Artificial Intelligence Fullstack
[Course]
Week 5 – Machine Learning –

- ❑ Ensemble Learning & its Types
- ❑ Random Forest Algorithm

[See examples / code in GitHub code repository]

It is not about Theory, it is 20% Theory and 80% Practical – Technical/Development/Programming [Mostly Python based]

ML – Ensemble Learning

Ensemble learning is a method where we use many small models instead of just one. Each of these models may not be very strong on its own, but when we put their results together, we get a better and more accurate answer. It's like asking a group of people for advice instead of just one person—each one might be a little wrong, but together, they usually give a better answer.

Types of Ensembles Learning in Machine Learning

Bagging (Bootstrap Aggregating):

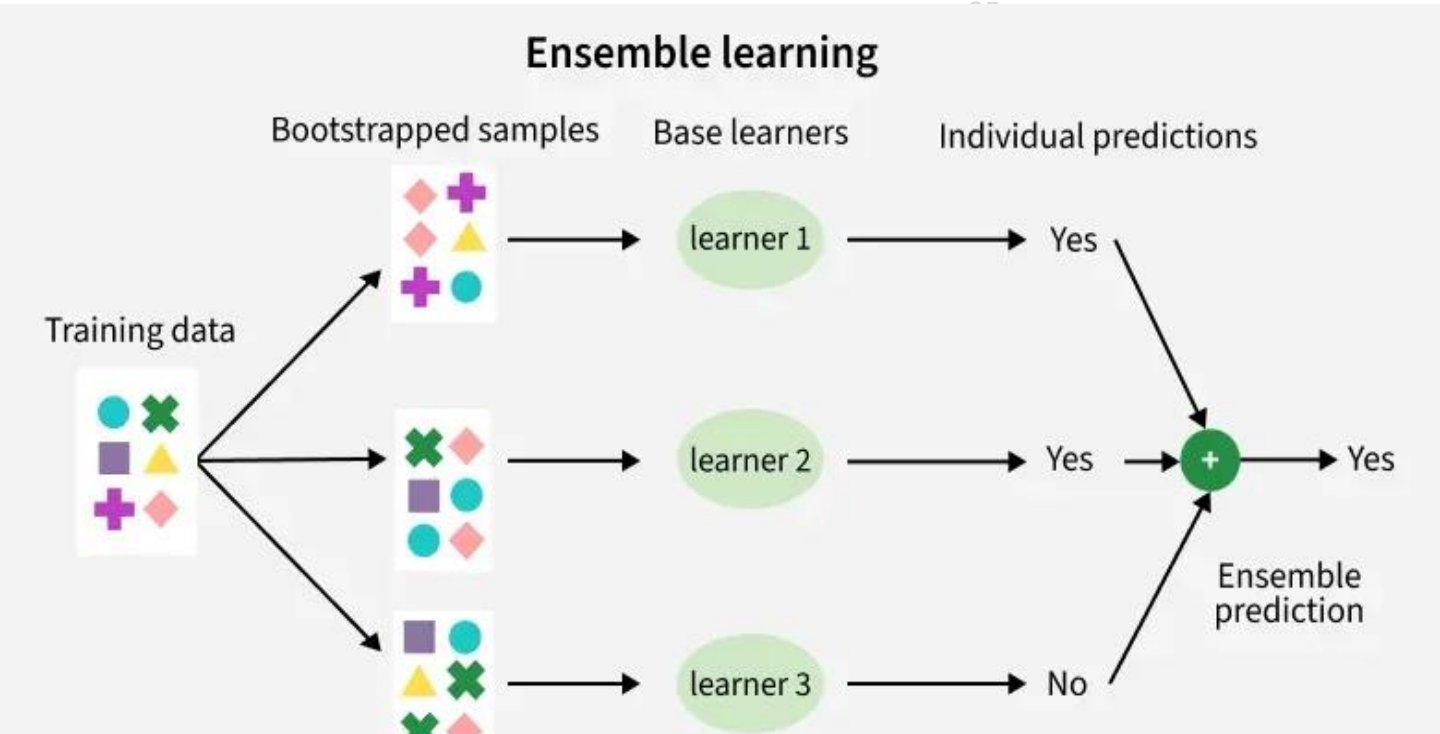
Models are trained independently on different random subsets of the training data. Their results are then combined—usually by averaging (for regression) or voting (for classification). This helps reduce variance and prevents overfitting.

Boosting:

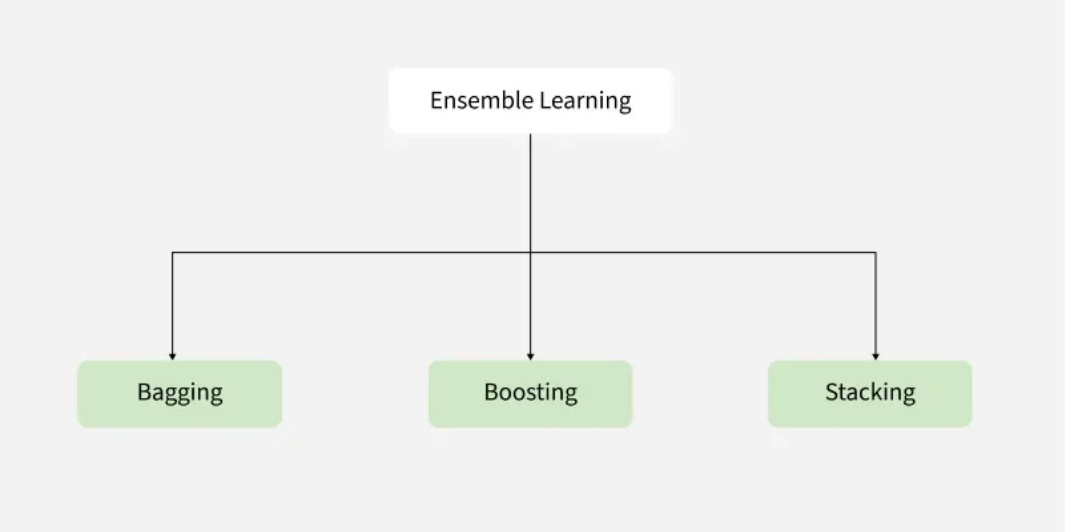
Models are trained one after another. Each new model focuses on fixing the errors made by the previous ones. The final prediction is a weighted combination of all models, which helps reduce bias and improve accuracy.

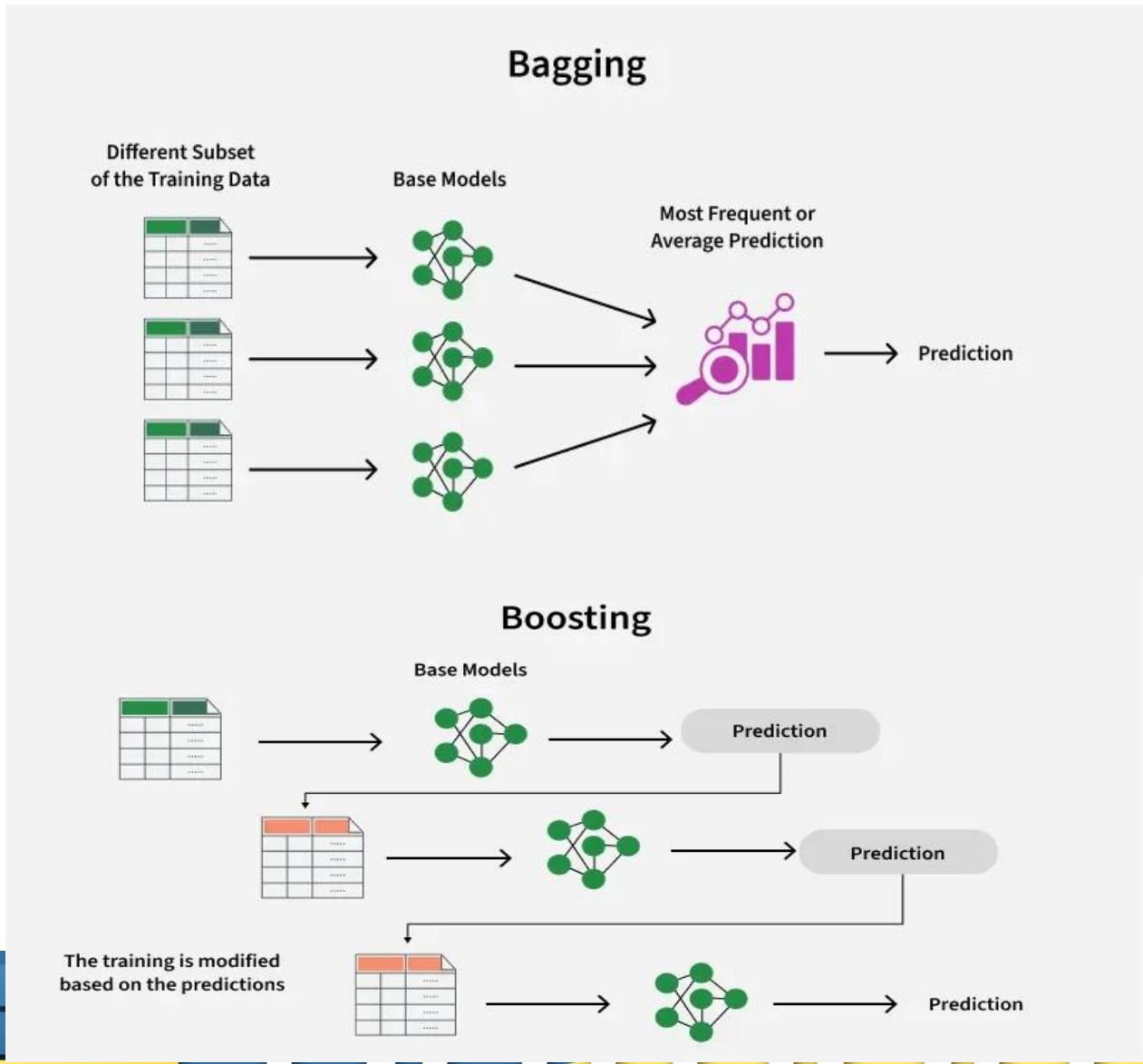
Stacking (Stacked Generalization):

Multiple different models (often of different types) are trained, and their predictions are used as inputs to a final model, called a meta-model. The meta-model learns how to best combine the predictions of the base models, aiming for better performance than any individual model.

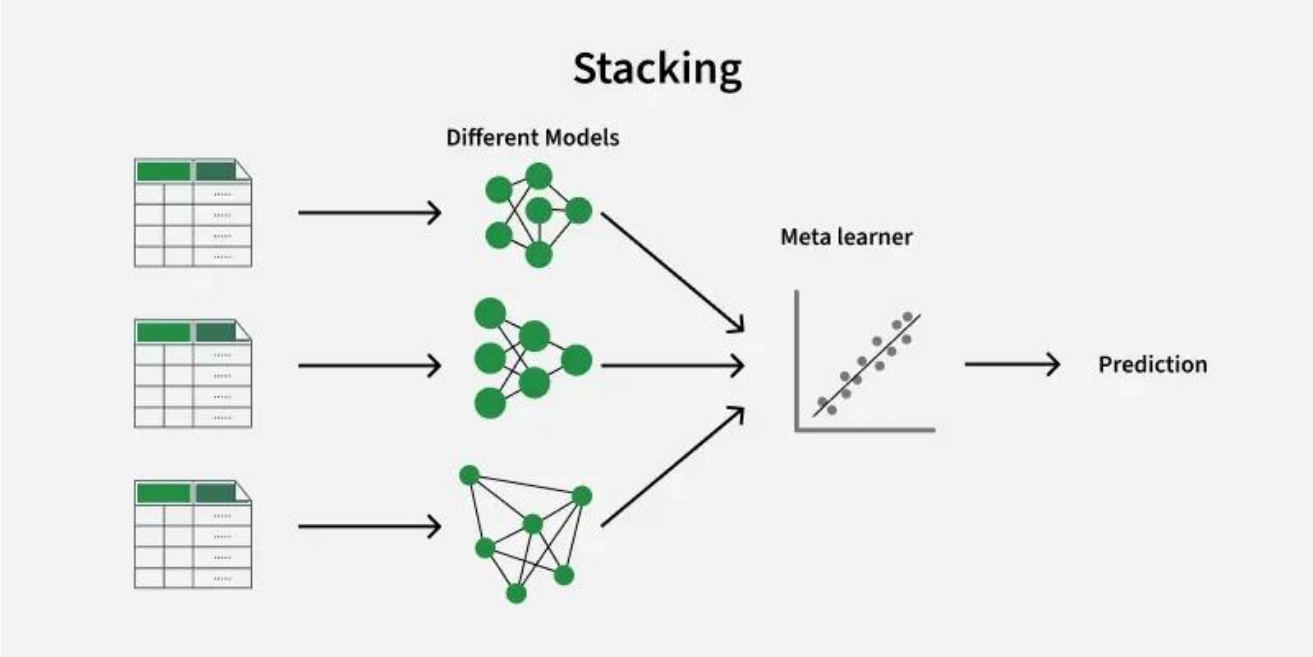


ML – Ensemble Learning





ML – Ensemble Learning



Benefits of Ensemble Learning in Machine Learning

Ensemble learning is a versatile approach that can be applied to machine learning model for: -

- **Reduction in Overfitting**: By aggregating predictions of multiple model's ensembles can reduce overfitting that individual complex models might exhibit.
- **Improved Generalization**: It generalizes better to unseen data by minimizing variance and bias.
- **Increased Accuracy**: Combining multiple models gives higher predictive accuracy.
- **Robustness to Noise**: It mitigates the effect of noisy or incorrect data points by averaging out predictions from diverse models.
- **Flexibility**: It can work with diverse models including decision trees, neural networks and support vector machines making them highly adaptable.
- **Bias-Variance Tradeoff**: Techniques like bagging reduce variance, while boosting reduces bias leading to better overall performance.

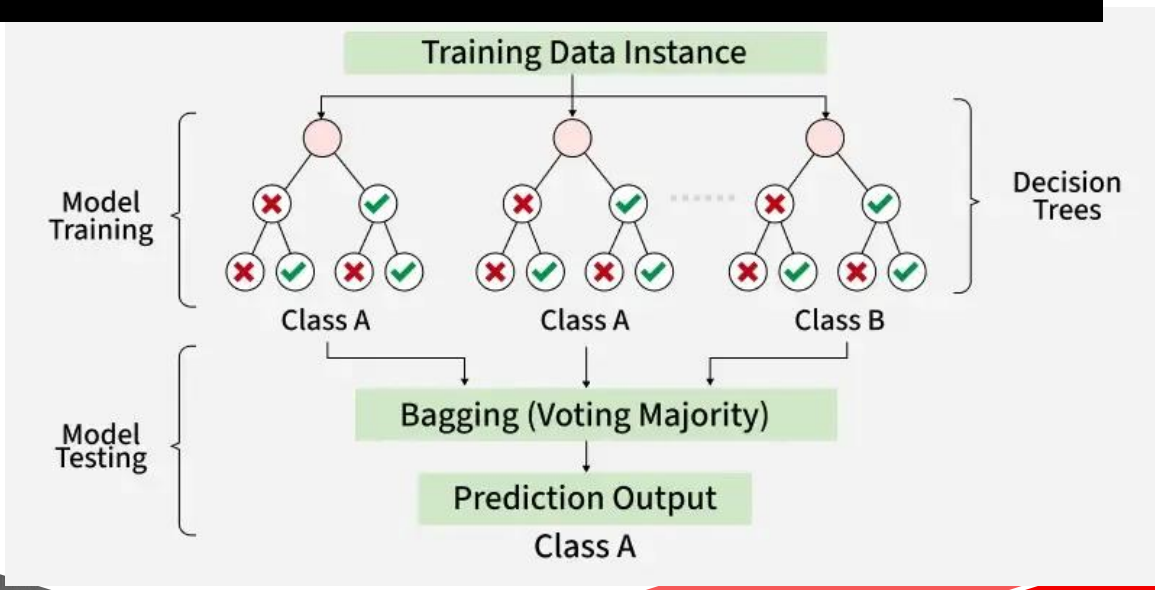
ML – Random Forest for Classification

Random Forest is a machine learning algorithm that uses many decision trees to make better predictions. Each tree looks at different random parts of the data and their results are combined by voting for classification or averaging for regression. This helps in improving accuracy and reducing errors.

Implementing **Random Forest for Classification Tasks**

Here we will predict survival rate of a person in titanic.

- ❑ Import libraries and load the Titanic dataset.
- ❑ Remove rows with missing target values ('Survived').
- ❑ Select features like class, sex, age, etc and convert 'Sex' to numbers.
- ❑ Fill missing age values with the median.
- ❑ Split the data into training and testing sets, then train a Random Forest model.
- ❑ Predict on test data, check accuracy and print a sample prediction result.



python

ML – Random Forest for Regression

Implementing Random Forest for Regression Tasks

- ❑ We will do house price prediction here.
- ❑ Load the California housing dataset and create a DataFrame with features and target.
- ❑ Separate the features and the target variable.
- ❑ Split the data into training and testing sets (80% train, 20% test).
- ❑ Initialize and train a Random Forest Regressor using the training data.
- ❑ Predict house values on test data and evaluate using MSE and R^2 score.
- ❑ Print a sample prediction and compare it with the actual value.

Advantages of Random Forest

- ❑ Random Forest provides very accurate predictions even with large datasets.
- ❑ Random Forest can handle missing data well without compromising with accuracy.
- ❑ It doesn't require normalization or standardization on dataset.
- ❑ When we combine multiple decision trees it reduces the risk of overfitting of the model.

Limitations of Random Forest

- ❑ It can be computationally expensive especially with a large number of trees.
- ❑ It's harder to interpret the model compared to simpler models like decision trees.

https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/
https://www.ibm.com/think/topics/random-forest

ML | Difference Between Boosting Algorithms

Algorithms	Gradient Boosting	AdaBoost	XGBoost	CatBoost	LightGBM
Year	-	1995	2014	2017	2017
Handling Categorical Variables	May require preprocessing like one-hot encoding	No	NO	Automatically handles categorical variables	No
Speed/Scalability	Moderate	Fast	Fast	Moderate	Fast
Memory Usage	Moderate	Low	Moderate	High	Low
Regularization	NO	No	Yes	Yes	Yes
Parallel Processing	No	No	Yes	Yes	Yes
GPU Support	No	No	Yes	Yes	Yes
Feature Importance	Available	Available	Available	Available	Available

Reference:

https://www.geeksforgeeks.org/machine-learning/gradientboosting-vs-adaboost-vs-xgboost-vs-catboost-vs-lightgbm/
https://www.analyticsvidhya.com/blog/2020/02/4-boosting-algorithms-machine-learning/
https://www.linkedin.com/advice/0/how-do-you-compare-contrast-different-boosting

ML | Boosting vs Bagging

Bagging is another ensemble learning technique used to improve accuracy. Unlike bagging where models are trained independently on different machine learning models, Boosting models are trained sequentially with each model correcting the errors of its predecessor

Feature	Boosting	Bagging
Combination Type	Combines predictions of different weak models	Combines predictions of the same type of model
Goal	Reduces **bias**	Reduces **variance**
Model Dependency	New models depend on previous models' errors	All the models have the same weightage
Weighting	Models are weighted based on performance	All models have equal weight.
Training Data Sampling	Each new model focuses more on the misclassified examples	Each model is trained on random subset of data
Error handling	Focuses on correcting errors made by previous models	Averages out errors from multiple models
Parallelism	Models are built sequentially less parallelizable	Models can be built in parallel
Overfitting	Less prone to overfitting with proper regularization	Can be prone to overfitting with complex base models
Model Complexity	Typically uses simpler models (like decision stumps)	Can use complex models (like full decision trees)
Example	AdaBoost, Gradient Boosting, XGBoost, LightGBM	Random Forest, Bagged Decision Trees

Reference:
https://www.geeksforgeeks.org/machine-learning/What-is-Bagging-classifier/

Exercises

See code here: https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/tree/main/Week5

You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. "DEBUGGING IS BEST STRATEGY TO LEARN A LANAGUAGE." So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.
Thanks.
Shahzad – Your AI – ML Instructor

Thank you - for listening and participating

- ❑**Questions / Queries**
- ❑**Suggestions/Recommendation**
- ❑**Ideas…..?**

Shahzad Sarwar
Cognitive Convergence
https://cognitiveconvergence.com
shahzad@cognitiveconvergence.com
voice: +1 4242530744 (USA) +92-3004762901 (Pak)