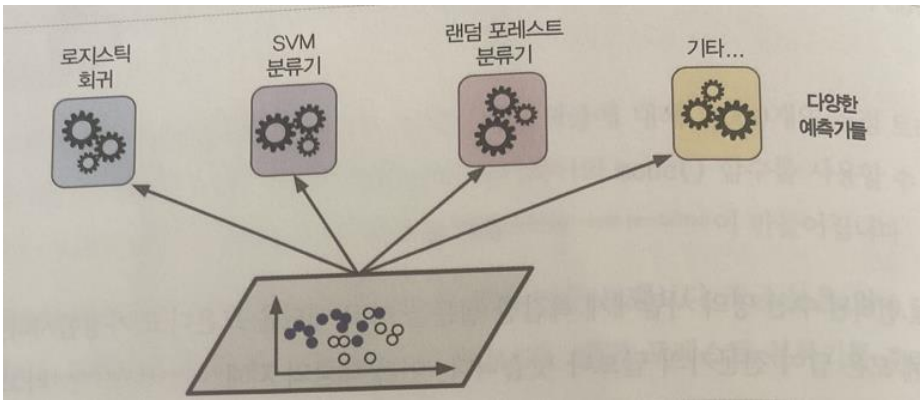
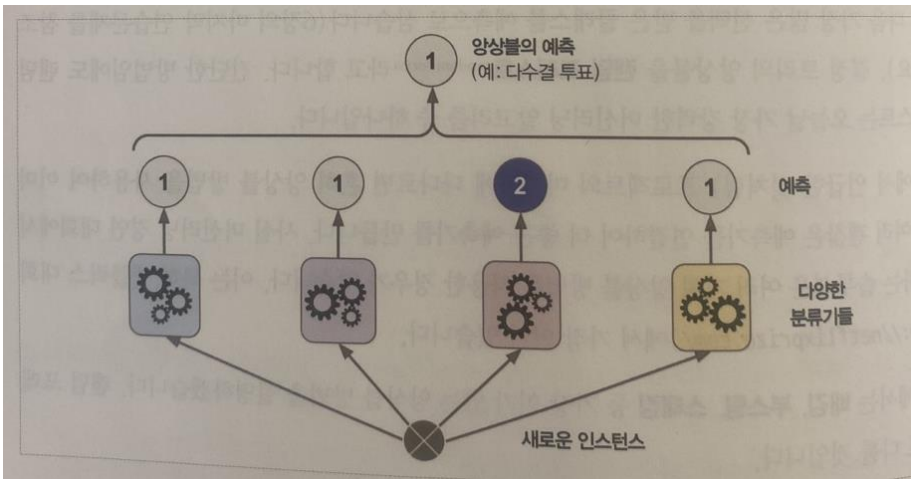
1. **대중의 지혜:** 무작위로 선택된 수천 명의 사람에게 복잡한 질문을 하고 대답을 모은다고 가정하면 많은 경우 이렇게 모은 답이 전문가의 답보다 낫다
2. **앙상블 학습:** 여러 개별 학습 모델들을 결합 (ensemble)하여 더 정확하고 안정적인 예측을 수행하는 방법으로 각 개별 모델은 같은 알고리즘을 사용할 수도 있고 다른 알고리즘을 사용할 수도 있음.
3. **Random forest:** 가장 강력한 머신 러닝 알고리즘 중 하나로 결정 트리의 앙상블 모델
4. **투표 기반 분류기:** 더 좋은 분류기를 만드는 매우 간단한 방법 중의 하나는 각 분류기의 예측을 모아서 가장 많이 선택된 class를 예측하는 것. 이렇게 다수결로 정해지는 분류기를 직접 투표 (hard voting) 분류기라고 한다.

   예: 1) 아래와 같이 여러 종류의 분류기를 훈련 시킴



   2) 각 분류기의 예측을 모아서 가장 많이 선택된 class로 예측



3) [sklearn.ensemble.VotingClassifier — scikit-learn 1.3.0 documentation]

5. **실습 (투표기반 분류기):** 투표기반 분류기가 다른 개별 분류기보다 성능이 더 좋음

```
LogisticRegression 0.864
RandomForestClassifier 0.896
SVC 0.896
VotingClassifier 0.912
```

6. **Bagging (배깅):**

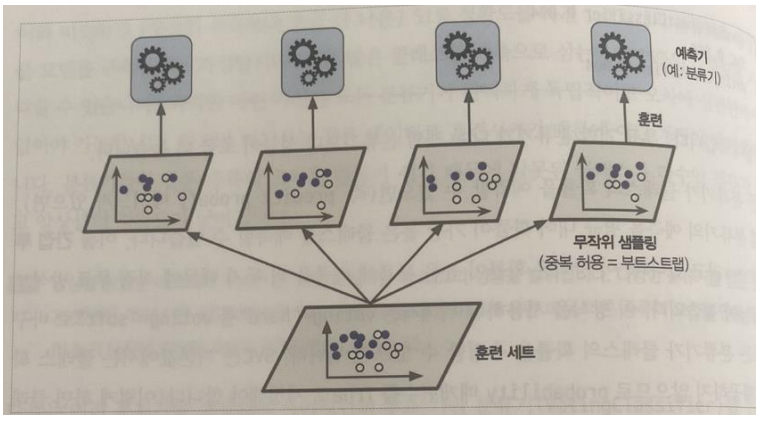
1) 훈련시에 같은 알고리즘을 사용하고 훈련 세트에서 subset을 무작위로 구성하여 분류기를 각기 다르게 학습시키는 것을 bagging과 pasting이라고 한다.

2) 훈련 세트에서 중복을 허용하여 sampling하는 방식을 bagging이라하며, 중복을 허용하지 않고 sampling하는 방식을 pasting이라 한다.

3) bagging의 장점: Bagging은 중복을 허용한 랜덤 샘플링을 통해 각기 다른 데이터 셋을 만들어 기본 분류기들을 학습시킵니다. 이로 인해 각 분류기가 다양한 데이터에 대해 학습하게 되어 일반화 능력이 향상됩니다. 특정 데이터에 대해 과적합되는 경향을 줄여준다. 즉, overfitting 줄여줌

7. Bagging과 pasting의 훈련 과정:

1) 훈련 세트에서 무작위로 sampling하여 여러 개의 예측기를 훈련시킴



2) 모든 예측기 (분류기)가 훈련을 마치면 모든 예측기의 예측을 모아서 새로운 샘플에 대한 예측을 만든다. 분류기일때는 통계적 최빈값 (가장 많은 예측 결과)을 사용

8. **(실습) Scikit-learn에서의 bagging과 pasting 사용 예 (총 500개의 sample)**

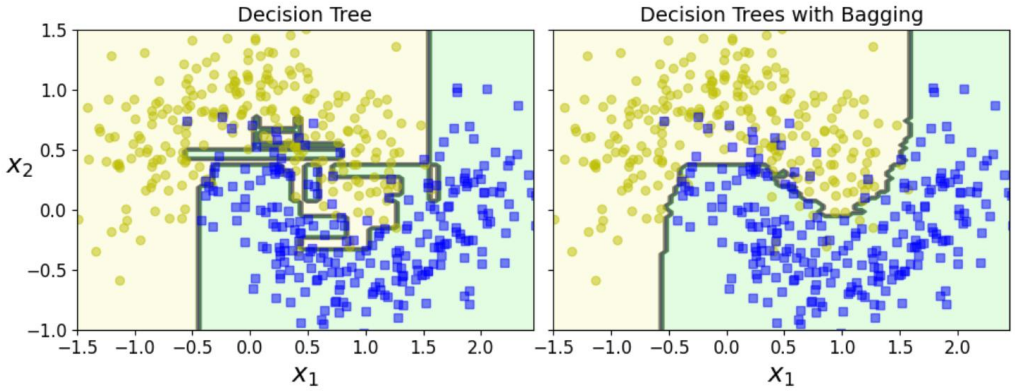
1) BaggingClassifier 사용

2) 각 분류기는 훈련 세트에서 중복을 허용하여 무작위로 선택된 100개의 sample로 훈련

3) 500개의 결정 트리 분류기를 앙상블로 훈련

4) 500개의 결정 트리 분류기의 결과를 모아서 가장 많은 예측 결과 사용

- 왼쪽: decision tree만 사용. 결정 경계가 불규칙함
- 오른쪽: bagging을 사용. 결정 경계를 보면 일반화가 훨씬 더 잘 됨



5. (실습) Random forest 분류기:

- **결정 트리의 앙상블 버전. 일반적으로 결정 트리의 bagging 혹은 pasting을 적용한 방법**
- **랜덤성을 활용하여 트리의 다양성 확보, overfitting 방지, 여러 결정 트리의 예측을 결합하여 강력한 분류 성능 보임**

아래 3단계로 동작

1) Data sampling: bagging이나 pasting을 사용하여 data를 sampling해서 결정 트리 구성

2) 랜덤 특징 선택: node 분할시에 전체 특징 중에서 best 특징을 찾는 대신 무작위로 선택한 특징 후보들 중에서 best 특징을 찾는 방법으로 무작위성을 더 늘림

   Why? 무작위성을 주입하여 트리를 더 다양하게 만듦

3) 예측: 새로운 데이터가 주어지면 각 트리별로 개별적으로 예측하고 다수결 투표로 최종 예측. 회귀문제에서는 평균으로 예측

6. Random forest 분류기의 동작 원리

1) 데이터 준비

2) 각각의 트리가 독립적으로 학습 및 예측되어, 최종 결과는 모든 트리의 예측을 종합하여 결정. 분류 문제의 경우 "다수결 투표"로 최종 class 결정

3) 트리 생성 과정

   1. 원본 데이터에서 중복을 허용하여 무작위로 데이터를 추출하여 학습 데이터 만듬
   2. 각 노드에서 분할 기준을 선택할 때, 전체 특징 중 무작위로 선별된 일부 특징만 고려. 구체적으로 분류 문제의 경우 특징 수가 d면 $\sqrt{d}$만큼 사용
   3. 결정 트리 학습

4) 새로운 데이터가 들어오면, 각 트리는 독립적으로 이 데이터를 분류

5) 트리들의 결과를 종합하여 최종 예측. 분류 문제의 경우 다수결 투표

7. (실습) Random forest에서의 특징 중요도

1) Random forest의 장점은 특징의 상대적 중요도를 측정하기 쉽다

2) Scikit-learn에서는 어떤 특징을 사용한 node가 평균적으로 Gini 불순도를 얼마나 감소시키는지 확인하여 특징의 중요도를 측정함. (feature_importance_ 변수)

3) Scikit-learn에서는 훈련이 끝난 뒤 특징마다 자동으로 이 점수를 계산하고 중요도의 전체 합이 1이되도록 결과값을 정규화한다.

```
sepal length (cm) 0.11249225099876375
sepal width (cm) 0.023119288282510 33
petal length (cm) 0.4410304643639577
petal width (cm) 0.4233579963547682
```

4) Iris dataset에 대해서 random forest로 학습시킨 후에 가장 중요한 특징을 찾음

Petal length (44%), Petal width (42%)

5) The following figure overlays the decision boundaries of 15 decision trees. As you can see, even though each decision tree is imperfect, the ensemble defines a pretty good decision

boundary: