Advanced Machine Learning and Deep Learning

Dr. Mohammad Pourhomayoun
Assistant Professor
Computer Science Department
California State University, Los Angeles

Ensemble Learning:
Bagging, Boosting, and Random Forests

Ensemble Learning

- **Ensemble Learning** is a popular and effective approach to improve the accuracy and performance of a machine learning problem.
- **Ensemble Learning** uses a **group of machine learning algorithms** (called **base** learners), and then combine the results of them to achieve higher accuracy.

Ensemble Learning

- **Example**: Construct a Strong Classifier by combining several Weak Classifiers!

- Each learner (e.g. classifier) alone may have very poor performance. But, a group of them together can achieve very accurate results.

- For the sake of simplicity (or other reasons), each classifier may make some assumptions, which might be or not be valid for the problem!

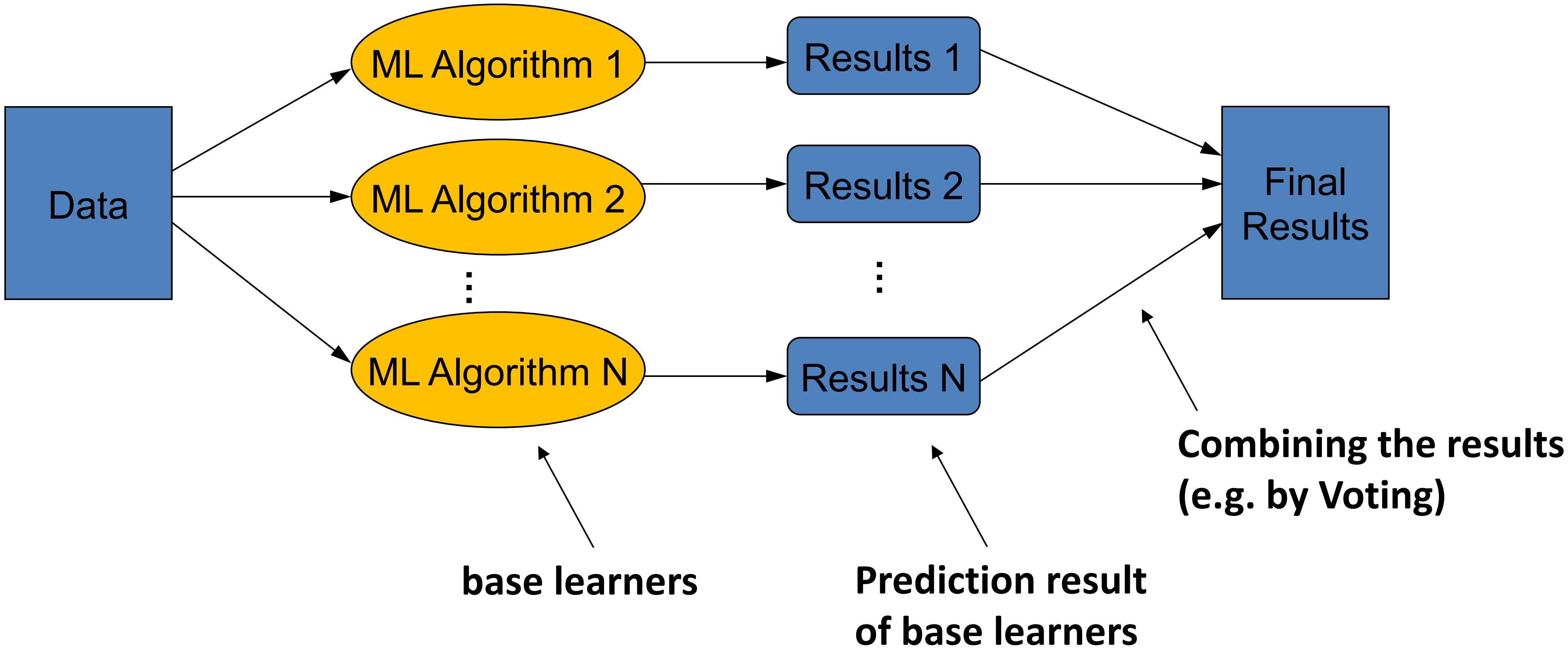
Different types of Ensemble Learning

- **Different learners of the Ensemble Learning may use:**

1. **Different learning Algorithms**
   - E.g. Combination of decision tree, KNN, and logistic regression
2. **Different choice of learning Parameters**
   - E.g. Several KNNs with various K's
3. **Different Features**
   - E.g. Several decision trees, each for a set of features
4. **Different Data Subsets**
   - E.g. Several decision trees, each for a section of the dataset
5. **Different Subproblems**
   - E.g. Several logistic regression classifiers, each for a part of the problem

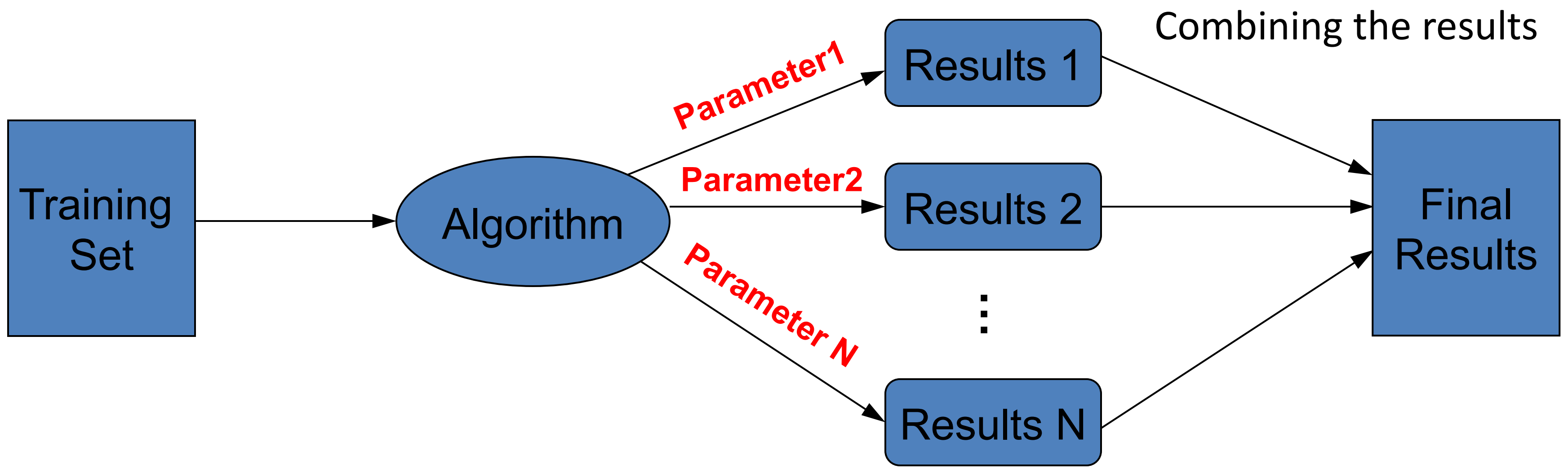
Different types of Ensemble Learning

1. **Different learning Algorithms:**



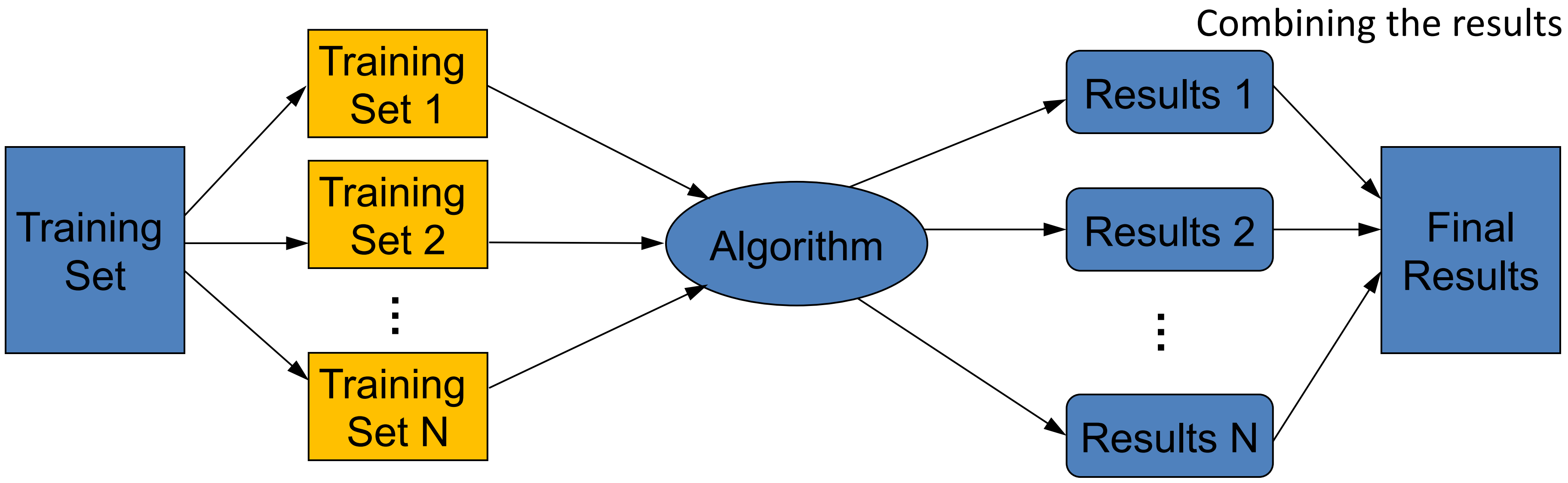
Different types of Ensemble Learning

2. **Different choice of learning Parameters:**



Different types of Ensemble Learning

3. **Different Features**
4. **Different Subsets**



An Important Note about Ensemble Learning

- The key in designing ensembles is **diversity** and **not necessarily high accuracy** of the base classifiers.

- Members of an ensemble group **<u>should vary in the examples they misclassify, so that they cover each other's mistakes!</u>**

- In other word, if we have several classifiers that are pretty accurate but they all misclassify the **same samples**, then ensemble learning will not achieve any better results! Therefore, most ensemble approaches, seek to promote diversity among the models they combine.

Combining the Results of Base Learners

- There are 3 main approaches for **combining** the results in Ensemble Learning:
  1. **Voting**
  2. **Stacking**
  3. **Cascading**

1- Combining the Results: Voting

- **Voting:**
  - Classifiers are combined in a **static** way.
  - Each base-level classifier gives a vote for its prediction.
  - **Plurality vote**: The final decision for each data sample (each prediction) is made based on the majority of votes.
    - E.g: Suppose we use 9 different decision tree classifiers for weather forecasting. 5 of them predict Rain for tomorrow, and 4 of them predict sunny. Thus, the final decision will be Rainy!
  - **Note**: Depending on the problem, some votes can be **weighted**. In this approach, the better base classifiers get higher voting weight.

2- Combining the Results: Stacking

- **Stacking:**
  - Classifiers are combined in a data-driven **dynamic** way.
  - An **upper level** machine learning method is used to learn how to combine the prediction results of the **base-level** classifiers.
  - The **upper level** classifier is used to make final decision from the predictions of the **base-level** classifiers.

3- Combining the Results: Cascading

- **Cascading:**
  - Classifiers are combined in an **iterative** way.
  - At each iteration, the training dataset is extended or modified based on the prediction results obtained in the previous iterations.
  - We will talk more about it later!

Question: Why Does Ensemble Learning work?

- Suppose we have 3 completely Independent Classifiers, each one with prediction accuracy of 70%, and we want to use Voting Method for making a prediction:
  - For a positive sample, the final prediction is positive if **at least 2 out of 3** classifiers vote for positive:

$$0.7^3 + 0.7\times0.7\times0.3 + 0.7\times0.3\times0.7 + 0.3\times0.7\times0.7 \ = 0.78$$

  - Thus, combining 3 classifiers with accuracy of 70% each, using just a simple voting method can improve the accuracy to 78%.
  - **In theory**, If we use 101 independent classifiers, then the final voting accuracy will be 99.9%!!!!
  - But, can we always achieve this accuracy **in practice**? Why not?

Intuitions: Why Does Ensemble Learning work?

- **Note:** Making decision based on Independent Binary Classifiers, using Voting Method has Binomial Probability Distribution:
- **Binomial Distribution:** The probability that ***x*** out of ***n*** independent classifiers vote correctly, where each classifier predicts correctly with probability of ***p***, is

$$P(X = x|p, n) = \frac{n!}{r!(n-x)!} p^x (1-p)^{n-x}$$

  - In theory, If we use 101 independent classifiers, then the final voting accuracy will be 99.9%.
  - In practice, the accuracy is usually lower than theory because the classifiers are **NOT** completely independent!

Advantages and Disadvantages of Ensemble Learning

- **Advantages of Ensemble Learning:**
  - Improve prediction performance and accuracy
  - Robust to Overfitting

- **Disadvantages of Ensemble Learning:**
  - The combined classifier is not so transparent
  - Not a compact representation

Three Popular Approaches for Ensemble Learning

- **Bagging**: Bagging (stands for **B**ootstrap **Agg**regating) was first proposed by ***Leo Breiman*** to improve the classifier results using a combination of several classifiers trained on randomly generated training sets.

- **Boosting:** Originally proposed by ***Robert Schapire*** to build a strong classifier using a set of extremely weak base classifiers (each one with accuracy of slightly better than random guess).

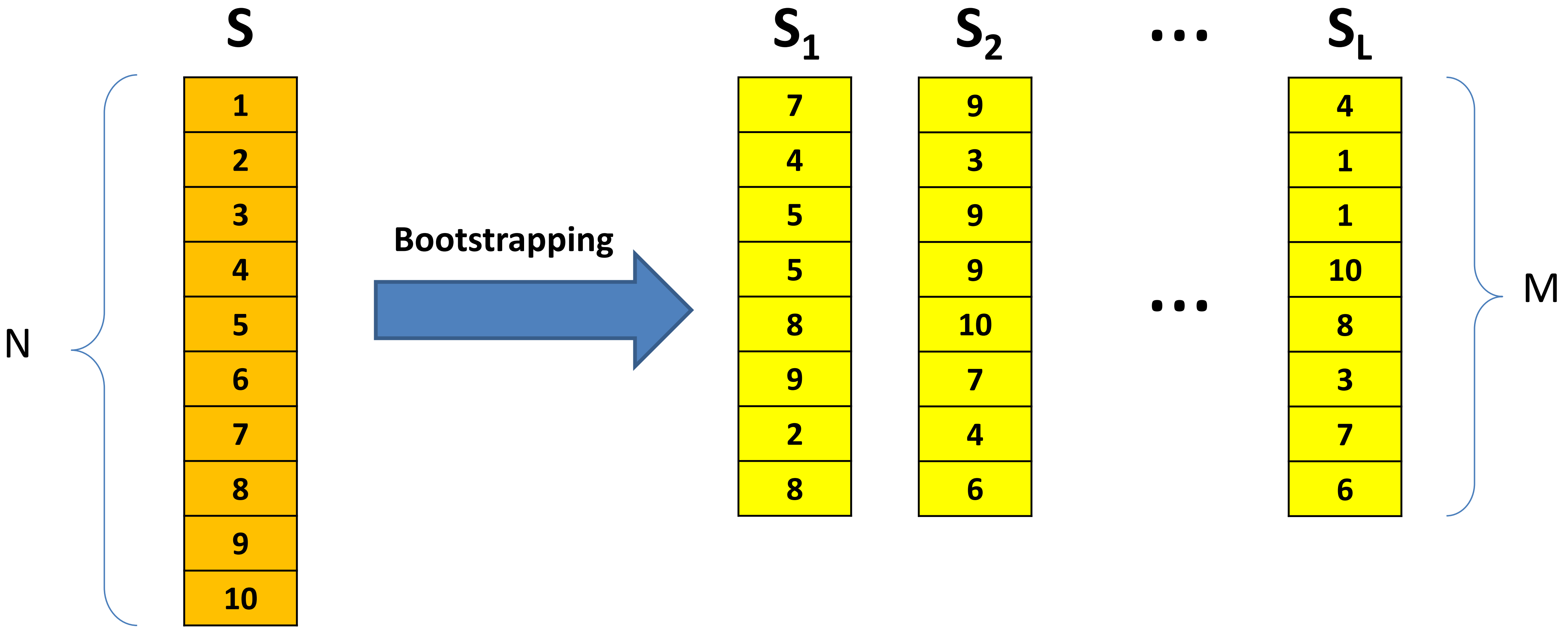
- **Random SubSpace (Random Forest):** First proposed by ***Leo Breiman*** to improve the accuracy of decision tree classifiers and address the overfitting problem.

Bagging
(Bootstrap Aggregation)

Bagging (Bootstrap Aggregation)

- Here are the main 4 steps for Bagging method:
- **Step1: Bootstrapping**: Suppose we have a **Training Dataset S** of **size N**. Bootstrapping generates **L** new training sets $\mathbf{S_1, S_2, \ldots, S_L}$ each of **size M**, by sampling from the original dataset **S** randomly and **with replacement.**
  - This type of sampling is called **Bootstrapping** or **Bootstrap Sampling.**
  - The bootstrap training sets $\mathbf{S_1, S_2, \ldots, S_L}$ may have overlap with each other.
  - By sampling with **replacement**, some data sample may be repeated in **each** $\mathbf{S_i}$.

Example for Bootstrap Sampling



Bagging (Bootstrap Aggregation)

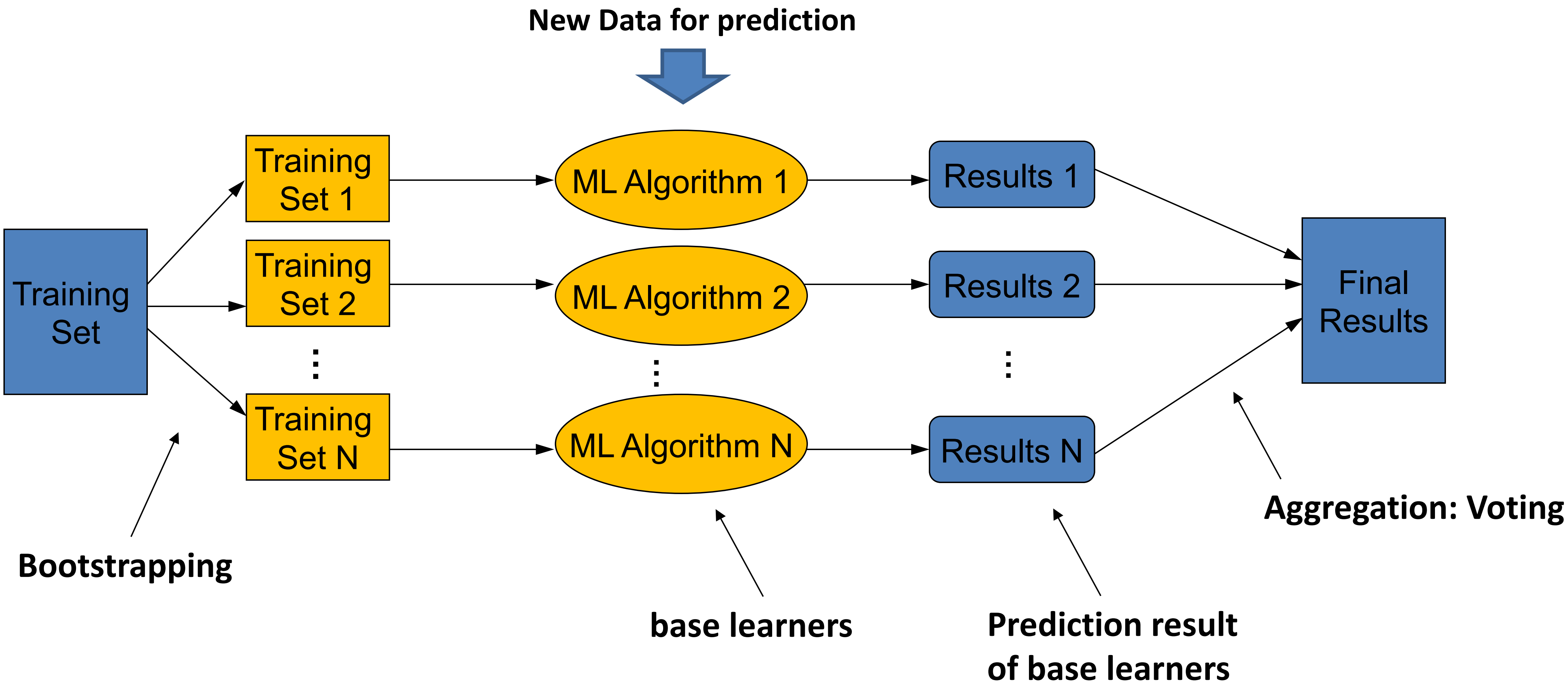
- **Step2: Training stage**: The **L** new training sets $\mathbf{S_1, S_2, \ldots, S_L}$ will be used to train **L** learners (can be either **L** classifiers $\mathbf{C_1, C_2, \ldots, C_L}$ or **L** regression models $\mathbf{R_1, R_2, \ldots, R_L}$).
    - E.g. Training L decision trees for classification, or L linear regression models for regression.

- **Step3: Testing Stage:** Given a new unknown data sample, The **L** trained models will be used to make prediction for the new sample. In other word, Each classifier $\mathbf{C_i}$ or regressor $\mathbf{R_i}$ returns its prediction.

Bagging (Bootstrap Aggregation)

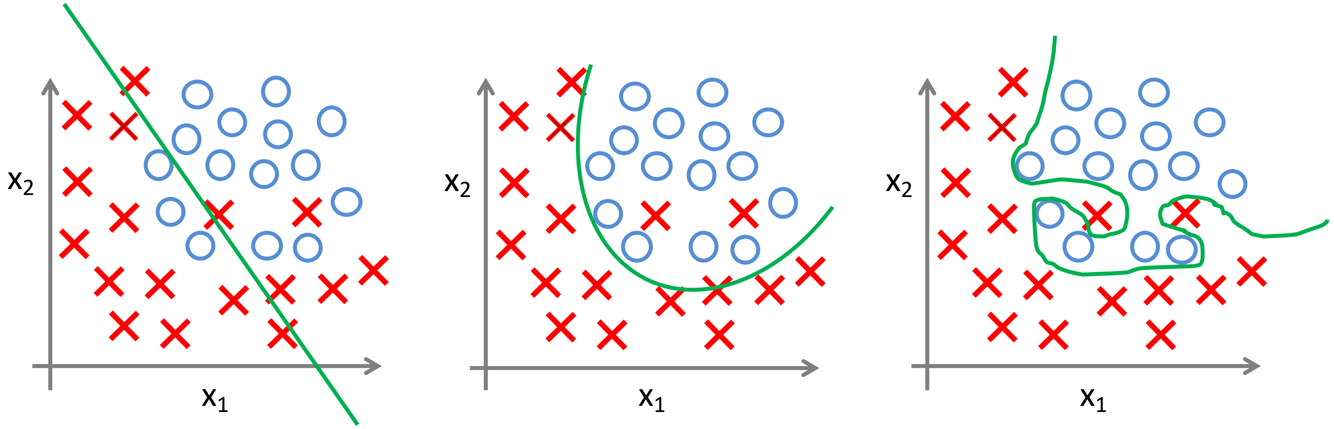
- **Step4: Combining the results:**
  - For Classification, we use **Voting** method. The final prediction is based on the majority vote of the **L** classifiers.
  - For Regression, we use **Averaging** method. The final prediction is the average of L predictions.
  - For Regression, depending on the application, sometimes we may prefer to use **Median** rather than average to get rid of outliers.

Bagging (Bootstrap Aggregation)



Important Notes

- For any machine learning method, there are two main sources of error:
  - **Bias**: Expected error due to inaccurate model in the learning algorithm that may cause to miss the relations between features and outputs (**underfit model**).
  - **Variance**: Expected error due to particular training sets, and high sensitivity of the system to small fluctuations in the training set (**overfit models**).



Under-fit (High Bias) | **Ideal fit** | **Over-fit (High Variance)**

Important Notes

- For any machine learning method, there are two main sources of error:
  - **Bias**: Expected error due to inaccurate model in the learning algorithm that may cause to miss the relations between features and outputs (**underfit model**).
  - **Variance**: Expected error due to particular training sets, and high sensitivity of the system to small fluctuations in the training set (**overfit models**).

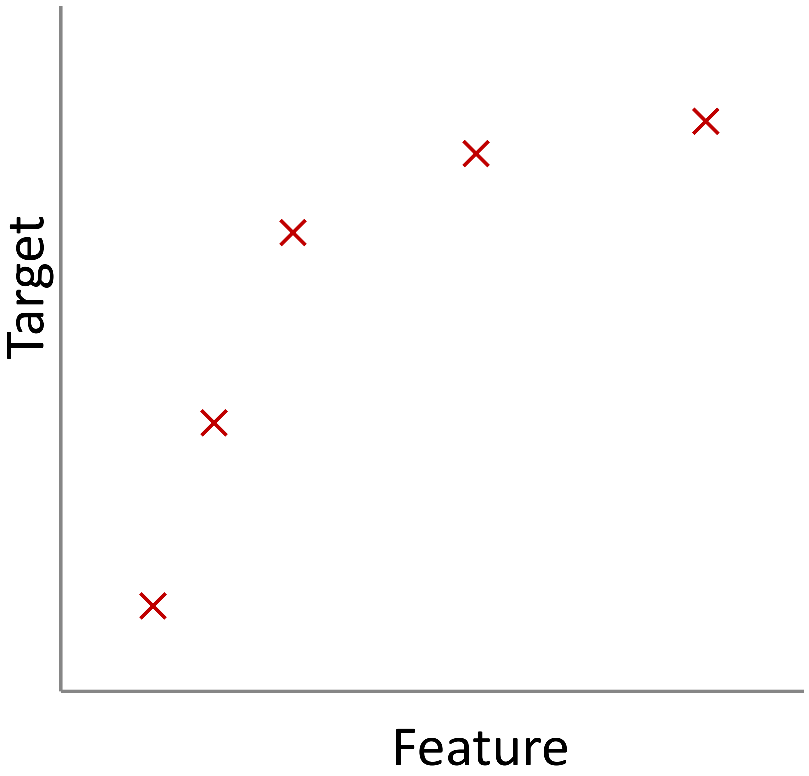
- Bagging works because it **reduces variance**. In other word, we don't suffer from random errors made by a single classifier. Thus, it is a good approach **to deal with overfitting**.

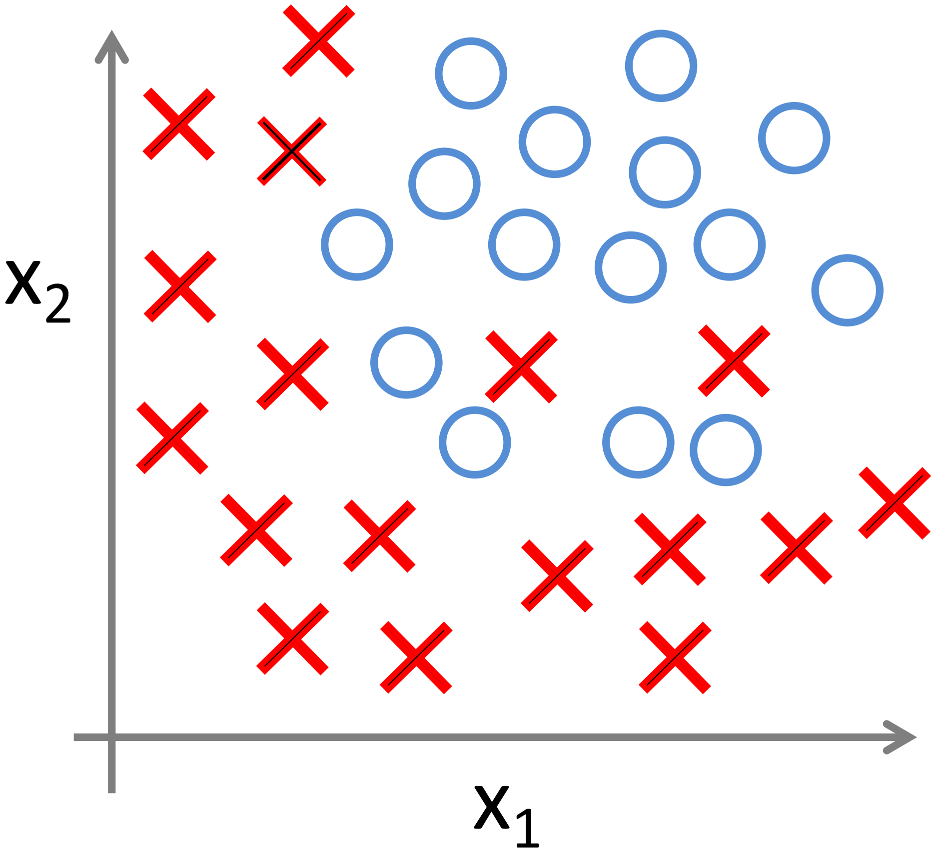
Important Notes

- Bagging has the best performance when the Learning Algorithm is **unstable (high variance)**: if small changes to the training set cause large changes in the prediction results.
  – Some candidates for Bagging: Decision Tree and Neural Networks.

- In some rare cases, when the learning algorithm is **very stable** (low variance), Bagging may degrade the accuracy. But, it is easy to find out and avoid it.

How Can Bagging Resolve Overfitting?





Thank You!

Questions?