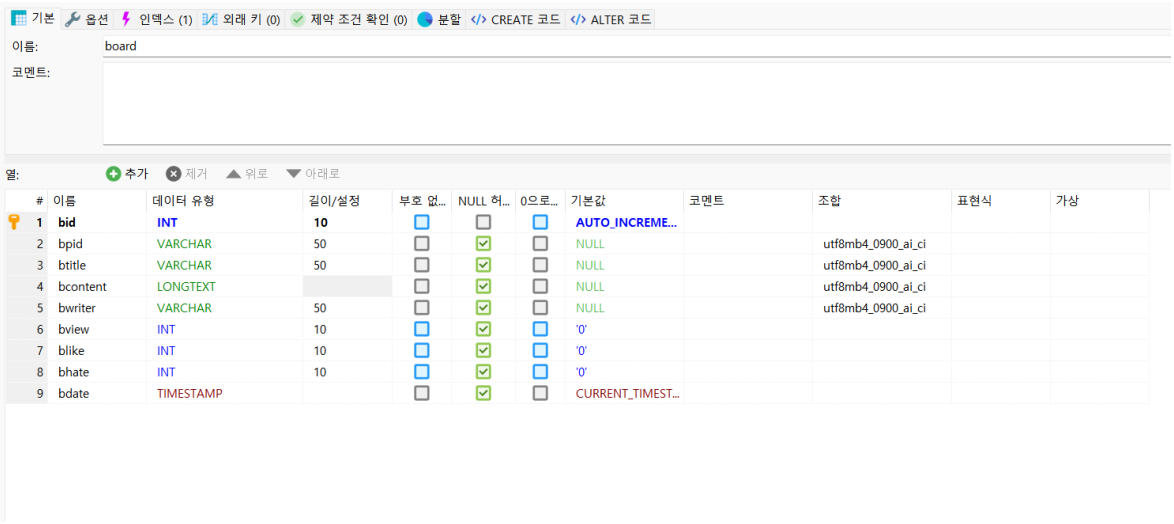
Spring Day 16

▼ Spring

Spring Board

DB 생성



#	이름	데이터 유형	길이/설정	부호 없...	NULL 허...	0으로...	기본값	코멘트	조합	표현식	가상
1	bid	INT	10	☐	☐	☐	AUTO_INCREME...				
2	bpid	VARCHAR	50	☐	☑	☐	NULL		utf8mb4_0900_ai_ci		
3	btitle	VARCHAR	50	☐	☑	☐	NULL		utf8mb4_0900_ai_ci		
4	bcontent	LONGTEXT		☐	☑	☐	NULL		utf8mb4_0900_ai_ci		
5	bwriter	VARCHAR	50	☐	☑	☐	NULL		utf8mb4_0900_ai_ci		
6	bview	INT	10	☐	☑	☐	'0'				
7	blike	INT	10	☐	☑	☐	'0'				
8	bhate	INT	10	☐	☑	☐	'0'				
9	bdate	TIMESTAMP		☐	☑	☐	CURRENT_TIMEST...				

```
CREATE TABLE `board` (
  `bid` INT(10) NOT NULL AUTO_INCREMENT,
  `bpid` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `btitle` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `bcontent` LONGTEXT NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `bwriter` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `bview` INT(10) NULL DEFAULT '0',
  `blike` INT(10) NULL DEFAULT '0',
  `bhate` INT(10) NULL DEFAULT '0',
  `bdate` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`bid`) USING BTREE
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
;
```

Board.java(DTO)

```
package com.carshop.board;

public class Board {

  private int bid, bview, blike, bhate;

  private String bpid, btitle, bcontent, bwriter, bdate;

  public int getBid() {
    return bid;
  }

  public void setBid(int bid) {
```

```
  this.bid = bid;
}

public int getBview() {
  return bview;
}

public void setBview(int bview) {
  this.bview = bview;
}

public int getBlike() {
  return blike;
}

public void setBlike(int blike) {
  this.blike = blike;
}

public int getBhate() {
  return bhate;
}

public void setBhate(int bhate) {
  this.bhate = bhate;
}

public String getBpid() {
  return bpid;
}

public void setBpid(String bpid) {
  this.bpid = bpid;
}

public String getBtitle() {
  return btitle;
}

public void setBtitle(String btitle) {
  this.btitle = btitle;
}

public String getBcontent() {
  return bcontent;
}

public void setBcontent(String bcontent) {
  this.bcontent = bcontent;
}

public String getBwriter() {
  return bwriter;
}

public void setBwriter(String bwriter) {
  this.bwriter = bwriter;
}

public String getBdate() {
  return bdate;
}

public void setBdate(String bdate) {
  this.bdate = bdate;
}



public Board() {

}

public Board(int bid, int bview, int blike, int bhate, String bpid, String btitle, String bcontent, String bwriter,
    String bdate) {

  this.bid = bid;
  this.bview = bview;
  this.blike = blike;
  this.bhate = bhate;
  this.bpid = bpid;
  this.btitle = btitle;
  this.bcontent = bcontent;
  this.bwriter = bwriter;
  this.bdate = bdate;
}
```

```
}
```

▼ CRUD

board_SQL.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE  mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="board">

  <insert id="insert"
    parameterType="com.carshop.board.Board" useGeneratedKeys="true" keyProperty="bid">
    <![CDATA[
      INSERT INTO board
      (btitle, bcontent, bwriter) VALUES (#{btitle}, #{bcontent}, #{bwriter})
    ]]>
  </insert>
```

BoardRepository.java

```
package com.carshop.board;

public interface BoardRepository {

  void setNewBoard(Board board);



}
```

BoardRepositoryImpl.java

```
package com.carshop.board;

import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

@Repository
public class BoardRepositoryImpl implements BoardRepository {


  @Autowired
  SqlSessionTemplate sqlSessionTemplate;

  @Override
  public void setNewBoard(Board board) {
    this.sqlSessionTemplate.insert("board.insert", board);
  }

}
```

BoardService.java

```
package com.carshop.board;

public interface BoardService {
```

```
  void setNewBoard(Board board);

}
```

BoardServiceImpl.java

```
package com.carshop.board;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class BoardServiceImpl implements BoardService{

  @Autowired    // DI 의존성 주입 IoC 제어의 역전
  private BoardRepository boardRepository;

  @Override
  public void setNewBoard(Board board) {
    boardRepository.setNewBoard(board);
  }

}
```

BoardController.java

```
package com.carshop.board;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/boards")
public class BoardController {

  @Autowired    // DI
  private BoardService boardService;

  @GetMapping("/addBoard")
  public String requestAddBoardForm(@ModelAttribute("NewBoard") Board board) {
    return "boards/addBoard";
  }

  @PostMapping("/addBoard")
  public String submitAddBoardForm(@ModelAttribute("NewBoard") Board board) {
    boardService.setNewBoard(board);
    return "redirect:/boards/list";
  }


}
```

addBoard.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>

<head>
  <title>게시판 등록</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">

</head>
<body class="text-center">
<div class="alert alert-dark" role="alert">
<div class="container"><h1>게시판 등록</h1>
```

```
      <form:form modelAttribute="NewBoard" action="./addBoard?${_csrf.parameterName}=${_csrf.token}" class="form-horizontal" en
        <fieldset>
        <!-- Spring Security 에서 username 불러오기 -->
        <sec:authentication property="principal" var="user"/>
          <form:input path="bwriter" type="hidden" value="${user.username}" class="form-control" />
          제목 :
          <form:input path="btitle" class="form-control" />
          내용 :
          <form:textarea path="bcontent" class="form-control" rows="10"/>

        <input type="submit" class="btn btn-primary" value="등록">


        </fieldset>
      </form:form>

    </div>
</div>

</body>
</html>
```

▼ C R U D

board_SQL.xml

```
<select id="select_list" resultType="com.carshop.board.Board">
<![CDATA[
  SELECT * FROM board ORDER BY bid DESC
  ]]>
</select>

<select id="select_detail" parameterType="String" resultType="com.carshop.board.Board">
<![CDATA[
  SELECT * FROM board WHERE bid = #{bid}
  ]]>
</select>
```

BoardRepository.java

```
List<Board> getAllBoardList();

Board getBoardById(String bId);
```

BoardRepositoryImpl.java

```
@Override
  public List<Board> getAllBoardList() {
    return this.sqlSessionTemplate.selectList("board.select_list");
  }

@Override
  public Board getBoardById(String bId) {
    return this.sqlSessionTemplate.selectOne("board.select_detail", bId);
  }
```

BoardService.java

```
List<Board> getAllBoardList();

Board getBoardById(String bid);
```

BoardServiceImpl.java

```
@Override
  public List<Board> getAllBoardList() {
    return boardRepository.getAllBoardList();
  }

@Override
  public Board getBoardById(String bid) {

    Board boardById = boardRepository.getBoardById(bid);

    return boardById;
  }
```

BoardController.java

```
@RequestMapping("/list")
  public String BoardList(Model model) {
    List<Board> list = boardService.getAllBoardList();
    model.addAttribute("boardList", list);
    return "boards/list";
  }

@GetMapping("/board")
  public String requestCarById(@RequestParam("id") String bid, Model model) {
    Board boardById = boardService.getBoardById(bid);
    model.addAttribute("board", boardById);
    return "boards/board";
  }
```

list.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
  pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>


<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>게시판</title>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" cross



</head>
<body>



  <div class="my-5">
    <div class="alert alert-dark">
      <div class="container">
        <h1>게시판</h1>
      </div>
    </div>
  </div>

  <div class="container">
    <div class="d-grid gap-2 d-md-flex justify-content-md-end">
    </div>
    <div style="padding-top: 50px">
      <table class="table table-hover">
        <tr>
          <th>번호</th>
          <th>제목</th>
          <th>작성자</th>
          <th>조회수</th>
```

```
          <th>좋아요</th>
          <th>싫어요</th>
          <th>작성일</th>
          <th></th>
        </tr>
        <form:form name="removeForm" method="put">
          <c:forEach items="${boardList}" var="board">
            <tr>
              <td>${board.bid}</td>
              <td><a href="/boards/board?id=${board.bid}">${board.btitle}</a></td>
              <td>${board.bwriter}</td>
              <td>${board.bview}</td>
              <td>${board.blike}</td>
              <td>${board.bhate}</td>
              <td>${board.bdate}</td>
<!--              <td> -->
<%--              <p><a href="javascript:removeCar('${car.cid}')" --%>
<!--                class="btn btn-danger btn-sm">삭제</a>  -->
<%--              <a href="/cars/update?cid=${car.cid}" --%>
<!--                class="btn btn-primary btn-sm">수정</a></td> -->
            </tr>
          </c:forEach>
        </form:form>
        <tr>
          <th></th>
          <th></th>
          <th></th>
        </tr>
      </table>

      <a href="<c:url value="/boards/addBoard" />" class="btn btn-primary">
        글쓰기</a>
    </div>
    <hr>

  </div>
</body>


<script>


  function removeCar(car) {
    $.ajax({
      type:"POST",
      url:"/cars/removeProduct",
      data:{carId: car},
      beforeSend : function(xhr)
          {
              xhr.setRequestHeader("${_csrf.headerName}", "${_csrf.token}");
          },
      success:function(result) {
        alert("제품이 삭제되었습니다.")
      },
      error:function(request, status, error) {
        alert(request.status + " " + request.responseText);
      }

    })
    window.location.reload();
  }

  </script>
</html>
```

board.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" cross

<head>
  <title>글 상세보기</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">

</head>
<body class="container">
<!-- 글 -->
<div class="card">
```

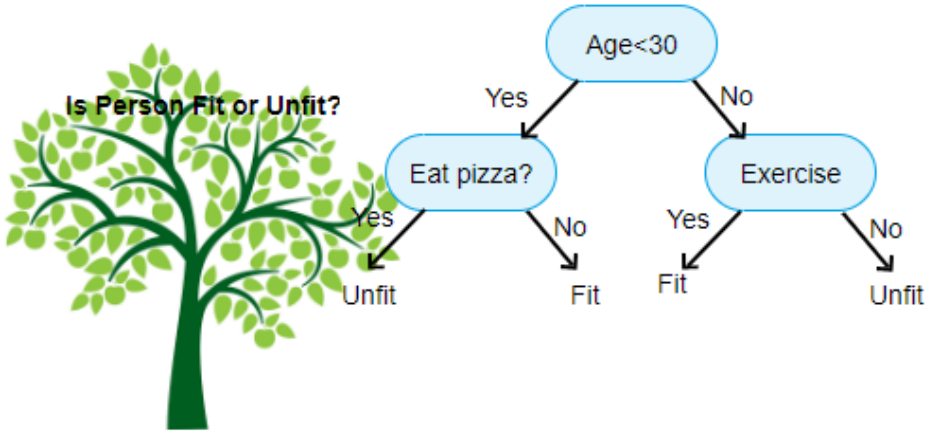
```
  <h5 class="card-header">${board.btitle}</h5>
  <div class="card-body">
    <h5 class="card-title">${board.bcontent}</h5>
    <br><br><br>
    <div class="d-flex justify-content-end">
    <p class="btn btn-success btn-sm">조회수 [${board.bview}]</p>
    <p class="btn btn-primary btn-sm">좋아요 [${board.blike}]</p>
    <p class="btn btn-danger btn-sm">싫어요 [${board.bhate}]</p>
    </div>
    <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}</div>
  </div>
</div>

<!-- 댓글 -->
<div class="card">
  <div class="card-body">
  댓글 리스트
  <br><br>
  <!-- 댓글 등록 -->
  <sec:authentication property="principal" var="user"/>
<input id="bwriter" type="hidden" value="${user.username}" />
<input id="bid" type="hidden" name="bid" value="${board.bid}" >
<textarea name="bcontent" id="bcontent" rows="5" class="form-control"></textarea>
<input type="button" class="btn btn-primary my-2" onclick="replyNewFunction()" value="댓글 등록">

  </div>
</div>




<div class="my-5">
</div>

</body>

<iframe name="cart" style="display: none;"></iframe>

<script type="text/javascript">

function addCartFunction() {
  $.ajax({
    type:"POST",
    url:"/cart/ajaxAdd",
    data:{cid:"${car.cid}"},
    beforeSend : function(xhr)
        {   /*데이터를 전송하기 전에 헤더에 csrf값을 설정한다*/
            xhr.setRequestHeader("${_csrf.headerName}", "${_csrf.token}");
        },
    success:function(result) {
      alert("제품이 장바구니에 추가되었습니다.")
    },
    error:function(request, status, error) {
      alert(request.status + " " + request.responseText);
    }

  })
}

</script>



</html>
```

▼ Python

Machine Learning

DT(결정 나무)

```
# DT(결정 나무)

# Decision Tree
# 이진 트리 알고리즘을 사용한다. 트리의 각 분기점에 데이터셋의 피처를 위치시킨다.
# 각 분기점에서 해당 피처에 관한 임의의 조건을 가지고 계속하여 2개의 줄기로 나뉘어지면서 데이터를 구분한다.
# (★중요) 결정 나무는 각 분기점에서 가장 분류가 잘되는 최적의 기준을 찾는다.


# Depth(깊이)가 깊을 수록 결과가 좋을 것만은 아니다.
# 오히려 Tree의 깊이가 너무 깊으면 모델이 훈련데이터에 과대적합되어
# 새로운 데이터 값을 주었을 때 예측도가 떨어질 수 있다.
```




```
import pandas as pd
import numpy as np
from sklearn import datasets

# 데이터 가져와서 데이터 프레임 만들기
iris = datasets.load_iris()
df = pd.DataFrame(iris['data'], columns=iris['feature_names'])
df.columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
df['Target'] = iris['target']


# 문제와 답인지 분리
X_data = df.loc[:,'sepal_length':'petal_width']
y_data = df.loc[:,'Target']


# 훈련용과 검증용으로 데이터 분리
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_data, y_data,
                                                    test_size=0.2,
                                                    # shuffle=True,
                                                    random_state=20)
```

```
# DT(결정 나무)

from sklearn.tree import DecisionTreeClassifier

# 객체 생성
DTC = DecisionTreeClassifier(max_depth=2, random_state=9)

# 학습
DTC.fit(X_train, y_train)

# 예측
y_knn_pred = DTC.predict(X_test)

# 평가
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_knn_pred)
```

```
# [max_depth=3]인 경우, 정확도 0.9333333333333333
# [max_depth=100]인 경우, 정확도 0.9333333333333333
# [max_depth=1]인 경우, 정확도 0.6333333333333333
# [max_depth=2]인 경우, 정확도 0.9
```

Ensemble(앙상블) - Voting

```
# Ensemble(앙상블) - Voting

# 앙상블 모델은 여러 모델을 결합하여 성능을 높이는 방법이다.
# 여러 모델의 예측 결과를 종합하여 모델의 예측 능력을 높인다.

# 예를 들면 [KNN, SVM, DT] 세 가지 알고리즘의 결과를 토대로 다수결로 최종 결과를 예측할 때
# 결과가 [KNN = 1], [SVM = 1], [DT = 2]라면 앙상블 모델은 1로 예측한다.

# 여러 알고리즘을 종합하여 최종 결과를 도출해내므로 예측 능력은 아주 우수한 편이나
# 개별 모델 학습에 비해 시간이 많이 걸린다는 단점이 있다.
```

```


# Ensemble(앙상블)을 위한 알고리즘 준비

import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

#데이터 가져와서 데이터 프레임 만들기
iris = datasets.load_iris()
df = pd.DataFrame(iris['data'], columns=iris['feature_names'])
df.columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
df['Target'] = iris['target']


# 문제와 답안지 분리
X_data = df.loc[:,'sepal_length':'petal_width']
y_data = df.loc[:,'Target']


# 훈련용과 검증용으로 데이터 분리
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_data, y_data,
                                                    test_size=0.2,
                                                    # shuffle=True,
                                                    random_state=20)



# 객체 생성
svc = SVC(kernel='linear')

# 학습
svc.fit(X_train, y_train)

# 예측
y_knn_pred = svc.predict(X_test)

# 평가
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_knn_pred)


# 객체 생성
knn = KNeighborsClassifier(n_neighbors=21)

# 학습
knn.fit(X_train, y_train)

# 예측
y_knn_pred = knn.predict(X_test)

# 평가
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_knn_pred)
```

```
# Ensemble(앙상블) - Voting

from sklearn.ensemble import VotingClassifier

# 객체 생성[(voting='hard') -> 다수결로 결정, (voting='soft') -> 비율(%)로 결정]
HVC = VotingClassifier(estimators=[('KNN', knn), ('SVM', svc), ('DT', DTC)], voting='hard')

# 학습
HVC.fit(X_train, y_train)

# 예측
y_HVC_pred = HVC.predict(X_test)

# 평가
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_HVC_pred)

# [voting='hard']인 경우, 정확도 0.9666666666666667
```

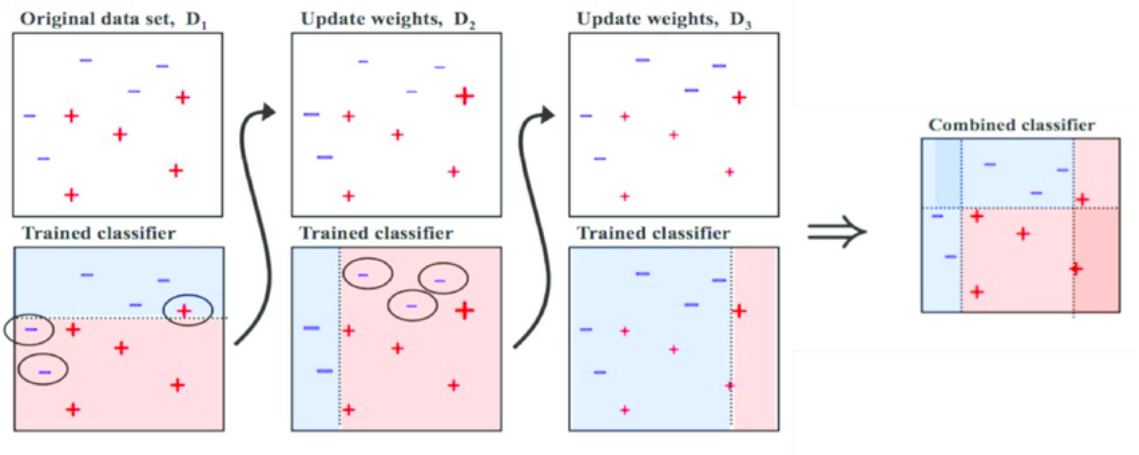
Ensemble(앙상블) - Bagging

```
# Ensemble(앙상블) - Bagging

# 예를 들면 한개의 DT(결정나무)만을 사용한다.
# 그 트리를 여러번 사용하여 각 모델의 개별 예측값을 추출한다.
# 그 예측값들에서 투표를 통해 결정한다.

# 이와 같이 같은 알고리즘 모델을 여러개 결합하여 예측하는 방법을 Bagging 이라고 한다.
# 각 트리는 전체 학습 데이터 중에서 서로 다른 데이터를 샘플링하여 학습하게 된다.
```

```
from sklearn.ensemble import RandomForestClassifier

# 객체 생성
rfc = RandomForestClassifier(n_estimators=50, max_depth=3, random_state=20)

# 학습
rfc.fit(X_train, y_train)

# 예측
y_rfc_pred = rfc.predict(X_test)

# 평가
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_rfc_pred)
```

Ensemble(앙상블) - Boosting

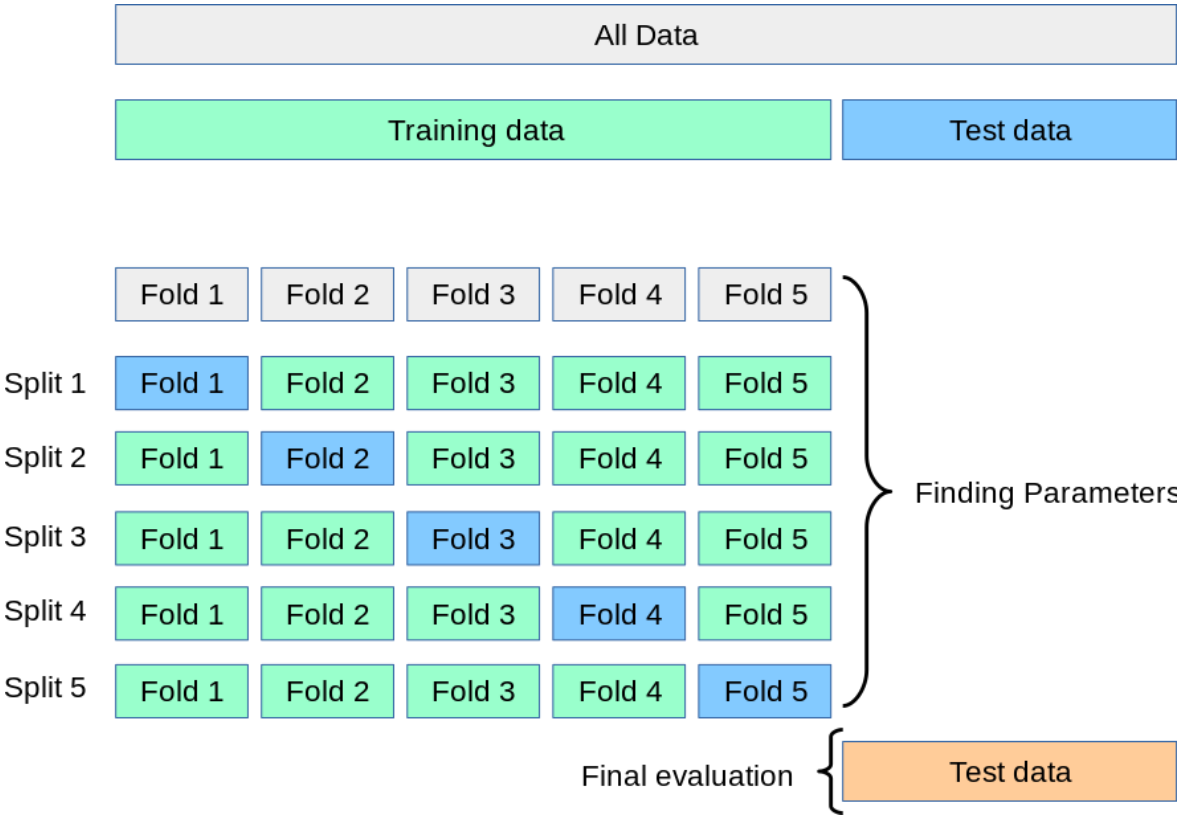
```
# Ensemble(앙상블) - Boosting

# 가장 많이 사용되는 알고리즘이다.
# 캐글, 데이콘 등의 경진대회에서도 가장 많이 사용된다.
# 학습 속도도 빠르고 예측 능력도 상당히 좋은 편이다.

# 부스팅(Boosting)은 여러개의 약한 학습기를 순차적으로 학습한다.
# 잘못 예측한 데이터에 대한 예측 오차를 줄일 수 있는 방향으로 모델을 계속 업데이트한다.
# 배깅(Bagging)은 여러 모델을 동시에 학습하여 예측하는 반면 부스팅(Boosting)은 순차적으로 학습하여 예측한다.
```




```
# !pip install xgboost

from xgboost import XGBClassifier

# 객체 생성
xgbc = XGBClassifier(n_estimators=3, max_depth=1, random_state=20)

# 학습
xgbc.fit(X_train, y_train)

# 예측
y_xgbc_pred = xgbc.predict(X_test)

# 평가
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_xgbc_pred)

# [n_estimators=50, max_depth=3, random_state=20]인 경우, 정확도 0.9333333333333333
# [n_estimators=3, max_depth=1, random_state=20]인 경우, 정확도 0.9
```

K - Fold

```
# K - Fold

# 1. 전체 데이터를 k개의 부분집합으로 나눈다.
# 2. 그 중 한 개를 검증용으로 사용하고 나머지를 훈련용으로 사용한다.
# 3. 그 다음에는 두번째를 검증용으로 사용하고 나머지를 훈련용으로 사용한다.
# 4. 1 ~ 3 과정을 반복한다.
# 5. 데이터를 겹치지 않게 분할하여 k번 검증하게 되므로 모델의 일반화 성능이 높아지게 된다.
```




```
from sklearn.model_selection import KFold

# 전체 데이터를 5개의 부분집합으로 분리한다(n_splits=5)
kfold = KFold(n_splits=5, shuffle=True, random_state=20)

num_fold = 1

for tr_idx, val_idx in kfold.split(X_train):
  print("%s Fold------------------------------" % num_fold)
  print("훈련:", len(tr_idx), tr_idx[:10])
  print("검증:", len(val_idx), val_idx[:10])
  num_fold += 1
```

```
# 각 Fold(폴드)별 정확도를 담을 리스트
val_scores = []

num_fold = 1

for tr_idx, val_idx in kfold.split(X_train, y_train):

    # 훈련용 데이터와 검증용 데이터를 행 인덱스를 기준으로 추출
    X_tr, X_val = X_train.iloc[tr_idx,:], X_train.iloc[val_idx,:]
    y_tr, y_val = y_train.iloc[tr_idx], y_train.iloc[val_idx]

    #학습
    rfc=RandomForestClassifier(max_depth=5, random_state=20)
    rfc.fit(X_tr, y_tr)

    #검증
    y_val_pred = rfc.predict(X_val)
    val_acc = accuracy_score(y_val, y_val_pred)
    val_scores.append(val_acc)
    num_fold += 1



val_scores

# 평균정확도

import numpy as np

np.mean(val_scores)
```

분류 & 회귀

```
# 분류 & 회귀


# 분류

# 분류는 문제(X)와 정답(Y) 사이의 관계를 찾았다.
# 특히 붓꽃 분류(0, 1, 2) 또는 타이타닉(0, 1)과 같이 정답이 연속적이지 않은 값. 즉, 이산적인 값을 찾았다.

# 회귀

# 회귀는 문제(X)와 정답(Y) 사이의 회귀 관계식을 찾는다[답(Y) = 연속적인 값을 갖는 숫자데이터]
# ex. 주가 예측, 집값 예측, 코인 예측, 중고차 가격 예측
```

보스턴 주택 가격 예측

```
# 보스턴 주택 가격 예측

import pandas as pd
import numpy as np
from sklearn import datasets
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# 기존 내장 데이터 셋에 인종차별 요소가 포함되어 있어 [load_boston()]이 삭제되었다.
# housing = datasets.load_boston()

# 페이지에서 가져오기(보스턴 주택 가격 예측)
data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]

data

target

data.shape

target.shape

data = pd.DataFrame(data)

target = pd.DataFrame(target)

df = pd.concat([data, target], axis=1)

df

# 피처값 설명

# CRIM    : 1인당 범죄 발생률
# ZN      : 주택 토지의 비율
# INDUS   : 해당 지역의 상업 지역 비율
# CHAS    : 찰스강의 인접 여부
# NOX     : 일산화질소 농도
# RM      : 방의 수
# AGE     : 오래된 주택의 비율
# DIS     : 5대 고용지역까지의 거리
# RAD     : 고속도로 접근성
# TAX     : 재산세
# PTRATIO : 교사 & 학생 비율
# B       : 흑인 거주비율
# LSTAT   : 저소득층 비율
# MEDV    : 소유주 거주 주택의 주택 가격(중간값)
# Target  : 집값


df.info()

df.isnull().sum()

df.columns =['CRIM','ZN','INDUS','CHAS','NOX','RM','AGE','DIS','RAD','TAX','PTRATIO','LSTAT','B','Target']
```

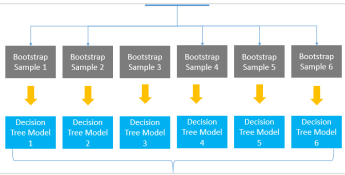
```
# 상관 관계 분석
df.corr()

plt.figure(figsize=(10,10))
sns.heatmap(df, annot=True)
plt.show()

plt.figure(figsize=(10,10))
sns.heatmap(df.corr(), annot=True)
plt.show()

# 집값(Target)에 가장 영향을 미치는 항목 순서

corr_order = df.corr().loc[:'B', 'Target'].abs().sort_values(ascending=False)
corr_order
```

Ensemble(앙상블)

머신러닝 - 11. 앙상블 학습 (Ensemble Learning): 배깅(Bagging)과 부스팅(Boosting)

앙상블(Ensemble) 앙상블은 조화 또는 통일을 의미합니다. 어떤 데이터의 값을 예측한다고 할 때, 하나의 모델을 활용합니다. 하지만 여러 개의 모델을 조화롭게 학습시켜 그 모델들의 예측 결과들을 이용한다면 더 정확한 예측값을 구할 수 있을 겁니다. 앙상블 학습은 여러 개의 결정 트리(Decision Tree)를 결합하여 하나의 결정 트리보다 더

https://bkshin.tistory.com/entry/머신러닝-11-앙상블-학습-Ensemble-Learning-배깅Bagging과-부스팅Boosting



Ensemble(앙상블) - Bagging

[머신러닝] 앙상블 학습 - 2) Bagging

지난 번 포스팅에서 앙상블 학습의 의미와 효과에 대해서 알아보았다. 이어서 이번 포스팅에서는 앙상블 학습 기법 중 한 종류인 Bagging에 대해서 정리하였다. 구체적으로는, 1) Bagging의 정의와 의의, 그리고 2) 모델 결합 방법에 대해서 알아본다. 본 포스팅은 고려대학교 강필성 교수님의 비즈니스 애널리틱스 강의 내용을 요약한 것이며,

https://sungkee-book.tistory.com/9

K-Fold Cross Validation(K-겹 교차검증)

K-Fold Cross Validation(교차검증) 정의 및 설명

정의- K개의 fold를 만들어서 진행하는 교차검증 사용 이유- 총 데이터 갯수가 적은 데이터 셋에 대하여 정확도를 향상시킬수 있음- 이는 기존에 Training / Validation / Test 세 개의 집단으로 분류하는 것보다, Training과 Test로만 분류할 때 학습 데이터 셋이 더 많기 때문- 데이터 수가 적은데 검증과 테스트에 데이터를 더 뺐기면

https://nonmeyet.tistory.com/entry/KFold-Cross-Validation교차검증-정의-및-설명