Ensemble Learning Methods

Penyusun Modul: Chairul Aulia

Editor: Citra Chairunnisa

Ensemble Learning

Ensemble models in machine learning **combine the decisions** from multiple models to **improve** the overall performance.
The main causes of error in learning models are due to **noise, bias**, and **variance**.
Ensemble methods **help to minimize** these factors. These methods are designed to improve the stability and accuracy of Machine Learning algorithms.

Bias

Bias is a phenomenon that **skews the result** of an algorithm in favor or against an idea.
Bias is considered a **systematic error** that occurs in the machine learning model itself due to **incorrect assumptions** in the ML process.
Technically, we can define bias as the error between average model prediction and the ground truth. Moreover, it describes how well the model matches the training data set:

- A model with a higher bias would not match the data set closely.
- A low bias model will closely match the training data set.

Characteristics of a high bias model include:

- Failure to capture proper data trends
- Potential towards underfitting
- More generalized/overly simplified
- High error rate

Variance

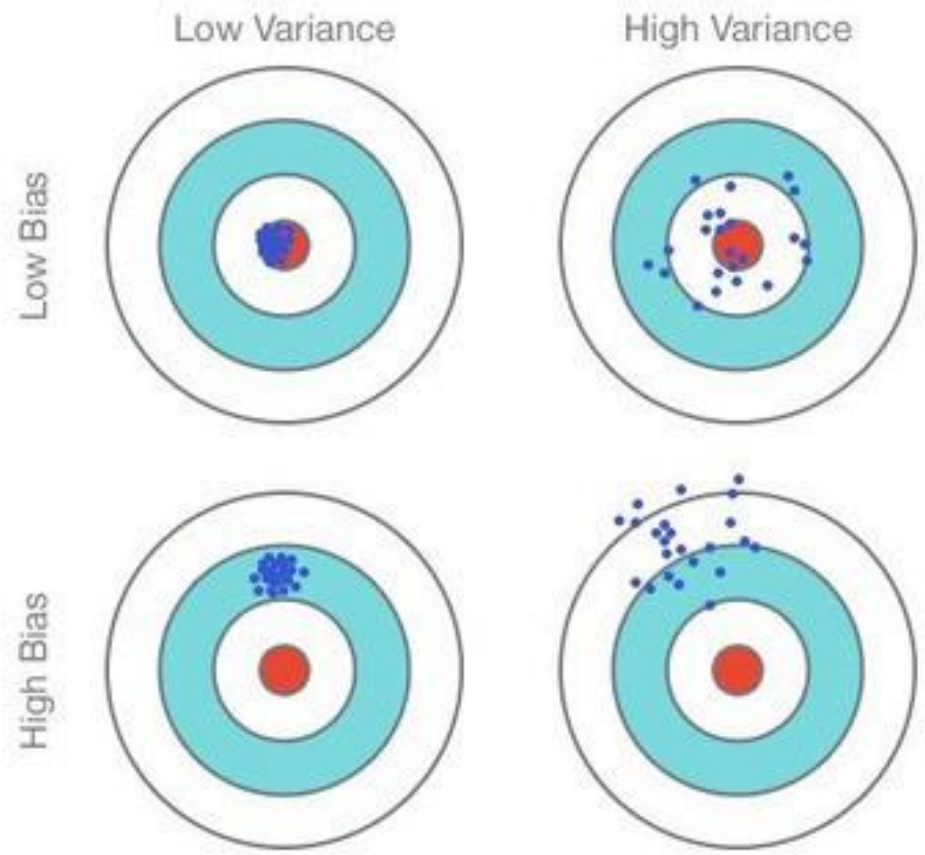
Variance refers to the changes in the model when using different portions of the training data set.
Simply stated, variance is the **variability** in the model prediction—how much the ML function can adjust depending on the given data set. Variance comes from highly complex models with a large number of features.

- Models with high bias will have low variance. For instance, a model that does not match a data set with a high bias will create an inflexible model with a low variance.
- Models with high variance will have a low bias.

Characteristics of a high variance model include:

- Noise in the data set
- Potential towards overfitting
- Complex models
- Trying to put all data points as close as possible

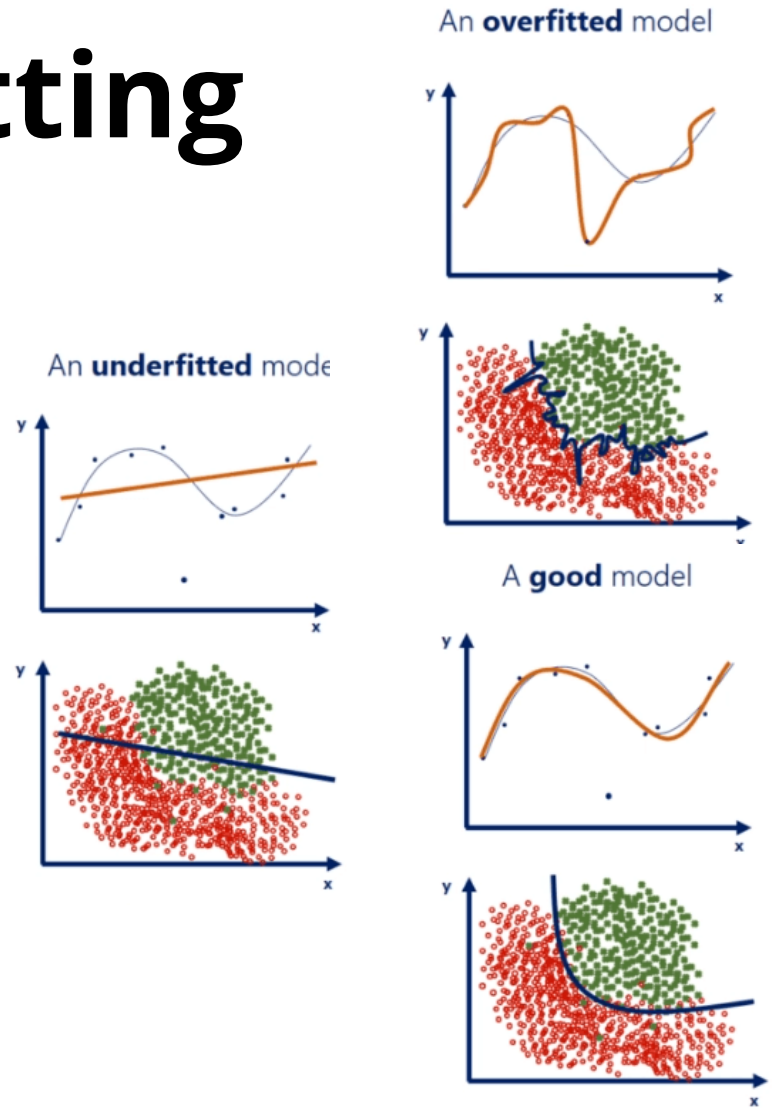
Bias and Variance



https://medium.com/@mp32445/understanding-bias-variance-tradeoff-ca59a22e2a83

Underfitting and Overfitting

- **Underfitting** happens when a model **unable to capture** the underlying pattern of the data. These models usually have high bias and low variance. It happens when we have very **less amount of data** to build an accurate model or when we try to **build a linear model with a nonlinear** data. Also, these kind of models are very simple to capture the complex patterns (like Linear and logistic regression)
- **Overfitting** happens when our model **captures the noise** along with the underlying pattern in data. It happens when we train our model a lot over the noisy dataset. These models have low bias and high variance. These models are very complex (like Decision trees) which are prone to overfitting.



https://skillplus.web.id/underfitting-dan-overfitting-model/

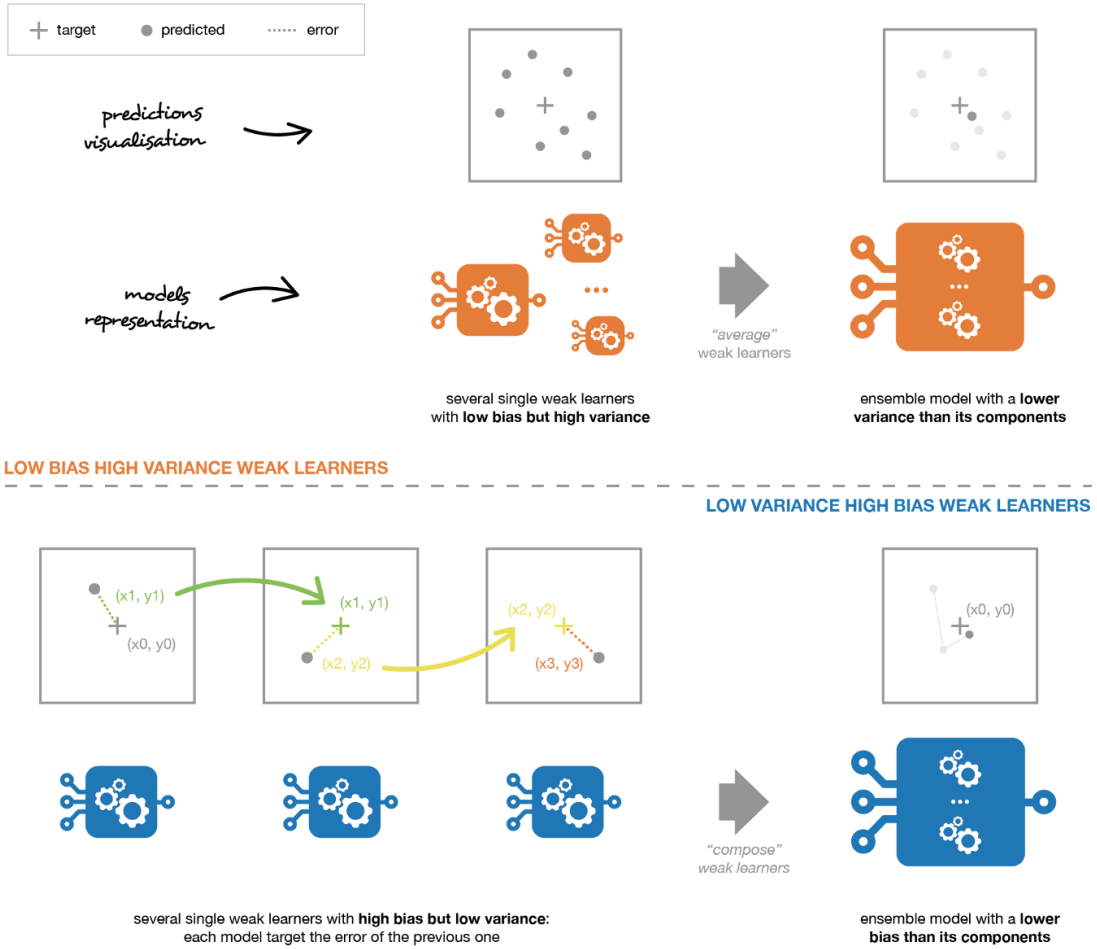
Bias-Variance Tradeoff

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand, if our model has a large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.
This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time.
There is no escaping the relationship between bias and variance in machine learning.

There are some ways to tackle this tradeoff, one of them is using Ensemble methods

Weak Learners

- In ensemble learning theory, we call weak learners (or base models) models that can be used as building blocks for designing more complex models by combining several of them.
- The idea of ensemble methods is to try reducing bias and/or variance of such weak learners by combining several of them together in order to create a strong learner (or ensemble model) that achieves better performances.
- Our choice of weak learners should be **coherent with the way we aggregate these models**. For example If we choose base models with low bias, it should be with an aggregating method that tends to reduce variance



Simple Ensemble Techniques

1. Taking the mode of the results/Max Voting

The mode is a statistical term that refers to the most frequently occurring number found in a set of numbers.
In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a separate vote. The prediction which we get from the majority of the models is used as the final prediction.

Example:

```
model1 = DecisionTreeClassifier()
model2 = KNeighborsClassifier()
model3= LogisticRegression()

model1.fit(X_train,y_train.ravel())
model2.fit(X_train,y_train.ravel())
model3.fit(X_train,y_train.ravel())

pred1=model1.predict(X_test)
pred2=model2.predict(X_test)
pred3=model3.predict(X_test)
mode_pred=mode([pred1, pred2, pred3])[0][0]
```

or

```
from sklearn.ensemble import VotingClassifier
model = VotingClassifier(estimators=[('dt', model1), ('knn', model2), ('lr', model2)], voting='hard')
model.fit(X_train,y_train)
model.score(X_test,y_test)
```

Simple Ensemble Techniques

2. Taking the average of the results
In this technique, we take an average of predictions from all the models and use them to make the final prediction.

3. Taking the weighted average of the results
This is an extension of the averaging method. All models are assigned different weights defining the importance of each model for prediction.

Example:

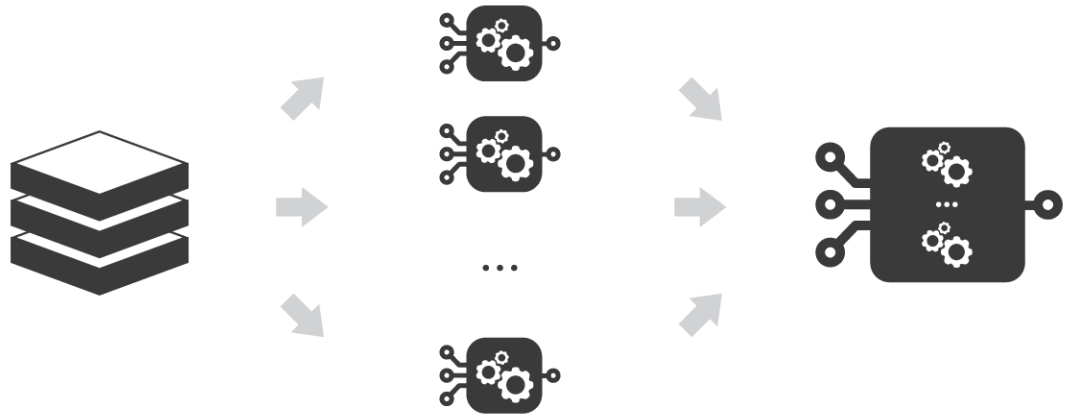
```
model1 = DecisionTreeRegressor()
model2 = KNeighborsRegressor()
model3= LogisticRegression()

model1.fit(X_train,y_train.ravel())
model2.fit(X_train,y_train.ravel())
model3.fit(X_train,y_train.ravel())

pred1=model1.predict(X_test)
pred2=model2.predict(X_test)
pred3=model3.predict(X_test)
ave_pred=(pred1+pred2+pred3)/3
weighted_ave_pred=(pred1*0.3+pred3*0.7)
```

Advanced Ensemble Techniques

- Stacking
- Bagging (Bootstrap AGGregatING)
- Boosting

Stacking

- Stacking often considers **heterogeneous weak learners** (different learning algorithms are combined) whereas bagging and boosting consider mainly homogeneous.
- Stacking learns to combine the base models using a meta-model whereas bagging and boosting combine weak learners following deterministic algorithms.

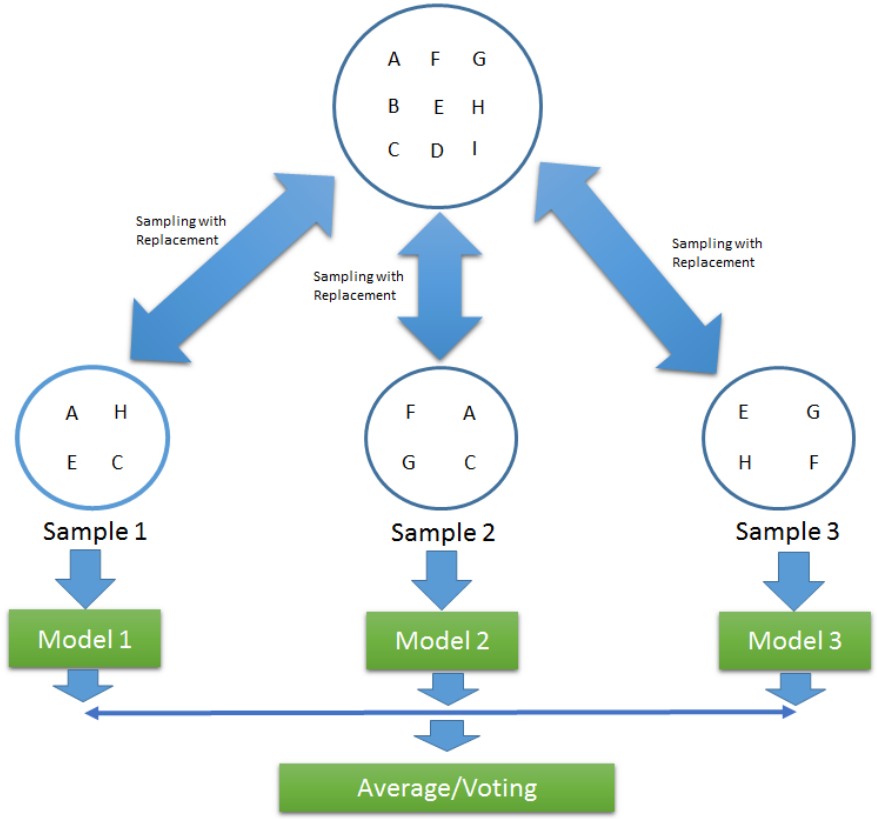


https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Bagging (Bootstrap Aggregating)

Bootstrapping is a technique of **sampling different sets** of data from a given training set by using replacement. After bootstrapping the training dataset, we train the model on all the different sets and **aggregate** the result.

For aggregating the outputs of base learners, bagging uses majority voting (most frequent prediction among all predictions) for classification and averaging (mean of all the predictions) for regression.



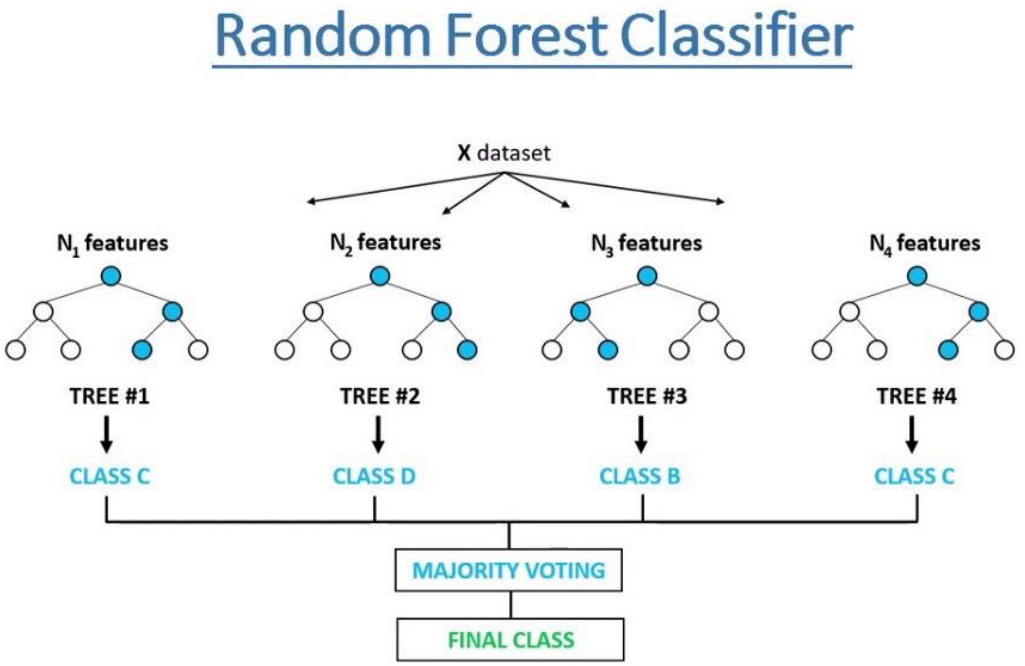
https://medium.com/analytics-vidhya/ensemble-methods-bagging-boosting-and-stacking-28d006708731

Bagging (Bootstrap Aggregating)

- Significantly decreases the variance without increasing bias.
- Bagging methods work so well because of diversity in the training data since the sampling is done by bootstrapping.
- If the training set is very huge, it can save computational time by training the model on a relatively smaller data set and still can increase the accuracy of the model.
- Works well with small datasets as well.

Random Forest

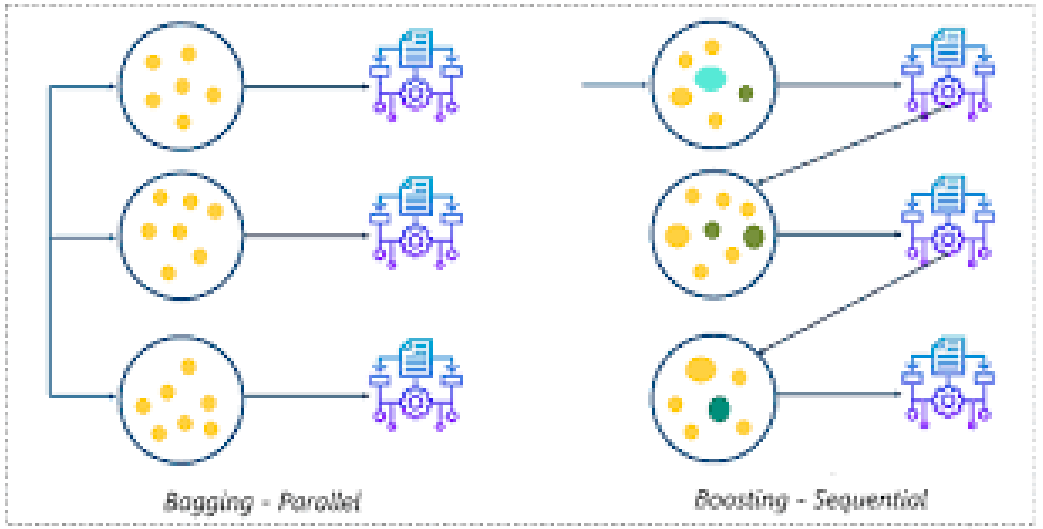
- Random Forest Model can be thought of as BAGGing, with a slight tweak. It decides where to split based on a **random selection of features**. Rather than splitting at similar features, it implements a level of differentiation because each tree will split based on different features
- Sampling over features makes all trees do not look at the exact same information to make decisions, so, it reduces the correlation between the different returned outputs. Another advantage of sampling over the features is that **it makes the decision making process more robust to missing data**



https://medium.com/@17611104_/mengenal-random-forest-dengan-rstudio-39e9f2c0c9df

Boosting

The idea of boosting is to train weak learners **sequentially**, each trying to correct its predecessor.
In boosting, the data set is weighted, so that observations that were **incorrectly classified** by classifier n are **given more importance** in the training of model n + 1, while in bagging the training samples are taken randomly from the whole population Ada Boost(Adaptive Boosting), Gradient Boosting, XG Boost(Xtreme Gradient Boosting) are few common examples of Boosting Techniques.



https://medium.com/@maniyaswanth123/bagging-and-boosting-a0b39e312117

Boosting

- Boosting is mainly focused on reducing bias.
- It relies on creating a series of weak learners each of which might not be good for the entire data set but is good for some part of the data set. Thus, each model actually boosts the performance of the ensemble.
- A disadvantage of boosting is that it is sensitive to outliers since every classifier is obliged to fix the errors in the predecessors. Thus, the method is too dependent on outliers.
- Another disadvantage is that the method is almost impossible to scale up. This is because every estimator bases its correctness on the previous predictors, thus making the procedure difficult to streamline.

Exercise

Explore some ensemble methods we've learned today using sklearn.ensemble module! Compare the results with other weak learners!

You can use whatever dataset you have or the sample dataset from sklearn