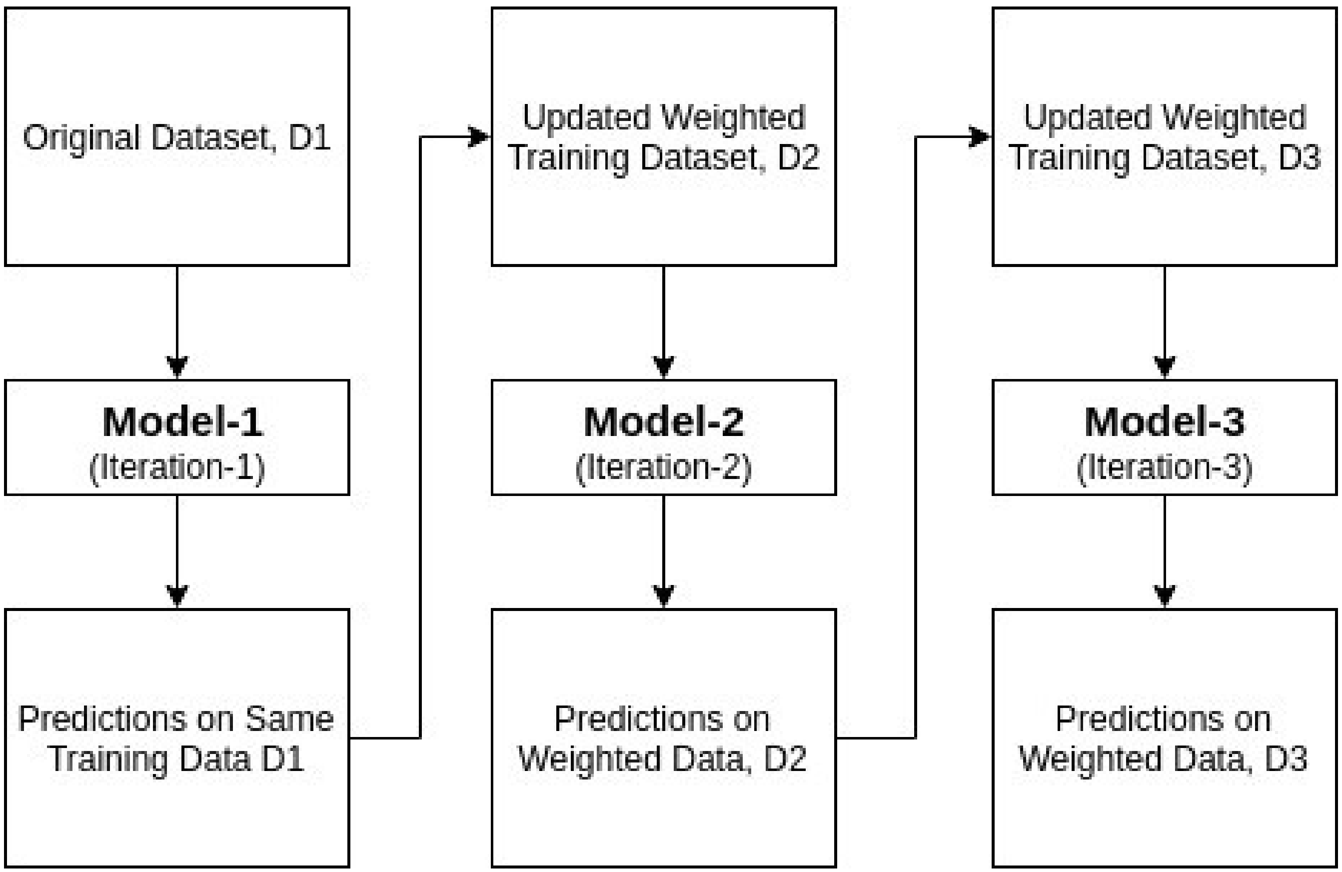
# Boosting

- Boosting algorithms are a set of the low accurate classifier to create a highly accurate classifier.
- Low accuracy classifier (or weak classifier) offers the accuracy better than the flipping of a coin.
- This is done by building a model from the training data, then creating a second model that attempts to correct the errors from the first model. Models are added until the training set is predicted perfectly or a maximum number of models are added.
- Highly accurate classifier( or strong classifier) offer error rate close to 0. Boosting algorithm can track the model who failed the accurate prediction.
- Boosting algorithms are less affected by the overfitting problem.

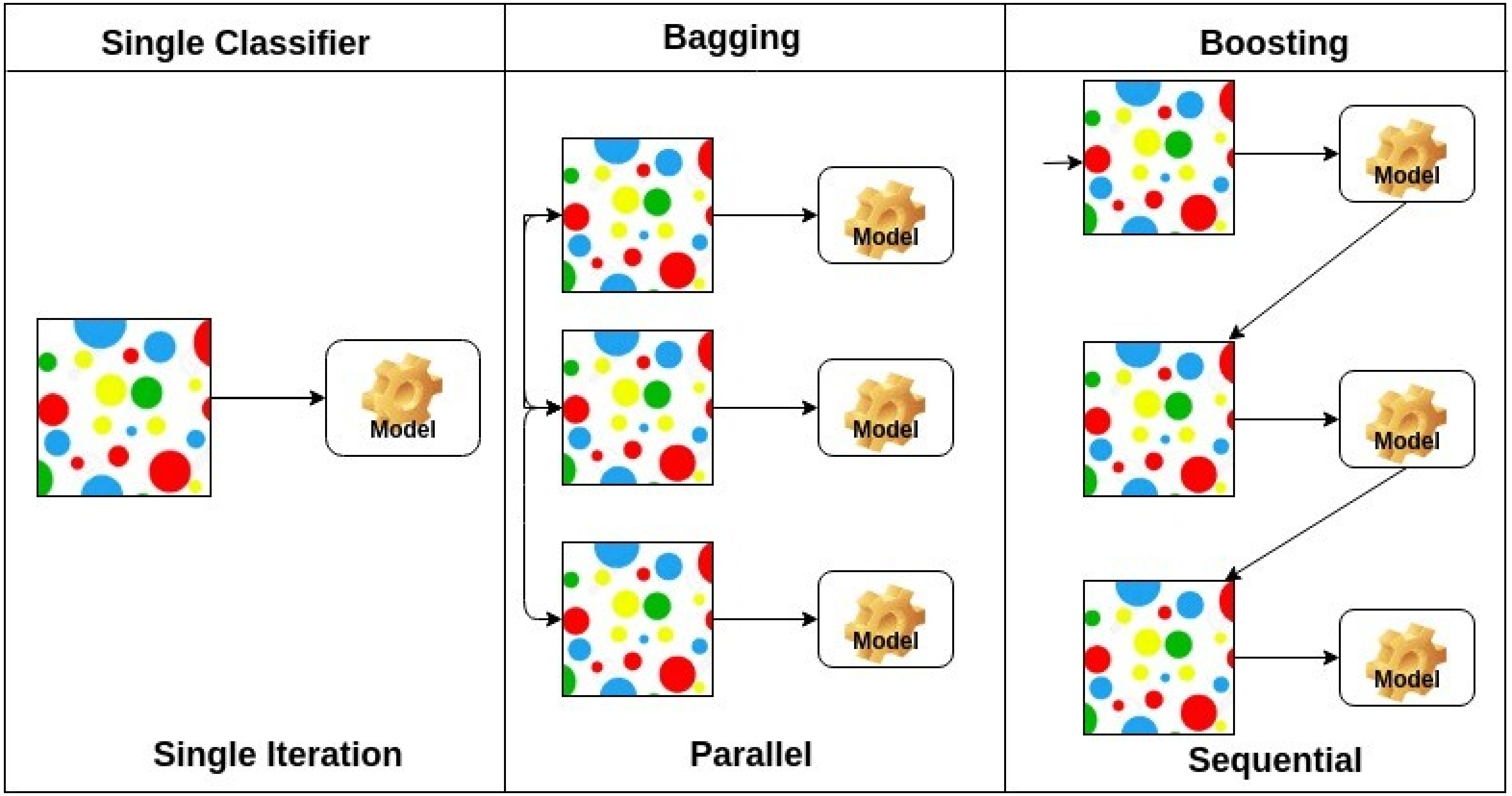
# Boosting Models

- Models that are typically used in Boosting technique are:
  - XGBoost (Extreme Gradient Boosting)
  - GBM (Gradient Boosting Machine)
  - ADABoost (Adaptive Boosting)

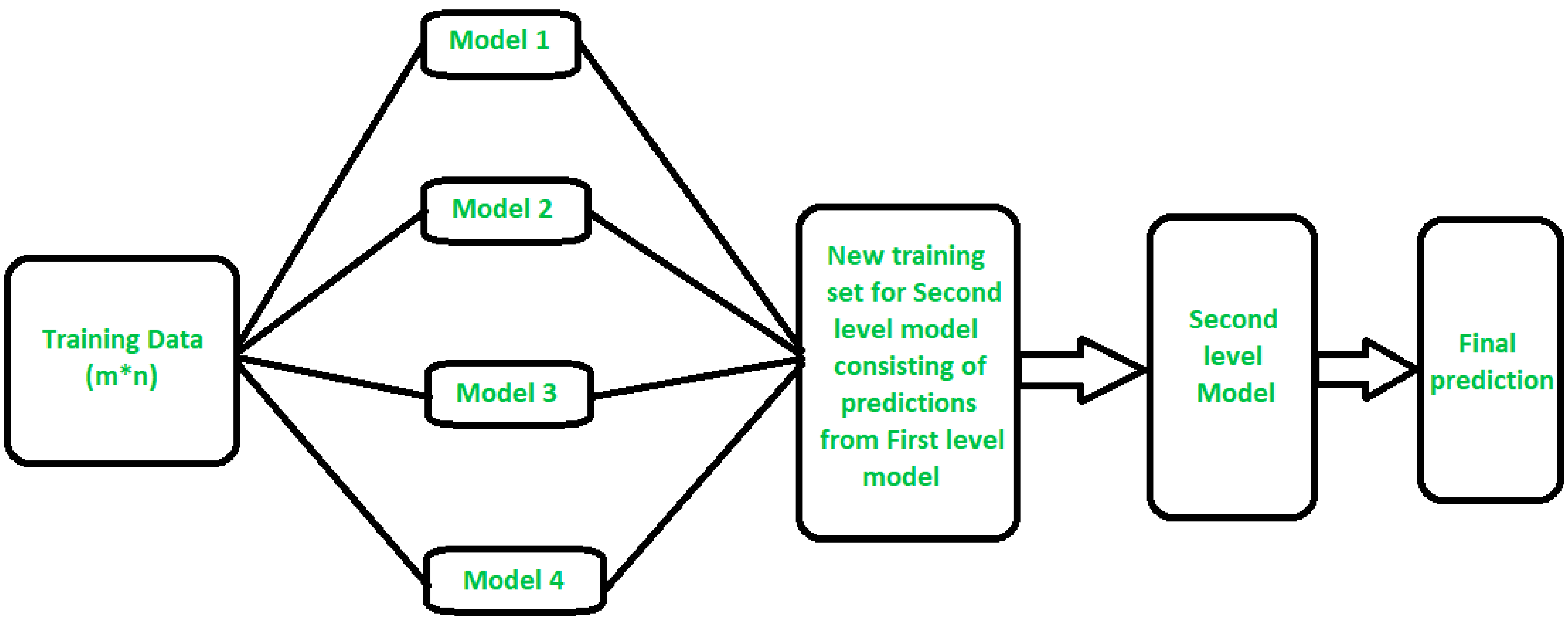
# How Adaboost Works?



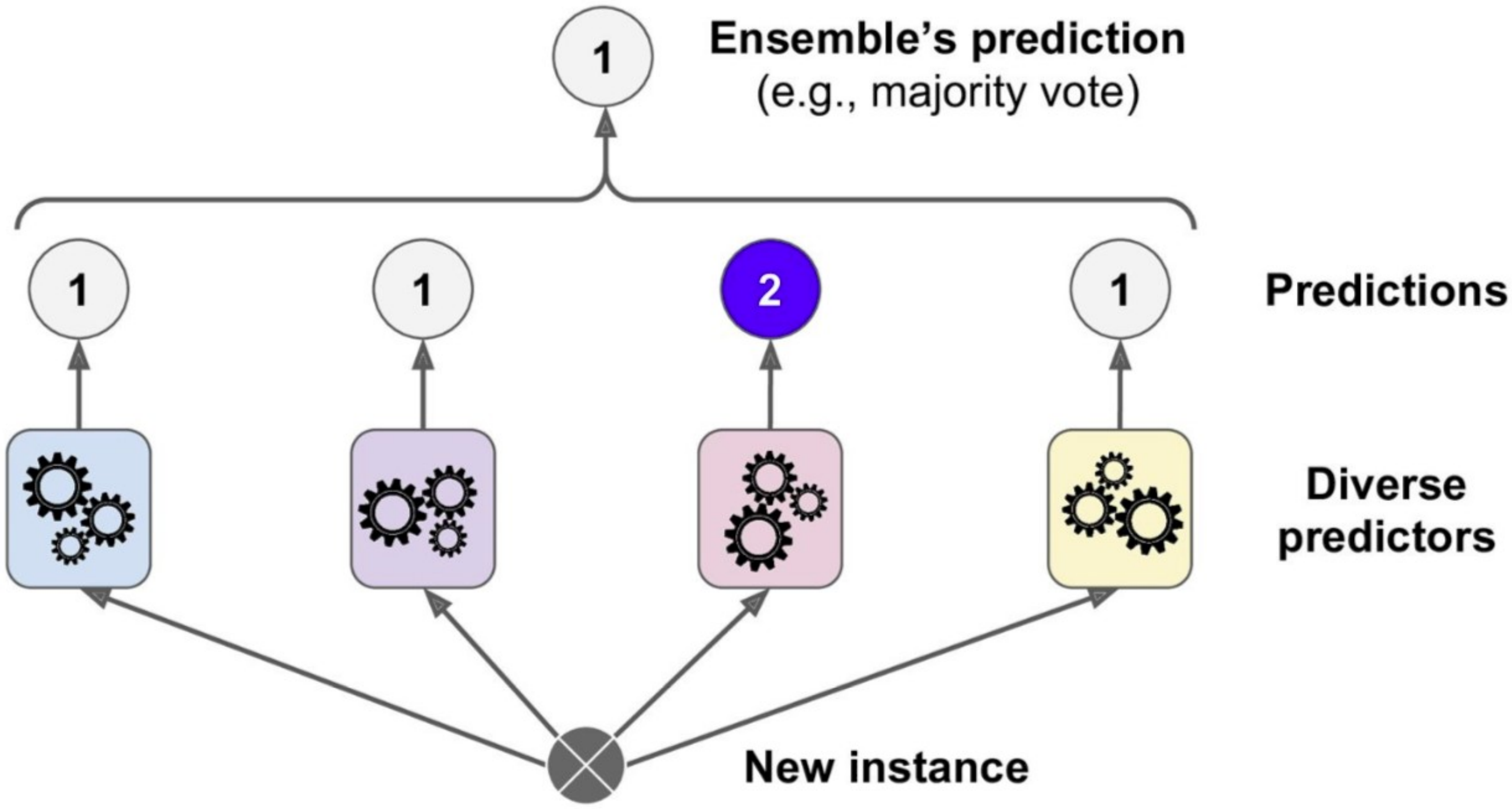
# Comparison


Single Classifier
Bagging
Boosting
Model
Model
Model
Model
Model
Model
Model
Single Iteration
Parallel
Sequential

# Stacking


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

# Voting


1
Ensemble's prediction
(e.g., majority vote)
1
1
2
1
Predictions
Diverse
predictors
New instance

# Useful resources

- www.mitu.co.in
- www.pythonprogramminglanguage.com
- www.scikit-learn.org
- www.towardsdatascience.com
- www.medium.com
- www.analyticsvidhya.com
- www.kaggle.com
- www.stephacking.com
- www.github.com

# Thank you



@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

c/MITUSkillologies

**Web Resources**
https://mitu.co.in
http://tusharkute.com

**contact@mitu.co.in**

**tushar@tusharkute.com**