15 Ensemble Methods

- Ensemble methods are techniques that combine multiple machine learning models to improve overall performance
- The idea is that by integrating different models, you can reduce errors, increase accuracy, and enhance robustness compared to using a single model
- There are different ways to create an ensemble, and they generally fall into two categories : **Bagging** and **Boosting**

01 Bagging (Bootstrap Aggregating)

- Bagging is a parallel ensemble method that reduces variance by training multiple models on different subsets of the data and averaging their predictions
- The key idea is to create multiple versions of the original dataset using bootstrap sampling (random sampling with replacement)
- Each model is trained independently on its own dataset, and the final prediction is made by averaging (for regression) or voting (for classification)

Steps

1. **Bootstrap Sampling**
    - Create multiple datasets by randomly sampling with replacement from the original dataset
    - This means that some data points may appear more than once, while others may not appear at all
2. **Model Training :** Train a separate model on each bootstrap sample
3. **Aggregation:**
    - For regression tasks, average the predictions of all models
    - For classification tasks, use majority voting

Advantages

- **Reduces variance :** By averaging multiple models, bagging reduces the risk of overfitting
- **Simple to implement :** It can be applied to any machine learning algorithm

Common Example

- **Random Forest:** A popular bagging technique that involves training multiple decision trees on different subsets of the data and aggregating their predictions

02 Boosting

- Boosting is a sequential ensemble method that reduces bias by combining weak learners to create a strong learner
- Unlike bagging, boosting focuses on correcting the errors of the previous models by assigning more weight to misclassified instances
- Each new model is trained to focus on the errors made by the previous models, and the final prediction is a weighted sum of all models

Steps

1. **Initialize Weights :** Start by assigning equal weights to all instances in the dataset
2. **Model Training :** Train a weak learner (usually a simple model like a shallow decision tree) on the data
3. **Error Calculation :** Evaluate the model's performance, and increase the weights of the misclassified instances to make them more important in the next iteration
4. **Iterate :** Repeat the process, creating multiple weak learners, each focusing more on the previously misclassified data
5. **Combine :** Combine all weak learners' predictions to form a strong learner. The combination is usually a weighted sum of the individual models' predictions

Advantages

- **Reduces bias :** Boosting can significantly reduce errors by focusing on hard-to-classify instances
- **High accuracy :** Boosting often achieves higher accuracy than bagging, especially on complex datasets

Common Examples

- **AdaBoost :** Adaptive Boosting adjusts the weights of incorrectly classified instances, focusing on difficult cases
- **Gradient Boosting :** Uses gradient descent to minimize the error of the ensemble in a more sophisticated way
- **XGBoost, LightGBM, CatBoost :** Variants of gradient boosting that are highly optimized for performance and efficiency

Comparison

- **Bagging :** Reduces variance, works well with high-variance models (e.g., decision trees), and is effective in reducing overfitting

- **Boosting :** Reduces bias, works well with weak learners, and can achieve better accuracy but is more prone to overfitting if not properly regularized