

Spring Day 13

▼ Spring

MyBatis 실습(CarShop)

▼ 환경설정

pom.xml 설정

```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.6</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.4</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>5.3.19</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-dbcp2 -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-dbcp2</artifactId>
  <version>2.7.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.bgee.log4jdbc-log4j2/log4jdbc-log4j2-jdbc4 -->
<dependency>
  <groupId>org.bgee.log4jdbc-log4j2</groupId>
  <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
  <version>1.16</version>
</dependency>
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.28</version>
</dependency>
```

root-context.xml 설정

```
<!-- mysql 연결 설정 -->
<bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close">
  <property name="driverClassName" value="com.mysql.cj.jdbc.Driver"/>
  <property name="url" value="jdbc:mysql://localhost:3306/difbfl4751?serverTimezone=UTC"/>
  <property name="username" value="difbfl4751"/>
  <property name="password" value="비밀번호"/>
</bean>

<bean id="sqlSessionFactory"
  class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="dataSource" />
  <property name="mapperLocations"
    value="classpath:/sqlmap/**/*.xml" />
</bean>
<bean id="sqlSessionTemplate"
  class="org.mybatis.spring.SqlSessionTemplate">
```

```

        <constructor-arg index="0" ref="sqlSessionFactory" />
    </bean>

```

servlet-context.xml 설정

```

<!-- 업로드 패스 설정 -->
<beans:bean class="java.lang.String" id="uploadPath">
<beans:constructor-arg
value="C:\develop\JSP\CarShop_MyBatis\CarShop_MyBatis\src\main\webapp\resources" /> <!-- 로컬에서 작업할 때 -->
</beans:bean>

```

▼ CRUD

car_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="car">

<insert id="insert" parameterType="com.carshop.controller.CardTO" useGeneratedKeys="true" keyProperty="cid" >
<![CDATA[
insert into car
(cid, cname, cprice, ccate, cdesc, cfilename)
values
(#{cid}, #{cname}, #{cprice}, #{ccate}, #{cdesc}, #{cfilename})
]]>
</insert>

```

CarController.java

```

@PostMapping("/add")
public String submitAddNewCar(@ModelAttribute("NewCar") CardTO car, HttpServletRequest request) {

    MultipartFile carimage = car.getCarimage();
    String saveName = carimage.getOriginalFilename();

    // String uploadpath = request.getRealPath("/resources/images");

    // File saveFile = new File("C:\\upload", saveName);
    File saveFile = new File(uploadPath + "\\images", saveName);
    // System.out.println(saveFile.getPath());

    if (carimage != null && !carimage.isEmpty()) {
        try {
            carimage.transferTo(saveFile);
            car.setCfilename(saveName);
        } catch (Exception e) {
            // throw new RuntimeException("차량 이미지 업로드가 실패했습니다.");
            e.printStackTrace();
        }
    }

    carService.setNewCar(car);
    return "redirect:/cars";
}

```

CarRepositoryImpl.java

```

@Autowired
SqlSessionTemplate sqlSessionTemplate;

```

```

@Override
public void setNewCar(CardTO car) {
    this.sqlSessionTemplate.insert("car.insert", car);
}

```

CardTO.java

```

package com.carshop.controller;

import java.io.Serializable;

import org.springframework.web.multipart.MultipartFile;

@SuppressWarnings("serial")
public class CardTO implements Serializable {
    //private static final long serialVersionUID = 3414129893883786050L;

    private String cid, cname, cprice, ccate, cdesc;
    private MultipartFile carimage;
    private String cfilename;

    public String getCfilename() {
        return cfilename;
    }

    public void setCfilename(String cfilename) {
        this.cfilename = cfilename;
    }

    public String getCid() {
        return cid;
    }
    public void setCid(String cid) {
        this.cid = cid;
    }
    public String getCname() {
        return cname;
    }
    public void setCname(String cname) {
        this.cname = cname;
    }
    public String getCprice() {
        return cprice;
    }
    public void setCprice(String cprice) {
        this.cprice = cprice;
    }
    public String getCcate() {
        return ccate;
    }
    public void setCcate(String ccate) {
        this.ccate = ccate;
    }
    public String getCdesc() {
        return cdesc;
    }
    public void setCdesc(String cdesc) {
        this.cdesc = cdesc;
    }
    public MultipartFile getCarimage() {
        return carimage;
    }
    public void setCarimage(MultipartFile carimage) {
        this.carimage = carimage;
    }

    public CardTO(String cid, String cname, String cprice, String ccate, String cdesc, MultipartFile carimage, String cfilename) {
        this.cid = cid;
        this.cname = cname;
        this.cprice = cprice;
        this.ccate = ccate;
        this.cdesc = cdesc;
        this.cfilename = cfilename;
    }

    public CardTO() {
        super();
    }
}

```

```
}
```

header.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<html>
<head>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbs
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIE4zV

<style>
  .bd-placeholder-img {
    font-size: 1.125rem;
    text-align: middle;
    -webkit-user-select: none;
    -moz-user-select: none;
    user-select: none;
  }

  @media ( min-width : 768px) {
    .bd-placeholder-img-lg {
      font-size: 3.5rem;
    }
  }

  .b-example-divider {
    height: 3rem;
    background-color: rgba(0, 0, 0, .1);
    border: solid rgba(0, 0, 0, .15);
    border-width: 1px 0;
    box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em
      rgba(0, 0, 0, .15);
  }

  .b-example-vr {
    flex-shrink: 0;
    width: 1.5rem;
    height: 100vh;
  }

  .bi {
    vertical-align: -.125em;
    fill: currentColor;
  }

  .nav-scroller {
    position: relative;
    z-index: 2;
    height: 2.75rem;
    overflow-y: hidden;
  }

  .nav-scroller .nav {
    display: flex;
    flex-wrap: nowrap;
    padding-bottom: 1rem;
    margin-top: -1px;
    overflow-x: auto;
    text-align: center;
    white-space: nowrap;
    -webkit-overflow-scrolling: touch;
  }
</style>

</head>
<body class="text-center">
<nav class="navbar navbar-expand navbar-dark bg-dark fixed-top" aria-label="Second navbar example">
  <div class="container-fluid">
    <a class="navbar-brand" href="/">CarShop</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample02" aria-controls="
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarsExample02">
      <ul class="navbar-nav me-auto">
```

```

<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="/">홈</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="/cars">차량보기</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="/cart">장바구니</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="/boards/list">게시판</a>
</li>
<li class="nav-item dropdown">
  <button class="btn btn-dark dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">
    회원관리
  </button>
  <ul class="dropdown-menu dropdown-menu-dark">
    <li><a class="dropdown-item" href="#">로그인</a></li>
    <li><a class="dropdown-item" href="#">로그아웃</a></li>
  </ul>
</li>
<sec:authorize access="hasRole('ROLE_ADMIN')">
  <li class="nav-item dropdown">
    <button class="btn btn-dark dropdown-toggle"
      data-bs-toggle="dropdown" aria-expanded="false">관리자메뉴</button>
    <ul class="dropdown-menu dropdown-menu-dark">
      <li><a class="dropdown-item" href="/cars/add">제품등록</a></li>
      <li><a class="dropdown-item" href="/cars/product">제품관리</a></li>
    </ul>
  </li>
</sec:authorize>
</ul>

<sec:authorize access="isAnonymous()">
  <form action="/cars/login" method="post">
    <input type="submit" class="btn btn-primary" value="login">
    <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
  </form>
</sec:authorize>
<sec:authorize="hasRole('ADMIN')">
<sec:authorize access="isAuthenticated()">
  <form action="/logout" method="post">
    <input type="submit" class="btn btn-success" value="logout">
    <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
  </form>
</sec:authorize>

</div>
</div>
</nav>

```

▼ CRUD

car_SQL.xml

```

<!-- parameterType(데이터 전송)을 설정하지 않아서 데이터를 넣지 않고 resultType(결과값)만 받는다. -->
<select id="select_list" resultType="com.carshop.controller.CardTO">
  <![CDATA[
    select * from car ORDER BY cprice DESC
  ]]>
</select>

<select id="select_detail" parameterType="String" resultType="com.carshop.controller.CardTO">
  <![CDATA[
    select * from car where cid = #{carid}
  ]]>
</select>

```

CarRepositoryImpl.java

```

@Override
public List<CardTO> getAllCarList() {
    return this.sqlSessionTemplate.selectList("car.select_list");
}

@Override
public CardTO getCarById(String carId) {
    return this.sqlSessionTemplate.selectOne("car.select_detail", carId);
}

```

▼ CRUD

car_SQL.xml

```

<update id="update" parameterType="com.carshop.controller.CardTO">

    UPDATE car SET cname=#{cname}, cprice=#{cprice}, ccate=#{ccate}, cdesc=#{cdesc}, cfilename=IFNULL(#{cfilename}, cfilename) WHERE

</update>

```

CarController.java

```

@PostMapping("/update")
public String submitUpdateCar(@ModelAttribute("updateCar") CardTO car, HttpServletRequest request) {

    MultipartFile carimage = car.getCarimage();

    if (carimage != null && !carimage.isEmpty()) {
        try {

            String fileName = carimage.getOriginalFilename();
            File saveFile = new File(uploadPath + "\\images", fileName);
            carimage.transferTo(saveFile);
            car.setCfilename(fileName);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    carService.update(car);
    return "redirect:/cars";
}

```

CarRepository.java

```

int setUpdateCar(CardTO car);

```

CarRepositoryImpl.java

```

@Override
public int setUpdateCar(CardTO car) {
    return this.sqlSessionTemplate.update("car.update", car);
}

```

CarService.java

```
boolean update(CarDTO car);
```

CarServiceImpl.java

```
@Override
public boolean update(CarDTO car) {
    int affectRowCount = this.carRepository.setUpdateCar(car);
    return affectRowCount == 1;
}
```

▼ CRU D

car_SQL.xml

```
<delete id="delete" parameterType="String">

    DELETE FROM car WHERE cid = #{carId}

</delete>
```

CarController.java

```
@ResponseBody
@RequestMapping("/removeProduct")
public void ajaxremoveProduct(@RequestParam String carId) {
    carService.remove(carId);
}
```

CarRepository.java

```
int delete(String carId);
```

CarRepositoryImpl.java

```
@Override
public int delete(String carId) {
    return this.sqlSessionTemplate.delete("car.delete", carId);
}
```

CarService.java

```
boolean remove(String carId);
```

CarServiceImpl.java

```
@Override
public boolean remove(String carId) {
    int affectRowCount = this.carRepository.delete(carId);
    return affectRowCount == 1;
}
```

▼ Python

DACON 데이터 분석 대회(Titanic)

데이터 살펴보기

```
import pandas as pd

train = pd.read_csv("train.csv")          # 학습용
test = pd.read_csv("test.csv")            # 검증용
submission = pd.read_csv("submission.csv") # 제출용

print(train.shape, test.shape, submission.shape)

# 데이터 살펴보기

# train 데이터프레임 내용 확인
train.head(3)

# test 데이터프레임 내용 확인
test.head(3)

# submission 제출파일 양식 확인
submission.head()

train.columns

# train 데이터프레임 개요 정보
train.info()

# 891개의 데이터를 확인할 수 있고 2개는 실수형(float) 5개는 정수형(int) 데이터타입이다.
# 일부열은 891개 보다 작은 것으로 보아 결측치가 존재함을 알 수 있다.

# train 데이터프레임 통계 정보
train.describe(include='all')

# include='all' 옵션을 사용하면 문자열도 일부 정보를 확인할 수 있다(생략하면 숫자데이터만 확인 가능)

# 결측값 분포

# !pip install missingno

import matplotlib.pyplot as plt

import missingno as msno

msno.bar(train, figsize=(10, 5), color=(0.7, 0.2, 0.2))
plt.show()

msno.matrix(test, figsize=(10, 5), color=(0.7, 0.2, 0.2))
plt.show()

# 숫자형 변수간의 상관관계를 계산하여 히트맵 그리기
import seaborn as sns

plt.figure(figsize=(8, 8))

sns.set(font_scale=0.8)
sns.heatmap(train.corr(), annot=True, cbar=True);
plt.show()
```


데이터 전처리

```
# 데이터 전처리

# 우선 기본 숫자 데이터만으로 모델을 만들어보자.

# 타이타닉 전체 데이터셋 준비
# 나중에 다시 분리하기 쉽게 'TrainSplit' 열을 만들어둔다.
train['TrainSplit'] = 'Train'
test['TrainSplit'] = 'Test'
data = pd.concat([train, test], axis=0)
print(data.shape)

# 데이터 분석을 위해 훈련용과 검증용 데이터셋을 하나로 합친다.

# 숫자형 특성(feature) 추출
data_num = data.loc[:, ['Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Survived']]

# 결측값 대체
data_num['Age'] = data_num['Age'].fillna(data_num['Age'].mean())
# 결측치를 전부 나이의 평균으로 넣기

data_num['Fare'] = data_num['Fare'].fillna(data_num['Fare'].mode()[0])
# 결측치를 전부 최빈값으로 넣기

msno.matrix(data_num, figsize=(10, 5), color=(0.7, 0.2, 0.2))
plt.show()

msno.bar(data_num, figsize=(10, 5), color=(0.7, 0.2, 0.2))
plt.show()

# 학습용 데이터와 예측 대상인 테스트 데이터 구분
selected_features = ['Pclass', 'Age', 'SibSp', 'Parch', 'Fare']

# 훈련용 문제지
X_train = data_num.loc[data['TrainSplit']=='Train', selected_features]

# 훈련용 답안지
y_train = data_num.loc[data['TrainSplit']=='Train', 'Survived']

# 검증용 문제지
X_test = data_num.loc[data['TrainSplit']=='Test', selected_features]

print("Train 데이터셋 크기: ", X_train.shape, y_train.shape)
print("Test 데이터셋 크기: ", X_test.shape)

# 학습 데이터를 훈련용 데이터 80% 검증용 데이터 20%로 분할한다.
# 100% 전체로 학습을 마치면 확인할 데이터가 없다.
# 따라서 20% 정도의 데이터를 검증용으로 확보한다.

# 훈련 - 검증 데이터 분할
from sklearn.model_selection import train_test_split

X_tr, X_val, y_tr, y_val = train_test_split(X_train, y_train, test_size=0.2,
                                           shuffle=True, random_state=20)
```

로지스틱 회귀 & 데이터 전처리2

```
# 결정나무(DT)
# from sklearn.tree import DecisionTreeClassifier

# 로지스틱 회귀(LR, Logistic Regression)
from sklearn.linear_model import LogisticRegression

lr_model = LogisticRegression()      # 객체 생성
lr_model.fit(X_tr, y_tr)             # 학습
y_val_pred = lr_model.predict(X_val)  # 예측

from sklearn.metrics import accuracy_score
print("Accuracy: %.4f" % accuracy_score(y_val, y_val_pred))

# [test_size=0.2, random_state=11]의 정확도 : 0.7374
# [test_size=0.2, random_state=20]의 정확도 : 0.7207

# 제출용 파일 만들기

# test 데이터에 대한 예측값 정리
y_test_pred = lr_model.predict(X_test)
```

```

# 제출양식에 맞게 정리
submission['Survived'] = y_test_pred.astype(int)

submission.to_csv("submission_0316.csv", index=False)

# 피쳐(feature) 엔지니어링

# 각 feature 값들의 변형을 통해 모델 성능이 개선될 수 있도록 여러가지 방법을 적용하는 것

# 타겟 레이블의 분포 확인
train['Survived'].value_counts(dropna=False)

# 결측값도 포함한다(dropna=False)

# 훈련용 셋의 생존여부 그림 확인
sns.countplot(x='Survived', data=data[data['TrainSplit']=='Train'])
plt.show()

sns.countplot(x='Pclass', hue='TrainSplit', data=data)
plt.show()

# 훈련용 데이터에서 Pclass별 생존 여부
sns.countplot(x='Pclass', hue='Survived', data=data[data['TrainSplit']=='Train'])
plt.show()

# 훈련용 데이터에서 성별 생존 여부
sns.countplot(x='Sex', hue='Survived', data=data[data['TrainSplit']=='Train'])
plt.show()

data['Sex']
# 학습이 가능하도록 데이터타입을 문자에서 숫자로 바꾼다.

# 레이블 인코딩(female: 0, male: 1)
data.loc[data['Sex']=='female', 'Sex'] = 0
data.loc[data['Sex']=='male', 'Sex'] = 1
data['Sex'] = data['Sex'].astype(int)

# 성별 분포 확인
data['Sex'].value_counts(dropna=False)

data['Sex']

# Name(이름)에서 Mr. Dona 등의 신분을 나타내는 정보를 추출.

data['Name'].unique()

title_name = data['Name'].str.split(" ", expand=True)[1]

title = title_name.str.split(".", expand=True)[0]
title.value_counts(dropna=False)

import numpy as np

title = title.replace(['Ms'], 'Miss')
title = title.replace(['Mlle', 'the Countess', 'Lady', 'Don', 'Dona', 'Mme', 'Sir', 'Jonkheer'], 'Noble')
title = title.replace(['Col', 'Major', 'Capt'], 'Officer')
title = title.replace(['Dr', 'Rev'], 'Priest')
data['Title'] = np.array(title)
data['Title'].value_counts(dropna=False)

# Name 열 삭제
data = data.drop('Name', axis=1)
data.columns

msno.matrix(data, figsize=(10, 5), color=(0.7, 0.2, 0.2))
plt.show()

# age 에 상당히 많은 수의 결측치를 확인할 수 있다.
# 결측치가 있는 데이터를 모두 삭제하게 되면 데이터 전체의 양이 너무 많이 줄어든다.
# 결측치를 나이의 평균으로 대체해도 나이가 너무 높아져서 데이터의 왜곡이 심해진다.
# 결측치를 최빈값으로 대체해도 나이 데이터가 0살부터 80살까지 넓게 분포가 되어 있으므로 최빈값도 의미가 없다.

#결측값 확인 및 대체
for title in data['Title'].unique():
    # 결측값 개수 확인(%s : 문자열 출력)
    print("%s 결측값 개수: " % title, data.loc[data['Title']==title, 'Age'].isnull().sum())

    # 각 Title의 중앙값으로 대체
    age_med = data.loc[data['Title']==title, 'Age'].median()
    data.loc[data['Title']==title, 'Age'] = data.loc[data['Title']==title, 'Age'].fillna(age_med)

msno.matrix(data, figsize=(10, 5), color=(0.7, 0.2, 0.2))
plt.show()

# [형제자매 & 배우자 수]와 [승객 나이] 및 [생존율] 관계
sns.boxplot(x='SibSp', y='Age', hue='Survived',
            data=data[data['TrainSplit']=='Train'])

```

```

plt.show()

sns.boxplot(x='Parch', y='Age', hue='Survived',
            data=data[data['TrainSplit']=='Train'])
plt.show()

# 가족 구성원의 수
data['FamilySize'] = data['SibSp'] + data['Parch'] + 1

data.loc[data['Fare'].isnull(),:]

# 상관관계분석에서 'Fare'와 'Pclass'의 상관계수가 가장 높았으므로 'Fare' 데이터를 데이터분석에 이용한다.

# 3등석 사람들의 요금의 평균
data.loc[data['Pclass']==3, 'Fare'].mean()

# 결측치 대체
data['Fare'] = data['Fare'].fillna(data.loc[data['Pclass']==3, 'Fare'].mean())

data.loc[data['Fare'].isnull(),:]

# Age 분포(히스토그램)
sns.displot(x='Age', kind='hist', hue='Survived',
            data=data[data['TrainSplit']=='Train'])
plt.show()

# Binning - 구간 나누기
bins = [0, 4, 8, 12, 16, 32, 36, 48, 56, 64, 100]
labels = ['Infant', 'Child1', 'Child2', 'Youth1', 'Youth2', 'Adult1', 'Adult2', 'Middle Aged', 'Senior', 'Elderly']
data['AgeBin'] = pd.cut(data['Age'], bins=bins, labels=labels)

# Age_bin(나이 구간)에 따른 생존율 비교
sns.countplot(x='AgeBin', hue='Survived',
              data=data[data['TrainSplit']=='Train'])

# x축 레이블을 반시계방향으로 45도 회전.
plt.xticks(rotation=45)
plt.show()

data['AgeBin']

data['Title']

```

인코딩 & DCON 제출

```

# 원핫인코딩

# 머신러닝에 학습을 시키기 위해서는 숫자 & 실수형 데이터만 필요하다.
# 문자 & 카테고리형 데이터를 숫자 & 실수형으로 변환해야 하는데
# 만약 종류가 10단계이거나 0-9까지의 숫자로 변환할 경우에 데이터분석을 진행하면 의도치 않게 계산결과 값이 커져서 데이터에 왜곡이 발생한다.
# 그렇기 때문에 원핫인코딩 등을 이용하여 문자 & 카테고리형 데이터를 계산 가능한 숫자 & 실수형 데이터로 인코딩해주어야 한다.

# 레이블 인코딩
from sklearn.preprocessing import LabelEncoder

for col in ['Title', 'AgeBin']:
    encoder = LabelEncoder()
    data[col]= encoder.fit_transform(data[col])

data['Title']

data['AgeBin']

data

data.columns

selected_features = ['Pclass', 'Sex', 'SibSp', 'Parch', 'Fare', 'Title', 'FamilySize', 'AgeBin']
selected_features

data['Pclass']= data['Pclass'].astype('float')
data['Sex']= data['Sex'].astype('float')
data['SibSp']= data['SibSp'].astype('float')
data['Parch']= data['Parch'].astype('float')
data['Fare']= data['Fare'].astype('float')
data['Title']= data['Title'].astype('float')
data['FamilySize']= data['FamilySize'].astype('float')
data['AgeBin']= data['AgeBin'].astype('float')

data.info()

train_pre = data[['Pclass', 'Sex', 'SibSp', 'Parch', 'Fare', 'Title', 'FamilySize', 'AgeBin', 'TrainSplit']]

```

```

train_pre

# 학습용 데이터와 테스트 데이터(예측 대상) 구분

y_train = data.loc[data['TrainSplit']=='Train', 'Survived']
X_train = data.loc[data['TrainSplit']=='Train', selected_features]
X_test = data.loc[data['TrainSplit']=='Test', selected_features]

print("Train 데이터셋 크기: ", X_train.shape, y_train.shape)
print("Test 데이터셋 크기: ", X_test.shape)

# 훈련 & 검증 데이터 분할
from sklearn.model_selection import train_test_split

X_tr, X_val, y_tr, y_val = train_test_split(X_train, y_train, test_size=0.2,
                                           shuffle=True, random_state=20)

print("훈련 데이터셋 크기: ", X_tr.shape, y_tr.shape)
print("검증 데이터셋 크기: ", X_val.shape, y_val.shape)

# 로지스틱 회귀모델(LR, Logistic Regression)
lr_model = LogisticRegression()
lr_model.fit(X_tr, y_tr)

y_tr_pred = lr_model.predict(X_tr)
print("Accuracy: %.4f" % accuracy_score(y_tr, y_tr_pred))

# 결정나무(DT)
from sklearn.tree import DecisionTreeClassifier

tree_clf = DecisionTreeClassifier(max_depth=12, random_state=12)

tree_clf.fit(X_tr, y_tr)

y_tr_pred = tree_clf.predict(X_tr)
print("Accuracy: %.4f" % accuracy_score(y_tr, y_tr_pred))

# 제출용 파일 만들기

# test 데이터에 대한 예측값 정리
y_test_pred = tree_clf.predict(X_test)

# 제출양식에 맞게 정리
submission['Survived'] = y_test_pred.astype(int)

submission.to_csv("submission_0316.csv", index=False)

```

mode()[0]

<https://community.dataquest.io/t/why-mode-0-not-just-mode/5057/2>


np.array()

numpy.array — NumPy v1.24 Manual

 <https://numpy.org/doc/stable/reference/generated/numpy.array.html>

과대적합(OverFitting) & 과소적합(UnderFitting)

과대적합(overfitting) 및 과소적합(underfitting) 개념(+Early Stopping)

 목차 1. 개념 1.1. 과대적합(Overfitting)이란? 1.2. 과소적합(Underfitting)이란? 2. 탐지 2.1. 분산과 편향 기반 탐지 2.2. 산점도 그래프 기반 탐지 2.3. 모델 복잡도 및 손실함수 기반 탐지 2.3.1. 과소적합 발생 구간 2.3.2. 과대적합 발생 구간 2.3.3. 최적의 모델 적합 포인트 3. 과대적합 방지방법: 학습 조기종료(Early Stopping) 1. 개념


🔗 <https://heytech.tistory.com/125>



과대적합과 과소적합의 해결방법

[머신 러닝] 과소적합(underfitting), 과대적합(overfitting)

편향과 분산 직선 모델은 너무 간단해서 복잡한 곡선 관계를 학습하지 못한다는 한계가 있습니다. 모델이 너무 간단해서 데이터의 관계를 잘 학습하지 못하는 경우 그 모델은 편향이 높다 라고 합니다. 모델의 복잡도를 높여서 트레이닝 데이터의 관계를 완벽히 학습했다면 그 모델은 편향이 낮다 라고 할 수 있습니다. 그렇다면, 편향이 낮은 모


 <https://justweon-dev.tistory.com/19>



원핫 인코딩(One-Hot Encoding)

[데이터 전처리] One-Hot Encodig 원핫인코딩

scikit-learn의 머신러닝 알고리즘은 입력값으로 문자열을 받는 걸 허락하지 않는다. 이를 위한 전처리 방법 중 하나인 원핫 인코딩. 고유값에 해당하는 컬럼에만 1, 나머지 컬럼에는 00으로 이루어진 벡터에 단 한개의 1의 값으로 해당 데이터의 값을 구별되러닝, 데이터

 <https://velog.io/@yeonheedong/데이터-전처리-One-Hot-Encoding-원핫인코딩>

이분류